

論文93-30B-3-2

적응 쌍선형 필터의 RPEM 알고리즘

RPEM Algorithm for Adaptive Bilinear Filter

白興基*, 黃智源**, 安奉萬*

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

要約

쌍선형 모델은 많은 비선형 시스템을 모델링하는 데 사용되고 있는 Volterra 모델에 비해 보다 적은 수의 계수로써 근사화 할 수가 있어 많은 수의 계수를 필요로 하는 비선형 적응 필터링 분야에 효과적으로 이용할 수 있다. 그러나 Volterra 필터와는 달리 쌍선형 필터는 궤환을 포함하기 때문에 필터의 안정성이 문제가 된다. 또 쌍선형 필터의 적응 알고리즘 역시 입력이 커지면 발산하거나 수렴특성이 좋지 않은 단점이 있다. 본 논문에서는 쌍선형 필터가 안정하기 위한 필요충분조건으로서 쌍선형 필터의 평균자승 안정에 대해 간단히 서술하였으며, 비교적 간단한 방법으로 쌍선형 필터가 안정하기 위한 입력신호의 크기를 구하는 방법을 제시하였다. 또 쌍선형 시스템의 직접형 구조에 대한 RPEM(recursive prediction error method) 알고리즘에 시변 칼만 필터링 기법을 도입하여 입력신호가 어느 정도 클 경우에도 발산하지 않고 수렴특성이 좋은 수정된 RPEM 알고리즘을 제시하였으며, 컴퓨터 시뮬레이션을 통해 새로운 RPEM 알고리즘이 기존의 RPEM 알고리즘에 비해 수렴특성이 우수함을 보였다.

Abstract

Bilinear models are attractive for adaptive filtering applications because they can approximate a large class of nonlinear systems adequately, and usually with considerable parsimony in the number of coefficients compared with Volterra models. But bilinear filters have stability problem because they involve nonlinear feedback. Adaptive algorithms for bilinear filters may be diverge and have poor convergence characteristics when input signal is large. In this paper, necessary and sufficient condition for mean square stability of bilinear filters for given input signal statistics is briefly described, and the method obtaining the input bound to guarantee the stability of bilinear filters is presented. New RPEM algorithm, which does not diverge and has the superior convergence characteristics compared with the conventional RPEM algorithm when input signal is large, is derived by applying the time-varying Kalman filtering concept to the conventional RPEM algorithm.

* 正會員, 全北大學校 電子工學科
(Dept. of Elec. Eng., Chonbuk Nat'l Univ.)

** 正會員, 裡里 農工專門大學 電子計算學科
(Dept. of Computerscience, Nat'l college of Iri A.T.)

(* 이 연구는 1992년도 한국과학재단 연구비지원에 의한 결과임.)

接受日字 1992年 9月 25日

1. 서론

선형필터와 선형 시스템 모델은 많은 응용분야에 사용되고 있고 이의 실현도 쉽지만 이를 적용할 수 없는 응용분야도 적지 않다. 이에 따라 비선형 구조를 이용한 시스템 해석이 채널등화, 방향제거, 잡음제거, 반도체 소자 모델링, 의용전자 현상의 모델링

등 여러 분야에 널리 이용되고 있다. 일반적으로 비선형 시스템은 Volterra 급수에 의해 잘 모델링된다.^{[1], [2]} 이 중에서 절단된 Volterra 급수에 바탕을 둔 2 차 Volterra 필터가 많이 사용되고 있으나 [3] - [5] 이 필터의 가장 큰 문제는 비선형 시스템을 모델링하는 데 많은 계수를 필요로 하며 이에 따라 많은 계산량을 필요로 한다는 점이다. 이같은 문제를 해결하기 위하여 계환을 포함하는 모델중 가장 간단한 모델로서 입력과 출력사이의 관계가 다음과 같이 주어지는 쌍선형 시스템에 관한 연구가 이루어졌다^[6]

$$y(n) = \sum_{i=1}^{N-1} a_i y(n-i) + \sum_{i=1}^{N-1} \sum_{j=0}^{N-1-i} b_{ij} y(n-i) u(n-j) + \sum_{j=1}^{N-1} c_j y(n-j) \quad (1.1)$$

IIR 필터가 FIR 필터에 비해 많은 선형 시스템을 간단하게 모델링하는 것처럼, 많은 비선형 시스템이 상대적으로 적은 수의 파라미터를 필요로 하는 비선형 계환 모델로 모델링될 수 있으며 이를 이용한 적응필터는 적은 수의 계산량으로 실현시킬 수 있다. 쌍선형 시스템의 또 다른 장점은 일반적인 조건하에서 임의의 정밀도를 갖는 어떤 Volterra 시스템도 이를 이용하여 근사화 할 수 있다는 점이다.^[7] 이와 같은 장점으로 인해 쌍선형 시스템은 제어 시스템, 인구 통계 모델, 의용전자 시스템, 경제분야 등 많은 분야에 이용되고 있다. 그러나 Volterra 필터와는 달리 쌍선형 시스템은 계환을 포함하기 때문에 시스템의 안정성이 문제가 된다. 쌍선형 시스템의 입력과 출력사이의 관계는 다음과 같다.

$$y(n) = \sum_{i=1}^{N-1} \left[a_i + \sum_{j=0}^{N-1-i} b_{ij} u(n-j) \right] y(n-i) + \sum_{j=0}^{N-1} c_j u(n-j) \quad (1.2)$$

위의 식을 보면 쌍선형 시스템의 안정조건이 입력 신호의 특성과 관계있음을 알 수 있다. 즉 입력신호의 크기가 커지면 시스템이 불안정해짐을 알 수 있다. 쌍선형 시스템의 안정성에 관한 연구는 Kuburly와 Liu 등에 의해 이루어졌다. [8] - [12] Kuburly는 쌍선형 시스템이 평균자승 안정하기 위한 필요충분조건을 구하였으며, Liu는 일반적인 쌍선형 시간수열(time series)의 정상해(stationary solution)가 존재하기 위한 충분조건을 구하였다. 그러나 이들 방법은 Kronecker 곱을 사용하기 때문에 행렬의 차수가 커지고 이에 따라 행렬의 고유치를 구하는 데 많은 계산량을 필요로 하기 때문에 실제 안정도를 조사하는 데 사용하기가 어려운 단점이 있다.

한편 쌍선형 시스템을 이용한 적응필터에 관한 연구는 아주 적다. 직접형 구조와 이에 관한 최소자승 알고리즘에 대해 참고문헌 [13] - [14] 에서 다루고 있으며, 격자구조와 이에 관한 최소자승 알고리즘에 대한 연구가 참고문헌 [15] - [19] 에 나타나 있다. 그러나 이러한 알고리즘 모두 입력신호가 커지면 알고리즘이 발산하거나 수렴특성이 좋지 않은 단점이 있다.

본 논문에서는 쌍선형 필터가 안정하기 위한 필요충분조건으로서 쌍선형 필터의 평균자승 안정에 대해 간단히 서술하였으며, 쌍선형 필터가 안정하기 위한 입력신호의 크기를 구하는 방법을 제시하였다. 본 논문에서 제안한 방법은 제어분야에서 선형 시스템이 안정하기 위한 견실한 영역(robustness bound)을 구하는 방법을 이용한 것으로 비교적 간단한 방법으로 쌍선형 필터가 안정하기 위한 입력의 크기를 구할 수 있다.

또 쌍선형 시스템의 직접형 구조에 대한 RPEM(recursive prediction error method) 알고리즘에 시변 칼만 필터링 기법을 도입하여 입력신호가 어느 정도 클 경우에도 발산하지 않고 수렴특성이 좋은 수정된 RPEM 알고리즘을 제시하였다. 본 논문에서 제시한 알고리즘은 출력오차(output error) 알고리즘으로 현재의 출력을 얻기 위해 과거의 적응필터의 출력을 이용하여 준최적(suboptimal) 최소자승 출력오차 적응필터를 실현할 수 있다.

Ⅱ장에 쌍선형 시스템이 안정하기 위한 조건을 제시하였으며, RPEM 알고리즘에 시변 칼만 필터링 기법을 도입한 수정된 새로운 RPEM 알고리즘에 관한 설명이 Ⅲ장에 나타나 있다. 이 알고리즘에 관한 여러가지 실험 결과가 Ⅳ장에 있으며 결론을 Ⅴ장에 나타냈다.

Ⅱ. 쌍선형 필터의 안정성 고찰

식 (1.2)로 주어지는 쌍선형 필터는 다음과 같이 간단한 벡터형태로 나타낼 수 있다.

$$\begin{aligned} \underline{x}(n) &= [A + B(n-1)]\underline{x}(n-1) + C u(n), \\ \underline{x}_0 &= \underline{x}(0) \end{aligned} \quad (2.1)$$

여기서

$$\underline{x}(n) = [y(n), y(n-1), \dots, y(n-N+2)]^T \quad (2.2)$$

$$\underline{u}(n) = [u(n), u(n-1), \dots, u(n-N+1)]^T \quad (2.3)$$

$$A = \begin{bmatrix} a_1 & a_2 & \cdots & a_{n-2} & a_{n-1} \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \quad (2.4)$$

$$b(n-1) = \sum_{j=0}^{N-1} B_j u(n-j) \quad (2.5)$$

$$C = [c_0, c_1, \dots, c_{N-1}] \quad (2.6)$$

$$b_j = \begin{bmatrix} b_{1j} & b_{2j} & \cdots & b_{N-1j} \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \quad (2.7)$$

$$c_j = [c_j, 0, \dots, 0]^T \quad (2.8)$$

이다. 본 장에서는 먼저 쌍선형 필터가 안정하기 위한 필요충분조건으로서 쌍선형 필터의 평균자승 안정에 대해 간단히 서술하고, 쌍선형 필터가 안정하기 위한 입력신호의 크기를 구하는 방법을 제시하고자 한다.

1. 쌍선형 필터의 평균자승 안정성(mean square stability).

본 절에서는 식 (2.1)로 주어지는 쌍선형 시스템이 안정하기 위한 조건을 구하기 위해 참고문헌 [8, 10]에서 제안한 평균자승 안정에 관해 설명하고자 한다. 먼저 다음과 같이 가정하자.

가정 1.

초기치 x_0 는 입력 $\{u(n); n \geq 1\}$ 과 관계없는 2 차 랜덤 벡터이다.

가정 2.

입력 $\{u(n); n \geq 1\}$ 은 넓은 의미의 2 차 정상 랜덤 수열(wide sense second order stationary random sequence)이다.

또 상태벡터 $x(n)$ 의 평균과 공분산행렬을 각각 다음과 같이 정의하자.

$$\underline{q}(n) = E\{x(n)\} \quad (2.9)$$

$$\underline{Q}(n) = E\{x(n)x^T(n)\} \quad (2.10)$$

벡터 $\underline{q}(n)$ 과 행렬 $\underline{Q}(n)$ 이 임의의 초기치 x_0 와 입

력 $u(n)$ 에 대해서도 항상 수렴하면 시스템은 안정이다. 이제 평균자승 안정(mean square stable : MSS)을 다음과 같이 정의하자.

정의

가정 1과 가정 2를 만족하는 임의의 초기조건 x_0 과 입력 $\{u(n); n \geq 1\}$ 에 대해 다음을 만족하는 벡터 $\underline{q}(n)$ 와 행렬 \underline{Q} 가 존재하면 식 (2.1)로 주어지는 시스템은 평균자승 안정이다.

$$a) \|\underline{q}(n) - \underline{q}\| \Rightarrow 0 \quad \text{as } n \Rightarrow \infty \quad (2.11)$$

$$b) \|\underline{Q}(n) - \underline{Q}\| \Rightarrow 0 \quad \text{as } n \Rightarrow \infty \quad (2.12)$$

가정 1과 가정 2로 부터 다음과 같은 식이 성립한다.

$$E\{u(n)x^T(n-1)\} = E\{u(n)\}E\{x^T(n-1)\} \quad (2.13)$$

$$E\{B(n-1)x(n-1)x^T(m-1)\} = E\{B(n-1)\}E\{x(n-1)\}E\{x^T(m-1)\} \quad 1 \leq m \leq n \quad (2.14)$$

$$E\{B(n-1)x(n-1)x^T(m-1)B^T(n-1)\} = E\{B(n-1)E\{x(n-1)x^T(m-1)\}B^T(n-1)\} \quad (2.15)$$

여기서 다음과 같이 정의하자.

$$\rho_k = E\{u(n-k)\} = \rho \quad (2.16)$$

$$\gamma_{kl} = E\{u(n-k)u(n-l)\} = \rho^2 \quad (2.17)$$

$$\underline{r} = E\{u(n)\} \quad (2.18)$$

$$R = E\{u(n)u^T(n)\} \quad (2.19)$$

$$F = E\{A + B(n)\} = A + \sum_{j=0}^{N-1} \rho_j B_j = A + \rho \sum_{i=0}^{N-1} B_i \quad (2.20)$$

상태벡터 $x(n)$ 의 평균 $\underline{q}(n)$ 과 공분산 $\underline{Q}(n)$ 은 다음과 같이 된다.

$$\underline{q}(n) = F \underline{q}(n-1) + C \underline{r} \quad (2.21)$$

$$\underline{Q}(n) = (F \otimes F) \underline{Q}(n-1) + \sum_{j=0}^{N-1} \sum_{l=0}^{N-1} \gamma_{jl} (B_j \otimes B_l) \underline{Q}(n-1) + F \underline{q}(n-1) \underline{r}^T C^T + C \underline{r} \underline{r}^T C^T + C R C^T \quad (2.22)$$

여기서 \otimes 는 Kronecker 곱이다. 또 다음과 같이 정의하자.

$$A = (F \otimes F) + \sum_{j=0}^{N-1} \sum_{l=0}^{N-1} \gamma_{jl} (B_j \otimes B_l) \quad (2.23)$$

$$P(X) = F \underline{Q}(n-1) \underline{r}^T C^T \quad (2.24)$$

$$V(X) = P(X) + P^T(X) + C R C^T \quad (2.25)$$

그러면 $Q(n)$ 은 다음과 같이 된다.

$$Q(n) = AQ(n-1) + V[q(n-1)] \quad (2.26)$$

위의 식으로 부터 쌍선형 시스템이 안정하기 위한 필요충분 조건은 다음과 같이 된다.

$$\rho_{\max}(A) < 1 \quad (2.27)$$

여기서 $\rho \max(A)$ 는 A행렬 고유치중 최대값이다. 만약 입력 $u(n)$ 이 백색잡음일 경우 행렬 A 은 다음과 같이 된다.

$$A = (A \otimes A) + \sum_{j=0}^{N-1} \sigma^2 (B_j \otimes B_j) \quad (2.28)$$

2. 쌍선형 필터가 안정하기 위한 입력조건

앞에서 설명한 쌍선형 시스템이 평균자승 안정하기 위한 필요충분조건은 쌍선형 시스템이 안정하기 위한 완전한 조건을 보여주나 Kronecker 곱에 의해 행렬의 차수가 커지고 이에 따라 행렬의 고유치를 계산하는 데 많은 계산량을 필요로 하기 때문에 실제로 쌍선형 시스템의 안정도를 조사하는 데 적용하기가 어렵다. 본 절에서는 제어분야에서 선형 시스템이 안정하기 위한 견실한 영역(robustness bound)을 구하는 방법을 이용하여 비교적 간단한 방법으로 쌍선형 필터가 안정하기 위한 입력의 크기를 구하는 방법에 대해 논의하고자 한다. 식 (2.1)로 부터 쌍선형 시스템은 다음과 같이 선형 섭동(perturbation)을 갖는 선형 동적(dynamic) 시스템으로 생각할 수 있다.

$$\underline{x}(n) = [A + E]\underline{x}(n-1)C\underline{u}(n) \quad (2.29)$$

여기서 는 섭동행렬로서 다음과 같다.

$$E = \sum_{j=0}^{N-1} B_j u(n-j) = \sum_{j=0}^{N-1} B_j K_j \quad (2.30)$$

식 (2.29)로 주어지는 시스템이 안정하기 위한 섭동행렬 E(정확히 표현하면 K_j)의 범위를 구하는 방법은 여러가지가 있으나²⁰⁻²⁴ 여기서는 참고문헌 [20]의 방법을 사용하여 보자. 만일 행렬 P가 안정이고, 행렬 P가 Lyapunov 방정식의 해이며, 행렬 P_1, P_2, \dots, P_N 가 다음과 같다고 하자.

$$PA + A^T P + 2 \cdot I = 0 \quad (2.31)$$

$$P_j = (E_j^T P + P E_j) / 2 \quad (2.32)$$

$$P_0 = [P_0, P_1, \dots, P_{N-1}] \quad (2.33)$$

쌍선형 시스템이 안정하기 위한 조건은 다음과 같다.

$$\sum_{j=0}^{N-1} |K_j| \sigma_{\max} P_j < 1 \quad (2.34)$$

여기서

$$|K_j| = |u(n-j)| \leq M_u \quad (2.35)$$

라 하면 쌍선형 시스템이 안정하기 위한 입력의 크기 M_u 는 다음과 같이 된다.

$$M_u < \frac{1}{\sum_{j=0}^{N-1} \sigma_{\max} P_j} < 1 \quad (2.37)$$

여기서 행렬 P_j 는 행렬 A와 같은 차원을 갖기 때문에 행렬의 고유치를 구하기가 쉬움을 알 수 있다.

III. 적용 쌍선형 필터의 RPEM 알고리즘

쌍선형 필터는 다음과 같이 간단한 벡터형태로 나타낼 수 있다

$$y(n) = \underline{\theta}^T \underline{\phi}(n) \quad (3.1)$$

여기서 $\underline{\phi}(n)$ 과 $\underline{\theta}$ 는 상태벡터와 계수벡터로 각각 다음과 같다.

$$\underline{\phi}(n) = [y(n-1), \dots, y(n-N+1), y(n-1)u(n), \dots, y(n-N+1)u(n-N+1), u(n), \dots, u(n-N+1)]^T \quad (3.2)$$

$$\underline{\theta} = [a_1, a_{N-1}, b_1, 0, b_{N-1}, \dots, c_0, c_{N-1}]^T \quad (3.3)$$

이제 시간 n에서의 계수벡터 $\underline{\theta}(n)$ 을 이용하여 y(n)을 추정된 값을 $\hat{y}(n)$ 이라 하면 $\hat{y}(n)$ 은 다음과 같다.

$$\hat{y}(n) = \underline{\theta}^T(n) \underline{\phi}(n) \quad (3.4)$$

계수벡터 $\underline{\theta}(n)$ 을 구하는 방법은 많이 있으나 여기서는 RPEM(recursive prediction error method) 알고리즘에 대해 고찰하여 보자. RPEM 알고리즘은 다음과 같다.¹²⁵⁾

$$R^{-1}(n) = \lambda^{-1} R^{-1}(n-1) - \frac{\lambda^{-2} R^{-1}(n-1) \underline{\phi}(n) \underline{\phi}^T(n) R^{-1}(n-1)}{1 + \lambda^{-1} \underline{\phi}^T(n) R^{-1}(n-1) \underline{\phi}(n)} \quad (3.5)$$

$$\underline{\theta}(n) = \underline{\theta}(n-1) + R^{-1}(n) \underline{\phi}(n) [y(n) - \underline{\theta}^T(n-1) \underline{\phi}(n)] \quad (3.6)$$

여기서 $\theta(n)$ 은 계수벡터 θ 의 추정치이고, 상태벡터 $\phi(n)$ 과 경사벡터 $\varphi(n)$ 은 각각 다음과 같이 정의된다.

$$\phi(n) = [\hat{y}(n-1), \dots, \hat{y}(n-N+1), \hat{y}(n-1)u(n), \dots, \hat{y}(n-N+1)u(n-N+1), u(n), \dots, u(n-N+1)]^T \quad (3.7)$$

$$\varphi(n) = \frac{\partial \hat{y}(n)}{\partial \theta} \quad (3.8)$$

쌍선형 필터에서 경사벡터 $\varphi(n)$ 은 다음과 같이 된다.

$$\varphi(n) = \sum_{i=1}^{N-1} \left[a_i(n-1) + \sum_{j=0}^{N-1} b_{ij}(n-1)u(n-j) \right] \varphi(n-i) + \phi(n) \quad (3.9)$$

일반적으로 쌍선형 필터는 입력신호에 따라 필터가 발산하는 경우가 있으며, RPEM 알고리즘 역시 입력신호가 커지면 발산하게 된다. 이러한 문제를 극복하기 위하여 본 논문에서는 RPEM 알고리즘에 시변 칼만필터링 기법을 적용하여 새로운 수정된 RPEM 알고리즘을 개발하고자 한다. 칼만필터링 기법을 적용하기 위해서는 쌍선형 필터를 상태 방정식 형태로 나타내야 한다. 쌍선형 필터의 상태방정식 형태는 다음과 같다.

$$\hat{x}(n+1) = A\hat{x}(n) + \sum_{j=0}^{N-1} B_j \hat{x}(n)u(n-j+1) + \sum_{j=0}^{N-1} c_j u(n-j+1) + v(n) \quad (3.10)$$

$$= \left[A + \sum_{j=0}^{N-1} B_j u(n-j+1) \right] \hat{x}(n) + \sum_{j=0}^{N-1} c_j u(n-j+1) + v(n)$$

$$y(n) = h^T \hat{x}(n) + e(n) \quad (3.11)$$

여기서

$$\hat{x}(n) = [y(n), y(n-1), \dots, y(n-N+2)]^T \quad (3.12)$$

$$h = [1, 0, \dots, 0]^T \quad (3.13)$$

이고, A, B_i, c 는 각각 식 (2.4), 식 (2.7), 식 (2.8)로 주어지며, $v(n), e(n)$ 은 서로 독립인 랜덤잡음으로 평균이 0이고 공분산 행렬과 분산이 각각 다음과 같이 주어진다.

$$R_v = E[v(n)v^T(n)] \quad (3.14)$$

$$r^2 = E[e(n)^2] \quad (3.15)$$

이제 상태방정식 형태의 RPEM 알고리즘을 구해 보자. 출력 $y(n)$ 의 추정치 $\hat{y}(n)$ 은 다음과 같은 상태 방정식을 통해 구해진다.

$$\hat{x}(n+1) = \left[A(n) + \sum_{j=0}^{N-1} B_j u(n-j+1) \right] \hat{x}(n) + \sum_{j=0}^{N-1} c_j(n) u(n-j+1) = F(n)\hat{x}(n) + c_j(n) \quad (3.16)$$

$$\hat{y}(n) = h^T \hat{x}(n) \quad (3.17)$$

여기서 $F(n)$ 와 $c(n)$ 은 각각 다음과 같다.

$$F(n) = A(n) + \sum_{j=0}^{N-1} B_j(n)u(n-j+1) \quad (3.18)$$

$$c(n) = \sum_{j=0}^{N-1} c_j(n)u(n-j+1) \quad (3.19)$$

경사벡터 $\varphi(n)$ 의 k 번째 요소를 $\varphi_{k(n)}$ 이라 하면 $\varphi_{k(n)}$ 은 다음과 같은 방법으로 구할 수 있다.

$$\varphi_{k(n)} = \frac{\partial \hat{y}(n)}{\partial \theta_k} = h^T \frac{\partial \hat{x}(n)}{\partial \theta_k} \quad (3.20)$$

여기서 $\frac{\partial \hat{x}(n)}{\partial \theta_k}$ 은 다음과 같은 식에서 순환적으로 구해진다.

$$\frac{\partial \hat{x}(n+1)}{\partial \theta_k} = F(n) \frac{\partial \hat{x}(n)}{\partial \theta_k} + \frac{\partial F(n)}{\partial \theta_k} \hat{x}(n) + \frac{\partial c(n)}{\partial \theta_k} \quad (3.21)$$

여기서 $\frac{\partial F(n)}{\partial \theta_k}$ 과 $\frac{\partial c(n)}{\partial \theta_k}$ 는 각각 다음과 같이 주어진다.

$\theta_k = a_p(n)$ 일 때

$$\left[\frac{\partial F(n)}{\partial \theta_k} \right]_{ij} = \begin{cases} 1, & i=1, j=p \\ 0, & \text{otherwise} \end{cases} \quad (3.22)$$

$$\left[\frac{\partial c(n)}{\partial \theta_k} \right]_i = 0, \quad \text{for all } i \quad (3.23)$$

$\theta_k = b_{pq}(n)$ 일 때

$$\left[\frac{\partial F(n)}{\partial \theta_k} \right]_{ij} = \begin{cases} u(n-q), & i=1, j=p \\ 0, & \text{otherwise} \end{cases} \quad (3.24)$$

$$\left[\frac{\partial c(n)}{\partial \theta_k} \right]_i = 0, \quad \text{for all } i \quad (3.25)$$

$\theta_k = c_q(n)$ 일 때

$$\left[\frac{\partial F(n)}{\partial \theta_k} \right]_{ij} = 0, \quad \text{for all } i, j \quad (3.26)$$

$$\left[\frac{\partial c(n)}{\partial \theta_k} \right]_i = \begin{cases} u(n-q), & i=1 \\ 0, & \text{otherwise} \end{cases} \quad (3.27)$$

RPEM 알고리즘이 안정(수렴)하기 위한 조건을 생각하여 보자. 선형시스템($B \neq 0$ 인 경우)에서는 행렬 A 의 고유치의 절대값이 1 보다 작으면 된다. 그러나 쌍선형 시스템에서는 행렬 $F(n)$ 에 시간에 따라 변하는 입력신호 $u(n)$ 이 포함되어 있기 때문에 알고리즘이 발산하는 경우가 생긴다. 즉 쌍선형 시스템은 시스템의 행렬이 $F(n)$ 이고 이 행렬이 시간에 따라 변하는 시변 선형 시스템으로 볼 수 있다. 이런 알고리즘을 안정화시키기 위하여 시변 칼만필터링의 기법을 적용하여 보자.^[25] 쌍선형필터의 출력의 추정치 $\hat{y}(n)$ 의 값은 다음과 같은 칼만필터에서 얻어진다.

$$\hat{x}(n+1) = F(n)\hat{x}(n) + c(n) + k(n)[y(n) - h^T \hat{x}(n)] \quad (3.28)$$

$$\hat{y}(n) = h^T \hat{x}(n) \quad (3.29)$$

$$k(n) = [F(n)P(n)h]s^{-1}(n) \quad (3.30)$$

$$P(n+1) = F(n)P(n)F^T(n) + R_1 - k(n)s(n)k^T(n) \quad (3.31)$$

$$s(n) = h^T P(n)h + r^2 \quad (3.32)$$

$R_1=0$ 일 때는 $P(n)=0$ 이 Riccati 방정식의 해가 되며, 결과적으로 $k(n)=0$ 이 된다. (칼만필터링을 고려하지 않은 단순한 RPEM 알고리즘이 된다.) 만약 $u(n)$ 이 동적특성을 나타내는 행렬 $F(n)$ 이 안정하도록 하는 그런 수열이라 하면 Riccati 방정식은 이 해에 수렴하게 된다. 그러나 $F(n)$ 불안정하게 되면 해 $k(n)$ 은 다음과 같은 동적특성을 나타내는 행렬이 안정해지도록 하여 준다.^[26]

$$F(n) - k(n)h^T = A + \sum_{j=0}^{N-1} B_j u(n-j) - k(n)h^T \quad (3.33)$$

따라서 위의 식을 이용하면 알고리즘이 좀더 안정하게 된다. $R_1 = 0$ 일 때는 수치적인 이유로 인해 $R_1 = \delta I$ 로 놓는 것이 바람직하다

$k(n)$ 이 계수벡터 θ 의 영향을 받기 때문에 식 (3.22)는 다음과 같이 수정되어야 한다.

$$\begin{aligned} \frac{\partial \hat{x}(n+1)}{\partial \theta_k} &= F(n) \frac{\partial \hat{x}(n)}{\partial \theta_k} + \frac{\partial F(n)}{\partial \theta_k} \hat{x}(n) + \frac{\partial c(n)}{\partial \theta_k} \\ &- k(n)h^T \frac{\partial \hat{x}(n)}{\partial \theta_k} + \frac{\partial k(n)}{\partial \theta_k} [y(n) - h^T \hat{x}(n)] \end{aligned} \quad (3.34)$$

여기서 $\frac{\partial k(n)}{\partial \theta_k}$ 은 다음과 같이 구해진다.

$$\frac{\partial k(n)}{\partial \theta_k} = \frac{\partial F(n)}{\partial \theta_k} P(n)h s^{-1}(n) + F(n) \frac{\partial P(n)}{\partial \theta_k} h s^{-1} - k(n)s^{-1}(n) \quad (3.35)$$

$$\frac{\partial s(n)}{\partial \theta_k} = h^T \frac{\partial P(n)}{\partial \theta_k} h \quad (3.36)$$

$$\begin{aligned} \frac{\partial P(n+1)}{\partial \theta_k} &= \frac{\partial F(n)}{\partial \theta_k} P(n)F^T(n) + F(n) \frac{\partial P(n)}{\partial \theta_k} F^T(n) + F(n)P(n) \frac{\partial F^T(n)}{\partial \theta_k} \\ &- \frac{\partial k(n)}{\partial \theta_k} s(n)k^T(n) - k(n) \frac{\partial s(n)}{\partial \theta_k} k^T(n) - k(n)s(n) \frac{\partial k^T(n)}{\partial \theta_k} \end{aligned} \quad (3.37)$$

상태 방정식 형태로 나타낸 기존의 RPEM 알고리즘이 표 1에 시변 칼만 필터링 기법을 적용하여 얻은 수정된 새로운 RPEM 알고리즘이 표 2에 나타나 있다.

표 1. RPEM 알고리즘

Table 1. RPEM Algorithm.

```

Initialization
x̂(0)=Ω
g(0)=Ω
R-1(0)=δI
DO (T-1.4) for k=1,2,...,N2+N-1
    ∂x̂(0)/∂θk = Ω
Iteration Procedure
DO (T-1.5)-(T-1.14) for n=1,2,...
    g(n)=[ŷ(n-1), ..., ŷ(n-N+1), ŷ(n-1)u(n), ..., ŷ(n-N+1)u(n-N+1),
           u(n), ..., u(n-N+1)]T
    F(n-1)=A(n-1)+∑j=0N-1Bj(n-1)u(n-j)
    c(n-1)=∑j=0N-1cj(n-1)u(n-j)
    x̂(n)=F(n-1)x̂(n-1)+c(n-1)
    ŷ(n)=hTx̂(n)
DO (T-1.10)-(T-1.11) for k=1,2,...,N2+N-1
    set ∂F(n-1)/∂θk} = ∂c(n-1)/∂θk} +
        ∂x̂(n)/∂θk} = F(n-1) ∂x̂(n-1)/∂θk} + ∂F(n-1)/∂θk} x̂(n-1) + ∂c(n-1)/∂θk}
    g(n)=[hT ∂x̂(n)/∂θ1}, hT ∂x̂(n)/∂θ2}, ..., hT ∂x̂(n)/∂θN} ]T
    R-1(n)=λ-1R-1(n-1) - λ-2R-1(n-1)g(n)gT(n)R-1(n-1)
        / (1+λ-1gT(n)R-1(n-1)g(n))
    g(n)=g(n-1)+R-1g(n)[y(n)-g(n-1)g(n)]
    
```

식 (3.21)-(3.26)으로 주어짐

표 2. 칼만필터를 포함한 RPEM 알고리즘

Table 2. RPEM Algorithm with Kalman Filter.

```

Initialization
x̂(0)=Ω
g(0)=Ω
R-1(0)=δI
P(0)=R(0)
DO (T-2.5)-(T-2.6) for k=1,2,...,N2+N-1
    ∂x̂(0)/∂θk} = Ω
    ∂P(0)/∂θk} = Ω
    
```

Iteration Procedure

DO (T-2.7)-(T-2.22) for $n=1,2,\dots$
 $\mathbf{d}(n)=[y(n-1), \dots, y(n-N+1), \hat{y}(n-1)u(n), \dots, \hat{y}(n-N+1)u(n-N+1), u(n), \dots, u(n-N+1)]^T$
 $F(n-1)=A(n-1)+\sum_{j=0}^{N-1} B_j(n-1)u(n-j)$
 $\mathbf{c}(n-1)=\sum_{j=0}^{N-1} \mathbf{c}_j(n-1)u(n-j)$
 $\mathbf{s}(n)=\mathbf{h}^T P(n-1)\mathbf{h}+r_2$
 $\mathbf{k}(n-1)=F(n-1)P(n-1)\mathbf{h} s^{-1}(n-1)$
 $\hat{\mathbf{x}}(n)=F(n-1)\hat{\mathbf{x}}(n-1)+\mathbf{c}(n-1)+\mathbf{k}(n-1)[y(n-1)-\mathbf{h}^T \hat{\mathbf{x}}(n-1)]$
 $\hat{y}(n)=\mathbf{h}^T \hat{\mathbf{x}}(n)$
 DO (T-2.14)-(T-2.18) for $k=1,2,\dots,N^2-N-1$
 set $\frac{\partial F(n-1)}{\partial \theta_k}, \frac{\partial \mathbf{c}(n-1)}{\partial \theta_k}$ †
 $\frac{\partial \mathbf{s}(n-1)}{\partial \theta_k} = \mathbf{h}^T \frac{\partial F(n-1)}{\partial \theta_k} \mathbf{h}$
 $\frac{\partial \mathbf{k}(n-1)}{\partial \theta_k} = -\frac{\partial F(n-1)}{\partial \theta_k} P(n-1)\mathbf{h} s^{-1}(n-1) + F(n-1) \frac{\partial P(n-1)}{\partial \theta_k} \mathbf{h} s^{-1}(n-1) - \mathbf{k}(n-1)s^{-1}(n-1) \frac{\partial s(n-1)}{\partial \theta_k}$
 $\frac{\partial \hat{\mathbf{x}}(n)}{\partial \theta_k} = F(n-1) \frac{\partial \hat{\mathbf{x}}(n-1)}{\partial \theta_k} + \frac{\partial F(n-1)}{\partial \theta_k} \hat{\mathbf{x}}(n-1) + \frac{\partial \mathbf{c}(n-1)}{\partial \theta_k} - \mathbf{k}(n-1)\mathbf{h}^T \frac{\partial \hat{\mathbf{x}}(n-1)}{\partial \theta_k} - \frac{\partial \mathbf{k}(n-1)}{\partial \theta_k} [d(n-1)-\mathbf{h}^T \hat{\mathbf{x}}(n-1)]$
 $\frac{\partial P(n)}{\partial \theta_k} = \frac{\partial F(n-1)}{\partial \theta_k} P(n-1)F(n-1) + F(n-1) \frac{\partial P(n-1)}{\partial \theta_k} F^T(n-1) + F(n-1)P(n-1) \frac{\partial F(n-1)}{\partial \theta_k} - \frac{\partial \mathbf{k}(n-1)}{\partial \theta_k} s(n) \mathbf{k}^T(n) - \mathbf{k}(n-1) \frac{\partial s(n-1)}{\partial \theta_k} \mathbf{k}^T(n-1) - \mathbf{k}(n-1)s(n-1) \frac{\partial \mathbf{k}(n-1)}{\partial \theta_k}$
 $\mathbf{w}(n)=[\mathbf{h}^T \frac{\partial \hat{\mathbf{x}}(n)}{\partial \theta_1}, \mathbf{h}^T \frac{\partial \hat{\mathbf{x}}(n)}{\partial \theta_2}, \dots, \mathbf{h}^T \frac{\partial \hat{\mathbf{x}}(n)}{\partial \theta_N}]^T$
 $R^{-1}(n)=\lambda^{-1}R^{-1}(n-1) - \frac{\lambda^{-2}R^{-1}(n-1)\mathbf{w}(n)\mathbf{w}^T(n)R^{-1}(n-1)}{1+\lambda^{-2}\mathbf{w}^T(n)R^{-1}(n-1)\mathbf{w}(n)}$
 $\hat{\mathbf{w}}(n)=\hat{\mathbf{w}}(n-1)+R^{-1}(n)\mathbf{w}(n)[y(n)-\hat{\mathbf{w}}(n-1)\mathbf{d}(n)]$
 $P(n)=F(n-1)P(n-1)F^T(n-1)+R_1 - \mathbf{k}(n-1)s(n-1)\mathbf{k}^T(n-1)$

식 (3.21)-(3.26)으로 주어짐

IV. 실험결과 및 검토

새로운 알고리즘의 성능을 평가하기 위하여 미지의 쌍선형 시스템의 식별 실험을 행하였다. 적응필터의 입력신호 $x(n)$ 은 전달함수가 다음과 같은 저역통과필터에 평균이 0 인 백색 가우시안 잡음을 통과하여 얻었고 분산이 1 과 3 이 되도록 조정하였다.

$$H(z) = \frac{1}{1-1.6z^{-1}+0.95z^{-2}} \quad (4.1)$$

원하는 응답신호 $y(n)$ 은 표 3에 있는 계수를 가진 쌍선형필터의 출력에 잡음을 섞었다. 이 때 사용된 관측잡음은 평균이 0인 백색 의사 가우시안 잡음으로 입력신호와는 상관관계가 없다. 모든 실험결과는 50 번의 독립된 실험의 양상불 평균치이다.

그림 1 - 그림 4는 각각 서로 다른 잡음레벨에 대해 입력신호의 분산이 각각 1, 3일 때의 두 알고리즘의 평균자승오차를 나타낸 것이다. (편의상 기존의 RPEM 알고리즘을 RPEM 1으로 새로운 RPEM 알

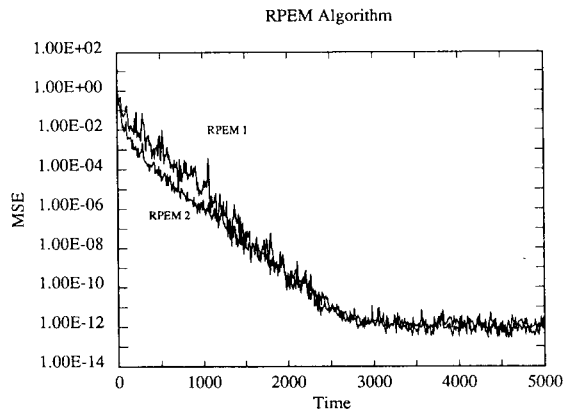


그림 1. SNR = ∞, σ_u = 1일 때의 학습곡선
 Fig. 1. Learning curves when SNR = ∞, σ_u = 1.

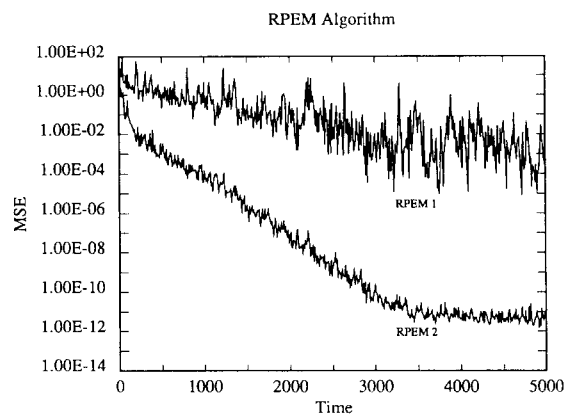


그림 2. SNR = ∞, σ_u = 3일 때의 학습곡선.
 Fig. 2. Learning curves when SNR = ∞, σ_u = 3.

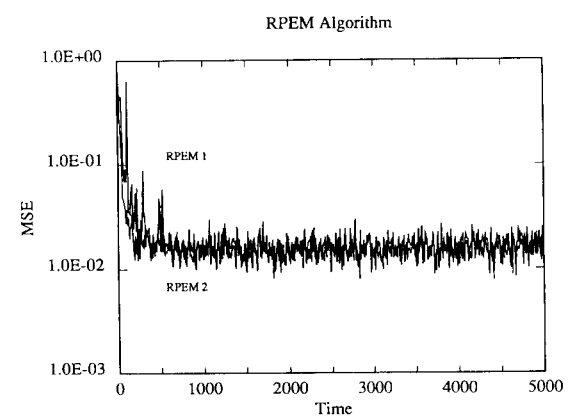


그림 3. SNR = 30, dB, σ_u = 1일 때의 학습곡선
 Fig. 3. Learning curves when SNR = 30, dB, σ_u = 1.

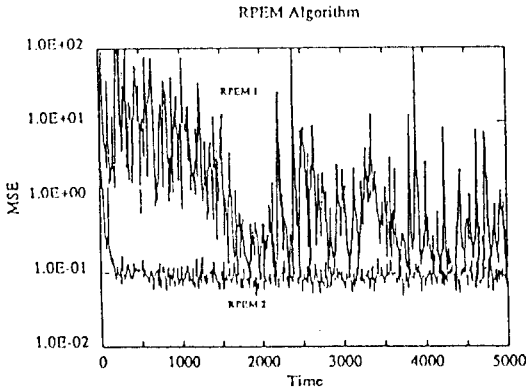


그림 4. SNR=30 dB, $\sigma_u = 3$ 일 때의 학습곡선
 Fig. 4. Learning curves when SNR=30 dB, $\sigma_u = 3$.

고리들을 RPEM 2로 나타내었음) 모두 신호대 잡음비 (SNR)가 ∞ , 30 dB 인 두 개의 서로 다른 잡음 레벨에 대해 실험이 행해졌다. 이 그림들로부터 입력신호가 작은 경우 기존의 RPEM 알고리즘과 새로운 RPEM 알고리즘이 거의 비슷하나, 입력신호가 큰 경우 새로운 RPEM 알고리즘이 기존의 RPEM 알고리즘에 비해 훨씬 더 안정함을 알 수 있다.

두 알고리즘의 성능차이는 평균 계수궤적을 보면 더욱 잘 알 수 있다. 신호대 잡음비가 ∞ , 30 dB이고 입력신호의 분산이 1, 3일 때의 두 알고리즘에 대한 계수 $b_{1,1}(n)$, $C_2(n)$ 의 평균궤적이 각각 그림 5 - 그림 12에 나타나 있다.

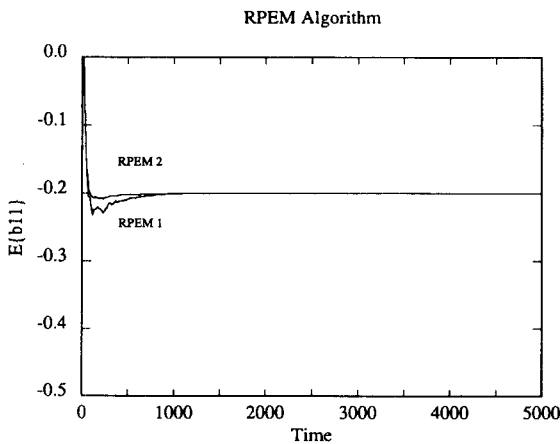


그림 5. SNR= ∞ , $\sigma_u = 1$ 일 때의 계수 $b_{1,1}(n)$ 의 평균궤적
 Fig. 5. Mean trajectories of coefficient $b_{1,1}(n)$ when SNR= ∞ , $\sigma_u = 1$.

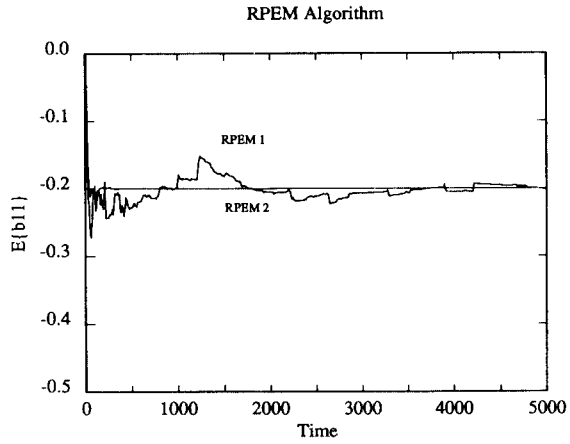


그림 6. SNR= ∞ , $\sigma_u = 3$ 일 때의 계수 $b_{1,1}(n)$ 의 평균궤적
 Fig. 6. Mean trajectories of coefficient $b_{1,1}(n)$ when SNR= ∞ , $\sigma_u = 3$.

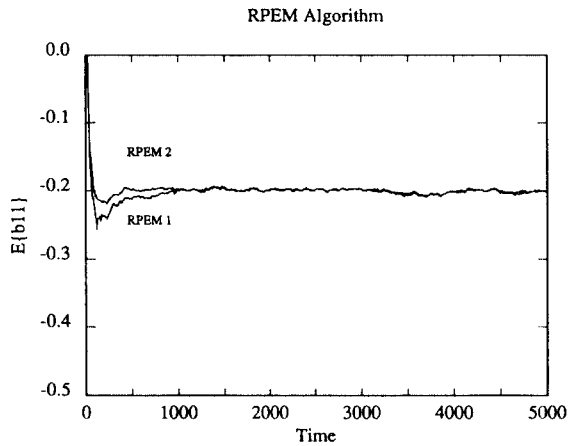


그림 7. SNR= 30 dB, $\sigma_u = 1$ 일 때의 계수 $b_{1,1}(n)$ 의 평균궤적
 Fig. 7. Mean trajectories of coefficient $b_{1,1}(n)$ when SNR=30 dB, $\sigma_u = 1$.

신호가 작은 경우 두 알고리즘 모두 최적치에 안정하게 수렴하나, 입력신호가 커지면 기존의 RPEM 알고리즘은 불안정하게 수렴하며 특히 잡음이 있는 경우 수렴특성은 아주 불안정하게 된다. 그러나 새로운 RPEM 알고리즘은 입력신호가 커져도 역시 안정하게 수렴함을 알 수 있다.

표 3. 실험에 사용된 미지의 쌍선형 시스템의 계수

Table 3. Coefficients of the unknown bilinear system used in the experiments.

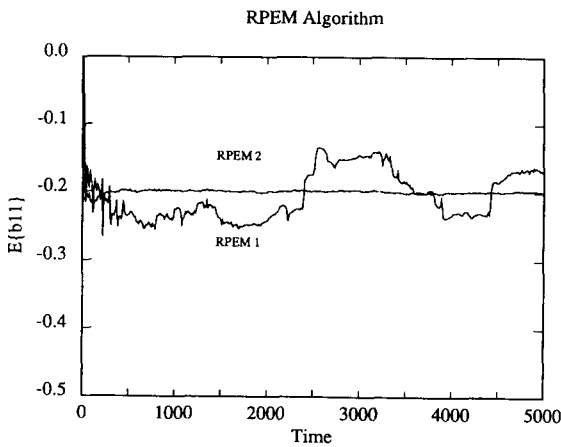


그림 8. SNR=30 dB, $\sigma_u=3$ 일 때의 계수 $b_{1,1}(n)$ 의 평균궤적
Fig. 8. Mean trajectories of coefficient $b_{1,1}(n)$ when SNR=30 dB, $\sigma_u=3$.

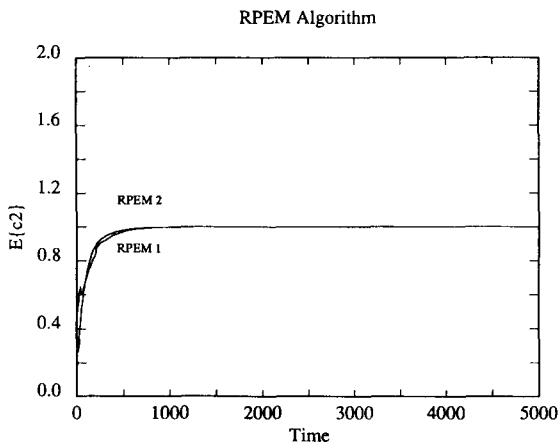


그림 9. SNR= ∞ , $\sigma_u=1$ 일 때의 계수 $c_2(n)$ 의 평균궤적
Fig. 9. Mean trajectories of coefficient $c_2(n)$ when SNR= ∞ , $\sigma_u=1$.

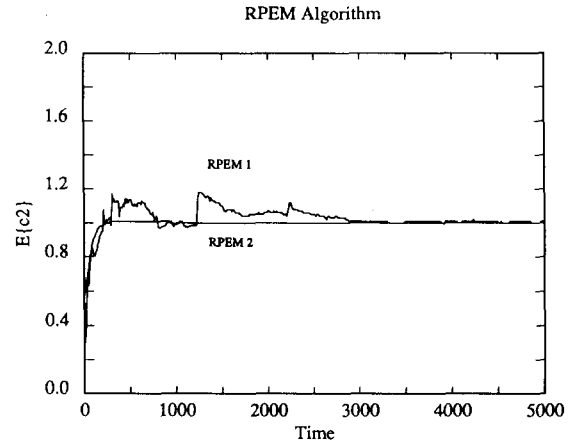


그림 10. SNR= ∞ , $\sigma_u=3$ 일 때의 계수 $c_2(n)$ 의 평균궤적.
Fig. 10. Mean trajectories of coefficient $c_2(n)$ when SNR= ∞ , $\sigma_u=3$.

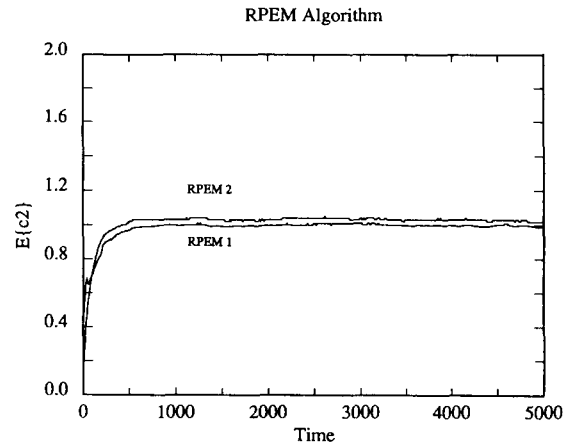


그림 11. SNR=30 dB, $\sigma_u=1$ 일 때의 계수 $c_2(n)$ 의 평균궤적
Fig. 11. Mean trajectories of coefficient $c_2(n)$ when SNR=30 dB, $\sigma_u=1$.

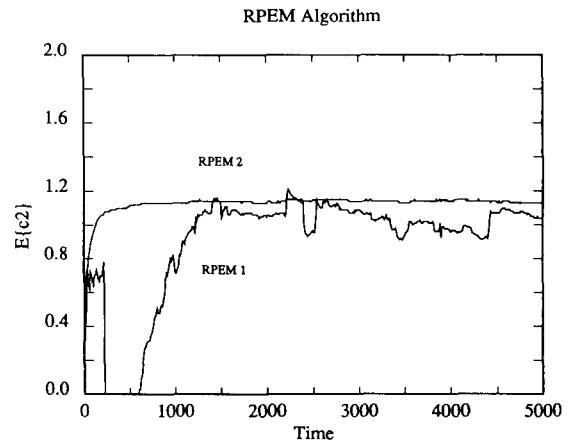


그림 12. SNR=30 dB, $\sigma_u=3$ 일 때의 계수 $c_2(n)$ 의 평균궤적
Fig. 12. Mean trajectories of coefficient $c_2(n)$ when SNR=30 dB, $\sigma_u=3$.

V. 결론

본 논문에서는 쌍선형 필터가 안정하기 위한 필요충분조건으로서 쌍선형 필터의 평균자승 안정에 대해 간단히 서술하였으며, 쌍선형 필터가 안정하기 위한 입력신호의 크기를 구하는 방법을 제시하였다. 본 논문에서 제안한 방법은 제어분야에서 선형 시스템이 안정하기 위한 견실한 영역(robustness bound)를 구하는 방법을 이용한 것으로 비교적 간단한 방법으로 쌍선형 필터가 안정하기 위한 입력의 크기를 구할 수 있다.

또 쌍선형 시스템의 직접형 구조에 대한 RPEM (recursive prediction error method) 알고리즘에 시변 칼만 필터링 기법을 도입하여 입력신호가 어느 정도 클 경우에도 발산하지 않고 수렴특성이 좋은 수정된 RPEM 알고리즘을 제시하였다. 본 논문에서 제시한 알고리즘은 출력오차(output error) 알고리즘으로 현재의 출력을 얻기 위해 과거의 적응필터의 출력을 이용하며 준최적(suboptimal) 최소자승 출력오차 적응필터를 실현할 수 있다.

한편 본 논문에서 제시한 RPEM 알고리즘 역시 LS 계열의 알고리즘이기 때문에 다른 TDL 필터의 알고리즘과 마찬가지로 수치적 문제(numerical problem)가 있을 것으로 생각된다. 이러한 문제를 해결하기 위하여 격자 알고리즘에 시변 칼만 필터링 기법을 적용하는 문제에 대한 연구가 현재 진행되고 있다.

參 考 文 獻

[1] W. J. Rugh, *Nonlinear System Theory: The Volterra/ Wiener Approach*. Baltimore: The John Hopkins Univ. Press. 1981.

[2] M. Schetzen, *The Volterra Wiener Theory of the Nonlinear Systems*. New York: Wiley and Sons. 1980.

[3] T. Koh and E. J. Power, "Second-order Volterra filtering and its application to nonlinear system identification." *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, no. 6, pp. 1445-1455, Dec. 1985.

[4] V. J. Mathews and J. Lee, "A fast least-squares second-order Volterra filter."

Proc. Int. Conf. Acoust., Speech, Signal Processing (New York), 1988.

[5] M. A. Syed and V. J. Mathews, "Lattice and QR decomposition-based algorithms for recursive least squares adaptive nonlinear filters." *Proc. IEEE Int. Symp. Circuits and Systems* (New Orleans, Louisiana), May 1990.

[6] X. Y. Gao, W. M. Snelgrove, and D. A. Johns, "Nonlinear IIR adaptive filtering using a bilinear structure." *Proc. IEEE Int. Symp. Circuits and Systems* (Portland, Oregon), May 1989.

[7] Roger W. Brockett, "Volterra series and geometric control theory." *Automatica*, vol. 12, pp. 167-176, 1976.

[8] 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.

[9] C. S. Kubrusly, "On discrete stochastic bilinear systems stability." *J. Math. Anal. Appl.*, 113, pp. 36-58, 1986.

[10] C. S. Kubrusly, "Identification of discrete-time stochastic bilinear systems stability." *Int. J. Control*, vol. 33, no. 2, pp. 291-309, 1981.

[11] J. Liu, "On the existence of a general multiple bilinear time series." *J. Time Series Anal.*, vol. 10, no. 4, pp. 341-355, 1989.

[12] J. Liu, "A simple condition for the existence of some stationary bilinear time series." *J. Time Series Anal.*, vol. 10, no. 1, pp. 33-39, 1989.

[13] S. A. Billings and W. S. F. Voon, "Least square parameter estimation algorithms for nonlinear systems." *Int. J. System Sci.*, vol. 15, no. 6, pp. 601-615, 1984.

[14] X. Yang, R. R. Mohler, and R. M. Burton, "Adaptive suboptimal filtering of bilinear filtering." *Int. J. Control*,

- vol. 52, no 1, pp. 135-158, 1990.
- [15] H. K. Baik, V. J. Mathews, and R. T. Short, "Adaptive lattice bilinear filters," *SPIE Conf. Advanced Signal Processing Algorithm, Architecture, and Implementation* (SanDiego, California), 1990.
- [16] H. K. Baik and V. J. Mathews, "Adaptive lattice bilinear filters," to be appeared to *IEEE Trans. Acoust., Speech, Signal Processing*, June, 1993.
- [17] H. K. Baik and V. J. Mathews, "Adaptive algorithms for identifying recursive nonlinear systems," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing* (Toronto, Canada), May 1991
- [18] 백홍기, "적응 쌍선형 격자필터(I) - 쌍선형 격자구조", 전자공학회 논문지, 제 29 권, B 편, 제 1 호, pp. 26-33, 1 월, 1992 년
- [19] 백홍기, "적응 쌍선형 격자필터(II) - 최소자승 격자 알고리즘", 전자공학회 논문지, 제 29 권, B 편, 제 1 호, pp. 34-42, 1 월, 1992 년
- [20] K. Zhou and P. P. Khargonekar, "Stability robustness bound for linear state-space models with structured uncertainty," *IEEE Trans. Automat. Contr.*, vol. AC-32, no. 7, pp. 621-623, July, 1987
- [21] Y. K. Foo and Y. C. Soh, "Stability analysis of a family of matrices," *IEEE Trans. Automat. Contr.*, vol AC-35, no. 11, pp. 1257-1259, Nov. 1990
- [22] R. K. Yedavalli, "Improved measures of stability robustness for linear state space models," *IEEE Trans. Automat. Contr.*, vol. AC-30, no. 6, pp. 577-579, June, 1985
- [23] R. K. Yedavalli, "Stability analysis of interval matrices: another sufficient condition," *Int. J. Contr.*, vol. 43, no. 3, pp. 767-772, 1986
- [24] R. K. Yedavalli and Z. Liang, "Reduced conservatism in stability robustness bounds by state transformation," *IEEE Trans. Automat. Contr.*, vol. AC-31, no. 9, pp. 863-866, Sep. 1986
- [25] L. Ljung and T. Soderstrom, "Theory and Practice of Recursive Identification," Cambridge: The MIT Press, 1983
- [26] A. H. Jazwinsky, *Stochastic Processes and Filtering Theory*, New York: Academic Press, 1970.

著者紹介



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 PostDoc 연수. 주관심분야는 디지털 신호처리, 적응 신호처리, 비선형신호처리 분야임.



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 ~ 1992년 2월 현대정공 기술연구소 연구원 1992年 3月 ~ 현재 전북대학교 대학원 전자공학과 박사과정 재학중. 1992年 9月 - 현재 전북대학교 공과대학 전자공학과 조교. 주관심분야는 신호처리, 적응신호처리, 음성신호처리 분야임.



黃智源 (正會員)

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) 1991年 3月 ~ 현재 전북대학교 대학원 전자공학과 박사과정 재학중. 1992年 3月 ~ 현재 이리 농공 전문대학 전자계산학과 전임강사. 주관심분야는 신호처리, 적응신호처리, 음성신호처리, 비선형신호처리 분야임.