

등가의 Wiener-Hopf 방정식을 이용한 LMS 알고리즘에 관한 연구

정회원 안 봉 만*, 황 지원**, 조 주 필***^o

Research on the Least Mean Square Algorithm Based on Equivalent Wiener-Hopf Equation

Bong-Man Ahn*, JeeWon Hwang**, Juphil Cho***^o *Regular Members*

요 약

본 논문은 등가의 Wiener-Hopf 방정식의 해를 LMS 알고리즘을 이용하여 구할 수 있는 방법과 격자필터에서 직접적으로 TDL 필터의 계수를 구할 수 있는 방법을 제안한다. 이를 위해 격자필터를 이용하여 생성한 직교입력 신호를 등가의 Wiener-Hopf 방정식에 적용하여 그 해를 최소평균자승 알고리즘을 이용하여 순환적으로 구하는 방법을 보인다. 이와 같은 경우 기존에는 오차와 regression 계수를 순환적으로 구할 수 있는데 반하여 본 논문에서는 오차와 TDL 필터의 계수를 순환적으로 구할 수 있는 장점이 있다. 또한 제안한 알고리즘의 수렴적 특성을 이론적으로 고찰하였다. 그 결과는 전통적 해석과 유사하게 나타남을 알 수 있었다. 성능 평가 결과를 통해 제안한 알고리즘이 매우 우수한 성능을 나타내고 있음을 확인하였다.

Key Words : Wiener-Hopf, LMS, TDL, Lattice, Regression coefficient

ABSTRACT

This paper presents the methods which obtain the solution of Wiener-Hopf equation by LMS algorithm and get the coefficient of TDL filter in lattice filter directly. For this result, we apply an orthogonal input signal generated by lattice filter into an equivalent Wiener-Hopf equation and shows the scheme that can obtain the solution by using the MMSE algorithm. Conventionally, the method like aforementioned scheme can get an error and regression coefficient recursively. However, in this paper, we can obtain an error and the coefficients of TDL filter recursively. And, we make an theoretical analysis on the convergence characteristics of the proposed algorithm. Then we can see that the result is similar to conventional analysis. Also, by computer simulation, we can make sure that the proposed algorithm has an excellent performance.

I. 서 론

현재까지 Wiener-Hopf 방정식은 필터의 구조에 관계없이 그 해를 순환적으로 구하는 방법 즉, 여러 알고리즘 개발에 근본이 되어왔다. 적응필터 입력은 비직교입력 신호를 사용하는 경우와 직교입력 신호를

사용하는 경우로 나눌 수가 있는데 전자의 경우는 대표적으로 TDL(tapped-delay-line) 필터가 대표적이고 후자는 격자(lattice) 필터가 대표적이다^{[1][2][13]}. 이 밖에 직교입력신호를 이용하는 적응필터기법들은 Gram Schmidt 직교과정^[2], Gauss-Seidel^[3], QR^{[1][2]} 등이 있다. 이들 필터를 적응시키는 알고리즘은 그 기반을

* 전북대학교 Next 사업단, **익산대학 컴퓨터학과, *** 군산대학교 전자정보공학부(stefano@kunsan.ac.kr)^o : 교신저자
논문번호 : KICS2008-01-019, 접수일자 : 2008년 1월 12일, 최종논문접수일자 : 2008년 5월 10일

Wiener-Hopf 방정식에 두고 LMS 형태, LS 형태등으로 개발되어 왔다^[1]. 또한 시간영역 신호를 주파수 또는 복소수영역으로 변환시켜서 사용하는 Transform Domain 적응 필터가 있다. 일반적으로 Transform Domain에서 사용하는 변환은 DFT, DCT, DHT 및 FFT 등이 있다^{[1][4][5]}. 시간영역에서 입력신호를 직교화시켜 사용하는 Lattice 필터^{[1][2]}, Gram Schmidt 직교과정^[2]등에서는 직교입력벡터를 이용하여 원하는 응답벡터를 예측하는 부분이 존재하는데 하나의 예로 격자필터의 joint-process estimation 단이 있다^{[1][2][8][9][10][11][12]}. 또한 Gram Schmidt 직교화 과정 및 QR등에서도 joint-process estimation 단과 유사한 단이 존재한다^{[1][2]}. Joint-process estimation 단에서 계수는 TDL 필터의 계수가 아니기 때문에 직교화 과정에 대한 적절한 역변환과정을 거쳐야 TDL 필터의 계수를 얻을 수 있다. 그러므로 격자필터와 같이 직교입력 신호를 사용하는 경우 Wiener-Hopf 방정식은 적합한 방법이라고 할 수는 없다. 그 이유로는 직교입력신호를 이용하는 필터 구조에서 오차를 최소화하여 구한 계수들이 TDL 계수와 다르기 때문에 적절한 역변환 과정을 거쳐서 TDL 계수로 변환해야 하는 제약이 있기 때문이다. 일반적으로 직교입력 벡터를 사용하는 알고리즘들은 비정상적인 환경에서 TDL 필터보다 우수한 수렴특성을 나타낸다^[1]. 우수한 수렴특성을 가지고 있음에도 TDL 필터 계수를 얻는데 제약이 존재하였다. LMS 계열의 격자필터는 J. Burg^[7], L. Griffiths^[8], J. D. Pack^[9] 등이 제안하였고 이것들은 음성처리, 잡음제거 및 선형 예측등에서 사용되어 왔다. 그 예로 격자필터에서 TDL 필터의 계수를 구하는데는 Levinson-Durbin 알고리즘이 필요한데 그 계산량은 격자필터의 전체 계산량보다 많은 $O(M^2)$ 이다^[15]. 본 논문은 직교입력벡터를 이용하는 경우 TDL 필터의 계수를 구할 수 있는 Wiener-Hopf 형태의 방정식을 제안한다. 또한 제안한 Wiener-Hopf 형태의 방정식 해를 순환적으로 구할 수 있는 LMS(Least Mean Square) 알고리즘을 제안한다. 이 방법은 격자필터에서 Levinson-Durbin 알고리즘을 사용하여 TDL 필터의 계수를 구하는 경우 보다 적은 계산량만 필요로 한다. 본 논문에서는 입력벡터를 직교화시키는 방법은 참고문헌 [2]에서 유도한 격자 필터를 사용하였다. 논문의 컴퓨터 시뮬레이션에서 참고문헌 [2][10][8]의 방법과 본 논문에서 제안한 알고리즘과 MMSE면에서 비교한 그림이 포함되어 있다. 결과적으로 본 논문은 다음과 같은 측면에서 그 공헌도가

있다고 판단된다. 첫번째로 계산량이 적은 Wiener-Hopf 버전의 약간 다른 방정식을 제안한다. Wiener-Hopf 방정식의 상관행렬은 $M \times M$ 인 full 행렬인 반면에 본 논문에서 제안한 상관행렬은 $M \times M$ 인 삼각행렬(triangular matrix)이다. 두 번째로 Levinson-Durbin 알고리즘을 사용하지 않고 격자필터에서 TDL 필터의 계수를 구하는 LMS 알고리즘을 제안하였다. 세 번째로 격자필터에서 TDL 필터의 계수를 구하는데 계산량이 적은 장점이 있다. 부가적인 장점으로는 첫 번째로 격자필터에서 필요하는 반사계수를 포함하고 오차와 TDL 필터의 계수를 모두 한번에 구할 수 있는 효율적인 알고리즘을 제안한 것이다. 논문의 구성은 제 II장에서 등가의 Wiener-Hopf 방정식을 이용한 LMS 알고리즘 유도에 대하여 다루었으며 III장에서는 제안한 알고리즘의 수렴특성에 대하여 논의하였으며 IV장에서는 컴퓨터 시뮬레이션을 수행한 결과를 제시하였으며 V장에서는 결론 및 고찰을 다루었다.

II. 등가의 Wiener-Hopf 방정식을 이용한 LMS 알고리즘 유도

2.1 RLS 알고리즘의 고찰

그림 1에서 입력신호 벡터 $x(n)$, 최적계수벡터 w_{opt} , 직교벡터 $b(n)$ 및 k 는 $M \times 1$ 인 벡터이고 Q^H 는 $M \times M$ 인 직교 변환 행렬이다. 그리고 $d(n)$ 는 원하는 응답, $\hat{d}(n)$ 는 예측된 원하는 응답 그리고 $v(n)$ 은 측정 잡음이다. 그리고 $v(n)$ 은 입력신호 벡터 $x(n)$ 와 독립적이다. 직교 변환 행렬 Q^H 은 선형 독립인 입력신호벡터 $x(n)$ 으로부터 상호 직교하는 벡터 $b(n)$ 를 얻는 과정을 나타낸다. 또한 벡터 $b(n)$ 와 계수벡터 k 선형 결합하여 예측된 원하는 응답 $\hat{d}(n)$ 를 생성한다. 본 절에서는 w_{opt} 의 순시 예측

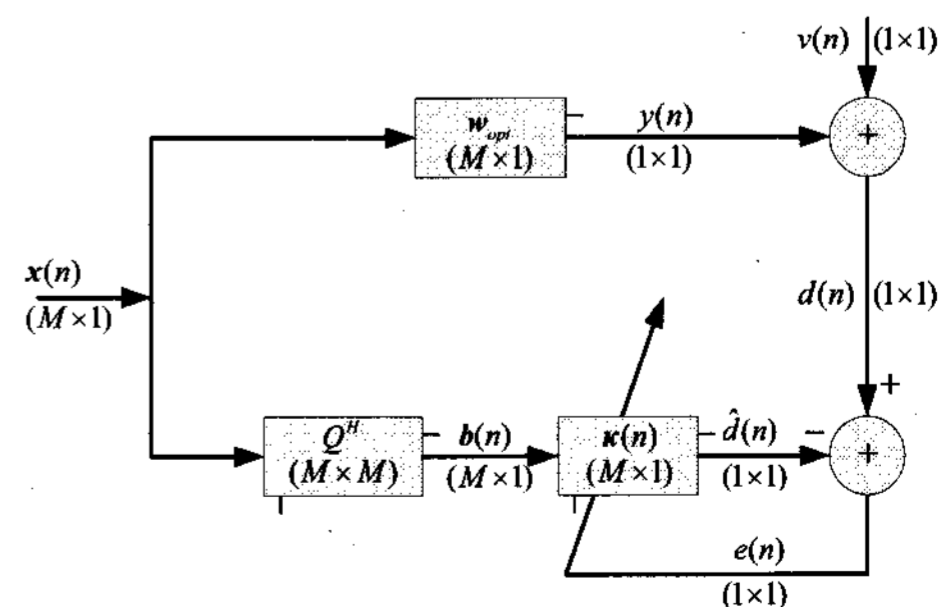


그림 1. 제안된 알고리즘의 구조도
Fig. 1. Structure of proposed algorithm

값 $w(n)$ 와 $k(n)$ 을 함께 표현하는 방법을 설명한다. 비용함수는 다음과 같다.

$$\begin{aligned} \mathcal{J}(k) &= E[(d(n) - k^H b(n))(d^*(n) - b^H(n)k)] \quad (2.1) \\ &= E[(d(n) - w_{opt}^H x(n))(d^*(n) - b^H(n)k)] \end{aligned}$$

여기서

$$b(n) = Q^H x(n) \quad (2.2)$$

$$w_{opt} = Qk. \quad (2.3)$$

또한 (H)는 Hermitian transpose를 나타낸다.

$$\frac{\partial \mathcal{J}(k)}{\partial k} = 2E[(d(n) - w_{opt}^H Q^{-H} Q^H x(n))(-b^H(n))] \quad (2.4)$$

식(2.4)에서 w_{opt} 를 구해진다.

$$R_{bx} w_{opt} = p_{bd} \quad (2.5)$$

여기서

$$R_{bx} = E[b(n)x^H(n)] \quad (2.6)$$

$$p_{bd} = E[b(n)d^*(n)] \quad (2.7)$$

식(2.5)은 식(2.1)에서 비용함수를 k 으로 편미분하여 얻은 결과이다. 이것은 식(2.1)의 물리적 의미를 유지하면서 w_{opt} 를 얻을 수 있다는 것을 의미한다.

식(2.1)과 식(2.5)는 새로운 방법이다. 그래서 식(2.5)를 등가의 Wiener-Hopf 방정식이라 하겠다. 식(2.5)는 서로 직교 변환된 입력 $b(n)$ 을 사용하는 필터에서 TDL 필터의 계수 w_{opt} 또는 $w(n)$ 을 구할 수 있음을 나타낸다. 또한 R_{bx} 는 $b(n)$ 과 $x(n)$ 의 상관행렬로 QR분해 알고리즘에서와 같이 상 삼각(upper triangular) 행렬 형태를 가짐을 쉽게 알 수 있다. 제안한 알고리즘은 기존의 LMS 알고리즘과 같이 식(2.8)과 식(2.9)와 같다.

$$e(n) = d(n) - w^H(n)x(n) \quad (2.8)$$

$$w(n) = w(n-1) + 2\mu b(n)e(n) \quad (2.9)$$

여기서 $w(n)$ 은 w_{opt} 의 순시 예측값이고 또한 TDL 계수벡터이다. μ 는 step-size이다.

직교변환된 입력 $b(n)$ 을 얻기 위하여 참고문헌[2] 격자알고리즘을 사용하였다. 제안한 알고리즘은 기존의 알고리즘과 달리 ladder 부분에서도 입력 벡터 $x(n)$ 을 사용한다. 오차 $e(n)$ 은 TDL 필터와 같이 구하고 계수벡터의 순환식에서는 역방향 예측오차 $b(n)$ 를 사용하는 형태를 갖는다.

표 1. 제안한 알고리즘.
Table 1. Proposed Algorithm.

Initialization

$$f_m(-1) = b_m(-1) = b_m(-2) = 0$$

$$f_0(n) = b_0(n) = x(n)$$

$$v_m(-1) = \epsilon > 0$$

$$k_m(-1) = 0$$

$$w(0) = 0$$

for $n = 1, 2, 3, 4, \dots$

for $m = 1, 2, \dots, M$

$$v_m(n) = \rho v_m(n-1) + |f_{m-1}(n)|^2 + |b_{m-1}(n-1)|^2 \quad (T1.1)$$

$$k_m(n) = k_m(n-1) + \frac{f_{m-1}(n-1)b_m^*(n-1) + b_{m-1}(n-2)f_m(n-1)}{v_m(n-1)} \quad (T1.2)$$

$$f_m(n) = f_{m-1}(n) + k_m(n)b_{m-1}(n-1) \quad (T1.3)$$

$$b_m(n) = b_{m-1}(n) + k_m^*(n)f_{m-1}(n) \quad (T1.4)$$

end m

$$b(n) = [b_0(n), b_1(n), \dots, b_{M-1}(n)]^H$$

$$e(n) = d(n) - w^H(n)x(n) \quad (T1.5)$$

$$w(n+1) = w(n) + 2\mu b(n)e(n) \quad (T1.6)$$

end n

여기서 식(T1.1)의 ρ 는 forgetting factor로 $0 < \rho \leq 1$ 이다. 참고문헌[2]의 격자 알고리즘 (T1.1)~(T1.4)는 공통으로 사용하는 식들이다. 계산량은 곱셈과 나눗셈만 포함하며 forgetting factor ρ , 상수 및 step-size μ 의 곱셈은 계산량에서 제외하겠다. 식(T1.1)에서 곱셈 $2M$, 식(T1.2)에서 곱셈 $2M$ 나눗셈 M , 식(T1.3)에서 곱셈 M , 식(T1.4)에서 곱셈 M 번 발생하고 식(T1.5) 곱셈 M 번 그리고 식(T1.6)에서 곱셈이 M 발생하여 총 곱셈의 계산은 $8M$ 총 나눗셈 계산은 M 번 발생한다. 그러므로 제안한 알고리즘은 $O(M)$ 이다. 여기서 M 은 필터의 차수이다. 식(T1.5)와 식(T1.6)의 계산량을 $2M$ 번의 계산량을 가지므로 기존 격자필터에서 Levinson-Durbin 알고리즘을 사용하는 경우보다 매우 적은 계산량을 사용하여 TDL 필터의 계수를 구할 수 있는 장점이 있다. 이와 같은 형태의 알고리즘에서는 (T1.2)의 반사계수(reflection coefficient), (T1.5)오차 및 (T1.6)의 TDL 필터의 계수를 동시에 구할 수 있어 매우 효율적인 알고리즘이 된다.

III. 수렴특성 해석

본 절은 본 논문에서 제안한 알고리즘에 대하여 다양한 수렴특성해석을 수행한다.

3.1 경사 특성

식(2.4)에서 k 에 대한 경사를 w 에 대한 경사로 표현할 수 있음을 보였다. 그래서 k 에 대한 경사를 다음과 같이 w 에 대한 표현으로 재정의한다 [13].

$$\Delta_{w_{opt}} = -2E[b(n)d(n)] + 2E[b(n)x^H(n)]w_{opt} \quad (3.1)$$

LMS 알고리즘에서 R_{bx} 와 p_{bd} 에 대한 순시 예측들이 사용되기 때문에 순시경사 방향, $\hat{\Delta}_w(n)$ 을 이용하고, 순시경사 방향을 평균하면 시간 n 을 떼어낸 계수벡터 w 에 대해 다음과 같은 식을 얻는다.

$$E[\hat{\Delta}_w(n)] = 2E[b(n)x^H(n)]w - E[b(n)d(n)] = \Delta_{w_{opt}} \quad (3.2)$$

식(3.2)에서 w 는 w_{opt} 이다. 그러므로 $\hat{\Delta}_w(n)$ 은 $\Delta_{w_{opt}}$ 의 unbiased된 순시 예측이다. 그리고 ergodic 환경에서는 $\hat{\Delta}_w(n)$ 는 $\Delta_{w_{opt}}$ 이 된다.

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=1}^N \hat{\Delta}_w(n+k) \rightarrow \Delta_{w_{opt}} \quad (3.3)$$

3.2 계수 벡터의 수렴특성

최적계수벡터 w_{opt} 와 이것의 예측값과 $w(n)$ 과의 차이를 계수 편차(deviation of weight vector) $\Delta_w(n)$ 라고 하자[1][13].

$$\Delta_w(n) = w(n) - w_{opt} \quad (3.4)$$

위의 정의를 사용하여 LMS 알고리즘의 계수 순환식을 다음식과 같이 표현 할 수 있다.

$$\Delta_w(n+1) = \Delta_w(n) + 2\mu b(n)e(n) \quad (3.5)$$

$$e(n) = d(n) - w^H(n)x(n) = w_{opt}^H x(n) + \eta(n) - w^H(n)x(n) \quad (3.6)$$

여기서 $\eta(n)$ 은 백색추정잡음으로 평균이 0, 분산은 σ_η^2 이다. 식(4.2), 식(4.3)을 이용해 기대값을 얻고 다음과 같은 식을 얻는다.

$$E[\Delta_w(n+1)] = E[(I - 2\mu b(n)x^H(n))\Delta_w(n)] + 2\mu E[\eta(n)b(n)] \quad (3.7)$$

식(3.7)에서 $\Delta_w(n)$ 의 요소들은 $x(n)$ 의 요소, 역방향예측오차 $b(n)$ 의 요소들이 확률적으로 독립적이라고 가정하고 역방향예측오차 $b(n)$ 의 요소들과 백색 추정잡음 $\eta(n)$ 은 직교한다. 그러므로 식(3.7)는 다음과 같이 간략화된다.

$$E[\Delta_w(n+1)] = (I - 2\mu E[b(n)x^H(n)])E[\Delta_w(n)] \quad (3.8)$$

위식에 식(2.5)을 적용하면 다음과 같이 표현된다.

$$E[\Delta_w(n+1)] = (I - 2\mu R_{bx})E[\Delta_w(n)] \quad (3.9)$$

식(3.8)에서 계수편차는 단지 과거의 입력신호와 역방향예측 오차에만 종속되고 또한 최적 해에서 오차신호는 역방향 예측오차 벡터의 요소와 직교하므로 이것을 이용하면 식(3.9)은 다음과 같이 쓸 수 있다.

$$E[\Delta_w(n+1)] = (I - 2\mu R_{bx})^{n+1} E[\Delta_w(0)] \quad (3.10)$$

위식의 R_{bx} 는 상삼각행렬이기 때문에 고유값(eigenvalue)들은 R_{bx} 의 대각요소들이다. 그러므로 식(3.10)은 다음식과 같이 쓸 수 있다.

$$E[\Delta_w(n+1)] = \begin{bmatrix} (1-2\mu\lambda_1)^{n+1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & (1-2\mu\lambda_M)^{n+1} \end{bmatrix} E[\Delta_w(0)] \quad (3.11)$$

여기서 $\lambda_i, i=1,2,\dots,M$ 은 고유값(eigenvalue)이다. 식(3.11)을 조사하면 평균면에서 계수편차 $E[\Delta_w(n+1)]$ 가 수렴하기 위한 조건에서 μ 는 다음의 범위에서 결정되어야 한다.

$$-1 < (1 - 2\mu\lambda_i) < 1, \text{ for } i=1,2,\dots,M \quad (3.12)$$

위에서 μ 가 결정되면 $n \rightarrow \infty$ 일때 $E[\Delta_w(n+1)]$ 는 0으로 수렴하고 μ 의 범위는 다음과 같다.

$$0 < \mu < \frac{1}{\lambda_{max}} \quad (3.13)$$

또한 이때 시정수는 다음과 같이 구해진다.

$$\tau_{i,D} = \frac{-1}{\ln(1-2\mu\lambda_i)} \approx \frac{1}{2\mu\lambda_i}, \quad \mu \ll 1. \quad (3.14)$$

3.3 계수편차벡터의 상관행렬

계수편차벡터의 covariance는 다음과 같이 정의된다[1][13].

$$cov[\Delta_w(n)] = E[\Delta_w(n)\Delta_w^H(n)] = E[(w(n) - w_{opt})(w(n) - w_{opt})^H] \quad (3.15)$$

식(3.7)을 식(3.15)에 대입하고 $\eta(n)$ 은 $\Delta_w(n)$ 과 독립이고 또한 $x(n)$ 과 $b(n)$ 과 각각 직교하기 때문에 $\eta(n)$ 가 한번 곱해진 항들은 제거할 수 있다. 그런 후에 $\Delta_w(n)$ 은 $x(n)$ 및 $b(n)$ 에 대하여 독립적이라 가정하면 다음과 같이 쓸 수 있다.

$$\begin{aligned} \text{cov}[\Delta_w(n+1)] &= \text{cov}[\Delta_w(n)] - 2\mu R_{bx} \text{cov}[\Delta_w(n)] - 2\mu \text{cov}[\Delta_w(n)] R_{bx}^H \\ &\quad + 4\mu^2 E[b(n)x^H(n)\Delta_w(n)\Delta_w^H(n)x(n)b^H(n)] \\ &\quad + 4\mu^2 E[\eta^2(n)b(n)b^H(n)] \end{aligned} \quad (3.16)$$

위식의 $E[b(n)x^H(n)\Delta_w(n)\Delta_w^H(n)x(n)b^H(n)]$ 을 정리하기 위하여 식(2.2)를 이용하여 $b(n)$ 에서 직교행렬을 분리한 형태로 쓰면 다음과 같이 나타낼 수 있다^[13].

$$\begin{aligned} Q^H E[x(n)x^H(n)E[\Delta_w(n)\Delta_w^H(n)]x(n)x^H(n)]Q & \quad (3.17) \\ &= 2R_{bx} \text{cov}[\Delta_w(n)] R_{bx}^H + R_{bb} \text{tr}[R_{xx} \text{cov}[\Delta_w(n)]] \end{aligned}$$

위의 식을 식(3.16)에 대입하면 다음의 식을 얻는다.

$$\begin{aligned} \text{cov}[\Delta_w(n+1)] &= \text{cov}[\Delta_w(n)] - 2\mu R_{bx} \text{cov}[\Delta_w(n)] - 2\mu \text{cov}[\Delta_w(n)] R_{bx}^H \\ &\quad + 8\mu^2 R_{bx} \text{cov}[\Delta_w(n)] R_{bx}^H + 4\mu^2 R_{bb} \text{tr}[R_{xx} \text{cov}[\Delta_w(n)]] \\ &\quad + 4\mu^2 \sigma_\eta^2 R_{bb} \end{aligned} \quad (3.18)$$

여기서 $\text{tr}[\cdot]$ 은 trace를 나타낸다. 식(3.18)에서 R_{bb} 은 대각행렬, R_{bx} 는 상삼각행렬이다. 또한 $Q^H Q = Q Q^H = I$ 이고 임의 B 에 대하여 $Q^H \text{tr}[B]Q = \text{tr}[Q^H B Q]I$ 이 성립하므로 다음 식이 성립한다.

$$\begin{aligned} Q^H \text{tr}[R_{xx} \text{cov}[\Delta_w(n)]]Q &= \text{tr}[Q^H R_{xx} Q \text{cov}[\Delta_w(n)]] \quad (3.19) \\ &= \text{tr}[\Lambda \text{cov}[\Delta_w(n)]] \end{aligned}$$

위의 식을 식(3.18)에 대입 후 정리하면 다음과 같은 식을 얻는다.

$$\begin{aligned} \text{cov}[\Delta_w(n+1)] &= \text{cov}[\Delta_w(n)] - 2\mu \Lambda \text{cov}[\Delta_w(n)] - 2\mu \text{cov}[\Delta_w(n)] \Lambda \\ &\quad + 8\mu^2 \Lambda \text{cov}[\Delta_w(n)] \Lambda + 4\mu^2 \Lambda \text{tr}[\Lambda \text{cov}[\Delta_w(n)]] \\ &\quad + 4\mu^2 \sigma_\eta^2 \Lambda \end{aligned} \quad (3.20)$$

여기서 $R_{bx} = \Lambda$, $R_{bx}^H = \Lambda^H = \Lambda$ 및 $R_{bb} = \Lambda$ 이다. $\text{cov}[\Delta_w(n+1)]$ 를 $v(n+1)$ 로 놓자.

$$\begin{aligned} v(n+1) &= (I - 4\mu \Lambda + 8\mu^2 \Lambda \Lambda + 4\mu^2 \Lambda \Lambda^H) v(n) + 4\mu^2 \sigma_\eta^2 \Lambda \\ &= B v(n) + 4\mu^2 \sigma_\eta^2 \Lambda \end{aligned} \quad (3.21)$$

$$\Lambda \text{tr}[\Lambda] = \lambda \lambda^H \quad (3.22)$$

$$B = \begin{cases} 1 - 4\mu \lambda_i + 8\mu^2 \lambda_i^2 + 4\mu^2 \lambda_i^2, & \text{for } i=j \\ 4\mu^2 \lambda_i \lambda_j, & \text{for } i \neq j \end{cases} \quad (3.23)$$

식 (3.21)에서 $v(n)$ 이 수렴할 필요조건은 $0 < \sum_{j=0}^{M-1} b_{ij} < 1$ 을 유지해야 한다.

$$\sum_{j=0}^{M-1} b_{ij} = 1 - 4\mu \lambda_i + 8\mu^2 \lambda_i^2 + 4\mu^2 \lambda_i \sum_{j=0}^{M-1} \lambda_j \quad (3.24)$$

그러므로 $\Delta_w(n)$ 의 covariance 해석에서 μ 의 범위는 다음과 같음을 알 수 있다.

$$0 < \mu < \frac{1}{2\lambda_{\max} + \sum_{j=0}^M \lambda_j} < \frac{1}{\sum_{j=0}^M \lambda_j} = \frac{1}{\text{Tr}[R_{bx}]} \quad (3.25)$$

3.4 오차신호 특성

오차신호는 다음과 같이 정의할 수 있다^{[11][13]}.

$$e(n) = d(n) - w^H(n)x(n) + \eta(n) \quad (3.26)$$

여기서 $d(n)$ 은 잡음이 포함되지 않은 원하는 응답이다. 그리고 $\eta(n)$ 은 평균이 0이고, 분산은 σ_η^2 이다. 원하는 응답 $d(n)$ 과 출력 $y(n)$ 은 다음과 같이 정의된다.

$$d(n) = h^H x_\infty(n) + \eta(n) \quad (3.27)$$

$$y(n) = k^H(n)b(n) = w^H(n)x(n) \quad (3.28)$$

식(2.1)의 비용함수 $J(k)$ 를 편의상 J 로 표현한다.

$$\begin{aligned} J &= E[|e^2(n)|] \quad (3.29) \\ &= E[(h^H x_\infty(n) + \eta(n) - w^H(n)x(n)) \\ &\quad (x_\infty^H(n)h + \eta^*(n) - b^H(n)k(n))] \end{aligned}$$

위의 식을 전개하여 정리한 후 $\eta(n)$ 은 $b(n)$ 및 $x(n)$ 와 직교하므로 다음과 같이 정리될 수 있다.

$$\begin{aligned} J &= E[h^H x_\infty(n)x_\infty^H(n)h] - E[h^H x_\infty(n)b^H(n)k(n)] \quad (3.30) \\ &\quad + E[\eta(n)\eta^*(n)] - E[w^H(n)x(n)x_\infty^H(n)h] \\ &\quad + E[w^H(n)x(n)b^H(n)k(n)] \end{aligned}$$

위의 식에서 $w(n)$ 와 $k(n)$ 이 모두 포함되어 있기 때문에 $k(n)$ 을 소거하자. $E[w(n)] = R_{bx}^{-1} p_{bd}$ 를 이용해 마지막 항의 $E[w^H(n)x(n)b^H(n)k(n)]$ 을 정리하면 다음 식을 얻을 수 있다.

$$\begin{aligned} J &= E[h^H x_\infty(n)x_\infty^H(n)h] - E[h^H x_\infty(n)b^H(n)k(n)] \quad (3.31) \\ &\quad + E[\eta(n)\eta^*(n)] - w^H(n)E[x(n)x_\infty^H(n)h] \\ &\quad + p_{bd}^H R_{bx}^{-H} E[x(n)b^H(n)]k(n) \end{aligned}$$

여기서 $R_{bx}^{-H} E[x(n)b^H(n)] = R_{bx}^{-H} R_{bx}^H = I$ 이므로 다음과 같이 정리할 수 있다.

$$\begin{aligned} J &= E[h^H x_\infty(n)x_\infty^H(n)h] - E[h^H x_\infty(n)b^H(n)k(n)] \quad (3.32) \\ &\quad + E[\eta(n)\eta^*(n)] - w^H(n)E[x(n)x_\infty^H(n)h] + p_{bd}^H k(n) \end{aligned}$$

위의 식에서 마지막 항 $p_{bd}^H k(n)$ 를 소거하기 위해 $d(n) = h^H x_\infty(n)$ 이므로 p_{bd} 는 다음과 같이 정리 될 수 있다.

$$p_{bd} = E[b(n)d^*(n)] = E[b(n)x_\infty^H(n)h] \quad (3.33)$$

식(3.32)에서 $E[h^H x_\infty(n)b^H(n)k(n)] = p_{bd}^H k(n)$ 이므로 위 식의 2번째 항과 마지막 항은 제거될 수 있다. 또한 $E[\eta(n)\eta^*(n)] = \sigma_\eta^2$ 이므로 다음과 같이 정리된다.

$$J = E[h^H x_\infty(n)x_\infty^H(n)h] + \sigma_\eta^2 - w^H(n)E[x(n)x_\infty^H(n)h] \quad (3.34)$$

위 식을 다음처럼 콘볼루션 형태로 나타낼 수 있다.

$$J = E\left[\sum_{i=1}^{\infty} h_i(n)x(n-i+1)\sum_{j=1}^{\infty} x(n-j+1)h_j(n)\right] - E\left[\sum_{i=1}^M w_i(n)x(n-i+1)\sum_{j=1}^M x(n-j+1)h_j(n)\right] + \sigma_\eta^2 \quad (3.35)$$

위의 식에서 $x(n)$ 이 백색 가우시안 잡음이라면 다음과 같이 정리된다.

$$J = E\left[\sum_{i=1}^{\infty} h_i(n)x(n-i+1)\right]^2 - E\left[\sum_{i=1}^M w_i(n)x(n-i+1)\right]^2 + \sigma_\eta^2 \quad (3.36)$$

위의 식에서 반복시간 n 이 ∞ 이라면 J 는 다음과 같이 쓸 수 있다.

$$J = E\left[\sum_{i=M+1}^{\infty} h_i(n)x(n-i+1)\right]^2 + \sigma_\eta^2 \quad (3.37)$$

$$= \sum_{i=M+1}^{\infty} h_i^2(n)\sigma_x^2 + \sigma_\eta^2$$

$$h_i(n) = w_i(n), \quad \text{for } i=1,2,\dots,M \quad (3.38)$$

위 식의 J 는 최소평균자승오차 J_{\min} 이 된다.

3.5 Excess Mean-Square Error and Misadjustment
 최적계수벡터 w_{opt} 와 이것의 예측값과 $w(n)$ 과의 차이를 계수 편차(deviation of weight vector) $\Delta_w(n)$ 를 식(3.4)에서 다음과 같이 정의하였다. 또한 최적 regression 벡터 k_o 와 이것의 예측값과 $k(n)$ 과의 차이를 식(3.40)로 정의하자

$$\Delta_w(n) = w(n) - w_{opt} \quad (3.39)$$

$$\Delta_k(n) = k(n) - k_o \quad (3.40)$$

식(2.1)의 비용 함수를 정의하기 위하여 두 개의 오차를 정의하면 다음과 같다.

$$e(n) = d(n) - w^H(n)x(n) = e_o^x(n) - \Delta_w^H(n)x(n) \quad (3.41)$$

$$e(n) = d(n) - k^H(n)b(n) = e_o^b(n) - \Delta_k^H(n)b(n) \quad (3.42)$$

식(3.41)과 식(3.42)의 최적계수를 이용할 오차 $e_o^x(n)$ 과 $e_o^b(n)$ 은 같다고 가정하였다.

$$e_o(n) = e_o^x(n) = e_o^b(n) \quad (3.43)$$

식(3.41), 식(3.42) 및 식(3.43)를 이용하여 오차의 제곱을 계산하면 다음과 같이 얻어진다.

$$e^2(n) = e_o^2(n) - e_o(n)b^H(n)\Delta_k(n) - \Delta_w^H(n)x(n)e_o^*(n) + \Delta_w^H(n)x(n)b^H(n)\Delta_k(n) \quad (3.44)$$

위 식에 기대값을 취하고 이것을 비용함수로 놓자.

$$J = E[e^2(n)] = J_{\min} - E[e_o(n)b^H(n)]E[\Delta_k(n)] - E[\Delta_w^H(n)]E[x(n)e_o^*(n)] + E\{tr[\Delta_w^H(n)x(n)b^H(n)\Delta_k(n)]\} \quad (3.45)$$

위에서 $E[e_o(n)b^H(n)] \approx 0$, $E[x(n)e_o^*(n)] \approx 0$ 이므로, $e_o(n)$ 은 $x(n)$, $b(n)$ 과 직교한다.

$$J = J_{\min} + E\{tr[x(n)b^H(n)\Delta_w^H(n)\Delta_k(n)]\} \quad (3.46)$$

식(3.46)에서 $E\{tr[\cdot]\}$ 을 정리하면 다음과 같은 식을 얻는다.

$$tr\{E[b(n)x^H(n)]E[\Delta_w^H(n)\Delta_k(n)]\} = tr\{R_{bx}E[\Delta_k(n)\Delta_w^H(n)]\} \quad (3.47)$$

식(3.46)에 $\Delta_k(n)$ 과 $\Delta_w(n)$ 의 요소가 모두 나타나므로 $\Delta_k(n)$ 를 제거하자.

$$tr\{R_{bx}E[\Delta_k(n)\Delta_w^H(n)]\} = tr\{R_{bx}QQ^{-1}E[\Delta_k(n)\Delta_w^H(n)]\} = tr\{R_{bx}QE[Q^{-1}\Delta_k(n)\Delta_w^H(n)]\} = tr\{R_{bx}QE[\Delta_w(n)\Delta_w^H(n)]\} = tr\{\Lambda E[\Delta_w(n)\Delta_w^H(n)]\} \quad (3.48)$$

식(2.3)을 이용하면 $Q^{-1}\Delta_k(n)$ 및 $Q^{-1}w_{opt}$ 은 다음과 같음을 알 수 있다.

$$Q^{-1}\Delta_k(n) = Q^{-1}k(n) - Q^{-1}k_o = w(n) - w_o = \Delta_w(n) \quad (3.49)$$

$$Q^{-1}w_{opt} = k_o \quad (3.50)$$

식(3.48)을 식(3.46)에 대입 정리하면 다음과 같은 식을 얻을 수 있다.

$$J = J_{\min} + tr\{\Lambda E[\Delta_w(n)\Delta_w^H(n)]\} = J_{\min} + tr\{\Lambda cov[\Delta_w(n)]\} \quad (3.51)$$

위식에서 $J - J_{\min}$ 을 ΔJ 로 정의하자.

$$\begin{aligned} \Delta J &= J - J_{\min} = \text{tr}[A \text{cov}[\Delta_w(n)]] \quad (3.52) \\ &= \sum_{i=1}^M \lambda_i v_i(n) = \lambda^H v(n) \end{aligned}$$

위의 식에서 $\sum_{i=1}^M \lambda_i v_i(n)$ 의 표현을 구하기 위하여 식(3.21)을 이용하자..

$$v_i(n+1) - v_i(n) = (-4\mu\lambda_i + 8\mu^2\lambda_i^2)v_i(n) + 4\mu^2\lambda_i \sum_{j=1}^M \lambda_j v_j(n) + 4\mu^2\sigma_\eta^2\lambda_i \quad (3.53)$$

여기서 $v_i(n+1) \approx v_i(n)$ for large n이라고 가정하자.

$$0 = (-4\mu\lambda_i + 8\mu^2\lambda_i^2)v_i(n) + 4\mu^2\lambda_i \sum_{j=1}^M \lambda_j v_j(n) + 4\mu^2\sigma_\eta^2\lambda_i \quad (3.54)$$

식(3.54)로부터 $\sum_{j=1}^M \lambda_j v_j(n)$ 에 대한 표현을 다음과 같이 얻을 수 있다.

$$\begin{aligned} \sum_{i=1}^M \lambda_i v_i(n) &= \frac{\sum_{i=1}^M 2\mu\lambda_i^2 v_i(n) + \sum_{i=1}^M \mu\sigma_\eta^2 \lambda_i}{1 - \mu \sum_{j=1}^M \lambda_j} = \frac{\sum_{i=1}^M \mu\sigma_\eta^2 \lambda_i}{1 - \mu \sum_{j=1}^M \lambda_j} \\ &= \frac{\mu\sigma_\eta^2 \text{tr}[R_{bx}]}{1 - \mu \text{tr}[R_{bx}]} \quad (3.55) \end{aligned}$$

Excess Mean Square Error는 다음과 같다.

$$J_{exc} = \lim_{n \rightarrow \infty} \Delta J = \frac{\mu\sigma_\eta^2 \text{tr}[R_{bx}]}{1 - \mu \text{tr}[R_{bx}]} \approx \mu\sigma_\eta^2 \text{tr}[R_{bx}] = \mu(M)\sigma_\eta^2\sigma_x^2 \quad (3.56)$$

Misadjustment(M_{misad})는 다음과 같이 정의된다.

$$M_{misad} = \frac{J_{exc}}{J_{\min}} = \frac{\mu\sigma_\eta^2 \text{tr}[R_{bx}]}{1 - \mu \text{tr}[R_{bx}]} = \frac{\mu \text{tr}[R_{bx}]}{1 - \mu \text{tr}[R_{bx}]} \quad (3.57)$$

IV. 컴퓨터 시뮬레이션

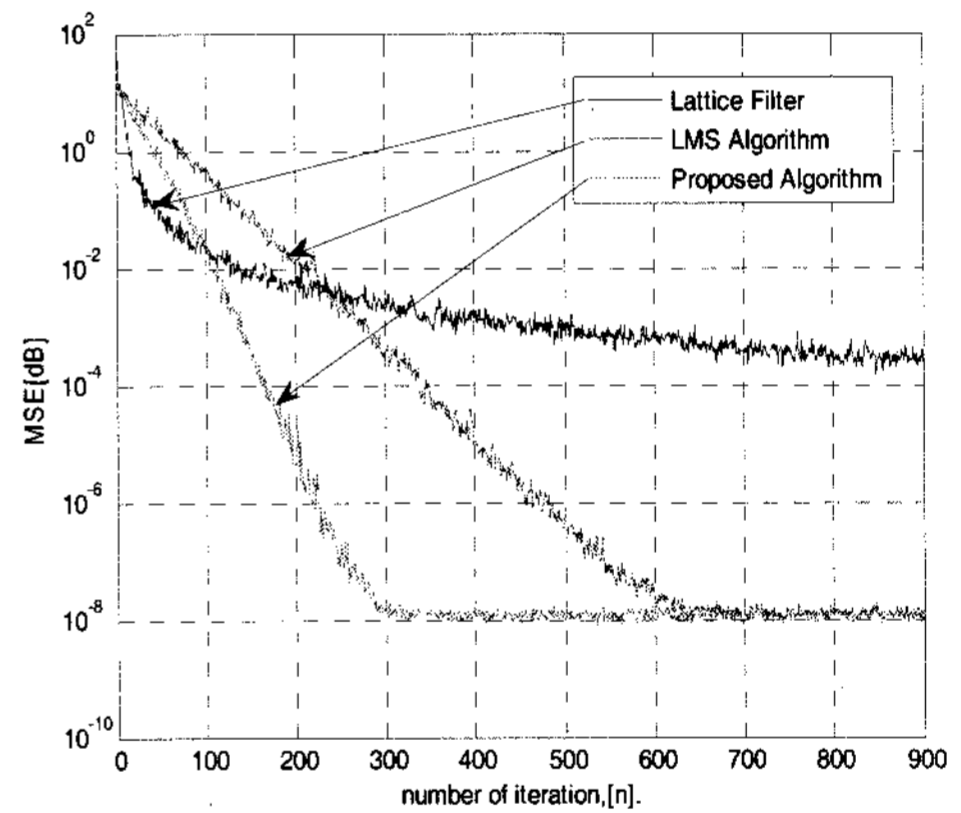
입력 $u(n)$ 은 백색가우시안 잡음으로 평균이 0 이고 분산 1이다. 유색잡음 $x(n)$ 은 $u(n)$ 을 다음과 같은 필터를 통과 시켜 얻는다.

$$H(z) = 1/(1 - 0.65z^{-1} + 0.693z^{-2} - 0.22z^{-3} + 0.309z^{-4} - 0.177z^{-5}) \quad (4.1)$$

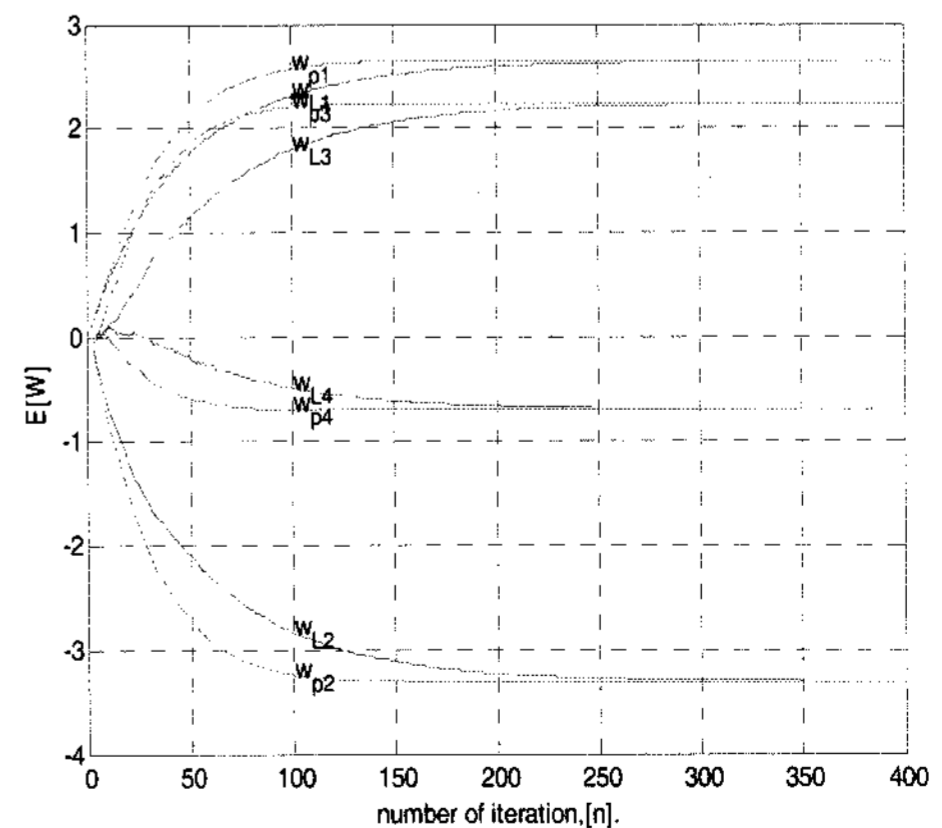
측정잡음 $v(n)$ 은 입력 $u(n)$ 과 상관관계가 없는 잡음으로 입력 $u(n)$ 에 대하여 적절한 S/N비를 갖도록 하였다. 원하는 응답은 유색잡음 $x(n)$ 을 식 (4.2)와 같은 시스템에 통과시켜 얻은 출력에 적절한 S/N비를 갖는 $v(n)$ 을 첨가하여 얻었다.

$$H(z) = 2.65 - 3.31z^{-1} + 2.24z^{-2} - 0.7z^{-3} \quad (4.2)$$

각 실험은 9000개의 데이터를 이용하였으며 100 번의 독립된 실험 평균하여 각각의 결과들을 산출하였다. 식(3.25)에서 $\mu = 1/(4\text{tr}[R_{bx}]) = 0.044$ 정도로 결정하였다. 격자단의 forgetting factor $\rho = 0.9998$ 로 하였다. 그림 2, 그림 3 및 그림 4의 (a)는 참고문헌 [2]의 격자필터의 학습곡선(검정색)과 LMS 알고리즘의 학습곡선(파란색) 및 제안한 알고리즘(빨강색)을 동시에 나타낸 것이고 (b)의 그림은 LMS 알고리즘의 계수(파란색) 및 제안한 알고리즘(빨강색)의 계수를 동시에 나타낸 것이다. 이때 $\rho = 0.9998$ 이다. S/N=-80dB인 그림2의 (a)에서 제안한 알고리즘은 약 300 반복구간에 수렴하고 LMS 알고리즘은 약 630 반복구간에서 수렴함을 알 수 있다. (b)의 계수들의 궤적에서는 제안한 알고리즘 및 LMS 알고리즘이 최적 값에 수렴함을 알 수 있다. S/N=-30dB인 그림3의 (a)에서 제안한 알고리즘은 약 180 반복구간에서

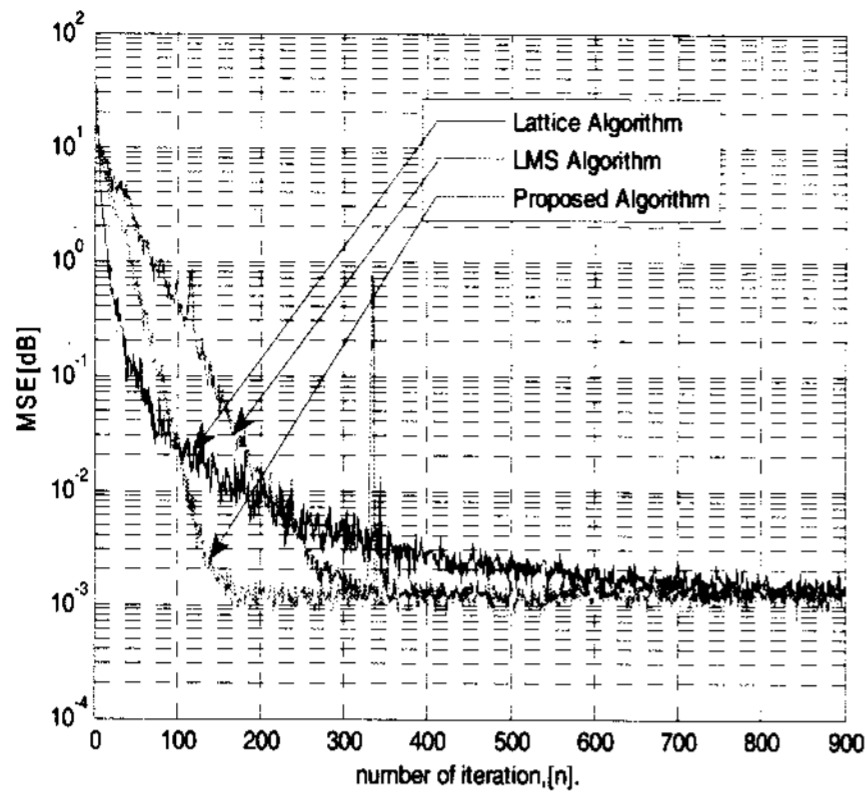


(a) 학습곡선(Learning Curves)

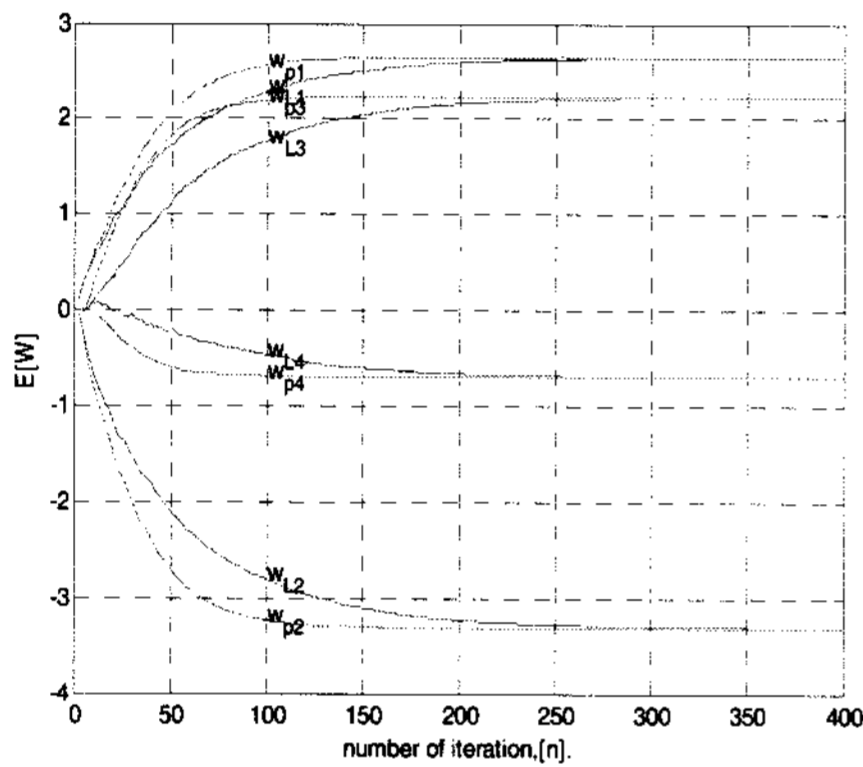


(b) 계수(Coefficients)

그림 2. S/N=-80dB인 경우 알고리즘들의 성능비교
Fig. 2. Performance comparison various algorithm with S/N=-80dB



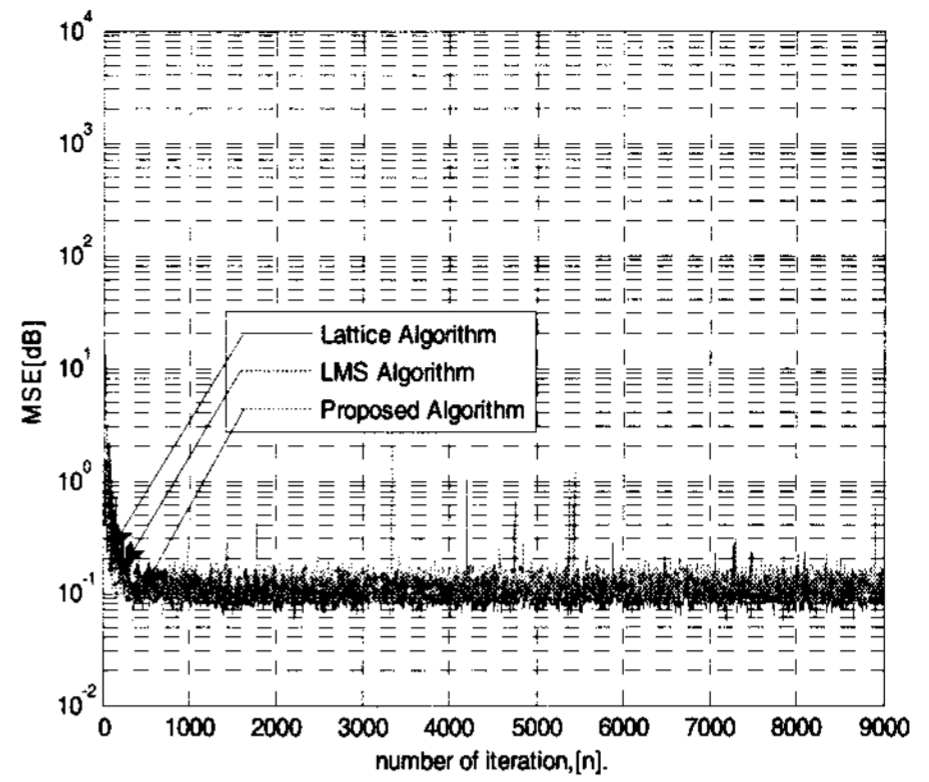
(a) 학습곡선(Learning Curves)



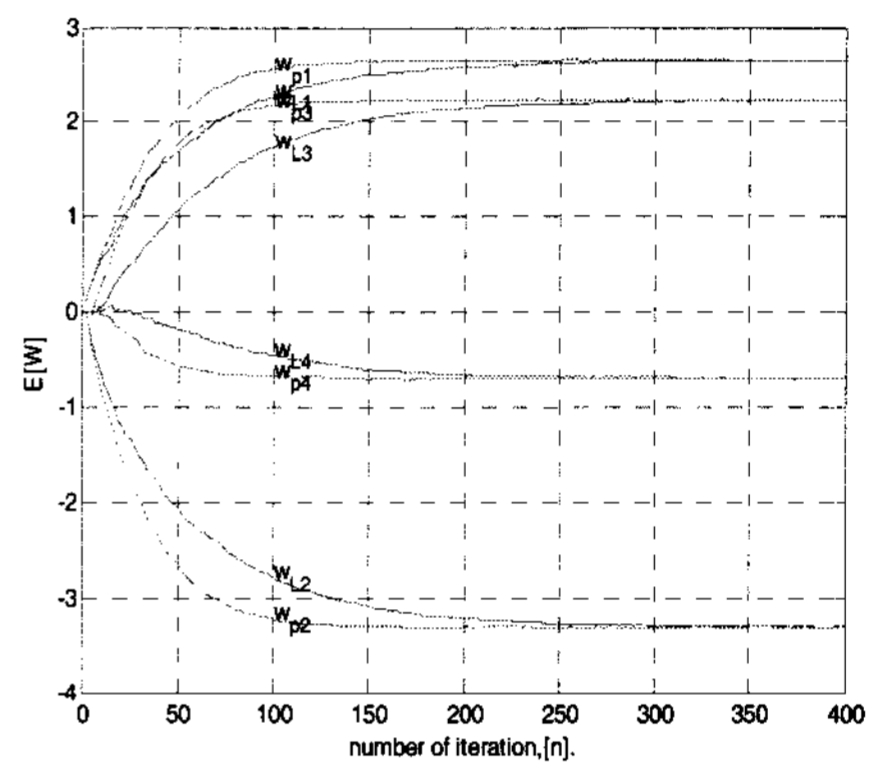
(b) 계수(Coefficients)

그림 3. S/N=-30dB인 경우 여러 알고리즘들의 성능 비교
Fig. 3. Performance comparison various algorithms with S/N=-30dB.

수렴하고 LMS 알고리즘은 약 310 반복구간에서 수렴함을 알 수 있다. 그리고 약 350 반복구간에서 학습곡선이 갑자기 커져다가 다시 수렴함을 알 수 있다. (b)의 계수들의 궤적에서는 제안한 알고리즘 및 LMS 알고리즘이 최적값에 잘 수렴함을 알 수 있다. S/N=-10dB인 그림3의 (a)에서 제안한 알고리즘은 약 100 반복구간에서 수렴하고 LMS 알고리즘은 약 180 반복구간에서 수렴함을 알 수 있다. 그리고 약 350 반복구간에서 학습곡선이 갑자기 커졌다가 다시 수렴함을 알 수 있다. (b)의 계수들의 궤적에서는 제안한 알고리즘 및 LMS 알고리즘이 최적값에 잘 수렴함을 알 수 있다. 결과적으로 제안한 알고리즘이 S/N비에 관계없이 가장 우수한 성능을 나타내고 있음을 확인할 수 있다. 그림 5의 (a), (b)는 $\mu=0.044$ 로 하고 $\rho=0.998$ 와 $\rho=0.98$ 인 경우의 학습곡선을 나타냈다. $\rho=0.9998$ 인 경우는 그림.2(a)에 나타나 있



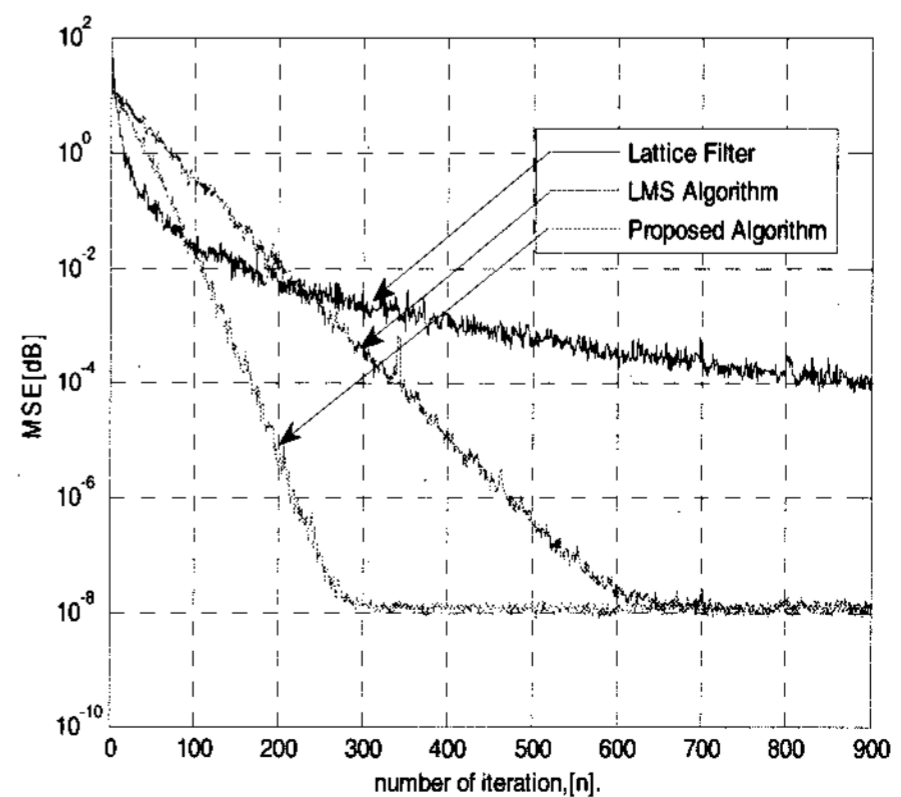
(a) 학습곡선 (Learning Curves)



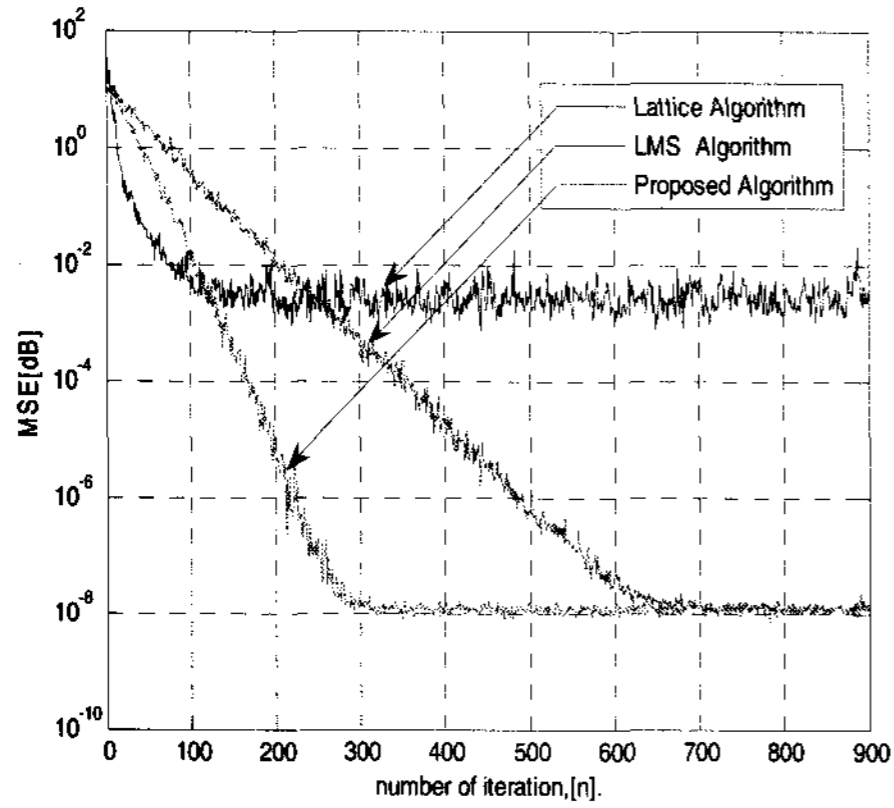
(b) 계수(Coefficients)

그림 4. S/N=-10dB인 경우 여러 알고리즘들의 성능 비교
Fig. 4. Performance comparison various algorithms with S/N=-10dB.

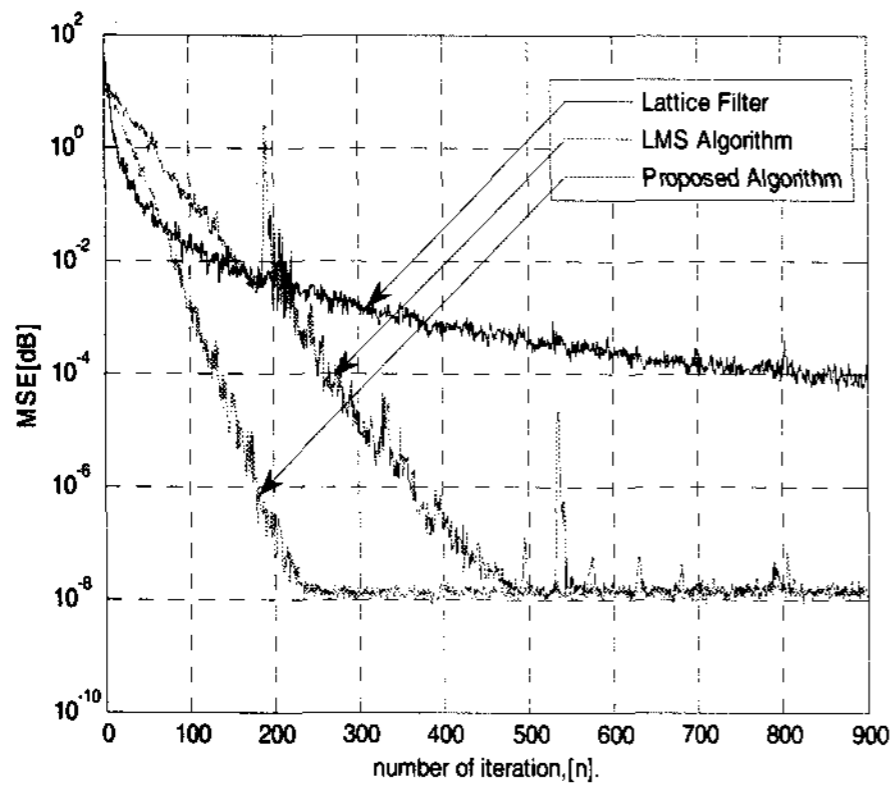
다. 그림 5의 (c), (d)에서 $\mu=0.0586$ 로 고정하고 $\rho=0.998$ 와 $\rho=0.98$ 인 경우 학습곡선을 나타낸 그림이다. 제안한 알고리즘과 LMS 알고리즘은 격자단에서 생성하는 직교입력벡터의 영향에는 둔감하고 μ 에 영향을 많이 받음을 알 수 있다.



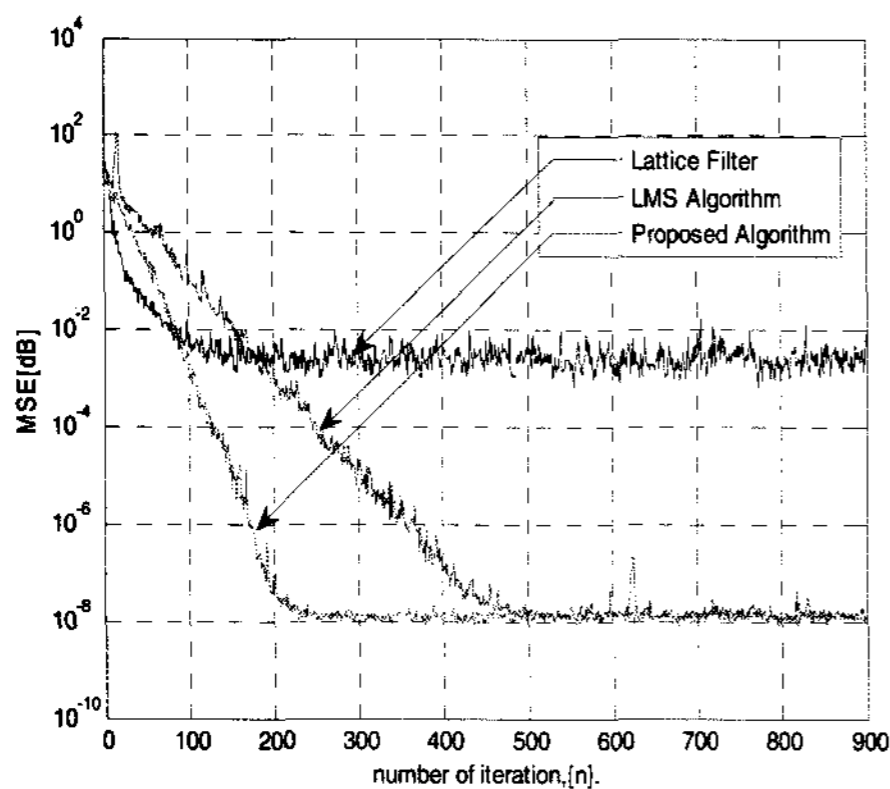
(a) $\rho=0.998, \mu=0.044$



(b) $\rho = 0.98, \mu = 0.044$



(c) $\rho = 0.998, \mu = 0.0586$



(d) $\rho = 0.98, \mu = 0.0586$

그림 5. S/N=-80dB인 경우 ρ 와 μ 에 따른 학습곡선
Fig. 5. Learning curves with ρ and μ , S/N=-80dB

V. 결 론

본 논문은 격자필터를 이용하여 생성한 직교입력신호를 등가의 Wiener-Hopf 방정식에 적용하여 그 해를 최소평균자승 알고리즘을 이용하여 순환

적으로 구하였다. 제안한 알고리즘의 계산량은 $O(M)$ 이고 unbiased된 순시예측이다. 본 논문의 학술적 또는 실용적 공헌도는 다음과 같다. 첫째로 계산량이 적은 Wiener-Hopf 버전의 약간 다른 방정식을 제안한다. Wiener-Hopf 방정식의 상관행렬은 $M \times M$ 인 full 행렬인 반면에 본 논문에서 제안한 상관행렬은 $M \times M$ 인 삼각행렬(triangular matrix)이다. 두 번째로 Levinson-Durbin 알고리즘을 사용하지 않고 격자필터에서 TDL 필터의 계수를 구하는 LMS 알고리즘을 제안하였다. 세 번째로 격자필터에서 TDL 필터의 계수를 구하는데 계산량이 적은 장점이 있다. 부가적인 장점으로는 첫 번째로 격자필터에서 필요하는 반사계수를 포함하고 오차와 TDL 필터의 계수를 모두 한번에 구할 수 있는 효율적인 알고리즘을 제안한 것이다. 그리고 제안한 알고리즘의 수렴적 특성을 이론적으로 다양하게 고찰하였다. 제안한 알고리즘의 이론적 고찰에서 회전되지 않은 형태로 그 결과들이 대부분 유도되었다. 간단한 컴퓨터 시뮬레이션에서 참고문헌 [2]의 격자필터와 LMS 알고리즘 및 제안한 알고리즘의 수렴특성을 비교하였다. 그 결과 제안한 알고리즘이 가장 우수한 특성을 나타내고 있음을 확인하였다.

참 고 문 헌

- [1] Haykin, *Adaptive Filter Theory-Fouth Edition*, Prentice Hall, 2002, pp.94-193, Chapter11-Chapter 12.
- [2] J. G. Proakis, C. M. Rader, F. Ling and C. L. Nikias, *Advanced Digital Signal Processing*. Macmillan Publishing Company, 1992, Chapter 4-Chapter 5.
- [3] F. Albu, M. Bouchard and Y Zakharov, "Pseudo-Affine Projection Algorithms for Multichannel Active Noise Control," *IEEE Transactions on Audio, Speech and Language Processing*, Vol.15, Issue 3, pp.1044 - 1052, March 2007.
- [4] Song Liu, Xiaodong Li and Jing Tian, "Transform domain adaptive filter in active noise control," *Signal Processing, 2002 6th International Conference on Vol.1*, pp.272-275, 26-30 Aug. 2002.
- [5] W. B. Mikhael and A. S. Spanias, "A fast frequency-domain adaptive algorithm," *Proceedings of the IEEE*, Vol.76, Issue 1, pp.80-82, Jan. 1988

