

# SUB-BAND 적응 디지털 필터 실시간 시스템 구현에 관한 연구

## A Study on Real Time Implementation of an Adaptive Digital Filter Using a Sub-band Structure

류 차 회\*, 유 재 하\*, 차 일 환\*, 윤 대 회\*  
(Cha-Hee Yoo, Jae-Ha Yoo, Il-Whan Cha, Dae-Hee Youn)

### 요 약

충격 응답 시간이 긴 시스템을 모델링하기 위한 실시간 적응 디지털 필터를 구현하였다. 대상 시스템의 충격 응답 시간이 길 때, 일반적인 적응 디지털 필터를 사용하는 경우 발생하는 수렴 속도 저하와 계산량 증가 문제를 해결하기 위해서 서브밴드 구조를 갖는 적응 디지털 필터를 구성하였다.

실시간 처리 시스템에서는 GQMF(Generalized Quadrature Mirror Filter)을 사용하여 입력 신호를 4개 대역으로 분할하여 각 대역별로 적응 필터링을 수행함으로써 수렴 속도를 향상시킨다. 또한 대역별 신호를 동시에 분산 처리하기 때문에 계산량 면에서 효율적이므로 시스템의 충격 응답이 긴 경우에는 실시간 처리가 가능하다.

하드웨어 구성은 범용 신호 처리 프로세서인 DSP56001을 호스트(host)프로세서로 사용하며, 적응 디지털 필터 칩 DSP56200을 사용하여 각 대역 적응 필터를 구성하였다.

실험은 충격 응답 시간이 16 kHz 필터링 시 2000 탭 길이로 가정된 시스템을 대상으로 부동 소수점(floating point) 시뮬레이션 결과와 실시간 처리 시스템의 결과를 비교하였다. 밴드를 나누지않은 기존의 방법과 서브밴드(subband) 시스템의 비교 실험 결과 입력이 백색 잡음인 경우 대역별 간섭에 의한 성능 저하가 있었으나, 유성과 유사한 특성을 갖는 유색 잡음인 경우 서브밴드 시스템이 단일 시스템에 비해 성능 향상을 보였다.

### ABSTRACT

When a conventional adaptive FIR(Finite Impulse Response) filter has many coefficients, it produces large computational complexity and slow convergence.

In this paper, an adaptive FIR digital filter using a sub-band structure is implemented to overcome these problems. The filter bank consists of GQMF(Generalized Quadrature Mirror Filter). Input signal is divided into 4 bands and each band signal is processed separately.

The real time system is implemented using the DSP56001, a general purpose signal processor, as a host processor and the DSP56200, a cascable adaptive FIR filter peripheral chip, as an adaptive filter for each band.

The simulation system has 2000 taps at a 16 kHz sampling rate. The real time system is compared with the computer simulations. From the results of computer simulations, the subband adaptive filter shows better performance than conventional adaptive filter for colored signal such as speech signal.

\*연세대학교 전자공학과  
접수일자: 1993년 6월 22일

### 1. 서 론

원거리 회의와 수자유(hand-free) 전화 통화시 발생하는 반향을 제거하거나, 음장(sound field)을 제어하는 문제들을 해결하기 위해서는 총각 응답 시간이 긴 시스템을 적응적으로 추정하는 방법이 필요하다[1][2].

이를 위해서 LMS(Least Mean Square) 알고리즘으로 필터의 계수를 조정해가는 적응 FIR 필터(Adaptive Finite Impulse Response Filter)를 사용하는 방법은 필터 탭수가 증가하기 때문에 고유치 분포율이 커져 수렴이 늦어지고, 계산량이 증가하여 실시간 구현에 많은 제한을 받는다[3].

이러한 문제를 해결하기 위한 몇 가지 연구가 진행되어 왔다. 탭수를 줄이는 방법으로, IIR(Infinite Impulse Response) 필터를 사용하는 경우[4]와 Legendre Laguerre 함수 등 직교 함수를 사용하여 탭수를 줄이는 방법[5] 등이 제안되었으나 시스템 안정성에 문제가 있으며, 수렴 속도가 늦어지는 단점이 있다.

수렴 속도를 향상시키는 방법으로는, 전체 탭 몇 개의 부분으로 나누어 계수를 조정할 때 가중치를 따로 가하는 방법[6]과 전단에 화이트닝 필터(Whitening Filter)를 달아 고유치 분포율을 작게하여 수렴을 빨리하도록 하는 방법[7] 등이 있으며, 입력 신호를 몇 개의 주파수 대역으로 구분하여 각 대역별로 적응 필터링 과정을 수행하는 서브밴드 적응 필터링(Subband Adaptive Filtering) 방법등이 제안되었다[8].

## II. 서브밴드 적응 디지털 필터

### 1. 시스템 추정 및 적응 알고리즘

신호의 통계적 특성을 모르거나 시변 상황에서 시스템을 추정할 때 계산이 간편하고 효과적인 계수 추정 방법으로 LMS(Least Mean Square) 알고리즘이 사용된다[11]. LMS 알고리즘으로 적응 필터 계수를 조정하는 식은 다음과 같다.

$$\hat{W}(k+1) = \hat{W}(k) + 2\mu e(k) X(k) \quad (1)$$

여기서  $\mu$ 는 수렴 속도를 결정하는 상수이며 시스템의 안정성을 고려하면 다음 식을 만족해야한다[11].

$$0 < \mu < (1/\lambda_{max}) \quad (2)$$

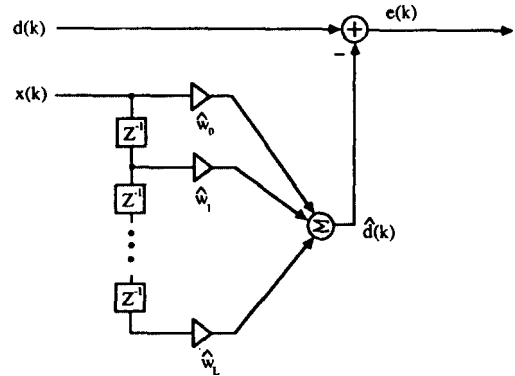


그림 1. 시스템 추정 필터 구조

여기서  $\lambda_{max}$ 는 참고 입력 신호 자기 상관 행렬  $R$ 의 최대 고유치(eigenvalue)를 나타낸다. LMS 알고리즘의 수렴 속도는 고유치 분포율에 의존하게 된다.

### 2. 서브밴드 적응 디지털 필터 구조

실제 응용에 있어서 대상 시스템의 총각 응답이 매우 길거나 입력 신호의 고유치 분포율에 큰 경우, LMS 알고리즘의 수렴 속도가 늦어지며, 계산량 증가로 실시간 구현에 제한을 받게된다. 이러한 현상은 입력 신호를 몇 개의 주파수 대역으로 나누어 각 대역별로 적응 필터링을 수행함으로써 각 대역별로 고유치 분포율을 낮추며, 계산량 면에서 효율적인 서브밴드 적응 필터링 방법으로 해결할 수 있다. 본 논문에서 구성한 서브밴드 적응 디지털 필터 구조는 그림 2와 같다.

서브밴드 적응 필터는 세 부분으로 나뉘며, 그 동작은 먼저 참고 입력 신호와 주입력 신호를 각각 분석 필터에 통과시켜 각 대역 적응 필터의 입력으로 사용한다. 적응 필터의 출력을 합성 필터에 통과시켜 그 합이 오차 신호가 된다. 각 대역 적응 필터 계수는 각 대역에서 발생한 오차 신호  $e_i(m)$ 을 사용하여 식 (1)과 같이 조정한다.

서브밴드로 나누지 않았을 때 2000개의 필터 길이를 사용하게 되는 경우 4개 밴드로 나누어 처리하면 4:1로 간축(decimation)하게 되므로 각 대역당 500개의 필터 길이를 갖는다.

대역당 계산량이 1/4로 감소하게되어 실시간 구현이 용이하다. 밴드를 나누지 않은 경우와 M 밴드로 나눈 경우 복잡 수를 비교하여 표 1에 나타냈다.

표 1에서 알 수 있듯이 M 밴드로 나누어 처리하면 계산량이 약 1/M로 줄어들게 되어 실시간 구현이 용

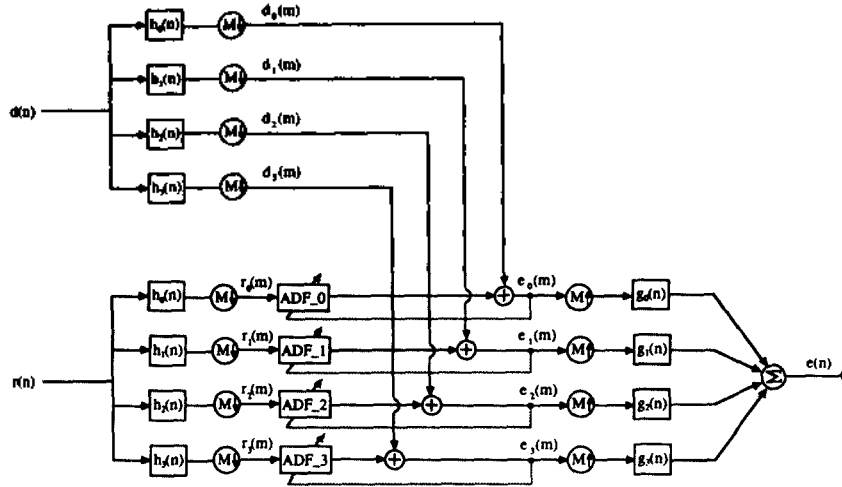


그림 2 서브밴드 적응 디지털 필터 구조

이하다.

표 1. M 밴드로 나눈 경우 급셈 수 비교

밴드를 나누지 않은 경우	2 * Tap_Num
M 밴드로 나눈 경우	2 * Tap_Num/M + (M + 1) * fp_Num

Tap\_Num : 밴드를 나누지 않은 경우 적응 필터 탭 수

M : 밴드 수

fp\_Num : 필터뱅크 탭 수

### 3. 필터뱅크

필터뱅크는 QMF(Quadrature Mirror Filter)를 일반화한 GQMF(Generalized QMF)를 사용하였다. 먼저 식 (3)을 만족하도록 프로토타입 저역 통과(prototype low-pass) 필터를 설계하였다[9].

$$E = E_r + \alpha E_s(f_{SB}) \quad (3)$$

$$\text{여기서 } E_r = 2 \sum_{\omega=0}^{\pi/2M} [H^2(\omega) + H^2(\pi/M - \omega) - 1]^2$$

$$E_s = \sum_{\omega=f_{SB}}^{\pi} (H^2(\omega))$$

$\alpha$  : 저지대역가중치

$f_{SB}$  : 저지대역 에지(edge)

식 (3)에서 E를 최소화하도록 h(b)을 구하는 알고리즘으로 Hooke와 Jeeves가 제안한 방법을 사용하였다. 이렇게 구한 h(n)을 식 (4)를 사용하여 각 대역

에서 변조시켜 분석 필터로 사용하였다[10].

$$h_k(n) = \sin[(\pi/2M)(2k-1)(n-(N-1)/2) + \theta(k)] h(n) \quad (4)$$

$$k = 1, 2, \dots, M \quad n = 0, 1, \dots, N-1$$

여기서 N은 각 밴드별 적응 필터의 탭수이며 각 밴드의 중심 주파수는  $(\pi/2M)(2k-1)$ 이다. GQMF에서는  $|\theta(k) - \theta(k-1)| = \pi/2$ 가 되면 이웃 대역간에 간섭 현상을 제거할 수 있다. 합성 필터는 식 (5)와 같이 분석 필터의 시간축 역(time-inverse)된 형태이다.

$$g_k(n) = M h_k(N-n-1) \quad (5)$$

### III. 실시간 처리 시스템

실시간 처리 시스템은 호스트 프로세서(host processor)로 DSP56001을 사용하였고, 적응 디지털 필터로서 DSP6200을 사용하였다. 이들의 특성과 전체 하드웨어 구성, 그리고 소프트웨어를 다음에 설명하였다.

#### 1. DSP56001과 DSP56200

##### 1-1. DSP56001 [12]

호스트 프로세서로 사용한 DSP56001은 다음과 같은 특징을 갖고 있다.

기본적으로 데이터 연산부, 어드레스부, 그리고 프로그램 제어부 등으로 나뉜다. 한번의 사이클에 24\*

24 비트 곱셈을하며 이를 저장하는 56비트 아큐뮬레이터를 2개 갖고 있다. 어드레스부에는 어드레스, 옵션, modifier 레지스터를 각각 8개씩 24개의 16비트 레지스터로 구성된다.

메모리는 24비트 512 워드 프로그램 메모리, X 데이터 메모리, Y 데이터 메모리로 구성되며 64 K 까지 확장이 가능하다.

1-2. DSP56200 [13]

(1) 동작 모드

DSP56200은 칩 하나로 최대 256 탭을 갖는 FIR 필터로써 칩을 여러개 연결하여 쉽게 탭수를 늘려 쓸 수 있으며 sum of product 연산을 한다.

하나의 FIR 필터(SFIR), 두개의 독립된 FIR 필터(DFIR), 그리고 적응 FIR 필터(AFIR)등 세가지 모드로 동작한다. 두개의 독립된 FIR 필터로 사용될 때는 128 탭을 각각 사용한다. AFIR 모드로 사용할 때는 계수  $h(i)$ 는 LMS(Least Mean Square) 알고리즘을 사용하여 조정한다. 출력은 원하는 신호와 추정된 신호 차로 식 (6)과 같이 계산된다.

$$y(n) = d(n) - \sum_{i=0}^{N-1} h(i) x(n-i) \quad (6)$$

계수를 조정하는 식은 식 (7)과 같다.

$$h_i(n+1) = h_i(n) + K e(n) x(n-i) \pm leakage \quad (7)$$

여기서 K는 수렴속도를 결정하는 상수이다.

(2) 기능별 구조

각 부분의 기능별 블록 다이어그램을 그림 3에 나타냈다.

16비트 계수 메모리와 24비트 데이터 메모리가 각각 256개씩 있고, 이들을 가지고 sum of product 연산을 하는 연산 장치가 있다. 또다른 DSP56200과 연결하여 사용할 때는 cascade 인터페이스를 사용한다. 비동기 패러렐 인터페이스는 데이터를 읽거나 쓸 때 사용한다. 이때는 읽기, 쓰기 그리고 칩선택 핀으로 동작하는데, 네개의 어드레스 선으로 8비트 데이터 버스를 통해 호스트 프로세서의 데이터를 DSP56200 내에 레지스터에 쓰거나 혹은 읽게된다. DSP56200 클럭은 최대 10.25 MHz를 사용하므로 하나의 곱셈-덧셈 연산을 하는데 걸리는 시간은 97.5ns이다. START 핀은 표본화율 (sampling rate)을 결정하고 적응 필터

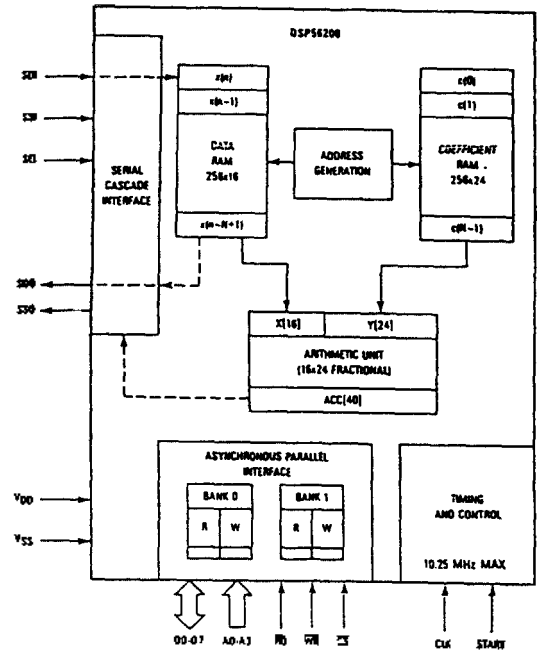


그림 3. DSP56200 기능별 블록 다이어그램

로 사용할때 START 펄스가 한번 발생할 때마다 필터 계수가 조정된다. 데이터를 읽고 쓸 때는 두 START 펄스 사이에서 이루어져야 한다.

2. 실시간 시스템 하드웨어 구성

그림 4에는 서브 밴드 적응 디지털 필터의 하드웨어 블록도를 나타냈다. 서브 밴드 적응 디지털 필터는 두 채널 A/D, 한 채널 D/A 보드와 4채널 적응 필터링을 하는 DSP56200 보드, 전체 시스템을 제어하

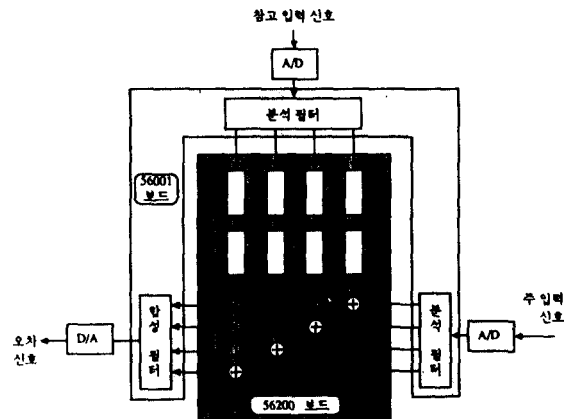


그림 4. 하드웨어 블록도

는 DSP56001 보드로 구성하였으며, 분석 필터와 합성 필터는 DSP56001에서 처리하도록 하였다.

참고 입력 신호와 주 입력 신호는 각각 A/D 변환기를 거쳐 이산 신호로 바뀐다. 이 이산 신호는 DSP56001에 구성된 분석 필터를 거쳐 각 대역 신호로 나뉘며, DSP56200 보드에서 대역별 참고 입력과 주 입력으로 들어간다. DSP56200 보드에서의 출력 신호 즉, 오차 신호들은 다시 DSP56001 보드에 구성된 합성 필터를 거쳐 전체 오차 신호가 된다. 이 전체 오차 신호를 D/A 변환기에 통과시켜 전송한다. 적응 필터는 한 대역당 2개의 DSP56200을 사용하여 4개 대역에 총 8개의 DSP56200을 사용하였다.

### 2-1. DSP56001과 DSP56200의 인터페이스

DSP56001과 DSP56200의 연결을 그림 5에 나타냈다. 데이터 버스(D0-D7), 어드레스 라인(A0-A3), 읽기 핀(RD), 쓰기 핀(WR)을 직접 연결하여 데이터 입, 출력으로 사용하였다. X/Y, DS, 그리고 12개의 어드레스 라인(A4-A15)을 어드레스 디코더 입력으로 사용하여 디코더 출력을 DSP56200 칩 선택과 START로 사용하였다. 칩 선택은 DSP56200 여덟개 각각 따로 주어지며 START는 여덟 개에 동시에 주어지도록 하였다. 적응 필터로 사용시 입력은 두 입력 신호와 참고 입력 신호가 되고 출력은 오차 신호

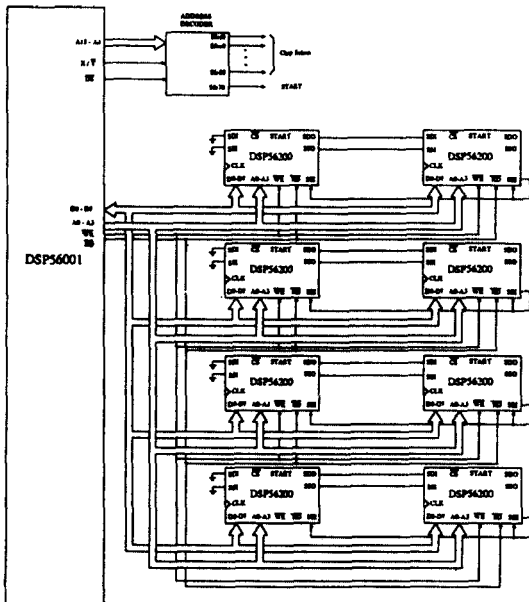


그림 5. DSP56001과 DSP56200 연결

가 되므로 두 START 펄스 사이에서 일어나는 칩 선택은 세번 있게된다.

### 2-2. 사용한 클럭수

데이터를 1개의 분석 필터와 4개의 합성 필터를 통과시키는데 740개의 클럭을 사용하였고 DSP56200에 데이터를 입력시키고 출력을 얻는데 170개의 클럭을 사용하였다. 데이터 I/O를 하는데 60개 클럭을, 기타 40개 클럭을 사용하여 총 1010개 클럭을 사용하였다. 16 kHz로 샘플링할 때에는 1280개 클럭까지 허용되므로 최대 사용클럭은 80%를 사용하게 된다. 따라서 실시간 처리가 가능하다.

## IV. 실험 및 결과 고찰

실시간 시스템의 성능 평가는 컴퓨터 시뮬레이션 을 통한 부동 소수점(floating point) 연산 결과와 비교하여 수행하였다. 또한 일반적인 적응 필터링 방법과 서브밴드 적응 필터링 방법을 비교하는 실험에서는, 입력 신호로서 고유치 분포율이 작은 백색 잡음 신호와 고유치 분포율이 큰 모의 음성 신호에 대해 실험하였다.

추정하려는 시스템의 충격 응답 시간은 125 ms로 하고, 시스템의 성능을 평가하기 위한 방법으로, 다음 두 가지 항목에 대해 실험하였다.

#### (1) ERLE(Echo Return Loss Enhancement)

시스템이 원거리 회피시 발생하는 반향을 제거하는 방향 제거기로 사용되는 경우에 성능 평가 방법으로, 반향 신호 자승 평균과 오차 신호 자승 평균의 비를 나타낸다.

$$ERLE(n) = 10 \log \frac{\sum_{k=n-N/2}^{k=n+N/2} y^2(k)}{\sum_{k=n-N/2}^{k=n+N/2} e^2(k)} \quad (8)$$

#### (2) 오차 신호의 스펙트럼

서브밴드 적응 필터링 과정에서 발생하는 인접 대역간의 간섭 현상을 관찰하기 위해 2초 경과후의 오차 신호에 대한 주파수 스펙트럼을 관찰하였다.

$$E(f) = 10 \log \frac{1}{N} \sum_{k=0}^{k=N} |DFT[e(k)]|^2 \quad (9)$$

1. 실험-I (백색 잡음 신호에 대한 실험)

먼저 2초 경과했을 때의 오차신호의 스펙트럼을 그림 6에 나타냈고, ERLE 곡선을 그림 7에 나타냈다.

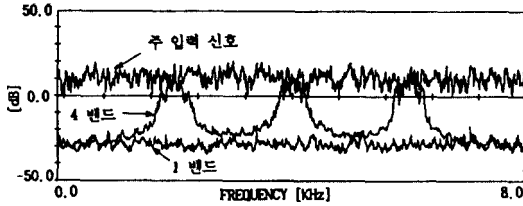


그림 6. 백색잡음 입력시 오차신호 스펙트럼

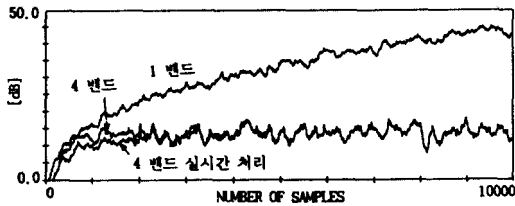


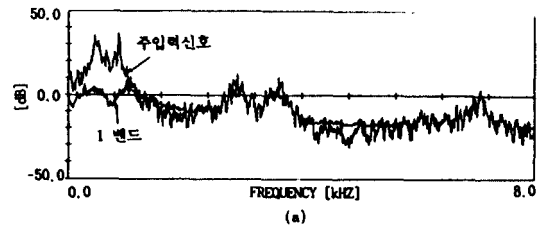
그림 7. 백색잡음 입력시 ERLE 곡선

이 실험에서는 서브밴드 방법보다 밴드를 나누지 않고 필터링한 경우 좋은 성능을 얻었다. 그림 6에서 밴드를 나누지 않고 필터링한 경우에는 오차 신호의 스펙트럼이 모든 대역에서 고르게 나타나지만, 4 밴드로 나눈 경우에는 이웃 대역 사이에서는 오차 신호가 많이 남아있는 것을 관찰할 수 있다. 이러한 이유는 적응 필터링하는 과정에서 이웃 대역 사이에 간섭 현상이 심하게 생기기 때문이다. 대역 중심에서는 밴드를 나누지 않고 처리한 경우와 비슷하지만 대역과 대역 사이에서는 오차 신호가 줄지 않음을 알 수 있다. 그림 7에서는 4 밴드인 경우 초기 오차도 크고 장상 상태에서의 오차는 더 크게 나타남을 알 수 있으며, 실시간 시스템에서 정상 상태에서는 부동 소수점 시뮬레이션 결과와 같지만 시작 부분에서 오차가 다소 크게 나타남을 알 수 있다.

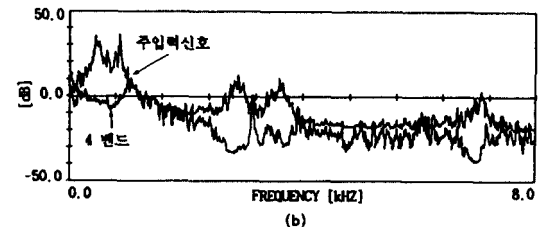
2. 실험-II (모의 음성 신호에 대한 실험)

서브밴드 적응 필터 시스템이 반향 제거와 같이 고유치 분포율이 큰 신호 환경에서 사용되는 경우에 대한 성능 평가 실험을 하였다. 입력 신호는 음성발생 시스템에서 추출하였다. 오차 신호의 스펙트럼, 그리고

오차 ERLE 곡선을 그림 8, 그림 9에 각각 나타냈다. 이 실험에서는 그림 9에서 알 수 있듯이 밴드를 나누지 않았을 때 보다 4 밴드로 나누어 처리한 경우에서 오차 신호가 작고 좋은 성능을 얻었다. 그러나 초기 오차가 다소 크게 나타남을 알 수 있다. 이러한 이유는 적응 필터링 초기에 이웃 대역간에 오차 신호가 크게 분포하기 때문에 일어나게 된다. 실시간 처리 시스템에서는 컴퓨터 부동 소수점 시뮬레이션에 비해 초기 수렴이 좋지 않다는 것을 알 수 있다. 정상 상태에서의 ERLE는 4 밴드인 경우 5 dB 더 좋은 성능을 나타냈다.



(a)



(b)

그림 8. 모의 음성 입력시 오차신호 스펙트럼

(a) 주입 음성 신호와 밴드를 나누지 않고 처리한 경우  
(b) 주입 음성 신호와 4 밴드로 나누어 처리한 경우

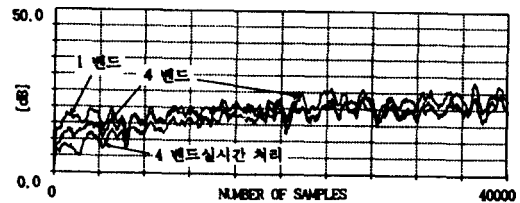


그림 9. 모의 음성 입력시 ERLE 곡선

그림 8에서 보듯이 밴드를 나누지 않고 적응 필터링한 경우에는 주 입력 신호의 첫번째와 두번째 포먼트(formant) 성분까지만 줄었지만, 4 밴드로 나누어 처리한 경우에는 주 입력 신호의 모든 포먼트 성분

이 줄었다. 이러한 이유는 LMS 알고리즘이 입력 신호 에너지가 많은 대역을 중심으로 필터링하기 때문이다. 밴드를 나누지 않은 경우는 에너지가 가장 많은 1 kHz를 중심으로 적응 필터링을 하지만, 4 밴드인 경우는 각 대역에서 에너지가 가장 큰 부분을 중심으로 적응 필터링을 하기 때문이다. 또한, 4 밴드로 나누어 처리하면 입력신호의 주파수 동적 범위가 작게되어 고유치 분포율이 줄어들기 때문에 전체적으로 밴드를 나누지 않고 처리한 경우보다 좋은 성능을 얻을 수 있었다.

## V. 결 론

본 논문에서는 충격 응답이 긴 시스템을 모델링하는 데 있어서 실시간 처리가 가능하도록 서브 밴드 구조를 갖는 적응 필터를 구현하였다. 실시간 처리 시스템에서는 범용 신호 처리 프로세서인 DSP56001과 적응 디지털 필터 칩인 DSP56200을 사용하여 4 밴드 적응 필터링 동작을 수행하도록 구성하여 16 kHz로 샘플링하였을 때, 2000 샘플 길이의 충격응답시간에 대한 실시간 처리가 가능함을 확인하였다.

125 ms의 긴 충격 응답 특성을 갖는 시스템을 대상으로 수렴 상태, 오차 신호 및 정상 상태에서의 오차의 스펙트럼을 관찰하였다. 4 밴드로 나누어 적응 필터링한 경우 빠른 수렴 상태를 보였고, 초기 오차가 다소 크게 나타났다. 음성 신호와 유사한 유색 잡음 환경에서 5 dB 향상된 성능을 보였으나, 대역간의 간섭 현상에 의한 성능 저하 현상도 관찰되었다. 앞으로의 과제는 적응 필터링을 하는 과정에서 발생하는 간섭 현상을 효율적으로 제거하는 방법과 이를 실시간 처리하기 위한 시스템에 대한 연구가 필요하다.

## 참 고 문 헌

1. Olivier Muron and Jacques Sicorav, "Modelling of Reverberators and Audioconference Rooms," *Proc. ICASSP*, pp. 921-924, 1986.
2. Jacques Sicorav, "Expeeriments to Identify and Track Nonstationites in Audioconference Rooms," *Proc. ICASSP*, pp. 2566-2569, 1986.
3. H. Yasukaw, S. Simada and I. Furukwa, "Acoustic Echo Canceller with High Speech Quality," *Proc. ICASSP*, pp. 2125-2128, 1987.
4. Hong Fan and W. K. Jenkins, "An Investigation of an Adaptive IIR Echo Canceller Advantage and

- Problem," *IEEE Trans. on ASSP* Vol. 36, No. 12, pp. 1819-1834, Dec. 1988.
5. Hector Perez and Shigeo Tsujii, "A System Identification Algorithm Using Othogonal Function," *Proc. ICASSP*, pp. 752-755, 1991.
6. Shoji Makino and Yutaka Kaneda, "Acoustic Echo Canceller Algorithm Based on The Variation Characteristics of a Room Impulse Response," *Proc. ICASSP*, pp. 1133-1136, 1990.
7. C. Acker, P. Vary and H. Ostendrap, "Acoustic Echo Cancellation Using Prediction Residual signal," *Proc. Eurospeech*, pp. 1297-1300, 1991.
8. Andre Gilloire, "Experiments sub-band Acoustic Echo Canceller for teleconferencing," *Proc. ICASSP*, pp. 2141-2144, 1987.
9. J.D. Johnston, "A Filter Family Designed for Use in Quadrature Mirror Filter Banks," *Proc. IEEE*, pp. 291-294, 1980.
10. H.J. Nassbaumer and M. Vetterli, "Computationally Efficient QMF Filter Banks," *Proc. ICASSP*, pp. 11.3.1-11.3.4, 1984.
11. B. Widrow and S.D. Stearns, *Adaptive Signal Processing*, Prentice-Hall Inc., 1991.
12. *DSP56000/DSP56001 Digital Signal Processor User's Manual*, Motorola Inc., 1990.
13. *Motorola Semiconductor Technical Data. DSP56200*, Motorola Inc., 1988.

▲류 차 희

1966년 6월 14일생



1986년 3월 ~ 1990년 2월(공학사)  
1990년 9월 ~ 1992년 8월(공학석사)  
1992년 9월 ~ 현재(공학박사 과정)

▲유 재 하

1966년 11월 21일생



1986년 3월 ~ 1990년 2월(공학사)  
1990년 3월 ~ 1992년 2월(공학석사)  
1992년 3월 ~ 현재(공학박사 과정)

▲차 일 환 1990년 9권 6호 참조

▲윤 대 희 1990년 9권 6호 참조