2018 European Control Conference (ECC)
June 12-15, 2018. Limassol, Cyprus

System Identification and Indirect Inverse Control Using Fuzzy Cognitive Networks with Functional Weights

Georgios Karatzinis, Yiannis S. Boutalis, *Member, IEEE* and Theodoros L. Kottas

***Abstract*— A Fuzzy Cognitive Network (FCN) is an operational extension of a Fuzzy Cognitive Map (FCM) which assumes, first, that it always converges to equilibrium points during its operation and second, it is in continuous interaction with the system it describes and may be used to control it. In this paper we show that the conditions that guarantee the convergence of the FCN may lead to a special, yet very powerful, form of the network that assumes functional interconnection weights with excellent system approximation abilities. Assuming that the plant is unknown it is initially approximated by a FCN and a procedure for adaptive estimation of its functional weights is proposed that guarantee approximation error convergence to zero. The FCN is then used for the Indirect adaptive Inverse Control of a plant. The methodology is tested on a coupled two-tank system.**

I. Introduction

Fuzzy Cognitive Maps (FCMs) are well-known inference networks capable and essential for various tasks such as modeling, control, pattern recognition applications, decision making, analysis and prediction in many different scientific areas [1-5]. They have been introduced by Kosko [6], based on Axelrod's work on cognitive maps [7], in order to model complex behavioral systems using causal relations. The graphical representation of the FCM is basically a cyclic directed graph structure for representing causal reasoning. The graphical form consists of nodes and weight interconnections. The nodes represent behavioral aspects of the system, that is, each one represents a system characteristic feature. Weight interconnections represent node interactions, that is, causal relationships between actions, goals, events, values and trends of the system.

An FCM integrates the accumulated experience and knowledge on the system operation by using human experts who know the operation of the system and its behavior in different conditions. The direction and intensity of causal relations involve the quantification of a fuzzy linguistic variable which is assigned by experts during the modeling phase [8]. Kosko enhanced the power of cognitive maps considering fuzzy values for their nodes and fuzzy degrees of interrelationships between nodes [1],[4]. He also proposed the differential Hebian rule [9] to estimate the FCM weights expressing the fuzzy interrelationships between nodes based on acquired data. Other remarkable learning algorithms are presented in [10-13].

Fuzzy Cognitive Networks (FCNs) [14] were initially introduced as a general computational and storage framework to facilitate the use of FCM in a strong interaction with the physical system they describe. The updating mechanism receives feedback from the real system, stores the acquired knowledge and imposes control values to it. FCNs and their storage mechanism assume that they reach equilibrium points, each one being associated with a specific operation condition of the underlying system. The concept of "steady" nodes which correspond to input values and influence but are not influenced by the other nodes of the FCM was also introduced in that work.

In designing a FCN, the conditions of existence and uniqueness of equilibrium points and convergence issues constitute the most crucial role. According to Kosko [8],[15], starting from an initial state, simple FCMs follow a path, which ends in a fixed point or limit cycle, while more complex ones may end in an aperiodic or "chaotic" attractor. These fixed points and attractors could represent meta rules of the form "If input then attractor or fixed point". For a FCN equipped with sigmoid functions, it has been proved that when the weight interconnections fulfill certain conditions, related to the size of the graph, the concept values will converge to a unique solution regardless their initial values [16], [17]. Adaptive parameter estimation algorithms for each unique solution are also given in [17]. Based on a bilinear modeling, an alternative bilinear adaptive estimation algorithm is developed in [18] in order to estimate the weights and sigmoid parameters associated with a new equilibrium point. Rigorous proofs for its stability and convergence properties were given, incorporating convergence issues in parameter projection schemes. The aforementioned conditions and proofs complete the proper operation of the FCM-based models, allowing them to converge and operate in cooperation with the system they describe making them suitable for many applications. Also, they limit experts contribution only to the initial graph by just defining the number of nodes and the causal relationships irrespectively of their initial values. In the above schemes the associations between equilibrium points and network weights and concept values are stored in a fuzzy rule database to be recalled later on by using a fuzzy inference system [19].

In this paper, we study a different form of storage procedure of the FCN acquired data knowledge, instead of using Fuzzy Rule Databases. Alternatively, we could use fuzzy basis functions [20] which combine both numerical data and linguistic information in a uniform fashion. However, in FCNs it is not simple to provide effective methods to assign accurately the linguistic terms, determine

Georgios Karatzinis is with the Department of Electrical and Computer Engineering, Democritus University of Thrace, Xanthi, Greece, 67100 GR (e-mail: gkaratzi@ee.duth.gr).

Yiannis S. Boutalis is with the Department of Electrical and Computer Engineering, Democritus University of Thrace, Xanthi, Greece, 67100 GR (e-mail: ybout@ee.duth.gr).

Theodoros L. Kottas is with the Department of Electrical Engineering, Technological Educational Institute of West Macedonia, Kozani, 50100 GR, Greece (email: tkottas@teiwm.gr).

the membership functions and implication method. These depend on experts' knowledge, to every unique problem which is under consideration, as well as the interpretation of the inference result. Our purpose is to lead the need of experts' judgment and knowledge for the issues related to the fuzzy database and fuzzy inference mechanism to zero using polynomial terms of pure data obtained from the operation of FCN. During the repetitive updating operation, the procedure uses input from the system variables, producing a new weight matrix for each new equilibrium state. We assume functional interconnection weights for keeping these weights for probable future use and may update their functional design. To the best of our knowledge this is the first time that such a functional representation of the FCM weights is introduced.

Similar to the use of a fuzzy rule database and fuzzy inference, the proposed approach allows weight retrieval even in situations, where the equilibrium conditions had not been exactly met during the training phase. This is achieved by handling pure data from the FCN producing functional deterministic relationships between concepts and weight interconnection values using regression analysis. Weights represent the dependent variables and node values the independent ones, consisting of multiple polynomials with proper order based on the statistics measures of R squared and Adjusted R squared. That form embraces relationships as a core aspect in a functional way using normalized data from the FCN rather than from fuzzy meta rules form. This way the need for effective and efficient selection of the rules and the determination of the appropriate parameters of the membership functions and properties of the principal elements in the fuzzy system (fuzzifier, fuzzy rule base, fuzzy inference engine and defuzzifier) is completely avoided.

The material of the paper is unfolded by giving first, in section II, the basics of FCN, its convergence and its weight updating. Next, in section III, the FCN with functional weights, which is the core idea of the paper is introduced, presenting the way by which the proper functions are estimated. Section IV proposes the use of such an FCN in a system identification inverse control scheme, which is further elaborated on a two-tank benchmark experiment in section V, presenting simulation results that demonstrate the performance of the scheme. Conclusions and thoughts for future works are given in section VI.

II. Fuzzy Cognitive Networks

FCN is an operational extension of FCM and a modeling methodology for complex systems, which originated from combination of fuzzy logic and neural networks. The graphical illustration of an FCM is a signed fuzzy graph with feedback, consisting of nodes and weighted interconnections. The nodes of the graph are related to concepts that are used to describe main behavioral characteristics of the system. Nodes are connected by signed and weighted arcs representing the causal relationships that exist among concepts. Graphical representation illustrates which concept influences other concepts, thus showing the interconnections between them. This simple illustration permits thoughts and suggestions in reconstructing FCM, such as the adding or deleting of an interconnection or a concept. In conclusion, an FCM is a fuzzy graph structure that allows systematic causal propagation, in particular, forward and backward chaining.

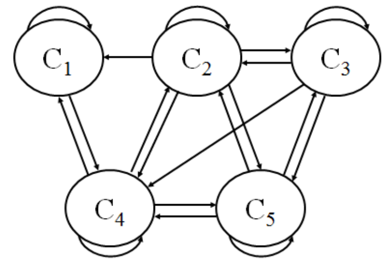

Figure 1. FCN with 5 nodes.

A. Fuzzy Cognitive Networks Representation

A graphical representation of FCN is depicted in Fig. 1. Each concept represents a characteristic of the system, in general, it represents events, actions, goals, values, and trends of the system. Each concept is characterized by a number A_i that represents its value, and it results from the transformation of the real value of the systems variable, which is represented by this concept, either in the interval [0,1] or in the interval [-1,1]. All concept values from vector A are expressed as:

$$\mathrm{A} = \begin{bmatrix} \mathrm{A}_1 & \mathrm{A}_2 & \dots & \mathrm{A}_n \end{bmatrix}^T$$

with n being the number of the nodes. Causality between concepts allows degrees of causality and not the usual binary logic, so the weights of the interconnections may range in the interval [-1,1]. The existing knowledge on the behavior of the system is stored in the structure of nodes and interconnections of the map. The value of w_{ij} indicates how strongly concept C_j influences concept C_i. The sign of w_{ij} indicates whether the relationship between concepts C_j and C_i is direct or inverse. The nodes that influence but are not influenced by other nodes are called steady. For the FCN of Fig. 1 weight interconnection matrix W is equal to

$$W = \begin{bmatrix} d_{11} & w_{12} & 0 & w_{14} & 0 \\ 0 & d_{22} & w_{23} & w_{24} & w_{25} \\ 0 & w_{32} & d_{33} & 0 & w_{35} \\ w_{41} & w_{42} & w_{43} & d_{44} & w_{45} \\ 0 & w_{52} & w_{53} & w_{54} & d_{55} \end{bmatrix}$$

The equation that calculates the values of concepts of FCNs is equal to:

$$A_i(k) = f\left(d_{ii} A_i(k-1) + \sum_{\substack{j=1 \\ j \neq i}}^{n} w_{ij} A_j(k-1) \right) \tag{1}$$

where $A_i(k)$ is the value of concept C_i at discrete time k, $A_i(k-1)$ the value of concept C_i at discrete time $k-1$ and $A_j(k-1)$ is the value of concept C_j at discrete time $k-1$. w_{ij} is the weight of the interconnection from concept C_j to concept C_i and d_{ii} is a variable that takes on values in the interval [0,1] depending upon the existence of "strong" or "weak" self-feedback to node i. Most common function f is the squashing sigmoid function with saturation level 0 and 1. Sigmoid function f is a continuous and differentiable node function used both in FCMs and FCNs and squashes the result in the interval [0,1] and is expressed as

$$f = \frac{1}{1 + e^{-c_l x}}$$

Where $c_l > 0$ is used to adjust its inclination.

B. Online Parameter Estimation of Fuzzy Cognitive Networks

Equation (1) is applied repetitively computing new node values. In FCNs, proofs and conditions for the existence and uniqueness of solutions for (1) have been studied analytically in [16],[17]. Following the inverse procedure it would be helpful to find appropriate weight sets directly related to desired equilibrium points of the FCN. Choosing a desired state A^{des} for the FCN this is equivalent to solving the equilibrium equation

$$\mathrm{A}^{des} = f\left(W^* \mathrm{A}^{des}\right) \tag{2}$$

in respect to W^*, where $\mathrm{A}^{des} = \begin{bmatrix} \mathrm{A}_1^{des} & \mathrm{A}_2^{des} & \dots & \mathrm{A}_n^{des} \end{bmatrix}^T$ and f is a vector valued function $f : \Re^n \to \Re$, defined as follows: $f(\mathrm{x}) = [f_1(\mathrm{x}_1) \quad f_2(\mathrm{x}_2) \quad \dots \quad f_n(\mathrm{x}_n)]^T$ where $\mathrm{x} \in \Re^n$ and $f_i\left(x_i\right) = \dfrac{1}{1+e^{-c_{li}x_i}}$, for $i = 1,2,\dots,\mathrm{n}$. Then

$$f^{-1}(\mathrm{A}^{des}) = \mathrm{W}^* \mathrm{A}^{des} \tag{3}$$

Where $f^{-1}(\mathrm{A}^{des}) = \begin{bmatrix} f_1^{-1}(\mathrm{A}_1^{des}) & f_2^{-1}(\mathrm{A}_2^{des}) & \dots & f_n^{-1}(\mathrm{A}_n^{des}) \end{bmatrix}^T$

$$f_i^{-1}(\mathrm{A}_i^{des}) = \mathrm{c}_{l_i}^* \, w_i^* \cdot \mathrm{A}^{des} \tag{4}$$

with w_i^* being the i^{th} row of W^* and c_{li} the scaling factor of function f corresponding to concept A_i. The form f_i^{-1} is straightforwardly computed if one tries to solve the sigmoid function in respect to its argument

$$f_i^{-1}(\mathrm{A}_i^{des}) = \ln\left(\frac{\mathrm{A}_i^{des}}{1-\mathrm{A}_i^{des}}\right) \tag{5}$$

Taking into account the limits

$$\lim_{x\to 0} f^{-1}\left(x\right) = -\infty, \quad \lim_{x\to 1} f^{-1}\left(x\right) = \infty$$

it is obvious that the operation interval is formulated in the interval (a,b) , where 0<a<b<1. In [9,10] two alternative approaches have been studied for solving (4). In the first one, (4) appears as a linear parametric model assuming that the sigmoid inclination parameters c_{li} for all nodes are equal to one. In the second case both inclination parameters and weight interconnections have to be estimated and (4) is clearly bilinear parametric model. In both modeling approaches, solution of (4) is carried out recursively and the updating algorithms use appropriate projection methods which guarantee that the conditions of existence and uniqueness are always fulfilled. The linear approach shows inability to give always the appropriate parameter estimates. The bilinear approach takes into account both weight and inclination parameters of the nodes' sigmoid functions outperforming the linear.

In order to display the performance differences of these alternative approaches, we consider the FCN of Fig. 1, and compare both approaches when trying to "capture" the following desired equilibrium node values

$$\mathrm{A}^{des} = \begin{bmatrix} 0.2 & 0.35 & 0.58 & 0.68 & 0.92 \end{bmatrix}^T$$

In both approaches the initial node values are the same random ones and they include node self-feedback. The linear model fulfilling the existence and uniqueness conditions converges to

$$\mathrm{A}_{li} = \begin{bmatrix} 0.2228 & 0.3502 & 0.58 & 0.6764 & 0.9197 \end{bmatrix}^T$$

The bilinear model fulfilling the existence and uniqueness conditions converges to

$$\mathrm{A}_{bi} = \begin{bmatrix} 0.2 & 0.35 & 0.58 & 0.68 & 0.92 \end{bmatrix}^T$$

It can be observed that the bilinear approach outperforms the linear one allowing more accurate convergence. The inclusion of the inclination parameters estimates facilitate the convergence to desired state.

III. Fuzzy Cognitive Networks With Functional Weights Representation

The obtained weight interconnections related to desired equilibrium points of the FCN and node values constitute the knowledge of the encountered operational situations. Instead of storing that knowledge in a fuzzy rule database, weights are formed in functional structure. Considering a polynomial

$$a_n x^n + a_{n-1}x^{n-1} + \dots + a_2 x^2 + a_1 x + a_0$$

it is non-linear in x but it is linear in the parameters a. In order to model the values of the weight interconnections, that is the dependent variables in terms of the values of the independent (node) variables, we have

$$\begin{aligned} W_1 &= y_1 = f\left(x_1,\dots,x_n\right) \\ W_2 &= y_2 = f\left(x_1,\dots,x_n\right) \\ &\vdots \\ W_k &= y_k = f\left(x_1,\dots,x_n\right) \end{aligned}$$

where x_i, (i=1,2,…,n) are the node variables that influence the dependent variables (weights W_i,i=1,2,…,k) and are used to fit weight interconnection data into a least squares linear regression model. It is conducive to use a polynomial generator to construct different n^{th} order polynomials. The repeatable increase of the order of the polynomials will change the best fit shape, that is the functional weights shape. The coefficient of determination "R squared" r^2 will lead to proper order that best fits the obtained data. The interconnection weights are constructed as high order functions. In order to estimate the values of parameters, linear least squares is needed. Linear least squares problems occur when solving overdetermined linear systems.

In our application the total of the pairs of node and weight interconnection values naturally give more equations than unknowns leading to overdetermined system. The solution of the linear least squares problem is a vector $x \in \mathbf{R}^n$ for which the norm of the residual $\boldsymbol{r}$ is minimized

$$\left\|\boldsymbol{r}\right\|_2 = \left\|\boldsymbol{b} - M\boldsymbol{x}\right\|_2 \to \min$$

where $\boldsymbol{b} \in \mathbf{R}^m$ and the full matrix $M \in \mathbf{R}^{m\times n}$ is overdetermined, that is $m > n$

$$M = \begin{bmatrix} \mathrm{A}_1^1 & \cdots & \mathrm{A}_n^1 & W_1^1 & \cdots & W_k^1 \\ & \vdots & & & \vdots & \\ \mathrm{A}_1^m & \cdots & \mathrm{A}_n^m & W_1^m & \cdots & W_k^m \end{bmatrix} \tag{6}$$

Where m is the number of equilibriums inserted as observations for least squares. Obtaining the normal equation $M^T M\boldsymbol{x} = M^T\boldsymbol{b}$ the least squares solution is uniquely determined by

$$x_{LS} = M^+\boldsymbol{b} = \left(M^T M\right)^{-1} M^T \boldsymbol{b}$$

with M^+ is the pseudo-inverse of M. Instead of normal equations we use QR decomposition because is numerically more stable and does not change the condition number [21].

Considering a matrix $M \in \mathbf{R}^{m\times n}$ with $m \geq n$ and rank equal to n there exists the Cholesky decomposition of $M^T M = R^T R$

where R is an upper triangular matrix. Since R is non singular $(MR^{-1})^T(MR^{-1})=I$ and matrix $Q=MR^{-1}$ has orthogonal columns. The QR decomposition is

$$M = QR = Q\begin{bmatrix} R_1 \\ 0 \end{bmatrix}$$

where R_1 is an $n\times n$ upper triangular matrix. Then the pseudo-inverse of $M\in\mathbf{R}^{m\times n}$, $m>n$ and rank equal to the minimum of m and n can be found by $M^+=R^{-1}Q^T$. Then the unique solution is given by

$$x_{LS} = R^{-1}Q^T\boldsymbol{b}$$

The method that is used for computing the QR decomposition is in our case the Householder transformations [22].

IV. Configuration of the Proposed Control Scheme

This section presents the proposed control configuration scheme combining the stored knowledge from the estimator, the construction of functional weights and the adaptive estimation of them. The final purpose is to use the functional structure of weight interconnections obtained from the FCN for the Indirect Inverse Control of the plant. Fig. 2a) shows the operation of the FCN in close cooperation with the real system it describes collecting off-line training data from previous operating conditions into FCN vectors. The following procedure is the same with that described in section II using updating algorithms and projection methods which guarantee that the conditions of existence and uniqueness are fulfilled. Both linear and bilinear approaches can be used. Fig 2b) shows that based on the stored acquired knowledge, polynomial generator produces the polynomials which are used for regression analysis. The use of statistics measure r^2 is significant to find the polynomials with proper order that best fit the weight interconnections data. The coefficients of the polynomials are estimated through QR decomposition. As new desired points of equilibriums are introduced the FCN computes the weight matrix to converge to them. The new set of data consisting of node and weight values is added to knowledge storage to update the form of functional weights. The M^{ith} denotes the new row added in (6) during the operation. However, the node that describes the control action is kept out in both Fig. 2a) and Fig. 2b) because it will be estimated using an inverse procedure. In order to retrieve weights the M^{ith}_{des} is used. This denotes a vector in which only the desired normalized nodes are loaded to the corresponding positions of the A^{sys} vector and the node that describes the control action is kept out again. This is the desired state for weight retrieval during the online adaptive procedure. The FCN block computes the converging state A^{eq} using the retrieved weights and a random initial state A^{init} including the control signal node. When using the bilinear approach the inclination parameters are also retrieved. The inverse control procedure uses the fact that the FCN sigmoid function in (2) is invertible and the FCN structure permits to derive the control node value from the desired states. The constructed functional weights adapt and change shape online, increasing the control performance by minimizing the error between desired and actual output.

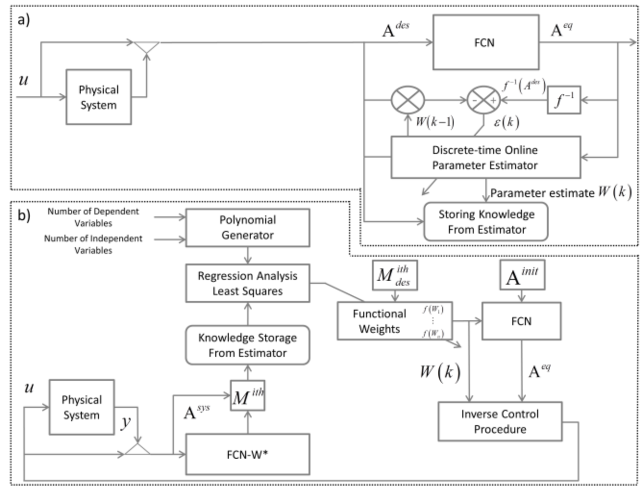

Figure 2. a) Interactive operation of the FCN for off-line training and b)Indirect Inverse Control using FCN with online adaptation of its functional weights.

V. Simulation Study

We apply the proposed approach in two case studies for the two coupled tanks system in two different tasks with and without online adaptive estimation of functional weights. This is a part of an experimental multiple tanks setup of the Laboratory of *Automatic Control Systems & Robotics*, DUTH.

A. Plant Description-Modeling

The nonlinear model of the coupled two tanks system presented in Fig. 3 is given by:

$$\frac{dh_1(t)}{dt} = -\frac{a_1}{\mathrm{A}}\sqrt{2gh_1(t)} + \eta u(t),$$
$$\frac{dh_2(t)}{dt} = \frac{a_1}{\mathrm{A}}\sqrt{2gh_1(t)} - \frac{a_2}{\mathrm{A}}\sqrt{2gh_2(t)}$$

Where $h_i(t)$ with (i=1,2) are the water level in the tanks, a_i are the outlet area of the tanks, A is the cross-sectional area of the tanks, g is gravitational constant, u is the voltage applied to the pump and η is the constant relating the control voltage with the water flow from the pump. The system after the linearization around the equilibrium point given by h_i^0 and u_0, yields

$$\Delta\dot{h}(t) = A\Delta h(t) + \mathrm{B}\Delta u(t) \quad (7)$$

where $\Delta h(t) = \left[h_1(t)-h_1^0 \quad h_2(t)-h_2^0\right]^T$, $\Delta u(t) = \left[u(t)-u_0 \quad 0\right]^T$

The plant parameters and the related reference levels are given in Table I. From (7) we get:

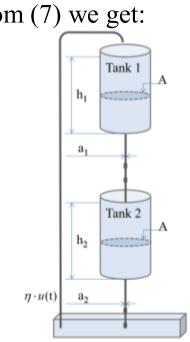

Figure 3. Two coupled tanks configuration.

$$\begin{bmatrix} \Delta\dot{h}_1(t) \\ \Delta\dot{h}_2(t) \end{bmatrix} = \begin{bmatrix} -\left(\frac{a_1}{A}\right)^2 \frac{g}{\eta u_0} & 0 \\ \left(\frac{a_1}{A}\right)^2 \frac{g}{\eta u_0} & -\left(\frac{a_2}{A}\right)^2 \frac{g}{\eta u_0} \end{bmatrix} \begin{bmatrix} \Delta h_1(t) \\ \Delta h_2(t) \end{bmatrix} + \begin{bmatrix} \eta \\ 0 \end{bmatrix} \Delta u(t)$$

B. FCN Design and Inverse Control

The FCN designed for the two coupled tanks system is shown in Fig. 4. The description of each node is given in Table II. The weight matrix of the FCN is depicted below:

$$W = \begin{bmatrix} w_{11} & 0 & 0 & 0 & 0 \\ 0 & w_{22} & 0 & 0 & 0 \\ 0 & 0 & w_{33} & 0 & 0 \\ w_{41} & w_{42} & 0 & w_{44} & 0 \\ 0 & w_{52} & w_{53} & 0 & w_{55} \end{bmatrix}$$

In the simulation example, the objective consists in tracking two different reference scenarios based on off-line training or alternatively both off-line and online adaptation for the indirect inverse control of tank 2.

The off-line training creates the FCN's knowledge storage based on tanks dynamics calculating the next states of water level in both tanks. The desired process is to control the water level in tank 2 through control signal applied in tank 1.

Simplifying the computations and the volume of storage data, we assume that the data inserted as observations for linear least squares is of the form:

$$M = \begin{bmatrix} \mathrm{A}^1(h_2) & \mathrm{A}^1(h_2(t+\Delta\mathrm{T})) & w_{11}^1 & \cdots & w_{55}^1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathrm{A}^m(h_2) & \mathrm{A}^m(h_2(t+\Delta\mathrm{T})) & w_{11}^m & \cdots & w_{55}^m \end{bmatrix}$$

Thus, M uses normalized water level of h_2, its next state and all weight interconnections on every observed row. After the off-line training M is an m×11 matrix for linear and m×12 matrix for bilinear, where m is the number of observations. Compared to Fig. 2b), in that case, the vector for weights retrieval through functional weights becomes:

$$M_{des}^{ith} = \left[\mathrm{A}\left(h_2^{cur}\right) \quad \mathrm{A}^{des}(h_2) \right]$$

TABLE I. PARAMETERS OF THE PLANT OBTAINED FROM 33-041 COUPLED TANKS SYSTEM OF FEEDBACK INSTRUMENTS

Variable	Description	Rated Value
h_i	Water level of tank i	0-25 cm
u	Voltage level of pump	0-5 V
h_i^0	Reference level of tank i	16 cm
u_0	Reference level of pump	2.91 V
A	Cross-sectional area	0.01389 m^2
α_i	Outlet area of tank i	5.265e-5 m^2
η	Constant relating voltage and flow	2.2e-3 m/V s

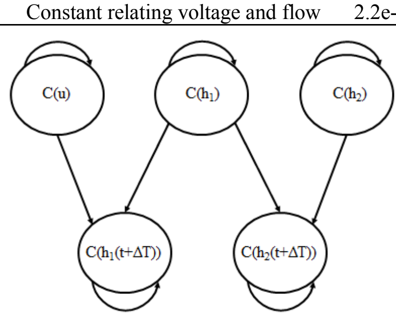

Figure 4. FCN for the two coupled tanks system

TABLE II. CONCEPT NODES FOR THE FCN OF TWO COUPLED TANKS

Variable	Description
C(u)	Voltage applied to pump
$C(h_1)$	Water level of tank 1
$C(h_2)$	Water level of tank 2
$C(h_1(t+\Delta T))$	Water level of tank 1 at time t+ΔT
$C(h_2(t+\Delta T))$	Water level of tank 2 at time t+ΔT

Where $\mathrm{A}(h_2^{cur})$ is the current measurement of water level of tank 2 and $\mathrm{A}^{des}(h_2)$ is the desired one at every iteration. Then, the FCN block computes the converging state A^{eq} using the retrieved weights and a random initial state A^{init} including the control signal node. In that application the control signal is applied only on tank 1 and is given for linear by:

$$\mathrm{A}(u) = \left(f^{-1}\left(\mathrm{A}^{des}\left(h_1(t+\Delta\mathrm{T})\right)\right) - w_{42}\mathrm{A}(h_1) - w_{44}\left(\mathrm{A}\left(h_1(t+\Delta\mathrm{T})\right)\right)\right) / w_{41}$$

and for bilinear approach by:

$$\mathrm{A}(u) = \left(f^{-1}\left(\mathrm{A}^{des}\left(h_1(t+\Delta\mathrm{T})\right)\right) - w_{42}\mathrm{A}(h_1)cl_4 - w_{44}\left(\mathrm{A}\left(h_1(t+\Delta\mathrm{T})\right)\right)cl_4\right) / (w_{41}cl_4)$$

C. Simulation results

The proposed approach is applied on the two coupled tank system and its performance is shown in Figs. 5-8. Figs. 5,6 show the tracking of tank 2 in two different case studies using only the off-line training procedure. In the first case the desired state follows a sinusoidal trajectory, while in the second case the desired trajectory switches between different set points. The fluctuation of water level at tank 1, which is the first that is affected by the water inflow (control) is also displayed. The same scenarios are carried out using both off-line training and online adaptation of functional weights, improving the tracking error. As can be seen from Fig. 7 and Fig. 8 the steady state error is minimized obtaining values very close to zero. In both approaches (off-line / off-line + on-line adaptation) the bilinear approach outperforms linear. This is expected because in small FCNs, when the system states reach marginal values in the node interval [0,1], due to the convergence conditions, the linear model cannot always keep good track of the desired system values. This does not happen with the bilinear model due to both weight and inclination estimation [18].

VI. CONCLUSION

In this paper the FCNs with functional weights are introduced. In this form their approximation abilities are enhanced avoiding the need for using fine-tuned fuzzy rule database and fuzzy inference procedure in order to achieve the same purpose. The proposed methodology is tested on a system approximation and indirect inverse control paradigm showing off excellent performance. The contribution of the proposed approach which adopts functional weighted relationships is summarized in: **(a)** lowering experts' knowledge and judgment, **(b)** providing non-linear weight functions, **(c)** the network may handle more than one relationship between nodes, **(d)** the network is more flexible and its structure makes easier its adaptation on small on-site control systems. Future work will include exploratory development of alternative but reliable recursive least squares procedure to update polynomial coefficients

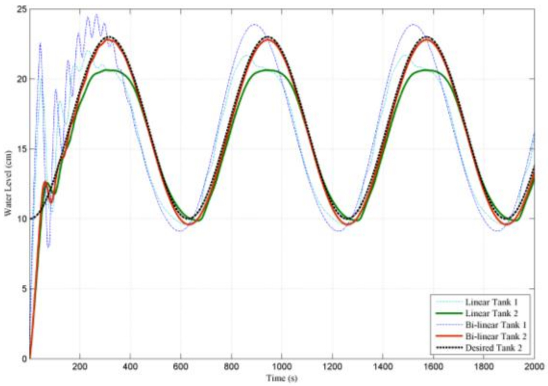

Figure 5. Water level of the two tanks using off-line training of the FCN with functional weights for Indirect Inverse Control of tank 2 (case 1).

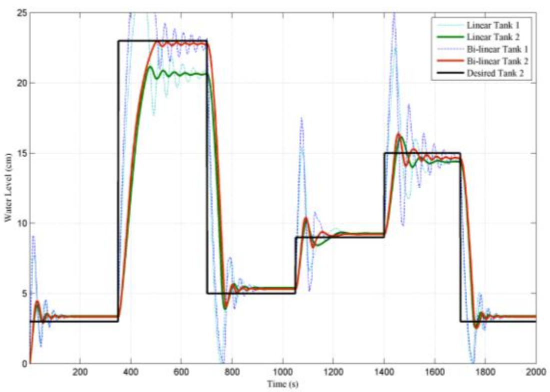

Figure 6. Water level of the two tanks using off-line training of the FCN with functional weights for Indirect Inverse Control of tank 2 (case 2).

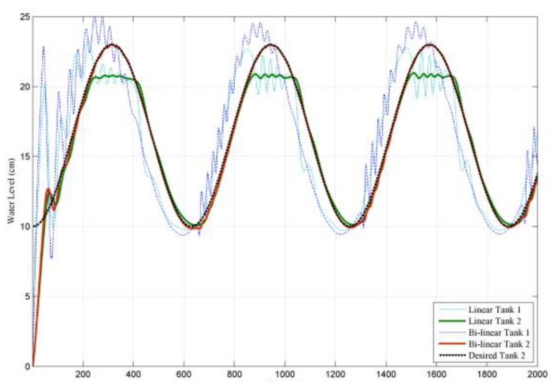

Figure 7. Water level of the two tanks using off-line and adaptive estimation of functional weights for Indirect Inverse Control of tank 2 (case 1).

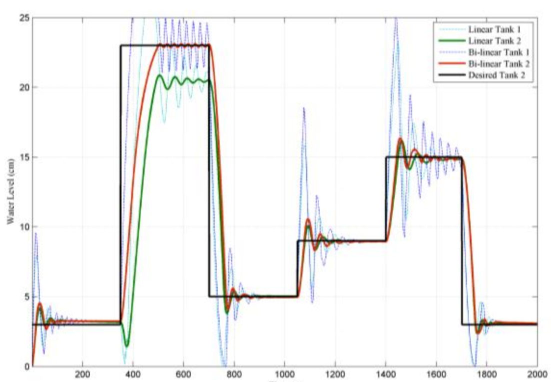

Figure 8.Water level of the two tanks using offline and adaptive estimation of functional weights for Indirect Inverse Control of tank 2 (case 2).

applying at the same time data reduction procedures. Moreover, functional weight updating procedures that will guarantee the stability of the closed loop system remain unaddressed and they have to be derived.

REFERENCES

[1] D.M. Case, C.D. Stylios, “Fuzzy Cognitive Map to model project management problems,” Annual Conference of the North American Fuzzy Information Processing Society,IEEE,Texas,USA, pp. 1-6,2016

[2] Papageorgiou, E. I. and J.L. Salmeron,”A Review of Fuzzy Cognitive Map Research during the last decade,” *IEEE Trans. Fuzzy Systems*, vol. 21,no. 1, pp. 66-79, Feb. 2013.

[3] W. Pedrycz, A. Jastrzebska, W. Homenda, “Design of fuzzy cognitive maps for modeling time series,” *IEEE Trans. Fuzzy Syst*., 24 , pp. 120-130, 2015.

[4] J.L. Salmeron, E.I. Papageorgiou, “Fuzzy grey cognitive maps and nonlinear hebbian learning in process control,” *Appl. Intell*., 41 (1), pp. 223-234, 2014.

[5] J.L. Salmeron, W. Froelich,”Dynamic optimization of fuzzy cognitive maps for time series forecasting,” *Knowl. Based Systems*, vol. 105, pp. 29-37, 2016.

[6] B. Kosko, “Fuzzy cognitive maps,” *Int. J. Man-Mach. Stud*., vol. 24, no. 1, pp. 65-75, 1986.

[7] R. Axelrod, *Structure of Decision: The Cognitive Maps of Political Elites*, Princeton University Press, New Jersey, 1976.

[8] B. Kosko , *Fuzzy Engineering*, Prentice-Hall Inc, New York, 1997 .

[9] B. Kosko, “Differential Hebbian Learning,” in *AIP Conf. Proceedings*, vol. 151, pp. 277–282, 1986.

[10] E. Papageorgiou, C. D. Stylios, and P. P. Groumpos, “Active Hebbian learning algorithm to train fuzzy cognitive maps,” *Int. J. Approx. Reason*., vol. 37, no. 3, pp. 219–249, 2004.

[11] W. Stach, L. A. Kurgan, W. Pedrycz, and M. Reformat, “Genetic learning of fuzzy cognitive maps,” *Fuzzy Sets Syst*., vol. 153, no. 3, pp. 371–401, 2005.

[12] 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., 2005.

[13] D. E. Koulouriotis, I. E. Diakoulakis, and D. M. Emiris, “Learning fuzzy cognitive maps using evolution strategies: a novel schema for modeling and simulating high-level behavior,” in Proc. IEEE Congr. Evol. Comput., pp. 364–371, 2001.

[14] T.L. Kottas, Y.S. Boutalis, M.A. Christodoulou, “Fuzzy cognitive networks: a general framework,” *Inteligent Desicion Technologies,* vol. 1, pp. 183–196, 2007.

[15] J. Dickerson, B. Kosko, “Virtual worlds as Fuzzy Cognitive Maps,” *Presence*, vol.3, no. 2, pp.173–189, 1994.

[16] Y. Boutalis, T. Kottas, M. Christodoulou, “On the existence and uniqueness of solutions for the concept values in fuzzy cognitive maps,” in *Proc. 47th IEEE Conf. Decis. Control*, Cancun, Mexico, Dec. 9–11, pp. 98–104, 2008.

[17] Y. Boutalis, T. Kottas, M. Christodoulou, “Adaptive estimation of fuzzy cognitive maps with proven stability and parameter convergence,” *IEEE Trans. Fuzzy Syst.,* vol.17, no. 4, pp. 874–889, 2009.

[18] T.L. Kottas , Y.S. Boutalis , M.A. Christodoulou , “Bi-linear adaptive estimation of fuzzy cognitive networks,” *Appl. Soft Comput*., vol. 12, pp. 3736–3756, 2012 .

[19] Y. Boutalis, D. Theodoridis, T. Kottas, and M. A. Christodoulou, *System Identification and Adaptive Control: Theory and Applications of the Neurofuzzy and Fuzzy Cognitive Network Models* (Advances in Industrial Control). Basel, Switzerland: Springer, 2014.

[20] L.-X. Wang and J. M. Mendel, “Fuzzy basis function, universal approximation, and orthogonal least squares learning,” *IEEE Trans. Neural Networks*, vol. 3, no. 5, pp. 807-814, Sept. 1992.

[21] G. Golub, “Numerical methods for solving linear least squares problems,” *Numerische Mathematik*, vol. 7, pp. 206-216,1965.

[22] A. S. Householder, “Unitary Triangularization of a Nonsymmetric Matrix,” *Journal of the ACM*, vol 5, pp. 339-342, 1958.