Log Sampling in Time and Frequency: Preliminary Theory and Application

3935 (K-1)

D. B. (Don) Keele, Jr.
DBK Associates, Elkhart, IN 46517, USA
Techron, Div. Crown International, Inc., Elkhart, IN 46517, USA
Audio Magazine, Hachette Filipacchi Magazines, Inc., New York, NY 10019, USA

**Presented at
the 97th Convention
1994 November 10–13
San Francisco**

This preprint has been reproduced from the author's advance manuscript, without editing, corrections or consideration by the Review Board. The AES takes no responsibility for the contents.

Additional preprints may be obtained by sending request and remittance to the Audio Engineering Society, 60 East 42nd St., New York, New York 10165-2520, USA.

AN AUDIO ENGINEERING SOCIETY PREPRINT

Log Sampling in Time and Frequency: Preliminary Theory and Application

D. B. (DON) KEELE, JR.

DBK Associates, Elkhart, IN 46517, USA
Techron, Div. Crown International, Inc., Elkhart, IN 46517, USA
Audio Magazine, Hachette Filipacchi Magazines, Inc., New York, NY 10019, USA

The impulse response of real-world physical systems decays faster at high frequencies than low frequencies. This is due to the approximate constant "Q " behavior of the resonators that often make up these systems. Considerable efficiency can be obtained in digitizing, storing, and manipulating impulse response data that has been sampled at a rate that starts high initially, and then falls inversely with time. This paper explores this concept, and develops an efficient log-log time-frequency transform that converts back and forth directly between log-spaced time samples and log-spaced frequency samples. A very efficient FIR convolver/equalizer configuration, with logarithmically-spaced time taps, is described.

0. INTRODUCTION

Digital signal processing (DSP) operates on signals that are discrete in time; i.e., the signal is defined at discrete times by a sequence of numbers [1, Chapter 2]. Because the real world usually supplies analog or continuous-time signals, these signals must be first converted to discrete-time or digital signals by the use of an analog to digital converter. In all but the most rare cases, the analog signals are converted to digital signals at *equal-spaced* values of time. The only restriction is that the analog signal must be bandlimited to a frequency which is less than or equal to one-half the sampling rate, to minimize aliasing distortion.

This paper considers a nonuniform sampling technique where continuous-time data is sampled at a high rate at the start of the data, and then falls inversely with time to a low sample rate at the end of the data. This forms a sequence whose sample times have *equal-percentage* values of time, a so called log-spaced sequence. This method is primarily applied to sampling of impulse responses and their manipulation. Real-world impulse responses have the characteristic that the high-frequency energy primarily exists at the start of the response, while low-frequency energy covers a wider time span.

According to [2, Chapter 2., "Nonuniform Sampling," F. Marvasti]:

> "One might argue that nonuniform sampling is the natural way for the discrete representation of a continuous time signal. For example, consider a non-stationary signal with high instantaneous frequency components in certain time intervals and low instantaneous frequency components in others. It is more efficient to sample the low-frequency regions at a lower rate than the high-frequency regions. This implies that with fewer samples per interval, one might approximate a signal with appropriate nonuniform samples. In general, fewer samples mean data compression: i.e., it signifies less memory and processing time for a computer and faster transmission time and/or lower bandwidth for digital transmission."

In equal-spaced or uniform time sampling, a sequence of samples $x[n]$ is obtained from a continuous-time signal $x_c(t)$ according to the relation $x[n] = x_c(nT)$ where n is an integer in the range $-\infty < n < \infty$. In this paper, an alternate sampling scheme is presented where the samples are taken at equal-percentage times according to the relation $x[n] = x_c(T_{ref} R^n)$, where R is a number in the range $R > 1$, and T_{ref} is a reference time (the sample time when $n = 0$). Although n can range over all integers, it is restricted to values of zero or greater in this paper $n \geq 0$.

Fig. 1 illustrates linear and logarithmic sampling showing how the instantaneous sample rate changes over time. The dots indicate the individual sample points. Both linear and logarithmic scales are shown. The linear sampling starts at time zero with a constant sample rate of 50 kHz. The log sampling parameters follow the later Example 2 of Section 4.2.2, with an initial sample rate of 50 kHz, starting at roughly 77.3 us, with 10 points per decade (time ratio of one-tenth decade or about one-third octave $R \approx 1.259$), and falls thereafter following $f_S(t) \approx 3.86/t$.

In parallel with sampling in time, is sampling in frequency. Conventionally, most spectral and frequency response data is sampled at equal-spaced intervals of frequency. This is in contrast to the way that typical frequency-based data is almost always viewed, on a log frequency scale! It also makes sense to logarithmically sample frequency data. Log-sampled frequency data, although more common than log-sampled time data, is still generated quite infrequently. This paper in addition to considering log sampling in time, also formalizes log sampling in frequency, including a conversion method that directly converts log-sampled time data to log-sampled frequency data.

Log-spaced sampling is included in the much wider research area of nonuniform sampling. The field of nonuniform sampling is an area undergoing much research [3]. The reader is referred to the excellent book [2, with emphasis on Chapter 4]] for more information including a comprehensive list of references. The log-periodic antenna structure is the result of log-spaced design methods [4, Chapter 14][5, Chapter 9]. To the best of my knowledge, no previous work has been published on log-spaced sampling in a digital signal processing context.

0.1. Conventional Linear-Spaced Sampling and Processing

A large body of research and knowledge has grown up based on the theory and application of digital signal processing of signals that are defined at *equal* intervals of time [6-7]. In audio, one of the most common operations performed by digital signal processing is the operation of filtering and convolution. DSP systems that perform these operations may be used to implement such common audio devices such as filters, equalizers, crossovers, reverberators, room auralization processors, to name a few.

All these common DSP-based processing devices perform a discrete convolution between the input data stream and an internally-stored discrete representation of the impulse response of the system being implemented. This impulse response is almost always expressed as a series of equally-spaced samples of the continuous impulse response. The amount of storage required to store the impulse response (or equivalently the number of taps on the delay line convolver used to implement the convolver) rises in direct proportions to the sample frequency and to the length of the impulse response.

For a device such as a room auralization processor, which may require full-bandwidth operation for a very long impulse response, the storage space (and hence convolver size) can be considerable. For example, a room processor used to simulate a large room with a 20 second reverberation time with a 50 kHz sample rate, may require more than 2 megabytes of storage for stereo operation (2 x 50k samples/sec x 20 seconds)! The processing required to accomplish this convolution in real time is considerable. In practical auralization processors, techniques such as multi-rate processing and limitation of the length of the impulse response ease the implementation task [8].

0.2. Logarithmic-Spaced Sampling

This paper considers an alternate scheme of storage and manipulation of impulse data which is more closely matched to the characteristics of typical impulses. Rather than sampling the impulse response data at equal intervals of time, a logarithmic-spaced sampling scheme is considered, where the sample rate is high initially and then falls inversely with time.

In log-spaced sampling, the samples occur at times which form a geometric sequence, where the *ratio* between one sample time and the next is *constant*. In conventional linear-spaced or uniform sampling, the samples occur at times which form an arithmetic sequence, where the *difference* between two consecutive sample times is *constant*. Even though linear-spaced sampling may or may not be periodic (a equal-spaced sequence is periodic if $x[n] = x[n+N]$ for all n), log-spaced sampling is inherently *not* periodic (this is why I choose not to call log-spaced sampling "log-periodic" sampling). Even if a finite-length log-spaced sequence is made periodic by repeating the sequence end to end (called periodic nonuniform sampling) no advantages result.

The impulse response of real-world physical systems usually decays much faster at high frequencies than low frequencies. This is due to the approximate constant "Q" behavior of the resonators that often make these systems. For a specific Q, a high-frequency resonator's energy decays much quicker than a low-frequency resonator. For a specific decay threshold (say 60 dB down), the ring-down time (or reverberation time) of a resonator is directly proportional to its Q and inversely proportional to its resonant frequency. This means that for a multi-resonant wide bandwidth system containing many resonators, all of which have the same approximate Q, the effective upper-bandwidth of the frequency content of the system's impulse response, decreases as time increases.

For systems that meet this criteria (this includes all causal minimum-phase low-pass and band-pass systems), considerable efficiency can be obtained in digitizing, storing, and manipulating the impulse response data, if the data is sampled *logarithmically* in time, rather than linearly in time. This means that the impulse response is sampled at a high initial rate and then sampled less frequently as time progresses, the sample rate falling inversely with time.

The impulse response, rather than being sampled at constant time intervals, is instead sampled at intervals that increase at a constant percentage value from each sample to the next, i.e. assuming a times two multiplier, if the first sample is at 10 us, the following samples would be taken respectively at 20, 40, 80, 160 us .....etc. The sample time multiplier need not be an integer value as it is here (x2), but can be any positive real value greater than one, i.e., if the multiplier is 1.01 the samples would occur at 10, 10.1, 10.201, 10.303 us ..... etc.

This paper examines this logarithmic sampling scheme in some detail and develops some preliminary theory and design techniques, including an efficient Fourier transform method that converts back and forth directly between log-spaced time samples and log-spaced frequency samples, and applies log sampling to the design of finite impulse response (FIR) convolvers.

1. AN EXAMPLE: DISPLAY OF IMPULSE RESPONSE PLOTTED VERSUS LINEAR TIME AND LOG TIME

The original idea for the log sampling scheme described in this paper occurred when I was observing some impulse data from a loudspeaker using a graphing program on my computer. Normally, impulse responses are always plotted versus linear time, on a scale that starts from zero and goes to some maximum value. Speaker impulse responses usually exhibit a lot of high-frequency activity at the start of the response, while the low-frequency output is spread out over the whole response.

Plotting the data on a linear time scale means that if you want to observe all the significant features of the response, you have to display the impulse response on several different time scales.

However, when displayed on a log time scale, which in effect expands the start of the response and compresses the end of the response, you can see all the significant features of the impulse response on one display! Computer graphing programs make it easy to plot data on a linear or log horizontal scale. Oscilloscopes do not normally allow you to observe data on a logarithmic time scale.

This section illustrates the display of impulse response data on linear and log time scales. The impulse data was generated by computer simulation of a multi-resonant band-pass system model.

1.1. Band-Pass Model

A wide-band minimum-phase band-pass system was formulated that provided a rich impulse response with significant ringing at 100 Hz and 1 kHz. This system was composed of cascaded high- and low-pass filters with added peak and shelf filters to modify the response.

Fig. 2 shows the block diagram of the multi-resonant band-pass system composed of the cascade connection of: 1) a tenth-order Butterworth high-pass filter at 100 Hz, 2) a 100 Hz $Q = 8$ second-order peak filter with a gain of 18 dB at its peak, 3) a 1 kHz $Q = 16$ second-order peak filter with a gain of 12 dB at its peak, 4) a second-order low-frequency shelf filter providing attenuation of 20 dB above 2 kHz with unity gain at very-low frequencies (a 200 Hz normalized transfer function of $H(s) = \dfrac{s^2 + 2s + 10}{10(s^2 + 2s + 1)}$ where $s = j\omega$), and 5) a tenth-order Butterworth low-pass filter at 1 kHz.

Fig. 3 shows the frequency response of the model plotted over a two-decade 20 Hz to 20 kHz log frequency scale. The response rolls off very rapidly at 60 dB/octave at high and low frequencies and exhibits peaks at 100 Hz and 1 kHz.

1.2. Impulse Response Displayed on a Linear Time Scale

Fig. 4 shows the impulse response of the model plotted over linear time scales of 0 to 0.2 s (top graph) and an expanded (x10) linear time scale of 0 to 0.02 s (bottom graph). Note that the longer 0.2 s time-span graph (top) clearly shows the 100 Hz low-frequency ring down, but does not clearly show the 1 kHz information on the left side of the graph. Conversely, the x10 expanded time scale (bottom graph) clearly shows the high-frequency 1-kHz ring down, but not the details of the much longer low-frequency 100-Hz ring down.

1.3. Impulse Response Displayed on a Log Time Scale

Fig. 5 shows the impulse response of the model plotted on a logarithmic time scale. The log time scale runs over a nearly 3-decade 400 us to 0.2 second range. Here, all the details of the impulse response are clearly shown including the high frequency ring down at short times, and the low-frequency ring-down at long times! It was this display of impulse response, plotted on a log time scale, that was the impetus for the research described in this paper.

2. SAMPLING AND RECONSTRUCTION IN THE TIME DOMAIN

The sampling and reconstruction of signals in the time domain is fundamental to the operation of DSP-based processors. A major application of discrete-time processing systems is in the processing of continuous-time signals. Most real-world processing devices must operate on continuous or analog signals, i.e., the input and output signals of the processor are continuous. Internally, the continuous input signal must be converted to a discrete-time signal by sampling, then be processed in the digital or discrete-time domain, and then finally converted back to a continuous signal by a reconstruction technique.

Fig. 6 shows a general block diagram of an system that accomplishes discrete-time processing of continuous-time signals [1, p. 91]. The system is a cascade of a continuous to discrete (C/D) converter followed by a discrete-time processing system followed by a discrete to continuous (D/C) converter. According to [1, p. 92],

> "The block diagram .... represents a large class of systems since the sampling rate and the discrete-time system can be chosen as we wish. Note that the overall system is equivalent to a continuous-time system since it transforms the continuous-time input signal $x_{\text{in}}(t)$ into the continuous-time output signal $y_{\text{out}}(t)$."

In Fig. 6, the input continuous-time signal is first sampled, then processed by a discrete-time processing system, and then converted back to a continuous-time signal by a reconstruction process.

2.1. Linear-Spaced Sampling in Time: A Review

The sampling, processing, and reconstruction operations outlined in the block diagram of Fig. 6 are almost always done on data which is sampled at equally-spaced values of time. The following two sections detail the operations that typically occur in the C/D and D/C blocks in Fig. 6.

2.1.1. Conventional Linear-Spaced Sampling

Fig. 7 shows the internal details of the ideal continuous-time to discrete-time C/D conversion block of Fig. 6, along with the impulse response of the ideal low-pass filter which serves as an anti-alias filter. Note that I have explicitly included the anti-alias filter as a part of the C/D conversion process.

The continuous-time input signal is first bandlimited to one-half the sample rate with an ideal perfect brick-wall low-pass filter, then sent through an impulse train modulator, and then converted from impulses to a sequence of numbers. Note that the C/D block is a representation of the mathematics of ideal sampling, and not a representation of actual physical circuits that might be used to approximate it. In the real-world, a high-rolloff filter and/or oversampling techniques may be used along with an analog-to-digital converter to implement the C/D converter.

Note that the impulse response of the ideal low-pass filter is given by the sinc function [1, p.88]:

$$h_{\text{LP}}(t) = \text{sinc}\left({}^{t}\!/\!_{T} \right). \tag{1}$$

where

T = sampling period

$$\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}.$$

This function is unity at $t = 0$ and has zero crossings at every integer multiple of T other than zero, i.e. $t = \pm nT$, $n \neq 0$ (all other sample times). This function is shown later in Fig. 9(a).

2.1.2. An Alternate View of Linear-Spaced Sampling: Decomposition into Shifted Sinc Functions

An alternate way to view the operation of the ideal C/D sampling operation of Fig. 7 is to view it as the decomposition of the input continuous-time signal into a series of shifted sinc functions. Of course, the amount of shift is equal to an integral multiple of the sample time period (shift = $\pm nT$).

The decomposition appears as:

$$x[n] = \int_{-\infty}^{+\infty} f(t)\ \mathrm{sinc}(t - nT)dt \qquad \text{for } n \text{ an integer} \tag{2}$$

where

$x[n]$ = sequence of numbers representing sample amplitudes at each instant of time, for linear-spaced samples
n = sample count
$f(t)$ = continuous-time function to be sampled
T = sampling period (= $1/f_S$)
f_S = sampling frequency in Hz.

Note that this decomposition works because the sinc function forms an orthogonal system for unit shifts:

$$\int_{-\infty}^{+\infty} \mathrm{sinc}(t - j)\mathrm{sinc}(t - k)dt = \begin{cases} 1 & \text{for } j = k \\ 0 & \text{for } j \neq k \end{cases} \quad . \tag{3}$$

where

j, k = integers.

This alternate view of the sampling operation can be thought of as a combined operation that includes the low-pass filtering and sampling in one operation. As will be shown in the next section, the decomposition operation of Eq. (2) is the inverse of the reconstruction operation Eq. (4).

2.1.3. Reconstruction of a Linear-Spaced Signal from Its Samples

Fig. 8 shows the internal details of the ideal discrete-time to continuous-time D/C conversion block of Fig. 6. The sequence of sample numbers are first converted to a sequence of impulses at the sample rate, and then passed through an ideal reconstruction filter, which is identical to the perfect brick-wall low-pass filter of Fig. 7. This reconstruction filter has the exact same frequency response and impulse response as that shown for the antialias filter of Fig. 7.

The reconstruction process for a linear-spaced sample sequence may be written as:

$$x_r(t) = \sum_{n=-\infty}^{+\infty} x[n]\ \mathrm{sinc}[(t - nT)/T] \tag{4}$$

where

$x_r(t)$ = reconstructed continuous-time signal
$x[n]$ = sequence of numbers representing sample amplitudes at each instant of time, for linear-spaced samples
n = sample count integer
T = sampling period (= $1/f_S$)
f_S = sampling frequency in Hz.

This is recognized as the inverse process of the decomposition process described earlier in Eq. (2). The sampling process of Eq. (2) decomposes the continuous-time input into a discrete-time signal which is a series of weights or samples of the shifted sinc functions. The reconstruction process of Eq. (4) multiplies the weights or samples by each of the individual shifted sinc functions and sums each product to reconstruct the continuous-time output.

In this situation, the sinc function acts as a sampling or interpolating function that serves to fill in the data between the samples. A sample or interpolation function by definition is unity at the sample time and zero at all other sample times [9, p. 98 in chapter on "Polynomial Splines and Wavelets- A Signal Processing Perspective," by M. Unser, A. Aldroubi, an excellent book!].

2.1.4. Efficient Sample Functions: The Cardinal Cubic-Spline Interpolator and Hannsinc Interpolator Functions

For practical numerical computations, the use of the sinc function as a sample or interpolation function is quite inefficient. The sinc function decreases very slowly on either side of its peak. It takes over ten thousand zero crossings to decay 90 dB! In addition, there is no efficient way to compute its values quickly. Furthermore, the sinc function is not bounded in time, it continues for ever. Mathematicians describe this characteristic as not having "compact support."

Two other functions may be used for practical interpolation: a cubic-spline based interpolator function, and the Hannsinc which is a Hann-windowed sinc function. The first function rolls off much faster than the sinc, but is not bounded, while the second also rolls off much faster and is bounded.

Note that neither of these two functions are orthogonal in the sense of Eq. (3), however, they are near orthogonal because their product rapidly goes to zero as they are shifted farther and farther apart. They both may be used for decomposition and reconstruction operations instead of the sinc function in Eqs. (2) and (4).

The Cardinal Cubic-Spline Interpolator Function

A much more efficient sample function is the cardinal cubic-spline interpolator $\eta^3(x)$ described later in Appendix 2 [9, pp. 95-97]. This function looks superficially like a sinc function but decays much more rapidly. It only takes about nine zero crossings to decay below 90 dB, and is quite efficient for machine computation because it is based on simple spline polynomials [10]. A bounded (truncated after 9 zero crossings, total width 18) version of this function was used to make most of the decompositions and reconstructions in this paper.

The Hannsinc Interpolator Function

If a Hann or cosine-squared window is applied to the sinc function, the Hannsinc interpolation function results. The width of the Hann window can be independently set to govern the support range of the Hannsinc. The Hannsinc function appears as

$$\mathrm{Hannsinc}\,(x,W)=\begin{cases}\dfrac{1}{2}\left[1+\cos\left(\dfrac{2\pi x}{W}\right)\right]\mathrm{sinc}(x) & \text{for } -\dfrac{W}{2}\le x\le\dfrac{W}{2}\\ 0 & \text{otherwise}\end{cases}\qquad(5)$$

where

W = width or support range of Hannsinc, W = 2, 4, 6, . . . even integers.

Effectively, a width of six or eight works well for the Hannsinc interpolator. Most of the visual reconstructions in this paper shown on graphs were done using the Hannsinc or the Log-warped version of the Hannsinc called the LogHannsinc (described later).

Fig. 9 shows comparisons of these three interpolator functions plotted with both linear and log vertical scales. The interpolation and sampling properties of the sinc, cardinal cubic-spline, and Hannsinc interpolators stems from the fact that they are precisely equal to unity at the origin and that they vanish at all other integer values.

2.2. Log-Spaced Sampling in Time: A New Theory

The previous linear-spaced sampling described in section 2.1 makes no assumptions about the continuous-time signal to be sampled other than it will be explicitly low-pass filtered to one half the sample frequency by the process of sampling and reconstruction. The signal to be sampled may exist over any time and may have short-term spectral components of any frequency which may exist at any time period.

In contrast, if the input signal is a typical real-world impulse response, the signal is restricted in several ways:

Firstly, the impulse response is assumed to be causal, which means the response exists only for positive times (technically it will also exist at zero time, but for purposes of this analysis it is assumed to exist only for positive times).

Secondly, the impulse response is assumed to be stable and of finite energy, which means the response decays to zero as time goes to infinity.

Thirdly, the frequency content of the impulse is assumed to die out in a very prescribed manner, i.e. the high frequencies decay faster than the low frequencies. This third restriction is based on the condition that real-world systems are composed of multiple resonators all of which have approximately the same Q or energy decay characteristics per cycle of decay. This means that the response can be thought of as having a constant-Q decay behavior. For the same Q and decay threshold, a high-frequency resonator decays faster than a low-frequency resonator.

If the high-frequency content of an impulse response dies out sooner than the low-frequency content, it is logical to sample the response at a higher rate at short times and less often at longer times. A sample rate which starts at a high initial rate and then falls inversely with time, fits this requirement. This will be shown as follows.

2.2.1. *Log-Spaced Sample Times*

This section develops a set of relationships that describe log-spaced sampling in the time domain.

Assume a sequence of samples whose sample times form a geometric sequence with a constant ratio R between one sample and the next (R >1). In general, this log-spaced sample sequence is infinite in extent ranging from (but not including) zero and going to infinity, and is by definition, not periodic. However, in this paper only finite sequences are considered, which start at some initial time $t_{\min}$, where $t_{\min} > 0$, and end at $t_{\max}$, where $t_{\max} > t_{\min}$. A sample at zero time ($t = 0$) is specifically prohibited.

For a geometric sequence of sample times, a recursion relation may be written relating one sample time to the previous sample time:

$$t_{n+1} = R t_n \qquad (6)$$

where

t_n = sample time at point n

R = multiplier or ratio relating one sample time to the previous sample time (a real number > 1)

n = sample count, = 0, 1, 2, . . . K

Eq. (6) may be written in a form that relates a particular sample time to the initial sample time, the sample count, and the multiplier:

$$t_n = t_{\min} R^n \qquad \text{for } n = 0, 1, 2, \ldots K \tag{7}$$

where

$t_{\min}$ = initial sample time (time of first sample, and also is the minimum sample time)
K = last count number (=N_{T} - 1)
N_{T} = total number of samples (=K+1).

Eqs. (6) and (7) describe a sampling process where the samples are not gathered at equal increments of time, but are gathered at intervals of time that increase at a constant percentage rate from one sample to the next. For example: assuming an initial sample time of 100 us and a multiplier of 1.1, the successive sample times are : 100, 110, 121, 133.1, 146.41, 161.051, etc.

Because this sampling process has a sampling period that changes over the elapsed time of the data gathering, the characteristics of the process are best described by a sample density per percentage range of time, i.e., 200 points per decade, or 3 points per octave, etc. Towards this end, Eq. (7) may be written in several exponential forms based on different number bases, i.e., base e, base 2, and base 10:

$$\begin{aligned} t_n &= t_{\min} R^n \\ &= t_{\min} e^{\frac{n}{N_e}} = t_{\min} 2^{\frac{n}{N_2}} = t_{\min} 10^{\frac{n}{N_{10}}} \end{aligned} \tag{8}$$

where

N_e = number points per e (ratio of e^1 = 2.7183..)
N_2 = number points per octave (ratio 2)
N_{10} = number points per decade (ratio 10).

With n=K, the time of the last sample is given by

$$\begin{aligned} t_K &= t_{\max} = t_{\min} R^K \\ &= t_{\min} e^{\frac{K}{N_e}} = t_{\min} 2^{\frac{K}{N_2}} = t_{\min} 10^{\frac{K}{N_{10}}}. \end{aligned} \tag{9}$$

The multiplier R can be written in terms of each of the number-points-per constants:

$$R = e^{\frac{1}{N_e}} = 2^{\frac{1}{N_2}} = 10^{\frac{1}{N_{10}}}. \tag{10}$$

A total time span ratio R_{T} can be defined, which is just the ratio between the end time and the start time:

$$
\begin{aligned}
R_{\mathrm{T}} &= \frac{t_{\max}}{t_{\min}} = R^{K} \\
&= e^{\frac{K}{N_e}} = 2^{\frac{K}{N_2}} = 10^{\frac{K}{N_{10}}}
\end{aligned} \tag{11}
$$

The total number of points N_{T} can be written in a number of ways that depend on the ratios and number-points-per values:

$$
\begin{aligned}
N_{\mathrm{T}} &= K + 1 = \frac{\log(R_{\mathrm{T}})}{\log(R)} + 1 \\
&= N_e \log_e(R_{\mathrm{T}}) + 1 = N_2 \log_2(R_{\mathrm{T}}) + 1 = N_{10} \log_{10}(R_{\mathrm{T}}) + 1 .
\end{aligned} \tag{12}
$$

The last forms of this equation show that the total number of points just depends on the sample density multiplied by the log of the total time span ratio plus one. This is recognized as being just the number points per decade multiplied by the time span in decades plus one, and similarly for other number systems.

All the equations for time-domain sampling are shown in one location in Table 1.

2.2.2. Log-Spaced Sample Rate

The effective instantaneous sample rate f_S, in samples per second, at each sample time (Eqs. (7) or (8)) can be derived by computing the reciprocal of the forward running time difference between each sample time and the next:

$$
\begin{aligned}
f_{S_n} &= \frac{1}{\Delta t_n} = \frac{1}{t_{n+1} - t_n} = \frac{1}{R t_n - t_n} \\
&= \frac{1}{(R - 1) t_n} \qquad \text{for } n = 0, 1, 2, \ldots K
\end{aligned} \tag{13}
$$

Eq. (13) may be converted to a form which approximately yields the sample rate as a function of continuous time with the substitution of $t_n = t$:

$$
f_S(t) \approx \frac{1}{(R-1)t} = \frac{\frac{1}{(R-1)}}{t} \tag{14}
$$

This form clearly illustrates that the sample rate falls inversely with time. It also shows that the smaller the ratio between one sample time and the next, the higher the sample rate at a specific time.

Eq. (13) may in turn be combined with Eq. (7) to yield the sample rate as a function of n:

$$f_{S_n} = \frac{1}{(R-1)t_{\min}R^n} \qquad \text{for } n = 0, 1, 2, \ldots K. \tag{15}$$

The initial or maximum sample frequency is just Eqs. (13) or (15) evaluated at $n = 0$:

$$f_{S0} = f_{S\max} = \frac{1}{(R-1)t_{\min}R^0} = \frac{1}{(R-1)t_{\min}}. \tag{16}$$

Eqs. (15) and (16) may be combined to yield an alternate expression for the instantaneous sample frequency:

$$f_{S_n} = \frac{f_{S\max}}{R^n} = f_{S\max}R^{-n} \qquad \text{for } n = 0, 1, 2, \ldots K. \tag{17}$$

All these equations, especially Eqs. (13), (14), and (17) clearly show that the effective sample rate decreases inversely with time, starting at a *maximum* at $t_{\min}$, and falling thereafter to a *minimum* at $t_{\max}$.

The ending or minimum sample rate is Eq. (17) evaluated at $n = K$:

$$f_{S_L} = f_{S\min} = \frac{f_{S\max}}{R^K} = f_{S\max}R^{-K}. \tag{18}$$

The sample rate may also be expressed in terms of the number-points-per constants:

$$\begin{aligned} f_{S_n} &= f_{S\max}R^{-n} \\ &= f_{S\max}e^{-\left(\frac{n}{N_e}\right)} = f_{S\max}2^{-\left(\frac{n}{N_2}\right)} = f_{S\max}10^{-\left(\frac{n}{N_{10}}\right)} \quad \text{for } n = 0, 1, 2, \ldots K. \end{aligned} \tag{19}$$

All these equations for log-sampling sample rate are shown in Table 2.

2.2.3. Log-Spaced Sampling Mechanics

For log sampling, the ideal C/D sampling operation of Fig. 7 can be thought of as an anti-aliasing filter whose corner frequency decreases inversely with time, followed by an impulse train modulator whose impulses occur at log times given by Eq. (7). The corner frequency of the anti-aliasing filter follows the sampling frequency, but at a fractional value assumed to be in the range of one-half (x0.5 = 1/2) to two fifths (x0.4 = 1/2.5), to minimize aliasing.

Rather than attempting to detail the mechanics of the log-spaced sampling process as described in the previous paragraph, I prefer to describe the log-spaced sampling process in an alternate manner which parallels the linear-spaced decomposition process described earlier in Eq. (2). This is done by defining a sampling function that is log warped, so that it operates in log time the same as the conventional sampling function operates in linear time. If the log-warped sample function is viewed on a log-time axis, it looks identical to the non-warped sampling function viewed on a linear-time axis.

A sampling function may be defined in log time by applying the following transformation of the time variable:

$$t' = N_x \log_x (t/t_c) \qquad (20)$$

where

t' = log time variable
t = linear time independent variable
t_c = center time of transformation (time in log scale that corresponds to zero time in linear time scale)
x = number base designator, i.e., $x = e$, 2, 10 etc.

This transformation maps the positive linear t axis $(0 > t > +\infty)$ onto the whole log t' axis $(-\infty > t' > +\infty)$, with times greater than t_c mapped to the positive t' axis and times less than t_c mapped to the negative t' axis. This transformation effectively expands the time axis for short times, and compresses the time axis for long times.

2.2.4. The Logsinc Sampling Function

An ideal sampling function for the log-time domain may be developed by applying the log transformation of Eq. (20) to the sinc function of Eq. (1), which is the ideal sampling function in the linear-time domain:

$$\log\operatorname{sinc}(t,t_c,N_x) = \operatorname{sinc}\left[N_x \log_x\left(\frac{t}{t_c}\right)\right] \qquad (21)$$

where

t_c = center time (time at which logsinc amplitude is unity)
N_x = number points per percentage x time (either e range, $x = e$; octave, $x = 2$; or decade, $x = 10$).

The logsinc function is unity at $t = t_c$ and has zero crossings at every percentage multiple above and below t_c, i.e., $\ldots \frac{t_c}{R^3}, \frac{t_c}{R^2}, \frac{t_c}{R}, Rt_c, R^2t_c, R^3t_c, \ldots$ etc., and only exists for $t > 0$. The logsinc acts as an interpolation function which is specifically designed to interpolate a sequence of samples whose sample times are arranged in a geometric sequence.

The logsinc function is shown in Fig. 10, assuming a sample density of one point per octave (multiplier $R = 2$), a center time of one second, and is plotted on both linear and log time axis. Note that when viewed on a log time scale, Figs. 10(b) and (c), the logsinc function looks exactly the same as a sinc function viewed on a linear time scale. When the logsinc is viewed on a linear time scale, Fig. 10(a), its oscillatory die down on either side of its maximum, are compressed as time decreases, and expanded as time increases, with its zero crossings exactly coinciding with the log-spaced sample points. Fig. 10(c) clearly shows that the zero crossings of the one-point-per-octave logsinc occur at multiples of two. Note that the logsinc is *inherently causal* because it *does not* exist below zero time.

Note that other log-warped sampling functions can be created using the cardinal cubic-spline function and Hannsinc functions of Sections 2.1.4. The log-warped Hannsinc function is called the LogHannsinc.

Shifting the logsinc up and down in log time simultaneously changes its width and its center time t_c, but not the sample density. Shifting the logsinc up in time increases its width and vice versa. Figs. 11(a) to (c) illustrates this behavior with a one-point-per-octave width-six LogHannsinc with several different values of center time, all plotted on log time scales.

Changes in sample density effect the width of the logsinc but not its center time. Figs. 12(a) to (c) illustrates this behavior with several LogHannsincs of different sample densities with the same center time, all plotted on log time scales.

2.2.5. Logsinc Frequency Spectrum

The logsinc's frequency spectrum is low-pass, but is not a brickwall function because its frequency content is spread out due to the log warping. Figs. 11(d) to (f) show the frequency spectrums of the LogHannsincs shown in Figs. 11(a) to (c), all plotted on log frequency scales. Note that shifting the logsinc (or LogHannsinc) up in log time by a certain ratio (towards larger log time values), shifts the spectrum down in log frequency by the same ratio, and vice versa.

Figs. 12(d) to (f) show spectrums of the LogHannsincs illustrated in Figs. 12(a) to (c), where the sample density is changed without changing the center time. Note that increasing the sample density sharpens the cutoff of the low-pass spectrum and shifts the corner higher in frequency.

2.2.6. Decomposition into Shifted Logsinc Functions

Mirroring the linear decomposition process described in Section 2.1.2 Eq. (2), the input continuous-time signal can be effectively sampled at log-spaced intervals by decomposing it into a series of shifted logsinc functions as follows:

$$x[n] = \int_{0^+}^{+\infty} f(t)\,\mathrm{logsinc}(t, t_{\min}R^n, N_X)\,dt \qquad n = 0, 1, 2, \ldots \qquad (22)$$

where

$x[n]$ = sequence of numbers representing sample amplitudes at each instant of time (for log-spaced samples)

$f(t)$ = continuous-time function to be sampled

$t_{\min}$ = initial time (time of first sample)

R = multiplier relating one sample time to the previous sample time (a real number > 1)

N_x = number points per percentage x time (either e range, $x = e$; octave, $x = 2$; or decade, $x = 10$; $N_x = \dfrac{1}{\log_x(R)}$).

The decomposition process given by Eq. (22) combines in one operation the variable-frequency low-pass filtering, and sampling at log-spaced time intervals described in the first paragraph of Section 2.2.3.

Interestingly, the logsinc function, is not orthogonal on a linear time scale for constant percentage shifts (as determined by numerical integration). However, when the orthogonality is evaluated on a log-time scale, it obviously is orthogonal. I do not know what impact this has on the decomposition-reconstruction process for log-spaced samples. However, the process appears to work quite well as determined by numerical computer model.

2.2.7. Reconstruction of a Log-Spaced Signal from Its Samples

The reconstruction of a signal from its log-spaced samples essentially follows the same procedure as described for linear-spaced samples, Sect. 2.1.3, except that the impulse response of the ideal reconstruction filter is a logsinc function rather than a sinc function.

The reconstruction process for a log-spaced sample sequence may be written as:

$$x_r(t) = \sum_{n=0}^{+\infty} x[n] \operatorname{logsinc}\left(t, t_{\min} R^n, N_x\right) \qquad n = 0, 1, 2, \ldots \qquad (23)$$

where

$x_r(t)$ = reconstructed continuous-time signal

$x[n]$ = sequence of numbers representing sample amplitudes at each instant of time (for log-spaced samples)

$t_{\min}$ = initial time (time of first sample)

R = multiplier relating one sample time to the previous sample time (a real number > 1)

N_x = number points per percentage x time (either e range, $x = e$; octave, $x = 2$; or decade, $x = 10$; $N_x = \dfrac{1}{\log_x(R)}$).

Paralleling the linear situation, the reconstruction process of Eq. (23) is the inverse of the decomposition process described in Eq. (22). The sampling process of Eq. (22) decomposes the continuous-time input into a discrete-time signal which is a series of weights or samples of the shifted logsinc functions. The reconstruction process of Eq. (23) multiplies the weights or samples by each of the individual shifted logsinc functions and sums each product to reconstruct the continuous-time output.

2.2.8. Required Sample Density Versus Resonator Q and Decay Threshold

The most significant descriptor for conventional linear-spaced or equal-spaced sampling is the sample frequency f_S, in samples per second (or equivalently the sample period T (= $1/f_S$)). Because the sampling frequency for log-spaced sampling changes over the elapsed time of the sampling process, the characteristics of the process are best described by a sample density per percentage range of time, i.e., 200 points per decade, or 3 points per octave, etc.

Inherent in the log-spaced sample process is fact that a specific frequency in a time signal is only sampled for a specific length of time, not over the whole signal's length. This means that in order to specify the log-spaced sample density, something must be known about how long a specific frequency in an impulse response takes to decay. This information is best determined from the decay characteristics of the resonators that make up the system that generates the impulse response. These decay characteristics are set by the resonator's center frequency, and Q. These parameters completely specify the decay rate of the resonator's impulse response, which is primarily composed of a ring down at the resonator's center frequency.

It can be shown that the time it takes for a second-order resonator to ring down to a specific threshold is given by (Appendix 1.):

$$T_d = \frac{L_{dB}}{20\pi \log_{10}(e)} \cdot \frac{Q}{f_0} \approx \frac{L_{dB}}{27.3} \cdot \frac{Q}{f_0} \tag{24}$$

where

T_d = ring-down or decay time of resonator, i.e., the time it takes the resonator to ring down from its maximum level, down to the threshold specified by L_{dB}

L_{dB} = threshold level in dB, defines cutoff point below which the resonator's decay is not significant

f_0 = center or resonant frequency of resonator in Hz

Q = resonator's energy-loss figure of merit.

For a specific Q, this equation shows that a resonator's ring-down depends inversely on its center frequency, i.e., a high-frequency resonator's impulse response decays much quicker than a low-frequency resonator.

Eq. (24) can be solved for the resonator's center frequency yielding

$$f_0 = \frac{L_{dB}}{20\pi \log_{10}(e)} \cdot \frac{Q}{T_d} \tag{25}$$

This equation shows how the resonator's center frequency depends on the ring-down time and the Q of the resonator. Short ring-down times are associated with high center frequencies, and long ring-downs with low frequencies, etc.

In order to sample a specific ring-down decay, it must be accurately sampled out to the time where its level drops below the threshold level. In other words, the sampling frequency must be high enough so that the whole ring-down is captured accurately out to its decay time, which is given by Eq. (24).

To properly digitize a sinusoidal time signal, it must be sampled at a rate at least twice as high as the frequency of the sinewave it contains. To be somewhat conservative, in this paper I will assume that the sample rate must be at least two and a half times (2.5) the highest frequency contained in the signal.

An expression relating the sample density (or equivalently the sample time multiplier) to the resonator's Q, can be developed by relating the expression for instantaneous sample frequency, Eq. (13), to the resonator's decay frequency, Eq. (25). The sampling ratio R (or equivalently the sample density) is chosen to make the instantaneous sample rate 2.5 times the resonator's center frequency at a time equal to its decay time, i.e., we want $f_{S_n} = 2.5 f_0$ at time T_d.This is be done as follows:

$$f_{S_n} = \frac{1}{(R-1)t_n} = 2.5 f_0 = \frac{2.5 L_{dB}}{20\pi \log_{10}(e)} \cdot \frac{Q}{T_d} \tag{26}$$

Of course, the sample time t_n and the decay time T_d are equal, and thus can be canceled, yielding:

$$\frac{1}{(R-1)} = \frac{2.5L_{dB}}{20\pi\log_{10}(e)} \cdot Q \tag{27}$$

This may in turn be solved for the multiplier ratio R giving:

$$\begin{aligned} R &= \frac{8\pi\log_{10}(e)}{L_{dB}Q} + 1 \\ &\approx \frac{10.92}{L_{dB}Q} + 1 \end{aligned} \tag{28}$$

Eq. (28) shows that the higher the Q the smaller the ratio R, and hence the higher the sample density required. Using the relations of Eq. (10), this expression may in turn be converted to a form containing the desired sample densities:

$$N_x = \frac{1}{\log_x\left(\frac{8\pi\log_{10}(e)}{L_{dB}Q} + 1\right)} \tag{29}$$

where

x = number base designator, i.e., $x = e$, 2, 10 etc.

This expression directly relates the required sample density to the Q of the resonant system and the decay threshold.

For example, assuming a base 10 system and a decay threshold of 60 dB, Eq. (29) simplifies to:

$$N_{10} \approx \frac{1}{\log_{10}\left(\frac{1}{5.5Q} + 1\right)} \tag{30}$$

For example, using this relationship, the impulse response of a system containing resonators whose individual Q's are no more than 20, can be accurately log sampled with a sample density of only 255 time points per decade. This means that the impulse response of a very-wide bandwidth four-decade system containing resonators at both 2 Hz and 20 kHz, could be accurately sampled with only roughly 1020 points (4 x 255).

Fig. 13 shows plots of Eq. (29) giving the required sample density, in points per decade, versus Q for various decay thresholds.

2.2.9. *Decomposition and Reconstruction Using Differenced Logsincs*

An alternate to log-spaced decomposition and reconstruction using shifted logsincs, described in Sections. 2.2.6 and 2.2.7, is an entirely equivalent procedure which uses differenced logsincs. As shown in Sec. 2.2.5 the logsinc function's spectrum is low pass, which means it has energy all the way down to DC. If the difference of two adjacent logsincs is formed however, the function is essentially bandpass, which means it has relatively low DC content. Note that the difference of two sincs (not logsincs!) is exactly bandpass with no DC content. The process of log warping distorts the differenced functions and adds some DC.

Rather than decompose the continuous-time signal in a series of weighted and shifted logsincs, the signal is decomposed into a single low-frequency low-pass logsinc and a series of higher-frequency differenced band-pass logsincs. Decomposition and reconstruction using differenced logsincs has the advantage that there is a one to one correspondence between the weight of a specific logsinc (or differenced logsinc) and the portion of the frequency spectrum it influences. This may ease the design of equalizers using log-spaced convolvers, because each tap-weight of the convolver effects only a specific portion of the frequency spectrum.

The relationship between the coefficients of the logsincs (just the log-spaced samples, $x[n]$ of Eq. (22)) and the new coefficients of the differenced logsincs is developed as follows (note that the symbols used here are different from those used elsewhere in this paper to clarify the development). Assume a series of coefficients a_n for standard logsinc expansion, and a series of coefficients d_n of the alternate expansion using differenced logsincs.

The standard reconstruction in a series of shifted logsincs appears in simplified form as follows (refer to Eq. (23) but assume a finite set of coefficients):

$$\begin{aligned} x_r(t) &= \sum_{n=0}^{K} a_n S_n \\ &= a_0 S_0 + a_1 S_1 + a_2 S_3 + \ldots + a_{K-1} S_{K-1} + a_K S_K \end{aligned} \qquad (31)$$

where

$x_r(t)$ = reconstructed continuous-time signal

a_n = log-spaced sample coefficients

S_n = logsinc(t) of appropriate location and width.

Note that logsinc S_0 is the narrowest and hence covers the widest bandwidth, while logsinc S_K is the widest and covers only the low frequencies.

Now assume an equivalent reconstruction in terms of adjacent differenced logsincs:

$$\begin{aligned} x_r(t) &= \sum_{n=0}^{K-1} d_n (S_n - S_{n+1}) + d_K S_K \\ &= d_0(S_0 - S_1) + d_1(S_1 - S_2) + \ldots + d_{K-1}(S_{K-1} - S_K) + d_K S_K \end{aligned} \qquad (32)$$

Note that in this expansion, the last S_K term is the low-pass logsinc covering the lowest frequencies, while the remaining terms are all band-pass differenced logsincs with the $(S_0 - S_1)$

differenced logsinc covering the highest frequency band, while the $(S_{K-1} - S_K)$ differenced logsinc covers the low-frequency band.

Eq. (32) can in turn can be expanded,

$$= d_0 S_0 - d_0 S_1 + d_1 S_1 - d_1 S_2 + . . . + d_{K-1} S_{K-1} - d_{K-1} S_K + d_K S_K$$

and common terms collected,

$$= d_0 S_0 + (d_1 - d_0) S_1 + (d_2 - d_1) S_2 + . . . + (d_K - d_{K-1}) S_K$$

and equated to the series of Eq. (31) to determine the relationship between the a_n in terms of the d_n:

$$\begin{aligned} a_0 &= d_0 \\ a_1 &= d_1 - d_0 \\ a_2 &= d_2 - d_1 \\ &\cdot \\ &\cdot \\ a_K &= d_K - d_{K-1}. \end{aligned} \qquad (33)$$

Eqs. (33) can in turn be solved for the d_n in terms of the a_n:

$$\begin{aligned} d_0 &= a_0 \\ d_1 &= a_0 + a_1 \\ d_2 &= a_0 + a_1 + a_2 \\ &\cdot \\ &\cdot \\ d_K &= \sum_{n=0}^{K} a_n, \text{ etc.} \end{aligned} \qquad (34)$$

Fig. 14 shows illustrations of the two different types of decomposition waveforms sets.

3. LOG SAMPLING IN THE FREQUENCY DOMAIN

As pointed out earlier, most spectral and frequency response data is sampled at equal-spaced intervals of frequency, although the data is most often viewed on a log frequency scale. This section formalizes the log-sampling process in the frequency domain, and develops an appropriate set of equations similar to the equations that describe log-sampling in time. Frequency-domain decomposition and reconstruction operations are not developed because these operations are not typically done on this type of data.

Similar to the previous section that developed equations describing log-sampling in the time domain, similar equations may be developed for log-sampling in the frequency domain. These equations are shown in Table 3.

Note that some of the variables used for time sampling have been used also for frequency sampling, i.e., $n, K, N_T, R, R_T, N_e, N_2,$ and N_3. This has been done to emphasize the parallel structure of the equations. Note that these variables need not be the same between log-time and log-frequency, because they can be chosen somewhat independently in the two domains. For example, the total number of points N_T can be different between time and frequency. You may want only 10 log-spaced points in the time domain, but 100 log-spaced points in frequency to yield higher frequency resolution. The sample densities $N_e, N_2,$ or N_3 may be different as well, etc.

4. RELATE TIME AND FREQUENCY FOR LOG SAMPLING

In this section, the relationships developed for log-sampling in time and frequency (Tables 1 and 3), are tied together. This is done by knowing that the initial sample rate $fs_{\max}$, the maximum frequency $f_{\max}$, and the minimum sample time $t_{\min}$ are directly related.

First, several relationships are developed for the maximum and minimum frequencies and times, and for the required total number of points, in terms of the resonator characteristics.

Second, two calculation procedures are developed which are based on two different sets of target parameters. The procedures start with the desired target requirements, which include the start and stop frequencies $f_{\min}$ and $f_{\max}$, the desired decay threshold L_{dB}, and the resonator Q or the total number of sample points N_T. The procedures then calculate the rest of the frequency and time parameters. Note that it is quite common to describe a system in the frequency domain rather than the time domain.

4.1. Development of Design Procedures

Three major assumptions are made to develop these procedures:

1. the sample rate is two and a half times the maximum frequency ($fs_{max} = 2.5 f_{\max}$),
2. the time span and the frequency span are equal, $R_T = f_{\max}/f_{\min} = t_{\max}/t_{\min}$, and
3. the time and frequency sample densities are equal, $N_x = N_{x\mathrm{Time}} = N_{x\mathrm{Freq}}$, or equivalently $R = R_{\mathrm{Time}} = R_{\mathrm{Freq}}$.

Assumptions 2 and 3 simplify the procedures.

Eq. (16) relates the initial (or maximum) sample rate to the starting (or minimum) sample time. From this relationship, the ending (or maximum) frequency in the frequency domain may be calculated as follows:

$$f_{\max} = \frac{fs_{\max}}{2.5} = 0.4 fs_{\max} = \frac{0.4}{(R-1)t_{\min}}. \tag{35}$$

This may in turn be solved for the start time $t_{\min}$, and expanded knowing the ratio R from Eq. (28):

$$\begin{aligned} t_{\min} &= \frac{0.4}{(R-1)f_{\max}} \\ &= \frac{0.4}{\frac{8\pi \log_{10}(e)}{L_{\mathrm{dB}}Q} f_{\max}} = \frac{0.4}{8\pi \log_{10}(e)} \bullet \frac{L_{\mathrm{dB}}Q}{f_{\max}} \\ &\approx 0.037 \frac{L_{\mathrm{dB}}Q}{f_{\max}} \end{aligned} \tag{36}$$

The stop time $t_{\max}$ may in turn be derived:

$$t_{\max} = R_{\mathrm{T}} t_{\min} = \left(\frac{f_{\max}}{f_{\min}} \right) t_{\min} = \frac{0.4}{8\pi \log_{10}(e)} \cdot \frac{L_{\mathrm{dB}}Q}{f_{\min}}. \tag{37}$$

Note that by specifying the start frequency $f_{\min}$, you are explicitly setting the ending sample time $t_{\max}$. This means that the sampling process continues long enough so that a resonator that rings down at lowest frequency $f_{\min}$, with the specified Q, will be properly sampled all the way to the time the decay falls below L_{dB}. Note also, that specifying the stop frequency $f_{\max}$, explicitly sets the initial sampling rate $f_{S_{\max}}$, and hence the start time $t_{\min}$.

The last count K and total points N_{T} (= K + 1) are calculated as follows from Eq. (11):

$$K = \mathrm{RoundUp}\left[\frac{\log_{10}(R_{\mathrm{T}})}{\log_{10}(R)} \right] = \mathrm{RoundUp}\left[\frac{\log_{10}\left(\frac{f_{\max}}{f_{\min}} \right)}{\log_{10}(R)} \right], \tag{38}$$

$$N_{\mathrm{T}} = K + 1 = \mathrm{RoundUp}\left[\frac{\log_{10}\left(\frac{f_{\max}}{f_{\min}} \right)}{\log_{10}(R)} + 1 \right]. \tag{39}$$

The $\text{RoundUp}[\;]$ operation rounds the result to the next highest integer. This may be in turn be expanded knowing the ratio R from Eq. (20):

$$N_T = \text{RoundUp}\left[\frac{\log_{10}\left(\dfrac{f_{\max}}{f_{\min}}\right)}{\log_{10}\left(\dfrac{8\pi\log_{10}(e)}{L_{\text{dB}}Q}+1\right)}+1\right], \tag{40}$$

which is the desired result. Note that the number of points rises in proportion to the log of the frequency span $f_{\max}/f_{\min}$ (or equivalently the time span), rather than to the span itself. This fact is the main contributor to the high efficiency of operations using log-spaced sampling, because the number of sample points required are very low as compared to equal-spaced sampling.

The complete procedures are shown in Table 4. To simplify the procedures, the calculation sequences appear in chain form where calculated values are used in later calculations. This is in contrast to the equations developed in this section, which are completely expanded.

4.2. Example Calculations

Two example designs are considered using the two developed procedures.

4.2.1. *Example 1: Specify Q, Start-Stop Frequencies, and Threshold*

Assume you want to capture the impulse response of a band-pass system which covers a frequency range of 20 Hz to 20 kHz, containing resonators having a maximum Q of 20, and requiring energy decays to be accurately preserved over a 90 dB dynamic range. Also assume the system's frequency response, including skirts, exists mainly in the 10 Hz to 24 kHz range (the response is down 90 dB or greater out of this range).

Therefore:

$f_{\min} = 10$ Hz, $f_{\max} = 24$ kHz, $Q = 20$, and $L_{\text{dB}} = 90$ dB.

According to procedure 1, the following parameters result:

R	≈ 1.00606389	N_e	≈ 165.36 points per e
R_T	$= 2400$ to 1	N_2	≈ 114.65 points per octave
K	$= 1287$	N_{10}	≈ 380.87 points per decade
N_T	$= 1288$ points	$f_{S_{\max}}$	$= 60$ kHz
$t_{\min}$	≈ 2.74851 ms	$f_{S_{\min}}$	$= 25$ Hz
$t_{\max}$	≈ 6.59642 s	$f_S(t)$	$\approx \dfrac{164.91}{t}$

Note how relatively far from zero the first time sample $t_{\min}$ is (2.74851 ms!). The main concern here is that the first portion of the impulse response would not be properly sampled. Eq. (22c) shows that the initial time rises in direct proportion to L_{dB} and Q (which are both relatively high), and inversely with $f_{\max}$. With a fixed L_{dB} and Q , the only way to decrease $t_{\min}$ is to increase the initial sampling rate and hence $f_{\max}$. The high start time may not be a problem if you assume that the impulse response of most real (bandpass) systems takes time to develop due to inherent delays in the system.

Note also that even though the last log sample is at $t_{\max}$, the reconstructed impulse response will extend significantly farther than $t_{\max}$, depending on the chosen interpolator.

4.2.2. Example 2: Specify Number Of Points , Start-Stop Frequencies, and Threshold

Assume you want to develop the parameters for a one-third-octave convolver operating over 20 Hz to 20 kHz, with a dynamic range of 60 dB. One-third-octave bands require 10 points per decade or 31 points (including end points) for the three decade range.

Therefore:

$$f_{\min} = 20 \text{ Hz}, f_{\max} = 20 \text{ kHz}, N_{\mathrm{T}} = 31, \text{ and } L_{\mathrm{dB}} = 60 \text{ dB}.$$

According to procedure 2, the following parameters result:

K	$= 30$	N_e	$\approx$ 4.34 points per e
R_{T}	= 1000 to 1	N_2	$\approx$ 3.01 points per octave
R	≈ 1.25892541	N_{10}	= 10 points per decade
Q	$= 0.70258393$	$f_{S\max}$	= 50 kHz
$t_{\min}$	$\approx$ 77.242 us	$f_{S\min}$	= 50 Hz
$t_{\max}$	$\approx$ 77.242 ms	$f_S(t)$	$\approx \dfrac{3.862}{t}$

With much fewer sample points, note how much smaller both minimum and maximum sample times are. Also note that the ratio R is approximately one-third-octave ($\approx 2^{1/3}$) as it should be, but that the Q is much to small for a one-third-octave response (requires a $Q \approx 1/(2^{1/6} - 2^{-1/6}) \approx 4.3$). Be aware that I have not shown that a one-third-octave equalizer response can be implemented with only 31 taps! More experiments are required! See Section 8.2.2 later.

5. LOG-SPACED FOURIER TRANSFORM

In this section a general method is developed to directly convert back and forth from a log-spaced sample sequence in the time domain, to a log-spaced sequence in the frequency domain. The method is based on approximating the discrete-time log-spaced sequence's continuous-frequency Fourier transform, which is then sampled at log-spaced intervals. The Fourier approximation is calculated by first interpolating the sequence, and then forming the transform in the opposite domain by summing the individual spectrums of each interpolator. The method is not based on any variation of the conventional equal-spaced-sample DFT or FFT techniques. Conventional DFT or FFT techniques can't easily be applied to log-spaced sampled sequences because the sequences aren't periodic, and the effective sample rate changes over the sequence length.

The method used to approximate Fourier transforms using interpolator spectrums, is detailed in [11, Schoenberg Lecture 10 Applications: 1. Approximations of Fourier Transforms, p. 109].

The procedure to calculate the log-spaced values in one domain directly from log-spaced values in the other domain is outlined as follows:

1. Choose an appropriate interpolator for the log sequence.

   The interpolator could be either the logsinc function Eq. (15), the more-efficient cubic-spline interpolator function described in Appendix 2, or the Hannsinc interpolator of Eq. (4b). The sequence itself is either a real causal sequence in the time domain, or a complex sequence in the frequency domain (the real and imaginary parts of the frequency spectrum are usually interpolated separately).
2. For the chosen interpolator, calculate appropriate log-spaced data in the opposite domain.

   This data is either a complex spectrum for the impulse response interpolator, or the impulse response for the complex frequency response interpolator. Note that these values need only to be calculated once for a specific (or reference) location of the interpolator on either the frequency or time axis. The data for other frequency or time locations is simply the reference-location interpolator data shifted up and down on the appropriate log scale. The real and causal nature of the impulse response is taken advantage of when working with data in the frequency domain, i.e., the real part of the interpolator spectrum must be even, and the imaginary part of the spectrum must be odd, and furthermore the real and imaginary parts are related by the Hilbert transform [1, p. 664].
3. Calculate log-spaced Fourier data in the opposite domain for the interpolated sequence.

   This is done by summing the individual spectral (forward) or time (reverse) contributions of each interpolator associated with each point in the original sequence.

The forward Fourier operation of step three is:

$$\begin{aligned}\tilde{X}[n] &= X_c\left(f_{\min}P^n\right) = \Im\{x(t)\} = \Im\left\{\sum_{m=0}^{K} x[m]\psi\left(t,t_{\min}R^m,N_x\right)\right\}\\ &= \sum_{m=0}^{K} x[m]\Im\left\{\psi\left(t,t_{\min}R^m,N_x\right)\right\}\\ &= \sum_{m=0}^{K} x[m]\tilde{\Psi}\left(f_{\min}P^n,t_{\min}R^m,N_x\right)\end{aligned} \qquad (41)$$

where

$\tilde{X}[n]$ = sequence of complex numbers representing sample amplitudes of spectrum at log-spaced intervals of frequency

$x[m]$ = sequence of numbers representing sample amplitudes at log-spaced intervals of time

$\Im$ = denotes the forward Fourier transform operator

$\psi\left(t,t_c,N_x\right)$ = an interpolation function defined for a log-spaced time sequence located at time t_c and having width N_x (a function continuous in time)

$\tilde{\Psi}\left(f,t_c,N_x\right)$ = spectrum of interpolation function ψ which is located at time t_c and having width N_x (a function continuous in frequency)

R = ratio relating one sample time to the previous sample time in a log-spaced time sequence (a real number > 1)

P = ratio relating one sample frequency to the previous sample frequency in a log-spaced frequency sequence (a real number > 1).

The final form of this equation clearly shows that the sequence's spectrum is formed by summing the spectrums of each of the individual interpolators. Note again that samples of the interpolator spectrum are only required at a finite set of log-spaced frequencies. These samples may be pre computed and then tabled for future use. Note also that these samples need only be computed once for one reference center time. Other log-spaced sampled spectrums, for other center times, are trivially generated by just shifting the samples in the reference spectrum sequence up and down in log frequency.

The development of a detailed equation for the inverse Fourier calculation corresponding to step three is straight forward, but is not shown here.

6. STORAGE AND CALCULATION EFFICIENCY

The efficiency of log-spaced sampling results directly from the dramatic reduction of sample points, as compared to conventional equal-spaced sampling. This reduction means less memory requirements, reduced processing time, and lower transmission time and/or bandwidth.

6.1. Storage and Transmission Efficiency

A measure of the efficiency of log sampling may be calculated by comparing the required number of sample points for conventional uniform-spaced sampling to the required number of points for log-spaced sampling.

In conventional uniform-spaced sampling, the required number of sample points rises in direct proportion to the sample rate and the total sampling duration, as shown in the following:

$$N_{\mathrm{T\,lin}} = f_S T_d = 2.5 f_{\max} T_d \qquad (42)$$

where

$N_{\mathrm{T\,lin}}$ = total number of linear-spaced sample points

f_S = sample rate in samples per second

$f_{\max}$ = highest frequency existing in signal to be sampled

T_d = total sampling duration.

In contrast to linear-spaced sampling, the required number of sample points for log-spaced sampling rises in direct proportion to the *log* of the sample time span and the sample density, as shown in the following (assuming that $K >> 1$):

$$N_{\mathrm{T\,log}} = K + 1 \approx K = \frac{\log_{10}(R_{\mathrm{T}})}{\log_{10}(R)} = N_{10}\log_{10}(R_{\mathrm{T}}) = N_{10}\log_{10}\left(\frac{t_{\max}}{t_{\min}}\right) \qquad (43)$$

where

N_{10} = sample density in points per decade

$N_{\mathrm{T\,log}}$ = total number of log-spaced sample points

R_{T} = sample time span, (= $t_{\max}/t_{\min}$)

$t_{\min}$ = starting sample time

$t_{\max}$ = ending sample time
K = last count.

Assuming a maximum frequency of 20 kHz, a sampling duration of 10 s, the number of points required for linear-spaced sampling is 2.5(20,000) 10 = 500,000 points. Assuming a sample density of 100 points per decade (an approximate Q of 8 and a threshold of 60 dB), and a 1000 to one time span, the number of points required for log-spaced sampling is 100log(1000) = 300 points. This represents a compression factor of about 500,000/300 ≈ 1,667 to 1, a major reduction in the amount of data that has to be stored, manipulated, or transferred!

6.2. Log-Spaced Fourier Transform Efficiency

In this section, the efficiency of the log-log Fourier transform (LLFT Section 5.) is compared to the conventional discrete Fourier transform (DFT), and the fast Fourier transform (FFT). First the efficiencies are compared by knowing the approximate number of operations that are required for a specific number of data points. Secondly, the efficiencies are compared by identifying the approximate number of operations that are required for a signal of specific length and specific upper frequency limit.

It is well known that the DFT requires approximately $2N^2$ operations to convert from one domain to the other, where N is the number of data points. It is also well known that the FFT requires about $N \log_2(N)$ operations for the same number of data points. Analysis shows that the LLFT presented here requires about the same number of operations as the DFT ($2N^2$), but with the added overhead of having to pre-compute and store the spectrum of the interpolator. Compared this way, the LLFT comes out a poor third behind the DFT and FFT.

However, when the number of operations of the LLFT is calculated for a signal of specific length and specific upper frequency limit, the conversion efficiency of the LLFT is quite high as compared to the DFT and FFT. This is because the LLFT manipulates considerably less data. This is shown by calculating the approximate number of operations $O(N)$ required for each method using the relations of Eqs. (42) and (43):

For the DFT:

$$\begin{aligned} O(N)_{DFT} &\approx 2N^2 = 2N_{\mathrm{Tlin}}^{\;2} \\ &= 2(2.5 f_{\max} T_d)^2 \\ &= 12.5 f_{\max}^{\;2} T_d^{\;2} \end{aligned} \qquad (44)$$

For the FFT:

$$\begin{aligned} O(N)_{FFT} &\approx N \log_2(N) = N_{\mathrm{Tlin}} \log_2(N_{\mathrm{Tlin}}) \\ &= 2.5 f_{\max} T_d \log_2(2.5 f_{\max} T_d) \end{aligned} \qquad (45)$$

For the LLFT:

$$\begin{aligned} O(N)_{LLFT} &\approx 2N^2 = 2N_{\mathrm{Tlog}}^{\;2} = 2[N_{10} \log_{10}(R_{\mathrm{T}})]^2 \\ &= 2N_{10}^{\;2} [\log_{10}(R_{\mathrm{T}})]^2 \end{aligned} \qquad (46)$$

Using the same example numbers from before, which assumes a maximum frequency of 20 kHz, and a sampling duration of 10 s, the DFT requires about $(2(500{,}000)^2 \approx 5\text{x}10^{11})$ 500 billion operations!, and the FFT requires about $(500{,}000\log_2(500{,}000) \approx 9.5\text{x}10^6)$ 9.5 million operations. Assuming a sample density of 100 points per decade, and a 1000 to one time span, the LLFT requires about $(2(300)^2 \approx 1.8\text{x}10^5)$ 180 thousand operations. These numbers yield a compression factor of about 2,800,000 to 1 comparing the DFT to the LLFT, and 52.8 to 1 comparing the FFT to the LLFT! Both are substantial reductions in the required number of operations!

Comparing the number of operations of the FFT to the LLFT yields a crossover point, below which the LLFT is more efficient. An equation for the crossover point may be derived by equating the number of operations required by each method but using different counts:

$$\begin{aligned} O(M)_{LLFT} &= O(N)_{FFT} \\ 2M^2 &= N\log_2(N) \\ M &= \sqrt{\frac{N\log_2(N)}{2}} \end{aligned} \qquad (47)$$

For the previous example, the LLFT is only more efficient than the FFT if the LLFT requires less than about 2176 points in the transform ($M = \sqrt{\frac{500\text{k}\log_2(500\text{k})}{2}} \approx 2{,}176$).

7. APPLICATION EXAMPLES: LOG DECOMPOSITION AND RECONSTRUCTION IN THE TIME DOMAIN

In this section, several examples of log-spaced decomposition and reconstruction are shown. The decomposition and reconstruction follow the procedures of Sections 2.2.6 and 2.2.7 using Eqs. (22) and (23), but use an interpolator which is a log-warped version of the cubic spline interpolator of Section 2.1.4, rather than the logsinc.

Firstly, a sinewave of several frequencies is decomposed and reconstructed. Secondly, the impulse response of a computer model is decomposed and reconstructed for several different log-spaced sample densities. The model's frequency response is also shown for each situation. In all situations, the reconstructed waveform is shown along with the original waveform.

7.1. Sinewave Decomposition and Reconstruction

Fig. 15 shows several log decomposition/reconstructions of sine waves of various frequencies (100 Hz, 400 Hz, and 4 kHz), with a fixed sample density of 32.78 points per decade, a starting sample rate of 50 kHz, with 90 log sample points total. This corresponds to a ratio R of 1.07276673, with a log start time of 274.85 us.

A width 18 log-warped cubic spline function (Fig. 9(b)) was used as the decomposition and reconstruction interpolator. The sample density of 32.78 points per decade was computed by using Eq. (29), assuming a 60 dB decay threshold and a Q of 2.5. Before decomposition and reconstruction, the sine waves existed over the whole time axis.

Note that after the log decomposition/reconstruction operation, the lower frequencies extend farther out in time than the high frequencies. The log decomposition and reconstruction process essentially forms a filter that attenuates frequencies after a specific time period elapses. The time period is short for high-frequencies and long for low frequencies.

The filtering operation essentially allows only a constant number of cycles of the sinewave to get through the filter, independent of the sine wave's frequency. The number of cycles may be computed by multiplying the delay time T_d of Eq. (24) by the center frequency f_0. This results in approximately 5.5 cycles in this situation, which is verified by the simulations.

Note that the log sampling essentially missed the first cycle of the 4 kHz sinewave (Fig. 15(a)) due to the fairly-high log start time.

7.2. System Impulse Response Decomposition and Reconstruction

The impulse response of a model similar to the model described in Section 1.1 (the frequency response was scaled up by about one-third octave) was log decomposed and reconstructed for several different values of sample density. These graphs are shown in Fig. 16. The original impulse waveform is shown plotted on a log time scale (Fig. 16(a) left graph), along with the spectrum of the impulse response (the frequency response) also plotted on a log scale (Fig. 16(a) right graph). Thereafter each reconstruction is shown along with its spectrum for comparison.

The original impulse response was sampled with 8192 equal-spaced points. As before, a width-18 log-warped cubic spline function (Fig. 9(b)) was used as the decomposition and reconstruction interpolator. Successive sample densities of 10, 25, 50, and 100 points per decade were used. Total number of log sample points ranged over 30, 75, 150, and 300 points. Initial sample rates were 19.3 kHz, 51.8 kHz, 106.1 kHz, and 214.7 kHz, respectively. The log sample time range ran over 0.2 ms to 0.2 s for all simulations. Note that a sample density of 100 points per decade was required to reconstruct the original data accurately (Fig. 16(e)).

Here, the time-domain log decomposition and reconstruction process essentially smoothes the frequency response with a constant Q or equal percentage bandwidth filter. The smoothing decreases with higher sampling densities. An essentially equivalent procedure using conventional techniques would be to: 1) convert the equal-spaced sampled impulse response data to the frequency domain using an FFT, 2) smooth the equal-spaced frequency data with a constant percentage bandwidth filter, and 3) convert the data back to the time domain using the FFT.

8. APPLICATION: FIR CONVOLVERS

One very important application area of log sampling is in the design of FIR convolvers. As pointed out before, DSP systems are often used to implement such common audio devices such as filters, equalizers, crossovers, reverberators, room auralization processors, etc. Implementing these devices are often complex because of the very-wide three-decade frequency range that audio covers. Infinite impulse response IIR filter designs often have to use multi-rate techniques to get around these problems, while FIR convolver designs have to be very long and store a lot of data to accomplish the same processing.

In this section, log sampling is applied to the design of FIR convolvers. Log-sampling techniques generate FIR filter structures that gracefully and naturally cover very wide frequency ranges with very few taps. The emphasis here is on the design of continuous-time convolvers for clarity. Extension to discrete-time designs is straight forward.

8.1. General FIR Convolver Design

This section briefly describes conventional FIR convolvers based on equal-spaced sampling techniques, and then develops and describes convolvers based on log-spaced sampling. The analysis and development is based on representing the convolvers as continuous-time tapped-delay-line filters rather than discrete-time FIR filters. Looking at the convolvers as tapped-delay-line filters illustrates the design concepts in the most general manner.

8.1.1. Convolvers with Equal-Spaced Taps

Fig. 17 shows the block diagram of a conventional FIR filter/convolver, which is based on equal-spaced sampling. The block diagram is depicted as a continuous-time tapped-delay-line filter rather than a discrete-time FIR filter. Note that all the delay blocks provide the same time delay. The output is a weighted sum of the individual outputs from each delay block. Note that even though the convolver operates in continuous time, a brickwall low-pass reconstruction filter is still required on the output of the system due to the discrete nature of the taps.

The output of this tapped-delay-line convolver appears as:

$$r(t) = h_{LP}(t) * \left[\sum_{n=0}^{K} a_n e(t - nT) \right] \qquad (48)$$

where

$r(t)$ = response or output signal as a function of time
$e(t)$ = excitation or input signal as a function of time
a_n = coefficient at tap n
T = delay time of each block in delay line
$h_{LP}(t)$ = impulse response of low-pass reconstruction interpolation filter
$*$ = denotes the operation of convolution in time

Eq. (48) may be rewritten to move the low-pass filter convolution operation inside the summation so that it operates on each tap individually :

$$r(t) = \sum_{n=0}^{K} h_{LP}(t) * a_n e(t - nT) \qquad (49)$$

Fig. 18 illustrates this entirely equivalent structure for the convolver of Fig. 17, which I will call the cascade structure. Here, identical low-pass reconstruction filters are used on each tap of the delay line. This structure, of course, is very inefficient because of the much greater number of low-pass filters required. This form is presented however, because it will be shown that it is required for the log-spaced convolver described in the next section. Note that it is possible to use a single low-pass reconstruction filter in the equal-spaced sample convolver, only if all the delay blocks provide the exact same delay. If the delays are different, separate low-pass filters must be used.

An alternate structure equivalent to Fig. 18 is shown in Fig. 19, which I call the parallel structure. Here each channel is driven in parallel and has its own delay. No tapped delay line is used. This is the most inefficient of all due to the duplication of both the filters and the time delay blocks! However, this structure will be useful later for a variant of the log-spaced convolver.

8.1.2. Convolvers with Logarithmic-Spaced Taps

Fig. 20 shows the cascade-structure log-spaced convolver. Note that this block diagram is essentially the same as Fig. 18, but with the additional requirement that each low-pass filter is distinct and matched to each specific tap. Additionally, each low-pass filter is more complex and special in that it's impulse response is log-warped in time, and of the form of Eq. (21). As is shown in Section 2.2.4, the log warping is required in order to optimally interpolate between the log-spaced sequence of taps. As noted previously, it is not possible to implement the log-spaced convolver with only one low-pass filter as in the form of block diagram of Fig. 17, because the tap delays are different.

For practical purposes, each of the low-pass log reconstruction filters of Fig. 20 has a finite impulse response and is implemented as a FIR filter. These filters are described by three parameters: center time t_c in seconds, sample density N_x in points per x (e, octave, decade, etc.), and the time span ratio R_F which is the ratio between the start t_{start} and end t_{end} times of the filters impulse response ($R_F = \frac{t_{end}}{t_{start}}$). The center time of the filter t_c ($= \sqrt{t_{start}t_{end}}$) must coincide with the sample time of the tap it is connected to.

The start and end times of the filter's impulse response are:

$$t_{start} = \frac{t_c}{\sqrt{R_F}}, \text{ and}$$
$$t_{end} = t_c\sqrt{R_F}\,. \qquad (50)$$

Note that even though the filters impulse response is supposed to start at t_{start}, the actual start is at zero. This means that the filter must be delayed by time t_{start} so that the filters output will be synchronized for proper reconstruction. This extra delay must be supplied by an additional delay block which is supplied by the delay line. The following expression yields the time delay for each delay block:

$$T_n = \begin{cases} T_0 = t_{\min}\left(1 - \dfrac{1}{\sqrt{R_F}}\right) & \text{for } n = 0 \\ T_0(R-1)R^{n-1} & \text{for } n = 1 \text{ to } K \end{cases} \qquad (51)$$

where

T_n = time delay for block n
T_0 = delay of block 0 (first block)
$t_{\min}$ = initial sample time (center time for first convolver reconstruction filter
R = multiplier or ratio relating one sample time to the previous sample time
n = sample count, = 0, 1, 2, . . . K
R_F = time span ratio of the reconstruction filter impulse response ($R_F = \frac{t_{end}}{t_{start}}$)
K = last sample count.

Note that when the log convolver of Fig. 20 is compared to the linear-spaced convolver of Fig. 17, that much greater computational overhead is required at each tap of the log convolver. This is a great disadvantage of the log structure, but remember the log convolver requires *considerably fewer* taps than the conventional convolver. Only time will tell if the tap filters can be efficiently implemented, and thus make the log structure more efficient than the conventional structure.

Fig. 21 shows the parallel-structure log-spaced convolver. Note that this block diagram is essentially the same as Fig. 19, but with the additional requirement that each low-pass filter is distinct and matched to each specific channel. The delay of each block is given by

$$T_n = t_{\min}\left(1 - \frac{1}{\sqrt{R_F}}\right)R^n \qquad \text{for } n = 0 \text{ to } K. \qquad (52)$$

Note that in some cases the time delay of a specific block is a small fraction of the time width of the reconstruction filter it is associated with. This means that the delay can logically be absorbed into the FIR reconstruction filter block it is connected to. This simplifies the structure of the convolver. Note that each resultant filter/delay block is scale invariant when looked at with a scale that varies as R^n.

Each of the log-spaced convolvers described in this section (and the next) can be designed using the differenced form of the interpolator function as described in Section 2.2.9, instead of the standard low-pass form.

8.1.3. Discrete-Time Multirate Log-Spaced Convolver

The structure of the continuous-time convolver of Fig. 21 suggests the discrete-time multirate structure of Fig. 22(a). Each circle with an up arrow or down arrow indicates the operation of up sampling and down sampling respectively, by the fractional amount R (which could be two for an octave convolver). The down-sampler block has appropriate low-pass filtering that reduces the upper-frequency limit of the data passing through it by the ratio R.

Parts such as the Analog Devices AD1890 and AD1891 asynchronous sample rate converters [12] may be used to implement the fractional sample rate conversion. In this situation, each FIR filter/delay block is *identical*, the changing sample rate automatically scales the impulse response up and down in time. The FIR filter/delay block's log-warped impulse response is critically sampled at equal-spaced intervals and stored in the block (see Fig. 22(b)). The highest-frequency highest-sample-rate channel is at the top, while the lowest-frequency lowest-sample-rate channel is at the bottom. This form of convolver suggests one that is based on multi-rate wavelet techniques [13], but is much simpler because the outputs of each channel are simply summed after appropriate synchronous up sampling.

8.2. Impulse and Frequency Responses Generated by Log-Spaced FIR Convolvers

In this section, two log convolvers of different lengths are analyzed to illustrate the variety of impulse and frequency responses available. It is surprising how well log convolvers perform with just a few number of taps. Both convolvers are assumed to operate over the standard audio range of 20 Hz to 20 kHz. In every case, the tap coefficients are listed in both standard and differenced form following the conversion equations listed in Section 2.2.9.

The steps involved in generating these responses are: 1) reconstruct a high-resolution impulse response working directly from the log-based samples or coefficients using Eq. (17), but with a log-warped cubic-spline interpolator rather than a logsinc, 2) plot this time response on log time scales, 3) differentiate the impulse response with respect to linear time, 4) convert impulse response to the frequency domain by conventional FFT or using the LLFT described in this paper, 5) plot magnitude of frequency response on log frequency scale.

The differentiation operation of step 3 is required to compensate for the lack of proper high-low frequency weighting of the reconstruction process, i.e., because the decomposition-reconstruction process uses constant amplitude interpolation functions whose energy is not constant with frequency (the long-time low-frequency interpolators have more energy than the short-time high-frequency interpolators).

8.2.1. *Ten Tap Octave Convolver*

Figs. 23(a) to (k) shows various impulse and frequency responses for an octave ratio (R = 2) convolver of 10 taps. It was designed assuming a convolver operating over 31.25 Hz to 16 kHz, with a dynamic range of 60 dB. Octave bands require one point per octave or about 3.32 points per decade or 10 points (including end points) for the stated range (512 to 1).

Therefore:

$$f_{\min} = 31.25 \text{ Hz}, f_{\max} = 16 \text{ kHz}, N_T = 10, \text{and } L_{dB} = 60 \text{ dB}.$$

According to procedure two of Table 4, the following parameters result:

K	$= 9$	N_e	$\approx$ 1.44 points per e
R_T	$= 512$ to 1	N_2	$= 1$ point per octave
R	$= 2$	N_{10}	$\approx$ 3.32 points per decade
Q	≈ 0.18191683	$fs_{\max}$	$= 40$ kHz
$t_{\min}$	$= 25$ us	$fs_{\min}$	$= 78.125$ Hz
$t_{\max}$	$= 12.8$ ms	$f_S(t)$	$\approx \frac{1}{t}$

Various impulse and frequency responses for this convolver are shown in Figs. 23(a) to (k). Log scales are used for both time and frequency graphs. Note that even though the last tap of the convolver is at $t_{\max} = 12.8$ ms, the impulse response of the convolver extends significantly farther out in time to about 0.2 s. Also note that if this convolver is implemented using equal-spaced FIR techniques, it would require an FIR filter of at least 8000 points (= 40k samples/s x 0.2s).

The spectrums were calculated using conventional FFT techniques using a 32,768 point data record scaled to 0 to 0.2048 s. This provided a time step size of 6.25 us (= 0.2048/32768) and a frequency step size of 4.8828125 Hz (= 1/0.2048). The impulse response was first constructed in an eight-decade 800 point log-spaced data record covering 0.1 us to 10 s, with 100 points per decade, and then interpolated to the 32,768 point linear-spaced data record.

Various responses are shown including low pass, band pass, high pass, and peak dip. Two sets of responses are shown that were generated with the 10 coefficients loaded with random numbers in the range of ±1.0 (Fig. 23(j) and (k)).

One interesting combination of log coefficients occurs when consecutive tap weights alternate between +1 and -1 (Fig. 23(f)). This generates a convolver impulse response that is a log down sweep. As a result, each portion of its time response is associated one on one with a portion of its frequency response. Increasing or decreasing a pair of adjacent coefficients by a factor of say 2, causes a 6 dB bump or dip in the response at the associated frequency. This is illustrated in Figs. 23(g) and (h). This one on one correspondence makes it easier to design for a specific frequency response. The major problem with this is the very dispersive nature of a log-down-sweep impulse response, which makes the convolver only suitable for use as an equalizer where phase is not important, i.e. shaping white or pink noise, etc.

8.2.2. *Thirty One Tap One-Third-Octave Convolver*

Figs. 23(l) and (m) shows two representative impulse and frequency response sets for a one-third-octave ratio ($R = 10^{1/10} \approx 2^{1/3} \approx 1.259$) convolver of 31 taps. The parameters of this convolver are the same as the convolver of the example in Section 4.2.2. Note that if this convolver is implemented using equal-spaced FIR techniques, it would require an FIR filter of length of about 30,000 points (= 50,000 samples/sec x 0.6 seconds).

As noted before, 31 points in the log-spaced sequence does not provide a high enough Q for a true one-third-octave response. According to the guidelines of this paper in Table 4, a required one-third-octave Q of about 4.3 requires a sample density of about 56 points per decade (according to Fig. 13), or 168 points for three decades.

Both sets of graphs were generated by filling the 31 coefficients with random numbers in the range of ±1.0. The higher resolution of these two graphs is quite evident when compared to the previous responses generated by the octave convolver. Even though limited to only 31 taps, this convolver can generate quite varied responses.

8.3. Room Auralization Processor

The log-spaced convolver techniques described in this paper are well suited for real-time audio processors required to "auralize" or simulate the sound of an acoustic space before it is built. To operate in real time, convolvers using conventional techniques must have very high computational capability. Log-spaced convolver designs may ease this burdan.

Research needs to be done to determine what effective ring-down Q needs to be maintained, so that psycho-acoustically, the room sound is not compromised as implemented using log-spaced techniques. Several log-spaced decomposition and reconstruction cycles, at various sample densities, should be applied to a full-length high-sample rate room impulse response, to determine how few log-spaced samples are required.

9. CONCLUSIONS

A non-uniform logarithmic-spaced sampling technique is described that is well matched to the impulse response decay characteristics of real-world systems. The sampling occurs at high rates for short times, and low rates for long times, with the sampling rate falling inversely with time. This places a high density of samples at the start of an impulse where the high-frequency content occurs, and lower densities later, where only lower frequencies exist. The impulse response of most real-world systems follows essentially a constant Q characteristic, where the system's high-frequency resonators ring down quickly, and the low-frequency resonators ring down slowly.

Log sampling is very efficient because of the very low number of samples required as compared to high sample rate equal-spaced techniques. Log sampling is a very efficient technique for data compression, where fewer samples imply less memory and processing time, faster transmission time and/or lower bandwidth for transmission. Log sampling in time is best described by a sample density, in points per decade or octave of time, rather than a sampling rate as it is in conventional equal-spaced sampling.

A log-spaced decomposition and reconstruction technique was developed, which is based on interpolators that have been log warped so that they function in log time the same as non-warped interpolators operate in linear time. These log-warped interpolation functions allow a log-spaced sequence of samples to be optimally interpolated, and thus reconstruct a continuous-time impulse response from its log-spaced samples.

An efficient technique was developed to convert directly back and forth from a sequence of log-spaced samples in the time domain to a sequence of log-spaced samples in the frequency domain. This technique is based on approximating the Fourier transform of the continuous-time impulse by summing up the individual spectrums of the interpolators used to interpolate the log-spaced sequence of samples of the impulse response.

Several forms of log-spaced convolvers were shown which may provide real utility in implementing such common audio devices as filters, equalizers, crossovers, reverberators, room auralization processors, etc. The log-spaced convolvers are much more complex than conventional equal-spaced designs, but are potentially much more efficient because considerably fewer taps or sample points are required.

10. REFERENCES

[1] A. V. Oppenheim and R. W. Schafer, "Discrete-Time Signal Processing," Prentice Hall, New Jersey (1989).

[2] R. J. Marks II, Editor, “Advanced Topics in Shannon Sampling and Interpolation Theory,” Springer-Verlag, New York (1993).

[3] R. W. Adams, “Nonuniform Sampling of Audio Signals,” *J. Audio Eng. Soc.*, vol. 40, pp. 886-894 (1992 Nov.).

[4] R. C. Johnson, H. Jasik, Editors, “Antenna Engineering Handbook,” McGraw-Hill, New York (1984).

[5] E. A. Wolff, “Antenna Analysis,” Artech House, Massachusetts (1988).

[6] A. Antoniou, “Digital Filters, Analysis, Design, and Applications,” McGraw-Hill, New York (1993).

[7] R. A. Roberts, C. L. Mullis, “Digital Signal Processing,” Addison-Wesley, Reading, Norwood, MA (1987).

[8] M. Kleiner, Editor, Special issue on auralization, *J. Audio Eng. Soc.*, vol. 41, pp. 861-945 (1993 Nov.).

[9] C. K. Chui, “Wavelets: A Tutorial in Theory and Applications”, Academic Press, Boston (1992).

[10] M. Unser, A. Aldroubi, and M. Eden, “Fast B-spline Transforms for Continuous Image Representation and Interpolation,” *IEEE Pattern Anal. Machine Intell.*, vol. 13, pp. 277-285 (1991).

[11] I. J. Schoenberg, “Cardinal Spline Interpolation,” Regional Conference Series in Applied Mathematics, vol. 12, Soc. for Industrial and Applied Mathematics (SIAM), Philadelphia (1973).

[12] R. W. Adams, “Theory and VLSI Implementation of Asynchronous Sample-Rate Converters,” presented at the 94th Convention of the Audio Engineering Society, Berlin, 1993 March 16-19, preprint no. 3570 (D4-9).

[13] O. Rioul, P. Duhamel, “Fast Algorithms for Discrete and Continuous Wavelet Transforms,” *IEEE Trans. Infor. Theory*, vol. 38, pp. 569-586 (Mar. 1992).

11. APPENDIX 1: RING-DOWN TIME OF SECOND-ORDER RESONATOR

The transfer function of a second-order low-pass filter is

$$H(s) = \frac{B}{s^2 + \frac{\omega_0}{Q}s + \omega_0^2} \qquad (53)$$

where

s = complex variable ($= j\omega = j2\pi f$)

ω_0 = corner frequency of filter in radians per second ($= 2\pi f_0$)

Q = Q of filter

B = a real constant.

An alternate form of Eq. (53) can be written in terms of the real α and imaginary ω_1 parts of the filters pole locations ($p_1 = \alpha + j\omega_1$, $p_1 = \alpha - j\omega_1$):

$$H(s) = \frac{\omega_1}{(s+\alpha)^2 + {\omega_1}^2} = \frac{\omega_1}{s^2 + 2\alpha s + {\omega_1}^2 + \alpha} \tag{54}$$

The pole real and imaginary parts can be written in terms of the filter parameters f_0 and Q:

$$\alpha = \frac{\omega_0}{2Q} = \frac{2\pi f_0}{2Q} = \frac{\pi f_0}{Q}, \text{ and}$$

$$\omega_1 = \sqrt{{\omega_0}^2 - \alpha} = \sqrt{{\omega_0}^2 - \frac{\omega_0}{2Q}}. \tag{55}$$

The inverse Laplace transform of Eq. (54) is

$$h(t) = e^{-\alpha t}\sin(\omega_1 t). \tag{56}$$

This is recognized as a decaying sinusoid at frequency ω_1. The filter's decay is described by the decay factor $e^{-\alpha t}$, which in turn can be equated to a specific decay threshold L:

$$e^{-\alpha t} = e^{-\frac{\pi f_0}{Q}t} = L. \tag{57}$$

This is turn can be solved for the time T_d to decay to this threshold

$$T_d = -\frac{Q\log_e(L)}{\pi f_0}. \tag{58}$$

This can be converted to L in dB by substituting $L = 10^{-L_{dB}/20}$

$$\begin{aligned} T_d &= -\frac{Q\log_e(L)}{\pi f_0} = -\frac{Q\log_e\left(10^{-L_{dB}/20}\right)}{\pi f_0} \\ &= -\frac{Q\log_{10}\left(10^{-L_{dB}/20}\right)/\log_{10}(e)}{\pi f_0} = -\frac{Q(-L_{dB}/20)/\log_{10}(e)}{\pi f_0} \\ &= \frac{L_{dB}}{20\pi\ \log_{10}(e)} \cdot \frac{Q}{f_0} \end{aligned} \tag{59}$$

Which is the desired result.

12. APPENDIX 2: CARDINAL CUBIC-SPLINE INTERPOLATOR

The following is a very abbreviated explanation of cubic-splines and their application to interpolation found in a section of [9, pp. 91-122]. Conventionally, cubic-spline interpolation of a finite set of data points is accomplished by a complex matrix techniques. The interpolation method described here is based not on matrix methods, but on a filtering approach, i.e., find the impulse response of an interpolation filter that when the sequence of data points is passed through it, the output is a continuous-time function with the region between the data points properly interpolated with cubic splines.

The cardinal cubic spline interpolator is based on the theory of polynomial splines. A function can be approximated by a series of equal-length polynomial line segments strung end to end. At each point where the individual segments connect, called knots, the value of the polynomial and a certain number of derivatives match. Because of their simple analytic form, splines are easy to manipulate numerically. Third order polynomials, called cubic splines, are the most popular and useful for numeric operations.

Schoenberg [11] has shown that any polynomial spline of degree n can be represented by a linear combination of shifted B-splines. The B-splines are symmetric bell-shaped functions of finite width that can be explicitly be defined with simple polynomial splines. Without going into any development detail (the reader is again referred to [9, pp. 91-122] for more details), the following equation defines the third-order B spline $\beta^3(x)$:

$$\beta^3(x) = \begin{cases} \frac{1}{6}(x+2)^3 & \text{for } -2 \le x < -1 \\ \frac{2}{3} - x^2\left(1 + \frac{x}{2}\right) & \text{for } -1 \le x < 0 \\ \frac{2}{3} - x^2\left(1 - \frac{x}{2}\right) & \text{for } 0 \le x < +1 \\ -\frac{1}{6}(x-2)^3 & \text{for } +1 \le x \le +2 \\ 0 & \text{otherwise} \end{cases} \qquad (40)$$

Note that $\beta^3(x)$ exists only over $-2 \le x \le +2$. This function is shown in Fig. 24, along with its derivatives.

The cardinal cubic spline interpolator is defined in terms of a series of shifted versions of $\beta^3(x)$. The coefficients of each shifted B spline are calculated numerically by first sampling $\beta^3(x)$ at the integers (for x = -2, -1, 0, +1, +2; $\beta^3(x)$= 0, 1/6, 2/3, 1/6, 0), and then numerically performing the convolution inverse. This results in an infinite series of coefficients, which can be truncated when the coefficients drop below ≈ 31.62×10^{-6} or about 90 dB down.

In terms of these coefficients (21 in all), the cardinal cubic spline interpolator $\eta^3(x)$ is given by

$$\eta^3(x) \approx \begin{cases} \left.\begin{matrix} +1.73205066\beta^3(x) \\ -0.46410161\beta^3(x-1) - 0.46410161\beta^3(x+1) \\ +0.12435565\beta^3(x-2) + 0.12435565\beta^3(x+2) \\ -0.03332100\beta^3(x-3) - 0.03332100\beta^3(x+3) \\ +0.00892834\beta^3(x-4) + 0.00892834\beta^3(x+4) \\ -0.00239234\beta^3(x-5) - 0.00239234\beta^3(x+5) \\ +0.00064103\beta^3(x-6) + 0.00064103\beta^3(x+6) \\ -0.00017176\beta^3(x-7) - 0.00017176\beta^3(x+7) \\ +0.0004602\beta^3(x-8) + 0.0004602\beta^3(x+8) \\ -0.00001234\beta^3(x-9) - 0.00001234\beta^3(x+9) \\ +0.00000332\beta^3(x-10) + 0.00000332\beta^3(x+10) \end{matrix}\right\} & \text{for } -9 \le x \le +9 \\ 0 & \text{otherwise} \end{cases} \tag{41}$$

The $\eta^3(x)$ function is plotted in the previous Fig. 9, on both linear and log vertical scales. Note that although it looks superficially like a sinc function, it dies down much faster. Even though this function appears complicated, it can be implemented quite efficiently using recursive digital filters [10]. Realize again, that $\eta^3(x)$ is the impulse response of a cubic-spline interpolation filter that will properly interpolate a series of data points that are sent through it. The interpolation property of $\eta^3(x)$ is due to the fact that it is precisely equal to one at x = 0 and vanishes at all other integer values.

TABLE 1. GENERAL EQUATIONS FOR LOG SAMPLING IN TIME DOMAIN

QUANTITY	RATIO	BASE *e*	BASE 2	BASE 10
Time at nth sample, t_n:	$= t_{\min} R^n$	$= t_{\min} e^{\frac{n}{N_e}}$	$= t_{\min} 2^{\frac{n}{N_2}}$	$= t_{\min} 10^{\frac{n}{N_{10}}}$
Count range:	$0 \le n \le K$	$0 \le n \le K$	$0 \le n \le K$	$0 \le n \le K$
Total points, $N_T = K + 1$:	$= \frac{\log(R_T)}{\log(R)} + 1$	$= N_e \log_e(R_T) + 1$	$= N_2 \log_2(R_T) + 1$	$= N_{10} \log_{10}(R_T) + 1$
First sample time, t_0:	$= t_{\min}$	$= t_{\min}$	$= t_{\min}$	$= t_{\min}$
Last sample time, t_K or $t_{\max}$:	$= t_{\min} R^K$	$= t_{\min} e^{\frac{K}{N_e}}$	$= t_{\min} 2^{\frac{K}{N_2}}$	$= t_{\min} 10^{\frac{K}{N_{10}}}$
Time span, $R_T = \frac{t_K}{t_0} = \frac{t_{\max}}{t_{\min}}$:	$= R^K$	$= e^{\frac{K}{N_e}}$	$= 2^{\frac{K}{N_2}}$	$= 10^{\frac{K}{N_{10}}}$
Ratio, R:	$= \frac{t_{n+1}}{t_n}$	$= e^{\frac{1}{N_e}}$	$= 2^{\frac{1}{N_2}}$	$= 10^{\frac{1}{N_{10}}}$
Ratio, R in terms of resonator Q, and decay threshold L_{dB}:	$= \frac{8\pi \log_{10}(e)}{L_{dB} Q} + 1$	Not applicable	Not applicable	Not applicable
Sample density: N_e, pts per e: N_2, pts per octave: N_{10}, pts per decade:	Not applicable	$N_e = \frac{1}{\log_e\left(\frac{8\pi \log_{10}(e)}{L_{dB} Q} + 1\right)}$	$N_2 = \frac{1}{\log_2\left(\frac{8\pi \log_{10}(e)}{L_{dB} Q} + 1\right)}$	$N_{10} = \frac{1}{\log_{10}\left(\frac{8\pi \log_{10}(e)}{L_{dB} Q} + 1\right)}$

TABLE 2. GENERAL EQUATIONS FOR LOG-SAMPLING SAMPLE RATE

QUANTITY	RATIO	BASE e	BASE 2	BASE 10
Sample rate at nth sample, f_{s_n}:	$= f_{s_{\max}} R^{-n}$	$= f_{s_{\max}} e^{-\left(\frac{n}{N_e}\right)}$	$= f_{s_{\max}} 2^{-\left(\frac{n}{N_2}\right)}$	$= f_{s_{\max}} 10^{-\left(\frac{n}{N_{10}}\right)}$
Count range:	$0 \leq n \leq K$	$0 \leq n \leq K$	$0 \leq n \leq K$	$0 \leq n \leq K$
Total points, $N_T = K + 1$:	$= \frac{\log(R_T)}{\log(R)} + 1$	$= N_e \log_e(R_T) + 1$	$= N_2 \log_2(R_T) + 1$	$= N_{10} \log_{10}(R_T) + 1$
First sample rate, f_{s0}:	$= f_{s_{\max}}$	$= f_{s_{\max}}$	$= f_{s_{\max}}$	$= f_{s_{\max}}$
Last sample rate, f_{sK} or $f_{s_{\min}}$:	$= f_{s_{\max}} R^{-K}$	$= f_{s_{\max}} e^{-\left(\frac{K}{N_e}\right)}$	$= f_{s_{\max}} 2^{-\left(\frac{K}{N_2}\right)}$	$= f_{s_{\max}} 10^{-\left(\frac{K}{N_{10}}\right)}$
Sample rate span, $R_T = \frac{f_{s0}}{f_{sK}} = \frac{f_{s_{\max}}}{f_{s_{\min}}}$:	$= R^K$	$= e^{\frac{K}{N_e}}$	$= 2^{\frac{K}{N_2}}$	$= 10^{\frac{K}{N_{10}}}$
Ratio, R:	$= \frac{f_{s_n}}{f_{s_{n+1}}}$	$= e^{\frac{1}{N_e}}$	$= 2^{\frac{1}{N_2}}$	$= 10^{\frac{1}{N_{10}}}$
Ratio, R in terms of resonator Q, and decay threshold L_{dB}:	$= \frac{8\pi \log_{10}(e)}{L_{dB} Q} + 1$	Not applicable	Not applicable	Not applicable
Sample density: N_e, pts per e: N_2, pts per octave: N_{10}, pts per decade:	Not applicable	$N_e = \frac{1}{\log_e\left(\frac{8\pi \log_{10}(e)}{L_{dB} Q} + 1\right)}$	$N_2 = \frac{1}{\log_2\left(\frac{8\pi \log_{10}(e)}{L_{dB} Q} + 1\right)}$	$N_{10} = \frac{1}{\log_{10}\left(\frac{8\pi \log_{10}(e)}{L_{dB} Q} + 1\right)}$

TABLE 3. GENERAL EQUATIONS FOR LOG SAMPLING IN FREQUENCY DOMAIN

QUANTITY	RATIO	BASE e	BASE 2	BASE 10
Frequency at nth sample, f_n:	$= f_{\min}R^n$	$= f_{\min}e^{\frac{n}{N_e}}$	$= f_{\min}2^{\frac{n}{N_2}}$	$= f_{\min}10^{\frac{n}{N_{10}}}$
Count range:	$0 \le n \le K$	$0 \le n \le K$	$0 \le n \le K$	$0 \le n \le K$
Total points, $N_T = K + 1$:	$= \frac{\log(R_T)}{\log(R)} + 1$	$= N_e \log_e(R_T) + 1$	$= N_2 \log_2(R_T) + 1$	$= N_{10} \log_{10}(R_T) + 1$
First sample frequency, f_0:	$= f_{\min}$	$= f_{\min}$	$= f_{\min}$	$= f_{\min}$
Last sample frequency, f_K or $f_{\max}$:	$= f_{\min}R^K$	$= f_{\min}e^{\frac{K}{N_e}}$	$= f_{\min}2^{\frac{K}{N_2}}$	$= f_{\min}10^{\frac{K}{N_{10}}}$
Frequency span, $R_T = \frac{f_K}{f_0} = \frac{f_{\max}}{f_{\min}}$:	$= R^K$	$= e^{\frac{K}{N_e}}$	$= 2^{\frac{K}{N_2}}$	$= 10^{\frac{K}{N_{10}}}$
Ratio, R:	$= \frac{f_{n+1}}{f_n}$	$= e^{\frac{1}{N_e}}$	$= 2^{\frac{1}{N_2}}$	$= 10^{\frac{1}{N_{10}}}$
Ratio, R in terms of resonator Q, and decay threshold L_{dB}:	$= \frac{8\pi \log_{10}(e)}{L_{dB}Q} + 1$	Not applicable	Not applicable	Not applicable
Sample density: N_e, pts per e: N_2, pts per octave: N_{10}, pts per decade:	Not applicable	$N_e = \frac{1}{\log_e\left(\frac{8\pi \log_{10}(e)}{L_{dB}Q} + 1\right)}$	$N_2 = \frac{1}{\log_2\left(\frac{8\pi \log_{10}(e)}{L_{dB}Q} + 1\right)}$	$N_{10} = \frac{1}{\log_{10}\left(\frac{8\pi \log_{10}(e)}{L_{dB}Q} + 1\right)}$

TABLE 4. LOG SAMPLING DESIGN PROCEDURES

PROCEDURE 1: (Follow down) **GIVEN:** Q, f_{min}, f_{max}, and L_{dB}	**PROCEDURE 2:** (Follow down) **GIVEN:** N_T, f_{min}, f_{max}, and L_{dB}	**WHERE:** f_{min} = minimum frequency in analysis band f_{max} = maximum frequency in analysis band Q = maximum resonator Q L_{dB} = decay threshold in dB N_T = total number of sample points in time or frequency
CALCULATION SEQUENCE: 1. $R = \frac{8\pi \log_{10}(e)}{L_{dB}Q} + 1$ 2. $R_T = \frac{f_{max}}{f_{min}}$ 3. $K = \text{RoundUp}\left[\frac{\log_{10}(R_T)}{\log_{10}(R)}\right]$ 4. $N_T = K + 1$	**CALCULATION SEQUENCE:** 1. $K = N_T - 1$ 2. $R_T = \frac{f_{max}}{f_{min}}$ 3. $R = e^{\frac{\log_e(R_T)}{K}}$ 4. $Q = \frac{8\pi \log_{10}(e)}{(R-1)L_{dB}}$	**WHERE:** R = time or frequency increment ratio (=t_{m+1} / t_m) R_T = time or frequency span (=f_{max}/f_{min} , or t_{max}/t_{min}) K = last count
CALCULATION SEQUENCE CONTINUED: 5. $t_{min} = \frac{0.4}{(R-1)f_{max}}$ 6. $t_{max} = R_T t_{min}$ 7. $N_e = 1/\log_e(R)$ 8. $N_2 = 1/\log_2(R)$ 9. $N_{10} = 1/\log_{10}(R)$ 10. $t_n = t_{min}R^n = t_{min}e^{\frac{n}{N_e}} = t_{min}2^{\frac{n}{N_2}} = t_{min}10^{\frac{n}{N_{10}}}$ 11. $f_n = f_{min}R^n = f_{min}e^{\frac{n}{N_e}} = f_{min}2^{\frac{n}{N_2}} = f_{min}10^{\frac{n}{N_{10}}}$ 12. $f_{S_{max}} = 2.5 f_{max}$ 13. $f_{S_{min}} = 2.5 f_{min}$ 14. $f_{S_n} = f_{S_{max}}R^{-n}$ $= f_{S_{max}}e^{\frac{-n}{N_e}} = f_{S_{max}}2^{\frac{-n}{N_2}} = f_{S_{max}}10^{\frac{-n}{N_{10}}}$ 15. $f_S(t) \approx \frac{1}{(R-1)t}$		**WHERE:** t_{min} = initial (start) sample time t_{max} = final (end) sample time N_e = sample density in time or frequency in number points per e ratio N_2 = sample density in time or frequency in number points per octave N_{10} = sample density in time or frequency in number points per decade n = sample count, $0 \leq n \leq K$ t_n = time at nth sample f_n = frequency at nth sample $f_{S_{max}}$ = maximum sampling rate $f_{S_{min}}$ = minimum sampling rate f_{S_n} = sampling rate at nth sample $f_S(t)$ = sampling rate as a function of time

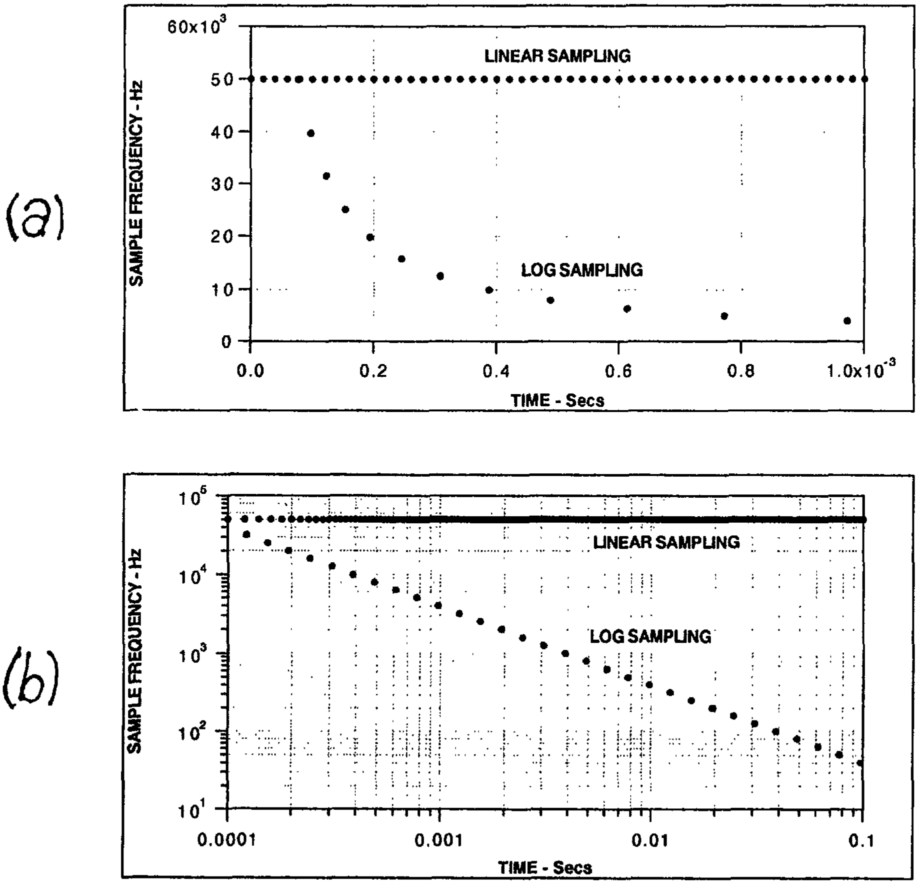


Fig. 1. Linear and logarithmic sampling illustrated by plotting the sample rate at each sample time. (a) Sample rate plotted vs. linear time on a scale of 0 to 1 ms. (b) Sample rate plotted vs. log time on a scale of 0.1 ms to 0.1 s. These graphs show how the instantaneous sample rate changes over time for both linear and log sampling for a specific example set of parameters. The dots indicate the individual sample points. The linear sampling starts at time zero with a constant sample rate of 50 kHz. The log sampling starts with an initial sample rate of 50 kHz, starting at roughly 77.3 us, with a constant 10 points per decade, and falls thereafter following $f_S(t) \approx 3.86/t$.

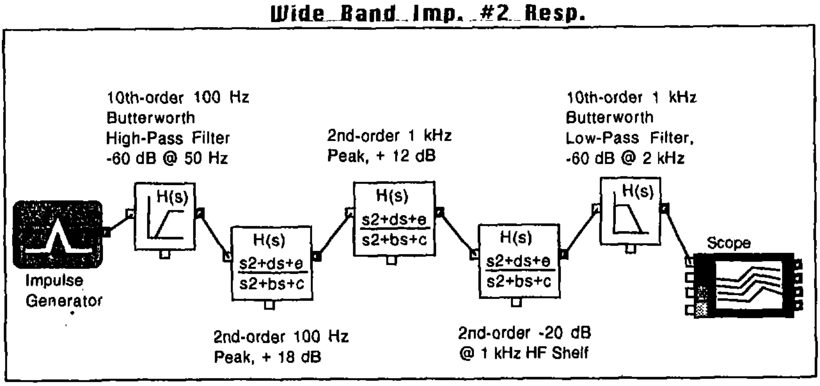


Fig. 2. Block diagram of a wide-band minimum-phase band-pass system used to generate an example impulse response (see Section 1.1 for more detail).

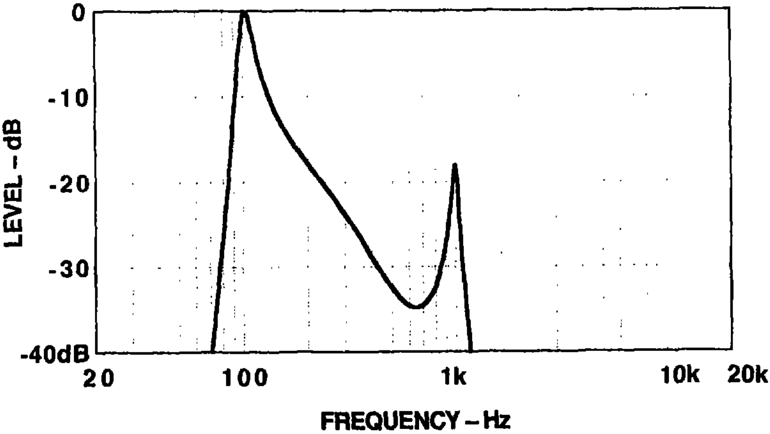


Fig. 3. Frequency response of the system shown in Fig. 2, plotted over a 20 Hz to 20 kHz log scale. The response rolls off very rapidly below 100 Hz and above 1 kHz, and has high Q peaks at both these frequencies.

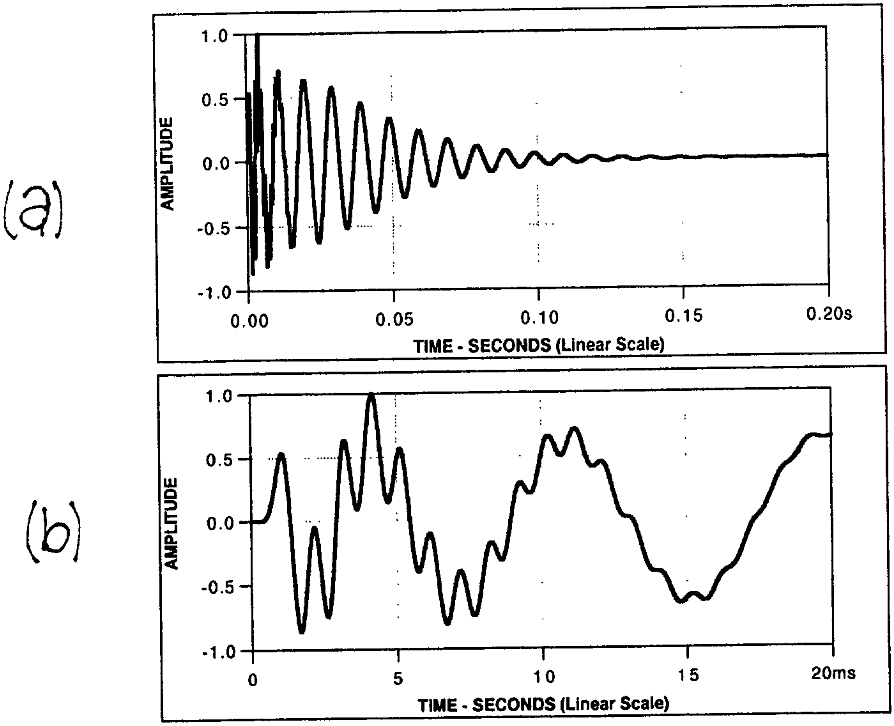


Fig. 4. Impulse response of the system shown in Fig. 2 plotted on linear time scales. (a) 0 to 0.2 s linear time scale. (b) 0 to 0.02 s expanded (x10) linear time scale. Note that both these scales are required to show all the details of the response.

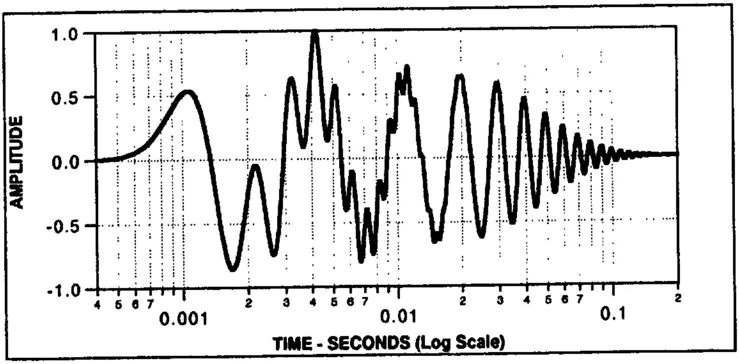


Fig. 5. Impulse response of the system shown in Fig. 2 plotted on a logarithmic nearly three-decade range of 400 us to 0.2 s time scale. Note that all the significant details of the response are shown on one logarithmic scale.

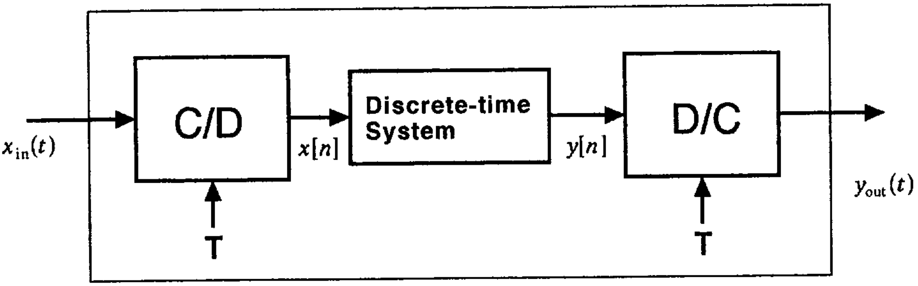


Fig. 6. General block diagram of a device that accomplishes discrete-time processing of continuous-time signals. The input continuous-time signal is first sampled, then processed by a discrete-time processing system, and then reconverted to a continuous-time signal by a reconstruction process.

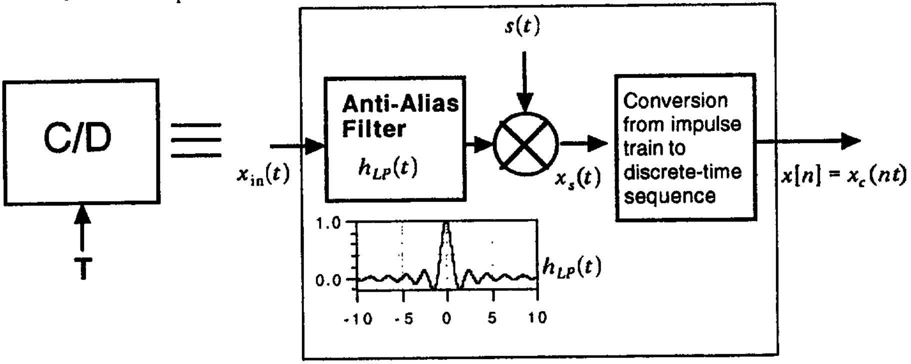
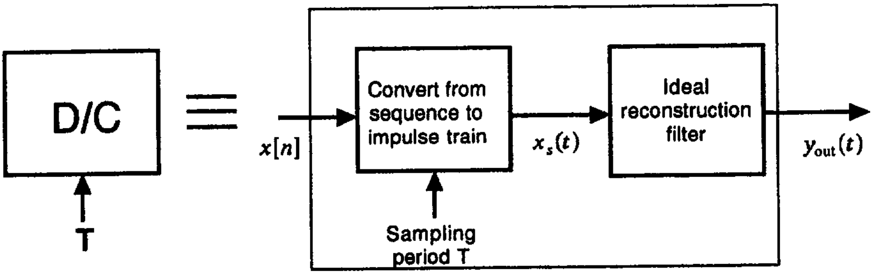


Fig. 7. Internal details of the continuous to discrete (C/D) converter block of Fig. 6. A perfect brick-wall low-pass anti-alias filter, set to half the sample rate, is followed by sampler that converts the input continuous-time signal to a sequence of numbers. The sampler first converts the signal to a series of weighted impulses and then converts the impulses to a sequence of numbers. The impulse response of the anti-alias filter, a sinc(t/T) function, is shown.



D/C
T
$x[n]$
Convert from sequence to impulse train
Sampling period T
$x_s(t)$
Ideal reconstruction filter
$y_{out}(t)$

Fig. 8. Internal details of the ideal discrete-time to continuous-time D/C conversion block of Fig. 6. The discrete number sequence is first converted to a sequence of weighted impulses, which is then sent to a reconstruction filter which forms the continuous-time output. The ideal reconstruction filter is exactly the same as the anti-alias filter of Fig. 7. The filter's sinc(t/T) impulse response optimally interpolates between the samples.

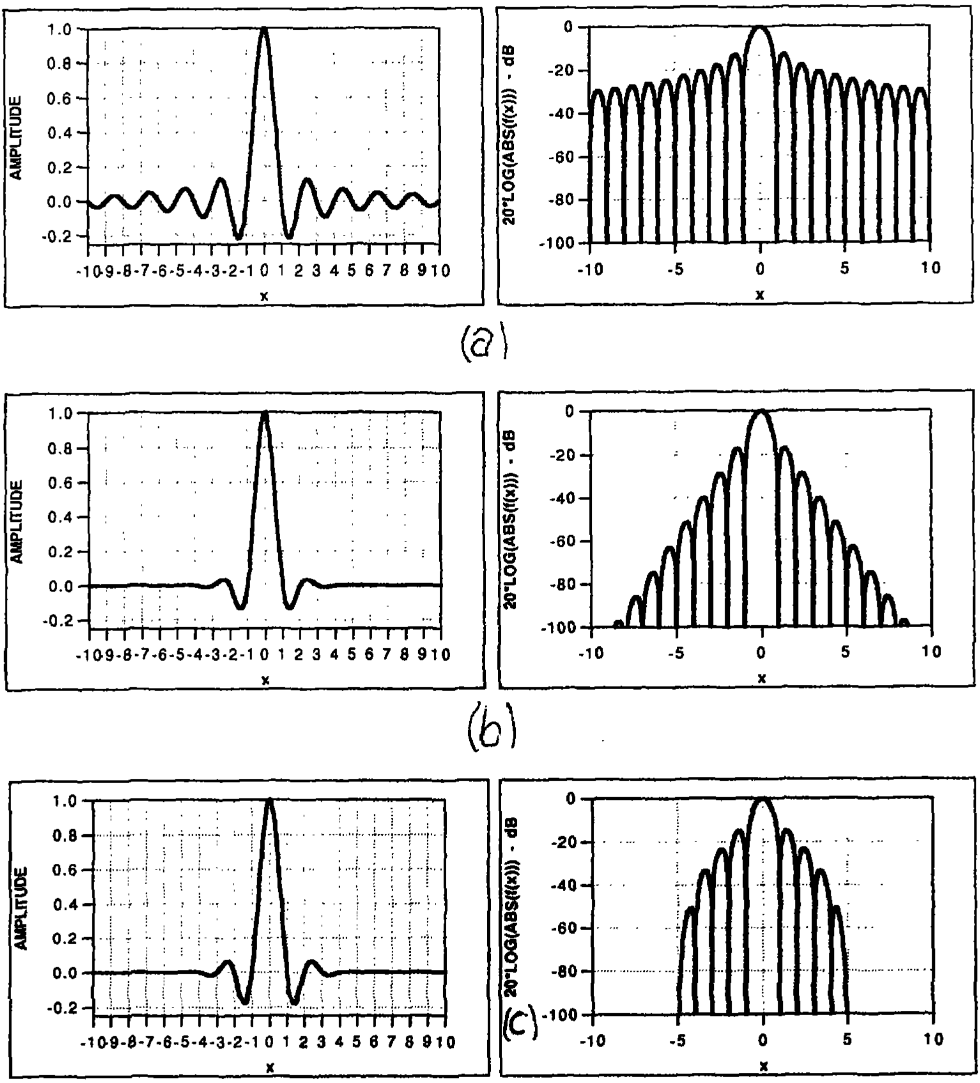


Fig. 9. Plots of three different sampling or interpolation functions. Both linear (left) and log (right) vertical amplitude scales are shown. (a) The sinc(x) function. (b) The cardinal cubic-spline function $\eta^3(x)$. (c) The Hannsinc(x) function of width 10, which is a Hann windowed sinc(x) function. The interpolation and sampling properties of these functions stems from the fact that they are precisely equal to one at the origin and zero at all other integer values.

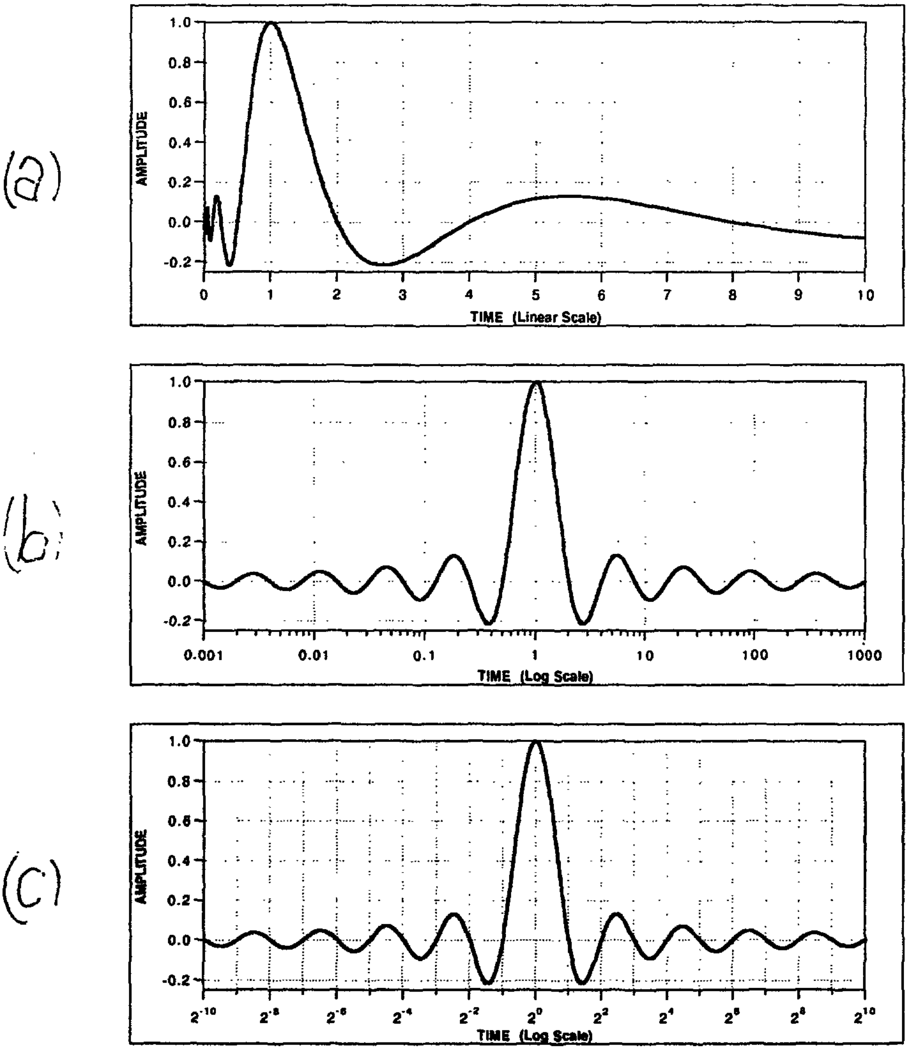


Fig. 10. Plot of a one-point-per-octave logsinc interpolation function, centered at one second, on linear and log time scales. (a) Linear time scale of 0 to 10 s. (b) Base ten log time scale running from 1 ms to 1000 s. (c) Base two log time scale running from 2^{-10} s to 2^{10} s. This function optimally interpolates a series of log-spaced samples whose successive sample times are in the ratio of two, i.e. 5, 10, 20, 40, 80 us, etc. Note that this function is unity at one, and has zero crossings at every multiple or sub-multiple of two. Note also that the logsinc's zeros occur at a series of geometric-spaced time values, rather than the sinc function's equal-spaced time values.

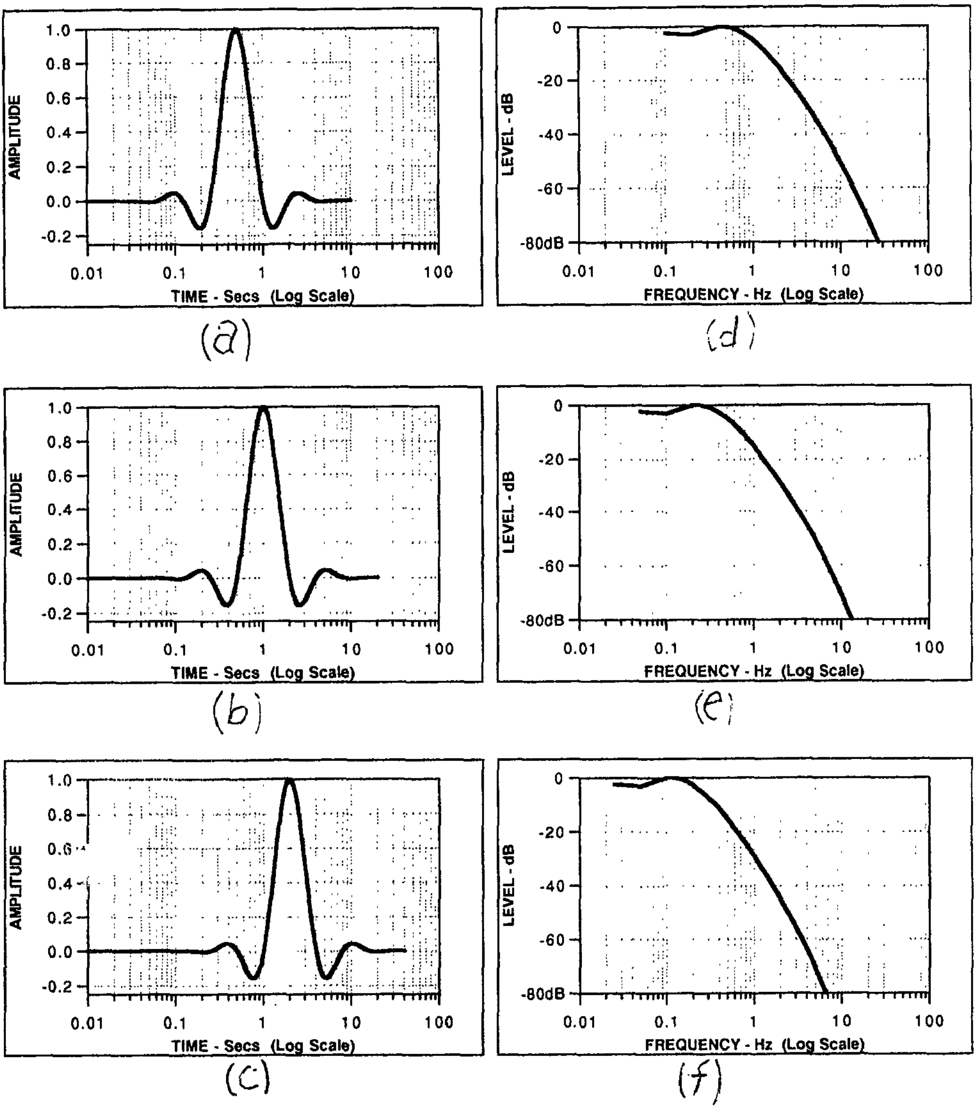


Fig. 11. Plots of a one-point-per-octave width-six LogHannsinc interpolator with several different values of center time, plotted on log time scales, with corresponding magnitude spectrums plotted on log frequency scales. (a) LogHannsinc centered at 0.5 s. (b) LogHannsinc centered at 1 s. (c) LogHannsinc centered at 2 s. (d) Spectrum of LogHannsinc centered at 0.5 s. (e) Spectrum of LogHannsinc centered at 1 s. (f) Spectrum of LogHannsinc centered at 2 s. Note that as the function shifts up in time, the spectrum shifts down in frequency by the same ratio, and vice versa.

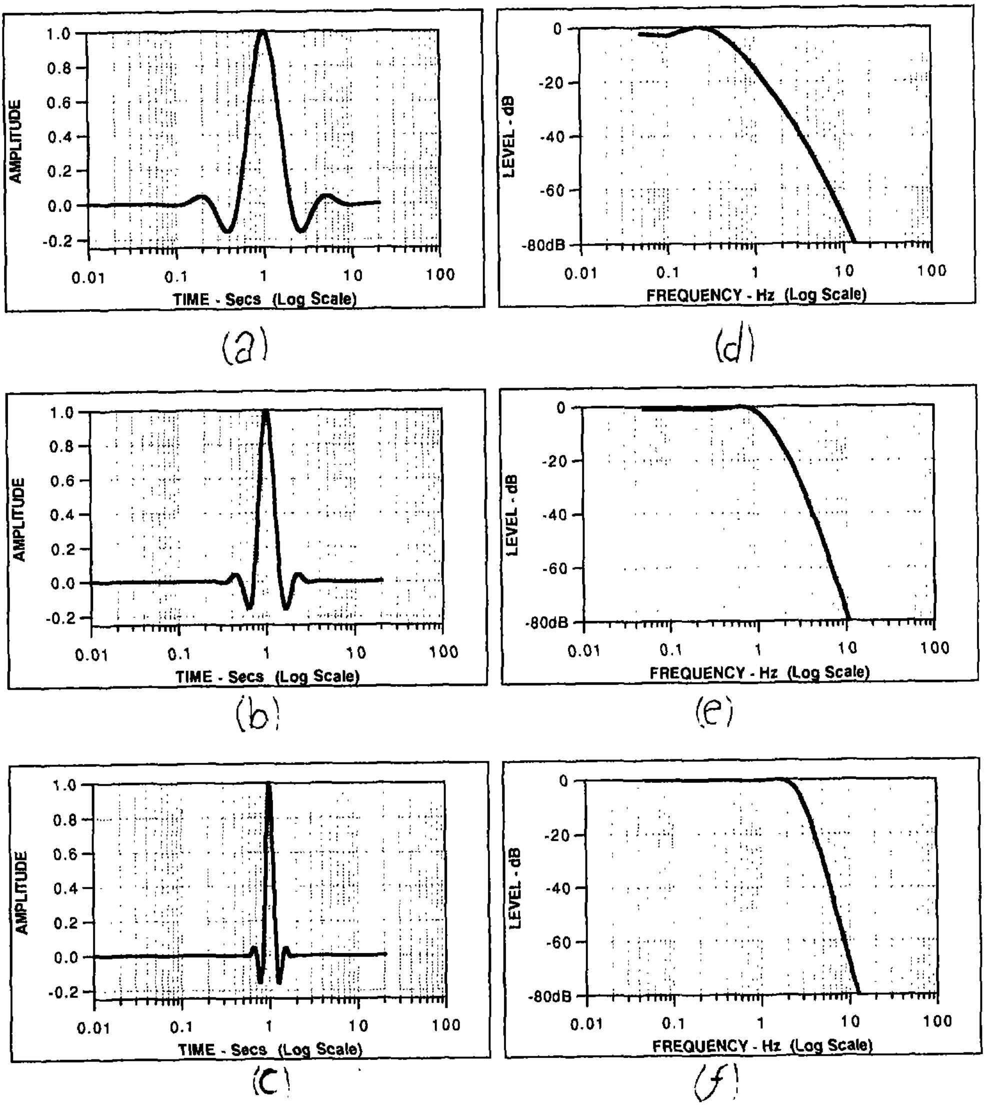


Fig. 12. Plots of several different width-six LogHannsinc interpolator functions, all centered at 1 s, with different sample densities, plotted on log time scales. Associated magnitude spectrums are shown. (a) A one-point-per-octave LogHannsinc. (b) A two-points-per-octave LogHannsinc. (c) A four-points-per-octave LogHannsinc. (d) Spectrum of a one-point-per-octave LogHannsinc. (e) Spectrum of a two-points-per-octave LogHannsinc. (f) Spectrum of a four-points-per-octave LogHannsinc. Note that as the sample density increases, the spectrum sharpens and shifts up in frequency.

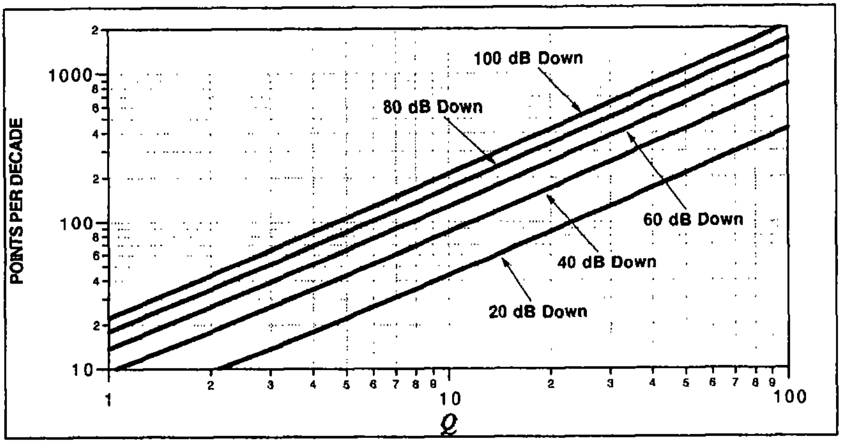


Fig. 13. Plots of Eq. (29) giving the required log sample density, in points per decade, versus *Q* for various decay thresholds. This graph yields the log sample density required to properly digitize the impulse response of a system whose resonator's Q is no more than that given.

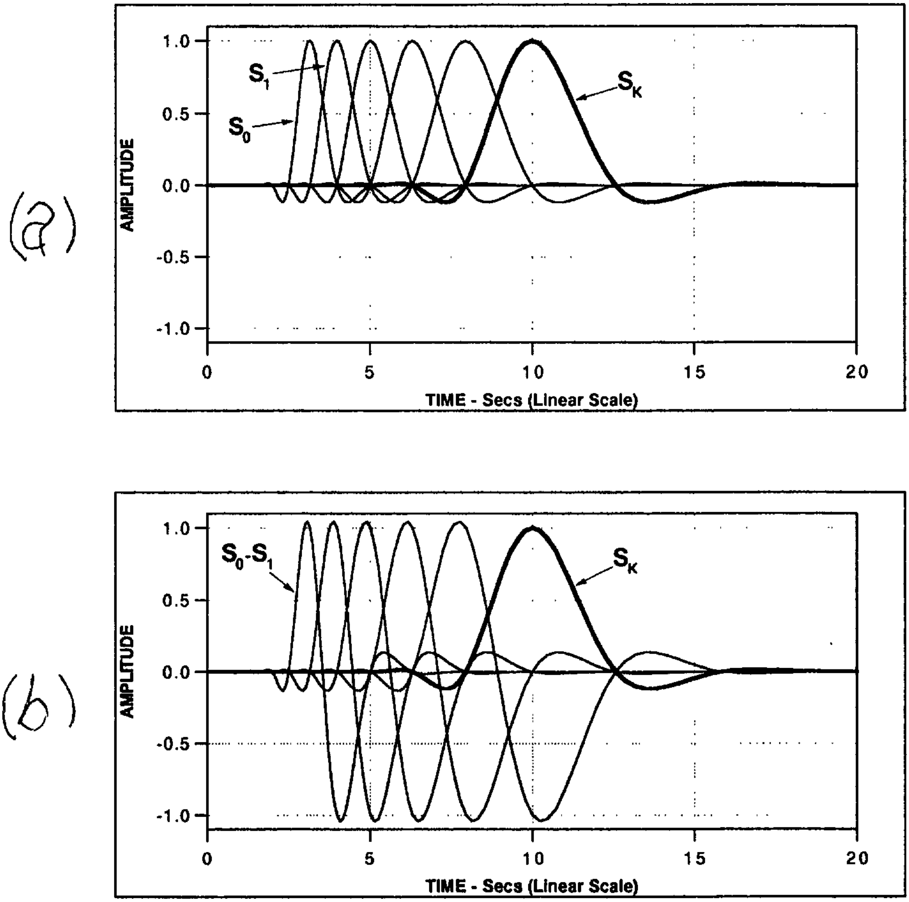
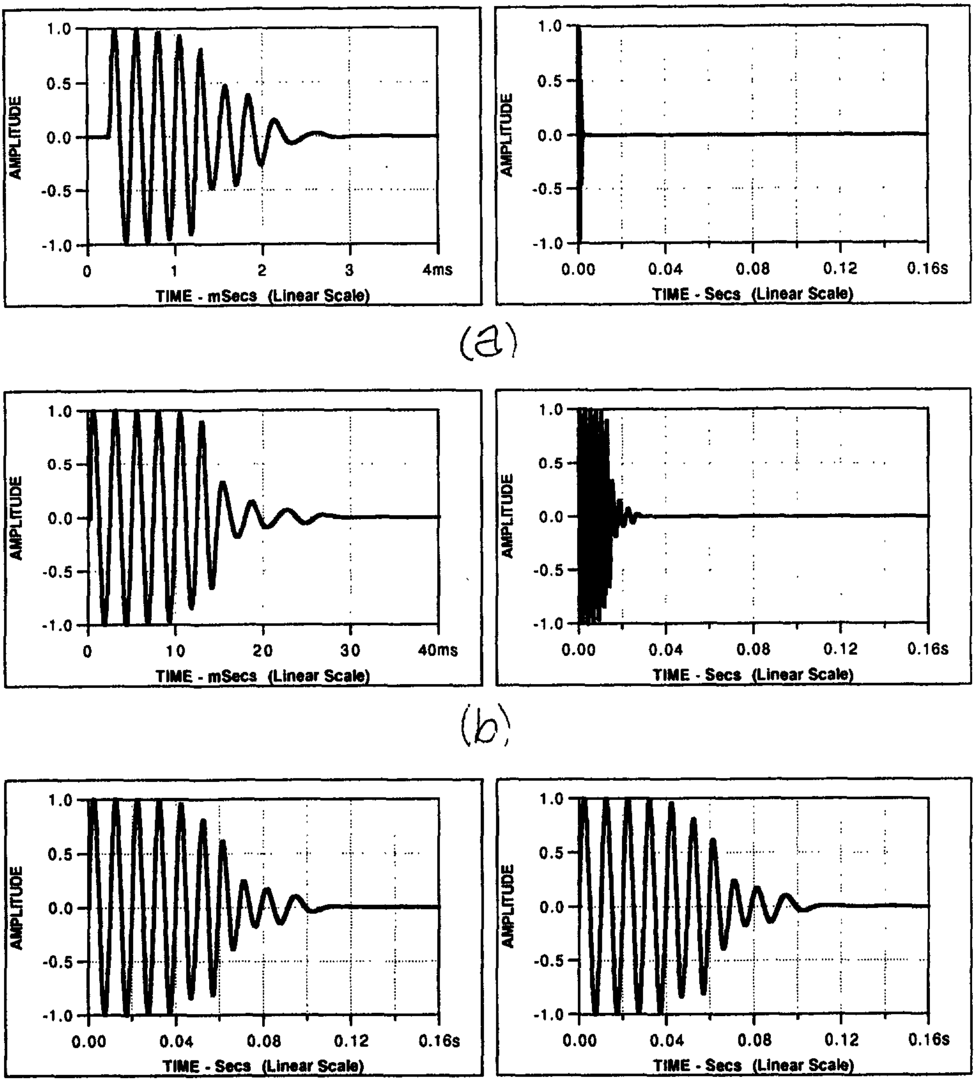


Fig. 14. Graphs of the two different types of log decomposition waveforms sets, plotted using width-six three-points-per-octave LogHannsincs, on linear time scales. (a) The fundamental set, based on log-shifted low-pass interpolation functions. (b) The differenced set, based on a single high-time (low-frequency) low-pass interpolation function, and several lower-time (higher-frequency) bandpass interpolations functions. The differenced set provides a set of functions which are localized both in time and frequency. The members of the differenced set are composed of differences of consecutive members of the first set.



(a)

(b)

(c)

Fig. 15. Plots of log-spaced decomposition and reconstructions of three sine waves of various frequencies, with a fixed sample density of 32.78 points per decade, 90 total sample points, and a starting sample rate of 50 kHz. In each pair of linear-scale plots, the full time scale is on the right (0 to 0.16 s), and an expanded time scale is on the left. (a) A 4 kHz sine wave. (b) A 400 Hz sine wave. (c) A 100 Hz sine wave. A width-18 log-warped cubic spline function was used as the decomposition and reconstruction interpolator. The sample density was selected assuming a 60 dB decay threshold and a system Q of 2.5. Note that as frequency decreases (top to bottom), the reconstructed sine wave extends farther out in time.

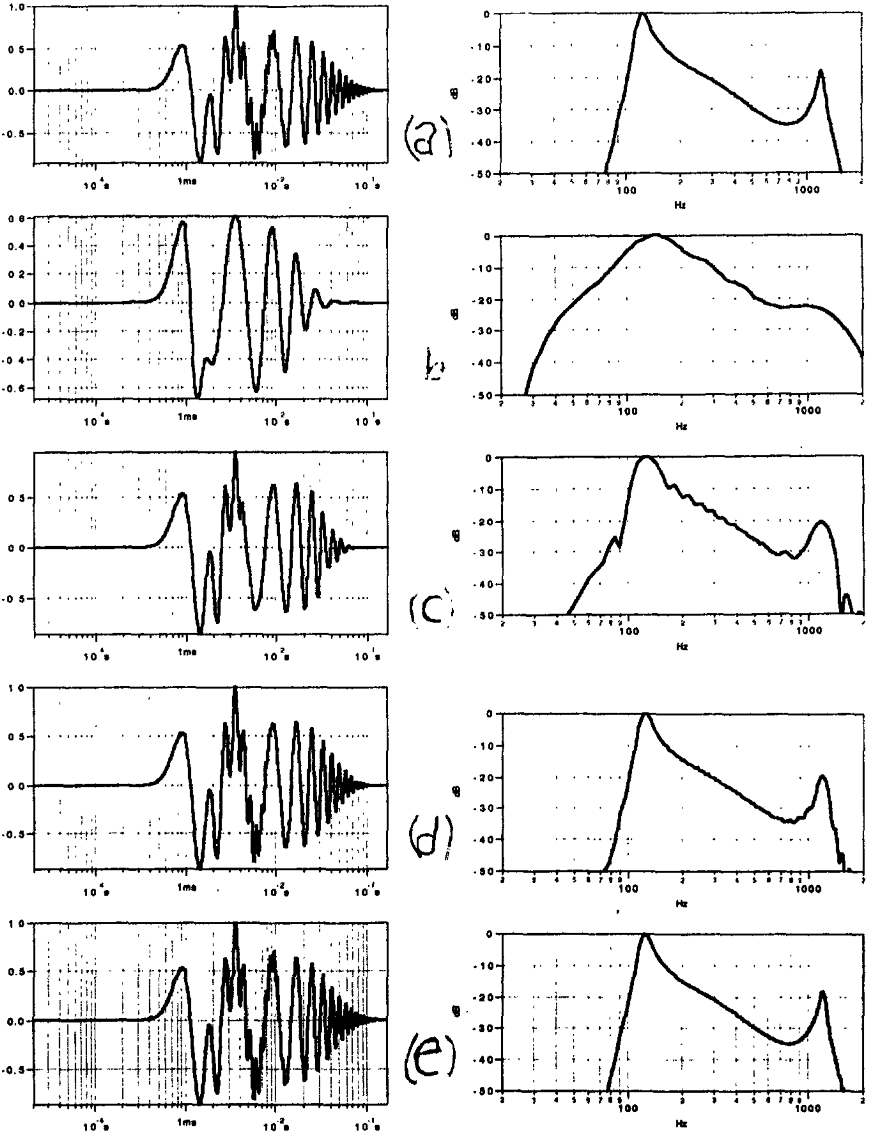


Fig. 16. The impulse response of a model similar to the one described in Section 1.1 was log decomposed and reconstructed for several different values of sample density and total number of sample points. In each pair of plots, the reconstructed impulse response is shown on the left plotted on a log time scale of 20 us to 0.2 s, while the spectrum magnitude of the reconstructed impulse response is shown on the right plotted on a log frequency scale of 20 Hz to 2 kHz. A width-18 log-warped cubic spline function was used as the decomposition and reconstruction interpolator. (a) Original impulse response and spectrum. (b) Sample density of 10 points/decade with 30 sample points. (c) Sample density of 25 points/decade with 75 sample points. (d) Sample density of 50 points/decade with 150 sample points. (e) Sample density of 100 points/decade with 300 sample points. Note that a sample density of 100 points/decade is required for best accuracy.

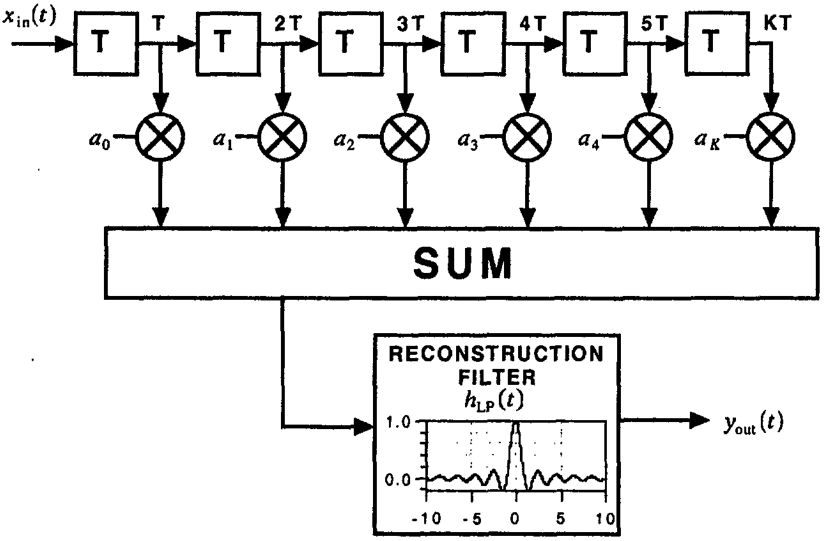


Fig. 17. Block diagram of a conventional continuous-time FIR filter/convolver with equal-spaced taps. Each delay block is identical and provides a delay T. Each tap is multiplied by a coefficient a_n, where n is in the range 0 to K. Only one reconstruction filter is required at the output.

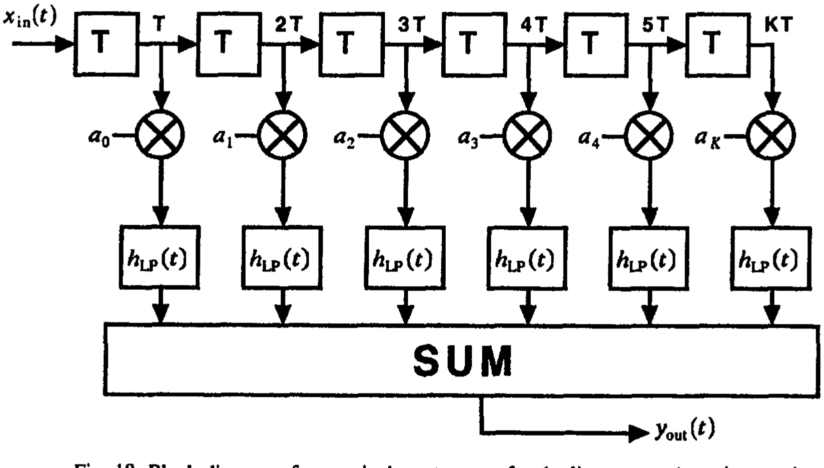


Fig. 18. Block diagram of an equivalent structure for the linear-spaced continuous-time convolver of Fig. 17, called the cascade structure, with individual identical reconstruction filters on each tap.

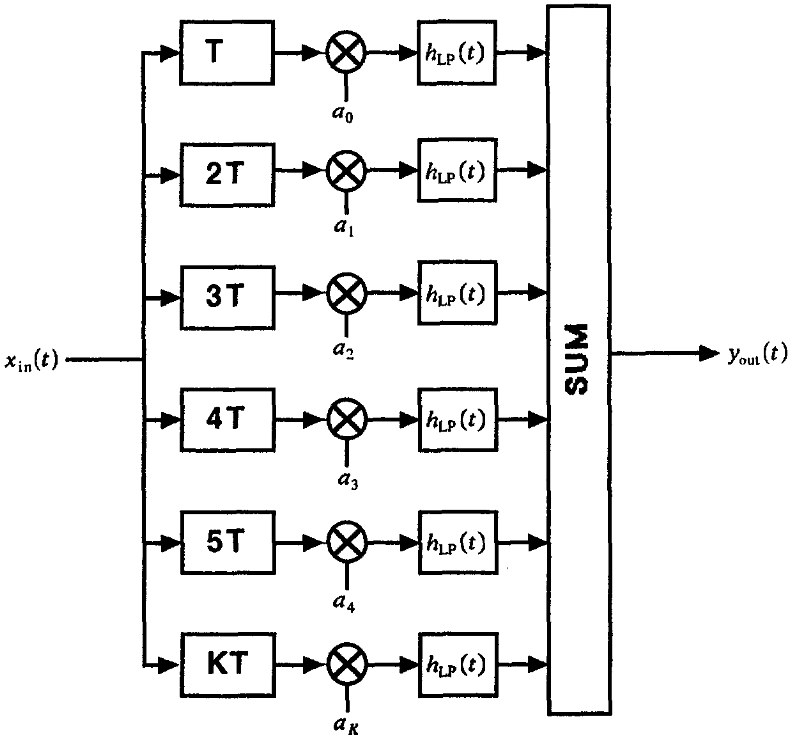


Fig. 19. Block diagram of an equivalent structure for the linear-spaced convolver of Fig. 18, called the parallel structure. Here, several processing channels are in parallel, each with its own appropriate delay and reconstruction filter.

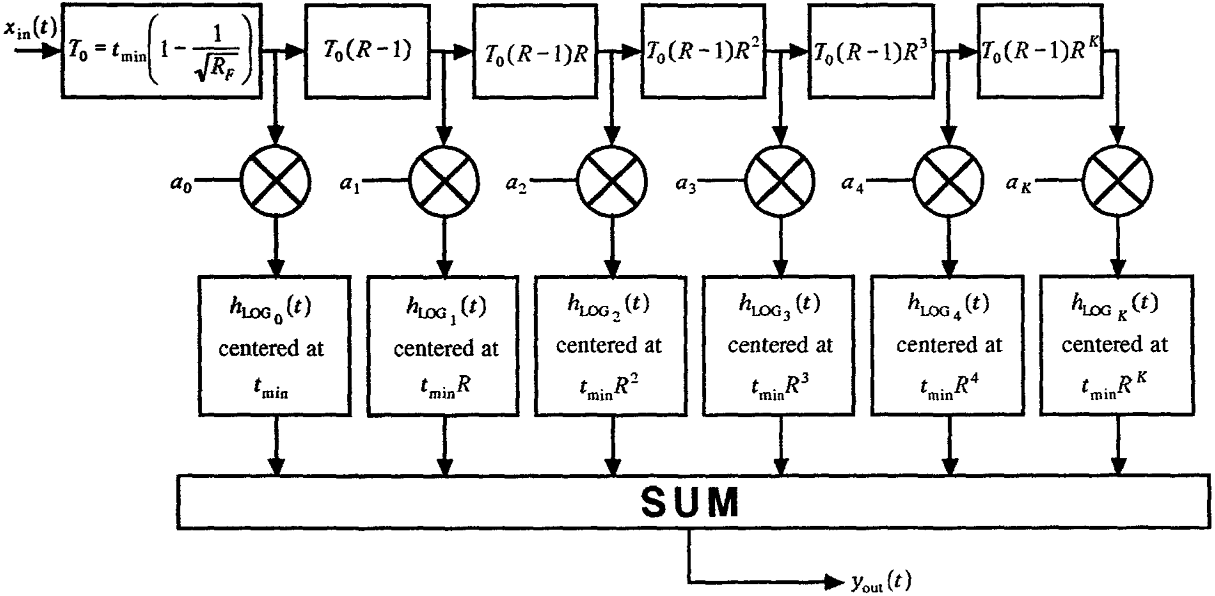

$x_{in}(t)$
$T_0 = t_{min}\left(1 - \frac{1}{\sqrt{R_F}}\right)$
$T_0(R-1)$
$T_0(R-1)R$
$T_0(R-1)R^2$
$T_0(R-1)R^3$
$T_0(R-1)R^K$
a_0
a_1
a_2
a_3
a_4
a_K
$h_{LOG_0}(t)$ centered at t_{min}
$h_{LOG_1}(t)$ centered at $t_{min}R$
$h_{LOG_2}(t)$ centered at $t_{min}R^2$
$h_{LOG_3}(t)$ centered at $t_{min}R^3$
$h_{LOG_4}(t)$ centered at $t_{min}R^4$
$h_{LOG_K}(t)$ centered at $t_{min}R^K$
SUM
$y_{out}(t)$

Fig. 20. Block diagram of the cascade-structure continuous-time log-spaced convolver. Here each delay block provides a delay which increases on a geometric basis going from left to right. Each reconstruction filter $h_{LOG_n}(t)$ is distinct and provides a different filter function for each tap. These filters provides an interpolation function which is a log-warped version of the impulse response of the standard reconstruction filter.

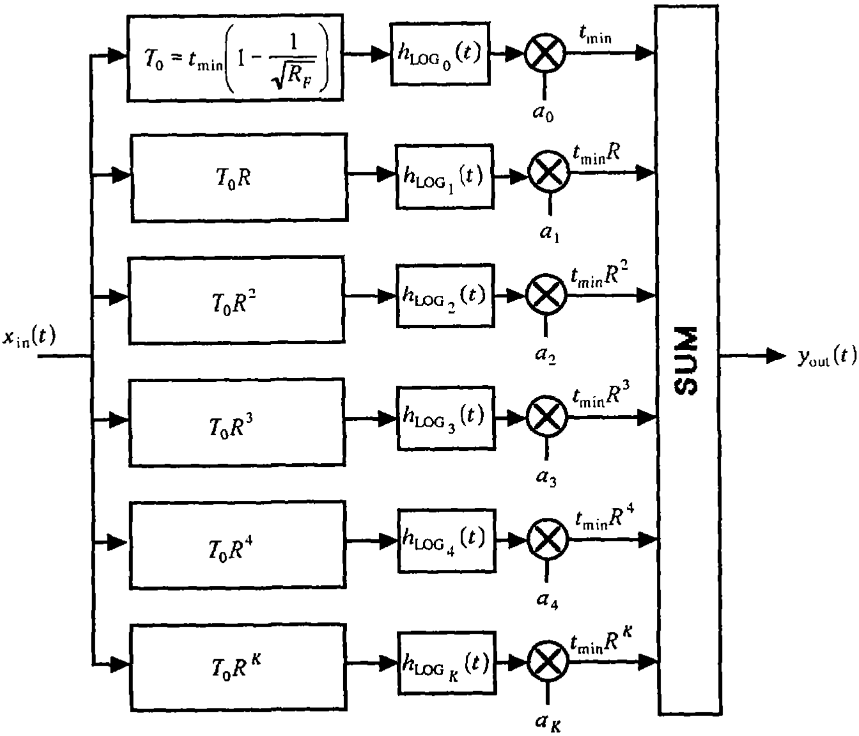


Fig. 21. Block diagram of the parallel-structure continuous-time log-spaced convolver. Here each channel has its own appropriate delay and log reconstruction filter.

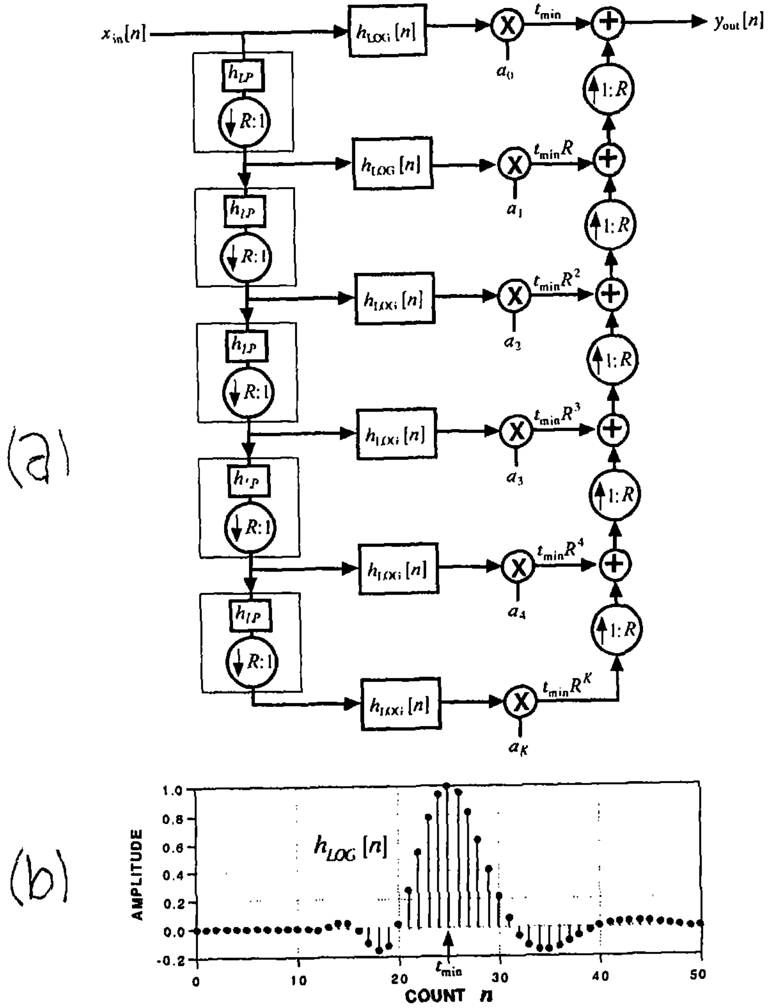


Fig. 22. Block diagram of a discrete-time log-spaced convolver using multi-rate sampie techniques. (a) Block diagram. (b) Discrete-time impulse response of each reconstruction filter block $h_{LOG}[n]$. Each of the up- and down-arrow circular blocks provide fractional up and down sampling respectively, with rate ratio R. Each rectangular block on the left contains a low-pass filter followed by a down sampler. Note that each channel of this convolver is *identical*. The required time scaling of each channel is accomplished automatically by the different sample rates.

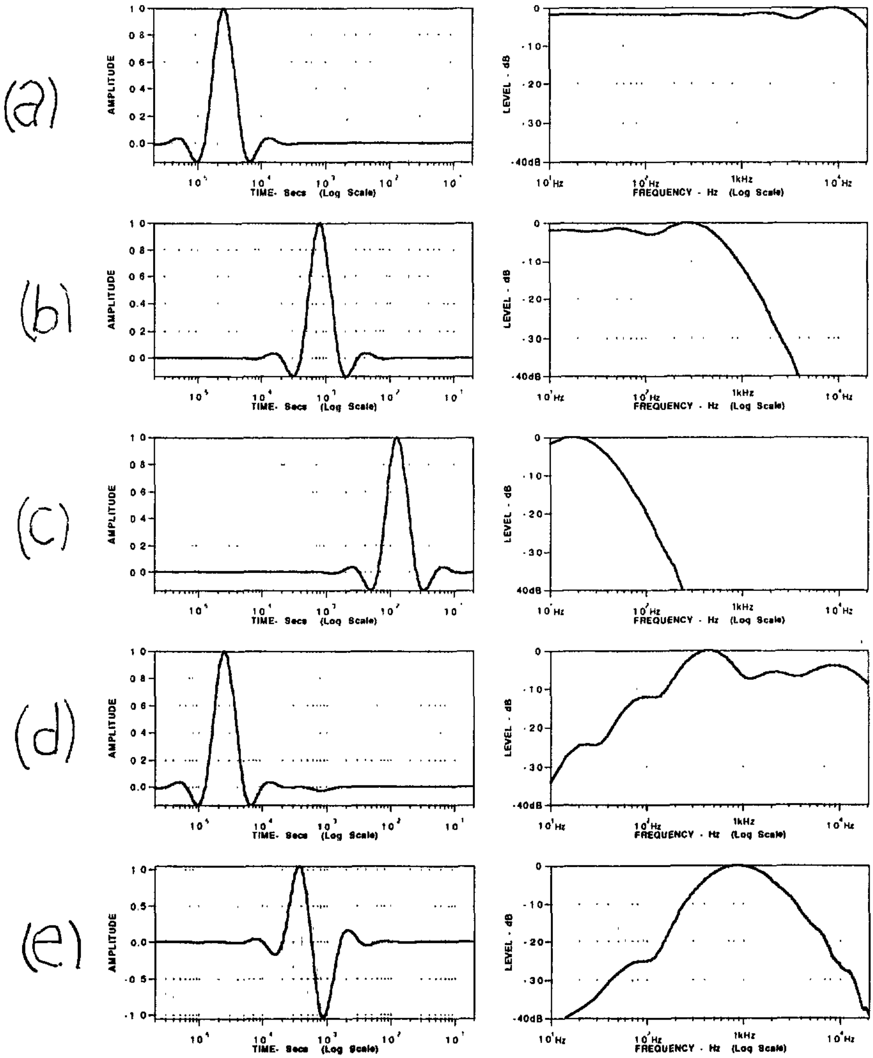


Fig. 23. Various impulse and frequency responses provided by a ten-tap octave and a 31-tap one-third-octave log-spaced convolvers. The left graph of each pair shows the convolvers impulse response plotted on a five-decade log-time scale of 2 us to 0.2 s. The right graph shows the magnitude frequency response of the convolver plotted on a 10 Hz to 20 kHz log-frequency scale. See last page for details of each graph pair.

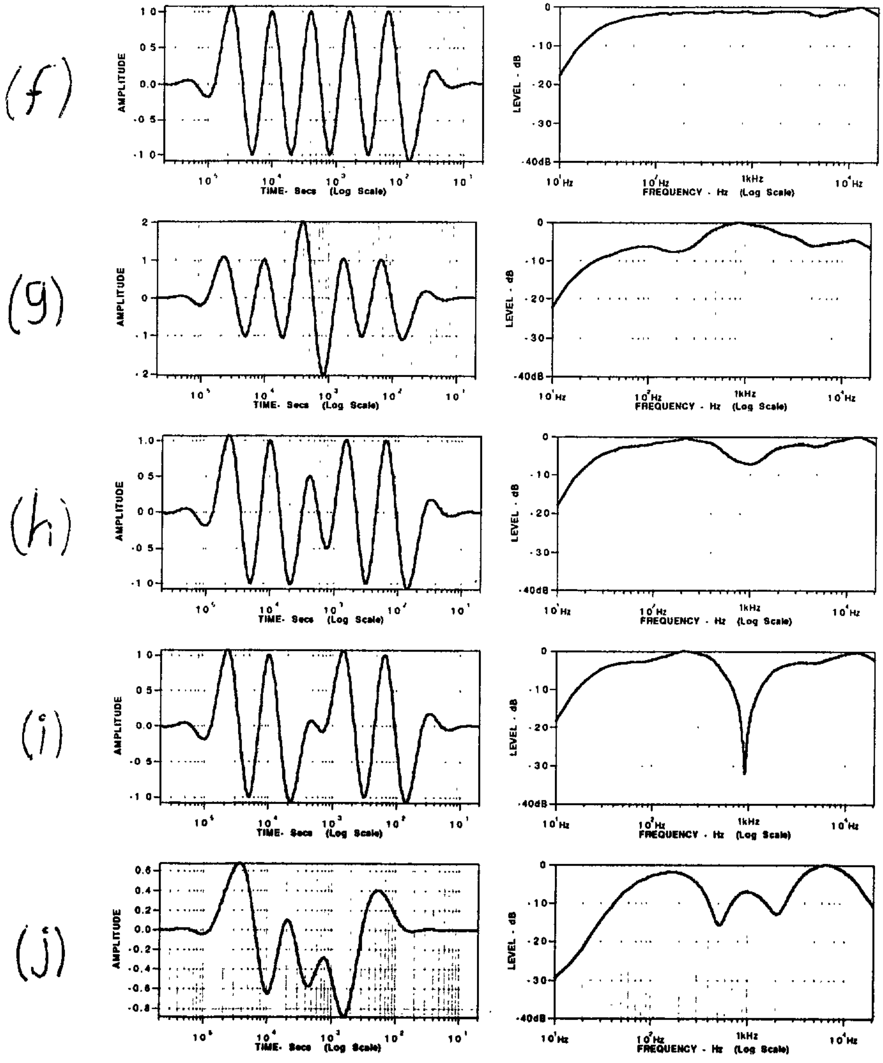


Fig. 23 Continued. See last page for details of each graph pair.

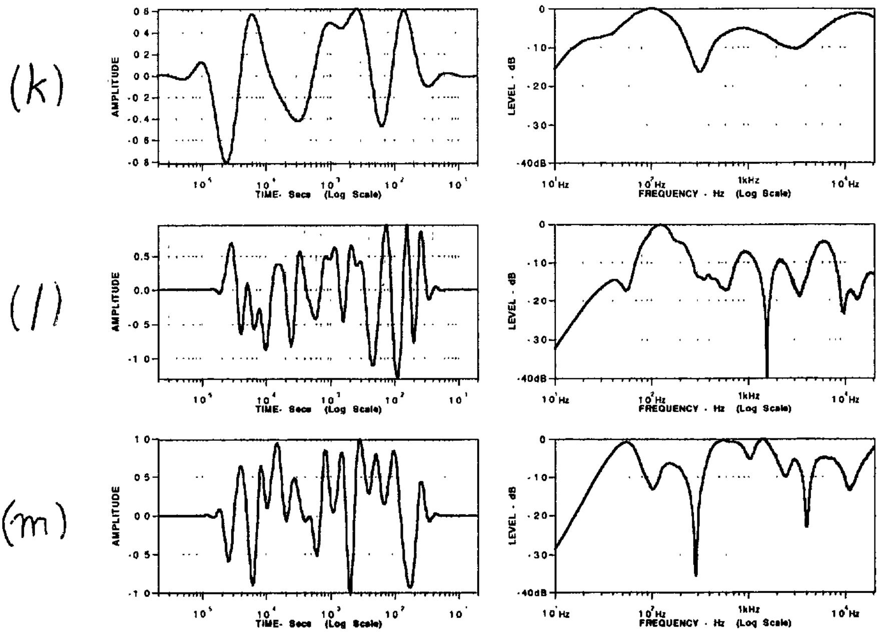


Fig. 23 Continued. See Section 8.2 for details of convolvers. The following table lists the convolvers coefficient values along with a brief description of the response.

FIG.	NUM.	COEFFICIENTS	COEFFICIENTS	DESCRIPTION
	TAPS	NORMAL	DIFFERENCED	OF
		a_0 to a_K	d_0 to d_K	RESPONSE
(a)	10	1,0,0,0,0,0,0,0,0,0	1,1,1,1,1,1,1,1,1,1	Widest flat response.
(b)	10	0,0,0,0,0,1,0,0,0,0	0,0,0,0,0,1,1,1,1,1	Mid-frequency low pass
(c)	10	0,0,0,0,0,0,0,0,0,1	0,0,0,0,0,0,0,0,0,1	Low-frequency low pass
(d)	10	1,0,0,0,0, -0.03125,0,0,0,0	1,1,1,1,1,0.96875, and on	High-pass
(e)	10	0,0,0,0,1,-1,0,0,0,0	0,0,0,0,1,0,0,0,0,0	Mid-frequency band pass
(f)	10	1,-1,1,-1,1,-1,1, -1,1,-1	1,0,1,0,1,0,1,0,1,0	Wide flat response. Log down sweep in time.
(g)	10	1,-1,1,-1,2,-2,1, -1,1,-1	1,0,1,0,2,0,1,0,1,0	+6 dB mid-frequency peak
(h)	10	1,-1,1,-1,0.5,-0.5,1, -1,1,-1	1,0,1,0,0.5,0,1,0,1,0	-6 dB mid-frequency dip
(i)	10	1,-1,1,-1,0,0,1,-1,1, -1	1,0,1,0,0,0,1,0,1,0	Large high-Q mid-frequency dip
(j)	10	Random set		Coefficients chosen at random
(k)	10	Random set		Coefficients chosen at random
(l)	31	Random set		Coefficients chosen at random
(m)	31	Random set		Coefficients chosen at random

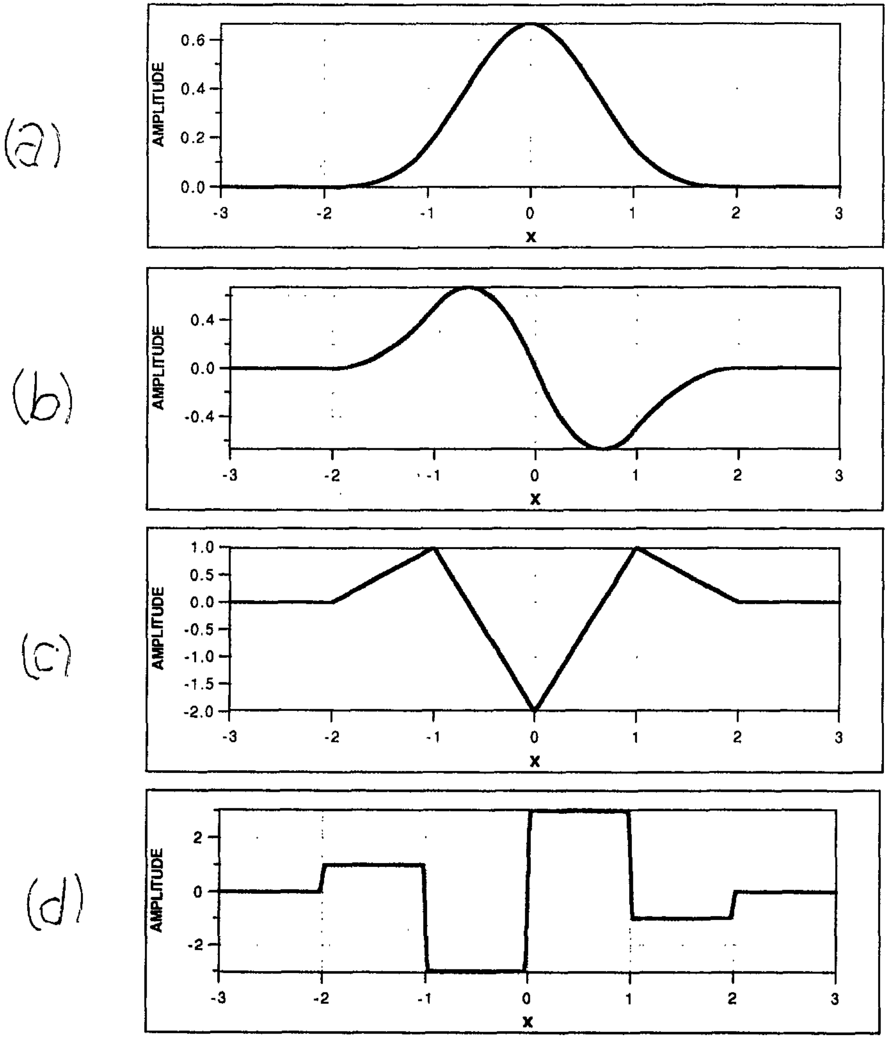


Fig. 24. Plot of third-order B spline $\beta^3(x)$ along with derivatives. (a) $\beta^3(x)$. (b) First derivative of $\beta^3(x)$. (c) Second derivative of $\beta^3(x)$. (d) Third derivative of $\beta^3(x)$.