

Evolving Intelligent Systems: Methods, Learning, & Applications

Nikola Kasabov, Dimitar Filev

Abstract—The basic concept, formulation, background, and a panoramic view over the recent research results and open problems in the newly emerging area of Evolving Intelligent Systems are summarized in this short communication. Intelligent systems can be defined as systems that incorporate some form of reasoning that is typical for humans. Fuzzy Systems are well known for being able to formalize human knowledge that still separates humans from machines. Artificial Neural Networks have proven to be a useful form of parallel processing of information that employs principles from the organization of the brain. Finally, the evolution is a phenomenon that was initially used to solve optimization problems inspired by the progress in Genetic Algorithms, Evolutionary Computing, and Genetic Programming. These types of evolutionary algorithms are mimicking the natural selection that takes place in populations of living creatures over generations. More recently, the evolution of individual systems within their life-span (self-organization, learning through experience, and self-developing) has attracted attention. These systems called ‘evolving’ came as a result of the research on practical intelligent systems and on-line learning algorithms that are capable of extracting knowledge from data and performing a higher level adaptation of model structure as well as model parameters. Evolving systems can also be considered an extension of the multi-model concept known from the control theory, and of the on-line identification of fuzzy rule-based models. They can also be regarded as an extension of the methods for on-line learning neural networks with flexible structure that can grow and shrink. This new concept of evolving intelligent systems can also be treated in the framework of knowledge and data integration. Evolutionary, population / generation based computation, can be applied to optimize parameters and features of an individual system, that learns incrementally from incoming data. The specifics of this paper lays in the generalization of the recent advances in the development of evolving fuzzy and neuro-fuzzy models and the more analytical angle of consideration through the prism of knowledge evolution as opposed to the usually used data-centered approach. This powerful new concept has been recently introduced by the authors in a series of parallel works and is still under intensive development. It forms the conceptual basis for the development of the truly intelligent systems. A number of applications of this technique to a range of industrial and benchmark processes have been recently reported. Due to the lack of space only some of them will be mentioned primarily with illustrative purpose.

Prof. Nikola Kasabov, FRSNZ, SrMIEEEE, is the Director of the Knowledge Engineering and Discovery Research Institute, Auckland University of Technology, Private Bag 93000, Auckland, New Zealand. e-mail: nkasabov@aut.ac.nz, www.kedri.info

Dr. Dimitar Filev, SrMIEEEE, is Senior Technical Leader, Intelligent Control & Information Systems, Ford Research & Advanced Engineering, 2101 Village Road, MD #1437, Dearborn, MI 48121, E-mail: dfilev@ford.com

I. INTRODUCTION

IT is widely accepted that systems that are capable of decision making and reasoning, that possess knowledge, are regarded as ‘intelligent’ [1, 9]. Currently, it is recognized that the techniques that contribute to increase of the ‘machine intelligence quotient’ [3] of a system are primarily fuzzy logic, artificial neural networks, machine learning and evolutionary algorithms [2-8]. The rationale behind the intelligent label of those methods is their ability to represent and deal with knowledge. These branches form the triad of the so called *computational intelligence*.

The problem of adaptivity of intelligent systems and their use in on-line mode for real-time applications in industry, defense, advanced technology, biology and medicine has attracted research attention recently. This led during the last few years to the formation of the area of *evolving intelligent systems*.

We use further the term *evolving intelligent systems* to portray the phenomenon of gradual development of the system structure (rule-base or the architecture of the neural network that represents this system) and their parameters. While conventional adaptive techniques [43] are suitable to represent objects with slowly changing parameters, they can hardly handle complex systems with multiple operating modes or abruptly changing characteristics since it takes a long time after every drastic change in the system to relearn model parameters. The evolving systems paradigm is based on the concept of evolving (expanding or shrinking) model structure that is capable of adjusting to the changes in the objects that cannot solely be represented by parameter adaptation.

Fuzzy systems and their multiple neuro-fuzzy interpretations provide the natural foundation for the evolving systems. This can be understood from the dual role played by the fuzzy rule-based models. From a data-geometric perspective, the rules are representations of multiple clusters (bundles, granules) of input-output data that correspond to different system modes, operating regimes, physiological states and behaviors. The clusters define a decomposition of the input-output data space. From a systems perspective, the rules encompass a multiple model interpretation. Multiple rules with fuzzy predicates and crisp / fuzzy consequents along with radial basis neural networks define collections of multiple models that are represented by multiple focal (generalized) data kernels in the input-output space. Multiple rules with fuzzy predicates and functional (linear / nonlinear) consequents define collections of multiple linear / nonlinear models. The multiple model interpretation enables complex

system decomposition. The reasoning mechanism applied to the fuzzy rule-based models essentially provides a knowledge-driven, condition-based aggregation operator combining the models in a system in flexible manner that corresponds to the current data structures. Since the data streams are often non-stationary or they are driven by different operating conditions and modes, it is logical to assume the structure of the data to also be dynamic, that is, to evolve. New input-output data formations are identified by new clusters and are associated with new rules. Evolving data streams drive the creation of new clusters. The new clusters generate new rules and evolve the structure of the system.

Fuzzy rule base models and radial basis networks are hardly the only structures that can blend the data-geometric and multiple model interpretations. A similar concept was explored and developed through the works Rajbman and his colleagues [67]. The notion of this approach was to decompose the state space of a dynamic system into disjunct regions by using pattern recognitions methods and identify the system dynamics of these regions with multiple (generally) nonlinear models each with its own specific structure and parameters. Other disjunct multiple model representations for indirect adaptive control were proposed by Narendra and his co-workers [68-70]. In the realistic situations, however, such disjoint (crisp) decomposition might be hard to accomplish due to the inherent lack of natural region boundaries in the system. The fuzzy model allows one to replace the crisp decomposition by a fuzzy decomposition, and to replace the crisp switching between the models by an interpolative reasoning mechanism [29].

The evolving concept closely relates to the process of identifying new rules from data. The flexibility of the structures presented by the fuzzy and neural models (exhibited in their proven role of universal approximators) and their natural ability for information granulation constitute the basic characteristic features of the evolving systems. An evolving intelligent system continuously collects new data and attempts to integrate this data with the existing clusters and corresponding models. If this is possible, i.e. if the new data is compatible with the existing clusters / model structure it uses the new data parameter adaptation. In the opposite case it updates the structure by creating new rules and models. Therefore, it develops its structure and parameters continuously, always adapting and modifying its knowledge contents.

During last several years there are increasing number of publications that treat the theoretical development and applications in evolving systems in both fuzzy systems and neural networks domains [17-25, 31, 34-35, 44-66]. The 2006 International Symposium on Evolving Fuzzy Systems is recognition of the growing importance of this area brought together more than sixty high quality contributions [32].

The remainder of the paper is organized as follows. Section II outlines the main contributions and directions in the Evolving Connectionist Systems (ECOS) approach. Section

III summarizes the Evolving Fuzzy Systems (EFS) methodology for system representation and learning. Section IV discusses some of the main evolving systems applications in industry, medicine, bioinformatics, neuroinformatics, and business. Section V concludes the paper.

II. EVOLVING CONNECTIONIST SYSTEMS (ECOS)

A. General Principles of ECOS

Evolving connectionist systems (ECOS) are modular connectionist-based systems that evolve their structure and functionality in a continuous, self-organised, on-line, adaptive, interactive way from incoming information. They can process both data and knowledge in a supervised and/or unsupervised way [10,11].

ECOS learn local models from data through clustering of the data and associating a local output function for each cluster. Clusters of data are created based on similarity between data samples either in the input space (this is the case in some of the ECOS models, e.g. the dynamic neuro-fuzzy inference system DENFIS [25]), or in both the input space and the output space (this is the case in the EFuNN models [10]). Samples that have a distance to an existing cluster center (rule node) N of less than a threshold R_{max} (for the EFuNN models it is also needed that the output vectors of these samples are different from the output value of this cluster center in not more than an error tolerance E) are allocated to the same cluster N_c . Samples that do not fit into existing clusters, form new clusters as they arrive in time. Cluster centers are continuously adjusted according to new data samples, and new clusters are created incrementally.

The similarity between a sample $S = (\mathbf{x}, \mathbf{y})$ and an existing rule node $N = (W_1, W_2)$ can be measured in different ways, the most popular of them being the normalized Euclidean distance:

$$d(S, N) = [\sum_{(i=1, \dots, n)} (x_i - W_1(i))^2] / n, \quad (1)$$

where n is the number of the input variables.

ECOS learn from data and automatically create or update a local output function for each cluster, the function being represented in the W_2 connection weights, thus creating local models. Each model is represented as a local rule with an antecedent – the cluster area, and a consequent – the output function applied to data in this cluster, e.g.:

IF (data is in cluster N_c) THEN (the output is calculated with a function F_c) (2)

Implementations of the ECOS framework require connectionist structures that support these principles. One implementation of ECOS is the evolving fuzzy neural network (EFuNN).

B. Evolving Fuzzy Neural Network EFuNN

A general EFuNN architecture has 5 feed-forward layers and a feedback layer of neurons, but the second and the fourth fuzzy

representation layers and also - the feedback layer, are optional – Fig. 1. The third layer contains rule nodes that evolve through supervised / unsupervised learning. The rule nodes represent prototypes of input-output data associations. Each rule node r is defined by two vectors of connection weights, $W_1(r)$ and $W_2(r)$, the latter being adjusted through supervised learning based on the output error, and the former being adjusted through unsupervised learning based on a similarity measure within a local area of the problem space. The fourth layer of neurons represents fuzzy quantization for the output variables, similar to the input fuzzy neurons representation. The fifth layer represents the real values for the output variables.

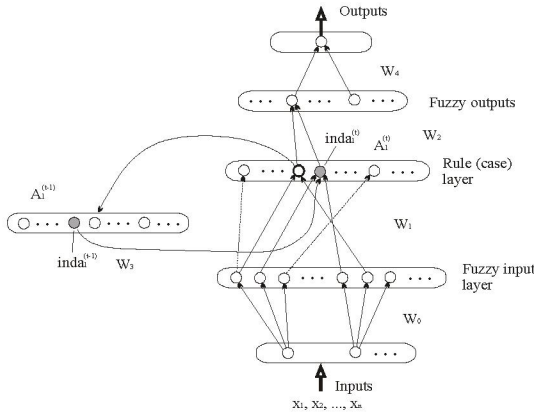


Figure 1 An EFuNN architecture with a short term memory and feedback connections (adapted from [10, 11, 46])

The evolving process can be based on either of the two assumptions: (1) rule nodes exist prior to learning and only connections are created during learning; (2) all nodes are created during the evolving process.

Each rule node (e.g., r_1) represents an association between a hyper-sphere from the fuzzy input space and a hyper-sphere from the fuzzy output space, the $W_1(r_j)$ connection weights representing the co-ordinates of the centre of the sphere in the fuzzy input space, and the $W_2(r_j)$ – the co-ordinates in the fuzzy output space. The radius of an input hyper-sphere of a rule node is defined as $(1 - Sthr)$, where $Sthr$ is the *sensitivity threshold* parameter defining the minimum activation of a rule node (e.g., r_1) to an input vector (e.g., (Xd_1, Yd_2)) in order for the new input vector to be associated to this rule node.

Through the process of associating (learning) a new data vector Xd to a rule node, the centre of this node hyper-sphere is adjusted in the fuzzy input space depending on a *learning rate* lr_1 and in the fuzzy output space depending on a *learning rate* lr_2 . The adjustment of the centre r_1^1 to its new position r_1^2 can be represented mathematically by the change in the connection weights of the rule node r_1 from $W_1(r_1^1)$ and $W_2(r_1^1)$ to $W_1(r_1^2)$ and $W_2(r_1^2)$ as it is presented in the following vector operations:

$$W_1(r_1^2) = W_1(r_1^1) + lr_1 * Ds(Xd, W_1(r_1^1)) \quad (3)$$

$$W_2(r_1^2) = W_2(r_1^1) + lr_2 * Err(Yd, Yd') * A_1(r_1^1) \quad (4)$$

where: $Err(Yd, Yd') = Ds(Yd, Yd')$ is the distance between the desired and the obtained in the system output vectors in the output space; $A_1(r_1^1)$ is the activation of the rule node r_1^1 for the input vector Xd .

While the connection weights from W_1 and W_2 capture spatial characteristics of the learned data (centers of hyper-spheres), the temporal layer of connection weights W_3 from Fig. 1 captures temporal dependences between consecutive data examples. If the winning rule node at the moment $(t-1)$ (to which the input data vector at the moment $(t-1)$ was associated), was $r_1 = inda_1(t-1)$, and the winning node at the moment t is $r_2 = inda_1(t)$, then a connection between the two nodes is established as follows:

$$W_3(r_1, r_2)^{(t)} = W_3(r_1, r_2)^{(t-1)} + lr_3 * A_1(r_1)^{(t-1)} * A_1(r_2)^{(t)} \quad (5)$$

where: $A_1(r)^{(t)}$ denotes the activation of a rule node r at a time moment (t) ; lr_3 defines the degree to which the EFuNN associates links between rules (clusters, prototypes) that include consecutive data examples (if $lr_3 = 0$, no temporal associations are learned in an EFuNN).

The following is a *new learning rule* that takes into account both spatial similarity and temporal correlation) through introducing two parameters Ss and Tc , such that the activation of a rule node r for a new data example d_{new} is defined as the following vector operation:

$$A_1(r, d_{new}) = f(Ss * D(W_1(r), d_{new}) + Tc * W_3(r^{(t-1)}, r)) \quad (6)$$

where: f is the activation function of the rule node r ; $D(W_1(r), d_{new})$ is the normalized fuzzy distance between the new input vector and the $W_1(r)$ representing the spatial component; $r^{(t-1)}$ is the winning neuron at time moment $(t-1)$. The second term in equation (16) represents the temporal component.

An EFuNN functional implementation can include pruning nodes and aggregating nodes [10,11]. An example of a pruning rule is:

IF (a rule node r_j is *OLD*) AND (average activation $A_{1av}(r_j)$ is *LOW*) AND (the density of the neighbouring area of neurons is *HIGH* or *MODERATE*) THEN the probability of pruning node (r_j) is *HIGH*.

Nodes can also be aggregated [10,11].

C. Dynamic Evolving Neuro-Fuzzy Inference Systems (DENFIS) [25]

While EFuNN is a fuzzy neural network that evolves incrementally its structure and functionality using supervised clustering, DENFIS is a dynamic fuzzy inference system that incrementally creates Takagi-Sugeno fuzzy rules to accommodate data in unsupervised learned clusters. New fuzzy rules are created and updated during the operation of the system. At each time moment the output of DENFIS is calculated through a fuzzy inference system based on m -most activated fuzzy rules which are dynamically selected from the existing fuzzy rule set. As the knowledge, fuzzy rules can be

inserted into DENFIS before, or during its learning process and, they can also be extracted during the learning process or after it. The fuzzy rules used in DENFIS are indicated as follows:

$$R_j: \text{ if } x_1 \text{ is } F_{1j} \text{ and } x_2 \text{ is } F_{12} \text{ and } \dots \text{ and } x_P \text{ is } F_{1P},$$

$$\text{ then } y_l = b_{l0} + b_{l1}x_1 + b_{l2}x_2 + \dots + b_{lP}x_P \quad (7)$$

where “ x_j is F_{lj} ”, $l = 1, 2, \dots, m$; $j = 1, 2, \dots, P$, are $M \times P$ fuzzy propositions that form m antecedents for m fuzzy rules respectively; x_j , $j = 1, 2, \dots, P$, are antecedent variables defined over universes of discourse X_j , $j = 1, 2, \dots, P$, and F_{lj} , $l = 1, 2, \dots, M$; $j = 1, 2, \dots, P$ are fuzzy sets defined by their fuzzy membership functions $\mu_{F_{lj}}: X_j \rightarrow [0, 1]$, $l = 1, 2, \dots, M$; $j = 1, 2, \dots, m$, are the consequent variables defined by linear functions.

In DENFIS, F_{lj} are defined by a *Gaussian* membership function. All fuzzy rules in DENFIS are created and updated during a possible ‘one-pass’ training process by applying the Evolving Clustering Method (ECM) and the Weighted Recursive Least Square Estimator with Forgetting Factors (WRLSE) [25].

The ECOS models have the following advantages: (1) incremental, fast learning (possibly ‘one pass’); (2) on-line adaptation; (3) ‘open’ structure; (4) allowing for time and space representation based on biological plausibility; (5) rule extraction and rule insertion; (6) data and knowledge integration (as discussed below).

D. Integrating Knowledge (Old Models) and Data in ECOS

As the *eCI*, and ECOS in particular, are adaptive, knowledge-based systems, they can accommodate both existing knowledge on the problem (e.g. formulas, models) and new data, allowing for incremental adaptation of the system’s rule representation.

In many domain areas, such as medical decision support, there are existing regression formulas and new data is accumulating in time, making the integration of both an important issue for a better decision support.

In [46,47] ECOS are used to accommodate regression formulas and new data in the following way. The formula is first used to generate “historical” data. This data is used to train an ECOS as an initial knowledge representation architecture. Then the ECOS system is further trained (adapted) on the new data.

In [48] a novel method of “Integrated kernel-regression knowledge-based neural networks” is presented for the integration of several, used in practice, regression formulas and new data into one system. The method optimizes the size of the clusters of the data using different kernels, and for each cluster – a suitable type of regression is chosen and the parameters are adapted on the data:

$$y(x) = G_1(x) F_1(x) + G_2(x) F_2(x) + \dots + G_M(x) F_M(x) \quad (8)$$

where, $x = [x_1, x_2, \dots, x_P]$ is the input vector; y is the output vector; G_l are kernel functions; and F_l are regression formulas, $l = 1, 2, \dots, M$.

E. Incremental parameter and feature optimization using GA, incremental PCA and LDA

An ECOS evolves its structure and functionality in time from incoming data, for which the dynamics may not be known in advance. That requires an incremental (possibly - on-line) parameter and feature optimization. One way to optimize these parameters and obtain an optimal for the time moment model according to certain criteria (e.g. classification accuracy) is through evolutionary computation, e.g. GA [7]. GA optimization can be applied on a population of individual models that are trained and tested on consecutive chunks of data, so that at any time of the operation of the ECOS the best model (e.g. the model with the highest accuracy/ fitness) is selected. A methodology and examples are given in [55, 59].

In Fig. 2 a simple ECOS model, called ECF, is optimized with the use of GA. ECF is characterized by 4 parameters (maximum field radius R_{max} , minimum field radius R_{min} , number of nodes m to use for a new vector; number of membership functions, epochs to train) and initial 12 input features describing the outcome of DLBCL cancer of 56 patients [40] is optimized as shown in fig. 2.

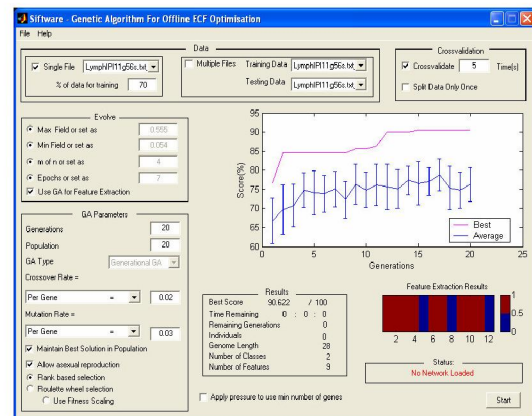


Figure 2 Using GA for parameter and feature optimization of a simple ECOS – ECF (experiments are done in a software environment NeuCom – www.theneucom.com)

The variables are a clinical variable (IPI index) and 11 genes selected in for the prognosis of DLBCL cancer outcome. In the experiment shown in Fig. 2 both the ECF parameters and features are optimised with the use of a GA which ran over 20 generations. At each generation, there are 20 ECF models in a population, having different parameter values and feature sets, and a fitness criteria of overall highest accuracy for the smallest number of features is used. The optimal ECF parameters are given in the figure and the best model has an overall accuracy of 90.66%, which is higher than any of the non-optimised models. The optimal values of all the ECF parameters and also the used variables (variables 5, 8 and 12 are not included) are shown in the figure.

In some cases, PCA or LDA transformations need to be performed on the input feature set to obtain a more compact input vectors and to improve the accuracy of the model. Incremental PCA and incremental LDA methods are presented in [49,54]. After features are selected in an incremental way, the ECOS is adapted to these features.

F. A Framework of Multimodel ECOS

So far, this paper described some methods for building adaptive, evolving, knowledge based models from data. eCI may require several evolving models. Fig. 3 presents a framework of eCI, that consists of several parts: several e-models (EM), higher level decision part, adaptation part, feature selection part where new features may be added in time, knowledge (rules) extraction part, interaction with environment and an output module [11].

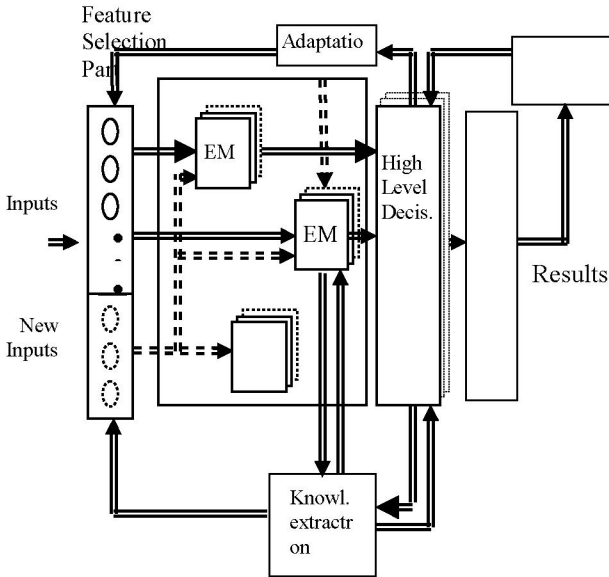


Figure 3. A general framework for an e-intelligent system (eCI)

III. EVOLVING FUZZY SYSTEMS (EFS)

A. EFS Structure

Evolving fuzzy systems are families of rules with fuzzy antecedents and generic consequents:

$$\text{IF } x_1 \text{ is close to } x_1^{i*} \text{ AND } \dots x_n \text{ is close to } x_n^{i*} \\ \text{THEN } y^i = f^i(x) \quad (9)$$

where R^i denotes the i^{th} fuzzy rule; $i=[1,N]$; x_j is close to x_j^{i*} is the j^{th} antecedent sets of the i^{th} rule, $j=[1,n]$ and x_j^{i*} is its focal point; y^i is the output of the i^{th} linear subsystem.

The model (9) covers a wide range of multiple model structures dependent on the type of the consequent:

- It is the first order Takagi-Sugeno type when the consequents $y^i = f^i(x)$ are linear functions of the input vector x , similarly to the model (7) considered in DENFIS:

$$f(x) = x_e^T \pi^i, x_e^T = [1 \ x^T] \quad (10)$$

where π^i is the parameter vector $\pi^i = [a_{0i} \ a_{1i} \ \dots \ a_{ni}]^T$ or an $m \times (n+1)$ dimensional matrix in the multiple output case of m outputs) of i^{th} local linear sub-system.

- It is the simplified Mamdani (sM) model (known also as zero order Takagi-Sugeno model) when the consequents are singletons or the consequent fuzzy set is replaced by its defuzzified value [29]:

$$f^i = a_i \quad (11)$$

- The consequent model can be also any nonlinear function, including a first principle based nonlinear model [29], a neural mapping, or a relational fuzzy model [71].

The overall output of the system, y is formed as a collection of loosely combined multiple subsystems, $y^i = f^i(x)$, with simpler structure that are combined by a linear combiner

$$y = \sum_{i=1}^N \lambda^i y^i \quad (12)$$

where λ^i is the normalized firing level of the i^{th} rule and N is the number of rules:

$$\lambda^i = \frac{\tau^i}{\sum_{j=1}^N \tau^j}, \tau^i = \prod_{j=1}^n \mu_j^i(x_j) \quad (13)$$

and μ_j^i is the membership value of the j^{th} input (antecedent) x_j , $j=[1,n]$:

$$\mu_j^i = e^{-\frac{4 \|x - x^{i*}\|_j^2}{(\sigma_j^i)^2}} \quad (14)$$

The linear combiner weights λ^i 's are driven by the firing levels of the rules. Dynamic reinforcement of certain models can be accomplished if (14) is replaced by more advanced aggregating operators, e.g.

$$\lambda^i = \frac{(\tau^i)^\beta}{\sum_{j=1}^N (\tau^j)^\beta} \quad (15)$$

where increased values of parameter β assign higher priorities to the consequent models associated with higher firing levels [29] gradually transforming the multiple model representation (9) from a universal approximator into a classifier.

B. eTS Learning – a Blend between Recursive Clustering and Recursive Least Square Learning

EFS is learned by combining unsupervised learning with respect to the antecedent part of the model (9) with the supervised in terms of the consequent parameters. Each one of the fuzzy rules of type (9) operate in certain sub-area of the input/output data space, $z_t = [x^T; y^T]^T$, $z_t = R^{n+m}$. To identify these regions, one can employ real-time clustering. The unsupervised clustering algorithm continuously analyzes the input-output data streams and identifies emerging new data structures.

The eTS learning algorithm [12, 13] clusters the input-output space into N fuzzy regions. The clusters define a fuzzy partitioning of the input space into subsets that are obtained by projecting the clusters centers onto the space of input (antecedent) variables. The learning algorithm also assigns to each of the clusters a linear subsystem. The eTS learning algorithm is a density-based clustering that stems from the Mountain clustering method [72] and extension called the Subtractive clustering [73] the *potential* was calculated for each data sample.

The eTS learning method expands the Mountain / Subtractive to the real time domain transforming it in a powerful tool for on-line learning with numerous applications in classification, rule-base generation, prediction, novelty detection [12, 35]. The eTS clustering is based on the recursive calculation of the potential $P_t(z_t)$ of each data point z_t in the input-output space $z_t = R^{n+m}$:

$$P_t(z_t) = \frac{t-1}{(t-1)(a_t+1) - 2c_t + b_t},$$

$$\text{where } a_t = \frac{p}{\sum_{j=1}^p (z_t^j)^2}; b_t = \sum_{i=1}^{t-1} \sum_{j=1}^p (z_i^j)^2; c_t = \sum_{j=1}^p z_t^j f_t^j; f_t^j = \sum_{i=1}^{t-1} z_i^j$$

and the potential of the clusters centers

$$P_t(z^*) = \frac{(t-1)P_{t-1}(z^*)}{(t-2) + P_{t-1}(z^*) + P_{t-1}(z^*) \|z^* - z_{t-1}\|^2}$$

Existing cluster centers are updated only if the new data z_t point is substantially different than the existing clusters as detailed in [74]. If the input vector associated with the current data point is too close to one of the existing antecedent focal points, i.e.

$$\min_{i=1}^N \|x_t - x^*\| < T \text{ then } z_t \text{ replaces the } i\text{-th cluster center, } i = \arg$$

$$\min_{i=1}^N \|x_t - x^*\| < T \text{ (where } T \text{ is a constant in the range } [0.15 -$$

0.30]). Otherwise z_t becomes a new cluster center.

For fixed cluster centers the EFS model transforms into a linear model

$$y = \psi^T \theta \quad (16)$$

where $\theta = [\pi_1^T, \pi_2^T, \dots, \pi_N^T]^T$ is a vector formed by the subsystem parameters $\pi_i = [\pi_{i0}, \pi_{i1}, \dots, \pi_{in}]$, $i = [1, N]$, and $\psi = [\lambda_1 x^T, \lambda_2 x^T, \dots, \lambda_N x^T]^T$ is a virtual input vector, which is formed by the system inputs that are weighted by the normalized activation levels λ_i , $i = [1, N]$. Under the linearity assumption, the subsystem parameters can be learned recursively using the Recursive Least Square (RLS, Kalman Filter [43, 27]) algorithm:

$$\theta_t = \theta_{t-1} + C_t \psi_t (y_t - \psi_t^T \theta_{t-1}) \quad (17)$$

$$C_t = C_{t-1} - \frac{C_{t-1} \psi_t \psi_t^T C_{t-1}}{1 + \psi_t^T C_{t-1} \psi_t} \quad (18)$$

where C is the covariance matrix of the virtual input vectors ψ . Obtained solution is optimal in least square sense.

The linearity assumption fails when the cluster centers are continuously updated. The correct recursive learning algorithm in this case is the Extended Kalman Filter [27]. One of the main advantages of the eTS learning method as a tool for developing EFS is due to its prototype-based clustering nature. The cluster centers remain unchanged except for the finite number of cases when a new center is created. Therefore, the linearity condition is consistently satisfied unless a new cluster is created. The problem of parameter learning when a drastic change in the structure (due to a newly created cluster) occurs is addressed by selectively resetting the covariance matrix C in (18). It is implemented by resetting the covariance matrix to a diagonal matrix with large values of the main diagonal elements when a new cluster is created. In this sense it is significantly different from the mean-based approaches k -nearest Neighbour Algorithm (k -NN) and Self-Organizing Maps (SOM) since these methods locate the centers of clusters at the mean of the local data structures and continuously update their positions. These cluster methods continuously update the cluster centers. Consequently, the assumption for linearity of the resulting model (16) is no longer valid.

In the above discussion we focused on the learning of EFS belonging to the cases of TS fuzzy models (10) and the simplified Mamdani (sM) models (11). For nonlinear multiple models in (9) the RLS method is substituted by an alternative recursive optimization method - the Extended Kalman Filter.

C. fSPC - Learning through Distance Based Output / Input Clustering

The simplified Mamdani (sM) model

$$\text{IF } x \text{ is close to } x_i^* \text{ THEN } y \text{ is } y_i^* \quad (19)$$

is applicable to slowly changing evolving systems that are dominated by fewer operating modes and rarely changing steady states. This is the case of the typical industrial systems that are designed to operate predominantly in certain operating modes where the transitions between the modes are rather exceptions. Since the sM models are special case of the TS model, this type of systems can be learned by the method of learning EFS described in the previous section. The fSPC learning approach [31] provides an alternative and pragmatic way for developing EFS that are based on the sM model. The fSPC is algorithm is inspired by the Statistical Process Control (SPC) - a method for process variability monitoring. The SPC control procedure naturally clusters the system output data into granules (clusters) that relate to same process control conditions and that are characterized with similar system behavior. The boundaries of the output granules (the process control limits) are defined through the T^2 Hotelling statistics:

$$(y - y_i^*)' C_{y_i}^{-1} (y - y_i^*) < \chi_{p, \alpha}^2 \quad (20)$$

where C_{y_i} is the output covariance and $\chi_{p, \alpha}^2$ is the $(1 - \alpha)$ th value of the chi-squared distribution with p degrees of freedom and α is the probability of a false alarm, e.g. $\chi_{2, 0.0027}^2 = 11.8290$, $\chi_{3, 0.0027}^2 = 14.1563$, while $\chi_{1, 0.0027}^2 = 9$

corresponds to the well-known $\pm 3\sigma$ limit rule for the case of a single output. From a system perspective the areas where the process is under statistical control can be considered as steady states related to a certain operating mode. In these states the system output can be approximated with a high probability with its mean y_i^* .

The output granules induce corresponding granules (clusters) in the input domain and define the parameters of the rule antecedents:

$$close =: \exp(-0.5(x - x_i^*)' C_{xi}^{-1} (x - x_i^*))$$

where x_i^* is the vector of input means and C_{xi} is the input covariance matrix. A sM model rule is the label of a pair of corresponding input/output granules.

For outputs that belong to an output granule, i.e. satisfy condition (20), the rule parameters associated with the respective granule are recursively updated through exponential smoothing:

$$y_i^*(t+1) = \alpha y_i^*(t) + (1 - \alpha) y(t+1);$$

$$C_{y_i}(t+1) = \alpha C_{y_i}(t) + (1 - \alpha) (y(t+1) - y_i^*(t)) (y(t+1) - y_i^*(t))^T$$

$$x_i^*(t+1) = \alpha x_i^*(t) + (1 - \alpha) x(t+1);$$

$$C_{x_i}(t+1) = \alpha C_{x_i}(t) + (1 - \alpha) (x(t+1) - x_i^*(t)) (x(t+1) - x_i^*(t))^T$$

where α , $0 < \alpha < 1$, is a learning rate parameter. Outputs that are not compatible with any of the existing output granules, i.e. do not satisfy condition (20), initialize new pair of input and output granules.

IV. APPLICATIONS

ECOS and EFS in general have been used for a range of applications so far, where adaptation to new data and knowledge representation are crucial requirements. Here we present only few examples of them.

A. Industrial Applications – Process Modeling

A simplified Mamdani model type EFS is applied to predict the average paint film thickness on the vehicle body for a given combination of factors governing the process in automotive paint booths [31]. These variables include: i. fluid flow rates of the applicators (robotized bells and guns) – the main parameter directly affecting the film thickness; ii. air down draft velocity, temperature and humidity (air is continuously supplied to the booth to remove the paint over spray and to virtually separate individual sections of the booth).

The EFS approximates the relationship between the process variable and paint film thickness on the horizontal and vertical surfaces of the vehicle body resulting in three alternative sM models (19) of the paint film thickness on the left and right vertical, and horizontal surfaces. The model maps the vector of process inputs $x = [FF$ (average fluid flow rate); DDb (down draft, bell zone); DDr(down draft, reciprocator zone); T (air temperature); H (air humidity)] to process output $y = FT$ (average film thickness).

By applying fSPC learning algorithm we obtain three sets of rule parameters $x^*(i)$, $y^*(i)$, $S^*(i)$. One of the models covering the horizontal surface input/output data shown in Figures 8 & 9 is listed below.

If $FF \approx 147.59$ and $DDb \approx 66.90$ and $DDr \approx 69.85$ and $T \approx 80.57$ and $H \approx 42.08$ Then $FB \approx 0.82$

If $FF \approx 150.68$ and $DDb \approx 66.56$ and $DDr \approx 70.42$ and $T \approx 82.41$ and $H \approx 33.82$ Then $FB \approx 0.80$

If $FF \approx 138.53$ and $DDb \approx 64.88$ and $DDr \approx 66.72$ and $T \approx 81.12$ and $H \approx 26.91$ Then $FB \approx 0.78$

If $FF \approx 129.87$ and $DDb \approx 60.04$ and $DDr \approx 62.23$ and $T \approx 80.50$ and $H \approx 21.56$ Then $FB \approx 0.75$

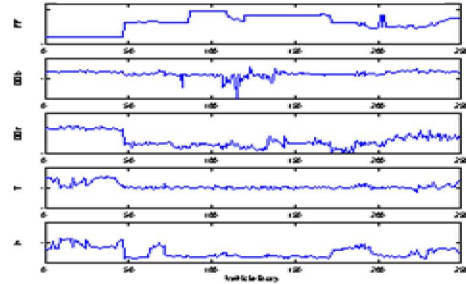


Figure 8. Input process data – horizontal surface: $x = [FF$ (average fluid flow rate); DDb (down draft, bell zone); DDr (down draft, reciprocator zone); T (air temperature); H (air humidity)] (adapted from [31])

Linguistic quantifiers close (\approx) are parameterized differently for individual rule antecedents and are completely defined by the vectors $x^*(i)$ and antecedent covariance matrices C_{xi} , $i=[1, 3]$.

A set of input/output data from 250 painted vehicles (part of the input data (horizontal surfaces) is presented in Figure 9; the actual scale and measurement units are omitted. The sM model is used as one step predictor of the film thickness in a supervisory adaptive control system that continuously adjusts the fluid flow rates and down drafts minimizing a cost function that optimizes desired film thickness and process efficiency.

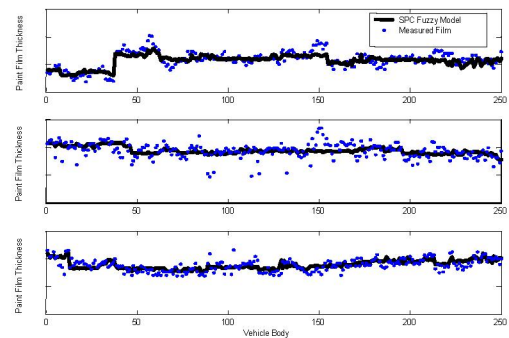


Figure 9. Measured vs. one step ahead predicted paint film thickness data

C. Medical Decision Support

In many medical decision support systems data is being made continuously, but the practises still use only one or two well established regression formulas for a prognostic or diagnostic

purposes. The area needs new techniques for adaptive learning and data and knowledge integration. Such techniques are offered in eCI. An example is give below.

A renal function prognostic system with the use of DENFIS is presented in [51]. The initial data set had 447 samples, collected at hospitals in New Zealand and Australia. Each of the patient records includes six variables (features): 1) age, 2) gender, 3) serum creatinine, 4) serum albumin, 5) race, 6) blood urea nitrogen concentrations, and one output - the glomerular filtration rate value (GFR) [38]. For every data cluster, a local model in DENFIS is derived as a logistic regression – Fig. 10.

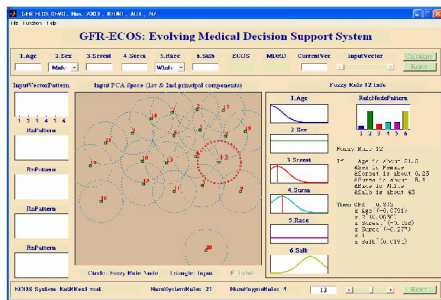


Figure 10. A snapshot of an adaptive medical decision support system for renal function evaluation [51]. The fuzzy rule on the right side is a regression model of the data derived and updated for the highlighted cluster.

C. Bioinformatics

Bioinformatics is the area concerned with the biological data storage, analysis, representation, modeling and knowledge discovery. A review of problems and possible solutions is given in [52, 59]. Several problems in bioinformatics have been successfully solved with the use of adaptive eCI:

(a) *Micro-array gene expression data analysis and pattern discovery* [40,59]. Figure 11 shows a graphical representation of 5 EFuNN rules, each representing a profile of samples clustered together, each of them belonging to the class of good prognosis (class 1) or - bad prognosis (class 2) [11, 46, 52, 56].

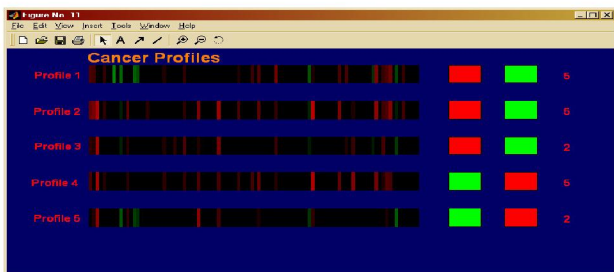


Figure 11. A graphical representation of 5 EFuNN rules, each representing a profile of samples clustered together, and belonging to the class of good prognosis (class 1), or - bad prognosis (class 2) [11,46,40].

(b) Gene regulatory network modeling (GRN)

GRNs describe the regulatory interaction between genes in a cell [42,59]. Co-expressed genes over time relate to each other – either one regulates the other, or both are regulated by same other genes. eCI are useful tools for building adaptive GRN models from time course gene expression data [50,53]. In [60] EFuNN and DENFIS have been used to derive a GRN of 4 genes from a cell line time course data. The GRN model is then used to predict future values of the genes over time. Rules can be extracted that explain the relationship between the expression of genes at different time moments, e.g.:

*IF g13(t) is High (0.87) and g23(t) is Low (0.9)
THEN g87 (t+dt) is High (0.6) and g103(t+dt) is Low*

C. Neuroinformatics and Brain Study

Brain models can be evolved incrementally from EEG brain data collected from individuals that belong to different categories, or from different brain states of the same individual, using standard EEG equipment – Fig. 12.

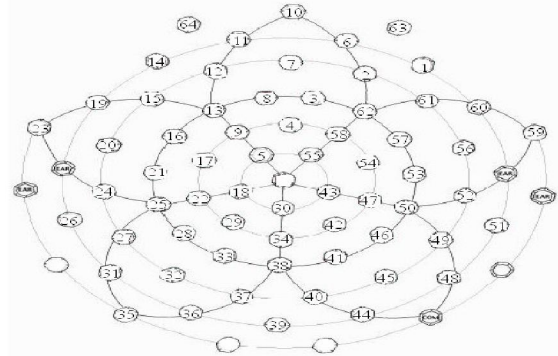


Figure 12. A standard set of EEG electrodes to collect data from a brain of an individual, used to evolve a model

After evolving models are trained on EEG channel data, rules Can be extracted in the form of: IF Channels 13 and 27 have high values, THEN the state of the brain is sleep. This type of research is reported in [11,46].

D. Multimodal Information Processing and Biometrics

Combining speech, image and other modalities in an adaptive way, where new speech samples can be added in time, new images, new modalities (e.g. fingerprints) for a person recognition, person identification and person verification is a promising area of application for eCI – Fig. 13 [11,46].

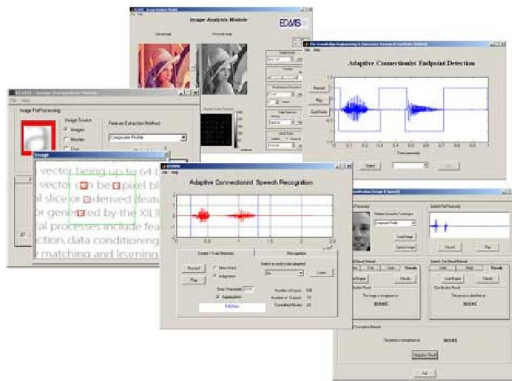


Figure 13. An example of a multimodal (speech and image) eCI

E. Financial and Business Forecast

Adaptive learning and future value prediction of financial and business time series with the use of eCI is reported in [46]. Fig.14 shows the weekly on-line prediction of the exchange rate Euro/US\$ for 1-,2-,3- and 4 weeks ahead using 3 input variables: ERate, Euro/Yen, Stock-E/US, with 4 week time lags each [46]. The lower figure shows the number of rule nodes evolved in an EFuNN structure applying also aggregation of nodes.

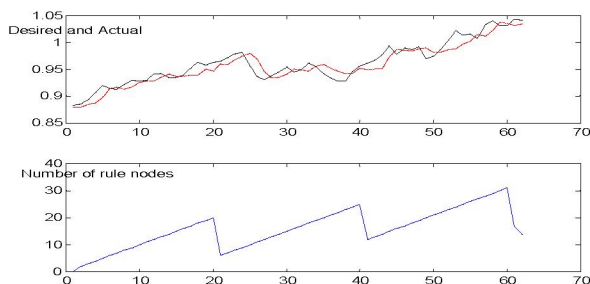


Figure 14. The weekly on-line prediction of the exchange rate Euro/US\$ and the number of the evolved and aggregated rule nodes in an EFuNN architecture.

F. Autonomous mobile robots

Evolving, autonomous learning robots, that communicate between each other, is an area of growing interest and potential for eCI. In [66] ECOS are used to control the position of robocup robots that adapt on the spot to the opponent – Fig. 15.



Figure 15. An eCI is used to locate the players on the field while adapting on-line to the opponent's strategy [66]

V. CONCLUSIONS

The concept of evolving intelligent systems has been presented as an effective tool to address the problem of modeling non-stationary, highly non-linear processes on-line in real-time. The basic elements of the concept and its procedure have been outlined without going into details, which are available in a number of recently published papers by the authors and their collaborators. In essence, the concept of evolving intelligent systems can be considered a higher level adaptation that concerns model structure as well as model parameters. It is demonstrated here through two modeling constructs that the authors have introduced recently and are continuing to develop, namely the evolving connectionist systems, ECOS, and evolving fuzzy systems, EFS.

Further research is planned in application areas as highlighted in the paper, along with some novel generic methods of eCI to be developed, such as: transductive evolving systems [61,62]; evolving spiking neural networks [63]; evolving neurogenetic models [64,65]; evolving quantum inspired neural networks [46], and others.

The true intelligent systems must evolve their structure, functionality and knowledge – they can not be fixed *a priori*.

VI. ACKNOWLEDGEMENTS

Part of this research is also supported by the FRST/NERF funded project AUTX02001 in New Zealand.

REFERENCES

- [1] Saridis, G.N.; Valavanis, K.P., Analytical design of intelligent machines. *Automatica*. Vol. 24, no. 2, pp. 123-133. 1988
- [2] Geyer-Schulz A. (1995) *Fuzzy Rule-Based Expert Systems and Genetic Machine Learning*, Studies in Fuziness, v.3, Berlin, Germany: Physica Verlag
- [3] Zadeh L.A., Fuzzy logic, neural networks, and soft computing, *Communications to the ACM*, vol.37(3), March 1994, pp.77-84, ISSN:0001-0782
- [4] Zadeh L. A. (1965) Fuzzy Sets, *Information and Control*, v.8, pp.338-353
- [5] Werbos P. (1990) Backpropagation Through Time: What it Does and How to do it, *Proc. of the IEEE Conference on Neural Networks*, v.78 (10), pp.1550-1560
- [6] Holland J. H., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975.
- [7] Goldberg D.E. (1989) *Genetic Algorithms in Search, Optimization and machine Learning*, Reading, MA, USA: Addison-Wesley
- [8] Koza J., *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, USA: MIT Press, 1992
- [9] Meystel, A. M., Albus, J. S., *Intelligent Systems: Architecture, Design, and Control*, John Wiley and Sons, 2001
- [10] Kasabov, N. "Evolving fuzzy neural networks for on-line supervised/unsupervised, knowledge-based learning," *IEEE Trans. SMC - part B, Cybernetics* 31, 902-918, 2001
- [11] Kasabov, N. *Evolving connectionist systems: Methods and applications in bioinformatics, brain study, and intelligent machines*, Springer Verlag, London, Heidelberg, NY, 2002
- [12] Angelov P., *Evolving Rule-based Models: A Tool for Design of Flexible Adaptive Systems*. Heidelberg, Germany: Springer-Verlag, 2002.
- [13] Angelov, P., D. Filev, "An approach to on-line identification of evolving Takagi-Sugeno models", *IEEE Trans. on Systems, Man and Cybernetics, part B*, vol.34, No1, pp. 484-498, 2004.
- [14] Ljung L. (1987) *System Identification: Theory for the User*, New Jersey, USA: Prentice-Hall

- [15] Wang, L.-X. "Fuzzy Systems are Universal Approximators," *Proc. FUZZ-IEEE*, San Diego, CA, USA, pp.1163-1170, 1992.
- [16] Hornik K. (1991) Approximation Capabilities of Multilayer Feedforward Network, *Neural Network*, v.4, pp.251-257.
- [17] Leng G., T.M. McGuinty, G. Prasad, "An approach for on-line extraction of fuzzy rules using a self-organizing fuzzy neural network," *Fuzzy Sets and Systems*, vol. 150 (2), pp.211-243, 2005.
- [18] Juang C.-F., X.-T. Lin "A recurrent self-organizing neural fuzzy inference network," *IEEE Trans. on NN*, vol. 10, pp. 828-845, 1999
- [19] Lin, F.-J., C.-H. Lin, P.-H. Shen, "Self-constructing fuzzy neural network speed controller for permanent-magnet synchronous motor drives," *IEEE Trans. on Fuzzy Systems*, Vol.9 (5), pp. 751-759, 2001.
- [20] Angelov P., C. Xydeas, Fuzzy Systems Design: Direct and Indirect Approaches, *Soft Computing*, vol. 10 (9), pp.836-849, July 2006, special issue on New Trends in the Fuzzy Modelling part I: Novel Approaches.
- [21] Kim K., J. Baek, E. Kim, M. Park, "TSK Fuzzy model based on-line identification," *Proc. 11th IFSA World Congress*, Beijing, China, 2005, pp.1435-1439.
- [22] Angelov, P., D. Filev, "Simpl. eTS: A Simplified Method for Learning Evolving Takagi-Sugeno Fuzzy Models," *The 2005 IEEE Intern. Conf. on Fuzzy Systems FUZZ-IEEE*, Reno, NE, USA, 2005, pp.1068-1073.
- [23] Angelov P., C. Xydeas, D. Filev, *On-line Identification of MIMO Evolving Takagi-Sugeno Fuzzy Models*, Intern. Joint Conf. on NN and Intern. Conf. on Fuzzy Systems, IJCNN-FUZZ-IEEE, Budapest, Hungary, 25-29 July, 2004, 55-60, ISBN 0-7803-8354-0.
- [24] Huang G.-B., P. Saratchandran, N. Sundarajan, "A generalized growing and pruning RBF (GGAP-RBF) neural network for function approximation," *IEEE Trans. on NN*, vol.16 (1) 57-67, 2005.
- [25] Kasabov N., Q. Song "DENFIS: Dynamic Evolving Neural-Fuzzy Inference System and Its Application for Time-Series Prediction," *IEEE Trans. on Fuzzy Systems*, Vol.10 (2), pp. 144-154, 2002.
- [26] Fayyad, U.M., G. Piatetsky-Shapiro, P. Smyth, *From Data Mining to Knowledge Discovery: An Overview*, *Advances in Knowledge Discovery and Data Mining*, MIT Press, 1996.
- [27] Kalman R. E., A New Approach to linear filtering and prediction problem, Transactions of the ASME, Ser. D, Journal of Basic Engineering, vol. 82, pp.34-45, 1960.
- [28] Crisan D., Particle Filters – A Theoretical perspective, Sequential Monte Carlo Methods in Practice, A Doucet, J.F.G., de Freitas, N. J. Gordon, Eds., Berlin: Springer Verlag, 2001.
- [29] Yager R., D. Filev, *Essentials of Fuzzy Modeling and Control*, New York, USA: John Wiley and Sons, 1994.
- [30] Takagi, T., M. Sugeno, Fuzzy identification of systems and its application to modeling and control, *IEEE Trans. on Syst., Man & Cybernetics*, vol. 15, pp. 116-132, 1985.
- [31] Filev, D., Tardiff, J., Fuzzy Modeling within the Statistical Process Control Framework, Proc. 2004 IEEE Int. Conference on Fuzzy Systems, Budapest, Hungary, July, 2004, 66-71 (Invited Paper)
- [32] Angelov, P., D. Filev, N. Kasabov, O. Cordon (Eds.) Evolving Fuzzy Systems, Proc. 2nd Intern. Symposium on Evolving Fuzzy Systems, 7-9 Sept. 2006, Ambleside, Lake District, UK, IEEE, 1-350pp., ISBN, 0-7803-9719-3, to appear
- [33] Yager R., "A model of participatory learning," *IEEE Transactions on Systems, Man, and Cybernetics*, 20 (5) 1229-1234, 1990.
- [34] Filev, D., Larsson, T., Intelligent adaptive control using multiple models, Proc. 2001 IEEE Joint Int. Conf. on Control Applications & Int. Symp. on Intelligent Control, CCA/ISIC'01, September, 2001, Mexico City, 314-319
- [35] Angelov, P., X.-W. Zhou Evolving Fuzzy Systems from Data Streams in Real-Time, In: Proc. 2nd Intern. Symposium on Evolving Fuzzy Systems (Angelov P., D. Filev, N. Kasabov, O. Cordon Eds.), 7-9 Sept. 2006, Ambleside, Lake District, UK, IEEE, 1-350pp., ISBN, 0-7803-9719-3, to appear
- [36] Kohn, K.W., Dimitrov, D.S. Mathematical Models of Cell Cycles. *Computer Modeling and Simulation of Complex Biological Syst.* (1999).
- [37] Kattan, M., Leung, D., Brennan, M. Postoperative Nomogram for 12-Year Sarcoma-Specific Death. *J. of Clinical Oncology* vol. 20, pp. 791-796 2002.
- [38] Levey, A. S. et al. A more Accurate Method to Estimate Glomerular Filtration Rate from Serum Creatinine: A new Prediction Equation. *Annals of Internal Medicine* 130, 461-470 (1999).
- [39] Anderson, K.M., Odell, P.M., Wilson, P.W.F., Kannel, W. B. Cardiovascular disease risk profiles. *American Heart Journal* 121, 293-298 (1991).
- [40] Shipp, M. et al. Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nature Medicine* 8, 68-74 (2002).
- [41] van Veer, L., Dai, J.H., Vijver, M.J.v., He, Y.D. et al. Gene expression pro-filing predicts clinical outcome of breast cancer. *Nature* 415, 530 (2002).
- [42] de Jong, H. Modeling and simulation of genetic regulatory systems: a literature review. *Journal of Computational Biology* 9, 67-102 (2002).
- [43] Astrom K., B. Wittenmark, Adaptive Systems, Addison-Wesley, USA (1988).
- [44] Lufhofer, E., Klement, E.P. Lujan, J.M. Guardiola, C., Model-based fault detection in multi-sensor measurement systems, Intelligent Systems, Proc. 2nd Intern. IEEE Conference, 22-24 June 2004, Vol.1, pp. 184-189.
- [45] Angelov, P., M. Everett, EvoMap: On-Chip Implementation of Intelligent Information Modelling using EVolving MAPping, 2005, Lancaster University, Lancaster, UK, pp.1-15.
- [46] Kasabov, N. *Evolving connectionist systems: Brain-, gene-, and , quantum inspired computational intelligence*, Springer Verlag, London, Heidelberg, NY, 2006
- [47] Kasabov, N. Adaptation and Interaction in Dynamical Systems: Modelling and Rule Discovery Through Evolving Connectionist Systems, *Applied Soft Computing*, 2006, Volume 6, Issue 3, pages 307-322
- [48] Song, Q., N. Kasabov, T. Ma, M. Marshall, Integrating regression formulas and kernel functions into locally adaptive knowledge-based neural networks: a case study on renal function evaluation, *Artificial Intelligence in Medicine*, 2006.
- [49] Pang, S., S. Ozawa and N. Kasabov, Incremental Linear Discriminant Analysis for Classification of Data Streams, *IEEE Trans. SMC-B*, vol. 35, No. 5, 2005, 905 – 914
- [50] Chan, Z., N.Kasabov, A Two-Stage Methodology for Gene Regulatory Network Extraction from Time-Course Gene Expression Data, *Expert Systems with Applications: An International Journal* (ISSN: 0957-4174), Special issue on Intelligent Bioinformatics Systems, December, 2005.
- [51] Marshall, M.R., Q. Song, T.M. Ma, S. MacDonell, N.Kasabov, Evolving Connectionist System versus Algebraic Formulae for Prediction of Renal Function from Serum Creatinine, *Kidney International*, vol. 67 (2005), pp. 1944 – 1954
- [52] Kasabov, N. Knowledge based neural networks for gene expression data analysis, modelling and profile discovery, *Drug Discovery Today: BIOSILICO*, vol. 2, No. 6, November 2004, pp. 253-261.
- [53] Kasabov, N., Z. S.H. Chan, Igor Sidorov and Dimiter Dimitrov, Gene Regulatory Network Discovery for Time Series Gene Expression Data – A Computational Intelligence Approach, *Lecture Notes in Computer Science*, Vol.3316, 2004, Springer Verlag, 1344-1353.
- [54] Ozawa, S., S. Pang and N. Kasabov, A Modified Incremental Principal Component Analysis for On-Line Learning of Feature Space and Classifier, *Lecture Notes in Artificial Intelligence LNAI*, Volume 3157, Springer-Verlag, Berlin, Heidelberg, 2004, 231 – 240.
- [55] Chan Z. and N.Kasabov, Evolutionary computation for on-line and off-line parameter tuning of evolving fuzzy neural networks, *Int. J. of Computational Intelligence and Applications*, Imperial College Press, vol. 4, N.3, September 2004, 309-319
- [56] Futschik, M., A.Reeve, and Kasabov, N. Evolving connectionist systems for knowledge discovery from gene expression data of cancer tissue, *Artificial Intelligence in Medicine*, 28 (2003) 165-189
- [57] Kasabov, N., Zeke Chan, Vishal Jain, Igor Sidorov and Dimiter Dimitrov, Computational Modelling of Gene Regulatory Networks, Chapter 8, in: Bajic., V and Tan Tin Wee (eds), *Information Processing and Living Systems*, Imperial College Press, Singapore, 2005, 673-686.
- [58] Kasabov, N., Z.Chan, Q.Song and D.Greer, Evolving neuro-fuzzy systems with evolutionary parameter self-optimisation, chapter in: *Do Adaptive Smart Systems exist?* Springer Verlag, Series Study in Fuzziness, vol.173, 2005

- [59] Dimitrov, D. S., Igor A. Sidorov and Nikola Kasabov Computational Biology, in: M. Rieth and W. Sommers (eds) *Handbook of Theoretical and Computational Nanotechnology*, Vol. 1 (1) American Scientific Publisher, Chapter 21, 2004
- [60] Kasabov, N. and D. Dimitrov, Evolving connectionist systems for gene regulatory network modelling and discovery, chapter in: L. Wang and Rajapakse (eds) *Advances in neuro-information processing*, Springer Verlag, 2004
- [61] Song Q., and N. Kasabov, TWNFI: Transductive Weighted Neuro-Fuzzy Inference Method for Personalised Modelling, *Neural Networks*, 2006
- [62] Song Q. and N. Kasabov, TNFI: Transductive neuro-fuzzy inference method for personalized modelling, *IEEE Transactions on Fuzzy Systems*, vol.13, No.6, December, 2005
- [63] Kasabov, N., L. Benuskova L and Wysoski SG (2005) Computational neurogenetic modeling: integration of spiking neural networks, gene networks, and signal processing techniques. In: ICANN 2005, LNCS 3697, W. Duch et al (Eds), Springer-Verlag, Berlin Heidelberg, pp. 509-514
- [64] Kasabov, N., and L. Benuskova, Computational Neurogenetics, *International Journal of Theoretical and Computational Nanoscience*, Vol. 1 (1) American Scientific Publisher, 2004, 47-61.
- [65] Benuskova L., Wysoski S.G., Kasabov, N. (2006) Computational neurogenetic modeling: integration of spiking neural networks and gene networks. Proc. IJCNN, IEEE Press, 2006
- [66] Huang, L., Song, Q., Kasabov, N., Evolving Connectionist Systems Based Role Allocation of Robots for Soccer Playing, Joint 2005 International Symposium on Intelligent Control & 13th Mediterranean Conference on Control and Automation (2005 ISIC-MED), June 27-29, 2005, Limassol, Cyprus
- [67] Rajbman, N.S., Extensions to nonlinear and minimax approaches. In: Trends and Progress in System Identification, Eykhoff, P. (Ed.), Pergamon Press: Oxford, 169-213, 1981
- [68] Narendra, K. and Xiang, C., "Adaptive control of discrete-time systems using multiple models", *IEEE Trans. Automat. Contr.*, Vol. 45, 1669-1685, 2000
- [69] Chen, L. and Narendra, K. S., Nonlinear adaptive control using neural networks and multiple models, *Automatica*, 2001.
- [70] Narendra, K.S., Balakrishnan, J., and Ciliz, M.K., Adaptation and learning using multiple models, switching, and tuning, *IEEE Control Systems*, 37-50, 1995
- [71] Filev, D., Identification of fuzzy relational models, Proc. of the IV IFSA Congress, Brussels, p.82-85, 1991
- [72] Yager R.R., Filev, D., Generation of fuzzy rules by mountain clustering," *Journal of Intelligent and Fuzzy Systems* 2, 209-219, 1994
- [73] Chiu, S. L., Fuzzy model identification based on cluster estimation, *Journal of Intelligent and Fuzzy Syst.* vol.2, pp. 267-278, 1994.
- [74] Angelov, P., Victor, J., Dourado, A., Filev, D., On-line evolution of Takagi-Sugeno Fuzzy Models", Proc. Of the 2nd IFAC Workshop on Advanced Fuzzy/Neural Control, 16-17 September 2004, Oulu, Finland, pp.67-72.