Higher-order Fuzzy Cognitive Maps*

Wojciech Stach
Department of Electrical and Computer Engineering
University of Alberta
Edmonton, Alberta, T6G 2V4, Canada
wstach@ece.ualberta.ca

Lukasz Kurgan
Department of Electrical and Computer Engineering
University of Alberta
Edmonton, Alberta, T6G 2V4, Canada
lkurgan@ece.ualberta.ca

Witold Pedrycz
Department of Electrical and Computer Engineering
University of Alberta
Edmonton, Alberta, T6G 2V4, Canada
pedrycz@ece.ualberta.ca

***Abstract* – FCMs are aimed at modeling and simulation of dynamic systems. They exhibit numerous advantages, such as model transparency, simplicity, and adaptability to a given domain, to name a few. FCMs have been applied to numerous industrial and research areas. In some cases generic FCMs suffer from a certain drawback that originates from their definition and concerns a limited, first-order dynamics of processing realized at the nodes of the maps. In this study, we introduce a concept of higher-order memory based FCMs. The proposed extension modifies the simulation model of a generic FCM while it does not negatively impact transparency and simplicity of the model itself. We discuss several architectural alternatives along with the ensuing computing and optimization aspects. Preliminary experimental results included in this paper show superiority of the extended higher-order memory based FCMs over a generic FCM in terms of the modeling accuracy.**

I. INTRODUCTION

Generally speaking, methods for modeling of dynamic systems can be divided into two categories [1]. The first category concerns *quantitative* methods, which can be applied both to well-understood systems, e.g., mathematical programming techniques of operations research, and to less well-defined systems, e.g., statistically-based methods of data mining. However, quantitative methods suffer from substantial drawbacks. First, significant effort and specialized knowledge outside the domain of interest is required to apply these techniques. Second, some dynamic systems are nonlinear, which may make quantitative approaches difficult to use. Finally, numerical data are often hard to collect or uncertain. The second category concerns *qualitative* methods, which are free from these limitations.

Fuzzy Cognitive Maps (FCMs) introduced by Kosko in 1986 are a convenient tool for qualitative modeling [2] [3]. Their main advantages include very simple and comprehensive graph representation, which results in an intuitive to understand model. In addition, FCMs are very flexible in terms of system design and applications since they have comprehensible structure and operation, are adaptable to a given domain, and are capable of abstract representation and fuzzy reasoning. The areas of applications of FCMs are very extensive and cover engineering, medicine, political sciences, earth and environmental sciences, economics and management, etc. Examples of specific applications include diagnosis of diseases [4], analysis of electrical circuits [5], failure modes effects analysis [6], fault management in distributed network environment [7], modeling of software development project [8][9], and many others. The scope and span of applications demonstrate usefulness of this method and justify further research in this area.

Being a powerful approach to modeling of dynamic systems, Fuzzy Cognitive Maps exhibit a substantial weakness. Given the definition of FCMs, their dynamics is of the first-order, meaning that the next state depends upon the one in the previous iteration. We note that this property limits ability of FCMs to model complex systems, especially for those that cannot be accurately described by models in which current state is calculated based on the previous state only. In order to enhance modelling capabilities of FCMs, this paper proposes an extension that introduces higher-order FCMs. We also introduce real-coded genetic algorithm based learning method for the newly proposed extended FCMs.

This paper is organized as follows. Section II presents Fuzzy Cognitive Maps including methods for their development. In Section III we provide motivation and summary of the related work and introduce the proposed extension. Section IV presents preliminary experimental results including comparative study between the proposed extended FCMs and generic FCMs. Section V summarizes this paper and outlines future research directions.

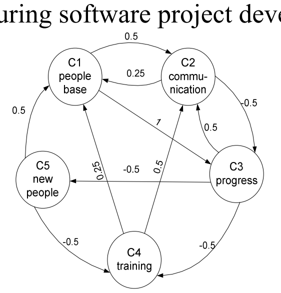
II. FUZZY COGNITIVE MAPS

A. General Overview

FCMs model a given dynamic system using concepts and cause-effects relationships, which link concepts and describe how they affect each other. Generally speaking, three types of relationships can be distinguished: positive, negative, or neutral. They are usually quantified using a floating-point value from -1 to 1 that expresses given relationship strength. Positive values reflect promoting (positive) effect, whereas negative describe inhibiting (negative) effect. The value of –1 represents full inhibiting, +1 full promoting and 0 denotes neutral causal effect. All other values in this range correspond to different, intermediate levels of causal effect. FCMs have a very simple and intuitive digraph representation, which includes nodes connected by directed edges. The graph's nodes represent concepts (events, actions, values, goals, etc.) relevant to a given domain and the causal relations between

* This work is partially supported by the Alberta Ingenuity Fund and Natural Sciences and the Engineering Research Council of Canada (NSERC).

them are depicted by directed edges. Each edge is associated with a weight value that reflects strength of corresponding relation. The graph can be equivalently expressed by a square matrix, called *connection matrix*, which stores all weights for the edges between corresponding concepts represented by corresponding rows and columns. Figure 1 shows an example of FCM that describes relationships that affect pace of work during software project development [8].



	C1	C2	C3	C4	C5
C1	0.5	0	1	0	0
C2	0.25	0	-0.5	0	0
C3	0	0.5	0	-0.5	-0.5
C4	0.25	0.5	0	0	0
C5	0.5	0	0	0.5	0

Fig. 1 An example of FCM (graph with corresponding connection matrix)

There are two mainstream techniques for creation of FCMs [10]. The first group denoted as *manual methods* includes techniques that exploit only human knowledge. In this case, an expert(s) designs and implements adequate model manually, using pencil and paper, based on his or her understanding of the modeled domain. This approach suffers from several disadvantages. To name a few, models created this way are subjective and biased by experts' insights into the system; it is also difficult to evaluate such models. However, for a long time this was the only way to develop FCMs, mainly because of lack of approaches for automated or semi-automated FCMs' development. Recently, several attempts have been made to provide support to this process based on available historical data. Methods from this group, denoted as *computational methods*, are aimed at learning FCM connection matrix, i.e. casual relationships (edges), and their strength (weights) based on historical data. In this case, the expert knowledge is substituted by a set of historical data and a computational procedure to optimize the connection matrix based on the data is applied. A number of algorithms for learning FCMs have been recently proposed. Two main learning paradigms are used: Hebbian learning and evolutionary algorithms.

Once FCM is developed, qualitative analysis of a given system may be carried out by simulating the model. Simulation consists of computing state of the system, which is described by a state vector, over a number of successive iterations. The state vector specifies current values of all concepts (nodes) at a particular iteration. Initial state vector describes the system's state at the beginning of simulation (at the *zero* iteration), and must be defined before simulation starts. Successive state vectors are determined based on the system's state at preceding iteration. Value of each node is calculated from the values of nodes, which exert influence on the given node through cause-effect relationships (nodes that are connected to the given node). The equation (1) shows the formula, which is used to perform simulations of the system dynamics.

$$C_j(t+1) = f\left(\sum_{i=1}^{N} e_{ij} C_i(t)\right) \quad (1)$$

where N is the number of concepts in a given system
$C_j(t)$ is the value of a given concept C_j at the iteration t
e_{ij} is the strength of mutual relationship that concept C_i exerts on concept C_j
f is a transformation function

The transformation function is used to reduce unbounded weighted sum to a certain range, which is usually set to [0, 1]. Several different transformation functions, which can be divided into discrete (e.g. taking only values 0 or 1) and continuous, have been used. The most commonly used function is sigmoid logistic function, which is described as follows.

$$f(x) = \frac{1}{1+e^{-Cx}} \quad (2)$$

where C is the parameter that determines the shape of the function

In most practical applications of FCMs the parameter C in formula (2) was assumed to be 5, which provides a reasonable balance between a unit step function and a linear function. The normalization hinders quantitative analysis, yet it allows for comparisons between nodes. Each node can be defined as active (value of 1), inactive (value of 0), or active to a certain degree (value between 0 and 1). Thus, the concepts values represent the degree of their "existence" at a particular iteration. Different scenarios can be considered by simulating FCMs with different initial conditions represented by initial state vector.

Simulations of FCMs lead the system to one of the three main groups of outcomes. The first group covers all the scenarios in which simulation heads to a fixed state vector value, which is called *hidden pattern* or *fixed-point attractor*. Alternatively, system may keep cycling between some fixed state vector values, which is known as a *limit cycle*. Finally, so called *chaotic attractor* may appear. This term refers to a simulation, in which the FCM continues to produce different state vector values for successive cycles. Figure 2 [11] illustrates sample simulation results for FCM with six concepts with applying logistic transformation function.

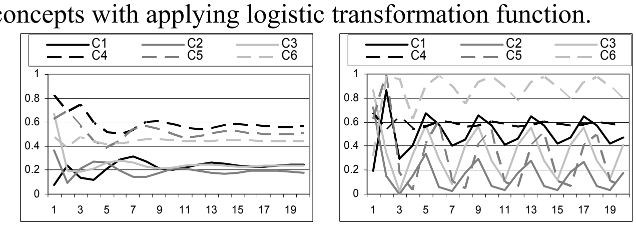


Fig. 2 Sample FCM simulation results

The left hand side Figure shows simulation that results in the fixed point attractor, whereas the right hand side Figure presents the limit cycle.

B. RCGA Learning Method for FCMs

A state-of-the-art approach to learning FCMs was introduced by Stach et al. in 2005 [11]. It applies real-coded genetic algorithm (RCGA) to develop FCM from a set of historical data. The core of this method is a learning module, which exploits RCGA to find FCM structure that is capable to mimic a given historical data. Figure 3 shows a high-level diagram of this method.

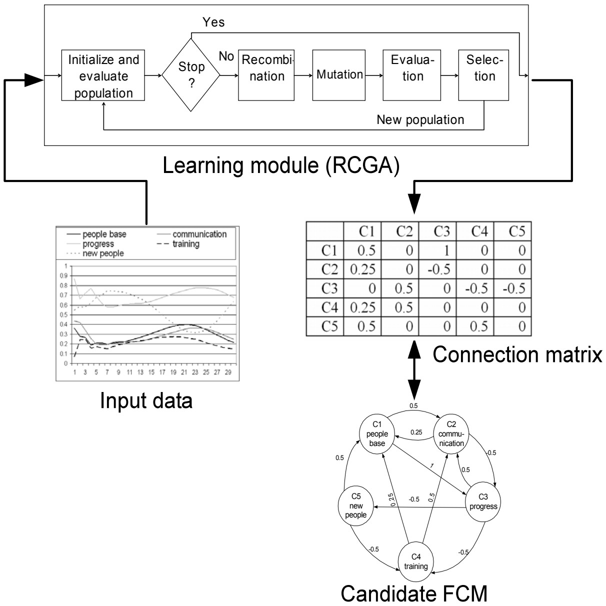


Fig. 3 High-level diagram of the RCGA learning method

The RCGA learning method is fully automated. Based on historical data it develops the FCM (called *candidate FCM*), which describes a given system. The detailed description of this method can be found in [11] [12]. The core element of this approach is a real-coded genetic algorithm (RCGA) [13]. Essentially, this floating-point extension to a generic genetic algorithm represents chromosome using a floating point vector. The vector's length corresponds to a number of variables being optimized, which makes this approach suitable for problems with continuous search space.

III. Higher-order Fuzzy Cognitive Maps

A. Motivation

The expression (1) that governs the dynamics of FCM captures only the first-order dynamics. In other words, a state of FCM at a particular iteration depends only on its state in a preceding iteration, as expressed by the following formula

$$C_j(t+1) = g\left(C_1(t), C_2(t), \ldots, C_j(t), \ldots, C_n(t), E\right) \quad (3)$$

where g stands for "is a function of"
E is the nxn connection matrix that stores all the relationships e_{ij} between concepts

This imposes considerable restrictions on the range of dynamic systems that can be modeled using this tool. More precisely, an accurate generic FCM model cannot be obtained for systems, in which their current state depends not only on the immediate preceding state, but is also influenced by the past states. Although several generalizations of FCMs, which are summarized in the next subsection, were proposed in the literature, none of them directly addresses this issue.

B. Related Work

Several other research groups introduced extensions and modifications to the generic FCMs. They are summarized in Table I.

TABLE I
Summary of Related Work

Extension	Reference	Memory	Learning
E-FCM	[14]	Applies fixed delay time between certain concepts	Only manual
FTCM	[15]	Applies fixed delay time between certain concepts	Only manual
RFCM	[16]	Depends only on immediate preceding state	Manual and automated via Hebbian learning
DRFCM	[17]	Depends only on immediate preceding state	Manual and automated via Hebbian learning
RBFCM	[18]	Depends only on immediate preceding state	Only manual
CNFCM	[19]	Depends only on immediate preceding state	Only manual
DCN	[20]	Depends only on immediate preceding state	Only manual
KM	[21]	Depends only on immediate preceding state	Only manual
Our method	This paper	Considers more than one iteration	Manual and automated via RCGA genetic learning

Extended FCMs proposed in 1992 [14] introduced three relationship types: nonlinear, conditional, and time-delayed. The next extension, Fuzzy Time Cognitive Maps, introduced time dependencies on relationships between particular concepts [15]. In Random FCMs [16] [17], the model was defined based on a random neural network (RNN), which incorporated probabilities for concepts activation. In 2000 Rule Based FCMs [18] were proposed. They combined FCMs with a rule based fuzzy architecture in order to enrich the representation of relationships. Certainty Neuron FCM applied a special type of transfer function to enrich generic FCM [19]. In Dynamic Cognitive Networks [20], weights between nodes were represented as functions of the corresponding source node values. Finally, most recent extension called Knowledge Maps [21] proposed four types of causal relations: simple cause-effect relations, time-delay causal relations, conditional probabilistic causal relations, and sequential relations, which are special kinds of time-delay causal relations. In short, we note that the previously proposed extensions targeted causal relationships representation and introduced the concept of time-delayed relationships between selected concepts. Those enhancements, however, have targeted fundamentally different shortcomings of FCMs when compared with extension proposed in this paper. This work presents a novel research direction, which is aimed at introducing higher-order memory. Moreover, most of the proposed extensions have not been supported by any learning algorithm. This means that their development was entirely based on expert(s) beliefs and manual development. The manual development of an extended FCM is more complicated and potentially susceptible to a human bias, since it requires more parameters to be established when compared to generic FCMs. In contrast, our extension is supported by an automated learning approach to establish corresponding models from data. Experimental evaluation of the quality of the proposed extension is also presented in this paper. Preliminary experimental results for development and simulation of the proposed extended FCM are performed and compared with the results generated by a generic FCM.

C. Proposed Approach

This paper introduces a higher-order memory extension into the generic FCM model, which is described as

$$C_j(t+1) = g\begin{pmatrix} C_1(t),...,C_n(t), C_1(t-1),...,C_n(t-1), \\ ..., C_1(t-K),...,C_n(t-K), E \end{pmatrix} \quad (4)$$

where K is the parameter that determines the order of dynamics of FCMs

For generic FCMs the parameter K equals 0. This general formula can be equivalently represented in the form

$$C_j(t+1) = f\left(\sum_{\substack{k=0 \\ k \le t}}^{K_j} \left(g_j(k) \sum_{i=1}^{n} e_{ij} C_i(t-k) \right) \right) \quad (5)$$

where K_j is the number of historical values that are taken into account to calculate current value of a given concept C_j
$g_j(k)$ is a coefficient that determines how much preceding values influence the current one for a given concept C_i

Parameter K determines the number of historical values that are used to calculate new value of a given concept, while function g determines strength of the effect of the historical values on the current value. If we assume $K_j=0$ and $g_j(0)=1$ for each concept C_j, then the extended formula reduces to the generic FCM, which is described by (1).

The experiments reported in this paper are performed with a few simplifications in (5). In particular, both K_j and $g_j(k)$ are assumed to be the same for each concept. With those assumptions, learning of the extended FCM requires establishing K additional variables during the learning process. Given the number of concepts and transformation function, the total number of variables equals $N*N + K$, in which the first part correspond to all relationships strength, and the second one determines set of coefficients that define g.

The RCGA-based learning optimizes those parameters such that the developed model fits the input data. The general architecture of our learning method is the same as shown in Figure 3. The difference lies in the chromosome representation, which consists of $N*N+K$ floating point values, limited to the range [-1, 1] for the first $N*N$ values, and to [0, 1] for the last K values. RCGA parameters are given in section IV.B.

IV. Experiments

A comparative study between FCMs with various orders of memory was performed. The objective was to determine the differences in accuracy between modeling using the generic and the higher-order FCMs.

A. Data Sets

The dataset used in experiments concern climate and weather observations for Canada in 2005 and have been acquired from National Climate Archive [22]. Four large cities have been selected for our experiments, namely Edmonton, Montréal, Toronto, and Vancouver. Five attributes, i.e. temperature, dew point, relative humidity, wind speed, and pressure, have been chosen to describe the weather patterns. The data pre-processing stage included linear normalization of each attribute to the range [0,1] for each location. This was done with respect to maximum and minimum value of a given attribute at a given location throughout the whole year. Next, four subsets from the data were taken for each location. Each subset concerned two weeks interval in different season (January 8-21 – *winter*, April 15-28 – *spring*, July 8-21 – *summer*, and October 8-21 – *fall*). Finally, we picked up observations recorded every four hours, which resulted in 84 data points per location per interval.

B. Experimental Setup

Firstly, several initial experiments were performed to tune up the RCGA algorithm. Comparing to the basic parameters reported in [11] for RCGA learning of generic FCMs, we modified the fitness function to obtain faster and better convergence in case of learning the extended FCMs. The fitness function is expressed by (6)

$$fit = a \sum_{t=1}^{T-1} \sum_{n=1}^{N} \left[\left(C_n(t) - \hat{C}_n(t) \right)^2 + \left(\Delta C_n(t) - \Delta \hat{C}_n(t) \right)^2 \right] \quad (6)$$

where $C_n(t)$ is the value of a node n at iteration t in the input data
$\hat{C}_n(t)$ – is the value of a node n at iteration t from simulation of the model
$\Delta C_n(t) = C_n(t) - C_n(t-1)$
$\Delta \hat{C}_n(t) = \hat{C}_n(t) - \hat{C}_n(t-1)$
T – is the input data length
N – is the number of concepts
a – normalization coefficient equals to $[10 \cdot (T-1) \cdot N]^{-1}$

Other RCGA parameters, which were established experimentally, include *recombination method* (randomly chosen from simple and flat crossover), *mutation method* (randomly chosen from random mutation, non-uniform mutation, and Mühlenbein's mutation), *selection method* (randomly chosen from roulette wheel and tournament), *probability of recombination* equal to 0.4, *probability of mutation* equal to 0.4, *population size* that equals 100 chromosomes, *max generation* size that equals 25000, and *max fitness* value that equals 0.999.

Two main groups of experiments have been carried out, i.e. in-sample and out-of-sample. The former one concerned modeling task, in which the goal was to find an accurate model of the input data, which consisted of first 42 observations in each case, see Figure 4 to see a sample input data.

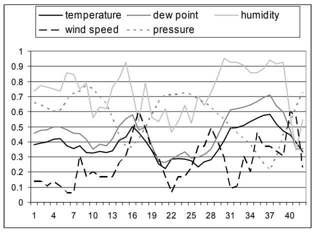


Fig. 4 Sample input data

The evaluation criterion is defined in the form

$$error = \frac{1}{(T-1)\cdot N}\sum_{t=1}^{T-1}\sum_{n=1}^{N}\left|C_n(t)-\hat{C}_n(t)\right| \qquad (7)$$

The latter group of experiments involved adding the test subset (remaining 42 observations in each case). The first subset, *training*, was used to establish a model, whereas the *test* one was used for evaluation of the model accuracy on unseen data – again, criterion (7) was used.

Both of the above mentioned groups of experiments have been performed for FCMs with different order of memory K to analyze how this parameter influences modeling accuracy.

C. Results

Close to a hundred of experiments have been performed to evaluate the proposed extension. Table II summarizes the experimental results. The reported values of each criterion are calculated taking averages across all four locations. Also, average and standard deviations are given for each experimental setup. The reported results can be compared to a baseline accuracy, which is around 43% and has been calculated running one experiment for each case (96 experiments in total). In this case, however, random FCMs have been generated and the out-of-sample test was performed. The results also report time required to learn the FCM model using RCGA algorithm to study the impact of the application of the higher-order memory on the training time.

TABLE II

SUMMARY OF EXPERIMENTAL RESULTS

Season	Criterion	K=0	K=1	K=2	K=3	K=4	K=5
winter	In-sample	0.108	0.108	0.105	0.104	0.095	0.088
	Out-of-sample	0.234	0.242	0.204	0.231	0.215	0.202
	Execution time [s]	1101	1319	1517	1709	1882	2078
spring	In-sample	0.086	0.082	0.079	0.076	0.080	0.074
	Out-of-sample	0.147	0.171	0.149	0.140	0.156	0.139
	Execution time [s]	1117	1321	1510	1714	1907	2096
summer	In-sample	0.070	0.067	0.071	0.070	0.068	0.067
	Out-of-sample	0.106	0.096	0.098	0.090	0.091	0.091
	Execution time [s]	1108	1321	1516	1718	1890	2078
fall	In-sample	0.059	0.055	0.053	0.053	0.054	0.056
	Out-of-sample	0.132	0.126	0.119	0.121	0.113	0.112
	Execution time [s]	1117	1321	1521	1708	1903	2255
avg	In-sample	0.081 ±0.005	0.078 ±0.005	0.077 ±0.004	0.076 ±0.003	0.074 ±0.003	0.071 ±0.001
	Out-of-sample	0.155 ±0.027	0.159 ±0.029	0.142 ±0.019	0.145 ±0.024	0.144 ±0.026	0.136 ±0.025
	Execution time [s]	1111	1321	1516	1712	1895	2127
Baseline		0.413 ±0.076	0.403 ±0.093	0.426 ±0.101	0.415 ±0.100	0.461 ±0.097	0.474 ±0.103

As can be seen from the table, in-sample error decreases with increasing the memory order – 0.081 for K=0 vs. 0.071 for K=5 on average. The relative difference is approximately 12.3%. The same is observed in case of the out-of-sample error, which decreases from 0.155 to 0.136 – approximately 12.2% reduction of the error. At the same time the execution time increases, which is a result of increasing the number of parameters that have to be established during the learning process. Almost double increase of training time is observed when comparing execution time between K=0 and K=5.

Table III presents statistical significance analysis of the difference between the results of the higher order FCMs and the generic FCM. A *paired t-test* values have been calculated based on errors obtained for all 16 experiments performed with a given order of memory for each location and season.

TABLE III

STATISTICAL RESULTS COMPARISON THROUGH PAIR T-TEST S

Test	Models	t-value	Test	Models	t-value
In-sample	K=0 vs. K=1	1.81	Out-of-sample	K=0 vs. K=1	-0.55
	K=0 vs. K=2	1.99		K=0 vs. K=2	1.48
	K=0 vs. K=3	1.86		K=0 vs. K=3	1.32
	K=0 vs. K=4	2.19		K=0 vs. K=4	1.67
	K=0 vs. K=5	2.78		K=0 vs. K=5	2.91

Critical t-value at 95% confidence equals 2.160 and thus the application of the higher order model provides statistically significantly more accurate model for K = 5 when compared with a generic FCM.

Figure 5 illustrates the relation between in-sample (left hand side) and out-of-sample (right hand side) errors depending on location for K = 0 through 5.

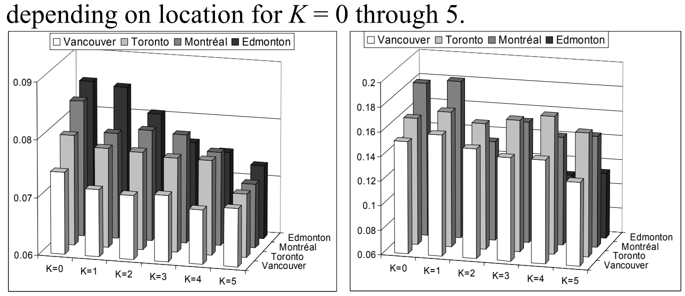


Fig. 5 Experimental results with respect to location

This view provides results that are similar to those shown in Table II, i.e. the error decreases with increasing of memory order. Considering the location dimension, in-sample errors is the smallest for Vancouver (0.071 on average versus 0.079 for Edmonton). The out-of-sample errors are the smallest for Edmonton, whereas the order of other cities in general is similar to the order for the in-sample tests. The results suggest that weather in the cities from the West coast may be more predictable than weather in the other considered cities.

Figure 6 shows example in-sample simulation result that corresponds to learning from input data given in Figure 4 for K=0 (plot on the left hand side), and K=5 (plot on the right hand side). Plots drawn as lines without markers correspond to simulations results, whereas plots with the circular markers give the input data.

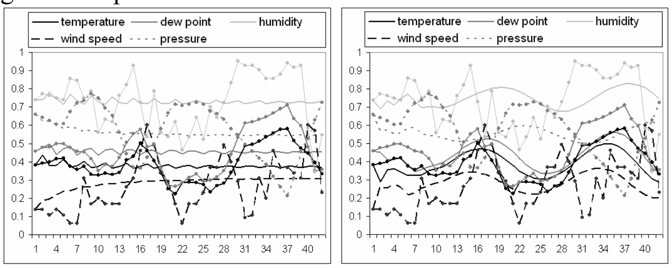


Fig. 6 Sample experimental results

The simulation results reveal that the generic FCMs are biased towards fixed point attractor or limit cycle behavior. This observation is consistent with an inherent property of generic FCMs, in which the current state depends only on the state at the previous iteration. This implies one of the above mentioned behaviors after the simulation reaches a state that already has been reached in one of the previous iterations.

V. Conclusions and Future Work

In this paper, a generalized higher-order memory Fuzzy Cognitive Maps are proposed. While the introduced extension retains the key advantages of the generic FCMs, including their transparency, it helps capture higher-order dynamics of the processes to be modeled. Additionally, a fully automated method for development of the proposed extended FCM based on a genetic algorithm was proposed. Some preliminary experimental results demonstrate the superiority of the generalized version of the FCM over the generic FCM. More specifically, the experiments indicate that higher-order FCMs has led to the improved accuracy of both in-sample and out-of-sample tests when compared with the performance of the generic FCM. The differences are shown to be statistically significant at 95% significance level. One has to note, though, that the improved accuracy came at a cost of longer training. This stems from the fact that the higher the order of FCMs is the more parameters need to be established.

This paper elaborates on introducing dynamics to FCMs by means of higher memory-order at a global level, i.e. involving the entire map, which is described by equation (4). As an alternative research direction we plan to explore local dynamics described as

$$C_j(t+1) = g\begin{pmatrix} C_1(t),...,C_j(t),C_j(t-1),C_j(t-2), \\ ...,C_j(t-K),C_{j+1}(t),...,C_n(t),E \end{pmatrix} \quad (8)$$

This approach considers introducing higher-order memory for a certain single concept, which would result in reducing the training time when compared with the proposed extension. On the other hand, this modification may results in a smaller improvement in accuracy.

References

[1] R.J.G.B. Campello, and W.C. Amaral, "Towards true linguistic modelling through optimal numerical solution," *Int. J. of Systems Science*, vol. 34, no. 2, 2003, pp. 139-157.

[2] B. Kosko, "Fuzzy cognitive maps," *Int. J. of Man-Machine Studies*, vol. 24, 1986, pp. 65-75.

[3] J. Aguilar, "A survey about fuzzy cognitive maps papers (Invited Paper)," *Int. J. of Computational Cognition*, vol. 3, no. 2, 2005, pp. 27-33

[4] P.R. Innocent, and R.I. John, "Computer aided fuzzy medical diagnosis," *Inf. Sciences*, vol. 162, no. 2, 2004, pp. 81-104.

[5] M.A. Styblinski, and B.D. Meyer, "Signal flow graphs vs. fuzzy cognitive maps in application to qualitative circuit analysis," *Int. J. of Man-Machine Studies*, vol. 35, 1991, pp. 175-186.

[6] C.E. Pelaez, and J.B. Bowles, "Applying fuzzy cognitive maps knowledge representation to failure modes effects analysis", *Proc. of the IEEE Annual Symposium on Reliability and Maintainability*, 1995, pp. 450-456.

[7] T.D. Ndousse, and T. Okuda, "Computational intelligence for distributed fault management in networks using fuzzy cognitive maps," *Proc. of IEEE Int. Conf on Communications Converging Techn. for Tomorrow's Application*, vol. 3, 1996, pp. 1558-1562.

[8] W. Stach, and L. Kurgan, "Modeling software development project using fuzzy cognitive maps", *Proc. of the 4th ASERC Workshop on Quantitative and Soft Software Engineering*, 2004, pp. 55-60.

[9] W. Stach, L. Kurgan, W. Pedrycz, and M. Reformat, "Parallel fuzzy cognitive maps as a tool for modeling software development project," *Proc. of the North American Fuzzy Information Processing Society Conference*, 2004, pp. 28-33.

[10] W. Stach, L.A. Kurgan, and W. Pedrycz, "A survey of fuzzy cognitive map learning methods," In: *Issues in Soft Computing: Theory and Applications,* P. Grzegorzewski, M. Krawczak, and S. Zadrozny, Eds., Exit, 2005, pp. 71-84.

[11] W. Stach, L. Kurgan, W. Pedrycz, and M. Reformat, "Genetic learning of fuzzy cognitive maps," *Fuzzy Sets and Systems*, vol. 153, no. 3, 2005, pp. 371-401.

[12] W. Stach, L. Kurgan, W. Pedrycz, and M. Reformat, "Learning fuzzy cognitive maps with required precision using genetic algorithm approach," *Electronics Letters*, vol. 40, no. 24, 2004, pp. 1519-1520.

[13] F. Herrera, M. Lozano, and J.L. Verdegay, "Tackling real-coded genetic algorithms: operators and tools for behavioural analysis," *Artificial Intelligence Review*, vol. 12, no. 4, 1998, pp. 265-319

[14] M. Hagiwara, "Extended fuzzy cognitive maps," *Proc. of the IEEE Int. Conf. on Fuzzy Systems*, 1992, pp. 795-801

[15] K.S. Park, and S.H. Kim, "Fuzzy cognitive maps considering time relationships," *Int. J. of Human Computer Studies*, vol. 42, no. 2, 1995, pp. 157-168

[16] J. Aguilar, "A fuzzy cognitive map based on the random neural model," *Lecture Notes in Artificial Intelligence*, vol. 2070, 2001, pp. 333-338

[17] J. Aguilar, "A dynamic fuzzy-cognitive-map approach based on random neural networks," *Int. J. of Computational Cognition*, vol. 1, no. 4, 2003, pp. 91-107

[18] J.P. Carvalho, and J.A.B. Tome, "Rule based fuzzy cognitive maps - qualitative systems dynamics," *Proc. of the North American Fuzzy Information Processing Society*, 2000, pp. 407-411

[19] A.K. Tsadiras, and K.G. Margaritis, "Cognitive mapping and certainty neuron fuzzy cognitive maps," *Inf. Sciences*, vol. 101, no. 1-2, 1997, pp. 109-130

[20] Y. Miao, Z.-Q. Liu, C.K. Siew, and C.Y. Miao, "Dynamical cognitive network – an extension of fuzzy cognitive map," *IEEE Trans. on Fuzzy Systems*, vol. 9, no. 5, 2001, pp. 760-770

[21] H. Zhuge, and X.-F. Luo, "Knowledge map: mathematical model and dynamic behaviors," *J. of Computer Science and Technology*, vol. 20, no. 3, 2005, pp.289-295

[22] National Climate Data and Information Archive, available at http://climate.weatheroffice.ec.gc.ca/