

적용 신호 처리에의 응용을 위한 고속 QR RLS 알고리즘의 연구

정 지영, 변 문식
인천 대학교 전자공학과

A Study on the Fast QR RLS Algorithm for Applications to Adaptive Signal Processing

Jee yeong Jeong, Youn-Shik Byun

Dept. of Electronic Engineering, InChon University

ABSTRACT

RLS algorithms are required for applications to adaptive line enhancers, adaptive equalizers for voiceband telephone and HF modems, and wide-band digital spectrum mobile radio in which their convergence time and tracking speed are significant.

The fast QR RLS algorithm satisfies above the requirements. Its computational complexity is linearly proportional to the tap number of a filter, N and its performance remains numerically stable.

From the result of simulation, the fast QR RLS algorithm represented Cioffi is better than gradient based algorithm in its initial performance when being applied to an adaptive line enhancer for cancelling noise.

[.서론

적용필터의 통신 및 산업분야에 대한 응용이 VLSI 기술의 급격한 발전에 따라 보편화 되었으며 이에 여러가지 적용필터 알고리즘이 연구되고 있다. LMS 알고리즘은 Wiener 필터의 파라미터를 근사치로 대체하여 유도되는 알고리즘으로서 계산이 단순하여 자주 사용되고 있다. 그러나 음성 대역 모델, 광대역 디지털 이동 무선 채널, HF 모델과 적응 잡음 제거기와 같이 알고리즘의 수렴속도와 추적능력이 중요한 요구조건이 되는 응용에서는 LMS 계열의 알고리즘보다 RLS 알고리즘의 필요성이 커진다. $O(N^2)$ 의 계산량을 갖는 RLS 알고리즘의 실시간 처리를 위하여 적용필터의 차수 N에 선형적으로 비례하는 고속 알고리즘에 관한 연구가 시작되었으며 그 결과 일반 RLS 알고리즘의 계산량을 줄여 계산량이 $O(N)$ 정도의 고속 RLS 알고리즘이 발표되었다[4][5][6]. 그러나 이상의 고속 알고리즘은 수치적으로 불안정한 수렴을 보이는 경향이 있다. 특정 조건의 신호를 처리할때, 혹은 짧은 어장이나 부동점 산술을 채택 했을 때는 수치적 수렴결과가 문제가 된다. 따라서 이러한 문제를 해결하기 위해 수치적 특성이 안정한 QR-RLS 알고리즘이 연구되었다. QR 본체의 직교변환으로 채택하여 LS 알고리즘의 문제를 회귀적으로 풀 수 있는

$O(N^3)$ 의 QR-RLS 알고리즘이 도입 되었다[7][8][9]. Cioffi는 [10]에서 FFT에서와 같이 기하학적인 개념을 적용하여 Prouder는 [11]에서 격자구조를 갖는 필터에 QR 본체를 적용하여, Ling은 [12]에서 변형된 Gram-Schmidt 직교화를 이용하여 각각 $O(20N) \sim O(30N)$ 사이의 계산적 복잡도를 지닌 고속 QR-RLS 알고리즘을 제시하였다. 본 연구는 Cioffi에 의해 제안된 고속 QR 알고리즘을 대수적으로 간단히 유도하고 컴퓨터 모의 실험을 통하여 고속 QR-RLS 알고리즘을 선형 등화기에 응용함으로써 기존의 LMS 알고리즘을 사용한 적응 선형 등화기와의 특성을 비교하였다.

II. 일반적인 QR-RLS 알고리즘

일반적인 N차 RLS 알고리즘은 비용함수 $\xi(W_N)$ 을 최소화 하는 필터의 탭계수 W_N 을 회귀적으로 추정한다.

$$\xi(W_N) = \langle e^N(n), e(n) \rangle \quad (1)$$

$$e(n) = d(n) - X_N^T W_N \quad (2)$$

단, $d(n) = [d(n) \quad X_N(n) = [x(n) \ x(n-1) \ \dots \ x(n-N+1)]$

$$\begin{bmatrix} \lambda^{1/2} d(n) \\ \lambda^{n/2} d(n) \end{bmatrix} \quad \begin{bmatrix} \lambda^{1/2} x(n-1) x(n-2) \dots x(n-N) \\ \lambda^{n/2} x(0) \dots 0 \end{bmatrix}$$

$$W_N = [w_0 \ w_1 \ \dots \ w_{N-1}]$$

$d(n)$ 은 시간 $0 \dots n$ 까지의 필터의 원하는 응답으로 구성되고 $X_N(n)$ 은 시간 $0 \dots n$ 까지의 필터의 입력이 되는 열 벡터로 구성된다. W_N 은 N차 RLS 필터의 탭 계수로서 열 벡터로 나타낸다.

QR 본체의 직교행렬로서 Givens 변환을 선택하며 2×2 Givens 변환을 연속적으로 사용하여 $X_N(n)$ 를 다음과 같이 변환할 수 있다.

$$Q_N(n) X_N(n) = \begin{bmatrix} \text{NULL} \\ R_N \end{bmatrix}$$

이러한 직교행렬 $Q_N(n)$ 을 (2)식의 양변의 앞에 곱하면

$$\underline{\varepsilon}(n) = \underline{D}(n) - \begin{bmatrix} \text{NULL} \\ R_N \end{bmatrix} W_N^H \quad (3)$$

이 되며 $\underline{\varepsilon}(n)$ 은 $\underline{e}(n)$ 과 유클리드 노름이 같다.

$$\underline{\varepsilon}^H(n)\underline{\varepsilon}(n) = \underline{\varepsilon}^H(n)Q_N^H(n)Q_N(n)\underline{e}(n) = \underline{e}^H(n)\underline{e}(n)$$

따라서 $(n+1)$ 개의 성분을 갖는 벡터를 편의상

$$\underline{\varepsilon}(n) = \begin{bmatrix} \underline{e}^-(n) \\ \underline{e}^+(n) \end{bmatrix} \quad \underline{D}_N(n) = \begin{bmatrix} \underline{d}^-(n) \\ \underline{d}^+(n) \end{bmatrix} \quad (4)$$

과 같이 분할하여 표시한다. 여기서 $-$ 는 N 개의 성분, $+$ 는 $(n-N+1)$ 개의 성분을 갖는다. 따라서

$$\underline{d}^-(n) = R_N W_N^H \quad (5)$$

을 복귀 대입법으로 풀어 간단히 W_N 를 구할 수 있다. $Q_N(n)$

은 $X_N(n)$ 의 오른쪽 행렬로부터 시작하여 R_N 의 반대각선상에 데이터를 압축하기 때문에 $Q_N(n)$ 의 시간 n 에 해당하는 N 회의 2×2 Givens 행렬 $Q^*(n)$ 을 분리하여 쓰면

$$Q_N(n) = Q^*(n) \begin{bmatrix} 1 & 0 \\ 0 & Q_N(n-1) \end{bmatrix} \quad (6)$$

와 같이 시간 경신식으로 나타낼 수 있다. 사후 추정오차 $\underline{e}(n)$ 는 다음과 같이 피닝(planning) 벡터 $\sigma(n)$ 을 사용하여 계산할 수 있다.

$$\underline{e}(n) = \underline{e}^-(n)\sigma(n) = \underline{e}^-(n)\Sigma_N(n) \quad (7)$$

단,

$$\underline{e}^-(n) = [\underline{\varepsilon}^-(n) \ 0 \ \dots \ 0], \quad \Sigma_N(n) = [\lambda^{1/2} \sigma(n) \ 0 \ \dots \ 0 \ \bar{\Sigma}_N(n)].$$

$$\Sigma_N(n) = Q_N(n)\sigma(n) = Q^*(n)\sigma(n)$$

이다. 따라서 $\underline{e}(n)$ 은

$$\underline{e}(n) = \underline{e}^-(n) \begin{bmatrix} \lambda^{1/2} \sigma(n) \\ 0 \end{bmatrix} \quad (8)$$

로서 나타낼 수 있다. 단,

$$\lambda^{1/2} \sigma(n) = \prod_{i=1}^N \cos \theta_i$$

(3)식으로부터 $\underline{e}^-(n)$ 의 최상단 성분 $\underline{e}^-(n)$ 은 (4)식에 의하여 $\underline{d}^-(n)$ 의 최상단 성분과 같음을 알 수 있다. $\underline{e}^-(n)$ 의 최상단 성분 $\underline{e}^-(n)$ 은 $\underline{d}^-(n)$ 의 최상단 성분과 같으며

$$Q^*(n) \begin{bmatrix} \underline{d}^-(n) \\ \lambda^{1/2} \sigma^-(n-1) \\ \lambda^{1/2} \sigma^-(n-1) \end{bmatrix} \quad (9)$$

에 의하여 최귀적으로 계산된다.

표. 고속 QR RLS 알고리즘

(1) 전향 예측 필터

전향 예측 필터의 원하는 응답 벡터는 $X(n)$ 이고 입력은 $X_N(n-1)$ 이다. $X(n)$ 과 $X_N(n-1)$ 에 의하여 확장된 데이터 행렬 $X_{N+1}(n)$ 을

$$X_{N+1}(n) = \begin{bmatrix} X(n) & X_N(n-1) \\ & 0 \end{bmatrix} \quad (10)$$

으로 정의하고 $\begin{bmatrix} Q_N(n-1) & 0 \\ 0 & 1 \end{bmatrix}$ 로 곱하면

$$\begin{bmatrix} Q_N(n-1) & 0 \\ 0 & 1 \end{bmatrix} X_{N+1}(n) = \begin{bmatrix} \phi^-(n) & \text{NULL} \\ \phi^-(n) & R_N(n-1) \\ \lambda^{1/2} \sigma(n); & \text{NULL} \end{bmatrix} \quad (11)$$

이 된다. 여기서 $\phi^-(n)$ 과 $\phi^+(n)$ 은 $\begin{bmatrix} Q_N(n-1) & 0 \\ 0 & 1 \end{bmatrix} x(n)$ 을

(4)식의 정의에 따라 편의상 분할하기 위해 사용된다.

$X_{N+1}(n)$ 을 $(N+1) \times (N+1)$ 삼각행렬 R_{N+1} 로 변환하기 위하여 $\phi^-(n)$ 의 모든 성분을 $\lambda^{n/2} \sigma(n)$ 의 위치로 압축하는 변환 $Q^*(n)$ 을 도입한다.

$$Q_{N+1}(n) X_{N+1}(n) = Q^*(n) \begin{bmatrix} Q_N(n-1) & 0 \\ 0 & 1 \end{bmatrix} X_{N+1}(n) = \begin{bmatrix} \text{NULL} & \text{NULL} \\ \phi^-(n) & R_N \\ \lambda^{1/2} \sigma(n) & \text{NULL} \end{bmatrix} = \begin{bmatrix} \text{NULL} \\ R_{N+1}(n) \end{bmatrix} \quad (12)$$

$\phi^-(n)$ 의 각 성분을 하단 성분부터 차례로 압축하는 $Q^*(n)$ 에서 시간 n 에서의 2×2 Givens 행렬을 분리하여 쓰면 $Q^*(n)$ 의 시간 경신식을 얻을 수 있다.

$$Q^*(n) = Q^*(n) \begin{bmatrix} 1 & 0 \\ 0 & Q^*(n-1) \end{bmatrix} \quad (13)$$

그리고 $Q_{N+1}(n)$ 의 경신식은 다음과 같다.

$$Q_{N+1}(n) = Q^*(n) Q_N(n) = Q^*(n) \begin{bmatrix} Q^*(n-1) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & Q^*(n-1) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & Q_N(n-2) & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (14)$$

$Q^*(n)$ 은 다음식에 의하여 결정된다.

$$\alpha(n) = \lambda \alpha_N(n-1) + [\phi^-(n)]^2 \quad (15-1)$$

$$\cos \theta^* = \frac{\lambda^{1/2} \alpha^{1/2}(n-1)}{\alpha^{1/2}(n)} \quad (15-2)$$

$$\sin \theta^* = \frac{\phi^-(n)}{\alpha^{1/2}(n)} \quad (15-3)$$

단, $\phi^-(n)$ 은 $\phi^-(n)$ 의 최상단 성분을 나타낸다.

(1) 후향 예측 필터

시간 n 에서의 후향 예측 필터의 원하는 응답은 $x(n-N)$ 이고 원하는 입력은 $X_N(n)$ 이다. $X_{N+1}(n)$ 에 $Q_N(n)$ 를 곱하면

$$W_N(n) X_{N+1}(n) = \begin{bmatrix} \text{NULL} & \psi^-(n) \\ R_N(n) & \psi^-(n) \end{bmatrix} \quad (16)$$

이 된다. $X_{N+1}(n)$ 을 전향 예측의 $R_{N+1}(n)$ 으로 변환하기 위하여 $\psi^-(n)$ 의 모든 성분을 하단 원소부터 차례로 $(N+1)$ 번째 열에, $\psi^-(n)$ 의 모든 성분을 역시 하단 원소부터 차례로 $(N+1)$ 번째 열에 압축하는 변환 $Q^*(n)$ 과 $Q^*(n)$ 을 도입한다. 따라서 $Q_{N+1}(n)$ 은

$$Q_{N+1}(n) = Q^*(n) Q^*(n) Q^*(n) Q^*(n) \quad (17)$$

이 된다. $Q^*(n)$ 의 시간 경신식은

$$Q^*(n) = Q^*(n) \begin{bmatrix} 1 & 0 \\ 0 & Q^*(n-1) \end{bmatrix} \quad (18)$$

으로 나타낼 수 있다. (18)식으로부터

$$Q_{N+1}(n) \begin{bmatrix} 1 & 0 \\ 0 & Q^*(n-1) \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & Q^*(n-1) \end{bmatrix} = Q^*(n) Q^*(n) Q^*(n) Q^*(n) \quad (19)$$

를 유도하여 (N+1)번째 성분만이 V이고 나머지 성분은 모두 0인 v(n)을 뒤에 곱한다. (17)의 좌변은 다음과 같다.

$$\begin{bmatrix} 1 & 0 \\ 0 & Q^b_{N(n-1)} \end{bmatrix} v(n) = \begin{bmatrix} 0 \\ v^d(n) \\ 1 \\ \text{NULL} \end{bmatrix} \quad (20)$$

여기서 v(n)의 V는 (20)식의 (N+1)번째 위치 원소가 1이 되도록 선택되었다고 가정한다.

$$Q^{w+1}(n) \text{과 } \begin{bmatrix} 1 & 0 \\ 0 & Q^w_{N(n-1)} \end{bmatrix} \text{은 하단 (N+1)개의 성분과 최}$$

상단 (n+1)번째 성분에만 작용하므로 (20)식의 좌변을 다음과 같이 필요한 부분만 선택하여 볼 수 있다.

$$Q^{w+1}(n) \begin{bmatrix} 1 & 0 \\ 0 & Q^w_{N(n-1)} \end{bmatrix} \begin{bmatrix} 0 \\ v^d(n) \\ 1 \\ \text{NULL} \end{bmatrix} \quad (21)$$

(20)식의 우변은

$$Q^w_{N(n)} Q^b_{N(n)} \begin{bmatrix} 0 \\ v^d(n) \\ 1 \\ \text{NULL} \end{bmatrix} \quad (22)$$

이 되고 $Q^b_{N(n)}$ 은 (N+1)번째 성분과 (n+1)번째 성분에 작용하는 1회의 Givens 변환이고 $Q^w_{N(n)}$ 은 하단 (N+1)개의 성분에만 작용하므로 (22)식의 (n+1)번째 성분과 하단 (N+1)개의 성분만을 써서 나타내면 다음과 같다.

$$Q^w_{N(n)} \begin{bmatrix} U_2 \\ U_1 \\ \text{NULL} \end{bmatrix} \quad (23)$$

따라서 (20)식을 다시 쓰면

$$Q^{w+1}(n) \begin{bmatrix} 1 & 0 \\ 0 & Q^w_{N(n-1)} \end{bmatrix} \begin{bmatrix} 0 \\ v^d(n) \\ 1 \\ \text{NULL} \end{bmatrix} = Q^w_{N(n)} \begin{bmatrix} U_2 \\ U_1 \\ \text{NULL} \end{bmatrix} \quad (24)$$

$$Q^w_{N(n)} Q^{w+1}(n) \begin{bmatrix} 1 & 0 \\ 0 & Q^w_{N(n-1)} \end{bmatrix} \begin{bmatrix} 0 \\ v^d(n) \\ 1 \\ \text{NULL} \end{bmatrix} = \begin{bmatrix} U_2 \\ U_1 \\ \text{NULL} \end{bmatrix} \quad (25)$$

이 된다. $Q^{w+1}(n)$ 은 (14)식의 갱신식을 사용하고 (25)식에 대하여 풀면 $Q^w_{N(n)}$ 의 N개의 회전각을 구할 수 있다.

(3) 피닝(Pinning) 추정 필터

마지막으로 구하고자 하는 $Q^w_{N(n)}$ 은 (20)식의 양변에 피닝(Pinning) 벡터 $\sigma(n)$ 을 뒤에 곱하여 구할 수 있다.

단, $\sigma(n)=[100, \dots, 0]$ 이며 $\hat{y}(n)\sigma(n)=\hat{\Sigma}_N(n)$ 으로 정의한다.

$$Q^{w+1}(n)\sigma(n)=Q^w_{N(n)}Q^b_{N(n)}Q^w_{N(n)}\sigma(n) \quad (26)$$

$$\begin{bmatrix} \hat{y}^{1/2_{N+1}(n)} \\ \hat{\Sigma}_N(n) \end{bmatrix} = Q^w_{N(n)}Q^b_{N(n)} \begin{bmatrix} \hat{y}^{1/2_N(n)} \\ \hat{\Sigma}_N(n) \end{bmatrix} \quad (27)$$

$Q^b_{N(n)}$ 은 하단 N개의 성분에는 작용하지 않으므로 (27)식의 하단 N개의 성분만을 선택하여 쓰면

$$\begin{bmatrix} v_0 \\ \hat{\Sigma}_N(n) \end{bmatrix} = Q^w_{N(n)} \hat{\Sigma}_{N+1}(n) \quad (28)$$

이 된다. (28)식으로부터 $Q^w_{N(n)}$ 의 N개의 회전각을 계산할

수 있다.

(4) 결합 처리 필드

이제 $Q^w_{N(n)}$ 을 알고 있기 때문에 고속 QR RLS 알고리즘의 결합 처리 추정은 (8)식과 (9)식을 그대로 사용할 수 있다.

N. 모의 실험 및 결론

고속 QR RLS 알고리즘을 잡음 제거기에 채택하여 LMS 알고리즘의 경우와 수행 결과를 비교하였다. LMS 알고리즘이 안정 상태에 이르는데 약 50회 정도의 반복을 요하나 고속 QR RLS 알고리즘의 경우, 20회 정도의 반복 수행 후 안정 상태에 이르는 것을 알 수 있다. 안정 상태에서의 수행은 QR RLS와 LMS는 거의 비슷한 결과를 보임을 알 수 있다.

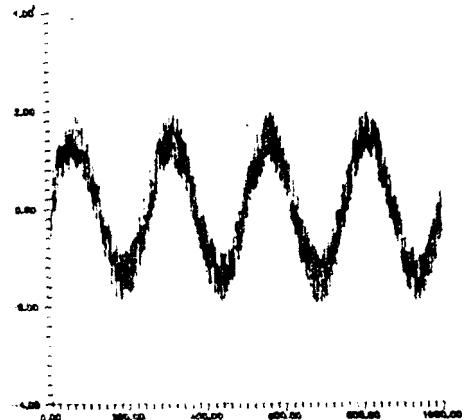


Fig. 1 Signal with white noise (SNR = 10dB)

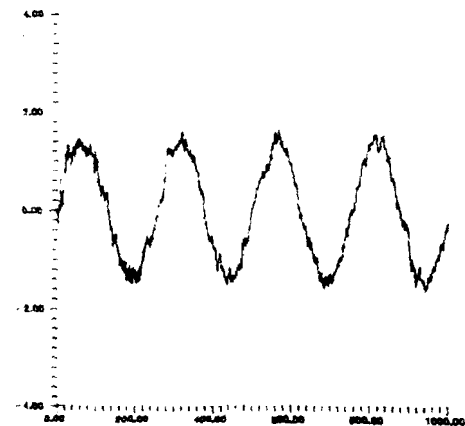


Fig. 2 Line-enhanced signal filtered by the fast QR RLS algorithm

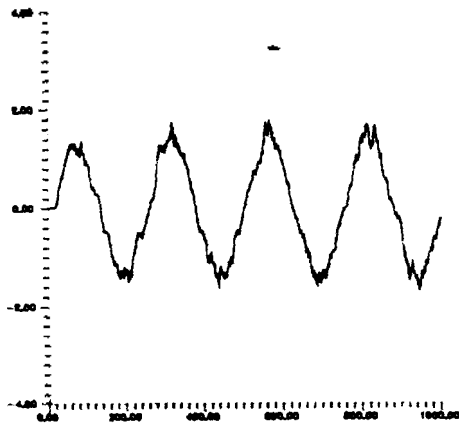


Fig. 3 Line-enhanced signal filtered by the LMS algorithm

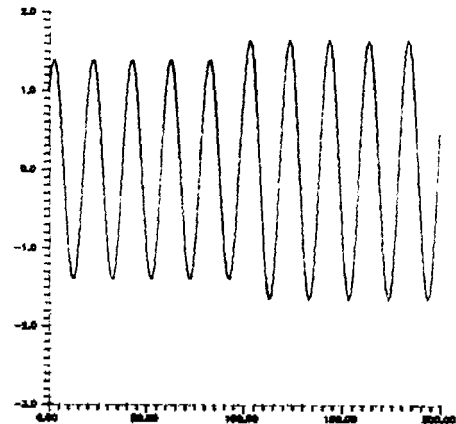


Fig. 4 Information signal without a additive noise

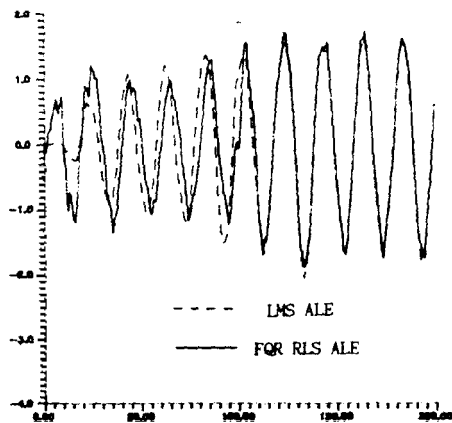


Fig. 5 LMS performance and fast QR RLS performance in the adaptive line enhancer.

V. 참고 문헌

- [1] S. Haykin, "Adaptive Filter Theory", Prentice-Hall, 1986.
- [2] G. Golub and C. Van Loan, "Matrix Computatins", John Hopkins Press, 1983.
- [3] 변 윤식, "적응신호처리에서의 응용을 위한 고속 적응 필터 알고리즘에 관한 연구", 인천대학교 공과대학 전자공학과, 1989.
- [4] J. M. Cioffi and T. Kailath, "Fast, RLS, Transversal Filters for Adaptive Filtering", IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-29, pp.304-337, Apr. 1984
- [5] G. Carayannis, D. Manolakis, and Kalouptsidis, "Fast Sequential Algorithm for least squares Filtering and prediction", IEEE Trans. Acoust. ., Speech, Signal Processing, vol. ASSP-31, pp.1392-1402, Dec. 1983
- [6] D. D. Falconer and L. Ljung, "Application of Fast Kalman Estimation to Adaptive Equalization", IEEE Trans. Commun., vol. COM-26, pp. 1439-1446, Oct. 1978
- [7] Kung, H. T., and C. E. Leiserson, "Systolic Arrays (for VLSI)", Sparse Matrix Proc. 1976, Soc. Ind. Appl. Math., 1978, pp. 256-282
- [8] W. M. Gentleman and H. T. Kung, "Matrix Triangularization by Systolic Arrays", Proc. SPIE, 298, Real-Time-Signal-Processing IV, pp. 19-26, Aug. 1981
- [9] Shreiber, R. J. and P. J. Kuekes, "Systolic Linear Algebra Machines in Digital Signal Processing", in VLSI and Modern Signal Processing, ed. S. Y. Kung, H. J. Whitehouse, and T. Kailath, Prentice-Hall, Englewood Cliffs, N. J., pp. 389-405
- [10] J. M. Cioffi, "The Fast Adaptive ROTOR's RLS Algorithm", Trans. Acout., Speech, and Signal Processing, vol. 38, NO. 4, April, 1990
- [11] Prouder, I. K., J. G. McWhirter, and T. J. Shepherd, "QRD-Based Lattice Filter Algorithms, SPIE, San Diego, Calif, 1989
- [12] Ling, F., "Efficient Least-Squares Lattice Algorithms Based on Givens Rotation with Systolic Array Implementations", Proc. ICASSP, Glasgow, Scotland, pp. 1290-1293, 1989