

# EXCITATION CONTROL OF A SYNCHRONOUS MACHINE USING POLYNOMIAL NEURAL NETWORKS

Yannis L. Karnavas \* — Demetrios P. Papadopoulos \*\*

This paper presents a practical design of an intelligent type of controller using polynomial neural network (PNN) concepts for excitation control of a practical power generating system. This type of controller is suitable for real time operation and aims to improve the dynamic characteristics of the generating unit by acting properly on its original excitation system. The modelling of the power system under study consists of a synchronous generator connected via a transformer and a transmission line to an infinite bus. Next, digital simulations of the above system are performed using fuzzy control techniques that are based on previous work. Then, two neural network controllers are designed by adopting the PNN architecture. The first one utilizes a single pi-sigma network (PSN) and the significant advantages over the standard multi layered perceptron (MLP) are discussed. Secondly, an enhanced controller is designed, leading to a ridge polynomial network (RPN) by combining multiple PSNs if needed. Both controllers used can be pre-trained rapidly from the corresponding fuzzy logic controller (FLC) output signal and act as model dynamics capturers. The dynamic performances of the FLC along with those of the two proposed controllers are presented by comparison using the integral square error criterion (ISE). The latter controllers show good convergence properties and accuracy for function approximation. Typical transient responses of the system are shown for comparison in order to demonstrate the effectiveness of the proposed controllers. The computer simulation results obtained demonstrate clearly that the performance of the developed controllers offers competitive damping effects on the generator oscillations, with respect to the associated ones of the FLC, over a wider range of operating conditions, while their hardware implementation is easier and the computational time needed for real-time applications is drastically reduced.

**K e y w o r d s:** Synchronous machine, excitation control, dynamic stability, polynomial neural networks

## List of symbols

$\delta$	load angle
$t$	time
$i_{fd}$	field current
$i_d, i_q$	stator currents in $d$ and $q$ axis circuits, respectively
$i_t$	machine terminal current
$u_{fd}$	field voltage
$u_d, u_q$	stator voltages in $d$ and $q$ axis circuits, respectively
$U_b$	busbar voltage
$u_t$	machine terminal voltage
$P_m$	generator-shaft mechanical power
$R_T, X_T$	transformer resistance and reactance
$R_L, X_L$	transmission line resistance and reactance
$i_{kd}, i_{kq}$	damper circuit currents in $d$ and $q$ axes
$\omega$	machine speed
$K_e, T_e$	exciter gain and time constant
$\Delta U_{ref}$	incremental (step) voltage reference (input) change
$u_e$	excitation error

$u_{e.c.s}$	excitation controller voltage signal
$S$	Laplace operator
$r$	plant reference operating set point
$y$	plant output

## 1 INTRODUCTION

The problem of power system dynamic stability has received growing attention over the last decades. The main reasons for this are the increasing size of generating units and the use of high-speed excitation systems. The effect of the high-speed excitation on dynamic stability is to add negative damping to the system thereby causing oscillations with weak damping. A design of such an excitation system should also be satisfactory for a wide range of operating conditions as well as for fault conditions. Practical methods for nonlinear control include an open-loop inverse model of the nonlinear plant dynamics and the use of feedback loops to cancel the plant nonlinearities. The approximation of a non-linear system with a linearized model yields to the application of adaptive control, where real-time measurements of the plant inputs are used, either to derive explicitly the plant model or design a controller based on this model (indirect adaptive control), or

\* Laboratory of Electrical Machines & Installations, Department of Electrology, Technological & Educational Institute of Crete, School of Applied Technology, P.O. Box 1939, Zip.710 04, Heraklion, Crete, Greece, E-mail: karnab@stef.teicrete.gr

\*\* Laboratory of Electrical Machines Dept. of Electrical and Computer Engineering, Democritos University of Thrace, Vasilissis Sofias 12, 671 00, Xanthi, Greece, E-mail: dpapadop@ee.duth.gr

to directly modify the controller output (direct adaptive control). Typical studies concerning applications of modern algebraic and optimal control methods in excitation controller design using linear system models and output feedback have been presented before, *eg* [1–3].

Fuzzy approaches to intelligent control schemes treat situations where some of the defining relationships can be described by fuzzy sets and fuzzy relational equations [4–7]. Fuzzy logic controllers (FLCs) constitute knowledge-based systems that include fuzzy rules and fuzzy membership functions to incorporate human knowledge into their knowledge base. Some studies concerning applications in excitation controller design using fuzzy set theory have been developed before [5, 7–9].

Most knowledge-based systems rely upon algorithms that are cumbersome to implement and require extensive computational time. Fuzzy logic provides an inference that enables approximate human reasoning capabilities to be applied to knowledge-based systems. The combination of fuzzy logic and artificial neural networks (ANNs) theory for excitation control purposes has also been presented before [7–8], due to the ability of an ANN to become a universal function approximator [10]. Neural net based control possesses the ability to generalize learning. Moreover, it has been considered as the fastest among presently used approaches.

The aim of this paper is to investigate the use of PNN as a replacement of an existing (designed previously) FLC, applied in the excitation part of a practical synchronous machine workbench system. The first type of the PNN used is called pi-sigma network. This network utilizes product cells as the output units to indirectly incorporate the capabilities of higher-order networks, while using a fewer number of weights and processing units. The motivation here is to develop a systematic type of controller which maintains the fast learning property of single-layer higher-order networks, while avoiding the exponential increase in the number of weights and processing units required. The network has a regular structure, exhibits much faster learning, and is amenable to the incremental addition of units to attain a desired level of complexity. If such an incremental addition of units takes place a ridge polynomial network is produced. The second controller proposed here refers to this kind of architecture. Simulation results show good convergence properties and accuracy for function approximation. Comparative results using a FLC output training data set are also provided to highlight the learning, and subsequently control, abilities of the proposed PSN and RPN controllers.

At first, digital simulations of the above system are performed with FLC controller based on previous works by the authors of this model under various disturbance conditions [8]. Next, a new effort is made on the design and simulation of the higher-order neural network controllers. The new results are compared with those of the FLC. The overall evaluation of the proposed PSN and RPN controllers is made through the ISE criterion.

The computer simulation results obtained over a number of simulations clearly illustrate that there are potential capabilities of this proposed approach. From a practical point of view, the required computational time is reduced while the hardware implementation is quite easier than a FLC. It is also demonstrated that the performance of the two developed controllers will offer competitive damping effects on the generator oscillations over a wider range of operating conditions, with the associated ones of the FLC design.

## 2 SYNCHRONOUS MACHINE MODEL UNDER STUDY

The power system model to which the new PSN and RPN controllers designs are applied consists of an 87.5kVA alternator-set model connected via a transformer and transmission line to an infinite bus, is taken from [3] and is shown here in Fig. 1. The machine model was selected to have parameters broadly typical of those associated with larger machines. This approach can facilitate hardware changes as required and allows for practical tests to be carried out without disruptions in main generator sources. The main parameters for this machine are given in [3], but for practical purposes are stated also in the Appendix A. It is readily seen that the per-unit reactances are representative of turbogenerators in the range of 20–30 MW. The linearized system equations in standard state-space differential form are

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}, \quad \mathbf{y} = \mathbf{Cx} \quad (1)$$

where  $\mathbf{x} \in R^n$ ,  $\mathbf{y} \in R^m$  and  $\mathbf{u} \in R^p$  are the state, output and input vectors respectively, and  $\mathbf{A}$ ,  $\mathbf{B}$  and  $\mathbf{C}$  are real constant system matrices of appropriate dimensions.

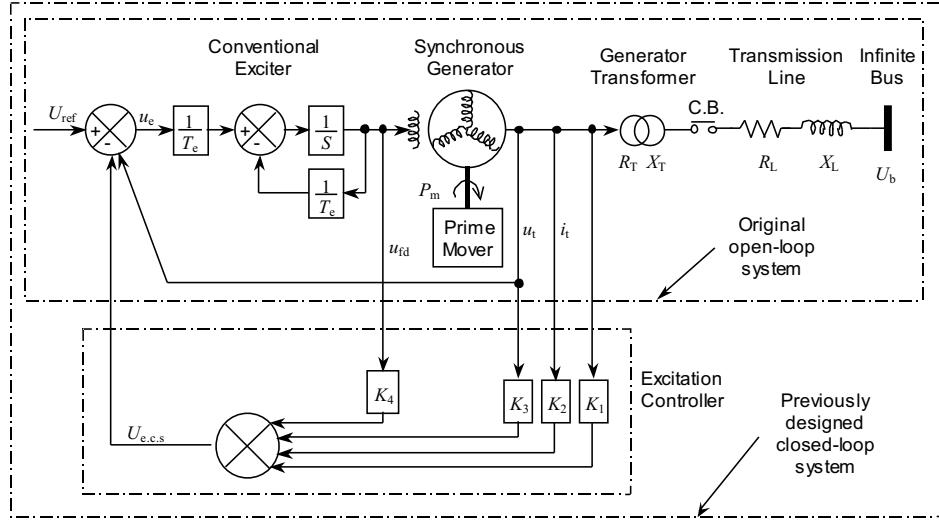
## 3 RELEVANT WORK PREVIOUSLY DEVELOPED

### 3.1 Conventional control

Based on [3] the non-transformed state-space model of the synchronous machine in the form of Eq. (1), is

$$\mathbf{x}^\top = [i_d \ i_q \ i_{fd} \ i_{kd} \ i_{kq} \ \delta \ \omega], \quad \mathbf{u}^\top = [u_{fd} \ P_m] \quad (2)$$

where the explicit numerical values of  $\mathbf{A}$  and  $\mathbf{B}$  [3] are based on the specific operating point given in Appendix A. In the same work, a 1<sup>st</sup>-order differential equation, describing the conventional exciter was introduced first and the 8<sup>th</sup>-order transformed open-loop model was derived. After that, based on the output energy criterion, the above model was re-arranged in decreasing order of state variable participation in the output energy of the system and after a relevant eigenvalue evaluation a satisfactory reduced order open-loop model (4<sup>th</sup>-order) was obtained. When output feedback was applied to the re-arranged 8<sup>th</sup>-order open-loop model, a gain vector was



**Fig. 1.** Simplified representation of synchronous machine plus exciter supplying power to the electric utility system through an interconnection network [3].

obtained resulting to an 8<sup>th</sup>-order transformed closed-loop model as well as to its associated reduced 4<sup>th</sup>-order model. Finally, the 8<sup>th</sup>-order transformed open-loop model was combined with the designed controller of the latter 4<sup>th</sup>-order closed-loop model to give an 8<sup>th</sup>-order transformed closed-loop model that resulted in improved overall performance.

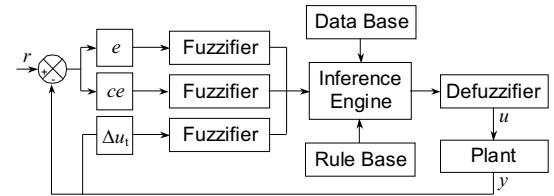
### 3.2. Intelligent control

#### 3.2.1. Fuzzy Logic Control

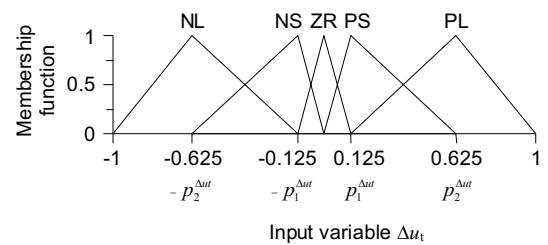
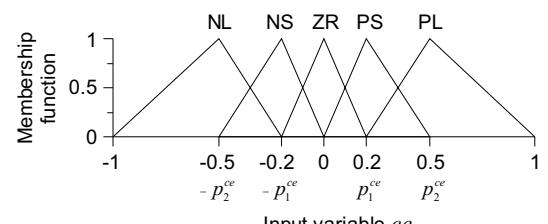
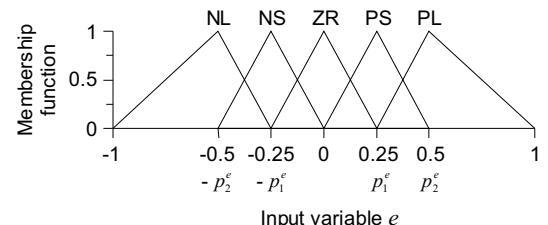
Based on [8], the basic configuration of the designed FLC comprises the four principal components: fuzzification interface, knowledge base, decision-making logic, and defuzzification interface (Fig. 2). The input variables are defined as the deviation from a reference or setpoint value, called the *error* ( $e$ ), and its first derivative, which in fuzzy control terms is usually called the *change in error* ( $ce$ ). It is found that the use of a third input variable which is the terminal voltage difference ( $\Delta u_t$ ) helps the FLC to display better performance. According to Fig. 1 and Fig. 2 the input variables are defined as:

$$\begin{aligned} e(t) &= \Delta U_{ref} - K_1 \delta(t) - K_2 i_t(t) - K_3 u_t(t) - K_4 u_{fd}(t) \\ ce(t) &= \Delta e(t)/\Delta t = [e(t) - e(t-1)] / T_s \\ \Delta u_t(t) &= u_t(t) - u_t(t-1) \end{aligned} \quad (3)$$

where  $T_s$  is the sampling time. The FLC output variable, which is the control input of the excitation system, is the field voltage  $u_{fd}$ .



**Fig. 2.** Structure of previously designed Fuzzy Logic Controller (FLC) [8].



**Fig. 3.** Membership functions of the (normalized) input fuzzy variables.

Next, the fuzzy set values of the input and output fuzzy variables are specified. With respect to a robust realization five fuzzy sets are defined for each linguistic (input and output) variable. The fuzzy set values of the fuzzy variables are chosen as follows: NL (Negative Large), NS (Negative Small), ZR (Zero), PS (Positive Small), PL (Positive Large) for the input variables, and VL (Very Large), (L) Large, (NR) Normal, VS (Very Small), S (Small) for the output variable. The input fuzzy variables, with their respective fuzzy set values, are shown in Fig. 3.

To determine how to modify the control variable  $u_{fd}$  from the monitored fuzzy input variables, two fuzzy associative matrices (FAM) were established. The synthesized FAMs for the FLC are shown in Fig. 4.

		ce				
		NL	NS	ZR	PS	PL
e	NL	VL	VL	VL		
	NS	VL	L	L		
	ZR	VL	L	NR	S	VS
	PS			S	S	VS
	PL			VS	VS	VS

$\Delta u_t$	NL	NS	ZR	PS	PL
$u_{fd}$	VL	L	NR	S	VS

Fig. 4. Fuzzy associative matrices for the output control variable  $u_{fd}$ .

### 3.2.2. Fuzzy Neural Control

Besides the FLC, an artificial neural network (NN) was employed also in [8] as a replacement for the plant dynamics evaluation inside the model-based control algorithm and the resulting neural network part along with the fuzzy logic part gave a new control structure as seen in Fig. 5 (FLNnC). The neural network was trained off-line, using the input and output of the conventional controller at the beginning of the on-line operation, the synapses between its neurons gave the same output as the conventional controller for the same input. As the controller operation continued (this was not desirable further) the neural network started learning the plant dynamics and also updating itself.

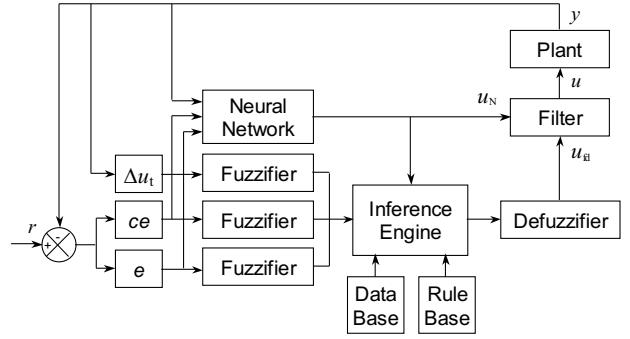


Fig. 5. Structure of previously designed Fuzzy Logic Neural network Controller (FLNnC) [8].

### 3.2.3. Genetic Fuzzy Control

In Fig. 3, the parameters  $p_1$ ,  $p_2$  (coordinates of the triangular shaped fuzzy sets) for each fuzzy variable were fixed to the values shown above. This could be considered as the main drawback of this type of FLC and was improved by the employment of a genetic algorithm (GA) [9]. Since triangular functions are chosen to represent the membership function, a fuzzy set's membership function is defined as a base function whose parameters can be adjusted adequately to meet the reference profile. The approach that could be followed for the base function is defined as follows (see Fig. 6):

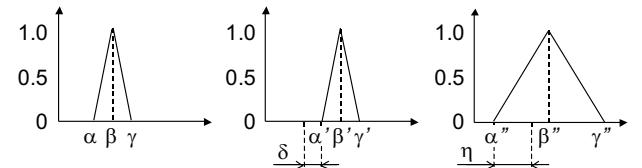


Fig. 6. A method for fuzzy set coefficient adjustment.

$$\begin{aligned}
 f(x : \alpha_i, \beta_i, \gamma_i) &= 0 &&; \alpha_i \geq x \\
 f(x : \alpha_i, \beta_i, \gamma_i) &= (x - \alpha_i)/(\beta_i - \alpha_i); \alpha_i \leq x \leq \beta_i && (4) \\
 f(x : \alpha_i, \beta_i, \gamma_i) &= (x - \gamma_i)/(\beta_i - \gamma_i); \beta_i \leq x \leq \gamma_i \\
 f(x : \alpha_i, \beta_i, \gamma_i) &= 0 &&; x \geq \gamma_i
 \end{aligned}$$

where  $\alpha_i$ ,  $\beta_i$ ,  $\gamma_i$  are the parameters. Thus, for each fuzzy set the following equations could be used:

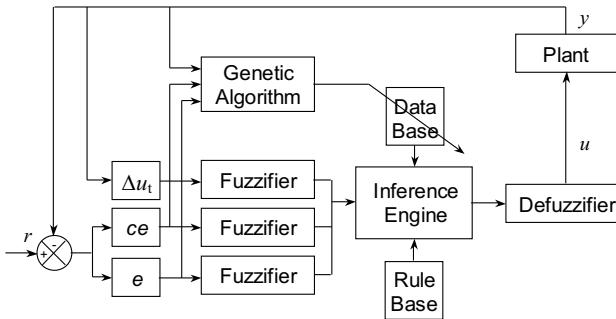
$$\begin{aligned}
 \alpha''_i &= (\alpha_i + \delta_i) - \eta_i \\
 \beta''_i &= (\beta_i + \delta_i) \\
 \gamma''_i &= (\gamma_i + \delta_i) + \eta_i && (5)
 \end{aligned}$$

where  $\delta_i$  and  $\eta_i$  were adjustment coefficients. The coefficient  $\delta_i$  causes the  $i^{\text{th}}$  membership function to move to

the right or to the left without changing shape. The coefficient  $\eta_i$  causes the  $i^{\text{th}}$  membership function to compress or elongate. The parameters  $\delta_i$  and  $\eta_i$  had to be optimised using the GA for each fuzzy subset. In our FLC case, there are three control variables with five subsets each. Thus the total number of parameters to be optimized by the GA involved were  $3 \times 5 \times 3 = 45$ . However, the nature of the excitation control problem requires fast response. The above achieved number of parameters is large. In order to reduce this number of parameters searched by the GA in each time instant, a common coordinate adjustment for the five fuzzy subsets of each of the three fuzzy variables  $e$ ,  $ce$ ,  $\Delta u_t$  (see Figs. 3 and 6) was employed as follows:

$$\begin{aligned}\gamma_2 &= \beta_3 = \alpha_4 = 0 \\ \gamma_5 &= -\alpha_1 = 1 \\ \gamma_3 &= \beta_4 = \alpha_5 = \alpha_3 = -\beta_2 = -\gamma_1 = p_1 \\ \gamma_4 &= \beta_5 = -\alpha_2 = -\beta_1 = p_2\end{aligned}\quad (6)$$

Finally, the total number of parameters was reduced to  $3 \times 2 = 6$  which turned out to be highly practical. The dynamic subsystems, *ie* the power system, the FLC and the GA, are interfaced with each other to form the overall genetic fuzzy controller (GFC) employed as shown in Fig. 7.



**Fig. 7.** Structure of previously designed Genetic Fuzzy Controller (GFC)[8].

#### 4 THE PROPOSED POLYNOMIAL NETWORK CONTROLLERS

##### 4.1 Higher-Order Feedforward Networks

Multi-layered perceptron (MLP) networks using the backpropagation learning rule or its variants have been successfully applied to applications involving pattern classification and function approximation as well as control schemes of power systems [8, 11]. Unfortunately, the training speeds for multi-layered networks are extremely slower than those for feedforward networks being composed of a single layer of threshold logic units (TLU), and using the perceptron, ADALINE or Hebbian type learning rules [12]. Moreover, these networks converge very

slowly in typical situations dealing with complex and nonlinear problems, and do not scale well with problem size [12].

Higher-order correlations among the input components can be used to construct a higher-order network to yield a nonlinear discriminant function using only a single layer of cells [13]. The building block of such networks in the higher-order processing unit (HPU), defined as a neural processing unit that includes higher-order input correlations, and its output  $y$ , is given in [14]:

$$y = \sigma \left( \sum_j w_j x_j + \sum_{j,k} w_{jk} x_j x_k + \sum_{j,k,l} w_{jkl} x_j x_k x_l + \dots \right) \quad (7)$$

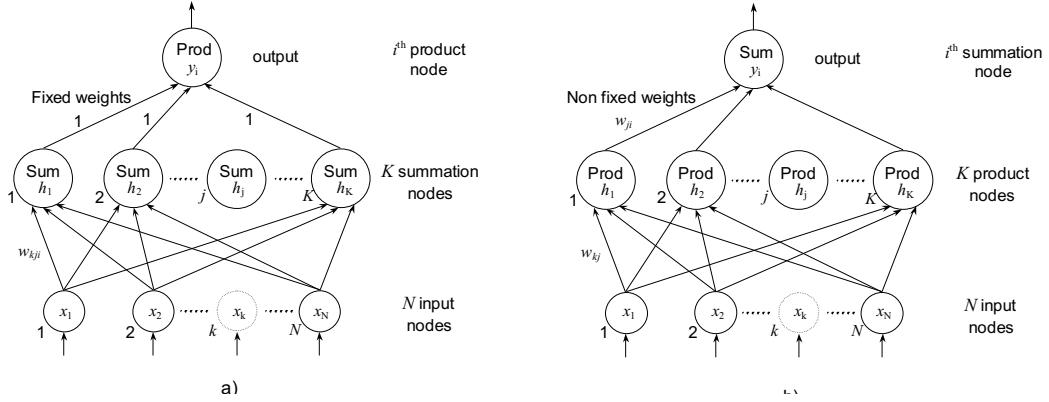
where  $\sigma(x)$  is a nonlinear function of input  $x$ ,  $x_j$  is the  $j^{\text{th}}$  component of  $x$ , and  $w_{jkl\dots}$  is an adjustable weight from product of inputs  $x_j$ ,  $x_k$ ,  $x_l \dots$  to the HPU. If the input is of dimension  $N$ , then a  $k^{\text{th}}$  order HPU needs a total of

$$\sum_{i=0}^k \binom{N+i-1}{i} \quad (8)$$

weights if all products of up to  $k \leq N$  components are to be incorporated [15].

A single layer of HPU (SLHPU) network is one in which only one layer of mapping from input to output using HPUs is considered. The order of a SLHPU network is the highest order of any of the constituent HPUs. Thus output of the  $k^{\text{th}}$  order SLHPU is a nonlinear function of up to the  $k^{\text{th}}$  order polynomials. Since it does not have hidden units as in sigma-pi network, reliable single layer learning rules such as perceptron, ADALINE, or Hebbian type rules can be used. However, to accommodate all higher-order correlations, the number of weights required increases combinatorially in the dimensionality of the inputs [15]. Limiting the order of the network leads to a reduction in the classification capability of the network.

There have been many approaches that maintain the powerful discrimination capability of higher-order networks while reducing higher-order terms. For example, sigma-pi networks use a hidden layer of higher-order TLUs [12]. A multi-layering strategy using sigma-pi units retains the full capability of a higher-order network using a smaller number of weights and processing units, but its learning speed is slower due to layering. Another approach is to use a priori information to remove the terms that are irrelevant to the problem in a single layer of higher-order TLUs [13],[16]. However, since it is often difficult to find the properties of input pattern space a priori, this strategy has limited applications.



**Fig. 8.** (a) Proposed Pi-Sigma network (PSN) controller architecture, (b) Standard MLFF neural network architecture.

#### 4.2 Pi-Sigma Network Architecture

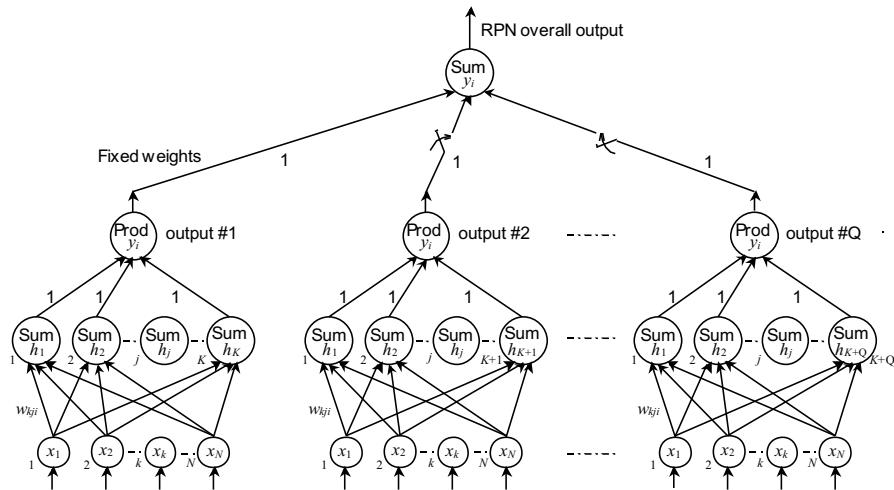
Figure 8(a) shows the proposed PSN controller for the present work. The input  $x$  is an  $N$  dimensional vector and  $x_k$  is the  $k^{\text{th}}$  component of  $x$ . The inputs are weighted and fed to a layer of  $K$  linear summing units, where  $K$  is the desired order of the network. Let  $h_{ji}$  be the output of the  $j^{\text{th}}$  summing unit for the  $i^{\text{th}}$  output,  $y_i$ , then,

$$h_{ji} = \sum_k w_{kji} x_k + \Theta_{ji}, \text{ and } y_i = \sigma\left(\prod_j h_{ji}\right) \quad (9)$$

where  $w_{kji}$  is an adjustable weight from input  $x_k$  to the  $j^{\text{th}}$  summing unit of the  $i^{\text{th}}$  output, and  $\Theta_{ji}$  is an adjustable threshold of the  $j^{\text{th}}$  summing unit of the  $i^{\text{th}}$  output. The  $\sigma(x)$  denotes the nonlinear activation function, and is selected as the logistic function,  $\sigma(x) = 1/(1+e^{-x})$  for all the results reported in this paper. Note

that connections from summing units to an output have fixed weights. Thus there is no notion of hidden units in the network and fast learning rules can be used.

The basic idea behind this type of network is that we can represent the input of a  $K^{\text{th}}$  order processing unit by a product of  $K$  linear combinations of the input components [17]. That is why this network is called pi-sigma instead of sigma-pi (Fig. 8(b)). In the SLHPU representation, the polynomial is represented by summation of all partial products of input components up to order  $K$ , thereby leading to an exponential increase in the number of adjustable weights required, as indicated in Table 1. From the network topology point of view, this leads to an irregular structure. In the case of the PSN, using an additional summing unit increases the network's order by 1 while preserving old connections and maintaining network topology.



**Fig. 9.** Proposed Ridge Polynomial Network (RPN) controller architecture.

**Table 1.** Number of weights required for PSN and SLHPU

order of network	# of weights			
	Pi-sigma		SLHPU	
	$N = 5$	$N = 10$	$N = 5$	$N = 10$
2	12	22	21	66
3	18	33	56	286
4	24	44	126	901

#### 4.2.1 Probabilistic Learning Rule Used in the PSN

There are a total of  $(N + 1)K$  adjustable weights and thresholds for each output unit, since there are  $N + 1$  weights associated with each summing unit. The learning rule is a randomized version of the gradient descent procedure. Since the output  $y_i$  is a function of the product of all the  $h_{ji}$ 's, we do not have to adjust all the variable weights at each learning cycle. Instead, at each update step, we randomly select a summing unit and update the set of  $N + 1$  weights associated with its inputs based on a gradient descent approach. Let the mean square error (MSE) be

$$e^2 = \frac{1}{2} \sum_p \sum_i (d_i^p - y_i^p)^2 \quad (10)$$

where superscript  $p$  denotes the  $p^{\text{th}}$  training pattern,  $d_i$  and  $y_i$  are the  $i^{\text{th}}$  components of desired and actual outputs, respectively, and the summation is over all outputs and all training patterns. Applying gradient descent on this estimate of the MSE, we obtain the update rules for weights and threshold for each iteration step as

$$\Delta\Theta_{li} = \eta(d_i - y_i)y'_i \prod_{j \neq i} h_{ji} \quad (11)$$

$$\Delta w_{kli} = \eta(d_i - y_i)y'_i \prod_{j \neq i} h_{ji}x_k = \Delta\Theta_{li}x_k \quad (12)$$

where  $\eta$  is the scaling factor or the learning rate, and  $y'_i$  is the first derivative of the logistic function that is,  $y'_i = \sigma'(x) = (1 - \sigma(x))\sigma(x)$ . It is repeated that these updates are applied only to the set of weights and threshold corresponding to the  $l^{\text{th}}$  summing unit which is chosen randomly at each step.

#### 4.3 Ridge Polynomial Network Architecture

The PSN provides only a constrained approximation of a power series. Because of this truncated approximation capability, the PSN can not uniformly approximate all continuous multivariate functions that can be defined on a compact set. However, universal approximation can be attained by summing the outputs of several PSN of different order. The resulting combined network of PSN is called ridge polynomial network.

For  $\mathbf{x} = (x_1, \dots, x_d)^\top$  and  $\mathbf{w} = (w_1, \dots, w_d)^\top \in R^d$ , the  $\langle \mathbf{x}, \mathbf{w} \rangle$  is denoted as their inner product. For a given compact set  $C \subset R^d$ , all functions defined on  $C$  in the form of  $f(\langle \mathbf{x}, \mathbf{w} \rangle)$ , where  $f$  is a continuous function in one variable, are called ridge functions. A ridge polynomial is a ridge function that can be represented as

$$\sum_{i=0}^n \sum_{j=1}^m a_{ij} \langle \mathbf{x}, \mathbf{w}_{ij} \rangle^i \quad (13)$$

for some  $a_{ij} \in R$  and  $\mathbf{w}_{ij} \in R^d$ .

Let  $\Pi_k^d$  denote the set of all polynomials in  $R^d$  with degree  $\leq k$ . Then, for any polynomial  $p(x)$  in  $\Pi_k^d$ , there exist  $w_{ji} \in R$  and  $\mathbf{w}_{ji} \in R^d$  such that,

$$p(x) = \sum_{j=1}^N \prod_{i=1}^j (\langle \mathbf{x}, \mathbf{w}_{ji} \rangle + w_{ji}) \quad (14)$$

The original form of the theorem is more complicated than the one presented here. However, since we are concerned with the existence of a representation of multivariate polynomials in terms of ridge polynomials, a simpler statement is adopted. The detailed proof can be found in [18].

Figure 9 shows the generic network architecture of the RPN using the PSN as basic building blocks. Since, in general, there may not be much a priori information about the function to be approximated, it is difficult to choose an appropriate network size. On the other hand, an RPN can be incrementally grown to meet a predefined error criterion.

#### 4.3.1 Incremental Learning Algorithm Used in the RPN

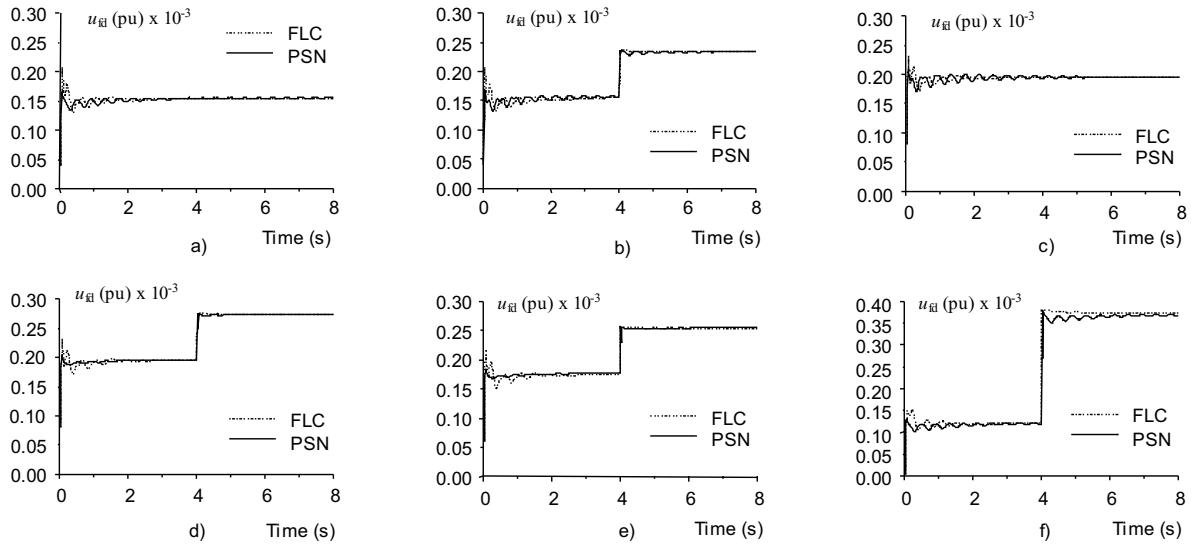
Instead of limiting the activation functions to linear ones, we can use activation functions such as sigmoid or hyperbolic tangent. Thus, using a fixed network architecture, an unknown function  $f$  in  $R^d$  can be approximated by the direct use of the RPN model of degree up to  $k$  based on

$$f(x) \approx \sigma \left( \sum_{i=1}^k P_i(\mathbf{x}) \right) \quad (15)$$

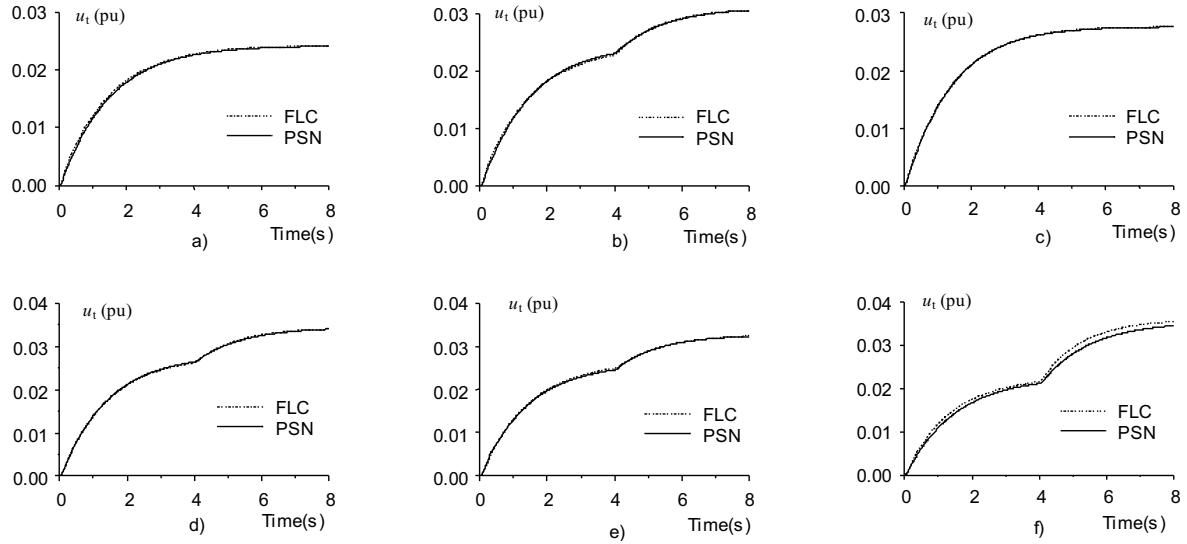
where

$$P_i(\mathbf{x}) = \prod_{j=1}^i (\langle \mathbf{w}_j, \mathbf{x} \rangle + w_{j0}), \quad i = 1, \dots, k \quad (16)$$

and  $\sigma(\cdot)$  is a suitable linear or nonlinear activation function,  $\mathbf{w}_j \in R^d$  and  $w_{j0} \in R$  are determined by learning process. Since each  $P_i$  in Eq. (15) is obtainable as the output of a PSN of degree  $i$  with linear output units, the learning algorithms for the PSN can be used for the RPN. Equipped with suitable error measure (eg MSE), every  $P_i$  is optimized by fixing other polynomials.



**Fig. 10.** Time responses of field voltage  $u_{fd}$  when FLC and PSN are applied. (A) through (F) are the corresponding cases of disturbance (Table 2).



**Fig. 11.** Time responses of terminal voltage  $u_t$  when FLC and PSN are applied. (A) through (F) are the corresponding cases of disturbance (Table 2).

**Table 2.** Disturbance Cases Applied to Power System Under Study

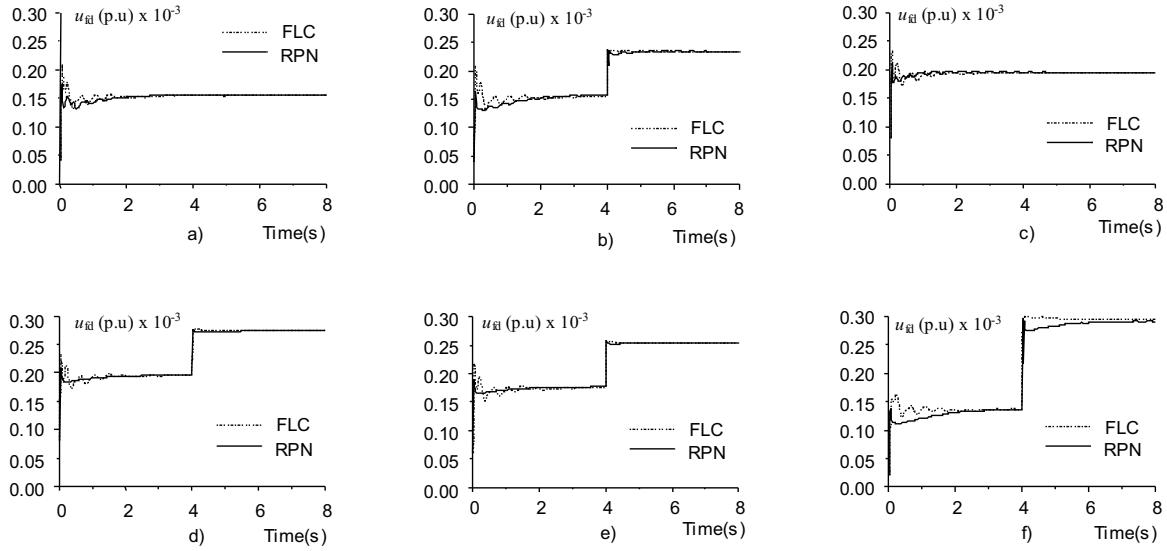
Case No	$\Delta u_e$ (pu)	At time instant (sec)	Change in $\Delta u_e$ (pu)	At time instant (sec)
A	0.0010	0.0	—	—
B	0.0010	0.0	0.002	4.0
C	0.0020	0.0	—	—
D	0.0020	0.0	0.002	4.0
E*	0.0015	0.0	0.002	4.0
F*	0.0005	0.0	0.004	4.0

(\*) The weights that PSN and RPN use are from Case D.

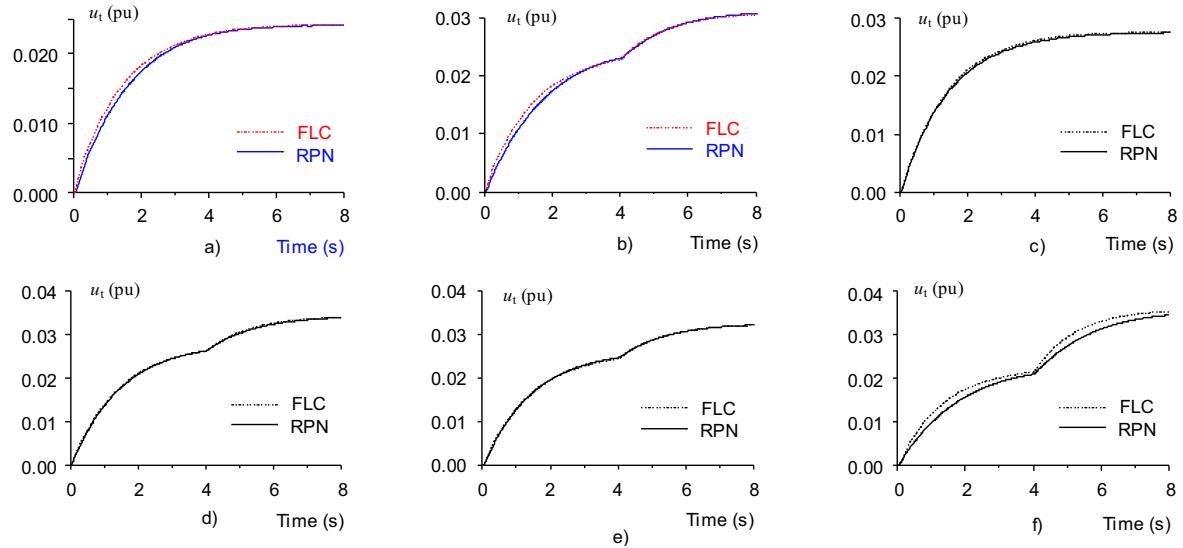
An incremental learning algorithm proceeds as follows. We denote  $k$  an algorithmic step at which  $P_k$  is added to the network with “predetermined”  $P_0, \dots, P_{k-1}$ . That is, with  $P_0 \equiv 0$ , the function  $f$  is approximated by the RPN at the  $k^{\text{th}}$  step as,

$$f(\mathbf{x}) \approx \sigma \left( \sum_{i=0}^{k-1} P_i(\mathbf{x}) + P_k(\mathbf{x}) \right) \quad (17)$$

where the weights in  $P_0, \dots, P_{k-1}$  are frozen once the  $k^{\text{th}}$  degree PSN is added. This is equivalent to approximating  $\sigma^{-1} \circ f(\mathbf{x}) - \sum_{i=0}^{k-1} P_i(\mathbf{x})$  with a  $k^{\text{th}}$  degree PSN,  $P_k(\mathbf{x})$ , with linear output units.



**Fig. 12.** Time responses of field voltage  $u_{fd}$  when FLC and RPN are applied. (A) through (F) are the corresponding cases of disturbance (Table 2).



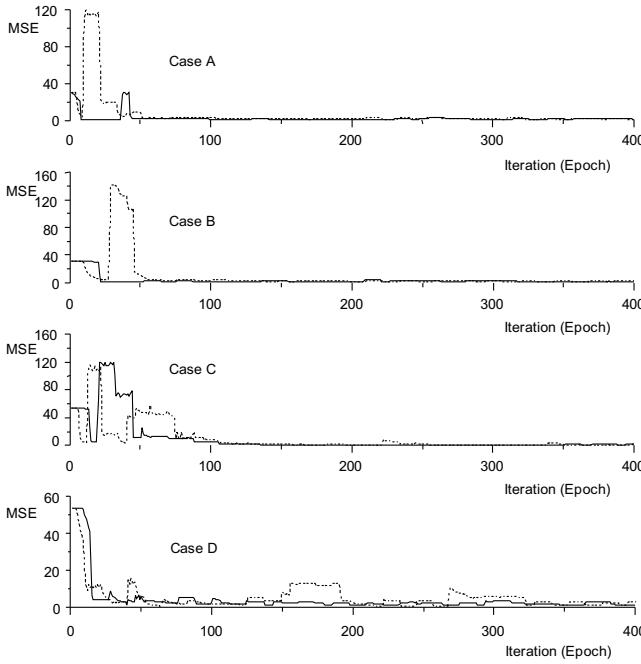
**Fig. 13.** Time responses of terminal voltage  $u_t$  when FLC and RPN are applied. (A) through (F) are the corresponding cases of disturbance (Table 2).

## 5 APPLICATION AND SIMULATION RESULTS

The designed new PSN and RPN controllers are applied to the power system of Fig. 1 and several cases of interest have been investigated. Some of the cases studied, the results of which are shown here, are summarized in Table 2. For these cases of disturbance, the FLC model was also simulated under the same conditions in order to compare the results with the associated ones of the PSN and RPN controllers.

For Cases A through D the FLC input ( $e(t)$ ,  $ce(t)$ ,  $\Delta u(t)$ ) and output ( $u_{fd}$ ) signals were used to train the two proposed controllers. To show the capability of the

learning algorithms used, the PSN as well as the RPN were not trained at all for the Case E and Case F, but the final weights of Case D were used instead. In the other Cases, the initial weights were randomly assigned values between  $-0.5$  and  $0.5$  for all the PSNs used. The learning rate was held constant for a PSN but decreased by a factor of 1.7 if another PSN was added. A new PSN was added if the ratio of the difference between MSEs of the previous epoch and the current epoch and the MSE of the previous epoch was less than a threshold,  $\varepsilon_{th}$ . Since the dynamic range of the error between the desired output and the actual algorithm output becomes smaller as learning proceeds,  $\varepsilon_{th}$  was decreased by a factor of



**Fig. 14.** MSE for the PSN and the RPN training phases (Cases A through D).

10 at each addition of a PSN. In case of a RPN used, the training started with a 3<sup>rd</sup> degree PSN. The initial learning rate and  $\varepsilon_{th}$  were 0.8 and 0.0001 respectively. The learning was quite stable and the MSE decreased drastically when additional PSNs were added.

Figures 10 and 11 show the time responses of the field voltage ( $u_{fd}$ ) and the controlled variable ( $u_t$ ) respectively, when FLC and PSN are applied, whereas Figs. 12 and 13 show the corresponding responses when FLC and RPN are applied for the disturbance Cases of Table 2. From these figures it is clear that the PSN and RPN performance competes the associated ones of the FLC. That proves the ability of these type of networks to capture the dynamics of the system they control (as can be seen from Figs. 10 and 12), not only for cases that they are trained for (*ie* Case A through D) but also for a wider operating region (Case E and F). Consequently, the relative improvement of the system's dynamics regarding the controlled variable is readily seen in Figs. 11 and 13. Apparently, the developed controllers show good convergence properties and accuracy for function approximation while their performance offers competitive damping effects on the generator oscillations, with respect to the associated ones of the FLC. It is to be noted that the FLC and PSN/RPN controllers have quite the same characteristics due to the fact that the FLC was used to train the PSN/RPN ones. The MSE for the cases that the PSN and RPN are trained from the FLC input and output variables are shown in Fig. 14. It can be seen that the RPN (with one or two more PSN blocks) can be trained quite faster and in a more stable manner than the single PSN type. Finally, the integration-square-error-time (ISET) criterion of the following form is used to evaluate

the quality performance of the relevant controller designs. The results are summarized in Table 3.

$$J = \int_0^8 u_t(t)t dt \quad (18)$$

**Table 3.** Performance index (Eq. 18) of applied controllers

Case	FLC	PSN	RPN
A	0.15950	0.15795	0.15705
B	0.17783	0.17751	0.17447
C	0.18289	0.18249	0.18097
D	0.20121	0.20118	0.19999
E	0.19060	0.18965	0.18948
F	0.18822	0.18092	0.17696

## 6 CONCLUSIONS

In this paper two implementations of excitation controllers for a synchronous machine using polynomial neural networks were developed and presented. The proposed controllers demonstrate robust stability properties, since their training is based on a fuzzy sub-system which takes the actual operating conditions into consideration. The results obtained amply demonstrate that the performance of the PSN as well as the RPN controllers are competitive with those obtained with the fuzzy logic excitation controller designs and techniques previously developed. It is evident that certain intelligent control applications could increase the efficiency and subsequently make the operation of a power system more economic. It is emphasized that the hardware implementation for such kind of controllers is easier than FLC ones and the computational time needed for real-time applications is drastically reduced. A practical implementation on a microprocessor system could be used as an addition to the existing controllers of such power systems or as a substitute for an optimum control and supervision.

## Appendix A

Principal system data & machine operating point  
(pu values on machine rating)

Synchronous Machine	$x_d$	1.758 pu	$R_a$	0.01450 pu
87.5 kVA, 415 V,	$x_q$	0.990 pu	$R_{kd}$	0.00422 pu
4-pole	$x_{fd}$	1.761 pu	$R_{kq}$	0.01260 pu
	$x_{kd}$	1.664 pu	$R_{fd}$	0.00268 pu
	$x_{kq}$	0.955 pu	$H$	1.434 s
	$x_{md}$	1.658 pu		
	$x_{mq}$	0.899 pu		
Conventional Exciter	$K_e$	1	$T_e$	0.472 s
Transformer	$R_T$	0.03630 pu	$X_T$	0.0838 pu
Transmission line	$R_L$	0.03744 pu	$X_L$	0.3903 pu
$P_t$	$Q_t$	$u_t$	$u_{fd}$	$\delta$
0.8 pu	0.84 pu	1.29 pu	0.0043 pu	0.56 rad
				0.97 pu

## REFERENCES

- [1] MAO, H.—MALIK, O. P.—HOPE, G. S.—FAN, J.: An Adaptive Generator Excitation Controller Based on Linear Optimal Control, *IEEE Trans. on Energy Conversion* **EC-5** No. 4 (1990), 673–678.
- [2] PAPADOPOULOS, D. P.: Excitation Control of Turbogenerators with Output Feedback, *Int. J. Electr. Power & Energy Systems* **8** (1986), 176–181.
- [3] PAPADOPOULOS, D. P.—SMITH, J. R.—TSOURLIS, G.: Excitation Controller Design of Synchronous Machine with Output Feedback Using High and Reduced Order Models, *Archiv für Elektrotechnik* **72** (1989), 415–426.
- [4] ROSS, T. J.: *Fuzzy Logic with Engineering Applications*, McGraw Hill College Div, 1995.
- [5] HASSAN, M. A. M.—MALIK, O. P.—HOPE, G. S.: A Fuzzy Logic Based Stabilizer for a Synchronous Machine, *IEEE Trans. Energy Conv.* **EC-6** No. 3 (1991), 407–413.
- [6] HANDSCHIN, E.—HOFFMANN, W.—REYER, F.—STEPHANBLOME, Th.—SCHLUCKING, U.—WESTERMANN, D.—AHMED, S. S.: A new Method of Excitation Control Based on Fuzzy Set Theory, *IEEE Trans. Power Syst.* **9** (1994), 533–539.
- [7] DJUKANOVIC, M. B.—DOBRIJEVIC, D. M.—CALOVIC, M. S.—NOVICEVIC, M.—SOBAJIC, D. J.: Coordinated Stabilizing Control for the Exciter and Governor Loops Using Fuzzy Set Theory and Neural Nets, *Int. J. Electr. Power & Energy Syst.* **8** (1997), 489–499.
- [8] KARNAVAS, Y. L.—PAPADOPOULOS, D. P.: Excitation Control of a Power Generating System Based on Fuzzy Logic and Neural Networks, *Eur. T. Electr. Power* **10** No. 4 (2000), 233–241.
- [9] KARNAVAS, Y. L.—PAPADOPOULOS, D. P.: A Genetic-Fuzzy System for the Excitation Control of a Synchronous Machine, In: ICEM '02, The 15<sup>th</sup> International Conference in Electrical Machines, Bruges, Belgium, 25–28 August, 2002, CD paper Ref. No. 204.
- [10] HORNICH, K.—STINCHCOMEBE, M.—WHITE, H.: Multi-layer Feedforward Networks are Universal Approximators, *Neural Networks* **2** (1989), 359–366.
- [11] KARNAVAS, Y. L.—PAPADOPOULOS, D. P.: AGC for Autonomous Power Station Using Combined Intelligent Techniques, *Int. J. Electr. Pow. Syst. Res.* **62** (2002), 225–239.
- [12] McCLELLAND, J.—RUMELHART, D.: *Parallel Distributed Processing*. Vol. 1, The MIT Press, 1987.
- [13] LEE, C.—MAXWELL, T.: Learning, Invariance, and Generalization in High-Order Neural Network, *Applied Optics* **26** (23) (1987).
- [14] LIPPmann, R. P.: Pattern Classification Using Neural Networks, *IEEE Communications Magazine* (1989), 47–64.
- [15] MINSKY, M.—PAPERT, S.: *Perceptrons*, The MIT Press, 1969.
- [16] REID, M. B. *et al*: Rapid Training of Higher-Order Neural Networks for Invariant Pattern Recognition, In: Proc. of IJCNN, Washington DC, 1:689–692, 1989/.
- [17] SHIN, Y.—GHOSH, J.: The pi-sigma Network: an Efficient Higher-Order Neural Network for Pattern Classification and Function Approximation, In: Proc. of IJCNN, Seattle, 1991 1:13–18.
- [18] SHIN, Y.—GHOSH, J.: Efficient Higher-Order Neural Networks for Function Approximation and Classification, *Int. J. Neural Syst.* **3** No. 4 (1992), 323–350.

Received 17 September 2003

**Yannis L. Karnavas** (1969), received the Diploma (1994) and PhD (2002) in Electrical Engineering from the Dept. of Electrical & Computer Engineering of Democritos University of Thrace, Greece. He has been working (1995–1998, 2000–2004) as scientific associate of the Electrical Machines Laboratory of the Technological Educational Institute of Kavala, Kavala, Greece. Currently, he is an Assistant Professor and Director of the Electrical Machines & Installations Laboratory of the Department of Electrology at the Technological Educational Institute of Crete. He is also a Chartered Electrical Engineer in the Technical Chamber of Greece and he engages in technical studies. He is an IEEE member. His research interests include power system operation and control, industrial process control and artificial intelligence applications to them.

**Demetrios P. Papadopoulos** (1942) received the BSEE (1965), MSEE (1968) and PhD (1970) in Electrical Engineering from Marquette University, Milwaukee, Wisconsin, USA During 1970 to 1972 he was Assistant Professor at the Department of Electrical Engineering of Gonzaga University, Spokane, Washington, USA From 1972 to 1997 he was with the Public Power Corporation of Greece working on special projects of power systems. Since 1981 he is Professor and Director of the Electrical Machines Laboratory of the Dept. of Electrical & Computer Engineering at Democritos University of Thrace (DUTH). From 1987 to 1988 he served as Vice-Rector at DUTH and from 1989 to 1991 as General Secretary of Eastern Macedonia — Thrace Region of Greece. He is Senior Member of IEEE, Member of the Technical Chamber of Greece, and also of the Societies ΠΙΜΕ, ΤΒΠΙ, ΗΚΝ and ΣΞ. His research interests are in electrical machines, and in power production with small hydro, wind conversion and cogeneration systems.