ANFIS Based Automatic Voltage Regulator with Hybrid Learning Algorithm

P. Mitra, ECE Department, Missouri University of Science and Technology, Rolla, MO 65401, USA
S. Maulik, EE Department, Meghnad Saha Institute of Technology, Kolkata, 700150, India
S. P. Chowdhury, University of Manchester, Ferranti Building, PO Box – 88, M60 1QD, UK
S. Chowdhury, University of Manchester, Ferranti Building, PO Box – 88, M60 1QD, UK

***Abstract* — In this paper, the authors propose a design methodology of Adaptive Neuro-Fuzzy Inference System (ANFIS) based Automatic Voltage Regulator (AVR) using hybrid learning algorithm to improve the small-signal performance of power system. Here, a zero order Sugeno fuzzy model is considered, whose parameters are tuned off-line through hybrid learning algorithm. In this algorithm, the least square estimation method is applied for the tuning of linear output membership function parameters and the error backpropagation method is used to tune the nonlinear input membership function parameters. The proposed method is verified through digital simulation with a single machine infinite bus system. It is found that the AVR is performing well in restoring the terminal voltage instantaneously and the damping characteristics of the rotor angle are also improved. The results establish that the design of ANFIS based AVR employing hybrid learning algorithm can be very useful in small signal stability of power system.**

***Index Terms*— AVR, Adaptive Neuro-Fuzzy Inference System, Sugeno-Fuzzy Model, Hybrid Learning Algorithm.**

I. Introduction

In modern power systems the AVR is widely used to sense the generator output voltage and then initiate corrective action by changing the exciter control in the desired direction. The AVR is so designed as to be able to respond quickly to a disturbance endangering the transient stability [1-3]. Also, it should be able to contribute in improving the small signal stability of the system. But, the conventional AVR is a fixed gain device. The gain is intentionally made high to ensure fast response. But, high gain of AVR circuit often becomes detrimental from the point of view of small-signal stability. So, a compromise is to be done. Thus, conventional AVR lacks flexibility. It can not be effective for both transient and small-signal stability simultaneously. To solve these problems Artificial Intelligence (AI) is widely used. Among AI techniques, fuzzy logic control appears to be the most promising, due to its lower computational burden and robustness. Also, a mathematical model is not required to describe the system in fuzzy logic based design. But, the main problem with the conventional fuzzy controllers is that the parameters associated with the membership functions and the rules depend broadly on the intuition of the engineers. If it is required to change the parameters, it is to be done by trial and error only. There is no scientific optimization methodology inbuilt in the general fuzzy inference system. To overcome this, Adaptive Neuro-Fuzzy Inference System (ANFIS) is used. In ANFIS, the parameters associated with a given membership function are chosen so as to tailor the input/output data set. In this paper a two-input Sugeno fuzzy model is used whose equivalent ANFIS architecture is shown in Fig. 1. From the figure it can be observed that the input membership functions of a Sugeno model are non-linear (in this paper Gaussian membership functions are taken), whereas the output membership functions are linear. So, while tuning the parameters from a given input-output data set, it is most effective to use hybrid learning algorithm as it is a combination of least square method and error backpropagation method. Since the input side functions are non-linear, error backpropagation method is a suitable optimization routine, and since the output side functions are linear, least square estimation becomes very effective.

II. Sugeno Fuzzy Model

Unlike Mamdani model [4], Sugeno output membership functions are either linear or constant. If a fuzzy system under consideration has two inputs x and y and one output f, then for a first order Sugeno fuzzy model, a common rule set with two fuzzy if-then rules is as follows:

Rule 1: If x *is* A_1 *and* y *is* B_1, *then* $f_1 = p_1x + q_1y + r_1$
Rule 1: If x *is* A_2 *and* y *is* B_2, *then* $f_2 = p_2x + q_2y + r_2$

For a zero-order Sugeno model, the output level is a constant ($p_i = q_i = 0$). The output level f_i of each rule is weighted by the firing strength w_i of the rule. For example, for an AND rule with *Input 1* = x and *Input 2* = y, the firing strength is

$$w_i = AND\ method\ (A_1(x), B_1(y))$$

where A_1 and B_1 are the membership functions for *Input 1* and *Input 2* respectively. The final output of the system is the weighted average of all rule outputs, computed as

$$\text{Final Output} = \frac{\sum_{i=1}^{n} w_i f_i}{\sum_{i=1}^{n} w_i} \quad (1)$$

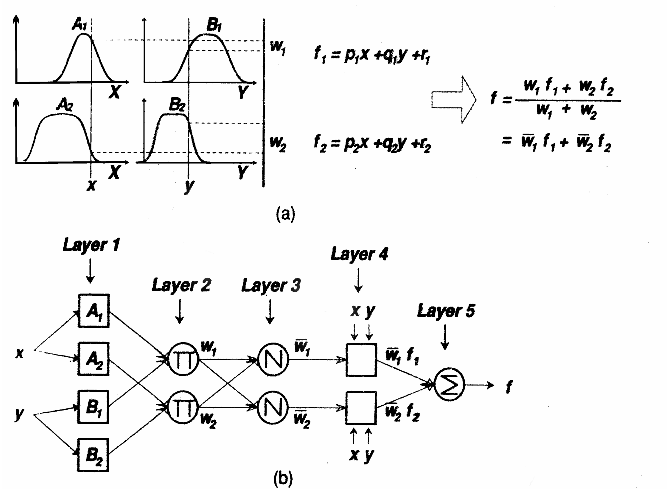

Fig. 1: (a) A two-input first-order Sugeno fuzzy model with two rules; (b) Equivalent ANFIS architecture [5]

III. ANFIS Architecture

Fig. 1(a) illustrates the reasoning mechanism for the Sugeno model discussed above while the corresponding ANFIS architecture is as shown in Fig. 1(b), where nodes of the same layer have similar functions [5]. The output of i^{th} node in layer l is denoted as $O_{l,i}$.

Layer 1: Every node i in this layer is an adaptive node with a node function

$$O_{l,i} = \mu_{A_i}(x) \quad \text{for } i = 1, 2, \text{ or}$$
$$O_{l,i} = \mu_{B_{i-2}}(y) \quad \text{for } i = 3,4 \qquad (2)$$

where x (or y) is the input to node i and A_i (or B_{i-2}) is a linguistic label ("small" or "large") associated with the node. Here the membership function for A (or B) can be any parameterized membership function. In this paper, generalized Gaussian membership function is taken as follows

$$\mu_A(x) = \exp[-(\frac{x-c_i}{a_i})^2] \qquad (3)$$

where $\{c_i, a_i\}$ is the parameter set. These are called premise parameters.

Layer 2: Every node in this layer is a fixed node labeled $\prod$, whose output is the product of all the incoming signals.

$$O_{2,i} = w_i = \mu_{A_i}(x) \times \mu_{B_i}(y) \quad i=1,2 \qquad (4)$$

Layer 3: Here, the i^{th} node calculates the ratio of the i^{th} rule's firing strength to the sum of all rule's firing strengths.

$$O_{3,i} = \overline{w}_i = \frac{w_i}{w_1 + w_2} \quad i=1,2 \qquad (5)$$

Layer 4: Every node i in this layer is an adaptive node with a node function

$$O_{4,i} = \overline{w}_i f_i = \overline{w}_i (p_i x + q_i y + r_i) \qquad (6)$$

where $\overline{w}_i$ is a normalized firing strength from layer 3 and $\{p_i, q_i, r_i\}$ is the parameter set of the node. These parameters are referred to as consequent parameters.

Layer 5: The single node in this layer is a fixed node labeled $\sum$, which computes the overall output as the summation of all incoming signals:

$$O_{5,1} = \sum_i \overline{w_i} f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \qquad (7)$$

IV. Hybrid Learning Algorithm

The Hybrid Learning Algorithm [5] is a combination of least square and backpropagation method. In the least square method, the output of a model y is given by the parameterized expression

$$y = \theta_1 f_1(\mathrm{u}) + \theta_2 f_2(\mathrm{u}) + \ldots\ldots + \theta_n f_n(\mathrm{u}) \qquad (8)$$

where $\mathbf{u} = [u_1, \ldots\ldots, u_n]^T$ is the models input vector, $f_1, \ldots\ldots, f_n$ are known functions of u, and $\theta_1, \ldots\ldots, \theta_n$ are unknown parameters to be optimized. To identify these unknown parameters θ_i , usually a training data set of data pairs $\{(u_i, y_i), i = 1, \ldots, m\}$ is taken; substituting each data pair in (8) a set of linear equations is obtained, which can be written as

$$\mathbf{A\theta = y} \qquad (9)$$

in matrix form. Where $\mathbf{A}$ is a $m \times n$ matrix

$$\mathbf{A} = \begin{bmatrix} f_1(u_1) & \ldots & f_n(u_1) \\ : & : & : \\ f_1(u_m) & \ldots & f_n(u_m) \end{bmatrix} \qquad (10)$$

$\boldsymbol{\theta}$ is an $n \times 1$ unknown parameter vector

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ : \\ \theta_n \end{bmatrix} \qquad (11)$$

$\mathbf{y}$ is an $m \times 1$ output vector

$$\mathbf{y} = \begin{bmatrix} y_1 \\ : \\ y_m \end{bmatrix} \qquad (12)$$

Since generally $m>n$, instead of exact solution of (9) an error vector $\mathbf{e}$ is introduced to account for the modeling error, as

$$\mathbf{A\theta + e = y} \qquad (13)$$

and searched for $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}$ which minimizes sum of squared error

$$E(\theta) = \sum_{i=1}^{m} (y_i - a_i^T \theta)^2 = e^T e \tag{14}$$

where $E(\theta)$ is called the objective function. The squared error in (14) is minimized when $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}$, called Least Squares Estimator (LSE) that satisfies the normal equation

$$\mathbf{A}^T \mathbf{A} \hat{\boldsymbol{\theta}} = \mathbf{A}^T \mathbf{y} \tag{15}$$

If $\mathbf{A}^T \mathbf{A}$ is non singular, $\hat{\boldsymbol{\theta}}$ is unique and is given by

$$\hat{\boldsymbol{\theta}} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{y} \tag{16}$$

In case of backpropagation learning rule the central part concerns how to recursively obtain a gradient vector in which each element is defined as the derivative of an error measure with respect to a parameter. Assuming that a given feedforward adaptive network has *L* layers and layer *l* has *N(l)* nodes, then the output function of node *i* in layer *l* can be represented as $x_{l,i}$ and $f_{l,i}$ respectively. For the node function $f_{l,I}$ we can write:

$$x_{l,i} = f_{l,i}(x_{l-1,1}, \ldots\ldots, x_{l-1,N(l-1)}, \alpha, \beta, \gamma, \ldots.) \tag{17}$$

where α, β, γ, etc. are the parameters of this node. Assuming that the given training data set has *P* entries, an error measure can be defined for the p^{th} $(1 \le p \le P)$ entry of the training data set as the sum of squared errors:

$$E_p = \sum_{k=1}^{N(L)} (d_k - x_{L,k})^2 \tag{18}$$

where d_k is the k^{th} component of the p^{th} desired output vector and $x_{L,k}$ is the k^{th} component of the actual output vector produced by presenting the p^{th} input vector to the network. The task here is to minimize an overall error measure, which is defined as $E = \sum_{p=1}^{P} E_p$. The basic concept in calculating the gradient vector is to pass a form of derivative information starting from the output layer and going backward layer by layer until the input layer is reached. To facilitate the discussion the error signal $\in_{l,i}$ is defined as

$$\in_{l,i} = \frac{\partial E_p}{\partial x_{l,i}} \tag{19}$$

This is actually ordered derivative and is different from ordinary partial derivative. For i^{th} output node (at layer *L*)

$$\in_{L,i} = \frac{\partial E_p}{\partial x_{L,i}} \tag{20}$$

$$\therefore \in_{L,i} = -2(d_i - x_{L,i}) \tag{21}$$

For the internal node at the i^{th} position of layer *l*, the error signal can be derived iteratively by the chain rule:

$$\begin{aligned} \in_{l,i} = \frac{\partial E_p}{\partial x_{l,i}} &= \sum_{m=1}^{N(l+1)} \frac{\partial E_p}{\partial x_{l+1,m}} \times \frac{\partial f_{l+1,m}}{\partial x_{l,i}} \\ &= \sum_{m=1}^{N(l+1)} \in_{l+1,m} \times \frac{\partial f_{l+1,m}}{\partial x_{l,i}} \end{aligned} \tag{22}$$

The gradient vector is defined as the derivative of the error measure with respect to each parameter. If α is a parameter of the i^{th} node at layer *l*, we have

$$\frac{\partial E_p}{\partial \alpha} = \frac{\partial E_p}{\partial x_{l,i}} \times \frac{\partial f_{l,i}}{\partial \alpha} = \in_{l,i} \frac{\partial f_{l,i}}{\partial \alpha} \tag{23}$$

The derivative of the overall error measure *E* with respect to α is

$$\frac{\partial E}{\partial \alpha} = \sum_{p=1}^{P} \frac{\partial E_p}{\partial \alpha} \tag{24}$$

Accordingly, for simplest steepest descent without line minimization, the update formula for generic parameter α is

$$\Delta\alpha = -\eta \frac{\partial E}{\partial \alpha} \tag{25}$$

in which η is the learning rate. So, for parameter α it may be written that,

$$\begin{aligned} \alpha_{new} &= \alpha_{old} + \Delta\alpha \\ &= \alpha_{old} - \eta \frac{\partial E}{\partial \alpha} \end{aligned} \tag{26}$$

In this type of learning, the update action occurs only after the whole set of training data pair is presented. This process of presentation of whole set of training data pair is called epoch.

It is assumed that *'S'* is the total set of parameters and *'S_1'* and *'S_2'* are the sets of input and output parameters respectively. For hybrid learning algorithm, each epoch consists of a forward pass and a backward pass. In the *forward pass*, when a vector of input data pair is presented, the node outputs of the system are calculated layer by layer till the corresponding row in the matrices **A** and **y** of equation (9) are obtained. The process is repeated for all the training data pair to form the matrices **A** and **y** completely. Then the output parameters of set S_2 are calculated according to the equation (16). After this, the error measure for each training data pair is to be calculated. The derivative of those error measures w.r.t. each node output are calculated following equations (20) and (22). Thus the error signal is obtained. In the *backward pass*, these error signals propagate from the output end towards the input end. The gradient vector is found for each training data entry. At the end of the backward pass for all training data pairs, the input parameters are updated by steepest descent method as given by equation (26).

V. Design of ANFIS Based AVR

A step-by-step method of designing ANFIS-based AVR is presented as follows:

A) Choice of input variable: In this step it is decided which state variables must be taken as the input signals to the controller. In this paper, deviation in terminal voltage (e) and its derivative ($\dot{e}$) are taken as input signals of the ANFIS-based AVR.

B) Choice of linguistic variables: The linguistic values may be viewed as labels of fuzzy sets. In this paper, seven linguistic variables for each of the input variables are used to describe them. These are, LP (Large Positive), MP (Medium Positive), SP (Small Positive), ZE (Zero), SN (Small Negative), MN (Medium Negative), LN (Large Negative).

C) Choice of membership functions: In this design, Gaussian membership functions are used to define the degree of membership of the input variables.

D) Choice of fuzzy model: A zero order Sugeno fuzzy model is chosen for ANFIS-based AVR.

E) Preparation of training data pair: In preparing the training data pair, the data should be representative of different kinds of disturbance situations, such that the designed AVR can be used for highest flexibility and robustness. In this paper, the training data pair is prepared by simulating a single-machine system with conventional AVR under a broad range of small and large disturbances.

F) Optimization of unknown parameters: Using the training data matrix, the unknown parameters of the Gaussian input membership functions (center (c_i) and spread (a_i)) and the output constants of each rule (r_i) of zero order Sugeno fuzzy model are optimized. Initially, it is assumed that the input membership functions are symmetrically spaced over the entire universe of discourse. Accordingly some initial values for the center and the spread of each input membership function are assumed, whereas, in case of output for each rule, all initial values are assumed to be zero. Then, the input parameters are optimized by error backpropagation algorithm and the output constants are optimized by least square method. The tuned AVR thus obtained is then used to obtain a stable output.

VI. Test System

The Single-Machine-Infinite-Bus test system has a generating station with four 555 MVA, 24 kV, 60 Hz units connected to an infinite bus through a transformer (reactance 0.15 p.u.) and two lines (reactances 0.5 and 0.93 p.u.) (Fig. 2). The small signal stability characteristics are analyzed about the steady state following the loss of tie-line 2.

VII. Results

Just as an example of successful optimization, Table 1 is given which shows the values of the centers (c_i) and spreads (a_i) of the input Gaussian membership functions of input 1 before and after optimization. The output constants of the zero order Sugeno model are also optimized in a similar manner. Fig. 3 shows the control surface of the tuned fuzzy AVR. Now, the performances of this ANFIS based AVR for single-machine system is demonstrated by Fig. 4 and Fig. 5. Comparing it with ordinary fuzzy AVR, it is found that the terminal voltage characteristics is almost as good as the fuzzy AVR, whereas the rotor angle oscillation damps a little earlier in case of ANFIS based AVR than ordinary fuzzy AVR.

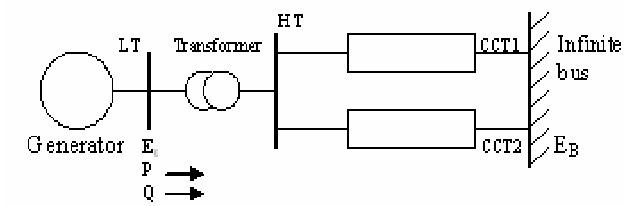

Fig. 2: Single Machine Infinite Bus Test System

		Before Optimization	After Optimization
LN	a_i	0.8366	0.1284
	c_i	-0.1067	-0.11
MN	a_i	0.8366	0.1114
	c_i	0.09027	0.06561
SN	a_i	0.8366	0.1029
	c_i	0.2873	0.29
ZE	a_i	0.8366	0.1098
	c_i	0.4843	0.4746
SP	a_i	0.8366	0.08301
	c_i	0.6813	0.6799
MP	a_i	0.8366	0.09175
	c_i	0.8733	0.8711
LP	a_i	0.8366	0.08121
	c_i	1.075	1.076

Table 1: Membership function parameters before and after optimization for ANFIS AVR

VIII. Conclusion

The paper introduces a new algorithm known as hybrid learning algorithm in designing adaptive neuro-fuzzy inference system based Automatic Voltage Regulator. The hybrid learning algorithm, which is a combination of least square estimation and error backpropagation, is used in this paper in order to tune the parameters of a zero order Sugeno fuzzy model. The step-by-step design procedure of the ANFIS based AVR is presented in this paper. The time-domain simulations reveal that this ANFIS based AVR can be very effective in small-signal stability of power system.

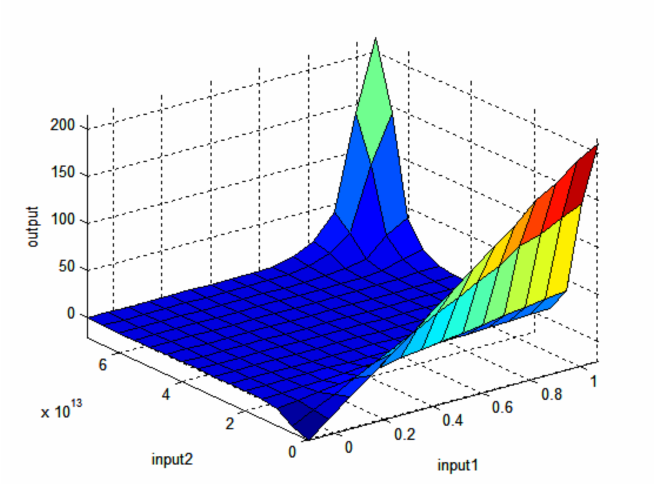

Fig. 3: Control Surface of ANFIS Based AVR

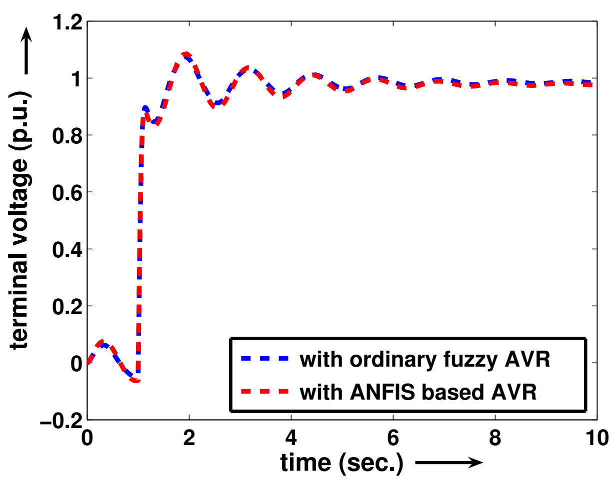

Fig. 4: Terminal voltage characteristics with ANFIS-based AVR and Conventional Fuzzy AVR

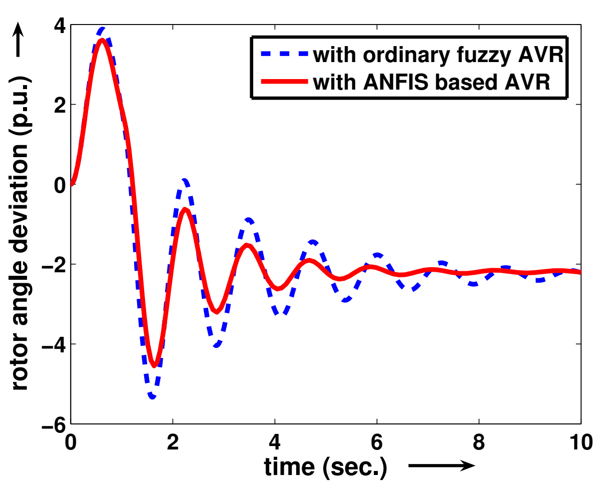

Fig. 5: Rotor angle characteristics with ANFIS-based AVR and Conventional Fuzzy AVR

IX. REFERENCES

[1] Kundur P.,"Power System Stability and Control", McGraw-Hill, New York, 1994.

[2] Kundur P., Klein M., Rogers G. J. and Zywno M. S., "Application of Power System Stabilizers for Enhancement of Overall System Stability", IEEE Trans. on Power Systems, Vol. 4, No. 2, pp. 614-626, May 1989.

[3] Anderson P. M. and Fouad A.A., "Power System Control and Stability", Iowa State University Press, Ames, IA, 1977.

[4] Ross T., "Fuzzy Logic with Engineering Applications", McGraw-Hill International Editions, Electrical Engineering Series, 1997.

[5] Jang R., Sun C., and Mijutani E., "Neuro-Fuzzy and Soft Computing", Pearson Education, 2004.

BIOGRAPHIES

Pinaki Mitra was born in 1974. He graduated in Electrical Engineering from Jadavpur University in 1997 and his Post Graduation was in Electrical Machines from the same University. At present he is a PhD student of the Real-Time Power and Intelligent Systems Laboratory, ECE Department, Missouri University of Science and Technology, Rolla, MO, USA. His research interest is application of computational intelligence techniques in power systems. He is a student member of IEEE (USA). Email: pinakimitra@yahoo.com

S. Maulik received her B. Tech and M.E. degrees in Electrical Engineering from Bhilai Institute of Technology and Jadavpur University, India in the year 2002 and 2005 respectively. Presently she is a lecturer of the EE Department of Meghnad Saha Institute of Technology, Kolkata, India. Email: sayantimaulik@yahoo.com

Prof. S. P. Chowdhury received his BEE, MEE and PhD in 1987, 1989 and 1992 respectively. In 1993, he joined Jadavpur University, India as Lecturer. He was promoted to Reader and Professor in 1998 and 2006 respectively. He visited Brunel University, UK in 2005 and 2006. Currently he is Senior Academic Visitor of The University of Manchester. He has published two books and over 100 papers in power systems. He is Fellow of the IET (UK) with C.Eng. IE (I) and the IETE(I) and Member of IEEE (USA). He is a member of Membership and Regions Board (MRB), MRB Finance Committee and Council of the IET (UK). Email: spchowdhury63@yahoo.com.

Dr. S. Chowdhury received her BEE and PhD in 1991 and 1998 respectively. She joined M/S M. N. Dastur & Co. Ltd as Electrical Engineer and Women's Polytechnic, Kolkata, India as Lecturer in 1991 and 1998 respectively. She was promoted to Senior Lecturer in 2006. She visited Brunel University, UK several times on collaborative research program. She has published two books and over 75 papers in power systems area. She is a Member of the IET (UK) and IE (I) and Member of IEEE (USA). Currently she is Academic Visitor of The University of Manchester, UK. Email: sunetr69@yahoo.com.