ANN APPLICATIONS TO LOAD FORECASTING

- There is a growing tendency towards unbundling the electricity system. This is continually confronting the different sectors of the industry (generation, transmission, and distribution) with increasing demand on planning management and operations of the network.
- The operation and planning of a power utility company requires an adequate model for electric power load forecasting.
- Load forecasting plays a key role in helping an electric utility to make important decisions on power, load switching, voltage control, network reconfiguration, and infrastructure development.
- Methodologies of load forecasts can be divided into various categories that include short-term forecasts, medium-term forecasts, and long-term forecasts.
- Short-term forecasting, gives a forecast of electric load one hour ahead of time. Such forecast can help to make decisions aimed at preventing imbalance in the power generation and load demand, thus leading to greater network reliability and power quality. Many methods have been used for load forecasting in the past. These include statistical methods such as regression and similar-day approach, fuzzy logic, expert systems, support vector machines, econometric models, end-use models, etc.
- The neural network is trained on input data as well as the associated target values. The trained network can then make predictions based on the relationships learned during training. A real life case study of the power industry in Nigeria was used in this work.

Implementation for Short Term Load Forecasting The data used in this study came from the Greek power system during the period 2013–2017 and refer to hourly load values. To make a more accurate forecast, weather data, such as temperature, were used in addition to the historical data of the loads. Data is separated into training and test sets in a ratio of 80% to 20%.

The improved neural network model consists of the following input variables:

- Hour: The time of day for which the load forecast will be made. The time is expressed as an integer with values ranging from 0 to 23.
- Week Day: It's a characteristic coding to decide the day of the week. The coding is done with integers ranging from 1 to 7, with 1 denoting Sunday, 2 denoting Monday, and so on.
- Holiday: Binary coding is used to indicate whether a day is a holiday or a working day
- Temperature: The hourly value (in Celsius) of the temperature of the day for which the load is forecast.
- D-1 Load: The load value of the day preceding the one for which prediction is made, at the corresponding time.
- D-7 Load: The value of the load at the corresponding time on the same day of the previous week.

H-1 Load: The value of the previous hour's load on which the forecast is based.
The architecture of the MLP neural network that was used to predict the hourly value of the load is shown in Figure 1. An input layer, a hidden layer, and an output layer represent the three layers of a neural network. Seven neurons make up the input level. Each neuron is associated with one of the variables listed above. There are 100 neurons in the hidden layer. The value 100 was chosen experimentally as it was found to produce better predictive values by dramatically reducing error. As can be seen from the

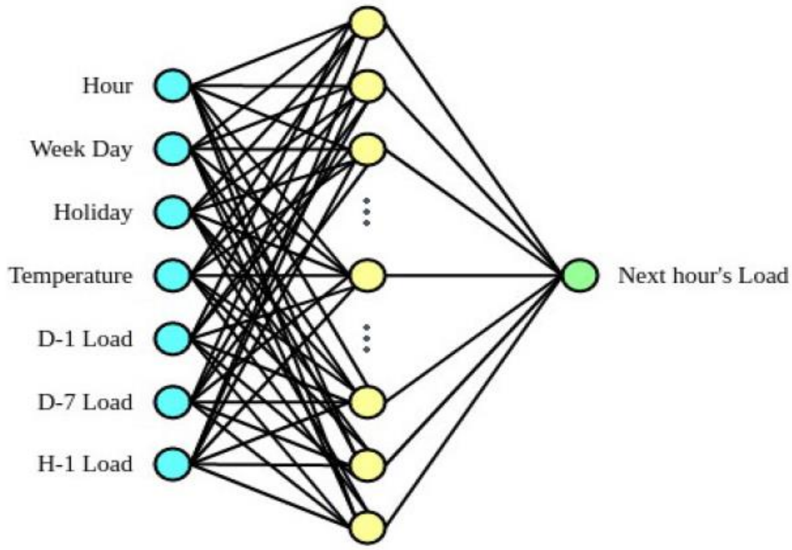


Figure 1. The structure of the proposed MLP neural network.

literature, neural networks with a single hidden layer address the STLF problem quite accurately. The output layer is composed of a single neuron and refers to the hourly load value for which the prediction is developed.
The pre-processing techniques for the data input to the neural network are of particular interest when developing the current prediction model. The MSE, MAE and MAPE metrics are used to assess and compare the various scaling methods for the input data.

Load demand pattern: A broad spectrum of factors affects the system's load level such as trend effects, cyclic-time effects, and weather effects, random effects like human activities, load management and thunderstorms. Thus the load profile is dynamic in nature with temporal, seasonal and annual variations. In this study, we develop a system as shown in Fig. 1 with inputs parameters such as past 24 h load, temperature, humidity, wind speed, sky condition (cloud cover), sunset (daylight time), season (month) and day of the week to forecast 24 ahead load demands (output) for the west of Iran using artificial neural networks.

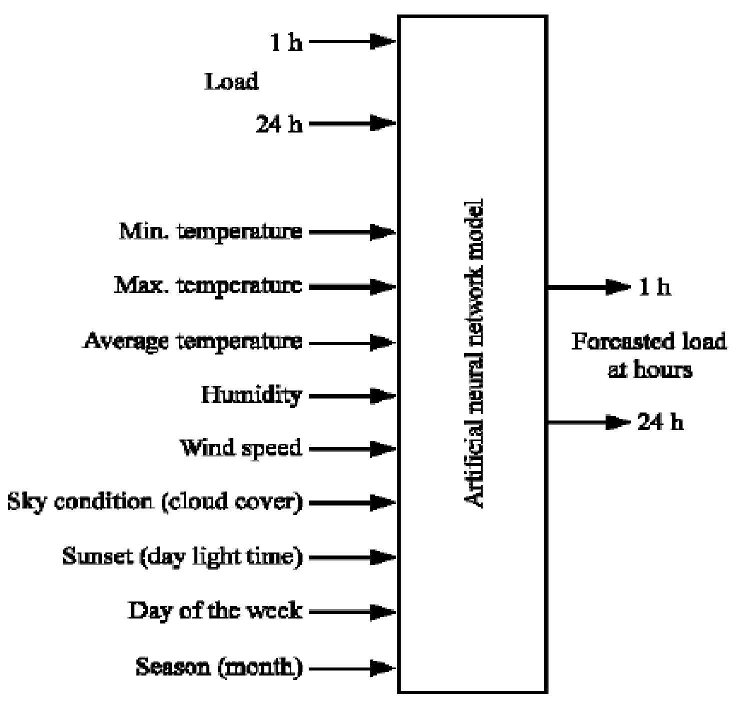


Fig. 1: Input-output schematic of system

Computational intelligence models: The three models of neural networks are selected among the main network architectures used in engineering. The basis of all models is neuron structure. These neurons act like parallel processing units as shown in Fig. 2, where X„ . . ., X are inputs and W„ . . ., W are input weights.

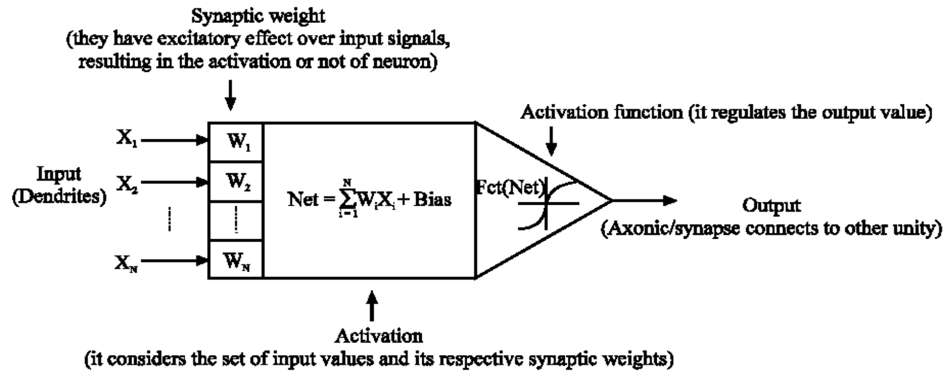


Fig. 2: Neuron model and excitation function

Multi-Layer Perceptron (MLP): This is perhaps the most popular network architecture in use today. Its units each perform a biased weighted sum of their inputs and pass this activation level through a transfer function to produce their output and the units are arranged in a layered feed forward topology. The network thus has a simple interpretation as a form of input-output model, with the weights and thresholds (biases) the free parameters of the model. Such networks can model functions of almost arbitrary complexity with the number of layers and the number of units in each layer, determining the function complexity. Important issues in Multilayer Perceptron design include specification of the number of hidden layers and the number of units in these layers. Once the number of layers and number of units in each layer, has been selected, the network's weights and thresholds must be set so as to minimize the prediction error made by the network. This is the role of the training algorithms. The best known example of a neural network training algorithm is back propagation. Modern second-order algorithm such as conjugate gradient descent and Levenberg-Marquardt are substantially faster for many problems, but Back propagation still has advantages in some circumstances and is the easiest algorithm to understand. With this background we designed and trained the network as follows: the three-layer network with Sigmoid transfer function for hidden layer and linear transfer function for output layer has been selected. The MLP structure is shown in Fig. 3.

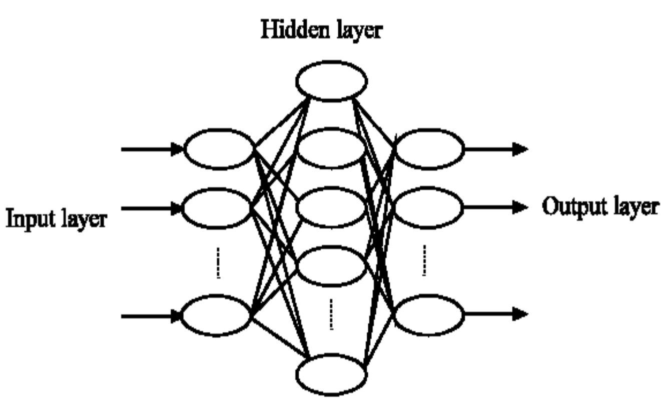


Fig. 3: Three layer MLP

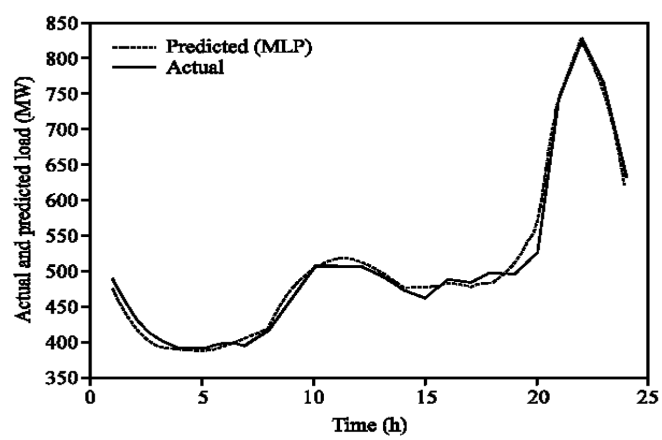


Fig. 4a: Comparison of 24 h ahead load forecasting using MLP

Back propagation training algorithms are often too slow for practical problems, so we can use several high performance algorithms that can converge from ten to one hundred times faster than back propagation algorithms. These faster algorithms fall into two main categories: heuristic technique (variable learning rate back propagation, resilient back propagation) and numerical optimization techniques (conjugate gradient, quasi- Newton, Levenberg-Marquardt). We tried several of these algorithms to get the best result. Levenberg-Marquardt is the fastest algorithm but as the number of weights and biases in the network increase, the advantage of this algorithm decrease, so we tried another algorithm which perform well on function approximation and converge rather fast. From these algorithms, conjugate gradient was suitable for our purpose. Neural networks generally provide improved performance with the normalized data. The use of original data as input to neural network may cause a convergence problem. All the data sets were therefore, transformed into values between -1 and 1 through dividing the difference of actual and minimum values by the difference of maximum and minimum values subtracted by 1. At the end of each algorithm, outputs were denormalized into the original data format for achieving the desired result. From one initial condition the algorithm converged to global minimum point, while from another initial condition the algorithm converged to a local minimum so it is better to try several different initial conditions in order to ensure that optimum solution has been obtained. Training goal for the networks was set to 10 ‘. Finding appropriate architecture needs trial and error method. Networks were trained for a fixed number of epochs. By this way, we found that 1 neurons for hidden layer at 500 epochs produce good result. Comparison of 24h a head load forecasting with MLP and exact load is shown in Fig. 4.

Elman recurrent networks: Elman networks are just like back propagation networks, with addition of a feedback connection from the output of the hidden layer to its input. This feedback path allows Elman networks to learn to recognize and generate temporal patterns, as well as spatial patterns. This makes Elman networks useful in such areas as signal processing and prediction where time plays a dominant role. Because Elman networks are an extension of two-layer Sigmoid/linear architecture, they inherit the ability to fit any input/output function with a finite number of discontinuities. They are also able to fit temporal patterns, but may need many neurons in the recurrent layer to fit a complex function. Also because of the more complex architecture of the recurrent model, there is a significant increase in training time compared with the MLP model. Figure 5 shows an Elman structure, where I„ ..., I, are inputs and O„ ..., O are outputs.

For finding the appropriate architecture of Elman recurrent network, previous steps at MLP designing was followed and found that 11 neurons for hidden layer at 1000 epochs produce good result. Comparison of 24 h ahead load forecasting with ERNN and exact load is shown in Fig. 6.

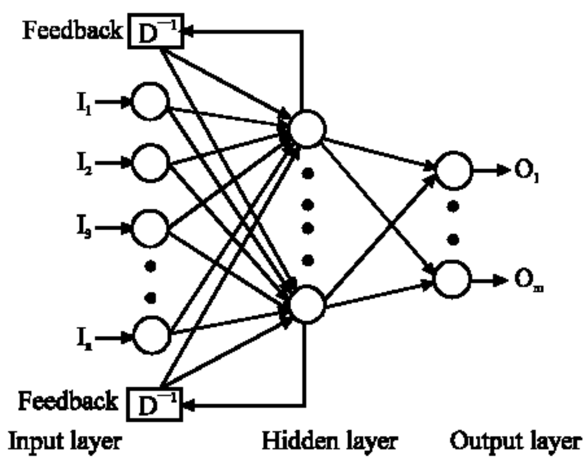


Fig. 5: Architecture of Elman recurrent network

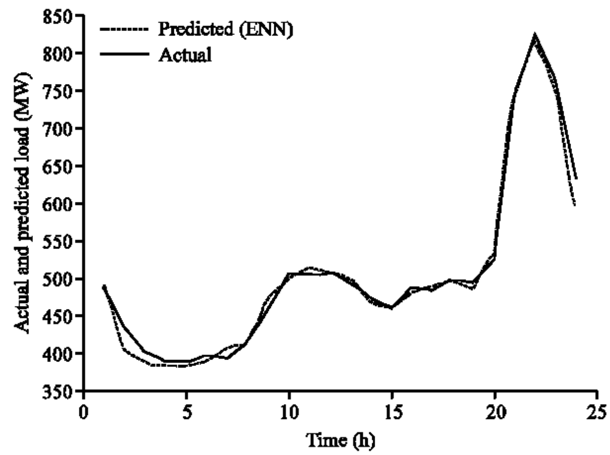
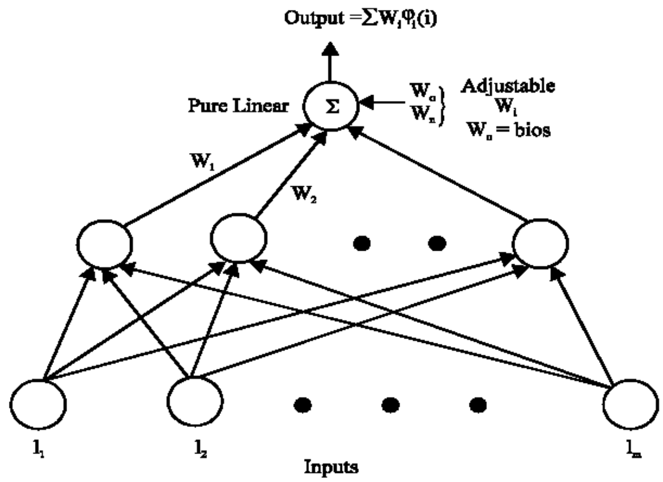


Fig. 6a: Comparison of 24 h ahead load forecasting using ERNN and exact load for 9-May-2005

Radial Basis Function (RBFN): Radial basis function was then applied to each center. There is one radial Gaussian function for each hidden unit which simulates the effect of overlapping and locally tuned receptive fields. The activation function of hidden nodes is radially symmetric in input space, the magnitudes of activation given a particular record is decreasing function of the distance between the input vector of the record and the center of the basis function. The role of hidden units is to perform a non-linear transformation of the input space A. Radial Basis Function Network is a hybrid learning neural network. It's a two layer fully-connected network with an input layer which performs no computation. It uses a linear transfer function for the output units and Gaussian function (Radial basis function) for input units (Hagan *et al.,* 1996; Powell, 1992; Zurada, 1992). Learning in the hidden layer is performed by using an unsupervised method, the K-mean algorithm. First, the user must choose a number of centers and this number will correspond to the number of neurons in hidden layer. The K-means algorithm is used to position the centers in the best way, so that each presented record is attached to its nearest center (or cluster). As it is an unsupervised learning method, only the inputs data are presented to K-means algorithm. Learning in the output layer is performed by computing a linear combination of activation of the basis functions, parameterized by weights W between hidden and output layer. Radial basis networks may require more neurons than standard feed-forward Back propagation networks, but often they can be designed in a fraction of the time it take to train standard feed-forward networks. They work best when many training vectors are available.

A Generalized Regression Neural Network (GRNN) is often used for function approximation. It has been shown in fig.7, that, given a sufficient number of hidden neurons, GRNNs can approximate a continuous function to an arbitrary accuracy. Probabilistic Neural Networks (PNN) can be used for classification problems. Their design is straightforward and does not depend on training. These networks generalize well. Figure.7 shows RBFN common structure, where I_i. . ., I_m are inputs.



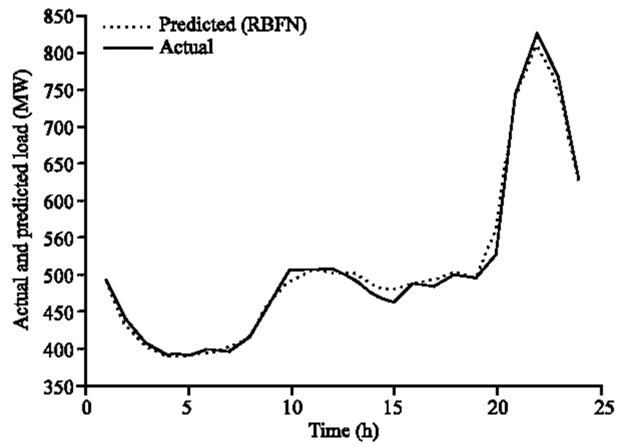


Fig 7 : RBFN structure

Fig.8: Comparison of 24 h ahead load forecasting using

Designing a radial basis function network often takes much less time than training a Sigmoid/linear network (Khan and Ondrusek, 2000) in RBFN, neurons increase till error goal or maximum number of neurons reach. The good result obtained with error goal of 10^{-5} ‘and maximum number of neurons equal to 19. Comparison of 24h ahead load forecasting with RBFN and exact load is shown in Fig. 8.

The assessment of the prediction performance of the different soft computing models was done by quantifying the prediction obtained on an independent data set. The Mean Absolute Percentage Error (MAPE) were used to study the performance of the trained forecasting models for the testing years. MAPE is defined as follows:

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{P_{actual\ i} - P_{predicted\ i}}{P_{actual\ i}}\right]$$

Where P actual,i , is the actual load on day I and Predicted I , is the forecast value of the load on that day. Where N represents the total number of data (hours). The Mean Absolute Percentage Error (MAPE) results (Table 1) for three important architectures of neural networks i.e., MultiLayer Perceptron (MLP), Elman Recurrent Neural Network (ERNN) and Radial Basis Function Network (RBFN) and their optimal structures are shown in Table 1. It has been observed that error associated with each method depends on several factors such as the homogeneity in data, the choice of model, the network parameters and finally the type of solution. The learning method for MLP and ERNN were based on back propagation algorithm. As the learning process is time-consuming in back propagation.

Table 1: Comparison of optimal structures and MAPE index for three types of neural network

Neural Network	MLP	ERNN	RBFN
No. of hidden layer	1	1	1
No. of hidden neuron	17	11	19
Activation function used in hidden layer	Tan-sigmoid	Tan-sigmoid	Gaussian
Activation function used in output layer	Pure linear	Pure linear	Pure linear
MAPE	0.38	0.76	0.17

❖ LOAD FREQUENCY CONTROL IN A SINGLE AREA POWER SYSTEM BY ARTIFICIAL NEURAL NETWORK

In a power system, load-frequency control (LFC) obtains an essential role to allow power exchanges and to supply better conditions for the electricity trading. Also, time delays in such systems can reduce system performance and even cause system instability on frequency or other parameters. The dynamic behavior of many power systems and resulted in industrial loads heavily depends on disturbances and in particular on changes in the operating point. Load frequency control in power systems is very important in order to supply reliable electric power with good quality. The goal of the LFC is to maintain zero steady state errors in a multi area interconnected power system. In addition, the power system should fulfill the proposed dispatch conditions. Power systems are divided into control areas connected by tie lines. All generators are supposed to constitute a coherent group in each control area. From the experiments on the power system, it can be seen that each area needs its system frequency to be controlled.

Basically, single area power system consists of a governor, a turbine, and a generator with feedback of regulation constant. System also includes step load change input to the generator. Simple block diagram of a single area power system with the controller is shown in Figure 1.

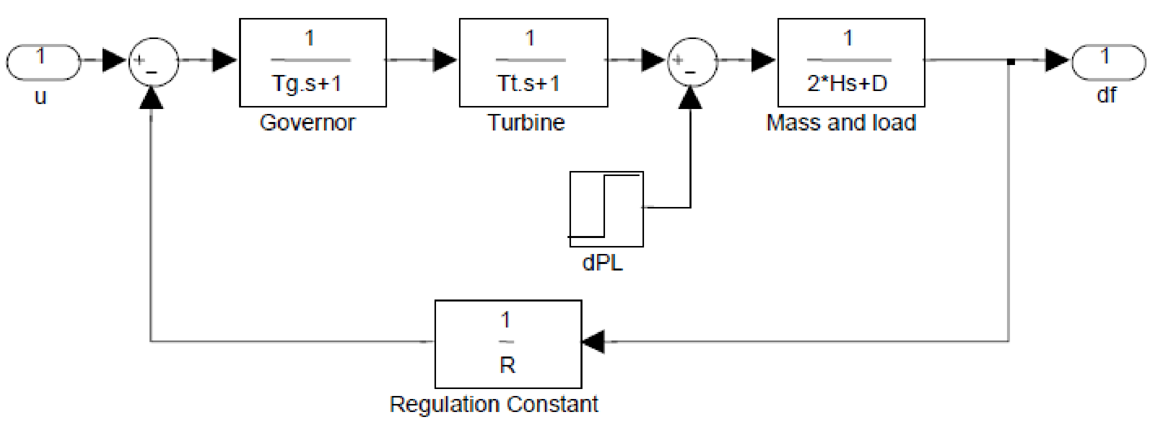


Fig. 1. A single area power system with the controllers (ΔPL = 0.01)

A lot of studies have been made in the past about the load frequency control. In the literature, some control strategies have been suggested based on the conventional linear control theory. These controllers may be unsuitable in some operating conditions due to the complexity of the power systems such as nonlinear load characteristics and variable operating points. To some authors, variable structure control maintains stability of system frequency.

However, this method needs some information for system states, which are very difficult to know completely. Also, the growing needs of complex and huge modern power systems require optimal and flexible operation of them. The dynamic and static properties of the system must be well known to design an efficient controller. On the other hand, to handle such a complex system is quite complicated]. Recently the LFC systems use the proportional integral (PI) controllers in practice. Since the dynamic behavior even for a reduced mathematical model of a power system is usually nonlinear, time-variant and governed by strong cross-couplings of the input variables, special care has to be taken for the design of the controllers. Gain scheduling is a controller design technique used for non-linear systems. Therefore, a gain scheduling controller can be used for this purpose. In this method, since parameter estimation is not required, control parameters can be changed very quickly. In addition, gain scheduling application is easier than both automatic tuning and adaptation of controller parameters methods. However, the transient response for this controller can be unstable because of abruptness in system parameters. Besides, it can not be obtained accurate linear time variant models at variable operating points [2]. To solve all these problems in the above mentioned papers, an ANN controller is proposed in this study. The ANN controller has been established to apply a single area power system in the different operating points under different load disturbances by using the learning capability of the neural Networks to improve the stability of the overall system and also its good dynamic performance achievement. it is shown that the overshoots and settling times with the proposed ANN controller are better than the outputs of the other controller.

Artificial Neural Network (ANN) Controller: The ANN controller architecture employed here is a Model Reference Neural Network, which is shown in Fig. 2. As with other techniques, the Model Reference Adaptive Control configuration uses two neural networks: a controller network and a model network. The Model network can be trained off-line using historical plant measurements. The controller is adaptively trained to force the plant output to track a reference model output. The model network is used to predict the effect of controller changes on plant output, which allows the updating of controller parameters. In the study, the frequency deviations, tie-line power deviation and load perturbation of the area are chosen as the neural network controller inputs.

The outputs of the neural network are the control signals, which are applied to the governors in the area. The data required for the ANN controller training is obtained from the designing the Reference Model Neural Network and applying to the power system with step response load disturbance. After a series of trial and error and modifications, the ANN architecture shown in Fig. 2 provides the best performance. It is a three-layer perceptron with five inputs, 13 neurons in the hidden layer, and one output in the ANN controller. Also, in the ANN Plant model, it is a three-layer perceptron with four inputs, 10 neurons in the hidden layer, and one output. The activation function of the networks neurons is hyperbolic tangent. The proposed network has been trained by using back-propagation algorithm. The root mean square (RMS) error criterion is being used to evaluate the learning performance. Learning algorithms cause the adjustment of the weights so that the controlled system gives the desired response

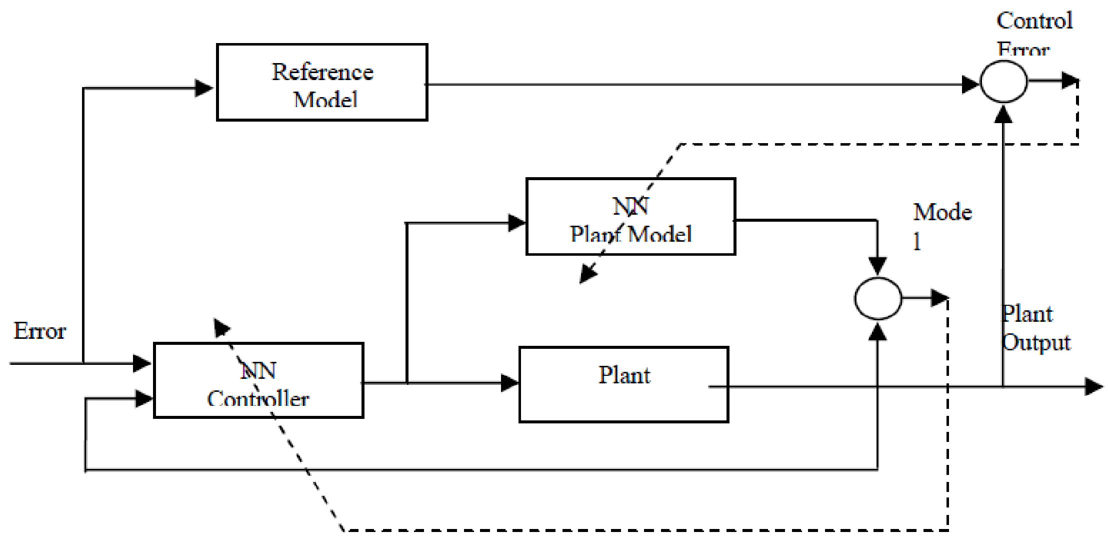


Fig. 2. The system added artificial neural network architecture

SIMULATION STUDY

The single area power system's parameters are given in Table 1. System block scheme and simulation results for the single area power system are shown in Figure 3 and 4. As can be observed, the settling time and overshoots with the proposed ANN controller are much shorter than that with the conventional PI controller.

From the figure, it is shown that the settling time of conventional PI controller is much longer than the proposed ANN controller and the overshoots of the proposed controller is almost 85% better than the PI controller's. Therefore, the proposed ANN controller provides better performance than conventional I controller for the single area power system.

Table 1. Parameters of the single area power system

Tg=0.2	Tt=0.5	R=0.05	D=0.8	H=5	Ki=7	Kp=10

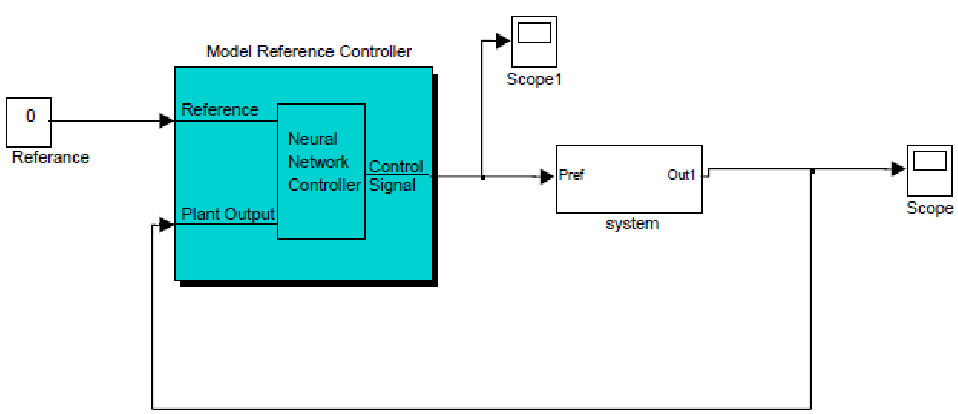


Fig. 3. System block scheme

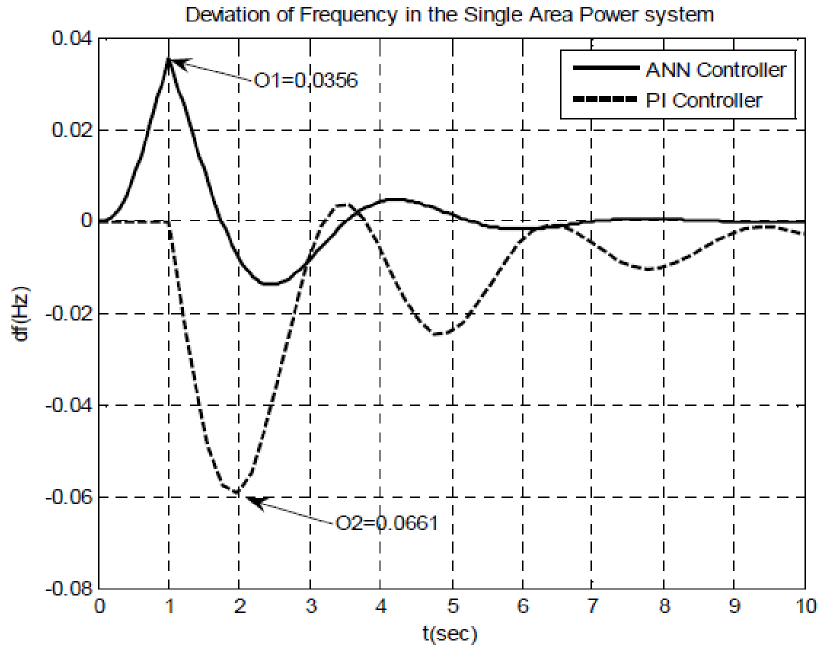


Fig. 4. Deviation of frequency of the single area power system (DPd,i=0.01 p.u.)

FUZZY LOGIC APPLICATION TO PSS

The basic system, which is simulated to analyze the effect of the proposed FLPSS on the system stability, consists of one nonlinear synchronous generator connected by long transmission line to a pure resistive load (Fig. 1). An exciter of the synchronous generator gets two external voltage signals as shown in Fig. 1, one from the Automatic Voltage Regulator (AVR) and another from the proposed FLPSS, to regulate its current and so that necessary damping torque can be obtained to damp out oscillations in minimum settling time.

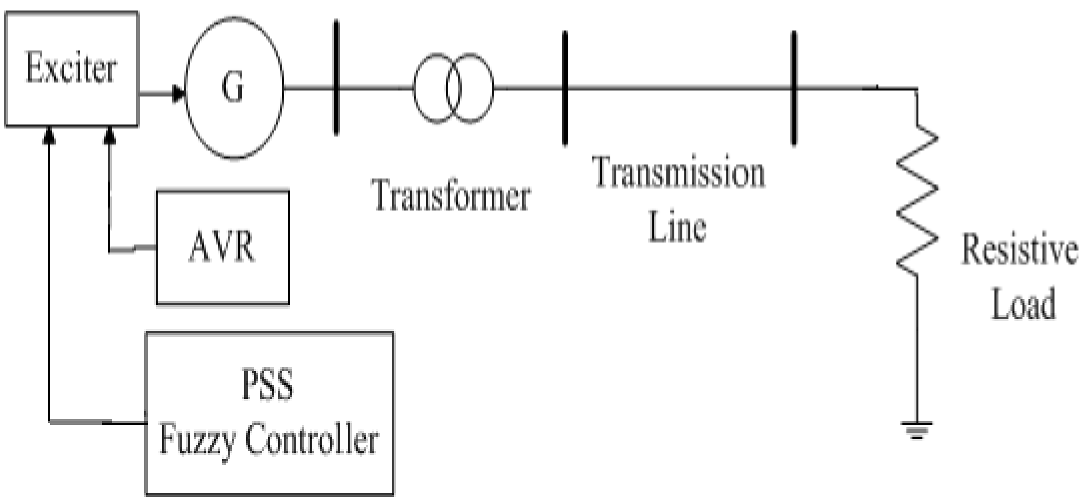


Fig 1. A Synchronous Generator and A Resistive Load System

For studying the robustness and the settling time, a two different perturbations (a Three phase to ground-LLLG fault and Single-phase to ground-LG fault on middle of the transmission line) are simulated. Also to examine an effectiveness of the proposed FLPSS on the case of a step increase in the input Pm, an analysis is carried out for a fixed input mechanical power (Pm).

Fuzzy Controlled Power System Stabilizer

Fuzzy Control System The concept of fuzzy logic given by Zadeh in 1965 has found applications in various areas including a controller for power system stabilizer. The aim of fuzzy control systems is normally to replace a skilled human operator with a fuzzy rule-based system. The fuzzy logic controller provides an algorithm which can convert the linguistic control strategy based on expert knowledge into an automatic control strategy. A fuzzy logic system, as shown in Fig. 2, comprises of four stages: a fuzzification interface, a knowledge base, an inference engine and a defuzzification interface.

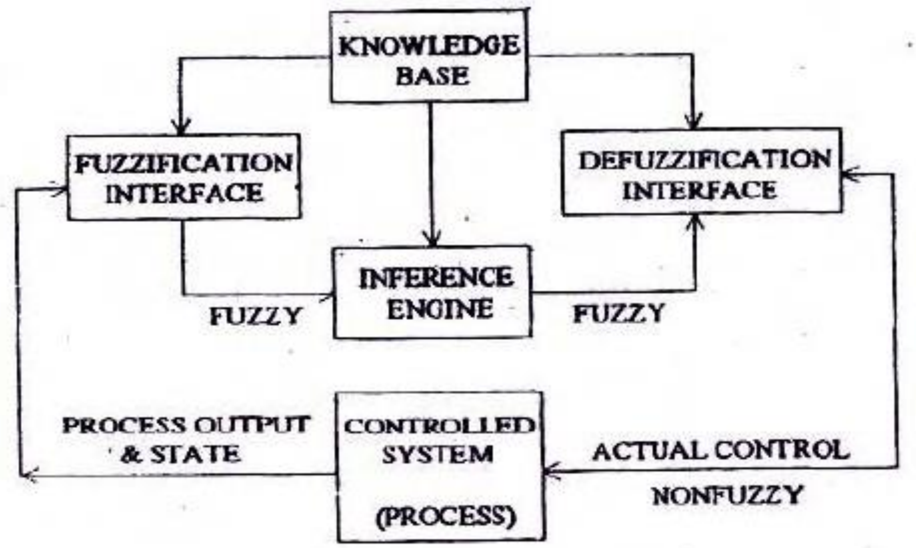


Fig 2. The General Structure of a Fuzzy Logic System

The fuzzification interface is mapping from the crisp domain into the fuzzy domain and converts input data into suitable linguistic values that can be viewed as label fuzzy sets.

Fuzzy sets can be characterized by membership functions. There are many types of membership functions e.g., the bell shaped, linear function, triangular function, trapezoidal function and exponential function

The knowledge base comprises knowledge of application domain and attendant control goals by means of set of linguistic control rules. The inference engine determines the operating condition from the measured values and selects the appropriate control actions using the rule base created from the expert knowledge.

The defuzzification inference performs scale mapping, which converts the range of values of output variables into corresponding universe of discourse and also converts the inferred decision from the linguistic variables back the numerical values.

Fuzzy Based PSS

The selection of control variables (controlled inputs and outputs) depends on the nature of the controlled system and the desired output. Usually the output error (e) and the rate or derivative of the output (de) is used as controller inputs. Since rotor angular speed is easily measurable, a proposed fuzzy logic based power system stabilizer uses two input variables: change in rotor angular speed (rotor-speed deviation) and time derivative of change of rotor angular speed (rotor acceleration) as shown in Fig.3. So, the fuzzy power system stabilizer has two-input and a single-output component which is shown in Fig. 3. An output of fuzzy logic controller is a voltage signal (Vpss) which is given to the exciter unit.

A linear triangular membership functions are used for both input and output variables. The membership function of an input variable rotor speed deviation is expressed into eight fuzzy sets, say; NVL, NL, NVB, NB, NM, NS, ZE, PS; the membership function of an input variable rotor acceleration is expressed into five fuzzy sets, say; NS, ZE, PS, PM, PB and the membership function of an output variable voltage signal (Vpss) is expressed into ten fuzzy sets, say; NVL, NL, NVB, NB, NM, NS, ZE, PS, PM, PB and they are defined in Table.1.

TABLE 1: INPUT AND OUTPUT LINGUISTIC VARIABLES

NVL	Negative Very Long
NL	Negative Long
NVB	Negative Very Big
NB	Negative Big
NM	Negative Medium
NS	Negative Small
ZE	ZERO
PS	Positive Small
PM	Positive Medium
PB	Positive Big

A set of rules which define the relation between the inputs and output of fuzzy controller are defined using the linguistic variables. The knowledge required to generate the fuzzy rules can be derived from an offline simulation. However, it has been noticed that, for monotonic systems, a symmetrical rule table is very appropriate, although sometimes it may need slight adjustment based on the behavior of the specific system. If the system dynamics are not known or are highly nonlinear, trial-and-error procedures and experience play an important rule in defining the rules [4].

In the rules of a proposed FLPSS, the input variables are connected by an „AND" method and it is meant that membership degree of a Vpss is the minimum value among the membership degree of the input variables. For the proposed FLPSS, 40 rules are defined and they are shown in Table.2.

TABLE 2: RULES FOR FUZZY PSS

Rotor Speed Deviation	Rotor Acceleration				
	NS	ZE	PS	PM	PB
NVL	NVL	NVL	NL	NVB	NB
NL	NL	NL	NVB	NB	NM
NVB	NVB	NVB	NB	NM	NS
NB	NB	NB	NM	NS	ZE
NM	NM	NM	NS	ZE	PS
NS	NS	NS	ZE	PS	PM
ZE	NS	ZE	PS	PM	PB
PS	ZE	PS	PS	PM	PB

The typical rules are having the following structure:
Rule 1: If rotor speed deviation is NM (negative medium) AND rotor acceleration is PS (positive small) then voltage (output of fuzzy PSS) is NS (negative small).
Rule 2: If rotor speed deviation is NB (negative big) AND rotor acceleration is NS (negative small) then voltage (output of fuzzy PSS) is NB (negative big).

From Fuzzification to Defuzzification for a proposed FLPSS

Maximum and minimum value, which defines the universe of discourse of rotor speed deviation, rotor acceleration and Vpss are mentioned in Table 3.

TABLE 3: RANGE OF THE INPUT VARIABLES AND OUTPUT VARIABLE

Fuzzy Variables	Max. Value (pu)	Min. Value (pu)
Rotor speed deviation	-0.85	0.01
Rotor acceleration	-0.05	0.25
Vpss	-3	1

For an example of rotor speed deviation = -0.62 and rate of change of rotor angular speed (rotor acceleration) = 0.0001, the crisp output of Vpss by defuzzification using the centroid method is given to be -0.2 and it can be understood by the following explanation:
-From Fig. 4, a membership degree for a given value -0.62 of a rotor speed deviation is 0.95 and a membership degree for a given value 0.0001 of rotor acceleration is 1. This process is called as Fuzzification Process and it is done by the triangular membership function definition in the proposed FLPSS.
-Since a given value (=-0.62) of a rotor speed deviation is belong to the NL fuzzy set of rotor speed deviation fuzzy variable and a given value (=0.0001) of a rotor acceleration is belong to the ZE fuzzy set of rotor acceleration fuzzy variable, an output signal Vpss will be NL. This process is called as Inference Process.
- A membership degree of an output signal Vpss for a given condition (IF part of rule) can be found by taking the minimum value among the membership degree of the input variables and thus value of membership degree for Vpss is 0.95.

Algorithm for FLPSS

1. The universes of discourse for each of the inputs and the output are defined.
2. The inputs are fuzzified according to the respective universe of discourses. (Fuzzification Process)
3. The fuzzy rule matrix is used to find out the activation AND the firing of control rules for this combination of inputs. (Inference Process)
4. Using the fuzzy values of output (Vpss) as obtained from the fuzzy relation matrix and the universe of discourse defined for the output variable, the crisp value of output is obtained by defuzzificition using the center of gravity method. (Defuzzification Process)
5. The above steps are repeated till the end of the simulation time.

A response of the rotor acceleration deviation to the occurrence of a LLLG fault and LG fault is shown in Fig. 5 and 6 respectively.
Both fault occurs at the middle of the transmission line at 700 km and were applied at 0.1 sec and cleared at 0.15 sec.
The settling time of system without PSS is around 8 second whereas to the system with fuzzy logic PSS takes around 0.5 second after clearing the LLLG fault and the settling time

of system without PSS is around 10 second whereas to the system with fuzzy logic PSS takes around 1 second after clearing the LG fault. It shows the ability of the FLPSS for stabling system and in specially Ts (settling time). So it can be said that after clearing the both faults, the system under FLPSS is coming back to its stable condition much faster than the system without PSS and it means that the proposed fuzzy logic power system stabilizer achieves a significantly fast damping for a rotor acceleration deviation.
A settling time of the rotor acceleration deviation under the different fault types for without PSS and with FLPSS is mentioned in Table 5.

TABLE 5: SETTLING TIMES OF THE ROTOR ACCELERATION DEVIATION UNDER THE VARIOUS FAULT TYPES

FAULT TYPE	SETTLING TIME (sec.)	
	without PSS	With FLPSS
LLLG	8	0.5
LG	10	1

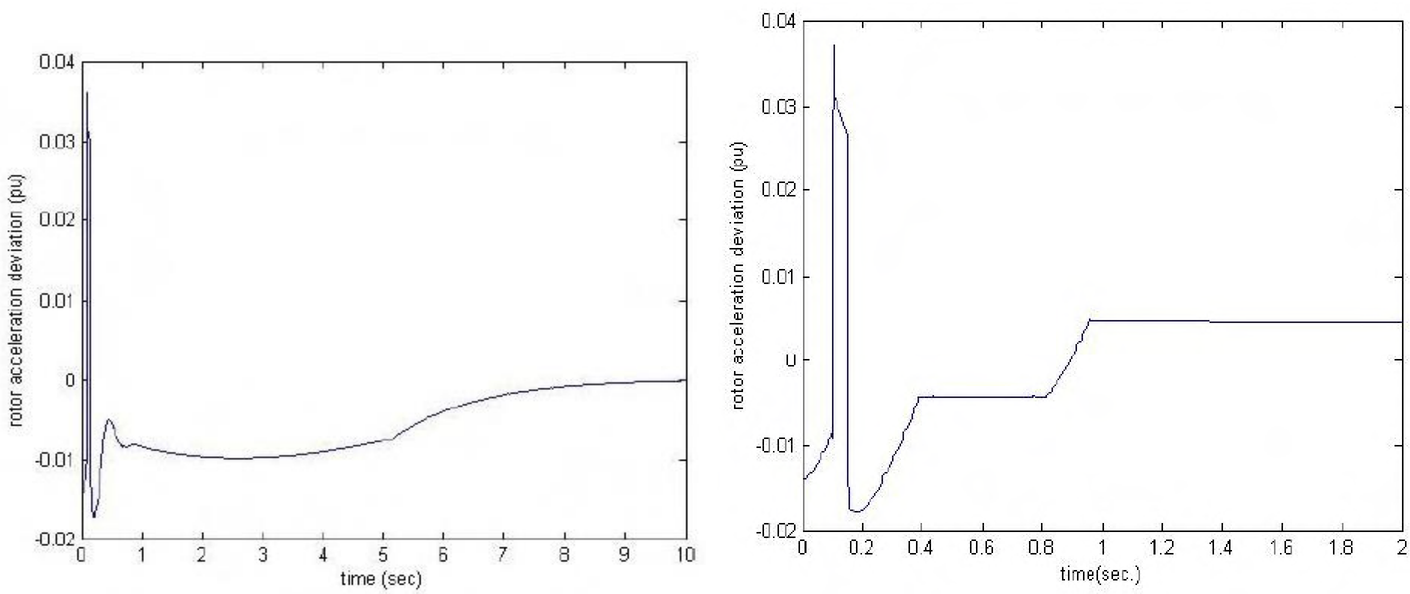


Fig. 6 (a) without PSS

Fig. 6 (b) with FLPSS

Fig. 6 Response of the per unit Rotor Acceleration Deviation to the LG Fault
The fuzzy logic power system stabilizer is designed for Single Machine Power System. Rotor speed deviation and rotor acceleration of synchronous generator were taken as the input signals to the fuzzy logic controller. The performance of the power system with fuzzy logic power system stabilizer is better one since it is effective for all test conditions. It was also shown in the simulation results that the fuzzy logic power system stabilizer can decrease the settling time of the system. The control signal, required, in all cases is with less magnitude.
The proposed FLPSS is useful for power stations which work under small and large signal disturbances. This controller is very suitable for the real time control of generators because of its simple control rules and its shorter computation time.