Development and Assessment of Online Dynamic State Estimator for Wind-Penetrated Power Systems

SC Rajendra Kumar

Development and Assessment of Online Dynamic State Estimator for Wind-Penetrated Power Systems

by

Shishir Chandra Rajendra Kumar

to obtain the degree of Master of Science

In
Electrical Engineering

at the Delft University of Technology,
to be defended publicly on Monday August 30, 2021 at 13:00

Student number:	5000149		
Project duration:	November, 2020 – August, 2021		
Thesis committee:	Prof. ir. M.A.M.M. van der Meijden,	TU Delft,	Committee Chair
	Dr. A.I. (Alex) Stefanov,	TU Delft,	Daily Supervisor
	Dr. Matija Naglic,	TenneT TSO,	Company Supervisor
	Dr. M. Ghaffarian Niasar,	TU Delft,	Graduation Committee

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

Acknowledgements

This thesis would not have come to fruition without the constant guidance and support from numerous people. This page is a tribute to those individuals.

If my entire thesis journey can be metaphorically thought as a mountain climb, I would first like to express my deepest gratitude to Dr. Matija Naglic, TenneT TSO. His timely guidance and feedback during the complete course of the master thesis is what helped me reach the summit, i.e., achieve my research goals.

They say that reaching the summit is only half the journey, making a good descent from the mountain equally important. I believe this descent was nothing but my thesis report, which could not have been possible without the instructive feedback and support from Dr. Alex Stefanov, TU Delft. I would like to, therefore, express my sincerest thanks to him.

During the course of the climb, what keeps one going is constant morale support. Likewise, my thesis journey was smoother and full of confidence, due to the morale support and encouragement from Prof. Mart van der Meijden, TU Delft. I would also like to express my heartfelt thanks to him.

Despite the cold blizzard (corona lockdown) raging throughout the course of my journey, my family and friends provided the warmth through motivation and encouragement, and helped me not quiver and waver from my goals. I, hence, would like to express my deepest thanks to all of them.

Finally, to those I have not mentioned here and have played a large or a small role in my thesis journey, I would like to convey my best regards to them.

Shishir Chandra

Tell me and I forget, Teach me and I remember, Involve me and I learn.

Benjamin Franklin

ABSTRACT

The TSO of a power system is mainly responsible for ensuring the stability of the grid. Through continuous monitoring and control of the power system, the TSO maintains stability through emergency actions. Until now, the conventional Static State Estimation has been the Energy Management System (EMS) tool for estimating and monitoring the grid state - bus voltages, currents, and powers. However, energy transition, which involves the decommissioning of conventional generation and their replacement by renewables, leads to a more dynamic grid. In such a case, the information provided by the static state estimator is insufficient. This has, hence, led to the development of the Dynamic State Estimator (DSE), to provide insight into the dynamic properties of a power system, such as rotor angle and rotor speed. The DSE typically uses a Wide-Area Monitoring (WAMS) architecture consisting of Phasor Measurement Units (PMU), to dynamically estimate the internal states of the generators under observation. Hence, the DSE provides improved situational awareness to the TSO. However, the existing literature do not elaborate on how their proposed DSE can be implemented in an online fashion to estimate the dynamic states in near real-time. Such an online implementation is of utmost importance as it showcases how a TSO can deploy the DSE in a real world scenario. Hence, this thesis proposes an online DSE algorithm that performs batch-wise estimation of dynamic states in a near real-time setting. By collecting measurements in batches and introducing the pre-processing steps necessary for these PMU measurements, the algorithm forms the premise for the real-world application of DSE. This algorithm is validated using a cyber-physical testbed comprising a Real Time Digital Simulator and a Synchrophasor Application Development Framework. Additionally, its performance is evaluated and a sensitivity study is conducted to find deterministic relationships between the input error introduced and the estimation error. Finally, future improvements are proposed to make the implementation more suitable to real-world application.

CONTENTS

List of Figures

List of Tables

List of Abbreviations

DFIG Doubly-Fed Induction Generator

DSA Dynamic Security Assessment

DSE Dynamic State Estimator

EKF Extended Kalman Filter

EMS Energy Management System

GNSS Global Navigation Satellite Systems

GSC Grid Side Converter

IGBT Isolated Gate Bipolar Transistor

KF Kalman Filter

PDC Phasor Data Concentrator

PLL Phase Locked Loop

PMU Phasor Measurement Unit

PV Photo Voltaic

PWM Pulse Width Modulation

RSC Rotor Side Converter

RTDS Real Time Digital Simulator

SADF Synchrophasor Application Development Framework

SCADA Supervisory Control And Data Acquisition

SMIB Single Machine Infinite Bus

SSE Static State Estimator

TSO Transmission System Operator

UKF Unscented Kalman Filter

WAMS Wide-Area Monitoring System

WAN Wide Area Network

1

INTRODUCTION

In this chapter, the thesis work is introduced by familiarizing the current research theme in Dynamic State Estimation. Also, the literature study is presented and the corresponding scientific gap is identified. This scientific gap is then addressed by formulating the problem into research questions. Finally, the structure of this thesis report is outlined.

1.1. RESEARCH THEME

FOR every critical disturbance that affects the power system, appropriate corrective measures are undertaken to return to the system's normalcy. Some of these actions are introduced at the control centre of the Transmission System Operator (TSO). However, appropriate remedial actions can only be taken with the help of continuous and efficient monitoring of the power system. Currently, the conventional Static State Estimation (SSE) is used to operate and monitor the power system [1]. It is one of the primary functions of the Energy Management System (EMS). The SSE employs the concept of weighted least squares to estimate the bus voltages, powers, and currents in the grid. It uses the Supervisory Control And Data Acquisition (SCADA) architecture, which comprises sensors in the grid which send measurements at a sample rate of several seconds [2]. However, the recent increase in decommissioning of conventional generation to be replaced by renewables tends to push the system to volatility. In such a case, the SSE can only capture the quasi-steady-state estimates of the grid and neglects any dynamic information such as rotor angle and rotor speed. Consequently, SSE leads to limited situational awareness [3].

To elaborate on renewable integration, The World Energy Outlook 2020 in [4] reports that renewables shall replace thermal generation as the primary form of electricity generation by 2025, as depicted in Fig. 1.1. In particular, wind and solar are expected to have increased energy generation during the period of 2020-2040, with wind being the dominant technology. However, the benefits of renewables are severely countered by the various challenges they bring to the power system stability. These challenges are caused by the instability in the power system due to the introduction of inertia-less and variable generation of the renewables. With the current global energy targets pushing for faster energy transition, there is an increased concern now towards addressing the renewable challenges.

To address the renewable challenges, several studies are made to explain and understand their consequences to the Power System Stability [5]. Power System Stability can be defined as the ability of the system to maintain a state of equilibrium, even after a disturbance occurs [6]. The literature in [7] finds that high solar PV generation with zero inertia can worsen the transient stability especially when faults happen at critical junctures of the grid. This was observed through the instability in the rotor angle. Furthermore, the power-electronic interface of the Doubly-Fed Induction Generators (DFIG) wind turbines affects synchronism and in turn transient rotor stability [8]. Although, it was observed that small signal stability is less affected for both solar and wind depending on fault location and type of controller used respectively.

With such implications on the grid, the TSO is largely responsible for ensuring power system stability through real-time monitoring and control. However, the system stability is put under threat due to the continuously increasing complexity of the grid, which cannot be captured by the current monitoring methods such as SSE. Also, in cases when the dynamic states of a machine, such as the rotor angle, are to be observed, the SSE is unsuitable as it uses a steady-state linear model and not a non-linear dynamic model

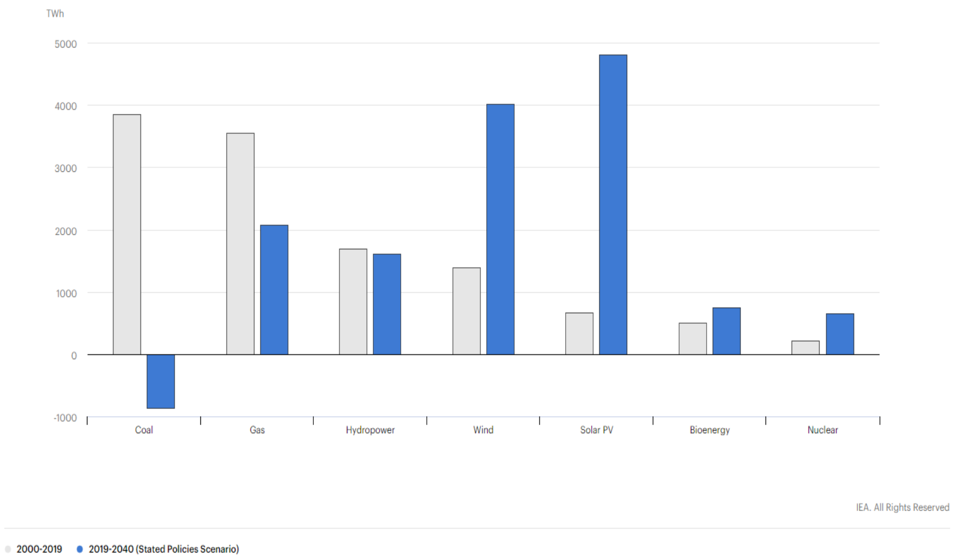


Figure 1.1: A visualization chart depicting the total energy produced per different forms of energy generation [4]

[3]. Nevertheless, the development of the Phasor Measurement Unit (PMU) has made feasible a Wide Area Monitoring System (WAMS) with geographically distributed PMUs sending time-synchronized phasor measurements to the central location. Utilizing such a WAMS architecture, a Dynamic State Estimation (DSE) method can be employed to estimate any dynamic changes in the system state. The DSE can provide improved situational awareness to the TSO by facilitating dynamic estimation and monitoring of all the machines in a wide area of the grid. Consequently, this improved observability will help the TSO in ensuring power system stability.

1.2. LITERATURE REVIEW AND SCIENTIFIC GAP

The Concept of DSE...
The very first occurence of DSE was by Miller et al in [9]. It was developed to estimate the internal flux linkages of the machines in the power system. Since then, DSE has come a long way in terms of applicability with applications ranging from parameter estimation, model callibration, to direct oscillation monitoring [10]. The Kalman Filter (KF) can be referred to as the heart of the DSE due to the central role it plays in the estimation. This Kalman Filter operates with two stages - a prediction stage and a filtering stage. The first stage involves the determination of the kth estimate using the previous estimate and previous covariance matrix at k-1. Whereas, the subsequent stage constitutes the calculation of the new covariance matrix from the new estimates and the corresponding

correction to the estimate.

There are several variants of Kalman Filters that are used for filtering. The research work in [10], studies and identifies the major filters used in DSE. These are – Extended Kalman Filter, Unscented Kalman Filter, Ensemble Kalman Filter, and Particle Filter. The performance of these Kalman Filters is explicitly studied and can be referred in [11]. Accordingly, an Extended Kalman Filter (EKF) focuses mainly on approximate linearization of a non-linear system model but tends to be computationally intense and can give erroneous results with highly non-linear models. Unlike EKF, the Unscented Kalman Filter (UKF), eliminates the linearization process used in EKF through a point-by-point unscented transformation. On the other hand, the Particle Filter estimates by computing the relative likelihood of N particles, again without the linearization assumption. However, this filter is computationally more intense than UKF. Similarly, the Ensemble Kalman Filter predicts and filters by using a Monte Carlo process [12]. The most suitable KF algorithm for DSE is still arguable. It can be observed that the best KF technique depends on the tradeoff between accuracy and computational power. The Particle Filter tends to give a better accuracy while EKF and UKF seem to be the least computationally intense. The literature in [13] further supports UKF as a better algorithm compared to EKF, for being more accurate, easy execution, and computationally less demanding.

In order to predict the state estimates in the prediction stage, the Kalman Filter requires a model. This model is simply the state-space representation of the dynamic behaviour of the machine to be estimated. There are several literature works as in [2], [14], [15] and [16], which showcase a variety of system models for a synchronous generator. However, the looming challenges from fast-paced energy transition has recently brought out research work related to modelling of DFIG wind turbines. In [12], an ensemble form of the Kalman filter is used to estimate callibrate a DFIG. A discretized system model constructed using the rotor and stator voltages as inputs, and the active power and reactive power of the DFIG, as measurements. This DSE is shown to help in estimating the stator and rotor currents, rotor speed, and the parameter – inertia, and hence, provides a base case for a DFIG implementation. However, in [12], converter dynamics are not included. In [17], a similar decoupled approach for the DSE of a DFIG wind turbine is researched. Although similar in state variables, this particular DSE achieves estimation through a simple 3rd order DFIG model with less known parameters. Moreover, the implementation eliminates the requirement to model the converter dynamics, hence reducing the complexity. Authors in [18] develop a UKF-based DSE with a system model including the rotor and grid control dynamics. Further improvement of [18] is considered in [19], where the UKF is replaced by a Particle Filter of the unscented form to improve accuracy, albeit at the cost of computational time. In [20], a wind model was developed to consider the stochastic nature of the wind input. The wind model statistically predicts the wind speed using historical data.

DSE Implementations...
There exist several literature works which explain DSE that help in achieving efficient power system monitoring. The literature in [1] uses an EKF to estimate the state variables of a synchronous machine in SMIB configuration. It also takes into account the unavailability of a field voltage measurement and proposes to solve it through an on-line model. Similarly, the research work in [2] explores a similar DSE implementation for three generators in an IEEE 9-bus system using UKF. However, the on-line aspect of the WAMS is not explained in this work. In [21], a comparison between Least Squares Estimation and UKF when applied to an application of multi-machine parameter estimation is shown. Also, the research in [22] discusses an improved estimator for a WAMS application which is less impacted by bad data and measurement noise. Similarly, the thesis work in [23] helps in improving the performance of the WAMS. It does so by developing a new PMU algorithm that improves accuracy at faster PMU sample rates.

Few literatures also explore online monitoring schemes, which do not use a DSE. For example, the research work in [24] proposes a WAMS application for dynamic voltage stability assessment by computing stability indices and power-voltage curves. Despite being limited to only voltage stability, this work is a clear example of how a monitoring scheme in an online fashion can improve observability to gauge how stable the system is. However in this work, online computation of these indices is straightforward and there are no challenges to overcome for the online implementation. The work in [25] is by far the closest WAMS research to real-time deployment for a TSO. The project presents a real-time WAMS monitoring the Great Britain power system and reviews the challenges encountered. However, during the validation of the PMUs using their test setup, the methodology to implement their monitoring scheme in an online fashion is not discussed.

Scientific Gap 1...
All the above literatures are a good reference to understand what DSE is, what its components are, and how to evaluate and improve its performance. However, these literatures do not speak of how their proposed DSE can be implemented in an online fashion to estimate the dynamic states in near real-time. Here, the phrase "near real-time" implies close to real-time, as any real world application has delay. Such an online implementation is of utmost importance as it showcases how a TSO can deploy the DSE in a real world scenario. Although, this online implementation can seem to be only an implementational challenge, there are many scientific challenges to be overcome. Some of these challenges are explained below:

- *"To estimate the dynamic states in an online fashion, should the DSE use sample-wise or batch-wise PMU measurements?"*
  When estimating in an offline setting, the required measurement samples are fed as a whole to the state estimator to iteratively compute all the corresponding state estimates. However, this cannot be the case in an online setting when there is new data arriving at the PMU's sample rate. Hence, with real-time streaming of PMU data, the estimation can mainly be performed in two ways - per sample or in batches of samples. It is therefore necessary to know which method is more preferred for online estimation

1

and under what circumstances.

- *"If batch-wise estimation is chosen above, what is the ideal size of the batch of measurements for estimation?"*
  In the batch-wise method, the incoming measurements are grouped together in batches of defined size B and then put for estimation. This means that the algorithm should wait for B measurements before estimating. If a large value is chosen for B, there is a consequent delay introduced. This can affect the presentation of the estimates on time. Hence, it is necessary to understand what values for B are acceptable for estimation without introducing considerable delay.

- *"How should the PMU measurements be pre-processed to make it suitable for estimation?"*
  The PMU measurements that arrive can be of the polar form or the rectangular form. The polar form of the PMU measurements, which is used in this thesis, consists of phasor magnitudes and phasor angles. However, these phasor magnitudes are not represented in the per unit form and these phasor angles are referenced with respect to the GPS clock signal. Also, there is a possibility of missing values in the incoming PMU measurements. Hence, the PMU measurements cannot be put for estimation directly and hence, have to be pre-processed. Thus, it is necessary to know what pre-processing steps have to be undertaken to make the phasor measurements suitable for estimation.

Hence, this thesis contributes by addressing these challenges and in turn developing an algorithm to perform DSE in an online fashion, to estimate dynamic states in near real-time.

Scientific Gap 2...
Apart from the absence of a DSE in an online fashion, the literatures do not explain clearly the DQ transformation methodology necessary for the DFIG model. To elaborate, the DFIG state-space model that is used in the DSE requires state inputs i.e. DQ rotor and DQ stator voltages. However, these state inputs are accessible by measuring devices only in the ABC domain, and hence, require to be transformed to the DQ one. This transformation of the state inputs from ABC domain to DQ domain can be a challenge depending on the platform the DSE is being implemented on. Hence, this thesis also contributes by identifying and presenting these transformations, which can serve as a guide to other monitoring and control implementations involving a DFIG.

Scientific Gap 3...
One of the final steps to validate an estimator is to perform a sensitivity analysis. It helps in observing how the error in the estimated state is influenced by the input errors introduced in DSE parameters, measurements, etc. The sensitivity analyses presented in the DSE literatures such as [12] and [15], are explained qualitatively. However, such sensitivity analyses for the DSE have not been quantified. To be specific, the literatures have not developed mathematical relationships between the various errors introduced and the estimation error. Such relationships help in predicting the state error for any given

value of error introduced. Hence, this thesis also contributes by determining such relationships.

1.3. RESEARCH QUESTIONS

To address and fill the above presented scientific gaps, this thesis answers the following research questions:

- What are the transformations, for the DFIG, that need to be performed to bring measurements from the ABC domain to the DQ domain?
- How to develop an online dynamic state estimator for power systems with a centralized control architecture?
- What is the sensitivity relationship between the input error introduced and the estimation error for synchronous generator model and DFIG model?

1.4. STRUCTURE OF THE THESIS

- **Chapter 1: Introduction**

  The thesis project is introduced by familiarizing the current research theme in Dynamic State Estimation. From the literature study presented in the research theme scientific gaps are identified and presented. This scientific gap is addressed by formulating the problem into research questions.

- **Chapter 2: Machine Modelling**

  The electrical and mechanical equations that govern the dynamic behaviour of the synchronous generator and the DFIG wind turbine are explained in this chapter. These equations help us in understanding how a mathematical state-space model can be constructed to best represent the machine dynamics.

- **Chapter 3: The Kalman Filter**

  This section lays down a detailed explanation of the discretization techniques of the process models. This is followed by a subsequent introduction to a linear kalman filter. Finally, the modifications to this kalman filter to accomodate non-linear process models are discussed.

- **Chapter 4: Online Estimator**

  This chapter explains the major contributions of this thesis. In particular, the DQ transformations that are necessary for the DFIG are identified and the online-DSE algorithm used to perform online estimation in a near real-time is presented.

- **Chapter 5: Case Study: Simulation and Results**

This part of the report explains how the simulation platform was integrated and the case study used in this thesis to validate the simulation. The final results of the on-line DSE is interpreted and the sensitivity study is layed out to show the relationship between input errors and the state output.

- **Chapter 6: Conclusion**

  This final chapter concludes the thesis by revisiting the contributions of this thesis work. The future improvements to the thesis are also discussed in this section.

2

MACHINE MODELLING

Chapter 2 explains the electrical and mechanical equations that govern the dynamic behaviour of the synchronous generator and the DFIG. These equations help in better understanding the dynamic state-space models of each machine.

Over the past century, electricity has found its way well into our lives, to an extent where it is a deep-rooted necessity. From the power system's perspective, this can be credited to the continuous developments in the generation, transmission and deliverance of electrical power to the consumers; leading to a sophisticated electrical power system. However, despite any such improvements to grid technologies, the primary form of power generation remains to be conventional - employing a three phase synchronous generator. These machines generate power at the same electrical frequency of the system, maintain synchronism, and eventually contribute to the system stability of the grid.

On the contrary, there is a push to move towards renewable energy sources. In particular, wind can be regarded as one of the most important renewable energy sources for mankind. From transportation through sailboats to grinding grain in wind-mills, wind energy has proven itself to be a catalyst in human development. The recent advancements in wind turbine technology tell us the same story. Despite the variable nature of wind, a wind turbine is capable of harnessing wind energy through its aerodynamically designed blades. In particular, variable speed wind turbines such as the Doubly-Fed Induction Generator (DFIG) are capable of harnessing wind energy at rotor speeds above and below the synchronous speed. The DFIG, hence, has gained immense popularity as the primary choice in wind generation.

Since, the grid in the future will contain generation from conventional synchronous generators, as well as from DFIG wind farms, it is important for the TSO to monitor their dynamics through DSE to ensure stability. To build such a DSE, we need to understand better the mathematical models that best represent these machines. Hence, in this chapter, the equations involved in the modelling of the synchronous generator and DFIG are discussed. Finally, the subsequent models used in this thesis are given.

2.1. SYNCHRONOUS MACHINE: AN ELECTRICAL PERSPECTIVE

With three windings, spaced 120° apart, the armature of a $3-\Phi$ synchronous machine has 3 phase voltages induced which produce a time-varying 3-phase flux. This time-varying flux has three components which are 120° apart in time, and interacts with the rotor flux produced in the rotor circuit. The two interacting circuits is explained as depicted in the Fig. 2.1. At steady state, the rotor runs at the same synchronous speed n_s governed by the equation 2.1 [6], where p depicts how many poles are on the rotor, and f depicts the electrical frequency.

$$n_s = \frac{120f}{p} \tag{2.1}$$

Every synchronous machine contains damper windings whose main function is to reduce any dynamic oscillations. The type and construction of these windings depend on the type of rotor used - salient or non-salient type. The effect of these two rotor designs can be extensively studied in [6]. Depending on the rotor pole pairs $\frac{p_f}{2}$, the electrical

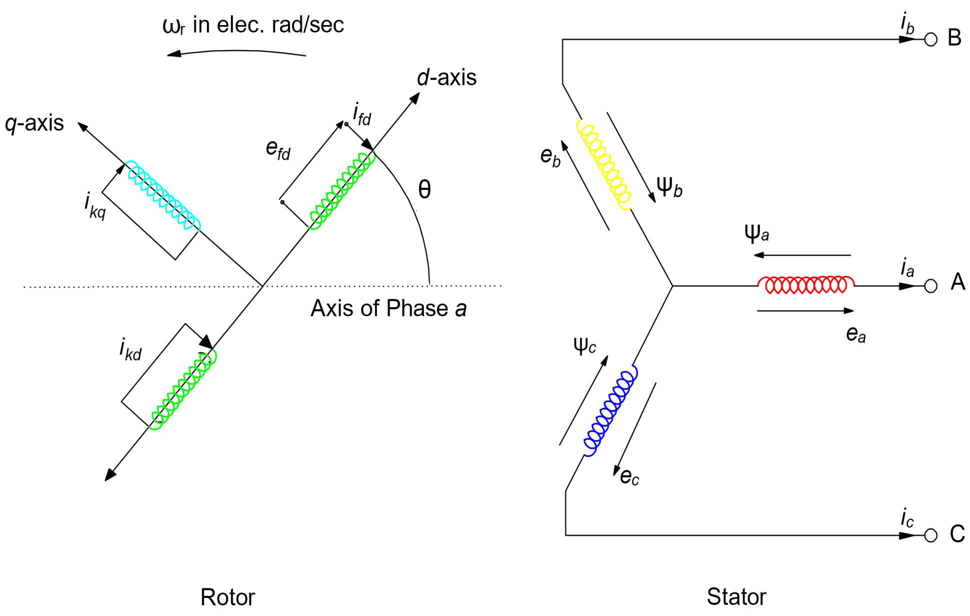


Figure 2.1: The 3ϕ Stator and Rotor winding circuits - adapted from [6]

angle in elec.rad/s is related to the mechanical angle in rad/s by equation 2.2 [6].

$$\theta = \frac{p_f}{2}\theta_m \tag{2.2}$$

Similarly, one can determine the Magneto-Motive Force (F) for the three phases as the set of equations in 2.3 [6] with γ being the angle along the stator for phase a and K being a constant,

$$\begin{aligned} F_a &= Ki_a\cos\gamma \\ F_b &= Ki_b\cos\left(\gamma - \tfrac{2\pi}{3}\right) \\ F_c &= Ki_c\cos\left(\gamma + \tfrac{2\pi}{3}\right) \end{aligned} \tag{2.3}$$

with the instantaneous three phase stator currents represented as in equations 2.4 [6], with $\omega_s = 2\pi f$ as the synchronous electrical frequency in elec.rad/s.

$$\begin{aligned} i_a &= I_m\cos(\omega_s t) \\ i_b &= I_m\cos\left(\omega_s t - \tfrac{2\pi}{3}\right) \\ i_c &= I_m\cos\left(\omega_s t + \tfrac{2\pi}{3}\right) \end{aligned} \tag{2.4}$$

Substituting equations 2.4 in equations 2.3, we can obtain the total Magnetomotive force

Notations	**Parameters**
e_a, e_b, e_c	Stator instantaneous voltages
i_a, i_b, i_c	Stator instantaneous currents
E_{fd}	Excitation voltage of the Rotor
i_f	Field current
R_{fd}, R_{kd}, R_{kq}	Rotor resistances
L_{xx}	Self-inductances of the Stator and Rotor the
L_{xy}	Magnetizing inductances of the stator and rotor
R_a	Resistance of the Stator
x_d and x_q	Reactances in the dq realm
x'_d and x'_q	Transient Reactances in the dq realm

Table 2.1: The notations and subsequent parameters of synchronous generators.

of the stator as travelling wave equation given in equation 2.5 [6].

$$\begin{aligned} F_{\text{total}} &= F_a + F_b + F_c \\ &= \frac{3}{2} K I_m \cos\left(\gamma - \omega_s t\right) \end{aligned} \tag{2.5}$$

This equation re-establishes the fact that the rotating magnetic field is sinusoidal spaced-out with a constant frequency ω_s.

Electrical Model of a Synchronous Machine

The formulation of the electrical equations of a synchronous machine requires a set of assumptions to be considered, as a dynamically-complete representation of the synchronous machine is mathematically complex. These assumptions is explained in page 54 of [6]. But before, diving into the model equations, let's define all the variables which shall help in the construction of this model as in Table 2.1.

Stator equations

The behaviour of the stator can be represented through the following phase voltage and flux linkage equations. The 3ϕ armature-voltage equations can be represented as in equations 2.6 [6]. Here, p indicates differential operation.

$$\begin{aligned} e_a &= \tfrac{d\psi_a}{dt} - R_a i_a = p\Psi_a - R_a i_a \\ e_b &= p\Psi_b - R_a i_b \\ e_c &= p\psi_c - R_a i_c \end{aligned} \tag{2.6}$$

The stator and rotor self-inductances vary sinusoidally based on θ, where θ is the angular difference between the stator and rotor. Hence, by reducing F in equations 2.3

into its components, the self inductances of stator windings are better represented as equations 2.7 [6], where the l_{aa0} and l_{aa2} are the parameters of the sinusoidally varying self-inductance as shown in figure 2.2.

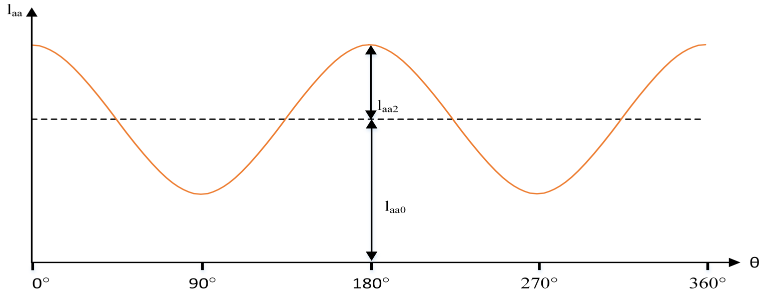


Figure 2.2: Variation of Self-Inductance of Stator - adapted from [6]

$$\begin{aligned} l_{aa} &= L_{aa0} + L_{aa2}\cos 2\theta \\ l_{bb} &= L_{aa0} + L_{aa2}\cos 2\left(\theta - \tfrac{2\pi}{3}\right) \\ l_{cc} &= L_{aa0} + L_{aa2}\cos 2\left(\theta + \tfrac{2\pi}{3}\right) \end{aligned} \tag{2.7}$$

Similarly the stator mutual inductances l_{ab}, l_{bc}, and l_{ca}, can be better represented as equations 2.8 [6], where l_{ab0} and l_{ab2} are the parameters of the sinusoidally varying mutual-inductance as shown in figure 2.3.

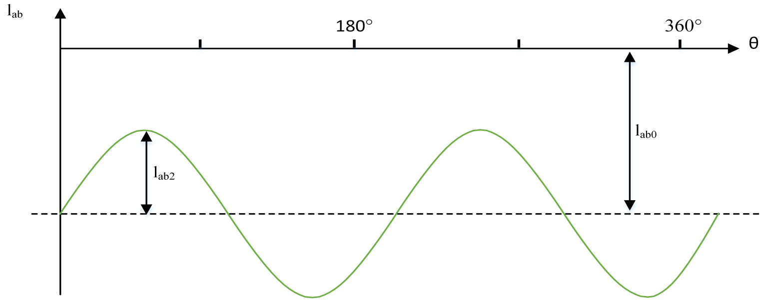


Figure 2.3: Variation of Mutual-Inductance - adapted from [6]

$$\begin{aligned} l_{ab} &= -L_{ab2}\cos\left(2\theta + \tfrac{\pi}{3}\right) - L_{ab0} \\ l_{bc} &= -L_{ab2}\cos(2\theta - \pi) - L_{ab0} \\ l_{ca} &= -L_{ab2}\cos\left(2\theta - \tfrac{\pi}{3}\right) - L_{ab0} \end{aligned} \tag{2.8}$$

Using equations in 2.7 and equations in 2.8, the flux linkages of the three phases of the stator winding can be expressed in the form of self and mutual inductance parameters as shown in page 66 of [6].

These equations are, however, very complex due to parameters such as the self- and mutual- inductances sinusoidally fluctuating with the angle θ. Hence, it would be beneficial to consider a new frame of reference such as the dq0 frame, which eliminates any variation in the parameters due to θ. Accordingly, the stator currents represented with equations in 2.4, can be determined in the dq0 frame through the transformation in 2.9 [6].

$$\begin{bmatrix} i_d \\ i_q \\ i_0 \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \cos\theta & \cos\left(\theta - \frac{2\pi}{3}\right) & \cos\left(\theta + \frac{2\pi}{3}\right) \\ -\sin\theta & -\sin\left(\theta - \frac{2\pi}{3}\right) & -\sin\left(\theta + \frac{2\pi}{3}\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} \tag{2.9}$$

Similarly, the stator voltages can also be transformed to the dq0 frame with the help of this matrix as in equations 2.10 [6].

$$\begin{aligned} e_d &= p\psi_d - \psi_q p\theta - R_a i_d \\ e_q &= p\Psi_q + \psi_d p\theta - R_a i_q \\ e_0 &= p\Psi_0 - R_a i_0 \end{aligned} \tag{2.10}$$

The varying inductances are also represented in the dq realm as in equation 2.11 [6].

$$\begin{aligned} L_d &= \tfrac{3}{2} L_{aa2} + L_{ab0} + L_{aa0} \\ L_q &= -\tfrac{3}{2} L_{aa2} + L_{ab0} + L_{aa0} \\ L_0 &= -2L_{ab0} + L_{aao} \end{aligned} \tag{2.11}$$

These inductances help in transforming the flux linkages as shown in the equations 2.12 [6].

$$\begin{aligned} \Psi_d &= -L_d i_d + L_{afd} i_{fd} + L_{akd} i_{kd} \\ \Psi_q &= -L_q i_q + L_{akq} i_{kq} \\ \Psi_0 &= -L_0 i_0 \end{aligned} \tag{2.12}$$

Rotor equations

The behaviour of the rotor can also be represented through the following set of equations. In particular, the equations in 2.13 depict the rotor voltage dynamics [6]. Unlike the stator, the self- and mutual inductances of the rotor are constant irrespective of the angle θ.

$$\begin{aligned} E_{fd} &= p\Psi_{fd} + R_{fd} i_{fd} \\ 0 &= p\psi_{kd} + R_{kd} i_{kd} \\ 0 &= p\Psi_{kq} + R_{kq} i_{kq} \end{aligned} \tag{2.13}$$

Similar to the dq0 transformation of the stator currents, voltages and flux linkages, the rotor flux linkages can also be moulded into the dq0 reference frame as in equations 2.14, where the field inductances are as given in [6].

$$\begin{aligned}\Psi_{fd} &= -\tfrac{3}{2}L_{afd}i_d + L_{fkd}i_{kd} + L_{fd}i_{fd}\\ \psi_{kd} &= -\tfrac{3}{2}L_{akd}i_d + L_{kkd}i_{kd} + L_{fkd}i_{fd}\\ \Psi_{kq} &= L_{kkq}i_{kq} - \tfrac{3}{2}L_{akq}i_q\end{aligned} \tag{2.14}$$

Electric Power equations

Based on the above $dq0$ frame equations of currents, voltages and flux linkages, the electromagnetic output power of the synchronous machine can be represented through the equation of 2.15 [6].

$$P_t = \left(\Psi_d i_q - \Psi_q i_d\right)\omega_r + \tfrac{3}{2}\left[\left(i_d p\dot{\Psi}_d + i_q\dot{\Psi}_q + 2i_\alpha\dot{\Psi}_0\right) - \left(i_d^2 + i_q^2 + 2i_0^2\right)R_a\right] \tag{2.15}$$

The part of the electromagnetic power of the synchronous machine that is transferred to the air gap at per unit rotor speed is nothing but the electrical torque as in equation 2.16 [6].

$$T_e = \frac{3}{2}\left(\Psi_d i_q - \Psi_q i_d\right)\frac{p_f}{2} \tag{2.16}$$

All these aforementioned equations are converted to the per unit domain and the respective methodology for this process can be understood from [6].

2.2. SYNCHRONOUS MACHINE: A MECHANICAL PERSPECTIVE

Apart from the electrical mechanism, the understanding of mechanical behaviour of the synchronous machine during a disturbance is necessary for power system stability studies. To be more specific, the net effect between the electrical T_e and mechanical T_m torques and its effect on the rotor speed and angle, has to be well represented in its mathematical model. Similar to the previous section, all the equations are converted to the per unit domain.

During any disturbance such as a three phase symmetrical fault, the net effect between the electrical torque T_e and mechanical torque T_m can be captured through the àccelerating torque T_a, which is given by equation 2.17 [6].

$$T_a = T_m - T_e \tag{2.17}$$

Since, the T_a is directly influenced by the time differential of the mechanical rotor speed ω_m, the complete equation of motion can be written as equation 2.18 [6], where J is the proportionality constant which represents moment of inertia of the rotor.

$$J\frac{d\omega_m}{dt} = T_a = T_m - T_e \tag{2.18}$$

This moment of inertia J can be modelled in two forms - the inertia constant H and the mechanical starting time T_M (not to be confused with mechanical torque T_m). In this implementation, we choose the former. Hence, the above equation is represented with inertia constant H as in equation 2.19 [6], with ω_{0m} as the rated angular speed of the rotor, and VA_{base} as the rated base apparent power.

$$H = \frac{1}{2}\frac{J\omega_{0m}^2}{\text{VA}_{\text{base}}} \tag{2.19}$$

Furthermore, since the mechanical rotor speed ω_m is the time differential of the rotor angle δ, the above mechanical equations is better represented as follows in equation 2.20 [6], with K_D being the per unit damping factor and ω_0 being the rated angular speed.

$$\frac{2H}{\omega_0}\frac{d^2\delta}{dt^2} = \bar{T}_m - \bar{T}_e - \frac{K_D}{\omega_0}\frac{d\delta}{dt} \tag{2.20}$$

Thus, equation 2.20, also known as the swing equation, is used as a good approximate to the mechanical behaviour of the synchronous machine.

2.3. SYNCHRONOUS MACHINE: A MATHEMATICAL MODEL

Using the aforementioned equations to represent the electrical and mechanical behaviour of the machine, a mathematical state space model can be derived for dynamic state estimation and monitoring as shown in literatures [1],[15], [16], [26], and [27].

Accordingly, the state space vector x contains the same state variables as in [1]. To be specific, these are transient voltages in the dq realm - e'_q and e'_d respectively, the rotor speed deviation $\Delta\omega$, and the rotor angle δ.

$$\mathbf{x} = \begin{bmatrix} \delta & \Delta\omega & e'_q & e'_d \end{bmatrix}^T \tag{2.21}$$

and a state input vector u with state inputs as mechanical torque T_m, field excitation voltage E_{fd}, and terminal voltage V_t [1],

$$\mathbf{u} = \begin{bmatrix} T_m & E_{fd} & V_t \end{bmatrix}^T \tag{2.22}$$

the state transition equation can be deduced as, with T'_{do} and T'_{qo} being the the dq transformed time constants [1].

$$
\begin{aligned}
\dot{\delta} &= \omega_o \Delta\omega \\
\dot{\Delta\omega} &= \tfrac{1}{2H}\left(T_m - T_e - D\Delta\omega\right) \\
\dot{e}'_q &= \tfrac{1}{T'_{do}}\left(E_{fd} - e'_q - \left(x_d - x'_d\right) i_d\right) \\
\dot{e}'_d &= \tfrac{1}{T'_{qo}}\left(-e'_d + \left(x_q - x'_q\right) i_q\right)
\end{aligned}
\tag{2.23}
$$

In order to have the state differentials $\dot{x}$ purely as in the form of states x and inputs u, the stator currents i_d and i_q are given similar to [1],

$$
\begin{aligned}
i_d &= \tfrac{e'_q - V_t \cos\delta}{x'_d} \\
i_q &= \tfrac{V_t \sin\delta}{x_q}
\end{aligned}
\tag{2.24}
$$

Moreover, the electromagnetic torque can be approximated to be the electromagnetic power P_e in the per unit domain and is given as a function of the states as [1],

$$
T_e \cong P_t = \frac{V_t}{x'_d} x_3 \sin x_1 + \frac{V_t^2}{2}\left(\frac{1}{x_q} - \frac{1}{x'_d}\right)\sin 2x_1 \tag{2.25}
$$

Hence, the final state transition equations are given by [28],

$$
\begin{aligned}
\dot{x}_1 &= \omega_o x_2 \\
\dot{x}_2 &= \frac{1}{2H}\left[T_m - \left(\frac{V_t}{x'_d} x_3 \sin(x_1) + \frac{V_t^2}{2}\left(\frac{1}{x_q} - \frac{1}{x'_d}\right)\sin(2x_1)\right) - Dx_2\right] \\
\dot{x}_3 &= \frac{1}{T'_{do}}\left[E_{fd} - x_3 - \left(x_d - x'_d\right)\left(\frac{x_3 - V_t \cos x_1}{x'_d}\right)\right] \\
\dot{x}_4 &= \frac{1}{T'_{qo}}\left[-x_4 + \left(x_q - x'_q\right)\left(\frac{V_t \sin x_1}{x_q}\right)\right]
\end{aligned}
\tag{2.26}
$$

Similarly, the measurement equation can be devised as follows with the measurements being the active power P and reactive power Q injected into the grid by the machine. A third measurement is added as the terminal frequency f_r, with f_0 being the rated frequency [28].

$$
\begin{aligned}
y_1 = P &= \frac{V_t}{x_d'} x_3 \sin x_1 + \frac{V_t^2}{2}\left(\frac{1}{x_q} - \frac{1}{x_d'}\right)\sin 2x_1 \\
y_2 = Q &= \frac{V_t}{x_d'} x_3 \cos x_1 - V_t^2\left(\frac{\cos^2 x_1}{x_d'} + \frac{\sin^2 x_1}{x_q}\right) \\
y_3 = f_r &= f_o\,(x_2 + 1)
\end{aligned}
\tag{2.27}
$$

Thus, the final mathematical model for the synchronous machine, comprising of state transition equations as in equations 2.26 and the output measurement equations as in equations 2.27, is obtained. This mathematical model is a good approximation of the dynamic behaviour of the synchronous generator and can be directly deployed in our state estimator.

2.4. DFIG: THE INDUCTION MACHINE

A DFIG, by principle of working, is similar to an induction machine but differs in the connection of the rotor. All slip-ring induction machines have their wound rotor connected to a 3-Φ starter through a slip-ring and brush arrangement. However, a doubly-fed induction machine has its wound rotor energized by grid power via a back-to-back converter. This construction can be better understood from the block diagram of a DFIG in Fig.2.4.

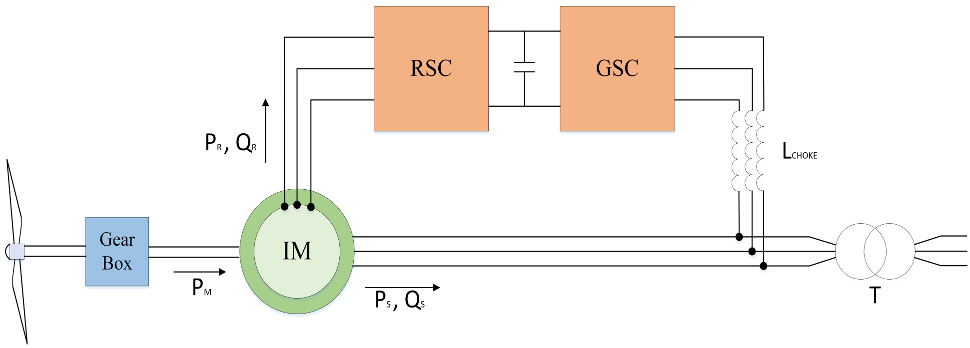


Figure 2.4: Construction of a Doubly-Fed Induction Generator - adapted from [29]

The mechanical power P_m is transferred from the rotating turbine blades to the rotor of the induction machine through a gear box as shown. The rotor currents, fed through the above converter configuration, help in managing the rotor flux at the desired value. By drawing reactive power from the connected grid, the stator is also energized to produce a 3-phase rotating magnetic field. Ultimately, three phase power is generated through the coupling of these rotor and stator magnetic fields at a torque determined by their cross product [29]. Since, the rotor speed n_r is not the same as the synchronous stator speed n_s, its slip can be defined as in [29],

Slip	Mode of Operation
0	No EMF generated
1	Rotor at standstill
0 < slip < 1	Motoring
slip > 1	Braking
slip < 0	Generating

Table 2.2: Mode of operation of an induction machine based on slip

$$s = \frac{n_s - n_r}{n_s} \tag{2.28}$$

where, the stator synchronous speed can be represented as [29],

$$n_s = \frac{120f}{p} \tag{2.29}$$

Based on the slip of the induction machine the mode it operates on varies. This can be best explained by Table 2.2.

Induction Machine: An equivalent representation

Since, the principle of working of an induction machine resembles to that of a trans-

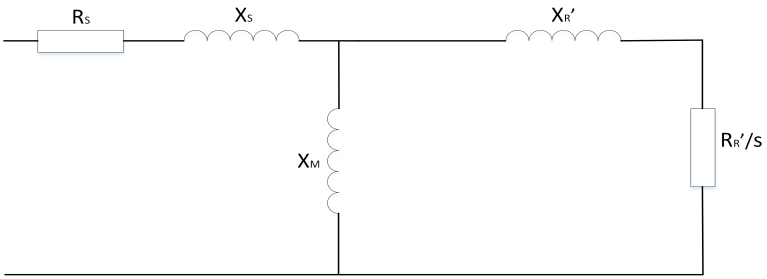


Figure 2.5: Representation of an Induction Machine - 1 - adapted from [29]

former, it is generally accepted to use a similar equivalent ciruit as shown in Fig. 2.5. This representation comprises stator parameters R_S and X_S and rotor parameters R_R and X_R. In this circuit, the parameters of the rotor are transformed to the stator. The stator and rotor reactances X_S and X_R respectively are used to represent the leakage fluxes that are lost and not useful [29]. The magnetising/mutual inductance X_M is however used to represent the flux that is useful in producing electrical power.

To better represent the generated power delivered, the equivalent circuit can be modified as in Fig.2.6. Here, the rotor reactance referred from the stator side R'_R/s is split into two

2

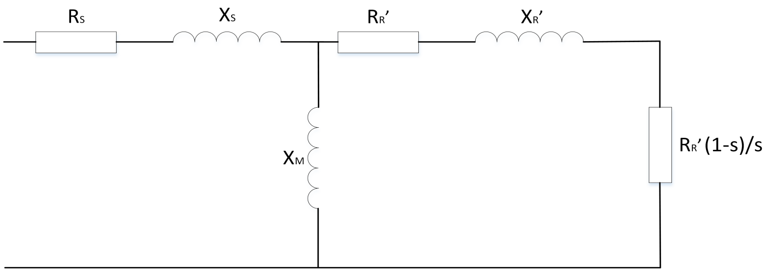


Figure 2.6: Equivalent Circuit of an Induction Machine - 2 - adapted from [29]

terms - R'_R and $R'_R(1-s)/s$. The R'_R term now signifies copper losses and the $R'_R(1-s)/s$ term signifies the rotor power generated. This is given as in [29],

$$P_{mech} = 3|i_r|^2\left(\frac{1-s}{s}\right)R'_r \tag{2.30}$$

The subsequent torque produced is given as in [29],

$$T_{\text{mech}} = 3\left|i'_r\right|^2\left(\frac{1-s}{s}\right)\frac{R'_r}{\omega_m} \tag{2.31}$$

Further analysis in [29] shows that the required torque can be obtained through the control of the rotor currents. Hence, a DFIG relies on the control scheme to maintain the rotor currents and in turn helps in producing optimal power at variable wind speeds.

2.5. DFIG: THE BACK-TO-BACK CONVERTER

This converter configuration, which can be divided into two stages - Rotor Side Converter (RSC) and Grid Side Converter (GSC), couples the rotor of the machine to the grid. Connecting the two converters is a dc storage capacitor. The GSC operates at synchronous system frequency and helps in keeping the capacitor at a constant, desired voltage [29]. Similarly, the RSC helps in managing the rotor currents at different turbine speeds. Hence, this converter is designed to operate at variable rotor frequencies [29].

As shown in Fig. 2.7, the configuration comprises two voltage source converters separated by a storage capacitor. Each voltage source converter is of two-level, three-leg, six switch type, with Insulated-gate Bipolar Transistors or IGBTs. The control of these switches are achieved by generation of firing pulses through the Pulse Width Modulation (PWM) technique. This is done in particular by comparing a carrier signal at a switching frequency f_{switch} with a modulating or reference signal to obtain a pulses of varied width. This variation of pulse width is generally controlled using the modulation index m and is given by the following equation [29] where t_{on} is the "ON" period for a switch in each leg of the bridge, and T_{sw} is the switching time period.

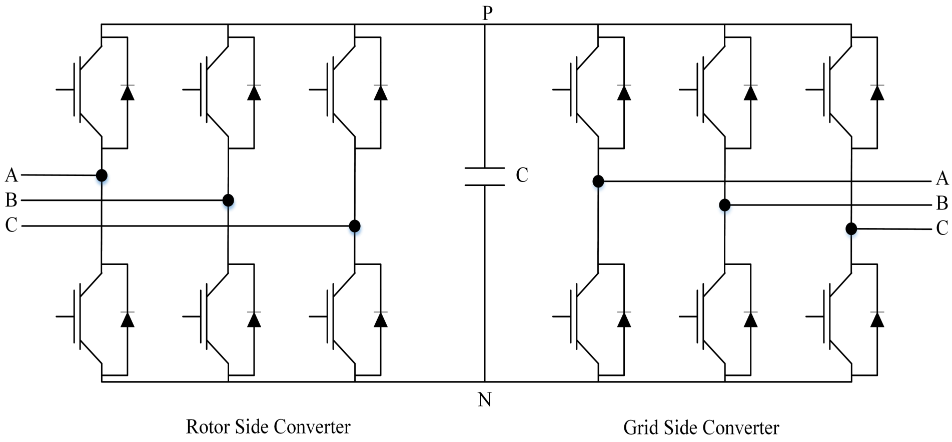


Figure 2.7: The converter configuration in a DFIG - adapted from [30]

$$m = \frac{t_{on}}{T_{sw}} \tag{2.32}$$

By defining modulation indices for each leg of the converter bridge as m_a, m_b, and m_c, the relationship [29] between capacitor voltage V_{dc} and phase voltage can be given as shown below.

$$\begin{aligned} V_a &= m_a V_{dc} \\ V_b &= m_b V_{dc} \\ V_c &= m_c V_{dc} \end{aligned} \tag{2.33}$$

The two converters in the configuration influence the behaviour of the DFIG wind generator. Hence, it is imperative to understand their dynamic control equations. The research in [20] explains the dynamics of the rotor side converter concisely. Accordingly, the control dynamics of this converter can be better represented without the fast current regulator.

Rotor Control

The reactive power Q is maintained by finding a reference rotor current in the q-axis as shown below [20].

$$\begin{aligned} \dot{x}_1 &= Q_{ref} - Q \\ i_{qr_{ref}} &= k_{p1}\dot{x}_1 + k_{i1}x_1 \end{aligned} \tag{2.34}$$

Similarly, a reference rotor current in the d-axis can be found using power loss P_{loss} as

2

shown below [20].

$$
\begin{aligned}
i_{dr_{ref}} &= \frac{\left(P_{ref} - P_{\text{loss}}\right) L_s}{\omega_r \Phi_s L_m} \\
\Phi_s &= \sqrt{\Phi_{ds}^2 + \Phi_{qs}^2}
\end{aligned}
\tag{2.35}
$$

In the same fashion, the rotor voltages are referenced as below [20].

$$
\begin{aligned}
v_{dr_{ref}} &= R_r i_{dr} - (\omega_s - \omega_r)\left(L_r i_{qr} + L_m i_{qs}\right) \\
v_{qr_{ref}} &= R_r i_{qr} + (\omega_s - \omega_r)\left(L_r i_{dr} + L_m i_{ds}\right)
\end{aligned}
\tag{2.36}
$$

Grid Control
In this control scheme, a d-axis reference grid current is formulated as shown below [20]. A q-axis reference grid current is taken to be zero.

$$
\begin{aligned}
\dot{x}_2 &= v_{dc_{ref}} - v_{dc} \\
i_{dg_{ref}} &= k_{p2}\dot{x}_2 + k_{i2}x_2
\end{aligned}
\tag{2.37}
$$

In the same fashion, the grid voltages are referenced as below [20].

$$
\begin{aligned}
v_{dg_{ref}} &= v_{ds} + \omega_s L_r i_{qg_{ref}} - R_r i_{dg_{ref}}, \\
v_{qg_{ref}} &= v_{qs} - \omega_s L_r i_{dg_{ref}} - R_r i_{qg_{ref}}
\end{aligned}
\tag{2.38}
$$

All notations can be interpreted from Table 2.4.

2.6. DFIG: THE WIND TURBINE

As mentioned before, the mechanical power P_m is transferred from the rotating turbine blades to the machine through a gearbox-coupling system. The rotor power can be derived from that of the stator using the equation shown below [29]. This means that the RSC is designed with a power rating defined by the slip s. Hence, the RSC has a power rating lower to that of the stator power, proving an economical design.

$$
P_r = -sP_s \tag{2.39}
$$

The amount of power that is drawn from the incident wind is dependent on the performance or power coefficient C_p. The performance coefficient is a measure of the efficient conversion of the wind energy to useful mechanical energy for power generation. This value is usually less than 50 percent for wind turbines and is usually given by the following equation [31], Here, β is the pitch angle, and the λ is the tip-speed ratio.

Coefficients	Values
W1	0.47
W2	0.0167
W3	7.5
W4	0.15
W5	0.00184
W6	0.01

Table 2.3: The coefficients used in the power coefficient equation

$$C_p = \left[\left(W_1 - W_2\beta \right) \sin\left[1.5707 \frac{(\lambda - 3y(\lambda))}{\left(W_3 - W_4\beta \right)} \right] - (\lambda - 3y(\lambda)) W_5\beta \right] + \frac{W_6}{(1+\lambda)} \tag{2.40}$$

The variables W_1, W_2, W_3, W_4, W_5, and W_6 are coefficient values. The table 2.3 shows the coefficient values used in the turbine design used in this thesis. With the power coefficient established, the turbine power output is represented as shown below [31].

$$P_m = C_p \frac{dS_a}{2} u^3 \tag{2.41}$$

Here, the power is represented through the air density d, wind speed u, and blade area S_a covered. The air density d is modelled as shown below [31], with T as the temperature, and A as the height.

$$d = \frac{1.0}{(1.0 + 0.00367T)} e^{(-0.000125A)} \tag{2.42}$$

Now, with the obtained mechanical power P_m, the torque can be obtained using the following relation as in [18], with w_r as the rotor speed.

$$T_m = \frac{-P_m}{w_r} \tag{2.43}$$

2.7. DFIG: THE DQ TRANSFORMATION

All control methodologies of the voltage source converters of a DFIG are mainly implemented in the DQ domain. Hence, in our case, the control variables such as rotor voltage, rotor current, stator voltage, and stator current have to be transformed to the d-q realm for feasibility of implementation. According to [32], three-phase variables such as voltages and currents can be brought into the d-q realm by using the concept of Park's transformation given by the following equation [32].

$$P = \sqrt{\frac{2}{3}} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \cos\theta & \cos(\theta - 120) & \cos(\theta + 120) \\ \sin\theta & \sin(\theta - 120) & \sin(\theta + 120) \end{bmatrix} \tag{2.44}$$

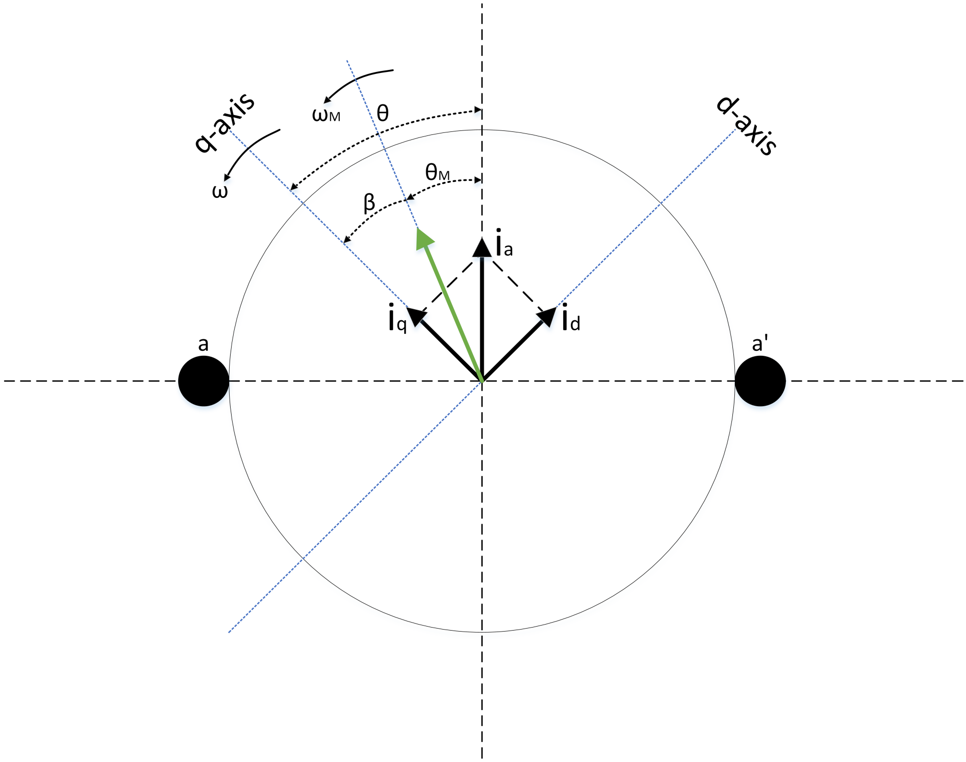


Figure 2.8: The DQ axes containing stator and rotor variables - adapted from [32]

Here, θ is the angle between the rotating DQ reference frame and the stator A-axis. It is represented by the formulation below [32]. A Phase Locked Loop (PLL) is used to derive this θ from the ABC instantaneous values.

$$\theta = \int_0^t \omega(\gamma) d\gamma + \theta(0) \tag{2.45}$$

By principle of d-q transformation, it is desired to transform to a d-q axis compared to a common frame. In most cases, the common frame is the synchronous frame. However, the rotor variables and the stator variables themselves exist in different frames, as the former is stationary and the latter rotates. Hence, the angle of transformation differs for both the stator and rotor variables [32]. This can be better explained with the Fig. 2.8.

Notations	Parameters
R_s, L_s	Stator Resistance and Inductance
R_r, L_r	Rotor Resistance and Inductance
L_m	Magnetizing Inductance
ω_r	Rotor Speed
ω_s	Synchronous Speed
T_m	Mechanical Torque
T_e	Electromagnetic Torque
H	Inertia Constant
D	Damping Coefficient

Table 2.4: Notations and their corresponding meanings for the DFIG model

To reiterate, the angle θ in Fig. 2.8 refers to the angle dividing the stator A-axis and the DQ frame. Furthermore, the angle θ_m is the angle dividing the current vector i_a and rotor current vector (in green). The same θ cannot be used for both the stator and rotor as the angle between the rotor current vector (in green) and the DQ frame is $\beta = \theta - \theta_m$ [32]. Hence, this new β is used for the dq-transformation of rotor variables and is given by the following equation [32].

$$\beta = \int_0^t \omega_r(\gamma)d\gamma + \beta(0) \tag{2.46}$$

By this method, both the rotor and stator variables are brought into the d-q axis in this thesis.

2.8. DFIG: THE MODEL

With the background on the induction machine, the converter, and the wind turbine we can now lay out and assemble the mathematical state space model to be used in this thesis.

As explained in [17], the dynamics of the slip-ring induction machine used can be best explained using the following state space equations. The notations used are tabulated in Table 2.4.

$$\begin{aligned}
\dot{\psi}_{qs} &= v_{qs} - R_s i_{qs} - \omega_s \psi_{ds} \\
\dot{\psi}_{ds} &= v_{ds} - R_s i_{ds} + \omega_s \psi_{qs} \\
\dot{\psi}_{qr} &= v_{qr} - R_r i_{qr} - (\omega_s - \omega_r)\psi_{dr} \\
\dot{\psi}_{dr} &= v_{dr} + R_r i_{dr} + (\omega_s - \omega_r)\psi_{qr}
\end{aligned} \tag{2.47}$$

Here, the fluxes can be represented using the stator, rotor inductances and currents as shown below [17],[12].

$$
\begin{aligned}
\psi_{ds} &= L_s i_{ds} + L_m i_{dr} \\
\psi_{dr} &= L_r i_{dr} + L_m i_{ds} \\
\psi_{qs} &= L_s i_{qs} + L_m i_{qr} \\
\psi_{qr} &= L_r i_{qr} + L_m i_{qs}
\end{aligned}
\tag{2.48}
$$

Since, the dynamics of the DFIG mainly revolve around the stator and rotor currents, state space equations can be formulated from the above flux and voltage equations, as shown below [12]. Here, the state variables are the dq axes currents of both the stator and rotor. Here, $L_{eq} = L_s L_r - L_m^2$.

$$
\begin{aligned}
\dot{i}_{ds} &= \frac{1}{L_{eq}} \left\{ -R_s L_r i_{ds} + \left[\omega_r L_m^2 + \omega_s \left(L_{eq}\right)\right] i_{gs} + R_r L_m i_{dr} + \omega_r L_r L_m i_{qr} + L_r v_{ds} - L_m v_{dr} \right\} \\
\dot{i}_{qs} &= \frac{1}{L_{eq}} \left\{ -\left[\omega_r L_m^2 + \omega_s \left(L_{eq}\right)\right] i_{ds} - R_s L_r i_{qs} - \omega_r L_r L_m i_{dr} + R_r L_m i_{qr} + L_r v_{qs} - L_m v_{qr} \right\} \\
\dot{i}_{dr} &= \frac{1}{L_{eq}} \left\{ R_s L_m i_{ds} - \omega_r L_s L_m i_{qs} - R_r L_s i_{dr} + \left[\omega_s \left(L_{eq}\right) - \omega_r L_s L_r\right] i_{qr} - L_m v_{ds} + L_s v_{dr} \right\} \\
\dot{i}_{qr} &= \frac{1}{L_{eq}} \left\{ \omega_r L_s L_m i_{ds} + R_s L_m i_{qs} - \left[\omega_s \left(L_{eq}\right) - \omega_r L_s L_r\right] i_{dr} - R_r L_s i_{qr} - L_m v_{qs} + L_s v_{qr} \right\}
\end{aligned}
\tag{2.49}
$$

Furthermore, the stator and rotor power can be compiled together to get the total power being injected by the DFIG into the grid by the following measurement equations [12]. However, unlike [12], the powers do not need to be multiplied by the a factor of 3/2.

$$
\begin{aligned}
P &= \left(v_{ds} i_{ds} + v_{qs} i_{qs} + v_{dr} i_{dr} + v_{qr} i_{qr}\right) \\
Q &= \left(v_{qs} i_{ds} - v_{ds} i_{qs} + v_{qr} i_{dr} - v_{dr} i_{qr}\right)
\end{aligned}
\tag{2.50}
$$

Also, based on the discussion on the wind turbine in section 2.6, the mechanical dynamics of the wind turbine can be better explained using the following state space equations [12].

$$
\begin{aligned}
T_e &= L_m \left(i_{qs} i_{dr} - i_{ds} i_{qr}\right) \\
\dot{\omega}_r &= \frac{1}{2H} \left(T_m - T_e - D\omega_r\right)
\end{aligned}
\tag{2.51}
$$

Finally, the set of equations 2.49 and 2.50 together constitute the process model of the DFIG used in this thesis with equations in 2.49 as the state transition function and the equations in 2.50 as the measurement function.

3

THE KALMAN FILTER

Chapter 3 lays down a detailed explanation of the discretization techniques of the process models. This is followed by a subsequent introduction to a linear kalman filter. Finally, the modifications to this kalman filter to accomodate non-linear process models are discussed.

BY statistical definition, estimation can be called an approximation of any information pertaining to a particular population, especially when uncertainties are involved. However, if estimation could be interpreted in a more familiar way, it is a way of life. From judging the right proportion of ingredients while baking a cake, to budgeting the monthly expenses, estimation is an integral part of our lives used to achieve a specific, successful result - here, a delicious cake or to not overspend. In statistics, an estimate is obtained by approximating the properties of sampled cases to the whole population as in [33].

In the realm of signal processing and engineering, simple mathematical functions can be used as approximations to estimate real-world phenomenon. These estimates are then updated by comparing with real-world measurements, which may or may not contain noise. Power systems, in particular, have used approximate, linear power equations as a static state estimation model to estimate voltages and angles of buses which do not have measuring devices. Generally, such a state estimator employs a weighted-least-squares method which minimizes an error function with weights based on the accuracy of each measurement.

Although, this procedure provides a static estimate as discussed in chapter 1, we require a dynamic state estimator which can estimate the dynamic behaviour of the internal parameters of the generators in the power grid. Such an estimator is made feasible through the use of a Kalman Filter, which predicts the current state estimate by using the previous state estimate, and subsequently updates the current estimate based on the measurements. Hence, this chapter is particularly important for this thesis, in order to have an understanding of the working of a Kalman Filter.

3.1. ESTIMATION: MODEL DISCRETIZATION

Before moving on to the estimation and filtering process, it is important to point out that the kalman filter operates with measurements which are in discrete time-steps. Hence, any system model used in the prediction process of the filter needs to be of the discrete form. This brings out the necessity to convert the models discussed in the previous chapters from the continuous domain to the discrete domain.

Let the state-transition equation of a continuous process be given by equation 3.1, where w_c is the process noise,

$$\dot{x} = f_c(x, u) + w_c \tag{3.1}$$

and its output measurement equation given by equation 3.2, where v_c is the measurement noise. Process noise w_c and measurement noise v_c shall be discussed in detail in the Kalman Filter section.

$$y = h_c(x, u) + v_c \tag{3.2}$$

When these equations are discretized as in equation 3.3 [15], the state transition equation is a function of the states, inputs, and process noise of the previous time-step. Similarly, the output measurement equation is a function of the states, inputs, and measurement noise of the current time-step.

$$\begin{aligned} x_k &= f\left(x_{k-1}, u_{k-1}\right) + w_{k-1} \\ y_k &= h\left(x_k, u_k\right) + v_k \end{aligned} \tag{3.3}$$

In order to discretize, various numerical integration techniques are employed. Generally, these techniques are of two types - Euler-based and Runge-Kutta-based methods. These two methods are explained below as discussed in [6].

Euler Method

For a continuous function given by the equation 3.4 and graphically represented by Fig.3.1 [6],

$$\frac{dv}{dt} = f(v,t) \tag{3.4}$$

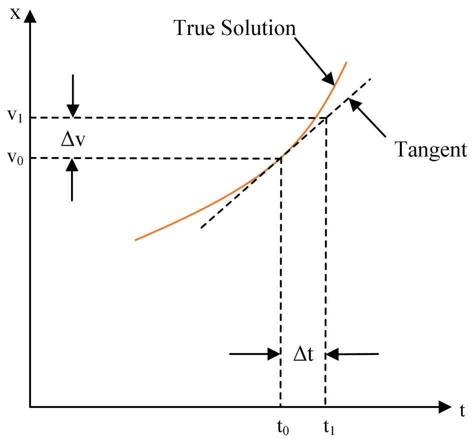


Figure 3.1: A continuous function of the first order - adapted from [6]

At $v = v_0$ and $t = t_0$, the tangent is used as an approximation, where it is represented by equation 3.5 [6],

$$\left.\frac{dv}{dt}\right|_{v=v_0} = f\left(v_0, t_0\right) \tag{3.5}$$

With this equation 3.5, any change in v i.e. Δv after a time step Δt, is given by 3.6 [6].

$$\Delta v = \left.\frac{dv}{dt}\right|_{v=v_0} \cdot \Delta t \tag{3.6}$$

By this principle, v_1 at $t = t_1$ is given by 3.7 [6].

$$v_1 = v_0 + \left.\frac{dv}{dt}\right|_{v=v_0} \cdot \Delta t \tag{3.7}$$

Similarly, a v_2 at $t = t_2 = t_1 + \Delta t$ can be given by 3.8 [6].

$$v_2 = v_1 + \left.\frac{dv}{dt}\right|_{v=v_1} \cdot \Delta t \tag{3.8}$$

This simple, first-order method of Euler's is successively applied to generate estimates at later time-steps. However, this method works only at smaller values of time-step to remain numerically stable. Modified versions of Euler's method involve predictor and corrector steps where average of derivatives is used to improve the accuracy. Although, this modification still requires time-steps of very low value.

Runge-Kutta Method

The Runge-Kutta methods are used as an approximation of Taylor Series. These methods are of two forms - Second order and Fourth order. Let us consider only the Fourth order method as it provides better accuracy. Accordingly, for determining v_{k+1} at $t = t_{k+1}$ from v_k at $t = t_k$ we use equation 3.9 [6].

$$v_{k+1} = v_k + \frac{1}{6}\left(j_1 + 2j_2 + 2j_3 + j_4\right) \tag{3.9}$$

with [6],

$$\begin{aligned}
j_1 &= f\left(v_k, t_k\right)\Delta t \\
j_2 &= f\left(v_k + \tfrac{j_1}{2}, t_k + \tfrac{\Delta t}{2}\right)\Delta t \\
j_3 &= f\left(v_k + \tfrac{j_2}{2}, t_k + \tfrac{\Delta t}{2}\right)\Delta t \\
j_4 &= f\left(v_k + j_3, t_k + \Delta t\right)\Delta t
\end{aligned} \tag{3.10}$$

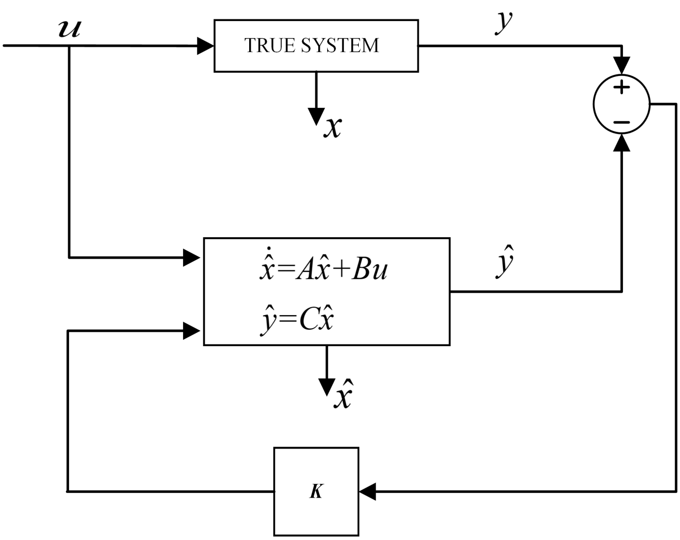
The interpretation of each j parameter can be understood from [6].

3.2. ESTIMATION: THE LINEAR KALMAN FILTER

The Kalman Filter is a technique to optimally approximate internal parameters of a system that cannot be directly measured. It is optimal because it finds the most optimal estimate based on information provided from several measurements, with or without noise. In this section, the principle behind a Kalman Filter is explained based on the MathWorks series on Kalman Filters in [34].

State Observers

The concept of observing certain internal and immeasurable parameters of the system is called state observation and this function is performed by a state observer. Fig. 3.2 shows a general state observer.



STATE OBSERVER

Figure 3.2: A State Observer - adapted from [34]

Any real-world system can be represented as a model which takes an input to produce an output. The model alone is nothing but a set of functions dependent on state variables. State variables help establish the link between the input state and the output state, so as to represent the true behaviour of the system. These variables usually cannot be measured and are of major interest to study and/or operate the system. Hence, state observers are an observation scheme that help in estimating the system's internal states.

They do so when the original system inputs are fed to a mathematical process model which approximates the behaviour of the true system. For a given input, the mathematical process model computes the estimated state variables and produces the estimated output. This estimated output, however, may differ from the true output due to uncertainties. These uncertainties are due to presence of the process and the measurement noise given by w_k and v_k respectively as shown in the discrete, true system model of 3.11 [34].

$$x_k = Ax_{k-1} + Bu_k + w_k$$
$$y_k = Cx_k + v_k \tag{3.11}$$

In order to eliminate this difference, this estimated output is compared with the measured output of the true system, to produce an error. This error is then used to update the mathematical model and its state variables through a feedback loop gain as shown in figure 3.2. This computed error dynamic can be given through the equation 3.12 [34]. Here, the term $(A - KC)$, and in turn K, helps in controlling the rate of decay of the error.

$$\dot{e}_{obs} = (A - KC)e_{obs} \tag{3.12}$$

Kalman Filter

Kalman filter is a well known stochastic state observer which observes the state of the system through the Kalman gain K. With the provision of measurements at discrete time steps, the kalman filter calculates the kalman gain based on these noises present in the system. Normal distributions with a mean of zero and covariances of Q and R can be used to depict w and v respectively [34].

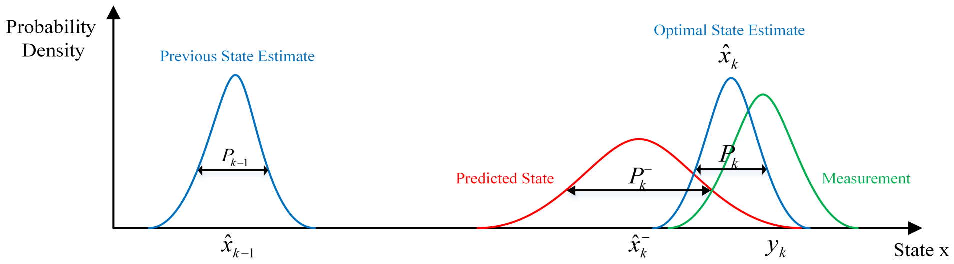


Figure 3.3: Kalman filter principle through probability distributions - adapted [34]

The algorithm of the Kalman filter can be explained as follows with the help of the probability distribution in Fig. 3.3. The process is split into two stages - the prediction stage and the update stage.

- Prediction stage: In this step, the Kalman Filter has access to the previous state estimate $\hat{x}_{k-1}$ along with the previous state covariance matrix P_{k-1} (shown in blue). With these values, current predicted state estimate $\hat{x}_k^-$ (shown in red) also know as a priori state estimate is computed along with the a priori covariance matrix P_k^-, as shown in 3.13 [34].

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k$$
$$P_k^- = AP_{k-1}A^T + Q \tag{3.13}$$

- Update stage: Now, the Kalman Filter has to update the a priori state estimate based on the measurements y_k available (shown in green). This update is made possible through the computation of the Kalman gain K given by 3.14 [34].

$$K_k = \frac{P_k^- C^T}{C P_k^- C^T + R} \tag{3.14}$$

With the help of this gain calculation, a posteriori state estimate $\hat{x}_k$ and a posteriori covariance matrix P_k is calculated (shown in blue) as in 3.15 [34] to obtain the optimal state estimate.

$$\begin{aligned} \hat{x}_k &= \hat{x}_k^- + K_k \left(y_k - C\hat{x}_k^- \right) \\ P_k &= \left(I - K_k C \right) P_k^- \end{aligned} \tag{3.15}$$

These two steps are performed at every time-step k based on the values obtained in the time step $k-1$. However, the major drawback of the kalman filter is that it is limited to only linear systems.

3.3. ESTIMATION: NON-LINEAR SYSTEMS

Since, many real-world problems are non-linear and the Kalman Filter does not work on non-linear processes, modifications are made to the kalman filter to adapt to a non-linear process model. One famous modification is the EKF. By linearizing the non-linear model through the use of jacobians, EKF enables estimation of non-linear phenomenon. However, this method tends to produce erroneous estimations when the non-linearity in the process is high. Similarly, there are many versions of the kalman filter such as the UKF, Particle Filter, and so on. In this thesis, the UKF algorithm is employed as it is computationally less demanding.

Unscented Kalman Filter

The UKF explained below is inspired from [35]. The idea of the unscented transformation is to approximate the probability distributions of the state rather than the non-linear model itself. It does so by generating $2n+1$ sigma points which statistically represent the state distributions of the previous state estimate [35]. The generated sigma points $\hat{x}_{k-1}^{(i)}$ are transformed through the state transition equation to obtain $\hat{x}_k^{(i)}$ as in equation 3.16 [35].

$$\hat{x}_k^{(i)} = f\left(\hat{x}_{k-1}^{(i)}, u_{k-1} \right) \tag{3.16}$$

The transformed sigma points help in computing the a priori state estimate which is given by the mean of all these points, as shown in equation 3.17 [35].

$$\hat{x}_k^- = \frac{1}{2n} \sum_{i=1}^{2n} \hat{x}_k^{(i)} \tag{3.17}$$

Similarly, the a priori covariance state matrix is given by equation 3.18 [35].

$$P_k^- = \frac{1}{2n}\sum_{i=1}^{2n}\left(\hat{x}_k^{(i)} - \hat{x}_k^-\right)\left(\hat{x}_k^{(i)} - \hat{x}_k^-\right)^T + Q_{k-1} \tag{3.18}$$

The $\hat{x}_k^{(i)}$ as in equation 3.16 is now transformed through the output equation to give $\hat{y}_k^{(i)}$ and estimated output $\hat{y}_k$ as in equation 3.19 [35].

$$\hat{y}_k^{(i)} = h\left(\hat{x}_k^{(i)}, u_k\right); \hat{y}_k = \frac{1}{2n}\sum_{i=1}^{2n}\hat{y}_k^{(i)} \tag{3.19}$$

The covariance matrices of the estimated output and cross-covariance matrix between $\hat{x}_k^{(i)}$ and $\hat{y}_k^{(i)}$ are computed as in equation 3.20 [35].

$$\begin{aligned} P_y^- &= \tfrac{1}{2n}\textstyle\sum_{i=1}^{2n}\left(\hat{y}_k^{(i)} - \hat{y}_k\right)\left(\hat{y}_k^{(i)} - \hat{y}_k\right)^T + R_k \\ P_{xy}^- &= \tfrac{1}{2n}\textstyle\sum_{i=1}^{2n}\left(\hat{x}_k^{(i)} - \hat{x}_k^-\right)\left(\hat{y}_k^{(i)} - \hat{y}_k\right)^T \end{aligned} \tag{3.20}$$

Now, with the available covariance matrices, the kalman gain K_k and subsequent state posteriori update $\hat{x}_k$ and covariance P_k is calculated as in equation 3.21 [35].

$$\begin{aligned} K_k &= P_{xy}P_y^{-1} \\ \hat{x}_k &= \hat{x}_k^- + K_k\left(y_k - \hat{y}_k\right) \\ P_k &= P_k^- - \left(K_k P_y K_k^T\right) \end{aligned} \tag{3.21}$$

Thus, the UKF described above uses the machine models discussed in the previous chapters as process models to perform dynamic state estimation in this thesis. One specific disadvantage of the UKF is of numerical instability. Numerical stability can be described as a property which explains the propagation of error, and can be seen in high dimensional state space models [36]. However, the problem of numerical instability is not a hindrance here, and hence, is not addressed in this thesis.

4

Online Estimator

Chapter 4 explains the major contributions of this thesis. To be specific, the DQ transformations that are necessary for the DFIG, the online-DSE algorithm, and the procedure for determining sensitivity relationships are explained in this chapter.

The dynamic modelling of the synchronous generator and the DFIG as state space equations are discussed in Chapter 2. Also, the formulation of a Kalman Filter, for the purpose of DSE is discussed in Chapter 3. Following, it is necessary to realize how these components can be put together to estimate in a near real-time setting. Hence, in this chapter, the major contribution of this thesis, i.e. online DSE algorithm featuring near real-time estimation is explained.

4.1. DQ TRANSFORMATIONS FOR A DFIG

Before we dive into the near real-time algorithm, we need to discuss the transformations necessary to bring the DFIG variables into the DQ axes. This is because, the DFIG variables being measured exist in the ABC domain, however, the DSE requires them to be in the DQ domain. Hence, before being sent to be estimated, these variables need to be transformed as explained in the theory of Section 2.7. Furthermore, the converter switches of the DFIG induce harmonics, which can affect the state estimator's performance. This has to be resolved using a low-pass filter.

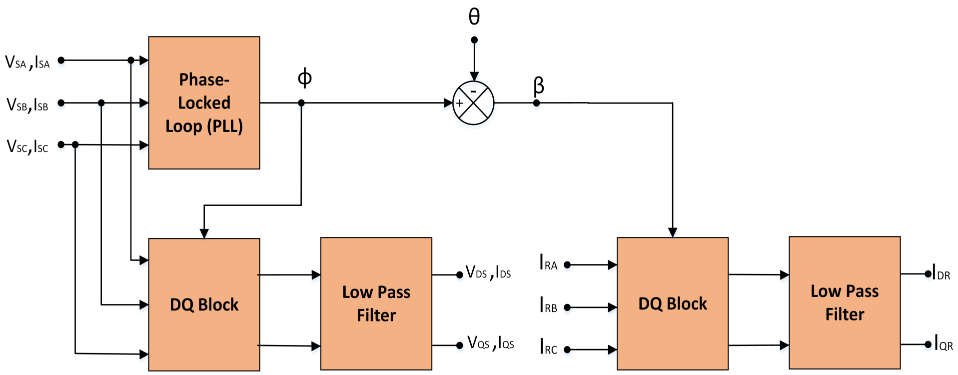


Figure 4.1: The Transformation of the Stator and Rotor variables into the DQ axes

The block diagram showing the DQ transformations that need to be done are shown in Fig. 4.1. The stator voltages and currents in the ABC domain, i.e. V_{SA}, V_{SB}, V_{SC}, I_{SA}, I_{SB}, and I_{SC} respectively, are fed to a Phase-Locked Loop (PLL). A PLL is a control technique which is used to extract the reference phase of the inputted signals. The reference phase for the stator signals is denoted by ϕ. This reference ϕ, along with the the stator ABC voltage and current signals are passed to the DQ Block to obtain the DQ stator voltages and currents i.e. V_{DS}, V_{QS}, I_{DS}, and I_{QS} respectively. The DQ block is used to convert a signal from the ABC domain to the DQ one. It is the exact equivalent of the Park's transformation which is shown in equation 2.44.

As explained in Section 2.7, the reference ϕ is the angle between the stator and the DQ-axis. Similarly, the angle between the rotor and the stator is the rotor angle θ. Therefore, to determine the angle between the rotor, i.e. β and the DQ-axis, the difference between the reference ϕ and the rotor angle θ is calculated. The Rotor current signals, i.e. I_{RA}, I_{RB}, I_{RC}, along with the angle β, are fed to another DQ block to obtain the rotor currents I_{DR} and I_{QR}. Finally, all the transformed DQ signals are passed through a low-pass butterworth filter, with an optimal cut-off frequency of 5 Hz, to block the inherent harmonics. The cut-off frequency is called optimal because at 5 Hz, all high frequency harmonics were eliminated as well as no considerable dynamic information was lost in the process of filtering.

However, unlike the stator voltage, the ABC rotor voltages V_{RA}, V_{RB}, and V_{RC}, cannot be passed to a DQ block to obtain the DQ rotor voltages, i.e. V_{DR} and V_{QR}. This is because the reference angle of the rotor voltages is dependant on the rotor frequency, and in turn the wind speed. Hence, the rotor voltages are analytically determined using the following equations. Here, R_R is the rotor resistance, s is the per unit slip, λ_{DR} and λ_{QR} are the DQ rotor fluxes.

4

$$\begin{aligned} V_{DR} &= R_R I_{DR} - s\lambda_{QR} \\ V_{QR} &= R_R I_{QR} + s\lambda_{DR} \end{aligned} \tag{4.1}$$

4.2. ASPECTS OF ONLINE DSE

Several aspects need to be considered before setting up the online algorithm. These are discussed below.

- **Delay definitions**
  When system states are to be estimated in real-time, there is delay d involved. To better understand this delay d, it can be broken down into two components, i.e. telecommunication delay d_c and estimation delay d_e.

  $$d = d_c + d_e \tag{4.2}$$

  This telecommunication delay d_c can be defined as the total time taken to transfer a bit of information from the PMU point to the online estimator point. Whereas, the estimation delay d_e can be defined as the total time taken to simply estimate the system state. During this thesis work, it was observed that the telecommunication delay is of the order of milliseconds and the estimation delay is of the order of microseconds. Hence, the total delay is dominantly equal to the telecommunication delay. However, the test to measure the delays was conducted in a lab environment, leading to smaller delay values. The delays can be larger in a real-world application.

  $$\begin{aligned} d_c &>>> d_e \\ d &\approx d_c \end{aligned} \tag{4.3}$$

- **Sample-wise or Batch-wise computation?**
  For online estimation, the estimation can be performed in two ways - sample-wise or

batch-wise. In the sample-wise approach, the system states are estimated for each single time-step using the corresponding measurement at that time step. Whereas in the batch-wise approach, the estimation is performed by collecting measurements for B time-steps and then estimating the corresponding system states for those time-steps. In a real-world application of the DSE, the grid being observed is of large-scale, with large number of rotating generators, and larger number of observed states. Hence, the amount of data being measured by the PMUs in the entire grid is vast. In such a case, the choice between the sample-wise and batch-wise approaches is based on the processing power of the available resources and other computational tasks to be run. The advantage of using the sample-wise approach is that every estimation run involves computation of only one sample and not for a collection of samples. This significantly reduces the processing power required in every estimation run. Moreover, if there are no other computational tasks to be computed apart from estimation, the available computational resources required is significantly less and sample-wise approach is, therefore, economical. However, it is common to have many other tasks that need computation along with estimation. In such a case, a sample-wise approach continuously occupies the computational resources and makes running other processes infeasible. In such a case, a batch-wise approach is more favourable. While collecting the measurements over a series of time-steps, the batch-wise approach can provide adequate computational time to other processes that need to be run. Once, enough measurements have been collected, the batch of measurements are processed all at once. However, a clear disadvantage of the batch-wise approach is that estimation is delayed due to the collection of measurements and is comparatively away from real-time. Hence, when it is more important to have the least amount of estimation delay or when the available resources are limited, the sample-wise approach is preferred. If there are other tasks that need to be run along with estimation, the batch-wise approach is preferred.

- **Preprocessing of PMU measurements**
  The PMU phasor measurements that arrive at the estimator can be set to be of the rectangular form or of the polar form depending on the PMU/PDC configuration. In our case, it was set to the polar form, which comprises of phasor magnitude and phasor angle. However, in order to make these magnitudes and angles suitable for estimation, certain preprocessing steps need to be undertaken. These are given below:

  – The **PMU phasor angles** are given with respect to the GPS clock signal. However, for estimation, the phasors need to be referenced with respect to a slack or reference bus phasor. Hence, every phasor angle θ_i needs to be subtracted from the slack phasor angle θ_s as shown below.

$$\theta_i = \theta_i - \theta_s \tag{4.4}$$

  – The **PMU magnitudes** need to be converted to the per unit form. The voltage magnitudes are transformed into per unit form as shown below.

$$V_{pu} = \frac{V_{mag} * \sqrt{3}}{V_{base} * 1000} \tag{4.5}$$

Similarly, the current magnitudes are transformed into per unit form as shown below.

$$I_{pu} = \frac{I_{mag} * V_{base}}{S_{base} * 1000} \tag{4.6}$$

Here, V_{base} and S_{base} are the chosen base voltages in kV and the chosen base apparent power in MVA. Consequently, the processed voltage and current phasors are multiplied together to obtain active and reactive powers in per unit.

– If the incoming phasor data contains **NaNs** or Not a Number, these missing values have to be treated. This is done by replacing the NaNs with the last known value in the previous time steps.

4.3. ONLINE DSE ALGORITHM

There are two types of power system control architectures that are used in a typical DSE. These are the centralized and the de-centralized control architectures respectively. In the centralized control architecture, all the geographically located PMUs send their measurements to a central location. The dynamic estimation of the entire grid is performed here. However, in the decentralized one, the grid is divided into many regions, which can include even substations. The states in each region are estimated by their own DSE which uses the locally available PMU measurements. Finally, all the regional DSEs send their local estimates to a central location, where the state of the entire grid is formed. In this thesis, the centralized control is followed, i.e. all the measurements are collected at a central point and the estimation of the entire system is computed as a whole. Therefore, the algorithm for estimation using the batch-wise approach for a power system with centralized control is explained below. The algorithm can also be easily toggled between the sample and batch approaches by setting $B = 1$ and $B > 1$ respectively.

- **Step 1:** This step indicates the beginning of the algorithm and contains the initialization of the parameters. These are the initial error covariance state matrix P, the process noise covariance Q, and measurement noise covariance R. Along with these parameters, the initial state matrix is provided for the UKF. The initial state matrix is populated with steady state values from a power flow calculation.

- **Step 2:** In this step, the algorithm waits and checks if 'B' amount of PMU data samples have arrived. Here, B is the size of the batch to be estimated. For large values of B, the algorithm waits for long time durations to collect this 'B' amount of data, leading to delayed estimation. Hence to avoid a very large delay, the batch size B is set at values which are a fraction of the chosen PMU sample rate F_R.

- **Step 3:** Once the batch of PMU measurements are collected, these measurements are pre-processed to make them suitable for estimation. As explained before in Section 4.2, the pre-processing steps include referencing of phasor angles with respect to a slack reference, per unit conversion of phasor magnitudes, and treatment of NaNs.

- **Step 4:** In this step, the processed batch of PMU measurements are used to estimate the system states using the UKF explained in Section 3.3. In an iterative manner for each timestep in the batch, the state-space models discussed in Section 2.3 and Section 2.8 are used to compute the a-priori estimates for all the synchronous generators and DFIG wind farms in the grid respectively. Subsequently, the processed PMU batch measurements are used to update the a-priori estimates to obtain the a-posteriori estimates and the new error covariance state matrix P. These a-posteriori estimates are stored or visually displayed to the TSO.

- **Step 5:** Finally, the state estimate of the last time-step of the batch is set as the new initial state. Similarly, the latest error covariance array P is updated as the new P. The algorithm, now, returns to Step 2 to wait for the next batch of PMU data.

4.4. SENSITIVITY RELATIONSHIPS

A sensitivity analysis helps in understanding how the estimation of the dynamic states is influenced by various input errors introduced. Hence, a sensitivity analysis was conducted as a part of the assessment of the online DSE, and the sensitivity relationships between the estimation error and the various input errors were determined. Such relationships help in understanding how resilient the estimation is against presence of any particular type of error. Also for any given input error, such relationships can help interpolate or extrapolate the estimation error. The procedure for determining the sensitivity relationship is given below, for when errors are introduced in a particular input.

- A particular state is chosen to be observed. Various errors are introduced in the input and the corresponding errors in the estimation of the chosen state is recorded.

- The errors introduced and the recorded estimation error are tabulated and plotted to reveal the nature of the relationship between the input error and the state error.

- Through the use of a curve fitting functionality, a trend line is fitted on the above plotted line. The equation of the trend line provides us the sensitivity relationship between the estimation error and the input error introduced.

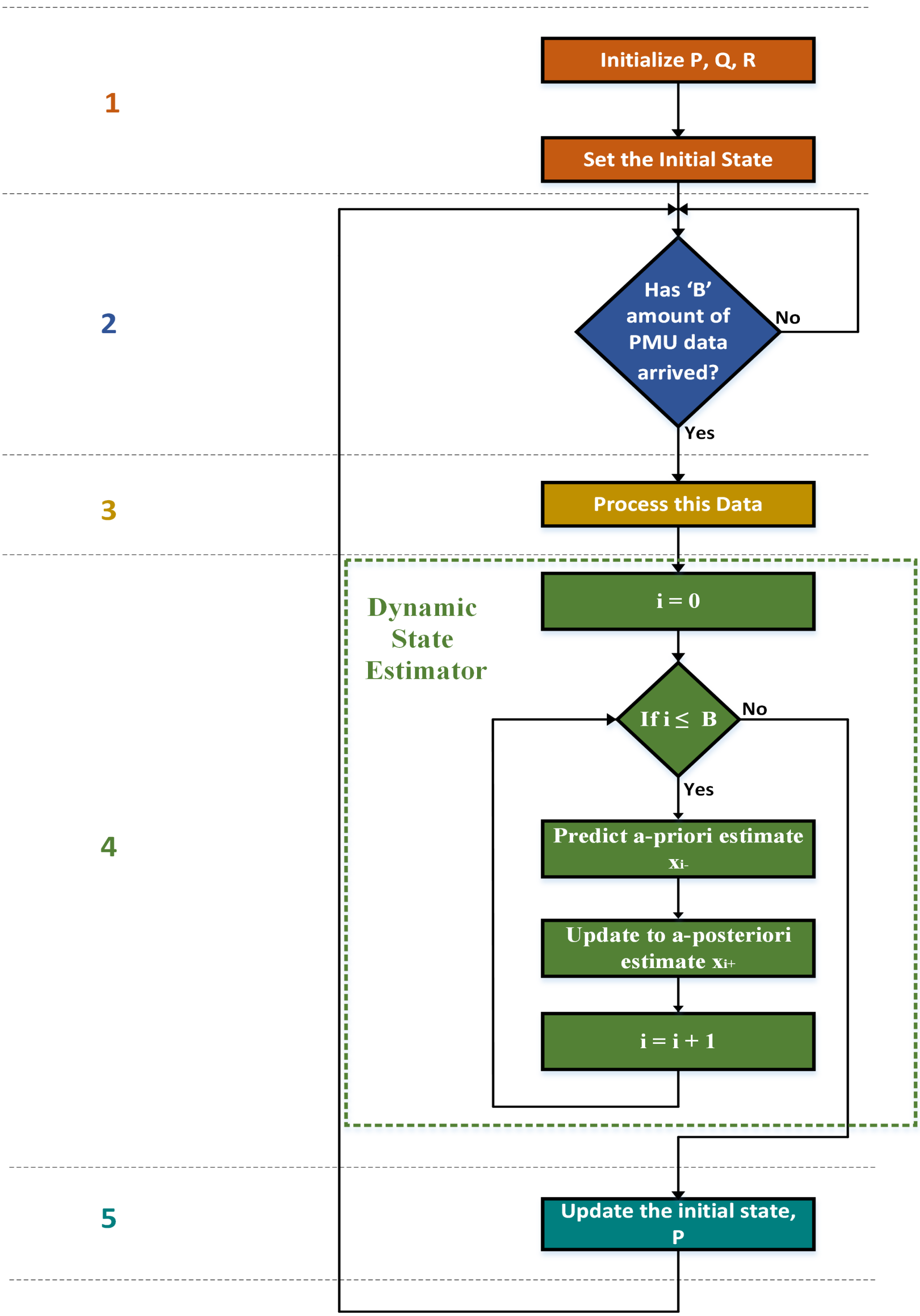


Figure 4.2: The flow-chart of the Online DSE algorithm

5

A Case Study: Simulation and Results

Chapter 5 explains the simulation platform, the case study used in this thesis, the results of the online estimator, and the final sensitivity analysis.

ONCE, the online DSE is built, it is necessary to validate its results and to comment on its performance. In this chapter, the cyber-physical setup which is used to validate the online DSE is explained. This is followed by an insight into the case study used - IEEE 14-bus system. The results obtained from this case study simulation are given and interpreted. Finally, a sensitivity study is conducted and its results are used to develop relationships between inputted error and estimation error.

5.1. SIMULATION PLATFORM

The cyber-physical platform used in the simulation is developed mainly using the Real Time Digital Simulator (RTDS) and the Synchrophasor Application Development Framework (SADF), as in [37]. This is better explained using the block diagram shown in Fig.5.1.

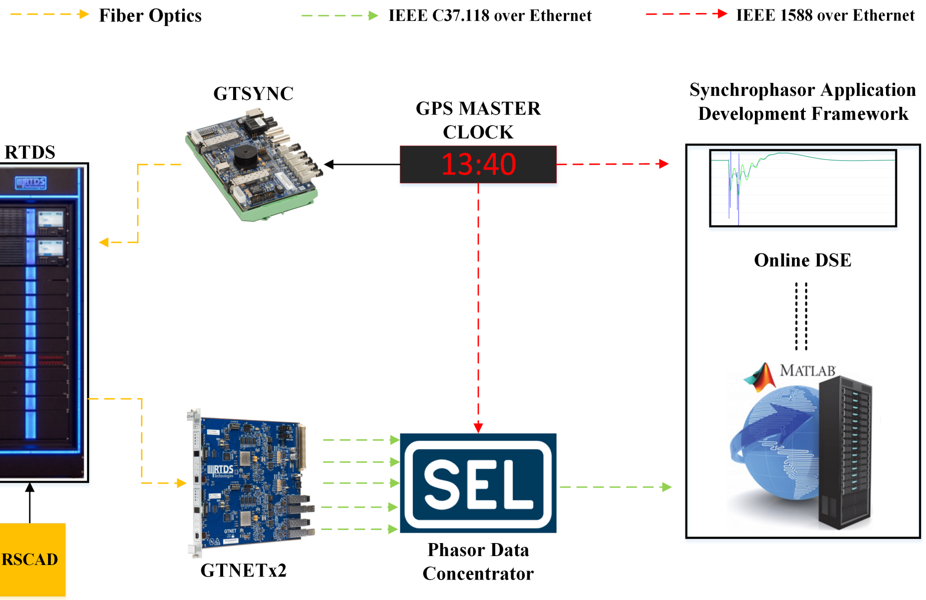


Figure 5.1: The cyber physical platform used to simulate the real-time DSE - adapted from [37]

The RTDS is a tool to simulate real-time, transient and dynamic power system phenomena. It is mainly used for In-the-loop testing of protection and control mechanisms [38]. It provides flexibility through adjustable grid settings and a variety of disturbances to choose from. The RSCAD software acts as the interface between the user and the RTDS system by providing a graphical environment to simulate grid phenomena [37]. The software comprises mainly a DRAFT tool and RUNTIME tool. The DRAFT tool contains a drawing layout to draft the test system to be simulated. This is facilitated through the provision of several libraries (eg.Power System library, Controls library, etc.) which allow the user to drag and drop system components (e.g. Buses, Lines, PMUs, etc.) onto the layout. The RUNTIME tool, on the other hand, provides an environment for real time simulation of the compiled DRAFT case. It includes plots, meters, etc. to visualize the real-time response of the system and also includes sliders, buttons, etc. for real-time control. The RSCAD software also provides a scripting functionality for automating the whole runtime process.

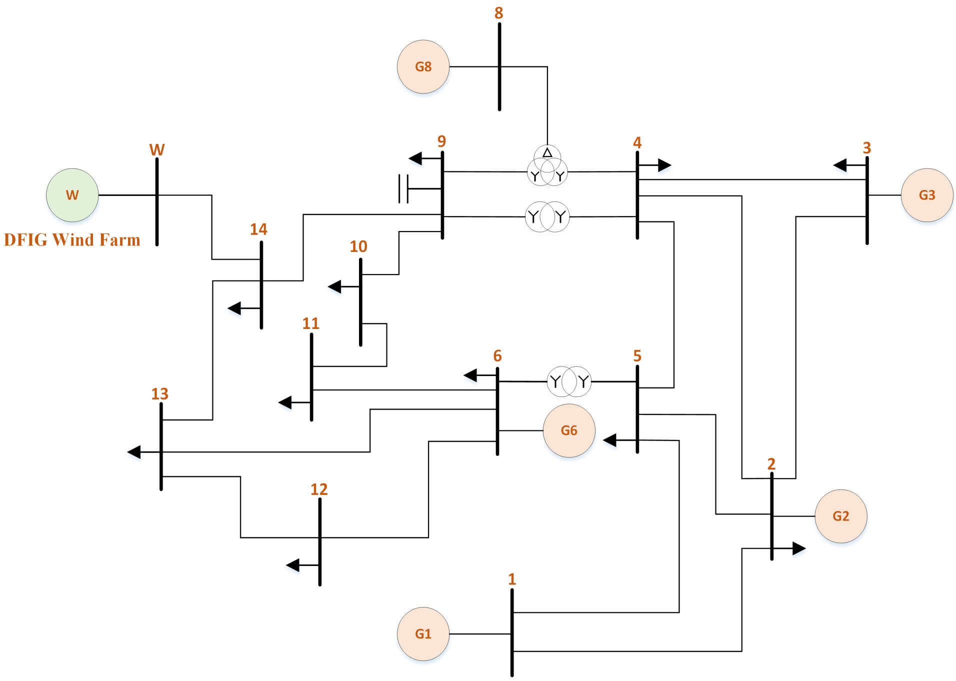


Figure 5.2: The IEEE 14-bus test system used in RSCAD - adapted from [39]

The GTNETx2 cards of the RTDS are mainly deployed when it is necessary to exchange data between RTDS and external system [40]. As in our case, the GTNETx2 cards are used to stream PMU data to an external device using the IEEE C37.118.2 protocol [40]. Generally, the GTNETx2 card can accomodate up to 8 PMUs with sampling rates ranging from 1 to 250 samples per second depending on the nominal system frequency. The GT-SYNC card is used to synchronize the RTDS simulation and other external devices with an external time reference, possibly from a GNSS Master clock as displayed in Fig.5.1 [41]. Hence, the PMU data streams leaving the RTDS, through the GTNETx2, are timestamped with respect to this clock.

A Phasor Data Concentrator (PDC) such as SEL-5073 PDC developed by Schweitzer Engineering Laboratories is used to collect and align various PMU data streams according to the measurement time, and forward this as a combined data stream to hierarchical higher level PDCs or applications such as the SADF [37]. The communication network between the GTNETx2 PMUs and the SEL-5073 PDC can be simulated using a Wide-Area Network (WAN) Simulator; This is explained in detail in [37]. The Synchrophasor Application Development Framework or SADF, developed by Naglic et al in [42] enables sim-

Parameter	Value
System Frequency f	60 Hz
Base MVA for G1	615 MVA
Base MVA for G2	60 MVA
Base MVA for G3	60 MVA
Base MVA for G4	25 MVA
Base MVA for G5	25 MVA
Base MVA for DFIG	22 MVA
System Base Voltage	18 kV

Table 5.1: Grid parameters necessary for pre-processing of PMU phasor data

Parameter	Value
Inertia Constant H	3.41
Synchronous Reactance: d-axis X_d	1.7241
Synchronous Reactance: q-axis X_q	1.6587
Transient Reactance: d-axis X_{dt}	0.2586
Transient Reactance: q-axis X_{qt}	0.4524
Open circuit time-constants: d-axis T_{do}	3.826
Open circuit time-constants: q-axis T_{qo}	0.5084
Damping Factor K_d	0

Table 5.2: Model parameters for synchronous generator model in the DSE

plified design of user-defined Wide-Area Monitoring, Protection and Control schemes, such as the DSE, by interfacing real-time PMU data from a PDC to the MATLAB programming environment. Thus, the RTDS, GTNETx2, GTSYNC, PDC, and SADF, together constitute the cyber-physical testbed to validate the online-DSE algorithm given in Section 4.3.

5.2. CASE STUDY

Real Time Digital Simulator

The power system used to test the developed online DSE is the IEEE 14-bus modified benchmark system as shown in Fig.5.2. The modification is the addition of a Wind Farm, comprising 10 DFIG Wind Turbines with a terminal bus W, to bus 14 of the IEEE 14-Bus. The single line diagram was drafted in RSCAD and was compiled to run an initial loadflow. This loadflow provides the initial conditions for the RUNTIME simulation. However, the DFIG Wind Farm model provided in RSCAD is built using the small timestep functionality and does not participate in this initial loadflow. In order to solve this, a "Loadflow-Source" block was added parallel to the DFIG, at bus 14. By doing so, the Loadflow-Source block emulates the DFIG in the initial loadflow calculation but does not participate in the RUNTIME simulation.

Parameter	Value
Synchronous Speed w_s	1
Stator Inductance L_s	4.45
Rotor Inductance L_r	4.459
Magnetizing Inductance L_m	4.358
Stator Resistance R_s	0.00462
Rotor Resistance R_r	0.006
Inertia Constant H_w	1.5
Damping Factor D	0

Table 5.3: Model parameters for DFIG model in the DSE

Phasor Measurement Units and Phasor Data Concentrator

The PMUs of P class were placed on bus terminals of 1, 2, 3, 6, 8, and 14 respectively. The sample rate for all the PMUs were set to the highest possible rate, i.e. 240 samples per second. This enables more samples provided to the DSE, leading to more accurate estimation. The PMUs are configured to send the terminal bus measurements, i.e. positive sequence voltage phasor and positive sequence current phasor. Along with the measurements, the DSE requires inputs for estimation. To be particular, the synchronous machine model requires the input data of the mechanical torque and the field voltage for estimation. Hence, the PMUs are also configured to carry these analog input signals. The inputs to the SEL-5073 PDC were configured to receive the incoming PMU streams from GTNETx2. Similarly, the output of SEL-5073 PDC was also configured to send the desired streams to SADF.

5

Online DSE

The batch-wise online DSE was implemented in the MATLAB programming environment.

- For the initialization process, the initial error covariance array P was set to a variance of 10^{-4}. Similarly, the process noise array Q and the measurement noise array R were both set to a variance of 10^{-6}.
- As discussed in Section 4.3, the batch size B was chosen to be a fraction of the PMU sample rate. Since, the chosen sample rate is 240 samples per second, the batch size was selected to be 60 samples, i.e. one-fourth the sample rate.
- In the online DSE algorithm, one of the crucial steps is the pre-processing of the PMU phasor magnitudes and angles. To do so, certain grid parameters have to be defined. These are given in Table 5.1.
- The state-space models of both the synchronous generator and the DFIG require the model parameters to be initialised. These are given in Table 5.2 and Table 5.3 for the synchronous generator model and the DFIG model respectively.
- The IEEE 14-bus grid under study consisted of 5 synchronous generators and 1 DFIG wind farm. The number states being observed for each synchronous generator and

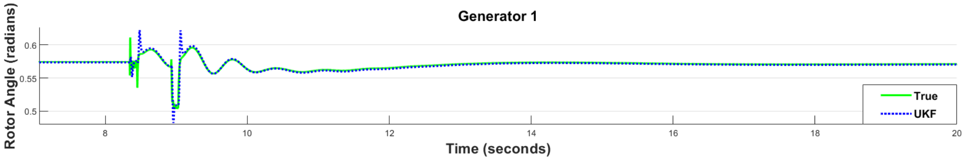
each DFIG were 4 and 5 respectively. Also, the number of measurements used to observe each synchronous generator and each DFIG were 3 and 2 respectively. Hence, the complete discrete state-space model of the grid, constructed in a decentralized manner, consisted of 25 states and 17 measurements, in total.

5.3. RESULTS

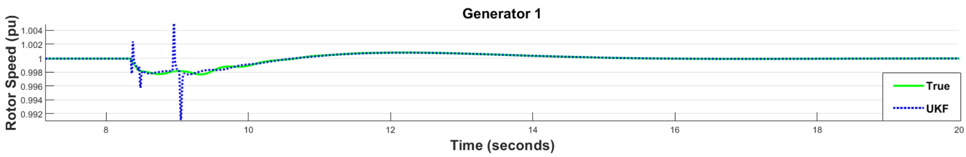
Once the simulation platform and the settings were set-up as in Section 5.1 and 5.2 respectively, the online DSE was validated using a three-phase short circuit fault. The fault was introduced at $t = 8.3s$ and was subsequently cleared after 100 milliseconds. During the period of this disturbance, the estimated waveforms of the state variables were compared with the true waveforms to gauge how well the DSE performs, qualitatively as well as quantitatively.

Synchronous Generators

The estimated waveforms are plotted over the true waveforms for the rotor angle and



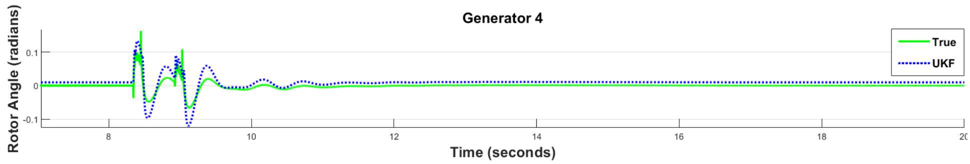
(a)



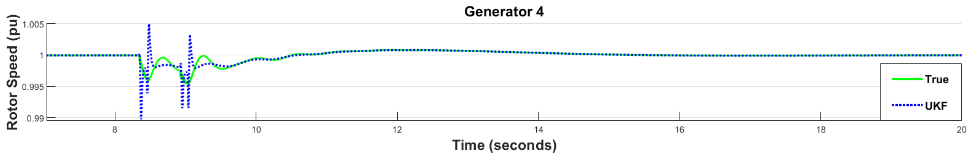
(b)

Figure 5.3: Estimation waveform against the true waveform for rotor angle (a) and rotor speed (b) of generator 1

rotor speed of the synchronous generator 1 as shown in Fig. 5.3. For the pre-fault and post-fault periods, the estimated steady state values match very well with the true ones for both rotor angle and rotor speed. During the fault period i.e. $t = 8.3s$ to $t = 16s$, the estimated signals are almost dynamically identical with the true signals. However, unlike the true signal, there are a few overshoots and undershoots in the estimated signal at a few points. This is due to the introduction of the terminal frequency measurement which also contains these anomalies. However, the overall estimation of the dynamic behaviour of rotor angle and rotor speed of generator 1 is more than sufficient and these few anomalies can be overlooked if necessary. However, for certain control implementations it becomes necessary to mitigate such anomalies. In such cases, filtering techniques can be employed to remove these sharp overshoots and undershoots.



(a)



(b)

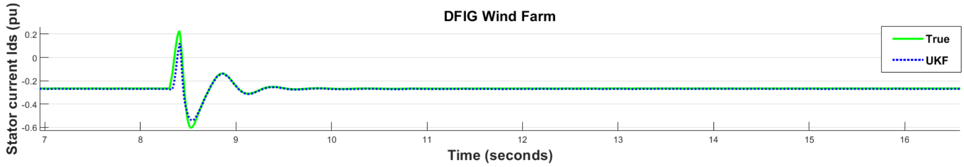
Figure 5.4: Estimation waveform against the true waveform for rotor angle (a) and rotor speed (b) of generator 4

The estimated waveforms against the true waveforms for the rotor angle and rotor speed of the synchronous generator 4 as shown in Fig. 5.4. Similar to the estimation results of generator 1 as in Fig. 5.3, the estimation of both the rotor angle and rotor speed of the generator 4 is a good representation of the true dynamics. Similar to generator 1, the estimation of generator 4 also suffers from the presence overshoots and undershoots due to the frequency measurement. This can be again overlooked. However for generator 4, it can be seen that there is a small error between the estimation and truth in both the steady-state and fault periods. This can be because the generator 1 is the slack generator, i.e. of the largest capacity, and the remaining generators are of smaller capacity. Consequently, there might be a slight difference in the dynamic behaviour between the slack generator and the other smaller generators. The estimation plots of the remaining generators tell the same story and are given in the Appendix.

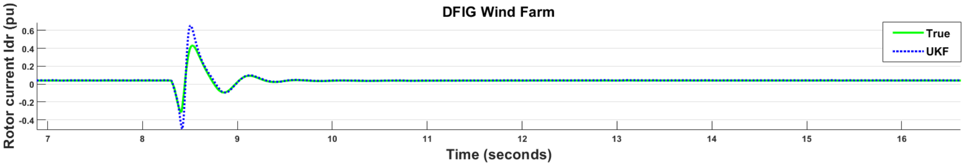
Doubly-Fed Induction Generator Wind Farm

The estimated waveforms are plotted over the true waveforms for the d-axis stator current and d-axis rotor current of the DFIG as shown in Fig. 5.5. It can be observed that the estimator captures the dynamic behaviour of the DFIG very well, during all the periods. Unlike the synchronous generator, there are no overshoots and undershoots as there is no frequency measurement used in the state-space model. At the positive and negative peaks, there is an observable mismatch. This can be attributed to a model being only an approximation of the true dynamic phenomena. However, this mismatch is not significant, and hence can be overlooked. The plots of the remaining states are provided in the Appendix.

To summarize, the results provide sufficient evidence that the online DSE is adequately capable of estimating the general dynamic behaviour of all the synchronous generators and DFIG wind farms in the grid.



(a)



(b)

Figure 5.5: Estimation waveform against the true waveform for d-axis stator current (a) and d-axis rotor current (b) of the DFIG wind farm

Performance Evaluation

In order to quantify the performance of the online DSE, the total error in estimation is given using the following equation [13].

$$E = \frac{1}{n}\sum_{j=1}^{n}\left|x_e^j - x_t^j\right| \tag{5.1}$$

Here, n is the total number of states being observed, x_e is the estimated state vector, and x_t is the true state vector. The simulation was run for 10 seconds with a three-phase short circuit fault at $t = 3s$, which is cleared after 100 milliseconds. The total error in estimation was found to be $E = 0.003$. This order of error is very well acceptable, as the estimation is still an adequate representation of the overall dynamics.

The telecommunication delay d_c and the estimation delay d_e of the online-DSE when it runs in a near-real time can also be measured. The telecommunication delay d_c is as defined in section 4.2 and, in our case, also includes the delay from SADF. The delay-measurement procedure was as follows. The host computer of the online DSE was connected to the RTDS via a LAN connection. A LAN connection helps in minimizing the delay in the telecommunication network. Subsequently, a data bit was sent from the RTDS to the estimator and the time instances at the sending endpoint and the receiving endpoint were recorded. Also, the time difference between these two points are computed. In total 7 trials were conducted to obtain 7 time differences. The median of these 7 time differences provided the telecommunication delay d_c as shown in Table 5.4. Similarly, predefined functions in the MATLAB programming environment were used to compute the time taken to estimate a batch of PMU measurements. Like in the previous procedure, 7 batches were estimated to obtain 7 delays. The median of these 7 values provided the estimation delay as shown in Table 5.4. The median was preferred over the mean as the latter gets biased due to outliers.

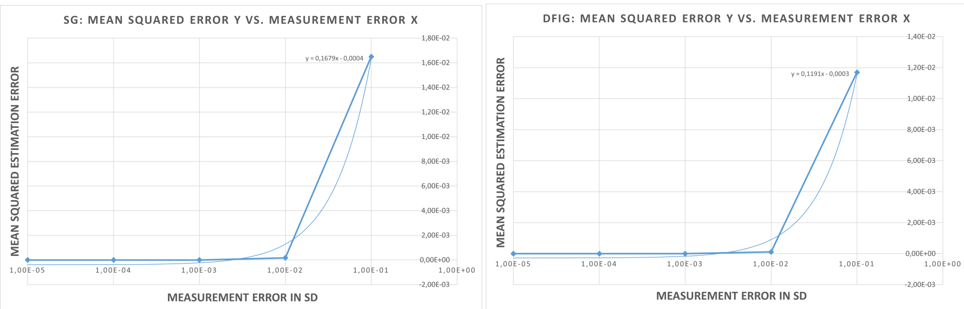
Trial	Telecommunication Delay	Estimation Delay
1	0.003	0.000025
2	0.005	0.000023
3	0.008	0.000024
4	0.0045	0.000025
5	0.002	0.000027
6	0.003	0.000025
7	0.004	0.000023
Median	**4 ms**	**25 μs**

Table 5.4: The 7 trials conducted to measure the telecommunication delay d_c and the estimation delay d_e

5.4. SENSITIVITY ANALYSIS

A sensitivity analysis was conducted to quantify the variation in the state estimates when variable input errors are introduced. The relationships were found for sensitivities to different types of errors in measurements, parameters, wind speed, and initial state.

Measurement Errors



(a) (b)

Figure 5.6: Variation of state error against measurement errors for synchronous generator (a) and DFIG respectively (b)

To gauge the sensitivity against measurement errors, a particular state of the synchronous generator and the DFIG model was chosen. This was the rotor angle of generator 1 and the d-axis stator current of the DFIG respectively. Errors in terms of standard deviation SD were introduced in the measurements, and the corresponding mean squared errors in the chosen states were recorded. These Mean squared errors for the synchronous generator and the DFIG were plotted against corresponding logarithmic measurement errors as shown in Fig. 5.6. For both the machine models, it can be observed that the relationship between the state error and measurement error is linear in nature. This means that for large measurement errors, the state error is linearly larger. Hence, it becomes significantly important to have fairly accurate PMU measurements to have good estimates. A mathematical relationship for the synchronous generator and the DFIG measurement sensitivities can be determined by performing a linear fit on the curves shown in Fig. 5.6. These equations are shown below, where y is the mean squared estimation error and x is the measurement error in terms of standard deviation.

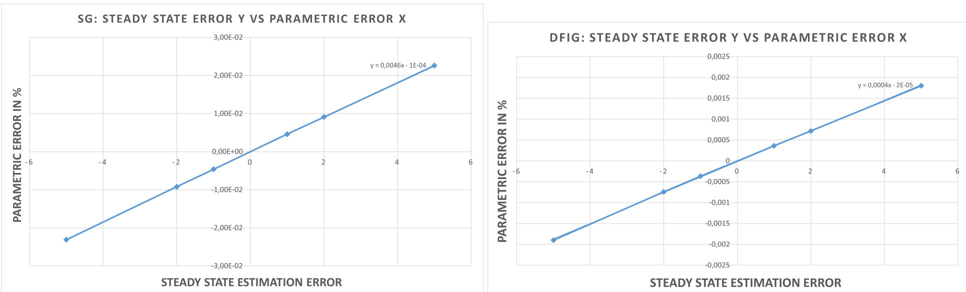
Synchronous generator:

$$y = 0.1679x - 0.0037 \tag{5.2}$$

DFIG:

$$y = 0.1191x - 0.0032 \tag{5.3}$$

Parametric Errors



(a) (b)

Figure 5.7: Variation of state error against parametric errors for synchronous generator (a) and DFIG respectively (b)

Similar to sensitivity analysis of measurement errors, a particular state of the synchronous generator and the DFIG was chosen. Positive and negative percentage errors were introduced and the corresponding variation in the steady-state errors of the observed state was recorded and plotted as in Fig. 5.7. For both the machine models, it can be observed that the relationship between the state error and parametric error is linear in nature. For the introduction of the largest parametric error of 5 percent, the state error is still significantly small. However, beyond this point errors in the state can become unacceptable. The state errors for larger parametric errors can be determined from the mathematical relationship derived between the input parametric errors x and the state errors y. This was done by performing a linear fit on the variation of state errors as shown in Fig. 5.7. The linear relationship equations are as given below.

Synchronous generator:

$$y = 0.0046x - 0.00331 \tag{5.4}$$

DFIG:

$$y = 0.0004x - 0.002902 \tag{5.5}$$

Wind Speed Errors

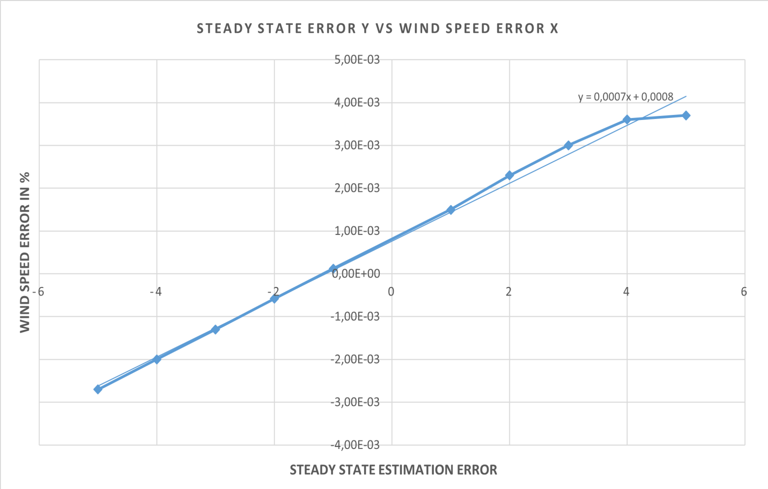


Figure 5.8: The variation of state error of the DFIG for different wind speed errors

5

A sensitivity analysis of the DFIG against errors in wind speed was also conducted. The steady state errors in the d-axis stator current was observed for different percentage errors in the wind speed. It can be observed that this relationship is again linear in nature as in 5.8. As the absolute percentage of error increases, there is an increase in the absolute state error recorded. Until 5 percent of wind speed errors, the recorded state errors are small and, hence, acceptable. Beyond this point, the state errors can be determined through the mathematical relationship between the input error and state error. This is determined by performing a linear fit on the variation in 5.8 and is shown below, with y being the steady state estimation error and x being the wind speed error introduced in the form of percentages.

$$y = 0.0007x - 0.0037 \tag{5.6}$$

Initial State Errors

The chosen state of the synchronous generator and the DFIG were tested against various initial state errors as in Fig. 5.9 and Fig. 5.10 respectively. it can be seen that the state estimator is robust against any given initial state error. However, the larger the initial state error is, the longer it takes to settle at the steady state value.

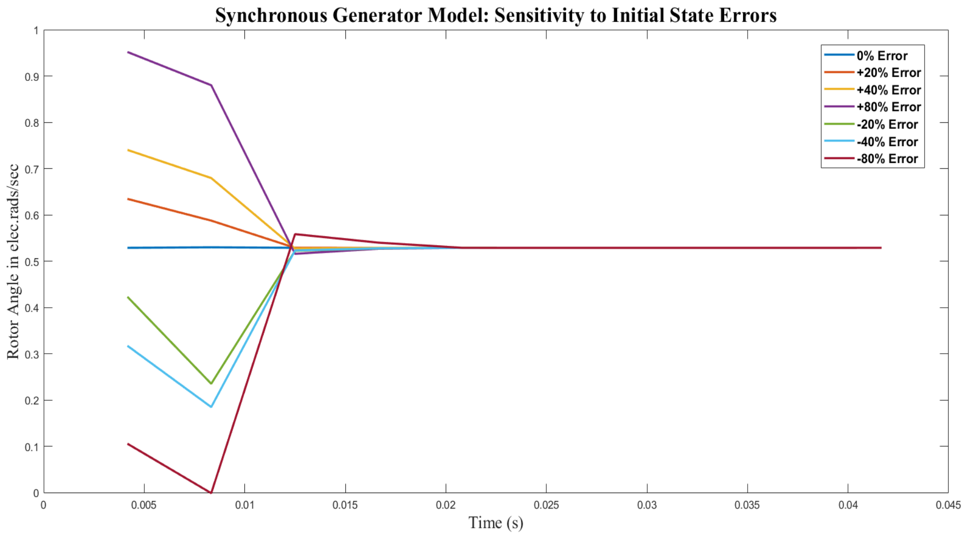


Figure 5.9: Variation of state error against initial state errors for synchronous generator

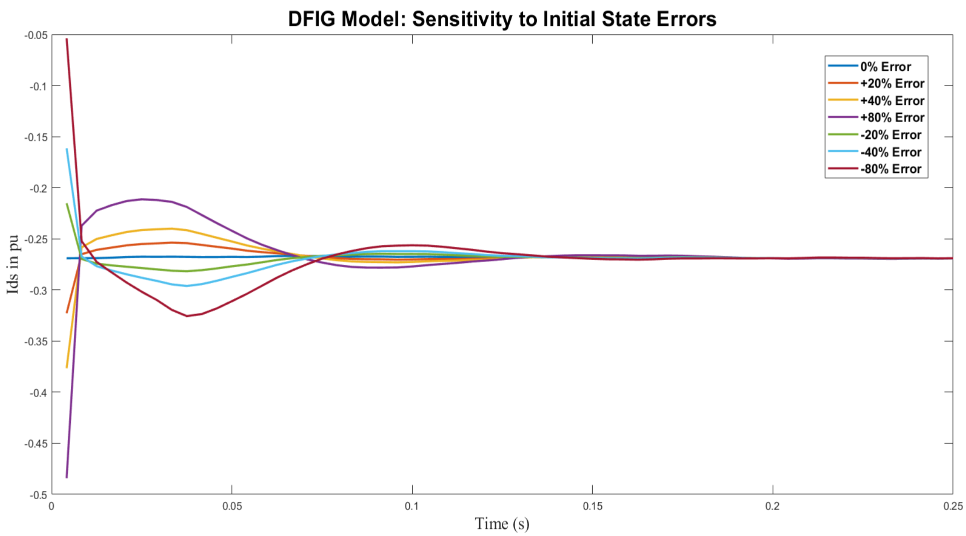


Figure 5.10: Variation of state error against initial state errors for DFIG

6

CONCLUSION

Chapter 6 concludes the thesis by explaining the scientific contributions of the research findings. Also, the future scope in this line of research is discussed.

This thesis work was performed with the objective to build an online DSE which can estimate states in a near real-time manner. To do so, this thesis was structured in the following manner. In Chapter 1, the research setting and goals were addressed. In chapter 2, the models of the synchronous generator and the DFIG were understood. In Chapter 3, the principle of the estimation process was explained using the kalman filter. In Chapter 4, the machine models and the kalman filter were combined to enable an online DSE algorithm, which can estimate in near-real time. In chapter 5, the simulation setting was explained and the subsequent results of the online DSE were explained. Now, the thesis is concluded by reflecting on the implications of the research work done by also providing any possible future improvements in this line of work.

"What are the transformations, for the DFIG, that need to be performed to bring measurements from the ABC domain to the DQ domain?"

In order to enable online DSE, the DFIG state inputs, i.e. stator and rotor voltages and currents, were needed to be transformed from the ABC domain to the DQ domain. Furthermore, a low-pass filter was required to remove any harmonic components that could travel to the SADF platform and interfere with the estimation. This thesis work provides the procedure for these necessary transformations, which serves as a guide for not only DSE, but other monitoring and control applications which require DFIG quantities in the DQ realm.

"How to develop an online dynamic state estimator for power systems with a centralized control architecture?"

In order to perform dynamic state estimation in an online fashion, this thesis provides the online DSE algorithm which estimates the states, of a power system with centralized control, in near real-time. The developed online DSE was found to be adequate in terms of capturing the dynamics of all the rotating machines in the grid. The online DSE produced estimates with an acceptable accuracy, and with an estimation delay in the order of a few microseconds and a telecommunication delay in the order of a few milliseconds. This work comes closest to being a good example of how a dynamic state estimator can estimate in the real world scenario. This is of keen interest to TSOs in the aspect of power system monitoring.

"What is the sensitivity relationship between the input error introduced and the estimation error for synchronous generator model and DFIG model?"

Another contribution to this thesis is towards the sensitivities of DSE. The sensitivity relationships between the inputted error and the estimation error for the synchronous generator and DFIG models of the DSE were derived. The measurement errors, parametric errors and wind speed errors were found to have a linear relation with the state error. From the slopes of the sensitivity relationship equations, we can understand that both the synchronous and DFIG models are most sensitive to measurement errors. Furthermore, such relationships help in interpolating and extrapolating the estimation error for any given input error. By also deriving such sensitivity relations for other Kalman filter variants, comparative analyses on the slopes, Area-Under-the-Curve (AUC), etc. can help better understand which kalman filter is more suitable.

"Future Improvements?"

Some important future improvements to the thesis are explained below.

Large Volumes of Data: As we move towards large-scale system with numerous measuring devices, the volume of data that needs to be processed by the online DSE increases dramatically. This can have drastic consequences on the computational power and time of the DSE. One solution could be to use the estimate only when a dynamic event occurs. This indeed saves computational energy, but limits certain functionalities. For example, the estimates provided by the DSE supports many EMS functions. One such function is anomaly detection, where anomalies such as dynamic events are detected from a continuous stream of data. In such a case, where the DSE is required to provide continuous estimates, this solution does not seem appealing. Therefore, the next step for large-scale implementations is through parallel and distributed computational techniques. By employing multiple processors and dividing the computation task amongst them, parallel computation seems to be a viable solution. However, a distributed form of computation, where the estimation is performed locally and in a distributed manner, and then collected together at a central point, seems to be most appealing.

Unknown Inputs: In this thesis implementation, the state estimator requires inputs to be fed in along with the measurements. For example, the dynamic estimation of the synchronous generator requires real time data of mechanical torque and field voltage. However, implementations such as in [16] have shown that dynamic estimation, using EKF, without the knowledge of the inputs is possible. This is of high importance to a TSO, as some of the inputs such as mechanical torque, in reality, that are required by the model are difficult to access through measuring devices. Hence, estimation methods which encourage no knowledge of these inputs are more desirable and becomes a future extension of this thesis.

Dynamic Security Assessment: As a part of the EMS, the online DSE becomes a supporting tool to other EMS functionalities. Mainly, the online DSE can provide near real-time, dynamic estimates as initial conditions to the Dynamic Security Assessment (DSA) algorithm [10]. This can greatly improve the DSA which currently uses only static estimates as initial conditions. Similarly, by extending the online DSE to incorporate parameter estimation algorithms, online parameter calibration of grid-connected machines can be easily performed.

A

APPENDIX

A.1. ONLINE DSE CODE

```
%% Online Plotter
function Online_Plotter()
    % Initialize Global variables
    global DATA demo
    global init_state P Q R x_estimate
    if ~isfield(demo,'window')
        figure('units', 'normalized', 'outerposition', [0 0 1 1],
             'name','Dynamic State Estimator for the Control Room of the
             Future','NumberTitle','off');
        demo.window = 60;
        demo.phasor = 2;
        demo.voltage = 1;
        demo.current = 2;
        demo.torque = 1;
        demo.field = 2;
        demo.processed = 0;
    end
    % Initial state for the first run
    if DATA.index_max < 2
        init_state = [0.57;0;1.0189;0.2396;
        0.333;0;1.0189;0.2396;
        0;0;1.0189;0.2396;
        0;0;1.0189;0.2396;
        0;0;1.0189;0.2396;
        0;0;0;0;1.2;0];
    end
    % Has new PMU data arrived?
```

A

```
if DATA.index_max > demo.processed
    demo.processed = DATA.index_max;
    clf;
    hold on;
    % Extract the PMU data from DATA struct
    if DATA.index_max <= demo.window
        demo.dataset = [DATA.TimeStamp(1:DATA.index_max,1)
            DATA.Magnitude(1:DATA.index_max,demo.phasor,1)];
        demo.timestamp = DATA.TimeStamp(1:DATA.index_max,1);
        demo.voltage_data =
            DATA.Magnitude(1:DATA.index_max,demo.voltage,:);
        demo.vang_data = DATA.Angle(1:DATA.index_max,demo.voltage,:);
        demo.current_data =
            DATA.Magnitude(1:DATA.index_max,demo.current,:);
        demo.cang_data = DATA.Angle(1:DATA.index_max,demo.current,:);
        demo.torque_data =
            DATA.Analog(1:DATA.index_max,demo.torque,:);
        demo.field_data = DATA.Analog(1:DATA.index_max,demo.field,:);
        demo.freq = DATA.Freq(1:DATA.index_max,1,:);
        demo.dfig = DATA.Analog(1:DATA.index_max,:,6:8);
    else
        demo.dataset = [DATA.TimeStamp(DATA.index_max-
        demo.window:DATA.index_max,1)
        DATA.Magnitude(DATA.index_max-
        demo.window:DATA.index_max,demo.phasor,1)];
        demo.timestamp = DATA.TimeStamp(DATA.index_max-
        demo.window:DATA.index_max,1,:);
        demo.voltage_data = DATA.Magnitude(DATA.index_max-
        demo.window:DATA.index_max,demo.voltage,:);
        demo.vang_data = DATA.Angle(DATA.index_max-
        demo.window:DATA.index_max,demo.voltage,:);
        demo.current_data = DATA.Magnitude(DATA.index_max-
        demo.window:DATA.index_max,demo.current,:);
        demo.cang_data = DATA.Angle(DATA.index_max-
        demo.window:DATA.index_max,demo.current,:);
        demo.torque_data = DATA.Analog(DATA.index_max-
        demo.window:DATA.index_max,demo.torque,:);
        demo.field_data = DATA.Analog(DATA.index_max-
        demo.window:DATA.index_max,demo.field,:);
        demo.freq = DATA.Freq(DATA.index_max-
        demo.window:DATA.index_max,1,:);
        demo.dfig = DATA.Analog(DATA.index_max-
        demo.window:DATA.index_max,:,6:8);
    end
    % Send the data to the estimator to receive estimates
    [x_estimate,P_final] = DSE_online(demo,init_state,P,Q,R);
    % Update the initial state and the error covariance matrix P
    init_state = x_estimate(:,end);
    P = P_final;
```

```
% Extracting each state
ra1 = x_estimate(1,:);
ra2 = x_estimate(5,:);
ra3 = x_estimate(9,:);
ra4 = x_estimate(13,:);
ra5 = x_estimate(17,:);
rs1 = 1+x_estimate(2,:);
rs2 = 1+x_estimate(6,:);
rs3 = 1+x_estimate(10,:);
rs4 = 1+x_estimate(14,:);
rs5 = 1+x_estimate(18,:);
ids = x_estimate(21,:);
iqs = x_estimate(22,:);
idr = x_estimate(23,:);
iqr = x_estimate(24,:);
wr = x_estimate(25,:);

% plot the estimates
title_plot = "IEEE-14 Bus System";
subplot(1,2,1)
plot(demo.timestamp, ra1);hold on;
xlabel('Timestamp (UTC)'); ylabel('Rotor Angle|G1|pu');
title('Generator Monitoring');
legend('G1','G2','G3','G4','G5');
ax = gca;
ax.XTick = demo.timestamp;
grid on
datetick('x', 'dd-mm-yyyy HH:MM:SS.FFF', 'keeplimits',
    'keepticks')
set(gca, 'XMinorTick', 'on', 'YMinorTick', 'on',
    'XTickLabelRotation', 45)
set(findall(gca, 'Type', 'Line'),'LineWidth',3);

subplot(1,2,2)
plot(demo.timestamp, ids);hold on;
xlabel('Timestamp (UTC)'); ylabel('D-axis Stator
    Current|DFIG|pu');
title('DFIG Monitoring');
legend('G1','G2','G3','G4','G5');
ax = gca;
ax.XTick = demo.timestamp;
sgtitle(['Online Monitoring: ' title_plot ])
grid on
datetick('x', 'dd-mm-yyyy HH:MM:SS.FFF', 'keeplimits',
    'keepticks')
set(gca, 'XMinorTick', 'on', 'YMinorTick', 'on',
    'XTickLabelRotation', 45)
set(findall(gca, 'Type', 'Line'),'LineWidth',3);
```

A

```
        drawnow limitrate
        disp([datestr(now, 'yyyy-mm-dd HH:MM:SS.FFF') ' PLOT ']);
    end
end
%% Retrieving Measurements
function [Tm,Tm2,Tm3,Tm4,Tm5,Ef,Ef2,Ef3,Ef4,Ef5,F_G1,F_G2,F_G3,
F_G4,F_G5,F_G6,Vmag_pu,Vmag_pu2,Vmag_pu3,Vmag_pu4,Vmag_pu5,P_m,
Q_m,P_m2,Q_m2,P_m3,Q_m3,P_m4,Q_m4,P_m5,Q_m5,vds,vqs,vdr,vqr,
P_DFIG,Q_DFIG] = measurements_online(demo)
    global Sbase Sbase2 Sbase3 Sbase4 Sbase5 Vbase

    %Inputs to G1
    Tm = demo.torque_data(:,1,1);        % Mechanical Torque
    Tm = Tm(~isnan(Tm));
    Ef = demo.field_data(:,1,1);         % Internal Field Voltage
    Ef = Ef(~isnan(Ef));

    %Inputs to G2
    Tm2 = demo.torque_data(:,1,2);       % Mechanical Torque
    Tm2 = Tm2(~isnan(Tm2));
    Ef2 = demo.field_data(:,1,2);        % Internal Field Voltage
    Ef2 = Ef2(~isnan(Ef2));

    %Inputs to G3
    Tm3 = demo.torque_data(:,1,3);       % Mechanical Torque
    Tm3 = Tm3(~isnan(Tm3));
    Ef3 = demo.field_data(:,1,3);        % Internal Field Voltage
    Ef3 = Ef3(~isnan(Ef3));

    %Inputs to G4
    Tm4 = demo.torque_data(:,1,4);       % Mechanical Torque
    Tm4 = Tm4(~isnan(Tm4));
    Ef4 = demo.field_data(:,1,4);        % Internal Field Voltage
    Ef4 = Ef4(~isnan(Ef4));

    %Inputs to G5
    Tm5 = demo.torque_data(:,1,5);       % Mechanical Torque
    Tm5 = Tm5(~isnan(Tm5));
    Ef5 = demo.field_data(:,1,5);        % Internal Field Voltage
    Ef5 = Ef5(~isnan(Ef5));

    %Inputs to DFIG
    vds = demo.dfig(:,1,1);         % Stator Voltage d-axis
    vds = vds(~isnan(vds));
    vqs = demo.dfig(:,2,1);         % Stator Voltage q-axis
    vqs = vqs(~isnan(vqs));
    vdr = demo.dfig(:,3,1);         % Rotor Voltage d-axis
    vdr = vdr(~isnan(vdr));
    vqr = demo.dfig(:,4,1);         % Rotor Voltage q-axis
```

A

```
vqr = vqr(~isnan(vqr));
P_DFIG = demo.dfig(:,3,2);          % DFIG active power injected
P_DFIG = P_DFIG(~isnan(P_DFIG))./2.2;
Q_DFIG = demo.dfig(:,4,2);          % DFIG reactive power injected
Q_DFIG = Q_DFIG(~isnan(Q_DFIG))./2.2;
%%

% System Frequency
F = demo.freq;
F_G1 = F(:,:,1);
F_G1 = F_G1(~isnan(F_G1));
F_G2 = F(:,:,2);
F_G2 = F_G2(~isnan(F_G2));
F_G3 = F(:,:,3);
F_G3 = F_G3(~isnan(F_G3));
F_G4 = F(:,:,4);
F_G4 = F_G4(~isnan(F_G4));
F_G5 = F(:,:,5);
F_G5 = F_G5(~isnan(F_G5));
F_G6 = F(:,:,6);
F_G6 = F_G6(~isnan(F_G6));
%%

%Measurements of G1
Vmag = demo.voltage_data(:,1,1);
Vmag = Vmag(~isnan(Vmag));
Vang = demo.vang_data(:,1,1);
Vang = Vang(~isnan(Vang));
Imag = demo.current_data(:,1,1);
Imag = Imag(~isnan(Imag));
Iang = demo.cang_data(:,1,1);
Iang = Iang(~isnan(Iang));

%Measurements of G2
Vmag2 = demo.voltage_data(:,1,2);
Vmag2 = Vmag2(~isnan(Vmag2));
Vang2 = demo.vang_data(:,1,2);
Vang2 = Vang2(~isnan(Vang2));
Imag2 = demo.current_data(:,1,2);
Imag2 = Imag2(~isnan(Imag2));
Iang2 = demo.cang_data(:,1,2);
Iang2 = Iang2(~isnan(Iang2));

%Measurements of G3
Vmag3 = demo.voltage_data(:,1,3);
Vmag3 = Vmag3(~isnan(Vmag3));
Vang3 = demo.vang_data(:,1,3);
Vang3 = Vang3(~isnan(Vang3));
Imag3 = demo.current_data(:,1,3);
```

```
Imag3 = Imag3(~isnan(Imag3));
Iang3 = demo.cang_data(:,1,3);
Iang3 = Iang3(~isnan(Iang3));

%Measurements of G4
Vmag4 = demo.voltage_data(:,1,4);
Vmag4 = Vmag4(~isnan(Vmag4));
Vang4 = demo.vang_data(:,1,4);
Vang4 = Vang4(~isnan(Vang4));
Imag4 = demo.current_data(:,1,4);
Imag4 = Imag4(~isnan(Imag4));
Iang4 = demo.cang_data(:,1,4);
Iang4 = Iang4(~isnan(Iang4));

%Measurements of G5
Vmag5 = demo.voltage_data(:,1,5);
Vmag5 = Vmag5(~isnan(Vmag5));
Vang5 = demo.vang_data(:,1,5);
Vang5 = Vang5(~isnan(Vang5));
Imag5 = demo.current_data(:,1,5);
Imag5 = Imag5(~isnan(Imag5));
Iang5 = demo.cang_data(:,1,5);
Iang5 = Iang5(~isnan(Iang5));

%% Treatment of the Measurement Data

% Generator 1
% Conversion to pu
Vmag_pu = (Vmag*sqrt(3))/(Vbase*1000);
Imag_pu = (Imag*Vbase)/(Sbase*1000);
% Changing phasor angle reference
Ref_ang = Vang;
Iang = -(Ref_ang - Iang);
Vang = Vang.*0;
% Deriving P and Q
V_m = Vmag_pu.*cos(Vang*pi/180)+Vmag_pu.*sin(Vang*pi/180)*1i;
I_m = Imag_pu.*cos(Iang*pi/180)+Imag_pu.*sin(Iang*pi/180)*1i;
S = V_m.*conj(I_m)*sqrt(3);
P_m = real(S);
Q_m = imag(S);

% Generator 2
% Conversion to pu
Vmag_pu2 = (Vmag2*sqrt(3))/(Vbase*1000);
Imag_pu2 = (Imag2*Vbase)/(Sbase2*1000);
% Changing phasor angle reference
Vang2 = -(Ref_ang - Vang2);
Iang2 = -(Ref_ang - Iang2);
% Deriving P and Q
```

A

```
    V_m2 = Vmag_pu2.*cos(Vang2*pi/180)+Vmag_pu2.*sin(Vang2*pi/180)*1i;
    I_m2 = Imag_pu2.*cos(Iang2*pi/180)+Imag_pu2.*sin(Iang2*pi/180)*1i;
    S2 = V_m2.*conj(I_m2)*sqrt(3);
    P_m2 = real(S2);
    Q_m2 = imag(S2);

    % Generator 3
    % Conversion to pu
    Vmag_pu3 = (Vmag3*sqrt(3))/(Vbase*1000);
    Imag_pu3 = (Imag3*Vbase)/(Sbase3*1000);
    % Changing phasor angle reference
    Vang3 = -(Ref_ang - Vang3);
    Iang3 = -(Ref_ang - Iang3);
    % Deriving P and Q
    V_m3 = Vmag_pu3.*cos(Vang3*pi/180)+Vmag_pu3.*sin(Vang3*pi/180)*1i;
    I_m3 = Imag_pu3.*cos(Iang3*pi/180)+Imag_pu3.*sin(Iang3*pi/180)*1i;
    S3 = V_m3.*conj(I_m3)*sqrt(3);
    P_m3 = real(S3);
    Q_m3 = imag(S3);

    % Generator 4
    % Conversion to pu
    Vmag_pu4 = (Vmag4*sqrt(3))/(Vbase*1000);
    Imag_pu4 = (Imag4*Vbase)/(Sbase4*1000);
    % Changing phasor angle reference
    Vang4 = -(Ref_ang - Vang4);
    Iang4 = -(Ref_ang - Iang4);
    % Deriving P and Q
    V_m4 = Vmag_pu4.*cos(Vang4*pi/180)+Vmag_pu4.*sin(Vang4*pi/180)*1i;
    I_m4 = Imag_pu4.*cos(Iang4*pi/180)+Imag_pu4.*sin(Iang4*pi/180)*1i;
    S4 = V_m4.*conj(I_m4)*sqrt(3);
    P_m4 = real(S4);
    Q_m4 = imag(S4);

    % Generator 5
    % Conversion to pu
    Vmag_pu5 = (Vmag5*sqrt(3))/(Vbase*1000);
    Imag_pu5 = (Imag5*Vbase)/(Sbase5*1000);
    % Changing phasor angle reference
    Vang5 = -(Ref_ang - Vang5);
    Iang5 = -(Ref_ang - Iang5);
    % Deriving P and Q
    V_m5 = Vmag_pu5.*cos(Vang5*pi/180)+Vmag_pu5.*sin(Vang5*pi/180)*1i;
    I_m5 = Imag_pu5.*cos(Iang5*pi/180)+Imag_pu5.*sin(Iang5*pi/180)*1i;
    S5 = V_m5.*conj(I_m5)*sqrt(3);
    P_m5 = real(S5);
    Q_m5 = imag(S5);

function [x_estimate,P_final] = DSE_online(demo,init_state,init_P,Q,R)
```

A

```
global w_syn n H Xd Xq Xdt Xqt Td0 Tq0 Kd w ws_w Ls_w Lr_w Lm_w Rs_w
    Rr_w H_w
%% Retrieving and synthesizing measurements
[Tm,Tm2,Tm3,Tm4,Tm5,Ef,Ef2,Ef3,Ef4,Ef5,F_G1,F_G2,F_G3,F_G4,F_G5,
F_G6,Vmag_pu,Vmag_pu2,Vmag_pu3,Vmag_pu4,Vmag_pu5,P_m,Q_m,P_m2,
Q_m2,P_m3,Q_m3,P_m4,Q_m4,P_m5,Q_m5,vds,vqs,vdr,vqr,P_DFIG,Q_DFIG]
= measurements_online(demo);

%Estimated state:
ns=4*n+6*w;                                     % Number of states

%Sigma points
Td0k = repmat(Td0,1,2*ns);
Tq0k = repmat(Tq0,1,2*ns);
H_rep=repmat(H,1,2*ns);
Kd_rep=repmat(Kd,1,2*ns);

%Simulation variables
t_step=1/240;
k_end=length(Vmag_pu);
x_hat=zeros(4*n+6*w,k_end);
x_hat(:,1) = init_state;
P = init_P;
%% The Unscented Kalman Filter Algorithm
for k=2:k_end

    %Generate Sigma Points from the previous state
    A=chol(ns*P);
    x_tilda=[A, -A];
    x_sigma=repmat(x_hat(:,k-1),1,2*ns)+x_tilda;

    %G1
    del_sigma=x_sigma(1,:);
    w_sigma=x_sigma(2,:);
    eq_sigma=x_sigma(3,:);
    ed_sigma=x_sigma(4,:);
    Tmk = repmat(Tm(k-1,:),1,2*ns);
    Efk = repmat(Ef(k-1,:),1,2*ns);
    Vtk = repmat(Vmag_pu(k-1,:),1,2*ns);

    %G2
    del_sigma2=x_sigma(5,:);
    w_sigma2=x_sigma(6,:);
    eq_sigma2=x_sigma(7,:);
    ed_sigma2=x_sigma(8,:);
    Tmk2 = repmat(Tm2(k-1,:),1,2*ns);
    Efk2 = repmat(Ef2(k-1,:),1,2*ns);
    Vtk2 = repmat(Vmag_pu2(k-1,:),1,2*ns);
```

A

```
%G3
del_sigma3=x_sigma(9,:);
w_sigma3=x_sigma(10,:);
eq_sigma3=x_sigma(11,:);
ed_sigma3=x_sigma(12,:);
Tmk3 = repmat(Tm3(k-1,:),1,2*ns);
Efk3 = repmat(Ef3(k-1,:),1,2*ns);
Vtk3 = repmat(Vmag_pu3(k-1,:),1,2*ns);

%G4
del_sigma4=x_sigma(13,:);
w_sigma4=x_sigma(14,:);
eq_sigma4=x_sigma(15,:);
ed_sigma4=x_sigma(16,:);
Tmk4 = repmat(Tm4(k-1,:),1,2*ns);
Efk4 = repmat(Ef4(k-1,:),1,2*ns);
Vtk4 = repmat(Vmag_pu4(k-1,:),1,2*ns);

%G5
del_sigma5=x_sigma(17,:);
w_sigma5=x_sigma(18,:);
eq_sigma5=x_sigma(19,:);
ed_sigma5=x_sigma(20,:);
Tmk5 = repmat(Tm5(k-1,:),1,2*ns);
Efk5 = repmat(Ef5(k-1,:),1,2*ns);
Vtk5 = repmat(Vmag_pu5(k-1,:),1,2*ns);

%DFIG1
ids_sigma=x_sigma(21,:);
iqs_sigma=x_sigma(22,:);
idr_sigma=x_sigma(23,:);
iqr_sigma=x_sigma(24,:);
ww_sigma=x_sigma(25,:);
delw_sigma=x_sigma(26,:);
Tmw = repmat(-0.7552,1,2*ns);
vdsk = repmat(vds(k-1,:),1,2*ns);
vqsk = repmat(vqs(k-1,:),1,2*ns);
vdrk = repmat(vdr(k-1,:),1,2*ns);
vqrk = repmat(vqr(k-1,:),1,2*ns);

% SG: Discretized Model through the Fourth Order
% Runge-Kutta Numerical Integration Method

[del_sigma,w_sigma,eq_sigma,ed_sigma] = gen_model(del_sigma,
w_sigma,eq_sigma,ed_sigma,Tmk,Efk,Vtk,t_step,w_syn,H_rep,
Xd,Xdt,Xq,Xqt,Kd_rep,Td0k,Tq0k);
[del_sigma2,w_sigma2,eq_sigma2,ed_sigma2] = gen_model(
del_sigma2,w_sigma2,eq_sigma2,ed_sigma2,Tmk2,Efk2,Vtk2,
t_step,w_syn,H_rep,Xd,Xdt,Xq,Xqt,Kd_rep,Td0k,Tq0k);
```

A

```
[del_sigma3,w_sigma3,eq_sigma3,ed_sigma3] = gen_model(
del_sigma3,w_sigma3,eq_sigma3,ed_sigma3,Tmk3,Efk3,Vtk3,
t_step,w_syn,H_rep,Xd,Xdt,Xq,Xqt,Kd_rep,Td0k,Tq0k);
[del_sigma4,w_sigma4,eq_sigma4,ed_sigma4] = gen_model(
del_sigma4,w_sigma4,eq_sigma4,ed_sigma4,Tmk4,Efk4,Vtk4,
t_step,w_syn,H_rep,Xd,Xdt,Xq,Xqt,Kd_rep,Td0k,Tq0k);
[del_sigma5,w_sigma5,eq_sigma5,ed_sigma5] = gen_model(
del_sigma5,w_sigma5,eq_sigma5,ed_sigma5,Tmk5,Efk5,Vtk5,
t_step,w_syn,H_rep,Xd,Xdt,Xq,Xqt,Kd_rep,Td0k,Tq0k);

% DFIG: Discretized Model through the Fourth Order
% Runge-Kutta Numerical Integration Method
[ids_sigma,iqs_sigma,idr_sigma,iqr_sigma,ww_sigma,
delw_sigma] = DFIG_model(ids_sigma,iqs_sigma,idr_sigma,
iqr_sigma,ww_sigma,delw_sigma,Tmw,Ls_w,Lr_w,Lm_w,ws_w,
H_w,vdsk,vqsk,vdrk,vqrk,Rs_w,Rr_w,t_step,w_syn);

x_sigma=[del_sigma;w_sigma;eq_sigma;ed_sigma;del_sigma2;
w_sigma2;eq_sigma2;ed_sigma2;del_sigma3;w_sigma3;eq_sigma3;
ed_sigma3;del_sigma4;w_sigma4;eq_sigma4;ed_sigma4;del_sigma5;
w_sigma5;eq_sigma5;ed_sigma5;ids_sigma;iqs_sigma;idr_sigma;
iqr_sigma;ww_sigma;delw_sigma];

%%
%Prioiri State Estimate
x_minus=(1/(2*ns))*(sum(x_sigma'))';
%Priori Covariance Matrix
x_minus_rep=repmat(x_minus,1,2*ns);
P_minus=(1/(2*ns))*(x_sigma-x_minus_rep)*
(x_sigma-x_minus_rep)'+Q;

%% Measurement Sigma Points

%G1
Vtkp = repmat(Vmag_pu(k,:),1,2*ns);
[y1_g1,y2_g1,y3_g1] =
    sgm(del_sigma,w_sigma,eq_sigma,Vtkp,Xdt,Xq);

%G2
Vtkp2 = repmat(Vmag_pu2(k,:),1,2*ns);
[y1_g2,y2_g2,y3_g2] =
    sgm(del_sigma2,w_sigma2,eq_sigma2,Vtkp2,Xdt,Xq);

%G3
Vtkp3 = repmat(Vmag_pu3(k,:),1,2*ns);
[y1_g3,y2_g3,y3_g3] =
    sgm(del_sigma3,w_sigma3,eq_sigma3,Vtkp3,Xdt,Xq);

%G4
```

A

```
Vtkp4 = repmat(Vmag_pu4(k,:),1,2*ns);
[y1_g4,y2_g4,y3_g4] =
    sgm(del_sigma4,w_sigma4,eq_sigma4,Vtkp4,Xdt,Xq);

%G5
Vtkp5 = repmat(Vmag_pu5(k,:),1,2*ns);
[y1_g5,y2_g5,y3_g5] =
    sgm(del_sigma5,w_sigma5,eq_sigma5,Vtkp5,Xdt,Xq);

%DFIG
vdsk = repmat(vds(k,:),1,2*ns);
vqsk = repmat(vqs(k,:),1,2*ns);
vdrk = repmat(vdr(k,:),1,2*ns);
vqrk = repmat(vqr(k,:),1,2*ns);

y_Pw = (vdsk.*ids_sigma+vqsk.*iqs_sigma+vdrk.*idr_sigma+
vqrk.*iqr_sigma);
y_Qw = (vqsk.*ids_sigma-vdsk.*iqs_sigma+vqrk.*idr_sigma-
vdrk.*iqr_sigma);
y_Fw = 60.*(ww_sigma)/(1+0.20375);

y_sigma=[y1_g1;y2_g1;y3_g1;y1_g2;y2_g2;y3_g2;y1_g3;
y2_g3;y3_g3;y1_g4;
y2_g4;y3_g4;y1_g5;y2_g5;y3_g5;y_Pw;y_Qw;y_Fw];

%y Predict
y_predict=(1/(2*ns))*(sum(y_sigma'))';
%%
%Covariance of predicted measurements Py
y_predict_rep=repmat(y_predict,1,2*ns);
P_y=(2*ns)^-1*(y_sigma-y_predict_rep)*(y_sigma-y_predict_rep)'+R;

%Cross covariance Pxy
P_xy=(2*ns)^-1*(x_sigma-x_minus_rep)*(y_sigma-y_predict_rep)';
%% Measurement update of state estimate

% Kalman Gain
K=P_xy*P_y^-1;

%G1
y_PG1=P_m(k,:);
y_QG1=Q_m(k,:);
y_FG1=F_G1(k,:);

%G2
y_PG2=P_m2(k,:);
y_QG2=Q_m2(k,:);
y_FG2=F_G2(k,:);
```

A

```
        %G3
        y_PG3=P_m3(k,:);
        y_QG3=Q_m3(k,:);
        y_FG3=F_G3(k,:);

        %G4
        y_PG4=P_m4(k,:);
        y_QG4=Q_m4(k,:);
        y_FG4=F_G4(k,:);

        %G5
        y_PG5=P_m5(k,:);
        y_QG5=Q_m5(k,:);
        y_FG5=F_G5(k,:);

        %DFIG1
        y_PW=P_DFIG(k,:);
        y_QW=Q_DFIG(k,:);
        y_FG6=F_G6(k,:);

        %v=sig^3*randn(nm,1); %Measurement Noise
        y=[y_PG1;y_QG1;y_FG1;y_PG2;y_QG2;y_FG2;y_PG3;y_QG3;y_FG3;
        y_PG4;y_QG4;y_FG4;y_PG5;y_QG5;y_FG5;y_PW;y_QW;y_FG6];%+v;
        x_hat(:,k)=x_minus+K*(y-y_predict);
        P=P_minus-K*P_y*K';
    end
    x_estimate = x_hat;
    P_final=P;
end
%% Discretized Model through the Fourth Order
%% Runge-Kutta Numerical Integration Method

function [del_sigma_up,w_sigma_up,eq_sigma_up,ed_sigma_up] =
gen_model(del_sigma,w_sigma,eq_sigma,ed_sigma,Tmk,Efk,Vtk,t_step,
w_syn,H,Xd,Xdt,Xq,Xqt,Kd_rep,Td0k,Tq0k)
    %k1
    k1_del = t_step.*(w_syn*w_sigma);
    k1_w = t_step.*((((2.*H).^-1)).*((Tmk)-((Vtk./Xdt).
    *eq_sigma.*sin(del_sigma))-(((Vtk.^2)./2).*((1/Xq)-(1/Xdt))
    .*sin(2.*del_sigma))-(Kd_rep.*w_sigma)));
    k1_eq = t_step.*(((Td0k.^-1)).*((Efk)-(eq_sigma)-((Xd-Xdt).*
    (eq_sigma-Vtk.*cos(del_sigma))./Xdt)));
    k1_ed = t_step.*((Tq0k.^-1).*(-(ed_sigma)+((Xq-Xqt).*
    ((Vtk.*sin(del_sigma))./Xq))));

    %k2
    k2_del = t_step.*(w_syn*(w_sigma+k1_w/2));
    k2_w = t_step.*((((2.*H).^-1)).*((Tmk)-((Vtk./Xdt).*
    (eq_sigma+k1_eq/2).*sin((del_sigma+k1_del/2).*1))-
```

A

```
    (((Vtk.^2)./2).*((1/Xq)-(1/Xdt))
    .*sin(2.*(del_sigma+k1_del/2).*1))-(Kd_rep.*(w_sigma+k1_w/2))));
    k2_eq = t_step.*(((Td0k.^-1)).*((Efk)-
    ((eq_sigma+k1_eq/2))-((Xd-Xdt).*
    ((eq_sigma+k1_eq/2)-Vtk.*cos((del_sigma+k1_del/2).*1))./Xdt)));
    k2_ed = t_step.*(((Tq0k.^-1)).*(-((ed_sigma+k1_ed/2))+
    ((Xq-Xqt).*((Vtk.*sin((del_sigma+k1_del/2).*1))./Xq))));

    %k3
    k3_del = t_step.*(w_syn*(w_sigma+k2_w/2));
    k3_w = t_step.*((((2.*H).^-1)).*((Tmk)-((Vtk./Xdt)
    .*(eq_sigma+k2_eq/2).*sin((del_sigma+k2_del/2).*1))
    -(((Vtk.^2)./2).*((1/Xq)-(1/Xdt)).*sin(2.*(del_sigma+k2_del/2).*1))
    -(Kd_rep.*(w_sigma+k2_w/2))));
    k3_eq = t_step.*(((Td0k.^-1)).*((Efk)-((eq_sigma+k2_eq/2))
    -((Xd-Xdt).*((eq_sigma+k2_eq/2)-
    Vtk.*cos((del_sigma+k2_del/2).*1))./Xdt)));
    k3_ed = t_step.*(((Tq0k.^-1)).*(-((ed_sigma+k2_ed/2))
    +((Xq-Xqt).*((Vtk.*sin((del_sigma+k2_del/2).*1))./Xq))));

    %k4
    k4_del = t_step.*(w_syn*(w_sigma+k3_w));
    k4_w = t_step.*((((2.*H).^-1)).*((Tmk)-((Vtk./Xdt).*
    (eq_sigma+k3_eq).*sin((del_sigma+k3_del).*1))-(((Vtk.^2)./2).*
    ((1/Xq)-(1/Xdt)).*sin(2.*(del_sigma+k3_del).*1))-
    (Kd_rep.*(w_sigma+k3_w))));
    k4_eq = t_step.*(((Td0k.^-1)).*((Efk)-((eq_sigma+k3_eq))
    -((Xd-Xdt).*((eq_sigma+k3_eq)-Vtk.*cos((del_sigma+k3_del).*1))./Xdt)));
    k4_ed = t_step.*(((Tq0k.^-1)).*(-((ed_sigma+k3_ed))
    +((Xq-Xqt).*((Vtk.*sin((del_sigma+k3_del).*1))./Xq))));

    del_sigma_up = del_sigma + (1/6)*(k1_del+2*k2_del+2*k3_del+k4_del);
    w_sigma_up = w_sigma + (1/6)*(k1_w+2*k2_w+2*k3_w+k4_w);
    eq_sigma_up = eq_sigma + (1/6)*(k1_eq+2*k2_eq+2*k3_eq+k4_eq);
    ed_sigma_up = ed_sigma + (1/6)*(k1_ed+2*k2_ed+2*k3_ed+k4_ed);
end
%%
%% Discretized Model through the Fourth Order
%% Runge-Kutta Numerical Integration Method

function [ids_sigma_up,iqs_sigma_up,idr_sigma_up,iqr_sigma_up,
ww_sigma_up,delw_sigma_up] = DFIG_model(ids_sigma,iqs_sigma,
idr_sigma,iqr_sigma,ww_sigma,delw_sigma,Tmw,Ls_w,Lr_w,Lm_w,
ws_w,H_w,vdsk,vqsk,vdrk,vqrk,Rs_w,Rr_w,t_step,w_syn)

    %k1
    Te_w = Lm_w*(iqs_sigma.*idr_sigma-ids_sigma.*iqr_sigma);
    k1_ids = t_step.*((1/(Ls_w.*Lr_w-Lm_w^2)).*(-Rs_w.*Lr_w.*
    ids_sigma+(ww_sigma.*(Lm_w^2)+ws_w.*(Ls_w.*Lr_w-Lm_w^2)).*
```

A

```
iqs_sigma+Rr_w.*Lm_w.*idr_sigma+ww_sigma.*Lr_w.*Lm_w.*
iqr_sigma+Lr_w.*vdsk-Lm_w.*vdrk));
k1_iqs = t_step.*((1/(Ls_w.*Lr_w-Lm_w^2)).*(-(ww_sigma.*
(Lm_w^2)+ws_w.*(Ls_w.*Lr_w-Lm_w^2)).*ids_sigma-Rs_w.*Lr_w.*
iqs_sigma+Rr_w.*Lm_w.*iqr_sigma-ww_sigma.*Lr_w.*Lm_w.*
idr_sigma+Lr_w.*vqsk-Lm_w.*vqrk));
k1_idr = t_step.*((1/(Ls_w.*Lr_w-Lm_w^2)).*(Rs_w.*Lm_w.*
ids_sigma-ww_sigma.*Ls_w.*Lm_w.*iqs_sigma-Rr_w.*Ls_w.*
idr_sigma+(-ww_sigma.*Lr_w.*Ls_w+ws_w.*(Ls_w.*Lr_w-Lm_w^2))
.*iqr_sigma-Lm_w.*vdsk+Ls_w.*vdrk));
k1_iqr = t_step.*((1/(Ls_w.*Lr_w-Lm_w^2)).*(Rs_w.*Lm_w.*
iqs_sigma+ww_sigma.*Ls_w.*Lm_w.*ids_sigma-Rr_w.*Ls_w.*
iqr_sigma-(-ww_sigma.*Lr_w.*Ls_w+ws_w.*(Ls_w.*Lr_w-Lm_w^2))
.*idr_sigma-Lm_w.*vqsk+Ls_w.*vqrk));
k1_ww = t_step.*((1/2.*H_w).*(Tmw-Te_w));
k1_delw = t_step.*(w_syn.*ww_sigma);

%k2
Te_w = Lm_w*((iqs_sigma+k1_iqs/2).*(idr_sigma+k1_idr/2)-
(ids_sigma+k1_ids/2).*(iqr_sigma+k1_iqr/2));
k2_ids = t_step.*((1/(Ls_w.*Lr_w-Lm_w^2)).*(-Rs_w.*Lr_w.*
(ids_sigma+k1_ids/2)+((ww_sigma+k1_ww/2).*(Lm_w^2)+ws_w.*
(Ls_w.*Lr_w-Lm_w^2)).*(iqs_sigma+k1_iqs/2)+Rr_w.*Lm_w.*
(idr_sigma+k1_idr/2)+(ww_sigma+k1_ww/2).*Lr_w.*Lm_w.*
(iqr_sigma+k1_iqr/2)+Lr_w.*vdsk-Lm_w.*vdrk));
k2_iqs = t_step.*((1/(Ls_w.*Lr_w-Lm_w^2)).*(-((ww_sigma+k1_ww/2)
.*(Lm_w^2)+ws_w.*(Ls_w.*Lr_w-Lm_w^2)).*(ids_sigma+k1_ids/2)-
Rs_w.*Lr_w.*(iqs_sigma+k1_iqs/2)+Rr_w.*Lm_w.*(iqr_sigma+k1_iqr/2)
-(ww_sigma+k1_ww/2).*Lr_w.*Lm_w.*(idr_sigma+k1_idr/2)+
Lr_w.*vqsk-Lm_w.*vqrk));
k2_idr = t_step.*((1/(Ls_w.*Lr_w-Lm_w^2)).*(Rs_w.*Lm_w.*
(ids_sigma+k1_ids/2)-(ww_sigma+k1_ww/2).*Ls_w.*Lm_w.*
(iqs_sigma+k1_iqs/2)-Rr_w.*Ls_w.*(idr_sigma+k1_idr/2)+
(-(ww_sigma+k1_ww/2).*Lr_w.*Ls_w+ws_w.*(Ls_w.*Lr_w-Lm_w^2))
.*(iqr_sigma+k1_iqr/2)-Lm_w.*vdsk+Ls_w.*vdrk));
k2_iqr = t_step.*((1/(Ls_w.*Lr_w-Lm_w^2)).*(Rs_w.*Lm_w.*
(iqs_sigma+k1_iqs/2)+(ww_sigma+k1_ww/2).*Ls_w.*Lm_w.*
(ids_sigma+k1_ids/2)-Rr_w.*Ls_w.*(iqr_sigma+k1_iqr/2)-
(-(ww_sigma+k1_ww/2).*Lr_w.*Ls_w+ws_w.*(Ls_w.*Lr_w-Lm_w^2))
.*(idr_sigma+k1_idr/2)-Lm_w.*vqsk+Ls_w.*vqrk));
k2_ww = t_step.*((1/2.*H_w).*(Tmw-Te_w));
k2_delw = t_step.*(w_syn.*(ww_sigma+k1_ww/2));

%k3
Te_w = Lm_w*((iqs_sigma+k2_iqs/2).*(idr_sigma+k2_idr/2)-
(ids_sigma+k2_ids/2).*(iqr_sigma+k2_iqr/2));
k3_ids = t_step.*((1/(Ls_w.*Lr_w-Lm_w^2)).*(-Rs_w.*Lr_w.*
(ids_sigma+k2_ids/2)+((ww_sigma+k2_ww/2).*(Lm_w^2)+ws_w.*
(Ls_w.*Lr_w-Lm_w^2)).*(iqs_sigma+k2_iqs/2)+Rr_w.*Lm_w.*
```

A

```
(idr_sigma+k2_idr/2)+(ww_sigma+k2_ww/2).*Lr_w.*Lm_w.*
(iqr_sigma+k2_iqr/2)+Lr_w.*vdsk-Lm_w.*vdrk));
k3_iqs = t_step.*((1/(Ls_w.*Lr_w-Lm_w^2)).*
(-((ww_sigma+k2_ww/2).*(Lm_w^2)+ws_w.*(Ls_w.*Lr_w-Lm_w^2))
.*(ids_sigma+k2_ids/2)-Rs_w.*Lr_w.*(iqs_sigma+k2_iqs/2)
+Rr_w.*Lm_w.*(iqr_sigma+k2_iqr/2)-(ww_sigma+k2_ww/2)
.*Lr_w.*Lm_w.*(idr_sigma+k2_idr/2)+Lr_w.*vqsk-Lm_w.*vqrk));
k3_idr = t_step.*((1/(Ls_w.*Lr_w-Lm_w^2))
.*(Rs_w.*Lm_w.*(ids_sigma+k2_ids/2)-(ww_sigma+k2_ww/2)
.*Ls_w.*Lm_w.*(iqs_sigma+k2_iqs/2)-Rr_w.*Ls_w.*(idr_sigma+k2_idr/2)
+(-(ww_sigma+k2_ww/2).*Lr_w.*Ls_w+ws_w.*(Ls_w.*Lr_w-Lm_w^2))
.*(iqr_sigma+k2_iqr/2)-Lm_w.*vdsk+Ls_w.*vdrk));
k3_iqr = t_step.*((1/(Ls_w.*Lr_w-Lm_w^2)).*(Rs_w.*Lm_w.*
(iqs_sigma+k2_iqs/2)+(ww_sigma+k2_ww/2).*Ls_w.*Lm_w.*
(ids_sigma+k2_ids/2)-Rr_w.*Ls_w.*(iqr_sigma+k2_iqr/2)-
(-(ww_sigma+k2_ww/2).*Lr_w.*Ls_w+ws_w.*(Ls_w.*Lr_w-Lm_w^2)).*
(idr_sigma+k2_idr/2)-Lm_w.*vqsk+Ls_w.*vqrk));
k3_ww = t_step.*((1/2.*H_w).*(Tmw-Te_w));
k3_delw = t_step.*(w_syn.*(ww_sigma+k2_ww/2));

%k4
Te_w =
    Lm_w*((iqs_sigma+k3_iqs).*(idr_sigma+k3_idr)-(ids_sigma+k3_ids)
.*(iqr_sigma+k3_iqr));
k4_ids = t_step.*((1/(Ls_w.*Lr_w-Lm_w^2)).*(-Rs_w.*Lr_w.*
(ids_sigma+k3_ids)+((ww_sigma+k3_ww).*(Lm_w^2)+ws_w.*
(Ls_w.*Lr_w-Lm_w^2)).*(iqs_sigma+k3_iqs)+Rr_w.*Lm_w.*
(idr_sigma+k3_idr)+(ww_sigma+k3_ww/2).*Lr_w.*Lm_w.*
(iqr_sigma+k3_iqr)+Lr_w.*vdsk-Lm_w.*vdrk));
k4_iqs = t_step.*((1/(Ls_w.*Lr_w-Lm_w^2)).*(-((ww_sigma+k3_ww)
.*(Lm_w^2)+ws_w.*(Ls_w.*Lr_w-Lm_w^2)).*(ids_sigma+k3_ids)
-Rs_w.*Lr_w.*(iqs_sigma+k3_iqs)+Rr_w.*Lm_w.*
(iqr_sigma+k3_iqr)-(ww_sigma+k3_ww).*Lr_w.*Lm_w.*
(idr_sigma+k3_idr)+Lr_w.*vqsk-Lm_w.*vqrk));
k4_idr = t_step.*((1/(Ls_w.*Lr_w-Lm_w^2)).*(Rs_w.*Lm_w
.*(ids_sigma+k3_ids)-(ww_sigma+k3_ww).*Ls_w.*Lm_w.*
(iqs_sigma+k3_iqs)-Rr_w.*Ls_w.*(idr_sigma+k3_idr)+
(-(ww_sigma+k3_ww).*Lr_w.*Ls_w+ws_w.*(Ls_w.*Lr_w-Lm_w^2)).*
(iqr_sigma+k3_iqr)-Lm_w.*vdsk+Ls_w.*vdrk));
k4_iqr = t_step.*((1/(Ls_w.*Lr_w-Lm_w^2)).*(Rs_w.*Lm_w.*
(iqs_sigma+k3_iqs)+(ww_sigma+k3_ww).*Ls_w.*Lm_w.*
(ids_sigma+k3_ids)-Rr_w.*Ls_w.*(iqr_sigma+k3_iqr)-
(-(ww_sigma+k3_ww).*Lr_w.*Ls_w+ws_w.*(Ls_w.*Lr_w-Lm_w^2))
.*(idr_sigma+k3_idr)-Lm_w.*vqsk+Ls_w.*vqrk));
k4_ww = t_step.*((1/2.*H_w).*(Tmw-Te_w));
k4_delw = t_step.*(w_syn.*(ww_sigma+k3_ww));

ids_sigma_up = ids_sigma +
    (1/6).*(k1_ids+2.*k2_ids+2.*k3_ids+k4_ids);
```

A

```
    iqs_sigma_up = iqs_sigma +
        (1/6).*(k1_iqs+2.*k2_iqs+2.*k3_iqs+k4_iqs);
    idr_sigma_up = idr_sigma +
        (1/6).*(k1_idr+2.*k2_idr+2.*k3_idr+k4_idr);
    iqr_sigma_up = iqr_sigma +
        (1/6).*(k1_iqr+2.*k2_iqr+2.*k3_iqr+k4_iqr);
    ww_sigma_up = ww_sigma + (1/6).*(k1_ww+2.*k2_ww+2.*k3_ww+k4_ww);
    delw_sigma_up = delw_sigma +
        (1/6).*(k1_delw+2.*k2_delw+2.*k3_delw+k4_delw);

end
%%
%% Measurement Equation for the Fourth Order Synchronous Generator Model

function [y1,y2,y3] = sgm(del_sigma,w_sigma,eq_sigma,Vtkp,Xdt,Xq)
    y1 = (((Vtkp./Xdt).*eq_sigma.*sin(del_sigma))+((Vtkp.^2)/2)
    .*((1/Xq)-(1/Xdt)).*sin(2*del_sigma));
    y2 = (((Vtkp./Xdt).*eq_sigma.*cos(del_sigma))-(Vtkp.^2)
    .*((((cos(del_sigma)).^2)/Xdt)+(((sin(del_sigma)).^2)/Xq)));
    y3 = 60.*(w_sigma+1);
end
```

A.2. DSE PLOTS

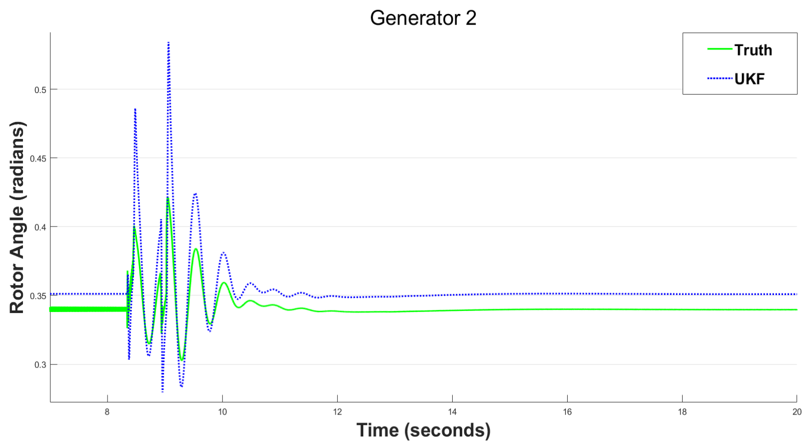


Figure A.1: Generator 2: True and Estimated Rotor angles

A

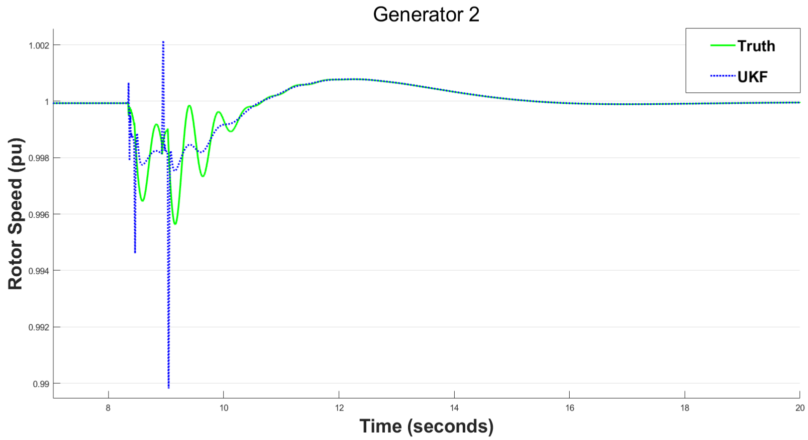


Figure A.2: Generator 2: True and Estimated Rotor Speeds

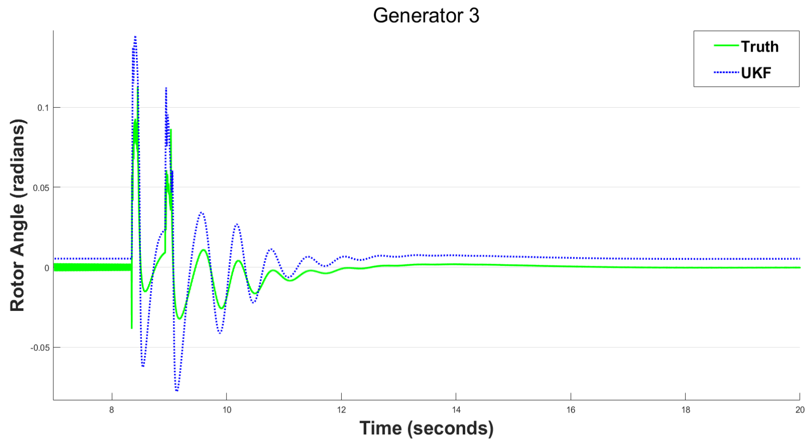


Figure A.3: Generator 3: True and Estimated Rotor angles

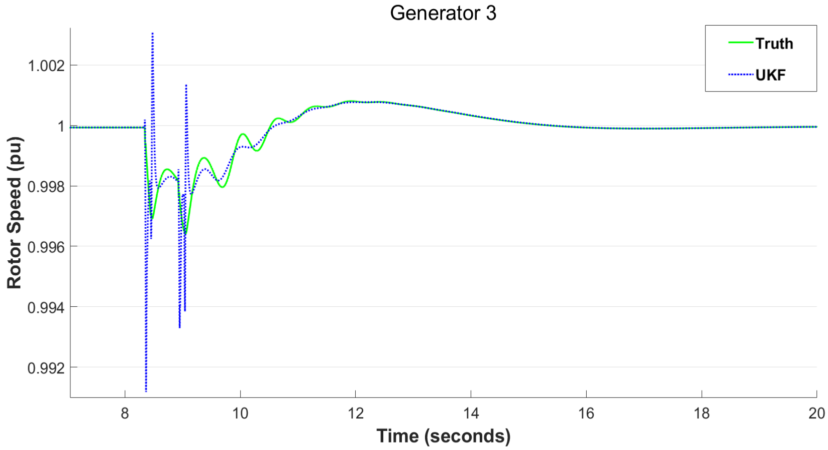


Figure A.4: Generator 3: True and Estimated Rotor Speeds

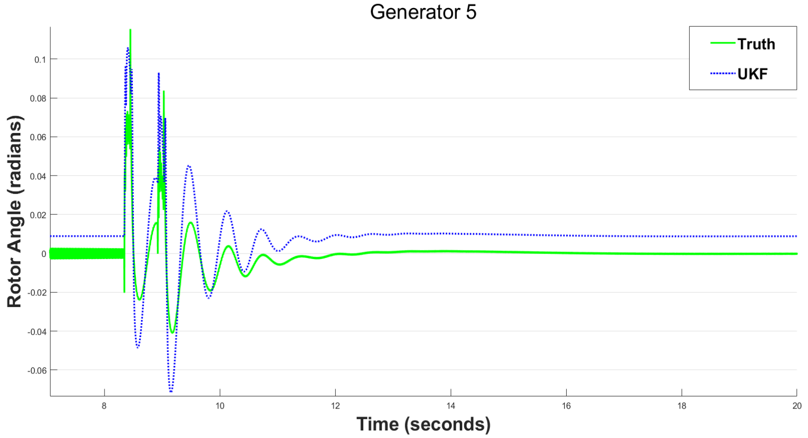


Figure A.5: Generator 5: True and Estimated Rotor angles

Figure A.6: Generator 5: True and Estimated Rotor Speeds

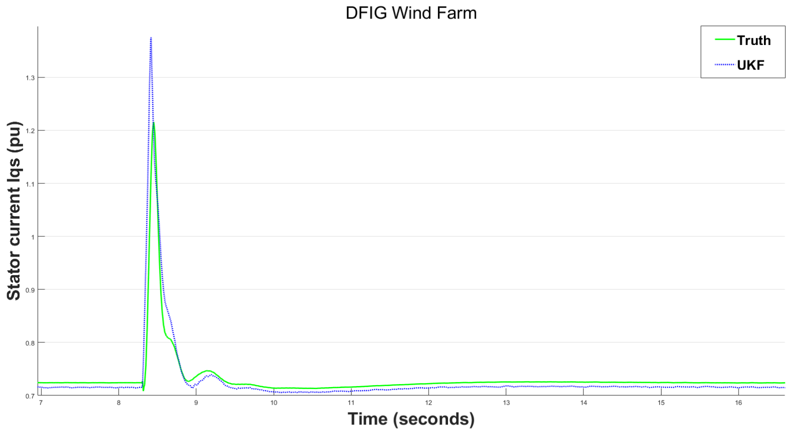


Figure A.7: DFIG Wind Farm: True and Estimated Stator q-axis current

A

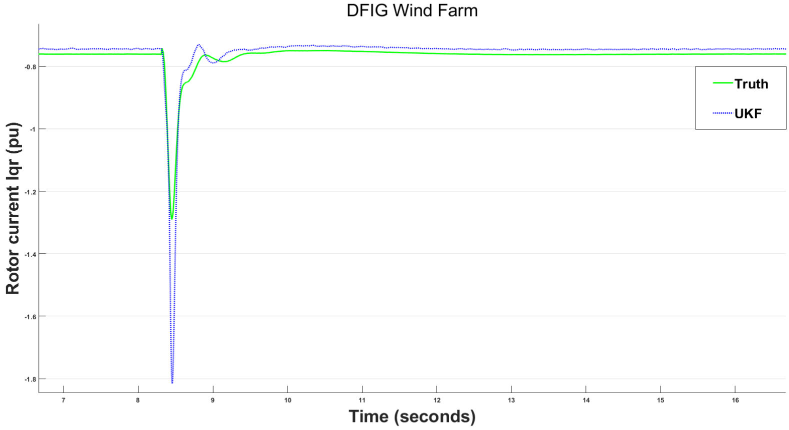


Figure A.8: DFIG Wind Farm: True and Estimated Rotor q-axis current

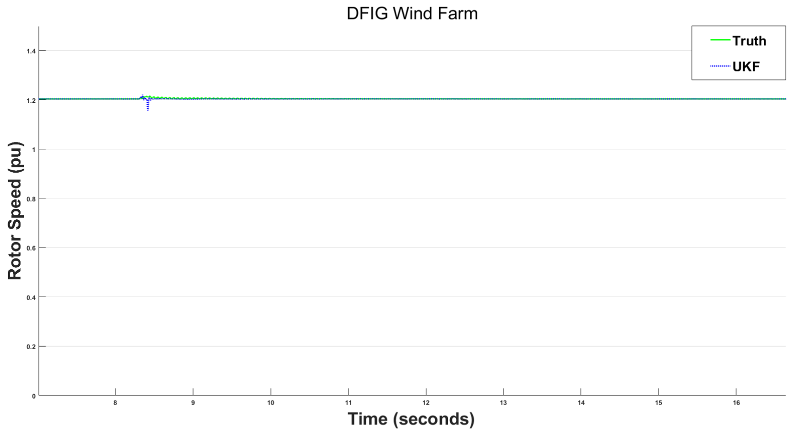


Figure A.9: DFIG Wind Farm: True and Estimated Rotor Speed

REFERENCES

[1] R. Ghaniabadi, S. M. Kouhsari, I. Pourkeivani, and M. Abedi, “Power system model-based dynamic state estimation using wide-area measurements (wams),” in *2019 International Power System Conference (PSC)*, 2019, pp. 536–541. DOI: 10.1109/PSC49016.2019.9081509.

[2] Wenzhong Gao and S. Wang, “On-line dynamic state estimation of power systems,” in *North American Power Symposium 2010*, IEEE, Sep. 2010, pp. 1–6, ISBN: 978-1-4244-8046-3. DOI: 10.1109/NAPS.2010.5619951.

[3] J. Zhao, J. Qi, Z. Huang, A. P. S. Meliopoulos, A. Gomez-Exposito, M. Netto, L. Mili, A. Abur, V. Terzija, I. Kamwa, B. Pal, and A. K. Singh, “Power system dynamic state estimation: Motivations, definitions, methodologies, and future work,” *IEEE Transactions on Power Systems*, vol. 34, pp. 3188–3198, Jul. 2019, ISSN: 0885-8950, 1558-0679. DOI: 10.1109/TPWRS.2019.2894769.

[4] W. E. Outlook. “Outlook for electricity – world energy outlook 2020 – analysis.” (2020), [Online]. Available: https://www.iea.org/reports/world-energy-outlook-2020/outlook-for-electricity.

[5] S. Impram, S. Varbak Nese, and B. Oral, “Challenges of renewable energy penetration on power system flexibility: A survey,” *Energy Strategy Reviews*, vol. 31, p. 100 539, 2020, ISSN: 2211-467X. DOI: https://doi.org/10.1016/j.esr.2020.100539.

[6] P. Kundur, *Power system stability and control.* New York: McGraw-Hill, 1993.

[7] E. Munkhchuluun, L. Meegahapola, and A. Vahidnia, “Impact on rotor angle stability with high solar-pv generation in power networks,” in *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, 2017, pp. 1–6. DOI: 10.1109/ISGTEurope.2017.8260229.

[8] M. Edrah, K. L. Lo, and O. Anaya-Lara, “Impacts of high penetration of dfig wind turbines on rotor angle stability of power systems,” *IEEE Transactions on Sustainable Energy*, vol. 6, no. 3, pp. 759–766, 2015. DOI: 10.1109/TSTE.2015.2412176.

[9] W. Miller and J. Lewis, “Dynamic state estimation in power systems,” *IEEE Transactions on Automatic Control*, vol. 16, no. 6, pp. 841–846, 1971. DOI: 10.1109/TAC.1971.1099844.

[10] J. Zhao, M. Netto, Z. Huang, S. Yu, A. Gomez-Exposito, S. Wang, I. Kamwa, S. Akhlaghi, L. Mili, V. Terzija, A. P. S. Meliopoulos, B. C. Pal, A. K. Singh, A. Abur, T. Bi, and A. Rouhani, “Roles of dynamic state estimation in power system modeling, monitoring and operation,” *IEEE Transactions on Power Systems*, pp. 1–1, 2020. DOI: 10.1109/TPWRS.2020.3028047.

[11] A. Sharma and S. K. Jain, “A review and performance comparison of power system state estimation techniques,” in *2018 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia)*, 2018, pp. 770–775. DOI: 10.1109/ISGT-Asia.2018.8467861.

[12] Rui Fan, Z. Huang, Shaobu Wang, R. Diao, and Da Meng, “Dynamic state estimation and parameter calibration of a DFIG using the ensemble kalman filter,” in *2015 IEEE Power & Energy Society General Meeting*, Denver, CO, USA: IEEE, Jul. 2015, pp. 1–5, ISBN: 978-1-4673-8040-9. DOI: 10.1109/PESGM.2015.7285990.

[13] H. Khazraj, F. Faria da Silva, and C. L. Bak, “A performance comparison between extended kalman filter and unscented kalman filter in power system dynamic state estimation,” in *2016 51st International Universities Power Engineering Conference (UPEC)*, Coimbra: IEEE, Sep. 2016, pp. 1–6, ISBN: 978-1-5090-4650-8. DOI: 10.1109/UPEC.2016.8114125.

[14] N. Zhou, D. Meng, Z. Huang, and G. Welch, “Dynamic state estimation of a synchronous machine using pmu data: A comparative study,” *IEEE Transactions on Smart Grid*, vol. 6, no. 1, pp. 450–460, 2015. DOI: 10.1109/TSG.2014.2345698.

[15] E. Ghahremani and I. Kamwa, “Local and wide-area pmu-based decentralized dynamic state estimation in multi-machine power systems,” *IEEE Transactions on Power Systems*, vol. 31, no. 1, pp. 547–562, 2016. DOI: 10.1109/TPWRS.2015.2400633.

[16] ——, “Dynamic state estimation in power system by applying the extended kalman filter with unknown inputs to phasor measurements,” *IEEE Transactions on Power Systems*, vol. 26, no. 4, pp. 2556–2566, 2011. DOI: 10.1109/TPWRS.2011.2145396.

[17] A. Paul, G. Joos, and I. Kamwa, “Decentralized dynamic state estimation of doubly fed induction generator using terminal measurements,” in *2018 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, Washington DC, DC, USA: IEEE, Feb. 2018, pp. 1–5, ISBN: 978-1-5386-2453-1. DOI: 10.1109/ISGT.2018.8403339.

[18] S. Yu, K. Emami, T. Fernando, H. H. C. Iu, and K. P. Wong, “State estimation of doubly fed induction generator wind turbine in complex power systems,” *IEEE Transactions on Power Systems*, vol. 31, pp. 4935–4944, Nov. 2016, ISSN: 0885-8950, 1558-0679. DOI: 10.1109/TPWRS.2015.2507620.

[19] S. S. Yu, J. Guo, T. K. Chau, T. Fernando, H. H.-C. Iu, and H. Trinh, “An unscented particle filtering approach to decentralized dynamic state estimation for DFIG wind turbines in multi-area power systems,” *IEEE Transactions on Power Systems*, vol. 35, pp. 2670–2682, Jul. 2020, ISSN: 0885-8950, 1558-0679. DOI: 10.1109/TPWRS.2020.2966443.

[20] W. Su, B. Liu, Z. Li, X. Mao, M. Huang, G. He, and X. Liu, “Dynamic state estimation for DFIG wind turbine with stochastic wind speed in power system,” in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, Florence: IEEE, 2018, pp. 1–5, ISBN: 978-1-5386-4881-0. DOI: 10.1109/ISCAS.2018.8351791.

[21] R. Jha and N. Senroy, "Wams based dynamic states and parameters estimation using least squares estimation and unscented kalman filter," in *2017 IEEE Innovative Smart Grid Technologies - Asia (ISGT-Asia)*, Auckland: IEEE, Dec. 2017, pp. 1–6, ISBN: 978-1-5386-4950-3. DOI: 10.1109/ISGT-Asia.2017.8378440.

[22] Tianshu Bi, Xiaohui Qin, and Qixun Yang, "Wams based dynamic state estimator during power system transient process," in *APCCAS 2008 - 2008 IEEE Asia Pacific Conference on Circuits and Systems*, 2008, pp. 777–780. DOI: 10.1109/APCCAS.2008.4746138.

[23] J. Zhao. "Performance improvement of wide-area-monitoring-system (wams) and applications development." (2018), [Online]. Available: https://trace.tennessee.edu/cgi/viewcontent.cgi?article=6462&context=utk_graddiss.

[24] L. M. Putranto and N. Hoonchareon, "Wide area monitoring system implementation in securing voltage stability based on phasor measurement unit data," in *2013 10th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, 2013, pp. 1–6. DOI: 10.1109/ECTICon.2013.6559477.

[25] P. Wall, P. Dattaray, Z. Jin, P. Mohapatra, J. Yu, D. Wilson, K. Hay, S. Clark, M. Osborne, and P. M. e. a. Ashton, "Deployment and demonstration of wide area monitoring system in power system of great britain," *Journal of Modern Power Systems and Clean Energy*, vol. 4, no. 3, pp. 506–518, 2016. DOI: 10.1007/s40565-016-0218-3.

[26] E. Ghahremani, I. Kamwa, W. Li, and L. Grégoire, "Synchrophasor based tracking of synchronous generator dynamic states using a fast ekf with unknown mechanical torque and field voltage," in *IECON 2014 - 40th Annual Conference of the IEEE Industrial Electronics Society*, 2014, pp. 302–308. DOI: 10.1109/IECON.2014.7048515.

[27] Ning Zhou, Da Meng, Zhenyu Huang, and G. Welch, "Dynamic state estimation of a synchronous machine using pmu data: A comparative study," in *2015 IEEE Power Energy Society General Meeting*, 2015, pp. 1–1. DOI: 10.1109/PESGM.2015.7285966.

[28] E. Ghahremani, I. Kamwa, W. Li, and L.-A. Grégoire, "Synchrophasor based tracking of synchronous generator dynamic states using a fast ekf with unknown mechanical torque and field voltage," in *IECON 2014 - 40th Annual Conference of the IEEE Industrial Electronics Society*, 2014, pp. 302–308. DOI: 10.1109/IECON.2014.7048515.

[29] H. Li and R. Curiac, "Understanding of induction motors made easy," in *2011 Record of Conference Papers Industry Applications Society 58th Annual IEEE Petroleum and Chemical Industry Conference (PCIC)*, 2011, pp. 1–6. DOI: 10.1109/PCICon.2011.6085874.

[30] V. I. Bandla Pavan Babu. "Analysis of back to back (btb) converter control strategies in different power system applications." (2020), [Online]. Available: https://iopscience.iop.org/article/10.1088/1757-899X/906/1/012016/pdf.

[31] RTDS-Technologies, "Wind-Turbine Driven Doubly-Fed Induction Generator," 2021.

[32] "Reference frame theory/induction machines." (), [Online]. Available: http://home.engineering.iastate.edu/~jdm/ee554/LastTopics1.pdf.

[33] C. L. Enloe, *Physical Science: What the Technology Professional Needs to Know*. John Wiley and Sons, 2000.

[34] MathWorks. "Understanding kalman filters." (2017), [Online]. Available: https://www.mathworks.com/videos/series/understanding-kalman-filters.html.

[35] C. Bila. "Power system dynamic state estimation and load modeling." (2013), [Online]. Available: https://repository.library.northeastern.edu/files/neu:1237/fulltext.pdf.

[36] J. Qi, K. Sun, J. Wang, and H. Liu, "Dynamic state estimation for multi-machine power system by unscented kalman filter with enhanced numerical stability," *IEEE Transactions on Smart Grid*, vol. 9, pp. 1184–1196, Mar. 2018, ISSN: 1949-3053, 1949-3061. DOI: 10.1109/TSG.2016.2580584.

[37] M. Naglic, "On power system automation: Synchronised measurement technology supported power system situational awareness," Ph.D. dissertation, Delft University of Technology. DOI: 10.4233/UUID:33A9138F-7224-4734-A326-D90A9D5980C1.

[38] R. Technologies. "Real time digital simulator." (2021), [Online]. Available: https://www.rtds.com/.

[39] RTDS-technologies, "IEEE 14 BUS POWER SYSTEM," 2021.

[40] "Gtnetx2: The rtds simulator's network interface card." (), [Online]. Available: https://knowledge.rtds.com/hc/en-us/articles/360034788593-GTNETx2-The-RTDS-Simulator-s-Network-Interface-Card.

[41] "Gtsync card." (), [Online]. Available: https://knowledge.rtds.com/hc/en-us/articles/360034815433-GTSYNC-Card.

[42] M. Naglic, "Synchro-measurement application development framework: An IEEE standard c37.118.2-2011 supported MATLAB library," *IEEE Transactions on Instrumentation and Measurement*, vol. 67, p. 11, 2018.