John Stautner and Miller Puckette
Experimental Music Studio
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

Designing Multi-Channel Reverberators

Introduction

We present here a methodology for the design of digital reverberators. Some properties of digital recursive networks that are useful for reverberation simulation are considered along with simplified assumptions of the behavior of sound in rooms. The method presented leads to a wide variety of possible reverberation networks, though only one such possibility is presented in detail.

These results have led to the design of some reasonable four-channel reverberators useful for computer music. While no attempt has been made to imitate the ambience of an existing room or concert hall, the methods described herein may lead to such applications when they are combined with a consideration of the perceptual importance of attributes of the soundfield in a real room.

Overview

In designing reverberators, we must attempt to account for both the early part of the reverberant response and the overall quality of the long-term response. It is very difficult to model exactly the long-term behavior of sound in an enclosure, but use of recursive delay networks can provide a satisfactory and practical approximation. Much can be said about the pattern of sound reflections during the first 100 msec or so of the reverberant response, however. It is convenient as well as perceptually meaningful to simulate these reflections directly by considering the configuration of the room to be modeled.

Both of these methods are present in our design procedure. First, we create a skeletal recursive network where each delay unit is assigned to a speaker. The output of a delay unit feeds a speaker and also feeds one or more of the other delay units in the network. In this way, the echoes spread among the speakers and increase in density during the reverberant response. The feedback pathways are characterized with a feedback matrix, G, and ways of choosing G to yield a stable response are presented.

Next, a specific feedback matrix is chosen that gives a skeletal network appropriate for use with four output channels arranged in a square. Methods of simulating the early response and further refinements of the design procedure are introduced within the context of this particular example.

Finally, some hints for efficient computation are given, and excerpts of a sample Music 11 implementation are shown.

Computer Music Journal, Vol. 6, No. 1, Spring 1982,
0148-9267/82/010052-14 $04.00/0

Model of the General Delay Network

A network of delays may be described as an extension of the comb filter (Fig. 1) whose system equation is usually written

$$y = z^{-\tau}(x + gy), \tag{1}$$

where x is the input, y is the output, τ is the delay in samples, g is the feedback coefficient and $z = e^{i\omega}$. We may generalize this recursive network by replacing the delay with n delays in parallel, each with an input and an output signal. The feedback signal is replaced by n feedback signals, each of which is some combination of the output signals. If we denote the feedback signal to the ith delay unit as f_i, then we have

$$f_i = \sum_{j=1}^{n} g_{ij} y_j. \tag{2}$$

If we adopt the language of matrices, the system equation becomes

$$Y = D(X + GY), \tag{3}$$

where X is a vector whose components are the n

Fig. 1. A comb filter with feedback coefficient g and delay of τ samples.

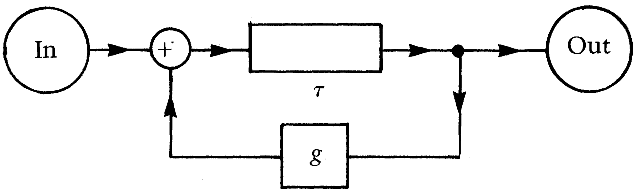
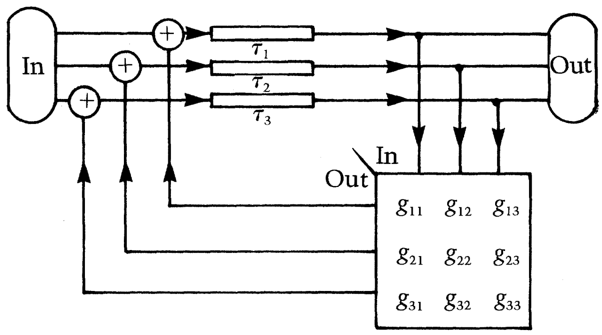


Fig. 2. The network of Fig. 1 generalized to vector signals.



inputs, Y a vector of n outputs, G the matrix with the components in Eq. (2), and D a matrix of the form

$$D = \begin{bmatrix} z^{-\tau_1} & & & & 0 \\ & z^{-\tau_2} & & & \\ & & z^{-\tau_3} & & \\ & & & \cdot & \\ & & & & \cdot \\ 0 & & & & \cdot \end{bmatrix} \qquad (4)$$

The form of the network is shown in Fig. 2.

The network may be modified by introducing other elements such as filters. For simplicity, however, we consider the stability and frequency response of the skeletal network shown. Additional elements will be useful later for imitating early echo patterns as an aid to directionality and enhancement of quality.

Iterating Eq. (3) gives the system function

$$Y = DX + DGDX + DGDGDX + \ldots,$$
$$= D(I + (GD) + (GD)^2 + (GD)^3 + \ldots)X,$$

or

$$Y = D(I + A + A^2 + A^3 + \ldots)X, \qquad (5)$$

where I stands for the identity matrix and A replaces GD for future convenience. The properties of this system function determine most of the qualities of the delay network.

Writing the system function in the form in Eq. (5) makes it clear that a condition for stability is that successive powers of A become smaller instead of larger. To develop sufficient conditions for this, let $|X|$ denote the vector norm of X (in real terms the root-mean-square [RMS] amplitude of the signal X), defined by

$$|X| = \sqrt{|X_1|^2 + |X_2|^2 + \ldots + |X_n|^2}.$$

Then if for any reason we have

$$|AX| < k|X| \qquad (6)$$

for all X and some positive $k < 1$, then the norms of the terms of the expression

$$X + AX + A^2X + A^3X + \ldots$$

decrease exponentially. This implies convergence of the sum, which in turn implies the stability of the network. Since D is unitary, that is, has the property

$$|DX| = |X|,$$

we may rewrite Eq. (6) as

$$|GX| < k|X|. \qquad (7)$$

That the condition in Eq. (7) is also sufficient for stability is also true but by no means obvious.

The simplest means of ensuring the truth of Eq. (7) is to let G take the form

$$G = U \begin{bmatrix} k_1 & & & & 0 \\ & k_2 & & & \\ & & \cdot & & \\ & & & \cdot & \\ 0 & & & & \cdot \end{bmatrix} \qquad (8)$$

where none of the k's exceed k in Eq. (7), and U is unitary. Since we are presently assuming that G has real entries, we may write the condition that U be unitary as

$$\sum_{k=1}^{n} U_{ik}U_{jk} = \begin{cases} 1 & \text{if } i = j; \\ 0 & \text{otherwise.} \end{cases}$$

The simplest example of a unitary matrix is the identity, which gives parallel comb filters for the network. A nontrivial example is

$$U = \begin{bmatrix} \sin(r) & \cos(r) \\ -\cos(r) & \sin(r) \end{bmatrix}$$

for any r. Since the product of two unitary matrices is again unitary, matrices of the above form can be applied repeatedly to mix the outputs of pairs of delays, thus forming more complex networks.

Equation (3) may now be rewritten as

$$Y = \frac{D}{I - A} X, \tag{9}$$

still in analogy to the comb filter. Since D consists only of delays, the numerator has no coloring effect. The denominator acts in much the same way as it does in the conventional comb filter. Multiplying Eq. (9) by $I - A$ gives

$$Y - AY = DX.$$

Since $|AY| < k|Y|$, we have

$$(1 - k)|Y| < |Y - AY| < (1 + k)|Y|.$$

Substituting DX for $Y - AY$, and using the fact that D is unitary,

$$\frac{1}{1 - k} > \frac{|Y|}{|X|} > \frac{1}{1 + k}.$$

This inequality gives an upper and lower bound on the frequency response of the network, taken as the ratio of the total RMS power of the speaker outputs Y to the RMS power of the input signals X. For a simple comb filter, these upper and lower bounds are nothing more than the peaks and troughs of the comb filter frequency response described by Schroeder (1962). When delays of various lengths are used in the general recursive network, however, the frequency response achieves these peaks and troughs at very irregular intervals in frequency, which removes the buzzy sound that comb filters give. The coloration is usually much less bothersome than that of the equivalent number of comb filters in parallel.

A Four-Channel Network

For most of the remainder of this discussion, we will assume the skeletal delay network given in Fig. 3, which is suitable for use in a four-channel playback environment where the speakers are arranged at the corners of a square. The feedback matrix G for this network is given by

$$G = \begin{bmatrix} 0 & 1 & 1 & 0 \\ -1 & 0 & 0 & -1 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \end{bmatrix} \cdot g,$$

where

$$|g| < \frac{1}{\sqrt{2}}.$$

A source signal may be introduced at any point in this network, such as the end of one of the delays.

An advantage of this delay network is that the speaker outputs are mutually incoherent signals that are spatially responsive to the input channel of the source. Past research (Meyer, Burgtorf, and Damaske 1965) has shown that incoherent output signals are important to create a diffuse soundfield, which is characteristic of good concert hall reverberation (Beranek 1962; Schroeder, Gottlob, and Siebrasse 1974). Consider also the growth of the soundfield: if the input is a pulse in channel 1, we first hear an output pulse in channel 1, then in the adjacent channels, and finally in the diagonally opposite channel before the echo density grows and spreads among the four channels. This behavior acts as an additional cue of source position and of the physical extent of the simulated space.

Definition of Early Response

Much research has been done to determine the importance of the early response, which occurs roughly during the first 50–100 msec of the reverberation impulse response. Measurements and simulations of the early reflections in concert halls have demonstrated that time separation, frequency

Fig. 3. Basic feedback network with source introduced in left front channel.

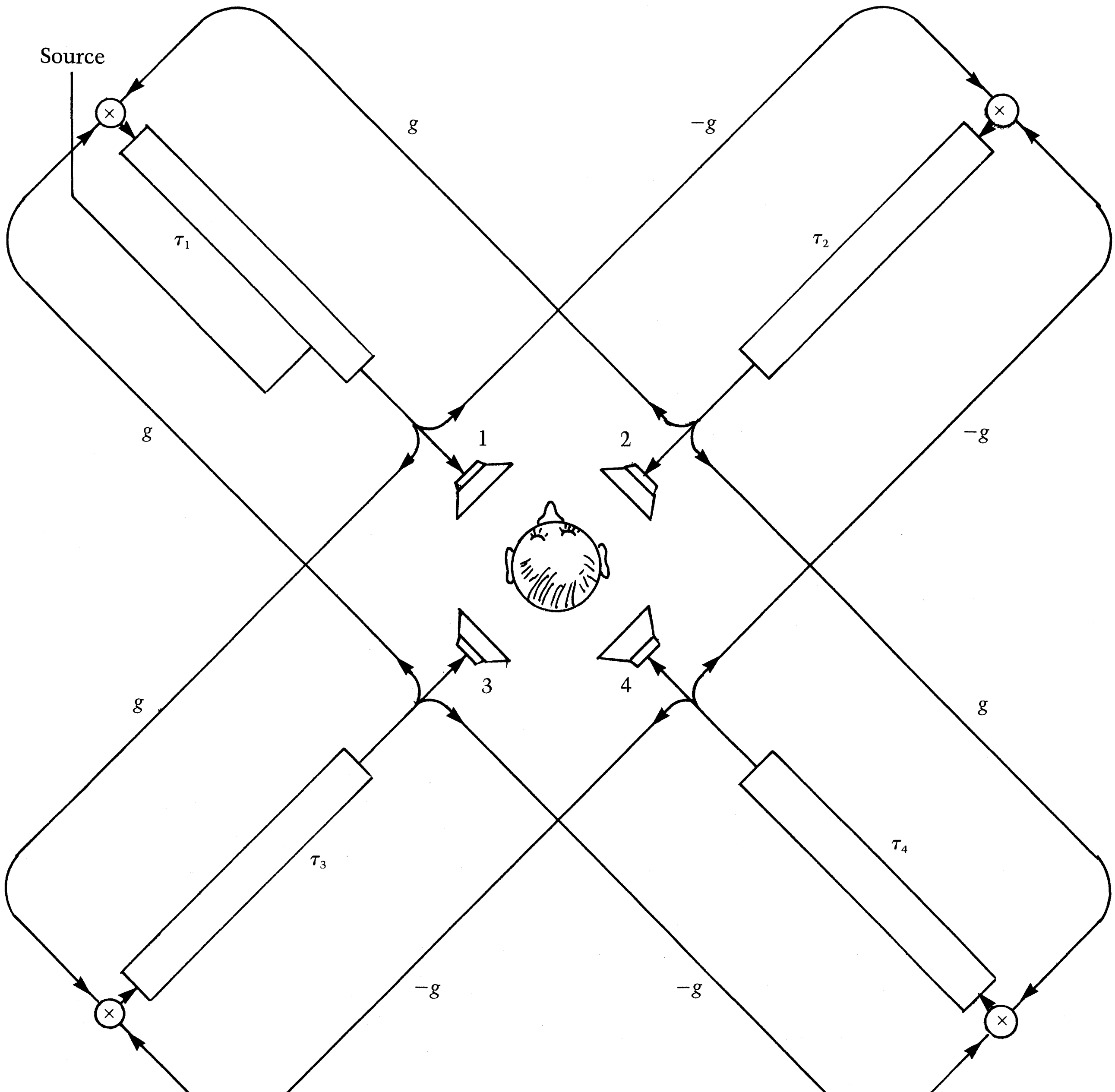
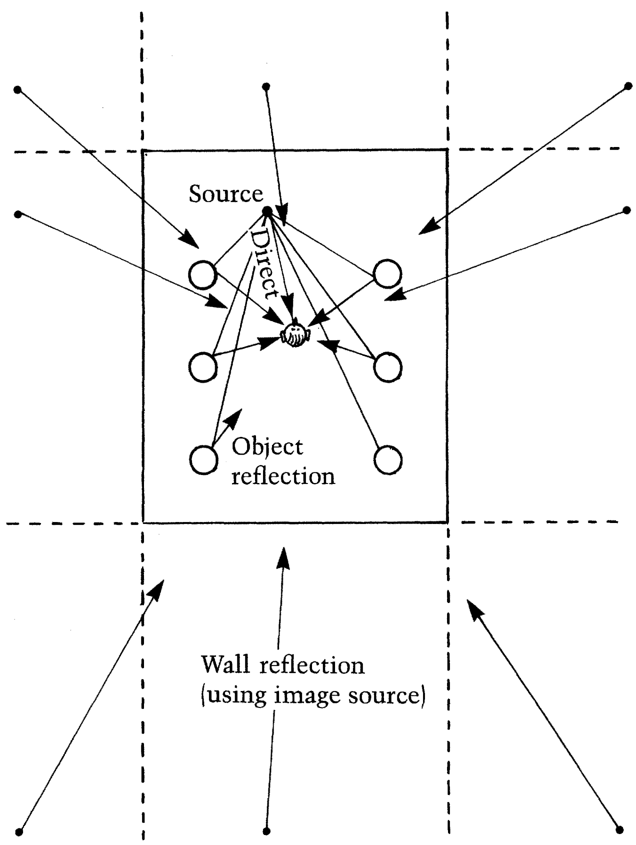


Fig. 4. *Various pathways of sound from source to listener. For objects, only the first reflections of the source signal from the objects are shown.*



characteristics, and incident angle of the reflections are important perceptual cues of reverberation (Jordan 1969; Barron 1971; 1974). The design of the reverberation network can be extended to simulate the effect of these early reflections.

Ultimately, these results may be applied in the design of reverberation networks that can simulate the favorable properties of existing concert halls or even aid in the design of new halls. Our purposes here are not to carry out such a simulation, but rather to find some guidelines for designing an early response leading to a good sounding reverberator suitable for general use. In doing this, we consider a greatly simplified description of the behavior of sound in rooms.

To model the early response, we simply add various proportions of the source signal directly into the delay loops for each channel. The length of each delay determines the amount of the early response that can be simulated for that channel and direction.

Actual delays, amplitudes, and incident directions of the early reflections may be chosen by defining a room of particular dimensions and employing the method of image sources to determine the reflections from the walls (Moorer 1979). An enhancement of this method is to define objects in the room and consider reflections from them as well.

The pathways of the sound rays for such a room are shown in Fig. 4. The amplitude of the sound pressure emanating from a simple source (point source) drops as 1/*distance* when the distance is greater than a few wavelengths or so. Near field effects of actual sound radiators are generally very complicated, so when the distance from the source is less than a few wavelengths we can approximate the amplitude of the sound pressure as being virtually unchanged.

Using these rules, we can find the amplitude of the direct sound signal or of that ray traveling the direct path from the source to the listener. In general, the attenuation of a ray coming from an image source will be approximately

$$att = k^m / src\_to\_lis,$$

where k is the amplitude reflection coefficient of the walls, m is the number of walls the ray has traversed, and *src_to_lis* is the distance from the image source to the listener position. The corresponding delay is

$$del = src\_to\_lis / VEL,$$

where *VEL* is the speed of sound. For objects, we may consider only the first reflection of the sound source from the object in order to simplify the calculations. As a further simplification, we will treat all objects as either large and flat like walls or cylindrical and with a given diameter.

Reflections from the large flat objects may be treated as a single wall reflection, with the corresponding amplitude attenuation.

For the reflections from cylinders, we have the following approximations for distances greater than a few wavelengths from the cylinder:

$$att = q \cdot (1/src\_to\_obj) \cdot (1/\sqrt{obj\_to\_lis})$$

and

$$del = (src\_to\_obj + obj\_to\_lis)/VEL,$$

where q is some attenuation coefficient representing the overall reflection characteristics of the object, *src_to_obj* is the distance from the source to the object, and *obj_to_lis* is the distance from the object to the listener. Near field effects of sound scattering from cylinders are very complicated, and it is sufficient to approximate the amplitude attenuation when the distance from the object is less than a few wavelengths simply as

$$att = q \cdot (1/src\_to\_obj).$$

The choice of values for k and q in the above relationships is not critical but should be made reasonably. Typical values for k can be found by consulting an acoustics text (e.g., Beranek 1954); normally, the quantity given is the energy absorption coefficient. The amplitude attenuation k is related to the energy absorption coefficient by

$$k = (1 - absorption)^{1/2}.$$

Absorption coefficients normally vary with the frequency and type of material, but a typical value for plaster is 0.04 at 500 Hz, yielding

$$k = \sqrt{1 - 0.04} = 0.98.$$

Setting a value for q is less straightforward, since q is highly dependent on the size and shape of the object and also on its position relative to the source and listener positions. For now, we may arbitrarily estimate a value of about 0.2. A more detailed way of choosing this coefficient will be discussed later.

The speaker channels for the reflections are chosen simply to correspond to that quadrant around the listener in which the incident sound ray approaches. This in effect samples the space into four discrete directions.

By addition of these early reflections into the delay network, the echo density of the early response is increased substantially. Without this initial density of reflections, the early part of the simulated reverberation response would sound ragged, especially for impulsive sounds. On the other hand, if the early reflections are too dense, the resulting sound will be mushy. Furthermore, the directional relationships of the reflections resemble the reverberation onset in a real room and contribute to the sense of spaciousness.

Long-Term Response

The choice of delay lengths gives at least a sense of the size of the room being modeled. As a rough guide, the lengths determine the interval between successive echoes, which is described statistically for real rooms by the mean free path of sound in the room. Shorter delays give more coloration (as do smaller rooms), which can be objectionable for long reverberation times.

Choosing incommensurate delay lengths is important to avoid flutter and achieve a smooth response. Schroeder suggests using lengths spanning a ratio of 1 : 1.5 (Schroeder 1962). Our longest delay lengths were typically about 1/10 sec. If much longer delays are desired, it may be useful to add a flat network with short delays to improve echo density.

Another important part of the long-term response of room reverberation is the frequency-dependent effect of air absorption. To simulate this effect, we insert lowpass filters at the output of each delay. The cutoff frequencies of the filters depend on the length of each delay (Moorer 1979).

Additional Modifications

A richer and more realistic sounding reverberation can be obtained by simulating the frequency characteristics of the reflections from walls and objects. Most wall materials show the greatest energy absorption in the middle frequencies between 500 and 2000 Hz or higher. This effect can be conveniently simulated by using a lowpass or a bandstop filter.

Some typical absorption coefficients for hard surfaces given by Kuttruff (1973) indicate a gently rolling off lowpass-filter effect. To simulate this behavior, we calculate the corresponding amplitude attenuation coefficients and then normalize these values so the maximum amplitude is 1. An example of these values is shown in Table 1. A first-order IIR filter with its attenuation (*att*) at 2000 Hz

Table 1. Finding the filter characteristics to simulate wall reflections

Domain of measurement	Frequency (Hz)					
	125	250	500	1000	2000	4000
Energy absorption coefficients for hard surfaces (from Kuttruff 1973)	0.02	0.02	0.03	0.03	0.04	0.05
Resulting amplitude attentuation of reflected wave	0.99	0.99	0.985	0.985	0.98	0.975
Normalized values to model filter, using attenuation of 0.99	1	1	0.995	0.995	0.99	0.985

chosen to match that of the wall attenuation is sufficient to model this behavior. The normalizing factor found is then used as the amplitude coefficient k in the equations given previously. For image sources whose rays have traversed several walls, the filter attenuation at 2000 Hz can be set to $(att)^m$, and the amplitude coefficient becomes k^m as before.

The frequency behavior of reflections from cylinders may be summarized as follows. For sound of wavelengths larger than the diameter of the object, very little sound is reflected, and it is reflected in all directions. Sound of wavelengths smaller than the diameter tends to be reflected back toward the source from the object and shadowed on the side of the object away from the source.

We may conveniently separate this behavior into two cases: (A) a listener on same side of object as source and (B) a listener on the opposite side. These situations are labeled in Fig. 5. For our purposes, it is satisfactory to model these two situations using a highpass, first-order FIR filter and a second-order IIR bandpass filter respectively. In the backward-reflecting case, we choose the half-power point of the highpass filter at about

$$freq_{1/2\ \mathrm{power}}\ (\mathrm{Hz}) = 10 \cdot VEL/(\pi \cdot diameter)$$

and use an amplitude coefficient q of about

$$1/\sqrt{2\pi}.$$

In the forward-reflecting case, we choose a center frequency and bandwidth of the bandpass filter at about

$$center\_freq\ (\mathrm{Hz}) = VEL/(\pi \cdot diameter)$$

$$bandwidth\ (\mathrm{Hz}) = 0.3 \cdot center\_freq$$

and set q at about

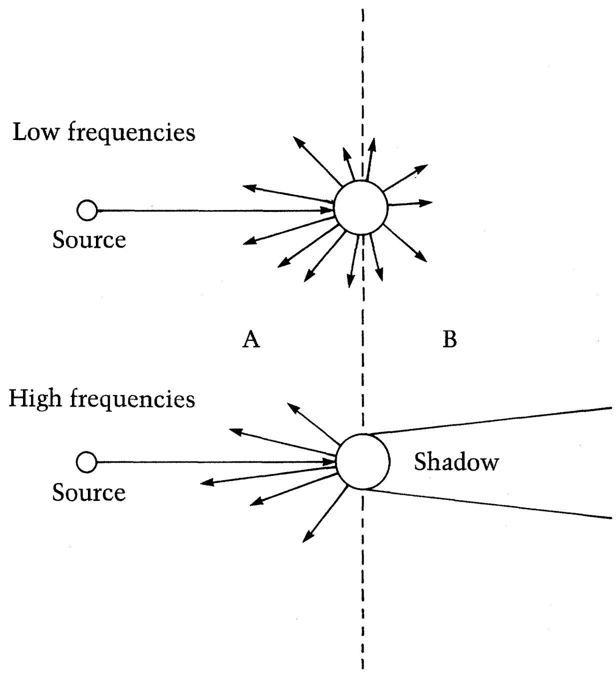
$$1/\sqrt{20\pi}.$$

We stress that these choices of filter properties are approximate and serve only to enrich the characteristics of the early response. The filters used here are quite standard and are described in great detail by, for example, Oppenheim and Schafer (1975). Further detailed descriptions of the scattering and absorption of sound may be pursued by referring to Morse's work (1948).

It may serve equally well to use some entirely different criteria in choosing these properties. It is advantageous, however, to create a systematic method that can easily be extended to allow the introduction of several sources into the simulated acoustic space, each with its own particular reflection relationships related to its simulated position in the space.

A further improvement can be gained by continuously varying the delay lengths in a random way. This has the effect of shifting the resonant peaks in the frequency response and decreases the possibility of flutter. Usually, a variation of 1 msec or less a few times per second is sufficient to enhance the quality of the sound without introducing

Fig. 5. Simplified diagram of scattering of sound from a cylinder at low and high frequencies. In case A, the listener is on the same side as the source. In case B, the listener is on the opposite side.



disturbing side effects. The reverberator design now stands as shown in Fig. 6.

Notes on Implementation

A program was written to design automatically a Music 11 reverberator using the preceding design methodology. The ROOM program takes as its input the dimensions of a rectangular room, coordinates and dimensions of objects in the room, a feedback matrix, a list of delay values, and a list of source locations. The output consists of Music 11 code to be used as part of a Music 11 orchestra file. An example of a reverberator with a single source in the left front channel implemented in this way is described in the section on sample Music 11 implementation.

More efficient use of memory space is obtained by avoiding the use of a delay for the source and reducing all delays of early reflections by a corresponding amount. This also has the effect of increasing the time window in which the early reflections can be calculated.

The reverberator design contains several optional branches around the statements that filter the early reflections and adds them into the delay. These branches become effective when the input source is not active, thereby eliminating unnecessary computation. This becomes very important in designing reverberators that allow several separate input sources.

A reverberator capable of handling several sources is a simple generalization of the single-source method. In our implementation, the ROOM program defines a set of early reflection characteristics for each source, corresponding to its simulated location. In practice, the calculation of early reflections in using a reverberator with many sources can become computationally intensive, but each additional source generally requires only a fractional amount of memory space compared to the amount of memory occupied by the delays. Furthermore, the use of conditional branches to avoid unnecessary computation, as described earlier, can greatly ease the matter.

In considering the various filter characteristics, we have assumed that filters of unity maximum gain are used in conjunction with an amplitude attenuation of the input signal. Further computational efficiencies can be achieved by scaling the filter coefficients by this attenuation value, thereby avoiding computationally costly multiplication of the signal prior to filtering.

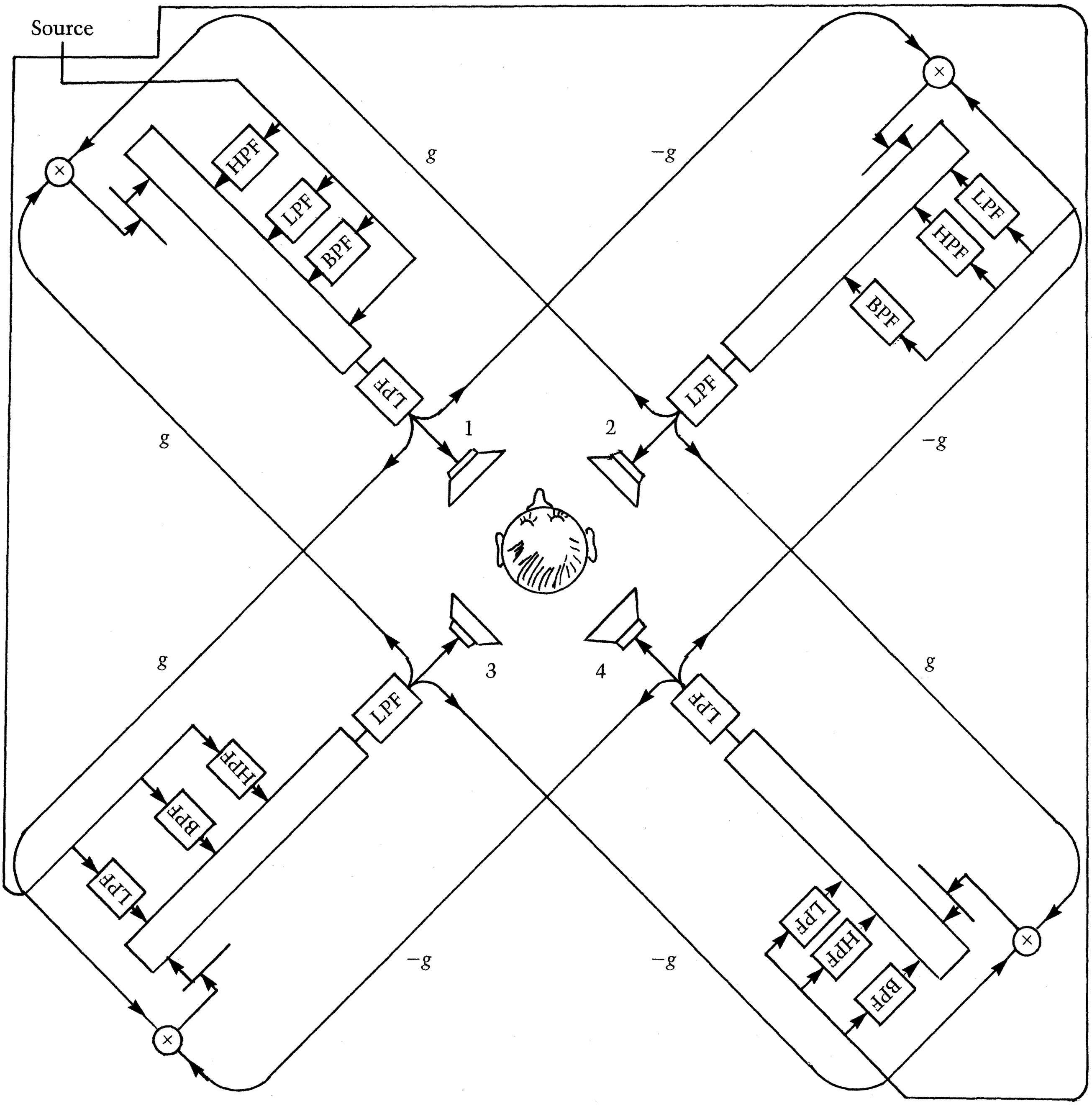
Observations and Suggestions for Further Work

A long list of possible improvements springs to mind. We have only gone into detail about one possible network, and new possibilities are born with each unitary matrix (of which there are plenty for everybody).

There is still no easy way to gauge overall coloration in a complex network, and trial and error is still required to control it. Coloration is probably less for networks containing at least some long delays; but incorporating them makes it harder to control echo density.

Another problem arising for long reverberation times and pure tone bursts is a fluctuation in am-

Fig. 6. The complete design of the reverberator. HPF = highpass filter; LPF = lowpass filter; BPF = bandpass filter.



plitude during the decay. This phenomenon is caused by the "excitation" of a small number of poles in the frequency response that create their own tones and beat against one another. This is a problem for every reverberator we have tried, and the only solution we can suggest is to arrange for the pole density to be so great that the beating is slower than the decay (i.e., for a reverberation time of 2 sec we need a pole density of two per Hertz). However, we are unable to think of any delay network resulting in a pole separation smaller in Hertz than the reciprocal of our total delay time in seconds. We strongly suspect that this is a fundamental limitation for reverberators. If that amount of space is not available, the best solution is to avoid using long reverberation times when the input is a pure tone burst.

A next step in the reverberation design procedure will be to allow for moving sources. This implies making extensions to the ROOM program in order to make it generate code that specifies the time-varying relationships of the early reflections as the source moves between two positions.

Further experiments using more than four channels would be worthwhile and could lead to an enhanced sense of acoustic ambience and greater realism. A study of the directional growth of reverberation in real rooms may help in choosing delay matrices for networks with many channels.

We also found that variation of the early response constraints, such as filter criteria, led to different qualities of reverberation, but we have no quantitative way of describing these relationships in perceptually meaningful terms. Some important research attacking these questions appears in the literature (Beranek 1962; Reichardt and Schmidt 1967; Barron 1971; 1974), and we would be interested in exploring simpler and more direct ways of defining the early response. Perhaps a statistical method of choosing the early reflection properties, combined with some perceptually meaningful constraints, can be found. Such a method may also lead to greater efficiency in computing the early response.

Conclusion

A methodology for the design and experimentation of reverberation networks has been presented. Some of the favorable properties of reverberators designed in this way are as follows:

1. The reverberant response is spatially sensitive to the location of the input source.
2. Several unique input sources may be added without significantly increasing the main memory space required.
3. A significant portion of the early room response characteristics may be modeled directly.
4. When a source signal is dormant, there is an increase in computational efficiency.

The methods we have presented are flexible enough to permit the design of stable recursive reverberators with an arbitrary number of output channels.

Acknowledgments

This work was supported by a grant from the M.I.T. Department of Humanities. We owe much to Barry Vercoe, whose initial insights led us to the present design, and to Roger Hale, who supplied us with additional ideas.

References

Barron, M. 1971. "The Subjective Effects of First Reflections in Concert Halls—the Need for Lateral Reflections." *Journal of Sound and Vibration* 15(4):475–494.

Barron, M. 1974. "The Effects of Early Reflections on Subjective Acoustic Quality in Concert Halls." Ph.D. thesis, Institute of Sound and Vibration Research, University of Southampton, England.

Beranek, L. 1954. *Acoustics*. New York: McGraw-Hill.

Beranek, L. 1962. *Music, Acoustics and Architecture*. New York: Wiley.

Jordan, V. L. 1969. "Room Acoustics and Architectural Acoustics Development in Recent Years." *Applied Acoustics* 2(1):59–81.

Kuttruff, H. 1973. *Room Acoustics*. New York: Wiley.

Meyer, E., W. Burgtorf, and P. Damaske. 1965. "Eine Ap-

paratur zur Electroakustischen Nachbildung von Schallfeldern, Subjektive Hörwirkungen beim Übergang Koharenz-Inkoharenz." *Acustica* 15(5): 339–344.
Moorer, J. 1979. "About This Reverberation Business." *Computer Music Journal* 3(2): 13–28.
Morse, P. M. 1948. *Vibration and Sound.* New York: McGraw-Hill.
Oppenheim, A., and R. Schafer. 1975. *Digital Signal Processing.* Englewood Cliffs, New Jersey: Prentice-Hall.
Reichardt, W., and W. Schmidt. 1967. "Die Wahrnehmbarkeit der Veranderung von Schallfeldparametern bei der Darbietung von Musik." *Acustica* 18(5): 274–282.
Schroeder, M. R. 1962. "Natural Sounding Artificial Reverberation." *Journal of the Audio Engineering Society* 10(3): 219–223.
Schroeder, M. R., D. Gottlob, and K. F. Siebrasse. 1974. "Comparative Study of European Concert Halls: Correlation of Subjective Preference with Geometric and Acoustic Parameters." *Journal of the Acoustical Society of America* 56(4): 1195–1201.
Strang, G. 1976. *Linear Algebra and Its Applications.* New York: Academic.

Appendix—Sample Music 11 Implementation

An example of a Music 11 reverberator designed with the aid of the ROOM program will be preceded here by a brief description of the unit generators used.

The Music 11 sound synthesis language computes samples of sound at two rates: the *control rate* and the *audio rate.* Typically, quantities that vary relatively slowly, such as amplitude envelopes or variable delay lengths, are calculated at the control rate, while actual waveform generation occurs at the faster audio rate. Variables updated at the control rate are represented as *kN* or *lkN*, where *N* is an integer and *l* indicates a variable local to the particular instrument. Similarly, audio rate quantities are designated by *aN* or *laN*. Constants are represented by *iN*. Global variables are values that can be passed between instruments and are designated by *gkN* for the control rate and *gaN* for the audio rate.

A delay space of the desired length in seconds is defined in the code by the **pipdef** statement. Any number of audio rate signals can be added into the delay using **pipad** statements up to the following **piprd** statement, which marks the end of the delay space and reads the audio rate samples from the delay. By using a **pipadv** statement, audio rate samples may be added into the delay space at a time-varying delay value determined by a control-rate variable.

Lowpass infinite-impulse response (IIR) filters are implemented using the **tone** unit generator, which takes the desired half-power point as an argument. The **reson** generator implements a second-order recursive filter with the desired center frequency and bandwidth between upper and lower half-power points. Finite-impulse response (FIR) filters are implemented by first delaying the input signal by one sample using the **delay1** unit generator and then adding the combined weights of the original signal and delayed signal. (In Fig. 7, both the filter coefficients and the amplitude attenuation factor appear in the implementation of the FIR filter; these constants are actually multiplied only once, and their combined product is then used for the audio rate calculations.)

The control-rate interpolating, random-number generator **randi** produces random straight-line segments between positive or negative random points lying within the given amplitude and occurring at the given frequency. These values are used to offset the feedback delay lengths slightly.

In the example presented later, the input signal is assumed to be generated in a sound-producing module or instrument occurring before the reverberator and is represented by the global audio-rate signal *ga*1. The global control rate variable *gk*1 is used as a flag to indicate whether the instrument is active or not in order to determine when the branches avoiding the early reflection calculations become effective. At the bottom of the reverberator, *ga*1 is set to zero to avoid recycling the last sample in the event that the sound-producing module has turned off at the end of a note.

Fig. 7. Music 11 implementation of a four-channel reverberator with one source. Music 11 unit generators and statements are in bold type, and comments are denoted by a semicolon.

```
instr 13                                       ;Reverberation instrument
ga1     init    0                              ;Initialize source signal to 0
k1      randi   .001, 3.1, .06                 ;Interpolated random numbers to
k2      randi   .0011, 3.5, .9                 ;  jitter delay lengths. First arg
k3      randi   .0017, 1.11, .7                ;  is max. range in ms, then cycle
k4      randi   .0006, 3.973, .3               ;  rate in Hz, then a seed.
la11    delay1  ga1                            ;Delay source by one sample for FIR filters
if gk1 = 0 kgoto nof11                         ;Skip unnecessary computation when source is off
a1      reson   ga1*.02286, 214.7, 64.411, 1   ;Early reflections for channel 1
a2      reson   ga1*.036248, 365, 109.5, 1     ;Bandpass filters
a3      reson   ga1*.027143, 219, 65.699, 1
a4      reson   ga1*.019974, 219, 65.699, 1
a5      tone    ga1*.070542, 9800              ;Lowpass filters
a6      tone    ga1*.061021, 10000
a7      tone    ga1*.05216, 9800
a8      tone    ga1*.046154, 9800
nof11:
        pipdef  .0683                          ;Pipe delay space for channel 1
if gk1 = 0 kgoto noa11                         ;If source is off, don't bother
        pipad   a1, .0006                      ;  adding in early reflections
        pipad   a2, .0001351
        pipad   a3, .0031383
        pipad   a4, .0046684                   ;Early reflections added into pipe
        pipad   a5, .0094834
        pipad   a6, .010068
        pipad   a7, .023716
        pipad   a8, .029589
noa11:                                         ;Feedback signal from channel 2 and 3 added
        pipadv  la2 + la3, .0663 + k1          ;  into time-varying delay.
la11    piprd                                  ;Read pipe output, channel 1.
if gk1 = 0 kgoto nof21                         ;Branch if source turned off
a1      =       ga1*.1014*.84379 - la11*.1014*.15621        ;Early reflections,
a2      =       ga1*.071322*.85075 - la11*.071322*.14925    ;  channel 2
a3      =       ga1*.055855*.85028 - la11*.055855*.14972
a4      tone    ga1*.03917, 10000
a5      =       ga1*.048756*.85015 - la11*.048756*.14985    ;Highpass FIR filter
a6      tone    ga1*.033955, 9800
a7      tone    ga1*.029319, 9800
a8      tone    ga1*.029319, 9800
a9      tone    ga1*.025974, 9800
nof21:
        pipdef  .0773                          ;Pipe delay, channel 2
if gk1 = 0 kgoto noa21                         ;Branch if source is dormant
        pipad   a1, .014705
        pipad   a2, .030699
        pipad   a3, .032913
        pipad   a4, .035149
        pipad   a5, .03897
        pipad   a6, .052999
```

```
         pipad  a7, .064321
         pipad  a8, .064321
         pipad  a9, .074434
noa21:
         pipadv -la1 - la4, .0753 + k2          ;Feedback from channels 1 and 4
la12     piprd                                  ;Read pipe, channel 2
if gk1 = 0 kgoto nof31                          ;Branch if source dormant
a1       reson  ga1*.024515, 210.57, 63.172, 1
a2       =      ga1*.08472*.84379 - la11*.08472*.15621
a3       =      ga1*.065767*.85015 - la11*.065767*.14985
a4       =      ga1*.055937*.85015 - la11*.055937*.14985
a5       tone   ga1*.034039, 10000
a6       =      ga1*.043571*.84647 - la11*.043571*.15353
a7       tone   ga1*.03061, 9800
a8       tone   ga1*.028892, 9800
nof31:
         pipdef .0902                           ;Pipe delay space, channel 3
if gk1 = 0 kgoto noa31                          ;Branch if source dormant
         pipad  a1, .010752
         pipad  a2, .018758
         pipad  a3, .022562
         pipad  a4, .03701
         pipad  a5, .045961
         pipad  a6, .05096
         pipad  a7, .062167
         pipad  a8, .065731
noa31:
         pipadv la1 - la4, .0882 + k3           ;Feedback from channels 1 and 4
la13     piprd                                  ;Read pipe output, channel 3
if gk1 = 0 kgoto nof41                          ;Branch if source dormant
a1       =      ga1*.080287*.84379 - la11*.080287*.15621
a2       =      ga1*.071111*.85041 - la11*.071111*.14959  ;Early reflections,
a3       =      ga1*.057238*.85015 - la11*.057238*.14985  ;  channel 4
a4       =      ga1*.048448*.84985 - la11*.048448*.15015
a5       tone   ga1*.030297, 10000
a6       =      ga1*.042724*.84647 - la11*.042724*.15353
a7       tone   ga1*.022991, 9800
a8       tone   ga1*.021125, 9800
nof41:
         pipdef .0991                           ;Pipe delay space, channel 4
if gk1 = 0 kgoto noa41                          ;Branch if source dormant
         pipad  a1, .028855
         pipad  a2, .033654
         pipad  a3, .041818
         pipad  a4, .053455
         pipad  a5, .057372
         pipad  a6, .05831
         pipad  a7, .090532
         pipad  a8, .098614
```

Fig. 7 (cont'd)

```
noa41:
        pipadv la2 - la3, .0971 + k4        ;Feedback from channels 2 and 3
la14    piprd                               ;Read pipe output, channel 4
i1      =       10.6301                     ;Distance of source from listener in meters
outq    la11 + ga1/i1, la12, la13, la14     ;Four channel reverb output, with source
                                            ;  in channel 1
la1     tone    la11*p4, 9000               ;Lowpass filters to simulate air
la2     tone    la12*p4, 9000               ;  absorption before signals are
la3     tone    la13*p4, 9000               ;  fed back to pipes. p4 is feedback
la4     tone    la14*p4, 9000               ;  coefficient "g" which controls reverb time
ga1     =       0                           ;Zero source signal before getting new samples
endin
```