Second-Order Volterra Filtering and Its Application to Nonlinear System Identification

TAIHO KOH AND EDWARD J. POWERS

Abstract—Some recent results on the design and implementation of second-order Volterra filters are presented. The (second-order) Volterra filter is a nonlinear filter with the filter structure of (second-order) Volterra series. A simple minimum mean-square error solution for the Volterra filter is derived, based on the assumption that the filter input is Gaussian. Also, we propose an iterative factorization technique to design a subclass of the Volterra filters, which can alleviate the complexity of the filtering operations considerably. Furthermore, an adaptive algorithm for the Volterra filter is investigated along with its mean convergence and asymptotic excess mean-square error. Finally, the utility of the Volterra filter is demonstrated by utilizing it in studies of nonlinear drift oscillations of moored vessels subject to random sea waves.

I. INTRODUCTION

THE concept of optimum linear filtering has had enormous impact on the recent development of various techniques to estimate and process stationary time series. The obvious advantage of a linear filter is its simplicity in design and implementation. However, with the minimum mean-square error criterion, the ultimate solution to the optimum filter is in finding the conditional mean which is, in general, a nonlinear function of observed data. So there remains the unanswered question of how much one pays, in terms of filter performance, for the simplicity of a linear filter. In some cases, the performance of a linear filter may be unacceptable. A typical example involves the case when one tries to relate two signals whose significant spectral components do not overlap in the frequency domain. Another important factor in favor of nonlinear filters is the vast capability of modern computers which enables us to overcome the complexity of the nonlinear filtering problem.

One constructive and versatile approach to nonlinear filters is to utilize the filter structure in the form of a Volterra series. Wiener's work [1] on the analysis of nonlinear systems using white Gaussian input and so called G functionals is well known. Following his work, a number of papers [2]-[10] have been devoted to utilizing the Volterra series for estimation and nonlinear system identification. More recently [11]-[20], discrete-time filters in a similar form, which we call Volterra filters, have been studied with and without adaptive implementation. From a theoretical viewpoint, the Volterra filter is attractive since it can deal with a general class of nonlinear systems while its output is still linear with respect to various higher order system kernels or impulse responses.

However, in spite of its long history and popularity in theoretical studies, relatively few researchers [11], [15] have attempted to apply the Volterra filtering technique to practical problems. One major reason for this seems to be the formidable complexity associated with the design of Volterra filters. For example, many works [4], [11], [18], [20] utilize a linearization technique in which the Volterra filter is regarded as a linear filter with a multidimensional input signal. So the particular structure of Volterra filters is ignored and a huge matrix problem occurs. The number of operations required to solve the problem increases exponentially with the highest order term of the Volterra filter.

In this context, the primary concern of the present study is to seek simplifications in both the design and implementation of Volterra filters. In particular, we concentrate our discussions on the second-order Volterra filter. The second-order Volterra filter, which consists of a parallel combination of linear and quadratic filters, is a prototype nonlinear filter by which one can improve the performance of a linear filter, considerably in some cases, with a relatively mild computational effort. The organization of this paper is as follows. Section II gives a brief review on the general theory of Volterra filtering in terms of the generalized orthogonal projection principle. In Section III, we derive a simple solution for the optimum second-order Volterra filter, based on the assumption that the filter input is Gaussian. Section IV presents an iterative factorization technique to facilitate the implementational complexity of the Volterra filters, along with a numerical example. Also, Section V considers an extended version of the LMS adaptive algorithm for the Volterra filter, along with detailed analyses of its mean convergence and asymptotic excess mean-square error. Finally, Section VI is intended to demonstrate the utility of the Volterra filtering techniques in practical applications. Some examples of experiments in which the Volterra filter is utilized to model and predict the nonlinear dynamic behavior of offshore structures are presented.

Manuscript received July 23, 1984; revised April 22, 1985. This work was supported in part by the Department of Defense Joint Services Electronics Program through the Air Force Office of Scientific Research under Contract F49602-82-C-0033.

T. Koh was with the Department of Electrical Engineering and the Electronics Research Center, University of Texas, Austin, TX 78712. He is now with AT&T Bell Laboratories, Murray Hill, NJ 07974.

E. J. Powers is with the Department of Electrical Engineering and the Electronics Research Laboratory, University of Texas, Austin, TX 78712.

0096-3518/85/1200-1445$01.00 © 1985 IEEE
between the filter weights and input. Consequently, it is
possible and often helpful to analyze and interpret in the
frequency domain using the transform of its
spectral filter weight sequence with parameter set \(S_k\). Also, \(S_k\) denotes a region of summation in \(m_1, \ldots, m_k\), which is a bounded subset of \(\mathbb{Z}^k\), the \(k\)th product set of integers. Since (1) represents a discrete-time Volterra series truncated at the \(p\)th term, it will be called the \(p\)th-order Volterra filter. To avoid unnecessary multiplicities, we assume that \(h_k(m_1, \ldots, m_k)\) is symmetric in that its value is unchanged for any permutation of \(m_1, \ldots, m_k\) [for example, \(h_2(m_1, m_2) = h_2(m_2, m_1)\)].

Two important aspects of the Volterra filter are to be noted. First, it consists of multidimensional convolutions between the filter weights and input. Consequently, it is possible and often helpful to analyze and interpret in the frequency domain using \(z\) transforms. For example, if we let \(y_k(n)\) denote the \(k\)th term of (1), i.e.,

\[
y_k(n) = \sum_{S_k} h_k(m_1, \ldots, m_k) x(n - m_1) \cdots x(n - m_k),
\]

its \(z\) transform \(Y_k(z)\) can be expressed in terms of \(X(z)\), \(z\) transform of \(x(n)\), as

\[
Y_k(z) = F_k[H_k(z_1, \ldots, z_k) X(z_1) \cdots X(z_k)]
\]

where \(H_k(z_1, \ldots, z_k) = \sum_{S_k} h_k(m_1, \ldots, m_k) z_1^{m_1} \cdots z_k^{m_k}\)
and \(F_k\) is the operation by which a \(k\)-dimensional \(z\) transform is reduced to a one-dimensional \(z\) transform [19].

The operation can be interpreted as a frequency combiner and will be discussed later with some insights on the role of \(H_k(z_1, \ldots, z_k)\).

A second and more important aspect of (1) is the fact that the filter output is linear with respect to the Volterra filter weights. The linearity is desirable when one deals with the minimum mean-square error criterion. Suppose \(x(n)\) and \(s(n)\) are zero-mean stationary random processes with discrete parameter \(n\) and we want to find the Volterra filter weights which minimize the mean-square error (MSE) between \(s(n)\) and the filter output \(y(n)\). (The zero-mean assumption is only for convenience and can be easily removed.) Then, by virtue of the linearity of the Volterra filter, the MSE has only one global minimum and we can find it directly by using the calculus of variations. Alternatively, we can proceed by invoking the orthogonal projection principle [5] as follows. The residual error of a minimum MSE Volterra filter is orthogonal not only to the filter input, but also to all possible products of the input, i.e., \(x(n - m_1) \cdots x(n - m_k)\), \(k = 1, \ldots, p\). So if we let \(y_{\text{opt}}(n)\) denote the output of the minimum MSE filter, we have

\[
E[(s(n) - y_{\text{opt}}(n))] = 0
\]

and

\[
E[(s(n) - y_{\text{opt}}(n)) x(n - m_1) \cdots x(n - m_k)] = 0
\]

for all \((m_1, \ldots, m_k) \in S_k\) and \(k = 1, \ldots, p\). Consequently, the optimum filter weights could be found, in principle, from the following equations:

\[
E[y(n) \prod_{j=1}^q x(n - \sigma_j)]
\]

\[
= h_0 E\left[\prod_{j=1}^q x(n - \sigma_j)\right]
\]

\[
+ \sum_{k=1}^p \sum_{S_k} h_k(m_1, \ldots, m_k) \cdot E\left[\prod_{i=1}^k x(n - m_i) \prod_{j=1}^q x(n - \sigma_j)\right]
\]

for all \((\sigma_1, \ldots, \sigma_q) \in S_q\) and \(q = 1, \ldots, p\).

Now we can see clearly the amount of statistical knowledge and computation that are required to find the optimum Volterra filter weights. For a \(p\)th-order Volterra filter, \(2p\) autocorrelation and \(p\) cross-correlation functions should be known. In most practical cases, such higher order correlation functions are not known and could be estimated only with a large amount of computations and little accuracies. Furthermore, if \(S_k\) is chosen to be the \(k\)th product set of \(N\) consecutive integers, a system of

\[
\sum_{k=0}^p \binom{N + k - 1}{k}
\]
linear equations with the same number of unknowns should be solved in terms of these correlation functions. The task often becomes overwhelming, even for \(p = 2\).

III. SECOND-ORDER VOLterra filter

To examine the case of the second-order Volterra filter (SVF) closely, it will be convenient to reformulate the SVF as
y(n) = h_0 + \sum_{j=0}^{N-1} a(j) x(n - j) + \sum_{j,k=0}^{N-1} b(j, k) x(n - j) x(n - k) \tag{3}

where \{a(j)\} and \{b(j, k)\} are called the linear and quadratic filter weights, respectively, and \(N\) denotes the filter length. Recall that the quadratic filter weights are assumed to be symmetric, i.e., \(b(j, k) = b(k, j)\). As before, we want to minimize the MSE between the primary signal \(s(n)\) and filter output \(y(n)\), i.e.,

\[
\xi = E[(s(n) - y(n))^2]
\]

where both \(s(n)\) and \(x(n)\) are assumed to be strictly stationary with zero means.

The first step to determine the minimum MSE SVF is to require the unbiasedness of the filter output. In other words, we should have \(E[y(n)] = 0\) since the primary signal \(s(n)\) has zero mean. Then, we have the following relationship between \(h_0\) and \(b(j, k)\):

\[
h_0 = -\sum_{j,k=0}^{N-1} b(j, k) r_s(j - k) \tag{4}
\]

where \(r_s(j) = E[x(n) x(n - j)]\) denotes the autocorrelation function of \(x(n)\). Inclusion of the zeroth-order term \(h_0\) is important. Some of the previous works [2], [4], [13], [20] did not have the zeroth-order output. But without it, the output of a minimum MSE SVF is not necessarily unbiased, and hence tends to yield a larger error than the SVF with \(h_0\) given by (4). By combining (3) and (4), the SVF is expressed as

\[
y(n) = \sum_{j=0}^{N-1} a(j) x(n - j) + \sum_{j,k=0}^{N-1} b(j, k) \cdot [x(n - j) x(n - k) - r_s(j - k)] \tag{5}
\]

The next step is to determine the linear and quadratic filter weights which yield the minimum MSE. This can be done, in principle, by directly applying the orthogonal projection principle as described in Section II. Due to its complexity, we will not present the general solution here. For an explicit form of the solution, see [4]. However, we note that the general solution requires computing the inverse of an \(N(N + 3)/2\) by \(N(N + 3)/2\) matrix whose elements are given in terms of second-, third-, and fourth-order autocorrelation functions of \(x(n)\). The number of operations required to compute the inverse is of order \(N^5\). This computational requirement becomes almost prohibitive for large \(N\). Besides, a singularity problem may occur [20].

**Gaussian Case**

In the following, we derive a simple solution for the optimum SVF under the technical assumption that the filter input \(x(n)\) is Gaussian. First, it is noted that (5) can be rewritten as

\[
y(n) = A'X(n) + \tr \{B[X(n) X'(n) - R_s]\} \tag{6}
\]

where

\[
X(n) = [x(n), \ldots, x(n - N + 1)]' \\
A = [a(0), \ldots, a(N - 1)]' \\
B = \begin{bmatrix}
 b(0, 0) & \cdots & b(0, N - 1) \\
 \vdots & \ddots & \vdots \\
 b(N - 1, 0) & \cdots & b(N - 1, N - 1)
\end{bmatrix}
\]

where \(R_s\) denotes the \(N\) by \(N\) autocorrelation matrix of \(x(n)\) whose \((j, k)\)th element is \(r_s(j - k)\). Throughout this paper, \('\) and \(\tr\) denote transposition and trace, respectively. \(A\) and \(B\) will be called the linear and quadratic filter operators, respectively.

Before the derivation, let us define the cross-correlation and cross-bicorrelation functions between \(s(n)\) and \(x(n)\) as follows:

\[
r_{sx}(j) = E[s(n) x(n - j)] \\
t_{sx}(j, k) = E[s(n) x(n - j) x(n - k)]
\]

Since \(s(n)\) and \(x(n)\) are assumed to be strictly stationary, both \(r_{sx}(j)\) and \(t_{sx}(j, k)\) are independent of the variable \(n\). In particular, the cross-bicorrelation \(t_{sx}(j, k)\) measures the "third-order" statistical dependency between \(s(n)\) and \(x(n)\), which is crucial in finding the optimum quadratic filter operator. Also note the symmetry of the cross-bicorrelation function, i.e., \(t_{sx}(j, k) = t_{sx}(k, j)\). We also define \(R_{sx}\) and \(T_{sx}\) as

\[
R_{sx} = [r_{sx}(0), \ldots, r_{sx}(N - 1)]' \\
T_{sx} = \begin{bmatrix}
 t_{sx}(0, 0) & \cdots & t_{sx}(0, N - 1) \\
 \vdots & \ddots & \vdots \\
 t_{sx}(N - 1, 0) & \cdots & t_{sx}(N - 1, N - 1)
\end{bmatrix}
\]

Now it is seen from (2) that the linear and quadratic filter operators with minimum MSE should satisfy the following matrix equations:

\[
E[X(n) s(n)] = E[X(n) A'X(n) + X(n) A X'(n) - R_s] \\
\cdot \tr \{B[X(n) X'(n) - R_s]\} \tag{7}
\]

\[
E[X(n) X'(n) s(n)] = E[X(n) X'(n) A X(n) + X(n) X'(n) A'X(n) - R_s] \\
\cdot \tr \{B[X(n) X'(n) - R_s]\}. \tag{8}
\]

On the other hand, it is easy to see that

\[
E[X(n) A'X(n)] = R_s A \\
E[X(n) \tr \{B[X(n) X'(n) - R_s]\}] = 0_{N \times 1} \\
E[X(n) X'(n) A'X(n)] = 0_{N \times N}.
\]

Furthermore, we have

\[
E[X(n) X'(n) \tr \{B[X(n) X'(n) - R_s]\}] = 2R_s B R_s
\]
which is a consequence of the facts that $B$ is symmetric and
\[
E[x_1x_3x_5x_4] = E[x_1x_2] E[x_3x_4]
\]
+ $E[x_1x_3] E[x_3x_4] + E[x_1x_4] E[x_2x_3]
\]
for zero-mean jointly Gaussian $x_1$, $x_2$, $x_3$, and $x_4$. Consequently, (7) and (8) become
\[
R_{xx} = R_x A
\]
\[
T_{xx} = 2R_x B R_x^T.
\]
So, if we assume that $R_x$ is positive definite with its inverse $R_x^{-1}$, the linear and quadratic filter operators with minimum MSE are given by the following two simple equations:
\[
A_0 = R_x^{-1} R_y
\]
\[
B_0 = (1/2) R_x^{-1} T_x R_x^{-1}.
\]
It is noted from (9) that the linear filter operator of an optimum SVF is exactly same as the optimum linear filter. Consequently, one can construct the SVF simply by adding a quadratic filter in parallel to a predesigned linear filter without changing the linear filter. Even though this might be what one expected naturally, there is no reason to presume it. Also, (10) implies that once the inverse $R_x^{-1}$ is computed, the quadratic filter operator as well as the linear one is directly obtained without solving another system of equations. Since the autocorrelation matrix $R_x$ is in the Toeplitz form, the number of operations required to compute its inverse is only of order $N^3$.

To evaluate the performance of the optimum SVF, we first note that the MSE of an SVF is given by
\[
\xi = r_x(0) + A'(R_x - 2R_x) A
\]
+ $2 \operatorname{tr} \{B(R_x B R_x^T - T_x)\}$
\]
where $r_x(0) = E[s^2(n)]$ and the following equalities have been used:
\[
E[s(n)] \{B(X(n) X'(n) - R_2)\} = \{BT_x\}
\]
\[
E[\operatorname{tr} \{B(X(n) X'(n) - R_2)\}^2] = 2 \operatorname{tr} (BR_x B R_x^T).
\]
Consequently, by substituting $A$ and $B$ in (11) by (9) and (10), the MSE of the optimum SVF becomes
\[
\xi_{opt} = r_x(0) - R_x R_x^{-1} R_x - (1/2) \operatorname{tr} (R_x^{-1} T_x R_x^{-1} T_x). \quad (12)
\]
Note that the first two terms of the right-hand side of (12) are equal to the MSE of an optimum linear filter. So, the third term, which is nonnegative, represents the reduction of MSE achieved by the quadratic filter operator. In particular, if $x(n)$ and $s(n)$ are jointly Gaussian, the third term is zero ($T_{xx} = 0_{N \times N}$) and the optimum linear filter is the best possible filter.

It seems useful at this point to discuss the input/output relationship of the SVF in the frequency domain. Let us first note that the autocorrelation function of the filter output $y(n)$ is given by
\[
r_y(n) = \sum_{j=0}^{N-1} a(j) a(k) r_x(n + j - k)
\]
+ $2 \sum_{j,k,m,s=0} b(j, k) b(m, s) \cdot r_x(n - j + m) r_x(n - k + s).
\]
We also define the power spectral densities of $x(n)$ and $y(n)$ as
\[
S_x(f) = \sum_{k=-\infty}^{\infty} r_x(k) e^{-j2\pi f k},
\]
\[
S_y(f) = \sum_{k=-\infty}^{\infty} r_y(k) e^{-j2\pi f k}.
\]
Then, from the duality between the multidimensional convolution and Fourier transform, it can be shown that
\[
S_y(f) = |H_A(f)|^2 S_x(f) + 2F_2[H_B(f_1, f_2)]^2 \cdot S_x(f_1) S_x(f_2) (f)
\]
\[
= \sum_{j=0}^{N-1} a(j) e^{-j2\pi j f_1} + \sum_{k=0}^{N-1} a(k) e^{-j2\pi k f_2}.
\]
Also, the operation $F_2$ is given by
\[
F_2[X(f_1, f_2)] (f) = \int_{-0.5}^{0.5} X(f', f - f') \, df' \quad (14)
\]
for any $X(f_1, f_2)$. Consequently, the operation transforms a function on the $(f_1, f_2)$ plane to another function of $f$ by integrating the first one along the line $f = f_1 + f_2$. So it is seen from (13) and (14) that, for fixed $f_1$, $S_y(f)$ is affected not only by $S_x(f)$, but also by all possible pairs $S_x(f_1)$ and $S_x(f_2)$, with $f_1 + f_2 = f$. Here the QTF plays a similar role as the LTF in that it acts as a weighting function in the operation $F_2$.

On the other hand, if we define the cross-spectral and cross-bispectral densities between $y(n)$ and $x(n)$ as
\[
S_{yx}(f) = \sum_{j=-\infty}^{\infty} r_{yx}(j) e^{-j2\pi j f}
\]
\[
C_{yx}(f_1, f_2) = \sum_{j=-\infty}^{\infty} t_{yx}(j, k) e^{-j2\pi (f_1 j + f_2 k)}
\]
where $r_{yx}(j) = E[y(n) x(n-j)]$ and $t_{yx}(j, k) = E[y(n) x(n-j) x(n-k)]$, then we have
\[
S_{yx}(f) = H_A(f) S_x(f), \quad (15)
\]
which is a familiar result from linear system theory, and
\[
C_{yx}(f_1, f_2) = 2H_B(f_1, f_2) S_x(f_1) S_x(f_2). \quad (16)
\]
Note from the above equation that the cross-bispectral density can specify the QTF uniquely, assuming that $S_x(f)$ is known and $S_x(f) \neq 0$ for all $f$. This also reveals the possibility that one can compute the QTF directly from (16) using an estimate of the cross-bispectral density. This technique, called cross-bispectral analysis, can be implemented efficiently by using the FFT algorithm. However, it suffers the same problem as the linear approach using (15): the QTF so obtained is not causal, in general. For more detail on cross-bispectral analysis, see [21].

**Estimation of Cross Bicorrelation**

Assuming that the optimum linear filter has been already constructed, the only additional information required to build an SVF is the cross-bicorrelation function $r_x(j, k), 0 \leq j, k \leq N - 1$. Given observations $x(n)$ and $y(n), n = 1, \cdots, M$, the following estimator seems to be suitable:

$$
\hat{r}_x(j, k) = (1/C_{jk}) \sum_{n=1}^{M-j} x(n) x(n + j - k) s(n + j) \tag{17}
$$

for $0 \leq k \leq j \leq M$. In particular, when we choose $C_{jk} = M$, it is straightforward to see that the estimator is asymptotically unbiased and its variance decreases linearly with $1/M$ for large $M$. A detailed statistical analysis of the estimator is beyond the scope of this paper and will not be discussed.

**IV. ITERATIVE FACTORIZATION**

It has been noted in Section III that the SVF with minimum MSE is easily determined from (9) and (10) when the filter input is Gaussian. However, the result does not solve the implementational complexity of the SVF: given the (optimum) linear and quadratic filter operators with length $N$, operations of order $N^2$ are still required at each instant of time to compute the filter output. Recall that a linear filter with the same length requires operations of only order $N$. The implementational complexity, which exists inherently in the general Volterra filters, poses another major problem in the application of the SVF. In some practical cases, tradeoffs between the filter performance and its computational requirement are necessary. In this section, two special classes of the SVF, which can be implemented with operations of order $N$, are discussed along with their design problems. In the following, we consider only the quadratic filter operator of the SVF since the linear filter operator of an optimum SVF, which is same as the optimum linear filter, is not affected by the particular form of the quadratic filter operator.

**Class I (Multiplier + Linear Filter)**

The quadratic filter of this class is given by

$$
y(n) = \sum_{j=0}^{N-1} g(j) [x^2(n - j) - r_x(0)]. \tag{18}
$$

As shown schematically in Fig. 1(a), the filter consists of a multiplier followed by a linear filter. It also corresponds to the case where the quadratic filter operator is diagonal, i.e., $b(j, k) = 0$ for $j \neq k$.

**Class II (Linear Filters + Multiplier)**

The quadratic filter of Class II is given by

$$
y(n) = \sum_{j,k=0}^{N-1} g_1(j) g_2(k) [x(n - j) - r_x(j - k)]. \tag{19}
$$

So it corresponds to the case that the quadratic filter operator can be factorized into two linear filters in that $b(j, k) = [g_1(j) g_2(k) + g_1(k) g_2(j)]/2$. Consequently, the filter consists of a parallel combination of two linear filters whose outputs are multiplied to give the actual filter output, as shown in Fig. 1(b).

Now suppose that, due to the implementational complexity, we have to sacrifice the optimality of the solution in (10) and we want to find the minimum MSE filter in Class I or II. In the case of Class I, it is straightforward to see that the MSE is minimized by choosing

$$
G = (1/2) R_{x^2}^{-1} R_{x^2} \tag{20}
$$

where

$$
G = [g(0), \cdots, g(N - 1)]',
$$

$$
R_{x^2} = [t_x(0, 0), \cdots, t_x(N - 1, N - 1)]'
$$

and $R_{x^2}$ is an $N \times N$ matrix whose $(j, k)$th element is given by $r_x^2(j - k)$. Incidentally, the MSE of the quadratic filter is also given by

$$
\xi = r_x(0) - R_{x^2}^{-1} R_{x^2} \xi. \tag{22}
$$

On the other hand, to consider the minimization in Class II, we first note that the filter in (19) can be rewritten as

$$
y(n) = G_1(X(n) X'(n) - R_x) G_2 \tag{21}
$$

where $G_1 = [g_1(0), \cdots, g_1(N - 1)]'$ and $G_2 = [g_2(0), \cdots, g_2(N - 1)]'$. The MSE between the filter output and $s(n)$ is then expressed in terms of $G_1$ and $G_2$ as

$$
\xi = r_x(0) + G_1 R_x G_2 + (G_1 R_x G_2)^2 - 2G_1 T_x G_2 \tag{22}
$$
where the following equalities have been used:

\[ E[s(n)G'_1(X(n))X'(n) - R_x]G_2 = G_1T_{st}G_2 \]

\[ E[(G'_1(X(n))X'(n) - R_x)G_2^2] = G'_1R_xG'_2G_2 + (G'_1R_xG_2)^2. \]

The next step is to minimize (22) over \(G_1\) and \(G_2\). However, the simultaneous minimization over \(2N\) variables presents potential numerical problems due to couplings between \(G_1\) and \(G_2\). As an alternative to the simultaneous minimization, we describe an iterative method in the following. In this and following sections, we define \(\alpha_j/Y\), given a scalar \(x\) and \(M \times N\) matrix \(Y\), as the \(M \times N\) matrix whose \((j, k)\)th element is the derivative of \(x\) with respect to the \((j, k)\)th element of \(Y\).

Now suppose \(G_1\) is already chosen properly and we need to find \(G_2\) which minimizes (22) with the choice of \(G_1\). Then, it is easy to see that

\[ \frac{\partial \xi}{\partial G_2} = 2G'R_xG'_1R_xG_2 + 2R_xG'_1G'_2G_2 - 2T_{st}G_1. \]

So, by letting \(\frac{\partial \xi}{\partial G_2} = 0\), we have

\[ (G'_1R_xG'_1R_x + R_xG'_1G'_2R_x)G_2 = T_{st}G_1. \]

Consequently, by noting that \(G'_1R_xG'_1R_x + R_xG'_1G'_2R_x\) is positive definite, the optimum \(G_2\) is given by

\[ G_2 = (G'_1R_xG'_1R_x + R_xG'_1G'_2R_x)^{-1} T_{st}G_1. \quad (23) \]

Furthermore, it can be shown by using the matrix inversion lemma [22] that (23) can also be expressed as

\[ G_2 = \alpha_1[R_x^{-1} - (\alpha_1/2) G'_1G'_1] T_{st}G_1 \quad (24) \]

\[ B_0 = \begin{bmatrix} 0.120 & 0.085 & 0.050 \\ 0.085 & 0.060 & 0.035 \\ 0.050 & 0.035 & 0.020 \end{bmatrix} \]

\[ = \left(\frac{1}{2}\right) \begin{bmatrix} 0.4 & [0.3, 0.2, 0.1] \\ 0.3 & 0.2 \\ 0.1 \end{bmatrix} + \begin{bmatrix} 0.3 \\ 0.2 \\ 0.1 \end{bmatrix}. \]

where \(\alpha_1 = (G'_1R_xG'_{11})^{-1}\) is a positive scalar. In a similar manner, for fixed \(G_2\), (22) is minimized when

\[ G_1 = \alpha_2[R_x^{-1} - (\alpha_2/2) G'_2G'_2] T_{st}G_2 \quad (25) \]

where \(\alpha_2 = (G'_2R_xG'_{22})^{-1}\).

Thus, by using the above result, the iterative technique is given as follows.

**Step 1:** Choose the first linear filter \(G_1\) arbitrarily with \(G_1 \neq 0_{N \times 1}\).

**Step 2:** With the choice of \(G_1\), use (24) to determine \(G_2\) which minimizes the MSE.

**Step 3:** With the choice of \(G_2\), use (25) to determine the optimum \(G_1\).

**Step 4:** Repeat steps 2 and 3 until the improvement in the MSE of (22) is negligible.

At each stage of the iteration, the technique provides a quadratic filter of Class II whose MSE is no larger than that for the previous step of the iteration. Also, by virtue of the matrix inversion lemma, computing \(G_1\) or \(G_2\) at each stage can be done very effectively once the inverse \(R_x^{-1}\) is obtained. It should be noted, however, that unlike quadratic filters of the general type, a quadratic filter of Class II does not possess a unique solution for optimum \(G_1\) and \(G_2\) since the factorization in (19) itself is not unique. Furthermore, the coupling between \(G_1\) and \(G_2\) could allow, in principle, local minima in the MSE surface of the quadratic filter. Consequently, the convergence of \(G_1\) and \(G_2\) through the iteration is not easily established in the general case. However, in many cases of practical interest, the convergence of its MSE, in addition to its computational efficiency, seems to guarantee the usefulness of the iteration technique.

**Example (Iterative Method)**

Suppose \(x(n)\) is a stationary Gaussian process with \(r_x(0) = 1, r_x(1) = 0.5, r_x(2) = 0.2\). Also, let \(s(n)\) be a stationary process with \(r_s(0) = 0.3\) and the cross bicorrelation between \(s(n)\) and \(x(n)\) be given by

\[ T_{st} = \begin{bmatrix} 0.4956 & 0.4880 & 0.3340 \\ 0.4880 & 0.4800 & 0.3280 \\ 0.3340 & 0.3280 & 0.2236 \end{bmatrix}. \]

The above matrix has been chosen deliberately such that the corresponding optimum quadratic filter operator, given in (10), is factorizable, i.e.,

\[ B = \begin{bmatrix} 0.1181 & 0.0855 & 0.0529 \\ 0.0855 & 0.0599 & 0.0343 \\ 0.0529 & 0.0343 & 0.0156 \end{bmatrix}. \]

However, it should be remembered that the factorization is not unique: once \(B\) is factorizable, there exist an infinite number of different pairs \(G_1\) and \(G_2\) such that \(B = (G_1G'_2 + G'_2G'_1)/2\). We note by using (12) that the corresponding MSE is approximately equal to 0.0679.

Table I shows the result of the iterative method as given above with the initial condition \(G_1 = [1, 1, 1]'\). Note that after five iterations, the MSE reaches very close to the minimum MSE of the optimum \(B\). Also, the filter operators \(G_1\) and \(G_2\) seem to converge to the correct values fairly well. After five iterations, the corresponding quadratic filter operator, i.e., \(B = (G_1G'_2 + G'_2G'_1)/2\), becomes

\[ B = \begin{bmatrix} 0.1181 & 0.0855 & 0.0529 \\ 0.0855 & 0.0599 & 0.0343 \\ 0.0529 & 0.0343 & 0.0156 \end{bmatrix}. \]
V. ADAPTIVE ALGORITHM

The least mean square (LMS) algorithm for the adaptive linear filter is well known and is given by

$$A(n + 1) = A(n) - 2\mu_A e(n) X(n). \quad (27)$$

Here $A(n)$ is the linear filter operator at time $n$ and $e(n) = A'(n) X(n) - s(n)$ denotes the residual error of the filter. Also, $\mu_A$ is a positive constant which controls the stability and convergence speed of the algorithm. In his original development of the algorithm, Widrow [23] justified the algorithm as a stochastic variant of the steepest descent method and showed, by assuming independence between $A(n)$ and $X(n)$, that when $0 < \mu_A < \lambda_{\text{max}}^{-1}$, where $\lambda_{\text{max}}$ denotes the largest eigenvalue of the autocorrelation matrix $R_x$, the expectation of $A(n)$ approaches the optimum linear filter operator $A_0$. He also derived the following approximation for the asymptotic excess MSE of the algorithm:

$$\xi_A = \mu_A \xi_{\text{opt}} \text{tr} (R_x) \quad (28)$$

where $\xi_{\text{opt}}$ is the MSE of the optimum linear filter.

In considering the adaptive implementation of the SVF, the linear filter operator can be updated by using the same LMS algorithm as given in (27), except that the residual error is given by

$$e(n) = A'(n) X(n) + \text{tr} \{B(n) [X(n) X'(n) - R_x]\} - s(n). \quad (29)$$

Then it is straightforward to see that the previous results for the standard LMS algorithm are still valid. Consequently, we only need to consider the adaptation of the quadratic filter operator. However, it should be noted from (29) that the zeroth-order term $-\text{tr} \{B(n) R_x\}$ is not constant in the adaptive implementation. Recall that the zeroth-order term is required to subtract the expectation of the quadratic filter output from the output of SVF. Consequently, when $R_x$ is not known, a recursive estimator (such as a low-pass filter) for the mean level of the quadratic filter output can replace the zeroth-order term. For simplicity, we will regard the zeroth-order term as given, and hence its adaptation will not be considered in this discussion.

As an extension of the LMS algorithm, the following adaptive algorithm for the quadratic filter operator has been considered by several authors [11], [12], [14]:

$$B(n + 1) = B(n) - \mu_B e(n) X(n) X'(n) \quad (30)$$

where $\mu_B$ is a positive constant. Conceptually, the above algorithm seems to be reasonable since $\partial e^2(n)/\partial B(n) = 2e(n) X(n) X'(n)$, and also its performance has been shown to be acceptable [11]. However, until recently, the convergence and stability criteria of the algorithm have remained not established. In the following, we present some results on the asymptotic behavior of the algorithm. The discussion is mainly heuristic and utilizes various assumptions on the adaptation process which, in spite of being somewhat restrictive, have been widely accepted. Actually, the assumptions made here are essentially equivalent to ones used in the work of Widrow [23] on the LMS algorithm.

**Expectation of $B(n)$**

We define the matrix norm $\|Q\|$ of $Q$ as $\|Q\| = \text{tr} (Q Q')^{1/2}$. The following inequalities related to the norm will be helpful:

$$0 \leq \text{tr} (A' RA) \leq \lambda_{\text{max}} \|A\|^2 \quad (31)$$

$$0 \leq \text{tr} (BRBR) \leq \lambda_{\text{max}}^2 \|B\|^2 \quad (32)$$

$$\|BRB\|^2 \leq \lambda_{\text{max}}^2 \text{tr} (BRBR) \quad (33)$$

**TABLE I**

<table>
<thead>
<tr>
<th># of iterations</th>
<th>$G_1$</th>
<th>$G_2$</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.2200</td>
<td>.1200</td>
<td>.0200</td>
</tr>
<tr>
<td>2</td>
<td>.2145</td>
<td>.1208</td>
<td>.0272</td>
</tr>
<tr>
<td>3</td>
<td>.2108</td>
<td>.1214</td>
<td>.0320</td>
</tr>
<tr>
<td>4</td>
<td>.2081</td>
<td>.1218</td>
<td>.0355</td>
</tr>
<tr>
<td>5</td>
<td>.2060</td>
<td>.1221</td>
<td>.0382</td>
</tr>
<tr>
<td>100</td>
<td>.1883</td>
<td>.1246</td>
<td>.0609</td>
</tr>
</tbody>
</table>
for any $A$, symmetric $B$, and positive definite $R$ where $\lambda_{\text{max}}$ denotes the largest eigenvalue of $R$. Furthermore, we have $\text{tr} (BRBR) = 0$ only when $|B| = 0$. The proof of these inequalities is essentially straightforward.

The first approximation is to assume that both $A(n)$ and $B(n)$ are independent of the pair $\{s(n), X(n)\}$. Then we have $E[e(n) X(n) X'(n)] = 2R_x E[B(n)] R_e - T_{xx}$. So, if we define $\delta B(n) = B(n) - B_0$ where $B_0$ is the optimum quadratic filter operator given by (10), taking expectations on both sides of (30) yields

$$E[\delta B(n + 1)] = E[B(n)] - 2\mu_B R_x E[\delta B(n)] R_e.$$  

(34)

Note that the expectation of the quadratic filter operator converges to the optimum $B_0$ if and only if $\|E[\delta B(n)]\| \to 0$ as $n \to \infty$. Trivially, if $\|E[\delta B(n)]\| = 0$ for some $n$, it remains at zero after $n$. For $\|E[\delta B(n + 1)]\| \leq \|E[B(n)]\|$ if the step size $\mu_B$ is chosen such that

$$0 < \mu_B < \lambda_{\text{max}}^{-2}.$$  

(35)

To see this, we simply note by using (32)-(35) that

$$\|E[\delta B(n + 1)]\|^2 = \|E[\delta B(n)]\|^2 - 4\mu_B \text{tr} \{E[\delta B(n)] R_e E[\delta B(n)] R_e\} + 4\mu_B^2 \|R_e E[\delta B(n)] R_e\|^2 \leq \|E[\delta B(n)]\|^2$$

because $\mu_B (1 - \mu_B \lambda_{\text{max}}^2) > 0$. In fact, it can be shown by using a similarity transform of $R_e$ that

$$\|E[\delta B(n + 1)]\| < \lambda_M \|E[\delta B(n)]\|$$

where

$$\lambda_M = \max \{|1 - 2\mu_B \lambda_{\text{max}}^2|, |1 - 2\mu_B \lambda_{\text{min}}^2|\}.$$  

Hence, the mean error between $B(n)$ and $B_0$ decreases monotonically to zero as time increases. Furthermore, a close examination of the adaptation process shows that each element of the matrix $E[\delta B(n)]$ is composed of a mixture of damped exponentials of which the fastest and slowest damping modes are determined by $\lambda_{\text{max}}$ and $\lambda_{\text{min}}$.

**Excess MSE**

In the adaptive implementation of the SVF, the fluctuations (over time) of the linear and quadratic filter operators cause some additional MSE at the filter output, even in the steady state of the adaptation process. So, the asymptotic MSE of an adaptive SVF is generally larger than the MSE of the optimum SVF. To evaluate the amount of the excess MSE we first note, by conditioning on both $A(n)$ and $B(n)$, that the MSE of an adaptive SVF can be written as

$$\xi(n) = \xi_{\text{opt}} + \xi_A(n) + \xi_B(n)$$

where the excess MSE’s of the linear and quadratic filters are given by

$$\xi_A(n) = E[\{\delta A'(n) R_e \delta A(n)\}]$$  

(36)

$$\xi_B(n) = E[\{\delta B'(n) R_e \delta B(n) \delta B(n) R_e\}].$$  

(37)

and $\delta A(n) = A(n) - A_0$ and $\delta B(n) = B(n) - B_0$ are the deviations of $A(n)$ and $B(n)$ from their optimum solutions. Then, by using (31) and (32), we have the following bounds for the excess MSE’s:

$$\xi_A(n) \leq \lambda_{\text{max}} E[\{\delta A(n)\}^2]$$  

(38)

$$\xi_B(n) \leq \lambda_{\text{max}}^2 E[\{\delta B(n)\}^2].$$  

(39)

Since we are mainly concerned with the asymptotic values of the excess MSE’s after the adaptation process reaches its steady state, we assume that $A(n)$ and $B(n)$ are close to $A_0$ and $B_0$, respectively, with proper choices of $\mu_A$ and $\mu_B$ such that $0 \leq \mu_A \leq \lambda_{\text{max}}^{-1}$, and $0 \leq \mu_B \leq \lambda_{\text{max}}^{-2}$. So we have $\delta A(n) = -2\mu_A e(n) X(n)$ and $\delta B(n) = -\mu_B e(n) X(n)$ and $X'(n)$. Also, under these conditions, both $X(n)$ and $X'(n)$ are uncorrelated with the residual error $e(n)$. Furthermore, we assume that they are independent of $e(n)$. Then

$$E[\{\delta A(n)\}^2] = 4\mu_A^2 \xi_{\text{opt}} \text{tr} (R_x)$$  

(40)

$$E[\{\delta B(n)\}^2] = \mu_B^2 \xi_{\text{opt}} \text{tr} (R_x^2) + \text{tr}^2 (R_x).$$  

(41)

Consequently, by combining (38)-(41), we have the following bounds for the asymptotic excess MSE’s (AEMSE):

$$\xi_A \leq 4\mu_A \xi_{\text{opt}} N_{rx}(0)$$  

$$\xi_B \leq 3\mu_B \xi_{\text{opt}} N_{rx}(0)^2.$$  

Notice that both linear and quadratic AEMSE’s are bounded by linear functions of $\mu_A$ and $\mu_B$, respectively.

In the following, we examine the AEMSE of the quadratic filter more closely by taking the dynamics of $\delta B(n)$ into consideration. The adaptive algorithm of (30) can be reformulated in terms of $\delta B(n)$ as follows:

$$\delta B(n + 1) = \delta B(n) + \mu_B e(n) X(n) X'(n)$$

(42)

$$= \delta B(n) - \mu_B [2R_e \delta B(n) R_e + W(n)].$$

where $W(n) = e(n) X(n) X'(n) - 2R_e \delta B(n) R_e$ is an $n \times n$ noise matrix with zero mean. Note the similarity between (34) and (42). Actually, when the noise term $W(n)$ is omitted, (42) becomes the recursive equation of the steepest descent method. So, $W(n)$ can be regarded as the additive measurement noise of the gradient matrix $2R_e \delta B(n) R_e$. We assume that the measurement noise is uncorrelated with $\delta B(n)$. Then we have

$$E[\{\delta B(n + 1)\}^2] = E[\{\delta B(n)\}^2] - 4\mu_B E \{\text{tr} \{\delta B(n) R_e \delta B(n) R_e\}\} + \mu_B^2 E[\{W(n)\}^2]$$

$$+ 4\mu_B^2 E[\{R_e \delta B(n) R_e\}^2].$$  

(43)
In the steady state of the adaptation, we can assume $E[\|\delta B(n + 1)\|^2] = E[\|\delta B(n)\|^2]$. In addition, for $\mu_B \ll \lambda_{\text{max}}^2$, we have from (33) that

$$\mu_B E[\|R, \delta B(n) R_r\|^2] \ll E[\text{tr} \{\delta B(n) R_r \delta B(n) R_r\}].$$

(44)

Consequently, (43) implies that

$$4\mu_B E[\text{tr} \{\delta B(n) R_r \delta B(n) R_r\}] = \mu_B^2 E[\|W(n)\|^2].$$

Furthermore, using the previous asymptotic assumption, i.e., $B(n) = B_0$ and $W(n) = \epsilon(n) X(n) X'(n)$, and (37) and (44), we have the following approximation for the quadratic AEMSE:

$$\xi_B = (\mu_B^2/4) \xi_{\text{opt}} [2 \text{tr} (R_r R_r) + \text{tr}^2 (R_x)].$$

VI. APPLICATION

In this section, we present some results of experiments which were conducted using field data to demonstrate the utility of the SVF techniques in practical applications. Specifically, we utilized the SVF techniques in the problem of modeling and predicting the dynamic behavior of moored vessels due to irregular sea waves. When moored in a random sea, ships and barges undergo large-amplitude oscillations near the undamped natural frequency of the vessel-mooring system. The relationship between random sea wave excitation and corresponding response of a moored vessel is highly nonlinear, which is revealed by the fact that the response occurs at a low-frequency region far below the frequencies of the incident sea waves. This phenomenon, known as the low-frequency drift oscillation (LFDO), has been an important subject in offshore engineering and related applications. We refer to [24] and [25] for detailed physical accounts of the phenomenon.

A typical example of the LFDO can be observed from Fig. 2, in which some data from a scaled model wave basin test [15] are shown. Fig. 2(a) and (b) shows the time series and power spectrum of the incident sea wave. Fig. 2(c) illustrates a barge-mooring configuration and Fig. 2(d) and (e) are the sway response of the barge and its power spectrum. Note that the sway response of the barge occurs at a frequency considerably less than those frequencies associated with the random sea waves. The LFDO of the barge, or a vessel-mooring system in general, in response to sea wave fluctuations can be explained in terms of the following physical picture. When random sea waves impinge upon a moored vessel, a weak second-order force is set up, this force being proportional to the square of the instantaneous wave heights. Due to this square-law relationship, various spectral components in the incident sea wave spectrum will mix to form sum and difference frequencies. Some of the difference frequencies will lie within the resonant bandwidth of the vessel-mooring system, which basically consists of a mass (the vessel) and spring (the mooring lines), and hence a large-amplitude low-frequency response occurs at the resonant frequency of the vessel-mooring system. Meanwhile, since the resonant frequency (approximately 0.01 Hz) is well below the dominant frequencies associated with the sea wave input, the linear response of the moored vessel is considerably smaller than one due to the second-order force. Note that this simplified model for the LFDO mechanism is actually represented by an SVF in Class 1, i.e., a square-law device followed by a linear filter.

The first experiment we did was to see how well the SVF can model the LFDO of the barge. In this case, the filter input is the incident sea wave, while the primary signal becomes the actual barge response at time $n$. The linear and quadratic filter operators were computed from (9) and (10) in which $R_x$, $R_{xx}$, and $T_{x}$ were estimated by using proper time averages. The filter length was chosen to be as large as 150 to ensure that the response of the barge with large mass can be modeled properly. Fig. 3 shows the resulting output of the SVF, overlapped with the actual sway response of the barge. Note the close correspondence between the two time series. The ratio of the average MSE to the variance of the barge response is about 0.09, which suggests that the SVF provides an excellent model for the LFDO of the barge. In Fig. 4, the output of the linear filter, given by (9), is shown for the purpose of comparison. Corresponding (normalized) MSE of the linear filter is 0.87, which is unacceptable for practical purposes. This large error reflects the fact that the LFDO is dominated by a quadratic nonlinearity and, hence cannot be properly modeled by a linear filter.

On the other hand, Fig. 5 shows the output of the SVF in Class 1 which has been discussed in Section IV. Here the filter operator was computed from (20). Note that the
filter shows a noticeable improvement in performance compared to the linear filter in Fig. 4. Its (normalized) MSE is 0.39, which is four times greater than the MSE of the SVF in Fig. 3, but substantially less than the MSE of linear filter. The usefulness of the SVF in Class I to modeling vessel-mooring systems is also consistent with the previous physical explanation for the LFDO phenomenon.

Secondly, we also investigated the possibility of predicting the barge response ahead of time by using the SVF technique. The availability of such predicted values would be useful in the control and stabilization of vessel-mooring systems. The prediction time interval was chosen to be 13.86 s, with a sampling interval of 1.386 s. Also, the filter length was $N = 150$. Fig. 6 shows an example of the predicted and actual responses of the barge. Based on the fact that the LFDO is dominated by the quadratic nonlinearity, the linear filter was suppressed and only the quadratic filter operator, as given in (10), was used in this experiment. Here predictions were made at every 6.93 s. Finally, Fig. 7 shows the predicted and actual responses of the barge, in which the adaptive algorithm of (30) was utilized to update the quadratic filter operator with step size $\mu_B = 0.5 \times 10^{-5}$. In order to demonstrate the performance of the adaptive algorithm in both the transient and steady states, results of two successive iterations are shown. In the first iteration, initial values of the quadratic filter operator were set to identically zero. However, in the second iteration, the initial values were set equal to the final values of the quadratic filter operator in the first iteration. Again, note the close correspondence between the predicted and actual barge responses near the end of the second iteration where the adaptation reaches its steady state.

VII. CONCLUSION

In this paper, various new techniques related to the second-order Volterra filtering problem have been presented, along with some practical application examples. The results presented here are intended mainly to reduce the complexity involved with the Volterra filtering problem. In addition to the simple solution for the optimum Volterra filter, both the iterative factorization and the adaptive algorithm can be utilized very effectively and make possible simple and versatile approaches to design and implement the Volterra filters. We also expect that the application results, which were discussed in the last section, will motivate further potential applications of Volterra filtering techniques in various fields.

On the other hand, extension of the results to higher order Volterra filtering problems is also an interesting subject of potential application. From a theoretical viewpoint, one can expect it to be relatively straightforward since the convolutional structure of Volterra series carries over to
all higher order terms of such filters. It should be noted, however, that statistical couplings existing among even or odd terms of a higher order Volterra filter might require dealing with some additional problems. A possible approach to cope with the problems is to utilize a set of orthonormalized functionals which are formed in terms of Hermite polynomials [6]. The orthogonality among the terms of a higher order (orthogonalized) Volterra filter can simplify its design and analysis considerably. In fact, the SVF in (5) can be recognized as a truncation of the orthonormalized form. Another important problem involved with higher order Volterra filters is that the computational complexity associated with actual filtering operations, which increases exponentially with the filter order, becomes quite demanding, even with a moderate filter length. One should carefully compare the computational complexity to the performance improvement to be achieved in justifying the use of higher order Volterra filters. Some practical problems associated with higher order Volterra filters and their treatment are currently being investigated by the authors.

ACKNOWLEDGMENT

We would like to thank Prof. R. W. Miksad of the University of Texas, Austin, and Dr. F. J. Fischer of Shell Development Company for the barge-mooring data.

REFERENCES


Edward J. Powers received the B.S. degree from Tufts University, Medford, MA, the M.S. degree from the Massachusetts Institute of Technology, Cambridge, and the Ph.D. degree from Stanford University, Stanford, CA, in 1957, 1959, and 1965, respectively. From 1959 to 1965 he was employed by Lockheed Missiles and Space Company in both Sunnyvale and Palo Alto, CA. At Lockheed, his work was concerned with plasma physics and various electromagnetic aspects of reentry vehicles. In 1965 he joined the University of Texas, Austin, where he is currently the Texas Atomic Energy Research Foundation Professor and Chairman of the Department of Electrical and Computer Engineering and Director of the Electromagnetics Research Center. His current research interests include time series analysis of random data associated with nonlinear wave and turbulence phenomena in a wide variety of different physical media.

Taiho Koh was born in Moonkyung, Korea, on March 12, 1954. He received the B.S. degree in electronic engineering from Yonsei University, Seoul, Korea, in 1977, and the M.S. and Ph.D. degrees in 1982 and 1985, respectively, both in electrical engineering and from the University of Texas, Austin.

Since 1980 he held a research position in the Electronics Research Center, University of Texas, Austin, where he engaged in research on time series analysis of random data associated with nonlinear wave and turbulence phenomena. His current research interests include the development of nonlinear digital filtering techniques, adaptive estimation algorithms, and parametric spectral estimations. He is currently a member of the Technical Staff of A&T&T Bell Laboratories, Murray Hill, NJ.