Hirschman Optimal Transform Block LMS Adaptive Filter

Osama Alkhouli, Victor DeBrunner and
Joseph Havlicek

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/51394

1. Introduction

The HOT is a recently developed discrete unitary transform that uses the orthonormal minimizers of the entropy-based Hirschman uncertainty measure [2]. 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 [3]. 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 [4]. The main theorem in [2] describes a method to generate an $N = K^2$-point orthonormal 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

$$\begin{bmatrix} 1\,0\,0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0\,1\,0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0\,0\,1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1\,0\,0 & e^{-j2\pi/3} & 0 & 0 & e^{-j4\pi/3} & 0 & 0 \\ 0\,1\,0 & 0 & e^{-j2\pi/3} & 0 & 0 & e^{-j4\pi/3} & 0 \\ 0\,0\,1 & 0 & 0 & e^{-j2\pi/3} & 0 & 0 & e^{-j4\pi/3} \\ 1\,0\,0 & e^{-j4\pi/3} & 0 & 0 & e^{-j8\pi/3} & 0 & 0 \\ 0\,1\,0 & 0 & e^{-j4\pi/3} & 0 & 0 & e^{-j8\pi/3} & 0 \\ 0\,0\,1 & 0 & 0 & e^{-j4\pi/3} & 0 & 0 & e^{-j8\pi/3} \end{bmatrix} \tag{1}$$

Equation (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 1/2 dimensionality" of the HOT in [3]. Consequently, for this chapter, we will use the terms HOT and DFT of the polyphase components interchangeably. The K^2-point HOT requires fewer computations than the K^2-point DFT. We used this computational efficiency of the HOT to implement fast convolution algorithms [5]. 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 chapter, 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 [6] to reduce computational complexity. Our proposed HOT block LMS adaptive filter requires less than half of the computations required in the corresponding DFT block LMS adaptive filter. This significant complexity reduction could be important in many real time applications.

The following notations are used throughout this chapter. Nonbold lowercase letters are used for scalar quantities, bold lowercase is used for vectors, and bold uppercase is used for matrices. Nonbold uppercase letters are used for integer quantities such as length or dimensions. The lowercase letter k is reserved for the block index. The lowercase letter n is reserved for the time index. The time and block indexes are put in brackets, whereas subscripts are used to refer to elements of vectors and matrices. The uppercase letter N is reserved for the filter length and the uppercase letter L is reserved for the block length. The superscripts T and H denote vector or matrix transposition and Hermitian transposition, respectively. The subscripts F and H are used to highlight the DFT and HOT domain quantities, respectively. The $N \times N$ identity matrix is denoted by $\mathbf{I}_{N\times N}$ or $\mathbf{I}$. The $N \times N$ zero matrix is denoted by $\mathbf{0}_{N\times N}$. The linear and circular convolutions are denoted by $*$ and $\star$, respectively. Diag $[\mathbf{v}]$ denotes the diagonal matrix whose diagonal elements are the elements of the vector $\mathbf{v}$.

2. The relation between the HOT and DFT in a matrix from

The algorithm that we proposing is best analyzed if the relation between the HOT and DFT is presented in matrix form. This matrix form is shown in Figure 1, where $\mathbf{I}_0$, $\mathbf{I}_1$,..., $\mathbf{I}_{K-1}$ are $K \times K^2$ matrices such that multiplication of a vector with $\mathbf{I}_i$ produces the i^{th} polyphase component of the vector. The matrix $\mathbf{I}_K$ is formed from $\mathbf{I}_0$, $\mathbf{I}_1$,..., $\mathbf{I}_{K-1}$, i.e.,

$$\mathbf{I}_K = \begin{bmatrix} \mathbf{I}_0 \\ \mathbf{I}_1 \\ \vdots \\ \mathbf{I}_{K-2} \\ \mathbf{I}_{K-1} \end{bmatrix}. \tag{2}$$

Since the rows of $\left\{\mathbf{I}_i\right\}$ are taken from the rows of the $K^2 \times K^2$ identity matrix, multiplications with such matrices does not impose any computational burden. For the special case $K = 3$, we have

$$\mathbf{I}_0 = \begin{bmatrix} 1\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,1\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,1\,0\,0 \end{bmatrix}, \tag{3}$$

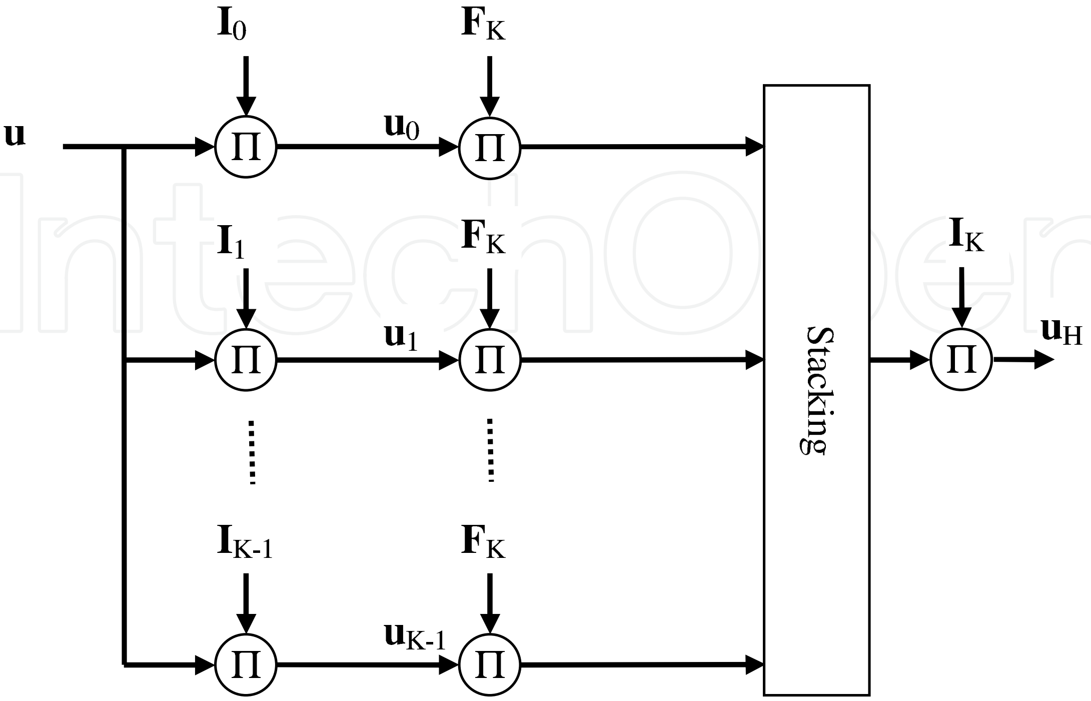

Figure 1. The Relation between HOT and DFTs of the polyphase components.

$$\mathbf{I}_1 = \begin{bmatrix} 0\,1\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,1\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,1\,0 \end{bmatrix}, \tag{4}$$

$$\mathbf{I}_2 = \begin{bmatrix} 0\,0\,1\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,1\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,1 \end{bmatrix}. \tag{5}$$

The K^2-point HOT matrix is denoted by $\mathbf{H}$. It satisfies the following:

$$\mathbf{H}\mathbf{H}^H = K\mathbf{I}_{K^2 \times K^2}, \tag{6}$$

$$\mathbf{H} = \mathbf{H}^T. \tag{7}$$

3. Convolution using the HOT

In this section, the “HOT convolution,” a relation between the HOT of two signals and their circular convolution, is derived. Let u and w be two signals of length K^2. The circular convolution of the signals is $y = w \star u$. In the DFT domain, the convolution is given by the pointwise multiplication of the respective DFTs of the signals, i.e., $y_F(k) = w_F(k)u_F(k)$. A similar relation in the HOT domain can be readily found through the relation between the DFT and HOT. The DFT of u can be written as

$$
\begin{aligned}
u_F(k) &= \sum_{n=0}^{K^2-1} u(n)\, e^{-j\frac{2\pi}{K^2}kn} \\
&= \sum_{i=0}^{K-1} e^{-j\frac{2\pi}{K^2}ki} \sum_{l=0}^{K-1} u(lK+i)\, e^{-j\frac{2\pi}{K}kl}.
\end{aligned}
\tag{8}
$$

The signal $u(lK+i)$, denoted by $u_i(l)$, is the i^{th} polyphase component of $u(n)$ with DFT given by

$$
u_{iF}(k) = \sum_{l=0}^{K-1} u_i(l)\, e^{-j\frac{2\pi}{K}kl}. \tag{9}
$$

Therefore, the DFT of the signal u can be written in terms of the DFTs of the polyphase components, or the HOT of u. The relation between the HOT and the DFTs of the polyphase components is descried in Figure 1. Equation (8) may be written as

$$
u_F(k) = \sum_{i=0}^{K-1} e^{-j\frac{2\pi}{K^2}ki} u_{iF}(k). \tag{10}
$$

Define the diagonal matrix

$$
\mathbf{D}_{i,j}(k) = \begin{bmatrix} e^{-j\frac{2\pi}{K^2}ki} & 0 & \cdots & 0 \\ 0 & e^{-j\frac{2\pi}{K^2}k(i+1)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & e^{-j\frac{2\pi}{K^2}kj} \end{bmatrix} \tag{11}
$$

Then the DFT of the signal can be written in a matrix form

$$
\mathbf{u}_F = \sum_{i=0}^{K-1} \mathbf{D}_{0,K^2-1}(i) \begin{bmatrix} \mathbf{F}_K \\ \mathbf{F}_K \\ \vdots \\ \mathbf{F}_K \end{bmatrix} \mathbf{u}_i. \tag{12}
$$

The above is the desired relation between the DFT and HOT. It should be noted that equation (12) represents a radix-K FFT algorithm which is less efficient than the radix-2 FFT algorithm. Therefore, HOT convolution is expected to be less efficient than DFT convolution. Now, we can use equation (12) to transform $\mathbf{y}_F = \mathbf{w}_F \otimes \mathbf{u}_F$ into the HOT domain. The symbol $\otimes$ indicates pointwise matrix multiplication and, throughout this discussion, pointwise matrix multiplication takes a higher precedence than conventional matrix multiplication. We have that

$$\sum_{i=0}^{K-1} \mathbf{D}_{0,K^2-1}(i) \begin{bmatrix} \mathbf{F}_K \\ \mathbf{F}_K \\ \vdots \\ \mathbf{F}_K \end{bmatrix} \mathbf{y}_i = \sum_{i=0}^{K-1} \sum_{j=0}^{K-1} \mathbf{D}_{0,K^2-1}(i+j) \begin{bmatrix} \mathbf{F}_K \mathbf{w}_i \\ \mathbf{F}_K \mathbf{w}_i \\ \vdots \\ \mathbf{F}_K \mathbf{w}_i \end{bmatrix} \otimes \begin{bmatrix} \mathbf{F}_K \mathbf{u}_j \\ \mathbf{F}_K \mathbf{u}_j \\ \vdots \\ \mathbf{F}_K \mathbf{u}_j \end{bmatrix}. \tag{13}$$

The above matrix equation can be separated into a system of K equations

$$\sum_{i=0}^{K-1} \mathbf{D}_{rK,(r+1)K-1}(i) \mathbf{F}_K \mathbf{y}_i = \sum_{i=0}^{K-1} \sum_{j=0}^{K-1} \mathbf{D}_{rK,(r+1)K-1}(i+j) \left(\mathbf{F}_K \mathbf{w}_i\right) \otimes \left(\mathbf{F}_K \mathbf{w}_j\right), \tag{14}$$

where $r = 0, 1, \ldots, K-1$. Since

$$\mathbf{D}_{rK,(r+1)K-1}(i) = e^{-j\frac{2\pi}{K}ri} \mathbf{D}_{0,K-1}(i), \tag{15}$$

the HOT of the output can be obtained by solving the following set of K matrix equations:

$$\sum_{i=0}^{K-1} e^{-j\frac{2\pi}{K}ri} \mathbf{D}_{0,K-1}(i) \mathbf{F}_K \mathbf{y}_i = \sum_{i=0}^{K-1} \sum_{j=0}^{K-1} e^{-j\frac{2\pi}{K}r(i+j)} \mathbf{D}_{0,K-1}(i+j) \left(\mathbf{F}_K \mathbf{w}_i\right) \otimes \left(\mathbf{F}_K \mathbf{u}_j\right). \tag{16}$$

Since the DFT matrix is unitary, the solution of equation (16) can be expressed as

$$\mathbf{D}_{0,K-1}(s) \mathbf{F}_K \mathbf{y}_s = \frac{1}{K} \sum_{r=0}^{K-1} \sum_{i=0}^{K-1} \sum_{j=0}^{K-1} e^{j\frac{2\pi}{K}r(s-(i+j))} \mathbf{D}_{0,K-1}(i+j) \left(\mathbf{F}_K \mathbf{w}_i\right) \otimes \left(\mathbf{F}_K \mathbf{u}_j\right), \tag{17}$$

where

$$\mathbf{F}_K \mathbf{y}_s = \frac{1}{K} \sum_{r=0}^{K-1} \sum_{i=0}^{K-1} \sum_{j=0}^{K-1} e^{j\frac{2\pi}{K}r(i+j-s)} \mathbf{D}_{0,K-1}(i+j-s) \left(\mathbf{F}_K \mathbf{w}_i\right) \otimes \left(\mathbf{F}_K \mathbf{u}_j\right). \tag{18}$$

Moreover, as

$$\sum_{r=0}^{K-1} e^{j\frac{2\pi}{K}r(i+j-s)} = K\delta(i+j-s), \tag{19}$$

where $\delta(n)$ denotes the periodic Kronecker delta of periodicity K, equation (18) can be simplified to

$$\mathbf{F}_K \mathbf{y}_s = \sum_{i=0}^{K-1} \sum_{j=0}^{K-1} \delta(i+j-s) \mathbf{D}_{0,K-1}(i+j-s) \left(\mathbf{F}_K \mathbf{w}_i\right) \otimes \left(\mathbf{F}_K \mathbf{u}_j\right), \tag{20}$$

where $s = 0, 1, 2, \ldots, K-1$. The pointwise matrix multiplication in equation equation (20) can be converted into conventional matrix multiplication if we define $\mathbf{W}_i$ as the diagonal matrix for $\mathbf{F}_K \mathbf{w}_i$. We have then that

$$\mathbf{F}_K\mathbf{y}_s = \sum_{i=0}^{K-1}\sum_{j=0}^{K-1}\delta(i+j-s)\mathbf{D}_{0,K-1}(i+j-s)\mathbf{W}_i\mathbf{F}_K\mathbf{u}_j. \tag{21}$$

Combining the above K equations into one matrix equation, the HOT convolution can be written as

$$\begin{bmatrix}\mathbf{F}_K\mathbf{y}_0\\ \mathbf{F}_K\mathbf{y}_1\\ \mathbf{F}_K\mathbf{y}_2\\ \vdots\\ \mathbf{F}_K\mathbf{y}_{K-2}\\ \mathbf{F}_K\mathbf{y}_{K-1}\end{bmatrix} = \begin{bmatrix}\mathbf{W}_0 & \mathbf{DW}_{K-1} & \mathbf{DW}_{K-2} & \cdots & \mathbf{DW}_2 & \mathbf{DW}_1\\ \mathbf{W}_1 & \mathbf{W}_0 & \mathbf{W}_{K-1} & \cdots & \mathbf{DW}_3 & \mathbf{DW}_2\\ \mathbf{W}_2 & \mathbf{W}_1 & \mathbf{W}_0 & \cdots & \mathbf{DW}_4 & \mathbf{DW}_3\\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\ \mathbf{W}_{K-2} & \mathbf{W}_{K-3} & \mathbf{W}_{K-4} & \cdots & \mathbf{W}_0 & \mathbf{DW}_{K-1}\\ \mathbf{W}_{K-1} & \mathbf{W}_{K-2} & \mathbf{W}_{K-3} & \cdots & \mathbf{W}_1 & \mathbf{W}_0\end{bmatrix}\begin{bmatrix}\mathbf{F}_K\mathbf{u}_0\\ \mathbf{F}_K\mathbf{u}_1\\ \mathbf{F}_K\mathbf{u}_2\\ \vdots\\ \mathbf{F}_K\mathbf{u}_{K-2}\\ \mathbf{F}_K\mathbf{u}_{K-1}\end{bmatrix} \tag{22}$$

where

$$\mathbf{D} = \begin{bmatrix}1 & 0 & \cdots & 0\\ 0 & e^{-j\frac{2\pi}{K^2}} & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & e^{-j\frac{2\pi}{K^2}(K-1)}\end{bmatrix} \tag{23}$$

Notice that the square matrix in equation (22) is arranged in a block Toeplitz structure.

A better understanding of this result may be obtained by comparing equation (22) with the K-point circular convolution

$$\begin{bmatrix}y_0\\ y_1\\ y_2\\ \vdots\\ y_{K-2}\\ y_{K-1}\end{bmatrix} = \begin{bmatrix}w_0 & w_{K-1} & w_{K-2} & \cdots & w_2 & w_1\\ w_1 & w_0 & w_{K-1} & \cdots & w_3 & w_2\\ w_2 & w_1 & w_0 & \cdots & w_4 & w_3\\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\ w_{K-2} & w_{K-3} & w_{K-4} & \cdots & w_0 & w_{K-1}\\ w_{K-1} & w_{K-2} & w_{K-3} & \cdots & w_1 & w_0\end{bmatrix}\begin{bmatrix}u_0\\ u_1\\ u_2\\ \vdots\\ u_{K-2}\\ u_{K-1}\end{bmatrix}. \tag{24}$$

The square matrix in equation (24) is also Toeplitz. However, equation (24) is a pure time domain result, whereas equation (22) is a pure HOT domain relation, which may be interpreted in terms of both the time domain and the DFT domain features. This fact can be explained in terms of fact that the HOT basis is optimal in the sense of the entropic joint time-frequency uncertainty measure $H_p(u) = pH(u) + (1-p)H(u_F)$ for all $0 \le p \le 1$. Before moving on to the computational complexity analysis of HOT convolution, we make the same observations about the term $\mathbf{DF}_K\mathbf{w}_i$ appearing in equation (22). This term is the complex conjugate of the DFT of the upside down flipped i^{th} polyphase component of w.

It should be noted that equation (22) does not show explicitly the HOT of $u(n)$ and $w(n)$. However, the DFT of the polyphase components that are shown explicitly in equation (22) are related to the HOT of the corresponding signal as shown in Figure. 1. For example, the 0^{th} polyphase component of the output is given by

$$\mathbf{y}_0(k) = \mathbf{F}_K^{-1}\mathbf{I}_0\mathbf{w}_H(k) \otimes \mathbf{I}_0\mathbf{u}_H(k) + \mathbf{F}_K^{-1}\mathbf{D}\sum_{i=1}^{K-1}\mathbf{I}_{K-i}\mathbf{w}_H(k) \otimes \mathbf{I}_i\mathbf{u}_H(k). \tag{25}$$

Next, we examine the computational complexity of HOT convolution. To find the HOT of the two signals w and u, $2K^2\log_2 K$ multiplications are required. Multiplication with the diagonal matrix D requires $K(K-1)$ multiplications. Finally, the matrix multiplication requires K^3 scalar multiplications. Therefore, the total number of multiplications required is $2K^2\log_2 K + K^3 + K^2 - K$. Thus, computation of the output y using the HOT requires $K^3 + 3K^2\log_2 K + K^3 + K^2 - K$ multiplications, which is more than $6K^2\log_2 K + K^2$ as required by the DFT. When it is required to calculate only one polyphase component of the output, only $K^2 + 2K^2\log_2 K + K\log_2 K$ multiplications are necessary. Asymptotically in K, we see that the HOT could be three times more efficient than the DFT.

4. Development of the basic algorithm

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

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \frac{\mu}{L}\sum_{i=0}^{L-1}\mathbf{u}(kL+i)e(kL+i), \tag{26}$$

where $d(n)$ and $y(n)$ are the desired and output signals, respectively, $\mathbf{u}(n)$ is the tap-input vector, L is the block length or the filter length, and $e(n) = d(n) - y(n)$ is the filter error. The DFT is commonly used to efficiently calculate the output of the filter and the sum in the update equation. Since the HOT is more efficient than the DFT when it is only required to calculate one polyphase component of the output, the block LMS algorithm equation (26) is modified such that only one polyphase component of the error in the k^{th} block is used to update the filter weights. For reasons that will become clear later, the filter length L is chosen such that $L = K^2/2$. With this modification, equation (26) becomes

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \frac{2\mu}{K}\sum_{i=0}^{K/2-1}\mathbf{u}(kL+iK+j)e(kL+iK+j). \tag{27}$$

Since the DFT is most efficient when the length of the filter is equal to the block length [7], this will be assumed in equation (27). 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. If $j = 0$, the output can be computed using the HOT as in equation (25). A second convolution is needed to compute the sum in equation (27). This sum contains only one polyphase component of the error. If this vector is up-sampled by K, the sum is just a convolution between the input vector and the up-sampled error vector. Although all the polyphase components are needed in the sum, the convolution can be computed by the HOT with the same computational complexity as the first convolution since only one polyphase component of the error vector is non-zero.

The block adaptive filter that implements the above algorithm is called the HOT block LMS adaptive filter and is shown in Figure 2. The complete steps of this new, efficient, adaptive algorithm are summarized below:

(a) Append the weight vector with $K^2/2$ zeros (the resulting vector is now K^2 points long as required in the HOT definition) and find its HOT.

(b) Compute the HOT of the input vector

$$\left[u\left((k-1)\frac{K^2}{2}\right) \cdots u\left(k\frac{K^2}{2}\right) u\left(k\frac{K^2}{2}+1\right) \cdots u\left((k+1)\frac{K^2}{2}-1\right) \right]^T. \tag{28}$$

Note that this vector contains the input samples for the current and previous blocks.

(c) Use the inverse HOT and equation (22) to calculate the j^{th} polyphase component of the circular convolution. The j^{th} polyphase component of the output can be found by discarding the first half of the j^{th} polyphase component of the circular convolution.

(d) Calculate the j^{th} polyphase component of the error, insert a block of $K/2$ zeros, up-sample by K, then calculate its HOT.

(e) Circularly flip the vector in (b) and then compute its HOT.

(f) Compute the sum in the update equation using equation (22). This sum is the first half of the elements of the circular convolution between the vectors in parts (e) and (d).

5. Computational complexity analysis

In this section, we analyze the computational cost of the algorithm and compare it to that of the DFT block adaptive algorithm. Parts (a), (b), and (e) require $3K^2\log_2 K$ multiplications. Part (c) requires $K\log_2 K + K^2$. Part (d) requires $K\log_2 K$ multiplications, and part (f) requires $K^2 + K^2\log_2 K$ multiplications. The total number of multiplications is thus $4K^2\log_2 K + 2K\log_2 K + 2K^2$. The corresponding DFT 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, the proposed algorithm provides better estimates than the LMS filter. The reduction of the computational complexity in this algorithm comes from using the polyphase components of the input signal to calculate one polyphase component of the output via the HOT.

It is worth mentioning that the fast exact LMS (FELMS) adaptive algorithm [8] also reduces the computational complexity by finding the output by processing the polyphase components of the input. However, the computational complexity reduction of the FELMS algorithm is less than that found in the DFT and HOT block adaptive algorithms because the FELMS algorithm is designed to have exact mathematical equivalence to, and hence the same convergence properties as, the conventional LMS algorithm. Comparing the HOT block LMS algorithm with the block LMS algorithms described in Chapter 3, the HOT filter performs computationally better.

The multiplication counts for both the DFT block and HOT block LMS algorithms are plotted in Figure 3. The HOT block LMS adaptive filter is always more efficient than the DFT block

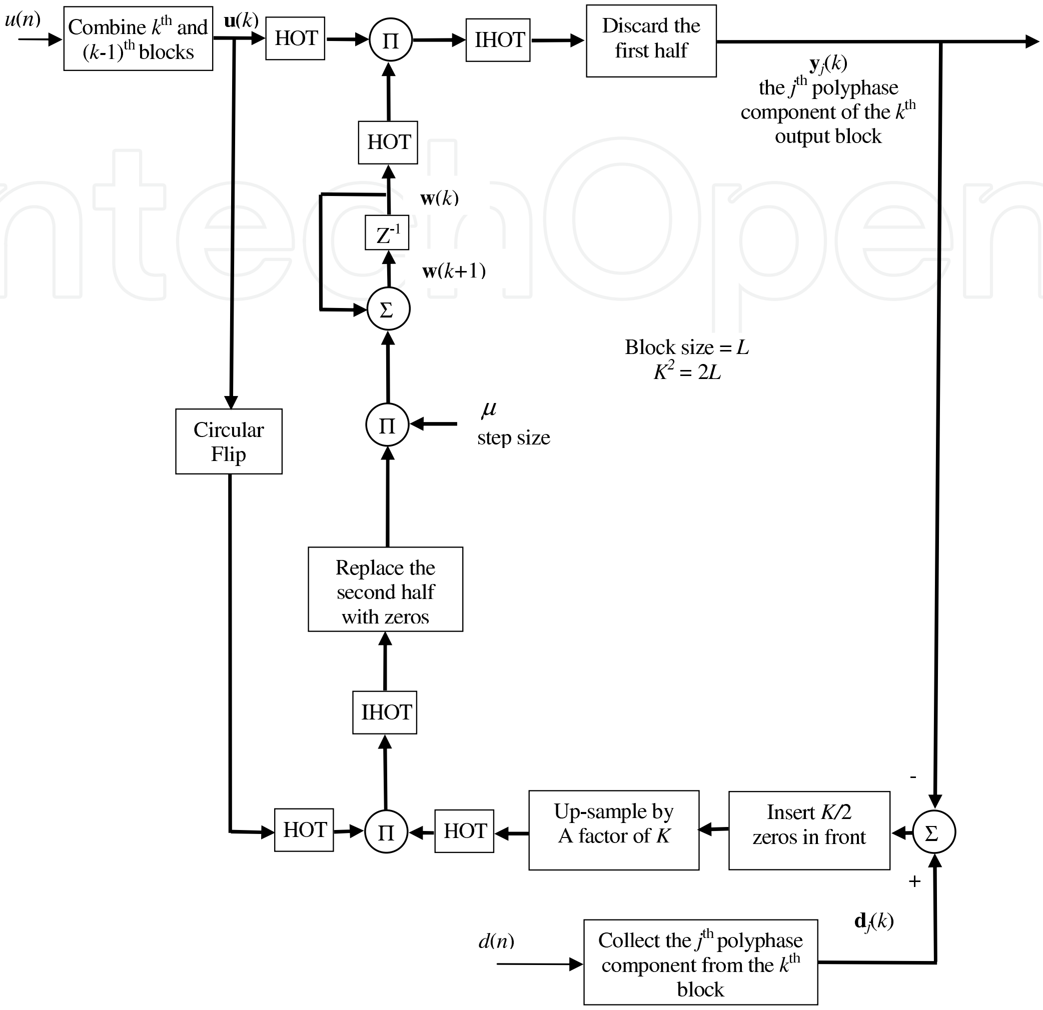
$u(n)$
Combine k^{th} and $(k\text{-}1)^{th}$ blocks
$\mathbf{u}(k)$
HOT
Π
IHOT
Discard the first half
$\mathbf{y}_j(k)$
the j^{th} polyphase component of the k^{th} output block
HOT
$\mathbf{w}(k)$
Z^{-1}
$\mathbf{w}(k+1)$
Σ
Block size = L
$K^2 = 2L$
Circular Flip
Π
μ
step size
Replace the second half with zeros
IHOT
-
HOT
Π
HOT
Up-sample by A factor of K
Insert $K/2$ zeros in front
Σ
+
$\mathbf{d}_j(k)$
$d(n)$
Collect the j^{th} polyphase component from the k^{th} block

Figure 2. HOT block LMS adaptive filter.

LMS adaptive filter and the asymptotic ratio between their computational cost is almost reached at small filter lengths. The computational complexity of the HOT filter can be further improved by relating the HOT of the circularly flipped vector in step (e) to the HOT of the vector in step (b). Another possibility to reduce the computational cost of the HOT block algorithm is by removing the gradient constraint in the filter weight update equation as has been done in the unconstrained DFT block LMS algorithm [9].

6. Convergence analysis in the time domain

In this section, we analyze the convergence of the HOT block LMS algorithm in the time domain. We assume throughout that the step size is small. The HOT block LMS filter minimizes the cost

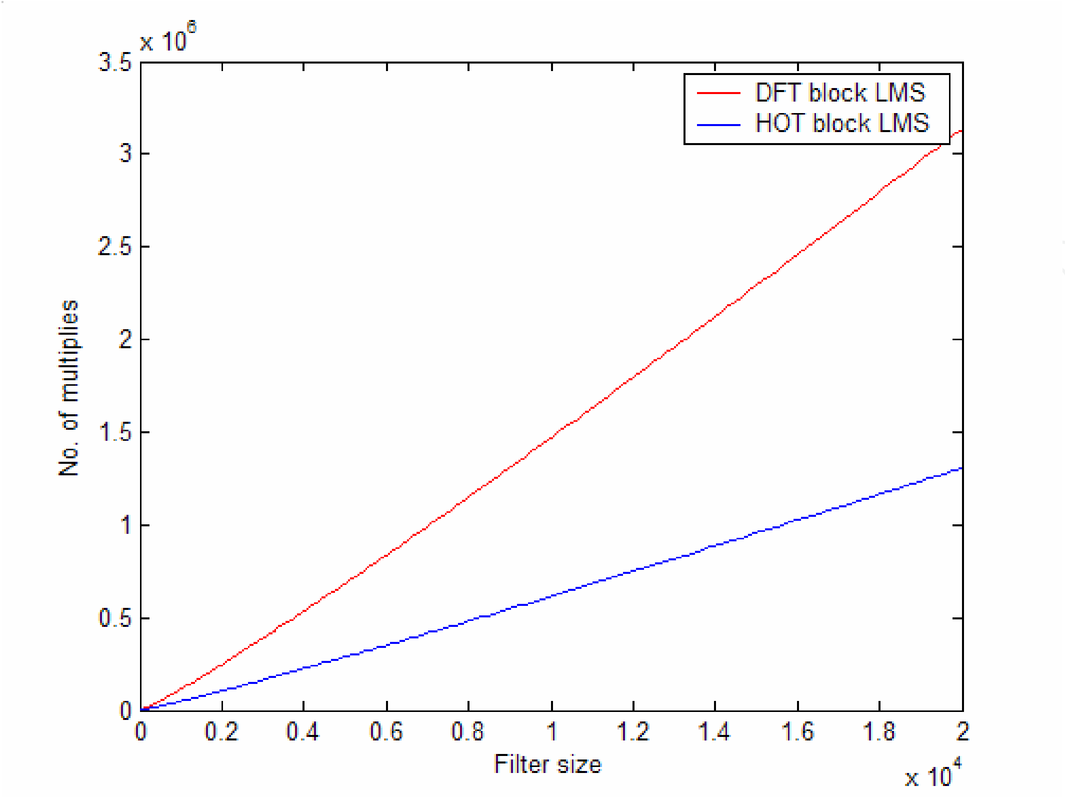

Figure 3. Multiplication counts for both the DFT block and HOT block LMS algorithms.

$$\hat{\xi} = \frac{2}{K} \sum_{i=0}^{\frac{K}{2}-1} \left| e(kL + iK + j) \right|^2, \tag{29}$$

which is the average of the squared errors in the j^{th} polyphase error component. From statistical LMS theory [10], the block LMS algorithm can be analyzed using the stochastic difference equation [10]

$$\boldsymbol{\epsilon}_T(k+1) = \Big(\mathbf{I} - \mu \boldsymbol{\Lambda} \Big) \boldsymbol{\epsilon}_T(k) + \boldsymbol{\phi}(k), \tag{30}$$

where

$$\boldsymbol{\phi}(k) = -\frac{\mu}{L} \mathbf{V}^H \sum_{i=0}^{L-1} \mathbf{u}(kL+i)\, e^o(kL+i) \tag{31}$$

is the driving force of for the block LMS algorithm [10]. we found that the HOT block LMS algorithm has the following driving force

$$\phi_{\mathrm{HOT}}(k) = -\frac{2\mu}{K}\,\mathbf{V}^H \sum_{i=0}^{\frac{K}{2}-1} \mathbf{u}(kL+iK+j)\,e^o(kL+iK+j). \tag{32}$$

It is easily shown that

$$E\phi_{\mathrm{HOT}}(k) = 0, \tag{33}$$

$$E\phi_{\mathrm{HOT}}(k)\phi_{\mathrm{HOT}}^H(k) = \frac{2\mu^2 J_{\min}\mathbf{\Lambda}}{K}. \tag{34}$$

The mean square of the l^{th} component of equation (34) is given by

$$E\,|\epsilon_l(k)|^2 = \frac{2\mu\frac{J_{\min}}{K}}{2-\mu\lambda_l} + (1-\mu\lambda_l)^{2k}\left(|\epsilon_l(0)|^2 - \frac{2\mu\frac{J_{min}}{K}}{2-\mu\lambda_l}\right), \tag{35}$$

where λ_l is the l^{th} eigenvalue of the input autocorrelation matrix. Therefore, the average time constant of the HOT block LMS algorithm is given by

$$\tau = \frac{L^2}{2\mu\sum_{l=1}^{L}\lambda_l}. \tag{36}$$

The misadjustment can be calculated directly and is given by

$$M = \frac{\sum_{l=1}^{L}\lambda_l E\,|\epsilon_l(\infty)|^2}{J_{\min}}. \tag{37}$$

Using equation (30), one may find $E|\epsilon_l(\infty)|^2$ and substitute the result into equation (37). The misadjustment of the HOT block LMS filter is then given by

$$M = \frac{\mu}{K}\sum_{l=1}^{L}\lambda_l. \tag{38}$$

Thus, the average time constant of the HOT block LMS filter is the same as that of the DFT block LMS filter [1]. However, the HOT block LMS filter has K times higher misadjustment than the DFT block LMS algorithm [2].

The HOT and DFT block LMS algorithms were simulated using white noise inputs. The desired signal was generated using the linear model $d(n) = w^o(n) * u(n) + e^o(n)$, where $e^o(n)$ is the measurement white gaussian noise with variance 10^{-4} and $W^o(z) = 1 + 0.5z^{-1} -$

[1] The average time constant of the DFT block LMS filter is [10] $\tau = L^2/2\mu\sum_{l=1}^{L}\lambda_l$.

[2] The misadjustment of the DFT block LMS algorithm is [10] $M = \frac{\mu}{K^2}\sum_{l=1}^{L}\lambda_l$.

$0.25z^{-2} + 0.03z^{-3} + 0.1z^{-4} + 0.002z^{-5} - 0.01z^{-6} + 0.007z^{-7}$. The learning curves are shown in Figure 4 with the learning curve of the conventional LMS algorithm. The step sizes of all algorithms were chosen to be the same. The higher mean square error of the HOT algorithm, compared to the DFT algorithm, shows the trade-off for complexity reduction by more than half. As expected the HOT and DFT block LMS algorithms converge at the same rate.

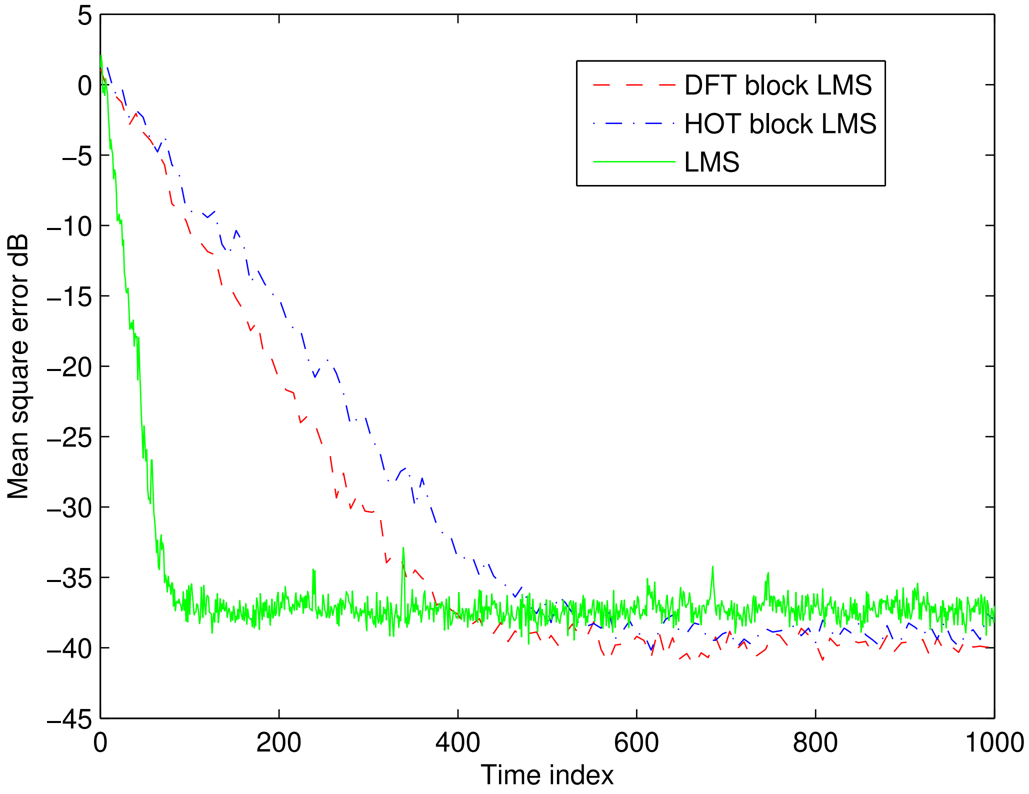

Figure 4. Learning curves of the DFT and HOT block LMS algorithms with the conventional LMS filter.

7. Convergence analysis in the HOT domain

Let $u(n)$ be the input to the adaptive filter and

$$\hat{\mathbf{w}}(k) = \left[w_0(k) \ w_1(k) \ \cdots \ w_{\frac{K^2}{2}-1}(k) \right]^T \tag{39}$$

be the tap-weight vector of the adaptive filter, where k is the block index. Define the extended tap-weight vector

$$\mathbf{w}(k) = \left[\hat{\mathbf{w}}^T(k) \ 0 \ 0 \cdots 0 \right]^T \tag{40}$$

and the tap-input vector

$$\mathbf{u}(k) = \left[u\left((k-1)\tfrac{K^2}{2}\right) \cdots u\left(k\tfrac{K^2}{2}\right) \; u\left(k\tfrac{K^2}{2}+1\right) \cdots u\left((k+1)\tfrac{K^2}{2}-1\right) \right]^T. \tag{41}$$

Denote the HOT transforms of $\mathbf{u}(k)$ and $\mathbf{w}(k)$ by $\mathbf{u}_H(k) = \mathbf{H}\mathbf{u}(k)$ and $\mathbf{w}_H(k) = \mathbf{H}\mathbf{w}(k)$, respectively, where $\mathbf{H}$ is the HOT matrix. The 0^{th} polyphase component of the circular convolution of $\mathbf{u}(k)$ and $\mathbf{w}(k)$ is given by

$$\mathbf{F}_K\mathbf{y}_0(k) = \mathbf{F}_K\mathbf{w}_0(k) \otimes \mathbf{F}_K\mathbf{u}_0(k) + \mathbf{D}\sum_{i=1}^{K-1} \mathbf{F}_K\mathbf{w}_{K-i}(k) \otimes \mathbf{F}_K\mathbf{u}_i(k). \tag{42}$$

Using $\mathbf{F}_K\mathbf{u}_i(k) = \mathbf{I}_i\mathbf{H}\mathbf{u}(k) = \mathbf{I}_i\mathbf{u}_H(k)$, equation (42) can be written in terms of the HOT of $\mathbf{u}$(k) and $\mathbf{w}$(k). The result is given by

$$\mathbf{F}_K\mathbf{y}_0(k) = \mathbf{I}_0\mathbf{w}_H(k) \otimes \mathbf{I}_0\mathbf{u}_H(k) + \mathbf{D}\sum_{i=1}^{K-1} \mathbf{I}_{K-i}\mathbf{w}_H(k) \otimes \mathbf{I}_i\mathbf{u}_H(k). \tag{43}$$

The 0^{th} polyphase component of the linear convolution of $\hat{\mathbf{w}}(k)$ and $\mathbf{u}$(n), the output of the adaptive filter in the k^{th} block, is given by the last $K/2$ elements of $\mathbf{y}_0(k)$. Let the desired signal be $d(n)$ and define the extended 0^{th} polyphase component of the desired signal in the k^{th} block as

$$\mathbf{d}_0(k) = \begin{bmatrix} \mathbf{0}_{\frac{K}{2}} \\ \hat{\mathbf{d}}_0(k) \end{bmatrix}. \tag{44}$$

The extended 0^{th} polyphase component of error signal in the k^{th} block is given by

$$\mathbf{e}_0(k) = \begin{bmatrix} \mathbf{0}_{\frac{K}{2}} \\ \hat{\mathbf{e}}_0(k) \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{\frac{K}{2}} \\ \hat{\mathbf{d}}_0(k) \end{bmatrix} - \begin{bmatrix} \mathbf{0}_{\frac{K}{2}\times\frac{K}{2}} & \mathbf{0}_{\frac{K}{2}\times\frac{K}{2}} \\ \mathbf{0}_{\frac{K}{2}\times\frac{K}{2}} & \mathbf{I}_{\frac{K}{2}\times\frac{K}{2}} \end{bmatrix} \mathbf{F}_K^{-1}$$

$$\times \left[\mathbf{I}_0\mathbf{w}_H(k) \otimes \mathbf{I}_0\mathbf{u}_H(k) + \mathbf{D}\sum_{i=1}^{K-1} \mathbf{I}_{K-i}\mathbf{w}_H(k) \otimes \mathbf{I}_i\mathbf{u}_H(k) \right]. \tag{45}$$

Multiplying equation (45) by the DFT matrix yields

$$\mathbf{F}_K\mathbf{e}_0(k) = \mathbf{F}_K \begin{bmatrix} \mathbf{0}_{\frac{K}{2}} \\ \hat{\mathbf{d}}_0(k) \end{bmatrix} - \mathbf{F}_K \begin{bmatrix} \mathbf{0}_{\frac{K}{2}\times\frac{K}{2}} & \mathbf{0}_{\frac{K}{2}\times\frac{K}{2}} \\ \mathbf{0}_{\frac{K}{2}\times\frac{K}{2}} & \mathbf{I}_{\frac{K}{2}\times\frac{K}{2}} \end{bmatrix} \mathbf{F}_K^{-1}$$

$$\times \left[\mathbf{I}_0\mathbf{w}_H(k) \otimes \mathbf{I}_0\mathbf{u}_H(k) + \mathbf{D}\sum_{i=1}^{K-1} \mathbf{I}_{K-i}\mathbf{w}_H(k) \otimes \mathbf{I}_i\mathbf{u}_H(k) \right]. \tag{46}$$

Define $\mathbf{u}_H^c(k) = \mathbf{H}\mathbf{u}^c(k)$, where $\mathbf{u}^c(k)$ is the circularly shifted version of $\mathbf{u}(k)$. The adaptive filter update equation in the k^{th} block is given by

$$\mathbf{w}_H(k+1) = \mathbf{w}_H(k) + \mu\,\mathbf{H}\begin{bmatrix} \mathbf{I}_{\frac{K^2}{2}\times\frac{K^2}{2}} & \mathbf{0}_{\frac{K^2}{2}\times\frac{K^2}{2}} \\ \mathbf{0}_{\frac{K^2}{2}\times\frac{K^2}{2}} & \mathbf{0}_{\frac{K^2}{2}\times\frac{K^2}{2}} \end{bmatrix}\mathbf{H}^{-1}\phi_H(k), \tag{47}$$

where $\phi_H(k)$ is found from

$$\begin{bmatrix} \mathbf{I}_0\phi_H(k) \\ \mathbf{I}_1\phi_H(k) \\ \mathbf{I}_2\phi_H(k) \\ \vdots \\ \mathbf{I}_{K-2}\phi_H(k) \\ \mathbf{I}_{K-1}\phi_H(k) \end{bmatrix} = \begin{bmatrix} \mathbf{F}_K\mathbf{e}_0(k) \\ \mathbf{F}_K\mathbf{e}_0(k) \\ \mathbf{F}_K\mathbf{e}_0(k) \\ \vdots \\ \mathbf{F}_K\mathbf{e}_0(k) \\ \mathbf{F}_K\mathbf{e}_0(k) \end{bmatrix} \otimes \begin{bmatrix} \mathbf{I}_0\mathbf{u}_H^c(k) \\ \mathbf{I}_1\mathbf{u}_H^c(k) \\ \mathbf{I}_2\mathbf{u}_H^c(k) \\ \vdots \\ \mathbf{I}_{K-2}\mathbf{u}_H^c(k) \\ \mathbf{I}_{K-1}\mathbf{u}_H^c(k) \end{bmatrix}, \tag{48}$$

as

$$\phi_H(k) = \mathbf{I}_K^{-1}\begin{bmatrix} \mathbf{F}_K\mathbf{e}_0(k) \\ \mathbf{F}_K\mathbf{e}_0(k) \\ \mathbf{F}_K\mathbf{e}_0(k) \\ \vdots \\ \mathbf{F}_K\mathbf{e}_0(k) \\ \mathbf{F}_K\mathbf{e}_0(k) \end{bmatrix} \otimes \begin{bmatrix} \mathbf{I}_0\mathbf{u}_H^c(k) \\ \mathbf{I}_1\mathbf{u}_H^c(k) \\ \mathbf{I}_2\mathbf{u}_H^c(k) \\ \vdots \\ \mathbf{I}_{K-2}\mathbf{u}_H^c(k) \\ \mathbf{I}_{K-1}\mathbf{u}_H^c(k) \end{bmatrix}. \tag{49}$$

Finally, the HOT block LMS filter in the HOT domain can be written as

$$\begin{aligned} \mathbf{w}_H(k+1) &= \mathbf{w}_H(k) \\ &+ \mu\,\mathbf{H}\begin{bmatrix} \mathbf{I}_{\frac{K^2}{2}\times\frac{K^2}{2}} & \mathbf{0}_{\frac{K^2}{2}\times\frac{K^2}{2}} \\ \mathbf{0}_{\frac{K^2}{2}\times\frac{K^2}{2}} & \mathbf{0}_{\frac{K^2}{2}\times\frac{K^2}{2}} \end{bmatrix}\mathbf{H}^{-1}\mathbf{I}_K^{-1}\begin{bmatrix} \mathbf{F}_K\mathbf{e}_0(k) \\ \mathbf{F}_K\mathbf{e}_0(k) \\ \mathbf{F}_K\mathbf{e}_0(k) \\ \vdots \\ \mathbf{I}_{K-1}\mathbf{e}_0(k) \\ \mathbf{I}_{K-1}\mathbf{e}_0(k) \end{bmatrix} \otimes \begin{bmatrix} \mathbf{I}_0\mathbf{u}_H^c(k) \\ \mathbf{I}_1\mathbf{u}_H^c(k) \\ \mathbf{I}_2\mathbf{u}_H^c(k) \\ \vdots \\ \mathbf{I}_{K-2}\mathbf{u}_H^c(k) \\ \mathbf{I}_{K-1}\mathbf{u}_H^c(k) \end{bmatrix}. \end{aligned} \tag{50}$$

Next, we investigate the convergence properties of equation (50). we assume the following linear statistical model for the desired signal:

$$d(n) = w^o(n) * u(n) + e^o(n), \tag{51}$$

where w^o is the impulse response of the Wiener optimal filter and $e^o(n)$ is the irreducible estimation error, which is white noise and statistically independent of the adaptive filter input. The above equation can be written in the HOT domain form

$$\begin{bmatrix} \mathbf{0}_{\frac{K}{2}} \\ \hat{\mathbf{d}}_0(k) \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{\frac{K}{2}\times\frac{K}{2}} & \mathbf{0}_{\frac{K}{2}\times\frac{K}{2}} \\ \mathbf{0}_{\frac{K}{2}\times\frac{K}{2}} & \mathbf{I}_{\frac{K}{2}\times\frac{K}{2}} \end{bmatrix} \mathbf{F}_K^{-1}$$

$$\times \left[\mathbf{I}_0 \mathbf{w}_H^o(k) \otimes \mathbf{I}_0 \mathbf{u}_H(k) + \mathbf{D} \sum_{i=1}^{K-1} \mathbf{I}_{K-i} \mathbf{w}_H^o(k) \otimes \mathbf{I}_i \mathbf{u}_H(k) + \mathbf{F}_K \mathbf{e}_0^o(k) \right]. \quad (52)$$

This form will be useful to obtain the stochastic difference equation that describes the convergence of the adaptive algorithm. Using the above equation to replace the desired signal in equation (46), we have

$$\mathbf{F}_K \mathbf{e}_0(k) = \mathbf{F}_K \begin{bmatrix} \mathbf{0}_{\frac{K}{2}\times\frac{K}{2}} & \mathbf{0}_{\frac{K}{2}\times\frac{K}{2}} \\ \mathbf{0}_{\frac{K}{2}\times\frac{K}{2}} & \mathbf{I}_{\frac{K}{2}\times\frac{K}{2}} \end{bmatrix} \mathbf{F}_K^{-1}$$

$$\times \left[\mathbf{I}_0 \boldsymbol{\epsilon}_H(k) \otimes \mathbf{I}_0 \mathbf{u}_H(k) + \mathbf{D} \sum_{i=1}^{K-1} \mathbf{I}_{K-i} \boldsymbol{\epsilon}_H(k) \otimes \mathbf{I}_i \mathbf{u}_H(k) + \mathbf{F}_K \mathbf{e}_0^o(k) \right], \quad (53)$$

where $\boldsymbol{\epsilon}_H(k)$ is the error in the estimation of the adaptive filter weight vector, i.e., $\boldsymbol{\epsilon}_H(k) = \mathbf{w}_H^o - \mathbf{w}_H(k)$. The i^{th} block in equation (50) is given by

$$\mathbf{F}_K \mathbf{e}_0(k) \otimes \mathbf{I}_i \mathbf{u}_H^c(k) = \text{Diag}\left[\mathbf{I}_i \mathbf{u}_H^c(k)\right] \mathbf{F}_K \mathbf{e}_0(k). \quad (54)$$

Substituting equation (53) into equation (54) yields

$$\mathbf{F}_K \mathbf{e}_0(k) \otimes \mathbf{I}_i \mathbf{u}_H^c(k) = \text{Diag}\left[\mathbf{I}_i \mathbf{u}_H^c(k)\right] \mathbf{F}_K \begin{bmatrix} \mathbf{0}_{\frac{K}{2}\times\frac{K}{2}} & \mathbf{0}_{\frac{K}{2}\times\frac{K}{2}} \\ \mathbf{0}_{\frac{K}{2}\times\frac{K}{2}} & \mathbf{I}_{\frac{K}{2}\times\frac{K}{2}} \end{bmatrix} \mathbf{F}_K^{-1} \times$$

$$\left[\text{Diag}\left[\mathbf{I}_0 \mathbf{u}_H(k)\right] \mathbf{I}_0 \boldsymbol{\epsilon}_H(k) + \mathbf{D} \sum_{i=1}^{K-1} \text{Diag}\left[\mathbf{I}_{K-i} \mathbf{u}_H(k)\right] \mathbf{I}_i \boldsymbol{\epsilon}_H(k) + \mathbf{F}_K \mathbf{e}^o(k) \right]. \quad (55)$$

Upon defining

$$\mathbf{T}_{i,j} = \text{Diag}\left[\mathbf{I}_i \mathbf{u}_H^c(k)\right] \mathbf{L}_K \text{Diag}\left[\mathbf{I}_j \mathbf{u}_H(k)\right], \quad (56)$$

where

$$\mathbf{L}_K = \mathbf{F}_K \begin{bmatrix} \mathbf{0}_{\frac{K}{2}\times\frac{K}{2}} & \mathbf{0}_{\frac{K}{2}\times\frac{K}{2}} \\ \mathbf{0}_{\frac{K}{2}\times\frac{K}{2}} & \mathbf{I}_{\frac{K}{2}\times\frac{K}{2}} \end{bmatrix} \mathbf{F}_K^{-1}, \tag{57}$$

the i^{th} block of equation (50) can be written as

$$\begin{aligned} \mathbf{F}_K \mathbf{e}^o(k) \otimes \mathbf{I}_i \mathbf{u}_H^c(k) = {} & \begin{bmatrix} \mathbf{T}_{i,0} & \mathbf{T}_{i,K-1} & \mathbf{T}_{i,K-2} & \cdots & \mathbf{T}_{i,1} \end{bmatrix} \begin{bmatrix} \mathbf{I}_0 \boldsymbol{\epsilon}_H(k) \\ \mathbf{D}\mathbf{I}_1 \boldsymbol{\epsilon}_H(k) \\ \mathbf{D}\mathbf{I}_2 \boldsymbol{\epsilon}_H(k) \\ \vdots \\ \mathbf{D}\mathbf{I}_{K-1} \boldsymbol{\epsilon}_H(k) \end{bmatrix} \\ & + \mathrm{Diag}\left[\mathbf{I}_i \mathbf{u}_H^c(k)\right] \mathbf{L}_K \mathbf{e}^o(k). \end{aligned} \tag{58}$$

Using the fact that

$$\mathrm{Diag}\,[\mathbf{v}]\,\mathbf{R}\,\mathrm{Diag}\,[\mathbf{u}] = \left(\mathbf{v}\mathbf{u}^T\right) \otimes \mathbf{R}, \tag{59}$$

equation (56) can be written as

$$\mathbf{T}_{i,j} = \left(\mathbf{I}_i \mathbf{u}_H^c(k) \left(\mathbf{I}_j \mathbf{u}_H(k)\right)^T\right) \otimes \mathbf{L}_K. \tag{60}$$

Define

$$\mathbf{U}_{K^2} = \mathbf{H} \begin{bmatrix} \mathbf{I}_{\frac{K^2}{2}\times\frac{K^2}{2}} & \mathbf{0}_{\frac{K^2}{2}\times\frac{K^2}{2}} \\ \mathbf{0}_{\frac{K^2}{2}\times\frac{K^2}{2}} & \mathbf{0}_{\frac{K^2}{2}\times\frac{K^2}{2}} \end{bmatrix} \mathbf{H}^{-1}. \tag{61}$$

Then

$$\begin{aligned} \mathbf{w}_H(k+1) = {} & \mathbf{w}_H(k) \\ & + \mu\, \mathbf{U}_{K^2} \mathbf{I}_K^{-1} \mathbf{T} \begin{bmatrix} \mathbf{I}_0 \boldsymbol{\epsilon}_H(k) \\ \mathbf{D}\mathbf{I}_1 \boldsymbol{\epsilon}_H(k) \\ \mathbf{D}\mathbf{I}_2 \boldsymbol{\epsilon}_H(k) \\ \vdots \\ \mathbf{D}\mathbf{I}_{K-1} \boldsymbol{\epsilon}_H(k) \end{bmatrix} + \mu\, \mathbf{U}_{K^2} \mathbf{I}_K^{-1} \begin{bmatrix} \mathrm{Diag}\left[\mathbf{I}_0 \mathbf{u}_H^c(k)\right] \\ \mathrm{Diag}\left[\mathbf{I}_1 \mathbf{u}_H^c(k)\right] \\ \mathrm{Diag}\left[\mathbf{I}_2 \mathbf{u}_H^c(k)\right] \\ \vdots \\ \mathrm{Diag}\left[\mathbf{I}_{K-1} \mathbf{u}_H^c(k)\right] \end{bmatrix} \mathbf{L}_K \mathbf{e}^o(k). \end{aligned} \tag{62}$$

The matrix $\mathbf{T}$ can be written as

$$\begin{aligned} \mathbf{T} = {} & \left(\mathbf{1}_{K\times K} \times \mathbf{L}_K\right) \otimes \\ & \begin{bmatrix} \mathbf{I}_0 \mathbf{u}_H^c(k)\left[\mathbf{I}_0 \mathbf{u}_H(k)\right]^T & \mathbf{I}_0 \mathbf{u}_H^c(k)\left[\mathbf{I}_{K-1} \mathbf{u}_H(k)\right]^T & \cdots & \mathbf{I}_0 \mathbf{u}_H^c(k)\left[\mathbf{I}_1 \mathbf{u}_H(k)\right]^T \\ \mathbf{I}_1 \mathbf{u}_H^c(k)\left[\mathbf{I}_0 \mathbf{u}_H(k)\right]^T & \mathbf{I}_1 \mathbf{u}_H^c(k)\left[\mathbf{I}_{K-1} \mathbf{u}_H(k)\right]^T & \cdots & \mathbf{I}_1 \mathbf{u}_H^c(k)\left[\mathbf{I}_1 \mathbf{u}_H(k)\right]^T \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{I}_{K-1} \mathbf{u}_H^c(k)\left[\mathbf{I}_0 \mathbf{u}_H(k)\right]^T & \mathbf{I}_{K-1} \mathbf{u}_H^c(k)\left[\mathbf{I}_{K-1} \mathbf{u}_H(k)\right]^T & \cdots & \mathbf{I}_{K-1} \mathbf{u}_H^c(k)\left[\mathbf{I}_1 \mathbf{u}_H(k)\right]^T \end{bmatrix}, \end{aligned}$$

where $\times$ denotes the Kronecker product and $\mathbf{1}_{K\times K}$ is the $K \times K$ matrix with all element being equal to one. The matrix $\mathbf{T}$ can be written as

$$\mathbf{T} = \begin{bmatrix} \mathbf{I}_0\mathbf{u}_H^c(k) \\ \mathbf{I}_1\mathbf{u}_H^c(k) \\ \vdots \\ \mathbf{I}_{K-2}\mathbf{u}_H^c(k) \\ \mathbf{I}_{K-1}\mathbf{u}_H^c(k) \end{bmatrix} \begin{bmatrix} \mathbf{I}_0\mathbf{u}_H(k) \\ \mathbf{I}_{K-1}\mathbf{u}_H(k) \\ \vdots \\ \mathbf{I}_2\mathbf{u}_H(k) \\ \mathbf{I}_1\mathbf{u}_H(k) \end{bmatrix}^T \otimes \left(\mathbf{1}_{K\times K} \times \mathbf{L}_K\right) = \left(\mathbf{I}_K\mathbf{u}_H^c(k)\mathbf{u}_H^T(k)\mathbf{I}_K^{c\,T}\right) \otimes \left(\mathbf{1}_{K\times K} \times \mathbf{L}_K\right),$$

where

$$\mathbf{I}_K^c = \begin{bmatrix} \mathbf{I}_0 \\ \mathbf{I}_{K-1} \\ \vdots \\ \mathbf{I}_1 \end{bmatrix}. \tag{63}$$

Finally, the error in the estimation of the adaptive filter is given by

$$\begin{aligned} \boldsymbol{\epsilon}_H(k+1) = {} & \left(\mathbf{I} - \mu\mathbf{U}_{K^2}\mathbf{I}_K^{-1}\left(\mathbf{I}_K\mathbf{u}_H^c(k)\mathbf{u}_H^T(k)\mathbf{I}_K^{c\,T}\right) \otimes \left(\mathbf{1}_{K\times K} \times \mathbf{L}_K\right)\mathbf{I}_K^D\right)\boldsymbol{\epsilon}_H(k) \\ & -\mu\mathbf{U}_{K^2}\mathbf{I}_K^{-1} \begin{bmatrix} \text{Diag}[\mathbf{I}_0\mathbf{u}_H^c(k)] \\ \text{Diag}[\mathbf{I}_1\mathbf{u}_H^c(k)] \\ \vdots \\ \text{Diag}[\mathbf{I}_{K-2}\mathbf{u}_H^c(k)] \\ \text{Diag}[\mathbf{I}_{K-1}\mathbf{u}_H^c(k)] \end{bmatrix} \mathbf{L}_K\mathbf{e}^o(k), \end{aligned} \tag{64}$$

where

$$\mathbf{I}_K^D = \begin{bmatrix} \mathbf{I}_0 \\ \mathbf{DI}_1 \\ \mathbf{DI}_2 \\ \vdots \\ \mathbf{DI}_{K-2} \\ \mathbf{DI}_{K-1} \end{bmatrix}. \tag{65}$$

Therefore, the adaptive block HOT filter convergence is governed by the matrix

$$\boldsymbol{\Psi} = \mathbf{H} \begin{bmatrix} \mathbf{I}_{\frac{K^2}{2}\times\frac{K^2}{2}} & \mathbf{0}_{\frac{K^2}{2}\times\frac{K^2}{2}} \\ \mathbf{0}_{\frac{K^2}{2}\times\frac{K^2}{2}} & \mathbf{0}_{\frac{K^2}{2}\times\frac{K^2}{2}} \end{bmatrix} \mathbf{H}^{-1}\mathbf{I}_K^{-1}\left(\mathbf{I}_K E\mathbf{u}_H^c(k)\mathbf{u}_H^T(k)\mathbf{I}_K^{cT}\right) \otimes \left(\mathbf{1}_{K\times K} \times \mathbf{L}_K\right)\mathbf{I}_K^D. \tag{66}$$

The structure of $\boldsymbol{\Psi}$ is now analyzed. Using the relation between the HOT and the DFT transforms, we can write

$$\mathbf{I}_K\mathbf{u}_H^c = \begin{bmatrix} \mathbf{F}_K\mathbf{u}_0^c \\ \mathbf{F}_K\mathbf{u}_1^c \\ \vdots \\ \mathbf{F}_K\mathbf{u}_{K-2}^c \\ \mathbf{F}_K\mathbf{u}_{K-1}^c \end{bmatrix}. \tag{67}$$

It can be easily shown that

$$\mathbf{F}_K\mathbf{u}_i^c = \begin{cases} \mathbf{F}_K^H\mathbf{u}_i & \text{if } i = 0, \\ \mathbf{D}^*\mathbf{F}_K^H\mathbf{u}_{K-i} & \text{if } i \neq 0. \end{cases} \tag{68}$$

Then we have

$$\mathbf{I}_K\mathbf{u}_K^c = \begin{bmatrix} \mathbf{F}_K^H\mathbf{u}_0 \\ \mathbf{D}^*\mathbf{F}_K^H\mathbf{u}_{K-1} \\ \vdots \\ \mathbf{D}^*\mathbf{F}_K^H\mathbf{u}_2 \\ \mathbf{D}^*\mathbf{F}_K^H\mathbf{u}_1 \end{bmatrix} \tag{69}$$

and

$$\mathbf{I}_K\mathbf{u}_H^c(k)\mathbf{u}_H^T(k)\mathbf{I}_K^{cT} = \begin{bmatrix} \mathbf{F}_K^H\mathbf{u}_0 \\ \mathbf{D}^*\mathbf{F}_K^H\mathbf{u}_{K-1} \\ \vdots \\ \mathbf{D}^*\mathbf{F}_K^H\mathbf{u}_2 \\ \mathbf{D}^*\mathbf{F}_K^H\mathbf{u}_1 \end{bmatrix} \begin{bmatrix} \mathbf{F}_K\mathbf{u}_0 \\ \mathbf{F}_K\mathbf{u}_{K-1} \\ \vdots \\ \mathbf{F}_K\mathbf{u}_2 \\ \mathbf{F}_K\mathbf{u}_1 \end{bmatrix}^T. \tag{70}$$

Taking the expectation of equation (70) yields

$$\mathbf{I}_K E\mathbf{u}_H^c(k)\mathbf{u}_H^T(k)\mathbf{I}_K^{cT} = \begin{bmatrix} \mathbf{F}_K^H E\mathbf{u}_0\mathbf{u}_0^T\mathbf{F}_K & \mathbf{F}_K^H E\mathbf{u}_0\mathbf{u}_{K-1}^T\mathbf{F}_K & \dots & \mathbf{F}_K^H E\mathbf{u}_0\mathbf{u}_1^T\mathbf{F}_K \\ \mathbf{D}^*\mathbf{F}_K^H E\mathbf{u}_{K-1}\mathbf{u}_0^T\mathbf{F}_K & \mathbf{D}^*\mathbf{F}_K^H E\mathbf{u}_{K-1}\mathbf{u}_{K-1}^T\mathbf{F}_K & \dots & \mathbf{D}^*\mathbf{F}_K^H E\mathbf{u}_{K-1}\mathbf{u}_1^T\mathbf{F}_K \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{D}^*\mathbf{F}_K^H E\mathbf{u}_1\mathbf{u}_0^T\mathbf{F}_K & \mathbf{D}^*\mathbf{F}_K^H E\mathbf{u}_1\mathbf{u}_{K-1}^T\mathbf{F}_K & \dots & \mathbf{D}^*\mathbf{F}_K^H E\mathbf{u}_1\mathbf{u}_1^T\mathbf{F}_K \end{bmatrix}.$$

Each block in the above equation is an autocorrelation matrix that is asymptotically diagonalized by the DFT matrix. Each block will be also pointwise multiplied by $\mathbf{L}_K$. Three-dimensional representations of $\mathbf{L}_K$ for $K = 16$ and $K = 32$ are shown in Figures 5 and 6, respectively. The diagonal elements of $\mathbf{L}_K$ are much higher than the off diagonal elements. Therefore, pointwise multiplying each block in the previous equation with $\mathbf{L}_K$ makes it more diagonal. If each block is perfectly diagonal, then $\mathbf{I}_K\left(\mathbf{I}_K E\mathbf{u}_H^c(k)\mathbf{u}_H^T(k)\mathbf{I}_K^{cT}\right)\otimes(\mathbf{1}_{K\times K}\times\mathbf{L}_K)\mathbf{I}_K^D$ will be block diagonal. Asymptotically the HOT block LMS adaptive filter transforms the K^2 modes into K decoupled sets of modes.

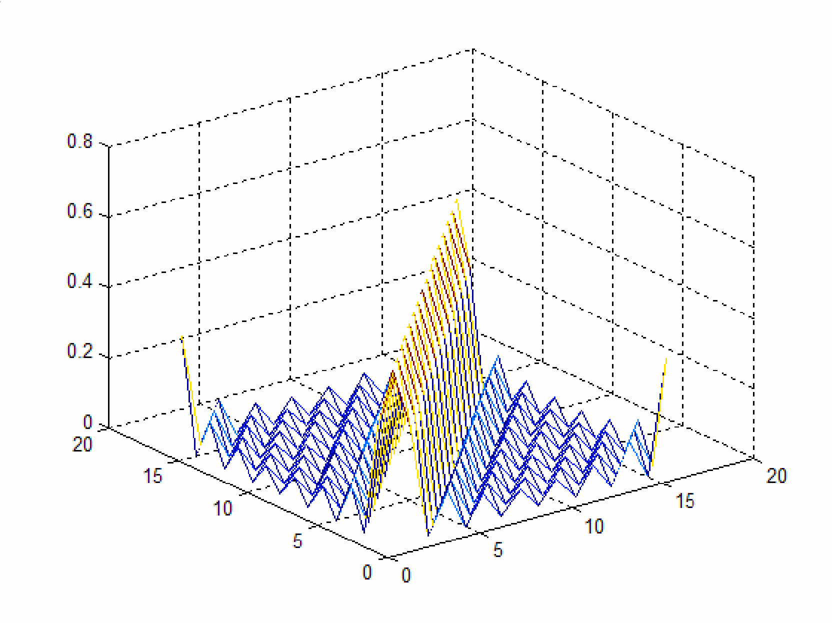

Figure 5. Three-dimensional representation of $\mathbf{L}_{16}$.

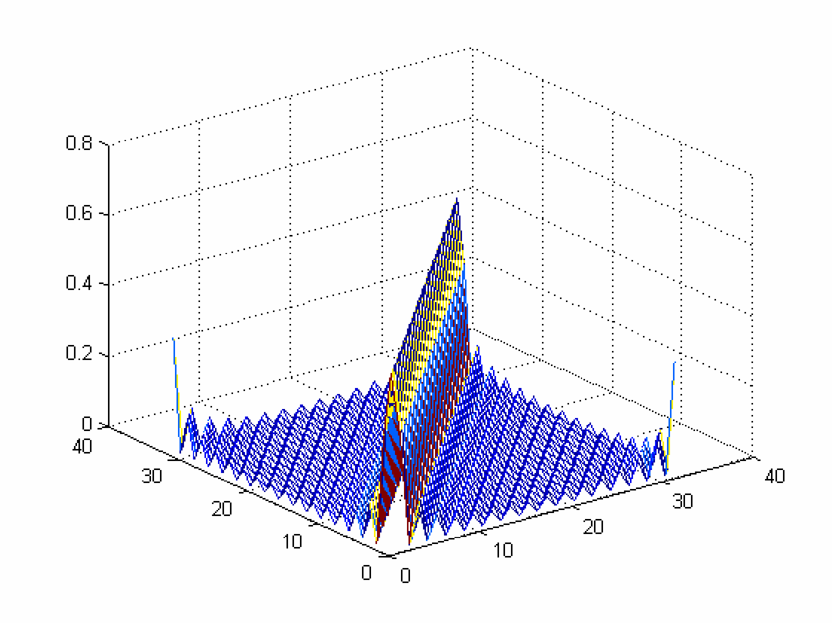

Figure 6. Three-dimensional representation of $\mathbf{L}_{32}$.

8. Conclusions

The "HOT convolution," a relation between the HOT of two signals and their circular convolution was derived. The result was used to develop a fast block LMS adaptive filter called the HOT block LMS adaptive filter. The HOT block LMS adaptive filter assumes that the filter and block lengths are the same. This filter requires slightly less than half of the multiplications that are required for the DFT block LMS adaptive filter. The reduction in the computational complexity of the HOT block LMS comes from using only one polyphase component of the filter error used to update the filter weights. Convergence analysis of the HOT block LMS algorithm showed that the average time constant is the same as that of the DFT block LMS algorithm and that the misadjustment is K times greater than that of the DFT block LMS algorithm. The HOT block LMS adaptive filter transforms the K^2 modes into K decoupled sets of modes.

Author details

Osama Alkhouli[1,⋆],
Victor DeBrunner[2] and Joseph Havlicek[3]

⋆ Address all correspondence to: osama_k@ou.edu

1 Caterpillar Inc., Illinois, USA
2 University of Oklahoma, School of Electrical and Computer Engineering, Oklahoma, USA
3 Florida State University, Electrical and Computer Engineering Department, Florida, USA

References

[1] I. I. Hirschman, "A note on entropy," *Amer. J. Math.*, vol. 79, pp. 152-156, 1957.

[2] H T. Przebinda, V. DeBrunner, and M. Özaydin, "The optimal transform for the discrete Hirschman uncertainty principle," *IEEE Trans. Infor. Theory*, pp. 2086-2090, Jul 2001.

[3] V. DeBrunner, M. Özaydin, and T. Przebinda, "Resolution in time-frequency," *IEEE Trans. ASSP*, pp. 783-788, Mar 1999.

[4] V. DeBrunner, J. Havlicek, T. Przebinda, and M. Özaydin, "Entropy-based uncertainty measures for $L^2(R)^n$, $\ell^2(Z)$, and $\ell^2(Z/NZ)$ with a Hirschman optimal transform for $\ell^2(Z/NZ)$," *IEEE Trans. ASSP*, pp. 2690-2696, August 2005.

[5] V. DeBrunner and E. Matusiak, "An algorithm to reduce the complexity required to convolve finite length sequences using the Hirschman optimal transform (HOT)," *ICASSP 2003*, Hong Kong, China, pp. II-577-580, Apr 2003.

[6] G. Clark, S. Mitra, and S Parker, "Block implementation of adaptive digital filters," *IEEE Trans. ASSP*, pp. 744-752, Jun 1981.

[7] E. R. Ferrara, "Fast implementation of LMS adaptive filters," *IEEE Trans. ASSP*, vol. ASSP-28, NO. 4, Aug 1980.

[8] J. Benesty and P. Duhamel, “A fast exact least mean square adaptive algorithm,” *IEEE Trans. ASSP*, pp. 2904-2920, Dec 1992.

[9] D. Mansour and A. H. Gray, “Unconstrained frequency-domain adaptive filter,” *IEEE Trans. ASSP*, pp. 726-734, Oct 1982.

[10] Simon Haykin, *Adaptive Filter Theory*. Prentice Hall information and system sciences series, Fourth edition, 2002.