


직교 입력 벡터를 이용하는 수정된 RLS 알고리즘에 관한 연구

안봉만¹ , 김관웅², 안현규³, 한병성¹

¹ 전북대학교 산학협력단

² (주)썬더테크놀로지 연구소

³ 전북대학교 교육대학원 수학교육과

A Study on the Modified RLS Algorithm Using Orthogonal Input Vectors

Bong Man Ahn¹, Kwang Woong Kim², Hyun Gyu Ahn³, and Byoung Sung Han¹

¹ Industrial Cooperation Foundation, Chonbuk Nation University, Jeonju 54896, Korea

² Institute of Sundertechnology, Jeonju 55440, Korea

³ Department of Mathematics Education, Graduate School of Education, Chonbuk Nation University, Jeonju 54896, Korea

(Received October 5, 2018; Revised October 18, 2018; Accepted October 18, 2018)

Abstract: This paper proposes an easy algorithm for finding tapped-delay-line (TDL) filter coefficients in an adaptive filter algorithm using orthogonal input signals. The proposed algorithm can be used to obtain the coefficients and errors of a TDL filter without using an inverse orthogonalization process for the orthogonal input signals. The form of the proposed algorithm in this paper has the advantages of being easy to use and similar to the familiar recursive least-squares (RLS) algorithm. In order to evaluate the proposed algorithm, system identification simulation of the 11th-order finite-impulse-response (FIR) filter was performed. It is shown that the convergence characteristics of the learning curve and the tracking ability of the coefficient vectors are similar to those of the conventional RLS analysis. Also, the derived equations and computer simulation results ensure that the proposed algorithm can be used in a similar manner to the Levinson-Durbin algorithm.

Keywords: Modified RLS algorithm, Orthogonal input vectors, Inverse orthogonalization process, System identification

1. 서론

적응필터 입력은 비직교 입력 신호를 사용하는 경우와 직교 입력 신호를 사용하는 경우로 나눌 수가 있는데 전자의 경우는 대표적으로 TDL 필터들이고 후자는 경우 격자(lattice) 필터 등이 있다 [1-3]. 이 밖에 직교 입력 신호들을 이용하는 적응필터들은 그램-슈미트(Gram

Schmidt) 직교화 과정 [2], 가우스-자이델(Gauss-Seidel) 방법 [4], QR [1,2] 분해법 등을 이용하고 있다. 이러한 필터들은 입력벡터를 직교화하는 부분과 원하는 응답 벡터를 예측하는 부분이 존재하는데 하나의 예로 격자 필터의 격자단과 접합과정예측(joint-process estimation) 단이 이에 속한다. 또한 QR 분해를 이용한 알고리즘과 그램-슈미트 직교화 과정을 사용하는 적응필터에서도 접합과정예측단과 유사한 역할을 하는 단이 존재한다 [1-3].

격자필터에서 격자단과 접합과정예측단에서 구한 계수는 TDL 필터의 계수가 아니기 때문에 직교화 과정에 대한 적절한 역변환과정을 거쳐야 TDL 필터의 계수를

✉ Byoung Sung Han; hbs@jbnu.ac.kr

얻을 수 있다. 이러한 것은 그램-슈미트 직교과정, QR 등을 이용하는 알고리즘에서도 그러하다 [1-3].

일반적으로 직교 입력 벡터를 사용하는 알고리즘들은 비정상적(nonstationary)인 환경에서 TDL 필터보다 우수한 수렴 특성을 나타낸다고 알려져 있다 [1]. 그러나 우수한 수렴 특성을 가지고 있음에도 TDL 필터계수를 얻는 데 제약이 존재한다. 참고문헌 [5-7]에 쌍선형(bilinear) 필터를 다채널 격자필터로 구현한 연구가 진행되었다. 이 경우 TDL 필터의 계수를 구하는 과정이 매우 복잡하고 어렵다. 그러므로 시간 영역에서 직교 입력 벡터를 사용하는 알고리즘들에서 이러한 제약을 회피하는 방법을 연구할 필요가 있다. 이것에 관련한 연구는 참고문헌 [8-10]에서 최소평균자승 알고리즘 형태로 연구되었다.

본 논문은 입력을 직교화하여 사용하는 알고리즘들에서 TDL 계수를 쉽게 구할 수 있는 방법을 최소자승 알고리즘 방법으로 개발하였다. 개발된 알고리즘은 최소자승 알고리즘에 기반하여 개발하였다 이와 같이 개발된 알고리즘은 기존 순환최소자승(recursive-least-square, RLS) 알고리즘과 유사하다. 그리고 평가를 위하여 11차 FIR 시스템의 시스템 식별을 수행하였다. 모의실험은 다양한 신호 대 잡음(signal-to-noise, S/N) 비를 이용하여 실행하였고 그 결과를 그래프와 표로 나타내었다. 모의실험 평가 결과는 제안한 알고리즘은 격자필터의 레빈슨-더빈(Levinson-Durbin) 알고리즘 [1]과 같이 TDL 필터계수를 구할 수 있음을 확인하였다.

2. 실험 방법

2.1 직교 입력 벡터를 이용하는 수정된 RLS 알고리즘 유도

참고문헌 [8]에서 제안한 알고리즘을 간단히 고찰한 후 제안하는 순환최소자승 알고리즘을 유도한다. 그림 1에서 입력신호 벡터 $\mathbf{x}(n)$, 최적계수벡터 \mathbf{w}_{opt} 의 선형 조합에 의하여 출력 $y(n)$ 이 생성되고 여기에 측정잡음(measurement noise) $e_0(n)$ 이 첨가되어 원하는 응답(desired response) $d(n)$ 이 만들어진다.

오차 $e_1(n)$ 을 이용하면 직교 입력들을 이용하지 않는 알고리즘을 얻을 수 있고 오차 $e_2(n)$ 을 이용하면 직교 입력들을 이용하는 알고리즘을 얻을 수 있다.

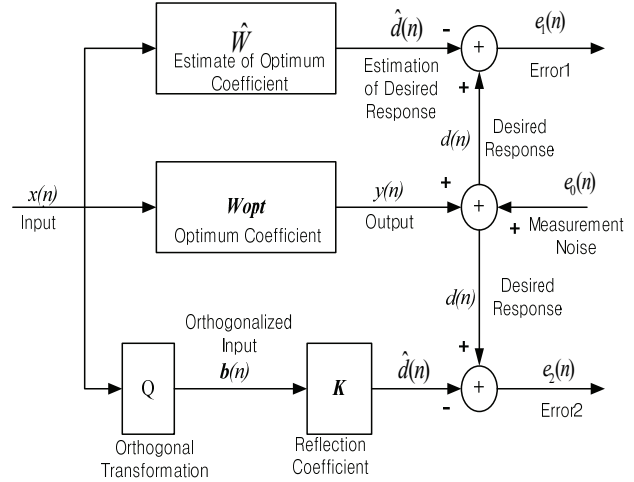


Fig. 1. Structure of proposed algorithm.

오차 $e_2(n)$ 을 얻는 과정은 입력신호 벡터 $\mathbf{x}(n)$ 에 직교변환 Q 을 적용하여 입력들이 서로 직교(orthogonal)들을 이루는 새로운 직교 입력 $\mathbf{b}(n)$ 을 얻는다. 직교 입력 $\mathbf{b}(n)$ 을 이용하고 최소자승(least square) 알고리즘을 이용하여 원하는 응답 $d(n)$ 을 이것을 예측한 값 $\hat{d}(n)$ 을 구한다. 그리고 오차 $e_2(n)$ 이 작아지도록 하는 계수벡터 \mathbf{k} 를 순환적으로 구하면 된다. 이때 측정잡음 $e_0(n)$ 은 오차 $e_2(n)$ 에 감소되지 않고 나타나게 된다. 직교 입력들을 이용하는 알고리즘들은 오차를 최소화하기 위하여 계수벡터 \mathbf{k} 를 구하기 때문에 TDL 필터의 계수벡터 $\hat{\mathbf{w}}$ 를 구하는 것은 별도의 과정을 수행하거나 별도의 차분 방정식 유도를 통하여 얻어진다.

참고문헌 [8]은 오차 $e_1(n)$ 과 $e_2(n)$ 가 확률적으로 같다고 가정한다. 이 관계를 이용한 알고리즘을 다음과 같이 제안하였다. 최소평균자승(least-mean-square) 면에서 정규 방정식을 식 (1)과 나타내었다.

$$E[\mathbf{b}(n)\mathbf{x}^H(n)]\mathbf{w}_{opt} = E[\mathbf{b}(n)d^*(n)] \quad (1)$$

식 (1)을 재정리하면 다음과 같다.

$$R_{bx}\mathbf{w}_{opt} = \mathbf{p}_{bd} \quad (2)$$

여기서

$$R_{bx} = E[\mathbf{b}(n)\mathbf{x}^H(n)] \quad (3)$$

$$\mathbf{p}_{bd} = E[\mathbf{b}(n)d^*(n)] \quad (4)$$

또한 참고문헌 [9,10]에 식 (2), 식 (3) 및 식 (4)를 이용하여 입력벡터들을 직교화하여 사용하는 알고리즘들에서 TDL 필터계수를 순환적으로 구하는 최소평균자승 알고리즘이 연구되어 있다.

식 (1)의 정규방정식을 최소자승 알고리즘 면에서 구하고 이것을 이용하여 순환최소자승 알고리즘을 유도한다.

일반적으로 순환최소자승 알고리즘의 비용함수(cost function) $\mathcal{E}(n)$ 은 다음과 같다 [1-3].

$$\mathcal{E}(n) = \sum_{i=1}^n \lambda^{(n-i)} \|e(i)\|^2 \quad (5)$$

여기서 λ ($0 < \lambda \leq 1$)는 망각계수(forgetting factor)이다. $e(n)$ 은 오차로 다음과 같다.

$$\begin{aligned} e(n) &= d(n) - \mathbf{w}^H(n)\mathbf{x}(n) \\ &= d(n) - \mathbf{k}^H(n)\mathbf{b}(n) \end{aligned} \quad (6)$$

비용함수 $\mathcal{E}(n)$ 을 계수벡터 \mathbf{w} 로 편미분하면 비용함수를 최소화하는 계수벡터가 얻어진다.

식 (5)를 최소화하는 최적의 계수벡터는 다음과 같은 정규방정식(normal equation)으로 얻어진다.

$$\Phi_{bx}(n)\hat{\mathbf{w}}(n) = \boldsymbol{\theta}_{bd}(n) \quad (7)$$

여기서

$$\begin{aligned} \Phi_{bx}(n) &= \sum_{i=1}^n \lambda^{n-i} \mathbf{b}(i)\mathbf{x}^H(i) \\ &= \lambda\Phi_{bx}(n-1) + \mathbf{b}(n)\mathbf{x}^H(n) \end{aligned} \quad (8)$$

$$\begin{aligned} \boldsymbol{\theta}_{bd}(n) &= \sum_{i=1}^n \lambda^{n-i} \mathbf{b}(i)d^*(i) \\ &= \lambda\boldsymbol{\theta}_{bd}(n-1) + \mathbf{b}(n)d^*(n) \end{aligned} \quad (9)$$

$\hat{\mathbf{w}}(n)$ 을 구하기 위해 $\Phi_{bx}(n)$ 의 역행렬을 행렬역공리(matrix inversion lemma)를 이용하여 얻는다.

$$\Phi_{bx}^{-1}(n) = \lambda^{-1}\Phi_{bx}^{-1}(n-1) - \frac{\lambda^{-2}\Phi_{bx}^{-1}(n-1)\mathbf{b}(n)\mathbf{x}^H(n)\Phi_{bx}^{-1}(n-1)}{1 + \lambda^{-1}\mathbf{x}^H(n)\Phi_{bx}^{-1}(n-1)\mathbf{b}(n)} \quad (10)$$

$$\mathbf{k}(n) = \frac{\lambda^{-1}\Phi_{bx}^{-1}(n-1)\mathbf{b}(n)}{1 + \lambda^{-1}\mathbf{x}^H(n)\Phi_{bx}^{-1}(n-1)\mathbf{b}(n)} \quad (11)$$

여기서 $\mathbf{k}(n)$ 은 이득벡터(gain vector)를 나타낸다. $\Phi_{bx}(n)$ 의 역행렬을 $P(n)$ 이라고 하면 식 (10)과 식 (11)은 다음과 같이 정리된다.

$$P(n) = \lambda^{-1}P(n-1) - \lambda^{-1}\mathbf{k}(n)\mathbf{x}^H(n)P(n-1) \quad (12)$$

$$\mathbf{k}(n) = \frac{\lambda^{-1}P(n-1)\mathbf{b}(n)}{1 + \lambda^{-1}\mathbf{x}^H(n)P(n-1)\mathbf{b}(n)} \quad (13)$$

그리고 식 (12)는 순환최소자승 알고리즘의 리카티(Riccati) 방정식이라 한다. 또한 식 (13)의 이득벡터는 다음과 같이 재표현된다.

$$\mathbf{k}(n) = P(n)\mathbf{b}(n) \quad (14)$$

식 (7), (9), (12) 및 (14)를 이용하여 최소자승 예측 계수벡터 $\hat{\mathbf{w}}(n)$ 의 시간갱신(time update)식을 얻는다.

$$\begin{aligned} \hat{\mathbf{w}}(n) &= \Phi_{bx}^{-1}(n)\boldsymbol{\theta}_{bd}(n) = P(n)\boldsymbol{\theta}_{bd}(n) \\ &= \lambda P(n)\boldsymbol{\theta}_{bd}(n-1) + P(n)\mathbf{b}(n)d(n) \\ &= \hat{\mathbf{w}}(n-1) + \mathbf{k}(n)\xi^*(n) \end{aligned} \quad (15)$$

여기서 $\xi(n)$ 은 사전예측오차(priori estimation error)를 나타낸다.

$$\xi(n) = d(n) - \hat{\mathbf{w}}^H(n-1)\mathbf{x}(n) \quad (16)$$

입력벡터 $\mathbf{x}(n)$ 에 대한 직교벡터 $\mathbf{b}(n)$ 는 참고문헌 [1]의 사후예측오차(posteriori estimation errors)를 이용한 순환최소자승격자(recursive least squares lattice, RLSL) 필터를 이용하여 얻었다.

이 과정을 그림으로 나타내면 그림 2와 같다.

제안한 알고리즘의 순환식은 식 (12), (13), (15) 및 (16)을 이용하여 얻을 수 있다. 이것을 정리하면 표 1과 같다.

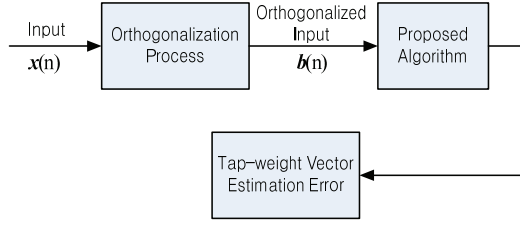


Fig. 2. Algorithm processing structure.

Table 1. Summary of the proposed algorithm.

Initialize the algorithm by setting

$$\hat{\mathbf{w}}(0) = \mathbf{0}$$

$$P(0) = \delta^{-1} I, \begin{cases} \text{small positive constant for high SNR} \\ \text{large positive constant for low SNR} \end{cases}$$

for each instant of time $n = 1, 2, 3, \dots$, compute

$$\boldsymbol{\pi}(n) = P(n-1)\mathbf{b}(n)$$

$$\mathbf{k}(n) = \frac{\boldsymbol{\pi}(n)}{\lambda + \mathbf{x}^H(n)\boldsymbol{\pi}(n)}$$

$$\xi(n) = d(n) - \hat{\mathbf{w}}^H(n-1)\mathbf{x}(n)$$

$$\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + \mathbf{k}(n)\xi^*(n)$$

$$P(n) = \lambda^{-1}P(n-1) - \lambda^{-1}\mathbf{k}(n)\mathbf{x}^H(n)P(n-1)$$

3. 결과 및 고찰

3.1 컴퓨터 모의실험 결과

컴퓨터 모의실험을 위하여 FIR 시스템의 시스템 식별 문제에 적용하였다. 백색잡음(white noise)을 발생시킨 후 이것을 다음의 저역통과필터에 통과시켜 유색잡음(colored noise)을 만들었다.

$$x(n) = 0.7x(n-1) + u(n) \quad (17)$$

여기서 입력 $u(n)$ 은 평균 0.0이고 분산이 1.0인 백색잡음이다.

시스템 식별(system identification) 문제에 적용한 FIR 시스템은 다음과 같다.

$$y(n) = 1.0x(n) - 0.5x(n-1) + 1.5x(n-2) - 1.3x(n-3) + 2.7x(n-4) - 2.3x(n-5) + 3.2x(n-6) - 3.0x(n-7) + 3.5x(n-8) + 4.0x(n-9) - 3.8x(n-10) \quad (18)$$

모든 모의실험은 서로 다른 입력을 이용하여 100번의 독립된 실험을 수행한 후 앙상블 평균(ensemble

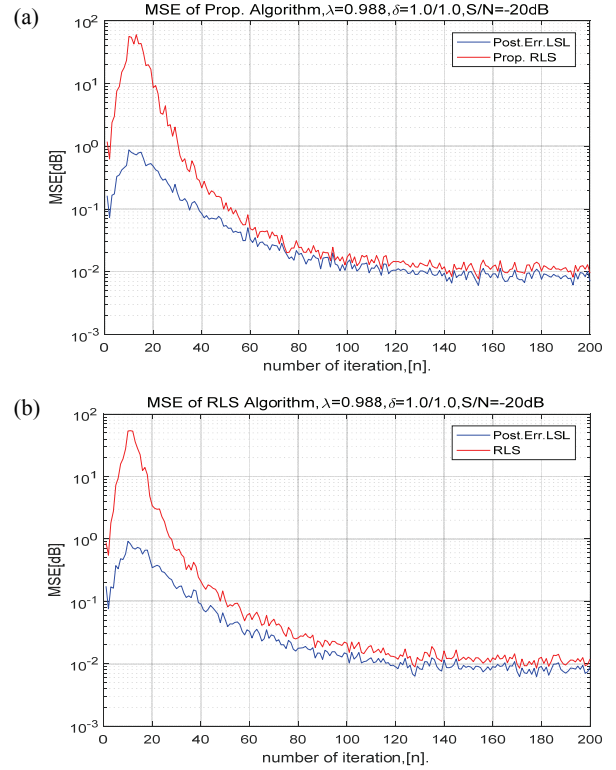


Fig. 3. Learning curves for proposed algorithm with S/N = -20 dB. (a) Learning curve of the proposed algorithm and the RLSL filter and (b) learning curve of RLS algorithm and RLSL filter.

average)하여 그 결과를 나타내었다. 이때 망각계수는 $\lambda = 0.988$ 로 하였다.

그림 3의 (a)와 (b)는 측정잡음이 -20 dB인 경우 순환최소자승격자 알고리즘('Post.Err.LSL'라 함)과 제안한 알고리즘('Prop.RLS'라 함) 알고리즘 및 순환최소자승('RLS'라 함) 알고리즘과의 학습곡선을 나타낸 것이다. 그림 3의 (a)와 (b) 학습곡선 비교에서 제안한 알고리즘과 순환최소자승 알고리즘이 초기에 오버슈트(overshoot)가 크게 나타난 것은 큰 문제가 아니다. 이것은 학습곡선들이 비슷하도록 초기값을 설정하였기 때문이다. 수렴된 평균자승오차 값도 약간 크게 나타남을 보여주고 있다.

참고문헌 [1]에서 순환최소자승 알고리즘의 학습곡선을 다음과 같이 나타내고 있다.

$$J(n) = \sigma^2 + \frac{M\sigma^2}{n-M-1}, \quad n > M+1 \quad (19)$$

여기서 $\sigma^2 = E[e_0(n)^2]$ 이고 $e_0(n)$ 은 측정오차이다. 그

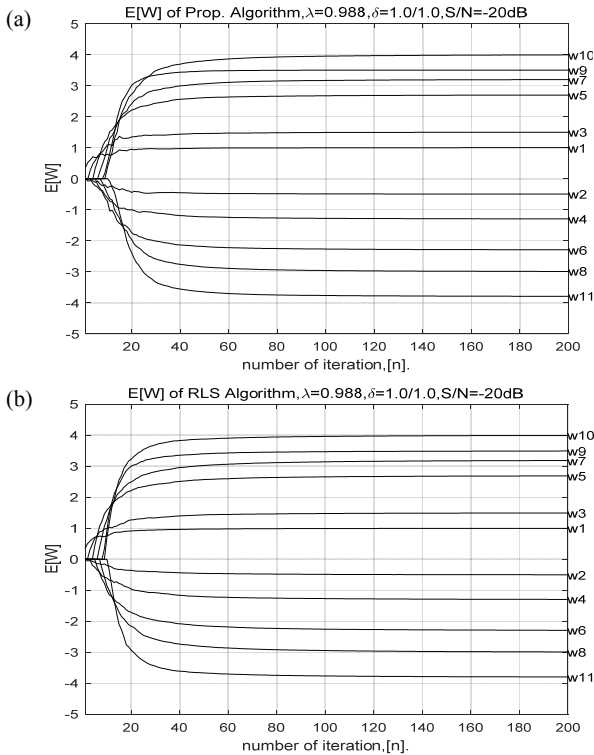


Fig. 4. $E[\hat{\mathbf{w}}]$ curves with S/N = -20 dB. (a) Proposed algorithm and (b) RLS algorithm.

리고 M 은 필터의 차수이다.

식 (19)에서 반복횟수 $n \rightarrow \infty$ 이면 측정오차의 분산, 즉 S/N 값을 나타낸다는 것이다.

그림 3의 (a)와 (b)에서 반복횟수가 약 $n = 160$ 일 때 수렴되었음을 나타내고 있고 이후 반복과정에서 평균제곱오차 값이 -20 dB를 나타내고 있어 식 (19)를 만족한다. 또한 참고문헌 [1]에서 순환최소자승 알고리즘의 계수벡터의 평균값에 대한 수렴 특성은 다음 식과 같이 나타내고 있다.

$$E[\hat{\mathbf{w}}(n)] = \mathbf{w}_o, n \geq M \quad (20)$$

여기서 \mathbf{w}_o 는 최적계수벡터이다.

그리고 계수벡터의 평균제곱오차 값은 식 (21)과 같이 나타내고 있다.

$$E[\boldsymbol{\epsilon}^H(n)\boldsymbol{\epsilon}(n)] = \frac{\sigma^2}{n-M-1} \sum_{i=1}^M \frac{1}{\lambda_i}, n > M+1 \quad (21)$$

여기서 $\boldsymbol{\epsilon}(n) = \hat{\mathbf{w}}(n) - \mathbf{w}_o$ 이고 λ_i 는 상관행렬 $\Phi_{ww}(n)$

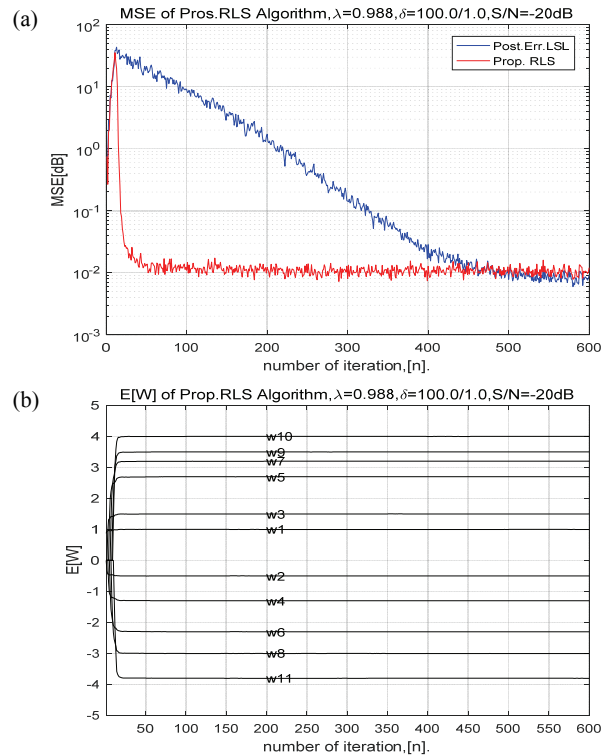


Fig. 5. The various curves with $\delta = 100$ and S/N = -20 dB. (a) Learning curves with $\delta = 100$ and (b) $E[\hat{\mathbf{w}}]$ curves of proposed algorithm with $\delta = 100$.

의 고유값(eigenvalue)이다.

식 (21)은 $\hat{\mathbf{w}}(n)$ 의 평균제곱오차 값이 상관행렬 $\Phi_{ww}(n)$ 의 가장 작은 고유값에 영향을 받는다는 것을 의미한다. 그러므로 알고리즘의 초기값을 적절히 선택해야 한다.

그림 4의 (a)와 (b)에서 $E[\hat{\mathbf{w}}]$ 곡선들이 식 (20)을 매우 만족한다는 것을 나타내고 있다. 그리고 제안한 알고리즘과 순환최소자승 알고리즘에서 초기에 수렴이 늦은 것은 상관행렬의 초기값들이 상관행렬의 고유값에 영향을 준 것으로 판단된다.

그림 5에 초기값 설정이 상관행렬의 고유값에 영향을 미치는 것을 나타내고 있다. 그림 5는 제안한 알고리즘의 상관행렬의 초기값과 순환최소자승 격자필터의 순방향 및 역방향 오차제곱의 초기값 δ 를 1에서 100으로 변화시킨 후 모의실험을 수행한 결과를 나타낸 그림이다. 그림 3의 (a)에서는 순환최소자승 격자필터 성능이 우수한 것같이 보이나 그림 5에서는 제안한 알고리즘이 더 우수함을 나타내고 있다. 그러므로 그림 5의 결과만 가지고 제안한 알고리즘이 우수하다고 주장할 수 없다.

Table 2. Mean values of $E[\hat{\mathbf{w}}]$ with various S/N ratio.

Classific.	w_0	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}
Opt. value	1.0	-0.5	1.5	-1.3	2.7	-2.3	3.2	-3.0	3.5	4.0	-3.8
-60 dB	1.001139	-0.500434	1.501584	-1.301236	2.702838	-2.302298	3.203319	-3.003041	3.504274	4.004094	-3.804210
-30 dB	1.001023	-0.500449	1.501804	-1.301599	2.702866	-2.301939	3.203087	-3.002968	3.504262	4.003903	-3.804104
-20 dB	1.001254	-0.500905	1.502210	-1.302121	2.703207	-2.302385	3.203978	-3.003562	3.504466	4.003721	-3.803945
-5 dB	0.999222	-0.502209	1.503217	-1.302608	2.705588	-2.303794	3.208567	-3.007067	3.506572	4.002217	-3.801812

Table 3. The average value of $E[\hat{\mathbf{w}}]$ accord. to the filter order mismatch.

Classific.	w_0	w_1	w_2	w_3	w_4	w_5	w_6	w_7
Opt. value	1.0	-0.5	1.5	-1.3	2.7	-2.3	3.2	-3.0
-20 dB	0.991537	-0.509616	1.525823	-1.307416	2.690024	-2.289809	3.164931	0.096068

표 2는 1,000개의 데이터로 100번 시뮬레이션 결과를 앙상블 평균한 다음 수렴 후 800개의 계수벡터들의 데이터를 시간 평균하여 나타낸 것이다. 표 2는 제안한 알고리즘이 계수벡터의 최적값을 잘 식별하고 있는 것을 나타내고 있다.

그리고 표 3은 미지의 시스템을 11차 FIR 시스템으로 하여 원하는 응답을 생성한 후 제안한 알고리즘을 이용하여 8차로 미지의 시스템 식별을 수행한 결과이다. 각 계수들은 수렴 후 800개의 데이터를 시간 평균한 값들이다. 각각의 값을 최적오차와 비교하면 \hat{w}_6 에서 \hat{w}_7 까지는 비교적 미지의 시스템을 잘 식별하나 \hat{w}_7 의 값은 오차가 있음을 알 수 있었다.

이상의 평가 결과에서 제안한 알고리즘에 의하여 계산된 $\hat{\mathbf{w}}(n)$ 의 값들이 최적값을 잘 추적(tracking)하는 것을 확인하였다. 기존 연구들에서 격자필터로부터 TDL 필터의 계수를 구하는 방법으로 레빈슨-더빈 알고리즘 [1]이 널리 사용되어 왔다.

본 연구에서 제안한 알고리즘 역시 격자필터로부터 TDL 필터의 계수를 정확하게 구할 수 있음을 수식과 모의실험을 통하여 보였다. 이와 같은 사실로부터 제안한 알고리즘이 레빈슨-더빈 알고리즘과 같이 TDL 필터계수를 구할 수 있음을 확인하였다.

4. 결론

본 논문은 입력 신호들을 직교화하여 사용하는 알고리즘들에서 직접적으로 계수벡터를 구하는 방법을 최

소자승 측면에서 제안하였다. 제안한 알고리즘이 기존에 사용하는 순환최소자승 알고리즘과 유사하여 사용하는 데 있어서 이점이 있다.

컴퓨터 모의실험은 11차 FIR 시스템의 시스템 식별 문제에 적용하였다. 학습곡선 비교는 사후 예측오차를 사용한 순환최소자승 격자필터 알고리즘 및 순환최소자승 알고리즘과 비교하였다. 그 결과 제안한 알고리즘이 초기에 오버슈트(overshoot) 값이 격자필터 알고리즘에 비하여 존재함을 알 수 있었다. 이것은 적절한 초기값 설정으로 문제임을 제시하였다. 그리고 수렴 후 MSE 값은 순환최소자승 알고리즘의 이론값과 같게 나타남을 보여주고 있다. 또한 제안한 알고리즘이 우수한 최적계수 추적 능력을 가지고 있는 것을 확인하였다. 모의실험 결과 레빈슨-더빈 알고리즘 [1]과 같이 TDL 필터계수를 구할 수 있음을 확인하였다.

향후 제안한 알고리즘의 초기값 결정과 수렴 특성 해석에 대한 연구가 진행되어야 한다. 그리고 응용분야에 대한 연구도 진행되어야 한다.

ORCID

Bong Man Ahn

<https://orcid.org/0000-0001-7460-3742>

감사의 글

본 논문은 전북테크노파크 “2017년 기술인프라연계 기술개발사업: 과제번호-201709014”의 지원으로 수행되었습니다.

REFERENCES

- [1] S. Haykin, *Adaptive Filter Theory-Fourth Edition* (Prentice Hall, New Jersey, 2002), p. 146.
- [2] J. G. Proakis, C. M. Rader, F. Ling, and C. L. Nikias, *Advanced Digital Signal Processing* (Macmillan Publishing Company, New York, 1992). p. 401.
- [3] P.S.R. Diniz, *Adaptive Filtering, Algorithms and Practical Implementation, Second Edition* (Kluwer Academic Publishers Group, Massachusetts, 2002). p. 16.
- [4] F. Albu, M. Bouchard, and Y. Zakharov, *IEEE Trans. Audio Speech Lang. Process.*, **15**, 1044 (2007). [DOI: <https://doi.org/10.1109/TASL.2006.881677>].
- [5] H. K. Baik, *J. Inst. Electron. Eng. Korea B*, **29**, 26 (1992).
- [6] H. K. Baik, *J. Inst. Electron. Eng. Korea B*, **29**, 34 (1992).
- [7] B. M. Ahn and H. K. Baik, *J. Inst. Electron. Eng. Korea B*, **32**, 80 (1995).
- [8] B. M. Ahn, J. W. Hwang, and J. P. Cho, *J. Inf. Commun. Convergence Eng.*, **33**, 743 (1999).
- [9] B. M. Ahn, J. W. Hwang, and J. P. Cho, *J. Inf. Commun. Convergence Eng.*, **33**, 403 (2008).
- [10] B. M. Ahn, J. W. Hwang, and J. P. Cho, *J. Inf. Commun. Convergence Eng.*, **33**, 562 (2008).