Nonlinear Dynamics (2006) 43: 277–310
DOI: 10.1007/s11071-006-8323-x

Multiwavelet Constructions and Volterra Kernel Identification

RICHARD J. PRAZENICA[1,*] and ANDREW J. KURDILA[2]
[1]*NASA Dryden Flight Research Center, Edwards, CA 93523, U.S.A.;* [2]*Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611, U.S.A.;* **Author for correspondence (e-mail: rjp@gerc.eng.ufl.edu; fax: +1-850-833-9366)*

(Received: 2 February 2004; accepted: 26 May 2005)

Abstract. The Volterra series is commonly used for the modeling of nonlinear dynamical systems. In general, however, a large number of terms are needed to represent Volterra kernels, with the number of required terms increasing exponentially with the order of the kernel. Therefore, reduced-order kernel representations are needed in order to employ the Volterra series in engineering practice. This paper presents an approach whereby multiwavelets are used to obtain low-order estimates of first-, second-, and third-order Volterra kernels. A family of multiwavelets is constructed from the classical finite element basis functions using the technique of intertwining. The resulting multiwavelets are piecewise-polynomial, orthonormal, compactly-supported, and can be constructed with arbitrary approximation order. Furthermore, these multiwavelets are easily adapted to the domains of support of the Volterra kernels. In contrast, most wavelet families do not possess this characteristic. Higher-dimensional multiwavelets can easily be constructed by taking tensor products of the original one-dimensional functions. Therefore, it is straightforward to extend this approach to the representation of higher-order Volterra kernels. This kernel identification algorithm is demonstrated on a prototypical oscillator with a quadratic stiffness nonlinearity. For this system, it is shown that accurate kernel estimates can be obtained in terms of a relatively small number of wavelet coefficients. These results indicate the potential of the multiwavelet-based algorithm for obtaining reduced-order models for a large class of weakly nonlinear systems.

Key words: intertwining, multiresolution analysis, multiwavelets, nonlinear system identification, Volterra series

1. Introduction

The Volterra series provides a convenient framework for the representation of nonlinear dynamical systems. The system response is expressed in terms of an infinite series of integral operators of increasing order. Each operator takes the form of a multidimensional convolution of the input with a kernel function. The first-order Volterra kernel represents the linear dynamics of the system while the higher-order kernels describe the nonlinear dynamics. Volterra series representations are applicable to a large class of nonlinear systems. They have been employed in a diverse array of engineering applications including the modeling of biological [1, 2] and aeroelastic [3–5] systems. Volterra filters have been used extensively in the electrical engineering field to compensate for signal distortion from nonlinear disturbances [6, 7]. In addition, Volterra models have been used in nonlinear control strategies such as nonlinear model predictive control [8].

In practice, a serious limitation of the Volterra series is that it is difficult to identify the kernels that characterize a given system. Kernel identification is fundamentally an ill-posed problem since the objective is to determine the structure of the system from input/output measurements. Furthermore, the Volterra series is not orthogonal and, therefore, the Volterra kernels must be identified simultaneously. To address this problem, Wiener developed a variation of the Volterra series that is orthogonal provided that the input signal is Gaussian white noise. A number of statistical approaches such as the cross-correlation technique of Lee and Schetzen [9] have been developed to measure Wiener kernels. These approaches

are limited, however, due to the requirement of a Gaussian white noise input which is difficult, if not impossible, to generate in experimental systems. Volterra kernel identification is also difficult due to the large number of terms that are typically needed to model the kernels. For example, consider a system with a memory of M samples (i.e., the present input to the system affects the next M samples of the response). A simple discrete Volterra model for such a system would require on the order of M^p discrete coefficients for the representation of the pth-order kernel. Even a Volterra model that includes only the first few kernels would require a prohibitively large number of coefficients. Not only is the identification of such a model difficult, the model would be too large to be of practical use in many applications such as control design. Clearly, it is necessary to obtain lower-order kernel representations.

In consideration of these limitations, many other approaches to kernel identification have been applied in both the time and frequency domains. Some techniques entail measuring the kernels by applying specific input excitations to the system. For example, Boyd et al. [10] developed a harmonic probing technique in the frequency domain and Silva [4, 5] applied discrete impulse inputs to measure time-domain kernels from computational fluid dynamics models. These methods require specific inputs that are difficult to apply to experimental systems, however. Neural networks have also been used to estimate Volterra kernels, and Wray and Green [11] have demonstrated that many artificial neural networks have associated infinite-dimensional Volterra series representations. Another common approach to kernel identification is to expand the kernels in terms of a set of basis functions. For example, Marmarelis [12] estimated kernels for biological systems in terms of discrete Laguerre functions and Reisenthel [13] used decaying exponential functions to identify first- and second-order kernels for aeroelastic systems. These methods seek to represent the kernels in terms of a relatively compact set of globally-supported basis functions. They do not require any specific input excitations, although sufficiently broadband inputs are required for accurate kernel estimation.

The approach taken in this paper is to represent first-, second-, and third-order kernels in terms of a multiwavelet basis. Wavelets and multiresolution analysis afford time and frequency localization of functions and signals. In many cases, a function can be represented in terms of a relatively small number of wavelet coefficients. Beylkin et al. [14] have demonstrated that wavelets are particularly effective for compressing various integral operators. Many subsequent researchers have sought to take advantage of these results. For example, Nikolaou and Mantha [15] used biorthogonal wavelets to compress first- and second-order Volterra kernels, and Chou and Guthart [16] derived sparse representations of Green's function operators in terms of an orthonormal wavelet basis.

In earlier work, similar to the approach taken by Nikolaou and Mantha, the authors considered reduced-order Volterra kernel representations in terms of biorthogonal wavelets [17]. This approach utilized a family of biorthogonal wavelets constructed by Cohen et al. [18]. These wavelets possess several desirable properties such as biorthogonality, symmetry or antisymmetry, and relatively compact support. In addition, this wavelet family can be extended to include functions of arbitrary approximation order. These biorthogonal wavelets do not exist in closed form but are rather defined by a limiting procedure. This is a disadvantage in that the functions can be more difficult to work with, particularly when computing quadratures. In addition, these wavelets are not easily adapted to finite domains and instead require some form of nontrivial modification at the boundaries. These boundary extensions, examples of which include periodic and antisymmetric treatments, can be quite simple or extraordinarily complicated depending on the application.

More recently, the authors constructed wavelets over the domain of support of the triangular form of the second-order Volterra kernel [19]. These triangular wavelets are orthonormal, compactly-supported, and are symmetric or antisymmetric. Furthermore, they were specifically constructed to fit the domain of the second-order kernel and exist in closed form. These functions are piecewise-constant, however,

and consequently do not yield very smooth kernel estimates. In addition, the triangular wavelet construction is specific to the second-order kernel. The extension of this approach to higher-order kernels is not straightforward and would require complicated wavelet constructions over domains with complex geometries. Finally, this algorithm is relatively slow due to the complicated indexing system needed to define the triangular wavelet functions.

In this paper, piecewise-polynomial multiwavelets are constructed for the representation of Volterra kernels. This family of multiwavelets is generated using the technique of intertwining. Donovan et al. [20, 21] demonstrated that this method can be used to construct multiwavelets of arbitrary approximation order from the classical Lagrangian finite elements. The resulting multiwavelets possess a number of properties that make them particularly suitable for kernel approximation. In particular, they are orthonormal, compactly-supported, and symmetric or antisymmetric. As previously mentioned, they are closed-form (piecewise-polynomial) functions and can be constructed with arbitrary approximation order. In addition, these multiwavelets are easily adapted to the finite domains over which the kernels are supported. Finally, higher-dimensional multiwavelets are generated as the tensor products of one-dimensional scaling functions and multiwavelets. Therefore, it is straightforward to extend this multiwavelet-based approach to consider higher-order kernels. This class of piecewise-polynomial multiwavelets combines many of the desirable properties of the biorthogonal and triangular wavelets without most of the drawbacks. Therefore, the multiwavelet-based kernel identification algorithm described in this paper represents a significant improvement over previous wavelet-based approaches in terms of speed, accuracy, generality, and ease of implementation.

2. Volterra Series Representations

In a Volterra series representation, the output y of a dynamical system can be expressed in terms of an infinite series of integral operators:

$$y(t) = y_1(t) + y_2(t) + y_3(t) + \cdots + y_\infty(t) \tag{1}$$

where y_i denotes the output of the ith-order Volterra operator. For causal, time-invariant, single-input/single-output systems, the outputs of the first-, second-, and third-order operators take the form

$$y_1(t) = \int_0^t h_1(\xi)u(t-\xi)\,d\xi \tag{2}$$

$$y_2(t) = \int_0^t \int_0^t h_2(\xi,\eta)u(t-\xi)u(t-\eta)\,d\xi\,d\eta \tag{3}$$

$$y_3(t) = \int_0^t \int_0^t \int_0^t h_3(\xi,\eta,\gamma)u(t-\xi)u(t-\eta)u(t-\gamma)\,d\xi\,d\eta\,d\gamma \tag{4}$$

where u is the system input and h_1, h_2, and h_3 denote the first-, second-, and third-order Volterra kernels. A system is completely characterized in terms of its Volterra kernels. The first-order kernel describes the linear dynamics while higher-order kernels characterize nonlinearities in the system. The output of a linear system can be expressed as Equation (2), which is the well-known linear convolution integral. Therefore, for a linear system, the first-order Volterra kernel is equivalent to the impulse response of the system. The Volterra series generalizes the convolution concept to nonlinear systems by expressing the nonlinear dynamics in terms of higher-order convolution integrals. Detailed discussions of the properties

of the Volterra series, including existence and convergence results, can be found in the texts by Rugh [22] and Schetzen [23].

This paper considers models in which the Volterra series in Equation (1) is truncated to include only the first-, second-, and third-order operators. In other words, these models assume that the second- and third-order kernels are sufficient to characterize the nonlinear dynamics. This is not always the case, of course, and the approach taken in this paper is applicable only to a class of weakly nonlinear systems. It has been shown, however, that this class includes many nonlinear systems occurring in engineering practice. As discussed by Boyd [24], the Volterra series is applicable to systems with fading memory, which is a stronger requirement than continuity. Basically, a system exhibits fading memory if the influence of inputs occurring in the distant past has a diminishing effect on the present output. This implies that the Volterra kernels of the system decay to zero in a finite period of time. In this paper, the memory lengths of the first-, second-, and third-order kernels are denoted as T_1, T_2, and T_3, respectively.

As discussed in numerous references on Volterra theory, the higher-order Volterra kernels can take a variety of different forms, including the symmetric, triangular, and regular forms [22, 23]. The symmetric forms of the kernels are unique for a given system. Therefore, it can be assumed, without loss of generality, that the kernels are symmetric. Then, the second-order kernel can be assumed to be symmetric on the $[0, T_2] \times [0, T_2]$ square domain. Similarly, the third-order kernel is symmetric over a $[0, T_3] \times [0, T_3] \times [0, T_3]$ three-dimensional domain. In this paper, the symmetric forms of the kernel are used and, from this point onward, the notation h_2 and h_3 will denote the symmetric second- and third-order kernels.

3. Multiwavelets and Multiresolution

3.1. Multiresolution Analysis

In this paper, multiwavelets are used to generate reduced-order representations of Volterra kernels. In simple terms, wavelets are oscillatory, compactly-supported functions that are typically constructed to possess certain properties such as orthogonality, smoothness, and symmetry or antisymmetry. A multiwavelet basis for $L^2(\mathbb{R})$, the vector space of square-integrable functions, is composed of the scaled translates and dilates of multiple wavelet functions $\{\psi^1, \ldots, \psi^r\}$. These multiwavelets are generated from r scaling functions $\{\phi^1, \ldots, \phi^r\}$. Note that the single-generator case, in which there is only one wavelet ψ generated by one scaling function ϕ, constitutes a special case of multiwavelets. Because multiwavelets employ multiple scaling functions and wavelets, there is more freedom to design these functions to satisfy a greater range of properties.

By construction, the multiwavelets $\{\psi^1, \ldots, \psi^r\}$ and their associated scaling functions $\{\phi^1, \ldots, \phi^r\}$ satisfy the two-scale equations:

$$\phi^s(x) = \sqrt{2} \sum_{p,t} a_p^{s,t} \phi^t(2x - p), \quad s = 1, \ldots, r \tag{5}$$

$$\psi^s(x) = \sqrt{2} \sum_{p,t} b_p^{s,t} \phi^t(2x - p), \quad s = 1, \ldots, r \tag{6}$$

The two-scale equations state that the multiwavelets and scaling functions are formed as linear combinations of scaling functions with half the support (or double the frequency). The coefficients $\{a_p^{s,t}\}$ and $\{b_p^{s,t}\}$ are elements of scaling function and wavelet filter matrices, respectively. The scaled translates

and dilates of the scaling functions and multiwavelets are defined as

$$\phi^s_{j,k}(x) := 2^{j/2}\phi^s(2^j x - k) \tag{7}$$

$$\psi^s_{j,k}(x) := 2^{j/2}\psi^s(2^j x - k) \tag{8}$$

The parameter $j \in \mathbb{Z}$, where $\mathbb{Z}$ is the set of all integers, is a dilation index that determines the scale, or resolution level, of the function. The parameter $k \in \mathbb{Z}$ is a translation index that gives the position of the function on the real line. The $2^{j/2}$ term is a normalization factor that renders the functions orthonormal in $L^2(\mathbb{R})$. Orthonormal multiwavelets and scaling functions satisfy the following orthogonality properties:

$$\begin{aligned}
&1) \qquad \int_{\mathbb{R}} \phi^s_{j,k}(x)\phi^t_{j,m}(x)\,dx = \delta_{k,m}\delta_{s,t} \\
&2) \qquad \int_{\mathbb{R}} \psi^s_{j,k}(x)\psi^t_{j,m}(x)\,dx = \delta_{k,m}\delta_{s,t} \\
&3) \qquad \int_{\mathbb{R}} \phi^s_{j,k}(x)\psi^t_{j,m}(x)\,dx = 0, \quad \forall\, k, m \in \mathbb{Z}, \quad s, t \in \{1, \ldots, r\}
\end{aligned} \tag{9}$$

As usual, the Kronecker delta $\delta_{k,m}$ is defined as

$$\delta_{k,m} := \begin{cases} 1, & k = m \\ 0, & k \neq m \end{cases} \tag{10}$$

The translates of the scaling functions on different levels of resolution form a series of approximation spaces $\{V_j\}_{j\in\mathbb{Z}}$, where each space V_j is defined as

$$V_j := \operatorname{span}\left\{\phi^s_{j,k} : s \in \{1, \ldots, r\},\ k \in \mathbb{Z}\right\} \tag{11}$$

In particular, the space V_0 is defined as the span of all translates of the scaling functions $\{\phi^1, \ldots, \phi^r\}$:

$$V_0 := \operatorname{span}\{\phi^s(\cdot - k) : s \in \{1, \ldots, r\},\ k \in \mathbb{Z}\} \tag{12}$$

The spaces $\{V_j\}$ are nested so that they satisfy

$$\{0\} \cdots \subset V_{j-1} \subset V_j \subset V_{j+1} \subset \cdots L^2(\mathbb{R}) \tag{13}$$

In the limit as j approaches $-\infty$, corresponding to an infinitely-coarse resolution level, the spaces $\{V_j\}$ approach $\{0\}$, the space composed of the zero function. As j approaches ∞, corresponding to an infinitely-fine level of resolution, the spaces $\{V_j\}$ approach $L^2(\mathbb{R})$. The series of approximation spaces $\{V_j\}_{j\in\mathbb{Z}}$ is termed a multiresolution analysis since it enables one to view a function or signal at varying levels of resolution.

Similarly, a series of wavelet spaces $\{W_j\}_{j\in\mathbb{Z}}$ can be defined, where

$$W_j := \operatorname{span}\left\{\psi^s_{j,k} :\ s \in \{1, \ldots, r\},\ k \in \mathbb{Z}\right\} \tag{14}$$

The wavelet spaces $\{W_j\}$ form the differences between adjacent approximation spaces $\{V_j\}$:

$$W_j = V_{j+1} \ominus V_j \tag{15}$$

where $\ominus$ denotes the difference between two vector spaces. Equation (15) is a direct consequence of the two-scale equations and the orthogonality properties in Equation (9). Therefore, a given approximation space V_j can be decomposed as

$$V_j = V_{j-1} \oplus W_{j-1} \tag{16}$$

where $\oplus$ denotes the sum of vector spaces. Applied recursively, Equation (16) leads to the following decomposition of the space V_j:

$$V_j = V_{j_0} \oplus W_{j_0} \oplus W_{j_0+1} \oplus \cdots \oplus W_{j-2} \oplus W_{j-1} \tag{17}$$

where j_0 is the coarsest level used in the decomposition. In the limit as j approaches ∞ and j_0 approaches $-\infty$, we have

$$L^2(\mathbb{R}) = \bigoplus_{l=-\infty}^{\infty} W_l \tag{18}$$

That is, the family of scaled translates and dilates of the multiwavelets $\{\psi^1, \ldots, \psi^r\}$ forms an orthonormal basis for $L^2(\mathbb{R})$.

In a multiwavelet-based multiresolution analysis, a function $f \in L^2(\mathbb{R})$ is first approximated in terms of a fine-resolution space V_j as

$$f_j(x) = \sum_{s=1}^{r} \sum_{k \in \mathbb{Z}} \alpha_{j,k}^s \phi_{j,k}^s(x) \tag{19}$$

where $\{\alpha_{j,k}^s\}$ are scaling function expansion coefficients and f_j denotes an approximation on level j. Equation (17) implies that an equivalent, multiscale representation of f_j can be written as

$$f_j(x) = \sum_{s=1}^{r} \sum_{k \in \mathbb{Z}} \alpha_{j_0,k}^s \phi_{j_0,k}^s(x) + \sum_{s=1}^{r} \sum_{l=j_0}^{j-1} \sum_{k \in \mathbb{Z}} \beta_{l,k}^s \psi_{l,k}^s(x) \tag{20}$$

This representation is in terms of the scaling functions on the coarsest resolution level j_0 and the multiwavelets on levels j_0 through $j-1$. The coefficients in this multiscale expansion are termed the discrete wavelet transform of f_j. The advantage of this representation is that the wavelet coefficients give information about the function f in both the spatial and frequency domains. The spatial localization is due to the compact support of the multiwavelets. Frequency localization can be studied by taking the Fourier transform of the multiwavelets and scaling functions and noting the rapid decay outside certain frequency bands. Indeed, the coarsest scaling function coefficents can be viewed as a low-pass filtering of the function f while the wavelet coefficients on different resolution levels act as a series of bandpass filters [25]. The multiscale expansion in Equation (20) is often a very efficient representation in that many of the coefficients are close or equal to zero. Therefore, a function can often be accurately represented using relatively few multiwavelet coefficients.

Furthermore, the discrete wavelet transform can be calculated in a fast, efficient manner from the single-scale coefficients in Equation (19). We have the following decomposition formulas:

$$\alpha_{j-1,m}^s = \sum_{t=1}^{r} \sum_{k} a_{k-2m}^{s,t} \alpha_{j,k}^t \tag{21}$$

$$\beta^s_{j-1,m} = \sum_{t=1}^{r} \sum_k b^{s,t}_{k-2m} \alpha^t_{j,k} \tag{22}$$

Applied recursively, Equations (21) and (22) can be used to calculate the multiscale coefficients in Equation (20) from the single-scale coefficients in Equation (19). The single-scale coefficients can be reconstructed through the recursive application of the formula

$$\alpha^s_{j,m} = \sum_{t=1}^{r} \sum_k a^{t,s}_{m-2k} \alpha^t_{j-1,k} + \sum_{t=1}^{r} \sum_k b^{t,s}_{m-2k} \beta^t_{j-1,k} \tag{23}$$

It should be noted that the decomposition formulas in Equations (21) and (22) can be used to define a sparse, invertible transformation matrix $[T]$ that operates on a vector of single-scale coefficients to calculate the discrete wavelet transform. While the matrix $[T]$ is seldom formed in practice, it provides a convenient notation that will be used later in this paper.

3.2. Tensor-Product Multiwavelets

Higher-dimensional multiwavelets, which are needed for the representation of higher-order Volterra kernels, are obtained as the tensor products of the one-dimensional scaling functions and multiwavelets. Two-dimensional scaling functions are constructed by taking the tensor products of the scaling functions $\{\phi^1, \ldots, \phi^r\}$ in one dimension with those in the other dimension. Hence, a total of r^2 two-dimensional scaling functions are obtained:

$$\Phi^{(s,t)}(x, y) := \phi^s(x)\phi^t(y) \tag{24}$$

where $s, t \in \{1, \ldots, r\}$. Given r associated wavelets $\{\psi^1, \ldots, \psi^r\}$, three different types of tensor-product wavelets are obtained, taking the form

$$\begin{aligned} \Psi^{1,(s,t)}(x, y) &:= \phi^s(x)\psi^t(y) \\ \Psi^{2,(s,t)}(x, y) &:= \psi^s(x)\phi^t(y) \\ \Psi^{3,(s,t)}(x, y) &:= \psi^s(x)\psi^t(y) \end{aligned} \tag{25}$$

where, once again, $s, t \in \{1, \ldots, r\}$. Therefore, there are a total of $3r^2$ two-dimensional wavelets. It is convenient to introduce the following notation for the scaled translates and dilates of the two-dimensional scaling functions and wavelets:

$$\begin{aligned} \Phi^{(s,t)}_{j,(k,m)}(x, y) &:= \phi^s_{j,k}(x)\phi^t_{j,m}(y) \\ \Psi^{1,(s,t)}_{j,(k,m)}(x, y) &:= \phi^s_{j,k}(x)\psi^t_{j,m}(y) \\ \Psi^{2,(s,t)}_{j,(k,m)}(x, y) &:= \psi^s_{j,k}(x)\phi^t_{j,m}(y) \\ \Psi^{3,(s,t)}_{j,(k,m)}(x, y) &:= \psi^s_{j,k}(x)\psi^t_{j,m}(y) \end{aligned} \tag{26}$$

Similar to the one-dimensional case, a multiresolution analysis in two dimensions is obtained in terms of the spaces

$$V_j := \operatorname{span}\left\{ \Phi^{(s,t)}_{j,(k,m)} : s, t \in \{1, \ldots, r\},\ k, m \in \mathbb{Z} \right\} \tag{27}$$

$$W_j^1 := \operatorname{span}\left\{\Psi_{j,(k,m)}^{1,(s,t)} : s,t \in \{1,\ldots,r\},\ k,m \in \mathbb{Z}\right\} \tag{28}$$

$$W_j^2 := \operatorname{span}\left\{\Psi_{j,(k,m)}^{2,(s,t)} : s,t \in \{1,\ldots,r\},\ k,m \in \mathbb{Z}\right\} \tag{29}$$

$$W_j^3 := \operatorname{span}\left\{\Psi_{j,(k,m)}^{3,(s,t)} : s,t \in \{1,\ldots,r\},\ k,m \in \mathbb{Z}\right\} \tag{30}$$

where $j \in \mathbb{Z}$. The space V_j can be decomposed as

$$V_j = V_{j-1} \oplus W_{j-1}^1 \oplus W_{j-1}^2 \oplus W_{j-1}^3 \tag{31}$$

When the decomposition is carried out over multiple levels, we have

$$V_j = V_{j_0} \bigoplus_{l=j_0}^{j-1} \left\{W_l^1 \oplus W_l^2 \oplus W_l^3\right\} \tag{32}$$

A single-scale representation of a two-dimensional function f in terms of the scaling functions that span V_j is given by

$$f_j(x,y) = \sum_{k,m,s,t} \alpha_{j,(k,m)}^{(s,t)} \Phi_{j,(k,m)}^{(s,t)}(x,y) \tag{33}$$

An equivalent multiscale expansion can be written as

$$\begin{aligned} f_j(x,y) = & \sum_{k,m,s,t} \alpha_{j_0,(k,m)}^{(s,t)} \Phi_{j_0,(k,m)}^{(s,t)}(x,y) \\ & + \sum_{l=j_0}^{j-1} \sum_{k,m,s,t} \beta_{l,(k,m)}^{1,(s,t)} \Psi_{l,(k,m)}^{1,(s,t)}(x,y) \\ & + \sum_{l=j_0}^{j-1} \sum_{k,m,s,t} \beta_{l,(k,m)}^{2,(s,t)} \Psi_{l,(k,m)}^{2,(s,t)}(x,y) \\ & + \sum_{l=j_0}^{j-1} \sum_{k,m,s,t} \beta_{l,(k,m)}^{3,(s,t)} \Psi_{l,(k,m)}^{3,(s,t)}(x,y) \end{aligned} \tag{34}$$

Similar to the one-dimensional case, the following decomposition formulas can be derived:

$$\alpha_{j-1,(a,c)}^{(u,v)} = \sum_{k,m,s,t} a_{k-2a}^{u,s} a_{m-2c}^{v,t} \alpha_{j,(k,m)}^{(s,t)} \tag{35}$$

$$\beta_{j-1,(a,c)}^{1,(u,v)} = \sum_{k,m,s,t} a_{k-2a}^{u,s} b_{m-2c}^{v,t} \alpha_{j,(k,m)}^{(s,t)} \tag{36}$$

$$\beta_{j-1,(a,c)}^{2,(u,v)} = \sum_{k,m,s,t} b_{k-2a}^{u,s} a_{m-2c}^{v,t} \alpha_{j,(k,m)}^{(s,t)} \tag{37}$$

$$\beta_{j-1,(a,c)}^{3,(u,v)} = \sum_{k,m,s,t} b_{k-2a}^{u,s} b_{m-2c}^{v,t} \alpha_{j,(k,m)}^{(s,t)} \tag{38}$$

where $\{a_p^{s,t}\}$ and $\{b_p^{s,t}\}$ are the scaling function and wavelet filters that appear in the two-scale relationships in Equations (5) and (6). Applied recursively, Equations (35) through (38) can be used to

calculate the multiscale coefficients in Equation (34) from the single-scale coefficients in Equation (34). Conversely, the following reconstruction formula can be derived:

$$\alpha_{j,(a,c)}^{(u,v)} = \sum_{k,m,s,t} \{a_{a-2k}^{s,u} a_{c-2m}^{t,v} \alpha_{j-1,(k,m)}^{(s,t)} + a_{a-2k}^{s,u} b_{c-2m}^{t,v} \beta_{j-1,(k,m)}^{1,(s,t)} + b_{a-2k}^{s,u} a_{c-2m}^{t,v} \beta_{j-1,(k,m)}^{2,(s,t)} + b_{a-2k}^{s,u} b_{c-2m}^{t,v} \beta_{j-1,(k,m)}^{3,(s,t)}\} \tag{39}$$

Applied recursively, Equation (39) can be used to reconstruct the single-scale coefficients from the multiscale coefficients.

A series of analogous results can be derived for three-dimensional tensor-product scaling functions and multiwavelets. The principle remains the same but the notation necessarily becomes more unwieldy. A complete set of three-dimensional results will not be given here. Instead, we simply note that the three-dimensional functions take the form

$$\begin{aligned}
\Phi^{(s,t,w)}(x, y, z) &:= \phi^s(x)\phi^t(y)\phi^w(z) \\
\Psi^{1,(s,t,w)}(x, y, z) &:= \phi^s(x)\phi^t(y)\psi^w(z) \\
\Psi^{2,(s,t,w)}(x, y, z) &:= \phi^s(x)\psi^t(y)\phi^w(z) \\
\Psi^{3,(s,t,w)}(x, y, z) &:= \phi^s(x)\psi^t(y)\psi^w(z) \\
\Psi^{4,(s,t,w)}(x, y, z) &:= \psi^s(x)\phi^t(y)\phi^w(z) \\
\Psi^{5,(s,t,w)}(x, y, z) &:= \psi^s(x)\phi^t(y)\psi^w(z) \\
\Psi^{6,(s,t,w)}(x, y, z) &:= \psi^s(x)\psi^t(y)\phi^w(z) \\
\Psi^{7,(s,t,w)}(x, y, z) &:= \psi^s(x)\psi^t(y)\psi^w(z)
\end{aligned} \tag{40}$$

where $s, t, w \in \{1, \ldots, r\}$. Equation (40) shows that there are seven different types of three-dimensional wavelets. Therefore, for the three-dimensional case, there are a total of r^3 scaling functions and $7r^3$ wavelets. In general, for a given dimension n, there are r^n scaling functions and $(2^n - 1)r^n$ wavelets.

4. Multiwavelet Constructions via Intertwining

This paper utilizes multiwavelets that have been constructed from classical finite elements using the technique of intertwining. The basis functions over one element can be viewed as the generators of a finitely-generated, shift-invariant space V_0. This is depicted in Figure 1 for the quadratic finite element functions. In this case, the functions ϕ^1 and ϕ^2, supported over $[-1, 1]$, are the generators of the space V_0. The scaled translates and dilates of these two generators define a family of subspaces $\{V_j\}_{j\in\mathbb{Z}}$ that forms a multiresolution analysis in $L^2(\mathbb{R})$. Unfortunately, the multiresolution analyses generated directly from the classical finite elements are not orthogonal. Donovan et al. [20] demonstrated that orthogonal multiresolution analyses could be generated from the classical finite elements through the technique of intertwining. They further showed that, by performing a second intertwining, an additional degree of freedom can be introduced into the design process. This is essential for obtaining scaling functions and multiwavelets that exhibit desirable approximation properties from the lower-order finite elements.

The classical finite element basis functions of a given order generate a nonorthogonal multiresolution analysis $\{V_j\}_{j\in\mathbb{Z}}$. In general, the nth-order finite element yields n generators $\{\phi^1, \ldots, \phi^n\}$. In order

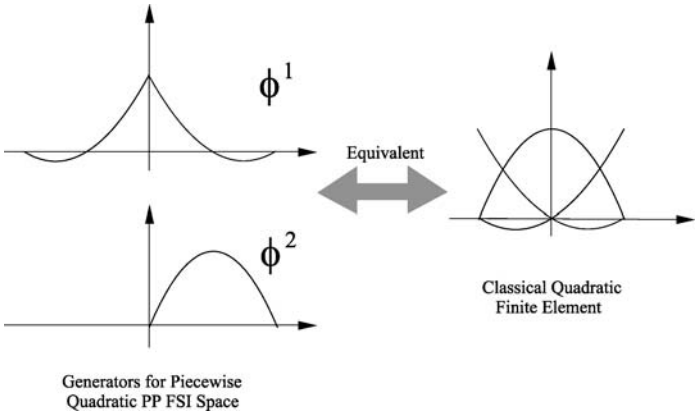

Figure 1. Classical quadratic finite element and associated finitely-generated shift-invariant (FSI) space.

to construct an intertwining multiresolution analysis from $\{V_j\}$, it is first required that the generators $\{\phi^1, \ldots, \phi^n\}$ be minimally supported in $[-1, 1]$.

Definition 1. [20] Suppose that $\{\phi^1, \ldots, \phi^n\}$ are generators of a multiresolution analysis, where $\{\phi^1, \ldots, \phi^k\}$ are supported on $[-1, 1]$ and the remaining $\{\phi^{k+1}, \ldots, \phi^n\}$ are supported on $[0, 1]$. The generators $\{\phi^1, \ldots, \phi^n\}$ are *minimally supported* on $[-1, 1]$ provided that

1) $\{\phi^1, \ldots, \phi^k\}$ are linearly independent on $[0, 1]$
2) $\{\phi^1, \ldots, \phi^k\}$ are linearly independent on $[-1, 0]$
3) $\{0\} = \operatorname{span}\big(\{\phi^1(x)\chi_{[0,1]}(x), \ldots, \phi^k(x)\chi_{[0,1]}(x)\}$
$\cap\{\phi^1(x)\chi_{[0,1]}(x-1), \ldots, \phi^k(x)\chi_{[0,1]}(x-1)\}\big)$

Note that $\chi_{[0,1]}$ denotes the characteristic function over $[0, 1]$, defined as

$$\chi_{[0,1]}(x) := \begin{cases} 1, & x \in [0, 1] \\ 0, & \text{otherwise} \end{cases} \tag{41}$$

The advantage of working with minimally-supported families of generators is that the restrictions of shifts of the generators $\{\phi^1, \ldots, \phi^n\}$ to any interval are linearly independent. That is,

$$\phi^1, \ldots, \phi^k, \phi^1(\cdot - 1), \ldots, \phi^k(\cdot - 1), \phi^{k+1}, \ldots, \phi^n$$

are all linearly independent on $[0, 1]$. The following spaces are defined:

$$A(V_0) := \operatorname{span}\{\phi^s : s = k+1, \ldots, n\} \tag{42}$$

$$B_\sigma(V_0) := \operatorname{span}\big(\{\phi^s(\cdot - \sigma)\chi_{[0,1]} : s = 1, \ldots, k\} \cup A(V_0)\big) \tag{43}$$

$$C_\sigma(V_0) := B_\sigma(V_0) \ominus A(V_0), \quad \sigma = 0, 1 \tag{44}$$

The process by which intertwining multiresolution analyses are constructed is defined in terms of the spaces $A(V_0)$, $B_\sigma(V_0)$, and $C_\sigma(V_0)$ defined above and the following lemma from Ref. [20]:

Lemma 2. [20] *Let $\{V_j\}_{j\in\mathbb{Z}}$ be a multiresolution analysis generated by n scaling functions $\{\phi^1, \ldots, \phi^n\}$ that are minimally supported on $[-1, 1]$. Suppose that there is a subspace $W \subset A(V_1) \ominus A(V_0)$ such that*

$$(I - P_W)C_0(V_0) \perp (I - P_W)C_1(V_0) \tag{45}$$

where P_W is the orthogonal projection onto W. Suppose that $\{w_1, \ldots, w_k\}$ is a basis for W. Then, the family of functions $\{\phi^1, \ldots, \phi^n,\ w_1, \ldots, w_k\}$ generates an orthogonal multiresolution analysis $\{\tilde{V}_j\}$ that intertwines $\{V_j\}$ in the sense that

$$\cdots V_0 \subset \tilde{V}_0 \subset V_1 \cdots \tag{46}$$

The method by which this lemma can be utilized to generate orthogonal multiwavelets is simple, at least in principle. The process starts by considering the multiresolution analysis $\{V_j\}$ that is generated by classical C_0 finite elements supported on $[-1, 1]$. A basis $\{w_1, \ldots, w_k\}$ for $A(V_1) \ominus A(V_0)$ is sought that satisfies Equation (45). If such a basis can be found, the desired orthogonal multiresolution analysis has been obtained. Unfortunately, in many cases of interest, the subspace $A(V_1) \ominus A(V_0)$ is not large enough to design multiwavelets that exhibit desirable properties such as symmetry or antisymmetry. This is the case when deriving multiwavelets from lower-order finite elements. As noted in Refs. [20, 21], this problem can be addressed by performing two successive intertwinings of multiresolution analyses. The second intertwining effectively lends another degree of freedom to the design process.

The starting point for the procedure is the finitely-generated shift-invariant (FSI) spaces $\{V_j\}_{j\in\mathbb{Z}}$ associated with the classical finite element spaces of a particular order. Given n generators $\{\phi^1, \ldots, \phi^n\}$, the FSI spaces are defined as

$$V_j := \operatorname{span}\left\{\phi^s_{j,k} : s \in \{1, \ldots, n\},\ k \in \mathbb{Z}\right\} \tag{47}$$

The first step in the intertwining process is to select a function $w \in V_1$. Then, a new set of generators is defined as

$$\{\tilde{\phi}^1, \ldots, \tilde{\phi}^{n+1}\} := \{\phi^1, \ldots, \phi^n, w\} \tag{48}$$

for a new multiresolution analysis $\{\tilde{V}_j\}_{j\in Z}$. This multiresolution analysis intertwines the original multiresolution analysis $\{V_j\}$ in the sense that

$$V_0 \subset \tilde{V}_0 \subset V_1 \subset \tilde{V}_1 \cdots \tag{49}$$

Note that $\{\tilde{V}_j\}$ is not constructed such that it is an orthogonal multiresolution analysis. If that were the case, the dimensionality of $A(V_1) \ominus A(V_0)$ would severely restrict the design of w, and hence of the multiresolution analysis $\{\tilde{V}_j\}$.

After the construction of $\{\tilde{V}_p\}$, a second intertwining is performed. That is, another function $\tilde{w} \in \tilde{V}_1$ is chosen and a new set of generators is defined:

$$\{\tilde{\tilde{\phi}}^1, \ldots, \tilde{\tilde{\phi}}^{n+2}\} := \{\tilde{\phi}^1, \ldots, \tilde{\phi}^{n+1}, \tilde{w}\} \tag{50}$$

In this case, $\tilde{w}$ is chosen to satisfy the conditions of Lemma 2. In other words, we seek $\tilde{w} \in A(\tilde{V}_1) \ominus A(\tilde{V}_0)$ such that

$$(I - P_{\tilde{W}})C_0(\tilde{V}_0) \perp (I - P_{\tilde{W}})C_1(\tilde{V}_0) \tag{51}$$

With the second intertwining, we have

$$\tilde{V}_0 \subset \tilde{\tilde{V}}_0 \subset \tilde{V}_1 \tag{52}$$

and $\{\tilde{\tilde{V}}_j\}$ is an orthogonal multiresolution analysis. In terms of the original multiresolution $\{V_j\}$, we have

$$\begin{aligned} &V_0 \subset \tilde{V}_0 \subset \tilde{\tilde{V}}_0 \\ &V_1 \subset \tilde{V}_1 \subset \tilde{\tilde{V}}_1 \\ &V_2 \subset \tilde{V}_2 \subset \tilde{\tilde{V}}_2 \\ &\quad\vdots \end{aligned} \tag{53}$$

Therefore, by performing two intertwinings of multiresolution analyses, an orthogonal multiresolution analysis can be obtained. Furthermore, the generators of the multiresolution analysis can be constructed so that they exhibit desirable properties such as symmetry or antisymmetry. The above procedure can be used to generate orthogonal multiresolution analyses from classical C_0 finite elements of arbitrary order. In general, with the exception of the linear functions, the nth-order finite element basis functions yield an orthogonal multiresolution analysis in terms of $r = n + 2$ generators $\{\phi^1, \ldots, \phi^r\}$. For convenience, once the final orthogonal generators have been obtained, the double tilde notation is dropped. The approximation spaces $\{V_j\}_{j\in\mathbb{Z}}$ that form the final, orthogonal multiresolution analysis are then defined as

$$V_j := \text{span}\left\{\phi^s_{j,k} : s \in \{1, \ldots, r\},\ k \in \mathbb{Z}\right\} \tag{54}$$

where we have the usual notation

$$\phi^s_{j,k}(x) = 2^{j/2}\phi^s(2^j - k) \tag{55}$$

The classical finite elements have the same general structure, regardless of the order. Over one element, there are exactly two functions that are nonzero at the boundaries. One takes the value of one at the left boundary and the other takes the value of one at the right boundary. The remaining basis functions are "interior", or zero at the boundaries. The number of interior functions increases with the order of the functions. When viewed as a shift-invariant space over $[-1, 1]$, the two boundary functions are combined into one symmetric function that takes the value of one at the midpoint. This is demonstrated in Figure 1 for the quadratic finite element. The remaining "interior" functions are supported over $[0, 1]$. Therefore, regardless of the order, the final set of r orthogonal generators is composed of one symmetric function ϕ^1, supported over $[-1, 1]$, and $r - 1$ interior functions over $[0, 1]$.

Once a set of orthogonal generators $\{\phi^1, \ldots, \phi^r\}$ has been constructed, it remains to derive an associated set of multiwavelets $\{\psi^1, \ldots, \psi^r\}$. These multiwavelets define the space W_0, which is the orthogonal complement to V_0:

$$W_0 = V_1 \ominus V_0 \tag{56}$$

Therefore, the multiwavelets must be composed of a linear combination of the scaling functions that span V_1. Furthermore, since W_0 is the orthogonal complement of V_0, the multiwavelets must be orthogonal to the scaling functions $\{\phi^1, \ldots, \phi^r\}$ that span V_0. First, a symmetric wavelet supported over $[-1, 1]$ can be constructed as

$$\psi^1(x) = \phi^1_{1,0}(x) - \langle \phi^1_{1,0}, \phi^1 \rangle \phi^1(x) \tag{57}$$

Note that the Gram–Schmidt procedure has been used to construct ψ^1 such that it is orthogonal to ϕ^1. By definition in Equation (55), the function $\phi^1_{1,0}$ is a scaled and contracted form of ϕ^1, also symmetric about the origin and supported over $[-\frac{1}{2}, \frac{1}{2}]$. Note that, because the remaining scaling functions $\{\phi^2, \ldots, \phi^r\}$ are supported over $[0, 1]$, there is no component of $\phi^1_{1,0}$ in these functions. Consequently, they are orthogonal to $\phi^1_{1,0}$. The symmetric wavelet ψ^1 is then normalized.

Next, an antisymmetric wavelet that is supported on $[-1, 1]$ is constructed. We start with a symmetric function

$$z(x) = \phi^1(x) - \langle \phi^1, \phi^1_{1,0} \rangle \phi^1_{1,0}(x) \tag{58}$$

The contribution of $\phi^1_{1,0}$ is removed from ϕ^1 in order to obtain a function that passes through the origin. The function z is made antisymmetric by multiplying by the function H, defined as

$$H(x) = \begin{cases} -1, & x \in [-1, 0] \\ 1, & x \in [0, 1] \\ 0, & \text{otherwise} \end{cases} \tag{59}$$

We obtain

$$\psi^2(x) = H(x)z(x) = H(x)\left[\phi^1(x) - \langle \phi^1, \phi^1_{1,0} \rangle \phi^1_{1,0}(x)\right] \tag{60}$$

Since ψ^2 is an antisymmetric function, it is guaranteed to be orthogonal to ϕ^1 and ψ^1. By construction, the translates of the functions ψ^2, ψ^1, and ϕ^1 are all mutually orthogonal. Clearly, from Equation (58), ψ^2 is also orthogonal to the remaining scaling functions $\{\phi^2, \ldots, \phi^r\}$ since both ϕ^1 and $\phi^1_{1,0}$ are orthogonal to these functions.

In order to complete the set of wavelets, $r - 2$ wavelets with support over $[0, 1]$ are constructed. Recall that these wavelets must be formed as linear combinations of the scaling functions that span V_1 and have support contained in $[0, 1]$. This implies that the remaining wavelets take the form

$$\psi^t(x) = c\phi^1_{1,1}(x) + \sum_{i=2}^{r} d_i \phi^i_{1,0}(x) + \sum_{i=2}^{r} e_i \phi^i_{1,1}(x), \quad t = 3, \ldots, r \tag{61}$$

In other words, the multiwavelets $\{\psi^3, \ldots, \psi^r\}$ are formed as linear combinations of only those scaling functions in V_1 whose support is contained in $[0, 1]$. In Equation (61), there are a total of $2r - 1$ constants that must be determined for each multiwavelet. In addition, the multiwavelets are required to be orthogonal to all translates of the scaling functions $\{\phi^1, \ldots, \phi^r\}$. This requirement provides $r + 1$ orthogonality constraints for each multiwavelet:

$$\begin{aligned} &\langle \psi^t, \phi^s \rangle = 0, \quad s = 1, \ldots, r \\ &\langle \psi^t, \phi^1(\cdot - 1) \rangle = 0 \end{aligned} \tag{62}$$

Note that since there are $2r - 1$ unknown constants and only $r + 1$ constraints, there is some freedom in the design of each interior wavelet as long as it is ensured that they are all linearly independent. As a final step, the Gram–Schmidt procedure is used to make the interior wavelets mutually orthogonal.

We then have a set of orthonormal wavelets $\{\psi^1, \ldots, \psi^r\}$ whose translates form a basis for the orthogonal complement space W_0. The wavelet spaces $\{W_j\}$ are defined as

$$W_j := \text{span}\left\{\psi^s_{j,k} : \ s \in \{1, \ldots, r\}, k \in \mathbb{Z}\right\} \tag{63}$$

where

$$\psi^s_{j,k}(x) = 2^{j/2}\psi^s(2^j x - k) \tag{64}$$

The procedure for constructing the wavelets is the same regardless of the order of the functions. In each case, there is exactly one symmetric wavelet ψ^1 supported over $[-1, 1]$ and exactly one antisymmetric wavelet ψ^2 supported over $[-1, 1]$. There are typically $r - 2$ interior wavelets $\{\psi^3, \ldots, \psi^r\}$ where, once again, $n = r - 2$ is the order of the original finite element.

Donovan et al. [20] have derived piecewise-linear multiwavelets from the classical linear finite elements using this procedure. The authors have applied this technique to construct piecewise-quadratic and piecewise-cubic multiwavelets in order to obtain smoother functions for Volterra kernel estimation. The piecewise-quadratic scaling functions and multiwavelets are shown in Figures 2 and 3, respectively. The corresponding scaling function and multiwavelet filter matrices are listed in Tables 1 and 2. As will be discussed in the following section, this family of piecewise-polynomial multiwavelets is particularly well suited for the representation of Volterra kernels because the functions can easily be adapted to the domains of support of the kernels.

5. Kernel Identification with Multiwavelets

5.1. First-Order Kernel

Recall that, for a causal, time-invariant system, the first-order Volterra operator takes the form

$$y_1(t) = \int_0^t h_1(\xi)u(t - \xi)\, d\xi \tag{65}$$

Table 1. Piecewise-quadratic scaling function filter matrices.

n	$[a_n]$
-2	$\begin{bmatrix} 0 & .0163446406689 & -.0342630216987 & -.0327571725062 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$
-1	$\begin{bmatrix} .113474174186 & .0163446406689 & -.422172442890 & -.236881317738 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$
0	$\begin{bmatrix} .707106781187 & .0163446406689 & .422172442890 & -.236881317738 \\ 0 & .618718433538 & -.220970869121 & -.0988211768803 \\ 0 & .707106781187 & 0 & 0 \\ 0 & -.338665211381 & -.316332942563 & -.201862521835 \end{bmatrix}$
1	$\begin{bmatrix} .113474174186 & .0163446406689 & .0342630216987 & -.0327571725062 \\ .342326598441 & .618718433538 & .220970869121 & -.0988211768803 \\ 0 & -.707106781187 & 0 & 0 \\ .699272287923 & -.338665211381 & .316332942563 & -.201862521835 \end{bmatrix}$

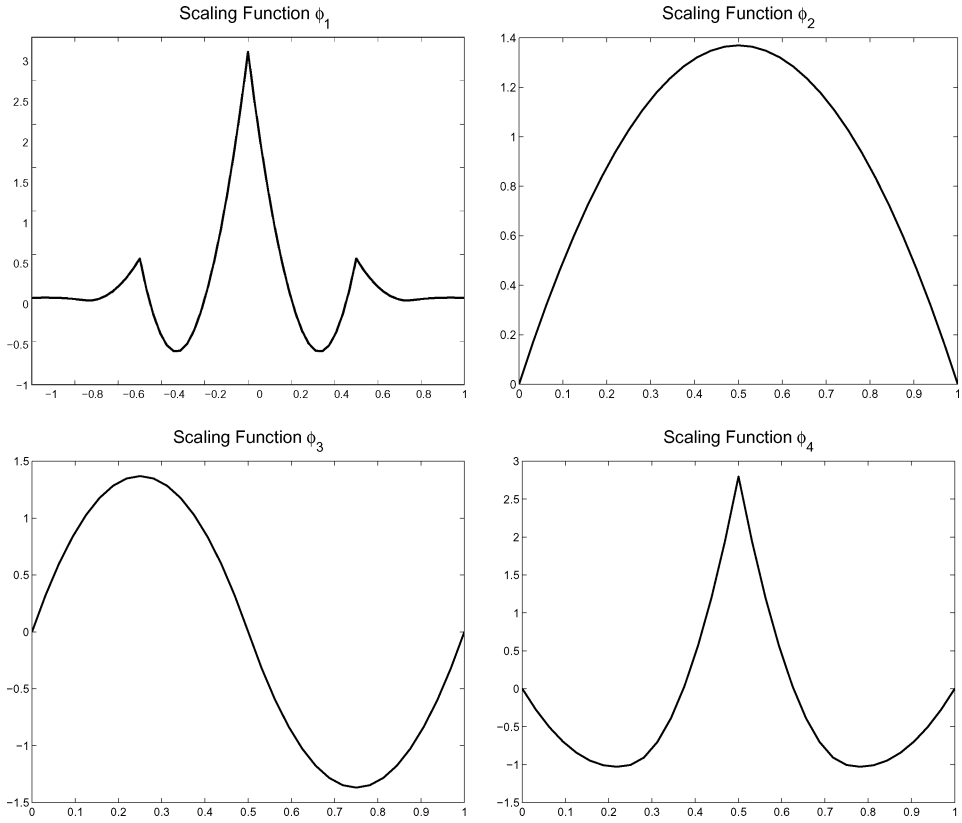

Figure 2. Piecewise-quadratic scaling functions.

Table 2. Piecewise-quadratic multiwavelet filter matrices.

n	$[b_n]$
-2	$\begin{bmatrix} 0 & -.0163446406689 & .0342630216987 & .0327571725062 \\ 0 & -.0231148125061 & .0484552299729 & .0463256376237 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$
-1	$\begin{bmatrix} -.113474174186 & -.0163446406689 & .422172442890 & .236881317738 \\ -.160476716113 & -.0231148125061 & .597041994395 & .335000772219 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$
0	$\begin{bmatrix} .707106781187 & -.0163446406689 & -.422172442890 & .236881317738 \\ 0 & .0231148125061 & .597041994395 & -.335000772219 \\ 0 & 0 & -.288675134594 & -.645497224366 \\ 0 & .0188731653801 & .223959608749 & .551430926265 \end{bmatrix}$
1	$\begin{bmatrix} -.113474174186 & -.0163446406689 & -.0342630216987 & .0327571725062 \\ .160476716113 & .0231148125061 & .0484552299729 & -.0463256376237 \\ 0 & 0 & -.288675134594 & .645497224366 \\ .539276980493 & .0188731653801 & -.223959608749 & .551430926265 \end{bmatrix}$

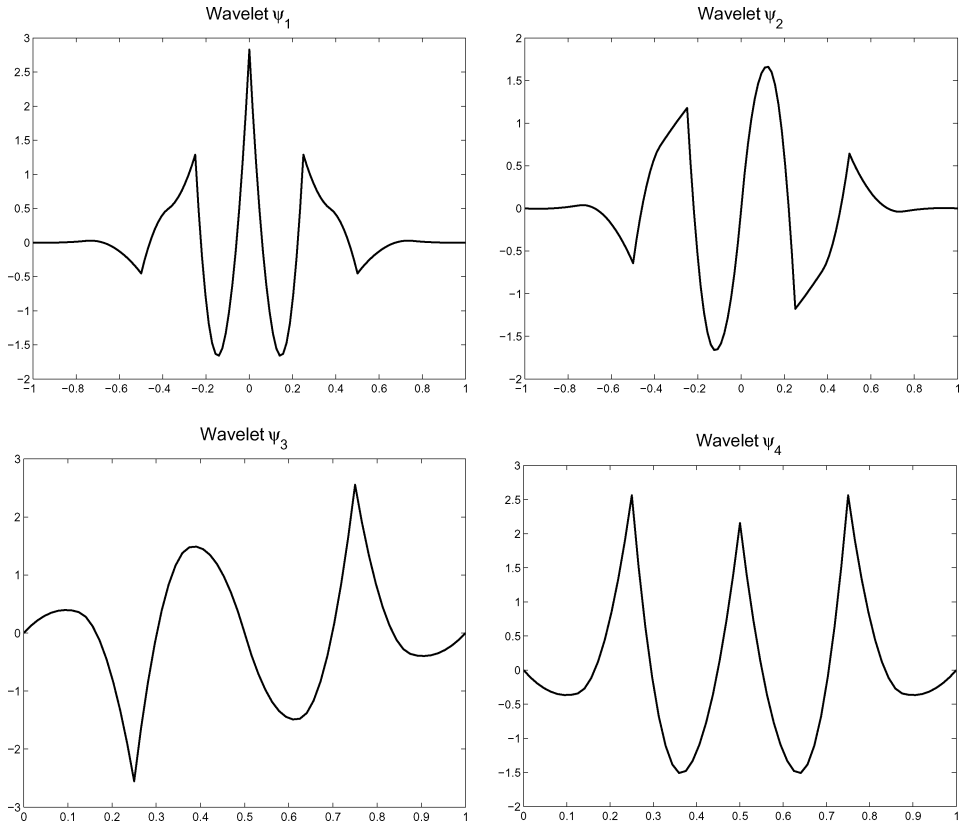

Figure 3. Piecewise-quadratic multiwavelets.

In order to derive a discrete form of Equation (65), the input u is first approximated using a zero-order hold. If the input and output are sampled at a rate of 2^j Hz and there are a total of N data points, a zero-order hold approximation of the input can be written as

$$u_j(t) = \sum_{k=0}^{N-1} u_{j,k}\chi_{j,k}(t) \tag{66}$$

where the characteristic function $\chi_{j,k}$ is defined as

$$\chi_{j,k}(t) = 2^{j/2}\chi(2^j t - k) = \begin{cases} 2^{j/2}, & t \in [2^{-j}k, 2^{-j}(k+1)] \\ 0, & \text{otherwise} \end{cases} \tag{67}$$

and χ is the characteristic function of the interval [0, 1]. The zero-order hold is simply a piecewise-constant approximation of the input. The coefficients $\{u_{j,k}\}$ are equivalent to scaled samples of the input:

$$u_{j,k} = 2^{-j/2}u(2^{-j}k), \qquad k = 0, \ldots, N-1 \tag{68}$$

Discretizing the output at a sampling rate of 2^j Hz, Equation (65) yields N equations for the discrete first-order outputs:

$$y_{1,j}(t_n) = \int_0^{t_n} h_1(\xi)u(t_n - \xi)\,d\xi, \qquad n = 1, \ldots, N \tag{69}$$

where $t_n = 2^{-j}n$. As shown in Ref. [17], the translated zero-order hold approximation of the input can be written as

$$u_j(t_n - \xi) = \sum_{k=0}^{n-1} u_{j,n-k-1}\chi_{j,k}(\xi) \tag{70}$$

Next, the first-order kernel h_1 is represented in terms of the piecewise-polynomial multiwavelets derived in the previous section. Recall that, under the assumption of fading memory, the Volterra kernels of a system decay to zero in a finite period of time. Then, the first-order kernel is zero after some finite time T_1 and is supported over the domain $[0, T_1]$. In order to use the piecewise-polynomial multiwavelets to represent the first-order kernel, it is first necessary to adapt the functions to the domain $[0, T_1]$. For example, consider Figure 4, which depicts all of the piecewise-linear scaling functions on level 0 that intersect the domain [0, 1]. It is clear that the scaling function ϕ^1 intersects the left boundary while the translated scaling function $\phi^1(\cdot - 1)$ intersects the right boundary. These functions can be modified to fit the domain [0, 1] by simply truncating the functions at the boundaries. Note that no boundary modifications are needed for the interior scaling functions $\{\phi^2, \phi^3, \phi^4\}$. In general, then, given r piecewise-polynomial scaling functions, the boundary-adapted approximation spaces $\{V_j[0, T_1]\}$ are defined as

$$V_j[0, T_1] := \operatorname{span}\left\{\phi^1_{j,0}\chi_{[0,T_1]}, \phi^1_{j,k}\Big|_{k=1}^{2^jT_1-1}, \phi^2_{j,k}\Big|_{k=0}^{2^jT_1-1}, \ldots, \phi^r_{j,k}\Big|_{k=0}^{2^jT_1-1}, \phi^1_{j,2^jT_1}\chi_{[0,T_1]}\right\} \tag{71}$$

where $\phi^1_{j,0}\chi_{[0,T_1]}$ and $\phi^1_{j,2^jT_1}\chi_{[0,T_1]}$ denote the restrictions of the functions $\phi^1_{j,0}$ and $\phi^1_{j,2^jT_1}$, which overlap the boundary, to $[0, T_1]$. In other words, the basis for $V_j[0, T_1]$ is composed of all level j scaling

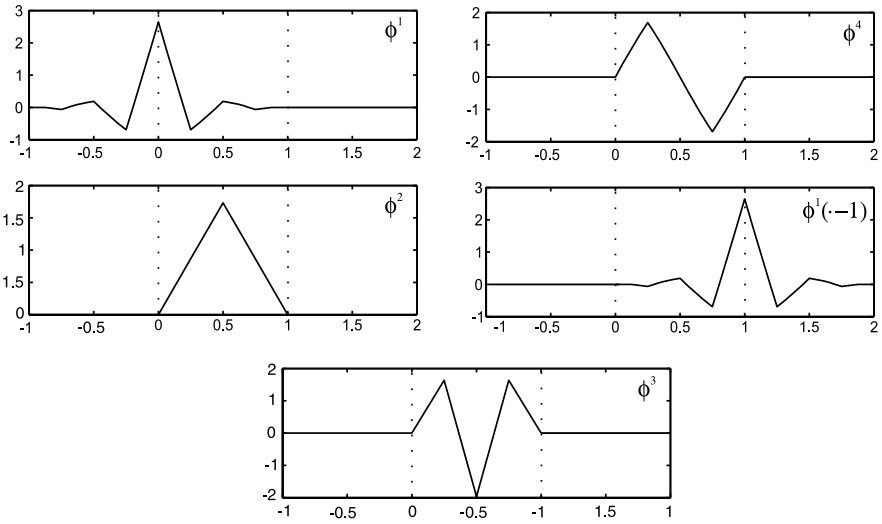

Figure 4. Piecewise-linear scaling functions that intersect [0, 1].

functions that intersect $[0, T_1]$. The two scaling functions that are nonzero at the boundaries are simply truncated to fit the domain. Therefore, referring to Equation (71), the space $V_j[0, T_1]$ is spanned by a total of $2^j T_1 r + 1$ scaling functions.

The boundary modifications for the wavelets are not quite as simple. In general, translates of the symmetric wavelet ψ^1 and the antisymmetric wavelet ψ^2 intersect both boundaries. Given r piecewise-polynomial multiwavelets, the boundary-adapted wavelet spaces $\{W_j[0, T_1]\}$ are defined as

$$W_j[0, T_1] := \operatorname{span}\Big\{\psi^1_{j,0}\chi_{[0,T_1]}, \psi^1_{j,k}\big|_{k=1}^{2^j T_1 - 1}, \psi^2_{j,k}\big|_{k=1}^{2^j T_1 - 1}, \psi^3_{j,k}\big|_{k=0}^{2^j T_1 - 1}, \ldots, \psi^r_{j,k}\big|_{k=0}^{2^j T_1 - 1}, \psi^1_{j,2^j T_1}\chi_{[0,T_1]}\Big\} \tag{72}$$

Similar to the approximation spaces $\{V_j[0, T_1]\}$, the symmetric wavelets $\psi^1_{j,0}$ and $\psi^1_{j,2^j T_1}$ that intersect the boundaries are simply restricted to $[0, T_1]$. Note, however, that the restrictions to $[0, T_1]$ of the antisymmetric wavelets $\psi^2_{j,0}$ and $\psi^2_{j,2^j T_1}$, which also intersect the boundaries, are not included in the wavelet basis. This is because, when restricted to $[0, T_1]$, these functions are linearly dependent on the other wavelet basis functions. Therefore, each wavelet space $W_j[0, T_1]$ is defined in terms of $2^j T_1 r$ wavelets. A more rigorous discussion of the boundary modifications of the piecewise-polynomial scaling functions and multiwavelets can be found in Ref. [21].

The first-order kernel can be approximated in terms of the boundary-adapted scaling functions on level j_1 as

$$h_{1,j_1}(\xi) = \sum_p \sum_{s=1}^{r} \alpha^s_{j_1,p}\phi^s_{j_1,p}(\xi) \tag{73}$$

Note that, due to computational considerations, the discretization level j_1 for the kernel is usually chosen to be coarser than the input/output discretization level j. As in Equation (71), the expansion in Equation (73) is in terms of $N_1 = 2^j T_1 r + 1$ scaling functions. Substituting Equations (70) and (73)

into Equation (69), the discrete first-order outputs are given by

$$y_{1,j}(t_n) = \sum_{k=0}^{n_1-1} \sum_{p} \sum_{s=1}^{r} u_{j,n-k-1} \alpha_{j_1,p} \int_0^{T_1} \phi^s_{j_1,p}(\xi) \chi_{j,k}(\xi)\, d\xi \tag{74}$$

where the integer n_1 is defined as

$$n_1 = \begin{cases} n, & n < 2^j T_1 \\ 2^j T_1, & n \geq 2^j T_1 \end{cases} \tag{75}$$

When $n \geq 2^j T_1$, the upper limit is due to the fact that the first-order kernel has been truncated at time T_1. Equation (74) can be written in matrix form as

$$\underline{y}_{1,j} = [M_1]\underline{\alpha}_1 \tag{76}$$

where $\underline{y}_{1,j}$ is a vector of discrete first-order outputs and $\underline{\alpha}_1$ is a vector of scaling function coefficients that represent the first-order kernel. The entries of the $N \times N_1$ matrix $[M_1]$ are calculated using Equation (74), where the integrals are evaluated using Gauss–Legendre quadrature.

Recall that a multiscale representation of the kernel, equivalent to the single-scale expansion in Equation (73), takes the form

$$h_{1,j_1}(\xi) = \sum_{p} \sum_{s=1}^{r} \alpha^s_{j_0,p} \phi^s_{j_0,p}(\xi) + \sum_{l=j_0}^{j_1-1} \sum_{p} \sum_{s=1}^{r} \beta^s_{l,p} \psi^s_{l,p}(\xi) \tag{77}$$

where j_0 is the coarsest level chosen in the wavelet decomposition. Denoting $\underline{\beta}_1$ as the vector of multiscale kernel coefficients, and defining $[T_1]$ as the invertible matrix that performs the discrete wavelet transform, Equation (76) can be written as

$$\underline{y}_{1,j} = [M_1][T_1]^{-1}\underline{\beta}_1 \tag{78}$$

The advantage of the expression in Equation (78) is that the first-order kernel is represented in terms of a structured multiscale vector of wavelet coefficients that is amenable to order reduction. Often, the coefficients on the finer-resolution scales contribute little to the overall structure of the kernels and can be neglected. This is achieved by simply truncating the vector $\underline{\beta}_1$, resulting in fewer kernel coefficients that must be identified. Then, a reduced-order model of the first-order Volterra operator takes the form

$$\underline{y}_{1,j} = [M_1][\hat{T}_1]^{-1}\hat{\underline{\beta}}_1 = [A_1]\hat{\underline{\beta}}_1 \tag{79}$$

where $\hat{\underline{\beta}}_1$ denotes the first $\hat{N}_1 < N_1$ terms in $\underline{\beta}_1$ and $[\hat{T}_1]^{-1}$ represents the first $\hat{N}_1$ columns of $[T_1]^{-1}$.

5.2. Second-Order Kernel

The second-order Volterra operator takes the form

$$y_2(t) = \int_0^t \int_0^t h_2(\xi, \eta) u(t-\xi) u(t-\eta)\, d\xi\, d\eta \tag{80}$$

where h_2 denotes the symmetric form of the second-order kernel. The second-order kernel can be expressed as a linear combination of two-dimensional tensor product scaling functions as

$$\begin{aligned} h_{2,j_2}(\xi,\eta) &= \sum_{p,q}\sum_{s,v=1}^{r} \alpha^{(s,v)}_{j_2,(p,q)} \Phi^{(s,v)}_{j_2,(p,q)}(\xi,\eta) \\ &= \sum_{p,q}\sum_{s,v=1}^{r} \alpha^{(s,v)}_{j_2,(p,q)} \phi^{s}_{j_2,p}(\xi)\phi^{v}_{j_2,q}(\eta) \end{aligned} \tag{81}$$

where, for computational reasons, j_2 is usually chosen to be a coarser discretization level than the input/output discretization. Due to the assumption of fading memory, the second-order kernel is assumed to decay to zero after some finite time T_2. Then, the symmetric second-order kernel is supported on the $[0, T_2]\times[0, T_2]$ square domain. The boundary-adapted scaling functions and wavelets in two dimensions are obtained as the tensor products of the one-dimensional scaling functions and wavelets that have been adapted to the domain $[0, T_2]$. There are a total of $N_2 = 2^{j_2}T_2 r + 1$ scaling functions on resolution level j_2 that have been adapted to $[0, T_2]$. Therefore, the single-scale expansion of the second-order kernel in Equation (81) is in terms of N_2^2 scaling functions.

Substituting the single-scale representation of the kernel and the zero-order hold approximation of the input into Equation (80), we obtain N equations for the discrete second-order outputs:

$$\begin{aligned} y_{2,j}(t_n) = &\sum_{k,m=0}^{n_2-1}\sum_{p,q}\sum_{s,v=1}^{r} u_{j,n-k-1}u_{j,n-m-1}\alpha^{(s,v)}_{j_2,(p,q)} \cdots \\ &\int_0^{T_2}\int_0^{T_2} \Phi^{(s,v)}_{j_2,(p,q)}(\xi,\eta)\chi_{j,(k,m)}(\xi,\eta)\,d\xi\,d\eta \end{aligned} \tag{82}$$

for $n = 1, \ldots, N$. The upper limit in the summations over k and m is given by

$$n_2 = \begin{cases} n, & n < 2^j T_2 \\ 2^j T_2, & n \geq 2^j T_2 \end{cases} \tag{83}$$

Equation (82) can be written in the matrix form

$$\underline{y}_{2,j} = [M_2]\underline{\alpha}_2 \tag{84}$$

where $\underline{y}_{2,j}$ is a vector of discrete second-order outputs and $\underline{\alpha}_2$ is a vector of single-scale second-order kernel coefficients.

A multiscale representation of the second-order kernel, equivalent to the single-scale expansion in Equation (81), can be written as

$$\begin{aligned} h_{2,j_2}(\xi,\eta) = &\sum_{p,q}\sum_{s,v=1}^{r} \alpha^{(s,v)}_{j_0,(p,q)} \phi^{s}_{j_0,p}(\xi)\phi^{v}_{j_0,q}(\eta) \\ &+ \sum_{l=j_0}^{j_2-1}\sum_{p,q}\sum_{s,v=1}^{r} \beta^{1,(s,v)}_{l,(p,q)} \phi^{s}_{l,p}(\xi)\psi^{v}_{l,q}(\eta) \\ &+ \sum_{l=j_0}^{j_2-1}\sum_{p,q}\sum_{s,v=1}^{r} \beta^{2,(s,v)}_{l,(p,q)} \psi^{s}_{l,p}(\xi)\phi^{v}_{l,q}(\eta) \\ &+ \sum_{l=j_0}^{j_2-1}\sum_{p,q}\sum_{s,v=1}^{r} \beta^{3,(s,v)}_{l,(p,q)} \psi^{s}_{l,p}(\xi)\psi^{v}_{l,q}(\eta) \end{aligned} \tag{85}$$

Similar to the first-order case, it is possible to construct an invertible matrix $[T_2]$ that decomposes the vector of single-scale coefficients $\underline{\alpha}_{j_2}$ into a vector of multiscale coefficients. Then, Equation (84) can be written as

$$\underline{y}_{2,j} = [M_2][T_2]^{-1}\underline{\beta}_2 \tag{86}$$

where $\underline{\beta}_2$ is defined as the vector of multiscale coefficients. Unfortunately, the construction of the transformation matrix $[T_2]$ is prohibitively expensive and it is not practical to formulate the model as in Equation (86). Instead, low-order representations of the second-order kernel are obtained by selecting a relatively coarse discretization level j_2 in Equation (82). Of course, the multiscale representation in Equation (85) is still valid, but the multiscale coefficients do not explicitly appear in the model.

The number of unknown coefficients in Equation (84) can be significantly reduced by taking advantage of the symmetry of the second-order kernel. The symmetry of the kernel implies that the number of unique coefficients can be reduced from N_2^2 to $\tilde{N}_2 = (N_2^2 + N_2)/2$. Therefore, an $\tilde{N}_2$-dimensional vector of unique second-order kernel coefficients can be defined as $\underline{\hat{\alpha}}_2$. This vector can be related to $\underline{\alpha}_2$ by a transformation matrix $[P_2]$:

$$\underline{\alpha}_2 = [P_2]\underline{\hat{\alpha}}_2 \tag{87}$$

Then, the discrete second-order Volterra model can be written in the form

$$\underline{y}_{2,j} = [M_2][P_2]\underline{\hat{\alpha}}_2 = [A_2]\underline{\hat{\alpha}}_2 \tag{88}$$

As a final note, $[P_2]$ is an extremely sparse matrix. All entries are zero except for a single entry in each row that takes the value of one. In general, it is more efficient to form the matrix $[A_2]$ directly. Therefore, $[P_2]$ is not formed explicitly in the algorithm and has only been introduced for notational convenience.

5.3. Third-Order Kernel

The third-order Volterra operator is given by

$$y_3(t) = \int_0^t \int_0^t \int_0^t h_3(\xi, \eta, \gamma)u(t-\xi)u(t-\eta)u(t-\gamma)\, d\xi\, d\eta\, d\gamma \tag{89}$$

where h_3 is the symmetric third-order kernel. A single-scale representation of the third-order kernel can be written as

$$\begin{aligned} h_{3,j_3}(\xi, \eta, \gamma) &= \sum_{p,q,z} \sum_{s,v,w=1}^{r} \alpha^{(s,v,w)}_{j_3,(p,q,z)} \Phi^{(s,v,w)}_{j_3,(p,q,z)}(\xi, \eta, \gamma) \\ &= \sum_{p,q,z} \sum_{s,v,w=1}^{r} \alpha^{(s,v,w)}_{j_3,(p,q,z)} \phi^{s}_{j_3,p}(\xi)\phi^{v}_{j_3,q}(\eta)\phi^{w}_{j_3,z}(\gamma) \end{aligned} \tag{90}$$

Due to computational considerations, the discretization level j_3 is usually chosen to be much coarser than the input/output discretization. The third-order kernel is assumed to decay to zero after some finite time T_3. Then, the symmetric third-order kernel is supported on a $[0, T_3] \times [0, T_3] \times [0, T_3]$ three-dimensional domain. The three-dimensional scaling functions must be adapted to this domain.

The boundary-adapted functions are obtained as the tensor products of the one-dimensional scaling functions that have been adapted to the domain $[0, T_3]$. There are a total of $N_3 = 2^{j_3} T_3 r + 1$ level-j_3 scaling functions with support in $[0, T_3]$. Therefore, the single-scale expansion of the third-order kernel in Equation (90) is in terms of N_3^3 scaling functions.

The zero-order hold approximation of the input in Equation (70) and the single-scale representation of the kernel in Equation (90) are substituted into Equation (89). This yields N equations for the discrete third-order outputs:

$$y_{3,j}(t_n) = \sum_{k,m,f=0}^{n_3-1} \sum_{p,q,z} \sum_{s,v,w=1}^{r} u_{j,n-k-1} u_{j,n-m-1} u_{j,n-f-1} \alpha_{j_3,(p,q,z)}^{(s,v,w)} \cdots \int_0^{T_3} \int_0^{T_3} \int_0^{T_3} \Phi_{j_3,(p,q,z)}^{(s,v,w)}(\xi, \eta, \gamma) \chi_{j,(k,m,f)}(\xi, \eta, \gamma)\, d\xi\, d\eta\, d\gamma \tag{91}$$

for $n = 1, \ldots, N$. The upper limit in the summations over k, m, and f is given by

$$n_3 = \begin{cases} n, & n < 2^j T_3 \\ 2^j T_3, & n \geq 2^j T_3 \end{cases} \tag{92}$$

Equation (91) can be written in the matrix form

$$\underline{y}_{3,j} = [M_3]\underline{\alpha}_3 \tag{93}$$

where $\underline{y}_{3,j}$ is a vector of discrete third-order outputs and $\underline{\alpha}_3$ is a vector of third-order kernel coefficients.

As usual, there is an equivalent multilevel representation of the third-order kernel. In theory, an invertible matrix $[T_3]$ can be formed that decomposes the single-scale coefficients into the vector of multiscale coefficients. Just as in the second-order case, however, it is not practical to form this matrix. Therefore, low-order approximations of the third-order kernel are obtained by choosing an appropriately coarse resolution level j_3 in Equation (90). Therefore, the multiscale kernel coefficients do not appear explicitly in the discrete third-order model.

The symmetry of the third-order kernel can be exploited to reduce the number of unknown coefficients by roughly a factor of six. More specifically, the number of unknowns can be reduced from N_3^3 to $\tilde{N}_3 = (N_3^3 + 3N_3^2 + N_3)/6$ unique coefficients. Then, similar to the second-order case, we have

$$\underline{\alpha}_3 = [P_3]\underline{\hat{\alpha}}_3 \tag{94}$$

where $\underline{\hat{\alpha}}_3$ is a vector of unique third-order kernel coefficients. Then, the discrete third-order Volterra model can be written in the form

$$\underline{y}_{3,j} = [M_3][P_3]\underline{\hat{\alpha}}_3 = [A_3]\underline{\hat{\alpha}}_3 \tag{95}$$

The sparse transformation matrix $[P_3]$ is not formed in practice since it is computationally more efficient to form the matrix $[A_3]$ directly.

5.4. Total Volterra Model

In this paper, it is assumed that the output can be expressed in terms of a truncated Volterra series that includes only the first-, second-, and third-order Volterra operators. Then, the vector of discrete outputs $\underline{y}_j$ can be written as a sum of discrete first-, second-, and third-order outputs:

$$\underline{y}_j = \underline{y}_{1,j} + \underline{y}_{2,j} + \underline{y}_{3,j} \tag{96}$$

Using Equations (79), (88), and (95), we obtain the model

$$\underline{y}_j = [A_1 \quad A_2 \quad A_3] \begin{Bmatrix} \underline{\hat{\beta}}_1 \\ \underline{\hat{\alpha}}_2 \\ \underline{\hat{\alpha}}_3 \end{Bmatrix} = [A]\underline{\beta} \tag{97}$$

Therefore, the kernel identification problem takes the form of a linear least-squares problem. Equation (97) can then be solved, in a least-squares sense, for the coefficients that represent the first-, second-, and third-order kernels. The goal is to obtain accurate kernel estimates in terms of as few coefficients as possible. It should be noted that this problem is ill-posed in that the objective is to determine the structure of the system from input/output measurements. Therefore, in order to obtain stable kernel estimates, a regularization technique must be used to solve the least-squares problem. There are many regularization methods discussed in the literature [26, 27]. In this paper, the truncated singular value decomposition is used to compute the least-squares solution of Equation (97).

6. Nonlinear Oscillator Example

The wavelet-based kernel identification approach detailed in the previous section is now demonstrated on a prototypical nonlinear dynamical system. The system is chosen to be a nonlinear oscillator having the following equation of motion:

$$m\ddot{y}(t) + c\dot{y}(t) + k_1 y(t) + k_2 y^2(t) = u(t) \tag{98}$$

where m is the mass, c is the damping coefficient, and k_1 is the linear stiffness coefficient. Equation (98) is nonlinear due to the $k_2 y^2(t)$ term, which represents a quadratic spring stiffness. The system parameters are chosen as

$$\begin{aligned} m &= 1\ \text{kg} \\ c &= 6\ \text{N s/m} \\ k_1 &= 4\pi^2\ \text{N/m} \\ k_2 &= 4\pi^2\ \text{N/m}^2 \end{aligned} \tag{99}$$

Given the equation of motion, it is possible to derive analytical expressions for the Volterra kernels in the frequency domain using the harmonic balancing technique. The harmonic balance method is discussed in detail in Ref. [28] and the procedure is carried out for the governing equations of a nonlinear aeroelastic system in Ref. [3]. Therefore, the details of the harmonic balancing approach will

not be given here. The procedure, applied to Equation (98), yields the following expressions for the first-, second-, and third-order kernels in the Laplace domain:

$$H_1(s_1) = \frac{1}{ms_1^2 + cs_1 + k_1} \tag{100}$$

$$H_2(s_1, s_2) = -k_2 H_1(s_1) H_1(s_2) H_1(s_1 + s_2) \tag{101}$$

$$\begin{aligned} H_3(s_1, s_2, s_3) &= -2k_2 H_1(s_1) H_2(s_2, s_3) H_1(s_1 + s_2 + s_3) \\ &= 2k_2^2 H_1(s_1) H_1(s_2) H_1(s_3) H_1(s_2 + s_3) H_1(s_1 + s_2 + s_3) \end{aligned} \tag{102}$$

Linear- and higher-order frequency response functions are obtained by substituting $\{j\omega_1, j\omega_2, j\omega_3\}$ for $\{s_1, s_2, s_3\}$ in the above expressions. It is interesting to note that the higher-order transfer functions H_2 and H_3 can be expressed in terms of the linear transfer function H_1. This is true in general for systems with polynomial nonlinearities. Also of interest is the fact that the linear transfer function is equivalent to the transfer function of the linear system obtained by deleting the nonlinear term in Equation (98):

$$m\ddot{y}(t) + c\dot{y}(t) + k_1 y(t) = u(t) \tag{103}$$

This is another general property of systems with polynomial nonlinearities.

Time-domain Volterra kernels are obtained by taking inverse Laplace transforms of the expressions in Equations (100) through (102). The first-order kernel is equivalent to the impulse response of Equation (103), a classical result from vibration theory:

$$h_1(t) = \frac{1}{m\omega_\mathrm{d}} e^{-\zeta\omega_\mathrm{n} t} \sin(\omega_\mathrm{d} t) \tag{104}$$

The damping ratio ζ, the natural frequency ω_n, and the damped natural frequency ω_d are defined as

$$\zeta = \frac{c}{2\sqrt{k_1 m}} \tag{105}$$

$$\omega_\mathrm{n} = \sqrt{\frac{k_1}{m}} \tag{106}$$

$$\omega_\mathrm{d} = \omega_n \sqrt{1 - \zeta^2} \tag{107}$$

For the values chosen in Equation (99), these parameters are given as:

$$\begin{aligned} \zeta &= .478 \\ \omega_\mathrm{n} &= 2\pi \text{ rad/s} = 1 \text{ Hz} \\ \omega_\mathrm{d} &= 5.52 \text{ rad/s} = .879 \text{ Hz} \end{aligned} \tag{108}$$

The time-domain forms of the second- and third-order kernels were obtained by taking multidimensional Laplace transforms of Equations (101) and (102). This calculation was performed in Mathematica. The analytical expressions for these time-domain kernels are extremely complicated and are not given here.

The multiwavelet-based kernel identification algorithm was used to identify first-, second-, and third-order kernels from simulated data from the nonlinear oscillator. The algorithm requires the user to select the polynomial order of the multiwavelets for the representation of the kernels. The current

implementation gives a choice between piecewise-linear, quadratic, or cubic functions. Both the linear and quadratic multiwavelets are composed of $r = 4$ scaling functions and associated wavelets, while the cubic multiwavelets are composed of 5. The additional scaling function associated with the cubic multiwavelets results in a significant increase in the total number of unknowns. Meanwhile, there is no increase in the number of unknowns in choosing the quadratic functions over the linear ones. In this case, one obtains better smoothness without adding more functions. Therefore, the piecewise-quadratic multiwavelets are used in this example and correspond to the best choice in general.

The input to the system was chosen to be a signal composed of the superposition of a number of linear chirp signals. The motivation for using such a "multichirp" input is that the higher-order frequency response functions encode information about the nonlinear response to combinations of different input frequencies. For example, consider an input consisting of the frequencies ω_1 and ω_2. The magnitude of the second-order frequency response function $H_2(j\omega_1, j\omega_2)$ gives the system gain at a frequency of $\omega_1 + \omega_2$ due to the nonlinear interaction of these two frequencies. Similarly, the third-order frequency response function predicts the system response to combinations of three frequencies. A very thorough discussion of the interpretation of higher-order frequency response functions is given in Ref. [28]. Here, it suffices to say that a simple linear chirp, which only excites one frequency at a time, is generally insufficient to completely identify higher-order kernels. Indeed, the response to a linear chirp will only give information about the higher-order kernels along diagonals in the frequency domain (i.e., points like $H_2(j\omega, j\omega)$). Therefore, in order to generate a suitable input that injects multiple frequencies into the system at once, several linear chirp signals were added together. Each of these signals corresponded to a chirp from 0 to 4 Hz translated to a different position on the time axis. The resulting multichirp input signal is shown in Figure 5.

The response of the system to the multichirp input was simulated by numerically integrating Equation (98) in MATLAB. Two data sets, corresponding to two different magnitudes of the input, were actually used to identify the kernels. The use of different magnitudes of the same input signal helps the algorithm discriminate between contributions from the first- and third-order kernels. The data were sampled at a rate of 128 Hz (or $j = 7$). The time duration of the simulation was 32 s for a total of $N = 4096$ points per data set.

The identified first-order kernel is shown in Figure 6. This kernel is approximated at a resolution level of $j_1 = 4$ and has a memory of $T_1 = 4$ s. Therefore, the kernel is represented in terms of $N_1 = 257$

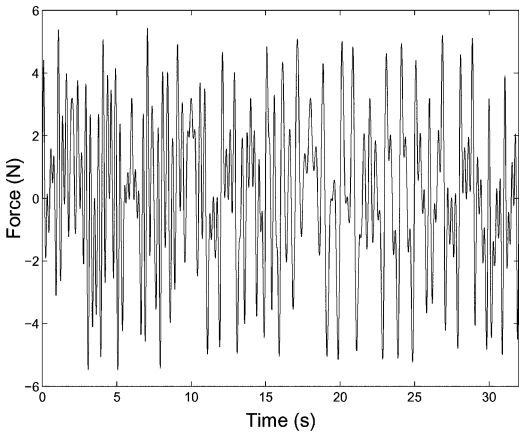

Figure 5. Multichirp input.

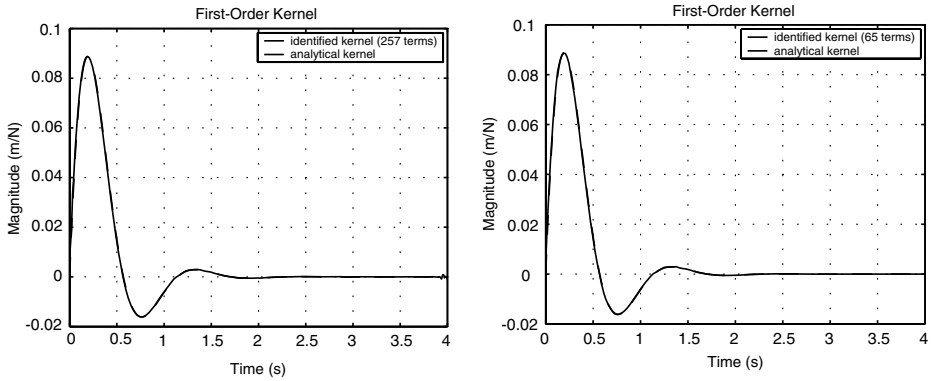

Figure 6. Identified and analytical first-order Volterra kernels (257- and 65-term approximations).

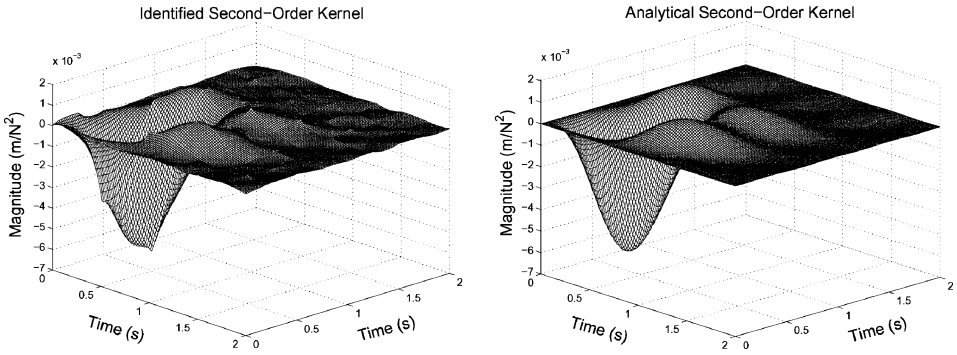

Figure 7. Identified and analytical second-order Volterra kernels.

wavelet functions. The analytical first-order kernel from Equation (104) is also shown in Figure 6. The identified kernel closely matches the analytical kernel. In this case, it is possible to reduce the number of wavelets needed to represent the kernel to $\hat{N}_1 = 65$. This reduced-order representation of the kernel, also shown in Figure 6, is clearly a very good approximation of the analytical kernel.

The identified second-order kernel is depicted in Figure 7. Also shown is the analytical kernel, computed in Mathematica. The identified kernel is approximated on a coarse resolution level of $j_2 = 1$ and has a memory of $T_2 = 2$ s. This representation corresponds to $\hat{N}_2 = 153$ unique wavelet coefficients. The identified kernel matches the analytical kernel very well despite the relatively coarse resolution of the model.

Finally, the identified third-order kernel is shown in Figure 8. Because the third-order kernel is supported over a three-dimensional domain, it is not possible to plot the entire kernel. Instead, two representative slices, corresponding to $\gamma = 0.5$ s and $\gamma = 1$ s, are plotted in the figure. The identified third-order kernel is approximated at a very coarse resolution level of $j_3 = 0$ and has a memory of $T_3 = 2$ s. Therefore, this representation is in terms of $\hat{N}_3 = 165$ unique wavelet coefficients. Comparing the slices of the identified kernel to the analytical kernel in Figure 8, it is clear that the approximation is very coarse and there is noticeable error. Nevertheless, the identified kernel still has the same basic form as the analytical kernel. In this case, increasing the resolution level does not improve the estimate of the third-order kernel. Augmenting the model to include the next-higher level of resolution increases the number of unknowns to 969. With so many unknowns to identify, the quality of the estimated

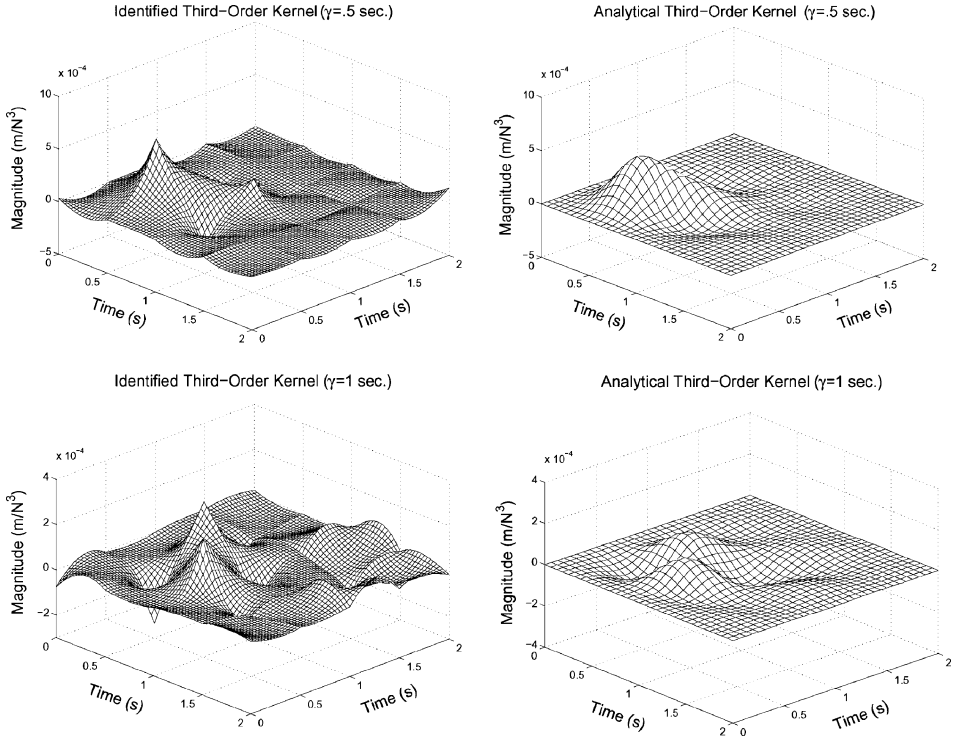

Figure 8. Identified and analytical third-order Volterra kernels (slices at $\gamma = .5$ s and $\gamma = 1$ s).

third-order kernel is degraded. This is a common characteristic of ill-posed problems, and the problem is exacerbated by the fact that the third-order kernel must be identified simultaneously with the first- and second-order kernels.

In general, systems with polynomial nonlinearities have infinite Volterra series representations. The nonlinear oscillator in Equation (98) is no exception. For many systems of interest, the first few Volterra kernels are sufficient to characterize the dynamics. As will be discussed later, this is usually dependent on the amplitude of the input. In this example, it has been assumed that the first three kernels can adequately characterize the system. To validate this assumption, the output predicted by the identified Volterra model is shown in Figure 9. The output corresponding to the larger input amplitude is shown in the figure. In this example, the linear response of the system can be obtained by simulating the response of Equation (103). This linear response is compared to the response predicted by the identified first-order kernel, which corresponds to the linear model. Since the linear response of the system is known, the nonlinear portion of the response can be obtained as the difference between the simulated responses of the nonlinear oscillator, Equation (98), and the linear oscillator, Equation (103). From this point onward, this signal will be termed the "nonlinear response." The total simulated output from the nonlinear oscillator is termed the "total response." With this terminology, then, the total response is simply the sum of the linear and nonlinear responses. The nonlinear response is compared to the response predicted by the identified second- and third-order kernels, which form the nonlinear model. Figure 9 shows that the identified Volterra model is capable of accurately reproducing the linear, nonlinear, and total responses.

It should be emphasized that the three Volterra kernels were identified simultaneously from the input and the total simulated response of the system. In practice, this may be the only information that

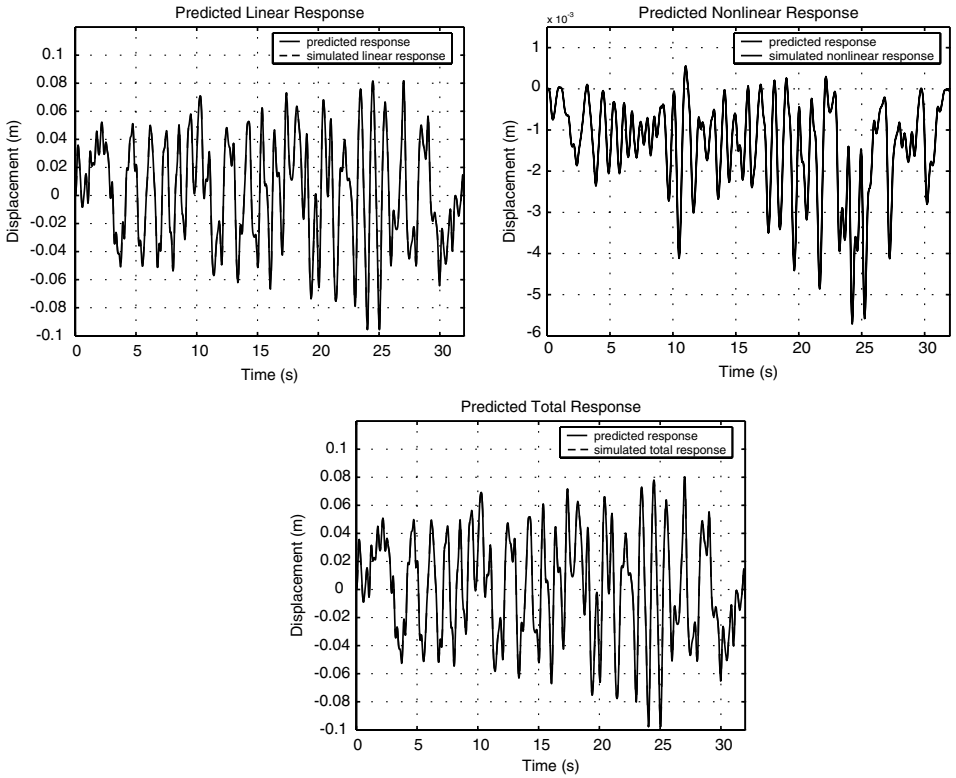

Figure 9. Predicted response for the multichirp input.

is available from a given system. Generally speaking, of course, the governing equation will not be known and we will not have an *a priori* decomposition of the total response into linear and nonlinear components. Therefore, it is important to note that the algorithm does not rely on such information.

Clearly, the identified kernels are able to accurately predict the responses to the multichirp inputs. The ability of the identified model to reproduce the training data is a necessary, but not sufficient, condition to validate the model. Indeed, the identified model is trained to reproduce this data so there is a danger of overfitting the model to the data. This is especially true as the number of unknown coefficients is increased. In this example, it is possible to verify that the model truly characterizes the system by comparing the identified kernels to the analytical kernels. Of course, analytical kernels are usually not available in practice, so the model must be validated by evaluating its ability to predict the responses to different input excitations.

Figure 10 shows the predicted response of the model to an input of the form:

$$u(t) = 4\sin(.5\pi t) + 2\sin(2\pi t) \tag{109}$$

This corresponds to the superposition of a 0.25 Hz sine wave with amplitude of 4 N and a 1 Hz sine wave with amplitude of 2 N. In order to evaluate the contribution of each kernel, the plots on the left show the responses predicted by the first-order kernel, the first- and second-order kernels, and the total model (all three kernels). The left-hand plots represent the information that would be available in practice for evaluating the model. For this particular system, additional insight can be gained from the

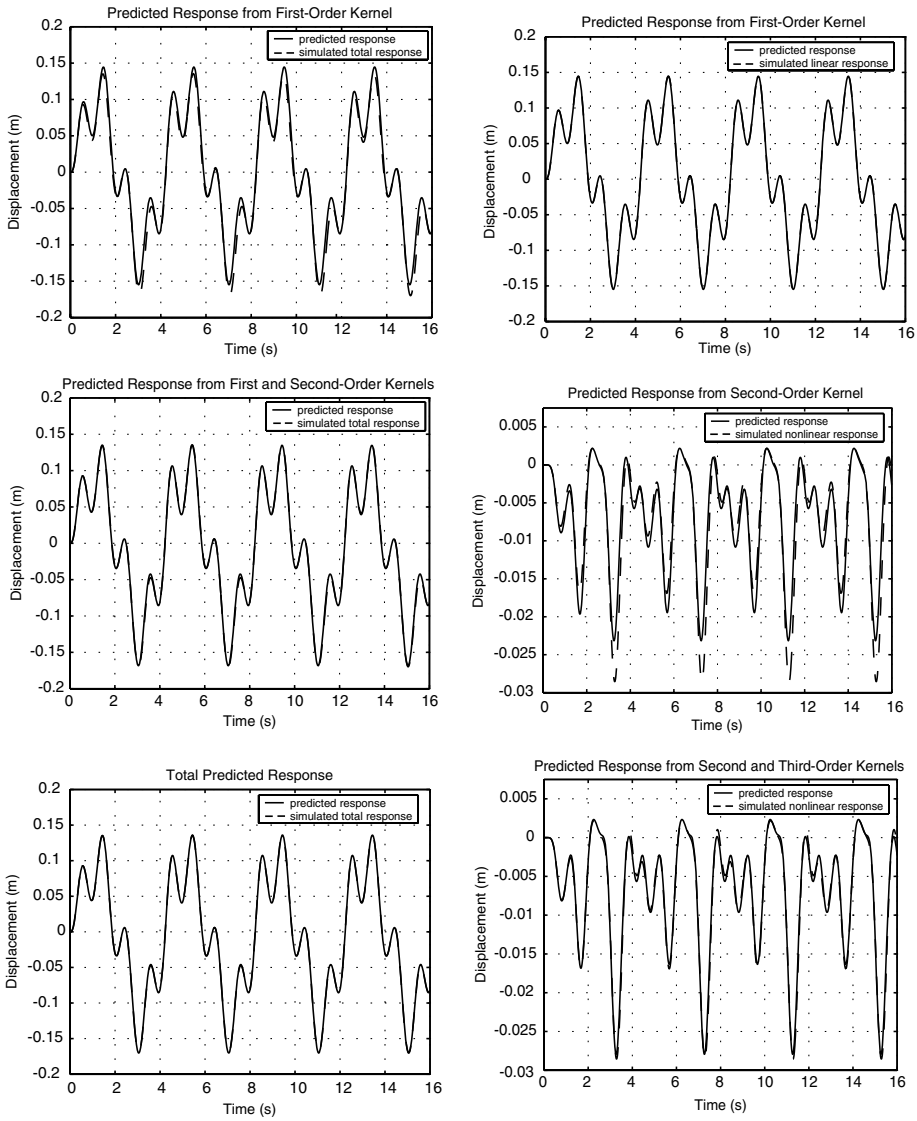

Figure 10. Predicted response for input $u(t) = 4\sin(.5\pi t) + 2\sin(2\pi t)$.

figures on the right side. These figures show the linear response predicted by the first-order kernel, the nonlinear response predicted by the second-order kernel alone, and the combined nonlinear response of the second- and third-order kernels. As shown in the figure, the first-order kernel provides an excellent approximation of the linear response. In fact, no error is discernible in the linear response plot. This is not surprising given how accurately the identified first-order kernel matches the analytical kernel. The nonlinear response is an order of magnitude smaller than the linear response, indicating a weak nonlinearity. Since the linear dynamics are dominant, the first-order kernel is able to capture most of the total response. The contribution of the second-order kernel eliminates most of the error from the linear approximation of the total response. The second-order kernel is capable of predicting most of the nonlinear response. The addition of the third-order contribution results in a very good approximation of the nonlinear response and removes any visible error in the prediction of the total response.

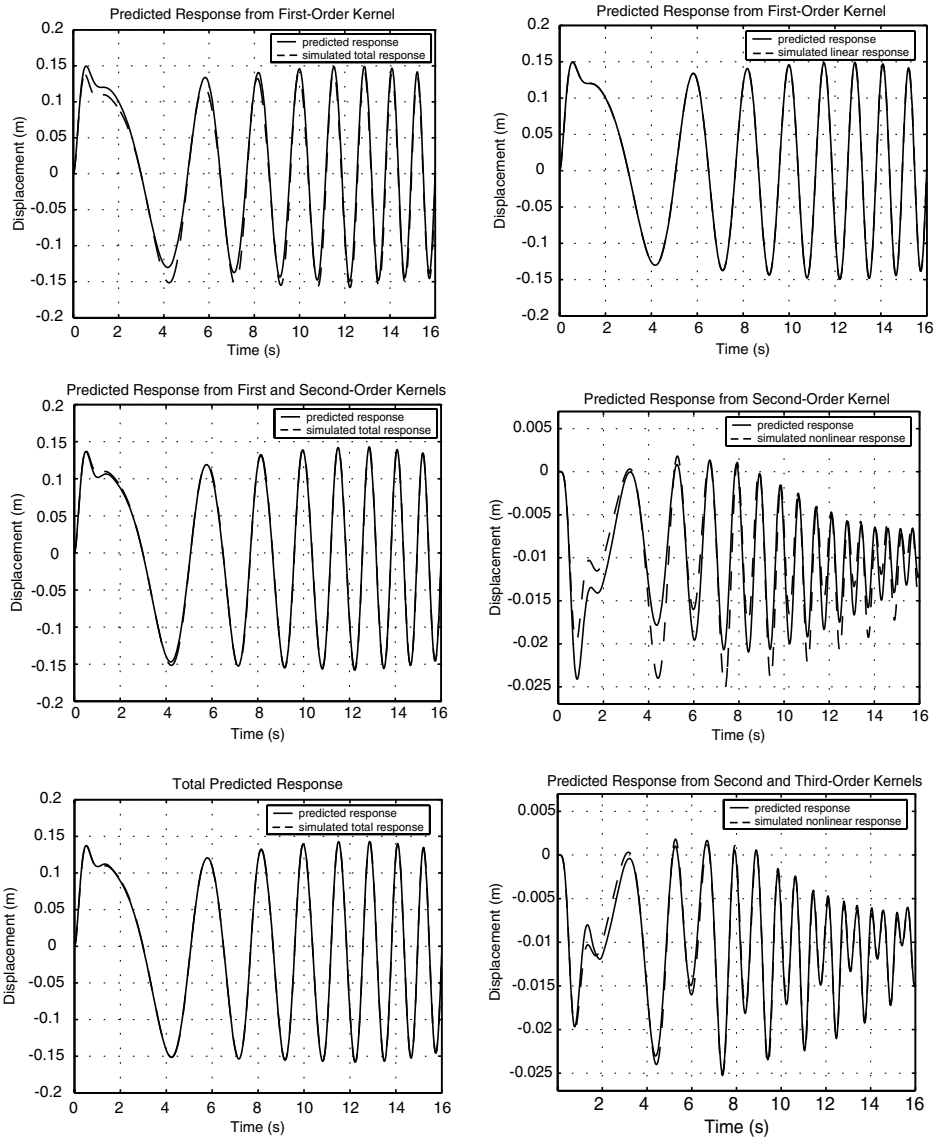

Figure 11. Predicted response for 0–1 Hz chirp input (amplitude of 5 N).

As another example, the response predicted by the identified model to a linear chirp input is given in Figure 11. The frequency range of the chirp was 0–1 Hz, the amplitude was 5 N, and the total simulation time was 16 s. The limited frequency range for this chirp was selected because it maximizes the excitation of the nonlinearity. The results are plotted in the same manner as in the previous example. These results show a similar trend as before. Once again, the first-order kernel accurately predicts the linear response and reproduces most of the total response. The second-order kernel captures much of the nonlinear response and removes most of the error in the prediction of the total response. Finally, the contribution of the third-order kernel significantly reduces the error in the predicted nonlinear response and provides a marginal improvement to the approximation of the total response.

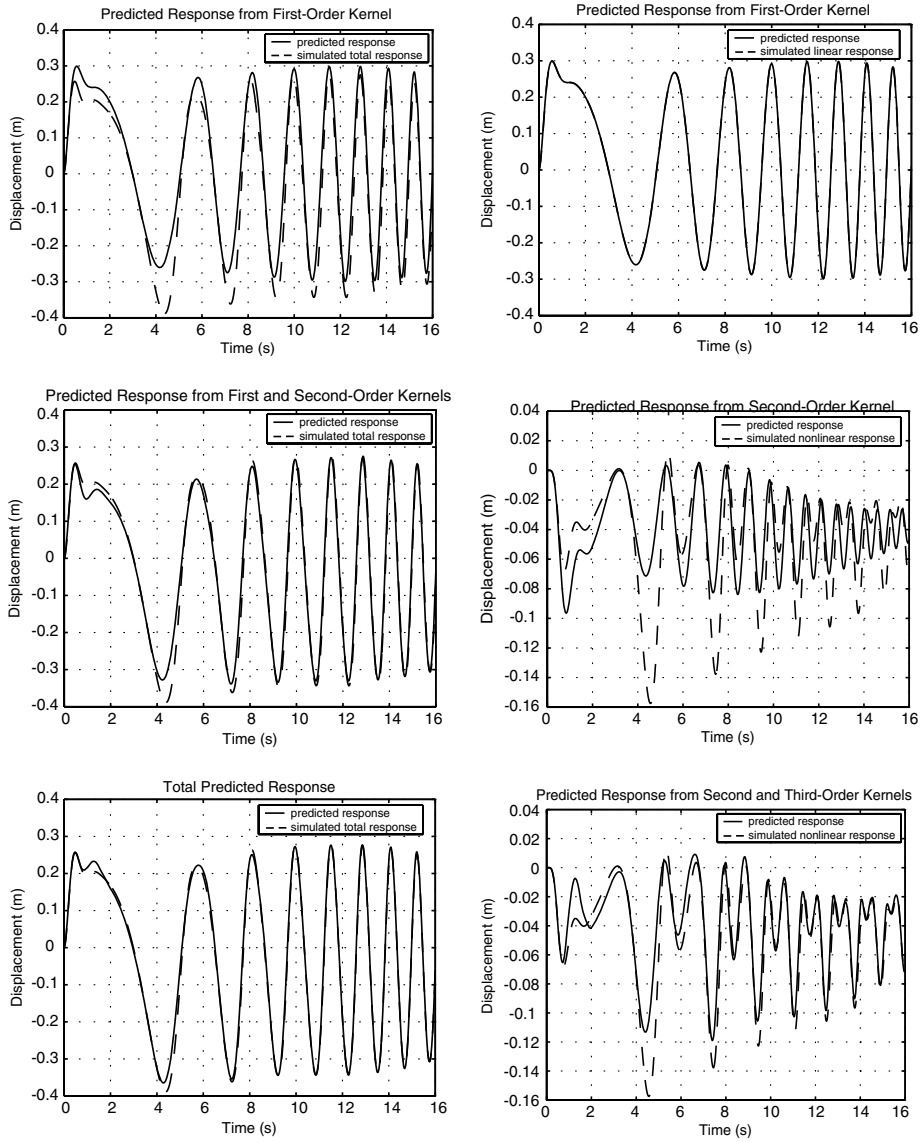

Figure 12. Predicted response for 0–1 Hz chirp input (amplitude of 10 N).

Errors in the predicted responses are due to a generally unknown combination of errors in the identified kernels and errors from unmodeled dynamics. The unmodeled dynamics are the contributions from the kernels of fourth order and higher and represent the truncation error of the three-term Volterra series. As a final example, consider the same chirp input from the last example with the amplitude doubled to 10 N. The predicted responses for this input are plotted in Figure 12. Several interesting observations can be made from these results. First, it is apparent that the relative contribution of the nonlinearity to the total response is much larger than in the previous case. In addition, the figure shows that the third-order kernel contributes a larger proportion of the nonlinear response. For systems with polynomial nonlinearities, raising the level of excitation generally increases the influence of the nonlinearity on the total response.

Furthermore, higher-order kernels have a greater contribution to the total response. This is evident from the expressions for the Volterra operators, which show that the contribution of the nth-order kernel is proportional to the nth power of the input. In this example, the second- and third-order kernels are no longer capable of completely characterizing the nonlinear response. As a result, there is noticeable error in the prediction of the total response. This error is due to the fact that kernels of fourth order and higher have an increased influence on the response. This truncation error can be expected to become more significant as the input amplitude is increased. In this example, however, if the input becomes too large, the quadratic spring stiffness will drive the system into a region of instability. This is because the quadratic term never changes sign. Therefore, it can be associated with a force that always acts in the same direction. Once the system becomes unstable, the Volterra series will no longer converge and cannot be used as a valid model regardless of how many terms are included in the series.

7. Conclusion

This paper has presented a kernel identification algorithm that employs multiwavelet representations of first-, second-, and third-order Volterra kernels. Piecewise-polynomial multiwavelets were constructed from classical finite elements using the technique of intertwining. This family of multiwavelets is particularly well-suited for the representation of Volterra kernels. The functions are orthonormal, compactly-supported, and possess good symmetry properties. It was demonstrated that the intertwining procedure generalizes to the construction of piecewise-polynomial multiwavelets of arbitrary approximation order. In addition, it was shown that these multiwavelets are easily adapted to the finite domains over which the kernels are supported. This collection of properties renders the multiwavelet-based algorithm superior to other wavelet-based approaches considered by the authors. As with the other approaches, this implementation reduces to a linear least-squares problem that must be solved for the kernel coefficients.

The multiwavelet-based algorithm was demonstrated on a prototypical nonlinear oscillator with a quadratic stiffness term. The identified first-, second-, and third-order kernels were compared to analytical kernels that were derived from the equation of motion using the harmonic balancing technique. The identified first- and second-order kernels closely resembled their analytical counterparts. The identified third-order kernel was a relatively coarse approximation, but did capture the basic features of the analytical kernel. The validity of the identified model was demonstrated by evaluating the ability of the kernels to predict the responses to novel input excitations. In all cases, the model was able to accurately predict the simulated response. It was noted, however, that as the input amplitude increases, unmodeled higher-order kernels have a greater influence on the response. Therefore, the validity of the identified Volterra model is generally dependent on the input excitation level.

The nonlinear oscillator example served to enumerate several other practical issues. One such issue is the need for appropriate input excitations for system identification. It was noted that linear chirp inputs, which sweep through one frequency at a time, are generally inadequate for identifying the higher-order, nonlinear kernels. When viewed in the frequency domain, these kernels model the nonlinear response due to the interaction of multiple frequencies. Therefore, a superposition of several translated chirp signals, or multichirp, was used as the input for the identification. A related point is that it was easier to identify the kernels using two data sets, corresponding to two different input amplitudes. This enabled the algorithm to better discern the contributions of the first- and third-order kernels. Another important issue is that, because the problem is ill-posed, adding coefficients (or levels of resolution) to the model does not necessarily improve the accuracy of the kernel estimates. Indeed, it can lead to overfitting the training data, underscoring the need to validate the kernels using novel data sets. Finally, the nonlinear

oscillator example lends some qualitative information about the computational efficiency of the kernel identification algorithm. The computational time required is dependent on numerous variables such as the sampling rate of the data, the length of the data sets, and the number of coefficients in the model (which is, in turn, dependent on the memory and resolution of the kernels). To give some idea of the cost, in the nonlinear oscillator example, the computation of the first-, second-, and third-order kernels required a total of 22 s using a Dell Precision 650 workstation with an Intel Xeon 3.2 GHz processor.

The kernel identification algorithm has several limitations in its present form. These are related to the fact that there is a limit to the total number of coefficients that can be included in the model. This is due, in part, to limitations encountered in the current MATLAB implementation of the algorithm. Also, as discussed before, obtaining accurate kernel estimates becomes more difficult as the number of unknowns is increased. This limitation implies that the algorithm may have difficulty modeling systems that possess high natural frequencies. Higher frequencies require more levels of resolution and, therefore, more wavelet coefficients. Similarly, the algorithm may not be able to model systems with large memory. Such systems are characterized by kernels that are nonzero over a longer time duration and, consequently, require more coefficients in the model. For a given system, these limitations are highly dependent on the sampling rate of the data and the total simulation time.

As a final note, the practical usefulness of the multiwavelet-based approach can best be assessed by evaluating its ability to extract kernels from experimental data. Such efforts are underway and some preliminary results have been obtained using flight data from the F/A-18 Active Aeroelastic Wing [29]. These results are promising and certainly merit further investigation.

Acknowledgements

The authors would like to thank Walter A. Silva of the NASA Langley Research Center for supporting much of this research. A portion of this research was conducted while the first author held a National Research Council postdoctoral fellowship at the NASA Dryden Flight Research Center. He would like to thank Martin J. Brenner of NASA Dryden for his support through this fellowship.

References

1. Chon, K., Chen, Y., Holstein-Rathlou, N.-H., and Marmarelis, V., 'Nonlinear system analysis of renal autoregulation in normotensive and hypertensive rats', *IEEE Transactions on Biomedical Engineering* **45**(3), 1998, 342–353.
2. French, A. S., Sekizawa, S., Hoger, U., and Torkkeli, P., 'Predicting the responses of mechanoreceptor neurons to physiological inputs by nonlinear system identification', *Annals of Biomedical Engineering* **19**, 2001, 187–194.
3. Marzocca, P., Librescu, L., and Silva, W. A., 'Aeroelastic response of nonlinear wing sections using a functional series technique', *AIAA Journal* **40**(5), 2002, 813–824.
4. Silva, W., 'Application of nonlinear systems theory to transonic unsteady aerodynamic responses,' *Journal of Aircraft* **30**(5), 1993, 660–668.
5. Silva, W., *Discrete-Time Linear and Nonlinear Aerodynamic Impulse Responses for Efficient CFD Analyses*, Ph.D. Dissertation, The College of William & Mary, Williamsburg, Virginia, 1997.
6. Cherry, J. A. and Snelgrove, W. M., 'On the characterization and reduction of distortion in bandpass filters', *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* **45**(5), 1998, 523–537.
7. Borys, A., *Nonlinear Aspects of Telecommunications: Discrete Volterra Series and Nonlinear Echo Cancellation*, CRC, New York, 2001.
8. Doyle, F. J., III, Pearson, R. K., and Ogunnaike, B. A., *Identification and Control Using Volterra Models*, Springer, London, 2002.
9. Lee, Y. W. and Schetzen, M., 'Measuring of the Wiener kernels of a nonlinear system by cross-correlation', *International Journal of Control* **2**(3), 1965, 237–254.

10. Boyd, S., Tang, Y. S., and Chua, L. O., 'Measuring Volterra kernels', *IEEE Transactions on Circuits and Systems* **30**(8), 1983, 571–577.
11. Wray, J. and Green, G., 'Calculation of the Volterra kernels of nonlinear dynamic systems using an artificial neural network', *Biological Cybernetics* **71**(3), 1994, 187–195.
12. Marmarelis, V. Z., 'Identification of nonlinear biological systems using Laguerre expansions of kernels', *Annals of Biomedical Engineering* **21**, 1993, 573–589.
13. Reisenthel, P., 'Prediction of unsteady aerodynamic forces via nonlinear kernel identification', in *International Forum on Aeroelasticity and Structural Dynamics*, Williamsburg, VA, 1999.
14. Beylkin, G., Coifman, R., and Rokhlin, V., 'Fast wavelet transforms and numerical algorithms', *Communications on Pure and Applied Mathematics* **44**, 1991, 141–183.
15. Nikolaou, M. and Mantha, D., 'Efficient nonlinear modelling using wavelets and related compression techniques', in *NSF Workshop on Nonlinear Model Predictive Control*, Ascona, Switzerland, 1998.
16. Chou, K. C. and Guthart, G. S., 'Representation of Green's function integral operators using wavelet transforms', *Journal of Vibration and Control* **6**, 2000, 19–48.
17. Kurdila, A. J., Prazenica, R. J., Rediniotis, O., and Strganac, T., 'Multiresolution methods for reduced-order models for dynamical systems', *Journal of Guidance, Control, and Dynamics* **24**(2), 2001, 193–200.
18. Cohen, A., Daubechies, I., and Feauveau, J. C., 'Biorthogonal bases of compactly supported wavelets', *Communications in Pure and Applied Mathematics* **45**, 1992, 485–559.
19. Prazenica, R. J. and Kurdila, A. J., 'Volterra kernel identification using triangular wavelets', *Journal of Vibration and Control* **10**(4), 2004, 597–622.
20. Donovan, G., Geronimo, J., and Hardin, D., 'Intertwining multiresolution analyses and the construction of piecewise-polynomial wavelets', *SIAM Journal of Mathematical Analysis* **27**(6), 1996, 1791–1815.
21. Donovan, G., Geronimo, J., and Hardin, D., 'Orthogonal polynomials and the construction of piecewise-polynomial smooth wavelets', *SIAM Journal of Mathematical Analysis* **30**(5), 1999, 1029–1056.
22. Rugh, W. J., *Nonlinear System Theory: The Volterra–Wiener Approach*, Johns Hopkins University Press, Baltimore, MD, 1980.
23. Schetzen, M., *The Volterra and Wiener Theories of Nonlinear Systems*, Wiley, New York, 1980.
24. Boyd, S. and Chua, L. O., 'Fading memory and the problem of approximating nonlinear operators with Volterra series', *IEEE Transactions on Circuits and Systems* **CAS-32**(11), 1985, 1150–1161.
25. Mallat, S., *A Wavelet Tour of Signal Processing*, Academic, New York, 1999.
26. Groetsch, C. W., *Inverse Problems in the Mathematical Sciences*, Vieweg, Wiesbaden, Germany, 1993.
27. Hansen, P. C., *Rank Deficient and Discrete Ill-Posed Problems*, SIAM, Philadelphia, PA, 1998.
28. Worden, K. and Tomlinson, G. R., *Nonlinearity in Structural Dynamics: Detection, Identification, and Modelling*, Institute of Physics, Philadelphia, PA, 2001.
29. Prazenica, R. J., Brenner, M. J., and Lind, R., 'Nonlinear Volterra kernel identification for the F/A-18 active aeroelastic wing', in *International Forum on Aeroelasticity and Structural Dynamics*, Amsterdam, The Netherlands, 2003.