A Design Method of a Nonlinear Inverse System by the Adaptive Volterra Filter.
- The Application of the Summational Affine Projection Algorithm -

Yoshinobu Kajikawa and Yasuo Nomura
Department of Electronics, Faculty of Engineering, Kansai University
3-3-35, Yamate-cho, Suita-shi, Osaka 564, Japan
kaji@joho.densi.kansai-u.ac.jp

ABSTRACT
In this paper, we propose the summational projection algorithm which has the convergence properties of high speed and high accuracy under high noise and colored input signal. We particularly discuss the adaptive algorithm for the adaptive Volterra filter which can be used to identify and design nonlinear systems. The proposal algorithm realizes the these convergence properties by controlling the length of the block in the updating algorithm. First of all, we present the general type of the proposed summational projection algorithm. Next, we show that the proposal algorithm is effective in the identification of nonlinear systems. Finally, we apply the proposed algorithm to the design method of a nonlinear inverse system.

1. Introduction
It is desirable that the distortions of transmission systems are eliminated from the viewpoint of information transmission. The distortions of transmission systems can be classified into linear and nonlinear distortions. Up to now, the elimination of the linear distortion using the linear filters has been studied [1]. However, if the linear distortion is eliminated for the system which has nonlinearity, the problem that the nonlinear distortion increases occurs. Therefore, it is necessary to eliminate both the linear and the nonlinear distortions at the same time. Recently, the Volterra series expansion [2] has been applied successfully to the analysis, design and identify of nonlinear systems [3-10]. In [7-10], nonlinear distortions in loudspeakers are reduced using Volterra filter. In these methods, we need to identify the Volterra kernels of loudspeakers in order to reduce the nonlinear distortion of loudspeakers. The adaptive Volterra filter [11] has widely been used to identify the Volterra kernel. However, when we identify the Volterra kernels in the actual system, we must consider the existence of additive noise, the variation of additive noise, and the characteristic-variation of unknown system. On the other hand, the outputs of the second-order Volterra filter are obtained by multiplying the filter coefficients by the product of the input signal. Therefore, the convergence property of the adaptive Volterra filter using the LMS algorithm becomes poor. Because the product of the input signal is colored signal even if the input signal is white signal. Consequently, the updating algorithm of the adaptive Volterra filter needs to be robust to the colored signal and the variations of environment. Therefore we propose a summational affine projection algorithm, which realizes the convergence properties of high speed and high accuracy under high additive noise and colored input signal. The proposed algorithm realizes the these convergence properties by controlling the length of the block in the updating algorithm.

2. The Volterra Series Expansion
Now, a discrete-time, time-invariant, and causal nonlinear system with finite memory can be expressed by means of an extension of the following Volterra series expansion [2].

\[
y(n) = h_0 + \sum_{k_1=0}^{N-1} h_1(k_1)x(n-k_1) + \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{N-1} h_2(k_1,k_2)x(n-k_1)x(n-k_2) + \cdots + \sum_{k_1=0}^{N-1} \cdots \sum_{k_p=0}^{N-1} h_p(k_1, \cdots, k_p)x(n-k_1) \cdots x(n-k_p) + \cdots
\]

where \(x(n)\) is the input signal, \(y(n)\) the output signals, and \(h_p(k_1, \cdots, k_p)\) the \(p\)-th order discrete Volterra kernel having generally a symmetry. Therefore, this series is invariance independently of the order of its terms, without loosing generality. Referring to Eq. (1), constant \(h_0\) is an offset term (DC component), \(h_1(k_1)\) is a linear impulse having a finite length, and \(h_p(k_1, \cdots, k_p)\) the \(p\)-th order impulse response which characterizes the nonlinearity of the system.

By introducing the \(p\)-th order Volterra operator \(H_p[x(n)]\), Eq. (1) can be simplified as

\[
y(n) = h_0 + \sum_{p=1} H_p[x(n)]
\]

where
\[ H_p[x(n)] = \sum_{k_1=0}^{N-1} \sum_{k_p=0}^{N-1} h_p(k_1, \ldots, k_p) \times x(n-k_1) \ldots x(n-k_p) \]  \hspace{1cm} (3)

Since this paper treats the second-order nonlinear components \((p=2)\), and assumes that all the Volterra kernels have a finite memory length, Eq. (1) is rewritten as

\[ y(n) = \sum_{k_1=0}^{N-1} h_1(k_1)x(n-k_1) \]
\[ + \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{N-1} h_2(k_1, k_2)x(n-k_1)x(n-k_2) \]  \hspace{1cm} (4)

3. The Design Method of Nonlinear Inverse System

Fig. 1 shows the tandem connection of the nonlinear inverse system and unknown nonlinear system.

Firstly, the design method of \(H_1\) in Fig. 1 is explained. It is necessary to design \(H_1\) so that this eliminates the linear distortion of an unknown nonlinear system. In other words, it is necessary to determine \(H_1\) so that the first order Volterra operator \(Q_1\) of the whole system satisfies

\[ Q_1[x(n)] = x(n) \]  \hspace{1cm} (5)

Therefore, \(H_1\) can be designed to realize \(H_1 = D_1^{-1}\) from Eq. (5) by the conventional inverse modeling [12].

The design method of \(H_2\) is described continuously. It is necessary to design \(H_2\) so that this eliminates the second order nonlinear distortion of the unknown nonlinear system. In other words, it is necessary to determine \(H_2\) so that the second order Volterra operator \(Q_2\) satisfies

\[ Q_2[x(n)] = 0 \]  \hspace{1cm} (6)

To obtain the design method of \(H_2\) which satisfies Eq. (6), Fig. 2 shows the block diagram of the second-order Volterra operator \(Q_2\) of the whole system.

Referring to Fig. 2, it is necessary to design \(H_2\) so that \(z_2(n)\) becomes zero; i.e. \(d(n)\) and \(o(n)\) cancel each other. To find what form of \(H_2\) realizes Eq. (6), Fig. 2 is modified: To erase \(D_1\) in Fig. 2, relationship \(D_1H_2 = z^a\) is used. Fig. 3 is obtained by inserting \(H_1\) after \(D_1\) and \(D_2\) in Fig. 2. Referring to Fig. 3, if

\[ H_2 \cdot z^a = -H_1 \cdot D_2 \cdot H_1 \]  \hspace{1cm} (7)

holds, the second-order Volterra operator of the whole system satisfies Eq. (6). Therefore, Fig. 1 is replaced by Fig. 4, by putting \(z^a\) after \(H_1\) and \(H_2\) in Fig. 1, and applying Eq. (7). Fig. 4 shows the final diagram. Note, a single \(H_1\) is used in Fig. 4, since \(H_1\) is commonly used for the first- and second-order of the nonlinear inverse system. To make a nonlinear inverse system using the structure shown in Fig. 4, a linear inverse filter \(H_1\) and an unknown linear system \(D_2\) are needed.

The procedure of construction of the system shown in Fig. 4 is as follows:

[Process 1] \(D_1\) and \(D_2\) are identified by the adaptive Volterra filters;

[Process 2] A first-order (linear) inverse system \(H_1\) for identified \(D_1\) is designed by using the conventional inverse modeling.

\[ x(n) \rightarrow H_1 \rightarrow y(n) \rightarrow D_1 \rightarrow o(n) \]  \hspace{1cm} (8)

Fig. 2 The block diagram of the second order Volterra operator \(Q_2\) of the system in Fig. 1.

\[ x(n) \rightarrow H_1 \rightarrow y(n) \rightarrow D_2 \rightarrow d(n) \]  \hspace{1cm} (9)

\[ H_2 \rightarrow y(n) \rightarrow D_1 \rightarrow o(n) \]  \hspace{1cm} (10)

\[ H_1 \rightarrow d(n) \rightarrow H_2 \rightarrow z(n) \]  \hspace{1cm} (11)

Fig. 3 The block diagram of the second order volterra operator \(Q_2\) given by modifying the block diagram in Fig. 2.

\[ x(n) \rightarrow H_1 \rightarrow y(n) \rightarrow H_2 \rightarrow z^a \]  \hspace{1cm} (12)

\[ D_1 \rightarrow o(n) \rightarrow D_2 \rightarrow z(n) \]  \hspace{1cm} (13)

\[ H_1 \rightarrow d(n) \rightarrow H_2 \rightarrow z(n) \]  \hspace{1cm} (14)

\[ z(n) \rightarrow H_1 \rightarrow y(n) \rightarrow D_2 \rightarrow z^a \]  \hspace{1cm} (15)

\[ D_1 \rightarrow o(n) \rightarrow D_2 \rightarrow z(n) \]  \hspace{1cm} (16)

\[ H_1 \rightarrow d(n) \rightarrow H_2 \rightarrow z(n) \]  \hspace{1cm} (17)

\[ z(n) \rightarrow H_1 \rightarrow y(n) \rightarrow D_2 \rightarrow z^a \]  \hspace{1cm} (18)

\[ D_1 \rightarrow o(n) \rightarrow D_2 \rightarrow z(n) \]  \hspace{1cm} (19)

\[ H_1 \rightarrow d(n) \rightarrow H_2 \rightarrow z(n) \]  \hspace{1cm} (20)

\[ z(n) \rightarrow H_1 \rightarrow y(n) \rightarrow D_2 \rightarrow z^a \]  \hspace{1cm} (21)

\[ D_1 \rightarrow o(n) \rightarrow D_2 \rightarrow z(n) \]  \hspace{1cm} (22)

\[ H_1 \rightarrow d(n) \rightarrow H_2 \rightarrow z(n) \]  \hspace{1cm} (23)

\[ z(n) \rightarrow H_1 \rightarrow y(n) \rightarrow D_2 \rightarrow z^a \]  \hspace{1cm} (24)

\[ D_1 \rightarrow o(n) \rightarrow D_2 \rightarrow z(n) \]  \hspace{1cm} (25)

\[ H_1 \rightarrow d(n) \rightarrow H_2 \rightarrow z(n) \]  \hspace{1cm} (26)

\[ z(n) \rightarrow H_1 \rightarrow y(n) \rightarrow D_2 \rightarrow z^a \]  \hspace{1cm} (27)

\[ D_1 \rightarrow o(n) \rightarrow D_2 \rightarrow z(n) \]  \hspace{1cm} (28)

\[ H_1 \rightarrow d(n) \rightarrow H_2 \rightarrow z(n) \]  \hspace{1cm} (29)

\[ z(n) \rightarrow H_1 \rightarrow y(n) \rightarrow D_2 \rightarrow z^a \]  \hspace{1cm} (30)

\[ D_1 \rightarrow o(n) \rightarrow D_2 \rightarrow z(n) \]  \hspace{1cm} (31)

\[ H_1 \rightarrow d(n) \rightarrow H_2 \rightarrow z(n) \]  \hspace{1cm} (32)

\[ z(n) \rightarrow H_1 \rightarrow y(n) \rightarrow D_2 \rightarrow z^a \]  \hspace{1cm} (33)

\[ D_1 \rightarrow o(n) \rightarrow D_2 \rightarrow z(n) \]  \hspace{1cm} (34)

\[ H_1 \rightarrow d(n) \rightarrow H_2 \rightarrow z(n) \]  \hspace{1cm} (35)

\[ z(n) \rightarrow H_1 \rightarrow y(n) \rightarrow D_2 \rightarrow z^a \]  \hspace{1cm} (36)

\[ D_1 \rightarrow o(n) \rightarrow D_2 \rightarrow z(n) \]  \hspace{1cm} (37)

\[ H_1 \rightarrow d(n) \rightarrow H_2 \rightarrow z(n) \]  \hspace{1cm} (38)

\[ z(n) \rightarrow H_1 \rightarrow y(n) \rightarrow D_2 \rightarrow z^a \]  \hspace{1cm} (39)

\[ D_1 \rightarrow o(n) \rightarrow D_2 \rightarrow z(n) \]  \hspace{1cm} (40)

\[ H_1 \rightarrow d(n) \rightarrow H_2 \rightarrow z(n) \]  \hspace{1cm} (41)

\[ z(n) \rightarrow H_1 \rightarrow y(n) \rightarrow D_2 \rightarrow z^a \]  \hspace{1cm} (42)

\[ D_1 \rightarrow o(n) \rightarrow D_2 \rightarrow z(n) \]  \hspace{1cm} (43)

\[ H_1 \rightarrow d(n) \rightarrow H_2 \rightarrow z(n) \]  \hspace{1cm} (44)

\[ z(n) \rightarrow H_1 \rightarrow y(n) \rightarrow D_2 \rightarrow z^a \]  \hspace{1cm} (45)

\[ D_1 \rightarrow o(n) \rightarrow D_2 \rightarrow z(n) \]  \hspace{1cm} (46)

\[ H_1 \rightarrow d(n) \rightarrow H_2 \rightarrow z(n) \]  \hspace{1cm} (47)

\[ z(n) \rightarrow H_1 \rightarrow y(n) \rightarrow D_2 \rightarrow z^a \]  \hspace{1cm} (48)

\[ D_1 \rightarrow o(n) \rightarrow D_2 \rightarrow z(n) \]  \hspace{1cm} (49)

\[ H_1 \rightarrow d(n) \rightarrow H_2 \rightarrow z(n) \]  \hspace{1cm} (50)

\[ z(n) \rightarrow H_1 \rightarrow y(n) \rightarrow D_2 \rightarrow z^a \]  \hspace{1cm} (51)

\[ D_1 \rightarrow o(n) \rightarrow D_2 \rightarrow z(n) \]  \hspace{1cm} (52)

\[ H_1 \rightarrow d(n) \rightarrow H_2 \rightarrow z(n) \]  \hspace{1cm} (53)
4. The Summational Affine Projection Algorithm

We show the summational affine projection algorithm below in the case of the system shown in Fig. 5.

\[ h(n+1) = h(n) + \mu A(n) h(n) \]
\[ A(n) = [a(1) a(2) \ldots a(i) \ldots a(p)] \]
\[ a(i) = \sum_{k=1}^{i} (1 - \mu/L)^{i-k} e(k-i+1) \cdot u(i) \]
\[ e(k-i+1) = d(k-i+1) - y(k-i+1) + v(k-i+1) \]
\[ \mathbf{u}(i) = \sum_{j=1}^{i} b_{ij}(k) \cdot \mathbf{x}(k-j+1) \]
\[ \mathbf{X}(k) = [x(k) \ x(k-1) \ \ldots \ x(k-p+1)] \]
\[ \mathbf{x}(k) = [x(k) \ x(k-1) \ \ldots \ x(k-N_1+1)] \]
\[ \mathbf{h}(n) = [h_1(1) \ h_2(2) \ \ldots \ h_1(N_1)] \]
\[ \mathbf{s}(n) = [1/s(1) \ 1/s(2) \ \ldots \ 1/s(i) \ \ldots \ 1/s(p)] \]
\[ \mu : \text{Step size parameter.} \]

5. The Control of Block Length

We should apply the control of block length to the previous algorithm in order to realize the convergence properties of high speed and high accuracy under high additive noise. To control the block length, we use the convergence parameter defined as the following equation to the proposed method.

\[ R_i(n) = \|a(i)/s(i)\|^2 \quad i = 1,2,\ldots,p \]

This convergence parameter decreases continuously until the convergence property of the adaptive Volterra filter reach the saturation condition and vibrates continuously in the saturation condition (refer to Fig. 6). Consequently, we can realize the convergence property of high speed and high accuracy under high additive noise by adding the following procedures to the summational affine projection algorithm.

1) If \( R(n) \leq R(n-1) \), the filter coefficients are updated at the current block length.
2) If \( R(n) > R(n-1) \), the block length is extended continuously until \( R(n) \leq R(n-1) \).

6. Examples and Results

The results of experiment that demonstrate the good properties of the summational affine projection algorithm are presented in this section. Table 1 shows the experiment condition. We compare the performance of the summational affine projection algorithm, NLMS algorithm, and the conventional affine projection algorithm. Fig. 7 and Fig. 8 show the convergence properties of the first and second order adaptive Volterra filters obtained using the above algorithms respectively. In Fig. 7 and Fig. 8, (a), (b), and (c) are the convergence properties of the proposed algorithm (\( p=3 \)), the conventional affine projection algorithm.
rithm \((p=3)\), and the NLMS algorithm respectively. The step size parameter \(\mu\) for the proposed algorithm is chosen as 8.0. The step size parameter \(\mu\) for the other algorithms is chosen as \(8/1024\). The initial block length \(L_0=256\) and the extension length of block length \(L_B=128\).

It can be seen from Fig. 7 and Fig. 8 that the proposed algorithm developed in this paper converges significantly faster than the NLMS Algorithm and the affine projection algorithm.

![Fig. 7](image1.png)

**Fig. 7** Convergence properties of the first order adaptive Volterra filter by the proposed method and by the conventional methods.

![Fig. 8](image2.png)

**Fig. 8** Convergence properties of the second order adaptive Volterra filter by the proposed method and by the conventional methods.

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Experiment condition.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tap length of unknown system (first order)</td>
<td>32</td>
</tr>
<tr>
<td>Tap length of unknown system (second order)</td>
<td>16</td>
</tr>
<tr>
<td>Tap length of the first order AVF</td>
<td>32</td>
</tr>
<tr>
<td>Tap length of the first order AVF</td>
<td>16</td>
</tr>
<tr>
<td>S/N between additive noise and desired signal</td>
<td>40dB</td>
</tr>
<tr>
<td>Input signal</td>
<td>White signal</td>
</tr>
</tbody>
</table>

Table 2 The condition of forward modeling for a nonlinear unknown system by the adaptive Volterra filter.

<table>
<thead>
<tr>
<th>Sampling Frequency</th>
<th>12kHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Signal</td>
<td>White Noise (5W)</td>
</tr>
<tr>
<td>Tap Length of (W_1)</td>
<td>256</td>
</tr>
<tr>
<td>Tap Length of (W_2)</td>
<td>128</td>
</tr>
<tr>
<td>Step Gain (\mu_1)</td>
<td>0.01</td>
</tr>
<tr>
<td>Step Gain (\mu_2)</td>
<td>0.01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 3</th>
<th>The design condition of a linear inverse system.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sampling frequency (F_s)</td>
<td>12 kHz</td>
</tr>
<tr>
<td>Tap length of (D_1)</td>
<td>256</td>
</tr>
<tr>
<td>Tap length of (H)</td>
<td>1024</td>
</tr>
<tr>
<td>Tap length of (\tilde{D}_1)</td>
<td>256</td>
</tr>
<tr>
<td>Inverse modeling delay (\Delta)</td>
<td>512</td>
</tr>
<tr>
<td>Step gain (\mu)</td>
<td>0.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 4</th>
<th>Comparison of second order nonlinear distortion levels between before and after elimination.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fundamental frequency (f_1)</td>
<td>93.75Hz</td>
</tr>
</tbody>
</table>

Next, let us design an actual nonlinear inverse system using the procedures described in the previous section. The condition of the modeling of an unknown nonlinear system in process 1 is shown in Table 2 and that of the design of a linear inverse system in process 2 is shown in Table 3. The modeling of the unknown nonlinear system was carried out using the input-output data of the loudspeaker recorded in a DAT on a computer. To estimate whether the nonlinear inverse system designed by the above procedures can eliminate the second order nonlinear distortion enough, we compare the level of the second order nonlinear distortion in the system of Fig.4 with that in the system where the linear inverse system is only connected to the unknown system, when various sinusoidal waves are applied to the two systems respectively. The results are shown in Table 4. It is clear from Table 4 that the second order nonlinear distortion is eliminated enough (the maximum elimination level is about 70dB).
7. CONCLUSION

This paper proposes a summation affine projection algorithm which has the convergence properties of high speed and high accuracy under high noise and colored input signal. And this algorithm is used to the design of a nonlinear inverse system which makes the nonlinear system linear. The proposed algorithm is shown to perform better than the other algorithm through experimental performance evaluation. We obtain an example of linearization with reduction of the secondary nonlinear distortion even by 70 dB.

REFERENCES