EXCITATION CONTROL OF A SYNCHRONOUS MACHINE USING POLYNOMIAL NEURAL NETWORKS

G Iswarya, M.Tech Student, PVKK Institute of Technology, Anantapur
D Mahesh Kumar, Head of the Department, PVKK Institute of Technology, Anantapur

ABSTRACT :

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

δ	load angle
t	time
i_{fd}	field current
i_d, i_q	stator currents in d and q axis circuits, respec-tively
i_t	machine terminal current
u_{fd}	field voltage
u_d, u_q	stator voltages in d and q axis circuits, respec-tively
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 reactancei_{kd}, i_{kq} damper circuit currents in d and q

axesω machine speed

K_e, T_e exciter gain and time constant

ΔU_{ref} incremental (step) voltage reference (input)chang

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 re- ceived 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 oscilla- tions 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. Prac- tical methods for nonlinear control include an open-loop inverse model of the nonlinear plant dynamics and the useof 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, ei- ther to derive explicitly the plant model or design a con- troller based on this model (indirect adaptive control), or to directly modify the controller output (direct adaptive control). Typical studies concerning applications of mod- ern 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 member- ship 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 exten- sive computational time. Fuzzy logic provides an infer- ence that enables approximate human reasoning capa- bilities 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]. Neu- ral 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 PNNas a replacement of an existing (designed previously) FLC, applied in the excitation part of a practical syn- chronous 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 in- corporate 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 ex- ponential increase in the number of weights and process- ing units required. The network has a regular structure, exhibits much faster learning, and is amenable to the in- cremental 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 architec- ture. Simulation results show good convergence proper- ties and accuracy for function approximation. Compara- tive results using a FLC output training data set are also provided to highlight the learning, and subsequently con- trol, 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 con- trollers. 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 num-ber of simulations clearly illustrate that there are poten- tial capabilities of this proposed approach. From a prac- tical point of view, the required computational time is reduced while the hardware implementation is quite eas- ier than a FLC. It is also demonstrated that the perfor- mance of the two developed controllers will offer compet- itive damping effects on the generator oscillations over a wider range of operating conditions, with the associated ones of the FLC design.

2 SYNCHRONOUS MACHINEMODEL 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 trans- former 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 facili- tate hardware changes as required and allows for practical tests to be carried out without disruptions in main gen- erator 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

$$\dot{x} = \mathrm{A}x + \mathrm{B}u, \; y = \mathrm{C}x \qquad (1)$$

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

3 RELEVANT WORK PREVIOUSLY DEVELOPED

Conventional control

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

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

where the explicit numerical values of A and B [3] are based on the specific operating point given in Ap-pendix A. In the same work, a 1st -order differential equa- tion, describing the conventional exciter was introduced first and the 8th -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 sat- isfactory reduced order open-loop model (4th -order) was obtained. When output feedback was applied to the re- arranged 8th -order open-loop mod∈l, a gain∈vector was ∈

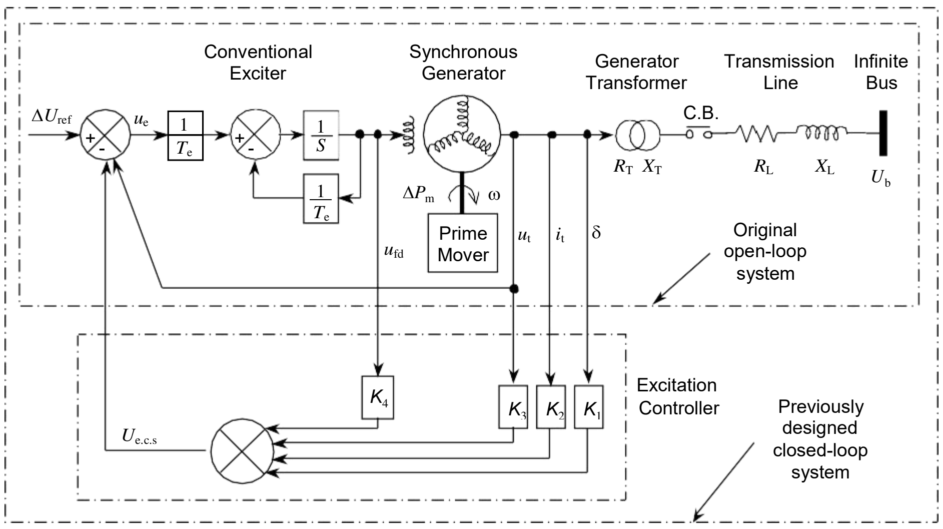


Fig. 1. Simplified representation of synchronous machine plus exciter supplying power to the electric utility system through an intercon- nection network [3].

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

Intelligent control

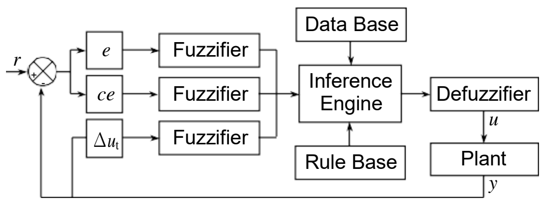


Fig. 2. Structure of previously designed Fuzzy Logic Controller(FLC) [8].

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 whichis the terminal voltage difference (Δu_t) helps the FLCL to display better performance. According to Fig. 1 and Fig. 2 the input variables are defined as:

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

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}.

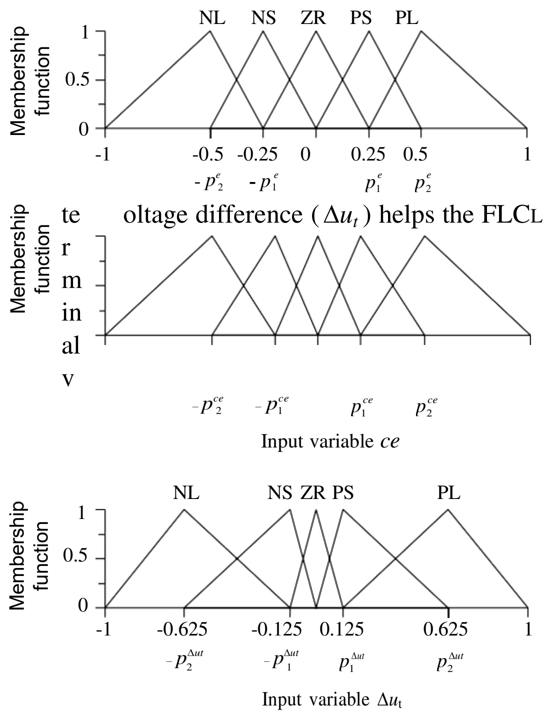


Fig. 3. Membership functions of the (normalized) input fuzzy vari-ables

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 linguis- tic (input and output) variable. The fuzzy set values of the fuzzy variables are chosen as follows: NL (Nega- tive Large), NS (Negative Small), ZR (Zero), PS (Posi- tive 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 asso- ciative matrices (FAM) were established. The synthesized FAMs for the FLC are shown in Fig. 4.

	ce				
	NL	NS	ZR	PS	PL

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

Δu_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 algo- rithm 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 con- ventional 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.

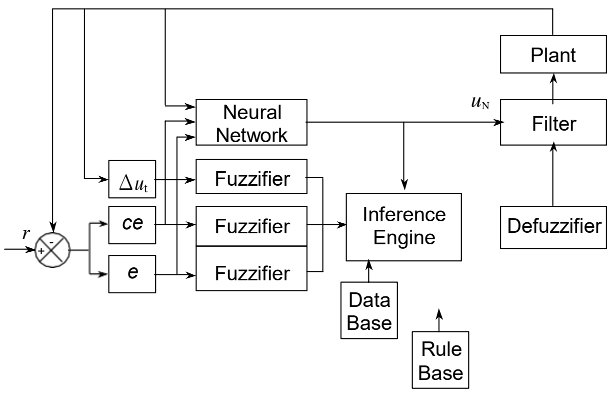


Fig. 5. Structure of previously designed Fuzzy Logic Neural net-work Controller (FLNnC) [8].

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 im- proved 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):

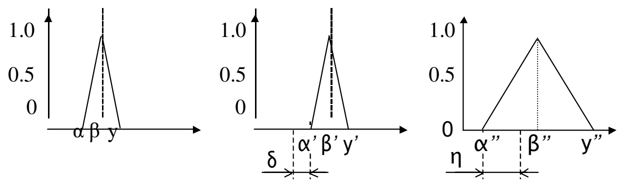


Fig. 6. A method for fuzzy set coefficient adjus

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

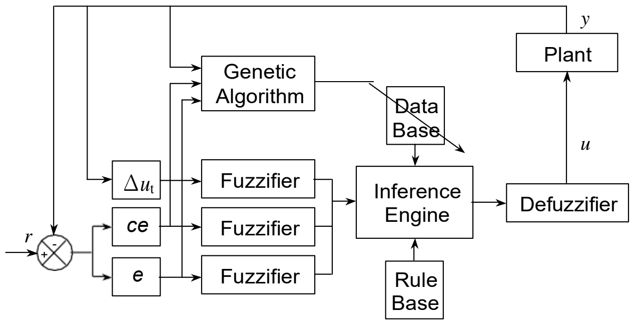
where α_i, β_i, γ_i are the parameters. Thus, for each fuzzyset the following equations could be used:

$$\alpha_i'' = (\alpha_i + \delta_i) - \eta_i$$
$$\beta_i'' = (\beta_i + \delta_i) \qquad (5)$$
$$\gamma_i'' = (\gamma_i + \delta_i) + \eta_i$$

where δ_i and η_i were adjustment coefficients. The coef- ficient δ_i causes the i^{th} membership function to move to the right or to the left without changing shape. The coeffi- cient η_i causes the i^{th} membership function to compress or elongate. The parameters δ_i and η_i had to be opti- mised 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 opti- mized by the GA involved were $3 \times 5 \times 3 = 45$. How- ever, 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 co- ordinate adjustment for the five fuzzy subsets of each of the three fuzzy variables e , ce , δu_t (see Figs. 3 and 6) was employed as follows:

$$\gamma_2 = \beta_3 = \alpha_4 = 0$$
$$\gamma_5 = -\alpha_1 = 1 \qquad (6)$$
$$\gamma_3 = \beta_4 = \alpha_5 = \alpha_3 = -\beta_2 = -\gamma_1 =$$
$$p_1\gamma_4 = \beta_5 = -\alpha_2 = -\beta_1 = p_2$$

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 Fwhere and the GA, are interfaced with each other to form the overall genetic fuzzy controller (GFC) employed as shown in Fig. 7.



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

Higher-order correlations among the input compo- nents 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 net- works in the higher-order processing unit (HPU), defined as a neural processing unit that includes higher-order in- put correlations, and its output y, is given in [14]:

$$y = \sigma \sum_{j} w_j x_j + \sum_{j,k} w_{jk} x_j x_k + \sum_{j,k,l} w_{jkl} x_j x_k x_l + \cdots \qquad (7)$$

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

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

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

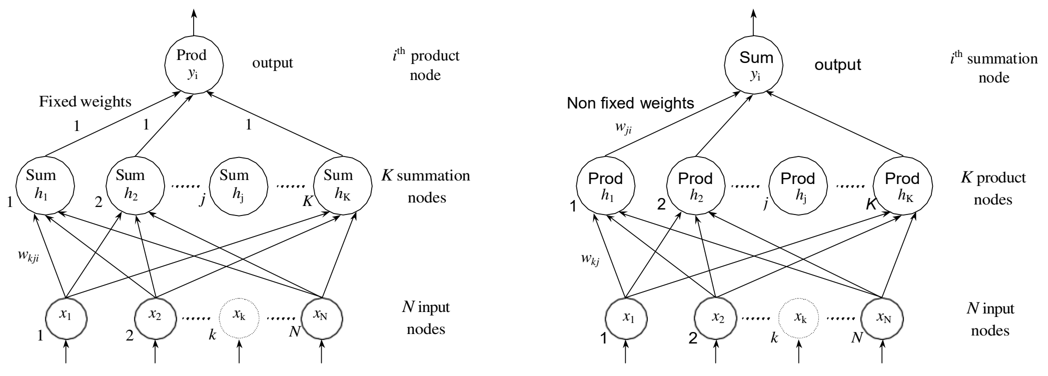
4 THE PROPOSED POLYNOMIALNETWORK CONTROLLERS

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 clas- sification and function approximation as well as con- trol schemes of power systems [8, 11]. Unfortunately, the training speeds for multi-layered networks are extremely slower than those for feedforward networks being com- posed of a single layer of threshold logic units (TLU), and using the perceptron, ADALINE or Hebbian type learn- ing rules [12]. Moreover, these networks converge very weights if all products of up to $k \leq N$ components areto 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 net- work is the highest order of any of the constituent HPUs. Thus output of the k^{th} order SLHPU is a nonlinear func- tion of up to the k^{th} order polynomials. Since it does not have hidden units as in sigma-pi network, reliable sin- gle layer learning rules such as perceptron, ADALINE, or Hebbian type rules can be used. However, to accommo- date all higher-order correlations, the number of weights required increases combinatorially in the dimensionality of the inputs [15]. Limiting the order of the network leadsto a reduction in the classification capability of the net- work.

There have been many approaches that maintain the powerful discrimination capability of higher-order net- works while reducing higher- order terms. For exam- ple, 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 thatare irrelevant to the problem in a single layer of higher- order TLUs [13],[16]. However, since it is often difficultto find the properties of input pattern space a priori, this strategy has limited applications.



a)

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

Pi-Sigma Network Architecture

Figure 8(a) shows the proposed PSN controller for the present work. The input x is an N dimensional vec- tor and x_k is the kth 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 jth summing unit for the ith output, y_i , then,

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

where w_{kji} is an adjustable weight from input x_k to the jth summing unit of the ith output, and Θ_{ji} is an adjustable threshold of the jth summing unit of the ith output. The $\sigma(x)$ denotes the nonlinear activation

function, and is selected as the logistic function, $\sigma(x) = \frac{1}{1+e^{-x}}$, 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 wecan represent the input of a Kth order processing unit by a product of K linear combinations of the input com-ponents [17]. That is why this network is called pi-sigma instead of sigma-pi (Fig. 8(b)). In the SLHPU represen- tation, 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 net- work topology.

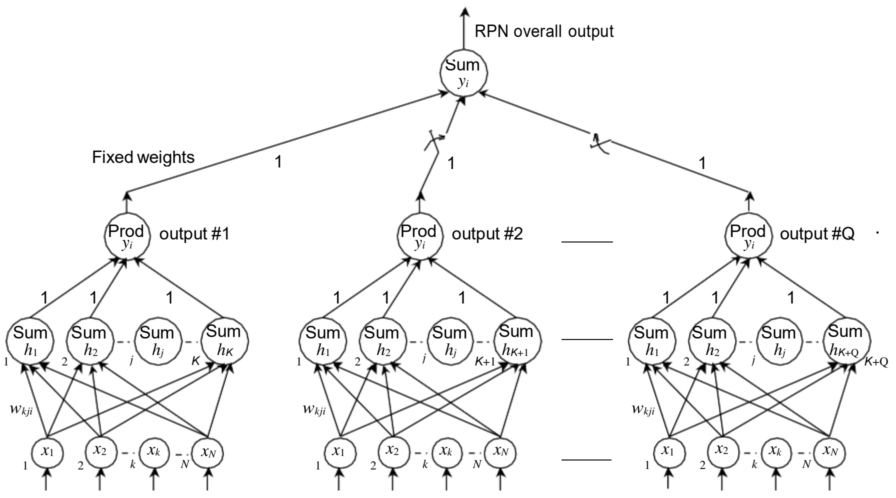


Fig. 9. Proposed Ridge Polynomial Network (RPN) controller architecture

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

Probabilistic Learning Rule Used in thePSN

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 productof 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

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

where superscript p denotes the pth training pattern, d_i and y_i are the ith components of desired and actual out- puts, 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 forweights and threshold for each iteration step as

$$\Delta\Theta_{li} = \eta(d_i - y_i)y_i' \prod h_{ji} \qquad (11)$$

For $x = (x_1, . . . , x_d)^T$ and $w = (w_1, . . . , w_d)^T \in R^d$, the $\langle x, w\rangle$ is denoted as their inner product. For a given compact set $C \subset R^d$, all functions defined on C in the functions. A ridge polyno-mial is a ridge function that can be represented as

$$\sum_{i=0} \sum_{j=1} a_{ij}\langle x, w_{ij}\rangle^i \qquad (13)$$

for some $a_{ij} \in R$ and $w_{ij} \in R^d$.

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

$$\sum \prod$$

$$p(x) = \quad \langle x, w_{ji}\rangle + w_{ji} \quad (14)$$
$$j=1$$
$$i=1$$

The original form of the theorem is more complicated that the one presented here. However, since we are concerned with the existence of a representation of multivari-ate 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,

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

where, η 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^{th} summing unit which is chosen randomly at each step

Ridge Polynomial Network Architecture

The PSN provides only a constrained approximation of a power series. Because of this truncated approxima- tion 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

$$P_i\ x = \prod_{j=1} \langle w_j, x\rangle + w_{j0}\ , \quad i = 1, \ldots, k \quad (16)$$

and $\sigma(\cdot)$ is a suitable linear or nonlinear activation function, $w_j \in R^d$ 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.

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

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

where

i

.

Table 2. Disturbance Cases Applied to Power System Under Study

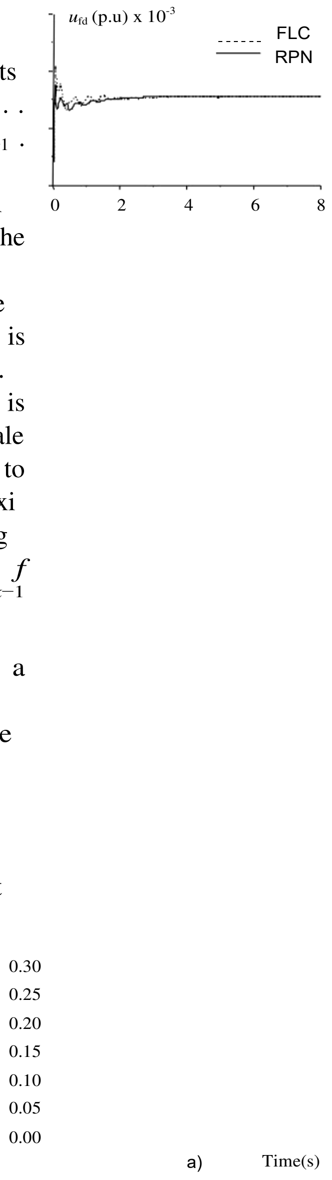
Case No	Δu_e (pu)	At time instant (sec)	Change in Δu_e (pu)	At time instant (sec)
A	0.0010	0.0	–	–
B	0.0010	0.0	0.002	4.0
C	0.0020	0.0	–	–
D	0.0020	0.0	0.002	4.0
E*	0.0015	0.0	0.002	4.0
F*	0.0005	0.0	0.004	4.0

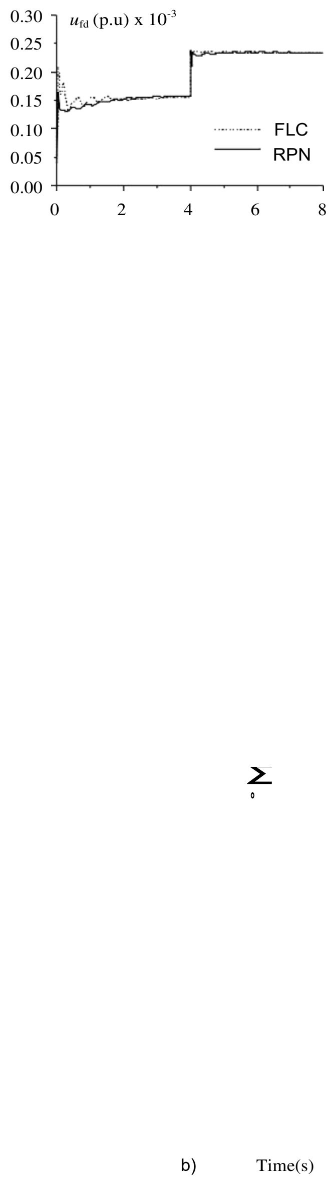
An incremental learning algorithm proceeds as follows. We denote k an algorithmic step at which P_k is added to the network with "predetermined" $P_0, \ldots, P_{k-1}$. That is, with $P_0 \equiv 0$, the function f is approximated by the RPN at the kth step as,

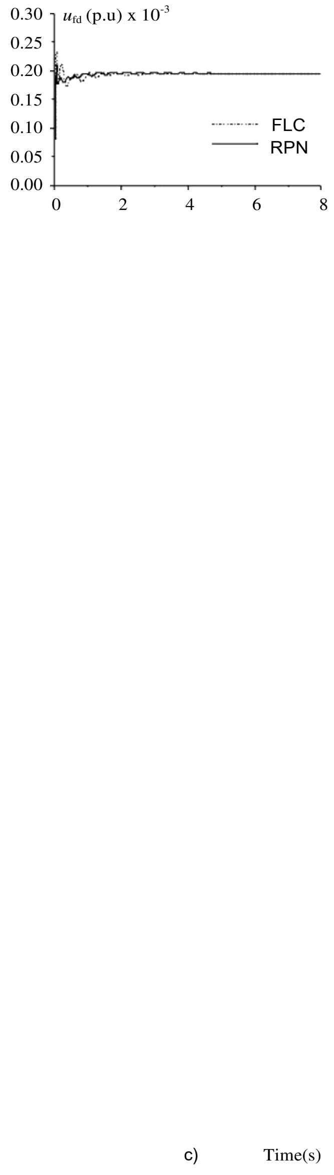
$$f(x) \approx \sigma \sum_{i=0}^{k-1} P_i(x) + P_k(x) \tag{17}$$

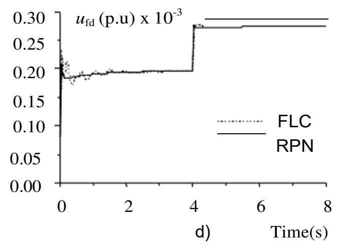
where the weights in $P_0, . . . , P_{k-1}$. are frozen once the k^{th} degree PSN is added. This is equivalent to approximating σ^{-1} (x) f_{k-1} $P_i(x)$ with a k^{th} degree PSN, $P_k(x)$, with linear output units.

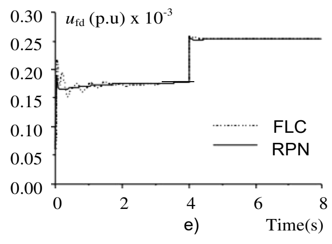
$\sum_{\circ}$ $-$ $i=0$











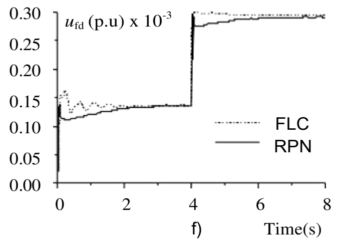


Fig. 12. Time responses of field voltage u_{fd} when FLC and RPN are applied. (A) through (F) are the corresponding cases of disturbance (Table 2).

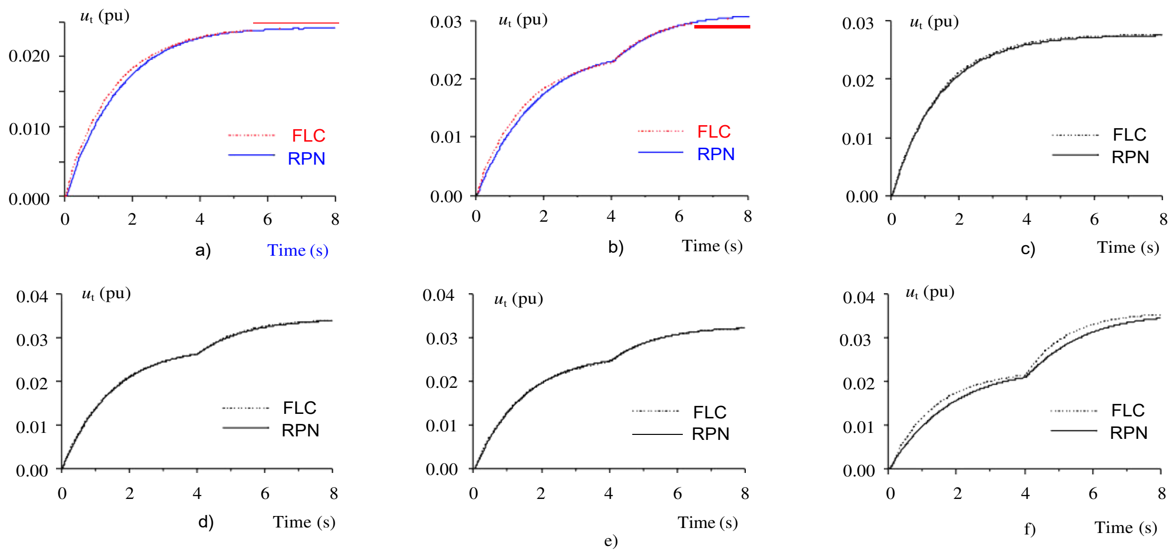


Fig. 13. Time responses of terminal voltage u_t when FLC and RPN are applied. (A) through (F) are the corresponding cases of disturbance (Table 2).

5 APPLICATION AND SIMULATION RESULTS

The designed new PSN and RPN controllers are ap- plied to the power system of Fig. 1 and several cases of interest have been investigated. Some of the cases stud- ied, 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 was 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, ε_{th} . Since the dynamic range of the error between the desired output and the actual algorithm output becomes smaller as learning proceeds, ε_{th} was decreased by a factor of

IJARST
International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

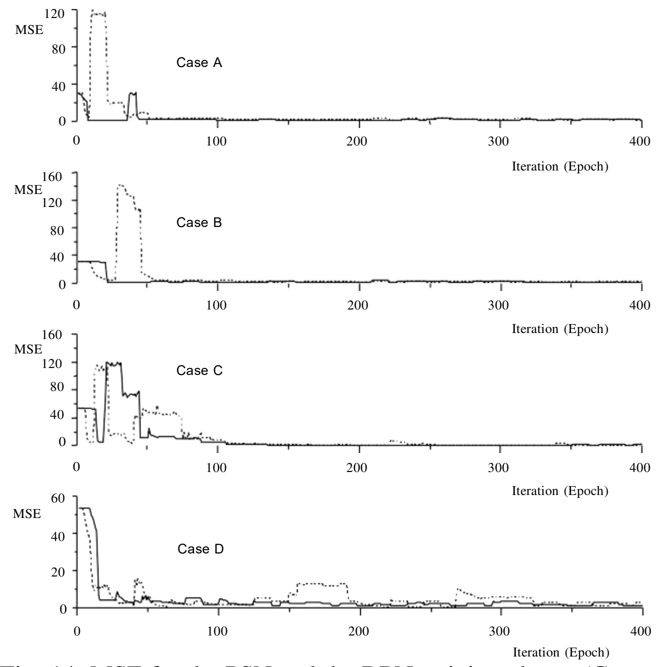


Fig. 14. MSE for the PSN and the RPN training phases (Cases Athrough D).

10 at each addition of a PSN. In case of a RPN used, the training started with a 3rd degree PSN. The initial learning rate and ε_{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) respec-tively, 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 per- formance 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 rel- ative improvement of the system's dynamics regarding the controlled variable is readily seen in Figs. 11 and 13. Apparently, the developed controllers show good conver- gence properties and accuracy for function approxima- tion 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 character- istics 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 sin- gle 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.

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

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 con- trollers for a synchronous machine using polynomial neu-ral 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 con- troller designs and techniques previously developed. It is evident that certain intelligent control applications could increase the efficiency and subsequently make the opera- tion of a power system more economic. It is emphasized that the hardware implementation for such kind of con- trollers is easier than FLC ones and the computational time needed for real-time applications is drastically re- duced. A practical implementation on a microprocessor system could be used as an addition to the existing con- trollers 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 87.5 kVA, 415 V, 4-pole	x_d	1.758 pu	R_a	0.01450 pu
	x_q	0.990 pu	R_{kd}	0.00422 pu
	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}	δ	U_b
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 Adap-tive Generator Excitation Controller Based on Linear OptimalControl, IEEE Trans. on Energy Conversion EC-5 No. 4 (2017), 673–678.

[2] PAPADOPOULOS, D. P. : Excitation Control of Turbogen-erators with Output Feedback, Int. J. Electr. Power & Energy Systems 8 (2016), 176–181.

[3] PAPADOPOULOS, D. P.—SMITH, J. R.—TSOURLIS, G. : Excitation Controller Design of Synchronous Machine with Out- put Feedback Using High and Reduced Order Models, Archiv für Elektrotechnik 72 (2016), 415-426.

[4] ROSS, T. J. : Fuzzy Logic with Engineering Applications, Mc-Graw Hill College Div, 2016.

[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 (2017), 407–413.

[6] HANDSCHIN, E.—HOFFMANN, W.—REYER, F.—STE- PHANBLOME, Th.—SCHLUCKING, U.—WESTERMANN,D.—AHMED, S. S.: A new Method of Excitation Control Based on Fuzzy Set Theory, IEEE Trans. Power Syst. 9 (2014),533–539.

[7] DJUKANOVIC, M. B.—DOBRIJEVIC, D. M.—CALOVIC, M.S.—NOVICEVIC, M.—SOBAJIC, D. J. : Coordinated Stabiliz-ing Control for the Exciter and Governor Loops Using Fuzzy SetTheory and Neural Nets, Int. J. Electr. Power & Energy Syst. 8 (2017), 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 (2013), 233–241.

[9] KARNAVAS, Y. L.—PAPADOPOULOS, D. P. : A Genetic- Fuzzy System for the Excitation Control of a Synchronous Ma- chine, In: ICEM '02, The 15th International Conference in Elec- trical Machines, Bruges, Belgium, 25-28 August, 2012, CD paperRef. No. 204.

[10] HORNICH, K.—STINCHCOMEBE, M.—WHITE, H. : Multi-layer Feedforward Networks are Universal Approximators, Neu- ral Networks 2 (2011), 359–366.

[11] KARNAVAS, Y. L.—PAPADOPOULOS, D. P. : AGC for Autonomous Power Station Using Combined Intelligent Tech-niques, Int. J. Electr. Pow. Syst. Res. 62 (2012), 225–239.

[12] McCLELLAND, J.—RUMELHART, D. : Parallel Distributed Processing. Vol. 1, The MIT Press, 1987.

[13] LEE, C.—MAXWELL, T. : Learning, Invariance, and General- ization in High-Order Neural Network, Applied Optics 26 (23)(2010).

[14] LIPPMANN, R. P. : Pattern Classification Using Neural Net-works, IEEE Communications Magazine (2009), 47–64.

[15] MINSKY, M.—PAPERT, S. : Perceptrons, The MIT Press, 2009.

[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, 2007/.

[17] SHIN, Y.—GHOSH. J. : The pi-sigma Network: an EfficientHigher-Order Neural Network for Pattern Classification andFunction Approximation, In: Proc. of IJCNN, Seattle, 2006 1:13–18.

[18] SHIN, Y.—GHOSH, J. : Efficient Higher-Order Neural Net-works for Function Approximation and Classification, Int. J. Neural Syst. 3 No. 4 (2002), 323–350.