

dSPACE 보드를 이용한 고속 LMS와 MDF 알고리즘의 실시간 구현

조우근*, 정원용**
경남대학교

Real-time Implementation of Fast LMS and MDF Algorithms using dSPACE board

WooGuen Joh*, Wonyong Chong**

Kyungnam University

e-mail : y2kjoh@freechal.com, wychong@kyungnam.ac.kr

요약

통신기술의 발달과 정보화 사회로 빠르게 변화되면서, 유선·무선, 핸드프리, 원거리 화상회의 등의 다양한 방식의 통신이 이루어지고 있다. 음성 통신의 어려운 문제 중에 하나는 주위의 소음이다. 소음은 상황에 따라서 다양하고 복잡하여 그 특성을 분석하기가 어렵다.

소음의 특성과 반향 등을 분석하기 위해서는 수천개의 적응필터 탭이 필요하게 된다. 따라서 실시간 소음제거를 위해서는 계산량이 많아 어려움이 따르므로 계산량 감소를 위해 FFT연산에 근거한 주파수 영역의 FDAF 적응필터를 이용하게 되었다. 하지만 계산량은 상당히 감소되었지만, 적응필터의 차수가 증가하면서 시간지연과 하드웨어적으로 복잡하게 되어 블록의 차수를 줄일 수 있는 MDF를 비교 검토 하였다.

I. 서론

음성은 전달과정에서 주위의 환경으로 인해 여러 전달 경로를 갖게 된다. 여러 전달 경로를 통해서 마이크로폰으로 전해진 음성은 주위 소음과 반향 등으로 인해 음성학적인 명확도가 크게 저하될 수 있다. 다양한 환경에 적응하며 긴 임펄스 응답을 갖는 시스템에 적합한 알고리즘으로 LMS 계열의 알고리즘이 많이 사용하고 있다. FIR시스템으로 수렴특성이 잘 알려져 있고 또 적은 연산

량과 시스템 안정성의 장점이 있는 반면 통계적 컨벌루션 관계로 인해 필터의 수렴 속도 및 정확도가 변하는 문제점을 갖는다. 또한 환경에 따라서 수천개의 필터 탭이 요구되므로 LMS 계열의 알고리즘에서는 연산량 급증으로 실시간 처리가 어려워지는 동시에 시간지연의 문제가 발생한다. 이러한 문제 해결을 위해 다양한 연구들이 진행되어왔으며, 크게 연산량을 감소시키는 연구와 수렴 속도를 증가시키는 두가지 방향으로 연구가 진행되었다 [1]

연산량을 줄이기 위한 한 방법으로 블록 LMS 방법을 이용한 주파수 영역의 적응 필터를 이용하는 방법이 있다. 1992년 Shynk에 의하여 overlap-save 방법을 이용한 고속 LMS 방법이 완성되었다. 그러나 고속 LMS 방법은 계산량을 줄일 수 있었지만, 적응필터의 차수가 길어지면 긴 차수의 FFT로 인해서 시간 지연이 발생하고 출력력이 지연되며 하드웨어적인 복잡도가 증가하게 된다. 이러한 문제를 해결하기 위해 블록의 차수를 줄일 수 있는 MDF(Multi-Delay Frequency domain adaptive filter) 방법은 적응하는 필터의 입력값을 저장하여 연산하는 블록의 단위를 줄일 수 있는 방법이다. 블록의 단위 감소로 필터가 적응하는데 발생하는 지연을 감소시킬 수 있고 긴 차수의 FFT로 인한 실행 시간 지연을 줄일 수 있다.

본 연구에서는 고속 LMS와 MDF를 dSPACE 실시간 처리 보드상에서 직접 구현하고 소음으로 오염된 음성신호를 실시간적으로 처리하여 그 결

과를 직접 확인하고자 한다.

II. 이론

2.1 BLMS(Block LMS)

매 샘플마다 이루어지는 적응 디지털 필터 계수의 갱신을 매 블록마다 즉, 매 샘플마다 램 계수를 갱신하는 LMS와 비교하여 BLMS라고 한다 [2][3].

$$w(n+1) = w(n) + \mu x(n)e(n) \quad (1)$$

$w(n)$ 은 적응 필터 계수이고, $x(n)$ 은 입력 신호 벡터로서 다음과 같다.

$$w(n) = \{ w_0(n), \dots, w_{N-1}(n) \}^T \quad (2)$$

$$x(n) = \{ x(n), \dots, x(n-N+1) \}^T \quad (3)$$

N 은 램 수이고 스텝 크기는 μ 로서 수렴비와 알고리즘의 정상 상태 수렴을 조절한다.

$$n = kL + i \quad i=0, 1, \dots, N-1, \dots \quad (4)$$

여기서 k 는 블록 시간의 값이고, n 은 원래 데이터의 샘플시간의 값을 나타낸다. 그리고 L 은 블록 크기로 램 수를 말한다. 일반적으로 필터 길이 N 과 블록 크기 L 을 같게 둔다. 이는 이때 FFT 알고리즘에서 가장 효과적인 값을 갖기 때문이다[3].

$$w(n+L+1) = w(n+L) + \mu x(n+L)e(n+L) \quad (5)$$

여기에서 계수의 갱신은 N 개의 샘플이 다 채워지기 전에는 수행하지 않는다.

k 번째 블록 관점에서 보면 적응 필터 계수와 출력은

$$w(k+1) = w(k) + \sum_{i=0}^{L-1} x(kL+i)e(kL+i) \quad (6)$$

$$y(kL+i) = \sum_{l=0}^{N-1} x(kL+i-l) w_l(k) \quad (7)$$

이 된다.

위의 식(6)과 (7)에서 알 수 있듯이 BLMS는 1회의 컨벌루션과 상관을 수행하기 위해서도 상당한 계산량이 필요하다는 것을 알 수 있다. 또한 LMS는 큰 램 수로 인해 LMS에 비해 수렴속도가 느려진다.

2.2 고속 LMS

순환 컨벌루션 방법으로 알고리즘을 수행하면 수행과정에서 원하지 않은 결과가 포함되는 것을 피하면서 주파수 영역에서 LMS 알고리즘을 수행할 수 있는 고속 LMS 알고리즘을 살펴보자. 이 방법은 FFT 알고리즘의 순환 컨벌루션 효과를 선형 컨벌루션으로 대체하기 위해 50%의 겹침을 갖는 'overlap-save' 방법을 이용하고 그림 1에 나타낸다.[1][5].

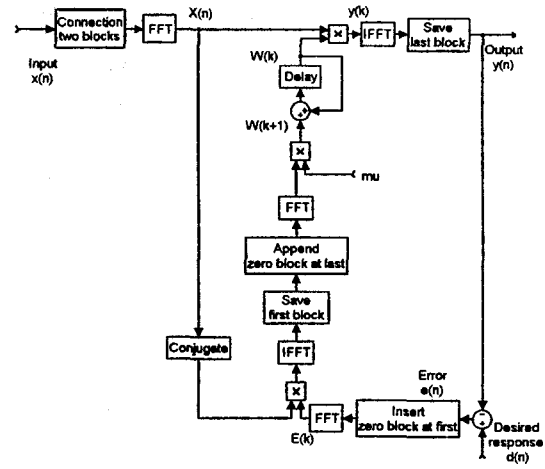


그림 1. 고속 LMS 알고리즘 블록 구조도

이 방법은 필터의 L 램을 같은 수의 0으로 채워서 $N(=2 \times L)$ 점 FFT 변환을 한다.

여기서 $N \times 1 (= (2 \times L) \times 1)$ 벡터 $W(k)$ 는

$$W(k) = FFT \begin{bmatrix} w(k) \\ 0 \end{bmatrix} \quad (8)$$

이고, 0 는 $L \times 1$ 의 0 벡터이다. $W(k)$ 는 시간 영역에서의 $w(k)$ 보다 램 수가 두 배 길다.

$$X(k) = FFT \begin{bmatrix} x(kL-L), \dots, x(kL-1) \\ x(kL), \dots, x(kL+L-1) \end{bmatrix} \quad (9)$$

앞부분의 L 개의 데이터는 $k-1$ 번째 블록 데이터이고 나머지 L 개의 데이터는 k 번째 입력 데이터이다. 여기서 50%는 과거의 데이터와 겹쳐져 있다. 즉 50% overlap 되어있다.

$$y^T(k) = [y(kL), y(kL+1), \dots, y(kL+L-1)] \\ = IFFT[X(k)W(k)] \text{의 마지막 } L \text{ 요소} \quad (10)$$

즉 $y(k)$ 는 뒷부분의 L 개 데이터만을 취한 샘플이다. 이는 순환 컨벌루션의 결과에서 알 수 있는데 앞부분에서 잘못된 값이 발생하기 때문이다

[2][5][6]. 출력 오차 신호 역시 $L \times 1$ 의 벡터로 이루어져 있다.

$$e(k) = [e(kL), e(kL+1), \dots, e(kL+L-1)]^T = d(k) - y(k) \quad (11)$$

$e(k)$ 값을 다시 주파수 영역으로 변환할 때는 선형 컨벌루션의 처음 L 개 데이터가 잘못된 값을 고려해서 다음과 같이 한다.

$$E(k) = FFT \begin{bmatrix} 0 \\ e(k) \end{bmatrix} \quad (12)$$

여기서 0 는 $L \times 1$ 의 0 벡터이다.

$$\phi(k) = IFFT [X^T(k) E(k)] \text{의 처음 } L \text{ 요소} \quad (13)$$

$\phi(k)$ 는 선형 상관에 해당하는데, 이 결과에서 확인해 보면 뒷부분 L 개 데이터에 잘못된 값이 존재한다. 마지막으로 필터 계수 갱신은

$$W(k+1) = W(k) + \mu FFT \begin{bmatrix} \phi(k) \\ 0 \end{bmatrix} \quad (14)$$

이고, fast LMS라고 한다.

2.3 MDF 알고리즘

앞 절의 고속 LMS의 가장 큰 문제점은 긴 차수로 인한 시간 지연이다. 그러므로 블록 LMS 방법으로 적용할 때 블록 벡터 크기를 줄여야만 이 문제를 해결할 있다. MDF(Multi-Delay Frequency domain adaptive filter)를 이용하면 메모리에 저장했다가 연산을 하게 되는데 저장된

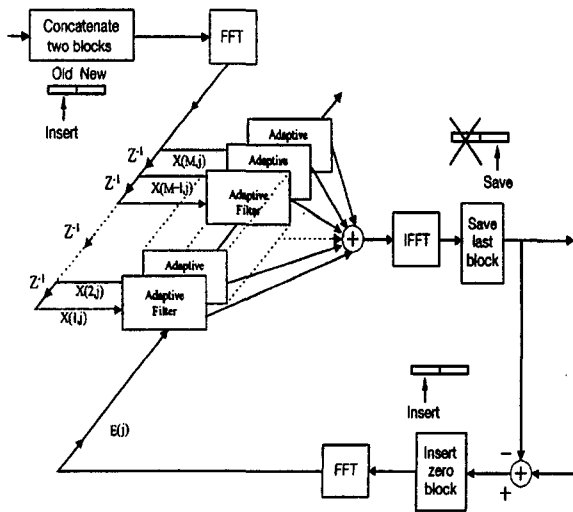


그림 2 MDF 알고리즘의 블록 구조도

데이터를 이용함으로써 FFT 차수를 줄일 수 있고 하드웨어적으로 부담을 줄일 수 있어 실제 구현에 적합하다.[6]

MDF에서 주파수 영역의 입력은 식(15) 각 데이터 값을 대각요소로 하는 행렬이다. 각 블록 index 별로 시간 영역에서 주파수 영역으로 변환시키고 지연 부분의 입력 벡터는 식(16)과 같은 방식으로 메모리에 저장된다. j 는 블록 index 값이고, M 은 지연차수를 나타낸다. FFT 연산을 하기 위하여 식(16)과 같이 블록의 한 단계 전 블록의 값을 이용한다.

$$X(M, j) = [x_0(j-1), \dots, x_{1/2M-1}(j-1), x_0(j), \dots, x_{1/2M-1}(j)]^T \quad (15)$$

$X(m, j) = X(m+1, j-1)$, $m=1, 2, \dots, M-1$ 위와 같은 방법으로 주파수 영역 입력 벡터를 만들기 때문에 하드웨어적인 부담을 줄일 수 있다. 주파수 영역의 출력은 식(17)과 같이 나타낸다.

$$Y(j) = \sum_{m=1}^M X(m, j) W(m, j) \quad (17)$$

적용 알고리즘이 M 개 중복되어 있다는 것을 제외하면 고속 LMS의 전개과정과 동일하다.

$$y(j) = IFFT [Y(j)] \text{의 마지막 } L \text{ 요소} \quad (18)$$

$$e(k) = [e(kL), e(kL+1), \dots, e(kL+L-1)]^T = d(k) - y(k) \quad (19)$$

$$E(k) = FFT \begin{bmatrix} 0 \\ e(k) \end{bmatrix} \quad (20)$$

$$\phi(m, j) = IFFT [X^T(m, j) E(j)] \text{의 처음 } L \text{ 요소} \quad (21)$$

$$W(m, j+1) = W(m, j) + \mu FFT \begin{bmatrix} \phi(m, j) \\ 0 \end{bmatrix} \quad (22)$$

Overlap-save 방법에 기반을 둔 고속 LMS의 전개식과 별다른 차이가 없다.[4][6]

III. 비교 및 분석

하드웨어로 적응필터를 사용할 경우에 DSP 칩에서 사용되는 전체 클럭 수의 70% 정도가 곱셈 연산을 하는데 사용된다. 실제 잡음제거기를 하드웨어로 구성하기 위해서는 곱셈 연산량을 감소시키는 일이 매우 중요하게 된다. 또한 하드웨어적인 복잡도 문제를 감안하여 알고리즘을 선정하게 된다.[4]

Overlap-save방법을 사용한 두 알고리즘의 곱셈 계산량을 표1에서 나타내었다. 적용 Weight의 탭 수를 1024로 계산한 결과이다. D는 MDF 지연차수로 여기에서는 4차로 연산한 알고리즘을 중심으로 비교한 결과들이다.

고속 LMS와 MDF 알고리즘을 표1을 통해 비교하면 고속 LMS의 곱셈 계산량은 동일한 weight 탭을 갖는 MDF에 비하여 0.72배 적다. 하지만 알고리즘 전체를 수행하기 위한 메모리 측면에서 보면, 적은 메모리 포인트 수를 갖는 MDF 알고리즘이 하드웨어적으로 오히려 단순하다.[6]

그림3은 두 알고리즘으로 ANC(Adaptive Noise Cancellation) 한 결과를 나타내고 있다. 그림은 위에서부터 순서대로 원신호, 잡음과 혼합된 신호, 고속 LMS로 처리한 신호, 그리고 MDF로 처리한 신호이다. 여기서 고속 LMS로 처리된 신호는 탭수만큼 지연후에 결과를 얻을 수 있다는 것이다. 음성을 8kHz로 샘플링 했을 경우에는 1024/8000=0.128초 만큼이 지연이 발생한 것이다. 반면에 MDF의 경우에는 지연차수에 반비례하여 시간 지연이 발생한다. 즉, 실험에서는 지연차수

가 4차이므로 0.128/4=0.032초 만큼의 지연만 발생하는 것이다. 탭수의 증가로 인한 시스템 응답이 지연되는 현상을 개선할 수 있다. 두 알고리즘으로 ANC 처리 결과로 본 SNR는 서로 유사한 결과를 보였다. MDF가 고속 LMS보다 weight 갱신을 D배 많은 만큼 초기 수렴속도는 다소 뛰어나게 나타났다.

IV. 결론

긴 임펄스 응답이 갖는 적응 잡음 제거기로 LMS 기반의 알고리즘들이 많이 연구되고 있다. 계산량문제와 적정 수렴특성 갖는 알고리즘으로 주파수 영역의 고속 LMS가 있다. 하지만 FIR 시스템의 특성상 발생하는 시간 지연으로 인해 빠른 실시간 구현에 적합하지 못하다. 이 시간 지연 문제를 해결한 MDF는 고속 LMS보다는 다소 계산량이 많지만 알고리즘 전체를 수행하기 위한 메모리 측면에서 보면, 적은 메모리 포인트 수를 갖는 MDF 알고리즘이 하드웨어적으로 오히려 단순하여 실시간 처리 측면에서 적합하다고 본다.

알고리즘	단위 출력 당 곱셈 연산량
고속LMS	$10M * \log_2(2M) + 16M$
MDF	$(4D+6)M * \log_2(2M/D) + 16MD$

표1. 고속 LMS와 MDF의 단위 출력 당 곱셈량

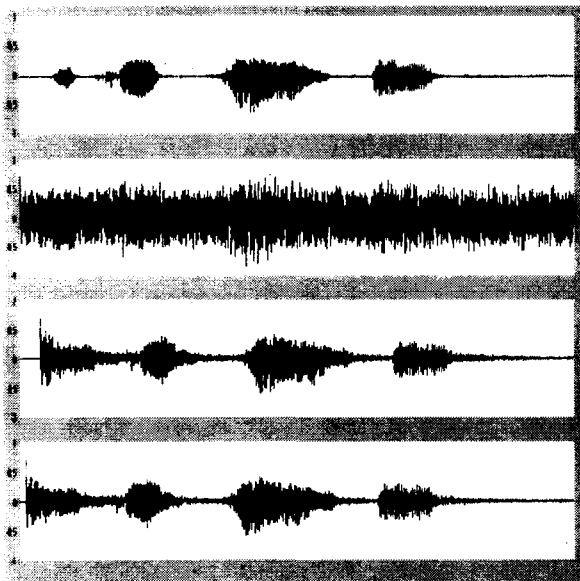


그림 3. 고속 LMS와 MDF의 ANC 수행 결과

참고 문헌

- [1] Sen M. Kuo, Dennis R. Morgan, "Active Noise Control Systems", pp24-43, 251-255
- [2] Simon Haykin, "Adaptive Filter Theory Third Edition", Upper Saddle River, NJ; Prentice-Hall, 1996.
- [3] Alan V. Oppenheim, Ronald W. Schaffer, Discrete-time Signal Processing, Englewood Cliffs, NJ: Prentice-Hall, 1989.
- [4] John J. Shynk, "Frequency-Domain and Multirate Adaptive Filtering", IEEE SP magazine. pp14-37, January 1992.
- [5] R.R. Leitch and M.O. Tokhi, 1987, "Active Noise Control Systems," IEE Proc., VOL .134, Pt. A, No.6, pp.525~546
- [6] "The generalized multidelay adaptive filter", IEEE Transactions on signal processing vol.43.NO.1, January 1993