International Journal of Neural Systems, Vol. 23, No. 6 (2013) 1350029 (14 pages)

DOI: 10.1142/S0129065713500299

EVOLVING RBF NEURAL NETWORKS FOR ADAPTIVE SOFT-SENSOR DESIGN

ALEX ALEXANDRIDIS
Department of Electronics
Technological Educational Institute of Athens
Agiou Spiridonos Aigaleo 12210, Greece
alexx@teiath.gr

Accepted 20 August 2013
Published Online 10 October 2013

This work presents an adaptive framework for building soft-sensors based on radial basis function (RBF) neural network models. The adaptive fuzzy means algorithm is utilized in order to evolve an RBF network, which approximates the unknown system based on input–output data from it. The methodology gradually builds the RBF network model, based on two separate levels of adaptation: On the first level, the structure of the hidden layer is modified by adding or deleting RBF centers, while on the second level, the synaptic weights are adjusted with the recursive least squares with exponential forgetting algorithm. The proposed approach is tested on two different systems, namely a simulated nonlinear DC Motor and a real industrial reactor. The results show that the produced soft-sensors can be successfully applied to model the two nonlinear systems. A comparison with two different adaptive modeling techniques, namely a dynamic evolving neural-fuzzy inference system (DENFIS) and neural networks trained with online backpropagation, highlights the advantages of the proposed methodology.

Keywords: Adaptive modeling; fuzzy means; online learning; radial basis function; soft-sensors.

1. Introduction

Soft-sensors are virtual sensors, with the ability to estimate the values of physical quantities, without actually measuring them. Soft-sensors usually employ input variables that are associated with the physical quantity to be estimated, but can be measured with less effort and/or cost. Based on the approach used for producing the correlation, soft-sensors can be categorized as model-driven or data-driven.[1] Model-driven soft-sensors rely on first principle models in order to describe the mechanisms that correlate the measured variables with the variable to be estimated. The development of such models can be rather cumbersome and time consuming, as quite often the physicochemical phenomena that lie in the background are poorly understood. Furthermore, such models are process-specific, so their applicability is rather limited. Data-driven sensors on the other hand, do not require explicit knowledge of first-principle equations describing the system, but are based on historical data derived from it. The recorded data are usually correlated using computational techniques including computational intelligence tools, multivariate statistical methods, etc.

Among computational intelligence tools, neural networks (NNs) have become very popular for the development of soft-sensors, due to their ability to identify and learn highly complex and nonlinear relationships by employing a black-box approach, i.e. without requiring any *a priori* knowledge about the process, but based on input–output data only. A soft-sensor based on feedforward NNs has been employed for real-time prediction of the moisture and fat content in olive pomace.[2] Fortuna *et al.*[3] present a number of strategies for improving the generalization capabilities of NN-based soft-sensors when small data sets are available. An NN-based predictive tool for estimating toxicity levels to *Vibrio fischeri*[4] has also been proposed. Singh *et al.*[5] present an artificial NN-based soft estimator of transducer static nonlinearity. An artificial nose was developed,[6] based on hybrid neural systems. The use of NNs in building soft-sensors has been thoroughly exploited in medicine, mainly as diagnostic tools,[7–10] and in civil engineering as tools for damage detection and integrity monitoring in buildings and infrastructures.[11–20] Other applications include failure and fault detection,[21–23] traffic modeling,[24–27] remote sensing,[28–31] and face recognition.[32] Fuzzy logic[33] and principal component analysis[34] have also been used for building data-driven soft-sensors.

Though such tools have been proven able to provide accurate predictions in static environments, they fail to take into account data retrieved from new experiments. However, an important concern in modeling dynamical systems is that often their dynamics vary with time. Therefore, a static data-driven model may not be adequate for providing accurate predictions, especially throughout a long time horizon. To this end, a number of adaptive modeling techniques have been proposed in the literature, updating online the model's structure and/or parameters. An adaptive conjugate gradient learning algorithm was introduced for effective training of multilayer NNs.[35] Wang *et al.*[36] introduced an online self-organizing scheme for parsimonious and accurate fuzzy neural networks (PAFNN), and a novel structure learning algorithm incorporating a pruning strategy into novel growth criteria. An online fuzzy NN is presented by Liu and Er[37] and the proposed algorithm is tested in a wide range of applications ranging from function approximation and nonlinear system identification to chaotic time-series prediction and real-world fuel consumption prediction problems. A biologically motivated architecture for object recognition that is capable of online learning of several objects based on interaction with a human teacher has also been proposed.[38] Beyer and Cimiano[39] present a novel online semi-supervised classification approach based on growing neural gas. Competitive, temporally asymmetric Hebbian learning has been used for training an adaptive visual neuronal model.[40] A resource allocating network for function interpolation has been introduced by Platt.[41] Tolu *et al.*[42] present an automatic incremental learning model, which learns the forward model of a robot arm. An adaptive version of the k-means algorithm applied for overlapped graph clustering has been presented by Bello-Orgaz *et al.*[43] The adaptation capabilities of fuzzy-based systems have also been thoroughly exploited.[44–47]

Not surprisingly, evolving modeling techniques have been incorporated in soft-sensor schemes, leading to the development of adaptive soft-sensors. An adaptive soft-sensor which can be deployed in real-life environments is presented by Kadlec and Gabrys.[1] Bo *et al.*[48] report a successful application of an adaptive soft-sensor based on NNs to an advanced control system. An adaptive data-driven soft-sensor based on systematic key variable selection of a process system, where the key variables are captured using stepwise linear regression is derived by Ma *et al.*[49] A critical mechanical property of cement-based materials was predicted in a nondestructive manner based on an adaptive soft-sensor.[50] Frequent sequential pattern mining (FSPM) methods for discovering significant evolution patterns from satellite images have also been used.[51] Two interesting publications present the application of NN-based adaptive soft-sensors in robotics.[52,53] A review of adaptation mechanisms for data-driven soft-sensors is given by Kadlec *et al.*[54] Undoubtedly there is a large field for adaptive soft-sensor-based applications, a fact which dictates the need for the development of adaptive modeling techniques with greater accuracy and low computational load.

Radial basis function (RBF) networks[55] constitute a popular NN architecture that has recently attracted attention from many researchers[56–67] due to the inherent simplicity of its structure and the fast training algorithms it employs. The fuzzy means (FM) algorithm is a powerful RBF training technique based on fuzzy clustering,[68] that has been introduced a decade ago[69] in order to replace the k-means algorithm in the selection of the RBF hidden layer nodes. The FM algorithm presents several advantages over the typical approach, including faster computational times and automatic determination of the size of the network, and has found many successful applications for modeling[70] and control[71] of nonlinear systems. Recently a nonsymmetric variant of the algorithm with improved prediction capabilities has been introduced.[72] The aforementioned work is based on evolutionary computation techniques,[19,73–78] in order to optimize the RBF network in offline mode.

As far as online training of RBF networks is concerned though, a relatively small number of algorithms has been proposed in the literature; Fung *et al.*[79] employed a procedure that combines an online candidate regressor selection with the given QR recursive parameter estimator for adaptive supervised network training. The algorithm may need to reinitiate the clustering procedure, in case the structure of the system changes. Liu *et al.*[80] used Volterra polynomial basis functions (VPBFs) to develop a growing network technique for online structure selection. The algorithm needs an initial offline structure selection procedure and does not have the ability to eliminate basis functions when they become redundant. A sequential growing and pruning algorithm for RBF networks was proposed by Huang *et al.*[81] The algorithm uses the concept of significance of a neuron and links it to learning accuracy. The significance is calculated using a piecewise-linear approximation for the Gaussian function, which, however, is efficient only for uniformly distributed input data.

In this work, we adopt the online variant of the FM algorithm[82] for the development of evolving RBF network models. This approach presents several advantages: (a) it exploits the improved accuracy and faster computational times of the original FM algorithm, in time-varying environments, (b) the method starts with an empty hidden layer and incrementally builds the RBF network by adding and deleting nodes as new data become available, (c) its performance is not affected by the distribution of the input data and (d) it does not use an iterative procedure to update the network at each time step and does not depend on an initial random selection of parameters. The adaptive FM algorithm is incorporated into a novel framework for building adaptive soft-sensors, able to evolve their structure and parameters online, taking into account new information coming from the system under identification. The methodology is evaluated on the online identification of two different nonlinear systems, namely a simulated DC motor and a real digester reactor.

The rest of this paper is organized as follows: In the next section, a brief overview of the concept of fuzzy partition and the FM algorithm are given. Section 3 presents the adaptive RBF network training methodology, and its incorporation to the soft-sensor framework. Section 4 illustrates the adaptive soft-sensor development, and provides results from its application on two cases. The paper concludes by outlining the advantages of the proposed approach.

2. The FM Algorithm

Consider a system with N normalized input variables x_i, where $i = 1, \ldots, N$. The domain of each input variable is partitioned into an equal number c of one-dimensional triangular fuzzy sets. Each fuzzy set can be written as:

$$A_{i,j} = \{a_{i,j}, \delta\alpha\}, \quad i = 1, \ldots, N, \;\; j = 1, \ldots, c, \tag{1}$$

where $a_{i,j}$ is the center element of fuzzy set $A_{i,j}$ and $\delta\alpha$ is half of the respective width (due to the symmetric partition all the widths are equal). This partitioning technique creates a total of c^N multi-dimensional fuzzy subspaces $\mathbf{A}^l$, where $l = 1, \ldots, c^N$. Each multi-dimensional fuzzy subspace is generated by combining N one-dimensional fuzzy sets, one for each input dimension. One can define the center vector $\boldsymbol{\alpha}^l$ and the side vector $\delta\boldsymbol{\alpha}$ of each fuzzy subspace:

$$\begin{aligned}\mathbf{A}^l &= \{\boldsymbol{\alpha}^l, \delta\boldsymbol{\alpha}\}\\ &= \left\{[a^l_{1,j}, a^l_{2,j}, \ldots, a^l_{N,j}], \left[\underbrace{\delta\boldsymbol{\alpha}, \delta\boldsymbol{\alpha}, \ldots, \delta\boldsymbol{\alpha}}_{N}\right]\right\},\\ &\qquad l = 1, \ldots, c^N\end{aligned} \tag{2}$$

where a^l_{i,j_i} is the center element of the one-dimensional fuzzy set A_{i,j_i} that has been assigned to input i. Each one of the produced fuzzy subspaces is a candidate for becoming an RBF center but only a few of those will be finally selected. The selection is based on the idea of the multi-dimensional membership function $\mu_{\mathbf{A}^l}(\mathbf{x}(k))$ of an input vector $\mathbf{x}(k)$ to a fuzzy subspace $\mathbf{A}^l$, given by Nie[83]:

$$\mu_{\mathbf{A}^l}(\mathbf{x}(k)) = \begin{cases} 1 - rd^l(\mathbf{x}(k)), & \text{if } rd^l(\mathbf{x}(k)) \le 1 \\ 0 & \text{otherwise} \end{cases}, \tag{3}$$

where $rd^l(\mathbf{x}(k))$ is the Euclidean relative distance between $\mathbf{A}^l$ and the input data vector $\mathbf{x}(k)$:

$$rd^l(\mathbf{x}(k)) = \sqrt{\sum_{i=1}^{N} (a^l_{i,j_i} - x_i(k))^2} \Big/ \sqrt{N}\delta a. \quad (4)$$

Equation (4) defines a hyper-sphere on the input space with radius equal to $\sqrt{N}\delta a$. The objective of the training algorithm is to select a subset of fuzzy subspaces as RBF centers, so that all the training data are covered by at least one hypersphere. Expressing this requirement in terms of Eq. (3), the subset of fuzzy subspaces is selected so that there is at least one fuzzy subspace that assigns a nonzero multi-dimensional degree to each input training vector. The algorithm responsible for the selection of this subset is described in Ref. 69 followed by an analysis of its low computational complexity. As this is a one-pass algorithm, it has been found to handle successfully large databases, as well as datasets of reasonably high input dimensionality.[72]

3. Adaptive RBF-Based Soft-Sensor Development

3.1. *Adaptive RBF network training*

An online version of the FM algorithm has been introduced,[82] allowing the RBF network to adapt its parameters online to new data received from the system under identification. The adaptive version does not need an initial NN model. It starts with zero hidden nodes and progressively builds the model as new data become available. Obviously, the predictions of the model are not very successful in the first steps, but as soon as more data are fed into the network, the prediction ability of the network is gradually improved.

The adaptive technique which will be described briefly in this subsection, relies on the fuzzy partition of the input space, described in the previous section. The method considers all the centers of the fuzzy subspaces as candidates for locating the hidden nodes of the network. However, among all the candidate centers, the algorithm dynamically identifies and selects only the subset of fuzzy subspaces that are close to the input examples. As will be shown subsequently, this is accomplished by employing the spherical-shaped membership function of Eq. (4). At each time instant the number of selected fuzzy subspaces is equal to the number of nodes in the hidden layer and the centers of the selected subspaces coincide with the centers of the hidden nodes. Therefore, at any time point a complete RBF model is available, which is first used for predicting the future behavior of the output variables and then is updated based on the proposed algorithm.

The adaptive FM algorithm evolves the RBF network based on two levels of adaptation, namely:

(a) Adaptation of the connection weights between the hidden layer and the output layer.
(b) Adaptation of the structure of the hidden layer based on a fuzzy partition of the input space.

An overview of the algorithm is given in Fig. 1.

As far as the first level of adaptation is concerned, the connection weights $\mathbf{w}$ of the hidden layer are updated using the Recursive Least Squares (RLS) with exponential forgetting algorithm,[84] according to the following equations:

$$\begin{aligned}
\mathbf{w}(k) &= \mathbf{w}(k-1) + \mathbf{q}(k)(y(k) - \mathbf{z}^T(k)\mathbf{w}(k-1)) \\
\mathbf{q}(k) &= \mathbf{P}(k-1)\mathbf{z}(k)(\lambda + \mathbf{z}^T(k)\mathbf{P}(k-1)\mathbf{z}(k))^{-1}, \\
\mathbf{P}(k) &= (\mathbf{I} - \mathbf{q}(k)\mathbf{z}^T(k))\mathbf{P}(k-1)/\lambda
\end{aligned} \quad (5)$$

where k stands for the current sample number, y stands for the output variable, $\mathbf{z}$ are the responses of the hidden layer nodes, $\mathbf{P}$ is the inverse of the covariance matrix, and λ is the forgetting factor. Applying the forgetting factor implies that a data point that is n times old, will be weighted by λ^n.

However, due to the local approximation approach of RBF networks, this type of adaptation may not be adequate when a new data point arrives, which is not sufficiently covered by the existing centers. In order to address this situation, the algorithm introduces the second level of adaptation, where new hidden nodes are added in order to describe data points that lie outside of the area covered by existing centers. As the continuous addition of hidden nodes could lead to large network configurations and increased computational complexity, the algorithm also provides suitable means for deleting hidden nodes when they become redundant.

As soon as the first input example arrives from the system, the algorithm determines the fuzzy subspace that is closer to that data point in the relative Euclidean distance sense (Eq. (4)). The center of this

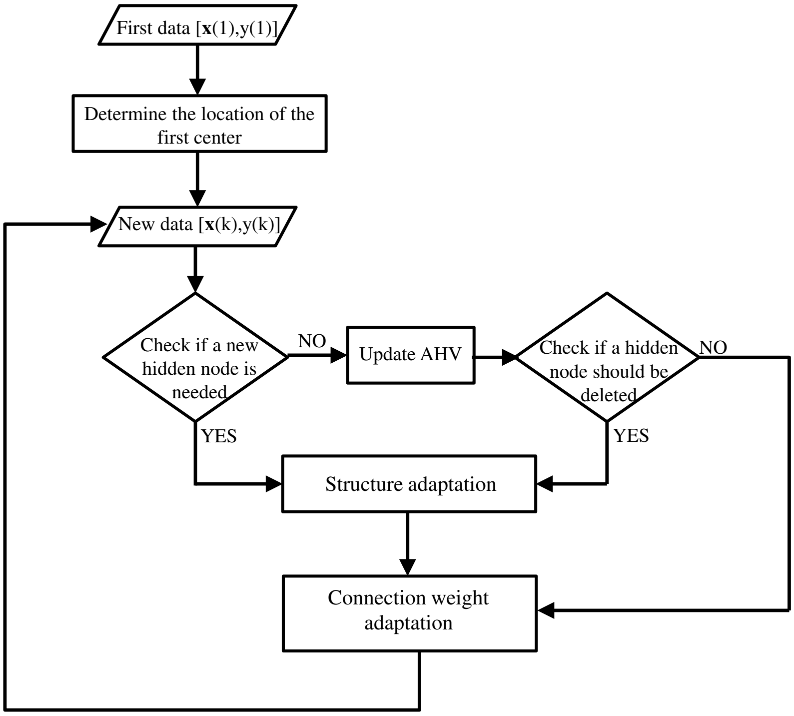

Fig. 1. Overview of the adaptive FM algorithm.

subspace becomes the center of the first hidden node and the respective fuzzy subspace initializes the list of the selected subspaces. As soon as the first hidden node is determined, the algorithm initiates two dynamic matrices, which are used to store important information. These are the Center Location Matrix (CLM) $\mathbf{G}$ and the Activation History Vector (AHV) $\mathbf{h}$. The CLM contains at each time instant the centers of the hidden layer nodes and its dimension is $L \times N$ where L is the number of the selected RBF centers and N is the dimensionality of the input space. The structure of CLM is depicted in Fig. 2. The size of the AHV is equal to the number of the selected RBF centers L and contains the last time instant that an input example was assigned to each fuzzy subspace.

When a new input example becomes available, the algorithm first checks whether the input vector can be assigned to an already selected fuzzy subspace using Eq. (4). If the answer is negative, a new node should be added to the hidden layer. This is achieved by selecting the fuzzy subspace which is closer to the input vector in the Euclidean relative distance sense and locating the center of the new hidden node, at

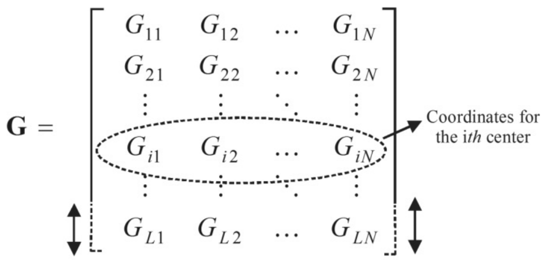

Fig. 2. Structure of the dynamically changing Center Location Matrix.

the center of the selected subspace. In this case, the new center is added to the CLM and the information in the AHV is updated.

If the algorithm decides that no new hidden node is needed, it checks whether an existing hidden node has not been assigned during the last N_d time steps to an input vector. If this is true, the hidden node is deleted and the respective fuzzy subspace is removed from the list of selected subspaces. In this way, the algorithm manages to sustain a number of hidden nodes that is sufficient enough to describe the system, but at the same time the structure of the network is kept within a reasonable size.

In case a center is added or deleted, the connection weights between the hidden layer and the output layer need to be recalculated. This is achieved by using a moving time window, where the last N_s past input–output data are stored. The connection weights are obtained by regressing the outputs of the hidden layer (formulated after the addition or deletion of a node) on the real outputs of the system. It should be noted that the regression is based only on the contents of the moving time window.

If the structure of the hidden layer remains unaltered, the method does not change the locations of the hidden node centers and updates only the connection weights between the hidden layer and the output layer using the RLS with exponential forgetting algorithm.

More details regarding the online RBF training methodology can be found in the original publication.[82]

3.2. *Incorporation of the adaptive RBF model to the soft-sensor*

In order to incorporate the adaptive training algorithm in a soft-sensor framework, an RBF network is initialized with an empty hidden layer. As soon as the first training example becomes available, the first hidden node is added and the soft-sensor is ready to provide predictions. Following the emergence of a new input data point, the algorithm makes first a prediction for the current value of the output variable without the need to measure it. At the next time step, the previous real output value is measured offline, adaptation is performed, and then the algorithm uses the updated model to predict the next value of the output variable. The soft-sensor operation is visually depicted in Fig. 3.

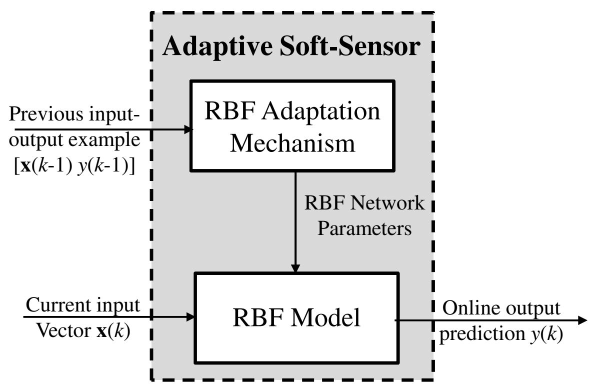

Fig. 3. Adaptive soft-sensor operational scheme.

4. Application: Design of Adaptive Soft-Sensors for Modeling Nonlinear Systems

The proposed methodology for building adaptive soft-sensors was applied for online identification of two different nonlinear systems, namely a simulated DC Motor and a real chemical reactor known as digester, commonly used in the pulp and paper industry. For comparison purposes, two additional adaptive models based on online learning state-of-the-art techniques were applied, namely a dynamic evolving neural-fuzzy inference system (DENFIS)[85] and a model based on Multi-Layer Perceptron (MLP) networks trained with an online backpropagation algorithm.[86]

Tuning of the proposed approach requires the selection of the parameters affecting the adaptation capabilities of the algorithm. To be more specific, the size N_s of the moving time window used for storing past input–output data was set to 90, the number of time steps N_d that a center is not assigned to an input example, before it is removed from the hidden layer was set to 100, and the forgetting factor λ was set to a value of 0.85. These values were common for both case studies. It was found that the results are relatively insensitive to this selection, when values close to the reported ones were applied. Significantly decreasing these values could result to instability, while the utilization of excessively larger values results to very slow adaptation. The size development of the evolving RBF network is controlled by the number of fuzzy sets used for partitioning the input dimensions. This parameter, together with the corresponding parameter for the DENFIS model, which is the distance threshold, and the number of hidden nodes for the MLP networks, were optimized separately for each case, based on a trial and error procedure.

4.1. *Case Study* I: *Online identification of a simulated DC motor*

The DC motor depicted in Fig. 4, can be described by the following nonlinear state equations, derived using fundamental electrical and mechanical

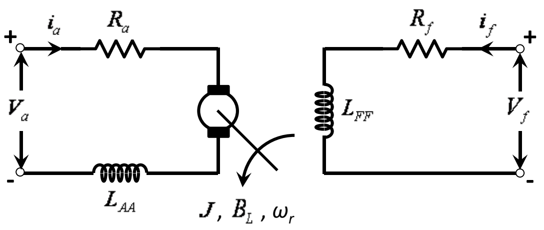

Fig. 4. A nonlinear DC motor.

Table 1. Parameters for the nonlinear DC Motor.

Parameter	Symbol	Value
Rotor speed	ω_r	State variable
Armature current	i_a	State variable
Field current	i_f	State variable
Armature voltage	V_a	Input
Field voltage	V_f	Input
Armature resistance	R_a	10.5479 Ω
Field resistance	R_f	320.6955 Ω
Armature inductance	L_{AA}	3.0948×10^{-6} H
Field inductance	L_{FF}	20.5245 H
Mutual inductance	L_{AF}	2.6116 H
Inertia	J	$0.015\ \mathrm{kg \cdot m^2}$
Coefficient of load torque	B_L	$0.0021\ \mathrm{N \cdot m \cdot s}$

laws[87]:

$$\begin{aligned}\frac{di_f}{dt} &= \frac{V_f - R_f i_f}{L_{FF}},\\ \frac{di_a}{dt} &= \frac{V_a - R_a i_a - L_{AF} i_f \omega_r}{L_{AA}}\\ \frac{d\omega_r}{dt} &= \frac{L_{AF} i_f i_a - B_L \omega_r}{J}.\end{aligned} \tag{6}$$

The notation is given in Table 1, together with values for each one of the DC motor parameters. The objective in this case is to build a soft-sensor based on the adaptive FM algorithm in order to estimate the rotor speed of the DC motor, without actually having to measure it online, but using only offline data for performing the adaptation task. The soft-sensor is based on a discrete dynamic RBF model comprising a total of 12 input variables:

$$w_r(k) = RBF\begin{pmatrix} V_a(k-1), \ldots, V_a(k-6), \\ V_f(k-1), \ldots, V_f(k-6) \end{pmatrix}. \tag{7}$$

The particular input variables to the model were chosen taking into account the sampling time, which was equal to 1 s in this case, in conjunction with the time lag needed for the DC motor to reach a steady state after a change occurs to the input.

The RBF network is initialized with an empty hidden layer. Following every discrete time step, the soft-sensor adaptation mechanism is presented with a new input–output data point. Based on the adaptive FM algorithm, the structure and the synaptic weights of the RBF network are evolved in order to describe the information contained in the new examples. Meanwhile the soft-sensor is available for providing predictions for the current value of the rotor speed.

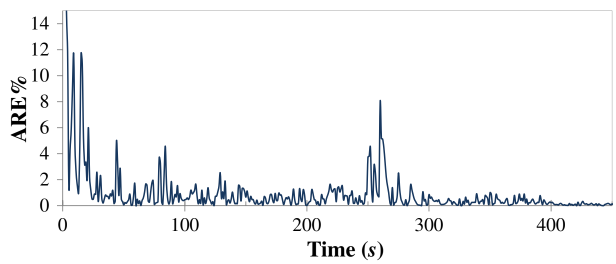

Fig. 5. Case I: Evolution of the ARE% for the RBF-based soft-sensor.

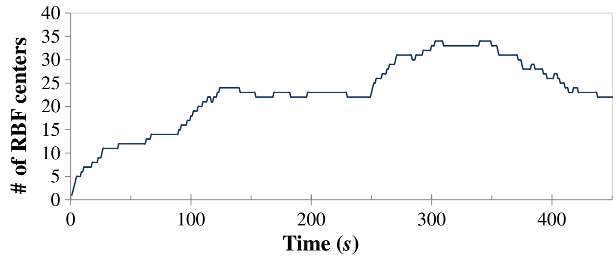

Fig. 6. Case I: Evolution of the number of RBF centers.

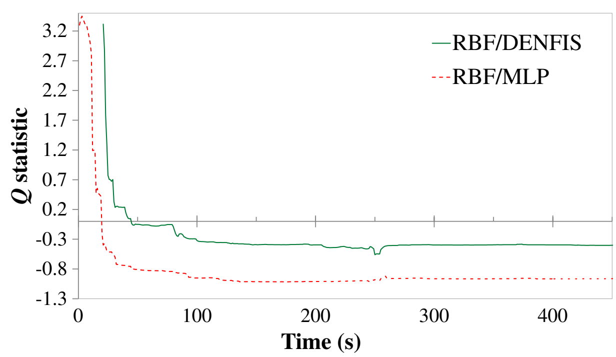

Fig. 7. Case I: Evolution of the Q statistic for the pairs RBF/DENFIS and RBF/MLP; a negative sign for Q implies an advantage for the RBF soft-sensor, whereas a positive sign implies an advantage for its rivals.

In order to test the soft-sensor, the DC Motor is simulated by changing the values for the two input voltages V_a and V_f every 5 time steps. The values for the two variables are drawn randomly, initially from a uniform distribution within the range [20 40]. After 250 time steps, the range for variable selection changes to [50 70]. The three adaptive soft-sensors, namely the adaptive RBF, DENFIS and adaptive MLP ones, are asked to predict the value of the current rotor speed at each time step.

Figures 5 and 6 show the evolution of the absolute relative error % (ARE%) and the number of RBF centers, respectively, as far as the adaptive RBF model is concerned, whereas Fig. 7 depicts the Q statistic[88] for the pairs RBF–DENFIS and RBF–MLP. In order to properly evaluate the online learning method proposed in this work, we use a technique known as "Predictive Sequential" or "Prequential",[88] where the error of a model is computed from the sequence of examples. For each new example in the stream, the actual model makes a prediction and the prequential error is computed based on an accumulated sum of a loss function between the prediction and observed values. The Q statistic is based on the prequential approach, and can further be used to compare two evolving algorithms A and B, by assessing their relative performance. It can be calculated as follows:

$$Q_i(A,B) = \log\left(\frac{s_i^A}{s_i^B}\right), \tag{8}$$

where s_i^A is the accumulated sum of a loss function between the predictions of algorithm A and the observed values. In this work, the sum of squared errors (SSE) is employed as the loss function; therefore s_i^A is calculated as follows:

$$s_i^A = \sum_{j=1}^{i} (y_j - \hat{y}_j^A)^2, \tag{9}$$

where $\hat{y}_j^A$ stands for the prediction of algorithm A for the jth time instant. For this particular example, the DENFIS algorithm was unable to provide a prediction for the first 20 data-points, therefore the Q statistic for the pair RBF/DENFIS is displayed from time instant 20 onwards. Table 2 depicts values for error-based metrics calculated over the entire time span 20–450, namely the maximum absolute relative error %, (maxARE%), the mean absolute relative error % (MARE%) and the standard deviation of the absolute relative error % (stdARE%), for the three models.

Table 2. Case I: Error-based metrics for the three soft-sensors.

Adaptive soft-sensor	maxARE%	MARE%	stdARE%
RBF	7.98	0.68	0.61
DENFIS	14.1	0.98	0.88
MLP	32.64	1.84	1.69

As it can be seen in Fig. 5, the very first predictions attempted by the RBF soft-sensor are rather inaccurate. This is expected, as the number of examples presented to the network up to that point is low, not allowing for the creation of an adequate number of RBF centers. Based on the Q statistic, as depicted in Fig. 7, the MLP-based adaptive soft-sensor is ahead during the first 20 time instants, whereas the DENFIS soft-sensor is unable to provide any predictions at all during this time period. From this point onwards, the RBF soft-sensor gains a clear advantage over its MLP counterpart, and keeps it for the rest of the simulation. As far as the comparison with DENFIS is concerned, it takes 45 time instants for the RBF soft-sensor to become better in terms of the Q statistic, although it obviously has become better earlier in terms of the prediction error. Though DENFIS adapts itself more efficiently during the first steps after time instant 20, the adaptive RBF model clearly outperforms it later on, as soon as enough RBF centers are added to the hidden layer.

As the simulation approaches time instant 250, the number of RBF centers is approximately stabilized to the value of 23. The change of input range occurring at that time reflects a change in the operating region of the system, testing the adaptation capabilities of the algorithms; only an adaptive model could successfully track the change, as a stationary model would essentially be asked to perform extrapolation. Following the change, the predictions of the three adaptive models initially deteriorate, as expected.

The adaptive FM algorithm adds centers to the RBF network in order to cover the new region in the input space and manages to quickly decrease the modeling error caused by extrapolation. After the change occurs, some of the previously selected RBF centers become obsolete and thus are deleted from the hidden layer. It can be seen that at the end of the simulation, the number of RBF nodes approximately converges to 22; this value is close to the number of RBF centers before the change occurred, but also to 23 which was the number of fuzzy rules produced by DENFIS at the end of the corresponding simulation. Notice that immediately after the change occurs, the Q statistic for the pair RBF/DENFIS becomes more negative, which means that the RBF soft-sensor adapts itself quicker to the new range of inputs. A few time steps later, DENFIS manages to compensate for the change and the Q statistic for both pairs converges to steady values.

The superiority of the adaptive RBF-based soft-sensor in this case is also confirmed by the metrics of Table 2.

4.2. *Case Study* II: *Online identification of a real digester reactor*

The continuous digester[89] is a chemical reactor where a very important process in a pulp and paper plant takes place. Its role is to convert wood chips to pulp by removing a wood component called lignin. The removal of lignin is achieved through combined chemical and thermal treatment, by means of adding a special mixture called the white liquor. The residual amount of lignin in the pulp exciting the digester is the kappa number, which is a critical parameter, affecting the quality of the produced paper. A typical continuous digester is depicted in Fig. 8.

In many digesters, the kappa number is measured offline, since online analyzers are expensive and often unreliable. On the other hand, the complexity and the high nonlinearity of the reactor makes it difficult to build soft-sensors for estimating the kappa number in real time, using first principles or simple stochastic methods. The use of black box identification techniques is more successful, but there are still some problems imposed by the fact that the reactor exhibits frequent changes in its dynamic behavior. Moreover, it is very often desirable to produce pulp of different quality, which means changing the operating region of the digester. Due to these reasons, an adaptive model would be very desirable, since it could serve either as an inferential sensor for the kappa number online estimation, or as the basis for an advanced model-based control scheme. In fact, the frequent changes in the dynamic behavior and operating region of the reactor pose serious challenges even for adaptive algorithms.

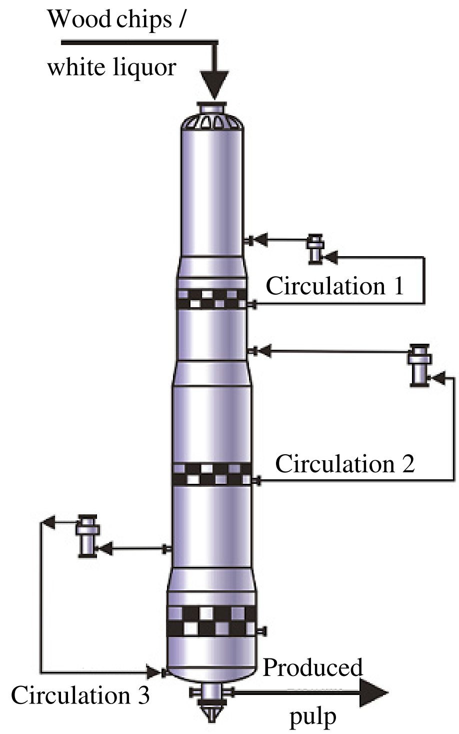

Fig. 8. A continuous digester.

For the particular application, a set of noisy dynamical data from an industrial continuous digester was available, consisting of 400 data points. The proposed methodology was used to build a discrete dynamic model that can predict the value of kappa number on an hourly basis. The large sampling time between predictions leaves enough time for the previous value of kappa number to be measured offline and then used to adapt the model. The input variables to the model were selected based on a previous offline modeling study.[90] To be more specific, the input vector consisted of hourly measured temperatures in three circulation zones around the reactor. Because of the very large retention times in the digester, past values of up to 12 h for each variable were used, summing up to a total of 36 input variables.

Similarly to the previous case, the three adaptive soft-sensors were employed for predicting the current value of the kappa number at each time step. Figures 9 and 10 show the evolution of the ARE% and the number of centers for the RBF model, respectively, whereas the Q statistic for the pairs RBF–DENFIS and RBF–MLP is displayed in Fig. 11. Table 3 depicts values for the maxARE%, MARE%

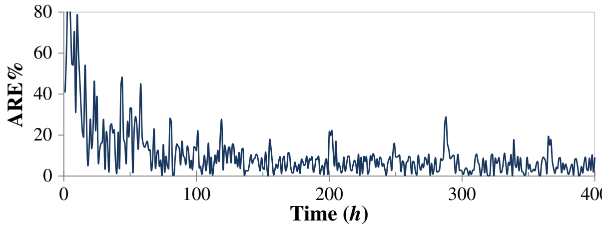

Fig. 9. Case II: Evolution of the ARE% for the RBF-based soft-sensor.

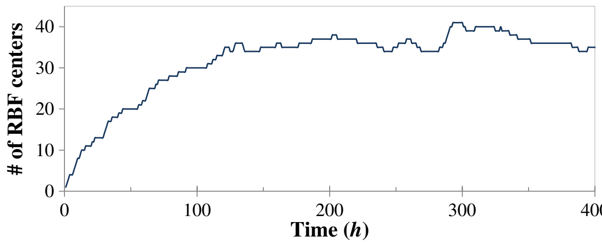

Fig. 10. Case II: Evolution of the number of RBF centers.

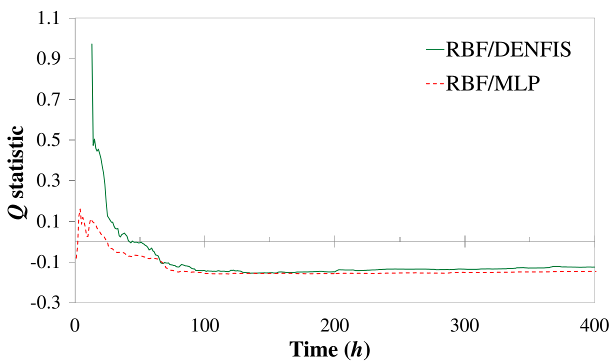

Fig. 11. Case II: Evolution of the Q statistic for the pairs RBF/DENFIS and RBF/MLP.

and stdARE% metrics, calculated over the time span 13–400 for the three models.

Once more, the RBF-based soft-sensor falls behind the other two schemes only for the first time steps, until a sufficient number of centers has been added to the hidden layer. Based on the Q statistic, the RBF model takes the lead over the MLP model at time instant 25, whereas it takes more time to outperform the DENFIS model, a fact that occurs at time instant 42 (it should be noted though that DENFIS was not able to provide predictions during the first 12 time steps). The RBF-based soft-sensor maintains its advantage for the rest of the simulation. From a certain point onwards, the Q statistic

Table 3. Case II: Error-based metrics for the three soft-sensors.

Adaptive soft-sensor	maxARE%	MARE%	stdARE%
RBF	48.3	8.92	8.1
DENFIS	48.9	10.52	9.0
MLP	53.5	12.44	11.7

for both pairs has almost converged to steady values. As far as the number of RBF centers selected by the adaptive FM algorithm is concerned, it progressively increases until reaching a value of 34 at time instant 120, and then continues with small variations around this number. This value was slightly higher compared to the number of fuzzy rules produced by DENFIS at the end of the corresponding simulation, which was equal to 29, though it is questionable whether the two values can be directly compared. The metrics presented in Table 3, also attest to the superiority of the adaptive RBF scheme over the remaining soft-sensors.

5. Conclusion

The FM algorithm offers an alternative to the standard method for training RBF networks, increasing the model accuracy and decreasing the computational load. In this work, the adaptive variant of the FM algorithm is exploited in order to produce a new methodology for building soft-sensors. At the heart of the soft-sensor lies an RBF network model, which starts with an empty hidden layer and evolves gradually with time. As new data become available from the system under identification, the RBF network is adapted on two levels: On the first level, the hidden layer is adjusted by adding RBF centers to cover sufficiently the input space and by deleting the redundant ones. On the second one, the synaptic weights are adapted using the RLS with exponential forgetting algorithm.

The resulting soft-sensor is validated through its application to the online identification of two different systems, namely a simulated nonlinear DC Motor and a real digester reactor. The results highlight the efficiency of the proposed approach, which manages to evolve the RBF network model in order to approximate the unknown systems. The method was compared to two different adaptive soft-sensor implementations, namely DENFIS and an MLP network trained with online backpropagation. It was shown that the proposed approach initially falls behind the other two methods in terms of accuracy; however, as it takes only a few time steps to accumulate a sufficient number of RBF nodes in the hidden layer, the adaptive RBF scheme soon takes the lead and offers increased accuracy and faster adaptation compared to its two rivals.

Future research plans include the extension of the proposed approach in order to include the non-symmetric variant of the FM algorithm for increased soft-sensor prediction accuracy, as well as application to other domains like robotics, fault detection and biometrics.

References

1. P. Kadlec and B. Gabrys, Adaptive local learning soft sensor for inferential control support, *IEEE Int. Conf. Computational Intelligence for Modelling Control & Automation*, Vienna, Austria (2008).
2. A. Jiménez, G. Beltrán, M. P. Aguilera and M. Uceda, A sensor-software based on artificial neural network for the optimization of olive oil elaboration process, *Sens. Actuators B Chem.* **129** (2008) 985–990.
3. L. Fortuna, S. Graziani and M. G. Xibilia, Comparison of soft-sensor design methods for industrial plants using small data sets, *IEEE Trans. Instrum. Meas.* **58** (2009) 2444–2451.
4. G. Melagraki, A. Afantitis, H. Sarimveis, O. Igglessi-Markopoulou and A. Alexandridis, A novel RBF neural network training methodology to predict toxicity to Vibrio fischeri, *Mol. Divers.* **10** (2006) 213–221.
5. A. P. Singh, T. S. Kamal and S. Kumar, Artificial neural network based soft estimator for estimation of transducer static nonlinearity, *Int. J. Neural Syst.* **14** (2004) 237–246.
6. C. Zanchettin and T. B. Ludermir, Hybrid neural systems for pattern recognition in artificial noses, *Int. J. Neural Syst.* **15** (2005) 137–149.
7. V. Kodogiannis and E. Wadge, The use of gas-sensor arrays to diagnose urinary tract infections, *Int. J. Neural Syst.* **15** (2005) 363–376.
8. M. A. González-Olvera, A. G. Gallardo-Hernández, Y. Tang, M. C. Revilla-Monsalve and S. Islas-Andrade, A discrete-time recurrent neurofuzzy network for black-box modeling of insulin dynamics in diabetic type-1 patients, *Int. J. Neural Syst.* **20** (2010) 149–158.
9. A. Y. Alanis, B. S. Leon, E. N. Sanchez and E. Ruiz-Velazquez, Blood glucose level neural model for type 1 diabetes mellitus patients, *Int. J. Neural Syst.* **21** (2011) 491–504.
10. R. J. Martis, U. R. Acharya, L. C. Min, K. M. Mandana, A. K. Ray and C. Chakraborty, Application of higher order cumulant features for cardiac health diagnosis using ECG signals, *Int. J. Neural Syst.* (2013), doi: 10.1142/S0129065713500147.
11. R. A. Osornio-Rios, J. P. Amezquita-Sanchez, R. J. Romero-Troncoso and A. Garcia-Perez, MUSIC-ANN analysis for locating structural damages in a truss-type structure by means of vibrations, *Comput.-Aided Civil Infrastruct. Eng.* **27** (2012) 687–698.
12. S. Arangio and F. Bontempi, Soft computing based multilevel strategy for bridge integrity monitoring, *Comput.-Aided Civil Infrastruct. Eng.* **25** (2010) 348–362.
13. D. H. Tran, B. J. C. Perera and A. W. M. Ng, Comparison of structural deterioration models for stormwater drainage pipes, *Comput.-Aided Civil Infrastruct. Eng.* **24** (2009) 145–156.
14. H. Xu and J. Humar, Damage detection in a girder bridge by artificial neural network technique, *Comput.-Aided Civil Infrastruct. Eng.* **21** (2006) 450–464.
15. H. Adeli and X. Jiang, Dynamic fuzzy wavelet neural network model for structural system identification, *J. Struct. Eng.* **132** (2006) 102–111.
16. X. Jiang and H. Adeli, Dynamic fuzzy wavelet neuroemulator for nonlinear control of irregular highrise building structures, *Int. J. Numer. Meth. Eng.* **74** (2008) 1045–1066.
17. A. Samant and H. Adeli, Enhancing neural network incident detection algorithms using wavelets, *Comput.-Aided Civil Infrastruct. Eng.* **16** (2001) 239–245.
18. W. Graf, S. Freitag, J. U. Sickert and M. Kaliske, Structural analysis with fuzzy data and neural network-based material description, *Comput.-Aided Civil Infrastruct. Eng.* **27** (2012) 640–654.
19. F. Y. Hsiao, S. S. Wang, W. C. Wang, C. P. Wen and W. D. Yu, Neuro-fuzzy cost estimation model enhanced by fast messy genetic algorithms for Semiconductor Hookup Construction, *Comput.-Aided Civil Infrastruct. Eng.* **27** (2012) 764–781.
20. S. Freitag, W. Graf and M. Kaliske, Recurrent Neural networks for fuzzy data, *Integr. Comput.-Aided Eng.* **18** (2011) 265–280.
21. O. Uluyol, K. Kim and E. O. Nwadiogbu, Synergistic use of soft computing technologies for fault detection in gas turbine engines, *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.* **36** (2006) 476–484.
22. E. D. Wandekokem, E. Mendel, F. Fabris, M. Valentim, R. J. Batista, F. M. Varejão and T. W. Rauber, Diagnosing multiple faults in oil rig motor pumps using support vector machine classifier ensembles, *Integr. Comput.-Aided Eng.* **18** (2011) 61–74.
23. J. Sedano, L. Curiel, E. Corchado, E. de la Cal and J. R. Villar, A soft computing method for detecting lifetime building thermal insulation failures, *Integr. Comput.-Aided Eng.* **17** (2010) 1–13.

24. G. A. P. Coronado, M. R. Muñoz, J. M. Armingol, A. de la Escalera, J. J. Muñoz, W. van Bijsterveld and J. A. Bolaño, Detection and classification of road signs for automatic inventory systems using computer vision, *Integr. Comput.-Aided Eng.* **19** (2012) 285–298.
25. X. Zeng and Y. Zhang, Development of recurrent neural network considering temporal-spatial input dynamics for freeway travel time modeling, *Comput.-Aided Civil Infrastruct. Eng.* **28** (2013) 359–371.
26. H. B. Celikoglu, An approach to dynamic classification of traffic flow patterns, *Comput.-Aided Civil Infrastruct. Eng.* **28** (2013) 273–288.
27. H. Adeli and X. Jiang, Neuro-fuzzy logic model for freeway work zone capacity estimation, *J. Transport. Eng.* **129** (2003) 484–493.
28. F. Badran, C. Mejia, S. Thiria and M. Crepon, Remote sensing operations, *Int. J. Neural Syst.* **6** (1995) 447–453.
29. H. Adeli and A. Panakkat, A probabilistic neural network for earthquake magnitude prediction, *Neural Netw.* **22** (2009) 1018–1024.
30. A. Panakkat and H. Adeli, Neural network models for earthquake magnitude prediction using multiple seismicity indicators, *Int. J. Neural Syst.* **17** (2007) 13–33.
31. A. Panakkat and H. Adeli, Recurrent neural network for approximate earthquake time and location prediction using multiple seismicity indicators, *Comput.-Aided Civil Infrastruct. Eng.* **24** (2009) 280–292.
32. Q. D. Tran, P. Kantartzis and P. Liatsis, Improving fusion with optimal weight selection in Face Recognition, *Integr. Comput.-Aided Eng.* **19** (2012) 229–237.
33. J. Mendes, F. Souza, R. Araújo and N. Gonçalves, Genetic fuzzy system for data-driven soft sensors design, *Appl. Soft Comput.* **12** (2012) 3237–3245.
34. E. Zamprogna, M. Barolo and D. E. Seborg, Optimal selection of soft sensor inputs for batch distillation columns using principal component analysis, *J. Process Control* **15** (2005) 39–52.
35. H. Adeli and S. L. Hung, An adaptive conjugate gradient learning algorithm for effective training of multilayer neural networks, *Appl. Math. Comput.* **62** (1994) 81–102.
36. N. Wang, M. J. Er, X. Y. Meng and X. Li, An online self-organizing scheme for Parsimonious and Accurate Fuzzy Neural Networks, *Int. J. Neural Syst.* **20** (2010) 389–403.
37. F. Liu and M. J. Er, A novel efficient learning algorithm for self-generating fuzzy neural network with applications, *Int. J. Neural Syst.* **22** (2012) 21–35.
38. H. Wersing, S. Kirstein, M. Götting, H. Brandl, M. Dunn, I. Mikhailova, C. Goerick, J. Steil, H. Ritter and E. Körner, Online learning of objects in a biologically motivated visual architecture, *Int. J. Neural Syst.* **17** (2007) 219–230.
39. O. Beyer and P. Cimiano, Online semi-supervised growing neural gas, *Int. J. Neural Syst.* **22** (2012) 1250023.
40. Z. Yang, K. L. Cameron, A. F. Murray and V. Boonsobhak, An adaptive visual neuronal model implementing competitive, temporally asymmetric Hebbian learning, *Int. J. Neural Syst.* **16** (2006) 151–162.
41. J. Platt, A Resource-allocating network for function interpolation, *Neural Comput.* **3** (1991) 213–225.
42. S. Tolu, M. Vanegas, J. A. Garrido, N. R. Luque and E. Ros, Adaptive and predictive control of a simulated robot arm, *Int. J. Neural Syst.* **23** (2013) 1350010.
43. G. Bello-Orgaz, H. D. Menéndez and D. Camacho, Adaptive k-means algorithm for overlapped graph clustering, *Int. J. Neural Syst.* **22** (2012) 1250018.
44. J. Lee and W. Pinheiro dos Santos, An adaptative fuzzy-based system to simulate, quantify, and compensate color blindness, *Integr. Comput.-Aided Eng.* **18** (2011) 29–40.
45. Z. M. Ma, F. Zhang, L. Yan and J. Cheng, Extracting knowledge from fuzzy relational databases with description logic, *Integr. Comput.-Aided Eng.* **18** (2011) 181–200.
46. L. Yan and Z. M. Ma, Comparison of entity with fuzzy data types in fuzzy object-oriented databases, *Integr. Comput.-Aided Eng.* **19** (2012) 199–212.
47. L. Yan and Z. M. Ma, Incorporating fuzzy information into the formal mapping from web data model to extended entity-relationship model, *Integr. Comput.-Aided Eng.* **19** (2012) 313–330.
48. C. M. Bo, J. Li, S. Zhang, C. Y. Sun and Y. R. Wang, The application of neural network soft sensor technology to an advanced control system of distillation operation, *Int. Joint Conf. Neural Networks*, Portland, Oregon (2002).
49. M.-D. Ma, J.-W. Ko, S.-J. Wang, M.-F. Wu, S.-S. Jang, S.-S. Shieh and D. S.-H. Wong, Development of adaptive soft sensor based on statistical identification of key variables, *Control Eng. Pract.* **17** (2009) 1026–1034.
50. A. Alexandridis, D. Triantis, E. Chondrodima, C. Stergiopoulos, G. Hloupis, I. Stavrakas and K. Ninos, An adaptive soft-sensor for non-destructive cement-based material testing, through the use of RBF networks, *IEEE Conf. Evolving and Adaptive Intelligent Systems (EAIS)*, Madrid, Spain (2012).
51. F. Petitjean, F. Masseglia, P. Gançarski and G. Forestier, Discovering significant evolution patterns from satellite image time series, *Int. J. Neural Syst.* **21** (2011) 475–489.
52. E. Nichols, L. J. McDaid and N. Siddique, Case study on a self-organizing spiking neural network for robot navigation, *Int. J. Neural Syst.* **20** (2010) 501–508.

53. N. R. Luque, J. A. Garrido, J. Ralli, J. J. Laredo and E. Ros, From sensors to spikes: Evolving receptive fields to enhance sensorimotor information in a robot-arm, *Int. J. Neural Syst.* **22** (2012) 1250013.
54. P. Kadlec, R. Grbic and B. Gabrys, Review of adaptation mechanisms for data-driven soft sensors, *Comput. Chem. Eng.* **25** (2011) 1–24.
55. J. Moody and C. Darken, Fast learning in networks of locally-tuned processing units, *Neural Comput.* **2** (1989) 281–294.
56. P. Anand, B. V. N. Siva Prasad and C. Venkateswarlu, Modeling and optimization of a pharmaceutical formulation system using radial basis function network, *Int. J. Neural Syst.* **19** (2009) 127–136.
57. A. Karim and H. Adeli, Comparison of the fuzzy — wavelet rbfnn freeway incident detection model with the California algorithm, *J. Transport. Eng.* **128** (2002) 21–30.
58. A. Karim and H. Adeli, Radial basis function neural network for work zone capacity and queue estimation, *J. Transport. Eng.* **129** (2003) 494–503.
59. S. Ghosh-Dastidar, H. Adeli and N. Dadmehr, Principal component analysis-enhanced cosine radial basis function neural network for robust epilepsy and seizure detection, *IEEE Trans. Biomed. Eng.* **55** (2008) 512–518.
60. Q. Junfei and H. Honggui, A repair algorithm for radial basis function neural network with application to chemical oxygen demand modeling, *Int. J. Neural Syst.* **20** (2010) 63–74.
61. D. Wu, K. Warwick, Z. Ma, M. N. Gasson, J. G. Burgess, S. Pan and T. Z. Aziz, Prediction of Parkinson's disease tremor onset using radial basis function neural network based on particle swarm optimization, *Int. J. Neural Syst.* **20** (2010) 109–116.
62. H. Adeli and A. Karim, Fuzzy-wavelet RBFNN model for freeway incident detection, *J. Transport. Eng.* **126** (2000) 464–471.
63. J. F. Qiao and H. G. Han, A repair algorithm for radial basis function neural network and its application to chemical oxygen demand modeling, *Int. J. Neural Syst.* **20** (2010) 63–74.
64. S. Colic, O. C. Zalay and B. L. Bardakjian, Responsive neuromodulators based on artificial neural networks used to control seizure-like events in a computational model of epilepsy, *Int. J. Neural Syst.* **21** (2011) 367–383.
65. U. R. Acharya, S. V. Sree, S. Chattopadhyay and J. S. Suri, Automated diagnosis of normal and alcoholic EEG signals, *Int. J. Neural Syst.* **22** (2012) 1250011.
66. U. R. Acharya, R. Yanti, J. W. Zheng, M. M. Krishnan, J. H. Tan, R. J. Martis and C. M. Lim, Automated diagnosis of epilepsy using cwt, hos and texture parameters, *Int. J. Neural Syst.* **23** (2013) 1350009.
67. L. Zhou, G. Yan and J. Ou, Response surface method based on radial basis functions for modeling large-scale structures in model updating, *Comput.-Aided Civil Infrastruct. Eng.* **28** (2012) 210–226.
68. U. Reuter, A fuzzy approach for modelling non-stochastic heterogeneous data in engineering based on cluster analysis, *Integr. Comput.-Aided Eng.* **18** (2011) 281–289.
69. H. Sarimveis, A. Alexandridis, G. Tsekouras and G. Bafas, A fast and efficient algorithm for training radial basis function neural networks based on a fuzzy partition of the input space, *Ind. Eng. Chem. Res.* **41** (2002) 751–759.
70. A. Alexandridis, P. Patrinos, H. Sarimveis and G. Tsekouras, A two-stage evolutionary algorithm for variable selection in the development of RBF neural network models, *Chemom. Intell. Lab. Syst.* **75** (2005) 149–162.
71. A. Alexandridis, M. Stogiannos, A. Kyriou and H. Sarimveis, An offset-free neural controller based on a non-extrapolating scheme for approximating the inverse process dynamics, *J. Process Control* **23** (2013) 968–979.
72. A. Alexandridis, E. Chondrodima and H. Sarimveis, Radial basis function network training using a non-symmetric partition of the input space and Particle Swarm Optimization, *IEEE Trans. Neural Netw. Learn. Syst.* **24** (2013) 219–230.
73. G. C. Marano, G. Quaranta and G. Monti, Modified genetic algorithm for the dynamic identification of structural systems using incomplete measurements, *Comput.-Aided Civil Infrastruct. Eng.* **26** (2011) 92–110.
74. R. Jafarkhani and S. F. Masri, Finite element model updating using evolutionary strategy for damage detection, *Comput.-Aided Civil Infrastruct. Eng.* **26** (2011) 207–224.
75. V. Plevris and M. Papadrakakis, A hybrid particle swarm — Gradient algorithm for global structural optimization, *Comput.-Aided Civil Infrastruct. Eng.* **26** (2011) 48–68.
76. L. Sgambi, K. Gkoumas and F. Bontempi, Genetic algorithms for the dependability assurance in the design of a long span suspension bridge, *Comput.-Aided Civil Infrastruct. Eng.* **27** (2012) 655–675.
77. R. Putha, L. Quadrifoglio and E. Zechman, Comparing ant colony optimization and genetic algorithm approaches for solving traffic signal coordination under oversaturation conditions, *Comput.-Aided Civil Infrastruct. Eng.* **27** (2012) 14–28.
78. T. Chabuk, J. A. Reggia, J. Lohn and D. Linden, Causally-guided evolutionary optimization and its application to antenna array design, *Integr. Comput.-Aided Eng.* **19** (2012) 111–124.
79. C. F. Fung, S. A. Billings and W. Luo, On-line supervised adaptive training using radial basis function networks, *Neural Netw.* **9** (1996) 1597–1617.

80. G. P. Liu, V. Kadirkamanathan and S. A. Billings, Online identification of nonlinear systems using Volterra polynomial basis function neural networks, *Neural Netw.* **11** (1998) 1645–1657.
81. G.-B. Huang, P. Saratchandran and N. Sundararajan, An efficient sequential learning algorithm for growing and pruning RBF (GAP-RBF) networks, *IEEE Trans. Syst., Man, Cybern. B, Cybern.* **34** (2004) 2284–2292.
82. A. Alexandridis, H. Sarimveis and G. Bafas, A new algorithm for online structure and parameter adaptation of RBF networks, *Neural Netw.* **16** (2003) 1003–1017.
83. J. Nie, Fuzzy control of multivariable nonlinear servomechanisms with explicit decoupling scheme, *IEEE Trans. Fuzzy Syst.* **5** (1997) 304.
84. K. J. Astrom and B. Wittenmark, *Adaptive Control* (Addison-Wesley, Reading, MA, 1995).
85. N. Kasabov and Q. Song, DENFIS: Dynamic evolving neural-fuzzy inference system and its application for time series prediction, *IEEE Trans. Fuzzy Syst.* **10** (2002) 144–154.
86. S. Haykin, *Neural Networks and Learning Machines* (Prentice Hall, Upper Saddle River, NJ, 2009).
87. P. C. Krause, *Analysis of Electric Machinery* (MacGraw-Hill International Editions, Singapore, 1987).
88. J. Gama, R. Sebastião and P. P. Rodrigues, Issues in evaluation of stream learning algorithms, in *Proc. 15th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, New York, NY, USA (2009).
89. A. Alexandridis and H. Sarimveis, Nonlinear adaptive model predictive control based on self-correcting neural network models, *AIChE J.* **51** (2005) 2495–2506.
90. H. Sarimveis, A. Alexandridis, S. Mazarakis and G. Bafas, A new algorithm for developing dynamic radial basis function neural network models based on genetic algorithms, *Comput. Chem. Eng.* **28** (2004) 209–217.