signals with low frequencies. The characteristics will be modified due to the changing performance of delta modulation. However, for this application, the performance of delta modulation is relatively insensitive as long as the quantized noise is approximately independent on a measured signal. Especially for signals with very low-frequency components, the method of omitting the integrator will be available.

ACKNOWLEDGMENT

The author wishes to thank Prof. J. Kawata and Prof. T. Kubota for their helpful comments and suggestions. The author also wishes to thank his many associates at the Department of Electronics, Tokyo Electrical Engineering College, Tokyo, Japan.

REFERENCES

[1] Y. Ishii, "A real-time signal processing system for correlation and spectrum analysis," *J. SICE* (Japan), vol. 8, pp. 734–748, Nov. 1969.
[2] B. LuBow, "Correlation entering new field with real-time signal analysis," *Electronics*, vol. 31, pp. 75–81, Oct. 1966.
[3] J. E. Abate, "Linear and adoptive delta modulation," *Proc. IEEE*, vol. 55, pp. 298–307, Mar. 1967.
[4] M. R. Winkler, "High information delta modulation," in *IEEE Int. Conv. Rec.*, pt. 8, 1963, pp. 260–265.
[5] A. Peled and B. Liu, "A new approach to the realization of nonrecursive digital filters," *IEEE Trans. Audio Electroacoust.*, vol. AU-21, pp. 477–484, Dec. 1973.
[6] N. S. Jayant, "On the power spectrum of the staircase function in linear delta modulation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-23, pp. 162–168, Apr. 1975.

Implementation of the Digital Phase Vocoder Using the Fast Fourier Transform

MICHAEL R. PORTNOFF, STUDENT MEMBER, IEEE

Abstract—**This paper discusses a digital formulation of the phase vocoder, an analysis-synthesis system providing a parametric representation of a speech waveform by its short-time Fourier transform. Such a system is of interest both for data-rate reduction and for manipulating basic speech parameters. The system is designed to be an identity system in the absence of any parameter modifications. Computational efficiency is achieved by employing the fast Fourier transform (FFT) algorithm to perform the bulk of the computation in both the analysis and synthesis procedures, thereby making the formulation attractive for implementation on a minicomputer.**

I. Introduction

THE REPRESENTATION of a speech signal by its short-time Fourier transform is of interest both as a means for data-rate reduction in communications and as a technique for manipulating the basic speech parameters. Systems based on this representation are often referred to as phase vocoders since the parameters obtained have traditionally been the magnitude and phase (or phase-derivative) of the short-time Fourier transform [1].

One difficulty in implementing such systems in digital form has been the rapid increase in the amount of computation required as the number of frequency bands is made large. Schafer and Rabiner [2] have shown how to greatly reduce the amount of computation required for the analysis procedure by formulating the system such that most of the computation is performed by the fast Fourier transform (FFT) algorithm. However, the computation required for the direct implementation of the synthesis procedure is at least as great as that required for the direct analysis, and it has, therefore, remained a problem.

In this paper, we present an analysis-synthesis system based on the discrete short-time Fourier transform. This system will be shown to be, mathematically, an identity system if no parameter modifications are introduced. The analysis procedure is a refinement of that proposed by Schafer and Rabiner in which the complex multiplies used to demodulate the channel signals are now eliminated. The synthesis procedure is new and is significantly more efficient than the direct procedure [2]. The computational savings is effected by reducing the number of interpolations required for each output value from N (where N is the number of frequency bands in the representation) to 1 and by performing the remaining computations using the FFT algorithm (a savings of approximately $\log_2 N$ versus N operations per output value).

II. Formulation

Let $x(n)$ represent samples of a speech waveform. The discrete short-time Fourier transform of $x(n)$ is defined by

$$X_k(n) = \sum_{r=-\infty}^{\infty} x(r)\, h(n-r)\, W_N^{-rk} \tag{1}$$

for $k = 0, 1, \cdots, N-1$, where $W_N = \exp\,[j(2\pi/N)]$ and $h(n)$

Manuscript received May 10, 1975; revised December 9, 1975. This work was supported by the Advanced Research Projects Agency, monitored by the ONR under Contract N00014-75-C-0951.

The author is with the Department of Electrical Engineering and Computer Science, Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, MA 02139.

is an appropriately chosen window. $X_k(n)$ may be interpreted as N samples of a time-varying spectrum with k the index associated with frequency and n the index associated with time. According to (1), $X_k(n)$ is obtained at each time sample n by weighting the sequence $x(r)$ by the window $h(n-r)$ and Fourier transforming the resulting sequence. In the next section it will be shown how to obtain $X_k(n)$ at a particular n by computing a single discrete Fourier transform (DFT) of a finite-duration sequence of length N.

By properly choosing $h(n)$, it can be guaranteed that the original sequence $x(n)$ is exactly recoverable from its short-time transform defined by (1). Furthermore, $x(n)$ is given in this case by

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X_k(n) W_N^{nk} \qquad \text{for all } n. \tag{2}$$

Although the necessary and sufficient conditions for $x(n)$ to be given by (2) can be derived directly from (1), it is informative to interpret (1) and (2) in terms of a bank of digital bandpass filters with contiguous passbands. Consider a set of N complex bandpass filters $\{h_k(n)\}$ with passbands equally spaced about the unit circle and with unit-sample responses

$$h_k(n) = \frac{1}{N} h(n) W_N^{nk}, \qquad k = 0, 1, \cdots, N-1, \tag{3}$$

where $h(n)$ is a prototype low-pass filter with real unit-sample response. If these filters are combined to form the structure shown in Fig. 1, then the output of the kth filter, denoted by $y_k(n)$, is given by the convolution

$$\begin{aligned} y_k(n) &= \sum_{r=-\infty}^{\infty} x(r) h_k(n-r) \\ &= \sum_{r=-\infty}^{\infty} x(r) \left[\frac{1}{N} h(n-r) W_N^{(n-r)k} \right] \\ &= \frac{1}{N} W_N^{nk} \sum_{r=-\infty}^{\infty} x(r) h(n-r) W_N^{-rk} \\ &= \frac{1}{N} W_N^{nk} X_k(n), \end{aligned} \tag{4}$$

where $X_k(n)$ is just the discrete short-time Fourier transform of $x(n)$ given by (1). From (1) and (4) a single channel of the filter bank is seen to be equivalent to the structure shown in Fig. 2.

The output of the filter bank $y(n)$ is given by the sum of the N channels $y_k(n)$, i.e.,

$$\begin{aligned} y(n) &= \sum_{k=0}^{N-1} y_k(n) \\ &= \frac{1}{N} \sum_{k=0}^{N-1} X_k(n) W_N^{nk}. \end{aligned}$$

It is, therefore, clear that if $x(n)$ is to be recovered from $X_k(n)$ by means of (2), then $h(n)$ must be chosen in such a manner that the output $y(n)$ is identical to the input $x(n)$.

The filter-bank system depicted in Fig. 1 is linear and shift

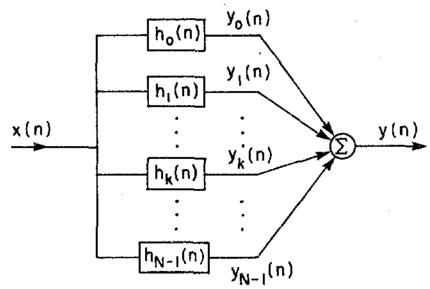

Fig. 1. Digital filter-bank analog for discrete short-time Fourier analysis.

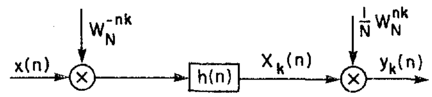

Fig. 2. Representation of the kth filter-bank channel in terms of the prototype low-pass filter $h(n)$.

invariant and thus completely characterized by its unit-sample response. Let $\tilde{h}(n)$ represent the overall unit-sample response relating the output $y(n)$ of the filter bank to the input $x(n)$. Then

$$\begin{aligned} \tilde{h}(n) &= \sum_{k=0}^{N-1} h_k(n) \\ &= \sum_{k=0}^{N-1} \frac{1}{N} h(n) W_N^{nk} \\ &= h(n) \left[\frac{1}{N} \sum_{k=0}^{N-1} W_N^{nk} \right] \\ &= h(n) \left[\frac{1}{N} \frac{1 - W_N^{nN}}{1 - W_N^{n}} \right] \\ &= h(n) \delta((n))_N, \end{aligned}$$

where $\delta((n))_N = 1$ for all $n \equiv 0 \bmod N$ and is zero otherwise. Thus, $\tilde{h}(n)$ is simply the unit-sample response $h(n)$ of the prototype low-pass filter sampled every N samples, specifically

$$\tilde{h}(n) = \begin{cases} h(n) & \text{for } n = 0, \pm N, \pm 2N, \cdots \\ 0 & \text{otherwise.} \end{cases}$$

Now if $y(n)$ is to be identically equal to $x(n)$, then $\tilde{h}(n)$ must itself be a unit sample. Therefore, necessary and sufficient conditions for $y(n) = x(n)$ for all n are[1] as follows.

1) $h(0) = 1$.

[1]This result also follows directly from (1) by multiplying (1) by $(1/N)\, W_N^{nk}$ and summing over k for $0 \leqslant k \leqslant N-1$ to obtain

$$\begin{aligned} \frac{1}{N} \sum_{k=0}^{N-1} X_k(n) W_N^{nk} &= \frac{1}{N} \sum_{k=0}^{N-1} \sum_{r=-\infty}^{\infty} x(r) h(n-r) W_N^{-rk} W_N^{nk} \\ &= \sum_{r=-\infty}^{\infty} x(r) h(n-r) \left[\frac{1}{N} \sum_{k=0}^{N-1} W_N^{(n-r)k} \right] \\ &= \sum_{q=-\infty}^{\infty} x(n+qN) h(-qN) \\ &= x(n) \text{ iff (5).} \end{aligned}$$

2) $h(n) = 0$ for $n = \pm N, \pm 2N, \pm 3N, \cdots$. (5)

These conditions are equivalent to the statement in the frequency domain that although each $h_k(n)$ is not necessarily an ideal bandpass filter, the sum of their N frequency responses is unity for all frequencies. Observe that the conditions (5) are precisely those constraints on the unit-sample response of a digital interpolating filter [3]. Moreover, if these conditions are not satisfied, then $\tilde{h}(n)$ will no longer be a unit sample, but a weighted sequence of unit samples with spacing N; hence $y(n)$ will not be identical to $x(n)$ and the resulting distortion will be perceived as reverberation in the output signal.

The most straightforward approach to designing the prototype low-pass filter $h(n)$ is by windowing [4]. Specifically, the unit-sample response

$$h_{\text{ideal}}(n) = \frac{\sin(n\pi/N)}{n\pi/N}$$

of an ideal low-pass filter with cutoff frequencies $\Omega_c = \pm(\pi/N)$ is multiplied by a smooth, finite-duration window (e.g., Hamming [5], Kaiser [6], Dolph-Chebyshev [7]) to obtain $h(n)$. The precise specifications of $h(n)$ are determined by the length and shape of the window; any $h(n)$ designed in this manner will satisfy conditions (5).

Alternatively, one might employ one of the recently proposed techniques for designing optimum (minimax) equiripple finite impulse response (FIR) interpolating filters [3], [8]. However, for a large number of frequency samples, the long length required for $h(n)$ tends to make these filters prohibitively expensive to design. Furthermore, the additional amount of computation incurred by using a suboptimum $h(n)$ designed by windowing is probably small compared with the total amount of computation in the overall system.

The short-time Fourier transform provides a parametric representation of the sequence $x(n)$ in terms of the parameters $X_k(n)$. If $X_k(n)$ is computed for $k = 0, 1, \cdots, N-1$ and for all n, then N complex parameters are required for each sample of $x(n)$. If $x(n)$ is real, then this represents an increase in complexity by a factor of $2N$. There are, however, properties of the discrete short-time Fourier transform that can be exploited to reduce the number of parameters required to represent $x(n)$ to an average of approximately one per sample of $x(n)$. First, if $X_k(n)$ is viewed for a particular value of n as N equally spaced samples of a Fourier transform, then, since $x(n)$ is assumed to be real, $X_k(n)$ is conjugate symmetric in k; that is,

$$X_k(n) = X^*_{((N-k))_N}(n)$$

where $((n))_N$ denotes the least residue of n modulo N. Thus, if N is even, $X_k(n)$ is completely specified by the values of $X_k(n)$ for $k = 0, 1, \cdots, N/2$, and only N real parameters are required (n.b., $X_k(n)$ is real for $k = 0$ and $k = N/2$). The second property of $X_k(n)$ that allows a further reduction in the number of parameters required to represent $x(n)$ is apparent when $X_k(n)$ is viewed for a particular value of k as a sequence in n. From Fig. 2 it can be seen that because it is the output of a low-pass filter with unit-sample response $h(n)$, each such sequence is approximately band-limited to the frequency range $-\pi/N < \Omega < \pi/N$. Thus, it follows from the sampling theorem that it is only necessary to compute $X_k(n)$ for every Rth value of n, where $R \leqslant N$. The sequences $X_k(n)$ can then be reconstructed by interpolation as part of the synthesis procedure.

If the sampling period R is chosen equal to N, which corresponds to the lowest sampling rate allowed by the sampling theorem, then the total number of real parameters in the short-time Fourier representation of $x(n)$ is exactly equal to the duration (total number of samples) of $x(n)$.[2] Although it is theoretically possible to reconstruct the sequences $X_k(n)$ if they are sampled every $R = N$ samples, in practice it is necessary to sample at a somewhat higher rate, because neither the low-pass filter nor the interpolator can be implemented ideally.

A procedure that is particularly well suited to designing interpolating filters for reconstructing the channel sequences is the algorithm proposed by Oetken *et al.* [9] for designing optimal FIR digital interpolating filters. This procedure is attractive because it is a simple and efficient procedure for designing filters of very high order. Furthermore, the design algorithm exploits the fact that the data to be interpolated can be oversampled to improve the performance of the filter.

III. Implementation of the Analysis System Using the FFT Algorithm

If the number of frequency bands N is chosen to be a highly composite number (usually an integral power of 2) then the FFT algorithm can be employed to compute efficiently the short-time Fourier transform $X_k(n)$ defined by (1). Observe that (1) does not have the form of a DFT and, therefore, cannot be computed directly with the FFT algorithm. The limits on the summation are given as infinite, but in practice are finite and determined by the length of $h(n)$. By recognizing $X_k(n)$ as samples, equally spaced in frequency, of the (continuous-valued) Fourier transform of $x(r)h(r-n)$, $X_k(n)$ can be expressed as the DFT of an N-point sequence obtained by time-domain aliasing of $x(r)h(n-r)$.

Substituting $s = r - n$ into (1) gives

$$X_k(n) = \sum_{s=-\infty}^{\infty} x(n+s)\,h(-s)\,W_n^{-(n+s)k}$$

$$= W_N^{-nk} \sum_{s=-\infty}^{\infty} x(n+s)\,h(-s)\,W_N^{-sk},$$

which can be rewritten as

$$X_k(n) = W_N^{-nk} \sum_{l=-\infty}^{\infty} \sum_{m=0}^{N-1} x(n+lN+m)\,h(-lN-m) \cdot W_N^{-(lN+m)k}$$

by taking $s = lN + m$ for $m = 0, 1, \cdots, N-1$ and $l = -\infty, \cdots, -1, 0, +1, \cdots, \infty$. Interchanging the orders of summation and using $W_N^N = 1$ gives

$$X_k(n) = W_N^{-nk} \sum_{m=0}^{N-1} \sum_{l=-\infty}^{\infty} x(n+lN+m)\,h(-lN-m)\,W_N^{-mk}$$

or

[2]When this representation is used as a vocoder, data-rate reduction is achieved by quantizing the parameters $X_k(n)$ [2].

$$X_k(n) = W_N^{-nk} \sum_{m=0}^{N-1} \tilde{x}_m(n) W_N^{-mk}, \tag{6}$$

where

$$\tilde{x}_m(n) = \sum_{l=-\infty}^{\infty} x(n + lN + m) h(-lN - m). \tag{7}$$

The expression

$$\tilde{X}_k(n) = \sum_{m=0}^{N-1} \tilde{x}_m(n) W_N^{-mk}$$

is recognized as the DFT of the N-point (in m) sequence $\tilde{x}_m(n)$ for fixed n and can, therefore, be computed directly with the FFT algorithm once $\tilde{x}_m(n)$ has been formed.

In addition to the computational savings gained by computing the short-time Fourier transform using the FFT, further savings may be gained by avoiding the complex multiplications by W_N^{-nk} in (6). Observing that $X_k(n)$ is given by

$$X_k(n) = W_N^{-nk} \tilde{X}_k(n),$$

where $\tilde{X}_k(n)$ is the DFT of $\tilde{x}_m(n)$, we can exploit the property of the DFT that a circular shift in one domain corresponds to multiplication by a complex exponential in the other domain. Thus, by circularly shifting $\tilde{x}_m(n)$ prior to computing its DFT, the multiplications by W_N^{-nk} are avoided. Specifically, (6) can be rewritten as

$$X_k(n) = \sum_{m=0}^{N-1} \tilde{x}_{((m-n))_N}(n) W_N^{-mk}$$

or

$$X_k(n) = \sum_{m=0}^{N-1} x_m(n) W_N^{-mk} \tag{8}$$

where

$$x_m(n) = \tilde{x}_{((m-n))_N}(n).$$

Based on the preceding analysis, the procedure for computing the discrete short-time Fourier transform coefficients $X_k(n)$ at a particular value of n is the following. Referring to Fig. 3, the input data sequence considered as a function of the dummy index r is multiplied by the window $h(n - r)$ (in practice $h(n)$ is often zero phase, in which case $h(n - r) = h(r - n)$). It is assumed that $h(n)$ is of finite duration and, in fact, chosen to have length equal to an even multiple of N, plus one. The resulting weighted sequence is partitioned into sections each of length N such that $x(r)|_{r=n}$ is the zeroth sample of one of the sections. The resulting N-point subsequences denoted by $x_m^{(l)}(n)$ for $0 \leqslant m \leqslant N - 1$ are then added together to form

$$\tilde{x}_m(n) = \sum_l x_m^{(l)}(n), \qquad m = 0, 1, \cdots, N - 1.$$

$\tilde{x}_m(n)$ is circularly shifted (in m) by n samples to obtain

$$x_m(n) = \tilde{x}_{((m-n))_N}(n),$$

and its DFT is computed by means of the FFT algorithm to give the desired $X_k(n)$, i.e.,

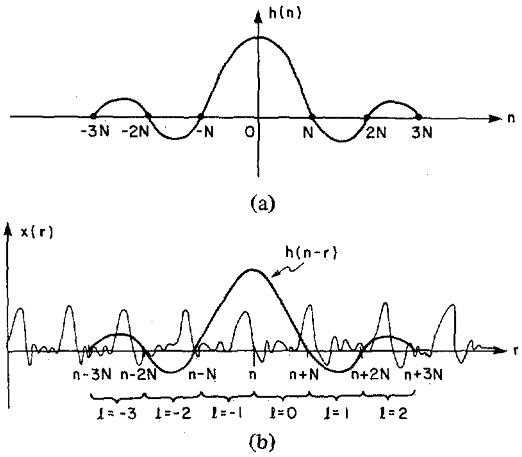

Fig. 3. (a) Typical unit-sample response for prototype low-pass filter $h(n)$. (b) $h(n)$ shifted and superimposed on input sequence $x(r)$.

$$X_k(n) = \sum_{m=0}^{N-1} x_m(n) W_N^{-mk} \qquad k = 0, 1, \cdots, N - 1.$$

IV. Implementation of the Synthesis System Using the FFT Algorithm

It has been shown that for any $h(n)$ satisfying conditions (5) the sequence $x(n)$ can be recovered from its discrete short-time Fourier transform by the relation

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X_k(n) W_N^{nk}. \tag{2}$$

According to Fig. 2, this operation may be interpreted as modulating each of the N signals $X_k(n)$ to the center frequencies $\Omega_k = 2\pi k/N$ and summing the resulting signals. It was argued in Section II that it is only necessary to compute $X_k(n)$ for every Rth value of n where $R \leqslant N$. Hence, the parameters to the synthesizer will be assumed to be the samples $X_k(rR)$ and not $X_k(n)$.

Clearly, each of the N signals $X_k(rR)$ could be interpolated to get $X_k(n)$, which could then be used in (2) to compute $x(n)$ directly [2]. Unfortunately, since $X_k(n)$ depends on n, (2) does not have the form of an (inverse) DFT and is computationally intractable for large values of N.

A synthesis procedure will now be formulated which, for a highly composite number N, permits $x(n)$ to be computed from the samples $X_k(rR)$ using the FFT algorithm. In addition to the computational savings afforded by employing the FFT, the number of computations required to perform the $1{:}R$ interpolation is reduced by the factor N.

Let the input parameters to the synthesizer be denoted by $S_k(r)$, where

$$S_k(r) = X_k(rR) \qquad \text{for all } r \text{ and } k = 0, 1, \cdots, N - 1.$$

Let $f(n)$ represent the unit-sample response of a $1{:}R$ FIR interpolating filter with length $2QR + 1$. The interpolated signals $X_k(n)$ are, therefore, given by

$$X_k(n) = \sum_{r=L^-}^{L^+} f(n - rR) S_k(r), \tag{9}$$

where the limits on the sum, determined by the length of $f(n)$, are

$$L^+(n) = \left[\frac{n}{R}\right] + Q$$

$$L^-(n) = \left[\frac{n}{R}\right] - Q + 1,$$

and where $[M]$ means "the largest integer contained in M." Substituting $X_k(n)$ given by (9) into the synthesis equation (2) gives

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1}\left\{\sum_{r=L^-}^{L^+} f(n-rR)S_k(r)\right\} W_N^{nk}.$$

Since the limits on both sums are finite, the order of summation can be interchanged to give

$$x(n) = \sum_{r=L^-}^{L^+} f(n-rR)\left\{\frac{1}{N}\sum_{k=0}^{N-1} S_k(r)\, W_N^{nk}\right\}$$

or

$$x(n) = \sum_{r=L^-}^{L^+} f(n-rR)\, s_n(r), \tag{10}$$

where

$$s_n(r) = \frac{1}{N}\sum_{k=0}^{N-1} S_k(r)\, W_N^{nk}. \tag{11}$$

Thus, for fixed values of r, $s_n(r)$ is the inverse DFT of $S_k(r)$ and can, therefore, be computed by the FFT algorithm. It is important to observe that $s_n(r)$ is periodic in n with period N. Since the FFT only computes values of $s_n(r)$ for one period ($n = 0, 1, \cdots, N-1$), it is necessary to interpret the subscript n in (11) as reduced modulo N.

The synthesis procedure implied by (10) and (11) can be interpreted as follows. Consider the two-dimensional "net" shown in Fig. 4. The points on the net represent the discrete set of points on which $X_k(n)$ is defined. The horizontal direction represents time and the vertical frequency. The points corresponding to the values of $X_k(n)$ available to the synthesizer, i.e., every Rth column, are indicated by shading. Inverting (8) gives $x_m(n)$ as the inverse DFT of $X_k(n)$ for each n, i.e.,

$$x_m(n) = \frac{1}{N}\sum_{k=0}^{N-1} X_k(n)\, W_N^{mk}. \tag{12}$$

Furthermore, $x_m(n)$ is defined on the net shown in Fig. 5. Because $S_k(r) = X_k(rR)$, it follows that $s_m(r) = x_m(rR)$ and, therefore, $s_m(r)$ is defined on the shaded points in Fig. 5. By comparing (12) with (2), it can be seen that the values of $x(n)$ are given by the values of $x_m(n)|_{m \equiv n \bmod N}$, which correspond to the points in Fig. 5 on the "helical" path $m \equiv n \bmod N$. The operation defined by (10) is, therefore, interpreted as interpolating $s_m(r)$ to obtain the unknown values of $x_m(n)$, but only those values of $x_m(n)$ on the path $m \equiv n \bmod N$ that are the values of $x(n)$.

The implementation of the synthesis procedure is, therefore,

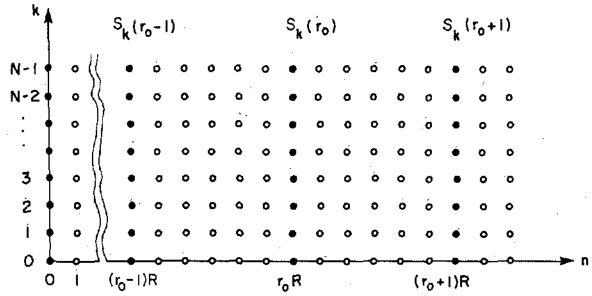

Fig. 4. Net on which $X_k(n)$ is defined. Shaded points represent values associated with $S_k(r) = X_k(rR)$.

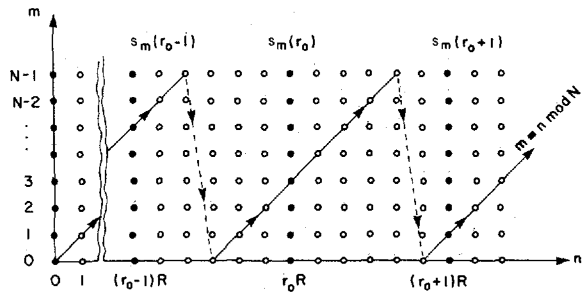

Fig. 5. Net on which $x_m(n)$ is defined. Shaded points represent values associated with $s_m(r) = x_m(rR)$. Values along path $m \equiv n \bmod N$ are $x(n) = x_n(n)$.

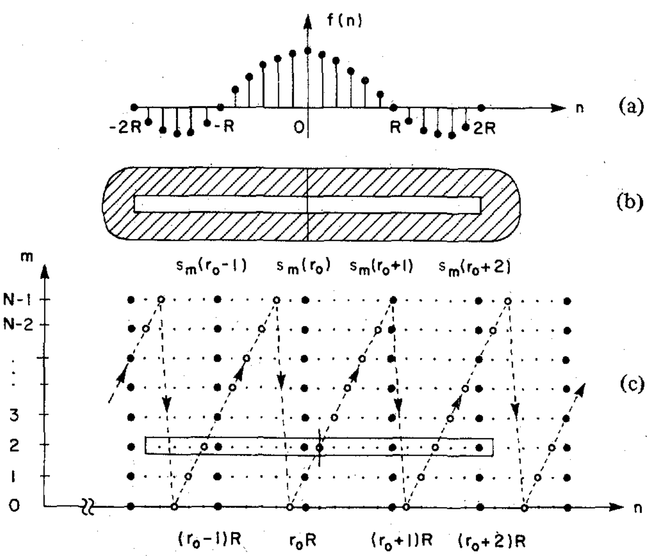

Fig. 6. (a) Typical unit-sample response for 1:R FIR digital interpolating filter. (b) Mask to extract values required for interpolation using $f(n)$. (c) Net associated with $x_m(n)$. • indicates points representing $s_m(r) = x_m(rR)$. ○ indicates points representing $x(n) = x_n(n)$.

as follows. First, the values of $s_m(r)$ are obtained by inverse transforming $S_k(r)$ using the FFT (11). The values of $x(n)$ are then obtained by interpolating $s_m(r)$ according to (10). Notice that for each value of $x(n)$, $2Q$ values of $s_m(r)$ are required. In fact, for R consecutive values of $x(n)$, these values are obtained from the same $2Q$ columns. Thus, it is natural to compute $x(n)$ in records of length R. For each output value, imagine a mask that extracts $2Q$ values of $s_m(r)$, as shown in Fig. 6. These values are then processed according to (10) to compute $x(n)$. Successive output values are obtained by shifting the mask one sample at a time along the path $m \equiv n \bmod N$ and repeating the process.

V. Conclusions

We have discussed a new implementation of the digital phase vocoder, a system that provides a parametric representation of a sequence in terms of its discrete short-time Fourier transform. If no parameter modifications are introduced, the system has been shown to be, mathematically, an identity system. The bulk of the computation in both the analysis and synthesis procedures is performed by the FFT, thereby making the system attractive for implementation on a minicomputer (especially if a high-speed FFT processor is available).

The system described has been implemented on a PDP-9 computer using block floating-point arithmetic. The system is being used to modify certain parameters of speech signals and currently allows as many as 512 frequency channels. When operated as an identity system, the synthesized output differs in no perceptual or measurable way from the input speech.

Acknowledgment

The author wishes to thank his advisor, Prof. A. V. Oppenheim, who carefully read and commented on this manuscript during its preparation.

References

[1] J. L. Flanagan and R. M. Golden, "Phase vocoder," *Bell Syst. Tech. J.*, vol. 45, pp. 1493-1509, Nov. 1966.
[2] R. W. Schafer and L. R. Rabiner, "Design and simulation of a speech analysis-synthesis system based on short-time Fourier analysis," *IEEE Trans. Audio Electroacoust. (Special Issue on 1972 Conference on Speech Communication and Processing)*, vol. AU-21, pp. 165-174, June 1973.
[3] —, "A digital signal processing approach to interpolation," *Proc. IEEE*, vol. 61, pp. 692-702, June 1973.
[4] R. W. Schafer, L. R. Rabiner, and O. Herrmann, "FIR digital filter banks for speech analysis," *Bell Syst. Tech. J.*, vol. 54, pp. 531-544, Mar. 1975.
[5] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975.
[6] J. F. Kaiser, "Nonrecursive digital filter design using the I_0-sinh window function," in *Proc. 1974 IEEE Int. Symp. Circuits and Systems*, San Francisco, CA, Apr. 1974, pp. 20-23.
[7] H. D. Helms, "Nonrecursive digital filters: Design methods for achieving specifications on frequency response," *IEEE Trans. Audio Electroacoust. (Special Issue on Digital Filters: The Promise of LSI to Signal Processing)*, vol. AU-16, pp. 336-342, Sept. 1968.
[8] H. S. Hersey and R. M. Mersereau, "An algorithm to perform minimax approximation in the absence of the Haar condition," M.I.T. Res. Lab. Electron., Cambridge, MA, Quarterly Progress Rep. 114, pp. 157-160, July 15, 1974.
[9] G. Oetken, T. W. Parks, and H. W. Schüssler, "New results in the design of digital interpolators," *IEEE Trans. Acoust., Speech, Signal Processing (Special Issue on 1974 Arden House Workshop on Digital Signal Processing)*, vol. ASSP-23, pp. 301-309, June 1975.

Limit Cycles in the Combinatorial Implementation of Digital Filters

TRÂN-THÔNG, MEMBER, IEEE, AND BEDE LIU, FELLOW, IEEE

Abstract—**The existence of limit cycles in combinatorial filters using two's complement truncation arithmetic is investigated in this paper. Exact results for limit cycles of period one and two are presented. Some results for longer period limit cycles are obtained using an effective value linear model. Bounds on these limit cycles are also derived. The accessability of the limit cycles is briefly discussed.**

I. Introduction

Combinatorial filters appeared recently in the literature [1], [2] as an alternative method for implementing digital filters. These filters do not employ hardware multipliers. Instead, the computation is carried out with read only memory (ROM) and an accumulator. Consequently, they offer considerable saving in hardware and power consumption with the potential for increased operating speed.

This paper is concerned with the stability of combinatorial filters under zero input condition. The problem is different from most of the past work on limit cycles [3]-[5] in that the combinatorial filter can be modeled as a digital filter with only one quantizer in each section instead of the usual one quantizer with each multiplier. The stability of an idealized filter structure with one quantizer using either sign-magnitude truncation or rounding arithmetic has been reported recently [6], [7], and the results are applicable to combinatorial filters using these two types of arithmetic. However, as a result of the elimination of multipliers in these filters, an implementation with two's complement is easier [1], [2]. In two's complement arithmetic the variance of the roundoff noise in rounding and in truncation are the same. However, in the latter case, there is a dc offset which is easily computed and can be removed in the final conversion to an analog signal.

Manuscript received March 24, 1975; revised September 3, 1975 and December 20, 1975. This work was supported by the Air Force Office of Scientific Research, USAF, under Grant AF-AFOSR 71-2101.

T.-Thông was with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08540. He is now with the Western Geophysical Company, Houston, TX 77001.
B. Liu is with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08540.