Analysis of the Frequency-Domain Block LMS Algorithm

B. Farhang-Boroujeny, *Senior Member, IEEE*, and Kheong Sann Chan

***Abstract*—In this paper, we present a new analysis of the frequency-domain block least-mean-square (FBLMS) algorithm. An earlier analysis uses a mapping of the frequency-domain information to the time-domain before proceeding with the analysis of the algorithm. We present a direct analysis of the FBLMS algorithm in the frequency domain. As compared with the previous analysis, the new analysis is easier to follow. It is also more rigorous than the previous works and gives a better insight to the effect of various processing components in the algorithm structure on its convergence behavior. In particular, we show how the transformation of input samples to the frequency domain, combined with the effect of the involved windowing matrices, and step-normalization affect the convergence behavior of both constrained and unconstrained versions of the FBLMS algorithm. We also report a procedure for derivation of misadjustment equations of various versions of the FBLMS algorithm.**

***Index Terms*—Adaptive filter, autocorrelation matrix, FBLMS, FFT, frequency domain, LMS, time domain.**

I. Introduction

In THE realization of adaptive filters, the least-mean-square (LMS) algorithm has always been the most popular adaptation scheme. Its conventional form, which was first proposed by Widrow and Hoff [1], has been very well analyzed and understood. Its main drawback is that it does not perform very well when the filter input is highly colored. In the past, researchers have developed many variations of the LMS algorithm, reducing the complexity, increasing the convergence rate, and tailoring it for certain specific applications [14]–[23]. The frequency-domain block LMS algorithm [which is also known as the fast block LMS (FBLMS)] was initially proposed by Ferrara [14], with the purpose of reducing the computational complexity in mind. It takes advantage of the existence of efficient algorithms for the computation of the discrete Fourier transform (DFT) and the fact that point-wise multiplication in the frequency domain is equivalent to convolution in the time domain [25]. In this way, a block of size L outputs of the filter are simultaneously calculated, holding the tap weights constant over the entire block. The tap weights are then updated from block to block instead of at successive samples of input. The block size L is typically (although not necessarily) chosen to be the same as or close to the filter size N.

Manuscript received December 28, 1998; revised March 31, 2000. The associate editor coordinating the review of this paper and approving it for publication was Dr. Sergio Barbarossa.

The authors are with the Department of Electrical Engineering, National University of Singapore, Singapore (e-mail: elefarhg@leonis.nus.edu.sg).

Publisher Item Identifier S 1053-587X(00)05971-7.

Ferrara's algorithm was initially proposed as an exact but fast implementation of the time-domain block LMS (BLMS) algorithm. Mansour and Gray [23] subsequently showed that it was possible to omit an operation that constrained certain time-domain quantities, saving two FFT's, at the cost of some increase in the misadjustment. This algorithm is referred to as the unconstrained FBLMS algorithm and is computationally cheaper than its constrained counterpart. It was also found that since the transformed samples of the filter input (known as frequency bins) are almost uncorrelated from one another, one may use separate normalized step-size parameters for different frequency bins [3], [8], thereby equalizing the convergence rates of all frequency bins. The step-normalization procedure resolves the problem of slow modes of the LMS algorithm and results in an algorithm that converges faster than the conventional LMS and BLMS algorithms [3], [4].

Previous analyses of this algorithm are reported in [4] and [5]. In [5], Sommen *et al.* have analyzed the FBLMS with a generalized window function while assuming that the elements of the frequency bins behave like independent Gaussian stationary random variables. They have derived the stability conditions and expressions for the misadjustments. However, there is no analysis of the eigenvalue spread of the matrices controlling the convergence of the algorithm. The most comprehensive convergence analysis of the FBLMS algorithm has been given by Lee and Un [4] almost a decade ago. They have analyzed the FBLMS algorithm by inverse transforming the frequency-domain equations back to the time-domain and showing that the matrix that controls the convergence behavior of the FBLMS algorithm is asymptotically circulant. Considering the properties of circulant matrices, it is then concluded that the frequency-domain matrix that controls the convergence behavior of the FBLMS algorithm is asymptotically diagonal. The conclusion that can be then drawn is that as the filter length N becomes very large, the eigenvalues of the matrix controlling the convergence behavior of the FBLMS algorithm will approach unity if the step-normalization (noted above) is applied to the various frequency bins.

The analysis provided by Lee and Un [4], although very enlightening from one point of view, is rather difficult to follow. In this paper, we propose an alternative method of analyzing the FBLMS algorithm and arriving at the same conclusions as those of Lee and Un but without resorting to inverse Fourier transforming the frequency-domain equations into the time domain. Our analysis provides a different view of the convergence behavior of the FBLMS algorithm. It is mathematically rigorous and provides more quantifiable results. In particular, by using Gerschgorin's theorem [7], we explain

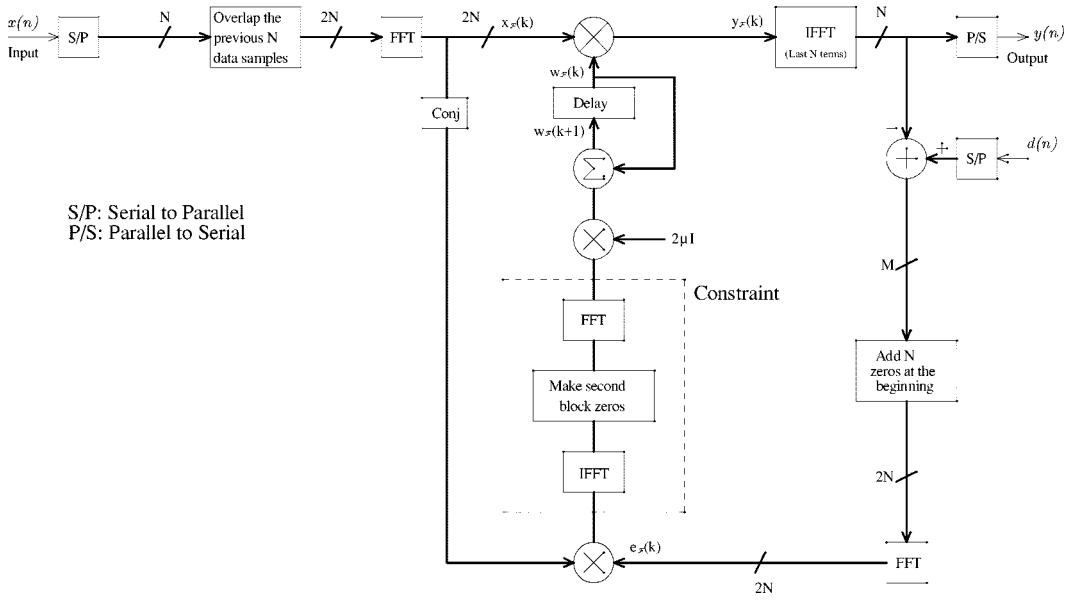

Fig. 1. FBLMS algorithm.

how the various steps in the FBLMS algorithm structure affect its convergence behavior.

This paper is organized as follows. In the next section, we present a summary of the FBLMS algorithm and give a formulation that is most appropriate for the analysis in the subsequent sections. This formulation of the FBLMS algorithm first appeared in [3]. The new analysis of the unconstrained FBLMS algorithm is presented in Section III. The analysis of the constrained FBLMS algorithm follows from the results of its unconstrained version and are presented in Section IV. The misadjustment analysis of the various versions of the FBLMS algorithm is reviewed in Section VI. We outline the general methodology for the derivations and report the final results from [8]. Numerical examples confirming the theoretical results are presented in Section VI. Concluding remarks are presented in the last section of the paper.

We use the following notations throughout the paper. We use nonbold lowercase letters for scalar quantities, bold lowercase for vectors, and bold uppercase for matrices. Nonbold uppercase letters are used for integer quantities such as lengths/dimensions of vectors/matrices. The lowercase letter "n" is used for the time index. The lowercase letter "k" is reserved for the block index. The time and block indexes are put in brackets, whereas subscripts are used to refer to elements of vectors and matrices. The superscripts T and H denote vector or matrix transposition and Hermitian transposition, respectively. We use $\boldsymbol{\mathcal{F}}$ to denote the DFT matrix and the subscript $\mathcal{F}$ to highlight the frequency-domain quantities. We keep all vectors in column form.

II. Summary of the FBLMS Algorithm

Fig. 1 shows a block diagram of the FBLMS algorithm. L new samples of the input are overlapped with $N-1$ of its past samples before they are passed to an N'-point FFT; $N'=N+L-1$. The frequency-domain tap-input and tap-weight vectors are thus

$$\boldsymbol{x}_{\mathcal{F}}(k)=\boldsymbol{\mathcal{F}}\begin{bmatrix} x(kL-N+1)\\ x(kL-N+2)\\ \vdots\\ x(kL-1)\\ x(kL)\\ x(kL+1)\\ \vdots\\ x(kL+L-1)\end{bmatrix}$$

and

$$\boldsymbol{w}_{\mathcal{F}}(k)=\boldsymbol{\mathcal{F}}\begin{bmatrix} w_0(k)\\ w_1(k)\\ \vdots\\ w_{N-1}(k)\\ 0\\ 0\\ \vdots\\ 0\end{bmatrix}\begin{matrix}\left.\begin{matrix}\\ \\ \\ \\ \end{matrix}\right\}N\\ \left.\begin{matrix}\\ \\ \\ \\ \end{matrix}\right\}L-1\end{matrix} \tag{1}$$

respectively, where $\boldsymbol{\mathcal{F}}$ is the $N'\times N'$ DFT matrix. For the purpose of our analysis, we also define the diagonal $N'\times N'$ matrix $\boldsymbol{\mathcal{X}}_{\mathcal{F}}(k)$, which takes the values of $\boldsymbol{x}_{\mathcal{F}}(k)$ down its diagonal

$$\boldsymbol{\mathcal{X}}_{\mathcal{F}}(k)=\text{diag}[\boldsymbol{x}_{\mathcal{F}}(k)]. \tag{2}$$

The $N'\times 1$ frequency-domain output vector is given as

$$\boldsymbol{y}_{\mathcal{F}}(k)=\boldsymbol{\mathcal{X}}_{\mathcal{F}}(k)\,\boldsymbol{w}_{\mathcal{F}}(k)=\boldsymbol{x}_{\mathcal{F}}(k)\otimes\boldsymbol{w}_{\mathcal{F}}(k) \tag{3}$$

where the symbol $\otimes$ denotes point-wise multiplication of the corresponding vectors. The reason for writing two alternative expressions for the frequency-domain output vector $\boldsymbol{y}_{\mathcal{F}}(k)$ is because they come in useful in the analysis and implementation of the FBLMS algorithm, respectively. On the one hand, implementing matrix–vector multiplication is not efficient since it is computationally expensive. On the other hand, formulations

based on matrix–vector multiplications are more convenient for analysis.

The frequency-domain output vector $\boldsymbol{y}_{\mathcal{F}}(k)$ is related to the time-domain samples of output according to [3], [8]

$$\boldsymbol{y}_{\mathcal{F}}(k) = \boldsymbol{\mathcal{F}} \begin{bmatrix} \star \\ \star \\ \vdots \\ \star \\ y(kL) \\ y(kL+1) \\ \vdots \\ y(kL+L-1) \end{bmatrix} \begin{matrix} \left.\begin{matrix} \\ \\ \\ \\ \end{matrix}\right\} N-1 \\ \left.\begin{matrix} \\ \\ \\ \\ \end{matrix}\right\} L \end{matrix} . \tag{4}$$

Since point-wise multiplication in the frequency domain corresponds to circular convolution in the time domain, the time-domain tap-weight vector of the filter is padded with zeros [see (1)] in such a way that the linear portion of the circular convolution (the part that does not wrap around) corresponds to the linear convolution needed for the algorithm. The first $N-1$ elements of $\boldsymbol{\mathcal{F}}^{-1}\boldsymbol{y}_{\mathcal{F}}(k)$ correspond to the circular part of the convolution that wraps around and does not contain any information relevant to the linear convolution; they are thus discarded. However, the last L elements of $\boldsymbol{\mathcal{F}}^{-1}\boldsymbol{y}_{\mathcal{F}}(k)$ are in the right order and contain the correct output samples of the filter for the current block. The frequency domain desired and error vectors are defined as follows [3], [8]:

$$\boldsymbol{d}_{\mathcal{F}}(k) = \boldsymbol{\mathcal{F}} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ d(kL) \\ \vdots \\ d(kL+L-1) \end{bmatrix} \tag{5}$$

and

$$\boldsymbol{e}_{\mathcal{F}}(k) = \boldsymbol{\mathcal{P}}_{O,L}(\boldsymbol{d}_{\mathcal{F}}(k) - \boldsymbol{y}_{\mathcal{F}}(k)) \tag{6}$$

where

$$\boldsymbol{\mathcal{P}}_{O,L} = \boldsymbol{\mathcal{F}} \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_L \end{bmatrix} \boldsymbol{\mathcal{F}}^{-1}. \tag{7}$$

$\boldsymbol{I}_L$ is the $L \times L$ identity matrix, and $\boldsymbol{0}$'s are zero matrices of appropriate dimensions, and the matrix $\boldsymbol{\mathcal{P}}_{O,L}$ has dimensions of $N' \times N'$. We note that premultiplication by $\boldsymbol{\mathcal{P}}_{O,L}$ is equivalent to setting the first $N-1$ elements of the corresponding time-domain vector to zero. This is done by first applying an inverse DFT to the frequency-domain samples, zeroing the time-domain samples of the wrapped around convolution points, and finally applying a DFT to get back to the frequency domain. We thus refer to $\boldsymbol{\mathcal{P}}_{O,L}$ as a windowing matrix. Moreover, we note that the vectors are already arranged in a way to perform the correlations required in the update of the FBLMS algorithm [3], [8]. The update equation for the constrained FBLMS algorithm thus becomes

$$\boldsymbol{w}_{\mathcal{F}}(k+1) = \boldsymbol{\mathcal{P}}_{N,O}(\boldsymbol{w}_{\mathcal{F}}(k) + 2\mu\boldsymbol{\mathcal{X}}^{*}_{\mathcal{F}}(k)\boldsymbol{e}_{\mathcal{F}}(k)) \tag{8}$$

where the asterisk denotes complex conjugate, μ, which is a positive scalar, is the step-size parameter, and $\boldsymbol{\mathcal{P}}_{N,O}$ is the windowing matrix used to constrain the last $L-1$ elements of the time-domain equivalent of $\boldsymbol{w}_{\mathcal{F}}(k)$ to zero. In a similar way to $\boldsymbol{\mathcal{P}}_{O,L}$, $\boldsymbol{\mathcal{P}}_{N,O}$ is defined as

$$\boldsymbol{\mathcal{P}}_{N,O} = \boldsymbol{\mathcal{F}} \begin{bmatrix} \boldsymbol{I}_N & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \boldsymbol{\mathcal{F}}^{-1}. \tag{9}$$

It is worth noting that $\boldsymbol{\mathcal{X}}^{*}_{\mathcal{F}}(k)\boldsymbol{e}_{\mathcal{F}}(k)$ is the stochastic gradient and is obtained by point-wise multiplying the input and error vectors in the frequency domain.

The term "constrained," which is used for the recursion (8), refers to the fact that the last $L-1$ elements of the time-domain equivalent of $\boldsymbol{w}_{\mathcal{F}}(k)$ are set to zero after every update. As was noted earlier, this operation is not necessary for the correct convergence of the FBLMS algorithm. Removing it results in the following update equation, corresponding to the unconstrained FBLMS algorithm:

$$\boldsymbol{w}_{\mathcal{F}}(k+1) = \boldsymbol{w}_{\mathcal{F}}(k) + 2\mu\boldsymbol{\mathcal{X}}^{*}_{\mathcal{F}}(k)\boldsymbol{e}_{\mathcal{F}}(k). \tag{10}$$

Furthermore, it is well known that the convergence behavior of the FBLMS algorithm is greatly improved if we replace the scalar step-size parameter μ by a diagonal matrix $\boldsymbol{\mu}$ whose diagonal elements are the set of normalized step-size parameters

$$\mu_i = \frac{\mu_o}{E[|x_{\mathcal{F},i}(k)|^2]}, \qquad \text{for } i = 0, 1, \cdots, N'-1 \tag{11}$$

where μ_o is a common unnormalized step-size parameter, and $x_{\mathcal{F},i}(k)$ denotes the ith element of $\boldsymbol{x}_{\mathcal{F}}(k)$ (or, equivalently, the ith diagonal element of $\boldsymbol{\mathcal{X}}_{\mathcal{F}}(k)$). In addition, for the convenience of formulations, we define the diagonal matrix $\boldsymbol{\Lambda}$ with the diagonal elements $E[|x_{\mathcal{F},i}(k)|^2]$ for $i = 0, 1, \cdots, N-1$. The step-normalized constrained and unconstrained FBLMS algorithms are then updated according to the equations

$$\boldsymbol{w}_{\mathcal{F}}(k+1) = \boldsymbol{\mathcal{P}}_{N,O}(\boldsymbol{w}_{\mathcal{F}}(k) + 2\mu_o\boldsymbol{\Lambda}^{-1}\boldsymbol{\mathcal{X}}^{*}_{\mathcal{F}}(k)\boldsymbol{e}_{\mathcal{F}}(k)) \tag{12}$$

and

$$\boldsymbol{w}_{\mathcal{F}}(k+1) = \boldsymbol{w}_{\mathcal{F}}(k) + 2\mu_o\boldsymbol{\Lambda}^{-1}\boldsymbol{\mathcal{X}}^{*}_{\mathcal{F}}(k)\boldsymbol{e}_{\mathcal{F}}(k) \tag{13}$$

respectively.

III. Convergence Analysis of the Unconstrained FBLMS Algorithm

Since convergence behavior of the constrained FBLMS algorithm follows easily from the results of its unconstrained counterpart, we begin our discussion with a study of convergence analysis of the unconstrained FBLMS algorithm.

We start with the recursion (13). To rearrange (13) in a proper form for convergence analysis, we recall (6), and note that the frequency-domain desired vector $\boldsymbol{d}_{\mathcal{F}}(k)$ can be extended as $\boldsymbol{d}_{\mathcal{F}}(k) = \boldsymbol{\mathcal{P}}_{O,L}(\boldsymbol{\mathcal{X}}_{\mathcal{F}}(k)\boldsymbol{w}_{\mathcal{F},o} + \boldsymbol{e}_{\mathcal{F},o}(k))$, where $\boldsymbol{w}_{\mathcal{F},o}$ is the DFT of the optimum tap-weight vector, padded with zeros, as was done in (1), and $\boldsymbol{e}_{\mathcal{F},o}(k)$ is the optimum output error vector in the frequency domain that will be achieved when $\boldsymbol{w}_{\mathcal{F}}(k) = \boldsymbol{w}_{\mathcal{F},o}$. In addition, we define the tap-weight error vector $\boldsymbol{v}_{\mathcal{F}}(k) = \boldsymbol{w}_{\mathcal{F}}(k) - \boldsymbol{w}_{\mathcal{F},o}$. Using these definitions in (6)

and substituting the result in (13), after some straightforward manipulations, we obtain

$$\boldsymbol{v}_{\mathcal{F}}(k+1) = (\boldsymbol{I}_{N'} - 2\mu_o \boldsymbol{\Lambda}^{-1} \boldsymbol{\mathcal{X}}^*_{\mathcal{F}}(k) \boldsymbol{\mathcal{P}}_{O,L} \boldsymbol{\mathcal{X}}_{\mathcal{F}}(k)) \boldsymbol{v}_{\mathcal{F}}(k) + 2\mu_o \boldsymbol{\Lambda}^{-1} \boldsymbol{\mathcal{X}}^*_{\mathcal{F}}(k) \boldsymbol{\mathcal{P}}_{O,L} \boldsymbol{e}_{\mathcal{F},o}(k). \tag{14}$$

Taking the expectation on both sides, using the independence assumption [8], which assumes the current tap-input and tap-weight vectors are independent of each other, and noting that according to the principle of orthogonality $E[\boldsymbol{\mathcal{X}}^*_{\mathcal{F}}(k) \boldsymbol{\mathcal{P}}_{O,L} \boldsymbol{e}_{\mathcal{F},o}(k)] = \boldsymbol{0}$, [8], we obtain

$$E[\boldsymbol{v}_{\mathcal{F}}(k+1)] = (\boldsymbol{I}_{N'} - 2\mu_o \boldsymbol{\Lambda}^{-1} \boldsymbol{\mathcal{R}}^u_{xx}) E[\boldsymbol{v}_{\mathcal{F}}(k)] \tag{15}$$

where $E[\cdot]$ denotes statistical expectation, and $\boldsymbol{\mathcal{R}}^u_{xx} = E[\boldsymbol{\mathcal{X}}^*_{\mathcal{F}}(k) \boldsymbol{\mathcal{P}}_{O,L} \boldsymbol{\mathcal{X}}_{\mathcal{F}}(k)]$. From (15), it is obvious that the convergence behavior of the step-normalized unconstrained FBLMS algorithm is controlled by the eigenvalues of the matrix $\boldsymbol{\Lambda}^{-1} \boldsymbol{\mathcal{R}}^u_{xx}$. Lee and Un have shown that the time-domain equivalent $\boldsymbol{\mathcal{F}}^{-1} \boldsymbol{\mathcal{R}}^u_{xx} \boldsymbol{\mathcal{F}}$ of $\boldsymbol{\mathcal{R}}^u_{xx}$ is asymptotically equivalent to a circulant matrix and therefore concluded that $\boldsymbol{\mathcal{R}}^u_{xx}$ is asymptotically equivalent to a diagonal matrix. The use of step-normalized matrix $\boldsymbol{\mu} = \mu_o \boldsymbol{\Lambda}^{-1}$ then equalizes all the diagonal elements of the matrix $\boldsymbol{\Lambda}^{-1} \boldsymbol{\mathcal{R}}^u_{xx}$, resulting in an algorithm with a single mode of convergence.

In this paper, we choose a different approach. We carry out the analysis directly in the frequency domain by showing that as N' grows, the $N' \times N'$ normalized matrix $\boldsymbol{\Lambda}^{-1} \boldsymbol{\mathcal{R}}^u_{xx}$ asymptotically approaches a matrix proportional to the identity matrix $\boldsymbol{I}_{N'}$. To this end, we observe that since $\boldsymbol{\mathcal{X}}_{\mathcal{F}}(k)$ is a diagonal matrix, $\boldsymbol{\mathcal{X}}^*_{\mathcal{F}}(k) \boldsymbol{\mathcal{P}}_{O,L}$ is equivalent to multiplying the rows of $\boldsymbol{\mathcal{P}}_{O,L}$ by the corresponding diagonal elements of $\boldsymbol{\mathcal{X}}^*_{\mathcal{F}}(k)$. Moreover, $(\boldsymbol{\mathcal{X}}^*_{\mathcal{F}}(k) \boldsymbol{\mathcal{P}}_{O,L}) \boldsymbol{\mathcal{X}}_{\mathcal{F}}(k)$ is equivalent to multiplying the columns of $\boldsymbol{\mathcal{X}}^*_{\mathcal{F}}(k) \boldsymbol{\mathcal{P}}_{O,L}$ by the corresponding diagonal elements of $\boldsymbol{\mathcal{X}}_{\mathcal{F}}(k)$. Then, since $\boldsymbol{\mathcal{X}}_{\mathcal{F}}(k) = \text{diag}[\boldsymbol{x}_{\mathcal{F}}(k)]$, we conclude that

$$\boldsymbol{\mathcal{R}}^u_{xx} = E[\boldsymbol{x}^*_{\mathcal{F}}(k) \boldsymbol{x}^T_{\mathcal{F}}(k)] \otimes \boldsymbol{\mathcal{P}}_{O,L} = \boldsymbol{\mathcal{R}}^*_{\mathcal{F}} \otimes \boldsymbol{\mathcal{P}}_{O,L} \tag{16}$$

where $\boldsymbol{\mathcal{R}}_{\mathcal{F}} = E[\boldsymbol{x}_{\mathcal{F}}(k) \boldsymbol{x}^H_{\mathcal{F}}(k)]$, and as before, $\otimes$ denotes point-wise multiplication of the matrices.

From (16), we see that the matrix $\boldsymbol{\mathcal{R}}^u_{xx}$, which controls the convergence of the unconstrained FBLMS algorithm, is made up of point-wise multiplications of the matrices $\boldsymbol{\mathcal{R}}^*_{\mathcal{F}}$ and $\boldsymbol{\mathcal{P}}_{O,L}$. We thus proceed with a thorough study of these matrices followed by a study of their point-wise product. However, before proceeding with this study, we need to make certain assumptions about the input process $x(n)$ to make the problem mathematically tractable.

We assume that $x(n)$ is an autoregressive moving average (ARMA) process that is generated by passing a unit-variance white process $\nu(n)$ through a pole-zero transfer function $H(z)$. An implication of this assumption that is directly related to our analysis in this paper is that the autocorrelation coefficients $\phi_{xx}(l)$ of $x(n)$ may be expanded as a summation of a number of exponentially decaying terms

$$\phi_{xx}(l) = E[x(n)x^*(n-l)] = \sum_{p=0}^{P-1} a_p \rho_p^{|l|} \quad \text{for } l = \cdots, -1, 0, 1, \cdots \tag{17}$$

where the ρ_ps denote the poles of the ARMA process, P denotes its order, and a_ps are a set of coefficients that can be derived from the ARMA model. Moreover, we have assumed that the number of poles of the ARMA process is greater than the number of its zeros. The autocorrelation coefficients $\phi_{xx}(l)$s are directly obtained from the inverse Fourier transform of the power spectral density of $x(n)$, which is given by [6], [8]

$$\Phi_{xx}(e^{\mathbf{j}\omega}) = |H(e^{\mathbf{j}\omega})|^2$$

since $\nu(n)$ is assumed to be a unit-variance white noise process, i.e., $\Phi_{\nu\nu}(z) = 1$.

A. Matrix $\boldsymbol{\mathcal{R}}_{\mathcal{F}} = E[\boldsymbol{x}_{\mathcal{F}}(k) \boldsymbol{x}^H_{\mathcal{F}}(k)]$

For convenience of our discussion, we present our study in terms of the matrix $\boldsymbol{\mathcal{R}}'_{\mathcal{F}} = (1/N') \boldsymbol{\mathcal{R}}_{\mathcal{F}}$. However, we note that since $\boldsymbol{\mathcal{R}}_{\mathcal{F}}$ and $\boldsymbol{\mathcal{R}}'_{\mathcal{F}}$ are related by a scaling factor, they share the same characteristics. In particular, they have exactly the same eigenvalue spread. We recall that for a matrix $\boldsymbol{R}$, the eigenvalue spread is defined as the ratio of its maximum eigenvalue to its minimum eigenvalue. This we denote by $\lambda_{\max}/\lambda_{\min}$. The reason that we choose to work with $\boldsymbol{\mathcal{R}}'_{\mathcal{F}}$ instead of $\boldsymbol{\mathcal{R}}_{\mathcal{F}}$ is that, as will be obvious from the discussions that follow, the diagonal elements of $\boldsymbol{\mathcal{R}}_{\mathcal{F}}$ increase without bound as its dimension N' increases, and this poses some inconvenience in the analysis.

It is shown in Appendix A that the elements of $\boldsymbol{\mathcal{R}}'_{\mathcal{F}}$ are as shown in (18) at the bottom of the page, where we recall that $N' = N + L - 1$ is the size of the matrices. We also note that the symbol "j" is used to denote $\sqrt{-1}$ and should not be confused with the variable j, which is often used as an index.

From (18), we observe that if for large values of the lag l, $\phi_{xx}(l) \to 0$, the diagonal elements of $\boldsymbol{\mathcal{R}}'_{\mathcal{F}}$ asymptotically approach the samples of the power spectral density of the input process

$$\Phi_{xx}(e^{-\mathbf{j}(2\pi i/N')}) = \sum_{l=-\infty}^{\infty} \phi_{xx}(l) e^{-j(2\pi i l/N')} \tag{19}$$

$$(\boldsymbol{\mathcal{R}}'_{\mathcal{F}})_{i,j} = \begin{cases} \displaystyle\sum_{l=-N'+1}^{N'-1} \left(1 - \frac{|l|}{N'}\right) \phi_{xx}(l) e^{-\mathbf{j}(2\pi l i/N')}, & i = j \\ \displaystyle\frac{1}{N'} \sum_{p=0}^{P-1} a_p \left[\frac{1}{\rho_p - e^{\mathbf{j}(2\pi j/N')}} \frac{1}{\rho_p^{-1} - e^{-\mathbf{j}(2\pi i/N')}} + \frac{1}{\rho_p^{-1} - e^{\mathbf{j}(2\pi j/N')}} \frac{1}{\rho_p - e^{-\mathbf{j}(2\pi i/N')}} \right], & i \neq j \end{cases} \tag{18}$$

as N' grows. The expressions for the nondiagonal elements are more complicated, but the important thing to notice is the factor of $1/N'$ outside of the summation. Assuming that none of the components ρ_p of the autocorrelation function are on the unit circle (in which case the autocorrelation function does not die out as $l \to \infty$—a case that we do not consider in this paper), then the summation is finite for all values of i and j, independent of N'. The factor $1/N'$ therefore pushes the nondiagonal elements to zero as the matrix size gets very large. This is what is meant when we say that the DFT transformation diagonalizes the autocorrelation matrix $E[\boldsymbol{x}(n)\boldsymbol{x}^H(n)]$ in an asymptotic sense [9]. As N' becomes very large, $\boldsymbol{\mathcal{R}}'_{\mathcal{F}}$, and thus, equivalently, $\boldsymbol{\mathcal{R}}_{\mathcal{F}}$ tends toward a diagonal matrix. However, it is shown in [10] that this is not a sufficient condition to ensure that the eigenvalues of $\boldsymbol{\mathcal{R}}_{\mathcal{F}}$ asymptotically approach its diagonal elements, as they would if the matrix was actually diagonal.

B. Windowing Matrix $\boldsymbol{\mathcal{P}}_{O,L}$

From the theory of circulant matrices [8], [24], we note that $\boldsymbol{\mathcal{P}}_{O,L}$ is the Fourier transform of a diagonal matrix and is therefore circulant. Moreover, the first column of $\boldsymbol{\mathcal{P}}_{O,L}$ is obtained according to the equation

$$(\boldsymbol{\mathcal{P}}_{O,L})_0 = \frac{1}{N'}\boldsymbol{\mathcal{F}}\begin{bmatrix}0\\ \vdots\\ 0\\ 1\\ \vdots\\ 1\end{bmatrix}\begin{matrix}\left.\begin{matrix}\\ \\ \\ \end{matrix}\right\}N-1\\ \left.\begin{matrix}\\ \\ \\ \end{matrix}\right\}L\end{matrix}. \tag{20}$$

It is then straightforward to show that the elements of the first column of $\boldsymbol{\mathcal{P}}_{O,L}$, i.e., the elements of $(\boldsymbol{\mathcal{P}}_{O,L})_0$, are given by

$$(\boldsymbol{\mathcal{P}}_{O,L})_{0,i} = \begin{cases} \dfrac{L}{N'} & i = 0 \\ \dfrac{1}{N'}\, e^{j(\pi i/N')(L+1)}\left(\dfrac{\sin(\pi i L/N')}{\sin(\pi i/N')}\right), & i \neq 0. \end{cases} \tag{21}$$

The subsequent columns of $\boldsymbol{\mathcal{P}}_{O,L}$ are then obtained by circularly shifting its first column.

C. Numerical Example

To gain some understanding of the nature of the matrices $\boldsymbol{\mathcal{R}}_{\mathcal{F}}$ and $\boldsymbol{\mathcal{P}}_{O,L}$ and their combined effect in the composite matrix $\boldsymbol{\mathcal{R}}^u_{xx} = \boldsymbol{\mathcal{R}}^*_{\mathcal{F}} \otimes \boldsymbol{\mathcal{P}}_{O,L}$, we present a numerical example. Consider the process $x_1(n)$ that has the following autocorrelation function:

$$\begin{aligned}\phi_{x_1x_1}(l) &= E[x_1^*(n)x_1(n-l)] \\ &= \frac{2\times(0.5)^{|l|} + 3\times(-0.3)^{|l|}}{5}.\end{aligned} \tag{22}$$

The process $x_1(n)$ may be generated by passing a unit-variance zero-mean white Gaussian noise through an ARMA model with the transfer function

$$H_1(z) = \frac{0.98745 - 0.18535z^{-1}}{1 - 0.2z^{-1} - 0.15z^{-2}}. \tag{23}$$

Fig. 2 shows a 3-D plot of the magnitudes of the elements of $\boldsymbol{\mathcal{R}}_{\mathcal{F}}$ for $x_1(n)$, when $N' = 32$. We note that although, as expected, the nondiagonal elements of $\boldsymbol{\mathcal{R}}_{\mathcal{F}}$ are much smaller than its diagonal elements, they remain nonzero.

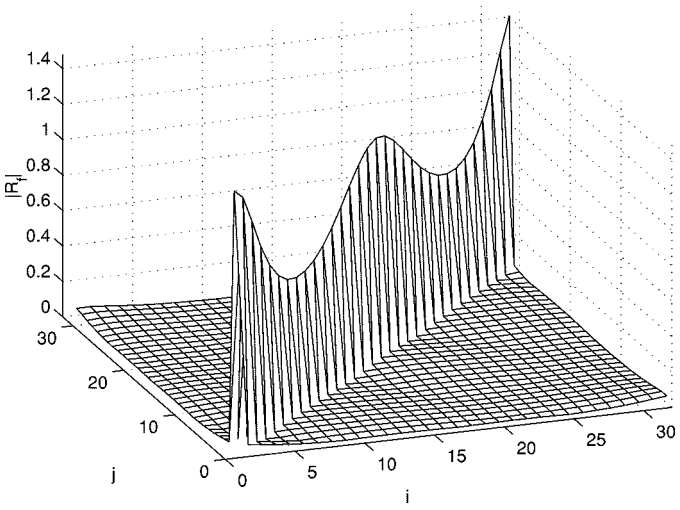

Fig. 2. Three-dimensional representation of $\boldsymbol{\mathcal{R}}_{\mathcal{F}}$ for process $x_1(n)$.

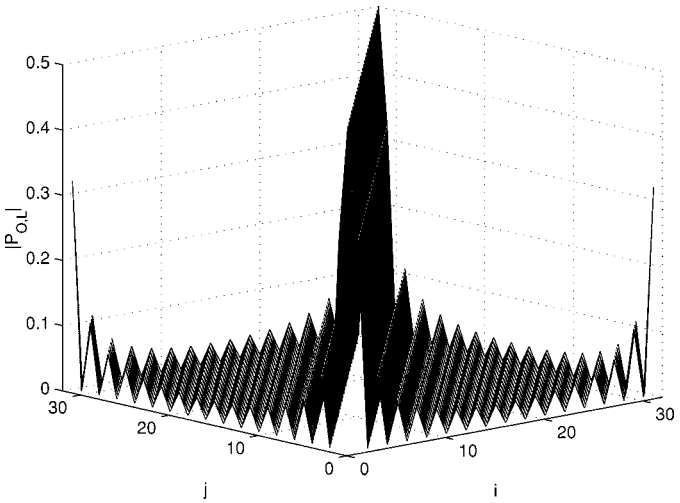

Fig. 3. Three-dimensional representation of $\boldsymbol{\mathcal{P}}_{O,L}$.

Fig. 3, on the other hand, presents a 3-D plot of the magnitude of the elements of $\boldsymbol{\mathcal{P}}_{O,L}$ when $L = 16$ and $N' = 32$. From this diagram, we clearly see that the nondiagonal elements of $\boldsymbol{\mathcal{P}}_{O,L}$, in magnitude, are smaller than their diagonal elements. Accordingly, we can argue that $\boldsymbol{\mathcal{R}}^u_{xx} = \boldsymbol{\mathcal{R}}^*_{\mathcal{F}} \otimes \boldsymbol{\mathcal{P}}_{O,L}$ is more diagonal than $\boldsymbol{\mathcal{R}}_{\mathcal{F}}$. This, of course, is a qualitative explanation of the effect of $\boldsymbol{\mathcal{P}}_{O,L}$ in further diagonalizing the matrix $\boldsymbol{\mathcal{R}}^u_{xx}$, as compared with $\boldsymbol{\mathcal{R}}_{\mathcal{F}}$. To have a more rigorous analysis, we need to quantify this observation. This can be best done by using Gerschgorin's theorem.

D. Gerschgorin's Theorem

Gerschgorin's theorem is stated as follows [7].

Theorem 1—Gerschgorin's Theorem: Let λ be an eigenvalue of an arbitrary $N\times N$ matrix $\boldsymbol{A} = [a_{i,j}]$. Then, there exists some integer i $(0 \le i \le N-1)$ such that

$$\begin{aligned}|a_{i,i} - \lambda| \le{}& |a_{i,0}| + |a_{i,1}| + \cdots + |a_{i,i-1}| \\ &+ |a_{i,i+1}| + \cdots + |a_{i,N-1}|.\end{aligned} \tag{24}$$

As many readers of this paper may not be familiar with Gerschgorin's theorem, a proof of that is provided in Appendix B for easy reference.

Stated another way, Gerschgorin's theorem says that each eigenvalue λ of $\boldsymbol{A}$ is "close" to one of its diagonal elements. How close is quantified by $r_i = \sum_{j=0,\, j\neq i}^{N-1} |a_{i,j}|$, which is the summation of the absolute values of the nondiagonal elements of the ith row of $\boldsymbol{A}$. If we define N so-called Gerschgorin disks in the complex plane, each centered at $a_{i,i}$ and of radius r_i, for $0 \leq i \leq N-1$, then the eigenvalues of $\boldsymbol{A}$ are distributed such that they all lie within the union of the Gerschgorin disks. A further extension of Gerschgorin's theorem states that when the Gerschgorin disks are disjoint, there will be one eigenvalue in each disk.

E. Asymptotic Behavior of $\boldsymbol{\mathcal{R}}_{xx}^{u}$

We are now ready to combine the results of the previous subsections and study the asymptotic behavior of $\boldsymbol{\mathcal{R}}_{xx}^{u}$ and its normalized version

$$\boldsymbol{\mathcal{R}}_{xx}^{u\mathcal{N}} = \boldsymbol{\Lambda}^{-1}\boldsymbol{\mathcal{R}}_{xx}^{u}. \tag{25}$$

We first note from Section III-A that the nondiagonal elements of $\boldsymbol{\mathcal{R}}'_{\mathcal{F}}$ tend to zero like $(1/N')$. However, when summed along any row of $\boldsymbol{\mathcal{R}}'_{\mathcal{F}}$, we add up $N'-1$ terms, where each is proportional to $(1/N')$. This will be a finite term and will not tend to zero as N' becomes infinitely large. In other words, the Gerschgorin disks have nonzero radii. This implies that the eigenvalues of $\boldsymbol{\mathcal{R}}'_{\mathcal{F}}$ and, thus, those of $\boldsymbol{\mathcal{R}}_{\mathcal{F}}$ may not all tend toward their respective diagonal elements as, indeed, they do not [10].

Now, consider the matrix $\boldsymbol{\mathcal{R}}_{xx}^{u} = \boldsymbol{\mathcal{R}}_{\mathcal{F}}^{*} \otimes \boldsymbol{\mathcal{P}}_{O,\,L}$. Again, for convenience and the same reasons that were given earlier for introducing $\boldsymbol{\mathcal{R}}'_{\mathcal{F}}$, we define $\boldsymbol{\mathcal{R}}'^{u}_{xx} = (1/N')\boldsymbol{\mathcal{R}}_{xx}^{u} = \boldsymbol{\mathcal{R}}_{\mathcal{F}}^{\prime *} \otimes \boldsymbol{\mathcal{P}}_{O,\,L}$ and study the properties of $\boldsymbol{\mathcal{R}}'^{u}_{xx}$. The radius of the ith Gerschgorin disk of $\boldsymbol{\mathcal{R}}'^{u}_{xx}$ is given by

$$r_i(\boldsymbol{\mathcal{R}}'^{u}_{xx}) = \sum_{j=0,\, j\neq i}^{N'-1} |(\boldsymbol{\mathcal{R}}'_{\mathcal{F}})_{i,\,j}| \cdot |(\boldsymbol{\mathcal{P}}_{O,\,L})_{i,\,j}|. \tag{26}$$

We show that $r_i(\boldsymbol{\mathcal{R}}'^{u}_{xx})$ tends to zero as $N' \to \infty$. To this end, we define $\gamma = \max_{i\neq j}[|(\boldsymbol{\mathcal{R}}_{\mathcal{F}})_{i,\,j}|] = \max_{i\neq j}[N' \times |(\boldsymbol{\mathcal{R}}'_{\mathcal{F}})_{i,\,j}|]$ as the maximum nondiagonal element of $\boldsymbol{\mathcal{R}}'_{\mathcal{F}}$, and from (18), we note that γ is a finite number. Using this in (26), we obtain

$$r_i(\boldsymbol{\mathcal{R}}'^{u}_{xx}) \leq \frac{\gamma}{N'} \sum_{j=0,\, j\neq i}^{N'-1} |(\boldsymbol{\mathcal{P}}_{O,\,L})_{i,\,j}|. \tag{27}$$

In addition, substituting (21) in (27), we get

$$\begin{aligned} r_i(\boldsymbol{\mathcal{R}}'^{u}_{xx}) &\leq \frac{\gamma}{N'^2} \sum_{j=1}^{N'-1} \left|\frac{\sin(\pi jL/N')}{\sin(\pi j/N')}\right| \\ &\leq \frac{\gamma}{N'^2} \sum_{j=1}^{N'-1} \left|\frac{1}{\sin(\pi j/N')}\right|. \end{aligned} \tag{28}$$

Moreover, assuming that N' is even, we may write

$$\begin{aligned} \sum_{j=1}^{N'-1} \left|\frac{1}{\sin(\pi j/N')}\right| &= \sum_{j=1}^{N'/2} \left|\frac{1}{\sin(\pi j/N')}\right| \\ &\quad + \sum_{j=N'/2+1}^{N'-1} \left|\frac{1}{\sin(\pi j/N')}\right| \\ &< 2\sum_{j=1}^{N'/2} \frac{1}{\sin(\pi j/N')}. \end{aligned} \tag{29}$$

The last inequality is due to the fact that the second summation on the right-hand side of the first line has one term less than the first summation. Noting that for $0 < x < \pi/2$, $1/\sin x < \pi/(2x)$, from (29), we get for $N' > 2$.

$$\sum_{j=1}^{N'/2} \frac{1}{\sin(\pi j/N')} < \frac{N'}{2} \sum_{j=1}^{N'/2} \frac{1}{j} < \frac{N'}{2} \ln N' \tag{30}$$

where the latter inequality can easily be confirmed numerically. Using (30) in (29) and the result in (28), we obtain

$$r_i(\boldsymbol{\mathcal{R}}'^{u}_{xx}) < \frac{\gamma \ln N'}{N'}. \tag{31}$$

The same results can also be obtained when N' is odd.

Noting that $(\ln N')/N'$ converges toward zero as N' grows, from (31), we conclude that the radii of the Gerschgorin disks of $\boldsymbol{\mathcal{R}}'^{u}_{xx}$ converge to zero as N' increases. Accordingly, for large N', the eigenvalues of $\boldsymbol{\mathcal{R}}'^{u}_{xx}$ and, thus, equivalently, those of $\boldsymbol{\mathcal{R}}_{xx}^{u}$, approach their respective diagonal elements. In other words, if we define diag$[\boldsymbol{\mathcal{R}}_{xx}^{u}]$ as the diagonal matrix consisting of the diagonal elements of $\boldsymbol{\mathcal{R}}_{xx}^{u}$ for large N'

$$\boldsymbol{\mathcal{R}}_{xx}^{u} \approx \text{diag}[\boldsymbol{\mathcal{R}}_{xx}^{u}] \tag{32}$$

in the *strong sense* that the eigenvalues of $\boldsymbol{\mathcal{R}}_{xx}^{u}$ and diag$[\boldsymbol{\mathcal{R}}_{xx}^{u}]$ are asymptotically the same. We note that $\boldsymbol{\mathcal{R}}_{\mathcal{F}}$ does not exhibit this property. Using (16) and (21) in (32) and noting that diag$[\boldsymbol{\mathcal{R}}_{\mathcal{F}}] = \boldsymbol{\Lambda}$, we get

$$\boldsymbol{\mathcal{R}}_{xx}^{u} \approx \frac{L}{N'}\,\boldsymbol{\Lambda}. \tag{33}$$

We are now ready to make our final comment with regard to the convergence behavior of the unconstrained FBLMS algorithm both with and without step-normalization.

Since the eigenvalues of $\boldsymbol{\mathcal{R}}_{xx}^{u}$ determine the various modes of convergence of the unconstrained FBLMS algorithm (without step normalization), (33) shows that the convergence behavior of the unconstrained FBLMS algorithm is determined by the diagonal elements of $\boldsymbol{\Lambda}$, which are nothing but the signal energy at various frequency bins. These are approximately equal to samples of the power spectral density of the filter input at the equally spaced frequencies $\omega_i = 2\pi i/N'$ for $i = 0,\, 1,\, \cdots,\, N'-1$ [see (19)].

On the other hand, from (33), we get, for large N'

$$\boldsymbol{\mathcal{R}}_{xx}^{u\mathcal{N}} = \boldsymbol{\Lambda}^{-1}\boldsymbol{\mathcal{R}}_{xx}^{u} \approx \frac{L}{N'}\,\boldsymbol{I}_{N'}. \tag{34}$$

This clearly shows that in the step-normalized unconstrained FBLMS algorithm all modes of convergence are equalized as N' increases. Therefore, one would expect to see very good convergence behavior from the step-normalized unconstrained FBLMS algorithm.

IV. Convergence Analysis of the Constrained FBLMS Algorithm

The convergence behavior of the constrained FBLMS algorithm without step normalization does not require any elaborate analysis of the form presented thus far. We can simply say that when step normalization is not applied to the constrained FBLMS algorithm, it is nothing but a computationally fast realization of the BLMS algorithm whose convergence behavior is well understood and is known to be very similar to the conventional LMS algorithm [2]. This may also be proved by analyzing the relevant frequency domain quantities at the cost of unnecessary complications in the analysis.

To arrive at some useful equations for convergence analysis of the constrained normalized FBLMS algorithm, we note that since $\boldsymbol{w}_{\mathcal{F}}(k)$ is already a constrained vector (meaning that the last $L-1$ elements of $\boldsymbol{\mathcal{F}}^{-1}\boldsymbol{w}_{\mathcal{F}}(k)$ are all zero), $\boldsymbol{\mathcal{P}}_{N,O}\boldsymbol{w}_{\mathcal{F}}(k) = \boldsymbol{w}_{\mathcal{F}}(k)$. Thus, the recursion (12) may be rearranged as

$$\boldsymbol{w}_{\mathcal{F}}(k+1) = \boldsymbol{w}_{\mathcal{F}}(k) + 2\mu_o\boldsymbol{\mathcal{P}}_{N,O}\boldsymbol{\Lambda}^{-1}\boldsymbol{\mathcal{X}}^{*}_{\mathcal{F}}(k)\boldsymbol{e}_{\mathcal{F}}(k). \quad (35)$$

Starting with this recursion and proceeding with similar line of derivations to those that led to (15), we find that the matrix whose eigenvalues control the convergence behavior of the step-normalized constrained FBLMS algorithm is

$$\boldsymbol{\mathcal{R}}^{c\mathcal{N}}_{xx} = \boldsymbol{\mathcal{P}}_{N,O}\boldsymbol{\mathcal{R}}^{u\mathcal{N}}_{xx}. \quad (36)$$

We thus need to study the eigenvalue spread of $\boldsymbol{\mathcal{R}}^{c\mathcal{N}}_{xx}$ as a measure of the convergence behavior of the step-normalized constrained FBLMS algorithm. However, note that since the rank of $\boldsymbol{\mathcal{P}}_{N,O}$ is N (this follows from the form of $\boldsymbol{\mathcal{P}}_{N,O}$) and $\boldsymbol{\mathcal{R}}^{u\mathcal{N}}_{xx}$ is full rank, the rank of $\boldsymbol{\mathcal{R}}^{c\mathcal{N}}_{xx}$ is N. This means that $N'-N = L-1$ of the eigenvalues of $\boldsymbol{\mathcal{R}}^{c\mathcal{N}}_{xx}$ are zero. These zero eigenvalues correspond to the fixed (set to zero) elements of $\boldsymbol{\mathcal{F}}^{-1}\boldsymbol{w}_{\mathcal{F}}(k)$ and thus have no contribution to the convergence modes of the algorithm. The convergence behavior of the step-normalized constrained FBLMS algorithm is therefore determined by the remaining N eigenvalues. In the rest of this section, we study the spread of the nonzero eigenvalues of $\boldsymbol{\mathcal{R}}^{c\mathcal{N}}_{xx}$ and show that this is always smaller than or equal to the eigenvalue spread of $\boldsymbol{\mathcal{R}}^{u\mathcal{N}}_{xx}$. For this, we use the following lemmas.

Lemma 1: Let the positive-definite Hermitian matrix $\boldsymbol{A}$ be partitioned as

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{B} & \boldsymbol{C} \\ \boldsymbol{D} & \boldsymbol{E} \end{bmatrix}.$$

Then

$$\frac{\lambda_{\max}(\boldsymbol{B})}{\lambda_{\min}(\boldsymbol{B})} \leq \frac{\lambda_{\max}(\boldsymbol{A})}{\lambda_{\min}(\boldsymbol{A})}$$

where $\lambda_{\max}(\cdot)$ and $\lambda_{\min}(\cdot)$ denote the maximum and minimum eigenvalues of the indicated matrices.

The proof of this lemma is given in Appendix C.

Lemma 2: If $\boldsymbol{A}$ and $\boldsymbol{B}$ are square matrices and $\boldsymbol{B}$ is invertible, then $\boldsymbol{A} \sim \boldsymbol{B}^{-1}\boldsymbol{A}\boldsymbol{B}$, where $\sim$ denotes that the two matrices have identical eigenvalues.

This lemma is a well-known result from the theory of matrices and is easily proved (for example, see [26]). We wish to compare the spread of nonzero eigenvalues of $\boldsymbol{\mathcal{R}}^{u\mathcal{N}}_{xx}$ and $\boldsymbol{\mathcal{R}}^{c\mathcal{N}}_{xx}$. However, we note that the results of the above lemmas may not be directly applied, as $\boldsymbol{\mathcal{R}}^{u\mathcal{N}}_{xx}$ is not a Hermitian matrix. To resolve this problem, we recall from the theory of the transform domain adaptive filters [8], [10] that step normalization and tap-input normalization are equivalent. In the context of the FBLMS algorithm, this is equivalent to replacing the transformed tap-input vector $\boldsymbol{x}_{\mathcal{F}}(k)$ by $\breve{\boldsymbol{x}}_{\mathcal{F}}(k) = \boldsymbol{\Lambda}^{-1/2}\boldsymbol{x}_{\mathcal{F}}(k)$ and using a common unnormalized step-size parameter. With this amendment, the comparison of the unconstrained and constrained FBLMS algorithms may be carried out by comparing the spread of nonzero eigenvalues of $\breve{\boldsymbol{\mathcal{R}}}^{u\mathcal{N}}_{xx} = E[\breve{\boldsymbol{x}}_{\mathcal{F}}(k)\breve{\boldsymbol{x}}^{H}_{\mathcal{F}}(k)]$ and $\breve{\boldsymbol{\mathcal{R}}}^{c\mathcal{N}}_{xx} = \boldsymbol{\mathcal{P}}_{N,O}\breve{\boldsymbol{\mathcal{R}}}^{u\mathcal{N}}_{xx}$. Using (9), we get

$$\breve{\boldsymbol{\mathcal{R}}}^{c\mathcal{N}}_{xx} = \boldsymbol{\mathcal{F}}\begin{bmatrix} \boldsymbol{I}_N & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}\boldsymbol{\mathcal{F}}^{-1}\breve{\boldsymbol{\mathcal{R}}}^{u\mathcal{N}}_{xx}. \quad (37)$$

Pre- and postmultiplying this by $\boldsymbol{\mathcal{F}}^{-1}$ and $\boldsymbol{\mathcal{F}}$ respectively, by lemma 2, we get

$$\breve{\boldsymbol{\mathcal{R}}}^{c\mathcal{N}}_{xx} \sim \begin{bmatrix} \boldsymbol{I}_N & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}\boldsymbol{\mathcal{F}}^{-1}\breve{\boldsymbol{\mathcal{R}}}^{u\mathcal{N}}_{xx}\boldsymbol{\mathcal{F}}. \quad (38)$$

Let

$$\boldsymbol{\mathcal{F}}^{-1}\breve{\boldsymbol{\mathcal{R}}}^{u\mathcal{N}}_{xx}\boldsymbol{\mathcal{F}} = \begin{bmatrix} \boldsymbol{B} & \boldsymbol{C} \\ \boldsymbol{D} & \boldsymbol{E} \end{bmatrix}$$

where the dimensions of $\boldsymbol{B}$, $\boldsymbol{C}$, $\boldsymbol{D}$ and $\boldsymbol{E}$ are the same as the corresponding submatrices in

$$\begin{bmatrix} \boldsymbol{I}_N & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}.$$

Then

$$\begin{bmatrix} \boldsymbol{I}_N & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}\boldsymbol{\mathcal{F}}^{-1}\breve{\boldsymbol{\mathcal{R}}}^{u\mathcal{N}}_{xx}\boldsymbol{\mathcal{F}} = \begin{bmatrix} \boldsymbol{B} & \boldsymbol{C} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}$$
$$\Rightarrow \breve{\boldsymbol{\mathcal{R}}}^{c\mathcal{N}}_{xx} \sim \begin{bmatrix} \boldsymbol{B} & \boldsymbol{C} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}. \quad (39)$$

Next, we show that

$$\begin{bmatrix} \boldsymbol{B} & \boldsymbol{C} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \sim \begin{bmatrix} \boldsymbol{B} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}. \quad (40)$$

To see this, we write

$$\begin{bmatrix} \boldsymbol{B} & \boldsymbol{C} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}\boldsymbol{q} = \lambda\boldsymbol{q} \quad (41)$$

and notice that the last $L-1$ elements of the eigenvector $\boldsymbol{q}$ must be zero to satisfy this equation. Hence, we observe that $\boldsymbol{C}$ plays no role in determining the eigenvalue λ and may be arbitrarily set to zero.

From the above observation, we obtain

$$\breve{\boldsymbol{\mathcal{R}}}^{c\mathcal{N}}_{xx} \sim \begin{bmatrix} \boldsymbol{B} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}$$

and

$$\breve{\boldsymbol{\mathcal{R}}}_{xx}^{u\mathcal{N}} \sim \begin{bmatrix} \boldsymbol{B} & \boldsymbol{C} \\ \boldsymbol{D} & \boldsymbol{E} \end{bmatrix}. \tag{42}$$

From these and using Lemma 1, we get

$$\frac{\lambda_{\max}(\boldsymbol{B})}{\lambda_{\min}(\boldsymbol{B})} = \frac{\lambda_{\max}(\breve{\boldsymbol{\mathcal{R}}}_{xx}^{c\mathcal{N}})}{\lambda'_{\min}(\breve{\boldsymbol{\mathcal{R}}}_{xx}^{c\mathcal{N}})} \leq \frac{\lambda_{\max}(\breve{\boldsymbol{\mathcal{R}}}_{xx}^{u\mathcal{N}})}{\lambda_{\min}(\breve{\boldsymbol{\mathcal{R}}}_{xx}^{u\mathcal{N}})} \tag{43}$$

where $\lambda'_{\min}(\breve{\boldsymbol{\mathcal{R}}}_{xx}^{c\mathcal{N}})$ denotes the smallest nonzero eigenvalue of $\breve{\boldsymbol{\mathcal{R}}}_{xx}^{c\mathcal{N}}$. In other words, the spread of the nonzero eigenvalues of $\breve{\boldsymbol{\mathcal{R}}}_{xx}^{c\mathcal{N}}$ is smaller than the eigenvalue spread of $\breve{\boldsymbol{\mathcal{R}}}_{xx}^{u\mathcal{N}}$. This means that the use of the windowing matrix $\boldsymbol{\mathcal{P}}_{N,O}$ in the constrained FBLMS algorithm results in a further reduction of the spread of the various modes of the algorithm, as compared with its unconstrained counterpart. Therefore, one should expect to see a better performance from the constrained FBLMS algorithm as compared with its unconstrained counterpart. The validity of this conclusion will be further explored below in the numerical examples.

V. Misadjustments

In [4], Lee and Un have given the misadjustment equations for different variations of the FBLMS algorithm, without the derivations, whereas in [5], the authors have given the derivations for the misadjustments of the FBLMS algorithm with a generalized windowing function ($\boldsymbol{\mathcal{P}}_{N,O}$ and $\boldsymbol{\mathcal{P}}_{O,L}$ are allowed to take on more general forms) and considering the effect of exponential averaging of the power-normalization vector. In this section, we report a general outline of the derivations for simplified expressions of the misadjustment equations from [8] and refer the reader to this text for the details.

We recall that for an adaptive algorithm, misadjustment is defined by the equation

$$\mathcal{M} = \frac{\xi_{\text{excess}}}{\xi_{\min}} \tag{44}$$

where ξ_{excess} is the excess MSE due to perturbation of the filter tap weights after the algorithm has reached its steady-state, and $\xi_{\min}$ is the minimum MSE that would be achieved by the optimum tap weights. For the conventional LMS algorithm, the excess MSE is given by the following equation and is evaluated after the convergence of the filter:

$$\xi_{\text{excess}} = E\left[|\boldsymbol{v}^H(n)\boldsymbol{x}(n)|^2\right] \tag{45}$$

where $\boldsymbol{v}(n) = \boldsymbol{w}(n) - \boldsymbol{w}_o$ is the tap-weight perturbation vector, and thus, $\boldsymbol{v}^T(n)\boldsymbol{x}(n)$ is an associated error quantity.

In the case of FBLMS algorithm, where the perturbation (error) vector $\boldsymbol{v}_{\mathcal{F}}(k)$ varies only once every block, the excess MSE is given by

$$\xi_{\text{excess}} = \frac{1}{LN'} E\left[||\boldsymbol{\mathcal{P}}_{O,L}\boldsymbol{\mathcal{X}}_{\mathcal{F}}(k)\boldsymbol{v}_{\mathcal{F}}(k)||^2\right] \tag{46}$$

where $\|\cdot\|$ denotes the length or Euclidean norm of a vector. We note that in (46), $\boldsymbol{\mathcal{P}}_{O,L}\boldsymbol{\mathcal{X}}_{\mathcal{F}}(k)\boldsymbol{v}_{\mathcal{F}}(k)$ is the frequency-domain equivalent of the error terms arising from the tap-weight perturbation vector $\boldsymbol{v}_{\mathcal{F}}(k)$. Since there are L output samples, we need to divide this length by L to get the mean-square of one sample. In addition, the DFT introduces a power gain of N' in the transformed samples. This also has been taken care of by dividing the result by N'.

The right-hand side of (46) may be expanded as follows:

$$\begin{aligned}\xi_{\text{excess}} &= \frac{1}{LN'} E\left[(\boldsymbol{\mathcal{P}}_{O,L}\boldsymbol{\mathcal{X}}_{\mathcal{F}}(k)\boldsymbol{v}_{\mathcal{F}}(k))^H(\boldsymbol{\mathcal{P}}_{O,L}\boldsymbol{\mathcal{X}}_{\mathcal{F}}(k)\boldsymbol{v}_{\mathcal{F}}(k))\right] \\ &= \frac{1}{LN'} E[\boldsymbol{v}_{\mathcal{F}}^H(k)\boldsymbol{\mathcal{X}}_{\mathcal{F}}^*(k)\boldsymbol{\mathcal{P}}_{O,L}\boldsymbol{\mathcal{X}}_{\mathcal{F}}(k)\boldsymbol{v}_{\mathcal{F}}(k)]\end{aligned} \tag{47}$$

where we have noted that $\boldsymbol{\mathcal{P}}_{O,L}^H\boldsymbol{\mathcal{P}}_{O,L} = \boldsymbol{\mathcal{P}}_{O,L}$. Assuming that $\boldsymbol{v}_{\mathcal{F}}(k)$ and $\boldsymbol{\mathcal{X}}_{\mathcal{F}}(k)$ are independent of each other, we obtain, from (47)

$$\xi_{\text{excess}} = \frac{1}{LN'} E[\boldsymbol{v}_{\mathcal{F}}^H(k)\boldsymbol{\mathcal{R}}_{xx}^u\boldsymbol{v}_{\mathcal{F}}(k)]. \tag{48}$$

Recalling that if $\boldsymbol{A}$ and $\boldsymbol{B}$ are $M \times N$ and $N \times M$ matrices, respectively, then $\text{tr}[\boldsymbol{AB}] = \text{tr}[\boldsymbol{BA}]$, where $\text{tr}[\cdot]$ denotes the trace of, and noting that $\boldsymbol{v}_{\mathcal{F}}^H(k)\boldsymbol{\mathcal{R}}_{xx}^u\boldsymbol{v}_{\mathcal{F}}(k)$ is a scalar, we obtain, from (48)

$$\begin{aligned}\xi_{\text{excess}} &= \frac{1}{LN'} E[\text{tr}[\boldsymbol{v}_{\mathcal{F}}^H(k)\boldsymbol{\mathcal{R}}_{xx}^u\boldsymbol{v}_{\mathcal{F}}(k)]] \\ &= \frac{1}{LN'} E[\text{tr}[\boldsymbol{v}_{\mathcal{F}}(k)\boldsymbol{v}_{\mathcal{F}}^H(k)\boldsymbol{\mathcal{R}}_{xx}^u]] \\ &= \frac{1}{LN'} \text{tr}[E[\boldsymbol{v}_{\mathcal{F}}(k)\boldsymbol{v}_{\mathcal{F}}^H(k)\boldsymbol{\mathcal{R}}_{xx}^u]] \\ &= \frac{1}{LN'} \text{tr}[\boldsymbol{K}_{\mathcal{F}}^H(k)\boldsymbol{\mathcal{R}}_{xx}^u]\end{aligned} \tag{49}$$

where $\boldsymbol{K}_{\mathcal{F}}(k) = E[\boldsymbol{v}_{\mathcal{F}}(k)\boldsymbol{v}_{\mathcal{F}}^H(k)]$.

Equation (49) is fundamental to the derivation of misadjustment expressions for different variations of the FBLMS algorithm. It has the familiar form that we have often seen in the misadjustment analysis of the conventional LMS algorithm. Using the update equations of various versions of the FBLMS algorithm and (49), in [8], the following results has been derived:

$$\mathcal{M}_{\text{FBLMS}}^{c} \approx \mu N\phi_{xx}(0) \tag{50}$$

$$\mathcal{M}_{\text{FBLMS}}^{u} \approx \mu N'\phi_{xx}(0) \tag{51}$$

$$\mathcal{M}_{\text{FBLMS}}^{c,\mathcal{N}} \approx \mu_o N/N' \tag{52}$$

$$\mathcal{M}_{\text{FBLMS}}^{u,\mathcal{N}} \approx \mu_o. \tag{53}$$

In these equations, as before, superscripts c and u refer to the constrained and unconstrained versions of the FBLMS algorithm, respectively, and $\mathcal{N}$ indicates that the step normalization has been applied. We also note that, similar to the results of the conventional LMS algorithm, (50)–(53) are valid only for small values of misadjustments (say smaller than 10%).

VI. Numerical Examples and Simulations

To summarize our theoretical observations of previous sections, we reiterate that in the FBLMS algorithm, the transformation of input samples by the DFT is the first step in diagonalizing the correlation matrix of the transformed data samples. The presence of the windowing matrix $\boldsymbol{\mathcal{P}}_{O,L}$ in the FBLMS recursion plays a key role in further diagonalizing the underlying correlation matrix. Moreover, the step normalization converts the diagonalized correlation matrix to a matrix proportional to

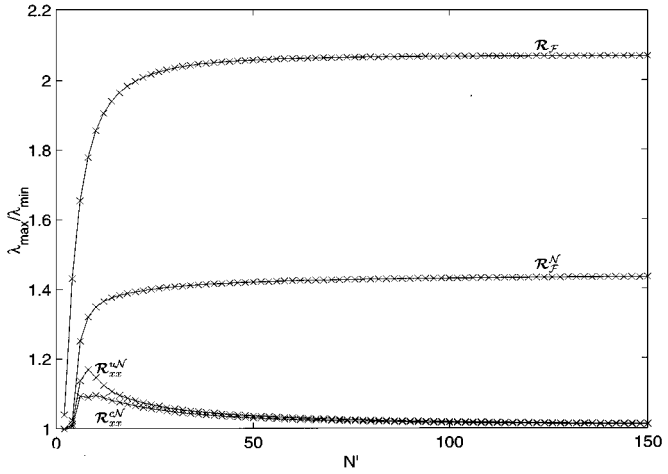

Fig. 4. Eigenvalue spread of the normalized matrices $\mathcal{R}_{\mathcal{F}}^{\mathcal{N}}$, $\mathcal{R}_{xx}^{u\mathcal{N}}$, and $\mathcal{R}_{xx}^{c\mathcal{N}}$ for process $x_1(n)$.

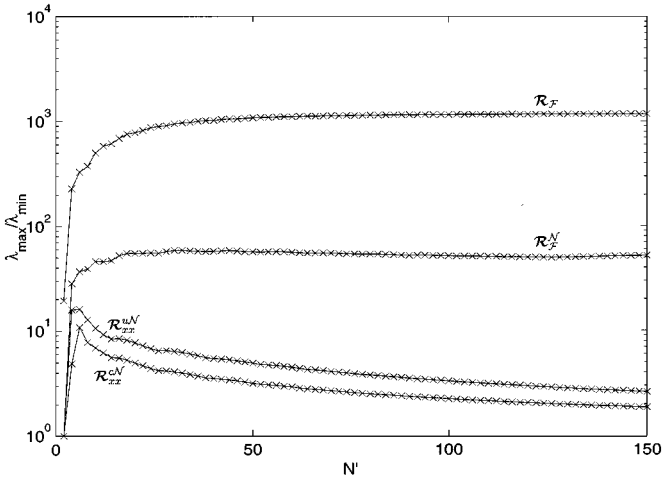

Fig. 5. Eigenvalue spread of the matrices $\mathcal{R}_{\mathcal{F}}$, $\mathcal{R}_{\mathcal{F}}^{\mathcal{N}}$, $\mathcal{R}_{xx}^{u\mathcal{N}}$, and $\mathcal{R}_{xx}^{c\mathcal{N}}$ for process $x_2(n)$, which is a low-pass process.

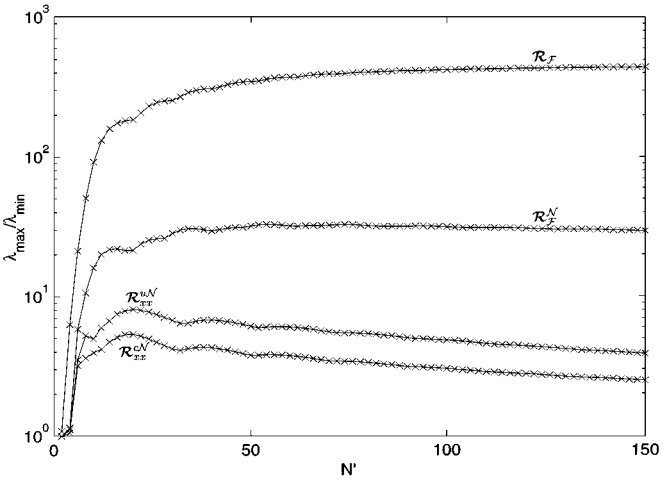

Fig. 6. Eigenvalue spread of the matrices $\mathcal{R}_{\mathcal{F}}$, $\mathcal{R}_{\mathcal{F}}^{\mathcal{N}}$, $\mathcal{R}_{xx}^{u\mathcal{N}}$, and $\mathcal{R}_{xx}^{c\mathcal{N}}$ for process $x_3(n)$, which is a bandpass process.

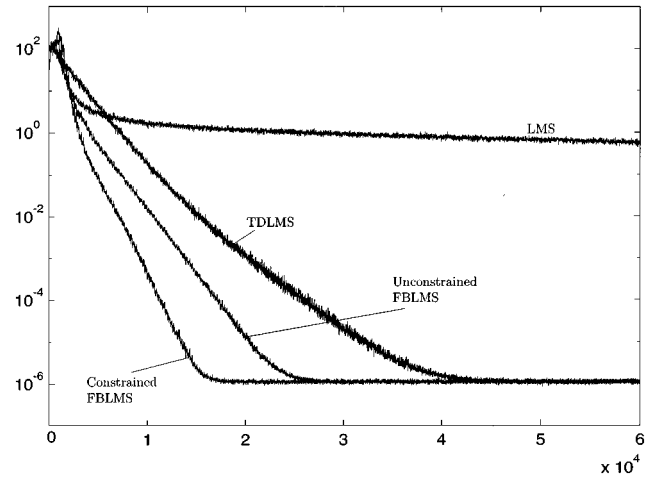

Fig. 7. Learning curves for the constrained FBLMS, unconstrained FBLMS, TDLMS using FFT, and time-domain LMS algorithms. $L = 256$, and $N' = 512$. Results are ensemble averages of 50 independent runs. The process $x_2(n)$ was used as input.

the identity matrix, equalizing all modes of convergence of the FBLMS algorithm. Further improvement may be brought about by applying the constraining matrix $\mathcal{P}_{N,O}$.

To see the effect of the DFT in diagonalizing the underlying correlation matrix and the further diagonalization (improvement) that is brought about by the windowing matrix $\mathcal{P}_{O,L}$, in Figs. 4–6, we have plotted the eigenvalue spread $\lambda_{\max}/\lambda_{\min}$ of the matrices $\mathcal{R}_{\mathcal{F}}$, $\mathcal{R}_{\mathcal{F}}^{\mathcal{N}} = \mathbf{\Lambda}^{-1}\mathcal{R}_{\mathcal{F}}$, $\mathcal{R}_{xx}^{u\mathcal{N}} = \mathbf{\Lambda}^{-1}\mathcal{R}_{xx}^{u}$, and $\mathcal{R}_{xx}^{c\mathcal{N}} = \mathcal{P}_{N,O}\mathcal{R}_{xx}^{u\mathcal{N}}$ for values of N' in the range of 2 to 150, and the block length $L = N'/2$. The input processes used in these three figures are $x_1(n)$, which has autocorrelation function given by (22) and $x_2(n)$ and $x_3(n)$, which are generated by passing white Gaussian noise through coloring filters $H_2(z) = 0.1 + 0.2z^{-1} + 0.3z^{-2} + 0.4z^{-3} + 0.4z^{-4} + 0.2z^{-5} + 0.1z^{-6}$ and $H_3(z) = 0.1 - 0.2z^{-1} - 0.3z^{-3} + 0.4z^{-3} + 0.4z^{-4} - 0.2z^{-5} - 0.1z^{-6}$, respectively. The process $x_1(n)$ is only slightly colored, whereas $x_2(n)$ and $x_3(n)$ are highly colored; $x_2(n)$ is a lowpass process, and $x_3(n)$ is a bandpass process. The figures confirm that the eigenvalue spread of $\mathcal{R}_{xx}^{u\mathcal{N}}$ will be lower than that of $\mathcal{R}_{\mathcal{F}}^{\mathcal{N}}$, verifying the conclusions that were drawn using Gerschgorin's theorem with regards to the asymptotic behavior of the matrices $\mathcal{R}_{\mathcal{F}}^{\mathcal{N}}$ and $\mathcal{R}_{xx}^{u\mathcal{N}}$ and the role of $\mathcal{P}_{O,L}$ in improving the convergence performance of the FBLMS algorithm. We also see that, as predicted, $\mathcal{R}_{xx}^{c\mathcal{N}}$ has a smaller eigenvalue spread (ignoring its zero eigenvalues) than $\mathcal{R}_{xx}^{u\mathcal{N}}$ and that the eigenvalue spread of both $\mathcal{R}_{xx}^{u\mathcal{N}}$ and $\mathcal{R}_{xx}^{c\mathcal{N}}$ asymptotically tend toward 1 as the filter length increases.

We have also included in Fig. 7 some simulation results running the constrained and unconstrained FBLMS, transform domain LMS (TDLMS) using FFT as transform, and the time-domain LMS algorithms, with $x_2(n)$ as input. The curves were generated using a filter length of $N = L = 256$, and the step sizes were chosen so that there is a 10% misadjustment in all cases. The results reconfirm that transformation of the filter input followed by step normalization will result in some improvement in the convergence behavior of the LMS algorithm. This is observed when comparing the learning curves of the time-domain LMS and the TDLMS algorithms. Moreover, as predicted by theory, further improvement is brought about by the windowing matrices $\mathcal{P}_{O,L}$ and $\mathcal{P}_{N,O}$ in the FBLMS algorithm. Furthermore, it is noted that the learning curves of the

FBLMS algorithm almost converges with only 1 mode, whereas the slow mode is visible for the TDLMS learning curve as the MSE approaches its minimum. The unconstrained FBLMS algorithm is slower than its constrained counterpart for two reasons: 1) According to (50) and (51) for similar misadjustment, the step-size parameter used by the constrained FBLMS algorithm is (N'/N) times larger than the one used by its unconstrained counterpart, and to a lesser extent 2) the eigenvalue spread of $\boldsymbol{\mathcal{R}}_{xx}^{c\mathcal{N}}$ is smaller than that of $\boldsymbol{\mathcal{R}}_{xx}^{u\mathcal{N}}$.

VII. Conclusion

In this paper, we presented an analysis of the FBLMS algorithm in the frequency domain. We studied, in detail, the effect of various processing components in the FBLMS structure on its convergence behavior. In particular, we showed that in addition to the step normalization, which is added to the FBLMS algorithm to improve its convergence behavior, the data windowings, which are inherent to the structure of FBLMS algorithm, also have some effect in further improving its convergence behavior. We also reported the misadjustment equations of different variations of the FBLMS algorithm from [8].

Appendix A
Evaluation of $\boldsymbol{\mathcal{R}}_{\mathcal{F}} = E[\boldsymbol{x}_{\mathcal{F}}(k)\boldsymbol{x}_{\mathcal{F}}^H(k)]$

Since $\boldsymbol{x}_{\mathcal{F}}(k)$ is the DFT of $\boldsymbol{x}(k)$, we may write

$$x_{\mathcal{F},i}(k) = \sum_{m=0}^{N'-1} x_m(k)e^{-\mathbf{j}(2\pi mi/N')}$$

$$x_{\mathcal{F},j}^*(k) = \sum_{n=0}^{N'-1} x_n^*(k)e^{\mathbf{j}(2\pi nj/N')}$$

as the ith and jth element of $\boldsymbol{x}_{\mathcal{F}}(k)$ and $\boldsymbol{x}_{\mathcal{F}}^*(k)$, respectively. Then, we obtain the (i, j)th element of $\boldsymbol{\mathcal{R}}_{\mathcal{F}}$ as

$$\begin{aligned}(\boldsymbol{\mathcal{R}}_{\mathcal{F}})_{i,j} &= E[x_{\mathcal{F},i}(k)x_{\mathcal{F},j}^*(k)] \\ &= \sum_{m=0}^{N'-1}\sum_{n=0}^{N'-1} \phi_{xx}(m-n)e^{-\mathbf{j}(2\pi/N')(mi-nj)} \end{aligned} \quad (54)$$

where $\phi_{xx}(m-n) = E[x_m(k)x_n^*(k)]$. We consider the two cases of $i = j$ and $i \neq j$ separately.

When $i = j$

$$(\boldsymbol{\mathcal{R}}_{\mathcal{F}})_{i,i} = \sum_{m=0}^{N'-1}\sum_{n=0}^{N'-1} \phi_{xx}(m-n)e^{-\mathbf{j}(2\pi i/N')(m-n)}. \quad (55)$$

With a change of variable $m-n = l$, after some rearrangement, we obtain

$$(\boldsymbol{\mathcal{R}}_{\mathcal{F}})_{i,i} = N' \sum_{l=-(N'-1)}^{N'-1} \left(1 - \frac{|l|}{N'}\right) \phi_{xx}(l)e^{-\mathbf{j}(2\pi li/N')}. \quad (56)$$

When $i \neq j$, we substitute (17) in (54) to get

$$(\boldsymbol{\mathcal{R}}_{\mathcal{F}})_{i,j} = \sum_{p=0}^{P-1} a_p \sum_{m=0}^{N'-1}\sum_{n=0}^{N'-1} \rho_p^{|m-n|}e^{-\mathbf{j}(2\pi/N')(mi-nj)}. \quad (57)$$

Now, the internal summation over n is broken up into two parts: from $n = 0$ to $m-1$ and from $n = m$ to $N'-1$. This, after some straightforward manipulation, gives

$$\begin{aligned}(\boldsymbol{\mathcal{R}}_{\mathcal{F}})_{i,j} = \sum_{p=0}^{P-1} a_p \Bigg[&\frac{1}{\rho_p - e^{\mathbf{j}(2\pi j/N')}} \frac{1}{\rho_p^{-1} - e^{-\mathbf{j}(2\pi i/N')}} \\ &+ \frac{1}{\rho_p^{-1} - e^{\mathbf{j}(2\pi j/N')}} \frac{1}{\rho_p - e^{-\mathbf{j}(2\pi i/N')}} \Bigg]. \end{aligned} \quad (58)$$

Dividing both sides of (56) and (58) by N' gives (18).

Appendix B
Proof of Gerschgorin's Theorem

Let $\boldsymbol{q}$ be an eigenvector corresponding to the eigenvalue λ of a matrix $\boldsymbol{A}$, and let q_i denote the element of $\boldsymbol{q}$ that has maximum magnitude. Then, noting that $(\boldsymbol{A} - \lambda\boldsymbol{I})\boldsymbol{q} = \boldsymbol{0}$ and writing out the equation for the ith line of this set of equations gives

$$\begin{aligned} &a_{i,0}q_0 + \cdots + a_{i,i-1}q_{i-1} + (a_{i,i} - \lambda)q_i \\ &\quad + a_{i,i+1}q_{i+1} + \cdots + a_{i,N-1}q_{N-1} = 0. \end{aligned}$$

Dividing through by q_i and rearranging the terms gives

$$\begin{aligned}(a_{i,i} - \lambda) = &- a_{i,0}\frac{q_0}{q_i} - \cdots - a_{i,i-1}\frac{q_{i-1}}{q_i} \\ &- a_{i,i+1}\frac{q_{i+1}}{q_i} - \cdots - a_{i,N-1}\frac{q_{N-1}}{q_i}. \end{aligned}$$

We now take the absolute value of both sides of this equation and use the fact that q_i is the maximum of all elements of $\boldsymbol{q}$ implying $|(q_j/q_i)| \leq 1$ for $0 \leq j \leq N-1$. Applying the triangle inequality $|a+b| \leq |a| + |b|$ yields (24).

Appendix C
Proof of Lemma 1

According to the minimax theorem the maximum eigenvalues of $\boldsymbol{A}$ and $\boldsymbol{B}$ may, respectively, be obtained by solving the following maximization problems [8]:

$$\lambda_{\max}(\boldsymbol{A}) = \max_{\|\boldsymbol{v}\|=1} \boldsymbol{v}^H\boldsymbol{A}\boldsymbol{v} \quad (59)$$

and

$$\lambda_{\max}(\boldsymbol{B}) = \max_{\|\boldsymbol{u}\|=1} \boldsymbol{u}^H\boldsymbol{B}\boldsymbol{u}. \quad (60)$$

If we define $\boldsymbol{A}$ as stated in the lemma and partition $\boldsymbol{v}$ as a stack of vectors $\boldsymbol{v}_0$ and $\boldsymbol{v}_1$ with lengths compatible with the submatrices $\boldsymbol{B}$, $\boldsymbol{C}$, $\boldsymbol{D}$, and $\boldsymbol{E}$, (59) may be expanded as

$$\lambda_{\max}(\boldsymbol{A}) = \max_{\|\boldsymbol{v}\|=1} [\boldsymbol{v}_0^H\boldsymbol{B}\boldsymbol{v}_0 + \boldsymbol{v}_0^H\boldsymbol{C}\boldsymbol{v}_1 + \boldsymbol{v}_1^H\boldsymbol{D}\boldsymbol{v}_0 + \boldsymbol{v}_1^H\boldsymbol{E}\boldsymbol{v}_1]. \quad (61)$$

Comparing (60) and (61), we conclude that

$$\lambda_{\max}(\boldsymbol{A}) \geq \lambda_{\max}(\boldsymbol{B}) \quad (62)$$

where equality occurs when $\boldsymbol{v}_1 = \boldsymbol{0}$, and thus, maximization of the right-hand side of (61) boils down to maximization of the first term in the brackets.

By a similar argument, we also obtain

$$\lambda_{\min}(\boldsymbol{A}) \leq \lambda_{\min}(\boldsymbol{B}). \tag{63}$$

Combining (62) and (63) completes the proof.

REFERENCES

[1] B. Widrow and M. E. Hoff, Jr., "Adaptive switching circuits," in *IRE WESCON Conv. Rec.*, 1960, pp. 96–104.

[2] G. A. Clark, S. K. Mitra, and S. R. Parker, "Block implementation of adaptive digital filters," *IEEE Trans. Circuits Syst.*, vol. CAS-28, pp. 584–592, 1981.

[3] G. A. Clark, S. R. Parker, and S. K. Mitra, "A unified approach to time and frequency domain realization of FIR adaptive digital filters," *IEEE Trans. Acoustic, Speech, Signal Process.*, vol. ASSP-31, pp. 1073–1083, 1983.

[4] J. C. Lee and C. K. Un, "Performance analysis of frequency-domain block LMS adaptive digital filters," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 173–189, Feb. 1989.

[5] P. C. W. Sommen, P. J. Gerwen, H. J. Kotmans, and A. J. E. M. Janssen, "Convergence analysis of a frequency-domain adaptive filter with exponential power averaging and generalized window function," *IEEE Trans. Circuits Syst.*, vol. CAS-34, July 1987.

[6] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 3rd ed, Singapore: McGraw-Hill, 1991.

[7] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*. Oxford, U.K.: Oxford Univ. Press, 1965.

[8] B. Farhang-Boroujeny, *Adaptive Filters: Theory and Applications*. Chichester, U.K.: Wiley, 1998.

[9] R. M. Gray, "On the asymptotic eigenvalue distribution of toeplitz matrices," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 725–729, Nov. 1972.

[10] F. Beaufays, "Transform domain adaptive filters: An analytic approach," *IEEE Trans. Signal Processing*, vol. 43, pp. 422–431, Feb. 1995.

[11] S. Haykin, *Adaptive Filter Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[12] B. Widrow and S. D. Stearns, *Adaptive signal processing*. Englewood Cliffs, NJ: Prentice-Hall, 1985.

[13] B. Farhang-Boroujeny and S. Gazor, "Generalized sliding FFT and its application to implementation of block LMS adaptive filters," *IEEE Trans. Signal Processing*, vol. 42, pp. 532–538, Mar. 1994.

[14] E. R. Ferrara, "Fast implementation of LMS adaptive filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 474–475, Aug. 1980.

[15] M. R. Asharif, T. Takebayashi, T. Chugo, and K. Murano, "Frequency-domain noise canceller: Frequency-bin adaptive filtering (FBAF)," in *Proc. ICASSP*, Tokyo, Japan, Apr. 7–11, 1986, pp. 41.22.1–4.

[16] M. R. Asharif and F. Amano, "Acoustic echo-canceler using the FBAF algorithm," *Trans. Commun.*, vol. 42, pp. 3090–3094, Dec. 1994.

[17] W. Kellermann, "Analysis and design of multirate systems for cancellation of acoustical echoes," in *Proc. IEEE ICASSP* , New York, 1988, pp. 2570–2573.

[18] S. S. Narayan, A. M. Peterson, and M. J. Narasimha, "Transform domain LMS algorithm," *IEEE Trans. Acoustics, Speech Signal Processing*, vol. ASSP-31, pp. 609–615, 1983.

[19] B. Farhang-Boroujeny and S. Gazor, "Selection of orthonormal transforms for improving the performance of the transform domain normalized LMS Algorithm," *Proc. Inst. Elect. Eng. F*, vol. 139, pp. 327–335, 1992.

[20] B. Farhang-Boroujeny, "Fast LMS/Newton algorithms based on autoregressive modeling and their applications to acoustic echo cancellation," *IEEE Trans. Signal Processing*, vol. 45, pp. 1987–2000, Aug. 1997.

[21] ——, "Analysis and efficient implementation of partitioned block LMS adaptive filters," *IEEE Trans. Signal Processing*, vol. 44, pp. 2865–2868, Nov. 1996.

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

[23] D. Mansour and A. H. Gray Jr., "Unconstrained frequency-domain adaptive filter," *IEEE Trans. Acoust. Speech Signal Processing*, vol. ASSP-30, pp. 726–734, Oct. 1982.

[24] S. Barnett, *Matrices: Methods and Applications*. New York: Oxford Univ. Press, 1990.

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

[26] F. Ayres Jr., "Theory and problems of matrices," in *Schaum's Outline Series*. New York: McGraw-Hill, 1974.

B. Farhang-Boroujeny (SM'90) received the B.Sc. degree in electrical engineering from Teheran University, Teheran, Iran, in 1976, the M.Eng. degree from University of Wales Institute of Science and Technology, Cardiff, U.K., in 1977, and the Ph.D. degree from Imperial College, University of London, London, U.K., in 1981.

From 1981 to 1989, he was with Isfahan University of Technology, Isfahan, Iran. Since 1989, he has been with the National University of Singapore. His current scientific interests are adaptive filters, data transmission, and recording channels. He is the author of the book *Adaptive Filters: Theory and Applications* (New York: Wiley, 1998).

Dr. Farhang-Boroujeny received the UNESCO Regional Office of Science and Technology for South and Central Asia Young Scientists Award in 1987, in recognition of his outstanding contribution in the field of computer applications and informatics.

Kheong Sann Chan was born in Melbourne, Australia, in 1972 and grew up in Singapore. He received the B.A. degree in mathematics and physics in 1994 and the B.Sc. degree in electrical engineering in 1996 from Northwestern University, Evanston, IL. He subsequently returned to Singapore to pursue the Ph.D. degree at the National University of Singapore.

His research work has focused on the analysis and implementation of adaptive filtering algorithms in the frequency domain.