A MIDI-BASED AURAL SIMULATION SYSTEM

James Mazanowski
Reflectone Incorporated
Tampa, Florida

ABSTRACT

An aural simulation system which takes advantage of the standard Musical Instrument Digital Interface (MIDI) provides improved performance and greater flexibility in recreating the sound environment of a device being simulated. Traditional methods of generating aural cues using additive synthesis techniques have proven to be costly and often ineffective in accurately simulating complex sounds, due to a limited number of harmonic-producing elements and envelope shapes available for each sound. In the MIDI-based approach, aural cues are produced by digital sampling modules under the control of a dedicated microcomputer with built-in MIDI controllers. The sampling modules have a large amount of memory, and are capable of playing back loops of actual recorded sounds, as well as loops produced from mathematically generated waveforms or synthesizers. The MIDI interface, which was originally designed as a communications link between electronic musical instruments, is also well suited to aural simulation tasks, since it contains all the commands necessary for sound effects activation and manipulation in real- time. These commands, which include note on and off (used for enabling and disabling playback of samples), pitch bending, and channel pressure (used for amplitude changes), allow any type of aural cue to be generated. Cross-fading among multiple samples is used to reproduce dynamically varying aural cues with great accuracy. Transient and steady-state aural cues are programmed quickly and are reproduced with relatively short samples. This paper describes the hardware and software implementation of the MIDI-based aural simulation system, and how it provides a more realistic and cost-effective reproduction of the simulator sound environment. Emphasis is placed on the integration of sound analysis and sample manipulation tools into the system, and on details of managing MIDI command transfers in real-time.

INTRODUCTION

The clear-cut goal of any sound synthesis method is to reproduce the spectral content of a sound so exactly that it cannot be distinguished from the its real-world counterpart. This goal poses a considerable engineering challenge, since real-world sounds often have very complex frequency and amplitude characteristics which can change dramatically over the full operating range of the device producing the sound. In recent years, however, significant advances in digital audio and digital signal processing (DSP)[1] technology have brought the possibility of exact sound reproduction closer to reality. Audio synthesis techniques have progressed to the point where virtually any desired waveform can be precisely generated and controlled in the digital domain. Portable digital audio tape (DAT) recorders allow actual sounds to be recorded in the field without loss of fidelity, and sophisticated analysis tools allow these sounds to be accurately analyzed and manipulated in the frequency domain.

Many significant advancements in digital audio synthesis and recording methods have come from the music industry. The impact of synthesizers on modern music production is well known. Digital sampling modules, also known as samplers, have added further sound generation capability in the form of actual recorded sound segments, commonly referred to as wavesamples. Samplers, which began as relatively simple audio loop recorders, have become much more elaborate with the addition of DSP features such as dynamic digital filtering and automatic loop finding. Some of the more recent samplers have introduced a SCSI bus port to allow hard disk drive storage of sample files. Built-in sequencers are also available in some samplers. A sequencer can be used for invoking certain timed aural cue scenarios, as well as self-test and demonstration functions.

Perhaps the most important new technology developed for musicians is the Musical Instrument Digital Interface (MIDI). The MIDI interface allows a wide variety of instruments from many different manufacturers to communicate with each other and with personal computers. The standard MIDI command set includes many commands which are readily adaptable to aural simulation needs. In addition, the MIDI sample dump standard allows for the transfer of sample files in either direction between a controlling computer and a sampler. This feature opens the door for any DSP-generated waveform or DSP-manipulated sample to be utilized in a sampler. This effectively adds the capability of arbitrary waveform generation to the system, greatly increasing its flexibility.

Taking advantage of the powerful combination of DSP software tools, sampling technology and the MIDI interface, an aural simulation system has been designed which presents aural cues using the "instruments" of one or more sampling modules. The system provides the following features:

- Sampling modules with eight instruments, autolooping, dynamic digital filtering, built-in sequencer, and a variety of DSP commands
- Dedicated system controller PC with eight MIDI output channels and two MIDI input channels
- SCSI hard disk drive for high-capacity storage of sound files
- Simultaneous playing of instruments while loading files to/from disk
- Up to 127 wavesamples per instrument
- 40 selectable sample rates, from 6.25 kHz to 52.1 kHz
- Wavesample panning, for precise positioning of sounds within a stereo field

THEORY AND ANALYSIS OF SOUND IN SIMULATION

No matter how complex a real-world sound is, its simulation can be made more manageable by breaking the sound down into a combination of less complex sounds. In fact, if one had the time and the resources to do so, the sound could be broken down into a mixture of a very large number of sine waves. For as the French mathematician Baron Jean Baptiste Joseph Fourier (1768-1830) stated, any sound, or any signal for that matter, is actually composed of a number (perhaps an infinite number) of (co-)sinusoidal components each with a unique amplitude and relative phase. The Fourier transform[2], which is the basis for frequency analysis, takes on different forms depending on the type of signal being analyzed. In the digital domain, which for simulation purposes is the area of primary interest, it is known as the discrete Fourier transform (DFT).

The DFT is a Fourier representation of a sequence of finite length, such as one period of a sampled function, and it yields a unique Fourier series for all periodic waveforms. A computationally efficient set of algorithms for the DFT is known collectively as the fast Fourier transform (FFT). The FFT is without a doubt the single most important signal analysis tool available. One major advantage the FFT has over other methods of frequency analysis is that it retains phase relationships, allowing spectral manipulation in the frequency domain, and a relatively straightforward transformation back to the time domain using the inverse FFT.

Signal Types

A generalized discussion of signal types[3] is readily applicable to the various types of sounds encountered in aural simulation practice. The type of signal determines how it is analyzed as well as how it fits into an aural simulation scheme. Figure 1 shows the basic divisions into different signal types. The two most basic types of signals are stationary and non-stationary. Stationary functions are those whose average properties do not vary with time and therefore are independent of the particular sample record used to determine them. Stationary signals are also commonly referred to as steady-state signals. These signals can be either deterministic or random.

The instantaneous value of a stationary deterministic signal is predictable at all points in time, that is, it has periodicity. Periodic signals are entirely composed of sine waves at discrete frequencies which are multiples of some fundamental frequency. The general representation of the Fourier series for periodic signals is given in Eq. (1).

$$f(t)=\frac{a_0}{2}+\sum_{n=1}^{\infty}(a_n\cos n\omega t+b_n\sin n\omega t) \qquad (1)$$

If T is the period in time of waveform $f(t)$, then the angular frequency ω is always equal to $2\pi/T$. The coefficients a_n and b_n in Eq. (1) are determined by Eqs. (2) and (3), respectively.

$$a_n=\frac{2}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} f(t)\cos n\omega t\,dt \quad (n=0,1,2\ldots) \qquad (2)$$

$$b_n=\frac{2}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} f(t)\sin n\omega t\,dt \quad (n=1,2,3\ldots) \qquad (3)$$

Equation (1) can be rewritten in another form, as shown in Eq. (4). This representation of the Fourier

$$f(t)=\frac{a_0}{2}+\sum_{n=1}^{\infty} c_n\cos(n\omega t-\Psi_n) \qquad (4)$$

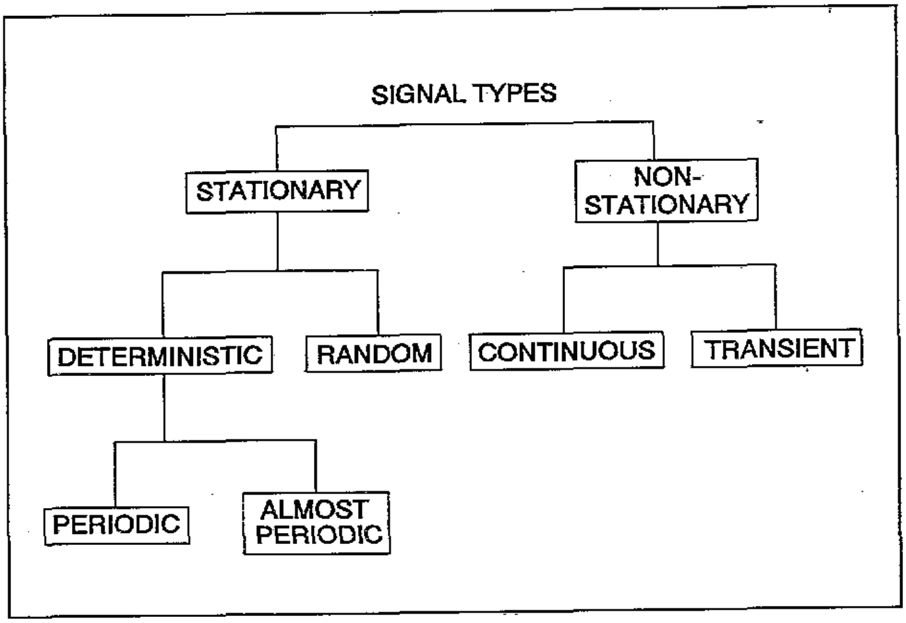

Figure 1. Division into different signal types

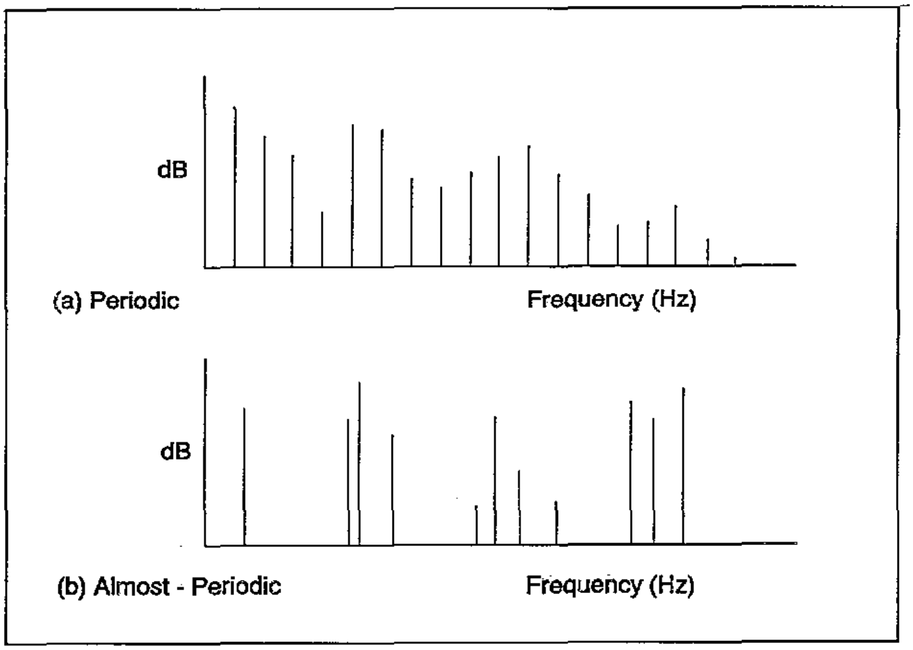

Figure 2. Typical periodic and aperiodic spectra

series shows the periodic function as the summation of an average DC value and harmonically related sine waves having unique relative phases. The amplitude and relative phases of the sine waves are given by Eqs. (5) and (6), respectively. Figure 2a shows

$$c_n=\sqrt{a_n^2+b_n^2} \tag{5}$$

$$\Psi_n=\tan^{-1}\left(\frac{b_n}{a_n}\right) \tag{6}$$

a graph of the spectra of a typical periodic signal. An example of this type of signal in the audio realm is the whine of an hydraulic pump motor when the demanded pump pressure is constant. The fast spin-up of the motor, if it is required to be simulated, is analyzed separately as a transient signal, described under non-stationary signals below.

Real-world variations in the harmonic content of periodic sounds produce a subclass of periodic signals known as complex periodic. This type of signal class is characterized by significant deviations from ideal harmonic relationships, such as those exhibited by sawtooth waves and triangle waves. In complex periodic signals, individual harmonics are often larger than previous harmonics, due to factors such as structural resonances, misaligned rotational parts, looseness of parts, etc. In certain cases, there may be only a few components present, with the fundamental and several of the lowest frequency harmonics absent. For example, a waveform with three sine wave components at 45 Hz, 60 Hz and 80 Hz has a fundamental frequency of 5 Hz, and all values of c_n in Eq. (4) are zero except n=9, 12 and 16.

In almost-periodic signals, the frequencies of the various sinusoids are not harmonically related. Technically, this means that the ratio between at least two of the frequencies must be an irrational number. However, in practice it is typical for almost-periodic signals to be a mixture of two or more independent sets of harmonics, such as the whines of a jet engine with two independently rotating shafts. Figure 2b shows a graph of the spectra of a typical almost-periodic signal.

Stationary random signals are known only in terms of properties such as mean values, bandwidths, probability density curves, etc. This is because their instantaneous values cannot be predicted. A prime example of this type of signal is air rush noise. Many stationary real-world sounds have properties associated with both deterministic and random components, such as the sound of a ground power cart diesel engine. However, these components can often be attributed to separate dynamic processes occurring in the sound-producing device and can therefore be analyzed and simulated separately. A summation of the simulated components reproduces the complete sound.

Non-stationary signals can be divided into continuous and transient signals. Continuous non-stationary signals are those which are also often referred to as dynamically varying signals. Two well-known examples in the aural simulation field are the jet turbine engine and the airplane propeller. Transient signals are those which start and finish at zero amplitude, such as the sound of a relay click. Actually, all non-stationary signals start and eventually finish at zero amplitude; however, transient signals do so in a relatively short period of time. Transient signals are therefore analyzed as a whole, whereas dynamically varying signals are usually analyzed in short segments, each of which will often be almost-stationary, or even stationary within specific regimes, such as when a jet turbine engine is operating with constant shaft speeds.

Analysis and Implementation

Implementation of stationary periodic aural cues using a sampler is straightforward. As is the case for making all samples, a clean, isolated recording of the sound is required. A highly directional microphone (e.g. a shotgun microphone) can be very useful in obtaining the needed isolation. In practice, a good sample loop for this type of sound usually requires fewer than 50 cycles of the waveform. If it is not possible to obtain an isolated recording of the sound, it can be simulated with a DSP-generated waveform obtained by extracting the spectral components known to be associated with the sound. For example, the backup hydraulic pump motor in an SH-60F helicopter receives its power from either the aircraft's auxiliary power unit or one of the aircraft's engines. Therefore the whine of the hydraulic pump is always in the presence of offending noise. A directional microphone is of no use in this case since each of these sound producing devices is in the same general location in the aircraft.

Analysis of the hydraulic pump whine shows that it resembles a sawtooth-like waveform but with higher frequency harmonics suppressed. Reproducing the first 30 harmonics is sufficient to create a very close approximation of the actual waveform. Frequency and amplitude values of the spectral peaks are input to the DSP waveform generator, which produces the simulated time series waveform. A format conversion utility and a sample uploader program are then invoked to transmit the waveform data to the sampler. Since a mathematically generated waveform has perfect zero crossing points, a clean sample loop can be made from it in only one or two cycles, further conserving sampler memory. Once within the sampler, a small amount of amplitude and frequency randomizing of the waveform is desirable. This is accomplished using low frequency oscillators (LFO's) built into the sampler.

Almost periodic sounds present a somewhat more complicated problem. The multiple sets of harmonics in these sounds must be simulated with multiple samples, especially if dynamic control over each set of harmonics is desired. In the case of the jet engine with two independently rotating shafts, it is common for one shaft to be stationary while the other is changing speed. Here, the harmonic families of each shaft are extracted in the frequency domain, using FFT analysis, and two synthesized samples are produced using the DSP waveform generation tool.

Stationary random sounds and stationary sounds which contain both random and periodic components are usually excellent candidates for sample loops. While they demand considerably longer loops than those required for periodic sounds, their length in time is not unreasonable. It has been demonstrated that aural cues as diverse as air rush, ground power cart diesel engines, and electric fans can be produced from sample loops on the order of one to two seconds long (representing between 4.4 and 8.8% of the sampler's available memory at a 44.6 kHz sample rate). Some sounds in this category must be simulated without any actual recording. An example is the change in air rush sound produced by aircraft icing. In this case, 1/3 octave band FFT analysis of normal air rush sound provides the baseline spectral data, and a digital equalizer is employed to boost or cut the amplitudes of each 1/3 octave band as required to produce the characteristic hiss sound caused by the icing condition.

Stationary transient sounds are also excellent candidates for direct sampling. Most transients have a duration of less than two seconds, so they do not have a large impact on memory utilization. If the duration of a transient is significantly longer than two seconds, then the designer may opt for synthesizing it to conserve sampler memory.

Analysis of the transient involves determining the characteristics of both the envelope and the signature spectrum, i.e. the spectrum of the transient within the envelope. The envelope signal is extracted by taking the Hilbert transform of the entire transient signal. The envelope data is used to program the sampler's built-in amplitude envelope generator. Analysis of the signature spectrum of the transient is made easier if it is first made into sample loop. When the sample loop is played continuously the signal becomes artificially periodic, and FFT analysis is then used to determine the spectral content of a synthesized waveform which duplicates the signature of the transient. To recreate the transient, a sample loop of this waveform is played with its amplitude decaying according to the sampler's envelope generator.

Non-stationary continuous (dynamically varying) sounds are analyzed in small segments which are as close to stationary as possible. In a typical scenario, sound recordings are made as the sound producing device is exercised through its full operating regime. The recordings are then subjected to segment by segment FFT analysis. Whenever a significant change in spectral content occurs, corresponding controlling parameter changes are noted, and separate sample loops are derived to replicate each segment of the spectrum.

Cross-fading is used to transition smoothly between samples. When pitch changes are required, care must be taken to ensure that only periodic components are shifted in pitch, since noise frequencies cannot be shifted more than a few percent without producing undesirable effects to the sound. Since the sampler has a pitch bend limit of one octave, a separate wavesample is required for each octave change in pitch. Memory can be conserved in this process by copying only a wavesample's parameter table, and transposing the root frequency of the "new" wavesample one octave via the parameter table. In this manner, the sampler uses the same wavesample data to produce sounds with pitch changes over multiple octaves.

The sound of an airplane propeller transitioning in and out of zero degrees blade pitch angle provides an interesting example of the analysis and implementation of a dynamically varying aural cue. Spectrum analysis of the sound when the propeller has zero degrees pitch angle reveals a prominent fundamental frequency component equivalent to the rotational speed of the propeller shaft, and prominent components for harmonics 1 through 3. Harmonics 4 through 11, while present, are significantly suppressed in relation to the lower frequency components and therefore do not contribute significantly to the sound at this point.

Further analysis shows that as the propeller blade angle is increased and the propeller begins to build forward thrust, harmonics 4 through 11 increase markedly in amplitude at the same time their spectral lines widen (indicating a change to more of a narrow-band noise function). Increasing blade angle does not significantly affect the frequency components below the 4th harmonic, however. Therefore, reproducing this sound necessitates separating these two sets of harmonics. This was successfully done by extracting the two sets of harmonics into two periodic waveforms, and then making the waveform for harmonics 4 through 11 multi-timbral to simulate the narrow-band noise change.

To make a periodic waveform multi-timbral its sample is first copied to multiple instruments. Next, the waveform copies are detuned slightly in frequency above and below the original frequency, to effectively spread the spectrum into the narrow band. Finally, the composite signal is digitally recorded on a DAT recorder and resampled into a single wavesample. This wavesample can be successfully pitch-bended through the entire propeller RPM range.

Frequency Discrimination

An aural simulation system must produce a large enough number of different tone frequencies so that transitions between adjacent frequencies are perceived as smooth to simulator occupants. The least difference in frequency between two tones that is barely detectable to a listener is known as the difference limen (Δf), or the just noticeable difference frequency. Studies in human hearing[4] have produced graphs of Δf versus frequency for tones at different levels above threshold. Results show that Δf is remarkably constant from 100 Hz to 2 kHz and then increases dramatically above 2 kHz. Graphs of the relative difference limen ($\Delta f/f$) have also been plotted versus the frequency. The point at which $\Delta f/f$ is the smallest, i.e. the point at which frequency discrimination in human hearing reaches its peak, occurs in the region near 2 kHz. In the case of almost all real-world aircraft sounds, individual frequency components are less than 30 dB above threshold. For 30 dB above threshold, the difference limen at 2 kHz is 5 Hz. Therefore, the tone generating device must be able to provide a frequency delta less than 5 Hz in this frequency region. The sampler was tested for the difference limen with a frequency bend range of +1/2 octave selected. The closest frequencies to 2000 Hz selectable were 1998.20 Hz and 2003.04 Hz. The resulting pitch delta is 4.84 Hz which meets the criterion.

Loudness Discrimination

The aural simulation system must also produce a large enough number of different loudness levels so that a transition from one level to either the next loudest or next softest level is perceived as smooth to simulator occupants. The smallest loudness change in a tone that is perceived as a distinct change to a listener is known as the difference limen (ΔI), or the just noticeable difference in intensity. Further hearing studies[4] have produced graphs of $\Delta I/I$, the relative difference limen, versus the level of the tone above threshold for three different frequencies. For a 1000 Hz tone, $\Delta I/I$ varies from 1.0 at 5 dB above threshold to 0.24 at 100 dB above threshold. Measurements of the sampler have shown that it exceeds the loudness discrimination criterion at all intensity points. The results apply equally well to any waveform the sampler produces.

AURAL SIMULATION METHODS

Traditional aural simulation methods have grown steadily in sophistication and cue generation capacity in recent years. As recently as a decade ago, when analog aural simulation systems were still dominant in the industry, several circuit cards were often required to produce a single aural cue. Hardware size and cost limitations forced designers to not only simplify the harmonic content of those cues which were presented, but also to eliminate altogether a number of cues which were considered to have less training value, but nevertheless were clearly audible to the crew. Improvement came with the first digitally-controlled aural simulation systems based on programmable sound generator integrated circuits. Such features as precise frequency selection of crystal-controlled periodic components, dynamically adjustable filter cutoff frequencies, and additional envelope shapes became available.

With a considerably higher sound generating capacity and a more modular design than their analog counterparts, these digitally controlled systems allowed a somewhat more comprehensive level of aural simulation to take place. In most cases all audible aural cues could be addressed instead of only the most important ones. However, the process of building unique sounds with custom software used an inordinate number of design hours and a point of diminishing returns was eventually reached.

The predominant sound simulation technique used in most of the systems designed before DSP integrated circuits were widely available was subtractive synthesis. This method involved using filters to remove unwanted higher harmonics from oscillator-generated random noise and periodic waveforms. Only the most basic filter configurations and periodic waveform shapes were available, however. The most common periodic waveform available was a triangle wave, produced by filtering a digitally produced square wave with a one-pole low-pass filter. The Fourier series expansion for a triangle wave is shown in Eq. (7). As the equation shows, this waveform contains only odd harmonics, therefore it cannot accurately simulate the many real-world sounds having even harmonics. The result of using only basic waveforms was at best only a fair simulation of the often extremely dense harmonic spectra of real-world sounds.

$$f(t)=\frac{8A}{\pi^2}\left(\sin\omega t-\frac{1}{3^2}\sin 3\omega t+\frac{1}{5^2}\sin 5\omega t-\frac{1}{7^2}\sin 7\omega t+\ldots\right) \quad (7)$$

The advent of low-cost, powerful DSP integrated circuits has made additive synthesis[5] more attractive. In additive synthesis, large numbers of sine waves are added together with precise frequencies and amplitudes to produce a composite sound. Although Fourier theory states that an infinite number of sine waves is required to reproduce any waveform having a Fourier series of infinite length, in practice 200 or so harmonics may be enough to construct a synthesized waveform indistinguishable from the real one. Nevertheless, managing the frequencies and amplitudes of 200 sine waves in real-time at an appropriate sample rate for audio simulation is a formidable task for even the fastest DSP boards available today. For a six channel aural simulation system, the number grows to 1200, and a single board DSP solution becomes very unlikely. A compromise solution employing aspects of both additive and subtractive synthesis remains the most feasible approach to aural cue generation using DSP hardware.

THE MIDI-BASED SYSTEM

Figure 3 shows a top level diagram of the MIDI-based aural simulation system. This is a typical configuration with two samplers and two stereo power amplifiers comprising four audio channels. The aural cue CPU is a PC AT-compatible music computer with 8 MIDI output channels, 2 MIDI input channels and a pop-up monochrome LCD display built in. It has a 3.5" floppy disk and a 20 Megabyte hard disk. Parameter transfers to and from the simulator host computer occur via an Ethernet interface, which during development is connected to a VAX network. Applications requiring smaller parameter transfer blocks can opt for a lower cost configuration which uses the computer's two RS-232C ports for transfers. The low parts count of the system allows for a significant improvement in reliability and reduction in life-cycle cost over previous aural simulation systems. With the exception of one 6-position DIP switch on the back of the music computer, the system is configured entirely in software.

The sampler has a 16 bit data storage format, and a 13 bit sample converter. It has 24 bit floating point internal processing for a 96 dB dynamic range. Memory capacity is one Megaword. Maximum sample times are as follows:

157 sec. at 6.25 kHz, 31.4 sec. at 32.9 kHz, 23.0 sec at 44.6 kHz, and 19.8 sec. at 52.1 kHz.

During real-time operation, it is advantageous to conserve sampler memory, so that a greater number of aural cues can be presented from each sampler without

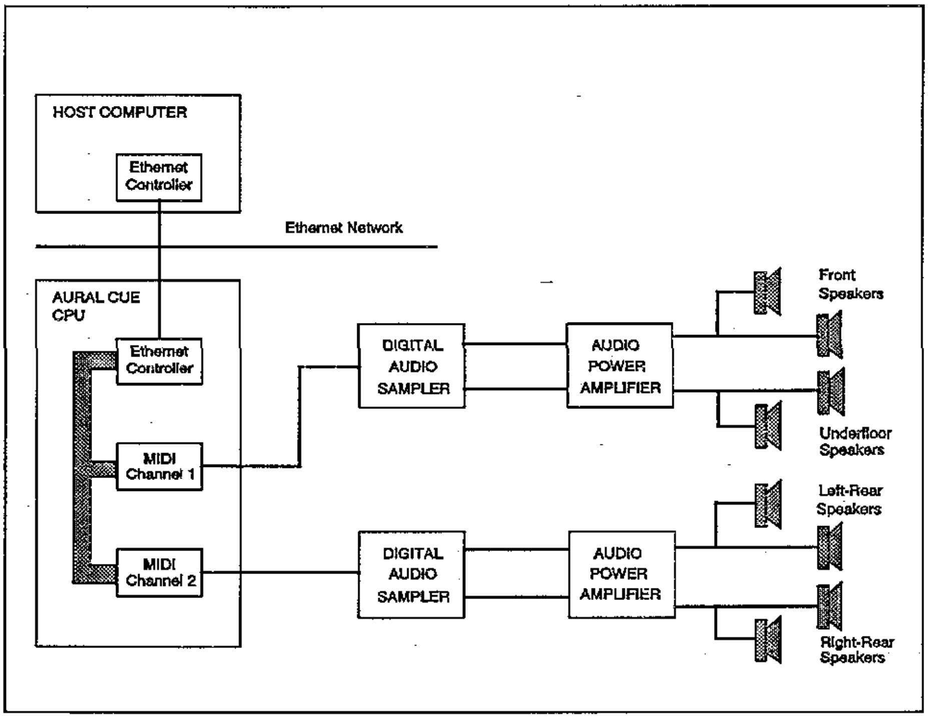

Figure 3. Top Level System Diagram

affecting the amount of spare memory present. Memory is conserved in three ways: First, the sample rate of each individual wavesample is selectable. Therefore, each sample rate can be set to the lowest value which adequately reproduces the bandwidth of the waveform contained in the wavesample.

Second, dynamically varying aural cues are separated into periodic and random components in the frequency domain before conversion back to time domain wavesamples. Frequency variations in these cues most often involve their periodic components while their random components remain relatively constant in pitch. Therefore, separating out these components results in a considerable reduction in the number of wavesamples containing random components. Samples with random components require substantially longer loops and thus more memory than wavesamples having only periodic components. To obtain necessary frequency sweeping of the periodic components, one or more periodic wavesamples are produced for each octave of frequency change. Pitch bending is performed on each wavesample, and a cross-fade algorithm is employed to smoothly transition between adjacent wavesamples. If the harmonic content of adjacent wavesamples is identical, it is not necessary to generate an entirely new wavesample for each octave. In this case only the wavesample parameters are copied, and the root frequency of the new wavesample is transposed one octave. This procedure allows sharing of identical wavedata among multiple wavesamples, and sampler memory is further conserved.

Finally, sounds which are mutually exclusive within specific operating regimes can often share the same portion of memory. Since the sampler can load files into an instrument from disk while simultaneously playing its other instruments, dynamic allocation of instruments in memory can be performed under software control. For example, in a jet aircraft simulator, a ground power cart sound file can be replaced in memory by a rapid decompression sound file as the aircraft becomes airborne and vice versa when the aircraft touches down. With this method, the point of transitioning from one sound file to the other is chosen during a portion of the flight regime when neither of the two sounds occurs. This allows sufficient time for the sound files to be loaded in from disk.

Audio outputs from the sampler consist of left and right stereo jacks, plus eight polyphonic solo outputs. Each wavesample assigned to the stereo outputs is placed in one of eight possible positions in the stereo field. The solo outputs, which can be assigned to entire instruments or individual wavesamples, providing the capability of routing specific sounds to additional speakers or to the simulator communications system.

All system software is modular in nature, and is written exclusively in C language code. Upon boot up of the system, the aural cue run-time program automatically begins executing. The system executive module, first performs its power-up self-testing and initialization, and then proceeds to establish communications with the samplers. Next, the samplers are given commands to load their required instrument files in from disk. The executive module then sends a message to the host computer, either via Ethernet or RS-232C, indicating that it is ready for real-time trainer operation. In the case of a system with an Ethernet interface, the executive program proceeds to execute its background modules while waiting for real-time data transfers. In the case of a system with RS-232C interfaces, the executive program reads the RS-232C input buffer at a 30 Hz rate, looking for new data. If no new data is present, the background modules are executed.

Background modules consist of a status/error processing routine and a parameter monitor routine. The status/error processing routine programs the system's output data block, which is transferred to the host computer at a 30 Hz rate. The parameter monitor routine allows system parameters to be displayed or changed during real-time operation, either from the aural cue computer, or from the host computer, if the host computer's parameter monitor program is operating.

Available off-line utility programs include a high-speed sample editor, and a system self-test module. The high-speed sample editor program provides a variety of menus which allow fast editing of wavesample and instrument parameters directly from the aural cue computer. The self-test module, which includes a daily operational readiness test, exercises each piece of hardware in the system. Test results are available on the aural cue computer's display, and in hard copy format on the host computer's printer.

MIDI COMMAND TRANSFERS

The hardware MIDI interface[6] operates at 31.25 (+/- 1%) Kbaud, asynchronous, with a start bit, 8 data bits, and a stop bit. Therefore, a single 10-bit transfer requires 320 microseconds to complete. Bytes are transferred LSB first. Software command transfers take the form of multi-byte "messages" consisting of one Status byte followed by one or two Data bytes. There are two types of messages: Channel and System. Embedded in each Channel message is a nibble which selects one of 16 voice channels which contain a variety of performance information. The effect that this channel information has on the equipment being controlled depends on the mode the equipment is in. MIDI has four different modes of operation: Omni On/Poly, Omni

On/Mono, Omni Off/Poly, and Omni Off/Mono. Since both Omni On modes do not apply to the aural simulation task they will not be discussed.

The Omni Off/Poly mode is useful during the development of synthesized aural cues. This mode allows the instruments selected on the front panel of the sampler to play the incoming MIDI data on a single channel (known as the base channel) polyphonically. In this mode, multi-timbral sounds can be produced for recording and later resampling.

The Omni Off/Mono mode is used during real-time operation of the system. This mode allows separate control of the parameters of each instrument in the sampler via MIDI voice channel commands. In voice channel commands, the lowest nibble of the first byte contains the number of the channel which receives the command. In the present case, channel numbers 0 through 7 correspond to instrument numbers 1 through 8 of the samplers. Since each sampler has its own assigned MIDI output port on the aural cue computer, each sampler can respond to commands for all eight of its instruments.

The MIDI voice channel commands used most often for the aural simulation application are note on/off, program change, control change, and polyphonic key pressure. Note on/note off commands are used to turn sounds on and off. No wavesample can be heard unless a corresponding note on command has been issued. Note on and off commands invoke envelope generators built into the sampler which provide wavesample attack and decay characteristics. Separate envelope generators are available for pitch, filter frequency and amplitude. A total of six different envelope breakpoints can be programmed, both in terms of duration and level.

Program changes are used to command the sampler to load sound files from disk into specific instruments. Sound files are stored on disk in a number of directories which are classified by major categories, such as engine sounds, hydraulic sounds, built-in test, etc. The sampler treats directories and sound files in the same fashion, e.g. it is possible for a directory to be made up of only other directories. This allows for a sophisticated and very functional directory tree.

Control changes are used for a number of dynamic effects, the most common of which are pitch bending, dynamic digital filtering, and low frequency oscillator control. Polyphonic key pressure commands are used to program the amplitudes of specific wavesamples in real-time. This is done by assigning the wavesamples of a given instrument to separate keys, allowing each wavesample to have its own amplitude setting.

The nature of the MIDI interface is such that transfers are only required when parameters change values. An output transfer module, operating at a 60 Hz iteration rate, is tasked with determining which parameters have changed since the previous iteration. Based on the changes detected, the required transfer commands are programmed based on priority as follows:

A. Periodic waveform pitch changes

B. Random signal filter cutoff frequency changes

C. Periodic waveform amplitude changes

D. Random signal amplitude changes

E. Modulation level changes

F. Stationary signal on/off commands

The order of this list is determined by the relative ability of human hearing to detect the various types frequency and amplitude changes, i.e. the difference limen criteria. Although the exact update rate required for these commands depends on the rate of change of the parameter involved, a 30 Hz update rate is typically sufficient for item A above, a 20 Hz rate for item B, and a 10 Hz rate for items C through F. MIDI channels are programmed so that spare throughput is always available, i.e. no detectable lag occurs in any frequency or amplitude change.

CONCLUSION

A MIDI-based aural simulation system combines the realism of actual recorded sound samples with the versatility of arbitrary waveform generation. The standard commands of the MIDI interface and numerous DSP command built into the sampler speed aural cue software development and implementation. A hardware configuration with very few parts provides high reliability with reduced life cycle costs.

REFERENCES

[1] Oppenheim A.V., Schafer R.W., Digital Signal Processing, Prentice Hall, Englewood Cliffs, NJ, 1975.

[2] Cheng, D.K., Analysis Of Linear Systems, Addison-Wesley, Reading, MA, 1959.

[3] Randall, R.B., Application Of B & K Equipment To Frequency Analysis, Brüel & Kjaer Instruments, Inc., Naerum, Denmark, 1977.

[4] Wever, E.G., Theory Of Hearing, Wiley and Sons, New York, NY, 1949.

[5] Gross, P., "Synthesis Techniques For The 1990's And Beyond", Electronic Musician, February, 1990, pp. 62-74.

[6] MIDI 1.0 Detailed Specification, The International MIDI Association, Los Angeles, CA, 1989.

AUTHOR BIOGRAPHY

James Mazanowski received his BSEE degree from the University of Connecticut in 1977, and joined Reflectone the same year. He has both system and circuit card design experience in several aural simulation and communication systems. He has programmed aural simulation software for more than 10 different aircraft types. He is presently involved in developing software tools used in automating the aural simulation analysis process.