論 文

주파수 영역에서 블럭적응 필터의 고속 수렴 알고리즘에 관한 연구

正會員 姜 哲 豪* 準會員 趙 海 南**

A Study on Fast Convergence Algorithm of Block Adaptive Filter in Frequency Domain

Chul Ho Kang * *and* Hai Nam Jo **, *Regular Members*

요 약 주파수 영역에서 블럭 적응 필터(Block Adaptive Filter)의 새로운 구현 방법을 제시하였다. 블럭 적응 필터링은 입력값을 Block 단위로 하였을때 출력에서도 블럭 단위 또는 유한한 값을 갖도록 하는 것이다. 본 논문에서는 Gordard 이론을 이용하여 Block Adaptive Filter의 고속 수렴 알고리즘을 전개하고 Saito의 알고리즘과 BLMS 알고리즘의 수렴 결과와 비교하였다. 그 결과 본 논문에서 제시한 주파수 영역블럭 적응 필터의 알고리즘(FBAF)이 BLMS 알고리즘의 수렴상태보다 빠르게 수렴하며 Saito의 알고리즘의 수렴 오차보다는 줄어들었다.

ABSTRACT A new implementation of Block Adaptive filter in frequency domain is presented in this paper. Block digital filtering involves the calculation of a block or finite set of filter out put from a block of input valt ues. A fast convergence algorithm of block adaptive filter is developed using Gordar theory and compared with the performance results of Saito algorithm and BLMS algorithm. From the result we can be shown that the convergence state of given algorithm is not only faster than BLMS algorithm but also the resulting convergence error is less than the convergence error of Saito algorithm.

1. 서 론

적응 디지탈 필터는 잡음및 echo제거, 신호의 enchancement, 선형예측, channel equalization과 같은 신호처리 분야에서 응용할 수 있다. 적응 필터의 가장 유사한 구조는 FIR(Finite Impulse Response) 필터로 구현 되는것이 일반적이다. 디지탈 필터의 블럭 실행은 속도가 빠른 병렬 processor를 사용할 수 있다. 또한 FFT(Fast Fouriler Transform)와 같은 블럭알고리즘은 직렬 processor로 필터를 구현할 때 유익하게 이용된다. 최근의 논문에서 FIR 블럭 LMS 적응 필터에 대한 이론과 구현 방법이 제시되었다[4]. 블럭 디지탈 필터링은 입력 값들을 블럭으로 하여 출력에서 블럭 또는 유한한 값을 갖도록 계산하는 것이다. 한번에 한 sample 씩 처리하는것 보다는 block으로 데이타를 처리함으로써 FIR적응 필터는 주파수 영역에서 뿐만 아니라 시간 영역에서도 효율적으로 구현될 수 있다. 한편 필터의 weight는 결정된 performance function을 최소화하는 적응 알고리즘에 의

*, ** 光云大學電子通信工學科
Dept. of Electronic Telecommunication Engineering, Kwangwoon University, Seoul. 132 Korea.
論文番號 : 85-37(接受 1985. 9. 24)

해 update된다. 일반적으로 performance function으로는 BMSE(Block Mean Square Error)가 이용된다[4]. Widrow와 Hoff가 1960년초에 최소 평균 자승(Least mean square) 알고리즘을 제시한 이래 적응 디지탈 필터에 대한 연구가 활발하게 진행되고 있다. LMS 알고리즘은 steepest descent 방법을 응용한 것이며 적응 필터 weight adaptation을 제어하는 correction 계수 μ는 안전성을 보장하기 위해 매우 작아야 한다[13]. 이와같은 조건때문에 LMS 알고리즘의 수렴 속도는 느리게 된다. 본 논문에서 제시한 알고리즘은 데이타를 블럭으로 처리함으로써 적응 필터를 구현하였으며 G. A. Clack 이론에 Gordard이론을 전개한 것으로 Tsuneo Saito가 제시한 알고리즘을 블럭 단위로 확장한 알고리즘에 대응한다. G. A. Clack이 제시한 알고리즘의 weight는 시간 영역에서 처리 하였지만 본 논문에서는 weight를 주파수 영역에서 처리 하였으며 제시된 알고리즘은 실 시간 처리(real time processing)에 적합한 방법으로 생각된다.

2. LMS 적응 필터

기존의 LMS 적응 필터는 차수가 $N-1$인 FIR 디지탈 필터이다. (그림 1) discrete-time instant K에서 출력 Y_K는 입력 x_K와 필터 weight W_K와의 중첩함으로 표시된다. 즉

$$Y_K = \sum_{i=0}^{N-1} w_{i\cdot K}\, x_{K-i} \quad K = 0,\ 1,\ 2, \qquad (1)$$

이 알고리즘은 다음 식에 의해 필터 weight 를 조절한다.

$$W_{K+1} = W_K + 2\,\mu\,\epsilon_K X_K \qquad (2)$$

여기서 μ는 수렴상수이고 W_K와 X_K는 각각 $N \times 1$인 weight vector와 $N \times 1$인 입력 vector이다.

$$W_K = [W_{0\cdot K},\ W_{1K},\ W_{2K},\ \cdots\cdots W_{(N-1)\,K}]^T \qquad (3)$$

$$X_K = [x_K,\ x_{K-1} \cdots\cdots x_{K-N+1}]^T \qquad (4)$$

그리고 E_K는 K번째 순간에서 실제 출력 Y_K와 바라는 응답 d_K간에 차로 주어지는 오차이다.

$$E_K = d_K - Y_K \qquad (5)$$

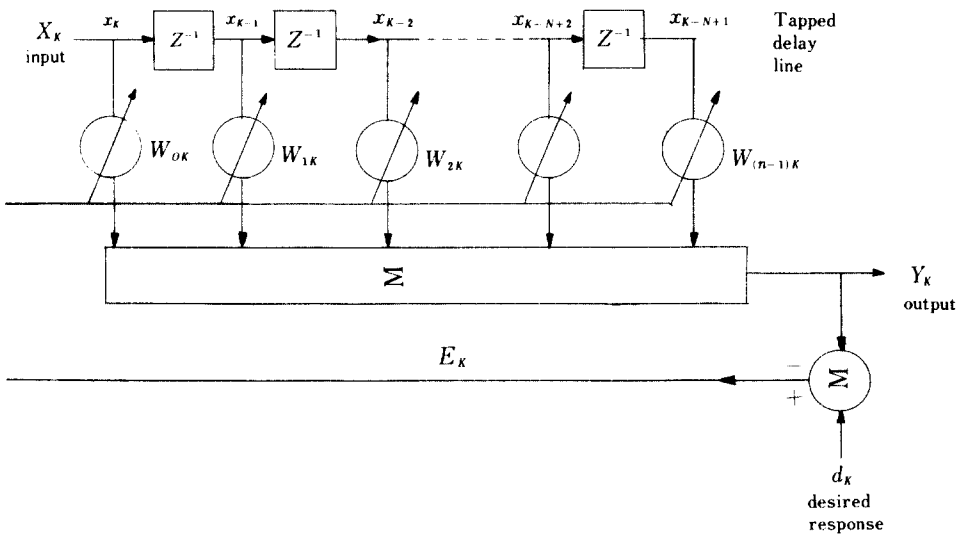


그림 1 LMS 적응 필터링
Conventional LMS Adaptive filtering.

3. BLOCK LMS 적응 필터

$J=0, 1, 2\cdots$를 block number라하고 L을 block length라 할때 블럭 적응 필터에 대한 두 개의 서로 다른 vector가 있다. 그 하나는 시간 K에서 필터의 레지스터에 저장된 입력 $N\times 1$ 인 vector X_K이고 또 하나는 $L\times 1$ vector로 표시되는 J 번째 입력 데이타 block X_L이다[4].

$$X_K=[x_K\ x_{K-1}\ \cdots\cdots\ x_{K-N+1}]^T \quad (6)$$

$$X_L=[X_{JL}\ X_{JL+1}\cdots\cdots X_{(J+1)L-1}]^T \quad (7)$$

마찬가지로 J 번째 출력 데이타 block Y_J는 $L\times 1$ vector로 주어진다.

$$Y_J=[Y_{JL}\ Y_{JL+1}\cdots\cdots\ Y_{(J+1)L-1}]^T \quad (8)$$

필터 weight는 블럭을 기초로 하여 조절되며 J 번째 block에 대한 weight W_J는 $N\times 1$ vector로 표시된다.

$$W_J=[W_{0J}\ W_{1J}\cdots\cdots W_{(N-1)\ J}]^T \quad (9)$$

여기서 첨자는 개개의 필터 weight를 나타낸다. 따라서 블럭 적응 필터 출력은 다음식으로 표시된다.

$$Y_J=\chi_J W_J \text{ (단 } J=0, 1, 2,\cdots) \quad (10)$$

이때 χ_J는 $L\times N$인 Matrix이다.

$$\chi_J=\begin{pmatrix} x_{JL} & x_{JL-1}\cdots\cdots & x_{JL-N+1} \\ x_{JL+1} & x_{JL}\cdots\cdots & x_{JL-N+2} \\ \vdots & \vdots & \vdots \\ x_{(J+1)L-1} & x_{(J+1)L-2}\cdots\cdots & x_{(J+1)L-N} \end{pmatrix}$$

$$=[X_{JL}\vdots X_{JL-1}\vdots\cdots\cdots\vdots X_{JL-N+1}] \quad (11)$$

partion by columns

$$=[X_{JL}\vdots X_{JL+1}\vdots\cdots\cdots\vdots X_{(J+1)L-1}]^T \quad (12)$$

partion by rows

식(9)와 식(12)를 식(10)에 대입하면 다음과 같이 표시된다.

$$Y_{JL+K}=X_{JL+K}^T\ W_J \text{ (단 } K=0, 1, 2, \cdots\cdots L-1) \quad (13)$$

식(13)은 linear convolution처럼 J 번째 블럭 출력 데이타인 Y_J의 element를 구성한다. 식(13)은 간단한 계산과정에 의해 나음과 같이 쓸 수 있다.

$$Y_{JL+K}=\sum_{i=0}^{N-1} x_{JL+K-i}\ W_{i\cdot J} \quad K=0, 1, \cdots\cdots L-1 \quad (14)$$

한편 필터 weight는 다음 식에 따라서 update된다.

$$W_{J+1}=W_J+2\mu_B/L\cdot\Phi_J \quad (15)$$

여기서 Φ_J는 다음 식으로 주어지며 element가 N개인 gradient 추정 vector이다.

$$\Phi_J=[\psi_{0J}\ \psi_{1J}\cdots\cdots\varphi_{(N-1)\cdot J\ J}]^T=\chi_J^T E_J \quad (16)$$

여기서 E_J는 $L\times 1$ 오차vector로써 다음 식으로 표시된다.

$$E_J=[\epsilon_{JL}\ \epsilon_{JL+1}\cdots\cdots\epsilon_{(J+1)L-1}]^T \quad (17)$$

그리고 $E_J=D_J-Y_J$이며 D_J는 J번째 desired response이고 블럭형태로써는 다음과 같이 표시된다.

$$D_J=[d_{JL}\ d_{JL+1}\cdots\cdots d_{(J+1)L-1}]^T \quad (18)$$

식(11)과 식(17)을 식(16)에 대입하면 J 번째 블럭 평균 자승 오차 gradient 추정 vector의 element인 표현식을 얻는다.

$$\varphi_{i\cdot J}=X_{JL-i}^T E_J \quad i=0, 1, \cdots\cdots N-1 \quad (19)$$

식(19)은 J번째 오차 블럭과 $(JL-i)$번째 입력데이타 block과의 correlation을 나타낸다. 위의 식에서처럼 블럭 적응 알고리즘은 식(10)의 block convolution, 식(16)의 gradient estimation, 그리고 식(15)의 weight update의 연속적인 연산으로 이루어 진다. 한편 입력을 블럭 단위로 처리할 경우 $L=N$인 경우가 가장 이상적이다[4]. 본 논문에서는 $L=N=4$, $L=N=2$로 하였다. 그리고 그림 2는 블럭 적응 밀터의 block-diagram이다.

$$Y=W_J\chi_J \tag{20}$$

또한 L-point 출력 신호를 산출하기 위해 IFFT가 수행된다. 이 weighted output은 L-point complex error 신호를 형성하기 위해 대응하는 주파수에서 바라는 응답값과 비교된다.

$$E_J=D_J-Y_J \tag{21}$$

여기서 E_J는 $L\times 1$인 vector이다. 만약 입력신

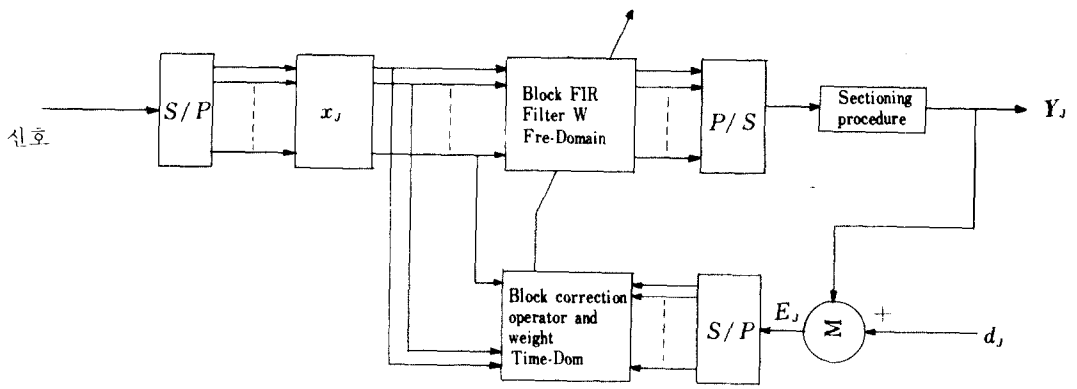


그림 2 BLMS 적응 필터링
BLMS Adaptive Filtering.

4. 주파수 영역에서의 블럭적응필터(FBAF)

4-1. 필터구조

주파수 영역 적응 디지탈 필터는 L point 데이타 블럭을 형성하기 위해 입력 신호 χ_J와 바라는 응답D_J을 sectioning함으로써 그림 3에 나타낸것처럼 구현된다. 이때 입력신호와 바라는 응답은 L-point Discrete Fourier Transform에 의해 주파수 영역으로 변환된다. $\chi_{(J+1)L-1}$와 $d_{(J+1)L-1}$를 입력신호와 바라는 응답의 J번째 block 데이타의 $(J+1)L-1$번째 주파수 bin이라 한다. 여기서 χ_J는 $L\times N$ 행렬이며 D_J는 $L\times 1$인 vector이다. 그리고 각 주파수 bin에 대응하는 L개의 complex weights가 있다. 그림 3에서 weighted output은 다음과 같다.

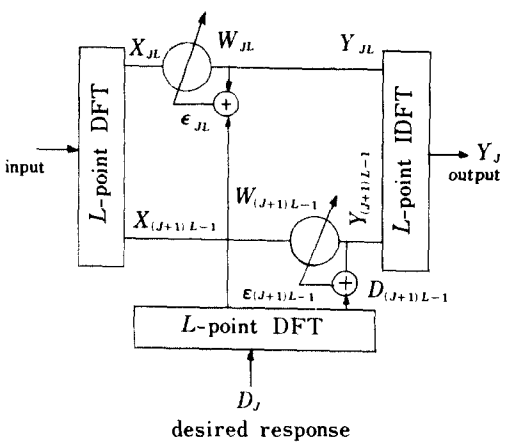


그림 3 주파수 영역에서의 블럭 적응 필터링
Block adaptive filtering in Frequency domain.

호가 관측시간동안 wide-sense stationary 하다면 주파수 성분들은 orthogonal 하고 L개의

complex weights는 독립적으로 update된다. 그러므로 주파수 영역 블럭 적응 필터는 단일 weight가 L개로 구성된 적응 필터로 생각할 수 있다.

4-2. 고속 수렴 알고리즘

이 절에서는 주파수 영역 블럭 적응 필터에 대한 fast convergence algorithm을 제시한다. 그리고 weight update 방정식은 다음과 같이 주어진다[12].

$$W_{J+1}=W_J+P_J\chi_J^*E_J \tag{22}$$

여기서 $P_J=[P_{J-1}^{-1}+\chi_J\chi_J^*]^{-1}$ (23)

*는 complex conjugate를 나타내며 variable cofficient P_J는 수렴비율을 제어하고 입력신호에 종속 되기때문에 weight update식이 빠르게 수렴할 것으로 기대된다. 그러므로 식(23)은 다음과 같이 다시 쓸 수있다.

$$P_J=[P_0^{-1}+\sum_{i=1}^{J}\chi_i^*\chi_i]^{-1} \tag{24}$$

식(20)과 식(21)을 식(22)에 대입하면 다음 식을 얻는다.

$$W_{J+1}=W_J+P_J\chi_J^*D_J-P_JW_J\chi_J^*\chi_J \tag{25}$$

식(23)의 양변에 (-1)승을 곱하면

$$\chi_J^*\chi_J=P_J^{-1}-P_{J-1}^{-1} \tag{26}$$

식(26)을 식(25)에 대입하면 다음과 같다.

$$W_{J+1}P_J^{-1}=W_JP_{J-1}^{-1}+\chi_J^*D_J \tag{27}$$

여기서 $W_0=0$ 이라면 식(27)은 다음과 같이 다시 쓸 수 있다.

$$W_{J+1}=P_J\sum_{i=1}^{J}\chi_i^*D_i \tag{28}$$

따라서 식(24)을 식(28)에 대입하면 다음과 같은 식을 얻는다.

$$W_{J+1}=\frac{\sum_{i=1}^{J}\chi_i^*D_i}{P_0^{-1}+\sum_{i=1}^{J}\chi_i^*\chi_i} \tag{29}$$

J가 충분히 크고 P_0^{-1}이 작을 경우 식(29)의 분모는 입력신호의 전력 스펙트럼이 되고 반면 분자는 입력과 바라는 응답의 cross power spectrum이 된다. 그러므로 $J=\infty$이고 $L=1$일 경우에 weight는 wiener 전달 함수에 수렴한다. LMS 알고리즘의 수렴율은 입력신호의 최대 최소spectrum 밀도의 비에 달려있다. 이런 근사법에서 수렴율은 각 주파수 bin의 전력을 정규화 함으로써 가속화 된다. 제시된 알고리즘은 G. A. Clack의 블럭 필터법과는 다소 다르며 그림 3에서 보여준 주파수 영역 필터 구조를 약간 수정함으로써 실현 될 수 있다. 실질적인 실현에 있어서 starting point information P_0는 수렴반응에 크게 영향을 미친다. 그러므로 식(24)은 다음과 같이 수정될 수 있다.

$$P_J=[\lambda P_{J-1}^{-1}+\chi_J^*\chi_J]^{-1} \tag{30}$$

여기서 λ는 상수이다. $(0<\lambda<0.5)$ 따라서 식(24)과 식(29)은 다음과 같이 다시 쓸 수 있다.

$$P_J=[\lambda^JP_0^{-1}+\sum_{i=1}^{J}\lambda^{J-i}\chi_i^*\chi_i]^{-1} \tag{31}$$

$$W_{J+1}=\frac{\sum_{i=1}^{J}\lambda^{J-i}\chi_i^*D_i}{\lambda^JP_0^{-1}+\sum_{i=1}^{J}\lambda^{J-i}\chi_i^*\chi_i} \tag{32}$$

여기서 J가 클 경우 P_0의 영향은 줄어들며 이러한 성질때문에 이 알고리즘은 입력신호가 non-stationary한 경우에도 추적 할 수 있다.

5. Computer Simulation결과 및 고찰

5-1. Simulation

이 절에서는 그림 4과 같은 방법으로 컴퓨터

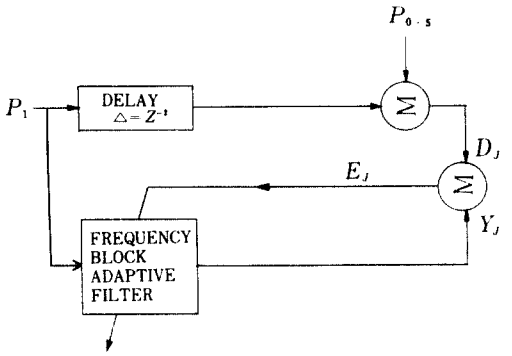


그림 4 시뮬레이션 과정
Simulation procedure.

시뮬레이션에 의해 실행된 Saito, BLMS, FBAAF(Frequency Block Adaptive Filter) 알고리즘의 실행 결과를 비교검토 하였다. 위의 알고리즘들의 수렴을 예상하기 위한 simulation과정에서 주파수 영역 블럭 적응 필터는 4개의 weights를 이용하였고 이 알고리즘들은 Z^{-2}인 unit fixed delalay를 갖는 디지탈 필터의 impulse 응답을 모델로 하여 실행하였다. 여기서 Z^{-1}은 단위지연 전달함수이다. 그림 4에서 평균이 zero이고 단위전력(P_1) 백색잡음의 독립표본들로 구성된 신호들이 적응필터와 지연필터에 병렬로 인가된다. 지연필터의 출력은 전력이 0.5($P_{0.5}$)이고 평균이 zero인 백색독립 잡음이 더하여져 적응처리의 바라는 응답 D_J로 오차신호 E_J를 형성하기 위해 적응 필터 출력 Y_J와 비교된다. 여기서 평균이 zero이고 전력이 1인 백색 Gaussian 잡음의 발생은 0～1사이의 random number를 12번 더하여 6을 빼고 표준편차 1을 곱한 것에 zero를 더하여 발생하였고 전력이 0.5인 백색 Gaussian 잡음의 발생 과정도 위와 유사하며 단지 표준편차만을 0.7로 바꾸었다. 그리고 이 두 신호는 stationary이다. 적응처리가 최적으로 수렴되었을 경우 오차는 0.5에 수렴한다. 그리고 적응 필터의 모든 초기 weight를 zero로 하고 초기 수렴을 고찰하기 위해 $P_J = 0$, $\lambda < 0.5$로 하여 실행한 알고리즘들의 결과를 그림 5, 6, 7에 제시하였다. 그림 5는 BLMS 적응 필터 알고리즘의 수렴상태의 결과와 본 논문에서 제시한 FBAF 알고리즘의 수렴 결과와 비교한 그림이며 그림 6는 FBAF와 SAITO알고리즘의 수렴 결과를 비교한 그림이다. 비교 결과 BLMS 알고리즘의 수렴상태보다 FBAF 알고리즘의 수렴상태가 빠르고 또 Saito 알고리즘의 수렴상태보다는 수렴오차가 더 적음을 알 수 있었다. 또한 그림 7은 FBAF 알고리즘의입력 데이타를 4×4, 2×2로 한 경우로 블럭 데이타를 4×4로 한 경우가 훨씬 빠르게 수렴함을 알 수 있다.

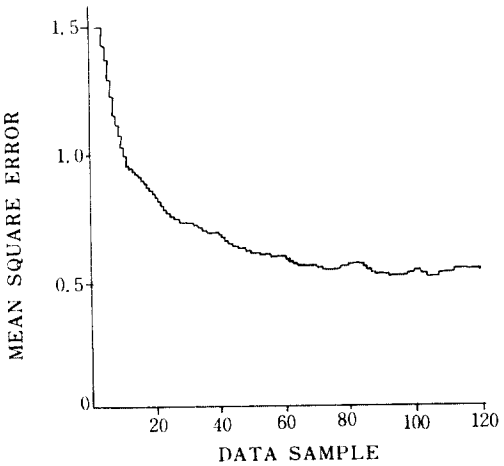
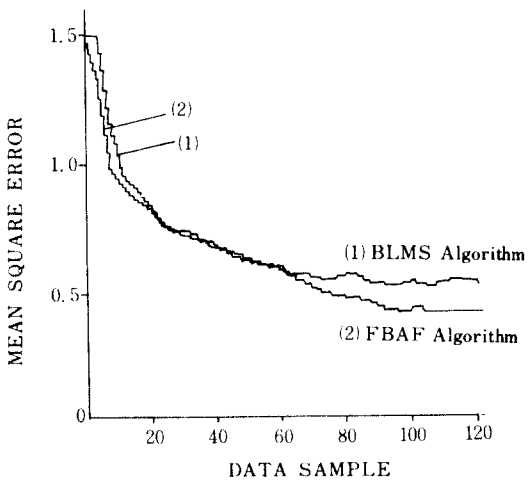


그림 5 (a) 블럭 적응필터의 수렴상태.
Convergence state of BLMS adaptive filter.



(b) BLMS와 FBAF 알고리즘의 수렴상태
Convergence state of BLMS and FBAF alogrithm.

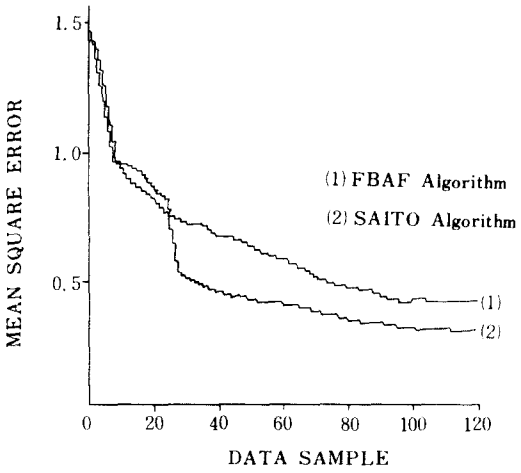


그림 6 BAF와 SAITO 알고리즘의 수렴상태.
Convergence state of FBAF and SAITO algorithm.

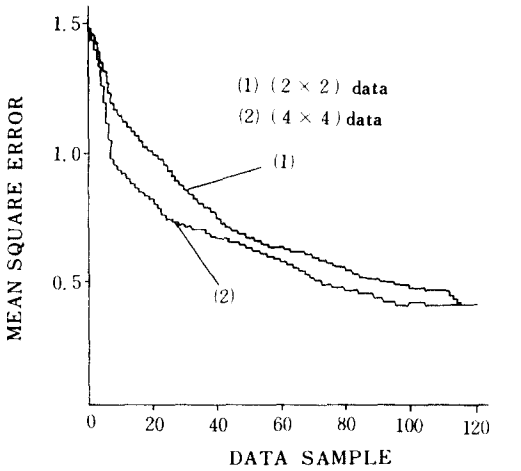


그림 7 FBAF의 (4×4), (2×2)의 수렴상태.
Convergence state of (4×4) and (2×2) for FBAF.

5-2. FBAF 알고리즘의 계산량

이 절에서는 FBAF 알고리즘의 계산량을 구하고 다른 알고리즘의 계산량과 비교한다. 기존의 LMS 알고리즘은 $K_{LMS}=2N^2$만큼의 곱셈이 필요하고 이 알고리즘을 주파수 영역에서 실행할때는 $K_{DEN}=(3N/2)\log_2^N+2N$만큼의 곱셈이 필요로 한다[1]. $N'/2$-point 복소수 FFT 알고리즘의 실수부 허수부를 적절히 이용하여 실행된 BLMS 적응 필터의 total 계산량은 $K_{BLMS}=(15/4)N'Q-20(N'/2-1)+14N'+N+b(5N')$이다[4].

나 $Q=\begin{cases}\log_2^{(N'/2)} & N' \text{ even}\\ \log_2^{(N'/4)} & N' \text{ odd}\end{cases}$ $b=\begin{cases}0 & \text{even}\\ 1 & \text{odd}\end{cases}$

한 개의 복소수 곱셈이 4개의 실수 곱셈과 같고 그림 3에서 보여준 필터 구조에서는 weight 적응이 $N/2+1$이므로 L-point 출력 데이타를 산출하기 위해 요구되는 곱셈량은 $K_{FBAF}=(3N/2)\log_2^N+(\frac{N}{2}+1)+N(L+2)$이다. 한편 Saito가 제시한 알고리즘의 계산량은 $K_F=(3N/2)\log_2^N+(6N/2)+4$가 된다[13].

표 1 계산량 비율

N	K_{LMS}/K_{FB}	K_{DEN}/K_{FBAF}	K_{BLMS}/K_{FBAF}	K_F/K_{FBAF}
32	1.560	0.231	1.149	0.259
64	1.717	0.148	0.721	0.160
128	1.828	0.089	0.440	0.095
256	1.976	0.054	0.267	0.056

위의 표에서 알 수 있듯이 K_{FBAF}의 계산량은 K_{LMS}의 계산량보다는 적음을 알 수 있다. 또 N이 32보다 적은 경우에 있어서도 K_{BLMS}의 계산량보다는 적지만 N이 32보다 클때는 주파수 영역에서 실행한 LMS, BLMS, Saito 등의 알고리즘들의 계산량보다 증가함을 알 수 있다.

5-3 Flow Chart

이 절에서는 앞의 과정을 컴퓨터로 simulation하기위한 과정을 간단히 그림 8에 flow chart로 나타냈다. 초기 weight는 zero로 하였다. 점선 괄호안의 (1)과(2)는 4-2절의 식(20)과 (21)을 계산하는 과정이고 (3)는 weight update하는 식(32)를 계산하는 과정이다. 또한 (4)는 입력을 block 단위로 읽어 들이는 과정이고 (5)는 block 단위로 입력할 데이타의 block number를 비교한다.

6. 결 론

본 논문에서는 주파수 영역에서 실행된 블럭

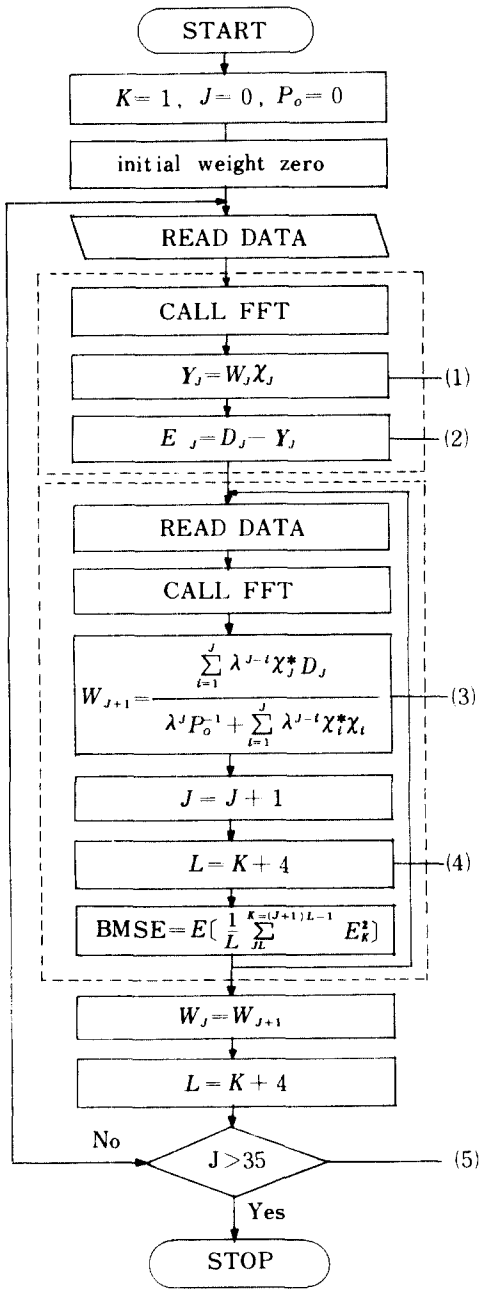


그림 8 유통도
Flow chart.

적응 필터에 대한 고속 수렴 알고리즘이 제시되었다. 제시된 알고리즘은 G. A. Clack 의 알고리즘에 Gordard 이론을 전개한 결과가 되며 또 Saito 알고리즘을 블럭화 한 것이다. 이러한 결과 BLMS 알고리즘의 수렴상태보다는 빠르게 수렴함을 알 수 있었고 Saito 알고리즘보다수렴오차를 줄일 수 있었다. 그러나 필터의 length N이 32보다 클 경우에는 BLMS, Saito Dentino 알고리즘의 계산량보다는 FBAF 알고리즘의 계산량이 증가함을 알 수 있었다.

參 考 文 獻

(1) M. J. dentino, J. McCool, and B. Widrow, "Adptive filtering in the frequency domain, "Proc. IEEE. vol. 66. pp. 1658-1659, Dcc. 1978.

(2) N. J. Bershad and P. L. Feintuch, "Analysis of the frequency domain adaptive filter, "Proc. IEEE, vol. 67. pp. 1658-1659, , Dec. 1979.

(3) E. R. Ferrara, "Fast implementation of LMS adaptive filters" IEEE Trans. Acoust, Speech, Signal Processing, vol. ASSP-28, pp. 474-475, Aug. 1980.

(4) G. A. Clark, S.K. Mitra, and S.R. Parker, "Block implementation of adaptive digital filters, "IEEE Trans. Circuits Syst., vol CAS-28, pp. 584-592, June 1981, and IEEE Trans. Acoust., Speech, Signal Processing, Joint Special Issue on Ad ptive Signal Processing, vol. ASSP-29, pp. 744-752, June 1981.

(5) A. V. Oppenheim and R. W. Schafer, Digital Signal Processing. Engelwood, Cliffs, NJ: Prentice-Hall, 1975.

(6) E. O. Brigham, The Fast Fourier Transform. Englewood Cliffs, NJ: Prentice-Hall, 1974.

(7) B. Widrow et al., "Adaptive noise cancelli ng: Principles and application, "Proc. IEEE, vol. 63, Dec. 1975.

(8) Bernari Widrow, IEEE Transaction on Antennas and Propagation, vol. Ap-24, no. 5, Sep. 1976.

(9) B. Widrow, J. Mccool , and M. Ball, "the complex Algorithm" Proc. IEEE (Collesp). vol. 63, pp. 719-720. Apr, 1975.

(10) G. A. Clacic , "A Unified Approach totime-and-Frequency domain Realilation of Fir Adaptive Digital filter, "IEEE Trans on Assp, vol. ASSP. 31 no-5, OCT 1983.

(11) D. Gordard "channel Equalilation Using Kalman filter for fast data Transmission, "IBM Journal of Reserch and Development, pp 267-273, May 1974.
Tsuneo Saito and Yukto Ho Shiko "fast Convergence

(12) Frequency Domain Adaptive Filter "The' Technology Reports of the TohoKU University vol. 48(1983) no. 2 Dec. pp, 191-197.

(13) IEEE Trasaction on Com, VOL. Com. 26 No . 10 OCT 1978 "Application of Fast Kalman Estimation to Adaptive Equalization" David D. Falconer AND Lenn RT.

(14) IEEE. Trans on Com . vol. Com-25 no. 7 JULY

1977, 666-672 "Selt-Orthogonaliling Adaptive Equalilation algorithn" Richad D. Gitlin, Francis R. Magee, JR.

(15) R. D. Gitlin, F. R. Magee, JR. "Selt-orthogonaliling adaptive equalilation algorithn," IEEE, Trans on Com. vol. com-25, no. 7 pp 666-672, 1977.

姜 哲 豪(Chul Ho Kang) 正會員

1952年 2月 5日生

1975年 2月：漢陽大學校電子工學科卒業

1979年 2月：서울大學校大學院電子工學科 卒業

1977年～1982年 2月：國方科學研究所 (研究員)

1982年 3月～現在：서울大學校大學院電子工學科(博士 課程修了)

1983年 3月～現在：光云大學電子通信科(助教授)

趙 海 南(Hai Nam Jo) 準會員

1961年 3月14日生

1984年 2月：光云大學電子通信工學科卒業.

1984年 3月～現在：光云大學大學院電子通信工學在學中.