HIRSCHMAN OPTIMAL TRANSFORM BLOCK LMS ADAPTIVE FILTER

Osama Alkhouli osama_k@ou.edu School of Elec. & Comp. Eng. University of Oklahoma

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 entropybased uncertainty measure first described by Hirschman [10]. 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 time and frequency. The fact that the HOT basis sequences have many zero-valued samples, and their resemblance to the DFT basis sequences, makes the HOT computationally attractive. Furthermore, it has been shown recently that a thresholding algorithm using the HOT yields superior frequency resolution of a pure tone in additive white noise to a similar algorithm based on the DFT [3].

The main theorem in [1] describes a method to generate an $N = K^2$ -point ortho-normal HOT basis, where *K* is an integer. A HOT basis sequence of length K^2 is the most compact bases in the time-frequency plane. The 3²-point HOT matrix is

$$\begin{bmatrix} I_3 & I_3 & I_3 \\ & -j\frac{2\pi}{3}I_3 & e^{-j\frac{4\pi}{3}}I_3 \\ & & -j\frac{4\pi}{3}I_3 & e^{-j\frac{8\pi}{3}}I_3 \end{bmatrix}$$
(1)

Eq. (1) indicates that the HOT of any sequence is related to the DFT of some polyphase components of the signal. In fact, we called this property the "1 and ½ dimensionality" of the HOT in [2]. Consequently, for this paper, we will use the terms HOT and DFT of the polyphase components interchangeably. The K^2 -

Victor E. DeBrunner victor.debrunner@eng.fsu.edu Dept. of Elec. & Comp. Eng. Florida State University

point HOT requires fewer computations than the K^2 -point DFT. We used this computational efficiency of the HOT to implement fast convolution algorithms in [4]. When K is a power-of-2 integer, then $K^2 \log_2 K$ (complex) multiplications are needed to compute the HOT, which is half that is required when computing the DFT.

In this paper, we use the computational efficiency of the HOT to implement a fast block-LMS adaptive filter. The fast Block-LMS adaptive filter was first proposed [5] to reduce computational complexity. Our proposed HOT Block-LMS adaptive filter requires less than half of the computations required in the corresponding FFT Block-LMS adaptive filter. This significant complexity reduction could be important in many real time applications. Of course, the complexity reduction comes at some performance cost.

In Section 2, we develop the notion of convolution with the HOT. In Section 3, we develop the HOT Block-LMS algorithm. Section 4 contains our convergence analysis. Simulations are provided in Section 5. There we also examine the performance and costs in some detail. Finally, we conclude.

2. HOT AND CIRCULAR CONVOLUTION

Suppose that *u* and *h* are two signals of length $N = K^2$ and the signal *y* is their circular convolution. Then, in the frequency domain we can write Y(k) = 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 $u_i(l) = u(lK+i)$, where $i, l = 0, 1, 2, \dots, K-l$. Then

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

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

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

where
$$D_{i,l}(k) = \text{diag}\left\{e^{-j\frac{2\pi}{K^2}ki}, \dots, e^{-j\frac{2\pi}{K^2}kl}\right\}$$
 and F_K is the K-

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

$$\begin{split} & \overset{K-1}{\underset{i=0}{\sum}} D_{0,K^{2}-1}(i) \begin{bmatrix} F_{K} \\ \vdots \\ F_{K} \end{bmatrix} y_{i} \\ & = \begin{pmatrix} K-1 \\ \sum D \\ i=0 \ 0, K^{2}-1(i) \begin{bmatrix} F_{K} \\ \vdots \\ F_{K} \end{bmatrix} h_{i} \end{pmatrix} \left(\begin{pmatrix} K-1 \\ \sum D \\ i=0 \ 0, K^{2}-1(i) \begin{bmatrix} F_{K} \\ \vdots \\ F_{K} \end{bmatrix} u_{i} \right) \\ & = \begin{pmatrix} K-1K-1 \\ \sum D \\ i=0 \ 0, K^{2}-1(i+j) \begin{pmatrix} F_{K} \\ \vdots \\ F_{K} \end{bmatrix} h_{i} \end{pmatrix} \left(\begin{pmatrix} F_{K} \\ \vdots \\ F_{K} \end{bmatrix} u_{j} \right) . \end{split}$$

$$(4)$$

The \cdot means element-wise multiplication. Eq (4) is not in the form that we require. To find the *j*th polyphase component of the output $F_K y_j$, we divide it into the following *K* equations:

$$\begin{split} & K^{-1} \sum_{i=0}^{K-1} D_{rK,(r+1)K-1}(i)F_{K}y_{i} \\ & = \sum_{i=0}^{K-1K-1} \sum_{j=0}^{K-1K-1} D_{rK,(r+1)K-1}(i+j)F_{K}h_{i}\bullet F_{K}u_{j}, \end{split}$$
 (5)

where r = 0, 1, 2, ..., K-1. But since

$$D_{rK,(r+1)K-1}(i) = e^{-j\frac{2\pi}{K}ri} D_{0,K-1}(i)$$
(6)

Eq. (5) simplifies to

$$\sum_{i=0}^{K-1} e^{-j\frac{2\pi}{K}ri} D_{0,K-1}(i)F_K y_i$$

$$= \sum_{i=0}^{K-1} \sum_{j=0}^{K-1} e^{-j\frac{2\pi}{K}r(i+j)} D_{0,K-1}(i+j)F_K h_i \bullet F_K u_j$$
(7)

Because the DFT matrix is unitary we have

$$F_{K}y_{s} = \sum_{i=0}^{K-1} \sum_{j=0}^{K-1} \delta(i+j-s) D_{0,K-1}(i+j-s) F_{K}h_{i} \bullet F_{K}u_{j}$$
(8)

Let H_i be the diagonal matrix corresponding to $F_K h_i$. Then the element-wise multiplication in Eq. (8) can be replaced with conventional matrix multiplication. Finally we combine the *K* equations in (8)

$$\begin{bmatrix} F_{K}y_{0} \\ F_{K}y_{1} \\ F_{K}y_{2} \\ \vdots \\ F_{K}y_{K-2} \\ F_{K}y_{K-1} \end{bmatrix} = \begin{bmatrix} H_{0} & DH_{K-1} & DH_{K-2} & \cdots & DH_{2} & DH_{1} \\ H_{1} & H_{0} & DH_{K-1} & \cdots & DH_{3} & DH_{2} \\ H_{2} & H_{1} & H_{0} & \cdots & DH_{4} & DH_{3} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ H_{K-2} & H_{K-3} & H_{K-4} & \cdots & H_{0} & DH_{K-1} \\ H_{K-1} & H_{K-2} & H_{K-3} & \cdots & H_{1} & H_{0} \end{bmatrix} \begin{bmatrix} F_{K}u_{0} \\ F_{K}u_{1} \\ F_{K}u_{2} \\ \vdots \\ F_{K}u_{K-2} \\ F_{K}u_{K-1} \end{bmatrix},$$
(9)

where $D = D_{0,K-1}(K)$. From Eq. (9) one can see that calculating the HOT of the output *y* requires $K^3 + 2K^2 \log_2 K + K^2 - K$ multiplications. When we require only one polyphase component, then only $K^2 + 2K^2 \log_2 K + K \log_2 K$ multiplications are necessary. By way of comparison, if the DFT is used to calculate the output then $K^2 + 6K^2 \log_2 K$ 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-byblock with the weight update equation

$$w(k+1) = w(k) + \frac{\mu}{L} \sum_{i=0}^{L-1} e(kL+i)u(kL+i), \quad (10)$$

where *d* and *y* are the desired and output signals, respectively, *u* is the vector that contains the input samples in the kth 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:

$$w(k+1) = w(k) + \frac{2\mu}{K} \sum_{i=0}^{K-1} e(kL+iK+j)u(kL+iK+j).$$
(11)

We assume that the block size or the filter length is $\frac{K^2}{2}$ (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:

- a) Append the weight vectors with $\frac{K^2}{2}$ zeros (the resulting vector is now K^2 points long as required in the HOT
- definition) and find its HOT.
 b) Use the inverse HOT and Eq. (9) to calculate the jth polyphase component of the circular convolution. The jth polyphase component of the output can be found by discarding the first half of the jth polyphase component of the circular convolution.
- c) Use the inverse HOT and Eq. (9) to calculate the jth polyphase component of the circular convolution. The jth polyphase component of the output can be found by discarding the first half of the jth polyphase component of the circular convolution.
- d) Calculate the jth polyphase component of the error, insert a block of $\frac{K}{2}$ zeros, up-sample by K, then calculate its HOT.



Fig. 1 HOT Block-LMS Algorithm.

- 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 $3K^2\log_2 K$ require e) multiplications, part c) requires $K \log_2 K + K^2$, part d) requires $K \log_2 K$, and part f) requires $K^2 + K^2 \log_2 K$. The total number of multiplications is thus $4K^2 \log_2 K + 2K \log_2 K + 2K^2$. The corresponding FFT block adaptive algorithm requires $10K^2 \log_2 K + 2K^2$ 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. COVERGENCE 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,

$$\hat{\xi}(k) = \frac{2}{K} \sum_{i=0}^{K/2-1} \left| e(kL + iK + j) \right|^2,$$
(12)

which is the average of squared errors in the jth error polyphase component. From the statistical LMS theory [7], the HOT Block-LMS can be analyzed using the stochastic difference equation

$$v(k+1) = (I - \mu \Lambda)v(k) + \phi(k),$$
 (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 Λ is the diagonal matrix that contains the eignvalues 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 $E\{\phi(k)\phi^H(k)\}=\frac{2\mu^2 J_o \Lambda}{K}$, 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

$$E\{|v_l(k)|^2\} = \frac{2\mu J_o/K}{2-\mu\lambda_l} + (1-\mu\lambda_l)^{2k} \left(|v_l(0)|^2 - \frac{2\mu J_o/K}{2-\mu\lambda_l}\right),$$
(15)

where λ_l is the l^{th} eigenvalue of the input autocorrelation matrix. Therefore, the average time constant of the HOT Block-LMS is

$$\tau = \frac{L^2}{2\mu \sum_{l=1}^{L} \lambda_l}.$$
(16)

The misadjustment can be calculated using the equation

$$\mathfrak{M} = \frac{\sum_{l=1}^{L} \lambda_l E\{\left|v_l(\infty)\right|^2\}}{J_o}$$
(17)

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

$$\mathfrak{M} = \frac{\mu}{K} \sum_{l=1}^{L} \lambda_l \tag{18}$$

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

$$P_{i}(k) = \gamma P_{i}(k-1) + (1-\gamma) \left| U_{i}(k) \right|^{2},$$
(19)

where $U_i(k)$ is the ith HOT sample of the input of the filter and γ is a constant close to, but less than 1. The step size for the *i*th mode is given by

$$\mu_i = \frac{\alpha}{P_i(k)},\tag{20}$$

where α 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 $H(z) = 0.1 + 0.2 z^{-1} + 0.3z^{-2} + 0.4z^{-3} + 0.4z^{-4} + 0.2z^{-5} + 0.1z^{-6}$. The desired input was generated using the linear model $d(n) = w_o^T u(n) + e_o(n)$, where $e_o(n)$ 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].



Fig. 2 The learning curves of the LMS, the FFT Block-LMS, and the HOT Block-LMS.

7. REFERENCES

- T. Przebinda, V. DeBrunner, and M. Özaydın, "The optimal transform for the discrete Hirschman uncertainty principle," *IEEE Trans. Infor. Theory*, pp. 2086-2090, Jul 2001.
- [2] V. DeBrunner, M. Özaydın, and T. Przebinda, "Resolution in time-frequency," *IEEE Trans. SP*, pp. 783-788, Mar 1999.
- [3] V. E. DeBrunner, J. P. Havlicek, T. Przebinda, and M. Özaydın, "Entropy-based uncertainty measures for $L^2(\mathbb{R}^n)$,

 $\ell^2(\mathbb{Z})$, and $\ell^2(\mathbb{Z}/N\mathbb{Z})$ with a Hirschman optimal transform for $\ell^2(\mathbb{Z}/N\mathbb{Z})$," *IEEE Trans. SP*, pp.2690 – 2699 Vol. 53, Issue 8, Part 1, Aug. 2005.

- [4] V. DeBrunner and E. Matusiak, "An algorithm to reduce the complexity required to convolve finite length sequences using the Hirschman optimal transform (HOT)," *ICASSP'03*, Hong Kong, China, pp. II-577-580, Apr 2003.
- [5] G. Clark, S. Mitra, and S. Parker, "Block implementation of adaptive digital filters," *IEEE Trans. ASSP*, pp. 744-752, Jun 1981.
- [6] J. Benesty and P. Duhamel, "A fast exact least mean square adaptive algorithm," *IEEE Trans. SP*, pp. 2904-2920, Dec 1992.
- [7] Simon Haykin "Adaptive Filter Theory", Prentice Hall information and system sciences series, Fourth edition, 2002.
- [8] J. J. Shynk, "Frequency-domain and multirate adaptive filtering," IEEE Signal Processing Mag., vol. 9, no. 1, pp. 14-37, Jan. 1992.
- [9] B. Farhang-Boroujeny and Kheong Sann Chan, "Analysis of the Frequency-Domain Block LMS Algorithm", *IEEE Trans. SP*, pp. 2332, Aug. 2000.
- [10] I. I. Hirschman, "A note on entropy," Amer. J. Math., vol. 79, pp. 152-156, 1957.