Max–Planck–Institut für biologische Kybernetik
Max Planck Institute for Biological Cybernetics

Technical Report No. TR-114

Implicit Wiener Series

Part I: Cross-Correlation vs. Regression in Reproducing Kernel Hilbert Spaces

Matthias O. Franz,[1] Bernhard Schölkopf[1]

June 2003

[1] Department Schölkopf, email: matthias.franz@tuebingen.mpg.de

This report is available in PDF–format via anonymous ftp at ftp://ftp.kyb.tuebingen.mpg.de/pub/mpi-memos/pdf/iws.pdf. The complete series of Technical Reports is documented at: http://www.kyb.tuebingen.mpg.de/techreports.html

Implicit Wiener Series

Part I: Cross-Correlation vs. Regression in Reproducing Kernel Hilbert Spaces

Matthias O. Franz, Bernhard Schölkopf

Abstract. The Wiener series is one of the standard methods to systematically characterize the nonlinearity of a neural system. The classical estimation method of the expansion coefficients via cross-correlation suffers from severe problems that prevent its application to high-dimensional and strongly nonlinear systems. We propose a new estimation method based on regression in a reproducing kernel Hilbert space that overcomes these problems. Numerical experiments show performance advantages in terms of convergence, interpretability and system size 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 in- and outgoing signals. If the system is linear, it can be characterized uniquely by measuring its impulse response, for instance by reverse correlation. For nonlinear systems, however, there exists a whole variety of system representations. One of them, the *Wiener expansion*, has found a somewhat wider use in neuroscience since its estimation constitutes a natural extension of linear system identification (e.g., Rieke, Warland, de Ruyter van Steveninck, & Bialek, 1996; Schetzen, 1989; Geiger & Poggio, 1975). The coefficients of the Wiener expansion can be estimated by a cross-correlation procedure that is conveniently applicable to experimental data.

Unfortunately, the estimation of the Wiener expansion by cross-correlation suffers from severe problems preventing its application to high-dimensional data and highly nonlinear systems. In the first part of this report, we want to overcome these problems by proposing a different estimation method based on regression in a *reproducing kernel Hilbert space* (RKHS). We will show that the new estimation method is superior to the classical one in terms of convergence, interpretability and system size that can be handled.

In the next section, we introduce the Volterra and Wiener expansion and discuss the problems of the cross-correlation procedure. In Sect. 3, we review linear regression in a RKHS. The new estimation method is described in Sect. 4, followed by some examples of use in Sect. 5. We conclude in Sect. 6 by briefly discussing the results and possible improvements of the new estimation procedure. Issues concerning the regularized estimation of the implicit Wiener series will follow in the second part of this report.

2 Volterra and Wiener theories of nonlinear systems

A system can be defined mathematically as a rule that assigns an output signal $\mathbf{y}$ to an input signal $\mathbf{x}$. This rule can be expressed in the form

$$\mathbf{y} = T[\mathbf{x}]$$

using the system operator T. A large class of systems can be characterized by their Wiener series expansion where the system operator consists of a linear combination of monomials in the components of theinput vector $\mathbf{x}$. Roughly speaking, the so-called Wiener class encompasses all systems with scalar outputs that are time-invariant with finite memory. This class is described by the Wiener theory of nonlinear systems which is based on the orthogonalization of a specific, complete set of time-invariant operators called the *Volterra operators*.

2.1 Volterra systems

Originally, Volterra operators are defined for continuous, scalar output time functions $y(t)$ and input time functions $x(t)$ (t stands for the time variable). Subject to certain restrictions, one can show that a time-invariant system

operator T can expressed as a series of integral operators

$$\begin{aligned} y(t) = h^{(0)} &+ \int_{\mathbb{R}} h^{(1)}(\tau_1)x(t-\tau_1)\,d\tau_1 \\ &+ \int_{\mathbb{R}^2} h^{(2)}(\tau_1,\tau_2)x(t-\tau_1)x(t-\tau_2)\,d\tau_1 d\tau_2 \\ &+ \int_{\mathbb{R}^3} h^{(3)}(\tau_1,\tau_2,\tau_3)x(t-\tau_1)x(t-\tau_2)x(t-\tau_3)\,d\tau_1 d\tau_2 d\tau_2 \\ &+ \cdots \end{aligned} \tag{1}$$

in which $h^{(0)}$ is a constant and for $n = 1, 2, ..$

$$h^{(n)}(\tau_1, ..., \tau_n) = 0 \quad \text{for any } \tau_j < 0, \quad j = 1, 2, 3, ..., n \tag{2}$$

such that all operators are causal, i.e., they do not depend on future values of $x(t)$ (Volterra, 1887). The series in Eq. (1) is called *Volterra series* and the functions $h^{(n)}(\tau_1, .., \tau_n)$ are the *Volterra kernels* of the system. Note that the Volterra kernels for a given output are not unique. There are, in fact, many asymmetric (with respect to permutations of the τ_i, i.e., $h^{(n)}(\ldots, \tau_i, \ldots, \tau_j, \ldots) \neq h^{(n)}(\ldots, \tau_j, \ldots, \tau_i, \ldots)$) Volterra kernels which give rise to the same operator, but every system operator corresponds only to one symmetric kernel. Fortunately, any asymmetric kernel can be symmetrized by a simple procedure (Schetzen, 1989). Therefore, we can assume symmetric kernels and unique Volterra expansions without loss of generality throughout this text.

Another way of expressing Eq. (1) is

$$y(t) = H_0[x(t)] + H_1[x(t)] + H_2[x(t)] + \cdots + H_n[x(t)] + \cdots \tag{3}$$

in which $H_0[x(t)] = h^{(0)}$ and

$$H_n[x(t)] = \int_{\mathbb{R}^n} h^{(n)}(\tau_1, .., \tau_n)x(t-\tau_1)\ldots x(t-\tau_n)\,d\tau_1 \ldots d\tau_n \tag{4}$$

is the nth-order *Volterra functional*. The application of the Volterra series in the continuous time form of Eq. (1) is mostly limited to the analysis of nonlinear differential equations. In practical signal processing, one uses a discretized form for a finite sample of data where the Volterra functionals are expressed as

$$H_n[\mathbf{x}] = \sum_{i_1=1}^{m} \cdots \sum_{i_n=1}^{m} h^{(n)}_{i_1\ldots i_n} x_{i_1} \ldots x_{i_n}. \tag{5}$$

Here, we assume that the input data is given as a vector $\mathbf{x} = (x_1, \ldots, x_m)^\top \in \mathbb{R}^m$. The vectorial data can be generated from any multi-dimsional input or, for instance, by a sliding window on a discretized time series. The discretized nth-order Volterra kernel is given as a finite number of m^n coefficients $h^{(n)}_{i_1\ldots i_n}$[1]. The discretized nth-order Volterra functional (5) is, accordingly, a linear combination of all ordered nth-order monomials of the components of $\mathbf{x}$. Clearly, the discretized Volterra functionals provide a practical approximation which shares the completeness and convergence properties of Volterra and Wiener theory only in the continuous limit.

Due to its power series character, the convergence of an infinite Volterra series usually is only guaranteed for input signals of sufficiently small amplitude (Schetzen, 1989). This limitation is circumvented by Wiener theory, described in the next section.

2.2 Wiener systems

Convergence of the Volterra series is comparable to the convergence of the Taylor series expansion of a function which often allows only for small deviations from the starting point. In fact, the Volterra series can be seen as a Taylor series with memory. The type of convergence required is very stringent since not only the error has to approach zero with increasing number of terms, but also the derivatives of the error. A less stringent notion of convergence applies if one represents a function by a series of orthogonal functions, namely only convergence in

[1] In a symmetric Volterra kernel, two coefficients are the same when their indices are permuted. The number of independent coefficients reduces to $(n+m-1)!/(n!(m-1)!)$ in this case.

the mean square sense which allows convergence over a much larger range than the Taylor series. The same idea can be applied to functionals instead of functions by using an orthogonal series of base functionals and stipulating convergence in the mean square sense.

The output of two different functionals in a Volterra series, however, is usually not orthogonal, i.e., their respective output is correlated. They can be orthogonalized with respect to a given distribution on the input signals by a procedure which is very similar to a Gram-Schmidt orthogonalization. The resulting functionals are sums of Volterra functionals of different order, or *nonhomogeneous Volterra functionals*. If the orthogonalization is done with respect to a white Gaussian input distribution with zero mean, one obtains the so-called *Wiener functionals* denoted by $G_n[x(t)]$ (Wiener, 1958). The class of systems for which the Wiener expansion

$$y(t) = \sum_{n=0}^{\infty} G_n[x(t)] \tag{6}$$

converges in the least squares sense is the above mentioned Wiener class that contains all "nonexplosive" (i.e., the system response must have finite variance for the Gaussian input) systems with finite memory (Schetzen, 1989). Examples of systems with infinite memory are all systems with more than one stable attractor that can be switched from one attractor to another by the input. This occurs for instance in systems described by nonlinear differential equations with more than one limit point or limit cycle. Whereas these systems still can be approximately dealt with by suitably restricting the input to prevent switching, this becomes completely impossible in chaotic systems. Here, the present output never becomes independent of its initial state, and even infinitesimal differences in the initial state lead to finite differences in the system output after sufficient time.

In addition to the larger range of convergence, the Wiener formulation provides a direct way of estimating the Volterra kernels from data in a system identification task. In such a setup, the system to be identified is thought to be a black box for which the input and output function are known. The system identification consists in finding the Wiener representation of the unknown system. If the input is a white Gaussian process with variance A, it can be shown (Schetzen, 1989) that the nth-order Volterra kernels k_n of the nth-degree Wiener functionals, the *leading Wiener kernels* or simply Wiener kernels, are given by the cross-correlations

$$\begin{aligned}
k^{(0)} &= \overline{y(t)} && (7)\\
k^{(1)}(\sigma_1) &= \frac{1}{A}\overline{y(t)x(t-\sigma_1)} && (8)\\
k(2)(\sigma_1,\sigma_2) &= \frac{1}{2A^2}\overline{y(t)x(t-\sigma_1)x(t-\sigma_2)} && (9)\\
k^{(3)}(\sigma_1,\sigma_2,\sigma_3) &= \frac{1}{3!A^3}\overline{y(t)x(t-\sigma_1)x(t-\sigma_2)x(t-\sigma_3)} && (10)\\
&\vdots\\
k^{(n)}(\sigma_1,\ldots,\sigma_n) &= \frac{1}{n!A^n}\overline{y(t)x(t-\sigma_1)\ldots x(t-\sigma_n)}. && (11)
\end{aligned}$$

where the bar indicates the average over time. Since the Wiener functionals are orthogonal by definition, the single Wiener kernels can be estimated independently.

Besides the Wiener kernel, every Wiener functional of degree $p > 1$ consists of a varying number of lower order Volterra operators containing the so-called *derived Wiener kernels*. These are derived from the leading Wiener kernel by the orthogonalization procedure. The zeroth- and first-degree Wiener functionals do not contain derived Wiener kernels and are given by

$$G_0[x(t)] = k^{(0)} \quad \text{and} \quad G_1[x(t)] = \int_{\mathbb{R}} k^{(1)}(\tau_1)x(t-\tau_1)\,d\tau_1. \tag{12}$$

For pth-order Wiener functionals G_p, the derived Wiener kernels $k^{(p)}_{p-2m}$ can be computed from the leading Wiener kernel k_p using the formula (Schetzen, 1989)

$$k^{(p)}_{p-2m}(\sigma_1,\ldots,\sigma_{p-2m}) = \frac{(-1)^m p! A^m}{(p-2m)!m!2^m}\cdot \int_{\mathbb{R}^m} k_p(\tau_1,\tau_1,\ldots,\tau_m,\tau_m,\sigma_1,\ldots,\sigma_{p-2m})\,d\tau_1\ldots d\tau_m. \tag{13}$$

For instance, the Volterra expansion of the second-degree Wiener functional

$$G_2[x(t)] = \int_{\mathbb{R}^2} k_2(\tau_1, \tau_2) x(t-\tau_1) x(t-\tau_2)\, d\tau_1 d\tau_2 - A \int_{\mathbb{R}} k_2(\tau_1, \tau_1)\, d\tau_1 \tag{14}$$

consists of a zero-order and a second-order Volterra functional, the third-degree Wiener functional

$$G_3[x(t)] = \int_{\mathbb{R}^3} k_3(\tau_1, \tau_2, \tau_3) x(t-\tau_1) x(t-\tau_2) x(t-\tau_3)\, d\tau_1 d\tau_2 d\tau_2 \\ - 3A \int_{\mathbb{R}^2} k_3(\tau_1, \tau_2, \tau_2) x(t-\tau_1)\, d\tau_1 d\tau_2 \tag{15}$$

of a first- and third order Volterra functional. In general, an odd-degree Wiener functional contains all lower odd-order Volterra functionals, an even-degree Wiener functional all lower even-order Volterra functionals.

Wiener and Volterra series can be viewed as two equivalent ways of characterizing a system. Both use monomials as base functions, but they group them into either nonhomogeneous or homogeneous Volterra functionals. Therefore, one representation can be converted into the other. The Volterra expansion of a Wiener series can be computed easily by adding up all operators of equal order. The Wiener expansion can be obtained by applying Gram-Schmidt orthogonalization to a Volterra series (Schetzen, 1989), but this procedure becomes rather tedious for higher-order functionals. Fortunately, the Wiener expansion can also be computed directly from the Volterra kernel H_p using (Schetzen, 1989)

$$H_p[x(t)] = \sum_{m=0}^{[p/2]} G_{p-2m}[x(t)] \tag{16}$$

where $[..]$ denotes integer truncation. The leading Wiener kernel of the Wiener functionals G_{p-2m} is derived from the Volterra kernel h_p according to

$$k^{(p-2m)}(\sigma_1, \ldots, \sigma_{p-2m}) = \frac{p! A^m}{(p-2m)! m! 2^m} \cdot \\ \int_{\mathbb{R}^m} h_p(\tau_1, \tau_1, \ldots, \tau_m, \tau_m, \sigma_1, \ldots, \sigma_{p-2m})\, d\tau_1 \ldots d\tau_m. \tag{17}$$

The derived Wiener kernels of H_p can be obtained from the leading kernel using Eq. (13).

2.3 Properties and problems of Wiener systems

Before we discuss the practical problems that arise during the computation of a Wiener series representation of a system, we summarize the most important properties of the Wiener expansion:

1. The pth-degree Wiener expansion of a system is the sum of Volterra operators of order up to p which minimizes the mean square error between true system output and its Volterra representation if the input is zero-mean, white Gaussian noise.

2. The Wiener functionals are orthogonal if the input is zero-mean, white Gaussian noise.

3. pth-degree Wiener functionals generally consist of several Volterra functionals up to order p.

4. The leading Wiener kernels can be computed by cross-correlating the system output with products of the input (cf. Eqns. (7) - (11)). The derived Wiener kernels are computed from the leading kernel using the orthogonality condition.

These properties will play an important role in the remainder of the text since we propose an alternative way of computing the Wiener series. We will show that the resulting expansions still fulfill the properties described above.

The practical computation of the Wiener expansion via cross-correlation poses some serious problems:

1. In practice, the cross-correlations have to be estimated at a finite resolution (cf. the discretized version of the Volterra operator in Eq. (5)). The number of expansion coefficients $h^{(n)}_{i_1 \ldots i_n}$ in Eq. (5) increases with m^n for an m-dimensional input signal and an nth-order Volterra kernel. However, the number of coefficients that

actually have to be estimated by cross-correlation is smaller. Since the products in Eq. (5)) remain the same when two different indices are permuted, the associated coefficients should be equal. As a consequence, the required number of measurements is $(n+m-1)!/(n!(m-1)!)$ (Schetzen, 1989). Nonetheless, the resulting numbers are huge for higher-order Wiener kernels. For instance, a 5th-order Wiener kernel operating on 16×16 sized image patches 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.

2. The estimation of cross-correlations requires large sample sizes. Typically, one needs several tens of thousands input-output pairs before a sufficient convergence is reached. Moreover, the variance of the estimator $\overline{y(t)x(t-\sigma_1)\dots x(t-\sigma_n)}$ in Eq. (11) increases with increasing values of the σ_i (Papoulis, 1991) such that only operators with relatively small memory can be reliably estimated.

3. Even if the Wiener kernels are known, they provide no obvious interpretation on which features of the input the associated Wiener functionals operate. In contrast, by looking at the impulse response of a linear system, one immediately sees to which features of the input it is tuned. In image processing, for instance, the structure of the filter mask reveals whether it is tuned to small scale edge elements or to image patches of constant brightness.

4. The estimation via cross-correlation works only if the input is Gaussian noise with zero mean, not for general types of input.

In this study, we propose a different method for estimating the Wiener expansion that overcomes the discussed practical problems. The proposed method relies on a regression technique taken from the field of kernel methods which is widely used in the machine learning community. This technique is the subject of the next section.

3 Linear regression in RKHS

3.1 Linear regression

The regression technique we will use as a new approach to estimating Wiener expansions is based on classical linear regression. We will review this method, since some of its properties are important for the derivation of its nonlinear extension. As we described above for the estimation of Wiener kernels, the experimental setting is that of system identification: We are given a set of N input values $\mathbf{x}_i \in \mathbb{R}^m$ and the asscociated scalar output values y_i of the system to be identified. In linear regression, we assume a linear relation between input and output of the form

$$y_i = \tilde{\mathbf{g}}^\top \mathbf{x}_i + b, \quad b \in \mathbb{R}, \tilde{\mathbf{g}} \in \mathbb{R}^m. \tag{18}$$

This can be expressed in a more convenient way by setting $\mathbf{g} = (\tilde{\mathbf{g}}^\top, b)^\top$ and collecting the data in the form

$$X = \begin{pmatrix} \mathbf{x}_1^\top & 1 \\ \mathbf{x}_2^\top & 1 \\ \vdots & \vdots \\ \mathbf{x}_N^\top & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} \tag{19}$$

such that $\mathbf{y} = X\mathbf{g}$. The coefficient vector $\mathbf{g}$ is chosen to minimize the quadratic loss function

$$L(\mathbf{g}) = (\mathbf{y} - X\mathbf{g})^\top (\mathbf{y} - X\mathbf{g}) = \mathbf{y}^\top \mathbf{y} - 2\mathbf{y}^\top X\mathbf{g} + \mathbf{g}^\top X^\top X\mathbf{g}. \tag{20}$$

Computing the differential with respect to $\mathbf{g}$ and setting it to zero,

$$dL = (-2X^\top \mathbf{y} + 2X^\top X\mathbf{g})\, d\mathbf{g} = 0 \tag{21}$$

this yields the so-called normal equation

$$X^\top X\mathbf{g} = X^\top \mathbf{y} \tag{22}$$

the solution of which is given by

$$\mathbf{g} = (X^\top X)^{-1} X^\top \mathbf{y} \tag{23}$$

if the inverse of $X^\top X$ exists. Note that the solution is a linear combination of the rows of X, i.e., it is an element of the *row space* $\mathcal{L}(X^\top)$ of X

$$\mathbf{g} \in \mathcal{L}(X^\top) := \{X^\top \mathbf{a} \mid \mathbf{a} \in \mathbb{R}^N\}. \tag{24}$$

This finding is important for the derivation of the linear regression method in nonlinear feature spaces described in the next section.

3.2 Regression in RKHS

We now extend classical linear regression to the case when the output is modeled by linear combinations from a dictionary of fixed nonlinear functions φ_j of the input $\mathbf{x}$, i.e.,

$$y_i = \sum_{j=1}^{M} \gamma_j \varphi_j(\mathbf{x}_i). \tag{25}$$

The number of functions M in the dictionary can be possibly infinite, as, for instance, in a Fourier or wavelet expansion. Alternatively, one can express Eq. (25) by using a nonlinear map $\phi(\mathbf{x}_i) = (\varphi_1(\mathbf{x}_i), \varphi_2(\mathbf{x}_i), \dots)^\top$ from $\mathbb{R}^m$ into some high-dimensional (possibly infinite-dimensional) space $\mathbb{F}$. y_i can be computed by a scalar product[2] with a coefficient vector $\gamma = (\gamma_1, \gamma_2, \dots)^\top \in \mathbb{F}$

$$y_i = \gamma^\top \phi(\mathbf{x}_i). \tag{26}$$

If we put all N mapped samples into an $N \times M$ design matrix Φ with

$$\Phi = \begin{pmatrix} \phi(\mathbf{x}_1)^\top \\ \phi(\mathbf{x}_2)^\top \\ \vdots \\ \phi(\mathbf{x}_N)^\top \end{pmatrix}, \tag{27}$$

the model can be written as

$$\mathbf{y} = \Phi\gamma \tag{28}$$

As before, the regression problem consists in finding the vector γ that minimizes the squared error. It could be solved in the same way as in the linear case, but if $\mathbb{F}$ is very high-dimensional this procedure is no more feasible due to computational reasons. We therefore restrict our attention to an important special case of nonlinear maps, namely those where the scalar product between two nonlinearly mapped input vectors can be expressed analytically by a *kernel* function k of the input vectors

$$\phi(\mathbf{x}_1)^\top \phi(\mathbf{x}_2) = k(\mathbf{x}_1, \mathbf{x}_2). \tag{29}$$

As a consequence, the evaluation of a possibly infinite number of terms in the scalar product in $\mathbb{F}$ reduces to the computation of the kernel k directly on the input.

Equation (29) is only valid for *positive definite* kernels, i.e., functions k with the property that the *Gram matrix* $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ is positive definite for all choices of the $\mathbf{x}_1, \dots, \mathbf{x}_N$. It can be shown that a number of kernels satisfies this condition including polynomial, Gaussian and sigmoid kernels (Schölkopf & Smola, 2002). The subspace of $\mathbb{F}$ spanned by the $\phi(\mathbf{x}_i)$ can be viewed as a space of functions $f(\mathbf{x})$ defined by

$$f(\mathbf{x}) = \sum\nolimits_{j=1}^{M} \gamma_j k(\mathbf{x}, \mathbf{z}_j). \tag{30}$$

This space has the structure of a *reproducing kernel Hilbert space (RKHS)*. By carrying out linear methods in $\mathbb{F}$, one can obtain elegant solutions for various nonlinear estimation problems (see Schölkopf & Smola, 2002), examples being Support Vector Machines.

If we can express a solution such as Eq. (23) only in terms of scalar products, we can use this property to manipulate the otherwise inaccessible elements of $\mathbb{F}$. For that purpose, we resort to our previous finding that the

[2]Note that with a slight abuse of notation, we use the transpose also to denote the scalar product in infinite-dimensional spaces.

solution of the linear regression problem is an element of the row space of the data matrix X. Consequently, we know that the regression result for the model in Eq. (28) is a linear combination of the mapped input samples $\phi(\mathbf{x}_i)$. Thus, the solution vector $\gamma \in \mathbb{F}$ can be written as

$$\gamma = \Phi^\top \mathbf{a} \tag{31}$$

with some coefficient vector $\mathbf{a} \in \mathbb{R}^N$. The representation of γ by using its coefficient vector with respect to the input samples is called its *dual* representation (Cristianini & Shawe-Taylor, 2000). Substituting this into Eq. (28), we obtain the model

$$\mathbf{y} = \Phi\Phi^\top \mathbf{a} = K\mathbf{a} \tag{32}$$

with the symmetric $N \times N$ *Gram* matrix $K = \Phi\Phi^\top = (k(\mathbf{x}_i, \mathbf{x}_j))_{ij}$. The dual problem now has the same functional form as in the linear case with the solution from Eq. (23)[3]

$$\mathbf{a} = (K^\top K)^{-1} K^\top \mathbf{y} = K^{-1}\mathbf{y} \tag{33}$$

since $K = K^\top$. Note that the solution depends on the input data only via the Gram matrix. The Gram matrix contains only scalar products of vectors from $\mathbb{F}$ which can be computed according to Eq. (29) if an appropriate map ϕ is used. Moreover, if the solution is used for predicting the system output y for a new input $\mathbf{x}$, again only scalar products are used:

$$y = \gamma^\top \phi(\mathbf{x}) = \mathbf{a}^\top \Phi\, \phi(\mathbf{x}) = \mathbf{a}^\top \mathbf{z}(\mathbf{x}) = \mathbf{y}^\top K^{-1} \mathbf{z}(\mathbf{x}), \tag{34}$$

where $\mathbf{z}(\mathbf{x}) = (k(\mathbf{x}_1, \mathbf{x}), k(\mathbf{x}_2, \mathbf{x}), \ldots, k(\mathbf{x}_N, \mathbf{x}))^\top \in \mathbb{R}^N$. Regression and prediction can be done on the simplified scalar products alone, without the need for explicitely mapping the inputs into $\mathbb{F}$.

Although we have directly shown the existence of a dual representation of the solution, the same conclusion can be drawn from a more general property of RKHSs referred to as the *Representer Theorem* (Kimeldorf & Wahba, 1971). It states the following: suppose c is an arbitrary cost function, Ω is a nondecreasing function on $\mathbb{R}_{>0}$ and $\|.\|_\mathbb{F}$ is the norm of the RKHS. If we minimize an objective function[4]

$$c((\mathbf{x}_1, y_1, f(\mathbf{x}_1)), \ldots, (\mathbf{x}_N, y_N, f(\mathbf{x}_N))) + \Omega(\|f\|_\mathbb{F}), \tag{35}$$

over all functions of the form (30), then an optimal solution[5] can be expressed as

$$f(\mathbf{x}) = \sum\nolimits_{j=1}^{N} a_j k(\mathbf{x}, \mathbf{x}_j), \quad a_j \in \mathbb{R}. \tag{36}$$

In other words, although we did consider functions which were expansions in terms of arbitrary points $\mathbf{x}_j$ (see (30)), 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 variables γ_j is transformed into one over N variables a_j, where N is the number of training points.

4 Estimating Wiener series by linear regression in RKHS

4.1 Volterra series as linear operator in RKHS

We now have the prerequisites to develop a new approach to estimating the Wiener series expansion. As our first step, we have to convert the Volterra series into a form suitable for regression in RKHS. Our starting point is the discretized version of the Volterra operators from Eq. (5)

$$H_n[\mathbf{x}] = \sum_{i_1=1}^{m} \cdots \sum_{i_n=1}^{m} h^{(n)}_{i_1 \ldots i_n} x_{i_1} \ldots x_{i_n} \tag{37}$$

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

[4] In our case, we use the mean square error as a cost function, and the regularizer Ω is set to zero.

[5] for conditions on uniqueness of the solution, see (Schölkopf & Smola, 2002)

which is also the base of the classical cross-correlation procedure. The nth-order Volterra operator is a weighted sum of all nth-order monomials of the input vector $\mathbf{x}$. For $n = 0, 1, 2, \ldots$ e define the map ϕ_n as

$$\begin{aligned}
\phi_0(\mathbf{x}) &= 1 && (38)\\
\phi_1(\mathbf{x}) &= (x_1, x_2, \ldots, x_m) && (39)\\
\phi_2(\mathbf{x}) &= (x_1^2, x_1x_2, x_2x_1, x_2^2, x_1x_3, \ldots, x_m^2) && (40)\\
\phi_3(\mathbf{x}) &= (x_1^3, x_1x_2x_3, x_1x_3x_2, x_2x_1x_3, x_2x_3x_1, \ldots, x_m^3) && (41)\\
&\vdots
\end{aligned}$$

such that ϕ_n maps the input $\mathbf{x} \in \mathbb{R}^m$ into a vector $\phi_n(\mathbf{x}) \in \mathbb{F}_n = \mathbb{R}^{m^n}$ containing all m^n ordered monomials of degree n. Using ϕ_n, we can write the nth-order Volterra operator in Eq. (5) as a scalar product in $\mathbb{F}_n$

$$H_n[\mathbf{x}] = \eta_n^\top \phi_n(\mathbf{x}) \tag{42}$$

with the coefficients stacked into the vector $\eta_n = (h^{(n)}_{1,1,\ldots1}, h^{(n)}_{1,2,\ldots1}, h^{(n)}_{1,3,\ldots1}, \ldots)^\top \in \mathbb{F}_n$. Fortunately, the functions ϕ_n constitute a RKHS. It can be easily shown that

$$\begin{aligned}
k_n(\mathbf{x}, \mathbf{z}) &= \phi_n(\mathbf{x})^\top \phi_n(\mathbf{z})\\
&= \sum_{i_1=1}^{m} \ldots \sum_{i_n=1}^{m} x_{i_1} \cdot \ldots \cdot x_{i_n} \cdot z_{i_1} \cdot \ldots \cdot z_{i_n} && (43)\\
&= \sum_{i_1=1}^{m} x_{i_1} \cdot z_{i_1} \ldots \sum_{i_m=1}^{m} x_{i_1} \cdot z_{i_m} = \left(\sum_{i=1}^{m} x_j \cdot z_j \right)^n = (\mathbf{x}^\top \mathbf{z})^n. && (44)
\end{aligned}$$

The same idea can be applied to the entire pth-order Volterra series. By stacking the maps ϕ_n into a single map $\phi^{(p)}(\mathbf{x}) = (\phi_0(\mathbf{x}), \phi_1(\mathbf{x}), \ldots, \phi_p(\mathbf{x}))^\top$, one obtains a mapping from $\mathbb{R}^m$ into $\mathbb{F}^{(p)} = \mathbb{R} \times \mathbb{R}^m \times \mathbb{R}^{m^2} \times \ldots \mathbb{R}^{m^p} = \mathbb{R}^M$ with dimensionality $M = \frac{1-m^{p+1}}{1-m}$ [6]. The entire pth-order Volterra series can be written as a scalar product in $\mathbb{F}^{(p)}$

$$\sum_{n=0}^{p} H_n[\mathbf{x}] = (\eta^{(p)})^\top \phi^{(p)}(\mathbf{x}) \tag{45}$$

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

$$k^{(p)}(\mathbf{x}_1, \mathbf{x}_2) = \phi^{(p)}(\mathbf{x}_1)^\top \phi^{(p)}(\mathbf{x}_2) = \sum_{n=0}^{p} (\mathbf{x}_1^\top \mathbf{x}_2)^n. \tag{46}$$

Thus, we have shown that the discretized pth-order Volterra series can be expressed as a linear operator in the RKHS spanned by all ordered monomials up to order p.

4.2 Implicit Wiener series estimation

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[7]. From Eq. (34), we obtain the following expression for the implicit Wiener series

$$\begin{aligned}
G_0[\mathbf{x}] &= \frac{1}{N} \mathbf{y}^\top \mathbf{1} && (47)\\
\sum_{n=0}^{p} G_n[\mathbf{x}] &= \sum_{n=0}^{p} H_n[\mathbf{x}] = \mathbf{y}^\top K_p^{-1}\, \mathbf{z}^{(p)}(\mathbf{x}) && (48)
\end{aligned}$$

[6] This result is obtained by applying the geometric sum formula.

[7] assuming symmetrized Volterra kernels which can always be obtained from any Volterra expansion.

where the Gram matrix K_p and the coefficient vector $\mathbf{z}^{(p)}(\mathbf{x})$ are computed using the kernel from Eq. (46) and $\mathbf{1} = (1, 1, \dots)^\top \in \mathbb{R}^N$. Note that the Wiener series and its Volterra functionals are represented only implicitely since we are using the RKHS representation as a sum of scalar products with the training points. Thus, we can avoid the "curse of dimensionality", i.e., there is no need to compute the possibly large number of coefficients explicitely.

The explicit Volterra and Wiener expansions can be recovered at least in principle from Eq. (48) 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

$$H_n[\mathbf{x}] = \mathbf{y}^\top K_p^{-1} \mathbf{z}_n(\mathbf{x}) \tag{49}$$

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

$$\eta_n = \Phi_n^\top K_p^{-1} \mathbf{y} \tag{50}$$

using the design matrix $\Phi_n = (\phi_n(\mathbf{x}_1)^\top, \phi_n(\mathbf{x}_1)^\top, \dots, \phi_n(\mathbf{x}_1)^\top)^\top$.

The individual Wiener functionals can only be computed by applying the regression procedure twice. If we are interested in the nth-degree Wiener functional, 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 functional for $n > 0$ is then obtained from the difference of the two results as

$$G_n[\mathbf{x}] = \sum_{i=0}^{n} G_i[\mathbf{x}] - \sum_{i=0}^{n-1} G_i[\mathbf{x}] = \mathbf{y}^\top \left[K_n^{-1}\, \mathbf{z}^{(n)}(\mathbf{x}) - K_{n-1}^{-1}\, \mathbf{z}^{(n-1)}(\mathbf{x}) \right]. \tag{51}$$

The corresponding ith-order Volterra operators of the nth-degree Wiener functional are computed analogously to Eqns. (49) and (50) as

$$H_i^{(n)}[\mathbf{x}] = \mathbf{y}^\top (K_n^{-1} - K_{n-1}^{-1}) \mathbf{z}_i(\mathbf{x}) \tag{52}$$

and

$$\eta_i^{(n)} = \Phi_i^\top (K_n^{-1} - K_{n-1}^{-1})\, \mathbf{y}. \tag{53}$$

4.3 Orthogonality

The resulting Wiener functionals must fulfill the orthogonality condition which in its strictest form states that a pth-degree Wiener functional must be orthogonal to all monomials in the input of lower order (Schetzen, 1989). Formally, we will prove the following

Theorem 1 *The functionals obtained from Eq. (51) fulfill the orthogonality condition*

$$E\left[m(\mathbf{x}) G_p[\mathbf{x}]\right] = 0 \tag{54}$$

where E denotes the expectation over the input distribution and $m(\mathbf{x})$ an ith-order monomial with $i < p$.

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 Eq. (25). In the case of the Wiener and Volterra expansions, the basis functions $\varphi_j(\mathbf{x})$ are monomials of the components of $\mathbf{x}$.

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

$$\frac{\partial L}{\partial \gamma_k} = \frac{\partial}{\partial \gamma_k} E\|e(\mathbf{x})\|^2 = -2E\left[\varphi_k(\mathbf{x}) e(\mathbf{x})\right] = 0. \tag{55}$$

This means that, for an expansion of the type of Eq. (25) minimizing the squared error, the error is orthogonal to all base 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 given by the sum of the higher-order Wiener functionals $e(\mathbf{x}) = \sum_{n=p}^{\infty} G_n[\mathbf{x}]$, so $G_p[\mathbf{x}]$ is part of the error. As a consequence of the linearity of the expectation, Eq. (55) implies

$$\sum\nolimits_{n=p}^{\infty} E\left[\varphi_k(\mathbf{x}) G_n[\mathbf{x}]\right] = 0 \quad \text{and} \quad \sum\nolimits_{n=p+1}^{\infty} E\left[\varphi_k(\mathbf{x}) G_n[\mathbf{x}]\right] = 0 \tag{56}$$

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, namely to all monomials with order smaller than p. □

Note that for both the regression and the orthogonality of the resulting functionals the assumption of white Gaussian noise was not required. In practice, this means that we can compute the implicit Wiener series for any type of input, not just for Gaussian noise. The resulting Volterra functionals (Eqns. (49) and (50)) should be - at least in principle - the same regardless of the input signal. This, however, is not the case for the computation of the Wiener functionals and their Volterra expansion in Eqns. (51) - (53). As we saw above, orthogonality of functionals can be only defined with respect to an input distribution. If we use Eqns. (51) - (53) for nongaussian input the resulting functionals will still be orthogonal, but with respect to the nongaussian input distribution. The resulting decomposition of the Volterra series into orthogonal functionals will be different from the Gaussian case. As a consequence, the functionals computed according to Eqns. (51) - (53) will be different from the Wiener functionals, even if the Volterra expansion is exactly the same in both cases. If one still wants to compute the Wiener expansion, but can only use nongaussian input, one needs to resort to the classical procedure of Eq. (17) that has to be applied to the explicit Volterra expansion from Eqns. (50) and (5).

5 Examples and experiments

5.1 Analytic toy example

To check whether the proposed solutions are consistent, we apply our formalism to a toy example for which a closed form analytic solution exists. We assume the following scenario: The system to be identified receives scalar input x (i.e., $m = 1$) and we only have two pairs of measurements (i.e., $N = 2$) (x_1, y_1) and (x_2, y_2). As our model, we choose a Wiener series up to functionals of degree $p = 1$

$$\hat{y} = G_0[x] + G_1[x] \tag{57}$$

with the model output $\hat{y}$. The associated scalar product is given by (cf. Eq. (46)

$$k^{(1)}(x_1, x_2) = 1 + x_1 x_2. \tag{58}$$

In order to compute the implicit expansion of Eq. (48), we need the Gram matrix

$$K_1 = \begin{pmatrix} k^{(1)}(x_1, x_1) & k^{(1)}(x_1, x_2) \\ k^{(1)}(x_2, x_1) & k^{(1)}(x_2, x_2) \end{pmatrix} = \begin{pmatrix} 1 + x_1^2 & 1 + x_1 x_2 \\ 1 + x_2 x_1 & 1 + x_2^2 \end{pmatrix} \tag{59}$$

which is identical to the matrix $X^\top X$ in linear regression (cf. Eq. (19)). The inverse is given by

$$K_1^{-1} = (X^\top X)^{-1} = \frac{1}{(x_1 - x_2)^2} \begin{pmatrix} 1 + x_2^2 & -1 - x_1 x_2 \\ -1 - x_1 x_2 & 1 + x_1^2 \end{pmatrix}. \tag{60}$$

The dual coefficient vector in Eq. (48) is

$$\mathbf{z}^{(1)}(x) = \begin{pmatrix} k^{(1)}(x_1, x) \\ k^{(1)}(x_2, x) \end{pmatrix} = \begin{pmatrix} 1 + x_1 x \\ 1 + x_2 x \end{pmatrix} = X \begin{pmatrix} x \\ 1 \end{pmatrix}, \tag{61}$$

such that the entire solution becomes

$$\hat{y} = \mathbf{y}^\top (X^\top X)^{-1} X \begin{pmatrix} x \\ 1 \end{pmatrix} = \mathbf{y}^\top X (X^\top X)^{-1} \begin{pmatrix} x \\ 1 \end{pmatrix} \tag{62}$$

which is identical to the result found for linear regression (cf. Eq. (23)).

If Eq. (50) is correct, we should obtain the explicit Volterra operator coefficients

$$\eta_0 = a \quad \text{and} \quad \eta_1 = b \tag{63}$$

with the coefficients a and b identical to those given by classical linear regression (Press, Teukolsky, Vetterling, & Flannery, 1992)

$$a = \frac{(\sum x_i^2)(\sum y_i) - (\sum x_i)(\sum x_i y_i)}{N \sum x_i^2 - (\sum x_i)^2} = \frac{x_2^2 y_1 + x_1^2 y_2 - x_1 x_2 (y_2 - y_1)}{(x_1 - x_2)^2} \tag{64}$$

$$b = \frac{N(\sum x_i y_i) - (\sum x_i)(\sum y_i)}{N \sum x_i^2 - (\sum x_i)^2} = \frac{y_2 - y_1}{x_2 - x_1}. \tag{65}$$

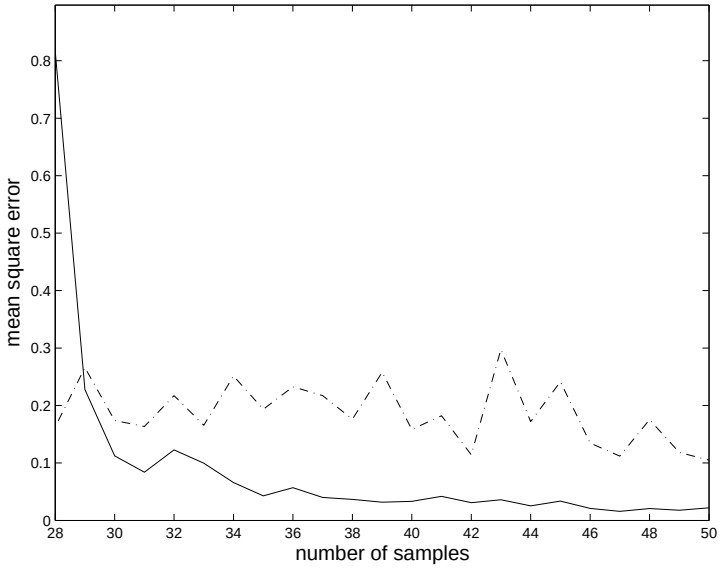

Figure 1: Mean squared error between true and estimated coefficients of the second-order Wiener kernel of a Reichardt-type correlation detector for varying number of training samples. The solid line depicts the error of the regression technique, the dash-dot line that of the cross-correlation technique.

According to Eq. (50), we obtain the coefficients using $\phi_0^\top = \mathbf{1}^\top$ and $\phi_1^\top = (x_1, x_2)$

$$\eta_0 = \mathbf{1}^\top K_1^{-1}\mathbf{y} \quad \text{and} \quad \eta_1 = (x_1, x_2)K_1^{-1}\mathbf{y} \tag{66}$$

which can be easily shown to be identical with a and b by substituting Eq. (60).

The corresponding Wiener functionals from Eqns. (47) and (51) are

$$G_0[x] = \frac{1}{2}(y_1 + y_2) \tag{67}$$

$$G_1[x] = \mathbf{y}K_1^{-1}\begin{pmatrix}1 + x_1 x \\ 1 + x_2 x\end{pmatrix} - \frac{1}{2}(y_1 + y_2) \tag{68}$$

which yields the identical Volterra series. Note that the zero-order Volterra functional of the first-degree Wiener functional given by Eq. (52)

$$H_0^{(1)}[x] = \mathbf{y}K_1^{-1}\mathbf{1} - \frac{1}{2}(y_1 + y_2) = a - \frac{1}{2}(y_1 + y_2) \tag{69}$$

is not zero as required for strict orthogonality of the operators. However, we have only two input samples , and orthogonality is merely required for the entire input distribution. For zero-mean white Gaussian input and a linear system with constant term a, we obtain indeed

$$E\left[\frac{1}{2}(y_1 + y_2)\right] = a \tag{70}$$

such that $H_0^{(1)}[x]$ becomes zero.

5.2 Numerical experiment on convergence

In this example, the system to be identified is a discretized Reichardt-type correlation detector (Hassenstein & Reichardt, 1956) of the form

$$y[n] = \left(\sum\nolimits_{k=0}^{4} h[k]x[n-k]\right) \times \left(\sum\nolimits_{k=0}^{4} l[k]x[n+1-k]\right) \tag{71}$$

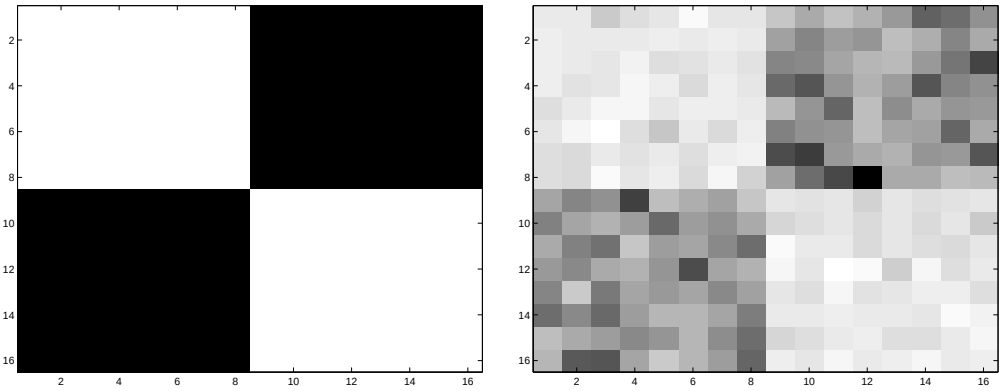

Figure 2: *Left:* 16×16 nonlinear receptive field of the test system; *Right:* Reconstructed receptive field from the fifth-order Volterra kernel by computing a preimage (after 2500 samples).

with some arbitrary, but fixed high pass h[k] and low pass l[k] of order 5. The data are generated by sliding two windows of width $x[n-4] \dots x[n]$ and $x[n-3] \dots x[n+1]$ over a time series of white, zero-mean Gaussian noise and simultaneously measuring the system output $y[n]$. Finally, we added white, zero-mean Gaussian measurement noise to the signal with a variance of 10% of the signal variance. We applied both estimation methods, cross-correlation and regression, to estimate the 21 free parameters of the second-degree Wiener model $\hat{y} = \sum_{i=0}^{2} G_i[\mathbf{x}]$. The modeling error ϵ was measured for the second order kernel using

$$\epsilon = \sum_{i=1}^{5} (\eta_i - h_i)^2 \tag{72}$$

and averaging ϵ over the 20 trials. We varied the number of training samples to see how the modeling error decreases with the number of samples.

As the result shows (Fig. 1), the modeling error of the regression technique decreases at a significantly faster rate than the cross-correlation method due to the unfavorable properties of the cross-correlation estimator. In fact, a comparable modeling error is only reached at sample sizes that are more than 10 times as large (not contained in the figure).

5.3 Reconstruction of a fifth-order nonlinear receptive field.

This experiment demonstrates the applicability of the proposed method to high-dimensional input. Our example is the fifth-order system

$$y = \left(\sum_{k,l=1}^{16} h_{kl} x_{kl} \right)^5 \tag{73}$$

that acts on 16×16 image patches by convolving them with a receptive field h_{kl} of the same size shown in Fig. $2a$ before the nonlinearity is applied. We generated 2500 image patches containing uniformly distributed white noise and computed the corresponding system output to which, as above, we added 10% Gaussian measurement noise.

The resulting data was used to estimate the implicit Wiener expansion using the regression procedure. In the classical cross-correlation procedure, 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 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 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 Eq. (49)

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 Fig. 2*b* demonstrates that the original receptive field is already recognizable after using 2500 samples. The example shows that the preimage technique elucidates to which input structures the Volterra-operator is tuned, similar to the classical analysis techniques in linear systems.

6 Conclusion

The benefits of the proposed estimation of the Wiener and Volterra expansions via kernel regression can be summarized as follows:

1. The implicit representation 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. Convergence is considerably faster than in the classical procedure because the estimation is done directly on the data. The regression method omits the intermediate step of estimating cross-correlations which converges very slowly.

3. Preimage techniques reveal input structures to which Wiener or Volterra operators are tuned. The preimage corresponds to a nonlinear receptive field where the input is convolved with a linear filter whose output is fed into a nonlinearity. The present method works only for Volterra kernels of odd order. More general techniques exist, including the case of other kernels and the computation of approximations in terms of several preimages ("reduced sets" Schölkopf & Smola, 2002). The latter corresponds to an invariant subspace of the Volterra operator (cf. Hyvärinen & Hoyer, 2000).

4. The method works also for non-Gaussian input. In particular, uniform noise turned out to lead to better results than Gaussian noise since its values are bounded. The Gaussian distribution sometimes produces very large values which are extremely amplified by the higher order monomial terms.

From the point of view of learning theory, the proposed estimation method has the drawback that the regularization term in the objective function (35) is currently set to zero in order to preserve the orthogonality property of the resulting Wiener functionals. This may possibly lead to a degraded generalization performance and an increased sensitivity to noise. The second part of this report will therefore discuss the regularized estimation of the Wiener series. However, one could argue that already in the present form, an implicit regularization is in effect: in the form stated, the representer theorem does not rule out the existence of solutions outside the span of the $k(., \mathbf{x}_j)$ with the same (albeit not a lower) value of the objective function. In that case, our algorithm chooses the solution which lies in the span (note that this need not be the case for the classical cross-correlation method). One can prove that if a regularizer with strictly monotone Ω (cf. Eq. (35)) is used, then *all* optimal solutions lie in the span (Schölkopf & Smola, 2002). Our algorithm thus searches a restricted space of possible solutions which coincides with the one searched by its regularized counterpart.

Acknowledgments

The ideas presented in this report have greatly profited from discussions with G. Bakır, M. Kuss, and C. Rasmussen.

References

Cristianini, N., & Shawe-Taylor, J. (2000). *Support vector machines*. Cambridge: Cambridge University Press.

Geiger, G., & Poggio, T. (1975). The orientation of flies towards visual patterns: On the search for the underlying functional interactions. *Biol. Cybern.*, **19**, 39 – 54.

Hassenstein, B., & Reichardt, W. (1956). Systemtheoretische Analyse der Zeit-, Reihenfolgen- und Vorzeichenauswertung bei der Bewegungsperzeption des Rüsselkäfers *Chlorophanus*. *Zeitschrift f. Naturforschung*, **11b**, 513 – 524.

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.

Kimeldorf, G. S., & Wahba, G. (1971). Some results on Tchebycheffian spline functions. *J. Math. Analysis and Applications*, **33**, 82 – 95.

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

Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (1992). *Numerical recipes in C*. Cambride, U.K.: Cambridge University Press.

Rieke, F., Warland, D., de Ruyter van Steveninck, R., & Bialek, W. (1996). *Spikes: Exploring the neural code*. Cambridge, MA: MIT Press.

Schetzen, M. (1989). *The Volterra and Wiener theories of nonlinear systems*. Malabar: Krieger.

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

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

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