A Modified Block FTF Adaptive Algorithm with Applications to Underwater Target Detection

Mohammed A. Hasan and Mahmood R. Azimi-Sadjadi, *Senior Member, IEEE*

Abstract— **In this paper, the problem of weighted block recursive least squares (RLS) adaptive filtering is formulated in the context of block fast transversal filter (FTF) algorithm. This "modified block FTF algorithm" is derived by modifying the constrained block-LS cost function to guarantee global optimality. This new soft-constrained algorithm provides an efficient way of transferring weight information between blocks of data. The tracking ability of the algorithm can be controlled by varying the block length and/or a soft constrained parameter. This algorithm is computationally more efficient compared with other LS-based schemes. The effectiveness of this algorithm is tested on a real-life problem dealing with underwater target identification from acoustic backscatter. The process involves the identification of the presence of resonance in the acoustic backscatter from a target of unknown shape submerged in water.**

I. Introduction

BLOCK LMS (BLMS) adaptive filtering was studied in several papers [1], [2] as an efficient means to perform adaptive filtering on nonstationary data. Using this scheme, the filter tap weights are updated once per each block rather than once per each sample as with the conventional filters. Among the benefits of this class of filters are: low computational requirements, especially when implemented using fast transform schemes, reduced round-off error effects, high inherent parallelism, etc [1]. However, the learning rate in this algorithm is not data dependent, and additionally, LMS-based algorithms typically suffer from speed-accuracy trade-off problems.

In [3], a block FTF structure was developed that can be used to process either consecutive blocks of data or discontinuous variable length blocks of data. In this method, a block-least squares (LS) cost function is minimized in every block independently. As a result, this algorithm guarantees a locally optimal solution in each block. This algorithm is computationally very efficient compared with other LS-based schemes. To transfer weight information from one block to the next, a soft constrained method was suggested by modifying the block-LS cost function. The tracking ability of this algorithm can be controlled by varying the block length and/or the soft constrained parameter [3]. For example, in many applications, fast tracking is required only over an initial "start-up" time, after which only much slower tracking will be sufficient. This can be achieved by using a) smaller size blocks at the start and then gradually increasing the block length and/or b) smaller soft-constrained parameter at the beginning and then increasing this parameter. The latter algorithm is referred to as "growing memory tracking" in [3]. However, due to the modifications to the block-LS cost function, the optimality is traded in favor of information transfer between the data blocks.

We will begin by showing how the soft-constrained term in [3] can be chosen to guarantee a near optimal solution. Then, a new modified block FTF algorithm is suggested that provides a more efficient means of weight transferring between the data blocks while maintaining the optimality of the solution. The implementation of this scheme using the original block FTF structure is then suggested. The effectiveness of our algorithm is tested on the underwater target identification problem from acoustic backscattered signals.

Various signal processing schemes [4]–[8] have been proposed in the literature to extract resonance information from the acoustic backscatter from objects of regular shapes such as thin spherical or cylindrical shells submerged in water. In [4], a joint time-frequency analysis of the impulse response of a spherical shell has been studied for characterization of the surface waves on an elastic shell using the Wigner–Ville distribution. The wavelet transform using five-cycle cosine-modulated Gaussian wavelet approximations was applied to the impulse response of a spherical shell of differing thickness to examine resonance characteristics of the target [5]. A wavelet-based classifier that uses an artificial neural network to adaptively compute discriminatory information on a target in the form of locations, sizes, and weights of Gaussian patches in time scale is described in [6]. The method in [7] uses a short-time Fourier transform to determine the resonance spectrum of submerged elastic cylindrical wires.

In [8], an RLS-based adaptive filtering scheme for the separation of resonant and specular components in the backscattered signal from objects of arbitrary shape is developed. The test results in this reference indicated that unlike the previous methods, no underlying model assumptions on the elastic return, and no *a priori* knowledge on the signal statistics would be required. In addition, this method offers excellent robustness in presence of noise and great performance for relatively stationary backscattered signals and is ideally suited for on-line processing situations. However, these results also revealed the interesting fact that for certain aspect angles, where the backscattered signal exhibits highly nonstationary

Manuscript received January 21, 1995; revised March 20, 1996. This work was supported by the Office of Naval Research. Technical agent was Naval Coastal Systems Station, Panama City, FL. The associate editor coordinating the review of this paper and approving it for publication was Dr. Andreas Spanias.

The authors are with the Department of Electrical Engineering, Colorado State University, Fort Collins, CO 80523 USA.

Publisher Item Identifier S 1053-587X(96)06674-3.

behavior, the system attempts to track the variations of the backscattered. This problem can be circumvented if the tracking ability of the adaptation algorithm can be controlled. Since there exists a time delay between the onset of the specular and resonant components, we start out with fast tracking before the appearance of the resonant component and slow tracking (or no tracking at all) when this component has occurred. This is addressed in this paper by using the proposed modified block FTF structure.

This paper is organized as follows. In Section II, the necessary notation, vector space interpretation of the LS problem, and analysis of soft-constrained algorithm are introduced. The modified block FTF is derived in Section III. Generalization of the modified block FTF is derived in Section IV. The performance of the algorithm is demonstrated, and the test results are presented in Section V.

II. Block FTF Adaptive Filter [3]

The original block FTF problem as stated in [3] is the determination of the weight vector W_{Nlk} that minimizes the block LS cost function

$$\xi_{Nlk} = \sum_{r=k-l+1}^{k} (d(r) - W_{Nlk}X_{Nr})^*(d(r) - W_{Nlk}X_{Nr}) \quad (1)$$

where $*$ denotes conjugate transpose, $X_{Nr} = [x^*(r) \quad \cdots \quad x^*(r-N+1)]^*$ is the input vector with $x(r)$ being the input signal at time r, and where

- $d(r)$ desired signal
- N number of parameters in the weight vector W_{Nlk}
- l length of the block
- k highest time index in the block.

Next, let us define the general aggregate data matrix and desired vector in (2), which appears at the bottom of the page, and $\boldsymbol{d}_{lk} = [d(k) \quad d(k-1) \quad \cdots \quad d(k-l+1)]^*$.

Defining the error block vector $\epsilon_{Nlk} = \boldsymbol{d}_{lk} - X_{Nlk}W^*_{Nlk}$, (1) can be rewritten as $\xi_{Nlk} = \epsilon^*_{Nlk}\epsilon_{Nlk}$. Minimizing (1) with respect to W_{Nlk} gives the normal equation $\epsilon^*_{Nlk}X_{Nlk} = 0$ from which the block-LS solution can be obtained as

$$W_{Nlk} = \boldsymbol{d}^*_{lk}K_{Nlk}. \quad (3)$$

In addition, we can write $\epsilon_{Nlk} = P^{\perp}_{Nlk}\boldsymbol{d}_{lk}$, and $\xi_{Nlk} = \boldsymbol{d}^*_{lk}P^{\perp}_{Nlk}\boldsymbol{d}_{lk}$, where $K_{Nlk} = X_{Nlk}\{X^*_{Nlk}X_{Nlk}\}^{\#}$, $P_{Nlk} = X_{Nlk}\{X^*_{Nlk}X_{Nlk}\}^{\#}X^*_{Nlk}$, and $\#$ represents any generalized inverse that usually reduces to the standard inverse when the quantity involved is invertible.

Note that in the multichannel case, $x(r)$ is a $p \times 1$ vector, and $d(r)$ is a $q \times 1$ vector that makes W_{nlk} a $q \times np$ matrix.

Equation (3) can be solved recursively in every block using either an RLS-based scheme that requires $O(N^2)$ operations per sample or an FTF-type algorithm [3] with only $O(N)$ computational efforts per sample.

A. Geometric Projections and Updating Formulae

To develop a transversal filter structure, the following definitions are needed. The operator that projects an L-dimensional vector onto the subspace spanned by the columns of an $L \times n$ data matrix X is $P_X = X(X^*X)^{-1}X^*$, and its orthogonal complement is $P^{\perp}_X = I - X(X^*X)^{-1}X^*$. Note that $P^2_X = P_X$, and consequently, $P^{\perp}_X P_X = 0$. A transversal filter operator defined in terms of the data matrix X is $K_X = X(X^*X)^{-1}$. All residuals and cross-correlation coefficients may be expressed in the form $u^*P^{\perp}_{X,z}v$, and all transversal filters may be written as u^*K_X [9], [10]. Updating transversal parameters involves appending new information to the data matrix. Thus, the new data matrix will have $\{X\} \oplus \{z\}$ as its subspace. The quantity $P_{X,z}$ will denote projection on the subspace $\{X\} \oplus \{z\}$. spanned by the columns of X and z. Augmenting a vector z to the collection of vectors X leads to the following projection updating rule (see [9] and [10]):

$$u^*P^{\perp}_{X,z}v = u^*P^{\perp}_X v - u^*P^{\perp}_X z(z^*P^{\perp}_X z)^{-1}z^*P^{\perp}_X v. \quad (4a)$$

The transversal filters are updated using

$$u^*K_{X,z} = [u^*K_X|0] + u^*P^{\perp}_X z(z^*P^{\perp}_X z)^{-1}[-z^*K_X|I], \quad (4b)$$

$$u^*K_{z,X} = [0|u^*K_X] + u^*P^{\perp}_X z(z^*P^{\perp}_X z)^{-1}[I|-z^*K_X] \quad (4c)$$

where u, v, and z are matrices in $C^{p\times L}$ where C is the field of complex numbers. The choice of the matrices u and v depends on the parameters to be updated, and z determines the type of update being performed on the parameters.

In the sequel, the unit vector e_i will be used to denote a vector with all entries as zeros except 1 in the ith position. The dimension of this vector is conformable with the context in which it appears.

Let $X = \begin{bmatrix} X_1 \\ x_2 \end{bmatrix}$ such that X_1 is an $l-1 \times n$ matrix forming the first $l-1$ rows of X, and x_2 is the lth row of X. Then

$$\begin{bmatrix} K_{X_1} \\ 0 \end{bmatrix} = K_X + P^{\perp}_X e_l(e^*_l P^{\perp}_X e_l)^{-1}[-e^*_l K_X]. \quad (5)$$

Similarly, if we use the decomposition $X = \begin{bmatrix} x_1 \\ X_2 \end{bmatrix}$, where x_1 is the first row of X and X_2 is a matrix of the last $l-1$ rows of X, then

$$\begin{bmatrix} 0 \\ K_{X_2} \end{bmatrix} = K_X + P^{\perp}_X e_1(e^*_1 P^{\perp}_X e_1)^{-1}[-e^*_1 K_X]. \quad (6)$$

$$\begin{aligned} X_{nlk} &= [\boldsymbol{x}_{lk} \quad \boldsymbol{x}_{lk-1} \quad \cdots \quad \boldsymbol{x}_{lk-n+1}] \\ &= \begin{bmatrix} x^*(k) & x^*(k-1) & \cdots & x^*(k-n+1) \\ x^*(k-1) & x^*(k-2) & \cdots & x^*(k-n) \\ \cdots & \cdots & \cdots & \cdots \\ x^*(k-l+1) & x^*(k-l) & \cdots & x^*(k-n-l+2) \end{bmatrix} \end{aligned} \quad (2)$$

Note that appending e_1 or e_l to the columns of X leads to the following projection updating rules:

$$P^{\perp}_{X;e_1} = \begin{bmatrix} 0 & \mathbf{0}^* \\ \mathbf{0} & P^{\perp}_X \end{bmatrix}, \quad \text{and} \quad P^{\perp}_{X;e_l} = \begin{bmatrix} 0 & P^{\perp}_X \\ \mathbf{0} & \mathbf{0}^* \end{bmatrix}. \tag{7}$$

The block FTF algorithm is the direct result of certain substitutions for u, v, z in (4)–(6) that lead to a closed set of recursions. The original block FTF algorithm is given in [3] (see Tables II–IV in this reference).

B. Block FTF and Soft-Constrained Algorithm

In the block FTF algorithm, the weights are updated blockwise using an order update procedure, and the filtering is performed on each block individually. To transfer the weight information interblockwise, the soft-constrained algorithm [3] can be used. In this algorithm, the following soft-constrained block-LS cost function is minimized:

$$\xi_{Nlk} = \mu_k \|W_{Nlk} - W_0\|^2 + \sum_{r=k-l+1}^{k} (d(r) - W_{Nlk}X_{Nr})^*(d(r) - W_{Nlk}X_{Nr}) \tag{8}$$

where μ_k is a nonnegative quantity used to weight the effect of initial conditions on the cost criterion, and W_0 is a $1 \times N$ row vector that represents some initial setting for W_{Nlk} prior to processing the data from $k-l+1$ to k. The vector W_0 could be the weight solution from the previous block, in which case, this algorithm provides a way to transfer information from previous data blocks to the current data block. The soft constraint term allows greater flexibility in varying the tracking ability of the block FTF and, at the same time, avoiding singularity difficulties. By choosing very large values of μ_k, the tracking ability of block FTF algorithm can be slowed down.

The updating rules for this algorithm are the same as in the order update algorithm [3, Table IV], except that the matrices X_{Nlk} and $\boldsymbol{d}_{lk}$ are replaced by $X_{N,l+N,k}$ and $\boldsymbol{d}_{l+N,k}$, respectively, which are defined by (9), which appears at the bottom of the page, and

$$\boldsymbol{d}_{l+N,k} = [d(k), d(k-1), \cdots, d(k-l+1), \mu^{1/2}W_{0,N-1}, \cdots, \mu^{1/2}W_{0,0}]^*$$

where $W_0 = [W_{0,0} \quad \cdots \quad W_{0,N-1}]$, and the weight solution is

$$W_{N,l+N,k} = \boldsymbol{d}^*_{l+N,k}K_{N,l+N,k}. \tag{10}$$

In [3], three different schemes were suggested to control the tracking capability of block FTF by changing the soft-constraint parameter μ_k and/or the block length l. These are called growing memory tracking, fixed memory tracking, and variable memory tracking. In the growing memory tracking, the tracking capability of the algorithm is steadily diminished by increasing the soft-constraint parameter μ_k while the block length is kept fixed. Fixed memory tracking, on the other hand, offers better tracking of slow variations in W_{Nlk}. In this algorithm, the soft-constraint parameter μ_k and the block length l take certain fixed values. Finally, in the variable memory tracking algorithm, both μ_k and l are varied from block to block. In the next section, we analyze the optimality properties of the soft-constrained block FTF algorithm.

C. Analysis of Soft Constrained Block FTF Algorithms

In this section, we investigate rigorously the effects of the parameter μ_k and the initial weight vector W_0 on the optimality of soft-constrained block-LS solution. For notational convenience, we will drop the subscripts of $X, W, \boldsymbol{d}$ and use X_i, W_i, and $\boldsymbol{d}_i$ to denote the data matrix, the weight vector, and desired vector for the ith block. In the soft constrained block-LS problem, the cost function to be minimized can be rewritten as

$$\text{Minimize } \xi_i = \mu_i(W_i - W_0)(W_i - W_0)^* + (\boldsymbol{d}^*_i - W_iX^*_i)(\boldsymbol{d}^*_i - W_iX^*_i)^*. \tag{11}$$

The factor μ_i allows the control of the influence of the soft constrained term on the squared error sum. The main purpose of using soft constraint, besides keeping the weights bounded and, hence, providing numerical stability, is to give certain weight to some initial weight vector W_0 of previous data block that might have been obtained by some other methods. It can easily be checked that the solution that minimizes ξ_i in (11) is

$$W_i^{\text{soft}} = W_0 + (\boldsymbol{d}^*_i - W_0X^*_i)X_i\{\mu_i I + X^*_iX_i\}^{-1}. \tag{12}$$

Let $W_i^{\text{LS}} = \boldsymbol{d}^*_iX_i(X^*_iX_i)^{-1}$ be the LS solution for the ith block; then, (12) can be rearranged so that

$$W_i^{\text{soft}} = W_0 + (W_i^{\text{LS}} - W_0)X^*_iX_i\{\mu_i I + X^*_iX_i\}^{-1} \tag{13}$$

or equivalently

$$W_i^{\text{soft}} = W_i^{\text{LS}} - \mu_i(W_i^{\text{LS}} - W_0)\{\mu_i I + X^*_iX_i\}^{-1}. \tag{14}$$

By examining the equivalent forms (12)–(14), it is obvious that substituting $\mu_i = 0$ in (14) yields the LS solution, i.e., maximum tracking, whereas substituting $\mu_i = \infty$ in (13) yields the initial weight vector W_0, i.e., no tracking. This implies that the soft-constrained solution for $\mu_i \neq 0$ is neither locally nor globally optimal. The following theorem shows the

$$X_{Nl+Nk} = [\boldsymbol{x}_{l+Nk} \quad \boldsymbol{x}_{l+Nk-1} \quad \cdots \quad \boldsymbol{x}_{l+Nk-N+1}] = \begin{bmatrix} x^*(k) & x^*(k-1) & \cdots & x^*(k-N+1) \\ x^*(k-1) & x^*(k-2) & \cdots & x^*(k-N) \\ \cdots & \cdots & \cdots & \cdots \\ x^*(k-l+1) & x^*(k-l) & \cdots & x^*(k-N-l+2) \\ 0 & 0 & \cdots & \mu^{1/2} \\ \cdots & \cdots & \cdots & \cdots \\ \mu^{1/2} & 0 & \cdots & 0 \end{bmatrix} \tag{9}$$

relation between the soft-constrained solution and the global LS solution of two consecutive blocks.

Theorem 2.1: Consider two consecutive blocks of lengths l_1 and l_2 with data matrices and desired vectors X_1, $\boldsymbol{d}_1$ and X_2, $\boldsymbol{d}_2$, respectively. Let W_1 and W_2 be the LS solutions of the two individual blocks, i.e., $W_i = \boldsymbol{d}_i^* X_i (X_i^* X_i)^{-1}$ for $i = 1, 2$. Then, the LS solution for the whole data sequence is

$$W^{\mathrm{LS}} = (\boldsymbol{d}_1^* X_1 + \boldsymbol{d}_2^* X_2)\{X_1^* X_1 + X_2^* X_2\}^{-1} \tag{15a}$$

or alternatively

$$W^{\mathrm{LS}} = W_1 + (W_2 - W_1) X_2^* X_2 \{X_1^* X_1 + X_2^* X_2\}^{-1} \tag{15b}$$

and the soft-constrained solution is

$$W^{\mathrm{soft}} = W_1 + (W_2 - W_1) X_2^* X_2 \{\mu I + X_2^* X_2\}^{-1}. \tag{16}$$

Additionally, there exists a constant $A \geq 0$ such that

$$\|W^{\mathrm{soft}} - W^{\mathrm{LS}}\|_{\min} \leq A(\lambda_{\max}(X_1^* X_1) - \lambda_{\min}(X_1^* X_1)).$$

This minimum is achieved when

$$\mu = \frac{\lambda_{\max}(X_1^* X_1) + \lambda_{\min}(X_1^* X_1)}{2}$$

for which the soft-constrained solution corresponds to a nearly optimal solution.

Proof: See Appendix A.

Remark: Comparing (15b) and (16) yields that W^{soft} and W^{LS} are identical if $X_1^* X_1 = \mu I$, which is not practically feasible. However, this suggests that to minimize the difference $\|W^{\mathrm{soft}} - W^{\mathrm{LS}}\|$, we have to choose μ that minimizes $\|X_1^* X_1 - \mu I\|$. Note that $W^{\mathrm{soft}} = W^{\mathrm{LS}}$ if all eigenvalues of $X_1^* X_1$ are equal. In that case, $X_1^* X_1 = \lambda I = \mu I$, i.e., the data are decorrelated and stationary, which rarely occurs in reality as mentioned before.

The results of Theorem 2.1 can easily be extended to the general case by considering all the previous data samples to be contained in the first block. As can be observed from these results, if the solution is to be near the global optimal solution for multiple blocks, the soft-constrained parameter should be chosen, depending on the eigenvalues of the prior data correlation matrix. Even though there are a number of algorithms available that allow recursive determination of these eigenvalues, this is not a particularly feasible scheme.

In the next section, we develop a new class of block FTF algorithms that allows greater flexibility in tracking by adjusting the block length, and at the same time, all the information in the previous blocks are fully transferred to determine the current block weight. This algorithm differs from the block FTF algorithm [3] in several aspects. As shown before, the block FTF in [3] transfers the useful information between blocks using weighted initial condition, whereas this algorithm transfers information using some weighted correlation matrices of previous blocks, which yields better tracking, whereas for certain weighting, it guarantees globally optimal solution; this is a feature that is not present in the block FTF [3]. Moreover, this algorithm can be used as a time update algorithm by choosing the block length to be one.

III. The Modified Block FTF Algorithm

A. A New Soft-Constrained Block-LS Cost Function

While the soft constrained block FTF algorithm of [3] is very convenient to use, it does not provide an optimal solution when needed especially for the first few blocks where fast tracking is desired. As shown above, the maximum amount of information that can be transferred from a previous block X_1 to a current block X_2 is dependent on the quantity $(\lambda_{\max}(X_1^* X_1) - \lambda_{\min}(X_1^* X_1))/2$, which could be very large in a nonstationary environment and highly correlated data. In this section, we develop a soft constrained algorithm that can transfer as much information as needed by weighting the previous data blocks. To establish a better choice of the soft constraint term, let us first consider the following soft-constrained LS problem:

$$\begin{aligned}\text{Minimize } \xi_i = {} & (W_i - W_{i-1}) P (W_i - W_{i-1})^* \\ & + (\boldsymbol{d}_i^* - W_i X_i^*)(\boldsymbol{d}_i^* - W_i X_i^*)^*\end{aligned} \tag{17}$$

where P is a positive semi-definite matrix. Note that (17) becomes the original soft-constrained problem in (11) if $P = \mu_i I$ and $W_{i-1} = W_0$. The solution of (17) can be shown to be

$$\begin{aligned}W_i^{\mathrm{soft}} &= (\boldsymbol{d}_i^* X_i + W_{i-1} P)\{P + X_i^* X_i\}^{-1} \\ &= W_{i-1} + (\boldsymbol{d}_i^* - W_{i-1} X_i^*) X_i \{P + X_i^* X_i\}^{-1}\end{aligned} \tag{18}$$

or, equivalently,

$$W_i^{\mathrm{soft}} = W_{i-1} + (W_i^{\mathrm{LS}} - W_{i-1}) X_i^* X_i \{P + X_i^* X_i\}^{-1} \tag{19}$$

where W_i^{LS} is the LS solution of the block of data matrix X_i and desired vector $\boldsymbol{d}_i$. Again, if we consider two consecutive blocks with data matrices X_1 and X_2, then comparing (19) when $i = 2$ with the global LS solution given by (15b) suggests that $W^{\mathrm{soft}} = W^{\mathrm{LS}}$ if and only if $P = X_1^* X_1$, i.e., to transfer all information from the block X_1 to the block X_2, the matrix P must be chosen to be $X_1^* X_1$. This result is very interesting as it shows how the soft-constrained LS method can be modified in order to maximally transfer the data and, more importantly, generate a globally optimal block-LS solution. This result is generalized in the following developments.

Now, let us consider M consecutive (not necessarily equal size) blocks with data matrices $X_1, \cdots, X_M$ and desired vectors $\boldsymbol{d}_1, \cdots, \boldsymbol{d}_M$, respectively, and define the following modified block-LS cost functions that must be minimized.

$$\begin{aligned}\text{Minimize } \xi_M^{\mathrm{soft}} = {} & (W - W_{M-1}^{\mathrm{soft}}) \left(\sum_{i=r}^{M-1} \mu_i X_i^* X_i \right) \\ & \cdot (W - W_{M-1}^{\mathrm{soft}})^* \\ & + (\boldsymbol{d}_M^* - W X_M^*)(\boldsymbol{d}_M^* - W X_M^*)^*\end{aligned} \tag{20}$$

and

$$\begin{aligned}\text{Minimize } \xi_M^{\mathrm{LS}} = {} & (\boldsymbol{d}_M^* - W X_M^*)(\boldsymbol{d}_M^* - W X_M^*)^* \\ & + \sum_{i=r}^{M-1} \mu_i (\boldsymbol{d}_i^* - W X_i^*)(\boldsymbol{d}_i^* - W X_i^*)^*\end{aligned} \tag{21}$$

where W_{M-1}^{soft} in (20) minimizes $\xi_{M-1}^{\mathrm{soft}}$ in the block $M - 1$.

Note that the first cost function represents the cost function for a soft-constrained block-LS problem while the second one is the weighted block-LS problem. The next theorem shows that these problems can lead to the same solution if the μ_i's are properly chosen.

Theorem 3.1: Consider the LS minimization problems defined in (20) and (21):

a) The solutions W_M^{soft} and W_M^{LS}, of these are, respectively, given by

$$W_M^{\text{soft}} = W_{M-1}^{\text{soft}} + (\boldsymbol{d}_M^* - W_{M-1}^{\text{soft}} X_M^*) \cdot X_M \left\{ X_M^* X_M + \sum_{i=r}^{M-1} \mu_i X_i^* X_i \right\}^{-1} \tag{22}$$

$$W_M^{\text{LS}} = W_{M-1}^{\text{LS}} + \{(\boldsymbol{d}_M^* - W_{M-1}^{\text{LS}} X_M^*) X_M + (\mu_{M-1} - 1)(\boldsymbol{d}_{M-1}^* - W_{M-1}^{\text{LS}} X_{M-1}^*) X_{M-1}\} \cdot \left(X_M^* X_M + \sum_{i=r}^{M-1} \mu_i X_i^* X_i \right)^{-1} \tag{23}$$

with initial condition $W_1^{\text{soft}} = W_1^{\text{LS}} = \boldsymbol{d}_1^* X_1 (X_1^* X_1)^{-1}$.

b) The two solutions are equal if and only if $\mu_i = 1$ for $i = 1, \cdots, M-1$. Note that $\mu_M = 1$.

Proof: See Appendix B.

Note that in the modified soft-constrained block-LS case in (20), tracking can be controlled by varying the number of blocks processed and the block lengths. The case $r = 1, \mu_i = 1$ for $i = 1, \cdots, M$ yields the optimal solution for the whole data set up to the upper edge of the Mth block, whereas the case $\mu_i = 0$ for $i = 1, \cdots, M-1$ yields the block FTF algorithm of [3].

To implement the results of Theorem 3.1 using the original block FTF, the following algorithm for the modified block FTF algorithm is given.

B. Implementation of Modified Block FTF

To be consistent with the notation of Section II, the ith block with data matrix X_i will be denoted by $X_{Nl_ik_i}$, where k_i is the highest time index in X_i and l_i is the size of the block.

One can view the block FTF algorithm in [3, Table IV] as a method of computing $\boldsymbol{d}_{l_ik_i}^* X_{Nl_ik_i} \{X_{Nl_ik_i}^* X_{Nl_ik_i}\}^{-1}$, where $X_{Nl_ik_i}$ is any matrix of the form given in (2). Let $\{X_i\}_{i=r}^M$ be as in Theorem 3.1, and let

$$X_{NL_Mk_M} = [X_M^* \quad \sqrt{\mu_{M-1}} X_{M-1}^* \quad \sqrt{\mu_{M-2}} X_{M-2}^* \quad \cdots \quad \sqrt{\mu_r} X_r^*]^*$$

where $L_M = \Sigma_{i=r}^M l_i$, and l_i is the size of the ith block X_i. This $L_M \times N$ matrix $X_{NL_Mk_M}$ satisfies the shifting property for each M if only if $\mu_i = 1, i = 1, \cdots, M$, in which case, the block LS solution can be obtained recursively by applying the fast block FTF algorithm stated in [3, Table IV].

As shown in Theorem 3.1, the updating equation for the weight vector can be rewritten as

$$W_{NL_Mk_M} = W_{NL_{M-1}k_{M-1}} + \Delta W_{NL_Mk_M} \tag{24}$$

where

$$\Delta W_{NL_Mk_M} = (\boldsymbol{d}_{l_Mk_M}^* - W_{NL_{M-1}k_{M-1}} X_{Nl_Mk_M}^*) \cdot X_{Nl_Mk_M} \left(\sum_{i=r}^{M} X_{Nl_ik_i}^* X_{Nl_ik_i} \right)^{-1} \tag{25}$$

and $\boldsymbol{d}_{l_Mk_M}^*$ is the desired vector for the Mth block.

As in [3, Table II], the correlator quantities $p_{nl_ik_i}$ and $h_{nl_ik_i}$ for a block with data matrix $X_{nl_ik_i}$ are, in our case, defined as $p_{nl_ik_i} = \boldsymbol{x}_{l_ik_i-n}^* \boldsymbol{x}_{l_ik_i}$ and $h_{nl_ik_i} = \boldsymbol{x}_{l_ik_i-n}^* \overline{\boldsymbol{d}}_{l_ik_i}$, where $\overline{\boldsymbol{d}}_{l_ik_i} = (\boldsymbol{d}_{l_ik_i}^* - W_{nL_{i-1}k_{i-1}} X_{nl_ik_i}^*) X_{nl_ik_i}$, which for convenience can be written collectively in matrix form as

$$H_{Nl_ik_i} = [h_{0l_ik_i} \quad \cdots \quad h_{N-1l_ik_i}] = \overline{\boldsymbol{d}}_{l_ik_i}^* X_{Nl_ik_i}$$
$$\boldsymbol{P}_{Nl_ik_i} = [p_{0l_ik_i} \quad \cdots \quad p_{N-1l_ik_i}] = \boldsymbol{x}_{l_ik_i}^* X_{Nl_ik_i}. \tag{26}$$

Since an optimal solution is required for each block, the parameters $\alpha_{0l_ik_i}, \beta_{0l_ik_i}, \beta_{0l_ik_i-1}, \Delta_{0l_ik_i}, h_{0l_ik_i}$, and $p_{0l_ik_i}$ are to be computed and saved for updating the parameters in the next block. In essence, we are implementing the block FTF algorithm for the single block $X_{NL_Mk_M}$. However, this is done in steps so that we can have access to the weight vector at the edge of each block for the filtering process. The steps for implementing the general version of this algorithm when $\mu_i \neq 0$ are described in the next section.

It should be observed that this algorithm can be applied to process any number of consecutive blocks of data. In fact, it is possible to apply this algorithm for sample by sample filtering if desired. In this case, l_1 should be set to be equal to N to ensure optimality; then, $l_i = 1$ should be chosen for $i \geq 2$.

Information is passed from one block to another using correlational quantities accumulated in $\alpha_{0L_ik_i}, \beta_{0L_ik_i}, \beta_{0L_ik_i-1}, \Delta_{0L_ik_i}, H_{Nl_ik_i}$ and $\boldsymbol{P}_{NL_ik_i}$. Tracking in this algorithm can be controlled by varying the lengths of individual blocks or the number of consecutive blocks used. The main drawback of this algorithm is that it cannot be used to solve (20) or (21) with the μ_i's being different from 1 due to the fact that the data matrix involved does not satisfy the shifting property. However, generalization of this algorithm in this case can be established to accommodate the changes of the data matrix at the edges of individual blocks. This will be addressed in the next section.

IV. Generalization of Modified Block FTF Algorithm

Let us re-examine (20) and consider two blocks of data of lengths l_1 and l_2 (with $l_1 + l_2 = l$). We shall later discuss the transfer of information from one block to another. Now, this two-block problem can be stated as the determination of a vector W that minimizes

$$\xi_2 = \mu_1 (W - W_1) X_1^* X_1 (W - W_1)^* + (\boldsymbol{d}_2^* - W X_2^*)(\boldsymbol{d}_2^* - W X_2^*)^* \tag{27}$$

where

X_1 data matrix for the first block
X_2 data matrix for the second block
$\boldsymbol{d}_2$ desired vector for the second block

W_1 some initial weight vector that, in this case, can be the weight vector for the first block.

Using the notation of Section II, matrices X_1, X_2 and the desired vector $\boldsymbol{d}_2$ can be written as $X_2 = X_{Nl_2k}, X_1 = X_{Nl_1k-l_2}$, and $\boldsymbol{d}_2 = \boldsymbol{d}_{l_2k}$, where k is the highest time index in the second block. As shown in Theorem 3.1, the solution to the minimization problem in (27) is $W = W_1 + \Delta W_1$, where

$$\Delta W_1 = (\boldsymbol{d}_2^* - W_1 X_2^*) X_2 (\mu_1 X_1^* X_1 + X_2^* X_2)^{-1}.$$

The main purpose of this section is to develop a fast algorithm for computing ΔW_1. Using the above notations, the cost function (27) can be rewritten as

$$\begin{aligned}\xi_{Nl_1l_2k} = {} & \mu_1 (W_{Nl_1l_2k} - W_1) X_{Nl_1k-l_2}^* X_{Nl_1k-l_2} \\ & \cdot (W_{Nl_1l_2k} - W_1)^* + (\boldsymbol{d}_{l_2k} - W_{Nl_1l_2k} X_{Nl_2k})^* \\ & \cdot (\boldsymbol{d}_{l_2k} - W_{Nl_1l_2k} X_{Nl_2k}) \end{aligned} \tag{28}$$

where $\mu_1 \geq 0, d(r)$ and X_{Nr} are defined as in Section II. In the multichannel case, $x(r)$ is a $p \times 1$ vector, and $d(r)$ is a $q \times 1$ vector, which makes $W_{nl_1l_2k}$ a $q \times np$ matrix. The derivation of the multichannel case is similar to that of the single-channel; therefore, the derivation here is stated for the multichannel case.

Before we proceed, we introduce the following notations, which are convenient for geometric interpretation of the block LS problem (27). The geometric interpretation of the block LS problem (27) can be established within the vector space $\mathcal{C}^l$, where $\mathcal{C}$ is the field of complex numbers, by defining the following vectors (or collection of vectors in the multichannel case):

$$\begin{aligned}\bar{\boldsymbol{d}}_{l_1l_2k} &= (\boldsymbol{d}_{l_2k}^* - W_{Nl_1k-l_2} X_{Nl_2k}^*), \mathbf{0})^*, \\ \boldsymbol{x}_{l_1l_2k} &= [x(k) \quad \cdots \quad x(k-l_1+1) \quad \sqrt{\mu_1} x(k-l_1) \\ &\qquad \cdots \quad \sqrt{\mu_1} x(k-l+1)]^*\end{aligned}$$

and the $l \times np$ data matrix of the two blocks is defined as (29), which appears at the bottom of the page. Clearly, $X_{nl_1l_2k} = \Lambda_{l_1,l_2} X_{nlk}$ and $\boldsymbol{x}_{l_1l_2k} = \Lambda_{l_1,l_2} \boldsymbol{x}_{lk}$, where Λ_{l_1,l_2} is a diagonal matrix defined by

$$\Lambda_{l_1,l_2} = \begin{bmatrix} I_{l_1} & 0 \\ 0 & \sqrt{\mu_1} I_{l_2} \end{bmatrix}.$$

Therefore, ΔW_1 or $\Delta W_{Nl_1l_2k}$ is

$$\Delta W_{Nl_1l_2k} = \bar{\boldsymbol{d}}_{l_1l_2k}^* K_{Nl_1l_2k} \tag{30}$$

where

$$K_{Nl_1l_2k} = X_{Nl_1l_2k} \{X_{Nl_1l_2k}^* X_{Nl_1l_2k}\}^{\#},$$

and

$$P_{Nl_1l_2k} = X_{Nl_1l_2k} \{X_{Nl_1l_2k}^* X_{Nl_1l_2k}\}^{\#} X_{Nl_1l_2k}^*$$

are as defined before.

Note that the updating rules in (5) become

$$\begin{bmatrix} K_{nl_1l_2-1k} \\ 0 \end{bmatrix} = K_{nl_1l_2k} + P_{nl_1l_2k}^{\perp} e_l (e_l^* P_{nl_1l_2k}^{\perp} e_l)^{-1} \cdot [-e_l^* K_{nl_1l_2k}] \tag{31a}$$

$$\begin{bmatrix} 0 \\ K_{nl_1-1l_2k-1} \end{bmatrix} = K_{nl_1l_2k} + P_{nl_1l_2k}^{\perp} e_1 (e_1^* P_{nl_1l_2k}^{\perp} e_1)^{-1} \cdot [-e_1^* K_{nl_1l_2k}] \tag{31b}$$

and

$$P_{nl_1l_2k;e_1}^{\perp} = \begin{bmatrix} 0 & \mathbf{0}^* \\ \mathbf{0} & P_{nl_1-1l_2k-1}^{\perp} \end{bmatrix} \quad \text{and} \quad P_{nl_1l_2k;e_l}^{\perp} = \begin{bmatrix} 0 & P_{nl_1l_2-1k-1}^{\perp} \\ \mathbf{0} & \mathbf{0}^* \end{bmatrix}. \tag{32}$$

In order to determine a recursive solution for $\Delta W_{Nl_1l_2k}$, relationships between $K_{nl_1-1l_2k}$ and $K_{nl_1l_2-1k}$ and $P_{nl_1-1l_2k}$ and $P_{nl_1l_2-1k}$ are required, which can be established as follows. Let $R_{nl_1l_2k}$ be defined by $R_{nl_1l_2k} = X_{nl_1l_2k}^* X_{nl_1l_2k}$. Then

$$\begin{aligned} R_{nl_1l_2k} = {} & \sum_{i=0}^{l_1-1} X_{nk-i} X_{nk-i}^* \\ & + \mu_1 \sum_{i=0}^{l_2-1} X_{nk-l_1-i} X_{nk-l_1-i}^* \end{aligned}$$

and

$$R_{nl_1l_2-1k} = R_{nl_1-1l_2k} + (1-\mu_1) X_{nk-l_1+1} X_{nk-l_1+1}^*. \tag{33}$$

Applying the matrix inversion lemma to (33) yields

$$R_{nl_1l_2-1k}^{-1} = R_{nl_1-1l_2k}^{-1} - \frac{R_{nl_1-1l_2k}^{-1} X_{nk-l_1+1} X_{nk-l_1+1}^* R_{nl_1-1l_2k}^{-1}}{\frac{1}{1-\mu_1} + X_{nk-l_1+1}^* R_{nl_1-1l_2k}^{-1} X_{nk-l_1+1}}. \tag{34}$$

Since

$$X_{nk-l_1+1}^* = \frac{1}{\sqrt{\mu_1}} e_{l_1-1}^* X_{nl_1-1l_2k}$$

$$X_{nl_1l_2k} = [\boldsymbol{x}_{l_1l_2k} \quad \cdots \quad \boldsymbol{x}_{l_1l_2k-n+1}] = \begin{bmatrix} x^*(k) & \cdots & x^*(k-n+1) \\ x^*(k-1) & \cdots & x^*(k-n) \\ \cdots & \cdots & \cdots \\ x^*(k-l_1+1) & \cdots & x^*(k-n-l_1+2) \\ \sqrt{\mu_1} x^*(k-l_1) & \cdots & \sqrt{\mu_1} x^*(k-n-l_1+1) \\ \sqrt{\mu_1} x^*(k-l_1-1) & \cdots & \sqrt{\mu_1} x^*(k-n-l_1) \\ \cdots & \cdots & \cdots \\ \sqrt{\mu_1} x^*(k-l+1) & \cdots & \sqrt{\mu_1} x^*(k-n-l+2) \end{bmatrix}. \tag{29}$$

it follows that

$$X^*_{nk-l_1+1}R^{-1}_{nl_1-1l_2k}X_{nk-l_1+1} = \frac{1}{\mu_1}(1 - f_{nl_1-1l_2k})$$

where $f_{nl_1l_2k} = e^*_{l_1}P^{\perp}_{nl_1l_2k}e_{l_1}$, and

$$R^{-1}_{nl_1-1l_2k}X_{nk-l_1+1}X^*_{nk-l_1+1}R^{-1}_{nl_1-1l_2k} = \frac{1}{\mu_1}K^*_{nl_1-1l_2k}e_{l_1-1}e^*_{l_1-1}K_{nl_1-1l_2k}.$$

It can also easily be verified that $X_{nl_1l_2-1k} = FX_{nl_1-1l_2k}$, where F is an identity matrix, except it is $1/\sqrt{\mu_1}$ in the $l_1 - 1 \times l_1 - 1$ position or, equivalently, $F = I_{l-1} + (1 - \sqrt{\mu_1}/\sqrt{\mu_1})e_{l_1-1}e^*_{l_1-1}$. Premultiplying both sides of (34) by $X_{nl_1l_2-1k}$ and simplifying the results gives

$$K_{nl_1l_2-1k} = F\left\{K_{nl_1-1l_2k} - \frac{P_{nl_1-1l_2k}e_{l_1-1}e^*_{l_1-1}K_{nl_1-1l_2k}}{\frac{1}{1-\mu_1} - f_{nl_1-1l_2k}}\right\}. \quad (35)$$

Postmultiplying both sides of the last equation by $X^*_{nl_1l_2-1k}$ yields

$$P_{nl_1l_2-1k} = F\left\{P_{nl_1-1l_2k} - \frac{P_{nl_1-1l_2k}e_{l_1-1}e^*_{l_1-1}P_{nl_1-1l_2k}}{\frac{1}{1-\mu_1} - f_{nl_1-1l_2k}}\right\}F. \quad (36)$$

Equations (35) and (36) are all that is required to derive equations that relate quantities with index $nl_1l_2 - 1k$ to those with index $nl_1 - 1l_2k$, in which case, we typically pre or postmultiply (35) and (36) by vectors of the form $u = Fv$. This means that the quantities between brackets including $K_{nl_1-1l_2k}$ and $P_{nl_1-1l_2k}$ in the right-hand side of these equations are multiplied by F^2v. It can easily be verified that $F^2 = I + (1 - \mu_1)/\mu_1 e_{l_1-1}e^*_{l_1-1}$. By defining a new quantity $G_{nl_1l_2k} = -e^*_{l_1}K_{nl_1l_2k}$ and premultiplying (35) by $u^*_1 = v^*_1F$, we obtain

$$u^*_1K_{nl_1l_2-1k} = v^*_1K_{nl_1-1l_2k} + \frac{v^*_1P^{\perp}_{nl_1-1l_2k}e_{l_1-1}}{\frac{1}{1-\mu_1} - f_{nl_1l_2-1k}}G_{nl_1-1l_2k}. \quad (37)$$

Similarly, premultiplying (36) by $u^*_i = v^*_iF$ and postmultiplying by $u_i = Fv_i$ for $i = 1, 2$, respectively, yields (38a), which appears at the bottom of the page, or, equivalently, (38b), which is also at the bottom of the page.

To develop a block FTF-like algorithm, a recursion for $W_{n+1l_1l_2k}$ in terms of $W_{nl_1l_2k}$ has to be determined. This recursion is the direct result of substitutions for u, v, X, and z in (37) and (38). By making the substitutions of Table I in (31a) and (31b), we obtain eight transversal-filter recursions as follows:

$$A_{n+1l_1l_2k} = [A_{nl_1l_2k} \quad 0] - \Delta_{nl_1l_2k}\beta^{-1}_{nl_1l_2k-1} \cdot [0 \quad B_{nl_1l_2k-1}] \quad (39a)$$

$$B_{n+1l_1l_2k} = [0 \quad B_{nl_1l_2k-1}] - \Delta^*_{nl_1l_2k}\alpha^{-1}_{nl_1l_2k} \cdot [A_{nl_1l_2k} \quad 0] \quad (39b)$$

$$C_{n+1l_1l_2k} = [C_{nl_1l_2k} \quad 0] - r^*_{nl_1l_2k}\beta^{-1}_{nl_1l_2k}B_{nl_1l_2k} \quad (39c)$$

$$D_{n+1l_1l_2k-1} = [D_{nl_1l_2k-1} \quad 0] - b^*_{nl_1l_2k-1} \cdot \beta^{-1}_{nl_1l_2k-1}B_{nl_1l_2k-1} \quad (39d)$$

TABLE I
SUBSTITUTIONS FOR GENERALIZED BLOCK FTF DERIVATION

#	u	X	z
49a	$x_{l_1l_2k}$	$X_{nl_1l_2k-1}$	$x_{l_1l_2k-n-1}$
49b	$x_{l_1l_2k-n-1}$	$X_{nl_1l_2k-1}$	$x_{l_1l_2k}$
49c	e_1	$X_{nl_1l_2k}$	$x_{l_1l_2k-n}$
49d	e_l	$X_{nl_1l_2k-1}$	$x_{l_1l_2k-n-1}$
49e	e_{l_1}	$X_{nl_1l_2k}$	$x_{l_1l_2k-n}$
49f	$x_{l_1l_2k-n-1}$	$X_{n+1l_1l_2k}$	e_1
49g	$x_{l_1l_2k-n-2}$	$X_{n+1l_1l_2k-1}$	e_l
49h	$\bar{\boldsymbol{d}}_{l_1l_2k}$	$X_{n+1l_1l_2k}$	$\boldsymbol{x}_{l_1l_2k-n-1}$
49i	$x_{l_1l_2k-n-1}$	$X_{n+1l_1l_2k-1}$	e_l

$$u^*_1P_{nl_1l_2-1k}u_2 = v^*_1\left\{P_{nl_1-1l_2k} + \frac{\frac{1-(1-\mu_1)f_{nl_1-1l_2k}}{\mu_1}e_{l_1-1}e^*_{l_1-1} - P^{\perp}_{nl_1-1l_2k}e_{l_1-1}e^*_{l_1-1}P^{\perp}_{nl_1-1l_2k}}{\frac{1}{1-\mu_1} - f_{nl_1-1l_2k}}\right\}v_2 \quad (38a)$$

$$u^*_1P^{\perp}_{nl_1l_2-1k}u_2 = v^*_1P^{\perp}_{nl_1-1l_2k}v_2 + u^*_1u_2 - v^*_1v_2 - \frac{\frac{1-(1-\mu_1)f_{nl_1-1l_2k}}{\mu_1}v^*_1e_{l_1-1}e^*_{l_1-1}v_2 - v^*_1P^{\perp}_{nl_1-1l_2k}e_{l_1-1}e^*_{l_1-1}P^{\perp}_{nl_1-1l_2k}v_2}{\frac{1}{1-\mu_1} - f_{nl_1-1l_2k}} \quad (38b)$$

TABLE II
BLOCK FTF PARAMETERS

Input Definitions	Filter Definitions
$p_{nl_1l_2k} = \boldsymbol{x}^*_{l_1l_2k-n}\boldsymbol{d}_{l_1l_2k}$	$A_{nl_1l_2k} = [I \; -\boldsymbol{x}^*_{l_1l_2k}K_{nl_1l_2k-1}]$
$h_{nl_1l_2k} = \boldsymbol{x}^*_{l_1l_2k-n}\boldsymbol{x}_{l_1l_2k}$	$B_{nl_1l_2k} = [-\boldsymbol{x}^*_{l_1l_2k-n}K_{nl_1l_2k} \; I]$
$X_{nk} = [x(k)^* \, , \cdots x^*(k-n+1)]^*$	$C_{nl_1l_2k} = -e^*_1 K_{nl_1l_2k}$
	$D_{nl_1l_2k} = -e^*_l K_{nl_1l_2k}$
	$G_{nl_1l_2k} = -e^*_{l_1} K_{nl_1l_2k}$
	$\Delta W_{nl_1l_2k} = \overline{\boldsymbol{d}}^*_{l_1l_2k} K_{nl_1l_2k}$

$$G_{n+1l_1-1l_2k-1} = [G_{nl_1-1l_2k-1} \quad 0] - h_{nl_1-1l_2k-1} \cdot \beta^{-1}_{nl_1-1l_2k-1} B_{nl_1-1l_2k-1} \tag{39e}$$

$$B_{n+1l_1-1l_2k-1} = B_{n+1l_1l_2k} - r_{n+1l_1l_2k}\gamma^{-1}_{n+1l_1l_2k} \cdot [C_{n+1l_1l_2k} \quad 0] \tag{39f}$$

$$B_{n+1l_1l_2k-1} = B_{n+1l_1-1l_2k-1} + b^S_{n+1l_1l_2-1k-1} \cdot [D_{n+1l_1l_2k-1} \quad 0] \tag{39g}$$

$$\Delta W_{n+2l_1l_2k} = [\Delta W_{n+1l_1l_2k} \quad 0] + \omega_{n+1l_1l_2k} \cdot \beta^{-1}_{n+1l_1l_2k} B_{n+1l_1l_2k} \tag{39h}$$

The ninth substitution of Table I into (37) yields the ninth transversal filter

$$B_{n+1l_1l_2-1k-1} = B_{n+1l_1-1l_2k-1} + \frac{h_{n+1l_1-1l_2k-1}}{\frac{1}{1-\mu_1} - f_{n+1l_1-1l_2k-1}} [G_{n+1l_1-1l_2k-1} \quad 0] \tag{39i}$$

The following scalar parameters can be updated making the substitutions of Table III in (4) with $u = v$

$$\begin{aligned}
\alpha_{n+1l_1l_2k} &= \alpha_{nl_1l_2k} - \Delta_{nl_1l_2k}\beta^{-1}_{nl_1l_2k-1}\Delta^*_{nl_1l_2k} \\
\beta_{n+1l_1l_2k} &= \beta_{nl_1l_2k-1} - \Delta^*_{nl_1l_2k}\alpha^{-1}_{nl_1l_2k}\Delta_{nl_1l_2k} \\
\gamma_{n+1l_1l_2k} &= \gamma_{nl_1l_2k} - r^*_{nl_1l_2k}\beta^{-1}_{nl_1l_2k}r_{nl_1l_2k} \\
\delta_{n+1l_1l_2k-1} &= \delta_{nl_1l_2k-1} - b^*_{nl_1l_2k-1}\beta^{-1}_{nl_1l_2k-1}b_{nl_1l_2k-1} \\
\beta_{n+1l_1-1l_2k-1} &= \beta_{n+1l_1l_2k} - r_{n+1l_1l_2k}\gamma^{-1}_{n+1l_1l_2k}r^*_{n+1l_1l_2k} \\
\beta_{n+1l_1l_2k-1} &= \beta_{n+1l_1l_2-1k-1} + b^S_{n+1l_1-1l_2k-1}b^*_{n+1l_1l_2k-1} \\
r_{n+1l_1l_2k-1} &= B_{n+1l_1l_2k-1}X_{n+2k-1}
\end{aligned}$$

The parameter $\beta_{n+1l_1l_2-1k-1}$ can be updated using $u_1 = u_2 = \boldsymbol{x}_{l_1-1l_2k-n-1}$ in (38b). Thus

$$\begin{aligned}
\beta_{n+1l_1l_2-1k-1} = {} & \beta_{n+1l_1-1l_2-1k-1} \\
& + (1-\mu_1)x(k-n-l_1-1) \\
& \cdot x^*(k-n-l_1-1) \\
& - \frac{(\mu_1-1)^2}{\mu_1}x(k-l_1-n) \\
& \cdot x^*(k-l_1-n) \\
& + \frac{h_{n+1l_1-1l_2k-1}h^*_{n+1l_1-1l_2k-1}}{\frac{1}{1-\mu_1} - f_{n+1l_1-1l_2-1k-1}}.
\end{aligned}$$

Using the right substitution, we obtain

$$\begin{aligned}
r_{n+1l_1l_2k} &= r_{nl_1l_2k-1} - \Delta^*_{nl_1l_2k}\alpha^{-1}_{nl_1l_2k}e_{nl_1l_2k} \\
e_{n+1l_1l_2k} &= e_{nl_1l_2k} - \Delta_{nl_1l_2k}\beta^{-1}_{nl_1l_2k-1}r_{nl_1l_2k-1} \\
\epsilon_{n+2l_1l_2k} &= \epsilon_{n+1l_1l_2k} - \omega^*_{nl_1l_2k}\beta^{-1}_{n+1l_1l_2k}r^*_{n+1l_1l_2k}.
\end{aligned}$$

All quantities used in the recursive solution of (27) are listed with their definitions in Table III. This concludes the derivation of the algorithm. The relevant equations of the modified block FTF algorithm are shown in Table IV. The sample lags $p_{nl_1l_2k}$ and $h_{nl_1l_2k}$ are defined by

$$\begin{aligned}
p_{nl_1l_2k} &= \boldsymbol{x}^*_{l_1l_2k}\boldsymbol{x}_{l_1l_2k-n} = \boldsymbol{x}^*_{l_1+l_2k}\Lambda_{l_1,l_2}\boldsymbol{x}_{l_1+l_2k-n} \\
h_{nl_1l_2k} &= \overline{\boldsymbol{d}}^*_{l_1l_2k}\boldsymbol{x}_{l_1l_2k-n} = \overline{\boldsymbol{d}}^*_{l_1+l_2k}\Lambda_{l_1,l_2}\boldsymbol{x}_{l_1+l_2k-n} \\
&= \overline{\boldsymbol{d}}^*_{l_2k}\boldsymbol{x}^*_{l_2k-n}
\end{aligned}$$

where Λ_{l_1,l_2} is defined before. The computations of $h_{nl_1l_2k}$ and $p_{nl_1l_2k}$ require $O(N)$ operations per sample. Table III presents the definition and computation of all parameters required in the algorithm.

It should be mentioned here that this algorithm reduces to that in [3, Table IV] if $\mu_1 = 1$ since then, $B_{n+1l_1-1l_2k-1} = B_{n+1l_1l_2-1k-1}, G_{nl_1l_2k-1} = 1$, and $h_{n+1l_1l_2-1k-1} = r_{n+1l_1l_2-1k-1}$. This block FTF method is numerically stable since the finite-precision error does not accumulate from block to block as the only quantities carried from the previous block is $W_{nl_1l_2k}$, and its effect decays as more blocks are processed.

In the multichannel case, $\alpha_{nl_1l_2k}$ and $\beta_{nl_1l_2k}$ are $p \times p$ matrices, and their inversion is left to the choice of the reader.

A. Initialization

The algorithm is started by computing the weights of the first block using [3, Table IV], in which certain parameters must be initialized. Then, depending on how much influence the previous block should have on the tracking, μ_1 is chosen, and the computed parameters of the previous block will serve as an initialization for the the current block parameter update applying Table IV. These initial values are computed directly from their definitions in Table III as follows:

$$\begin{aligned}
A_{0l_1l_2k} &= B_{0l_1l_2k} = 1, \quad \alpha_{0l_1l_2k} = p_{0l_1l_2k} = \boldsymbol{x}^*_{l_1l_2k}\boldsymbol{x}_{l_1l_2k}, \\
\beta_{0l_1l_2k-1} &= p_{0l_1l_2k-1} = \boldsymbol{x}^*_{l_1l_2k-1}\boldsymbol{x}_{l_1l_2k-1}, \\
b_{0l_1l_2k-1} &= x(k-l+1), \\
\beta_{0l_1l_2k} &= \alpha_{l_1l_2k}, \quad e_{0l_1l_2k-1} = r_{0l_1l_2k} = x(k), \\
r_{0l_1l_2k-1} &= x(k-1), \quad \gamma_{0l_1l_2k} = \delta_{0l_1l_2k-1} = 1, \\
\Delta_{0l_1l_2k} &= \boldsymbol{x}^*_{l_1l_2k}\boldsymbol{x}_{l_1l_2k-1}, \quad \Delta W_{1l_1l_2k} = h_{0l_1l_2k}p^{-1}_{0l_1l_2k}.
\end{aligned} \tag{40}$$

The quantities $C_{nl_1l_2k}, D_{nl_1l_2k}$, and $G_{nl_1l_2k}$ have meaning only for $n \geq 1$. The algorithm for the general case of $\mu_i \neq 1$ is given below.

TABLE III
GENERALIZED BLOCK FTF PARAMETERS

Input	Definition	Computation
$e_{nl_1l_2k}$	$\boldsymbol{x}^*_{l_1l_2k}P^{\perp}_{nl_1l_2k-1}e_1$	$A_{nl_1l_2k}X_{n+1k}$
$r_{nl_1l_2k}$	$\boldsymbol{x}^*_{l_1l_2k-n}P^{\perp}_{nl_1l_2k}e_1$	$B_{nl_1l_2k}X_{n+1k}$
$\gamma_{nl_1l_2k}$	$e_1^*P^{\perp}_{nl_1l_2k}e_1$	$1+C_{nl_1l_2k}X_{nk}$
$\delta_{nl_1l_2k}$	$e_l^*P^{\perp}_{nl_1l_2k}e_l$	$1+\sqrt{\mu_1}D_{nl_1l_2k}X_{nk-l+1}$
$\epsilon_{nl_1l_2k}$	$\boldsymbol{d}^*_{l_1l_2k}P^{\perp}_{nl_1l_2k}e_1$	$d(k)-W_{nl_1l_2k}X_{nk}$
$b_{nl_1l_2k}$	$\boldsymbol{x}^*_{l_1l_2k-n}P^{\perp}_{nl_1l_2k}e_l$	$\sqrt{\mu_1}B_{nl_1l_2k}X_{n+1k-l+1}$
b^S_{nl-1k}	$B_{nl-1k}X_{n+1k-l+1}$	$B_{nl-1k}X_{n+1k-l+1}$
$\alpha_{nl_1l_2k}$	$\boldsymbol{x}^*_{l_1l_2k}P^{\perp}_{nl_1l_2k-1}\boldsymbol{x}_{l_1l_2k}$	$A_{nl_1l_2k}[p^*_{0l_1l_2k}, \cdots, p^*_{nl_1l_2k}]^*$
$\beta_{nl_1l_2k}$	$\boldsymbol{x}^*_{l_1l_2k-n}P^{\perp}_{nl_1l_2k}\boldsymbol{x}_{l_1l_2k-n}$	$B_{nl_1l_2k}[p_{nl_1l_2k}, \cdots, p_{0l_1l_2k}]^*$
$\Delta_{nl_1l_2k}$	$\boldsymbol{x}^*_{l_1l_2k}P^{\perp}_{nl_1l_2k-1}\boldsymbol{x}_{l_1l_2k-n-1}$	$A_{nl_1l_2k}[p_{n+1l_1l_2k}, \cdots, p_{1l_1l_2k-n}]^*$
$\omega_{nl_1l_2k}$	$\boldsymbol{d}^*_{l_1l_2k}P^{\perp}_{nl_1l_2k}\boldsymbol{x}_{l_1l_2k-n}$	$[h^*_{0l_1l_2k}, \cdots, h^*_{nl_1l_2k}]B^*_{nl_1l_2k}$
$\xi_{nl_1l_2k}$	$\boldsymbol{d}^*_{l_1l_2k}P^{\perp}_{nl_1l_2k}\boldsymbol{d}_{l_1l_2k}$	$\xi_{0l_1l_2k}-W_{nl_1l_2k}[h^*_{0l_1l_2k}, \cdots, h^*_{n-1l_1l_2k}]^*$
$f_{nl_1l_2k}$	$e^*_{l_1}P^{\perp}_{nl_1l_2k}e_{l_1}$	$1+G_{nl_1l_2k}X_{n+1k-l_1}$
$h_{nl_1l_2k}$	$\boldsymbol{x}^*_{l_1l_2k-n}P^{\perp}_{nl_1l_2k}e_{l_1}$	$B_{n+1l_1l_2k}X_{n+2k-l_1-n}$

Algorithm

Step 1: Initialization of Parameters:

$i=1$

l_i: length of the i-th block

L_i: length of the block

$$[X_i^* \quad \sqrt{\mu_{i-1}}X^*_{i-1} \quad \sqrt{\mu_{i-2}}X^*_{i-2} \quad \cdots \quad \sqrt{\mu_1}X^*_1]^*$$

k_i: highest time index in the i-th block

$L_i=l_i, \quad W_{NL_0k_0}=0$

$A_{0l_ik_i}=B_{0l_ik_i}=B_{0l_ik_i-1}=1,$

$$\alpha_{0l_ik_i}=\beta_{0l_ik_i}=\boldsymbol{x}^*_{l_ik_i}\boldsymbol{x}_{l_ik_i}$$

$\beta_{0l_ik_i-1}=\boldsymbol{x}^*_{l_ik_i-1}\boldsymbol{x}_{l_ik_i-1}, \quad \gamma_{0l_ik_i}=\delta_{0l_ik_i-1}=1$

$H_{Nl_ik_i}=\overline{\boldsymbol{d}}^*_{l_ik_i}X_{Nl_ik_i}, \quad \boldsymbol{P}_{Nl_ik_i}=\boldsymbol{x}^*_{l_ik_i}X_{Nl_ik_i}$

$e_{0l_ik_i}=r_{0l_ik_i}=x(k_i), \quad r_{0l_ik_i-1}=x(k_i-1),$

$b_{0l_ik_i-1}=x(k_i-l_i)$

$\Delta_{0l_ik_i}=\boldsymbol{x}^*_{l_ik_i}\boldsymbol{x}_{l_ik_i-1}, \quad \Delta W_{1l_ik_i}=h_{0l_ik_i}p^{-1}_{0l_ik_i}$

Step 2: Order Updating and Filtering in Each Block:

- Apply steps 1–20 of Table IV for $n=1,\cdots,N$ to determine $\Delta W_{NL_ik_i}$ and update the weight using $W_{NL_ik_i}=W_{NL_{i-1}k_{i-1}}+\Delta W_{NL_ik_i}$.
- Filter to obtain the output and error signals in the ith block using $W_{NL_ik_i}$.

Step 3: Parameter Updating:

$$i=i+1, \quad L_i=L_{i-1}+l_i$$

$$A_{0L_ik_i}=B_{0L_ik_i}=B_{0L_ik_i-1}=1$$

$$\overline{\boldsymbol{d}}^*_{l_ik_i}=\boldsymbol{d}^*_{l_ik_i}-W_{NL_{i-1}k_{i-1}}X^*_{Nl_ik_i},$$

$$H_{Nl_ik_i}=\overline{\boldsymbol{d}}^*_{l_ik_i}X_{Nl_ik_i}$$

$$\boldsymbol{P}_{NL_ik_i}=\boldsymbol{P}_{NL_{i-1}k_{i-1}}+(\mu_{i-1}-1)\boldsymbol{P}_{Nl_{i-1}k_{i-1}}+\boldsymbol{x}^*_{l_ik_i}X_{Nl_ik_i}$$

$$\alpha_{0L_ik_i}=\alpha_{0L_{i-1}k_{i-1}}+(\mu_{i-1}-1)\alpha_{0l_{i-1}k_{i-1}}+\boldsymbol{x}^*_{l_ik_i}\boldsymbol{x}_{l_ik_i}$$

$$\beta_{0L_ik_i}=\beta_{0L_{i-1}k_{i-1}}+(\mu_{i-1}-1)\beta_{0l_{i-1}k_{i-1}}+\boldsymbol{x}^*_{l_ik_i}\boldsymbol{x}_{l_ik_i}$$

$$\beta_{0L_ik_i-1}=\beta_{0L_{i-1}k_{i-1}-1}+(\mu_{i-1}-1)\beta_{0l_{i-1}k_{i-1}-1}+\boldsymbol{x}^*_{l_ik_i-1}\boldsymbol{x}_{l_ik_i-1}$$

$$\beta_{0L_ik_i}=\alpha_{0L_ik_i}=p_{0L_ik_i}, \quad e_{0L_ik_i}=r_{0L_ik_i}=x(k_i)$$

$$r_{0L_ik_i-1}=x(k_i-1), \quad \gamma_{0L_ik_i}=\delta_{0L_ik_i-1}=1$$

$$b_{0L_ik_i-1}=x(k_i-L_i),$$

$$\Delta_{0L_ik_i}=\Delta_{0L_{i-1}k_{i-1}}+(\mu_{i-1}-1)\Delta_{0l_{i-1}k_{i-1}}+\boldsymbol{x}^*_{l_ik_i}\boldsymbol{x}_{l_ik_i-1}$$

$$\Delta W_{1L_ik_i}=h_{0l_ik_i}p^{-1}_{0L_ik_i}$$

Go to Step 2.

V. TEST RESULTS

The modified block FTF algorithm in Sections III and IV developed in this paper was examined on the actual acoustic backscatter data. This data set consisted of an incident signal and the backscattered data for a submerged cylindrical elastic target and an irregularly shaped cement chunk of similar size and for different aspect angles. The incident signal was wideband linear frequency modulated (FM). The adaptive block FTF system had 32 tap weights. The reference input to the adaptive system was the incident waveform, whereas the desired signal was the backscattered signal. The goal was to produce an accurate estimate of the specular component at the output of the adaptive filter so that the resonant return, which is typically present for elastic targets, can be separated at the error signal. There are two principal assumptions behind the development of this scheme: 1) The specular part is more correlated with the incident part, and 2) there is a time lag between the onset of the specular and that of the resonant part. These characteristics, which are dependent on

TABLE IV
GENERALIZED BLOCK FTF ALGORITHM

Steps	Updating Equations
1	$\Delta_{nl_1l_2k} = [p^*_{1l_1l_2k}, \quad \cdots, \quad p^*_{n+1l_1l_2k}]B^*_{nl_1l_2k-1}$
2	$C_{n+1l_1l_2k} = [C_{nl_1l_2k} \quad 0] - r^*_{nl_1l_2k}\beta^{-1}_{nl_1l_2k}B_{nl_1l_2k}$
3	$B_{n+1l_1l_2k} = [0 \quad B_{nl_1l_2k-1}] - \Delta^*_{nl_1l_2k}\alpha^{-1}_{nl_1l_2k}[A_{nl_1l_2k} \quad 0]$
4	$\beta_{n+1l_1l_2k} = \beta_{nl_1l_2k-1} - \Delta^*_{nl_1l_2k}\alpha^{-1}_{nl_1l_2k}\Delta_{nl_1l_2k}$
5	$\omega_{n+1l_1l_2k} = [h^*_{0l_1l_2k}, \quad \cdots, \quad h^*_{n+1l_1l_2k}]B^*_{n+1l_1l_2k}$
6	$\Delta W_{n+2l_1l_2k} = [\Delta W_{n+1l_1l_2k} \quad 0] + \omega_{n+1l_1l_2k}\beta^{-1}_{n+1l_1l_2k}B_{n+1l_1l_2k}$
7	$A_{n+1l_1l_2k} = [A_{nl_1l_2k} \quad 0] - \Delta_{nl_1l_2k}\beta^{-1}_{nl_1l_2k-1}[0 \quad B_{nl_1l_2k-1}]$
8	$D_{n+1l_1l_2k-1} = [D_{nl_1l_2k-1} \quad 0] - b^*_{nl_1l_2k-1}\beta^{-1}_{nl_1l_2k-1}B_{nl_1l_2k-1}$
9	$\gamma_{n+1l_1l_2k} = \gamma_{nl_1l_2k} - r^*_{nl_1l_2k}\beta^{-1}_{nl_1l_2k}r_{nl_1l_2k}$
10	$\delta_{n+1l_1l_2k-1} = \delta_{nl_1l_2k-1} - b^*_{nl_1l_2k-1}\beta^{-1}_{nl_1l_2k-1}b_{nl_1l_2k-1}$
11	$r_{n+1l_1l_2k} = r_{nl_1l_2k-1} - \Delta^*_{nl_1l_2k}\alpha^{-1}_{nl_1l_2k}e_{nl_1l_2k}$
12	$e_{n+1l_1l_2k} = e_{nl_1l_2k} - \Delta_{nl_1l_2k}\beta^{-1}_{nl_1l_2k-1}r_{nl_1l_2k-1}$
13	$B_{n+1l_1-1l_2k-1} = B_{n+1l_1l_2k} - r_{n+1l_1l_2k}\gamma^{-1}_{n+1l_1l_2k}[C_{n+1l_1l_2k} \quad 0]$
14	$\alpha_{n+1l_1l_2k} = \alpha_{nl_1l_2k} - \Delta_{nl_1l_2k}\beta^{-1}_{nl_1l_2k-1}\Delta^*_{nl_1l_2k}$
15	$\beta_{n+1l_1-1l_2k-1} = \beta_{n+1l_1l_2k} - r_{n+1l_1l_2k}\gamma^{-1}_{n+1l_1l_2k}r^*_{n+1l_1l_2k}$
16	$h_{n+1l_1l_2-1k-1} = B_{n+1l_1l_2-1k-1}X_{n+2k-l_1-n+1}$
17	$G_{n+1l_1-1l_2k-1} = [G_{nl_1-1l_21k-1} \quad 0] + h_{nl_1-1l_2k-1}\beta^{-1}_{nl_1-1l_2k-1}B_{nl_1-1l_2k-1}$
18	$h_{n+1l_1-1l_2k-1} = 1 + G_{n+1l_1-1l_2k-1}X_{n+1k-l_1+1}$
19	$B_{n+1l_1l_2-1k-1} = B_{n+1l_1-1l_2k-1} + \dfrac{h_{n+1l_1-1l_2k-1}}{\dfrac{1}{1-\mu_1} - f_{n+1l_1-1l_2k-1}}[G_{n+1l_1-1l_2k-1} \quad 0]$
20	$b^S_{n+1l_1-1l_2k-1} = B_{n+1l_1-1l_2k-1}X_{n+2k-l}$
21	$b_{n+1l_1l_2k-1} = b^S_{n+1l_1-1l_2k-1}\delta_{n+1l_1l_2k-1}$
22	$B_{n+1l_1l_2k-1} = B_{n+1l_1l_2-1k-1} + b^S_{n+1l_1l_2-1k-1}[D_{n+1l_1l_2k-1} \quad 0]$
23	$\beta_{n+1l_1l_2-1k-1} = \beta_{n+1l_1-1l_2-1k-1} + (1-\mu_1)x(k-n-l_1-1)x(k-n-l_1-1)^* - \dfrac{(\mu_1-1)^2}{\mu_1}x(k-l_1-n)x(k-l_1-n)^* + \dfrac{h_{n+1l_1-1l_2k-1}h^*_{n+1l_1-1l_2k-1}}{\dfrac{1}{1-\mu_1} - f_{n+1l_1-1l_2-1k-1}}$
24	$\beta_{n+1l_1l_2k-1} = \beta_{n+1l_1l_2-1k-1} + b^S_{n+1l_1-1l_2k-1}b^*_{n+1l_1l_2k-1}$
25	$r_{n+1l_1l_2k-1} = B_{n+1l_1l_2k-1}X_{n+2k-1}$

the physical nature of the object, are discussed in [11]–[13]. The latter property would enable the adaptive filter to generate a "good" estimate of the specular component prior to the appearance of the resonant, and the former property guarantees that the system continues to provide a reasonable estimate of the specular part, even after the resonance has appeared. Consequently, the adaptive scheme should have fast tracking at the beginning, whereas after the resonance has occurred, slow tracking would be more desirable.

To study the effects of varying the block length on the overall performance of the modified block FTF algorithm, two different choices for the block length were considered. In the first case, a fixed block length of 300 samples was employed, whereas in the second case, variable-length blocks of sizes 100, 200, 300, and L, where L represents a block length containing the rest of the samples in the backscattered signal, were used. The weight updating rule generated a weight vector for each block, and then, the filtering was performed in each block in one pass. The overall performance of the variable block length scheme was substantially better than those of the fixed-size cases as the tracking ability can be controlled by varying the block sizes. This can be attributed to the fact that the first few hundred samples of the backscattered signal mostly contain the specular part, after which, the resonant part may appear. Thus, using smaller blocks at the beginning generally would provide a better tracking capability for the specular part, whereas after resonance has appeared, switching to bigger block lengths would reduce the tracking of the backscattered signal.

Finally, the results of the modified block FTF algorithm developed in this paper were benchmarked against those of the standard RLS scheme in [8]. This comparison indicated that block FTF with variable block length as described above provided substantially better results for the target cases, whereas the RLS results for the nontarget cases were slightly better. However, overall the winner was the modified block FTF algorithm with variable block length. The results for aspect angle 55° are shown in Figs. 1–4. Note that 0° corresponds to broadside incident. These plots show the backscattered signals in Figs. 1(a)–4(a), estimates of the resonant in Figs. 1(b)–4(b), and specular components in Figs. 1(c)–4(c) and their corresponding spectra, respectively. As can be seen in the results

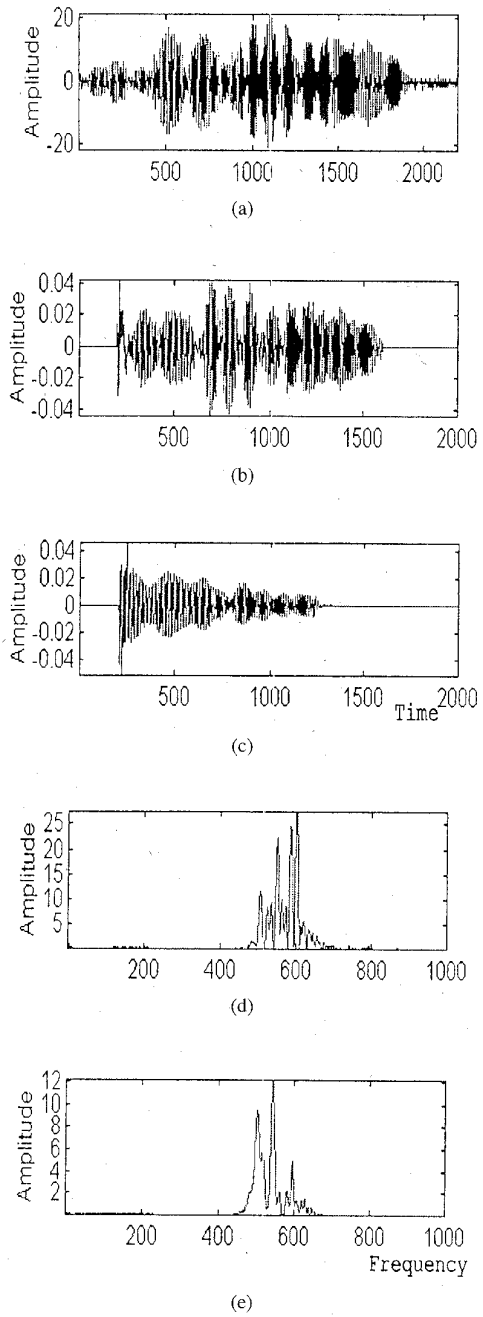


Fig. 1. Backscattered signal, error, and output of the RLS-based adaptive system and their spectra (target at aspect angle 55°).

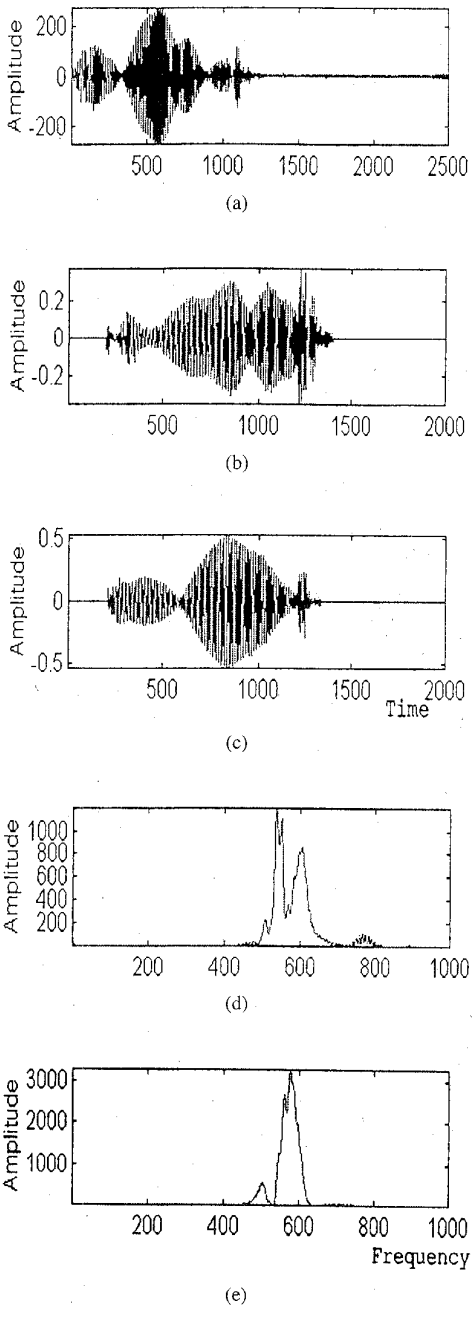


Fig. 2. Backscattered signal, error, and output of the RLS-based adaptive system and their spectra (nontarget at aspect angle 55°).

of Figs. 1 and 3, for the target cases, the output of the adaptive system, which provides the estimate of the specular part, has a broad-band spectrum, whereas the error signal, which gives an estimate of the resonant part, generally contains one or more narrowband components, indicating the presence of an elastic target. This separation of broadband and narrowband components is not so evident in the results of Figs. 2 and 4, for the nontarget cases, which is typically due to absence of resonance. Additionally, comparing the results in Fig. 1 for the RLS case with those in Fig. 3 for the modified block FTF clearly indicates the superiority of the signal separation for the latter scheme. This can especially be seen in the results of Fig. 3(c) for the output of the modified block FTF, which shows close resemblance of this specular return estimate to the linear FM incident. Overall, the classification rates of the block FTF method with variable block length for both the target and the nontarget cases measured based on the data from seventy two aspect angles from 0 to 360° with 5° in between were measured approximately to be 86 and 79%, respectively. These rates are determined based on the evaluation of the characteristics of the spectra of the output and error signals.

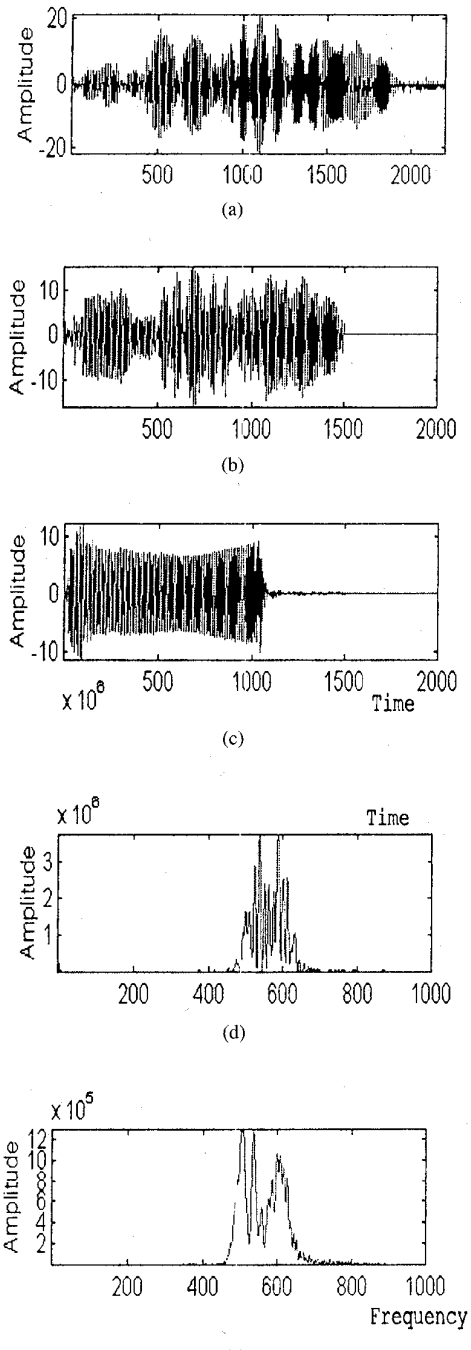


Fig. 3. Backscattered signal, error, and output of the modified BFTF system and their spectra (target at aspect angle 55°).

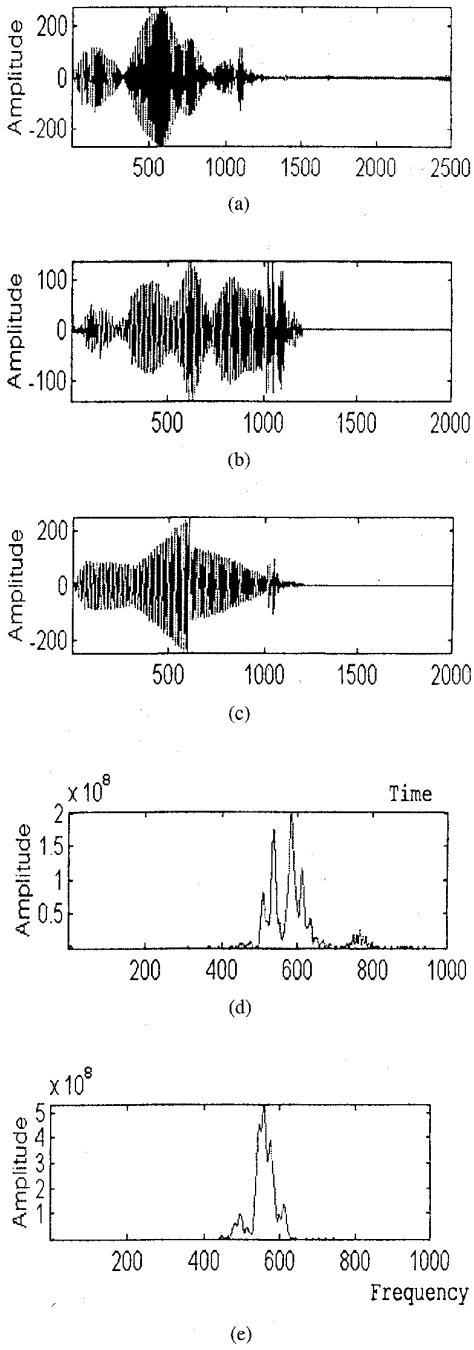


Fig. 4. Backscattered signal, error, and output of the modified BFTF system and their spectra (nontarget at aspect angle 55°).

VI. Conclusion

The problem of block RLS adaptive filtering of nonstationary data is addressed. In many nonstationary environments, it is essential to control the tracking ability of the adaptation process while maintaining the optimality of solution. This is accomplished in this work by developing a modified block FTF adaptive algorithm that can process independent variable length blocks of data or consecutive blocks of data with a better information transfer between the blocks. This modified block FTF adaptive algorithm provides optimal solution to certain types of soft-constrained LS problems. Additionally, it allows variable tracking by changing the number of blocks, their lengths, and/or a soft constrained parameter. The main feature of this approach is that it can provide exact implementation of LS solution by a proper choice of the soft-constrained parameter. Consequently, it can be considered to be a modification and generalization to the original block FTF. The performance of this algorithm is tested on real data, which involved variable tracking of nonstationary data.

APPENDIX A
PROOF OF THEOREM 2.1

From the following matrix inversion formulae

$$\{X_1^*X_1+X_2^*X_2\}^{-1} = (X_1^*X_1)^{-1} - (X_1^*X_1)^{-1}(X_2^*X_2)\{X_1^*X_1+X_2^*X_2\}^{-1} \tag{A.1}$$

$$\{X_1^*X_1+X_2^*X_2\}^{-1} = (X_2^*X_2)^{-1} - (X_2^*X_2)^{-1}(X_1^*X_1)\{X_1^*X_1+X_2^*X_2\}^{-1} \tag{A.2}$$

and (15), it follows that

$$W^{\mathrm{LS}} = W_1 + (W_2 - W_1)X_2^*X_2\{X_1^*X_1 + X_2^*X_2\}^{-1}. \tag{A.3}$$

It can easily be verified, using the identity $A^{-1} - B^{-1} = A^{-1}(B - A)B^{-1}$, that

$$\{\mu I + X_2^*X_2\}^{-1} - \{X_1^*X_1 + X_2^*X_2\}^{-1} = \{\mu I + X_2^*X_2\}^{-1}(X_1^*X_1 - \mu I)\{X_1^*X_1 + X_2^*X_2\}^{-1}. \tag{A.4}$$

Thus, subtracting (A.3) from (16) and using (A.4), we obtain

$$W^{\mathrm{soft}} - W^{\mathrm{LS}} = (W_2 - W_1)X_2^*X_2\{\mu I + X_2^*X_2\}^{-1} \cdot (X_1^*X_1 - \mu I)\{X_1^*X_1 + X_2^*X_2\}^{-1}. \tag{A.5}$$

Thus

$$\|W^{\mathrm{soft}} - W^{\mathrm{LS}}\| \le \|W_2 - W_1\|\|X_2^*X_2\|\|\{\mu I + X_2^*X_2\}^{-1}\|\|X_1^*X_1 - \mu I\|\|\{X_1^*X_1 + X_2^*X_2\}^{-1}\|. \tag{A.6}$$

To determine $\|X_1^*X_1 - \mu I\|$, let

$$\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_N \ge 0$$

be the eigenvalues of the positive semi-definite matrix $X_1^*X_1$. Then

$$\|(X_1^*X_1 - \mu I)\| = |\lambda_{\max}(X_1^*X_1 - \mu I)| = \max\{|\lambda_i - \mu|\} = \max\{|\lambda_1 - \mu|, |\lambda_N - \mu|\}.$$

This implies that the minimum of $\|X_1^*X_1 - \mu I\|$ occurs at $\mu = (\lambda_1 + \lambda_N)/2$, in which case, $\|X_1^*X_1 - \mu I\| = (\lambda_1 - \lambda_N)/2$, i.e., proportional to the eigenvalue spread of $X_1^*X_1$.

On the other hand

$$\|\{\mu I + X_2^*X_2\}^{-1}\| \le \frac{1}{\lambda_{\min}(X_2^*X_2)}$$

provided that $\lambda_{\min}(X_2^*X_2) \neq 0$. Therefore, if the constant A is chosen as

$$A = \frac{\|W_2 - W_1\|\|X_2^*X_2\|\|\{X_1^*X_1 + X_2^*X_2\}^{-1}\|}{2\lambda_{\min}(X_2^*X_2)}$$

then

$$\|W^{\mathrm{soft}} - W^{\mathrm{LS}}\| \le 2A\|X_1^*X_1 - \mu I\|.$$

The conclusion of the theorem follows by setting $\mu = (\lambda_1 + \lambda_N)/2$. Q.E.D.

APPENDIX B
PROOF OF THEOREM 3.1

a) Let W_M^{soft} be the optimal solution of (20); then

$$\frac{\partial \xi_M^{\mathrm{soft}}}{\partial W_M} = 2(W_M^{\mathrm{soft}} - W_{M-1}^{\mathrm{soft}})\sum_{i=r}^{M-1}\mu_i X_i^*X_i - 2(\boldsymbol{d}_M^* - W_M^{\mathrm{soft}}X_M^*)X_M = 0. \tag{B.1}$$

Let $\Sigma_j = X_j^*X_j + \Sigma_{i=r}^{j-1}\mu_i X_i^*X_i$ be the weighted autocorrelation matrix; then, $\Sigma_{j+1} = \Sigma_j + X_{j+1}^*X_{j+1} + (\mu_j - 1)X_j^*X_j$. Thus, (B.1) simplifies to

$$W_M^{\mathrm{soft}}\Sigma_M = \boldsymbol{d}_M^*X_M + W_{M-1}^{\mathrm{soft}}\left(\sum_{i=r}^{M-1}\mu_i X_i^*X_i\right) = \boldsymbol{d}_M^*X_M + W_{M-1}^{\mathrm{soft}}(\Sigma_M - X_M^*X_M).$$

Therefore

$$W_M^{\mathrm{soft}}\Sigma_M = \boldsymbol{d}_M^*X_M + W_{M-1}^{\mathrm{soft}}(\Sigma_M - X_M^*X_M)$$

from which it follows that

$$(W_M^{\mathrm{soft}} - W_{M-1}^{\mathrm{soft}})\Sigma_M = \boldsymbol{d}_M^*X_M - W_{M-1}^{\mathrm{soft}}X_M^*X_M = (\boldsymbol{d}_M^* - W_{M-1}^{\mathrm{soft}}X_M^*)X_M.$$

Next, let W_M^{LS} be the optimal solution of (21). Then

$$\frac{\partial \xi_M^{\mathrm{LS}}}{\partial W} = -2(\boldsymbol{d}_M^* - W_M^{\mathrm{LS}}X_M^*)X_M - 2\sum_{i=r}^{M-1}\mu_i(\boldsymbol{d}_i^* - W_M^{\mathrm{LS}}X_i^*)X_i = 0$$

from which it follows that

$$W_M^{\mathrm{LS}}\Sigma_M = \boldsymbol{d}_M^*X_M + \sum_{i=r}^{M-1}\mu_i\boldsymbol{d}_i^*X_i. \tag{B.2}$$

In addition

$$W_{M-1}^{\mathrm{LS}}\Sigma_{M-1} = \boldsymbol{d}_{M-1}^*X_{M-1} + \sum_{i=r}^{M-2}\mu_i\boldsymbol{d}_i^*X_i. \tag{B.3}$$

Since $\Sigma_{M-1} = \Sigma_M - X_M^*X_M + (\mu_{M-1} - 1)X_{M-1}^*X_{M-1}$, then (B.3) becomes

$$W_{M-1}^{\mathrm{LS}}\Sigma_M = \boldsymbol{d}_{M-1}^*X_{M-1} + \sum_{i=r}^{M-2}\mu_i\boldsymbol{d}_i^*X_i + W_{M-1}^{\mathrm{LS}}X_M^*X_M - (\mu_{M-1} - 1)\cdot W_{M-1}^{\mathrm{LS}}X_{M-1}^*X_{M-1}. \tag{B.4}$$

Subtracting (B.3) from (B.2) yields

$$(W_M^{\mathrm{LS}} - W_{M-1}^{\mathrm{LS}})\Sigma_M = (\boldsymbol{d}_M^* - W_{M-1}^{\mathrm{LS}}X_M^*)X_M + (\mu_{M-1} - 1)(\boldsymbol{d}_{M-1}^* - W_{M-1}^{\mathrm{LS}}X_{M-1}^*)X_{M-1}$$

which is equivalent to (23).

b) Since $W_1^{\mathrm{soft}} = W_1^{\mathrm{LS}}$, it follows from (22) and (23) that $W_i^{\mathrm{soft}} = W_i^{\mathrm{LS}}$ if and only if $\mu_i = 1$ for $i = 1, \cdots, M$. In this case, we obtain the standard LS solution for both cases.

Q.E.D.

ACKNOWLEDGMENT

The authors would like to thank Dr. J. Wilbur, Dr. G. Dobeck, and R. Manning at Coastal Systems Station for their valuable comments and support of this work.

REFERENCES

[1] G. A. Clark, S. K. Mitra, and S. R. Parker, "Block implementation of adaptive digital filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, no. 3, pp. 774–752, June 1981.

[2] W. B. Mikhael and F. H. Wu, "Fast algorithms for block FIR adaptive digital filtering," *IEEE Trans. Circuits Syst.*, vol. CAS-34, no. 10, pp. 1152–1160, Oct. 1987.

[3] J. M. Cioffi, "The block-processing FTF adaptive algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, no. 1, pp. 77–90, Feb. 1986.

[4] J. P. Sessarego, J. Sageloli, and M. Zakharia, "Time-frequency analysis of signals related to scattering problems in acoustics-Part I: Wigner-Ville analysis of echoes scattered by spherical shell, in *Wavelets* J. M. Combes, A. Grossmann, and P. Tchamitchian, Eds. New York: Springer-Verlag, 1987.

[5] J. Wilbur and S. G. Kargl, "Application of wavelets to acoustic resonance-elastic targets surrounded by biologics," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP'93)*, Minneapolis, MN, Apr., 1993, pp. 492–495,

[6] B. Telfer, H. Szu, and G. Dobeck, "Adaptive wavelet classification of acoustic backscatter," *SPIE (Wavelet Applications)*, pp. 661–668, 1994.

[7] M. de Billy, "Determination of the resonance spectrum of elastic bodies via the use of short pulses and Fourier Transform Theory", *J. Acoust. Soc. Amer.*, vol. 79, no. 2, Feb. 1986.

[8] M. R. Azimi-Sadjadi, J. Wilbur, and G. J. Dobeck, "Isolation of resonance in acoustic backscatter from elastic targets using adaptive estimation schemes," *IEEE J. Ocean. Eng.*, vol. 20, pp. 346–353, Oct. 1995.

[9] J. M. Cioffi and T. Kailath, "Fast recursive-least squares transversal filters for adaptive filtering," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 304–337, Apr. 1984.

[10] S. T. Alexander, *Adaptive Signal Processing, Theory and Applications*. New York: Springer-Verlag, 1986.

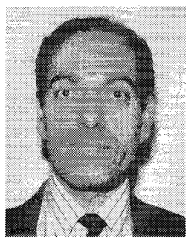
[11] L. G. Zhang, N. H. Sun, and P. L. Martson, "Mid-frequency enhancement of the backscattering of tone bursts by thin spherical shells," *J. Acoust. Soc. Amer.*, vol. 91, no. 4, pt. 1, Apr. 1992.

[12] G. S. Sammelmann, and R. H. Hackman, "The acoustic scattering by a submerged, spherical shell. II: The high frequency region and the thickness quasiresonance," *J. Acoust. Soc. Amer.*, vol. 89, no. 5, May 1991.

[13] G. S. Sammelmann, D. H. Trivett, and R. H. Hackman, "The acoustic scattering by a submerged, spherical shell. I: The bifurcation of the dispersion curve for the spherical antisymmetric Lamb wave," *J. Acoust. Soc. Amer.*, vol. 85, no. 1, Jan. 1989.

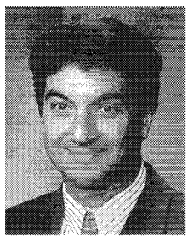
[14] B. Toplis and S. Pasupathy, "Tracking improvements in fast RLS algorithms using a variable forgetting factor," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, no. 2, pp. 206–227, Feb. 1988.

[15] D. T. M. Slock and T. Kailath, "Fast transversal filters with data sequence weighting," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, no. 3, pp. 346–359, Mar. 1989.



Mohammed A. Hasan received the Ph.D. degree in mathematics from Colorado State University, Fort Collins, in 1991. He received the M.S. degree in electrical engineering from the same University in 1992.

His research interests include adaptive systems, digital signal/image processing, sensor array processing, estimation theory, numerical analysis, optimization, numerical linear algebra, and computational and applied mathematics.



Mahmood R. Azimi-Sadjadi (SM'89) was born in Tehran, Iran, in 1952. He received the B.S. degree from the University of Teheran in 1977 and the M.Sc. and Ph.D. degrees from the Imperial College, University of London, England, in 1978 and 1982, respectively, all in electrical engineering.

He served as an Assistant Professor in the Department of Electrical and Computer Engineering, University of Michigan–Dearborn. Since July 1986, he has been with the Department of Electrical Engineering, Colorado State University, Fort Collins, where he is now a Professor. His areas of interest are digital signal/image processing, multidimensional system theory and analysis, adaptive filtering and system identification, target detection and tracking, and neural networks. He is coauthor of the book *Digital Filtering in One and Two Dimensions* (New York: Plenum, 1989).

Dr. Azimi-Sadjadi received the 1994 Senior ASEE-Navy Summer Faculty Fellowship Award, the 1990 Battelle Summer Faculty Fellowship Award, and the 1984 Dow Chemical Outstanding Young Faculty Award of the American Society for Engineering Education.