A Unifying View of Wiener and Volterra Theory and Polynomial Kernel Regression

Matthias O. Franz
mof@tuebingen.mpg.de
Bernhard Schölkopf
bs@tuebingen.mpg.de
Max-Planck-Institut für biologische Kybernetik, D-72076 Tübingen, Germany

Volterra and Wiener series are perhaps the best-understood nonlinear system representations in signal processing. Although both approaches have enjoyed a certain popularity in the past, their application has been limited to rather low-dimensional and weakly nonlinear systems due to the exponential growth of the number of terms that have to be estimated. We show that Volterra and Wiener series can be represented implicitly as elements of a reproducing kernel Hilbert space by using polynomial kernels. The estimation complexity of the implicit representation is linear in the input dimensionality and independent of the degree of nonlinearity. Experiments show performance advantages in terms of convergence, interpretability, and system sizes that can be handled.

1 Introduction

In system identification, one tries to infer the functional relationship between system input and output from observations of the ingoing and outgoing signals. If the system is linear, it can be always characterized uniquely by its impulse response. For nonlinear systems, however, there exists no canonical representation that encompasses all conceivable systems. The earliest approach to a systematic, that is, a nonparametric, characterization of nonlinear systems dates back to V. Volterra, who extended the standard convolution description of linear systems by a series of polynomial integral operators with increasing degree of nonlinearity, very similar in spirit to the Taylor series for analytical functions (Volterra, 1887). The last 120 years have seen the accumulation of huge amount of research done on both the class of systems that can be represented by Volterra operators and their application in such diverse fields as nonlinear differential equations, neuroscience, fluid dynamics and electrical engineering (overviews and bibliography in Schetzen, 1980; Rugh, 1981; Mathews & Sicuranza, 2000; Giannakis & Serpedin, 2001).

A principal problem of the Volterra approach is the exponential growth of the number of terms in the operators, with both degree of nonlinearity and input dimensionality. This has limited its application to rather low-dimensional systems with mild nonlinearities. Here, we show that this problem can be largely alleviated by reformulating the Volterra and Wiener series as operators in a reproducing kernel Hilbert space (RKHS). In this way, the whole Volterra and Wiener approach can be incorporated into the rapidly growing field of kernel methods. In particular, the estimation of Volterra or Wiener expansions can be done by polynomial kernel regression that scales only linearly with input dimensionality, independent of the degree of nonlinearity. Moreover, RKHS theory allows us to estimate even infinite Volterra series, which was not possible before. Our experiments indicate that the RKHS formulation also leads to practical improvements in terms of prediction accuracy and interpretability of the results.

In the next section, we review the essential results of the classical Volterra and Wiener theories of nonlinear systems. (This section is mainly a review for readers who are not familiar with Wiener and Volterra theory.) In section 3, we discuss newer developments since the mid-1980s that lead to our new formulation, which is presented in section 4. A preliminary account of this work has appeared in Franz and Schölkopf (2004).

2 Volterra and Wiener Theory of Nonlinear Systems

2.1 The Volterra Class. A system can be defined as a map that assigns an output signal $y(t)$ to an input signal $x(t)$ (we assume for the moment that $x(t)$ and $y(t)$ are functions of time t). Mathematically, this rule can be expressed in the form

$$y(t) = Tx(t) \tag{2.1}$$

using a system operator T that maps from the input to the output function space. The system is typically assumed to be time invariant and continuous; the system response should remain unchanged for repeated presentation of the same input, and small changes in the input functions $x(t)$ should lead to small changes in the corresponding system output functions $y(t)$. In traditional systems theory, we further restrict T to be a sufficiently well-behaved compact linear operator H_1 such that the system response can be described by a convolution,

$$y(t) = H_1 x(t) = \int h^{(1)}(\tau) x(t-\tau)\, d\tau, \tag{2.2}$$

of $x(t)$ with a linear kernel (or impulse response) $h^{(1)}(\tau)$. A natural extension of this convolution description to nonlinear systems is the Volterra series operator,

$$y(t) = Vx(t) = H_0 x(t) + H_1 x(t) + H_2 x(t) + \cdots + H_n x(t) + \cdots, \qquad (2.3)$$

in which $H_0 x(t) = h_0 = \text{const.}$ and

$$H_n x(t) = \int h^{(n)}(\tau_1, \ldots, \tau_n) x(t - \tau_1) \ldots x(t - \tau_n)\, d\tau_1, \ldots, d\tau_n \qquad (2.4)$$

is the nth-order Volterra operator (Volterra, 1887, 1959). The integral kernels $h^{(n)}(\tau_1, \ldots, \tau_n)$ are the Volterra kernels. Depending on the system to be represented, the integrals can be computed over finite or infinite time intervals. The support of the Volterra kernel defines the memory of the system; that is, it delimits the time interval in which past inputs can influence the current system output. The Volterra series can be regarded accordingly as a Taylor series with memory: whereas the usual Taylor series represents only systems that instantaneously map the input to the output, the Volterra series characterizes systems in which the output also depends on past inputs.

The input functions typically come from some real, separable Hilbert space such as $L^2[a, b]$, the output functions from the space $C[a, b]$ of bounded continuous functions. Similar to the Taylor series, the convergence of a Volterra series can be guaranteed for only a limited range of the system input amplitude. As a consequence, the input functions must be restricted to some suitable subset of the input space. For instance, if the input signals form a compact subset of the input function space, one can apply the Stone-Weierstraß theorem (a generalization of the Weierstraß theorem to nonlinear operators; see, e.g., Hille & Phillips, 1957) to show that any continuous, nonlinear system can be uniformly approximated (i.e., in the L^∞-norm) to arbitrary accuracy by a Volterra series operator of sufficient but finite order (Fréchet, 1910; Brilliant, 1958; Prenter, 1970).[1]

Although this approximation result appears to be rather general on first sight, the restriction to compact input sets is quite severe. An example of a compact subset is the set of functions from $L^2[a, b]$ defined over a closed time interval with a common upper bound (proof in Liusternik & Sobolev, 1961). In practice, this means that the input signals have to be nonzero only on a finite time interval and that the approximation holds only there. Many natural choices of input signals are precluded by this requirement, such as the unit ball in $L^2[a, b]$ or infinite periodic forcing signals.

[1] If one further restricts the system to have fading memory (i.e., the influence of past inputs decays exponentially), the uniform approximation by finite Volterra series can be extended to bounded and slew-limited input signals on infinite time intervals (Boyd & Chua, 1985).

2.2 The Wiener Class. So far, we have discussed only the representation of a general nonlinear system. Now we come to the problem of obtaining such a representation from data. For a linear system, this is a straightforward procedure since it suffices to test the system on a set of basis functions from the input space (e.g., delta functions or sinusoids). In a nonlinear system, however, we ideally have to measure the system response for all possible input functions. One way to achieve this is by testing the system on realizations of a suitable random process.

The stochastic input in Wiener theory is the limiting form of the random walk process as the number of steps goes to infinity (or, equivalently, as the step size goes to zero), which is now known as the Wiener process (Papoulis, 1991). One can show that the Wiener process assigns a nonzero probability to the neighbourhood of every continuous input function (Palm & Poggio, 1977). Thus, the realizations of the Wiener process play a similar role in Wiener theory as the sinusoidal test inputs in linear system theory since they are capable of completely characterizing the system.

In system identification, we are given only pairs of input and output functions, whereas the system itself is treated as a black box. The appropriate Volterra representation has to be found by minimizing some error measure between the true output and the model output, such as the integral over the squared error. Thus, the approximation has to be only in the L^2-norm, not in the L^∞-norm as in Volterra theory. A weaker approximation criterion typically relaxes the restrictions imposed on the input and output set and on the type of systems that can be represented by a Volterra series (Palm, 1978). Wiener theory relaxes the approximation criterion even further: assuming that the input is generated by the Wiener process, it requires only an approximation in the mean squared error sense over the whole process, not for any single realization of it.

The minimization of the mean squared error for the estimation of the Volterra kernels requires the solution of a simultaneous set of integral equations. This can be avoided by using an orthogonal least-squares framework as proposed by Wiener (1958) and Barrett (1963). Since the distribution of the input is known for the Wiener process, we can choose an input-specific decomposition of the system operator T,

$$y(t) = G_0 x(t) + G_1 x(t) + G_2 x(t) + \cdots + G_n x(t) + \cdots, \tag{2.5}$$

into a Wiener series of operators G_n that are mutually uncorrelated, that is, orthogonal with respect to the Wiener process. The Wiener operators G_n are linear combinations of Volterra operators up to order n. They can be obtained from the original Volterra series by a procedure very similar to

Gram-Schmidt orthogonalization. For instance, the second-degree Wiener operator[2],

$$G_2x(t) = \int h_2(\tau_1, \tau_2)x(t-\tau_1)x(t-\tau_2)\,d\tau_1 d\tau_2 - \int h_2(\tau_1, \tau_1)\,d\tau_1, \qquad (2.6)$$

consists of a zero-order and a second-order Volterra operator. The integral kernel of the highest-order (i.e., nth order) Volterra operator of G_n is called the leading Volterra kernel of G_n. As a result of the orthogonalization, the G_n can be estimated independent of each other. Moreover, any truncation of this orthogonalized series minimizes the mean squared error among all truncated Volterra expansions of the same order.

All systems that produce square integrable output for the Wiener input process can be approximated in the mean square sense by finite-order Wiener series operators (Ahmed, 1970). In practice, this means that the systems must be nondivergent and cannot have infinite memory. Due to the different types of inputs and convergence, the classes of systems that can be approximated by infinite Volterra or Wiener series operators are not identical. Some systems of the Wiener class cannot be represented as a Volterra series operator and vice versa (Palm & Poggio, 1977; Korenberg & Hunter, 1990). However, a truncated Wiener or Volterra series can always be transformed into its truncated counterpart.

One of the reasons for the popularity of the Wiener series is that the leading Volterra kernels can be directly measured via the cross-correlation method of Lee and Schetzen (1965). If one uses gaussian white noise with standard deviation A instead of the Wiener process as input, the leading Volterra kernel of G_n can be estimated as

$$h^{(n)}(\tau_1, \ldots, \tau_n) = \frac{1}{n!A^n}\overline{\left(y(t) - \sum_{l=0}^{n-1} G_l x(t)\right) x(t-\tau_1)\ldots x(t-\tau_n)}, \qquad (2.7)$$

where the bar indicates the average over time. The zero-order kernel is simply the time average $h^{(0)} = \overline{y(t)}$ of the output function. The other lower-order Volterra kernels of G_n can be derived from the leading kernel by again applying a Gram-Schmid-type orthogonalization procedure.

2.3 Discrete Volterra and Wiener Systems. In practical signal processing, one uses a discretized form for a finite sample of data. Here, we assume

[2] Strictly speaking, the integrals in the Wiener operators have to be interpreted as stochastic integrals (e.g., Papoulis, 1991) with respect to the Wiener process; that is, the equality holds only in the mean squared sense. For conditions under which the equality also holds for specific inputs, see Palm & Poggio (1977).

that the input data are given as a vector $\mathbf{x} = (x_1, \ldots, x_m)^\top \in \mathbb{R}^m$ of finite dimension. The vectorial data can be generated from any multidimensional input or, for instance, by a sliding window over a discretized image or time series. A discrete system is simply described by a function $T : \mathbb{R}^m \to \mathbb{R}$, not by an operator as before. The discretized Volterra operator is defined as the function

$$H_n(\mathbf{x}) = \sum\nolimits_{i_1=1}^{m} \cdots \sum\nolimits_{i_n=1}^{m} h^{(n)}_{i_1 \ldots i_n} x_{i_1}, \ldots, x_{i_n}, \tag{2.8}$$

where the Volterra kernel is given as a finite number of m^n coefficients $h^{(n)}_{i_1,\ldots,i_n}$ (Alper, 1965). It is, accordingly, a linear combination of all ordered nth-order monomials of the elements of $\mathbf{x}$.[3] Analogous to the continuous-time Volterra series, it can be shown by applying the Stone-Weierstraß theorem that all continuous systems with compact input domain can be uniformly approximated by a finite, discrete Volterra series. For systems with exponentially fading memory, the uniform approximation can be extended to all input vectors with a common upper bound (Boyd & Chua, 1985).

The discrete analog to the Wiener series is typically orthogonalized with respect to gaussian input $\mathbf{x} \sim \mathcal{N}(0, A)$ since this is the only practical setting where the popular cross-correlation method can be applied. All properties of continuous Wiener series operators described above carry over to the discrete case. In particular, any square-summable functional with gaussian input can be approximated in the mean square sense by a finite, discrete Wiener series (Palm & Poggio, 1978).

2.4 Problems of the Cross-Correlation Method. The estimation of the Wiener expansion via cross-correlation poses some serious problems:

1. The estimation of cross-correlations requires large sample sizes. Typically, one needs several tens of thousands of input-output pairs before a sufficient convergence is reached. Moreover, the variance of the cross-correlation estimator in equation 2.7 increases with increasing values of the time delay τ_i (Papoulis, 1991) such that only operators with relatively small memory can be reliably estimated.

2. The estimation via cross-correlation works only if the input is gaussian noise with zero mean, not for general types of input. In physical experiments, however, deviations from ideal white noise and the resulting estimation errors cannot be avoided. Specific inputs, on the other hand, may have a very low probability of being generated by

[3] Throughout this letter, we assume that the Volterra kernels are symmetrical with respect to permutations of the indices i_j. A nonsymmetrical kernel can be converted into a symmetrical kernel without changing the system output (Volterra, 1959; Mathews & Sicuranza, 2000).

white noise. Since the approximation is computed only in the mean square sense, the system response to these inputs may be drastically different from the model predictions.[4]

3. In practice, the cross-correlations have to be estimated at a finite resolution (cf. the discretized version of the Volterra operator in equation 2.8). The number of expansion coefficients in equation 2.8 increases with m^n for an m-dimensional input signal and an nth-order Wiener kernel. However, the number of coefficients that actually have to be estimated by cross-correlation is smaller. Since the products in equation 2.8 remain the same when two different indices are permuted, the associated coefficients are equal in symmetrical Volterra operators. As a consequence, the required number of measurements is $(n+m-1)!/(n!(m-1)!)$ (Mathews & Sicuranza, 2000). Nonetheless, the resulting numbers are huge for higher-order Wiener kernels. For instance, a fifth-order Wiener kernel operating on 256-dimensional input contains roughly 10^{12} coefficients, 10^{10} of which would have to be measured individually by cross-correlation. As a consequence, this procedure is not feasible for higher-dimensional input signals.

4. The cross-correlation method assumes noise-free signals. For real noise-contaminated data, the estimated Wiener series models both signal and noise of the training data, which typically results in reduced prediction performance on independent test sets.

3 Estimating Wiener Series by Linear Regression in RKHS

3.1 Linear Regression. The first two problems can be overcome by adopting the framework of linear regression: given observations $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$, linear regression tries to estimate y as a function of $\mathbf{x}$ by

$$y = f(\mathbf{x}) = \sum\nolimits_{j=1}^{M} \gamma_j \varphi_j(\mathbf{x}), \tag{3.1}$$

using $\gamma_j \in \mathbb{R}$ and a dictionary of M functions $\varphi_j : \mathbb{R}^m \to \mathbb{R}$ where M is allowed to be infinite. In the case of pth-order Volterra or Wiener series, this dictionary consists of all monomials of $\mathbf{x}$ up to order p (see equation 2.8).

[4] A number of studies develop an orthogonal framework with respect to other input classes (Schetzen, 1965; Ogura, 1972; Segall & Kailath, 1976). None of these, however, can be applied to input classes different from the one they were developed for.

Instead of assuming an infinite amount of data, the γ_j are found by minimizing the mean squared error over the dataset

$$c((\mathbf{x}_1, y_1, f(\mathbf{x}_1)), \ldots, (\mathbf{x}_N, y_N, f(\mathbf{x}_N))) = \frac{1}{N}\sum\nolimits_{j=1}^{N}(f(\mathbf{x}_j) - y_j)^2, \tag{3.2}$$

which disposes of the cumbersome cross-correlation estimator (Korenberg, Bruder, & McIlroy, 1988; Mathews & Sicuranza, 2000). Moreover, the input signal class is no more restricted to gaussian noise, but can be chosen freely, for example, from the "natural" input ensemble of the system. As long as the input is known to the experimenter, there is no need for controlling the input as in the classical system identification setting. Note, however, that the obtained Volterra models will approximate the Wiener series only for sufficiently large data sets of gaussian white noise. Korenberg et al. (1988) have shown that the linear regression framework leads to Wiener models that are orders of magnitude more accurate than those obtained from the cross-correlation method. Unfortunately, the solution of this regression problem requires the inversion of an $M \times M$ matrix (Mathews & Sicuranza, 2000). This is again prohibitive for high-dimensional data and higher orders of nonlinearity since M scales like m^n.

3.2 Regression in RKHS. If we stack the basis functions $\varphi_j(\mathbf{x})$ into a common vector $\Phi(\mathbf{x}) = (\varphi_1(\mathbf{x}), \varphi_2(\mathbf{x}), \ldots)$, we can interpret $\Phi(\mathbf{x})$ as a nonlinear mapping from $\mathbb{R}^m$ into another, possibly infinite-dimensional space $\mathcal{H}$. For certain dictionaries, $\mathcal{H}$ constitutes a dot product space where the dot product can be expressed in terms of a positive definite[5] kernel function[6] $k(\mathbf{x}, \mathbf{x}') = \Phi(\mathbf{x})^\top \Phi(\mathbf{x}')$, which can be evaluated without computing Φ, that is, the possibly infinite-dimensional dot product can be replaced by a simple function evaluation (see, e.g., Schölkopf & Smola, 2002).

An important property of such a space is that it can be identified with a suitable closure of the space of functions,

$$f(\mathbf{x}) = \sum\nolimits_{j=1}^{K} \alpha_j k(\mathbf{x}, \mathbf{z}_j), \tag{3.3}$$

with an arbitrary set of points $\mathbf{z}_1, \ldots, \mathbf{z}_K$ from $\mathbb{R}^m$; in this case, any expansion of type 3.1 can be expressed in the form 3.3 (Schölkopf & Smola, 2002). This space has the structure of an RKHS, which allows the application of the so-called representer theorem. It states the following: suppose c is an arbitrary

[5] That is, the Gram matrix $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ is positive definite for all choices of the $\mathbf{x}_1, \ldots, \mathbf{x}_N$ from the input domain.

[6] Note that with a slight abuse of notation, we will nevertheless use the transpose to denote the dot product in that space.

cost function, Ω is a nondecreasing function on $\mathbb{R}_+$, and $\|\cdot\|_{\mathbb{F}}$ is the norm of the RKHS. If we minimize an objective function

$$c((\mathbf{x}_1, y_1, f(\mathbf{x}_1)), \ldots, (\mathbf{x}_K, y_K, f(\mathbf{x}_K))) + \Omega(\|f\|_{\mathbb{F}}), \tag{3.4}$$

over all α_j and $\mathbf{z}_j$ in equation 3.3, then an optimal solution[7] can be expressed as

$$f(\mathbf{x}) = \sum\nolimits_{j=1}^{N} \alpha_j k(\mathbf{x}, \mathbf{x}_j), \quad \alpha_j \in \mathbb{R}. \tag{3.5}$$

In other words, although we did consider functions that were expansions in terms of arbitrary points $\mathbf{z}_j$ (see equation 3.3), it turns out that we can always express the solution in terms of the training points $\mathbf{x}_j$ only. Hence, the optimization problem over an arbitrarily large number of M weights γ_j is transformed into one over N weights α_j, where N is the number of training points.

In our case, the cost function is given by equation 3.2, and the regularizer Ω is zero. The optimal weight set $\alpha = (\alpha_1, \ldots, \alpha_N)$ is readily computed by setting the derivative of equation 3.2 with respect to the weights α_j equal to zero; it takes the form $\alpha = K^{-1}\mathbf{y}$ where $\mathbf{y} = (y_1, \ldots, y_N)^\top$; hence,[8]

$$y = f(\mathbf{x}) = \alpha^\top \mathbf{k}(\mathbf{x}) = \mathbf{y}^\top K^{-1}\mathbf{k}(\mathbf{x}), \tag{3.6}$$

where $\mathbf{k}(\mathbf{x}) = (k(\mathbf{x}, \mathbf{x}_1), k(\mathbf{x}, \mathbf{x}_2), \ldots, k(\mathbf{x}, \mathbf{x}_N))^\top \in \mathbb{R}^N$. As a result, we have to invert an $N \times N$ matrix instead of an $M \times M$ matrix in linear regression. For high-dimensional data, we typically have $M = m^n \gg N$. In this case, a time complexity[9] of $O(mN^2 + N^3)$ and memory complexity of $O(N^2)$ compares favorably to the exponential complexity of the original linear regression problem, which is $O(m^{3n})$ and $O(m^{2n})$, respectively.

3.3 Volterra Series as Linear Operator in RKHS. In order to apply the RKHS framework to the problem of estimating the Volterra and Wiener expansion of a system, we have to find a suitable kernel. Our starting point is the discretized version of the Volterra operators from equation 2.8. The nth-order Volterra operator is a weighted sum of all nth-order monomials of the input vector $\mathbf{x}$. For $n = 0, 1, 2, \ldots$ we define the map Φ_n as

$$\Phi_0(\mathbf{x}) = 1 \quad \text{and} \quad \Phi_n(\mathbf{x}) = (x_1^n, x_1^{n-1}x_2, \ldots, x_1 x_2^{n-1}, x_2^n, \ldots, x_m^n) \tag{3.7}$$

[7] For conditions on the uniqueness of the solution, see Schölkopf and Smola (2002).

[8] If K is not invertible, K^{-1} denotes the pseudo-inverse of K.

[9] The evaluation of the kernel function typically has complexity $O(m)$, which holds true in the polynomial kernels described below.

such that Φ_n maps the input $\mathbf{x} \in \mathbb{R}^m$ into a vector $\Phi_n(\mathbf{x}) \in \mathcal{H}_n = \mathbb{R}^{m^n}$ containing all m^n ordered monomials of degree n evaluated at $\mathbf{x}$. Using Φ_n, we can write the nth-order Volterra operator in equation 2.8 as a scalar product in $\mathcal{H}_n$,

$$H_n(\mathbf{x}) = \eta_n^\top \Phi_n(\mathbf{x}), \tag{3.8}$$

with the coefficients stacked into the vector $\eta_n = (h^{(n)}_{1,1,\ldots,1}, h^{(n)}_{1,2,\ldots,1}, h^{(n)}_{1,3,\ldots,1}, \ldots)^\top \in \mathcal{H}_n$. Fortunately, the monomials constitute an RKHS. It can be easily shown (e.g., Schölkopf & Smola, 2002) that

$$\Phi_n(\mathbf{x}_1)^\top \Phi_n(\mathbf{x}_2) = (\mathbf{x}_1^\top \mathbf{x}_2)^n =: k_n(\mathbf{x}_1, \mathbf{x}_2). \tag{3.9}$$

This equivalence was used as early as 1975 in an iterative estimation scheme for Volterra models, long before the RKHS framework became commonplace (Poggio, 1975).

The estimation problem can be solved directly if one applies the same idea to the entire pth-order Volterra series. By stacking the maps Φ_n with positive weights a_n into a single map $\Phi^{(p)}(\mathbf{x}) = (a_0\Phi_0(\mathbf{x}), a_1\Phi_1(\mathbf{x}), \ldots, a_p\Phi_p(\mathbf{x}))^\top$, one obtains a mapping from $\mathbb{R}^m$ into $\mathcal{H}^{(p)} = \mathbb{R} \times \mathbb{R}^m \times \mathbb{R}^{m^2} \times \ldots \times \mathbb{R}^{m^p} = \mathbb{R}^M$ with dimensionality $M = \frac{1-m^{p+1}}{1-m}$. The entire pth-order Volterra series can be written as a scalar product in $\mathcal{H}^{(p)}$,

$$\sum\nolimits_{n=0}^{p} H_n(\mathbf{x}) = \left(\eta^{(p)}\right)^\top \Phi^{(p)}(\mathbf{x}), \tag{3.10}$$

with $\eta^{(p)} \in \mathcal{H}^{(p)}$. Since $\mathcal{H}^{(p)}$ is generated as a Cartesian product of the single spaces $\mathcal{H}_n$, the associated scalar product is simply the weighted sum of the scalar products in $\mathcal{H}_n$:

$$\Phi^{(p)}(\mathbf{x}_1)^\top \Phi^{(p)}(\mathbf{x}_2) = \sum\nolimits_{n=0}^{p} a_n^2 \left(\mathbf{x}_1^\top \mathbf{x}_2\right)^n =: k^{(p)}(\mathbf{x}_1, \mathbf{x}_2). \tag{3.11}$$

A special case of this kernel is the inhomogeneous polynomial kernel used in the Volterra estimation approach of Dodd and Harrison (2002),

$$k^{(p)}_{inh}(\mathbf{x}_1, \mathbf{x}_2) = \left(1 + \mathbf{x}_1^\top \mathbf{x}_2\right)^p, \tag{3.12}$$

which corresponds to

$$\left(1 + \mathbf{x}_1^\top \mathbf{x}_2\right)^p = \sum_{n=0}^{p} \binom{p}{n} \left(\mathbf{x}_1^\top \mathbf{x}_2\right)^n \tag{3.13}$$

via the binomial theorem. If a suitably decaying weight set a_n is chosen, the approach can be extended even to infinite Volterra series. For instance, for $a_n = \sqrt{1/n!}$ we obtain the translation-variant gaussian kernel

$$k^{(\infty)}(\mathbf{x}_1, \mathbf{x}_2) = e^{\mathbf{x}_1^\top \mathbf{x}_2} = \sum\nolimits_{n=0}^{\infty} \frac{1}{n!}\left(\mathbf{x}_1^\top \mathbf{x}_2\right)^n, \tag{3.14}$$

or for $\|\mathbf{x}\| < 1$, $\alpha > 0$, Vovk's infinite polynomial kernel (Saunders et al., Smola, 1998),

$$k_{\mathrm{Vovk}}(\mathbf{x}_1, \mathbf{x}_2) = \left(1 - \mathbf{x}_1^\top \mathbf{x}_2\right)^{-\alpha} = \sum\nolimits_{n=0}^{\infty} \binom{-\alpha}{n} (-1)^n \left(\mathbf{x}_1^\top \mathbf{x}_2\right)^n. \tag{3.15}$$

The latter two kernels have been shown to be universal; the functions of their associated RKHS are capable of uniformly approximating all continuous functions on compact input sets in $\mathbb{R}^m$ (Steinwart, 2001). As we have seen in the discussion of the approximation capabilities of discrete Volterra series, the family of finite polynomial kernels in its entirety is also universal since the union of their RKHSs comprises all discrete Volterra series. Isolated finite polynomial kernels, however, do not share this property.

3.4 Implicit Wiener Series Estimation. We know now that both finite and infinite discretized Volterra series can be expressed as linear operators in an RKHS. As we stated above, the pth-degree Wiener expansion is the pth-order Volterra series that minimizes the squared error if the input is white gaussian noise with zero mean. This can be put into the regression framework: assume we generate white gaussian noise with zero mean, feed it into the unknown system, and measure its output. Since any finite Volterra series can be represented as a linear operator in the corresponding RKHS, we can find the pth-order Volterra series that minimizes the squared error by linear regression. This, by definition, must be the pth-degree Wiener series since no other Volterra series has this property.[10] . From equations 2.7 and 3.6, we obtain the following expressions for the implicit Wiener series,

$$G_0(\mathbf{x}) = \frac{1}{N}\mathbf{y}^\top \mathbf{1} \quad \text{and} \quad \sum\nolimits_{n=0}^{p} G_n(\mathbf{x}) = \sum\nolimits_{n=0}^{p} H_n(\mathbf{x}) = \mathbf{y}^\top K_p^{-1}\, \mathbf{k}^{(p)}(\mathbf{x}), \tag{3.16}$$

where the Gram matrix K_p and the coefficient vector $\mathbf{k}^{(p)}(\mathbf{x})$ are computed using the kernel from equation 3.11 and $\mathbf{1} = (1, 1, \ldots)^\top \in \mathbb{R}^N$. The system is now represented as a linear combination of kernels evaluated at the

[10] Assuming symmetrized Volterra kernels, which can be obtained from any Volterra expansion.

training points instead of a linear combination of monomials; that is, the Wiener series and its Volterra functionals are represented only implicitly. Thus, there is no need to compute the possibly large number of coefficients explicitly.

The explicit Volterra and Wiener expansions can be recovered at least in principle from equation 3.16 by collecting all terms containing monomials of the desired order and summing them up. The individual nth-order Volterra operators ($p > 0$) are given implicitly by

$$H_n(\mathbf{x}) = a_n \mathbf{y}^\top K_p^{-1} \mathbf{k}_n(\mathbf{x}), \tag{3.17}$$

with $\mathbf{k}_n(\mathbf{x}) = ((\mathbf{x}_1^\top \mathbf{x})^n, (\mathbf{x}_2^\top \mathbf{x})^n, \ldots, (\mathbf{x}_N^\top \mathbf{x})^n)^\top$.[11] For $p = 0$, the only term is the constant zero-order Volterra operator $H_0(\mathbf{x}) = G_0(\mathbf{x})$. The coefficient vector $\eta_n = (h^{(n)}_{1,1,\ldots,1}, h^{(n)}_{1,2,\ldots,1}, h^{(n)}_{1,3,\ldots,1}, \ldots)^\top$ of the explicit Volterra operator is obtained as

$$\eta_n = a_n \Phi_n^\top K_p^{-1} \mathbf{y}, \tag{3.18}$$

using the design matrix $\Phi_n = (\phi_n(\mathbf{x}_1), \phi_n(\mathbf{x}_2), \ldots, \phi_n(\mathbf{x}_N))^\top$. Note that these equations are also valid for infinite polynomial kernels such as $k^{(\infty)}$ or k_{Vovk}. Similar findings are known from the neural network literature where Wray and Green (1994) showed that individual Volterra operators can be extracted from certain network models with sigmoid activation functions that correspond to infinite Volterra series.

The individual Wiener operators can be recovered only by applying the regression procedure twice. If we are interested in the nth-degree Wiener operator, we have to compute the solution for the kernels $k^{(n)}(\mathbf{x}_1, \mathbf{x}_2)$ and $k^{(n-1)}(\mathbf{x}_1, \mathbf{x}_2)$. The Wiener operator for $n > 0$ is then obtained from the difference of the two results as

$$\begin{aligned} G_n(\mathbf{x}) &= \sum\nolimits_{i=0}^{n} G_i(\mathbf{x}) - \sum\nolimits_{i=0}^{n-1} G_i(\mathbf{x}) \\ &= \mathbf{y}^\top \left[K_n^{-1}\, \mathbf{k}^{(n)}(\mathbf{x}) - K_{n-1}^{-1}\, \mathbf{k}^{(n-1)}(\mathbf{x}) \right]. \end{aligned} \tag{3.19}$$

The corresponding ith-order Volterra operators of the nth-degree Wiener operator are computed analogous to equations 3.17 and 3.18.

[11] Note that the time complexity of computing the explicit Volterra operator is $O(m^n N^2 + N^3)$, and the corresponding memory complexity is $O(m^n N + N^2)$. Thus, using the implicit estimation method as an intermediate step is still preferable over direct linear regression for $m^n > N$.

3.5 Orthogonality. The resulting Wiener operators must fulfill the orthogonality condition, which in its strictest form states that a pth-degree Wiener operator must be orthogonal to all monomials in the input of lower order. However, we have constructed our operators in a different function basis; we have expanded the pth-order Wiener operators in terms of kernels instead of monomials. Thus, we have to prove the following:

Theorem 1. *The operators obtained from equation 3.19 fulfill the orthogonality condition*

$$E\left[m(x)G_p(x)\right] = 0 \tag{3.20}$$

where E denotes the expectation over the training set and $m(x)$ an rth-order monomial with $r < p$.

Proof. We will show that this a consequence of the least-squares fit of any linear expansion in a set of basis functions of the form of equation 3.10. In the case of the implicit Wiener and Volterra expansions, the basis functions $\varphi_j(\mathbf{x})$ are polynomial kernel functions $k^{(r)}(\mathbf{x}, \mathbf{x}_i)$ evaluated at the training examples $\mathbf{x}_i$.

We denote the error of the expansion as $e(\mathbf{x}) = y - \sum_{j=1}^{M} \alpha_j \varphi_j(\mathbf{x})$. The minimum of the expected quadratic loss with respect to the expansion coefficient α_k is given by

$$\frac{\partial}{\partial \alpha_k} E\,\|e(\mathbf{x})\|^2 = -2E\left[\varphi_k(\mathbf{x})e(\mathbf{x})\right] = 0. \tag{3.21}$$

This means that for an expansion of the type of equation 3.1 minimizing the squared error, the error is orthogonal to all basis functions used in the expansion.

Now let us assume we know the Wiener series expansion (which minimizes the mean squared error) of a system up to degree $p - 1$. The approximation error is then given by the sum of the higher-order Wiener operators $e(\mathbf{x}) = \sum_{n=p}^{\infty} G_n(\mathbf{x})$, so $G_p(\mathbf{x})$ is part of the error. As a consequence of the linearity of the expectation, equation 3.21, implies

$$\sum\nolimits_{n=p}^{\infty} E\left[\varphi_k(\mathbf{x})G_n(\mathbf{x})\right] = 0 \quad \text{and} \quad \sum\nolimits_{n=p+1}^{\infty} E\left[\varphi_k(\mathbf{x})G_n(\mathbf{x})\right] = 0 \tag{3.22}$$

for any φ_k of order less than p. The difference of both equations yields $E\left[\varphi_k(\mathbf{x})G_p(\mathbf{x})\right] = 0$, so that $G_p(\mathbf{x})$ must be orthogonal to any of the lower-order basis functions—to all kernel functions $k^{(r)}(\mathbf{x}, \mathbf{x}_i)$ with order r smaller than p. Since any monomial $m(\mathbf{x}_i)$ of degree $r < p$ evaluated on a training data point $\mathbf{x}_i$ can be expressed as a linear combination of kernel functions up to degree r, orthogonality on the training set must also hold true for any monomial of order $r < p$.

For both regression and orthogonality of the resulting operators, the assumption of white gaussian noise was not required. In practice, this means that we can compute a Volterra expansion according to equation 3.16 for any type of input, not just for gaussian noise. Note, however, that the orthogonality of the operators can be defined only with respect to an input distribution. If we use equation 3.19 for nongaussian input, the resulting operators will still be orthogonal, but with respect to the nongaussian input distribution. The resulting decomposition of the Volterra series into orthogonal operators will be different from the gaussian case. As a consequence, the operators computed according to equation 3.19 will not be the original Wiener operators, but an extension of this concept as proposed by Barrett (1963).

3.6 Regularized Estimation. So far we have not addressed the fourth problem of the cross-correlation procedure: the negligence of measurement noise. The standard approach in machine learning is to augment the mean squared error objective function in equation 3.4 with a penalizing functional Ω, often given as a quadratic form,

$$\Omega = \lambda \, \alpha^\top R \alpha, \quad \lambda > 0, \tag{3.23}$$

with a positive semidefinite matrix R. R is chosen to reflect prior knowledge that can help to discriminate the true signal from the noise. λ controls the trade-off between the fidelity to the data and the penalty term. The resulting Wiener series is given by

$$\sum\nolimits_{n=0}^{p} G_n(\mathbf{x}) = \sum\nolimits_{n=0}^{p} H_n(\mathbf{x}) = \mathbf{y}^\top (K_p + \lambda R)^{-1} \, \mathbf{k}^{(p)}(\mathbf{x}) \tag{3.24}$$

instead of equation 3.16. When choosing $R = I_N$, one obtains standard ridge regression, which leads to smoother, less noise-sensitive solutions by limiting their RKHS norm. Alternatively, Nowak (1998) suggested selectively penalizing noise-contaminated signal subspaces by a suitable choice of R for the estimation of Volterra series.

Regularization also offers possibilities for compensating for some of the difficulties associated when using higher-order polynomials for regression, such as their poor extrapolation capabilities, or the notoriously bad conditioning of the Gram matrix. As a result, the prediction performance of polynomials on many standard data sets is worse than that of other function bases such as gaussians. However, by a suitable choice of the regularization matrix R, these problems can be alleviated (Franz, Kwon, Rasmussen, & Schölkopf, 2004). Moreover, there is a close correspondence between regularized regression in RKHS and gaussian process regression (Wahba, 1990; Schölkopf & Smola, 2002; Rasmussen & Williams, 2006). This correspondence can be used to approximate arbitrary other kernels by polynomial kernels (Gehler & Franz, 2006).

If one is interested in single Wiener operators, the regularized estimation has a decisive disadvantage: the operators computed according to equation 3.19 are no more orthogonal. However, orthogonality can be still enforced by considering the (smoothed) output of the regularized Wiener system on the training set

$$\tilde{y} = \mathbf{y}^\top (K_p + \lambda R)^{-1} K, \tag{3.25}$$

as a modified, "noise-corrected" training set for equation 3.19, which becomes

$$G_n(\mathbf{x}) = \mathbf{y}^\top (K_p + \lambda R)^{-1} K \big[K_n^{-1} \mathbf{k}^{(n)}(\mathbf{x}) - K_{n-1}^{-1} \mathbf{k}^{(n-1)}(\mathbf{x})\big]. \tag{3.26}$$

The resulting Wiener operators are an orthogonal decomposition of the regularized solution over the training set.

4 Experiments[12]

The principal advantage of our new representation of the Volterra and Wiener series is its capability of implicitly handling systems with high-dimensional input. We will demonstrate this in a reconstruction task of a fifth-order receptive field. Before doing so, we compare the estimation performance of the kernelized technique to previous approaches.

4.1 Comparison to Previous Estimation Techniques. Our first data set comes from a calibration task for a CRT monitor used to display stimuli in psychophysical experiments. The data were generated by displaying a gaussian noise pattern ($\mathcal{N}(128, 64^2)$) on the monitor, which was recorded by a cooled change-coupled device camera operating in its linear range. The system identification task is to quantify the nonlinear distortion of the screen and the possible interaction with previous pixels on the same scan line. The input data were generated by sliding a window of fixed length m in scanning direction over the lines of the gaussian input pattern; the system output value is the measured monitor brightness at the screen location corresponding to the final pixel of the window.

We used three techniques to fit a Wiener model: (1) Classical cross-correlation with model orders 1, 2, and 3 and window size 1 to 4; (2) direct

[12] Code for the implicit estimation of Volterra series can be found online at http://www.kyb.tuebingen.mpg.de/bs/people/mof/code.

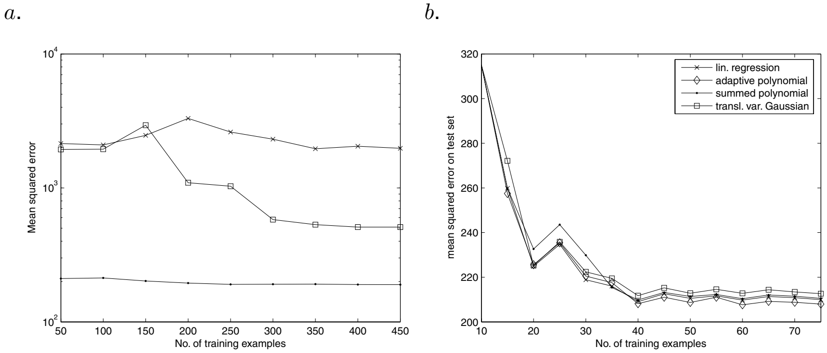

Figure 1 Mean squared error on the test set for varying training set size. (a) First- (x) and second-order (squares) cross-correlation leads to test errors orders of magnitude higher than the regression techniques (dots). (b) Performance of the tested regression techniques (see the key) for training set size below 75.

linear regression with monomials as basis functions; and (3) kernel regression with the adaptive polynomial 3.11, the inhomogeneous polynomial 3.12, and the infinite Volterra series kernel of equation 3.14. For techniques 2 and 3, we used the standard ridge regularizer $R = I_M$ and $R = I_N$, respectively. The regularization parameter λ in equation 3.23, the weights a_i in the adaptive polynomial kernel 3.11, the window size m, and the model order p were found by minimizing the analytically computed leave-one-out error (Vapnik, 1982). We varied the number of training examples from 10 to 1000 to characterize the convergence behavior of the different techniques. The independent test set always contained 1000 examples.

As the result shows (see Figure 1a), the mean squared error on the test set decreases at a significantly faster rate for the regression methods due to the unfavorable properties of the cross-correlation estimator. In fact, a comparable test error could not be reached even for the maximal training set size of 1000 (not contained in the figure). We display the cross-correlation results only for $m = 2$ and $p = 1, 2$, which had the lowest test error. The third-order cross-correlation produced test (and training) errors above 10^5 on this data set.

We observe small but significant differences between the tested regression techniques due to the numerical conditioning of the required matrix inversion (see Figure 1b). For a training set size above 40, the adaptive polynomial kernel performs consistently better since the weights a_i can be adapted to the specific structure of the problem. Interestingly, the infinite Volterra kernel shows a consistently lower performance in spite of the higher approximation capability of its infinite-dimensional RKHS.

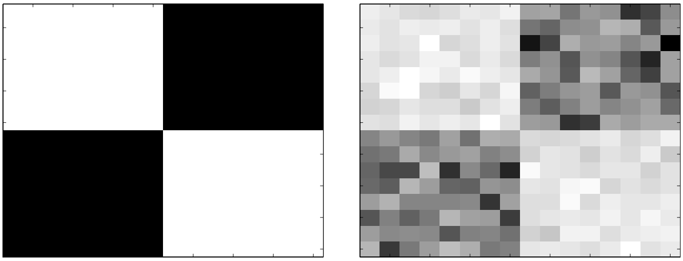

Figure 2 (Left) 16 × 16 linear kernel of the test system. (Right) Reconstructed linear kernel from the fifth-order Volterra kernel by computing a preimage (after 2500 samples).

4.2 Reconstruction of a Fifth-Order Linear-Nonlinear Cascade. This experiment demonstrates the applicability of the proposed method to high-dimensional input. Our example is the fifth-order LN cascade system $y = (\sum_{k,l=1}^{16} h_{kl} x_{kl})^5$ that acts on 16 × 16 image patches by convolving them with a linear kernel h_{kl} of the same size shown in Figure 2 (left) before the nonlinearity is applied. We generated 2500 image patches containing uniformly distributed white noise and computed the corresponding system output to which we added 10% gaussian measurement noise. The resulting data were used to estimate the implicit Wiener expansion using the inhomogeneous polynomial kernel 3.12. In classical cross-correlation and linear regression, this would require the computation of roughly 9.5 billion independent terms for the fifth-order Wiener kernel. Moreover, even for much lower-dimensional problems, it usually takes tens of thousands of samples until a sufficient convergence of the cross-correlation technique is reached.

Even if all entries of the fifth-order Wiener kernel were known, it would be still hard to interpret the result in terms of its effect on the input signal. The implicit representation of the Volterra series allows for the use of preimage techniques (e.g., Schölkopf & Smola, 2002) where one tries to choose a point $\mathbf{z}$ in the input space such that the nonlinearly mapped image in $\mathbb{F}$, $\phi(\mathbf{z})$, is as close as possible to the representation in the RKHS. In the case of the fifth-order Wiener kernel, this amounts to representing $H_5[\mathbf{x}]$ by the operator $(\mathbf{z}^\top \mathbf{x})^5$ with an appropriately chosen preimage $\mathbf{z} \in \mathbb{R}^{256}$. The nonlinear map $z \mapsto z^5$ is invertible, so that we can use the direct technique described in Schölkopf and Smola (2002) where one applies the implicitly given Volterra operator from equation 3.17 to each of the canonical base vectors of $\mathbb{R}^{256}$ resulting in a 256-dimensional response vector $\mathbf{e}$. The preimage is obtained as $\mathbf{z} = \sqrt[5]{\mathbf{e}}$. The result in Figure 2 (right) demonstrates that the original linear kernel is already recognizable after using 2500 samples. The example shows that preimage techniques are capable of revealing the input structures to which the Volterra operator is tuned, similar to the classical analysis techniques in linear systems.

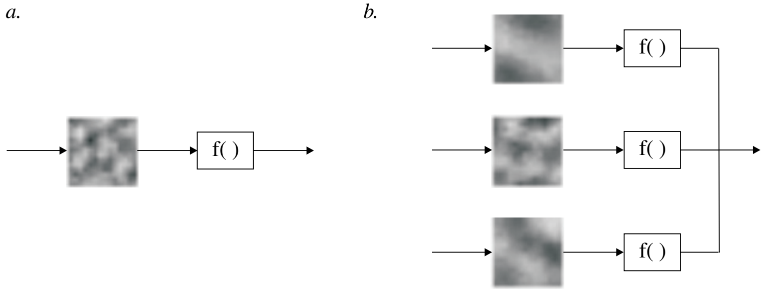

Figure 3 Representation of a Volterra or Wiener system by (a) a cascade of a linear system (preimage) and a static nonlinearity $f(x)$ (e.g., $(1+x)^p$ or e^x, depending on the choice of the kernel) and (b) a set of several parallel cascades (reduced set).

5 Conclusion

We have presented a unifying view of the traditional Wiener and Volterra theory of nonlinear systems and newer developments from the field of kernel methods. We have shown that all properties of discrete Volterra and Wiener theory are preserved by using polynomial kernels in a regularized regression framework. The benefits of the new kernelized representation can be summarized as follows:

1. The implicit estimation of the Wiener and Volterra series allows for system identification with high-dimensional input signals. Essentially, this is due to the representer theorem: although a higher-order series expansion contains a huge number of coefficients, it turns out that when estimating such a series from a finite sample, the information in the coefficients can be represented more parsimoniously using an example-based implicit representation.

2. The complexity of the estimation process is independent of the order of nonlinearity. Even infinite Volterra series expansions can be estimated.

3. Regularization techniques can be naturally included in the regression framework to accommodate measurement noise in the system outputs. As we have shown, one still can extract the corresponding Wiener operators from the regularized kernel solution while preserving their orthogonality with respect to the input. The analysis of a system in terms of subsystems of different orders of nonlinearity can thus be extended to noisy signals.

4. Preimage techniques reveal input structures to which Wiener or Volterra operators are tuned. These techniques try to represent the

system by a cascade consisting of a linear filter followed by a static nonlinearity (see Figure 3a).

5. As in standard linear regression, the method also works for nongaussian input. At the same time, convergence is considerably faster than in the classical cross-correlation procedure because the estimation is done directly on the data. Both regression methods omit the intermediate step of estimating cross-correlations, which converges very slowly.

Although we obtained useful experimental results for problems where the number of terms in the Wiener expansion largely exceeds the number of training examples, this will not always be feasible, for example, in cases where the Volterra kernels cannot be approximated by smooth functions. In this sense, we cannot circumvent the curse of dimensionality. However, if the system to be identified has a suitably smooth underlying structure, the proposed technique (in particular, the regularized variant) can take advantage of it.

The preimage method in our experiment works only for Volterra kernels of odd order. More general techniques exist (Schölkopf & Smola, 2002), including the case of other kernels and the computation of approximations in terms of parallel cascades of preimages and nonlinearities (reduced sets; cf. Figure 3b). In the case of a second-order system, the reduced set corresponds to an invariant subspace of the Volterra operator (cf. Hyvärinen & Hoyer, 2000). It can be shown that the entire class of discrete Volterra systems can be approximated by cascades where the nonlinearities are polynomials of sufficient degree (Korenberg, 1983) and that any doubly-finite discrete Volterra series can be exactly represented by a finite sum of such cascades (Korenberg, 1991). There also exists an iterative technique of directly fitting such cascade expansions to the training data by choosing the preimages from a fixed set of candidates generated from the data and adapting the nonlinearity (Korenberg, 1991). Each iteration requires only the inversion of a $(p + 1) \times (p + 1)$ matrix (p being the degree of nonlinearity); thus, convergence can be very fast depending on the training data. The resulting parallel cascades will generally be different from reduced-set expansions, which have a fixed number of elements and a prescribed nonlinearity defined by the kernel. However, both cascade representations can be converted into their corresponding Volterra series expansions, which makes a comparison of the results possible.

Having seen that Volterra and Wiener theory can be treated just as a special case of a kernel regression framework, one could argue that this theory is obsolete in modern signal analysis. This view is supported by the fact that on many standard data sets for regression, polynomial kernels are outperformed by other kernels, such as the gaussian kernel. So why do we not replace the polynomial kernel by some more capable kernel and forget about Wiener and Volterra theory altogether? There are at least two arguments against this point of view. First, our study has shown that in contrast to other kernels, polynomial kernel solutions can be directly transformed into their corresponding Wiener or Volterra representation. Many entries of the Volterra kernels have a direct interpretation in signal processing applications (examples in Mathews & Sicuranza, 2000). This interpretability is lost when other kernels are used. Second, Wiener expansions decompose a signal according to the order of interaction of its input elements. In some applications, it is important to know how many input elements interact in the creation of the observed signals, such as in the analysis of higher-order statistical properties (an example on higher-order image analysis can be found in Franz & Schölkopf, 2005).

Acknowledgments

The ideas presented in this article have greatly profited from discussions with P. V. Gehler, G. Bakır, and M. Kuss. This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. The publication reflects only the authors' views.

References

Ahmed, N. U. (1970). Closure and completeness of Wiener's orthogonal set G_n in the class $L^2(\Omega, B, \mu)$ and its application to stochastic hereditary differential systems. *Information and Control, 17*, 161–174.

Alper, A. (1965). A consideration of the discrete Volterra series. *IEEE Trans. Autom. Control, AC-10*(3), 322–327.

Barrett, J. F. (1963). The use of functionals in the analysis of non-linear physical systems. *J. Electron. Control, 15*, 567–615.

Boyd, S., & Chua, L. O. (1985). Fading memory and the problem of approximating nonlinear operators with Volterra series. *IEEE Trans. Circuits. Syst., CAS-32*, 1150–1161.

Brilliant, M. B. (1958). *Theory of the analysis of nonlinear systems* (RLE Tech. Rep. No. 345). Cambridge, MA: MIT.

Dodd, T. J., & Harrison, R. F. (2002). A new solution to Volterra series estimation. In *CD-ROM Proc. 2002 IFAC World Congress*. Oxford: Elsevier.

Franz, M. O., Kwon, Y., Rasmussen, C. E., & Schölkopf, B. (2004). Semi-supervised kernel regression using whitened function classes. In C. E. Rasmussen, H. H. Bülthoff, M. A. Giese, & B. Schölkopf (Eds.), *Pattern recognition: Proc. 26th DAGM Symposium* (pp. 18–26). Berlin: Springer.

Franz, M. O., & Schölkopf, B. (2004). Implicit estimation of Wiener series. In A. Barros, J. Principe, J. Larsen, T. Adali, & S. Douglas (Eds.), *Proc. 2004 IEEE Signal Processing Society Workshop* (pp. 735–744). New York: IEEE.

Franz, M. O., & Schölkopf, B. (2005). Implicit Wiener series for higher-order image analysis. In L. K. Saul, Y. Weiss, & L. Bottou (Eds.), *Advances in neural information processing systems, 17* (pp. 465–472). Cambridge, MA: MIT Press.

Fréchet, M. (1910). Sur les fonctionelles continues. *Annales Scientifiques de L'Ecole Normale Supérieure, 27*, 193–216.

Gehler, P. V., & Franz, M. O. (2006). *Implicit Wiener Series. Part II: Regularised estimation* (MPI Tech. Rep. No. 148). Tübingen, Germany: Max-Planck-Institute for Biological cybernetics.

Giannakis, G. B., & Serpedin, E. (2001). A bibliography on nonlinear system identification. *Signal Processing, 81*, 533–580.

Hille, E., & Phillips, R. S. (1957). *Functional analysis and semi-groups*. Providence, RI: American Mathematical Society.

Hyvärinen, A., & Hoyer, P. (2000). Emergence of phase- and shift-invariant features by decomposition of natural images into independent feature subspaces. *Neural Computation, 12*, 1705–1720.

Korenberg, M. J. (1983). Statistical identification of parallel cascades of linear and nonlinear systems. In *Proc. IFAC Symp. Identification and System Parameter Estimation* (pp. 669–674).

Korenberg, M. J. (1991). Parallel cascade identification and kernel estimation for nonlinear systems. *Ann. Biomed. Eng., 19*(4), 429–455.

Korenberg, M. J., Bruder, S. B., & McIlroy, P. J. (1988). Exact orthogonal kernel estimation from finite data records: Extending Wiener's identification of nonlinear systems. *Ann. Biomed. Eng., 16*(2), 201–214.

Korenberg, M. J., & Hunter, I. W. (1990). The identification of nonlinear biological systems: Wiener kernel approaches. *Ann. Biomed. Eng., 18*, 629–654.

Lee, Y. W., & Schetzen, M. (1965). Measurement of the Wiener kernels of a non-linear system by crosscorrelation. *Intern. J. Control, 2*, 237–254.

Liusternik, L., & Sobolev, V. (1961). *Elements of functional analysis*. New York: Unger.

Mathews, V. J., & Sicuranza, G. L. (2000). *Polynomial signal processing*. New York: Wiley.

Nowak, R. (1998). Penalized least squares estimation of Volterra filters and higher order statistics. *IEEE Trans. Signal Proc., 46*(2), 419–428.

Ogura, H. (1972). Orthogonal functionals of the Poisson process. *IEEE Trans. Inf. Theory, 18*(4), 473–481.

Palm, G. (1978). On representation and approximation of nonlinear systems. *Biol. Cybern., 31*, 119–124.

Palm, G., & Poggio, T. (1977). The Volterra representation and the Wiener expansion: Validity and pitfalls. *SIAM J. Appl. Math., 33*(2), 195–216.

Palm, G., & Poggio, T. (1978). Stochastic identification methods for nonlinear Systems: An extension of Wiener theory. *SIAM J. Appl. Math., 34*(3), 524–534.

Papoulis, A. (1991). *Probablity, random variables and stochastic processes*. New York: McGraw-Hill.

Poggio, T. (1975). On optimal nonlinear associative recall. *Biol. Cybern., 19*, 201–209.

Prenter, P. M. (1970). A Weierstrass theorem for real, separable Hilbert spaces. *J. Approx. Theory, 3*, 341–351.

Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian processes for machine learning*. Cambridge, MA: MIT Press.

Rugh, W. J. (1981). *Nonlinear system theory*. Baltimore: Johns Hopkins University Press.

Saunders, C., Stitson, M. O., Weston, J., Bottou, L., Schölkopf, B., & Smola, A. (1998). *Support vector machine—reference manual* (Tech. Rep.). Egham, UK: Department of Computer Science, Royal Holloway, University of London.

Schetzen, M. (1965). A theory of nonlinear system identification. *Intl. J. Control, 20*(4), 577–592.

Schetzen, M. (1980). *The Volterra and Wiener theories of nonlinear systems*. Malabar, FL: Krieger.

Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels*. Cambridge, MA: MIT Press.

Segall, A., & Kailath, T. (1976). Orthogonal functionals of independent-increment processes. *IEEE Trans. Inf. Theory, 22*(3), 287–298.

Steinwart, I. (2001). On the influence of the kernel on the consistency of support vector machines. *JMLR, 2*, 67–93.

Vapnik, V. (1982). *Estimation of dependencies based on empirical data*. New York: Springer.

Volterra, V. (1887). Sopra le funzioni che dipendono de altre funzioni. In *Rend. R. Academia dei Lincei 2° Sem.*, pp. 97–105, 141–146, 153–158.

Volterra, V. (1959). *Theory of functionals and of integral and integro-differential equations*. New York: Dover.

Wahba, G. (1990). *Spline models for observational data*. Philadelphia: Society for Industrial and Applied Mathematics.

Wiener, N. (1958). *Nonlinear problems in random theory*. New York: Wiley.

Wray, J., & Green, G. G. R. (1994). Calculation of the Volterra kernels of non-linear dynamic systems using an artificial neural network. *Biol. Cybern., 71*, 187–195.

Received November 1, 2005; accepted April 5, 2006.