# Integrated Direct Sub-band Adaptive Volterra Filter and Its Application to Identification of Loudspeaker Nonlinearity

Satoshi Kinoshita
Department of Electrical and Electronic Engineering,
Kansai University
3-3-35, Yamate-cho, Suita-shi, Osaka 564-8680, Japan
Email: k729049@kansai-u.ac.jp

Yoshinobu Kajikawa
Department of Electrical and Electronic Engineering,
Kansai University
3-3-35, Yamate-cho, Suita-shi, Osaka 564-8680, Japan
Email: kaji@kansai-u.ac.jp

***Abstract*—In this paper, we propose a novel realization of sub-band adaptive Volterra filter, which consists of input signal transformation block and only one adaptive Volterra filter. The proposed realization can focus on major frequency band, in which a target nonlinear system has dominant components, by changing the number of taps in each sub-band in order to simultaneously realize high computational efficiency and high identification performance. The proposed realization of sub-band adaptive Volterra filter is applied to the identification of electro-dynamic loudspeaker systems and the effectiveness is demonstrated through some simulations. Simulation results show that the proposed realization can significantly improve the estimation accuracy.**

## I. INTRODUCTION

Loudspeaker systems have linear and nonlinear distortions that may impair the sound quality. To remove the distortions by using digital signal processing, the digital filter must treat not only linearity but also nonlinearity for the target system. The nonlinear signal processing unit is generally realized by Volterra filter (VF) [1]. VFs are based on Volterra series expansion [2], which is a good model for the nonlinearity of loudspeaker systems [3]. One of the modeling methods is to use adaptive Volterra filter (AVF) [4]. Using Volterra series expansion, we have already proposed a nonlinear inverse system to remove the distortions [5]. If AVFs can accurately identify the nonlinearity of loudspeaker systems, the nonlinear inverse system, which is connected in front of loudspeaker systems, can reduce the nonlinear distortions, so that the radiated sound can have high quality.

However, the computational complexity of VFs is much greater than that of linear filters. To reduce the computational complexity, adaptive simplified Volterra filters (ASVF) have been proposed in [6]–[8]. In [6], small coefficients are replaced to delay unit based on the characteristics of the first-order VF. In [7] and [8], the second-order VF is approximated by setting some coefficients far from the main diagonal to zero because the coefficients with the most significant amplitude lay on the diagonals near the main one. However, the ASVFs can reduce a little computational complexity and may not be able to ensure sufficient estimation accuracy because the removed coefficients may be included in a large value.

In [9], [10], the parallel-cascade Volterra filter (APCVF) has been proposed. The APCVF is based on the use of singular value decomposition. The APCVF can reduce the computational complexity while maintaining the identification accuracy because many branches with small eigenvalues can be removed. However, the appropriate number of branches is unknown before the identification, that is, many trials are required to obtain the appropriate number of branches and the APCVF is no suitable for realtime applications.

On the other hand, sub-band Volterra filters have been proposed in [11]–[14]. In [11], linear filters included in the parallel cascade realization are divided into sub-bands. In [12]–[14], first- and second-order Volterra filters are regarded as one large linear filter and sub-band processing is applied to this linear filter. We call this sub-band approach “indirect sub-band AVF (ISAVF)”.

Separately from the above realizations, we have already proposed the nonlinear inverse system using the multirate signal processing to reduce the computational complexity [15]. The multirate signal processing technique proposed in [15] has been generalized to higher order VFs in [16]. AVFs have also been expanded by using the multirate signal processing in case of the system identification [17]. In [17], adaptive sparse-interpolated Volterra filters (ASIVF) can have many zero coefficients by the band limitation of input signal and can consequently reduce the computational complexity. However, the performance depends on the characteristics of Volterra kernels of a target nonlinear system. ASIVF requires band-limited input signal (less than $F_s/4$ where $F_s$ is sampling frequency) to reduce the computational complexity. In this case, ASIVF cannot estimate the components outside the band [0, $F_s/4$] of the target nonlinear system.

To solve this problem, we have also proposed a sub-band adaptive Volterra filter to identify nonlinear systems [18]. We call this sub-band realization “direct sub-band AVF (DSAVF)” in this paper. DSAVF proposed in [18] is one of extensions of AVFs with sparse coefficients proposed in [15], [17] and

978-0-9928-6265-7/16/$31.00 ©2016 IEEE

consists of VFs with sparse coefficients and different band limitations in each sub-band, respectively. DSAVF can treat all frequency bands by parallel processing while maintaining high identification performance. If the distortion is concentrated in the low frequency like loudspeaker systems, DSAVF can deploy more taps against low frequency band. However, DSAVF has a disadvantage that the system configuration becomes so complicated as the number of sub-bands ($D$) increases because the number of combinations of input signal increases exponentially. In this paper, we propose a realization for DSAVF, which consists of input signal transformation (IST) block and only one AVF to integrate many sub-bands. We call this realization Integrated DSAVF (IDSAVF). IDSAVF can focus on dominant frequency band by changing the tap lengths in each sub-band.

## II. Sub-band VOLTERRA filters

### A. Discrete Volterra Series

Loudspeaker systems can be modeled by using the Volterra series expansion [2]. For simplicity, we assume that Volterra kernels have a finite memory of $N$ and do not treat the third or more terms. The input-output relation is represented by

$$\begin{aligned} y(n) &= y_1(n) + y_2(n), \\ y_1(n) &= \sum_{k_1=0}^{N-1} h_1(k_1)x(n-k_1), \\ y_2(n) &= \sum_{k_1=0}^{N-1}\sum_{k_2=0}^{N-1} h_2(k_1,k_2)x(n-k_1)x(n-k_2), \end{aligned} \quad (1)$$

where $x(n)$ and $y(n)$ are the sampled input and output signals, respectively; $y_1(n)$ and $y_2(n)$ are the output signals for first- and second-order discrete Volterra kernels, respectively; $h_1(k_1)$ and $h_2(k_1, k_2)$ are the first- and second-order discrete Volterra kernels, respectively.

### B. Volterra Sampling Theorem

From Eq. (1), even if the sampling frequency is equal to the Nyquist frequency, the output signals of second-order Volterra kernel contain the components more than $f_s/2$, where $f_s$ is sampling frequency, so that aliasing occurs. In order to avoid the aliasing, the input signal or the second-order Volterra kernel must be band-limited to $f_s/4$. This band-limitation is known as the Volterra sampling theorem. However, the Volterra sampling theorem is not required to the system identification using AVFs [19].

### C. Elimination of Redundancy in Band-limited Volterra Filters

In [16], the multi-rate signal processing has been applied to VFs. If the frequency band of the second-order VF is limited to $f_s/2D$ or less, where $D$ is the number of sub-bands, the references [15], [17] have introduced reduction methods of the computational complexity to $1/D^2$. These methods are based on the multi-rate signal processing. In [15], [17], the coefficients of the second-order VF $h_2^{'}(k_1, k_2)$ in time domain are given by

$$h_2^{'}(k_1,k_2) = \begin{cases} h_2(k_1,k_2) & k_1,k_2 = 0, D, 2D \cdots \\ 0 & Otherwise. \end{cases} \quad (2)$$

If the imaging components in the output signal of the VF $h_2^{'}(k_1, k_2)$ is removed and the gain is adjusted, the output signals of VFs $h_2(k_1, k_2)$ and $h_2^{'}(k_1, k_2)$ are equivalent. Concretely, if the frequency band of the second-order VF is limited to $f_s/2D$ or less and the output signal of the second-order VF $h_2^{'}(k_1, k_2)$ is multiplied by $D^2$, then the output signals of VF $h_2(k_1, k_2)$ and $h_2^{'}(k_1, k_2)$ are equivalent. Here, $h_2^{'}(k_1, k_2)$ is more efficient than $h_2(k_1, k_2)$ in aspect of the computational complexity because there is only one non-zero coefficient every $D^2$ coeffcents and the other multiplications can be ignored in the output calculation.

### D. Derivation of Sub-band Volterra Filter

We define the $N$-dimensional vectors $\mathbf{x}_1(n)$ and $\mathbf{x}_2(n)$ and the $N\times N$ matrix $\mathbf{H}$ as

$$\mathbf{x}_1(n) = \mathbf{x}_2(n) = [x(n), x(n-1), \cdots, x(n-(N-1))]^{\mathrm{T}},$$

$$\mathbf{H} = \begin{pmatrix} h_2(0,0) & \cdots & h_2(0,N-1) \\ \vdots & \ddots & \vdots \\ h_2(N-1,0) & \cdots & h_2(N-1,N-1) \end{pmatrix}. \quad (3)$$

Hence, $y_2(n)$ in Eq. (1) is written in the form

$$y_2(n) = \mathbf{x}_1^{\mathrm{T}}(n)\mathbf{H}\mathbf{x}_2(n). \quad (4)$$

If we define a $N\times N$ decimation matrix $\mathbf{A}_N$, the transformed vectors $\mathbf{x}_1^{'}(n)$ and $\mathbf{x}_2^{'}(n)$ and the transformed matrix $\mathbf{H}^{'}$ are expressed as

$$\begin{aligned} \mathbf{x}_1^{'}(n) &= \mathbf{x}_2^{'}(n) = \mathbf{A}_N\mathbf{x}_1(n) = \mathbf{A}_N\mathbf{x}_2(n), \\ &= [x_0(n), x_1(n), \cdots, x_{N-1}(n)]^{\mathrm{T}}, \\ \mathbf{H}^{'} &= \mathbf{A}_N^{\mathrm{T}}\mathbf{H}\mathbf{A}_N, \\ &= \begin{pmatrix} h_{2,0,0}(0,0) & h_{2,0,1}(0,0) & \cdots & h_{2,0,N-1}(0,0) \\ h_{2,1,0}(0,0) & h_{2,1,1}(0,0) & \cdots & h_{2,1,N-1}(0,0) \\ \vdots & \vdots & \ddots & \vdots \\ h_{2,N-1,0}(0,0) & h_{2,N-1,1}(0,0) & \cdots & h_{2,N-1,N-1}(0,0) \end{pmatrix}. \end{aligned} \quad (5)$$

Hence, the transformed output signal $y_2^{'}(n)$ is written in the form

$$\begin{aligned} y_2^{'}(n) &= \mathbf{x}_1^{'}(n)^{\mathrm{T}}\mathbf{H}^{'}\mathbf{x}_2^{'}(n) \\ &= (\mathbf{A}_N\mathbf{x}_1(n))^{\mathrm{T}}\mathbf{A}_N^{\mathrm{T}}\mathbf{H}\mathbf{A}_N(\mathbf{A}_N\mathbf{x}_2(n)) \\ &= \mathbf{x}_1^{\mathrm{T}}(n)\mathbf{A}_N^{\mathrm{T}}\mathbf{A}_N^{\mathrm{T}}\mathbf{H}(\mathbf{A}_N\mathbf{A}_N)\mathbf{x}_2(n) \\ &= \mathbf{x}_1^{\mathrm{T}}(n)(\mathbf{A}_N\mathbf{A}_N)^{\mathrm{T}}\mathbf{H}(\mathbf{A}_N\mathbf{A}_N)\mathbf{x}_2(n). \end{aligned} \quad (6)$$

If the decimation matrix $\mathbf{A}_N$ is symmetric and orthogonal in Eq. (6), $\mathbf{A}_N\mathbf{A}_N$ becomes the unit matrix, so that $y_2'(n)$ is equal to $y_2(n)$. In other words, if we transform the input signal and second-order VF by the symmetric and orthogonal matrix $\mathbf{A}_N$, $y_2'(n)$ is equal to $y_2(n)$.

Let us discuss transforming $\mathbf{x}_1(n)$, $\mathbf{x}_2(n)$ and $\mathbf{H}$ by the $D\times D$ symmetric and orthogonal decimation matrix $\mathbf{A}_D$. $\mathbf{x}_1(n)$, $\mathbf{x}_2(n)$ and $\mathbf{H}$ are separated by partial vectors and matrix which have $D$ and $D\times D$ samples respectively. If we define the $D$-dimensional partial vectors $\mathbf{x}_{d,D}(n)$ and the $D\times D$ partial matrixes $\mathbf{H}_{d_1,d_2,D}$ by

$$\begin{aligned}\mathbf{x}_{d,D}(n) &= [x(n-dD), x(n-1-dD), \cdots, x(n-(D-1)-dD)]^{\mathrm{T}},\\ \mathbf{H}_{d_1,d_2,D} &= \begin{pmatrix} h_2(d_1D, d_2D) & \cdots & h_2(d_1D, D-1+d_2D) \\ \vdots & \ddots & \vdots \\ h_2(D-1+d_1D, d_2D) & \cdots & h_2(D-1+d_1D, D-1+d_2D)\end{pmatrix},\end{aligned} \tag{7}$$

where $d$, $d_1$ and $d_2$ are integers from 0 to $N/D-1$, respectively. Eq. (3) is written in the form

$$\begin{aligned}\mathbf{x}_1(n) &= \mathbf{x}_2(n) = [\mathbf{x}_{0,D}^{\mathrm{T}}(n), \cdots, \mathbf{x}_{N/D-1,D}^{\mathrm{T}}(n)]^{\mathrm{T}},\\ \mathbf{H} &= \begin{pmatrix} \mathbf{H}_{0,0,D} & \cdots & \mathbf{H}_{0,N/D-1,D} \\ \vdots & \ddots & \vdots \\ \mathbf{H}_{N/D-1,0,D} & \cdots & \mathbf{H}_{N/D-1,N/D-1,D}\end{pmatrix},\end{aligned} \tag{8}$$

where $\mathbf{x}_1(n)$, $\mathbf{x}_2(n)$ and $\mathbf{H}$ are split by $D$ samples by using $\mathbf{x}_{d,D}(n)$ and $\mathbf{H}_{d_1,d_2,D}$, respectively. If Eq. (8) is substituted for Eq. (4), $y_2(n)$ is represented by

$$y_2(n) = \sum_{d_1=0}^{N/D-1}\sum_{d_2=0}^{N/D-1} \mathbf{x}_{d_1,D}^{\mathrm{T}}(n)\mathbf{H}_{d_1,d_2,D}\mathbf{x}_{d_2,D}(n). \tag{9}$$

Let us extend the discussion of the symmetric and orthogonal matrix $\mathbf{A}_N$ to $\mathbf{A}_D$. If we define the $D$-dimensional transformed vectors $\mathbf{x}'_{d,D}(n)$ and the $D\times D$ transformed matrix $\mathbf{H}'_{d_1,d_2,D}$,

$$\begin{aligned}\mathbf{x}'_{d,D}(n) &= \mathbf{A}_D\mathbf{x}_{d,D}(n)\\ &= [x_0(n-dD), x_1(n-dD), \cdots, x_{D-1}(n-dD)]^{\mathrm{T}},\\ \mathbf{H}_{d_1,d'_2,D} &= \mathbf{A}_D^{\mathrm{T}}\mathbf{H}_{d_1,d_2,D}\mathbf{A}_D\\ &= \begin{pmatrix} h_{2,0,0}(d_1D, d_2D) & \cdots & h_{2,0,D-1}(d_1D, d_2D) \\ \vdots & \ddots & \vdots \\ h_{2,D-1,0}(d_1D, d_2D) & \cdots & h_{2,D-1,D-1}(d_1D, d_2D)\end{pmatrix}.\end{aligned} \tag{10}$$

$y_2(n)$ is represented by

$$y_2(n) = \sum_{d_1=0}^{N/D-1}\sum_{d_2=0}^{N/D-1} \mathbf{x}_{d_1,D}^{'\mathrm{T}}(n)\mathbf{H}'_{d_1,d_2,D}\mathbf{x}'_{d_2,D}(n). \tag{11}$$

Let us define the $D\times N/D$ input signals matrix $\mathbf{X}'_D(n)$ as

$$\mathbf{X}'_D(n) = [\mathbf{x}'_{0,D}(n), \mathbf{x}'_{1,D}(n), \cdots, \mathbf{x}'_{N/D-1,D}(n)], \tag{12}$$

where $\mathbf{X}'_D(n)$ consists of a universal set of the $N/D$ $D$-dimensional vectors $\mathbf{x}'_{d,D}(n)$. Hence $\mathbf{X}'_D(n)$ is also equal to the transposed matrix of a universal set of the $D$ $N/D$-dimensional vectors as

$$\mathbf{X}'_D(n) = [\mathbf{x}''_{0,D}(n), \mathbf{x}''_{1,D}(n), \cdots, \mathbf{x}''_{D-1,D}(n)]^{\mathrm{T}}, \tag{13}$$

where $N/D$-dimensional vector $\mathbf{x}''_{s,D}(n)$ and the $N/D\times N/D$ matrix $\mathbf{H}''_{s_1,s_2,D}$ are given by each element in $\mathbf{x}'_{d,D}(n)$ and $\mathbf{H}'_{d_1,d_2,D}$ as

$$\begin{aligned}\mathbf{x}''_{s,D}(n) &= [x_s(n), x_s(n-D), \cdots, x_s(n-(N/D-1)D)]^{\mathrm{T}},\\ \mathbf{H}''_{s_1,s_2,D} &= \begin{pmatrix} h_{2,s_1,s_2}(0,0) & \cdots & h_{2,s_1,s_2}(0,(N/D-1)D) \\ \vdots & \ddots & \vdots \\ h_{2,s_1,s_2}((N/D-1)D,0) & \cdots & h_{2,s_1,s_2}((N/D-1)D,(N/D-1)D)\end{pmatrix},\end{aligned} \tag{14}$$

where $s$, $s_1$ and $s_2$ are integers from 0 to $D-1$, respectively. $\mathbf{x}''_{s,D}(n)$ shows input vector of the $S$-th sub-band in the case of the division $D$. Equation (11) is written in the form

$$y_2(n) = \sum_{s_1=0}^{D-1}\sum_{s_2=0}^{D-1} \mathbf{x}_{s_1,D}^{''\mathrm{T}}(n)\mathbf{H}''_{s_1,s_2,D}\mathbf{x}''_{s_2,D}(n). \tag{15}$$

Note that each sample interval in Eq. (14) is $D$, so that this interval is decimated to $1/D$. In general, the gain of each row of $\mathbf{A}_D$ is 1, so that $\mathbf{A}_D\mathbf{A}_D$ is $1/D$ times unit matrix. In this case, the output signal of the second-order VF is represented by

$$\begin{aligned}y_2(n) &= D^2\sum_{s_1=0}^{D-1}\sum_{s_2=0}^{D-1}\sum_{k_1=0}^{N/D-1}\sum_{k_2=0}^{N/D-1} h_{2,s_1,s_2}(k_1,k_2)\\ &\quad\cdot\ x_{s_1}(n-k_1)x_{s_2}(n-k_2),\end{aligned} \tag{16}$$

where $s_1$ and $s_2$ represent the index of sub-band, respectively. The large $s_1$ and $s_2$ are high frequency sub-band. Here, the input signals $x_{s_1}(n)$ and $x_{s_2}(n)$ and the second-order VF $h_{2,s_1,s_2}(k_1,k_2)$ are obtained according to the following procedure. First of all, $x(n)$ and $h_2(k_1,k_2)$ are split into $D$ regions and $D^2$ regions as shown in Eq. (8), respectively. Next, $x(n)$ and $h_2(k_1,k_2)$ are transformed by $\mathbf{A}_D$, respectively. Then, all input signals and filter coefficients are sorted in the order of time for each frequency band. Finally, the sorted input signals and coefficients are decimated to $1/D$ and $1/D^2$, respectively. From Eq. (16), the number of DSAVFs corresponds to the combinations of the two input signals. For example, when the input signals are divided into $D$ sub-bands, the number of DSAVFs is equal to $D^2$. In addition, when the tap length of the second-order VF is $N$, the tap length of $h'_{2,d_1,d_2,D}(k_1,k_2)$ is equal to $N/D^2$. If only VF $h'_{2,0,0,D}(k_1,k_2)$ is used, the output signal agrees with those of the methods proposed in [15], [17].

## III. New Realization of DSAVF

### A. New Realization Using IST and AVF

We propose a novel realization for DSAVF by using IST and AVF. The proposed realization can simplify the system configuration of DSAVF that generally consists of many filters.

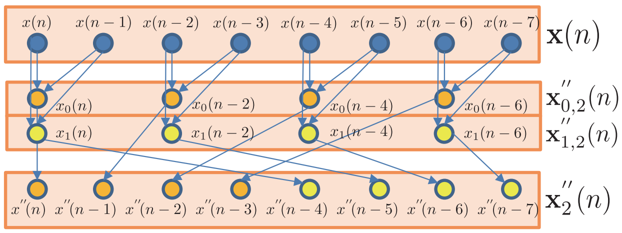



Fig. 1. Relation of $\mathbf{x}(n)$, $\mathbf{x}''_{d,D}(n)$ and $\mathbf{x}''_D(n)$.

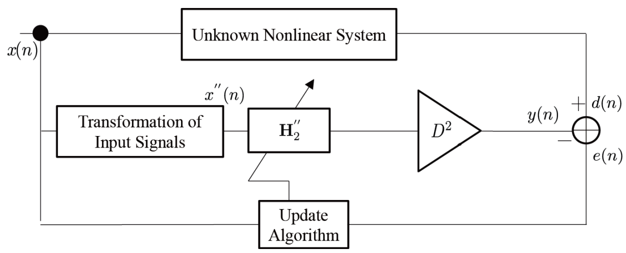



Fig. 2. New realization for DSAVF in the system identification.

If we define the $N$-dimensional vector $\mathbf{x}''_D(n)$ and the $N \times N$ matrix $\mathbf{H}''_D$ by

$$\begin{aligned} \mathbf{x}''_D(n) &= [\mathbf{x}''^{\mathrm{T}}_{0,D}(n), \mathbf{x}''^{\mathrm{T}}_{1,D}(n), \cdots, \mathbf{x}''^{\mathrm{T}}_{D-1,D}(n)]^{\mathrm{T}}, \\ \mathbf{H}''_D &= \begin{pmatrix} \mathbf{H}''_{0,0,D} & \cdots & \mathbf{H}''_{0,D-1,D} \\ \vdots & \ddots & \vdots \\ \mathbf{H}''_{D-1,0,D} & \cdots & \mathbf{H}''_{D-1,D-1,D} \end{pmatrix}. \end{aligned} \tag{17}$$

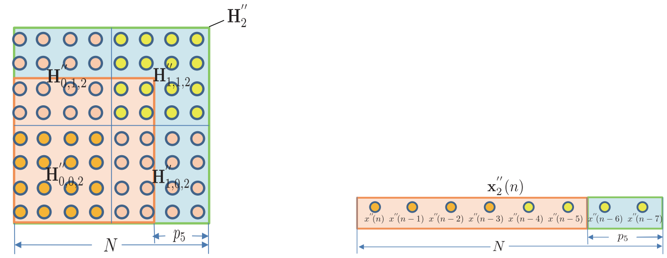
$\mathbf{x}''_D(n)$ is $N$-dimensional vector, which consists of $\mathbf{x}''_{0,D}(n)$, $\mathbf{x}''_{1,D}(n)$, $\cdots$, $\mathbf{x}''_{D-1,D}(n)$. For example, if $D$ is equal to 2, $\mathbf{x}''_2(n)$ is the vector which consists of $\mathbf{x}''_{0,2}(n)$ and $\mathbf{x}''_{1,2}(n)$. Since both $\mathbf{x}''_D(n)$ and $\mathbf{x}(n)$ are $N$-dimensional vectors, the relationship of $\mathbf{H}''_D$ and $\mathbf{H}$ is similar to that of $\mathbf{x}''_D(n)$ and $\mathbf{x}(n)$. Eq. (15) is written as

$$y_2(n) = \mathbf{x}''^{\mathrm{T}}_D(n)\mathbf{H}''_D\mathbf{x}''_D(n). \tag{18}$$

Hence, Eq. (16) is written in the form

$$y_2(n) = D^2 \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{N-1} h''_2(k_1, k_2) x''(n-k_1) x''(n-k_2), \tag{19}$$

where $h''_2(k_1, k_2)$, $x''(n-k_1)$ and $x''(n-k_2)$ are arranged according to the orders of $h_{2,d_1d_2}(k_1, k_2)$, $x_{d_1}(n-k_1)$ and $x_{d_2}(n-k_2)$ as shown in Eq. (17), respectively. Equation (19) shows that a single second-order VF can calculate the total output signal of all sub-bands. For example, in the case of $D = 2$, the relationships among $\mathbf{x}(n)$, $\mathbf{x}''_{d,D}(n)$, and $\mathbf{x}''_D(n)$ are shown in Fig. 1. The transformation of input signals as shown in Fig. 1 and $D^2$ times gain adjustment can realize the system configuration for DSAVF shown in Fig. 2.




(a) IDSAVFs. (b) Input signal vector.

Fig. 3. Assignment of $N$.

### *B. Advantage of truncated IDSAVF*

In this section, we explain the advantage of IDSAVF for computational complexity. Figure 3 shows the tap arrangement of input signal in the IST and the relation between tap length $N$ and each sub-band in the AVF in case of $D = 2$. In Fig. 3, $p_5$ is the truncated length. IDSAVF has the characteristic as follows: the bottom-left area is the component of low frequency, the top-right area is the component of high frequency, the bottom-right and top-left are the components of bilinear. If any number of divisions $D$ is selected except for 1, truncated IDSAVF can delete coefficients in the order from the high-frequency. Therefore, the truncated IDSAVF can process wideband singals and focus on lower sub-bands. In other words, the computational complexity can be reduced while maintaining the estimation accuracy. The truncated IDSAVF is especially efficient for loudspeaker nonlinearity identification because the nonlinear distortions of loudspeakers are usually concentrated in the low frequency band.

### *C. Computational Complexity in Proposed Method*

In this section, let us compare the computational complexity of IDSAVF (proposed) with those of the conventional reduction methods. TABLE I shows the comparison of the computational complexity, where $p_0$, $p_1$, $p_2$, $p_3$, $p_4$ and $p_5$ are the truncated tap length, the length of delay unit, the number of deleted FIR filters, the number of deleted branches and the decimation rate, respectively.

TABLE I
CALCULATION OF AVF.

| | Multiplications per output and update | Example |
|---|---|---|
| AVF | $2(N-p_0)^2+2N-2p_0+4$ | $29044(N=128, p_0=8)$ |
| ASVF (delay) | $2(N-p_1)^2+2N-2p_1+4$ | $29044(N=128, p_1=8)$ |
| ASVF (diagonal) | $2N^2+2N+4-2p_2(p_2+1)$ | $29068(N=128, p_2=44)$ |
| APCVF | $2N^2+2(2-p_3)N-4p_3+2+3(N-p_3)$ | $29195(N=128, p_3=17)$ |
| ASIVF | $2(N/p_4)^2+2N/p_4+4$ | $8346(N=128, p_4=2)$ |
| IDSAVF (Proposed) | $2(N-p_5)^2+2(N-p_5)+3$ | $29045(N=128, p_5=8)$ |

## IV. SIMULATION RESULTS

### *A. Simulation of Loudspeaker Nonlinearity Identification*

In order to compare the conventional and proposed methods, we identify the second-order nonlinear distortion of a loudspeaker system. In the simulation, the parameters $p_0$, $p_1$, $p_2$, $p_3$, $p_4$, $p_5$, and $N$ were set so that the computational complexity of all methods except for ASIVF becomes almost

TABLE II
SIMULATION CONDITIONS.

| | |
|---|---|
| Transform method | Hadamard |
| Input voltage | 1.0V |
| System length of an unknown nonlinear system ($N$) | 128 |
| Conventional parameters ($p_0, p_1, p_2, p_3, p_4$) | 8, 8, 44, 17, 2 |
| Proposal parameter ($p_5$) | 8 |
| Number of sub-bands ($D$) | 2 |
| Update algorithm of AVF | NLMS |
| Step size parameter of AVF | 1.0 |
| Sampling frequency | 8kHz |

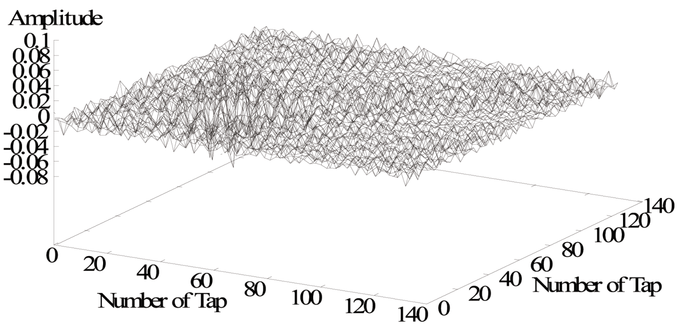



Fig. 4. A second-order unknown system.

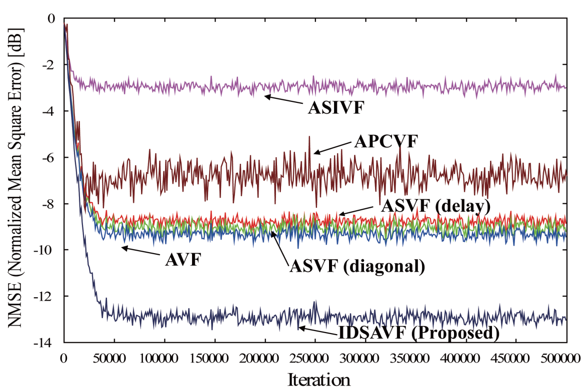



Fig. 5. Comparison of convergence properties between the IDSAVF and the conventional AVFs.

same. The input signal of ASIVF is band-limited less than $F_s/4$, so that the computational complexity of ASIVF is almost 28 % of other methods. Table II and Fig. 4 show the simulation conditions and an unknown second-order Volterra kernel in a nonlinear loudspeaker system, respectively. In order to evaluate the identification accuracy, we define NMSE as

$$NMSE = 10 \log_{10} \frac{\sum e^2(n)}{\sum d^2(n)}. \tag{20}$$

Figure 5 shows that NMSE of IDSAVF is the lowest in the steady state. This is because the IDSAVF can preserve the most important nonlinear components (in this case, lower frequency components) while reducing the computational complexity. The ASIVF is the highest because of low computational complexity.

## V. CONCLUSION

In this paper, we proposed a novel realization method for DSAVFs. The proposed realization consists of the IST block and only one AVF. Simulation results show that the proposed realization can significantly improve the estimation accuracy while reducing the computational complexity. In the future, we will develop an automatic parameter setting method for IDSAVF.

## ACKNOWLEDGMENT

This work is supported by MEXT-Supported Program for the Strategic Research Foundation at Private Universities, 2013-2017

## REFERENCES

[1] Y. Kajikawa and Y. Nomura, “Design of Nonlinear Inverse System by Means of Adaptive Volterra Filters,” Electron. Commun. Japan, Part 3 (Fundamental Electronic Science), vol. 80, no. 8, pp. 36–45, 1997.
[2] M. Schetzen, The Volterra and Wiener Theories of Nonlinear Systems, Kriger, Florida, 1989.
[3] A. J. Kaizer, “Modeling of the Nonlinear Response of an Electrodynamics Loudspeaker by a Volterra Series Expansion,” J. Audio Eng. Soc., vol. 35, no. 6, pp. 421–432, Jun. 1987.
[4] V. John Mathews, “Adaptive Polynomial Filters,” IEEE SP Magazine, vol.8, no.3, pp. 10–26, Jul. 1991.
[5] Y. Kajikawa, “The Adaptive Volterra Filter: Its Present and Future,” Electron. Commun. Japan, Part 3 (Fundamental Electronic Science), vol. 83, no. 12, pp. 51–61, 2000.
[6] A. Stenger, L. Trautmann, and R. Rabenstein, “Nonlinear Acoustic Echo Cancellation with 2nd Order Adaptive Volterra Filters,” in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), 1999, pp. 877–880.
[7] A. Fermo, A. Carini, and G. L. Sicuranza, “Simplified Volterra Filters for Acoustic Echo Cancellation in GSM Receivers,” in Proc. European Signal Processing Conference (EUSIPCO), Tampere, Sep. 2000.
[8] F. Kuech and W. Kellermann, “Nonlinear Line Echo Cancellation Using A Simplified Second Order Volterra Filter,” in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), 2002, pp. 1117–1120.
[9] T. M. Panicker and V. J. Mathew, “Parallel-Cascade Realizations and Approximations of Truncated Voterra Systems,” IEEE Trans. on Signal Processing, vol. 46, no. 10, pp. 2829–2832, 1998.
[10] T. M. Panicker, V. J. Mathews, and G. L. Sicuranza, “Adaptive Parallel-Cascade Truncated Volterra Filters,” IEEE Trans. Signal Processing, vol. 46, no. 10, pp. 2664–2673, 1998.
[11] H. Furuhashi, Y. Kajikawa, and Y. Nomura, “Linearization of Loudspeaker Systems Using a Subband Parallel Cascade Volterra Filter,” IEICE Trans. on Fundamentals, vol. E90-A, no. 8, pp. 1616–1619, Aug, 2007.
[12] D. Zhou, V. DeBrunner, Y. Zhai, and M. Yeary, “Efficient Adaptive Nonlinear Echo Cancellation Using Sub-band Implementation of The Adaptive Volterra Filter,” in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), 2006, pp. 277–280.
[13] T. G. Burton, R. A. Goubran, and F. Beaucoup, “Nonlinear System Identification Using a Subband Adaptive Volterra Filter,” IEEE Trans. Instrum. Meas., vol. 58, no. 5, pp. 1389–1397, May 2009.
[14] T. G. Burton and R. A. Goubran, “A Generalized Proportionate Subband Adaptive Second-Order Volterra Filter for Acoustic Echo Cancellation in Changing Environments,” IEEE Trans. on Audio, Speech and Language Processing, vol. 19, no. 8, pp. 2364–2373, 2011.
[15] S. Kinoshita, Y. Kajikawa, and Y. Nomura, “Volterra Filters Using Multirate Signal Processing and Their Application to Loudspeaker Systems,” in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), 2001, pp. 3497–3500.
[16] D. Schwingshackl and G. Kubin, “Polyphase Representation of Multirate Nonlinear Filters and Its Applications,” IEEE Transactions on Signal Processing, vol. 55, no. 5, pp. 2145–2157, May 2007.
[17] E. L. O. Batista, O. J. Tobias, and R. Seara, “A Sparse-Interpolated Scheme for Implementing Adaptive Volterra Filters,” IEEE Transactions on Signal Processing, vol. 58 no. 4, pp. 2022–2035, Apr. 2010.
[18] S. Kinoshita, Y. Kajikawa, and Y. Nomura, “Adaptive Volterra Filters Using Multirate Signal Processing and Their Application to Identification of Loudspeaker Systems,” Electron. Commun. Japan, Part 3 (Fundamental Electronic Science), vol. 87, no. 7, pp. 45–54, Jul. 2004.
[19] W. A. Frank, “Sampling Requirements for Volterra System Identification,” IEEE Signal Processing Letters, vol. 3, no. 9, pp. 266–268, Sep. 1996.