한국음향학회지 제28권 제3호 pp. 229~238 (2009)

범용 디지털 신호처리기를 이용한 국악기 사운드 엔진 개발

Sound Engine for Korean Traditional Instruments Using General Purpose Digital Signal Processor

강 명 수*, 조 상 진**, 권 순 덕***, 정 의 필*
(Myeongsu Kang*, Sangjin Cho**, Sundeok Kwon***, Uipil Chong*)

*울산대학교 컴퓨터정보통신공학부, **울산대학교 전기전자정보시스템공학부, ***경남정보대학 산학비즈니스본부
(접수일자: 2009년 2월 3일; 채택일자: 2009년 3월 16일)

본 논문에서는 TMS320F2812 신호처리기를 이용하여 가야금과 태평소의 사운드 엔진을 구현하였다. Commuted Waveguide Synthesis (CWS) 기반의 가야금과 태평소 모델을 신호처리기에 탑재하고 악기 선택 버튼을 두어 해당 악기의 사운드 샘플을 매 연성 시간마다 합성하도록 하였다. 합성음은 SPI 통신을 이용하여 DAC로 전송되며 오디오 인터페이스를 거쳐 스피커를 통해 재생된다. 합성 모델의 지연 라인은 합성음의 피치를 조절하는데, 이 지연라인의 길이를 결정하기 위해 GPIO를 이용하여 한 샘플을 합성하는데 필요한 시간을 측정하였다. 가야금은 28.6 μs, 태평소는 21 μs가 소요되었다. 태평소와 가야금의 동시 발음수를 고려하였을 때 태평소는 동시 발음수 1을 가지므로 21 μs, 가야금은 일반적으로 동시 발음수가 2이므로 57.2 μs의 연산 시간이 필요하다. 이는 실시간 연주가 충분히 가능한 시간이다. 제안한 사운드 엔진의 경우, 인터럽트 서비스 루틴에서 각 사운드 샘플의 합성과 DAC로의 전송이 일어난다. 인터럽트 서비스 루틴은 시스템의 안정성을 보장하기 위해 타이머의 주기 매칭 이벤트를 이용하여 60 μs마다 주기적으로 호출된다. 이와 같이 합성된 음을 녹음하여 원음과 스펙트럼으로 비교한 결과, 가야금은 원음과 매우 유사한 음을 합성할 수 있었고, 태평소는 '무(無), 황(黃), 태(太), 중(仲)' 음을 제외한 나머지 음에 대해서 태평소의 음색을 잘 표현하는 음을 합성 할 수 있었다.
핵심용어: 사운드 엔진, Commuted Waveguide Synthesis, 국악기, 가야금, 태평소
투고분야: 음악음향 및 음향심리 분야 (8.6)

This paper describes a sound engine of Korean traditional instruments, which are the Gayageum and Taepyeongso, by using a TMS320F2812. The Gayageum and Taepyeongso models based on commuted waveguide synthesis (CWS) are required to synthesize each sound. There is an instrument selection button to choose one of instruments in the proposed sound engine, and thus a corresponding sound is produced by the relative model at every certain time. Every synthesized sound sample is transmitted to a DAC (TLV5638) using SPI communication, and it is played through a speaker via an audio interface. The length of the delay line determines a fundamental frequency of a desired sound. In order to determine the length of the delay line, it is needed that the time for synthesizing a sound sample should be checked by using a GPIO. It takes 28.6 μs for the Gayageum and 21 μs for the Taepyeongso, respectively. It happens that each sound sample is synthesized and transferred to the DAC in an interrupt service routine (ISR) of the proposed sound engine. A timer of the TMS320F2812 has four events for generating interrupts. In this paper, the interrupt is happened by using the period matching event of it, and the ISR is called whenever the interrupt happens, 60 μs. Compared to original sounds with their spectra, the results are good enough to represent timbres of instruments except 'Mu, Hwang, Tae, Joong' of the Taepyeongso. Moreover, only one sound is produced when playing the Taepyeongso and it takes 21 μs for the real-time playing. In the case of the Gayageum, players usually use their two fingers (thumb and middle finger or thumb and index finger), so it takes 57.2 μs for the real-time playing.
Keywords: Sound Engine, Commuted Waveguide Synthesis, Korean traditional instruments, Gayageum, Taepyeongso
ASK subject classification: Musical Acoustics and Psychoacoustics (8.6)

I. 서 론

디지털 신호처리 기술의 발달과 함께 여러 가지 악기 음 합성 방식들이 개발되었다. 대표적인 악기 음 합성 방법으로는 가산합성 (additive synthesis)방식, 감산합성 (subtractive synthesis)방식, 주파수변조 (frequency modulation, FM) 합성방식, 샘플링 (sampling) 방식 등이 있다. 샘플링 방식은 가장 자연스러운 사운드를 합성할 수 있는 반면 사운드 엔진을 구현하기 위해서는 많은

책임저자: 정 의 필 (upchong@ulsan.ac.kr)
680-749 울산광역시 남구 무거동 산 29 울산대학교 컴퓨터정보통신공학부
(전화: 052-259-2220; 팩스: 052-259-1687)

양의 메모리가 요구되며, FM 방식은 인공적인 사운드를 만드는데 유용하지만 악기음과 어울리지 않는 부자연스러운 음을 만들어 내기도 한다. Julius Orion Smith III가 제안한 Commuted Waveguide Synthesis (CWS)는 이러한 단점을 극복할 수 있는 합성법이다. 본 논문에서는 CWS기반의 가야금과 태평소 모델을 범용 신호처리기로 구현한 사운드 엔진을 제안한다. 2005년과 2006년에는 각각 TMS320C6713 DSK와 Intel Bulverde PXA272A를 이용한 가야금의 사운드 엔진을 구현한 시도가 있었고 [1,2], 비슷한 시기에 Erdem Motuk외 2명이 물리적 모델링을 이용한 악기 음 합성을 FPGA로 구현하였다 [3]. 본 논문에서는 기존의 연구에서 사용한 프로세서보다 다소 성능은 떨어지나 저렴한 범용 디지털 신호처리기를 이용하여 가야금과 태평소의 사운드 엔진을 구현하였다. 이러한 사운드 엔진의 개발은 향후 새로운 인터페이스와 결합하여 기존의 틀에서 벗어난 연주를 할 수 있는 뉴직 인터페이스 개발에 기여할 수 있을 것이다. 본 논문의 구성은 다음과 같다. II장에서는 대상악기의 소개와 사운드 엔진의 구현에 대해서, 사운드 엔진의 구동환경과 합성한 음의 결과는 III장에서 다루었다. IV장에서는 제안한 사운드 엔진의 문제점과 해결 방안에 대해 토의 하였고, 결론 및 향후 과제는 V장에 기술하였다.

II. 사운드 엔진 구현

2.1. 대상 악기

본 논문에서는 한국의 대표 전통악기인 가야금과 태평소를 대상으로 사운드 엔진을 구현하였다. 가야금은 풍류 가야금 (법금)과 산조 가야금 두 종류가 있으며 크기와 모양에 있어 차이가 있다. 본 논문에서는 산조 가야금을 대상으로 가야금의 사운드 엔진을 구현하였다. 산조 가야금은 그림 1에서와 같이 열두 줄의 명주 현과 안족, 현침 등으로 구성되며, 12개의 안족이 각각의 현을 지지하고 있고, 이를 좌우로 움직이면서 조율한다 [4]. 태평소는 고려 왕조 때 중국에서부터 소개되어 우리나라에서 연주된 한국 전통 관악기로, 호적, 쇄납, 날라리 등으로 불리며 국악기 중 특히 음이 높고 음량이 큰 악기이다 [5]. 그림 2에서 볼 수 있듯이 태평소는 동팔랑, 원추 모양의 관대, 서, 지공 등으로 구성된다.

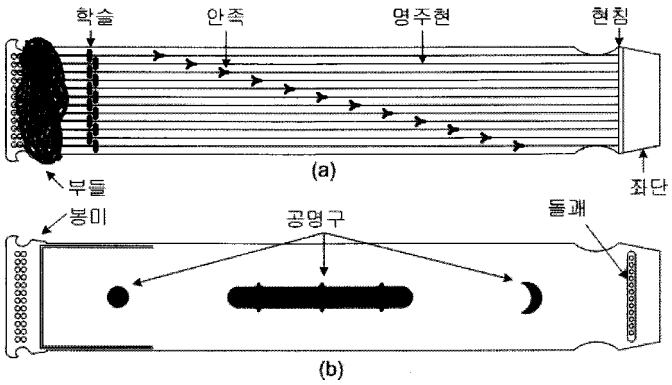


그림 1. 산조 가야금의 구조. (a)윗면, (b)아랫면
Fig. 1. Structure of the Sanjo Gayageum, (a)top and (b)bottom view.

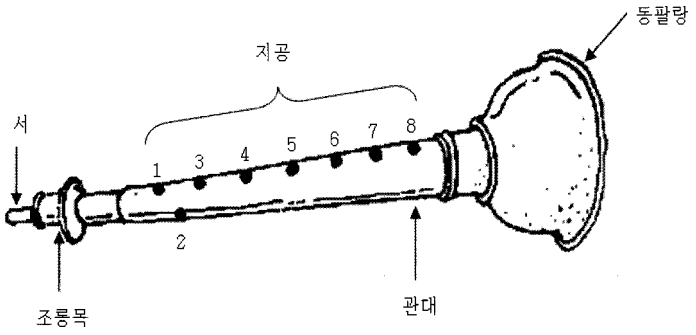


그림 2. 태평소의 구조
Fig. 2. Structure of the Taepyeongso.

관대에는 앞에 7개 뒤에 1개 총 8개의 지공이 있는데, 이 지공의 개폐 유무에 의해 음정이 결정된다. 표 1은 지공의 개폐에 따른 태평소의 율명을 나타낸 것이다. 테스트 음의 경우, 태평소의 기본음은 아니지만 음향학적 분석을 위해 사용하였다 [6].

2.2. Commuted Waveguide Synthesis를 이용한 합성 모델

가야금 사운드는 CWS를 이용해 합성하였으며 ,아래 그림 3은 CWS를 이용한 가야금 모델이다.

L은 지연 라인의 길이로 합성음의 기본 주파수를 결정하며 식 (1)과 같이 나타난다.

$$L = \frac{f_s}{f_o} \tag{1}$$

디지털 영역에서는 지연 라인의 길이는 정수 값을 가져야 하나 실제 지연 라인의 길이는 실수 값을 가진다. 즉, $L = L_i + L_f$와 같이 표현할 수 있으며 L_i는 정수부, L_f는 미소지연부이다. L_i는 가야금 모델에서의 지연에 해당하고, L_f 는 1보다 작은 양의 실수 값을 가지며 그림 3에서 미소지연 필터 (fractional delay filter) $F(z)$로 표현 되었다. 미소지연 필터는 FIR 또는 전대역 통과 필터 (all-pass filter)를 사용하는데, 본 논문에서는 FIR 필터인 라그랑즈 보간기 (Lagrange interpolator)를 이용하였다. 라그랑즈 보간기의 필터 계수 $h(n)$은 식 (2)와 같다 [7].

$$h(n) = \prod_{\substack{k=0 \\ k \neq n}}^{N} \frac{D-k}{n-k}, \quad n = 0, 1, 2, 3, \ldots, N \tag{2}$$

여기서 N은 FIR 필터의 차수이고, D는 원하는 지연이다.

지연 라인의 길이와 미소지연 필터를 이용하면 원하는 기본 주파수를 갖는 출력을 얻을 수 있으나 실제 자연계에 존재하는 주파수 의존적인 파동의 감쇄 현상은 표현할 수 없다. 이를 해결하기 위해 식 (3)과 같은 1차 all-pole 저역 통과 필터 (low-pass filter)를 사용하였다 [8].

$$H_l(z) = g\frac{1+a_1}{1+a_1 z^{-1}} \tag{3}$$

여기서 g는 0 Hz에서의 필터의 이득으로 $|g| < 1$의 값을, a_1은 차단 주파수를 결정하는 필터 계수로서 시스템의 안정성을 위해 $-1 < a_1 < 0$의 조건을 갖도록 하였다.

태평소 역시 CWS를 기반으로 하여 사운드를 합성하였

표 1. 지공의 개폐에 따른 태평소의 율명
Table 1. Notes on the Taepyeongso by which tone-holes are covered.

율명 \ 지공	1	2	3	4	5	6	7	8
테스트	●	●	●	●	●	●	●	●
중(仲)	●	●	●	●	●	●	●	○
임(林)	●	●	●	●	●	●	○	○
남(南)	●	●	●	●	●	○	○	○
무(無)	●	●	●	●	○	○	○	○
황(黃)	●	●	●	○	○	○	○	○
태(太)	●	●	○	○	○	○	○	○
중(仲)	○	●	○	○	○	○	○	○

● : 닫힘 ○ : 열림

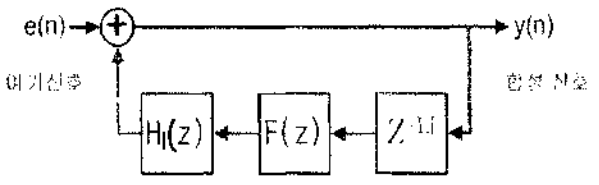


그림 3. CWS 기반의 가야금 모델. $H_l(z)$: 루프필터, $F(z)$: 미소 지연 필터, Z^{-L_i} : 지연 라인
Fig. 3. Gayageum model based on CWS. $H_l(z)$: loop filter, $F(z)$: fractional delay filter, Z^{-L_i} : delay line.

으며 그림 4는 본 논문에서 사용한 태평소 모델이다.

태평소 관대 내부를 이동하는 파동은 서와 동팔랑의 접합부분에서 임피던스의 차이에 의한 반사 (reflection)가 발생한다. 따라서 이를 표현하기 위해 관의 양 끝단에 반사 필터를 추가하였다 [9]. 반사 필터는 원추형관 (conical bore)의 특성 임피던스와 원통의 특성 임피던스의 합과 차의 비로 표현된다. 이는 연속 시간에서의 특성이므로 이산 시간에서의 특성으로 변환한 후 식 (5)의 주파수 응답과 비교하여 제곱 오차의 가중 합 (weighted sum of the squared error)이 최소가 되는 계수를 계산하여 반사 필터를 설계하였다.

$$R_L(z) = \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1}} \tag{5}$$

여기서, $b_0 = 0.0188, b_1 = 0.0188, a_1 = -0.8364$이고, MATLAB의 invfreqz함수로 구한 값이다 [6].

관대 내부에서는 공기의 점성이나 온도에 의해 자체적으로 감쇄가 발생하는데, 본 논문에서는 공기의 점성에 의한 감쇄만을 고려하였다. 이를 위하여 파수 (number of wave)를 정의하고 한 샘플에 대한 감쇄를 계산하고 지연라인의 길이만큼 제곱하면 관대 내부의 공기 점성에 의한 감쇄특성을 얻을 수 있다 [10]. 반사 필터 설계와 동일한 방법으로 식 (6)과 같은 감쇄 필터를 설계하였다.

$$B(z) = \frac{b_0}{1 + a_1 z^{-1}} \tag{6}$$

여기서, $b_0 = 0.8417, a_1 = -0.1343$이다 [6].

2.3. 범용 디지털 신호처리기

사운드 엔진을 위한 디지털 신호처리기로 TI (Texas Instrument)사의 TMS320F2812를 이용하였다. TMS320F2812는 150 MHz로 동작하고 고정 소수점 (fixed point) 연산을 지원하며, 12bit의 분해능을 가진 ADC (Analog-to-Digital Converter) 16채널이 내장되어 있다. 또한, 외부 주변 장치들과의 통신을 위해 SPI (Serial Peripheral Interface), eCAN (Enhanced Controller Area Network), SCI (Serial Communications Interace), McBSP (Multichannel Buffered Serial Port) 모듈을 제공한다 [11]. 그림 5는 제안한 사운드 엔진의 전체 구성도이다.

제안한 시스템에는 버튼을 이용하여 악기를 선택할 수

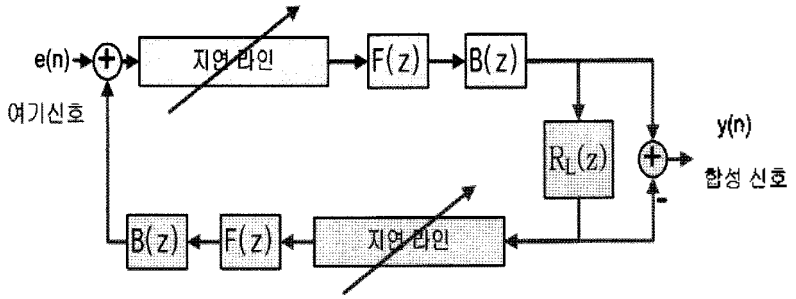


그림 4. CWS 기반의 태평소 모델. $B(z)$: 감쇄 필터, $F(z)$: 미소 지연 필터, $R_L(z)$: 반사 필터

Fig. 4. Taepyeongso model based on CWS. $B(z)$: damping filter, $F(z)$: fractional delay filter, $R_L(z)$: reflection filter.

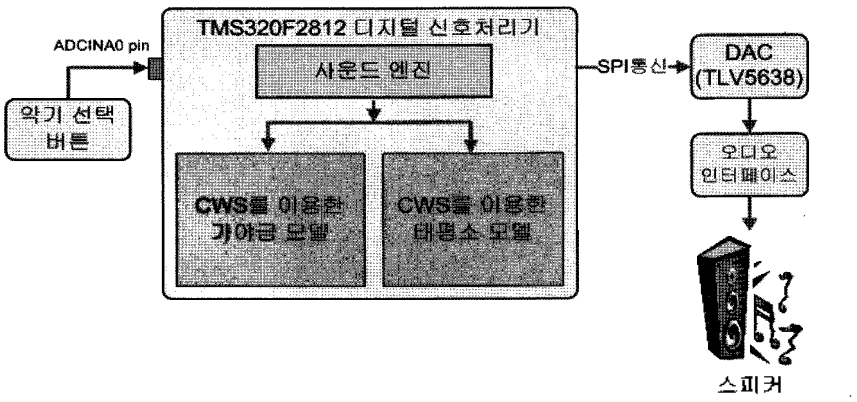


그림 5. 제안한 사운드 엔진의 구성도

Fig. 5. Overall system of the proposed sound engine.

있도록 하였다. 악기를 선택하기 위한 방법으로는 두 가지를 고려해 볼 수 있는데 GPIO (General Purpose Input Output)와 ADC 변환을 이용하는 방법이다. 본 논문에서는 ADC 변환을 이용하여 악기가 선택하도록 하였다. 이는 향후 가야금과 태평소의 사운드 엔진만이 아닌 추가적인 악기의 선택을 고려해서이다. GPIO의 경우 엔코더 (encoder)를 사용하더라도 logic high/low의 상태만을 확인 할 수 있어 두 가지 악기의 선택만이 가능할 뿐이다. 악기의 선택은 디지털 신호처리기에서 발생되는 3.3V의 입력 전압을 이용하여 버튼의 상태에 따른 출력 전압을 ADC 변환을 통해 이루어진다. 3.3V의 출력이 발생할 때는 ADC 변환 값이 4095, 0V의 출력이 발생할 때는 ADC 변환 값이 0이므로 4095와 0사이의 임계값으로 악기를 선택할 수 있게 된다.

가야금과 태평소 소리를 합성하기 위한 알고리듬은 신호처리기에 탑재되어 있으며 합성된 샘플은 SPI 통신을 이용하여 DAC (Digital-to-Analog Converter)로 전송되고 이는 오디오인터페이스를 거쳐 스피커를 통해 재생된다. 그림 6은 디지털 신호처리기 상에서 제안한 사운드 엔진의 동작 흐름도로서 크게 3단계로 나뉜다.

단계 1. 시스템의 초기화

신호처리기에 전원을 인가하면 가장 먼저 시스템의 초기화 작업이 진행된다. 이 단계에서는 시스템의 전체 구동 속도를 설정하고 이를 분주하여 주변 회로에 공급되는 클럭 (clock)을 결정하는 등의 작업이 이루어진다.

단계 2. 주변 회로의 초기화 및 설정

시스템 초기화 작업이 끝나면 사용되는 주변 회로의 초기화 및 설정 작업이 이루어진다. 제안한 사운드 엔진을 구현하기 위해서는 인터럽트 벡터 테이블 (interrupt vector table), ADC 그리고 SPI에 대한 초기화 및 설정 작업이 필요하다. 그리고 인터럽트 서비스 루틴에서 사운드 샘플이 합성되는 시간을 측정하기 위한 GPIO의 초기화 및 설정 작업이 필요한데, 샘플의 합성 시간은 합성음에 대한 CWS 모델의 지연 라인 길이와 관계있기 때문이다. 예를 들어, 샘플링 주파수 44100 Hz와 22050 Hz에 대해서 기본 주파수가 150 Hz인 사운드 샘플을 합성하고자 한다면 지연 라인은 $\frac{44100(\fallingdotseq 1/22.67\mu s)}{150} - 294$과 $\frac{22050(\fallingdotseq 1/45.35\mu s)}{150} - 147$의 길이를 가져야한다. 이 경우 하나의 샘플을 합성하는데 걸리는 시간이 40 μs이하로 설정되어야 한다면 지연 라인의 길이를 294로 하여 합성하여야 할 것이다.

제안한 사운드 엔진에서 일정 시간마다 하나의 사운드 샘플을 합성하고 이를 DAC로 전송하기 위해 인터럽트 서비스 루틴을 호출 하는 방법을 이용하였다. 인터럽트 벡터 테이블의 초기화 및 타이머의 주기 매칭 (period

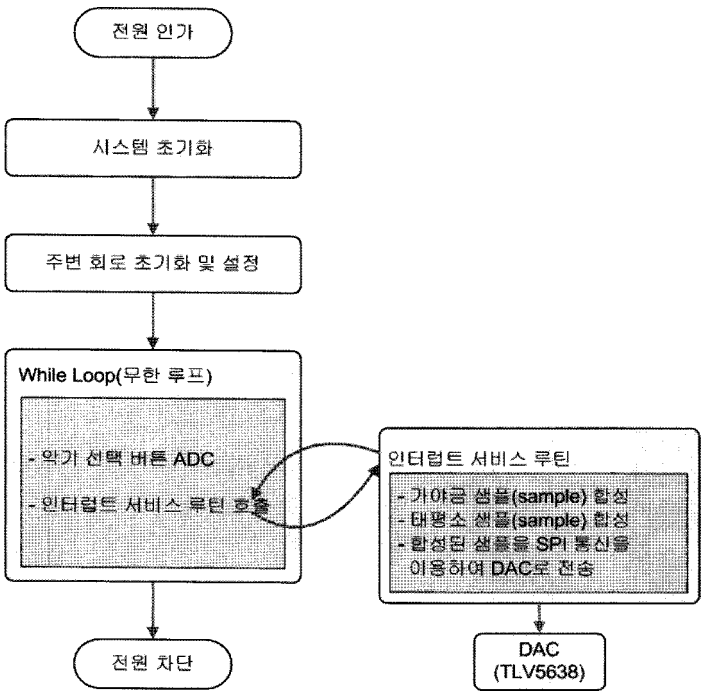


그림 6. 신호처리기 상에서 제안한 사운드 엔진의 동작 원리
Fig. 6. Operating scheme of the proposed sound engine on TMS320F2812.

matching) 이벤트를 이용해 인터럽트를 발생시키고 인터럽트가 발생하는 것을 주기로 인터럽트 서비스 루틴이 호출 되도록 처리하였다.

버튼의 상태에 따른 출력 전압에 대한 ADC 변환을 이용하여 악기를 선택할 수 있도록 하기위해 ADC 모듈의 초기화 작업과 사용자 설정이 필요하다. Spectrum Digital사에서 제공하는 라이브러리를 이용해 ADC 모듈의 초기화 작업을 수행하였고, ADC 모듈의 구동 클럭 결정 → ADC 변환 채널 수 결정 → ADC 변환 순서 결정 → ADC 변환 샘플링 모드 설정 → ADC 변환 모드 설정 과정을 거쳐 사용자 설정을 하였다. 본 논문에서 ADC 구동 속도는 25 MHz로 사용하였으며, 악기 선택 버튼 1개만 ADC 변환이 필요하므로 1개의 ADC 채널만 사용하였다. 또한, 버튼의 상태를 계속해서 확인할 필요가 있기 때문에 연속 변환 모드로 ADC를 설정하여 사용하였다.

인터럽트 서비스 루틴에서 합성된 음은 SPI 통신을 이용해 DAC로 전송된다. SPI는 신호처리기를 마스터로, DAC를 슬레이브로 설정하여 마스터에서의 사운드 출력 결과가 DAC로 전송 되도록 설정하였다. 사용된 DAC는 TLV5638로 두 개의 12비트의 출력 전압을 위한 출력 단자가 있으며 상위 4비트는 제어 비트로 하위 12비트는 데이터 비트로 사용한다 [12].

단계 3. While Loop (무한루프)

시스템과 주변 회로의 초기화 및 설정이 이루어지면 while 루프가 전원이 차단될 때까지 무한히 반복된다. while 루프 내에서는 버튼의 ADC 변환과 가야금과 태평소 사운드 합성을 위한 파라미터 초기화 작업이 이루어지며, 일정 시간마다 인터럽트 서비스 루틴의 호출이 일어난다.

인터럽트 서비스 루틴에서는 선택된 악기에 대한 사운드 샘플의 합성과 SPI를 이용한 샘플 전송을 수행한다. 인터럽트의 발생주기는 타이머 주기 레지스터의 값에 따라 달라지는데 가야금의 동시 발음수에 맞춰 사운드 샘플은 60 μs 마다 합성 되도록 타이머 레지스터의 값을 설정하였다.

III. 구현 결과

본 논문에서는 범용 신호처리기를 이용한 사운드 엔진을 구현하였고, 그림 7은 제안한 사운드 엔진의 구동 환경을 보여준다. CCS3.1 (Code Composer Studio 3.1)과 JTAG 에뮬레이터를 이용하여 가야금과 태평소의 합성 알고리듬을 TMS320F2812 범용 신호처리기에 다운로드 하고 악기 선택 버튼을 이용하여 대상악기를 선택한다. 해당 악기의 사운드 샘플은 신호처리기 내에서 합성되며, 합성된 샘플은 오디오 인터페이스를 통해 스피커로 출력된다. 신호처리기의 GPIO를 출력으로 설정하여 합성 알고리듬이 실행되는 순간 3.3V, 종료 되는 순간 0V를 출력하면 오실로스코프를 이용하여 해당 알고리듬의 실행 시간을 측정할 수 있다.

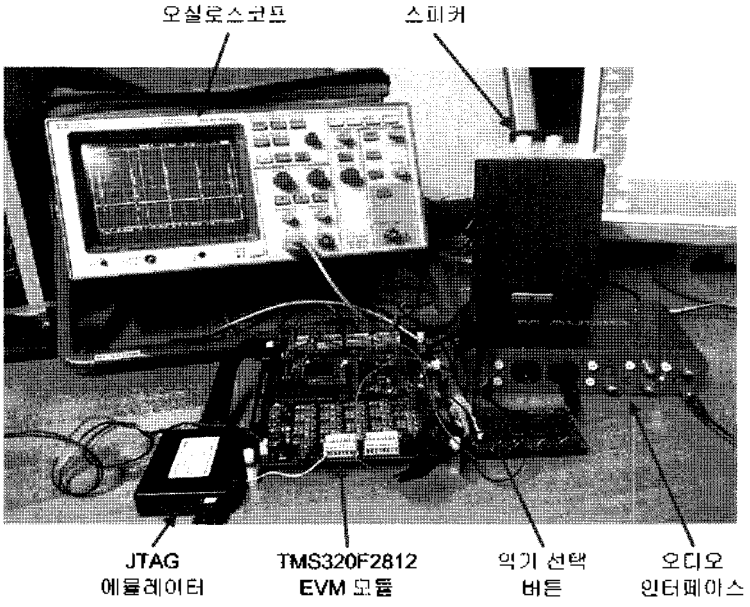


그림 7. 제안한 사운드 엔진을 위한 구동 환경
Fig. 7. Environment for driving the proposed sound engine.

3.1. Commuted Waveguide Synthesis를 이용한 가야금과 태평소 음 합성 결과

가야금의 몸통 특성을 여기 신호로 합성한 소리는 가야금의 음색을 잘 표현하는데 그림 8과 같이 원음과 합성음(제안한 사운드 엔진 시스템의 출력 결과를 녹음한 음)의 스펙트럼을 비교함으로써 확인 할 수 있다.

그림 8에서 볼 수 있듯이 원음과 합성음에는 약간의 차이가 발생한다. 이는 실험 환경에서 기인한 것으로 그림 9(a)의 가야금 11번 현에 대한 노이즈 프로파일을 통해 확인할 수 있다. 11번 현의 성분는 제 8배음 (3500 Hz 부근)까지가 사운드의 음색을 표현하는데 직접적으로 영향을 미치며 그 이후의 배음에 대해서는 노이즈의 크기 (-40 dB)

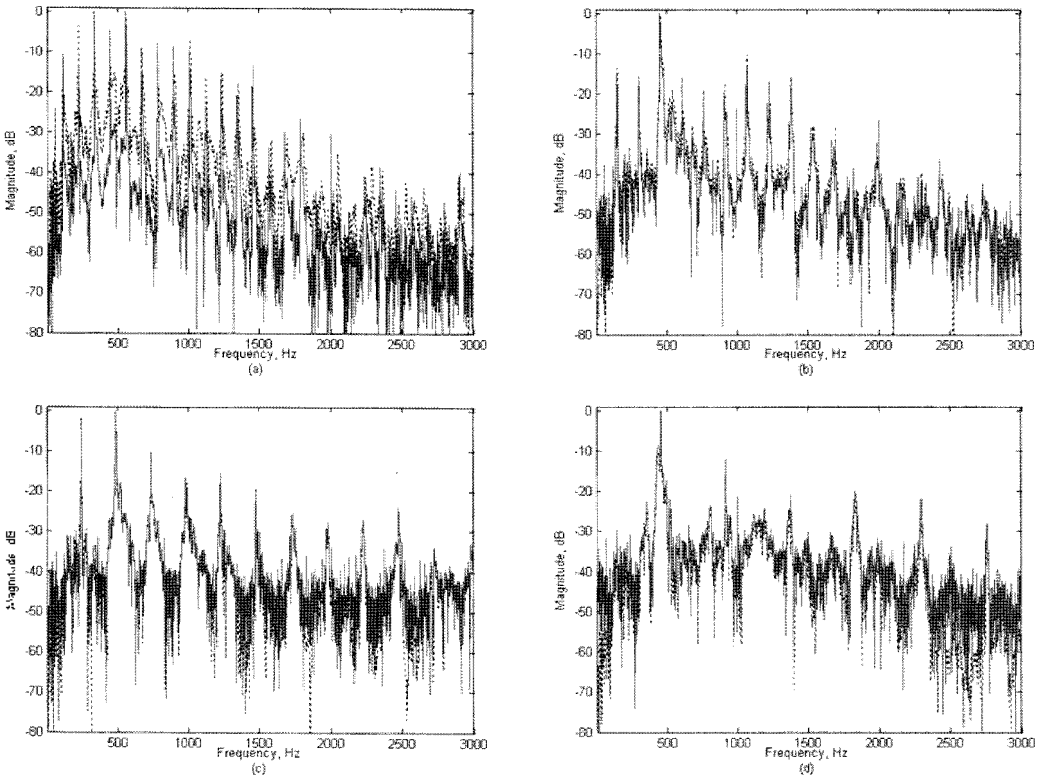


그림 8. 가야금 원음 (점선)과 합성음(실선)의 스펙트럼 결과 비교. (a) 2번 현, (b) 5번 현, (c) 8번 현, (d) 11번 현
Fig. 8. Comparison of the original Gayageum sounds (dotted line) and the synthesized Gayageum sounds (solid line). (a) 2nd string, (b) 5th string, (c) 8th string, (d)11th string.

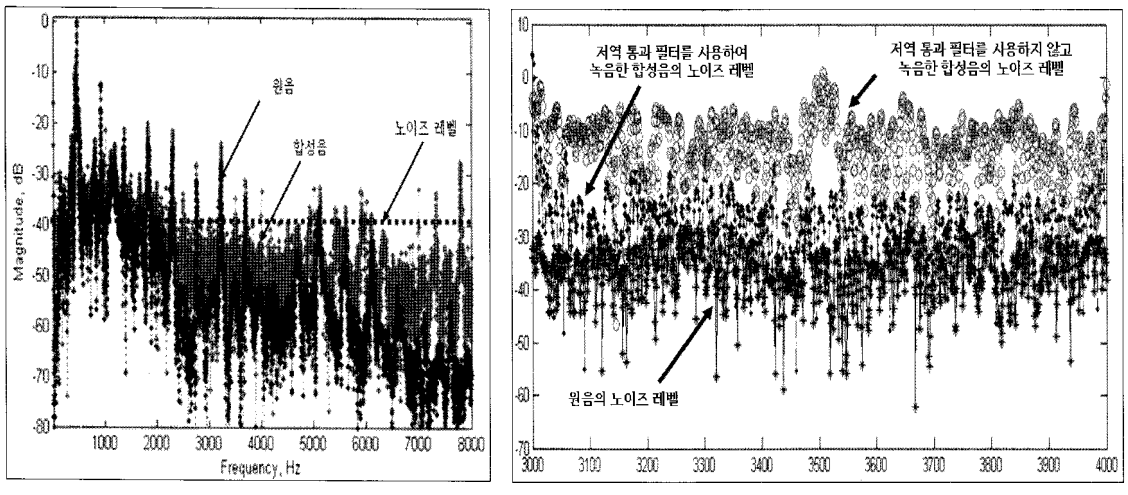


그림 9. (a) 가야금 11번 현에 대한 노이즈 프로파일과 (b) 저역 통과 필터 사용여부에 의한 출력의 노이즈 레벨 비교
Fig. 9. (a) Noise profile of the 11th Gayageum string and (b) Comparison of the recorded synthesized sound by low-pass filter.

가 배음의 크기보다 대체로 크게 나타나 사운드의 음색을 표현하는데 거의 영향을 미치지 못한다. 즉, 노이즈가 합성음의 고주파 대역에 영향을 미치기 때문에 가야금 원음과는 다른 날카로운 사운드가 발생한다. 이를 해결하기 위해 출력단에 2차 Butterworth 저역 통과 필터 (차단주파수, $f_c = 3500Hz$, 샘플링주파수, $fs = \frac{1}{43.35 \times 10^{-6}} Hz$)를 추가한 결과 기존의 출력 보다는 날카로운 사운드가 많이 사라졌음을 확인할 수 있었다. 그림 9(b)는 가야금 11번 현의 출력 단에 저역 통과 필터 추가 여부에 대한

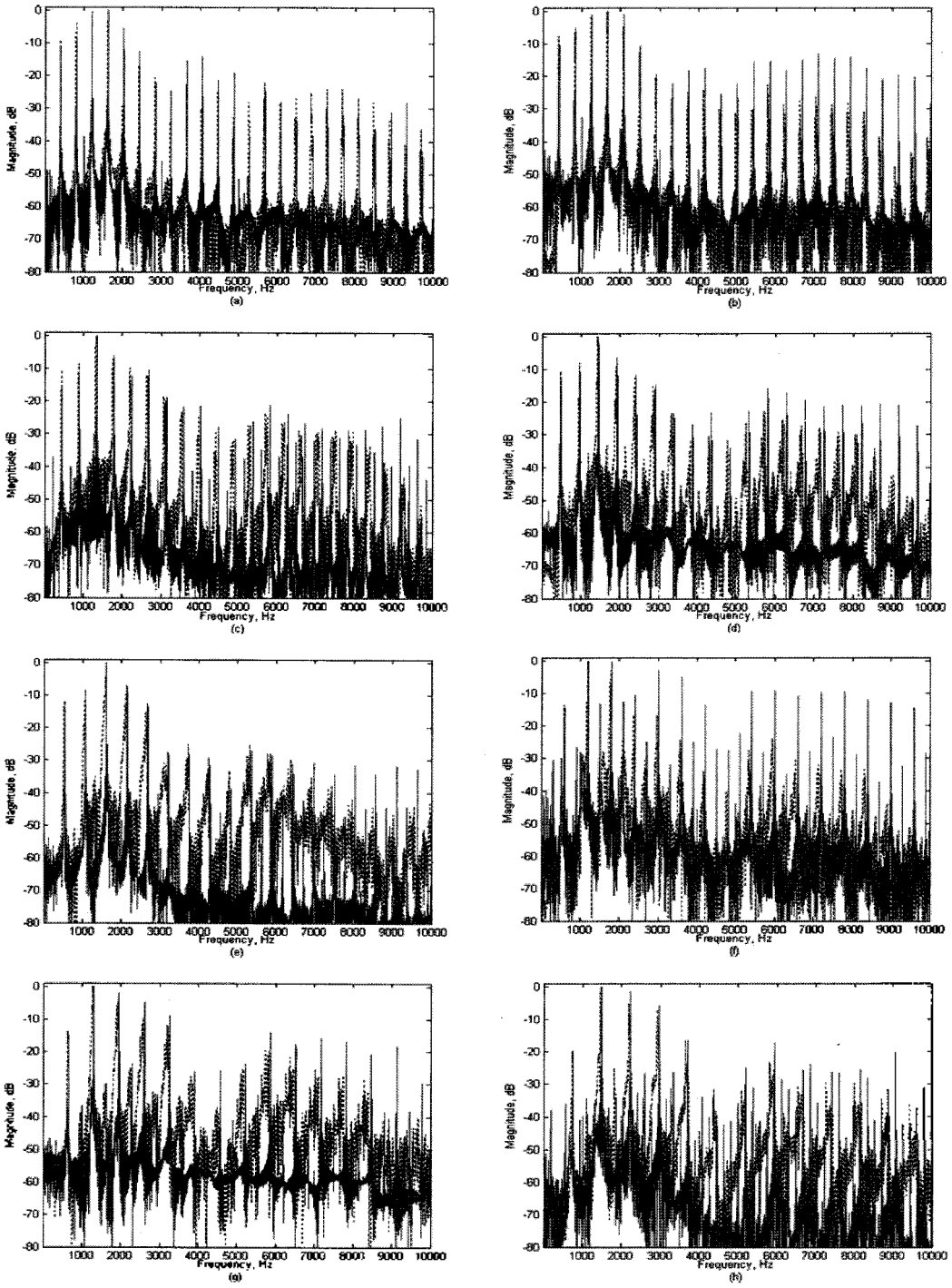


그림 10. 태평소 원음 (점선)과 합성음 (실선)의 스펙트럼 결과 비교. (a) 테스트, (b) 중 (仲), (c) 임 (林), (d) 남 (倆), (e) 무 (無), (f) 황 (黃), (g) 태 (太), (h) 중 (仲)

Fig. 10. Comparison of the original Taepyeongso sounds (dotted line) and the synthesized Taepyeongso sounds (solid line). (a) Test, (b) Joong (仲), (c) Im (林), (d) Nam (倆), (e) Mu (無), (f) Hwang (黃), (g) Tae (太), (h) Joong (仲).

노이즈 프로파일을 보여준다. 높은 차수의 저역 통과 필터를 사용할수록 더 나은 결과가 나타나겠지만, 이는 알고리듬의 수행 시간에 영향을 미친다. 2차 Butterworth 저역 통과 필터의 연산 시간은 15.2 μs 의 짧지 않은 시간이 걸리며 더 높은 차수의 필터를 사용하게 되면 그로 인해 지연 라인의 길이가 줄어들어야 하는 결과가 발생하여 출력 결과의 음질이 떨어진다. 따라서 범용 신호처리기의 출력단에 아날로그 저역 통과 필터의 사용이 사운드 엔진을 구현함에 있어 효율적이라 판단된다.

태평소는 관대의 특성을 여기 신호로 사용하여 합성하였고, 그림 10은 태평소의 원음과 합성음의 스펙트럼 비교 결과이다.

'테스트, 중(侏), 임(琳), 남(倹)' 음의 합성음은 원음과 유사하지만 나머지 '무(無), 황(黃), 태(太), 중(仲)' 음의 합성음에 대해서는 원음과 많은 차이가 있다. 이는 태평소 모델 자체의 문제라고 판단된다. 만약 사운드 엔진의 문제라면 모든 음에 대해서 같은 문제가 발생하겠지만 그림 10에서 보듯이 '테스트, 중(侏), 임(琳), 남(倹)' 음에 대해서는 문제가 발생하지 않는다. 따라서 원인은 '무(無), 황(黃), 태(太), 중(仲)' 음에 대한 여기 신호의 문제라고 할 수 있다. 또한, '무(無), 황(黃), 태(太), 중(仲)' 음의 여기 신호 분석 결과 여기 신호 자체에 불필요한 배음 성분이 포함되어 있었으며 이와 같은 배음 성분들이 해당 음을 합성하는데 있어 불필요하게 추가되었기 때문에 원음과 차이가 발생하게 되었다.

3.2. 합성음 데모

아래의 웹 페이지에서 가야금과 태평소 사운드의 데모를 확인 할 수 있다. 첫 번째는 원음, 두 번째는 매틀랩(matlab)을 이용한 합성음, 세 번째는 제안한 사운드 엔진에서 합성한 음을 사운드 포지 (sound forge)로 녹음한 것이다.

http://signal.ulsan.ac.kr/elekto/documents/ASK_sound_engine.html

IV. 토 의

제안한 사운드 엔진은 가야금과 태평소 사운드를 실시간으로 합성할 수 있다. 여기에 다양한 형태의 인터페이스가 추가된다면 가야금과 태평소의 현대화에 기여할 수 있을 것이다. 또한, 이를 신개념의 국악기로 발전시킨다면 실제 공연에 활용될 수 있을 뿐만 아니라 국악의 세계화에도 기여할 수 있을 것이다. 하지만 이러한 사운드 엔진을 발전시키기 위해서는 해결해야할 문제점이 있어 이를 극복할 방안에 대해 토의하고자 한다.

가야금 합성음의 경우 원음과 스펙트럼을 비교해 보았을 때 매우 유사함을 확인할 수 있었지만 태평소의 경우는 '테스트, 중(侏), 임(琳), 남(倹)' 음을 제외한 나머지 음들에 대해서는 원음과 많은 차이가 발생하였다. 이는 '무(無), 황(黃), 태(太), 중(仲)' 음에 대한 잘못된 여기 신호의 사용에서 나타나는 결과이다. 잘못된 여기 신호의 사용으로 원하지 않는 배음들이 나타났으며 이로 인해 원음과 다른 음색을 갖게 되었다. 이는 새로운 태평소 모델을 사운드 엔진의 알고리듬으로 사용하거나 올바른 여기 신호를 이 모델에 적용하면 해결 할 수 있을 것이다. 그럼에도 불구하고, 이와 같은 태평소 모델을 제안한 사운드 엔진에 사용한 이유는 가야금 모델에서 약간의 수정으로 태평소의 사운드 엔진을 구현할 수 있었기 때문이다.

물리적 모델링을 이용한 사운드 합성은 많은 연산 양과 메모리를 요구하므로 일반적으로 유동 소수점 (floating-point) 연산을 지원하는 TMS320C6713이나 FPGA와 같은 신호처리기를 이용한다. 하지만, 본 논문에서는 고정 소수점 (fixed-point) 연산을 지원하고 상대적으로 작은 메모리를 가진 TMS320F2812를 이용하였는데, 이는 기존보다 저사양의 신호처리기에 물리적 모델링 알고리듬을 탑재한 사운드 엔진이 사용 가능한지를 확인하고자 함이었고, 가야금과 태평소의 사운드 엔진 모두 사용 가능하다는 결론을 내릴 수 있었다.

VI. 결론 및 향후 과제

본 논문에서는 CWS기반의 현악기와 태평소 모델을 이용하여 사운드 엔진을 구현하였다. 태평소의 경우 한 샘플이 만들어져 재생되는 시간이 약 21 μs가 소요되었고 동시 발음수가 하나이기 때문에 제안한 태평소의 사운드 엔진을 다른 인터페이스와 결합하여 실시간으로 연주하더라도 문제가 되지 않는다. 가야금은 적은 양의 여기 신호를 사용하는 태평소와 달리 많은 양의 여기 신호를 사용한다. 약 900여개의 샘플이 태평소의 사운드 엔진을 위한 여기 신호로 사용된 반면, 가야금의 경우는 몸통의 특성을 잘 표현하기 위해서 약 10,000여개의 샘플이 여기 신호로 사용되었다. 900여 샘플의 여기 신호를 적재하기

위한 신호처리기의 내부 메모리 공간 (18K word)은 충분하지만 10,000여 샘플의 여기 신호는 신호처리기의 내부 메모리에 모두 적재하기에 공간이 충분하지 않다. 결과적으로 본 논문에서는 TMS320F2812 신호처리기의 외부 메모리에 가야금의 여기 신호를 적재하여 사운드 엔진을 구현하였다. 외부 메모리 접근 시간은 내부 메모리 접근 시간의 약 1.5배가 소요되며 (내부 메모리 접근 속도 150 MHz = 6.67 ns, 외부 메모리 접근 속도 100 MHz = 10 ns) 가야금 사운드 샘플이 합성되어 재생되는데 28.6 μs가 소요되었다. 전통적인 가야금의 연주법에는 최대 2개의 줄을 동시에 튕기거나 뜯으므로 동시 발음수는 2이고, 이를 위해서는 인터럽트 서비스 루틴이 60 μs마다 호출되어야 하며 가야금의 사운드 엔진도 가야금의 연주를 묘사할 수 있는 외부 인터페이스와 결합하여 실시간 연주가 가능하다.

향후 태평소의 사운드 엔진에 대한 문제점을 극복하고 이 사운드 엔진에 가야금과 태평소의 연주를 묘사할 수 있는 사용자 편의의 인터페이스를 개발한다면 다소 접근하기 어려운 국악의 대중화 및 일반화에 기여할 수 있을 것이다.

감사의 글

이 논문은 2009년 정부 (교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임 (KRF-2006-521-H00002).

참 고 문 헌

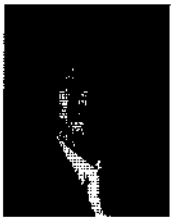
1. Sangjin Cho, Sungmin Choi, Joongbae Pyoun, Hoon Oh, Uipil Chong, "Implementation of Sound Engine of Gayageum Based on Intel Bulverde PX272A Processor," *Proc. AES 29th International Conference*, pp. 72-77, Sep., 2006.
2. 조상진, 오훈, 정의필, "TMS320C6713 DSK를 이용한 가야금 사운드 합성," *대한전자공학회 추계 종합학술대회 논문집*, 28권, 2호, 435-438쪽, 2005.
3. Erdem Moluk, Roger Woods, and Stefan Bilbao, "FPGA-based Hardware for Physical Modelling Sound Synthesis by Finite Difference Schemes," *2005 IEEE International Conference on Field Programmable Technology Conference*, pp. 103-110, Dec., 2005.
4. 조상진, 정의필, "안족이 있는 악기의 개선된 현의 모델 개발," *한국음향학회지*, 26권, 7호, 328-333쪽, 2007.
5. *태평소*, http://ko.wikipedia.org/wiki/%ED%83%9C%ED%8F%89%EC%86%8C, 2009.
6. 변중배, *태평소 음향분석 및 물리적 모델링*, 울산대학교 석사 학위 논문, 2008.
7. T. I. Laakso, V. Valimaki, M. Karjalainen, and U. K. Laine, "Splitting the Unit Delay-Tools for Fractional Delay Filter Design," *IEEE Signal Processing Mag.*, vol. 13, no. 1, pp. 30-60, 1996.
8. V. Valimaki, J. Huopaniemi, M. Karjalainen, and Z. Janosy, "Physical Modeling of Plucked string Instruments with Application to Real-Time Sound Synthesis," *J. Audio Eng. Soc.*, vol. 44, no. 5, pp. 331-353, 1996.
9. Gary P. Sacvone, "Time-Domain Synthesis of Conical Bore Instrument Sounds," *Int. Computer Music Conference*, 2002.
10. A. H. Benade, "On the propagation of sound waves in a cylindrical conduit," *J. Acoustical Soc. Am.*, vol. 44, no. 2, pp. 616-623, 1968.
11. *TMS320F2812*, http://focus.ti.com/lit/ds/symlink/tms320f2812.pdf, 2009.
12. *TLV5638*, http://focus.ti.com/lit/ds/symlink/tlv5638.pdf, 2009.

저자 약력

• 강 명 수 (Myeongsu Kang)
한국음향학회지 제28권 2호 참조

• 조 상 진 (Sangjin Cho)
한국음향학회지 제23권 7호 참조

• 권 순 덕 (Sundeok Kwon)



1999년 : 울산대학교 전자계산학과 졸업 (공학사)
2001년 : 울산대학교 대학원 전자계산학과 졸업 (공학석사)
2003년 : 울산대학교 대학원 전자계산학과 박사수료
현재 : 경남정보대학 산학협력교수
※주관심분야 : 오디오, 신호처리, 데이터베이스

• 정 의 필 (Uipil Chong)
한국음향학회지 제23권 7호 참조