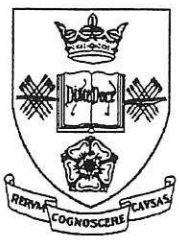
A New Fast Cellular Automata Orthogonal Least Squares Identification Method



S. A.Billings S. S. Mei

Department of Automatic Control and System Engineering
The University of Sheffield

Research Report No. 838

May, 2003

A New Fast Cellular Automata Orthogonal Least Squares Identification Method

S. A. Billings, S. S. Mei
Department of Automatic Control and System Engineering
University of Sheffield
Mappin Street, Sheffield S1 3JD
United Kingdom

Abstract

A novel fast cellular automata orthogonal least squares (FCA-OLS) identification method is introduced by extending and developing the CA-OLS identification method presented in [2]. As a simulation example, cellular automata rule 110 is analysed and identified. The simulation results show that the new method significantly reduces the computational time compared to existing methods.

Keywords: cellular Automata, identification, polynomial, orthogonal least squares

1. Introduction

Cellular Automata (CA) were originally introduced by von Neumann and Ulam as a possible idealization of biological systems [1], with the particular purpose of modelling biological self-reproduction. Cellular automata have been applied to model spatio-temporal systems in a wide variety of fields, including physics, biology, computer science, image processing, etc. Recently the new book by Wolfram [7], which includes many new research results, has stimulated further interest in CA. In most cases studies of CA involve showing how incredible complexity can be produced by seemingly simple CA rules, or using CA models to make discrete simulations of complex natural processes. Many such rule sets have been investigated, simulated and studied. But all these studies rely on the apriori knowledge of the rule or the non-linear underlying model, and only a few results have been reported on CA identification [3, 5, 9, 14]. That is given the CA pattern how do we find the CA rule. The main reason why this problem has been ignored is probably because of the complexity induced by both the spatial and temporal interactions but this is one of the core problems of spatio-temporal systems.

At present, there are only a few methods for identifying cellular automata rules from data and most of these assume a known structure. Andre *et. al.* [9] used genetic programming with automatically defined functions to evolve a rule for the majority classification task for one-dimensional cellular automata, and Yang and Billings [2] introduced a method using Genetic Algorithms. However, no clear structure of the related neighbourhoods was obtained and the detection process was complicated and very time consuming. Billings and Yang 2003a proposed a Cellular Automata Orthogonal Least Squares algorithm (CA-OLS). These authors mapped the binary rules into simple polynomial forms, and then developed the CA-OLS to determine the neighbourhood and the unknown model parameters. The identified polynomial model was then mapped back to a logical expression. The CA-OLS method produced for the first time a powerful method for determining the rules of high-dimensional CA in the form of a parsimonious model. This was achieved based only on the observations of the data or CA patterns and behaviours.

The core orthogonal least squares method, which was adapted to CA identification in [2], was improved in [12] [13]. It was pointed out that the orthogonalisation could be carried out using simplified iterations instead of time expensive computations at every step [12]. The modified algorithm is computationally less expensive. Another faster algorithm was derived on the basis of computing the estimates using correlation computations instead of the orthogonal terms [13]. In the present study, these two improvements will be combined and extended to produce a new fast CA-OLS algorithm which is tailored to CA identification. This new algorithm significantly reduces the computation time compared to the original CA-OLS method in [2].

The paper is organized as follows. In section 2, background results on cellular automata are introduced. The CA-OLS method presented in [2] is reviewed in section 3. In section 4 the new fast CA-OLS method is introduced. Simulation results are listed in section 5, and conclusions are given in section 6.

2. Cellular Automata

Cellular automata which were first introduced by von Neumann as a "cellular space" for modelling biological self-reproduction, were later studied extensively by Wolfram at the beginning of the 1980s [6]. The CA structure investigated by Wolfram [11] can be viewed as a discrete lattice of sites (cells) where each cell can take the value 0 or 1, called binary CA. Attention in this study will be restricted to binary CA. The next state of a cell in a cellular automaton is assumed to depend on its neighbourhood and the transition rule.

A cellular automaton consists of three parts: a neighbourhood, a local transition rule and a discrete lattice structure consisting of a large number of cells which are occupied by states from a finite set of discrete values. The neighbourhood of a cell is the set of all the cells capable of directly influencing the evolution of that cell, which sometimes includes the cell itself. The local transition rule updates all cells synchronously, at a given time step, by assigning to each cell a value according to the cell's neighbourhood.

Consider a d-dimensional lattice L which consists of a set of all integer coordinate vectors $j=(j_1,\cdots,j_d)\in \mathbf{Z}^d$. Defined over lattice L, the rule of the n-cell CA can be expressed as follows

$$x_j(t)=f(N(x_j(t))) \tag{1}$$

where $x_j(t)\in \mathbf{B}$ is the updated state of the jth cell in the lattice L at time step t, f is the transition function describing the local transition rule, and $N(x_j(t))$ is the neighbourhood of the jth cell in L at time step t, which is defined as below

$$\begin{aligned} N(x_j(t)) = (&x_{j+p_1(1)}(t-1), x_{j+p_2(1)}(t-1),\cdots, x_{j+p_{m(1)}(1)}(t-1),\cdots,\\ &x_{j+p_1(2)}(t-2), x_{j+p_2(2)}(t-2),\cdots, x_{j+p_{m(1)}(2)}(t-2),\cdots,\\ &x_{j+p_1(\tau)}(t-\tau), x_{j+p_2(\tau)}(t-\tau),\cdots, x_{j+p_{m(\tau)}(\tau)}(t-\tau))\\ =&(c_1^j(t),\cdots,c_l^j(t),\cdots,c_n^j(t)) \end{aligned} \tag{2}$$

where $c_l^j(t)$, standing for $x_{j+p_i(k)}(t-k)$, $(1 \le k \le \tau,\ 1 \le i \le m(k)$, and $\sum_{k=1}^{\tau} m(k) = n)$ is the state of the lth $(l = i + \sum_{1 \le k' < k} m(k'))$ entry in the neighbourhood region of the jth cell at time step t, $p_i(k)$ is the integer coordinate difference vector between the jth cell and the lth entry in the corresponding neighbourhood region, which specifies the spatial location of the neighbourhood cell, and k gives the temporal location of the corresponding neighbourhood cells.

The function f in function (1) is usually represented as a Boolean function of the cells within the neighbourhood. Billings *et. al.* [2] showed that this transition function can also be expressed as an exact integer parameter polynomial, which is linearly combined with neighbourhood product terms. Eqn (1) can therefore be reformed as the following polynomial form:

$$x_j(t) = \phi(t)\Theta \tag{3}$$

where $\phi(t)$ is $1 \times N$ matrix with each entry composed of the product of the subset of the neighbourhood at evolution time step t, $\phi_i(t) = \prod_{l_i \in L_i} c_{l_i}^j(t)$, $L_i \subseteq \{1, 2, \cdots, n\}$ is the neighbourhood indices subset ($L_i \ne L_j$, for all $i \ne j$, $1 \le i \le N$, $1 \le j \le N$), $N \le 2^n$ is the number of model regressors, and $\Theta = [\theta_1\ \theta_2 \ldots \theta_N]^T$ is an $N \times 1$ integer parameter vector.

3. The CA-OLS Identification Method

Because binary CA rules can be reformulated into a polynomial form, the identification of CA rules is equivalent to estimating the parameter vector Θ in equation (3). The candidate model size N can be excessively large. Therefore subset model selection is necessary, and an efficient method is the Cellular Automata Orthogonal Least Squares (CA-OLS) algorithm proposed in [2] which is briefly reviewed below.

Consider M evolution steps, the transition model can then be written in the following form

$$\mathbf{x}_j = \Phi\Theta \tag{4}$$

where $\mathbf{x}_j = [x_j(1)\ x_j(2) \cdots x_j(M)]^T$ is the measurement vector including values of the jth cell updated at M different time steps, and $\Phi = [\phi(1)\ \phi(2) \cdots \phi(M)]^T$ are the corresponding neighbourhood products at the different time steps.

Consider the orthogonal decomposition of $\Phi = EQ$ based on the Gram-Schmidt transformation method, where $E = [\mathbf{e}_1 \ldots \mathbf{e}_k \ldots \mathbf{e}_N]$ is an orthogonal matrix, $\mathbf{e}_k$ are the columns of E

$$\mathbf{e}_k = \phi_k - \sum_{r=0}^{k-1} q_{rk}\mathbf{e}_r \tag{5}$$

and Q is an upper triangular matrix with unity diagonal elements, with entries

$$q_{ik} = \mathbf{e}_i^T \phi_k / \mathbf{e}_i^T \mathbf{e}_i \quad (i<k,\ 1 \le i, j \le N) \tag{6}$$

Equation (4) can then be represented as $\mathbf{x}_j = EQ\Theta = E\tilde{\Theta}$

where

$$\tilde{\Theta} = Q\Theta = [\tilde{\theta}_1 \quad \ldots \quad \tilde{\theta}_n \quad \ldots \quad \tilde{\theta}_N] \tag{7}$$

with each entry given as

$$\tilde{\theta}_k = \mathbf{e}_k^T \mathbf{x}_j / \mathbf{e}_k^T \mathbf{e}_k \tag{8}$$

The contribution of each candidate term can be assessed using Error Reduction Ratio (ERR). The contribution due to $\mathbf{e}_d$ (d=1,..., N) [2] is given by

$$[err]_d = \tilde{\theta}_d^2 \mathbf{e}_d^T \mathbf{e}_d / \mathbf{x}_j^T \mathbf{x}_j \tag{9}$$

The ERR provides a criterion for forward subset selection, where the regressors can be ranked according to the contribution they make to the model. The selection is terminated at the N_sth step either when $1 - \sum_{d=1}^{N_s} [err]_d < C_{off}$ where C_{off} is preset tolerance, $N_s < N$ or when $N_s = N$.

Once N_s significant regressors have been selected, the corresponding parameters Θ can be calculated

$$\theta_{N_s} = \tilde{\theta}_{N_s}$$

$$\theta_m = \tilde{\theta}_m - \sum_{k=m+1}^{N_s} q_{mk} \theta_k, \; m = N_s - 1, \ldots, 1. \tag{10}$$

4. A New Fast CA-OLS Algorithm

The above CA-OLS algorithm is built upon the orthogonal least squares algorithm [16] that was subsequently used to select wavelet regressors in [12]. In [12] it was shown that the sum in equation (7) need not be calculated at every time step but can be replaced by

$$p_k^d(t) = p_{k-1}^d(t) - q_{(k-1)k} e_{(k-1)} \qquad \forall d \in I_k$$

with

$$p_0^d(t) = \phi_d(t) \qquad \forall d \in I_0 \tag{11}$$

The modified orthogonalisation algorithm that results is mathematically equivalent to the original algorithm but it is computationally less expensive.

Zhu and Billings [13] suggested another improvement to the original orthogonalisation algorithm. They demonstrated that the calculation in (5) can be substituted by the products $\mathbf{p}_i^T \mathbf{x}_j$ and $\mathbf{p}_i^T \mathbf{p}_i$ and that further calculations (6), (8), and (9) can be derived on the basis of computing the estimates using correlation computations instead of the orthogonal terms.

In this study, based on [13], the algorithm presented in [16] and [12] is further improved by substituting the products $\mathbf{p}_i^T \mathbf{x}_j$ and $\mathbf{p}_i^T \mathbf{p}_i$ into eqn (11) and introducing the correlation

computations into the estimates as well. Adapting these results to the CA case produces the new fast FCA-OLS algorithm which is derived below.

Define

$$R_{px}^{(k)}(d)=\mathbf{p}_d^{(k)T}\,\mathbf{x}_j \tag{12}$$

$$R_{pp}^{(k)}(i,j)=\mathbf{p}_i^{(k)T}\mathbf{p}_j^{(k)} \tag{13}$$

Multiplying by $\mathbf{x}_j$ from both sides and Substituting (6), (12) and (13) into (11) yields

$$R_{px}^{(k)}(d)=R_{px}^{(k-1)}(d)-\frac{(R_{pp}^{(k-1)}(d,l_{k-1}))(R_{px}^{(k-1)}(l_{k-1}))}{R_{pp}^{(k-1)}(l_{k-1},l_{k-1})} \tag{14}$$

Similarly, the iterative equation regarding to $R_{pp}^{(k)}(i,j)$ can also be obtained as

$$R_{pp}^{(k)}(i,j)=R_{pp}^{(k-1)}(i,j)-\frac{(R_{pp}^{(k-1)}(i,l_{k-1}))(R_{pp}^{(k-1)}(j,l_{k-1}))}{R_{pp}^{(k-1)}(l_{k-1},l_{k-1})} \tag{15}$$

Then equations (6), (8) and (9) can be replaced by

$$q_{ik}=R_{px}^{(i)}(l_k,l_i)\big/R_{pp}^{(i)}(l_i,l_i) \qquad i\in[1,k-1] \tag{16}$$

$$\tilde{\theta}_k=R_{px}^{(k)}(l_j)\big/R_{pp}^{(k)}(l_j,l_j) \tag{17}$$

$$[ct]_d=(R_{px}^{(k)}(d))^2\big/R_{pp}^{(k)}(d,d) \qquad \forall d\in I_k \tag{18}$$

Based on the above improvements, a new fast CA-OLS algorithm is obtained. This new algorithm is summarised below:

1) Set the initial regressor indices as $I_1=\{1,2,...,N\}$, and compute

$$R_{pp}^{(1)}(d,d)=\sum_{\tau=1}^{M}(p_d^{(1)}(\tau))^2 \qquad \forall d\in I_1 \tag{19}$$

$$R_{px}^{(1)}(d)=\sum_{\tau=1}^{M}(p_d^{(1)}(\tau)x_j(\tau) \qquad \forall d\in I_1 \tag{20}$$

Select the regressors using

$$l_1=\arg\max_{d\in I_1}\left[[ct]_1^{(d)}\right]=\arg\max_{d\in I_1}\left[\frac{(R_{px}^{(1)}(d))^2}{R_{pp}^{(1)}(d,d)}\right] \qquad \forall d\in I_1 \tag{21}$$

$$\tilde{\theta}_1^{d}=\frac{R_{px}^{(1)}(d)}{R_{pp}^{(1)}(d,d)} \qquad \forall d\in I_1 \tag{22}$$

2) Update the regressor indices $I_k = I_{k-1} \setminus \{l_{k-1}\}$, and compute

$$R_{pp}^{(k)}(d,d) = R_{pp}^{(k-1)}(d,d) - \frac{(R_{pp}^{(k-1)}(d,l_{k-1}))^2}{R_{pp}^{(k-1)}(l_{k-1},l_{k-1})} \quad \forall d \in I_k \tag{23}$$

$$R_{px}^{(k)}(d) = R_{px}^{(k-1)}(d) - \frac{R_{pp}^{(k-1)}(d,l_{k-1})}{R_{pp}^{(k-1)}(l_{k-1},l_{k-1})} R_{px}^{(k-1)}(l_{k-1}) \quad \forall d \in I_k \tag{24}$$

Select the kth regressor with the largest contribution

$$l_k = \arg\max_{d \in I_k} \left[[ct]_k^{(d)} \right] = \arg\max_{d \in I_k} \left[\frac{(R_{px}^{(k)}(d))^2}{R_{pp}^{(k)}(d,d)} \right] \tag{25}$$

Prepare for the next selection step and compute

$$R_{pp}^{(1)}(d,l_k) = \sum_{\tau=1}^{M} p_d^{(1)}(\tau) p_{l_k}^{(1)} \quad \forall d \in I_K \tag{26}$$

$$R_{pp}^{(i)}(d,l_k) = R_{pp}^{(i-1)}(d,l_k) - \frac{R_{pp}^{(i-1)}(d,l_{i-1}) R_{pp}^{(i-1)}(l_k,l_{i-1})}{R_{pp}^{(i-1)}(l_{i-1},l_{i-1})} \quad \forall d \in I_k,\ i = 2,\ldots,k \tag{27}$$

$$\tilde{\theta}_k^d = \frac{R_{px}^{(k)}(d)}{R_{pp}^{(k)}(d,d)} \quad \forall d \in I_k \tag{28}$$

$$q_{ik} = \frac{R_{pp}^{(i)}(l_k,l_i)}{R_{pp}^{(i)}(l_i,l_i)} \quad i \in [1,k-1] \tag{29}$$

3) The procedure is terminated at the N_sth step either when $1 - \sum_{d=1}^{N_s} [ct]_d < C_{off}$, where C_{off} is a desired tolerance, $N_s < N$, or when $N_s = N$.

4) Calculate the corresponding N_s parameter estimates using

$$\theta_{N_s} = \tilde{\theta}_{N_s}, \quad \theta_m = \tilde{\theta}_m - \sum_{k=m+1}^{N_s} q_{mk} \theta_k, m = N_s - 1, \ldots, 1.$$

5. Simulation Studies

Simulation examples will be used to demonstrate that the new FCA-OLS estimator produces exactly the same results as the old CA-OLS, and to compare the computation speed of both methods.

5.1 Identification Of CA Rule 110

The one dimensional three cell (neighbourhood radius r=1) cellular automata rule 110 has been extensively studied by many authors in the cellular automata community [6, 7, 15], rule 110 is one of the simplest 'complex' or 'class-4' rules as defined by the phenomenological classification scheme for cellular automata introduced by Wolfram in [6]. The rule number label 110, follows the convention introduced in [11], and is the decimal representation of the string (01101110) which encodes the three-inputs, one-output rule.

Table 1. CA rule 110

111	110	101	100	011	010	001	000
0	1	1	0	1	1	1	0

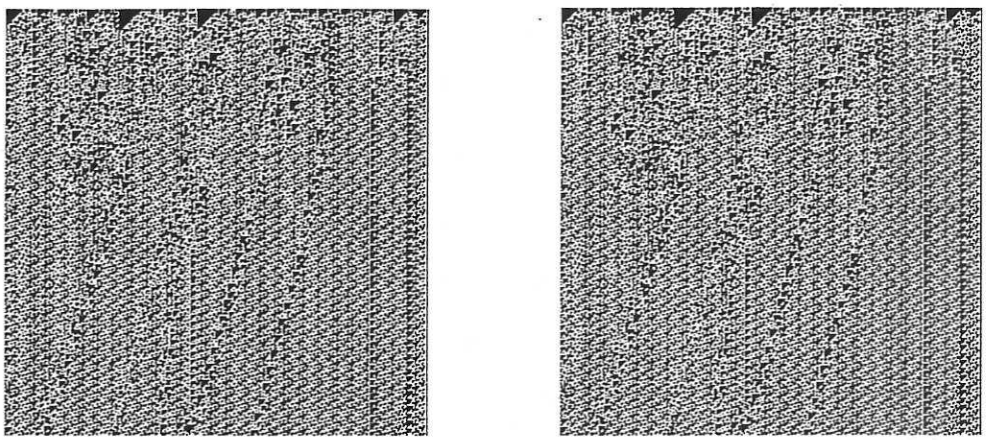
Fig. 1(a) shows an example of the complex spatio-temporal pattern generated by rule 110 with a random initial condition. Periodic boundary conditions were used, which means the first cell and the last one in the lattice are identical. The particle-like structure of gliders can be distinguished from the background, which is periodic with spatial periodicity 14 and temporal periodicity 7. A list of gliders for rule 110 is given in Table15 in the appendix of [6]. The identification data were extracted from the patterns shown as Fig. 1(a) to identify a CA model using the above new FCA-OLS method. The identification data consisted of 100 data points of input/output data ($x_{j-1}(t-1), x_j(t-1), x_{j+1}(t-1)$) and $x_j(t)$ from the spatial location (x_4, x_5, x_6). The identified polynomial equation corresponding to the output $x_j(t)$ is listed in Table 2.

Table 2. Estimated result of CA rule 110

Cell to be updated	Terms	Estimates	ERRs	STDs
$x_j(t)$	$x_j(t-1)$	0.10000E+01	0.49831E+00	0.26667E-16
	$x_{j-1}(t-1)x_j(t-1)x_{j+1}(t-1)$	-0.10000E+01	0.21356E+00	0.42936E-16
	$x_{j+1}(t-1)$	0.10000E+01	0.17812E+00	0.17778E-16
	$x_j(t-1)x_{j+1}(t-1)$	-0.10000E+01	0.11002E+00	0.34097E-16
	$x_{j-1}(t-1)x_{j+1}(t-1)^*$	0.17462E-30	0.21888E-62	0.25915E-16

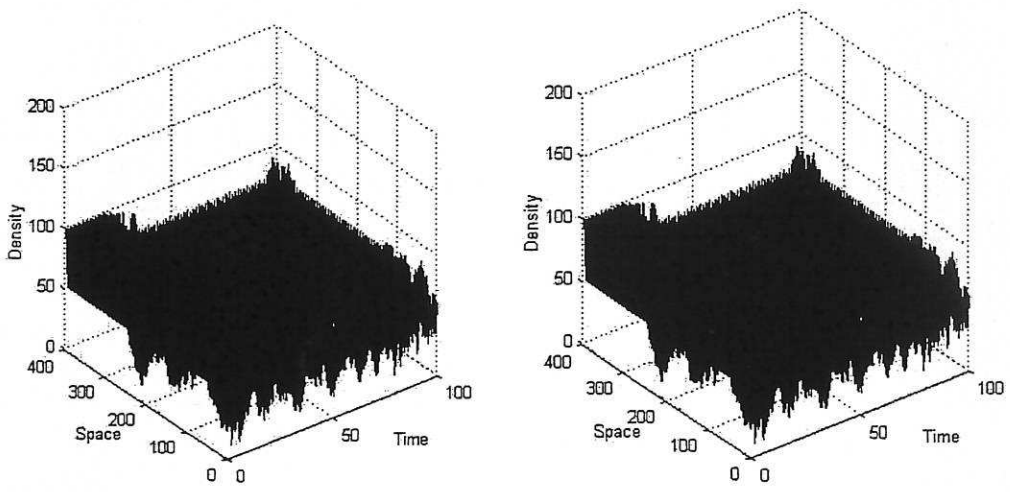
*This term was not selected because of the extremely small ERR value for this term.

The identified polynomial equation was used as the CA rule to implement a one-dimensional CA model with the same initial configuration on a one dimensional 400-cell lattice. The periodic boundary conditions were used. This produced the model predicted pattern, which is shown in Fig. 1(b). From this model predicted pattern it can be seen that the particle-like structure of gliders and background which appear in the pattern shown in Fig. 1(a) is replicated in the CA model produced output in Fig. 1 (b). This shows an excellent agreement with the original pattern produced from the original data based on CA rule 110.



(a) (b)

Fig. 1 A comparison of the patterns. The lattice size is 400, and the number of time steps is 400. (a)The spatio-temporal pattern produced by CA rule 110; (b)The predicted from the identified model.



(a) (b)

Fig. 2 Comparison of the density ρ_{it} of sites with value 1
(a) Density of the pattern shown in Fig. 1(a); (b) Density of the pattern shown in Fig. 1((b).

To quantitatively assess the performance of the identified CA rule, the statistical fraction (density) method was analysed [11] and used to compare the identified CA rule predicted output to the original CA pattern. In this study the fraction, denoted by ρ_{it}, was defined by

$$\rho_{it} = \#_1(r_i,t)/(\#_0(r_i,t)+\#_1(r_i,t)) \tag{30}$$

where $\#_d(r_i,t)$ denotes the number of cells with digit d in the region r_i configuration at time step t. For example, if the configuration of a region r_0, which is [1 20], in a lattice at time step t_0 is (00001110101010100000), $\#_1(r_0,t_0)=7$, $\#_0(r_0,t_0)=13$. In this study, the region r_i of cell i was set as $[i-10 \;\; i+10]$, in which 21 cells are included, and the periodic boundary conditions were applied for the edge cells. The densities ρ_{it} illustrated in Fig. 2 were obtained from the identified model predicted pattern and the original pattern produced with CA rule 110 shown in Fig. 1. The result shows that the both densities are in very good agreement.

5.2 Computation Time Comparison

Several other examples of CA patterns produced with different rules were also simulated. All the results showed this new fast algorithm produces exactly the same estimates as the old CA-OLS algorithm. To investigate the computational effort of the new fast FCA-OLS method, a program was designed to compare the new algorithm to the routine presented in [2]. Both methods were run on a Sun workstation under the same conditions. Table 3 shows that the new fast FCA-OLS routine results in a significant reduction in the computational time.

Table 3 Comparison result of computational cost

Data points	Dimension	Neighbourhood	Time(CA-OLS)	Time(New FCA-OLS)
500	1	3	100%	47.61%
1000	2	5	100%	46.87%
5000	3	7	100%	46.01%

6. Conclusions

It has been demonstrated that simple CA models can produce complex spatio-temporal patterns, and that these models can be identified in a polynomial form using the CA-OLS estimator presented in [2]. A new fast FCA-OLS estimator has been introduced in this study. Simulation results show the efficiency of the new fast CA-OLS estimator which produces exactly the same estimates as the CA-OLS in [2]. The advantage of the new method is that the computational time is significantly reduced. Additionally, this advantage does not result in a greater memory requirement.

Acknowledgement

SAB gratefully acknowledges that part of this work was supported by the Engineering and Physical Science Research Council, UK. SSM acknowledges the support of the University of Sheffield under the university scholarship scheme.

Reference:

[1] J. von Neumann, (1951), The general logical theory of automata, *Cerebral Mechanisms in Behaviour*—The Hixon Symposium, New York Wiley.

[2] Billings S. A. and Yang, Y., (2003a), Identification of the neighbourhood and CA Rules from Spatio-temporal CA patterns, *IEEE Transactions on Systems Man and Cybernetics, Part B*, 33, 332-339.
[3] Billings S. A. and Yang, Y., (2003b), Identification of probabilistic cellular automata, *IEEE Transactions on Systems Man and Cybernetics, Part B*, 33, 225-236.
[4] Yang Y. and Billings SA, (2000a), Extracting Boolean rules from CA patterns, *IEEE Transaction on System, Man, Cybernetics* B 30 (4): 573-581.
[5] Yang, Y. and Billings, S. A., (2000b), Neighbourhood detection and rule selection from cellular automata patterns, *IEEE Transactions on Systems Man and Cybernetics, Part A* 30 (6): 840-847.
[6] Wolfram S., (1986), *Theory and applications of cellular automata,* World Scientific, Singapore.
[7] Wolfram S., (2002), *A New Kind of Science*, Wolfram Media.
[8] Koza, J. R., (1994), Genetic programming ll: Automatic discovery of reusable programs, MIT Press, Cambridge, MA.
[9] Andre, D. Bennet, F. H. III, and Koza, R. R., (1996), Discovery by genetic programming of a cellular automata rule that is better than any known rule for the majority classification problem, in Koza, J. R. (Ed.), *Genetic Programming 1996, Proceedings of the 1st Annual conference*, MIT Press, Cambridge, MA.
[10] Wolfram S., (1984), Universality and complexity in cellular automata, *Physica D* 10 (1).
[11] Wolfram S., (1983), Statistical mechanics of cellular automata, *Reviews of Modern Physics*, Vol. 55, July 1983, pp. 601-644.
[12] Zhang, Q., 1994, Using wavelet network in non parametric estimation. *IRISA, France, Publication Interne,* 32p.
[13] Zhu, Q.M., Billings, S. A., (1996), Fast identification of nonlinear stochastic models and radial basis function neural networks, *International Journal of Control,* 64,871-886.
[14] Adamatzky A., (1997), Automatic programming of cellular automata: identification approach, *Kybernetes*, 26 (2-3): 126-133.
[15] Li W. and Nordahl M. G., (1992), Transient behaviour of cellular automataon rule 110, *Physics letters A*,166, 335-339.
[16] Billings S. A., Chen S., Korenberg M. J., Identification of MIMO nonlinear system using a forward regression orthogonal estimator, *International Journal of Control*, 49, 1989, 2157-2189.