UNIVERSITÉ DE LIÈGE

Faculté des Sciences Appliquées

Dynamic State Estimation of Synchronous Generators using Phasor Measurement Units.

par

Alberto DEL ÁNGEL HERNÁNDEZ
Master of Science in Electrical Engineering,
Instituto Politécnico Nacional, México

Thèse présentée en vue de l'obtention
du grade de Docteur en Sciences Appliquées.

Année Académique 2006-2007

Acknowledgments

First of all, I would like to thank my supervisor, Professor Louis Wehenkel, for his time, patience, dedication, support and guidance during all my PhD studies. Also I thank for the confidence that he had put in me during these last years of studies

Thanks to accept to be my promoter of thesis and thanks a lot for his experience, and everything what I have learned in the field of research.

Thank you very much Dr Wehenkel.

I am deeply grateful with Professor Mania Pavella, for her kind support received when I was arriving for the first time to Liege.

I also thank to Dr. Daniel Olguin-Salinas who gave me the opportunity to come to the University of Liege to follow my PhD studies. Thanks for his help and advice during all the time that I have stayed in Liege.

Thank to Dr. Mevludin Glavic for his help and support during my PhD studies.

I would like to express my sincere gratitude to Dr. Pierre Geurts, Dr. Damien Ernst and Dr. Raphael Maree for their discussions, help and research collaborations.

I would like to express my gratitude to Philip Mack for the permission to use PEPITo software, and all the staff of PEPITe for provides me their help kindly.

I thank to the Comision de Operación y Fomento de Actividades Academicas (COFAA), the Programa SUPERA of ANUIES (Mexico) and the Instituto Politecnico Nacional of Mexico for the financial support they granted me during the first years of my PhD studies.

I gratefully acknowledge the Comité National d'Accueil, Asbl. (Belgium) for the financial support that I had receive during this last year.

I would like to thank to my parents for their support and also to my family for their patience and understanding during this time far our homeland.

Summary

This main objective of this thesis is to develop and improve techniques for estimating and predicting rotor angles, speeds and accelerations in the time frame of transient (angle) stability of electric power systems. The investigated dynamic state estimation technique is based on the use of voltage and current phasors that can be acquired in real-time using a PMU (phasor measurement unit) located at the EHV bus of the power system. The research is based on simulation data because techniques for the direct measurement of rotor angle were not available.

Abbreviations and nomenclature

Main abbreviations used:

ac	Alternating current
AL	Automatic learning
ANN	Artificial Neural Network
COI	Centre of inertia
dc	Direct current
EHV	Extra High Voltage
EPS	Electrical Power System
FFT	Fast Fourier Transform*
MIS	Mexican Interconnected System
MLP	Multilayer Perceptron
OMIB	One Machine Infinite Bus system
PMU	Phasor Measurement Unit
PSCAD	Power System Computer Aided Design
DT	Decision trees
RT	Regression Trees
DFT	Discrete Fourier Transform
GPS	Global Position Satellite

Notations

δ	Rotor angle
ϖ	Rotor speed
$\hat{\delta}$	Rotor angle predicted
$\hat{\omega}$	Rotor speed predicted
μ	Mean
σ	Standard deviation
r	Linear correlation coefficient

Contents

List of figures

List of Tables

CHAPTER 1 MOTIVATION

1.1 CONTEXT AND MOTIVATIONS

1.1.1 Transient Stability Assessment (TSA).

In conventional practice, security assessment is carried out in preventive mode by analytically modeling the network and solving a load-flow equation or a time-domain simulation repeatedly for all of the prescribed disturbances, one contingency at a time. This normal practice is not entirely satisfactory, because the computations are lengthy and are necessarily limited to a small subset of the huge set of possible contingencies. It is therefore desirable to develop emergency control techniques able to process rapidly enough real-time measurements in order to detect instabilities and compute appropriate control actions.

Power system transient (angle) stability assessment consists of evaluating the ability of the system to face various disturbances without loss of synchronism and of proposing appropriate remedial actions whenever deemed necessary [PERV00]. To monitor and control in real-time transient stability, the rotor angle and speed of the synchronous generators are the most important reference quantities. Indeed, if these quantities can be estimated with sufficient accuracy and speed, they can be exploited in order to monitor in real-time loss of synchronism and devise automatic closed loop stabilization schemes [WREP05].

There are differences between the real-time stability prediction problem and conventional transient stability assessment. In conventional transient stability assessment, the critical clearing time (CCT) is to be found via repeated simulations of the power system dynamics with different values of the fault clearing time. On the other hand, in the real-time stability prediction problem, the CCT is not of interest. Instead, one wishes to monitor the progress of the system dynamics in real-time thanks to synchronized phasor measurements acquired subsequently to the actual fault inception and clearing [RLLM^{+}95].

PMUs (Phasor Measurement Units) are power system devices that provide an accurate

measurement of real-time phasors of bus voltages and line currents. A number of PMUs are already installed in several utilities around the world for various applications such as monitoring, control, protection, and state estimation. The capabilities of a PMU are illustrated in Fig. 1.1. The measurements set is composed of the bus voltage magnitude V_B and angle θ_B, as well as the line and injection currents magnitude and angles $I_1, I_2, I_3, I_L, \theta_1, \theta_2, \theta_3,$ and θ_L.

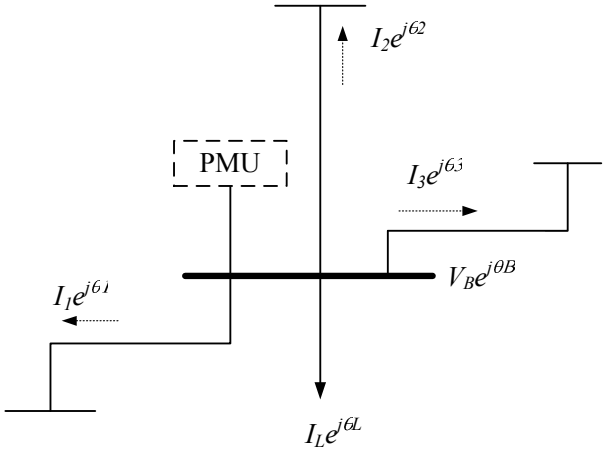


Figure 1.1 Phasor measurements from a PMU. Adopted from [LT95]

Assuming that a PMU is installed in the substation of a power plant, one can use its measurements in order to estimate electromechanical state-variables of each generator of this plant, such as rotor angles and speed, and to predict the trajectory of these quantities over a certain interval of time. The simplest way to determine rotor angles from phasor measurements is to rely on the classical generator model to compute rotor angles [RLLM+95] by

$$E'\angle\delta = V_t\angle\theta_{tV} - jX'I_t\angle\theta_{tI}\,, \tag{1.1}$$

where E' is the internal electromotive force of the generator, V_t is generator terminal voltage magnitude, X' is the generator's transient reactance in the direct axis, and I_t is the generator terminal current. Having calculated rotor angles at successive time instants the rotor speed can then be approximated by

$$\omega(t) = \frac{\delta(t+1) - \delta(t)}{\Delta t}\,. \tag{1.2}$$

These formulas assume that the generator terminal voltage, i.e. MV (medium voltage), and its terminal currents phasors are available. In more general situations, phasor measurements are not taken directly from generator buses. In this case, for algebraic relation of measured voltages V_m and the generator (internal) voltages and currents, the reduced admittance matrix Y_{BUS} can be solved for the generator internal voltages,

$$\begin{bmatrix} I_g \\ \hline 0 \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} \begin{bmatrix} V_g \\ \hline V_m \end{bmatrix} \tag{1.3}$$

where V_m are the exact values of the measured voltages, V_g are the generator internal voltages and I_g are the generator internal currents.

Taking into account measurement errors, a simple manipulation gives,

$$Y_{21}V_g + Y_{22}(V_m + \varepsilon) = 0 \quad \Rightarrow \quad V_m = -Y_{22}^{-1}Y_{21}V_g + \varepsilon \tag{1.4}$$

This equation can then be solved for the generator voltages by a least squares approach. It is important to observe that the use of relations (1.1, 1.2, 1.3 and 1.4) requires a priori knowledge of system parameters or reduced admittance matrix whose entries may experience changes due to factors influencing it and reliable system parameter identification may be required.

Another problem that may arise and obstacle phasor measurements from providing a real picture of rotor angles is the lack of direct measurements of the plant auxiliaries. To make better use of PMUs it is necessary to cope with the aspects identified above.

Since PMUs are mainly placed at EHV network buses, we suppose in this thesis that the power plants whose dynamic state one wishes to monitor are equipped with a PMU located at the EHV side of the step-up transformers of the generators of this power plant. The measurements provided by such a PMU can be exploited in order to estimate and predict the center of angle (COA) and its derivative, rather than individual generator angles and speeds. A good and fast estimate and prediction of the centre of angle dynamics obtained from local information only should be sufficient to detect impending loss of synchronism of a power plant before it is too late to react.

Synchronized phasor measurements have been recognized to offer a unique opportunity for improving the response of protection and control systems to an evolving power swing.

At the foundation of all possible improvements is the prospect of predicting in real-time the outcome of an evolving electromechanical transient oscillation [Pha93]. These oscillations are in fact related in a non-linear fashion to the electrical variables that can be measured by PMUs and the main idea developed in this thesis is to employ a machine learning approach to map the patterns of inputs (electrical variables measured by PMU) to outputs (the mechanical rotor angle and speed of the generators of a power plant).

The proposed approach using Artificial Neural networks to estimate and predict rotor angles and speeds based on real-time phasor measurements is motivated by the growing need for the real-time monitoring and control of power systems transient dynamics and by the fact that PMU devices become more and more widely available on real systems (WECC, USA [BPA99], Spain and Italy [DW02], Nordic countries [ELO00], Brazil [DEAS^{+}04], Hydro-Québec, Canada [KG02,KGH01].

To realize the mapping of the variables measured by PMU to the rotor quantities we use the multilayer feed-forward Artificial Neural Network (ANN). The reason for choosing this kind of method is related to its advantages over conventional computing methods. Those advantages are robustness to input and system noise, learning from examples, ability to handle situations of incomplete information and corrupted data, and performing in real time.

1.2 OBJECTIVE OF THE THESIS

The main objective of this doctoral thesis is to develop and improve techniques for estimating and predicting rotor angles and speeds, in the time-frame of transient (angle) stability of electric power systems. For this purpose we use Artificial Neural Networks trained by supervised learning algorithms; an ANN is characterized by its architecture, training or learning algorithms and activation functions. The architecture describes the connections between the neurons. It consists of an input layer, an output layer and generally, one or more hidden layers. In supervised ANNs, the learning algorithm makes use of both input-output data. The weights are updated for every set of input/output data. The Multilayer Perceptron falls into this category and we have used it in this thesis.

The investigated dynamic state estimation technique is based on the use of voltage and current phasors that can be acquired in real-time using a PMU located at the Extra-High

Voltage bus of a power plant. The research is based on simulated data because direct measurements of rotor angles and PMU quantities were not available.

Based on simulations with a very detailed power system model, our objective is hence to show the feasibility of using multi-layer perceptrons to estimate and predict rotor angles and speeds sufficiently accurately and sufficiently quickly for closed loop emergency control.

1.3 ORGANIZATION OF THE THESIS

The thesis is organized as follows:

Chapter 2 covers a brief description of the background of Electric Power Systems (EPS) and Automatic Learning (AL). We provide some basic definitions, talk about different algorithms of AL, and elaborate about why we decided use ANNs in this thesis.

Chapter 3 describes the problem formulation that we want to tackle in real-time and the basic principle of the proposed approach based on off-line training of multilayer perceptrons.

Chapter 4 discusses a first set of results obtained by training neural networks on simulated datasets from a single-phase electromechanical model of a One-Machine-Infinite-Bus (OMIB) obtained with MATLAB SIMULINK.

Chapter 5 is devoted to the main results of the thesis, based on training neural networks on a simulated dataset for a three-phase EMTP type multi-machine model of a part of the Mexican Interconnected System (MIS) obtained with the PSCAD software.

Chapter 6 gives the main conclusions and discusses further work directions.

Appendix A provides a short explication of the Synchronous Machine, Exciter and Governor modelling used in our work. Appendix B gives some basic definitions of PMU and describes also the algorithm used in PSCAD environment in order to model the PMUs.

CHAPTER 2 BACKGROUND

2.1 POWER SYSTEMS DYNAMICS

2.1.1 Electrical power systems. A historical survey [MEH00].

Power Systems have evolved from the original central generating station concept to a modern highly interconnected system with improved technologies affecting each part of the system separately [SP98].

In 1881, the first central station electric power generation was opened in New York. This station had a capacity of four 250-hp boilers supplying steam to six engine-dynamo sets. This central station used a 110-dc underground distribution network with copper conductors insulated with a jute wrapping. The invention of the transformer, then known as the “inductorium”, made ac systems possible. The first practical ac distribution system in the United States was installed in Massachusetts in 1866. Early ac distribution utilized 1000 V overhead lines. By 1895, Philadelphia had about twenty electric companies with distribution systems operating at 100 V and 500 V two-wire dc and 220 V three-wire dc, single-phase, two-phase and three-phase ac; with frequencies of 60, 66, 125, and 133 cycles per second; and feeders at 1000-1200 V and 2000-2400 V. Underground distribution of voltage up to 5 kV was made possible by the development of rubber-base insulated cables and paper insulate, lead-covered in the early 1900s. Common distribution voltages in today’s systems are in 5, 15, 25, 35, and 69 kV voltage classes.

The growth in size of power plants and in the higher voltage equipment was accompanied by interconnections of the generation facilities. These interconnections decreased the probability of service interruptions, made the utilization of the most economical units possible, and decreased the total reserve capacity required to meet forced equipment outages [MEH00].

Extra high voltage (EHV) has become the dominant factor in the transmission of electric power over long distances. By 1896, an 11 kV three-phase line was transmitting 10 MW from Niagara Falls to Buffalo over a distance of 20 miles. Today, transmission voltages

of 230 kV, 287 kV, 345 kV, 500 kV, 735 kV, and 765 kV are commonplace. One prototype is the 1200 kV transmission tower.

Protecting isolated systems has been a relatively simple task, which is carried out using over current directional relays with selectivity being obtained by time grading. High-speed relays have been developed to meet the increased short-circuit currents due to the large size units and the complex interconnections.

For reliable service an electric power system must remain intact and be capable of withstanding a wide variety of disturbances. It is essential that the system be operated so that more probable contingencies can be sustained without loss of load and so that the most adverse possible contingencies do not result in widespread and cascading power interruption [MEH00].

An electric power system is a set of interconnected devices that allow the transportation of electrical energy from power generation stations to the centers of consumption. The following definition can be useful in order to understand this process.

While no two electric power systems are alike, all share some common fundamental characteristics including [MEH00]:

- *Electric power is generated using synchronous machines that are driven by turbines (steam, diesel, hydraulic, or internal combustion).*
- *Generated power is transmitted from the generating sites over long distances to load centers that are spread over wide areas.*
- *Three phase ac systems comprise the main means of generation, transmission and distribution of electric power.*
- *Voltage and frequency levels are required to remain within tight tolerance levels to assure a high quality product.*

The following definitions mentioned below have been taken from [PM94] and [Kun04].

Power System: A "*power system*" is a conglomeration of generating units, transformers (of all kinds), transmission lines, loads, capacitors (shunt / series), reactors (shunt /series), static VAR compensators, conversion equipment (for HVDC integration) with associated

auxiliaries, and switch-gears to connect various components. Of course, this is not to exclude the remainder of power plants, protection subsystems and control subsystems.

Operating state: An "*operating condition*" or" *operating point*" or "*operating state*" of a power system is a set of physical quantities or physical variables that can be measured or calculated and which can meaningfully describe the state (status) of the system completely i.e. characterize the system.

Steady-state operating condition: of power system is an operating condition in which all the physical quantities that characterize the system can be considered to be constant for the purpose of analysis.

Synchronous operation of a machine: A machine is in synchronous operation with a network or another machine to which it is connected if its average electrical speed (product of its rotor angular velocity and the number of pole-pairs) is equal to the angular frequency of the ac network voltage or to the electrical speed of another machine**.**

Synchronous operation of a power system: A power system is in synchronous operation if all its connected synchronous machines are in synchronous operation with the ac network and with each other.

2.1.2 Power system dynamics.

The major components of a power system can be represented in a block-diagram format, as shown in Figure 2.1.

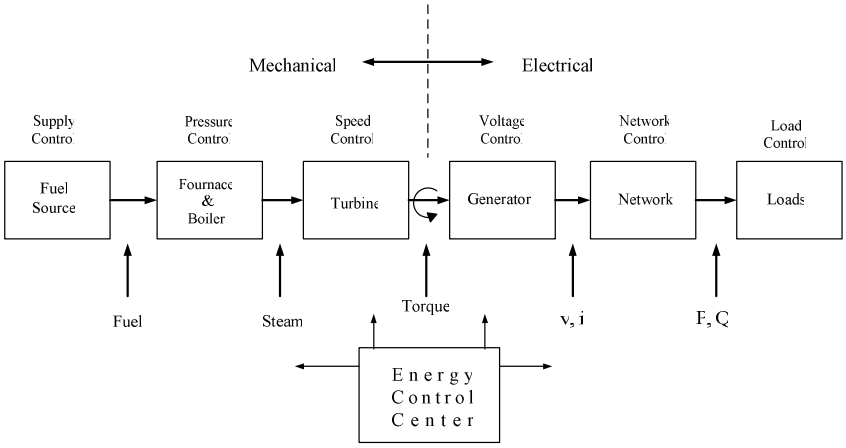


Figure 2.1 System dynamic structure. Adapted from [SP98]

While this block diagram representation does not show all of the complex dynamic interactions between components and their controls, it serves to broadly describe the dynamic structures involved. Historically, there has been a major division into mechanical and electrical subsystems as shown. This division is not absolute, however, since the electrical side clearly contains components with mechanical dynamics (tap-changing-under-load transformers, motor loads, etc) and the mechanical side clearly contains components with electrical dynamics (auxiliary motor drives, process controls, etc.). Furthermore, both sides are coupled through the monitoring and control functions of the energy control center [SP98].

Mathematically, dynamics of a power system can be described by a set of algebraic-differential equations,

$$\dot{x} = f(x, y), \tag{2.1}$$

$$0 = g(x, y), \tag{2.2}$$

where the first set of differential equations describes the dynamical part of the system (generators, motors, including their controls, and other devices whose dynamics are modeled), while the second set of algebraic equations describes the static part (what is considered as the static part depends on the particular problem of interest), x is the vector of dynamical variables and y is vector of static variables. For example, in the framework of transient stability, the dynamical variables that are modeled correspond to phenomena which time-constants range between a few milliseconds to a few seconds, while the static

part concern the slower phenomena (e.g. related to boiler and water reservoir dynamics) which are considered invariant, and the faster ones (e.g. related to electromagnetic transients, electronic devices) which are modeled by their equilibrium equations.
Imposing in the equations (2.1) $\dot{x} = 0$ yields the overall equilibrium equations, which allow one to compute the steady state conditions of the system in its pre-fault state or in its post-fault state [PERV00].

2.1.3 Power system security.

Security is a very important property of electric power systems. Security is freedom from risk or danger. Power Systems, however, can never be secure in this absolute sense. Accordingly, in a power system context, security can only be a qualified absence of risk, specifically of risk of disruption of continued system operation. From a control perspective, the objective of power system operation is to keep the electrical flows and bus voltage magnitudes and angles within acceptable limits, despite changes in load or available resources and despite external perturbations. From this perspective, security may be defined as the probability of the system's operating point to remain in a viable state space, given the probabilities of the changes in the system (contingencies) and its environment (weather, customer demands, etc) [BBBB92].

The determination of security levels, for given operating conditions, traditionally has been done using deterministic criteria. Under deterministic criteria, an operating condition is identified as secure if it can withstand the effects of each and every contingency in a pre-specified contingency set. Withstanding the effects means that the given contingencies will not violate branch loading or nodal voltage criteria in steady state conditions or make the system dynamically unstable [IEEE04].
The task of assessing the level of security for a given operating condition or topology configuration, often leads to the definition of a *security margin* using some selected variables or parameters. The choice of these variables or parameters depends on the type of phenomena limiting the system. Given the high complexity of power systems and the large range of time constants of its dynamics, the study of power system security has led

to the decomposition of the notion of security into various sub-problems along different criteria. Below, we introduce the main notions and definitions that have been introduced in this field.

Power system stability. Stability is the ability of an electrical power system, for a given initial operating condition, to regain a state of operating equilibrium after being subjected to a physical disturbance, with most system variables bounded so that practically the entire system remains intact [IEEE/CIGRE04].

The terminology about power system stability proposed in [IEEE/CIGRE04] is based on the following considerations:

- The physical nature of the resulting mode of instability as indicated by the main system variable in which instability can be observed.
- The size of the disturbance considered which influences the method of calculation and prediction of stability.
- The devices, process and the time span that must be taken into consideration in order to assess stability.

There are several main divisions in the study of power system dynamics and stability [SP98]. De Mello classified dynamic processes into three categories:

- Electrical machine and system dynamics.
- System governors and generating control.
- Prime-mover energy supply dynamics and control.

Concordia and Schultz classify dynamics studies according to four concepts [SP98, CS75]:

- The time of the system condition: past, present and future.
- The time range of the study: microsecond through hourly response.
- The nature of the system under study: new station, new line, etc.

- The technical scope of the study: fault analysis, load shedding, sub synchronous resonance, etc.

All these classifications share a common thread: they emphasize that the system is not in steady state and that many models for various components must be used in varying degrees of detail to allow efficient and practical analysis [SP98].

Disturbance. A disturbance in a power system is a sudden change or sequence of changes in one or more of the parameters of the system, or in one or more of the physical quantities [PM94].

Small disturbance. A small disturbance is a disturbance for which the set of equations that describe the power system may be linearized for purpose of analysis [PM94].

Large disturbance. A large disturbance is a disturbance for which the equations that describe the power system cannot be linearized for the purpose of analysis [PM94].

Pre-fault system. A power system immediately preceding the initiation of a large disturbance is termed a “pre-disturbance (pre-fault) system”. The system is usually considered to be in steady state in this phase [PM94].

Fault-on system. In the during disturbance (or during fault or fault-on) system the power system is under the continuous influence of a disturbance (or a sequence of disturbances); this phase lasts for the entire duration of the disturbance. This is the initial stage of the transient period [PM94].

Post-fault system: A power system immediately following the complete isolation of a large disturbance is termed a “post-disturbance (post-fault) system”. During the post-fault phase the transient period continues and the system may or may not eventually reach a steady-state. The post-disturbance phase decides whether the system is stable or not [PM94].

Figure 2.2 gives the overall picture of the power system stability problem, identifying its categories and subcategories.

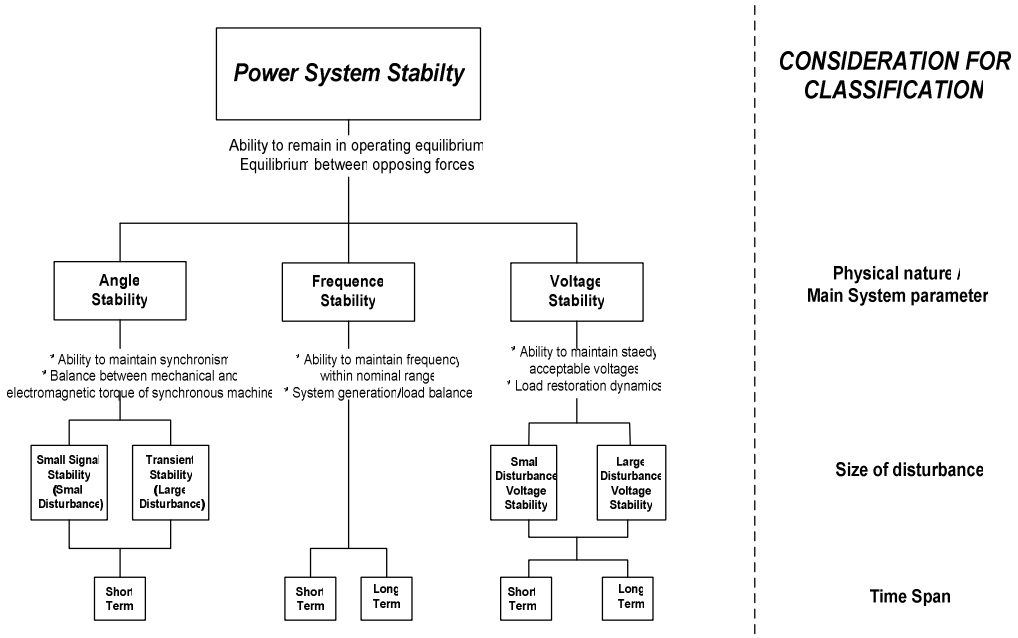


Figure 2.2 Classification of Transient Stability. Adapted from [Kun94].

Classification of power security assessment and control can be defined as follows [RV03]:

Static security assessment (SSA), are methodologies that verify bus voltage and line power flow limits for the post-contingency steady state operating condition, considering that the transition between the pre-contingency and the post-contingency steady state operating states has taken place without suffering any instability phenomena in any part of the system. Static security assessment essentially verifies the existence of a post-fault steady state that satisfies all constraints deemed important for this state to survive for a long enough period of time.

Dynamic Security Assessment (DSA), are methodologies for evaluating the stability and quality of the transient processes between the pre-contingency and post-contingency steady states. DSA aims at ensuring that the system will be stable after the contingency occurrence and that the transient caused by such contingency will be well damped, of small amplitude and with little impact on the quality of service. Dynamic security assessment essentially verifies that the system will reach and remain in a neighborhood of its post-fault steady state conditions.

The following are description of the corresponding forms of instability phenomena:

Voltage instability is the inability of a power system to maintain steady voltages at all buses in the system after being subjected to a disturbance from a given initial operating condition. It depends on the ability to maintain/restore equilibrium, between load demand and load supply from the power system. Voltage instabilities generally occur in the form of a progressive fall or rise of voltages of some buses [Kun04].

Frequency instability refers to the inability of a power system to maintain steady frequency following a severe system upset resulting in a significant imbalance between generation and load. It depends on the ability to maintain/restore equilibrium between system generation and load, with minimum unintentional loss of load. Frequency instability generally occurs in the form of rapid frequency drops leading to tripping of generating units and/or loads [Kun04].

Rotor angle instability is the inability of synchronous machines of an interconnected power system to remain in synchronism after being subjected to a disturbance. It depends on the ability to maintain/restore equilibrium between electromagnetic torque and mechanical torque of each synchronous machine in the system. Transient instability may occurs in the form of increasing angular swings of some generators leading to their loss of synchronism with

other generators [Kun04]. One often distinguishes between plant mode instability, where a single power plant looses his stability and inter-area mode instability where all the plants of large area loose their synchronism with respect to the rest of the interconnection.

For convenience in analysis and for gaining useful insight into the nature of stability problems, it is useful to characterize rotor angle instability in terms of the following two subcategories [Kun04]:

Small-disturbance (or small-signal) rotor angle instability is concerned with the inability of the power system to maintain synchronism under small disturbances. In today's power systems, small-disturbance rotor angle stability problem is usually associated with insufficient damping of oscillations. It is generally characterized by negatively damped power swings among remote generators of the interconnection, typically leading to the tripping of interconnection lines and/or generators.

Large-disturbance rotor angle instability or *Transient Stability* is concerned with the inability of the power system to maintain synchronism when subjected to a severe disturbance, such as a short circuit on a transmission line. The resulting system response involves large excursions of generator rotor angles and speeds, followed by generator tripping due to over or under speed protections.

And finally, perhaps the most import classification of dynamic phenomena is their natural time range of response. A typical classification is shown in figure 2.2. This time-range classification is important because of its impact on component modeling and on the response speed of control and protective devices needed to counter the corresponding instabilities. It should be intuitive that is not necessary to solve the complex transmission line wave equations to investigate the impact of a change in boiler control set points. This brings to mind the statement that "the system is not in steady state". Depending on the

nature of the dynamic disturbance, portions of the power system can be considered in "quasi-steady state" [SP98].

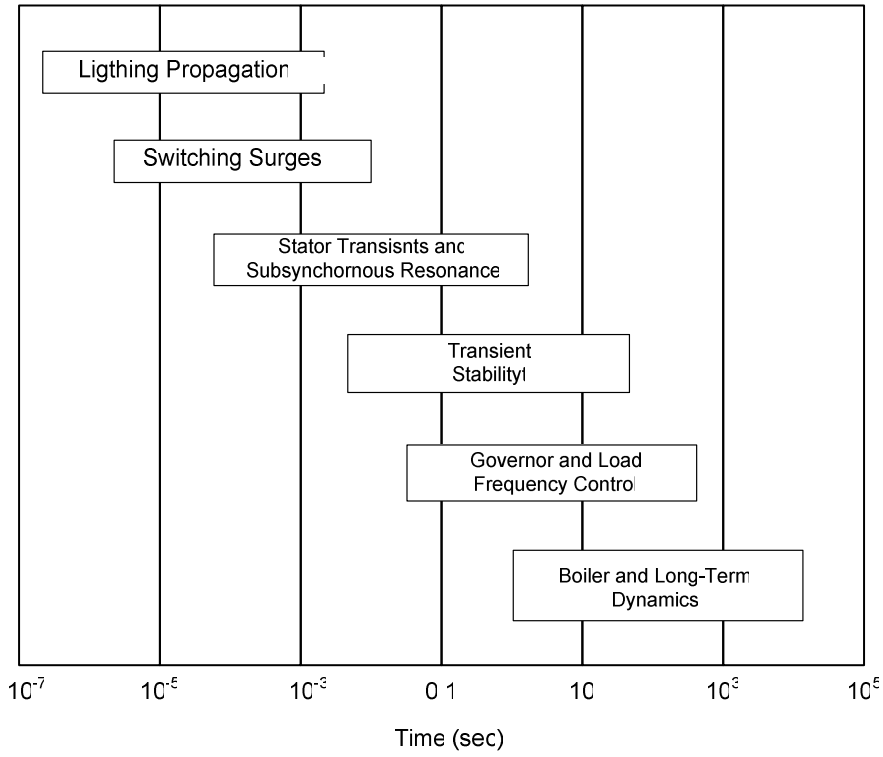


Figure 2.3 Time Ranges of dynamic phenomena. Adapted from [SP98]

2.1.4 The swing equations

This thesis is devoted to the estimation of rotor angles and speeds in the framework of transient stability. We will therefore consider the behavior of the system immediately following a disturbance such as a short circuit on a transmission line, the opening of a line or the switching on a major load to name just a few.

Since a synchronous machine is a rotating body, the laws of mechanics applying to rotating bodies apply to it [Kim64]. The equations of central importance in power system transient stability analysis are the rotational inertia equations describing the effect of unbalance between the electromagnetic torque and the mechanical torque of the individual machines [Kun94]. A brief description of the establishment of the swing equation is expressed below, along the lines given in [Kun94].

When there is an unbalance between the torques acting on the rotor, the net torque causing acceleration or deceleration is

$$T_a = T_m - T_e \tag{2.3}$$

Where T_a denotes the accelerating torque, T_m is the mechanical torque and T_e electromagnetic torque. All units are N.m. and in the above equations, Tm and T_e are positive for a generator and negative for a motor.

The combined inertia of the generator and turbine is accelerated by the unbalance in the applied torques. Hence, the equation of motion is:

$$J\frac{d\omega_m}{dt} = Ta = T_m - T_e, \tag{2.4}$$

where J is the combined moment of inertia of generator and turbine; kg.m^2, ω_m is the angular velocity of the rotor, mech. rad/sec and t time in seconds.

Defining the inertia constant H as the kinetic energy in watts-seconds at rated speed divided by the VA_{base} and denoting by ω_{0m} the rated angular velocity in mechanical radians per second, the inertia constant is obtained by

$$H = \frac{1}{2}\frac{J\omega^2{}_{0m}}{VA_{base}}. \tag{2.5}$$

The moment of inertia J in terms of H is obtained by

$$J = \frac{2HJ}{\omega^2{}_{0m}}VA_{base}. \tag{2.6}$$

Substituting the above equation in eq. (2.4) gives

$$\frac{2H}{\omega^2{}_{0m}}VA_{base}\frac{d\omega_m}{dt} = T_m - T_e, \tag{2.7}$$

and rearranging yields

$$2H\frac{d}{dt}\left(\frac{\omega_m}{\omega_{0m}}\right)=\frac{T_m-T_e}{VA_{base}/\omega_{0m}}. \tag{2.8}$$

The equation of motion in per unit form is thus

$$2H\frac{d\overline{\omega}_r}{dt}=\overline{T}_m-\overline{T}_e, \tag{2.9}$$

where

$$\overline{\omega}_r=\frac{\omega_m}{\omega_{0m}}=\frac{\omega_r/P_f}{\omega_0/P_f}=\frac{\omega_r}{\omega_0}\ , \tag{2.10}$$

where ω_r is the electrical angular velocity of the rotor in electrical rad/sec, ω_0 is its rated value, and P_f is number of field poles.

If δ is the angular position of the rotor in electrical radians with respect to a synchronously rotating reference and δ_0 is its value at $t=0$,

$$\delta(t)=\delta_0+\int_0^t\omega_r(\tau)d\tau-\omega_0 t\quad . \tag{2.11}$$

In other words we have

$$\frac{d\delta}{dt}=\omega_r-\omega_0=\Delta\omega_r \tag{2.12}$$

and

$$\begin{aligned}\frac{d^2\delta}{dt^2}&=\frac{d\omega_r}{dt}=\frac{d(\Delta\omega_r)}{dt}\\&=\omega_0\frac{d\overline{\omega}_r}{dt}=\omega_0\frac{d(\Delta\overline{\omega}_r)}{dt}.\end{aligned} \tag{2.13}$$

Substituting for $\frac{d\omega_r}{dt}$ given by the above equation in equation (2.9), we get

$$\frac{2H}{\omega_0}\frac{d^2\delta}{dt^2} = \overline{T}_m - \overline{T}_e \quad . \tag{2.14}$$

It is often desirable to explicitly isolate in these equations a component of damping torque which has its origin in a linear dependence of the mechanical and/or electromagnetic torques on the speed deviation. This leads to a slightly different swing equation formulated as follows:

$$\frac{2H}{\omega_0}\frac{d^2\delta}{dt^2} = \overline{T}'_m - \overline{T}'_e - K_D\Delta\overline{\omega}_r \tag{2.15}$$

Equation (2.15) represents the equation of motion of a synchronous machine. It is commonly referred to as the swing equation because it represents swings in rotor angle δ during disturbances.

The state-space form requires the component model to be expressed as a set of first order differential equations. The swing equation (2.15), expressed as two first order differential equations, becomes

$$\frac{d\Delta\overline{\omega}_r}{dt} = \frac{1}{2H}\left(\overline{T}_m - \overline{T}_e - K_D\Delta\overline{\omega}_r\right) \tag{2.16}$$

$$\frac{d\delta}{dt} = \omega_0\Delta\overline{\omega}_r \tag{2.17}$$

In the above equations, time t is in seconds, rotor angle δ is in electrical radians, and ω_0 is equal to $2\pi f$. The block diagram form representation of equations (2.16) and (2.17) is shown in figure 2.4.

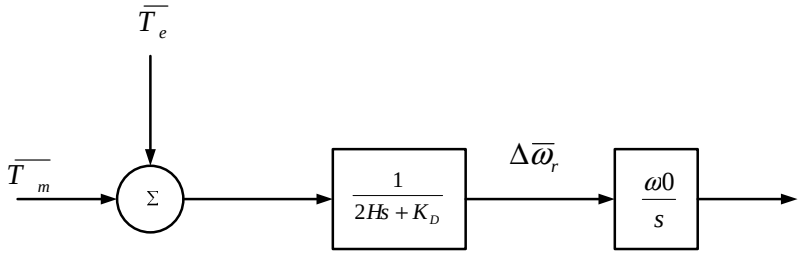


Figure 2.4 Block diagram representation of swing equation. Adapted from [Kun 94].

Characterization of the power system operating states.

In his pioneering work, DyLiacco introduced the idea that the power system may operate in the following modes: Normal, Alert, Emergency and Restorative [EPRI81,Dyl68]. More recently, Fink and Carlsen expanded this concept by identifying the constraints satisfied or violated in each mode of operation. Three sets of generic equations (one differential and two algebraic ones) govern power system operation: the differential set encodes the physical laws governing the dynamic behavior of the systems components. The two algebraic sets comprise 'equality constraints', which refer to the system's total load and total generation, and 'inequality constraints', which state that some system variables, such as currents and voltages, must not exceed maximum levels representing the limitations of physical equipment [FC78]. :

- Normal: all equality and inequality constraints are satisfied; reserve margins are adequate to withstand stresses.
- Alert: all constraints are still satisfied; reserve margins are such that some disturbance could result in a violation of some inequality constraints.
- Emergency: some inequality constraints are violated; the system is still intact and control actions could be initiated to restore the system to at least the alert state.
- In extremis: equality constraints and inequality constraints are violated; the system will no longer be intact and a portion of the load will be lost.

- Restorative: control actions are being taken to pick up the lost load and to reconnect the system.

Figure 2.5 shows the five different operating states of the power system.

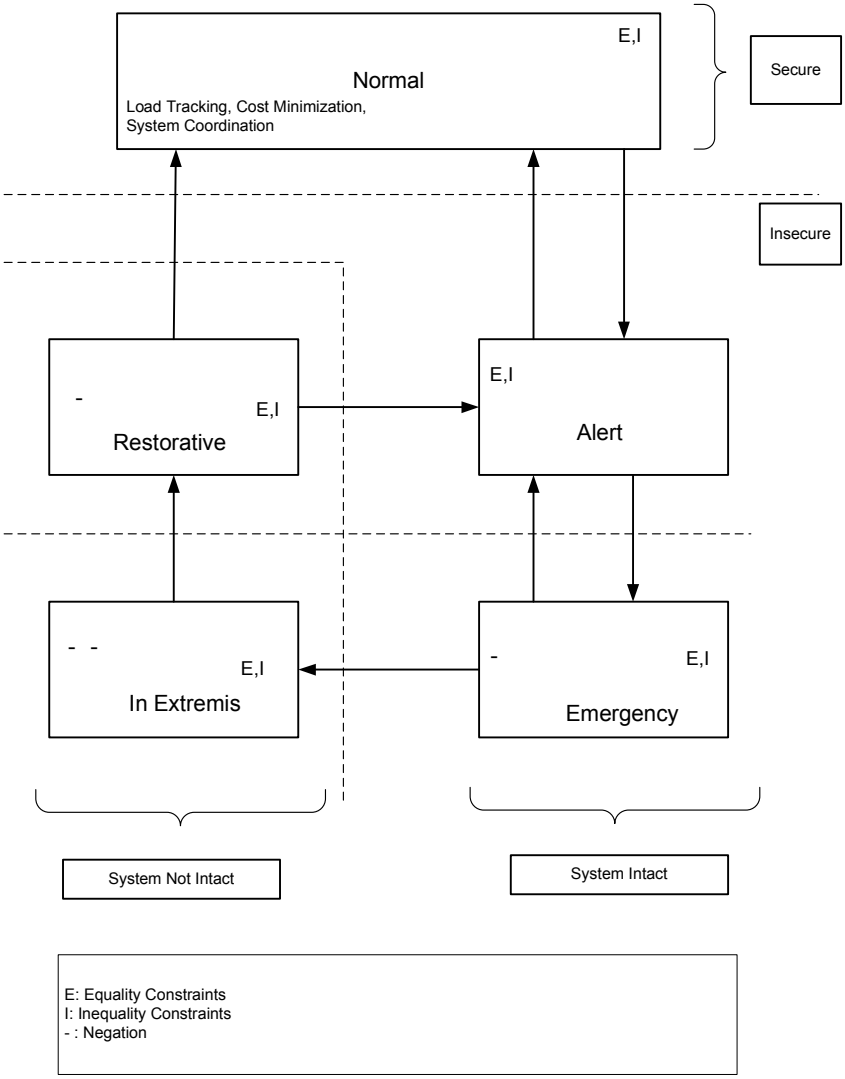


Figure 2.5 Power system operating states. Adapted from [SC02]

2.2 STATE ESTIMATION AND AUTOMATIC LEARNING

2.2.1 Automatic learning

Automatic learning (AL) is a term used to denote a highly multidisciplinary research field of methods, which aim to extract information (knowledge) from databases containing large amounts of low-level data. AL encompasses three main families of methods:

- Statistical data analysis and modeling,
- Artificial neural networks (ANN)
- Symbolic machine learning in artificial intelligence.

During the last 20 years, many researches worked in the topic, applying different techniques (statistical pattern recognition, neural networks and machine learning) to different power system problems (load forecasting, system identification and state estimation, stability assessment and control) [Weh98].

Automatic learning methods essentially aim at extracting a model of a system from the sole observation (or the simulation) of this system in some situations. By model, we mean some relationships between the variables used to describe the system in some encountered situations or to help understating its behaviour [Weh98].

2.2.2 Supervised learning

Supervised learning is the part of automatic learning that focuses on modelling input/output relationships. More precisely, the goal of supervised learning is to identify a mapping from some input variables to some output variables on the sole basis of a sample of observations of the values of these variables. The variables are often called (input or output) *attributes*, observations are called *objects* and the sample of objects is the *learning sample*. In the context of security assessment, an object would thus correspond to an operating state of a power system, or more generally to a simulated security scenario. The input attributes would be relevant parameters describing its electrical state and topology and the output could be information concerning its security, in the form of either a discrete classification (e.g. secure/insecure) or a numerical security margin [Weh98].

The general problem of supervised learning is formally stated as follows [Geu02]:

> *For any value of N and a learning sample LsN and without any a priori knowledge of the functions P(.), Y(.), or, A(.), find a function f(.) defined on A which minimizes the expected prediction error defined by:*
>
> $$Err(f) = E_{\underline{A},Y}\{L(Y, f(\underline{A}))\} = \int_U L(y(o), f(\underline{a}(o)))dP(o) \qquad (2.18)$$
>
> *where U denotes the universe of all possible objects, y(.) the output attribute (a function defined on U) and a(.) the vector of input attributes (another function defined on U). L(.,.) is a loss function which measures the discrepancy between its two arguments, P(.) is a sampling probability distribution defined over U and LsN is a sample of N observed objects for which both y and a are given as inputs to the supervised learning algorithm.*

There are two main types of supervised learning problems:

- Classification problem: the output attribute takes a finite number of discrete values.
- Regression problem: the output attribute takes a possible infinite number of real values.

In this thesis, we are focused integrally to the regression problem.

2.2.3 Main classes of supervised learning algorithms

In this section, we provide a brief overview of the main types of supervised algorithms that exist in the literature. In this dissertation we will mainly use ANNs and more specifically MLPs. MLPs will be explained with more details in a subsequent section.

2.2.3.1 Linear regression

Linear regression is one of the oldest forms of machine learning. It is a long established statistical technique that involves simply fitting a line (or a hyperplane) to some data. The easiest case for linear regression is when the examples have a single numeric input attribute and a numeric output value, i;e; if there are *N* examples, where the attributes for each example is called x_i and the label for each is y_i. We can envision each example as being a point in 2-dimensional space, with an x-coordinate of x_i and a y-coordinate of y_i. Linear regression would seek the line $f(x) = mx + b$ that minimizes the sum-of-squares-error for the training samples:

$$\sum_{i=1}^{N} \left(y_i - f(x_i)\right)^2 . \tag{2.19}$$

The quantity $\left|y_i - f(x_i)\right|$ is the distance from the value predicted by the hypothesis line to the actual value – the error of the hypothesis for the training sample *i*. Squaring this value gives grater emphasis to large errors and saves us dealing with complicated absolute values in the mathematics while minimizing eq. (2.19) ω_{rt} *m* and *b*.

2.2.3.2 Decision trees

Decision tree learning is a method for classification problems, in which the learned function is represented by a decision tree. Learned trees can also be represented as sets of if-then rules to improve human readability. Decision trees classify instances by sorting them down the tree from the root to some leaf node, which provides the classification of the instance. each node in the tree specifies test of some attribute of the instance, and each brand descending from that node corresponds to one of the possible values for this attribute. An instance is classified by starting at the root node of the tree, testing the attribute specified by this node, then moving down the tree branch corresponding to the value of the attribute in the given example. This process is then repeat for the subtree rooted at the new node. [Mit97].

Figure 2.6 illustrates a typical learned decision tree.

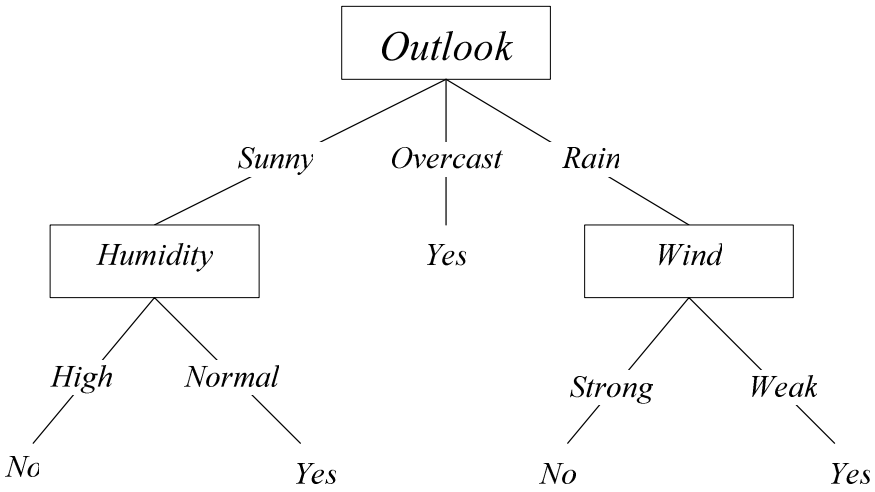


Figure 2.6 A decision tree for the concept Play Tennis. Adapted from [Mit97]

The decision tree shown in Fig. 2.6 corresponds to the expression:

$$\begin{aligned} &(Outlook = Sunny \ \wedge \ Humidity = Normal) \\ \vee \ &(Outlook = Overcast) \\ \vee \ &(Outlook = Rain \ \wedge \ Wind = Weak) \end{aligned} \tag{2.20}$$

A decision tree (DT) is obtained from a partitioning tree by attaching classes to its terminal nodes. The tree is seen as a function; associating to any object the class attached to the terminal node, which contains the object [Weh98].

The main strength of decision trees is their interpretability. By merely looking at the test nodes of a tree one can easily sort out the most salient attribute and find out how they influence the output. Another very important asset is the ability of the method to identify the most relevant attributes for each problem. The last characteristic od DT is computational efficiency: tree growing computational complexity is practically linear in the number of candidate attributes and in number or learning states, allowing one to tackle easily problems with a few hundred candidate attributes and a few thousand learning states [Weh98].

Although variety of decision tree learning methods have been developed with somewhat differing capabilities and requirements, decision tree learning is generally best suited to problems with the following characteristics [Mit97] :

- Instances are represented by attribute-value pairs: Instances are described by a fixed set of attributes and their values.
- The target function has discrete output values: the decision tree in Fig. 2.6 assigns a Boolean classification to each example (i.e. yes or no).
- Disjunctive description may be required: decision trees naturally represent disjunctive expressions.
- The training data may contain errors: Decision tree learning methods are robust to errors, both errors in classification of the training examples and errors in the attribute values that describe these examples.
- The training data may contain missing attribute values. Decision tree methods can be used even when some training examples have unknown values.

2.2.3.3 Regression trees

Regression trees may be considered as a variant of decision trees, designed to approximate real-valued functions instead of being used for classifications tasks.
The inner nodes of regression trees are marked with test as in decision trees. The difference is, that the leaves of regression trees may be marked with arbitrary real values, whereas in decision trees the leafs may only be market with elements of a finite set of discrete values. A further extension is to allow linear functions as label of leaf nodes. In this case the function at the leaf node reached for a specific example is evaluated for the instance's attributes values, to determine the value of the target attribute. This allows for global approximating by using multiple local approximations.
Regression tree induction is a well-known approach for improving along a continuous, output dimension [BFOS84].
Regression trees decompose the attribute space into a hierarchy of regions. Similary to decision trees, regression trees are built in a top-down approach: starting with the top-node and the complete learning set, an attribute a_i and a threshold value v_i are selected

to decompose the learning set into two subsets, corresponding to states for which $a_i < v_i$ and $a_i \geq v_i$ respectively.

The procedure continues splitting until either the variance has been sufficiently reduced or it is not possible to reduce it further in a statistically significant way [Weh96].

*Figure 2.7*shows a representation of a regression tree and an approximation of the numerical output.

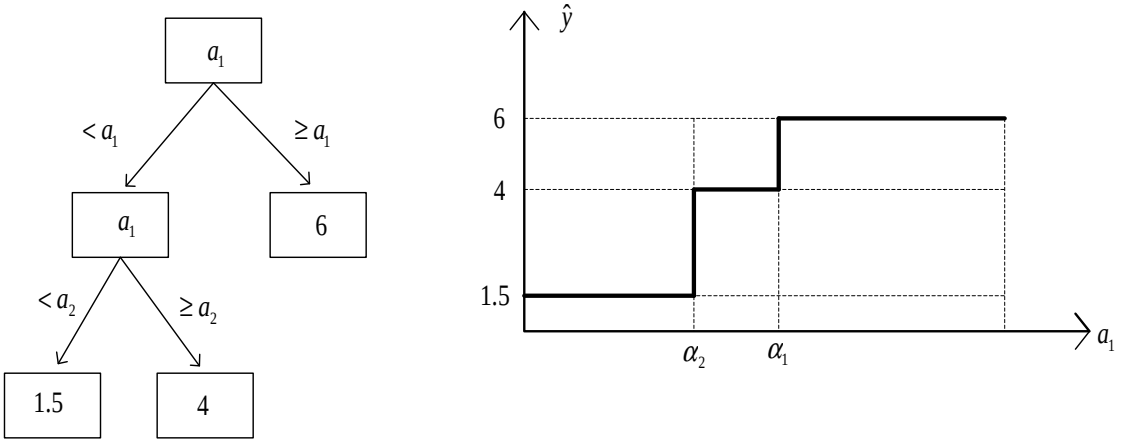


Figure 2.7 Example of a regression tree. Adapted from [Ola04]

Regression trees have been used in fields as diverse as air pollution, criminal justice, and the molecular structure of toxic substances. Its accuracy has been generally competitive with linear regression. It can be much more accurate on non linear problems but tends to be somewhat less accurate on problems with good linear structure [BFOS84].

2.2.3.4 Ensemble methods.

Ensemble methods consist in growing several models with a classical machine learning algorithm. Then, the predictions of these models are aggregated to provide a final prediction potentially better than individual ones. One of the most popular family of ensemble methods is defined by Perturb and Combine methods, that consist in perturbing the learning algorithm and/or the learning sample so as to produce different models from the same learning sample. The predictions of these models are then aggregate bya simple average or a majority vote in the case of classification [Geu03].

Bagging

The Bagging Algorithm (***B***ootstrap ***agg***regat***ing***) by Breiman [Bre96] votes classifiers generated by different bootstrap samples (replicates). A bootstrap sample is generated by uniformly sampling m instances from the training set with replacement. T bootstrap samples $B_1, B_2, ..., B_T$ are generated and a classifier C_i is built from each bootstrap sample B_i. A final classifier C^* is built from $C_1, C_2, ..., C_T$ whose output is the class predicted most often by its sub-classifiers, with ties broken arbitrarily.

Table 2.1 shown the Bagging algorithm how works:

Table 2.1 The Bagging algorithm. Adapted from [BK99]

Input: training set S, Inducer $\mathcal{I}$, integer T (number of bootstrap samples)

1. for $i = 1$ *to* T {
2. S' = bootstrap sample from S (i.i.d. sample with replacement)
3. $C_i = \mathcal{I}(S')$
4. }
5. $C^*(x) = \arg\max_{y \in Y} \sum_{i:C_i(x)=y} 1$ (the most often predicted label y)

Output : classifier C^*

For a given bootstrap sample, an instance in the training set has probability $1-(1-1/m)^m$ of being selected at least once in the m time instances are randomly selected from the training set. For large m, this is about $1-1/e = 63.2\%$, which means that each bootstrap sample contains only about 63.2 % unique instances from the training set [BK99].

Boosting

Boosting was introduce by Schapire early 90's as a method for boosting the performance of a weak learning algorithm. The Adaboost algorithm (Adaptive Boosting), introduced by Freund & Schapire [FS99], solve many of the practical difficulties of the earlier boosting algorithms. The AdaBoost algorithm is given in *Table 2*.2. the algorithm takes as input a training set $(x_1, y_1), ..., (x_m, y_m)$ where each x_i belongs to some *domain* or

instance space X, and each label y_i is in some label set Y. AdaBoost calls a given *weak* or *base learning algorithm* repeatedly in a series of rounds $t = 1,...,T$. One of the main ideas of the algorithm is to maintain a distribution or set of weights over the training set. The weight of this distribution on training sample i on round t is denoted $D_t(i)$.

The weak learner's job is to find a weak hypothesis $ht : X \to \{-1,+1\}$ appropriate for the distribution Dt. The goodness of a weak hypothesis is measured by its error

$$e_t = \Pr_{i\sim D_t}\left[h_t(x_i) \neq y_i\right] = \sum_{i:h_t(x_i)\neq y_i} D_t(i) \tag{2.21}$$

In practice, the weak learner may be an algorithm that can use the weights D_t on the training samples. Alternatively, when this is not possible, a subset of the training examples can be sampled according to D_t, and these (unweighted) resampled examples can be used to train the weak learner.

Table 2.2 the Boosting algorithm AdBoost. Adapted from [FS99]

Given: $(xi, yi),...,(xm, ym)$ where $x_i \in X$, $y_i \in Y = \{-1,+1\}$

Initialize $D_1(i) = 1/m$

For $t = 1, \ldots, T$:

- Train weak learner using distrubution D_t
- Get weak hypothesis $h_t : X \to \{-1,+1\}$ with error

$$e_t = \Pr_{i\sim D_t}\left[h_t(x_i) \neq y_i\right]$$

Choose $\alpha = \frac{1}{2}\ln\left(\frac{1-e_t}{e_t}\right)$.

Update:

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases}$$
$$= \frac{D_t(i)\exp\left(-\alpha_t y_i h_t(x_i)\right)}{Z_t}$$

where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

Output the final hypothesis:

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right).$$

ExtraTrees

Geurts [Geu03] presents a new learning algorithm based on decision tree ensembles, where the trees of the ensemble are built by selecting the tests during their induction fully at random. This extreme randomization makes the construction of the ensemble very fast even on very large datasets with high dimensionality.

The extra-trees algorithm builds an ensemble of unpruned decision or regression trees according to the classical top-down procedure. Its two main differences with other tree-based ensemble methods are that it splits nodes by choosing cut-points fully at random and that it uses the whole learning sample (rather than a bootstrap replica) to grow the trees.

Table 2.3 Extra-Trees splitting algorithm for numerical attributes [GEW06].

Split_a_node(S)
Input: the local learning subset S corresponding to the node we want to split
Output: a split $[a < ac]$ or nothing
- If **Stop_split**(S) is TRUE then return nothing.
- Otherwise select K attributes $\{a_1, \ldots, a_K\}$ among all non constant (in S) candidate attributes;
- Draw K splits $\{s_1, \ldots, s_K\}$, where $s_i =$ **Pick_a_random_split**(S, a_i), $\forall i = 1, \ldots, K$;
- Return a split s_* such that $\text{Score}(s_*, S) = \max_{i=1,\ldots,K} \text{Score}(s_i, S)$.

Pick_a_random_split(S, a)
Inputs: a subset S and an attribute a
Output: a split
- Let $a^S_{\max}$ and $a^S_{\min}$ denote the maximal and minimal value of a in S;
- Draw a random cut-point a_c uniformly in $\left[a^S_{\min}, a^S_{\min}\right]$;
- Return the split $[a < a_c]$.

Stop_split(S)
Input: a subset S
Output: a boolean
- If $|S| < n_{\min}$, then return TRUE;
- If all attributes are constant in S, then return TRUE;
- If the output is constant in S, then return TRUE;
- Otherwise, return FALSE

Table 2.3 shows the Extra-Tree procedure for numerical attributes. It has two parameters: K, the number of attributes randomly selected at each node and nmin, the minimum

smple size for splitting a node. It is used a several times with the (full) original learning sample to generate an ensemble model (denoted by M the number of trees of this ensemble). The predictions of the trees are aggregated to yield the final prediction, by majority vote in classification problems and arithmetic average in regression problems.

Fuzzy Decision Trees

A fuzzy decision tree is a method able to partition the input space into a set of rectangles and then approximate the output in each rectangle by a smooth curve, instead of a constant or a class like in the case of crisp tree-based methods. A fuzzy tree is an approximation structure to compute the degree of membership of objects to a particular class (or concept) or to compute a numerical output of objects, as a function of the attribute values of these objects. The goal is recursively split the input space into (overlapping) subregions of objects which have the same membership degree to the target class (in the case of classification problems) or the same output value (in the case of regression problems) [Ola04].

Figure 2.8 shows an example of splitting a fuzzy decision tree and its correspondent graph.

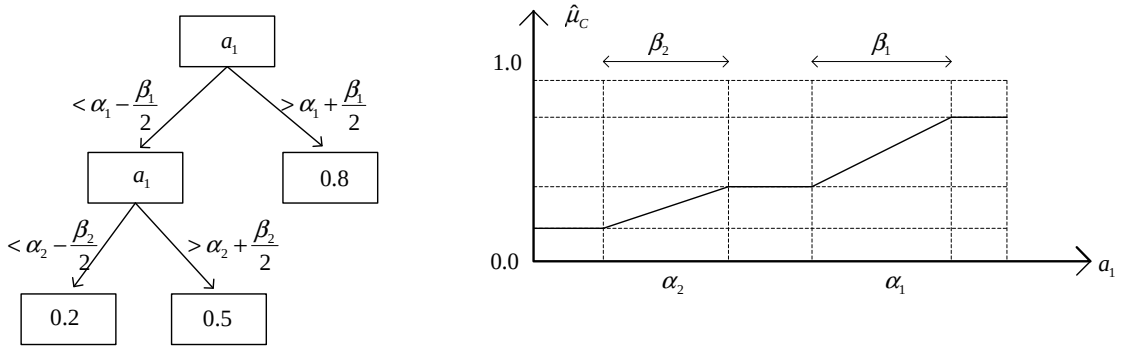


Figure 2.8 Example of a fuzzy decision tree. Adapted from [Ola04].

A fuzzy decision tree structure is determined by the graph of the tree and by the attributes attached to the its test nodes. The discretization thresholds (α) and width (β) values of all these attributes, shown in Fig. 2.8, together the labels of all the terminal nodes

represent the parameters of the tree-based model. There is a search over both structure and parameter spaces so as to learn a model from experience [Ola04].

2.2.3.5 Nearest Neighbor algorithm

The Nearest-Neighbor (1NN) method has been applied both for classification and regression. Let an arbitrary instance x be described by the attribute vector

$$\langle a_1(x), a_2(x), ...a_n(x) \rangle, \tag{2.22}$$

where $a_r(x)$ denotes the value of the *r*th attribute of instance x. Then the distance between two instances x_i and x_j is define to be $d(x_i, x_j)$, where

$$d(x_i, x_j) \equiv \sqrt{\sum_{r=1}^{n} \left(a_r(x_i) - a_r(x_j)\right)^2} \quad . \tag{2.23}$$

In Nearest-Neighbor learning the target function may be either discrete-valued or real-valued. Considering learning discrete-valued target functions of the form $f : \Re^n \to V$, where V is the finite set $\{v_1, ..., v_s\}$. The *k*-Nearest-Neighbor algorithm is shown in Table 2.4, the value $\hat{f}(x_q)$ returned by this algorithm as its estimate of $f(x_q)$ is just the most common value of f among the k training examples nearest to x_q. If we chose k=1, then the 1-Nearest-Neigbor algorithm assigns to $\hat{f}(x_q)$ the value of $f(x_q)$ where xi is the training instance nearest to x_q. For larger values of k, the algorithm assigns the most common value among the k nearest training sample [Mit97].

Table 2.4 k-Nearest-Neighbor algorithm for approximation of a discrete value [MIT97].

Training algorithm:

- For each training sample $\langle x, f(x) \rangle$, add the example to the list *training _ examples*

Classification algorithm:

- Given a query instance x_q to be classified,
  - Let $x_1 \ldots x_k$ denote the k instances from *training _ examples* that are nearest to x_q
  - Return

$$\hat{f}(x_q) \leftarrow \arg\max_{v \in V} \sum_{i=1}^{k} \delta(v, f(xi))$$

where $\delta(a,b) = 1$ if $a = b$ and where $\delta(a,b) = 0$ otherwise.

The main advantages of this algorithm are that it can in principle represent very complex input-output relations very well and is very simple to implement. On the other hand, the disadvantages can be numbered as follows:

- It does not handle many irrelevant attributes well. If we have lots of irrelevant attributes, the distance between examples is dominated by the differences in these irrelevant attributes and so becomes meaningless.
- It doesn't look much like humans learning.
- Hypothesis function is too complex to describe explicitly.
- Computational inefficiency.

The nearest neighbor algorithm and its variants are particularly well suited to collaborative filtering, where a system is to predict a given person's preference based on others people's preferences. Collaborative filtering fits into the nearest neighbor search well because attributes tend to be numeric and similar in nature, so it makes sense to give them equal weight in the distance computation [Geu02].

2.2.4 Artificial neural networks (ANN)

2.2.4.1 Biological neural networks

A neuron is a special biological cell that process information. It is composed of a cell body and two types of out-reaching tree-like branches: the axon and the dendrites as shown in Figure *2.9*. A neuron receive signals (impulses) from others neurons through its dendrites (Receivers) and transmits signals generated by its cell body along the axon (transmitter), which eventually branches into strands and sub strands. At the terminal of these strands are the synapses. A synapse is an elementary structure and functional unit between two neurons (an axon strand of one neuron and a dendrite of another). When the impulse reaches the synapse's terminal, certain chemicals called neurotransmitters are released. The neurotransmitters diffuse across the synaptic gap, to enhance or inhibit, depending on the type of synapse, the receptor neuron's own tendency to emit electrical impulses. The synapse's effectiveness can be adjusted by the signal passing through it so that the synapses can learn from the activities in which their participate. This dependence on history acts as a memory, which is possibly responsible for human memory [JMM96].
The cerebral cortex contains about 10^{11} neurons, this neurons are massively connected. Each neuron is connected to 10^3 to 10^4 other neurons. In total, the human brain contains approximately 10^{14} to 10^{15} interconnections.

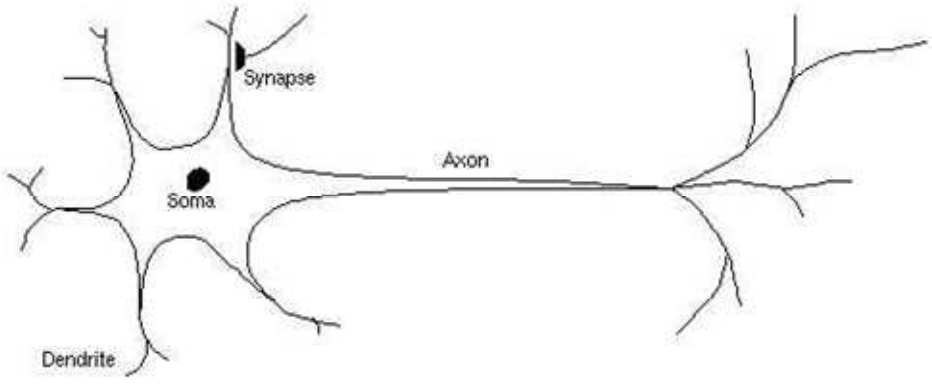


Figure 2.9 A sketch of a biological neuron. Adapted from [JMM96.]

2.2.4.2 Background

Artificial Neural networks (ANNs) are inspired by biological nervous systems and they were first introduced as early as 1960. Nowadays, studies of ANNs are growing rapidly for many reasons:

- ANNs work with pattern recognition at large
- ANNs have a high degree of robustness and ability to learn
- ANNs are prepared to work with incomplete and unforeseen input data

The development of artificial neural networks started several decades ago with the work on the perceptron [Hay94]. The perceptron is basically a simple linear threshold unit, thus able to represent only linear boundaries in the attribute space; its limited representation capabilities have motivated the consideration of more complexes ANNs composed of multiple interconnected layers of perceptrons [Weh98] and called Multi-Layer Perceptrons (MLPs).

ANNs can be viewed as weighted directed graphs in which artificial neurons are nodes are directed edges (with weights) are connections between neurons outputs and neurons inputs. Based on the connection pattern (architecture), ANN's can be grouped into two categories [JMM96]:

- feed-forward networks, in which graphs have no loops.
- recurrent (or feedback) networks, in which loops occur because of feedback connections.

An MLP is characterized by its architecture, training or learning algorithms and activation functions. The architecture describes the connections between the neurons. It consists of an input layer, an output layer and generally, one or more hidden layers in-between. To each connection feeding hidden or output layers corresponds a weight. These weights can then be adjusted to tune the input-output relationship of an MLP to solve a given problem.

MLPs are normally used for supervised learning. In this context, the learning algorithm makes use of both input-output data. Base on a set of input-output data, the weights are

updated so as to minimize the discrepancy between the given outputs and those computed by the MLP from the given inputs. In this research, the most common algorithm, backpropagation, is used. Backpropagation denotes actually an efficient algorithm for computing the derivatives of the MLP output with respect to the weight values. It is used as a main building block construct gradient descent or quasi-Newton algorithms to minimize the discrepancy between MPL outputs and the desired ones provide in a training sample..

Once trained, a network response can be, to a degree, insensitive to minor variations in its input. This ability to see through noise and distortion to the pattern that lies within the inputs is vital to pattern recognition in a real world environment.

A multi-layer network with one hidden layer is shown in Figure2.1010.

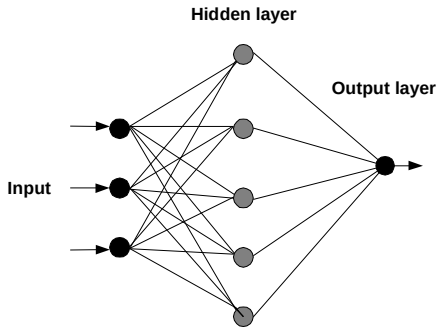


Figure2.10 MLP with a single hidden layer

2.2.4.3The backpropagation method [Hay94,Nat97]

Backpropagation was created generalizing the Widrow-Hoff learning rule to multiple-layer networks and nonlinear differentiable activation functions.

Standard back propagation is a gradient descent algorithm, as is the Widrow-Hoff learning rule, in which the network weights are moved along the negative of the gradient of the performance function. The term *backpropagation* refers to the manner in which the gradient is computed or nonlinear multiplayer networks. There are a number of variations on the basic algorithm that are based on other standard optimization techniques, such as

conjugate gradient and Newton methods. Properly trained backpropagation networks tend to give reasonable answers when presented with inputs that they have never seen. Typically, a new input leads to an output similar to the correct output for inputs vectors used in training that are similar to the new input being presented. The simplest implementation of backpropagation learning updates the network weights and biases in the direction in which the performance function decreases most rapidly – the negative of the gradient

There are two different ways in which this gradient descent algorithm can be implemented: incremental mode and batch mode. In the incremental mode, the gradient is computed and the weights are updated after each input is applied to the network. In the batch mode the weights and biases of the network are updated only after the entire training set has been applied to the network. The gradients calculated at each training example are added together to determine the change in the weights and biases. The batch mode was used in this work.

The learning phase of a layered perceptron is where all its arc weight are adjusted according to a specified learning rule in order to minimize a specified objective function (energy function). A commonly used objective function E is the mean square error (MSE) between the actual neural network outputs and the specified targets for a set of N training patterns. The weight updating problem is to find a set of weights that minimizes the predefined objective function [ElS96].

2.2.4.4 The Levenberg-Marquardt algorithm [HM94]

The Levenberg-Marquardt algorithm was designed to approach second-order training speed without having to compute the Hessian matrix (the square matrix of second partial derivatives of a scalar-valued function). When the performance function has the form of a sum of squares (as is typical in training feed-forward networks), then the Hessian matrix can be approximate as

$$H = J^T J \qquad 2.22$$

and the gradient can be computed as

$$g = J^T e \tag{2.23}$$

Where J is the Jacobian matrix that contains first derivatives of the network errors with respect to the weights and biases, and e is a vector of network errors. The Jacobian matrix can be computed trough a standard backpropagation technique that is much less complex than computing the Hessian matrix.

Suppose that we have a function $V(\underline{x})$ which we want to minimize with respect to the parameter vector $\underline{x}$, then Newtons's methods would be

$$\Delta\underline{x} = -\left[\nabla^2 V(\underline{x})\right]^{-1} \nabla V(\underline{x}) \tag{2.24}$$

where $\nabla^2 V(\underline{x})$ is the Hessian matrix, defined as follows [Zur92]:

$$\nabla_x{}^2 V(\underline{x}) = \nabla_x \left[\nabla_x V(\underline{x})\right] \tag{2.25}$$

$$\nabla_x^2 V(\underline{x}) \triangleq \begin{bmatrix} \frac{\partial^2 V}{\partial x_1^2} & \frac{\partial^2 V}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 V}{\partial x_1 \partial x_n} \\ \frac{\partial^2 V}{\partial x_2 \partial x_1} & \frac{\partial^2 V}{\partial x_2^2} & \cdots & \frac{\partial^2 V}{\partial x_2 \partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial^2 V}{\partial x_n \partial x_1} & \frac{\partial^2 V}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 V}{\partial x_n^2} \end{bmatrix} \tag{2.26}$$

Note that the Hessian matrix is of size nxn and is symmetric. The matrix is often denoted by H, thus $\nabla^2 V(\underline{x})$=H.

and $\nabla V(\underline{x})$ is the gradient and is equal:

$$\nabla V(\underline{x}) \triangleq \begin{bmatrix} \frac{\partial V}{\partial x_1} \\ \frac{\partial V}{\partial x_2} \\ \vdots \\ \frac{\partial V}{\partial x_n} \end{bmatrix} \tag{2.27}$$

If we assumed that $V(\underline{x})$ is a sum of squared function

$$V(\underline{x}) = \sum_{i=1}^{N} e_i^2(\underline{x}) \tag{2.28}$$

Then it can be shown that

$$\nabla^2 V(\underline{x}) = J^T(\underline{x}) J(\underline{x}) + S(\underline{x}) \tag{2.29}$$

$$\nabla V(\underline{x}) = J^T(\underline{x}) e(\underline{x}) \tag{2.30}$$

where $J(\underline{x})$ is the Jacobian matrix

$$J(\underline{x}) = \begin{bmatrix} \frac{\partial e_1(x)}{\partial x_1} & \frac{\partial e_1(x)}{\partial x_2} & \cdots & \frac{\partial e_1(x)}{\partial x_n} \\ \frac{\partial e_2(x)}{\partial x_1} & \frac{\partial e_2(x)}{\partial x_2} & \cdots & \frac{\partial e_2(x)}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_N(x)}{\partial x_1} & \frac{\partial e_N(x)}{\partial x_2} & \cdots & \frac{\partial e_N(x)}{\partial x_n} \end{bmatrix} \tag{2.31}$$

and

$$S(\underline{x}) = \sum_{i=1}^{N} e_i(\underline{x}) \nabla^2 e_i(\underline{x}) \tag{2.32}$$

For the Gauss-Newton method it is assumed that $S(\underline{x}) \approx 0$, and update (2.24) becomes

$$\Delta \underline{x} = \left[J^T(\underline{x}) J(\underline{x}) \right]^{-1} J^T(\underline{x}) \underline{e}(\underline{x}) \tag{2.33}$$

The Levenberg-Marquardt modification to the Gauss-Newton method is

$$\Delta \underline{x} = \left[J^T(\underline{x}) J(\underline{x}) + \mu I \right]^{-1} J^T(\underline{x}) \underline{e}(\underline{x}) \tag{2.34}$$

The parameter μ is multiplied by some factor (β) whenever a step would result in an increased $V(\underline{x})$. When a step reduces $V(\underline{x})$, μ is divided by β. Notice that when μ is large the algorithm becomes steepest descent (with step $1/\mu$), while for small μ the algorithm becomes Gauss-Newton. The Levenberg-Marquardt algorithm can be considered a trust-region modification to Gauss-Newton. Table 2.5 illustrates the Levenberg-Marquardt modification to the backpropagation algorithm.

Table 2.5 Levenberg-Marquardt algorithm. Adapted from [HM94].

1) Present all inputs to the network and compute the corresponding network outputs,
and errors $\left(\underline{e}_q = \underline{t}_q - \underline{a}_q^M\right)$. Compute the sum of squares of errors over all inputs (V(x)).
where $\underline{e}_q$ is the error for the qth input and $\underline{a}_q^M$ is the output of the network when the qth input is presented.
2) Compute the Jacobian matrix.
3) Solve (2.34) to obtain $\Delta\underline{x}$.
4) Recompute the sum of squares of errors using $\underline{x} + \Delta\underline{x}$.
If this new sum of squares is smaller tahn taht computed in step 1, then reduce μ by β,
let $\underline{x} = \underline{x} + \Delta\underline{x}$, and go back to step 1. If the sum of squares is not reduced, then increase μ by β
and go back to step 3.
5) the algorithm is assumed to have converged when the norm of the gradient (2.30) is less
than some predetermined value, or when the sum of squares has been reduced to some error goal.

2.2.4.5 Feed-forward neural networks

Figure *2.11* depicts an example feed-forward neural network. A neural network can have any number of *layers*, *units* per layer, *network inputs*, and *network outputs*. This network has four units in the first layer (layer A) and three units in the second layer (layer B), which are called *hidden layers*. This network has one unit in the third layer (layer C), which is called the *output layer*. Finally, this network has four network inputs and one network output. Some texts consider the network inputs to be an additional layer, the input layer, but since the network inputs do not implement any of the functionality of a unit, the network inputs will not be considered a layer in this discussion.

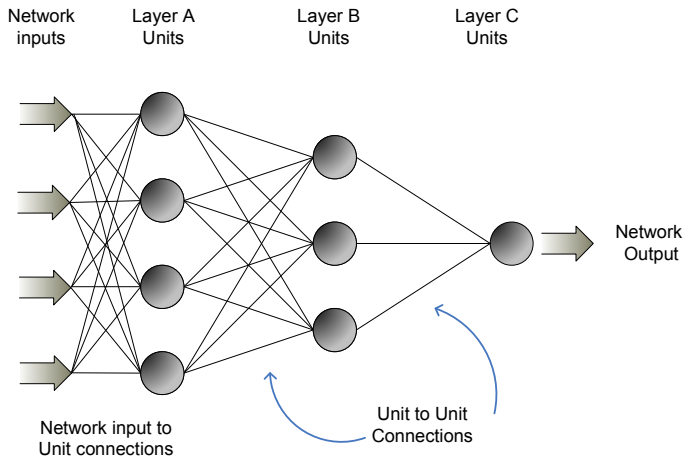


Figure 2.11 MLP with two hidden layers

If a unit is in the first layer, it has the same number of inputs as there are network inputs; if a unit is in succeeding layers, it has the same number of inputs as the number of units in the preceding layer. Each network-input-to-unit and unit-to-unit *connection* (the lines in Figure 2.11) is modified by a *weight*. In addition, each unit has an extra input that is assumed to have a constant value of one. The weight that modifies this extra input is called the *bias*. All data propagate along the connections in the direction from the network inputs to the network outputs, hence the term *feed-forward*

CHAPTER 3 PROBLEM FORMULATION

3.1 ROTOR ANGLE AND SPEED ESTIMATION AND PREDICTION PROBLEM.

It has been shown in the literature [SP89, APSZ95, MCK95, ABDA+96, CKP98, LSTW99, LT95, AR96, SLT99, LT00, KGL01, KK02, BG04] that it is possible to detect impending loss of synchronism by monitoring rotor angles and speeds of the main power plants of an interconnection and that it is possible to use this information to determine in real-time and in a closed loop fashion control actions in the form of generator tripping, dynamic breaking or fast-valving. However, for these emergency control schemes to work properly, it is necessary to obtain highly accurate estimates of rotor angles and speeds. Also, to minimize the risk of detecting and reacting when it is already too late (loss of synchronism may happen in some circumstances within less than a few hundred milliseconds after fault inception), it is necessary to obtain these estimates as quickly as possible, and if possible ahead in time via appropriate prediction schemes.

Synchronous time frame rotor angles and speeds cannot be obtained easily by direct measurements. On the other hand, their estimation can, at least in principle, be carried out from the three-phase voltage and current phasors at the machine's low voltage bus [RLLM+95]. However, given the better accuracy of EHV phasor measurements with respect to medium voltage ones, we suppose that the PMUs will in practice be installed on the EHV side of the step up transformer of the power plant. Also, since for transient instability monitoring it is not necessary to estimate angles and speeds of each individual generator of a given power plant, we propose to use phasor measurements from the EHV side of the step-up transformer to estimate and predict only the rotor angle and speed of the COI of the considered power plant.

A PMU is a power system device that provides measurements of real-time phasors of bus voltage and line currents. Basically, it samples (same time sampling) input voltage and current waveforms using a common synchronizing signal from the global positional satellite, GPS [Pha93], and calculates a phasor (modulus and angle) for the fundamental frequency via Discrete Fourier Transform applied on a moving data window whose width can vary from fraction of a sine wave cycle to multiple of the

cycle [Pha93,IEEE95]. These quantities are typically provided for the nodal voltage and line and transformer currents in all three phases at the bus were the PMU is connected.

In other words, the problem to be tackled can be formulated as follows:

> *Given, a time-series of three-phase voltage phasors and of out-flowing (three-phase) current phasors acquired at the EHV side of a power plant, sampled at a certain rate (typically one or two cycles), and represented in a synchronous reference-frame at nominal frequency, compute an estimate of rotor angle and speed of the centre of inertia of the power plant in the same reference-frame, and compute a prediction of these quantities at the next and subsequent time-steps.*

Our work will aim at showing the feasibility of such a local dynamic state estimation scheme for aggregated rotor dynamics of a power plant and we will also evaluate at the same time how long ahead in time it would be possible to predict angles and speeds.

3.2 APPROACH PROPOSAL TO SOLVE THE PROBLEM.

The relationship between EHV PMU measurements and the dynamic state of the COI of the power plant is essentially non-linear and typically corrupted by measurement noise and modelling uncertainties. Therefore, we propose to use automatic learning techniques, more specifically supervised multilayer perceptron training in order to provide a black-box state estimation algorithm able to cope with such difficulties.

Indeed, it is well known that neural networks, and more generally automatic learning, can cope with uncertainties and non-linearities, at least provided that the dimensionality of their input space remains moderate. Note that this is the case in our analysis, since typically the number of input variables will be in the range of a few tens (at most 100) while by simulation it is possible to generate automatically a very large sample of training scenarios (typically a few thousand). These scenarios can thus cover a representative sample of power system configurations, fault scenarios, modelling assumptions and they can also take into account measurement noise. Training a neural network on such very large and representative scenarios thus may presumably lead to a robust and at the same time very efficient state estimation algorithm.

The approach investigated in this thesis thus essentially consists in generating off-line, and based on numerical simulations, a representative training set composed of system trajectories comprising inputs (sequences of voltage and current phasor measurements) and output sequences of rotor angles and speeds of the COI of the studied power plant. Obviously, it is of paramount importance that the set of simulation scenarios is representative of all power system configurations and fault scenarios.

It is clear that the neural network model needs to be updated when major changes occur in the power system around the studied power plant, such as the installation of a new transmission or generation equipment. On the other hand, since the relationship between the COI of the power plant and PMU measurements depends on the number of generators in operation in the plant, one suggestion is to train different neural network models for each combination of generators in operation, and to use in real-time the one corresponding to the actual configuration.

In order to evaluate the feasibility of the proposed approach we will carry out experiments with two different power system models: the first one is a One-Machine-Infinite-Bus (OMIB) system and the second one is a reduced version of the Mexican Interconnected System (MIS). In our simulations, we have take care to use rather detailed models of the power system dynamics and PMU device and we have considered unbalanced as well as unbalanced conditions (e.g. due to single phase faults).

CHAPTER 4 EXPERIMENTS WITH THE OMIB SYSTEM.

4.1 THE OMIB SYSTEM MODEL [Kun94]

The OMIB test system, shown in Fig. 4.1, represents a thermal generating plant consisting of four 555 MVA, 24 kV, 60 HZ units, connected to the rest of the system through a double circuit transmission line. The equivalent machine is modeled with two damper windings in the q-axis**.** The network reactances shown in Fig. 4.1 are in per unit on 2220 MVA, 24 kV base. Resistances are assumed to be negligible and the infinite bus was modeled as an ideal 3-phase AC voltage source (zero source impedance). Governor is not modeled in this example. Mechanical power *Pm* is considered constant. Although very simple this system is very helpful in understanding transient stability basic effects and concepts [Kun94].

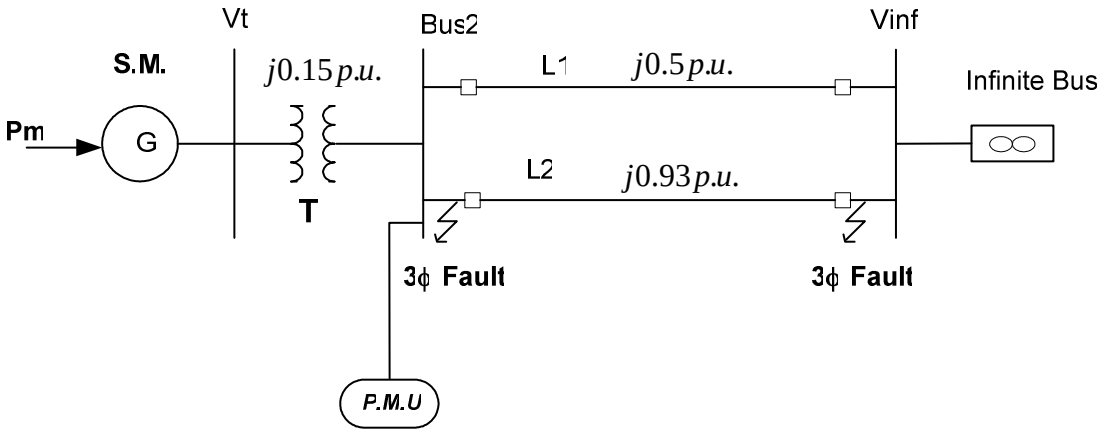


Figure 4.1 OMIB test system (single-phase diagram). Adopted from [Kun94].

The parameters of the synchronous machine in per unit on 2220 MVA, 60 Hz base are provided in Table 4.1.

The excitation system used is an IEEE standard type AC1A, and the parameters of this device are given in Table 4.2 and 4.3 while the block diagram is illustrated in Figure 4.2.

Table 4.1 Synchronous Machine Parameters OMIB test system

Vbase	24 [kV]
Ibase	13.3512312 [kA]
ω	376.991118 [rad/s]
H	3.5 [MW/MVA]
D	0.0 [p.u.]
Ra	0.003 [p.u.]
Ta	0.278 [sec]
Xp	0.15 [p.u.]
Xd	1.81 [p.u.]
Xd'	0.30 [p.u.]
Tdo'	8.0 [sec]
Xd"	0.23 [p.u.]
Tdo"	0.030 [sec]
Xq	1.76 [p.u.]
Xq'	0.65 [p.u.]
Tqo'	1.0 [sec]
Xq"	0.25 [p.u.]
Tqo"	0.070 [sec]

Table 4.2 IEEE Alternator type AC1A Forward Path Parameters

Lead Time Constant (TC)	0.0 sec
Lag Time Constant (TB)	0.0 sec
Regulator Gain (KA)	200 p.u.
Regulator Time Constant (TA)	0.015 sec
MaxReg. Internal Volatge (VAMAX)	7.0 p.u.
MinReg. Internal Voltage (VAMIN)	-6.4 p.u.
Max Regulator Output (VRMAX)	6.03 p.u.
Min Regulator Output (VRMIN)	-5.43 p.u.

Table 4.3 IEEE Alternator type AC1A Exciter Parameters

Rate Feedback Gain (KF)	0.03 p.u.
Rate Feedback Time Constant (TF)	1.0 sec
Exciter time Constant (TE)	0.80 sec
Exct. Constants related to fiel (KE)	1.0 p.u.
Filed circuit Conmutatinf react (KC)	0.20 p.u.
Demagnitizing factor (KD)	0.38 p.u.
Saturation at VE1	0.1 p.u.
Exciter Voltage for SE1	4.18 p.u.
Saturation at VE2	0.03 p.u.
Exciter Voltage for SE2	3.14 p.u.

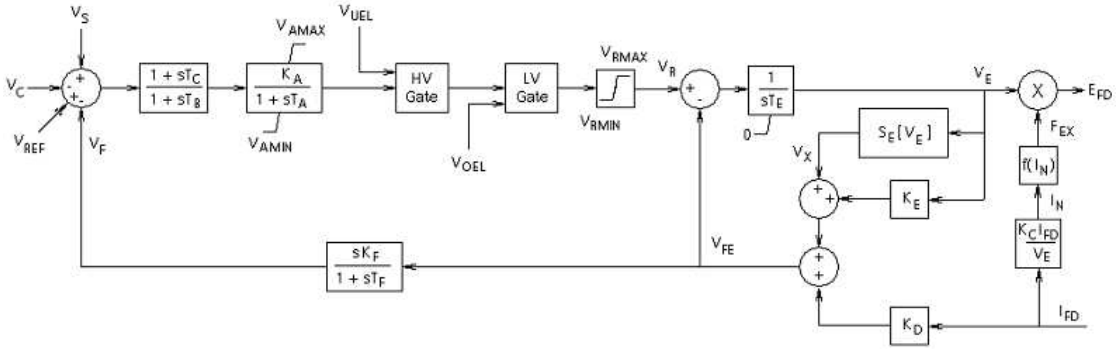


Figure 4.2 Exciter IEEE standard type AC1A. Adopted from [Hyd00]

4.2 DEVELOPMENT OF THE NEURAL NETWORKS FOR ROTOR ANGLE AND SPEED ESTIMATION

The purpose of the ANNs is to estimate the rotor angle and speed of a synchronous machine using voltage and current measurements, which are obtained by the PMU. We have trained two different neural networks: one to estimate the rotor angle (ANN1) and another to estimate the rotor speed (ANN2).

4.2.1 Input selection

The inputs to the neural network ANN1 are the voltage, current, angle of voltage and angle of current at the EHV bus, at time instants t, $t-1$ and $t-2$,(where the time step is 16.66 ms), totaling 12 inputs. The output of the neural network model consists of one neuron representing the rotor angle for a specific operating condition,

$$\delta(t) = f\begin{Bmatrix} v(t), v(t-1), v(t-2), i(t), i(t-1), i(t-2), \theta_v(t), \\ \theta_v(t-1), \theta_v(t-2), \theta_i(t), \theta_i(t-1), \theta_i(t-2) \end{Bmatrix} \tag{4.1}$$

where $v(t)$ and $i(t)$ are the positive sequence terminal voltage and current at the time t, $v(t-1)$, $v(t-2)$, $i(t-1)$ and $i(t-2)$ are the voltage and current at the time $t-1$ and $t-2$, θ_v and θ_i are the voltage and current angles at the same time instants.

On the other hand, for ANN2 we use the same inputs as with ANN1, with three inputs added, the rotor angle obtained from the output of ANN1 at time instants t, $t-1$ and $t-2$. For this reason the number of inputs for ANN2 is 15. The output of the ANN2 consists of one neuron representing the rotor speed as illustrated in Fig. 4.3.

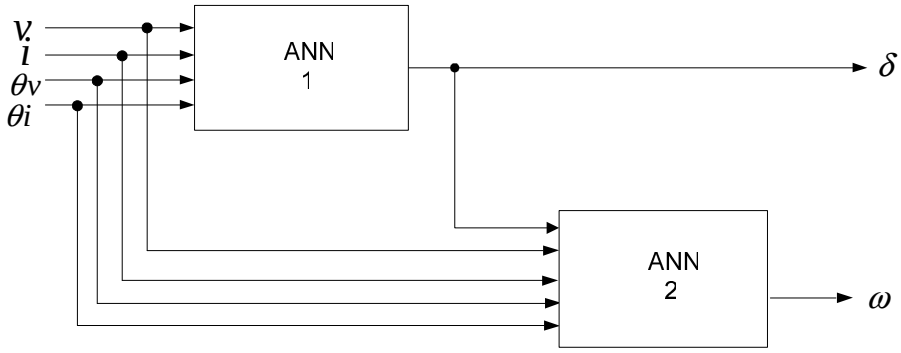


Figure 4.3 Arrangement of the ANNs for angle and speed estimation

4.2.2 Selection of ANN

The ANNs used are of the multi-layer feed-forward type, with one hidden layer. Fig. 4.4 represents the multi-layer feed-forward network used for the purpose.

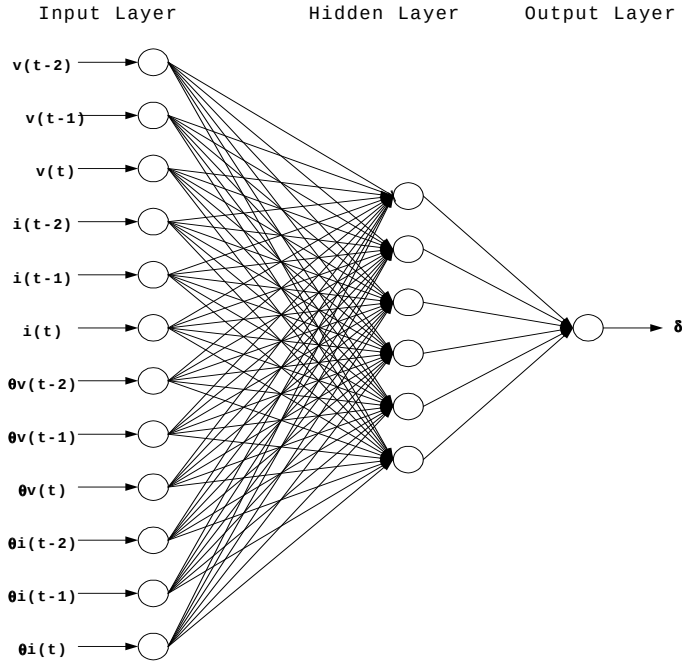


Figure 4.4 Proposed layered feed-forward ANN model for rotor angle estimation

The number of units in the hidden layer is determined experimentally, from studying the network behavior during the training process taking into consideration some factors like convergence rate, error criteria, etc. In this regard, different configurations were tested and the best suitable configuration was selected based on the accuracy level required. The number of hidden units for the ANN1 is 40 and the number of hidden units for ANN2 is 35. Hyperbolic

tangent activation functions are used for these units, while a linear activation function is used for the output neurons for borh of ANNs . The neural networks were trained off-line.

4.3 SIMULATION RESULTS

The *Neural Network Toolbox* from the MATLAB™ [Nat97] software tool was used to create, train and test the neural networks. The training algorithm used is the Levenberg-Marquardt algorithm because it provides fast convergence.
The initial weights as well as the initial biases employed random values between 0-1. The inputs and targets are normalized so that they have values between –1 and 1.

A power system may be subjected to different kinds of disturbances. It is impossible to use all the responses of the teaching system under different disturbances as the training set. The contingencies represented in this well-kwon test system are three-phase short circuit at beginning of the transmission line L2 and at the end of the same line near to infinite bus.
All the three-phase faults were applied at 0.1 sec. The faults were released either by self-clearance or by tripping the faulted line. This is common practice in stability studies. All the disturbances were applied to different generation levels [1100, 850, 600, 500, and 300 MW]. The training data uses 180 patterns, each containing 80 input-output pairs (in average). Total number of input-output pairs is equal to 14400. To test the neural networks 60 unseen patterns are used. Generation of the data for training and testing is summarized in Table 4.4. For each short-circuit and generation level, 3 out of 9 patterns are with fault duration randomly chosen from interval [0.05,CCT-0.01] *ms*, 3 from interval [CCT-0.01,CCT+0.01] *ms*, and 3 from interval [CCT+0.01,0.35] *ms*.

Table 4.4 Generation of training and testing data**.**

	Training				Testing			
Gen. Level (MW)	Self-clearing fault		Tripping the line		Self-clearing fault		Tripping the line	
	Beg. of L2	End of L2	Beg. of L2	End of L2	Beg. of L2	End of L2	Beg. of L2	End of L2
1100	9	9	9	9	3	3	3	3
850	9	9	9	9	3	3	3	3
600	9	9	9	9	3	3	3	3
500	9	9	9	9	3	3	3	3
300	9	9	9	9	3	3	3	3

Testing patterns consist of one pattern from all three, above mentioned, intervals that are not used in training. All real-time environments exhibit some level of noise from instrumentation. The effects of noise on the response of the system are assessed by randomly perturbing the inputs (additive noise uniformly distributed in the range [-0.02,0.02]) to the neural networks. The noise is added to voltage and current magnitude, only. First the ANN1 is trained and tested, according to the procedure described above, and then the same training and testing patterns are used with the ANN2

To generate the ANNs training and validation data sets, the MATLAB™/ *SIMULINK* software tool [Hyd00] is used. Also, using this simulation tool the values of voltage and current phasors to compute the rotor angle and speed using the generator classical model, were obtained. The sampling interval in the simulations is taken equal to 20 ms (every cycle of fundamental frequency, this is reasonable value in view of the fact that modern PMUs are capable to provide the measurements every 1-5 cycles [Tay00]).

As a measure of performance, the root mean square error defined as:

$$RMSE = \sqrt{\frac{1}{p}\sum_{p}(t_p - o_p)^2} \tag{4.2}$$

is determined for each of two ANNs after 1000 iterations of the training rule. In (4.2), p represents the number of input-output training pairs, t_p is the target output for the $p-th$ training, o_p is the output of the ANN. The RMSEs for training and testing are given in Table 4.5. For the comparison, the RMSEs obtained using the classical generator model for all three presented cases are given in Table 4.6 (in equation (4.2) target output is replaced by exact angle and speed values and the output of the ANN with the values obtained using the classical generator model).

Table 4.5 Root mean square error after 1000 iterations

ANN	Training error	Testing error
ANN1	0.0020 (rad.)	0.0092 (rad.)
ANN2	0.0004 (rad./s)	0.0024 (rad./s)

Table 4.6 Root Mean Square Error for the Classical Generator Model

	Stable	Unstable	Critically stable
Angle (rad)	0.1307	0.1607	0.1803
Speed (rad/s)	0.6004	0.9576	0.6988

Results obtained for three cases (stable, critically stable, and unstable) are presented and compared against the computation of the variables based on the classical generator model. All three presented cases correspond to the faults at the beginning of the line L2 released by opening the faulted line. CCT is equal to 0.292 seconds for this particular fault. If the fault duration is less than the CCT, the system response is stable. The evolution of rotor angles and speeds (exact, estimated, and obtained based on classical generator model) are illustrated in Fig. 4.5 and 4.6. As the exact values of the rotor angles and speeds are considered those extracted directly from the simulation model.

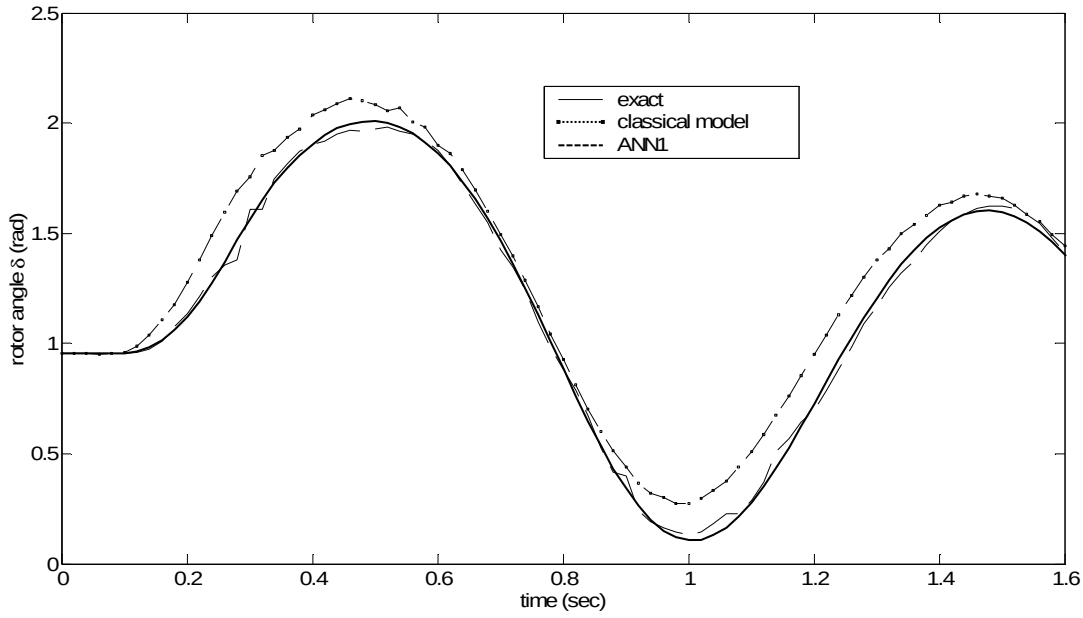


Figure 4.5 Rotor angle (stable case)

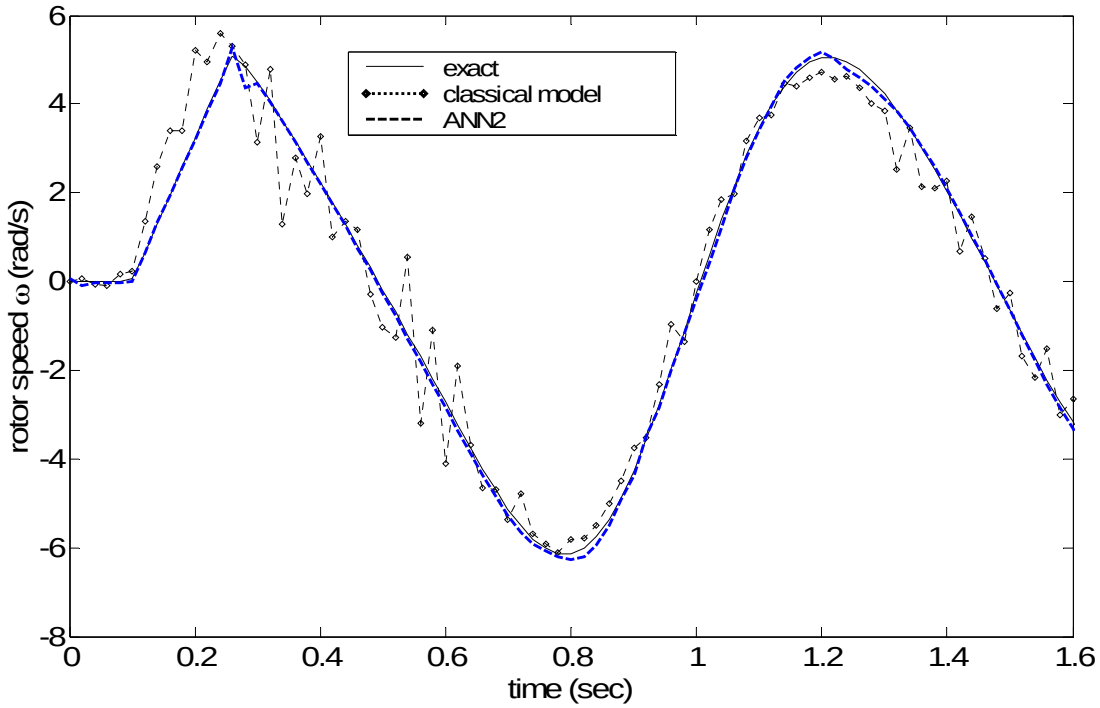


Figure 4.6 Rotor speed (stable case)

An unstable system response (fault duration greater than the CCT) is illustrated in Fig. 4.7 and 4.8. When the fault duration is equal to the CCT system becomes critically stable. Fig. 4.9 and 4.10 represent the variables evolution for this case.

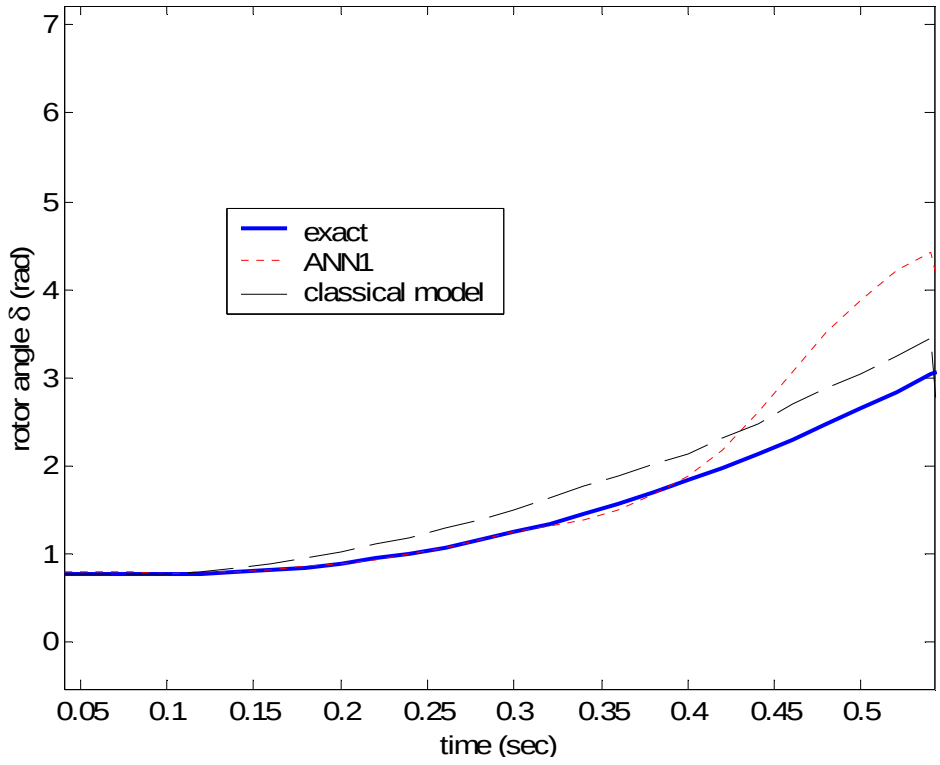


Figure 4.7 Rotor angle (unstable case)

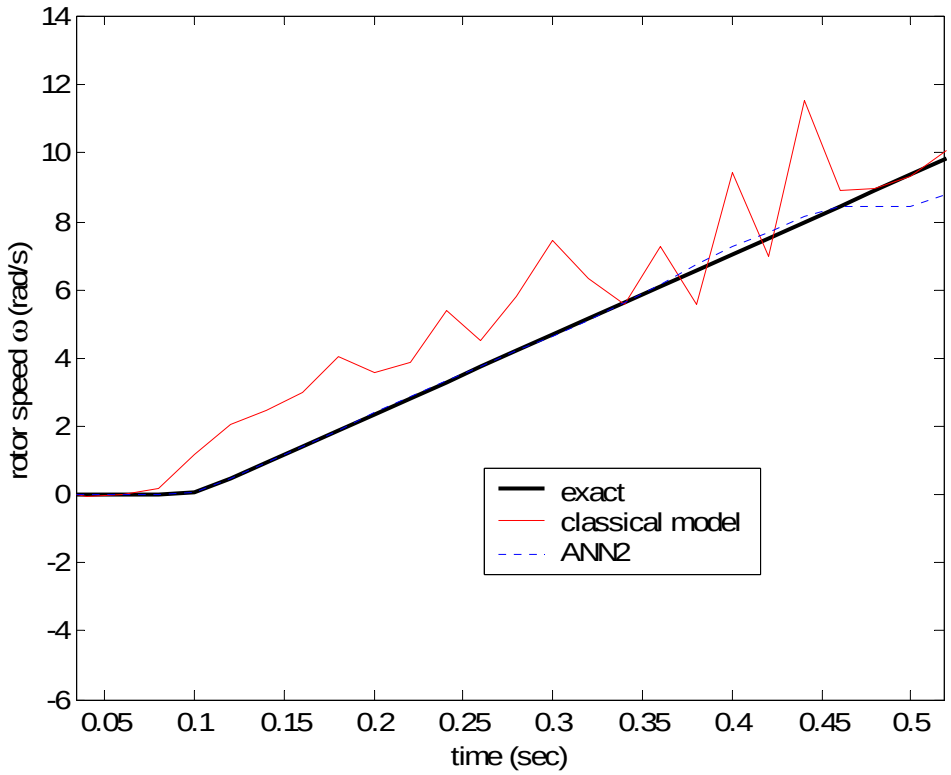


Figure 4.8 Rotor speed (unstable case)

Observe from Fig. 4.5, 4.7 and 4.9 that much better tracking of the rotor angle was obtained by its estimation using the proposed methodology than if we rely on the classical generator model and simple algebraic relations (1.1,1.2) see chapter 1 of this thesis. Presence of the noise in measured variables results in slightly harsh aspect of rotor angle calculated by (1.1). Rather harsh aspect in rotor speed is observable in all presented system responses if analytical formulas (1.1,1.2) derived from the classical generator model are used. The harsh aspects in rotor angle and speed are much less observable in the estimation using the ANNs. If the level of accuracy, in transient stability assessment and control, is high then observed errors in the computation of the variables using (1.1,.1.2) can result in wrong prediction and control actions determination. The results clearly indicate that the ANN-based approach to estimate rotor angles and speeds from phasor measurements, has potential to be useful in tracking transient behavior of a power system following a disturbance.

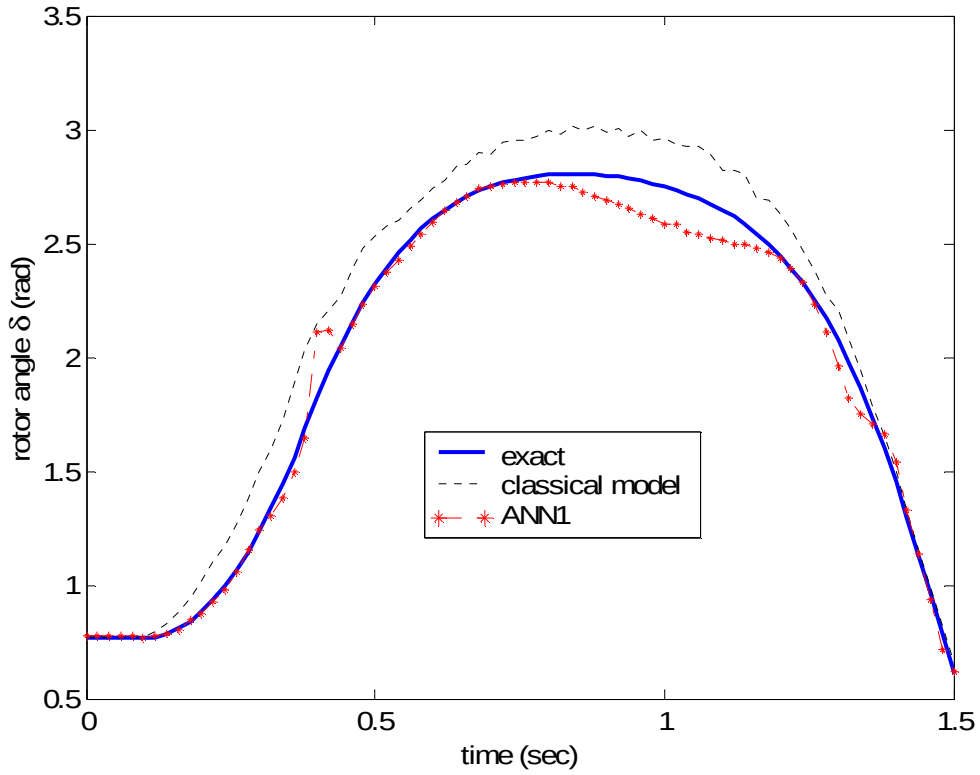


Figure 4.9 Rotor angle (critically stable case)

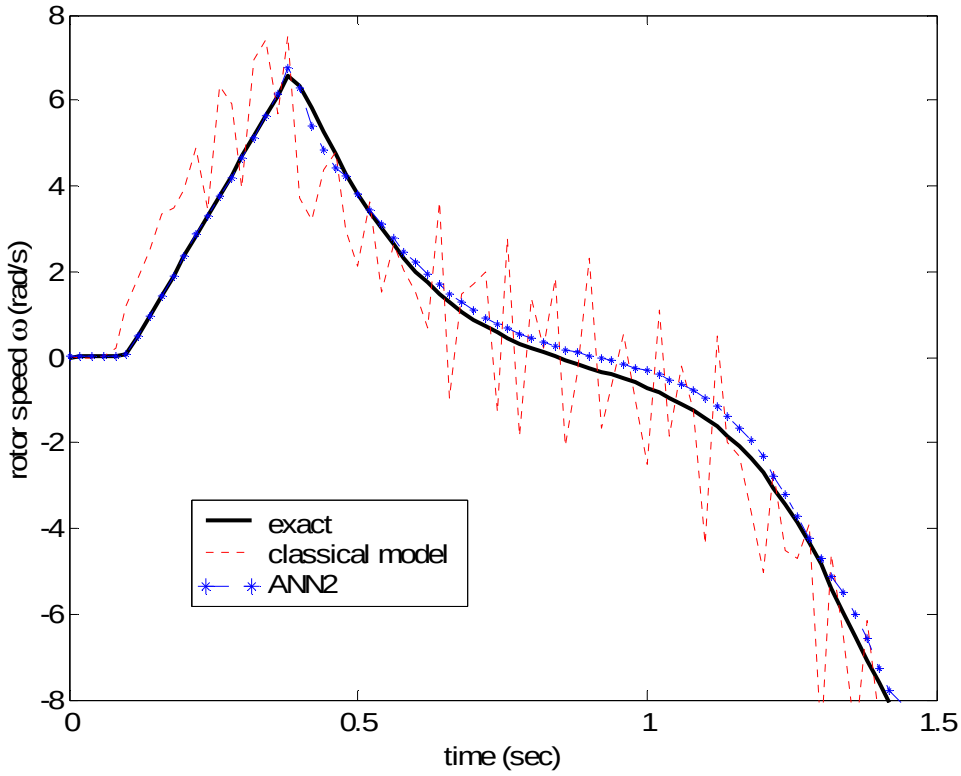


Figure 4.10 Rotor speed (critically stable case)

CHAPTER 5 SIMULATIONS AND TRAINING RESULTS ON THE MEXICAN INTERCONNECTED SYSTEM

5.1 MODELS

In this section we introduce the models used for the Mexican Interconnected System (MIS) that we have used in order to generate our simulation results. All the simulations reported in this thesis were carried out using the PSCAD/EMTDC software [Man03, Man03a].

5.1.1 Mexican interconnected system (MIS)

General structure of the MIS

The bulk Mexican interconnected system comprises a huge 400/230 kV transmission system stretching from the border with Central America to its interconnection with USA. The MIS consists of six areas designated as north (N), north-eastern (NE), western (W), central (C), south-eastern (SE), and the peninsular systems. A simplified diagram of major system elements is shown in Fig. 5.1.

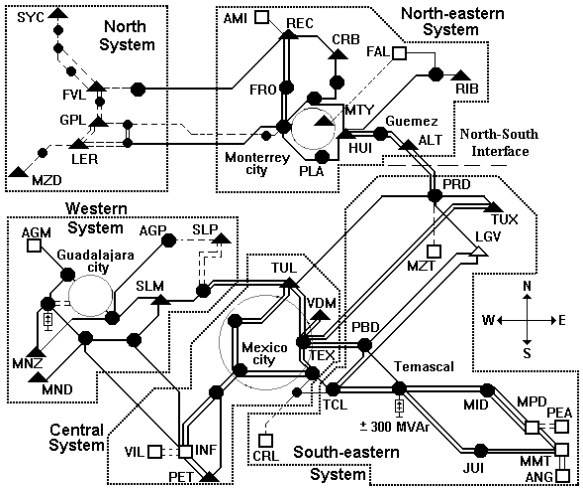


Figure 5.1 Mexican Interconnected System. Adopted from [RRC97]

The test system used in our simulations is a reduced version of the MIS shown on Fig. 5.1. It is formed by one power plant which has 5 hydro-generators, three transmission

lines that connect this power plant with the rest of the system, one load in the 400 kV bus bar of the power plant and the rest of the system is represented by two large synchronous machines and two large equivalent loads (see Figs. 5.3 and 5.6). A detailed model is used to represent these latter two synchronous machines similar to the salient pole rotor one used in the power plant. Excitation systems and governors are also modeled for all synchronous machines. A standard IEEE exciter type AC1A was used as AVR model for these two large synchronous machines..

Parameters of a part of South-East Mexican Interconnected System in PSCAD.

The synchronous machine represents a hydraulic (salient pole) generator model with one damper winding in the q-axis. The set of equations that represent this model are given in Appendix A and correspond to equations (A.10-A.17). Table 5.1 shows the parameters used for each one of the five machines of the hydro-plant.

Table 5.1 Parameters of synchronous machine for the MIS test system (Power Plant)

Synchronous machine parameter for the MIS	
T'_{do}	5.2 sec
T''_{do}	0.029 sec
T''_{qo}	0.034 sec
H	4.3
D	1.0
X_d	0.75 p.u;
X_q	0.43 p.u.
X'_d	0.24 p.u.
X''_d	0.17 p.u.
X''_q	0.17 p.u.
X_l	0.11 p.u.

The exciter is based on an IEEE type SCRX solid-state exciter. The schematic diagram is shown in Fig. 5.2 and its parameter in Table 5.2.

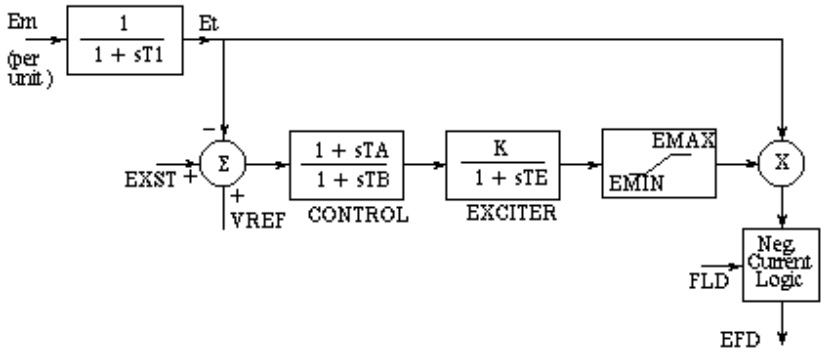


Figure 5.2 IEEE Excitation system type SCRX [MAN03]

Table 5.2 Parameters of the excitation system SRCX

T_A	1.02 (sec)
T_B	15.0 (sec)
K	220.0 (p.u.)
T_E	0.03 (sec)
E_{MIN}	-3.0 (p.u).
E_{MAX}	4.2 (p.u.)

The hydro-turbine is modeled using the same device shown in A.4. The parameters used in our simulations are given in Tables 5.3 and 5.4.

Table 5.3 Hydro-Turbine Rated Conditions

Head at rated conditions	1.0 (p.u.)
Output power at rated conditions	1.0, 0.90, 0.80 and 0.70
Gate position at rated conditions	1.0 (p.u.)
Rated No-load Flow	0.5 (p.u).
Initial Output Power	1.0 (p.u).
Initial Operating Head	1.0 (p.u.)

Table 5.4 Hydro-Turbine Non-Elastic Water column parameters

Water Starting Time (TW)	2.26 (sec.)
Penstock Head Loss Coefficient (fp)	0.02 (p.u.)
Turbine Damping Constant (D)	0.5

The hydro-governor model used is shown in A.5 and its parameters for this specific case are shown in Table 5.5.

Table 5.5 Hydro-Governor Parameters

Dead Value Band	0.0 (p.u.)
Permanent Droop (Rp)	0.25 (p.u.)
Maximun Gate Position (Gmax)	0.8945 (p.u.)
Munimun Gate Position (Gmin)	0.50 (p.u.)
Max Gate Opening Rate (MXGTOR)	0.17 (p.u./s)
Min Gate closing Rate	-0.17 (p.u./s)
Pilot Valve Servomotor Tiem Constant (Tp)	0.05 (sec)
Servo gain (Q)	5 (p.u.)
Main Servo Time Constant	0.2 (sec)
Temporary Droop (Rt)	0.4 (p.u.)
Reset or dashpottime constant	5.0 (sec)

The step-up transformers have a delta-star configuration. Their model is based on the theory of mutual coupling. Table 5.6 gives the parameters of each transformer.

Table 5.6 Transformer parameters for the MIS test sytem

Tmva	150.0 (MVA)
f	60.0 (Hz)
Xl	0.1 (p.u.)
V1	13.8 (kV)
V2	400 (kV)

The transmission lines were modeled using a simple couple PI section model, whose parameter are shown in Table 5.7.

Table 5.7 Transmission line parameters for the MIS test system

Parameter	TL # 1	TL # 2	TL # 3
F	60.0 Hz	60.0 Hz	60.0 Hz
V_{LL} rated	230.0 kV	230.0 kV	230.0 kV
MVA	100.0 MVA	100.0 MVA	100.0 MVA
R	0.0013 p.u.	0.0016 p.u.	0.0041 p.u.
X	0.0177 p.u.	0.0216 p.u.	0.0599 p.u.
B	0.5072 p.u.	1.6181p.u.	1.417 p.u.

Loads were represented as a function of voltage magnitude and frequency, where the load real and reactive power our considered separately using the well known expressions:

$$P = P_0 \cdot \left(\frac{V}{V_0}\right)^{NP} \cdot (1 + K_{PF} \cdot dF) \tag{5.1}$$

$$Q = Q_0 \cdot \left(\frac{V}{V_0}\right)^{NQ} \cdot (1 + K_{QF} \cdot dF) \tag{5.2}$$

In order to represent transformation and any distribution transmission that might exist between the connection point and the actual load, the above characteristics are approximated through representation by a series inductance, selected as 10% of the impedance of the real power portion of the load. The real power is represented as a shunt resistance and the reactive power as a shunt inductor in parallel with the shunt resistor, both connected to ground. The parameters of the load are illustrated in Table 5.8

Table 5.8 Load parameters for MIS

Rated Real Power	47.133 MW
Rated Reactive Power	6.1333 MVAR
Rated Load Voltage (rms L-G)	7.967 kV
Volt index for Power (dP/dV)	2
Volt index for Q (dQ/dV)	2
Freq index for Power (dP/dF)	0
Freq index for Q (dQ/dF)	0
Fundamental Frequency	60 Hz

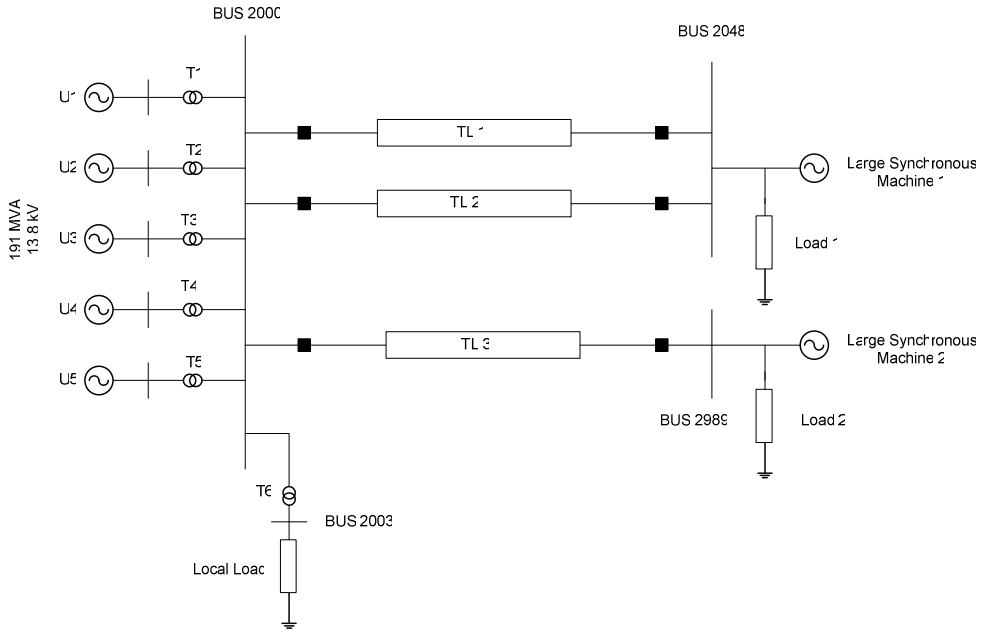


Figure 5.3 One-line diagram of the subsystem of the MIS used as test system

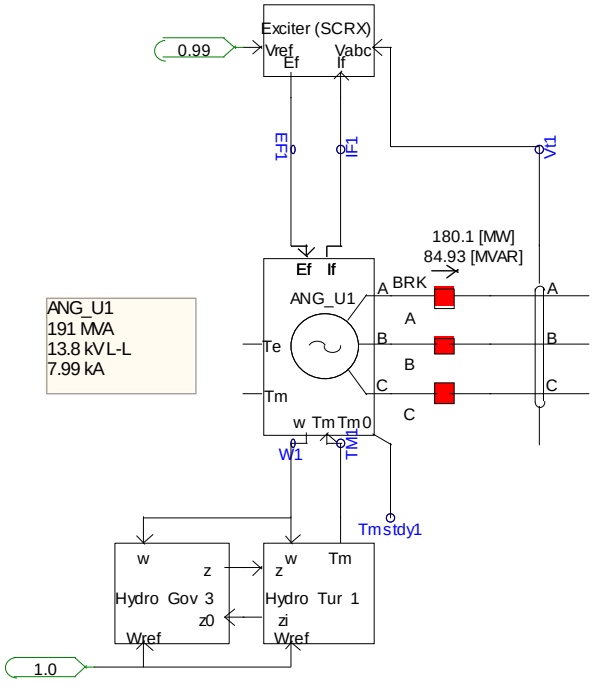


Figure 5.4 Synchronous Machine, excitator and governor in PSCAD/EMTDC simulator.

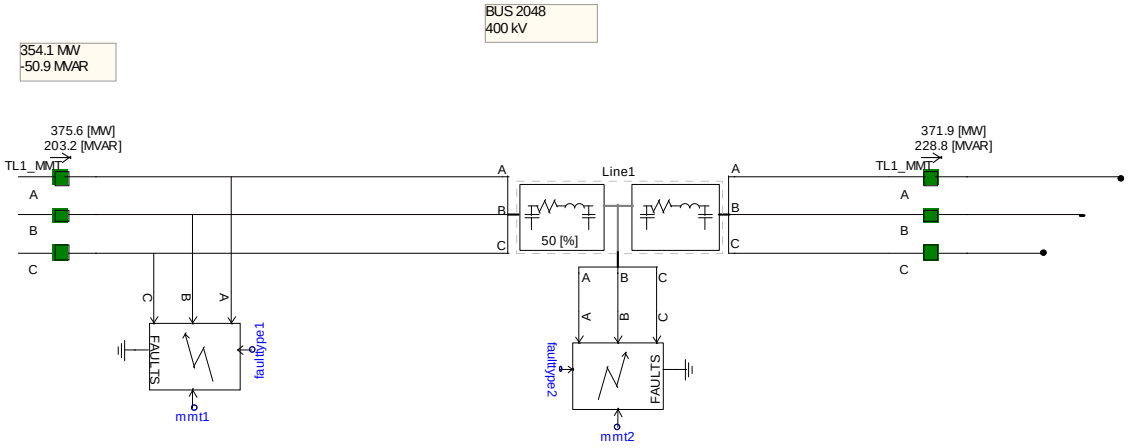


Figure 5.5 Transmission lines representation in PSCAD/EMTDC simulator.

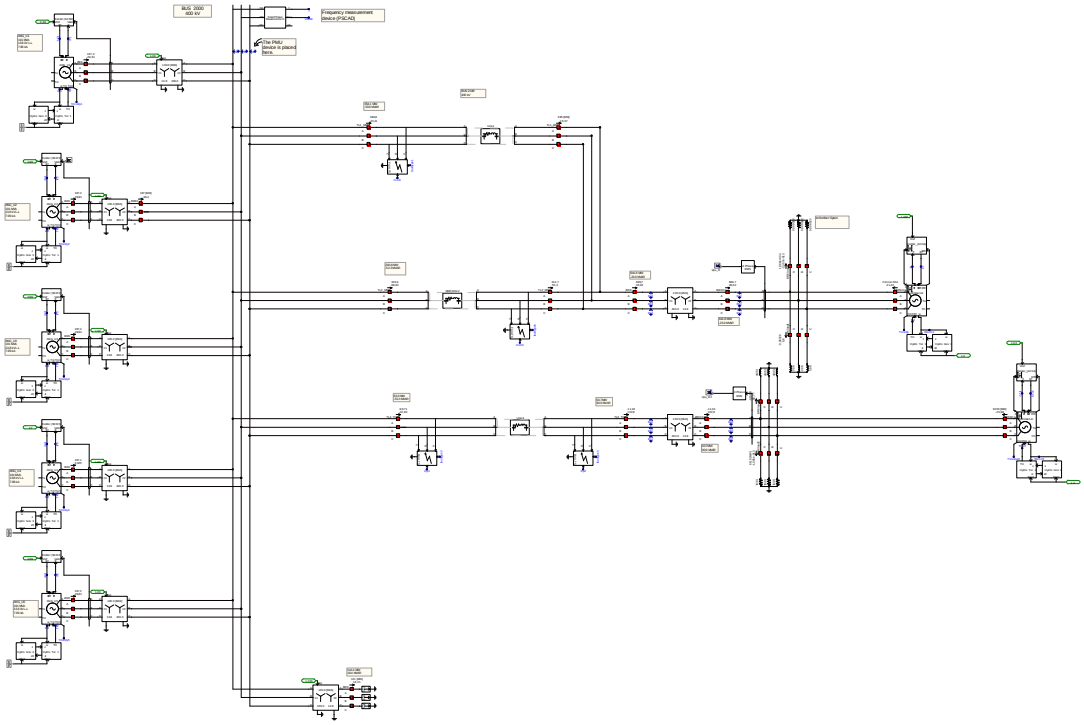


Figure 5.6 Three-phase representation of the test system in the PSCAD software

5.2 DATA BASE GENERATION

5.2.1 Principle

The database generation aims at obtaining a representative sample of simulation scenarios from which the data can be extracted that will be used to train the neural network models used for rotor angle/speed estimation and prediction. Each simulation scenario is defined by first combining a steady state operating condition and a fault scenario (type, location, duration, clearing scheme).

For the MIS test system, the database generation has used different levels of generation to define the steady state operating condition as well as changes in the topology of the network. Single and three phase faults were modeled as disturbances and different places were chosen also to apply the disturbance together with different assumptions about the fault clearing mechanism. The data was generated using PSCAD software, and the training and testing process were carried out with the PEPITo software [Pep04]. Below we describe in details the conditions that were used to generate these simulation results used to build up our training and testing database.

a) Active power generation

We have considered four different levels of active power generation in the studied power plant. In addition to full loading of all units, we have considered 90%, 80% and 70% loading of the units. The load level was essentially kept unchanged, and thus the change in generation of the studied power plant was compensated by increasing the generation of the two large equivalent power plants. The detailed information about the active power generations and the local load are shown in Table 5.9.

Table 5.9 Active Power generation and local load consumption

Active Power	100 % (MW)	**90 % (MW)**	80 % (MW)	70 % (MW)
U1-U5	189.54	**170.0**	151.4	132.8
SM1	19391.7	**19459.4**	19531.4	19603.3
SM2	7592.05	**7618.56**	7646.74	7674.93
Power Plant	947.70	**850.63**	757.22	664.1
Local Load	144.2	**144.3**	144.3	144.4

The steady state rotor angle δ of the units U1 to U5 (the rotor angle reference rotates at nominal frequency ω_n) under these generation levels is summarized in Table 5.10.

Table 5.10 Rotor angle at different generation levels.

Level generation	Rotor angle δ in degrees
100 %	33.22
90%	**30.33 °**
80 %	27.57
70 %	24.22

b) Topology

The changes in the topology of the network were modelled changing the impedance's value of the transmission line 1 and 2. In this case, this value was modified -5 % and +10 % .

c) Disturbances

The disturbances considered in our simulations are the following:

- Three phase fault to ground
- Single-phase fault to ground

The disturbance were releases in three different ways:

- Self-clearing
- Tripping the faulted line
- Tripping and reclosing the faulted line

Figure 5.7 shows the fault locations on the one-line diagram.

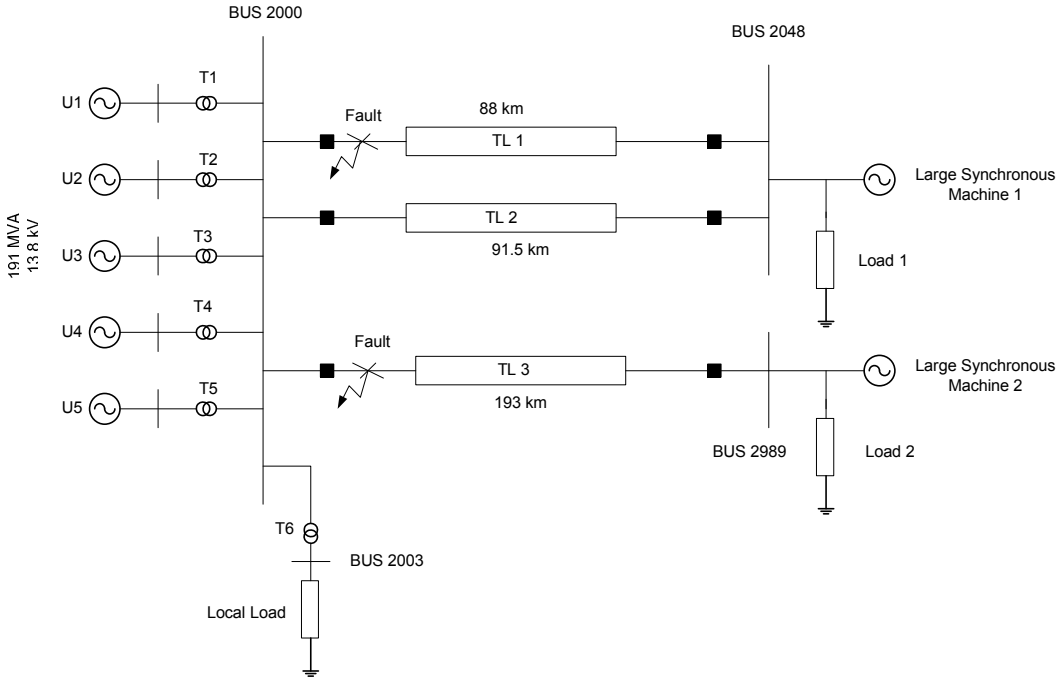


Figure 5.7 Test system diagram

5.2.2 Practical procedure for the database generation

For the generation of the database, direct measurements from PSCAD software of voltage, current with their respective phase angles in three-phase form, i.e. $V_a, V_b, V_c, I_a, I_b, I_c$, etc, were gather for each simulation. In this process, a FFT device is used to represent the PMU in order to obtain these measurements (see Appendix B). Also the actual values of the rotor angle (δ) and the rotor speed (ω) were taken from PSCAD via ideal measurements.

The next step was selected two points in the network, where three-phase and single-phase faults are applied. The near end of the transmission lines #1 and #3 was selected for this proposes. Where place # 1 is the beginning of TL #1 and place #2 the beginning of TL #3.

For each steady state condition (combination of power generation level and topology), the Critical Clearing Time (CCT) was calculated and with this value we selected different fault clearing times in order to obtain stable and unstable cases (loss or not of synchronism) of the generators of the power plant.

For example, for the100 % generation level base case, three-phase fault placed # 1 for the fault location and release by self-extinction, the CCT is 0.30 sec., and we used the fault clearing times shown in Table 5.11.

Table 5.11 Fault clearing times used for 100 % level generation, case base, 3-phase fault

0.15	0.17	0.18	0.19	0.20	0.21	0.22	0.24	0.25	0.26
0.27	0.28	0.29	**0.30**	0.302	0.31	0.32	0.33	0.34	0.35

These 20 fault clearing times have been combined with single vs three phase fault assumptions and three clearing schemes mentioned about, leading to 120 PSCAD time-domain simulations for the 100% load level case base and location #1 of the fault.

In the same way, we considered the 24 combinations of 4 generation levels, 3 topology assumptions and 2 fault locations, yielding a total number of 2880 simulations.

The duration of the simulation period was fixed to 1.5 sec, the fault was applied at 0.1 sec and PSCAD simulations where carried with a fixed time step of 50 μs.

5.2.3 Post-processing of the simulation results

When a simulation with PSCAD is finished, the next step is to arrange the raw output information contained in the output files generated by PSCAD into files that MATLAB and PEPITo can read. In these files we include all the inputs and outputs to be used in the

training of the neural network, and keep only the values at the appropriate times step (one sample every cycle, at 60Hz).

The variables used as input are:

- Voltage at the bus 2000.
- Currents flow the three transmission lines.
- Current through local load (bus 2003)

We have three phase measurements with their respective phase angles for each variable as follow:

Table 5.12 Input arrangement to the NN using all the attributes

voltage at bus 2000	V_a	V_b	V_c
and phase angle	θ_a	θ_b	θ_c
current T.L. #1	I_{1a}	I_{1b}	I_{1c}
and phase angle	θ_{1a}	θ_{1b}	θ_{1c}
current T.L. # 2	I_{2a}	I_{2b}	I_{2c}
and phase angle	θ_{2a}	θ_{2b}	θ_{2c}
current T.L. # 3	I_{3a}	I_{3b}	I_{3c}
and phase angle	θ_{3a}	θ_{3b}	θ_{3c}
current local load (bus 2003)	I_{load}	I_{load}	I_{load}
and phase angle	θ_{load}	θ_{load}	θ_{load}

All in all, including the recording of output values (rotor angles and speeds), this leads to a files containing 2912 numbers for each simulated scenario.

Taking into account the fact that in our work we use input values of three successive time samples $(t, t-1, t-2)$ in order to estimate rotor angles and and speeds at time t and $t+1$, each input/output pair is represented by $X_{(t-2)}, X_{(t-1)}, X_{(t)}, Y_{(t)}, Y_{(t+1)}$, where $X_{(t)}$ is the

value of input attribute vector at time t, $X_{(t-2)}$ and $X_{(t-1)}$ correspond to the value of each attribute at the two previous time steps, and $Y_{(t)}$ (respectively $Y_{(t+1)}$) correspond to the two output quantities at the present time and one time step ahead into the future.

All in all, the resulting database thus contained roughly 24 million values, organized into a matrix of (roughly) 260,000 lines and 94 columns.

5.2.4 Separation of the database into learning and test sets.

The above database was split into a learning and a test set in the following way. For each conditions, combining a steady state condition, a fault location, a fault type and a fault clearing scenarios, the 20 scenarios corresponding to 20 different fault clearing times were separated into a subset of 15 learning scenarios and 5 testing scenarios, chosen so as to have both stable and unstable case in both subsets.

This leads to a total number of about 200,000 input output pairs in the learning set and 60,000 in the test set.

5.3 RESULTS

The results are organized into 4 successive sections. The first section provides preliminary results about the design of a neural network scheme for the estimation of the rotor angle directly. Since these results were not satisfactory, we propose in the second section an alternative scheme were the outputs of the neural network target the sinus and cosinus of the rotor angle (we call this the rectangular representation of the rotor angle). The results obtained with this scheme are much better, and we believe sufficiently accurate for practical use. The third section focuses on the estimation of the rotor speed, and the two last sections provide some preliminary results concerning the prediction of rotor angle and speed some time steps ahead in time.

5.3.1 Direct estimation of the rotor angle δ.

In the preliminary trials reported in this section, the output value on which the neural networks are trained is the rotor angle, relative to a synchronous reference. For all these trials we used 50% of the available data, in order to reduce computing times.

We study first the case were we train the neural network for both unbalanced and balanced fault conditions. In that case, we considered a training set of 88200 samples and a test set of 28800 samples and 90 input variables.

Then we study a simpler situation, were only balanced fault conditions are considered, which allows to reduce the number of training and testing samples by a factor of 2 and the number input variables by a factor of 3 (considering only the quantities of a single phase, instead of all three phases).

We have considered different neural network structures using either one or two hidden layers with various numbers of units.

Case MLP configuration 90 – X - δ

We used as inputs all the 90 variables given in Table 5.12, the output is the rotor angle, each variable is estimate in a different Neural Network. The networks are trained on

88200 samples drawn from both balanced and unbalanced fault conditions. They tested on an independent test set of 28800 drawn from the same balanced and unbalanced conditions simulated with different fault clearing times.

With respect to number of hidden units used, 5, 15, 20 and 30 hidden units were employed. The results for this kind of configuration are shown in Table 5.13.
The mean square error (MSE) is defined as

$$MSE = \frac{1}{p}\sum_{p}(t_p - o_p)^2 \tag{5.1}$$

where p represents the number of input-output training pairs, t_p is the target output for the $p-th$ training, o_p is the output of the ANN. In the case of estimation of the rotor angle the units are given in degrees and for the case of rotor speed estimation the unit is in p.u. the number of training cycles varying from 50 to 100, in our case cycles means the number of epochs during the ANN training process. Linear correlation coefficient is a statistic representation how closely two variables co-vary. It can vary from -1 (perfect negative correlation through 0 (no correlation) to +1 (perfect positive correlation).
The correlation factor is defined by [Sap90]:

$$r = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{s_x s_y} \tag{5.2}$$

where s_x and s_y are the standard deviation of x and y,

$$s_x^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x}) \tag{5.3}$$

$$s_y^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y}) \tag{5.4}$$

and the numerator is the covariance observed.

Table 5.13 MLP configuration 90-X-δ - Errors and correlation factor varying the number of hidden units

Net configuration	MSE (LS)	MSE (TS)	Correlation factor (TS)	Error min TS(deg)	Error MaxTS (deg)	cycles
90 - 5 - δ	0.1014	0.1248	0.9032	-732.44	1.19E3	50
90 – 15 - δ	0.03567	0.0475	0.9644	-717.16	870.14	50
90 – 20 -δ	0.01964	0.02767	0.9796	-461.77	568.24	50
90 – 20 - δ (a)	0.01134	0.01912	0.9857	-682.90	529.00	100

We observe that the MSE error decreases (both on the LS and on the TS) and the correlation factor increases when we increase the number of hidden units in the MLP under study. However, the maximum and minimum errors (computed on the test sample) are very large (-732.44 ° and 1.19 E3 °). Notice that increasing the number of training cycles did not allow us to reduce these errors significantly.

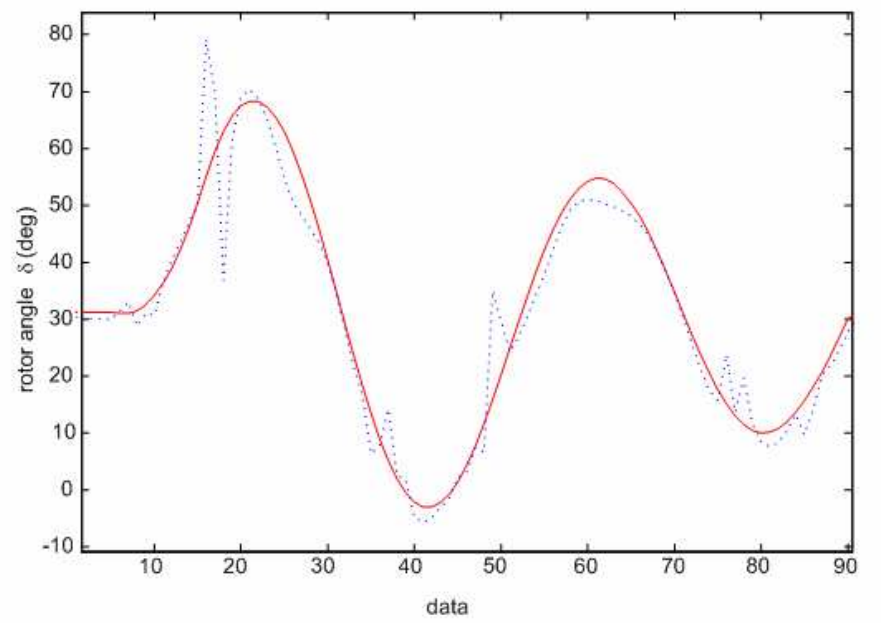
Case MLP configuration 90 – X – X - δ

In the next trials we have added one hidden layer. Table 5.14 shows the obtained results after training the MLP.

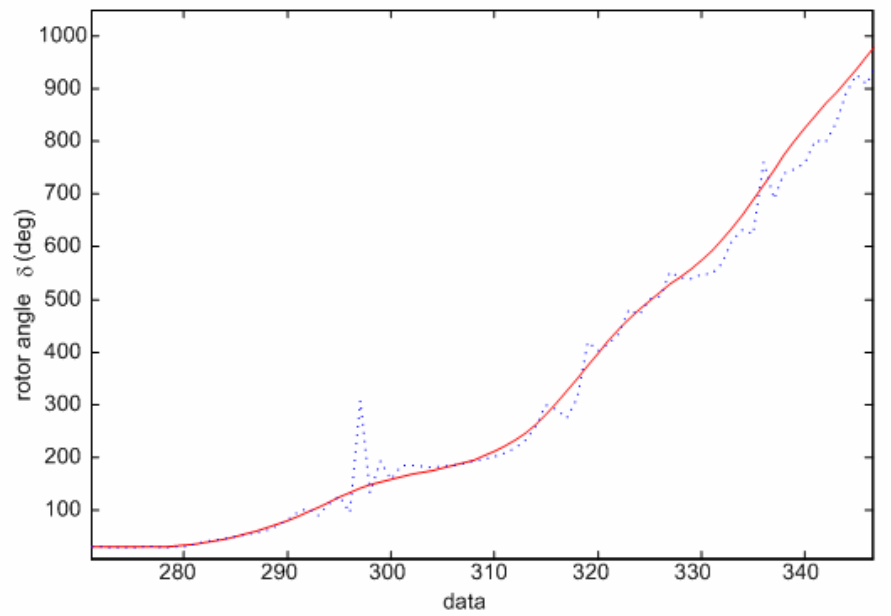
Table 5.14 MLP configuration 90-X-X-δ - Errors and correlation factor varying the number of hidden units

Net configuration	MSE (LS)	MSE (TS)	Correlation factor	Error min(deg)	Error Max(deg)	cycles
90 -15-15- δ	0.003288	0.008084	0.994	-455.62	510.069	70
90 –18–18- δ	0.002406	0.007462	0.9945	-577.77	575.21	100

Figure 5.8 illustrates these results graphically.



(a)



(b)

Figure 5.8 True vs. estimate rotor angle for MLP 90 -18-18-δ. Stable test scenario (a) and unstable test scenario(b)

We observe that this configuration leads to significant improvement of the average error rates with respect to the previous configuration, but the maximum and minimun errors remain still very large.

Given the disappointing results obtained above, we made a further set of trials on a simplified version of the problem, where we have considered only balanced (i.e. three-phase) fault conditions. This allowed also to significantly reduce the amount of training and testing data (the number of samples is devided by a factor 2, and the amount of input variables by a factor 3, by using only the values of phase A). These results are reported in the next 3 subsections.

Case MLP configuration 30 – X - δ

The number of hidden units used for this NN configuration were of 7 and 10.
Table 5.15 summarizes the results obtains using this kind of configuration.

Table 5.15 Results using 30-X-δ MLP configuration (angles in degrees)

NN configuration	30-7-δ	30-10-δ
MSE(LS)	0.01435	0.02140
MSE(TS)	0.02090	0.03070
Correlation factor	0.9838	0.9792
Error min.	-539.89	-533.02
Error max.	717.77	696
cycles	150	100

Maximum and minimum errors are again unacceptably large.

Case MLP configuration 30 –X - X - δ

This new configuration tested is basically the same that used above, with the difference that we have added one hidden layer.
The results obtained with this configuration are shown in Table 5.16

Table 5.16 Results using 30-X-X -δ MLP configuration (angles in degrees)

Net configuration	MSE (LS)	MSE (TS)	Correlation factor	Error min(deg)	Error Max(deg)	cycles
30 - 25 -5 - δ	0.002348	0.00665	0.9949	-190.52	538.70	100
30 – 25 -25- δ	0.002729	0.00720	0.9945	-487.48	439.01	150
30 – 20 -5-δ	0.003098	0.00852	0.9935	-271.35	495.16	100

We observe again an improvement in accuracy resulting from the introduction of a second hidden layer. Nevertheless, there are still very large errors.

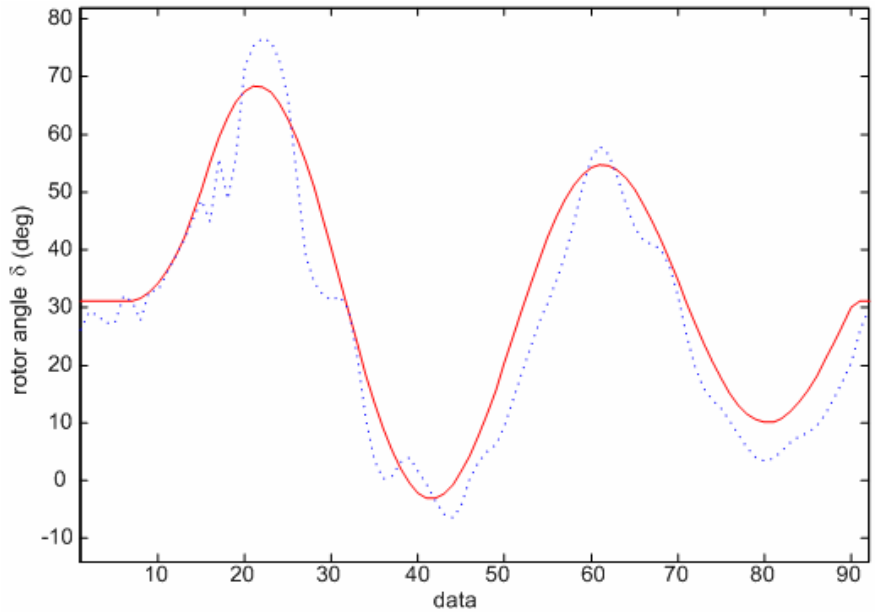
Case MLP configuration 18 –X - X - δ

Finally, a further set of simulations were obtained by further reducing the input variables to the neural nets, by using the magnitude and phase angle of voltage measurements of phase A, and only the current magnitudes as inputs to the neural network. Current phase angles were thus not used for this configuration, leading to 18 input variables. The best results were again obtained with two hidden layers and are displayed in Table 5.17.

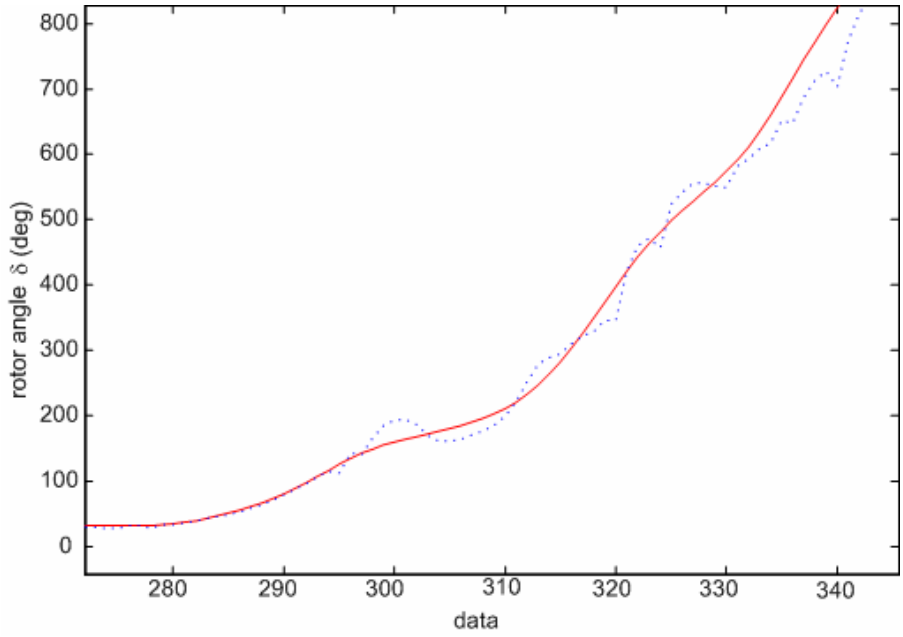
Table 5.17 Results using 18 –X –X - δ configuration (angles in degrees)

Net configuration	MSE (LS)	MSE (TS)	Correlation factor	Error min(deg)	Error Max(deg)	cycles
18 - 20 -10 - δ	0.002299	0.005969	0.9954	-191.69	480.69	250
18 – 20 -20- δ	0.001962	0.005391	0.9959	-154.30	526.39	200
18 – 30 -10-δ	0.0020	0.00551	0.9958	-282.29	481.23	250

As one can see from these results, this simplification only marginally improved accuracy, which still remains below expectations. To further highlight this, Figure 5.9 provides a graphical comparison of actual rotor angle and those obtained from neural network predictions. These latter show some very large peaks, apparently appearing at random places during the transients.



(a)



(b)

Figure 5.9 True vs. estimate rotor angle for MLP 18 -20-20-δ. Stable test scen ario (a) nad unstable test scenario(b)

5.3.2 Estimation of δ by using a rectangular representation.

On the use of a rectangular representation of the rotor angle.

The difficulty to obtain accurate results in the direct estimation of rotor angles, as carried out in the previous sections, can be explained by the fact that these rotor angles become quite large in the context of unstable scenarios (actually unbounded), while the phases and amplitudes of the phasor measurements used as inputs to the neural network remain bounded (phases within the interval [-180°,180°], amplitude between 0 and some upper bound fixed by system parameters). This makes it difficult, if not impossible, to yield an estimation scheme using a limited number of past measurements to provide good accuracy both for the large excursions (over 1000°) of the rotor angle in unstable cases and the smaller variations (typically 10°-80°) in stables ones. We believe that the training of the MLPs in such conditions actually leads to overfitting the unstable scenarious at the price of a rather bad approximation of the stable ones, as shown in Figs. 5.8 and 5.9.

In order to circumvent this problem, we propose to use instead of the direct estimation of the rotor angle δ, an indirect scheme where two neural networks are trained in parallel, in order to respectively provide an approximation of sin(δ) and cos(δ). Indeed, these values have the advantage of remaining bounded even in unstable conditions, and at the same time they vary smoothly over time. On the other hand, the rotor angle can be straightforwarly recovered from these values (up to a multiple of 360°).

It is import remake that the estimation of rotor speed (ω) we don't find this problem, because in both cases stable and unstable the value for omega remaining bounded.

Rectangular representation of the rotor angle for three phase faults cases only.

Table 5.18 shows training and testing results obtained for two such MLPs, both with two hidden layers of 8 neurons each, and a linear output neuron. In these simulations we considered only the three phase faults as the last configuration mentioned above, with the same training and testing scenarios. Notice that in the present case we also decided to drop from the 30 input variables the values corresponding to $t-1$ cycle, since we found that they didn't bring a significant amount of information (thus the MLPs use only twenty

inputs). In order to recover the rotor angle from the approximations of sin(delta) and cos(delta), we used the two MLPs obtained after 150 cycles of training, which corresponded to convergence of the MSE both on the training and on the testing samples. Notice that no overfitting is observed with these structures and input and output variables.

Table 5.18 Results of training and testing MLPs with rectangular representation of δ

Net configuration	MSE (ls)	MSE (ts)	Correlation factor(ts)	Error min(ts)	Error max(ts)	cycles
20-8-8-sin(δ)	0.004229	0.005422	0.9951	-0.593	0.586	50
20-8-8-sin(δ)	0.003219	0.004265	0.9962	-0.563	0.495	100
20-8-8-sin(δ)	0.003118	0.005422	0.9962	-0.546	0.474	150
20-8-8-cos(δ)	0.001957	0.002743	0.9976	-0.352	0.331	50
20-8-8-cos(δ)	0.001378	0.002121	0.9982	-0.510	0.323	100
20-8-8-cos(δ)	0.001205	0.001840	0.9984	-0.547	0.335	150

Compared to the previous results we obtained a significant increase in accuracy when using this approach to estimate the rotor angle. Indeed, except for a small number of large errors, the approximation provides a standard error (on the test sample) of 3.73 degrees, with minimum and maximum errors of respectively -32.4 and 26.5 degrees. The quality of this estimation scheme is illustrated on next figures.

Figure 5.10 shows a few test scenarios both the true (i.e. taken from the simulations) rotor angle, and the approximation recomputed from the two MLP outputs. Both are trimmed to the interval [-180°,180°].

Figure 5.11, on the other hand, illustrates the fact that sometimes there remain large errors, but we see that these are merely due to the fact that the signal discontunuety at 180° does appear at two different (but successive) time steps for the true signal and its estimation. Although such errors appear as large and influence the accuracy statistics in a negative way, from a practical point of view they are not problematic.

Figure 5.12 further highlights the overall performances by a scatter-plot representing the true angle and its estimation for all the 14400 test samples. It shows that, except for the already discussed large errors, the approximation is of very good quality.

In figure 5.13 we illustrate two training scenarios (the two first in our data set), showing some small estimation errors at fault application and clearing instants. These small peaks of limited amplitude of one or two examples are typical errors incurred by this scheme. We deem that this kind of estimation error could also be easily detected and corrected in real time, taking advantage of the fact that rotor angles must vary smoothy.

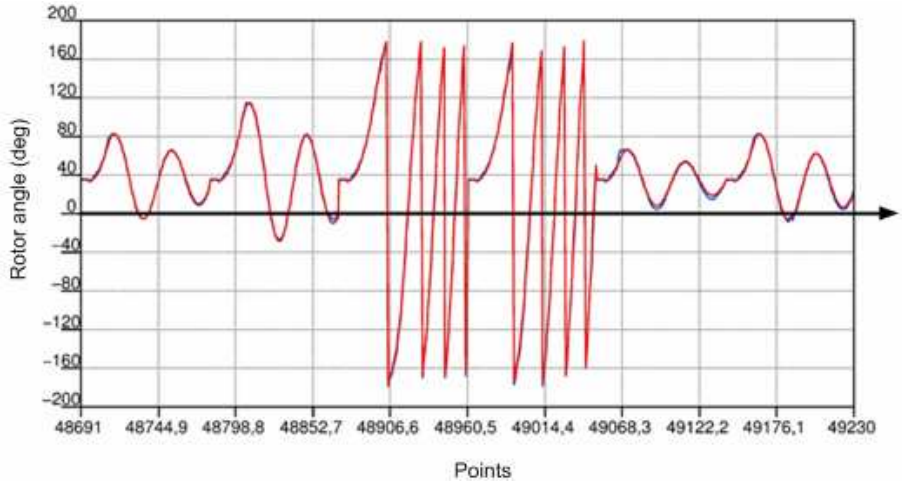


Figure 5.10 Rotor angle vs estimation for six test scenarios (2 stable, 2 unstables, 2 stables). 20-8-8-sin(δ)and 20-8-8cos(δ) trained for 150 cycles.

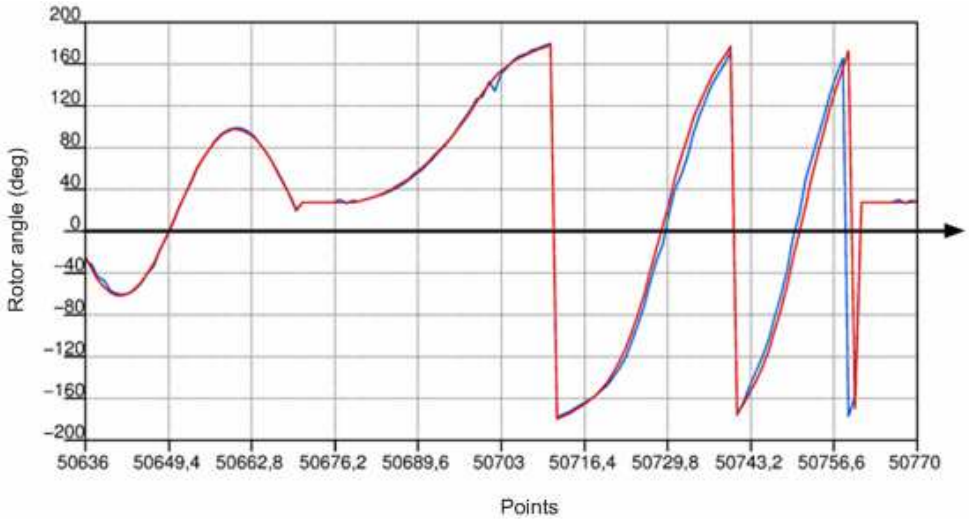


Figure 5.11 Rotor angle vs estimation for a stable followed by an unstable test scenario.

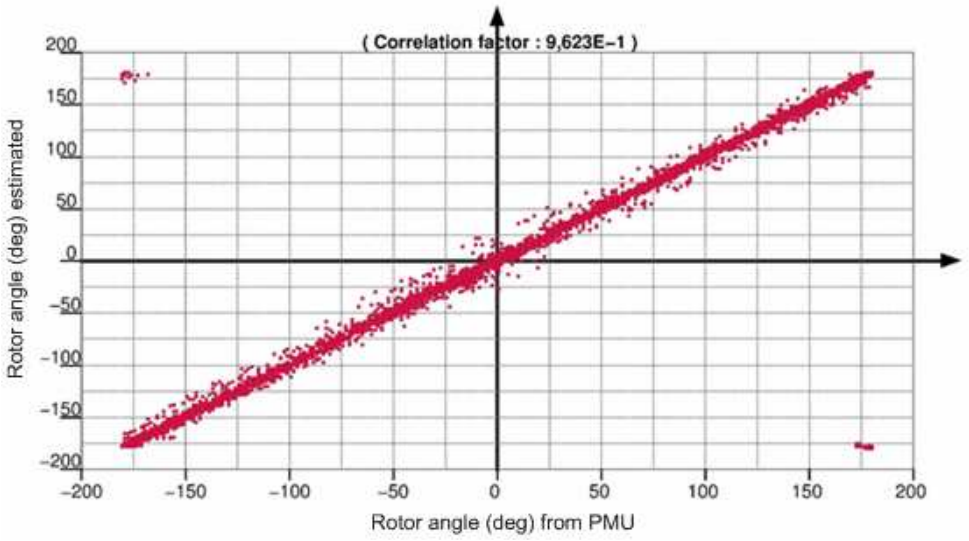


Figure 5.12 Rotor angle vs estimation for all 14400 test sample**.**

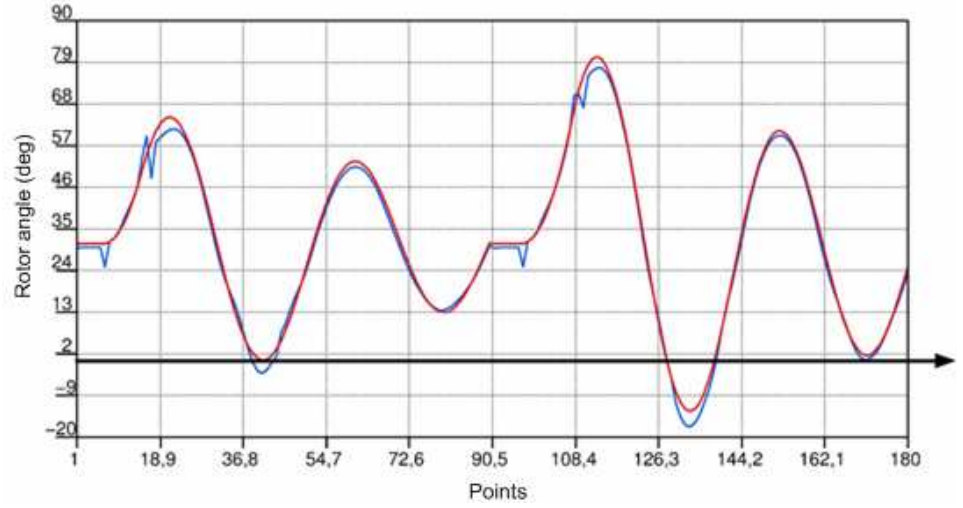


Figure 5.13 Rotor angle vs estimation for all the two first test scenarios

Rectangular representation for both balanced and unbalanced faults.

Table 5.19 shows training and testing results.

Table 5.19 Results using 60 -10 -10 –sin(δ) /cos(δ) MLP configuration (angles in degrees)

Net configuration	MSE (LS)	MSE (TS)	Correlation factor	Error min	Error Max	cycles
60-10-10-cos(δ)	8.83E-4	0.00144	0.9988	-0.318	0.467	100
60-10-10-sin(δ)	0.002580	0.004116	0.9963	-0.5079	0.5290	100

Figure 5.14 shows for two test scenarios the true rotor angle (solid line), and the approximation recomputed from the two MLP outputs (dotted line).

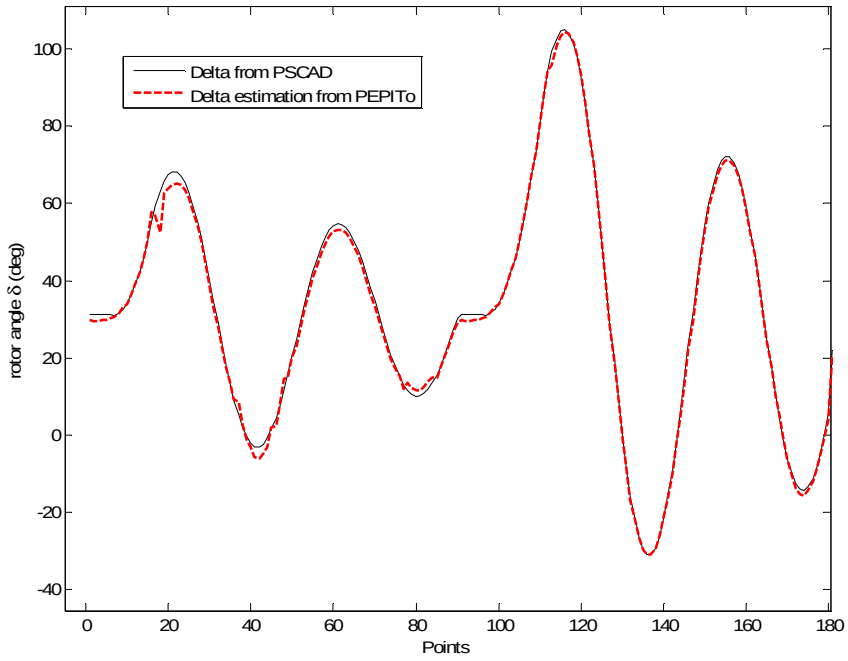


Figure 5.14 Rotor angle vs estimation for a stable test scenario.

Figure 5.15 shows two test unstable scenarios, trimmed to the interval [-180°,180°].

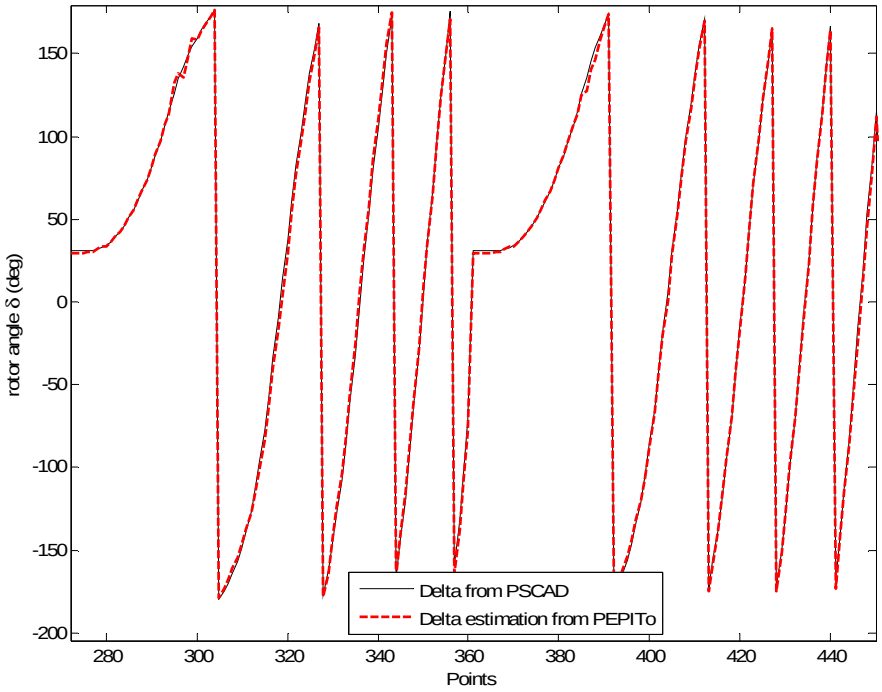


Figure 5.15 Rotor angle vs estimation for unstable test scenarios.

Figure 5.16 is devoted for a few test scenarios that represent a single-phase fault.

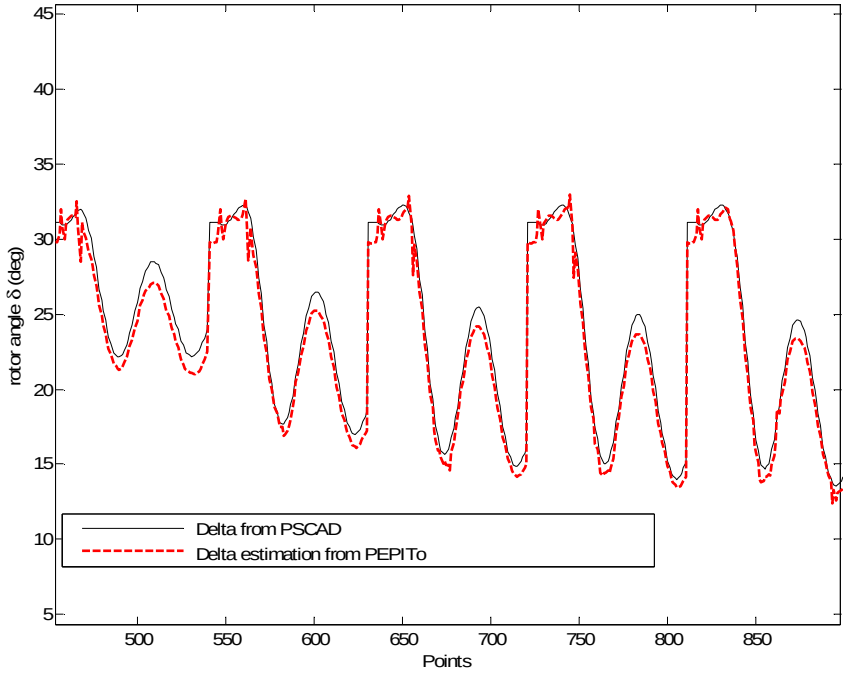


Figure 5.16 Rotor angle vs estimation for single phase fault occurrence.

Figure 5.17 shows a single phase faults followed by a stable three phase fault.

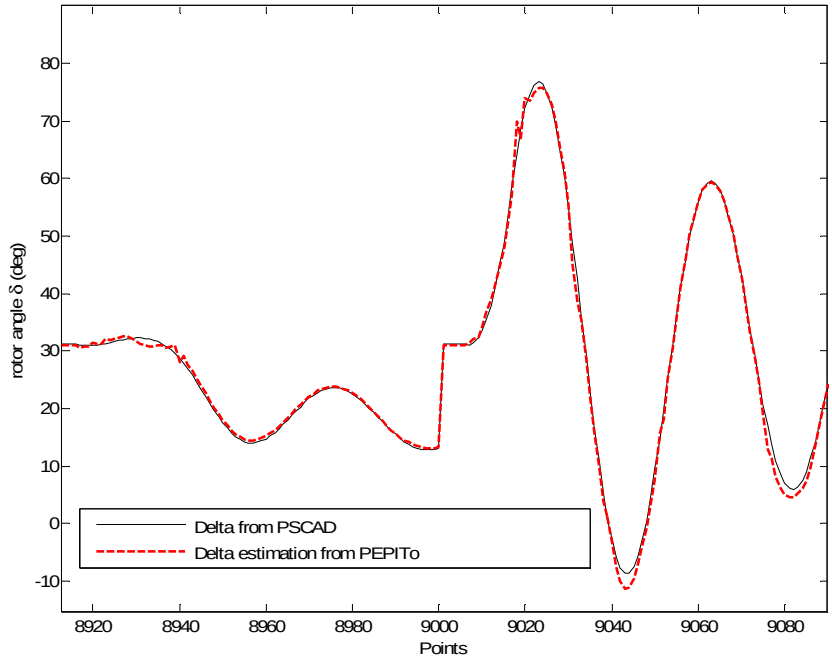


Figure 5.17 Rotor angle vs estimation for a single phase fault followed by an stable three phase fault test scenario.

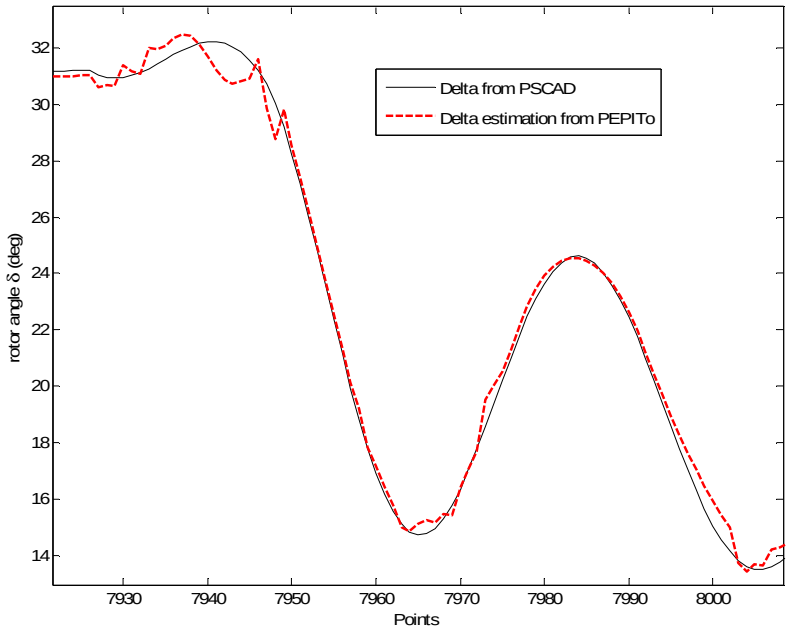


Figure 5.18 Rotor angle vs estimation for a stable followed by an unstable test scenario.

Figure 5.18 illustrates the estimation of rotor angle delta for a single phase fault, showing some small estimation errors at fault application and clearing instants.

Discussion

Our investigations have shown that direct estimation of the rotor angle with neural networks led to inappropriate results, essentially because the phasors computed by the PMU device are trimmed to the interval [-180°,180°] while the output quantity may become very large in the case of unstable scenarios.

Among the different possible approaches that could have been proposed to circumvent this difficulty, we have proposed to use a rectangular representation of the rotor angle leading to an estimation scheme were two neural networks are used, one to estimate the sinus of the rotor angle and the other one to estimate the cosinus of it. These quantities are smooth functions of the rotor angle, which itself varies smoothly over time, and from them it is possible to recover the rotor angle up to a multiple of 360 degrees.

The resulting scheme leads to robust rotor angle estimation both in the case of balanced and unbalanced fault conditions.

5.3.3 Estimation of ω for the MIS

This approach used in this thesis can be extended in a straightforward way to the estimation of rotor angles by substituting $\omega(t)$ for $\delta(t)$, in the data preparation step.

For estimating the value of rotor speed, the MLP configuration used had the form 60 – X–X- ω, were X denotes the number of hidden neurons (in our case we played with three different numbers of configurations) The training set and testing set remain the same that for the case of rotor angle estimation.

Table 5.20 shows training and testing results obtained from this kind of MLPs, where Mu (μ) is the Mean and sigma (σ) is the standard deviation.

Table 5.20 Results using 60 – X – X- ω configuration (ω in p.u.)

Net configuration	Correlation factor	Error min.	Error max.	Mu (μ)	Sigma (σ)
60-5-5-ω	0.988	-47.72E-3	103.26E-3	25.88E-6	3.57E-3
60-8-8-ω	0.995	-82.57e-3	55.47e-3	21.66e-6	2.35e-3
60-10-10-ω	0.997	-32.74e-3	59.76e-3	14.55e-6	1.69e-3

Figure 5.19 further highlights the overall performances by a scatter-plot representing the true rotor speed and its estimation for all the test samples. It shows, the approximation if of very good quality.

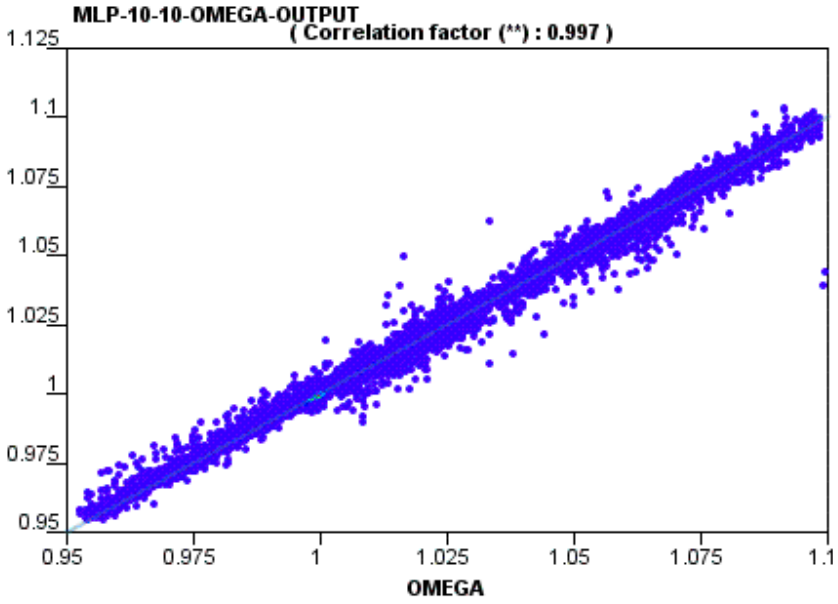


Figure 5.19 Rotor speed vs estimation for all test samples. MLP 60 -10-10-ω configuration

Figure 5.20 shows a test stable scenario the true rotor speed taken from simulation and the approximation using two different configuration of MLP, both with two hidden layers, and ones used 8 neuron units each and the second using 10 neuron units. The best approximation according to the Table 5.20 and seeing this picture, we find that NN with

10 neuron units had the best performance. Figure 5.21 illustrates an unstable test scenario also doing the comparison between this two kind of configuration.

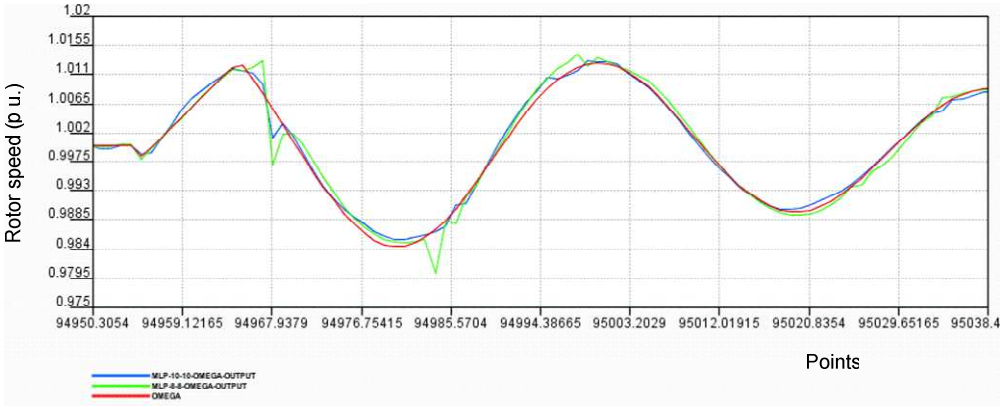


Figure 5.20 Comparison true vs. estimate rotor speed for different MLP configurations. Stable test scenario

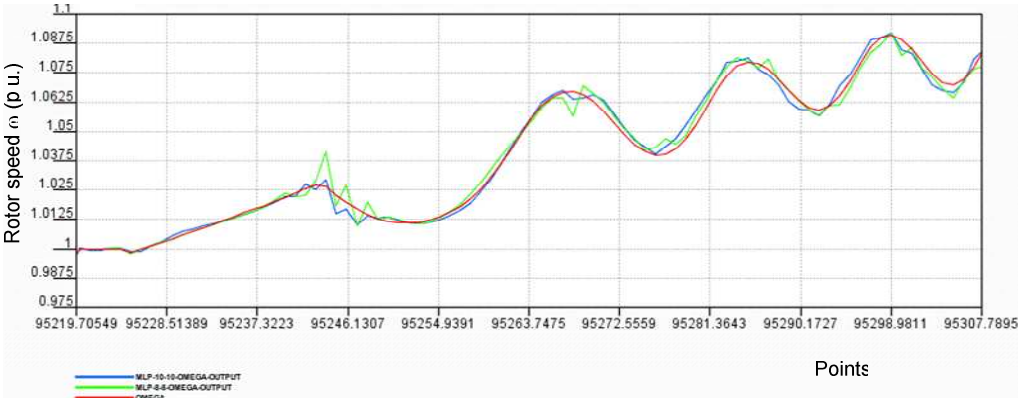


Figure 5.21 Comparison true vs. estimate rotor speed for different MLP configurations. Unstable scenario

Figure 5.22 shows a few test scenarios both the true rotor speed (solid line), taken from PSCAD simulations, and the approximation from MLP training.(dashed line) under stable conditions and Figure 5.23 shows the unstable cases.

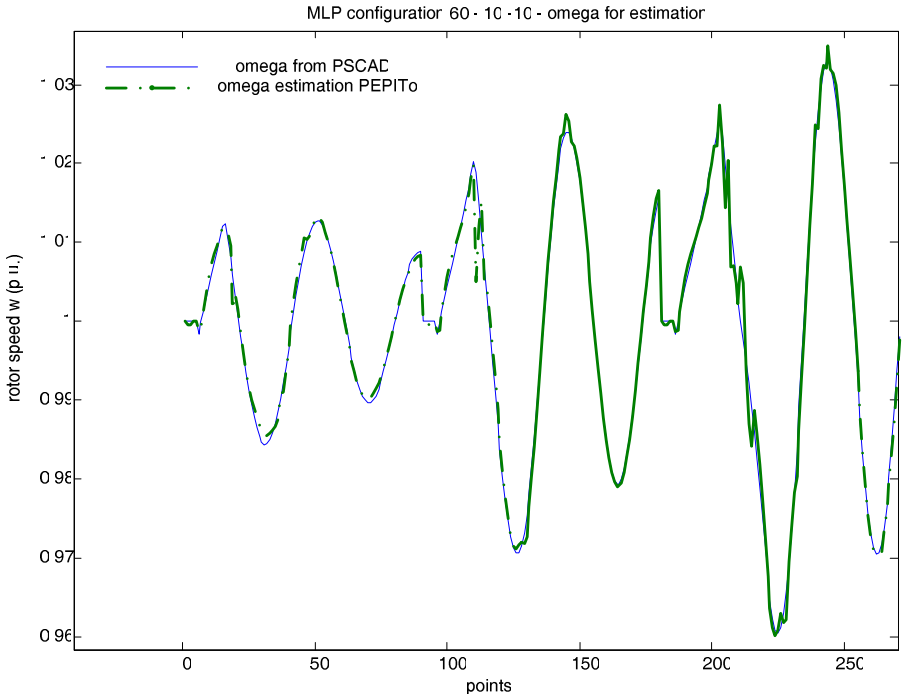


Figure 5.22 True vs. estimate rotor speed for MLP 60 -10-10-ω

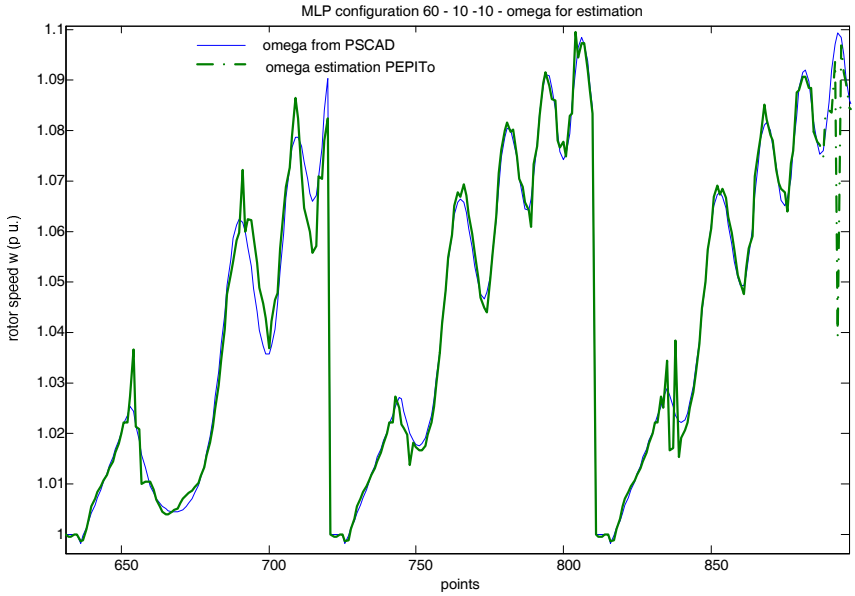


Figure 5.23 True vs. estimate rotor speed for MLP 60 -10-10-ω

Figure 5.24 and 5.26 shows for a few test scenarios following a single phase fault showing some small estimation errors at fault application and clearing instants. Figure 5.25 show two test scenarios representing a three-phase fault occurrence.

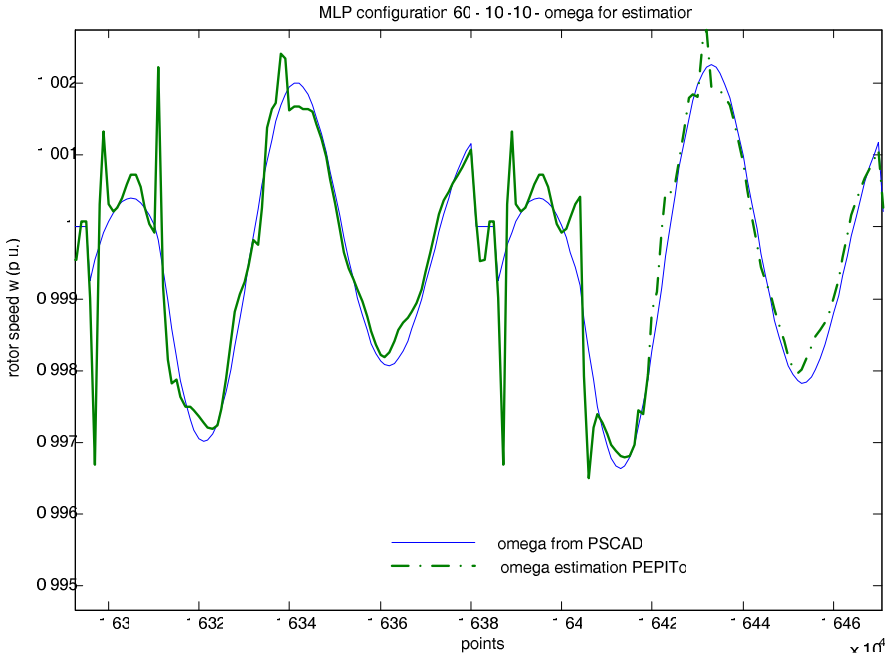


Figure 5.24 True vs. estimate rotor speed for MLP 60 -10-10-ω

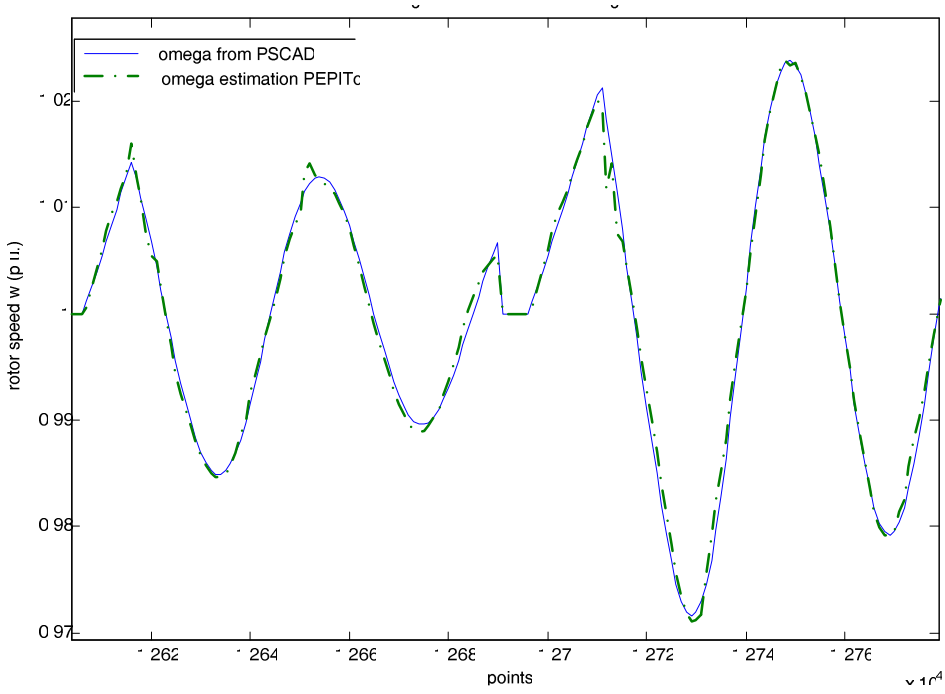


Figure 5.25 True vs. estimate rotor angle MLP 60 -10-10-ω

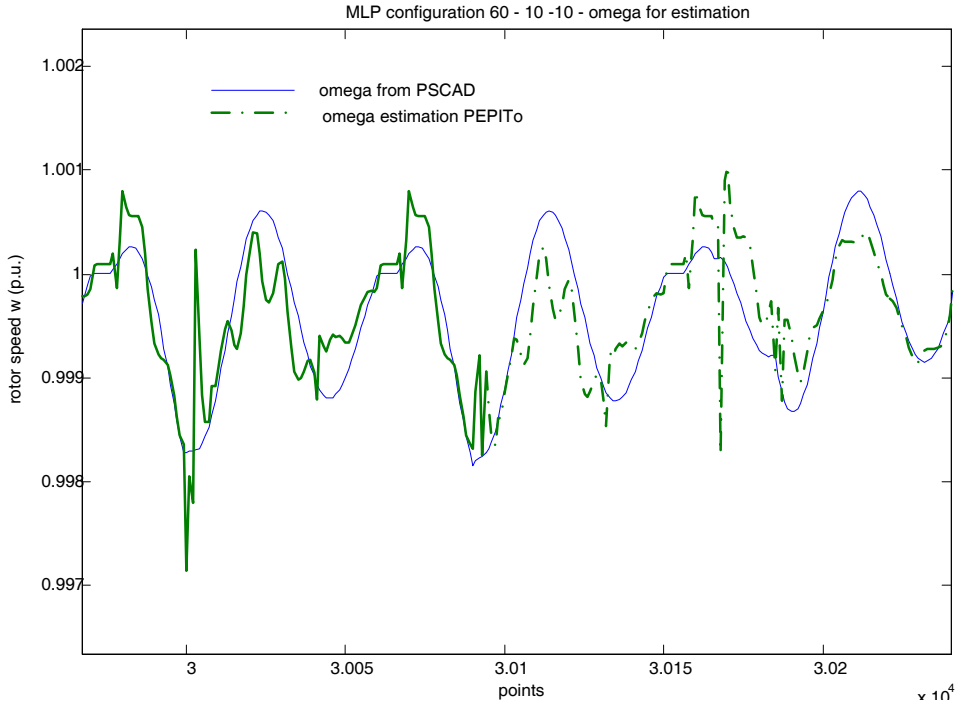


Figure 5.26 True vs. estimate rotor angle for MLP 60 -10-10-ω

Discussion

The previous result show that rotor angle estimation can be carried out directly from phasor measurement quantities in an already satisfactory way. Further investigations should be carried out in order to see whether using the estimated rotor angles as additional input to the neural network could improve the speed estimation. Also, some postprocessing would be needed in order to filter out errors appearing during the fault application and clearing.

5.3.4 Prediction of δ for the MIS

We extend the approach of section 5.3.2 in a straightforward way to the prediction of rotor angles by substituting $\delta(t)$ for $\delta(t+\Delta t)$, in the data preparation step. In our trials we used a value of Δt equal to 16.66 ms.

For this case, the set of scenarios was split into 1052 training scenarios and 355 testing scenarios, yielding a training set of 94680 samples and a testing set of 31950 samples.

Except for a small number of large errors, due to the fact that the signal discontinuity at 180 ° does appear at two different (but successive) time steps for the true signal and its prediction., the prediction of rotor angle delta provides a minimum and maximum errors

of -29.90° and 45.78° respectively. The good quality of this prediction is shown on Figures 2.28 to 5.32.

Table 5.21 shows testing results obtained for two such MLPs, both with two hidden layers of 8 neurons each, and a linear output neuron. In these simulations we considerer three phase faults and single phase faults. On Figure 5.27 the points in the upper left and lower right corner correspond to 94 large instantaneous errors. If we not take in account this large errors, the correlation factor is 0.9968. Figure 5.28 we represent the true value of rotor angle (solid line) obtained from PSCAD simulations, dotted line represents the value computed from the 60 – 8 – 8-sin$(\hat{\delta})$ and 60 – 8 – 8-cos$(\hat{\delta})$ trained for 100 cycles.

Table 5.21 Results using 60 – 8 – 8-$\hat{\delta}$ configuration ($\hat{\delta}$ in degrees.)

Net configuration	Correlation factor	Error min(deg	Error Max(deg)	Mu(μ)	Sigma(σ)	cycles
60-8-8-cos($\hat{\delta}$)	0.998	-0.677	0.401	663.01e-6	38.39e-3	100
60-8-8-sin($\hat{\delta}$)	0.9930	-0.600	0.514	3.68e-3	49.09e-3	100

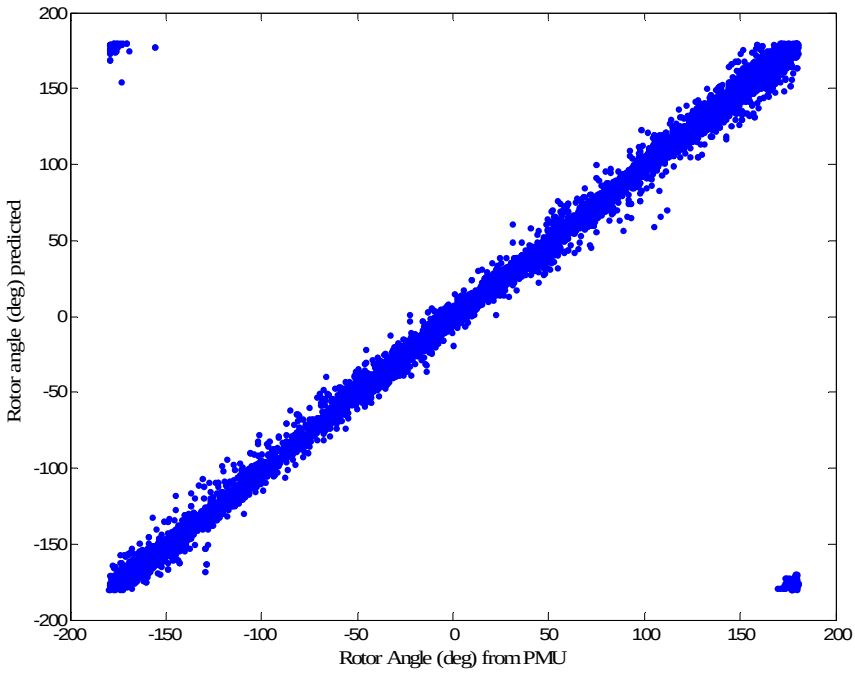


Figure 5.27 Rotor angle vs prediction for all 31950 test samples.

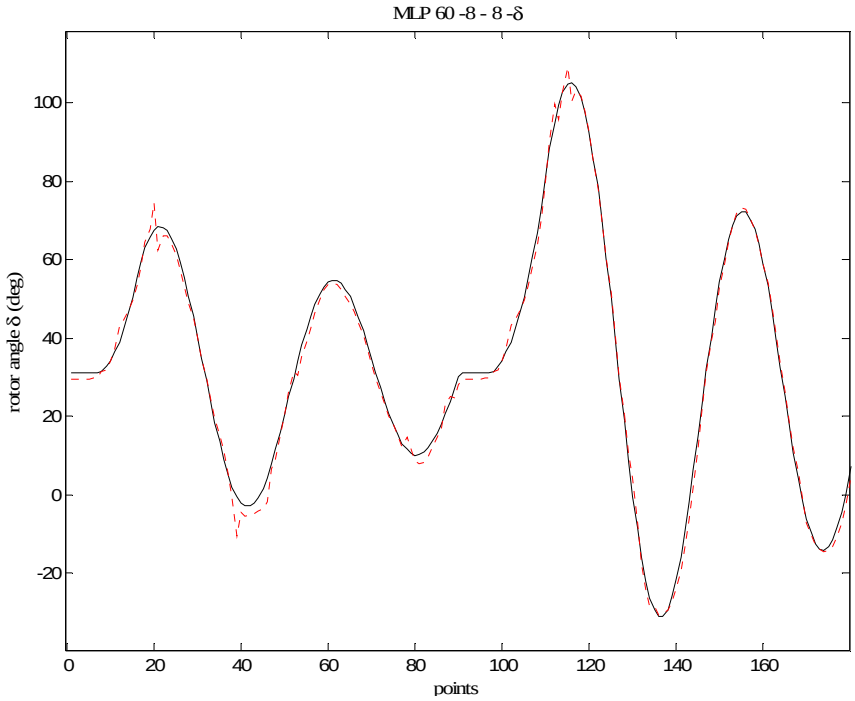


Figure 5.28 True vs. predicted rotor angle for MLP 60 -8-8- $\hat{\delta}$

Figure 5.29 shows two test scenarios form an unstable case, as the figures mentioned above, the solid line represents the true value of rotor angle, and the prediction recomputed from the two MLPs outputs in dotted line, both are trimmed to the interval [-180°,180°]. Figure 5.30 illustrates a few test scenarios that correspond to a single phase fault.

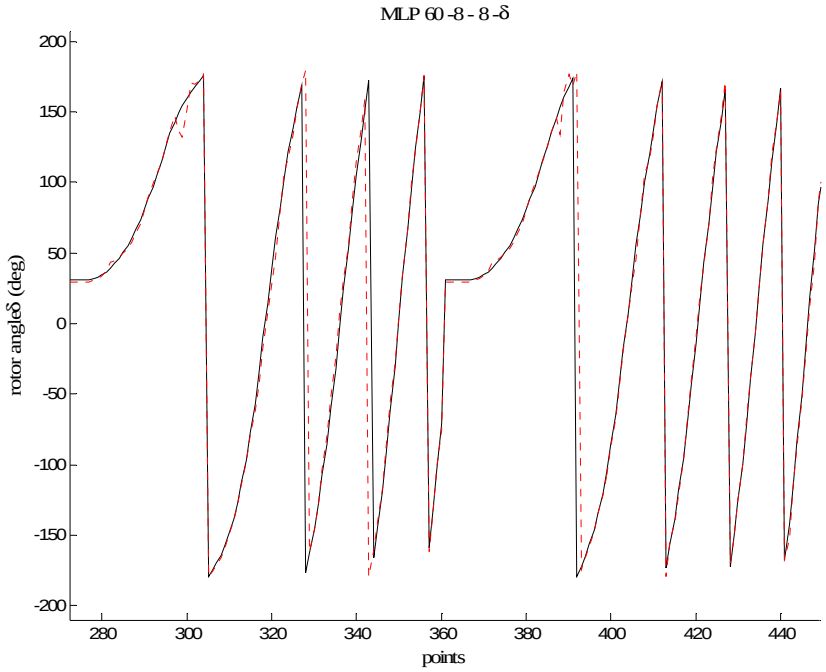


Figure 5.29 True vs. predicted rotor angle for MLP 60 -8-8- $\hat{\delta}$

Figures 5.31 and 5.32 suggest that the prediction of rotor angle is of very good quality.

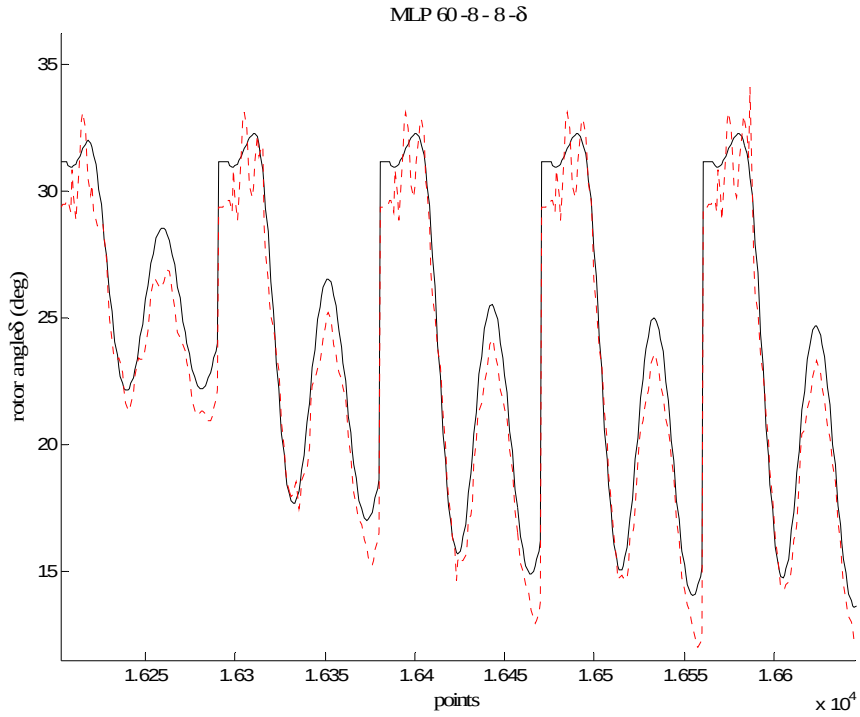


Figure 5.30 True vs. predicted rotor angle for MLP 60 -8-8- $\hat{\delta}$

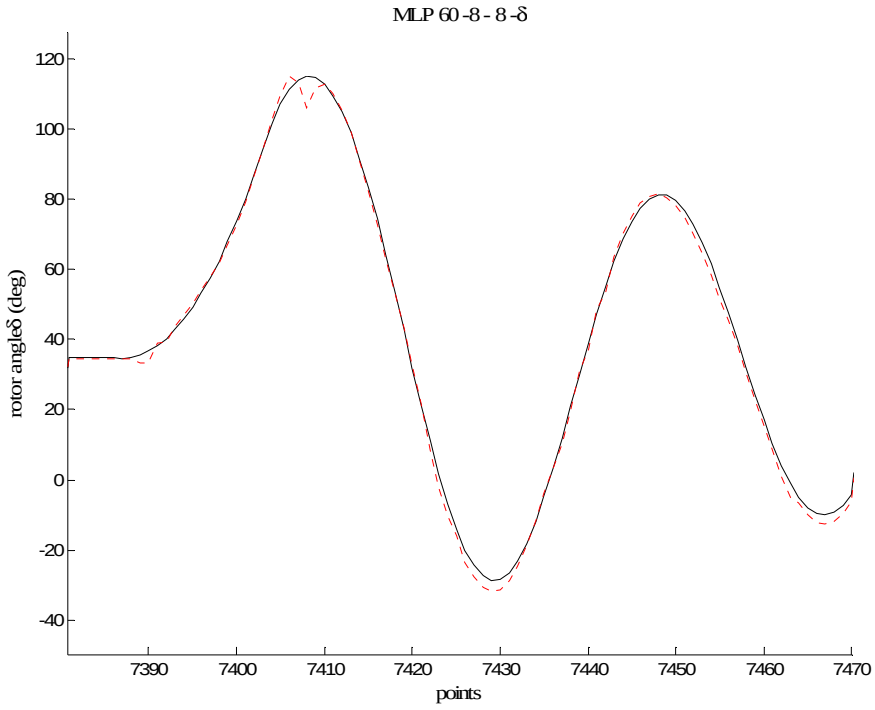


Figure 5.31 True vs. predicted rotor angle for MLP 60 -8-8- $\hat{\delta}$

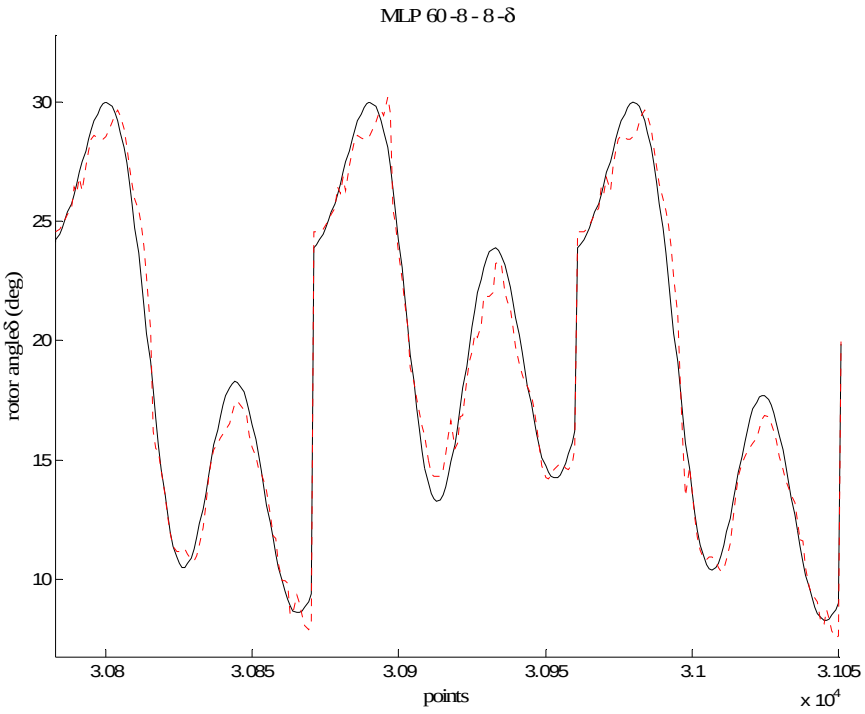


Figure 5.32 True vs. predicted rotor angle for MLP 60 -8-8- $\hat{\delta}$

5.3.5 Prediction of ω for the MIS

This approach used in this thesis can be extended in a straightforward way to the prediction of rotor angles by substituting $\omega(t)$ for $\omega(t+\Delta t)$, in the data preparation step. For this case, the set of scenarios was split into 1052 training scenarios and 355 testing scenarios, yielding a training set of 94680 samples and a testing set of 31950 samples, each one described by 90 instantaneous input values and one output(ω).

Table 5.22 shows testing results obtained for MLPs, with two hidden layers changing the number of hidden units, and a linear output neuron. In these simulations we considerer three phase faults and single phase faults. On figure 5.33 further highlights the overall performances by a scatter-plot representing the true rotor speed and its prediction for all the 31950 test samples. Figure 5.34 we represent the true value of rotor speed (solid line) obtained from PSCAD simulations, dotted line represents the value computed from the 60 – 15 – 5- ω trained for 100 cycles.

Table 5.22 Results using 60 – X – X- $\hat{\omega}$ configuration (ω in p.u.)

Net configuration	Correlation factor	Error min.	Error max.	Mu (μ)	Sigma (σ)
60-8-8- $\hat{\omega}$	0.993	-36.09E-3	66.90E-3	277.65E-6	2.65E-3
60-10-10- $\hat{\omega}$	0.997	-74.84e-3	53.70e-3	18.79e-6	1.87e-3
60-15-5- $\hat{\omega}$	0.997	-37.30e-3	26.67e-3	-2.19e-6	1.87e-3

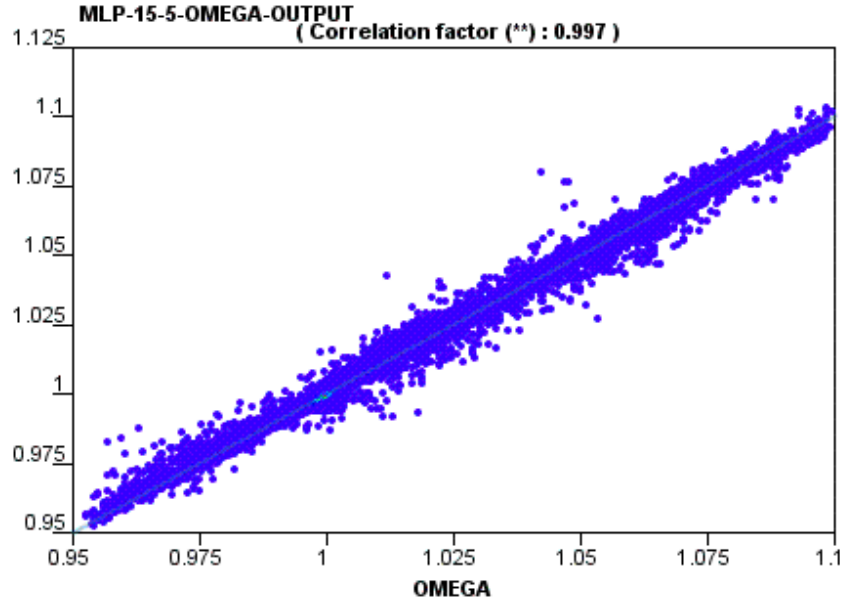


Figure 5.33 Rotor speed vs prediction for all 31950 test samples MLP 60 -15-5- $\hat{\omega}$

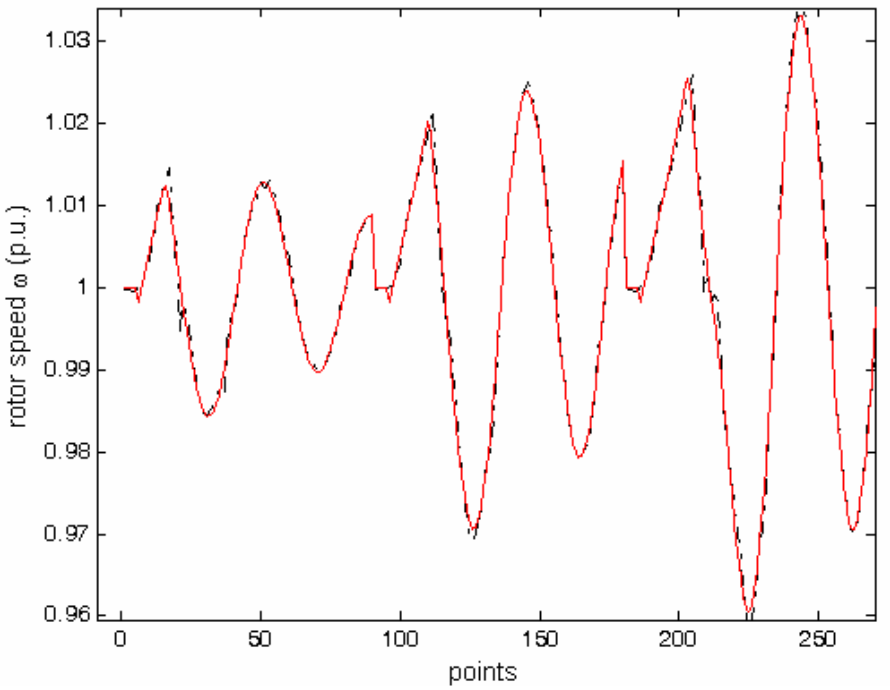


Figure 5.34 True vs. predicted rotor speed for MLP 60 -15-5- $\hat{\omega}$

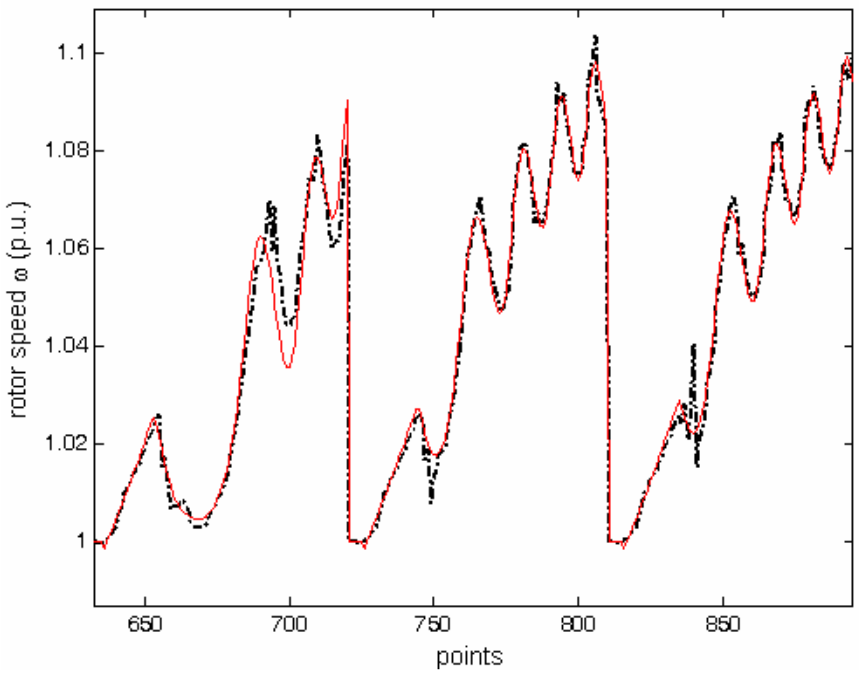


Figure 5.35 True vs. predicted rotor speed for MLP 60 -15-5- $\hat{\omega}$. Unstable test scenario

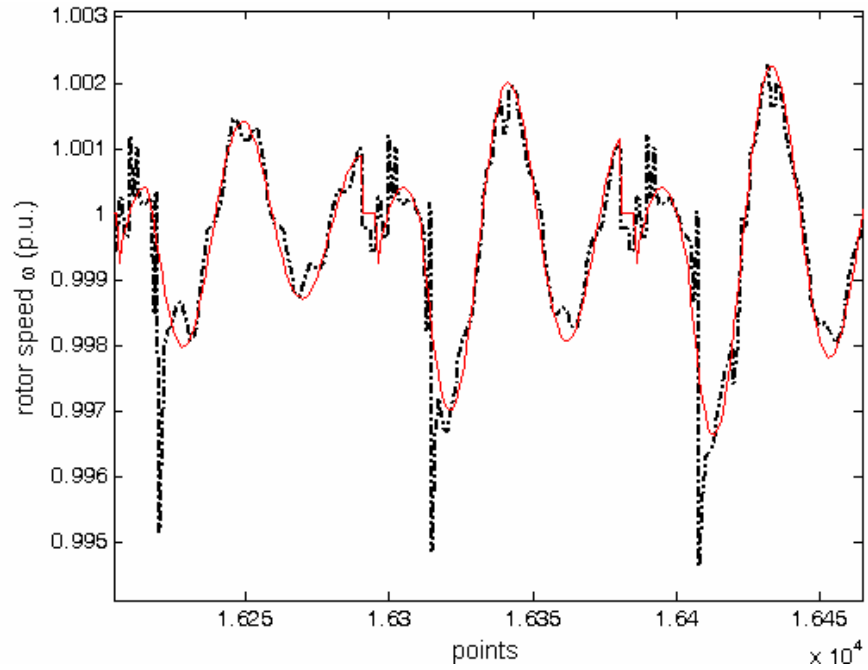


Figure 5.36 True vs. predicted for MLP 60 -15-5- $\hat{\omega}$. Single phase fault test scenario

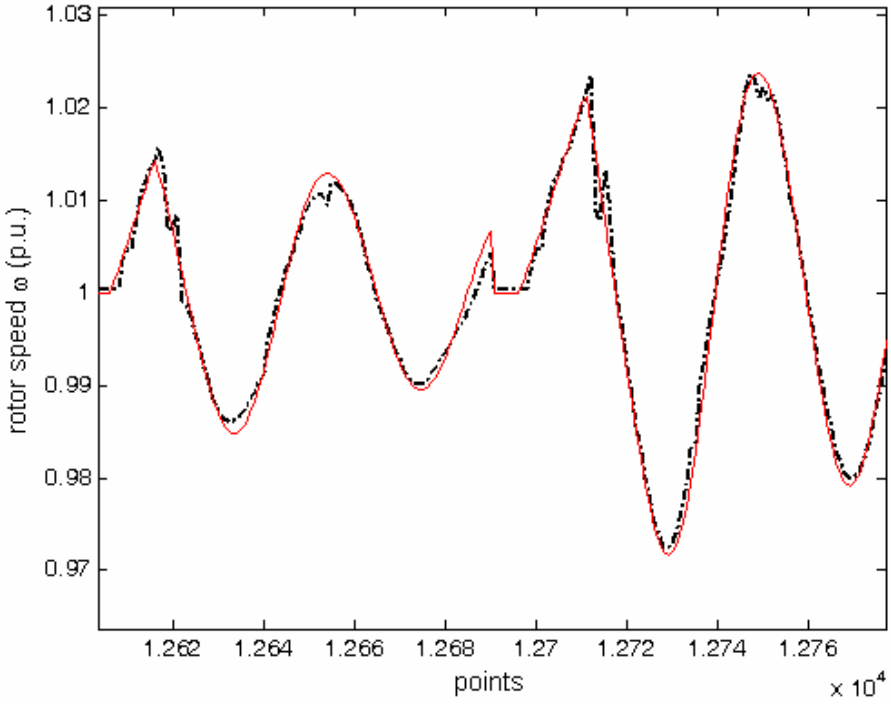


Figure 5.37 True vs. predicted rotor speed for MLP 60 -15-5- $\hat{\omega}$.Stable test scenario

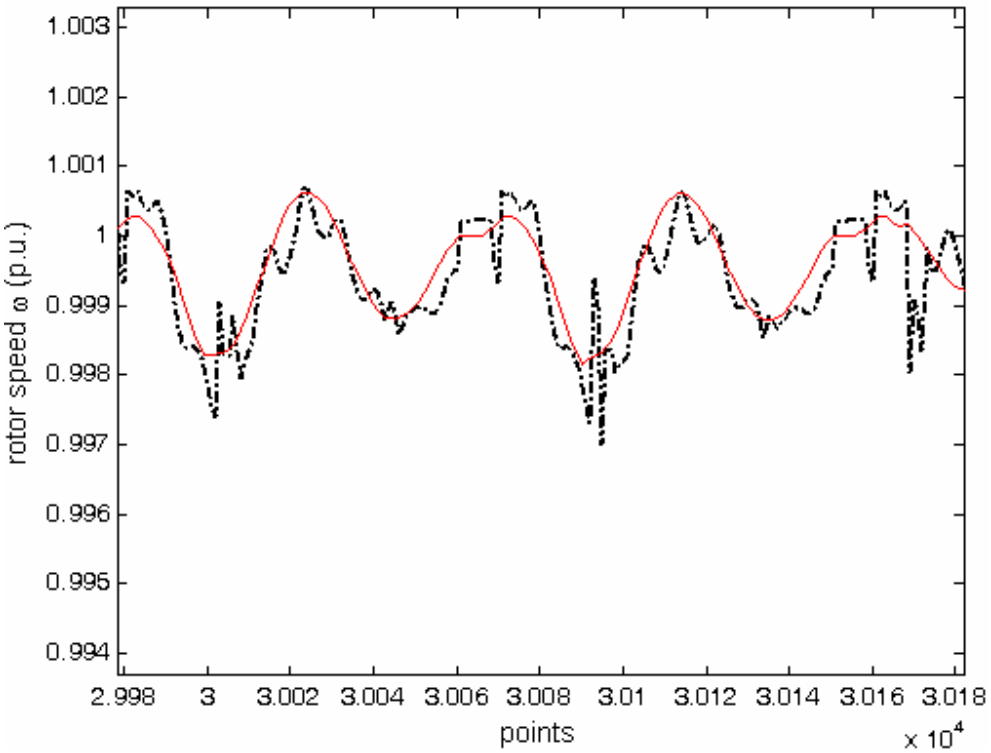


Figure 5.38 True vs. predicted rotor speed for MLP 60 -15-5- $\hat{\omega}$. Single phase fault case

5.3.6 Discussion

In this chapter we have provided our investigations using a detailed model of a part of the MIS which allowed to simulate the transient behaviour of a power plant subject the a large variety of balanced and unbalanced fault conditions, while extracting measurements of three-phase phasors by a module which simulates the actual PMU device.

We found that it is possible to obtain accurate estimation and prediction of rotor angles and speeds, provided that an appropriate coding of neural network outputs is used for the rotor angles (so-called rectangular representation).

Although our results would need to be further validated, by using a richer set of testing scenarios, and also by taking into account measurement errors and variations in settings of the power plant control loops, we believe that these conclusions will remain valid.

From a methodological point of view, although good results were obtained by using MLPs, it would also be interesting to assess the possibility of using other supervised learning methods, and possibly other sets of input variables.

CHAPTER 6 CONCLUSION AND FUTURE WORK

6.1 CONCLUSIONS

This thesis focused on the estimation and prediction of rotor angles and rotor speeds of synchronous machines using PMU measurements as inputs to artificial neural networks, in order to train them over the data sets and evaluate their performances over independent test sets.

We have chosen multi-layer perceptrons over others supervised learning methods because this kind of algorithm is known to provide a good generalization provided enough training data is available in comparison to the dimensionality of the input space. Ideed, MLPs are smooth universal approximators, and we are working in a context where the input space is of relatively low dimensionality while the target input-output mapping is rather smooth. Another practically very important characteristic that has motivated the use of MLPs is that, in spite of a rather slow off-line training algorithm, they can be used in a very efficient manner to estimate or predict the quantities of interest in real-time. This very high computational efficiency is particularly important in the context of transient stability monitoring where it is necessary to reduce as much as possible delays due to complex data processing.

In order to familiarize with the problem, we started our investigations by using a simple test case, that consisted of an OMIB system. In this context, we used rather small data sets of pairs of input-output data using a simplified dynamic model of the simulated power system, implemented in MATLAB-SIMULINK. This model used a single phase model of the system, assuming balanced conditions, and considered only the electromechanical dynamics modeled in transient stability studies. This first set of investigations used also the MATLAB neural network module and did not consider the prediction of angles and speeds ahead in time. It yielded however promising results, specially in comparison with an analytical approach exploiting the classical model to

compute rotor quantities from electrical phasors. The results of this preliminary study were published in a conference paper [DAGW03].

In a second stage, we have applied the same approach while considering a much more detailed and realistic test system. In particular, to take into account the behavior of a multi-machine system, we used a simplified model of the Mexican interconnected system. This system consists basically of a power plant with five synchronous generators represented in detail, that are interconnected by three lines to the rest of system. The rest of the system is represented by two large synchronous machines in order to represent with more fidelity the behavior of a real world system. In the simulations carried out on this system we have also used a detailed 'electromagnetic-transient and three-phase' dynamical model and a more exact representation of the PMU device (modeling the FFT used to compute phasors from instantaneous measurements). The higher complexity of this problem obliged us to use a larger database of simulation scenarios to train the neural networks. Hence we used the professional PEPITe [Pep04] data mining software in most of the experiments on this system. In this study we found out that it was preferable to use rectangular coordinates to represent the rotor angle targeted by the MLP [DAGE^{+}06]. Further investigations have been reported in this thesis showing also quite promising results concerning the possibility of predicting rotor angles and speed ahead of time. These latter results have not yet been published.

6.2 FUTURE WORK

There are many possible directions for future work.

The most direct continuation would be to enhance the validation of the proposed approach by considering other test power systems (in particular focusing on hydro-plants), a more accurate representation of the PMU device (including measurement noise, distortions to measurement transformers, and time jitter due to the limited accuracy of GPS signals) and more extensive simulations on a broader range of system conditions.

A second direction of future research would consist in applying other supervised learning methods to our datasets. In particular, we believe that it is worth comparing more systematically the compromise between precision and computational requirements of

multilayer perceptrons with other methods recently proposed in the automatic learning literature (ensemble methods, kernel-based methods, support-vector machines etc.).

Also, since the long-term goal of our research is to enable real-time emergency control for limiting the risk of loss of synchronism, we believe that it would be of interest to investigate more deeply how automatic learning could be used in order to determine directly the appropriate control actions. In this context, it would be particularly interesting to investigate the possibility of determining these control actions by using only locally acquired PMU measurements.

Finally, the approach investigated in this thesis should certainly be considered as a candidate for other power systems instability monitoring questions, such as voltage instability and negatively damped oscillations.

Bibliography

BIBLIOGRAPHY

[ABDA+96] E. Abu-Al-Feilat, M. Bettayeb, H. Al-Duwaish, M. Abido and A. Mantawy, A neural network-based approach for on-line dynamic stability assessment using synchronizing and damping torque, Electric Power System Research, Vol. 39, pp. 103-110, 1995.

[AF93] P. M. Anderson and A. A. Fouad, Power System Control and Stability, IEEE Press, 1993.

[ALA02] O. Anaya-Lara and E. Acha, Modelling and Analysis of Custom Power Systems by PSCAD/EMTDC, IEEE Transaction on Power Delivery, Vol. 17, No. 1, pp. 266-272, 2002.

[APSZ95] D. W. Aukland, I. E. D. Pickup, R. Shuttleworth and C. Zhou, Artificial neural network-based method for transient response prediction, IEE Proceedings Gen. Trans. Distribution, Vol. 142, No. 3, pp. 323-329,1995.

[AR96] F. Aboytes and R. Ramirez, Transient Stability Assessment in Longitudinal Power Systems using Artificial Neural Networks, IEEE Transactions on Power Systems, Vol. 11, No. 4, pp. 2003-2010, 1996.

[AW01] J. Arrillaga and N. A. Watson, Computer Modelling of Electrical Power Systems, John Wiley & Sons, 2001.

[BBBB92] N. Balu, T. Bertarnd et al, On-Line Power System Security, Proceeding of the IEEE, Vol. 80, No. 2, pp.262-280, 1992.

[BFOS84] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone, Classification and regression trees, Wadsworth Int. 1984

[BG04] A. G. Bahbah and A. G. Girgis, New Method for Generator's Angle and Angular Velocities Prediction for Transient Stability Assessment of Multimachine Power Systems Using Recurrent Artificial Neural Networks, IEEE Transactions on Power Systems, Vol. 19, No. 2, pp.1015-1022, 2004

[BG99] A.G. Bahbah and A. A. Girgis, Input feature selection for real-time transient stability assessment for artificial neural networks (ANN) using ANN sensitivy analysis, IN Proceedings of the 21 st IEEE International Conference of Power Industry Computer applications, PICA '99, pp. 295-300, 1999.

[BK99] E. Bauer and R. Kohavi, An empirical comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants, Machine Learning, Vol. 36, pp. 105-139, 1999.

[BPA99] Bonneville Power Administration, WAMS Final Report, BPA, 1999.

[Bre96] L. Breiman, Bagging Predictors, Machine Learning, Vol. 24, pp. 123-140, 1996.

Bibliography

[ChZ97] T. J. Cholewo and J. M. Zurada, Sequential Network Construction for Time Series Prediction, In Proceedings of the IEEE International Joint Conference on Neural Networks, Houston, USA, pp. 2034-2039, 1997.

[CKP98] E. H. Camm, A. Keyhani and S. Pilluta, Developed of a neural network model for rotor angle estimation, Engineering Intelligent Systems, Vol. 6, No. 1, pp. 13-18, 1998.

[CLG00] R. Caruana, S. Lawrence and L. Giles, Overfitting in Neural Nets: Backpropagation, Conjugate Gradient, and Early Stopping, Neural Information Processing Systems, Denver, Colorado, 2000.

[CY96] Z. Chi and H. Yan. ID3-derived fuzzy-logic and optimized defuzzication for handwritten numerical recognition. IEEE Transactions on Fuzzy Systems, 4(1):24-31, February 1996.

[DAGE+06] A. Del Angel, , P. Geurts, D. Ernst ,M. Glavic and L. Wehenkel, Estimation of rotor angles of synchronous machines using artificial neural networks and local PMU-based quantities. Neurocomputing, available on line 27 February 2007.

[DAGW03] A. Del Angel, M. Glavic, L. Wehenkel, Using Artificial Neural Networks to Estimate Rotor Angles and Speeds from Phasor Measurements, In Porceedings of Intelligent System Applications in Power. ISAP 2003, Lemnos, Greece, Paper ISAP03/017, 2003.

[DEAS+04] I. C. Decker, J. G. Ehrensperger, M. N. Agostini, A.S. De Silva, A.L. Bettiol, S.L. Zimath, Synchronized Phasor Measurement System: Development and Applications, In proceeding of IX Symposium of Specialist in Electrical Operational and Expansion Planning (SEPOPE), paper SP-086, Rio de Janeiro, Brazil, 2004.

[DW02] A. Diu, L. Wehenkel, EXaMINE – Experimentation of a Monitoring and Control System for Managing Vulnerabilities of the European Infrastructure for electric Power Systems, In Proceedings of IEEE/PES Summer Meeting, Chicago, USA, 2002.

[ELO00] ELORSK: Wide Area Measurements of Power System Dynamics – The North American WAMS Project and its applicability to the Nordic Countries, Lund, Sweden, 2000.

[ElS96] M.A. El-Sharkawi, Neural Networks' power, Potentials IEEE, Vol. 15, No. 5, pp.12-15, 1996.

[EP00] D. Ernst and M. Pavella, Closed-Loop Transient Stability Emergency Control, IEEE Power engineering Society Winter Meeting, Vol. 1, pp. 58-62,2000.

[Esh94] M. A. El-Sharkawi, What role can neural network play in power system engineering, IEEE Power Engineering Review, Vol. 14, No. 2, pp. 14-16,

Bibliography

1994.

[ESN96] M.A. El-Sharkawi and D. Niebur, Dynamic Security Assessment of Power Systems using Back Error Propagation Artificial Neural Networks, IEEE Power Engineering Society, 1996

[FS99] Y. Freund and R.E. Schapire, A short introduction to Boosting, Journal of Japanese Society for Artificial Intelligence, Vol. 14, No. 5, pp. 771-780, 1999.

[GBD92] S. Geman, E. Bienenstock, and R. Dousant. Neural Networks and the bias/variance dilemma. Neural Computation, 4:1-58, 1992.

[GDKI01] A. M. Gole, P. Demchenko, D. Krell and G.D. Irwin, Integrating Electromagnetic Transient Simulation with other Design Tools, In International Conference on Power Systems Transients, (IPST'2001), Rio de Janeiro, Brazil, 2001.

[Geu02] P. Geurts, *Contributions to Decision Tree Induction: Bias/Variance tradeoff and time series classification*. PhD thesis, University of Liege, Dept. of Electrical Engineering & Computer Science, Belgium, May 2002.

[Geu03] P. Geurts, Extremely randomized trees, Technical report, University of Liege, Department of Electrical Engineering and Computer Science, 2003.

[GEW06] P. Geurts, D. Ernst and L. Wehenkel, Extremely randomized trees, Machine Learning, Vol. 63, No. 1, pp. 3- 42, 2006.

[Hay94] S. Haykin, *Neural Networks, A comprehensive Foundatio*n, IEEE press, New York, 1994

[HHN05] K.E. Holbert, G. T. Heydt and H. Ni, Use of satellite technologies for Power System Measurements, Command and Control, Proceedings of the IEEE, Vol. 93, No. 5, pp. 947-955, 2005

[HKP90] J. Hertz, A. Krogh and R. G. Palmer, Introduction to the Theory of Neural Computation, Adison-Wesley Publishing Company, 1990.

[HM94] M. T. Hagan and M.B. Menhaj, Training Feedforward Networks with the Marquardt algorithm, IEEE Transaction on Neural Networks, Vol. 5 no. 6, pp. 989-993, 1994.

[Hyd00] Hydro-Quebec TEQSIM International, Power System Blockset for use with Simulink, User's Guide version 2.0, The Matlab Works Inc., 2000.

[IEEE/CIGRE04] IEEE/CIGRE Joint Task Force on Stability Terms and Definition, Definition and classification of Power System Stability, IEEE Transaction on Power Systems, Vol. 19, No. 2, pp. 1387-1401, may 2004.

[IEEE04] Task Force on Probalistic Aspects of reliability Criteria IEEE, Probabilistic Security assessment for power systems operation, Power Engineering Society,

Bibliography

General meeting, Vol. 1, pp. 212-220, 2004.

[IEEE95] IEEE 1344, Standard for Synchrophasors for Power Systems, New York, 1995

[JMM96] A. K. Jain , J. Mao and K.M. Mohiuddin, Artificial Neural Networks: A Tutorial, Computer, Vol. 29, No. 3, pp.31-44, 1996.

[KG02] I. Kamwa, R. Grondin, PMU configuration for system dynamic performance measurement in large multi-area Power Systems, IEEE Transaction on Power Systems, vol. 17, pp. 385-394, 2002

[KGH01] I. Kamwa, R. Grondin, Y. Hebert, Wide Area Measurement Based Stabilizing Control of Large Power Systems – A decentralized/Hierarchical Approach, IEEE Transactions on Power Systems, vol. 16, pp. 136-153, 2001
.

[KGL01] I. Kamwa, R. Grondin, L. Loud, Time-varing Contingency Screening for Dynamic Security Assessment Using Intelligent-Systems Techniques, IEEE Transactions on Power Systems, Vol. 16, No. 3, pp. 526-536, 2001.

[KH00] E.S. Karapidakis nad N. D. Hatziargyriou, Application of Artificial Neural Networks for Security Assessment of Medium Size Power Systems, 10th Mediterranean Electrotechnical Conference, MELECON 2000, vol. 3, pp. 1189-1192, 2000.

[Kim64] E. W. Kimbark, Power System Stability, Volume I Elements of Stability Calculations, John Wiley and Sons, 1964.

[KK02] G.G. Karady and M. A. Mohamed, Improving transient stability using generator tripping based on tracking rotor-angle and active power, Power Engineering Society Summer Meeting, Vol. 3, pp. 1576-1581, 2002.

[Kun04] P. Kundur, (convener). "Definition and Classification of Power System Stability." IEEE Transactions on Power Systems Vol. 19 ,No. 2: 1387-1401, 2004.

[Kun94] P. Kundur. Power System Stability and Control. McGraw Hill, 1994.

[KWS94] P.C. Krause, O.Wasynczuk and S. D. Sudhoff. Analysis of Electric Machinery.IEEE press, 1994.

[LSTW99] C. H. Liu, M. Ch. Su, S-S. Tsay and Y-J. Wang, Application of a novel fuzzy Neural Network to real-time transient stability swings prediction based on synchronized phasor measurement, IEEE Trans. on Power Systems, vol. 14, No. 2, pp. 685-692, 1999.

[LT00] C. W. Liu ans J.S. Thorp, New Methods for Computing Power System Dynamic Response for Real-Time Transient Stability Prediction, IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications, Vol. 47, No. 3, pp. 324-337, 2000.

Bibliography

[LT95] C-W. Liu and J. Thorp, Application of synchronized Phasor measurements to real-time transient stability prediction, IEE Proceedings Generation, Trans. and Distribution, Vol. 142, No. 4, pp. 355-360, 1995.

[LTWS99] Ch. Liu, S. Tsay, Y. Wang and M. Su, Neuro-fuzzy approach to real-time transient stability prediction based on synchronized Phasor measurements, Electric Power Systems Research, Vol. 49, pp. 123-127, 1999.

[Man03] Manitoba HVDC Research Centre, PSCAD/EMTDC Power Systems Computed Aided Design, User's guide on the use of PSCAD Simulation Software Tutorial, 2003

[Man03a] Manitoba HVDC Research Centre, EMTDC Transient Analysis for PSCAD Power System Simulation, User's guide, A comprehensive resource for EMTDC, 2003

[MCK95] J. D. McCalley and B. A. Krause, Rapid transmission capacity margin determination for dynamic security assessment using artificial neural netwoks, Electric Power System Research,, Vol. 34, pp. 37-45, 1995.

[MCZ95] A. Malinowski, T. J. Cholewo and J. M Zurada, Capabilities and Limitations of Feedforward Neural Networks with Multilevel Neurons, In Proceedings of the IEEE International Symposium on Circuits and Systems, vo. 1, pp. 131-134, Seattle, USA, 1995.

[MEH00] J. A. Momoh, M. E. El-Havary, Electric Systems, Dynamics, and Stability with Artificial Intelligence Applications, 2000

[Mit97] T. M. Mitchell, Machine Learning, WCB/McGraw-Hill, 1997.

[Nat97] M. A. Natick, Neural Network Toolbox for use with Simulink user's guide, The MathWorks Inc. 1997

[Ola04] A. C. Olaru, Contributions to Automatic Learning: Soft Decision Tree Induction, PhD thesis, University of Liege, Dept. of Electrical Engineering & Computer Science, Belgium, 2004.

[Pai81] M. A. Pai, Electric Power System Stability, Vol. 3 Analysis by the Direct Method of Lyapunov, North-Holland Publishing Company, 1981.

[Pep04] PEPITo User's guide, 2004. www.pepite.be

[PERV00] M. Pavella, D. Ernst and D. Ruiz-Vega. Transient Stability of Power Systems: A UInified Approach to Assessment and Control. Kluwer Academic Publishers. 2000.

[Pha93] A.G. Phadke, Synchronized Phasor Measurements in Power systems, IEEE Computer Applications in Power, No. 6, pp. 10-15, 1993.

Bibliography

[PK96] Y. M. Park and W. Kim, Discrete-time adaptive sliding mode power system stabilizer with only input/output measurements, Electrical Power & Energy Systems, Vol. 18, No. 8, pp. 509-517, 1996.

[PM94] M. Pavella and P. G. Murthy, Transient Stability of Power Systems, John Wiley & Sons, 1994.

[PZ94] T. Parisini and R. Zoppoli, Neural Networks for Feedback Feedforward Nonlinear Control Systems, IEEE Transactions on Neural Networks, Vol. 5, No. 3, pp.436-449, 1994.

[RA05] M. A. Razi and K. Athappilly, A comparative predictive analysis of neural networks (NNs), nonlinear regression and classification and regression tree(CART) models, Expert Systems with Applications, vol. , No. , pp. 1-10, 2005.

[RLLM+95] S. Rovnyak, C-W. Liu, J. Lu, W. Ma and J. Thorp, Predicting future behavior of transient events Rapidly Enough to evaluate remedial control options in real time, IEEE Transaction on Power Systems, vol. 10, pp. 1195-1203, 1995.

[Sap90] G. SAPORTA, Probabilités analyse des données et statistique, Ed. Tecnip, 1990.

[SC02] K.S. Swarup, P.B. Corthis, ANN Approach Assess System Security, IEEE Computer Applications in Power, Vol. 15, No. 3, pp. 32-38,2002.

[SLT99] M. C. Su, C. W. Liu and S.S. Tsay, Neural-Network_based Fuzzy Model and its Application to Transient Stability Prediction in Power Systems, IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and reviews, Vol. 29, No. 1, pp. 149-157, 1999.

[SP89] D.J. Sobajic and Y. H. Pao, Artificial Neural-Net Based Dynamic Security Assessment for Electrical Power Systems, IEEE Transactions on Power Systems, Vol. 4, No. 1, pp. 220-224,1989.

[SP98] P. W. Sauer and M. A. Pai, Power System Dynamics and Stability, Prentice-Hall, 1998.

[SVSH89] R. P. Schulz, L.S. VanSlyck and S. H. Horowitz, Applications of Fast Phasor Measurements on Utility Systems, Power Industry Application Conference PICA'89, pp. 49-55, 1989.

[Tay00] C. Taylor(Convener) CIGRE Task Force 38.02.17, Advanced Angle Stability Controls, CIGRE Technical brochure ,No. 155, 2000.

[TEMW+05] C. W. Taylor, D. C. Erikson, K. E. Martin, Robert E. Wilson and V. Venkatasubramanian, WACS-Wide-Area Stability and Voltage Control System: R&D and Online Demonstration, Proceedings of the IEEE, Vol. 93,

Bibliography

No. 5, pp.892-906, 2005.

[VH02] G. K. Venayagamoorthy and R. G. Harley, Two separate Continually Online-Trained Neurocontrollers for Excitation and Turbine Control of a Turbogenerator, IEEE Transactions on Industry Applications, Vol. 38, No. 3, pp.887-893, 2002.

[VHW02] G. K. Venayagamoorthy, R. G. Harley and D.C. Wunsch, Comparison of Heuristic Dynamic Programming and Dual Heuristic Programming Adaptive Critics for Neurocontrol of a Turbogenerator, IEEE Transactions on Neural Networks, Vol. 13, No. 3, pp. 764-773, 2002.

[VHW03] G. K. Venayagamoorthy, R. G. Harley and. D.C. Wunsch, Dual Heuristic Programming Excitation Neurocontrol for Generators in a Multimachine Power System, IEEE Transaction on Industry Applications, Vo. 39, No. 2, pp. 382-394, 2003.

[Weh96] L. Wehenkel, Contingency severity assessment for voltage security using non-parametric regression techniques, IEEE Transaction on Power Systems, Vol. 11, No. 1, pp. 101-111,1996.

[Weh98] L. Wehenkel, *Automatic Learning techniques in power systems*, Kulwer Academic Publishers, Boston, 1998.
The effects of noise on the response of the system can be assessed by randomly perturbing the inputs (additive noise uniformly distributed).

[WLL05] Y. J. Wang, C. W. Liu and C. H. Liu, A PMU based special protection scheme: a case study of Taiwan power system, Electric Power Research, Vol. 27, No. , pp. 215-223, 2005.

[WREP05] L. Wehenkel, D. Ruiz-Vega, D. Ernst, M. Pavella, Preventive and Emergency Control of Power Systems, in Real-Time Stability in Power Systems – Techniques for Early Detection of the Risk of Blackout, S. Savulescu (ed.) Springer, pp. 199-232, 2005. [2]

[Zur92] J. M. Zurada, Introduction to Artificial Neural Systems, West Publishing Company, 1992.

Appendix A Modeling Power System Components

Generator model

A salient/round rotor Synchronous Machine fully developed model is available in PSCAD/EMTDC. The model is programmed in state variable form, using generalized machine theory.
The generalized machine model transforms the stator windings into equivalent commutador windings, using the dq0 transformation as follows:

$$\begin{bmatrix} Ud \\ Uq \\ Uo \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \cos(\theta-120°) & \cos(\theta-240°) \\ \sin(\theta) & \sin(\theta-120°) & \cos(\theta-240°) \\ 1/2 & 1/2 & 1/2 \end{bmatrix} \cdot \begin{bmatrix} Va \\ Vb \\ Vc \end{bmatrix} \tag{A.1}$$

The three-phase rotor winding may also be transformed into a two-phase equivalent winding, with additional windings added to each axis to fully represent that particular machine, as is shown in Figure A.1. Support subroutines are included in the machine model library for calculating the equivalent circuit parameters of a synchronous machine from commonly supplied data.
The d-axis equivalent circuit for the generalized machine is shown in figure A.2. Figure A.3 illustrates the flux paths associated with various d-axis inductances.

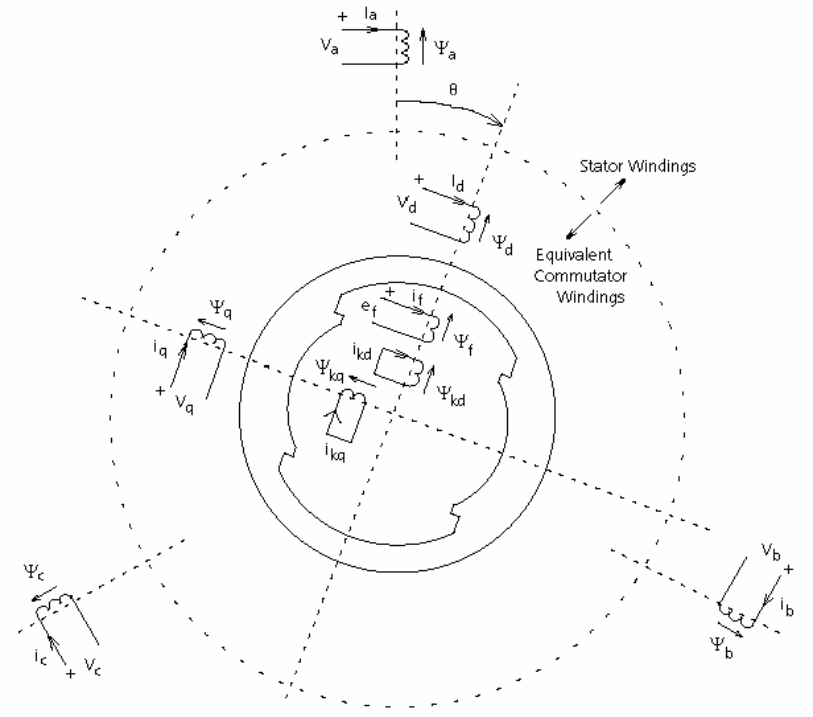


Figure A.1 Conceptual diagram of the three phase and dq windings

where,

k = Amortisseur windings
f = Field windings
a,b,c = Stator windings
d = Direct-axis (d-axis) windings
q = Quadrature-axis (q-axis) windings

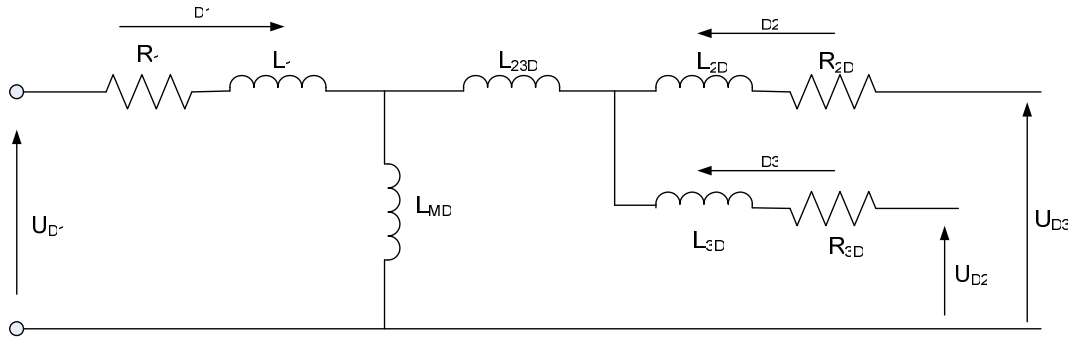


Figure A.2 D-axis Equivalent Circuit

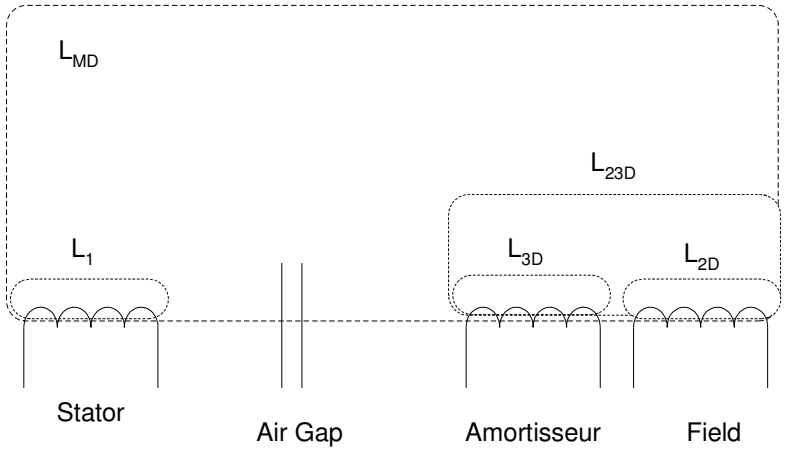


Figure A.3 Flux Paths Associated with Various d-axis Inductances

A second damper winding on the q-axis is included and it can also be used as a round rotor machine to model steam turbine generator.

Referring to Figures A.4 and A.5, the d-axis voltage U_{D2} and current I_{D2} are the field voltage and current respectively. The damper circuit consist of parameters L_{3D} and R_{3D} with U_{D3}=0. The additional inductance L_{23D} accounts for the mutual flux, which link only the damper and field windings and not the stator windings. the following equations can be derived:

$$\begin{bmatrix} U_{D1} - \omega \cdot \psi_q - R_1 \cdot i_{D1} \\ U_{D2} - R_{2D} \cdot i_{D2} \\ U_{D3} - R_{3D} \cdot i_{D3} \end{bmatrix} = L_D \cdot \frac{d}{dt} \begin{bmatrix} i_{D1} \\ i_{D2} \\ i_{D3} \end{bmatrix} \tag{A.2}$$

where,

$$L_D = \begin{bmatrix} L_{MD} + L_1 & L_{MD} & L_{MD} \\ L_{MD} & L_{MD} + L_{23D} + L_{2D} & L_{MD} + L_{23D} \\ L_{MD} & L_{MD} + L_{23D} & L_{MD} + L_{23D} + L_{3D} \end{bmatrix} \tag{A.3}$$

$$\psi_q = L_1 \cdot i_{Q1} + L_{MQ} \cdot \left(i_{Q1} + i_{Q2} + i_{Q3}\right) \tag{A.4}$$

$$\omega = \frac{d\theta}{dt} \tag{A.5}$$

Similar equations hold for the q-axis except the speed voltage term, $\upsilon \cdot \psi_d$, is positive, and:

$$\psi_d = L_1 \cdot i_{D1} + L_{MD} \cdot \left(i_{D1} + i_{D2} + i_{D3}\right) \tag{A.6}$$

Inversion of equation A.2 gives the standard state variable form $\dot{X} = AX + BU$ with state vector X consisting of the currents, and the input vector U, applied voltages. That is:

$$\frac{d}{dt}\begin{bmatrix} i_{D1} \\ i_{D2} \\ i_{D3} \end{bmatrix} = L_D{}^{-1} \cdot \begin{bmatrix} -\upsilon \cdot \psi_q - R_1 \cdot i_{D1} \\ -R_{2D} \cdot i_{D2} \\ -R_{3D} \cdot i_{D3} \end{bmatrix} + L_D{}^{-1} \cdot \begin{bmatrix} U_{D1} \\ U_{D2} \\ U_{D3} \end{bmatrix} \tag{A.7}$$

$$\frac{d}{dt}\begin{bmatrix} i_{Q1} \\ i_{Q2} \\ i_{Q3} \end{bmatrix} = L_Q{}^{-1} \cdot \begin{bmatrix} -\upsilon \cdot \psi_d - R_1 \cdot i_{Q1} \\ -R_{2D} \cdot i_{Q2} \\ -R_{3D} \cdot i_{Q3} \end{bmatrix} + L_Q{}^{-1} \cdot \begin{bmatrix} U_{Q1} \\ U_{Q2} \\ U_{Q3} \end{bmatrix} \tag{A.8}$$

In the above form, Equation A.7 and A.8 are particularly easy to integrate. The equations are solved using trapezoidal integration to obtain the currents. The torque equation is given as:

$$T = \psi_q \cdot i_{D1} - \psi_d \cdot i_{Q1} \tag{A.9}$$

The dq-axis model includes the transient and sub transient characteristics of the machine and the set of differential equations describing the generator dynamics is given by equations A.10 –A.15:

$$pE'_q = \frac{\left(E_f - E_q\right)}{T'_{d0}} = \frac{\left(Ef + \left(X_d - X'_d\right) I_d - E'_q\right)}{T'_{d0}} \tag{A.10}$$

$$pE'_d = \frac{-E_d}{T'_{q0}} = \frac{\left(-\left(X_q - X'_q\right) I_q - E'_d\right)}{T'_{q0}} \tag{A.11}$$

$$E_q'' - V_q = R_a I_q - X_d'' I_d \tag{A.12}$$

$$E_d'' - V_d = R_a I_d - X_q'' I_q \tag{A.13}$$

$$pE_q'' = \frac{\left(E_q' + \left(X_d' - X_d''\right) I_d - E_q''\right)}{T_{d0}''} \tag{A.14}$$

$$pE_d'' = \frac{\left(E_d' + \left(X_q' - X_q''\right) I_q - E_d''\right)}{T_{q0}''} \tag{A.15}$$

The final two state equations are provide by the rotor swing equations :

$$\frac{d\omega}{dt} = \frac{1}{M}\left(T_m - T_e - D\omega\right) \tag{A.16}$$

$$\frac{d\delta}{dt} = \omega_s(\omega - 1) \tag{A.17}$$

where,
Tm – turbine torque
Te – electrical torque
M – inertia constant
D – Damping constant
ω - Machine speed
ωs – Synchronous speed
δ - Rotor angle

Governor

The governor can be represent by IEEE type thermal governor model. In this case the approximate mechanical-hydraulic control (GOV1) was used. The schematic diagram for this device is given in figure A.4

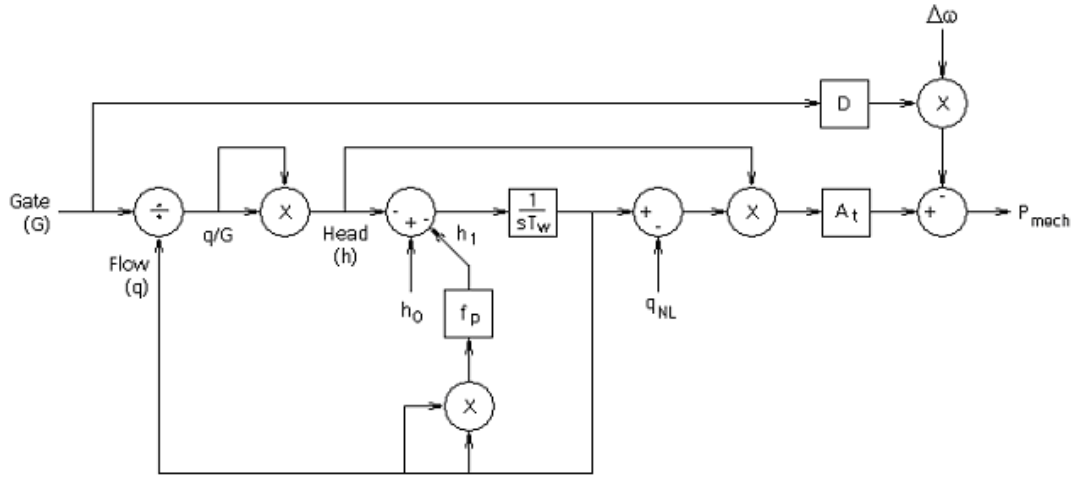


Figure A.4 hydro turbine model representation in PSCAD software.

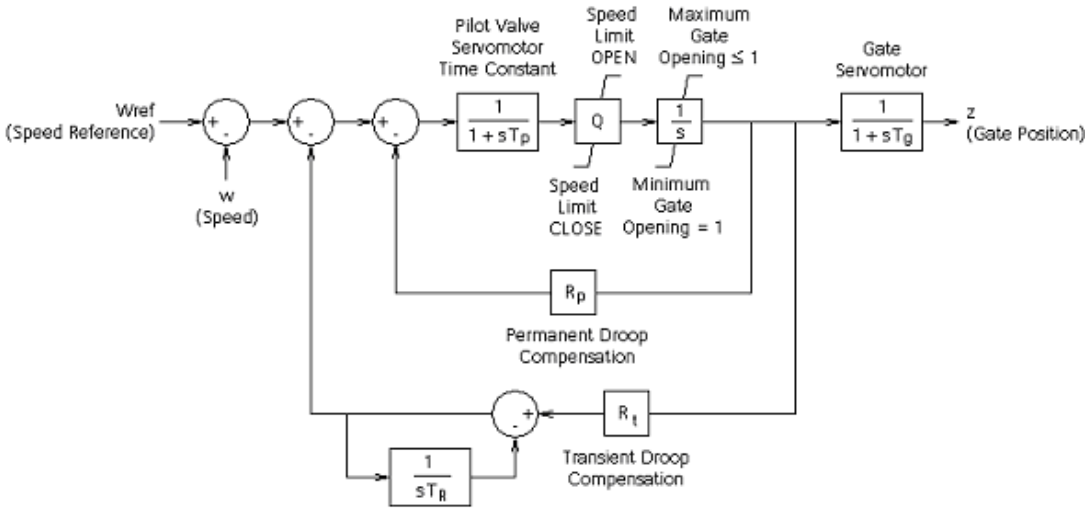


Figure A.5 hydro governor model type IEEE GOV 1

Appendix B Phasor Measurements Units (PMU)

Introduction.

Phasors are basic tools of ac circuit analysis, usually introduced as a means of representing steady state sinusoidal waveforms of fundamental power frequency.

Even when a power system is not quite in a steady state, phasor are often useful in describing the behaviour of the power system. When the power system is undergoing electromechanical oscillations during power swings, the waveforms of voltages and currents are not in steady state, and neither is the frequency of the power system at its nominal value. Under these conditions, as the variations of the voltages and currents are relatively slow, phasor may still be used to describe the performance of the network, the variations being treated as a series of steady state conditions. Recent developments in time synchronized techniques, coupled with the computer based measurement technique, have provided a novel opportunity to measure the phasors, and phase angle differences in real time.

Consider the steady state waveform of a nominal power frequency signal as shown in Figure B.1. Starting to observe the waveform at the instant, the steady-state waveform may be represented by a complex number with a magnitude equal to the RMS value of the signal and with a phase angle equal to angle ϕ. In a digital measurement system, samples of the waveform for one (nominal) period are collected according at $t = 0$, and the fundamental frequency component of the Discrete Fourier Transform (DFT) is calculated according to the relation:

$$X = \frac{\sqrt{2}}{N} \sum_{k=1}^{N} X_k \varepsilon^{-j2k\pi/N} , \tag{B.1}$$

where N is the total number of samples in one period, X is the phasor, and x_k is the wave form samples. This definition of the phasor has the merit that it uses a number of samples (N) of the wave form, and is the correct representation of the fundamental frequency component, when other transient component are present [Pha93].

Phasors can be measured for each of the three phases (a, b, c), and the positive sequence phasor can be computed according to its definition:

$$X_1 = \frac{1}{3}\left(X_a + \alpha X_b + \alpha^2 X_c \right) \tag{B.2}$$

where $\alpha = \varepsilon^{j2\pi/3}$.

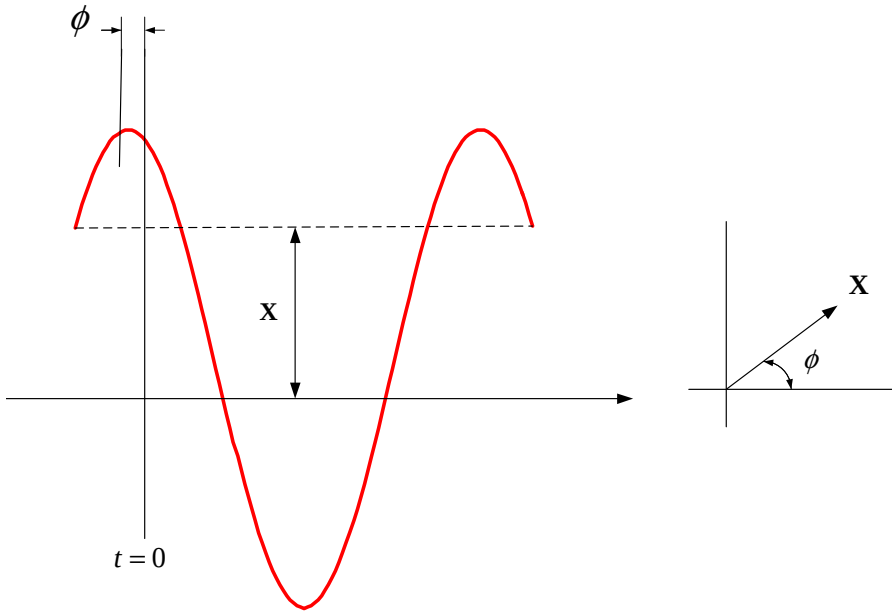


Figure B.1 Phasor representation of a sinusoidal waveform. Adapted from [Pha03].

Synchronization signals could be distributed over any of the traditional communication media currently in use in power systems. Most communication systems, such as leased lines, microwave, or AM radio broadcasts, place a limit on the achievable accuracy of synchronization, which is too coarse to be in practical use. Fibre-optic links could be used to provide high precision synchronization signals, if a dedicated fibre is available for this purpose. If a multiplexed fibre channel is used, synchronization errors of the order of 100 microseconds are possible, and are not acceptable for power system measurements. The Geostationary Operational Environmental Satellites (GOES) systems have also been used for synchronization purposes, but their performance is not sufficiently accurate [Wil92].

The technique of choice at present is the Navstar Global Positioning System (GPS) satellite transmissions. This systems is designed primarily for navigation purposes, but if furnishes a common-access timing pulse, which is accurate to within 1 microsecond at any location at earth. The system uses transmissions from a constellation of satellites in non stationary orbits at about 10 000 miles above the earth's surface. For accurate acquisition of the timing pulse, only one of the satellites need be visible to the antenna. The experience with the availability and dependability of the GPS satellite transmission has been exceptionally good. [Pha93].

Appendix B

FFT Algorithm.

Fast Fourier Transform (FFT) is a classic filtering method. By performing Fourier transforms over a window of N points, the frequency components $k\, f_e / N$ (where f_e is the sampling frequency) can be calculate using the formula below:

$$X(k) = \frac{1}{N} \sum_{i=0}^{N-1} v_i \cdot e^{-j2p^{\frac{ik}{N}}}, \tag{B.3}$$

where $v_i = v\left(\frac{i}{f_e}\right)$.

More generally, $X(k)$ can be considered as the output from a filter which inputs $v(t)$, with exponential coefficients. We thus have:

$$X(k) = X(k,n) = \frac{1}{N} \cdot \sum_{i=0}^{N-1} v_{n+i-N+1} \cdot e^{-j2p\frac{ik}{N}} \tag{B.4}$$

The relationship between FFT and demodulation is thus expressed as follows:

$$X(k) = X(1,n) = \left[\frac{1}{N} \cdot \sum_{i=0}^{N-1} V_{n+i-N+1} \right] \cdot e^{-j2p\frac{n+1}{N}} \tag{B.5}$$

The filter for cutting out the high frequency component consists in averaging over N points, which cancels out frequencies which are multiples of f_e/N. If the sampling frequency is right (i.e. if it is multiple of the frequency which contains the data signal) this filter will remove the $2 f_0$ frequency together with the various harmonics [DCHH92].

Basic definitions.

The follow definition of a real-time or synchronized phasor is provided in the IEEE Standard 1344-1995 [IEEE95]:

- **Anti-aliasing**: By the Nyquist Theorem, the maximum reproducible frequency is one-half the sampling rate. Aliasing is caused when frequencies higher than one-half of the sampling rate are present. This results in the higher frequencies being ‘aliased’ down to look like lower frequency components. Anti-aliasing is providing low pass filtering to block out frequencies higher than those than ca be accurately reproduced by the given sampling rate.

Appendix B

- **Nyquist rate**: the minimum rate that an analog signal must be sampled in order to be represented in digital form. This rate is twice the frequency of that signal.
- **Phase lock**: the sate of synchronization between two ac signals in which they remain at the same frequency and with constant phase difference. This term is typically applied to a circuit that synchronized a variable oscillator with an independent signal
- **Phasor**: a complex equivalent of a simple sine wave quantity such that the complex modulus in the sine wave amplitude and the complex angle (in polar form) is the sine wave phase angle.
- **Synchronism**: the state where connected alternating-current systems, machines, or a combination operate at the same frequency and where the phase angle displacement between voltages in them are constant, or vary about a steady and stable average value.
- **Synchronized phasor:** a phasor calculated from data examples using a standard time signal as the reference for the sampling process. In this case, the phasors form remote sites have a defined common phase relationship.

With real-time waveforms, it is necessary to define a time reference to measure phase angles synchronously. The IEEE standard 1344-1995[IEEE95] defines the start of the second as the time reference for establishing the phasor phase angle value.

The synchronized Phasor measurements convention is shown in Figure B.2

The instantaneous phase angle measurement remains constant at rated frequency when using the start of the second phase reference. If the signal is at off-nominal frequency, the instantaneous phase varies with time. The IEEE standard 1344-1995 defines a steady-state waveform where the magnitude, frequency, and phase angle measurement performance for a waveform do not change. This standard has no requirements regarding Phasor measurement performance for a wave form in transient state [BSG04].

Appendix B

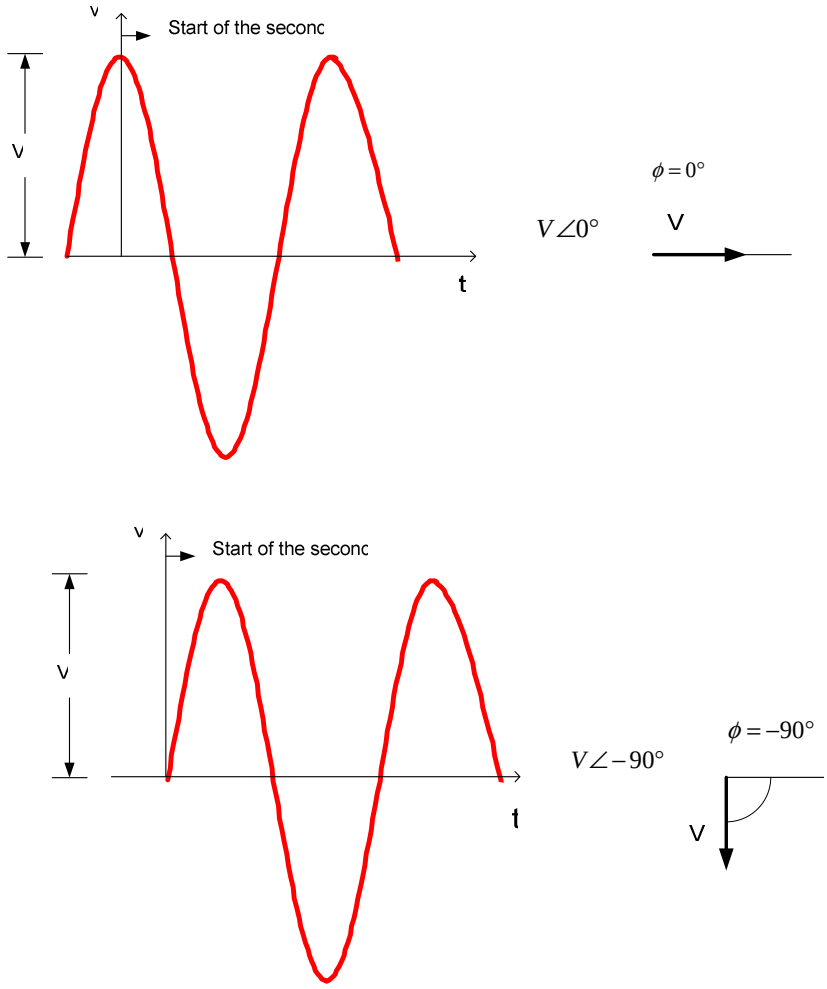


Figure B.2 Synchrophasor measurement convention with respect to time. Adopted from [BSG04].

Phasor Measuring Units.

The PMU was developed at the Virginia Polytechnic Institute, Blacksburg, in the mid-1980s. The GPS time-synchronized PMU measures current and voltages in Phasor detail (i.e. magnitude and phase). Phasor Measurements provide the capability to investigate power system stability in greater detail [HHN05].

Phasor measuring units (PMU) using synchronization signals from the GPS satellite system have evolved into mature tools and are now being manufactured commercially. Figure B.3 shows a typical synchronized Phasor measurement system configuration. The GPS transmission is received by the receiver section, which delivers a phase-locked sampling clock pulse to the analogue-to-digital converter system. The sampled data are converted to a complex number which represents the Phasor of the sampled wave. Phasors of the three phases are combined to produce the positive sequence measurement [BNKH^{+}05].

Appendix B

The GPS receiver provides the 1 pulse-per-second (*pps*) signal, and a time tag, which consists of the year, day, hour, minute and second. The time could be the local time, or the UTC (Universal Time Coordinate). The 1-pps signal is usually divide by a phase-locked oscillator into the required number of pulses per second for sampling of the analog signals. These signals are derived from the voltage and current transformer secondary sides, with appropriate anti-aliasing and surge filtering.

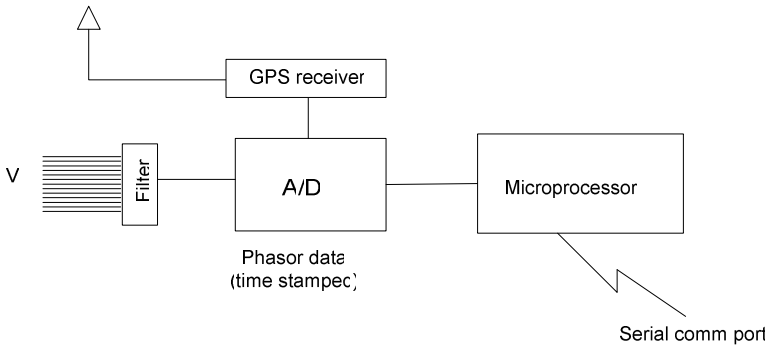


Figure B.3 Phasor measurement unit. Adopte from [BNKH+05]

PMU representation in PSCAD [Man03a].

PSCAD include as a device the Fast Fourier Transform (FFT), Figure B.4 shows this device in the PSCAD environment, which can determine the harmonic magnitude and phase of the input signal as a function of time. The input signals first sampled before they are decomposed into harmonic constituents. Options are provided to use one, two or three inputs. In the case of three inputs, the component can provide output in the form of sequence components. In our simulations we have selected the three 1-phase FFTs combined in one block. The input is processed to provide the magnitudes **Mag** and phase angle **Ph** of the fundamental frequency and its harmonics (including the DC component **dc**)

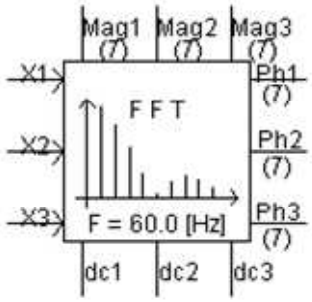


Figure B.4 FFT representation in PSCAD. Adopted from [Man03].

Appendix B

In Fig. B.4 the number 7 means the number of harmonics that FFT block calculates and its imply that the number of samples per period of the fundamental frequency is set to be 16.

The task of frequency scanning involves a few data processing stages:

- Low-Pass Filtering (Anti-Aliasing)
- Sampling & Fourier Transform
- Phase and Magnitude Error Correction.

Figure B.5 illustrates graphically this process inside the FFT block.

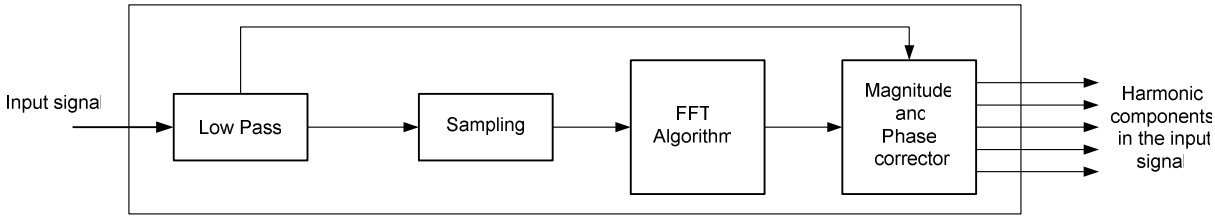


Figure B.5 On-line frequency scanner in PSCAD/EMTDC.

Computations are performed on-line, at each sampling instance, and are based on a sampled data window of the preceding input signal cycle. In accordance with the Nyquist Criteria, data sampling is performed at a frequency greater than double the highest harmonic frequency of interest. Sampling rates may be one of, 16, 32, 64, 127 or 255 samples/cycle of fundamental frequency, which are written to a buffer. In our simulations 16 samples/cycles is selected.

Since the number of samples in a window represents a period of fundamental frequency, the dynamics of a cycle preceding a sample are captured in the computations. It should be noted that outputs of this subroutine contain valid information only if a complete data window is available for computations

It is important to be aware of the inherent aliasing effects due to sampling of the input signal. A low pass, anti-aliasing filter is recommended at all times, unless the input signal is guaranteed not to have any higher order harmonics. This filter is provided within the component.

The harmonic computations are based on a standard Fast Fourier Transformation (FFT) technique, used in digital signal processing. The basis function for computation of phase

angle can either be a fundamental frequency cosine waveform or a sine waveform starting at time = 0.

The harmonics computed are with respect to a given constant fundamental frequency. For situations where the fundamental frequency is variable, the use of a frequency-tracking device is available to the user. The frequency-tracking unit uses the fundamental component of the input signal corresponding to the previous sampling instance (as computed by the FFT routine), to monitor small changes in the frequency of the input signal. This element is meant to monitor minor fluctuations of frequency. Frequency tracking may be enabled or disabled at users discretion.

Gibbs ringing effect, as a result of rectangular data windows, is usually not a problem with harmonics of the fundamental frequency. However, if the sampling frequency is not synchronized to the fundamental frequency of the input signal, the Gibbs effect distortions introduced on the measurement of harmonics may be significant. Therefore, use of the frequency-tracking feature may not be needed unless the fundamental component is guaranteed to be free of frequency swings.