Monograph:
Wan, Y., Dodd, T.J. and Harrison, R.F. (2003) A kernel method for non-linear systems identification – infinite degree volterra series estimation. Research Report. ACSE Research Report no. 842 . Automatic Control and Systems Engineering, University of Sheffield

A KERNEL METHOD FOR NON-LINEAR SYSTEMS IDENTIFICATION – INFINITE DEGREE VOLTERRA SERIES ESTIMATION

Yufeng Wan, Tony J. Dodd, and Robert F. Harrison
Department of Automatic Control & Systems Engineering
The University of Sheffield, Mappin Street, Sheffield S1 3JD, UK
{y.wan, t.j.dodd, r.f.harrison}@sheffield.ac.uk
Research Report No. 842

Abstract

Volterra series expansions are widely used in analyzing and solving the problems of non-linear dynamical systems. However, the problem that the number of terms to be determined increases exponentially with the order of the expansion restricts its practical application. In practice, Volterra series expansions are truncated severely so that they may not give accurate representations of the original system. To address this problem, kernel methods are shown to be deserving of exploration.

In this report, we make use of an existing result from the theory of approximation in reproducing kernel Hilbert space (RKHS) that has not yet been exploited in the systems identification field. An exponential kernel method, based on an RKHS called a generalized Fock space, is introduced, to model non-linear dynamical systems and to specify the corresponding Volterra series expansion. In this way a non-linear dynamical system can be modelled using a finite memory length, infinite degree *Volterra series expansion, thus reducing the source of approximation error solely to truncation in time. We can also, in principle, recover any coefficient in the Volterra series.*

1 Introduction

Volterra series models are widely used in analyzing and solving the problems of non-linear dynamical systems. The term Volterra series derives from the work of Vito Volterra, an Italian mathematician, at the end of the nineteenth century, the idea of which can be regarded as an extension of the linear convolution model. Volterra series have been shown to provide a good representation for a wide range of non-linear systems [2]. Without the involvement of the previous, predicted output signals, such a model can give a relatively accurate prediction of the output over the domain of interest for systems of "fading memory" type [2], assuming we can measure the input signals exactly. Because of this and some other practical advantages such as their being linear-in-the-parameters [9], much has been done in the development and application of Volterra series models. At the same time, a noticeable shortcoming restricts the practical use of even truncated Volterra series because they involve exponentially many parameters. Thus, in practice, only "small" models have been found to be useful with a concomitant loss of precision. The use of kernel methods for finite degree, finite length Volterra series estimation has already been addressed [8, 5, 3, 4].

This report is primarily concerned with discrete-time, finite length, *infinite* degree Volterra series expansions. A method based on exponential kernels is used to model non-linear dynamical systems and to determine the parameters of the corresponding Volterra series. The next section gives the theoretical basis of the subsequent discussion. The particular Generalized Fock (GF) space, $\mathcal{F}$, and the corresponding kernel, k, are constructed. Proof that k is a reproducing kernel and $\mathcal{F}$ is its reproducing kernel Hilbert space (RKHS) is given. In section 3 the method of computing infinite degree Volterra series expansions using the kernel constructed in section 2, is discussed. The problem of how to recover the original Volterra series from the kernel is discussed in section 4, and, in section 5, two synthetic system identification examples are given. Finally, we discuss the limitations and potential of the method.

2 Generalized Fock Spaces

Consider a discrete-time, finite length[1], infinite degree Volterra series expansion

$$y\left(\underline{u}\right) = h_0 + \sum_{n=1}^{\infty} \left\{ \sum_{m_1=1}^{M} \cdots \sum_{m_n=1}^{M} h_n\left(m_1, \ldots, m_n\right) \prod_{j=1}^{n} u_{m_j} \right\}, \tag{1}$$

[1] The use of a finite memory length method implies that the dynamics under consideration must be of the class of "fading memory" systems [2].

where

$$\underline{u} = [u_1, \ldots, u_M]^T$$

and M is the memory length. A sufficient but not necessary condition that guarantees the convergence of Eq.(1) is that [9]:

$$\sum_{m_1=1}^{M} \sum_{m_2=1}^{M} \cdots \sum_{m_n=1}^{M} | h_n(m_1, m_2, \ldots, m_n) | < \infty \qquad (2)$$

A GF space, $\mathcal{F}$, can be constructed [6], consisting of the elements, f,

$$\mathcal{F}: \; f = h_0^f + \sum_{n=1}^{\infty} \frac{1}{n!} f_n, \qquad (3)$$

in which the f_n, given by

$$f_n(\underline{u}) = \sum_{m_1=1}^{M} \cdots \sum_{m_n=1}^{M} h_n^f(m_1, \ldots m_n) \prod_{j=1}^{n} u_{m_j} \qquad (4)$$

completely characterize f.

We then define the inner product [7] in the specified GF space, $\mathcal{F}$, as:

$$< f, g >_{\mathcal{F}} = \sum_{n=0}^{\infty} \frac{p^n}{n!} (f_n, g_n)_n, \qquad (5)$$

where

$$\begin{aligned} (f_n, g_n)_n &= h_0^f h_0^g \qquad (6) \\ &+ \sum_{m_1=1}^{M} \sum_{m_n=1}^{M} h_n^f(m_1, \ldots, m_n) h_n^g(m_1, \ldots, m_n). \end{aligned}$$

In Eq.(5), $P = \{p^0, p^1, \ldots, p^n, \ldots, p^\infty\}$ denotes a set of weighting constants, in which p is a constant chosen according to prior knowledge and satisfying the convergence condition given by Eq.(2). The relationship between p and h_n will be established later.

We then construct an exponential kernel

$$k(\underline{u}, \underline{v}) = \exp\left(\frac{< \underline{u}, \underline{v} >_{l^2}}{p}\right) \qquad (7)$$

in which $< \underline{u}, \underline{v} >_{l_2}$ denotes a dot product in l_2,

$$< \underline{u}, \underline{v} >_{l_2} = \underline{u}^T \underline{v} = \sum_{i=1}^{M} u_i v_i. \qquad (8)$$

According to the theory of Taylor Series, we know that $e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$. So the exponential kernel k can be rewritten as

$$k(\underline{u}, \underline{v}) = \sum_{n=0}^{\infty} \frac{1}{n!} \frac{(< \underline{u}, \underline{v} >_{l^2})^n}{p^n} = \sum_{n=0}^{\infty} \frac{1}{p^n n!} (< \underline{u}, \underline{v} >_{l^2})^n \qquad (9)$$

We can now prove that k is a reproducing kernel in $\mathcal{F}$ through the following two steps:

1. For a fixed input sequence, $\underline{u}$, we substitute Eq.(8) into Eq.(9) such that

$$\begin{aligned} & k(\underline{u}, \underline{v}) \\ = & \sum_{n=0}^{\infty} \frac{1}{n!} \frac{\left(\sum_{i=1}^{M} u_i v_i\right)^n}{p^n} \\ = & 1 + \sum_{n=1}^{\infty} \frac{1}{n!} \left[\frac{1}{p^n} \sum_{m_1=1}^{M} \cdots \sum_{m_n=1}^{M} \left(\prod_{j=1}^{n} u_{m_j} \prod_{j=1}^{n} v_{m_j}\right)\right] \\ = & 1 + \sum_{n=1}^{\infty} \frac{1}{n!} \left\{\sum_{m_1=1}^{M} \cdots \sum_{m_n=1}^{M} \right. \\ & \left. \left[\left(\frac{1}{p^n} \prod_{j=1}^{n} u_{m_j}\right) \prod_{j=1}^{n} v_{m_j}\right]\right\} \qquad (10) \end{aligned}$$

where h_n^k is defined as follows:

$$h_n^k = \begin{cases} 1 & , n = 0; \\ \frac{1}{p^n} \prod_{j=1}^{n} u_{m_j} & , n = 1, 2, \ldots, \infty. \end{cases} \qquad (11)$$

Substituting Eq.(11) into Eq.(10), we get

$$k(\underline{u}, \underline{v}) = \sum_{n=0}^{\infty} \frac{1}{n!} k_n(\underline{v}) \qquad (12)$$

in which

$$k_n(\underline{v}) = h_0^k + \sum_{m_1=1}^{M} \cdots \sum_{m_n=1}^{M} h_n^k(m_1 \ldots m_n) \prod_{j=1}^{n} v_{m_j}. \qquad (13)$$

Eqs.(12), (13) are of the general forms, (3), (4), which means $k(\underline{u}, .) \in \mathcal{F}$.

2. For every $f \in \mathcal{F}$, we have

$$< k(\underline{u}, .), f(.) >_{\mathcal{F}} = \sum_{n=0}^{\infty} \frac{p^n}{n!} (k_n, f_n)_n. \qquad (14)$$

If we replace $(k_n, f_n)_n$ in Eq.(14) with Eq.(7), $\langle k(\underline{u}, .), f(.)\rangle_{\mathcal{F}}$ can be re-written as follows:

$$\begin{aligned} & < k(\underline{u}, .), f(.) >_{\mathcal{F}} \\ = & \sum_{n=0}^{\infty} \frac{p^n}{n!} \left(h_0^k h_0^f + \sum_{m_1=1}^{M} \cdots \sum_{m_n=1}^{M} h_n^k h_n^f\right) \\ = & h_0^k h_0^f + \sum_{n=1}^{\infty} \frac{p^n}{n!} \left(\sum_{m_1=1}^{M} \cdots \sum_{m_n=1}^{M} \frac{1}{p^n} \prod_{j=1}^{n} u_{m_j} h_n^f\right) \\ = & h_0^f + \sum_{n=1}^{\infty} \frac{1}{n!} \left(\sum_{m_1=1}^{M} \cdots \sum_{m_n=1}^{M} h_n^f \prod_{j=1}^{n} u_{m_j}\right) \qquad (15) \end{aligned}$$

in which we have used $h_0^k = 1$. Recalling the definition of f_n, (4), we can rewrite Eq.(15) as:

$$
\begin{aligned}
< k(\underline{u}, .), f(.) >_{\mathcal{F}} &= \sum_{n=0}^{\infty} \frac{1}{n!} f_n(\underline{u}) \\
&= f(\underline{u}) \qquad (16)
\end{aligned}
$$

Eq.(16) demonstrates the reproducing property of the kernel, $k(\underline{u}, \underline{v})$.

The function $k : \mathbb{R}^M \times \mathbb{R}^M \to \mathbb{R}_+$ having the above two properties is called the "reproducing kernel" [1] of the space $\mathcal{F}$. Equipped with such a k, $\mathcal{F}$ is known as a RKHS. The characteristic of such a kernel, k, is encompassed in the Moore-Aronszajn theorem:

Theorem 2.1 *[10] To every RKHS there corresponds a unique positive-definite function (the reproducing kernel) and conversely given a positive-definite function, k, on $\mathbb{R}^M$ we can construct a unique RKHS of real-valued functions on $\mathbb{R}^M$ with k as its reproducing kernel.*

Property 2 above is very useful and will be utilized in the next section to get an important result.

Given that the kernel, $k(\underline{u}, .)$, belongs to the GF space, $\mathcal{F}$, functions in $\mathcal{F}$ corresponding to the infinite degree Volterra series expansions, Eq.(3), can now be expressed in terms of the kernels,

$$f(\underline{u}) = \sum_{i=1}^{N} a_i k(\underline{u}_i, \underline{u}) \qquad (17)$$

where N is the number of samples and $a_i \in \mathbb{R}$.

3 The Best Approximation of The Original Volterra Series Expansion

It is well known that, given (16), the best approximation, $\hat{y}$, of the original Volterra series expansion, y, is given by the projection of y in the closed subspace of $\mathcal{F}$ spanned by $k(\underline{u}_1, .), \ldots, k(\underline{u}_N, .)$. Therefore, $\hat{y}$ is given by

$$\hat{y}(\underline{u}) = \sum_{i=1}^{N} a_i k(\underline{u}_i, \underline{u}) \qquad (18)$$

where

$$\underline{u}_i = [u_{i1}, \ldots, u_{iM}]^T.$$

In Eq.(18), $\underline{a} : \underline{a} = [a_1, \ldots, a_N]^T$ is the coefficient vector that we need to specify. It can be obtained by the following equations [7]:

$$\underline{a} = K^{-1}\underline{y}, \qquad (19)$$

where

$$\underline{y} = [y_1, \ldots, y_N]^T$$

and K is the kernel Gram matrix,

$$K_{ij} = k(\underline{u}_i, \underline{u}_j) = \exp\left(\frac{< \underline{u}_i, \underline{u}_j >_{l^2}}{p}\right). \qquad (20)$$

Note that it has been proven [11] that, under some restrictions, the kernel Gram matrix, K, is nonsingular, providing a unique solution to (19).

Theorem 3.1 *[11] If $\underline{u}_i, i = 1, 2, \ldots, N$, are distinct elements of $\mathbb{R}^M$, then the $N \times N$ matrix K,*

$$K_{ij} = \exp\left(\frac{< \underline{u}_i, \underline{u}_j >_{l^2}}{p}\right), \qquad (21)$$

is nonsingular.

The above statements give the idea and the way to compute the infinite degree Volterra model with the exponential kernel, k, given by Eq.(7). In the following section, we will discuss how to recover the terms of the original infinite degree Volterra series model, which is valuable for the analysis and interpretation of systems in practice.

3.1 Numerical Considerations

We note that Theorem 3.1 gives a theoretical guarantee for the existence of a unique solution, $\tilde{\underline{a}}$, but, in practice, even in the noise free case, numerical sensitivity may present a problem and we often need to solve Eq.(19) via the pseudo-inverse or by introducing Tikhonov (ridge or weight decay) regularization (with parameter, ρ), thus $\tilde{\underline{a}} = (K + \rho I)^{-1}\underline{y}$. The reason for this is that the kernel matrix, K, typically becomes ill-conditioned as the number of samples becomes large. It can be seen from Eq.(17) that the predicted output, $f(\underline{u})$, is the weighted sum of the kernels, $k(\underline{u}_i, \underline{u})$. As the sample size, N, increases, depending on the numerical precision of the computer used to run the programme, at least two rows of the kernel Gram matrix, K, will tend to co-linearity. Since our ultimate purpose is system identification in a noisy environment, we adopt regularisation and accept a biased solution.

4 Recovery of The Infinite Degree Volterra Model

Volterra series models are widely used in analyzing non-linear systems because they contain important characteristics of the physical systems and are qualitatively well-behaved, like linear finite impulse response models. At the same time, an infinite degree Volterra model can, in principle, give an arbitrarily accurate representation for the corresponding fading memory, nonlinear system, so the recovery of the model is of great importance for us in many cases.

Substituting Eq.(8) and (9) into (18), we have

$$\hat{y}(\underline{u}) = \sum_{i=1}^{N} a_i k(\underline{u}_i, \underline{u}) = \sum_{i=1}^{N} a_i \left[\sum_{n=0}^{\infty} \frac{1}{n!p^n} \left(\sum_{m=1}^{M} u_{im} u_m\right)^n\right]. \qquad (22)$$

We can expand the polynomial $\left(\sum_{m=1}^{M} u_{im}u_m\right)^n$ in Eq.(22) to rewrite it as

$$\begin{aligned}\hat{y}\left(\underline{u}\right) &= \frac{1}{p^0}\sum_{i=1}^{N} a_i + \sum_{m_1=1}^{M}\left(\frac{1}{p^1}\sum_{i=1}^{N} a_i u_{im_1}\right) u_{m_1} \\ &+ \sum_{m_1=1}^{M}\sum_{m_2=1}^{M}\left(\frac{1}{2!p^2}\sum_{i=1}^{N} a_i u_{im_1}u_{im_2}\right) u_{m_1}u_{m_2} \\ &+ \cdots + \sum_{m_1=1}^{M}\sum_{m_2=1}^{M}\cdots\sum_{m_n=1}^{M} \\ &\left(\frac{1}{n!p^n}\sum_{i=1}^{N} a_i u_{im_1}\ldots u_{im_n}\right) u_{m_1}\ldots u_{m_n} \\ &+ \ldots \end{aligned} \tag{23}$$

Given that

$$h_n = \frac{1}{n!p^n}\sum_{i=1}^{N} a_i \prod_{j=1}^{n} u_{im_j}. \left(n = 0, 1, \ldots, \infty\right), \tag{24}$$

Eq.(23) is equivalent to the following,

$$\hat{y}\left(\underline{u}\right) = h_0 + \sum_{n=1}^{\infty}\left\{\sum_{m_1=1}^{M}\cdots\sum_{m_n=1}^{M} h_n\left(m_1, \ldots, m_n\right)\prod_{j=1}^{n} u_{m_j}\right\}, \tag{25}$$

which is the same as the form of the infinite degree Volterra series expansion, (1). Therefore, we find that the original infinite degree Volterra series model can be recovered from the kernel model. The coefficient of each term, h_n, is given by Eq.(24).

5 Example

We choose examples of two Wiener-type systems (see figure 1) to illustrate the method. The first requires a model of infinite polynomial degree but has a finite memory length and is thus exactly matched to our solution. The second has infinite memory length and the same non-linearity. The linear block is designed, for illustrative purposes, so that the number of sizeable components of its impulse response function is small. We only examine the first and second order generalised frequency response functions.

5.1 Finite Memory, Infinite Degree Wiener Model

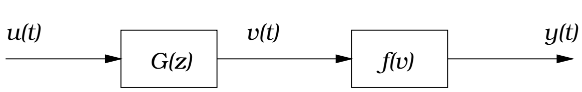

Figure 1: Linear-Non-linear Wiener model

The process is given as

$$y_t = \frac{e^{-v_t^2}}{1 + e^{-5v_t}} \tag{26}$$

$$v_t = 0.5u_t + 2.5u_{t-1} + 1.7u_{t-2}. \tag{27}$$

i.e. $M = 3$. The Taylor series of y_t in terms of v_t is

$$y_t = 0.5 + 1.25v_t - 0.5v_t^2 + \mathcal{O}\left(v_t^3\right). \tag{28}$$

Substituting Eq.(27) into (28), we have

$$\begin{aligned} y_t &= 0.5 + 1.25\left(0.5u_t + 2.5u_{t-1} + 1.7u_{t-2}\right) \\ &-0.5\left(0.5u_t + 2.5u_{t-1} + 1.7u_{t-2}\right)^2 \\ &+\mathcal{O}\left(v_t^3\right). \end{aligned} \tag{29}$$

The corresponding first and second order Volterra kernels are

$$h_1 = \begin{bmatrix} 0.625 & 3.125 & 2.125 \end{bmatrix}$$

$$h_2 = -\begin{bmatrix} 0.125 & 0.625 & 0.425 \\ 0.625 & 3.125 & 2.125 \\ 0.425 & 2.125 & 1.445 \end{bmatrix}.$$

Without introducing noise, the model was simulated to generate $(1998+2M)$ pairs[2] of input-output samples in which the input signal sequence, $\{u_t\}$, was uniformly distributed between 0 and 0.1. The regularization parameter, ρ, and the parameter, p, were chosen to be 1×10^{-9} and 0.009, respectively. This value of ρ was selected as the smallest one that avoided ill-conditioning of K. p was chosen because it led to the minimum difference between the Volterra kernels of the true and the estimated model. Assuming M was known, we applied the exponential kernel method to estimate the system. The resulting performance of the simulation in the time domain (not shown) has a mean square error on the order of 10^{-11}.

The estimated first and second order Volterra kernels are

$$\hat{h}_1 = \begin{bmatrix} 0.6246 & 3.1230 & 2.1236 \end{bmatrix}$$

$$\hat{h}_2 = -\begin{bmatrix} 0.1272 & 0.6020 & 0.4048 \\ 0.6020 & 3.0123 & 2.0543 \\ 0.4048 & 2.0543 & 1.3943 \end{bmatrix}.$$

Theoretically, the infinite degree, finite length Volterra series discussed in this paper can simulate the target model exactly. The difference between the true values and the estimated values was caused by machine imprecision and bias incurred through regularization. The corresponding first and second order generalised frequency responses are shown as follows.

As shown in figure 2, the estimated gain and phase of the first order frequency response are indistinguishable. In the second order frequency response figures $4-6$, the maximum absolute errors in the gain and phase are approximately 1.3dB and 11 degrees, respectively.

[2]Half of the data are used for training and the remainder for testing, ensuring 1000 samples regardless of M.

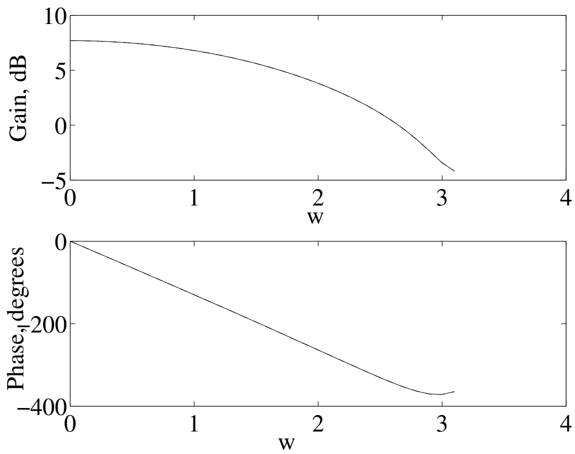

Figure 2: Target first order frequency response ('–') specified in Eqs.(26), (27) and the estimate ('- -') with $M = 3, p = 0.009$

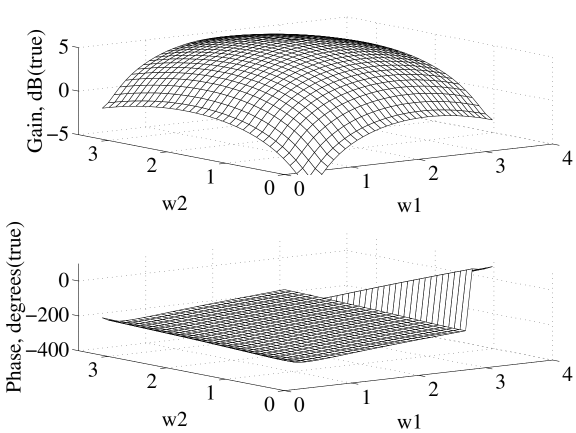

Figure 3: Target second order frequency response specified in Eqs.(26), (27)

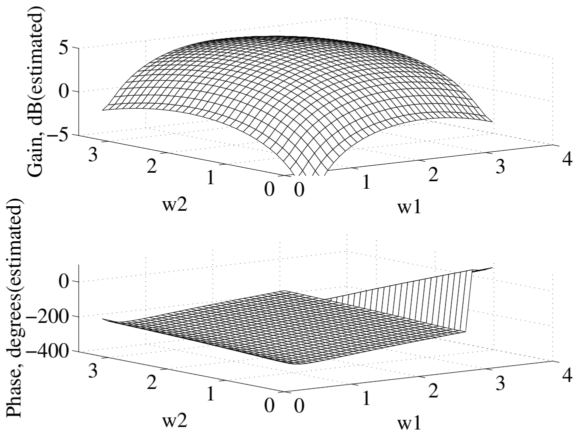

Figure 4: Estimated second order frequency response with $M = 3, p = 0.009$

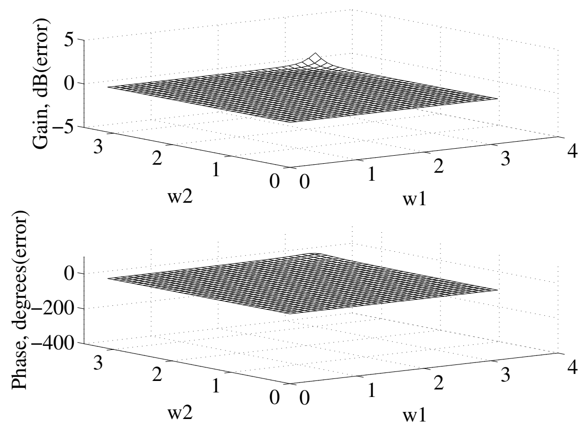

Figure 5: Differences between the true and the estimated gain and phase of the second order frequency response for the finite memory, infinite degree Wiener model

5.2 Infinite Memory, Infinite Degree Wiener Model

The structure of the model is the same as above. But in this case, the linear process component is given as

$$v_t = -0.5v_{t-1} - 0.1v_{t-2} + 0.5u_{t-1}. \tag{30}$$

Thus, theoretically, this model has infinite memory length. But as shown in figure 6, the linear impulse response sequence fades in 5 to 7 seconds.

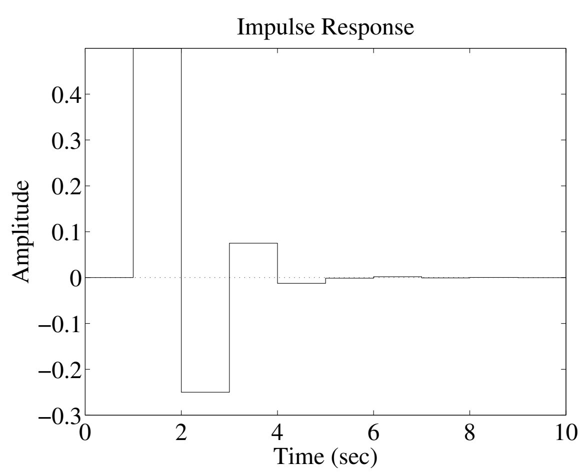

Figure 6: The impulse response of the linear block specified by Eq.(30)

By substituting Eq.(30) into (28), the first and second order Volterra kernels of the model can be computed,

$$h_1 = \big[0 \quad 0.625 \quad -0.3125 \quad 0.0938 \quad \cdots$$

$$\cdots \quad -0.0156 \quad -0.0016 \quad 0.0023 \quad -0.0010\big].$$

$$h_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & \cdots \\ 0 & -0.1250 & 0.0625 & -0.0188 & \cdots \\ 0 & 0.0625 & -0.0313 & 0.0094 & \cdots \\ 0 & -0.0188 & 0.0094 & -0.0028 & \cdots \\ 0 & 0.0031 & -0.0016 & 0.0005 & \cdots \\ 0 & 0.0003 & -0.0002 & 0 & \cdots \\ 0 & -0.0005 & 0.0002 & -0.0001 & \cdots \\ 0 & 0.0002 & -0.0001 & 0 & \cdots \end{bmatrix}$$

$$\begin{bmatrix} \cdots & 0 & 0 & 0 & 0 \\ \cdots & 0.0031 & 0.0003 & -0.0005 & 0.0002 \\ \cdots & -0.0016 & -0.0002 & 0.0002 & -0.0001 \\ \cdots & 0.0005 & 0 & -0.0001 & 0 \\ \cdots & -0.0001 & 0 & 0 & 0 \\ \cdots & 0 & 0 & 0 & 0 \\ \cdots & 0 & 0 & 0 & 0 \\ \cdots & 0 & 0 & 0 & 0 \end{bmatrix}.$$

The input samples, $\{u_t\}$, are uniformly distributed between 0 and 1 and $M = 5$. The parameters, ρ and p, are set to be 1×10^{-5} and 1, respectively. Again, the parameter configuration was determined in terms of the minimum difference between the Volterra kernels of the true and the estimated model. Again excellent predictive performance was obtained with a mean square error on the order of 10^{-5}.

The associated gain and phase of the first and second order frequency response are shown in figures 7-9. The resulting estimated first and second order Volterra kernels are

$$\hat{h}_1 = \begin{bmatrix} 0.0050 & 0.5912 & -0.2937 & 0.0878 & -0.0131 \end{bmatrix}$$

$$\hat{h}_2 = \begin{bmatrix} -0.0014 & 0 & -0.0019 & 0.0010 & -0.0015 \\ 0 & -0.1233 & 0.0622 & -0.0183 & 0.0043 \\ -0.0019 & 0.0622 & -0.0306 & 0.0096 & -0.0004 \\ 0.0010 & -0.0183 & 0.0096 & -0.0043 & 0.0016 \\ -0.0015 & 0.0043 & -0.0004 & 0.0016 & 0.0047 \end{bmatrix}.$$

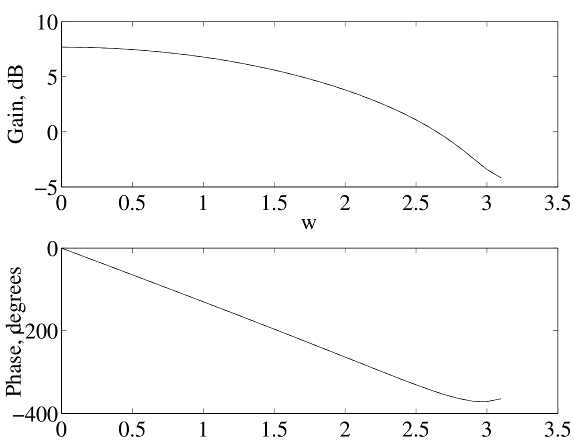

Figure 7: Target first order frequency '–'response specified in Eqs.(26), (30) and the estimate '- -'with $M = 5, p = 1$.

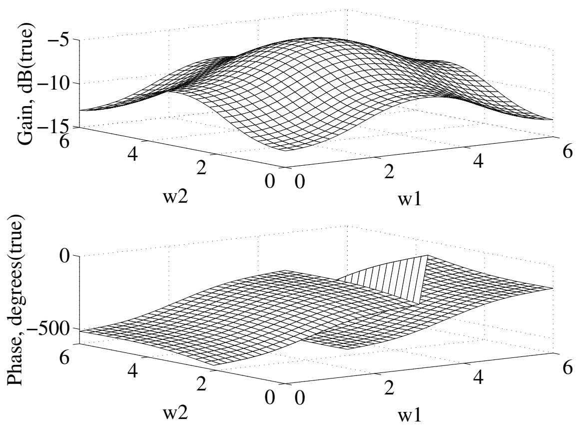

Figure 8: Target second order frequency response specified in Eqs.(26), (30).

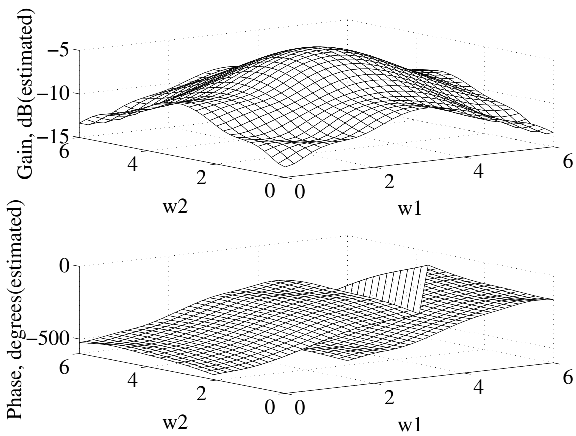

Figure 9: Estimated second order frequency response with $M = 5, p = 1$.

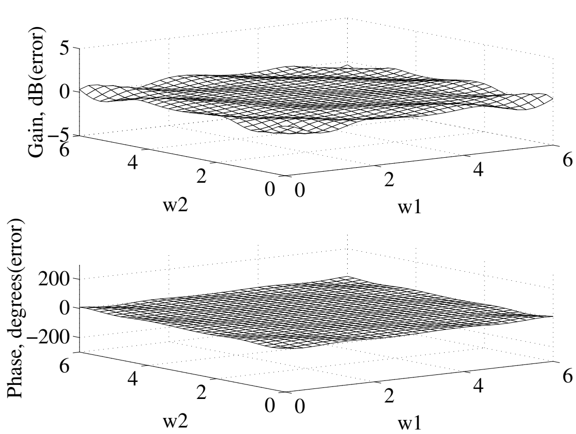

Figure 10: The differences between the true and the estimated gain and phase of the second order frequency response for the infinite memory, infinite degree Wiener model.

The error graphs are shown in figure 10.

As shown in figure 10, the maximum absolute error in the gain and the phase of the second order frequency response are approximately 0.7dB and 10 degrees, respectively. The resulting Volterra model could not capture the behavior of the original system when the memory length, M, was set to be less than 5.

6 Conclusion

By using the exponential kernel method, we can, in principle, approximate any system which can be represented by a finite memory, infinite degree Volterra series to arbitrary accuracy. The original terms of the Volterra series can also be recovered by using the algorithms in section 4.

A key step in training such a model is to compute the coefficient, $\underline{a}$, in Eq.(17). As is shown in section 5, even though $\underline{a}$ can be computed from Eq.(19) in principle, it is often the case that regularization must be employed to get the solution, $\underline{\tilde{a}}$, when the kernel Gram matrix is poorly conditioned, which, of course, induces bias. It should be noted that, despite this drawback, the exponential kernel method has the potential to solve a wide range of non-linear system identification problems because the model that is used in this method is of infinite degree and leaves only the finite memory length as a source of approximation error. This maintains the restriction of the proposed methodology to systems with "fading memory". We note that, while regularization introduces bias, its introduction does not appear unduly to harm the estimates.

A major advantage of the method is that the computational burden associated with direct estimation of Volterra series is made manageable and scales with sample size, N. This means that while the technique is still restricted to fading memory systems, the memory length imposes no particular limitation owing to the low computational cost of the dot product in $\mathbb{R}^M$.

Further work in the area is underway. In particular, methods to reduce numerical sensitivity; direct computation of the generalised frequency response functions from kernel representations, and the performance of the method in noisy environments (input and output) are under investigation.

References

[1] Aronszajn, N. (1950). Theory of reproducing kernels. *Transactions of the American Mathematical Society 68, 337-404.*

[2] Boyd, Stephen and Chua, Leon O. (1985). Fading memory and the problem of approximating nonlinear operators with Volterra series. *IEEE Transactions on Circuits and Systems, Vol. Cas-32, No. 11.*

[3] Dodd, T. J. and Harrison, R. F. (2002a). A new solution to Volterra series estimation. *CD-Rom Proceedings of the 2002 IFAC World Congress.*

[4] Dodd, T. J. and Harrison, R. F. (2002b). Estimating Volterra filters in Hilbert space. *Proceedings of IFAC International Conference on Intelligent Control Systems and Signal Processing, Faro.*

[5] Drezet, P. M. L. (2001). Kernel Methods and Their Application to Systems Identification and Signal Processing. *PhD Thesis, The University of Sheffield, Sheffield.*

[6] De Figueiredo, Rui J. P. (1983). A generalized Fock space framework for non-linear system and signal analysis. *IEEE Transactions on Circuits and Systems. Vol. Cas-30, No. 9.*

[7] De Figueiredo, Rui J. P. and Dwyer, III, Thomas A. W. (1980). A best approximation framework and implementation for simulatoin of large-scale non-linear systems. *IEEE Transactions on Circuits and Systems. Vol. Cas-27, No. 11.*

[8] Harrison, R. F. (1999). Computable Volterra filters of arbitrary degree. *MAE Technical Report No. 3060, Princeton University.*

[9] Schetzen, Martin (1980). *The Volterra and Wiener Theories of Non-linear Systems.* New York; Chichester (etc.): Wiley.

[10] Wahba, G. (1990). *Spline Models for Observational Data. SIAM. Series in Applied Mathematics. Vol. 50. Philadelphia.*

[11] Zyla, L. V. and De Figueiredo, Rui J. P. (1983). Non-linear system identification based on a Fock space framework. *SIAM J. Control and Optimization. Vol. 21. No. 6.*