This is the prepublication draft of the paper published in
International Journal of Bifurcation and Chaos, Vol. 18, No. 12, pp.3611–3624, 2008.

Generalised Cellular Neural Networks (GCNNs) Constructed Using Particle Swarm Optimisation for Spatio-Temporal Evolutionary Pattern Identification

H. L. Wei and S. A. Billings*

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: Particle swarm optimisation (PSO) is introduced to implement a new constructive learning algorithm for training generalised cellular neural networks (GCNNs) for the identification of spatio-temporal evolutionary (STE) systems. The basic idea of the new PSO-based learning algorithm is to successively approximate the desired signal by progressively pursuing relevant orthogonal projections. This new algorithm will thus be referred to as the orthogonal projection pursuit (OPP) algorithm, which is in mechanism similar to the conventional projection pursuit approach. A novel two-stage hybrid training scheme is proposed for constructing a parsimonious GCNN model. In the first stage, the orthogonal projection pursuit algorithm is applied to adaptively and successively augment the network, where adjustable parameters of the associated units are optimised using a particle swarm optimiser. The resultant network model produced at the first stage may be redundant. In the second stage, a forward orthogonal regression (FOR) algorithm, aided by mutual information estimation, is applied to refine and improve the initially trained network. The effectiveness and performance of the proposed method is validated by applying the new modelling framework to a spatio-temporal evolutionary system identification problem.

Keywords: Cellular neural networks; coupled map lattices; evolutionary algorithms; mutual information; neural networks; orthogonal least squares; parameter estimation; particle swarm optimisation; spatio-temporal evolutionary systems.

1. Introduction

Cellular neural networks (CNNs) [Chua & Yang, 1988a, 1988b; Chua & Roska, 1993, 2002; Roska & Chua, 1993] are a class of nonlinear continuous computing & processor arrays that are well suited for signal and image processing. During past decades, CNNs have been widely investigated for both static and dynamic image processing applications, see for example the work by Roska and Chua [1992], Crounse et al. [1993, 1996], Chua et al. [1995], Crounse and Chua [1995], Thiran et al. [1995], Venetianer et al. [1995], Stoffels et al. [1996]; see also the recently published papers by Lin and Yang [2002], Sbitnev and Chua [2002], Itoh and Chua [2003, 2005], Morfu and Comte [2004], Shi [2004], Chen et al. [2005], Cai and Min [2005], Kanakov et al. [2006], and He and Chen (2006). The mathematical representation of CNNs consists of a large set of coupled nonlinear ordinary differential equations (ODEs) that may exhibit a rich spatio-temporal dynamics [Gilli et al., 2002]. It has been shown that CNN dynamics present a broader class of behaviours than PDEs, and the equivalence between discrete-space CNN models and continuous-space PDE models has been rigorously investigated in Gilli et al. [2002].

Spatio-temporal evolutionary (STE) phenomena widely exist in various areas of science and engineering [Kaneko, 1993; Adamatzky, 1994; Chua & Roska, 2002; Wolfram, 2002]. One salient feature of STE systems, compared with classical pure temporal signals or static images, is that there exists an inherent evolution law that determines the evolution procedure of an STE system. The individual value of a state at a local position of the current pattern, at the present time instant, is dependent on individual values at several local positions of one or more previous patterns. In many cases, the evolution law of a real world STE system is unknown and needs to be identified from observed patterns. Compared to classical pure temporal signal modelling and static image processing, the identification and modelling of high dimensional STE systems is much more challenging.

This study considers the identification problem for STE systems, and the objective is to introduce a novel automatic and systematic identification procedure that can be used to effectively identify, from available observations, the evolution dynamics of an STE system, by constructing a class of discrete-time generalised CNN (GCNN) models. The construction procedure of the GCNN model is composed of two stages. In the first stage, a new constructive learning method, called the orthogonal projection pursuit (OPP), implemented with a particle swarm optimisation (PSO) algorithm, is used to form an initial coarse GCNN model, by recruiting a number of optimised basis functions into the model. The coarse model produced by the OPP learning algorithm may be redundant. Thus, in the second stage, a forward orthogonal regression (FOR) learning algorithm [Billings & Wei, 2007a; Wei & Billings, 2007], implemented using a mutual information estimation method, is then applied to refine and improve the initially obtained GCNN model by removing redundant basis functions from the model.

The construction of the GCNN model involves solving some nonlinear-in-the-parameters problems. Traditionally, Gauss-Newton type nonlinear optimisation methods are often applied to estimate the

unknown model parameters, with a stipulation that the gradient of the associated object functions are differentiable and easy to explicitly calculate. In this study, however, the recently developed particle swarm optimisation (PSO) algorithm [Eberhart & Kennedy, 1995; Kennedy & Eberhart, 1995] is employed as an alternative to solve the associated nonlinear optimisation problem where the objective function is not differentiable. Compared with classical nonlinear least squares algorithms, the PSO algorithm, as a population-based evolutionary method, possesses several desirable attractive properties, for example, this type of algorithm is easy to implement but quite efficient in dealing with a wide class of nonlinear optimisation problems. 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.

The paper is organised as follows. In section 2, the architecture of the GCNN model is presented. In section 3, a two-stage hybrid training scheme, involving both the OPP+PSO approach and a forward orthogonal regression algorithm, is described in detail. In section 4, an example is presented to demonstrate the application of new modelling framework. Finally, the work is summarised in section 5.

2. The Architecture of the GCNN Model

In this study, the 2-D case, which has obvious physical meaning and which is widely applied in practice, is taken as an example to illustrate the construction procedure of the GCNN model. It is known that space-invariant CNN models are widely applied to describe real world problems in most applications [Chua & Roska, 2002]. The discrete-time counterpart of the standard space-invariant CNN representation will thus be employed as the elementary building block to construct the GCNN model, where a number of optimised discrete-time CNN cell blocks, which are used as the basis functions, are superposed and integrated to represent a given STE system.

2.1 The discrete-time CNN cell model

Assume that the 2-D image or pattern produced by an STE system, at the time instant t, consists of a $I \times J$ rectangular array of cells, $C^t(i,j)$, with Cartesian coordinates (i,j), i=1,2, …, I, j=1,2, …, J. Following Chua and Roska [2002], let $S_r^t(i,j)$ be the sphere of influence of the radius r of cell $C^t(i,j)$, at the time instant t, defined as

$$S_r^t(i,j) = \{C^t(i,j) : \max_{1 \le p \le I, 1 \le q \le J} \{|i-p|, |j-q|\} \le r\} \tag{1}$$

where t=1,2, …, i=1,2, …, I, j=1,2, …, J, and r is a non-negative integer number indicating how many neighborhood cells are involved in the evolution procedure. The sphere $S_r^t(i,j)$ is sometimes referred to as the $(2r+1)\times(2r+1)$ neighbourhood. Let $s_{i,j}(t) \in \mathbb{R}$ be the state variable representing the cell

$C^t(i,j) \in S_r^t(i,j)$. From the definition of $S_r^t(i,j)$, a total of $(2r+1)^2$ state variables are involved in (1), see Table 1, where the symbol $C(i,j)$ will be used to indicate cells at an arbitrary evolution time instant.

Let $s_{i,j}(t)$ be the (i,j)th cell to be updated at time t. The discrete-time, discrete-space and continuous-state CNN cell model is given below

$$\begin{aligned} s_{i,j}(t) = c_0 &+ \sum_{C^{t-1}(p,q)\in S_r^{t-1}(i,j)} A^{(1)}(i-p,j-q)y_{p,q}(t-1) + \sum_{C^{t-1}(p,q)\in S_r^{t-1}(i,j)} B^{(1)}(i-p,j-q)u_{p,q}(t-1) \\ &+ \sum_{C^{t-2}(p,q)\in S_r^{t-2}(i,j)} A^{(2)}(i-p,j-q)y_{p,q}(t-2) + \sum_{C^{t-2}(p,q)\in S_r^{t-2}(i,j)} B^{(2)}(i-p,j-q)u_{p,q}(t-2) + \cdots \\ &+ \sum_{C^{t-\tau}(p,q)\in S_r^{t-\tau}(i,j)} A^{(\tau)}(i-p,j-q)y_{p,q}(t-\tau) + \sum_{C^{t-\tau}(p,q)\in S_r^{t-\tau}(i,j)} B^{(\tau)}(i-p,j-q)u_{p,q}(t-\tau) \end{aligned} \tag{2}$$

$$y_{i,j} = g(s_{i,j}) = \frac{1}{2}[|s_{i,j}+1| - |s_{i,j}-1|] \tag{3}$$

where τ is the time lag, defined as a positive integer, indicating how many past images or patterns are involved in the evolution procedure; $s_{i,j} \in \mathbb{R}$, $y_{p,q} \in \mathbb{R}$, $u_{p,q} \in \mathbb{R}$, and $c_0 \in \mathbb{R}$ are the state, output, input, and threshold of the cell $C(i,j)$, respectively; $A^{(k)}$ and $B^{(k)}$, with $k = 1,2,\cdots,\tau$, are called the feedback and the input synaptic operators [Chua & Roska, 2002]. Notice that the standard nonlinearity g may be defined as many other functions [Chua & Roska, 1993].

Table 1. The $(2r+1)\times(2r+1)$ neighbourhood

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

2.2 The GCNN model

For sake of simplicity of description, consider the zero-input (autonomous) class of STE systems. In an autonomous STE system, no external input image is imposed, and the output image at any time *t* is due exclusively to some initial conditions. Model representations for these situations can easily be extended, in a straightforward way, to other more complex cases. For an autonomous STE system, the state equation (2) becomes

$$\begin{aligned} s_{i,j}(t) &= c_0 + \sum_{|p-i|\le r}\sum_{|q-j|\le r} A^{(1)}(i-p,j-q)y_{p,q}(t-1) \\ &\quad + \sum_{|p-i|\le r}\sum_{|q-j|\le r} A^{(2)}(i-p,j-q)y_{p,q}(t-2)+\cdots \\ &\quad + \sum_{|p-i|\le r}\sum_{|q-j|\le r} A^{(\tau)}(i-p,j-q)y_{p,q}(t-\tau) \\ &= c_0 + \sum_{p=-r}^{r}\sum_{q=-r}^{r} a^{(1)}_{p,q}y_{i+p,j+q}(t-1) + \sum_{p=-r}^{r}\sum_{q=-r}^{r} a^{(2)}_{p,q}y_{i+p,j+q}(t-2)+\cdots \\ &\quad + \sum_{p=-r}^{r}\sum_{q=-r}^{r} a^{(\tau)}_{p,q}y_{i+p,j+q}(t-\tau) \end{aligned} \tag{4}$$

where $a^{(k)}_{m,n} = A^{(k)}(m,n)$ for $k = 1,2,\cdots,\tau$. Combining (3) and (4), yields,

$$y_{i,j}(t) = g(s_{i,j}(t)) = g\left(c_0 + \sum_{k=1}^{\tau}\sum_{p=-r}^{r}\sum_{q=-r}^{r} a^{(k)}_{p,q}y_{i+p,j+q}(t-k) \right) \tag{5}$$

Equation (5) involves a total of $d = (2r+1)^2\tau + 1$ variables. For convenience of description, introduce *d* single-indexed variables $x_k(t)$ as below

$$\mathbf{s}(t-k) = [s_{i-r,j-r}(t-k),\cdots,s_{i,j}(t-k),s_{i+r,j+r}(t-k)] \tag{6}$$

$$\mathbf{x}(t) = [x_1(t),x_2(t),\cdots,x_d(t)] = [\mathbf{s}(t-1),\mathbf{s}(t-2),\cdots,\mathbf{s}(t-\tau)] \tag{7}$$

where $[x_{1+(k-1)(2r+1)^2}(t),\cdots,x_{k(2r+1)^2}(t)] = \mathbf{s}(t-k)$ for $k = 1,2,\cdots,\tau$. For the case $\tau = 1$, the description (7) is shown in Table 1. Equation (5) then becomes

$$y_{i,j}(t) = g\left(c_0 + \sum_{m=1}^{d} c_m x_m(t) \right) \tag{8}$$

where each c_m corresponds to one and only one $a^{(k)}_{p,q}$ with $|p| \le r$, $|q| \le r$ and $k = 1,2,\cdots,\tau$.

Now assume that the true model of a STE system to be identified is of the form

$$y(t) = f(\mathbf{x}(t)) = f(x_1(t),x_2(t),\cdots,x_d(t)) \tag{9}$$

where $y(t)$ represents the state variable $s_{i,j}(t)$ corresponding to the central cell $C^t(i,j)$. For a real-world STE system, the true model f is generally unknown and needs to be identified from available observations. The task of STE system identification is to construct, based on available data, a model that can represent, as close as possible, the observed evolution behaviour. 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 is also required, 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.

In this study, the CNN cell model (8) is used as the elementary building block to approximate the unknown function f in (9). Let $g(\mathbf{x};\mathbf{c}) = g(c_0 + c_1x_1 + \cdots + c_dx_d)$, where $\mathbf{c} = [c_0, c_1, \cdots, c_d]^T$, $\mathbf{x} = [x_1, x_2, \cdots, x_d]^T$ and g is given by (3). The basic idea for constructing an GCNN model is to successively approximate the function f by progressively minimising the approximation errors. This 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,etc.; at the jth step, the approximation f_j is augmented by including the jth construction function $g_j(\mathbf{x};\mathbf{c}_j)$ that produces the largest decrease in the approximation error, that is, it minimises the objective function: $\min_{\alpha,\mathbf{c}} \| f - (f_{j-1} + \alpha g(\mathbf{x};\mathbf{c}) \|^2$. The true function f is generally unknown, the relevant observations of this function are therefore often used for model estimation.

Assume that after the mth step search, the approximation error has been deduced to a desired level, that is, a GCNN model that consists of a total of m CNN building blocks provides a satisfactory representation for a given STE system, in the sense that,

$$\left\| f - \sum_{j=1}^{m} \alpha_j g(\mathbf{x};\mathbf{c}_j) \right\| \le \varepsilon \tag{10}$$

where ε is a predetermined threshold of approximation error. The coarse GCNN model can then be chosen as

$$f \approx f_j = \sum_{j=1}^{m} \alpha_j g(\mathbf{x};\mathbf{c}_j) \tag{11}$$

Notice that the m functions $g_j = g(\mathbf{x};\mathbf{c}_j)$, with j=1,2, …, m, involved in the coarse GCNN model (11) may be redundant, some refinement procedure may thus be required to improve the generalisation performance of the coarse model. Details of the two stage procedure for constructing the GCNN model is presented in the next section.

3. Constructing the GCNN model

Inspired by the successful applications of projection pursuit regression (PPR) [Friedman & Stuetzle, 1981] and other constructive learning algorithms [Jones, 1992; Mallat & Zhang, 1993; Hwang et al., 1994; Kwok & Yeung, 1997a, 1997b; Wei & Billings, 2004; Billings & Wei, 2005], this study proposes a simple orthogonal projection pursuit (OPP) learning scheme, implemented by a particle swarm optimisation (PSO) algorithm. Similar to other constructive algorithms, models produced by the OPP algorithm may, however, be highly redundant. To remove or reduce redundancy, a forward orthogonal regression (FOR) learning algorithm [Billings & Wei, 2007a; Wei & Billings, 2007], implemented using a mutual information estimation method, is applied to refine and improve the initially generated model by the OPP algorithm.

Note that in the following, the inner product is defined for sampled vectors in *N*-dimensional Euclidian space, for example, the inner product of the two vectors $\mathbf{u} = [u(1), u(2), \cdots, u(N)]^T$ and $\mathbf{v} = [v(1), v(2), \cdots, v(N)]^T$ 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 defined in (7), where the inner product is imposed to functions in $L^2(\mathbb{R})$.

3.1 The OPP algorithm for coarse model identification

Let $\mathbf{y} = [y(1), y(2), ..., y(N)]^T \in \mathbb{R}^N$ be the vector of given observations of the output signal, $\mathbf{x}_k = [x_k(1), x_k(2), \cdots, x_k(N)]^T$ the vector of the observations for the *k*th input variable, with *k*=1,2, …, *d*. For any given $\mathbf{c} = [c_0, c_1, \cdots, c_d]^T$, let $\mathbf{g}(\mathbf{X}; \mathbf{c}) = g(c_0 + c_1\mathbf{x}_1 + \cdots + c_d\mathbf{x}_d)$, where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_d]$.

The basic idea of the OPP algorithm for coarse model identification is to successively approximate the function *f* by progressively minimising the approximation errors. 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 $\mathbf{g}_1 = \mathbf{g}(\mathbf{X}; \mathbf{c}_1)$ such that $(\mathbf{c}_1, w_1) = \arg\min_{\mathbf{c}, w} \{\| \mathbf{r}_0 - w\mathbf{g}(\mathbf{X}; \mathbf{c}) \|^2\}$. The associated residual vector may be defined as $\mathbf{r}_1 = \mathbf{r}_0 - w_1\mathbf{g}_1$, 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 coefficients $(\mathbf{c}_1, w_1)$ may not always be identical to the true (theoretical) optimal value $\mathbf{c}_1^*$, no matter what optimisation algorithms are applied. As a consequence, $\mathbf{r}_1 = \mathbf{r}_0 - w_1\mathbf{g}_1$ 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 $\alpha_1 = < \mathbf{r}_1, \mathbf{g}_1 > / \| \mathbf{g}_1 \|^2$.

Assume that at the (*n*-1)th step, a total of (*n*-1) construction vectors $\mathbf{g}_j = \mathbf{g}(\mathbf{X}; \mathbf{c}_j)$, 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 nth construction vector can be obtained by choosing $(\mathbf{c}_n, w_n) = \arg\min_{\mathbf{c},w}\{\| \mathbf{r}_{n-1} - w\mathbf{g}(\mathbf{X};\mathbf{c}) \|^2\}$ and $\mathbf{g}_n = \mathbf{g}(\mathbf{X};\mathbf{c}_n)$. The associated residual vector can be defined as

$$\mathbf{r}_n = \mathbf{r}_{n-1} - \alpha_n \, \mathbf{g}(\mathbf{X};\mathbf{c}_n) \tag{12}$$

where

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

From (12),

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

By respectively summing (12) and (14) for n from 2 to m+1, yields

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

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

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 the *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 PSO based OPP algorithm can briefly be summarised as follows.

The PSO+OPP algorithm:

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

n=1;

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

//{mOPP is the maximum number of construction functions // permitted to be included in the network} //

$(\mathbf{c}_n, w_n) = \arg\min_{\mathbf{c},w}\{\| \mathbf{r}_{n-1} - w\mathbf{g}(\mathbf{X};\mathbf{c}) \|^2\}$; //{Starting from some random (but reasonable) // value for the parameter vector $\mathbf{c}$, optimise the // associated functions using the PSO algorithm.}//

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

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

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

n=n+1;

end while

Note that for each n in the inner loop of the PSO+OPP algorithm, the associated PSO algorithm repeatedly runs 10 times, and the coefficients that produce the smallest value for the object function are chosen to be the parameters for the nth step search. It is clear from (14) that the sequence $\| \mathbf{r}_n \|^2$ is strictly decreasing and positive; thus, by following the method given in 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. Notice that in the OPP algorithm, the elementary building blocks are some CNN cell models, where the 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 or difficult to obtain. However, like the conventional projection pursuit regression algorithm, the OPP algorithm may produce redundant models. To refine and improve the OPP produced network models, the forward orthogonal regression (FOR) learning algorithm, assisted by a mutual information method [Billings & Wei, 2007a; Wei & Billings, 2007], is then applied to remove any severe redundancy.

3.2 The PSO algorithm for parameter optimisation

Particle swarm optimisation (PSO), originally inspired by sociological behaviour associated with, for example, bird flocking [Kennedy et al., 2001], is a population-based stochastic optimisation algorithm that was first proposed by Kennedy and Eberhart in 1995 [Kennedy & Eberhart, 1995; Eberhart & 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. 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 $\boldsymbol{\theta} = [\theta_1, \theta_2, \cdots, \theta_s]^T \in \Theta \subset \mathrm{R}^s$. 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

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

$$\boldsymbol{\theta}_i(t+1) = \boldsymbol{\theta}_i(t) + \mathbf{v}_i(t+1) \tag{18}$$

where i=1,2, …, L; 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 $c_1 = c_2$ =2.05 [Kennedy & Eberhart, 1995; Eberhart & Kennedy, 1995].

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 & Engelbrecht, 2004]

$$\mathbf{p}_i(t+1) = \begin{cases} \mathbf{p}_i(t), & \text{if } \pi(\boldsymbol{\theta}_i(t+1)) \geq \pi(\mathbf{p}_i(t)) \\ \boldsymbol{\theta}_i(t+1), & \text{if } \pi(\boldsymbol{\theta}_i(t+1)) < \pi(\mathbf{p}_i(t)) \end{cases} \tag{19}$$

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

$$\mathbf{p}_g(t+1) = \arg\min_{\mathbf{p}_i} \pi(\mathbf{p}_i(t+1)), \qquad 1 \leq i \leq L. \tag{20}$$

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

$$\pi_{n-1}(\boldsymbol{\theta}, w) = \| \mathbf{r}_{n-1} - w\mathbf{g}(\mathbf{X};\boldsymbol{\theta}) \|^2 = \sum_{t=1}^{N} [r_{n-1}(t) - wg(\theta_0 + \theta_1 x_1(t) + \cdots + \theta_d x_d(t))]^2 \tag{21}$$

where N is the number of training samples.

With regard to the termination of the optimisation procedure, the criterion can be chosen as follows. 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 The FOR algorithm for model refinement

Assume that a total of M basis functions of the form $g_j = g(c_{j,0} + c_{j,1}x_1 + \cdots + c_{j,d}x_d)$, where g is defined by (3), are involved after having performed the PSO based OPP procedure on the given data set. Denote the set of these M functions by

$$\Omega = \{g_j : g_j = g(c_{j,0} + c_{j,1}x_1 + \cdots + c_{j,d}x_d), j = 1,2,\cdots,M\} \tag{22}$$

Note that all the parameters $c_{j,k}$ have already been estimated as part of the coarse model identification procedure. Experience shows that the set Ω may be highly redundant, and a refinement procedure thus needs to be performed to produce a parsimonious model.

The objective of this refinement stage is to reselect the most significant construction functions from the set Ω, to form a more compact model for a given nonlinear identification problem. Let $\mathbf{y}$ and $\mathbf{x}_k$ be defined as in previous sections, and let $\mathbf{g}_j = \mathbf{g}(\mathbf{X};\mathbf{c}_j) = g(c_{j,0} + c_{j,1}\mathbf{x}_1 + \cdots + c_{j,d}\mathbf{x}_d)$, where j=1,2, …, M and $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_d]$. Also, let $D = \{\mathbf{g}_j : j = 1,2,\cdots,M\}$. The model refinement problem amounts to finding, from the vector dictionary D, a full dimensional subset $D_m = \{\mathbf{p}_1, \cdots, \mathbf{p}_m\}$ $= \{\mathbf{g}_{i_1}, \cdots, \mathbf{g}_{i_m}\}$, where $\mathbf{p}_k = \mathbf{g}_{i_k}$, $i_k \in \{1,2,\cdots,M\}$ and k=1,2, …, m (generally $m << M$), so that $\mathbf{y}$ can be satisfactorily approximated using a linear combination of $\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_m$ as below

$$\mathbf{y} = \beta_1\mathbf{p}_1 + \cdots + \beta_m\mathbf{p}_m + \mathbf{e}_m \tag{23}$$

where $\mathbf{e}_m$ is the associated model residual vector.

The orthogonal least squares (OLS) type algorithms [Billings et al., 1989; Chen et al., 1989; Billings & Zhu, 1994; Aguirre & Billings, 1995; Chen et al., 2003; Chen et al., 2004; Wei et al., 2004] can be used to determine model basis functions (model terms). In this study, however, a variation of the OLS algorithm, called the forward orthogonal regression (FOR) algorithm, implemented using a mutual information method [Billings & Wei, 2007a; Wei & Billings, 2007], is employed for the model refinement. Assume that $\mathbf{x}$ and $\mathbf{y}$ are two random discrete variables, with alphabet $\mathfrak{X}$ and $\mathfrak{y}$, respectively, and with a joint probability mass function $p(x, y)$ and marginal probability mass functions $p(x)$ and $p(y)$. The mutual information $I(\mathbf{x},\mathbf{y})$ is the relative entropy between the joint distribution and the product distribution $p(x)p(y)$, given as [Cover & Thomas, 1991]

$$I(\mathbf{x},\mathbf{y}) = \sum_{x\in\mathfrak{X}}\sum_{y\in\mathfrak{y}} p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right) \tag{24}$$

The mutual information $I(\mathbf{x},\mathbf{y})$ is the reduction in the uncertainty of $\mathbf{y}$ due to the knowledge of $\mathbf{x}$, and vice versa. Mutual information provides a measure of the amount of information that one variable shares with another one. If $\mathbf{y}$ is chosen to be the system output (the response), and $\mathbf{x}$ is one regressor in a linear model, $I(\mathbf{x},\mathbf{y})$ can then be used to measure the coherence of $\mathbf{x}$ with $\mathbf{y}$ in the model. Several algorithms have been proposed to estimate mutual information from observed data, see for example Moddemeijer [1989, 1999], Darbellay and Vajda [1999], and Paninski [2003] and the references therein.

Detailed discussions on the utility of the mutual information for model term selection can be found in Billings and Wei [2007a] and Wei and Billings [2007]. Now, let $\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_n$ be the n selected

linearly independent basis vectors after the nth step search, and let $\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_n$ be a group of orthogonal vectors, generated from the vectors $\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_n$, by means of some orthogonal transformation. Following Billings et al. [1989], Chen et al. [1989], the error reduction ratio (ERR), produced by including the nth basis vector $\mathbf{q}_n$, or equivalently by including $\mathbf{p}_n$, is defined as

$$\mathrm{ERR}_n = \frac{\gamma_n^2 \| \mathbf{q}_n \|^2}{\| \mathbf{y} \|^2} \tag{25}$$

where $\gamma_n = < \mathbf{y}, \mathbf{q}_n > / \| \mathbf{q}_n \|^2$. ERR can be used to measure the significance of individual model terms in that it provides an index indicating the contribution made by each selected individual model term to explain the total variance in the desired output signal.

Let $\mathbf{e}_n$ be the residual vector produced at the nth search step. Similar to in the OPP algorithm, the model residual vector $\mathbf{e}_n$ can be used to form a criterion to terminate the search procedure. Following the suggestion in Billings and Wei [2007b], the following adjustable prediction error sum of squares (APRESS), also referred to as the adjustable generalised cross-validation (AGCV), will be used to monitor the regressor search procedure

$$\mathrm{APRESS}_n = \frac{\mathrm{MSE}(n)}{(1 - \lambda n / N)^2} \tag{26}$$

where $\mathrm{MSE}(n) = \| \mathbf{e}_n \|^2 / N$ is the mean-square-error that is associated to the model of n model terms. The number of regressors (wavelet functions) will be chosen as the value where APRESS arrives it minimum. Billings and Wei [2007b] suggest that the adjustable parameter λ be chosen between 5 and 10.

Following Billings and Wei [2007a] and Wei and Billings [2007], the mutual information based forward orthogonal regression (FOR) algorithm, is briefly summarised below.

The FOR-MI algorithm:

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

for j=1 to M

$\mathbf{q}_j^{(1)} = \boldsymbol{\varphi}_j$;

$I^{(1)}[j] = MI(\mathbf{r}_0, \mathbf{q}_j^{(1)})$; // Calculate the mutual information for all // candidate basis vectors.//

end for

$\ell_1 = \arg\max_{i \in U_1} \{I^{(1)}[i]\}$; $V_1 = \{\ell_1\}$;

$\mathbf{p}_1 = \boldsymbol{\varphi}_{\ell_1}$; $\mathbf{q}_1 = \mathbf{p}_1$; $\gamma_1 = \dfrac{< \mathbf{y}, \mathbf{q}_1 >}{\| \mathbf{q}_1 \|^2}$; $\mathbf{r}_1 = \mathbf{r}_0 - \gamma_0 \mathbf{q}_1$;

$$\mathrm{ERR}[1] = \frac{\gamma_1^2 \| \mathbf{q}_1 \|^2}{\| \mathbf{y} \|^2}; \mathrm{APRESS}[1] = \frac{1}{(1 - \lambda / N)^2} \frac{\| \mathbf{r}_1 \|^2}{N};$$

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

For n=2 to M

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

for $j \in U_n$

$$\mathbf{q}_j^{(n)} = \boldsymbol{\varphi}_j - \sum_{k=1}^{n-1} \frac{< \boldsymbol{\varphi}_j, \mathbf{q}_k >}{\| \mathbf{q}_k \|^2} \mathbf{q}_k \ ;$$

$I^{(n)}[j] = MI(\mathbf{r}_{n-1}, \mathbf{q}_j^{(n)})$; //Calculate the mutual information for all

// for all candidate basis vectors.//

//{if $\| \mathbf{q}_j^{(n)} \|^2 \le \varepsilon$, set $I^{(n)}[j] = 0$ }//

end for (end loop for j)

$\ell_n = \arg\max_{j \in U_n} \{I^{(n)}[j]\}$; $V_n = \{\ell_n\} \bigcup \{ \arg_{j \in U_n} (\| \mathbf{q}_j^{(n)} \|^2 < \varepsilon) \}$;

$\mathbf{p}_n = \boldsymbol{\varphi}_{\ell_n}$; $\mathbf{q}_n = \mathbf{q}_{\ell_n}^{(n)}$; $\gamma_n = \dfrac{< \mathbf{y}, \mathbf{q}_n >}{\| \mathbf{q}_n \|^2}$; $\mathbf{r}_n = \mathbf{r}_{n-1} - \gamma_n \mathbf{q}_n$;

$\mathrm{ERR}[n] = \dfrac{\gamma_n^2 \| \mathbf{q}_n \|^2}{\| \mathbf{y} \|^2}$; $\mathrm{APRESS}[n] = \dfrac{1}{(1 - \lambda n / N)^2} \dfrac{\| \mathbf{r}_n \|^2}{N}$;

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 (basis functions) 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 $\boldsymbol{\beta} = [\beta_1, \beta_2, \cdots, \beta_m]^T$, relative to the model (23), can easily be calculated from the triangular equation $\mathbf{R}\boldsymbol{\beta} = \boldsymbol{\gamma}$, where R is an upper triangular matrix and $\boldsymbol{\gamma} = [\gamma_1, \gamma_2, \cdots, \gamma_m]^T$ with $\gamma_i = < \mathbf{y}, \mathbf{q}_i > / \| \mathbf{q}_i \|^2$ for i=1,2,…, m.

4. Case Studies

Consider the generalised coupled map lattice model below

$$\begin{aligned} s_{i,j}(t) &= (1 - c_1 - c_2)\phi(s_{i,j}(t-1)) \\ &+ \frac{c_1}{4}[\phi(s_{i-1,j}(t-1)) + \phi(s_{i,j-1}(t-1)) + \phi(s_{i,j+1}(t-1)) + \phi(s_{i+1,j}(t-1))] \\ &+ c_2 \phi\left(\frac{1}{4}[s_{i-1,j-1}(t-1) + s_{i-1,j+1}(t-1) + s_{i+1,j-1}(t-1) + s_{i+1,j+1}(t-1)] \right) \end{aligned} \tag{27}$$

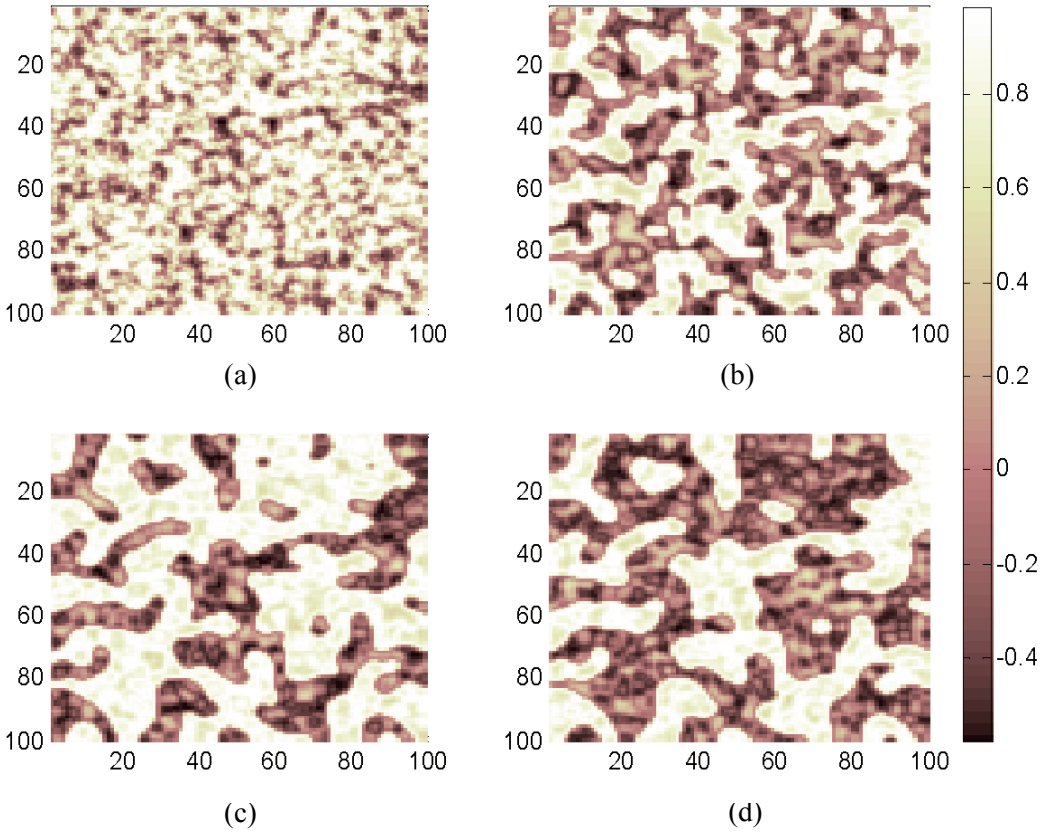

Fig. 1. Instant snapshots produced by the model (27), with a=1.6, c_1=0.28, and c_2=0.12. (a) t=5; (b) t=20; (c) t=75; (d) t=100.

where ϕ is defined as the typical logistic map function $\phi(x) = 1 - ax^2$, c_1 and c_2 are constants. In this study, the case where a=1.6, c_1=0.28, and c_2=0.12 was considered. Starting with an initial pattern of size 100×100, where values of all the initial cells were randomly selected in [0,1], the model (27) was simulated. Some patterns are shown in Fig. 1.

A dataset consisting of 2000 data pairs that were randomly selected from the first 100 patterns was created and this dataset was then used for GCNN model identification. A Gaussian white noise sequence with zero mean and standard deviation of 0.005 was added to the training dataset. Some conditions, relative to the model identification procedures using the PSO+OPP and FOR algorithms, are listed in Table 2. The ERS (error-to-signal ratio) criterion, relative to the PSO+OPP algorithm, is shown in Fig. 2. The adjustable generalised cross-validation (AGCV), defined by (26), is shown in Fig. 3, which suggests that a GCNN model with 31 model terms would be a good choice.

Table 2. Some conditions involved in the identification procedure.

Size of the arrays of cells	100×100
Number of model variables	9
mOPP in the OPP algorithm	200
η in the OPP algorithm	10^{-5}
Swarm size in the PSO algorithm	100
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	500
δ in the PSO algorithm	10^{-5}
ε in the FOR algorithm	10^{-10}
λ in the FOR algorithm	10

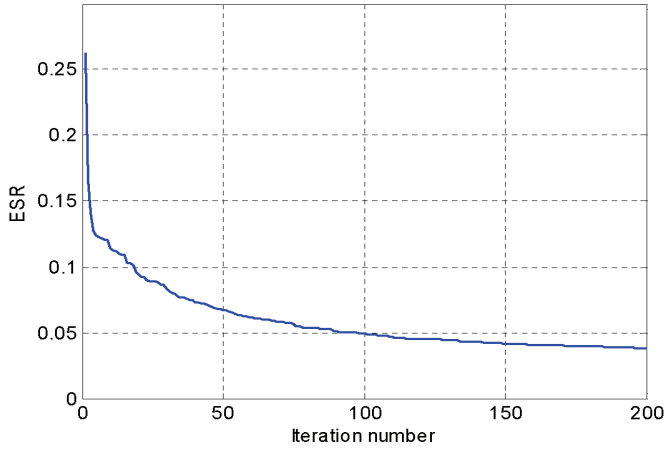

Fig. 2. The ESR(error-to-signal ratio) criterion, relative to the PSO+OPP algorithm.

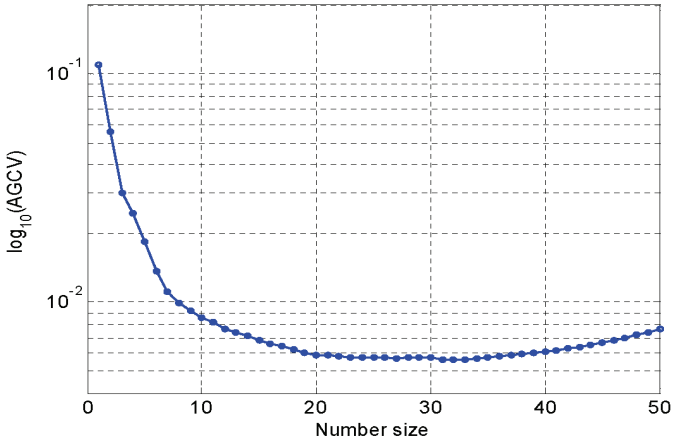

Fig. 3. The AGCV criterion, relative to the FOR algorithm.

To evaluate the performance of the identified GCNN model of 31 model terms, the patterns produced by this model were investigated and compared with those produced by the original model (27). Denote the observation of the pattern measured at the time instant t by $X(t)$. The k-step-ahead prediction, denoted by $\hat{X}(t+k \mid X(t); f)$, where f represents the identified nonlinear function, is the iteratively produced result using the identified model, on the basis of $X(t)$, but without using information on observations for patterns at any other time instants. As an example, starting with a random initial pattern of size 128×128, where values were uniformly distributed in $[0,1]$, the one and ten step-ahead predicted patterns were calculated using the identified GCNN model. The one and ten step ahead predicted patterns at time instant t=20, along with the corresponding patterns produced by the model (26), are shown in Fig. 4, which clearly shows that the identified GCNN model can be used to reconstruct the dynamics possessed by the original model (27).

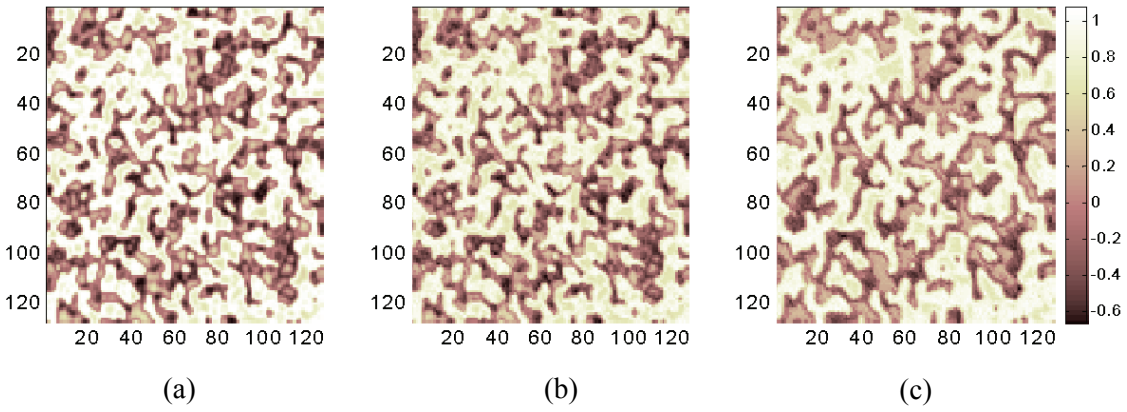

Fig. 4. A comparison of the 1- and 10-step-ahead predicted patterns at the time instant t=20, produced by the identified GCNN model, with that produced by the original model (27). (a) The pattern produced by the original model (27); (b) 1-step-ahead predicted pattern; (c) 10-step-ahead predicted pattern.

5. Conclusions

The proposed generalized CNN (GCNN) modeling framework provides a powerful model identification approach for spatio-temporal evolutionary (STE) systems. The introduction of the novel two-stage training scheme, where the PSO based orthogonal projection pursuit (OPP) algorithm is employed for a coarse model identification and the mutual information assisted forward orthogonal regression (FOR) algorithm is used for model refinement, enable the GCNN modeling procedure to be very effective because of the following features. Firstly, the network training procedure is almost self-implemented, meaning that by starting with some given conditions (initial, boundary and termination), all the within and between-network parameters can be estimated and calculated by the proposed

algorithms. Secondly, the model identification procedure can produce a transparent model, where individual neurons are explicitly available. Thirdly, by applying the FOR algorithm, the initially produced model by the OPP algorithm, can be significantly refined and improved, and a parsimonious model containing only a small number of neurons can then be obtained.

By introducing the PSO algorithm, which is easy to implement, the calculation of gradients required by classical nonlinear optimisation algorithms can be avoided. This makes the new modelling framework very suitable for STE system identification, where relevant object functions may not be differentiable or relevant gradients are very difficult to obtain.

Acknowledgements

The authors gratefully acknowledge that this work was supported by Engineering and Physical Sciences Research Council (EPSRC), U.K.

References

Adamatzky, A. [1994] *Identification of Cellular Automata* (London: Taylor & Francis).

Aguirre, L. A. & Billings, S. A. [1995] "Retrieving dynamical invariants from chaotic data using NARMAX models," *Int. J. Bifurcation and Chaos*, **5**, 449-474.

Billings, S. A., Chen, S. & Korenberg, M. J. [1989] "Identification of MIMO nonlinear systems suing a forward regression orthogonal estimator," *Int. J. Control*, **49**, 2157-2189.

Billings, S. A. & Yang, X. X. [2003] "Identification of the neighborhood and CA rules from spatio-temporal CA patterns," *IEEE Trans. Syst., Man, Cybern., B: Cybern.*, **33**, 332–339.

Billings, S. A. & Wei, H. L. [2005] "The wavelet-NARMAX representation: a hybrid model structure combining the polynomial models and multiresolution wavelet decompositions", *Int. J. Systems Science*, **36**, 137-152.

Billings, S. A. & Wei, H. L. [2007a] "Sparse model identification using a forward orthogonal regression algorithm aided by mutual information," *IEEE Trans. Neural Networks*, **18**, 306-310.

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

Billings, S. A. & Zhu, Q. M. [1994] "A structure detection algorithm for nonlinear dynamical rational models", *Int. J. Control*, **59**, 1439-1463.

Cai, H. & Min, L. Q. [2005] "A kind of two-input CNN with application," *Int. J. Bifurcation and Chaos*, **15**, 4007-4011.

Chen, S., Billings, S. A. & Luo, W. [1989] "Orthogonal least squares methods and their application to nonlinear system identification," *Int. J. Control*, **50**, 1873–1896.

Chen, S., Hong, X. & Harris, C. J. [2003] "Sparse kernel regression modeling using combined locally regularized orthogonal least squares and D-optimality experimental design," *IEEE Trans. Automatic Control*, **48**, 1029-1036.

Chen, S., Hong, X., Harris, C. J. & Sharkey, P. M. [2004] "Sparse modeling using orthogonal forward regression with press statistic and regularization," *IEEE Trans. Syst., Man, Cybern., B: Cybern.*, **34**, 898–911.

Chen, F. J., Chen, F. Y. & He, G. L. [2005] "Image processing via CNN genes with Von Neumann neighborhoods," *Int. J. Bifurcation and Chaos*, **15**, 3999-4006.

Chua, L. O. & Yang, Y. [1988a] "Cellular neural networks: Theory," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, **35**, 1257–1272.

Chua, L. O. & Yang, Y. [1988b] "Cellular neural networks: Applications," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, **35**, 1273–1290.

Chua, L. O. & Roska, T. [1993] "The CNN paradigm," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, **40**, 147–156.

Chua, L. O., Hasler, M., Moschytz, G. S. & Neirynck, J. [1995] "Autonomous cellular neural networks - a unified paradigm for pattern-formation and active wave-propagation," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, **42**, 559-577.

Chua, L. O. & Roska, T. [2002] *Cellular Neural Networks and Visual Computing* (Cambridge: Cambridge University Press).

Clerc, M. & Kennedy, J. [2002] "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," *IEEE Trans. Evol. Comput.*, **6**, 58–73.

Cover, T. M. & Thomas, J. A. [1991] *Elements of Information Theory* (New York: John Wiley & Sons).

Crounse, K. R. & Chua, L. O. [1995] "Methods for image-processing and pattern-formation in cellular neural networks - a tutorial," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, **42**, 583-601.

Crounse, K. R., Roska, T. & Chua, L. O. [1993] "Image half-toning with cellular neural networks," *IEEE Trans. Circuits Syst. II, Analog and Digital Signal Processing.*, **40**, 267-283.

Crounse, K. R., Chua, L. O., Thiran, P. & Setti, G. [1996] "Characterization and dynamics of pattern formation in cellular neural networks," *Int. J. Bifurcation and Chaos*, **6**, 1703-1724.

Darbellay, G.A. & Vajda, I. [1999] "Estimation of the information by an adaptive partitioning of the observation space," *IEEE Transactions on Information Theory*, **45**, 1315-1321.

Eberhart, R. C. & Kennedy, J. [1995] "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.

Friedman, J. H. & Stuetzle, W. [1981] "Projection pursuit regression," *J. Amer. Statist. Assoc.*, **76**, 817–823.

Gilli, M., Roska, T., Chua, L. O. & Civalleri, P. P. [2002] "CNN dynamics represent a broader class than PDEs," *Int. J. Bifurcation and Chaos*, **12**, 2051-2068.

Harrer, H. & Nossek, J. A. [1992] "Discrete0time cellular networks," *Int. J. Circuit Theory Applicat.*, **20**, 453-467.

He, Q. B. & Chen, F. Y. [2006] "Designing CNN genes for binary image edge smoothing and noise

removing," *Int. J. Bifurcation and Chaos*, **16**, 3007-3013.

Huang, G. B., Chen, L. & Siew, C. K. [2006] "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Trans. Neural Networks*, **17**, 879–892.

Hwang, J. N., Lay, S. R., Maechler, M., Martin, R. D. & Schimert, J. [1994] "Regression modeling in back-propagation and projection pursuit learning," *IEEE Trans. Neural Networks*, **5**, 342–353.

Itoh, M. & Chua, L. O. [2003] "Equivalent CNN cell models and patters," *Int. J. Bifurcation and Chaos*, **13**, 1055-1161.

Itoh, M. & Chua, L. O. [2005] "Image processing and self-organizing CNN," *Int. J. Bifurcation and Chaos*, **15**, 2939-2958.

Jones, L. K. [1992] "A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training," *Ann. Statist.*, **20**, 608–613.

Kanakov, O. I., Shalfeev, V. D. & Forti, G. L. [2006] "Stationary patterns in CNN-like ensembles with modified cell output functions," *Int. J. Bifurcation and Chaos*, **16**, 2207-2220.

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

Kennedy, J. & Eberhart, R. C. [1995] "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.

Kwok, T. Y. & Yeung, D. Y. [1997a] "Constructive algorithms for structure learning in feedforward neural networks for regression problems," *IEEE Trans. Neural Netw.*, **8**, 630–645.

Kwok, T. Y. & Yeung, D. Y. [1997b] "Objective functions for training new hidden units in constructive neural networks," *IEEE Trans. Neural Netw.*, **8**, 1131–1148.

Lin, S. S. & Yang, T. S. [2002] "On the spatial entropy and patterns of two-dimensional cellular neural networks," *Int. J. Bifurcation and Chaos*, **12**, 115-128.

Mallat, S. & Zhang, Z. [1993] "Matching pursuit with time-frequency dictionaries," *IEEE Trans. Signal Process.*, **41**, 3397–3415.

Miller, J. & Huse, D. A. [1993] "Macroscopic equilibrium from microscopic irreversibility in a chaotic coupled map lattice," *Phys. Rev.* ***E*** **48**, 2528–2535.

Moddemeijer, R. [1989] "On estimation of entropy and mutual information of continuous distributions," *Signal Processing*, **16**, 233-246.

Moddemeijer, R. [1999] "A statistic to estimate the variance of the histogram-based mutual information estimator based on dependent pairs of observations," *Signal Processing*, **75**, 51-63.

Morfu, S. & Comte, J. C. [2004] "A nonlinear oscillators network devoted to image processing," *Int. J. Bifurcation and Chaos*, **14**, 1385-1394.

Paninski, L. [2003] **"**Estimation of entropy and mutual information," *Neural Computation*, **15**, 1191-1253.

Roska, T. & Chua, L. O. [1992] "Cellular neural networks with nonlinear and delay-type template elements and nonuniform grids," *International Journal of Circuit Theory and Applications*, **20**, 469-481,

Roska, T. & Chua, L. O. [1993], "The CNN universal machine - an analogic array computer," *IEEE Trans. Circuits Syst. II, Analog Digital Sign. Proc.*, **40**, 163–173.

Sbitnev, V. I. & Chua, L. O. [2002] "Local activity criteria for discrete-MAP CNN," *Int. J. Bifurcation and Chaos*, **12**, 1227-1272.

Shi, B. E. [2004] "Oriented spatio patter formation in a four layer CMOS cellular network," *Int. J. Bifurcation and Chaos*, **14**, 1209-1221.

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

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

Stoffels, A., Roska, T. & Chua, L. O. [1996] "On object-oriented video coding using the CNN universal machine," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, **43**, 948-952.

Thiran, P., Crounse, K. R., Chua, L. O. & Hasler, M. [1995] "Pattern-formation properties of autonomous cellular neural networks," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, **42**, 757-774.

van den Bergh, F. & Engelbrecht, A. P. [2004] "A cooperative approach to particle swarm optimization," *IEEE Trans. Evolutionary Computation*, **8**, 225-239.

Venetianer, P. L., Werblin, F., Roska, T. & Chua, L. O. [1995] "Analogic CNN algorithms for some image compression and restoration tasks," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, **42**, 278-284.

Wei, H. L. & Billings, S. A. [2004] "Identification and reconstruction of chaotic systems using multiresolution wavelet decompositions," *Int. J. Systems Science*, **35**, 511-526.

Wei, H. L., Billings, S. A. & Liu, J. [2004] "Term and variable selection for nonlinear system identification," *Int. J. Control*, **77**, 86-110.

Wei, H. L. & Billings, S. A. [2007], "Model structure selection using an integrated forward orthogonal search algorithm assisted by squared correlation and mutual information," *Int.J. Modelling, Identification and Control* (in press).

Wolfram, S. [2002] *A New Kind of Science* (Champaign, IL: Wolfman Media).