

# 적응 2차 볼테라 필터의 효율적인 구현

## The effective implementation of adaptive second-order Volterra filter

정 익 주\*★

Ik Joo Chung\*★

### Abstract

In this paper, we propose an efficient method for implementing the adaptive second-order Volterra filter. To reduce computational load, the UCFD-SVF has been proposed. The UCFD-SVF, however, shows deteriorated convergence performance. We propose a new method that initializes the adaptive filter weights periodically on the fact that the energy of the filter weights is slowly increased. Furthermore, we propose another method that the interval for the weight initialization is variable to guarantee the performance and we shows the method gives the better performance under the non-stationary environment through the computer simulation for the adaptive system identification.

### 요 약

본 논문에서는 적응 2차 볼테라 필터를 효율적으로 구현할 수 있는 새로운 방법을 제안한다. 연산량 감소를 위해 제안된 UCFD-SVF는 수렴 성능이 저하되는 단점이 있다. UCFD-SVF의 적응 필터 계수가 적응이 진행되면서 그 에너지가 급격하게 증가하지 않는다는 점을 이용하여 적응 필터 계수를 주기적으로 초기화는 방법을 제안하였다. 또한 일정한 수렴 성능을 보장하기 위해 가변적인 간격으로 적응 필터 계수를 초기화하는 방법을 제안하였고, 비정상 환경에서 우수한 수렴 특성을 가짐을 적응 시스템 확인 응용을 위한 컴퓨터 모의 실험을 통해 보였다.

*Key words : Non-linear adaptive filter, Adaptive second-order Volterra filter, Frequency-domain adaptive filter, Unconstrained algorithm, Computational complexity*

### 1. 서론

적응 필터는 디지털 신호처리의 다양한 응용 분야에서 활용되고 있다. 그 대표적인 응용 분야로는 시스템 확인, 에코 및 소음 제거, 채널 등화, 어레이 안테나 등이 있다[1]-[3]. 적응 필터에서 가장 중요한 부분은 적응 알고리즘이다. Widrow가 제안한

LMS(Least Mean Square) 알고리즘은 오랜 시간 검증된 안정성과 구현의 간단함으로 인하여 지금도 가장 널리 사용되는 알고리즘의 하나이다[2]. 그러나 그 이후, LMS 알고리즘의 느린 수렴 속도를 개선하기 위해 많은 알고리즘들이 제안되었다. 입력 신호의 자기상관 행렬의 역행렬을 순환적으로 계산하는 RLS(Recursive Least Square), 입력 신

\* Dept. of Electronics Engineering, Kangwon National University

★ Corresponding author

E-mail : ijchung@kangwon.ac.kr, Tel : +82-33-250-6322

※ Acknowledgment

This study was supported by 2017 Research Grant from Kangwon National University (No.520170072)

Manuscript received Jun. 2, 2020; revised Jun. 19, 2020; accepted Jun. 23, 2020.

This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

호의 상관성을 제거하기 위해 변환 영역에서 적응시키는 변환 영역 알고리즘 등이 대표적인 예이다 [4][5].

한편, 지금까지 실제 응용에 적용되는 적응 필터는 기본적으로 선형 적응 필터이다. 선형 적응 필터는 모델링 하고자 하는 미지의 시스템이 선형 시스템일 때 최적의 성능을 발휘한다. 그러나 만약 미지의 시스템이 비선형 성분을 포함하고 있다면 선형 적응 필터로는 만족스런 성능을 얻을 수 없다. 한 예로 오디오 신호 처리에서는 대부분 스피커 출력을 포함하게 되는데 스피커의 비선형성이 문제가 될 수 있다. 이런 경우의 응용에서는 비선형 필터를 이용하면 상당한 성능 개선을 이룰 수 있다[6].

비선형성은 그 종류도 무수하고, 또한 수학적으로 다루기 어려우며, 연산량도 엄청날 정도로 증가되기 때문에 비선형 적응 필터의 실질적인 응용이 매우 제한되는 것이 사실이다. 그럼에도 불구하고 선형 적응 필터의 성능 한계로 인해 비선형 적응 필터에 대한 연구도 활발히 진행되었다. 가장 대표적인 비선형 적응 필터로는 적응 볼테라(Volterra) 필터가 있다. 볼테라 급수 전개에 기반한 적응 볼테라 필터는 비선형 필터 중, 비교적 수학적으로 다루기 용이하며 적당한 커널 차수를 사용할 경우, 고성능 연산기를 사용한다는 가정 하에 어느 정도 실용화가 가능한 비선형 적응 필터이다[7]-[9].

커널 차수가 증가될 경우 연산량이 비약적으로 증가되기 때문에 대부분의 볼테라 필터 응용은 2차 커널까지 고려하고 있다. 따라서 본 연구에서는 적응 2차 볼테라 필터(Second-order Volterra Filter, SVF)의 효율적인 구현에 대하여 논한다.

본 논문의 구성은 다음과 같다. II장에서는 기존의 적응 2차 볼테라 필터 구현에 대하여 살펴보고, III장에서 적응 2차 볼테라 필터의 효율적인 구현 방법을 제안하고 제안된 방법의 성능을 컴퓨터 모의 실험을 통해 보인다. IV장에서는 결론을 맺는다.

## II. 주파수 영역 적응 2차 볼테라 필터(FD-SVF)

시간 영역에서의 적응 볼테라 필터의 출력  $y_v(n)$ 은 식(1)과 같이 주어진다. 여기서  $x(n)$ 는 입력 신

호,  $n$ 은 시간 인덱스,  $h_r(\cdot)$ 는  $r$ 차 볼테라 커널이다. 대부분의 적응 볼테라 필터는 연산량 문제로 인하여 2차 커널까지만 사용한다. 본 논문에서도  $h_1(i)$ 와  $h_2(i,j)$ 만을 사용하는 적응 2차 볼테라 필터로 한정한다.

$$y_v(n) = \sum_{i=0}^{N-1} h_1(i)x(n-i) + \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} h_2(i,j)x(n-i)x(n-j) + \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} h_3(i,j,k)x(n-i)x(n-j)x(n-k) + \dots \quad (1)$$

한편, 고속 컨볼루션을 이용하여 적응 필터의 연산량을 줄이기 위해 주파수 영역 적응 필터 알고리즘이 제안되었다.  $h_1(i)$ 에 해당하는 선형 필터 부분은 이미 잘 알려진 주파수 영역 적응 필터 알고리즘과 동일할 뿐만 아니라,  $h_2(i,j)$ 에 해당하는 2차 커널의 연산량에 비하면 미미하기 때문에 2차 커널 부분의 주파수 영역 적응 필터 알고리즘에 집중할 것이다. 다음은 기존에 제안된 2차 커널에 해당하는 주파수 영역 적응 필터 알고리즘이다[8].

### 1. 필터 출력 연산

$$y(n) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} h_2(i,j)x(n-i)x(n-j) \quad (2)$$

식(2)는 2차 커널에 해당하는 필터 출력이다. 이 출력의 연산을 주파수 영역에서 수행을 한다.

$$\mathbf{Y}^l(k) = \sum_{p+q=k} \mathbf{H}_2^l(p,q)\mathbf{X}^l(p)\mathbf{X}^l(q) + \sum_{p+q=k+2N} \mathbf{H}_2^l(p,q)\mathbf{X}^l(p)\mathbf{X}^l(q), \quad (3)$$

$$k=0,1,2,\dots,2N-1$$

여기서  $\mathbf{H}_2^l(p,q)$ 는 주파수 영역에서의 2차 커널 필터의 계수이고,  $N$ 은 적응 필터의 차수이다. 즉,  $h_2(i,j)$ 의 차수는  $N \times N$ 이다. 주파수 영역에서 FFT 기반의 고속 컨볼루션을 이용하기 위해서는  $h_1(i)$ 의 차수 역시  $N$ 이어야 한다. 향후 표기법에서  $\mathbf{A}$ 는 벡터나 행렬을 의미하며,  $\mathbf{A}(k)$  또는  $\mathbf{A}(p,q)$ 는 그 벡터나 행렬의  $k$ 번째 또는  $(p,q)$ 번째 엘리먼트를 의미한다.

$$\mathbf{X}^l = \text{FFT}_{2N}\{\mathbf{x}^l\} \quad (4)$$

여기서

$$\mathbf{x}^l = [x(lN-N), \dots, x(lN-1), x(lN), \dots, x(lN+N-1)]$$

은 과거  $N$ 개의 입력을 포함하는  $2N$  차원의  $l$ 번째 블록의 입력 벡터이다.  $l$ 은 블록 인덱스이다.  $\text{FFT}_K\{\}$ 는  $K$ 점 FFT 연산을 의미한다.  $l$ 번째 블록의 필터 출력은 다음과 같다.

$$\begin{aligned} \mathbf{y}^l &= [y(lN), y(lN+1), \dots, y(lN+N-1)] \\ &= \text{UH}_N \langle \text{IFFT}_{2N}\{\mathbf{Y}^l\} \rangle \end{aligned} \quad (5)$$

여기서  $\text{IFFT}_K\{\}$ 는  $K$ 점 역 FFT 연산을 의미한다.  $\text{UH}_N \langle \mathbf{y} \rangle$ 은 벡터  $\mathbf{y}$ 의 후반부  $N$ 개의 엘리먼트를 추출하는 연산을 의미한다.

## 2. 필터 계수 적응 연산

$$\mathbf{e}^l = \underbrace{[0, 0, \dots, 0, e(lN), \dots, e(lN+N-1)]}_N \quad (6)$$

여기서  $e(n) = d(n) - y(n)$ 이고,  $d(n)$ 는 원하는 신호이다.

$$\mathbf{E}_2^l(p, q) = \frac{1}{2N} \mathbf{E}^l((p+q)_{2N}), \quad 0 \leq p, q < 2N \quad (7)$$

여기서  $\mathbf{E}^l = \text{FFT}_{2N}\{\mathbf{e}^l\}$  이고,  $(x)_K$ 는  $x$ 의  $K$  모듈로 연산을 의미한다.

$$\mathbf{G}^l(p, q) = \mathbf{E}_2^l(p, q) \overline{\mathbf{X}}^l(p) \overline{\mathbf{X}}^l(q), \quad 0 \leq p, q < 2N \quad (8)$$

$$\begin{aligned} h_2^{l+1}(m, n) &= 2\text{DIFFT}_{2N}\{\mathbf{H}_2^l(p, q) + \mu \mathbf{G}^l(p, q)\}, \\ &0 \leq m, n < 2N, \quad 0 \leq p, q < 2N \end{aligned} \quad (9)$$

$2\text{DIFFT}_K\{\}$ 는  $K \times K$ 점 역 이차원 FFT 연산을 의미한다.  $\overline{\mathbf{X}}$ 는  $\mathbf{X}$ 의 공액 복소수를 의미한다. 그리고 식(9)의  $h_2^{l+1}(m, n)$ 는 연산 중간 과정에서 얻게 되는 시간 영역 2차 커널 적응 필터 계수이다. 최종적으로 주파수 영역 2차 커널의 적응 연산은 다음과 같다.

$$\begin{aligned} \mathbf{H}_2^{l+1}(p, q) &= 2\text{DFFT}_{2N}\{\text{LHLH}_N \langle h_2^{l+1}(m, n) \rangle\}, \\ &0 \leq m, n < 2N, \quad 0 \leq p, q < 2N \end{aligned} \quad (10)$$

여기서  $2\text{DFFT}_K\{\}$ 는  $K \times K$ 점 이차원 FFT 연산

을 의미하고  $\text{LHLH}_N \langle h_2^{l+1}(m, n) \rangle$ 는 다음과 같다

$$\text{LHLH}_N \langle h_2^{l+1}(m, n) \rangle = \begin{cases} h_2^{l+1}(m, n), & \text{if } 0 \leq m, n < N \\ 0, & \text{otherwise} \end{cases}$$

한편, 위의 연산에서 식(5)의 역 FFT는  $h_1(i)$ 에 해당하는 선형 필터 연산 시에 함께 연산되므로 실제 연산량에서는 제외된다. 또, 식(4)의  $\mathbf{X}^l(k)$ 와 식(7)의  $\mathbf{E}^l(k)$ 는  $h_1(i)$ 를 위한 선형 적응 필터 연산 과정에서 계산된 값을 사용하기 때문에 역시 실제 연산량에서는 제외된다. 기존 주파수 영역 2차 적응 볼테라 필터의 2차 커널의 적응을 위한 연산량은 표 1과 같다. 연산량은 통상의 경우처럼 연산에 소요되는 실수 곱셈의 수로 계산하였다. 표 1에서  $K$ 점 FFT 및  $K \times K$ 점 2D FFT에 소요되는 실수 곱셈은 가장 보편적으로 사용되는  $4 \times \frac{K}{2} \log K$ 와  $4 \times K^2 \log K$ 를 적용하였다. 복소 곱셈은 실수 곱셈의 4배로 계산하였다. 식(9)의 연산량에서  $\mu$ 는 실수이기 때문에 (실수)  $\times$  (복소수) 연산이 이루어진다. 그리고 식(7)의 나누기는  $\mu$ 에 포함해서 고려할 수 있기 때문에 실제 나눗셈은 수행되지 않는다.

Table 1. Number of real multiplications of FD-SVF.

표 1. FD-SVF의 실수 곱셈 연산량

Eq.	Number of real multiplications
Eq.(3)	$4 \times (2N(2N+1) + 2N(2N-1)) = 32N^2$
Eq.(8)	$4 \times 2 \times (2N)^2$
Eq.(9)	$4 \times (2N)^2 \log(2N) + 2 \times (2N)^2$
Eq.(10)	$4 \times (2N)^2 \log(2N)$
Total	$32N^2 \log(2N) + 72N^2$

## III. 적응 볼테라 필터의 효율적 구현

본 장에서는 적응 볼테라 필터의 2차 커널 적응을 위한 효율적인 구현 방법을 제안한다. 2절에서 언급한 구현 방식에서 가장 많은 연산을 차지하는 부분은 역 2차원 FFT와 2차원 FFT를 포함하는 식(9)과 식(10)이다. 식(9)과 식(10)을 살펴보면 식(9)에서 역 2차원 FFT를 이용해 시간 영역으로 변환한 후, 식(10)의  $\text{LHLH}_N \langle h_2^{l+1}(m, n) \rangle$  연산을 통해  $m \geq N$  또는  $n \geq N$ 에 대하여  $h_2^{l+1}(m, n) = 0$ 으로 초기화한 후, 다시 2차원 FFT를 수행한다. 이 과정은 시간 영역에서 구한 적응 필터 계수와 동일한

결과를 주파수 영역에서 얻기 위해 필요한 과정이다. 한편, 선형 적응 필터를 주파수 영역에서 구현할 때, 역 FFT와 FFT를 생략하여 연산량을 줄이는 방법이 제안되었으며 이 방법은 unconstrained 주파수 영역(unconstrained frequency-domain, UCFD) 알고리즘으로 알려져 있다[10]. 이 UCFD 알고리즘을 FD-SVF에 적용하면, 식(9)과 식(10)은 생략되기 때문에 주파수 영역 2차 커널의 적응 연산을 수행하는 수식은 다음과 같이 변경된다.

$$\mathbf{H}_{2,uc}^{l+1}(p,q) = \mathbf{H}_{2,uc}^l(p,q) + \mu \mathbf{G}^l(p,q), \quad (11)$$

여기서  $\mathbf{H}_{2,uc}^l(p,q)$ 는 UCFD-SVF의 주파수 영역 2차 커널 적응 필터 계수이다.

Table 2. Number of real multiplications of UCFD-SVF.

Eq.	Number of real multiplications
Eq.(3)	$4 \times (2N(2N+1) + 2N(2N-1)) = 32N^2$
Eq.(8)	$4 \times 2 \times (2N)^2$
Eq.(11)	$2 \times (2N)^2$
Total	$72N^2$

UCFD-SVF의 연산량은 표 2와 같다. 표 1과 비교해 보면 연산량이 대폭 감소되는 것을 알 수 있다. 그러나 UCFD 알고리즘은 연산량이 감소하는 대신에 성능 저하가 발생하는 것으로 알려져 있다. UCFD-SVF가 FD-SVF에 비하여 얼마나 성능 저하가 발생하는지를 알아보기 위해 컴퓨터 모의 실험을 수행하였다. 대부분의 적응 필터 응용들이 기본적으로 시스템 확인을 기반으로 하고 있기 때문

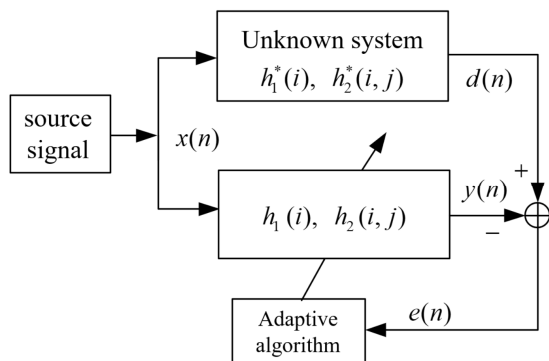


Fig. 1. The Block diagram of the non-linear adaptive system identification.

그림 1. 비선형 적응 시스템 확인의 블록도

에 모의 실험을 위해 비선형 적응 시스템 확인을 실험 환경으로 선정하였다.

그림 1은 실험에 적용된 적응 비선형 시스템 확인의 블록도를 보여준다. 여기서  $h_1^*(i)$ 와  $h_2^*(i,j)$ 는 각각 미지의 시스템의 1차 커널과 2차 커널의 계수이다.  $N = 128$ 로 설정했으며,  $h_1^*(i)$ 은 실제 room 임펄스 응답의 계수로 설정하였고,  $h_2^*(i,j)$ 은  $h_1^*(i)$ 의 자기 상관 행렬을 기반으로 생성하여 사용하였다. 실제의 경우 비선형 성분이 선형 성분에 비하여 상당히 작기 때문에 이를 표현하기 위하여 2차 커널의 계수의 분산은 1차 커널 대비 25dB 작도록 설정하였다.

그림 2는 FD-SVF와 UCFD-SVF의 수렴 특성을 보여 준다. 동일한 조건에서 수렴 특성을 비교하기 위해 두 SVF의 수렴 인자를 동일한 값으로 설정하였다. 그림을 통해 알 수 있듯이 UCFD-SVF가 FD-SVF에 비하여 저조한 수렴 성능을 보임을 알 수 있다. 한편, 비선형 필터의 성능을 선형 필터와 비교해 보기 위해 대표적인 선형 적응 필터인 NLMS 알고리즘을 이용한 적응 필터의 수렴 성능도 함께 표시하였다.

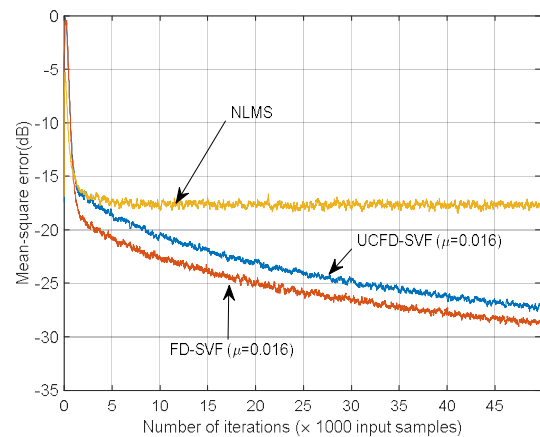


Fig. 2. Comparison of convergence behaviors for FD-SVF and UCFD-SVF.

그림 2. FD-SVF와 UCFD-SVF의 수렴 특성 비교

UCFD-SVF는 식(9)과 식(10)을 생략하여 근사적으로 계산하기 때문에 수렴을 보장하는 수렴 인자의 범위가 FD-SVF보다 좁다. 따라서 실제에 있어서는 두 SVF의 성능은 더 차이가 나게 된다. 그림 3은 두 SVF가 안정적으로 수렴할 수 있는 최대 수렴 인자를 실험적으로 구하여 성능을 비교한 것

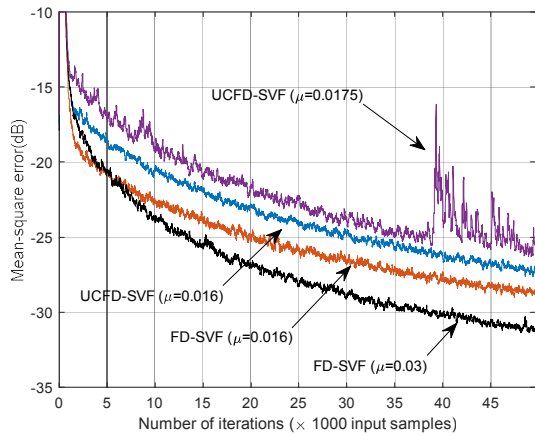


Fig. 3. Comparison of convergence behaviors for FD-SVF and UCFD-SVF with various convergence factors.  
 그림 3. 여러 수렴인자에 대한 FD-SVF와 UCFD-SVF의 수렴 특성 비교

이다. 그림 3에서 보듯이 UCFD-SVF의 경우는 수렴 인자가 0.0175만 되어도 정상적인 수렴을 하지 않는 반면, FD-SVF의 경우는 수렴인자가 0.03일 때에도 안정적으로 수렴함을 알 수 있다. 따라서  $\mu = 0.016$  일 때의 UCFD-SVF와  $\mu = 0.03$  일 때의 FD-SVF가 두 SVF의 실질적인 성능 차이라고 할 수 있다.

한편, 이 성능 차이는 식(10)을 통해 그림 4와 같이 매 블록 연산 시,  $m \geq N$  또는  $n \geq N$ 에 해당하는 영역에 대하여  $h_2^{l+1}(m,n) = 0$ 으로 초기화를 수행하는데서 기인한다. UCFD-SVF의 성능 저하는 그림 4에서 빗금친 영역에 해당하는  $h_2^{l+1}(m,n) = 0$ 이 0이 아니기 때문에 발생 한다. 따라서 UCFD-SVF의 경우 빗금친 영역에 해당하는  $h_2^{l+1}(m,n)$ 이 적응이 진행되면서 어떻게 증가하는 지를 살펴볼 필요가 있다.

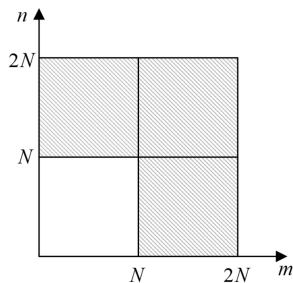
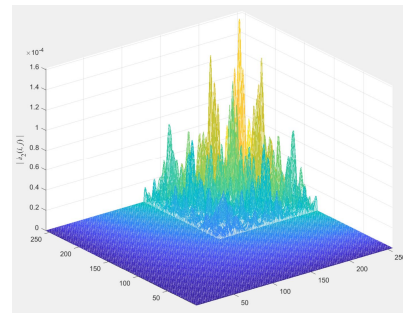


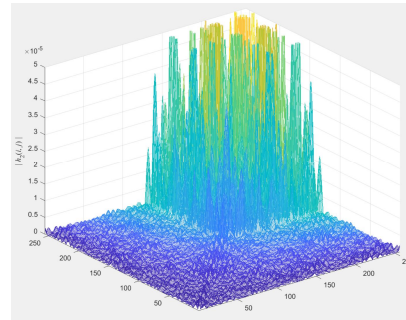
Fig. 4. Area for zero initialization of time-domain filter weights.  
 그림 4. 시간 영역 필터 계수의 초기화 영역

그림 5는 그림 1의 모의 실험 과정에서  $h_2^{l+1}(m,n)$ 의 빗금친 영역의 값을 0을 설정한 후, 적응이 진행

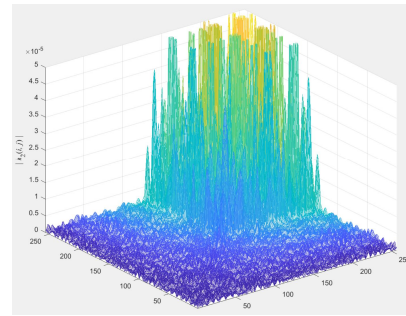
되면서  $h_2^{l+1}(m,n)$ 의 값이 어떻게 변화하는지를 보여 준다. 그림 5 (a)는 빗금친 영역에 대하여  $h_2^{l+1}(m,n) = 0$ 를 수행한 직후의 모습이다. 그리고 그림 5 (b), 그림 5 (c), 그림 5 (d)는 각각 10 블록 적응 후, 20 블록 적응 후, 50 블록 적응 후의 모습이다. 그림 5를



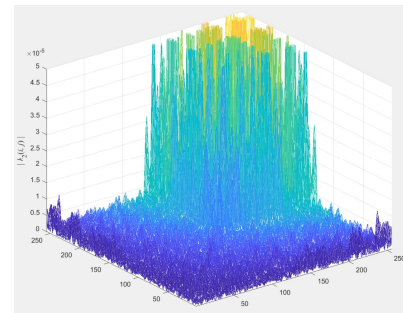
(a)



(b)



(c)



(d)

Fig. 5. Behaviors of time-domain filter weights as the adaptation progresses.

그림 5. 적응이 진행됨에 따라 시간 영역 필터 계수의 변화 모습

통해 적응이 이루어지면서 빗금친 영역의  $h_2^{l+1}(m,n)$  값이 증가되는 것을 볼 수 있으나 그 증가하는 정도가 그리 크지 않음을 알 수 있다.

그림 6은 적응이 진행되면서 빗금친 영역의  $h_2^{l+1}(m,n)$ 의 에너지가 어떻게 증가되는 지를 보여준다.

$$P(l) = \sum_{m=0}^{2N-1} \sum_{n=N}^{2N-1} (h_2^{l+1}(m,n))^2 + \sum_{m=N}^{2N-1} \sum_{n=N}^{2N-1} (h_2^{l+1}(m,n))^2 + \sum_{m=N}^{2N-1} \sum_{n=0}^{2N-1} (h_2^{l+1}(m,n))^2 \quad (12)$$

여기서  $P(l)$ 은  $h_2^{l+1}(m,n)=0$ 이 설정된 이후  $l$  블록이 진행된 이후의 빗금친 영역에서의  $h_2^{l+1}(m,n)$ 의 에너지이다. 그림 6를 통해서도 적응 진행됨에 따라 빗금친 영역에 해당하는  $h_2^l(m,n)$ 의 에너지가 일정 크기에 도달하는데 까지는 어느 정도 시간이 걸림을 알 수 있다.

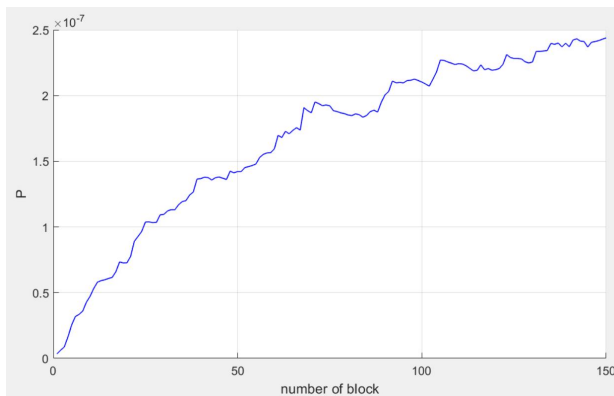


Fig. 6. Energy Behaviors of the filter weights corresponding to the shaded area in Fig.4 as the adaptation progresses.

그림 6. 적응이 진행되면서 그림 4의 빗금친 영역에 해당하는 필터 계수의 에너지 변화 모습

그림 5와 그림 6를 통해 매 블록마다 식(9)와 식(10)을 이용하여  $h_2^{l+1}(m,n)=0$ 을 수행하지 않고 일정 블록마다 수행하더라도 일정 정도의 성능을 유지할 수 있음을 예상할 수 있다. 따라서 첫 번째로 제안하는 방법은 UCFD 알고리즘 수행 시에 특정 블록 간격마다 식(9)와 식(10)을 수행 하는 것이다.

Method 1

$$\mathbf{H}_{2,m1}^{l+1}(p,q) = \mathbf{H}_{2,m1}^l(p,q) + \mu \mathbf{G}^l(p,q)$$

if mod( l , K ) == 0

$$h_{2,m1}^{l+1}(m,n) = 2\text{DIFF}T_{2N}\{\mathbf{H}_{2,m1}^l(p,q)\},$$

$$\mathbf{H}_{2,m1}^{l+1}(p,q) = 2\text{DFFT}_{2N}\{\text{LHLH}_N < h_{2,m1}^{l+1}(m,n) >\},$$

endif

여기서  $\mathbf{H}_{2,m1}^l(p,q)$ 과  $h_{2,m1}^{l+1}(m,n)$ 은 각각 Method 1의 2차 커널 주파수 영역 및 시간 영역 적응 필터 계수이다, mod( )는 모듈로 연산을 의미하며 따라서  $K$  블록마다  $h_{2,m1}^{l+1}(m,n)=0$ 을 수행하게 된다. 연산량은 표 3과 같다. Peak 연산량은 FD-SVF와 동일하나, 평균 연산량은  $K$ 가 증가함에 따라서 상당히 줄어든다.

Table 3. Number of real multiplications of Method 1.

표 3. Method 1의 실수 곱셈 연산량

Eq.	Number of real multiplications
Peak	$32N^2 \log(2N) + 72N^2$
Average	$\frac{1}{K}(32N^2 \log(2N) + 72N^2) + \frac{K-1}{K}(72N^2)$

그림 7은  $K=3$ 일 때의 Method 1의 수렴 성능을 보여준다. FD-SVF에 비하여 약간의 수렴 성능이 저하됨을 알 수 있다. 연산량(실수 곱셈)을 비교해 보면 FD-SVF가 5373952, 제안된 Method 1은 2577749이다. 이는 FD-SVF의 연산량에 48%에 해당한다.

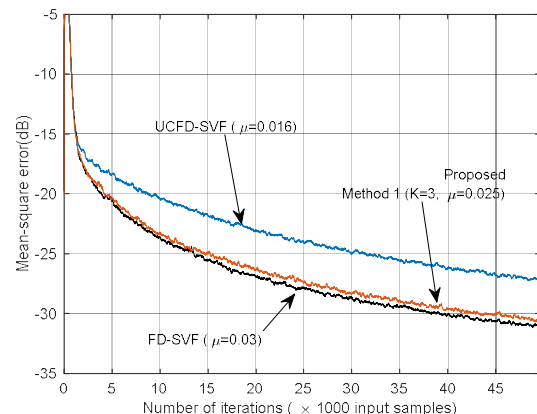


Fig. 7. Convergence behavior of the Method 1 (K=3)

그림 7. Method 1의 수렴 특성 (K=3)

그림 8은  $K=5, K=10, K=15, K=20$ 일 때의 수렴 특성을 보여준다. 표 4는 각각의  $K$ 에 대하여 연산량을 보여준다.  $K$  값이 증가할수록 수렴 성능이 점

차 낮아지지만, 연산량 역시 감소하게 된다. 그림 8에서 적용된 수렴 인자는 실험적으로 안정적인 수렴 성능을 보장하는 수렴 인자를 사용한 것인데 예상대로  $K$  값이 작을수록 큰 값을 적용할 수 있다. 따라서 제안된 Method 1은 연산량과 수렴 성능 사이를  $K$  값을 이용하여 조율할 수 있다. 예들어  $K=20$ 일 경우, 수렴 성능은 UCFD-SVF와 FD-SVF의 중간 정도의 성능을 보이는 반면, 연산량은 FD-SVF의 1/4 정도 된다.

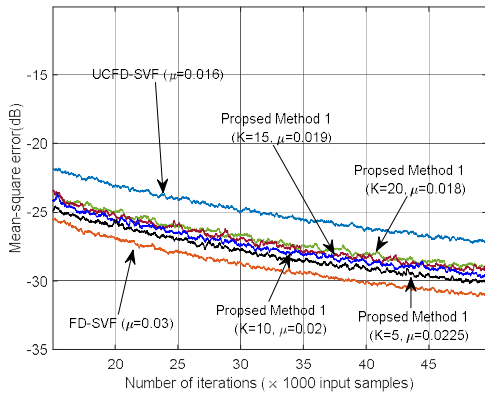


Fig. 8. Convergence behaviors of the Method 1 with various  $K$ s.

그림 8. 여러  $K$  값에 대한 Method 1의 수렴 특성

Table 4. Comparison of the number of real multiplications for the FD-SVF and the Method 1 ( $N = 128$ ).

표 4. FD-SVF와 Method 1의 실수 곱셈 연산량 비교 ( $N = 128$ )

$K$	FD-SVF	Method 1	Percentage
3	5373952	2577749	48.0 %
5	5373952	2018509	37.6 %
10	5373952	1599078	29.7 %
15	5373952	1459268	27.1 %
20	5373952	1389363	25.8 %

한편, Method 1은  $K$  값을 일정한 값으로 고정한다. 실험을 통해 특정  $K$  값일 때 어느 정도 수렴 성능을 예측할 수는 있지만, 원하는 수렴 성능을 보장할 수는 없다. 이를 보완하기 위해 Method 2를 제안한다.

■ Method 2

$$\mathbf{H}_{2,m1}^{l+1}(p,q) = \mathbf{H}_{2,m1}^l(p,q) + \mu \mathbf{G}^l(p,q)$$

$$h_{2,m1}^{l+1}(m,n) = 2\text{DIFFT}_{2N}\{\mathbf{H}_{2,m1}^l(p,q)\}$$

if  $P(l+1) > threshold$

$$\mathbf{H}_{2,m1}^{l+1}(p,q) = 2\text{DIFFT}_{2N}\{LHLH_N < h_{2,m1}^{l+1}(m,n) >\}$$

endif

일정한 성능을 보장하기 위해 그림 4의 빗금친 영역의 에너지인  $P(l+1)$ 이 일정한  $threshold$ 보다 클 때마다  $h_{2,m1}^{l+1}(m,n) = 0$ 을 수행하는 방식이다. 그러나 이 방식은 Method 1과 달리  $P(l+1)$ 을 계산하기 위해  $h_{2,m1}^{l+1}(m,n)$ 을 if 문 내부가 아닌 외부에서 매 블록마다 연산해야 하기 때문에 Method 1에 비하여 연산량이 증가한다. 한편,  $P(l+1)$ 로부터 기인하는 연산량을 감소시키기 위해 실제 구현 시에는 식(12)의 제곱 연산을 절대값 연산으로 대체하였다. 특정한  $threshold$ 가 설정되었을 때,  $K$  값을 적용하는 과정에서 상황에 따라서 변동되므로 평균  $K$  값을  $K_{avr}$  이라면 Method 2의 연산량은 표 5와 같다. 역시 Peak 연산량은 FD-SVF와 동일하다. 그리고  $threshold$ 가 작을수록  $K_{avr}$ 가 작아질 것이므로 연산량은 증가하는 반면 수렴 성능은 향상된다.

Table 5. Number of real multiplications of Method 2.

표 5. Method 2의 실수 곱셈 연산량

Eq.	Number of real multiplications
Peak	$32N^2 \log(2N) + 72N^2$
Average	$\frac{1}{K_{avr}} (32N^2 \log(2N) + 72N^2) + \frac{K_{avr} - 1}{K_{avr}} (16N^2 \log(2N) + 72N^2)$

그림 9는 Method 2의 수렴 특성을 보여준다. 여기서는 그림 7의 결과와 동일한 조건에서 비교해 보기 위해  $K_{avr}$ 이 최대한 3이 되도록 하는  $threshold$ 를 실험을 통해 설정하였다. FD-SVF에 상당히 근접하는 수렴 특성을 보인다. 그림 7의 Method 1과 다른 점은 Method 1의 경우  $K=3$ 일 때, 안정적인 수렴을 보여주는 수렴 인자는 0.025이나, Method 2의 경우는 FD-SVF와 동일하게 0.03으로 설정하여도 안정적으로 수렴하였다. 이는  $K$  값이 고정되어 있지 않고 필요에 따라서 시간 영역 적응 필터를 자주 초기화할 수 있기 때문이다. 실험에서 실제  $K_{avr}$  값은  $K_{avr} = 2.93$ 였다. 따라서 연산량은 3992551로 FD-SVF의 74%에 해당한다.

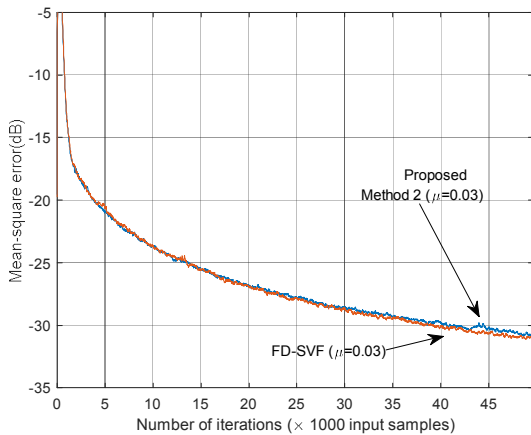


Fig. 9. Convergence behavior of the Method 2 with a specific threshold.

그림 9. 특정 문턱값을 가지는 Method 2의 수렴 특성

한편, Method 2는 일정한 성능을 보장한다는 장점이 있기는 하지만 정상 상태에서  $K_{avr}$ 이 지속적으로 Method 1의  $K$ 와 유사한 값을 유지한다면, 증가된 연산량에도 불구하고 Method 1과 유사한 성능을 보이게 된다. 그렇기 때문에 Method 2의 장점은  $K_{avr}$ 이 지속적으로 일정한 값이 유지되는 정상(stationary) 환경이 아닌 비정상(non-stationary) 환경에서 그 효용성을 찾을 수 있다. 에코 제거의 경우, 미지의 시스템은 음향이 전파되는 공간이 되는데, 이 공간에서 물체의 움직임이 있을 경우 room acoustics가 바뀌게 된다.

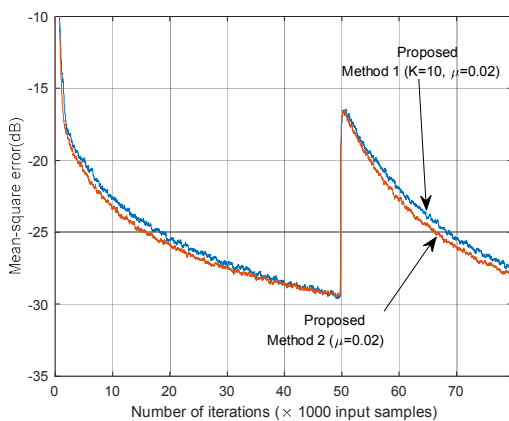


Fig. 10. Comparison of convergence behaviors for the Method 1 and Method 2 under the non-stationary environment.

그림 10. 정상 환경 하에서 Method 1과 Method 2의 수렴 특성 비교

그림 10은 이러한 상황을 시뮬레이션 해보기 위해 적응 필터가 어느 정도 수렴한 상태에서 미지의 시스템의 계수를 변경하여 인위적으로 비정상 상

태를 만든 것이다. Method 1과 동일한 조건에서 비교해 보기 위해 동일한 수렴인자 값을 사용했으며, Method 2의 경우  $K_{avr}$  값이 정상 상태에서 Method 1의  $K$  값인 10과 최대한 유사하도록 실험을 통해 *threshold*를 설정하였다. Fig. 10에서 보듯이 Method 2가 빠른 수렴 속도를 보인다. Method 1의  $K$  값이 고정되어 있는 반면, Method 2의 경우는 평균적인  $K$  값, 즉  $K_{avr}$ 이 Method 1의  $K$  값과 유사하더라도 적응 필터가 적응 단계에서는 일정한 성능을 만족시키기 위해  $K_{avr}$ 이 가변적이 되기 때문이다.

#### IV. 결론

본 논문에서는 대표적인 비선형 적응 필터 중에 하나인 적응 2차 볼테라 필터의 효율적인 구현 방법을 제시하였다. 적응 2차 볼테라 필터의 연산량은 2차 커널의 계수 적응 연산 과정이 대부분을 차지한다. 연산량 감소를 위해 2차원 FFT와 역 FFT를 생략한 UDFD-SVF가 제안되었으나 수렴 성능이 저하된다. 본 논문에서는 UDFD-SVF의 시간 영역 적응 필터 계수가 적응이 진행되면서 급격하게 변화하지 않는다는 점을 이용하여 시간 영역 적응 필터 계수를 주기적으로 초기화는 방식을 제안하였다. 또한 일정한 수렴 특성을 보장하기 위해 문턱값을 이용하여 가변적인 간격으로 시간 영역 적응 필터 계수를 초기화하는 방식을 제안하였고, 비정상 환경에서 우수한 수렴 특성을 가짐을 컴퓨터 모의 실험을 통해 보였다.

#### References

[1] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*, Prentice Hall, 1985.  
 [2] B. Widrow et al., "Stationary and nonstationary learning characteristics of the LMS adaptive filter," *Proc. IEEE*, vol.64, no.8. pp.1151-1162 1976. DOI: 10.1109/PROC.1976.10286  
 [3] S. M. Kuo and D. R. Morgan, "Active noise control: A tutorial review," *Proc. IEEE*, vol.87, no.6, pp.943-973, 1999. DOI: 10.1109/5.763310  
 [4] S. S. Narayan, A. M. Peterson, and M. J. Narasimha, "Transform domain LMS algorithm," *IEEE Trans on Acoust., Speech, Signal Proc.*,



vol.30, no.3, 609–615, 1983.

DOI: 10.1109/TASSP.1983.1164121

[5] J. M. Cioffi, T. Kailath, “Fast, Recursive–Least–Squares Transversal Filters for Adaptive Filtering,” *IEEE Trans on Acoust., Speech, Signal Proc.*, vol.32, no.2, 304–337, 1984.

DOI: 10.1109/TASSP.1984.1164334

[6] F. Gao and W. Snelgrove, “Adaptive Linearization of A Loudspeaker,” *Proc. ICASSP*, pp.3589–3592, 1991. DOI: 10.1109/ICASSP.1991.150251

[7] T. Koh and E. J. Powers, “Second-order volterra filtering and its application to nonlinear system identification,” *IEEE Trans on Acoust., Speech, Signal Proc.*, vol.33, no.6, 1445–1455, 1985.

DOI: 10.1109/TASSP.1985.1164730

[8] S. Im and E. J. Powers, “A Block LMS Algorithm for Third–Order Frequency–Domain Volterra Filters,” *IEEE Signal Processing Lett.*, vol.4, no.3, 75–78, 1997. DOI: 10.1109/97.558643

[9] L. A. Azpoceta–Ruiz et al., “Adaptive Combination of Volterra Kernels and Its Application to Nonlinear Acoustic Echo Cancellation,” *IEEE Trans on Audio, Speech, and Language Processing*, vol.19, no.1, 97–110, 2011. DOI: 10.1109/TASL.2010.2045185

[10] D. Mansour and A. H. Gray Jr. “Unconstrained Frequency–Domain Adaptive Filter,” *IEEE Trans on Acoust., Speech, Signal Proc.*, vol.30, no.5, 726–734, 1982. DOI: 10.1109/TASSP.1982.1163949

---

## BIOGRAPHY

---

### **Ik Joo Chung** (Member)



1986 : BS degree in Electronics Engineering, Seoul National University

1988 : MS degree in Electronics Engineering, Seoul National University

1992 : PhD degree in Electronics Engineering, Seoul National University

1992~ : Professor in Department of Electronics Engineering, Kangwon National University