

주파수 영역에서 블럭 적응 필터의 고속 수렴 알고리즘에 대한 연구

조 해 남, 강 철 호
 광운대학 전자통신공학과

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

Jo. Hai Nam, Kang. Chul Ho
 Dept. Electronic Communication Eng., Kwang Woon Univ.

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 values. A fast convergence algorithm of block adaptive filter is developed using Gordard 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. 서론

적용 디지털 필터는 잡음 및 echo제거, 신호의 enhancement, 선형예측, channel equalization과 같은 신호 처리 분야에서 응용할 수 있다. 적응 필터의 가장 유사한 구조는 FIR필터로 구현 되는 것이 일반적이다. 최근의 논문에서 FIR블럭 LMS 적응 필터에 대한 이론과 구현방법이 제시됐다. [4] 필터의 무게는 결정된 performance function을 최소화 하는 적응 알고리즘에 의해서 update된다. 일반적으로 performance function으로는 BMSE(Block Mean Square Error)가 이용된다. LMS알고리즘은 steepest descent 방법을 응용한 것이며 적응 필터 무게 adaptation을 제어하는 correction 계수 μ 는 안전성을 보장하기 위해 매우 작아야 한다. [13] 이와 같은 조건 때문에 LMS알고리즘의 수렴 속도는 느리게 된다. 본 논문에서 제시한 알고리즘은 데이터를 블럭으로 처리함으로써 적응 필터를 구현 하였으며 G.A.CLACK 이론에 Gordard 이론을 전개한 것으로 SAITO가 제시한 알고리즘을 블럭 단위로 확장한 알고리즘에 대응한다. G.A.CLACK이 제시한 알고리즘의 무게는 시간 영역에서 처리하였지만 본 논문에서는 무게를 주파수 영역에서 처리하였으며 제시된 알고리즘은 real time processing에 적합한 방법으로 생각된다.

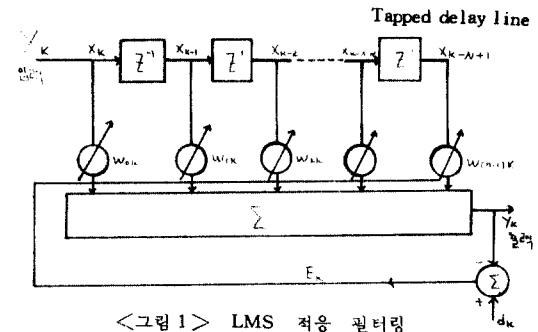
2. LMS 적응 필터

기존의 LMS 적응 필터는 차수가 N-1인 FIR 디지털 필터이다. [그림 1] discrete time instant K에서 출력 Y_k 는 입력 x_k 와 필터 무게 W_k 와의 중첩함으로써 표시된다. 즉

$$Y_k = \sum_{i=0}^{N-1} W_{i,k} x_{k-i} \quad (K=0,1,2, \dots) \quad (1)$$

이 알고리즘은 다음식에 의해 필터 무게를 조절한다.

$$W_{k+1} = W_k + 2\mu E_k \times \dots \quad (2)$$



<그림 1> LMS 적응 필터링
 Conventional LMS Adaptive Filtering

여기서 μ 는 수렴상수이고 W_K 와 x_K 는 각각 $N \times 1$ 인 무게 벡터와 $N \times 1$ 인 입력 벡터이다.

$$W_K = [W_{0,K}, W_{1,K}, W_{2,K}, \dots, W_{(n-1),K}]^T \quad (3)$$

$$x_K = [x_{K,0}, x_{K,1}, \dots, x_{K,N-1}]^T \quad (4)$$

그리고 E_K 는 K 번째 순간에서 실제 출력 Y_K 와 바라는 응답 d_K 간에 차로 주어지는 오차이다.

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

3. BLOCK LMS 적응 필터 [4]

$J = 0, 1, 2, \dots$ 를 block number라 하고 L 을 block length라 할때 블록 적응 필터에 대한 두개의 서로 다른 벡터가 있다. 그 하나는 시간 K 에서 필터의 레지스터에 저장된 입력 $N \times 1$ 인 벡터 x_K 이고 하나는 $L \times 1$ 인 벡터로 표시되는 J 번째 입력 데이터 block X_{JL} 이다.

$$x_K = [x_{K,0}, x_{K,1}, \dots, x_{K,N-1}]^T \quad (6)$$

$$X_{JL} = [x_{JL,0}, x_{JL,1}, \dots, x_{(J+1)L-1}]^T \quad (7)$$

마찬가지로 J 번째 출력 데이터 block Y_J 는 $L \times 1$ 벡터로 주어진다.

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

필터 무게는 블록을 기초로 하여 조절되며 J 번째 block에 대한 무게 W_J 는 $N \times 1$ 인 벡터로 표시된다.

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

여기서 첨자는 개개의 필터 무게를 나타낸다. 따라서 블록 적응 필터 출력은 다음식으로 표시된다.

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

이때 x_J 는 $L \times N$ 인 행렬이다.

$$x_J = \begin{bmatrix} x_{JL} & x_{JL-1} & \dots & x_{JL-N+1} \\ x_{JL+1} & x_{JL} & \dots & x_{JL-N+2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{(J+1)L-1} & x_{(J+1)L-2} & \dots & x_{(J+1)L-N} \end{bmatrix} \quad (11)$$

partition by columns

$$= [x_{JL} : x_{JL-1} : \dots : x_{JL-N+1}]^T \quad (12)$$

partition by rows

(9) 식과 (12)식을 (10)식에 대입하면 다음과 같이 표시된다.

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

(13)식은 linear convolution 처럼 J 번째 블록 출력 데이터인 Y_J 의 element를 구성한다. (13)식은 간단한 계산과정에 의해 다음과 같이 쓸수 있다.

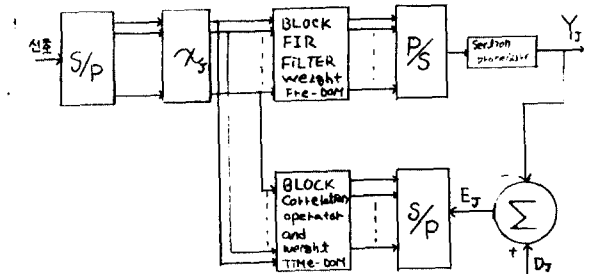
$$Y_{JL+K} = \sum_{i=0}^{N-1} x_{JL+K-i} W_{i,J} \quad K = 0, 1, \dots, L-1 \quad (14)$$

한편 필터무게는 다음식에 따라서 update된다.

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

여기서 ϕ_J 는 다음식으로 주어지며 element가 N 개인 gradient 추정 벡터이다. [4]

$$\phi_J = [\phi_{0J}, \phi_{1J}, \dots, \phi_{(N-1)J}]^T = x_J^T E_J \quad (16)$$



S/p = Serial to parallel, P/s = parallel to Serial converter
<그림 2> BLMS 적응 필터링

BLMS Adaptive Filtering

E_J 는 $L \times 1$ 인 오차 벡터로써 다음식과 같다.

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

그리고 $E_J = D_J - Y_J$ 이며 D_J 는 J 번째 바라는 응답이고 출력 형태로는 다음과 같이 표시한다.

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

(11)식과 (17)식을 (10)식에 대입하면 J 번째 블록 평균 자승 오차 gradient 추정 벡터의 element인 표현식을 얻는다.

$$\phi_{i,J} = x_{JL-i}^T \cdot E_J \quad i = 0, 1, \dots, N-1 \quad (19)$$

(19)식은 J 번째 오차 블록과 $(JL-i)$ 번째 입력 데이터, 블록과의 correlation을 나타낸다. 위의 식에서 처럼 블록 적응 알고리즘은 (10)식의 block convolution, (15)식의

gradient estimation, 그리고 (5)식의 무게 update의 연속적인 연산으로 이루어진다. 입력을 블록 단위로 처리할 경우 $L=N$ 인 경우가 가장 이상적이다.[4] 본 논문에서는 $L=N=4$, $L=N=2$ 인 경우로 하였다.

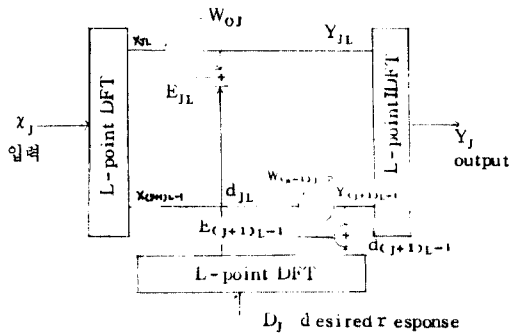
그림 2는 블록 적응 필터의 block diagram이다.

4. 주파수 영역에서 블록 적응 필터 [12][13]

4-1 필터 구조

주파수 영역 적응 디지털 필터는 L point 데이터 블록을 형성하기 위해 입력신호 x_j 와 바라는 응답 D_j 을 sectioning함으로써 그림 3에 나타낸것 처럼 구현된다. 이때 입력 신호와 바라는 응답은 L -point discrete fourier transform에 의해 주파수 영역으로 변환한다.

$X_{(j+1), L-1}$ 와 $d_{(j+1), L-1}$ 을 입력신호와 바라는 응답의 J 번째 블록 데이터의 $(J+1)L-1$ 번째 주파수 bin이라 한다. 여기서 x_j 는 $L \times N$ 행렬이며 D_j 는



<그림 3> 주파수 영역에서의 블록 적응 필터링

BLOCK Adaptive Filtering in Frequency domain

$L \times 1$ 인 벡터이다 그리고 각주파수 bin에 대응하는 L 개의 복소 무게가 있다. 그림 3에서 weighted output은 다음과 같다.

$$Y_j = W_j x_j \quad (20)$$

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

$$E_j = D_j - Y_j \quad (21)$$

여기서 E_j 는 $L \times 1$ 인 벡터이다. 만약 입력신호가 관측시간 동안 Wide-sense Stationary하다면 주파수 성분들은 orthogonal하고 L 개의 complex weights는 독립적으로 update 된다. 그러므로 주파수 영역 블록 적응 필터는 단일 무게가 L 개로 구성된 적응 필터로 생각할 수 있다.

4-2 고속 수렴 블록 적응 필터

본 소절에서는 주파수 영역 블록 적응 필터에 대한 고속 수렴 알고리즘을 제시한다. 그리고 무게 update 방정식은 다음과 같이 주어진다. [13]

$$W_{j+1} = W_j + P_j x_j^* E_j \quad (22)$$

$$\text{여기서 } P_j = [P_0^{-1} + x_j x_j^*]^{-1} \quad (23)$$

*는 complex conjugate를 나타내며 variable coefficient P_j 는 수렴비율을 제어하고 입력신호에 종속되기 때문에 무게 update식이 빠르게 수렴 할것으로 기대 된다. 그러므로 (22)식은 다음과 같이 다시 쓸수 있다.

$$P_j = [P_0^{-1} + \sum_{i=1}^J x_i x_i^*]^{-1} \quad (24)$$

(20)식과 (21)식을 (22)식에 대입하면 다음 식을 얻는다.

$$W_{j+1} = W_j + P_j x_j^* D_j - P_j W_j x_j^* x_j \quad (25)$$

(25)식의 양변에 (-1) 승을 취하면

$$x_j^* x_j = P_j^{-1} - P_j^{-1} \quad (26)$$

(26)식을 (25)식에 대입하면 다음과 같다.

$$W_{j+1} P_j^{-1} = W_j P_j^{-1} + x_j^* D_j \quad (27)$$

여기서 $W_0 = 0$ 이라면 (27)식은 다음과 같이 다시 쓸수 있다.

$$W_{j+1} = P_j \sum_{i=1}^J x_i^* D_i \quad (28)$$

따라서 (28)식을 (25)식에 대입하면 다음과 같은 식을 얻는다.

$$W_{j+1} = \frac{\sum_{i=1}^J x_i^* x_i}{P_0^{-1} + \sum_{i=1}^J x_i^* x_i} \quad (29)$$

J 가 충분히 크고 P_0^{-1} 이 작을 경우 (29)식의 분모는 입력 신호의 전력 스펙트럼이 되고 반면 분자는 입력과 바라는 응답의 cross power spectrum이 된다. 그러므로 $J = \infty$ 이고 $L=1$ 일 경우에 무게는 Wiener 전달 함수에 수렴한다. 실질적인 실현에 있어서 starting point information P_0 는 수렴 반응에 크게 영향을 미친다. 그러므로 (29)식은 다음과 같이 수정될 수 있다.

$$P_j = [\lambda P_0^{-1} + x_j x_j^*]^{-1} \quad (30)$$

여기서 λ 는 상수이다. ($0 < \lambda < 0.5$) 따라서 (29)식과 (30)식은 다음과 같이 다시 쓸수 있다.

$$P_J = [\lambda^J P_0 - 4 \sum_{i=1}^J \lambda^{J-i} x_i^* x_i]^{-1} \quad (31)$$

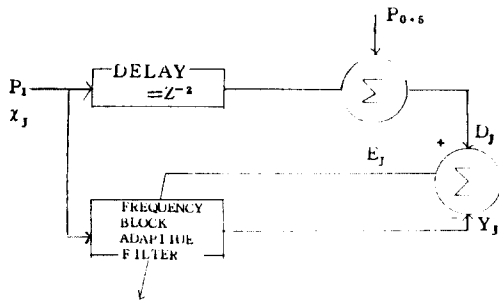
$$W_{J+1} = \frac{\sum_{i=1}^J \lambda^{J-i} x_i^* D_i}{\lambda^J P_0^{-1} + \sum_{i=1}^J \lambda^{J-i} x_i^* x_i} \quad (32)$$

여기서 J 가 클 경우 P_0 의 영향은 줄어들며 이러한 성질 때문에 이 알고리즘은 입력신호가 non-stationary 한 경우에도 추적할 수 있다.

5. Computer Simulation 결과 및 고찰

5-1 Simulation

이 절에서는 그림 4와 같은 방법으로 컴퓨터 시뮬레이션에 의해 실행된 SAITO, BLMS, FBAF (Frequency BLOCK Adaptive Filter) 알고리즘의 실행 결과를 비교 검토하였다. 알고리즘들의 수렴을 예상하기 위한 simulation 과정에서 4개의 무계를 이용하였다.

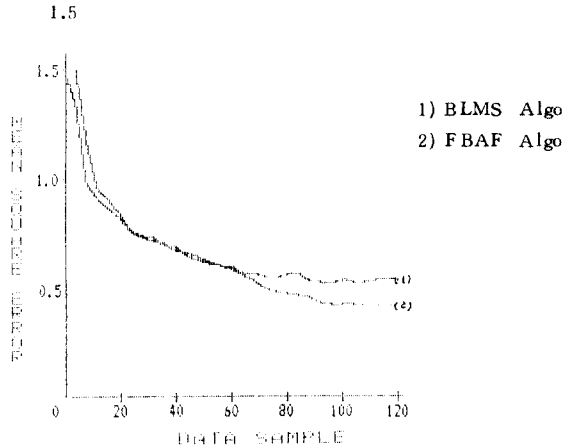


<그림 4> 시뮬레이션 과정

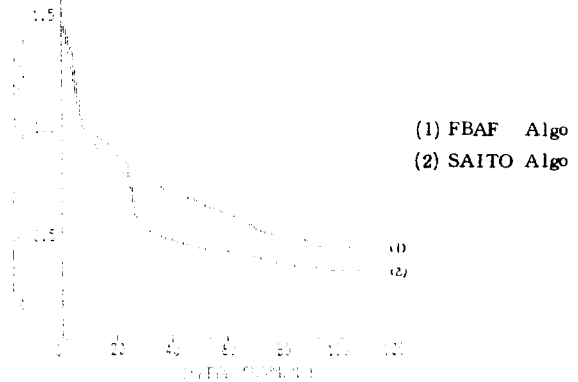
Simulation procedure

그림 4에서 평균이 zero이고 단위전력 (P_1) 백색 잡음의 독립표본 들로 구성된 신호들이 적응필터와 지연필터에 병렬로 인가된다. 지연필터의 출력은 전력이 $0.5(P_{0.6})$ 이고 평균이 zero 인 백색 독립 잡음이 더하여져 바라는 응답 신호로 이용된다. 여기서 평균이 zero이고 전력이 1인 백색 Gaussian 잡음의 발생은 0~1 사이의 random number 들 12번 더하여 6을 빼고 표준편차 1을 곱한 것에 zero 를 더하여 발생하였고 전력이 0.5인 백색 Gaussian 은 표준편차를 0.7로 하여 발생 시켰다. 그리고 이 두신호는 stationary이다. 적응 처리가 최적으로 수렴 되었을 경우 오차는 0.5에 수렴한다. 그리고 적응 필터의 초기 무계를 zero로 하였고 초기수렴을 고찰하기 위해 $P_J = 0, \lambda < 0.5$ 로 하여 실행한 알고리즘 들의 결과를 그림 5, 6, 7에 비교 제시하였다. 비교결과 BLMS알고리즘의 수렴상태보다 FBAF 알고리즘의 수렴상태가 빠르고 또 SAITO알고리즘의 수렴상태

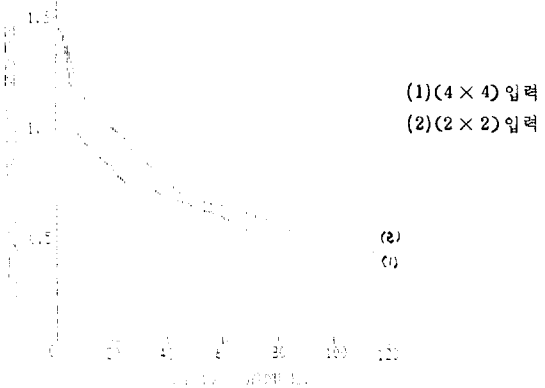
보다는 수렴 오차가 더 적음을 알수 있다. 그림 7은 FBAF 알고리즘의 입력 데이터를 $4 \times 4, 2 \times 2$ 로 한 경우로 불러 데이터를 4×4 로 한 경우가 훨씬 빠르게 수렴함을 알수 있다.



<그림 5> BLMS와 FBAF 알고리즘의 수렴상태
Convergence state of BLMS and FBAF Algorithm



<그림 6> FBAF와 SAITO 알고리즘의 수렴상태
Convergence state of FBAF and SAITO Algorithm



<그림 7> FBAF의 $(4 \times 4), (2 \times 2)$ 의 수렴상태
Convergence state of (4×4) and (2×2) for FBAF

5-2 FBAF 알고리즘의 계산량

이 소절에서는 FBAF 알고리즘의 계산량을 구하고 다른 알고리즘의 계산량과 비교한다. LMS 알고리즘은 $K_{LMS}=2N^2$ 만큼의 곱셈이 필요하고 이 알고리즘을 주파수 영역에서 실행할 때는 $K_{DEN}=(3M/2 \times \log_2 N + 2N)$ 만큼의 곱셈이 필요하다. [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] 단 $Q = \begin{cases} \log_2(N'/2) & N' \text{ even} \\ \log_2(N'/4) & N' \text{ odd} \end{cases}$, $b = \begin{cases} 0 & \text{even} \\ 1 & \text{odd} \end{cases}$

FBAF 알고리즘의 계산량은 $K_{FBAF}=(3N/2)\log_2 N+(N/2+1)+N(L+2)$ 이다 그리고 SAITO가 제시한 알고리즘의 계산량은 $K_F=(3N/2)\log_2 N+(6N/2)+4$ 이다. [13]

N	K_{LMS}/K_{FBAF}	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

표 1-1 계산량 비율
Table 1-1 Computation rate

6. 결론

본 논문에서는 주파수 영역에서 실행된 블럭 적응 필터에 대한 빠른 수렴 알고리즘이 제시되었다.

제시된 알고리즘은 G.A CLACK의 알고리즘에 Gordard 이론을 전개한 결과가 되며 또 SAITO 알고리즘을 블럭화한 것이다. 이러한 결과 BLMS 알고리즘의 수렴상태 보다는 빠르게 수렴함을 알수 있었고 SATIO 알고리즘 보다는 수렴오차를 줄일수 있었다. 그러나 필터의 length N이 32보다 클 경우에는 BLMS, SAITO, DENTINO 알고리즘의 계산량 보다는 FBAF 알고리즘의 계산량이 증가함을 알수있다.

REFERENCES

1. M.J. dentino, J. McCool, and B. Widrow, "Adaptive filtering in the frequency domain, Proc. IEEE. vol. 66, pp. 1658-1659, Dec. 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 Adaptive Signal Processing, vol. ASSP-29, pp. 744-752, June 1981.
5. A.V. Oppenheim and R.W. Schaffer, Digital Signal Processing. Englewood, 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 cancelling," Principles and application, Proc. IEEE, vol. 63, Dec. 1975.
8. BERNARD 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 Vol. 63, pp. 719-720, Apr, 1975.
10. S. Shanker Narayan and A.M. Peterson, "Frequency domain least-mean-square algorithm". Proc. IEEE Vol. 69 - 1981.
11. G.A. CLACK, "A Unified Approach to time-and-Frequency domain Realilation of Fir Adaptive Digital filter", IEEE Trans on Assp, vol. ASSP. 31 No. 5, OCT. 1983.
12. D. Gordard "Channel Equalilition Using Kalman filter for fast data Transmission", IBM Journal of Reserch and Development, pp. 267-273, May 1974.
13. TSUNEO SAITO AND YUKTO HO Shiko "FAST Convergence Frequency Domain Adaptive Filter", The Technology Reports of the Tohoku University vol. 48 (1983) No. 2 Dec. pp. 181-197.
14. IEEE TRANSACTION ON COM, VOL. COM. 26 No. 10 OCT 1978" Application of Fast Kalman Estimation to Adaptive Equalization" DAVID D. FALCONER AND LENN RT.
15. IEEE. TRANS ON COM. Vol. COM-25 No. 7 JULY 1977 666-672. "Self-Orthogonalizing Adaptive Equalilition algorithm" RICHAD D. GITLIN, FRANCIS R. MAGEE, JR.