
2차경로의 온라인 모델링이 포함된 비선형 능동소음제어기의 설계

오원근*

Nonlinear Active Noise Control with On-Line Secondary Path Modeling

Won-Geun Oh*

이 논문은 2000년도 순천대학교 공과대학학술재단 교수해외파견연구비를 지원받았음

요 약

본 논문에서는 신경망을 이용한 새로운 비선형 능동소음제어 시스템(ANC; active noise control system)을 제안하였다. 제안한 시스템에서는 2개의 다층신경망을 사용하여 제어기와 2차 경로의 모델을 구성하였으며, 특히 기존의 비선형 ANC와는 달리 2차 경로의 모델링이 온라인으로 수행되기 때문에 2차경로가 시변 시스템인 경우에도 적용이 가능하다. 제안한 방법을 비선형 시변 시스템에 대해서 모의 실험한 결과 좋은 소음감소 효과를 보여주었다.

ABSTRACT

In this paper, we present a new nonlinear active noise control scheme using neural networks. Two neural networks are used, one is for the active controller and another one is for the secondary path model. This scheme is suitable for the plant which has time-varying secondary path dynamics, because the secondary path modeling is performed via on-line fashion. Simulation results of active noise control with nonlinear primary/secondary path are presented. The results show that the new algorithm can reduce the noise level greatly.

키워드

능동소음제어, 비선형시스템, 신경회로망, 신호처리

1. 서론

ANC(Active Noise Control) 기술은 원래의 소음에 인공적으로 생성한 반대위상의 소음을 더하여 전체적인 소음 레벨을 감소시키는 기술이다. ANC의 원리는 1930년대부터 알려졌으나 본격적으로 구현되고 응용되기 시작한 것은 DSP(Digital Signal Processing) 기술이 발전하기 이후라고 할 수 있으며, 현재는 공조시스템, 배기구 및 자동차 소음제어, 헤드셋 그리고 비행기의 소음제어 등 다양한 분야의 소음감소에 응용되고 있다[1].

그림 1에 이와 같은 ANC 시스템의 개념도를 나타내었다. 이 시스템에서는 먼저 감쇄시켜야 할 원소음을 마이크로폰으로 입력 받아, 이를 이용하여 크기는 같고 위상이 반대인 2차 소음을 제어기에서 인공적으로 생성하여 원소음과 합성함으로써 전체적인 소음 레벨을 감소시키는 구조로 되어있다.

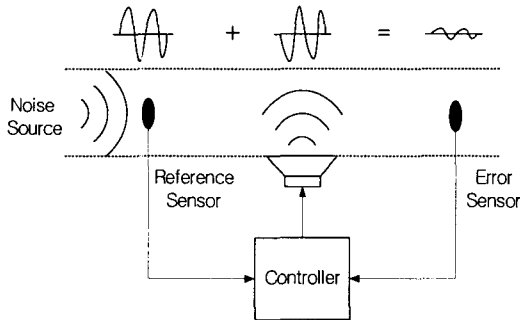


그림 1. 능동소음제어시스템
Fig. 1 Active Noise Control System

이러한 ANC 알고리즘중 현재 가장 널리 사용되고 있는 것은 그림 2에 나타낸 FXLMS (filtered-x LMS) 알고리즘이다. 여기에서 P는 원소음이 통과하는 주경로(primary path)이며, S는 제어기출력과 스피커 출력사이의 동역학을 나타내는 이차경로(secondary path), 그리고 C는 능동소음제어기로서 LMS(Least Mean Square) 적응알고리즘을 사용하는 선형 적응필터이다[1].

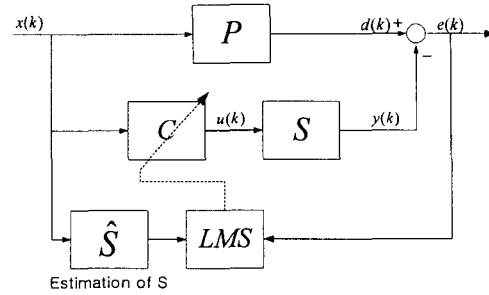


그림 2. FXLMS 능동소음제어 시스템
Fig. 2 FXLMS Active Noise Control System

이때 이차경로의 동역학 S는 제어기 자체를 제외한 D/A 변환기, A/D 변환기, 각종필터, 스피커, 앰프등의 동역학을 의미하며 제어기의 출력은 이러한 이차경로의 존재로 인해 왜곡되어 스피커로 출력되게 되므로, 제어기 C의 설계시 이를 고려하여야 한다. \hat{S} 는 이차경로 S의 모델로써 이와 같은 이차경로에 의한 스피커 출력의 왜곡을 상쇄하기 위한 것이다. \hat{S} 를 구하는 방법으로는 S가 시불변임을 가정하고 offline으로 사전에 모델링하는 방법이 주로 사용되고 있지만 응용분야에 따라서는 이러한 이차경로가 시변(time-varying)시스템인 경우가 있으며, 이때에는 실시간 모델링이 바람직하다. 선형 ANC에서 이차경로의 실시간 모델링에 대한 연구는 Sommerfeldt[2], Kuo 등[3][4]이 있다.

한편 ANC 기술이 적용되는 실제 환경에서는 주경로와 이차경로가 비선형 시스템 이거나, 이차경로가 비최소위상(nonminimum phase) 플랜트인 경우가 있다. 이때는 선형제어기로는 충분한 소음감소 효과를 얻기 힘들며, 비선형제어기의 사용이 바람직하다. Snyder와 Tanaka[5]는 MLP(multilayer perceptron) 신경망을 이용한 비선형 ANC를 제안하였으며, Bouchard등[6], Bouchard[7]은 heuristic 한 방법으로 [5]의 알고리즘의 속도와 계산량을 감소시키는 알고리즘을 제안하였다. 또한 Strauch와 Mulgrew [8]는 RBF(Radial Base Function)을 이용한 신경망 제어기를 비선형성 소음과 비최소위상 특성을 갖는 액추에이터가 있는 시스템에 적용하였다. 이러한 비선형 ANC 시스템들은 비선형 환경에

서 기존의 선형 ANC에 비해서 더 나은 소음 감소 성능을 보여주고 있어, 응용분야에 따라 선택적으로 사용한다면 좋은 결과를 기대할 수 있다.

비선형 ANC에서도 이차경로의 모델링은 성능에 영향을 주는 중요한 문제이다. 기존의 비선형 ANC 연구에서는 이차경로를 모두 offline으로 모델링하고 있으며, 온라인 모델링에 대한 연구는 전무한 상황이다. 본 논문에서는 주경로, 이차경로가 모두 미지의 비선형 시스템인 경우에 대해서, 이차경로의 실시간 모델링이 포함된 비선형 ANC를 개발하였다. 제어기와 이차경로의 모델은 모두 MLP 신경회로망을 사용하였으며, 이차경로를 실시간으로 모델링하면서 동시에 제어기가 동작하는 구조로 되어있다. 제안한 방법을 비선형 환경에서 모의실험한 결과 좋은 소음감소 효과를 보여주었다.

II. 비선형 ANC 알고리즘

2.1 이차경로의 실시간 모델링과 비선형 ANC

선형 ANC에서 이차경로의 온라인 모델링 방법으로 가장 효율적인 것은 그림 3에 보인 parallel modeling 또는 overall modeling 이라고 불리는 방법이다[2][3][4]. 이 알고리즘은 FXLMS에 이차경로의 모델을 얻기 위한 부분을 추가한 것으로써 모델 부분은 제어기와 동시에 온라인으로 동작하며, 플랜트에서 나오는 오차 $e(k)$ 와 모델에서 나오는 $\hat{e}(k)$ 의 차이를 최소화하도록 모델의 패러미터를 조정해 나간다. 이와 같은 과정을 통해 모델은 P와 S를 동시에 모델링하게 되며, 이렇게 동시에 P, S를 모델링하는 이유는 이차경로 S만을 단독으로 모델링할 경우에 생기는 바이어스를 없애기 위한 것이다[1].

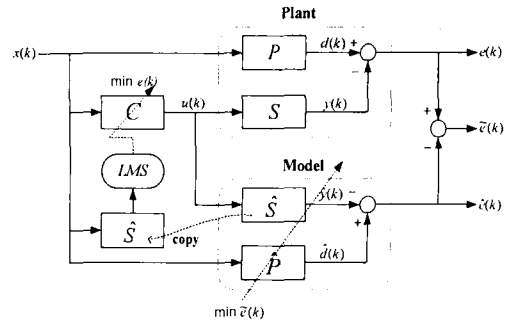


그림 3. 이차경로의 병렬모델링
Fig. 3 parallel modeling of secondary path

본 논문에서는 이와 같은 선형 ANC의 parallel modeling과 유사한 구조로, 이차경로를 온라인으로 모델링할 수 있는 비선형 ANC 시스템을 개발하였다. 제안한 시스템을 그림 4에 나타내었다. 여기서 P, S는 미지의 비선형 시스템이며, 각각 소음의 주경로와 이차경로를 의미한다. 또한 NNc 는 비선형 능동소음제어기이며 NNm 은 이차경로의 실시간으로 모델이다. NNc 와 NNm 에는 모두 다층신경망을 사용하였으며, 입력은 tapped delay line으로 구성하였다.

2.2 모델신경망 NNm

NNm 은 모델링을 위한 신경망으로 선형 ANC에서 overall modeling 처럼 P와 S를 동시에 모델링하는 역할을 한다. 사용한 신경망은 입력을 $u(k), x(k)$ 의 tapped delay line으로 구성된 다층신경망이다. NNm 의 입출력 관계식은 다음과 같다.

$$\begin{aligned} \hat{e}(k) &= NNm[u(k), u(k-1), \dots, u(k-N_u+1), \\ &\quad x(k), x(k-1), \dots, x(k-N_x+1)] \\ &= f\left(\sum_{i=1}^{N_1} v_{ii}^{(2)} \cdot f_i\left(\sum_{j=1}^{N_n} v_{ij}^{(1)} \cdot q_j + c_i^{(1)}\right) + c_1^{(2)}\right) \quad (1) \end{aligned}$$

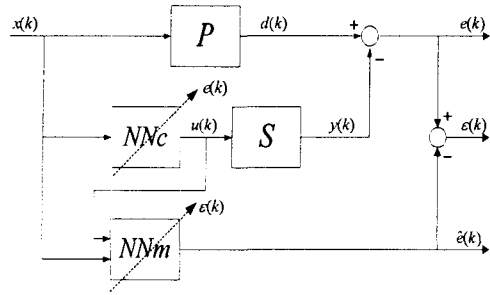


그림 4. 비선형 ANC 시스템
Fig. 4 Nonlinear ANC system

여기에서 $v_j^{(n)}$ 은 n-1번째 층(layer)의 j번째 뉴런과 n번째 층의 i번째 뉴런 사이의 가중치이며, $c_i^{(n)}$ 은 n번째 층 i번째 뉴런의 바이어스(bias)를 의미한다. 뉴런의 활성화함수 $f(\cdot)$ 는 시그모이드 함수를 사용하였으며, q_j 는 다음과 같은 입력벡터 Q의 j번째 원소이다.

$$Q = [u(k), \dots, u(k - N_u + 1), x(k), x(k - N_x + 1)] \quad (2)$$

NNm의 무게값은 다음과 같이 일반적인 역전파 학습방법(backpropagation)을 사용하여, 시스템의 실제 오차 $e(k)$ 와 NNm 출력 $\hat{e}(k)$ 의 차이인 $\epsilon(k)$ 를 최소화하도록 학습된다.

$$E_m = \frac{1}{2} \epsilon^2(k) = \frac{1}{2} [e(k) - \hat{e}(k)]^2 \quad (3)$$

$$\begin{aligned} \Delta v_{ij}(k) &= -\eta_m \frac{\partial E_m(k)}{\partial v_{ij}(k)} \\ &= -\eta_m \frac{\partial E_m(k)}{\partial \hat{e}(k)} \frac{\partial \hat{e}(k)}{\partial v_{ij}(k)} \\ &= \eta_m \epsilon(k) \frac{\partial \hat{e}(k)}{\partial v_{ij}(k)} \end{aligned} \quad (4)$$

여기에서 η_m 은 학습률(learning rate)이다. 위의 학습 과정에서 $e(k)$ 는 NNm의 무게값의 변화와 무관하다고 가정하며, 이 가정은 이차경로의 동역학이

천천히 변하거나 학습률 η_m 을 낮게 설정함으로써 근사적으로 대부분 만족될 수 있다. 이와 같은 학습 과정은 선형 ANC의 parallel modeling을 비선형으로 확장한 것으로, 학습이 만족스럽게 이루어져 NNm의 출력이 $\hat{e}(k) \approx e(k)$ 가 되면 NNm에 P, S의 동역학이 학습되었다고 볼 수 있다.

2.3 제어 신경망 NNc

신경망 NNc는 비선형 능동소음제어기이며, 원소음 $x(k)$ 를 N_x 개의 tapped delay line으로 구성된 입력력을 갖는 다층퍼셉트론이다. NNc의 입출력 관계식은 다음과 같다.

$$\begin{aligned} u(k) &= NNc[x(k), x(k-1), \dots, x(k-N_x+1)] \\ &= \sum_{i=1}^{M_1} w_{i1}^{(2)} \cdot f_i \left(\sum_{j=1}^{M_n} w_{ij}^{(1)} \cdot r_j + b_i^{(1)} \right) + b_i^{(2)} \end{aligned} \quad (5)$$

여기에서 $w_{ij}^{(m)}$ 은 m-1번째 층(layer)의 j번째 뉴런과 m번째 층의 i번째 뉴런 사이의 가중치이며, $b_i^{(n)}$ 은 n번째 층 i번째 뉴런의 바이어스(bias)를 의미한다. 또한 $f(\cdot)$ 는 시그모이드 함수를 사용하였으며, r_j 는 다음과 같은 입력벡터 R의 j번째 원소이다.

$$R = [x(k) \ x(k-1) \ x(k-N_x+1)] \quad (6)$$

이 같은 신경망 NNc의 목적은 소음레벨 $e(k)$ 가 최소가 되도록 하는 신호를 출력하는 것이 목적이므로 다음과 같이 오차함수를 설정한다.

$$E_c(k) = \frac{1}{2} e^2(k) = \frac{1}{2} [d(k) - y(k)]^2 \quad (7)$$

(7)을 최소화하는 무게값을 구하기 위한 학습방법은 최급강하법(gradient descent)을 사용하여 다음과 같이 유도된다. 먼저 2차경로 S가 다음과 같이 시간지연이 있는 비선형함수의 형태를 갖는다고 가

정하자.

$$y(k) = S[u(k - d_1), u(k - d_1 - 1), \dots, u(k - d_1 - N_s)] \quad (8)$$

여기에서 d_1 은 미지의 시간지연항으로 고정된 값 일 필요는 없다. 이와 같이 가정하면 오차 $e(k)$ 에 영향을 주는 NNc 의 무게값은 $w_{ij}^{(m)}(k - d_1 - N_s), \dots, w_{ij}^{(m)}(k - d_1)$ 이므로 다음과 같이 무게값의 증분 $\Delta w_{ij}^{(m)}(k)$ 를 설정한다.

$$\Delta w_{ij}^{(m)}(k) = -\eta_c \sum_{n=0}^{N_u} \frac{\partial E_c(k)}{\partial w_{ij}^{(m)}(k - n)} \quad (9)$$

여기에서 η_c 는 학습률이며, N_u 는 $N_u \geq d_1 + N_s$ 가 되도록 충분히 큰 값으로 선택한다. (7)을 (9)에 대입하고 전개하면 다음과 같다.

$$\begin{aligned} \Delta w_{ij}^{(m)}(k) &= -\eta_c \cdot e(k) \cdot \sum_{n=0}^{N_u} \frac{\partial e(k)}{\partial u(k - n)} \frac{\partial u(k - n)}{\partial w_{ij}^{(m)}(k - n)} \\ &= \eta_c \cdot e(k) \cdot \sum_{n=0}^{N_u} \frac{\partial y(k)}{\partial u(k - n)} \frac{\partial u(k - n)}{\partial w_{ij}^{(m)}(k - n)} \quad (10) \end{aligned}$$

(10)에서 \sum 내의 첫번째 항 $\partial y(k) / \partial u(k - n)$ 는 S 의 Jacobian으로써 S 의 정확한 식을 모르기 때문에 직접 계산은 불가능하며, 다음과 같이 모델신경망 NNm 을 이용하여 근사적으로 Jacobian을 구하는 방법을 사용한다. NNm 이 수렴한 후에는

$$\hat{e}(k) \cong e(k) = d(k) - y(k) \quad (11)$$

이므로 양변을 $u(k - n)$ 로 편미분하면 다음과 같다.

$$\frac{\partial \hat{e}(k)}{\partial u(k - n)} \cong \frac{\partial e(k)}{\partial u(k - n)} = -\frac{\partial y(k)}{\partial u(k - n)} \quad (12)$$

(12)를 이용하면 Jacobian의 근사값은 다음과 같이 구해진다.

$$\begin{aligned} \frac{\partial y(k)}{\partial u(k - n)} &= -\frac{\partial \hat{e}(k)}{\partial u(k - n)} \\ &= \sum_{i=1}^{N_1} f'(net_i^{(2)}) \cdot v_{ii}^{(2)} \cdot f'(net_i^{(1)}) \cdot v_{ii}^{(1)} \quad (13) \end{aligned}$$

이와 같이 모델을 이용하여 근사적으로 Jacobian을 얻어내는 방법은 neurocontrol 분야에서 자주 이용되고 있으며[9][10], 이때 모델을 통해 얻어진 Jacobian의 부호(sign)만 정확하게 모델링 된다면 전체 제어기의 성능에는 큰 영향이 없다는 것이 알려져있다[9].

다음으로 (10)에서 \sum 내의 두번째 항인 $\partial u(k - n) / \partial w_{ij}^{(m)}(k - n)$ 은 (5)를 이용하면 다음과 같이 계산된다.

$$\frac{\partial u(k - n)}{\partial w_{ij}^{(m)}(k - n)} = \delta_i^{(m)}(k - n) \cdot out_j^{(m-1)}(k - n) \quad (14)$$

여기에서

$$\begin{aligned} \delta_i^{(m)}(k - n) &= \begin{cases} f'_i(net_i^{(m)}(k - n)), & \text{output neuron} \\ f'_i(net_i^{(m)}) \cdot \sum_{l=1}^{N_{l+1}} \delta_l^{(m+1)}(k - n) \cdot w_{li}^{(m+1)}(k - n), & \text{otherwise} \end{cases} \quad (15) \end{aligned}$$

이다. 여기에서 시간 k 에서 (14)를 계산하기 위해서는 $k - N_s$ 에서 k 까지의 무게값과 각 뉴런의 출력값을 모두 고려하여야 하므로, 많은 저장공간과 계산량을 필요로한다. 그러나 학습률이 낮아서 무게값 $w_{ij}^{(m)}$ 가 시간 N_s 동안 크게 변하지 않고(즉, $w_{ij}^{(m)}(k - n) \approx w_{ij}^{(m)}(k)$), 원소음 $x(k)$ 가 시간 N_s 동안 급격하게 변하지 않는다면 다음과 같은 근사식이 성립하므로 이를 이용하면 저장공간과 계산량을 크게

출일 수 있다.

$$\frac{\partial u(k-n)}{\partial w_{ij}^{(m)}(k-n)} \approx \frac{\partial u(k)}{\partial w_{ij}^{(m)}(k)} \quad (16)$$

2.4 Jacobian 근사값의 필터링

(13)을 이용해서 구한 근사적인 Jacobian은 실제로 구해보면 대부분 미세한 변화가 심한 것을 알 수 있다. 따라서 Jacobian이 0근방의 값을 가질 때 부호의 변화가 심해서 gradient의 방향이 계속 변화함에 따라 학습이 이루어지지 못할 수 있기 때문에, 다음과 같은 필터링 과정을 통해 부호의 변화가 연속적으로 일어나지 않도록 방지한다. S의 Jacobian을 $J_n(k)$ 라 했을 때 필터링 알고리즘은 다음과 같다.

$$\tilde{J}_n(k) = \varphi\left(\sum_{j=k-7+1}^k J_n(j)\right) \quad (17)$$

여기에서 φ 는 다음과 같은 데드존(deadzone) 함수이다.

$$\varphi(x) = \begin{cases} x+a, & x < -a \\ 0, & -a \leq x \leq a \\ x-a, & x > a \end{cases} \quad (18)$$

이상의 알고리즘을 요약하면 시간 k에서의 계산 과정은 다음과 같다.

- (step 1) $x(k)$ 를 입력받는다.
- (step 2) NNc 의 출력을 계산한다.
- (step 3) NNm 의 출력을 계산한다.
- (step 4) Jacobian의 근사값 $J_n(k)$ 을 계산한다.
- (step 5) $J_n(k)$ 를 필터링한다.
- (step 5) NNc 와 NNm 의 무게값을 갱신한다.

III. 모의실험

제안한 시스템을 다음과 같은 시스템에 대해서 Matlab을 사용하여 모의실험을 하였다. 먼저 주경로 P는 미지의 비선형 시스템으로 입출력 식은 다음과 같다.

$$\begin{aligned} d(k) &= g(k) + 0.08g^2(k) - 0.04g^3(k) \\ g(k) &= x(k-3) - 0.3x(k-4) + 0.2x(k-5) \end{aligned} \quad (19)$$

이차경로 S는 다음과 같은 미지의 비선형 시변 시스템이다.

$$S = \begin{cases} S_1 & 0 \leq k < 1000 \\ S_2 & 1000 \leq k < 2000 \\ S_3 & 2000 \leq k < 3000 \end{cases} \quad (20)$$

여기에서,

$$\begin{aligned} S_1 : \\ y(k) &= y_1(k) + 0.03y_1^2(k) - 0.05y_1^3(k) + 0.025y_1^4(k) \\ y_1(k) &= u(k-2) + 1.5u(k-3) - u(k-4) \end{aligned}$$

$$\begin{aligned} S_2 : \\ y(k) &= 0.9y_1(k) + 0.02y_1^2(k) + 0.03y_1^3(k) + 0.025y_1^5(k) \\ y_1(k) &= u(k-2) + 0.5u(k-3) \end{aligned}$$

$$\begin{aligned} S_3 : \\ y(k) &= 1.1y_1(k) + 0.05y_1^2(k) + 0.02y_1^3(k) + 0.04y_1^5(k) \\ y_1(k) &= 0.9u(k-2) + 0.4u(k-3) \end{aligned} \quad (21)$$

이다. 그리고 시스템의 입력인 원소음은 다음과 같다.

$$x(k) = 0.5 \sin\left(\frac{\pi k}{12}\right) + 0.4 \cos\left(\frac{\pi k}{6}\right) + 0.3 \sin\left(\frac{\pi k}{4}\right) + n_0(k) \quad (22)$$

여기에서 $n_0(k)$ 는 평균이 0 이고 분산이 0.04인 가우시안잡음이다. 이와 같은 시스템에 대해서 모델 신경망 NN_m 으로는 입력층, 은닉층, 출력층 뉴런의 개수가 각각 14, 11, 1개인 다층신경망을 사용하였고, 학습은 모멘텀항이 있는 역전파학습법이 적용되었다. 학습시의 학습률은 $\eta_m = 0.01$, $\alpha_m = 0.3$ 이다. 제어신경망 NN_c 는 입력층, 은닉층, 출력층 뉴런의 수가 각각 7, 7, 1개인 다층신경망을 사용하여 학습률 $\eta_c = 0.02$ 로 학습을 수행하였다. 그리고 Jacobian 필터링 시의 패러미터는 $T = 7$, $\alpha = 0.1$ 이다.

모의실험의 결과를 그림 5~7에 나타내었다. 그림 5는 원소음 $x(k)$ 와 제어된 소음 $e(k)$ 를 보인것으로 전체적인 소음 레벨이 줄어든 것을 볼 수 있다. 그림 6은 모델링 오차 $e(k)$ 와 $\hat{e}(k)$ 의 차이인 $\varepsilon(k)$ 의 추이를 보인것으로써 비선형 시변환경에서도 온라인으로 모델링이 잘 이루어짐을 알 수 있다. 그림 7은 $x(k)$ 와 $e(k)$ 의 파워스펙트럼으로 입력 $x(k)$ 의 3개의 주요 주파수 성분이 10~18 dB 정도 감소됨을 알 수 있다. 고주파 성분의 파워는 약 3dB가량 원소음보다 더 증가함을 볼 수 있으나 전체적으로는 10dB이상의 소음의 파워가 감소되어 제안한 알고리즘이 잘 동작함을 보여주고 있다.

IV. 결론

본 논문에서는 2개의 다층신경망을 이용한 비선형 ANC 시스템을 개발하였다. 이 시스템은 기존의 선형 feedforward ANC 시스템에서의 이차경로의 온라인 모델링 알고리즘을 비선형으로 확장하여, 비선형 환경에서의 능동소음제어와 이차경로의 온라인 모델링을 동시에 수행한다. 모의실험 결과 많지 않은 뉴런을 사용한 신경망으로도 좋은 성능을 보임으로써 전용 뉴런칩을 사용하지 않고 일반적인 DSP 칩으로도 제안한 방법을 사용할 수 있음을 보여주었다. 하지만 신경망 제어시스템이 같은 공통적인 문제점인 뉴런의 개수, 그리고 휴리스틱하게 설정해야 하는 패러미터들이 있다는 점과 안정성에

대한 이론적인 검증이 힘들다는 점은 향후에 계속적인 연구과제로 남는다. 앞으로 DSP칩을 사용하여 제안한 방법을 구현하는 연구를 수행할 예정이다.

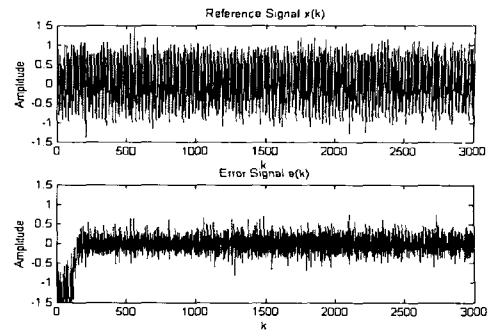


그림 5. 원소음 $x(k)$ 와 제어된 소음 $e(k)$
Fig 5. Original noise $x(k)$ and controlled noise $e(k)$

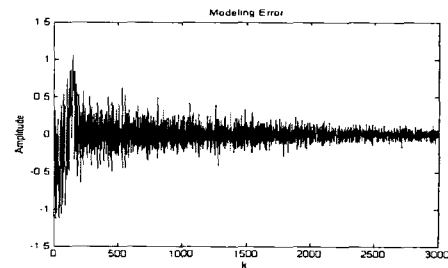


그림 6. 모델링 오차 $\varepsilon(k)$
Fig 6. Modeling Error $\varepsilon(k)$

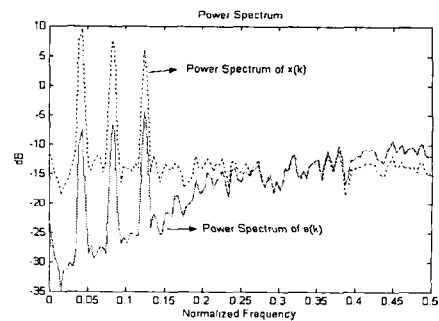


그림 7. $x(k)$ 와 $e(k)$ 의 파워스펙트럼
Fig 7. Power Spectrum of $x(k)$ and $e(k)$

참 고 문 헌

- [1] Sen M. Kuo and Dennis R. Morgan, "Active Noise Control: A Tutorial Review," *Proceedings of the IEEE*, Vol. 87, No. 6, pp. 943~973, June 1999
- [2] Scott D. Sommerfeldt, "Multi-channel Adaptive Control of Structural Vibration," *Noise Control Engineering Journal*, Vol. 37, No. 2, pp. 77~89, Sep.~Oct. 1991
- [3] Javier Tapia and Sen M. Kuo, "New Adaptive On-line Modeling Technique for Active Noise Control Systems," *IEEE International Conference on Systems Engineering*, Pittsburgh, Pennsylvania, August 1990, pp. 280~283
- [4] Sen M. Kuo, Mingjie Wang, and Ke Chen, "Active Noise Control System with Parallel On-Line Error Path Modeling Algorithm," *Noise Control Engineering Journal*, Vol. 39, pp. 119~127, Nov.~Dec. 1992
- [5] Scott D. Snyder and Nobuo Tanaka, "Active Control of Vibration Using a Neural Network," *IEEE Trans. on Neural Networks*, Vol. 6, No. 4, pp. 819~828, July 1995
- [6] Martin Bouchard, Bruno Paillard, and Chon Tan Le Dinh, "Improved Training of Neural Networks for the Nonlinear Active Control of Sound and Vibration," *IEEE Trans. on Neural Networks*, Vol. 10, No. 2, pp. 391~401, March 1999
- [7] Martin Bouchard, "New Recursive-Least-Square Algorithms for Nonlinear Active Control of Sound and Vibration Using Neural Networks," *IEEE Transactions on Neural Networks*, Vol. 12, No. 1, January 2001
- [8] Paul Strauch and Bernard Mulgrew, "Active Control of Nonlinear Noise Processes in a Linear Duct," *IEEE Trans. on Signal Processing*, Vol. 46, No. 9, pp. 2404~2412, September 1998
- [9] M. Saerens and A. Soquet, "Neural controller based on back-propagation algorithm," *IEE Proceedings F*, Vol. 138, No. 1, pp. 55~62, February 1991
- [10] Jose R. Noriega and Hong Wang, "A Direct Adaptive Neural-Network Control for Unknown Nonlinear Systems and Its Application," *IEEE Transactions on Neural Networks*, Vol. 3, No. 1, January 1998

저 자 소 개



오원근(Won-Geun Oh)

1989년 2월 한양대학교 전자
통신 공학 학사

1991년 2월 한양대학교 대학
원 전자통신공학 석사

1997년 2월 한양대학교 대학
원 전자통신공학 박사

1997년 3월 ~ 현재 (국립)순천대학교 정보통신공
학부 조교수

※관심분야 : 신호처리, 능동소음제어, 신경망,
비선형시스템, 마이크로프로세서 응용