Journal of Sound and Vibration (1998) **213**(5), 915–941

FREQUENCY DOMAIN ANALYSIS OF NARX NEURAL NETWORKS

J. E. Chance, K. Worden and G. R. Tomlinson

Department of Mechanical Engineering, University of Sheffield, Mappin Street, Sheffield S1 3JD, England

(*Received* 14 *January* 1997, *and in final form* 6 *February* 1998)

A method is proposed for interpreting the behaviour of NARX neural networks. The correspondence between time-delay neural networks and Volterra series is extended to the NARX class of networks. The Volterra kernels, or rather, their Fourier transforms, are obtained via harmonic probing. In the same way that the Volterra kernels generalize the impulse response to non-linear systems, the Volterra kernel transforms can be viewed as higher-order analogues of the Frequency Response Functions commonly used in Engineering dynamics; they can be interpreted in much the same way.

1. INTRODUCTION

The recent past has seen an enormous increase in proposals for the use of Artificial Neural Networks (ANNs) for Engineering applications; particularly in control systems, fault diagnostic systems and "smart" structures. However, despite the intense activity of academia and industrial R&D departments, the uptake of ANN technology by industry–European industry in particular—has been minimal. The reason for this is the apparent "black box" nature of ANNs which makes them resistant to traditional methods of certification and therefore excludes them from safety-critical applications.

This paper offers a partial remedy to this problem for a restricted class of neural networks often used in modelling and control applications—the so-called NARX (Non-linear Auto-Regressive with eXogenous) networks, a superset of the so-called Time-Delay Neural Networks (TDNN). (The former terminology is from the literature of time-series analysis and economic theory.) The most substantial body of work on these networks has been produced by Billings and co-workers and a representative sample of references can be found in reference [1].

The Volterra functional series [2] is often used in the system identification of non-linear input–output processes. They provide a description of the dynamics which is invariant of the excitation conditions. There are many applications to the analysis of both Engineering and Physiological systems. In the field of Physiology, the Volterra kernels (which are the generalized expansion coefficients of the series) allow a qualitative explanation of the response of real neurons [3, 4]. Despite the utility of the representation, methods of calculating the kernels quickly and accurately have proved difficult to find [5]. Recent work by Wray and Green [6] and Marmarelis [4] exploits the fact that the Volterra series representation corresponds closely with representation by a TDNN; an algorithm has been obtained which allows the Volterra kernels to be computed using the weights from a TDNN model of the input–output process.

0022–460X/98/250915 + 27 $25.00/0/sv981539

In engineering applications, it is often the Fourier transforms of the Volterra kernels, so-called Higher-order Frequency Response Functions (HFRFs), which prove to be more informative. Just as the standard FRF or, loosely speaking, the Transfer Function, describes how inputs at certain frequencies will lead to elevated outputs i.e., resonances, the HFRFs yield information about how energy is transferred *between* frequencies in non-linear systems leading to the phenomenon of combination resonances. In contrast to the situation for the kernels themselves, a fast and accurate means of obtaining the HFRFs does exist. This is based on fitting a non-linear time-series model—a NARMAX model—to the input–output data from the system and extracting the HFRFs using a *harmonic probing* algorithm [7]. An application of this procedure to the analysis of non-linear wave forces, which discusses in detail the interpretation of HFRFs, is documented in reference [8].

The object of the current paper is to show how HFRFs can be obtained directly from neural network models. A more general class of networks is considered i.e., the NARX class which includes the TDNN networks as a subclass. The HFRFs completely characterize the network at each order of non-linearity and therefore offer a means of validating and possibly verifying* neural network models used in identification and control. Given a NARX network it will be possible using these methods to analyse qualitatively the response of the system to signals containing multiple harmonic components. A by-product of the analysis is that the procedures, in principle, be used to *identify* HFRFs for input–output systems by training a NARX network to model the system.

The layout of the paper is as follows: section 2 introduces the relevant material about the Volterra series. Section 3 introduces the concept of the HFRF and discusses how they can be interpreted. The method of harmonic probing is introduced in section 4 and applied to the case of NARX model implemented as a Multi-Layer Perceptron (MLP). Sections 5 and 6 respectively derive the first two HFRFs for MLP networks with polynomial activation functions and Radial Basis Function (RBF) networks. Section 7 illustrates the theory by computing the HFRFs for a Duffing oscillator system via trained neural network models.

2. THE VOLTERRA SERIES

In the time-domain analysis of linear dynamical systems the *impulse response function* $h(\tau)$ is known to characterize the system completely. For such a system, excited by an input signal $x(t)$, the response $y(t)$ is given by the convolution integral,

$$y(t)=\int_{-\infty}^{\infty}\mathrm{d}\tau h(\tau)x(t-\tau). \tag{1}$$

(This is sometimes referred to as Duhamel's integral.)

This relationship is manifestly linear and will not hold for non-linear systems; however, the theory was extended by Volterra [2] in the early part of this century to cover the more general case. The output of a non-linear system is composed of additional higher-order contributions. Volterra showed that the total response, $y(t)$, is given by,

$$y(t)=y_0+y_1(t)+y_2(t)+y_3(t)+\cdots+y_n(t), \tag{2}$$

* The distinction between the two processes is important; the process of validation establishes if the model conforms to requirements, the process of verification is the procedure by which correct operation is assured [9].

where y_0 is a constant and,

$$y_1(t) = \int_{-\infty}^{\infty} \mathrm{d}\tau h_1(\tau_1) x(t - \tau_1), \tag{3}$$

$$y_2(t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathrm{d}\tau_1 \, \mathrm{d}\tau_2 \, h_2(\tau_1, \tau_2) x(t - \tau_1) x(t - \tau_2), \tag{4}$$

and the general term is,

$$y_n(t) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \mathrm{d}\tau_1 \, \mathrm{d}\tau_2 \cdots \mathrm{d}\tau_n h_n(\tau_1, \tau_2 \cdots \tau_n) x(t - \tau_1) x(t - \tau_2) \cdots x(t - \tau_n). \tag{5}$$

This is essentially a generalization of the standard Taylor series to the case of *functionals* i.e., mappings between functions. The generalized coefficients of the series h_n, are the nth order *Volterra kernels*, and these can be thought of as multi-dimensional, or higher-order, impulse response functions. The y_n terms are unique but the Volterra kernels h_n are not. However, it can be shown that there is always a representation using symmetric kernels which *is* unique [10]. (Symmetric in this case simply means that the kernels are invariant under the interchange of any of their arguments.) The series provides a representation of a given functional or system $y(t) = S[x(t)]$, which is insensitive to the input $x(t)$, provided that the system is time-invariant and contains only analytic non-linearities (i.e., the functions representing the non-linearities in the equations of motion have a convergent Taylor expansion) [11].

One important caveat is based on the observation that the Volterra series is essentially a generalized Taylor expansion and as such it will have an associated radius of convergence. For input signals $x(t)$ which have excursions below a certain bound, the series will converge and the representation is valid. However, for signals of higher amplitude, the series may diverge, or require so many terms before an accurate representation is reached, that the model has little value. A particular problem associated with non-linear systems is that of bifurcation, or multi-valued response; the Volterra series is by definition, single-valued and will not give a representation of systems over this range of their behaviour. It is assumed in this paper that the systems are weakly non-linear, i.e., far from bifurcation.*

Due to the fact that a given frequency component of the system response will usually contain contributions from several Volterra kernels, they are difficult to measure in practice and there is no generally accepted method of calculating the kernels experimentally. The problem is essentially one of inter-kernel interference [13].

To overcome the measurement problem, the theory was extended by Wiener [14] and the Wiener series has been the focus of a great deal of interest since. The kernels of the Wiener series avoid the problem of interaction because they are orthogonal (or decorrelated in a sense) if the input is a white Gaussian noise sequence. In the limit where the level of excitation tends to zero i.e., where the RMS of $x(t)$ is low, the Wiener kernels approach the system Volterra kernels. Because of the orthogonality, a method for estimating the Wiener kernels (and hence approximating the Volterra kernels), was proposed by Lee and Schetzen [15], based on cross-correlations. The method basically

* The word *weakly* is being used somewhat loosely here. Under conditions of rigor, this would imply that the non-linearity satisfied appropriate stringent continuity conditions to eliminate abrupt changes in its form, here the word is used in the sense of Wiener and Spina [12] and simply means that the Volterra representation exists and is convergent.

extends the classical method of estimating FRFs from cross-power and auto-power spectra by computing higher-order cumulants of the input and output processes.

Unfortunately, the Wiener method also suffers from a number of limitations. First, is the fact that the theory requires a white Gaussian input signal; this is physically impossible because such a signal would have infinite bandwidth and therefore contain infinite power. Any physically realisable signal must have finite bandwidth and can only be white over a limited range of frequencies. The second problem is that experimentally Wiener kernels are highly susceptible to noise and require large amounts of data in order to "average away" the noise. An additional problem faced in structural dynamics is that even a filtered input signal will not be white over a given frequency band due to interactions between the exciter and the structure. As a consequence of these factors, even under the most favourable experimental conditions, the measured Wiener kernels may only be poor approximations to the system Volterra kernels. The Lee and Schetzen [15] cross-correlation method has recently been superceded by the Toeplitz matrix inversion technique of Korenberg and Hunter [5]; this in turn appears to offer little advantage over the neural network based method of Wray and Green [6].

3. HIGHER-ORDER FREQUENCY RESPONSE FUNCTIONS

It is well-known that linear systems admit dual time and frequency-domain representations. For a linear input–output map, equation (1) shows how to compute the response $y(t)$ for any input $x(t)$, given the system impulse response function $h(t)$. The corresponding frequency-domain expression is simply obtained by taking the Fourier transform of both sides, noting that the right side is a convolution. The result is,

$$Y(\omega) = H(\omega)X(\omega), \tag{6}$$

where,

$$H(\omega) = \int_{-\infty}^{\infty} \mathrm{d}\omega\ \mathrm{e}^{-\mathrm{i}\omega t} h(t)$$

is the system FRF, and $Y(\omega)$ and $X(\omega)$ have similar definitions. The utility of this representation lies in the fact that the input to output transformation is simply multiplication by a function. Also, it will be shown that much useful information is summarized in $H(\omega)$.

Less well-known is the fact that non-linear systems also have a dual frequency domain representation based on the Volterra series. By direct extension of the linear case, the higher-order FRFs $H_n(\omega_1, \ldots, \omega_n)$, can be defined as the multi-dimensional Fourier transforms of the kernels,

$$H_n(\omega_1, \ldots, \omega_n) = \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} \mathrm{d}\tau_1 \cdots \mathrm{d}\tau_n h_n(\tau_1, \ldots, \tau_n)\, \mathrm{e}^{-\mathrm{i}(\omega_1\tau_1 + \cdots + \omega_n\tau_n)}, \tag{7}$$

with inverse,

$$h_n(\tau_1, \ldots, \tau_n) = \frac{1}{(2\pi)^n} \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} \mathrm{d}\omega_1 \cdots \mathrm{d}\omega_n H_n(\omega_1, \ldots, \omega_n)\, \mathrm{e}^{+\mathrm{i}(\omega_1\tau_1 + \cdots + \omega_n\tau_n)}. \tag{8}$$

Symmetry of the kernels implies symmetry of the kernel transforms so, for example, $H_2(\omega_1, \omega_2) = H_2(\omega_2, \omega_1)$.

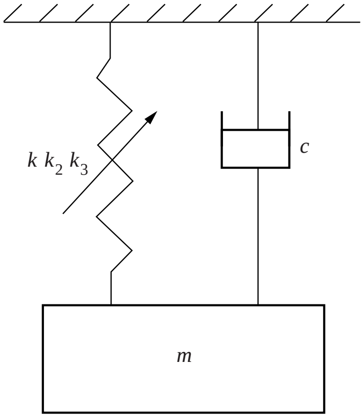

Figure 1. Asymmetric SDOF Duffing oscillator with linear, quadratic and cubic stiffnesses k, k_2 and k_3, respectively, mass m and damping coefficient c.

It is then a straightforward matter to obtain the frequency-domain dual of expression (2),

$$Y(\omega) = Y_1(\omega) + Y_2(\omega) + Y_3(\omega) + \cdots \tag{9}$$

where*

$$Y_1(\omega) = H_1(\omega)X(\omega), \tag{10}$$

$$Y_2(\omega) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} \mathrm{d}\omega_1 H_2(\omega_1, \omega - \omega_1)X(\omega_1)X(\omega - \omega_1), \tag{11}$$

$$Y_3(\omega) = \frac{1}{(2\pi)^2}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \mathrm{d}\omega_1\, \mathrm{d}\omega_2\, H_3(\omega_1, \omega_2, \omega - \omega_1 - \omega_2)X(\omega_1)X(\omega_2)X(\omega - \omega_1 - \omega_2). \tag{12}$$

In order to discuss the interpretation of these quantities, an example will be given. Consider the Duffing oscillator system shown in Figure 1, specified by the equation of motion,

$$m\ddot{y} + c\dot{y} + ky + k_2y^2 + k_3y^3 = x(t), \tag{13}$$

where overdots denote differentiation with respect to time. The first three HFRFs are given by,

$$H_1(\omega) = \frac{1}{-m\omega^2 + \mathrm{i}c\omega + k}, \tag{14}$$

$$H_2(\omega_1, \omega_2) = -\frac{k_2}{2} H_1(\omega_1)H_1(\omega_2)H_1(\omega_1 + \omega_2) \tag{15}$$

* Note that the expressions can be made symmetrical at the expense of introducing a delta-function and an extra integration i.e.,

$$Y_2(\omega) = \frac{1}{2\pi}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \mathrm{d}\omega_1\, \mathrm{d}\omega_2\, \delta(\omega - \omega_1 - \omega_2)H_2(\omega_1, \omega_2)X(\omega_1)X(\omega_2).$$

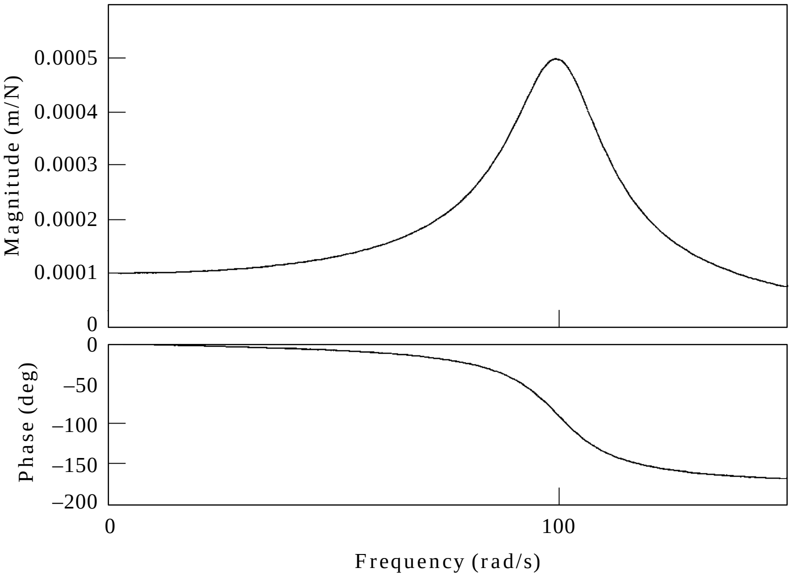

Figure 2. Amplitude and phase of $H_1(\omega)$ from Duffing oscillator.

and

$$H_3(\omega_1, \omega_2, \omega_3) = -\tfrac{1}{6}H_1(\omega_1 + \omega_2 + \omega_3) \times \{4k_2(H_1(\omega_1)H_2(\omega_2, \omega_3) + H_1(\omega_2)H_2(\omega_3, \omega_1) + H_1(\omega_3)H_2(\omega_1, \omega_2)) + k_3H_1(\omega_1)H_1(\omega_2)H_1(\omega_3)\}. \quad (16)$$

Note that the constant k_2 multiplies the whole expression for H_2, so that if the square-law term is absent from the equation of motion, H_2 vanishes. This reflects a quite general property of the Volterra series; if all non-linear terms in the equation of motion for a system are odd powers of x or y, then the associated Volterra series has no even-order kernels. As a consequence it will possess no even-order HFRFs. It is also a general property of systems that all higher-order FRFs can be expressed in terms of H_1. The exact form of the expression depends on the system. The expressions were obtained using the harmonic probing algorithm which is discussed in detail in the next section.

Having presented some concrete examples, the interpretation of the higher-order FRFs can be discussed. The magnitude of the expression (14) (it is of course a complex function) is given in Figure 2 on the frequency interval 0–150 rad/s. The interpretation of this figure, (traditionally given together with the phase and universally called the Bode plot), is well known; the peak in the magnitude at $\omega = \omega_r = 99$ rad/s shows that for this frequency of excitation the amplitude of the linear part of the response $y_1(t)$ is a maximum. The magnitude plot thus allows the immediate identification of those excitation frequencies at which the vibration level of the system is likely to be high.

Interpretation of the second-order FRF is also straightforward. The magnitude of H_2 for the Duffing system above is given in Figure 3 as a surface plot over the ω_1, ω_2 plane. The frequency ranges for the plot are the same as for H_1 in Figure 2. A number of ridges are observed. These are in direct correspondence with the peak in H_1 as follows. According to equation (15), H_2 is a constant multiple of $H_1(\omega_1)H_1(\omega_2)H_1(\omega_1 + \omega_2)$. As a consequence, H_2 possesses local maxima at positions where the H_1 factors have maxima. Consequently there are two ridges in the H_2 surface corresponding to the lines $\omega_1 = \omega_r$ and $\omega_2 = \omega_r$. These

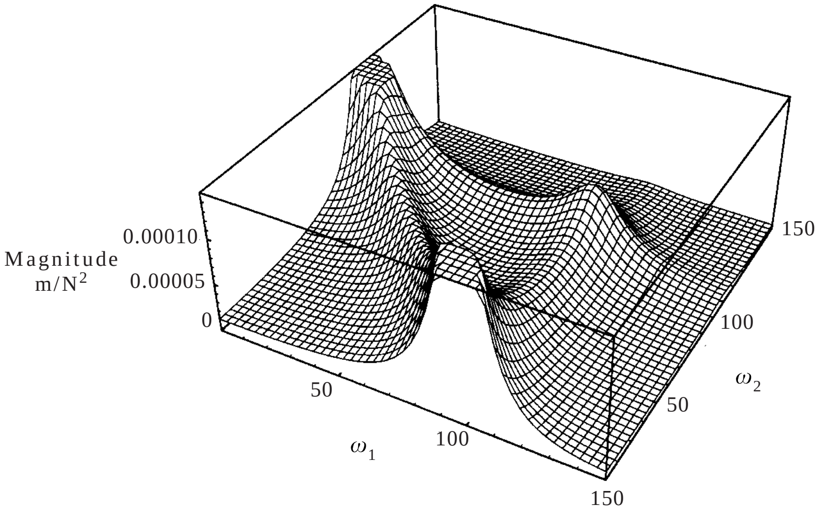

Figure 3. Second-order FRF, $H_2(\omega_1, \omega_2)$ surface from Duffing oscillator.

are along lines parallel to the frequency axes. In addition, H_2 has local maxima generated by the $H_1(\omega_1 + \omega_2)$ factor along the line $\omega_1 + \omega_2 = \omega_r$. This ridge has an important consequence; it indicates that one can expect a maximum in the second-order output $y_2(t)$ if the system is excited by two sinusoids whose sum frequency is the linear resonant frequency. This clearly shows why estimation of a FRF by linear methods is inadequate for non-linear systems; such a FRF would usually indicate a maximum in the output for a harmonic excitation close to the linear resonant frequency. However, it would fail to predict that one could excite a large non-linear component in the output by exciting at $\omega = \omega_r/2$; this is a consequence of the trivial decomposition $e^{i(\omega_r/2)t} = \frac{1}{2}e^{i(\omega_r/2)t} + \frac{1}{2}e^{i(\omega_r/2)t}$ which means that the signal can be regarded as a "two-tone" input with a sum frequency at the linear resonance ω_r. The importance of the second-order FRF is now clear. It reveals those pairs of excitation frequencies which will conspire to produce large levels of vibration as a result of second-order non-linear effects.

The arguments above show that the higher FRFs provide directly visible information about the possible excitation of large non-linear vibrations through the cooperation of certain frequencies.

In order to see the important structure in the H_2, it is often sufficient to plot only the leading diagonal i.e., $H_2(\omega, \omega)$ as in Figure 4 for system (13). This format also allows simple comparisons between the functions.

In the following sections it is shown how to obtain HFRFs for NARX networks based on sigmoidal Multi-Layer Perceptrons (MLPs), Radial-Basis Function Networks (RBFs) and polynomial MLPs.

4. HARMONIC PROBING OF NARX MODELS: THE MULTI-LAYER PERCEPTRON

If the governing equations of motion are known, the HFRFs of a system can be obtained analytically by the use of the *harmonic probing* algorithm, introduced by Bedrosian and Rice [16]. Although this was originally designed for continuous-time systems, the algorithm was extended to the type of discrete-time systems considered here, by Billings and Tsang [7].

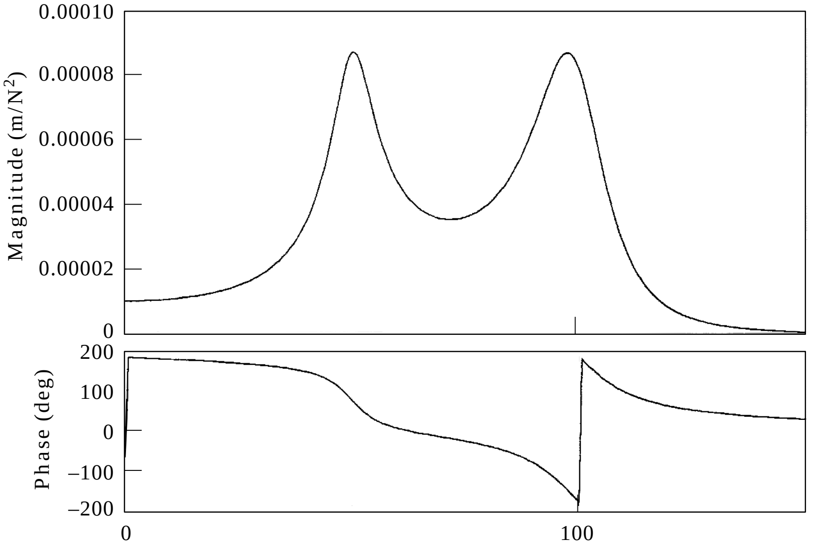

Figure 4. Leading diagonal ($\omega_1 = \omega_2$) of $H_2(\omega_1, \omega_2)$ surface from Duffing oscillator.

The TDNN is simply a mapping from a discrete set of lagged input measurements to the predicted output,

$$\hat{y}_i = f(x_i, x_{i-1}, x_{i-2}, \ldots, x_{i-N}), \tag{17}$$

where $x_i = x(t - (i-1)\Delta t)$. The non-linear function f is implemented as a sum of sigmoidal functions of weighted linear combinations of the inputs. In the usual terminology of time-series modelling, this would be called an NX model, as it is Non-linear and the predicted output is regressed on a series on external or eXogenous inputs. It is well-known in system identification theory, that this type of model usually requires many lagged values of the input in order to cover the memory of the system. Much more *parsimonious* models i.e., containing far fewer terms, can be obtained by introducing a dependence on past values of the output i.e.,

$$\hat{y}_i = f(y_{i-1}, \ldots, y_{i-n_y}; x_i, \ldots, x_{i-(n_x-1)}), \tag{18}$$

where y_i is defined similarly to x_i.

This type of model is referred to as a NARX model, the AR part of the term standing for *Auto-Regressive*, meaning that the current y depends on past values of y. The most general models include some means of modelling the noise process, usually expressing the system noise as the result of passing white Gaussian noise through a non-linear filter; introducing this so-called *Moving Average* filter leads to the generic NARMAX model [17]. In the general situation, noise must be taken into account as it can lead to incorrect or *biased* estimates of the system parameters, in this case the network weights. Although the theory and practice for noise models in polynomial NARMAX models is established, the problem for neural network models is more complex and has not yet been satisfactorily resolved. A first step has been taken via the inclusion of linear noise models [18]. The approach taken here is to ignore the issue of noise. In the first case, the noise model is discarded before computing the HFRFs, in the second, the demonstration of the procedures will use data from numerical simulation which is essentially noise-free.

In reference [19], it is shown that polynomial NARX models exist for all input–output systems subject to very reasonable conditions. This fact, taken with the theorems of Cybenko [20] and Funahashi [21] which state that sigmoidal neural networks can approximate any function, leads to the conclusion that neural network NARX models exist for most input–output systems. The NARX networks are rather restricted examples of recurrent networks in that there is a feedback loop between the network output and a subset of the network inputs. Because of this, it could be argued that a fully recurrent network may be more appropriate for system modelling. In fact, it can be shown that NARX networks are essentially capable of computing any function that a fully recurrent network can [22–24]. Note that the existence of the representations does not guarantee that the iterative procedures used to train neural networks will arrive at this representation. The problem of local minima may arise.

Having established that there is some value in adopting a NARX network structure for system modelling, the following analysis will show how to obtain the HFRFs. First, it is necessary to establish notation. The Multi-Layer Perceptron structure for the NARX model is given in Figure 5. This implements the input–output map via the following function,

$$y_i = s + \sum_{j=1}^{n_h} w_j \tanh\left(\sum_{k=1}^{n_y} v_{jk} y_{i-k} + \sum_{m=0}^{n_x-1} u_{jm} x_{i-m} + b_j\right), \tag{19}$$

where y_i and x_i are the current response and excitation values; v_{jk} and u_{jm} are the weights of the connections between the jth hidden layer unit and the kth lag in y and mth lag in x input units, respectively; w_j is the connection weight from the jth hidden unit to the output unit; s is the bias weight for the output unit and b_j is the weight between the bias

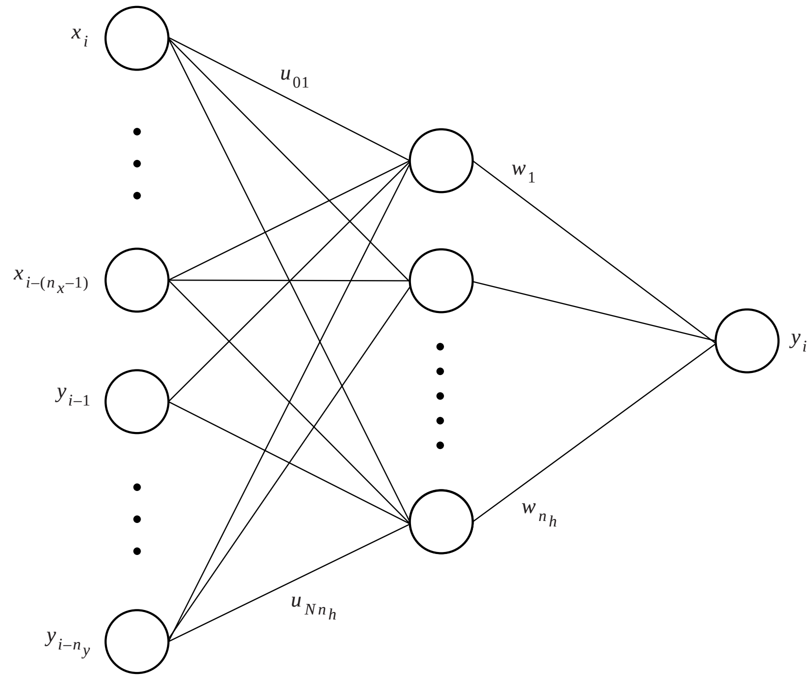

Figure 5. Example of an NARX type neural network.

element and the jth hidden unit. The caret previously over y has been discarded as no confusion can possibly arise as to when an equation furnishes an approximation.

In practice, the situation is a little more complicated than equation (19) implies. During training of the network, all *network* inputs are normalized to lie in the interval $[-1, 1]$ and the output is normalized onto the interval $[-0{\cdot}8, 0{\cdot}8]$. Equation (19) actually holds between the normalized quantities. The transformation back to the physical quantities, once the HFRFs have been calculated is extremely straightforward and the details are omitted here.

The NARX models considered here will arise from models of dynamical systems. In that case, a further simplification can be made. It is assumed that the effects of all the bias terms cancel overall, as the systems being modelled will not contain constant terms in their equations of motion. In dynamical systems this can always be accomplished with an appropriate choice of equilibrium position for y if the excitation x is also adjusted to remove its dc term. The reason is to eliminate the y_0 part of the response in the previously shown Volterra expansions. If this term is included, a slightly more complicated harmonic probing algorithm is required [25]; the generalization is in any case, not difficult.

The basis of the harmonic probing method is to examine the response of the system to certain very simple inputs. In order to identify $H_1(\omega)$, for example, the system is "probed" with the single harmonic,

$$x_i^p = \mathrm{e}^{\mathrm{i}\Omega t}. \tag{20}$$

Substituting this expresion into the Volterra series (2), the corresponding response is,

$$y_i^p = H_1(\Omega)\,\mathrm{e}^{\mathrm{i}\Omega t} + H_2(\Omega, \Omega)\,\mathrm{e}^{2\mathrm{i}\Omega t} + H_3(\Omega, \Omega, \Omega)\,\mathrm{e}^{3\mathrm{i}\Omega t} + \cdots \tag{21}$$

Now, consider the consequences of substituting expressions (20) and (21) into the network function (19) and expanding it as a polynomial. None of the higher-order terms in (21) can combine in any way to generate a component at the fundamental frequency of excitation Ω. As a result, if the coefficient of $\mathrm{e}^{\mathrm{i}\Omega t}$ is extracted from the resulting expression, the *only* HFRF which can appear is $H_1(\Omega)$; thus, the expression can be rearranged to given an analytical expression for H_1. First, note that equation (19) is in an inappropriate form for this operation as it stands. The reason being that the term of order n in the expansion of the tanh function will contain harmonics of all orders up to n, so extracting the coefficient of the fundamental requires the summation of an infinite series. The way around this problem is to use a trick of Wray and Green [6] and expand the tanh around the bias; this yields,

$$y_i = s + \sum_{j=1}^{n_h} w_j \sum_{t=0}^{\infty} \left\{ \frac{\tanh^{(t)}(b_j)}{t!} \left(\sum_{k=1}^{n_y} v_{jk} y_{i-k} + \sum_{m=0}^{n_x-1} u_{jm} x_{i-m} \right)^t \right\}, \tag{22}$$

so each term in the expansion is now a homogeneous polynomial in the lagged xs and ys. $\tanh^{(t)}$ is the tth derivative of tanh at b_j.

The only term in this expansion which can affect the coefficient of the fundamental harmonic is the linear one; therefore take,

$$y_i = \sum_{j=1}^{n_h} w_j \frac{\tanh^{(1)}(b_j)}{1} \left(\sum_{k=1}^{n_y} v_{jk} y_{i-k} + \sum_{m=0}^{n_x-1} u_{jm} x_{i-m} \right) \tag{23}$$

as the expression to be probed. Account must be taken of the effect of time-delays on the harmonic signals, this is straightforward to compute.

$$x_{i-k} = \Delta^k x_i = \Delta^k\,\mathrm{e}^{\mathrm{i}\Omega t} = \mathrm{e}^{-k\mathrm{i}\Omega\Delta t}\,\mathrm{e}^{\mathrm{i}\Omega t}, \tag{24}$$

$$y_{i-k} = \Delta^k y_i = \Delta^k H_1(\Omega)\, \mathrm{e}^{\mathrm{i}\Omega t}\, \mathrm{e}^{-k\mathrm{i}\Omega\Delta t} H_1(\Omega)\, \mathrm{e}^{\mathrm{i}\Omega t}, \tag{25}$$

where Δ is the backward shift operator.

Extracting the coefficient of $\mathrm{e}^{\mathrm{i}\Omega t}$ from equation (23) after substitution of x^p and y^p gives,

$$H_1(\Omega) = \sum_{j=1}^{n_h} u_j \tanh^{(1)}(b_j) \sum_{k=1}^{n_y} \Delta^k H_1(\Omega) + \sum_{j=1}^{n_h} w_j \tanh^{(1)}(b_j) \sum_{m=0}^{n_x-1} u_{jm}\Delta^m. \tag{26}$$

So finally, H_1 is obtained as,

$$H_1(\Omega) = \frac{\sum_{j=1}^{n_h} w_j \tanh^{(1)}(b_j) \sum_{m=0}^{n_x-1} u_{jm}\, \mathrm{e}^{-\mathrm{i}\Omega m\delta t}}{1 - \sum_{j=1}^{n_h} w_j \tanh^{(1)}(b_j) \sum_{k=1}^{n_y} v_{jk}\, \mathrm{e}^{-\mathrm{i}\Omega k\delta t}}. \tag{27}$$

The extraction of H_2 is a little more complicated, this requires probing with two independent harmonics, so,

$$x_i^p = \mathrm{e}^{\mathrm{i}\Omega_1 t} + \mathrm{e}^{\mathrm{i}\Omega_1 t}. \tag{28}$$

Computation using equations (2)–(4) shows the corresponding response is,

$$y_i^p = H_1(\Omega_1)\, \mathrm{e}^{\mathrm{i}\Omega_1 t} + H_1(\Omega_2)\, \mathrm{e}^{\mathrm{i}\Omega_2 t} + 2H_2(\Omega_1, \Omega_2)\, \mathrm{e}^{\mathrm{i}(\Omega_1+\Omega_2)t} + \cdots \tag{29}$$

The argument proceeds as for H_1; if these expressions are substituted into the network function (22), the only HFRFs to appear in the coefficient of the sum harmonic $\mathrm{e}^{\mathrm{i}(\Omega_1+\Omega_2)t}$, are H_1 and H_2, where H_1 is already known from equation (26). So as before, the coefficient can be rearranged to give an expression for H_2 in terms of the network weights and H_1. The only terms in equation (22) which are relevant for the calculation are those at first- and second-order. The calculation is straightforward but tedious and yields,

$$H_2(\Omega_1, \Omega_2) = \frac{1}{2!D} \sum_{j=1}^{n_h} w_j \frac{\tanh^{(2)}(b_j)}{2!} \{A_j + B_j + C_j\}, \tag{30}$$

where,

$$A_j = \sum_{k=1}^{n_y} \sum_{l=1}^{n_y} v_{jk} v_{jl} H_1(\Omega_1) H_1(\Omega_2)(\mathrm{e}^{-\mathrm{i}\Omega_1 k\delta t}\, \mathrm{e}^{-\mathrm{i}\Omega_2 l\delta t} + \mathrm{e}^{-\mathrm{i}\Omega_2 k\delta t}\, \mathrm{e}^{-\mathrm{i}\Omega_1 l\delta t}),$$

$$B_j = \sum_{k=0}^{n_x-1} \sum_{l=0}^{n_x-1} u_{jk} u_{jl}(\mathrm{e}^{-\mathrm{i}\Omega_1 k\delta t}\, \mathrm{e}^{-\mathrm{i}\Omega_2 l\delta t} + \mathrm{e}^{-\mathrm{i}\Omega_2 k\delta t}\, \mathrm{e}^{-\mathrm{i}\Omega_1 l\delta t}),$$

$$C_j = 2 \sum_{k=1}^{n_y} \sum_{l=0}^{n_x-1} v_{jk} u_{jl}(H_1(\Omega_1)\, \mathrm{e}^{-\mathrm{i}\Omega_1 k\delta t}\, \mathrm{e}^{-\mathrm{i}\Omega_2 l\delta t} + H_1(\Omega_2)\, \mathrm{e}^{-\mathrm{i}\Omega_2 k\delta t}\, \mathrm{e}^{-\mathrm{i}\Omega_1 l\delta t})$$

and

$$D = 1 - \sum_{j=1}^{n_h} w_j \tanh^{(1)}(b_j) \sum_{k=1}^{n_y} v_{jk}\, \mathrm{e}^{-\mathrm{i}(\Omega_1+\Omega_2)k\delta t}.$$

Derivation of H_3 is considerably more lengthy and requires probing with three harmonics. It is shown in the Appendix. The results section of this paper will present examples of these calculations for H_1 and H_2.

5. POLYNOMIAL ACTIVATION FUNCTION NETWORKS

The finite polynomial activation function provides a possible alternative to the sigmoidal, or tanh, function used thus far. Recent work by Marmarelis and Zhao [4] shows the polynomial neural network to be particularly efficient for the purpose of calculating Volterra kernels. They obtained significantly more accurate kernels when using these activation functions, compared to the more commonly employed sigmoidal function.

5.1. THEORY

The expressions for the first- and second-order HFRFs of NARX networks incorporating polynomial activation functions will now be derived. The processing is performed using the function,

$$f(z) = a_0 + a_1 z + a_2 z^2 + a_3 z^3 + \cdots + a_n z^n, \tag{31}$$

where a_i are the polynomial coefficients. The network equation is given by,*

$$\begin{aligned} y_i = s + \sum_{j=1}^{n_h} w_j \Bigg[a_0 &+ a_1 \left(\sum_{k=1}^{n_y} v_{jk} y_{i-k} + \sum_{m=0}^{n_x-1} u_{jm} x_{i-m} + b_j \right) \\ &+ a_2 \left(\sum_{k=1}^{n_y} v_{jk} y_{i-k} + \sum_{m=0}^{n_x-1} u_{jm} x_{i-m} + b_j \right)^2 \\ &\vdots \\ &+ a_n \left(\sum_{k=1}^{n_y} v_{jk} y_{i-k} + \sum_{m=0}^{n_x-1} u_{jm} x_{i-m} + b_j \right)^n \Bigg]. \end{aligned} \tag{32}$$

To calculate $H_1(\Omega)$, this equation is expanded and taking only the relevant terms linear in x or y, gives

$$\begin{aligned} y_i = a_1 &\left(\sum_{j=1}^{n_h} w_j \sum_{k=1}^{n_y} v_{jk} y_{i-k} + \sum_{j=1}^{n_h} w_j \sum_{m=0}^{n_x-1} u_{jm} x_{i-m} \right) \\ &+ 2a_2 \left(\sum_{j=1}^{n_h} w_j \sum_{k=1}^{n_y} v_{jk} y_{i-k} b_j + \sum_{j=1}^{n_h} w_j \sum_{m=0}^{n_x-1} u_{jm} x_{i-m} b_j \right) \\ &\vdots \\ &+ n a_n \left(\sum_{j=1}^{n_h} w_j \sum_{k=1}^{n_y} v_{jk} y_{i-k} b_j^{n-1} + \sum_{j=1}^{n_h} w_j \sum_{m=0}^{n_x-1} u_{jm} x_{i-m} b_j^{n-1} \right). \end{aligned} \tag{33}$$

* The neural network software used bias elements b_j and the formulae reflect this. In fact, these are not needed because of the explicit inclusion of the term a_0 in the activation function.

Probing as before with equations (24) and (25), equating coefficients of $\mathrm{e}^{\mathrm{i}\Omega t}$ and rearranging gives,

$$H_1(\Omega) = \frac{\sum_{j=1}^{n_h} w_j(a_1 + 2a_2 b_j + 3a_3 b_j^2) \sum_{m=0}^{n_x - 1} u_{jm}\, \mathrm{e}^{-\mathrm{i}m\Omega\delta t}}{1 - \sum_{j=1}^{n-h} w_j(a_1 + 2a_2 b_j + 3a_3 b_j^2) \sum_{k=1}^{n_y} v_{jk}\, \mathrm{e}^{-\mathrm{i}k\Omega\delta t}}, \tag{34}$$

or for a general polynomial, defining

$$f(b_j) = \sum_{i=1}^{n_p} a_i b_j^i, \tag{35}$$

and replacing the tanh term of equation (27) with the derivative of the new activation function gives

$$H_1(\Omega) = \frac{\sum_{j=1}^{n_h} w_j f^{(1)}(b_j) \sum_{m=0}^{n_x - 1} u_{jm}\, \mathrm{e}^{-\mathrm{i}m\Omega\delta t}}{1 - \sum_{j=1}^{n-h} w_j f^{(1)}(b_j) \sum_{k=1}^{n_y} v_{jk}\, \mathrm{e}^{-\mathrm{i}k\Omega\delta t}}. \tag{36}$$

Following a similar method and probing with $x_i = \mathrm{e}^{\mathrm{i}\Omega_1 t} + \mathrm{e}^{\mathrm{i}\Omega_2 t}$ gives $H_2(\Omega_1, \Omega_2)$ up to third-order as,

$$H_2(\Omega_1, \Omega_2) = \frac{1}{2!D} \sum_{j=1}^{n_h} w_j \{(a_2 + 3a_3 b_j)(A_j + B_j + C_j)\}, \tag{37}$$

where

$$D = 1 - \sum_{j=1}^{n_h} w_j(a_1 + 2a_a b_j + 3a_3 b_j^2) \sum_{k=1}^{n_y} v_{jk}\, \mathrm{e}^{-\mathrm{i}k(\Omega_1 + \Omega_2)\delta t}$$

and A_j, B_j and C_j are defined by equation (30).

When implementing polynomial activation functions, two important decisions must be made: the first is what value should be given to the coefficients a_i; the second is what order of polynomial should be utilized? The latter is easily answered: to model the time data accurately, the order of the polynomial should be at least the same order as the non-linearity in the training data. In the following study, which will again employ the Duffing oscillator, the non-linearity order is three and so a third-order polynomial will be used. The first question is less easily answered. The choice of coefficient can have a crucial bearing on the network's ability to model the data. In this paper values of $a_1 = 1$, $a_2 = 0$ and $a_3 = -1/3$ will be employed. (This is a truncation of tanh.)

6. RADIAL BASIS FUNCTION NETWORKS

Much of the recent work on system identification has abandoned the MLP structure in favour of the Radial Basis Function networks introduced by Broomehead and Lowe [26]. The essential differences between the two approaches are in the computation of the hidden node activation and in the form of the non-linear activation function. At each hidden node in the MLP network, the activation z is obtained as a weighted sum of incoming signals from the input layer,

$$z_i = \sum_j w_{ij} x_j .$$

This is then passed through a non-linear activation function which is sigmoidal in shape, the important features of the function are its continuity, its monotonicity and its asymptotic approach to constant values. The resulting hidden node response is *global* in the sense that it can take non-zero values at all points in the space spanned by the network input vectors.

In contrast, the RBF network has *local* hidden nodes. The activation is obtained by taking the Euclidean distance squared from the input vector to a point defined independently for each hidden node—its centre c_i (which is of course a vector of the same dimension as the input layer).

$$z_i = \| x_i - c_i \| .$$

This is then passed through a *basis function* which decays rapidly with its argument i.e., it is significantly non-zero only for inputs close to c_i. The overall output of the RBF network is therefore the summed response from several locally-tuned units. It is this ability to cover selectively connected regions of the input space which makes the RBF so effective for pattern recognition and classification problems. The RBF structures also allows an effective means of implementing the NARX model for control and identification [27, 28].

For the calculation given here, a Gaussian basis function is assumed as this is by far the most commonly used to date. Also, following Poggio and Girosi [29], the network is modified by the inclusion of direct linear connections from the input layer to the output. The resulting NARX model is summarized by,

$$y_i = s + \sum_{j=1}^{n_h} w_j \exp\left\{ -\frac{1}{2\sigma_j^2}\left[\sum_{k=1}^{n_y} (y_{i-k} - v_{jk})^2 + \sum_{m=0}^{n_x-1} (x_{i-m} - u_{jm})^2 \right]\right\}$$
$$+ \underbrace{\sum_{j=1}^{ny} a_j y_{i-j} + \sum_{j=0}^{n_x-1} b_j x_{i-j},}_{\text{from linear connections}} \qquad (38)$$

where the quantities v_{jk} and u_{jm} are the hidden node centres and σ_i is the standard deviation or *radius* of the Gaussian at hidden node i. The first part of this expression is the standard RBF network.

As with the MLP network the appearance of constant terms in the exponent will lead to difficulties when this is expanded as a Taylor series. A trivial rearrangement yields the more useful form,

$$y_i = s + \sum_{j=1}^{n_h} w_j \gamma_j \exp\left\{ -\frac{1}{2\sigma_j^2}\left[\sum_{k=1}^{n_y} (y_{i-k}^2 - 2v_{jk} y_{i-k}) + \sum_{m=0}^{n_x-1} (x_{i-m}^2 - 2u_{jm} x_{i-m}) \right]\right\}$$
$$+ \sum_{j=1}^{ny} a_j y_{i-j} + \sum_{j=0}^{n_x-1} b_j x_{i-j}, \qquad (39)$$

where

$$\gamma_j = \exp\left\{ -\frac{1}{2\sigma_j^2}\left[\sum_{k=1}^{n_y} v_{jk}^2 + \sum_{m=0}^{n_x-1} u_{jm}^2 \right]\right\}. \tag{40}$$

Now, expanding the exponential and retaining only the linear terms leads to the required expression for obtaining H_1,

$$y_i = \sum_{j=1}^{n_h} \frac{w_j \gamma_j}{\sigma_j}\left\{ \sum_{k=1}^{n_y} v_{jk} y_{i-k} + \sum_{m=0}^{n_x-1} u_{jm} x_{i-m} \right\}. \tag{41}$$

Substituting the probing expressions for H_1 i.e., equations (20) and (21) yields,

$$H_1(\Omega) = \frac{\sum_{j=1}^{n_h} \gamma_j w_j \frac{1}{\sigma_j^2} \sum_{m=0}^{n_x-1} u_{jm}\, \mathrm{e}^{-\mathrm{i}\Omega m \delta t} + \sum_{j=0}^{n_x-1} b_j\, \mathrm{e}^{-\mathrm{i}\Omega j \delta t}}{1 - \sum_{j=1}^{n_y} a_j\, \mathrm{e}^{-\mathrm{i}\Omega j \delta t} - \sum_{j=1}^{n_h} \gamma_j w_j \frac{1}{\sigma_j^2} \sum_{k=1}^{n_y} v_{jk}\, \mathrm{e}^{-\mathrm{i}\Omega k \delta t}}. \tag{42}$$

The second-order FRF H_2 is obtained as described in the previous section.

$$\begin{aligned} H_2(\Omega_1, \Omega_2) = \frac{1}{2!D} \sum_{j=1}^{n_h} w_j \gamma_j \Bigg\{ & -\frac{1}{2\sigma_j^2} \sum_{k=0}^{n_x-1} \sum_{l=0}^{n_x-1} (\mathrm{e}^{-\mathrm{i}k\Omega_1\delta t}\, \mathrm{e}^{-\mathrm{i}l\Omega_2\delta t} + \mathrm{e}^{-\mathrm{i}k\Omega_2\delta t}\, \mathrm{e}^{-\mathrm{i}l\Omega_1\delta t}) \\ & + \frac{1}{\sigma_j^4} \sum_{k=0}^{n_x-1} \sum_{l=0}^{n_x-1} u_{jk} u_{jl} (\mathrm{e}^{-\mathrm{i}k\Omega_1\delta t}\, \mathrm{e}^{-\mathrm{i}l\Omega_2\delta t} + \mathrm{e}^{-\mathrm{i}k\Omega_2\delta t}\, \mathrm{e}^{-\mathrm{i}l\Omega_1\delta t}) \\ & - \frac{1}{2\sigma_j^2} \sum_{k=1}^{n_y} \sum_{l=1}^{n_y} H_1(\Omega_1) H_1(\Omega_2) (\mathrm{e}^{-\mathrm{i}k\Omega_1\delta t}\, \mathrm{e}^{-\mathrm{i}l\Omega_2\delta t} + \mathrm{e}^{-\mathrm{i}k\Omega_2\delta t}\, \mathrm{e}^{-\mathrm{i}l\Omega_1\delta t}) \\ & + \frac{1}{\sigma_j^4} \sum_{k=1}^{n_y} \sum_{l=1}^{n_y} v_{jk} v_{jl} H_1(\Omega_1) H_1(\Omega_2) (\mathrm{e}^{-\mathrm{i}k\Omega_1\delta t}\, \mathrm{e}^{-\mathrm{i}l\Omega_2\delta t} + \mathrm{e}^{-\mathrm{i}k\Omega_2\delta t}\, \mathrm{e}^{-\mathrm{i}l\Omega_1\delta t}) \\ & + \frac{1}{\sigma_j^4} \sum_{k=1}^{n_y} \sum_{l=0}^{n_x-1} v_{jk} u_{jl} (H_1(\Omega_1)\, \mathrm{e}^{-\mathrm{i}k\Omega_1\delta t}\, \mathrm{e}^{-\mathrm{i}l\Omega_2\delta t} + H_1(\Omega_2)\, \mathrm{e}^{-\mathrm{i}k\Omega_2\delta t}\, \mathrm{e}^{-\mathrm{i}l\Omega_1\delta t}) \Bigg\}, \end{aligned} \tag{43}$$

where

$$D = 1 - \sum_{j=1}^{n_y} a_j\, \mathrm{e}^{-\mathrm{i}(\Omega_1+\Omega_2) j \delta t} - \sum_{j=1}^{n_h} \gamma_j w_j \frac{1}{\sigma_j^2} \sum_{k=1}^{n_y} \mathrm{e}^{-\mathrm{i}(\Omega_1+\Omega_2) k \delta t}. \tag{44}$$

7. ILLUSTRATION OF THE THEORY

In order to demonstrate the use of the theory developed over previous sections, an application to system identification will be given. As mentioned in the introduction, arguably the fastest method of obtaining the HFRFs is to fit a discrete-time model; a successful application of this procedure to wave force analysis based on polynomial NARMAX models can be found in reference [8]. However, given the wide availability of neural network software, it would be advantageous to obtain HFRFs by probing of neural network NARX models. The object of this section is to examine this possibility.

7.1. GENERATION AND VALIDATION OF THE DATA

The asymmetric Duffing oscillator specified in equation (13) was chosen for the simulations with $m = 1$ kg, $c = 20$ Ns/m, $k = 10^4$ N/m, $k_2 = 10^7$ N/m^2 and $k_3 = 5 \times 10^9$ N/m^3. The differential equation of motion was stepped forward in time using a fourth-order Runge–Kutta scheme as given in reference [31]. The excitation $x(t)$ used was a white Gaussian sequence with zero mean and RMS of 0·5, hand-limited with the range 0–100 Hz. A time-step of 0·001 s was adopted and the data was decimated by a factor of 5, giving a final Δt of 0·005 s—corresponding to a sampling frequency of 200 Hz. A thousand points of sampled input and displacement data were taken for network training.

The data required validation before the results of applying the neural network procedure could be interpreted with confidence. This was accomplished by a two-step process. First, a polynomial NARMAX model was obtained for the data. An orthogonal least-squares estimator was used to fit the coefficients and the model terms were selected using the forward regression algorithm as described in reference [32]. The model obtained was,

$$\begin{aligned} y_i = 1{\cdot}8121 y_{i-1} \quad & - 1{\cdot}1429 y_{i-2} \\ + 0{\cdot}12924 y_{i-3} \quad & - 1{\cdot}9897 \times 10^2 y_{i-1}^2 \\ - 9{\cdot}6942 \times 10^4 y_{i-1}^3 \quad & + 3{\cdot}0145 \times 10^{-6} u_i \\ + 1{\cdot}7688 \times 10^{-5} u_{i-1}. \end{aligned} \tag{45}$$

The H_1 function was computed for the NARMAX model using harmonic probing and compared with the exact result obtained from equation (14); the comparison is shown in Figure 6 and the two functions show impressive agreement. In order to compare the H_2 functions, the values along the diagonal i.e., $H_2(\omega, \omega)$ were obtained for the model and compared with the true result. Figure 7 shows that this exercise demonstrated good

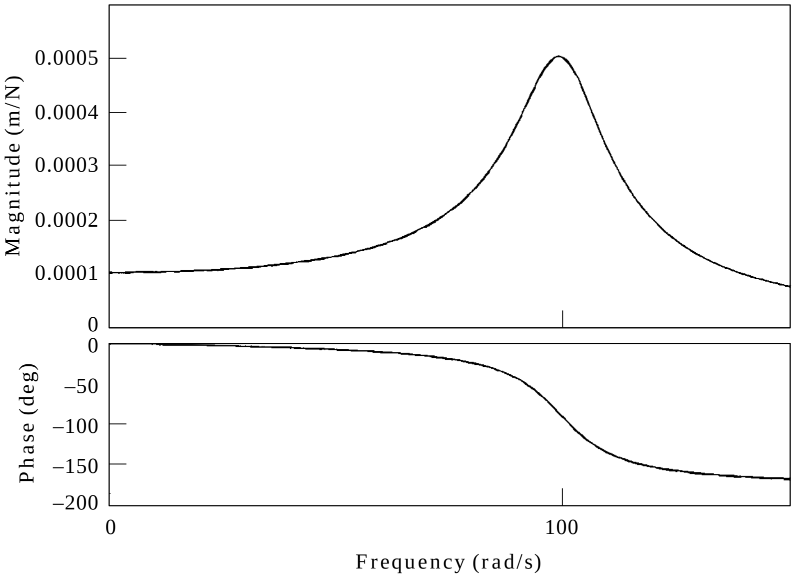

Figure 6. $H_1(\Omega)$ calculated using the NARMAX method (——) compared to theory (–––).

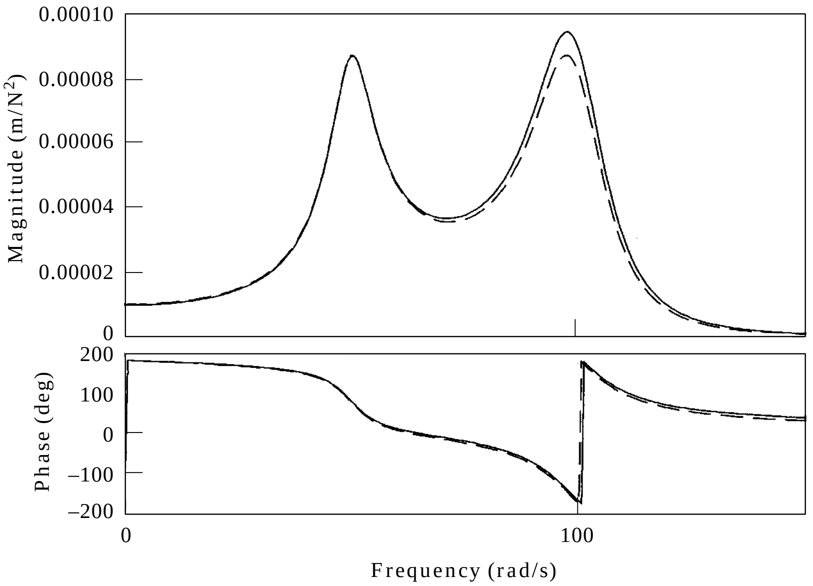

Figure 7. $H_2(\Omega_1, \Omega_2)$ calculated using the NARMAX method (——) compared to theory (– – –).

agreement between the functions. This serves to show that the data generated by integrating the equation of motion is truly characteristic of the system and that the true HFRFs (14) and (15) therefore serve as a reference for the neural network models fitted later.

7.2. MODEL VALIDITY

This is now the appropriate point to discuss how models are validated. There are several indicators of goodness-of-fit for NARX models in order of stringency these include: (1) One-Step-Ahead (OSA) prediction error, and (2) Model Predicted Output (MPO) error.

7.2.1. *One-step-ahead predictions*

Given the NARX representation of a system

$$y_i = f(y_{i-1}, \ldots, y_{i-n_y}; x_{i-1}, \ldots, x_{i-n_x}), \tag{46}$$

the one-step-ahead prediction of y_i is made using measured values for all past inputs *and* outputs.

$$\hat{y}_i = f(y_{i-1}, \ldots, y_{i-n_y}; x_{i-1}, \ldots, x_{i-n_x}). \tag{47}$$

The one-step-ahead series can then be compared to the measured outputs. Good agreement is clearly a necessary condition for model validity.

7.2.2. *Model predicted output*

In this case, the inputs are the only measured quantities used to generate the model output, i.e.,

$$\hat{y}_i = f(\hat{y}_{i-1}, \ldots, \hat{y}_{i-n_y}; x_{i-1}, \ldots, x_{i-n_x}). \tag{48}$$

In order to avoid a misleading transient at the start of the record for $\hat{y}$, the first n_y values of the measured output are used to start the recursion. As above, the estimated outputs must be compared with the measured outputs, with good agreement a necessary condition for accepting the model. This test is generally stronger than the previous one; in fact the one-step-ahead predictions can be excellent in some cases when the model predicted output shows complete disagreement with the measured data.

In order to have an objective measure of the closeness of two sequences of data, the normalized mean-square error (MSE) is introduced, the definition is

$$\mathrm{MSE}(\hat{y}) = \frac{100}{N\sigma_y^2} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2. \tag{49}$$

This MSE has the following useful property; if the mean of the output signal $\bar{y}$ is used as the model i.e., $\hat{y}_i = \bar{y}$ for all i, the MSE is 100·0, i.e.,

$$\mathrm{MSE}(\hat{y}) = \frac{100}{N\sigma_y^2} \sum_{i=1}^{N} (y_i - \bar{y})^2 = \frac{100}{\sigma_y^2} \cdot \sigma_y^2 = 100.$$

Experience shows that a MSE of less than 5·0 indicates good agreement while one of less than 1·0 reflects an excellent fit.

7.2.3. *Results from* tanh *activation function networks*

Using the time data generated from section 7.1, networks were trained and tested. A *C* program was written to train the networks for various momentum and learning coefficients, number of input and hidden layer units. This program then calculated the first- and second-order FRFs of the network.

Figure 8 shows the $H_1(\Omega)$ that best approximates the theoretical result. This almost perfect overlay was obtained from a network with 10 input units and two hidden units (i.e., 10:2:1), Momentum Coefficient (MC) of 0·1 and a Learning Coefficient (LC) of 0·35 and after 100,000 presentations of data from the training set, the network had converged to an MPO error of 0·35, shown compared to desired output in Figure 9.

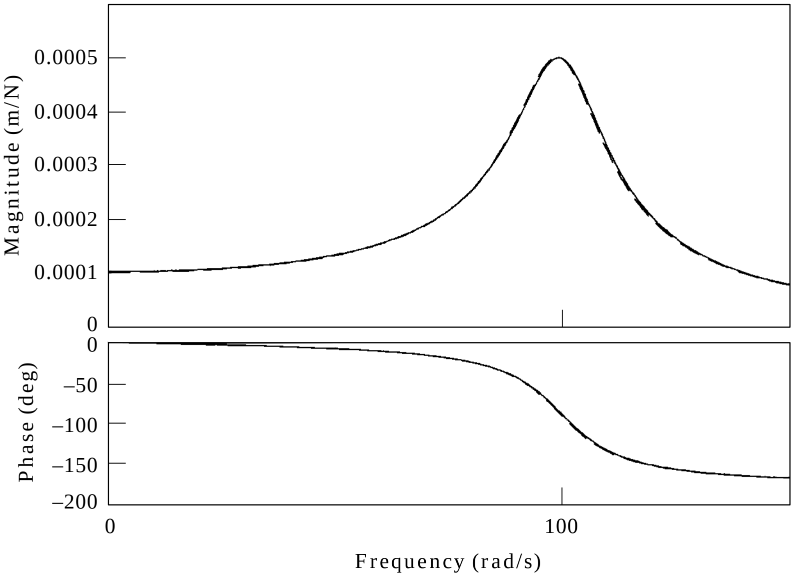

Figure 8. "Best fit" $H_1(\Omega)$ from the Duffing oscillator obtained by harmonic probing a 10:2:1 NARX network (——), compared to theory (–––).

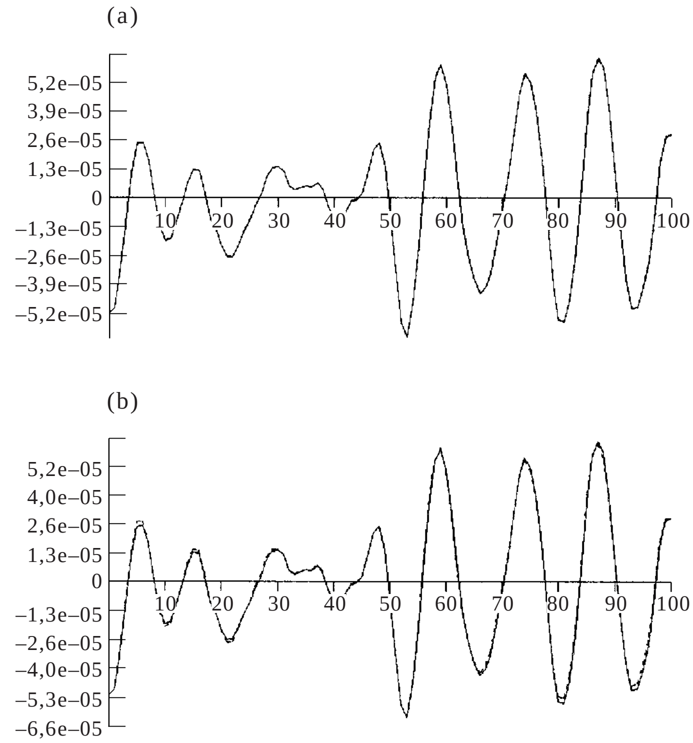

Figure 9. OSA prediction (a) and MPO (b) compared to the desired output from tanh network trained on the Duffing oscillator data. ——, Measured data; – – –, predicted data.

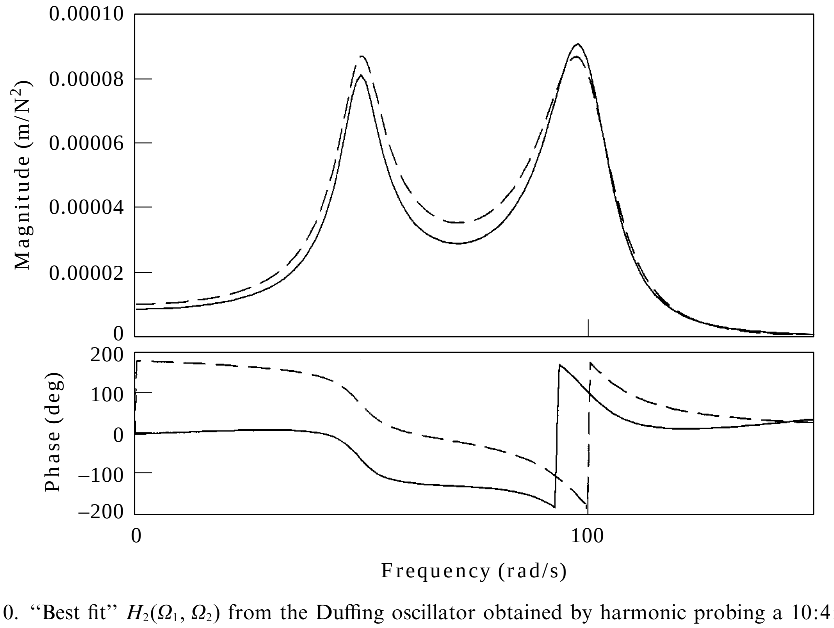

Figure 10. "Best fit" $H_2(\Omega_1, \Omega_2)$ from the Duffing oscillator obtained by harmonic probing a 10:4:1 NARX network (——), compared to theory (– – –).

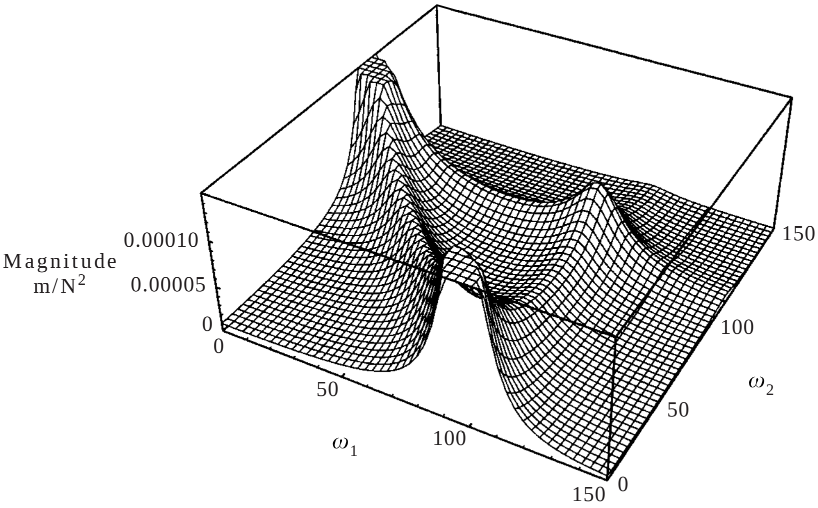

Figure 11. Duffing oscillator $H_2(\Omega_1, \Omega_2)$ surface for the 10:4:1 NARX network.

The second-order FRF proved a little more difficult to estimate accurately. The "best" $H_2(\Omega_1, \Omega_2)$ estimation is compared to the theoretical kernel transform in Figure 10 along its leading diagonal ($\Omega_1 = \Omega_2$). This was calculated from a 10:4:1 network with an MC of 0·15 and an LC of 0·2, trained to an MPO error of 0·27. The full $H_2(\Omega_1, \Omega_2)$ surface is shown in Figure 11, and by visual inspection it compares well to the theoretical surface of Figure 3. However, the corresponding $H_1(\Omega_1)$ from the same network, shown in Figure 12, shows some discrepancy from theory. The $H_2(\Omega_1, \Omega_2)$ calculated from the 10:2:1 network (that produced the near perfect $H_1(\Omega)$ result) is shown in Figure 13 and is greatly in error.

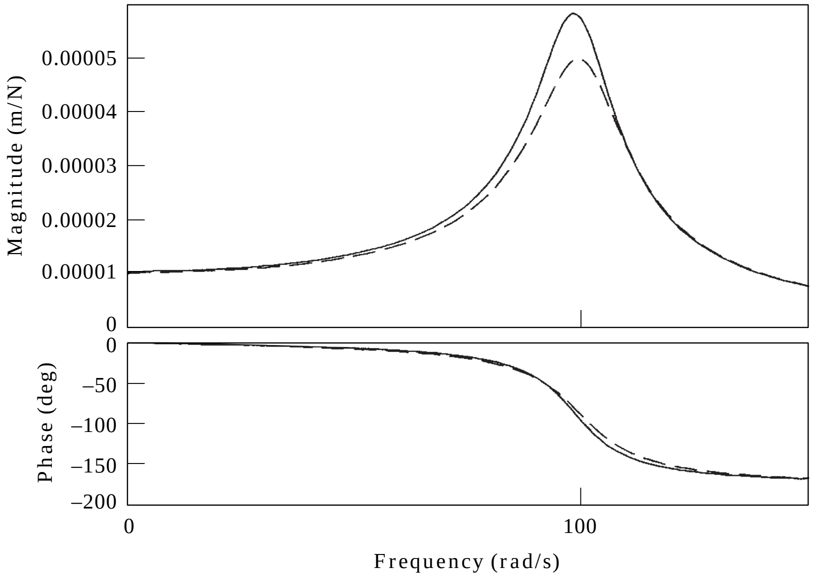

Figure 12. $H_1(\Omega)$ from the 10:4:1 network which produced the best $H_2(\Omega_1, \Omega_2)$ (——) compared to theory (– – –).

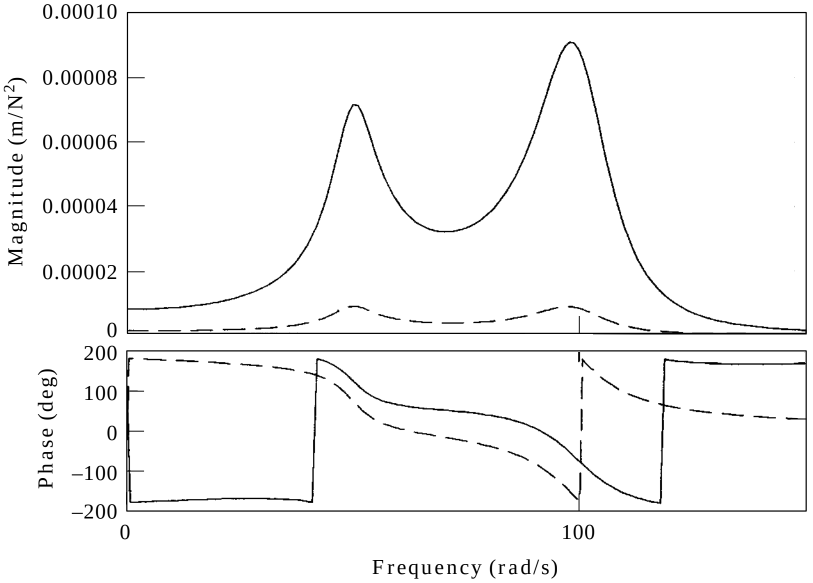

Figure 13. $H_2(\Omega_1, \Omega_2)$ from the 10:2:1 network that produced the best $H_1(\Omega)$ (——) compared to theory (– – –).

7.2.4. *Results from polynomial activation function networks*

The first-order FRF is compared to theory in Figure 14, obtained from a 10:2:1 network trained to 0·18 MPO error. The best $H_2(\Omega_1, \Omega_2)$, is shown in Figure 15. This was achieved using a 10:4:1 network with 0·5 error.

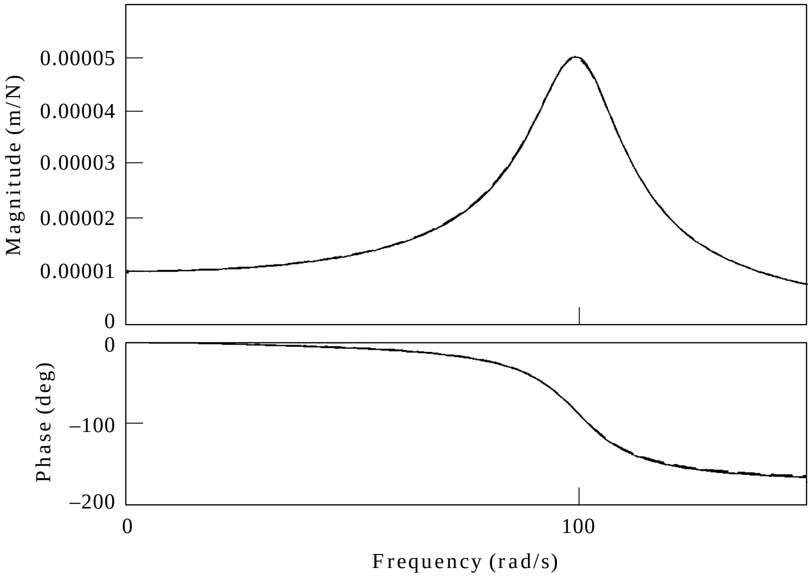

Figure 14. $H_1(\Omega)$ from the Duffing oscillator obtained by harmonic probing a 10:2:1 polynomial network (——), compared to theory (– – –). Polynomial coefficients: $a_1 = 1$, $a_2 = 0$, $a_3 = -1/3$.

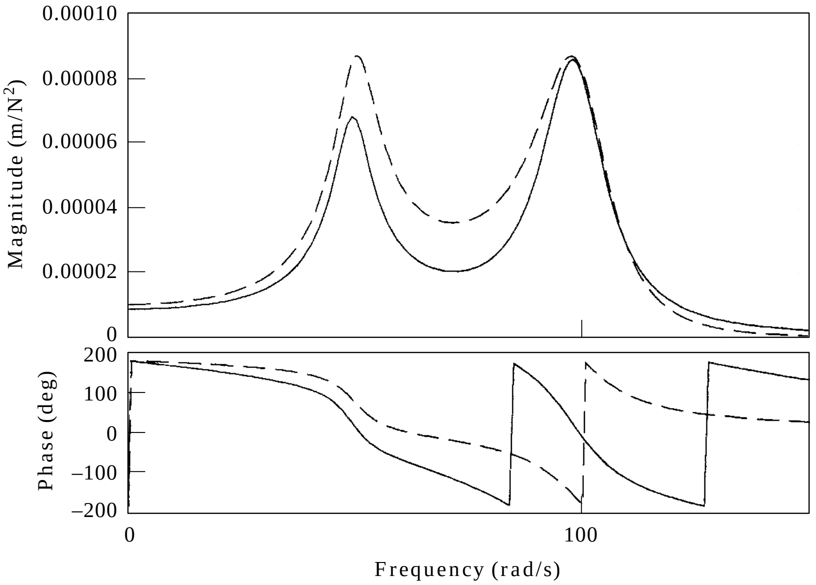

Figure 15. $H_2(\Omega_1, \Omega_2)$ from the Duffing oscillator obtained from a 10:4:1 polynomial network (——), compared to theory (– – –). Polynomial coefficients: $a_1 = 1$, $a_2 = 0$, $a_3 = -1/3$.

7.2.5. *Results from radial basis function networks*

The RBF networks were trained and tested using the displacement data of section 7.1. The usual spread of results was observed, the best $H_1(\Omega)$ being given by a 6:2:1 network, trained to 0·48 MPO error after 100,000 presentations, shown in Figure 16. A 4:4:1 network produced the best $H_2(\Omega_1, \Omega_2)$ after training to 0·72 MPO error. This is shown in Figure 17.

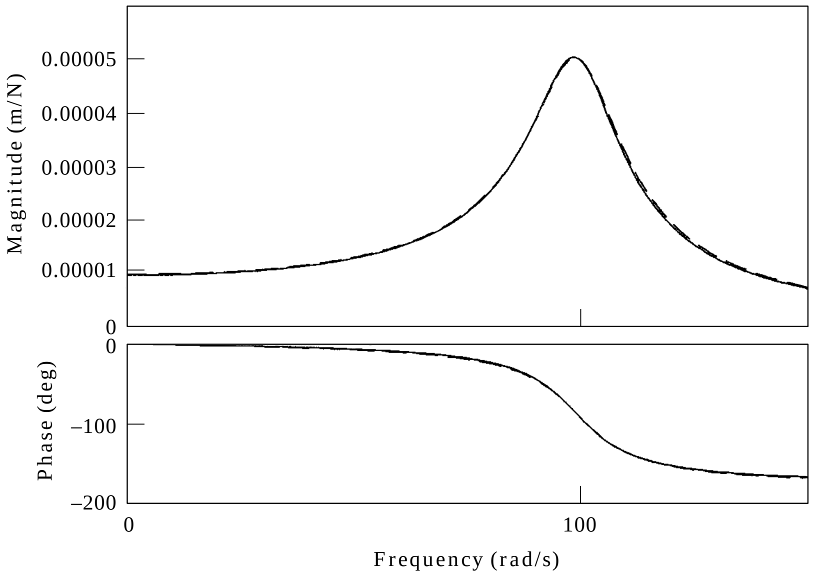

Figure 16. $H_1(\Omega)$ from the Duffing oscillator obtained from a 6:2:1 RBF network (——), compared to theory (– – –).

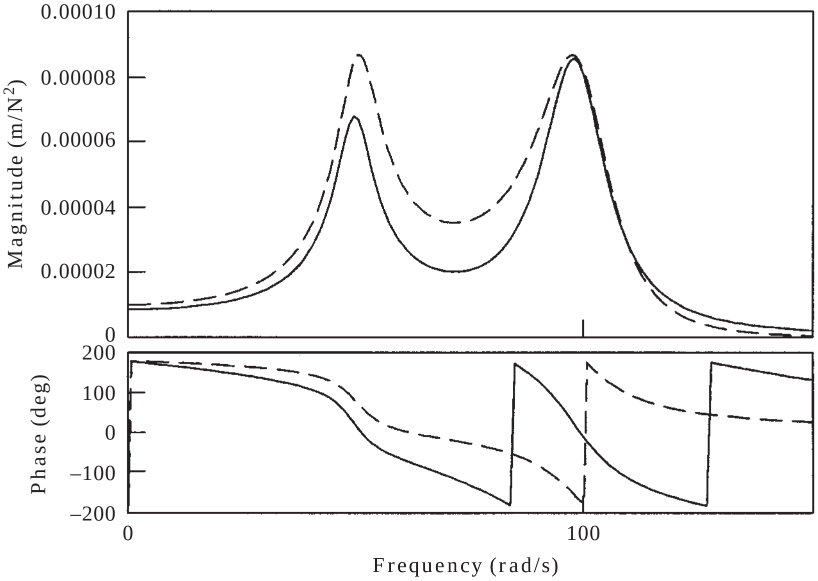

Figure 17. $H_2(\Omega_1, \Omega_2)$ from the Duffing oscillator obtained from a 4:4:1 RBF network (——), compared to theory (– – –).

8. DISCUSSION

There seems to be little connection between the MPO error and the quality of the results. The results indicate possible *over-parameterization* by the network in modelling the time data rather than accurately modelling the system in question. In system identification, over-parameterization often causes misleading results [34]. Over-parameterization is caused by the model having many more degrees of freedom than the system it is modelling. As the complexity of the network increases, its data modelling abilities generally improve, up to a point. Beyond that point the error on the training data continues to decrease but the error over a testing set begins to increase. Neural network users have long known this and *always* use a training and testing set for validation; this may be necessary in this approach to system identification. Often the MPO error is taken as conclusive.

In other terms, the network is finding local minima rather than the global minimum of the solution space. In fact better results were obtained from training a linear network (which does not suffer from local minima) with data from a linear system, here the network with the lowest MPO error gave the best $H_1(\Omega)$ fit. The NARMAX model of section 7.1 shows better agreement. This may be attributable to the fact that its parameter estimation algorithm does not suffer from local minima, and can also obtain a parsimonious model structure as opposed to the over-parameterized one of the neural network. The problem has long been considered in the case of NARMAX models, the orthogonal estimation algorithms used there together with the forward and backward selection algorithms offer a possible solution to the problem of over-parameterization [32]. Note that the MPO error for the NARMAX model was much lower than the network MPO error.

The problem of over-parameterization of neural networks has no clear-cut solution. One promising approach is based on *regularization* [35], a technique which prevents the network developing structure associated with noise in the data. In order to investigate this method, several model networks were trained with Gaussian noise added to the input patterns, a procedure equivalent to Tikhonov regularization [35]. Unfortunately, this had no visible effect. However, adding noise is known to give an inefficient implementation of regularization and further methods—like weight decay—will be explored in future work.

Another approach to the problem is based on *pruning*, where the network nodes and connections which contribute little or nothing to the network dynamics are removed during training. Established techniques like "Optimal Brain Surgery" are discussed in reference [35]; reference [36] proposes a promising method based on optimization techniques; this paper is also of interest because the networks used for illustration are NARX models of a non-linear system.

Because the application to identification is essentially a side-issue here, the ideas of regularization and pruning will be pursued in a later study.

9. CONCLUSIONS

The paper presents analytical expressions for the low-order (first to third) Volterra kernel transforms or HFRFs of the NARX neural network structure. An application is presented in which HFRFs are calculated from networks which model dynamical systems and thus provide approximations to the HFRFs of the original systems. In principle this method provides an attractive means of identifying HFRFs; however, before it can be applied, some means of quantifying the network model validity is required. The problem of over-parameterization must be eliminated.

ACKNOWLEDGMENTS

The authors would like to thank Dr Peter Cowley of Rolls Royce Applied Sciences Laboratories for a very useful discussion. The comments of Dr Gary Green, of the Medical School, University of Newcastle, on an early draft of the paper proved valuable. Dr Green is also to be thanked for making the results of his work available to the authors.

REFERENCES

1. S. A. Billings and S. Chen 1991 in *Neural Networks for Control and Systems*: *Principles and Applications*. Peregrinus Publ. Neural networks and system identification.
2. V. Volterra 1959 *Theory of Functionals and of Integral and Integro-Differential Equations*. New York: Dover.
3. P. Hearne 1995 *PhD Thesis*, *University of Newcastle upon Tyne*. An integrated approach to neuronal modelling.
4. V. Z. Marmarelis and X. Zhao 1994 in *Advanced Methods of System Modelling*, Volume 3, 243–259. New York: Plenum Press. On the relation between Volterra models and feedforward artificial neural networks.
5. M. J. Korenberg and I. W. Hunter 1990 *Annals of Biomedical Engineering* **18**, 629–654. The identification of nonlinear biological systems: Wiener kernel approaches.
6. J. Wray and G. G. R. Green 1994 *Biological Cybernetics* **71**, 187–195. Calculation of the Volterra kernels of nonlinear dynamic systems using an artificial neural network.
7. S. A. Billings and K. M. Tsang 1989 *Mechanical Systems and Signal Processing* **3**, 319–339. Spectral analysis for nonlinear systems, part I: parametric non-linear spectral analysis.
8. K. Worden, S. A. Billings, P. K. Stansby and G. R. Tomlinson 1994 *Journal of Fluids and Structures* **8**, 18–71. Identification of nonlinear wave forces.
9. D. Lowe and C. M. Bishop 1995 *Talk presented at British Aerospace Workshop on the Dependability of Neural Networks*, *Sowerby Research Centre*, *Bristol*. Some theoretical background on neural networks.
10. M. Schetzen 1980 *The Volterra and Wiener Theories of Nonlinear Systems*. New York: John Wiley Interscience.

11. G. Palm and T. Poggio 1977 *SIAM Journal on Applied Mathematics* **33**, part 2, 195–216. The Volterra representation and the Wiener expansion: validity and pitfalls.
12. D. D. Wiener and J. F. Spina 1980 *Sinusoidal Analysis and Modeling of Weakly Nonlinear Circuits With Applications to Nonlinear Interference Effects*. Princeton, NJ: Van Nostrand Reinhold.
13. J. C. Peyton Jones and S. A. Billings 1990 *International Journal of Control* **52**, 319–346. Interpretation of non-linear frequency response functions.
14. N. Wiener *Report no*. 129, *Radiation Laboratory*, *M.I.T.*, *Cambridge*, *MA*. (Also published as U.S. Department of Commerce Publications PB-58087.) Response of a nonlinear device to noise.
15. Y. W. Lee and M. Schetzen 1965 *International Journal of Control* **2**, 237–254. Measurement of the Wiener kernels of a nonlinear system by cross-correlation.
16. E. Bedrosian and S. O. Rice 1971 *Proceedings IEEE* **59**, 1688–1707. The output properties of Volterra systems driven by harmonic and Gaussian inputs.
17. S. Chen and S. A. Billings 1989 *International Journal of Control* **49**, 1013–1032. Representations of non-linear systems: the NARMAX model.
18. S. A. Billings, H. B. Jamaluddin and S. Chen 1992 *International Journal of Control* **55**, 193–224. Properties of neural networks with applications to modelling non-linear dynamical systems.
19. I. J. Leontaritis and S. A. Billings 1985 *International Journal of Control* **41**, 303–328. Input–output parametric models for nonlinear systems. Part I: deterministic nonlinear systems.
20. G. Cybenko 1989 *Mathematics of Control*, *Signals and Systems* **2**, 303–314. Approximation by superpositions of sigmoidal functions.
21. K. Funahashi 1989 *Neural Networks* **2**, 183–192. On the approximate realisation of continuous mapping by neural networks.
22. H. T. Siegelman, B. G. Horne and C. L. Giles 1995 *Technical Report UMIACS-TR*-95-12 *and CS-TR*-3408, *Institute for Advanced Computer Studies*, *University of Maryland*, *College Park*. Computational capabilities of recurrent NARX neural networks.
23. H. T. Siegelman and E. D. Sontag 1991 *Applied Mathematics Letters* **4**, 77–80. Turing computability with neural nets.
24. H. T. Siegelman and E. D. Sontag 1992 in *Proceedings of the 5th ACM Workshop on Computational Learning Theory*, 440–449. ACM Press. On the computational power of neural networks.
25. S. A. Billings and H. Zhang 1995 *Mechanical Systems and Signal Processing* **9**, 537–553. Computation of non-linear transfer functions when constant terms are present.
26. D. S. Broomehead and D. Lowe 1988 *Complex Systems* **2**, 321–355. Multivariable functional interpolation and adaptive networks.
27. S. Chen, S. A. Billings, C. F. N. Cowan and P. M. Grant 1990 *International Journal of Control* **52**, 1327–1350. Practical identification of NARMAX models using radial basis functions.
28. S. Chen, S. A. Billings, C. F. N. Cowan and P. M. Grant 1990 *International Journal of Systems Science* **21**, 2513–2539. Non-linear systems identification using radial basis functions.
29. T. Poggio and F. Girosi 1990 *Proceedings of IEEE* **78**, 1481–1497. Network for approximation and learning.
30. J. Wray and G. G. R. Green 1991 in *Proceedings of IEE Control* 91, Volume 1, Conference Publication Number 332, 261–265. Analysis of networks that have learnt control problems.
31. W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling 1986 *Numerical Recipes—The Art of Scientific Computing*. Cambridge: Cambridge University Press.
32. S. Chen, S. A. Billings and W. Luo 1980 *International Journal of Control* **50**, 1873–1896. Orthogonal least-squares methods and their application to non-linear system identification.
33. S. A. Billings, S. Chen and R. J. Backhouse 1989 *Journal of Mechanical Systems and Signal Processing* **3**, 123–142. Identification of linear and nonlinear models of a turbocharged automotive diesel engine.
34. T. Söderström and P. Stoica 1989 *System Identification*. Englewood Cliffs, NJ: Prentice-Hall.
35. C. M. Bishop 1995 *Neural Networks for Pattern Recognition*. Oxford: Oxford University Press.
36. S. W. Stepniewski and A. J. Keane 1996 *Neural Computing and Applications*, submitted. Pruning backpropagation neural networks using modern stochastic optimisation techniques.

APPENDIX: DERIVATION OF $H_3(\Omega_1, \Omega_2, \Omega_3)$ FOR tanh NETWORK

Starting with the network equation

$$y_i = s + \sum_{j=1}^{n_h} w_j \sum_{t=0}^{\infty} \left\{ \frac{\tanh^{(t)}(b_j)}{t!} \left(\sum_{k=1}^{n_y} v_{jk} y_{i-k} + \sum_{m=0}^{n_x - 1} u_{jm} x_{i-m} \right)^{t} \right\}, \tag{A1}$$

and taking the $t = 3$ term gives the equation,

$$H_3(\Omega_1, \Omega_2, \Omega_3) = \frac{1}{3!D} Y^{(2)}(\Omega_1, \Omega_2, \Omega_3) + Y^{(3)}(\Omega_1, \Omega_2, \Omega_3), \tag{A2}$$

where

$$D = 1 - \sum_{j=1}^{n_h} \rho_j \sum_{k=1}^{n_y} v_{jk} \, \mathrm{e}^{-\mathrm{i}k(\Omega_1 + \Omega_2 + \Omega_3)\delta t}.$$

A.1. $Y^{(2)}(\Omega_1, \Omega_2, \Omega_3)$ TERM

$$Y^{(2)}(\Omega_1, \Omega_2, \Omega_3) = \sum_{j=1}^{n_h} \rho_j^{(2)} (A_j + B_j), \tag{A3}$$

where

$$\rho_j^{(2)} = w_j \tanh^{(2)} b_j$$

$$\begin{aligned} A_j = \sum_{k=1}^{n_y} \sum_{l=1}^{n_y} v_{jk} v_{jl} & 2H_1(\Omega_1) H_2(\Omega_2, \Omega_3) (\mathrm{e}^{-\mathrm{i}k\Omega_1\delta t} \, \mathrm{e}^{-\mathrm{i}l(\Omega_2 + \Omega_3)\delta t} + \mathrm{e}^{-\mathrm{i}l\Omega_1\delta t} \, \mathrm{e}^{-\mathrm{i}k(\Omega_2 + \Omega_3)\delta t}) \\ & + 2H_1(\Omega_2) H_2(\Omega_1, \Omega_3) (\mathrm{e}^{-\mathrm{i}k\Omega_2\delta t} \, \mathrm{e}^{-\mathrm{i}l(\Omega_1 + \Omega_3)\delta t} + \mathrm{e}^{-\mathrm{i}l\Omega_2\delta t} \, \mathrm{e}^{-\mathrm{i}k(\Omega_2 + \Omega_3)\delta t}) \\ & + 2H_1(\Omega_3) H_2(\Omega_1, \Omega_2) (\mathrm{e}^{-\mathrm{i}k\Omega_3\delta t} \, \mathrm{e}^{-\mathrm{i}l(\Omega_1 + \Omega_2)\delta t} + \mathrm{e}^{-\mathrm{i}l\Omega_3\delta t} \, \mathrm{e}^{-\mathrm{i}k(\Omega_1 + \Omega_2)\delta t}), \end{aligned}$$

$$\begin{aligned} B_j = \sum_{k=1}^{n_y} \sum_{l=0}^{n_x - 1} v_{jk} u_{jl} & 2H_2(\Omega_2, \Omega_3) \, \mathrm{e}^{-\mathrm{i}l\Omega_1\delta t} \, \mathrm{e}^{-\mathrm{i}k(\Omega_2 + \Omega_3)\delta t} \\ & + 2H_2(\Omega_1, \Omega_3) \, \mathrm{e}^{-\mathrm{i}l\Omega_2\delta t} \, \mathrm{e}^{-\mathrm{i}k(\Omega_1 + \Omega_3)\delta t} + 2H_2(\Omega_1, \Omega_2) \, \mathrm{e}^{-\mathrm{i}l\Omega_3\delta t} \, \mathrm{e}^{-\mathrm{i}k(\Omega_1 + \Omega_2)\delta t}). \end{aligned}$$

A.2. $Y^{(3)}(\Omega_1, \Omega_2, \Omega_3)$ TERM

$$y_i = s + \sum_{j=1}^{n_h} w_j \frac{\tanh^{(3)}(b_j)}{3!} \left(\sum_{k=1}^{n_y} v_{jk} y_{i-k} + \sum_{m=0}^{n_x - 1} u_{jm} x_{i-m} \right)^{3}, \tag{A4}$$

which when expanded gives (ignoring the constant s),

$$\begin{aligned} y_i = \sum_{j=1}^{n_h} w_j \left\{ \frac{\tanh^{(3)}(b_j)}{3!} \right. & \Bigg\{ \left(\sum_{k=1}^{n_y} v_{jk} y_{i-k} \right)^{3} + \left(\sum_{l=0}^{n_1 - 1} u_{jl} x_{i-l} \right)^{3} \\ & + 3 \left(\sum_{l=0}^{n_x - 1} u_{jl} x_{i-l} \right)^{2} \left(\sum_{k-1}^{n_y} v_{jk} y_{i-k} \right) + 3 \left(\sum_{l=0}^{n_x - 1} u_{jl} x_{i-l} \right) \left(\sum_{k=1}^{n_y} v_{jk} y_{i-k} \right)^{2} \Bigg\}. \end{aligned}$$

And this gives,

$$Y^{(2)}(\Omega_1, \Omega_2, \Omega_3) = \sum_{j=1}^{n_h} \beta_{1j}\{A_j + B_j + C_j + D_j\}\, \mathrm{e}^{\mathrm{i}(\Omega_1 + \Omega_2 + \Omega_3)t}, \tag{A5}$$

$$\rho_j = w_j \tanh^{(1)} b_j,$$

$$\beta_{1j} = \frac{1}{3!} w_j \tanh^{(3)} b_j.$$

It can be shown that,

$$\begin{aligned} A_j = \sum_{k=1}^{n_y} \sum_{l=1}^{n_y} \sum_{m=1}^{n_y} v_{jk} v_{jl} v_{jm} \{& H_1(\Omega_1) H_1(\Omega_2) H1(\Omega_3) \\ & (\mathrm{e}^{-\mathrm{i}k\Omega_1\delta t}\, \mathrm{e}^{-\mathrm{i}l\Omega_2\delta t}\, \mathrm{e}^{-\mathrm{i}m\Omega_3\delta t} + \mathrm{e}^{-\mathrm{i}k\Omega_1\delta t}\, \mathrm{e}^{-\mathrm{i}m\Omega_2\delta t}\, \mathrm{e}^{-\mathrm{i}l\Omega_3\delta t} \\ & \mathrm{e}^{-\mathrm{i}l\Omega_1\delta t}\, \mathrm{e}^{-\mathrm{i}k\Omega_2\delta t}\, \mathrm{e}^{-\mathrm{i}m\Omega_3\delta t} + \mathrm{e}^{-\mathrm{i}m\Omega_1\delta t}\, \mathrm{e}^{-\mathrm{i}k\Omega_2\delta t}\, \mathrm{e}^{-\mathrm{i}l\Omega_3\delta t} \\ & \mathrm{e}^{-\mathrm{i}l\Omega_1\delta t}\, \mathrm{e}^{-\mathrm{i}m\Omega_2\delta t}\, \mathrm{e}^{-\mathrm{i}k\Omega_3\delta t} + \mathrm{e}^{-\mathrm{i}m\Omega_1\delta t}\, \mathrm{e}^{-\mathrm{i}l\Omega_2\delta t}\, \mathrm{e}^{-\mathrm{i}k\Omega_3\delta t})\}, \end{aligned} \tag{A6}$$

$$\begin{aligned} B_j = \sum_{k=0}^{n_x-1} \sum_{l=0}^{n_x-1} \sum_{m=0}^{n_x-1} u_{jk} u_{jl} u_{jm} \{& H_1(\Omega_1) H_1(\Omega_2) H1(\Omega_3) \\ & (\mathrm{e}^{-\mathrm{i}k\Omega_1\delta t}\, \mathrm{e}^{-\mathrm{i}l\Omega_2\delta t}\, \mathrm{e}^{-\mathrm{i}m\Omega_3\delta t} + \mathrm{e}^{-\mathrm{i}k\Omega_1\delta t}\, \mathrm{e}^{-\mathrm{i}m\Omega_2\delta t}\, \mathrm{e}^{-\mathrm{i}l\Omega_3\delta t} \\ & \mathrm{e}^{-\mathrm{i}l\Omega_1\delta t}\, \mathrm{e}^{-\mathrm{i}k\Omega_2\delta t}\, \mathrm{e}^{-\mathrm{i}m\Omega_3\delta t} + \mathrm{e}^{-\mathrm{i}m\Omega_1\delta t}\, \mathrm{e}^{-\mathrm{i}k\Omega_2\delta t}\, \mathrm{e}^{-\mathrm{i}l\Omega_3\delta t} \\ & \mathrm{e}^{-\mathrm{i}l\Omega_1\delta t}\, \mathrm{e}^{-\mathrm{i}m\Omega_2\delta t}\, \mathrm{e}^{-\mathrm{i}k\Omega_3\delta t} + \mathrm{e}^{-\mathrm{i}m\Omega_1\delta t}\, \mathrm{e}^{-\mathrm{i}l\Omega_2\delta t}\, \mathrm{e}^{-\mathrm{i}k\Omega_3\delta t})\}, \end{aligned} \tag{A7}$$

$$\begin{aligned} C_j = 3 \times & \sum_{k=0}^{n_x-1} \sum_{l=0}^{n_x-1} \sum_{m=1}^{n_y} u_{jk} u_{jl} v_{jm} \\ & \{H_1(\Omega_3)\, \mathrm{e}^{-\mathrm{i}k\Omega_1\delta t}\, \mathrm{e}^{-\mathrm{i}l\Omega_2\delta t}\, \mathrm{e}^{-\mathrm{i}m\Omega_3\delta t} + H_1(\Omega_2)\, \mathrm{e}^{-\mathrm{i}k\Omega_1\delta t}\, \mathrm{e}^{-\mathrm{i}m\Omega_2\delta t}\, \mathrm{e}^{-\mathrm{i}l\Omega_3\delta t} \\ & + H_1(\Omega_3)\, \mathrm{e}^{-\mathrm{i}l\Omega_1\delta t}\, \mathrm{e}^{-\mathrm{i}k\Omega_2\delta t}\, \mathrm{e}^{-\mathrm{i}m\Omega_3\delta t} + H_1(\Omega_1)\, \mathrm{e}^{-\mathrm{i}m\Omega_1\delta t}\, \mathrm{e}^{-\mathrm{i}k\Omega_2\delta t}\, \mathrm{e}^{-\mathrm{i}l\Omega_3\delta t} \\ & + H_1(\Omega_2)\, \mathrm{e}^{-\mathrm{i}l\Omega_1\delta t}\, \mathrm{e}^{-\mathrm{i}m\Omega_2\delta t}\, \mathrm{e}^{-\mathrm{i}k\Omega_3\delta t} + H_1(\Omega_1)\, \mathrm{e}^{-\mathrm{i}m\Omega_1\delta t}\, \mathrm{e}^{-\mathrm{i}l\Omega_2\delta t}\, \mathrm{e}^{-\mathrm{i}k\Omega_3\delta t}\}, \end{aligned} \tag{A8}$$

$$\begin{aligned} D_j = 3 \times & \sum_{k=0}^{n_x-1} \sum_{l=1}^{n_y} \sum_{m=1}^{n_y} u_{jk} v_{jl} v_{jm} \\ & \{H_1(\Omega_2) H_1(\Omega_3)\, \mathrm{e}^{-\mathrm{i}k\Omega_1\delta t}\, \mathrm{e}^{-\mathrm{i}l\Omega_2\delta t}\, \mathrm{e}^{-\mathrm{i}m\Omega_3\delta t} + H_1(\Omega_2) H_1(\Omega_3)\, \mathrm{e}^{-\mathrm{i}k\Omega_1\delta t}\, \mathrm{e}^{-\mathrm{i}m\Omega_2\delta t}\, \mathrm{e}^{-\mathrm{i}l\Omega_3\delta t} \\ & + H_1(\Omega_1) H_1(\Omega_3)\, \mathrm{e}^{-\mathrm{i}l\Omega_1\delta t}\, \mathrm{e}^{-\mathrm{i}k\Omega_2\delta t}\, \mathrm{e}^{-\mathrm{i}m\Omega_3\delta t} + H_1(\Omega_1) H_1(\Omega_3)\, \mathrm{e}^{-\mathrm{i}m\Omega_1\delta t}\, \mathrm{e}^{-\mathrm{i}k\Omega_2\delta t}\, \mathrm{e}^{-\mathrm{i}l\Omega_3\delta t} \\ & + H_1(\Omega_1) H_1(\Omega_2)\, \mathrm{e}^{-\mathrm{i}l\Omega_1\delta t}\, \mathrm{e}^{-\mathrm{i}m\Omega_2\delta t}\, \mathrm{e}^{-\mathrm{i}k\Omega_3\delta t} + H_1(\Omega_1) H_1(\Omega_2)\, \mathrm{e}^{-\mathrm{i}m\Omega_1\delta t}\, \mathrm{e}^{-\mathrm{i}l\Omega_2\delta t}\, \mathrm{e}^{-\mathrm{i}k\Omega_3\delta t}\}. \end{aligned} \tag{A9}$$