Proceedings of the 12th
KACC, October 1997

고속하다마드 변환을 이용한 적응 필터의 구현

Implementation of Adaptive Filters Using Fast Hadamard Transform

° 곽대연*, 박진배*, 윤태성**
* 연세대학교 전기공학과(Tel:+82-2-361-2773; Fax:+82-2-392-4230; E-mail:jbpark@bubble.yonsei.ac.kr)
** 창원대학교 전기공학과(Tel:+82-551-79-7513; E-mail:tsyoon@sarim.changwon.ac.kr)

Abstracts We introduce a fast implementation of the adaptive transversal filter which uses least-mean-square(LMS) algorithm. The fast Hadamard transform(FHT) is used for the implementation of the filter. By using the proposed filter we can get the significant time reduction in computation over the conventional time domain LMS filter at the cost of a little performance. By computer simulation, we show the comparison of the propsed Hadamard-domain filter and the time domain filter in the view of multiplication time, mean-square error and robustness for noise.

Keywords Fast Hadamard Transform, Adaptive Filters, Least Mean Square(LMS) Algorithm, Controllability Parameter, Gradient Descent Method

1. 서론

적응필터는 잡음제거등 간섭신호의 소거, 선형예측, 시스템 동정과 통신에서의 channel equalization등, 여러 영역에서 쓰이는 중요한 신호처리 기기이다[5][6][8]. 일반적인 시간영역에서의 적응필터에서는 입력신호가 시간지연소자에 의해 지연되어 N개의 시퀀스(sequence)를 만들고 이들 각각에 가중치(weight)를 곱한 후 다시 더하여 출력 신호를 낸다. 이들 출력신호는 우리가 원하는 기준신호(desired signal)와의 비교를 통해 오차를 얻고 이 오차를 이용하여 가중치를 변화시키는 방법을 이용한다. 이때 주로 최소평균자승오차(LMS)기법이 사용된다[9].

본 논문에서는 가장 효과적인 직교 변환인 하다마드 변환을 적응필터에 적용하여 하다마드 필터를 구현한다. 그리고 연산횟수와 성능 그리고 잡음에 대한 영향에 대하여 고려하여 하다마드 필터와 시간 영역에서 구현한 필터를 비교한다.

2. 하다마드 변환

N개의 시퀀스, $x(i), i=0,1,\cdots,N-1$를 가지고 하다마드 변환을 하는 간단한 방법은 하다마드 행렬[7]을 이용하는 것이다.

2.1 하다마드 행렬

$N\times N$하다마드 행렬은 식 (1)과 같이 정의 된다. 단 여기서 N은 2의 거듭제곱수로 한정된다.

$$H_N = \begin{bmatrix} H_{N/2} & H_{N/2} \\ H_{N/2} & -H_{N/2} \end{bmatrix} \tag{1}$$

이때 $H_1 = 1$이다.

그리고 하다마드 행렬 H_N의 각 행(또는 열)들은 서로간에 직교성(orthogonality)을 갖고 있는데 이는 식 (2)로 표현할 수 있다.

$$H_N \cdot H_N = N I_N \tag{2}$$

여기서 I_N은 $N\times N$단위행렬(identity matrix)이다.

다음의 그림 1은 식 (1)을 통해 얻어진 4×4 하다마드 행렬과 8×8 하다마드 행렬의 예를 나타낸다.

$$H_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & - & 1 & - \\ 1 & 1 & - & - \\ 1 & - & - & 1 \end{bmatrix} \quad H_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & - & 1 & - & 1 & - & 1 & - \\ 1 & 1 & - & - & 1 & 1 & - & - \\ 1 & - & - & 1 & 1 & - & - & 1 \\ 1 & 1 & 1 & 1 & - & - & - & - \\ 1 & - & 1 & - & - & 1 & - & 1 \\ 1 & 1 & - & - & - & - & 1 & 1 \\ 1 & - & - & 1 & - & 1 & 1 & - \end{bmatrix}$$

그림 1 하다마드 행렬의 예(여기서 '-'는 '-1'을 뜻한다.)
Fig. 1 An example of Hadamard matrix(Here '-' means '-1')

2.2 이산 하다마드 변환

N개의 시퀀스로 이루어진 벡터 $\boldsymbol{x}$를 다음과 같이 정의하자.

$$\boldsymbol{x} = [x(0), x(1), \cdots, x(N-1)]^{\mathrm{T}} \tag{3}$$

식 (3)과 같이 정의된 시퀀스에 대해 하다마드 변환을 하여 얻어진 시퀀스를 벡터로 표현했을 때, 이를 $\boldsymbol{X}$라고 하자. 그러면 $\boldsymbol{x}$와 $\boldsymbol{X}$ 사이에는 다음의 식 (4)와 같은 관계가 성립된다.

$$\boldsymbol{X} = \frac{1}{N} H_N \boldsymbol{x} \tag{4}$$

그리고 하다마드 행렬은 non-singular 행렬이므로 역행렬이 존재하고 하다마드 역변환(IHT; inverse Hadamard transform)을 식 (4)를 참조하여 식 (5)와 같이 얻을 수 있다.

$$\boldsymbol{x} = N H_N^{-1} \boldsymbol{X} \tag{5}$$

식 (5)의 하다마드 역변환 식은 식 (2)를 통해 다음의 식 (6)과 같이 간단히 주어진다.

$$x = H_N X \tag{6}$$

여기서 H_N은 그림 1에서 처럼 단지 -1과 1만을 그 원소로 가지고 있으므로 하다마드 변환과 하다마드 역변환은 실수 덧셈과 뺄셈만을 필요로 한다.

2.3 고속하다마드 변환

식 (4)와 그림 1을 통해서 N개의 시퀀스를 하다마드 변환하기 위해서는 $N(N-1)$번의 덧셈 또는 뺄셈 연산이 필요하다는 것을 알 수 있다. 그러나 이와 같은 연산횟수는 고속 푸리에변환(FFT; fast Fourier transform)과 유사한 방법, 즉 행렬의 factorization[4]과 partitioning[1]을 이용한 고속 하다마드 변환(FHT; fast Hadamard transform)을 통해 덧셈과 뺄셈의 연산횟수를 $N\log_2 N$으로 줄일 수 있다.

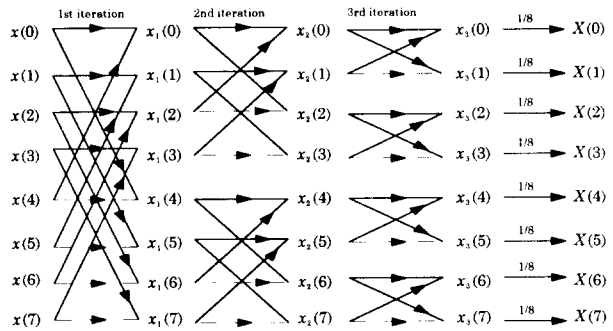


그림 2 고속하다마드 변환의 과정(N=8일때, '—' 덧셈, '--'뺄셈)
Fig. 2 The process of the FHT(when N=8, '—' addition, '--' subtraction)

그림 2는 8×1 벡터 $\boldsymbol{x}=[x(0),x(1),\cdots,x(7)]^{\mathrm{T}}$에 고속 하다마드변환을 수행하여 $\boldsymbol{X}=[X(0),X(1),\cdots,X(7)]^{\mathrm{T}}$를 얻는 과정을 보여주고 있다. 그림의 과정을 통해 이 과정이 고속 푸리에 변환 과정[2]과 유사함을 알 수 있다. 그리고 그림 2에서 $\boldsymbol{x}_k(i)$ $(i=1,2,\cdots,7)$는 k번째의 iteration을 통해 얻어지는 값을 의미한다.

3. 시간영역에서의 LMS 적응필터

그림 3은 일반적인 시간영역에서의 적응필터의 동작을 나타내는 블럭다이어그램이다. 시간 i에서 입력신호와 그들의 지연신호(delayed signal)들을 나타내는 벡터 $\boldsymbol{x}_i$와 그들 각각에 대응하는 가중치(weight) 벡터 $\boldsymbol{w}_i$를 다음과 같이 정의하자.

$$\begin{aligned}\boldsymbol{x}_i &= [x(i),x(i-1),\cdots,x(i-N+1)]^{\mathrm{T}}\\ \boldsymbol{w}_i &= [w_i(0),w_i(1),\cdots,w_i(N-1)]^{\mathrm{T}}\end{aligned} \tag{7}$$

그리고 i번째 출력신호 y_i는 다음의 식 (8)처럼 식 (7)에서 정의된 입력벡터와 가중치 벡터의 내적으로 주어진다.

$$y_i = \boldsymbol{w}_i^{\mathrm{T}} \cdot \boldsymbol{x}_i \tag{8}$$

그리고 우리가 원하는 기준신호를 d_i라고 했을 때, 기준신호와 출력값 사이의 오차(error)는 식 (9)로 주어진다.

$$\varepsilon_i = d_i - y_i \tag{9}$$

적응필터에서 이와 같이 얻어진 오차는 다음의 가중치 벡터 $\boldsymbol{w}_{i+1}$의 조정을 위해 사용되어진다. 가중치의 조정은 최소 평균자승오차(LMS)알고리듬으로 행하여지고 이 방법은 경사하강법(gradient decent)[10]에 기반을 두어 수행된다.

이 방법을 통해 다음의 가중치 $\boldsymbol{w}_{i+1}$는 현재의 가중치 $\boldsymbol{w}_i$와 다음과 같은 관계로 얻어진다.

$$\boldsymbol{w}_{i+1} = \boldsymbol{w}_i + 2\varepsilon_i \mu \boldsymbol{x}_i \tag{10}$$

여기서 μ는 안정도(stability)와 수렴정도(convergence rate)를 결정하는 파라메터(parameter)로 여기선 안정도 파라메터라 하겠다.

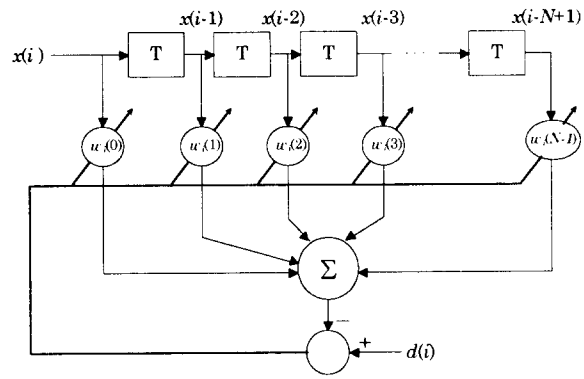


그림 3 기존의 시간영역에서의 LMS 적응필터
Fig. 3 Conventional time-domain LMS adaptive filter

4. 고속하다마드 변환을 이용한 LMS 적응필터

제안된 고속 하다마드변환을 이용한 적응필터의 기본적인 적응 알고리듬은 시간영역에서 구현한 일반적인 적응필터와 유사하다. 그러나 입력신호와 기준신호를 고속하다마드 변환을 수행하여 이들 변환된 신호에서 가중치를 얻어낸다는 과정에서 다른 점을 발견할 수 있다.

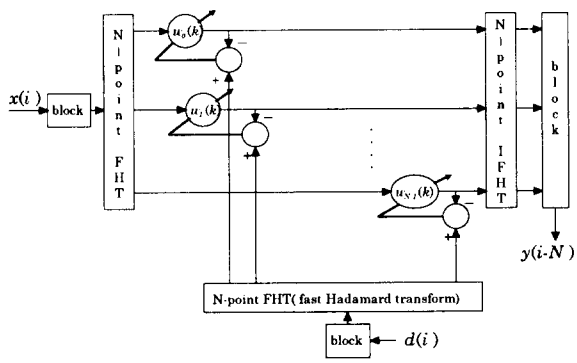


그림 4 고속 하다마드 변환을 이용한 LMS 적응필터
Fig. 4 LMS adaptive filter using FHT

우선 입력 신호 $x(i)$와 기준신호 $d(i)$를 N-단위시간까지 저

장하여 하나의 단위데이터블럭으로 묶는다. 즉, 하나의 단위데이터블럭안에 입력신호에 대하여 $x(N-1), x(N-2), \cdots, x(0)$의 값들이 존재하고 기준신호에 대하여도 $d(N-1), d(N-2), \cdots, d(0)$의 값들이 존재하게 된다. 이들은 다음처럼 벡터로 표현이 가능하다.

$$\begin{aligned} \boldsymbol{x}_k(k) &= [x(N-1), x(N-2), \cdots, x(0)]^T \\ \boldsymbol{d}_k(k) &= [d(N-1), d(N-2), \cdots, d(0)]^T \end{aligned} \tag{11}$$

여기서 $\boldsymbol{x}_k(k)$의 k는 $\boldsymbol{x}_k(k)$가 k번째 블럭임을 나타내준다.

그 다음에는 이들 각각에 고속하다마드변환을 수행한다. 이때 고속하다마드변환을 위해 $N \times N$ 하다마드행렬이 필요하게 된다. 이와같은 변환을 통하여 얻어진 신호는 각각 다음과 같은 단위데이터블럭을 이루게된다.

$$\begin{aligned} \boldsymbol{X}_k(k) &= [X(N-1), X(N-2), \cdots, X(0)]^T \\ \boldsymbol{D}_k(k) &= [D(N-1), D(N-2), \cdots, D(0)]^T \end{aligned} \tag{12}$$

그리고 하다마드 영역에서의 출력신호 $\boldsymbol{Y}_k(k)$와 그 역변환으로 얻어지는 $\boldsymbol{y}_k(k)$는 식 (13)과 같이 얻어진다.

$$\begin{aligned} \boldsymbol{Y}_k(k) &= \boldsymbol{W}_H(k)\ \boldsymbol{X}_k(k) \\ \boldsymbol{y}_k(k) &= \boldsymbol{H}_N\ \boldsymbol{Y}_k(k) \end{aligned} \tag{13}$$

여기서 $\boldsymbol{W}_H(k)$는 하다마드 영역에서의 가중치를 대각원소로 갖는 대각행렬(diagonal matrix)이다

이와 같은 과정을 통해 얻어진 $\boldsymbol{Y}_k(k)$와 $\boldsymbol{D}_k(k)$는 하다마드 영역에서의 오차를 얻어내기 위해 사용된다. 이때 오차는 다음과 같이 주어진다.

$$\boldsymbol{\varepsilon}_k(k) = \boldsymbol{D}_k(k) - \boldsymbol{Y}_k(k) \tag{14}$$

여기서 $\boldsymbol{\varepsilon}_k(k)$는 k번째 데이터블럭에 대한 오차벡터이다. 그리고 $\boldsymbol{W}_k(k)$는 같은 블럭에서의 가중치 벡터로 다음과 같이 정의된다.

$$\boldsymbol{W}_k(k) = [W_{N-1}(k), W_{N-2}(k), \cdots, W_0(k)]^T \tag{15}$$

이와 같은 가중치 행렬내의 각 값들은 독립적으로 갱신되는데 그 갱신방법은 시간영역에서와 마찬가지로 다음과 같은 경사하강법을 사용한다.

$$\boldsymbol{W}_k(k+1) = \boldsymbol{W}_k(k) + 2\mu\ \boldsymbol{E}_k(k)\ \boldsymbol{X}_k(k) \tag{16}$$

여기서 $\boldsymbol{E}_k(k)$는 벡터 $\boldsymbol{\varepsilon}_k(k)$의 원소들을 식 (15)에서 처럼 대각요소(diagonal element)로 갖는 행렬이다.

이와같은 적응알고리듬을 통해 데이터블럭 단위로 가중치의 갱신과 적응이 이루어지게 된다.

5. 계산상의 효율 비교

시간영역에서 구현한 적응필터는 N개의 데이터를 가지고 있는 단위블럭에 대해 N^2번의 적응과정이 수행된다. 이는 N개의 서로다른 가중치에 각 데이터가 N개의 새로운 입력이 들어갈때마다 적응이 이루어지기 때문이다.

반면 하다마드영역에서 구현한 적응필터는 N개의 데이터를 가진 단위블럭에 대해 N개의 서로 다른 가중치 값들에 대해 한번씩의 갱신이 이루어지므로 N번의 적응과정만이 필요하다. 이러한 적응과정을 수행시에 한번의 적응과정에 대해 한번의 곱셈연산이 필요하다. 그리고 적응과정의 수와 가중치의 갱신과정의 횟수는 갖고 역시 이 과정에서도 한번의 적응과정에서 한번의 곱셈연산이 필요하게 된다. 따라서 N개의 데이터 샘플에 대해서 기존의 시간영역의 필터는 $2N^2$번의 곱셈연산이 필요하게 된다. 반면에 고속 하다마드 변환을 이용한 적응필터는 N번의 적응과정과 역시 N번의 가중치 갱신과정만이 필요하기 때문에 N개의 데이터 샘플에 대해서 $2N$번의 곱셈과정이 필요할 뿐이다. 다음의 표 1은 두 필터의 곱셈연산횟수를 비교한 것이다.

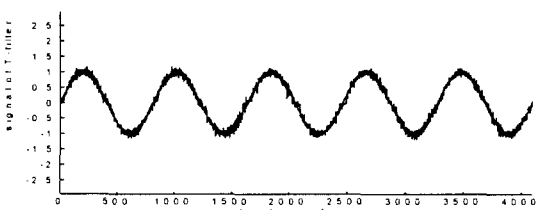
표 1 곱셈연산횟수의 비교
Table 1 Comaprison of multiplication number

N	시간영역 필터	하다마드 필터
4	32	8
8	128	16
16	512	32
32	2048	64
n	$2n^2$	2n

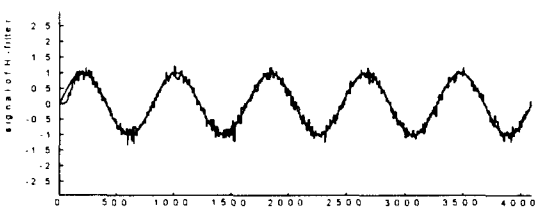
6. 하다마드 필터의 특성 고찰

하다마드 적응필터의 성능에 대해 알아보기 위해 잡음을 제거하는 필터로서 두 필터를 사용하여 다음과 같은 실험을 행하였다. 즉, 필터의 입력신호로는 sine함수에 정규분포를 갖는 잡음이 섞인 신호를 사용하고 출력에서는 입력과 같은 sine함수를 얻도록 하였다. 이때, 필터에서 N값은 8로 취하고, 전체 샘플 시간은 4,096으로 취하였다. 그리고 잡음의 분산은 0.1로 택하였다.

아래 그림 5는 각 필터의 출력을 나타낸다. 이때, 각 필터에 쓰인 안정도 파라메터 값은 해당 필터가 최소의 MSE값을 갖도록 정하였다. 즉 시간영역 필터는 μ값이 0.07일때, 0.0067의 최소 MSE값을 갖고 하다마드 필터는 μ값이 0.23일때 0.0157의 최소 MSE값을 나타내었다.



(a) 시간영역 필터에 의한 결과(MSE =0.0067, μ=0.07)



(b) 하다마드 필터에 의한 결과(MSE=0.0157, μ=0.23)

그림 5 각 필터의 출력신호 (잡음분산 = 0.1)

Fig. 5 Output signal of the two filters(noise variance = 0.1)

하다마드 필터와 시간영역 필터에서 안정도 파라메터 μ값은 각 필터에 대해 동일한 영향을 끼치지 않는다. 따라서 즉 그림 6과 같이 파라메터에 대하여 다른 MSE 특성이 나타난다. 그림에서 실선은 시간영역 필터에 대한 자취이고 점선은 하다마드 필터에 대한 자취이다.

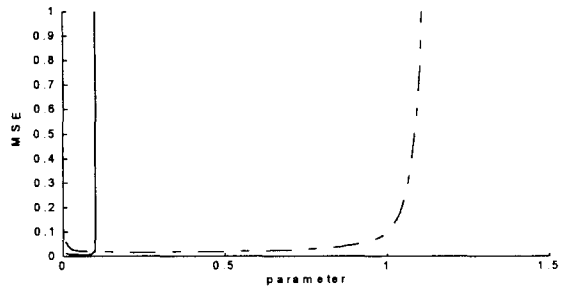
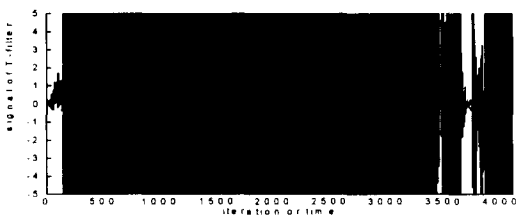


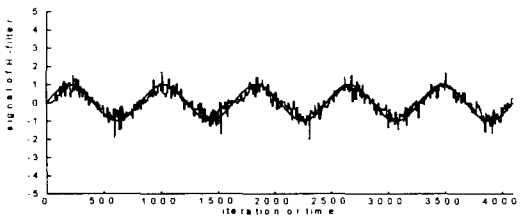
그림 6 안정도 파라메터 변화에 따른 MSE비교 (실선이 시간필터 점선이 하다마드 필터임)

Fig. 6 MSE comparison for controllability parameter(solid line is for time filter and dashed line is for Hadamard filter)

이 실험에서 시간영역에서 구현한 적응필터는 출력신호가 발산하지 않기 위해서는 0.1보다 작은 값을 요구하며 이 값은 시간영역 필터가 최소의 MSE값을 갖게하는 0.07과 매우 유사한 값으로 잡음의 분산이 약간만 커져도 발산할 수 있음을 말한다. 반면 하다마드 영역에서 구현한 적응필터는 이보다 매우 큰 1부근에서 발산이 시작되고 최소 MSE값을 갖게하는 파라메터 0.23과는 매우 거리가 멀다. 이런 경우에는 잡음의 통계적 특성이 달라지더라도 쉽게 발산하지 않게 된다. 다음의 그림 7은 이를 설명해 주는 실험 결과로 잡음의 분산이 0.1에서 1로 변했을때의 결과이다.



(a) 시간영역 필터에 의한 결과(MSE =∞, μ=0.07)



(b) 하다마드 필터에 의한 결과(MSE=0.1050, μ=0.23)

그림 7 각필터의 결과 (잡음분산 = 1)

Fig. 7 Output signal of the two filters (noise variance = 1)

7. 결론

본 논문에서는 하다마드 변환을 이용한 적응필터를 구현하여 모의실험을 통해 필터의 특성을 시간영역 필터와 비교하여 보았다. 하다마드 필터는 비록 최소 MSE값이 0.0157로 시간영역 필터의 최소 MSE값인 0.0067에 비해 약 2배 정도의 값을 나타내었지만 곱셈연산의 횟수와 잡음에 대한 민감도 정도에서 우수한 특성이 나타난다. 우선 연산횟수에 대해서는 4,096개의 데이터에 대시간영역 필터의 경우 2×4,096×4,096=33,554,432 번의 곱셈연산이 필요한 반면 하다마드 필터의 경우 2×4,096 = 8,192번의 곱셈연산이 필요할 뿐이다. 두번째 하다마드 영역에서 구현한 필터는 그림 6에서와 같이 안정도 파라메터에 대한 수렴영역이 넓을 뿐더러 그림 7에서 처럼 잡음의 분산이 1로 커졌을 때 시간영역 필터는 출력신호가 진동하며 발산하는 반면 하다마드 필터는 쉽게 발산하지 않는 특성을 보임을 알 수 있다.

참고문헌

[1] N. Ahmed and S. M. Cheng, "On Matrix Partitioning and a Class of Algorithms," *IEEE Trans. Education,* vol. E-13, pp. 103-105, Aug. 1970.

[2] J. W. Cooley and J. W, Tukey, "An Algorithm for the Machine Computation of Complex Fourier Series," *Mathematics of Computation,* vol. 19, pp. 297-301, Apr. 1965.

[3] M. Dentino, J. McCool, and B. Widrow, "Adaptive Filtering in the Frequency Domain," *Proc. IEEE,* vol. 66, pp. 1658-1659, Dec. 1978.

[4] Y. A. Geadah and M. J. G. Corinthios, "Natural Dyadic and Sequence-order Algorithms and Processors for the Walsh-Hadamard Transform," *IEEE Trans. Computers,* vol. C-26, no. 5, pp. 435-442, May 1977.

[5] R. W. Lucky, "Automatic Equalization for Digital Communication," *Bell Syst. Tech. J.* vol. 44, pp 547-588, Apr. 1965.

[6] J. McCool and B. Widrow, "Principles and Applications of Adaptive Filters: A Tutorial Review," Naval Undersea Center, San Diego, CA, *Tech. Pull.* 530, Mar, 1977.

[7] W. K. Pratt, J. Kane, and H, C. Andrews, "Hadamard Transform Image Coding," *Proc. IEEE,* vol. 57, pp. 58-68, Jan. 1969.

[8] B. Widrow, J. Glover, J. McCool, J. Kaunitz, C. S. Williams, R. H. Hearn, J. R. Zeidler, E. Dong, and R. C. Goodlin, "Adaptive Noise Cancelling: Principles and Applications," *Proc. IEEE,* vol. 63, pp. 1692-1716. Dec. 1975.

[9] B. Widrow, J. McCool, M. G. Larimore, and C. R. Johnson, Jr., "Stationary and Nonstationary Learning Characteristics of the LMS Adaptive Filter," *Proc. IEEE,* vol 64, pp. 1151-1162, Aug. 1976.

[10] D. J. Wilde, *Optimum-Seeking Methods,* Englewood Cliffs, NJ: Prentice-Hall, 1964.