

뉴로-퍼지 제어를 이용한 능동 소음제어

*박희경, 공성곤
 송실대학교 전기공학과

Adaptive Noise Canceling by Neuro-Fuzzy Controller

Hee-Kyoung Park and Seong-Gon Kong
 Department of Electrical Eng., Soongsil University

Abstract - 본 논문에서는 뉴로-퍼지 제어를 이용한 능동 소음제어기를 구현하였다. 능동 소음제어기는 잡음에 의하여 왜곡된 신호로부터 잡음을 제거하여 원 신호를 복원하는 제어시스템이다. 일반적으로 잡음의 특성이 시간에 따라 변화하고, 전달특성이 비선형적이므로 고정된 제어기에 의해서는 제어할 수 없다. 이 논문에서는 뉴로-퍼지 제어를 사용하였고, 파라미터를 오차 역전파 학습을 통하여 변화시킴으로써 잡음의 특성에 효과적으로 적응하는 능동 소음제어기를 구성하였다. 시뮬레이션을 통하여 여러 종류의 신호에 대해서 랜덤 노이즈를 발생시키고 구성된 제어기의 성능을 확인하였다.

기의 구성을 나타낸다.

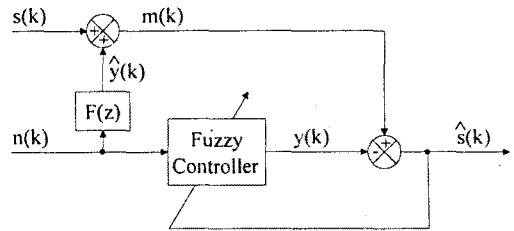


그림 1: 전체 구성도

1. 서 론

소음공해에 대한 관심이 높아지면서 특히 자동차나 항공기 내에서의 소음이 심각하게 대두되고 있다. 능동 소음 제어방법은 제어를 이용하여 원래의 소음과 진폭은 같고 위상이 반대인 음파를 발생시켜 소음을 감쇄시키는 방법이다[1][2]. 본 논문에서는 밀폐된 공간에서의 소음을 제어하기 위한 방법으로 뉴로-퍼지 제어를 제안하였다. 퍼지 제어기는 비선형성 특성을 가지고 있으며 제어기의 구조와 파라미터를 학습을 통하여 변화시킴으로써 강인한 성능의 제어를 구성할 수 있다. 가우시안 함수를 전건부의 멤버십 함수로 사용하는 초기화된 제어기의 파라미터의 결정은 초기에 들어오는 입출력 데이터를 사용하여 Subtractive 클러스터링 방법으로 구성한다. Subtractive 클러스터링 방법으로 구한 중심값 중 입력에 해당하는 중심값을 입력 멤버십 함수의 중심값으로 사용한다. 이렇게 초기화된 퍼지 제어기의 성능향상을 위하여 신경망 기법인 역전파 학습을 사용한다. 즉, 입출력 데이터를 이용하여 제어기의 파라미터를 조정함으로써 강인한 퍼지 제어를 구성하여 소음신호를 추종한다. 추종된 퍼지 제어기의 출력 신호를 크기는 같고 위상이 반대인 신호로 변환하여 더해줌으로써 소음을 제거하고 원신호를 복원하게 된다.

2. 퍼지 제어기 구성

2.1 시스템 구성

밀폐공간에서의 소음제어는 몇 가지 가정을 전제로 한다. 먼저 전달하고자 하는 정보를 포함하는 원신호와 왜곡된 소음과의 합인 측정신호를 정확하게 알고 있으며 소음에 대한 정보 또한 알고 있다는 가정이다. 또한 원신호와 소음간의 상관관계가 충분히 작아야 하며 왜곡되어지는 경로의 차수에 대한 사전지식이 어느 정도 확보되어야 한다.

그림 1은 이 논문에서 사용하고 있는 뉴로-퍼지 제어

여기서 $s(k)$ 는 전달하고자 하는 정보를 나타내는 원신호이고 $n(k)$ 는 외부에서 유입되는 소음이다. 이 소음은 $F(z)$ 라는 전달함수를 가지는 전달경로를 통하여 변형된 잡음신호 $\hat{y}(k)$ 가 되어 원신호와 합쳐진다[3]. 퍼지 제어를 통해서 왜곡되는 소음을 출력으로 추종하게 되고 추종된 결과 $y(k)$ 와 측정신호 $m(k)$ 와의 차를 복원신호 $\hat{s}(k)$ 라고 한다. 이때 소음과 측정신호의 데이터를 이용하여 퍼지제어기의 파라미터를 조정하게 된다. 측정신호 $m(k)$ 는 다음 식과 같이 표현된다.

$$m(k) = s(k) + F[n(k)] = s(k) + \hat{y}(k) \quad (1)$$

퍼지 제어기의 출력 $y(k)$ 가 왜곡된 소음 $\hat{y}(k)$ 를 정확하게 추종하게 된다면 측정신호 $m(k)$ 에서 퍼지 출력 신호 $y(k)$ 를 뺀 복원신호 $\hat{s}(k)$ 는 원신호 $s(k)$ 의 추정치가 된다. 원신호와 소음의 상관관계가 작기 때문에 퍼지 제어기의 출력은 측정신호 중 왜곡된 소음신호만을 추종하게 된다. 왜곡된 소음신호를 추종하게 되면 측정신호에서 빼줌으로써 원신호를 복원하게 된다. 퍼지 제어기는 3개의 규칙과 3개의 입력을 가지고 있으며 각각의 입력에 3개씩의 가우시안 함수를 전건부 멤버십함수로 갖는 T.S.K. 모델로 초기화하였다.

2.2 퍼지 제어기의 파라미터 초기화

본 논문에서는 초기 퍼지 제어기로 T.S.K. 퍼지 모델을 사용하였으며 전건부 멤버십 함수는 가우시안 함수로 구성하였다[4]. T.S.K. 퍼지모델은 후건부의 출력이 전건부의 입력들의 선형결합으로 이루어진다. 퍼지 제어기의 입력은 소음이 되며 출력은 왜곡된 소음을 추종하는 신호가 되고 각각의 입력력 범위는 -2.5에서 2.5까지로 하였다.

잡음의 전달경로는 다음과 같은 전달함수 $F(z)$ 를 이용하여 임의로 정의하였다.

$$F(z) = \frac{8 - 8z^{-1}}{15 - 5z^{-1}} \quad (2)$$

퍼지 제어기의 입력은 현재 소음과 1차 지연 소음 그리고 1차 지연 출력으로 구성하고 후진부의 출력을 이것들의 선형결합의 형태로 만들면 퍼지 제어기는 잡음의 전달경로 $F(z)$ 와 유사한 특성을 가지게 된다. 만약 $F(z)$ 의 차수가 늘어난다면 퍼지 제어기의 입력 또한 같은 차수로 구성하여 $F(z)$ 를 근사화할 수 있다. 퍼지 제어기가 $F(z)$ 를 근사화하면 출력은 왜곡된 소음 $\hat{y}(k)$ 를 추종하게 된다.

앞의 방법으로 퍼지 제어기의 구조를 결정하면 다음은 퍼지 제어기의 파라미터를 초기화해야 한다. 본 논문에서는 제어기의 파라미터 중 전진부 멤버십의 중심값을 Subtractive 클러스터링 방법을 이용하여 구하였다[5]. 이 방법은 먼저 각각의 데이터를 입력력 공간상에 하나의 점으로 나타낸다. 그리고 각 점에서 다른 모든 점들 간의 거리를 계산하여 계산된 거리의 합을 이용하여 데이터가 가장 많이 있는 점을 첫 번째 클러스터링 센터로 정한다. 두 번째 센터를 정하기 전에 먼저 첫 번째 센터의 영향을 없애주어야만 근접한 영역에서 여러개의 센터가 발생하는 것을 피할 수 있다. 첫 번째 센터의 영향을 제거한 후 다시 같은 방법으로 두 번째 센터를 구하는 순서로 반복이 된다. 본 논문에서는 전진부 멤버십 함수를 3개로 하였으므로 3개의 클러스터링 센터를 구한다. 가우시안 함수의 분산값은 같은 크기의 임의의 값으로 하였다. 각각의 멤버십 함수에 대한 소속값은 식 (3)과 같다.

$$A_{ij}(x_j) = \exp\left[-\frac{(x_j - c_{ij})^2}{2\sigma^2}\right] \quad (3)$$

여기서 $A_{ij}(x_j)$ 와 c_{ij} 는 j 번째 입력변수에 대한 i 번째 멤버십 함수의 소속도와 중심값이고 σ^2 는 분산값이다.

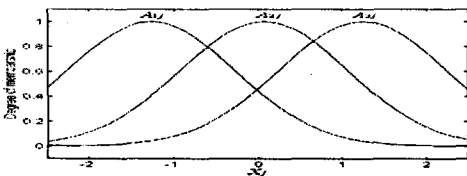


그림 2: 전진부 멤버십 함수

그림 2는 입력 퍼지변수 x_j 에 대한 멤버십 함수를 나타낸다. 멤버십 함수의 형태는 가우시안 함수로 하였고, 멤버십 함수의 중심값은 클러스터링 과정을 통하여 결정한다.

추론부는 아래규칙과 같이 나타내었으며 각각의 규칙에 있는 3개씩의 파라미터 a_0, a_1, b_1 는 0으로 초기화하였다. 다음은 본 논문에서 사용된 퍼지 규칙이다.

- R1 : If x_1 is A_{11} and x_2 is A_{12} and x_3 is A_{13}
then $y(k)$ is $a_{10}x_1 + a_{11}x_2 + b_{11}x_3$
- R2 : If x_1 is A_{21} and x_2 is A_{32} and x_3 is A_{23}
then $y(k)$ is $a_{20}x_1 + a_{21}x_2 + b_{21}x_3$
- R3 : If x_1 is A_{31} and x_2 is A_{22} and x_3 is A_{33}
then $y(k)$ is $a_{30}x_1 + a_{31}x_2 + b_{31}x_3$

위의 규칙에 사용된 3개의 입력변수는 다음과 같다.
 $x_1 = n(k), x_2 = n(k-1), x_3 = y(k-1)$
퍼지화 방법은 싱글톤 방법을 사용하였으며 퍼지추론 방식은 product implication, additive aggregation, 그리

고 비퍼지화 방법은 무게 중심법을 사용했다.

2.3 퍼지 제어기의 학습 과정

초기화된 퍼지 제어기를 오차를 줄이는 방향으로 파라미터를 학습함으로써 소음을 능동적으로 제어한다. 이를 위해 사용된 학습 데이터는 소음과 측정신호가 된다. 소음을 입력으로 측정신호를 출력으로 하며 오차인 복원신호를 줄이는 방향으로 역전파 학습을 사용한다. 즉 현재 퍼지 제어기의 출력과 현재 측정신호와와의 차이인 복원신호를 제공하여 이것을 최소화하도록 후진부 파라미터들을 학습한다. 복원신호의 제공은 일반적인 표현에 의해 다음 식과 같이 나타낸다.

$$E(k) = \frac{1}{2} (y(k) - m(k))^2 \quad (4)$$

$$= \frac{1}{2} \hat{s}^2(k)$$

또한 퍼지 제어기의 출력은 식 (5)와 같이 쓸 수 있다.

$$y(k) = \sum_{i=1}^3 \tau_i (a_0^k n(k) + a_1^k n(k-1) + b_1^k y(k-1)) \quad (5)$$

여기서 a_0^k, a_1^k, b_1^k 는 각각 k 번째 학습된 값이고 τ_i 는 i 번째 규칙의 정규화된 적합도이다. 따라서 각각의 파라미터는 다음 식 (6)-(8)과 같이 $E(k)$ 를 줄이는 방향으로 학습하게 된다.

$$a_0^{k+1} = a_0^k - \alpha \frac{\partial E(k)}{\partial a_0^k}$$

$$= a_0^k - \alpha \frac{\partial E(k)}{\partial \hat{s}(k)} \frac{\partial \hat{s}(k)}{\partial a_0^k} \quad (6)$$

$$= a_0^k - \alpha \hat{s}(k) \tau_i n(k)$$

$$a_1^{k+1} = a_1^k - \alpha \frac{\partial E(k)}{\partial a_1^k}$$

$$= a_1^k - \alpha \hat{s}(k) \tau_i n(k-1) \quad (7)$$

$$b_1^{k+1} = b_1^k - \alpha \frac{\partial E(k)}{\partial b_1^k}$$

$$= b_1^k - \alpha \hat{s}(k) \tau_i y(k-1) \quad (8)$$

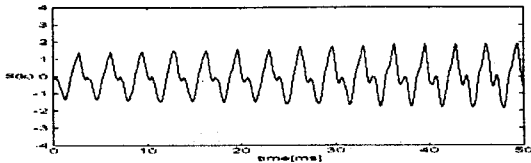
위 식에서 α 는 학습속도를 나타내는 상수로써 임의의 양수 값을 가진다.

3. 시뮬레이션

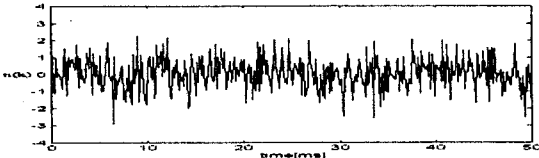
제안한 뉴로-퍼지 제어기의 성능을 확인하기 위하여 실제 음성신호 $s_1(k)$, 랜덤신호 $s_2(k)$, 그리고 신호가 없는 경우 $s_3(k)=0$ 등 3가지의 다른 종류의 원신호를 사용하여 시뮬레이션하였다. 학습속도는 $\alpha=0.01$, 분석구간은 50[msec]로 하였다. 실제 음성신호는 8[kHz]로 샘플링하였다. 퍼지 제어기의 초기화는 처음 데이터 30개만을 사용하였다. 클러스터링에 의해 초기 퍼지 제어기의 첫 번째 입력변수 $n(k)$ 의 멤버십 함수의 중심값은 다음과 같다. $c_{11} = -1.2704, c_{21} = 0.0751, c_{31} = 1.2460$ 이다. 두 번째 입력변수 $n(k-1)$ 의 중심값은 $c_{12} = -1.349, c_{22} = -0.7012, c_{32} = 0.6268$ 이다. 퍼지 제어기의 세 번째 입력변수는 1차 지연된 출력 $y(k-1)$ 이며 처음 초기화 과정에서의 퍼지 출력은 0이 되므로 3개의 멤버십 함수의 중심값은 임의로 다음과 같이 초기화하였다. $c_{13} = -1.5, c_{23} = 0, c_{33} = 1.5$ 이다. 분산값은 임의로 1로 하였다.

학습전에 0으로 초기화된 퍼지제어기의 파라미터들은 학습에 의하여 다음과 같이 최적화 되었으며, 그 값들은 다음과 같다.

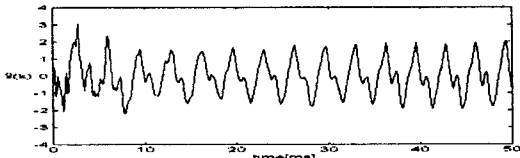
R1: $a_{10} = 0.288$ $a_{11} = 0.242$ $b_{11} = 0.133$
 R2: $a_{20} = 0.524$ $a_{21} = 0.599$ $b_{21} = 0.258$
 R3: $a_{30} = 0.510$ $a_{31} = 0.629$ $b_{31} = 0.308$



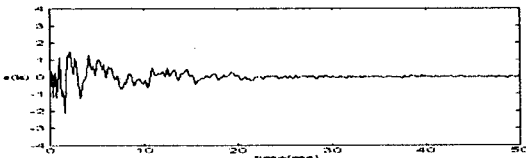
(a) 원신호 $s_1(k)$



(b) 소음 $n(k)$



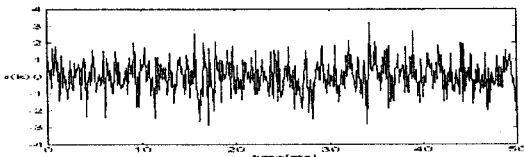
(c) 복원신호 $\hat{s}_1(k)$



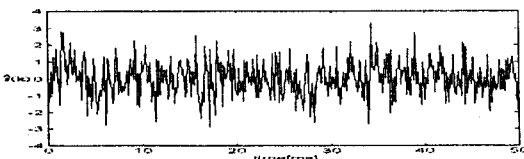
(d) 오차 $s_1(k) - \hat{s}_1(k)$

그림 3: 음성신호에 대한 소음제어 결과

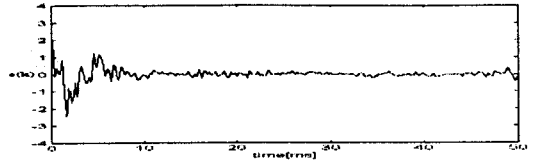
그림 3(a)는 음성신호 $s_1(k)$ 를 나타낸다. 음성신호는 유성음을 8[kHz]로 샘플링하여 얻었다. 그림 3(b)는 평균이 0이고 분산값이 1인 랜덤소음을 나타낸다. 그림 3(c)는 뉴로-퍼지 제어기의 출력과 추정신호와의 차이인 복원신호이다. 충분히 학습된 퍼지 출력은 왜곡된 소음을 유추하게 되고 추정신호에서 뉴로-퍼지 능동 소음제어기의 출력을 빼면 원신호를 복원하게 된다. 그림 3(d)는 제어를 통해 소음을 제거시킨 결과를 나타낸다. 최종적으로 20[ms]정도의 시간이 경과했을 때 퍼지 제어기가 충분히 학습되었으며 왜곡된 소음을 제거하여 소음이 현저히 줄어드는 것을 알 수 있다.



(a) 원신호 $s_2(k)$



(b) 복원신호소음 $\hat{s}_2(k)$



(c) 오차 $s_2(k) - \hat{s}_2(k)$

그림 4: 랜덤 입력

그림 4는 원신호로 랜덤 신호를 가했을 때의 결과 그림이다. 그림 4(a)는 원신호이고 그림 4(b)는 복원된 원신호를 나타내며 그림 4(c)는 복원신호와 원신호와의 차를 나타낸다.

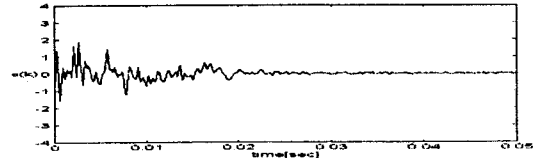


그림 5: 원신호 제거실험시 오차 $s_3(k) - \hat{s}_3(k)$

그림 5은 원신호를 가지지 않았을 때의 소음제어 결과 그림이다.

4. 결론

본 논문에서는 외부로부터 유기되는 소음을 제어하기 위한 방법으로 뉴로-퍼지 제어를 구성하였다. 초기화는 Subtractive 클러스터링 방법을 사용하였고 학습은 역전파 학습방법을 사용하였다. 원신호를 실제 음성신호와 랜덤 신호로 사용하였으며 원신호를 제거하였을 때의 결과도 확인하였다. 시뮬레이션의 결과를 통하여 소음을 제거하여 원신호를 복원하는 것을 확인할 수 있었다. 향후 계획은 학습 성능을 더 개선시키는 방법으로 입력 멤버십 함수의 파라미터도 학습에 의해 변환시키는 방법을 제시할 것이다. 또한 왜곡된 경로에 대하여 좀더 일반적이고 복잡한 경로를 고려해야 할 것이다.

(참고 문헌)

- [1] F. Jiang and H. Tsuji, "Adaptation for Active Noise Control," *IEEE Trans. on Control Systems*, Vol. 17 No. 6, 36-47, Dec. 1997.
- [2] J. Gu and S. F. Yau, "A Model-based Approach to Active Noise Cancellation using Loud Speaker Array," *Proc. of ICASSP-97*, Vol. 1, 379-382, 1997.
- [3] D. Guicking, "Active Control of Sound and Vibration," *ASJ Symposium on Active Control of Sound and Vibration*, 109-118, 1991.
- [4] 이상령, 오경환, "신경망을 이용한 퍼지규칙 생성과 추론망 구축," 퍼지 및 지능 시스템학회 논문지, Vol. 7, No. 3, 43-53, 1997. 8.
- [5] R. Yager and P. Filev, *Essentials of Fuzzy Modeling and Control*, John-Wiley and Sons, 1994.
- [6] Y. Shi and M. Otani, "A Learning Algorithm for Tuning Fuzzy Rule Based on the Gradient Descent Method," *Int. Conf. on Fuzzy Systems*, Vol. 1, 55-61, 1996.