

Electronic Theses and Dissertations, 2004-2019

2017

Online Neuro-Adaptive Learning For Power System Dynamic State Estimation

Rahul Birari
University of Central Florida

 Part of the [Electrical and Computer Engineering Commons](#)
Find similar works at: <https://stars.library.ucf.edu/etd>
University of Central Florida Libraries <http://library.ucf.edu>

This Masters Thesis (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations, 2004-2019 by an authorized administrator of STARS. For more information, please contact STARS@ucf.edu.

STARS Citation

Birari, Rahul, "Online Neuro-Adaptive Learning For Power System Dynamic State Estimation" (2017).
Electronic Theses and Dissertations, 2004-2019. 5686.
<https://stars.library.ucf.edu/etd/5686>

ONLINE NEURO-ADAPTIVE LEARNING FOR POWER SYSTEM DYNAMIC STATE
ESTIMATION

by

RAHUL BIRARI
B.E. University of Mumbai, 2012

A thesis submitted in partial fulfilment of the requirements
for the degree of Master of Science
in the Department of Electrical & Computer Engineering
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Fall Term
2017

© 2017 Rahul Birari

ABSTRACT

With the increased penetration of Distributed Generators (DGs) in the contemporary Power System, having knowledge of rapid Real-Time electro-mechanical Dynamic States has become crucial to ensure the safety and reliability of the grid. In the conventional SCADA based Dynamic State Estimation (DSE) speed was limited by the slow sampling rates (2-4 Hz) so State Estimation was limited to static states such as Voltage and Angle at the buses.

Fortunately, with the advent of PMU based synchro-phasor technology in tandem with WAMS, it has become possible to avail rapid real time measurements at the network nodes. In this paper, we have proposed a novel Machine Learning (Artificial Intelligence) based Real-Time Neuro-adaptive Algorithm for rotor angle and angular frequency estimation of synchronous generators, here proposed algorithm is based on reinforcement learning and adapts in real-time to achieve accurate state estimates. Applicability and accuracy of the proposed method is demonstrated under the influence of noise and faulty conditions. Simulation is carried out on Multi-Machine scenario (68 bus 16 generator NETS-NYPS model).

Dedicated to my famliy, friends and all of those who helped me.

TABLE OF CONTENTS

LIST OF FIGURES	vi
LIST OF TABLES	vii
CHAPTER 1: INTRODUCTION	1
1.1 Introduction to the Power System State Estimation	1
1.2 Phasor Measurement Unit (PMU)	2
1.3 Literature Review	3
1.4 Thesis Organization	5
CHAPTER 2: ARTIFICIAL NEURAL NETWORK	6
2.1 Types of Training	6
2.2 Design of Neural Network	7
2.2.1 Single Neuron	7
2.2.2 Single Hidden Layer Neural Network	8
2.2.2.1 Forward Propagation	9
2.2.2.2 Objective Function	10

2.2.2.3	Back-Propagation	11
2.2.2.3.1	Chain Rule	11
2.2.2.3.2	Gradient Descent	12
2.2.2.3.3	Learning Rate	12
CHAPTER 3: POWER SYSTEM MODELING		13
3.1	Transient	13
3.2	Generator Model	14
3.2.1	Swing Model	14
3.2.2	Swing Equation numerical Integration (Rectangular Rule)	15
3.3	Network Model	16
3.3.1	16 Gerator 64 Bus Model	16
3.3.2	Measurement Model	16
CHAPTER 4: KALMAN FILTER		18
4.1	Kalman Filter	18
4.2	Extended Kalman Filter (EKF)	18
4.2.1	Jacobian Matrices	19
4.2.1.1	System Matrix A	20

4.2.1.2	Measurement Matrix H	21
4.3	Unscented Kalman Filter (UKF)	22
4.3.1	Numerical Stability	24
CHAPTER 5: NEURO-ADAPTIVE DYNAMIC STATE ESTIMATION		25
5.1	Introduction to Dynamic State Estimation	25
5.2	Neuro Adaptive Dynamic State Estimation	25
5.2.1	Problem Formulation	26
5.2.2	Learning Objective & Algorithm	27
5.2.3	Neural Network Design	28
CHAPTER 6: SIMULATIONS AND RESULTS		31
6.1	Test Case Studies	31
6.1.1	Case -1	31
6.1.2	Case -2	34
CHAPTER 7: CONCLUSION		37
LIST OF REFERENCES		38

LIST OF FIGURES

Figure 2.1: Single Neuron	7
Figure 2.2: Single Hidden Layer Neural Network	8
Figure 3.1: Generator Equivalent Model	14
Figure 5.1: Neuro-Adaptive Dynamic State Estimation Algorithm	26
Figure 6.1: NETS-NYPS (16 Generators 68 Bus System)	32
Figure 6.2: Case-1 Generator Angles - simulated with $\eta = 0$ (Fault at bus 28 to 29 applied at $t = 1.0\text{sec}$)	33
Figure 6.3: Case-1 Generator Speed - simulated with $\eta = 0$ (Fault at bus 28 to 29 applied at $t = 1.0\text{sec}$)	33
Figure 6.4: Case-2 Generator Angles - simulated with $\eta = 0.7$ (Fault at bus 28 to 29 applied at $t = 5.0\text{sec}$)	34
Figure 6.5: Case-2 Generator Angles - simulated with $\eta = 0.7$ (Fault at bus 28 to 29 applied at $t = 5.0\text{sec}$)	35
Figure 6.6: Case-2 Generator Frequency -simulated with $\eta = 0.09$ (Fault at bus 28 to 29 applied at $t = 5.0\text{sec}$)	36

LIST OF TABLES

Table 5.1: Neural Network Architecture	30
--------------------------------------------------	----

CHAPTER 1: INTRODUCTION

1.1 Introduction to the Power System State Estimation

From the 19th century with the first Power System built in England it has kept growing throughout the world to light up billions of domestic and industrial entities, it is undeniably a fundamental part of the human life and advancement.

Fundamentally, Power System a big network of transmission lines, distribution, loads and generators. It is a highly dynamic system, with time varying loads and energy sources. At any given time, complete knowledge of operating conditions of the power system can be assessed by knowing variable quantities known as states. With respect to time evolution, states can be categorized into *static states* (slow changing) and *dynamic states* (Relatively fast as compared to Static States). States such as Voltage (V), Current (I), relative angles (θ), Real (P) and Reactive Power (Q) are Static States and Generator rotational speed (ω) & angle (δ) are the Dynamic States of the system.

Having accurate knowledge of the states is essential for control, monitoring and analysis of the system to ensure proper functioning of the system. In reality all the states of the system are not easily accessible it can be computed from the measurements (by knowing the relation between measurement and states). Many of the times measurements itself are corrupted by noise and/or accurate relation between measurement and state is not accurately known in such cases we have to rely on the best estimates of the states calculated using the state estimation algorithms.

Initially, Power System was assumed to be Quasi-Steady State. With the increasing deployment of Distributed Generators (DG), SMART loads, Demand Response (DR), Electric Vehicles Charging Units etc. dynamics of the Power Systems has become rapidly varying and the *quasi-stable* assumption will not be favorable to ensure the proper knowledge of the states. It has become im-

portant to know the Real-Time Electro-Mechanical States *dynamic states* of the system for better control, monitoring, protection and reliability of the system.

In traditional grid *static state* estimation was based on slow measurements (0.2-0.1 Hz) acquired through Supervisory Control and Data Acquisition (SCADA) system. Due to lack of fast measurements at faster rates it was not possible to perform *dynamic state* estimation. With introduction of rapid phasor measurement technology [8] [9] [10] based devices known as Phasor Measurement Units (PMU) it has become possible to avail rapid synchronized measurements of the wide area power system.

1.2 Phasor Measurement Unit (PMU)

Phasor is a complex quantity. In electrical often a sinusoidal associated with its relative angle. Phasor Measurement Unit (PMU) is an embedded system which incorporates sensing circuits, Digital Signal Processing (DSP) unit, Global Positioning System (GPS) and communication Unit. It computes phasors quantities such as Voltage, Current magnitudes and angle, Real & Reactive Power from the measurements at the system buses which are measured synchronously which are linked to the common time base using GPS; time accuracy of about $\pm 2 \mu$ sec [17]. This time stamped measurements are transmitted to control center, where all these measurements from various system nodes are processed to observe steady and dynamic states of the system.

Time stamp on the various PMUs are essential to synchronize measurements from the all the system buses where PMU is installed. At the time of writing, PMUs support sampling rate of up to 600Hz [9].

1.3 Literature Review

The traditional grid has for long been dependent on the synchronous generators. With increasingly sophisticated asynchronous Distributed Generation (DG) technologies such as solar, wind & storage devices with its unconventional benefits, is rapidly penetrating the power grid. Hence, power flow patterns which were traditionally assumed to be quasi-steady-state can no longer be predicted accurately with the same assumptions. Having accurate knowledge of the dynamic states the system in Real-Time is vital to the safe and stable operation of the power grid. Rotational angle and frequency of the synchronous generators are most important quantities for the transient stability and control. Traditionally, supervisory control and data acquisition (SCADA) based approaches were unable to capture rapidly changing electromechanical dynamics of the system (such as generators rotor angle and frequency) due to its slow sampling rates (usually 0.2-0.4 Hz).

Fortunately, with the advent of Phasor Measurement Units (PMUs) in tandem with Wide Area Measurement System (WAMS) technology, it has become possible to acquire rapid real-time measurements at the system nodes over a wide geographical area in synchronized manner, these available measurements are known as synchro-phasors. Time synchronization is possible due to GPS in PMUs which are synchronized to the same time clock. Measurements acquired from PMUs contains phasors such as Voltage, Current, Angle, and Real & Reactive Power which is transmitted in time stamped packets to the central control center to be processed in a synchronous manner in order to avail the dynamic states of the system.

Dynamic State Estimation (DSE) of the Synchronous generators is a nonlinear problem. For linear Kalman Filter based approach deems highly compromising in terms of accuracy. Therefore, numerous approaches for nonlinear DSE has been proposed and studied in the literature [4, 5, 7, 22, 24]. Which, includes Extended Kalman Filter (EKF) [22], Unscented Kalman filtering (UKF) [4–6, 16, 19]. EFK linearizes the system model at the current state estimate which assumes

state trajectories to be linear around the neighborhood which may result in erroneous estimates. Moreover, linearization requires computations of the Jacobian matrices at each iteration; which increases the computational overhead. To which UKF is presented to be a derivative free solution for this problem which does not require linearization instead it uses nonlinear transformation on states and covariances for approximating the Probability Distribution Function to the Gaussian distribution. Which also requires generation of sigma points. Even though UKF and EKF has a good performance it suffers from the Curse of Dimensionality problem as dimension of state space increases. Moreover, for Kalman Filter to produce optimal estimate its crucial to have accurate knowledge of the system model and noise statistics.

In the recent literature [3], few AI based methods have been investigated AI based methods are shown to be more effective, since, it doesnt require having knowledge of the exact model and parameters, which limits the performance of many of the conventional methods. Neural network is widely known for its universal function approximation capabilities. So, efficient algorithm can be designed to un-tap its potential in accurately estimating states of the highly nonlinear systems; furthermore it can be adapted on-line to learn the adapting parameters of the system without explicitly knowing the mathematical model for the system. However, these techniques needs off-line supervised learning [?, 3], which requires large good quality data which should well covers every possible scenario beforehand to carry out neural network training. After training neural network stops adapting and hence becomes static; which may not respond to the disturbances.

Algorithm proposed in this paper uses a small network which learns and adapts in Real-Time, which serves as a plug n play ad-hoc algorithm which doesnt require any prior training. Adaptive neural network also responds to the changing environment which makes it an efficient method.

This thesis is organized in following structure. Section-II discusses Neural Network modeling. Section-III discusses problem formulation; which includes power system dynamic and algebraic

modeling. Section-IV discusses the proposed ANN based DSE method. Section-V provides the result of the proposed method on various simulation and test cases.

1.4 Thesis Organization

The organization of the thesis is as follows. Chapter-2 discusses the State of the art Neural Network techniques with its variants. Chapter-3 discusses the Power System Network and power generator model along with the test model used for the simulation. Chapter-4 Discusses the Nonlinear State estimation algorithm using Neural Network which is developed to estimate generator angle and angular frequency along with the simulation results.

CHAPTER 2: ARTIFICIAL NEURAL NETWORK

An Artificial Neural network (ANN) is a state of the art Machine Learning algorithm which known for its universal function approximation capability. Machine Learning is a branch of Artificial Intelligence (AI) which gives computers ability to adapt and learn without being explicitly programmed. Depending on the nature of learning. There are three forms of training methodologies as follows .

2.1 Types of Training

In supervised learning pair of correct behavior y and inputs x are known beforehand the task is to learn function f such that error between y_a and y is minimized, where $y_a = f(x)$.

In unsupervised learning correct behavior y for the given input x is not known beforehand and in such scenarios algorithm is programmed to analyze certain features of the input data to categorize or to predict the output y .

Reinforcement learning is a Reward/ Penalty based algorithm where x and z are known and the task is fit function f to find correct y .

$$y = f(x) \quad z = f(y) \quad (2.1)$$

2.2 Design of Neural Network

2.2.1 Single Neuron

Neural network is at the center of the State Estimation algorithm we are going to develop in the subsequent chapters. Artificial Neural Network (ANN) is the state of the art algorithm widely known for its universal function approximation capabilities. In general, ANN is composed of basic processing unit called as neuron as shown in the in the *figure (2.2)*. Each neuron has an output, an activation function, a summer, a bias and can have multiple inputs with associated weighted links (known as weights denoted by 'w').

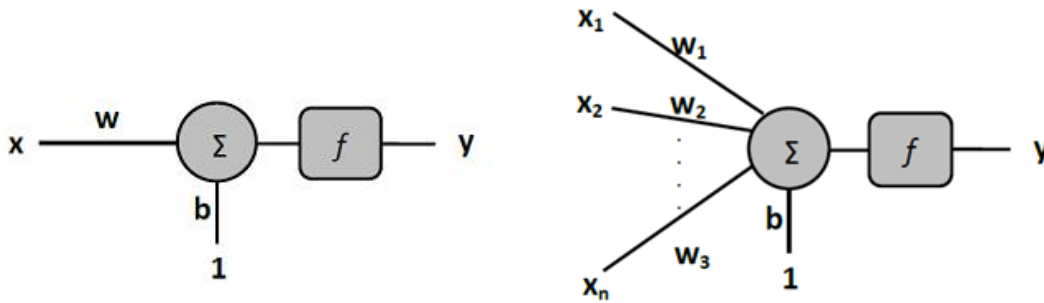


Figure 2.1: Single Neuron

For N input neuron with sigmoid activation function $\sigma(x)$ output y is calculated as,

$$s = \sum_{i=1}^N x_i w_i \quad y = \frac{1}{1 + e^{-s}} \quad (2.2)$$

Here, x is the input vector and w is the weight vector.

2.2.2 Single Hidden Layer Neural Network

Function approximation capabilities can be enhanced by adding a multiple neuron in the network like structure layer. Such multiple inputs, multiple outputs, single hidden layered neural Network is depicted in the *figure (2.2)*.

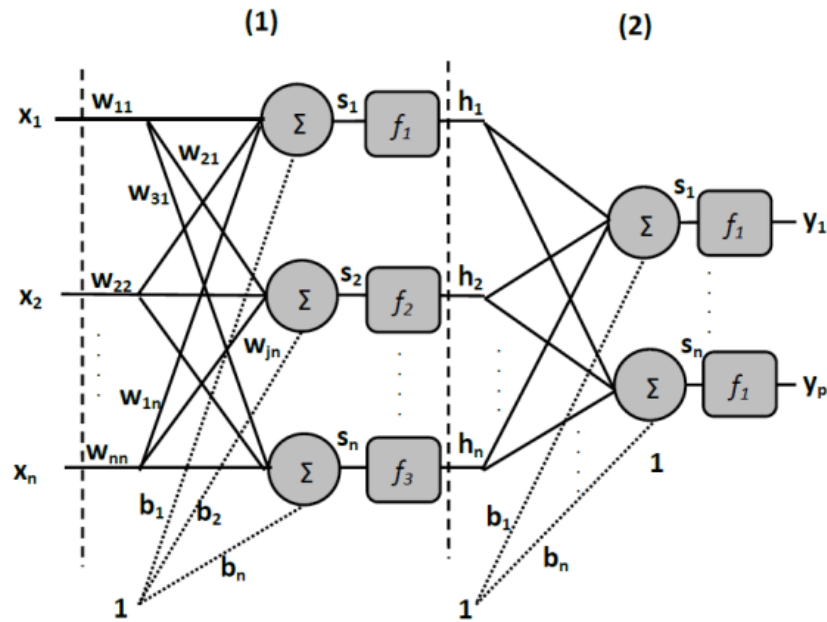


Figure 2.2: Single Hidden Layer Neural Network

Data from the input layers are multiplied by corresponding weights, summed together and passed through the activation function to produce outputs of the hidden layer. Output of the hidden layer is applied to subsequent layer and multiplied with corresponding weights to produce hidden outputs this process is iterated till the output of final layer is generated completing a *forward pass* or also known as *forward propagation*.

2.2.2.1 Forward Propagation

So, as explained before forward propagation is process of calculating outputs based on inputs. For an ANN with N inputs, M outputs, and H unit single hidden layers; with sigmoid activation function $\sigma(x)$ in the hidden layer and linear output layer as depicted in the *figure (2.2)*, output y of the network when presented input vector x can be calculated using the *equation.(2.3) - equation.(2.5)*.

$$s_j = b_j + \sum_{i=1}^N w_{ji}^{(1)} x_i \quad j \in \{1, H\} \quad (2.3)$$

$$h_j = \sigma(s_j) = \frac{1}{1 + e^{-s_j}} \quad j \in \{1, H\} \quad (2.4)$$

$$y_j = \sum_{i=1}^M h_i w_{ij}^{(2)} + b_j^{(2)} \quad j \in \{1, M\} \quad (2.5)$$

Here, considered neural network is single hidden layer network. Where, x is an input vector y is the output vector. $w^{(1)}$ and $w^{(2)}$ are the weight matrices of the input to hidden and hidden to output layer respectively. $w_{ji}^{(1)}$ denotes the weight (scalar) connecting j^{th} hidden neuron to the i^{th} input. $b_j^{(1)}$ is the bias of the j^{th} neuron in the hidden layer. $b_j^{(2)}$ is the bias in stage two i.e. of the output layer.

Choice of activation function is usually chosen to be a continuously differentiable nonlinear or linear function. Sigmoid is the most commonly used activation function in many applications. Output layer in our case is linear and other layers are sigmoid.

2.2.2.2 Objective Function

Learning of the neural network is carried iteratively using based on the training methodologies discussed until the desired performance objective is met. Performance objective for the optimization algorithm is often defined in terms of expectation of the Least Mean Square (LMS) measure. Performance objective is slightly modified here in back-propagation algorithm to make it simpler, following is the general objective function of the neural network training, and for M outputs can be defined by as,

$$J = \frac{1}{2} \sum_{i=1}^M (y_{a_i} - y_{nn_i})^2 \quad (2.6)$$

We will see J with a subscript J_k most of the time denoting cost at the time instance k , after using subscript explicitly it may be dropped thereafter just to make expression easily representable.

In order to minimize the objective function, error between actual output y_a and neural network output y_{nn} is back-propagated through chain rule of derivate (known as back-propagation algorithm) and the weights are updated using gradient descent based optimization, this algorithm is iterated till the desired cost objective is achieved. Goal can be to minimize or maximize the objective function based on the formulation.

Objective as a function of weights and biases is not quadratic in nature due to nonlinear activation function (sigmoid). Depending on the size of the network, surface of the cost function becomes even more complex. Also, parameters of the network are initialized randomly which is important to ensure neural network don't get stuck in the same local minima each time parameters re-initialized.

Initial network parameters are the crucial factor on which convergence time is dependent. In some cases if desired performance is not met within the reasonable time then network parameter can be

reinitialized and algorithm is re-run until the desired performance is achieved.

2.2.2.3 Back-Propagation

2.2.2.3.1 Chain Rule

Output of the neural network is function of weights and biases which are variable parameters used to tune the network until the desired objective is met. Back propagation is nothing but the chain rule of derivative as given in the *equation.(2.7)*, which transfers error from the output layer to the weights and biases.

$$\frac{\partial J}{\partial W} = \frac{\partial J}{\partial y} \cdot \frac{\partial y}{\partial h} \cdot \frac{\partial h}{\partial s} \cdot \frac{\partial s}{\partial W} \quad (2.7)$$

$$\frac{\partial J}{\partial y_i} = \frac{1}{2} \cdot \frac{\partial}{\partial y_i} \sum (t_i - y_i)^2 = (y_i - t_i) \quad i = 1, 2, \dots, M \quad (2.8)$$

$$\frac{\partial y}{\partial h} = \frac{\partial}{\partial h} \sigma(x) = \sigma'(x) = h(s)(1 + h(s)) \quad (2.9)$$

$$\frac{\partial y}{\partial h} = 1 \quad \text{for linear output} \quad (2.10)$$

$$\frac{\partial h}{\partial s} = \frac{\partial h}{\partial s} h(s)(1 + h(s)) \quad (2.11)$$

$$\frac{\partial s}{\partial w} = h(s) \quad (2.12)$$

$$\frac{\partial s}{\partial a} = W \quad (2.13)$$

2.2.2.3.2 Gradient Descent

Once we have corresponding error value for each parameter (weight and bias) it can be updated using the optimization algorithm. Gradient descent (*equation.(2.14)*) is a simple yet effective optimization algorithm used to optimize parameters based on the gradient obtained using back-propagation.

$$W = W - \gamma \frac{dJ}{dW} \quad (2.14)$$

2.2.2.3.3 Learning Rate

Here, γ is a positive constant known as learning rate. Usually, $\gamma < 1$. If the γ is too low learning algorithm takes too long to converge, If gamma is too high solution can be obtained fast but with there's high probability of divergence. It should be chosen to be moderate and depends on many factors such as size of the neural network, objective function etc.

Its a good practice to start with small value such as 0.001 and increase it by the factors of 10 till the satisfactory performance is achieved and then it can be fined tuned.

CHAPTER 3: POWER SYSTEM MODELING

3.1 Transient

Power System is a network of power source, transmission & distribution systems and loads.

In steady state form generator is expected to run at a constant frequency known as a synchronous frequency (ω_{sync}) and constant rotor angle (δ) relative to rotor angle of the swing bus generator. Hence in the steady state generator dynamics $d\delta/dt$ and $d\omega/dt$ are expected to be zero. Ideally, in the steady state in mechanical power P_m matches the electrical power P_e ; difference $P_m - P_e$ is zero.

As any parameter such as input torque, load at the Electrical terminal (P_e) or any internal machine parameter, short circuit in the generator happens it causes the imbalance between P_m and P_e and causes non-zero $d\omega/dt$ and consequently Nature of this disturbance can be seen as the swinging oscillations over nominal values of ω and δ called as Swing or Electromechanical Oscillations of the Power System (OEs). Effects can also be directly seen on V and θ of the buses.

Commonly, Power system has multiple such interconnected generators serving shared loads, any type of disturbance is propagated through the network with the electro-mechanical velocity; less than few micro-seconds.

3.2 Generator Model

3.2.1 Swing Model

For a multi-machine power system model of i^{th} synchronous generator can be modeled using a set of nonlinear ordinary differential equation (ODE) *equation.(3.1)* (also known as the classical model of the generator or swing model). Swing model basically governs the swing of the generators shaft in case of imbalance between P_m and P_e . Usually this imbalance is caused in the overload at the terminals, internal short circuit, faulty situations etc.

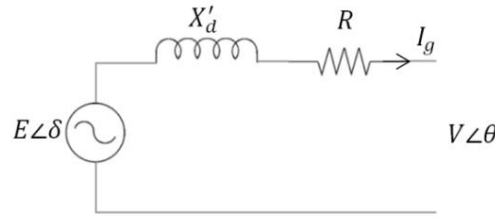


Figure 3.1: Generator Equivalent Model

The synchronous machine can be represented by the simplified electrical model known as classical model as in the *figure (3.1)*. Here, R is considered to be negligibly small and hence $R = 0$. X'_d is the d-axis reactance of the rotor. I_g is the current injected by generator into the grid.

$$\begin{aligned} \frac{d\omega_i}{dt} &= \frac{\omega_s}{2H} [P_{m_i} - P_{e_i} - D_i(\omega_i - \omega_s)] \\ \frac{d\delta_i}{dt} &= \omega_B(\omega_i - \omega_s) \end{aligned} \quad (3.1)$$

Where,

$$P_{e_i} = \frac{|E_i||V_i|}{X'_d} \sin(\delta_i - \theta_i) \quad (3.2)$$

3.2.2 Swing Equation numerical Integration (Rectangular Rule)

Swing governed by the swing model above is continuous in nature, for simulation purposes swing equation is discretized and numerical integration is performed which approximates the continuous behavior in the simulated environment. Numerical integration techniques such as Euler's method or rectangular rule, Runga-Kutta method can be used. Here in *equation.(3.3)* forward rectangular method is used.

After sampling *equation.(3.1)* at a sampling interval T with forward rectangular based numerical integration we obtain discretized form of the swing model as,

$$\begin{aligned}\omega_{i_{k+1}} &= \omega_{i_k} + T \frac{\omega_s}{2H_i} [P_{m_i} - P_{e_{i_k}} - D_i(\omega_{i_k} - \omega_s)] \\ \delta_{i_{k+1}} &= \omega_B(\omega_i - \omega_s)\end{aligned}\tag{3.3}$$

Here, i is the generator index, δ is generator angle in radians (also known as the power angle), ω is generator angular frequency, ω_B is the base frequency $\omega_B = 2\pi f$, (here $f = 60\text{Hz}$), ω_s is angular frequency of swing generator, P_m is the mechanical power (assumed to be constant), P_e is the electrical power, E is the internal emf the generator X'_d is the transient reactance seen by the generator, D is the damping co-efficient of the generator. V is the voltage at the bus node and θ is the bus angle.

Where, T is the simulation step time. Numerical Accuracy can be improved further by reducing simulation step time T and/or by using numerical integration techniques such as Modified Euler's method, 4th order Runge-Kutta method etc.

3.3 Network Model

3.3.1 16 Generator 64 Bus Model

The network model used here for simulation is the New England Test System (NETS)/New York Power System (NYPS) 16-generator 64-bus system model [13]. This model is simulated in MATLAB for 20.0 seconds with the simulation step of 0.02 sec. Three phase fault applied at the bus number 28 on line between 28 to 29 at 5.10s and cleared at 5.15s.

3.3.2 Measurement Model

Measurement model for the Multi-machine power system consisting of the m bus n generators in terms of bus voltage and angles can be given by equation (3.7) relates generators internal voltage and angle ($E\angle\delta$) to the voltage and angle ($V\angle\theta$) at bus nodes by the following algebraic equation. The relation can be obtained using the expanded system matrix model *equation.(3.4)* [24].

$$\begin{bmatrix} E\angle\delta \\ V\angle\theta \end{bmatrix} = \begin{bmatrix} Y_{GG} & Y_{GL} \\ Y_{LG} & Y_{LL} \end{bmatrix} \begin{bmatrix} E\angle\delta \\ V\angle\theta \end{bmatrix} = \begin{bmatrix} I_G\angle\delta \\ 0 \end{bmatrix} \quad (3.4)$$

Where, Y_{exp} is the expanded nodal matrix, comprises of Y_{GG} , Y_{GL} , Y_{LG} and Y_{LL} sub-matrices which are generator, generator to load, load to generator and load to load admittance matrices respectively, I_G is the current flowing out of the generator terminal. $R_V = Y_{LL}^{-1}Y_{LG}$ can also be defined as reconstruction matrix.

$$V = Y_{LL}^{-1}Y_{LG}E \quad (3.5)$$

$$V\angle\theta = R_V E\angle\delta \quad (3.6)$$

$$\begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_m \end{bmatrix} = \begin{bmatrix} R_{V_{11}} & R_{V_{11}} & \dots & R_{V_{1n}} \\ R_{V_{21}} & R_{V_{22}} & \dots & R_{V_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ R_{V_{m1}} & R_{V_{m2}} & \dots & R_{V_{mn}} \end{bmatrix} \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_m \end{bmatrix} \quad (3.7)$$

Measurements are represented in rectangular co-ordinates in the formulation in the subsequent chapters so real and imaginary part of the measurement function obtained from *equation.(3.7)* is obtained as given in *equation.(3.8)* to *equation.(3.9)*.

$$\begin{aligned} Re(V_i) = |R_{V_{i1}}||E_1|\cos(\angle R_{V_{i1}+\delta_1}) + |R_{V_{i2}}||E_2|\cos(\angle R_{V_{i2}+\delta_2}) + \\ \dots + |R_{V_{in}}||E_n|\cos(\angle R_{V_{in}+\delta_n}) \end{aligned} \quad (3.8)$$

$$\begin{aligned} Im(V_i) = |R_{V_{i1}}||E_1|\sin(\angle R_{V_{i1}+\delta_1}) + |R_{V_{i2}}||E_2|\sin(\angle R_{V_{i2}+\delta_2}) + \\ \dots + |R_{V_{in}}||E_n|\sin(\angle R_{V_{in}+\delta_n}) \end{aligned} \quad (3.9)$$

CHAPTER 4: KALMAN FILTER

4.1 Kalman Filter

Kalman Filter is a widely known state estimation algorithm for estimating the correct states based on noisy measurements obtained over time. It works in two steps, prediction and correction step. Prediction step uses system model such as *equation.(3.3)* to predict the next state and error covariance. Correction step uses measurements and error to calculate accurate state.

Kalman filter is an optimal estimator in a sense that it minimized the mean of the squared error given system model is linear. Many Kalman Filtering based approaches has been developed in the literature. Power system swing model discussed in our case is highly nonlinear and linear approximation performs poorly. Which requires nonlinear variants of the kalman filter such as Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF). This chapter develops EKF and UKF based formulations which are used to compare againts Neuro-Adaptive Learning based algorithm.

4.2 Extended Kalman Filter (EKF)

Extended Kalman filter is also a similar two-step Kalman Filter variant, In its formulation it uses nonlinear system for the state prediction with the linearized system based error covariance update. We have used the similar formulation of EKF as in the [23], in this thesis only stage-2 approach as given in [23] is considered.

In kalman filter system is modeled as a stochastic process as given in *equation.(4.1)*.

$$\frac{dx}{dt} = f(x, u, t, w) \quad (4.1)$$

Here, $x \in R^n$ is a n dimensional state vector, $u \in R^p$ is input and $w \in R^n$ is noise vector. State prediction of EKF is calculated based on *equation.(4.2)*, for P_k^- system is linearized at \hat{x}_k , which requires calculation of Jacobian matrices A_k . Also H_k for calculation of K_k in *equation.(4.3)*.

$$\text{Prediction} \begin{cases} \hat{x}_k^- = f(\hat{x}_{k-1}, u_{k-1}, 0) \\ P_k^- = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T \end{cases} \quad (4.2)$$

$$\text{Correction} \begin{cases} K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V_k^T)^{-1} \\ \hat{x}_k = \hat{x}_k^- + K_k (z_k - h(\hat{x}_k^-, 0)) \\ P_k = (I - K_k H_k) P_k^- \end{cases} \quad (4.3)$$

4.2.1 Jacobian Matrices

Jacobian matrix here is the partial derivatives of function with respect to systems states. Matrix A is a $n \times n$, H is $m \times n$, V and W are $n \times n$ matrices. As given in the *equation.(4.4)* jacobian matrices are calculated at current estimate \hat{x}_k .

$$A = \left. \frac{\partial f(x)}{\partial x} \right|_{\hat{x}_k} \quad W = \left. \frac{\partial f(x)}{\partial x} \right|_{\hat{x}_k} \quad H = \left. \frac{\partial h(x)}{\partial x} \right|_{\hat{x}_k} \quad V = \left. \frac{\partial h(x)}{\partial x} \right|_{\hat{x}_k} \quad (4.4)$$

4.2.1.1 System Matrix A

A_c is the continuous time system matrix. Entries of the matrix A_c are the partial derivatives of the generator model with respect to states in this case δ and ω . Elements of A_c are calculated as given in equation.(4.5)-equation.(4.12).

$$A_{c_{[2i-1,2i-1]}} = 0 \quad (4.5)$$

$$A_{c_{[2i-1,2i]}} = \omega_B \quad (4.6)$$

$$A_{c_{[2i,2i-1]}} = -\frac{\omega_0 |E_i|}{2H_i X'_{d_i}} \left[\cos \delta_i \operatorname{Re}(V_i) + \sin \delta_i \frac{\partial \operatorname{Re}(V_i)}{\partial \delta_i} + \sin \delta_i \operatorname{Im}(V_i) - \cos \delta_i \frac{\partial \operatorname{Im}(V_i)}{\partial \delta_i} \right] \quad (4.7)$$

$$A_{c_{[2i,2i]}} = -\frac{\omega_0}{2H_i} D_i \quad (4.8)$$

$$A_{c_{[2i-1,2j-1]}} = 0 \quad (4.9)$$

$$A_{c_{[2i-1,2j]}} = 0 \quad (4.10)$$

$$A_{c_{[2i,2j-1]}} = -\frac{\omega_0 |E_i|}{2H_i X'_{d_i}} \left[\sin \delta_i \frac{\partial \operatorname{Re}(V_i)}{\partial \delta_j} - \cos \delta_i \frac{\partial \operatorname{Im}(V_i)}{\partial \delta_j} \right] \quad (4.11)$$

$$A_{c_{[2i,2j]}} = 0 \quad (4.12)$$

To compute on a digital computer system is discretized hence, transition matrix of the system is given by *equation.(4.13)* and *equation.(4.14)*. Here in *equation.(4.13)* and *equation.(4.14)* forward rectangular rule is used to carry out numerical integration.

$$x_k = Ax_{k-1} + w_{k-1} \quad (4.13)$$

$$A = I + A_c T \quad (4.14)$$

Here, T is the sampling time.

4.2.1.2 Measurement Matrix H

Similarly, measurement function $h(x, t)$ is linearized around \hat{x} to avail linearized measurement model. Here measurement vector $y = [Re(V_1)Im(V_1)Re(V_2)Im(V_2)...Re(V_m)Im(V_m)]$. Individual terms of the matrix H are given in the *equation.(4.16)* -*equation.(4.18)* which are obtained by taking partial derivatives of the *equation.(3.8)* and *equation.(3.9)* with respect to states δ and ω .

$$H = \begin{bmatrix} \frac{\partial ReV_1}{\partial \delta_1} & \frac{\partial ReV_1}{\partial \omega_1} & \frac{\partial ReV_1}{\partial \delta_2} & \frac{\partial ReV_1}{\partial \omega_2} & \dots & \frac{\partial ReV_1}{\partial \delta_n} & \frac{\partial ReV_1}{\partial \omega_n} \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ \frac{\partial ReV_2}{\partial \delta_1} & \frac{\partial ReV_2}{\partial \omega_1} & \frac{\partial ReV_2}{\partial \delta_2} & \frac{\partial ReV_2}{\partial \omega_2} & \dots & \frac{\partial ReV_2}{\partial \delta_n} & \frac{\partial ReV_2}{\partial \omega_n} \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\partial ReV_n}{\partial \delta_1} & \frac{\partial ReV_n}{\partial \omega_1} & \frac{\partial ReV_n}{\partial \delta_2} & \frac{\partial ReV_n}{\partial \omega_2} & \dots & \frac{\partial ReV_n}{\partial \delta_n} & \frac{\partial ReV_n}{\partial \omega_n} \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 \end{bmatrix} \quad (4.15)$$

$$\frac{\partial Re(V_i)}{\partial \delta_j} = -|R_{V_{ij}}||E_j|sin(\angle R_{V_{ij}+\delta_j}) \quad (4.16)$$

$$\frac{\partial Im(V_i)}{\partial \delta_j} = |R_{V_{ij}}||E_j|cos(\angle R_{V_{ij}+\delta_j}) \quad (4.17)$$

$$\frac{\partial Re(V_i)}{\partial \omega_j} = \frac{\partial Im(V_i)}{\partial \omega_j} = 0. \quad (4.18)$$

4.3 Unscented Kalman Filter (UKF)

Unscented Kalman Filter is based on Unscented Transform technique devised by Uhlmann for approximating probability distribution passed through a non-linear function. It is based on the idea that it is easier to approximate probability distribution than approximating the nonlinear dynamic function.

Formulation of Unscented Kalman Filter is based on 4 steps *i)* Generation of Sigma Points *ii)* State Prediction *iii)* Measurements Prediction *iv)* Update

First step in UKF is to generate sigma points as given in *equation.(4.19) - equation.(4.20)*. Mean and co-variance of the sigma point is similar to mean and co-variance (P_x) of the actual state x . For the system with N dimensional state vector there are $2N$ sigma points.

Generation of Sigma Points

$$\chi_0 = \hat{x}_{k-1} \quad (4.19)$$

$$\chi_i = \hat{x}_{k-1} + \frac{1}{\eta}(\sqrt{Px})_i \quad i = 1, 2, \dots, N \quad (4.20)$$

$$\chi_i = \hat{x}_{k-1} - \frac{1}{\eta}(\sqrt{Px})_l \quad i = N + 1, \dots, 2N \quad (4.21)$$

Here, $(\sqrt{Px})_l$ is the l^{th} column vector of the matrix $(\sqrt{Px})_l$

In the step generated sigma points are given to the system model to generate $2N$ predicted states using sigma points based on which priori state estimate \hat{x}_k^- is calculated.

State Prediction

$$\hat{x}_k^- = \frac{1}{2N} \sum_{i=0}^{2N} f(\hat{\chi}_{ik-1}, u_{k-1}, 0) \quad (4.22)$$

$$P_{xx_k}^- = \frac{1}{2N} \sum_{i=0}^{2N} (\hat{\chi}_{ik} - \hat{x}_k^-)(\hat{\chi}_{ik} - \hat{x}_k^-)^T + Q_{k-1} \quad (4.23)$$

Measurements Prediction

In the measurement step similar approach as in the previous step is used. All the generated sigma points are used to generate $2N$ measurements and then mean measurement is calculated.

$$\hat{Z}_k = h(\hat{\chi}_k^-, 0) \quad (4.24)$$

$$\hat{z}_k = \frac{1}{2N} \sum_{i=0}^{2N} h(\hat{\chi}_{ik}^-, 0) \quad (4.25)$$

$$P_{yy_k}^- = \frac{1}{2N} \sum_{i=0}^{2N} (\hat{Z}_{ik} - \hat{z}_k^-)(\hat{Z}_{ik} - \hat{z}_k^-)^T + Q_{k-1} \quad (4.26)$$

$$P_{xy_k}^- = \frac{1}{2N} \sum_{i=0}^{2N} (\hat{Z}_{ik} - \hat{z}_k^-)(\hat{Z}_{ik} - \hat{\chi}_{ik}^-)^T + R_{k-1} \quad (4.27)$$

Update

$$K_k = P_{xy}P_{yy}^{-1} \quad (4.28)$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - \hat{z}_k) \quad (4.29)$$

$$P_{xx_k} = \hat{P}_{xx_k}^- - K_k P_{yy} K_k^T \quad (4.30)$$

4.3.1 Numerical Stability

Although UKF is computationally efficient as compared to EKF and has an advantages over EKF as it does not need linearization. It has its own drawback for higher dimensional systems. As number of states increases above twenty there will be higher degree of nonlinearity which may cause instability in the UKF approach [12].

Specifically, in UKF as dimension of state space increases positive definiteness of P_{xx} matrix cannot be ensured. Which is required to calculate square root based cholesky based factorization. So, to ensure the positive definiteness of the matrix nearest Semi-positive definite matrix P_{xx} is calculated in each step as discussed in [12].

CHAPTER 5: NEURO-ADAPTIVE DYNAMIC STATE ESTIMATION

5.1 Introduction to Dynamic State Estimation

State estimation is basically a problem of inferring the internal unknown state based on the available related noisy measurements. Estimate of the state x is denoted by \hat{x} . More specifically $\hat{x}_{j|k}$ is the estimate of the state x at time instance j based on measurements y_k at or upto instance k . When, $j > k$, $x_{j|k}$ is called a *smoothed* estimate, when $j = k$, $x_{j|k}$ is called a *filtered* estimate and when $j < k$, $x_{j|k}$ is called a *predicted* estimate.

Dynamic State Estimation in Power System has been an active research lately. Its very important to know correct state of the system to take appropriate control and monitoring action to ensure reliability and stability of the Power System. With advancement in the Phasor Measurement Units (PMU) in the Power System it has become easily possible to get samples of the Voltages and Angles at the systems nodes at much higher rates. This has made it possible to know the dynamic states of the system.

5.2 Neuro Adaptive Dynamic State Estimation

In this subsection we introduce the Neuro-Adaptive Dynamic State Estimation Algorithm (NA-DSE) as depicted in the *figure (5.1)* where we need the models developed in the previous subsections.

5.2.1 Problem Formulation

In the proposed algorithm Feed Forward Neural Network (FFNN) is used to predict state estimate $\hat{\delta}_{k+1}$ and $\hat{\omega}_k$ at instant k as depicted the fig (5.1) & algorithm (1). Input to the neural network are the bus voltages at the generator terminals V and angle θ available from the PMUs. Neural network outputs are the estimated states; generators internal angles δ (a step-ahead prediction) and angular frequency ω of the generator shaft.

$$\begin{bmatrix} \hat{\delta}_{j,k+1} \\ \hat{\omega}_{j,k} \end{bmatrix} = f_{NN}(W_{NN}, \begin{bmatrix} V_k \\ \theta_k \end{bmatrix}) \quad (5.1)$$

Here, W_{NN} is weight and biases of the neural network, k is the current time instance, $j \in 1, m$ is the generator. This algorithm utilizes a step-ahead prediction for the angle estimation $\hat{\delta}_{k+1}$.

In traditional supervised learning approach it is essential to generate representative training set of thousands of input output pairs; which should necessarily include all the possible fault case scenarios. All of these is required beforehand to train neural network in off-line mode. Apparently, neural network needs to be updated in case of any measure changes in the systems configuration.

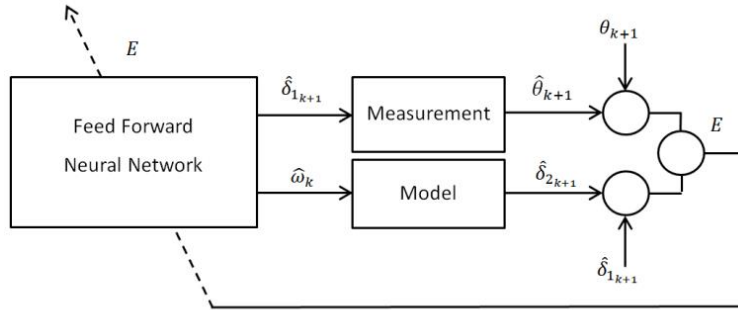


Figure 5.1: Neuro-Adaptive Dynamic State Estimation Algorithm

In contrast, in the proposed algorithm adaptation of the ANN is carried out instantaneously in Real-Time. The major issue with for state estimation with ANN in real time is that it does not have true state available to training. So, problem in this approach is to design an appropriate error measure so that correct error is back propagated in order to minimize the difference between estimated state and actual state.

For this purpose, objective function is designed based on the instantaneous measurements y_k which is a function of state x_k as given in the given in the *equation.(5.3)* and *equation.(5.4)*. Formulation of the algorithm is similar to two step approach in Kalman Filter. Neural Network Estimates the current state based on the measurement and system model is used to predict the next state. Subsequently, next step measurements are estimated as a function of states and error is computed as the actual measurements are availed from the PMU. Instantaneous error is back-propagated to ANN1 and ANN2 which learn to minimize the measure in equation (5.2). Weights and biases are updated using the gradient descent based optimization.

5.2.2 Learning Objective & Algorithm

Since, actual state is not available in state estimation algorithm supervised learning based objective function cannot be used. Instead y_k which is a function x_k is used to formulate the objective function for learning. This approach is called as reinforcement learning.

Here, in the equation (5.2) J_1 & J_2 are the objective function for δ and ω training respectively. Subscript k denotes the training instant, which can be subsequently be dropped after mentioning explicitly. $\eta < 1$ is the weight factor for the error accumulation term, also can be seen as a memory of the objective function, varying η will affect the response of the estimation. Typical value of the

eta is suggested to be around 10γ for good results in this particular case.

$$J_{j_k} = \alpha_j E_{j_k} + \eta_j \sum_{i=1}^k E_{j_i} \quad j \in \{1, 2\} \quad (5.2)$$

$$E_{1_k} = \sum_{j=1}^n \frac{1}{2} (\theta_{j_{k+1}} - \hat{\theta}_{j_{k+1}})^2 \quad (5.3)$$

$$E_{2_k} = \sum_{j=1}^n \frac{1}{2} (\delta_{j_{k+1}} - \hat{\delta}_{j_{k+1}})^2 \quad (5.4)$$

ANN can be seen as a short term local function approximator, hence, need to use of large multi-layer ANN is obviated. Hence, quick response is achieved, with the sufficiently small learning rates. Clearly, ANN can be a single hidden layer network with moderate number of neurons in the hidden later to achieve accurate nonlinear function mapping. Empirically based on error measures size of the hidden layer is determined to be approximately half the size of input to achieve good results.

Supervised learning is not a particularly necessary or recommended in this approach, unless large and deep network is being used.

5.2.3 Neural Network Design

Size of the neural network depends used in this paper is 32 and 16 hidden layers for δ and ω respectively with 32 input 16 output in both the cases. There are 2 neural networks for δ and ω estimation each has 32 inputs. ANN1 has $[V_1\theta_1 V_2\theta_2 \dots V_{32}\theta_{32}]^T$ as inputs and $x = [\delta_1\delta_2 \dots \delta_{32}]^T$ as output. ANN2 has $[\delta_1\theta_1\delta_2\theta_2 \dots \delta_{32}\theta_{32}]^T$ Gradient descent algorithm based numerical optimization has been used in Back-Propagation based learning.

Algorithm 1: NA-DSE Real Time Adaptive Learning Algorithm

beginnote: inputs and outputs are in vector form **INITIALIZATION**

$$W_{NN_1} \leftarrow N(0, \sigma^2 = 0.1^2)$$

$$W_{NN_2} \leftarrow N(0, \sigma^2 = 0.01^2)$$

$$0 < \alpha \leq 1 \quad 0 \leq \eta < 1 \quad \gamma_1, \gamma_2 < 1$$

$$\hat{\delta}_0 = 1_{n \times 1}$$

$$\hat{\omega}_0 = 1_{n \times 1}$$

while $k \geq 1$ **do****STEP-1 (ESTIMATE STATES)**

$$\hat{\delta}_{k+1} = f_{NN_1}(V_k, \theta_k) \quad \text{-- PREDICT}$$

$$\hat{\omega}_k = f_{NN_2}(\delta_k, \theta_k) \quad \text{-- ESTIMATE}$$

STEP-2 (PREDICT NEXT STATES)

$$\hat{\delta}_{k+1} = f_{model}(\hat{\delta}_k, \hat{\theta}_k)$$

STEP-3 (PREDICT MEASUREMENT)

$$\hat{\theta}_{k+1} = h_{meas}(E \angle \hat{\delta}_{k+1})$$

STEP-4 (CALCULATE ERROR)

$$E_{1k} = \sum_{j=1}^n \frac{1}{2} (\theta_{j_{k+1}} - \hat{\theta}_{j_{k+1}})^2$$

$$E_{2k} = \sum_{j=1}^n \frac{1}{2} (\delta_{j_{k+1}} - \hat{\delta}_{j_{k+1}})^2$$

$$J_{jk} = \alpha_j E_{jk} + \eta_j \sum_{i=1}^k E_{ji} \quad j \in \{1, 2\}$$

STEP-5 (ADAPT NEURAL NETWORK)

$$W_{NN_1} := W_{NN_1} - \gamma_1 * \frac{\partial J_1}{\partial W_{NN_1}}$$

$$W_{NN_2} := W_{NN_2} - \gamma_2 * \frac{\partial J_2}{\partial W_{NN_2}}$$

$$k \leftarrow k + 1$$

Introducing term η makes neural network comparatively stable and allows smaller learning rate which makes it less sensitive to the noise.

Table 5.1: Neural Network Architecture

Parameters	ANN1- δ	ANN2- ω
Hidden	32	16
Inputs	32	32
Outputs	16	16
γ	0.07	0.04
α	0.4	0.6
η	0.7	0.09

CHAPTER 6: SIMULATIONS AND RESULTS

Three test scenarios have been considered here to carry out the simulation. Performance of the proposed method is compared with the Extended Kalman and Unscented Kalman based methods. It is evident from the results that computational complexity for the proposed method is around 64% as that of the EKF method.

More real-life system is used for simulation purpose which is a 16 generators 68 bus NETS-NYPS system. Test has been simulated for 10 seconds with the time step of 0.01 seconds. Parameter such as H-inertia, D-Damping ratio is assumed to be known. In almost all of the real life cases PMU measurements at generator buses are measured and available so, in this scenario generator buses measurements are assumed to be available.

6.1 Test Case Studies

6.1.1 Case -1

Three phase fault at the bus 29 (from 28-29) has been generated at $t=1.1s$ and cleared at $t = 1.15 sec$. Single hidden layer Feed Forward ANN with sigmoid activation function in the hidden layer and linear output function is used. For 16 generators.

PMU measurements are assumed to be noisy with standard deviation of $1e-4$ per, also for extended kalman filter $w_k \sim N(0, W_k = 0.75 \times 10^{-5})$ and $v_k \sim N(0, V_k) = diag(1 \times 10^{-5}) P_{k_0} = 0_{n \times n}$. Where n is the number of states to be estimated. Performance measure of the estimation error is computed as a Root mean square error for time instance T_f for EKF and ANN and is calculated using the equation (6.1).

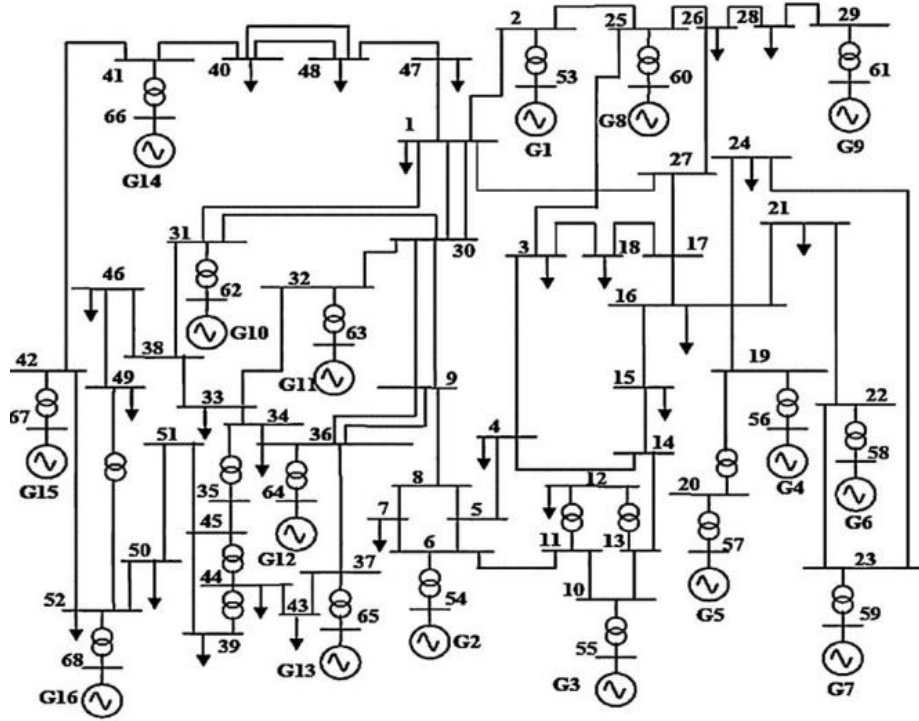


Figure 6.1: NETS-NYPS (16 Generators 68 Bus System)

In this case $\eta = 0$ so error accumulation is not considered and as can be seen from the figure (6.6) continuous offset in the estimation of ω and $\hat{\omega}$ can be observed. Introducing the error accumulation in the objective function reduces the this error as objective of ANN is to minimize the cost function. Hence, a better tracking can be achieved.

As compared to EKF ANN is closely tracks the state trajectories which is clearly visible from the *figure (6.1.2) & figure (6.6)*. Also, ANN based algorithm does not assume noise characteristics be known. Whereas performance of the Kalman Filter based approached depends on the knowledge of the accurately knowing the noise statistics.

$$E = \frac{1}{T_f} \sqrt{\sum (x_l^i - \hat{x}_k^i)^2} \quad (6.1)$$

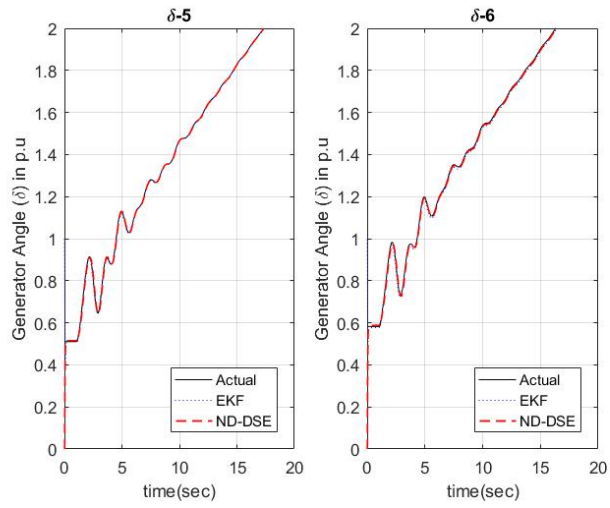


Figure 6.2: Case-1 Generator Angles - simulated with $\eta = 0$ (Fault at bus 28 to 29 applied at $t = 1.0$ sec)

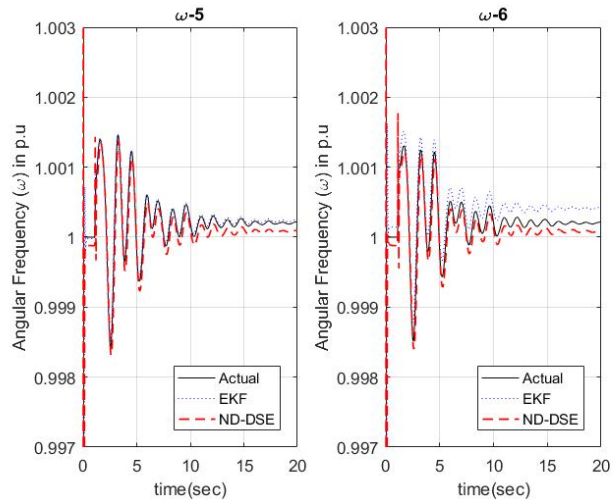


Figure 6.3: Case-1 Generator Speed - simulated with $\eta = 0$ (Fault at bus 28 to 29 applied at $t = 1.0$ sec)

6.1.2 Case -2

In this case formation of neural networks remains the same as discusses in the case-1, here in case-2 we have added an extra error accumulation term $\eta \sum E$ in the cost function as discussed early at the end of the case-1. Performance of the algorithm as can be seen from figures (6.1.2) & (6.6) can be seen to be quite improved.

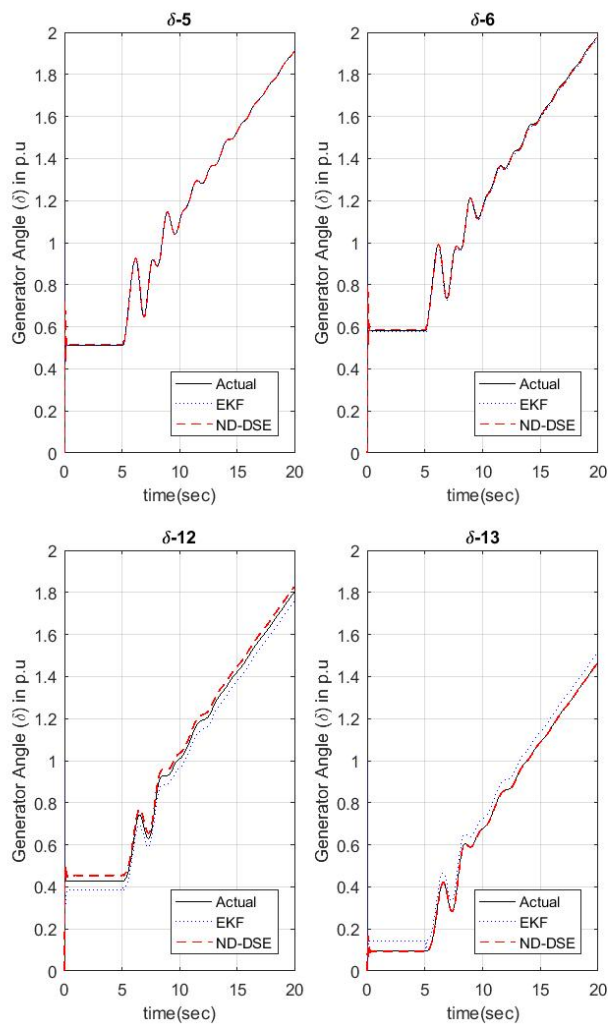


Figure 6.4: Case-2 Generator Angles - simulated with $\eta = 0.7$ (Fault at bus 28 to 29 applied at $t = 5.0$ sec)

Here, these parameters serves as a heuristic addition to twig the sensitivity of the ANN to the error where, α serves as a sensitivity of the ANN to the current error and η decides the memory of the accumulated error. Values of both of these factors depends on the factors such as size of the neural network, output scale, simulation time step. But, suggested values for α is between 0.5 to 0.7 and for η is 0.8 depending on the performance.

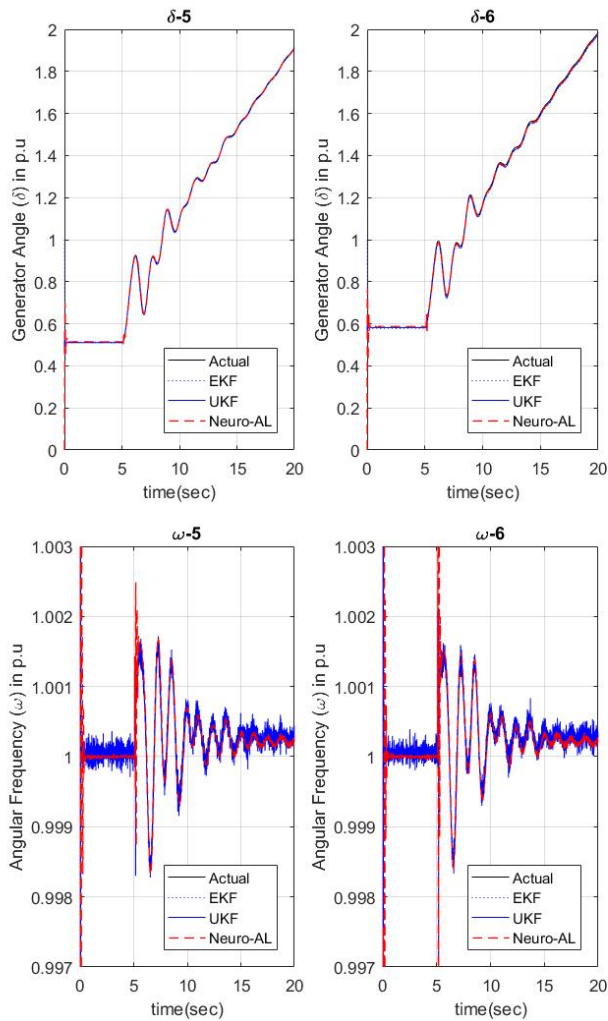


Figure 6.5: Case-2 Generator Angles - simulated with $\eta = 0.7$ (Fault at bus 28 to 29 applied at $t = 5.0$ sec)

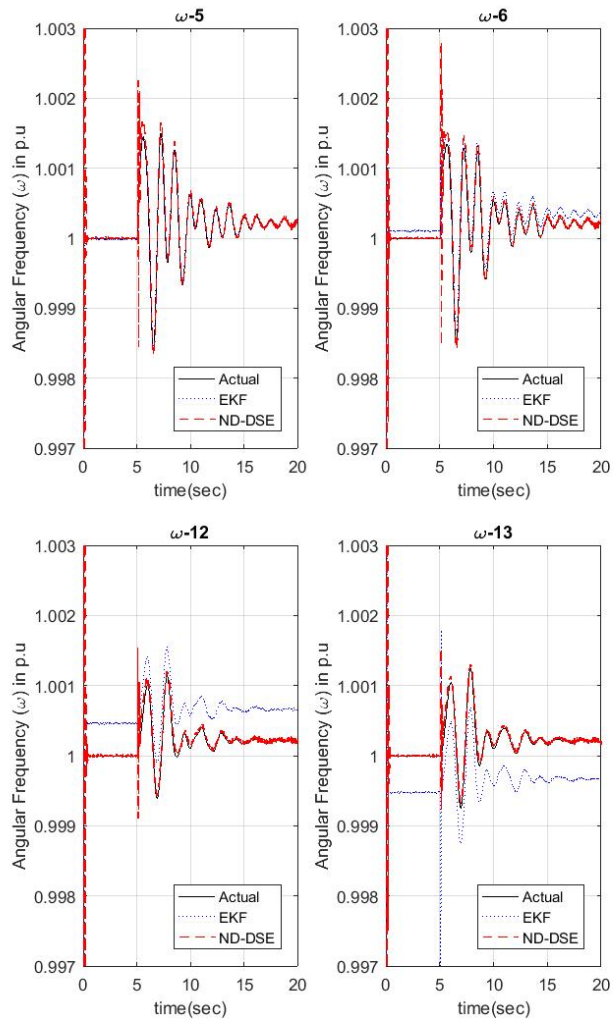


Figure 6.6: Case-2 Generator Frequency -simulated with $\eta = 0.09$ (Fault at bus 28 to 29 applied at $t = 5.0$ sec)

CHAPTER 7: CONCLUSION

In this paper, we discussed a Machine Learning (AI) based approach to the on-line dynamic state estimation of angle and frequency of synchronous generators using rapid measurements available from PMU. This algorithm utilizes a universal function approximation capabilities of the neural network to model nonlinear relation between measurement and the state. Also, the proposed algorithm learns in real-time without any prior training on neural network and also, is computationally around 500% efficient than EKF algorithm. Considering this approach as a starting point of feasible use of ANN in power system it serves as basic framework to lay a sophisticated work in the future.

LIST OF REFERENCES

- [1] A Chakraborty and P P Khargonekar. Introduction to wide-area control of power systems. *American Control Conference (ACC), 2013*, pages 6758–6770, 2013.
- [2] Min Chee Choy, Dipti Srinivasan, and Ruey Long Cheu. Neural networks for continuous online learning and control. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, 17(6):1511–1531, 2006.
- [3] Alberto Del Angel, Pierre Geurts, Damien Ernst, Mevludin Glavic, and Louis Wehenkel. Estimation of rotor angles of synchronous machines using artificial neural networks and local PMU-based quantities. *Neurocomputing*, 70(16-18):2668–2678, 2007.
- [4] Esmaeil Ghahremani and Innocent Kamwa. Dynamic state estimation in power system by applying the extended kalman filter with unknown inputs to phasor measurements. *IEEE Transactions on Power Systems*, 26(4):2556–2566, 2011.
- [5] Esmaeil Ghahremani and Innocent Kamwa. Online State Estimation of a Synchronous Generator Using Unscented Kalman Filter From Phasor. *IEEE Transactions on Energy Conversion*, 26(4):1099–1108, 2011.
- [6] Esmaeil Ghahremani and Innocent Kamwa. Local and wide-area PMU-based decentralized dynamic state estimation in multi-machine power systems. *IEEE Transactions on Power Systems*, 31(1):547–562, 2016.
- [7] Chakphed Madtharad, Suttichai Premrudeepreechacharn, and Neville R. Watson. Power system state estimation using singular value decomposition. *Electric Power Systems Research*, 67(2):99–107, 2003.

- [8] A G Phadke. A New Measurement Technique For Tracking Voltage Phasors ,Power System Frequency,And Rate of Change of Frequency. *IEEE Transactions on Power Apparatus and Systems*, PAS-102(5):14, 1983.
- [9] A G Phadke. Synchronized phasor measurements in power systems. *IEEE Computer Applications in Power*, 6(2):10–15, 1993.
- [10] A. G. Phadke, J. S. Thorp, and K. J. Karimi. State estimation with phasor measurements. *IEEE Transactions on Power Systems*, 1(1):233–238, 1986.
- [11] S. Pillutla, A. Keyhani, and I. Kamwa. Neural network observers for on-line tracking of synchronous generator parameters. *IEEE Transactions on Energy Conversion*, 14(1):23–30, 1999.
- [12] Junjian Qi, Kai Sun, Senior Member, Jianhui Wang, and Senior Member. Dynamic State Estimation for Multi-Machine Power System by Unscented Kalman Filter with Enhanced Numerical Stability. *IEEE Transactions on Smart Grid*, 3053(c):1–13, 2016.
- [13] Graham Rogers. Power System Toolbox Version 3.0 ©. *Rivers*, (905):1–123, 2008.
- [14] Kr Shih and S Huang. Application of a robust algorithm for dynamic state estimation of a power system. *Power Systems, IEEE Transactions on*, 17(1):141–147, 2002.
- [15] Dan Simon. Kalman Filtering. *Embedded System Programming*, (June):72–79, 2001.
- [16] Abhinav Kumar Singh and Bikash C Pal. Decentralized Dynamic State Estimation in Power Systems Using Unscented Transformation. *794 IEEE TRANSACTIONS ON POWER SYSTEMS*, 29(2):794–804, 2014.
- [17] Energy Society. *IEEE Standard for Synchrophasor Measurements for Power Systems IEEE Power & Energy Society*, volume 2011. 2012.

- [18] Jeffrey K Uhlmann, Simon J Julier, and Michael Csorba. covariance intersection. 2017.
- [19] E A Wan and R van der Merwe. The Unscented Kalman Filter for Nonlinear Estimation. *IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pages 153–158, 2000.
- [20] Shaobu Wang, Wenzhong Gao, and A P Sakis Meliopoulos. An Alternative Method for Power System Dynamic State Estimation Based on Unscented Transform. *IEEE Transactions on Power Systems*, 27(2):942–950, 2012.
- [21] Jie Yan, Chen Ching Liu, and Umesh Vaidya. PMU-based monitoring of rotor angle dynamics. *IEEE Transactions on Power Systems*, 26(4):2125–2133, 2011.
- [22] Jinghe Zhang, Greg Welch, and Gary Bishop. Observability and estimation uncertainty analysis for PMU placement alternatives. *North American Power Symposium 2010, NAPS 2010*, 2010.
- [23] Jinghe Zhang, Greg Welch, and Gary Bishop. A Two-Stage Kalman Filter Approach for Robust and Real-Time Power System State Estimation. *IEEE TRANSACTIONS ON SUSTAINABLE ENERGY*, 5(2):1–8, 2014.
- [24] Kevin Schneider Zhenyu Huang. Feasibility Studies of Applying Kalman Filter Techniques to Power System Dynamic State Estimation. *The 8th International Power Engineering Conference (IPEC 2007)*, pages 376–382, 2007.
- [25] N Zhou, D Meng, Z Huang, and G Welch. Dynamic State Estimation of a Synchronous Machine Using PMU Data: A Comparative Study. *IEEE Transactions on Smart Grid*, 6(1):450–460, 2015.