ABSTRACT

By analyzing an instrumental sound into several orthogonal, time variant descriptors, the sound production hardware and the sound synthesis software may be simplified. With this approach each descriptor can be sampled at a rate consistent with its rate of change (bandwidth). Sampling in multiple domains retains much of the universality of direct sampling, but because of the data compression method the number of descriptors is a function of the sound complexity, not its duration. This author has designed and constructed a digital polyphonic synthesizer for this purpose.

In this implementation, a complete sound is represented by a sequence of timbres, each defined by a fixed length waveform table; a sequence of envelope samples; a pitch period; and an envelope period. As in the classic wavetable waveform synthesizer, these tables consist of instantaneous samples of the desired waveform, read out to a D/A converter at an integer sub-multiple of the pitch period, and repeated at the pitch period. Here, however, a list of waveform tables defines the sequence of waveshapes that make up the sound, providing dynamic timbres. Each waveform table can be considered to be a sample of the sound timbre. Each envelope samples the amplitude output of the waveform D/A is weighted with the samples of the envelopes. The envelope magnitude in a logarithmic multiplying D/A converter. This pseudo-floating point approach effectively increases the digital resolution and keeps the wavetable quantization noise at a constant level below the current envelope magnitude. The envelope magnitude and the waveform table used are determined by the envelope period, synchronized such that switching is not audible. Since the execution of even a complex sound is a straightforward operation of list processing, the sound production hardware is readily implemented with function-dense, large-scale integration devices (Direct Memory Access Controllers and Multifunction Programmable Controllers). This greatly permits the construction of a complete 8-voice polyphonic synthesizer with less than 850 k.c.m. for the sound production and 325 k.c.m. for the supervisory micro-computer hardware. The architecture is programmable in the sense that the descriptor tables and table lengths for each voice are independently specifiable.

Several techniques may be used to build the waveform tables and the amplitude tables. In all cases these advantages that the building algorithms are run offline, not in real time. Borrowing a term from the speech synthesis world, each descriptor set (a wavetable, envelope sample and a pitch period) might be considered to be a "phone", and sounds are constructed in a building block fashion as a time composite of these phones. Additionally, a real acoustic instrument (or the output of an analytic model for an instrument) may be sampled, using short-time Fourier analysis and envelope detection to define the waveform tables and the envelope samples.

1. INTRODUCTION

digital instruments by nature must build their approximations to the continuous, analog sound. A real acoustic instrument from a sequence of quantized, discrete samples. These samples represent samples of the audio signal, or sound pressure level. The fidelity with which the complexity of a real sound can be reproduced is a function of the number of quantization levels (the number of bits, or digits, per sample) the rate of sampling, given enough digits and enough samples, a signal of arbitrary complexity, duration, and dynamic range can be reproduced. In a practical system, the storage...
requirements will ultimately limit the resolution and sample rate.

With the direct sampling Pulse-Code Modulation (PCM) technique used in digital recording, the audio signal is encoded as time-domain samples, hopscotching on at an 12-bit sufficient to represent the most rapidly changing component, the highest frequency in the signal. The approach to be developed here diversifies the sampling process by sampling several parameters simultaneously. The sampling density necessary is that necessary to represent the most rapidly changing parameter. This multidimensional sampling allows a more flexible trade-off between sampling density, fidelity, and hardware complexity.

2. DESCRIPTION

The hierarchal scheme employed here achieves a substantial data compression, or bit rate reduction, by segmenting the sampling process into several dimensions. This approach permits the sampling of each dimension at a rate and resolution that is most appropriate. The several dimensions or descriptors used here to describe a complete sound are timbre, pitch period and envelope amplitude.

At the highest level of the hierarchy, the subject sound is partitioned into M equal-length time segments, extending from the attack event through the final decay event. Each of the time segments is represented by one set of the descriptors. In the direct sampling case, the shorter the period, the better the representation, but it is not the rapidly changing sound pressure level that is sampled. Each time segment is represented by a sample of the envelope, period, the pitch period, and the timbre during that segment.

The pitch and the number of segments used and thus the sampling rate will depend on the rate of change of these descriptors.

The timbre 'samples' are made up of a table of samples themselves, time domain samples of a cycle (or an integer number of cycles) of the current waveform. These timbre samples can also be explained as representing the short-time spectrum of the waveform signal [1]. Reconstruction takes place as in a fixed-waveform, waveform synthesizer, by sequentially reading out the table to a Digital-to-Analog Converter (DAC), at an integer multiple (M) of the pitch period. For each of the M segments of the sound, then, a possibly unique waveform is employed and a succession of waveforms and therefore timbres result.

A somewhat accurate analog can be drawn with the analog synthesizer: the data table contents are chosen to give a constant, normalized average level from the DAC. The periodic output of the wavetable to the DAC can be modeled as a Voltage-Controlled Oscillator (VCO). The VCO period is the pitch period and its waveform is determined by the wavetable used. The waveforms, and thus the VCO waveforms, are the variant. The waveform from the DAC output is combined with the envelope sample in a four-quadrant multiplying DAC. The multiplying DAC provides the digital equivalent of the Voltage-Controlled Amplifier (VCA).

The waveforms and the envelope values are updated synchronously with the cyclic output of the waveform such that the changes occur at a known, fixed location or phase of the waveform. Choice of a transition point at the lowest energy zero crossing minimizes the transient effects of the wavetable and envelope changes.

The waveform DAC employed in this implementation is a linear, 8-bit wide, and all the waveforms are 8-bits wide. The waveform and envelope table lengths are programmable and can be from 7 to 32,768 samples long. Up to 256 distinct wavetables can be executed in sequence to define one sound. The envelope DAC is a logarithmic, 8-bit wide, 1.5 dB weighting. It provides an effective dynamic range of about 70 dB. The exponential weighting of the envelope samples is ideally suited to the logarithmic sensitivity of human hearing. When the envelope DAC has a constant amplitude and the envelope is impressed upon the waveform after the wave is generated, the noise from the waveform DAC is also modulated by the envelope. As a result the quantization noise remains a constant level below the waveform level. In this case with an 8-bit converter the quantization noise is a constant 52 dB below the full-scale output level [2].

While an 8-bit resolution would not be adequate for a direct sampled system, it works here for two reasons. The waveform DAC is always operated with a full-scale input, so that the problem of decreasing signal-to-noise ratio with decreasing signal level does not appear. Also, the quantization noise created by the wavetable DAC occurs at an integer multiple of the pitch period. This synchronous, harmonic noise is quite tolerable and easy to remove with filters that track the pitch period.

A major goal of the chosen coding method and its implementation was to permit a hardware design that is regular, easily
expandable/contractable, programmable, and exploits the high-level functionality available with today's large-scale integration (LSI) integrated circuits. These qualities also fit well with a realization in custom LSI, which would permit inclusion of some powerful extensions without significantly affecting the design complexity.

In the present implementation, designed with off-the-shelf components, all major functions are provided by LSI devices. Figure 1 depicts the system block diagram. The table look-up operations for the envelope sequence, timbre sequence and the wavetables are executed by Direct Memory Access (DMA) controllers (AMD 8517A) clocked by programmable counters (Intel 8254). Because sound definition is not done in real-time, a microprocessor (Intel 8086) is used to implement the operator interface, to build the descriptor tables, and to supervise the DMA controllers and timers during sound execution. The processor environment includes 12k Bytes of Random-Access Memory (RAM), 16k Bytes of Read-Only Memory (ROM), a serial interface to a development system, and a serial port for use with the operator console. Interfaces are included to scan a 61-key keyboard, and an array of front panel controls and displays. All software is written in the FORTH language, and a complete FORTH interpreter, compiler and text editor is resident in the system ROM.

All descriptor tables reside in a central real-time memory which allows access by both the DMA controllers and the microprocessor. New sounds can be built by the microprocessor while the DMA controllers execute previously defined sounds.

Eight voices were implemented, and each voice is capable of executing a unique set of descriptors. Each voice has two DMA channels dedicated to the look-up of the descriptor tables. Figure 2 depicts these memory reference operations. A waveform DMA channel cycles at a multiple of the pitch period, employing the auto-initialize capability of the controller, repetitively addressing the current wavetable. The contents are latched and presented to a linear, 8-bit DAC. A double-buffering scheme guarantees that the memory latency time does not produce phase jitter in the output waveforms.

The second DMA channel serves two purposes: to look-up the envelope samples, and to look-up a pointer to the current wavetable. The envelope samples are fetched from the central memory and latched to drive the logarithmic, multiplying DAC (Analog Devices AD7118). The latch strobe is synchronized with the waveform DMA channel so that the envelope update and the waveform update always occurs at the same point in the wavetable.

The pointer to the current wavetable is additively combined with the address generated by the waveform DMA controller. In effect, the wavetable pointer provides a base address that indicates the current waveform, with the waveform DMA controller providing the offset, or phase variable within the wavetable. This indirect addressing re-definition of a wavetable sequence as a simple matter of changing a list, the waveforms themselves need not be moved.

FIGURE 1

FIGURE 2
Also, the wavetable sequence may be unequally sampled to permit efficient representation of, for example, rapid timbre variations during the attack with much slower variations through sustain. The types of counting sequences executed by the envelope DAC are dependent upon the nature of the sound desired. A percussive sound, with a fixed sustain duration, can be implemented with a single, one-shot sequence. More complex sounds, requiring a variable sustain period, are realized with three part sequences: a one-shot attack/decay, a repetitive sustain, and a final one-shot release period.

The output of the envelope DAC is fed to a 6th order maximally flat (Butterworth) low-pass filter to remove the noise components of the waveform and above. The filter is a monolithic, switched-capacitor device (National Semiconductor MF4), with cutoff frequency set by a clock input. The clock frequency for each voice is determined by a programmable counter. Generally, the filter frequency is a constant factor beyond the pitch period.

1. FEATURES

Among the advantages of this approach to sound synthesis are:

1. The descriptors chosen: timbre, pitch, and envelope amplitude, are readily perceivable and lend themselves to direct manipulation, in an interactive manner by the composer or performer.
2. The descriptors: tables are not constructed in real-time. This means that the complexity of the construction method does not restrict the realizability. Quite complex or calculation intensive sequences can be built without burdening the hardware during sound execution.
3. The compression results from the storage method. In comparison to direct sampling, the storage necessary is reduced for sounds with static timbres, or timbres that changes slowly with respect to the pitch period.
4. In comparison to synthesizers that rely on algorithmic methods to calculate the sound in real-time, the hardware complexity is greatly reduced. The only operations that take place in real-time are counting and table look-up.
5. The timbre sequence table look-up is implemented with a level of indirection, that is, the address of the wavetable to be used is itself fetched from a sequence in memory. Because of this indirection, the timbre tables may change at a rate slower than the envelope period. This allows efficient representations of sounds where the timbre rate-of-change varies.

As with any sampled-data system, the fidelity is a function of the information density in the sound to be constructed and the information density in the coding method. For arbitrary amounts of storage and rates of sampling, signals of arbitrary complexity can be represented. The programmability of this system permits a wide choice in sampling densities and thereby a choice in the accuracy of the representation, however, the approach to an ideal representation comes from a quite different direction than that of a direct sampled PCM system.

As a result it places unique constraints on the reproducibility of a given sound:

1. The partials present in the sound must be integer multiples of the fundamental frequency (f = period) of the wavetable. Note that the wavetable period is not necessarily the pitch period, but may be any integer multiple of the pitch period. If integer X is the ratio of the wavetable period to the pitch period, the available lines in the spectrum, or the available harmonic components, will be spaced at an interval of [f / X]. Thus as the table is lengthened, containing more cycles of the pitch period, the spectrum, or timbres is better defined.
2. The wavetables must be defined such that adjacent timbres are close enough that the transition is inaudible. This would normally result from an analysis-synthesis where the sampling density can be arbitrarily specified. Other ad hoc techniques of timbre sequence definition may require interpolation if discontinuities are not desired.
3. The present implementation does not support, in isolation, the pitch period at the same rate as the timbre. Due to complications of hardware complexity, the pitch period is set by the microprocessor only once for each sounding.

4. METHODS OF DEFINITION

Much of the descriptive model employed here assumes an analysis-synthesis method of sound definition. That is, the acoustic output of a real instrument is sampled and analyzed to produce the descriptor table. Alternatively, the sampled signal is the output of a sound production algorithm or an analytic model for a musical instrument.

There are several possible approaches to the analysis algorithms, and the pitch period can be determined by one cycle sampled at [N * pitch period] will
provide the table contents. In this case the signal would only have to be sampled as finely as the table increment. More generally, however, the pitch period will be found after sampling, and the sampling is not synchronous with the pitch period. In this case, a multirate signal processing technique can be used to filter and resample the original signal at the desired multiple of the pitch period.\[3\]

A timbre sequence can also be derived from a given initial waveform by recursive linear transformation. With the initial waveform built by additive synthesis, for example, the successive timbres are defined as functions of the previous table. If a linear transformation is used, this method of waveform construction is termed "subtractive synthesis" as spectral components present in the original waveform can only be removed. Given a transformation with a low-pass characteristic for example, will result in longer durations for the lower partials and comparatively shorter for the higher partials.

A little more formally, an initial table, with waveform \(s_i(n)\) is cyclically convolved with the filtering function \(h(n)\) yielding a new sequence \(x_i(n)\). \(x_i(n)\) denotes an \(N\)-length wavetable, the \(i\)th table of the timbre sequence. For each table the discrete Fourier transform \(X_i(n)\) is calculated, where \(X_i(n) = \sum \text{deltacyclconv} \rightarrow (n) \otimes h(n)\)

where \(\otimes\) denotes cyclic convolution. In the spectral domain, denoting the Discrete Fourier Transform of \(X_i(n)\) as \(X_i(w)\),

\[
X_i(w) = \text{DFT}[x_i(n)] = X(w) * H(w).
\]

and

\[
X_i(w) = X(w) * H(w)
\]

where \(*\) denotes multiplication.

This approach facilitates the generation of timbre sequences from minimal defining parameters and has been used extensively by the author. Particularly where \(h(n)\) contains complex poles, the resultant timbre sequence can be quite interesting. The transformation operator can also be non-linear, though in this event it could not be described as a convolution and the spectral domain results would be dependent upon \(x(n)\). This simple method can be used to generate complex timbre sequences. An interesting variant results when the timbre sequence is reconstructed in retrograde order. The sequence built by subtractive synthesis can be executed to appear as a non-recursive anti-causal additive sequence that is not necessarily realizable with hardware that must calculate waveforms in forward-time.

The directly perceptible descriptors used allow the ad hoc generation of timbres and envelope sequences. Particularly in the case of sounds with nearly discontinuous timbres, such as the chirp of an opening organ pipe or the snap of a harpsichord plucked, a "cut and paste" method of building the timbre sequence is effective. This technique can be extended to construct sounds from a set of predefined sequences or phonemes. From a basis set of a limited number of phonemes an effectively infinite number of sounds may be built without the need to rebuild the wavetables.

Where discontinuities in the timbre sequence are not desired, interpolation may be necessary between arbitrarily defined waveforms. Interpolation of timbres presents some interesting challenges as the interpolation is in the spectral domain. The \(N\)-length wavetable defines a frequency spectrum \(B(w)\) that is related to the wavetable by the discrete Fourier transform:

\[
B(w) = X(w) * H(w)
\]

Expressed this way, the sampling problem in the spectral domain is determined by the bandwidth or rate of change of each of the \(N\) individual Fourier coefficients. It is necessary to sample the timbre at a rate sufficient to contain the fastest changing coefficient. The aliasing caused by inadequate sampling of the timbre results in artificial discontinuities in the spectral components. Since the construction filter for the spectral samples is a zero-order hold, results from the periodic repetition of the wavetable, interpolation between successive timbres may be necessary. In practice, since the timbres are defined by a time domain waveform, this interpolation would not take place in real-time, but during the building of the tables.

5. EXTENSIONS

There are three major ways in which the performance and flexibility of the present hardware implementation can be enhanced. First, the technique of FM synthesis \[4\] can be combined with the time variant wavetable approach by using a high frequency VCO to generate the wavetable clock. An additional flexibility results from the ability to arbitrarily define the "carrier", which need not be sinusoidal. The resultant spectrum thus contains a series of tensor products and is given by:

\[
\sum_{i=1}^{N} a_i \cdot B_i(w)
\]

where \(a_i\) are the amplitudes of each timbre, and \(B_i(w)\) are their frequency spectra. The final spectrum is then given by a weighted sum of the individual timbre spectra.
harmonics of the carrier, each modulated by the same source. This enhancement would provide a greater variety of possible timbres and also allow pitch changes for vibrato, bending and glissando effects.

Two other extensions are enhancements of the current hardware that have not been incorporated due to the substantial number of devices involved. This is more a result of the use of highly integrated LSI devices than due to inherent complexity and a design based on custom LSI would undoubtedly include them. First, the pitch period should be sampled at least as rapidly as the timbre. Second, the output reconstruction filter clock could be made more efficiently and accurately produced with the same programmable counters used to determine the pitch period. This way the filter cutoff frequency would follow the pitch period.

6. SUMMARY

This paper has presented a relatively simple and straightforward method of coding musical sounds, while retaining a certain amount of sound complexity.

The major points developed here are:

1. The sound is analysed into several constituent components, or descriptors, sampled at short-times timbre, envelope, and pitch period. This results in a bit rate expression for some sounds.

2. The implementation of the synthesis hardware is a regular, organized array of counting operators (DMA controllers and programmable counters) and random access memory. This regularity lends itself well to LSI and the use of high level functions.

3. The mathematics involved in sound construction is not run in real-time. The descriptor definitions are done off-line, while the execution of a sound involves simply sequential table look-up.

4. The descriptors are directly perceivable and thus can be used as control parameters to be directly manipulated by the user or real-time performer.

REFERENCES


ICMC '85 Proceedings

38