NEURO-FUZZY SYSTEM FOR ENHANCED FAULT DIAGNOSIS IN INDUSTRIAL FACILITY

Konstantin D. DIMITROV
Technical University - Sofia, Bulgaria

Abstract. The present paper describes the development of a modular neuro-fuzzy system, designated for enhanced and flexible fault diagnosis. Some specific neural algorithms for identification, recognition, evaluation and classification of the process parametric values are created and applied in the systems structure. The so-developed neuro-fuzzy system is then applied for fault diagnosis in an industrial zinc galvanizing facility, i.e., under real operational conditions.

Keywords: process fault diagnosis, neuro-fuzzy system, neural networks, fuzzy models, pattern recognition

1. Introduction

In the recent years, considerable efforts of engineers and scientists were focused on a creation and a development of fuzzy and neuro-fuzzy system, designated for systems and/or process fault diagnosis, for adaptive control and supervision, as well as for modeling and prognosis of systems and/or process behavior [2, 8, 9, 10].

In general, such types of so advanced and highly-performing systems find their application mainly in the area of evaluation and real-time control of the operational reliability in industrial complexes, for optimal and adaptive control of industrial processes and for on-line fault diagnosis with continuous and real-time evaluation of the systems states [2, 3, 4, 7]. Some sophisticated fuzzy modeling approaches were developed for the purpose and respectively applied during the creation of specific fuzzy models, thus providing an alternative but also extremely flexible modeling tools (compared to the traditional mathematical methods) [9, 10, 11]. The fuzzy modeling structures could be developed mainly by the application of two general types of methodologies.

The first type of the fuzzy modeling methodologies is based on a creation of fuzzy relational matrixes (constructed respectively by fuzzy equations), that are supposed to relate the systems/process input and output variables in a fuzzy manner [5, 7, 8].

The information utilized for the creation of the fuzzy relational models (i.e., matrixes) could be either the measured numerical data (under real operation conditions), either the simulated data bases (obtained during experiments, carried on a specific laboratory equipment) [4, 6, 8]. The so-created fuzzy models possess good generalization capacities and the relation equations are also quite adequate. The issue here is, that, the created relational models are designated mainly for systems with two (up to three) fuzzy inputs, i.e., they are not quite appropriate for multivariable systems [5, 7]. The so-developed matrixes (based on the available numerical data) are in general high-dimensional, complicated for computation and relatively complex [5, 8]. This is due mainly to the fact, that, the fuzzy models must adapt the numerical data bases, (which in fact "carry" the modeling knowledge and where the created fuzzy models must work), to some specific environment (designed formally via some general linguistic methods and represented by membership grade calculations (i.e., functions) [5, 7, 8].

The second type of the fuzzy modeling methodologies is based on sets of fuzzy rules, that must relate the systems/process local input-output relations, (mostly in a linear and numerical way) [9, 10, 11].

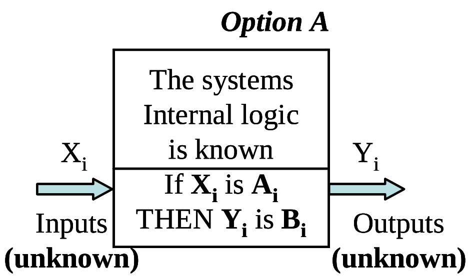
The so-developed fuzzy models are in general quite simple, but at the same time also flexible and powerful enough to treat multivariable systems (at numerical level) [9, 10, 11]. The issue here, that, the identification algorithms are in general rather complicated and usually involve linguistic approximation techniques, non-linear programming and least-square optimization [10, 11]. Recently, Neural Networks learning and generalization capacities are also applied into the development of the fuzzy algorithms of identification [2, 3, 12].

The present paper describes the development of a modular neuro-fuzzy system, designated for fault diagnosis. Some specific algorithms for identification, recognition and classification of the process parametric values are created and applied in the systems structure. The so-developed neuro-fuzzy system is then applied for an enhanced fault diagnosis (FD) in an industrial zinc galvanizing facility, i.e., under real operational conditions.

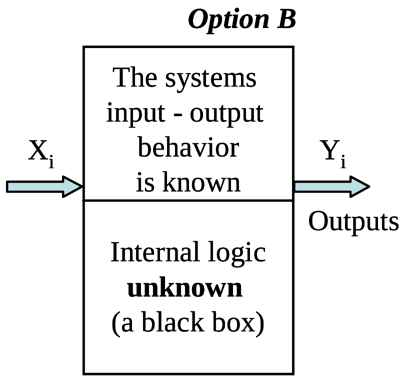
2. Development of the neuro-fuzzy system, applied for an enhanced FD – creation of the fuzzy models, the neural networks structures and identification learning algorithms

2.1. General characteristics and relationships of the modular FD neuro-fuzzy system

Depending from the type and the characteristics of the diagnosed process (and/or system), ***two general options*** should be considered during the creation of the FD systems structures, figure 1.



a)



b)

Figure 1. Options for development and application of Fuzzy and Neural modules
a) *Option A* – systems internal logic is known;
b) *Option B* – systems input-output behavior is known

The systems structure, shown at figure 1a, represents the option, when the systems (process) logic is known, but there is no preliminary knowledge (i.e., examples) for its input-output behavior. For this particular case (referred as *"Option A"*) it should be more convenient to develop a fuzzy reasoning model as a set of ***IF/THEN fuzzy rules***. Such a model should be able to describe (and even to make a prognosis) of the systems behavior.

For the systems structure, shown at figure 1b, (and referred as *"Option B"*) it is more convenient to use the available input-output data to train a Neural Network (NN), and thus – to model the internal behavior of the system, i.e., since the systems internal logic is not, this particular case could be considered as a *"Black box"* FD system.

These two (rather idealized) general structures possess their advantages, but they also have their specific issues [4, 5, 12]. So, our goal (for this particular case), is to create a "hybrid" system, developed as a combination of these general structures. Such a system shall possess a ***neuro-fuzzy structure*** and is expected to be more flexible and adaptable to the operation challenges.

All logistic and technologic structures, as well as all essential technologic and logistics processes of the developed hot dip zinc galvanizing facility are presented in details in [1]. The general structure of the created "hybrid" Neuro-fuzzy System, designated for enhanced FD in a zinc galvanizing facility is presented at figure 2.

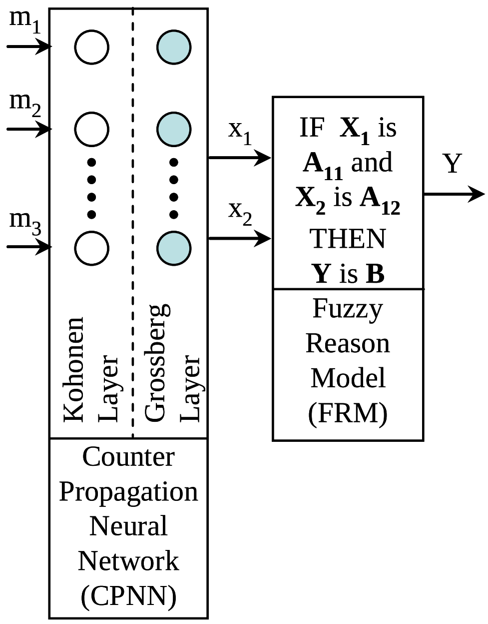


Figure 2. General structure of a Hybrid, Neuro-Fuzzy System (a "Grey Box")

The created structure is developed as a "Grey Box" and can perform an adequate modelling in a timely, reliable and cost-saving manner. The so-developed neuro-fuzzy system is extremely adaptable and flexible and is therefore able to perform FD over the physically damaged ***systems components*** (i.e., components of logistics and/or technological structures), but also over the behaviour of the systems ***main technologic processes*** (i.e., ***the gas furnace heating processes, the tank heating processes, the waste liquids treatment and neutralization process,*** etc., etc.).

The Neural Network (NN) module is trained to receive three data flows as an inputs (measurements, effectuated over the electrical, electronic and mechanical components), which are mapped (i.e., processed via the NN learning and recognition algorithms) into two flows of numerical values, which serve as inputs to the fuzzy reasoning model (i.e., the fuzzy algorithms).

In fact, the output data, generated by the NN indicate (i.e., they are features) *the degree of the failure/fault occurrence in the systems components and/or process variables* (the generated outputs are numbers between 0 and 1), as well as the *signal to noise ratio*, reflecting the disturbances (the generated output are numbers between 0 and 20). The so-generated neural outputs (features) are fed as inputs to the fuzzy module, where the fuzzy relations map (process) the submitted information and provide *decisions (results)* about the *actual state(s) of systems components and process variables (under consideration).* Therefore, the *purpose for the creation of a flexibility "Hybrid" Neuro-Fuzzy structure is based of the following facts:*

- The sets of numerical measurements provide in general too much details (some noise and disturbances are almost always presented in the numerical data flows), and could not be really effective for on-line processing of the FD features;
- The NN filtering, smoothing and mapping of the processed numerical measurement flows into some kind of *feature spaces* (i.e., *a degree of failure/fault occurrence, a signal-to-noise ratio,* etc.) could be more reliable with a combination of a fuzzy controller (module).

Thus, the so-created Neuro-fuzzy hybrid structure shall be flexible and able to provide an adaptive and enhanced FD in a real technologic facility for hot dip zinc galvanizing.

In cases, when the input data flows contain many various and very complex kinds of species - then another option in the arrangement of the hybrid systems structure can be created. In such kinds of systems (named "fuzzy-neural systems"), the fuzzy processing of the data flows ***precedes*** the neural computing. Such hybrid fuzzy-neural system first provides *a classification of all particular groups of input species,* (performed by the fuzzy algorithms of the fuzzy module), and after that – *a discrimination of the individual species in each classified group* (performed by the NN algorithms).

Two general types of NN structures are suitable for the development of the actual hybrid Neuro-fuzzy system for FD – the Back-propagation NN (BPNN) and the Counter-propagation NN (CPNN).

The BPNN is well-known structure, which possesses good generalization capacities, and is able to represent various fuzzy processes functionally or structurally. The issue here is that, the self-construction of such a system is difficult. This is due to the fact, that, the gradient descent learning algorithm, (adopted by the BPNN) is a relatively slow process, which requires continuously repeated presence of the training samples (species), in order to achieve the necessary convergence of the learning. These requirements in fact lead to difficulties in the self-organization (i.e., the self-construction) of the system from the available data (measured, generated and simulated).

The CPNN is a two-layer network, which is able to perform vector-to-vector mapping process (similar to the hetero-associative memory NN). The advantage of the CPNN is that, it can be trained to perform associative mapping much faster, but with adequate self-organization of its structure (compared to the BPNN).

The CPNN is very useful in the pattern mapping and association as well as in data compression and classification – some real merits, when FD must be performed over complicated systems and/or processes (as the ones, developed in a hot dip zinc galvanizing facility).

2.2. Development of the Fuzzy reasoning model – structure and internal relations

It could be assumed, that, the system to be modeled possesses $\boldsymbol{r}$ inputs and $\boldsymbol{n}$ outputs, denoted respectively by $\boldsymbol{m_1, m_2, \dots, m_r}$ and $\boldsymbol{y_1, y_2, \dots, y_n}$. By considering the input output *values* at different time instants, as different *variables*, the systems dynamics could be expressed by the following relation,

$$\boldsymbol{Y_k = F_k(x_1, x_2, \dots, x_q)} \qquad (1)$$

where $\boldsymbol{q}$ corresponds to $\boldsymbol{q}$-th variable.

The *fuzzy reasoning model* (that must be developed) should be able to describe the systems behavior adequately. Since the systems input-output relationships are not known, they could be modeled at ***linguistic level*** by sets of IF-THEN rules (please see also figure 2), of the following form,

Rule $^{(j)}$:

IF $\boldsymbol{X_1}$ **is** $\boldsymbol{A_{1,j}}$ **and … and** $\boldsymbol{X_q}$ **is** $\boldsymbol{A_{q,j}}$ **THEN** $\boldsymbol{Y_1}$ **is** $\boldsymbol{B_{1,j}}$ **and … and** $\boldsymbol{Y_m}$ **is** $\boldsymbol{B_{m,j}}$ (2)

where $\boldsymbol{X_i}$ and $\boldsymbol{Y_k}$ are the fuzzy systems variables (corresponding to $\boldsymbol{x_i}$ and $\boldsymbol{y_k}$ respectively), and $\boldsymbol{A_{q,j}}$

and $B_{m,j}$ represent fuzzy sets (defined in corresponding clusters of measurement/modeled space). The $A_{q,j}$ and $B_{m,j}$ sets (for this fuzzy model) are characterized by *membership functions* (selected symmetrical for this particular case), which are respectively expressed via two functional parameters – $E^{(x)}_{q,j}$ and $h^{(x)}_{q,j}$ (respectively $E^{(y)}_{m,j}$ and $h^{(y)}_{m,j}$), where $E^{(x)}_{q,j}$, ($E^{(y)}_{m,j}$) is the center of the $A_{q,j}$ ***set*** ($B_{m,j}$ ***set***), and $h^{(x)}_{q,j}$ ($h^{(y)}_{m,j}$) is the half width of the $A_{q,j}$ and $B_{m,j}$ sets. For this particular case the membership functions are selected to be symmetrical, because the diagnosed zing galvanizing processes possess symmetrical characteristic development – please see [1] for details on systems logistics and technologic processes.

The ***Rule*** $^{(j)}$ might then be restructured under the following form,

Rule $^{(j)}$:

$$\textbf{IF } [E^{(x)}_j, H^{(x)}_j] \textbf{ THEN } [E^{(y)}_j, H^{(y)}_j] \tag{3}$$

where $E^{(x)}_j = \{E^{(x)}_{1,j}, \ldots, E^{(x)}_{p,j}\}$ and $E^{(y)}_j = \{E^{(y)}_{1,j}, \ldots, E^{(x)}_{r,j}\}$ are the two vectors of the sets centers (i.e., "central" vectors), which are associated with the two vectors of sets widths (i.e., "width" vectors), respectively, $H^{(x)}_j = \{h^{(x)}_{1.,j}, \ldots, h^{(x)}_{p,j}\}$ and $H^{(y)}_j = \{h^{(y)}_{1.,j}, \ldots, h^{(y)}_{p,j}\}$.

Each couple of the kind $[E^{(x)}_j, H^{(x)}_j]$ is able to create an ***input rule pattern***. Each real (i.e., the current) *numerical output* of the system y^R_k, is created as a *response* to the current numerical inputs $x^R = [x^R_1, x^R_2, \ldots, x^R_q]$, and can respectively be determined over the following two stages:

- ***Stage A:*** *Pattern Recognition (PR)*;
- ***Stage B***: *Weights Evaluation (WE)*, i.e., criteria for adequateness.

The *Degree of recognition* $R_D^{(j)}$, between the ***input x*** and the developed (via ***Rule*** $^{(j)}$) ***pattern*** of the kind $[E^{(x)}_j, H^{(x)}_j]$, is determined by the following relation,

$$R_D^{(j)} = 1 - D^{(j)}_R \{\mathbf{x}, [E^{(x)}_j, H^{(x)}_j]\} \tag{4}$$

where $D^{(j)}_R$ is the ***relative distance*** between x and $[E^{(x)}_j, H^{(x)}_j]$.

The relative distance $D^{(j)}_R$ is defined by the relations,

$$D^{(j)}_R = \frac{DM^{(j)}}{h}, \text{ if } DM^{(j)} \le h, \quad \text{or} \qquad D^{(j)}_R = 1 \quad \text{otherwise} \tag{5}$$

where

h is the width of all $A_{q,j}$ fuzzy sets, (which is selected to be identical for all sets, due to the symmetrical properties of the technologic cycles), and

$DM^{(j)}$ is the metric distance, which can be defined and calculated via three main types of criteria of the following kind,

- ***Euclidian distance:***

$$DM^{(j)}_{EU} = \left[\sum_{i=1}^{p}\left(E^{(x)}_{j,i} - x_i\right)\right]^{1/2} \tag{6}$$

- ***Maximal distance:***

$$DM^{(j)}_{MAX} = \max_{1 \le i \le p}\left|E^{(x)}_{j,i} - x_i\right| \tag{7}$$

- ***Hamming distance:***

$$DM^{(j)}_{HAM} = \sum_{i=1}^{p}\left|E^{(x)}_{j,i} - x_i\right| \tag{8}$$

It should be noted, also that, $R_D^{(j)} \in [0, 1]$, and also $D^{(j)}_R \in [0, 1]$.

Once the ***degrees of recognition*** $R_D^{(j)}$ are determined, i.e., the procedures of PR stage are completed, then, ***the real (the current) outputs*** y^R_k ***of the system*** can be deduced, during the WE stage, which is developed as follows:

$$y^R_k = \frac{\sum_{j=1}^{N} R_D^{(j)} E_j^{(y)}}{\sum_{j=1}^{N} R_D^{(j)}}. \tag{9}$$

It must be noted also, that, the "width" vector $H^{(y)}_j$, which is associated with the THEN part of ***Rule*** $^{(j)}$, should not be included in the WE stage, since the membership functions (for this particular case) are symmetrical and have identical width.

Thus, the ***Rule*** $^{(j)}$, developed in (3), could be transformed as follows,

$$\textbf{Rule}^{(j)}_T\textbf{: IF } [E^{(x)}_j, H^{(x)}_j] \textbf{ THEN } E^{(y)}_j \tag{10}$$

Thus, *the development of a Fuzzy Reasoning Model (FRM) is reduced (for this particular case) to the creation of Rule Bases from the type, presented in (10). The only exigency is, that, the input model variables must be preliminary specified.*

2.3. Development of the CPNN – structure and identification algorithms

The structure of the CPNN (please see figure 2) consists of an input layer, a hidden layer, named Kohonen layer (with Q and P cells) and an output layer named a Grossberg layer (with N cells). The

CPNN is designed to approximate any kind of continuous function $\boldsymbol{F}$, defined in compact set $\{R\}$, *via specific sets of samples* (X^S, Y^S), where the X^S vectors are randomly selected (drawn).

Once the CPNN structure is determined, then, two main types of neural Algorithms, must be designed and applied during the FD, performed over the modules of the hot dip zinc galvanizing system – the ***Forward Algorithm (FORCPNN)*** and the ***Training Algorithm (TRACPNN).***

The ***FORCPNN***-algorithm must be able to compute every k^{th} output of the neural system, that corresponds to some particular kind of input $\boldsymbol{x}$, at the time instant (randomly selected).

The algorithmic structure is designed as follows:

Stage A^{FOR}. Determination of the *winning cell* "$\boldsymbol{K}$" in the Kohonen layer of the CPNN via competitive rules – accordingly to the distances D^K in the weight vector $\omega^k(t)$, and with respect to the current input $\boldsymbol{x}$, i.e.,

$$D^k\left[\omega^K(t),\ x\right]=\min_{k=1,P} D^K\left[\omega^k(t), x\right]= \\ =\min_{k=1,P}\left\|\omega^k(t)-x\right\| \tag{11}$$

where $\|\omega^k(t) - x\|$ is the Euclidian metric distance.

Stage B^{FOR}. Computation of the outputs $y^k_{KOH}(t) \in [0, 1]$ of the Kohonen layer by the *winning-cell competition rule*, i.e.,

$$y^k_{KOH}(t) = 1,\ \text{if}\ k = K,\quad \text{and} \\ y^k_{KOH}(t) = 0,\ \text{otherwise} \tag{12}$$

Stage C^{FOR}. Computation of the outputs $y^k_{GR}(t)$ of the Grossberg layer by,

$$y^k{}_{GR}(t)=\sum_{k=1}^{N} y^k_{KOH}(t)\cdot\mu_j^k(t)=\mu_j^K(t) \tag{13}$$

where μ_j^k is the weight, connecting the k^{th} cell of the Kohonen layer to the j^{th} cell of the Grossberg layer.

The so-developed algorithms can provide a partially self-organization in the CPNN structure. Thus, the neural structure is capable to map a R^Q-set into a R^M-set, as a result of its training (via specific sets of training examples).

The particular *knowledge*, acquired via the CPNN training process, is entirely represented by the *associated* ***weights*** ω^k and μ^k.

Since the CPNN could be considered as a "Hard" module of the Hybrid Neuro-fuzzy system, (linked with the "Soft" Fuzzy module), the weight vectors could also be considered as "stable" (since they emanate and connect the Kohonen and the Grossberg layer of the neural structure). The systems rule base can respectively be expressed as follows,

$$\textbf{IF } \omega^k \quad \textbf{THEN } \mu^k \tag{14}$$

where, $\mu^k = (\mu^k_1, \ldots, \mu^k_m)$, and the effects of the $H_{(j)}$ are ignored for this particular case (please see the above explanations). This practically means, that, each ***Rule***$^{(j)}$ from the form (10), could be represented via the CPNN-rule (based on the properties of the Kohonen layer), developed under the form (14).

The ***TRACPNN***-algorithm of the hybrid Neuro-fuzzy system will be developed as a *self-organized semi-supervised training process*, which must be associated (i.e., must react as a response), to the sets of specific couples of training samples (X^S, Y^S).

The ***TRACPNN***-algorithm consists of two modules (algorithmic loops) – a *Kohonen module*, which is developed as ***an unsupervised process*** (and designed for training of the ω^k - weights), and a *Grossberg module*, which is a truly ***supervised process*** (and respectively designed for training of the μ^k - weights).

Since the number of the Kohonen cells must be selected in advance (but also must remain fixed during the training), the stages of the Kohonen algorithmic loop are developed as follows:

Stage A^{TRA}_{KOH}. Determination of the *winning cell* "$\boldsymbol{K}$" in the Kohonen layer via (10). The calculation must be developed with respect to the current systems inputs X^S, presented as samples (patterns).

Stage B^{TRA}_{KOH}. Computation of the outputs $y^k_{KOH}(t)$ of the Kohonen layer via (11), always with respect to the input samples.

Stage C^{TRA}_{KOH}. Adaptation of the Kohonen weights ω^k, as follows,

$$\omega^k(t)=\omega^k(t-1)+ \\ +\theta_t\left[X^S(t)-\omega^k(t-1)\right]\cdot y^k{}_{KOH}(t) \tag{15}$$

where $0 \le \theta_t \le 1$ represents the gain, which decreases monotonically with time (a harmonic series for $\theta_t = 1/t$, which satisfies the convergence issue is selected and applied in this study).

Once the $\omega^k(t)$ are stabilized, (i.e., the adaptation process is completed), then the learning procedure could also begin into the Grossberg layer.

Stage D^{TRA}_{GR}. The *desired output* Y^S for each stabilized weight vector $\omega^k(t)$ could now be generated via an *adjustment of the weights*, connecting the Kohonen cells and the Grossberg sells, by the following relation, i.e.,

$$\mu_j^k(t) = \mu_j^k(t-1) + \\ + \alpha\left[Y_j^S - \mu_j^k(t-1)\right] \cdot y^k{}_{KOH}(t) \qquad (16)$$

where μ^k_j is the weight, connecting the k^{th} cell from the Kohonen layer to the j^{th} cell of the Grossberg layer;
$\alpha \in [0, 1]$, is a constant update rate;
Y^S_j is the jth component of the training sample (pattern) Y^S.

3. Application of the developed hybrid neuro-fuzzy system for an enhanced FD in an industrial Hot dip zinc galvanizing system – i.e., under real operation conditions

The created hybrid Neuro-fuzzy system was applied for enhanced Fault Diagnosis (FD) in an industrial Hot dip zinc system.

The systems module, subjected to FD procedures, was the ***gas fired galvanizing furnace***. All details, related to the systems processes and components are presented in [1].

The measurements were effectuated under real operation conditions and were in total ***184 measured pair data***.

The measured data samples represented respectively the *gas flow rate* (i.e., the systems inputs), and *the concentration of carbon dioxide* (CO_2) in the outlet gas flow of the furnace (i.e., the systems outputs). The fuzzy rules (in the fuzzy reasoning module) were created via the *Forward Algorithm (FORCPNN)* and the *Training Algorithm (TRACPNN).* The performances of the resulting *fuzzy model* were tested via a replacement of the competitive algorithm (13) with the recognition algorithm (9).

The *Kohonen layer* of the created CPNN consisted of *22 Kohonen cells*, (generating respectively 22 fuzzy rules).

The *width of the fuzzy sets* was selected to be ***h* = 1.0**, and *the training* (the learning) *rate* was determined to be **α = 0.5**. The performance index, of the system resulting from the utilization of the 22 fuzzy rules was 0.2798.

The current (the real) and the modeled results, obtained during the application of the created hybrid neuro-fuzzy system for FD of the gas fired galvanizing furnace are presented at figure 3.

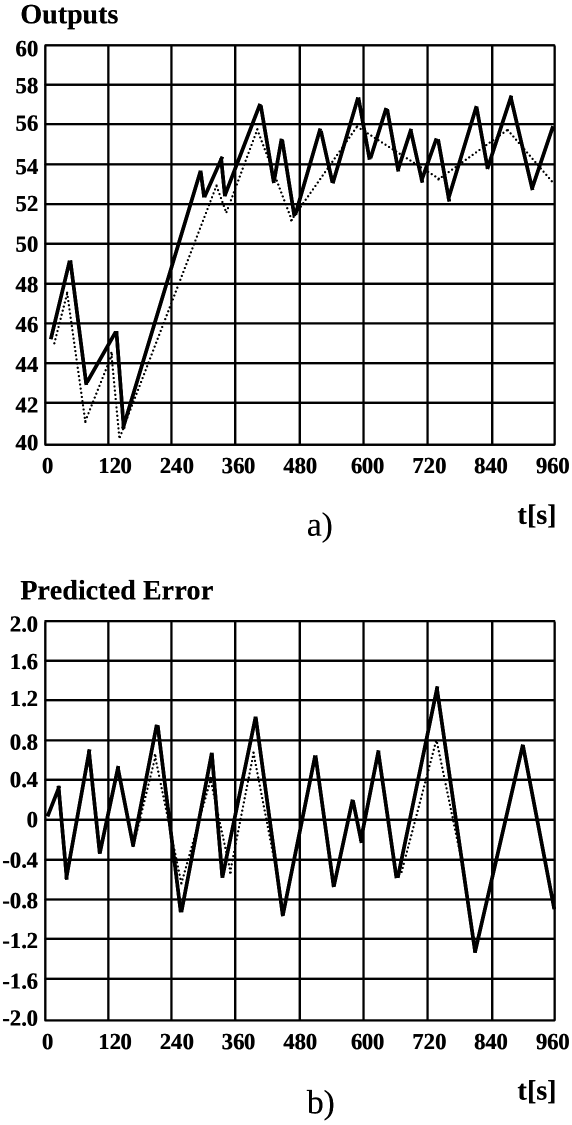


Figure 3. Desired (solid line) and modeled (dashed line) FD decisions, generated under real operation of a gas furnace
a) outputs of the system; b) predicted error

The diagrams, presented at figure 3a show the desired systems outputs (presented by a solid line), and the modeled outputs (presented by a dashed line).

Respectively the diagrams, presented at figure 3b, show the corresponding predicted error, generated during the execution of the neural algorithms.

4. Conclusions

4.1. A hybrid neuro-fuzzy system, designated for an enhanced FD in a complex industrial facility for zinc galvanizing was developed in this study – all necessary modular structures, the characteristics and the relationships in the fuzzy reasoning models, as well as and in the CPNN were created and applied for the purpose.

4.2. Once the design of the fuzzy and neural modules was completed – then two main types of neural Algorithms, were developed – the *Forward Algorithm (FORCPNN)* and the *Training Algorithm (TRACPNN).*

4.3. The so-developed hybrid neuro-fuzzy system as well as the created neural algorithms was applied for an enhanced FD in an industrial Hot dip zinc galvanizing system – i.e., utilized under real operation conditions.

References

1. Dimitrov, K.D.: *Development of Logistics Structures in a Zinc Galvanizing Facility via Design for Reliability Approach (DFR)*. XVIII National Scientific and Technical Conference "ADP – 2009", p. 420-431
2. Dimitrov, K.D.: *Process Modeling in Industrial Systems via Neural Networks*. XVIII National Scientific and Technical Conference "ADP – 2009", p. 582-592
3. Dimitrov, K.D.: *Some application of Neural Networks for Condition Monitoring in Industrial Systems*. XVIII National Scientific and Technical Conference "ADP – 2009", p. 593-598
4. Dimitrov, K.D.: *Neural Networks for fault diagnosis and process control in construction technologic systems*. Sofia, "Machine building" vol. 12, 1997, p. 250-255
5. Kosko, B.: *Neural Networks and Fuzzy Systems*. Englewood Cliffs, N.J., Prentice-Hall, 1992
6. Lind, M.: *Representing Goals and Functions of Complex Systems – an introduction to multilevel flow modeling*. TU of Denmark, Lyngby, 1996
7. Mamdani, E.H.: *Application of fuzzy algorithms for control of dynamic plants*, Proc. IEEE, 2005, p. 1585-1589
8. Narendra, K., et al.: *Identification and control of dynamic systems using neural networks*. Trans. Neural Networks, vol. 1, p. 4-27, 2000
9. Shaw, I.S., Kruger, J.J.: *New fuzzy learning model with recursive estimation for dynamic systems*. Fuzzy Sets and Systems, vol. 49, p. 217-220, 1999
10. Sugeno, M., Tanaka, K.: *Successive identification of a fuzzy model and its application*. Fuzzy Sets and Systems, vol. 42, p. 315-334, 1999
11. Sugeno, M., Yasukawa, T.: *A fuzzy logic-based approach to qualitative modeling*. IEEE Trans. Fuzzy Systems, vol. 1, p.7-31, 1998
12. Zurada, J.M.: *Introduction to Artificial Neural Systems*. West Publishing Company, 2003

Received in January 2011