RZ 918 (#31394) 9/5/78
Communications 29 pages

Research Report

ON REDUCING THE NUMBER OF OPERATIONS IN ADAPTIVE EQUALIZERS

Dietrich Maiwald, Hans Peter Kaeser and Felix Closs

IBM Zurich Research Laboratory, 8803 Rüschlikon, Switzerland

Typed by: A. Schönholzer on MT 82

IBM Research Division
San Jose · Yorktown · Zurich

RZ 918 (#31394) 9/5/78
Communications 29 pages

ON REDUCING THE NUMBER OF OPERATIONS IN ADAPTIVE EQUALIZERS

Dietrich Maiwald, Hans Peter Kaeser and Felix Closs

IBM Zurich Research Laboratory, 8803 Rüschlikon, Switzerland

Typed by: A. Schönholzer on MT 82

ABSTRACT: — In adaptive equalization, the coefficients of a transversal filter are automatically adjusted according to a minimum mean-square error criterion. The convolution for the filter process and the cross-correlation in the coefficient-adjustment algorithm require a large number of arithmetic operations. Recent advances in the theory of computational complexity have yielded efficient methods which allow that processing load to be significantly reduced. This paper describes an equalizer for complex signals in which filtering and coefficient adjustment are performed in the frequency domain employing Winograd's Fourier transform algorithms. Compared to a conventional time-domain implementation, the number of multiplications is reduced by typically a factor four without increasing the number of additions. A fast initialization algorithm will be presented which overcomes the somewhat slower adjustment of the equalizer due to the blockwise processing of signal samples. The convergence behavior of the adaptive equalizer will be shown for random data signals transmitted over a realistic telephone channel.

August 24, 1978

I. INTRODUCTION

IN THE DESIGN of voiceband modems a trend can be observed to implement analog signal processing functions of the transmitter and receiver with digital processors [1], [2]. A major problem in devising digital modems, especially the receiver section, is the high processing speed required even for a moderate data-transmission rate of 4800 bps. Convolutions and correlations to be computed for filtering and equalization must be performed on complex-valued signals (in-phase and quadrature components) and thus demand a large number of multiplications per unit of time. This problem can be solved by special fast-multiplier hardware which, however, contributes significantly to total system cost. As an alternate solution addressed in the sequel, one can apply the computationally very efficient algorithms for calculating convolutions and correlations which have been published recently [3]-[5].

This paper concentrates on adaptive equalization, the most demanding receiver function in terms of multiplications and additions. Adaptive equalizers are used in modem receivers for data rates above 2400 bps to compensate the linear distortion of the telephone channel, Fig. 1. The channel characteristics are usually unknown and subject to variations. The equalizer is, therefore, automatically adjusted by a gradient method which attempts to minimize intersymbol interference and noise at its output [6]. Fast convergence of this adjustment process is crucial for achieving short receiver start-up delay.

A computationally efficient implementation will be presented which, e.g., for a 20-tap equalizer reduces the number of multiplications by a factor 3.6 and the additions by 7% compared to conventional designs. These savings in the number of arithmetic operations are achieved by performing the convolution and correlation required in adaptive equalization [7], [8] in the discrete frequency domain. It is well known that long convolutions and correlations can be executed more efficiently in the frequency domain than in the time domain [9]. The advent of very powerful algorithms for computing the Discrete Fourier Transform (DFT) [3] makes it attractive to apply frequency-domain processing also to the relatively short convolutions/correlations used in modem receivers where the number of coefficients hardly exceeds 40.

Walzman [10],[11] first investigated adaptive equalization in the frequency domain. Using isolated test pulses as input signal he showed that the frequency-domain equalizer converges significantly faster than its time-domain counterpart. In this paper we address convergence speed when random data sequences are transmitted. Signal waveforms measured on heavily distorted telephone channels are taken as input to the equalizer. Results indicate that with these signals the frequency-domain equalizer converges slower than the time-domain implementation. However, as will be described, this drawback can be overcome by incorporating a method for fast initial start-up of the equalizer.

In Section II of the paper various algorithms for performing linear convolution (fixed-coefficient filtering) are reviewed and their computational complexity estimated. In Section III the most promising method, frequency-domain processing, is extended to perform adaptive filtering. Section IV addresses the computational load of the proposed frequency-domain approach to adaptive equalization when using Winograd's Fourier transform algorithms. Section V describes a fast initial start-up procedure and compares by simulation the convergence behavior of the adaptive equalizer with that of a conventional time-domain realization. In a concluding section the results are summarized and some directions are indicated for future research.

II. COMPUTATIONAL EFFICIENCY OF ALGORITHMS FOR LINEAR CONVOLUTION

Digital fixed-coefficient filtering implies convolving the input sequence $\{x_n\}$, which commonly is a sampled analog signal, with the unit-sample response $\{h_n\}$ of the filter thus yielding the output sequence $\{y_n\}$. Assuming a filter with an impulse response of M samples, e.g., a transversal filter having M taps, the process can be described by the digital convolution sum

$$y_n = \sum_{k=0}^{M-1} x_k \, h_{n-k} \; . \qquad (1)$$

In data-transmission applications which involve quadrature amplitude modulation, the signal samples x_n, y_n as well as the filter coefficients h_n usually have complex values (in-phase and quadrature components) [12].

In the following subsections several methods for calculating the complex-valued convolution sum (1) will be described and their computational efficiency estimated. A primary criterion for the computational load produced by executing an algorithm, is the number of multiplications and additions of real-valued numbers which must be performed. Beyond this measure the designer of a digital signal-processing system is ultimately interested in criteria like number of machine cycles and data/program storage size, which take into account also the amount of control necessary for implementing the algorithm. However, reference to these parameters depends on the processor architecture and hence would be of limited generality.

In this paper we, therefore, compare the number of multiplications and additions required to compute a specified processing function according to some given algorithm. Particular computational aspects like round-off errors or manipulation of trigonometric functions which influence the numerical precision of the result, will not be considered.

II.1 Direct Computation of the Digital Convolution

For complex variables equation (1) can be rewritten as

$$
\begin{aligned}
y_n^r &= \sum_{k=0}^{M-1} (x_k^r h_{n-k}^r - x_k^i h_{n-k}^i) \\
y_n^i &= \sum_{k=0}^{M-1} (x_k^i h_{n-k}^r + x_k^r h_{n-k}^i) ,
\end{aligned}
\qquad (2)
$$

where the superscripts r and i represent the real and imaginary parts of the complex samples, respectively. Note that all operations in (2) now involve real-valued numbers. Direct calculation of the convolution according to the defining formula (2) requires

$$4\,M \quad \text{multiplications/output sample}$$
$$4(M-1) + 2 = 4M-2 \quad \text{additions/output sample.} \tag{3}$$

An interesting alternative to the computing load (3) can be derived when applying to (2) G. Golub's method [13] for complex multiplication, which trades one of the four corresponding real multiplications with three extra additions:

$$y_n^r = \underbrace{\sum_{k=0}^{M-1}\left[(x_k^r + x_k^i)(h_{n-k}^r - h_{n-k}^i)\right]}_{I} + \underbrace{\sum_{k=0}^{M-1} x_k^r\, h_{n-k}^i}_{II} - \underbrace{\sum_{k=0}^{M-1} x_k^i\, h_{n-k}^r}_{III}$$
$$y_n^i = \sum_{k=0}^{M-1} (x_k^r\, h_{n-k}^i + x_k^i\, h_{n-k}^r) = II + III\,. \tag{4}$$

In case of a fixed-coefficient filter the term $h_{n-k}^r - h_{n-k}^i$ can be precomputed and stored; if previous values of $x_k^r + x_k^i$ are memorized the summation yields one extra addition per incoming sample. Thus direct calculation of (4) requires only

$$3\,M \quad \text{multiplications/output sample}$$
$$3(M-1) + 4 = 3M+1 \quad \text{additions/output sample}\,. \tag{5}$$

The computational loads Eqs. (3) and (5) for time-domain convolution are plotted as functions of the filter length M in Fig. 2 (number of multiplications) and in Fig. 3 (number of additions), respectively.

II.2 Digital Convolution Using Multidimensional Transform Techniques

In a recent paper [4] the Chinese Remainder Theorem was used for mapping one-dimensional convolutions into multidimensional convolutions which are cyclic in all dimensions. Application of efficient rectangular transform algorithms for computing the relatively short convolutions in each of the dimensions then allows the number of operations for performing a digital circular convolution to be reduced.

In the processing application considered in this paper, the impulse response of the filter $\{h_n\}$ is typically of short length M, while the input sequence $\{x_n\}$ can be regarded as indefinitely long. Fixed-coefficient digital filtering of this sort can be implemented with transforms having the circular convolution property by applying the conventional overlap-add or overlap-save techniques [14]. The former technique implies blocking the input sequence in sections of L samples, using a circular transform of length $N \geq L+M-1$, and reconstructing the filtered output by adding overlapping samples. For each filter length M there is an optimum N, depending on the cyclic convolution scheme used, which requires the minimum amount of computation per output point. As will become evident in Section III we must nevertheless assume the fixed relation $N = 2M$ in the sequel and thus require $N-M = M$ extra additions per block of M output samples for performing the overlap-add process.

In order to evaluate convolution (1) of complex samples using the transform method described above and real-valued 2M-point circular convolution algorithms, two computational alternatives exist:

(i) When applying the conventional expansion (2) the calculation takes

$$
\begin{array}{l}
\text{4 times the operations of a 2M-point circular} \\
\qquad\qquad \text{convolution per } M \text{ output samples} \\
\text{4 additions per output sample for the} \\
\qquad\qquad\qquad \text{overlap-add process} \\
\text{2 additions per output sample for real and} \\
\qquad\qquad\qquad\qquad \text{imaginary parts.}
\end{array}
\tag{6}
$$

(ii) If formulation (4) is used and the term $h^r_{n-k} - h^i_{n-k}$ precomputed and stored, the amount of computation reduces to

$$
\begin{array}{l}
\text{3 times the operations of a 2M-point circular} \\
\qquad\qquad \text{convolution per } M \text{ output samples} \\
\text{3 additions per output sample for the} \\
\qquad\qquad\qquad \text{overlap-add process} \\
\text{4 additions per output sample for real and} \\
\qquad\qquad\qquad\qquad \text{imaginary parts.}
\end{array}
\tag{7}
$$

The number of multiplications and additions per output point for circular convolutions of real-valued samples can be taken from Table III of [4] for various transform lengths N, assuming that the filter response is fixed and that its transform has been precomputed and stored. Using these numbers Eqs. (6) and (7) have been evaluated for several filter lengths M. The results are displayed in Figs. 2 and 3. Note that efficient Agarwal & Cooley algorithms exist only for distinct values $N = 2M$ of transform points which are highly composite numbers; the interconnecting lines between individual results are drawn merely for better identification.

II.3 Digital Convolution Using the Discrete Fourier Transform

Highly efficient algorithms have become available for computing the discrete Fourier transform (DFT) of a finite-duration sequence [3]. For this reason, it is computationally efficient to consider implementing a convolution of two relatively short sequences by calculating their DFT's, multiplication in the transform domain, and computing the inverse DFT of the result. Term-by-term multiplication of two DFT's corresponds to a circular convolution of the sequences involved. The overlap-add method described in Subsection II.2 can be used to ensure that successive circular convolutions of length $N = 2M$ have the effect of a linear convolution as defined by (1) for filtering purposes.

Since a DFT is inherently complex-valued no special provisions have to be made for handling complex signals. It will again be assumed that the impulse response $\{h_n\}$ of the filter is fixed so that its N-point DFT can be precomputed and stored. In order to conform with the numbers available for DFT computations, all complex-valued operations have been translated into multiplications and additions of real-valued numbers using the correspondences

$$\begin{aligned} &1 \text{ complex multiplication} = 4 \text{ real multiplies} + 2 \text{ real adds} \\ &1 \text{ complex addition} \quad\quad\; = 2 \text{ real additions.} \end{aligned} \tag{8}$$

Then complex linear convolution in the frequency domain takes, per block of M output samples, the number of operations for two Fourier transforms of length $N = 2M$, $4N$ multiplications and $2N$ additions in the transform domain, and $2(N-M) = 2M$ additions for overlap-add. This yields a total of

2 times the operations of a 2M-point
(complex) DFT per M output samples
8 multiplications/output sample
6 additions/output sample. (9)

The results are plotted in Figs. 2 and 3 as a function of the filter length M using the operation counts of the Winograd Fourier Transform (WFT) algorithms given in Tables 2 and 3 of [3]. Results for filter lengths where efficient Winograd algorithms exist are interconnected by straight lines merely for improved clarity of the graphical representation.

II.4 Digital Convolution Using Number-Theoretic Transforms

Number-theoretic transforms are being proposed increasingly to implement digital filters as a means of eliminating round-off errors and of improving the computational efficiency [4]. A large class of multiplication-free transforms exists which have the circular convolution property and which can be applied to filtering, e.g., by using the overlap-add technique described earlier. However, in contrast to the algorithms considered so far, hardware implementation of multiplication-free algorithms differs considerably from conventional signal processors. Additions have to be performed modulo a particular number and the processor generally needs an increased word length for handling longer transforms.

In signal-processing applications like receivers of data-processing systems, conventional arithmetic operations will still be required for functions other than filtering. This demands a processor architecture supporting both arithmetic modes. Because of the differing hardware complexity we believe that a comparison with other algorithms on the basis of multiplications and additions alone is not meaningful. We, therefore, exclude number-theoretic transform methods from further consideration and make reference instead to recent publications [15], [16] , which evaluate the computational complexity and overall accuracy of these methods for filtering applications.

II.5 Comparison

The results in Fig. 2 reveal a considerably smaller number of multiplications if complex-valued digital filters are implemented in a transform domain instead of in the time domain. Beginning at approximately six taps the computational savings increase with the filter length. Reductions in the number of multiplications come about predominantly since blockwise processing of signals can be performed more efficiently than the sample-by-sample calculations typical for time-domain processing. A drawback inherent in all transform methods is their use of additional storage for blocking of the data. Comparing the two specific transform methods considered here, Agarwal & Cooley slightly outperforms Winograd for filters having less than 20 taps. For longer filters Winograd's algorithms require consistently fewer multiplications, and the numbers plotted could be reduced even further by introducing Golub's scheme instead of (8).

The number of additions required by the various processing schemes for fixed-coefficient complex filtering represents a less homogeneous result. The comparison in Fig. 3 indicates that Agarwal & Cooley's algorithms are wasteful regarding the number of additions; the amount of computation is generally larger than for time-domain processing. The same finding holds for short filters implemented with Winograd's algorithms. However, at about $M = 20$ a crossover occurs and for longer filters the Winograd method needs fewer additions than all alternative solutions considered.

Equalizers employed in telephone modems are transversal filters with typically 16 to 40 taps depending on the bandwidth required for data transmission. Combining the results of Figs. 2 and 3, frequency-domain processing using Winograd's algorithms turns out to be the most efficient approach for computing linear convolutions within this range of tap gains. The processing approach will, therefore, be adopted and extended in the following to allow adaptive filtering of complex signals.

III. ADAPTIVE EQUALIZATION IN THE DISCRETE FREQUENCY DOMAIN

Fig. 4 shows the time-domain implementation of an adaptive equalizer as it is commonly employed in data-transmission systems [6]. Adaptive equalization consists of two distinct processes: (i) Filtering of the sampled input signal x_n using a transversal filter with M coefficients c_m $(0 \leq m \leq M-1)$ spaced T seconds apart,

$$y_n = \sum_{m=0}^{M-1} x_m \, c_{n-m} \, , \tag{10}$$

and (ii) updating of the equalizer coefficients at every sampling time nT of the signal using the recursive algorithm [17]

$$c_m^{(n+1)} = c_m^{(n)} - \alpha e_n x_{n-m}^{*} \quad (0 \leq m \leq M-1), \tag{11}$$

which minimizes the mean-square error

$$|e_n|^2 = |y_n - d_n|^2 \tag{12}$$

at the decision circuit. It has been mentioned earlier that in modem applications the signal samples x_n, y_n as well as the tap gains c_m usually have complex values; $*$ denotes complex conjugation and $\alpha > 0$ is a real constant. Using the insight gained in Section II it will be described how the discrete time-domain operations Eqs. (10) and (11) can be performed more efficiently by frequency-domain processing.

III.1 Linear Convolution Using the Discrete Frequency Domain

Convolution (10) of the sequences $\{x_n\}$ and $\{c_m\}$ is implemented by computing their discrete Fourier transforms (DFT's), element-by-element multiplication, and finally by computing the inverse DFT (see Subsection II.3). For linear (non-circular) convolution of an L-point segment of $\{x_n\}$ with the M-point sequence $\{c_m\}$ the transform length must be $N \geq L + M-1$. The output sequence $\{y_n\}$ is constructed using the overlap-add method. Thus if one defines the N-point DFT's

$$C(k) = \sum_{i=0}^{N-1} c'(i)\, W_N^{ik}, \quad \text{where} \quad c'(i) = \begin{cases} c_i & \text{for } 0 \leq i \leq M-1 \\ 0 & \text{for } M \leq i \leq N-1 \end{cases} \tag{13}$$

and

$$X(k) = \sum_{i=0}^{N-1} x'(i)\, W_N^{ik}, \quad \text{where} \quad x'(i) = \begin{cases} x_i & \text{for} \quad 0 \leqq i \leqq L-1 \\ 0 & \text{for} \quad L \leqq i \leqq N-1, \end{cases} \tag{14}$$

with

$$W_N^{ik} = e^{-j(2\pi/N)ik}, \tag{15}$$

then

$$\tilde{y}(i) = \frac{1}{N} \sum_{k=0}^{N-1} [C(k) \cdot X(k)]\, W_N^{-ik} \quad (0 \leqq i \leqq N-1) \tag{16}$$

is a segment from which the filtered output sequence $\{y_n\}$ is obtained by overlap-add operations.

III.2 Automatic Adjustment of the Transfer Function

The transfer function (13) of an adaptive equalizer is time varying. Translating the minimum mean-square error criterion (12) for optimal filter adjustment into the frequency domain means finding that transfer characteristic $C(k)$, which minimizes

$$\varepsilon = \sum_{k=0}^{N-1} |E(k)|^2 = \sum_{k=0}^{N-1} |C(k) \cdot X(k) - D(k)|^2, \tag{17}$$

where

$$D(k) = \sum_{i=0}^{N-1} d'(i)\, W_N^{ik}, \quad \text{with} \quad d'(i) = \begin{cases} d_i & \text{for} \quad 0 \leqq i \leqq M-1 \\ 0 & \text{for} \quad M \leqq i \leqq N-1 \end{cases} \tag{18}$$

is the N-point DFT of the desired output sequence. Minimization of (17) is subject to the constraint

$$\sum_{k=0}^{N-1} C(k)\, W_N^{-ik} = 0 \quad \text{for} \quad M \leq i \leq N-1 \,, \tag{19}$$

since $C(k)$ must be the DFT of a realizable M-tap filter [see (13)], viz., the one used in (10). Restriction (19) generally rules out the quotient $C(k) = D(k)/X(k)$ as a direct solution of (17); for an exceptional case see Subsection V.1. It can be shown that updating algorithm (11) for solving (17) translates into

$$C(k)^{(b+1)} = C(k)^{(b)} - \beta_W P\{E(k) \cdot X(k)^*\} \quad (0 \leq k \leq N-1) \,, \tag{20}$$

where $P\{\cdot\}$ is a projection operator [18], [11] which enforces (19) on the unconstrained gradient $E(k) \cdot X(k)^*$, and where $\beta > 0$ is a real-valued gain factor. Equation (20) is computed for every new block b of M input samples $(L = M)$. For complex signals P becomes a complex $N \times N$ matrix and multiplication in (20) represents a sizeable processing load.

Alternatively, constraint equation (19) can be satisfied by interpreting the projection operator $P\{\cdot\}$ as performing the inverse DFT of the gradient, setting to zero the last $N-M$ points, and taking the DFT of the result. A schematic implementation of (16) and (20) employing this scheme is shown in Fig. 5. If $C(k)^{(0)}$ is an initial equalizer setting conforming to (19), [e.g., $C(k) = 0$ for all k], algorithm (20) will always yield a feasible equalizer characteristic. The important advantage of this latter projection method is that it operates in terms of DFT's, where computationally efficient algorithms are available [3]. It can be shown that the multi-DFT projection operator is mathematically identical to Rosen's projection matrix. This finding is in contrast to the original paper [10], where the DFT scheme has already been mentioned as an "alternate gradient projection" method and was found to be inferior in performance.

IV. COMPUTATIONAL EFFICIENCY OF ADAPTIVE EQUALIZATION

The rationale behind a frequency-domain approach to adaptive equalization is potential saving in the number of computations compared to a time-domain implementation. Such savings have been identified in Section II for fixed-coefficient filtering. Using again as a measure for computational efficiency the numbers of real multiplications and real additions required to compute the processing functions specified in Section III, time-domain and frequency-domain processing schemes for adaptive equalization will be evaluated and compared in the sequel. In order to ensure a uniform representation, relationship (8) is applied for translating complex-valued operations into multiplications and additions of real numbers.

IV.1 Time-Domain Processing

Equalization (10) of complex input samples x_n using M complex filter coefficients c_m requires, according to (3), $4M$ multiplications and $4M-2$ additions per output sample y_n. Updating (11) of the coefficients takes another $4M+2$ multiplications and $4M$ additions for every output sample, since the product αe_n must be evaluated only once per update. Summing these contributions yields a total of

$$\begin{array}{ll} 8M+2 & \text{multiplications/output sample} \\ 8M-2 & \text{additions/output sample.} \end{array} \qquad (21)$$

IV.2 Frequency-Domain Processing

Convolution (16) amounts to the computational load of two DFT's of N complex points each, $4N$ multiplications, and $2N$ additions per block of M output samples (see Subsection II.3). Updating (20) of the equalizer characteristic is accomplished by performing three DFT's of N complex points each, $6N$ multiplies, and $4N$ adds for every block of M output samples. The overlap-add operation for reconstructing the output requires $2(N-M)$ additions per processed block. Since the processing block length must be $L = M$ to ensure complete updating (20) of an M-tap filter, the transform size of the DFT's for performing linear convolution has to be $N \geq 2M-1$. Assuming for simplicity a choice of $N = 2M$ the operations count sums up to

5 times the operations of a 2M-point
(complex) DFT per M output samples
20 multiplications/output sample
14 additions/output sample. (22)

IV.3 Comparison

In Fig. 6 multiplications and additions per output sample according to Eqs. (21) and (22) are plotted as a function of the number of complex filter coefficients for a range typically used in modem equalizers. For evaluating (22), operation counts of the Winograd Fourier Transform (WFT) algorithms have been extracted from [3]. A peculiarity of all fast Fourier transforms is that efficient algorithms exist only for distinct numbers of transform points which are highly composite. This is indicated in Fig. 6 by not connecting the individual results of (22) to a curve.

While the number of multiplications required for time-domain processing increases linearly with the number of filter coefficients, the frequency-domain approach requires less than 50 multiplies per output sample throughout the range of interest. When using very short equalizers these savings of multiplications for frequency-domain processing are accompanied by a slightly increased number of additions. At 20 coefficients, an equalizer length which seems appropriate for 4800 bps modems, a crossover occurs; here the frequency-domain version saves 73% of the multiplications and some 7% of the additions required in a time-domain implementation. For longer equalizers these savings in operations become more significant. Considering a 40-tap frequency-domain equalizer which might find application in a 9600 bps modem, only 15% of the multiplications and 57.5% of the additions of the time-domain version need be computed.

V. EQUALIZER INITIALIZATION AND PERFORMANCE MEASUREMENTS

Updating algorithm (20) operates on blocks of M input samples; thus convergence of the equalizer is inherently slower than for the time-domain algorithm (11), which updates the equalizer coefficients every sample time. However, frequency-domain processing is ideally suited for applying a fast initialization scheme called "cyclic equalization" [19], which provides a good starting point of the transfer function, $C(k)^{(0)}$, for subsequent adaptations.

V.1 Cyclic Equalization

If a known training sequence of $L = N$ samples is repeatedly transmitted at the onset of adaptive equalization, the processing mechanism described in Section III degenerates to circular operation. In this case constraint equation (19) becomes meaningless and (17) can be solved direct by computing

$$C(k) = D(k)/X(k) \quad (0 \leq k \leq N-1), \tag{23}$$

where $X(k)$ is the DFT of the received N-point start-up sequence and $D(k)$ is the desired spectrum stored at the receiver. The training sequence should have a uniform energy distribution over the entire transmission band; pseudo-random sequences meet this requirement.

The resulting (circular) transfer function $C(k)$ corresponds to an $N = 2M$ tap transversal filter. This means that identification of the unknown transmission channel is performed at a closer frequency spacing than in the time-domain method proposed in [19]. One can now choose those M out of N coefficients which lead to an optimal equalizer setting. This is achieved by transforming $C(k)$ back into the time domain and selecting that subsequence of length M which contains the largest absolute tap-gain values. The procedure described yields a superior initial equalizer setting than the time-domain approach [19] where one is restricted to M coefficients which must be rotated to have the largest tap value at a center position.

In realistic systems the input samples x_n are corrupted by additive noise. Its influence on the start-up performance can be reduced by evaluating (23) for several consecutive blocks of the training sequence and averaging the resultant estimates of the transfer function (see Subsection V.2).

V.2 Computer Simulations

Feasibility and performance of adaptive equalization based on frequency-domain processing have been checked by simulations on a general-purpose computer using floating-point arithmetic. An 8-PSK (Phase-Shift-Keying) system was simulated with parameters closely resembling the standardized 4800 bps modem specifications. Performance is measured for a worst-case transmission channel characterized by the heavy amplitude and phase distortions produced when connecting two realistic C 1-type telephone lines in tandem; Fig. 7 shows attenuation and group-delay characteristics of this channel. The noise spectrum of the transmission channel is assumed to be flat within the bandwidth of the data signal; synchronization and demodulation at the receiver are perfect.

Based upon an equalizer length of $M = 16$ taps, each processed block in (16) and (20) contains 16 data symbols. At a signaling frequency of 1600 Hz this is equivalent to a signal duration of 10 msec. Since $N = 2M = 32$ is not a Winograd number [3], Fourier transforms of length $N = 36$ are implemented. Thus cyclic start-up uses a 36-point pseudo-random training sequence known at the receiver. Simulations reveal that at a signal-to-noise ratio SNR = 30 dB two segments of periodic data are sufficient to yield an accurate initial estimate of the equalizer characteristic. This estimate is subsequently improved by applying updating algorithm (20) during normal data transmission, i.e., while transmitting a random (unknown) data sequence.

Fig. 8 shows the transient behavior of a 16-tap frequency-domain equalizer when adapting to the telephone channel (see Fig. 7) with SNR = 30 dB. Mean-square error MSE(b) is averaged over five independent equalization runs. The solid curve simulating fast start-up exhibits almost instantaneous convergence. The dotted curve results for initial equalization from $C(k)^{(0)} = 0$ and assumes that no decision errors occur, i.e., the random-data sequence must be known at the receiver throughout the convergence process. Adaptation of the corresponding time-domain equalizer from all-zero tap gains using algorithm (11) is shown as a dashed curve in Fig. 8. Rapid convergence is achieved by controlling the gain α in two steps [8] rather than by performing cyclic equalization which would afford a departure from the normal time-domain processing mode [19]. Comparison with the solid curve in Fig. 8 demonstrates the performance potential of the frequency-domain approach in combination with fast initial start-up.

The simulations in Fig. 8 are based on floating-point (single) precision of all arithmetic operations including the Fourier transforms. To predict also the performance of a more realistic digital implementation, quantitative analyses of a 16-bit fixed-point program of the Winograd Fourier Transform (WFT) for 40 complex points have been made. With straightforward scaling a signal-to-truncation error ratio of 50 dB was observed when computing a WFT and its inverse in tandem. This arithmetic inaccuracy is well below the signal-to-noise ratio of actual telephone channels and should not degrade performance of adaptive equalization in any significant manner.

VI. SUMMARY AND CONCLUSIONS

The advent of very efficient algorithms for signal processing in the transform domain has made it timely to apply them to equalization in voice-band modems and to assess their potential for reducing the amount of computation in digital modem implementations. In this paper an adaptive equalizer for complex signals has been described where filtering and coefficient adjustment are performed in the frequency domain employing efficient Fourier transform techniques. The approach utilizes an iterative mean-square error algorithm for parameter optimization which resembles the conventional time-domain algorithm in the frequency domain but requires gradient projection for achieving feasible equalizer characteristics. Convergence of the equalizer for random data is slower than in comparable time-domain implementations due to blockwise processing of the signal samples. This drawback is overcome by a fast initialization algorithm based on cyclic equalization, which can be easily integrated in the normal operating mode of the frequency-domain approach.

Simulations of a modem receiver for data transmission at 4800 bps over realistic telephone channels show that mean-square error performance is equivalent, whether adaptive equalization is performed using the proposed frequency-domain technique with fast start-up or by conventional time-domain processing. However, processing load expressed in multiplications and additions per output sample is significantly reduced when operating in the frequency domain and using Winograd's efficient algorithms for computing the discrete Fourier transformation (see Fig. 6). Reductions in the number of multiplications range between a factor of 3.6 for a 20-tap equalizer (for possible application in medium-speed modems) and 6.8 for a 40-tap equalizer (which might be employed in high-speed modems); the reductions of add operations are between 7% and 42.5%, respectively. This gives frequency-domain processing a great potential of implementing a high-speed receiver with processor hardware which was originally designed for medium-speed time-domain processing.

It remains an intricate question, what fraction of the computational savings expressed as multiplications and additions in Figs. 2, 3 and 6 does in fact materialize as reduced processing load on a digital signal processor, where number of machine cycles executed is the measure of interest. Block-by-block processing of signal samples implied in transform-domain techniques not only increases the requirements for random-access storage but also demands a rather complex control structure, which regulates the flow of data, and monitors addressing of the various buffer areas. Our preliminary estimates for implementing complex-valued convolutions with a fixed 20-tap filter on a general-purpose signal processor [20] showed a factor of two reduction in the number of machine cycles in favor of the frequency-domain approach proposed. This 50% saving must be related to 78% fewer multiplications and 24% fewer additions per output sample identified in Figs. 2 and 3, respectively. Additional research is required to better understand how a reduction in the number of arithmetic operations maps into an improvement in throughput for a signal processor of given architecture.

Consequences to the overall numerical accuracy when performing signal processing in a transform domain instead of by direct computation under the fixed-point arithmetic constraint of many signal processors, is another issue deserving further research [16]. Integer-number processing as implied in [4] and more generally in number-theoretic transforms may very well turn out to be of ultimate significance for selecting the most economic approach.

REFERENCES

[1] A.C. Salazar *et al.*, "Implementation of voiceband modems on digital signal processor," *IEEE National Telecommunications Conf. Record* 75 CH 1015-7 CSCB, Dec. 1-3, 1975, New Orleans, pp. 23-13 to 23-16.

[2] P.J. van Gerwen *et al.*, "Microprocessor implementation of high-speed data modems," *IEEE Trans. Commun. Technol.*, vol. COM-25, no. 2, pp. 238-250, Feb. 1977.

[3] S. Winograd, "On computing the discrete Fourier transform," IBM Res. Lab. Rep. RC 6291, Dec. 30, 1976.

[4] R.C. Agarwal and J.W. Cooley, "New algorithms for digital convolution," *IEEE Trans. Ac ust., Speech, Signal Processing,* vol. ASSP-25, no. 5, pp. 392-410, Oct. 1977.

[5] R.C. Agarwal and C.S. Burrus, "Number theoretic transforms to implement fast digital convolution," *Proc. IEEE,* vol. 63, no. 4, pp. 550-560, April 1975.

[6] R.W. Lucky, J. Salz, and E.J. Weldon, Jr., *Principles of Data Communication.* New York: McGraw-Hill, 1968.

[7] A. Gersho, "Adaptive equalization of highly dispersive channels for data transmission," *Bell Syst. Tech. J.,* vol. 48, no. 1, pp. 55-77, Jan. 1969.

[8] G. Ungerboeck, "Theory on the speed of convergence in adaptive equalizers for digital communication," *IBM J. Res. Develop.,* vol. 16, no. 6, pp. 546-555, Nov. 1972.

[9] T.G. Stockham, "High speed convolution and correlation," in *1966 Spring Joint Comput. Conf., AFIPS Proc.,* vol. 28, pp. 229-233, 1966.

[10] T. Walzman, "Automatic equalizaton and discrete Fourier transform techniques", Ph. D. dissertation, Polytech. Inst. Brooklyn, Brooklyn, N.Y., 1971.

[11] **T. Walzman and M. Schwartz, "Automatic equalization using the discrete** frequency domain," *IEEE Trans. Inform. Theory,* vol. IT-19, no. 1, pp. 59-68, Jan. 1973.

[12] H.B. Voelcker, "Toward a unified theory of modulation", *Proc. IEEE,* vol. 54, no. 3, (Part I), pp. 340-353, March 1966 and no. 5, (Part II), pp. 735-755, May 1966.

[13] R.C. Singleton, "An algorithm for computing the mixed radix fast Fourier transform," *IEEE Trans. Audio Electroacoust.*, vol. AU-17, no. 3, pp. 93-103, June 1969.

[14] A.V. Oppenheim and R.W. Schafer, *Digital Signal Processing*. Englewood Cliffs, N.J.: Prentice Hall, 1975.

[15] H.J. Nussbaumer, "Relative evaluation of various number theoretic transforms for digital filtering applications," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-26, no. 1, pp. 88-93, Feb. 1978.

[16] P.R. Chevillat, "Transform-domain filtering with number theoretic transforms," *IEEE Trans. Acoust. Speech, Signal Processing*, vol. ASSP-26, no. 4, pp. , August 1978.

[17] H. Kobayashi, "Simultaneous adaptive estimation and decision algorithm for carrier-modulated data-transmission systems," *IEEE Trans. Commun. Technol.*, vol. COM-19, no. 3, pp. 268-280, June 1971.

[18] J.B. Rosen, "The gradient projection method for nonlinear programming," *SIAM J. Appl. Math.*, vol. 8, no. 1, pp. 181-218, March 1960.

[19] K.H. Mueller and D.A. Spaulding, "Cyclic equalization -- a new rapidly converging equalization technique for synchronous data communication," *Bell Syst. Tech. J.*, vol. 54, no. 2, pp. 369-406, Feb. 1975.

[20] A. Peled, "On the hardware implementation of digital signal processors," *IEEE Trans. Acoust. Speech, Signal Processing*, vol. ASSP-24, no. 1, pp. 76-86, Feb. 1976.

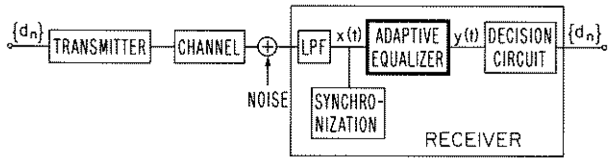

Fig. 1. Block diagram of data-transmission systems.

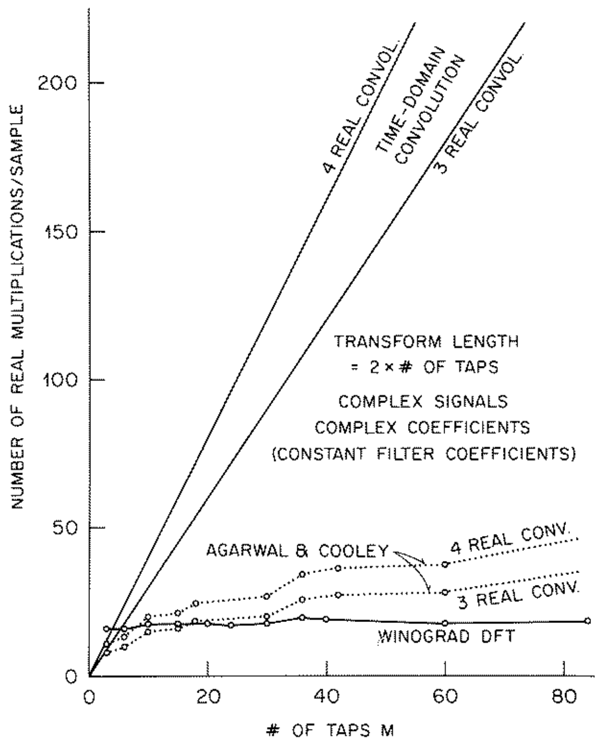

Fig. 2. Number of multiplications per output sample required for fixed-coefficient complex-valued filtering using various convolution algorithms.

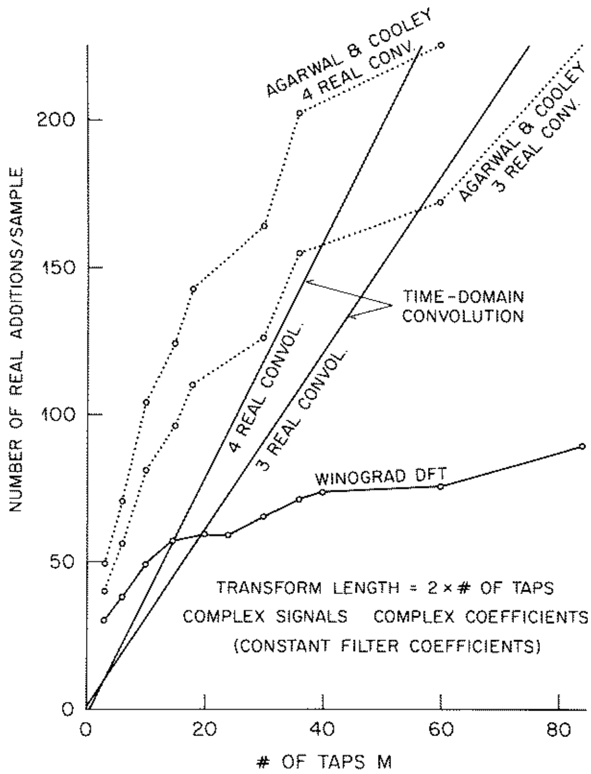

Fig. 3. Number of additions per output sample required for fixed-coefficient complex-valued filtering using various convolution algorithms.

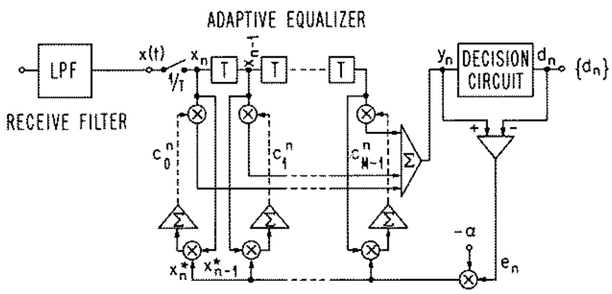

Fig. 4. Time-domain realization of an adaptive transversal equalizer.

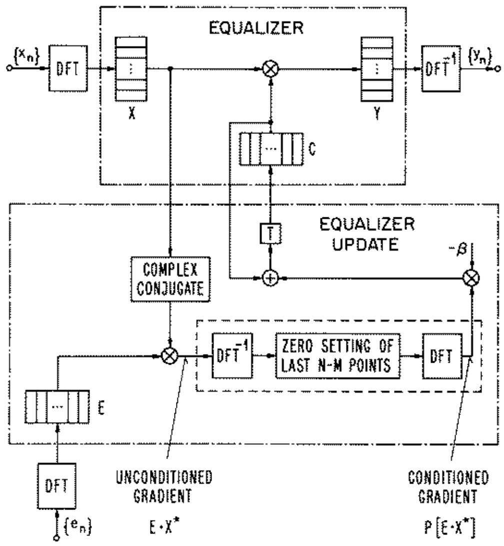

Fig. 5. Frequency-domain realization of an adaptive transversal equalizer.

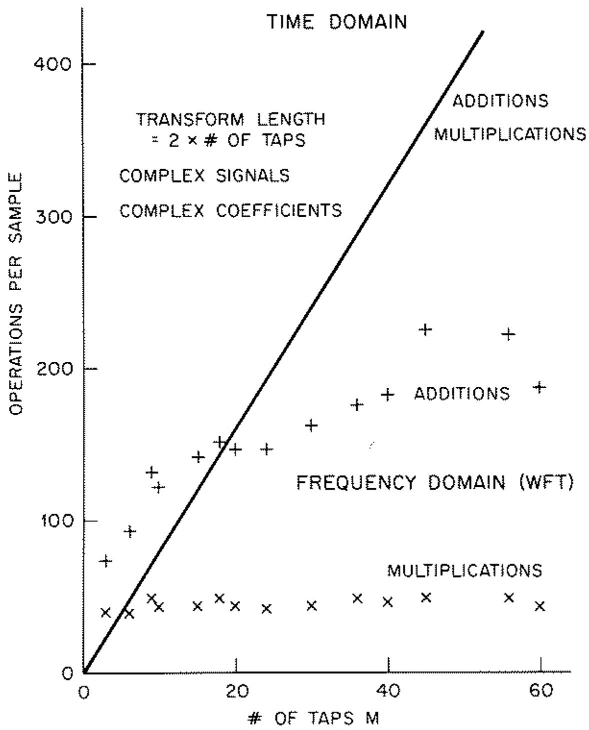

Fig. 6. Comparison of computational loads for adaptive equalization in the time domain and in the frequency domain.

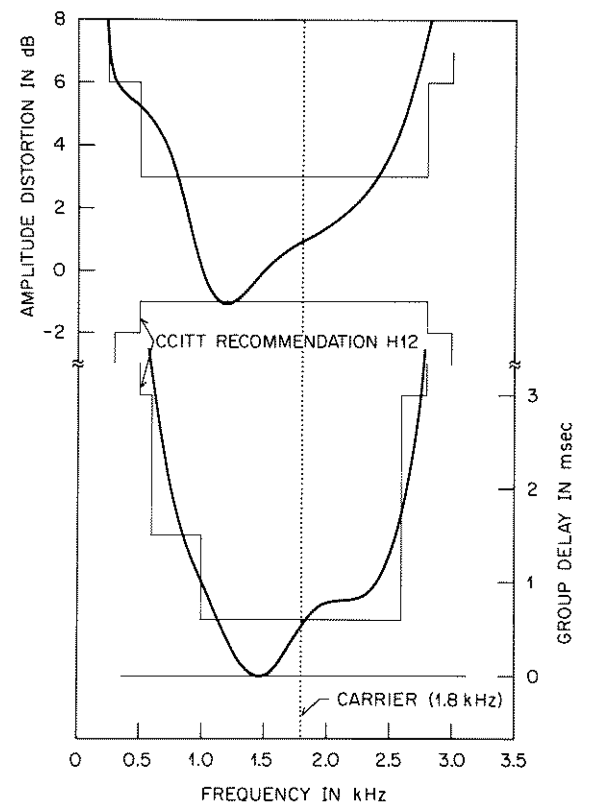

Fig. 7. Channel characteristics of two C-1 telephone lines connected in tandem.

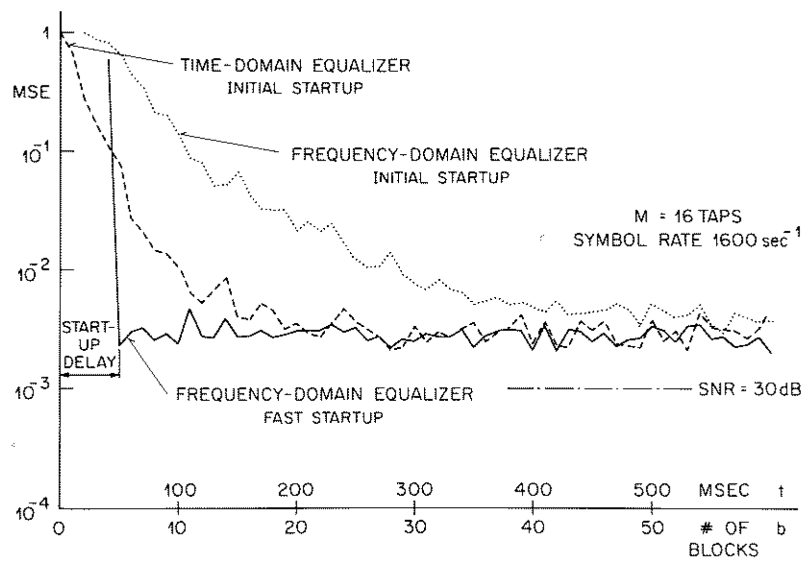

Fig. 8. Convergence properties of a 16-tap adaptive equalizer for various implementations (4800 bps modem, C 1-tandem telephone line).