DEPARTMENT OF ELECTRICAL ENGINEERING

LAB REPORT OF EEE-401
(Academic Year: 2021-2022)

Design and Simulation of Electrical Systems

Load-Frequency Control

SUBMITTED BY

1. Swayam Sahu (18085071)
2. Tushar Gupta (18085073)
3. Tushar Singh Diwakar (18085074)

UNDER THE GUIDANCE OF

Prof. Dr. Maria Thomas
Department of Electrical Engineering

LOAD FREQUENCY PROBLEM

If the system is connected to numerous loads in a power system, then the system frequency and speed change with the characteristics of the governor as the load changes. So, if constant frequency is required the operator needs to adjust the velocity of the turbine by changing the characteristics of the governor when required.

SPEED GOVERNING SYSTEM

1. MATHEMATICAL MODELLING OF LOAD:

   The model gives relation between the change in frequency as a result of change in generation when the load changes by a small amount.

   The block diagram representation of the generator-load model is:

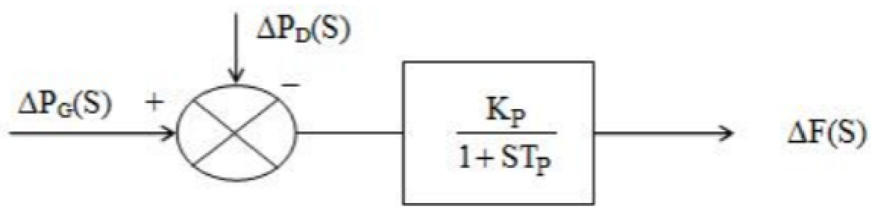


Figure 1: Mathematical modelling Block Diagram of Load

   From the above block diagram we have

   $F(s) = [\triangle P_G(s) - \triangle P_L(s)]\ \frac{K_p}{1+sT_p}\ Df^0$

   Where, $T_p = \frac{2H}{Df^0}$ = power system time constant

   And $K_p = \frac{1}{D}$ = power system gain.

2. MATHEMATICAL MODELLING OF TURBINE MODEL:

   The turbine model requires a relation between changes in power output of the steam turbine to changes in its steam valve opening $\triangle X_E$.

$G_T(s) = \frac{\Delta P_t(s)}{\Delta X_E(s)} = \frac{K_T}{1+sT_T}$

Thus the model representation of the turbine the transfer function is:

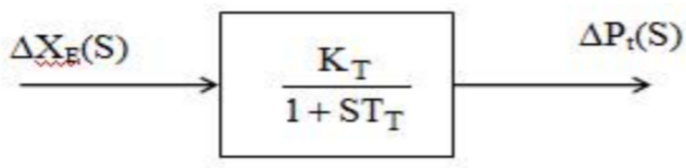


Figure 2: Mathematical modelling Block Diagram of Turbine Model

3. MATHEMATICAL MODELLING OF GOVERNOR MODEL:

Speed governor model can be developed by considering various steady state conditions. The resulting speed governor model is:

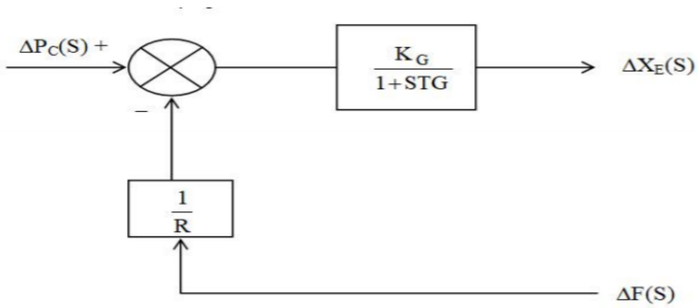


Figure 3: Mathematical modelling Block Diagram of Load

From the above block diagram we can get the following expression:

$\triangle X_E(s) = \frac{K_G}{1+sT_G}\ [\triangle P_C(s) - \frac{1}{R}\triangle F(s)]$

Where,

R = Speed regulation of the governor

K_G = Gain of speed governor

T_G = time constant of speed governor

4. TRANSFER FUNCTION OF SINGLE AREA SYSTEM:

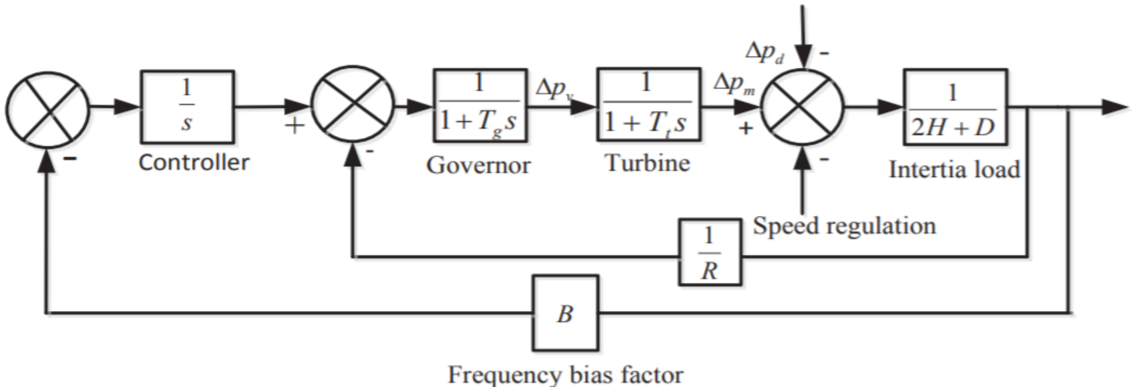


Figure 4: System block scheme using PID controller

Combining all the above block diagrams, close loop transfer function for an isolated single area system that relates the load change $\triangle P_L(s)$ to the frequency deviation $\Omega(s)$ is :

$$\frac{\Omega(s)}{-\Delta PL} = \frac{(1+T_T)(1+T_g s)}{(2H+D)(1+T_T)(1+T_g s)+\frac{1}{R}}$$

SELECTION OF PARAMETERS (From **Power System Analysis by Hadi Saadat**)

Let an isolated power station have the following parameters.

a. Turbine Time Constant, Tt = 0.5 seconds,
b. Governor Time Constant, Tg = 0.2 seconds,
c. Generator Inertia Constant, H = 5 seconds,
d. The governor speed regulation is set to R = 0.05 per unit. The turbine rated power is 300 MW at nominal frequency of 60 Hz. A sudden load change of 50 MW (ΔP_L = 0.2 per unit) occurs.
e. The load varies by 0.8 percent for a 1 percent change in frequency, i.e. D = 0.8.

COMPARISON OF DIFFERENT CONTROL METHODS

In the power system, the system load keeps changing from time to time according to the needs of the consumers, which leads to the change in frequency of the grid. So designed controllers are required for the regulation of the system variations in order to maintain the stability of the power system and its reliable operation.

The proposed artificial neural networks controller shows that the proposed controller can generate an improved dynamic response for a step load change in comparison to the PID (Tuned P = 0.767, I = 0.172, D =0.440) controller or without any controller.

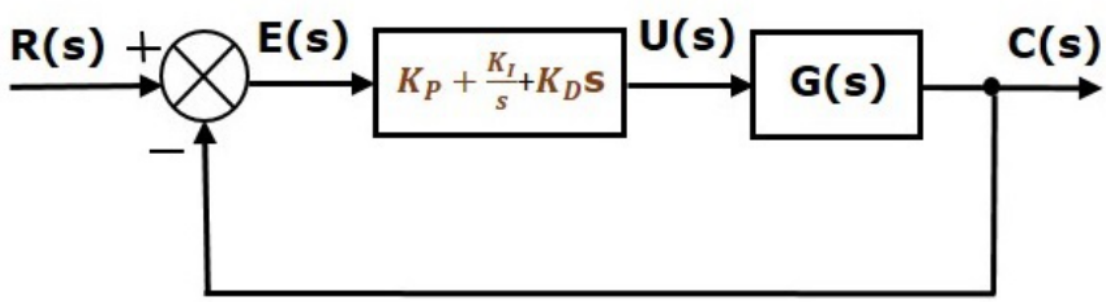


Figure 5: Block diagram of PID controller structure

The PID controller has following transfer function:

$$u(t)=K_P e(t) + K_I \int e(t)dt + K_D \frac{de(t)}{dt}$$

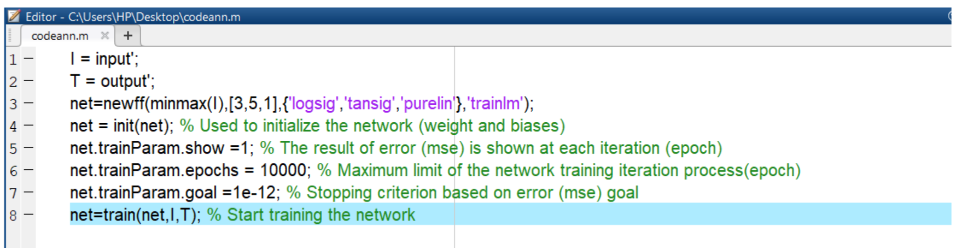
$$\Rightarrow \frac{U(s)}{E(s)} = K_P+K_I s+K_D s$$

The artificial neural network takes input from the PID input terminal and computes the weighted sum of the inputs and includes a bias. This computation is represented in the form of a transfer function.

$$Z = \sum_{1}^{n} \omega_i X$$

Since this is a classic regression problem for input-output relation of PID controllers. The use of **Adaptive Moment Estimation (or Adam)** Optimiser can prevent gradient descent stuck at local minima and RMSProp changes the learning rate as per the cost function. Taking the formulas used in the gradient descent and RMSprop, we get;

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)\left[\frac{\delta L}{\delta w_t}\right] v_t = \beta_2 v_{t-1} + (1-\beta_2)\left[\frac{\delta L}{\delta w_t}\right]^2$$




```
I = input';
T = output';
net=newff(minmax(I),[3,5,1],{'logsig','tansig','purelin'},'trainlm');
net = init(net); % Used to initialize the network (weight and biases)
net.trainParam.show =1; % The result of error (mse) is shown at each iteration (epoch)
net.trainParam.epochs = 10000; % Maximum limit of the network training iteration process(epoch)
net.trainParam.goal =1e-12; % Stopping criterion based on error (mse) goal
net=train(net,I,T); % Start training the network
```

Figure 7: ANN code in MATLAB

Backpropagation Algorithm

Almost any function can be approximated using the multilayer network if we have sufficient numbers of neurons in the hidden layer. In Fact it has been shown that two layer networks, with sigmoid transfer functions in the hidden layer and linear transfer functions in the output layer can approximate virtually any function of interest to any degree of accuracy , provided sufficiently many hidden units are available.

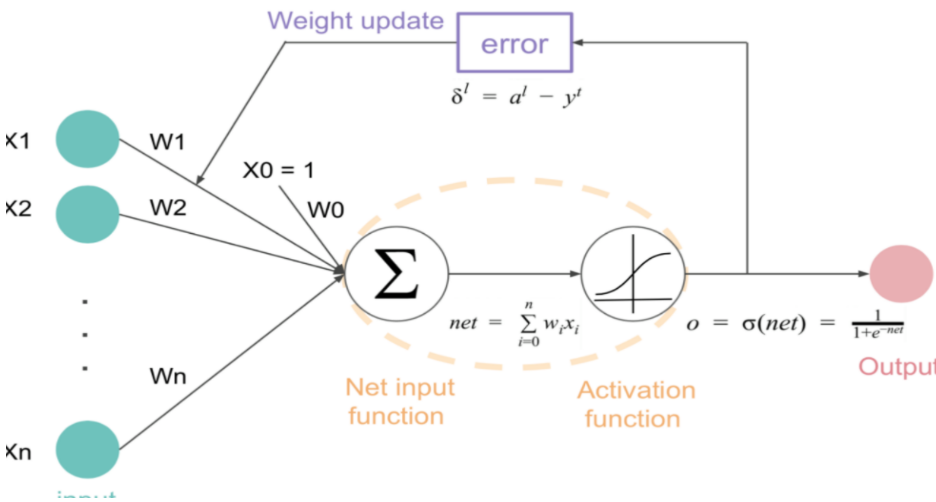


Figure 6:Neural Network Architecture

$$Z = \sum_{1}^{n} \omega_i X + b$$

NNTOOL method provides the facility to train through one of the methods Say conjugate gradient method, Levenberg-Marquardt method for back propagation. In the neural network we can employ TANSIG as a transfer function in the hidden layer and PURELIN in the output layer. Then the obtained weights and biases are chosen as the initial weights and biases.

The range over which error signal is in transient state, is observed. Responding values of the proportional, integral and derivative constants are set. This set is kept as a target. Range of the error signal is taken as the input. This input – target pair is fed and a new neural network is formed using “nntool” in the MATLAB Simulink software.

For multilayer networks output of one layer becomes input to the following layer.

Step-1

Propagation of input forward through the network

$$a^0 = p \ ,$$

$$a^{m+1} = f^{m+1}\left(W^{m+1} a^m + b^{m+1} \right) \qquad for\ \ m = 0, 2$$

$$a = a^m$$

Step -2

Propagation of sensitivities backward through the network

$$s^M = -2F^M(n^M)*(t-a),$$

$$s^m = F^m(n^m)*(W^{m+1})^T s^{m+1}, \quad for\ m = M-1, \ldots, 2, 1.$$

Step -3

Updation of weights and biases using approximate steepest descent rule

$$W^m(k+1) = W^m(k) - \alpha s^m (a^{m-1})^T \ ,$$

$$b^m(k+1) = b^m(k) - \alpha s^m \ .$$

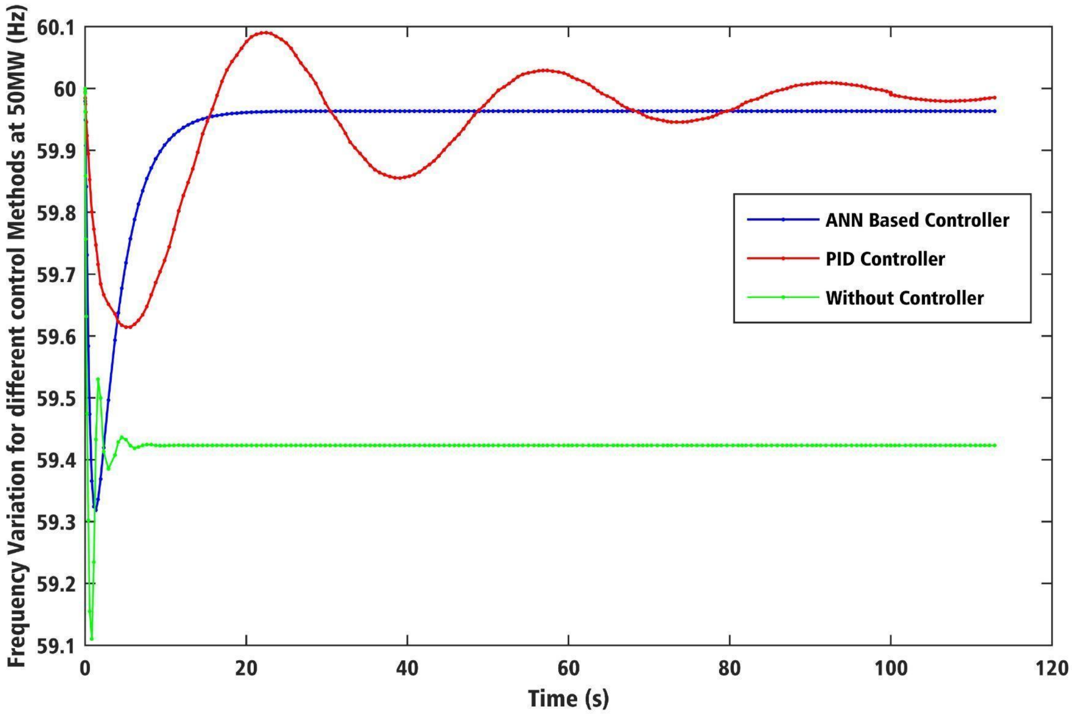


Figure 8: Different Control Schemes

CIRCUIT DIAGRAM

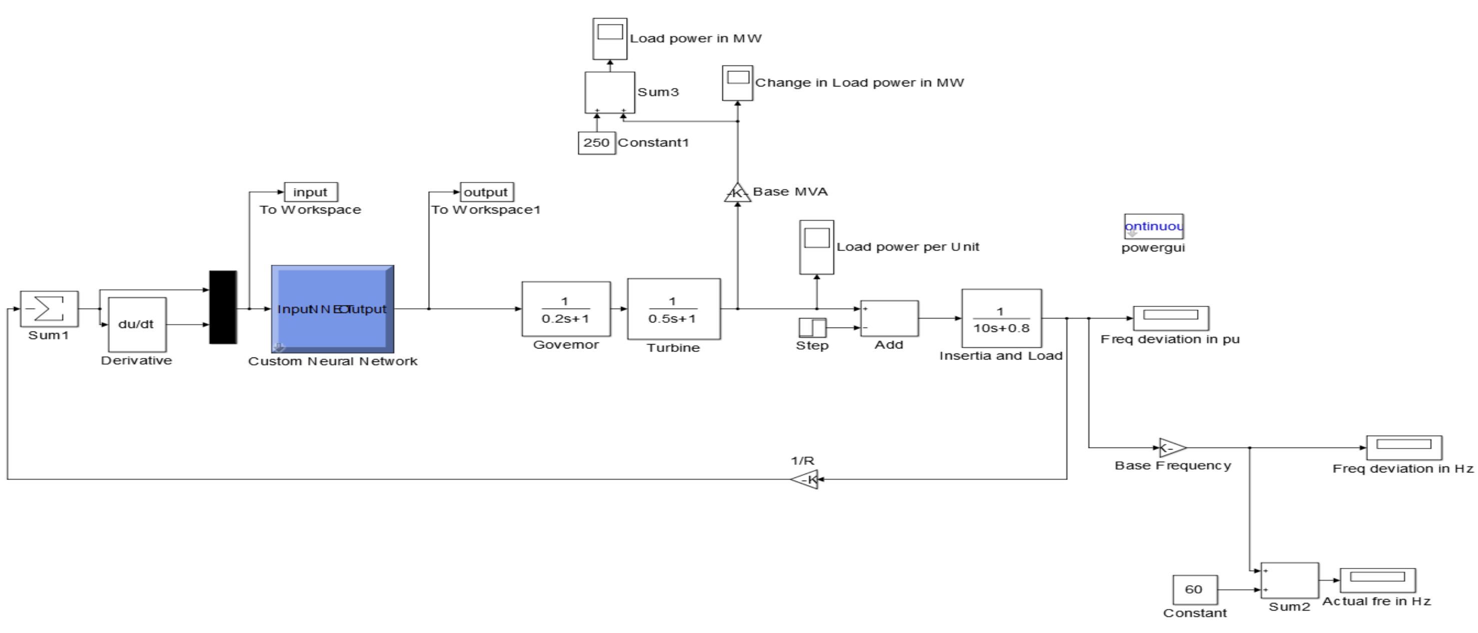


Figure 9: Load frequency control using ANN based controller

LOAD VARIATION

Sudden Load Change	**$\triangle P_L$ p.u. (T_L = 250MW)**	**$P_L' = \triangle P_L + P_L$**
55(iii)	0.22	305
52(iv)	0.208	302
50(i)	0.20	300
60(ii)	0.24	310
58(v)	0.232	308

Table 1: Load variation data

FREQUENCY LOAD VARIATION USING ANN BASEDCONTROLLER:

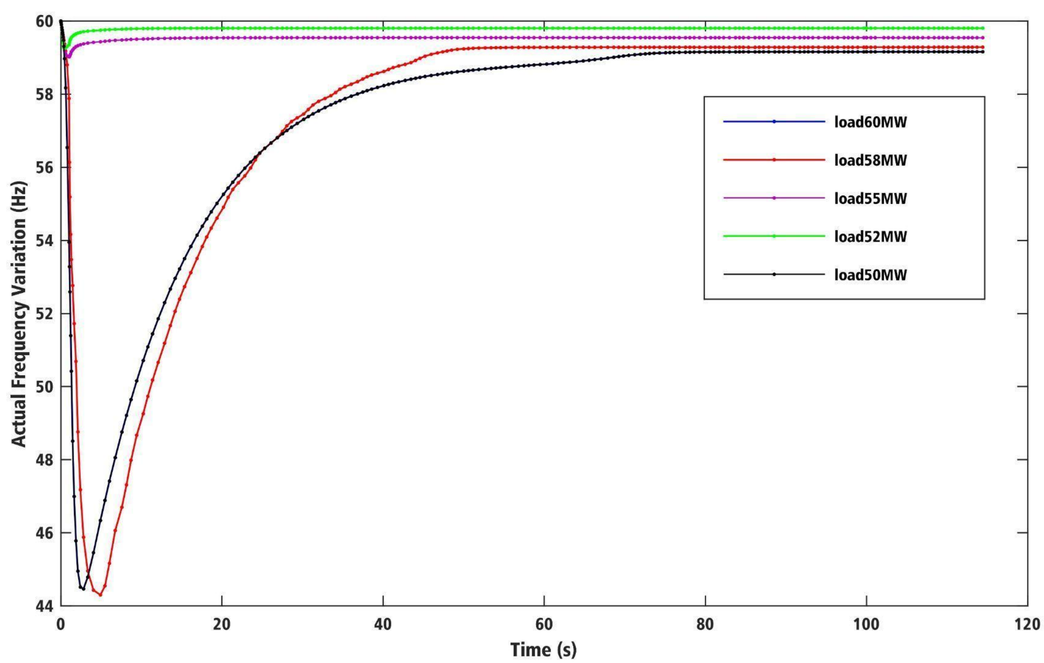


Figure 10: Actual Frequency at different Loads

CONCLUSION

From the simulation results obtained for load disturbances for ANN controller, PID controller we can conclude that the settling time and overshoots with the proposed ANN controller are much shorter than that with the conventional PID controller.

Variation of Frequency is also analysed at various loads (i.e. at 60MW, 58MW, 55MW, 52MW and 50MW) which shows proposed Artificial Neural Network based controllers attain steady state with good dynamic response.

Simulation Files and the ANN code file has been uploaded in the given repository-

https://github.com/Tusharsd123/Load-Frequency-Control-