Time Based Effects Algorithms for Real-Time Effects-Processing and their Musical Applications

Masters Project

Dafna Naphtali

1996

table of contents

introduction

Composers and performers are becoming increasingly interested in using interactive elements in their work. As a result, they face new decisions with respect to which pieces of equipment will best serve the compositional needs of their music on a practical and creative level. It has often been necessary for more than one version of a composition to be created so that a performer without access to an interactive or real-time system will still be able to perform the piece.

The greater availability and the decrease in price of powerful computer chips have recently made the use of real-time digital signal processing of audio as a compositional element much more commonplace. Processes and pieces of music that previously could only realized through the use of facilities at large institutions such as IRCAM or a university, are now realizable by the unattached composer or performer. Since specific pieces of equipment (however powerful) cannot always be available, it is important to ensure the "portability" of a particular work. To that end, it would be useful for a composer to specify all acoustic effects-processing in ways that transcend the hardware and software for which it was written.

Types of available real-time audio transformation systems fall into several categories: those that can only make time-based transformations and those that also have the ability to perform operations directly on extracted spectral information (e.g., analysis/resynthesis methods). All transformations made in the time domain have their effect in the frequency domain as well (see below) and visa-versa, but the implementations differ.

Other distinctions between systems are made on the basis of the physical design of the system and the flexibility of the software. There exists a continuum between a dedicated effects-processor controlled remotely through MIDI and a system such as the IRCAM Signal Processing Workstation in which both control and audio signals can be handled in the same highly flexible programming environment. Other systems including the FAR/COSMOS being developed at CNMAT, KYMA and the MARS workstation allow for real-time synthesis and control but off-line analysis for audio effects-primarily associated with real-time spectral manipulation.

In addition to a discussion of commonly used effects and their functional descriptions, I would like to categorize and analyze the musical uses of sound effects-processing in the work of several composers. In some compositions effects-processing has been used to complement the playing of the performer rather than to develop a separate and distinct voice in the piece. Using this "instrument paradigm" [Rowe 1991] as a compositional strategy, it may not be necessary for performance gestures to be tracked and analyzed by a computer (as is often done in other forms of interactive computer music), since the very nature of the effects-processing causes the output gestures to be very closely related to the input. Simply by playing louder or brighter the performer can have a large degree of control over the resultant sound in ways that are often not possible with MIDI. The "control intimacy" is greater by virtue of the fact that the control signal mapped directly to the output is a function of the sample rate, rather than the limited bandwidth available for the tracking of performance gestures with MIDI. [Loy]

Another way that an effects-processor might be used is to have the input audio signal generate a separate and distinct voice in the composition. Many different kinds of time-based and frequency domain transformations fall into this second category of possibilities. In the case of a more sophisticated and powerful system (e.g., DSP Max), not only can audio be processed in real time but a real-time spectral analysis of the audio signal can be made. The spectral information gathered about the performance (often complementing that gathered by way of MIDI) can be used to control parameters of complex synthesis algorithms, going far beyond simple envelope followers.

In the most elementary of examples, the creation of the second voice in the composition can be done using multiple delays to create rhythmic patterns. The performer generally has a degree of control over these patterns, but not nearly as much control as when the output is not rhythmically distinct from their own performance (as above). The second voice being rhythmically distinct is perhaps the minimum sort of processing needed to solicit this sense of a separate voice. Many other musical effects can be created combining several effects for this purpose.

This paper does not concern the gathering of gestural information, but rather the characterization of the resultant music in some way: be it generative, transformative or sequenced [Rowe 1991]. All of these behaviors are all possible given a model of an interactive music system that uses only the player's acoustical events as sound sources. The model differs from a system in which a separate synthesizer (although perhaps not of a sampler) generates pitches using its own distinctive sound(s). In the pieces of music discussed in this paper, only the performer's sound is used to create a complex accompaniment in a unique sonic art that can be at once highly abstract, holistic, and visceral.

In discussing the various types of transformations that are possible, I will give musical examples from my own compositions as well as those of other musicians working with various systems. I will try to answer the following questions:

- Is it possible or necessary to make a music specification for pieces that uses real-time acoustic signal processing that can transcend the hardware for which they were written?
- Is it worthwhile to write music for specific pieces of equipment?
- How does one handle the interesting sound transformations available on a particular piece of equipment that are anomalies to that system?

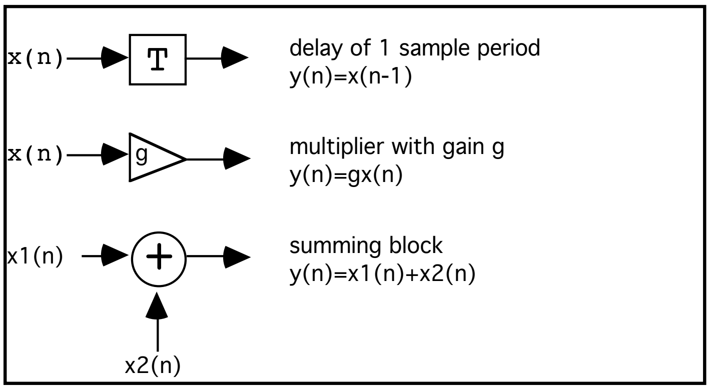
My own work has revolved around a dedicated digital effects-processor, (the Eventide H3000), connected to a computer running a Max patcher that controls the effects algorithms in real time with MIDI. Many of the sounds that I have created using the Eventide are unique to the instrument. Ironically, (as I discovered during the course of my work), some of these sounds are caused by errors or problems that are inherent to all digital effects' systems and specific to the design of the Eventide H3000. Other sounds are created by artifacts of the algorithm used for a particular effect. These sounds can often be recreated on a different system, given the precise parameters used.

In this paper I will attempt to classify the effect algorithms and associated musical behaviors and results that were used in my composition "Chew/Buzz/SLAM" and other pieces composed using the Eventide, in order to learn whether this piece can be ported either to another effects-processing platform or to a different model by Eventide (e.g., the DSP4000).

For each effect, I will include discussion of the basic function of the circuit rather than its mathematical and electrical construction. It is my purpose here to discuss the musical usefulness of the various algorithms and look at some typically modulated parameters and typical types of audio input and classify them in some way. For more technical descriptions of the design of the filters and delays mentioned here, please refer to the many papers on the subject listed in the bibliography.

basic components of signal processing

All digital signal processing circuits or algorithms comprise 3 basic digital components: multipliers, summers and delay elements. Various configurations of these operators are used for the many kinds of musically useful audio processing techniques; filtering, delay, reverb, equalization, compression, limiting, and noise reduction. In the case of compressors, limiters and noise reduction algorithms, the algorithms also contain a non-linear component such as a comparator for choice making.



Digital signal processing algorithms are mathematical operations that act upon as a series of numbers (amplitude values). They allow for more complexity and greater speed of operations than the equivalent analog operations. Circuits that are too complex for analog circuitry can be more easily realized with digital designs. The sounds created as output from these digital signal processing circuits therefore have a uniquely "digital" nature.

Relatively powerful processing is needed for any degree of complexity in an algorithm. In any "sample synchronous" operation, all the processing for each sample must be completed within one sample period (approximately 20 microseconds at 44.1K sampling rate). This high computational overhead makes complex digital signal processing algorithms only realizable on very powerful computers or specialized chips.

basic components: digital filters

The frequency spectrum of an audio signal is one of its most easily recognizable features. Because of this, the ability to alter that spectrum can be a very powerful musical tool. This can be done using a basic component of any effects-processor: the filter. [Pohlmann, 1991]

The goal of any filter design is to alter the signal's spectrum by emphasizing or de-emphasizing particular frequencies of the filter's input or by altering its phase. [Moore, 1990 p. 42] A typical analog filter is made up of a network of components that is designed to block or to attenuate certain frequencies by using the physical properties of capacitor and resistor combinations. [Horn, 1989]

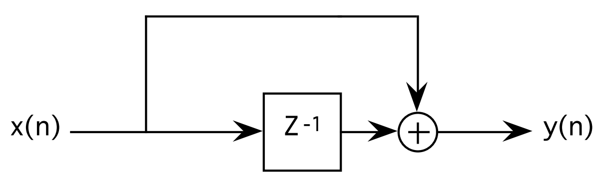
In contrast a digital filter does a computation on a digital input signal (which is just a sequence of numbers) and produces a new sequence of numbers from it as its filtered output signal. [Smith, Tutorial on Digital Filters]

For the purposes of this paper, all digital filters discussed are assumed to be linear time-invariant filters, meaning they do not change their frequency response over time, and that the filter's output is not changed if several input signals are filtered prior to being summed or summed prior to being filtered.

Julius Smith said that "any medium through which a musical signal passes... can be regarded as a filter," since the sound is modified in some way. In the case of a digital filter, the sound is modified by the phase cancellation that occurs when delayed copies of a signal are added to the original.

Delay is used to create any basic digital filter. By averaging the sum (or difference) of consecutive samples and multiplying those samples by a gain of less than one (unity), the creation of a high or low pass filter becomes a relatively simple mathematical operation. The delay can be one sample long or much longer, depending on the design and purpose of the filter. [Pohlmann, 1989]

basic design of digital filter:



A digital filter that averages the sum of the current input sample with the previous sample will smooth the output signal, and attenuate high frequencies. At high frequencies, transitions from the positive to negative portions of a waveform (zero crossings) happen quickly (in as few as 2 samples at the Nyquist frequency), causing the overall amplitude of these samples to be lowered by the averaging process.

If the same filtering algorithm were applied with the delayed sample being subtracted rather than added to the input (a negative gain on the feedback loop) the opposite effect would result, and a high pass filter would be created.

All *linear time-invariant* digital filters can be described by a "difference equation" that indicates the relationship between the input and feedback gain values.

$$y(n)=ax(n) + by(n-1)$$

x = input   y = output

a = input gain   b = output gain

Summing a delay with its input not only impacts on the timing of the output but also on its frequency spectrum. This is due to phase cancellation that occurs between the 2 signals.

Filters can attenuate high or low frequencies and in various configurations be used to selectively "bandpass" or "notch" out certain portions of the spectrum. The sharpness of the "cutoff" of the filter at its "filter cutoff" frequency is determined by the "order" of the filter.

"The maximum time span of a filter, expressed in samples, used to create an output value determines the order of the filter." The low-pass filter described above was a first order filter.

A stable filter is one in which the gain element is less than unity, therefore the impulse response approaches zero as n goes to infinity.

impulse response/convolution:

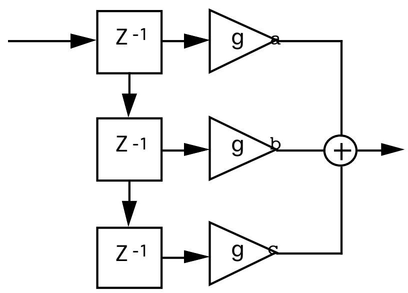
The impulse response of a filter gives a clear description of the change in spectral quality and any phase-shift that is introduced to a signal that is passed through it. An impulse is an infinitely small burst of noise (perhaps 1 sample), which is input into the filter. The output of the filter caused by the input of this impulse will reveal its frequency response. This is much the same as sending an impulse into a reverberant room (such as clapping one's hands) to test the "sound" of the room. The amount of time the sound takes to decay, the spectral resonances of the room, all constitute what is commonly referred to as "room tone". The impulse response is the equivalent description of the filter.

The impulse response itself can be sampled, and used to filter other signals. "A digital filter changes the frequency response of a signal by replacing each signal sample with a scaled replica of the filter impulse response." [Pohlmann, 1989]

FIR filters:

A filter designed to temporarily store and process a large and finite number of samples is called a Finite Impulse Response (FIR) filter.

y(n)=ax(n)+bx(n-1)+cx(n-2)+dx(n-3)..., etc.



FIR filters are non-recursive: there is no feedback of the input signal. Several delay elements will be typically implemented. These delay elements can be treated separately or can be ganged together to create a Multitap delay line, the longest tap defining the length of the delay line.

Transversal filter - the output value of this filter depends not only on the input, but also on several previous input values that are stored in a tapped delay line without a feedback loop.

The filter is always stable, and can have a linear phase response (i.e., the change in phase is the same at all frequencies), which is important for most audio applications.

IIR filters:

IIR filters use feedback loops to recall past samples. An example is the "exponential time average filter", in which the current input value is added to the last output value (not the previous sample) and divided by 2. The feedback loop makes the filter equivalent to an FIR filter that has an infinite amount of memory for storage of previous samples. The impulse response lasts for an infinite amount of time, and the response never completely decays to zero.

The filter can have both poles and zeros (resonances and anti-resonances). It can introduce a phase-shift into the signal. It can also be unstable under certain circumstances.

Linear phase cannot be realized in the IIR filter except when all poles in the transfer function lie on the unit circle (i.e., when the filter is actually a series of cascaded first order filters).

The IIR filter is recursive; meaning it has a feedback loop. The feedback loop must have a delay element in order for the value being fed back to be known before it is calculated.

Poles or resonances are caused by the feedback network as in the structure of the IIR filter. Zeros or anti-resonances in a filter's response are caused when input samples are summed in a "feedforward" network as in the FIR filter. A filter's order is determined by the number of poles and zeros it has, "whichever is greater". Various combinations of structures are possible to create for a variety of sophisticated purposes.

gain/stability/spectrum:

In an FIR filter, the relationship between the gain elements (of the direct signal and gain of the signal being delayed), the delay time(s) employed, as well as whether the output of the delay section is added or subtracted from the direct signal, all play a role in determining the spectral quality of the filter (where the resonances and anti-resonances are).

In IIR filters, the relationship between the gain elements (of the direct signal, and the signal being fedback), as well as the polarity of the feedback network, both play a role.

sources of errors/musically useful problems:

- **Coefficient errors**

The coefficients of a filter's difference refer to the value of the gain elements (multipliers) being applied to either a feedback network or to input samples being summed together (as in the FIR filters). If the values of the coefficient are not chosen carefully nor specified with sufficient accuracy (because of low bit resolution or roundoff error), noise and distortion will result. Errors such as these could cause an otherwise stable filter to become unstable, as poles and zeros are shifted slightly out of alignment. As many as 24 bits may be needed for the accurate calculation of values needed for 16-bit audio. [Pohlmann, 1989]

- **Limit cycle errors/oscillation**

When a signal is removed from a filter, it will leave a decaying sum. If the sum does not go to zero (as in the case of all IIR filters), it may oscillate at a constant amplitude (limit cycle oscillation). This can be eliminated by offsetting the filter's output so that truncation can set the output to zero after a period of time or when the value reaches some predetermined minimum.

- **Truncation/quantization error**

If the word length of a system is limited in some way, its A/D converter may introduce quantization errors. For example, if 2 n-bit numbers are added together to yield an n-bit word, it is easy to predict if there will be overflow. Using a multiply (as in the case of the gain elements) can be more dangerous and may <u>double</u> the number of bits required (even for the calculation). Usually this cannot be accommodated with a finite word length, as in a fixed point DSP chip, most often found on outboard dedicated effects-processors. The output sample value will need to be either rounded off or truncated resulting in quantization noise. All multiply operations should be followed by quantization.

- **Roundoff error**

Roundoff error results in a maximum error of 1/2 of the LSB of a sample with an RMS value of .288 LSB. If this error is allowed to accumulate over successive calculations, dithering information can be lost.

- **Overflow**

Exceeding the length of the register, will yield a computational error. "Wraparound" occurs when a 1 is added to a maximum value's positive 2's complement number, resulting in the maximum negative value. The overflow to a non-existent bit causes a sudden and large change in the amplitude of the output waveform, yielding a loud pop (e.g., in a loudspeaker). This can be avoid by setting the output to the maximum positive value whenever 2 positive numbers being added together result in a negative number. A side effect of this remedy however is that the signal would then be clipped. Clipping, however, is more desirable than the first discontinuity mentioned -- in other words, even in the digital "headroom" is needed and available given enough bits.

- Gibbs phenomena

A very fast transition must result in ringing or overshoot from the target value -- which could be caused by the truncation of an impulse response.

basic components: time domain effects

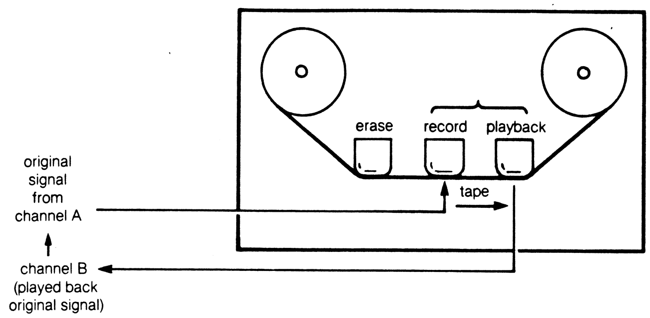
Although all alterations made to a sound will have their effect upon that sound's spectrum, effects that are made upon a sound without its prior conversion to its spectral representation are considered "time domain effects".

time domain effects: delay

The simplest and most commonly used of these is a simple delay. As a musical device, the echo has been used in folk music around the world for centuries: from Swiss yodels to African call and response, and songs in round. In contemporary music delay and reverberation effects have been used in acoustic performances as well through the use of digital delay systems. Examples that come to mind are the cavern music of Pauline Oliveros, and early tape experiments of John Cage and later, Steve Reich.

Early delay systems

The earliest method used to artificially create the effect of an echo or simple delay was to take advantage of the spacing between the record and playback heads on a multitrack tape recorder. The output of the playback head could be read by the record head and rerecorded on a different track of the same machine. That signal would then be read again by the playback head (on its new track). The signal will have been delayed by the amount of time it took for the tape to travel from the record head to the playback head.



[Woram, 1989 p. 227]

The delay time is a function of the spacing between the tape heads and the tape speed being used. The problem with this method is that delay times are is limited to those that can be created at the playback speed of the tape. At 15ips, tape heads spaced 3/4" to 2" create echoes at 50ms to 133ms; at 7ips, 107ms to 285ms.

[Woram, 1989 p. 222]

DN 11/7/12 8:56 AM
Comment [1]:

By using a second tape recorder a longer sequence of delays can be produced. Even with an auxiliary tape recorder, however, there are remain problems with using tape delays if the desired output is the emulation natural echoes and reverberation:

- all the delays will be integer multiples of the first delay
- the level of the delayed output decreases linearly over time. (in a natural acoustic environment the delays would decay more exponentially)

A side effect of creating the delays by rerecording the material is that after several repetitions the signal begins to degrade affecting its overall spectral qualities. If the feedback is slightly higher than unity, the midrange frequencies are emphasized on each repetition as the high and low frequencies die out more quickly. If the sound is shorter than the delay time, it eventually degrades into a "howl with a periodic amplitude envelope." [Chamberlin, 1985 p. 47]

A degree of unpredictability enhances the use of any musical device including echo and delay. By using a digital delay system that is designed and configured to that end, one can overcome the inherent inflexibility and static quality of most tape delay systems.

Digital Delay

Digital delay enables a greater range of delay times than the use of tape delay does. Multiple outputs and variable delay times are also easier to produce. In a digital system, the maximum delay time is a function of the available memory, whether it be on a general purpose computer or a dedicated piece of "outboard" equipment. In some cases the available delay time must be shared by 2 channels of the system (i.e., 2 channels of 500ms each = 1000ms delay time available)

Since the delay is the most basic component of a digital effects system it is important to discuss it in greater detail before moving on to some of the effects that are based upon it. There are some physical and perceptual phenomena that need to be considered at this point as well.

- Basic Design of a Delay

"A delay block is a simple storage unit such as a memory location. A sample is placed in memory, stored, then recalled when needed some time later, and output." [Pohlmann, 1989]

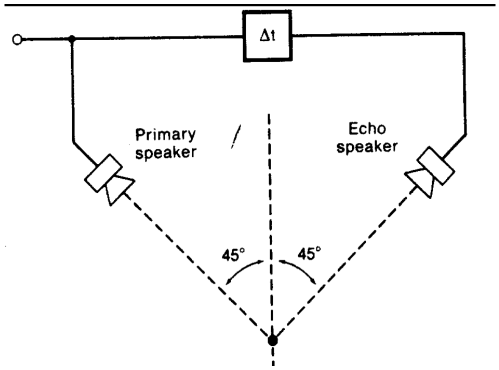
The delay component is used both in digital filters and in delay and reverberation applications. One important distinction between the two applications is the size of the delay. In the case of the digital filter the delay is only 1 sample and in all other delay applications it can be greater. As described below, (Haas effect) when the delay time is short, filtering occurs, and at longer delay times an echo is heard.

Perception of Delay - Haas Effect

Helmut Haas in is article "The Influence of Single Echo on the Audibility of Speech" (1949) made some observations about human perception of simple delay at various delay times, using a speaker system in the diagram below. He noted that when a delay reaches 10ms an echo is no longer perceived as a discrete event, even if the amplitude is the same as the direct signal. The "perceived sound source in never confused by the arrival of echoes". [Haas, 1949]

This observation, and others below are useful when considering the possible uses of delay in a musical context that are under discussion in this paper.

  - 0-10ms delay - Haas said that "as the delay is increased from 0 to 10ms the sound source appears to move to the location of the primary speaker" [the sound source]. He noted further that the center image is not restore until the amplitude of the delayed signal is raised 10dB.



[Woram, 1989 p. 224]

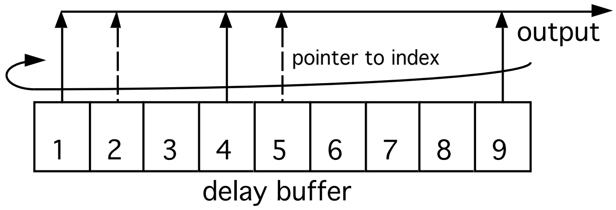
- 10-30ms delay - the sound source is still heard as coming from the primary speaker, with the echo adding a "liveliness" or "body" to the sound. In a concert hall, listeners become aware of the reflected sounds but not discern their direction of arrival.
- 30-50ms delay - the listener becomes of the delayed signal, but still senses the direct signal as the primary source.
- 50ms or more - a discrete echo is heard.

Multitap Delay

The design of the simplest digital delay line involves samples of a digital signal that are "passed at the sample rate from one stage to the next". [Moore, 1990 p. 362] Delay is easily produced with a small memory buffer.   The size of this memory buffer determines the total available delay time. Unfortunately, this technique can only produce delays that are a multiple of the sample period,

Since many delay applications require more than one delayed signal to be played concurrently, multitap delay lines are useful for creating several echoes or delays of varying length.

For fixed delay times multiple delays can be output on a sample by sample basis using a circular buffering technique. Rather than copying samples into successive locations in an output queue, indices are used to keep track of the current sample in its place in the delay buffer. The delay function will increment the index and wrap around to the beginning of the delay line as needed. This buffering method can keep the amount of computation constant regardless of the length of the delay.

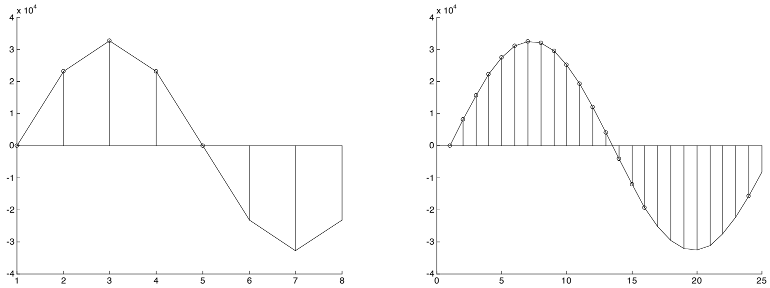


Time varying delay - interpolating delay lines

For fixed delays of an integral number of samples, (multiples of the sample rate), a simple non-interpolating delay line is sufficient. It is generally desirable, however, to use delay times that are variable for musical applications. To avoid the discontinuities and audible clicks that would be introduced in the output signal, a time-varying delay requires an implementation involving some sort of interpolation.

Linear interpolation, although much used by manufacturers of outboard effects-processing units, is not ideal. This is because in many cases the underlying (input) sampled signal does not change in a linear fashion. This is especially true at higher frequencies. Linear interpolation distorts high frequencies more than low, when compared to the Nyquist rate, because each individual wave cycle is sampled less frequently at high frequencies.

If an input signal has been oversampled (sampled at a high rate), the straight line between the samples of the linearly interpolated signal will more closely resemble its original, smoother spectrum or shape. The ideal interpolator is like an ideal low-pass filter: passing all signals to the critical sampling frequency and acting as a brick-wall filter above it, to avoid adding harmonic distortion.
Although linear interpolation on an audio signal increases the signal to noise ratio as compared to a signal that has not been interpolated, a higher order interpolation in needed to replicate more closely the behavior of an actual audio signal.

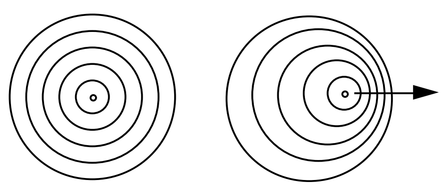


At a sample rate of 44.1KHz, an 11025KHz sine wave will be sampled 8 times per sample period; a 1764 Hz sine wave will be sampled 25 times per period, more closely approximating the underlying waveform, whether they are actual samples or some are interpolations.

Doppler shift

An interesting feature of the interpolating delay line is the characteristic pitch shift is produced. To understand the reason for this pitch shift it is first necessary to understand the Doppler shift phenomenon.

A stationary sound source normally propagates sound energy in a uniform way, in all directions around itself, at the speed of sound. If that sound source then begins to move toward a stationary listener (or visa versa), the successive wavefronts that were formerly encountered (by the listener) at a certain rate are compressed in time and encountered with greater frequency. Due to the relative motion of the sound source to the listener, the sound's frequency has in effect been raised. If the sound source instead moves away from the listener, the opposite holds true: the wavefronts are encountered at a slower rate than previously, and the pitch seems to have been lowered. [Moore, 1990]



[Moore, 1991 p. 367]

"a waveform in a delay buffer is a record of its recent history...the delay value may be interpreted in terms of an equivalent distance... between the sound source and the listener". [Moore, 1990]

When successive delayed sounds are varied in time an auditory illusion is created: the distance between the sound source and the listener seems to being changing. This could happen during the playback of a particular sound (while it is still present in the delay line). The distance between successive peaks in the waveform would then become greater as the delay time is increased and become smaller as delay time is decreased, with the same Doppler effect as the case of the stationary listener and moving sound source.

Another property of the interpolating delay line to consider is that the delayed signal is <u>not</u> being resampled at a lower rate when the delay time is increased but rather interpolation between existing samples is implemented since information about a waveform in a delay line is limited to the original sample rate.

time domain effects: comb filters

Ricard Strauss often used an orchestration device to bring about motion in a static chord. By having the instrumentalists in a choir interchange their notes in a rhythmic pattern while maintaining the overall balance of the chord at all times, he created a musical device that is not unlike the chorus effect.

Chorus

Hearing a signal combined with a delayed replica of itself (in the range of 0-50ms) may lead a listener to believe they are hearing a "unison duet". This "doubling" or "chorus" effect is more effective, however, if the two signals are continuously varied in both in onset time and amplitude.

In the case of an actual duet, the fundamental overtone varies slightly between the two signals resulting in a beating as phase cancellation occurs between the two signals. The "slight, random spectral modulation" is a natural result of slight differences in the pitch of the fundamentals and further intensified by the use of vibrato.

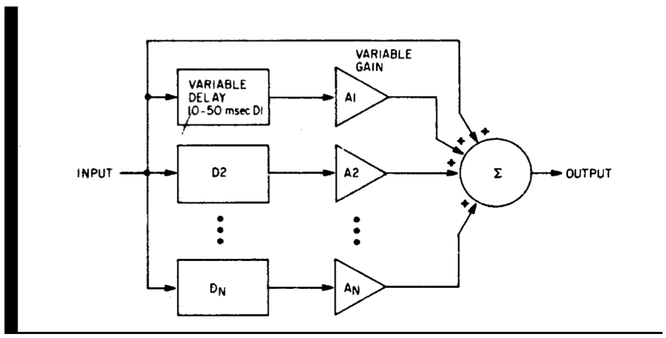
"The resultant amplitude of two beating complex waves remains relatively constant, but individual harmonics may momentarily cancel." [Woram, 1989]

As more and more copies of the signal are added together (to simulate more instruments), "changes in the amplitude and phase of the component harmonics become more random and pronounced." The overall spectral envelope is unchanged, however, enabling the listener to discern the difference between various types of audio input. [Chamberlin, 1985]

Some analog and digital simulations of chorus include a low-frequency oscillator to modulate the amplitude and delay time over several milliseconds. [Woram, 1989]

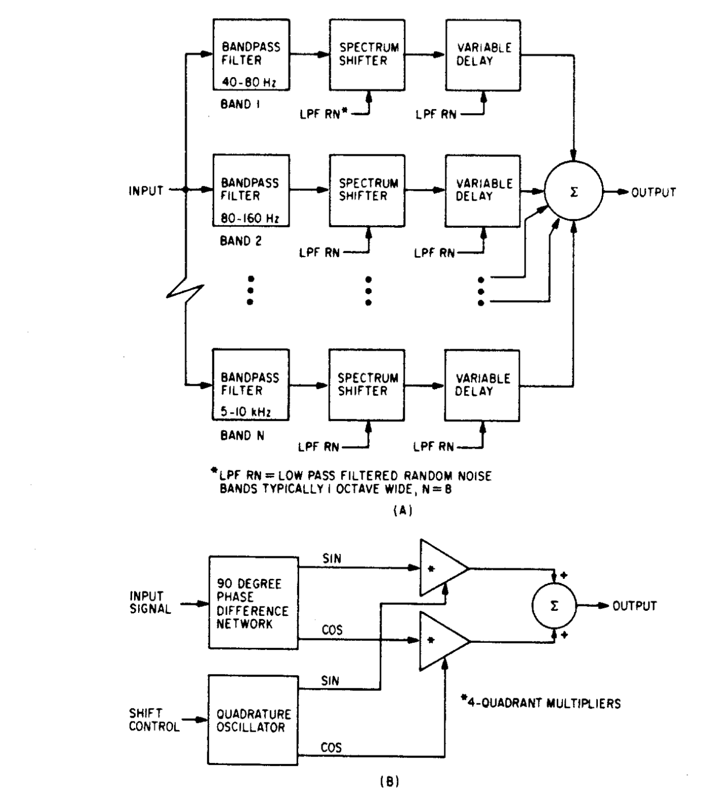
Two basic designs for chorus effect are:

- Direct Chorus simulation - directly simulates multiple sound sources with a number of audio delay lines, constantly modulating the delay times over a narrow range (10-40ms). The modulation of the delays creates slight pitch changes, as well as phase cancellation. The number of delays can easily be changed to enable the simulation of small or large groups of voices or instruments.



[Chamberlin, 1985 p. 513]

- Parametric Chorus simulation - uses filters to separate the input spectrum into frequency bands. Amplitude and phase modulators plus spectrum shifters are next used to randomly manipulate the individual bands. The sounds are added back together and then output. This method is not as effective for simulating small groups of players.



[Chamberlin, 1985 p. 514]

Flange

Flanging is related to chorus, insofar as it resulted from an "early attempt at doubling". The output from the sync head of one machine was input to the record head of another, and by mixing the playback of both, delay was introduced. By pressing on the supply reel of one of the machines, it is possible to slightly vary its the playback or record speed.

As in chorusing, a signal is combined with a delayed replica of itself, reinforced some frequencies canceling others. [Woram, 1989 p. 237] The result is a comb-filter frequency response, which can be varied over time and "swept" up and down the frequency spectrum.

- Basic Design of Flange

Digital flanging can be accomplished by continuously and slowly varying delay time on a copy of a signal that is mixed with the original at the output. Delay times commonly used range from 0-20ms. "Negative flanging" occurs when the signal is subtracted rather than added to its replica. [Alten, 1994]

Because of the continuous variation of delay time, and because delay times that are not multiples of the sample rate are likely to be needed, the flange effect may become "choppy" if an interpolating delay is not implemented. Chamberlin points out that this is necessary in spite the resulting increase in the noise level. [Chamberlin, 1985 p. 499-500]

Phase-shift

Phase-shift is a very similar effect that uses a phase shift instead of a time delay circuit. Peaks and dips in the spectrum are more irregular and farther apart than in flanging. The sound produced has been described as a "pulsating, wavering vibrato, underwater."

Karplus-Strong algorithm / short delay times

The Karplus-Strong algorithm produces "sounds with spectra that vary in constant, rapid, and varied ways" [Moore, 1991 p. 279] It is computationally efficient, and is very effective for simulating plucked string and drum sounds.

The algorithm consists of a recirculating delay line with a filter in its feedback loop. The delay line is filled with **n** samples of random amplitude (noise). As the samples are recirculated through the filter in the feedback loop, a few interesting things happen that have potential musical value. The **n** samples that are passed through the delay line create a "periodic sample pattern" with a period of **n**. Although the input signal is pure noise, a "steady complex sound" is heard with pitch content that is related to the period of this sample pattern "or $\frac{R}{n\text{Hz}}$ where R is the sample period."

* Periodicity Pitch

"This subjective effect is known as periodicity pitch and has been well studied in the field of psychoacoustics (see, for example, Roederer, 1975). `Even pure, continuously varying white noise can exhibit periodicity pitch if it is presented to a listener along with a delayed copy of itself. This effect works even when the undelayed and delayed white noise is presented dichotically-that is, separately to each ear." [Moore, 1990 p. 280]

The filter in the feedback loop has two effects upon the signal: it both filters and further delays the signal.

A low pass filter put in the feedback loop serves to "rob" the noise of a little of its high frequency components at each pass through the circuit, until all that remains is a pure sinusoid, which then decays to zero. This is replicates the acoustical properties of a plucked string or drum, in which a "complex, harmonic-rich sound.. gradually decays to a near-sinusoid." The filter also adds a certain amount of delay to the feedback loop, affecting the perceived pitch of the sound. [Moore, 1990 p. 280]

The decay time of the signal in this type of recirculating delay is dependent upon the fundamental frequency "associated with the total loop delay." It also depends upon the harmonic being considered: upper harmonics decay (much) more quickly than does the fundamental, and high frequencies (much) more rapidly than low.

time domain effects: reverb

"Perceptual studies have demonstrated that listeners judge their environments and distance from sound primarily by analyzing the accompanying reverberation." [Chamberlin, 1985 p. 500]

Reverb is the "simplest, yet most effective" acoustic signal processing function. Nearly any kind of reverberation is possible and transitions between different kinds of reverb are not difficult to make. Listeners are sensitive to very small changes in a reverberated signal. Because of this the design and parameter values needed for a good reverb are as much art as science. Altering cues sent to a listener about directionality and environment, is somewhat abstract as a compositional tool. However there are several composers who use reverberation in compositional ways, and many of the parameters if set using unorthodox values, can create unusual (and not at all natural) sounds.

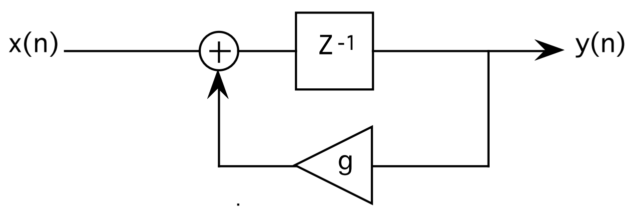
Typical programmable parameters of digital reverb:

• Early reflections	arrival time for one or more reflections direction of arrival amplitude of each reflection frequency response
• Echo clusters	arrival time of 1st cluster reflection direction of arrival amplitude of 1st cluster decay of amplitude within cluster frequency response
• Reverberation	Delay before onset Arrival direction (usually of 2 returns) Amplitude at start of reverberation Frequency response Reverberation time (maybe adjustable by freq.) Density (i.e. number and spacing of reflections)

[Woram, 1989]

Digital Reverb

A simple (digital) reverb can be a delay of 30ms or greater mixed with the undelayed sound. With a feedback loop added, multiple echoes can easily be created. [Woram, 1989 p. 225]



[Moore, 1990 p. 280]

The "magnitude of delay and the relative amplitude of direct and delayed sound are parameters for the echo." The closer the gain of the feedback to 1, the longer the echoes will go on, with no loss of fidelity as in an analog system. [Chamberlin, 1985 p. 508]

Actual reverb is not as "echoey" as what would result from the simple digital reverberator described above. Natural reverberation is more like "white noise with a smoothly decreasing amplitude. The amplitude decrease approximates an inverse exponential function, ... a constant number of decibels per second." [Chamberlin, 1985 p. 500]

Rate of decrease is described as the time required for a 60 dB attenuation in reverb amplitude. Concert halls generally fall between 1.5 and 3 seconds.

Echo density - a single delay line reverberator "suffers from a low (and constant) echo density of 0.03 echoes/msec." "In a concert hall the echo density builds up so quickly that no echoes are perceived"

The time between the initial signal and when the echo density reaches 1/msec is a measure of the effectiveness of an artificial reverb. This should take about 100ms. A delay of 10ms to 20ms (Early reflections) between the source and the first echo if needed as well if the sound is to be perceived as nearby.

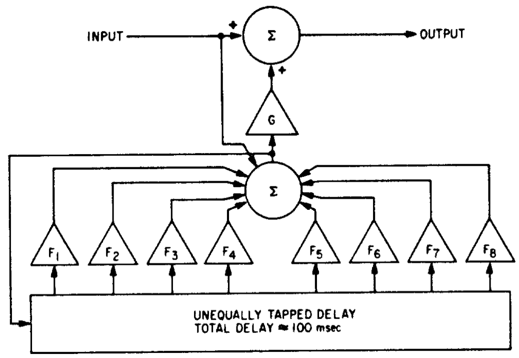
If the reverberation amplitude decrease is not smooth, a different and unnatural echo effect is perceived. Concert hall reverb has an "uneven amplitude response"; peaks and notches in its spectrum are closely spaced, irregular and not very high or deep. "It is not unusual to find several peaks and valleys per hertz of bandwidth with an average difference between peak and valley of 12dB."

Some examples of expected behavior are:

> High echo density with low resonance density creates the sound of an empty locker room or metal garbage can. The "small size of the reverberant chamber precludes resonant modes spanning a large number of wavelengths of moderate frequency sound."
>
> Low echo density with high resonance density is "produced by the feedback delay line reverberator with a very long delay time, which does not sound like reverberation at all." This is because of the perceived separateness of the source and reverberated signal (see Haas effect).

It is important at this point to discuss some reverb configurations and their possible compositional uses:

Tapped delay line digital reverberator :



[Chamberlin, 1985 p. 510]

"Just about any assemblage of delays, summers, and multipliers will produce some kind of reverberation if it doesn't oscillate instead." [Chamberlin, 1985 p. 510] The oscillation found at the limits of acceptable reverberation settings, if controlled can be used as a compositional element. Movement between distinct echoes and reverberation of greater echo density can also be effective.

To increase echo density, several delays of unequal length are used. The placement of the taps and values of the feedback constants are very important in determining the sound of the system.

"The taps should be exponentially distributed, but placed at prime number locations. The distribution insures a maximum rate of echo buildup" because less simultaneous echoes will be heard (which would tend to reinforce those echoes as they repeat, causing a more uneven decay).

The values used for feedback gain "strongly interact with each other" and it is difficult to determine if a particular set of values will not create "sustained oscillation" rather than reverberation. Hal Chamberlin suggests typical delay values are around 0.8 seconds "with the long delay taps being somewhat more and the short taps being somewhat less."

Cascading simple reverb modules

Another problem encountered in the design of reverberation systems is that "at certain frequencies, peaks and valleys in individual amplitude responses may coincide to produce an exceptionally strong peak or valley." Carefully choosing delay times while also scrambling the delays for good echo density can be quite a complex endeavor, and could make use of a computer program to calculate possible values.

A better solution for good reverb

The multiple-echo module can be modified so that it has a flat amplitude response by using an all-pass filter model. The use of feed-forward and feedback (creating both poles and zeros) enables the peaks and notches in the frequency response to cancel each other out. An allpass filter acts as a delay that has a non-linear frequency response.

Requiring far fewer digital signal processing modules, cascades of as few as 3 allpass filters can simulate concert hall reverb, but more than 7 is better. Delays still need to be exponentially distributed with a prime number of samples.

Good approximation: first stage (longest delay) should be in the 50ms range, then multiply by a constant somewhat less than 1.0 (e.g., 78). Succeeding delays will be 39, 30.4, 23.7, etc.

The shortest delay should not be much less than 10ms if a "distant, hollow" sound is to be avoided. A resonance that is added at short delay times is a result of the periodicity pitch that is created as any signal is recirculated through a very short delay line (see discussion of Karplus-Strong algorithm).

Feedback factors and delays determine reverb time. "Reverb time" of a single stage is the number of delay circulations until the signal has been attenuated by 60db. If feedback values are kept constant and the delay times used get successively shorter, the feedback factors are determined by the longest delay time used.

$$R_t = 6.9 * \frac{D}{L_n}(F)$$

R_t = reverb time in seconds

F = feedback factor

D = delay in seconds

Reverb time of a cascade = longest section time.

[Chamberlin, 1985, p. 512]

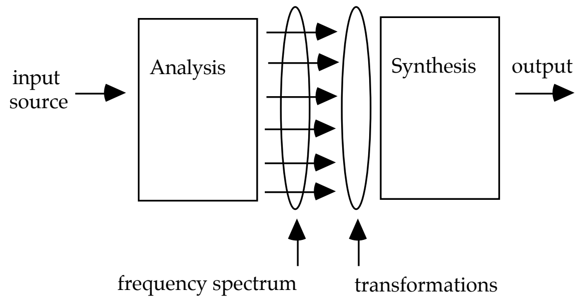
Consistent with the Haas effect, subtle differences between the right and left signal output of a stereo reverberator cause the output to sound as if it is coming "from all directions", while the source signal "retains its original directivity."

basic components: frequency domain effects

In addition to the basic effects algorithms discussed above, computer-based systems for real-time effects-processing will generally also be able to do a variety of other processes. These processes include real-time analysis/synthesis and transformations, real-time synthesis using a continuous signal from a performer to control various parameters, and direct-to-disk audio sampling and playback. Real-time synthesis with performer control is possible using some outboard synthesis modules, and real-time sampling and playback may also be possible. One class of signal processing that is generally not available using a commercial dedicated effects-processor is the computer intensive manipulations possible using analysis/synthesis methods. A few of these effects are described briefly below for background only, since the body of the paper deals with effects that do not make use of these processes.

analysis/synthesis methods

Analysis/Synthesis is a three stage audio processing technique. The three stages are analysis, transformation, and synthesis. A sound source is first input into an analysis stage, where spectral information is extracted. This information may then undergo some sort of transformation before being input to a synthesis section.



phase vocoding and transformations

typically modulated parameters/typical values

A phase vocoder is one example of an analysis/synthesis method. A mathematical operation (Discrete Fourier Transform or Short Time Fourier Transform) is performed on the incoming amplitude samples to extract the spectral information in a series of "time slices". Once the spectral information has been gathered transformations can be made to the audio in ways not possible in a merely time based system.

Some possible transformations include:

- Pitch-shifting without time stretching or compression
- Time stretching or compression without pitch-shifting
- Convolution - multiplication of one signal by another to mix their spectral characteristics.
- High Resolution Filtering
- Frequency dependent compression/expansion
- Pitch tracking

In a real-time system, spectral information gathered from a performance can be used as control signals for real-time synthesis algorithms.

dedicated effects-processors vs. software-based systems

With a good MIDI implementation (many parameters that are accessible, simply and simultaneously, by MIDI with low bandwidth) the advantages of outboard effects-processors are easily identified. The decision to work with a processor that is <u>not</u> computer based, is a choice between quick and easy access to processing with a fixed implementation (in the case of outboard equipment) and a more dynamic implementation, in which decisions about how to spend processing power of the DSP chip are, to a larger extent, left to the user. When programming decisions are left to the end-user, there is obviously a steeper learning curve.

Additionally, the ability to do frequency domain manipulations are available only in real-time computer based systems. There are outboard effects-processors which list "Phase Vocoding" as an available algorithm, but usually these are very limited by the small number of frequency bands used.

using the Eventide H3000

The Eventide company has manufactured a series of digital effects-processors that are powerful, easily programmable and relatively quiet. The newest model, the DSP4000 allows the user to graphically create new effects algorithms using a "box and line" interface and preprogrammed low-level effects and control modules. In addition, control over the transition from one effects "patch" to another has been added as a feature. The Eventide H3000 does not have these features; however it is still a very powerful effects-processor with an extensive MIDI implementation.

Computer Processing of Control Input Data

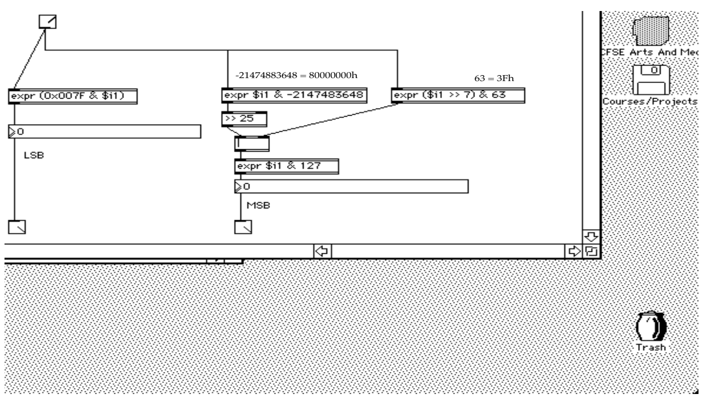
One way to take advantage of the timbral and textural possibilities attainable with real-time digital effects-processing, and keep a low overhead of computing power, is by implementation of parameter control using MIDI.

The inherent problem with attempting to do parameter modulation with MIDI data is the limited number of possible data values, (0 to 127), that can be used to choose and control parameters. Many of the parameters require more than 128 discrete values to be fully controlled, and there are often more than 128 parameters from which to choose. As a result a series of MIDI Control messages must be sent out to the effects-processor to obtain an acceptable level of control.

The MIDI implementation of the H3000 is based on the use of "unregistered" or "extended" parameter numbers. By using 2 controller messages to send a 14-bit number (split into a MSB and LSB), many more parameters can be addressed that the usual 128, and much greater resolution can be obtained for parameter control.

An example of this is the pitch shift model used in many of the base algorithms of the H3000. A total of 3 octaves of pitch shift is available (up to 2 octaves down or 1 octave up) by sending a 14-bit number indicating the exact pitch shift in cents (hundredths of a semitone). If the same range of pitch shift were to be available on a system not using
14-bit addressing, the pitch shift could not be as smooth as it is on the H3000, since a total range of 3 octaves or 3600 cents would need to be mapped to the 128 possible values available with a single message controller. Other effects-processors, including the Digitech TSR/24 and the Ensoniq DP/4, use various models for pitch shifting to facilitate a large pitch shift using only one continuous controller.

The Eventide H3000 uses non-registered parameter numbers -- the MSB of the parameter number is sent as a value of controller #99, and the LSB as a value of controller #98. The data value is then sent out as the MSB and LSB of controllers #6 and #38 respectively. In order to have one source of continuous control (0-127) over the two numbers, I created a patcher in MAX called BitSplitter:



Sound examples and discussion

A group of Max patches that I built to control the Eventide H3000 were used for two musical projects and an interactive movement and dance workshop, as described below. For Dinu Ghezzo's "Manifesto" and for "Chew/Buzz/SLAM", the controller setup was similar. It consisted of a DrumKat MIDI percussion controller, and the MIDI Remote Controller (hardware sliders) by Lexicon. The Max patches were written to allow control over pitch shift, delay times, "repeat" functions, and input and output of the unit, remotely.

For the "Active Space Workshop" with John Crawford and Lisa Naugle, (Tisch School of the Arts, NYU, in March 1995), dancers and others moved through a performance space with a video motion tracking system (Rokebe Very Nervous System) and piezo electric pickups as MIDI triggers and continuous controllers. The dancers' movements and the triggers controlled parameters on the effects-processor, which was processing audio and music being made by others present at the workshop. Ideas for future development of this combination of systems include using a wireless microphone on the dancer so he or she can both make and control the processing of vocal sounds.

Problems encountered with the H3000:

Even though the Eventide H3000 is a powerful and useful machine, I encountered several problems with the unit in the creation of pieces using live audio processing. Many of the delay algorithms available are not interpolating delays, and therefore discontinuities (clicks) are introduced into the signal whenever delay times are modulated while audio is in the signal path. These delay algorithms must be used with caution and only when delay time values will not be modulated.

In addition, certain parameter combinations will consistently produce tones that are unrelated to the input signal. These are a result of either roundoff error, oscillations produced by the delay line length, high feedback levels and the specifics of the internal processing power and sampling rate of the chip that the unit is built around. These sounds are difficult if not impossible to accurately reproduce on another piece of equipment since they are so specific to the H3000's architecture. In several of my pieces I have chosen to exploit these sounds, which had the unfortunate side effect of causing some small portions of my piece difficult to port to another system including the DSP4000 also made by Eventide.

Another problem with the H3000 is more serious. If too many MIDI messages are sent to the unit too quickly, occasionally the audio output completely shuts down. It is possible to monitor the MIDI data buffer and check for overflow problems, in this case it is not reported, and yet the output intermittently is completely stopped. In some cases audio output only resumes when the unit is powered down or a MIDI volume message is sent to it.

The engineers at Eventide were not familiar with this problem, and therefore were of little help. Richard Zvonar, who has written a Max editor/librarian and has done quite a bit of professional work with Eventide products concurred with me and mentioned that he uses the Max "speedlim" object to thin out the MIDI stream and processing requests being made to the unit, to prevent this problem.

ICE-9

A composition that I wrote for flute and tape, ICE-9 (1994), made use of the interactive MIDI parameter control of the Eventide to generate compositional materials. The primary sounds on the tape are pitch-shifted bells that were altered by mapping the velocity of a keyboard controller directly to the pitch-shift of the Eventide. Other sounds on the tape were samples that had been altered using SoundHack and were then also processed using various Eventide algorithms. By moving quickly to the extremes of the available range of a parameter and keeping the gain near unity, the output of the unit can be caused to oscillate, clip and feedback in (nearly) predictable (and interesting) ways. This output was recorded and used both as source material and as a gestural guideline, creating the overall shape of the second half of the piece.

Chew/Buzz/SLAM

Chew/Buzz/SLAM is a composition I wrote in 1994, for solo voice and live processing using an Eventide H3000 effects-processor. A Max patch controls the Eventide, and a DrumKat and Lexicon MRC (MIDI Remote Control sliders) are used by the performer (myself) to control of the Max patch.

Additional sound sources were from pre-recorded sounds played back during the performance. These sounds were played back to cover up the gaps in processing that are created when a new effects patch is called up on the unit. On the H4000 it is possible to crossfade or "segue" between these, but not on the H3000.

Some basic controls are available to the performer at all times during the piece, to facilitate handling any audio problems (feedback, discontinuities in a delay line) quickly. A bypass toggle can also be controlled remotely at all times by the performer. In addition, volume (controller-7) messages can be sent at anytime to change output level of the entire unit, and individual control of left and right input/output levels has been implemented. The feedback level of any given patch can be lowered significantly or faded down at anytime to deal with sudden changes in input level that might cause the unit to unintentionally feedback or oscillate.

Compositionally, the piece was created in a very intuitive way. After building a Max interface for the H3000, I experimented with various kinds of vocal sounds and algorithms for controlling the processing. The sounds in which I was most interested required that quick and drastic changes be made to several parameters nearly simultaneously. The piece has an improvisational element to it, and it is scored using a graphical score that describes the vocal line, and the Eventide's physical operation as well as the desired behavior from the unit.

At all times, and in every patch, certain Max "presets" were created as points of departure for the improvisation. The presets are predetermined values for all the parameters in a given algorithm that can be sent with one action (usually mapped to the DrumKat, and often set up so it is possible to cycle through them).

The score of Chew/Buzz/SLAM

In the very first section, a solo vocal line ascends in a pentatonic, staircase-like motion, at times moving microtonally. The vocal line is legato and sustained, and is being fed into the Eventide, on a patch called "Ping-Pong", but not being output. "Ping-Pong", as might be expected, is a slapback echo with a fixed (and incessant) delay. There are built-in control parameters for pitch ("detune"), feedback, and high frequency damping ("hi-cut"). After about 20 seconds, the output of the Eventide is faded in using the "Mix" button (going from completely dry to wet).

Because the vocal line is sustained, the output of the delay is not heard as distinct, but rather as an ethereal chord generated from the sung pentatonic scale. The vocal line then becomes shorter, consisting of vocalized sounds with glottal stops for percussive effect and some growling. All of these sounds are heard bouncing back and forth in the "Ping-Pong" slap back echo. The first algorithm is then started, which very slightly and randomly detunes the output of the delay. The range of values of randomness (using the "drunk" object) becomes controllable by the performer, and the sounds sung/spoken become more percussive.

A second algorithm is available at this point to do the same sort of randomization on the "high-cut" high frequency damping parameter. A big difference in the impact of the two algorithms is that once a sound has been low-pass filtered (high-frequency damping), its high frequencies cannot be added back. It is a destructive process: anything currently in the delay line is affected. If the "hi-cut" is subsequently lessened (allowing for more high frequencies to pass) only new audio events being input into the delay line will be affected.

At this point the tape part begins to play. The purpose of using a pre-taped portion is to cover the silence during patch changes on the Eventide. In future performances of the piece or whenever possible, the audio portions will (hopefully) be played back directly from hard disk or CD, so as to give the performer more direct control over the timing of these audio segments.

To make the audio portion combine well with the live material it only contains sounds created using the Eventide. By pre-recording material, it was possible to do some additional processing to the sounds by adding reverb or additional pitch shifting. For example, a "Gregorian chant" was pitch shifted down 3 or 4 octaves, which is not possible with a single Eventide H3000. Additional percussive vocal sounds appear in the tape being processed with the same DSP algorithm as the live vocal, adding complexity, fullness, and in some cases, spatialization, to the overall sound.

In the second section of the piece, as the now percolating sounds on the tape fade out, the patch on the effects-processor has been changed to "Long Digiplex". This patch consists of a single 1.4 second interpolating delay line, that outputs in mono. Control parameters include delay length, feedback (from -100 to 99%), mix level, and a repeat toggle (it samples up to 1.4 seconds of audio and repeats it continuously until it is toggled off). Additionally, as "expert" parameters, control is given over "glide speed" and "glide enable". If glide is enabled, the "glide speed" parameter determines how long it takes to interpolate between delay time changes. A change from a delay of 10ms to 1000ms can be set to happen quickly or very slowly and smoothly.

As was mentioned above, a by-product of changing a delay time on an interpolating delay line is a Doppler like pitch shift. This Doppler-shift is one of the main musical devices in this section of the piece. Another is the use of very short delay times to create a Karplus-Strong type plucked string sound.

The repeat toggle is used in this section to sample gibberish sounds and to replay them as an audio loop as new material is sung. In the original score, this was to be done "silently", that is while the output volume was down so that the repeated gibberish and tones could be faded in a few moments later. This proved to be a problem since discontinuities could be introduced into the sound and that would be painfully obvious once the sound has been brought up. If the input gain could simultaneously be scaled up and down as audio was being fed in (as in Leonello Tarabella's piece above) this problem perhaps could be circumvented.

Very short delay times (1-2 ms) with high feedback create the plucked string sound mentioned above if short percussive and non-vocal sounds are made by the singer. The plucked string sound is musically amplified on the tape at some point, where many copies of it are played back slightly detuned, creating a cloud of plucked strings. This would only be possible live if 2 effects-processors were being used. Two millisecond delay times have a very hollow sound, which works better with slightly vocalized sound. This is more of a comb filter effect.

Several parameter value configurations stored as "presets" are used in this section to go quickly between long delay times with low feedback levels and this plucked string effect. This change in delay times creates a pitch shift on anything left in the delay line. By varying the glide speed and the sorts of percussive vocal sounds (chewing, kissing, clicking) and loud vocalizations input to the system, many different kinds of effects. Sustained sounds sung by the vocalist created a kind of crashing wave and it is possible to cause the unit to oscillate (its amplitude output) by sending in a loud signal while doing this.

An algorithm is used to very quickly change the delay times with a "metro" object. If a sustained sound is input to this algorithm, the output has the quality of an analog oscillator-based synthesizer. Percussive sounds input into this algorithm cause a more textured sort of output.

The third section does not use algorithms as parameter control sources, and only uses the presets as starting points for improvisation. The effects algorithm used in this section is called "Reverse-Shift", because the delays are all played backwards, giving much the same sound as playing a tape recording backwards, and pitch-shifted. As with all time-domain based pitch-shifting, playback speed of each pitch-shifted audio event is correspondingly sped up or slowed down.

The left and right inputs are summed and then sent into two delay lines with independent control over pitch shift, length of delay, feedback and mix. The output of each delay line is then sent to its respective output. The feedback loops of the two delays are summed before being input into each delay line again.

Some of the combinations of parameter values create unchanging tones all by themselves, regardless of the audio input, especially at short delay settings. This is perhaps limit-cycle oscillation.

By pitch-shifting something with the feedback setting above zero, the sound will cycle through being continually shifted up or down. Since the feedback loops are summed together before being split up again into the delay lines, it is not easy to predict what certain sound and parameter combinations will do.

musical and technical specifics of Chew/Buzz/SLAM:

Generalities

In experimenting with the H3000 I made note of some behaviors under certain conditions that are a combination of the parameters and algorithms used as well as the character of sound being input (in terms of frequency spectrum as well as actual length in time). I would like to make some generalizations about certain parameters and algorithms and their particular musical properties.

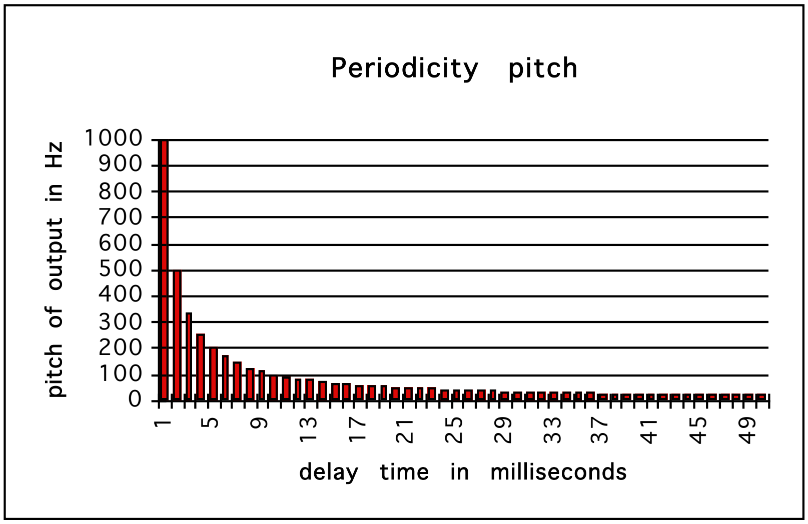
Various delay times and their effect

Especially interesting was what happens at various delay times when the feedback on a delay line is kept high (with and without any pitch shift).

Short delay times create textural effects, if the delay is recirculated immediately after being sampled. If the delay times are very short, they create comb-filters as was described above. Delay times between 1ms and 30ms - 50ms create resonances that are heard very clearly. If an impulse is sent into the delay under these conditions a plucked string tone is heard. If noise is sent into the delay and continually repeated (such as with a "sustain" or "repeat" function), a steady tone is heard.

The generated frequencies have a definite pitch, which will descend through the overtone series as the delay is changed from 1ms to , 2ms, 3ms, etc.

An impulse or short audio signal that is delayed 1ms with high feedback gain, is shifted and replicated 1000 times per second creating a 1kHz resonance (periodicity pitch) regardless of the complexity of the tone's spectrum. The frequency of the created resonance can be determined if the length of the period is known. The period of the repeated delay is equal to the delay length (in this case it is the same as the delay time), and therefore the frequency is 1/period. At 2ms this is 1/.2s = 500Hz. Below is a chart of the frequencies registered up to 10ms delays.



At delay times of 50-100ms all pitch shifts are heard with only a slight echo added. The behavior is not as frequency dependent: there is little difference between what happens to high and low frequency sounds.

At 150ms the sounds that have been pitch shifted up as they recirculate through the delay start to sound watery and ascending. Sounds that have been shifted down on the other hand still sound fairly normal (except for the pitch shift).

At 200ms (with no pitch shift at all), complete short snippets of sound can be circulated through the delay altering the textural effect and a ringing is heard, (especially if high pitch shifting is added). The ringing does not change its pitch according to the frequency of the input nor or the pitch shift employed. At 250ms similar behavior was noted, only with lower ringing. It is interesting that at 260ms the ringing is completely gone, but at 300ms delay time, it is back and longer. This is perhaps because at 200ms the pitch generated as the periodicity pitch is 5Hz and at 250Hz it is 4Hz, both too low in frequency to be heard as pitches but still interacting with the system in some manner thereby creating their own oscillations.

Also interesting is that these resonating frequencies can themselves be pitch shifted creating an effect not unlike analog synthesis. They are still heard as resonances, not as distinct delays, because the pitch shift is independent of the delay modules.

Differences between high and low frequency shifts

The output of high and low frequency pitch shifts changes differently (and non-linearly) over most of the range of available delay times. When using a delay algorithm such as the Long Digiplex in which the outputs of the left and right delay lines are summed as part of being recirculated, this can create some not easily predictable, yet very interesting musical results.

The reason for the difference between recirculated high and low pitch shifted sound is twofold. As a pitch is pitch shifted up or down, (since we are discussing time-based effects-processing), the length of the sample is shortened or lengthened accordingly. Consequently, any sound input that is pitch shifted up will become increasingly textural as it grows shorter and higher and higher in pitch. Any sound that is being pitch shifted down will be lengthened. Even if the sound was originally shorter than the full length of the delay line, it will grow longer and longer until the delay line can no longer hold the complete sound at which point it will begin to truncate (presumably from the end).

Any silence at the beginning of the delayed sound will also be lengthened and consequently the sound will become even more delayed at its beginning. As a result, even a feedback of unity can be stable with a downward pitch shift of 1200 cents (one octave) or more, and a delay of 1400ms. At a pitch shift down of two octaves, it takes a noticeable amount of time for the delay to recirculate, and it seems that the sustain parameter in Long Digiplex was a good idea for creating any sort of textural effects, since what ever is in the delay line dies out after only a few repetitions.

the technical explanation :

To shift a sound up in pitch (in a time based system) one of two things can be done -- either the sample rate can be increased to accommodate "reading" through the sound stored in the delay line at a faster rate, (which is not very practical as most systems do not have variable sample rates), or the file can be read more quickly by skipping every other sample (in the case of a doubling of pitch).

For a pitch shift of a non-integral amount some interpolation would still be needed, but samples nevertheless will probably be thrown away. When a sound is pitch shifted down, either the sample rate can be lowered to read through the delay line more slowly (impractical as was just mentioned) or interpolation or repetition can be made between each sample that is read to effectively lengthen the amount of time needed to read through a single cycle of a waveform. When a delay time is suddenly changed in an interpolating delay line, the glide and interpolation to the new delay time create a de facto pitch shift up or down as a byproduct according to the Doppler effect (see discussion above).

What should also be mentioned is that the sounds being lowered in pitch (the delay time being lengthened) are being interpolated more and more between actual samples read, increasing the amount of roundoff and other related errors that can accumulate and create various kinds of noise and other harmonic and non-harmonic sounds. The sounds being raised in pitch (delay time being shortened) are in effect having their sample rate lowered as samples must be thrown away and interpolated (for non-integral delay times), increasing the possibility of aliasing and clipping. Interestingly enough, the high frequency pitches tend to already be underrepresented in terms of sample rate compared to lower frequency sounds which always have many more samples taken per period than do sounds at frequencies closer to the critical sampling (Nyquist) rate.

section 3: interaction of delay and pitch shift

The actual parameters that were used for the 3rd section of Chew/Buzz/SLAM are outlined below. Rather than create a series of automated algorithms to modulate the parameters with user control of the overall effect as was done in the first section and part of the second, 8 preset parameter settings were saved in the Max patch that controlled the piece. During the performance, the performer cycles (via the DrumKat) through the presets in various orders and with various kinds of sounds as input. The amount of pitch shift in the left and right channels and the combining of the two in the feedback loop creates complex chords and arpeggiated structures that ascend or descend.

Preset #1:	pitch shift	delay
L channel:	457 cents	1156ms
R channel:	1 cent	400ms

The feedback during for all the presets is set at 80-90%.

Musical result:

Since the outputs of both channels are summed before being fedback into the two delay lines with their respective pitch shifts and delays, the result is a combination of out of tune ascending 4ths created and based on any input sustained pitch or sound. This is due to the pitch shift up of 457 cents (4.5 semitones), on the left channel, which is recirculated and subsequently pitch shifted again. Shorter sounds ascend in frequency in shorter and shorter snippets.

Preset #2	pitch shift	delay
L channel:	185 cents	774ms
R channel:	0 cents	912ms

The change from Preset #1 is abrupt and as a result, a slight Doppler shift is heard in both channels. The pitch shift is higher on the left channel and the pitch shifted up further because of the sudden change to a shorter delay. On the right channel the delay is suddenly longer creating a Doppler shift down.

Musical result:

The pitch shift of 185 cents creates whole tone arpeggios whenever a sustained tone is input. As the arpeggios sustained they created a complex chord, based on the octatonic scale, which changes slowly with the input.

Anything leftover in the delay line from preset #1 will be subject to additional Doppler effect pitch shifts. The shifts initially go down on the right channel as the delay is lengthened and up on the left, but begin to fan out into as they are summed and shifted again. Anything that was shifted up or down will be shifted both up and down on the second and subsequent passes through the delays. Sounds that were heading higher in frequency under preset #1 will level off or even start to descend as a result of all of these pitch shifting effects when moving to preset #2 and because the pitch shift of 185 cents is lower than that of the 457 cents in preset #1

Preset #3	pitch shift	delay
L channel:	185 cents	977ms
R channel:	1 cents	400ms

This preset is very similar to the first one, with a pitch shift of nearly a major second, and a small pitch shift in the right channel. Anything left in the delay line is subject to a Doppler shift, which is going up in the right and down slightly in the left channel.

Musical result:

An ascending scale/chord as described above. It is close the whole tone scale in the second preset: only the delay times are different, creating more subtle Doppler effects. When this difference is compounded by the recirculation and subsequent pitch shifts, the subtle differences between this preset and the last one add up to a different sound color. At this point in the piece the performer adds fricatives and more percussive sounds the sound palette. An exploitation of the Doppler shift on anything left in the delay line is part of the musical applications of the effects algorithms.

Preset #4	pitch shift	delay
L channel:	-343 cents	733ms
R channel:	0 cents	511ms

Musical result:

Any sounds left in the delay line as well as all new signals that are input are pitch shifted down in arpeggios that sound like a minor 7th chord: intervals somewhere between 3 and 4 semitones. Since this is the only pitch shift, and the first one that is in a downward direction, (the arpeggios slow down) attention is drawn as well to the rhythmic changes made by the change in the delay times as well. This is especially noticeable on the material that had been previously circulating through the delays using the parameters of preset #3 and was present in the delay line when the parameter change was made. Also interesting are those gestural characteristics, such as vibrato, that become exaggerated by the slowing down due to the pitch shift down.

Preset #5	pitch shift	delay
L channel:	-2400 cents	1400ms
R channel:	-2014 cents	1377ms

There is a rather sudden shift in both the delay times (much longer) and the pitch shift. The change in pitch shift on the left channel is from 3 1/2 semitones of shift down to a full two octaves down. The right channel, which previously had no pitch shift at all, is suddenly shifted down an octave and a minor 6th.

Musical result:
Short vocalizations have a very animal-like quality as they are downshifted on each repetition by two octaves. Each sound is very distinct and repeats only a few times until it is pitch shifted out of range. Longer, sustained notes create an arpeggiated chord, as in the previous sections, and divide the octave in thirds creating an augmented chord.

In addition, there is a residue tone present regardless of the quality of the input (even when no input is present) that can only be attributed to limit cycle noise or roundoff error (see above). This residue tone (a high frequency sine wave) is pitch-shifted as well and circulated through the delays with all the other.

Preset #6	pitch shift	delay
L channel:	1200 cents	62ms
R channel:	1028 cents	1ms

Both channels have suddenly been pitch shifted up one octave, with very short delay times being used.

Musical result:
In spite of the big changes in pitch shift and high feedback, the high and low pitch shifts are heard as resonances and comb-filtering, since the delay times are relatively short. Moving back to preset #5 from these parameter settings, the comb-filtered sounds enter the delay line and are pitch shifted down in descending minor 6ths and major 3rds. This is due to the pitch shift changes and to the Doppler effect.

Preset #7	pitch shift	delay
L channel:	1200 cents	110ms
R channel:	1028 cents	266ms

The delay times have been lengthened slightly but barely enough to hear a Doppler effect.

Musical result:

The constant pitch shifting upward combined with short delay times creates a textural effect that is high in frequency and can be described as watery or shimmering. Every nuance and gesture of the input sound is amplified, as the frequency envelope of the overall sound follows the musical gestures. Moving between this preset and #5, which has opposite qualities of low pitch shift and long delays, can create some very nearly uncontrollable delay glides up or down. Moving from preset #5 to #7 takes the highly textured sound and pitch shifts it down over and over.

Preset #8	pitch shift	delay
L channel:	-1715 cents	1400ms
R channel:	-2014 cents	1377ms

Musical result:

All the pitch shifts are brought low again. Whatever is left in the delay line from previous presets is affected by the new settings. For example, if the comb-filtering effects in preset #6 are used before #8, the comb-filtered sounds are pitch-shifted down sounding much raspier than any new sounds entering the system. It is also interesting to put one of the inputs on a 1ms delay and allow the comb-filtered signal to continue to be a part of the delayed signal that is being recirculated.

Overall the changes in pitch shift and delay in this section of Chew/Buzz/SLAM can be seen in the chart below. For diagrams of the chords that are created in several of the presets, please see the Appendix.

other work for real-time effects-processing

Below is a discussion of the work of several composers and performers who have used effects-processing in various ways as compositional material or in their improvisation. The musicians represented have various degrees of technical knowledge about the actual effects-processing they have used, but the purpose of the discussion was to have them clarify their musical approaches to the use of effects-processing. The questions they were asked concerned: what type of signal processing algorithms were used in their pieces; what kinds of control sources were used, what sort of internal organization (e.g., the design of their computer programs). In terms of their compositional intentions I asked them to delineate their ideas on the role of the effects-processor as an "interactor": whether it was used as a separate "player" in their compositions or as purely an extension of the human player. I was also interested in the types of sounds they most often used and any aesthetic judgments they could offer about what they considered interesting.

Mari Kimura: Ensoniq DP/4

In a recent conversation, violinist-composer Mari Kimura described the technical problems she has encountered when porting her pieces for violin and interactive signal processing, as well as the compositional decisions she has made concerning the types of processing she has used and the role she has assigned to the output of these processes in her pieces.

Ms. Kimura seldom uses delay or other effects to create rhythmic patterns or a sense of the altered signal as being a separate "player" in her compositions. Rather she treats the processed sound as an extension of her violin, compositionally integrating it with her own playing (as an instrument paradigm). [Rowe, 1991] The musical behaviors are generally transformative, with improvisation being an important part of her pieces. She often creates a sound "environment" before composing the music and determining the improvisations that will be played using that environment.

Some of the transformation that Ms. Kimura has used in her pieces are interesting to look at more closely in the context of the discussion of portability. She uses a DP/4 effects-processor made by Ensoniq. This effects-processor has some very powerful algorithms and programs available, a flexible architecture and a very good MIDI implementation. There are 4 discrete inputs and outputs which can be configured in a variety of serial or parallel routings of processing units in between.

Ms. Kimura does not use the changing delay times to create Doppler like pitch shifts, although she does use pitch-shift and harmonization quite a bit to create chords and clusters. She makes use of phasing in her acoustic pieces, by creating beat frequencies between two pitches in an octave or unison double stop (sometimes using an open string for one of the notes). She has tried to make use of the beat frequencies as a musical idea in her electronic pieces by using subtle pitch shifts on her acoustic sound.

The range of pitch shift that is available on the DP/4 is one octave up or down, but by chaining together 4 consecutive pitch-shift units, a total of 4 octaves of pitch shift can be obtained (this, of course, leaving no other processing available).

When the length of a reverb is changed in some way it is to create a sense of a change of sonic environment rather than to create an abstract or synthetic sound of some sort. This is not unlike the type of processing described by Todd Winkler below. Available delay time is a maximum of 1500 ms (without any memory upgrades), but as with the case of the pitch-shift, longer delay times can be created by chaining effects units together.

The parameter values on the DP/4 are programmed in two different ways. For example, the parameter for fine pitch-shift is coded internally as a number between 0 and 127 (for use with the Sysex implementation), but what is displayed on the front panel is a number between 0 and 99. This disparity, although convenient for some, takes the control of mapping degrees of a pitch shift to MIDI controller values out of the hands of the user and into the hands of the engineers and programmers at Ensoniq.

Jean-Claude Risset

Jean-Claude Risset in his piece "Variant for violin and treatment numeric", written in 1994 for violinist Mari Kimura, was able to specify the types of signal processing that were to be applied to the violin in very simple descriptions making the piece possible to be played on a variety of systems. In several performances the choices and difficulties encountered with porting the piece to a platform other that the DP/4 are outlined below.

Risset calls for 6 different effects in his score, providing detailed descriptions of the physical and acoustical properties of the parameters used rather than their implementation on a particular piece of equipment. (see Appendix)

He calls for transpositions or pitch-shifts with attenuation of the output of 10dB, to create subtle chords with the violin. This is fairly easy to do on most effects-processors, except some older models that specify only diatonic harmonizations.

Three types of reverb are specified by their length only (Normal, 2-3 secs.; Very Long, 5 secs.; Very, very long, Infinite). Unfortunately, this leaves decisions about many other aspects of the quality of the reverb used in the hands of the performer. Since the acoustics of the hall in which the piece is to be performed will have an impact on the overall sound of any reverb, this perhaps is not too bad.

Delay and echo are used in this piece for their ability to create complex rhythmic patterns. A single delay of 1.5 seconds with 6dB attenuation is used, as is "delay repeté", which is also 1.5 seconds, but with feedback that has been attenuated 10dB on each repetition.

One final effect is used that is not always available on the effects-processors used by the performer. Ring modulation is specified (415 anreau parameter), but if it is not available, the composer has requested that a pitch shift of -1 and -2 semitones or later -3 and -2 be used in its place.

Ms. Kimura successfully ported the piece to a ZOOM effects-processor on one occasion (after passing up an older Lexicon and a Yamaha processor because of insufficient MIDI implementations or lack of chromatic pitch shifting.) On another occasion the piece was ported to the Eventide H4000 for a show at the Kitchen in New York.

Ron Smith: DP/4

Canadian composer Ronald Bruce Smith has had the experience of writing a piece for flute and a powerful real-time effects-processing computer program and then later porting the piece to a dedicated effects-processor (the Ensoniq DP/4), and other platforms.

The piece, "Esquisse" was originally composed for a real-time synthesis program called "Waveguides" developed at CNMAT at UC Berkeley by Brian Link. A waveguide synthesis algorithm is generally a recirculating delay line that models an oscillating physical system, like the Karplus-Strong algorithm discussed in the first section of this paper.

Smith says that he generally uses acoustic effects-processing in his compositions as an extension of the live instrument being played, rather than as a separate musical entity. He does not make use of "noise" or other electronic type sounds that can be created using signal processing algorithms. He instead does this through the frequent use of digital delays to set up complex rhythmic patterns or textures: a frequently used device that he uses well. These patterns, which tend to be "elastic" rather than "repetitive" or "pulse oriented", create "heterophonies" around a melody through gradual harmonic changes that are based upon harmonic and inharmonic spectra, often the spectra of a bell.

Smith commented that, as it is for many composers working with both acoustic and electronic mediums, his writing for acoustic instruments "is continuously being informed and refined" by his work with computers and electronics.

Ideas from digital delay processing have therefore influence his purely acoustic music, through the use of heterophony around a melody that is modeled what can be output from a digital delay.

In the case of Esquisse the effects-processor was used to create a heterophony by using delay, and some harmonization in the "more active parts of the work", and to "amplify the harmonic structure of the work" by additional harmonization.

In notating the effects-processing in the score of the piece, Smith was very precise in terms of pitch (using microtonal notation), time delays (in milliseconds), tap delays (using Western rhythmic notation) and the relative amplitudes of the effects in the processor(s). This is similar to the way in which Jean-Claude Risset notated his piece for Mari Kimura, but the complexity of the effects, and therefore the notation necessary, is greater in Esquisse.

The piece was later ported to the DP/4 effects-processor (Ensoniq -- please see discussion below) controlled by a Max patch, and more recently to the Silicon Graphics platform, by Miller Puckette using DSP Max. This was done by reconstructing the effects based on information in the score alone, with little assistance from the composer necessary.

Todd Winkler

Todd Winkler has described his compositional uses of effects-processing as being wide-ranging (in his paper session at the International Computer Music Convention in 1991). Although I am unaware of which dedicated effects-processor he is currently using, his commentary on the subject is relevant to this discussion.

Winkler has used delay and pitch shift to create rhythmic canons and counterpoint lines accompanying and being triggered by a soloist. He has also sequenced continuous controller information to be triggered by certain notes in a score, for timbral manipulations of the performer's acoustic sound.

He says it is important that the musical gestures be both musical and meaningful in the context of the signal processing being done. One example he gives is that of the velocity of notes in a 6 second violin phrase being used to create a breakpoint envelope that is later used as a continuous control for reverberation time. [Winkler, 1991]

Eric Singer: Digitech TSR/24

Eric Singer is a New York City based computer programmer and saxophonist. With his background as an improviser and computer control of musical instruments through MIDI, he has a different approach to the effects-processing than some of the conservatory trained musicians with less technical backgrounds that are interviewed or discussed in this paper . His approach to his pieces, like that of Mari Kimura's is based on finding an interesting performing "environment" in which to improvise. Unlike Ms. Kimura's pieces, they are not notated with traditional music notation. When asked to describe a typical piece, he will provide a processing diagram (included below) indicating signal flow and indicated what type of musical sounds might be played with certain settings.

Singer uses the TSR-24 from Digitech with added memory as his effects-processor, using Max to control the unit via continuous controllers and a pitch tracker on his saxophone, a MIDI wind-controller, and a series of footswitches to send MIDI information.

He indicated that he is predominantly interested in working with the effects-processing in an instrument paradigm, using the effects-processing to amplify and expand the sound of his saxophone.

Singer uses the multiple effects setups available on his effects-processor, frequently feeding the output of a multitap delay into other modules, such as flange or a sample and hold module (like the repeat function on the Eventide H3000, but able to hold up to 5 seconds of audio). He frequently will use reverbs with long decay times to create a textural effects made up of his own sound and musical gestures against which he then improvises. By playing longer tones into this sort of reverb setup he can create a harmonic pad of saxophone sounds. In most cases he creates sounds that are closely linked with his own acoustic sound.

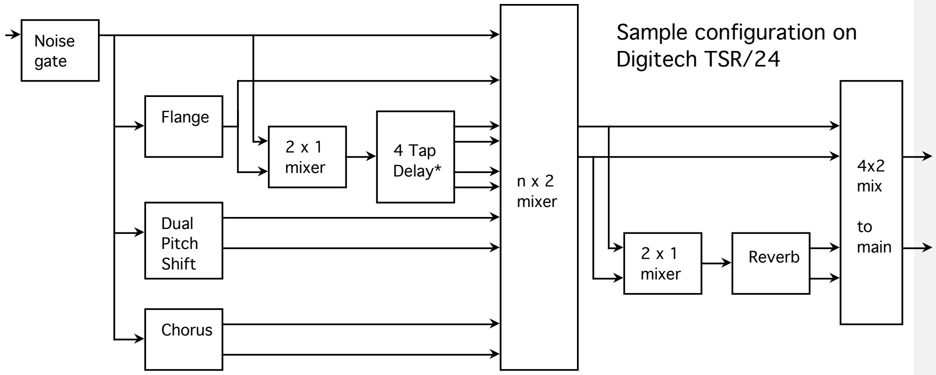
When asked about his use of noise or more electronic sounds in his composing, he indicated that he will sometimes use key clicks and pops input into long reverbs and resonators ("tight fed back delays") or flange with high feedback to hear the "sound of the box" rather than his saxophone.

The pitch tracking is sometimes used to time phrases as he plays them so that he can ramp the feedback of the flange up or down according to the phrase length or a repeated pattern. In this way he can also dynamically control attack hold and delay rates according to his actual playing.

Frequency oscillation due to high feedback levels give him pitches that are independent of the material being input to the unit (as on the Eventide). He noticed that if he uses a flange and pitch shift when the unit starts to oscillate, what is fedback is also pitch shifted further magnifying the effect.

When using chorus effect, he will sometimes use a square wave LFO set to a medium to high rate with deep chorus to create arpeggios or rippling chords that "shimmer". Multitaps with different pitch shifts are sometimes used to set up preprogrammed "licks".

There are no interpolating delays on the TSR-24 so he does not use any Doppler like effects, since delay changes introduce discontinuities. There is a "Whammy" effect, however, on some of the newer Digitech units which is a programmable interpolating pitch shift (there is a glide time between changes in pitch shift).



Sample configuration on Digitech TSR/24

Patches constructed on TSR-24 can use any number of effects-processing modules, limited only be available memory. The documentation indicates clearly how much CPU and RAM memory is used by each module, enabling the user to know precisely how the memory resources are being used.

Richard Zvonar

Richard Zvonar -- a sound designer, audio systems designer, and composer (among other things) -- has collaborated with various performers over the years using a sophisticated approach to real-time effects-processing. He uses both the Eventide H3000 and DSP4000, has written a custom editor/librarian for the H3000 in the Max programming environment, and has designed several of the presets for the new DSP4000. In his ongoing musical collaboration with bassist Robert Black has involved the use of the H3000 in live and often improvisitory settings (see Black's CD "State of the Bass" on O.O. Records). Interestingly enough, Zvonar does not use any computer software to control the effects algorithms, but controls them directly himself during the performance (using his hands on the front panel!). At one time he used his custom Max patches for control, but has since given up the idea because the computer control added unnecessary complexity to the collaboration. Black is not interested in controlling the effects, only in improvising with whatever is programmed for him on the unit. The effects-processor certainly plays the role of a separate instrument in this case: one that is played by Zvonar in an intense duel/duet with Black's bass playing.

The Interactive Performance Group is a student group that I founded at NYU that is devoted to fostering collaboration between artists and technical persons for the creation of interactive art, music and performance pieces. This has been accomplished by putting on concerts, inviting guest speakers in the field, and most importantly in the Digital Salons -- evenings in which performers and technical persons can work together on new ideas in an informal setting.

The most recent Digital Salon was May 25, 1996, at Harvestworks/Studio PASS. After several structured improvisations and rehearsed pieces were played, musicians in several different configurations used my Max software to control the Eventide H3000.

That evening, as well in some other recent experiences at Harvestworks, I explained the nature of some of the effects algorithms and parameter changes I was making to them to many different musicians. I benefited from their responses both musically and verbally to the sounds they were creating. The instrumentalists included violin, voice, flutes, flugelhorn, contra-bass clarinet, cello and various percussion instruments including waterphone and Brazilian indigenous instruments. Overall, the generalities I made about the response of the effects algorithms that I made earlier in this paper held true regardless of the instrument.

The responses expected can be based on groupings of sound quality rather than instrument: the envelope of the sound (how percussive is it?), the harmonic content of the sound (how high are the harmonics? how complex is the sound harmonically?).

The Contra-bass clarinetist had an obviously different experience than the flute player when his sound was pitch shifted and recirculated through a delay. When pitch shifted down an octave the contra-bass clarinet may immediately be too low to be heard or handled by the DACs. If the sound is shifted up it may die away before it traverses the full frequency range (unless the feedback level is set very high). Likewise, if a flute is pitch shifted up, it will very quickly be stripped of its harmonics by the low pass filter on the DAC, but can be shifted down many times before it is too low.

The percussive quality of the sound will affect the way the filters respond, determining whether a periodicity pitch will be heard on a short delay length. On band delay algorithms the output of the Eventide is more interesting when the sounds do not move quickly, but are not very percussive and are rich in harmonics. If multiple delay taps are being used to set up rhythmic patterns, the degree of percussiveness of the input sound determines the texture of the output.

An interesting comment made that evening was from the flugelhorn player (who doubled on clarinet) who indicated that he felt less control intimacy when the sound was pitch shifted down. I surmised that this is true because although timbrally the sounds that were pitch shifted up and recirculated were all the same (close to sine waves) and moved up very quickly in a "wash" -- the rhythmic proximity of the sound was much more important in establishing a sense of control for the performer. As his musical expressions and phrases were shifted down and recirculated, they became slower and slower with longer silences between each onset. The sounds that had been shifted up maintained the phrasing and expression as they were shifted up, therefore leaving the player with a greater sense of the output sound's relationship to his own.

Cort Lippe: ISPW

Composer Cort Lippe has worked extensively in the medium of real-time acoustic processing. While working at IRCAM as a technical assistant during the 1980's, he was instrumental to the realization of many pieces written by visiting composers, first on the 4x machine and later on the IRCAM Signal Processing Workstation (ISPW).

The ISPW uses a dual Intel i860 card on a NeXT cube. Signal processing modules were written at IRCAM by Miller Puckette and others to control the signal processing on the ISPW using the Max programming environment. Processing and control code are all in the same Max patch, and the machine is fast and powerful enough to handle even a Fast Fourier Transform in real time. These signal processing modules, used for the 4x machine and later on the ISPW include: harmonizers, delay lines, frequency shifters and samplers, synthesis algorithms, filtering, reverberation, spatialisation, and FFT's.

In his own compositions using the ISPW and Signal processing capabilities of Max, Lippe has sought to fully exploit the possibilities of the system, making it nearly impossible (in his own opinion) to port the pieces to anything but a more powerful platform. He uses many time-based effects as well as sophisticated analysis/synthesis techniques to process live sound from the performer, and to use pitch and spectral information gathered from the live performance as control gestures for the processing done to that audio or to audio that has been prerecorded to hard disk. Time-based analysis is also used for gathering of gestural information, using envelope followers to track articulations such as flutter-tongue, staccato, legato, sforzando, and crescendo. [Settel, Lippe, Puckette and Lindeman, 1992]

In the past Lippe has written pieces in which the computer sometimes acts as a performer: interacting with the live performer in its own voice consisting of sonic materials drawn solely from the human performer's own sound. At other times, the computer acts as an extension of the instrument -- both in the live processing of the sound of the instrument and in the performer's precise control over synthesis algorithms whose output augments and extends their instruments sound.

Because the performers' scores are traditionally notated Lippe's Max programs for the ISPW should be considered part of the notation of the pieces as well. It would be impossible to recreate the piece using the paper score alone considering much of the information about the electronic portions of his pieces is written into the computer programs themselves. Information pertinent to the performer (cues, general behavior of the computer portion) are notated for the performer to ease the rehearsal process, but information about what is actually going on musically in the electronic portions of the piece remains in the graphical layout of the Max patch and perhaps only in the mind of the composer.

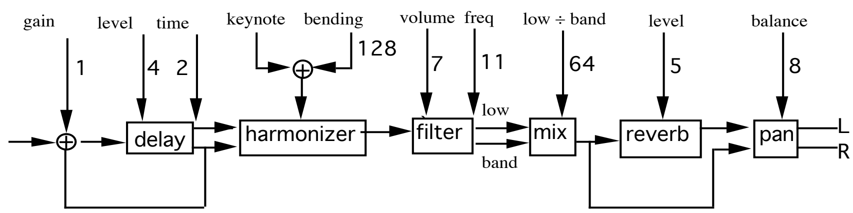
The ability to gather the quantity of information about a performance that is made possible by the ISPW, as well as the complex synthesis and audio processing algorithms that can be programmed, put the ISPW and therefore its compositional uses in a separate category than the dedicated effects-processors that are used by most of the composers represented in this paper.

Leonello Tarabella: MARS Workstation

In Italy, many of the computer music facilities are based in Computer Science Departments rather than in the Conservatories or Music Schools. This is largely due to tradition. Perhaps as a result, or as a related phenomenon is the large number of real-time DSP "boxes" created at the various outposts, each with locally produced hardware and software, and very few commercially available.

In a recent visit to New York University Dr. Leonello Tarabella of CNUCE in Pisa performed one of his pieces for the MARS workstation using 3 live musicians (trumpet, clarinet and voice) and controlling the output of the MARS with a homemade infrared controller. In an interview with Dr. Tarabella he both described the DSP algorithm used in the piece as well as the DSP manipulations of the musical signals and their musical implications during each of the piece's four 1-2 minute sections.

In the first section, the input is fed through the filter and a harmonizer with the infrared controller determining the Q of the filter employed. The musicians were instructed to play short fragments of sound using their mouthpieces (or fricatives and percussive sounds for the



numbers indicate the MIDI control change used

vocalist).

The second section, which the composer described as sounding like "bubbles", consisted of very short delays that created an effect not unlike that of the Karplus-Strong algorithm. The noisier the input during this section, the more affected output from the algorithm. The infrared controller was used for gain and delay time. The musicians were instructed to continue with the short bursts of sound, now using their full instruments and elongating the phrases.

In part three (after a short cutoff), the primary musical idea was that of creating polyrhythms with 2 unequal delay lines (with high feedback gain). The delays were both much longer than the delay times used in the second section. After a period of time in which the musicians took turns adding their own motivic and rhythmic elements to the group constructed polyrhythm, the infrared controller was used to make pitch changes to the material in the delay lines. This was done using a frequency domain pitch shift, which has a distinct quality: since the pitch and time factors are independent of one another in this algorithm, it is possible to alter the pitch of the polyrhythmic elements without disturbing the rhythm or timing.

The final section (Gregorian Chant) used the same sort of setup as part three, only with much longer delay lines. The musical idea was for the performers to each add a long unwavering tone, which was added to the delay line with input gain being controlled by the infrared controller. As each note was added to the composite chord being created, the idea was for the note to be as unwavering as possible, so it would not modulate as it circulated through the delays, and so the beginning and end of the delayed signal could not be detected, creating a wide "carpet of sound". Toward the end of the piece, the pitch of the sonic material being circulated through the delays was shifted independently of time, using the infrared "Twin Towers" as the control source of gain as well. This was done in an improvised way, that in this particular performance resulted in a modulation of volume much like a very rhythmic tremolo. The musicians were instructed to wait until some time had passed and then begin improvising with the altered sound "carpet" until the piece came to its conclusion.

FAR/COSMOS/CAST at CNMAT

A consortium of CNMAT at UC Berkeley, CCRMA at Stanford, IRCAM, and other changing partners, has been working on an analysis and real-time synthesis system over the past several years. The system (formerly known as FAR, COSMOS, and now as CAST) runs on a Silicon Graphics Machine under UNIX, using a Macintosh running Max as a front end for control by telnet or UDP. Only a few compositions have been written for the system to date. Although real-time synthesis is accomplished, and real-time manipulations of "format" files containing spectral information are possible, all analysis is currently done off-line. This is largely due to the operating system and hardware of the Silicon Graphics Machine, which is not as optimized for audio as was the black box NeXT system. That the analysis is done off-line should not impede a composer from composing a piece that is interactive in every way for the performer and uses the performers gestures to manipulate their own audio.

porting pieces

As Todd Winkler observed in 1991, there is no standardization between effects-processing devices concerning the modulation of effects parameters and the usage of MIDI continuous controllers. How a device responds and which continuous controller numbers control which parameter is different from unit to unit. Some devices only allow for the modulation of 2 parameters at a time, while others can have all parameters be modulated, at any time. In addition, scaling of parameter numbers varies from machine to machine.

In order to begin to port Chew/Buzz/SLAM to another platform it is necessary that I be able to specify with much more accuracy the actual parameters and physical algorithms being used on the Eventide. The Max patch I wrote was specific to a particular piece of equipment, but should be easily converted to the algorithms and internal structure of a different piece of effects-processing hardware. In general it will be easier to port to a processor of "equal or greater value", or simply to a general purpose computer capable of real-time DSP. The greater the flexibility of the platform to which the piece will be ported, the more likely to be successful.

porting to the Eventide DSP4000:

As mentioned above, to port my piece "Chew/Buzz/SLAM", to a different piece of equipment it was necessary to first create more detailed specifications about what effects-processing is being done in the piece. As one would expect it is more possible to port a piece from a dedicated effects-processor to a general purpose computer (one capable of real-time effects-processing), though it certainly may be more time consuming.

On the other hand, as was mentioned above, when moving a piece from one piece of dedicated effects-processing hardware to another, there are many considerations and factors that add up to the success of the endeavor or lack thereof. It is certainly a prerequisite that the effects-processor that is chosen for the port be at least as powerful or flexible as the original effects-processor for which it was written. This is not necessary if the effects specified in the piece are fairly simple and can be accommodated on any basic effects-processor (see the discussion of Jean-Claude Risset's piece below).

Moving the piece to the DSP 4000 made by Eventide proved to be relatively simple. There are several reasons for this. The unit is highly programmable, and there is an excellent MIDI implementation. Most parameters can be mapped on the fly by selecting from one of several MIDI controllers on any channel to control the parameters chosen. To directly address parameters, however, one needs to use System Exclusive messages rather than extended parameter numbers as on the H3000.

I found that if I selected similar parameters and algorithms as were used on the H3000 for "Chew/Buzz/SLAM" (or build the algorithms myself), the sound of the musical gestures was easily replicated on the new effects-processor.

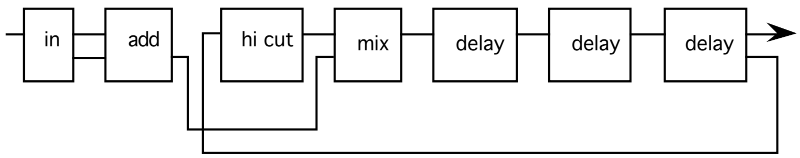
There were three patches used on the Eventide H3000 in the performance of the piece: Ping Pong (a third party patch made by ModFactory), Long DigiPlex and Reverse Shift. Each of them had several algorithms that were applied to their parameters as described above, and each had preset parameter values that were used as starting points for the improvisations and real-time modulations of those parameters made by the performer.

The DSP4000 has many excellent preset algorithms, but the user can design their own patches as well. With the box and line "patch editor" new algorithms and presets can be created from a toolbox of DSP tools including delays, multipliers and adders, mixers, filters, pitch-shifters, as well as many higher level algorithms and control modules. Since the H3000 manual supplies functional diagrams of each of the algorithms upon which its presets are based, it was a simple task to recreate the algorithms for the piece.

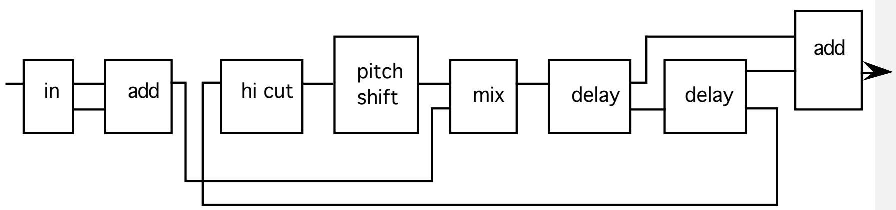
Maximum delay available in any of the delay objects is 660ms, but in most "patches" the delay units are strung together to create longer delay times. This only presented a problem when attempting to create long reverse reverbs (see below). Also mapping the delay time changes that are used in the piece requires the computations of the values needed for each delay unit used to create the delay time needed, either internally in the patch itself or in the Max patch or other program that is running the unit in performance.

"Ping Pong", is a slap back echo. It has an unvarying delay time, with a controllable feedback loop, high frequency damping and detune (slight pitch shifting). This algorithm would be fairly straightforward to replicate on the DSP4000, but I found that a preset "Frippertronics" is close in function to the "Ping Pong" algorithm, missing only the pitch shift (which can be simply added to it using the "patch editor"). The several delays chained together to create a static delay time, a high-cut module and the added pitch shift in the feedback loop, were all that were needed to create the sound that was used in the first part of the piece.

Frippertronics

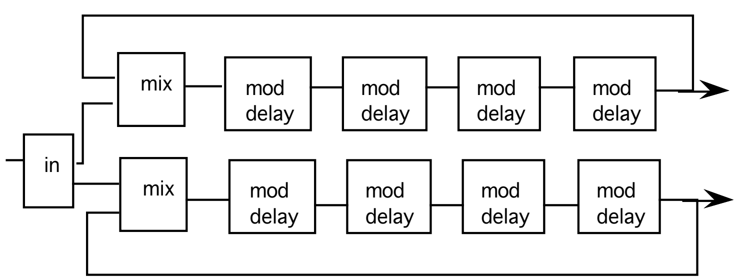


Frippertronics modified for Chew/Buzz/SLAM - Ping-Pong



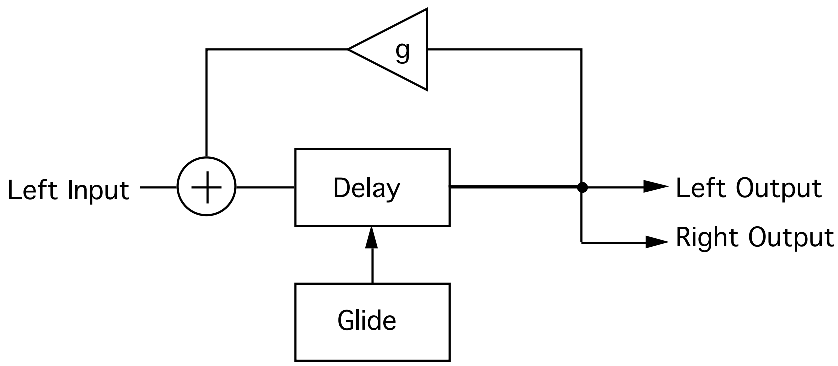
The second part of the piece used the "Long DigiPlex" algorithm on the H3000, which is a 1400ms interpolating delay line with a feedback loop and glide control on the interpolation. A preset on the DSP4000 (Dual Long Delay) is very similar and can be modified to suit the needs of this section of the piece.

Dual Long Delay DSP4000 Patch

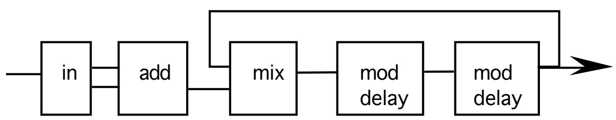


Several of the delay modules, each of which can be set to a maximum of 660ms of delay, can be removed from the DSP4000 preset since only 1400ms were used (read: available) in the original implementation of the piece on the H3000. With high feedback values (gain) and short delay times (1-30ms), I was able to exactly replicate the Karplus-Strong type plucked string sounds when inputting short impulses that were achieved on the H3000 as well as many of the other effects that I initially thought were anomalies to the H3000.

Long Digiplex H3000 Algorithm



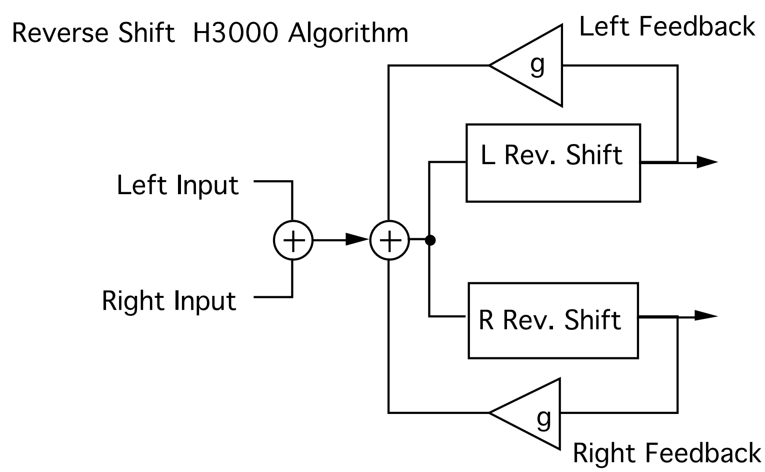
Dual Long Delay DSP4000 Patch modified



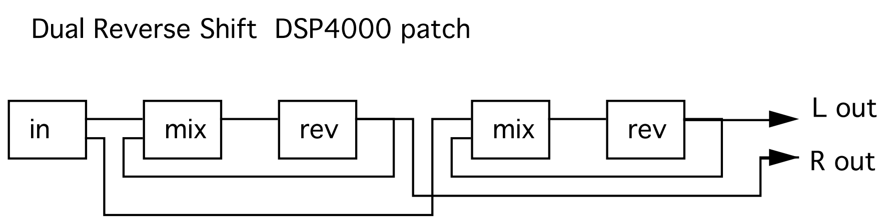
The **moddelay** object (an interpolating delay) in the DSP4000 has a glide speed control as does "Long DigiPlex" the H3000 (up to 1000ms per second of delay change), and so it can easily be limited to what was available in the previous implementation of the piece. The **mix** object handles gain control in the feedback loop. The entire Dual Long Delay patch was simplified to recreate the H3000's "Long DigiPlex" with a successful outcome: if given identical parameter values, the result of changing the delay times is a Doppler shift with the same musical effect as the one created with "Long DigiPlex".

The most interesting part of this experiment by far has been the process of porting the next section of the piece. I gained insight by observing the elements that ported successfully and those that did not. The failures in some cases indicated that the algorithm needed some adjustment or that a different approach needed to be taken. In other cases the port was impossible because the original version of the piece capitalized on "features" of the H3000 that were in reality implementation problems (depending on your perspective) that were improved in the DSP4000. This will be explained below.

"Dual Reverse-Shift" is a DSP4000 preset that allows pitch shifts of 3 octaves up or down, but keeps the output of the two delay lines separate, so it was not close enough to the algorithm of the H3000 to use. Instead I built a patch based on the H3000 algorithm.



[Eventide H3000 Manual]



[from front panel]

The reverse delay module on the DSP4000 has a delay "length" of up to 660ms (how much audio it can reverse at one time) and a variable time for the onset of the reverse shifted audio (called "delay" in the patch). The direction of the delay can be changed from reverse to forward at any time, causing the delay to be converted to a normally functioning delay unit with a pitch shift. The pitch-shift available is more than adequate: the maximum pitch shift values are much greater than on the H3000: up to 3 octaves up or down.

Reverse delay modules can be chained together as the delay modules to create the desired length of delay time. This however represents the first real difficulty I encountered in porting the piece. It is necessary to chain together 3 reverse reverb modules for each channel, setting the maximum delay to a total of1400ms. Note that the three delay modules must all be used anytime the delay time needed is over 660ms. If only two are used, the reversed audio is reversed a second time and is output playing forward (!). By using three units the audio is reversed again and output in reverse as desired. The length of the delay lines and distribution amongst the three modules could be computed externally in a computer program or Max patch.

The pitch shift on all three reverse delay units should be controlled by the same MIDI controller number so they move in unison. Alternately, the pitch shift values could be kept at zero in all the reverse delay modules and a pitch shifter put at the beginning or the end of the chain (its placement will affect only the first pass through in an obvious way.) Other interesting effects could be created if independent control of pitch shift was kept in the individual delays, but this was not an option in the original implementation of the piece on the H3000.

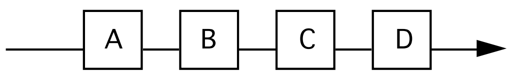
The relationship between delay time and length is also interesting. The delay time determines at what point in the delay line the audio starts being reversed. If delay and length are set to the same number a comb-filtering effect is created because only approximately 1ms of audio is reversed. On the H3000 the audio segments in their entirety were reversed (up to 1400ms), and so the delay should be set to zero to recreate the same effects as were used in Chew/Buzz/SLAM.

The feedback gain on the mixer module was initially set to 0dB, and instead of feeding back or oscillating as the H3000 does, the DSP4000, presumably as a safety measure, simply shuts off outgoing audio when the levels become too high.

porting to the DP/4:

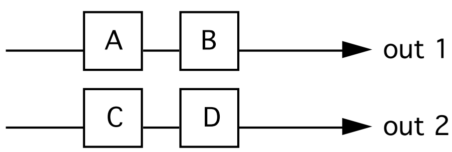
Ensoniq makes a flexible and powerful dedicated effects-processor built around four of their own ESP1 effects-processor chips. The unit was designed so that input and output, as well as internal processing, could be configured in a variety of ways. There are 4 inputs that can be configured as separate mono inputs, 2 stereo pair, a stereo pair and 1 mono input, or as one input (mono or stereo). Internally, there are 4 effects (corresponding to the 4 ESP chips) that can have one of 32 possible routing configurations between them. A pair of effects (AB or CD) can be routed in parallel, serially or with one of two configurations that involve a feedback loop. The pairs of effects can then likewise be routed similarly.

AB - serial, CD - serial, AB-CD serial

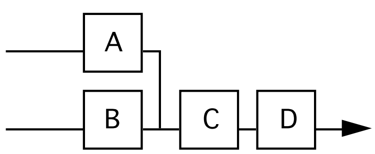


Example DP/4 effects routings configurations

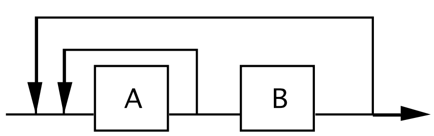
AB - serial, CD - serial AB-CD parallel



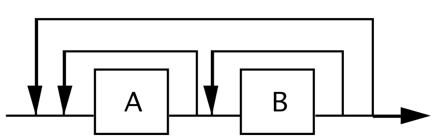
AB - parallel, CD - serial, AB-CD serial



DP/4 Feedback configurations:



Feedback 1



Feedback 2

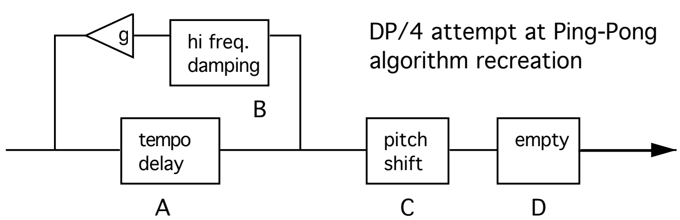
Parameter values on the DP/4 are sometimes scaled differently internally than externally in the user display. This makes it difficult to know exactly what to expect at a certain value without refering to the Sysex specification, where the "fraction type translation table" has been provided to translate the internal values into displayed values. The scaling is predetermined and not controllable by the user. Below is an example from the DP/4 Sysex implementation:

table of scaling values from DP/4 SYSEX chart.

Dual Delay parameters	Internal	Displayed
Mix	0,127	0,99
Volume	0,127	0,99
Left Input Delay Time	0,840	0,840
Left Input Delay Time (fine)	0,99	0,99
Left Input Delay Regen	0,127	0,99
Left Input Delay Pan	-128,127	-99,99
-		
-		
-		

In trying to port Chew/Buzz/SLAM , I found that although the DP/4 is very flexible in architecture and has high quality 24-bit signal processors, the unit is generally too different from the Eventide to recreate this particular piece. Even though there are 32 different internal routing configurations for the processors, it is easy to imagine a situation in which one might need a configuration that is not provided. The first section of Chew/Buzz/SLAM is just one such instance.

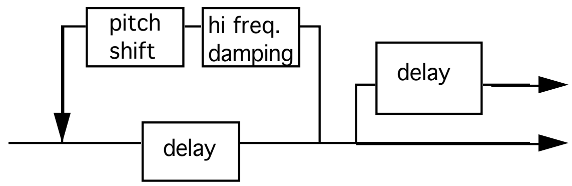
I tried to recreate the Ping-Pong algorithm made by ModFactory (a third party developer) for the Eventide H3000. Ping-Pong (as was described above) is a slapback echo with pitch-shift and high frequency damping as parameters. The delay time is fixed and not modulated.



DP/4 attempt at Ping-Pong algorithm recreation

A-B use feedback 1 routing
C-D use serial routing
AB-CD uses serial routing

The pitch shift and damping could be placed just about anywhere in the chain with the same effect on the overall sound. The audio, however, needs to be output in stereo, bouncing individual echoes back and forth between the right and left channels, each delay being approximately 400ms.



The two delays are each 400ms, but there is output from one or the other of the delays every 200ms. To do this the sound has to be both output by a delay <u>and</u> passed to a second delay module (this models what is physically happening in a slapback echo), or alternately, to prevent both delays from "firing" at the same time, an additional delay (a one-shot delay without feedback) of 200ms needs to be put on one of the channels.

The first of those two choices is nearly impossible, since not one of the available processing architectures on the DP/4 allows for a sound to be fed to another DSP module and output at the same time. The closest possible routing is to set two delays to a feedback architecture that allows the dry mix of the first to completely bypass the second effect as well, and then set the wet/dry mix to 50%. Using the "Dual Delay" algorithm this sounded like 200ms apart 400ms delays, but not like Ping-Pong, because of the lack of panning. The panning of the delays is a compositional element in the piece. The tape accompaniment that comes in toward the end of the first section includes material that bounces between hard left and right creating a spatialization effect and echoing the parts of the opening. This precludes the use of a simple 200ms repeating delay.

To attempt to recreate Ping-Pong by using a Multitap delay, taps were set at 200ms and at 400ms. The feedback loops for the delays are set for each tap and there is no control for an overall feedback level. Therefore the resultant delays, even if they are panned hard left and right, sound like panned 200ms and 400ms delays rather than like a slap back delay. This is because of the extra repetition of the 200ms delay in one of the channels. It may be possible to use a second multitap in series with its output in a feedback configuration feeding the input of the first multitap delay.

The delays in the second multitap delay would be set to 0ms and all sound simply passed through and fed back to the first delay. This is one way to create an overall feedback for the multitap delay. Delays are created at 200ms, 400ms, 600ms, 800ms, and so on, and are panned alternately left and right. The feedback must be set high enough to create the very slow decay of a slapback echo. Unfortunately, the buildup of noise is quite fast if the feedback is high.

Part 2 Long DigiPlex:

It is possible to create plucked string sounds as were used in the second section of Chew/Buzz/SLAM by sending an impulse into the processor configured with very short delay times and high feedback. As was just noted, however, the DP/4 is prohibitively noisy at high feedback levels for many applications.

There are no interpolating delays on the DP/4, making the Doppler effects created by the change in delay time impossible to recreate. There are some pitch shift algorithms that interpolate between changed values yielding a similar musical result that could possibly serve as a substitute for the Doppler effects.

There are several very good pitch shift algorithms, each of which is optimized for a different function, (e.g., the fast pitch shift for pitch correction works very quickly but only with a range of a semitone up or down). Other pitch shifts have a very large range compared to the H3000. The pitch shift also has a different sound than that of the H3000. This is most likely a function of the filters used in the DACs as well as the internal sample rate of the unit.

The MIDI implementation of the DP/4 will allow 2 parameters to be modulated per DSP unit. The same controller number can be mapped to several units for more global control (e.g., one controller can be mapped for feedback gain in every module).

Obviously, in porting a piece from one platform to another a great degree of flexibility is needed from the new piece of hardware to be used. Porting a piece from DP/4 to the Eventide H3000 probably would be no less difficult to do than a port from the H3000 to the DP/4 -- perhaps even impossible given that the DP/4 has 4 processors and therefore 4 different effects that can be simultaneously used. Porting a piece to the DSP4000 or a general purpose computer will almost certainly be possible, because of its option to create custom configurations and algorithms.

Ron Smith

As mentioned above, Ron Smith's Esquisse was composed for a real-time software synthesis unit called "Waveguides". To facilitate the performance of the piece by performers with access to more commercially available equipment he later ported the piece to the Ensoniq DP/4 effects-processor, controlled by custom software written in the Max programming environment, and a commercial editor/librarian program. He chose the DP/4 predominantly because of the fine degree of control possible over parameters (for pitch shift and delay time) and because it is a relatively inexpensive unit that many performers already own.

Smith bases many of his harmonizations on harmonic and inharmonic spectra, and so was by having control over pitch shift in cents he can very closely approximate these spectra on the DP/4. Many of the more inexpensive effects-processors, which he evaluated before settling on the DP/4, model their pitch harmonization upon equal tempered preset or user-defined scales. Smith found (as did Mari Kimura) that this implementation, while very useful for most commercial music applications, is inferior for the realization of pieces that were originally written for much more sophisticated hardware.

porting to the Digitech TSR-24:

The Eventide H3000 uses two continuous controllers (14-bits) to create a large range of pitch shifts. The Ensoniq DP/4 uses internal scaling of 0 to 127 to get smooth pitch shifting over a large range of values, and can increase its pitch shift if several pitch-shifts modules are linked serially. The Digitech TSR-24 uses yet another pitch shift paradigm, which unfortunately makes it a bad host for Chew/Buzz/SLAM.

Pitch shift in this case is accomplished by pitch shifting the input to a pitch that is specified as a MIDI note number, and then sending a pitchbend value for any values between the semitones. This makes a continuously changing pitch shift greater than a whole tone sound quite different from the same process on the Eventide H3000 or the Ensoniq DP/4. On a long glissando it is possible to hear the pitch shift hiccup as it adjusts to the next MIDI note number and value for pitch shift, after reaching the maximum pitch bend for the note above or below it.

porting pieces from the 4X to ISPW and SGI:

When the ISPW was first introduced, Cort Lippe (and others working at IRCAM) spent long periods of time porting interactive pieces for the 4x machine to the new platform. It was a pursuit likely to be repeated by computer musicians for years to come -- as new and more powerful technologies are made available and affordable -- fueled by the desire for longevity and continued playability of the pieces of music for signal processing environments.

Many pieces in the IRCAM repertoire that were originally written for the 4x machine needed to be ported to ISPW to allow the pieces to continue to be performed using the new hardware. Using the 4x, as many three machines were required to technically run a piece, although simpler configurations were possible: the 4x for DSP, a 68000 machine host machine to run the 4x and a Macintosh running Max to communicate with the 4x through MIDI. Once the pieces were transferred to the ISPW, control code and signal processing code all cohabited one machine.

The ISPW is capable of all processes that existed on the 4X. The issues surrounding the port of pieces from the one platform to the other had to do with the allocation of resources and differences in the way the two machines handle sample data.

The difference in speed between the two machines in negligible. The 4X machine has 8 processor cards whereas on a fully loaded ISPW has the equivalent power in 6 cards (3 sets of dual Intel i860 cards). In porting pieces, it was necessary to consider on which card individual elements or processes in a compositional algorithm reside on the 4X and how to redistribute these processes to 6 cards of equal capacity, keeping in mind the subtle differences in each machine's operation.

The 4x machine is "sample synchronous" -- meaning all operations are made on a sample by sample basis. The ISPW works with blocks of samples, making the machine more appropriate for analysis/resynthesis methods which also work with sample blocks. The use of sample blocks introduces a latency of operation that does not exist on the 4X machine. Sample synchronous operations are more appropriate for direct synthesis. The ISPW is therefore more efficient when doing real-time analysis/resynthesis (FFT/IFFTs), but according to Lippe, the oscillators are better on the 4X machine. The two machines are essentially equivalent in terms of power and flexibility.

With the demise of the black box NeXT hardware, porting pieces created at IRCAM has become an issue once again. The SGI platform is now running a version of DSP Max (although not commercially available as of yet), in a software-only version of the program that uses the host processor of the SGI for processes. This works well for "1 card pieces" -- pieces that did not use a fully loaded ISPW but just one i860 card. Pieces that were written using more real-time DSP functions or FFTs cannot yet be fully ported to this platform. Phillipe Manoury's "Pluton", successfully ported from 4X to ISPW has yet to have a replacement version on the SGI that uses its full configuration. In August the new SGI R10000 chip will be released which may be powerful enough to enable the port of full 4X or ISPW pieces to that platform.

To ensure that a piece can continue to be played in the future without having to maintain outdated or unsupported hardware, and to ensure that a piece can be played by musicians with a variety of signal processors, requires some careful forethought.

conclusion

There are many different approaches to the use of real-time effects-processing in a compositional context. The range of styles represented is not purely a matter of aesthetics and taste. How a given composer composes in this medium is affected by several factors other than style. The type(s) of equipment and software available to the composer, as well as the composer's ability to understand and control the digital signal processes that they are using affect and determine how they will compose for the medium. A composer attempting to write using a dedicated effects-processor will find many "preset" effects, programmed by professional sound developers, that they can use effectively for compositional uses. However, it is easy to fall into cliché if these sounds are not developed further and made dynamic as part of the composing process. By investigating the acoustical properties of the basic time-domain effects found on dedicated effects-processors, a composer can improve their ability to work with a variety of interesting real-time effects using relatively inexpensive equipment.

As far as notation is concerned, there are many different kinds of descriptions that can be made for the real-time effects-processing in a piece. A technical description alone for the musical behavior of a piece is not enough. This would be much the same as a traditionally notating piece of music in which the instrument to be played (and perhaps its tuning) is indicated without any further indication of <u>how</u> the instrument is to be played (e.g., the tape splice diagrams used as notation for Cage's "William's Mix").

Technical descriptions vary too, as some composers only provide parameter values for a specific piece of equipment (e.g., pitch shift values, amount of reverb), while others provide detailed diagrams of the construction and configuration of the DSP algorithms. The descriptions of the algorithms say more about the actual sound of the piece to a person who is trained to understand those algorithms and DSP functions, but is less helpful to an initiated performer who needs to realize the piece on a different dedicated effects-processor.

In Howard Risatt's <u>New Music Vocabulary</u> written in 1975, a guide to notation for contemporary music, only four notational descriptions of "amplified sounds" are included: two of them indicating an amount of amplification, one about when to start and stop the tape portion of a piece, and one that is simply "reverb on" and "reverb off". This can adequately describe a simple process only. Descriptions of effects-processing likewise cannot be limited to parameter descriptions, unless the processes are themselves are simple or will never be ported to different platform.

Many electroacoustic pieces for soloist and tape include graphic notation to help the live performer recognize musical events in the tape part by their behavior or timbre or anything other than the time lapsed since the beginning of the piece. Graphical notation and verbal descriptions of the behavior expected from a real-time effects-processor, as well as detailed diagrams of the actual algorithms are all needed to completely notate a piece of any complexity in which the effects-processing is a central component of the composition. To ensure the piece's playability in a variety of settings, and its survival through changes in the available and affordable technology, it is necessary to make a music specification for a piece transcends the particular equipment and software for which it was written.

In researching the sounds and transformations found in my own "Chew/Buzz/SLAM" I found that some of the effects that I assumed to be anomalies of the Eventide H3000 were in fact typical by-products of using unorthodox parameter values for some of the effects algorithms, and can be recreated on a different system. Other by-products were the result of the specific sampling rate and hardware specifications, and although they could be musically recreated (perhaps through synthesis), they were not musically important and will be left behind in a port of the piece to another platform. In attempting to describe more accurately the effects that I used both technically and in terms of their behavior with various kinds of input, I learned much about the general behavior of time-based effects-processing algorithms, and this knowledge will carry over to future pieces on the Eventide H3000 and other platforms.

Many composers choose to push the limits of the available technology or to write for specific pieces of equipment. This is their choice to not limit their aesthetic decisions to a common denominator in available technology. This also does not rule out a later port of their pieces to more powerful or programmable platforms, but does limit the number of performances that can be given to the piece in the meantime. Composers will eventually become more savvy about many kinds of real-time effects as more powerful technology and real-time frequency domain processors become commercially available and accessible, even if their knowledge is only gained experientially. The techniques used in pieces that push the processing limit of our real-time effects systems today will be much more commonplace for the average composer, and those who push the limits will probably be exploring new technical boundaries with their music.

Appendix

Bibliography

Alten, Stanley R. Audio in Media, Fourth Edition. Belmont, CA: Wadsworth Publishing Company, 1994.

Chamberlin, Hal. Musical Applications of Microprocessors. Hasbrouck Heights, NJ: Hayden Book Company, 1985.

Chambers, Evan. "Computer Music World View: Sketch of an Ethnomusicological and Aesthetic Approach." Proceedings of the International Computer Music Conference, October 1994.

Dechelle, De Cecco, Puckette & Zicarelli. "The IRCAM 'Real-Time Platform': Evolution and Perspectives." Proceedings of the International Computer Music Conference, October 1994.

Ensoniq Corporation. DP/4 MIDI SysEx Implementation Specification version 1.00. Malvern, PA.

Ensoniq Corporation. DP/4 Parallel Effects Processor Musician's Manual v. 1.02. Malvern, PA.

Eventide Corporation. H3000SE UltraHarmonizer Instruction Manual. Little Ferry, NJ., 1989.

Eventide Corporation. DSP4000 Operation Manual. Little Ferry, NJ.

Garnett, Guy. "Music, Signals, and Representations: A Survey." Representation of Musical Signals, ed. G. de Poli, A. Piccialli, and C. Roads. Cambridge, MA: MIT Press, 1991.

Horn, Delton T. Basic Electronics Theory. Blue Ridge Summit, PA: TAB Books, 1989.

Kimura, Mari. "Performance Practice in Computer Music." Computer Music Journal, Spring 1995, pp. 64-74.

Kimura, Mari. Interview and concert program notes. New York University, May 1996.

Larke, Kevin. Interviews and course notes. New York University, Spring 1996

Lippe, Corte. Interview and personal communication. March 1996, August 1994

Lippe, Corte & Miller Puckette. "Musical Performance Using the IRCAM Workstation." Proceedings of the International Computer Music Conference, Montreal, Canada,1991.

Lippe, Corte & Zack Settl. "Real-Time Musical Applications using FFT-based Resynthesis." Proceedings of the International Computer Music Conference , Aarhus, Denmark, 1994.

Loy, Gareth. "Musicians Make a Standard: The MIDI Phenomenon." The Music Machine. ed. Curtis Roads. Cambridge, MA: MIT Press, 1985.

Moore, F. Richard. Elements of Computer Music. Englewood Cliffs, NJ: Prentice Hall, 1990.

Moore, F. Richard. "The Dysfunctions of MIDI." Computer Music Journal 12(1):19-28 1988.

Moorer, J.A. About This Reverberation Business, Paris: Centre Georges Pompidou, 1978.

Pecquet, Frank. "Music and Technology." Array (Newsletter of the International Computer Music Association), Vol 15, No. 2., Summer 1995.

Pohlmann, Ken. Advanced Digital Audio. Carmel, IN: SAMS, 1991.

Pohlmann, Ken. Principles of Digital Audio. Carmel, IN: SAMS, 1985 and 1989.

Pressing, Jeff. "Novelty, Progress and Research Methods in Computer Music Composition." Proceedings of the International Computer Music Conference. Aarhus, Denmark, October 1994.

Risatti, Howard. New Music Vocabulary. Urbana, IL: University of Illinois Press, 1975.

Risset, Jean-Claude. "Variant for violin and treatment numeric." unpublished score, 1994.

Rowe, Robert. Interactive Music Systems. Cambridge, MA: MIT Press, 1993.

Settel, Z. and C. Lippe. "Real-Time Musical Applications using FFT-based Resynthesis." Proceedings of the International Computer Music Conference. Aarhus, Denmark, October 1994.

Settel, Z., C. Lippe, M. Puckette and Eric Lindemann. "Recent developments: The IRCAM Signal Processing Workstation." Proceedings of the International Computer Music Conference. San Jose, CA, October 1992.

Singer, Eric. Interview and personal communication, July 1996.

Smith, Julius O. "Fundamentals of Digital Filter Theory." The Music Machine. ed. Curtis Roads. Cambridge, MA: MIT Press, 1985.

Smith, Julius O. "An Introduction to Digital Filter Theory." Digital Audio Signal Processing: an Anthology. ed. J. Strawn. Los Altos, CA: Kaufmann, 1983.

Smith, Ronald Bruce. Interview and personal communication. August 1996.

Smith, Ronald Bruce. "Esquisse.", unpublished score, 1992.

Tarabella, Leonello. Interview, personal communication, diagram/score. July 1994, May 1996.

Winkler, Todd. "Interactive Signal Processing for Acoustic Instruments." Proceedings of the International Computer Music Conference. Montreal, Canada, 1991.

Woram, John. Sound Recording Handbook. Indianapolis, IN: Howard W. SAMS & Company, 1989.

Zvonar, Richard. Interview. September, 1995.