International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 9, September - 2013

Adaptive Neural Network-Based Flat Controller for One Axis Synchronous Generators .

M.A. Gadam
Electrical/Electronics Department,
Federal Polytechnic, Bauchi,
Bauchi, Nigeria.

E.C. Anene
Electrical/Electronics Department,
Abubakar Tafawa Balewa University,
Bauchi,Nigeria.

Abstract—**This paper presents an artificial neural network based flat controller for synchronous generators. A neural network to identify the field voltage (efd) characteristics of a synchronous machine equipped by a flat controller was developed. A neural network of 2 inputs, two hidden and one output layers was used by MATLAB toolbox to simulate the excitation (efd).The network has the features of a simple structure, adaptability and fast response. The error between the Neural Network approximated efd of the flat controller and the desired efd was found to be minimal after the training. Results show that the neural network emulates the efd and were successfully applied to stabilize the system under 3-cycle fault condition. The network response is evaluated in a synchronous generator connected to a single machine infinite bus**

Keywords: Multilayer Percepterons, Neural Networks, Identification, synchronous generator, Flat systems, Feedback Linearization.

I. INTRODUCTION

Parameters in the electrical power system (PS) change with time, slowly due to environmental effects or rapidly due to faults. Thus it is necessary to update the control law with system changes.

The design of adaptive controllers to improve the performance of the power system has been a topic of research for a long time. Neural networks (NN) are a suitable choice for the control of complex nonlinear plants since the conventional control methods show limitations in performance. Due to advantages of high computation speed, generalization and learning ability, neural networks have been successfully applied to the identification and control of nonlinear systems [1].

One of the promising applications of NN in PS is in the area of power stabilization. Neural network based power system stabilizers (PSS) [2] have been shown to be very effective in damping out the PS lower frequency oscillations and experimentally have been shown to have much better performance over a conventional PSS. Another important application of NN controller is for transient stability enhancement. This is a subject of this paper.

The flatness-based controller has been employed to enhance the first swing stability. The approach is to compute using “system flatness theory” the linearizing state or flat output for the reduced order model of the synchronous machine and design a flatness-based feedback law [3]. ANN back-propagation will be used to train a controller using the input-output data generated by the flatness-based feedback law and the neural controller will replace the flat controller to control the system.

The flatness-based scheme is an offshoot of the input-state feedback linearization making use of the same linearizing output. Once the system is shown to be flat, in effect implying that the system possesses a well characterized dynamics, it becomes amenable to trajectory generation and system tracking without the need for simultaneous on-line parameter identification and controller adaptation as would be the case in adaptive control strategies [4]. This is because all system parameters and control becomes a function of the linearizing output that can enable the generation of reference trajectories a-priori. The construction of the feedback law is done by a simple inversion of system equations with respect to the control. In our scheme, the Neural network is used to learn the constructed ‘Flatness’ based feedback law and the performance is evaluated in simulation where the machine is configured in a single machine infinite bus system.

II. DESCRIPTION OF THE MATHEMATICAL POWER SYSTEM MODEL

A simplified dynamic model of the power system is used, namely a single generator connected through two parallel transmission lines to a very large network approximated by an infinite bus. The model is shown below:

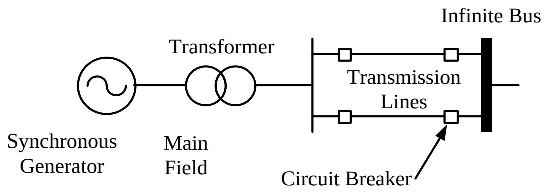

Fig.1 Single machine infinite bus model

Modeling of a synchronous machine is done by the use of equivalent circuits for the magnetic (the rotor and stator windings and couplings) and the external electrical circuits (the bus and loading). It also uses the appropriate representation of the mechanical parts of the system (speed and inertia system) and adequate representation of auxiliary (turbine and excitation system) equipment. The machine has two stator circuits and two rotor circuits described by two differential equations and together with the swing equation gives a fourth order synchronous machine representation:

$$\tau_{q0}\dot{e}_d^{'} = -e_d^{'} - (x_q - x_q^{'})i_q \quad (1)$$

$$\tau_{d0}\dot{e}_q^{'} = e_{fd} - e_q^{'} - (x_d - x_d^{'})i_d \quad (2)$$

$$\frac{2H}{w_R}\frac{d^2\delta}{dt^2} = P_m - D(\omega - \omega_0) - e_d^{'}i_d - e_q^{'}i_q \quad (3)$$

$$\dot{\delta} = \omega - \omega_0 \quad (4)$$

The currents i_q and i_d are therefore given by

$$i_d = \frac{1}{(r_a + R_e)^2 + (x'_d + x_e)(x'_q + x_e)}(-(r_a + R_e)(e'_d - V_\infty \sin\delta) + (x'_q + x_e)(e'_q - V_\infty \cos\delta) \quad (5)$$

$$i_q = \frac{1}{(r_a + R_e)^2 + (x'_d + x_e)(x'_q + x_e)}(-(r_a + R_e)(e'_d - V_\infty \sin\delta) + (x'_q + x_e)(e'_q - V_\infty \cos\delta) \quad (6)$$

By further assumptions, $e_d^{'}$ becomes an algebraic expression since the synchronous machine is connected to an infinite bus, the d-q terminal voltage v_d, v_q are constrained load equation

III. THE MLP BASED FLAT CONTROLER

A. *Flatness*

A system is said to be diffrerentially flat [5] if there exists a set of independent variables referred to as flat output such that every other system variable (including the input variables) is a function of the flat output and a finite number of its successive time derivatives. More precisely, the system

$$f(\dot{x}, x, u) = 0 \quad (7)$$

with $x \in R^n$ and $u \in R^m$ is differentially flat if one can find a set of variables called flat output;

$$y = h(x, u, \dot{u}, \ddot{u}, \ldots\ldots, u^{(r)}) \quad (8)$$

$y \in R^m$ and system variables,

$$x = \alpha(y, \dot{y}, \ddot{y}, \ldots\ldots, y^{(q)}) \quad (9)$$

and control,

$$u = \beta(y, \dot{y}, \ddot{y}, \ldots\ldots, y^{(q+1)}) \quad (10)$$

with q a finite integer such that the system equations

$$0 = f(\frac{d\alpha}{dt}(y, \dot{y}, \ddot{y}, \ldots\ldots, y^{(q)})), \alpha(y, \dot{y}, \ddot{y}, \ldots\ldots, y^{(q)}), \beta(y, \dot{y}, \ddot{y}, \ldots\ldots, y^{(q+1)})) \quad (11)$$

are identically satisfied .[3] shows the computation of the flat output for the synchronous generator here considered.

The state of the SMIBS is a function of the flat output δ and its derivatives up to order $\alpha = 2$. The endogenous feedback system to the following closed loop system is of order $\alpha + 1 = 3$, so that from the linear system

$$\dddot{\delta} = v \quad (12)$$

the compensator follows. Considering the systems' dynamical equations, perform the following state transformations:

$$\begin{aligned} \dot{z}_1 &= z_2 = \dot{y}_1 = \dot{\delta} = \omega - \omega_0 \\ \dot{z}_2 &= z_3 = \ddot{y}_1 = \ddot{\delta} = \dot{\omega} \\ \dot{z}_3 &= \dddot{y}_1 = \dddot{\delta} = \ddot{\omega} = v \end{aligned} \quad (13)$$

This yields the equivalent normal form for the system, from which we can compute the nonlinear controller by inverting the expressions from $\ddot{\omega}$ and e_{fd}. The state transformations are invertible and exist throughout the transient operating zone $0 < \delta < 180^o$. Using the network parameters of figure (1), e_{fd} is proved to be a function of the flat variable and its derivatives, that is

$$e_{fd} = \beta(\delta, \dot{\delta}, \ddot{\delta}) \tag{14}$$

B. MLP Neural Network

Among the Neural Networks, feed forward networks, namely Multilayer Perceptrons (MLP) is very much used for different applications. This network type has been proven to be a universal function approximator. Therefore MLP can be powerful tool for system identification.

In this paper, the MLPN consists of three layers of neurons (input, hidden, and output layers as shown in Fig. 2) interconnected by the input and output weight matrices W and V, respectively. The weights of the MLPN are obtained using the backpropagation algorithm [6]. The activation function for neurons in the hidden layer is the following sigmoidal function

$$h(x) = \frac{1}{1+exp^{(-x)}} \tag{15}$$

During online training, the MLPN starts with small random initial values for its weights, and then computes a one-pass backpropagation algorithm at each time step k, which consists of a forward pass propagating the input vector through the network layer by layer, and a backward pass to update the weights by the gradient descent rule. The output layer neurons are formed by the inner products between the nonlinear regres sion vector from the hidden layer and the output weight matrix V. The inner weights (W) for the MLPN are updated by (19)

$$\Delta W_l = \frac{\partial J}{\partial W_l} = \frac{\partial J}{\partial t_A}\frac{\partial t_A}{\partial p_L}\frac{\partial p_L}{\partial q_L}\frac{\partial q_L}{\partial p_l}\frac{\partial p_l}{\partial q_l}\frac{\partial q_l}{\partial W_l}$$

$$= [\{h(ql)(1-h(ql))X\}\sum_{j=1}^{m_i} 1.W_L] \tag{16}$$

where

$J(k) = (1/2)\sum\sum_j [E_j(k)]^2$, where E(k) is the error between outputs of the plant and MLPN, and k indicates discrete sampling time;

t_A target value;

L,l output and hidden layers, respectively;

m_i number of neurons in the hidden layer;

p output of the activation function for a neuron;

q regression vector as the activity of a neuron;

X input vector of the MLPN.

The function h is the sigmoid function in (15). By trial and error, 20 neurons in the hidden layer are optimally chosen for the off-line training. These values depend on a tradeoff between convergence speed and accuracy.

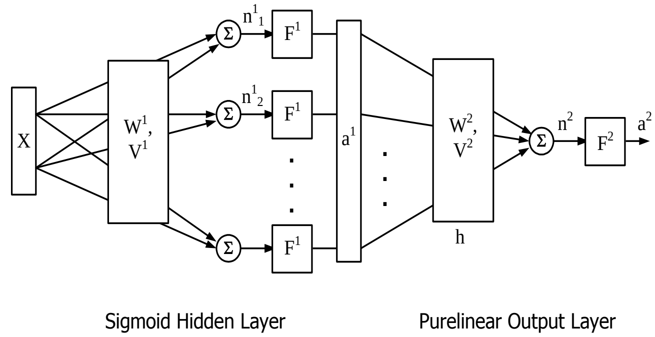

Fig.2 Multi-layer Perceptron Neuron structure

C. Flat-Neural Controller

In this approach, the signals of the system measurements control by a flat controller are to be adapted as input signal to the ANN controller. The latter uses the back propagation MLP to train the network to reduce the error between the system output and the desired output. The ANN Controller replaces the flat controller and produces an output signal to control the system and keep it at the desired operating position.

A pattern from the training set to be identified is the efd and is presented in the input layer of the network and the error is calculated in the output layer. The error is propagated backwards towards the input layer and the weights are updated. This procedure is repeated for the entire training pattern. At the end of the iteration, test patterns are presented to ANN and classification performance of the ANN is evaluated. Further training is continued till the desired classification performance is reached.

Forward propagation: The output of each node in the successive layers is calculated.

$$O(output\ of\ a\ node) = \frac{1}{1+\exp(-\sum w_{ij}\, x_{ij})} \tag{17}$$

Where;

w_{ij} : The weight matrix connecting nodes of the previous layer i with nodes of next layer j.

x_{ij} : The variables of a pattern.

o : The output of a node in the successive layer.

The error E(p) of a pattern number p is calculated;

$$E(p) = (\tfrac{1}{2})\sum(d(p) - O(p))^2 \tag{18}$$

Where, d is the target which is efd.

Reverse propagation: The error δ for the nodes in the output layer is calculated,

$$\delta(output\ layer = O(1-O)(d-O) \tag{19}$$

The new weights between output layer and hidden layer are updated,

$$w(n+1) = w(n) + \eta\delta(output\ layer)O(hidden\ layer)$$
(20)

where: η is the learning rate.

The error δ for the nodes in the hidden layer is calculated

$$\delta(hidden\ layer) = O(1-O)\sum \delta(output\ layer)w(updated\ weights\ between\ hidden\ and\ output\ layer) \quad (21)$$

The weights between hidden and input layer are updated,

$$w(n+1) = w(n) + \eta\delta(\text{hidden})\text{O}(\text{input layer}) \quad (22)$$

The above steps complete one weight updating. The remaining training patterns are presented in further update iterations using equations 17-22. The training of the network is stopped once the desired Mean square Error (MSE) is reached as given below

$$E(MSE) = \sum E(p) \quad (23)$$

The final updated weights are saved and used to generate the efd pattern in simulation runs of the SMIBS.

IV. SIMULATION

The simulation results for the third order SMIBS equipped with a controller referred to as Field Voltage Dynamic Feedback Controller (FVDFC) is presented in Figure 3. The figure clearly shows the response of the third order SMIBS equipped with FVDFC designed in [3], to a three-phase short circuit fault of 3-cycles duration. Note that the field voltage settled to steady state in 3 seconds. Fig. 3 described above is the pattern the MLP was trained to recognize.

The system and network parameters used in the computation of the control law are assumed to remain constant during the simulation period. The system simulation data are summarized in Tables 1-3. The per-unit values are expressed on a common 100MVA base. The operating point of the system was determined using the data presented in Tables 1 to 3 in the appendix. The trained MLP was later used to replace the efd and further simulations done with the SMIBS to ascertain its response.

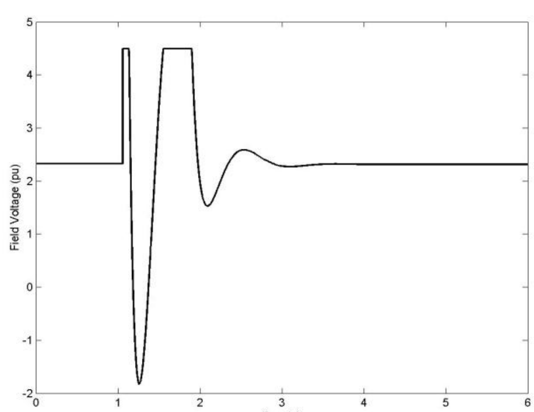

Fig.3 Control signal (Field Voltage) in a 3-Cycle Fault using the flat controller (FVDFC).

V. RESULTS

The input-output data set of the FVDFC was used to train a neural network at 1000 epochs with 0.001 acceptable mean square error tolerances and the result is shown in figure 4. The result shows how the neural network emulates the output of the FVDFC and points A, B, C, D, E, F, G and H in fig.4 shows the performance of the neural network at the transition points and the points are magnified and shown in different plots in figures 4a, b, c, d, e, f, h and h respectively.

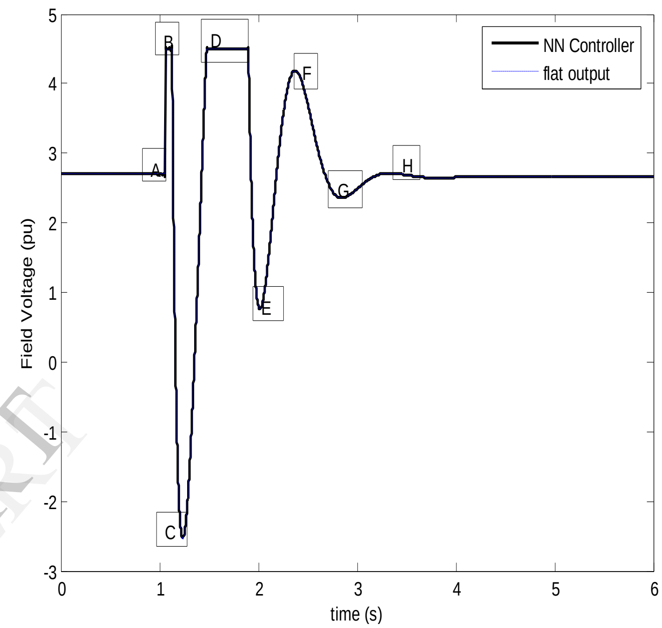

Fig. 4 Neural Network approximation of the field voltage

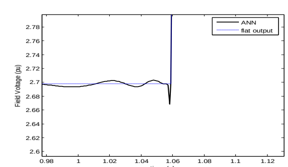

Fig.4a Point A

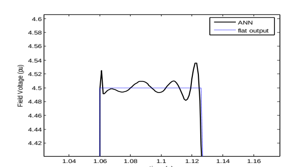

Fig.4b Point B

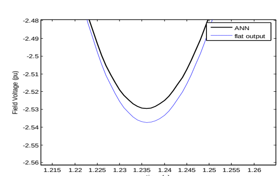

Fig.4c Point C

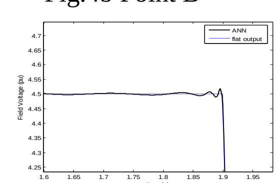

Fig.4d Point D

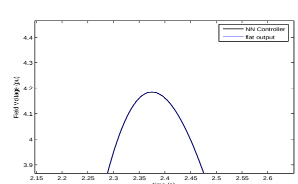

Fig.4e Point E

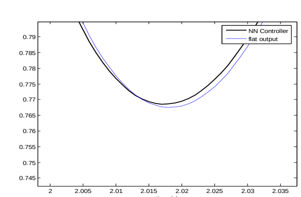

Fig.4f Point F

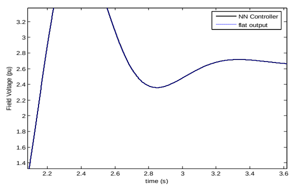

Fig.4g Point G

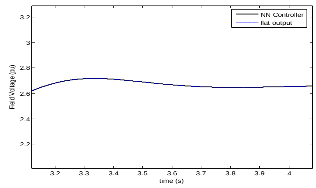

Fig.4h Point H

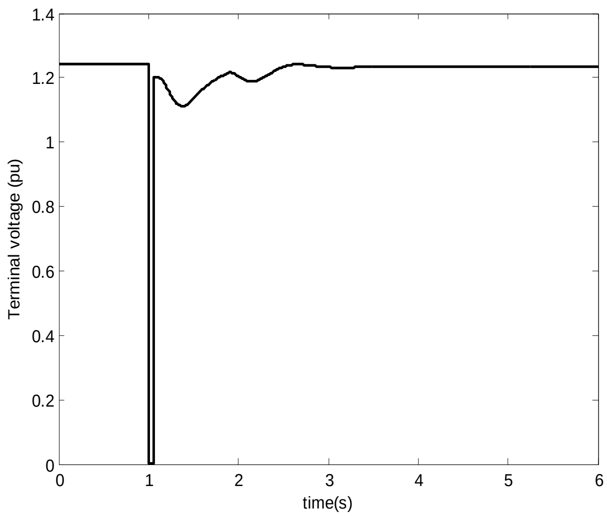

Fig.5 Response of Terminal Voltage in a 3-Cycle Fault with Flat Neural Controller.

The figures clearly show the responses of the third order SMIBS equipped with Flat Neural Network designed in this work, to a three-phase short circuit fault of 3-cycles duration. A close look at Figure 7 reveals that the system was restored to equilibrium in three seconds. Figures 8, 5 and 6 give the responses of the speed deviation, terminal voltage and electrical power respectively.

Note also that the controller damped out the fault oscillation within 4.2 seconds from the time of fault inception

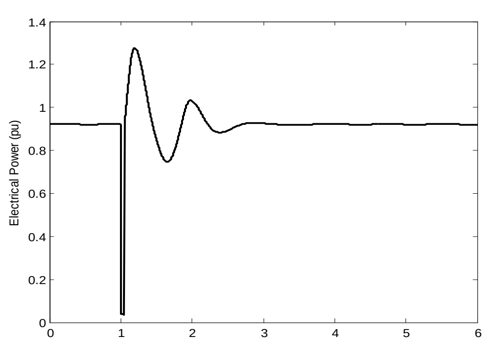

Fig.6 Response of Electric Power using Flat Neural controller

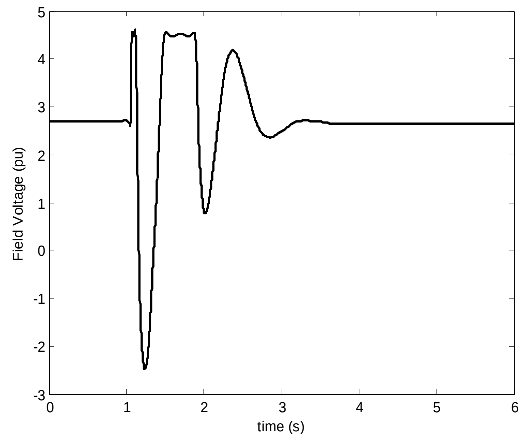

Fig.7 Response of Field Voltage in a 3-Cycle Fault with Flat Neural controller.

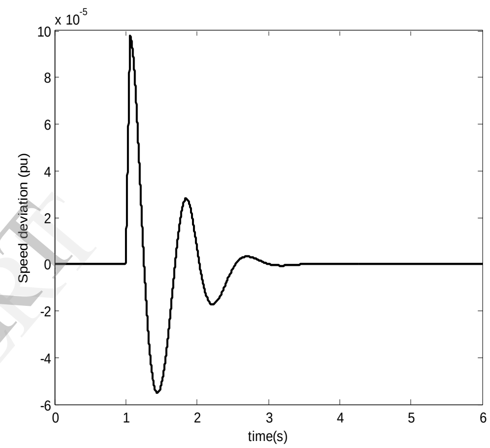

Fig.8 Response of Field Voltage in a 3-Cycle Fault with Flat Neural Controller.

VI. CONCLUSION

The dynamic behavior of the single machine infinite bus system (SMIBS) with the controller appropriately incorporated is amply demonstrated via several simulations in Matlab environment. Simulations were done with the designed flat controller attached to the field voltage of SMIBS.

To overcome the drawbacks of the conventional power system stabilizers, an MLP neural network trained to learn the dynamics of flat controller is presented. The proposed method is evaluated in a single machine infinite bus power system. The design is based on the flat output of the synchronous generator, which is the phase angle. Simulation results for the 3-cycle faults demonstrate the effectiveness and robustness of the flatness based neural controller. Such a nonlinear adaptive controller will yield a better and fast damping under small and large disturbances even with changes in the operating conditions. Better and fast damping means that generators can operate more close to their maximum generation capacity. Thus, ensuring that generators remain stable under sever faults such as three fault short circuits.

REFERENCES

[1] Wenxin Liu,Ganesh K. Venayagarmoorthy, Donald C. Wunsch II, "Adaptive Neural Network Based Power System Stabilizer Design," IEEE Trans. Energy conv., Vol.17, No.3, pp. 406-414, May 2006.

[2] Abido M.A "Adaptive tuning of Power System Stabilizers using Radial Basis Function Neural Networks," IEEE Trans. Energy conv., Vol.17, No.3, pp. 406-414, May 2006.

[3] E.C.Anene and O.U. Aliyu, J. Levine and G.K.Venayagamoorthy, "Flatness-Based Feedback Linearization of A Synchronous Machine model with Static Excitation and Fast Turbine Valving," IEEE PAS Power meeting, Tamper Florida, July, 2007.

[4] E.C. Anene, Agee J.T. and Levine J.and O.U. Aliyu, " A New Technique For feedback Linearization And Applications In Power System Stabilisation," IASTED Int. Conf. on POWER, ENERGY and APPL, Gaborone Bostwana, September, 2006.

[5] Kiss B, Levine J. "On the Control of a Reduce Scale Model of the US Nave Cranes" IEEE Conf. on Intell. Syst. Slovakia, 1999.

[6] Jung-Wook Park, Ganesh K.V., Ronald G.H, "MLP/RBF Neural Networks–Based Online Global Model Identification of Synchronous Generator," IEEE Trans. Ind. Electr., Vol.52, No.6, pp.1685-1695, December,2005.

[7] P. S. Babu, A. Ghosh, Sachchidanand, "A Coordinated Fuzzy Fastvalving Controller for Transient Stability Enhancement of SMIB System", ECC'97, Brussels, July,1997

VII. APPENDIX

TABLE I. SYSTEM DATA OF THE GENERATOR AND NETWORK

Parameter	***Value***	***Unit***
Machine Power	1.0	Pu
Power Factor	0.85	Pu
Infinite Bus Volts.	1.0	Pu
Machine Speed	314.519	rad/s
Reactance Xe	0.4	pu
Resistance Re	0.02	pu

TABLE II. SYSTEM FIELD VOLTAGE LIMITS

Parameter	***Value***	***Unit***
$e_{fd\,\max}$	4.5	Pu
$e_{fd\,\min}$	-4-5	pu

TABLE III. TURBINE INITIALISATION DATA

Parameter	***symbol***	***Value***	***Unit***
Turbine Power	P^0	1.0	Pu
Turbine Time constant	τ_t	0.7	S
Governor Time constant	τ_g	0.3	S
Turbine speed regulation	R_T	20	Pu
Governor max. Limit	$p_{gv\,\max}$	1.05	pu