논문 99-24-9B-22 한국통신학회논문지 '99-9 Vol.24 No.9B

Gram-Schmidt 직교화를 이용한 affine projection 알고리즘의 근사적 구현

정회원 김 은 숙*, 정 양 원*, 박 선 준*, 박 영 철**, 윤 대 희*

An approximated implementation of affine projection algorithm using Gram-Schmidt orthogonalizationn

Eun-Sook Kim*, Yang-Won Jung*, Seon-Joon Park*, Young-Cheol Park**, Dae-Hee Youn*
Regular Members

요 약

Affine projection 알고리즘은 RLS보다 적은 계산량으로 LMS보다 우수한 수렴성능을 나타낸다. 그러나 affine projection 알고리즘은 역행렬 연산을 필요로 하기 때문에 여전히 LMS에 비해 과중한 계산을 필요로 한다. 본 논문에서는 affine projection 알고리즘을 분석하여 이 알고리즘이 Gram-Schmidt 구조로 해석될 수 있음을 보이고 이를 이용하여 NLMS와 비슷한 계산량으로 affine projection 알고리즘을 근사적으로 구현할 수 있는 새로운 알고리즘을 제안하였다. 제안한 방법은 NLMS와 비슷한 계산량을 가지면서 기존의 affine projection 알고리즘과 비슷한 수렴성능을 나타내었다.

ABSTRACT

The affine projection algorithm has known to require less computational complexity than RLS but have much faster convergence than NLMS for speech-like input signals. But the affine projection algorithm is still much more computationally demanding than the LMS algorithm because it requires the matrix inversion. In this paper, we show that the affine projection algorithm can be realized with the Gram-Schmidt orthogonalization of input vectors. Using the derived relation, we propose an approximate but much more efficient implementation of the affine projection algorithm. Simulation results show that the proposed algorithm has the convergence speed that is comparable to the affine projection algorithm with only a slight extra calculation complexity beyond that of NLMS.

I. 서 론

적응 필터링 기법은 반향제거, 소음제어, 음장제어 등과 같이 신호 처리의 여러 응용분야에서 광범위하게 사용되고 있다. 적응필터를 구현하기 위한 방법으로 크게 LMS(Least Mean Square) 알고리즘과 RLS(Recursive Least Square) 알고리즘으로 나눌 수 있다. LMS와 NLMS(Normalized LMS) 알고리즘은 매우 간단하여 구현이 쉽기 때문에 광범위하게 사용되고 있으나 음성신호와 같이 상관관계가

* 연세대학교 전기 컴퓨터 공학과 음향, 음성 및 신호처리 연구실 (eskim@cyclon.yonsei.ac.kr),
** 연세대학교 신호처리 연구센터
논문번호 : 98504-1121, 접수일자 : 1998년 11월 21일
※ 본 연구는 과학재단 특정 기초 연구 (95-0100-03-05-3) 지원으로 수행되었습니다.

큰 입력신호에 대해서는 수렴속도가 현저히 느려지게 되는 단점이 있다. 반면에, RLS 알고리즘은 이러한 유색신호(colored signal)에 대해 매우 좋은 수렴성능을 보이지만 계산이 복잡하여 구현이 어렵다는 단점이 있다.

최근 들어 NLMS와 RLS의 중간성능을 가지는 affine projection 알고리즘에 대한 관심이 증가하고 있다[1-4]. 입력신호가 음성신호와 같이 AR process로 모델링 될 수 있는 신호인 경우에 affine projection 알고리즘은 RLS에 비해 적은 계산량으로 NLMS보다 훨씬 빠른 수렴성능을 보인다. 그러나 affine projection 알고리즘은 역행렬 연산을 필요로 하기 때문에 여전히 LMS에 비해 과중한 계산을 필요로 한다. 따라서 계산량을 줄이기 위한 고속 알고리즘에 대한 연구가 꾸준히 이루어지고 있는 실정이다[5-7].

본 논문에서는 affine projection 알고리즘의 수렴성능을 그대로 유지하면서 계산량을 NLMS 알고리즘 수준으로 줄일 수 있는 새로운 적응 알고리즘을 제안하였다. Affine projection 알고리즘의 계수 갱신식은 행렬간의 곱 형태로 표현되며 특히 역행렬 연산을 필요로 한다. 본 논문에서는 행렬의 형태로 표현되는 affine projection 알고리즘의 계수 갱신식을 단순한 벡터의 형태로 유도하였는데, 계수 갱신에 필요한 벡터를 찾기 위하여 Gram-Schmidt 직교화가 사용되었다. 즉, 현재 신호벡터를 주 입력신호(desired signal)로 하고 과거 $(P-1)$ 개의 신호벡터를 참조 입력신호(reference signal)로 하는 벡터 단위의 $(P-1)$ 차 Gram-Schmidt 직교화를 이용하여 최종 오차벡터를 구할 수 있는데 이렇게 구한 오차벡터만을 이용하여 F 차 affine projection 알고리즘을 정확히 표현할 수 있음을 보였다. 다음으로 입력 신호가 stationary한 상황이라면, 최종 오차벡터를 구하기 위하여 F 개의 입력신호 벡터들을 이용한 벡터 단위의 Gram-Schmidt 직교화를 하는 대신 F 개의 입력 샘플에 대한 Gram -Schmidt 직교화를 수행하는 것과 동일한 결과를 얻음을 보였고, 그 결과 stationary 상황에서 F 차 affine projection 알고리즘이 P 개의 입력 샘플들간의 Gram-Schmidt 예측기와 L 개의 TDL(Tapped Delay Line) 구조로 간단하게 구현될 수 있음을 보였다.

본 논문에서는 이와 같이 입력 신호 $x(n)$으로부터 $(P-1)$ 차 Gram-Schmidt 변환을 통하여 최종 오차를 얻어낸 다음 이를 TDL 구조를 이용하여 오차벡터 $\boldsymbol{z}(n)$을 구성하고 이를 계수적응에 사용함으로써 affine projection 알고리즘을 근사적으로 구현하였다. 제안된 방법에서는 LMS에 비해 $(P-1)$ 차 Gram-Schmidt 직교화를 위해 약 $(P-1)^2+6(P-1)$ 번의 계산만이 추가로 필요하게 된다. 모의실험 결과 제안된 방법은 affine projection 알고리즘과 같은 수렴성능을 보였다.

본 논문은 2장에서는 F 차 affine projection 알고리즘이 $(P-1)$ 차 Gram-Schmidt 구조로 해석되는 것을 보이고 3장에서는 Gram-Schmidt 직교화를 이용한 고속 affine projection 알고리즘을 제안한다. 4장에서 제안된 방법의 실효성을 모의실험을 통하여 보이고 5장에서 결론을 맺고자 한다.

Ⅱ. Affine projection 알고리즘의 해석

1. Affine projection 알고리즘[1-4]

Affine projection 알고리즘은 *a posteriori* 오차를 제거하는 통계적 gradient 형태의 알고리즘으로 나타난다. 적응필터의 길이를 L 이라 할 때, 입력 데이터 행렬 $\boldsymbol{X}(n)$과 주 입력 벡터(desired signal vector) $\boldsymbol{y}(n)$을 각각 다음과 같이 정의하자.

$$\boldsymbol{X}(n)=[\ \boldsymbol{x}_L(n)\ \boldsymbol{x}_L(n-1)\ \cdots\ \boldsymbol{x}_L(n-P+1)],$$
$$\boldsymbol{x}_L(n)=[x(n)\ x(n-1)\ \cdots\ x(n-L+1)]^T \tag{1}$$

$$\boldsymbol{y}(n)=[y(n)\ y(n-1)\ \cdots\ y(n-P+1)]^T \tag{2}$$

Projection 차수를 F 라고 할 때 P 차의 *a priori* 오차 벡터 $\boldsymbol{e}(n)$ 과 F 차의 *a posteriori* 오차 벡터 $\boldsymbol{\varepsilon}(n)$ 은 각각 다음과 같이 표현될 수 있다.

$$\boldsymbol{e}(n)=\boldsymbol{y}(n)-\boldsymbol{X}^T(n)\ \boldsymbol{h}(n) \tag{3}$$

$$\boldsymbol{\varepsilon}(n)=\boldsymbol{y}(n)-\boldsymbol{X}^T(n)\ \boldsymbol{h}(n+1) \tag{4}$$

여기서 $\boldsymbol{h}(n)$은 L 개의 계수값들로 구성된 적응 필터 계수 벡터이고, $\boldsymbol{e}(n)$ 과 $\boldsymbol{\varepsilon}(n)$ 은 각각 다음과 같이 정의된다.

$$\boldsymbol{e}(n)=[\ e_0(n)\ \ e_1(n)\ \ \cdots\ e_{P-1}(n)\ ]^T$$

$$\boldsymbol{\varepsilon}(n) = [\ \varepsilon_0(n)\ \ \varepsilon_1(n)\ \ \cdots\ \varepsilon_{P-1}(n)\ ]^T$$

Affine projection 알고리즘은 P 개의 *a posteriori* 오차 $\boldsymbol{\varepsilon}(n)$을 영으로 만드는 계수 벡터를 찾아가는 알고리즘이므로 식 (3)의 $\boldsymbol{y}(n)$을 식 (4)에 대입한 후 $\boldsymbol{\varepsilon}(n)=0$ 으로 놓으면 다음 식을 얻을 수 있다.

$$\boldsymbol{X}(n)^T \Delta \boldsymbol{h}(n) = \boldsymbol{e}(n) \tag{5}$$

여기서 $\Delta \boldsymbol{h}(n) = \boldsymbol{h}(n+1) - \boldsymbol{h}(n)$이다.

A posteriori 오차를 영으로 하는 적응 필터의 계수는 식 (5)를 풀면 구할 수 있다. 그런데 식 (5)는 L 개의 미지수와 P 개의 방정식인 형태이고 $P<L$ 이므로 underdetermined 시스템이 된다[8]. 따라서 식 (5)에서 구하는 $\Delta \boldsymbol{h}(n)$의 해는 다음과 같은 minimum norm solution 으로 주어진다.

$$\Delta \boldsymbol{h}(n) = \boldsymbol{X}(n)[\ \boldsymbol{X}(n)^T \boldsymbol{X}(n)]^{-1}\ \boldsymbol{e}(n) \tag{6}$$

식 (6)을 이용한 affine Projection 알고리즘의 계수 갱신식은 다음과 같다

$$\boldsymbol{h}(n+1) = \boldsymbol{h}(n) + \boldsymbol{X}(n)[\ \boldsymbol{X}(n)^T \boldsymbol{X}(n)]^{-1}\ \boldsymbol{e}(n) \tag{7}$$

배경 잡음등으로 인한 예측 오차의 갑작스런 증가는 알고리즘의 성능을 매우 저하시킨다[13]. 따라서 이를 방지하고 알고리즘을 보다 효율적으로 만들기 위하여 수렴상수 혹은 오차조절상수(error regulation factor) μ를 도입하여 적응 알고리즘을 표현할 수도 있다.

Affine Projection알고리즘은 projection 차수에 따라 다른 적응 알고리즘으로 표현되어 질 수 있다. projection 차수 P 가 1이면 식 (7)은 NLMS (Normalized LMS)알고리즘의 계수 갱신식과 유사한 형태가 된다. 일반적으로 NLMS 알고리즘은 입력 신호로 만들어지는 1차원 평면에 대한 projection 알고리즘이라고 말할 수 있다. 따라서 affine projection 알고리즘은 NLMS 알고리즘의 일반화한 형태라고 볼 수 있다. Projection 차수가 커질수록 많은 양의 데이터를 가지고 계수를 갱신하므로 수렴속도가 증가하게 된다. 위 식 (7)에서 $P>L$ 이 되면 식 (7)은 overdetermined 시스템이 되어 Least-Square solution에 의해 해를 구할 수 있다.

$P=L$ 인 경우 affine Projection알고리즘은 RLS 알고리즘의 계수 갱신식과 동일한 형태를 갖는다. 만약 q 차 AR 모델링에 의한 신호가 입력으로 사용될 경우 projection 차수는 AR 모델링 차수 q (q << L)정도만 사용하여도 빠른 수렴 속도를 얻을 수 있다. 따라서 음성신호등과 같이 AR 모델링이 가능한 입력신호를 사용하는 경우 많지 않은 계산량으로도 빠른 수렴속도를 얻을 수 있다.

2. Affine projection 알고리즘의 분석

P 차 affine projection 알고리즘은 P 개의 *a posteriori* 오차를 영으로 만들도록 계수를 조정해 가는 알고리즘이다. 즉, 식 (4)로부터 $\boldsymbol{h}(n+1)$ 은 다음과 같은 관계를 만족시켜야 됨을 알 수 있다.

$$\boldsymbol{x}_L(n-i)^T \boldsymbol{h}(n+1) = y(n-i), \quad i=0,1,2,\cdots P-1 \tag{8}$$

식(3)을 오차벡터 $\boldsymbol{e}(n)$ 의 각 element들로 다시 표현하면 다음과 같다.

$$e_i(n) = y(n-i) - \boldsymbol{x}_L(n-i)^T \boldsymbol{h}(n), \quad 0 \le i \le P-1 \tag{9}$$

그런데 윗 식에서 $e_1(n), \cdots, e_{P-1}(n)$ 은 결국 시간 $n-1$ 에서 구한 *a posteriori* 오차 $\varepsilon_0(n-1), \cdots, \varepsilon_{P-2}(n-1)$ 와 같다. 따라서 오차벡터 $\boldsymbol{e}(n)$은 다음과 같이 첫 번째 element 만 값을 가지고 나머지는 모두 영인 벡터가 된다. 즉,

$$\boldsymbol{e}(n) = [\ e(n)\ \ 0\ \ \cdots\ \ 0\ ]^T \tag{10}$$

임을 알 수 있다.

식 (1)에 주어진 입력 신호 행렬 $\boldsymbol{X}(n)$ 에 대하여 살펴보려고 한다. 다음과 같은 관계식을 만족시키는 ULT(Unit Lower Triangular) 변환 행렬 $\boldsymbol{T}$ 가 존재한다고 가정하자.

$$\boldsymbol{Z}(n) = \boldsymbol{X}(n)\,\boldsymbol{T} \tag{11}$$

여기서 $\boldsymbol{Z}(n)$은 입력신호 행렬 $\boldsymbol{X}(n)$이 변환 행렬 $\boldsymbol{T}$ 를 거쳐서 출력되는 $L \times F$ 신호행렬로서 다음과 같이 표현되며,

$$Z(n) = [\ z_1(n)\ z_2(n)\ \cdots\ z_P(n)],$$
$$z_i(n) = [z_{i,1}(n)\ z_{i,2}(n)\ \cdots z_{i,L}(n)]^T, \quad i=1,2,\cdots P \tag{12}$$

이때, $Z(n)$을 구성하는 각각의 element 벡터들은 $z_i(n)$은 $z_i(n)^T z_j(n)=0,\ i\neq j$ 를 만족한다.
즉,

$$Z(n)^T Z(n) = diag\ [\ z_1(n)^T z_1(n)\ \ z_2(n)^T z_2(n)\ \cdots\ z_P(n)^T z_P(n)\ ] \tag{13}$$

이 된다.

변환 행렬 T 는 ULT 행렬이고 determinant가 1 이므로 항상 역행렬이 존재하게 되며 따라서 다음과 같은 관계가 성립한다.

$$X(n) = Z(n)\ T^{-1} \tag{14}$$

따라서 (11), (13), (14) 관계식들을 이용하면 식 (6)은 다음과 같이 다시 쓸 수 있다.

$$X(n)[\ X(n)^T X(n)]^{-1} e(n) = Z(n)[\ Z(n)^T Z(n)]^{-1}\ T^T e(n) \tag{15}$$

그런데 윗 식에서 $Z(n)^T Z(n)$ 행렬은 식 (13)과 같이 표현되는 대각 행렬이다. 또한 변환행렬 T 는 ULT 행렬이므로 T^T 는 UUT(Unit Upper Triangular) 변환 행렬이 된다. 오차벡터 $e(n)$ 은 식 (10)과 같이 표현되므로 $T^T e(n) = e(n)$이 됨을 알 수 있고, 따라서 식(6)은 다음과 같이 간단하게 표현될 수 있다.

$$\Delta h(n) = \begin{bmatrix} \frac{z_1(n)}{z_1(n)^T z_1(n)} \\ \frac{z_2(n)}{z_2(n)^T z_2(n)} \\ \cdots \\ \frac{z_P(n)}{z_P(n)^T z_P(n)} \end{bmatrix}^T \begin{bmatrix} e(n) \\ 3 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$= \frac{z_1(n)}{z_1(n)^T z_1(n)} e(n) \tag{16}$$

식 (16)의 결과를 이용하여 affine projection 알고리즘의 계수 적응식을 다시 표현하면 다음과 같은 관계식을 얻어낼 수 있다.

$$h(n+1) = h(n) + \mu \frac{z_1(n)}{z_1(n)^T z_1(n)} e(n) \tag{17}$$

여기서 μ 는 알고리즘의 오차를 조절하기 위하여 도입한 수렴상수이다.

따라서, 식(15)의 관계를 이용하면 affine projection 계수갱신을 위하여 역행렬 $[\ X(n)^T X(n)]^{-1}$을 계산하는 대신 ULT 행렬 T 에 의해 변환된 최종오차벡터 $z_1(n)$을 이용함으로서 벡터연산만으로 계수갱신이 가능해 진다. 새로운 적응식은 NLMS와 같은 형태를 띠게 되며 적응 알고리즘이 수렴하기 위한 수렴상수의 범위는 $0<\mu<2$ 가 된다.

한편 식 (11)~(14)의 관계를 만족시키는 변환행렬 T 는 Gram-Schmidt 직교화 과정을 통해 쉽게 얻을 수 있으며[10], 이 때 T 는 입력 신호 벡터들 $x_L(n)$, $x_L(n-1)$, $\cdots$, $x_L(n-P+1)$ 간의 Gram-Schmidt 직교화 계수를 의미하게 된다. 일반적으로 변환행렬 T 는 다음과 같이 정의되고[8],

$$T = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ c_{1,1} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ c_{P-1,P-1} & c_{P-1,P-2} & \cdots & 1 \end{bmatrix} \tag{18}$$

또한 $(P-1)$ 개의 ULT 행렬로 분할 가능하다[10].

$$T = T_1\ T_2 \cdots\ T_{P-1} \tag{19}$$

이 때 T_i 는 i 번째 행을 구성하는 원소만이 값을 가지는 ULT 행렬이다. 편의상 $P=4$ 라고 하면,

$$T=\begin{bmatrix} 1 & 0 & 0 & 0 \\ -\alpha_{1,2} & 1 & 0 & 0 \\ -\alpha_{1,1} & 0 & 1 & 0 \\ -\alpha_{1,0} & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -\alpha_{2,1} & 1 & 0 \\ 0 & -\alpha_{2,0} & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\alpha_{3,0} & 1 \end{bmatrix} = T_1\ T_2\ T_3 \tag{20}$$

와 같이 표현된다. 최종 오차벡터 $z_1(n)$ 을 구하는 과정을 이해하기 쉽도록 그림으로 나타내면 그림 1과 같다.

그림 1과 식 (20) 으로부터 T_i 는 i 번째 stage 에서의 Gram-Schmidt 필터계수로 구성된 행렬임을 알 수 있다. 그림 1로부터 Gram-Schmidt 직교를 위한 적응필터 계수 $\alpha_{i,j}(n)$ 을 구해보면 다음과 같다.

$$\alpha_{i,j}(n) = \frac{g_{i-1,P-i}(n)^T\, g_{i-1,j}(n)}{g_{i-1,P-i}(n)^T\, g_{i-1,P-i}(n)} \tag{21}$$

여기서 $g_{i,j}(n)= g_{i-1,j}(n)-\alpha_{i,j}(n)\ g_{i-1,P-i}(n)$ 이고 $g_{0,i}= x_L(n-i+1)$ 이다.

이상에서 살펴본 바와 같이 P 차 affine projection 알고리즘에서 계수갱신에 필요한 벡터를 구하는 과정은 입력 신호 벡터를 주 입력 신호 벡터로 하고 과거 $(P-1)$ 개의 입력 신호 벡터들을 참조신호로 하여 블록 단위의 Gram-Schmidt 직교

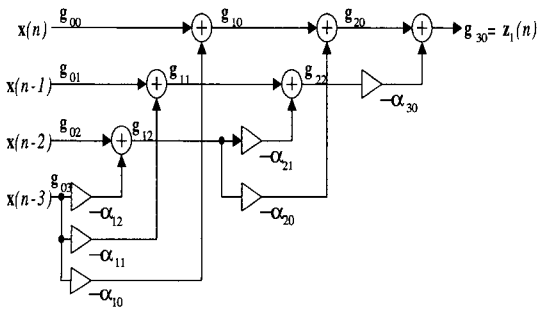


그림 1. Affine Projection 알고리즘의 Geometric Interpretation (P=4)

화와 동일해 진다. 다음 장에서는 이렇게 Gram-Schmidt 직교화를 이용하여 오차신호를 얻어낼 수 있는 원리에 착안하여 NLMS와 비슷한 계산량만으로 affine projection 알고리즘을 근사적으로 구현할 수 있는 새로운 방법을 제안한다.

Ⅲ. Affine projection 알고리즘의 근사적 구현

Affine projection 알고리즘을 윗 절에서 살펴본 바와 같이 벡터단위의 Gram-Schmidt 구조[11]를 이용하여 구현하는 경우 최종 오차벡터를 구하기 위하여 $(P-1)$ 단계를 거쳐서 매 단계마다 오차벡커들을 계산해야 한다. 따라서 최종 오차벡터를 구하기 위하여 총 $(P-1)^2L+(P-1)L$ 번의 연산이 필요하게 되므로 역행렬을 구하지 않더라도 projection 차수 P 가 증가함에 따라 계산량이 LP^2 에 비례하여 증가하게 된다. 결국 Gram-Schmidt 직교화를 이용하면 affine projection 알고리즘을 정확하게 구현할 수는 있으나 많은 계산량을 요구하므로 실지 구현에는 어려움이 있다.

본 연구에서는 이렇게 벡터단위의 Gram-Schmidt 직교화를 이용하여 계수적응에 필요한 오차신호벡터를 정확히 구할 수 있는 원리에 착안하여 입력신호를 샘플단위로 받아들이는 conventional Gram-Schmidt 직교화를 이용하여 오차신호를 계산해 냄으로서 계산량을 현저히 줄일 수 있는 새로운 알고리즘을 제안하고자 한다.

1. Gram-Schmidt 직교화를 이용한 오차벡터의 계산

식 (1)과 식 (12)를 대입하여 식 (11)을 다시 표현하면 다음과 같다.

$$[\ z_1(n)\ z_2(n)\cdots z_P(n)] = [\ x_L(n)\ x_L(n-1)\cdots x_L(n-P+1)]\ T(n) \tag{22}$$

이 때 $\mathfrak{I}_i(n)$을 다음과 같이 정의하자.

$$\mathfrak{I}_i(n)= T(n)\ 1_i \tag{23}$$

여기서 1_i 는 i 번째 element만 1이고 나머지

element는 모두 0인 벡터이다.

$\mathcal{T}_i(n)$을 이용하여 식 (22)를 다음과 같이 표현할 수 있다.

$$z_i(n) = X(n)\ \mathcal{T}_i(n), \quad i=1,2,\cdots,F \tag{24}$$

그런데 계수적응을 위하여 $z_1(n)$ 벡터만이 필요하므로 식 (24)를 $z_1(n)$ 에 대하여 표현하면 다음과 같다.

$$z_1(n) = X(n)\ \mathcal{T}_1(n) \tag{25}$$

편의상 F 개의 데이터로 구성된 입력벡터를 $x_P(n)$ 이라 하자. 입력 신호 행렬 $X(n)$은 다음과 같이 표현될 수도 있다.

$$X(n)=[\ x_L(n)\ x_L(n-1)\cdots x_L(n-P+1)]$$

$$=\begin{bmatrix} x(n) & x(n-1)\cdots & x(n-P+1) \\ x(n-1) & x(n-2)\cdots & x(n-P) \\ \vdots & & \vdots \\ x(n-L+1) & x(n-L)\cdots & x(n-L-P+2) \end{bmatrix}$$

$$=\begin{bmatrix} x_P(n)^T \\ x_P(n-1)^T \\ \vdots \\ x_P(n-L+1)^T \end{bmatrix} \tag{26}$$

따라서 오차벡터 $z_1(n)$ 을 구성하는 각각의 element들은 다음과 같이 계산되어 진다.

$$z_{1,j}(n) = x_P(n-j+1)^T\ \mathcal{T}_1(n), \quad j=1,2,\cdots,L \tag{27}$$

만약 $n > 2L+F$ 이고 $2L$ 샘플 구간동안 입력신호의 통계적 특성이 변하지 않는다면 $T(n)$을 구성하는 적응필터의 계수들은 일정한 값을 유지하게 된다. 즉,

$$\mathcal{T}_1(n) = \mathcal{T}_1(n-1) = \cdots = \mathcal{T}_1(n-L+1) \tag{28}$$

따라서 다음과 같은 관계가 성립한다.

$$z_1(n) = X(n)\ \mathcal{T}_1(n)$$

$$\cong\begin{bmatrix} x_P(n)^T\ \mathcal{T}_1(n) \\ x_P(n-1)^T\ \mathcal{T}_1(n-1) \\ \cdots \\ x_P(n-L+1)^T\ \mathcal{T}_1(n-L+1) \end{bmatrix}$$

$$=\begin{bmatrix} z_{1,1}(n) \\ z_{1,1}(n-1) \\ \vdots \\ z_{1,1}(n-L+1) \end{bmatrix} \tag{29}$$

즉, $z_1(n)$ 을 구성하기 위하여 매 샘플마다 벡터연산을 하는 대신 F 개의 신호를 입력으로 하는 샘플단위의 Gram-Schmidt 직교화를 통하여 오차신호 $z_{1,1}(n)$을 구한 다음 TDL 구조를 이용하여 이 값을 하나씩 shift 시킴으로서 $z_1(n)$을 근사적으로 구할 수 있는 것이다. 결국 벡터 단위의 연산이 필요하지 않으므로 계산량을 현저히 줄일 수 있게 된다. 따라서 본 논문에서는 위와 같은 방법으로 $z_1(n)$을 구성하고 이를 계수적응에 이용함으로서 고속의 affine projection 알고리즘을 구현하고자 한다. 제안한 방법의 블럭 구성도는 그림 2와 같다.

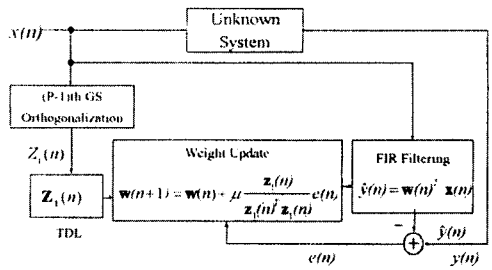


그림 2. 제안된 고속 affine projection 알고리즘의 블럭 구성도

2. Gram-Schmidt 직교화 필터계수의 계산

앞장에서 affine projection 알고리즘을 Gram-Schmidt 직교화를 이용하여 근사적으로 구현하는 방법을 제시하였다. 오차벡터 $z_1(n)$은 샘플단위의

conventional Gram-Schmidt 직교화를 이용하여 구할 수 있지만 Gram-Schmidt 직교화를 위한 적응필터의 계수는 매 샘플마다 벡터단위의 직교가 일어나도록 구해져야 한다. 따라서 식 (21)과 같이 Gram-Schmidt 직교필터의 계수를 정확하게 구하기 위해서는 벡터 단위의 연산이 필요하게 된다. 그러나 LMS 알고리즘이나 Least-Square 알고리즘 등과 같은 적응 알고리즘을 사용하면 Gram-Schmidt 직교 필터의 계수를 recursive하게 구할 수 있다[10][11]. Gram-Schmidt 직교 필터의 계수를 구하기 위한 적응 알고리즘으로는 크게 두 가지를 생각할 수 있는데, 하나는 입력신호에 rectangular window를 씌워 처리하는 것이고 다른 하나는 입력신호에 exponential window를 씌워 처리하는 것이다. 전자의 경우는 Levinson-Durbin 알고리즘을 이용하여 투영계수를 order-recursive하게 구할 수 있으며[8] 후자의 경우는 weighted Least-Square estimation을 이용하여 time-recursive하게 구할 수 있다[11]. 본 논문에서는 Ling등이 Least-Square estimation을 위해 제안한 RMGS (Recursive Modified Gram-Schmidt) 알고리즘을 이용하여 $\alpha_{i,j}(n)$을 time-recursive하게 구하였다. Weighted Least-Square estimation을 위해서는 forgetting factor λ 를 결정해야 하는데, 일반적으로 λ 는 적응필터의 길이 L 에 의해 결정되며 $1-\frac{1}{L} < \lambda < 1$ 이 적당하다[12].

RMGS 알고리즘을 이용하면 $(P-1)^2+6(P-1)$번의 연산만으로 최종 오차 벡터 $\boldsymbol{z}_1(n)$을 구할 수 있다[11]. 즉, 제안된 알고리즘은 입력신호를 Gram-Schmidt 직교화를 거쳐 상관관계가 제거된 신호를 만든 후에 이를 계수적응에 사용함으로서 affine projection을 근사적으로 구현 가능하므로 계산량은 NLMS에 비해 크게 증가하지 않으면서 affine projection 알고리즘과 비슷한 수렴성능을 가지는 장점을 가지게 된다. 제안된 알고리즘을 표 1에 정리하였다.

표 1에서 알 수 있듯이 오차벡터를 구하기 위하여 $(P-1)$ 차 Gram-Schmidt 직교화를 위하여 $(P-1)^2+6(P-1)$의 연산만이 추가로 필요하게 된다. 일반적으로 affine projection 알고리즘은 음향반향 제거기등과 같이 수백~수천탭의 길이(L)를 가지는 적응필터를 요구하는 응용분야에 적용되며 음성신호와 같이 AR proecss로 모델링이 가능한 입력신호를 대상으로 하는 경우 일반적으로 $P=10$ 정도로 모델링한다. 따라서 $P \ll L$이 성립하기 때문에 Gram-Schmidt 직교화에 필요한 계산량은 전체 계산량에 비하여 무시 가능할 정도로 적다. 따라서 제안된 방법은 NLMS 알고리즘과 비슷한 계산량으로 affine projection 알고리즘을 근사적으로 구성할 수 있게 된다. 아래 표 2 에서 적응 알고리즘의 계산량을 비교하였다.

표 1. 제안한 고속 affine projection 알고리즘

초기화

$$\beta_1(n)=1, \quad r_{ii}(-1)=\delta \quad (i=1,\cdots,P)$$

$$\boldsymbol{g}_{o,i}(n) = \boldsymbol{x}(n-i+1)$$

Gram-Schmidt 직교화

for i=1 to P-1 do

$$r_{ii}(n)=\lambda r_{ii}(n-1)+\beta_i(n)g_{i-1,P-i}(n)^2$$

$$\beta_{i+1}(n)=\beta_i(n)-\beta_i(n)^2 g_{i-1,P-i}(n)^2/r_{ii}(n)$$

for j=0 to P-1-i do

$$g_{i,j}(n)= g_{i-1,j}(n)-\alpha_{i,j}(n)\, g_{i-1,P-i}(n)$$

$$\alpha_{i,j}(n+1)=\alpha_{i,j}(n)+\beta_i(n)g_{i,j}(n)g_{i-1,P-i}(n)/r_{ii}(n)$$

오차벡터 $z_1(n)$ 구성

$$z_{1,1}(n) = g_{P-1,0}(n)$$

$$\boldsymbol{z}_1(n)=[z_{1,1}(n)\ z_{1,1}(n-1) \cdots z_{1,1}(n-L+1)]^T$$

계수적응

$$e(n) = y(n)-\boldsymbol{h}(n)^T\boldsymbol{x}(n)$$

$$\boldsymbol{h}(n+1)=\boldsymbol{h}(n)+\mu\frac{e(n)}{\boldsymbol{z}_1{}^n{}^T\boldsymbol{z}_1(n)}\boldsymbol{z}_1(n)$$

표 2. 각 적응 알고리즘의 계산량 비교
(L : 적응필터탭수, P : projection차수)

적응 알고리즘	계산량 (매 샘플당 곱셈량)
Affine Projection	$(P+1)L+O(P^3)$
NLMS	$2L$
제안된 방법	$2L+(P+2)^2-9$

Ⅳ. 실험결과

본 논문에서는 제안한 방법의 실효성을 보이기 위하여 제안된 고속 알고리즘과 affine projection 알고리즘을 음향 반향제거기에 적용하여 그 성능을 비교하였다. 입력신호로 백색잡음 신호를 그림 3과 같은 주파수 응답을 가지는 10차 AR process를 통과시켜 얻은 유색잡음 신호를 사용하였으며 반향경로는 약 $4m \times 3m$의 방에서 측정한 실지 반향경로의 임펄스 응답을 256 탭으로 잘라서 사용하였다. 반향경로의 길이와 동일한 탭을 가지도록 적응필터의 길이를 256 으로 선택하였으며 projection 차수를 각각 *P=1*(NLMS), *P=4, P=8* 인 경우에 대하여 실험하였다. 입력신호에 반향경로를 콘벌루션 시켜서 음향반향신호를 만들었으며 약 *40 dB* 의 배경잡음을 반향신호에 더하여 주 입력신호로 사용하였다.

제안된 알고리즘의 유효성을 보이기 위하여 ERLE와 *Misalignment* 를 비교하였다. ERLE는 반향신호와 반향이 제거된 신호간의 전력비를 나타내는 것으로 다음 식과 같이 정의된다.

$$ERLE = 10\log_{10}\frac{\sum_{i=0}^{N-1} y(n-i)^2}{\sum_{i=0}^{N-1}(y(n-i)-\hat{y}(n-i))^2}(dB) \quad (30)$$

또한 *Misalignment* 는 실지 시스템 응답과 적응알고리즘을 이용하여 구한 적응필터 계수간의 불일치 정도를 의미하는데, 즉 적응필터가 얼마나 실지 응답에 가깝게 수렴했는지를 나타내는 척도이다.

$$Misalignment = 10\log_{10}\frac{E\{(\hat{\boldsymbol{h}}-\boldsymbol{h})^T(\hat{\boldsymbol{h}}-\boldsymbol{h})\}}{E\{\boldsymbol{h}^T\boldsymbol{h}\}}(dB) \quad (31)$$

그림 4는 affine projection 알고리즘과 본 논문에서 제안한 근사적 affine projection 알고리즘을 각각 사용하였을 경우의 ERLE를 비교한 것이고 그림 5는 *Misalignment* 를 비교한 것이다. 그림으로부터 알 수 있듯이 Gram-Schmidt를 이용하여 근사적으로 affine projection 알고리즘을 구현한 경우 affine projection 알고리즘과의 성능차이를 느낄 수 없다.

또한, 제안한 방법의 계산량은 NLMS 계산량 (약 512)에 비하여 *P=4, P=8* 인 경우에 각각 27, 92번의 추가연산만을 필요로 하게 되며, 적응필터의 길이가 길어질수록 전체 계산량에 비하여 추가 연산량이 차지하는 비중이 점점 줄어들게 된다. 따라서, 제안한 방법으로 affine projection 알고리즘을 근사적으로 구현하는 경우, NLMS 알고리즘과 비슷한 계산량만으로 affine projection 알고리즘의 성능을 그대로 유지할 수 있다.

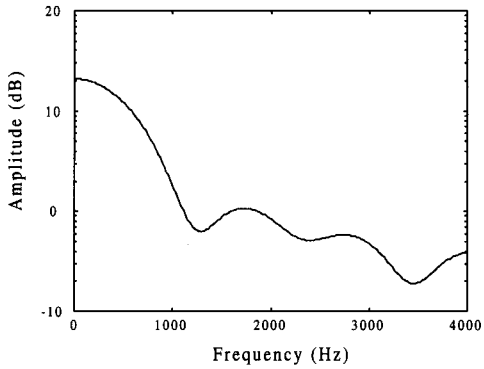


그림 3. AR 필터의 주파수 응답

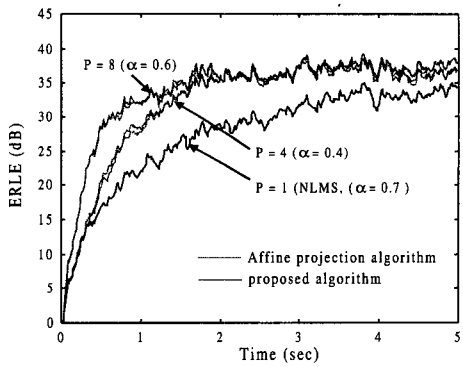


그림 4. Projection 차수에 따른 ERLE 비교

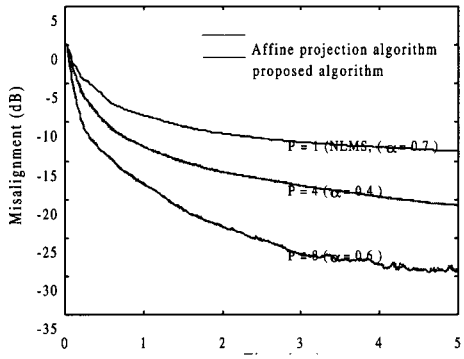


그림 5. Projection 차수에 따른 Misalignment 비교

V. 결 론

본 논문에서는 Gram-Schmidt 직교화를 이용하여 고속 affine projection 알고리즘을 근사적으로 구현하였다. 제안된 알고리즘은 NLMS 알고리즘에 비하여 약 $(P-1)^2+6(P-1)$ 번의 추가연산만으로 affine projection 알고리즘의 성능을 그대로 유지할 수 있으므로 음성신호등과 같이 AR process로 모델링이 가능한 신호를 대상으로 하는 응용분야에 매우 효율적으로 적용가능하다. 음향반향 제거기에 적용하여 모의 실험한 결과 매우 우수한 성능을 보였다.

참 고 문 헌

[1] T. Furukawa, H. Kubota and S. Tsuji, "The Orthogonal Projection Algorithm for Block Adaptive Signal Processing," *Proc. ICASSP*, pp.1059-1062, 1989

[2] K. Ozeki and T. Umeda, "An Adaptive Filtering Algorithm Using an Orthogonal Projection to an Affine Subspace and Its Property," *Electron. Commun. Japan,* Vol. 67-A, pp.19-27, 1984.

[3] D. Morgan and S. Kratzer, "On a Class of Computationally Efficient, Rapidly Converging, Generalized NLMS Algorithms," *IEEE Signal Processing Letters*, Vol. 3, No. 8, Aug. 1996.

[4] P. C. W. Sommen and C. J. van Valburg, "Efficient Realisation of Adaptive Filter Using an Orthogonal Projection Method," *Proc. ICASSP*, pp.940-943, 1989.

[5] M. Tanaka, Y. Kaneda, S. Makino and J. Kojima, "A Fast Projection Algorithm for Adaptive Filtering", *IEICE Trans. Fund.*, vol.E78-A, No.10, pp.1355-1361, Oct. 1995.

[6] M. Tanaka, Y. Kaneda, S. Makino and J. Kojima, "Fast Projection Algorithm and Its Step Size Control," *Proc. ICASSP*, pp.945-948, 1995.

[7] S. L. Gay and S. Tavathia, "The Fast Projection Algorithm," *Proc. ICASSP*, pp. 3023-3026, 1995.

[8] S. Haykin, *Adaptive Filter Theory*, Prentice Hall. 1991.

[9] S. Makino and Y. Kaneda, "Exponentially Weighted Step-Size Projection Algorithm for Acoustic Echo Cancellers", *IEICE Trans. Fundamentals*, vol.E75-A, No.11, pp.1500-1507, Nov. 1992.

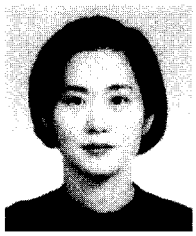
[10] N. Ahmed and D. H. Youn, "On a Realization and Related Algorithm for Adaptive Prediction", *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol.ASSP-28, No.5, pp. 493-497, oct. 1980.

[11] F. Ling, D. Manolakis and J. G. Proakis, "A Recursive Modified Gram-Schmidt Algorithm for Least-Squares Estimation", *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol.ASSP-34, No.4, pp.829-836, Aug. 1986.

[12] V. J. Mathews and S. C. Douglas, *Adaptive FIlters*, 1997.

[13] M. Montazeri and P. Duhamel, "A Set of Algorithms Linking NLMS and Block RLS Algorithms", *IEEE Trans. Signal Processing*, Vol.43, No.2, pp.444-453, Feb. 1995.

김 은 숙(Eun-Sook Kim) 정회원



1988년 2월 : 연세대학교 전자공학과 졸업

1990년 2월 : 연세대학교 전자공학과 석사

1990년 3월~1993년 9월 : 한국통신 연구원 전자공학과 박사과정

1999년 8월 : 연세대학교 전기, 컴퓨터공학과 박사

<주관심 분야> 적응 디지털 신호처리, 스테레오 음향 반향 제거

정 양 원(Yang-Won Jung) 정회원

1998년 2월 : 연세대학교 전자공학과 졸업

1998년 3월~현재 : 연세대학교 전기, 컴퓨터공학과 석사과정

<주관심분야> 적응 디지털 신호처리, 스테레오 음향 반향 제거

박 선 준(Seon-Joon Park) 정회원

1996년 2월 : 연세대학교 전자공학과 졸업
1998년 2월 : 연세대학교 전자공학과 석사
1998년 3월~현재 : 연세대학교 전기, 컴퓨터공학과 박사과정

<주관심 분야> 적응 디지털 신호처리, 음질 향상, 디지털 보청 시스템, 음향 궤환 제거

박 영 철(Young-Cheol Park) 정회원

한국 통신학회지 1997년 2월호 참조
현재 : 연세대학교 신호처리 연구센터 연구 교수

윤 대 희(Dae-Hee Youn) 정회원

한국 통신학회지 1998년 5월호 참조
현재 : 연세대학교 기계, 전자공학부 교수