ABSTRACT

In this paper, we derive a “convolution theorem” suitable for the Hirschman optimal transform (HOT), a unitary transform derived from a discrete-time, discrete-frequency version of the entropy-based uncertainty measure first described by Hirschman [1]. We use the result to develop a fast block-LMS adaptive filter which we call the HOT Block-LMS adaptive filter. This filter requires slightly less than half of the computations that are required for the FFT Block-LMS adaptive filter. The simulations show that the convergence rates of both the HOT and FFT Block-LMS adaptive filters are similar.

Index Terms — adaptive filters, fast Fourier transforms, entropy

1. INTRODUCTION

The HOT is a recently developed discrete unitary transform that uses the orthonormal minimizers of the entropy-based Hirschman uncertainty measure [1]. This measure is different from the energy-based Heisenberg uncertainty measure that is only suited for continuous time signals. The Hirschman uncertainty measure uses entropy to quantify the spread of discrete-time signals in time and frequency [2]. Since the HOT bases are among the minimizers of the uncertainty measure, they have the novel property of being the most compact in discrete domain we can write

\[ y_{i,l} = u_i u_l \]

where \( u_i \) and \( u_l \) are two signals of length \( 2^K \) and \( y_{i,l} \) is their circular convolution. Then, in the frequency domain we can write \( \mathcal{F}\{u\} = U(k)H(k) \). To replace the DFT with the HOT we need the relationship between the DCT and HOT of the signals \( u, h, \) and \( y \). Define \( u_i(l) = u(K+i) \), where \( i,l = 0, 1, 2, \ldots, K-1 \). Then

\[ U(k) = \sum_{i=0}^{K-1} e^{-j\frac{2\pi i}{K}k} U_i(k), \] (2)

where \( U_i(k) \) is the DFT of \( u_i(l) \). In matrix form, Eq. (2) can be written

\[ U = \sum_{i=0}^{K-1} D_{0,K^2-1}(i) \begin{bmatrix} F_K \\ u_i \end{bmatrix} \] (3)

where \( D_{i,j}(k) = \text{diag}\left\{ e^{-j\frac{2\pi i}{K^2}k}, \ldots, e^{-j\frac{2\pi i}{K^2}K} \right\} \) and \( F_K \) is the \( K^2 \)-point DFT matrix. Eq. (3) relates the DFT of the signal \( u \) to its HOT. Therefore, circular convolution in the HOT domain is given by
The HOT could be three times more efficient than the DFT. Multiplications are required. Asymptotically in \( K \), we see that the HOT could be three times more efficient than the DFT.

### 3. THE HOT BLOCK-LMS ADAPTIVE FILTER

In a block adaptive filter, the adaptation proceeds block-by-block with the weight update equation

\[
 w(k+1) = w(k) + \mu \frac{L-1}{L} \sum_{i=0}^{L-1} e(kL+i)u(kL+i),
\]

where \( d \) and \( y \) are the desired and output signals, respectively, \( u \) is the vector that contains the input samples in the \( k \)-th block, \( L \) is the block size or the filter length, and the error is

\[
 e(n) = d(n) - y(n).
\]

The FFT is commonly used to efficiently calculate the output of the filter and the sum in the update equation. For our proposed HOT Block-LMS algorithm, we use the update equation:

\[
 w(k+1) = w(k) + \frac{2\mu}{K} \sum_{i=0}^{K-1} e(kL+i)u(kL+i),
\]

where \( K \) is the block size or the filter length.

We assume that the block size or the filter length is \( K^2/2 \), (our reason for this assumption will become clear) and the block size is equal to the filter order. The parameter \( j \) determines which polyphase component of the error signal is being used in the adaptation. This parameter can be changed from block to block. Therefore, in the adaptation we are only using one polyphase component, rather than the whole signal. The output and the sum in this update equation are efficiently calculated:

\[
 w(k+1) = w(k) + \frac{2\mu}{K} \sum_{i=0}^{K-1} e(kL+i)u(kL+i),
\]

\[
 w(k+1) = w(k) + \frac{2\mu}{K} \sum_{i=0}^{K-1} e(kL+i)u(kL+i),
\]

We assume that the block size or the filter length is \( K^2/2 \). (our reason for this assumption will become clear) and the block size is equal to the filter order. The parameter \( j \) determines which polyphase component of the error signal is being used in the adaptation. This parameter can be changed from block to block. Therefore, in the adaptation we are only using one polyphase component, rather than the whole signal. The output and the sum in this update equation are efficiently calculated:

\[
 w(k+1) = w(k) + \frac{2\mu}{K} \sum_{i=0}^{K-1} e(kL+i)u(kL+i),
\]

We assume that the block size or the filter length is \( K^2/2 \). (our reason for this assumption will become clear) and the block size is equal to the filter order. The parameter \( j \) determines which polyphase component of the error signal is being used in the adaptation. This parameter can be changed from block to block. Therefore, in the adaptation we are only using one polyphase component, rather than the whole signal. The output and the sum in this update equation are efficiently calculated:

\[
 w(k+1) = w(k) + \frac{2\mu}{K} \sum_{i=0}^{K-1} e(kL+i)u(kL+i),
\]
e) Circularly flip the vector in b) and then compute its HOT.
f) Compute the sum in the update equation using Eq. (9) – this sum is that of the first half elements of the circular convolution between the vectors in part e) and d).

The HOT Block LMS adaptive filter is shown in Figure 1. Now we look at the computational cost of the algorithm and compare it to that of the FFT block adaptive algorithm. Parts a), b), and e) require $2^{2 \log_2 K}$ multiplications, part c) requires $2^{2 \log_2 K} + 1$, part d) requires $2^{2 \log_2 K}$, and part f) requires $2^{2 \log_2 K} + 2^{2 \log_2 K}$. The total number of multiplications is thus $2^{2 \log_2 K} + 2^2 \log_2 K + 2^2$. The corresponding FFT block adaptive algorithm requires $2^{2 \log_2 K} + 2^2$ multiplications – asymptotically more than twice as many. Therefore, by using only one polyphase component for the adaptation in a block, the computational cost can be reduced by a factor of 2.5. While this complexity reduction comes at the cost of not using all available information, our proposed algorithm provides better estimates than the LMS filter. The reduction of the computational complexity in our algorithm comes from using the polyphase components of the input signal to calculate one polyphase component of the output using HOT. It is worth mentioning that the fast exact LMS adaptive algorithm (FELMS) [6] also reduces the computational complexities by finding the output via processing the polyphase components of the input. However, the computational complexity reduction of the FELMS is less than that found in the FFT and HOT block adaptive algorithms because the FELMS is designed to have exact mathematical equivalence to, and hence the same convergence properties as, the conventional LMS.

4. CONVERGENCE ANALYSIS

In this section we look at the convergence analysis of the HOT Block-LMS in time domain. We will assume small step size. The HOT-LMS minimizes the cost,$$
\hat{g}(k) = \frac{2}{K} \sum_{i=0}^{K/2-1} e(kL + iK+j)^2, \quad (12)
$$
which is the average of squared errors in the $j$th error polyphase component. From the statistical LMS theory [7], the HOT Block-LMS can be analyzed using the stochastic difference equation
\[ v(k+1) = (I - \mu \Lambda) v(k) + \phi(k), \quad (13) \]
where $v(k) = Q^H (w_o - w(k))$. $Q$ is the eigenvector matrix of the input autocorrelation matrix, $w_o$ is the optimal Wiener solution, and $\Lambda$ is the diagonal matrix that contains the eigenvalues of the input autocorrelation matrix. The driving force $\phi(k)$ for the HOT Block-LMS is given by
\[ \phi(k) = -\frac{2\mu}{K} Q^H \sum_{i=0}^{K/2} e_o(kL + iK+j) u(kL + iK + j), \quad (14) \]
where $e_o$ is the error produced by the Wiener filter. It is easily shown that $E(\phi(k)) = 0$ and $E(\phi(k)\phi^H(k)) = \frac{2\mu^2 J_o}{K} \Lambda$, where $J_o$ is the minimum MSE produced by the Wiener filter. The mean square of the $l$th component of $v(k)$ is given by
\[ E[|y(k)|^2] = \frac{2\mu J_o}{2 - \mu \lambda_l} \left[ \frac{2\mu^2 J_o \Lambda}{2 - \mu \lambda_l} \right], \quad (15) \]
where $\lambda_l$ is the $l$th eigenvalue of the input autocorrelation matrix. Therefore, the average time constant of the HOT Block-LMS is
\[ \tau = \frac{\frac{l^2}{2\mu} \sum_{l=1}^{L} \lambda_l}{\lambda_l}. \quad (16) \]
The misadjustment can be calculated using the equation

$$\mathfrak{M} = \frac{1}{J_\alpha} \sum_{i=1}^{L} \lambda_i E\left[ v_i(\infty)^2 \right]$$

(17)

Using eq (15), we can find $E\left[ v_i(\infty)^2 \right]$ and substitute the result into eq (17). The misadjustment of the HOT Block-LMS is

$$\mathfrak{M} = \frac{\mu}{K} \sum_{i=1}^{L} \lambda_i$$

(18)

The average time constant of the HOT Block-LMS is the same as that of the FFT Block-LMS. However, the HOT Block-LMS has $K$ times higher misadjustment than the FFT Block-LMS.

Similar to the FFT Block-LMS [8], the convergence of the HOT Block-LMS can be improved by equalizing the modes of the filter by using different step size for each mode. The suitable step size for each mode can be found by estimating the power in each HOT sample using

$$P_l(k) = \gamma P_l(k-1) + (1 - \gamma) |U_i(k)|^2,$$

(19)

where $U_i(k)$ is the $i^{\text{th}}$ HOT sample of the input of the filter and $\gamma$ is a constant close to, but less than 1. The step size for the $i^{\text{th}}$ mode is given by

$$\mu_i = \frac{\alpha}{P_l(k)},$$

(20)

where $\alpha$ is a constant. As we will show in the next section, by assigning to each weight an individual step size, the convergence rate improvement of the HOT Block-LMS is similar to the improvement that can be achieved by the FFT Block-LMS.

5. SIMULATIONS

In this section we simulate the HOT Block-LMS adaptive filter to investigate its performance and compare its convergence rate with the LMS and FFT Block-LMS. The input of the adaptive filter was generated by coloring unit variance white noise using the FIR filter $H(z) = 0.1 + 0.2 z^{-1} + 0.3 z^{-2} + 0.4 z^{-3} + 0.4 z^{-4} + 0.2 z^{-5} + 0.1 z^{-6}$. The desired input was generated using the linear model $d(n) = T \text{ow} u(n) + e_0(n)$, where $e_0(n)$ is the measurement white noise with variance of $10^{-4}$. The learning curves of the LMS, the FFT Block-LMS, and the HOT Block-LMS are shown in Figure 2. The step size was adjusted such that all of the adaptive filters converge at the same rate when they are fed with white noise (the corresponding Figure is not included for the lack of space).

The learning curves show that the convergence rate of the computationally more efficient HOT Block-LMS is close to that of the FFT Block-LMS. This is due to the ability of the HOT basis to approximately diagonalize the autocorrelation matrix of the input.

6. CONCLUSIONS

This paper has proposed the computationally efficient HOT Block-LMS algorithm. In addition to its efficiency, the simulations show that its convergence rate is close to that of the FFT Block-LMS. Our next goal is to quantitatively determine the convergence rate as has been done with the FFT Block LMS [9].