한국음향학회지 제29권 제1호 pp. 18~23 (2010)

정규화 기법을 이용한 낮은 연산량의 가변 망각 인자 RLS 기법

Low-Complexity VFF-RLS Algorithm Using Normalization Technique

이 석 진*, 임 준 석**, 성 굉 모*
(Seok-jin Lee*, Jun-seok Lim**, Koeng-mo Sung*)

*서울대학교 전기컴퓨터공학부, **세종대학교 전자공학과
(접수일자: 2009년 11월 18일; 채택일자: 2009년 12월 2일)

RLS (Recursive Least Squares) 알고리즘은 적응 알고리즘의 대표적인 알고리즘이다. 하지만, 기본적인 RLS 알고리즘은 빠르게 움직이는 신호와 같은 비정상 (non-stationary) 신호환경에서는 좋은 성능을 가질 수 없다는 단점이 있다. 이를 해결하기 위하여 가변 망각 인자를 가지는 RLS 알고리즘이 등장하였으나, 기존의 가변 망각 인자 RLS 알고리즘은 연산량이 너무 많다는 단점이 있다. 본 논문에서는 이를 해결하기 위하여, 상대적으로 적은 연산량으로 AFF-RLS 알고리즘과 비슷한 성능을 내는 RLS 알고리즘을 제안한다.
핵심용어: 가변 망각 인자, RLS, 정규화 기법, 적응 알고리즘
투고분야: 음향통신가술 분야 (6)

The RLS (Recursive Least Squares) method is a broadly used adaptive algorithm for signal processing in electronic engineering. The RLS algorithm shows a good performance and a fast adaptation within a stationary environment, but it shows a poor performance within a non-stationary environment because the method has a fixed forgetting factor. In order to enhance 'tracking' performances, RLS methods with an adaptive forgetting factor had been developed. This method shows a good tracking performance, however, it suffers from heavy computational loads. Therefore, we propose a modified AFF-RLS which has relatively low complexity in this paper.
Keywords: Variable Forgetting Factor, RLS, Normalization Method, Adaptive Algorithm.
ASK subject classification: Acoustic Communication (6)

I. 서 론

RLS (Recursive Least Squares) 알고리즘은 전자공학, 특히 신호처리 분야에서 널리 쓰이는 적응 알고리즘 중 하나이다. 또한 RLS 알고리즘은 정상 (Stationary) 신호 환경에서 좋은 성능과 빠른 적응 속도를 보여주는 알고리즘이지만, 고정된 망각 인자 (혹은 메모리)를 가지고 있기 때문에 비정상 (Non-stationary) 신호 환경에서는 좋지 않은 성능을 보여준다.

비정상 환경에서 추적 (Tracking) 성능을 높이가 위해서는, RLS 알고리즘의 망각 인자를 가변적으로 조절할 필요가 있다. 이러한 방법의 기본적인 알고리즘이 [1]에 소개된 바 있다. 이는 기존의 RLS 알고리즘에 비해 좋은 성능을 보여주지만, 연산량이 크게 증가하여 복잡도가 너무 크다는 단점을 가지고 있다.

본 논문에서는, 위의 문제를 개선한 MAFF-RLS 알고리즘을 제안한다. 이를 위하여, NLMS (Normalized Least Mean Squares) 알고리즘에서 사용되었던 정규화 (Normalization) 기법을 살펴보고, LMS (Least Mean Squares) 알고리즘의 필터 계수 갱신 식과 AFF-RLS 알고리즘의 망각 인자 갱신 식의 유사성 (Analogy)을 이끌어 내어, 이를 이용하여 정규화된 MAFF-RLS 알고리즘을 도출해 낸다.

책임저자: 이 석 진 (sjlee@acoustics.snu.ac.kr)
서울시 관악구 신림9동 서울대학교 132동 302호
(전화: 02-880-8427; 팩스: 02-880-8207)

Ⅱ. Conventional AFF-RLS method

본 장에서는 [1]에 소개된 바 있는 AFF-RLS 알고리즘에 대하여 언급한다.

2.1. AFF-RLS method

RLS 알고리즘은 LS (Least Squares) 알고리즘을 재귀적으로 풀어낸 알고리즘이다. RLS 알고리즘에서 최소화하고자 하는 비용 함수 (Cost Function)는 다음과 같다.

$$\varepsilon(n)=\sum_{i=1}^{n}\beta(n,i)|e(i)|^2 \tag{1}$$

여기서 가중치 (Weighting Factor) $\beta(n,i)$는 오래된 값들의 비중을 줄이기 위해 사용되며, 일반적으로 다음과 같은 '지수적 가중치'를 사용한다.

$$\beta(n,i)=\lambda^{n-i} \tag{2}$$

여기서 가중치 λ는 '망각 인자 (Forgetting Factor)로 불리며, 1에 매우 가까운 (그리고 1보다 작은) 값을 사용한다.

위와 같이 정의된 비용 함수를 Least Square로 풀고, 이를 재귀적으로 나타내면 다음과 같은 RLS 알고리즘이 유도된다 [1].

$$\underline{k}(n)=\frac{P(n-1)\underline{u}(n)}{\lambda+\underline{u}^{H}(n)P(n-1)\underline{u}(n)} \tag{3}$$

$$\xi(n)=d(n)-\underline{\hat{w}}^{H}(n-1)\underline{u}(n) \tag{4}$$

$$\underline{\hat{w}}(n)=\underline{\hat{w}}(n-1)+\underline{k}(n)\xi^{*}(n) \tag{5}$$

$$P(n)=\lambda^{-1}P(n-1)-\lambda^{-1}k(n)u^{H}(n)P(n-1) \tag{6}$$

위의 RLS 식에서 $\underline{k}(n)$은 이득 벡터 (Gain vector)이며, $\xi(n)$는 선험적 추정 오류 (A priori estimation error), $\underline{\hat{w}}(n)$는 필터 계수, $P(n)$은 입력 신호의 자기상관 행렬 (Autocorrelation matrix)의 역행렬 (Inverse Matrix)를 나타낸다.

AFF-RLS 알고리즘은 RLS 알고리즘의 망각 인자를 가변적으로 조절하는 알고리즘이며, 망각 인자는 다음의 비용 함수를 최소화하는 값으로 조절된다.

$$J'(n)=\frac{1}{2}E\left[|\xi(n)|^2\right] \tag{7}$$

위 비용 함수의 망각 인자 λ에 따른 기울기를 구하면 아래와 같다.

$$\nabla_{\lambda}(n)=\frac{\partial J'(n)}{\partial\lambda}=\frac{1}{2}E\left[\frac{\partial\xi(n)}{\partial\lambda}\xi^{*}(n)+\frac{\partial\xi^{*}(n)}{\partial\lambda}\xi(n)\right]$$

$$=-\frac{1}{2}E\left[\underline{\psi}^{H}(n-1)u(n)\xi^{*}(n)+u^{H}(n)\underline{\psi}(n-1)\xi(n)\right] \tag{8}$$

여기서,

$$\underline{\psi}(n)=\frac{\partial\underline{\hat{w}}(n)}{\partial\lambda} \tag{9}$$

$$S(n)=\frac{\partial P(n)}{\partial\lambda} \tag{10}$$

위의 $\psi(n)$, $S(n)$는 아래와 같이 재귀적으로 계산될 수 있다 [1].

$$\psi(n)=\left[I-k(n)u^{H}(n)\right]\underline{\psi}(n-1)+S(n)\underline{u}(n)\xi^{*}(n) \tag{11}$$

$$S(n)=\lambda^{-1}\left[I-k(n)u^{H}(n)\right]S(n-1)\left[I-u(n)k^{H}(n)\right]$$
$$+\lambda^{-1}k(n)k^{H}(n)-\lambda^{-1}P(n) \tag{12}$$

여기서 다음과 같은 'Method of steepest descent'을 이용하면 망각 인자의 갱신식이 완성된다.

$$\lambda(n)=\lambda(n-1)-\alpha\hat{\nabla}_{\lambda}(n)$$
$$=\lambda(n-1)+\alpha\,\mathrm{Re}\left[\underline{\psi}^{H}(n-1)\underline{u}(n)\xi^{*}(n)\right] \tag{13}$$

여기서 α는 'learning-rate'값으로, Step-size parameter와 같은 의미를 갖는다.

식 (3)~(6)의 RLS 알고리즘에서 망각 인자 λ값을 식 (11)~(13)으로 갱신해주면 AFF-RLS 알고리즘이 완성된다.

Ⅲ. Proposed Algorithm

본 장에서는 NLMS 알고리즘에 사용된 정규화 기법을 살펴보고, NLMS와 AFF-RLS의 유사성을 살펴본 뒤, 이를 AFF-RLS에 적용하여 NAFF-RLS (Normalized AFF-

RLS) 알고리즘을 도출한다. 그리고 이를 이용하여 적은 연산량을 가지는 MNAFF-RLS (Modified NAFF-RLS) 알고리즘을 도출한다.

3.1. NLMS 알고리즘

본 논문의 목적을 위하여 먼저 NLMS 알고리즘에 대해 간단히 살펴본다. LMS 알고리즘은 다음과 같은 'Method of steepest descent' 알고리즘에 기반을 두고 있다 [1].

$$\underline{\hat{w}}(n+1) = \underline{\hat{w}}(n) - \frac{\mu}{2}\nabla J(n) \tag{14}$$

한편, LMS에서 최소화하고자 하는 비용 함수는 다음과 같다.

$$J(n) = \left|e^2(n)\right| \tag{15}$$

$$e(n) = d(n) - \underline{\hat{w}}^H(n)\underline{u}(n) \tag{16}$$

따라서, 비용 함수 $J(n)$의 필터 계수 $\hat{w}$에 대한 기울기는 다음과 같다.

$$\nabla J(n) = -2\underline{p} + 2R\underline{w}(n) \tag{17}$$

여기서 $R = E\left[\underline{u}\underline{u}^H\right]$, $p = E\left[\underline{u}d^*\right]$이다.

LMS 알고리즘에서는 위와 같은 자기상관행렬 (Auto-correlation matrix)와 상호상관벡터 (Cross-correlation vector)를 순간 값 (Instantaneous value)을 이용하여 나타낸다. 즉,

$$\hat{R}(n) = \underline{u}(n)\underline{u}^H(n) \tag{18}$$

$$\underline{\hat{p}}(n) = \underline{u}(n)d^*(n) \tag{19}$$

위의 추정 값을 가지고 'Method of steepest descent' 알고리즘 (식 (14))을 이용하면, 다음과 같이 LMS 알고리즘의 필터 계수 갱신식이 유도된다.

$$\begin{aligned}\underline{\hat{w}}(n+1) &= \underline{\hat{w}}(n) - \frac{\mu}{2}\nabla J(n) \\ &= \underline{\hat{w}}(n) + \mu\underline{u}(n)e^*(n)\end{aligned} \tag{20}$$

식 (20)에서 보듯이, 필터 계수의 조절은 Step-size parameter μ,Tap-input vector $\underline{u}(n)$, 그리고 estimation error $e^*(n)$의 곱으로 이루어져 있다. 따라서, $\underline{u}(n)$값이 큰 경우 'gradient noise amplification' 문제에 의해 성능이 저하될 수 있다 [1]. 이를 해결하기 위해 NLMS (Normalized LMS) 알고리즘이 개발되었다. [1]에 따르면, NLMS의 갱신 식은 다음과 같다.

$$\underline{\hat{w}}(n+1) = \underline{\hat{w}}(n) + \frac{\tilde{\mu}}{\left\|\underline{u}(n)\right\|^2}\underline{u}(n)e^*(n) \tag{21}$$

위의 식에서 보듯이, NLMS 알고리즘은 LMS 알고리즘의 Step-size parameter를 Tap-input vector의 Squared norm으로 정규화한 것과 같다. 또한, Tap-input vector의 크기가 매우 작을 경우, 분모의 값이 매우 작아지는 문제가 생길 수 있으므로, 이를 해결하기 위해 다음과 같은 Regularization factor를 추가하기도 한다.

$$\underline{\hat{w}}(n+1) = \underline{\hat{w}}(n) + \frac{\tilde{\mu}}{\delta + \left\|\underline{u}(n)\right\|^2}\underline{u}(n)e^*(n) \tag{22}$$

3.2. MNAFF-RLS 알고리즘

NAFF-RLS 알고리즘을 도출하기 위하여, 먼저 LMS의 필터 계수 갱신 식과 RLS 망각 인자 갱신 식의 유사성을 끌어낼 필요가 있다. 이러한 유사성의 정당성을 확보할 수 있는 이유는, 두 방법이 모두 같은 기법 ('Method of steepest descent')을 사용하였다는 것과, 그렇기 때문에 같은 문제 ('Gradient noise amplification')를 가 지고 있다는 것이다.

식 (13)과 식 (20)의 갱신 식을 살펴보면, 두 식이 공통적으로 다음과 같은 구조를 가지고 있음을 알 수 있다.

$$z_{new} = z_{old} + c_{step}s_{scaling}\varepsilon \tag{23}$$

여기서 z는 갱신할 값이며, c_{step}은 Step-size parameter라 불리는 양의 상수 값, 그리고 ε는 추정 오류이다. 따라서, 두 갱신식이 표 1과 같은 유사성을 가짐을 알 수 있다.

위와 같은 유사성을 가지고, NLMS에서 사용되었던 Normalization 기법을 AFF-RLS의 망각 인자 갱신 식에

표 1. RLS의 망각 인자 갱신식과 LMS의 필터 계수 갱신식의 유사성.

Table 1. Analogy between Forgetting Factor Update Equation of the RLS and Filter Weight Update Equation of the LMS.

	Weight Vector Update Equation of LMS	Forgetting Factor Update Equation of AFF-RLS
$z(n)$	$\underline{\hat{w}}(n)$	$\lambda(n)$
c_{step}	μ	α
ε	$e(n)$	$\xi(n)$
$S_{scaling}$	$\underline{u}(n)$	$\underline{\psi}^H(n-1)\underline{u}(n)$

사용하면 다음과 같은 새로운 망각 인자 갱신 식을 얻을 수 있다.

$$\lambda(n)=\lambda(n-1)+\frac{\alpha}{\left|\underline{\psi}^H(n-1)\underline{u}(n)\right|^2}\mathrm{Re}\left[\underline{\psi}^H(n-1)\underline{u}(n)\xi^*(n)\right]\Big|_{\lambda_-}^{\lambda_+} \quad (24)$$

AFF-RLS 알고리즘에서 식 (13)의 갱신 식 대신 식 (24)를 이용하여 망각 인자를 갱신하면 NAFF-RLS 알고리즘이 완성된다. 일반적인 AFF-RLS 알고리즘의 경우 이러한 정규화 과정이 큰 도움을 주지 못하지만, 연산량을 줄인 MAFF-RLS의 경우 기울기 에러의 값이 비교적 크기 때문에 위의 기법이 큰 도움이 된다.

한편, AFF-RLS 알고리즘의 입력이 i.i.d. (independently identically distributed)이고, 알고리즘의 추정 오류가 통계적으로 독립일 때, 망각 인자가 1에 가깝고 n이 충분히 크다면 다음과 같은 근사식이 성립한다 [2][3].

$$I-\underline{k}(n)\underline{u}^H(n)=c(n)I \quad (25)$$

여기서,

$$c(n)=1-\frac{\underline{k}^H(n)\underline{u}(n)}{L} \quad (26)$$

위의 식 (25)와 같은 근사식을 NAFF-RLS 알고리즘에 적용하면 $\hat{S}(n)$을 다음과 같이 갱신할 수 있다.

$$\hat{S}_n(n)=\left[\hat{\lambda}_n(n-1)\right]^{-1}\left|c(n)\right|^2\hat{S}_n(n-1)-\left[\hat{\lambda}_n(n-1)\right]^{-1}c(n)P(n) \quad (27)$$

식 (27)의 양변에 Tap-input vector $\underline{u}(n)$을 곱하면, 다음과 같은 식이 유도된다.

$$\hat{S}_n(n)=\left[\hat{\lambda}_n(n-1)\right]^{-1}\left|c(n)\right|^2\hat{S}_n(n-1)\underline{u}(n)-\left[\hat{\lambda}_n(n-1)\right]^{-1}c(n)P(n)\underline{u}(n) \quad (28)$$

위의 식 (28)에서 sampling interval이 충분히작아서 다음과 같은 근사가 가능하다고 가정하자.

$$\hat{S}_n(n-1)\underline{u}(n)\cong\hat{S}_n(n-1)\underline{u}(n-1) \quad (29)$$

그리고 행렬 $\hat{S}_n(n)$과 입력 벡터 $\underline{u}(n)$의 곱을 다음과 같이 $\underline{\hat{q}}(n)$이라고 정의하자.

$$\underline{\hat{q}}(n)=\hat{S}_n(n)\underline{u}(n) \quad (30)$$

위와 같이 정의된 $\underline{\hat{q}}(n)$은 L×1의 벡터가 된다. 또한 RLS 알고리즘에서 이득 벡터 $\underline{k}(n)$는 다음과 같은 관계를 갖는다 [1].

$$\underline{k}(n)=P(n)\underline{u}(n) \quad (31)$$

따라서, 식 (28), (29), (31)에 의해 $\underline{\hat{q}}(n)$은 다음과 같은 식을 통해 갱신될 수 있다.

$$\underline{\hat{q}}(n)=\left[\lambda(n-1)\right]^{-1}\left[\left|c(n)\right|^2\underline{\hat{q}}(n-1)-c(n)\underline{k}(n)\right] \quad (32)$$

위의 과정을 통해 최종적으로 MNAFF-RLS 알고리즘의 망각인자 갱신식이 다음과 같이 유도된다.

$$\underline{\hat{q}}(n)=\left[\lambda(n-1)\right]^{-1}\left[\left|c(n)\right|^2\underline{\hat{q}}(n-1)-c(n)\underline{k}(n)\right] \quad (33)$$

$$\underline{\hat{\psi}}_n(n)=c(n)\underline{\hat{\psi}}_n(n-1)+\underline{\hat{q}}(n)\xi^*(n) \quad (34)$$

$$\hat{\lambda}_n(n)=\hat{\lambda}_n(n-1)+\frac{\alpha}{\left|\underline{\hat{\psi}}_n^H(n-1)\underline{u}(n)\right|^2}\mathrm{Re}\left[\underline{\hat{\psi}}_n^H(n-1)\underline{u}(n)\xi^*(n)\right]\Big|_{\lambda_-}^{\lambda_+} \quad (35)$$

식 (3)~(6)의 RLS 알고리즘에 식 (33)~(35)의 망각

인자 갱신 식을 이용하면 MNAFF-RLS 알고리즘이 완성된다.

아래 표 2에 각 알고리즘의 연산량을 비교한 표가 나타나있다. 이를 통해 MNAFF-RLS 알고리즘은 MAFF-RLS 알고리즘, AFF-RLS 알고리즘에 비해 매우 작은 연산량을 가지고 있음을 알 수 있다.

Ⅳ. 모의 실험 및 결과

4.1. 실험 환경

각 알고리즘의 성능을 비교하기 위하여 그림 1과 같이 간섭 신호 제거를 위한 GSC (Generalized Sidelobe Canceller)에 알고리즘을 적용시켜 그 성능을 비교하였다. 모의 실험 시나리오는 그림 2와 같으며, 실험 설계는 [3]과 같은 설계를 사용하였다.

본 모의 실험은 하나의 GSC (beamformer), 하나의 목적 신호 (Desired signal), 하나의 움직이는 간섭 신호 (Moving interference signal)로 구성되었다. GSC의 센서 개수는 8개이며, 배열 센서에 입력되는 목적 신호의 SNR은 20 dB, 간섭 신호의 SNR은 26 dB로 설정하였고 각 음원은 far-field에 존재한다고 가정하였다. 각 적응 알고리즘들은 Beamformer의 계수를 조절하여 간섭 신호를 제거하는 데에 쓰였다.

표 2. 각 알고리즘의 연산량 비교
Table 2. Computational cost comparison of each method.

	Complex Multiplication	Complexity for Forgetting Factor update
RLS	$2.5L^2+3L$	0
AFF-RLS	$9L^2+7L$	$6.5L^2+4L$
MAFF-RLS	$4L^2+5L$	$1.5L^2+2L$
MNAFF-RLS	$2.5L^2+7L$	$4L$

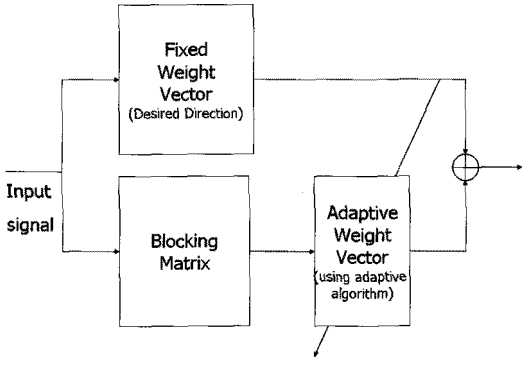

그림 1. 성능 평가를 위한 GSC의 다이어그램
Fig. 1. Block diagram of GSC to evaluate each method.

RLS 알고리즘의 망각 인자 값은 0.99를 사용하였고, AFF-RLS, MAFF-RLS, MNAFF-RLS 알고리즘의 'learning-rate'는 모두 0.02를 사용하였다.

성능을 나타내는 성능 지수로는 SINR (Signal-to-Interference-and-Noise-Ratio)를 사용하였다. SINR은 SNR (Signal-to-Noise-Ratio)와 마찬가지로 그 값이 높을수록 좋은 성능을 가짐을 나타낸다.

4.2. 실험 결과

그림 3에 모의 실험의 SINR 결과가 나타나있다. 그림 3의 결과 그래프를 살펴보면, 모든 SINR 그래프는 가운데에서 Valley를 가지고 있으며, 이는 목적 신호와 간섭 신호의 방위가 같아짐으로써 일어나는 성능 저하를 나타낸다. 따라서, 본 성능 테스트의 핵심은 성능 저하에서 얼마나 빠르게 회복하며, 회복한 후의 SINR 값이 얼마나 높은 값을 가지는가로 나타낼 수 있다. 그래프를 살펴보면, MNAFF-RLS 알고리즘은 MAFF-RLS 알고리즘보다 좋은 성능을 보여주며, AFF-RLS 알고리즘과 거의 유사한 성능을 보여주는 것을 알 수 있다.

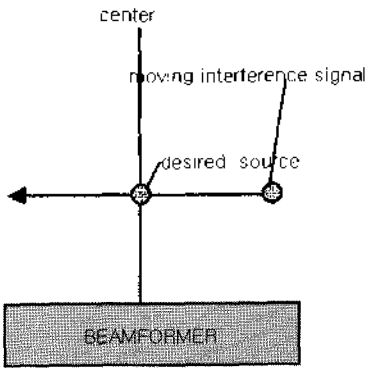

그림 2. 모의 실험 시나리오
Fig. 2. Simulation Scenario.

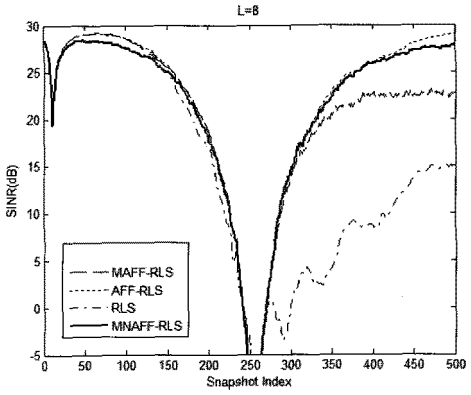

그림 3. 모의 실험의 SINR 결과 추이
Fig. 3. SINR Result.

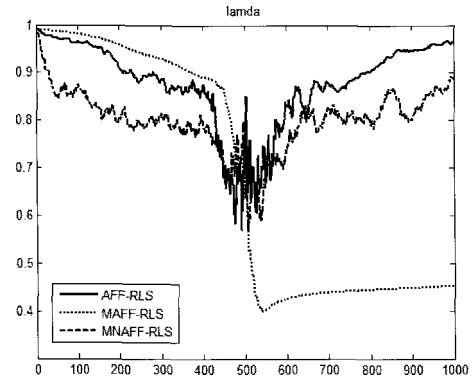

그림 4. 모의 실험의 망각 인자 변화 추이
Fig. 4. Forgetting Factor Result.

그림 4에는 AFF-RLS, MAFF-RLS, MNAFF-RLS 알고리즘의 망각인자 변화 추이가 나타나있다. 그림 4의 그래프를 살펴보면 MNAFF-RLS 알고리즘은 AFF-RLS와 유사한 망각 인자 변화를 나타냄을 확인할 수 있다.

V. 결 론

본 논문에서는, 비교적 적은 연산량으로 망각인자를 조절하는 MNAFF-RLS 알고리즘을 제안하였다. 이를 위하여, AFF-RLS 알고리즘에서 망각 인자를 조절하는 방법에 대하여 고찰하고, 이와 같은 원리를 사용한 LMS 알고리즘을 살펴보았다. 또한, LMS 알고리즘에서 NLMS 알고리즘을 도출하는 데 사용되었던 Normalization 기법을 살펴보고, LMS와 AFF-RLS 알고리즘의 유사성을 이끌어낸 후 이를 이용하여 AFF-RLS 알고리즘에 Normalization 기법을 적용하여 NAFF-RLS 알고리즘을 도출하였다. 또한, NAFF-RLS 알고리즘에 Approximation 기법을 적용하여 연산량을 줄인 MNAFF-RLS 알고리즘을 도출하였다.

최종적으로 도출된 MNAFF-RLS 알고리즘을 GSC (Generalized Sidelobe Canceller)에 적용하여 각 알고리즘의 성능을 평가하였다. 모의 실험 결과, 제안하는 MNAFF-RLS 알고리즘이 MAFF-RLS 알고리즘에 비해 좋은 성능을 가지며, AFF-RLS 알고리즘과 유사한 성능을 가짐을 확인할 수 있었다. 이를 통하여 AFF-RLS 알고리즘보다 적은 연산량을 가지고 유사한 성능을 낼 수 있는 MNAFF-RLS의 제안 및 검증이 완료되었다.

참 고 문 헌

1. Haykin, *Adaptive Filter Theory* (Fourth edition), Prentice Hall, New Jersey, 663, 2002.
2. 송성욱, "Self-Tunning Adaptive Algorithm and Applications", 박사학위논문, 서울대학교, 2003.
3. 이석진, 임준석, 성굉모, '비정상 간섭잡음 제거를 위한 광대역 MAFF-RLS 마이크로폰 GSC', *한국음향학회지*, 28권, 6호, 520-525쪽, 2009.

저자 약력

•이 석 진 (Seok-Jin Lee)
2006년 8월 : 서울대학교 공과대학 전기·컴퓨터공학부 (공학사)
2008년 8월 : 서울대학교 공과대학 전기·컴퓨터공학부 (공학석사)
2008년 9월 ~ 현재 : 서울대학교 공과대학 전기·컴퓨터공학부 박사과정

•임 준 석 (Jun-Seok Lim)
1986년 2월 : 서울대학교 전자공학과 (공학사)
1988년 2월 : 서울대학교 전자공학과 (공학석사)
1988년 3월 ~ 1993년 4월 : 국방과학연구소
1996년 8월 : 서울대학교 전자공학과 (공학박사)
1996년 7월 ~ 1998년 2월 : LG 종합연구소
1998년 3월 ~ 현재 : 세종대학교 전자공학과 교수
※ 관심분야 : 레이더/소나 신호처리, 통계 신호처리, 수중음향학

•성 굉 모 (Koeng-Mo Sung)
1965년~1971년 : 서울대학교 전자공학과
1971년 ~ 1973년 : 독일 아헨공대 Vordipolm
1973년 ~ 1977년 : 독일 아헨공대 전자통신공학 Dipl.-Ing.
1977년 ~ 1982년 : 독일 아헨공대 음향공학 Dr.-Ing.
1977년 ~ 1983년 : 독일 아헨공대 음향공학연구소 연구원
1983년 ~ 현재 : 서울대학교 공과대학 전기·컴퓨터공학부 교수.