

논문 2021-16-31

단일칩 능동 소음 제어기 및 평가 시스템 개발 (Development of A Single-Chip Active Noise Controller And Its Evaluation System)

정 익 주*
(Ikjoo Chung)

Abstract : In this paper, we developed the evaluation system for the active noise control so that the algorithms can be easily evaluated in real-time on the system. We implemented the active noise controller based on a single-chip with only additional op-amps for signal conditioning because the TMS320C280049 MCU includes almost all necessary peripherals for the active noise controller. Due to the difficulty in testing algorithms on embedded-type hardware unlike in computer simulation, we also developed GUI-based evaluation software which makes it simple to test algorithms on the hardware. Using the GUI software, we can optimize the parameters of the algorithms with ease in a specific noise environment because the parameters can be adjusted in real-time when the algorithm is running on the hardware.

Keywords : Active noise control, Evaluation system, FxLMS, TMS320C280049 MCU, GUI software

1. 서 론

능동 소음 제어 기술의 개념이 등장한 지는 상당히 오래 되었지만, 이를 구현할 수 있는 제반 기술이 따라주지 못해 구현에 어려움을 겪다가 최근, 디지털 신호처리 기술의 발전 및 고속 연산이 가능한 프로세서 덕분에 구현이 가능해지면서 여러 분야에서 이 기술의 상용화가 활발히 이루어지고 있다 [1-3]. 활발한 상용화에도 불구하고, 현재의 능동 소음 제어 기술은 기술적 한계로 인하여, 상용화 가능한 응용이 상당히 제한된다. 이렇게 응용 분야가 제한되는 가장 큰 이유는 능동 소음 제어가 소음을 제거하는 유효한 영역인 정숙 영역 (quiet zone)이 좁기 때문이다. 따라서 현재의 능동 소음 제어 응용 분야는 좁은 정숙 영역으로도 효과가 있는 응용이 대부분이다. 헤드폰과 덕트에 적용된 사례가 대표적인 경우이다 [4, 5]. 헤드폰의 경우는 이어컵 (ear cup) 내부가 정숙 영역에 해당하며, 덕트의 경우는 덕트 내부 공간 중 한 단면에 해당한다. 덕트 내부에서는 음파가 평면파를 이루기 때문에 한 지점에서 소음이 제거되면 이론적으로는 그 이후 공간에서는 소음이 사라지게 된다. 단, 덕트의 경우도 덕트의 단면적이 넓을 경우 소음 제거 효과가 떨어지게 된다.

한편, 좁은 정숙 영역을 넓히기 위해 다채널 능동 소음 제어 기술이 개발되었다. 다수의 오차 마이크와 제어 스피커를 사용하여 개별적인 정숙 영역을 오버랩 시킴으로써 전체 정숙 영역을 넓히는 방법이다. 다채널 능동 소음 제어 기술 적용 사례의 대표적인 경우가 자동차 실내의 소음을 제거하는 응용이다 [6]. 비교적 좁은 공간이라고 할 수 있는

자동차 실내 공간 정도의 소음을 다채널 능동 소음 제어 기술을 이용하여 제거하기 위해서는 자동차에 장착된 다수의 스피커와 별도로 4~6개의 오차 마이크를 필요로 한다.

상용화에 성공하여 대량 생산되는 제품인 능동 소음 제어 헤드폰의 경우는 가격을 낮추고 전력 소비를 줄이기 위해 ASIC화된 능동 소음 제어를 포함하고 있다. ASIC이 가격적인 측면이나 전력 소비적인 측면에서 유리하기는 하지만, 초기 개발비가 많이 드는 단점이 있다. 특히, 해당 기술의 시장성이 검증되지 않은 응용 제품에 적용할 경우, ASIC과 같이 초기 비용이 많이 소요되는 방식은 적합하지 않다. 따라서 능동 소음 제어 기술이 시험적이면서도 창의적인 다양한 응용 분야에 적용되기 위해서는 저가의 범용 MCU를 이용한 상용화가 가능해야 한다 [7].

본 연구에서는 상기의 목적에 부합하고자 저가의 MCU를 이용한 단일칩 능동 소음 제어기를 개발하였다. 뿐만 아니라, 개발된 능동 소음 제어기에 다양한 알고리즘을 탑재하고 여러 제어 파라미터들을 실시간으로 조정하면서 능동 소음 제어기를 평가 할 수 있는 GUI 기반의 평가 소프트웨어도 함께 개발하였다. 이 평가 소프트웨어는 개발된 하드웨어와 연동하여 사용자가 여러 탑재 알고리즘의 제어에 필요한 다양한 파라미터들을 실시간으로 조정할 수 있게 한다. 이렇게 함으로써, 컴퓨터 모의 실험으로는 확인할 수 없는 실시간 동작 시의 여러 문제점들을 점검하고 최종으로 실시간으로 검증된 알고리즘을 제품에 탑재할 수 있도록 한다.

논문의 구성은 다음과 같다. 2장에서는 능동 소음 제어기 하드웨어 및 탑재된 다채널 FxLMS 알고리즘들에 대하여 소개하고, 3장에서는 하드웨어와 연동하여 동작하는 GUI 기반의 평가 소프트웨어 구현 기술을 설명하고 4장에서는 결론을 맺는다.

*Corresponding Author (ijchung@kangwon.ac.kr)

Received: Oct. 12, 2021, Revised: Nov. 18, 2021, Accepted: Nov. 26, 2021
Kangwon National University (Prof.)

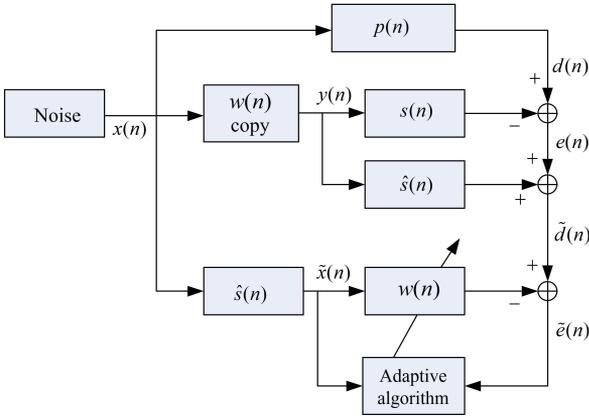


그림 1. feedforward 방식의 MFxLMS 알고리즘의 블록도
Fig. 1. The block diagram of the MFxLMS algorithm for feedforward configuration

II. 능동 소음 제어 알고리즘 및 단일칩 능동 소음 제어기 하드웨어

1. 능동 소음 제어 알고리즘 및 단일칩 능동 소음 제어기 하드웨어

능동 소음 제어 알고리즘으로는 FxLMS 알고리즘이 널리 사용된다. 본 연구에서도 FxLMS 알고리즘 기반으로 개선된 알고리즘을 사용하고 있다 [8]. 본 논문의 구현은 [8]에서 소개된 정규화된 다채널 FxLMS 알고리즘을 기반으로 한다. [8]에 의하면 정규화된 다채널 FxLMS 알고리즘은 비정상 환경에서 우수한 성능을 보이며, 다중 정현파 및 엔진 소리와 같은 실제 잡음에서도 잘 동작한다. 특히, 채널 간 간섭을 최소화 할 수 있다. 능동 소음 제어기는 구조에 따라서 feedforward, feedback, hybrid 방식으로 나뉘어진다. 본 연구에서는 응용에 따라서 적합한 방식을 용이하게 적용해 볼 수 있도록 위의 세 가지 방식을 모두 구현하여 하드웨어 탑재하였다. 탑재된 각 방식의 알고리즘은 다음과 같다.

1.1 Feedforward 방식

FxLMS 알고리즘에서 2차 경로 (secondary path) 지연으로 인한 성능 저하를 개선하기 위해 지연을 보상하는 Modified FxLMS (MFxLMS)가 제안되었다 [9]. 그림 1은 MFxLMS 알고리즘의 블록도이다. 여기서 $p(n)$ 은 잡음의 주경로, $s(n)$ 은 2차 경로를 나타낸다. 그리고 $\hat{s}(n)$ 은 2차 경로의 추정치이다. FxLMS에 비하여 연산량이 증가되기는 하지만 안정성과 수렴 속도의 향상으로 인하여 최근 주목을 받는 알고리즘이다. 본 연구에서도 feedforward 방식에서는 Modified FxLMS 알고리즘을 적용하였다. 또한 다채널 능동 소음 제어가 가능하도록 다채널로 확장하여 구현하였다. 다음은 다채널 MFxLMS 알고리즘이다 [8].

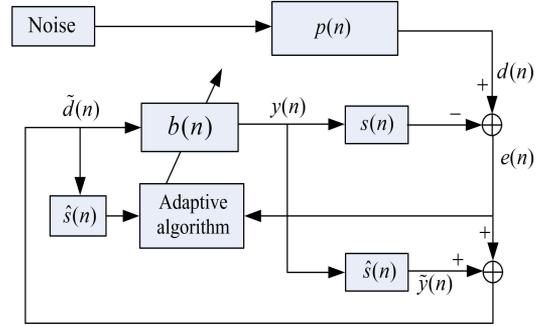


그림 2. feedback 방식의 FxLMS 알고리즘의 블록도
Fig. 2. The block diagram of the FxLMS algorithm for feedback configuration

$$y_k(n) = \sum_{j=1}^J \mathbf{w}_{k,j}^T(n) \mathbf{x}_j(n), \quad k = 1, 2, \dots, K, \quad (1a)$$

$$\mathbf{w}_{k,j}(n) = [w_{k,j,0}(n) \ w_{k,j,1}(n) \ \dots \ w_{k,j,L-1}(n)]^T, \quad (1b)$$

$$\mathbf{x}_j(n) = [x_j(n) \ x_j(n-1) \ \dots \ x_j(n-L+1)]^T, \quad (1c)$$

$$\tilde{\mathbf{x}}_{m,k,j}(n) = \hat{s}_{m,k} \ast \mathbf{x}_j(n), \quad j = 1, 2, \dots, J, \quad k = 1, 2, \dots, K, \quad m = 1, 2, \dots, M, \quad (2)$$

$$\tilde{d}_m(n) = e_m(n) + \sum_{k=1}^K \hat{s}_{m,k}(n) \ast y_k(n), \quad m = 1, 2, \dots, M, \quad (3)$$

$$\tilde{e}_m(n) = \tilde{d}_m(n) + \sum_{k=1}^K \sum_{j=1}^J \mathbf{w}_{k,j}^T(n) \tilde{\mathbf{x}}_{m,k,j}(n), \quad m = 1, 2, \dots, M, \quad (4)$$

$$\mathbf{w}_{k,j}(n+1) = \gamma \mathbf{w}_{k,j}(n) + \mu \sum_{m=1}^M \left(\frac{\tilde{\mathbf{x}}_{m,k,j}(n) \tilde{e}_m(n)}{\left(\sum_{k=1}^K \sum_{j=1}^J \|\tilde{\mathbf{x}}_{m,k,j}(n)\| + \epsilon \right)} \right), \quad k = 1, 2, \dots, K, \quad j = 1, 2, \dots, J, \quad (5)$$

여기서 J 는 참조 마이크의 개수, K 는 제어 스피커의 개수, M 는 오차 마이크의 개수이다. $x_j(n)$ 는 j 번째 참조 마이크를 통해 들어온 입력이고 $\mathbf{w}_{k,j}(n)$ 는 j 번째 참조 마이크 입력에 기반하여 k 번째 제어 스피커 신호를 생성하기 위한 적응 필터 계수이다. $\hat{s}_{m,k}(n)$ 는 k 번째 스피커와 m 번째 오차 마이크 전달 함수의 추정치이다. $e_m(n)$ 은 m 번째 오차 마이크로 들어온 입력이다. 식 (5)에서는 비정상 신호에 대하여 안정성을 개선하기 위해 입력 신호로 정규화하는 알고리즘을 적용하였다. μ 는 convergence factor, γ 는 leaky factor 그리고 ϵ 는 regularization factor이다.

1.2 Feedback 방식

그림 2는 feedback 방식의 FxLMS 알고리즘의 블록도를 보여준다. feedback 방식에선 오차 신호와 적응 필터 출력을 이용하여 잡음 신호에 해당하는 입력 신호를 합성한다. 오차 신호에 들어 있는 정보만을 이용하여 잡음 제거하기 때문에 이 경우는 일종의 예측기와 같은 형태로 동작한다. 따라서 feedforward 방식과 달리 시간적으로 상관도가 높은 신호 또는 주기적인 신호의 잡음만을 제거할 수 있다. 적용된 feedback 방식의 다채널 FxLMS 알고리즘은 다음과 같다.

$$y_k(n) = \sum_{j=1}^M \mathbf{b}_{k,j}^T(n) \mathbf{d}_j(n), \quad k=1,2,\dots,K, \quad (6a)$$

$$\mathbf{b}_{k,j}(n) = [b_{k,j,0}(n) \ b_{k,j,1}(n) \ \dots \ b_{k,j,L-1}(n)]^T, \quad (6b)$$

$$\mathbf{d}_j(n) = [\tilde{d}_j(n) \ \tilde{d}_j(n-1) \ \dots \ \tilde{d}_j(n-L+1)]^T, \quad (6c)$$

$$\tilde{\mathbf{x}}_{m,k,j}(n) = \hat{s}_{m,k} * \mathbf{d}_j(n), \quad j=1,2,\dots,M, \quad k=1,2,\dots,K, \quad m=1,2,\dots,M, \quad (7)$$

$$\tilde{d}_j(n) = e_j(n) + \sum_{k=1}^K \hat{s}(n) * y_k(n), \quad j=1,2,\dots,M, \quad (8)$$

$$\mathbf{b}_{k,j}(n+1) = \gamma \mathbf{b}_{k,j}(n) + \mu \sum_{m=1}^M \left(\frac{\tilde{\mathbf{x}}_{m,k,j}(n) e_m(n)}{\sum_{k=1}^K \sum_{j=1}^J \|\tilde{\mathbf{x}}_{m,k,j}(n)\| + \epsilon} \right), \quad k=1,2,\dots,K, \quad j=1,2,\dots,M, \quad (9)$$

여기서 $\mathbf{b}_{k,j}(n)$ 는 j 번째 오차 마이크를 기반으로 합성된 입력에 기반하여 k 번째 제어 스피커 신호를 생성하기 위한 적응 필터 계수이다.

1.3 Hybrid 방식

Hybrid 방식은 feedforward 방식과 feedback 방식을 결합해서 사용하는 방식이다. 그림 3의 feedforward 부분은 MFxLMS, feedback 부분은 FxLMS를 적용한 hybrid 방식의 블록도이다. 그림에서 볼 수 있듯이 제어 신호 $y(n)$ 이 feedforward 부분의 적응 필터 출력과 feedback 방식의 적응 필터 출력의 합임을 알 수 있다. 따라서 hybrid 방식의 알고리즘은 feedforward 부분의 수식과 feedback 부분의 수식에서 다음 부분만 수정된다. 다만, hybrid 방식의 연산량 증가로 인하여 hybrid 방식의 경우는 단일 채널만 가능하도록 구현하였다. 따라서 $J=1, K=1, M=1$ 인 경우이므로 인덱스는 생략하였다.

$$y(n) = \mathbf{w}^T(n) \mathbf{x}(n) + \mathbf{b}^T(n) \mathbf{d}(n). \quad (10)$$

2. 단일칩 능동 소음 제어 하드웨어

능동 소음 제어 시스템을 저가의 단일칩으로 구현하기 위해서는 MCU의 선택이 중요하다. 능동 소음 제어 구현에 필요한 주변 장치들을 최대한 내장하고 있어야 하며, 다채널 FxLMS 알고리즘의 연산량을 감당할 수 있어야 한다. 본 연구에서는 위의 조건을 만족하는 저가의 MCU인 Texas Instruments사의 TMS320F280049를 선정하여 구현하였다. TMS320F280049의 특징은 다음과 같다 [10].

- 단일 싸이클 MAC 연산을 지원하는 CPU
- CLA 보조 프로세서
- 50KB의 내장 SRAM
- 256KB의 플래쉬 메모리
- 3개의 12bit A/D 변환기
- 2개의 12bit D/A 변환기
- 7개의 PGA (OP 앰프)

TMS320F280049는 저가의 MCU임에도 불구하고 과거 DSP로 분류되었던 C2000 DSP를 기반으로 하고 있기 때문에 디지털 신호 처리에 특화되어 있어서 고속 연산이 가능

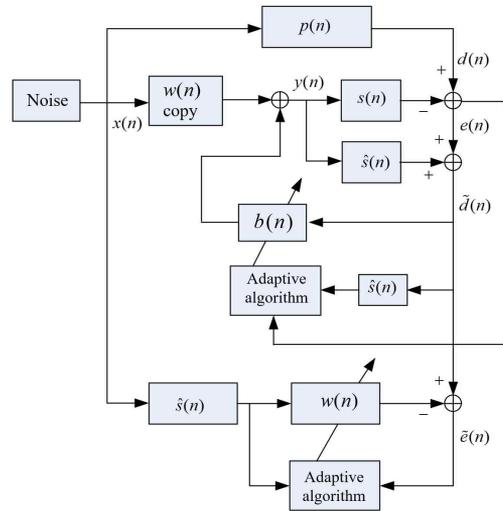


그림 3. hybrid 방식의 MFxLMS 및 FxLMS 알고리즘의 블록도
Fig. 3. The block diagram of the MFxLMS and the FxLMS algorithms for hybrid configuration

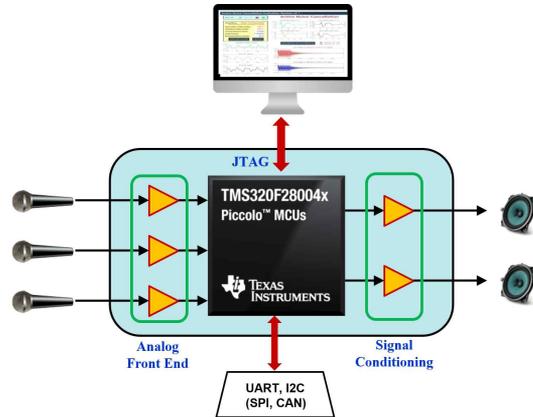


그림 4. 능동 소음 제어 평가 시스템의 블록도
Fig. 4. The block diagram of the active noise control evaluation system

하다. 또한 위에서 나열한 바와 같이 능동 소음 제어를 구현에 필수적인 주변장치인 A/D 변환기와 D/A 변환기 및 아날로그 신호 조절을 위한 OP 앰프를 다수 내장하고 있어서 다채널 능동 소음 제어를 구현하는데 적합하다. 그림 4는 다음 절에서 설명할 GUI 기반의 평가 소프트웨어를 포함한 전체 능동 소음 평가 시스템의 블록도이다. 3개의 A/D 변환기와 2개의 D/A 변환기를 이용할 수 있으므로 그림과 같이 다채널 능동 소음 제어가 가능하다.

그림 5는 본 연구를 통해 제작된 능동 소음 제어 하드웨어 평가 보드이다. 그림에 나타난 바와 같이 마이크 입력단의 anti-aliasing 및 스피커 출력단의 anti-imaging을 위한 저역 필터 구현에 필요한 OP 앰프 외에는 추가의 부품 없이 TMS320F280049 단일칩 만으로 구현되었다. (3.3V를 직접 인가한다면, 5V-to-3.3V 레귤레이터도 필요 없다.) 7개의 PGA가 내장되어 있기는 하지만, 입력 3개, 출력 2개를 위한

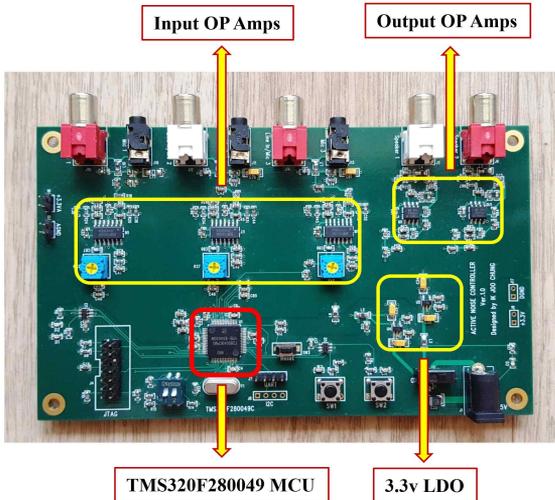


그림 5. TMS320F280049를 이용한 능동 소음 제어 평가 보드
 Fig. 5. The active noise control evaluation board using the TMS320F280049.

저역 필터를 구현하기에는 충분치 않기 때문에 추가의 OP 앰프가 필요하다. 저역 필터는 4차 체비셰프로 설계하였으며, 입력단은 Multiple-feedback 방식으로, 출력단은 Sallen-key 방식으로 구현하였다. 구현된 저역 필터의 cutoff 주파수는 1.5 kHz이다.

III. 알고리즘 포팅 및 GUI 평가 소프트웨어

1. 알고리즘 포팅

II 절에서 소개한 능동 소음 제어 알고리즘을 기반으로 개발된 하드웨어에 다음 알고리즘들이 포팅되었다. (아래에서 J x K x M은 J개의 참조 마이크, K개의 제어 스피커, M개의 오차 마이크를 의미한다.)

- Feedforward MFxLMS 1 x 1 x 1
- Feedforward MFxLMS 1 x 2 x 2
- Hybrid MFxLMS, FxLMS 1 x 1 x 1
- Feedback FxLMS 0(1) x 1 x 1
- Feedback FxLMS 0(2) x 2 x 2

Hybrid 방식의 경우에는 feedforward 부분은 MFxLMS, feedback 부분은 FxLMS로 구현되었다. feedback 방식에서는 참조 마이크를 사용하지 않는다. 내부적으로 참조 신호를 합성하여 사용하는데, 괄호는 합성된 참조 신호의 개수를 의미한다. 한편, Feedforward 방식과 Hybrid 방식의 경우에는 제어 스피커의 소리가 참조 마이크로 역수신되어 능동 소음 제어 기능을 떨어뜨리게 되는데 이를 방지하기 위해서는 역으로 수신되는 신호를 상쇄시키기 위한 feedback neutralization 기능이 필요하다. 1 x 1 x 1 구조의 경우는 feedback neutralization 필터를 지원한다. (1 x 2 x 2의 경우는 추가 연산의 여력이 없어서 지원하지 않는다. feedback

방식의 경우는 참조 마이크가 없기 때문에 feedback neutralization 기능이 필요하지 않다.)

알고리즘을 포팅하는 과정에서 수행한 코드 최적화는 [7]에서 제안한 방법을 동일하게 적용하였다. 다만, 아래와 같은 이유로 보조 프로세서인 CLA를 이용한 병렬 처리는 수행하지 않았다. [7]에서는 CLA를 이용하여 병렬 처리를 함으로써 CPU 연산의 약 20% 정도에 해당하는 추가 연산 능력을 얻을 수 있었다. 한편, CLA는 C 컴파일러를 사용하기에는 적합하지 않기 때문에 어셈블리 프로그램을 필요로 한다. 본 연구에서는 [7]과 달리 식 (5)와 식 (9)에 leaky factor와 regularization factor를 포함하고 있고, 이로 인해 CPU에서 처리해야할 연산량이 증가하였기 때문에 CLA의 연산 비중이 10% 내외로 줄어든다. 반면, 이번 연구에서는 [7]과 달리 여러 알고리즘들 구현하고 있기 때문에 CLA 어셈블리 프로그래밍을 포함한 병렬 처리 개발 부담이 크다. 또한 상용화 시에 지속적으로 개선 및 변경 등을 수행하게 되는데, 이 과정에서 코드 유지 관리 및 검증이 쉽지 않다. 결론적으로 개발 및 유지 관리 비용을 고려할 때 CLA를 이용하여 추가적으로 얻는 연산 능력의 이점이 상대적으로 크지 않다고 할 수 있다.

표 1은 코드 최적화를 적용하여 구현된 각 알고리즘이 처리할 수 있는 적응 필터의 최고 차수를 보여준다. 표에서 2차 경로 함수의 차수는 적응 필터의 최고 차수를 고려하여 먼저 적절한 값으로 지정한 후에 실시간 연산이 가능한 적응 필터의 최고 차수를 구하는 방식으로 적응 필터의 최고 차수를 결정하였다. hybrid 방식의 경우에는 feedback 적응 필터의 차수 역시 적절히 선택한 후, feedforward 적응 필터의 최고 차수를 구하였다. 표에서 굵은체가 실험을 통해 얻은 최고차를 보여준다. 최적화 과정에서 loop unrolling

표 1. 각 알고리즘의 적응 필터 최고 차수
 Table 1. The maximum order of the adaptive filter in each algorithm.

Algorithm	Filter	Maximum order
Feedforward 1x2x2	adaptive filter	340
	secondary path filter	170
Feedforward 1x1x1	adaptive filter	1205
	secondary path filter	400
Hybrid 1x1x1	feedforward adaptive filter	655
	feedback adaptive filter	500
	secondary path filter	300
Feedback 0x2x2	adaptive filter	205
	secondary path filter	100
Feedback 0x1x1	adaptive filter	1395
	secondary path filter	400

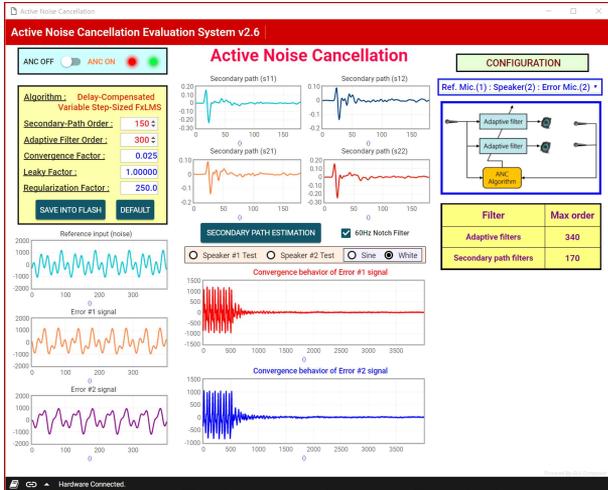


그림 6. GUI 평가 응용 프로그램 (Feedforward 1 x 2 x 2)
Fig. 6. GUI evaluation software (Feedforward 1 x 2 x 2)

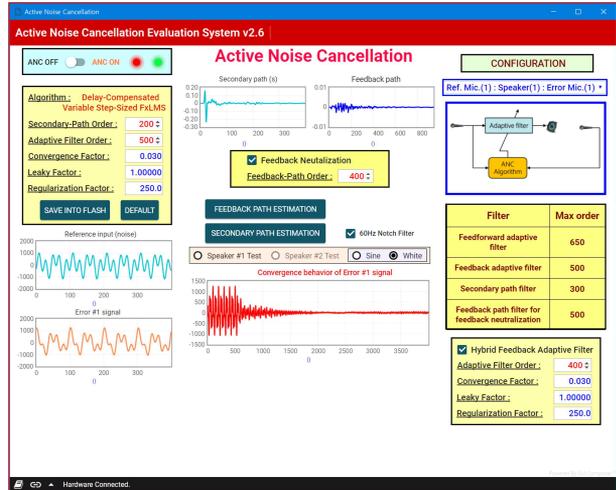


그림 7. GUI 평가 응용 프로그램 (Hybrid 1 x 1 x 1)
Fig. 7. GUI evaluation software (Hybrid 1 x 1 x 1)

이 적용되었기 때문에 적응 필터의 차수는 5의 배수로 얻어진다. 표를 보면 feedback 0 x 2 x 2 방식의 연산량이 제일 많은 것을 알 수 있다. 이 경우는 실질적으로 2 x 2 x 2에 해당하기 때문에 연산량이 제일 많다.

2. GUI 평가 소프트웨어

서론에서도 언급한 바와 같이 능동 소음 제어 기술의 실시간 구현이 용이하지 않기 때문에 능동 소음 제어와 관련된 많은 연구들이 room impulse response modeling을 통한 컴퓨터 모의 실험에 의존해 왔다. 그런데 이런 음향 모델이 단순히 FIR 필터로 모델링되기 때문에 모의 실험 모델이 실제 물리적 현상을 완벽하게 설명하지 못한다. 또한 능동 제어 시스템에 적용되는 적응 알고리즘의 수렴성 증명이 수학적 복잡성을 완화하기 위한 조건 하에서 이루어지기 때문에 실제 여러 조건에서 수렴하는 지는 별개의 문제이다. 이와는 별도로 알고리즘에 따라서는 컴퓨터 연산 정밀도에서 기인하는 수치적 불안정성에 민감할 수 있기 때문에 장시간의 실시간 테스트를 통해 실질적인 안정성을 확인해야 한다. 따라서, 실시간 구현을 통한 알고리즘의 검증이 반드시 이루어져야 한다. 한가지 문제점은 하드웨어를 이용하여 알고리즘을 검증하기 위한 실험들이 컴퓨터 모의 실험의 경우보다 상당히 번거롭다는 것이다. 여러 파라미터들, 예를 들어 convergence factor, leaky factor, regularization factor, 필터 차수들의 여러 조합에 대해 임베디드 형태의 하드웨어에서 실험을 하게 되는데, 이럴 경우 파라미터가 바뀔 때 마다, 컴파일을 새로 하고 플래쉬에 저장하는 형태로 실행해야 한다. 또한, 실험 중의 결과를 확인하기가 쉽지 않다. 컴퓨터 모의 실험에서는 여러 단계에서의 신호 파형을 도시하는 형태로 동작 상황을 쉽게 모니터링 할 수 있지만, 실시간으로 동작하는 하드웨어에서는 이런 신호 상태들을 확인하는 것이 쉽지 않다. 개발된 GUI 평가 소프트웨어는 하드웨어 상에서도 실시간으로 여러 파라미터들을 설정하고, 실시간 동작

시의 신호들을 모니터링할 수 있도록 함으로써, 개발된 하드웨어 상에서 포팅된 알고리즘을 용이하게 실험 및 평가를 할 수 있도록 한다.

개발된 GUI 소프트웨어는 JTAG 포트를 통해 개발된 하드웨어와 통신을 한다. MCU 실행 코드를 생성하는 과정에서 심볼 테이블을 보존하고 GUI 소프트웨어는 이 심볼 테이블을 통해 MCU의 변수를 인지하고 원하는 변수나 배열 정보를 읽어 도시한다.

그림 6은 feedforward 1 x 2 x 2 방식에 해당하는 GUI 소프트웨어 실행 모습이다. GUI 소프트웨어를 통하여 다음과 같은 기능을 수행할 수 있다.

- 능동 소음 제어 기능의 ON/OFF
- feedforward, feedback, hybrid 등의 알고리즘 선택
- 2차 경로 및 feedback neutralization을 위한 경로 추정 및 추정 계수 도시
- 적응 필터 차수, 2차 경로 차수, convergence factor, leaky factor, regularization factor 설정 및 초기화
- 스피커 출력 확인을 위한 백색 잡음 및 정현파 테스트 출력
- 마이크 전원 노이즈 제거를 위한 60Hz notch 필터 ON/OFF
- 참조 마이크, 오차 마이크 입력 파형의 오실로스코프 형태의 실시간 파형 도시
- 적응 필터 개시 시에 수렴 과정을 보기 위한 convergence behavior 도시

그림 6의 좌상단에서 능동 소음 제어기의 여러 파라미터들을 설정할 수 있다. 좌하단의 3개의 연속된 그래프는 참조 마이크, 오차 마이크 #1, 오차 마이크 #2에 수신된 신호를 차례로 보여준다. 중상단의 4개의 그래프는 추정된 2차 경로의 room impulse response를 보여준다. 중하단의 연속된 두 개의 그래프는 능동 소음 제어가 개시된 후, 수렴

하는 과정을 캡처하여 보여준다. 이상단은 현재 설정된 알고리즘 (feedforward $1 \times 2 \times 2$)을 보여주며 또한 원하는 알고리즘을 선택할 수 있도록 한다.

그림 7은 hybrid $1 \times 1 \times 1$ 방식에 해당하는 GUI의 모습이다. 좌상단 부분은 feedforward 부분의 파라미터를 설정하는 부분이고 우하단은 feedback 부분의 파라미터들을 설정하는 부분이다. 오차 마이크가 하나만 이용되기 때문에 오차 마이크 #2의 그래프는 생략된다. 중앙 부분은 feedback neutralization 기능을 ON/OFF 시키고 feedback neutralization의 필터 차수를 설정하는 부분이다.

GUI 소프트웨어에서 보듯이 GUI 소프트웨어는 임베디드 하드웨어에서 여건 상 제공하기 어려운 사용자 인터페이스를 확장하는 기능을 제공한다. 실제로 그림 5의 하드웨어에는 능동 소음 제어 기능을 ON/OFF 하는 버튼과 2차 경로는 추정을 실행하는 버튼, 총 2개의 사용자 버튼만이 있다.

IV. 결론

본 연구에서는 단일칩을 이용한 실시간 능동 소음 제어 평가 보드 및 GUI 평가 소프트웨어를 개발하였다. 연구에서 사용한 TMS320C280049 MCU는 능동 소음 제어기 구현에 필요한 대부분의 주변장치를 내장하고 있기 때문에 신호 조절을 위한 OP 앰프 이외에는 추가적인 부품 없이 단일칩으로 구현이 가능하였다. 개발된 하드웨어에는 feedforward, feedback, hybrid 방식의 단일 채널 및 다채널 알고리즘을 포팅하였다. 포팅 과정에서 연산 최적화를 수행하여 실시간으로 수행할 수 있는 적응 필터의 차수를 최대화 하였다.

한편, 개발된 평가 보드를 용이하게 실험할 수 있는 GUI 평가 소프트웨어도 함께 개발하였다. 이 소프트웨어를 이용하여 능동 소음 제어기의 여러 파라미터들을 실시간으로 조정하여 주어진 소음 환경에서 최적의 파라미터를 선택할 수 있다. 또한, 능동 소음 제어기가 동작할 때, 실시간으로 여러 신호들을 실시간으로 모니터링할 수 있어서 능동 소음 제어기의 수렴성 및 성능을 용이하게 확인할 수 있다.

References

[1] S. M. Kuo, D. R. Morgan, "Active Noise Control : A Tutorial Review," Proc. IEEE Vol. 87, No. 6, pp. 943-973, 1999

[2] P. Leug, "Process of Silencing Sound Oscillations," U.S. Patent 2043416, 1936

[3] C. H. Hansen, "Current and Future Industrial Applications of Active Noise Control," Noise Control Engineering Journal, Vol. 53, No. 5, pp. 181-196, 2005

[4] S. M. Kuo, S. Mitra, Woon-Seng Gan, "Active Noise Control System for Headphone Applications," IEEE Transactions on Control Systems Technology, Vol. 14, No. 2, pp. 331-335, 2006

[5] J. C. Burgess, "Active Adaptive Sound Control in a duct: A Computer Simulation," J. Acoust. Soc. Amer., Vol. 70, No. 3, pp. 715 - 726, 1981

[6] P. Samarasinghe, W. Zhang, T. Abhayapala, "Recent Advances in Active Noise Control Inside Automobile Cabins: Toward Quieter cars," IEEE Signal Processing Magazine, Vol. 33, No. 6, pp. 61-73, 2016

[7] I. J. Chung, "The Efficient Implementation of the Multi-channel Active Noise Controller Using a Low-cost Microcontroller Unit," J. Acoust. Soc. Kr., Vol. 38, No. 1, 2019 (in Korean)

[8] I. J. Chung, "Multi-channel Normalized FxLMS Algorithm for Active Noise Control," J. Acoust. Soc. Kr., Vol. 35, No. 4, pp. 280-287, 2016 (in Korean)

[9] M. Rupp and A. H. Sayed, "Modified FxLMS Algorithms with Improved Convergence Performance," Proc. IEEE ASILOMAR-29, no. 2, 1255-1259, 1995

[10] Texas Instruments, TMS320F28004x Piccolo Microcontrollers Technical Reference Manual, 2015.

Ikjoo Chung (정익주)



1986 Electronics Engineering from Seoul National University, Seoul, Republic of Korea (B.S.)

1988 Electronics Engineering from Seoul National University, Seoul, Republic of Korea (M.S.)

1992 Electronics Engineering from Seoul National University, Seoul, Republic of Korea (Ph.D.)

1992~Electronics Engineering, in Kangwon National University (Prof.)

Field of Interests: Adaptive Signal Processing, Speech Signal Processing, Active Noise & Vibration Control

Email: ijchung@kangwon.ac.kr