Transform Domain CPtNLMS Algorithms

Kevin T. Wagner

Naval Research Laboratory
Radar Division
Washington, DC 20375, USA

Miloš I. Doroslovački

The George Washington University
Department of Electrical and Computer Engineering
Washington, DC 20052, USA

Abstract—The concept of self-orthogonalizing adaptation is extended from the least mean square algorithm to the general case of complex proportionate type normalized least mean square algorithms. The derived algorithm requires knowledge of the input signal's covariance matrix. Implementation of the algorithm using a fixed transform such as the discrete cosine transform or discrete wavelet transform is presented for applications in which the input signal's covariance matrix is unknown.

Index Terms—Adaptive filtering, convergence, least mean square algorithms.

I. INTRODUCTION

The self-orthogonalizing adaptive filter [1] was originally introduced in an effort to improve the mean square error (MSE) convergence of the least mean square (LMS) [1] algorithm when the input signal was colored. The original derivation of the self-orthogonalizing filter required knowledge of the covariance matrix. The concept of self-orthogonalizing adaptive filter was subsequently extended to situations where the covariance matrix was unknown. In this extension, the discrete cosine transform (DCT) matrix was used as an estimate of the Karhunen-Loeve transform (KLT). The DCT-LMS [1] algorithm was proposed which initially whitened the input signal using the DCT matrix and then applied the LMS algorithm to the whitened input signal. The DCT-LMS algorithm offered improved convergence in colored input signal scenarios at the expense of increased computational complexity.

In this work the self-orthogonalizing adaptive filter is introduced to the general case of complex proportionate type normalized least mean square (CPtNLMS) [2] algorithms for complex colored input signals and complex impulse responses. The self-orthogonalizing CPtNLMS algorithm is derived for an arbitrary gain control matrix. The resulting algorithm is named the transform domain CPtNLMS (TD-CPtNLMS) algorithm. Next, the TD-CPtNLMS algorithm is extended to the case of an unknown covariance matrix. The DCT and Haar wavelet transform matrices are proposed as possible substitutes for the unknown eigenvector matrix which results in the DCT-CPtNLMS and Haar-CPtNLMS algorithms. The trade off in convergence performance between applying standard CPtNLMS algorithms on the original sparse impulse response and colored input signals as opposed to applying the DCT-CPtNLMS algorithm and Haar-CPtNLMS algorithm on the transformed impulse response (not necessarily sparse anymore) and a whitened input signal is compared through

simulation.

This paper is organized in the following fashion. In the next section the complex adaptive filter algorithm is introduced along with notation. Then the TD-CPtNLMS algorithm is derived. Next, a discussion on the implementation of the TD-CPtNLMS algorithm when the covariance matrix is unknown. Finally, simulation results are presented and discussed.

II. COMPLEX ADAPTIVE FILTER ALGORITHM INTRODUCTION AND NOTATION

A. Notation

Let $\mathbf{A}$ denote an arbitrary complex-valued matrix. The conjugate, transpose, and conjugate transpose of the matrix $\mathbf{A}$ are denoted by $\mathbf{A}^*$, $\mathbf{A}^T$, and $\mathbf{A}^H$; respectively. These operators have the same meanings when employed on vectors and scalars where applicable. Let the i^{th} component of any vector $\mathbf{a}$ be denoted as a_i and the $(k, l)^{\text{th}}$ entry of any matrix $\mathbf{A}$ as $[\mathbf{A}]_{kl}$ throughout this work.

B. Complex Adaptive Filter Framework

All signals are complex throughout this work. Let us assume there is some complex input signal denoted as $\mathbf{x}(k)$ for time k that excites an unknown system with complex impulse response $\mathbf{w}$. Let the output of the system be $y(k) = \mathbf{x}^H(k)\mathbf{w}(k)$ where $\mathbf{x}(k) = [x(k), x(k-1), \dots, x(k-L+1)]^T$ and L is the length of the filter. The measured output of the system, $d(k)$, contains complex-valued measurement noise $v(k)$ and is equal to the sum of $y(k)$ and $v(k)$. The impulse response of the system is estimated with the adaptive filter coefficient vector, $\hat{\mathbf{w}}(k)$, which has length L also. The output of the adaptive filter is given by $\hat{y}(k) = \mathbf{x}^H(k)\hat{\mathbf{w}}(k)$. The error signal $e(k)$ is equal to difference of the measured output, $d(k)$ and the output of the adaptive filter $\hat{y}(k)$.

For notational convenience we will suppress the time-indexing notation and instead use the following convention. For an arbitrary vector $\mathbf{a}$ we denote $\mathbf{a}(k+1) = \mathbf{a}^+$ and $\mathbf{a}(k) = \mathbf{a}$.

III. TRANSFORM DOMAIN CPtNLMS ALGORITHM DERIVATION

The CPtNLMS algorithm using one real-valued gain to update both the real and imaginary parts of the estimated coefficients was originally derived by minimizing

$$(\hat{\mathbf{w}}^+ - \hat{\mathbf{w}})^H \mathbf{G}^{-1} (\hat{\mathbf{w}}^+ - \hat{\mathbf{w}})$$

under the condition

$$d = \mathbf{x}^H \hat{\mathbf{w}}^+$$

where $\mathbf{G}$ is called the stepsize control matrix and is a real-valued, nonnegative, diagonal matrix with $\text{Tr}[\mathbf{G}] = L$. The obtained CPtNLMS algorithm was [2]

$$\hat{\mathbf{w}}^+ = \hat{\mathbf{w}} + \frac{\beta \mathbf{G} \mathbf{x} (d - \mathbf{x}^H \hat{\mathbf{w}})}{\mathbf{x}^H \mathbf{G} \mathbf{x}}.$$

To find the CPtNLMS algorithm in the transform domain we first choose the transform matrix $\mathbf{Q}^H$. Now the input signal to the algorithm is

$$\tilde{\mathbf{x}} = \mathbf{\Lambda}^{-\frac{1}{2}} \mathbf{Q}^H \mathbf{x}.$$

Optimally, i.e. to get $\tilde{\mathbf{x}}$ white, we would take $\mathbf{Q}$ and $\mathbf{\Lambda}$ such that

$$\mathbf{R} = \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^H$$

where $\mathbf{R}$ is the autocorrelation matrix of $\mathbf{x}$, $\mathbf{Q}$ is the matrix of eigenvectors of $\mathbf{R}$, and $\mathbf{\Lambda}$ is the diagonal matrix of eigenvalues of $\mathbf{R}$. Now we can start by minimizing

$$(\hat{\mathbf{w}}_T^+ - \hat{\mathbf{w}}_T)^H \mathbf{G}_T^{-1} (\hat{\mathbf{w}}_T^+ - \hat{\mathbf{w}}_T)$$

under the condition

$$d = \tilde{\mathbf{x}}^H \hat{\mathbf{w}}_T^+$$

where $\mathbf{G}_T$ is called the stepsize control matrix and is a real-valued, nonnegative, diagonal matrix with $\text{Tr}[\mathbf{G}_T] = L$.

The method of Lagrange multipliers will be used to cast this constrained minimization problem into one of unconstrained minimization. The minimization problem can be rewritten as

$$\begin{aligned} \min_{\hat{\mathbf{w}}_T^+} J(\hat{\mathbf{w}}_T^+) &= (\hat{\mathbf{w}}_T^+ - \hat{\mathbf{w}}_T)^T \mathbf{G}_T^{-1} (\hat{\mathbf{w}}_T^+ - \hat{\mathbf{w}}_T) \\ &+ \lambda (d - \tilde{\mathbf{x}}^H \hat{\mathbf{w}}_T^+) \\ &+ \lambda^* (d^* - \tilde{\mathbf{x}}^T (\hat{\mathbf{w}}_T^+)^*). \end{aligned} \quad (1)$$

Next taking the derivative of $J(\hat{\mathbf{w}}_T^+)$ with respect to $\hat{\mathbf{w}}_T^+$ and setting the result to zero yields

$$\frac{\partial J(\hat{\mathbf{w}}_T^+)}{\partial \hat{\mathbf{w}}_T^+} = (\hat{\mathbf{w}}_T^+ - \hat{\mathbf{w}}_T)^H \mathbf{G}_T^{-1} - \lambda \tilde{\mathbf{x}}^H = \mathbf{0}^T. \quad (2)$$

where $\mathbf{0}$ is the column vector of zeros and length L . Taking the conjugate transpose of equation (2), multiplying from the left by $\mathbf{G}_T$, and rearranging terms allows us to write

$$\hat{\mathbf{w}}_T^+ = \hat{\mathbf{w}}_T + \lambda^* \mathbf{G}_T \tilde{\mathbf{x}}. \quad (3)$$

Next we substitute the recursion for the estimated impulse response given in (3) into the constraint equation $d = \tilde{\mathbf{x}}^H \hat{\mathbf{w}}_T^+$ to yield

$$d = \tilde{\mathbf{x}}^H \hat{\mathbf{w}}_T + \lambda^* \tilde{\mathbf{x}}^H \mathbf{G}_T \tilde{\mathbf{x}}. \quad (4)$$

Defining the a priori error as $e = d - \tilde{\mathbf{x}}^H \hat{\mathbf{w}}_T$ and rearranging terms in (4) allows us to solve for λ^* which is given by

$$\lambda^* = \frac{e}{\tilde{\mathbf{x}}^H \mathbf{G}_T \tilde{\mathbf{x}}}. \quad (5)$$

Substituting the solution for λ^* into (3) results in

$$\hat{\mathbf{w}}_T^+ = \hat{\mathbf{w}}_T + \frac{\mathbf{G}_T \tilde{\mathbf{x}} e}{\tilde{\mathbf{x}}^H \mathbf{G}_T \tilde{\mathbf{x}}}. \quad (6)$$

Next the step-size parameter β is introduced to allow control over the update. The resulting algorithm is called the TD-CPtNLMS algorithm and is given by

$$\hat{\mathbf{w}}_T^+ = \hat{\mathbf{w}}_T + \beta \frac{\mathbf{G}_T \tilde{\mathbf{x}} e}{\tilde{\mathbf{x}}^H \mathbf{G}_T \tilde{\mathbf{x}}}. \quad (7)$$

In both cases d is the same and given as

$$d = \mathbf{x}^H \mathbf{w} + v.$$

The optimal (Wiener) weights in the transform domain are

$$\mathbf{w}_T = \mathbf{\Lambda}^{\frac{1}{2}} \mathbf{Q}^H \mathbf{w}.$$

Note that the optimal weights depend on $\mathbf{\Lambda}$ and $\mathbf{Q}^H$, and therefore their sparsity property as well.

IV. TD-CPtNLMS IMPLEMENTATION

If the input signals covariance matrix, $\mathbf{R}$, is not available a priori then the TD-CPtNLMS algorithm is not feasible. To overcome this shortcoming we propose the following modification. Choose some orthogonal transformation, such as the discrete cosine transform (DCT), discrete Fourier transform (DFT), or discrete wavelet transform (DWT), for $\mathbf{Q}$.

At this stage we are still left to estimate $\mathbf{\Lambda}$. In practice the transformed input signal, $\mathbf{x}' = \mathbf{Q}^H \mathbf{x}$, is used to form an estimate of the eigenvalues. The following recursion can be used to estimate the eigenvalues

$$[\tilde{\mathbf{\Lambda}}]_{ii} = \epsilon [\tilde{\mathbf{\Lambda}}]_{ii} + \frac{1}{k} (|x'_i(k)|^2 - \epsilon [\tilde{\mathbf{\Lambda}}]_{ii}) \quad (8)$$

where $\tilde{\mathbf{\Lambda}}$ is the estimate of the eigenvalue matrix, $\mathbf{\Lambda}$, and $0 \leq \epsilon \leq 1$ is the forgetting factor.

V. SIMULATION RESULTS

In this section, we investigate for sparse unknown systems and the colored inputs the question of whether the convergence of CPtNLMS algorithms is better in the original signal domain or in the transformed signal domain, assuming the same steady-state mean square output error in both cases. The input signal used in all of the simulations was a stationary, real, and colored input signal. The input signal consists of colored noise generated by a single pole system as follows:

$$x(k) = \gamma x(k-1) + \zeta(k) \quad (9)$$

where $x(0) = \zeta(0)$, $\zeta(k)$ is a white, real, Gaussian, stationary process with power $\sigma_\zeta^2 = 1$, and γ is a real pole. The value $\gamma = 0.95$ was used, which implies $\sigma_x^2 = \sigma_\zeta^2 / (1 - |\gamma|^2) = 10.2564$. A real impulse response of a telephone network echo path with length $L = 512$ was used. The measurement noise used in all simulations was white, real, Gaussian, and stationary with power $\sigma_v^2 = 10^{-4}$. The parameter values $\beta = 0.1$, and $\epsilon = 1$ were used. The following parameters related to the NLMS and PNLMS algorithms were also used $\rho = 0.01$, $\delta = 0.0001$, and $\delta_p = 0.01$ [2]. The DCT-WF algorithms used the step-size

parameter of $\beta_0 = \beta/(\sigma_x^2 L) = 1.9^{-4}$ to ensure that the steady state MSE was the same for all of the algorithms displayed. The term $\sigma_x^2 = 1$ and is defined to be the variance of $\tilde{x}_i$ for all $i = 1, 2, \dots, L$. Additionally, the self-orthogonalizing water-filling (SO-WF), DCT water-filling (DCT-WF) and Haar water-filling (Haar-WF) algorithms used the following parameters $(\omega, \alpha) = (2, 0.9999)$, $(\omega, \alpha) = (2, 0.9999)$, and $(\omega, \alpha) = (2, 0.9999)$, respectively. The parameters (ω, α) are related to the implementation of the water-filling algorithm [3] and have been tuned to increase the convergence speed of the algorithms.

The first set of simulations examined the mean square error (MSE) versus iteration for the normalized LMS (NLMS) [1], proportionate NLMS (PNLMS) [4], self-orthogonalizing NLMS (SO-NLMS), self-orthogonalizing PNLMS (SO-PNLMS), and SO-WF algorithms. The self-orthogonalizing algorithms used knowledge of the covariance matrix $\mathbf{R}$. In Figure 1, the impulse response in the original domain and the transformed domain are plotted. The transformed impulse response is no longer sparse. Next the learning curve comparison when using the transformation matrix $\mathbf{Q}^H$ is displayed in Figure 2. The SO-NLMS algorithm has the best learning curve performance, followed by the SO-WF and SO-PNLMS algorithms. Since the impulse response is dispersive in the transform domain the SO-NLMS algorithm outperforms the SO-PNLMS algorithm.

The second set of simulations examined the MSE versus iteration for the NLMS, PNLMS, DCT-NLMS, DCT-PNLMS, and DCT-WF. The DCT was used as an estimate of $\mathbf{Q}^H$ in the DCT-NLMS, DCT-PNLMS, and DCT-WF algorithms. In Figure 3, the impulse response in the original domain and the transformed domain are plotted. The transformed impulse response is no longer sparse. Next the learning curve comparison when using the DCT is displayed in Figure 4. Here the DCT-NLMS has the best overall convergence performance followed by the DCT-WF and then the DCT-PNLMS algorithm. Note that the DCT-NLMS has better convergence performance than the DCT-PNLMS algorithm because the transform domain signal is no longer sparse. The PNLMS is better than the NLMS algorithm in the original domain because the impulse response is sparse in this domain. The DCT-NLMS and DCT-PNLMS algorithms outperform their counterpart algorithms in the original signal domain. The counterpart algorithm to the DCT-WF is the complex colored water-filling (CCWF) [5] algorithm. The CCWF is not stable for the value of β chosen in these simulations and has not been included in the figure. Note, the similarity between the KLT and DCT approach, which is expected because asymptotically the DCT becomes the KLT in this scenario [1]. This can be seen further in Figures 1 and 3 where the transform domain impulse responses are similar.

The third set of simulations examine the MSE versus iteration for the NLMS, PNLMS, Haar-NLMS, Haar-PNLMS, and Haar-WF algorithms. The Haar wavelet orthonormal basis was used as an estimate of $\mathbf{Q}^H$ in the Haar-NLMS, Haar-PNLMS, and Haar-WF algorithms. Because $L = 512$, a nine level Haar

Fig. 1. Impulse Response in Original Domain and Transform Domain when $\mathbf{R}$ is Known

Fig. 2. Learning Curve Comparison when $\mathbf{R}$ is Known

wavelet transform was used in these algorithms. In Figure 5 the impulse response in the original domain and the transformed domain are plotted. The transformed impulse response is still sparse in this scenario. Next the learning curve comparison when using the Haar wavelet transform is displayed in Figure 6. The Haar-WF has the best overall convergence performance followed by the Haar-PNLMS, PNLMS, Haar-NLMS and finally the NLMS algorithm. The Haar-PNLMS algorithm has superior convergence performance relative to the Haar-NLMS algorithm because the impulse response is sparse in the transform domain. The Haar-NLMS algorithm does not converge as quickly as the DCT-NLMS algorithm because the Haar wavelet transform does not whiten the colored input signal as well as the DCT. Hence there is a trade-off between whitening the input signal and sparsity of impulse response in the transform domain.

VI. CONCLUSION

The transform domain CPtNLMS algorithm was derived by minimizing the weighted square l^2 -norm of the difference

Fig. 3. Impulse Response in Original Domain and Transform Domain when using the DCT

Fig. 5. Impulse Response in Original Domain and Transform Domain when using the Haar Transform

Fig. 4. Learning Curve Comparison using the DCT

Fig. 6. Learning Curve Comparison using the Haar Transform

between the current weight vector estimate and the weight vector estimate at the next time step. The minimization is performed under the condition that the observation is equal to the inner product of the transformed input signal and the weight vector estimate at the next time step.

The derivation of the TD-CPTNLMS algorithm requires knowledge of the input signal covariance matrix. In order to eliminate this requirement, we proposed choosing some fixed orthogonal transformation, such as the DCT or DWT matrix, as the transformation matrix. The eigenvalues of the transformed input signal are estimated online.

Simulation results were used to show that a trade off exists between whitening the input signal and maintaining the sparsity of the transformed domain impulse response, when choosing the transformation matrix. For instance, the DCT transformation matrix whitens the input signal but does not maintain the sparsity of the impulse response in the transform domain. Therefore, the DCT-NLMS can have superior learning curve performance relative to the DCT-PNLMs algorithm. On

the other, hand the Haar transformation matrix does not whiten the input signal as well as the DCT transformation matrix, but maintains sparsity of the impulse response in the transform domain. Hence, using the Haar transformation matrix can result in the Haar-PNLMs having superior learning curve performance relative to the Haar-NLMS algorithm.

REFERENCES

- [1] S. Haykin, *Adaptive Filter Theory*, fourth edition, Prentice Hall, 2002.
- [2] K. Wagner and M. Doroslovački, "Complex Proportionate-Type Normalized Least Mean Square Algorithms," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Kyoto, Japan, April 19-24, 2012.
- [3] K. Wagner and M. Doroslovački, "Proportionate-Type Normalized Least Mean Square Algorithms With Gain Allocation Motivated by Mean-Square-Error Minimization for White Input," in *IEEE Transactions on Signal Processing*, vol.59, no.5, pp.2410-2415, May 2011.
- [4] D. Duttweiler, "Proportionate normalized least-mean-squares adaptation in echo cancellers," *IEEE Trans. Speech Audio Processing*, vol. 8, pp. 508-518, Sept. 2000.
- [5] K. Wagner and M. Doroslovački, "Complex Colored Water-filling Algorithm for Gain Allocation in Proportionate Adaptive Filtering," in *46th Asilomar Conference on Signals, Systems and Computers*, Nov. 2012.