

QR 분해법을 이용한 적응 쌍선형 격자 알고리즘 (QR-Decomposition based Adaptive Bilinear Lattice Algorithms)

安奉萬*, 黃智源*, 白興基*

(Bong-Man Ahn, Jee-Won Hwang and Heung-Ki Baik)

要約

본 논문은 QR 분해법을 이용한 쌍선형 격자 필터의 최소자승 알고리즘에 관한 논문이다. 쌍선형 최소자승 격자 알고리즘에 QR 분해법을 적용한 쌍선형 최소자승 격자 알고리즘을 제안하였다. 제안한 알고리즘을 식오차 방법과 출력오차 방법으로 쌍선형 시스템 식별 문제에 적용하여 제안한 알고리즘의 성능을 평가하였다. 컴퓨터 시뮬레이션 결과 본 논문에서 제안한 알고리즘의 수렴특성은 측정 잡음이 있는 경우 Baik이 제안한 쌍선형 순환 최소자승 격자 알고리즘의 수렴특성과 비슷하였으며, 측정 잡음이 없는 경우에는 더 우수함을 알 수 있었다.

Abstract

This paper presents new QRD-based recursive least squares algorithms for bilinear lattice filter. Bilinear recursive least squares lattice algorithms are derived by using the QR decomposition for minimization covariance matrix of prediction error by applying Givens rotation to the bilinear recursive least squares lattice algorithms. The proposed algorithms are applied to the bilinear system identification to evaluate the performance of algorithms. Computer simulations show that the convergence properties of the proposed algorithms are superior to that of the algorithms proposed by Baik when signal includes the measurement noise.

1. 서론

순환 최소자승(RLS) 알고리즘은 최소 평균 자승(LMS) 알고리즘에 비해 우수한 수렴특성과 궤적 능력 때문에 적응 신호처리, beamforming, 어레이

신호처리, 채널 등화기등 신호처리 분야와 통신 분야에서 널리 사용되고 있다그러나 RLS 알고리즘은 수렴속도가 빠른 대신 수치적 특성(numerical property)이 나쁘다는 단점이 있다. 이러한 단점을 극복하기 위한 방법으로 격자필터를 이용하는 방법과 QR 분해법을 이용한 방법이 있다.

QR 분해법(QR-decomposition)에 근거한 최소자승 알고리즘은 시스템릭 어레이(systolic array)를 사용하여 병렬처리에 적합한 구조를 얻을 수 있을 뿐만

*正會員, 全北大學校 工科大學 電子工學科
(Dept. of Elec. Eng., Chonbuk Nat'l Univ)
接受日字 : 1994年 3月 8日

아니라 우수한 수치적안정도 때문에 순환 최소 자승을 구현하는 많은 방법중에 하나의 적합한 방법으로 알려져 있다.^{[11][2][19]} 일반적으로 QR 분해법을 이용하여 순환 최소 자승 문제의 해를 구하는 방법은 먼저 직교행렬을 이용하여 데이터 행렬을 삼각행렬 모양으로 변환하고, 다음에 원하는 계수를 삼각행렬로부터 후진대입법(back-substitution)으로 계산한다.^{[11][2][19]} 이 방법은 계수를 순환적으로 구할 수 있고, 유한어장 효과(finite wordlength effect)에 강인한 회전 연산자의 집합을 이용하여 삼각행렬을 구하므로 계산의 효율성이 높고 수치적 안정도가 좋다는 장점이 있다.

적응 신호처리 분야에서 QR 분해를 순환적으로 구하기 위해서 병렬처리 구조에 적합한 Givens 회전법으로 주로 사용하고 있으나, 수정된 Gram-Schmidt, Householder 변환법을 이용한 논문들도 있다.^{[17][18]} Gentleman과 Kung^[15]은 알고리즘 구현에 삼각형부분과 선형부분으로 구성되는 시스템릭어레이를 제안하였고, McWhiter^[16]는 QRD RLS 알고리즘의 구현에 선형부분을 포함하지않는 간단한 구조의 시스템릭 어레이를 제안하였다.

한편 선형필터로 처리할 수 없는 분야가 생김에 따라 비선형 필터에 대한 연구도 활발히 진행되어 왔다.^[4-12] 대표적인 비선형 필터로 Volterra 필터와 쌍선형 필터가 있다. Volterra 필터는 많은 계수를 포함하고 있고 이에 따라 많은 계산량을 필요로 한다. 그러나 쌍선형 필터는 출력을 제한하기 때문에 필터의 안정성이 문제가 되나 Volterra 필터에 비해 적은 수의 계수를 필요로 하는 장점이 있다. 참고문헌 [26-28]에서 쌍선형 필터의 안정도에 관하여 연구되어졌다. Volterra 필터나 쌍선형 필터의 RLS 알고리즘 역시 수치적 특성이 나쁘기 때문에 수치적 특성이 좋은 격자 알고리즘과 QR 분해법을 이용한 비선형 필터에 관한 연구가 최근에 진행되어 왔다. 2차 Volterra 필터의 격자 알고리즘이 Mathews^[3]에 의해, 쌍선형 필터의 격자구조와 격자 알고리즘이 Baik^[13-14]에 의해 연구되었으며, Mathews와 Syed^[23-24]에 의해 2차 Volterra 필터의 QRD-RLS 알고리즘이 연구되어졌으나 쌍선형 필터의 QRD-RLS 알고리즘에 관한 연구는 아직 보고된 바 없다.

다채널 격자 필터의 RLS 알고리즘으로 Ling 과 Proakis^[20]가 Gram-Schmidt 직교화 과정을 이용한 다채널 최소 자승 격자 알고리즘을 제안하였으며, Lewis^[20]는 기존의 다채널 격자 알고리즘을 Givens 회전법과 대수학적으로 등가인 Cholesky 분해를 이용하여 QR-MLSL 알고리즘을 제안하였다이 방법은 기존의 필터식을 그대로 적용할 수가 있어 QR 분해법

을 이용한 알고리즘으로 확장하기 쉬운 장점이 있다.

본 논문에서는 Baik^[13-14]이 제안한 쌍선형 필터의 순환 최소 자승 격자 알고리즘에 참고문헌 [20-24]의 방법과 비슷한 방법의 Givens 회전법을 적용하여 QR 분해법을 이용한 쌍선형 적응 격자 알고리즘을 유도하였다. 또 제안한 알고리즘을 식오차 방법과 출력오차 방법으로 쌍선형 시스템의 식별 문제에 적용하여 제안한 알고리즘의 성능을 평가하였다. 컴퓨터 시뮬레이션 결과 본 논문에서 제안한 알고리즘의 수렴특성은 측정 잡음이 있는 경우 Baik^[13-14]이 제안한 순환 최소 자승 격자 알고리즘(이하 RLSL 이라함.)의 수렴특성과 비슷하였으며, 측정잡음이 없는 경우에서는 더 우수함을 알 수 있었다.

본 논문은 1장에서 RLSL 알고리즘에 대하여 간단히 다루었고, 2장에서는 QR 분해법을 이용한 쌍선형 적응 격자 알고리즘(이하 QRD-LSL 이라함.)에 대하여 다루었으며, 3장에서는 컴퓨터 시뮬레이션 결과를 설명하였다.

마지막으로 4장에서는 결론과 앞으로의 연구 방향에 대하여 서술하였다.

II. 적응 쌍선형 격자 알고리즘

시간 n일 때 적응필터의 계수에 의한 쌍선형 모델의 출력 y(k)의 추정치를 $\hat{y}_n(k)$ 라하면 $\hat{y}_n(k)$ 는 다음과 같다.

$$\hat{y}_n(k) = \sum_{i=0}^{N-1} a_i(n)x(k-1) + \sum_{i=0}^{N-1} \sum_{j=1}^{N-1} c_{ij}(n)x(k-i)y(k-j) + \sum_{j=1}^{N-1} b_j(n)y(k-j) = W^T(n)U(k) \quad (1)$$

여기서

$$U(k) = [x(k), \dots, x(k-N+1), x(k)y(k-1), \dots, x(k-N+1)y(k-N+1), y(k-1), \dots, y(k-N+1)]^T \quad (2)$$

$$W(n) = [a_0(n), \dots, a_{N-1}(n), c_{0,1}(n), \dots, c_{N-1,N-1}(n), b_1(n), \dots, b_{N-1}(n)]^T \quad (3)$$

이고, $(\cdot)^T$ 는 전치행렬을 나타낸다. 식(2.1)에서 W(n)은 다음과 같은 성능지수 함수를 최소화하여 얻을 수 있다.

$$\xi_N(n) = \sum_{k=1}^n \lambda^{n-k} [y(k) - \hat{y}_n(k)]^2 \quad (4)$$

여기서 λ 는 망각요소(forgetting factor)로 $0 < \lambda \leq 1$ 이고, $\hat{y}_n(k)$ 는 다음과 같이 두 가지로 표현될 수 있다.

첫 번째 방법으로 식 (1)에서 지연된 출력의 형태가 다음과 같이 원하는 신호(desired signal)의 지연된 형태로 나타나는 식오차 방법이 있다.

$$\begin{aligned} \hat{y}_n(k) = & \sum_{i=0}^{N-1} a_i(n)x(k-i) \\ & + \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} c_{ij}(n)x(k-i)d(k-j) \\ & + \sum_{j=1}^{N-1} b_j(n)d(k-j) \end{aligned} \quad (5)$$

두 번째 방법으로 예측된 출력을 지연시켜 그대로 사용하는 방법으로서 출력오차 방법이 있다.

$$\begin{aligned} \hat{y}_n(k) = & \sum_{i=0}^{N-1} a_i(n)x(k-i) \\ & + \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} c_{ij}(n)x(k-i)\hat{y}_{n-j}(k-j) \\ & + \sum_{j=1}^{N-1} b_j(n)\hat{y}_{n-j}(k-j) \end{aligned} \quad (6)$$

식오차 방법은 $W(n)$ 에 바이어스되는 단점이 있고, 출력오차 방법의 결과는 시간 n 에서 적응 필터의 계수가 과거에 추정된 $\hat{y}_n(k)$ 에 종속되므로 최소자승면에서 준최적값(suboptimal value)을 제공한다.

식(4)를 최소화하는 최적값 $W^*(n)$ 은 다음과 같이 구할 수 있다.

$$W^*(n) = R^{-1}(n)P(n) \quad (7)$$

여기서 $R(n)$ 과 $P(n)$ 는 각각 다음과 같다.

$$R(n) = \sum_{k=1}^n \lambda^{n-k} U(k)U^T(k) \quad (8)$$

$$P(n) = \sum_{k=1}^n \lambda^{n-k} U(k)y(k) \quad (9)$$

식(1)과 같은 비선형 모델을 격자필터로 개발하기 위하여 본 논문에서는 참고문헌[13-14]에서와 같이 비선형 문제를 등가의 다채널 필터링 문제로 변환시켰다. 방법은 입력벡터 $U(n)$ 을 $2N$ 개의 부분집합으로 나눌 수 있고, 이 $2N$ 개의 부분집합은 각각 서로 다른 입력채널의 연속적인 입력신호로 이루어졌다고 생각할 수 있다.

첫번째 채널은 N 개의 계수를 가지며, 2, 3, 4번째 채널은 $N-i$ 개의 계수를 가지고, $k \geq 3$ 인 경우 $(2k-1)$ 번째와 $2k$ 번째 채널은 $N-k+1$ 개의 계수를 가지게 된다.

표현을 간단하게 하기 위하여 같은 수의 계수를 갖

는 채널들을 하나의 채널벡터로 묶으면 $2N$ 개의 채널은 다음과 같이 N 개의 채널로 정리할 수 있다.

$$\begin{aligned} x_0(n) &= [x(n)]^T \\ x_1(n) &= [y(n-1), x(n-1)y(n-1), x(n)y(n-1)]^T \\ x_2(n) &= [x(n-2)y(n-1), x(n)y(n-2)]^T \\ &\vdots \\ x_{N-1}(n) &= [x(n-N+1)y(n-1), x(n)y(n-N+1)]^T \end{aligned} \quad (10)$$

여기서 x_i 는 $i+1$ 번째 채널벡터로 $N-i$ 의 지연을 갖는다. 이것은 다음과 같이 나타낼 수 있다.

$$N_i = \begin{cases} N & i=1 \\ N-1 & i=2,3,4 \\ N-i+1 & i=2k-1, 2k(k \leq 3) \end{cases} \quad (11)$$

여기서 N_i 는 i 번째 채널의 계수의 수를 나타내고 가중치 벡터의 총수 L 은 다음과 같다.

$$L = N^2 + N - 1 \quad (12)$$

식(10)의 관계식을 이용하여 입력벡터와 계수벡터를 표현하면 아래와 같다.

$$\begin{aligned} X_N(n) = & [x_0^T(n), \dots, x_0^T(n-N+1), x_1^T(n), \\ & \dots, x_1^T(n-N+2), \dots, x_{N-1}^T(n)]^T \end{aligned} \quad (13)$$

$$\begin{aligned} C_N(n) = & [c_{0,0}^T(n), \dots, c_{0,N-1}^T(n-N+1), c_{1,0}^T(n), \\ & \dots, c_{1,N-2}^T(n-N+2), \dots, c_{N-1,0}^T(n)]^T \end{aligned} \quad (14)$$

입력벡터 $X_N(n)$ 과 $U(n)$ 의 관계와 계수벡터 $C_N(n)$ 과 $W(n)$ 의 관계는 각각 다음과 같다.

$$X_N(n) = T_N U(n) \quad (15)$$

$$C_N(n) = T_N W(n) \quad (16)$$

여기서 T_N 은 변환행렬이다.

식(13)과 식(14)를 사용하여 출력 $\hat{y}_n(n)$ 을 나타내면 다음과 같다.

$$\hat{y}_n(n) = C_N^T(n) X_N(n) \quad (17)$$

식(17)을 이용하여 식(4)을 최소화하는 계수벡터 $C_N(n)$ 을 구하면 다음과 같다.

$$C_N(n) = R_N^{-1}(n) \Delta_N(n) \quad (18)$$

$$R_N(n) = \sum_{k=1}^n \lambda^{n-k} X_N(k) X_N^T(k) \quad (19)$$

$$\Delta_N(n) = \sum_{k=1}^n \lambda^{n-k} X_N(k) y(k) \quad (20)$$

$R_N^{-1}(n)$ 은 역행렬 공리를 사용하여 다음과 같이 구할 수 있다.

$$R_N^{-1}(n) = \lambda^{-1} R_N^{-1}(n-1) - \frac{\lambda^{-2} R_N^{-1}(n-1) X_N^T(n) X_N(n) R_N^{-1}(n-1)}{1 + \lambda^{-1} X_N^T(n) R_N^{-1}(n-1) X_N(n)} \quad (21)$$

쌍선형 필터의 격자식을 유도하기 위하여 참고문헌 [13-14]에서와 같이 입력 데이터에 대해 Gram-Schmidt 직교화 과정을 수행하면 m 차 순방향 예측 오차, 역방향 예측오차 및 보조예측오차 벡터의 차수 순환식 및 접합과정 예측기의 예측오차 벡터에 대한 차수 순환식을 얻을 수 있다. 표 1에 완전한 쌍선형 필터의 격자식을 나타내었다.

표 1. 쌍선형 격자필터
Table 1. Lattice Bilinear Filter.

Initialization	
$f_0(n) = b_0(n) = z(n)$	(T-1.1)
$f_p^{(m)}(n) = \begin{cases} [d(n-1), z(n-1)d(n-1), z(n)d(n-1)]^T, & p=1 \\ [z(n-p)d(n-1), z(n)d(n-p)]^T, & p=2,3,\dots,N-1 \end{cases}$	(T-1.2)
$e_0(n) = d(n)$	(T-1.3)
Iteration Procedure	
DO (T-1.4)-(T-1.7), for $m = 1, 2, \dots, N-1$	
$f_m(n) = \begin{bmatrix} f_{m-1}(n) - K_m^{ff}(n) b_{m-1}(n-1) \\ f_{m-1}^{(m)}(n) - K_m^{fm}(n) b_{m-1}(n-1) \end{bmatrix}$	(T-1.4)
$b_m(n) = \begin{bmatrix} b_{m-1}(n-1) - K_m^{bb}(n) f_{m-1}(n) \\ f_{m-1}^{(m)}(n) - K_m^{bm}(n) f_{m-1}(n) \end{bmatrix}$	(T-1.5)
DO (T-1.6), for $p = m+1, m+2, \dots, N-1$	
$f_p^{(m)}(n) = f_{m-1}^{(p)}(n) - K_p^{fp}(n) b_{m-1}(n-1)$	(T-1.6)
$e_m(n) = e_{m-1}(n) - k_m^{ee}(n) b_{m-1}(n)$	(T-1.7)
$e_N(n) = e_{N-1}(n) - k_N^{eN}(n) b_{N-1}(n)$	(T-1.8)

† For equation error formulation, $d(n) = y(n)$.
For output error formulation, $d(n) = y(n) - e_N(n)$.

III. QR 분해법을 이용한 적응 쌍선형 격자 알고리즘

총래의 순환 최소 자승 알고리즘은 식 (7)에서 $W(n)$ 을 순환적으로 계산하기 위하여 $R^{-1}(n)$ 을 구하는 데 역행렬 공리를 사용하였다. 그러나 이러한 최소 자승 알고리즘 계열의 고속 알고리즘들은 일반적으로 수치적 안정도가 좋지 못하다. 이를 해결하는 하나의 방법으로 QR 분해법을 이용한 알고리즘들이 있다. 이러한 알고리즘들은 모든 연산을 수치적으로 안정한 직교변환을 사용하여 수행하기 때문에 수치적

안정도가 높다. Gentleman과 Kung^[15]은 입력벡터 $X_N(n)$ 을 QR 분해하여 가중치벡터 $C_N(n)$ 을 계산하는 알고리즘을 제안하였다. 이 알고리즘은 선형부분과 삼각형부분의 시스템릭 어레이로 구성되어 있다. 또한 McWhiter^[16]은 삼각형부분의 어레이만을 사용한 단순한 시스템릭 어레이 구조를 제안하였다. McWhiter의 QR 분해 최소자승 알고리즘은 QR 분해와 대수학적으로 등가인 Cholesky 분해법을 이용하여 유도되었다. 이와 같은 방법을 이용한 알고리즘에 관한 연구로서는 Lewis^[20], Bohme^[21] 등이 있다. 본 장에서는 2장에서 다룬 RLSL 알고리즘에 Lewis, Bohme 등과 유사한 QR 분해법을 이용하여 QRD-LSL 알고리즘을 유도하였다. 일반적으로 QR 분해법은 다음과 같다.

$$Q(n) \begin{bmatrix} \Phi(n-1) \\ X_N^T(n) \end{bmatrix} = \begin{bmatrix} \Phi(n) \\ 0^T(n) \end{bmatrix} \quad (22)$$

여기서 $\Phi(n)$ 은 상삼각행렬(upper triangular matrix)이고, $0(n)$ 은 요소가 0인 벡터이다. $Q(n)$ 은 N 개의 Givens 회전법에 근거한 직교행렬로 다음과 같다.

$$Q(n) = Q_L(n) \cdots Q_2(n) Q_1(n) \quad (23)$$

이 때 각 회전행렬들은 다음과 같이 구성된다.

$$Q_i(n) = \begin{bmatrix} I_{i-1} & 0 & \cdots & 0 \\ 0 & \cos \theta_i(n) & \cdots & \sin \theta_i(n) \\ 0 & 0 & \ddots & 0 \\ \vdots & \vdots & I_{N-i} & \vdots \\ 0 & 0 & \cdots & 0 \\ 0 & -\sin \theta_i(n) & \cdots & \cos \theta_i(n) \end{bmatrix}, i=1,2,\dots,L \quad (24)$$

여기서 회전각 $\theta(n)$ 은 $X_N(n)$ 의 i 번째 요소를 제거하도록 다음과 같이 구한다.

$$\cos \theta_i(n) = \frac{\phi_i(n)}{\sqrt{\phi_i^2(n) + x_i^2}} \quad (25)$$

$$\sin \theta_i(n) = \frac{x_i}{\sqrt{\phi_i^2(n) + x_i^2}} \quad (26)$$

여기서 $\phi_i(n)$ 는 $\Phi(n)$ 의 (i,i) 번째 요소이고 x_i 는 $X_N(n)$ 의 i 번째 요소이다. 위의 방법으로 $X_N(n)$ 을 계속 제거하면 $\Phi(n)$ 은 상삼각행렬이 된다. 그리고 $X_N(n)$ 의 자기 상관행렬 $R_N(n)$ 을 Cholesky 분해하면 다음과 같이 쓸 수 있다.

$$R_N(n) = \Phi_N^T(n) \Phi_N(n) \quad (27)$$

여기서 $\Phi_N(n)$ 은 양의 대각요소를 갖는 유일한 Cholesky 요소이다.

또한 기술하기 편하게 하기 위하여 다음과 같이 정의하자.

$$\Gamma_N(n) = \Phi_N^{-T}(n)\Delta_N(n) \quad (28)$$

$$\underline{\beta}_N(n) = \Phi_N^{-T}(n)X_N(n) \quad (29)$$

식(3.7)과 식(3.8)을 이용하면 가중치벡터와 추정 오차는 다음과 같이 구할 수 있다.

$$C_N(n) = \Phi_N^{-T}(n)\Gamma_N(n) \quad (30)$$

$$e(n) = y(n) - \Gamma_N^T(n)\underline{\beta}_N(n) \quad (31)$$

일반적으로 직교행렬 $Q(n)$ 은 다음과 같은 성질을 갖는다.

직교행렬의 성질

1. Givens 회전 행렬은 유니터리 행렬 (unitary matrix)이다.

$$Q^T Q = I \quad (32)$$

$$\| QX \|^2 = \| X \|^2 \quad (33)$$

2. 만일

$$B_1 = Q A_1 \quad (34)$$

이고

$$B_2 = QA_2 \quad (35)$$

이면

$$B_1^T B_2 = A_1^T A_2 \quad (36)$$

이다.

위의 두 가지 성질을 이용하여 기존의 RLSL 알고리즘을 QRD-LSL 알고리즘으로 바꿀 수 있다. 또한 시간 $n-1$ 에서 $\Phi_N(n-1)$ 과 $\Gamma_N(n-1)$ 이 알려져 있다고 가정하면 시간 n 에서 $X_N(n)$ 과 $y(n)$ 에 대한 직교화 과정이 필요하게 된다. 이과정은 다음과 같이 쉽게 구할 수 있다.

$$Q(n) \begin{bmatrix} \sqrt{\lambda}\Phi_N(n-1) & \sqrt{\lambda}\Gamma_N(n-1) & 0 \\ X_N^T(n) & y(n) & 1 \end{bmatrix} = \begin{bmatrix} \Phi_N(n) & \Gamma_N(n) & \underline{\beta}_N(n) \\ 0^T & \tilde{e}(n) & \tilde{\alpha}_N(n) \end{bmatrix} \quad (37)$$

여기서 $\tilde{\alpha}_N(n)$ 과 likelihood 변수 $\alpha_N(n)$ 의 관계는 다음과 같다.

$$\tilde{\alpha}_N(n) = \sqrt{\alpha_N(n)} \quad (38)$$

그리고 $\tilde{e}(n)$ 은 $e(n)$ 을 $\tilde{\alpha}_N(n)$ 으로 정규화하여 얻는다.

$$\tilde{e}(n) = \frac{e(n)}{\tilde{\alpha}_N(n)} \quad (39)$$

식(38)은 다음과 같이 나타낼 수 있다.

$$\tilde{\alpha}_N(n) = \prod_{i=1}^L \cos\theta_i(n) \quad (40)$$

식(37)을 전치하여 그 식의 앞에 곱해 주면 다음과 같이 몇 가지의 중요한 결과를 얻을 수 있다.

$$\begin{aligned} \Phi_N^T(n)\Phi_N(n) &= \lambda\Phi_N^T(n-1)\Phi_N(n-1) + X_N(n)X_N^T(n) \\ &= \lambda R_N(n-1) + X_N(n)X_N^T(n) \\ &= R_N(n) \end{aligned} \quad (41)$$

$$\begin{aligned} \Phi_N^T(n)\Gamma_N(n) &= \lambda\Phi_N^T(n-1)\Gamma_N(n-1) + X_N(n)y(n) \\ &= \lambda\Delta_N(n-1) + X_N(n)y(n) \\ &= \Delta_N(n) \end{aligned} \quad (42)$$

$$\Phi_N^T(n)\underline{\beta}_N(n) = X_N(n) \quad (43)$$

$$\Gamma_N^T(n)\underline{\beta}_N(n) + \tilde{e}(n)\tilde{\alpha}_N(n) = d(n) = \Gamma_N^T(n)\underline{\beta}_N(n) + e(n) \quad (44)$$

Givens 회전법을 이용한 격자필터를 유도하기 위해서는 순방향 예측오차, 역방향 예측 오차 그리고 접합 과정 예측기 등의 문제에 위의 처리과정을 적용해야 한다. 이 처리 과정에서 필요한 각 변수에 대한 정의는 표 2에 나타내었다.

표 2. 쌍선형 필터의 QRD-LSL 알고리즘의 변수와 RLSL 알고리즘의 변수의 관계
Table 2. Relationships Between The Variables Employed in QRD-LSL and RLSL Algorithms For Bilinear Filter.

$\hat{\alpha}_m(n) = \sqrt{\alpha_m(n)}$	$K_m^f(n) = \Phi_{m-1}^{fT}(n)\Phi_{m-1}^f(n)$	$\Gamma_{m-1}^f(n) = \Phi_{m-1}^{fT}(n)\Delta_{m-1}^f(n)$
$\hat{c}_m(n) = c_m(n)/\hat{\alpha}_m(n)$	$H_m^b(n) = \Phi_{m-1}^{bT}(n)\Phi_{m-1}^b(n)$	$\Gamma_{m-1}^b(n) = \Phi_{m-1}^{bT}(n)\Delta_{m-1}^b(n)$
$\hat{f}_m(n) = f_m(n)/\hat{\alpha}_m(n-1)$	$\hat{a}_{m-1}^f(n) = \Phi_{m-1}^{fT}(n)\hat{f}_{m-1}(n)$	$\Gamma_{m-1}^f(n) = \Phi_{m-1}^{fT}(n)\Delta_{m-1}^f(n)$
$\hat{b}_m(n) = b_m(n)/\hat{\alpha}_m(n)$	$\hat{a}_{m-1}^b(n) = \Phi_{m-1}^{bT}(n)\hat{b}_{m-1}(n)$	$\Gamma_{m-1}^b(n) = \Phi_{m-1}^{bT}(n)\Delta_{m-1}^b(n)$
$\hat{f}_m^b(n) = f_m^b(n)/\hat{\alpha}_m(n-1)$	$\hat{b}_m^f(n) = b_m^f(n)/\hat{\alpha}_m(n)$	$\Gamma_{m-1}^{fb}(n) = \Phi_{m-1}^{fbT}(n)\Delta_{m-1}^{fb}(n)$

표 2를 이용하여 표 1의 RLSL 알고리즘의 반사계 수 변환하면 다음과 같다.

$$K_m^f(n) = \Phi_{m-1}^{-b}(n-1)\Gamma_{m-1}^b(n) \quad (45)$$

$$K_m^b(n) = \Phi_{m-1}^{-f}(n)\Gamma_{m-1}^f(n) \quad (46)$$

$$K_m^y(n) = \Phi_{m-1}^{-b}(n)\Gamma_{m-1}^y(n) \quad (47)$$

$$K_m^{f(m)}(n) = \Phi_{m-1}^{-b}(n-1)\Gamma_{m-1}^{b(m)}(n) \quad (48)$$

$$K_m^{b(m)}(n) = \Phi_{m-1}^{-f}(n)\Gamma_{m-1}^{f(m)}(n) \quad (49)$$

또한 표 2.의 변수와 식(45) - 식(49)을 이용하여 표 1의 식(T-2.8)에서 식(T-2.17)을 QR 분해하면 다음과 같이 나타낼 수 있다.

$$Q_{m-1}'(n) \begin{bmatrix} \sqrt{\lambda}\Phi_{m-1}'(n-1) & \sqrt{\lambda}\Gamma_{m-1}'(n-1) & \sqrt{\lambda}\Gamma_{m-1}^{f(m)}(n-1) & 0 \\ \tilde{f}_{m-1}'(n) & \tilde{b}_{m-1}'(n) & \tilde{f}_{m-1}^{(m)T}(n) & \tilde{\alpha}_{m-1}(n-1) \end{bmatrix} = \begin{bmatrix} \Phi_{m-1}'(n) & \Gamma_{m-1}'(n) & \Gamma_{m-1}^{f(m)}(n) & \beta_{m-1}'(n) \\ 0^T & \tilde{b}_{m-1}'(n) & \tilde{b}_{m-1}^{(m)T}(n) & \tilde{\alpha}_{m-1}'(n) \end{bmatrix} \quad (50)$$

$$Q_{m-1}^b(n-1) \begin{bmatrix} \sqrt{\lambda}\Phi_{m-1}^b(n-2) & \sqrt{\lambda}\Gamma_{m-1}^b(n-1) & \sqrt{\lambda}\Gamma_{m-1}^{b(m)}(n-1) & \sqrt{\lambda}\Gamma_{m-1}^{*b}(n-1) & 0 \\ \tilde{b}_{m-1}^b(n-1) & \tilde{f}_{m-1}^b(n) & \tilde{f}_{m-1}^{(m)T}(n) & \tilde{\epsilon}_{m-1}(n) & \tilde{\alpha}_m(n-1) \end{bmatrix} = \begin{bmatrix} \Phi_{m-1}^b(n-1) & \Gamma_{m-1}^b(n) & \Gamma_{m-1}^{b(m)}(n) & \Gamma_{m-1}^{*b}(n) & \beta_{m-1}^b(n) \\ 0^T & \tilde{f}_{m-1}^b(n) & \tilde{f}_{m-1}^{(m)T}(n) & \tilde{\epsilon}_m(n) & \tilde{\alpha}_m(n-1) \end{bmatrix} \quad (51)$$

식(50), 식(51)으로부터 QR 분해법을 이용한 완전한 쌍선형 필터의 순환식을 얻을수 있다. QR 분해법을 이용한 완전한 적응 쌍선형 격자 알고리즘은 표 3에 나타내었다. QR 분해법을 이용한 적응 쌍선형 격자 알고리즘에서 식(45) 식(49)의 반사계수는 후진 대입법으로 풀 수 있으나 다음과 같은 식을 이용하여 순환적으로 구할 수 있다.

$$Q_{m-1}^{f'}(n) \begin{bmatrix} \Phi_{m-1}^{-f'}(n-1) / \sqrt{\lambda} \\ 0^T \end{bmatrix} = \begin{bmatrix} \Phi_{m-1}^{-f'}(n) \\ \tilde{g}_m^{f'}(n) \end{bmatrix} \quad (52)$$

$$Q_{m-1}^b(n) \begin{bmatrix} \Phi_{m-1}^{-bT}(n-1) / \sqrt{\lambda} \\ 0^T \end{bmatrix} = \begin{bmatrix} \Phi_{m-1}^{-bT}(n) \\ \tilde{g}_m^{bT}(n) \end{bmatrix} \quad (53)$$

식(52)을 전치하여 식(50)의 앞에 곱하면 다음 식을 얻을 수 있다.

$$\Phi_{m-1}^{-f}(n-1)\Phi_{m-1}^f(n-1) = \Phi_{m-1}^{-f}(n)\Phi_{m-1}^f(n) = I \quad (54)$$

$$\Phi_{m-1}^{-f}(n)\Gamma_{m-1}^f(n) = \Phi_{m-1}^{-f}(n-1)\Gamma_{m-1}^f(n-1) - \tilde{g}_m^f(n)\tilde{b}_m^T(n) \quad (55)$$

$$\Phi_{m-1}^{-f}(n)\Gamma_{m-1}^{f(m)}(n) = \Phi_{m-1}^{-f}(n-1)\Gamma_{m-1}^{f(m)}(n-1) - \tilde{g}_m^f(n)\tilde{b}_m^{(m)T}(n) \quad (56)$$

위의 식(55)는 $\Phi_{m-1}^f(n)$ 의 역행렬을 이와 같은 방법으로 구할 수 있음을 나타내고, 식(55)와 식(56)에 반사계수 정의식(46) 와 (49)를 대입 정리하면 다음과 같이 구하여 진다.

$$K_m^b(n) = K_m^b(n-1) - \tilde{g}_m^f(n)\tilde{b}_m^T(n) \quad (57)$$

$$K_m^{b(m)}(n) = K_m^{b(m)}(n-1) - \tilde{g}_m^f(n)\tilde{b}_m^{(m)T}(n) \quad (58)$$

이와 같은 방법으로 순방향 예측오차, 순방향 보조 예측오차 및 접합과정 예측기의 반사계수를 구하면 다음과 같다.

$$K_m^f(n) = K_m^f(n-1) - \tilde{g}_m^b(n-1)\tilde{f}_m^T(n) \quad (59)$$

$$K_m^{f(p)}(n) = K_m^{f(p)}(n-1) - \tilde{g}_m^b(n-1)\tilde{f}_m^{(p)T}(n) \quad (60)$$

$$K_m^v(n) = K_m^v(n-1) - \tilde{g}_m^b(n)\tilde{\epsilon}_m(n) \quad (61)$$

$$K_m^v(n) = K_m^v(n-1) - \tilde{g}_m^b(n)\tilde{\epsilon}_m^T(n) \quad (62)$$

직접형 필터의 계수는 위에서 구한 반사계수를 Baik^[15-61]이 제안한 방법에 적용하여 구할 수 있다. 그 결과는 4장의 실험결과에 나타내었다.

표 3. QR 분해법을 이용한 순환 최소화승 쌍선형 격자 알고리즘

Table 3. QRD Based least squares lattice adaptive bilinear algorithm.

Initialization	
DO (T-3.1) for $m = 0, 1, \dots, N-1$	
$\Phi_m^f(n) = \Phi_m^f(0) = \begin{cases} \delta & \text{if } m = 0 \\ \delta I_{2(m+1)}, & \text{otherwise} \end{cases}$	(T-3.1)
$f_0(n) = b_0(n) = z(n)$	(T-3.2)
DO (T-3.2)-(T-3.5) for $n = 1, 2, \dots$	
$\hat{a}_0(n) = 1$	(T-3.3)
$f_0(n) = b_0(n) = z(n)$	(T-3.4)
$f_m^{(p)T}(n) = \begin{cases} [d(n-1), z(n-1)d(n-1), z(n)d(n-1)]^T, & p = 1 \\ [z(n-p)d(n-1), z(n)d(n-p)]^T, & p = 2, 3, \dots, N-1 \end{cases}$	(T-3.5)
$\epsilon_0(n) = d(n)$	(T-3.6)
Iteration Procedure	
DO (T-3.6) - (T-3.10) for $m = 1, 2, \dots, N-1$	
$Q_m^f(n) \begin{bmatrix} \sqrt{\lambda}\Phi_{m-1}^f(n-1) & \sqrt{\lambda}\Gamma_{m-1}^f(n-1) \\ \tilde{f}_{m-1}^f(n) & \tilde{b}_{m-1}^f(n-1) \end{bmatrix} = \begin{bmatrix} \Phi_{m-1}^f(n) & \Gamma_{m-1}^f(n) \\ 0^T & \tilde{b}_{m-1}^f(n) \end{bmatrix}$	(T-3.7)
$Q_m^b(n-1) \begin{bmatrix} \sqrt{\lambda}\Gamma_{m-1}^{b(p)}(n-1) \\ \tilde{f}_{m-1}^{b(p)T}(n) \end{bmatrix} = \begin{bmatrix} \Gamma_{m-1}^{b(p)}(n) \\ \tilde{f}_{m-1}^{b(p)T}(n) \end{bmatrix}$	(T-3.8)
DO (T-3.8) for $p = m+1, m+2, \dots, N-1$	
$Q_m^b(n-1) \begin{bmatrix} \sqrt{\lambda}\Gamma_{m-1}^{b(p)}(n-1) \\ \tilde{f}_{m-1}^{b(p)T}(n) \end{bmatrix} = \begin{bmatrix} \Gamma_{m-1}^{b(p)}(n) \\ \tilde{f}_{m-1}^{b(p)T}(n) \end{bmatrix}$	(T-3.9)
$Q_m^b(n) \begin{bmatrix} \sqrt{\lambda}\Gamma_{m-1}^{b(m)}(n-1) \\ \tilde{f}_{m-1}^{b(m)T}(n) \end{bmatrix} = \begin{bmatrix} \Gamma_{m-1}^{b(m)}(n) \\ \tilde{f}_{m-1}^{b(m)T}(n) \end{bmatrix}$	(T-3.10)
$\tilde{f}_m(n) = \begin{bmatrix} f_m(n) \\ \tilde{f}_m^{(m)T}(n) \end{bmatrix}$	(T-3.11)
$\tilde{b}_m(n) = \begin{bmatrix} b_m(n) \\ \tilde{b}_m^{(m)T}(n) \end{bmatrix}$	(T-3.12)
$Q_m^v(n) \begin{bmatrix} \sqrt{\lambda}\Phi_{m-1}^v(n-1) & \sqrt{\lambda}\Gamma_{m-1}^v(n-1) & 0 \\ \tilde{b}_{m-1}^v(n) & \tilde{f}_{m-1}^v(n) & \tilde{\alpha}_{m-1}^v(n) \end{bmatrix} = \begin{bmatrix} \Phi_{m-1}^v(n) & \Gamma_{m-1}^v(n) & \beta_{m-1}^v(n) \\ 0^T & \tilde{b}_{m-1}^v(n) & \tilde{\alpha}_{m-1}^v(n) \end{bmatrix}$	(T-3.13)
$Q_m^v(n) \begin{bmatrix} \sqrt{\lambda}\Phi_{m-1}^v(n-1) & \sqrt{\lambda}\Gamma_{m-1}^v(n-1) & 0 \\ \tilde{b}_{m-1}^v(n) & \tilde{f}_{m-1}^v(n) & \tilde{\alpha}_{m-1}^v(n) \end{bmatrix} = \begin{bmatrix} \Phi_{m-1}^v(n) & \Gamma_{m-1}^v(n) & \beta_{m-1}^v(n) \\ 0^T & \tilde{b}_{m-1}^v(n) & \tilde{\alpha}_{m-1}^v(n) \end{bmatrix}$	(T-3.14)
$\epsilon_m(n) = \tilde{v}_m(n)\tilde{\alpha}_m^T(n)$	(T-3.15)
† For equation error formulation, $d(n) = p(n)$	
‡ For output error formulation, $d(n) = \hat{y}(n) - z(n)$	

IV. 컴퓨터 시뮬레이션 결과

알고리즘의 성능을 평가하기 위하여 미지의 쌍선형 시스템의 시스템 식별 실험을 하였다또 본 논문에서

제한한 알고리즘의 성능을 조사하기 위하여 식오차 방법과 출력오차 방법에 대해 RLSL 알고리즘과의 성능을 비교하였다. 적응필터의 입력신호 $x(n)$ 은 전달함수가 식(63)과 같은 저역통과 필터에 평균이 0인 백색 가우시안 잡음을 통과시켜 얻었고 출력신호의 분산은 1이 되도록 조정하였다.

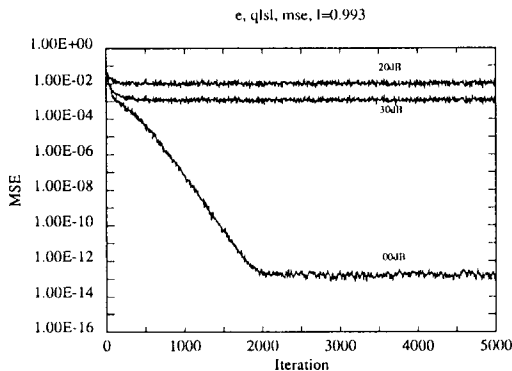
$$H(z) = \frac{1}{1 - 1.6z^{-1} + 0.95z^{-2}} \quad (63)$$

원하는 응답신호 $d(n)$ 은 표 4에 있는 계수를 이용하여 구한 다음 S/N비가 ∞ dB, 30dB, 20dB가 되도록 측정잡음을 첨가하여 얻었다. 이 측정잡음은 평균이 0인 백색 가우시안 잡음으로 입력신호와는 상관관계가 없는 것으로 선택하였다. 각 경우에 50번의 독립된 실험을 수행하여 앙상블 평균(ensemble average)을 구하였다.

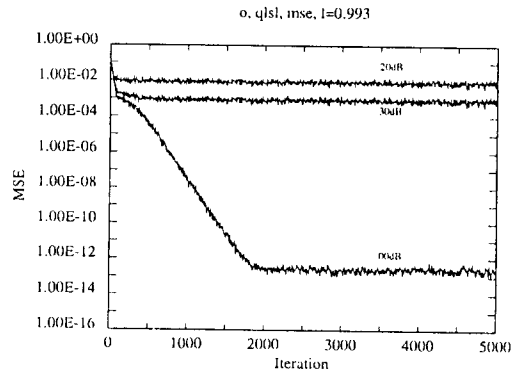
표 4. 시스템 식별에 사용한 미지의 쌍선형 계수
Table 4. Coefficients of the Unknown Bilinear Filter used in the System Identification.

$a_0 = 1.0$	$a_1 = 1.0$	$a_2 = 1.0$
$c_{01} = 0.3$	$c_{11} = -0.2$	$c_{21} = 0.1$
$c_{02} = 0.1$	$c_{12} = -0.2$	$c_{22} = 0.3$
	$b_1 = 0.5$	$b_2 = -0.5$

그림 1은 각각 $\lambda = 0.993$ 일 때 식오차 방법과 출력오차 방법에 대한 평균 자승오차를 나타낸다. 이 그림을 보면 제안한 알고리즘은 식오차 방법과 출력오차 방법에서 약 2000번 정도에서 수렴함을 알 수 있고 수렴특성이 우수함을 보여준다.



(a)



(b)

그림 1. 학습곡선

- (a) 식오차 알고리즘
- (b) 출력오차 알고리즘

Fig. 1. Learning curves

- (a) Equation error algorithm,
- (b) Output error algorithm.

표 5. 마지막 2000개 평균자승 오차 데이터 샘플의 시간 평균값

Table 5. Time-Averaged Mean Squared Error over the Last 2000 Data Samples.

(a) QRD-LSL algorithm

λ	Equation Error		Output Error	
	$\lambda = 0.993$	$\lambda = 0.95$	$\lambda = 0.993$	$\lambda = 0.95$
∞ dB	1.74038E-13	1.67065E-14	2.51531E-13	2.29836E-14
30dB	1.32052E-03	7.54598E-04	9.16178E-01	5.61879E-04
20dB	1.13963E-02	6.22165E-03	9.15875E-03	5.51856E-03

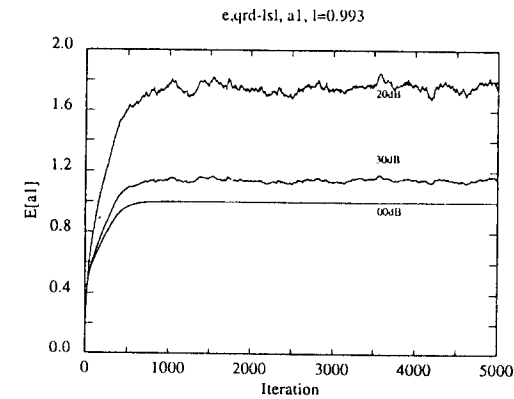
(b) RLSL algorithm

λ	Equation Error		Output Error	
	$\lambda = 0.993$	$\lambda = 0.95$	$\lambda = 0.993$	$\lambda = 0.95$
∞ dB	8.99419E-11	1.30692E-11	1.06553E-10	1.73703E-11
30dB	1.32052E-03	7.54579E-04	9.16178E-04	5.61879E-04
20dB	1.13963E-02	6.22165E-03	9.15876E-03	5.51856E-03

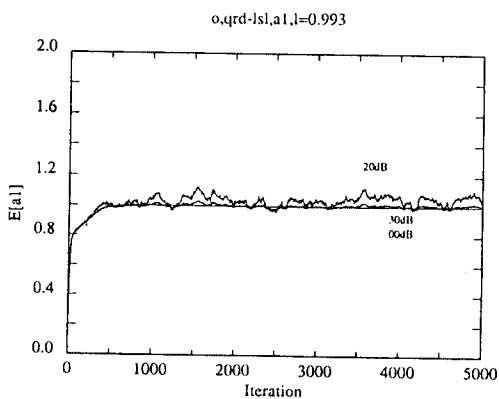
표 5는 $\lambda=0.993$ 인 경우와 $\lambda=0.95$ 인 경우 QRD-

LSL 알고리즘과 RLSL 알고리즘의 식오차 방법과 출력오차 방법에 대한 평균 자승오차를 나타낸다. 이 값은 $n=3001$ 부터 5000까지 2000개의 값을 표본 평균치들의 시간 평균값을 나타낸다. 표 5에서 식오차 방법과 출력오차 방법에서 측정잡음이 있는 경우 QRD-LSL 알고리즘과 RLSL 알고리즘은 거의 같은 수렴특성을 보이는 반면에 잡음이 없는 경우 QRD-LSL 알고리즘이 RLSL 알고리즘에 비하여 우수한 수렴특성을 나타내고 있다.

그리고 QRD-LSL 알고리즘과 RLSL 알고리즘에서 측정잡음이 있는 경우 출력오차 방법이 식오차 방법보다 우수함을 알 수 있다. 또한 λ 클 때 평균 자승오차가 커짐을 알 수 있다. 식오차 방법과 출력오차 방법에서 QRD-LSL 알고리즘의 계수 궤적을 보면 두 가지 방법의 성능 차이를 쉽게 알 수 있다.



(a)



(b)

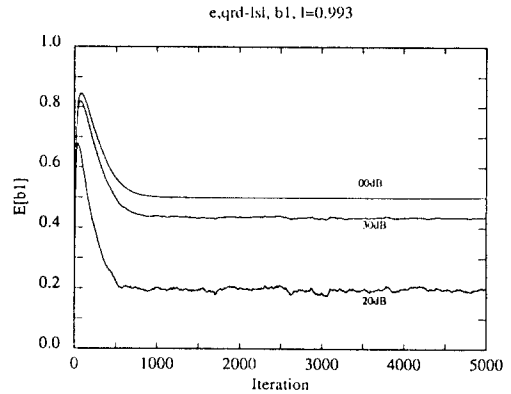
그림 2. 계수 $a_1(n)$ 의 평균 궤적

- (a) 식오차 알고리즘
- (b) 출력오차 알고리즘

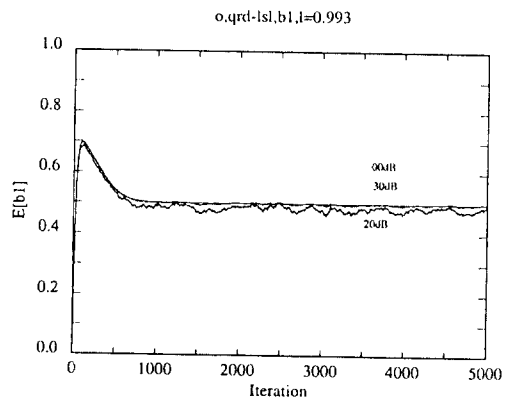
Fig. 2. Mean trajectory of coefficients $a_1(n)$

- (a) Equation error algorithm,
- (b) Output error algorithm.

$\lambda=0.993$ 일 때 계수 $a_1(n)$, $b_2(n)$, $c_{12}(n)$ 에 대한 표본평균 궤적을 그림 2, 그림 4에 나타내었다. 그림에서 비선형 부분에 해당되는 계수 $E[c_{12}]$ 가 선형부분에 해당하는 계수 $E[a_1]$ 와 $E[b_2]$ 에 비하여 궤적의



(a)



(b)

그림 3. 계수 $b_1(n)$ 의 평균 궤적

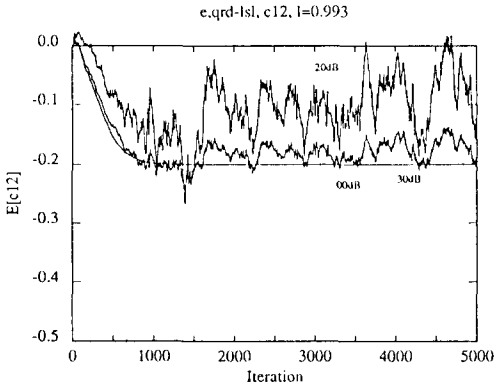
- (a) 식오차 알고리즘
- (b) 출력오차 알고리즘

Fig. 3. Mean trajectory of coefficients $b_1(n)$.

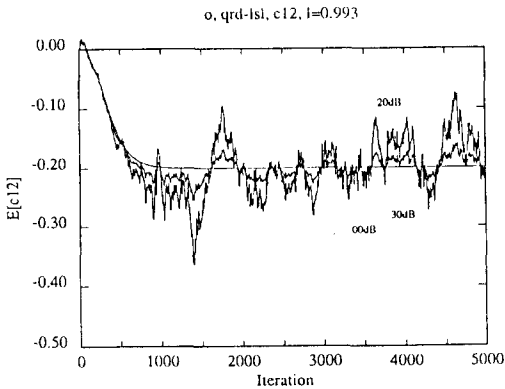
- (a) Equation error algorithm,
- (b) Output error algorithm.

변화가 심함을 알 수 있다. 또한 다른 계수에서도 같은 결과를 보임을 알 수 있고, RLSL 알고리즘에서도 같은 효과를 보임을 알 수 있다.

표 6은 $\lambda=0.993$ 일 때와 $\lambda=0.95$ 경우에 각 계수에 대한 표본 평균치들중 마지막 2000개의 샘플에 대한 계수 제적의 시간 평균치를 나타낸다. 여기서 측정잡음이 있는 경우 식 오차 방법에서 구한 계수들이 원하는 신호에 바이어스되기 때문에 다른 값에 수렴함을 알 수 있다.



(a)



(b)

그림 4. 계수 $c_{12}(n)$ 의 평균 궤적

- (a) 식오차 알고리즘
- (b) 출력오차 알고리즘

Fig. 4. Mean trajectory of coefficients $c_{12}(n)$
 (a) Equation error algorithm,
 (b) Output error algorithm.

표 6. 각 계수의 마지막 2000개 시간 평균치
 Table 6. Time-averaged coefficients over the last 2000 data samples.

(a) $\lambda = 0.993$

QRD-LSL	Equation Error			Output Error			W*
	∞ dB	30dB	20dB	∞ dB	30dB	20dB	
a_0	1.000	0.996	0.991	1.000	0.999	0.991	1.0
a_1	1.000	1.150	1.766	1.000	1.007	1.051	1.0
a_2	1.000	1.005	0.817	1.000	0.997	0.991	1.0
b_1	0.500	0.434	0.197	0.500	0.498	0.181	0.5
b_2	-0.500	-0.471	-0.323	-0.500	-0.499	-0.491	-0.5
c_{01}	0.300	0.304	0.303	0.300	0.304	0.311	0.3
c_{02}	0.100	0.099	0.099	0.100	0.097	0.091	0.1
c_{10}	-0.200	-0.209	-0.266	-0.200	-0.207	-0.225	-0.2
c_{11}	-0.200	-0.175	-0.086	-0.200	-0.195	-0.180	-0.2
c_{21}	0.100	0.086	0.109	0.100	0.106	0.117	0.1
c_{22}	0.300	0.317	0.319	0.300	0.296	0.290	0.3

(b) $\lambda = 0.95$

QRD-LSL	Equation Error			Output Error			W*
	∞ dB	30dB	20dB	∞ dB	30dB	20dB	
a_0	1.000	1.001	1.014	1.000	1.000	1.001	1.0
a_1	1.000	1.180	1.809	1.000	1.044	1.354	1.0
a_2	1.000	1.038	0.950	1.000	1.010	1.081	1.0
b_1	0.500	0.411	0.138	0.500	0.480	0.325	0.5
b_2	-0.500	-0.465	-0.317	-0.500	-0.492	-0.435	-0.5
c_{01}	0.300	0.300	0.283	0.300	0.303	0.298	0.3
c_{02}	0.100	0.010	0.112	0.100	0.100	0.103	0.1
c_{10}	-0.200	-0.199	-0.237	-0.200	-0.208	-0.214	-0.2
c_{11}	-0.200	-0.175	-0.109	-0.200	-0.191	-0.158	-0.2
c_{21}	0.100	0.068	0.180	0.100	0.104	0.072	0.1
c_{22}	0.300	0.332	0.355	0.300	0.303	0.347	0.3

V. 결론

본 논문에서는 기존의 쌍선형 필터의 RLSL 알고리즘에 QR 분해와 등가인 Cholesky 분해법을 이용하여 새로운 최소 자승 격자 알고리즘(QRD-LSL)을 제안하였다.

이 새로운 알고리즘은 기존의 식들을 그대로 이용할 수가 있어 다른 QR 알고리즘보다 쉽게 유도할 수 있는 장점이 있다. 또한 본 논문에서 제안한 알고리즘은 모든 연산을 수치적으로 안정한 직교화 변환을 사용하여 수행하므로 수치적 안정도가 높다. 제안한 알고리즘의 성능을 평가하기 위하여 미지의 쌍선형 시스템의 시스템 식별 실험을 하였으며, 식오차방법과 출력오차 방법에 대해 제안한 알고리즘과 RLSL 알고리즘과의 성능을 비교하였다.

컴퓨터 시뮬레이션 결과 최소 자승 오차의 비교에서 측정 잡음이 있는 경우 RLSL 알고리즘과 거의 같은 성능을 나타냄을 알 수 있고, 측정잡음이 없는 경우에는 본 논문에서 제안한 알고리즘이 RLSL 알고리즘에 비해 우수한 성능을 나타냄을 알 수 있다.

제안한 식오차 알고리즘은 원하는 신호에 바이어스 되기 때문에 단 하나의 최소점으로 수렴하는 장점을 가지고 있으나 측정잡음이 큰 경우 쌍선형 시스템의 가중치 벡터에 수렴하지 못함을 알 수 있고, 출력오차 알고리즘에서는 준최적값을 제공하므로 측정잡음이 큰 경우에도 쌍선형 시스템의 가중치 벡터에 근처에 수렴함을 알 수 있다. 즉 출력오차 알고리즘이 식오차 알고리즘에 비하여 측정잡음에 강인함을 알 수 있다. 또 본 논문에서 제안한 알고리즘은 쌍선형 필터 뿐만 아니라 일반적인 비선형 필터에도 쉽게 확장할 수 있다.

앞으로 QRD 분해법을 이용한 알고리즘의 수렴특성 및 안정도가 이론적으로 해석되어야 할 것으로 생각된다.

參 考 文 獻

- [1] Haykin, *Adaptive Filter Theory*, Englewood Cliffs, N.J., Prentice Hall, 1991.
- [2] J. G. Proakis, C. M. Rader, F. Ling and C. L. Nikias, *Advanced Digital Signal Processing*, Macmillan Publishing Company.
- [3] V. J. Mathews and J. Lee, "A fast least-squares second-order Volterra filter," *Proc. Int. Cong. Acoust., Speech, Signal Processing*(New York), 1988.
- [4] M. Bellafemina and S. Benedetto, "Identification and equalization of nonlinear channels for digital transmission," *Proc. IEEE Int. Symp. Circuits and Systems*(Kyoto, Japan), pp.1477-1480, June 1985.
- [5] J. R. Casar-Corredera, M. Carcia-Otera, and A. Figueiras-Videl, "Data echo nonlinear cancellation," *Proc. Int. Conf. Acoust., Speech, Signal Processing* (Tampa, Florida), pp.32.4.1-4, Mar. 1985.
- [6] L. W. Prieto, J. C. Sueiro and A. F. Vidal, "Optimal variable LMS look-up table plus transversal filter nonlinear echo cancellers," *Proc. Int. Conf. Acoust., Speech, Signal Processing*(San Francisco, California), Mar. 1992.
- [7] G. L. Sicouranza, A. Buccioni, and P. Mitri, "Adaptive echo cancellation with nonlinear digital filters," *Proc. Int. Conf. Acoust., Speech, Signal Processing*(San Francisco, California), pp.3.10.1-4, Mar. 1984.
- [8] M. J. Coker and D. M. Simkins, "A nonlinear adaptive noise canceller," *Proc. Int. Conf. Acoust., Speech, Signal Processing*, pp.470-473, 1980.
- [9] J. C. Stapleton and S. C. Bass, "Adaptive noise cancellation for a class of nonlinear dynamic reference channels," *IEEE Trans. Circuits and Systems*, vol. CAS-32, no. 2, pp.143-150, Feb. 1985.
- [10] S. Narayanan, "Application of Volterra series to intermodulation distortion of transistor feedback amplifiers," *IEEE Trans. Circuits Theory*, vol. CT-17, pp. 518-527, Nov. 1970.
- [11] P. J. Marmarelis and V. Z. Marmarelis, *Analysis of Physiological Systems*, New York, Plenum, 1978.
- [12] J. G. McWhirter and T. J. Shepherd, "Systolic array processor for MVDR beamforming," *Proc. Inst. Elec. Eng.*, vol. 136, Pt. F, pp.75-80, 1989.
- [13] H. K. Baik and V. J. Mathews, "Adaptive lattice bilinear filters," *IEEE Trans. Signal Processing*, vol. 41, no. 6, pp.2033-2046, June, 1993.
- [14] H. K. Baik and V. J. Mathews, "Adaptive algorithms for identifying recursive nonlinear systems," *Proc. Int. Conf. Acoust., Speech, Signal Processing*(Toronto, Canada), May, 1991.
- [15] W. M. Gentleman and H. T. Kung, "Matrix triangularization by systolic arrays," *Proc. SPIE, Real Time Signal Processing* 6, vol. 298, pp. 298, 1981.
- [16] J. G. McWhirter, "Recursive least-squares minimization using a systolic array," *Proc. SPIE., Real Time Signal Processing* 6, vol. 431, pp.105-112, 1983.
- [17] F. Ling, D. Manolakis and J. G. Proakis, "A recursive modified Gram-

- Schmidt algorithm for least-squares estimation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp.829-836, Aug. 1986.
- [18] K. J. R. Liu, S. F. Hsieh and K. Yao, "Two-level pipelined implementation of systolic block Householder transformations with application to RLS algorithm," *to appear in the IEEE Trans. on Signal Processing*.
- [19] M. G. Bellanger, "Computational complexity and accuracy issues in fast least squares algorithms for adaptive filtering," *Proc. IEEE ISCAS(Finland)*, pp.2635-2539, 1988.
- [20] P. S. Lewis, "QR-based algorithms for multichannel adaptive least squares lattice filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, pp.421-432, Mar. 1990.
- [21] B. Yang and J. F. Bohme, "Rotation based RLS algorithms: Unified derivation, numerical properties and parallel implementations," *to appear in the IEEE Trans. on Signal Processing*.
- [22] H. Lev-Ari, "Modular Architectures for Adaptive Multichannel Lattice Algorithms," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, no. 4, pp.543-552, Apr. 1987.
- [23] M. A. Syed and V. J. Mathews, "QR-Decomposition based algorithm for adaptive volterra filtering," *to appear in the IEEE Trans. on Circuits and Systems*.
- [24] M. A. Syed and V. J. Mathews, "Lattice algorithms for recursive least squares adaptive volterra filtering," *to appear in the IEEE Trans. on Circuits and Systems*.
- [25] F. Ling and J. G. Proakis, "A generalized multichannel least squares lattice algorithm based on sequential processing stages," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, no. 2, pp.381-389, 1984.
- [26] C. S. Kubrusly and O. L. V. Costa, "Mean square stability conditions for discrete stochastic bilinear systems," *IEEE Trans. Automat. Contr.*, vol. AC-30, no. 11, pp. 1082-1087, Nov. 1985.
- [27] J. Liu, "On the existence of a general multiple bilinear time series," *J. Time Series Anal.*, vol. 10, no. 4, pp.341-355, 1989.
- [28] J. Lee and V. J. Mathews, "Adaptive bilinear predictors," *Proc. Int. Conf. Acoust., Speech, Signal Processing (Adelaide, South Australia)*, vol. 3, pp.489-492, 1994.

著 者 紹 介



安 奉 萬(正會員)

1962年 9月 17日生. 1988年 2月 전북대학교 공과대학 전자공학과 졸업(공학사). 1990年 2月 전북대학교 대학원 전자공학과 졸업(공학석사). 1990年 5月 ~ 1992年 2月 현대정공 기술연구소 연구원. 1992年 9月 ~ 1994年 2月 전북대학교 전자공학과 조교. 1994年 2月 전북대학교 대학원 전자공학과 박사과정 수료. 주관심 분야는 신호처리, 적응신호처리, 음성신호처리 등임.

黃 智 源(正會員) 第 30卷 3號 B編 參照

1957年 6月 28日生. 1985年 2月 전북대학교 공과대학 전자공학과 졸업(공학사). 1987年 2月 전북대학교 대학원 전자공학과 졸업(공학석사). 1993年 8月 전북대학교 대학원 전자공학과 박사과정 수료. 1994年 4月 ~ 현재 이리 농공 전문대학 전자계산학과 조교수. 주관심 분야는 신호처리, 적응신호처리, 음성신호처리, 비선형신호처리 등임.



白 興 基(正會員)

1955年 1月 5日生. 1977年 2月 서울대학교 공과대학 전자공학과 졸업(공학사). 1979年 8月 서울대학교 대학원 전자공학과 졸업(공학석사). 1987年 2月 서울대학교 대학원 전자공학과 졸업(공학박사). 1981年 3月 ~ 현재 전북대학교 공과대학 전자공학과 교수. 1990年 1月 ~ 1990年 12月 미국 Utah 대학 PostDoc연수. 주관심 분야는 디지털 신호처리, 적응신호처리, 비선형신호처리 등임.