Fundamental Relations Between the LMS Algorithm and the DFT

BERNARD WIDROW, FELLOW, IEEE, PHILIPPE BAUDRENGHIEN, MEMBER, IEEE,
MARTIN VETTERLI, MEMBER, IEEE, AND PAUL F. TITCHENER, MEMBER, IEEE

Abstract —The digital Fourier transform (DFT) and the adaptive least mean square (LMS) algorithm have existed for some time. This paper establishes a connection between them. The result is the "LMS spectrum analyzer," a new means for the calculation of the DFT. The method uses a set of N periodic complex phasors whose frequencies are equally spaced from dc to the sampling frequency. The phasors are weighted and then are summed to generate a reconstructed signal. Weights are adapted to realize a best least squares fit between this reconstructed signal and the input signal whose spectrum is to be estimated. The magnitude squares of the weights correspond to the power spectrum.

For a proper choice of adaptation speed, the LMS spectrum analyzer will provide an exact N-sample DFT. New DFT outputs will be available in steady flow after the introduction of each new data sample.

I. Introduction

BOTH THE DIGITAL Fourier transform (DFT) and the LMS adaptive algorithm [1] have been known for a long time, and both of these techniques have enjoyed wide practical application. It is the purpose of this paper to demonstrate relationships between the DFT and the LMS algorithm by showing how the DFT can be calculated by making use of the LMS algorithm. This study leads to a new way to calculate the DFT. The resulting algorithm lends itself to parallel computation and VLSI implementation. Using a parallel implementation, processing speed is essentially independent of the size of the data block N. Computation time per data sample is thus also independent of the number of frequency points in the calculated spectrum.

II. The LMS Spectrum Analyzer

The LMS spectrum analyzer, an adaptive system that can be used in the calculation of the DFT, is shown in Fig. 1. The input signal d_j to be Fourier analyzed is sampled, and the time index is j. The sampling period is T, and the sampling frequency is $\Omega = 2\pi/T$ rad/s. The input d_j could be real or complex. As the system is configured, this input serves as the "desired response" for the adaptive process. The weights $w_0, w_1, \cdots, w_{N-1}$ will in general be complex. The same is true for the weighted sum y_j and for the error ϵ_j. The weights are adjusted or adapted in accord

with the complex LMS algorithm of Widrow, McCool, and Ball [2], [3]:

$$W_{j+1} = W_j + 2\mu\epsilon_j\overline{X}_j. \tag{1}$$

The terms in this equation are defined as follows. W_j is the current complex weight vector. W_{j+1} is the next complex weight vector:

$$W_j \triangleq \begin{bmatrix} w_{0j} \\ w_{1j} \\ \vdots \\ w_{(N-1)j} \end{bmatrix}. \tag{2}$$

The complex error ϵ_j is given by

$$\epsilon_j = d_j - y_j \tag{3}$$

where

$$y_j = X_j^T W_j \tag{4}$$

and where

$$X_j \triangleq \frac{1}{\sqrt{N}} \begin{bmatrix} 1 \\ e^{i\frac{2\pi}{N}j} \\ \vdots \\ e^{i\frac{2\pi(N-1)}{N}j} \end{bmatrix}. \tag{5}$$

Note that $\overline{X}_j$ is the conjugate of X_j and that $i = \sqrt{-1}$. The result of the adaptive process is the weight vector W_j, which will turn out to be associated with the DFT of d_j. The complex LMS algorithm minimizes the mean square of the complex error ϵ_j, i.e., minimizes the mean of the sum of the squares of its real and imaginary parts.

The objective is to cause a weighted sum of harmonic phasors to best match the input d_j, thereby resolving d_j into its Fourier components by means of an adaptive algorithm.

The choice of the phasor components of the X_j vector can be explained in the following way. We use a total of N basis frequencies, including zero frequency. The fundamental basis frequency is Ω/N. The entire set of basis frequencies spans the frequency range from zero to the sampling frequency Ω. The fundamental phasor, a time function expressed in terms of the discrete time index j, is
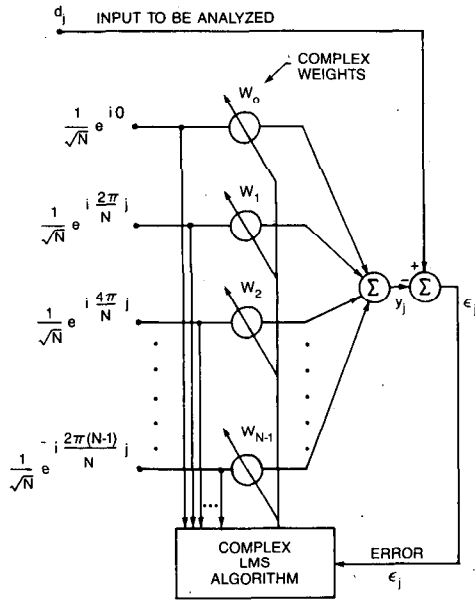
$$e^{i\frac{\Omega}{N}jT}. \tag{6}$$

Fig. 1. LMS Spectrum analyzer.

Since $\Omega T = 2\pi$, this can be written as

$$e^{i\frac{\Omega}{N}jT} = e^{i\frac{2\pi}{N}j}. \tag{7}$$

All of the components of the X vector are powers of (7). A normalization factor $1/\sqrt{N}$ has been included to simplify the analysis of the system of Fig. 1. It causes the X_j vector to have unit power.

The complex LMS algorithm is used to obtain the weights. The weight iteration formula is given by (1). Using (3) and (4), this iteration formula can be expressed as

$$\begin{aligned} W_{j+1} &= W_j + 2\mu\epsilon_j \overline{X}_j \\ &= W_j + 2\mu \overline{X}_j(d_j - y_j) \\ &= W_j + 2\mu \overline{X}_j(d_j - X_j^T W_j) \\ &= W_j + 2\mu d_j \overline{X}_j - 2\mu \overline{X}_j X_j^T W_j \\ &= (I - 2\mu \overline{X}_j X_j^T)W_j + 2\mu d_j \overline{X}_j. \end{aligned} \tag{8}$$

Let the initial weight vector be set to zero. On a step-by-step basis, the weight vector versus time can be induced. Using formula (8),

$$\begin{aligned} W_1 &= (I - 2\mu \overline{X}_0 X_0^T)W_0 + 2\mu d_0 \overline{X}_0 \\ &= (1 - 2\mu \overline{X}_0 X_0^T)0 + 2\mu d_0 \overline{X}_0 \\ &= 2\mu d_0 \overline{X}_0. \end{aligned} \tag{9}$$

Next,

$$\begin{aligned} W_2 &= (I - 2\mu \overline{X}_1 X_1^T)W_1 + 2\mu d_1 \overline{X}_1 \\ &= 2\mu d_0 \overline{X}_0 - 4\mu^2 \overline{X}_1 X_1^T \overline{X}_0 d_0 + 2\mu d_1 \overline{X}_1 \\ &= 2\mu(d_0 \overline{X}_0 + d_1 \overline{X}_1) - 4\mu^2 \overline{X}_1(X_1^T \overline{X}_0)d_0. \end{aligned} \tag{10}$$

The product $(X_1^T \overline{X}_0)$ of (10) can be evaluated as follows:

$$X_1^T \overline{X}_0 = \frac{1}{N}\begin{bmatrix} 1 & e^{i\frac{2\pi}{N}} & e^{i\frac{2\pi(2)}{N}} & \cdots & e^{i\frac{2\pi(N-1)}{N}} \end{bmatrix}\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

$$= \sum_{l=0}^{N-1} e^{i\frac{2\pi l}{N}} = 0. \tag{11}$$

The sum is geometric and can easily be shown to be zero. Basically, we are adding N unit magnitude complex vectors that are equally spaced about the unit circle. As a result of (11), (10) can be simplified to

$$W_2 = 2\mu(d_0 \overline{X}_0 + d_1 \overline{X}_1). \tag{12}$$

Next,

$$\begin{aligned} W_3 &= (I - 2\mu \overline{X}_2 X_2^T)W_2 + 2\mu d_2 \overline{X}_2 \\ &= 2\mu(d_0 \overline{X}_0 + d_1 \overline{X}_1 + d_2 \overline{X}_2) \\ &\quad - 4\mu^2 \overline{X}_2(X_2^T \overline{X}_0 d_0 + X_2^T \overline{X}_1 d_1). \end{aligned} \tag{13}$$

There are two products in (13) that need to be evaluated:

$$X_2^T \overline{X}_0 = \frac{1}{N}\begin{bmatrix} 1 & e^{i\frac{2\pi(2)}{N}} & \cdots & e^{i\frac{2\pi(N-1)(2)}{N}} \end{bmatrix}\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} = 0 \tag{14}$$

$$X_2^T \overline{X}_1 = \frac{1}{N}\begin{bmatrix} 1 & e^{i\frac{2\pi(2)}{N}} & \cdots & e^{i\frac{2\pi(N-1)(2)}{N}} \end{bmatrix}\begin{bmatrix} 1 \\ e^{-i\frac{2\pi}{N}} \\ \vdots \\ e^{-i\frac{2\pi(N-1)}{N}} \end{bmatrix}$$

$$= 0. \tag{15}$$

Therefore,

$$W_3 = 2\mu(d_0 \overline{X}_0 + d_1 \overline{X}_1 + d_2 \overline{X}_2). \tag{16}$$

We can easily generalize this result:

$$W_j = 2\mu \sum_{m=0}^{j-1} d_m \overline{X}_m, \qquad j = 1, \cdots, N. \tag{17}$$

An interesting case is that for $j = N$. From (17) we obtain

$$W_N = 2\mu \sum_{m=0}^{N-1} d_m \overline{X}_m = 2\mu \frac{1}{\sqrt{N}} \sum_{m=0}^{N-1} d_m \begin{bmatrix} 1 \\ e^{-i\frac{2\pi}{N}m} \\ \vdots \\ e^{-i\frac{2\pi(N-1)}{N}m} \end{bmatrix}$$

$$= \frac{2\mu}{\sqrt{N}}\begin{bmatrix} \sum_{m=0}^{N-1} d_m \\ \sum_{m=0}^{N-1} d_m e^{-i\frac{2\pi}{N}m} \\ \vdots \\ \sum_{m=0}^{N-1} d_m e^{-i\frac{2\pi(N-1)}{N}m} \end{bmatrix}. \tag{18}$$

Except for the scale factor, it is clear from the above that the elements of W_N comprise the values of the DFT of d_j over the uniform time window from $j = 0$ to $j = N - 1$.

Formula (17) is based on the orthogonality of X vectors at different times and it applies for $1 \leqslant j \leqslant N$. Beyond this range, we need a new formula since, for example, X_0 is identical to and of course not orthogonal to X_N. Let us calculate W_{N+1}. Once again from (8),

$$W_{N+1} = \left(I - 2\mu \bar{X}_N X_N^T \right) W_N + 2\mu d_N \bar{X}_N$$

$$= \left(I - 2\mu \bar{X}_N X_N^T \right) \left(2\mu \sum_{m=0}^{N-1} d_m \bar{X}_m \right) + 2\mu d_N \bar{X}_N$$

$$= 2\mu \sum_{m=0}^{N} d_m \bar{X}_m - 4\mu^2 \bar{X}_N \left(X_N^T \sum_{m=0}^{N-1} d_m \bar{X}_m \right). \quad (19)$$

A product in (19) needs to be evaluated:

$$\bar{X}_N \left(X_N^T \sum_{m=0}^{N-1} d_m \bar{X}_m \right) = \bar{X}_N \left(X_N^T \left[d_0 \bar{X}_0 + \sum_{m=1}^{N-1} d_m \bar{X}_m \right] \right). \quad (20)$$

Since $X_N = X_0$ because of the periodicity of (14),

$$X_N^T \bar{X}_0 = X_0^T \bar{X}_0 = \frac{1}{N} [1 \quad 1 \quad \cdots \quad 1] \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} = 1. \quad (21)$$

Furthermore,

$$X_N^T \sum_{m=1}^{N-1} d_m \bar{X}_m = X_0^T \sum_{m=1}^{N-1} d_m \bar{X}_m = 0. \quad (22)$$

This results from the orthogonality between X_0 and $X_1, X_2, \cdots, X_{N-1}$. Returning now to (19) and making use of (20), (21), and (22), we can obtain W_{N+1}:

$$W_{N+1} = 2\mu \sum_{m=0}^{N} d_m \bar{X}_m - 4\mu^2 \bar{X}_N d_0$$

$$= 2\mu \sum_{m=1}^{N} d_m \bar{X}_m + 2\mu (1 - 2\mu) \bar{X}_0 d_0. \quad (23)$$

The next weight vector can be obtained similarly. It follows that

$$W_{N+2} = 2\mu \sum_{m=0}^{N+1} d_m \bar{X}_m - 4\mu^2 \bar{X}_{N+1} d_1 - 4\mu^2 \bar{X}_N d_0$$

$$= 2\mu \sum_{m=2}^{N+1} d_m \bar{X}_m + 2\mu (1 - 2\mu) \left(\bar{X}_1 d_1 + \bar{X}_0 d_0 \right). \quad (24)$$

This result can be generalized as

$$W_j = 2\mu \sum_{m=(j-N)}^{j-1} d_m \bar{X}_m + 2\mu (1 - 2\mu) \sum_{m=0}^{j-N-1} d_m \bar{X}_m,$$

$$j = N + 1, \cdots, 2N. \quad (25)$$

Formula (25) applies to W's between W_{N+1} and W_{2N}.

With some further algebraic work along the same lines, a completely general formula for W_j can be derived which would be applicable over all $j \geqslant 1$, assuming the initial condition $W_0 = 0$. The result is

$$W_j = 2\mu \sum_{m=j-N}^{j-1} d_m \bar{X}_m$$

$$+ 2\mu (1 - 2\mu) \sum_{m=j-2N}^{j-N-1} d_m \bar{X}_m$$

$$+ 2\mu (1 - 2\mu)^2 \sum_{m=j-3N}^{j-2N-1} d_m \bar{X}_m$$

$$+ 2\mu (1 - 2\mu)^3 \sum_{m=j-4N}^{j-3N-1} d_m \bar{X}_m$$

$$\vdots \quad (26)$$

In using this formula, it is understood that the allowed ranges of the index m for each of the sums are set by the upper and lower limits unless these limits are negative. The ranges of m must first be $m \geqslant 0$; then they are determined by the sum's limits. Thus, in applying the formula, one sees that for $N \geqslant j > 0$ only the first term in the series (26) exists, all the rest are zero, and the first term agrees as it should with the previously derived formula (17). For $j = N$, this formula gives W_N as in (18). W_N is proportional to the DFT of the first N samples of d_m.

A critical choice of μ is the value $\mu = 1/2$. Making this choice, the series (26) reduces to its first term regardless of the value of j. Let $\mu = 1/2$ and let j be any integer multiple of N. At time lN, the formula for W_j becomes

$$W_{lN} = \sum_{m=lN-N}^{lN-1} d_m \bar{X}_m$$

$$= \frac{1}{\sqrt{N}} \begin{bmatrix} \sum_{m=lN-N}^{lN-1} d_m \\ \sum_{m=lN-N}^{lN-1} d_m e^{-i\frac{2\pi}{N}m} \\ \vdots \\ \sum_{m=lN-N}^{lN-1} d_m e^{-i\frac{2\pi(N-1)}{N}m} \end{bmatrix}. \quad (27)$$

It is evident from this expression that W_{lN} is proportional to the DFT of the input signal d_j.

Thus far, we have established the following. If we set $\mu = 1/2$, the weight vector W_j will be exactly proportional to the DFT of the previous N samples of the input d_j at times j that are integer multiples of N. The DFT will be correct in both magnitude and phase. Furthermore, from (26), the magnitude squares of the weights at any time j can be seen to be exactly proportional to the magnitude squares of the DFT of the last j samples of input data.
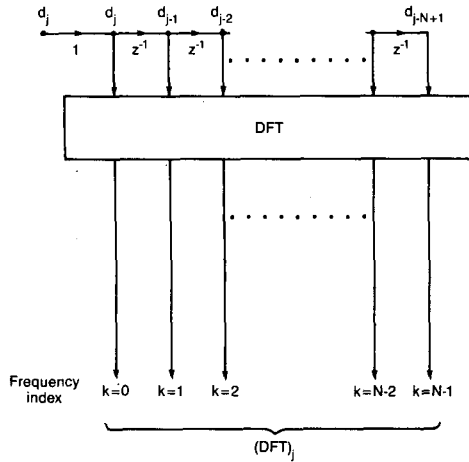
Fig. 2. Scheme for steady-flow calculation of the discrete Fourier transform (DFT).

Thus, the magnitude squares of the weights give the DFT power spectrum of the input d_j.

III. STEADY-FLOW COMPUTATION OF THE DFT BY THE LMS SPECTRUM ANALYZER

Normally, the DFT is calculated with blocks of data N samples long. Sometimes it is of interest to recompute and display the DFT with the arrival of each new input data sample. The result is a "steady-flow" DFT. This steady-flow DFT can be calculated with the system of Fig. 1.

Calculation of the steady-flow DFT is suggested by the block diagram of Fig. 2. The input data are applied to a tapped delay line, and the signals at the taps are applied as inputs to the DFT. The output $(DFT)_j$ is a vector whose components are indexed in terms of frequency k and whose values vary with the time index j. We define the steady-flow DFT vector for data sliding along in time as

$$(DFT)_j \triangleq \frac{1}{N} \begin{bmatrix} \sum_{m=0}^{N-1} d_{j-(N-1)+m} \\ \sum_{m=0}^{N-1} d_{j-(N-1)+m} e^{-i\frac{2\pi}{N}m} \\ \vdots \\ \sum_{m=0}^{N-1} d_{j-(N-1)+m} e^{-i\frac{2\pi k}{N}m} \\ \vdots \\ \sum_{m=0}^{N-1} d_{j-(N-1)+m} e^{-i\frac{2\pi(N-1)}{N}m} \end{bmatrix}. \quad (28)$$

We can relate the steady-flow DFT vector to the adaptive process in the following way. The adaptive weight vector as a function of the time index j is given by the general formula (26). Consider this formula with $\mu = 1/2$.

Then, for all j,

$$W_j = \sum_{m=j-N}^{j-1} d_m \overline{X}_m$$

$$= \frac{1}{\sqrt{N}} \begin{bmatrix} \sum_{m=j-N}^{j-1} d_m \\ \sum_{m=j-N}^{j-1} d_m e^{-i\frac{2\pi}{N}m} \\ \sum_{m=j-N}^{j-1} d_m e^{-i\frac{2\pi k}{N}m} \\ \vdots \\ \sum_{m=j-N}^{j-1} d_m e^{-i\frac{2\pi(N-1)}{N}m} \end{bmatrix}. \quad (29)$$

Although formula (29) is similar to (28), they differ significantly. To make them look more alike, let the index m in (29) be changed to $m + j - N$, and let the new index run from 0 to $N - 1$. The result will be no change in the summations but this will make it easier to connect (29) and (28). Accordingly,

$$W_j = \frac{1}{\sqrt{N}} \begin{bmatrix} \sum_{m=0}^{N-1} d_{m+j-N} \\ \sum_{m=0}^{N-1} d_{m+j-N} e^{-i\frac{2\pi}{N}(m+j-N)} \\ \vdots \\ \sum_{m=0}^{N-1} d_{m+j-N} e^{-i\frac{2\pi k}{N}(m+j-N)} \\ \vdots \\ \sum_{m=0}^{N-1} d_{m+j-N} e^{-i\frac{2\pi(N-1)}{N}(m+j-N)} \end{bmatrix}. \quad (30)$$

The $-N$ terms in the exponents can be deleted since they only account for additive multiples of $2\pi i$ in the arguments of the exponentials. Therefore,

$$W_j = \frac{1}{\sqrt{N}} \begin{bmatrix} \sum_{m=0}^{N-1} d_{m+j-N} \\ e^{-i\frac{2\pi}{N}j} \sum_{m=0}^{N-1} d_{m+j-N} e^{-i\frac{2\pi}{N}m} \\ \vdots \\ e^{-i\frac{2\pi(N-1)}{N}j} \sum_{m=0}^{N-1} d_{m+j-N} e^{-i\frac{2\pi(N-1)}{N}m} \end{bmatrix}. \quad (31)$$

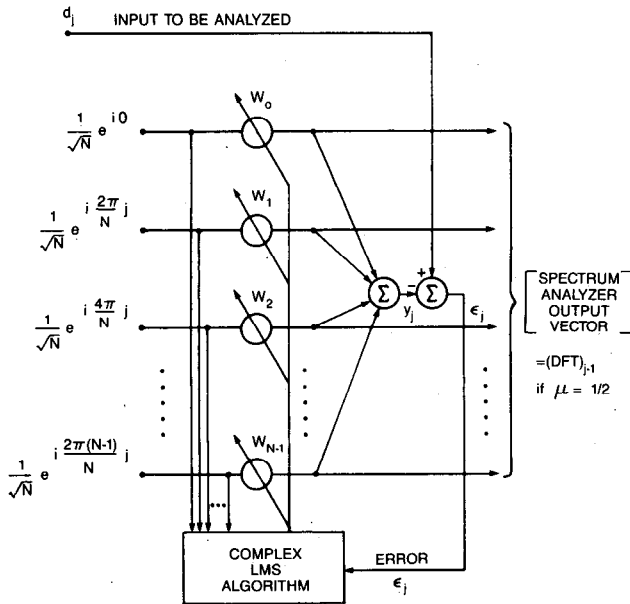Formula (31) is quite similar to (28), yet it is still different.

Fig. 3.   LMS spectrum analyzer for the computation of the steady-flow DFT.

The relationship between them is as follows:

$$
(DFT)_{j-1}
= \frac{1}{\sqrt{N}}
\begin{bmatrix}
1 & & & & \\
& e^{i\frac{2\pi}{N}j} & & & \\
& & e^{i\frac{2\pi(2)}{N}j} & & \\
& & & \ddots & \\
& & & & e^{i\frac{2\pi(N-1)}{N}j}
\end{bmatrix}
W_j. \quad (32)
$$

The weights at time j are multiplied by the corresponding phasor components of the vector X_j (defined by (5)) to give the DFT vector of time $j-1$. Fig. 3 is a block diagram showing how the adaptive system of Fig. 1 can be used to provide the steady-flow DFT.

We had seen in Section II that the components of the weight vector give the DFT at times $j = N, 2N, 3N, \cdots$. Now we see that the components of the weighted X vector give the steady-flow DFT at all times. (The weighted X vector components are equal to the weights themselves at times $j = 0, N, 2N, \cdots$.) Recall that all of these results are based on the initial weights being zero. With $\mu = 1/2$, the adaptive algorithm is stable and "forgets" initial conditions after N samples of data have been inputted. A nonzero initial weight vector can be regarded as having resulted from adapting with some data before time zero. Equation (26) shows that with $\mu = 1/2$, data older than N samples have no effect on W_j.

IV. A Geometric Coherent-Average DFT

When μ is set at some value other than $1/2$, a new form of DFT is calculated by the LMS spectrum analyzer of Fig. 3. The weighted X vector, the set of outputs of the

system of Fig. 3, is in general given by

$$
\begin{pmatrix}
\text{LMS} \\
\text{spectrum} \\
\text{analyzer} \\
\text{output} \\
\text{vector}
\end{pmatrix}_j
= \frac{1}{\sqrt{N}}
\begin{bmatrix}
1 & & & & \\
& e^{i\frac{2\pi}{N}j} & & & \\
& & e^{i\frac{2\pi(2)}{N}j} & & \\
& & & \ddots & \\
& & & & e^{i\frac{2\pi(N-1)}{N}j}
\end{bmatrix}
W_j. \quad (33)
$$

The weight vector W_j is expressed for all time j by formula (35). It is an infinite sum of sums. Thus far, with $\mu = 1/2$, we had only to contend with the first sum. Refer to (32):

$$
\begin{pmatrix}
\text{LMS} \\
\text{spectrum} \\
\text{analyzer} \\
\text{output} \\
\text{vector}
\end{pmatrix}_j
= (DFT)_{j-1}. \quad (34)
$$

Now refer to formula (26), and let μ take a value other than $1/2$. Then,

$$
\begin{pmatrix}
\text{LMS} \\
\text{spectrum} \\
\text{analyzer} \\
\text{output} \\
\text{vector}
\end{pmatrix}_j
= 2\mu(DFT)_{j-1} + \cdots. \quad (35)
$$

The second sum of (26) involves the input data d_j taken over a range of time which is lagged by N samples relative to the corresponding range of time of the first sum. The time range of the third sum is lagged by $2N$ samples relative to the time range of the first sum, and so forth. The data samples in the sums are multiplied by the respective components of $\overline{X}_j$, and in accord with (33), these components are premultiplied by the respective components of X_j, yielding the LMS spectrum analyzer output vector. We note that X_j and $\overline{X}_j$ are periodic with a period of N samples. The first sum of (26) gave us the DFT at time $j-1$. The second sum will give the DFT at time $j-1-N$, and so forth. With proper coefficients for the terms of the series, (35) becomes

$$
\begin{pmatrix}
\text{LMS} \\
\text{spectrum} \\
\text{analyzer} \\
\text{output} \\
\text{vector}
\end{pmatrix}_j
= 2\mu(DFT)_{j-1}
$$
$$
+ 2\mu(1-2\mu)(DFT)_{j-1-N}
$$
$$
+ 2\mu(1-2\mu)^2(DFT)_{j-1-2N}
$$
$$
+ 2\mu(1-2\mu)^3(DFT)_{j-1-3N}
$$
$$
\vdots \qquad (36)
$$

From this formula, one can see that the output of the LMS spectrum analyzer of Fig. 3 is related to the steady-flow DFT. It is a geometrically weighted coherent sum of DFT's, each taken from a data block of N samples. The first DFT is taken from the present and $N-1$ previous input data samples and is weighted by the coefficient 2μ. The second DFT is taken from the Nth previous sample and the $N-1$ samples previous to that. It is weighted by the coefficient $2\mu(1-2\mu)$, and so forth. A geometric ratio can be defined as

$$r \triangleq (1-2\mu). \tag{37}$$

Formula (36) can be written as a geometric sum:

$$\begin{pmatrix} \text{LMS} \\ \text{spectrum} \\ \text{analyzer} \\ \text{output} \\ \text{vector} \end{pmatrix}_j = 2\mu \sum_{l=0}^{\infty} r^l (DFT)_{j-1-lN}. \tag{38}$$

Fig. 4 illustrates the computation of the geometric coherent-average steady-flow DFT as specified by formulas (36) and (38). The data waveform moves to the left with time. The present data sample d_j enters the picture at the "present time" as designated on the time axis. Each sample time, an infinite number of N-point DFT's are taken from the sampled data waveform. The outputs of the DFT's are weighted in proportion to $1, r, r^2, \cdots$. The system output is a geometrically weighted coherent sum of DFT's in accord with (38). To best understand this illustration, think in terms of the data moving while the DFT windows remain stationary. The DFT windows weight uniformally within each DFT data block, and geometrically from block to block.

In order for the adaptive algorithm of Figs. 1 and 3 to be stable or, equivalently, for the DFT computation of Fig. 4 to be stable, it is necessary and sufficient that the magnitude of the geometric ratio be less than 1. Accordingly, for stability,

$$|r| < 1$$

or

$$|1-2\mu| < 1$$

or

$$1 > \mu > 0. \tag{39}$$

Assuming that the stability condition (39) has been met and that the adaptive process has been going on for a very long time, one can visualize from Fig. 4 that the oldest data will be weighted by a very high power of r, which will be a very small number. The effects of very old data will therefore be inconsequential.
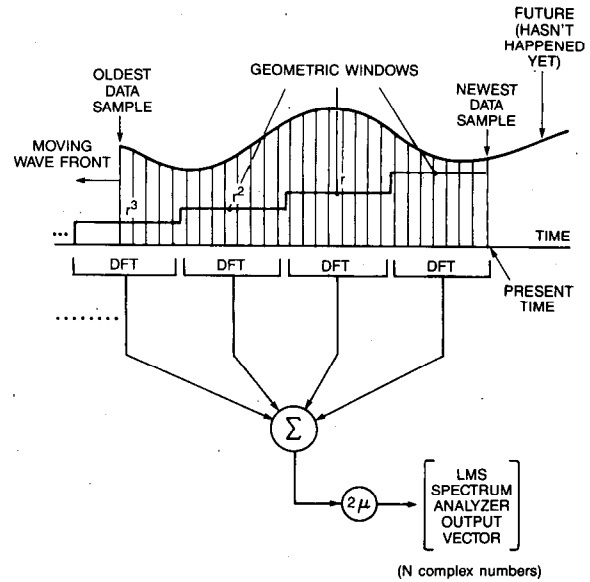

Fig. 4. Computation of a geometric coherent-average steady-flow DFT.

V. CONCLUSIONS

This paper established a connection between the digital Fourier transform (DFT) and the adaptive least mean square (LMS) algorithm. The existence of a close relationship between least squares fitting methods and spectral estimation is not surprising once we recall that the classic Fourier coefficients can be obtained by best least squares fitting of a finite number of sines and cosines to the input signal.

We have shown that the exact N-sample DFT can be computed by means of the adaptive LMS algorithm, with a particular choice of the speed of adaptation ($\mu = 1/2$). More generally, it was shown that the "LMS spectral analyzer" computes a sliding geometric coherent-average DFT. The output is a weighted coherent sum of DFT's taken on adjacent N-sample data blocks. The data are weighted uniformly within each block, and geometrically from block to block.

The DFT calculates only an approximation to the Fourier transform. The LMS algorithm calculates only an approximation to the least squares solution. By making $\mu = 1/2$, the errors in these approximations match and the LMS spectrum analyzer outputs an exact DFT.

REFERENCES

[1] B. Widrow and M. E. Hoff, "Adaptive switching circuits," in *1960 WESCON Conf. Rec.*, pt. 4, pp. 96–140.

[2] B. Widrow, J. M. McCool, and M. Ball, "The complex LMS algorithm," *Proc. IEEE*, vol. 63, pp. 719–720, Apr. 1975.

[3] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1985.

[4] L. R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975.

[5] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975.

[6] S. Haykin, *Introduction to Adaptive Filters*. New York: Macmillan, 1984.

[7] Haykin, *Adaptive Filter Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1986.

[8] C. F. N. Cowan and P. M. Grant, *Adaptive Filters*. Englewood Cliffs, NJ: Prentice-Hall, 1985.

Bernard Widrow (M'58-SM'75-F'76) was born in Norwich, CT, on December 24, 1929. He received the S.B., S.M., and Sc.D. degrees from the Massachusetts Institute of Technology, Cambridge, in 1951, 1953, and 1956, respectively.

From 1951 to 1956, he was a Staff Member of the M.I.T. Lincoln Laboratory and a Research Assistant in the Department of Electrical Engineering of M.I.T. He joined the M.I.T. faculty in 1956 and taught classes in radar, theory of sampled-data systems, and control theory. In 1959, he joined the faculty of Stanford University, Stanford, CA, and is now Professor of Electrical Engineering. He is presently engaged in research and teaching in systems theory, pattern recognition, adaptive filtering, and adaptive control systems. He is an Associate Editor of *Adaptive Control and Signal Processing, Neural Networks, Information Sciences*, and *Pattern Recognition*. He is coauthor with S. D. Stearns of *Adaptive Signal Processing* (Prentice-Hall). In 1986, he received the IEEE Alexander Graham Bell Medal for fundamental contributions to adaptive filtering, adaptive noise canceling, and adaptive antennas.

Professor Widrow is a member of the American Association of University Professors, the Pattern Recognition Society, Sigma Xi, Tau Beta Pi, and is a Fellow of the American Association for the Advancement of Science.

Martin Vetterli (S'86-M'86) was born in Switzerland in 1957. He received the Engineer's degree from the Eidgenösssische Technische Hochschule Zürich, Switzerland, in 1981, the Master of Science degree from Stanford University, Stanford, CA, in 1982, and the Doctorat ès Science degree from the Ecole Polytechnique Fédérale de Lausanne, Switzerland, in 1986.

In 1981, he worked as a Software Engineer for Siemens, Switzerland; in 1982, he was a Research Assistant with the Computer Science Department of Stanford University; and from 1983 to 1986, he was a Researcher at the Ecole Polytechnique. In the Fall of 1984, he was a temporary member of the Technical Staff at AT&T Bell Laboratories in Holmdel, NJ, working on VLSI design. Since the Fall of 1986, he has been an Associate Research Scientist with Columbia University's Center for Telecommunications Research in New York, NY. His research interests include algorithm design for VLSI, computational complexity, multirate signal processing, and packet video transmission.

Dr. Vetterli was the recipient of the 1984 Best Paper Award from the European Signal Processing Society (EURASIP) and of the 1986 Research Prize from the Brown Bovery Corporation (Switzerland).

✠

Philippe Baudrenghien (M'85) was born in Liège, Belgium, on November 20, 1958. He received the "Licence en Sciences Appliquées" from the Université Libre de Bruxelles, Belgium, in 1981, and the M.S. and Ph.D degrees in electrical engineering from Stanford University, in 1982 and 1984, respectively.

Since 1985, he has been with the Radio Frequency Group of the Super Proton Synchrotron Division at CERN (European Center for Nuclear Research) in Geneva, Switzerland, where he is engaged in the design and development of Beam Control and Beam Observation Systems.

Paul F. Titchener (S'79-M'82) was born in Cincinnati, OH, in 1953. He received the B.S. degree (magna cum laude) in electrical engineering from the University of Cincinnati in 1978, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, in 1980 and 1983, respectively.

From 1983 to 1984, he was employed by Hunter Geophysics in Mountain View, CA, and directed the application of digital-signal processing methods to geophysical measurements. Since 1984, he has worked in the research group at Comdisco Resources in San Francisco, CA. His areas of professional interest and activity include adaptive signal processing and signal-processing software systems.

Dr. Titchener is a member of the Audio Engineering Society, the Society of Automotive Engineers, the Hayward Engineering Club, and Tau Beta Pi.