時間 영역에서의 非線形 離散系 식별

論 文
29—11—3

"Identification of Nonlinear Discrete Systems in the Time Domain"

崔 悰 鎬*
(Chong - Ho Choi)

Abstract

The problem of nonlinear time-invariant system identification by estimation of Wiener kernels is studied for discrete time systems with inputs having symmetric probability distributions. G-functionals are constructed in the time domain, and two distinct methods for identification are presented. It is further shown that under idealized conditions, these seemingly different techniques yield the same results. The results of identification of asimulated second degree system is presented.

I. Introduction

Volterra [16] represented the input-output relation for a class of nonlinear time invariant systems as

$$y(t)=k_0+\int_{-\infty}^{\infty}k_1(u)x(t-u)du+\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}k_2(u_1,u_2)x(t-u_1)x(t-u_2)du_1du_2+\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}k_3(u_1,u_2,u_3)x(t-u_1)x(t-u_2)\cdot x(t-u_3)du_1du_2du_3+\cdots \quad (1.1)$$

which is known as a Volterra series. We will call $k_n(u_1,u_2,\ldots,u_n)$ the "nth order Volterra kernel" when the u_i's are independent variables, and if $k_{n+m}(u,\ldots,u_{n+m})=0$ for $m>0$, we will call the system an "nth degree nonlinear system" (or simply an "nth degree system").

Wiener [7, 17] reformulated the Volterra series representation into sums of multi-dimensional convolutions constructed in the following way. Define the nth order multi-dimensional convolution with kernal $h_{n,i}, i\leq n$

$$W_{n,i}[x(t)]=\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}h_{n,i}(u_1,\ldots,u_n)\cdot x(t-u_1)\ldots x(t-u_i)du_1\ldots du_n \quad (1.2)$$

where $x(t)$ is a zero-mean white Gaussian process. Let $G_n[h_n,x(t)]$ be a linear combination of these convolutions

$$G_n[h_n,x(t)]=\sum_{i=0}^{n}a_{n,i}W_{n,i}$$

subject to the condition that

$$\overline{G_n[h_m,x(t)]G_n[h_n,x(t)]}=0 \text{ if } m\neq n, \quad (1.3)$$

where "$\overline{\quad}$" indicates the time average over the interval $(-\infty,\infty)$. The functional G_n constructed in this way is called the "nth degree G-functional," and it is orthogonal to all functionals of degree less than n. The kernel h_n is called the "nth order Winer kernel."

The first four G-functionals are given by

$$G_0[h_0,x(t)]=h_0$$

$$G_1[h_1,x(t)]=\int_{-\infty}^{\infty}h_1(u)x(t-u)du$$

$$G_2[h_2,x(t)]=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}h_2(u_1,u_2)x(t-u_1)\cdot x(t-u_2)du_1du_2-A\int_{-\infty}^{\infty}h_2(u,u)du$$

$$G_3[h_3,x(t)]=\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}h_3(u_1,u_2,u_3)x(t-u_1)x(t-u_2)x(t-u_3)du_1du_2du_3-3A\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}h_3(u_1,u_2,u_2)\cdot x(t-u_1)du_1du_2. \quad (1.4)$$

* 正 會 員 : 서울大 工大 計測制御工學科 助敎授
接受日字 : 1980年 9月 12

where the power spectrum density of the zero mean white Gaussian process $x(t)$ is $\phi XX(f)=A$. Using the G-functionals, the output $y(t)$ can be represented as

$$y(t)=\sum_{n=0}^{\infty} G_n[h_n,x(t)]. \tag{1.5}$$

Generally the kernels h_n in Eq. (1.5) are different from the k_n in Eq. (1.1). However both sets are assumed symmetric to guarantee uniqueness.

Due to the orthogonality property of the G-functionals, the Wiener series can be truncated after n G-functionals, giving the best nth order polynomial approximation to the system response in the sense of the least integral square error if the input $x(t)$ is a sample function of a zero mean white Gaussian process. Higher order G-functionals are independent and can be added later without affecting the estimate of G-functionals already found. Thus for systems representable via Volterra series, identification becomes the problem of determining the Wiener kernels.

Several methods have been presented to find Wiener kernels of a continuous causal system from the given input and output pairs. Wiener [7] suggested expanding the nth order kernel $h_n(u_1,\ldots,u_n)$ as a sum of Laguerre functions $l_m(u)$,

$$h_n(u_1,\ldots,u_n)=\sum_{m_1=0}^{\infty}\cdots\sum_{m_n=0}^{\infty} c_{m_1,\ldots,m_n}\, l_{m_1}(u_1)\ldots l_{m_n}(u_n)$$

and using phase shift networks to determine the coefficients $c_{m_1,\ldots,m_n}$ This method requires excessive computation and is not widely used. Later Lee and Schetzen [8] introduced a method using cross correlation techniques to compute Wiener kernels. Their formulae for the kernals are

$$\begin{aligned}
h_0&=\overline{y(t)}\\
h_1(u)&=\frac{1}{A}\overline{y(t)x(t-u)}\\
h_2(u_1,u_2)&=\frac{1}{2A^2}\overline{y(t)x(t-u_1)x(t-u_2)}\\
&\qquad \text{where } u_1\neq u_2 \qquad (1.6)\\
h_n(u_1,\ldots,u_n)&=\frac{1}{n!A^n}\overline{y(t)x(t-u_1)\ldots x(t-u_n)}\\
&\qquad \text{where } u_i\neq u_j \text{ for } i\neq j
\end{aligned}$$

where A is the power spectrum density of the white Gaussian input process $x(t)$ and $y(t)$ is the output of the system excited by $x(t)$. Hower estimates of $h_n(u_1,\ldots,u_n)$ for $u_i=u_j$ $(i\neq j)$ are very sensitive by this method.

Krauz [6] was interested in identifying biological systems that are excited by random impulse trains. He used a Poisson process as the input $x(t)$ and built a set of orthogonal functionals $\{G_n[h_n,x(t)]\}$, and a method of identifying Wiener kernels h_n. Krauz obtained essentially the same formulae as Eq. (1.6) except that the power spectrum density A is changed to the mean arrival rate in the Poisson process.

French and Butz [3] proposed calculating Wiener kernels in the frequency domain using FFT algorithms to reduce the comutaptional effort, and later [4] suggested Walsh functions as a means of computing the kernels. They used a white Gaussian input process, but the derivation of their formulae [3] involved some difficulties.

Marmarelis [12, 13] suggested using a "constant-switching-pace symmetric random signal," which changes its amplitude every constant time interval, for identification in the time domain. This process has finite power, but when used as an input to continuous time systems, leads to "approximately orthogonal" functionals instead of the orthogonal G-functionals. It should be clear that the kernels, which we called Wiener kernels when the input is a zero-mean white Gaussian process, will change form, depending upon the particular input.

Palm and Poggio [14] and Yasui [18] have written highly mathematical treatises exploring the connection between the Wiener and Volterra nonlinear system representations, and many papers [2, 5, 6, 8, 9, 10, 11, 13] dealt with the actual computation of the Wiener kernels (mostly up to second order).

In this paper we consider the construction of G-functionals for nonlinear discrete systems and then demonstrate two methods for identification of Wiener kernels in the time domain. Section III reports the results of identifying a simulated second degree system with adjustable parameters. For more detailed presentatron, see Choi [1].

II. Identification in the Time Domain

A. G-functionals for Discrete Systems

In this section, G-functionals for discrete systems will be constructed using a Gram-Schmidt type orthogonalization procedure, similar to Wiener's approach for continuous systems. Let T be a unit time interval and $x(iT)$ and $y(iT)$ be the input and output of the system at time iT. We will only consider causal systems with finite memory NT. The resulting restrictions, on the kernels, $k_n(u_1,\dots,u_n)$ of such systems are

$k_n(u_1,\dots,u_n)=0$ if any $u_j<0$,

and $k_n(u_1,\dots,u_n)=0$ if any $u_j\geq NT$.

The corresponding Volterra series representation (Eq. (1.1)) for a discrete system is

$$y(iT)=k_0+\sum_{j=i-N+1}^{i} k_1[(i-j)T]x(jT)+\sum_{j_1=i-N+1}^{i}\sum_{j_2=i-N+1}^{i} k_2[(i-j_1)T,(i-j_2)T]\cdot x(j_1T)x(j_2T)+\cdots. \tag{2.1}$$

To avoid an unnecessary proliteration of symbols, when it is obvious from the context, we will eliminate T in the arguments of x, y, k_n, etc. Also when the bounds of the summation are clear from the context, we will omit these as well.

There are an uncountable number of different ways in representing k_n ($n\geq 2$) in the sense that all the different representations give the same input-output relations. When k_n is not symmetrical with respect to its arguments, Wiener [17] suggested that one can always make it symmetrical by simply taking all the permutations of its arguments, adding them and dividing by the number of permutations. From now on the kernels k_n (or h_n) are assumed to be symmetrical.

We mentioned in the previous section that Wiener kernels change as the input changes from a white Gaussian process to a Poisson process. Hence describing the characteristics of the input precisely before finding the G-functionals is essential. We assume that the input $x(iT)$, $i=\cdots, -1,0,1,2,\cdots$ has the following properties:

1) Random variables $x(iT)$'s are mutually independent;

2) Random variables $x(iT)$'s have identical p.d.f.;

3) The p.d.f. $fX(.)$ of $x(iT)$ is an even function;

4) $E[x^n(iT)]$ is bounded for all n.

The first two assumptions allow us to conclude that the process is ergodic and hence time averages and ensemble averages are the same. Assumption 3) allows us to solve for the form of the discrete time G-functionals in a straightforward manner, while the last assumption guarantees that our procedure will remain well-posed.

As we denoted the time average with "——", we denote the ensemble average with "〈 〉", i.e., $\langle x\rangle=E[x]$. Also we denote the variance and nth ($n>2$) moments of $x(i)$ by A and $c_nA^{n/2}$.

$$A=\langle x^2(i)\rangle$$
$$c_nA^{n/2}=\langle x^n(i)\rangle \quad n=2,4,6,\dots \tag{2.2}$$

From these relationships we can always compute A and c_n if we know the p.d.f. $fX(\cdot)$. Note that a Gaussian distribution function is one of the p.d.f.'s which satisfy third assumption on $x(i)$.

The assumptions on the input process yield the following lemmas which we state without proof.

Lemma 1 : $\langle x(i_1)x(i_2)\dots x(i_k)\dots x(i_m)\rangle=0$ if there exists i_k such that $i_k\neq i_j$ for all $j\neq k$, or an odd number of arguments are equal.

Lemma 2 : The set D generated by(i_1,i_2,i_3,i_4) such that $\langle x(i_1)x(i_2)x(i_3)x(i_4)\rangle\neq 0$ is $D=A_1\cap A_2\cap A_3\cap B$ where

$$A_1=\{(i_1,i_2,i_3,i_4)\,|\,i_1=i_2,i_3=i_4,i_1\neq i_3\}$$
$$A_2=\{(i_1,i_2,i_3,i_4)\,|\,i_1=i_3,i_2=i_4,i_1\neq i_2\}$$
$$A_3=\{(i_1,i_2,i_3,i_4)\,|\,i_1=i_4,i_2=i_3,i_1\neq i_2\}$$
$$B=\{(i_1,i_2,i_3,i_4)\,|\,i_1=i_2=i_3=i_4\}.$$

Lemma 3 : $\langle x(i_1)x(i_2)\rangle=A\delta_{i_1,i_2}$ where δ_{i_1,i_2} is a kronecker delta ($\delta_{i_1,i_2}=1$ for $i_1=i_2$, zero otherwise).

Lemma 4 : $\langle x(i_1)x(i_2)x(i_3)x(i_4)\rangle=A^2\{\delta_{i_1,i_2}\,\delta_{i_3,i_4}\cdot(1-\delta_{i_1,i_3})+\delta_{i_1,i_3}\,\delta_{i_2,i_4}(1-\delta_{i_1,i_2})+\delta_{i_1,i_4}\,\delta_{i_2,i_3}(1-\delta_{i_1,i_2})+c_4\delta_{i_1,i_2}\,\delta_{i_1,i_3}\,\delta_{i_1,i_4}\}$.

Let us now build G-functionals consistent with the orthogonality requirement using a Grahm-Schmidt orthogonalization process. For notational convenience we will use $y_n(i)$ to denote $G_n[h_n, x(i)]$, as it might be considered the output at

time iT arising from the nth degree G-functional system. With this notation Eq. (1.5) can be written as

$$y(i)=\sum_{n=0}^{\infty} y_n(i). \tag{2.3}$$

Using the ergodicity of the input process, the orthogonality requirement of G-functionals in Eq. (1.3) can be replaced for discrete systems by

$$\langle G_m[h_m,x(i)]\ G_n[h_n,x(i)]\rangle=0 \text{ if } m\neq n. \tag{2.4}$$

From this requirement and with the previous lemmas, the first few G-functionals for the discrete systems are found to be

$$y_0(i)=h_0 \tag{2.5}$$

$$y_1(i)=\sum_j h_1(i-j)x(j) \tag{2.6}$$

$$y_2(i)=\sum_{j_1,j_2} h_2(i-j_1,i-j_2)x(j_1)x(j_2)-A\sum_{j=0}^{N-1} h_2(j,j) \tag{2.7}$$

$$\begin{aligned} y_3(i)=&\sum_{j_1,j_2,j_3} h_3(i-j_1,i-j_2,i-j_3)x(j_1)x(j_2)x(j_3)\\ &-3A\sum_{\substack{j_1,j_2\\ j_1\neq j_2}} h_3(i-j_1,i-j_2,i-j_2)x(j_1)\\ &-c_4A\sum_{j_1} h_3(i-j_1,i-j_1,i-j_1)x(j_1). \end{aligned} \tag{2.8}$$

We realize that $y_0(i), y_1(i)$ and $y_2(i)$ in discrete systems have basically the same form as the corresponding G-functionals in continuous systems, with the only difference being the replacement of integrals by summations. But $y_3(i)$ in Eq. (2.8) does not have the same form as G_3 in continuous systems. When $x(i)$ has a Gaussian distribution, $c_4=3$ and Eq. (2.8) becomes

$$\begin{aligned} y_3(i)=&\sum_{j_1,j_2,j_3} h_3(i-j_1,i-j_2,i-j_3)x(j_1)x(j_2)x(j_3)\\ &-3A\sum_{j_1,j_2} h_3(i-j_1,i-j_2,i-j_2)x(j_1), \end{aligned} \tag{2.9}$$

which is the same form as G_3 in continuous systems. When $x(i)$ does not have a Gaussian distribution, generally $c_4\neq 3$ and $y_3(i)$ does not have the same form as Eq. (2.9). In general we cannot get G-functionals for discrete systems simply by changing the integration signs in G-functionals given in Eq. (1.4)

We can continue to build higher degree G-functionals in the same way, but the algebraic work increases rapidly. In the next section we will present a method of finding Wiener kernels in the time domain.

B. Cross Correlation Method of Computing Wiener Kernels (Method T1)

We will assume that the system is time invariant, causal, has a memory NT, and yields finite output for finite input. Writing the last three assumptions explicitly,

1) $h_n(i_1,\ldots,i_n)=0$ if $i_j<0$ for any j

2) $h_n(i_1,\ldots,i_n)=0$ if $i_j\geq N$ for any j

3) $|h_n(i_1,\ldots,i_n)|<M$ for any set of $(i_1,\ldots,i_n)$,

where M is some positive number.

By using a scheme such as pictured in Fig. (2.1), we can relate $y(i)z(i)$ to the Wiener kernels using the orthogonality properties of the G-functionals.

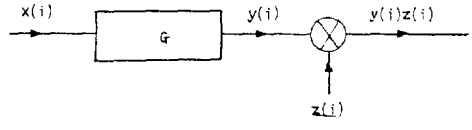

Fig. 2.1. Scheme for time domain identification.

Let $z(i)=i$ for all i, and note that $z(i)$ has the form of G_0, i.e. is a zero degree G-functional. If we take the ensemble average of $y(i)z(i)$ using Eq. (2.3) and the orthogonality of G-functionals, we have

$$\langle y(i)z(i)\rangle=\langle y_0(i)z(i)\rangle+\langle\sum_{n=1}^{\infty} y_n(i)z(i)\rangle=h_0.$$

Since $x(i)$ is a strictly stationary process, $y(i)$ is a strictly stationary process [15]. The ensemble average in the above equation can be replaced with a time average using the first two assumptions on $x(i)$, and we arrive at

$$h_0=\overline{y(i)}. \tag{2.10}$$

In order to find $h_1(i)$, multiply $y(i)$ in Fig.(2.1) by $z(i)=x(i-i_1)$ where $i_1(\geq 0)$ is fixed. Note that

$$z(i)=\sum_j \delta_{i-i_1,j}x(j)\ (i_1 \text{ is fixed}). \tag{2.11}$$

Comparing Eq. (2.11) with Eq. (2.6), we see that $z(i)$ has the form of G_1, i.e. is a first degree G-functional. Taking the ensemble average of $y(n)z(n)$ using Eqs. (2.3), (2.4), (2.11) and Lemma 1, yields

$$\langle y(i)z(i)\rangle=\sum_j h_1(i-j)\ \langle x(j)x(i-i_1)\rangle=Ah_1(i_1).$$

If we replace the ensemble average with time average as we did above, the previons equation becomes

$$h_1(i_1)=\frac{1}{A}\overline{y(i)x(i-i_1)} \quad i_1=0,1,\ldots,N-1. \quad (2.12)$$

In order to find $h_2(i_1,i_1)$, we multiply the output $y(i)$ in Fig. (2.1) by $z(i)=x(i-i_1)x(i-i_2)$, where i_1 (≥ 0) and $i_2(\geq 0)$ are fixed. We note that $z(i)$ can be written as

$$z(i)=\sum_{j_1,j_2}\delta_{i-i_1,j_1}\,\delta_{i-i_2,j_2}\,x(j_1)x(j_2) \quad (2.13)$$

and thus has the form of G_2 except for a constant term (see Eq. (2.7)). Following the same procedure as before

$$\langle y(i)z(i)\rangle=\langle y_0(i)z(i)\rangle+\langle y_1(i)z(i)\rangle$$
$$+\langle y_2(i)z(i)\rangle+\langle\sum_{n=3}^{\infty}y_n(i)z(i)\rangle. \quad (2.14)$$

We note that any nth degree functional $W(i)$ with kernel k_n can be expressed as a sum of orthogonal G-functionals of degree less than or equal to n, with the nth degree G-functional having kernel $h_n=k_n$ Lower order G-functionals in the sum will be chosen accordingly. Hence we can write

$$W(i)=\sum_{m=0}^{n}a_mG_m[h_m,x(i)].$$

As G_m is orthogonal to G_r for $r>m$, it follows that $W(i)$ is orthogonal to all G-functionals of degree greater than n. Since $z(i)$ is a second degree functional, the last term in Eq. (2.14) is zero.

After some algebraic manipulatons using the previous lemmas, Eq. (2.14) reduces to

$$\langle y(i)z(i)\rangle=\{Ah_0-A^2\sum_{j=0}^{N-1}h_2(j,j)\}\delta_{i_1,i_2}$$
$$+A^2\Big[\sum_{j_1=i-N+1}^{i}h_2(i-j_1,i-j_1)\delta_{i_1,i_2}\cdot$$
$$\{1+(c_4-1)\,\delta_{j_1,i-i_1}\}$$
$$+2h_2(i_1,i_2)(1-\delta_{i_1,i_2})\Big]. \quad (2.15)$$

Replacing the ensemble average with a time average when $i_1\neq i_2$, Eq. (2.15) becomes

$$h_2(i_1,i_2)=\frac{\overline{y(i)x(i-i_1)x(i-i_2)}}{2A^2} \quad i_1,i_2=0,1\cdots,N-1.$$
$$i_1\neq i_2 \quad (2.16)$$

When $i_1=i_2$, Eq. (2.15) becomes

$$\langle y(i)z(i)\rangle=\{Ah_0-A^2\sum_j h_2(j,j)\}$$
$$+A^2\{\sum_{j_1}h_2(i-j_1,i-j_1)$$
$$+(c_4-1)h_2(i_1,i_1)\}$$
$$=Ah_0+(c_4-1)A^2h_2(i_1,i_1).$$

Replacing the ensemble average with time average, the previous equation becomes

$$h_2(i_1,i_1)=\frac{\overline{y(i)x(i-i_1)x(i-i_1)}-Ah_0}{(c_4-1)A^2}$$
$$i_1=0,1,\ldots,N-1. \quad (2.17)$$

We do not have to worry about the case $c_4=1$ in Eq. (2.17), because $c_4>1$ except when $fX(r)=1/2\delta(r-\sqrt{A})+1/2\delta(r+\sqrt{A})$.

C. A General Method of Comupting Wiener Kernels

We now present another method of finding Wiener kernels by solving simultaneous linear equations and show that these solutions lead to Eqs. (2.12), (2.16), and (2.17) under an ideal situation.

Let us consider the case of the first order kernel. In the derivation of Eq. (2.12), we had

$$\langle y(i)x(i-i_1)\rangle=\sum_j h_1(j)\,\langle x(i-j)x(i-i_1)\rangle$$
$$+\langle\sum_{n+1}y_n(i)x(i-i)\rangle. \quad (2.18)$$

Let $\langle y(i)x(i-i_1)\rangle$, $\langle x(i-j)x(i-i_1)\rangle$, and $\langle\sum_{n+1}y_n(i)$ $x(i-i_1)\rangle$ be replaced by $r(i_1)$, $p(i_1-j)$, and $e(i_1)$, respectively. Rearranging Eq. (2.18), we have

$$\sum_{j=0}^{N-1}p(i-j)h_1(j)=r(i_1)-e(i_1) \quad i_1=0,1,\cdots,N-1.$$
(2.19)

Let $\hat{p}(i)$, $\hat{r}(i)$, and $\hat{e}(i)$ be estimates of $p(i)$, $r(i)$ and $e(i)$ respectively. In a real situatin we can deal with only a finite number of samples, so $\hat{p}(i)$, $\hat{r}(i)$, and $\hat{e}(i)$ are in general different from their corresponding theoretical values. We can compute $\hat{p}(i)$ from the input processes and $\hat{r}(i)$ can be computed from the input and output processes. Since we cannot compute $\hat{e}(i)$, we assume that $\hat{e}(i)=e(i)$, i.e., $\hat{e}(i)=0$. Then Eq. (2.19) becomes

$$\sum_{j=0}^{N-1}\hat{p}(i_1-j)h_1(j)=\hat{r}(i_1) \quad i_1=0,1,\cdots,N-1. \quad (2.20)$$

which is a set of N simultaneous linear equations with N unknowns.

In deriving this equation, we only assumed that $\hat{e}(i)$ is zero for all i, and we did not assume that

$\hat{p}(i)=A\delta_{i,0}$. Hence, it is natural to expect that the solution of Eq. (2.20) will be better than the one obtained by Eq. (2.12). If we assume that $\hat{p}(i)=p(i)$, then it is easy to see that the solution of Eq. (2.20) conforms to Eq. (2.12).

In the same way as above, consider the case of the second order kernel. Note that

$$\begin{aligned}\langle y(i)x(i-i_1)x(i-i_2)\rangle &= h_0\langle x(i-i_1)x(i-i_2)\rangle \\ &+\sum_{j_1,j_2} h(j_1,j_2)\ \langle x(i-j_1)x(i-j_2)\cdot \\ &x(i-i_1)x(i-i_2)\rangle - A\sum_j h_2(j,j)\cdot \\ &\langle x(i-i_1)x(i-i_2)\rangle + \sum_{n\neq 0,2} y_n(i)\cdot \\ &x(i-i_1)x(i-i_2)\rangle \qquad (2.21)\end{aligned}$$

Let $\langle y(i)x(i-i_1)x(i-i_2)\rangle$, $\langle x(i-i_1)x(i-i_2)\rangle$, $\langle x(i-j_1)x(i-j_2)x(i-i_1)x(i-i_2)\rangle$, and $\langle \sum_{n\neq 0,2} y_n(i)x(i-i_1)x(i-i_2)\rangle$ be $r(i_1,i_2)$, $s(i_1-i_2)$, $p(j_1,j_2,i_1,i_2)$ and $e(i_1,i_2)$, respectively, and each of these estimates with finite samples be $\hat{r}(i_1,i_2)$, $\hat{s}(i_1-i_2)$, $\hat{p}(j_1,j_2,i_1,i_2)$ and $\hat{e}(i_1,i_2)$, respectively. Then with finite samples, Eq. (2.21) becomes

$$\begin{aligned}&\sum_{i_1,i_2}\{\hat{p}(j_1,j_2,i_1,i_2)-A\hat{s}(i_1-i_2)\ \delta_{j_1,i_2}\}\ h_2(j_1,j_2) \\ &\quad=\hat{r}(i_1,i_2)-h_0\hat{s}(i_1-i_2)-\hat{e}(i_1,i_2) \\ &\qquad i_1=0,1,\cdots,N-1 \\ &\qquad i_2=0,1,\cdots,N-1. \qquad (2.22)\end{aligned}$$

The terms $\hat{p}(j_1,j_2,i_1,i_2)$, $\hat{s}(i_1,-i_2)$, and $\hat{r}(i_1,i_2)$, can be evaluated from the input and output processes, and we assume that $\hat{e}(i_1,i_2)=e(i_1,i_2)$ i.e., $\hat{e}(i_1,i_2)=0$, as we did before. If we substitute for h_0 the computed value $\hat{h}_0$ from Eq. (2.10), then Eq. (2.22) is a set of N^2 simultaneous linear equations with N^2 unknowns. The kernel $h_2(i_1,i_2)$ can be found by solving these equations. Replacing $\hat{p}(j_1,j_2,i_1,i_2)$ and $\hat{s}(i_1-i_2)$ with each of their theoretical values using Lemmas 1, 3 and 4 and then noting that $h_2(i_1,i_2)$ is symmetric with respect to its arguments Eq. (2.22) becomes Eq. (2.23).

$$A^2\begin{pmatrix} c_4-1 & & & & & & & & \\ & 1 & & & & & & & \\ & & 1 & & & & & & \\ & & & \ddots & & & & & \\ & & & & 1 & & & & \\ & & & & & c_4-1 & & & \\ & & & & & & 1 & & \\ & & & & & & & \ddots & \\ & & & & & & & & c_4-1 \end{pmatrix}\begin{pmatrix} h_2(0,0) \\ 2h_2(0,1) \\ 2h_2(0,2) \\ \vdots \\ 2h_2(0,N-1) \\ h_2(1,1) \\ 2h_2(1,2) \\ \vdots \\ h_2(N-1,N-1) \end{pmatrix}=\begin{pmatrix} \hat{r}(0,0)-\hat{h}_0 \\ \hat{r}(0,1) \\ \hat{r}(0,2) \\ \vdots \\ \hat{r}(0,N-1) \\ \hat{r}(1,1)-\hat{h}_0 \\ \hat{r}(1,2) \\ \vdots \\ \hat{r}(N-1,N-1)-\hat{h}_0 \end{pmatrix}. \qquad (2.23)$$

This method can be applied to finding the higher order kernels, but the computational effort increases geometrically.

III. Simulation Results

A simulation was performed to estimate the zero, first and second order kernels of a model using the method *T*1 presented in the previous section. It was hoped that the simulation might provide some insight into how errors are affected by the relative powers of G-functionals and how many data points should be used to get good estimates of kernels.

The model used as an unknown system to be identified is of second degree with the following Volterra kernels:

$$k_0=0$$

$$k_1(i)=s_1 e^{-2\pi i/16}\sin\left(\frac{2\pi i}{16}\right)\quad i=0,1,\cdots,15$$

$$k_2(i_1,i_2)=e^{-2\pi(i_1+i_2)/16}\sin\left(\frac{2\pi i_1}{16}\right)\sin\left(\frac{2\pi i_2}{16}\right)$$
$$i_1,i_2=0,1,\cdots,15.$$

Note that $k_2(i_1,i_2)=k_1(i_1)k_1(i_2)$ when $s_1=1$. We varied s_1 from 0 to 5 changing the relative power generated by each G-functional. In this model the system memory N is 16, T is 1.

The simulation was performed with 4800 data points. The input was generated by the IBM pseudorandom number generator RANDU (see IBM Scientific Subroutine Package), and these random numbers were shifted and scaled so that they should have been distributed uniformly on $(-\sqrt{3},\sqrt{3})$ with zero mean, unit variance and $c_4=1.8$. The sample mean of these sequences, however, was not zero, and it was necessary to subtract the sample mean from these sequences to achieve zero mean data for the simulation input.

We also studied the convergence of the kernel estimates as the number of data points increased.

Letting $s_1=3$ we generated three additional sets of 4800 point input and output sequences. We then repeatedly computed the kernels, adding one more data set to the data sets already used.

The simulation results were studied based on several different criteria. Let $v_1(i)$ and $v_2(i)$ be the output at time i due to the first and second order Volterra kernels respectively, and define

$$p_{11}=\frac{\sum_i\{v_1(i)\}^2}{\sum_i\{v_2(i)\}^2},$$

$$p_{12}=\frac{\sum_i\{v_1(i)v_2(i)\}}{\sum_i\{v_2(i)\}^2}.$$

p_{11} is the relative power generated by the first order Volterra kernel with respect to the power generated by the second order kernel. p_{12} is the average crossproduct of the outputs due to the first and second order Volterra kernels with respect to the power due to the second order kernel. p_{12} theoretically, should be zero. p_{11} is plotted in Fig. (3.1). Note that $v_1(i)$ is proportional to s_1, while $v_2(i)$ is independent of s_1. Hence, the parabolic nature of the p_{11} curve in Fig. (3.1) and the linear p_{12} curve in Fig. (3.2) is to be expected.

The error criterion used for the nth order Wiener kernel is

$$e_n=\left[\frac{\sum_{i_1,\ldots,i_n}\{h_n(i_1,\ldots,i_n)-\hat{h}_n(i_1,\ldots,i_n)\}^2}{\sum_{i_1,\ldots,i_n}\{h_n(i_1,\ldots,i_n)\}^2}\right]^{1/2} \quad (3.1)$$

Figure (3.3) shows that e_n increases with n for $n=0,1,2$. In general the identification of a kernel becomes more accurate as the relative power generated by that kernel increases.

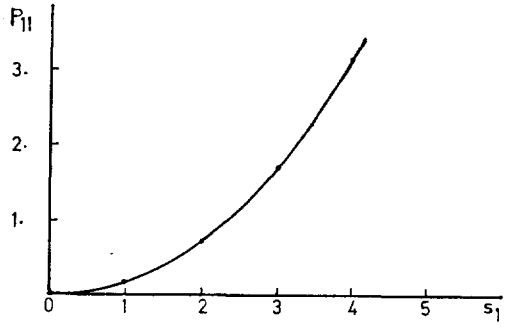

Fig. 3.1. Relative power generated by the first order kernel.

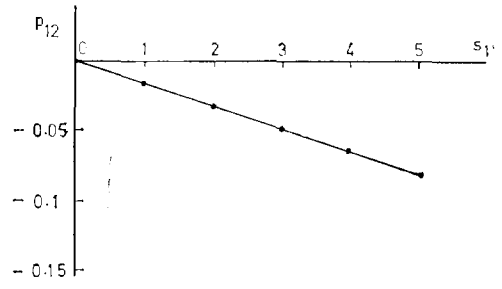

Fig. 3.2. Average crossproduct of the outputs due to the first and the second Volterra kernels normalized by the second order kernel.

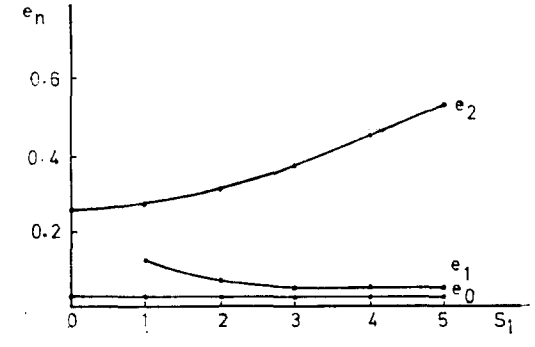

Fig. 3.3. Relative error of the Wiener kernel.

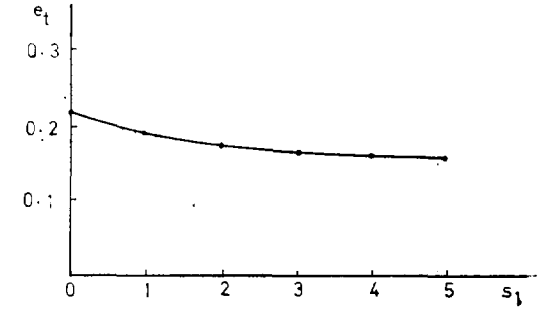

Fig. 3.4. Relative output error.

We used another error criterion, e_t, which measures the square of the output error due to the errors in the estimated kernels.

$$e_t=\left[\frac{\langle\{\sum_n G_n[h_n,x(i)]-\sum_n G_n[\hat{h}_n,x(i)]\}^2\rangle}{\langle\{\sum_n G_n[h_n,x(i)]\}^2\rangle}\right]^{1/2} \quad (3.2)$$

where $\hat{h}_n$ is the estimate of h_n. For the second order system under investigation, the numerator inside the square root in Eq. (3.2) is

$$(h_0-\hat{h}_0)^2+A\sum_{i}\{h_1(i)-\hat{h}_1(i)\}^2$$
$$+4A^2\sum_{i_1}\sum_{i_2}\{h_2(i_1,i_2)-\hat{h}_2(i_1,i_2)\}^2$$
$$+(c_4-1)\ A^2\sum_{i}\{h_2(i,i)-\hat{h}_2(i,i)\}^2.$$

As s_1 increases, the first order kernel generates more power and its estimate becomes more accurate while the estimate of h_2 becomes less accurate. From Fig. (3.4), as s_1 increases, e_t decreases.

Figures (3.5) and (3.6) show how errors in the kernels estimated decrease as the number of data points, increases. In real situations, we do not know the exact Wiener kernels; only estimates are known.

As more data areprocessed, new kernel estimates are obtained. We can compute the equivalent error of the $(l-1)$ th estimate of the Wiener kernel h_n by assuming that the lth estimate is the true Wiener kernel.

$$e_n(l)=\left[\frac{\sum_{n}\{\hat{h}^{l-1}(i_1,\ldots,i_n)-\hat{h}^{l}(i_1,\ldots,i_n)\}^2}{\sum_{n}\{\hat{h}^{l}(i_1,\ldots,i_n)\}^2}\right]^{1/2} \qquad (3.3)$$

Figure (3.7) shows the errors based on Eq. (3.3)

Finally we compute the relative output power P generated by the system G composed of kernels estimated with respect to the power gererated by the original system. Figure (3.8) shows that G generates output power which is very close to the

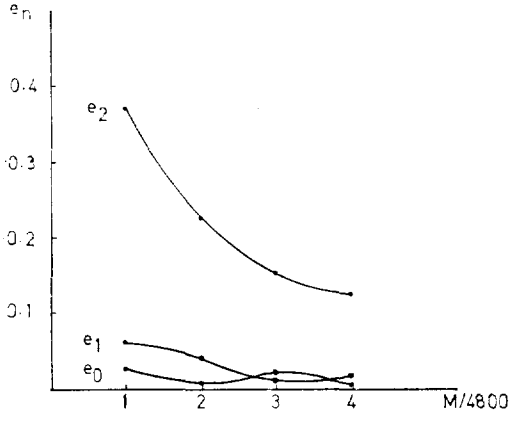

Fig. 3.5. Relative error of the Wiener kernel.

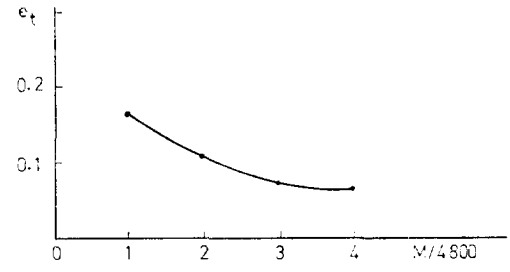

Fig. 3.6. Relative output error.

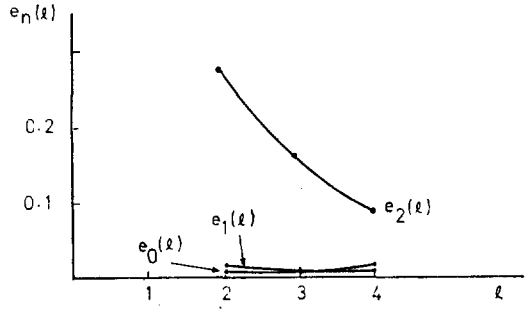

Fig. 3.7. Estimates of relative errors.

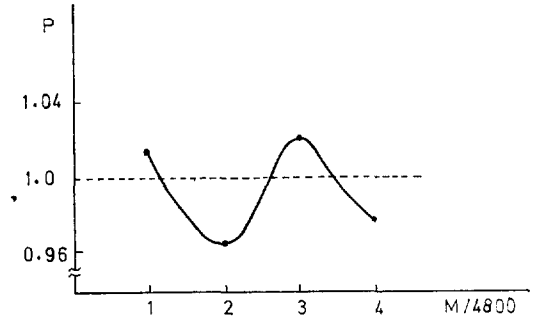

Fig. 3.8. Relative output power generated by the computed kernels.

output power generated by the real system even though the estimated kernels have big errors. Hence, P of the system G is not a good criterion to evaluate the estimates of the kernels.

IV. Conclusions

Two different methods have been presented for identification of nonlinear discrete systems in the time domain. It was shown that these seemingly different methods are the same under ideal conditions.

It is conceptually clear how to extend the identification techniques to higher order kernels. However, the extensions involve handling very complicated formulae, and the computational effort increases with the order of kernel.

A simple example was presented to give some insight into the actual identification problems. Up to 19,200 (4×4800) data points were used for identification of a second degree nonlinear system. Dividing the number of data points by the total number of unknown elements in the kernels yields more than 100 data points per unknown element.

Still the error in the second order kernel estimate is more than 10%. This attests to the difficulty of obtaining accurate estimates of even low order kernels in relatively simple systems. The errors of the estimates ars mainly due to the finite length of the data and the nonwhiteness of the pseudorandom numbers. The estimation of kernel errors remain an important research problem.

References

1. Choi, C.H.; Identification of Nonlinear Discrete Systems, Ph. D. Dissertation, University of Florida, Gainesville, FL 1978.
2. French, A.S.; "Practical nonlinear system analysis by Wiener kernel estimation in the frequency domain," preprint from Biological Cybernetics, Springer-Verlag, 1976.
3. French, A.S. and Butz, E.G.; "Measuring the Wiener kernels using the fast Fourier transform algorithm," Int. J. of Control, Vol. 17, No. 3, pp. 529~539, 1973.
4. French, A.S. and Butz, E.G.; "The use of Walsh functions in the Wiener analysis of nonlinear systems," IEEE Trans. Computers. Vol. C-23, No. 3, pp. 225~232, March 1974.
5. Harris, G.H. and Lapidus, L.; "The indentification of nonlnear systems," Industrial and Engineering Chemistry, Vol. 59, No. 6, pp. 66~81, June 1967.
6. Krauz, H.; "Identification of nonlinear systems using random impulse train inputs," Biological Cybernetics, 19, 217~230, 1975.
7. Lee, Y.W.; "Contributions of Norbert Wiener to linear theory and nonlinear theory in engineering," from Selected Papers of Norbert Wiener, S.I.A.M. and M.I.T. Press, Cambridge, pp. 17~33, 1964.
8. Lee, Y.W. and Schetzen, M.; "Measurement of Wiener kernels of a nonlinear system by cross-correlation," Int. J. Control, Vol. 2, pp. 237~254, 1965.
9. Marmarelis, P.Z.; "Nonlinear identification of bioneuronal systems through white noise simulation," Proc. of the Joint Automatic Control Conference, Stanford Univ., pp. 117~126, Aug. 1972.
10. Marmarelis, P.Z., and Naka, K.I.; "White noise analysis of a neuron chain; An application of Wiener theory," Science, Vol. 175, No. 4027, pp. 1276~1278, March 1972.
11. Marmarelis, P.Z., and Naka, K.I.; "Nonlinear analysisand synthesis of receptive field responses in the catfish retina," J. Neurophysiology, Vol. 37, pp. 869~895, 1974.
12. Marmarelis, P.Z.; "A family of quasi-white random signals and its optimal use in biological system identification, Part I: Theory," Biological Cybernetics, 27, pp. 49~56, 1977.
13. Marmarelis, P.Z., "A family quasi-white random signals and its optimal use in biological system identification, Part II: Application to the photoreceptor of calliphora erythrocephola," Biological Cybernetics, 27, pp. 57~62, 1977.
14. Palm, G., and Poggio, T., "Volterra representation Wiener expansions: Validity and pitfalls," SIAM J. Applied Math., Vol. 33, No. 2, pp. 195~216, Sept. 1977.
15. Parzen, E., Stochastic Process, Holden-Day, 1962.
16. Volterra, V., The Theory of Functionals and Integro-Differential Equations, Dover, 1959.
17. Wiener, N., Nonlinear Problems in Random Theory, M.I.T. Press, 1958.
18. Yasui, S., "Wiener and Volterra series, and nonlinear systems: Part I, Representation and identification problems" unpublished manuscript, Cal. Tech., Jan. 1976.