Lattice Dynamical Wavelet Neural Networks Implemented Using Particle Swarm Optimisation for Spatio-Temporal System Identification

H. L. Wei, S. A. Billings, Y. F. Zhao and L. Z. Guo

Research Report No. 959

Department of Automatic Control and Systems Engineering
The University of Sheffield
Mappin Street, Sheffield,
S1 3JD, UK

July 2007

Lattice Dynamical Wavelet Neural Networks Implemented Using Particle Swarm Optimisation for Spatio-Temporal System Identification

H. L. Wei, S. A. Billings, Y. F. Zhao and L. Z. Guo

Department of Automatic Control and Systems Engineering
The University of Sheffield
Mappin Street, Sheffield
S1 3JD, UK

s.billings@shef.ac.uk, w.hualiang@shef.ac.uk

Abstract: Starting from the basic concept of coupled map lattices, a new family of adaptive wavelet neural networks, called lattice dynamical wavelet neural networks (LDWNN), is introduced for spatio-temporal system identification, by combining an efficient wavelet representation with a coupled map lattice model. A new orthogonal projection pursuit (OPP) method, coupled with a particle swarm optimisation (PSO) algorithm, is proposed for augmenting the proposed network. A novel two-stage hybrid training scheme is developed for constructing a parsimonious network model. In the first stage, by applying the orthogonal projection pursuit algorithm, significant wavelet-neurons are adaptively and successively recruited into the network, where adjustable parameters of the associated wavelet-neurons are optimised using a particle swarm optimiser. The resultant network model, obtained in the first stage, may however be redundant. In the second stage, an orthogonal least squares (OLS) algorithm is then applied to refine and improve the initially trained network by removing redundant wavelet-neurons from the network. The proposed two-stage hybrid training procedure can generally produce a parsimonious network model, where a ranked list of wavelet-neurons, according to the capability of each neuron to represent the total variance in the system output signal is produced. Two spatio-temporal system identification examples are presented to demonstrate the performance of the proposed new modelling framework.

Keywords: Coupled map lattices, evolutionary algorithms, lattice dynamical systems, neural networks, orthogonal least squares, parameter estimation, particle swarm optimisation, spatio-temporal images, wavelets.

1. Introduction

Spatio-temporal systems are complex systems where the system states evolve spatially as well as temporally. Unlike classical control systems where the current output is a function of previous inputs and outputs only in time, the output of a spatio-temporal system depends not only on past values in time but also values at different spatial locations. Spatio-temporal phenomena are widely found in biology, chemistry, ecology, geography, medicine, physics, and sociology (Kaneko 1993, Jahne 1993, Silva and Principe 1997, Astic et al. 1998, Bascompte and Sole 1998, Czaran 1998, Spors and Grinvald 2002, Dimitrova and Berezney 2002, Berezney et al. 2005, Dolak and Schmeiser 2005). In order to analyse, control or predict the dynamics of spatio-temporal systems, several efficient mathematical representations, including the well known cellular automata (CA) (Wolfram 1994), coupled map lattices (CML's) (Kaneko 1989, 1993), and cellular neural networks (CNN's) (Chua and Yang 1988a, 1988b, Chua and Roska 2001), have been proposed in the past decades. These lattice dynamical systems (LDS's) (Chow and Mallet-Paret 1995) have successfully been applied in various areas of science and engineering, see for example Albano et al. (1995), Raabe (2002), Ohtaki et al. (2002), Aydogan et al. (2005) and the references therein.

Whilst the forward problem of spatio-temporal systems has been extensively studied in the literature, with an assumption that the associated models are known and are used to describe some specific dynamics, the inverse problem, which is concerned with finding models based on given observations for structure-unknown spatio-temporal images, has received relatively little attention and relatively few results have been achieved. Identification plays an important role for solving the inverse problem relative to spatio-temporal systems, where the structure of analytical models is not available. Recently, efforts have been made to seek to solve the identification problem of spatio-temporal systems, and several efficient identification methods and algorithms have been proposed; these include local state reconstruction and partitioned filtering methods (Parlitz and Merkwirth 2000, Sitz et al. 2003), radial basis function neural networks (Leung et al. 2000), statistical methods (Mandelj et al. 2001, Xia and Leung 2005), polynomial models (Coca and Billings 2001, Billings and Coca 2002, Billings and Yang, 2003), wavelet models (Guo and Billings 2004, Billings et al. 2005), and other approaches (Marcos-Nikolaus and Martin-Gonzalez 2002, Chen and Ji 2005).

The key point in the identification of a spatio-temporal system, where the true evolution law is totally unknown and where only observed data are available, is to exploit some effective elementary building blocks, which can be used to construct efficient nonlinear lattice dynamical models that sufficiently reveal and depict the underlying dynamics of the system. Wavelet transforms (Daubechies 1992), due to their inherent property and excellent capability for the time-frequency domain representation of arbitrary signals, may be one of the best candidates to form the most powerful elementary building blocks. Wavelets have now been applied in almost all areas of science and engineering in recent years. A popular representation form among the existing wavelet models, for

dynamical systems, is the wavelet neural network (WNN). The primary motivation of combining well defined wavelets with conventional neural networks is to construct powerful wavelet based modelling frameworks (Zhang and Benveniste 1992, Bakshi and Stephanopoulos 1993, Pati and Krishnaprasad 1993), by exploiting the theoretical rigor of wavelets and the adaptive learning capability of conventional neural networks (Delyon et al. 1995, Juditsky et al. 1995, Unser 1996). Indeed, the introduction of wavelet network modelling frameworks have not only significantly enriched the storehouse of existing artificial neural networks and related model classes, but also more importantly enhanced the capability of conventional neural networks for nonlinear signal representation (Zhang and Benveniste 1992, Lin and Chin 2004). Wavelet network models provide powerful alternatives, to traditional artificial neural networks, for function learning (Zhang et al. 1995, Rying et al. 2002), dynamical modelling (Zhang 1997, Coca and Billings 1997, Oussar et al. 1998, Billings and Coca 1999, Wei and Billings 2004a, 2004b, Billings and Wei 2005a, Chen et al. 2006, Huang and Su 2007) and systems control (Sanner and Slotine 1998, Liu 2001, Wai and Chang 2003, Ho et al. 2005, Lin et al. 2006, Hsu et al. 2006, Xu and Tan 2007).

The central objective of this study is to introduce a new family of adaptive wavelet neural networks, where wavelet transforms will be incorporated into a specific type of CML model. This wavelet-based coupled map lattice model will be referred to as the lattice dynamical wavelet neural network (LDWNN). The construction procedure of the new network model is composed of two stages. At the first stage, linear combinations of a number of wavelet functions are chosen as the building blocks to form the initial candidate wavelet neurons. A variation of the conventional projection pursuit regression (PPR) method, called the orthogonal projection pursuit (OPP), implemented by a particle swarm optimisation (PSO) algorithm, is used to augment the network by recruiting a number of optimised wavelet neurons in a stepwise manner. Compared with other nonlinear least squares algorithms, including the back-propagation and Gaussian-Newton algorithms, the PSO algorithm, as a population-based evolutionary method, possesses several desirable attractive properties, for example, this kind of algorithm is easy to implement but quite efficient in dealing with a wide class of nonlinear optimisation problems (Eberhart and Kennedy 1995, Kennedy and Eberhart 1995). As a stochastic algorithm, PSO does not need any information on the gradients of the relevant object functions, this ensures that the PSO is very suitable for nonlinear optimisation problems where the relevant object functions are not differentiable or the gradients are computationally expensive or very difficult to obtain (Kennedy et al. 2001, van den Bergh 2002). The OPP learning algorithm, similar to the conventional projection pursuit regression (Friedman and Stuetzle 1981), may produce a redundant model. Thus, the objective of the second stage is to remove redundancy from the initially trained network, to produce a parsimonious representation. To achieve this aim, an orthogonal least squares (OLS) learning algorithm (Billings 1989, Chen et al. 1989, 1991) is applied to refine and improve the initially obtained network by removing potential redundant wavelet neurons from the network.

As will be noted from the proposed learning algorithm, the training procedure for the new network

model does not need any pre-specified dictionary, as required by existing wavelet-based CML models (Guo and Billings 2004, Billings et al. 2005). Also, as will be seen later from the illustrative examples, by combining the PSO based nonlinear OPP training scheme with an effective linear forward orthogonal regression algorithm, the resultant wavelet network model can provide good generalisation performance for a wide range of dynamical nonlinear modelling problems. Moreover, one feature of the new wavelet network, produced by the above two-stage hybrid learning algorithm, is that now the resultant model is transparent to model users; involved wavelet neurons are ranked according to the capability of each neuron in representing the total variance in the system output signal. This is desirable for many application cases where physical insight on the individual variables and associated wavelet neurons are of interest. In summary, the main contributions of this work include two aspects: the introduction of a new type of adaptive wavelet network that can lead to transparent models where the significance of both model variables and the associated wavelet neurons can easily be interpreted; the development of a hybrid learning scheme that combines a nonlinear optimisation (OPP+PSO) method with a linear least squares algorithm. Moreover, the proposed wavelet network is nearly self-implemented, that is, all within-network parameters can automatically be adjusted by the proposed algorithms. This is desirable for any structure-unknown or black-box modelling problems.

The rest of the paper is organised as below. In section 2, the architecture of the new lattice dynamical wavelet neural network is presented. In section 3, a two-stage hybrid training scheme, involving both the OPP+PSO approach and a forward orthogonal regression algorithm, is addressed in detail. In section 4, two examples are presented to demonstrate the effectiveness and performance of the new modelling framework. Finally, the work is summarised in section 5.

2. The Architecture of the New LDWNN

The new lattice dynamical wavelet neural network (LDWNN) model is constructed using wavelet frames, which are sets of non-independent vectors and thus form redundant bases for vectors defined in a given space. Starting with the discretisation of the wavelet transform, this section represents the architecture of the new LDWNN.

2.1 Wavelet frames and wavelet series

Consider a wavelet family below

$$\psi^{a,b}(x) = a^{-1/2}\psi\left(\frac{x-b}{a}\right) \tag{1}$$

where $a \in \mathrm{R}^+$, $b \in \mathrm{R}$, and the mother wavelet ψ is admissible. The admissibility condition is depicted using the Fourier transform $\hat{\psi}(\xi)$ of the function ψ as $C_\psi = \int_{-\infty}^{\infty} \xi^{-1} |\hat{\psi}(\xi)|^2 d\xi < \infty$. It has been shown (Daubechies 1992) that for reasonable ψ, there exists a grid $G = \{(a_m, b_n) : a_m \in \mathrm{R}^+, b_n \in \mathrm{R}; m, n \in \mathrm{Z}\}$,

such that the family $\psi_{m,n}(x) = a_m^{-1/2}\psi(a_m x - b_n)$, with $(a_m, b_n) \in G$, constitute a frame for $L^2(\mathrm{R})$ (the space of all square integrable functions), with frame bounds *A*, *B*; that is, for all $f \in L^2(\mathrm{R})$

$$A \| f \|^2 \le \sum_{m,n} |< f, \psi_{m,n} >|^2 \le B \| f \|^2 \tag{2}$$

where the symbols '<, >' and '|| ||' denote the inner product and the norm, respectively, following the ordinary definitions. The fact that $\psi_{m,n}$, whose parameters are restricted to a grid *G*, constitute a frame for $L^2(\mathrm{R})$ can guarantee that for any $f \in L^2(\mathrm{R})$, there exits a sequence $\{c_{m,n} : m,n \in \mathrm{Z}\} \in l^2(\mathrm{Z}^2)$ (the set of all double square summable sequences of complex numbers indexed by integers) such that

$$f(x) = \sum_m \sum_n c_{m,n} \psi_{m,n}(x) \tag{3}$$

A special choice of the grid *G* is to let $a_m = a_0^m$, $b_n = n b_0 a_0^m$, with $a_0 > 1$, $b_0 > 0$. Daubechies (1992) gave a theoretical approach for calculating the wavelet coefficients $c_{m,n}$ in (3). For some very special choices of ψ and *G*, the family $\psi_{m,n}$ can constitute an orthogonal basis for $L^2(\mathrm{R})$. The most popular choice is $a_0 = 2$, $b_0 = 1$, for which there exists ψ, with good time-frequency localisation properties, such that $\psi_{m,n}(x) = 2^{-m/2}\psi(2^{-m}x - n)$ constitute an orthogonal basis for $L^2(\mathrm{R})$. Orthogonal wavelet bases play an important role in wavelet multiresolution analysis (MRA) (Mallat 1989), because now any $f \in L^2(\mathrm{R})$ can be perfectly represented as

$$f(x) = \sum_n a_{m_0,n} \phi_{m_0,n}(x) + \sum_{m \ge m_0} \sum_n c_{m,n} \psi_{m,n}(x) \tag{4}$$

where m_0 can be any integer number. The wavelet ψ, along the associated scale function ϕ, form an MRA. An important property of orthonormal decompositions is that the well known Pareval's theorem holds, that is, the energy of the signal *f* is conserved, without any loss, in the wavelet coefficients.

This study pivots on nonlinear dynamical modeling problems, where relative observations are often sparse and where the independent (input) variables involved in the *dynamical* model are often formed by some variables representing the past states in time and at different spatial locations; this is different from a typical signal decomposition, where a given signal is represented using a *static* model formed by some wavelet-based elementary building blocks. When trying to construct dynamical models for nonlinear dynamical systems, using wavelet frames or orthogonal wavelet bases, the following issues arise:

- How to choose the primary parameters a_0 and b_0 to form a wavelet frame model? Is the choice $a_0 = 2$ and $b_0 = 1$ usually optimal for constructing dynamical wavelet frame models? Clearly, the choices of optimal values for a_0 and b_0 are still an open problem when wavelet decompositions are used for nonlinear dynamical modelling.

- It is in general impossible to form a wavelet model that contains an infinite or a large number of wavelet functions; a truncated wavelet model is thus often considered. Then, how to determine the effective range of the dilation and translation parameter indices m and n?

Clearly, there are no unique solutions to the above issues because the choices of these parameters are indeed problem specific. One best alternative is perhaps to let the data speak for themselves, that is, to let the relevant observed data themselves adaptively and automatically choose these parameters. This motivates the introduction of adaptive wavelet network models for nonlinear dynamical system modeling.

2.2 Adaptive versus fixed grid wavelet models

In practical applications, both continuous and discrete wavelet transforms have been introduced to construct wavelet models for function learning or dynamical modelling, and these models can be catalogued into two types: adaptive wavelet models and fixed grid wavelet (network) models (Billings and Wei 2005b). In adaptive wavelet networks, unknown parameters of the relevant wavelet functions are allowed to vary continuously within some specified space. To adaptively estimate these unknown parameters, efficient nonlinear least squares algorithms, for example the most commonly used back-propagation and Gaussian-Newton algorithms (Haykin 1999), often need to be involved for network training (Zhang and Benveniste 1992, Oussar et al. 1998). It has been proved that the convergence and the performance of the back-propagation algorithm strongly depends on the initialisation of relevant networks (Zhang and Benveniste 1992). Most existing adaptive wavelet networks are in structure similar to classical single-hidden-layer neural networks, and thus may lack physical interpretabilities for either model variables or relevant wavelet-neurons. This may be undesirable for some application cases where physical insight for both the model variables and the relative neurons are required. Another issue relative to adaptive wavelet networks is the determination of the size or complexity of the associated network models.

In fixed grid wavelet models, elementary building blocks, derived from some dyadic wavelets, are usually used to form relevant model terms (wavelet-neurons), and the dilation (scale) and translation (position) parameters of relevant wavelets are often predetermined and allowed to vary only in a fixed lattice; only the weights have to be optimised by training the network. Fixed grid wavelet models are in general easy to train without involving any nonlinear least squares problems (Zhang 1997, Coca and Billings 1997, Billings and Wei 2005b, Wei and Billings 2006a). An alternative solution for training this kind of network is to convert the networks into a linear-in-the-parameters problem, which can then be solved by using linear least squares type algorithms (Billings and Coca 1999, Xu and Ho 2002, Billings and Wei 2005a, Wei et al. 2006). Compared to nonlinear least squares algorithms, for example the well-known back-propagation algorithm for classical neural network training, methods involved in most fixed grid wavelet models are more constructive in that they often can automatically determine the network size and estimate the network coefficients in a reasonable number of iterations

(Zhang 1997). However, to train such a wavelet model, a specific dictionary, which contains a large number of candidate dyadic wavelets and which is often over-redundant, needs to be pre-determined. To construct such a dictionary, one needs to estimate the values for both the coarsest and the finest (scale) resolution levels, along with the corresponding shift parameters (these parameters are restricted to be integers). Although some rules of thumb are available (Zhang 1997, Wei and Billings 2004b), these values are still problem-specific and need to be pre-determined for each application. For some complex nonlinear dynamical modelling problems, the relative dictionary may involve a large number of candidate wavelet basis functions providing that the finest resolution level is chosen to be a large value; this may not be desirable for data arranging and storing, and for model subset selection.

2.3 The new LDWNN for spatio-temporal system modelling

CML's are a class of dynamical models, with discrete time and discrete space, but with continuous state variables (Kaneko 1993). Take the 2-D CML model, involving the nearest-neighbour cell coupling on a squared lattice with *Moore neighbourhoods*, as an example, this can be expressed as

$$s_{i,j}(t) = \sum_{-r\le p,q\le r} \alpha_{p,q}^{(1)} \Phi_{p,q}^{(1)}(s_{i+p,j+q}(t-1)) + \sum_{-r\le p,q\le r} \alpha_{p,q}^{(2)} \Phi_{p,q}^{(2)}(s_{i+p,j+q}(t-2)) + \mathrm{L} + \sum_{-r\le p,q\le r} \alpha_{p,q}^{(\tau)} \Phi_{p,q}^{(\tau)}(s_{i+p,j+q}(t-\tau)) \tag{5}$$

where t=1,2, …, i=1,2, …, I, j=1,2, …, J, $s_{i,j} \in \mathrm{R}$ is the state representing the cell C(i,j), τ is the time lag, $\Phi_{p,q}^{(\tau)}$ are some linear or nonlinear functions, $\alpha_{p,q}^{(\tau)}$ are connecting coefficients, and $r \ge 0$ is referred to as neighborhood radius indicating how many neighborhood cells are involved in the evolution procedure for generating each centre cell $s_{i,j}(t)$ from the past state space. Clearly, if r=0, model (5) will become a pure temporal process. The evolution law for boundary cells often needs to be pre-specified. If both I and J are very large, boundary conditions may not affect the resultant patterns; if, however, one of the two numbers is small, boundary conditions may significantly distort the original patterns. For details about how to set boundary conditions, see Chua and Roska (2001).

A typical case of the model (5) is that the time lag τ is assumed to be unity, that is, τ =1, and all the functions $\Phi_{p,q}^{(\tau)}$, with $-r \le p \le r$ and $-r \le q \le r$, are assumed to be the same. This simple space-invariant CML model is given below:

$$s_{i,j}(t) = \sum_{-r\le p,q\le r} \alpha_{p,q} \Phi(s_{i+p,j+q}(t-1)) \tag{6}$$

The evolution function Φ in the CML model (6) is often assumed to be known as some deterministic functions (Kaneko 1993). However, for real-word complex evolutionary images, a pre-determined function Φ may not sufficiently characterise the underlying dynamics; it may be better to learn, from avalable real observed data, an appropriate model structure for a given spatio-temporal system.

Table 1 Variables x_k, with k=1, 2, …, $(2r+1)^2$, represent $(2r+1)^2$ different cells

$C(i\text{-}r, j\text{-}r)$ x_1	…	$C(i\text{-}r, j)$ x_r	…	$C(i\text{-}r,j+r)$ x_{2r+1}
…	…	…	…	…
$C(i, j\text{-}r)$ $x_{r(2r+1)+1}$	…	$C(i,j)$ $x_{r(2r+1)+(r+1)}$	…	$C(i,j+r)$ $x_{(r+1)(2r+1)}$
	…		…	…
$C(i+r,j\text{-}r)$ $x_{2r(2r+1)+1}$	…	$C(i+r,j)$ $x_{2r(2r+1)+(r+1)}$	…	$C(i+r,j+r)$ $x_{(2r+1)\,(2r+1)}$

Now, assume that the true evolution functions $\Phi_{p,q}^{(k)}$ in (5) and Φ in (6) are unknown, but some relevant observations are available. The task of spatio-temporal system identification is to construct a model that can represent, as close as possible, the observed evolution procedure. Unlike constructing static models for typical data fitting, the objective of dynamical modelling is not merely to seek a model that fits the given data well, it also requires, at the same time, that the model should be capable of capturing the underlying system dynamics carried by the observed data, so that the resultant model can be used in simulation, analysis, and control studies.

Note that a total of $d=(2r+1)^2$ state variables are involved in the CML model given by (6). For convenience of description, introduce d single-indexed variables x_k, with k=1,2, …, d, to represent the d involved cells in the neighborhood, see Table 1. Also, let y represent the central cell $C(i, j)$. Then, the objective is to identify, from available data, a d-dimensional model

$$y(t)=f(\mathbf{x}(t))=f(x_1(t),x_2(t),\mathrm{L}\ ,x_d(t)) \tag{7a}$$

or, in an explicit form, with respect to the state variables

$$\begin{aligned} s_{i,j}(t)=f(\mathbf{s}(t))=f(&s_{i-r,j-r}(t-1),\mathrm{L}\ ,s_{i-r,j}(t-1),\mathrm{L}\ ,s_{i-r,j+r}(t-1),\mathrm{L}\ ,\\ &s_{i,j-r}(t-1),\mathrm{L}\ ,s_{i,j}(t-1),\mathrm{L}\ ,s_{i,j+r}(t-1),\mathrm{L}\ ,\\ &s_{i+r,j-r}(t-1),\mathrm{L}\ ,s_{i+r,j}(t-1),\mathrm{L}\ ,s_{i+r,j+r}(t-1)) \end{aligned} \tag{7b}$$

where $\mathbf{x}(t)$ and $\mathbf{s}(t)$ are state vectors formed by the relative state variables.

One of the most commonly used approaches for constructing the high dimensional model (7) is to approximate the multivariate function f using a set of functions of fewer variables (often univariate)

$$f(\mathbf{x}(t))=\sum_j w_j g_j(\mathbf{x}(t);\boldsymbol{\theta}_j) \tag{8}$$

where g_j are called the construction functions (hidden units), $\boldsymbol{\theta}_j$ are the associated parameter vectors,

and w_j are coefficients (weights).

Wavelets, due to their inherent property and excellent capability in time-frequency domain representation and approximation of arbitrary signals, can be used as the elementary building blocks to represent these construction functions g_j in (8). In practice, three types of wavelet models are often involved: single-hidden-layer wavelet networks (Zhang and Benveniste 1992), radial wavelet networks (Zhang 1997, Billings and Wei 2005b), and tensor product wavelet networks (Oussar et al. 1998, Billings and Wei 2005a, Wei and Billings 2004a, 2006). Taking the classical single-hidden-layer wavelet neural network as an example, the construction functions g_j in (8) are often expressed as $g_j(\mathbf{x};\boldsymbol{\theta}_j)=\psi(\mathbf{a}_j^T\mathbf{x}-b_j)$, where ψ is some wavelet, $\mathbf{x},\mathbf{a}_j\in \mathrm{R}^{+d}$, $b_j\in \mathrm{R}$, $\boldsymbol{\theta}_j=[\mathbf{a}_j^T,b_j]^T$. This is a kind of 'linear-interaction and then nonlinear-transform' process.

Inspired by the CML models (5) and (6), the construction functions g_j in (8) are chosen as below:

$$g_j(\mathbf{x};\boldsymbol{\theta}_j)=c_{1,j}\psi(x_1;a_{1,j},b_{1,j})+c_{2,j}\psi(x_2;a_{2,j},b_{2,j})+\mathrm{L}\ +c_{d,j}\psi(x_d;a_{d,j},b_{d,j}) \tag{9}$$

where $\psi(x_k;a_{k,j},b_{k,j})=\psi(a_{k,j}x_k-b_{k,j})$, with k=1,2, …, d, are wavelet basis functions, and $\boldsymbol{\theta}_j=[a_{1,j},b_{1,j},c_{1,j},\mathrm{L}\ ,a_{d,j},b_{d,j},c_{d,j}]^T$ are the parameter vectors that need to be optimised. Clearly, the construction functions g_j in (9) are of the form 'nonlinear-transform and then linear-interaction', which is totally different from the cases of typical single-hidden-layer wavelet neural networks.

Assume that a total of m construction functions, $g_1,g_2,\mathrm{L}\ ,g_m$, are involved in the network, then equation (8) can be expressed as

$$f(\mathbf{x}(t))=\sum_{j=1}^{m}w_j g_j(\mathbf{x}(t);\boldsymbol{\theta}_j)=\sum_{j=1}^{m}\sum_{k=1}^{d}\tilde{c}_{k,j}\psi(x_k;a_{k,j},b_{k,j}) \tag{10}$$

where $\tilde{c}_{k,j}=w_j c_{k,j}$. Now, the remaining key problem is how to construct the wavelet network model (10), where the elementary building block is $g(\mathbf{x};\boldsymbol{\theta})=\sum_{k=1}^{d}c_k\psi(x_k;a_k,b_k)$, where $\boldsymbol{\theta}=[a_1,b_1,c_1,\mathrm{L}\ ,a_d,b_d,c_d]^T$ is unknown and needs to be optimised. Some points need to be considered:

- Which training strategy should be used to construct such a network?
- With a chosen wavelet ψ, the parameters $\boldsymbol{\theta}=[a_1,b_1,c_1,\mathrm{L}\ ,a_d,b_d,c_d]^T$ are unknown and need to be optimised. How to calculate these parameters?
- How to determine the size of the network, or the number of construction functions?
- How to measure the significance of each model variable and the involved wavelet neurons?

Unlike in fixed grid wavelet network models, where a dictionary of candidate basis functions needs to be initially provided, based on which some search and pruning algorithms are applied to find a set of significant basis functions (Zhang 1997, Xu and Ho 2002, Billings and Wei 2005b), this study will consider a type of growing wavelet neural network, where a constructive learning algorithm that

can be used to automatically and adaptively augment such a network will be provided.

3. Training the New LDWNN

Many constructive learning algorithms, applicable to constructing typical neural networks (Haykin 1999), can be found in the literature (Fahlman and Lebiere 1990, Jones 1992, Kwok and Yeung 1997a, Reed and Marks 1999). The projection pursuit regression (PPR) (Friedman and Stuetzle 1981) and some variations (Hwang et al. 1994, Kwok and Yeung 1997b) are among the class of the most commonly used approaches for augmenting single-hidden-layer neural networks. The basic idea of these kind of algorithms is to successively approximate the function *f* by progressively minimising approximation errors. It generally starts from $f_0 = 0$ (the initial approximation function is set to be zero), evolves in a stepwise manner by searching through steps *j*=1,2, …,*m*; at the *j*th step, the approximation f_j is augmented by including the *j*th construction function $g_j(\mathbf{x};\boldsymbol{\theta}_j)$ that produces the largest decrease in the approximation error, that is, it minimises the objective function: $\min_{\alpha,\boldsymbol{\theta}} \| f - (f_{j-1} + \alpha g(\mathbf{x};\boldsymbol{\theta}) \|^2$.

Inspired by the successful applications of these popular constructive learning algorithms, this study proposes a practical orthogonal projection pursuit (OPP) learning scheme, assisted by a particle swarm optimisation (PSO) algorithm. Similar to other popular constructive algorithms, networks produced by the OPP algorithm may be redundant. To remove or reduce redundancy, an orthogonal least squares (OLS) type learning algorithm (Billings 1989, Chen et al. 1989, 1991) is applied to refine and improve the initially generated network by the OPP+PSO algorithm. Detailed discussions on the network training procedure are given below.

3.1 The OPP algorithm aided by PSO for first stage network training

Let $\mathbf{y} = [y(1), y(2),..., y(N)]^T \in \mathrm{R}^N$ be the vector of given observations of the output signal, $\mathbf{x}_k = [x_k(1), x_k(2),\mathrm{L}\ , x_k(N)]^T$ the vector of the observations for the *k*th input variable, with *k*=1,2, …, *d*. For any given $\boldsymbol{\theta} = [a_1, b_1, c_1, \mathrm{L}\ , a_d, b_d, c_d]^T$, let $\boldsymbol{\psi}_k = [\psi(x_k(1); a_k, b_k), \mathrm{L}\ , \psi(x_k(N); a_k, b_k)]^T$ and $\mathbf{g}(\mathbf{X};\boldsymbol{\theta}) = \sum_{k=1}^{d} c_k \boldsymbol{\psi}_k$, where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \mathrm{L}\ , \mathbf{x}_d]$.

The OPP algorithm is implemented in a stepwise fashion; at each step a construction vector that minimises the projection error will be determined. Starting with $\mathbf{r}_0 = \mathbf{y}$, find a construction function $\mathbf{g}_1 = \mathbf{g}(\mathbf{X};\boldsymbol{\theta}_1)$ such that $\boldsymbol{\theta}_1 = \arg\min_{\boldsymbol{\theta}} \{\| \mathbf{r}_0 - \mathbf{g}(X;\boldsymbol{\theta}) \|^2\}$. The associated residual vector may be defined as $\mathbf{r}_1 = \mathbf{r}_0 - \mathbf{g}_1$, which can be used as the "fake desired target signal" to produce the second construction vector $\mathbf{g}_2$. However, it should be noted that the coefficient $\boldsymbol{\theta}_1$ is not always identical to the true (theoretical) optimal value $\boldsymbol{\theta}_1^*$, no matter what optimisation algorithms are applied. As a consequence,

$\mathbf{r}_1 = \mathbf{r}_0 - \mathbf{g}_1$ may not be orthogonal with the construction vector $\mathbf{g}_1$. To make the associated residual orthogonal with the relevant construction vector, the residual is then defined as $\mathbf{r}_1 = \mathbf{r}_0 - \alpha_1 \mathbf{g}_1$, where $\alpha_1 = <\mathbf{r}_1, \mathbf{g}_1> / \| \mathbf{g}_1 \|^2$. Note that from now on the inner product is for sampled vectors in *N*-dimensional Euclidian space, for example, the inner product of the two vectors $\mathbf{u} = [u(1), u(2), \text{L}\ , u(N)]^T$ and $\mathbf{v} = [v(1), v(2), \text{L}\ , v(N)]^T$ is defined as $<\mathbf{u}, \mathbf{v}> = \mathbf{u}^T \mathbf{v} = \sum_{k=1}^{N} u(k) v(k)$; this is different from that previously defined in (2), where the inner product is imposed to functions in $L^2(\text{R})$.

Assume that at the (*n*-1)th step, a total of (*n*-1) construction vectors $\mathbf{g}_j = \mathbf{g}(\mathbf{X}; \boldsymbol{\theta}_j)$, with *j*=1,2, …, *n*-1, have been obtained. Let $\mathbf{r}_{n-1}$ be the residual vector associated with these (*n*-1) obtained vectors when they are used to approximate the desired signal $\mathbf{y}$. The *n*th construction vector can be obtained by choosing $\boldsymbol{\theta}_n = \arg\min_{\boldsymbol{\theta}} \{ \| \mathbf{r}_{n-1} - \mathbf{g}(X; \boldsymbol{\theta}) \|^2 \}$ and $\mathbf{g}_n = \mathbf{g}(\mathbf{X}; \boldsymbol{\theta}_n)$. The associated residual vector can be defined as

$$\mathbf{r}_n = \mathbf{r}_{n-1} - \alpha_n \, \mathbf{g}(\mathbf{X}; \boldsymbol{\theta}_n) \tag{11}$$

where

$$\alpha_n = \frac{<\mathbf{r}_{n-1}, \mathbf{g}_n>}{\| \mathbf{g}_n \|^2} = \frac{<\mathbf{r}_{n-1}, \mathbf{g}(\mathbf{X}; \boldsymbol{\theta}_n)>}{\| \mathbf{g}(\mathbf{X}; \boldsymbol{\theta}_n) \|^2} \tag{12}$$

Inserting (12) into (11), yields,

$$\mathbf{r}_n = \mathbf{r}_{n-1} - \frac{<\mathbf{r}_{n-1}, \mathbf{g}_n>}{\| \mathbf{g}_n \|^2} \mathbf{g}_n \tag{13}$$

From (13),

$$\| \mathbf{r}_n \|^2 = \| \mathbf{r}_{n-1} \|^2 - \frac{<\mathbf{r}_{n-1}, \mathbf{g}_n>^2}{\| \mathbf{g}_n \|^2} = \| \mathbf{r}_{n-1} \|^2 - \alpha_n <\mathbf{r}_{n-1}, \mathbf{g}_n> \tag{14}$$

By respectively summing (13) and (14) for *n* from 2 to *m*+1, yields

$$\mathbf{y} = \sum_{n=1}^{m} \frac{<\mathbf{r}_{n-1}, \mathbf{g}_n>}{\| \mathbf{g}_n \|^2} \mathbf{g}_n + \mathbf{r}_m = \sum_{n=1}^{m} \alpha_n \mathbf{g}_n + \mathbf{r}_m \tag{15}$$

$$\| \mathbf{r}_m \|^2 = \| \mathbf{y} \|^2 - \sum_{n=1}^{m} \frac{<\mathbf{r}_{n-1}, \mathbf{g}_n>^2}{\| \mathbf{g}_n \|^2} = \| \mathbf{y} \|^2 - \sum_{n=1}^{m} \alpha_n <\mathbf{r}_{n-1}, \mathbf{g}_n> \tag{16}$$

The residual sum of squares, also called the sum of squares error, $\| \mathbf{r}_n \|^2$, can be used to form a criterion to stop the growing procedure. For example, the criterion can be chosen as *error-to-signal*

ratio: ESR $=\| \mathbf{r}_n \|^2 / \| \mathbf{y} \|^2$; when ESR becomes smaller than a pre-specified threshold value, the growing procedure can then be terminated.

Now the OPP algorithm can briefly be summarised as follows.

The OPP algorithm:

Initialisation: $\mathbf{r}_0 = \mathbf{y}$; $f_0 = 0$; ESR=0;

while { ESR $\geq \eta$ or $n \leq m$PEM }; //{η is a pre-specified very small threshold value.}//

//{mOPP is the maximum number of construction functions

// permitted to be included in the network} //

for n=1 to mOPP

//{Starting from some random (but reasonable) value for the parameter vector $\boldsymbol{\theta}$, optimise the following function using the PSO algorithm.}//

$$\boldsymbol{\theta}_n = \arg\min_{\boldsymbol{\theta}} \left\{ \| \mathbf{r}_{n-1} - \mathbf{g}(\mathbf{X}; \boldsymbol{\theta}) \|^2 \right\};$$

$$\alpha_n = \frac{< \mathbf{r}_{n-1}, \mathbf{g}_n >}{\| \mathbf{g}_n \|^2};$$

$$\mathbf{r}_n = \mathbf{r}_{n-1} - \alpha_n \mathbf{g}_n;$$

$$\text{ESR} = \| \mathbf{r}_n \|^2 / \| \mathbf{y} \|^2;$$

end for

end while

It is clear from (14) that the sequence $\| \mathbf{r}_n \|^2$ is strictly decreasing and positive; thus, by following the method given in Zhang (1993), Kwok and Yeung (1997b) and Huang et al. (2006), it can easily be proved that the residual $\mathbf{r}_n$ is a Cauchy sequence, and as a consequence, the residual $\mathbf{r}_n$ converges to zero. The algorithm is thus convergent. The above OPP algorithm is in structure similar to the projection pursuit regression (Friedman and Stuetzle 1981) and other constructive learning algorithms (Mallat and Zhang 1993, Hwang et al. 1994, Kwok and Yeung 1997a, 1997b), but the implementation of the OPP algorithm is totally different from these existing algorithms. For example, in the projection pursuit regression method, the construction functions are nonparametric and in general unknown before hand; in the OPP algorithm, however, the construction functions are formed by a linear combination of d individual parametric functions. In the matching pursuit method, the construction functions are restricted to a specified dictionary, where relevant adjustable parameters of individual candidates are permitted to vary in a given grid, while in the OPP algorithm no such limits are imposed on construction functions. Moreover, in the OPP algorithm, the elementary building blocks are linear combinations of some wavelets, where unknown parameters are optimised by using some PSO algorithm that does not need any information on the gradients of the object functions, this enables the PSO to be very suitable for nonlinear optimisation problems where the relevant object functions are not differentiable or the gradients are computationally expensive to obtain (Kennedy et al. 2001). However, like the projection pursuit regression and the matching pursuit algorithms, the OPP algorithm may produce redundant models. To refine and improve the OPP produced network models, the orthogonal least squares (OLS) learning algorithm (Billings 1989, Chen et al. 1989, 1991) is then applied to remove any severe redundancy.

3.2 The PSO algorithm for parameter estimation

Particle swarm optimisation (PSO), originally inspired by some sociological behaviour associated with, for example, bird flocking (Kennedy and Eberhart 2001), is a population-based stochastic optimisation algorithm that was first proposed by Kennedy and Eberhart in 1995 (Kennedy and Eberhart 1995, Eberhart and Kennedy 1995). In PSO, the population is referred to as a *swarm*, while the individuals are referred to as *particles*; each particle moves, in the search space, with some random *velocity*, and remembers and retains the *best position* it has ever been. The mechanism of PSO can succinctly be explained as follows. The position of each particle can be viewed as a possible solution to a given optimization problem. In each iteration (one step move), each particle accelerates its move toward a new potential position, by adaptively using information about its own *personal best position* obtained so far, as well as the information of the *global best position* achieved so far by any other particles in the swarm. Thus, if any promising new position is discovered by any individual particle, then all the other particles will move closer towards it (Parsopoulos and Vrahatis 2004). In this way, PSO will finally find, in an iterative manner, a best solution to the given optimisation problem.

Now consider an s dimensional optimisation problem, where the relevant parameter vector to be optimised is denoted by $\boldsymbol{\theta} = [\theta_1, \theta_2, \mathrm{L}\ , \theta_s]^T \in \Theta \subset \mathrm{R}^s$. Assume that a total of L particles are involved in the relevant swarm. Denote the position of the ith particle at the present time t by $\boldsymbol{\theta}_i(t)$, the relative velocity by $\mathbf{v}_i(t)$, the personal best position by $\mathbf{p}_i(t)$, and the global best position obtained so far by $\mathbf{p}_g(t)$. Following Kennedy et al. (2001), Shi and Eberhart (1998a, 1998b), Clerc and Kennedy (2002), PSO can be implemented using the iterative equations below

$$\mathbf{v}_i(t+1) = w(t)\mathbf{v}_i(t) + c_1 r_1[\mathbf{p}_i(t) - \boldsymbol{\theta}_i(t)] + c_2 r_2[\mathbf{p}_g(t) - \boldsymbol{\theta}_i(t)] \tag{17a}$$

$$\boldsymbol{\theta}_i(t+1) = \boldsymbol{\theta}_i(t) + \chi \mathbf{v}_i(t+1) \tag{17b}$$

where i=1,2, …, L; $w(t)$ are the inertia weights, c_1 and c_2 are the acceleration coefficients, also referred to as the cognitive and social parameters; $\chi = 2/|2 - \phi - \sqrt{\phi^2 - 4\phi}|$, with $\phi = c_1 + c_2 > 4$, is a constriction factor used to obtain good convergence performance by controlling explosive particle movements; r_1 and r_2 are random numbers that are uniformly distributed in [0,1]. Typical choices for c_1 and c_2 are to set $c_1 = c_2 = 2$ (Kennedy and Eberhart 1995, Eberhart and Kennedy 1995). Also, values initially starting from unity and then gradually declining to zero are considered as a good choice for w (Eberhart and Shi 1998, Shi and Eberhart 1998a, 1998b, van den Bergh and Engelbrecht 2004).

Let $\pi(\boldsymbol{\theta})$ be the function that needs to be minimised, then the personal best position of each particle can be updated as below (van den Bergh and Engelbrecht 2004)

$$\mathbf{p}_i(t+1) = \begin{cases} \mathbf{p}_i(t), & \text{if } \pi(\boldsymbol{\theta}_i(t+1)) \geq \pi(\mathbf{p}_i(t)) \\ \boldsymbol{\theta}_i(t+1), & \text{if } \pi(\boldsymbol{\theta}_i(t+1)) < \pi(\mathbf{p}_i(t)) \end{cases} \tag{18}$$

While the global best position achieved by any particle during all previous iterations is defined as

$$\mathbf{p}_g(t+1) = \arg\min_{\mathbf{p}_i} \pi(\mathbf{p}_i(t+1)), \qquad 1 \leq i \leq L. \tag{19}$$

In the OPP algorithm discussed in the previous section, the objective function is defined as

$$\pi_{n-1}(\boldsymbol{\theta}) = \| \mathbf{r}_{n-1} - \mathbf{g}(\mathbf{X};\boldsymbol{\theta}) \|^2 = \sum_{t=1}^{N} [r_{n-1}(t) - g(\mathbf{x}(t);\boldsymbol{\theta})]^2 \tag{20}$$

where N is the number of training samples, $\mathbf{X}$ and $\boldsymbol{\theta}$ are defined as in the previous section, $\mathbf{x}(t) = [x_1(t), x_2(t), \mathrm{L}\ , x_d(t)]^T$ is defined as in section 2, and $g(\mathbf{x}(t);\boldsymbol{\theta}) = \sum_{k=1}^{d} c_k \psi(x_k(t); a_k, b_k)$.

With regard to the termination of the optimisation procedure, the criterion can be chosen as below. Let 'mPSO' be the maximum number of permitted iterations. The optimization procedure can then be terminated when either the iteration index exceeds 'mPSO', or when the parameter to be optimized becomes stable, that is, when $\| \boldsymbol{\theta}(t+1) - \boldsymbol{\theta}(t) \|^2 \leq \delta$, where δ is a pre-specified small number, say $\delta \leq 10^{-5}$.

3.3 Refine the network using the forward orthogonal regression algorithm

Assume that a total of m construction functions $g_j(\mathbf{x};\boldsymbol{\theta}_j)$, where $\mathbf{x}(t) = [x_1(t), x_2(t), \mathrm{L}\ , x_d(t)]^T$ and j=1,2, …, m, are involved in the network produced at the first stage. It is known that each g_j involves d individual wavelets, thus a total of $M = d \times m$ elementary wavelet neurons are involved in the network. Denote the set of these M wavelets by

$$\Omega = \{\psi_{k,j} : \psi_{k,j}(x_k) = \psi(x_k; a_{k,j}, b_{k,j}), (k,j) \in \Gamma\} \tag{21}$$

where $\Gamma = \{(k,j) : k = 1,2,\mathrm{L}\ , d; j = 1,2,\mathrm{L}\ , m\}$. Note that all the parameters $a_{k,j}$ and $b_{k,j}$ have already been estimated at the first stage.

The objective of this refinement stage is to reselect the most significant wavelet functions from the set Ω, to form a more compact model for given nonlinear identification problems. Let $\mathbf{y}$ and $\mathbf{x}_k$ be defined as in the previous section, and let $\boldsymbol{\psi}_{k,j} = [\psi(x_k(1); a_{k,j}, b_{k,j}), \mathrm{L}\ , \psi(x_k(N) : a_{k,j}, b_{k,j})]^T$, where $(k,j) \in \Gamma$. Also, let

$$D = \{\boldsymbol{\varphi}_{(k-1)m+j} : \boldsymbol{\varphi}_{(k-1)m+j} = \boldsymbol{\psi}_{k,j}, (k,j) \in \Gamma\} = \{\boldsymbol{\varphi}_1, \mathrm{L}\ , \boldsymbol{\varphi}_2, \boldsymbol{\varphi}_M\} \tag{22}$$

The network refinement problem amounts to finding, from the dictionary D, a full dimensional

subset $D_n = \{\mathbf{p}_1, \mathrm{L}\ , \mathbf{p}_n\} = \{\boldsymbol{\varphi}_{i_1}, \mathrm{L}\ , \boldsymbol{\varphi}_{i_n}\}$, where $\boldsymbol{\alpha}_k = \boldsymbol{\varphi}_{i_k}$, $i_k \in \{1,2,\mathrm{L}\ , M\}$ and k=1,2, …, n (generally $n << M$), so that $\mathbf{y}$ can be satisfactorily approximated using a linear combination of $\mathbf{p}_1, \mathbf{p}_2, \mathrm{L}\ , \mathbf{p}_n$ as below

$$\mathbf{y} = \beta_1 \mathbf{p}_1 + \mathrm{L}\ + \beta_n \mathbf{p}_n + \mathbf{e} \tag{23}$$

or in a compact matrix form

$$\mathbf{y} = \mathbf{P}\boldsymbol{\beta} + \mathbf{e} \tag{24}$$

where the matrix $\mathbf{P} = [\mathbf{p}_1, \mathrm{L}\ , \mathbf{p}_n]$ is assumed to be of full column rank, $\boldsymbol{\beta} = [\beta_1, \mathrm{L}\ , \beta_n]^T$ is a parameter vector, and $\mathbf{e}$ is the approximation error vector. The regression matrix $\mathbf{P}$ in (24) is full rank in columns and thus can be orthogonally decomposed as

$$\mathbf{P} = \mathbf{QR} \tag{25}$$

where $\mathbf{Q}$ is an $N \times n$ matrix with orthogonal columns $\mathbf{q}_1, \mathbf{q}_2, \mathrm{L}\ , \mathbf{q}_n$, and $\mathbf{R}$ is an $n \times n$ unit upper triangular matrix whose entries $r_{ij} (1 \le i \le j \le n)$ are calculated during the orthogonalization procedure. Inserting (25) into (24), yields,

$$\mathbf{y} = \mathbf{Q}(\mathbf{R}\boldsymbol{\beta}) + \mathbf{e} = \mathbf{Q}\boldsymbol{\gamma} + \mathbf{e} \tag{26}$$

where $\boldsymbol{\gamma} = [\gamma_1, \gamma_2, \mathrm{L}\ , \gamma_n]^T$, with $\gamma_i = < \mathbf{y}, \mathbf{q}_i > / \| \mathbf{q}_i \|^2$ and i=1,2, …, n. From (26),

$$\| \mathbf{y} \|^2 = \sum_{i=1}^{n} \gamma_i^2 \| \mathbf{q}_i \|^2 + \| \mathbf{e} \|^2 \tag{27}$$

Thus, the output variance consists of two parts: the desired output, $\sum_{i=1}^{n} \gamma_i^2 \| \mathbf{q}_i \|^2$, which can be explained by the selected regressors (wavelet functions); and the residual part, $\| \mathbf{e} \|^2$, representing the unexplained variance. Note that each term $\gamma_i \mathbf{q}_i$ in (27) makes an individual contribution to the designed signal $\mathbf{y}$, by giving an increment to the desired output variance. The significance, of the ith vector $\mathbf{q}_i$, caused by including the ith vector $\mathbf{p}_i$, can be measured by introducing the concept of the error reduction ratio (ERR) (Billings 1989, Chen et al. 1989, 1991), which is defined as

$$\mathrm{ERR}_i = \frac{\gamma_i^2 \| \mathbf{q}_i \|^2}{\| \mathbf{y} \|^2} \tag{28}$$

The ERR criterion provides a useful index to indicate which candidate vectors are important and should be included in the model. As in the OPP algorithm, the error-to-signal ratio (ESR) can be used to form a criterion to stop the search procedure. Following the suggestion in Billings and Wei (2007a), the penalised ESR criterion

$$\mathrm{PESR}_n = (1-\lambda n/N)^2(1-\sum_{i=1}^{n}\mathrm{ERR}_i) \tag{29}$$

will be used to monitor the regressor search procedure; the number of regressors (wavelet functions) will be chosen as the value where PESR arrives it minimum. Billings and Wei (2007a) suggest that the adjustable parameter λ be chosen between 5 and 10.

The forward orthogonal regression (FOR) algorithm used in this study is briefly summarised below. Some recently improved versions or variants of the OLS algorithm can be found in Chen et al. (2003, 2004), Billings and Wei (2007b), and the references therein.

The FOR algorithm:

Step 1**:** Set $U_1 = \{1,2,\mathrm{L}\ ,M\}$;

for j=1 to M

$\mathbf{q}_j^{(1)} = \boldsymbol{\varphi}_j$; //{if $\| \mathbf{q}_j^{(1)} \|^2 \le \varepsilon$, set $\mathrm{err}^{(1)}[j]=0$ }//

$$\gamma_j^{(1)} = \frac{<\mathbf{y},\mathbf{q}_j^{(1)}>}{\| \mathbf{q}_j^{(1)} \|^2};$$

$$\mathrm{err}^{(1)}[j] = \frac{(\gamma_j^{(1)})^2 \| \mathbf{q}_j^{(1)} \|^2}{\| \mathbf{y} \|^2};$$

end for

$l_1 = \arg\max_{i\in U_1}\{\mathrm{err}^{(1)}[i]\}$;

$V_1 = \{l_1\} \mathrm{U} \{\arg_{j\in U_1}(\| \mathbf{q}_j^{(1)} \|^2 < \varepsilon)\}$;

$\mathbf{p}_1 = \boldsymbol{\varphi}_{l_1}$; $\mathbf{q}_1 = \mathbf{p}_1$; $\gamma_1 = \gamma_{l_1}^{(1)}$;

$\mathrm{err}[1] = \mathrm{err}^{(1)}[l_1]$; $\mathrm{serr}[1] = \mathrm{err}[1]$; $\mathrm{esr}[1] = 1-\mathrm{serr}[1]$;

$\mathrm{pesr}[1] = (1-\lambda/N)^2\,\mathrm{esr}[1]$;

Step n, $n \ge 2$ **:**

For n=2 to M

$U_n = U_{n-1} \setminus V_{n-1}$;

for $j \in U_n$

$$\mathbf{q}_j^{(n)} = \boldsymbol{\varphi}_j - \sum_{k=1}^{n-1}\frac{<\boldsymbol{\varphi}_j,\mathbf{q}_k>}{\| \mathbf{q}_k \|^2}\mathbf{q}_k; \quad \text{//\{if } \| \mathbf{q}_j^{(n)} \|^2 \le \varepsilon, \text{ set } \mathrm{err}^{(n)}[j]=0 \text{ \}//}$$

$$\gamma_j^{(n)} = \frac{<\mathbf{y},\mathbf{q}_j^{(n)}>}{\| \mathbf{q}_j^{(n)} \|^2};$$

$$\mathrm{err}^{(n)}[j] = \frac{(\gamma_j^{(n)})^2 \| \mathbf{q}_j^{(n)} \|^2}{\| \mathbf{y} \|^2};$$

end for (end loop for j)

$l_n = \arg\max_{j\in U_n}\{\mathrm{err}^{(n)}[j]\}$;

$V_n = \{l_n\} \mathrm{U} \{\arg_{j\in U_n}(\| \mathbf{q}_j^{(n)} \|^2 < \varepsilon)\}$;

$\mathbf{p}_n = \boldsymbol{\varphi}_{l_n}$; $\mathbf{q}_n = \mathbf{q}_{l_n}^{(n)}$; $\gamma_n = \gamma_{l_n}^{(n)}$;

$\mathrm{err}[n] = \mathrm{err}^{(n)}[l_n]$; $\mathrm{serr}[n] = \sum_{k=1}^{n}\mathrm{err}[k]$; $\mathrm{esr}[n] = 1-\mathrm{serr}[n]$;

$\mathrm{pesr}[n] = (1-\lambda n/N)^2\,\mathrm{esr}[n]$;

for k=1 to n

$$r_{k,n} = \frac{< \mathbf{p}_n, \mathbf{q}_k >}{\| \mathbf{q}_k \|^2}, \text{ for } k < n; \; r_{k,n} = 1, \text{ for } k = n;$$

end for (end loop for k)

end for (end loop for n)

The FOR algorithm provides an effective tool for successively selecting significant model terms (hidden units) in supervised learning problems. Terms are selected step by step, one term at a time. The inclusion of redundant bases, which are linearly dependent on the previous selected bases, can be efficiently excluded by eliminating the candidate basis vectors for which $\| \mathbf{q}_j^{(n)} \|^2$ are less than a predetermined threshold ε, say $\varepsilon \le 10^{-10}$. Assume that a total of m significant vectors are selected, then the unknown parameter $\boldsymbol{\beta} = [\beta_1, \beta_2, \mathrm{L}\ , \beta_m]^T$ can easily be calculated from the triangular equation $\mathbf{R}\boldsymbol{\beta} = \boldsymbol{\gamma}$, where R is an upper triangular matrix and $\boldsymbol{\gamma} = [\gamma_1, \gamma_2, \mathrm{L}\ , \gamma_m]^T$ with $\gamma_i = < \mathbf{y}, \mathbf{q}_i > / \| \mathbf{q}_i \|^2$ for i=1,2,…, m.

Let $\{\psi_{k,j} : \psi_{k,j}(x_k) = \psi(a_{k,j} x_k - b_{k,j}), j = 1,2,\mathrm{L}\ , m_k\}$ be the OLS produced set of the wavelet functions that are relevant to the kth input (independent) variable x_k, with k=1,2, …, d. The wavelet network model obtained at the OPP stage will then reduce to

$$\begin{aligned} f(\mathbf{x}(t)) &= \sum_{j=1}^{m_1} \theta_{1,j} \psi(x_1; a_{1,j}, b_{1,j}) + \sum_{j=1}^{m_2} \theta_{2,j} \psi(x_2; a_{2,j}, b_{2,j}) + \mathrm{L} + \sum_{j=1}^{m_d} \theta_{d,j} \psi(x_d; a_{d,j}, b_{d,j}) \\ &= f_1(x_1) + f_2(x_2) + \mathrm{L} + f_d(x_d) \end{aligned} \tag{30}$$

where $f_k(x_k) = \sum_{j=1}^{m_k} \theta_{k,j} \psi(x_k; a_{k,j}, b_{k,j})$. The network (30) can be viewed as a wavelet-based implementation of the well known and widely applied generalised additive model (Hastie and Tibshirani 1990), which can not only avoid the curse of dimensionality, but also provides the ability to detect nonlinear dynamics and nonlinear patterns, without sacrificing interpretability of the relevant component functions. Generalised additive models, combined with other modelling techniques, have recently become extremely popular and been widely applied in diverse areas (Aerts et al. 2002, Ruppert et al. 2003, Wood 2004, Brezger and Lang 2006, Lado et al. 2006).

Note that the wavelet neural network (30), as a generalised additive model, is in structure different from that given by (10).

4. Numerical Examples

This section presents two examples, one for artificial data and another for real data, to illustrate the application procedure of the new network modelling procedure.

4.1 Kaneko's 2-D CML model

The well known 2-D CML model (Kaneko 1989), involving five nearest-neighbour cells coupled on a squared lattice with *von Neumann* neighbourhoods, is given below

$$s_{i,j}(t) = (1-c)\Phi(s_{i,j}(t-1)) + \frac{c}{4}[\Phi(s_{i,j-1}(t-1)) + \Phi(s_{i,j+1}(t-1)) + \Phi(s_{i-1,j}(t-1)) + \Phi(s_{i+1,j}(t-1))] \quad (31)$$

where the function Φ was chosen as the logistic map $\Phi(x) = 1 - ax^2$. It has been shown that this model can produce rich spatio-temporal patterns. In the example here the coefficients a and c were chosen to be 1.5 and 0.4, respectively. Starting with some given initial and boundary conditions that are shown in Table 2, the model was simulated and a set of snapshot patterns were obtained; some of these patterns are presented in Fig. 1.

A total of N=4000 simulated data pairs, $\{\mathbf{x}(k), y(k)\}_{k=1,2,\ldots,N}$, were used for the network training. Note that $y(k)$ represents the value of the relevant central cell at the present time instant, and $\mathbf{x}(k) = [x_1(k), x_2(k), \mathrm{L}\ , x_5(k)]^T$ represent the values of the five involved cells in the neighbourhood at the previous time instant. These 4000 data pairs were formed as follows. Firstly, 10 adjacent pattern pairs were randomly chosen; pattern pairs here are referred to two patterns that are with abutting time instants, for example, patterns at the abutting time instants 11 and 12 form an adjacent pattern pair. Secondly, 400 data pairs were randomly chosen in each of these 10 pattern pairs. To make the training data more 'realistic', an additive Gaussian noise ξ, with zero mean and a standard deviation σ_ξ = 0.02, was added to the 'output' $y(k)$ for k=1,2, …, N.

The Mexican hat wavelet function, defined as $\psi(x) = (1-x^2)e^{-x^2/2}$, was used as the elementary building block for constructing the wavelet network model. All the experiment conditions involved in the modelling procedure for this example are shown in Table 2. The error-to-signal ratio, ESR, calculated by the OPP algorithm, is shown in Fig. 2, and the penalised error-to-signal ratio, PESR, produced by the FOR algorithm, is shown in Fig. 3. It is clear, from the PESR index, that the most appropriate number of wavelets is 11, where the PESR index arrives at its minimum; thus a total of 11 wavelets were included in the final network model below:

$$\begin{aligned} s_{i,j}(t) = {} & \sum_{k=1}^{4} c_{1,k}\psi(s_{i,j}(t-1); a_{1,k}, b_{1,k}) + \sum_{k=1}^{2} c_{2,k}\psi(s_{i,j-1}(t-1); a_{2,k}, b_{2,k}) \\ & + c_{3,1}\psi(s_{i,j+1}(t-1); a_{3,1}, b_{3,1}) + \sum_{k=1}^{2} c_{4,k}\psi(s_{i-1,j}(t-1); a_{4,k}, b_{4,k}) \\ & + \sum_{k=1}^{2} c_{5,k}\psi(s_{i+1,j}(t-1); a_{5,k}, b_{5,k}) \end{aligned} \quad (32)$$

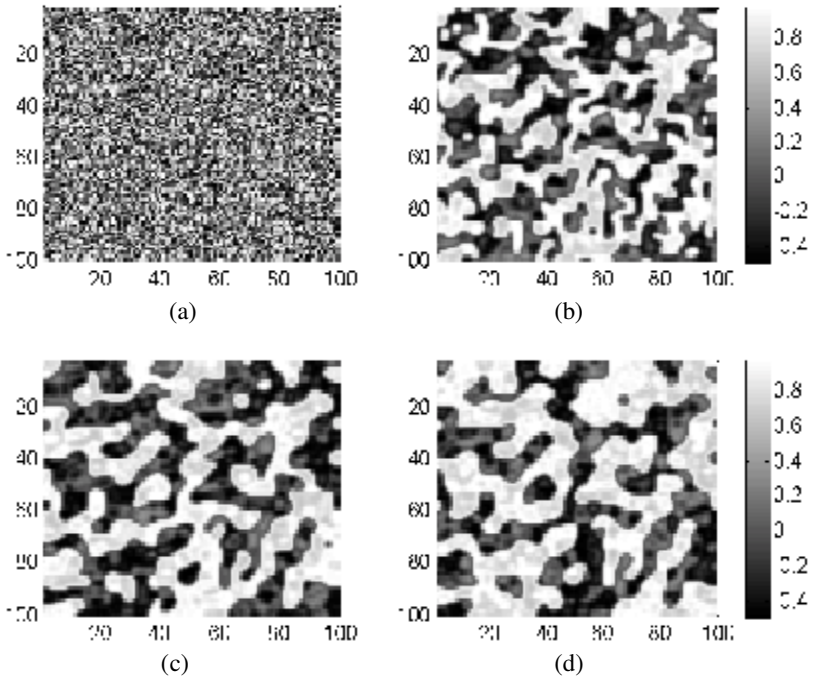

Fig. 1 Snapshot patterns at different time instants for Kaneko's 2-D CML system. (a) t=0; (b) t=25; (c) t=75; (d) t=100.

Table 2 Some conditions involved in the LDWNN modelling for Kaneko's 2-D CML system

Size of the arrays of cells	100×100
Initial condition for $s_{i,j}(0)$, with i,j =1,2, …, 100.	Uniformly distributed on [0,1]
Boundary conditions	Periodic
Neighbourhood cells involved in the state vectors of the model	$x_1=C(i,j)$, $x_2=C(i,j\text{-}1)$, $x_3=C(i,j+1)$, $x_4=C(i\text{-}1,j)$, $x_5=C(i+1,j)$.
mOPP in the OPP algorithm	100
η in the OPP algorithm	10^{-4}
Swarm's size in the PSO algorithm	50
w in the PSO algorithm	Linearly declines from 1 to 0.1
c_1, c_2 in the PSO algorithm	c_1= c_2=2.05
χ in the PSO algorithm	0.7298
mPSO in the PSO algorithm	200
δ in the PSO algorithm	10^{-5}
ε in the FOR algorithm	10^{-10}
λ in the FOR algorithm	10
Wavelet functions	Mexican hat

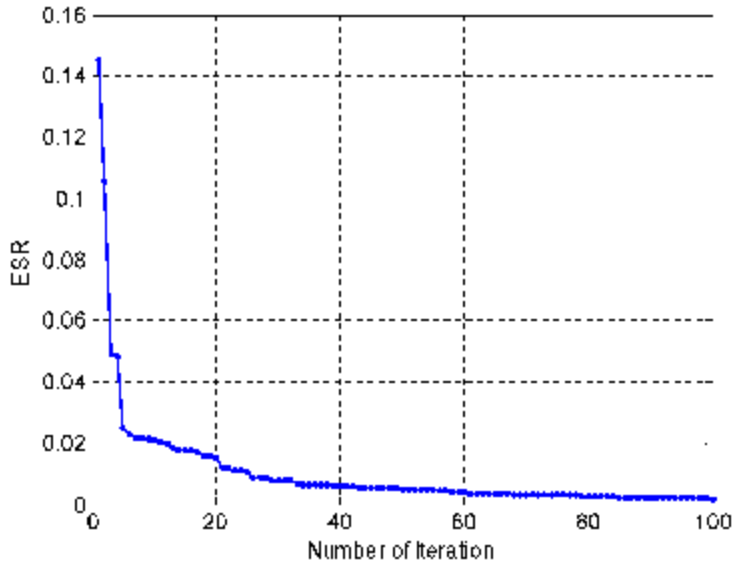

Fig. 2 The error-to-signal ratio (ESR) index calculated by the OPP algorithm for the LDWNN modeling of Kaneko's 2-D CML system.

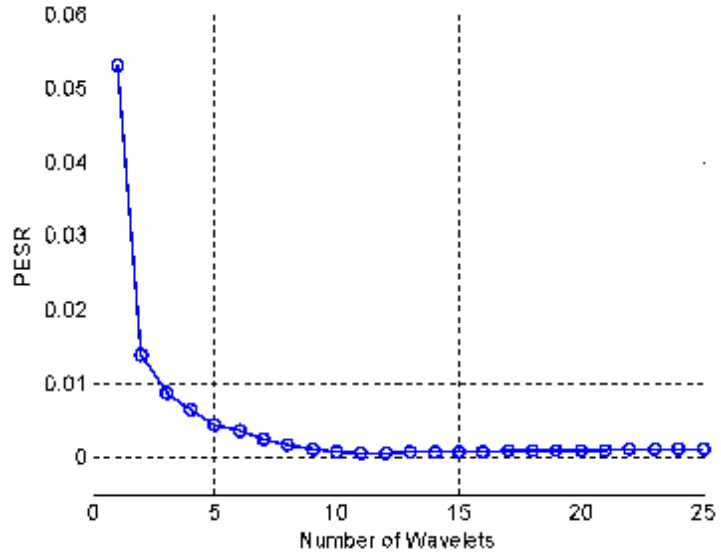

Fig. 3 The penalised error-to-signal ratio (PESR) index calculated by the FOR algorithm for the LDWNN modeling of Kaneko's 2-D CML system.

Table 3 Model parameters and the associated ERR values for the LDWNN modeling of Kaneko's 2-D CML system.

Cell	Parameter			ERR (%)	
	c	a	b	Individual	Sum
$C(i, j)$ $s_{i,j}$	0.779377353142	1.176540901604	0.077788826608	94.7250	99.1753
	0.333764225837	0.215927012031	-0.931464740048	3.9008	
	0.099949271090	3.149905378383	2.039778682928	0.5156	
	-0.363719053503	3.307204181260	4.289126530991	0.0339	
$C(i, j\text{-}1)$ $s_{i,j-1}$	0.097154040977	1.645410684733	-0.185214031788	0.2047	0.2631
	0.054706585000	1.909477838374	0.849633019892	0.0583	
$C(i, j\text{+}1)$ $s_{i,j+1}$	-0.113301597742	2.151296601341	2.079671488586	0.2153	0.2153
$C(i\text{-}1, j)$ $s_{i-1,j}$	0.093035314725	1.070816173200	-0.217875930508	0.1149	0.1499
	-0.085864916016	4.057608672508	4.201270159626	0.0351	
$C(i\text{+}1, j)$ $s_{i+1,j}$	0.020620174266	2.140464718671	0.213686362585	0.0856	0.1476
	-0.118901535729	0.784322249023	1.403515386813	0.0620	
—	—	—	—	99.9513	99.9513

The associated model parameters, along with the error reduction ratio (ERR) of individual wavelets, are given in Table 3. The total run time to produce the model (32), using Matlab (R14) on a Sun-500 workstation (1.28 GHz), was 37.24 minutes. The ERR index shows that the wavelet functions relative to the state variable $s_{i,j}$, corresponding to the *cell*(i,j), dominates the 'output' of the wavelet network model, meaning that the state variable $s_{i,j}$ plays, via relevant wavelet functions, a dominant role, among all the five involved variables, in the associated spatio-temporal evolution. This conclusion is coincident with the original model (31), where the weight coefficient assigned to the function of the state variable $s_{i,j}$ is much larger compared with the others.

To evaluate the performance of the proposed wavelet neural network modelling framework, both the wavelet network model (32) and the original model (31) were simulated, using the same initial condition for $s_{i,j}(0)$, with i, j =1,2, …, 100, which was randomly distributed on [0,1]. But note that the individual initial values for $s_{i,j}(0)$ here were totally different from those used for producing the training data set from the original model (31) (see Fig. 1a), even though they follow, in a statistical sense, the same distribution. The simulation results from the model (32) are referred to as the model predicted output (MPO), which means that, starting from given initial conditions (the initial pattern), the model will produce values to form the next pattern; using the newly produced pattern as the new initial values (no other information is needed), the model will evolve forward a further step in time; subsequent patterns are thus generated step by step. Notice that MPO is a much more sevewre test tan the often used one step ahead (OSA) predicted output since the later can look good even for very poor models. The boundary condition adopted here is the same as described in Table 2. A comparison of the model predicted output from the identified network model (32), with patterns produced by the

original model (31), at time instances t=25, 50, 75, 100, are shown in Fig. 4. It is clear from Fig. 4 that the identified wavelet network model (32) perfectly represents the original model (31).

For a comparison, a wavelet series model of the form (3), where the Mexican hat wavelet was used as the elementary building blocks, was also used to identify the 2-D CML model (31), based on the same training data $\{\mathbf{x}(k), y(k)\}_{k=1,2,\ldots,N}$. The initial wavelet series model involves an over-complete dictionary containing 1000 candidate wavelet basis functions. The finally identified wavelet series model includes 16 dyadic wavelets.

To measure the performance of the identified wavelet models, the local 2-D mean-square-error (LMSE), defined as below, was considered

$$\mathrm{LMSE}(t) = \frac{1}{IJ}\sum_{i=1}^{I}\sum_{j=1}^{J} | s_{i,j}(t) - \hat{s}_{i,j}(t) |^2 \tag{33}$$

where $s_{i,j}(t)$ represent the observations at the time instant t, $\hat{s}_{i,j}(t)$ represent the corresponding predicted values from a given model, and I and J define the size of the associated pattern. The local mean-square-errors, for the model predicted outputs of both the identified LDWNN model (32) and the wavelet series model, are shown in Fig. 5, where the range of the time instants are from 1 to 100. It is clear from Fig.7 that the LDWNN model (32) is superior to the wavelet series model for representing the 2-D Kaneko's CML model (31).

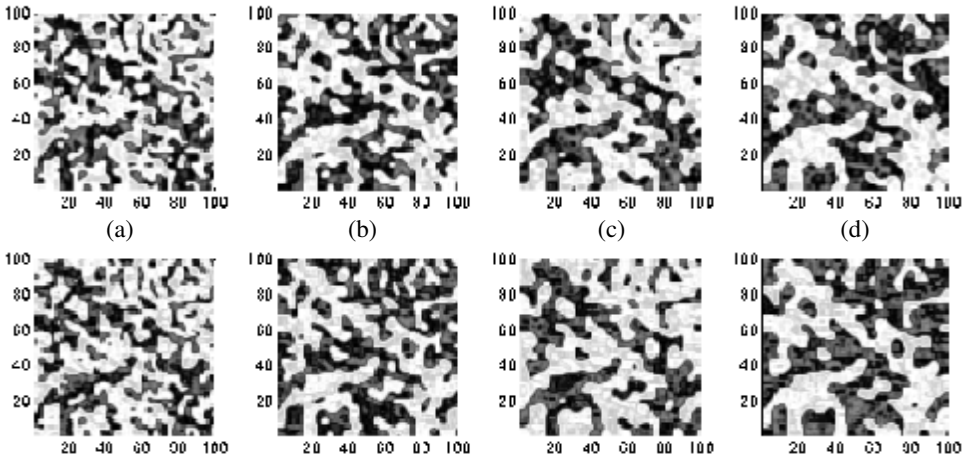

Fig. 4 A compassion of some patterns produced by the original model (31) [(a)-(d)] and the identified LDWNN model (32) [(e)-(h)]. (a)-(d): t=25, 50, 75, 100; (e)-(h) t=25, 50, 75, 100. Note that the values of the initial patterns here are different from those used in Fig. 3.

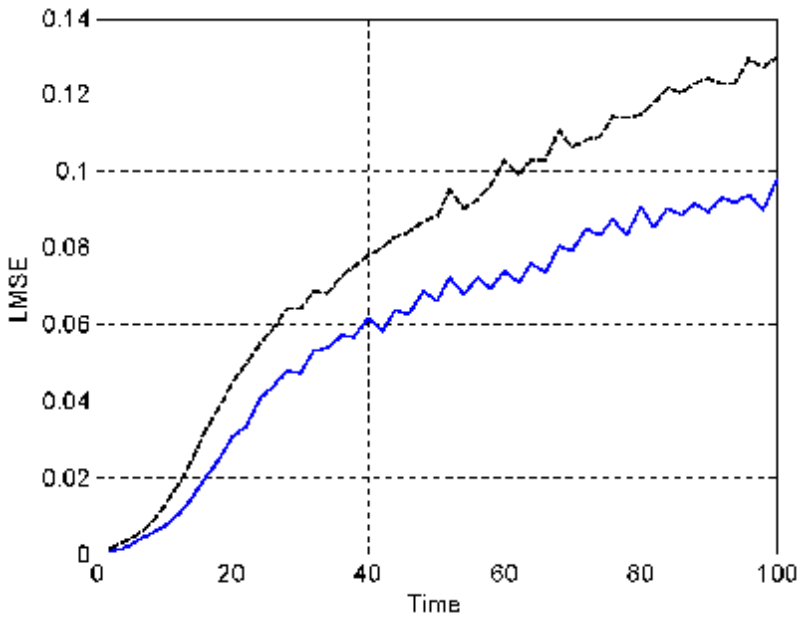

Fig. 5 A compassion of the local mean-square-errors, for model predicted outputs of both the identified LDWNN model (32) and the wavelet series model for Kaneko's 2-D CML system. Solid line is for the LDWNN model and the dashed line is for the wavelet series model.

4.2 Identification of the Belousov-Zhabotinsky (BZ) Reaction

The Belousov-Zhabotinsky (Belousov 1959, Zhabotinsky 1964, Winfree 1972, Kuramoto 1984) reaction, or BZ reaction, as an excitable medium, is an important class of chemical reactions exhibiting a spatio-temporal oscillatory behaviour. As a classical example of nonequilibrium thermodynamics, the BZ reaction provides an interesting chemical model of nonequilibrium biological phenomena, and the modelling and identification of these type of reactions is of extreme interest for theoretical analysis of relevant phenomena.

By adopting the recipe given by Winfree (1972), an experiment resulting in a thin layer BZ reaction was carried out, and a set of images were captured with equal time intervals during the experiment, using a digital video camera that is connected to a PC via a USB socket. The sampled images were pre-processed and saved as patterns with a resolution of 480 by 640 pixels. Some of these patterns are shown in Fig. 6. In this example, the LDWNN modelling framework was applied to these sampled images, and the objective is to demonstrate the applicability and effectiveness of the new network model for the identification of the BZ reaction.

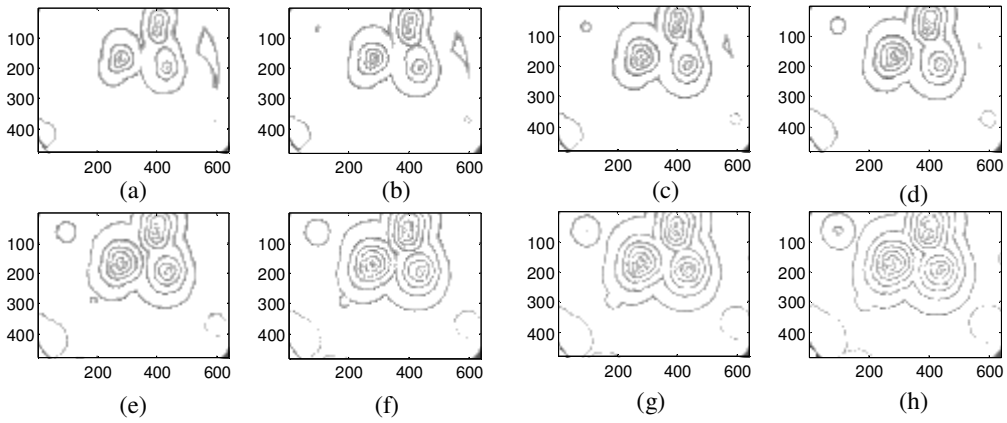

Fig. 6 Some snapshots for the BZ reaction at different time instants. (a) t=10; (b) t=20; (c) t=30; (d) t=40; (e) t=50; (f) t=60; (g) t=70; (h) t=80.

A total of N=2500 data pairs, $\{\mathbf{x}(k), y(k)\}_{k=1,2,\ldots,N}$, were used for the network training, where $y(k)$ represents the value of the relevant central cell at the present time instant, and $\mathbf{x}(k) = [x_1(k), x_2(k), \mathrm{L}\ , x_9(k)]^T$ represent the values of the nine involved cells at a squared lattice with the *Moore neighbourhoods*, at the previous time instant (see Table 4). These 2500 data pairs were formed as follows. Firstly, 5 adjacent pattern pairs were randomly chosen from the first 50 sampled patterns. Secondly, 500 data pairs were randomly chosen in each of these 5 pattern pairs.

The Mexican hat wavelet was used as the elementary building blocks for constructing the wavelet network model. All the experiment conditions involved in the modelling procedure for this example are shown in Table 4. A total of 100 construction functions of the form (9) were optimised during the optimisation procedure using the OPP+PSO algorithm. Significant individual wavelets were then selected from the pool of the form (21), which contains 900 individual candidate wavelets of the form $\psi_{k,j} = \psi(x_k; a_{k,j}, b_{k,j})$, with k=1,2, …, 9 and j=1,2, …, 100, and where both the dilation and translation parameters have already been optimised. The penalised error-to-signal ratio, PESR, produced by the FOR algorithm and shown in Fig. 7, suggests that a total of 15 wavelets should be included in the wavelet network model. The structure of the wavelet neural network model is of the form (30), and the associated parameters of the identified LDWNN model are shown in Table 5. The total run time to produce such a LDWNN model, using Matlab (R14) on a Sun-500 workstation (1.28 GHz), was 49.33 minutes.

To evaluate the performance of the identified LDWNN model, the short-term predictive capability of the model was inspected. Denote the observation of the image (pattern) measured at the present time instant t by $X(t)$. The s-step-ahead prediction, denoted by $\hat{X}(t+s \mid t; f, X(t))$, where f represents

the given identified model, is the iteratively produced result by the model, on the basis of $X(t)$ but without using information on observations for patterns at any other time instants. As an example, the 1-, 2-, 3- and 4-step-ahead predictions, on the basis of the measurements at the time instants t=20, t=60 and t=90 were considered respectively, and these are shown in Figs. 8, 9, and 10. Clearly, the identified model provides good short-term predictions in the sense that these predictions capture the main features of the observed images.

Table 4 Some conditions involved in the LDWNN modelling for the identification of the BZ reaction.

Size of the arrays of cells	480×640
Neighbourhood cells involved in the state vectors of the model	$x_1=C(i\text{-}1, j\text{-}1)$, $x_2=C(i\text{-}1, j)$, $x_3=C(i\text{-}1, j+1)$, $x_4=C(i, j\text{-}1)$, $x_5=C(i, j)$, $x_6=C(i, j+1)$, $x_7=C(i+1,j\text{-}1)$, $x_8=C(i+1,j)$, $x_9=C(i+1,j+1)$.
mOPP in the OPP algorithm	100
η in the OPP algorithm	10^{-4}
Swarm's size in the PSO algorithm	50
w in the PSO algorithm	Linearly declines from 1 to 0.1
c_1, c_2 in the PSO algorithm	$c_1= c_2$=2.05
χ in the PSO algorithm	0.7298
mPSO in the PSO algorithm	300
δ in the PSO algorithm	10^{-5}
ε in the FOR algorithm	10^{-10}
λ in the FOR algorithm	10
Wavelet functions	Mexican hat

Table 5 Model parameters and the associated ERR values for the LDWNN modeling of the BZ reaction.

Cell	Parameter			ERR (%)	
	c	a	b	Individual	Sum
$C(i\text{-}1, j\text{-}1)$ $s_{i\text{-}1,j\text{-}1}$	-0.713603169209	1.258349616681	-0.503348315305	1.5022	1.5022
$C(i\text{-}1, j)$ $s_{i\text{-}1,j}$	-0.486569276355	35.612505232244	20.463158789644	0.0164	0.0436
	-1.075032700406	133.933972890732	89.387255141179	0.0147	
	1.632422695371	29.859288386832	11.764047566000	0.0125	
$C(i\text{-}1, j+1)$ $s_{i\text{-}1,j+1}$	-0.758816360524	0.789781622826	-0.887347297004	0.0372	0.0508
	0.270962699699	24.353507058447	12.543692700367	0.0136	
$C(i, j\text{-}1)$ $s_{i,j\text{-}1}$	—	—	—	—	—
$C(i, j)$ $s_{i,j}$	-0.223970914586	10.886848715425	7.741959531790	0.0186	0.0322
	-0.416193555114	44.526215096110	25.171768052370	0.0136	
$C(i, j+1)$ $s_{i,j+1}$	0.026906263378	0.233652027575	0.909961362936	96.2540	96.2540
$C(i+1, j\text{-}1)$ $s_{i+1,j\text{-}1}$	1.084267775435	59.731312786112	31.308369474603	0.2079	0.2168
	0.179508477989	24.814788778609	18.812945138697	0.0043	
	-0.436716328775	51.648192913951	32.364962790656	0.0046	
$C(i+1, j)$ $s_{i+1,j}$	-0.413383510157	1.027215053431	-0.503348315305	1.4022	1.4022
$C(i+1, j+1)$ $s_{i+1,j+1}$	-0.373501561261	21.173951646665	17.652079735464	0.0171	0.0300
	0.195440741994	0.885029752728	1.098194085103	0.0129	
—	—	—	—	99.5390	99.5390

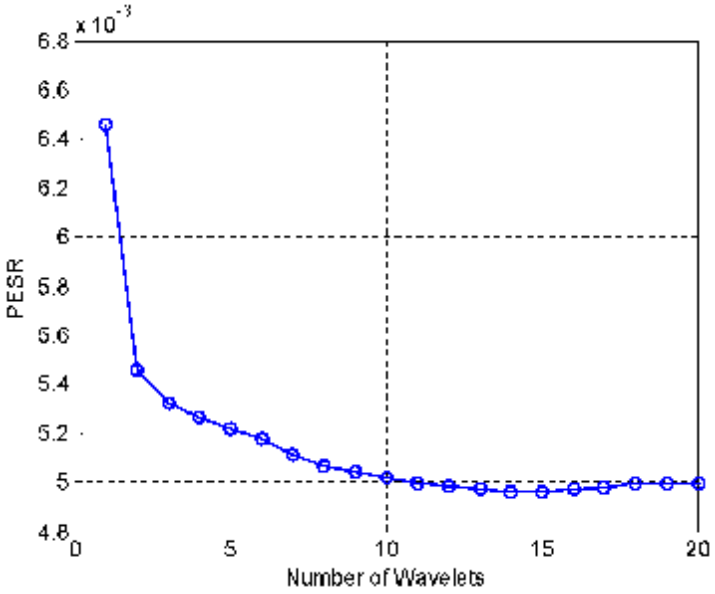

Fig.7 The penalised error-to-signal ratio (PESR) index calculated by the FOR algorithm for the LDWNN modeling of the BZ reaction.

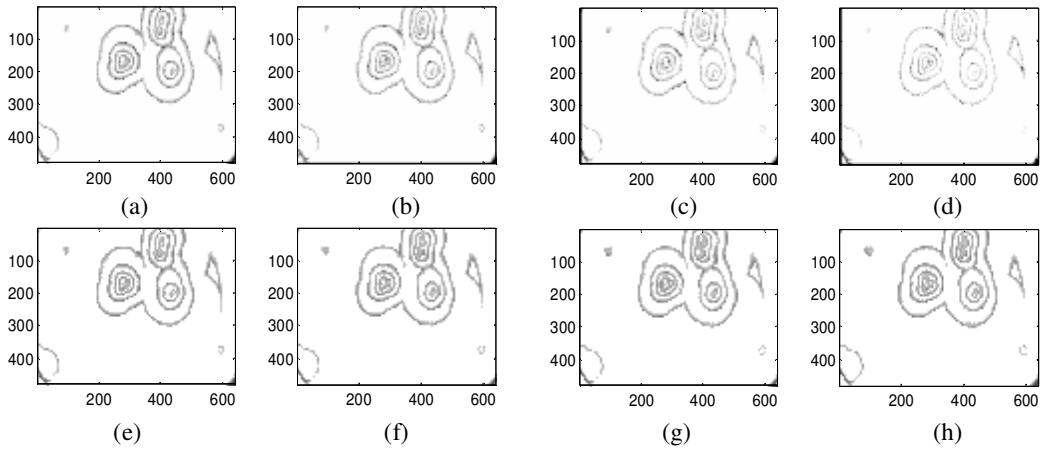

Fig. 8 The 1-, 2-, 3- and 4-step-ahead predictions, on the basis of the observation at the time instant t=20, for the BZ reaction. (a) 1-step; (b) 2-step; (c) 3-step; (d) 4-step ; (e) true measurement for (a); (f) true measurement for (b); (g) true measurement for (c); (h) true measurement for (d).

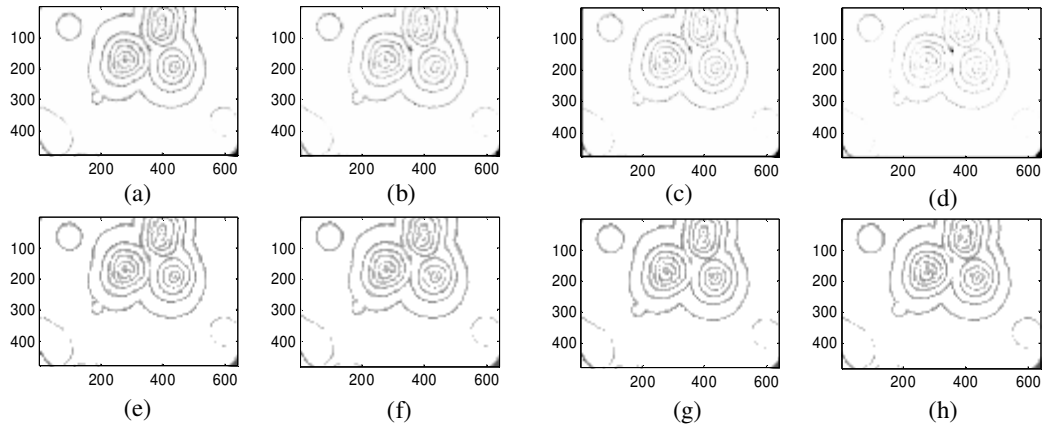

Fig. 9 The 1-, 2-, 3- and 4-step-ahead predictions, on the basis of the observation at the time instant t=60, for the BZ reaction. (a) 1-step; (b) 2-step; (c) 3-step; (d) 4-step ; (e) true measurement for (a); (f) true measurement for (b); (g) true measurement for (c); (h) true measurement for (d).

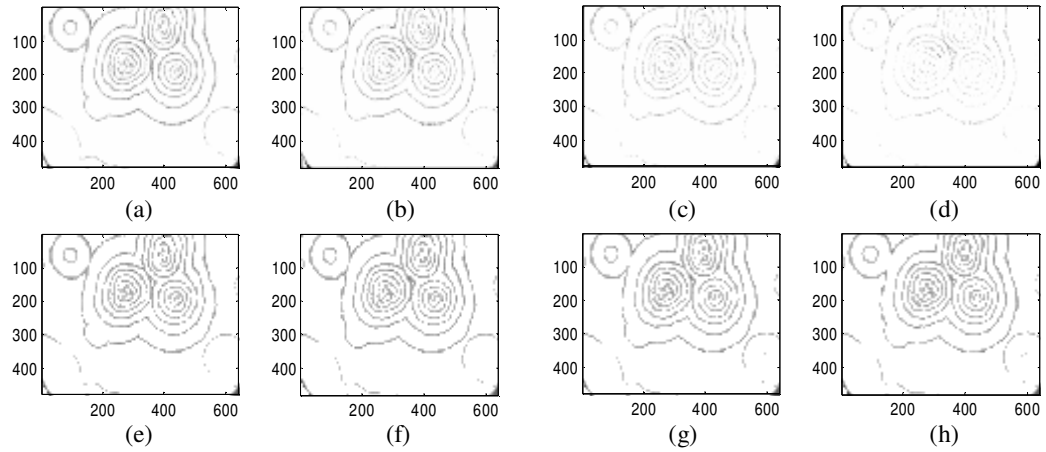

Fig. 10 The 1-, 2-, 3- and 4-step-ahead predictions, on the basis of the observation at the time instant t=90, for the BZ reaction. (a) 1-step; (b) 2-step; (c) 3-step; (d) 4-step ; (e) true measurement for (a); (f) true measurement for (b); (g) true measurement for (c); (h) true measurement for (d).

5. Conclusions

A novel two-stage training scheme has been proposed for constructing a new class of lattice dynamical wavelet neural networks (LDWNN). It has been demonstrated, using both artificial and real data, that the proposed LDWNN model is effective for spatio-temporal system identification. The proposed network possesses a few desirable features, for example, the network is almost self-implemented, meaning that by starting with some given conditions (initial, boundary and termination), all within-network parameters can be estimated and calculated by the proposed algorithms; the network provides a transparent model, where individual wavelet-neurons are explicitly available.

The main drawback of the new network is perhaps the computation time, which is mainly spent on the nonlinear optimisation procedure using the PSO algorithm. However, by introducing the PSO algorithm, which is easy to implement, the calculation of gradients required by classical nonlinear optimisation algorithms can now be avoided; this makes the LDWNN very suitable for complex identification problems where relevant object functions may not be differentiable or relevant gradients are very difficult to obtain. This means that wavelets, which are not smooth or even not differentiable, may also be chosen as the elementary building blocks. In fact, many other functions, even though they themselves are not 'wavelets' in the strict sense, can also be used as elementary building blocks, if there is strong evidence that these functions possess desirable properties and can lead to a good model for a given modelling problem.

Acknowledgements

The authors gratefully acknowledge that this work was supported by Engineering and Physical Sciences Research Council (EPSRC), U.K. They gratefully acknowledge the help from Dr A. F. Routh who supervised the B-Z experiments.

References

M. Aerts, G. Claeskens, and M. P. Wand MP, "Some theory for penalized additive models," *J. Statist. Plann. Infer.*, vol. 103, pp. 455-470, Apr. 2002

A. M. Albano, N. B. Abraham, P. E. Rapp, A. Passamante, F. H. Busse, and C. K. R. T. Jones (ed.), *Measures of Spatio-Temporal Dynamics*, *Proceedings of the Workshop on Measures of Spatio-Temporal Dynamics*. Bryn Mawr College, Bryn Mawr, Pa, USA June 11-13, 1995.

L. Astic, V. Pellier-Monnin, and F. Godinot, "Spatio-temporal patterns of ensheathing cell differentiation in the rat olfactory system during development," *Neuroscience*, vol. 84, no.1, pp. 295-307, May 1998.

D. Aydogan, A. Elmas, A. M. Albora, and O. N. Ucan, "A new approach to the structural features of the Aegean Sea: Cellular neural network," *Marine Geophysical Researches* vo. 26, no. 1, pp. 1-15, Mar. 2005.

B. Bakshi and G. Stephanopoulos, "Wave-Net: a multiresolution, hierarchial neural network with localized learning," *AIChE Journal*, vol. 39, no. 1, pp. 57-81, Jan. 1993.

J. Bascompte and R. V. Sole (ed.), 1998, *Modelling Spatiotemporal Dynamics in Ecology*. Berlin: Springer, 1998.

B.P. Belousov, "A periodic reaction and its mechanism," in Collection of Short Papers on Radiation Medicine (in Russian), Medgiz, Moscow, pp.145-152, 1959.

R. Berezney, K. S. Malyavantham, A. Pliss, S. Bhattacharya, and R. Acharya, " Spatio-temporal dynamics of genomic organization and function in the mammalian cell nucleus," *Advances In Enzyme Regulation*, vol. 45, pp.17-26, 2005.

S. A. Billings, S. Chen, and M. J. Korenberg, "Identification of MIMO nonlinear systems suing a forward regression orthogonal estimator," *Int. J. Control*, vol. 49, no.6, pp.2157-2189, Jun.1989.

S. A. Billings and D. Coca, "Discrete wavelet models for identification and qualitative analysis of chaotic systems," *Int. J. Bifurcat. Chaos*, vol. 9, no.7, pp.1263-1284, July 1999.

S.A. Billings and D. Coca, ''Identification of coupled map lattice models of deterministic distributed parameter systems,'' *Int. J. Sys. Sci*., 33, pp. 623–634, 2002.

S. A. Billings, L. Z. Guo, and H. L. Wei, "Identification of coupled map lattice models for spatio-temporal patterns using wavelets," *Int. J. Control*, vol.14, No. 14, pp. 1021-1038, Nov 2005.

S. A. Billings and H. L. Wei, "The wavelet-NARMAX representation: A hybrid model structure combining polynomial models with multiresolution wavelet decompositions," *Int. J. Syst. Sci.*, vol. 36, no.3, pp. 137-152, Feb. 2005a.

S. A. Billings and H. L. Wei, "A new class of wavelet networks for nonlinear system identification," *IEEE Trans. Neural Networks*, 16, no.4, pp. 862-874, July 2005b.

S. A. Billings and H. L. Wei, "An adaptive orthogonal search algorithm for model subset selection and nonlinear system identification," *Int. J. Control*, 2007a (in press).

S. A. Billings and H. L. Wei, "Sparse model identification using a forward orthogonal regression algorithm aided by mutual information," *IEEE Trans. Neural Networks*, vol.18, no.1, pp. 306-310, Jan. 2007b.

S.A. Billings and Y. Y. Yang, ''Identification of the neighbourhood and CA Rules from Spatio-temporal CA patterns,'' *IEEE Trans. Syst. Man Cybern. B*, 33, pp. 332–339, 2003.

A. Brezger and S. Lang, "Generalized structured additive regression based on Bayesian P-splines," *J. Comput. Statist. Data Anal*., vol.50, no.4, pp. 967–991, Feb. 2006.

S. Chen, S. A. Billings and W. Luo, "Orthogonal least squares methods and their application to nonlinear system identification," *Int. J. Control*, vol. 50, no. 5, pp. 1873–1896, Nov. 1989.

S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least-squares learning algorithm for radial basis function networks," *IEEE Trans. Neural Networks.*, vol. 2, no. 2, pp. 302–309, Mar. 1991.

S. Chen, X. Hong, and C. J. Harris, "Sparse kernel regression modeling using combined locally regularized orthogonal least squares and D-optimality experimental design," *IEEE Trans.*

Automatic Control, vol. 48, no. 6, pp. 1029-1036, June 2003.

S. Chen, X. Hong, C. J. Harris, and P. M. Sharkey, "Sparse modeling using orthogonal forward regression with press statistic and regularization," *IEEE Trans. Syst., Man, Cybern., B: Cybern.*, vol. 34, no. 2, pp. 898–911, Apr. 2004.

Z. S. Chen and C. Y. Ji, "Spatial-temporal modeling of malware propagation in networks," *IEEE Trans. Neural Networ*., vol.16, no.5, pp.1291-1303, Sep. 2005.

Y. H. Chen, B. Yang and J. W. Dong, "Time-series prediction using a local linear wavelet neural network," *Neurocomputing*, vol. 69, pp.449-465, Jan. 2006.

S.-N. Chow and J. Mallet-Paret, "Pattern formation and spatial chaos in lattice dynamical systems-Part I," *EEE Trans. Circuits Syst.,* vol. 42, no.10, pp. 746-751, Oct. 1995.

L. O. Chua and L. Yang, "Cellular neural networks: Theory," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 35, no. 12, pp. 1257–1272, Dec. 1988a.

L. O. Chua and L. Yang, "Cellular neural networks: Applications," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 35, no. 12, pp. 1273–1290, Dec. 1988b.

L. O. Chua and T. Roska, *Cellular Neural Networks: Foundations and Applications*. Cambridge: Cambridge University Press, 2001.

M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," *IEEE Trans. Evol. Comput.*, vol. 6, no.1, pp. 58–73, Feb. 2002.

D. Coca, and S. A. Billings, "Continuous-time system identification for linear and nonlinear system identification using wavelet decompositions," *Int. J. Bifurcat. Chaos*, vol. 7, no.1, pp. 87-96, Jan. 1997.

D. Coca and S. A. Billings, "Identification of coupled map lattice models of complex spatio-temporal pattern," *Phys. Lett.*, vol. A287, pp. 65–73, 2001.

T. Czaran, *Spatiotemporal Models of Population and Community Dynamics*. London: Chapman & Hall, 1998.

I. Daubechies, *Ten Lectures on Wavelets*. Philaelphia, Pennsylvania: Society for Industrial and Applied Mathematics, 1992.

B. Delyon, A. Juditsky and A. Beneveniste, "Accuracy analysis for wavelet approximations," *IEEE Trans. Neural Networks*, vol. 6, no. 2, pp. 332–348, Mar. 1995.

D. S. Dimitrova and R. Berezney, "The spatio-temporal organization of DNA replication sites is identical in primary, immortalized and transformed mammalian cells," *J. Cell Science*, vol. 115, no. 21, pp.4037-4051, Nov. 2002.

Y. Dolak and C. Schmeiser, "Kinetic models for chemotaxis: Hydrodynamic limits and spatio-temporal mechanisms," *J. Math. Bio*., vol.51, no.6, pp.595-615. Dec. 2005.

R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6 th Symp. Micro Mach. Human Sci.*, pp. 39-43, Nagoya, Japan, Oct. 4-6, 1995.

S. E. Fahlman and C. Lebiere, "The cascade-correlation learning architecture," in *Advances in Neural Information Processing Systems 2*, D. S. Touretzky, Ed. San Mateo, CA: Morgan Kaufmann, pp. 524–532, 1990.

J. H. Friedman and W. Stuetzle, "Projection pursuit regression," *J. Amer. Statist. Assoc.*, vol. 76, no. 376, pp.817–823, Dec.1981.

L. Z. Guo and S. A. Billings, "Identification of coupled map lattice models of stochastic spatio-temporal dynamics using wavelets," *Dyn. Syst.*, vol. 19, pp. 265–278, 2004.

T. J. Hastie and R. J. Tibshirani, *Generalized Additive Models*. London: Chapman & Hall, 1990.

S. Haykin, *Neural networks: A Comprehensive Foundation* (2nd Ed). New York: Macmillan: Maxwell Macmillan International, 1999.

D. W. C. Ho, J. M. Li and Y.G. Niu, "Adaptive neural control for a class of nonlinearly parametric time-delay systems," *IEEE Trans. Neural Networks*, vol. 16, no.3, pp. 625-635, May 2005.

C. F. Hsu, C. M. Lin and T. T. Lee, "Wavelet adaptive backstepping control for a class of nonlinear systems," *IEEE Trans. Neural Networks*, vol. 17, no.5, pp.1175-1183, Sep. 2006.

G. B. Huang, L. Chen, and C. K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Trans. Neural Netw.*, vol. 17, no. 4, pp. 879–892, Jul. 2006.

C. S. Huang and W. C. Su, "Identification of modal parameters of a time invariant linear system by continuous wavelet transformation," *Mech Syst Signal Process*, vol. 21, no.4, pp. 1642-1664, May 2007.

J. N. Hwang, S. R. Lay, M.Maechler, R. D. Martin, and J. Schimert, "Regression modeling in back-propagation and projection pursuit learning," *IEEE Trans. Neural Networks*, vol. 5, no. 3, pp. 342–353, May 1994.

B. Jahne, *Spatio-Temporal Image Processing: Theory and Scientific Applications*. Berlin: Springer-Verlag, 1993.

L. K. Jones, "A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training," *Ann. Statist.*, vol. 20, no. 1, pp. 608–613, 1992.

A. Juditsky, H. Hjalmarsson, A. Benveniste, B. Delyon, L. Ljung, J. Sjoberg, and Q. H. Zhang, "Nonlinear black-box models in system identification: Mathematical foundations," *Automatica*, vol.31, no.12, pp.1725-1750, Dec. 1995.

K. Kaneko, "Spatio-temporal chaos in one- and two-dimensional coupled map lattices," *Physica D*, vol. 37, pp. 60-82, 1989.

K. Kaneko, *Theory and Application of Coupled Map Lattices*. New York: Wiley, 1993.

J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proc.IEEE Int. Conf. Neural Networks*, vol. IV, pp.1942–1948, Perth, Australia, 1995.

J. Kennedy, R. C. Eberhart and Y. Shi, *Swarm Intelligence*, San Francisco: Morgan Kaufmann

Publishers, 2001.

Y. Kuramoto, *Chemical Oscillations*, *Waves*, *and Turbulence*. Berlin: Springer, 1984.

T. Y. Kwok and D. Y. Yeung, "Constructive algorithms for structure learning in feedforward neural networks for regression problems," *IEEE Trans. Neural Netw.*, vol. 8, no. 3, pp. 630–645, May 1997a.

T. Y. Kwok and D. Y. Yeung, "Objective functions for training new hidden units in constructive neural networks," *IEEE Trans. Neural Netw.*, vol. 8, no. 5, pp. 1131–1148, Sep. 1997b.

M. J. Lado, C. Cadarso-Suarez, J. Roca-Pardinas, and P. G. Tahoces, "Using generalized additive models for construction of nonlinear classifiers in computer-aided diagnosis systems," *IEEE Trans. Inf. Technol. Biomed.*, vol. 10, no. 2, pp. 246–253, Apr. 2006.

H. Leung, G. Hennessey, and A. Drosopoulos, "Signal detection using the radial basis function coupled map lattice," *IEEE Trans. Neural Netw.*, vol. 11, no. 5, pp. 1133–1151, Oct. 2000.

C. J. Lin and C. C. Chin, "Prediction and identification using wavelet-based recurrent fuzzy neural networks," *IEEE Trans. Syst. Man, Cyber. B*, vol.34, no.5, pp.2144-2154, Oct. 2004.

F. J. Lin, H. J. Shieh and P. K. Huang, "Adaptive wavelet neural network control with hysteresis estimation for piezo-positioning mechanism," *IEEE Trans. Neural Networks*, vol. 17, no.2, pp.432-444, Mar. 2006.

G. P. Liu, *Nonlinear Identification and Control: A Neural Network Approach*. Berlin: Springer-Verlag, 2001.

S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, pp. 674–693, Jul. 1989.

S. Mallat and Z. Zhang, "Matching pursuit with time-frequency dictionaries," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.

S. Mandelj, I. Grabec and E. Govekar, ''Statistical approach to modeling of spatiotemporal dynamics," *Int. J. Bifurcation and Chaos*, vol. 11, pp. 2731–2738, 2001.

P. Marcos-Nikolaus, J.M. Martin-Gonzalez, and R.V. So le, ''Spatial forecasting: detecting determinism from single snapshots'', *Int. J. Bifurcation and Chaos*, vol. 12, pp. 369–376, 2002.

M. Ohtaki, H. Honjo, and H. Sakaguchi, "A coupled map lattice model for dendrite in diffusion field," *J. Crystal Growth*, vol. 237, pp.159-163, Apr. 2002.

Y. Oussar, I. Rivals, L. Personnaz and G. Dreyfus, "Training wavelet networks for nonlinear dynamic input-output modelling," *Neurocomputing*, vol. 20, pp. 173-188, Aug. 1998.

U. Parlitz and C. Merkwirth, ''Prediction of spatiotemporal time series based on reconstructed local states'', *Phys. Rev. Lett.*, vol.84, pp. 2820–2823, 2000.

K. E. Parsopoulos and M. N. Vrahatis, "On the computation of all global minimizes through particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no.3, pp. 211–224, June 2004.

Y. C. Pati, and P. S. Krishnaprasad, "Analysis and synthesis of feedforward neural networks using discrete affine wavelet transforms," *IEEE Trans. Neural Networks*, vol. 4, no.1, pp.73-85, Jan. 1993.

D. Raabe, "Cellular automata in materials science with particular reference to recrystallization simulation," *Ann. Rev. Mater. Res.*, vol. 32, pp. 53-76, 2002.

R. D. Reed and R. J. Marks II, *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*. Cambridge, MA: The MIT Press, 1999.

E. A. Rying, G. L. Bilbro and J. C. Lu, "Focused local learning with wavelet neural networks," *IEEE Trans. Neural Networks*, vol. 13, pp.304-319, March 2002.

R. M. Sanner and J. J. E. Slotine, "Structurally dynamic wavelet networks for adaptive control of robotic systems," *Int. J. Control*, vol. 70, no.3, June 1998.

Y. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization," in *Lecture Notes In Computer Science*, V. W. Porto, N. Saravanan, D. Waagen, and A. E. Eiben (Eds), vol. 1447, pp. 591-600, 1998a.

Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer," in *Proc. IEEE Conf. Evolutionary Computation*, pp. 69-73, Anchorage, AK, USA, 4th -9th May, 1998b.

F. L. Silva, J. C. Principe, and L. B. Almeida (ed.), *Spatiotemporal Models in Biological and Artificial Systems*. Washington: IOS Press, 1997.

A. Sitz, J. Kurths, and H. U. Voss, "Identification of nonlinear spatio-temporal systems via partitioned filtering," *Physical Review E*, vol. 68, 016202, 2003.

H. Spors and A. Grinvald, "Spatio-temporal dynamics of odor representations in the mammalian olfactory bulb," *Neuron*, vol. 34, no.2, pp.301-315, Apr. 2002.

M. Unser, "Approximation power of biorthogonal wavelet expansions," *IEEE Trans. Signal Process.*, vol. 44, pp519-527, Mar. 1996.

F. van den Bergh and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE Trans. Evolutionary Computation*, vol. 8, no.3, pp.225-239, June 2004.

R. J. Wai and H. H. Chang, "Backstepping wavelet neural network control for indirect field-oriented induction motor drive," *IEEE Trans. Neural Networks*, vol. 15, no. 2, pp. 367-382, March 2003.

H. L. Wei and S. A. Billings, "A unified wavelet-based modelling framework for nonlinear system identification: the WANARX model structure," *Int. J. Control*, vol.77, no.4, pp.351-366, Mar. 2004a.

H. L. Wei and S. A. Billings, "Identification and reconstruction of chaotic systems using multiresolution wavelet decompositions," *Int. J. Syst. Sci*, vol. 35, no.9, pp. 511-526, July 2004b.

H. L. Wei and S. A. Billings, "Long term prediction of nonlinear time series using multiresolution wavelet models," *Int. J. Control*, vol. 79, no.6, pp. 569-580, June 2006.

H. L. Wei, S. A. Billings and M. A. Balikhin, "Wavelet based nonparametric NARX models for nonlinear input-output system identification," *Int. J. Syst. Sci*, vol. 37, no.15, pp.1089-1096, Dec. 2006.

A. T. Winfree, "Spiral waves of chemical activity," Science, vol. 175, no.4022, pp. 634-636, 1972.

S. Wolfram, *Cellular Automata and Complexity*. New York: Addison-Wesley, 1994.

S. N. Wood, "Stable and efficient multiple smoothing parameter estimation for generalized additive models," *J. Aerica. Statist.Assoc.*, vol. 99, no. 467, pp.673-686, Sep. 2004.

Y. S. Xia and H. Leung, "Nonlinear spatial-temporal prediction based on optimal fusion," *IEEE Trans. Neural Netw.*, vol. 17, no. 4, pp. 975–988, July 2006.

J. H. Xu and D. W. C. Ho, "A basis selection algorithm for wavelet neural networks," *Neurocomputing*, vol. 48, pp. 681-689, 2002.

J. X. Xu and Y. Tan, "Nonlinear adaptive wavelet control using constructive wavelet networks," *IEEE Trans. Neural Networks*, vol. 18, no.1, pp. 115-127, Jan. 2007.

A.M. Zhabotinsky, Periodic liquid phase reactions. Proc. Acad. Sci. USSR, 157, pp.392-395, 1964.

J. Zhang, G. G. Walter, Y. B. Miao and W.N.W. Lee, "Wavelet neural networks for function learning," *IEEE Trans. Signal Processing*, vol. 43, no.6, pp. 1485-1497, June 1995.

Q. Zhang, and A. Benveniste, "Wavelet networks," *IEEE Trans. Neural Networks*, vol. 3, no.6, pp. 889-898, Nov. 1992.

Q. Zhang, "Using wavelet network in nonparametric estimation," *IEEE Trans. Neural Networks*, vol. 8, no.2, pp. 227-236, 1997.

Z. Zhang, *Matching Pursuit*. PhD dissertation, New York University, 1993.