

## 신경망필터를 이용한 음질향상

### Speech Enhancement using the Neural Network Filter

김종우 · 공성곤

Jong-Woo Kim and Seong-Gon Kong

숭실대학교 전기공학과

#### 요 약

본 논문에서는 잡음환경에서의 음질향상(Speech Enhancement) 시스템 구현을 목적으로 한다. 이를 위한 적응필터로서 LMS(Least Mean Square)알고리즘 FIR필터를 적용한다. 또 정밀 필터로서 다층신경망(MLP, Multi-Layer Perceptron) 필터를 적용한다. 잡음환경에서의 음성신호 복원 및 음질향상 시스템은 잡음에 의해 왜곡된 음성신호에서 잡음성분만을 제거함으로써 음성신호를 복원하는 시스템이다. 신경망 필터는 오차 역전파 학습 알고리즘에 의해 오차를 최소화 하는 방향으로 필터의 파라미터를 수정한다. 제안한 필터로 잡음환경에서의 음성신호복원 시스템을 구성하고, 실험을 통해 필터의 성능을 확인한다.

#### 1. 서 론

산업환경의 발전과 이에 동반되어 발생하는 소음공해로 인한 피해가 증가하는 추세이다. 일반적으로 소음은 상호 음성통신의 장애뿐 아니라 정신적, 육체적인 피로를 가중시켜 심신의 능력을 저하시키는 환경오염의 한 종류로서 취급된다. 소음공해로 인한 육체적, 정신적인 피해가 증가함에 따라 소음을 제거하려는 노력이 계속되어져 왔다. 또 현대에 이르러 유·무선 통신의 급속한 보급과 발전에 따라 그 중요성이 더해지고 있다. 유·무선통신에 있어서 잡음은 정확한 음성의 전달에 직접적인 악영향을 주는 요인이므로 반드시 제거해야 한다. 전동기장치를 운전중인 실내와 같이 소음이 심한 환경에서는 음성신호가 소음에 의해 오염되어 상호 음성통신 및 통신장비를 이용한 음성통신에 많은 어려움이 있다. 또, 현대에 이르러 보안의 필요성에 의해 창출된 첨단과학인 인지과학 중에서 음성인식을 이용한 화자인식등의 전처리과정으로 음성신호에서 잡음을 제거하는 것은 필수과정으로 인식되어 있다.

본 논문에서의 Speech Enhancement는 잡음환경에서의 음성신호 복원 및 음질향상을 의미한다. 2장에서는 실험한 음성신호복원 시스템의 구조와 작동원리에 대해 설명한다. 3장에서는 적용한 신경망 필터의 구조 및 작동원리에 대해 설명한다. 4장에서는 실험환경 설정과 실험결과를 보임으로써 필터의 성능을 확인한다. 시스템을 구성하고 지능적인 알고리즘을 이용하여 필터의 설계 및 시스템 적용을 그 목적으로 한다.

#### 2. 음성신호복원 시스템

##### 2.1 음성신호복원 시스템 구조

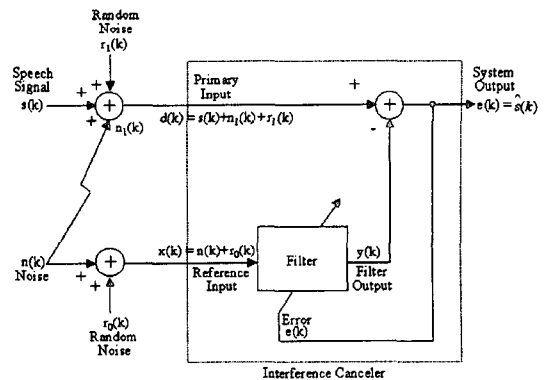


그림 1. 시스템 블록다이어그램  
Fig. 1. System Block-Diagram

그림 1은 전동기의 소음이 심한 공장내에서 전화통신을 고려한 잡음환경에서의 음성신호복원 시스템의 구성도이다.  $s(k)$ 는 복원하고자 하는 음성신호이다.  $n(k)$ 는 잡음원의 잡음신호이다.  $n_1(k)$ 는 시지연된 잡음신호이다.  $r_0(k)$ 와  $r_1(k)$ 는 임의의 잡음원으로부터의 잡음으로 고려하는 랜덤노이즈이다.  $y(k)$ 는 필터의 출력이다. Primary Input은  $s(k) + n_1(k) + r_1(k)$ 으로 설정한다. 필터의 입력인 Reference Input은  $n(k) + r_0(k)$ 으로 설정한다. 오차  $e(k)$ 는 Primary Input과 필터의 출력  $y(k)$ 와의 차로 설정한다. 결국 필터는 시지연된 잡음원신호를 추종하여 이를 상쇄시키고 랜덤노이즈를 제

거하는 역할을 담당한다.  $\hat{s}(k)$ 는 System Output이며 오차  $e(k)$ 는 같은 신호가 된다. 결국 필터는 입력으로 Reference Input을 취하며, 시지연된 잡음원신호를 추종하여 이를 상쇄시키고 랜덤노이즈를 제거하는 역할을 담당한다.  $\hat{s}(k)$ 는 System Output이며, 이는 음성신호  $s(k)$ 를 추종하며 제어기 성능이 완벽하여 오차가 없는 경우 음성신호  $s(k)$ 와 설정한 System Output  $\hat{s}(k)$ 와 오차  $e(k)$ 는 같은 신호가 된다.

만약 음성신호  $s(k)$ , 잡음원의 잡음신호  $n(k)$ , 시지연된 잡음신호  $n_1(k)$ , 필터의 출력  $y(k)$ , 임의의 잡음원으로부터의 랜덤노이즈  $r_0(k)$ 와  $r_1(k)$ 가 그 통계적 특성이 변하지 않는 statistically stationary 신호라 가정하고, 잡음원의 잡음신호  $n(k)$ 와 시지연된 잡음신호  $n_1(k)$ 사이에만 상관관계(correlation)가 존재하고, 그외의 모든 신호는 상호 uncorrelated의 관계에 있다고 가정한다면, 다음과 같은 식으로부터 오차함수를 설정하고 원하는 출력을 얻을 수 있는 지표를 생성할 수 있다.

$$e(k) = s(k) + n_1(k) + r_1(k) - y(k) \quad (1)$$

식(1)은 Primary Input인  $s(k) + n_1(k) + r_1(k)$ 와 필터 출력  $y(k)$ 의 차로 오차함수  $e(k)$ 를 설정한 식이다. 식(1)의 양변을 제곱하여 정리하면, 식(2)와 같다.

$$e(k)^2 = s(k)^2 + (n_1(k) + r_1(k) - y(k))^2 + 2s(k)(n_1(k) + r_1(k) - y(k)) \quad (2)$$

식(2)의 양변에 기대치를 취하면 식(3)과 같다.

$$E[e(k)^2] = E[s(k)^2] + E[(n_1(k) + r_1(k) - y(k))^2] + 2E[s(k)(n_1(k) + r_1(k) - y(k))] \quad (3)$$

식(3)에서  $2E[s(k)(n_1(k) + r_1(k) - y(k))]$ 는 상관관계가 없는 신호간의 기대치로 orthogonal property를 가지므로 "0"이 된다. 이를 정리하여 다시 쓰면 식(4)와 같다.

$$E[e(k)^2] = E[s(k)^2] + E[(n_1(k) + r_1(k) - y(k))^2] \quad (4)$$

음성신호 복원이 목적이므로, 식(4)에서 알 수 있듯이 음성신호의 파워  $E[s(k)^2]$ 에 영향을 주지 않고 설정한 오차함수  $E[e(k)^2]$ 를 최소화 하려면,  $E[(n_1(k) + r_1(k) - y(k))^2]$ 이 최소화되어야 한다. 그러므로 이 개념을 정리한 것이 식(5)이다.

$$E_{\min}[e(k)^2] = E[s(k)^2] + E_{\min}[(n_1(k) + r_1(k) - y(k))^2] \quad (5)$$

식(5)에서 기대치를 제거하고 정리하면 식(6)과 같은 식을 얻을 수 있다.

$$e(k) - s(k) = n_1(k) + r_1(k) - y(k) \quad (6)$$

식(6)에서 알 수 있듯이  $n_1(k) + r_1(k) - y(k)$ 이 최소

값을 가질 때는  $e(k) = s(k)$ 이다. 그러므로 필터의 역할은 Reference Input인  $n(k) + r_0(k)$ 을 입력으로 하여 필터 출력  $y(k)$ 가  $n_1(k) + r_1(k)$ 을 추종함에 있다. 이로써  $n_1(k) + r_1(k) - y(k)$ 을 "0"으로 만들 때 음성신호  $s(k)$ 를 복원할 수 있다.

### 3. LMS 알고리즘 적응필터

#### 3.1 필터의 구조

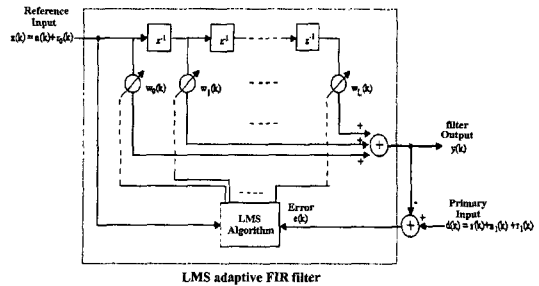


그림 2. LMS알고리즘 적응필터 세부구조  
Fig. 2. Structure of Adaptive Filter with LMS Algorithm

그림 2는 LMS알고리즘을 적용한 적응필터의 세부 구조이다. 신호처리에 있어서 LMS알고리즘은 간단하고도 빠른 연산속도 때문에 많이 사용한다. LMS알고리즘은 입력에 대해서 오프라인 아닌 온라인 처리가 가능하다. 여기서  $w_0(k), w_1(k), \dots, w_L(k)$ 는 적응필터의 각 탭 가중치이며,  $z^{-1}$ 은 이산시간에서의 한샘플 시지연을 나타낸다. 적응필터는  $n(k) + r_0(k)$ 을 입력으로 하여 필터의 출력으로  $y(k)$ 를 얻는다.  $s(k) + n_1(k) + r_1(k)$ 와  $y(k)$ 의 차로 오차  $e(k)$ 를 생성한다. 오차  $e(k)$ 와  $n(k) + r_0(k)$ 을 입력으로하여 LMS알고리즘을 동작시키며, 필터의 각 가중치를 업데이트 한다. 필터의 차수는  $L$ 차가 된다.

#### 3.2 파라미터 업데이트

입력을 행렬식으로 표현하여 이를  $X(k)$ 로 놓으면 식(7)과 같다.

$$X(k) = [x(k) \ x(k-1) \ \dots \ x(k-L)]^T \quad (7)$$

각 가중치를 행렬식으로 표현하면, 식(8)과 같다.

$$W(k) = [w_0(k) \ w_1(k) \ \dots \ w_L(k)]^T \quad (8)$$

입력이 식(7)과 같고 가중치가 식(8)과 같을 때, 적응필터의 출력  $y(k)$ 는 식(9)와 같다.

$$y(k) = X(k)^T W(k) \quad (9)$$

오차  $e(k)$ 는 desired output인  $s(k) + n_1(k) + r_1(k)$ 와 필터의 출력  $y(k)$ 의 차로써 설정한다.  $s(k) + n_1(k) + r_1(k)$ 를  $d(k)$ 로 치환하고, 오차  $e(k)$ 를 식으로 표현하면 식(10)과 같다.

$$e(k) = d(k) - X(k)^T W(k) \tag{10}$$

MSE(Mean-Square-Error)는 설정한 오차함수에 제곱을 하여 기대값을 취한 것으로  $\xi$ 라 표기하고, 이를 나타내면 식(11)과 같다.

$$\xi = E[e(k)^2] \tag{11}$$

경사하강(Gradient Descent)법에 의한 가중치 학습식은 식(12)와 같다.

$$W(k+1) = W(k) - \mu \nabla(k) \tag{12}$$

$$\left( \text{단, } \nabla(k) = \begin{bmatrix} \frac{\partial \xi}{\partial w_0} & \frac{\partial \xi}{\partial w_1} & \dots & \frac{\partial \xi}{\partial w_L} \end{bmatrix}^T \right)$$

식(12)에서 gradient  $\nabla(k)$ 는 MSE  $\xi$ 를 가중치에 대하여 편미분한 것이다. 여기서  $\mu$ 는 수렴속도와 안정도를 조절하는 양의상수이다.  $W(k)$ 는 현재  $k$ 번째 입력에 대한 가중치이고,  $W(k+1)$ 은  $k+1$ 번째 입력에 대하여 학습 될 가중치를 나타낸다. 일반적인 방법과 달리 LMS알고리즘에서는 MSE  $\xi$ 를  $E[e(k)^2]$ 로 하지 않고 확률근사(stochastic approximation)를 통하여  $e(k)^2$ 로 근사화시켜 생각한다. 이것이 LMS알고리즘의 주요한 idea로써 이를 표현하면 식(13)과 같다.

$$E[e(k)^2] \approx e(k)^2 \tag{13}$$

$E[e(k)^2]$ 를  $e(k)^2$ 로 근사화시켰으므로, 가중치 학습식과 근사값  $\hat{V}(k)$ 를 구해보면 식(14)와 같다.

$$W(k+1) = W(k) - \mu \hat{V}(k) \tag{14}$$

$$\left( \text{단, } \hat{V}(k) = \begin{bmatrix} \frac{\partial e(k)^2}{\partial w_0} & \frac{\partial e(k)^2}{\partial w_1} & \dots & \frac{\partial e(k)^2}{\partial w_L} \end{bmatrix}^T \right)$$

식(14)에서  $\hat{V}(k)$ 를 편미분하기 위해서는 Chain-Rule을 이용하여야 한다. 이를 정리하면 식(15)와 같다.

$$\begin{aligned} &\Rightarrow \left[ \frac{\partial e(k)^2}{\partial e(k)} \frac{\partial e(k)}{\partial w_0} \quad \frac{\partial e(k)^2}{\partial e(k)} \frac{\partial e(k)}{\partial w_1} \quad \dots \quad \frac{\partial e(k)^2}{\partial e(k)} \frac{\partial e(k)}{\partial w_L} \right]^T \\ &\Rightarrow \left[ 2e(k) \frac{\partial e(k)}{\partial w_0} \quad 2e(k) \frac{\partial e(k)}{\partial w_1} \quad \dots \quad 2e(k) \frac{\partial e(k)}{\partial w_L} \right]^T \\ &\Rightarrow 2e(k) \left[ \frac{\partial e(k)}{\partial w_0} \quad \frac{\partial e(k)}{\partial w_1} \quad \dots \quad \frac{\partial e(k)}{\partial w_L} \right]^T \end{aligned}$$

$$\Rightarrow -2e(k)X(k) \tag{15}$$

식(15)의 결과를 식(12)의 가중치 학습식에 대입하여 정리하면 다음식과 같다.

$$W(k+1) = W(k) + 2\mu e(k)X(k) \tag{16}$$

식(16)은 LMS알고리즘을 적용하였을 때 가중치 업데이트 식이다.  $k$ 번째 스텝에 대한 입력  $X(k)$ 과 오차  $e(k)$ , 수렴속도 및 안정도에 관여하는 양의상수  $\mu$ , 그리고 상수 2의 곱으로 이루어진 텀이  $k$ 번째 스텝에 대한 웨이트  $W(k)$ 를 업데이트시켜  $k+1$ 번째 스텝에 대한 웨이트  $W(k+1)$ 를 생성한다. 양의 상수  $\mu$ 는 step-size로서 이는 식(17)과 같은 조건을 만족해야 한다.

$$0 < \mu < \frac{1}{\lambda_{\max}} \tag{17}$$

식(17)에서  $\lambda_{\max}$ 는 입력상관행렬(input correlation matrix)  $R$ 의 고유값(eigenvalue)  $\lambda$ 중 가장 큰 값이다.  $R$ 은  $E[X(k)X(k)^T]$ 이다. 그러나 실질적으로  $R$ 을 구하기 어렵기 때문에 실제 실험에서는 1보다 작은 양의상수로 적용하며, 수렴속도 및 수렴율을 감안하여 경험에 의한 값으로 설정한다.

## 4. 신경망 필터

### 4.1 신경망 필터의 구조

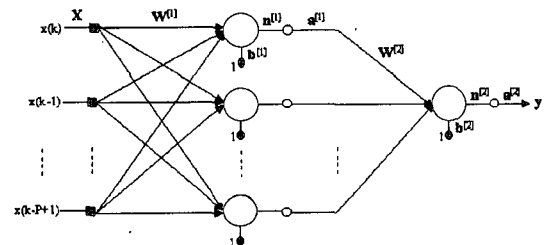


그림 3. 신경망필터의 구조  
Fig. 3. Structure of Neural-Network filter

그림 3은 신경망필터의 구성도이다. 필터입력인 Reference Input을  $x(k)$ 라 치환한다. 입력으로는  $x(k)$ 부터  $x(k-P+1)$ 까지로  $P$ 개만큼의 입력을 갖는다.

그림 4는 신경망필터 내부의 연산구조를 나타낸다. 그림 3과 4에서  $X$ 는 입력매트릭스이다.  $W^{[1]}$ 는 입력층에서 은닉층으로 연결되는 가중치 매트릭스이다.  $W^{[2]}$ 는 은닉층에서 출력층으로 연결되는 가중치 매트릭스이다.  $n^{[1]}$ 는 활성화함수를 거치기 전의 은닉층 출력 매트릭스이다.  $n^{[2]}$ 는 활성화함수를 거치기 전의 출

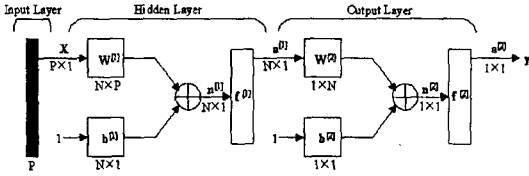


그림 4. 신경망 필터의 내부 연산구조  
Fig. 4. Internal computation of Neural-Network filter

력층 출력 매트릭스이다.  $a^{[1]}$ 는 활성화함수를 거친 은닉층의 출력이다.  $a^{[2]}$ 는 활성화함수를 거친 출력층의 출력이다.  $b^{[1]}$ 은 은닉층의 바이어스 매트릭스이다.  $b^{[2]}$ 은 출력층의 바이어스 매트릭스이다.  $y$ 는 신경망필터의 최종출력으로  $a^{[2]}$ 와 같다. 은닉층의 활성화함수  $f^{[1]}$ 과  $f^{[2]}$ 는 Hyperbolic Tangent Sigmoid 함수로 은닉층과 출력층의 활성화함수이다.

### 4.2 파라미터 학습

본 논문에서 사용한 신경망 필터의 구조는 P-N-1이다. 입력층과 은닉층, 그리고 출력층을 가지며 각 층을 연결하는 가중치를 업데이트 하는 알고리즘으로 오차 역전파 학습 알고리즘을 사용한다.

$$a^{[1]} = f^{[1]}(W^{[1]}X + b^{[1]}) \quad (18a)$$

$$y = a^{[2]} = f^{[2]}(W^{[2]}a^{[1]} + b^{[2]}) \quad (18b)$$

식(18a)는 은닉층의 출력이고, 식(18b)는 출력층의 출력으로 최종출력을 나타낸다.

$$w_{ij}^{[m]}(k+1) = w_{ij}^{[m]}(k) - \alpha \frac{\partial \hat{E}}{\partial w_{ij}^{[m]}} \quad (19)$$

$$b_i^{[m]}(k+1) = b_i^{[m]}(k) - \alpha \frac{\partial \hat{E}}{\partial b_i^{[m]}} \quad (20)$$

식(19)와 식(20)은 근사화시킨 MSE  $\hat{E} = e(k)^2$ 와 Gradient Descent방법을 이용한 각층의 가중치와 바이어스의 업데이트 일반식이다. 설정한  $\hat{E}$ 를  $w_{ij}^{[m]}$ 로 편미분하기 위해서는 Chain-Rule을 적용해야 한다. Chain-Rule을 적용하여  $\hat{E}$ 를  $w_{ij}^{[m]}$ 로 편미분하면 다음 식과 같다.

$$\frac{\partial \hat{E}}{\partial w_{ij}^{[m]}} = \frac{\partial \hat{E}}{\partial n_i^{[m]}} \frac{\partial n_i^{[m]}}{\partial w_{ij}^{[m]}} \quad (21)$$

$$\frac{\partial \hat{E}}{\partial b_i^{[m]}} = \frac{\partial \hat{E}}{\partial n_i^{[m]}} \frac{\partial n_i^{[m]}}{\partial b_i^{[m]}} \quad (22)$$

sensitivity  $s_i^{[m]}$ 를 다음식으로 정의한다.

$$s_i^{[m]} = \frac{\partial \hat{E}}{\partial n_i^{[m]}} \quad (23)$$

식(19)와 식(20)에 sensitivity를 적용시켜 매트릭스폼으로 전개하면 다음식과 같다.

$$W^{[m]}(k+1) = W^{[m]}(k) - \alpha s^{[m]}(a^{[m-1]})^T \quad (24)$$

$$b^{[m]}(k+1) = b^{[m]}(k) - \alpha s^{[m]} \quad (25)$$

$W^{[2]}$ 와  $b^{[2]}$ 를 업데이트하는데 관여하는 sensitivity를  $s^{[2]}$ 라 할 때, 이와 관계된 식은 다음과 같다.

$$s^{[2]} = -2e(k) F^{[2]'}(n^{[2]}) \quad (26)$$

$W^{[1]}$ 와  $b^{[1]}$ 을 업데이트하는데 관여하는 sensitivity를  $s^{[1]}$ 라 할 때, 이와 관계된 식은 다음과 같다.

$$s^{[1]} = F^{[1]'}(n^{[1]}) (W^{[2]')^T s^{[2]} \quad (27)$$

본 논문에서 사용한 출력층의 활성화 함수는 Pure Linear이므로 이의 미분형 매트릭스폼은 다음과 같다.

$$F^{[2]'}(n^{[2]}) = [1 - (a^{[2]})^2] \quad (28)$$

본 논문에서 사용한 은닉층의 활성화 함수는 Hyperbolic Tangent Sigmoid이므로 이의 미분형 매트릭스폼은 다음과 같다.

$$F^{[1]'}(n^{[1]}) = \begin{bmatrix} 1 - (a_1^{[1]})^2 & 0 & \dots & 0 \\ 0 & 1 - (a_2^{[1]})^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 - (a_N^{[1]})^2 \end{bmatrix} \quad (29)$$

## 5. 실험 및 결과

### 5.1 실험환경 설정

실험에 사용한 음성신호 및 잡음신호는 8[KHz] 샘플링을 거쳐 추출하며, 총 샘플의 개수는 5초동안의 샘플들로 40000개의 데이터로 구성된다. LMS알고리즘 적응필터의 경우 오프라인 학습이 필요없다. 이는 온라인으로 처리가 가능하기 때문에 학습데이터를 설정할 필요가 없지만, 신경망 필터의 경우 학습데이터를 설정해야 한다. 학습을 거친 후 테스트 데이터의 입력으로 해당하는 필터의 성능을 알 수 있다.

그림 5는 “안녕하세요. 지능신호처리 연구실 김중우입니다.”라는 음성신호로  $s(k)$ 에 해당한다.

잡음원으로서 발생하는 전동기소음을  $n(k)$  잡음원으로 설정한다. 그림 6은 Reference Input으로 잡음원과 랜덤노이즈의 합이다. 임의의 잡음원으로부터의 잡음을 랜덤노이즈로 고려하며, 미비한 신호라 가정한다.

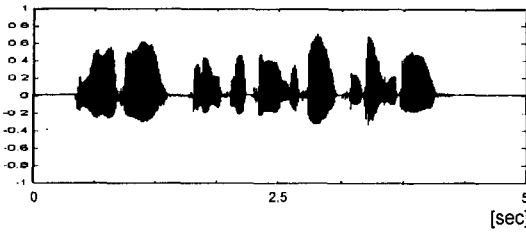


그림 5. 음성신호  $s(k)$   
Fig. 5. Speech signal  $s(k)$

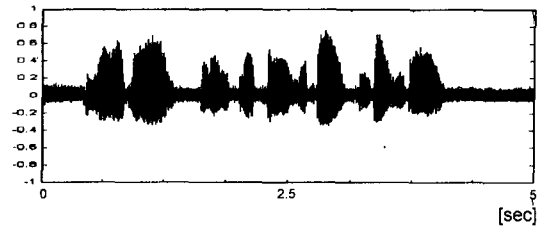


그림 8. 9차 FIR필터 경우 시스템출력  
Fig. 8. System output with 9th order FIR filter



그림 6. 필터입력  $n(k) + r_0(k)$   
Fig. 6. Filter input(Reference Input)  $n(k) + r_0(k)$

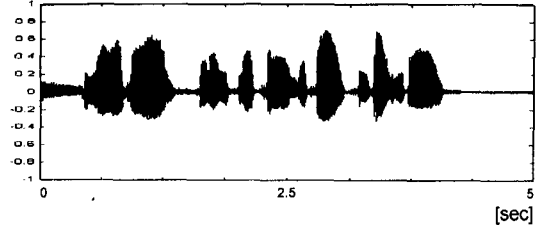


그림 9. 10차 FIR필터의 경우 시스템출력  
Fig. 9. System output with 10th order FIR filter

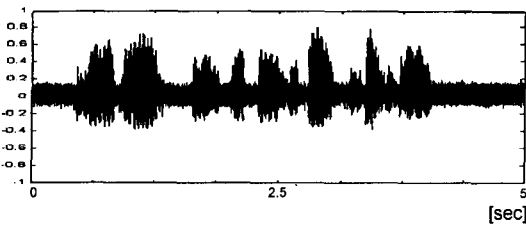


그림 7. 오염된 음성신호  $s(k) + n_1(k) + r_1(k)$   
Fig. 7. Distorted speech signal  $s(k) + n_1(k) + r_1(k)$



그림 10. 20차 FIR필터의 경우 시스템출력  
Fig. 10. System output with 20th order FIR filter

그림 7은 오염된 음성신호인 Primary Input을 나타낸다. 적용한 필터의 역할은 그림 6의 신호를 입력으로 하여 그림 7의 신호에서 잡음성분만을 제거함에 있다.

### 5.2 LMS알고리즘 적응필터 성능확인

실험에서는  $n_1(k)$ 의 시지연을 10샘플 1.25[ms]로 설정하였으며, 학습상수는 0.05이다.

그림 8은 필터차수를 9차로 설정 한 경우의 결과로 SNR은 5.80[dB]이다.

그림 9는 필터차수를 10차로 설정 한 경우의 결과로 SNR은 11.85[dB]이다. 9차보다 많은 성능향상을 알 수 있다.

그림 10은 적응필터의 필터차수 20차인 경우의 결과이다. 10차 필터와 마찬가지로 좋은 성능을 보인다. 그러나 SNR은 9.93[dB]로 10차필터에 비해 오히려 떨어짐을 확인할 수 있다. 결과로서 적절한 필터차수와 학습상수의 설정이 적응필터의 성능에 큰 영향

을 미침을 확인 가능하다.

### 5.3 신경망 필터의 성능확인

신경망 필터는 입력층과 은닉층의 뉴런수를 가변할 수 있는 구조로 설계한다. 학습상수는 경험에 의해 가장 좋은 성능을 내는 0.1로 고정하고 필터의 구조를 가변하는 경우 결과를 나타낸다. 입력층 14, 은닉층 5, 출력층 1의 구조의 신경망필터로 실험하였으며,  $n_1(k)$ 의 시지연을 10샘플 1.25[ms]로 설정한다. Iteration은 500회 실행한다. 학습데이터는 62.5[ms]의 500샘플을 취한다.

그림 11은 62.5[ms]동안의 학습데이터로 500샘플에 해당한다. 학습데이터의 개수가 너무 적으면 학습이 이루어지지 않으며 너무 많으면 계산량이 많아지기 때문에 적절한 학습데이터의 선정은 중요하며 경험에 의존한다.

그림 12는 학습을 마친 후 학습데이터에 의한 시스

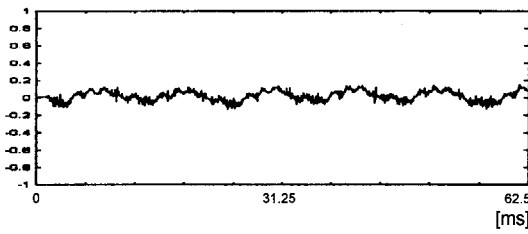


그림 11. 학습데이터  
Fig. 11. Training data

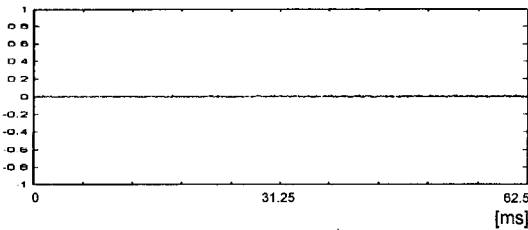


그림 12. 학습을 마친 후 학습데이터에 의한 시스템출력  
Fig. 12. System output with training data after BP training

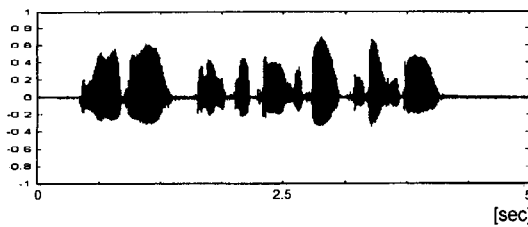


그림 13. 테스트 데이터를 입력한 경우 시스템 출력  
Fig. 13. System output with test data

템출력을 나타낸다.

그림 13은 학습데이터에 의해 오차를 최소화 하는 방향으로 튜닝된 웨이트와 바이어스를 가진 신경망필터에 전체 테스트데이터를 입력하여 얻은 결과로 SNR은 약 26.36[dB]로 좋은 성능을 보인다. 참고로 신경망의 구조가 10-3-1과 10-5-1인 경우 각각의

SNR은 19.69[dB]와 20.15[dB]이다. 신경망의 구조를 14-5-1 이상으로 설정한 경우 오히려 SNR이 떨어짐을 확인할 수 있으며, LMS알고리즘 적응필터와 마찬가지로 적절한 필터의 구조설정이 성능에 큰 영향을 미침을 확인할 수 있다.

## 6. 결 론

본 논문에서는 잡음에 의해 오염된 음성신호를 복원하여 음질을 향상시키기 위한 시스템 구축을 목적으로 하며, 이는 Interference Canceling 모델을 기본으로 한다. 시스템의 필터부분에는 LMS알고리즘 적응필터와 신경망필터를 적용한다. 실험결과에서 알 수 있듯이 두가지의 필터를 시스템에 적용시 좋은 성능을 보임을 확인할 수 있다.

## 참고문헌

- [1] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*, Prentice-Hall, 1985.
- [2] S. Haykin, *Adaptive Filter Theory*, Upper Saddle River, New Jersey, Prentice-Hall, 1996.
- [3] Hagan, Demuth, Beale, *Neural Network Design*, PWS Publishing Company, 1996.
- [4] W. G. Knecht and M. E. Schenkel, "Neural Network Filters for Speech Enhancement," *IEEE Trans. on Speech And Audio Processing*, Vol. 3, No. 6, 433, Nov. 1995.
- [5] Sen M. Kuo and D. R. Morgan, *Active Noise Control Systems : Algorithms and DSP Implementations*, John-Wiley and Sons, 1996.
- [6] L. H. Tsoukalas and R. E. Uhrig and L. A. Zadeh, *Fuzzy and Neural Approaches in Engineering*, John-Wiley and Sons, 1994.
- [7] W. G. Knecht and M. E. Schenkel, "Neural Network Filters for Speech Enhancement," *IEEE Trans. on Speech And Audio Processing*, Vol. 3, No. 6, 433, Nov. 1995.
- [8] B. Widrow *et al*, "Adaptive noise cancelling : principles and applications," *Proc. IEEE*, Vol. 63, 1692-1716, Dec. 1975.



**김 종 우 (Jong-Woo Kim)**  
 1995년 : 원광대학교 전기공학과(공학사)  
 2000년 : 숭실대학교 전기공학과(공학 석사)  
 2000년 7월~현재 : (주)해태전자 전자연구  
 구소 연구원  
 관심분야 : 신호처리, 퍼지시스템 및 제어, 신경회로망



**공 성 곤 (Seong-Gon Kong)**  
 1982년 : 서울대학교 전기공학과(공학사)  
 1987년 : 서울대학교 대학원 전기공학과  
 (공학석사)  
 1991년 : Univ. of Southern California  
 (공학박사)  
 1992년 3월~현재 : 숭실대학교 전기공  
 학과 부교수

주관심분야 : 지능시스템이론, 영상 및 신호처리, 로보틱스, 진화연산, 인공지능