This is a repository copy of *Identification of N-state spatio-temporal dynamical systems using a polynomial model*.

White Rose Research Online URL for this paper:
http://eprints.whiterose.ac.uk/74617/

Monograph:
Guo, Y., Billings, S.A. and Coca, D. (2007) Identification of N-state spatio-temporal dynamical systems using a polynomial model. Research Report. ACSE Research Report no. 960 . Automatic Control and Systems Engineering, University of Sheffield

Identification of *N*-state Spatio-Temporal Dynamical Systems Using a Polynomial Model

Yuzhu Guo, Steve A. Billings and Daniel Coca

IDENTIFICATION OF *N*-STATE SPATIO-TEMPORAL DYNAMICAL SYSTEMS USING A POLYNOMIAL MODEL

Yuzhu Guo, Steve A. Billings and Daniel Coca

Abstract

A multivariable polynomial model is introduced to describe *n*-state spatio-temporal systems. Based on this model, a new neighbourhood detection and transition rules determination method is proposed. Simulation results illustrate that the new method performs well even when the patterns are corrupted by static and dynamical noise.

Keywords: Identification; spatio-temporal system; polynomial model.

1. Introduction

Spatio-temporal systems are a class of dynamical systems which can be studied by discretizing space and time. So that the cells evolve in discrete time steps according to deterministic rules that depend on a neighbourhood of influence. Because complex behaviours of patterns can be generated from simple mathematical constructs, spatio-temporal systems have been used for simulating various complex systems, such as chemical, biological and ecological systems (Baier et al., 2002, Ermentrout, 1998, Filipe and Gibson, 1998, Rekeczky et al., 1997).

Binary cellular automata, which are one of the simplest classes of spatio-temporal systems, have been extensively studied in recent years. Several cellular automata identification methods were introduced (Adamatzky, 1994, Maeda and Sakama, 2003, Richards et al., 1990). These studies were all based on the simple rule table model and none gave a clear neighbourhood structure or a parsimonious expression of the rule. Hence the detection processes can become complicated especially when the neighbourhood becomes large. The identification of the Boolean rule was studied by Yang and Billings (2000a). Yang and Billings (2000b) divided the identification procedure into two stages: neighbourhood detection and rule determination for the first time. The CA-OLS method and several neighbourhood detection approaches were later proposed (Billings and Yang, 2003, Mei et al., 2005, Zhao and Billings, 2006).

Binary cellular automata however can only be applied to systems with two states o and 1. But there are many processes that exhibit more than two states one famous example is excitable media in which any cell can take on three states: quiescent, excited and refractory. However there are very few studies on the identification of *n*-state spatio-temporal systems. In this paper a polynomial model of *n*-state spatio-temporal system is given for the first time, and a completely new approach for the identification of *n*-state spatio-temporal systems is proposed including neighbourhood detection and rule determination.

The paper is organized as follows. After a short discussion of the necessary preliminaries of *n*-state spatio-temporal systems in section2, a new polynomial model

is introduced in section 3 and it is proved that this model can be used to describe this class of spatio-temporal systems exactly. A neighbourhood detection approach using the polynomial model is given in section 4. To show the validity of the new methods, an example is provided in section 5 and two kinds of noise are considered. Finally, conclusions are given in section 6.

2. *N*-state Spatio-Temporal Systems

An *n*-state spatio-temporal system is specified by a triple $< S, R, f >$ of cell state set *S*, neighbourhood *R* and cell-state transition function $f : S^d \to S$, where *n* denotes the size of the state set *S* and *d* is the size of the neighbourhood *R*. The states of the *n*-state spatio-temporal system come from a finite state set *S*, which is represented by $\{0, 1, 2, L, n-1\}$. All cells change states synchronously at discrete time steps. The next state of each cell depends on the states of the neighbouring cells at a past time according to an update rule. The neighbourhood of a cell is the set of cells in both the spatial and temporal dimensions that are directly involved in the evolution of the cell. The neighbourhood, denoted as *R*, is a *d*-tuple (ξ_1, ξ_1, L, ξ_d). The transition rules are a function $f : S^d \to S$. The state $f(R)$ is the state of a cell whose *d* neighbours were at states $R = (\xi_1, \xi_1, L, \xi_d)$ at a past time. The function $f : S^m \to S$ expresses the deterministic dependence between a cell and its neighbourhood.

The state set of the neighbourhood is the Cartesian product $S^d = S \times S \times L \times S$ of states of all the neighbours that gives all the state combinations of these neighbours. The function values at set S^d, $f(S^d)$, uniquely determine the transition rules of the spatio-temporal system. Because the domain S^d is finite, the function may be expressed by simply tabulating all the arguments $R \in S^d$ and their corresponding function values $c = f(R)$. Each pair (c, R) is named as a case and there are n^d cases for an *n*-state *d*-neighbourhood spatio-temporal system.

For a spatio-temporal pattern, the collection of all cell states in the grid at some time step is called a configuration. A sequence of configurations of a spatio-temporal pattern is denoted as C_0, L, C_t, L, in which each configuration includes all cell states. Here, C_0 is the initial configuration of the spatiotemporal system and C_t represents the configuration of the spatiotemporal system at time *t* (Maeda and Sakama, 2003).

This paper focuses on the identification of patterns in which all cells change synchronously in discrete time steps according to local and identical transition rules. In other words, all cells use the same rules, and the rules are applied to all cells at the same time. The underlying lattice is an infinite rectangular grid throughout this paper.

3. Model for an *N*-state Spatio-Temporal System

3.1 Table form model
Local transition rules express the deterministic dependence between the cell and its neighbourhood. For each case of a neighbourhood, $R \in S^d$, there must be one unique cell state $c = f(R)$ corresponding to this case. There are many ways to represent a

local transition rule: by a formula, by a plot, by an algorithm and so on. The most intuitive approach is where the transition table is simply tabulating for all possible cases, that is, all cases of neighbourhood S^d and the corresponding cell values $f(S^d)$.

An example of a table model of an *n*-state, *d*-neighbourhood spatio-temporal system with a local transition rule is given as Table 1. Each row represents an evolution case. Columns $\xi_1 \sim \xi_d$ show all state combinations of the neighbourhood, and the values in columns c determine the transition rules.

Table 1 Transition table

ξ_1	ξ_2	...	ξ_d	c
0	0	...	0	*
0	0	...	1	*
Λ	Λ	Λ	Λ	Λ
n-1	n-1	...	n-1	*

3.2 Polynomial form model

The transition table is a basic representation of the local transition rules and it is widely used. However it is tedious especially when the number of cases is very large. In this part a new parsimonious polynomial form model is presented to replace the table form expression.

Yang and Billings (2000b) showed that 2-state spatio-temporal systems, binary cellular automata systems, can be exactly expressed by a polynomial. This conclusion will now be extended to *n*-state spatio-temporal systems to show that *n*-state spatio-temporal systems can be exactly expressed by a polynomial. This development will in turn enable the introduction of a new class of neighbourhood detection and rule identification procedures for *n*-state spatio-temporal systems.

Denote the count of the finite state of cells as *n* and the size of the neighbourhood as *d*, let *c* represent a cell in this spatio-temporal system and $R = (\xi_1, \xi_2, \mathrm{L}\ \xi_d)$ represent the neighbourhood of *c*. For $\alpha_i = (\alpha_{i1}, \alpha_{i2}, \mathrm{L}\ ,\alpha_{id}), \alpha_{ij} \leq n-1$, construct monomials $P_i = R^{\alpha_i} = \xi_1^{\alpha_{i1}} \xi_2^{\alpha_{i2}} \mathrm{L}\ \xi_d^{\alpha_{id}}$, $i = 1, 2, \mathrm{L}\ , n^d$ and polynomial $P = \sum_{i=1}^{n^d} \theta_i P_i$, where θ_i are the parameters of this polynomial. It will be shown that an *n*-state, *d*-neighbourhood spatio-temporal system can be exactly represented by the polynomial model $P = \sum_{i=1}^{n^d} \theta_i P_i$. Such a polynomial will be constructed in this section.

It has been shown in subsection 3.1 that transition rules can be described as a table. Each row in the transition table is a case, which is a *d+1*-tuple $(\xi_1, \xi_2, \mathrm{L}\ , \xi_d, c)$ and can be considered as a point in the *d+1*-dimentional space. For an *n*-state, *d*-neighbourhood spatio-temporal system, the transition table is a point set in the *d+1*-dimensional space. Next, it will be shown that this table model can be represented by

a multivariate polynomial model at the case set S^d. In other words, there is a polynomial which has the same values at all cases listed in the table. Hence, the aim is to find a polynomial which goes exactly through these points. To find a polynomial equivalent to the transition table at set S^d is a problem of multivariate polynomial interpolation. A multivariate interpolation polynomial in Lagrange form will be constructed as follows.

To illustrate the simple idea behind normal form interpolation, univariate Lagrange interpolation will be briefly considered. Polynomial interpolation is the interpolation of a given data set by a polynomial. In other words, given some data points, the aim is to find a polynomial which goes exactly through these points. A Lagrange polynomial is a linear combination of Lagrange fundamental polynomials. Given a set of data points $(x_0, y_0), \mathrm{L}, (x_k, y_k)$, the Lagrange polynomial is $L(x) = \sum_{j=0}^{k} y_j l_j(x)$, here

$l_j(x) = \prod_{i=0, i \neq j}^{k} \dfrac{x - x_i}{x_j - x_i}$ are the Lagrange fundamental polynomials. As can easily be seen, $l_i(x)$ is a polynomial and has degree k and $l_i(x_j) = \delta_{ij}$. Thus function $L(x)$ is a polynomial with degree k and $L(x_i) = \sum_{j=0}^{k} y_j l_j(x) = y_i$.

The univariate Lagrange polynomial interpolation can be easily extended to a multivariate polynomial by the tensor product of univariate interpolation polynomials.

It was noted above that the state set of the neighbourhood is the Cartesian product $S^d = S \times S \times \mathrm{L} \times S$ of states of the neighbours. For every neighbour, the state can be any one of the states ranging from 0 to $n-1$. Rearranging the transition table in the $d+1$ dimensional space, every dimension of $\xi_1 \sim \xi_d$ has n points. For example, when $d=2$, Table 1 can be rearranged as shown in Table 2.

Table 2 Transition table for m=2

ξ_1 / ξ_2	0	1	…	n-1
0	*	*	…	*
1	*	*	…	*
Ν	Ν	Ν	O	Ν
n-1	*	*	…	*

For each dimension ξ_i, a 1 dimensional Lagrange fundamental polynomial set $L_i = \{l_{i1}(\xi_i), l_{i2}(\xi_i) \mathrm{L}\ l_{in}(\xi_i)\}$ can be constructed as $l_{ij} = \prod_{k=0, k \neq j-1}^{n-1} \dfrac{(\xi_i - k)}{(j-k)}$, that is, $l_{ij} = 1$ if $\xi_i = j$ and $l_{ij} = 0$ elsewhere.

Denote the indexes $\gamma = (\gamma_1, \gamma_2, \mathrm{L}, \gamma_d) \in S^d$. Computing the tensor product of these univariate Lagrange fundamental polynomial sets as $L = L_1 \otimes L_2 \otimes \mathrm{L} \otimes L_d$, that

is $L = \left\{ h \mid h_\gamma = l_{1,\gamma_1} l_{2,\gamma_2} \mathrm{L}\ l_{d,\gamma_d} \right\}$, yields the multivariate Lagrange fundamental polynomials, which give a 1-1 mapping with the cases in the transition table.

$$h_\gamma = \begin{cases} 1 & if : (\xi_1, \xi_2, \mathrm{L}\ , \xi_d) = \gamma \\ 0 & if : (\xi_1, \xi_2, \mathrm{L}\ , \xi_d) \neq \gamma \end{cases} \tag{1}$$

Define the interpolation polynomial as $g\left(\xi_1, \xi_2, \mathrm{L}\ \xi_d\right) = \sum_{\gamma \in S^d} f(\gamma) h_\gamma \left(\xi_1, \xi_2, \mathrm{L}\ \xi_d\right)$.

Obviously, $g(R) = c$ holds for $\forall (R, c)$ in the transition table. Equally, $g\left(\xi_1, \xi_2, \mathrm{L}\ \xi_d\right)$ can be expressed as $\sum_{i=1}^{n^d} \theta_i \xi_1^{\alpha_1} \xi_1^{\alpha_2} \mathrm{L}\ \xi_d^{\alpha_d}$, where θ_i are parameters of the polynomial and $0 \leq \alpha_i \leq n-1$. Then the n-state d-neighbourhood spatio-temporal system is described by the polynomial model $g\left(\xi_1, \xi_2, \mathrm{L}\ \xi_d\right)$ exactly. Define a monomial term set $T = \left\{ \xi_1^{\alpha_1} \xi_2^{\alpha_2} \mathrm{L}\ \xi_d^{\alpha_d} \right\}$, where $0 \leq \alpha_i \leq n-1$. Arrange all the terms in set T in an increasing order, that is, the terms of lower degree come before the terms of higher degree, into a monomial term vector $P_{n^d \times 1} = \begin{bmatrix} p_1 & p_2 & \mathrm{L} & p_{n^d} \end{bmatrix}, p_i \in T$. Hence the polynomial model can now be described in compact form as $c = P\Theta$, where $P_{n^d \times 1}$ is the monomial term vector and $\Theta_{1 \times n^d} = \begin{bmatrix} \theta_1 & \theta_2 & \mathrm{L} & \theta_{n^d} \end{bmatrix}^T$ is the parameter vector.

Based on the results above, the polynomial model as a representation of n-state, d-neighbourhood spatio-temporal systems has the following advantages:
1) Provides an accurate description of n-state spatio-temporal systems; 2) the model is linear-in-the-parameter and these parameters can be estimated using least squares based estimation methods; 3) the number of terms in the polynomial is n^d which only depends on the number of states (n) and the size of neighbourhood (d), but does not rely to the lattice structure or the dimension of the spatio-temporal system.

4. Neighbourhood Detection

Neighbourhood detection is a core problem in the identification of spatio-temporal systems because once the neighbourhood and the model terms are known it is really easy to identify the model. In this section, a new neighbourhood detection method based on the new polynomial model is introduced.

4.1 Candidate transition table

Assume that a candidate neighbourhood denoted as $W = \left\{ \xi_1, \xi_2, \mathrm{L}\ \xi_d, \mathrm{L}\ \xi_{d+d1} \right\}$, which is large enough to cover the potential neighbourhood denoted as R, is given. Without loss of generality, let $R = \left\{ \xi_1, \xi_2, \mathrm{L}\ , \xi_d \right\}$ be the true neighbourhood of the d-neighbourhood spatio-temporal system. That is, W is a proper superset of R and $\left\{ \xi_{d+1}, \xi_{d+2}, \mathrm{L}\ , \xi_{d+d1} \right\} = W - R$ is the relative complement of R in W. For an n-state spatio-temporal system whose state set of all cells is S, the state set of the candidate neighbourhood is S^{d+d1} . Define a function $v : S^{d+d1} \rightarrow S$ and let

$v(\xi_1,\xi_2,L\ \xi_d,L\ \xi_{d+d1}) = f(\xi_1,\xi_2,L\ \xi_d)$. Hence, function v with $d+d1$ neighbours as inputs can describe the d-neighbourhood system as the transition function f does. Obviously, the values of function v do not rely on the states of neighbours $\xi_{d+1}, L\ , \xi_{d+d1}$. That is the value of function v does not change with the difference of the states of combination of $(\xi_{d+1},\xi_{d+2},L\ ,\xi_{d+d1})$ if the states of $(\xi_1,\xi_2,L\ ,\xi_d)$ are fixed. Tabulate all the cases $(\xi_1,\xi_2,L\ ,\xi_d,L\ ,\xi_{d+d1},c)$ to give another transition table, namely the candidate transition table, which is different from the transition table consisted of $(\xi_1,\xi_2,L\ ,\xi_d,c)$.

Spatio-temporal systems are autonomous systems and there are no external inputs which influence the evolution. An identification procedure can therefore only be established based on the data from the observed spatio-temporal patterns. Hence, the transition table has to be obtained by extracting all cases from a spatio-temporal pattern one by one, which can be extremely tedious. In addition, because of the unknown exact neighbourhood, an initial candidate neighbourhood should be used, which makes the condition worse. Fortunately, a terse polynomial model was proposed in section 3. This model is equivalent to the candidate transition table and can describe the pattern exactly. Construct all the monomial terms $\xi_1^{\alpha_1}\xi_2^{\alpha_2}L\ \xi_{d+d1}^{\alpha_{d+d1}}, 0\le\alpha_i\le n-1$ arranging all the terms in an increasing order as a term vector, namely, $P = \begin{bmatrix} 1 & \xi_1^1 & \xi_2^1 & L & \xi_1^{n-1}\xi_2^{n-1}L\ \xi_{d+d1}^{n-1} \end{bmatrix}$. Hence the polynomial model v can be expressed as

$$v(\xi_1,\xi_2,L\ \xi_{d+d1}) = \begin{bmatrix} 1 & \xi_1^1 & \xi_2^1 & L & \xi_1^{n-1}\xi_2^{n-1}L\ \xi_{d+d1}^{n-1} \end{bmatrix}\begin{bmatrix} \theta_1 \\ \theta_2 \\ M \\ \theta_{n^{d+d1}} \end{bmatrix} \qquad (2)$$

Once data has been collected from the patterns the parameters $\begin{bmatrix} \theta_1 & \theta_2 & L & \theta_{n^{d+d1}} \end{bmatrix}^T$ can be estimated using a least squares based method.

Then based on the polynomial model above, construct a candidate transition table as in Table 3, which describes the pattern exactly, where the columns $\xi_1 \sim \xi_{d+d1}$ give all the state combinations of the neighbourhood and column c gives the corresponding values of function v.

Table 3 Candidate transition table

ξ_1	ξ_2	...	ξ_d	...	ξ_{d+d1}	c
0	0	...	0	...	0	$v(0,0,L\ ,0)$
0	0	...	0	...	1	$v(0,0,L\ ,1)$
Ν	Ν	Ν	Ν	Ν	Ν	Ν
n-1	n-1	...	n-1	...	n-1	$v(n-1,n-1,L\ ,n-1)$

4.2 Cyclic symmetry

Denote a cell in a spatio-temporal pattern as c, and the associated neighbourhood as R and a candidate neighbourhood as W, where $W \supset R$. Let $E = W - R$, which consists of cells that are in the candidate neighbourhood but not in the true neighbourhood. Denote the state set of this spatio-temporal pattern as $S = \{0, 1, L, n-1\}$. Define the operator $\oplus$ as the modulo n addition operator. Define a function $f(x)$, whose domain is the cell state set S, as cyclic symmetric about variable x if $f(x) = f(x \oplus 1)$ for $\forall x \in S$, in other words, $f(0) = f(1) = L = f(n+1)$. From the above discussion, the evolution of cell c does not depend on the cells in set E, that is, function $v(\xi_1, \xi_2, L \xi_d, L \xi_{d+d1})$ is cyclic symmetric about the cells in set E. Contrarily, function $v(\xi_1, \xi_2, L \xi_d, L \xi_{d+d1})$ is cyclic asymmetric about cells in neighbourhood R. This property can be used to eliminate the cells of E from the candidate neighbourhood W to determinate the exact neighbourhood R. For convenience, a normalised cyclic asymmetric loss function $J_{ca}^n(\xi_i)$ is introduced to evaluate the cyclic asymmetry with respect to ξ_i. This loss function can be defined as

$$J_{ca}^n(\xi_i) = \sum_k sign\left(w\left(v(\xi_1^{(k)}, L, \xi_i^{(k)}, L, \xi_{d+d1}^{(k)}) - v(\xi_1^{(k)}, L, \xi_i^{(k)} \oplus 1, L, \xi_{d+d1}^{(k)})\right)\right) \quad (3)$$

where k is the index of the data from the pattern and the function $w(x)$ is defined as $w(x) = 1$ if $|x| \leq \varepsilon$ and $w(x) = 0$, otherwise. This means that for the data $(\xi_1^{(k)}, L, \xi_i^{(k)}, L, \xi_{d+d1}^{(k)}, c^{(k)})$ from the spatio-temporal pattern, the loss function of ξ_i increases by 1, if the function $v(\xi_1^{(k)}, L, \xi_i^{(k)}, L, \xi_{d+d1}^{(k)})$ is cyclic asymmetry about ξ_i, otherwise, the value of the loss function will not change. Hence, if the initial values of the loss function about $\xi_1 \sim \xi_{d+d1}$ are set to zeros, those loss functions about the cells in set E will still be zero after all the data has been considered. For a noisy pattern a threshold J_{cutoff} should be selected. The neighbour ξ_j in the candidate neighbourhood should be eliminated if $J_{ca}^n(\xi_j) < J_{cutoff}$. In practice the state set S^{d+d1} and values $v(S^{d+d1})$ are used as the data set to evaluate the loss function in order to reduce the computational expense.

In practice, another problem must be considered. For a spatio-temporal system with n states and $d + d1$ candidate neighbours, the number of cases should be n^{d+d1}. This number will be enormous when n and $d + d1$ are very large. Because the data available will always be finite, the pattern may not include all these cases and this destroys the cyclic symmetry of the function. Hence a mask function is introduced to deal with the problem by differentiating the cases that are not included in the pattern and those that are included. The mask function can be constructed as follows. From the conclusion in section 3, n^{d+d1} Lagrange fundamental polynomials can be constructed with respect to an n-state, $d + d1$-neighbourhood spatio-temporal system. These Lagrange fundamental polynomials represent a 1-1 mapping to the cases of the spatio-temporal system. That is there is only one case which can make the value of the corresponding Lagrange fundamental polynomial equal to 1, while all the other cases make the values of the fundamental polynomials become zero. So these Lagrange

fundamental polynomials can be used as the mask function. Considering the values of a Lagrange fundamental polynomial based on all the data from a pattern, if the values are zero for all the conditions, the corresponding case is not included in this pattern.

4.3 Identification of *n*-state spatio-temporal patterns

The procedure for the identification of *n*-state spatio-temporal patterns can now be summarised as:

(1) Select a candidate neighbourhood W, which is larger enough to cover the potential neighbourhood.

(2) Construct the polynomial terms $P_{n^d \times 1} = \begin{bmatrix} p_1 & p_2 & L & p_{n^d} \end{bmatrix}$ and give the structure of the polynomial model $v(\xi_1, \xi_2, L, \xi_{d+d1}) = P\Theta$.

(3) Collect data from the spatio-temporal pattern and estimate the parameters Θ in the polynomial model.

(4) Eliminate the cases not included in the pattern from the candidate transition table using the mask function.

(5) Select the threshold value as $J_{cutoff} = \dfrac{1}{2(d+d1)} \sum_{j=1}^{d+d1} J_{ca}^n(\xi_j)$ and calculate the loss function of cyclic asymmetric $J_{ca}^n(\xi_i)$ about all the ξ_i's based on the candidate pseudo transition table. Eliminate the neighbour ξ_i from the candidate neighbourhood, if $J_{ca}(s_i) < J_{cutoff}$.

(6) Repeat step (5) until the correct neighbourhood is obtained.

(7) Using the correct neighbourhood from step (5), repeat step (2) and (3), to get the final polynomial model of the spatio-temporal pattern.

After obtaining the polynomial model, two kinds of prediction methods can then be used to predict the behaviours of the spatio-temporal system: one step ahead prediction and model predicted output or many steps ahead prediction. Model predicted output is a more strict criteria for evaluation the performance of the estimator than the one step ahead prediction and is used to validate the identified model in this paper.

5. Simulation Analysis

For simplicity, in this section only one example of a 3-state, 3-neighourhood 1-dimensional spatio-temporal noise free pattern and two kinds of noisy patterns will be considered. But lots of simulations have been done to illustrate the validity of these new methods, including examples with more states and more neighbours, including examples of 2-dimensional patterns.

5.1 Effects of noise

Noise in spatio-temporal systems is different from that in traditional dynamical systems. The difference is that the magnitude of the noise in a spatio-temporal system is equal to the magnitude of the signal, that is, the signal-to-noise ratio is always 1. Hence, the proportion of cells whose states become flipped due to noise is used to quantify the spatio-temporal noise instead of the traditional signal-to-noise-ratio.

There are two kinds of noise in spatio-temporal systems due to different origins: static noise and dynamical noise. Static noise, introduced by external factors, does not involve the evolution of the spatio-temporal pattern. It can be added after the evolution has finished. Unlike static noise, dynamical noise is introduced by internal factors and involves the evolution of the spatio-temporal pattern. Dynamical noise is added into the patterns as part of the evolution of the patterns and is a complex effect.
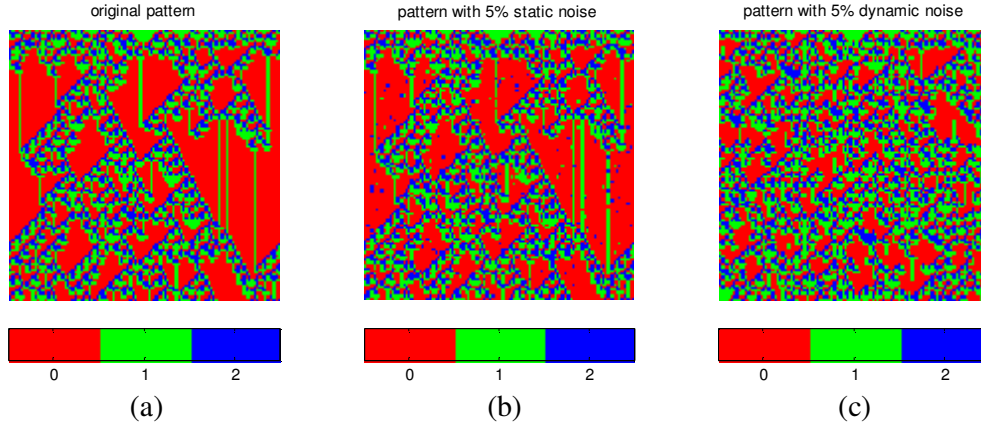

Figure. 1. The effects of static and dynamic noise.
(a) noise free pattern (b) pattern with 5% static noise (c) pattern with 5% dynamic noise

Figure 1 shows the effect of static noise and dynamic noise on a spatio-temporal pattern. The pattern disturbed by 5% dynamic noise in Figure 1 (c) looks significantly different from the original pattern Figure 1 (a) than the pattern disturbed by 5% static noise in Figure 1 (b). However simulation results shows that, compared with dynamical noise, static noise is more challenging for the identification of spatio-temporal patterns since the cells corrupted by dynamic noise continue to comply with the transition rules, while the patterns corrupted with static noise do not.

5.2 Simulation examples

Three examples are presented in this section. The first example, the identification of a 1-D, 3-state, 3-neighbourhood pattern, is described in more detail to show all the steps in the identification. For the other two examples, only the identification results are given for simplicity.

5.2.1 Identification of a 1-dimensional, 3-state, 3-neighbourhood pattern

A 1-dimentional, 3-state, 3-neighbourhood pattern with a von Neumann neighbourhood structure on a 100×100 lattice is shown in Figure 2 (a). Half the pattern in Figure 2 (a) was taken as the data set used for model identification. Denote a cell in this 1 dimensional pattern at position j and time step t as c_j^t. Select a candidate neighbourhood $W = \left\{ c_{j-2}^{t-1} \quad c_{j-1}^{t-1} \quad c_j^{t-1} \quad c_{j+1}^{t-1} \quad c_{j+2}^{t-1} \right\}$ denoted as $\{\xi_1, \xi_2, \xi_3, \xi_4, \xi_5\}$. Construct the 1×3^5 term vector $P = \left[1 \quad \xi_1 \quad \xi_2 \quad L \quad \xi_1^2 \xi_2^2 \xi_3^2 \xi_4^2 \xi_5^2 \right]$ and express the candidate transition

function $v(\xi_1,\xi_2,\xi_3,\xi_4,\xi_5)$ as a linear combination of these terms that is $v(\xi_1,\xi_2,\xi_3,\xi_4,\xi_5) = P\Theta$. Collect data from the pattern and estimate the parameter vector Θ. For simplicity, the results are not given here. The candidate transition table can then be constructed using the function v. There are 3^5 cases in this table. All these cases are then checked with the mask function. Seven cases of 243 are not included in this pattern and were eliminated from the candidate transition table. Calculating the cyclic asymmetry loss function about each cell in the candidate neighbourhood, gave the results in Table 4. Selecting half of the average value of all the $J_{ca}^n(\xi_j)$'s as the threshold value, that is $J_{cutoff} = \dfrac{1}{2\times5}\sum_{j=1}^{5} J_{ca}^n(\xi_j) = 67.62$. ξ_1 and ξ_5 were eliminated from the candidate neighbourhood because $J_{ca}^n(\xi_1), J_{ca}^n(\xi_5)$ were less than J_{cutoff}. The detected neighbourhood R is $\{\xi_2,\xi_3,\xi_4\}$, that is, the left shifted von Neumann structure. The polynomial model can then be built using the detected neighbourhood as

$$
\begin{aligned}
f(\xi_2,\xi_3,\xi_4) = {}& \xi_2 - \xi_2\xi_3 + 0.5\xi_2^2\xi_3 + \xi_2\xi_3^2 - 0.5\xi_2^2\xi_3^2 - \xi_4 + 7\xi_2\xi_4 - 4\xi_2^2\xi_4 + 7.5\xi_3\xi_4 \\
& -21.375\xi_2\xi_3\xi_4 + 9.375\xi_2^2\xi_3\xi_4 - 3\xi_3^2\xi_4 + 8.875\xi_2\xi_3^2\xi_4 - 3.875\xi_2^2\xi_3^2\xi_4 \\
& +\xi_4^2 - 4\xi_2\xi_4^2 + 2\xi_2^2\xi_4^2 - 4\xi_3\xi_4^2 + 12.375\xi_2\xi_3\xi_4^2 - 5.375\xi_2^2\xi_3\xi_4^2 + 1.5\xi_3^2\xi_4^2 \\
& -5.375\xi_2\xi_3^2\xi_4^2 + 2.375\xi_2^2\xi_3^2\xi_4^2
\end{aligned}
\tag{4}
$$

Using the initial condition of the pattern in Figure 2 (a) as the initial condition with the polynomial model as the rule, the model predicted output pattern is shown in Figure 2 (b).

Table 4 Value of cyclic asymmetry loss function (J_{cutoff} = 67.62)

$J_{ca}^n(\xi_1)$	$J_{ca}^n(\xi_2)$	$J_{ca}^n(\xi_3)$	$J_{ca}^n(\xi_4)$	$J_{ca}^n(\xi_5)$
0	177	148	158	0

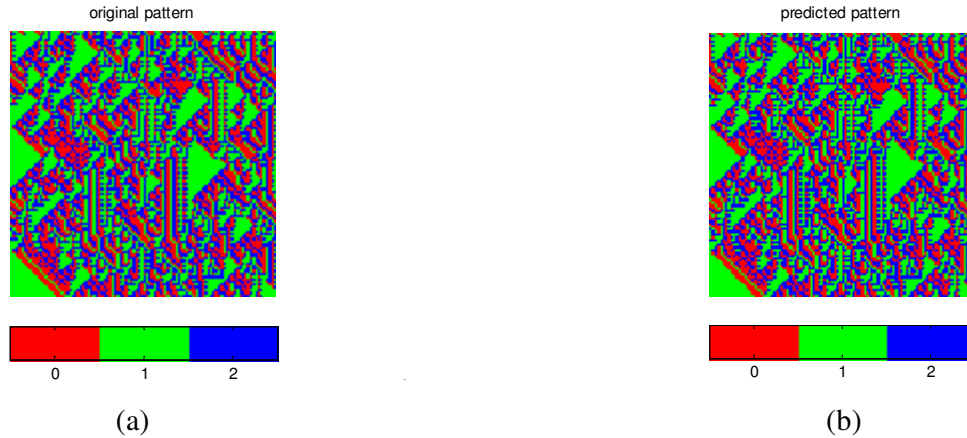(a)   (b)

Figure 2 1-dimensional, 3-state, 3-neighbourhood spatio-temporal pattern (a) the data set used in the identification; (b) the predicted pattern using the identified model

5.2.2 Identification of noisy patterns

In Figure 3 (a), a 1-dimensional, 3-state, 3-neighbourhood spatio-temporal is shown. Then 5% static noise was added to the pattern in Figure 3 (a) and the noisy pattern is shown in figure 3 (b). Half the pattern in figure 3 (b) was used as the data set for model identification. The values of loss functions about every cell in the candidate neighbourhood are given in Table 5. It is shown in Table 5 that cells ξ_1 and ξ_5 should be eliminated from the candidate neighbourhood to give the true neighbourhood with a von Neumann structure. The model predicted output pattern based on the identified model is shown in figure 3 (c).

original pattern   noised pattern (static 5%)   predicted pattern
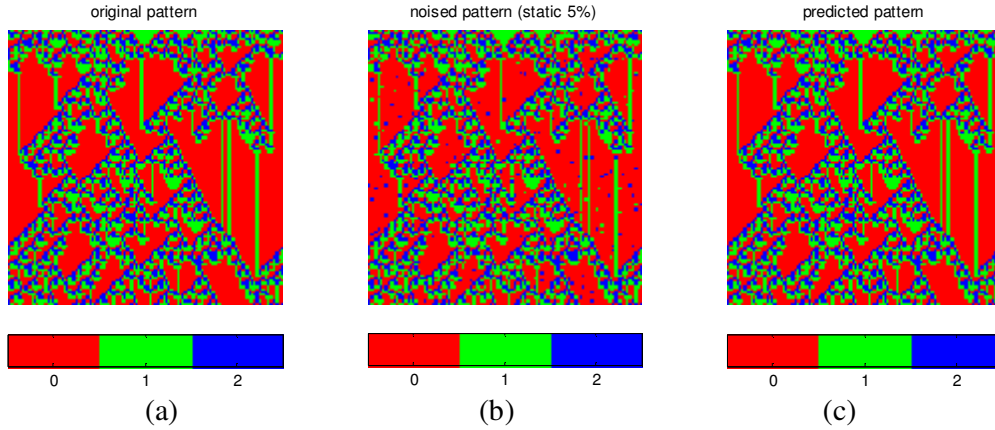
(a)   (b)   (c)

Figure 3 1-dimentional, 3-state, 3-neighbourhood spatio-temporal pattern
(a) the original pattern; (b) 5% static noisy pattern; (c) the predicted pattern

Table 5 Value of cyclic asymmetry loss function (J_{cutoff} = 68.88)

$J_{ca}^{n}(\xi_1)$	$J_{ca}^{n}(\xi_2)$	$J_{ca}^{n}(\xi_3)$	$J_{ca}^{n}(\xi_4)$	$J_{ca}^{n}(\xi_5)$
22	152	147	147	24

The same pattern as in Figure 3 (a) is given in Figure 4 (a), but now the pattern was corrupted by 10% dynamic noise to give the pattern in Figure 4 (b). This noisy pattern was then used as the data set to identify the model. Table 6 shows that this pattern was a von Neumann structure. The model predicted output pattern based on the identified model is given as Figure 4 (c).

original pattern   noised pattern (dynamic 10%)   predicted pattern
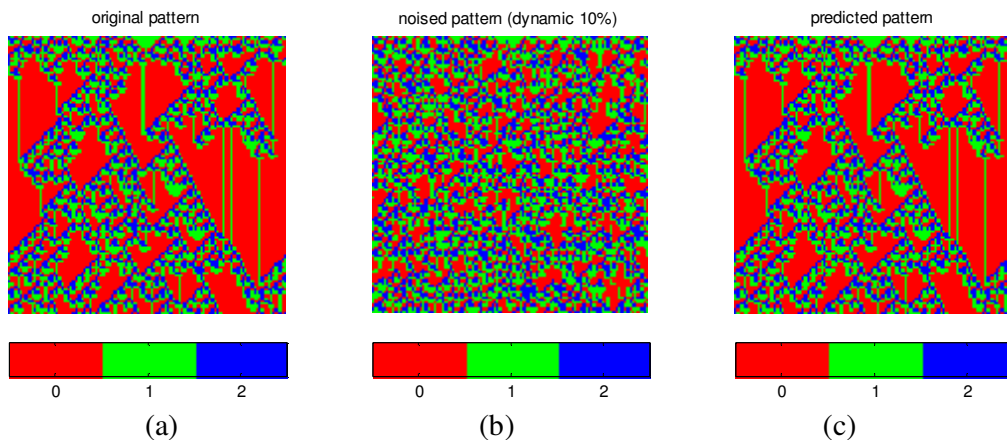
(a)   (b)   (c)

Figure 4 1-dimensional, 3-state (0-red, 1-green, 2-blue), 3-neighbourhood pattern (a) the original pattern; (b) the 10% dynamical noisy pattern; (c) the predicted pattern

Table 6 Value of cyclic asymmetry loss function (J_{cutoff} = 72.66)

$J_{ca}^{n}(\xi_1)$	$J_{ca}^{n}(\xi_2)$	$J_{ca}^{n}(\xi_3)$	$J_{ca}^{n}(\xi_4)$	$J_{ca}^{n}(\xi_5)$
0	186	170	160	3

6. Conclusions

A complete solution to the identification of n-state spatio-temporal systems has been introduced in this paper including algorithms for neighbourhood detection and the determination of the transition rules. The linear-in-the-parameters polynomial model provides a simple and accurate representation of n-state spatio-temporal systems.

The simulation results show that the neighbourhood detection method is effective and is not sensitive to less than 10% dynamic noise (evolutionary noise), and 5% static noise (measurement noise). However this does not mean the polynomial model extracted based on the correct neighbourhood is accurate enough to exactly predict the pattern, because some transition rules may be more sensitive to noise than others and an exact estimation may only be extracted at lower noise levels. It has also been shown that the neighbourhood detection method in this paper is effective for spatio-temporal patterns which are incomplete.

References

ADAMATZKY, A. I. (1994) *Identification of Cellular automata,* New York, Taylor & Francis.

BAIER, G., MÜLLER, M. & ØRSNES, H. (2002) Excitable spatio-temporal chaos in a model of glycolysis. *Journal of Physical Chemistry B,* 106**,** 3275-3282.

BILLINGS, S. A. & YANG, Y. (2003) Identification of the neighborhood and CA rules from spatio-temporal CA patterns. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics,* 33**,** 332-339.

ERMENTROUT, B. (1998) Neural networks as spatio-temporal pattern-forming systems. *Reports on Progress in Physics,* 61**,** 353-430.

FILIPE, J. A. N. & GIBSON, G. J. (1998) Studying and approximating spatio-temporal models for epidemic spread and control. *Philosophical Transactions of the Royal Society of London Series B Biological Sciences,* 353**,** 2153-2162.

MAEDA, K. I. & SAKAMA, C. (2003) Discovery of Cellular Automata Rules Using Cases. *LECTURE NOTES IN COMPUTER SCIENCE***,** 360-368.

MEI, S. S., BILLINGS, S. A. & GUO, L. Z. (2005) A neighborhood selection method for cellular automata models. *International Journal of Bifurcation and Chaos in Applied Sciences and Engineering,* 15**,** 383-393.

REKECZKY, C., TAHY, A., VEGH, Z. & ROSKA, T. (1997) CNN-based spatio-temporal nonlinear filtering and endocardial boundary detection in echocardiography. *International Journal of Circuit Theory and Applications,* 27**,** 171-207.

RICHARDS, F. C., MEYER, T. P. & PACKARD, N. H. (1990) Extracting cellular automaton rules directly from experimental data. *Physica D: Nonlinear Phenomena,* 45**,** 189-202.

YANG, Y. & BILLINGS, S. A. (2000a) Extracting Boolean rules from CA patterns. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics,* 30**,** 573-581.

YANG, Y. & BILLINGS, S. A. (2000b) Neighborhood detection and rule selection from cellular automata patterns. *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans.,* 30**,** 840-847.

ZHAO, Y. & BILLINGS, S. A. (2006) Neighborhood detection using mutual information for the identification of cellular automata. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics,* 36**,** 473-479.