Discrete Representation of Signals

ALAN V. OPPENHEIM, SENIOR MEMBER, IEEE, AND DONALD H. JOHNSON

Abstract—**In processing continuous-time signals by digital means, it is necessary to represent the signal by a digital sequence. There are many ways other than periodic sampling for obtaining such a sequence. The requirements for such representations and some examples are discussed within the framework of simulating linear time-invariant systems. The representation of digital sequences by other digital sequences is also discussed, with particular emphasis on the use of such representations to implement a nonlinear warping of the digital frequency axis. Some applications and hardware implementation of this digital-frequency warping are described.**

I. Introduction

WITH THE increasing speed and decreasing cost of digital hardware, digital methods are playing an important role in signal processing. In some applications, the input and output are continuous-time signals but the processing is digital. In such cases, the input must first be represented by a sequence; after this sequence is processed, the output sequence must then be reconverted to a continuous-time signal. In other instances (such as spectral analysis), the objective of the signal processing is a set of measurements on a continuous-time signal; the input must be converted to a digital sequence but it is unnecessary to convert back to a continuous-time signal. There is, of course, a variety of applications in which the signals to be processed are inherently discrete signals. If conceptually useful, they can be viewed as representing continuous-time signals; however, this often introduces additional complications that may not be warranted. An example of such a signal is arithmetic roundoff noise or limit cycles generated in a digital filter.

When the signal processing involves a continuous-time input or output, the continuous-time signal must be represented by a discrete-time signal, i.e., a sequence. The most common procedure when the continuous-time signal is bandlimited is to choose the sequence values to be samples of the continuous-time function equally spaced in time. This representation is commonly referred to as periodic sampling. There are, however, many ways other than periodic sampling in which a continuous-time function can be represented by a sequence.

In this paper a class of representations of continuous-time functions by sequences is discussed. In addition, a parallel notion is developed—the representation of a sequence by other sequences. Many of the results presented appear elsewhere by the present authors and by others. Thus to some extent this paper can be considered as a review of these results. A major objective is to provide this review within a common framework and, toward this end, results of other authors are combined with our own results and point of view.

Manuscript received November 12, 1971; revised March 13, 1972. This work was supported in part by the Joint Services Electronics Program under Contract DAAB07-71-C-0300, in part by the Advanced Research Projects Agency under Lincoln Laboratory Purchase Order CC-570, and in part by the National Science Foundation under Grant GK-31353.

The authors are with the Department of Electrical Engineering and Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, Mass. 02139.

In Section II, a discussion of the discrete representation of continuous-time signals for the digital simulation of linear time-invariant continuous-time filters is presented, and several examples of such representations are discussed. In Section III, discrete representation of discrete-time signals is discussed, corresponding to the represention of one digital sequence by another digital sequence. The development of the representation of discrete-time signals parallels closely the development of the representation of continuous-time signals. For the representation of continuous-time signals, it is shown that requiring the representation to map linear time-invariant continuous-time systems to linear shift-invariant discrete systems is equivalent to requiring that the representation map the Laplace transform of the continuous-time signals to the z-transform of the discrete-time signal by a substitution of variables. Consequently, in addition to its application to simulation, such a representation is useful in instances such as spectral analysis where it is desirable for the analog frequency axis to map to the digital-frequency axis.

In the representation of discrete-time signals by other discrete-time signals, the notion of mapping the z-transform by a substitution of variables assumes the major emphasis. In particular, it is possible to maintain the *form* of the spectrum for a discrete-time signal but transform the frequency axis in a nonlinear manner. The ability to do this has potential application in a number of contexts, such as unequal resolution and vernier spectral analysis, and the correction of the frequency distortion inherent in divers' speech due to the effect of pressure and the content of their breathing mixture.

II. Discrete Representation of Continuous-Time Signals

In this section, the discussion is directed toward the simulation of signal processing techniques which can be carried out with a linear time-invariant filter. While there are many examples of signal processing which do not fall into this category, linear filtering does represent a wide class of signal-processing problems. Just as linear time-invariant filtering plays an important role in continuous-time signal processing, linear shift-invariant digital filtering plays an important role in digital signal processing. The reason for this lies partly in the fact that linear time-invariant systems, either in the continuous-time or in the discrete-time cases, are analytically manageable and consequently much insight into processing signals with such systems has been gained. For this reason it is desirable to choose a discrete representation for continuous-time signals which will permit linear time-invariant systems to be implemented digitally with linear shift-invariant discrete systems. This is similar to the condition imposed by Steiglitz in demonstrating the isomorphism between digital and analog signal processing [1].

A linear time-invariant filter is conveniently characterized in terms of the convolution integral so that if $x(t)$, $y(t)$, and $h(t)$ are the system input, output, and impulse response, respectively, then

$$g(t) = \int_{-\infty}^{+\infty} f(\tau)h(t-\tau)d\tau \triangleq f(t) \otimes h(t) \tag{1}$$

where $\otimes$ is used to denote continuous-time convolution. Let us denote the discrete representation of $f(t)$, $g(t)$, and $h(t)$ by f_n, g_n, and h_n, respectively. We want f_n, g_n, and h_n to correspond to the input, output, and unit sample response of a discrete linear shift-invariant system, so that g_n must be the discrete convolution of f_n and h_n:

$$g_n = \sum_{k=-\infty}^{+\infty} f_k h_{n-k} \triangleq f_n * h_n \tag{2}$$

where $*$ is used to denote discrete convolution.

The validity of (2) for any f_n and h_n implies that the representation of a continuous-time function by a sequence must be linear, i.e., if x_n and y_n are the representations of $x(t)$ and $y(t)$, then x_n+cy_n is the representation of $x(t)+cy(t)$ where c is an arbitrary constant. This "linearity property" can be derived as follows. Let f_n denote the discrete representation of $f(t)=x(t)+cy(t)$. Then from (1) and (2),

$$g(t) = f(t) \otimes h(t)$$
$$g_n = f_n * h_n$$

since

$$f(t) = x(t) + cx(t),$$
$$f(t) = x(t) \otimes h(t) + cy(t) \otimes h(t)$$

so that

$$f_n = x_n * h_n + cy_n * h_n$$
$$= [x_n + cy_n] * h_n. \tag{3}$$

Since we require (3) to hold for any h_n, it follows that

$$f_n = x_n + cy_n.$$

Now let us express an arbitrary sequence f_n as a linear combination of weighted delayed unit samples:

$$f_n = \sum_{k=-\infty}^{+\infty} f_k \delta_{n-k}$$

where δ_{n-k} represents a sequence whose values are zero except for $n=k$ so that $\delta_0=1$. Let $\phi_k(t)$ denote the continuous-time function represented by the sequence δ_{n-k}. Then by virtue of the linearity property previously derived,

$$f(t) = \sum_{k=-\infty}^{+\infty} f_k \phi_k(t)$$

or, for convenience, changing the summation index,

$$f(t) = \sum_{n=-\infty}^{+\infty} f_n \phi_n(t). \tag{4}$$

Alternatively, with $F_L(s)$ and $\Phi_n(s)$ designating the Laplace transforms of $f(t)$ and $\phi_n(t)$, then

$$F_L(s) = \sum_{n=-\infty}^{+\infty} f_n \Phi_n(s). \tag{5}$$

The relationship between $f(t)$ and f_n in (4) can be viewed as an expansion of $f(t)$ in terms of a set of functions $\{\phi_n(t)\}$; for convenience we will require that given the set of functions $\{\phi_n(t)\}$, the sequence f_n representing $f(t)$ is unique. This then requires that the set of functions $\{\phi(t)\}$ be linearly independent, i.e., that any one of the functions is not expressible as a linear combination of the others. In this discussion, we have not assumed that the set of functions $\{\phi_n(t)\}$ is complete, but we have assumed that $f(t)$ is in the space spanned by $\{\phi_n(t)\}$.

Equations (1) and (2) can now be used to derive necessary and sufficient conditions on the set of functions $\{\Phi_n(s)\}$ in (5). Toward this end, let $G_L(s)$, $F_L(s)$, and $H_L(s)$ denote the Laplace transforms of $g(t)$, $f(t)$, and $h(t)$, respectively. Then (1) can be rewritten as

$$G_L(s) = F_L(s)H_L(s). \tag{6}$$

From (5) and its counterparts for $G_L(s)$ and $H_L(s)$, it follows that

$$\sum_{n=-\infty}^{+\infty} g_n \Phi_n(s) = \sum_{r=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} f_k h_r \Phi_r(s) \Phi_k(s). \tag{7}$$

If we carry out the substitution of variables $n=r+k$ on the right-hand side of (7), the equation can be rewritten as

$$\sum_{n=-\infty}^{+\infty} g_n \Phi_n(s) = \sum_{n=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} f_k h_{n-k} \Phi_{n-k}(s) \Phi_k(s). \tag{8}$$

But using (2) to express the left-hand side of (8) in terms of f_n and h_n, we obtain

$$\sum_{n=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} f_k h_{n-k} \Phi_n(s)$$
$$= \sum_{n=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} f_n h_{n-k} \Phi_{n-k}(s) \Phi_k(s). \tag{9}$$

In order that (9) may hold for any sequences f_k and h_{n-k}, we require that

$$\Phi_n(s) = \Phi_{n-k}(s)\Phi_k(s) \tag{10}$$

or equivalently

$$\Phi_n(s)\Phi_m(s) = \Phi_{n+m}(s) \tag{11}$$

from which a general form for the functions $\{\Phi_n(s)\}$ can be derived. In particular,

$$\Phi_{n+1}(s) - \Phi_1(s)\Phi_n(s) = 0$$

which is a difference equation in $\Phi_n(s)$ with a general solution of the form

$$\Phi_n(s) = c[\Phi_1(s)]^n. \tag{12}$$

Furthermore, from (11), $\Phi_0^2(s)=\Phi_0(s)$, which is satisfied only for $c=0$ and $c=1$. Thus the nontrivial solution to (12) is

$$\Phi_n(s) = [\Phi_1(s)]^n. \tag{13}$$

Consequently, (13) must be satisfied if the expansion in (5) is to result in a discrete representation for which continuous convolution is mapped to discrete convolution. The condition on $\Phi_1(s)$ under which the set of functions $\{\phi_n(t)\}$, corresponding to $\{\Phi_n(s)\}$ as given in (13), is complete has been discussed by Masry, Steiglitz, and Liu [2].

Since a continuous-time function can be represented by its Laplace transform and a sequence can be represented by its z-

transform, the discrete representation as expressed in (4) and (5) can be thought of as a mapping from the Laplace transform of the continuous-time function to the z-transform of the sequence. However, a mapping from the Laplace transform to the z-transform cannot always be expressed as a mapping from the s plane to the z plane (z cannot be written as a function of s). If the mapping also preserves convolution, i.e., if (13) is satisfied, then the discrete representation of (4) or (5) will correspond to expressing z as a function of s. To show this result, let the z-transform of the sequence f_n be denoted by $F_D(z)$ so that

$$F_D(z) = \sum_{n=-\infty}^{+\infty} f_n z^{-n}. \tag{14}$$

Since $\Phi_n(s) = [\Phi_1(s)]^n$, then (5) can be rewritten as

$$F_L(s) = \sum_{n=-\infty}^{+\infty} f_n \{[\Phi_1(s)]^{-1}\}^{-n}. \tag{15}$$

Comparing (14) and (15), it is clear that the variables z and s are related by

$$z = [\Phi_1(s)]^{-1} \triangleq M(s). \tag{16}$$

A common example of a discrete representation that preserves convolution for band-limited continuous-time signals is periodic sampling. In this case, the continuous-time signal is represented by a sequence f_n consisting of samples of the signal equally spaced in time so that

$$f_n = Tf(nT) \tag{17}$$

where T designates the sampling period. It is well known that periodic sampling can be viewed as an expansion of the continuous-time function $f(t)$ in the form of (4) with the functions $\{\phi_n(t)\}$ given by

$$\phi_n(t) = \sin\frac{\pi}{T}(t - nT)/\pi(t - nT). \tag{18}$$

The Laplace transform of the functions in (18) converge only on the $j\omega$ axis. On the $j\omega$ axis their transforms are

$$\Phi_n(j\omega) = \begin{cases} e^{-j\omega nT} = [e^{j\omega T}]^{-n}, & |\omega T| \leq \pi \\ 0, & |\omega T| > \pi. \end{cases} \tag{19}$$

Thus we see that the transform of these functions has the desired form as given in (13). The advantage to a discrete representation based on periodic sampling is that the coefficients in the expansion are easily obtained. Furthermore, as opposed to many other representations, the function $f(t)$ is not required for all t in order to obtain each of the sequence values of f_n. The major disadvantage to this representation is that it requires that the function $f(t)$ be band-limited. When we are considering simulating a continuous-time system digitally, the frequency response of the continuous-time linear system is usually not truly band-limited. In addition, periodic sampling corresponds to a linear relationship between analog frequencies and digital frequencies. As we will see in a later discussion, it is sometimes advantageous to have a nonlinear relationship between analog and digital frequencies.

Another example of a discrete representation that preserves convolution is the Poisson transform. For a continuous-time function $f(t)$ that is zero for $t<0$, its Poisson transform is defined as

$$f_n = \int_0^\infty \frac{t^n}{n!} e^{-t} f(t)dt, \qquad n = 0, 1, 2, \cdots. \tag{20}$$

The sequence f_n then provides a discrete representation of the continuous-time function $f(t)$. Piovoso and Bolgiano [3] have shown that the Poisson transform has the desired property of mapping continuous convolution to discrete convolution, i.e., the Poisson transform of the convolution of two time functions is the discrete convolution of the Poisson transforms of the components of the convolution.

In the case of periodic sampling, the function $M(s)$ of (16) which specifies the mapping from the s plane to the z plane is $M(j\omega) = e^{j\omega T}$. When ω is real, corresponding to the imaginary axis in the s plane, the magnitude of $M(j\omega)$ is unity corresponding to the unit circle in the z plane. Thus the representation resulting from periodic sampling maps the spectrum of the analog signal as viewed on the imaginary axis of the s plane to the spectrum of the digital signal as observed on the unit circle. In contrast, the function $M(s)$ resulting from the use of the Poisson transform is $M(s) = 1/(1-s)$. It is straightforward to verify that the $j\omega$ axis in the s plane maps to a circle in the z plane with radius one half and center at $z = 1/2$ so that it passes through the points $z = 0$ and $z = 1$. Consequently, it is tangent to the unit circle at $z = 1$ but otherwise does not coincide with the unit circle. Thus when an analog function is mapped to the z plane by use of the Poisson transform, its Fourier transform does not map onto the unit circle, but rather onto the circle to which the $j\omega$ axis in the s plane is mapped. This does not, of course, affect the validity of the Poisson transform for simulation since both the system input and the system impulse response are mapped in the same way.

As discussed by Steiglitz [1], another example of a discrete representation with the desired property is based on the use of the bilinear transformation in relating the s and z planes. The bilinear transformation is commonly used for designing frequency selective digital filters [4]. This transformation has the property that no aliasing results when it is used to map a continuous-time filter to a discrete-time filter. On the other hand, the use of the bilinear transformation introduces a distortion in the frequency axis; the imaginary axis in the s plane is mapped nonlinearly onto the unit circle in the z plane. A common design procedure in discrete-time filtering is to derive the filter from a continuous-time design by means of the bilinear transformation while the discrete-time signal is obtained from the continuous-time signal by means of periodic sampling. Because of the distortion in the frequency axis resulting from the bilinear transformation, this procedure is generally restricted to situations in which the distortion in the frequency axis is not important. This is true, for example, in the design of digital filters with piecewise constant frequency characteristics.

If we wish to use the bilinear transformation for simulation when the filter characteristics are not piecewise constant, the same nonlinear distortion must be applied to the frequency characteristics of the input signal and the output signal; *both* the signals and the systems must be mapped to discrete signals and systems using the bilinear transformation. Since the bilinear transformation corresponds to a mapping from the s plane to the z plane with

$$z = M(s) = \left[\frac{a + s}{a - s}\right] \tag{21}$$

it can also be viewed as an expansion of a time function $f(t)$ in terms of a series with the form of (4). $\Phi_n(s)$, the Laplace transform of $\phi_n(t)$, is from (13) and (16)

$$\Phi_n(s) = \left[\frac{a+s}{a-s}\right]^{-n}. \tag{22}$$

The functions $\{\phi_n(t)\}$ corresponding to (22) are

$$\phi_n(t) = \begin{cases} 2a(-1)^{n-1}e^{-at}L_{n-1}^{(1)}(2at)u_{-1}(t) \\ \qquad + (-1)^n u_0(t), & n > 0 \\ u_0(t), & n = 0 \\ 2a(-1)^{n-1}e^{at}L_{-n-1}^{(1)}(-2at)u_{-1}(-t) \\ \qquad + (-1)^n u_0(t), & n < 0 \end{cases} \tag{23}$$

where

$$L_{n-1}^{(1)}(x) = -\frac{d}{dx}L_n(x)$$

$$L_n(x) = \frac{e^x}{n!}\frac{d^n}{dx^n}(x^n e^{-x}). \tag{24}$$

The coefficients f_n in the expansion of $f(t)$ can be obtained by utilizing the fact that, for the set of functions given in (23),

$$\int_{-\infty}^{+\infty} t\phi_n(t)\phi_m(t)dt = \begin{cases} n, & n = m \\ 0, & n \neq m \end{cases} \tag{25}$$

so that

$$f_n = \frac{1}{n}\int_{-\infty}^{+\infty} tf(t)\phi_n(t)dt, \qquad n \neq 0. \tag{26}$$

We can obtain f_0 from the f_n by noting that if $f(t)$ is bounded, so that it contains no impulses at $t=0$, then

$$\sum_{n=-\infty}^{+\infty} (-1)^n f_n = 0. \tag{27}$$

If we restrict $f(t)$ to be zero for $t<0$, then from (26) and (27) it will follow that the coefficients f_n can be obtained by exciting a linear time-invariant network with $f(-t)$. To derive this result, we note that from (23) and (26), if $f(t)$ is zero for $t<0$, f_n will be zero for $n<0$. From (26), it now follows that for $n>0$,

$$f_n = \frac{1}{n}\int_0^{+\infty} f(t)[t\phi_n(t)]dt \tag{28a}$$

$$= \frac{1}{n}\int F_L(-s)[-\Phi_n'(s)]\frac{ds}{2\pi} \tag{28b}$$

where the prime denotes differentiation with respect to s. Since $\Phi_n(s) = [(a-s)/(a+s)]^n$,

$$-\Phi_n'(s) = n\frac{2a}{(s+a)^2}\left[\frac{a-s}{a+s}\right]^{n-1}$$

so that

$$f_n = \int F_L(-s)\frac{2a}{(s+a)^2}\left[\frac{a-s}{a+s}\right]^{n-1}\frac{ds}{2\pi}. \tag{29}$$

The integral of (28a) therefore corresponds to the response at $t=0$ of a linear time-invariant filter with impulse response $h_n(t) = (1/n)t\phi_n(t)$ or system function

$$H_n(s) = \frac{2a}{(s+a)^2}\left[\frac{a-s}{a+s}\right]^{n-1} \tag{30}$$

to an input $f(-t)$. To obtain f_0 we note from (27) that

$$f_0 = -\sum_{n=1}^{\infty}(-1)^n f_n$$

$$= -\sum_{n=1}^{\infty}(-1)^n \int F_L(-s)\frac{2a}{(s+a)^2}\left[\frac{a-s}{a+s}\right]^{n-1}\frac{ds}{2\pi}.$$

Interchanging the order of summation and integration, we find that

$$f_0 = \int \frac{ds}{2\pi}F_L(-s)\frac{2a}{(s+a)^2}\sum_{n=1}^{\infty}(-1)^{n-1}\left[\frac{a-s}{a+s}\right]^{n-1}$$

$$= \int \frac{ds}{2\pi}F_L(-s)\frac{2a}{(s+a)^2}\sum_{n=0}^{\infty}\left[\frac{s-a}{s+a}\right]^{n}$$

$$= \int \frac{ds}{2\pi}F_L(-s)\frac{2a}{(s+a)^2}\frac{(s+a)}{2a}$$

$$f_0 = \int \frac{ds}{2\pi}F_L(-s)\frac{1}{s+a}.$$

Consequently, f_0 is obtained as the output at $t=0$ of a linear filter with input $f(-t)$ and system function $1/(s+a)$. In summary, the sequence f_n can be obtained by means of the network in Fig. 1 where $\tilde{f}_n(t)$ denote the outputs at the indicated taps and $f_n = \tilde{f}_n(0)$. The network of Fig. 1, which is similar, but not identical, to a Laguerre network [5], is used to obtain the coefficients f_n from the continuous-time function $f(t)$. To "desample," i.e., to obtain $f(t)$ from the f_n, a network similar to that in Fig. 1 can be used. In particular, we recall that

$$f(t) = \sum_{n=0}^{\infty} f_n\phi_n(t)$$

$$F_L(s) = \sum_{n=0}^{\infty} f_n\Phi_n(s)$$

where $\Phi_n(s) = [(a-s)/(a+s)]^n$. Consequently, the function $\phi_n(t)$ is obtained as the impulse response of a network with system function $[(a-s)/(a+s)]^n$, and thus $f(t)$ can be recovered from the sequence f_n by the means shown in Fig. 2.

In general, the discrete sequence representing a continuous-time signal will be of infinite length. For practical reasons it is usually necessary to truncate the representation in some way so that only a finite number of nonzero values is used. To consider the effect, in the case of the bilinear transformation, of modifying the representation to contain only a finite number of nonzero terms, let us denote the approximation to

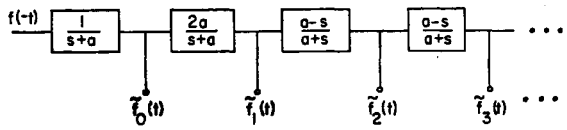

Fig. 1. All-pass network for conversion of a continuous-time signal $f(t)$ to its bilinear representation f_n. The representation f_n is related to the output of each stage by $f_n = \tilde{f}_n(0)$.

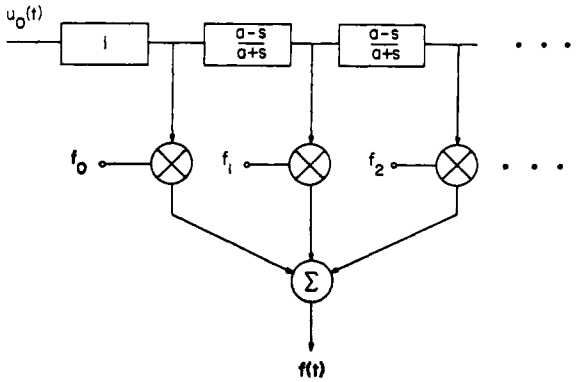

Fig. 2. All-pass network for reconstruction of a continuous-time signal $f(t)$ from its bilinear representation f_n.

$f(t)$ using only a finite number of nonzero terms by

$$f^A(t) = \sum_{n=0}^{N-1} f_n{}^A \phi_n(t)$$

with the exact representation of $f(t)$ given as

$$f(t) = \sum_{n=0}^{\infty} f_n \phi_n(t).$$

If the coefficients $f_n{}^A$ are chosen to equal the coefficients f_n for $n=0, 1 \cdots, N-1$, and since

$$\int_{-\infty}^{+\infty} t\phi_n(t)\phi_m(t)dt = \begin{cases} n, & n = m \\ 0, & n \neq m \end{cases} \tag{31}$$

then $f^A(t)$ will correspond to an approximation to $f(t)$ that minimizes the weighted integral square error given by

$$E = \text{error} = \int_0^{+\infty} t[f(t) - f^A(t)]^2 dt. \tag{32}$$

Hence, generally speaking, truncation weights errors for large t more than for small t so that we would expect $f^A(t)$ to approximate $f(t)$ more closely as t increases.

More generally, rather than obtaining the finite duration sequence $f_n{}^A$ from f_n by simple truncation, it can be obtained by "windowing":

$$f_n{}^A = w_n f_n \tag{33}$$

where $w_n = 0$ for $n < 0$ and $n \geq N$. For the case of simple truncation $w_n = 1$ for $0 \leq n < N$. To describe the effect of applying the window w_n to the sequence, let $F^A(\Omega)$, $W(\Omega)$, and $F(\Omega)$ represent the z-transforms on the unit circle of the sequences $f_n{}^A$, w_n, and f_n, respectively. Then

$$F^A(\Omega) = \frac{1}{2\pi}\int_{-\pi}^{\pi} W(\Omega - \alpha)F(\alpha)d\alpha \tag{34}$$

i.e., multiplication of the squence f_n by the window w_n corresponds to "smearing" the transform of the sequence f_n with the transform of w_n. Alternatively, we can view this as a "smearing" of the spectrum of the original time function in such a way that spectral resolution decreases as the frequency increases. To see that this is so, let $f^A(t)$ denote the continuous-time function represented by the sequence $f_n w_n$ and let $G^A(\omega)$ and $G(\omega)$ denote the Fourier transforms of $f^A(t)$ and $f(t)$, respectively. From (21) with $s = j\omega$ and $z = e^{j\Omega}$, the relationship between Ω and ω is given by

$$\Omega = 2 \arctan \frac{\omega}{a}. \tag{35}$$

Therefore,

$$G^A(\omega) = F^A\left(2 \arctan \frac{\omega}{a}\right)$$

$$G(\omega) = F\left(2 \arctan \frac{\omega}{a}\right)$$

so that (34) can be rewritten as

$$G^A(\omega) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} W\left[2 \arctan \frac{\omega}{a} - 2 \arctan \frac{\beta}{a}\right] \cdot G(\beta) \frac{2a}{a^2 + \beta^2} d\beta. \tag{36}$$

As an approximation, let us assume that the effective width of the spectral window $W(\Omega)$ is sufficiently small so that over the width of the window, the bilinear frequency transformation of (35) can be approximated by a linear characteristic. Then (36) can be approximated by

$$G^A(\omega) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} W\left[(\omega - \beta)\frac{2a}{a^2 + \omega^2}\right] \cdot G(\beta) \frac{2a}{a^2 + \omega^2} d\beta. \tag{37}$$

The argument $(\omega - \beta)[2a/(a^2+\omega^2)]$ can be interpreted as a linear scaling of the spectral window, while maintaining the shape so that the window becomes wider as ω increases. Thus the distortion due to truncating the representation by the application of a window can be viewed in terms of "smearing" the analog spectrum in such a way that spectral resolution decreases as the analog frequency increases.

One of the advantages of a discrete representation based on the bilinear transformation, as developed above in contrast to periodic sampling, is that the analog signal need not be band-limited. On the other hand, periodic sampling is easier to implement and the periodic samples can be obtained without buffering, in the sense that periodic sampling is a memoryless transformation. Bilinear sampling requires the entire waveform to be available before any of the values in the discrete sequence can be obtained, corresponding to infinite buffering of the input.

For the simulation of an analog filter it is possible to avoid the need for infinite buffering of the input by sectioning. To illustrate the procedure, let us consider an analog filter with impulse response $h(t)$, system function $H(s)$, and input $x(t)$. The discrete filter used for the simulation will have a unit sample response h_n where, assuming the system is causal,

$$h(t) = \sum_{n=0}^{\infty} h_n \phi_n(t).$$

Now the entire input $x(t)$ can be expanded in terms of the $\{\phi_n(t)\}$, yielding a sequence of coefficients x_n, this sequence filtered, and the resulting output y_n used to construct the continuous-time output $y(t)$ as

$$y(t) = \sum_{n=-\infty}^{+\infty} y_n \phi_n(t).$$

However, as has already been discussed, this requires infinite buffering since no output values can be obtained until all of the input is available. As an alternative, let us divide the input $x(t)$ into sections of equal duration T so that, assuming $x(t)=0$, $t<0$,

$$x(t) = \sum_{k=0}^{+\infty} x_k(t - kT) \tag{38}$$

where $x_k(t)=0$ for $t<0$ and for $t>T$. Now each of the sections $x_k(t)$ can be expanded as

$$x_k(t) = \sum_{n=0}^{\infty} x_{n,k} \phi_n(t) \tag{39}$$

where we note that the sequence $x_{n,k}$ is zero for $n<0$ since each of the sections is zero for $t<0$. Since the system is linear, the response of the continuous-time filter can be expressed as the sum of the responses to each of the sections, i.e., if $y_k(t-kT)$ is the response to $x_k(t-kT)$, then

$$y(t) = \sum_{k=0}^{+\infty} y_k(t - kT).$$

To implement the simulation we obtain the discrete representation $\{x_{n,k}\}$ for each section, process this sequence with a discrete linear system having a unit sample response h_n, and use the resulting sequence to construct $y_k(t-kT)$. In general, although each section of the input is of finite duration T, the sequence representing it will be of infinite duration as will the output for each section. In a practical implementation, it would be necessary to truncate both the representation $\{x_{n,k}\}$ for each section and the output for each section $y_k(t-kT)$. The effect of this truncation must be taken into account in evaluating this procedure.

III. Discrete Representation of Sequences

The foregoing examples provide several means for obtaining a discrete representation of continuous-time functions consistent with the requirement that the representation preserve convolution. It also is possible to convert from one discrete representation to another by means of a suitable transformation of the sequences. In particular, let us consider $f(t)$ represented by two different sequences f_n and g_k so that

$$f(t) = \sum_{n=-\infty}^{+\infty} f_n \phi_n(t) \tag{40a}$$

$$f(t) = \sum_{k=-\infty}^{+\infty} g_k \lambda_k(t). \tag{40b}$$

If we assume that the set of functions $\{\phi_n(t)\}$ is complete so that each $\lambda_k(t)$ can be expanded in terms of the $\{\phi_n(t)\}$, then

$$\lambda_k(t) = \sum_{n=-\infty}^{+\infty} \psi_{k,n} \phi_n(t). \tag{41}$$

The sequences f_n and g_k are then related by

$$f_n = \sum_{k=-\infty}^{+\infty} g_k \psi_{k,n}. \tag{42}$$

Equation (42) can be viewed as an expansion of the sequence f_n in terms of a set of sequences $\{\psi_{k,n}\}$, with the sequence g_k representing the coefficients in the expansion. Thus the sequences $\{\psi_{k,n}\}$ play the same role in (42) that the functions $\{\phi_n(t)\}$ play in (40a).

Since the mapping from $f(t)$ to f_n and also from $f(t)$ to g_k preserves convolution, it follows that an expansion of (42) must also preserve convolution, i.e.. if $f_n = f_{1,n} * f_{2,n}$ where $*$ denotes discrete convolution, and if $g_{1,k}$ and $g_{2,k}$ denote the coefficients in the expansion of $f_{1,n}$ and $f_{2,n}$, respectively, then according to (42) the sequence g_k will equal the convolution of $g_{1,k}$ and $g_{2,k}$. The properties required of the set of sequences $\{\psi_{k,n}\}$ in order that convolution be preserved can be determined in a manner analogous to (5) through (14) with the result that if $\Psi_k(z)$ denotes the z-transform of $\psi_{k,n}$, then the $\{\Psi_k(z)\}$ must satisfy the relation

$$\Psi_r(z)\Psi_{k-r}(z) = \Psi_k(z) \tag{43}$$

so that $\Psi_k(z)$ must be given by

$$\Psi_k(z) = [\Psi_1(z)]^k. \tag{44}$$

Thus (44) plays the same role in mapping sequences to preserve discrete convolution that (13) plays in mapping continuous convolution to discrete convolution. When the $\{\Psi_k(z)\}$ satisfy (44), the z-transforms of the sequences f_n and g_k can be related by a substitution of variables. Specifically, let $F(z)$ denote the z-transform of f_n and $G(\hat{z})$ denote the $\hat{z}$-transform of g_k. Then with $\hat{z} = [\Psi_1(z)]^{-1} \triangleq m(z)$, it follows from (42), in a manner similar to the derivation of (16), that

$$F(z) = G[m(z)]. \tag{45}$$

Thus changing from the representation f_n to the representation g_k is equivalent to mapping the complex frequency variable z for f_n to the complex frequency variable $\hat{z}$ for g_k by means of a substitution of variables $\hat{z} = m(z)$.

Thus far we have been concerned with the representation of a continuous-time function by a sequence so that linear time-invariant systems could be simulated by linear shift-invariant systems. We found that we can achieve this result if we map the s plane to the z plane in an invertible fashion. Similarly, we can convert from one representation to another, each of which preserves convolution, by mapping one z plane to another. There are some applications, however, in which we are concerned not so much with simulation as with the mapping of one complex plane to another. We would now like to turn our attention to the discussion of such problems.

A. *Digital Warping of Spectra* [6]

In a variety of applications, it is useful to transform a sequence to a new sequence whose Fourier transform is equal to the Fourier transform of the original sequence on a distorted (warped) frequency scale. In this section, the basis for carrying out such a transformation using the ideas presented in the foregoing will be discussed, followed by several applications of frequency warping.

We found in the previous section that the transformation between the sequences f_n and g_k as expressed by (42) is equivalent to the mapping of one complex frequency plane to another if (44) is satisfied. We did not, however, focus upon any particular choice for the mapping $\hat{z} = m(z)$. Let us require that the function $\hat{z} = m(z)$ map the unit circle in the z plane to the unit circle in the $\hat{z}$ plane. Letting Ω be angular frequency in the

z plane and $\hat{\Omega}$ be angular frequency in the $\hat{z}$ plane, we want $\hat{z}=m(z)$ to satisfy

$$e^{j\hat{\Omega}} = m[e^{j\Omega}] \tag{46}$$

or

$$\hat{\Omega} = \theta(\Omega) \tag{47}$$

where $m[e^{j\Omega}]=e^{j\theta(\Omega)}$.

If the function $\hat{z}=m(z)$ satisfies (46), the Fourier transform of g_k will be the Fourier transform of f_n on a warped frequency axis.

As the z-transform of the sequence $\psi_{k,n}$ must satisfy (44), (46), and (47), the z-transform of $\psi_{k,n}$ evaluated on the unit circle is found by

$$\Psi_k(e^{j\Omega}) = e^{-jk\theta(\Omega)}. \tag{48}$$

Therefore, if the spectrum of g_k is to be a warped version of the spectrum of f_n, the Fourier transform of the sequence $\psi_{k,n}$ must have an all-pass characteristic, i.e., the magnitude of the spectrum of $\psi_{k,n}$ must be unity independent of frequency. The *negative* of the phase of $\Psi(e^{j\Omega})$ will then correspond to the mapping between the frequency axes. In particular, if we want the function $\hat{z}=m(z)$ to be rational and also map the circle $z=e^{j\Omega}$ to the circle $\hat{z}=e^{j\hat{\Omega}}$, then $m(z)$ must be of the form

$$\hat{z} = m(z) = \prod_k \frac{1 - a_k z^{-1}}{z^{-1} - a_k^*}$$

where a_k^* denotes the complex conjugate of a_k.

In order that $m(z)$ may be invertible and in the form already given, the coefficients a_k must be such that the interval $-\pi<\Omega\leq\pi$ maps one-to-one to the interval $-\pi<\hat{\Omega}\leq\pi$. A necessary (but not sufficient) condition for this to be true is that the number of zeros minus the number of poles of $m(z)$ which lie inside the unit circle be plus or minus unity. A useful choice for $m(z)$, to which we will restrict the remaining discussion, is a first-order all-pass of the form

$$\hat{z} = m(z) = \frac{1 - az^{-1}}{z^{-1} - a^*} \tag{49}$$

where $0<|a|<1$. Then the z-transforms of the sequences $\{\psi_{k,n}\}$ used in the expansion of f_n must be

$$\Psi_k(z) = \left(\frac{z^{-1} - a^*}{1 - az^{-1}}\right)^k. \tag{50}$$

We see that the z-transform of the sequence $\psi_{k,n}$ corresponds to a kth-order all-pass having unity gain with k poles at $z=a$ and k zeros at $z=1/a^*$. For real values of the parameter a, the mapping between the frequency variables $\hat{\Omega}$ and Ω is given by

$$\hat{\Omega} = \theta(\Omega) = \arctan\left[\frac{(1 - a^2)\sin\Omega}{(1 + a^2)\cos\Omega - 2a}\right]. \tag{51a}$$

An alternate form of (51a) is

$$\hat{\Omega} = \theta(\Omega) = \Omega + 2\arctan\left[\frac{a\sin\Omega}{1 - a\cos\Omega}\right]. \tag{51b}$$

To determine the inverse of the relationships in (51), we note that the inverse of (49) is

$$z = \frac{1 + a\hat{z}^{-1}}{\hat{z}^{-1} + a^*}$$

which corresponds to replacing a by $-a$ and interchanging z and $\hat{z}$ in (49). Consequently, the inverse of (51a) is obtained by replacing a by $-a$ in that equation:

$$\Omega = \theta^{-1}(\hat{\Omega}) = \arctan\left[\frac{(1 - a^2)\sin\hat{\Omega}}{(1 + a^2)\cos\hat{\Omega} + 2a}\right].$$

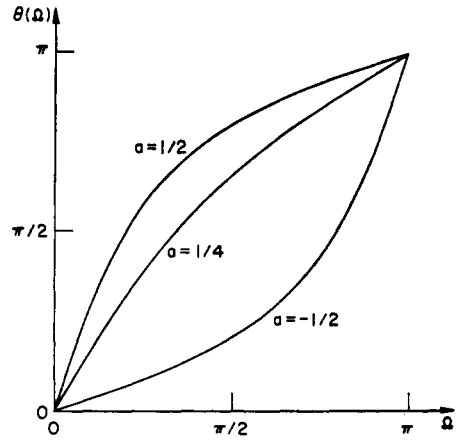

Fig. 3. Frequency-warping function of (51) for several values of parameter a.

A plot of the frequency-warping function of (51a) for several values of the parameter a is given in Fig. 3. We see that the frequency mapping corresponding to this choice of sequences $\{\psi_{k,n}\}$ is inherently nonlinear. Therefore, the Fourier transform of the sequence g_k, the sequence derived from f_n, will be the Fourier transform of f_n evaluated on a nonlinear scale:

$$F[e^{j\Omega}] = G[e^{j\theta(\Omega)}].$$

In particular, computing $G[e^{j\hat{\Omega}}]$ at a set of uniformly spaced samples in $\hat{\Omega}$ (using the fast Fourier transform algorithm, for instance) corresponds to the evaluation of $F(e^{j\Omega})$ at a set of nonuniformly spaced samples. If the parameter a is picked to be real and between 0 and 1, the effect on the spectrum of f_n will be to sample with higher resolution at low frequencies and with lower resolution at higher frequencies. If instead, a is negative between 0 and -1, the effect is reversed, i.e., the spectrum of f_n is evaluated with greater resolution at high frequencies than at low frequencies. By letting the parameter a assume complex values, the point of maximal resolution can be placed at any desired frequency.

The sequence g_k can be determined from the sequence f_n by using the fact that the set of sequence $\{\psi_{k,n}\}$ is orthogonal with a weighting sequence of n:

$$\sum_{n=-\infty}^{\infty} n\psi_{r,n}\psi_{k,n} = \begin{cases} k, & k = r \\ 0, & k \neq r. \end{cases} \tag{52}$$

This relationship can be derived by applying Parseval's theorem to the left side of (52). Consequently,

$$g_k = \frac{1}{k}\sum_{n=-\infty}^{\infty} n\psi_{k,n}f_n, \qquad k \neq 0. \tag{53}$$

In the remainder of the discussion, we will assume, for convenience, that the original sequence f_n is zero for $n<0$. The results are easily extended to the more general case. With this restriction, (53) becomes

$$g_k = \begin{cases} \frac{1}{k}\sum_{n=0}^{\infty} n\psi_{k,n}f_n, & k>0 \\ 0, & k<0. \end{cases} \tag{54}$$

This relationship is similar to (28a) which was used to determine the coefficients in the expansion of a continuous-time function based on the bilinear transformation. It was shown from (28a) that the coefficients can be obtained by processing the continuous-time function with the cascade chain of networks of Fig. 1. In a similar manner, the coefficients g_k can be obtained by processing the sequence f_n with a cascade chain of digital networks. Specifically, the sum in (54) corresponds to the response at $n=0$ of a linear discrete system to an input f_{-n}, with the unit sample response of the system being $h_{k,n}=(1/k)n\psi_{k,n}$ or system function

$$H_k(z) = \frac{-z}{k}\frac{d\Psi_k(z)}{dz}$$

$$H_k(z) = \frac{(1-a^2)z^{-1}}{(1-az^{-1})^2}\left[\frac{z^{-1}-a}{1-az^{-1}}\right]^{k-1}, \qquad k=1,2,\cdots. \tag{55}$$

Equation (54) cannot be used to determine g_0. However, from (44),

$$g_0 = f_0 - \sum_{k=1}^{\infty} g_k\psi_{k,0}$$

or since $\psi_{k,0}=(-a)^k$,

$$g_0 = f_0 - \sum_{k=1}^{\infty} g_k(-a)^k.$$

Thus, using (54),

$$g_0 = f_0 - \sum_{n=0}^{\infty} f_n n \sum_{k=1}^{\infty}\frac{1}{k}(-a)^k\psi_{k,n}.$$

Let

$$\sum_{k=1}^{\infty}\frac{1}{k}(-a)^k\psi_{k,n} = v_n \tag{56}$$

so that

$$g_0 = f_0 - \sum_{n=0}^{\infty} f_n n v_n$$

$$= \sum_{n=0}^{\infty} f_n(\delta_n - nv_n) \tag{57}$$

where δ_n is a unit sample, i.e., $\delta_n=1$ for $n=0$, and $\delta_n=0$ otherwise. Thus, from (57), g_0 can be obtained as the response at $n=0$ to an input f_{-n} of a linear discrete system having a unit sample response $h_{0,n}=(\delta_n-nv_n)$. From (55) and (56), the z-transform of nv_n is given by

$$\sum_{n=0}^{\infty} nv_n z^{-n} = -\frac{az^{-1}}{1-az^{-1}}.$$

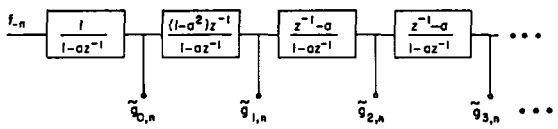

Fig. 4. All-pass network used to implement spectral warping of discrete-time signals. The sequence g_k, whose spectrum is the warped version of the input sequence f_n, is related to the output of each stage by $g_k=\tilde{g}_{k,0}$.

Therefore, the system function corresponding to the unit sample response $h_{0,n}$ is

$$H_0(z) = \frac{1}{1-az^{-1}}.$$

In summary, we can find the sequence g_k by passing the sequence f_{-n} through a linear shift-invariant system $H_k(z)$ given by

$$H_k(z) = \begin{cases} \frac{(1-a^2)z^{-1}}{(1-az^{-1})^2}\left(\frac{z^{-1}-a}{1-az^{-1}}\right)^{k-1}, & k>0 \\ \frac{1}{1-az^{-1}}, & k=0. \end{cases} \tag{58}$$

The output of the system $H_k(z)$ at $n=0$ is then the desired coefficient g_k. We note from (58) that for $k>2$, the kth system corresponds to cascading the $(k-1)$th with an all-pass network and that $H_1(z)$ corresponds to $H_0(z)$ in cascade with a first-order network. Thus the set of systems $\{H_k(z)\}$ can be implemented as a cascade chain as illustrated in Fig. 4. With the output of each section denoted by $\tilde{g}_{k,n}$, the desired coefficients g_k are given by $g_k=\tilde{g}_{k,0}$. We note that replacing a by $-a$ in (58) is equivalent to multiplying $H_k(z)$ by $(-1)^k$ and replacing z by $-z$, which in turn corresponds to multiplying $h_{k,n}$ by $(-1)^n(-1)^k$. Since

$$g_k = \sum_{n=0}^{\infty} f_n h_{k,n}$$

multiplying $h_{k,n}$ by $(-1)^{n+k}$ is equivalent to multiplying f_n by $(-1)^n$ and g^k by $(-1)^k$. Thus if the input to the network of Fig. 4 is multiplied by $(-1)^n$ and the resulting sequence g_k is multiplied by $(-1)^k$, then we have equivalently implemented the network of Fig. 4 with a replaced by $-a$. In a hardware realization of this network, for example, only positive values of a need to be implemented.

In the use of the bilinear transformation for the discrete representation of continuous-time signals, we noted that, in general, the representation would be of infinite length and would, in a practical implementation, need to be truncated. Truncation of the sequence corresponded to a "smearing" of the spectrum of the original time function with a spectral window that increases in width as frequency increases.

Truncation of the representation based on the sequences expressed in (44) causes a similar type of effect. In particular, we will assume that the truncation is carried out by multiplying g_k by a finite duration window so that the truncated sequence is represented by $g_k^A=w_kg_k$. With f_n^A denoting the sequence represented by g_k^A:

$$f_n^A = \sum_{k=0}^{\infty} g_k^A\psi_{n,k}$$

and with $F^A(\Omega)$ and $F(\Omega)$ denoting the Fourier transforms of

f_n^A and f_n, we have, paralleling the derivation of (36),

$$F^A(\Omega) = \frac{1}{2\pi}\int_{-\pi}^{\pi} F(\beta)W[\theta(\Omega) - \theta(\beta)]\frac{d\theta(\beta)}{d\beta}\,d\beta$$

or with $\theta(\Omega)$ given by (51a)

$$F^A(\Omega) = \frac{1}{2\pi}\int_{-\pi}^{\pi} F(\beta)W[\theta(\Omega) - \theta(\beta)]\frac{1 - a^2}{1 + a^2 - 2a\cos\beta}\,d\beta.$$

As with (37), we will assume that the window is sufficiently narrow so that over the width of the window the characteristic $\theta(\Omega)$ is approximately linear. Then

$$W[\theta(\Omega) - \theta(\beta)] = W\left[(\Omega - \beta)\frac{1 - a^2}{1 + a^2 - 2a\cos\Omega}\right]$$

so that

$$F^A(\Omega) = \frac{1}{2\pi}\int_{-\pi}^{\pi} F(\beta)W\left[(\Omega - \beta)\frac{1 - a^2}{1 + a^2 - 2a\cos\Omega}\right]\cdot\frac{1 - a^2}{1 + a^2 - 2a\cos\Omega}\,d\beta.$$

As with (37), the factor of $[(\Omega-\beta)(1-a^2)]/(1+a^2-2a\cos\Omega)$ can be interpreted as a linear scaling of the shape of the spectral window. For positive a, this corresponds to "smearing" the spectrum of the original sequence in such a way that spectral resolution decreases as frequency increases. For negative a, spectral resolution decreases as frequency decreases.

B. Applications of Digital Frequency Warping

1) Evaluation of Spectra at Nonuniformly Spaced Samples [6]: The ability to be able to distort the frequency variable in the Fourier transform of a sequence has proved to be useful in several contexts. The first to be discussed is directed toward warping the spectrum to achieve spectral analysis with unequal resolution.

In many applications, we are concerned with the problem of computing samples of the z-transform of a sequence on the unit circle. To obtain samples equally spaced around the unit circle, the most efficient procedure is to compute the discrete Fourier transform (DFT) using the fast Fourier transform (FFT) algorithm. Often we would like to obtain samples that are unequally spaced, corresponding, for example, to a constant-Q spectral analysis of the original sequence. An algorithm for accomplishing this nonlinearly spaced spectral analysis with an efficiency similar to that achievable with the FFT algorithm is not known. One procedure sometimes used is to evaluate explicitly the samples at the desired frequencies, using, for example, the Goertzel algorithm [7]. Another approximate procedure is to sum equally spaced frequency samples in bands. A related procedure corresponds to performing the spectral analysis of a sequence with a recursive or nonrecursive filter bank.

As an alternative to these methods, unequally spaced samples of the spectrum can be obtained by implementing an equal resolution spectral analysis on a distorted frequency scale. To carry out such an analysis , the sequence f_n whose spectrum is desired is processed by the network of Fig. 4 and the DFT of the output sequence g_k is computed using the FFT algorithm. To illustrate the effect of the warping, a

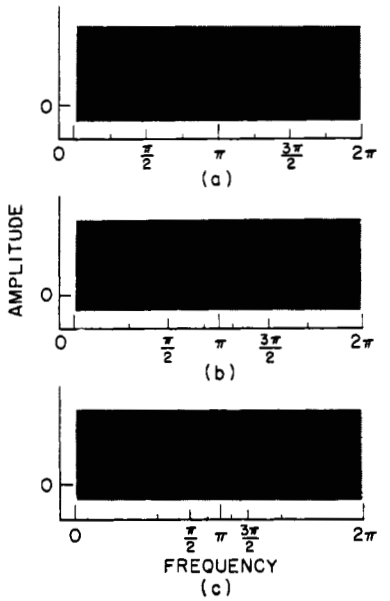

Fig. 5. Example of digital warping of spectra. (a) Original spectral magnitude. (b) Spectral magnitude of warped version of the original with $a = 1/4$. (c) Spectral magnitude of warped version of the original with $a = 1/2$.

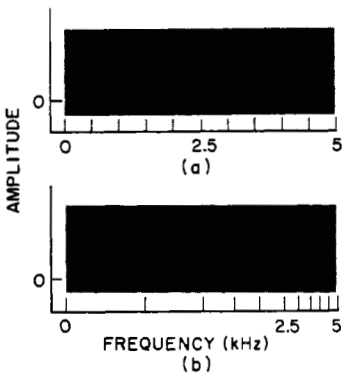

Fig. 6. Example of digital frequency warping on a sample of speech. (a) Original spectral magnitude. (b) Warped spectral magnitude with $a = 1/2$.

sequence whose spectrum is linear was processed in the manner just described. Specifically, the spectrum of the original sequence is indicated in Fig. 5(a). The DFT of the sequence obtained from the network of Fig. 4 is shown in Fig. 5(b) and (c) for two different values of parameter a. Another example is shown in Fig. 6. Fig. 6(a) represents the DFT of 51.2 ms of speech weighted with a Hanning window; Fig. 6(b) shows the DFT of the sequence obtained by processing the same speech sample with the network of Fig. 4.

The fact that the original sequence must first be processed by the network of Fig. 4 means that if the original sequence is of length N and the transformed sequence is of length M, then the computation time is proportional to NM excluding the time required for computation of the DFT. The use of the Goertzel algorithm to evaluate spectra at unequally spaced samples would likewise require computation proportional to NM. However, the network of Fig. 4 involves a single parameter. Consequently, a hardware implementation of frequency warping can utilize a fixed coefficient multiplier, which offers a considerable advantage over the use of algorithms requiring a coefficient memory and a more general multiplier.

2) Vernier Spectrum Analysis: In some applications it is desirable to carry out a vernier spectrum analysis corresponding to a high-resolution sampling of the spectrum over a small frequency range. One algorithm that has been proposed for such analysis is the chirp z-transform [8]. Another common procedure consists of modulating the signal to center the frequency band on which the vernier analysis is to be carried out around $\Omega = 0$, low-pass filtering and sampling the modulated signal, and transforming the result. As an alternative, vernier analysis can be carried out with the use of frequency warping. Referring to Fig. 3, we note that for a positive, the slope of the warping function $\theta(\Omega)$ near the origin is greater than unity. Thus the transform of a sequence processed with the all-pass network of Fig. 4 with a positive has highest resolution near $\Omega = 0$. As a increases toward unity, the effect is to provide a vernier analysis around $\Omega = 0$. In general, of course, we would like to obtain a vernier analysis around an arbitrary frequency Ω. This is accomplished by multiplying the original sequence by the complex exponential sequence $e^{j\Omega_0 n}$ prior to processing with the network of Fig. 4, to center the desired frequency range at $\Omega = 0$.

3) Helium Speech Translation: In the previous discussion, spectral warping was used to compute spectra on a nonlinear frequency scale. Another class of applications is directed toward the use of spectral warping to correct for a nonlinear distortion in frequency. An example of such an application is the correction of the speech of divers [9]. It is well known that when underwater divers breathe helium-rich gas mixtures at ambient pressures higher than atmospheric pressure, their speech is highly unintelligible. The effect of the change in breathing mixture and pressure results in a nonlinear shift in the resonant frequencies of the vocal tract. However, the fundamental frequency of the speech remains close to normal; consequently, when correcting for the frequency distortion, the fundamental frequency of the diver's speech should not be changed. This can be accomplished by extracting individual pitch periods of the speech, processing each period of the speech to compensate for the nonlinear frequency distortion, and then resynthesizing the speech with the original periodicity. Zue [10] has investigated the application of frequency warping within this context with encouraging results.

C. Hardware Implementation of Digital Frequency Warping

One of the advantages of the use of the network of Fig. 4 is that it can be implemented with digital hardware in a relatively straightforward and inexpensive manner. Thus, for example, it can be used peripherally to a digital computer in conjunction with an FFT program to implement a spectral analysis with unequal resolution.

To indicate one possible form of the hardware configuration for the network of Fig. 4, the network is redrawn in Fig. 7 indicating a realization with multipliers, adders, and delays. The set of equations by means of which the network of Fig. 7 is iterated is

$$\tilde{g}_{0,n} = a[\tilde{g}_{0,n-1} - 0] + f_{-n} \tag{59a}$$

$$\tilde{g}_{1,n} = a[\tilde{g}_{1,n-1} - 0] + \tilde{g}_{0,n-1} \tag{59b}$$

$$\tilde{g}_{k,n} = a[\tilde{g}_{k,n-1} - \tilde{g}_{k-1,n}] + \tilde{g}_{k-1,n-1}, \quad k = 2, 3, \cdots. \tag{59c}$$

With the difference equations written in the form of (59), the arithmetic operations involved are identical for each iteration

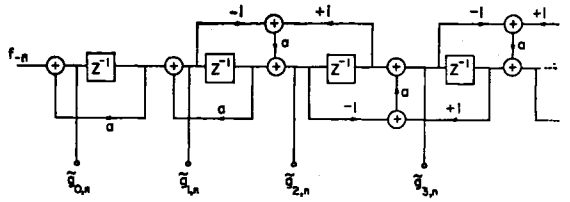

Fig. 7. Detailed realization of network in Fig. 4.

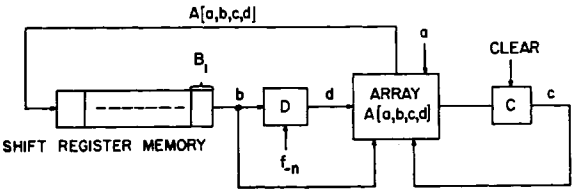

Fig. 8. Block diagram of a hardware realization of Fig. 7.

in both k and n. In particular, our implementation of the networks utilizes an arithmetic array with inputs a, b, c, and d. With $A[a, b, c, d]$ denoting the array output, the array implements the arithmetic function

$$A[a, b, c, d] = a(b - c) + d. \tag{60}$$

Thus the input a to the array represents the coefficient that controls the amount of spectral warping obtained. For the computation of $\tilde{g}_{0,n}$, the inputs b, c, and d are $\tilde{g}_{0,n-1}$, 0, and f_{-n}, respectively, so that

$$\tilde{g}_{0,n} = A[a, \tilde{g}_{0,n-1}, 0, f_{-n}].$$

Similarly,

$$\tilde{g}_{1,n} = A[a, \tilde{g}_{1,n-1}, 0, \tilde{g}_{0,n-1}]$$

$$\tilde{g}_{k,n} = A[a, \tilde{g}_{k,n-1}, \tilde{g}_{k-1,n}, \tilde{g}_{k-1,n-1}], \qquad k = 2, 3, \cdots.$$

The network of Fig. 7 must be iterated in both k and n. The iteration in the index k corresponds to computing the outputs $\tilde{g}_{k,n}$ along the network chain for a fixed n. The iteration in n then corresponds to updating each of these outputs for a new input sample. With the network iterated in k and then n, so that for each value of n all of the outputs $\tilde{g}_{k,n}$ are updated, the network states, corresponding to the delay outputs in the network of Fig. 7, can be stored in shift register memory.

A general structure that can be used for the implementation using a shift register memory is indicated in Fig. 8. Values in the shift register memory correspond to the outputs for each of the delay registers in the network of Fig. 7. For $k = 0$, register D contains f_{-n}, otherwise it is connected to the output of the shift register memory. For $k = 0$ and $k = 1$, the register C is cleared, otherwise it is connected to the output of the arithmetic array. The length of the shift register memory is equal to the length of the sequence g_k, which will be denoted by M. For each value of n, the network of Fig. 8 is iterated M times. In Table I the contents of the registers as a function of k are indicated.

The hardware configuration indicated in Fig. 8 has been implemented using TTL logic and dynamic shift register memory. With an 18-bit register length and $M = 512$, the hardware implementation contains 100 integrated-circuit packages. The configuration can operate at more than a 2-MHz clock rate. Consequently, to transform 512 data

TABLE I

CONTENTS OF THE SPECIFIED REGISTER IN THE HARDWARE REALIZATION OF THE DIGITAL WARPING NETWORK (FIG. 8) AS A FUNCTION OF k

k	B_1	D	C
0	$\tilde{g}_{0,n-1}$	f_{-n}	0
1	$\tilde{g}_{1,n-1}$	$\tilde{g}_{0,n-1}$	0
2	$\tilde{g}_{2,n-1}$	$\tilde{g}_{1,n-1}$	$\tilde{g}_{1,n}$
3	$\tilde{g}_{3,n-1}$	$\tilde{g}_{2,n-1}$	$\tilde{g}_{2,n}$
.	.	.	.
.	.	.	.
.	.	.	.

Note: k = the number of the stage in the all-pass network.

points to obtain a new 512-point sequence requires approximately 0.13 s. Computation time for a 512-point DFT using the FFT algorithm, implemented on a fixed-point 18-bit computer such as a DEC PDP-9, requires between 0.5 and 1 s; thus the inclusion of frequency warping to obtain unequal resolution using this piece of peripheral hardware increased computation time by from 10 to 20 percent over that required for an FFT alone. As the application warrants it, faster circuitry can, of course, be used.

IV. CONCLUSIONS

This paper has been directed toward a discussion and review of the discrete representation of signals within two contexts. The first related to the representation of continuous-time signals and systems by sequences for digital simulation. The second was the representation of a sequence by another to implement a frequency warping. Some of the ideas and results in the paper have found practical application while others are more speculative. It is hoped that as these ideas are further developed, additional applications will arise.

ACKNOWLEDGMENT

Much of the material in Section III-A was developed in collaboration with Prof. K. Steiglitz and other parts have benefited from discussions with him. The hardware implementation in Section III-C was carried out with the assistance of J. Tierney. Dr. R. Schafer, Dr. L. Rabiner, and Prof. H. Grebe offered many helpful comments on early drafts of the paper. The work has also benefited from the interest, suggestions, and encouragement of Dr. B. Gold.

REFERENCES

[1] K. Steiglitz, "The equivalence of digital and analog signal processing," *Inform. Contr.*, vol. 8, pp. 455–467, Oct. 1965.
[2] E. Masry, K. Steiglitz, and B. Liu, "Bases in Hilbert space related to the representation of stationary operators," *SIAM J. Appl. Math.*, vol. 16, pp. 552–562, May 1968.
[3] M. Piovoso and L. Bolgiano, "Digital simulation using Poisson transform sequences," in *Proc. Symp. Computer Processing in Communication*. Brooklyn, N. Y.: Polytechnic Press, 1969, pp. 505–514.
[4] B. Gold and C. Rader, *Digital Processing of Signals*. New York: McGraw-Hill, 1969.
[5] Y. W. Lee, "Synthesis of optimum linear systems by means of orthonormal functions," in *Statistical Theory of Communication*. New York: Wiley, 1960, ch. 19.
[6] A. Oppenheim, D. Johnson, and K. Steiglitz, "Computation of spectra with unequal resolution using the fast Fourier transform," *Proc. IEEE* (Lett.), vol. 59, pp. 299–301, Feb. 1971.
[7] C. R. Arnold, "Spectral estimation for transient waveforms," *IEEE Trans. Audio Electroacoust.*, vol. AU-18, pp. 248–257, Sept. 1970.
[8] L. R. Rabiner, R. W. Schafer, and C. M. Rader, "The chirp z-transform algorithm," *IEEE Trans. Audio Electroacoust.*, vol. AU-17, pp. 86–92, June 1969.
[9] G. Fant and J. Lindqvist, "Pressure and gas mixture effects on divers' speech," Speech Transmission Lab., Royal Inst. Tech. (KTH), Stockholm, Sweden, Quart. Progr. and Status Rep., Apr. 1968, pp. 7–17.
[10] V. Zue, "Translation of divers' speech using digital frequency warping," M.I.T. Res. Lab. Electron., Cambridge, Mass., Quart. Progr. Rep. 101, Apr. 15, 1971, pp. 175–182.