MODEL REFERENCE ADAPTIVE CONTROL ALGORITHM FOR POWER SYSTEM INTERAREA OSCILLATIONS DAMPING

by

TSWA-WEN PIERRE-PATRICK BANGA-BANGA

Thesis submitted in fulfilment of the requirements for the degree

Master of Engineering: Electrical Engineering

in the Faculty of Engineering and the Built Environment

at the Cape Peninsula University of Technology

Supervisor: Dr C Kriger
Co-supervisor: Dr Y. D. Mfoumboulou

Bellville Campus

DECLARATION

I, Tswa-wen Pierre-Patrick Banga-Banga, declare that the contents of this dissertation/thesis represent my own unaided work, and that the dissertation/thesis has not previously been submitted for academic examination towards any qualification. Furthermore, it represents my own opinions and not necessarily those of the Cape Peninsula University of Technology.

March 2022

Signed   **Date**

ABSTRACT

Low-Frequency Electromechanical Oscillations (LFEOs) represent a real threat to power system networks as they are the primary cause of inter-area oscillations and because they limit the generation's output. Mitigating their effects is therefore crucial as it may lead to system collapse if not properly damped.

As Rotor angle instability is the primary cause of LFEOs, this thesis presents a novel Model-Reference Adaptive Control scheme that enhances its stability. The proposed scheme is tested using the Single-Machine Infinite Bus (SMIB) network. Communication between the Real-Time Digital Simulator (RTDS) and SEL-3555 RTAC is done using the IEC 61850 communication standard. The results obtained validate the proposed decentralized control architecture.

The robustness of the proposed interarea oscillations damping controller is verified through a Hardware-in-the-Loop (HIL) Lab-scale implementation comprising of the RTDS, SEL-3555 RTAC, SEL-3355 rugged computer. The system modelling and simulation are performed in the RSCAD software whilst the RTDS GTNET card in conjunction with the SEL-3555 controller are used for Analog GOOSE messages exchange between the system and the controller. With Gaussian noise added as input to the generator to emulate small load variations responsible for the rotor angle instability, the results showed that the rotor angle remain stable. Furthermore, when subjected to faults, the recovery time is less than 500 ms.

Lastly, a comparison between the results previously obtained through digital simulation via the MATLAB/SIMULINK software is carried out.

This thesis deliverables contribute to opening and bringing the knowledge behind Model-Reference Adaptive Control (MRAC) and its application in power system in conjunction with the IEC 61850 standard as follows:

- Power system small signal rotor angle stability enhancement. The developed testbed shows that the stability of the rotor angle can be guaranteed irrespective of the contingencies.
- Application of the IEC 61850 standard in a MRAC control strategy for power system small signal stability studies. Analog GOOSE messages are utilized for data

exchange between the power system and the controller thus leveraging the interchangeability, interoperability, future proofing, and security that this communication protocol brings. The developed algorithm can therefore be used in a different controller provided it is IEC 61850 compatible with little or no changes.

- Development of control and automation devices for smart grids by Original Equipment Manufacturers (OEMs). The proposed algorithm can be incorporated into Intelligent Electronic Devices (IEDs) for instance to work in conjunction with power system stabilizers and the synchronous generator excitation systems.

ACKNOWLEDGEMENTS

I wish to thank:

DEDICATION

This thesis is dedicated to my parents, specifically my mother who never had the opportunity to attend university, and to my siblings for their love, encouragement, and prayers.

Lastly to my soon to be wife Estelle Mbaya Tshiaba for her support and understanding.

TABLE OF CONTENTS

CHAPTER THREE: DEVELOPMENT OF A NONLINEAR MODEL OF THE SYNCHRONOUS GENERATOR

CHAPTER FOUR: DESIGN OF A NONLINEAR REFERENCE MODEL BASED CONTROLLER AND MODEL-REFERENCE ADAPTIVE CONTROLLER (MRAC) FOR POWER SYSTEM LOW FREQUENCY ELECTROMECHANICAL OSCILLATIONS DAMPING

CHAPTER SEVEN: HARDWARE IN THE LOOP (HIL) REAL-TIME SIMULATION OF THE CLOSED-LOOP SYSTEM

CHAPTER EIGHT: CONCLUSONS AND RECOMENDATIONS

LIST OF FIGURES

LIST OF TABLES

APPENDICES

GLOSSARY OF TERMS

Abbreviations	Definition/Explanation
AGC	Automatic Generation Control
CSAEMS	Centre for Substation Automation and Energy Management System
GOOSE	Generic Object-Oriented Substation Event
GTNET	Giga Transceiver Network Card
HIL	Hardware-in-the-Loop
IEEE	Institute of Electrical and Electronics Engineering
IP	Internet Protocol
IEC	International Electromechanical Commission
MATLAB	MATrix LABoratory
PMU	Phasor Measurement System
RTDS	Real-Time Digital Simulator
RTAC	Real-Time Automation Controller
SEL	Schweitzer Engineering Laboratories
FACT	Flexible Alternating Current Transmission System
SSC	Static Synchronous Compensator
SSSC	Static Synchronous Series Compensator
HVDC	High Voltage Direct Current System
PSS	Power System Stabilizer
MRAC	Model-Reference Adaptive Control
LQR	Linear Quadratic Regulator
TCSC	Thyristor Controlled Series Compensator
SCADA	Supervisory Control and Data Acquisition
WAC	Wide-Area Control
WAMS	Wide-Area Monitoring System
WADC	Wide-Area Damping Controller

CHAPTER ONE
INTRODUCTION

1.1   Introduction

Power systems have become exposed to the small-signal stability problem due to the deficiency of damping or synchronising torques (Vittal, 2000; Morison et al., 2004; Rueda et al., 2011); more specifically insufficient damping of oscillations(Kundur, 1994b). This vulnerability has a direct correlation to the current liberalized electricity framework as well as constrained transmission networks that lead power systems to operate closer to their designed technical limits.

Low-Frequency Electromechanical Oscillations (LFEOs) caused by small variations in the system load are inherent to power systems. The transition from a state of stability to an unstable one is doubtlessly caused by a change in the operating condition, thus the apparition of contingencies such as ringdown oscillations. A rapid system collapse is therefore expected if not properly damped (Rogers, 2000a).

As a consequence of the presence of high damping observed in power systems with short lines, oscillations do not cause any problem therein (Turunen, 2011). However, the system's power transfer capability can be highly affected as they do represent, under certain operating conditions, a serious threat to the system stability (Turunen, 2011). The stability of the rotor angle needs to be ensured as it oscillates because of LFEOs, and therefore the ability of the interconnected synchronous machines to remain in synchronism is of utmost importance. Factors like the initial operating condition, the transmission system strength and type of generator excitation control used are very deterministic of the nature of the given power system's response to small disturbances. As for generators connected radially to a large power system, and in the absence of automatic voltage regulators, the instability is due to a lack of sufficient synchronizing torque.

However, with sufficient acting voltage regulators, this problem becomes one of ensuring sufficient damping of oscillations. Furthermore, instability is normally evidenced by oscillations of increasing amplitude (Kundur, 1994b). These small disturbances are either classified by their interaction characteristics as (Turunen, 2011; Khairudin, 2016):

- Inter-area mode oscillations
- Local plant mode oscillations
- Interplant mode oscillations

- Torsional (sub-synchronous) mode oscillations
- Control-mode oscillations

Or by the operating conditions of the power system as:

- Ambient (spontaneous) oscillations
- Transient oscillations
- Forced oscillations

The Chapter is structured as follows.

Section 1.2 introduces the problem awareness, while the problem statement is presented in Section 1.3. The research aims and objectives as well as assumptions are presented in Sections 1.4 and 1.5 respectively. The research delimitations are provided in Section 1.6, and the motivation in Section 1.7. Section 1.8 propounds assumptions made, while Section 1.9 highlights the contributions of this research. Section 1.10 provides an overview of the methodology used. Lastly, Section 1.11 presents the thesis outline and 1.12 concludes this chapter.

1.2   Awareness of the Problem

Inter-area oscillations have been the cause of major blackouts worldwide (M. Eremia and Shahidehpour, 2013), and various approaches have been proposed since the 1960s in an effort to address their effects.

A very comprehensive summary on various blackouts around the globe including the major three i.e., the August 14th, 2003, US and Canada blackout, the September 28th, 2003, Italian blackout as well as the November 4th, 2006, European Incident is presented in (M. Eremia and Shahidehpour, 2013). From the study on these events, the voltage collapse, cascade overload, frequency collapse, loss of synchronism, and system separation were listed as causes with the first two being the major types of incidents leading to such contingencies.

For all those incidents, power system oscillations were recognized as both initiating and triggering events. Thus, the need to have a control scheme that can perform the relevant corrective action within the first few seconds of their appearance. The weakness of the previously used SCADA/EMS systems with their 1-5 second(s) measurement intervals which are inadequate for any real-time control led to the use of wide-area measurement

system (WAMS), which is based on the phasor measurement units (PMUs) and the global positioning system (GPS), to provide a more effective and quick real-time system information to system operators and realize real-time control (M. Eremia and Shahidehpour, 2013).

The oscillations responsible for system collapse are those involving groups of generators in one area swinging against another group in another area. These oscillations are called inter-area oscillations. The oscillations involving generators within an area, also called local oscillations, are generally damped accurately by the standard Power System Stabilizers (PSSs) using generators' speed or speed deviation as input. Those stabilizers' outputs feed the excitation system.

1.3   Problem Statement

Oscillations may be harmful as they can become sources of instability. To avoid any consequences resulting in their presence in a given power system network i.e., total, or partial power system breakdown, the monitoring of the ability of the system to damp out the oscillations is of the utmost importance (Turunen et al., 2008). Once detected, relevant actions must be taken, be they protective or control ones.

Though the presence of high damping in power systems with short lines is often counted as a mitigating factor to oscillations, these oscillations do represent however, under certain operating conditions, a serious threat to the system stability as they limit the power transfer capacity (Turunen, 2011).

Real-time monitoring of power system oscillation is very important for power system operation; thus, software solutions i.e., algorithms are used for real-time oscillation monitoring. They continuously perform signal processing on measured synchrophasor signals to provide estimate coefficients needed to implement an accurate damping of oscillations (Vanfretti et al., 2012).

Past as well as ongoing works in the field of power system oscillations damping are an illustration of its importance.

1.4 Research Aim and Objectives

The purpose of this research is to damp oscillations that may cause harm to the power system. Guaranteeing this is accomplished will result in ensuring that the maximum power transfer principle is achieved. The latter being highly impacted by whether an efficient and robust oscillations damping exists in a power system.

1.4.1 Aim of the Thesis

The aim of this research is to design, develop and implement an adaptive algorithm for power system inter-area oscillations.

1.4.2 Objectives

The aim is attained through theoretical derivations and practical implementation

1.4.2.1 Objectives – Theoretical Analysis

- To review methods used for power system interarea oscillations analysis, and oscillations damping.
- To conduct a review on existing methods and algorithms for inter-area damping controllers and stability improvement.
- To formulate a Model Reference Adaptive Control (MRAC) scheme based on the error dynamics to enhance the stability of the synchronous generator rotor angle.
- To design an MRAC algorithm using the IEC 61850-8-1 Generic Object-Oriented Substation Event (GOOSES) messages.

1.4.2.2 Objectives – Practical Real-Time Implementation

- Using digital simulation in the MATLAB environment, to design and implement the proposed interarea oscillations damping algorithm.
- Validate the performance of the control scheme through case studies.
- Conversion of the 4th order Single-Machine Infinite Bus (SMIB) Simulink model to C/C++ Code.
- Modelling of the converted model in the RSCAD software.
- Conversion of the reference model as well as the adaptation and controller to an IEC 61131-3 compatible code via the SIMULINK PLC coder.
- Implementation of the proposed algorithm for real-time interarea oscillations damping using external hardware interfaced to the RTDS.
- Development of the proposed algorithm in the Schweitzer Engineering Laboratories (SEL) 3555 RTAC.

- Real-time lab-scale implementation of an IEC 61850 based MRAC algorithm using the RTDS and SEL-3555 RTAC in a Hardware-in-the-Loop (HIL).

1.5   Hypothesis

The hypotheses are formulated for the design of the proposed decentralized MRAC for power system oscillations damping as follows:

- **Hypothesis 1:** The proposed algorithm will enhance the stability of the rotor angle. Thus, it is suitable to be deployed in a larger power system in a decentralized architecture to damp interarea oscillations.
- **Hypothesis 2:** Beside its ability to mitigate the impact of LFEOs, the proposed controller can also enhance transient stability.

1.6   Delimitation of the Research

1.6.1   Within Scope

Within the scope of the project, the below tasks will be completed:

- Derivation of the 3rd and 4th order dynamics equations of the synchronous generator in a SMIB configuration.
- Derivation of the reference model and design of an optimal control strategy that ensures its stability.
- Derivation of adaptive controller and adaptation law that would ensure the synchronous generator rotor angle remains stable as it follows the behaviour of the optimally controlled reference model.
- Design and implementation of the proposed algorithm in the MATLAB environment through digital simulations.
- Use the IEC 61850-8-1 GOOSE messages to exchange data between the RTDS and SEL-3555 RTAC.
- Design, and implementation of the proposed algorithm using a real-time Hardware-in-the-Loop (HIL) platform.

1.6.2   Beyond Scope

- Other power system stability issues such as voltage and frequency stability are considered beyond the scope of this research.
- Security and possible delay in the GOOSE publishing/subscriptions are regarded as out of scope of this thesis.

1.7 Motivation for the Research Project

Due to its complexity, and as it may vary both in size and structural component, the monitoring of a power system operation and stability is indeed challenging (Kundur, 1994).

In today's liberalized electricity market framework and the physical limitations in the transmission network, power systems tend to operate closer to their technical limits. This represents a real threat to the system stability as it becomes vulnerable to the small signal stability problem (Vittal, 2000; Morison et al., 2004; Rueda et al., 2011). It is then critical to have an accurate and fast automatic identification of the disturbance type to help the operator perform corrective action to eliminate their impacts on the system (Avdakovic et al., 2014). These corrective measures can either be achieved through physical devices that act on the system at generating units at the interconnection between areas or through algorithms.

The need for real-time monitoring and control of rotor stability is motivated by the potential damage its instability could lead to. As emphasized by Eremia and Shahidehpour (2013) when analysing various blackouts around the globe, oscillations involving a group of generators in one area oscillating against another group in another area are responsible for system collapse.

1.8 Assumptions

The proposed algorithm is based on the 4th order representation of the synchronous generator in an SMIB configuration and can be deployed in a larger power system in a decentralized control architecture. In light of the preceding, the following assumptions are considered in this thesis:

- It is assumed that existing control strategies exist i.e., Power System Stabilizers, Excitations, and Governors.
- The publishing/subscription of GOOSE messages by the RTDS GTNET card and those by the SEL-3555 RTAC is secure.
- Virtually no delay or data loss in the publishing or subscription to GOOSE messages.

1.9   Contributions of the Research Project

The research contributions are as follows:

- The review methods used for power system interarea oscillations analysis, and oscillations damping.

- The review on existing methods and algorithms for inter-area damping controllers and stability improvement.

- The formulation of a Model Reference Adaptive Control (MRAC) scheme based on the error dynamics to enhance the stability of the synchronous generator rotor angle.

- The development and real-time implementation of a MRAC based control algorithm, lab-scale testbed for Hardware-in-the-Loop (HIL) simulation with industrial grade equipment such as the RTDS, SEL-3555 RTAC and SEL-3355 rugged computer for power system interarea oscillations damping.

- The utilisation of the IEC 61850 communication standard for data exchange between the RTDS GTNET card and the SEL-3555 RTAC. The SEL-3355 rugged computer is used as the interface but is also the platform where various configuration software are installed.

1.10   Design and Methodology

Data collected when the network is subjected to small disturbances in the form of Gaussian noise are utilised in the validation of the proposed algorithm. These small changes are the very roots of LFEOs.

Besides those data, other contingencies such as faults, setpoint changes and their combination is also part of the proposed case studies.

Real-time data from the modelled network in the (RTDS) RSCAD software as well as controlled data from the SEL-3555 RTAC are published and subscribed using the IEC 61850-8-1 GOOSE messages.

In the overall architecture, these GOOSE messages are utilised to ensure that the stability of the synchronous generator rotor angle is maintained prior and after disturbances are applied onto the network.

1.11 Thesis Outline

This thesis is divided into eight chapters. A brief summary of each chapter's content is provided below:

1.11.1 Chapter One

This chapter introduces the thesis, presents the awareness of the problem, research problem to solve, hypotheses, its aim and objectives, its scope, assumptions made, a brief overview of the design and methodology used, and its contributions.

1.11.2 Chapter Two

This chapter provides a review of power system stability, with an emphasis on the small signal (rotor) angle stability problem in power system, and the cause and impact on the overall stability of the network if not mitigated. Moreover, a review of previous work done in the area of detection of oscillations caused by small signal stability problem as well as control strategy to mitigate their effects are presented. Lastly, it introduces the specifications of the IEC 61850 MRAC based interarea oscillations damping control algorithm.

1.11.3 Chapter Three

Chapter 3 presents the synchronous generator's (rotor) dynamics with an emphasis on the mathematical representation/model suitable for both the control systems of generators, their synthesis and dynamic analysis of the small-signal stability as well as their modelling in the full range of electromechanical oscillations. This is followed by the modelling and simulation of the derived model in MATLAB®.

1.11.4 Chapter Four

This chapter introduces the Lyapunov stability theory and its application in the design of both the Lyapunov-based nonlinear controller as well as the Model-Reference Adaptive Controller (MRAC) for the 3rd and 4th order representation of a synchronous generator respectively in a Single-Machine Infinite Bus (SMIB) configuration.

1.11.5 Chapter Five

The performance of the Lyapunov reference-model based nonlinear controller together with the MRAC through simulations in the MATLAB environment are presented in this chapter. Various case studies for the validation of the performance of these controllers are described and analysed.

1.11.6 Chapter Six

This chapter provides the details on the implementation of the proposed real-time small signal rotor angle enhancement algorithm for power system interarea oscillations damping. Also provided are detailed explanation around the communication medium used, real-time modelling as well as various equipment used in the design and implementation of the IEC 61850 based adaptive control strategy lab-scale testbed for power system inter-area oscillations damping.

1.11.7 Chapter Seven

This chapter presents the results of the practical implementation of the IEC 61850 MRAC based power system interarea oscillation damping algorithm using the SEL -3555 RTAC together with the RTDS in a Hardware-in-the-Loop (HIL) configuration. Moreover, discussions around the obtained results are also covered.

1.11.8 Chapter Eight

This chapter highlights the conclusion drawn from the research, its deliverables as well and direction for future work. References and appendices are presented thereafter.

1.12 Conclusion

Presented in this chapter are the research background, aim and objectives, hypotheses, and delimitations. Also provided are the motivations that led to this research, and assumptions made. Finally, a brief overview of its contributions is also given.

Chapter Two provides an ample review of previous work done in the power system rotor angle stability problem and approaches employed in mitigating the impact of oscillations originating from this stability problem.

Two main methodologies are used throughout the literature: the decentralized and centralized control schemes. Both of these are discussed in this chapter.

REVIEW ON THE LITERATURE FOR THE POWER SYSTEM ELECTROMECHANICAL OSCILLATIONS AND DAMPING

2.1 Introduction

In this chapter, a review of methods used in power system oscillations damping is presented with an emphasis on the Low Frequency Electromechanical Oscillations (LFEOs).

After a brief review on the (power system) stability problem, methods used for oscillations estimation, oscillations damping, as well as stability improvement are presented. Thereafter, a comparative analysis of these methods is conducted, highlighting their strengths and shortfalls.

2.2 Power System Stability Problem

2.2.1 Background

As defined by an IEEE/CIGRE Joint Task Force on Stability Term and Definitions, power system stability is "the ability of an electric power system, for a given initial operating condition, to regain a state of operating equilibrium after being subjected to a physical disturbance, with the most system variables bounded so that practically the entire system remains intact" (Mircea. Eremia and Shahidehpour, 2013; Kundur et al., 2004). Three quantities are important for power system from the point of view of defining and classifying power system stability status: (i) angles of nodal voltages δ, also called power or load angles; (ii) frequency f; and (iii) nodal voltage magnitudes V (Machowski et al., 2008). Instability in a power system may be manifested in many ways depending on the system configuration and operating mode (Kundur, 1994).

The power system stability problem can be classified and studied in three categories:

- Rotor angle stability
- Frequency stability
- Voltage stability

2.2.2 Rotor Angle Stability

The rotor angle stability is the ability of the interconnected synchronous machines of a power system to remain in synchronism. The stability here involves the study of the electromechanical oscillations inherent in power systems. A fundamental factor in this

problem is the way the power outputs of synchronous machines vary as their rotors oscillate (Kundur, 1994).

Two essential elements characterize a given synchronous machine: the rotor and stator.

An important characteristic that has a bearing on power system stability is the highly nonlinear relationship between the interchanging power and the regular positions of the rotors of synchronous machines (Kundur, 1994).

The study of the rotor angle stability implies the analysis of the transient stability and the oscillatory stability of the power system.

2.2.2.1 Large Signal Rotor Angle Stability

It is the ability of a power system to maintain synchronism during severe disturbances. Even so, the stability depends on both the initial operating state of the system and the severity of the disturbance.

Usually, the system is altered so that the post disturbance steady-state operation differs from that prior to the disturbance.

The resulting system response involves large executions of generator rotor angles and is influenced by a nonlinear power-angle relationship. In transient stability, the study period of interest is usually limited to 3 to 5 seconds following the disturbance; although it may extend to about 10 seconds for very large systems with dominant modes of oscillations (Kundur, 1994).

2.2.2.2 Small Signal Rotor Angle Stability

The nature of a system response to small disturbances depends on several factors including the initial operating condition, the transmission system strength and the type of generator excitation control used. For a generator connected radially to a large power system, in the absence of automatic voltage regulators (i.e., with a constant voltage field) the instability is due to lack of sufficient synchronizing torque. With continuously acting voltage regulators, the small disturbance stability problem is one of ensuring sufficient damping of system oscillations and instability is normally through oscillations of increasing amplitude (Kundur, 1994).

These small disturbances are either classified by their interaction characteristic (Turunen, 2011) as:

- Inter-area mode oscillations

- Local plant mode oscillations

- Interplant mode oscillations

- Torsional (sub-synchronous) mode oscillations

- Control-mode oscillations

Or by the operating conditions of the power system as:

- Ambient (spontaneous) oscillations

- Transient oscillations

- Forced oscillations

Figure 2.1 illustrates this classification
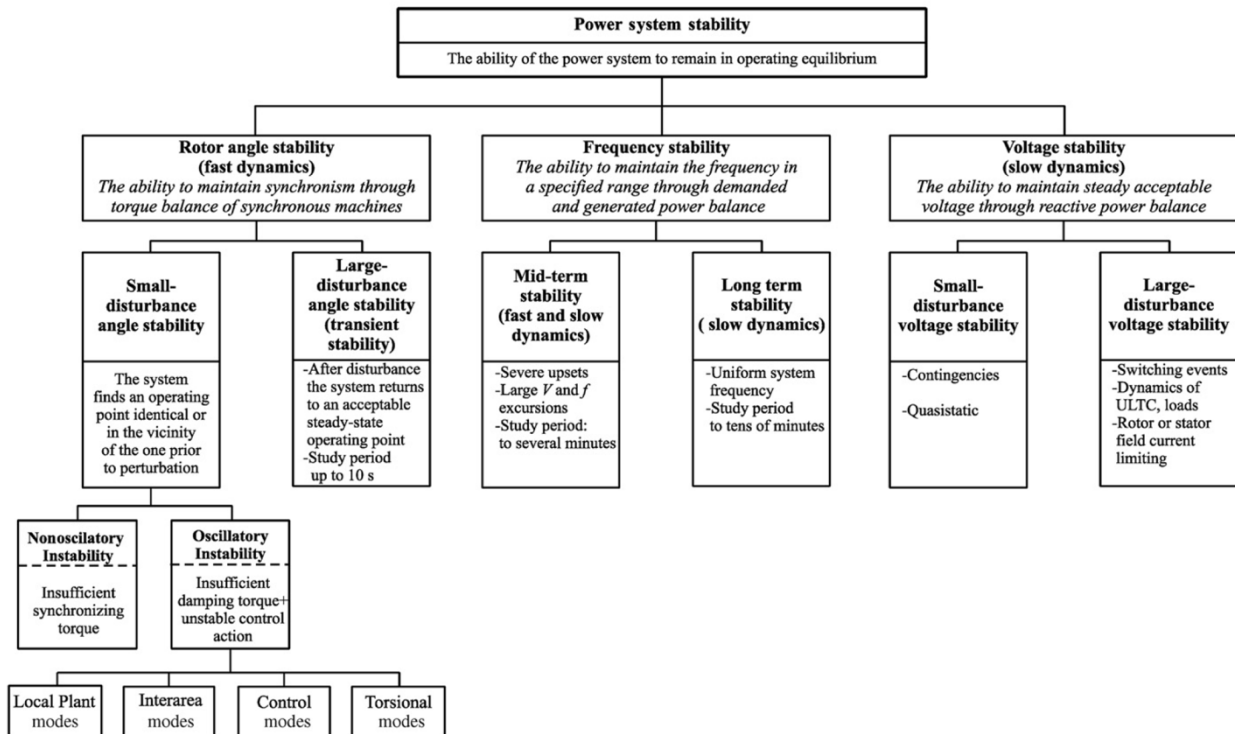
Figure 2.1: Dynamics phenomenon-based power system stability classification (Eremia and Shahidehpour, 2013)

The small signal rotor angle stability is considered in this research, with an emphasis on inter-area oscillations. Local oscillations are also explored since the proposed decentralized control scheme is built upon a modified version of existing local controllers.

2.2.3 Frequency Stability

This stability problem refers to the ability of a power system to maintain steady state frequency after a significant unbalance between generation and load occurs in the system (Eremia and Shahidehpour, 2013; Kundur et al., 2004).

In some regions, Europe for example, the frequency control is performed in three levels i.e., primary, secondary, and tertiary. Within the first few seconds after disturbance occurs, the primary control acts to stabilize the frequency. The secondary control automatically balances the generation and load thus bringing the system frequency within predefined limits. The task of tertiary control depends on the organizational structure of a given power system and the role that power plants play in this structure (Machowski et al., 2008). Its purpose is to maintain sufficient automatic active power reserve in the system to oppose any power unbalances and frequency deviations (Eremia and Shahidehpour, 2013).

This stability type is not covered in the research.

2.2.4 Voltage Stability

This is the ability of a power system to maintain steady state acceptable voltages at all buses in the system under normal operating conditions and after being subjected to a disturbance (Machowski et al., 2008). Instability occurs when the load supplied does not meet the load demand, and this is related to imbalance of reactive power (Mircea. Eremia and Shahidehpour, 2013; Eremia et al., 2000; Cutsem and Vournas, 2001; Taylor et al., 1994).

Two contingencies are often responsible for this stability problem: a sudden change in load demand with the loss of load in an area as a typical scenario as well as the limitations on the load supply capacity resulting to a tripping of a transmission line (Eremia and Shahidehpour, 2013).

This phenomenon can be classified per the severity of the disturbances or be analysed on various time frames. With respect to the first category, the voltage stability can be classified into large-disturbance and small-disturbance voltage stability. The large-disturbance voltage stability relates to the ability of a power system to maintain steady state voltages within acceptable limits following large disturbances whereas the small-disturbance voltage stability to its ability to maintain steady voltages when subjected to small disturbances. Based on a time frame categorization, voltage stability is classified into short-term voltage stability and long-term voltage stability. The first involves fast acting

equipment that take part in the system dynamics, and the study period of interest is of a few seconds. The second one involves slow acting equipment but may also be caused by controlled load, and its duration ranges from a dozen of seconds to minutes depending on whether the equipment is close to loads such as tap changing transformers or generator over-excitation limiters (Mircea. Eremia and Shahidehpour, 2013; Morison et al., 1993)

Figure 2.2 illustrates the voltage stability classification as described above.
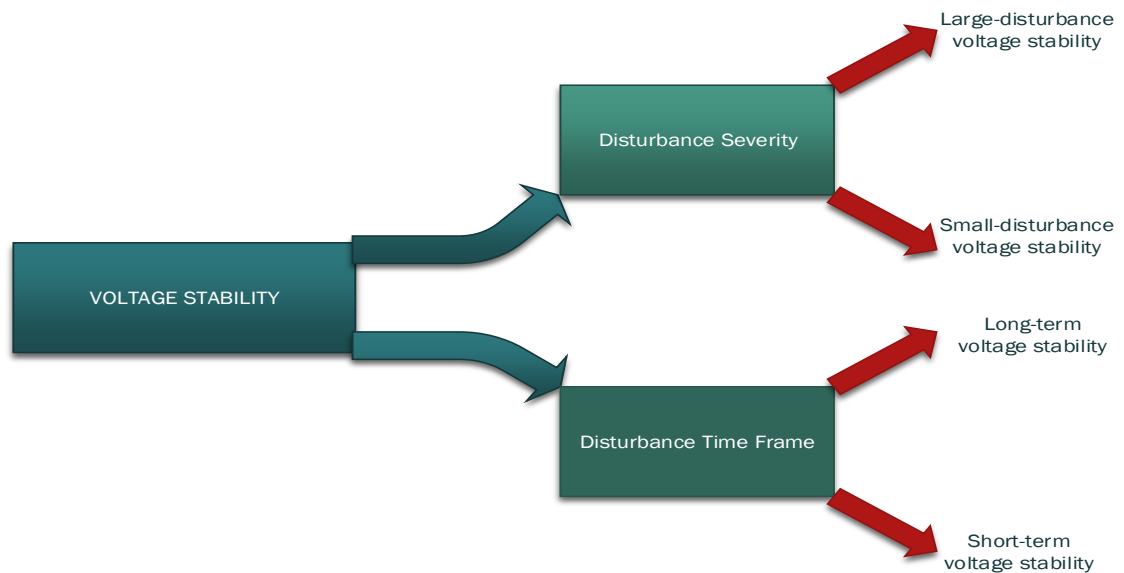
Figure 2.2: Voltage stability classification

This stability type will not be covered in the current work, but as mentioned in the conclusion to this section, it is associated to the rotor angle stability. An enhancement in the angle stability will necessarily improve the voltage stability in the power system.

2.2.5 Correlation Between Rotor, Voltage and Frequency Stability

Both voltage and (rotor) angle stabilities are closely associated as one may lead to the other, and a parallelism can be drawn between the minimum voltage criteria for voltage stability and the equal area criteria for large-disturbance angle stability (Eremia and Shahidehpour, 2013). The equal area criterion deals with excessive energy generation and the critical clearing time, whilst the minimum voltage criterion deals with excessive energy demand and critical reactive power support (Mircea. Eremia and Shahidehpour, 2013; Xu et al., 1994). Excessive generation implies acceleration of the generators' rotor, leading to angular instability and therefore loss of synchronism, while excessive load demand causes

severe voltage drops and load reduction, which may lead to voltage collapse (Eremia and Shahidehpour, 2013).

This is illustrated in Figure 2.3, with the shadowed surface in (a) indicating the minimum energy necessary to be activated immediately to help the power system survive to voltage stability problems.

Other work such as Wei et al., (2016) revealed that rotor angle stability can positively influence frequency stability. In fact, extending their previous works pertaining to LFEOs damping and load-following characteristics, their proposed generator rotor angle droop control is said to also be able to replace primary frequency control thus replacing the current model of Automatic Generator Control (AGC).
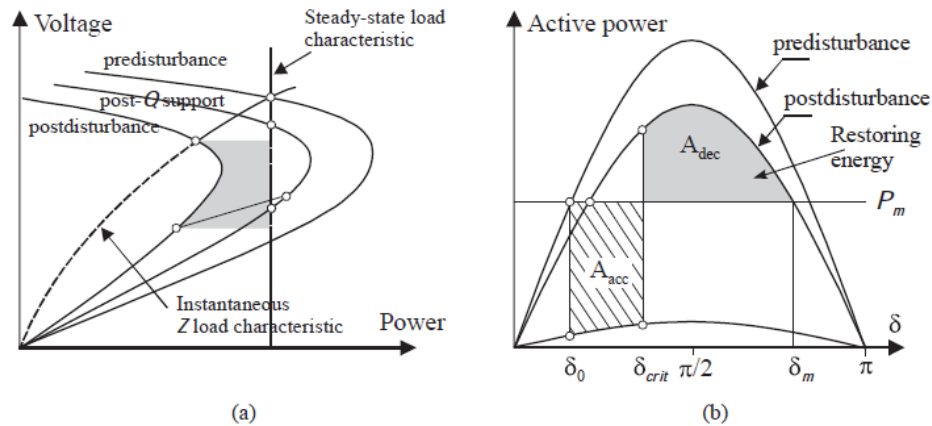
Figure 2.3: Parallelism between voltage stability (a) and angle stability (b) (Eremia and Shahidehpour, 2013), Where: P_m is the mechanical power; P_e is the electrical power; A_{dec} is the accelerating area when $P_m < P_e$; A_{acc} is the accelerating area when $P_m > P_e$; δ_o is the initial value of the rotor angle δ; δ_m is the rotor angle at full swing

2.3 Literature Review of Methods Used for Power System Inter-Area Oscillations Analysis, Oscillations Damping, Damping Controllers and Stability Improvement
2.3.1 Power System Inter-Area Oscillations Analysis
2.3.1.1 Overview

Power system electromechanical oscillations stability and damping are estimated either with time-domain simulations, linear analysis, or signal analysis. The first two methods rely on the mathematical model of the power system whilst with the last one, the focus is on the signal itself which provides all the necessary information regarding oscillations in each

network (Kundur, 1994). A detailed description or these methods can be found in (Turunen, 2011).

Both the frequency of oscillations and the damping ratio represent the key features needed when analysing the stability of a given power system using either one of the aforementioned methods.

The designation "synchronous machine" comes from the fact that the frequency of the stator electrical quantities is synchronized with the rotor mechanical speed as stated in (Kundur, 1994) "When the rotor is driven by a prime mover (turbine), the rotating magnetic field of the field winding induces alternating voltages in the three phase armature windings of the stator. The frequency of the induced alternating voltages and of the resulting currents that flow in the stator windings when a load is connected depends on the speed of the rotor."

Therefore, for two or more interconnected synchronous machines, the corresponding rotors must be in synchronism as must voltages and currents of all machines have the same frequency; respective rotors synchronized to that frequency.

The stability being a condition of equilibrium between opposing forces, the interconnected synchronous machines maintain synchronism with one another through restoring forces, cancelling accelerating or decelerating forces with respect to the other machines (Kundur, 1994).

Oscillatory instability is then caused by insufficient damping torque or unstable control action whereas non-oscillatory stability is due to insufficient synchronizing torque.

2.3.1.2 Time-Domain Simulation

The network is modelled by the algebraic equations that form a non-linear mathematical model of the system, and numerical integration methods are employed to simulate the system behaviour in the time domain. The simulations are used in producing data for the performance analysis of the damping estimation method. Also, they enable the calculation of reference damping for comparisons.

Realistic results of the power system behaviour are attained with this method. The main potential drawbacks are due to some deficiencies in the system model that could lead to conservative power transfer limits and/or to reduced system security (Turunen, 2011).

2.3.1.3 Linear Analysis

The same power system model is used similarly to time-domain simulations though non-linear differential and algebraic equations are linearized around the equilibrium point and a set of linear differential and algebraic equations is obtained (Turunen, 2011).

Although important components (generators, AVRs, governors, loads) in a power system have very non-linear characteristics, when the power system is operating under the ambient conditions, the resulting oscillations are essentially linear allowing the use of linear analysis methods in assessing oscillation stability and damping (Pal and Chaudhuri, 2005; Rogers, 2000).

The results achieved with the linear analysis are applicable only when the assumption of linear behaviour is valid.

2.3.1.4 Signal Analysis

Instead of the system model, only signals measured from the power system are used. This method provides critical information for early detection, mitigation, and avoidance of large-scale cascading failures and forms the basis of smart, wide-area automated analysis and control systems (Messina, 2009).

More realistic results are achieved than from the simulation-based methods though their main drawback is that they might produce inaccurate results (Turunen, 2011). The methods making use of the signal analysis approach are classified in two categories: parametric and non-parametric methods. The first work on data is to make inferences about the system generating the data while the later work on data is to estimate characteristics of the data itself (Thambirajah et al., 2011).

Using this approach, signal processing and mathematical tools such as the Wavelet Transforms can be used in the estimation of the frequency, and methods such as the Half-Power Bandwidth (HPB) for the computation of the oscillations damping ratio.

The purpose of this thesis being a controller design for power oscillations damping, a set of nonlinear equations describing the generators behaviour are used for the controller design, and synchrophasor measurements exported from the power system network are used to analyse the behaviour of the system in real-time and for real-time implementation of the designed control.

2.3.1.5 Summary on Power System Interarea Oscillations Analysis

Various methods used in power system oscillations estimation were presented in this section.

The usefulness of the signal-based method is justified by the need of performing the oscillation damping in real-time based on the actual system data rather than simulated ones nor those obtained by mathematical manipulations as in time-domain simulations or linear analysis.

On the basis of the above, the designed controller's inputs are therefore going to be synchrophasor measurements from the RTDS.

2.4   Power System Inter-Area Oscillations Damping Controllers and Stability Improvement

2.4.1   Overview

Several methods have been used since the early 1960s in an aim to damp oscillations in a power system. Depending on the aim of the controller design, Power System Stabilizers (PSSs), or power electronics devices such as Flexible Alternating Current Transmission System (FACTs) are utilized to mitigate the effect of oscillations within the power system. These controllers can also be in a form of mathematical expressions developed to control one or more parameters of the synchronous generator that would result in damping of the oscillations.

2.4.2   Literature Survey

Inter-area oscillations refer to oscillations between two or more groups or generators in different areas swinging against each other. As a result, instability of rotor angles is caused. Research interests and the number of publications of research papers in mitigating these oscillations for the period 1965-2021 are illustrated in Figures 2.4 and 2.5; with the last thirty years showing a huge growth in the number of the publications.
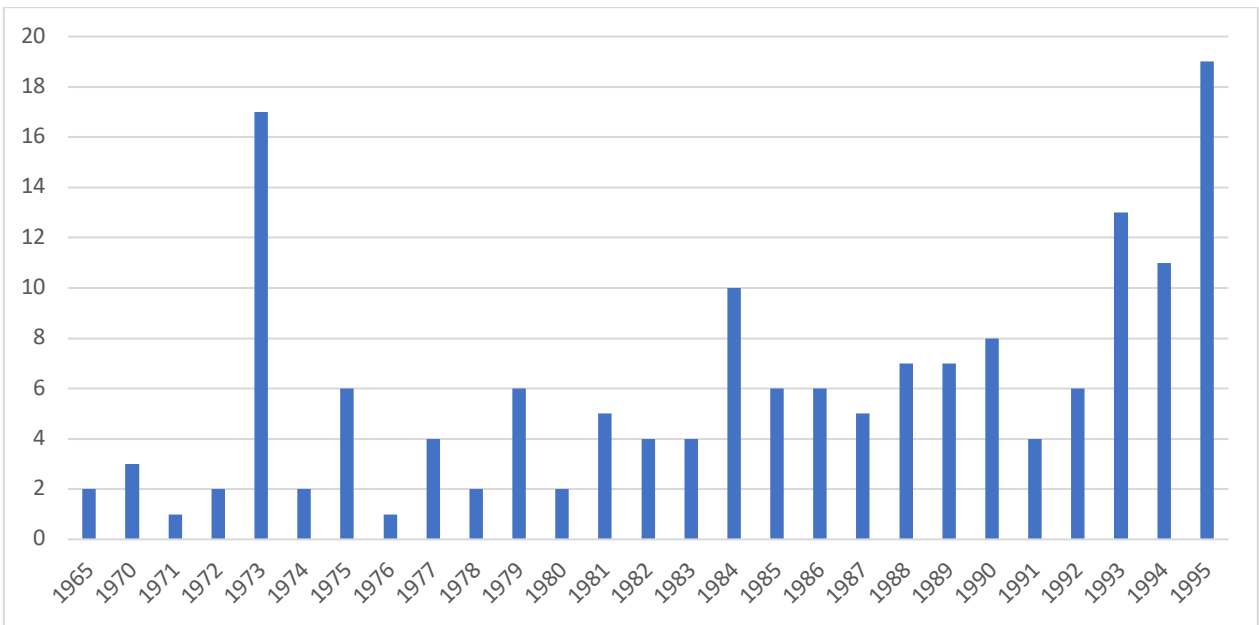
Figure 2.4: Survey of research articles per year for power systems inter-area oscillations damping from 1965-1995
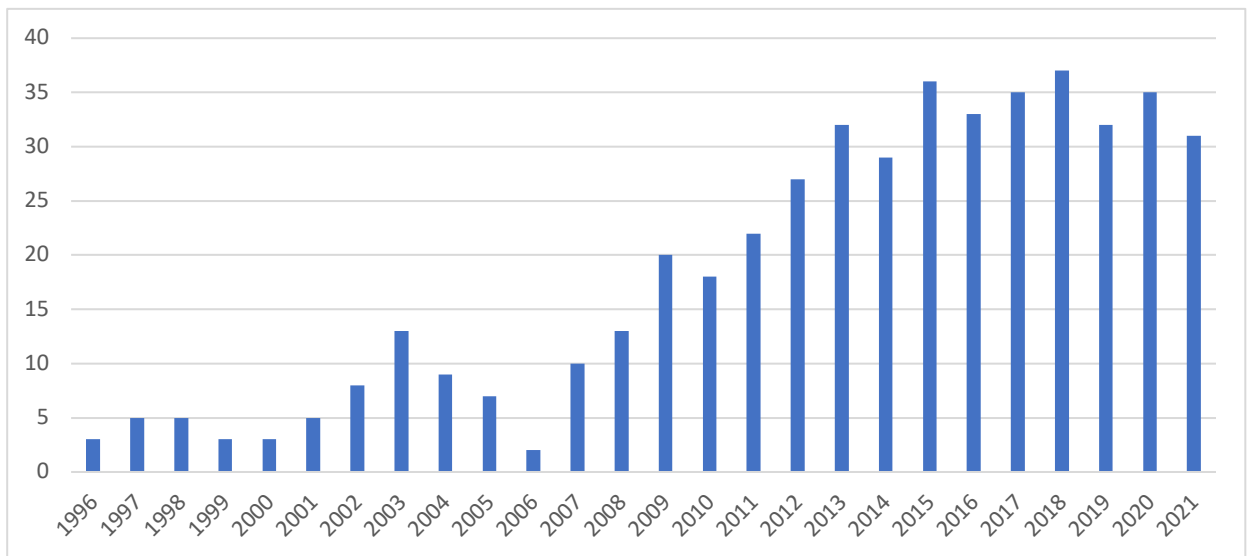
Figure 2.5: Survey of research articles per year for power systems inter-area oscillations damping from 1993-2021

From the literature, the methods used in the design of oscillation damping controllers can be grouped in two categories:

- Decentralized controllers
- Centralized controllers

Table 2.1: Literature review of power system inter-area oscillations controller design categories

Category	Methods	Type of Measurements	Real-Time Implementation
Decentralized controllers	A controller is placed at each generating unit.	Local measurements: • Rotor angle, • Rotor speed • Q-axis voltage	Only recently with local PMUs. Previous works were either implemented through digital simulations (MATLAB) or time-domain simulations
Centralized controllers	Unified controller often placed at the tie-line between groups of generators (areas). This controller can be measurement-based or FACT-based.	Wide area measurements from PMU based systems such as: • Exported power between areas • Voltage angle difference between areas, …	Wide-area inter-area oscillations control

Besides the clear distinction in methods used in the mitigation of power system oscillations, there are in literature those whose approach could fit in the aforementioned classification. Amongst the most recent ones is Fan et al., (2020) whose proposed Power Oscillation Damping (POD) controller is said to be applicable to actuators such as Power System Stabilizers (PSSs) as well as High-Voltage Direct Current (HVDC) transmission. PMUs are used to extract the modal signal which is then used as a feedback signal to modulate the PSS's output or the transferred power on the HVDC lines.

Other than the aforementioned, some control architectures made use of both methods to produce a coordinated controller like in Morsali et al., (2014) where a coordination of a Proportional-Integral-Differential (PID) controller and the IEEE type PSS2B Power System Stabilizer (PSS) together with a Thyristor Controlled Series Capacitor (TCSC) was used to improve the power system small-signal stability. Based on the IEEE Standard 421.5TM-2005 which recommended the use of the IEEE type PSS2B with the accelerating power as the stabilizing signal, the authors integrated a TCSC for power flow control and the mitigation of sub-synchronous resonance.

Using the same hybrid architecture, Gholinezhad et al., (2017) presented a novel Power System Stabilizer-Static Synchronous Series Compensator (PSS-SSSC) controller to

improve the damping of LFEOs. While sensing and signal transmission are considered as effectiveness parameters, Biogeography-Based Optimization (BBO) algorithm is used for searching the optimal controller parameters.

2.4.3 Review of the Decentralized-Based Controllers

Power system inter-area oscillations damping is a field that has attracted many researchers as a result of damaging effects they can cause to a grid when poorly damped (Anon, 1997; Zhu et al., 2003). Relative mature technologies and devices such as Power System Stabilizers (PSS) are included to the additional excitation system of synchronous machines in an aim of providing enough damping ratio to mitigate the effect of local oscillations (Tse and Tso, 1993; Fereidouni et al., 2013). Nevertheless, as illustrated by Kundur et al., (1989), an appropriate selection of a PSS parameters can enhance a grid's performance when subjected to disturbances.

Kamwa et al., (1999) proposed a coordination of decentralized controllers using constrained optimization to enhance stability in a grid. From the results obtained, the improvement of the modal performance index which affects the controller's operation as well as a good identification of optimization constraints can ensure a *closed-loop robustness* of the power system.

Lei et al., (2001)'s work focused on finding the best tuning procedure possible for FACTs device stabilizers and PSSs to improve the damping of overall power system oscillations in a grid in a way that is said to be *optimal and globally coordinated*.

Hussein et al., (2010) introduced the use of an adaptive fuzzy controller as a PSS to damp LFEOs. Based on a comparative study with the IEEE standard Multi-Band Power System Stabilizer (MB-PSS), the proposed is said to be more robust since it can mitigate oscillations at different operating points. While the nominal model of the power system is identified on-line via a variable structure identifier, the gains of the Feedback Linearization (FL) control law are tuned via Particle Swarm Optimization (PSO) method.

Hashmani and Erlich (2012) presented decentralized PSS controllers that utilize both locals and remote signals to enhance damping of inter-area oscillations with each local controller designed per the oscillation mode of interest. While the local signals can either be the generator speed or angle, the remote signal is chosen in such a way that it carries the

maximum of information of the given inter-area mode. The electrical power exchanged at the tie-line is said to be a good candidate and has been proven by many studies such as in Banga-Banga et al., (2017). The proposed decentralized controller is based on the H_∞ control theory.

(Fereidouni et al., 2013) emphasized on the identification of a dominant generator that influences the overall power system stability as well as the low-frequency oscillations. This identification process is done through eigenvalues analysis which helps understand the characteristics of the power system dynamic states.

In Morsali et al., (2014), the power system small signal stability is said to be enhanced through a coordinated control between a PID-based IEEE type PSS2B and a Thyristor Control Series Capacitor (TCSC)-based controller whose parameters are tuned using Improved Swarm Particle Optimizations (IPSO) algorithm by minimizing Integral of Time Squared Error (ITSE) performance index.

Wei et al., (2014) introduced a novel approach to mitigate Low-Frequency Electromechanical Oscillations (LFEOs) by using a combination of PSS and a rotor angle controller whose input comes from a PMU. The input used for the governor is the measured absolute rotor angle rather than the generator speed.

Baek (2014) proposed a coordination between PSSs and controllable Series Capacitive Reactance Compensators (SCRCs) to damp LFEOs. While the PSS is used to damp local oscillations, the SCRC is used to enhance the transfer capacity for the transmission line and provide sufficient damping of these oscillations.

Gholinezhad et al., (2017) work revealed that a coordination between PSS and SSSC controller can improve LFEOs. Instead of the Particle Swarm Optimization (PSO) algorithm, optimal controller parameters are obtained through a Biogeography-Based Optimization (BBO) algorithm while sensing and signal transmission delays are taken into consideration as *effective parameters*.

Shi et al., (2016) presented a Flywheel Energy Storage System (FESS)-based stabilizer whose location is dependent on Damping Torque Indices (DTIs) under multi-operational

conditions. While its tuning is dependent on the operational condition of these indices, a compensation angle is used to determine its parameters.

Extending the work done by Wei et al., (2014) and considering the impact of the penetration of Distributed Generation (DG) on frequency fluctuation, Wei et al., (2016b) proposed a modified rotor angle controller by removing the integration bloc of the previously proposed structure. That block is said to lead to negative damping torque for very low frequency input and frequent turbine valve adjustment. This newly proposed structure, also referred to as Rotor Angle Droop (RAD) control is said to suppress LFEOs more effectively bit also maintain generation load balance as well as the system frequency to rated value.

Wu et al., (2016) used an input-output analysis to examine the Power Spectral Density (PSD) and variance amplification of stochastically forced systems in the design of decentralized controllers. These local controllers were designed considering the sparsity-promoting optimal control.

Farahani and Ganjefar, (2017) proposed a different type of PSSs based on adaptive fuzzy sliding mode control to damp power system oscillations. This is achieved through a Wavelet Neural Network (WNN) sliding-mode control. Using this approach, learning rates of the WNN are extracted using a discrete-type Lyapunov function while an observer is used to adjust the bound of uncertainties in real time.

Similar to Hashmani and Erlich (2012), Maherani and Erlich (2018) presented a decentralized wide area controller considering both locals and remote signals. However, the controller was based on H_∞ and the communication delays from the wide-area control signals are taken into consideration as parametric uncertainties.

Khosravi-Charmi and Amraee (2018) proposed a four-bands PSSs to damp inter-area oscillations in the presence of wind power penetration. While the modal information is extracted via subspace identification using ambient phasor measurement data, the wide-area tuning of PSSs is achieved with the very extracted modal data. The parameters of the multi-band PSSs are optimized using an evolutionary algorithm in a wide-area basis.

Yohanandhan and Srinivasan (2018) introduced a measurement-based indirect adaptive wide-area supplementary damping controller based on an Adaptive Model Predictive Controller (AMPC). The online identification of plant model parameters is achieved through a Recursive Polynomial Model Estimator (RPME).

Prakash et al., (2019) introduced a Time-Delay Compensator (TDC) using the Simevents toolbox of MATLAB to cater for the variability in communication delays in obtaining wide-area signals of interest. Furthermore, these remote signals are obtained via Phasor Measurement Units (PMUs) in compliance with the IEEE C37.118 protocol. Lastly, the location of the proposed Wide-Area Power System Stabilizers (WAPSSs) is determined based on geometric measures of controllability and observability whereas their parameters are tuned using the Jaya Algorithm (JA).

Zacharia et al., (2020) proposed a novel coordination between generators' governors and PSSs to damp inter-area oscillations. The role of the governor in the damping of these oscillations was highlighted in Wei et al., (2014) and Wei et al., (2016), but in contrast to their approach whereby the (rotor) angle was utilized as the input, the traditional structure was utilized i.e., the rotor speed deviation.
This coordination is said to increase the damping capability in presence of inter-area oscillations.

Decentralized methods for power systems inter-area oscillations damping are discussed and evaluated in Table 2.2 below.

Table 2.2: Decentralized controllers for Power system inter-area oscillations damping

Author(s)	Aim	Methods	Innovation	Drawback	Power System Model Considered	Simulation/Real-time implementation
Kamwa I., et al, 1999	Decentralized controllers for Power system inter-area oscillations damping	Power system stabilizing controllers based on parameter optimization of compensators with generalized structures. Three models namely the single-machine infinite bus system, multiple PSSs coordination, and the four PSSs of the Kundur's two area test system were used to validate the algorithm	Two loops PSS based on speed deviation and accelerating power to damp both local and interarea oscillations	The use of time-domain simulation in the validation of the proposed method may lead in deficiencies in the system model that could then lead to conservative power transfer limits and/or to reduce system security (Turunen, 2011).	1. Single machine-infinite bus 2. IEEE benchmark two-area system	Time-domain simulation
Xianzhang L., et al., 2001.	Decentralized controllers for Power system inter-area oscillations damping	Parameter-constrained nonlinear optimization algorithm for a global tuning procedure for FACTS Device Stabilizers and power system stabilizers in a multi-machine power system.	Improvement in damping of overall power system oscillations	Time-domain simulation	Three-area 400KV power system (Erche M., et all, 1992)	Time-domain simulation
Hussein *et al.*, 2010	Decentralized controllers for Power system inter-area oscillations damping	Indirect adaptive fuzzy-based PSS to damp inter-area oscillations	The inputs to the fuzzy controller are taken from online measurement thus making the controller useful for a real-time control scheme.	1. Small variations in load were not taken into consideration in the validation of the proposed architecture 2. The controller might not be efficient since relying solely on the speed and speed deviation	IEEE benchmark two-area system	Digital simulation

Decentralized controllers for Power system inter-area oscillations damping cont'd

Author(s)	Aim	Methods	Innovation	Drawback	Power System Model Considered	Simulation/Real-time implementation
Miotto and Covacic, 2011	Analysis of the performance of a multi-machine power system with the presence of Thyristor Controlled Series Capacitor (TCSC) device working with both a Power Oscillation Damping (POD) and PSS controllers.	Classical control of phase compensation in the frequency domain whereas the TCSC/POD location is obtained by means of the residue of the transfer function of the open loop controller.	Power flow control and electromechanical oscillations damping	Single-input PSS (Speed based) installed on one generator in a multi-machine power system	IEEE benchmark two-area system	Digital simulation (MATLAB)
Hashmani and Erlich, 2012	Power system electromechanical oscillations damping through decentralized controllers.	Deployment of the fuzzy-based PSS (controller) at each generating unit	The use of H$_\infty$ control theory alongside an algebraic Riccati equation in the design of the robust controller	Relying only on generators parameters in the design of controller and not making provision for global signal such as *the power flow* could impact the robustness of the proposed controller.	16-machine, three-area	Digital simulation
Fereidouni *et al*., 2013	Low frequency oscillation damping with power system stabilizers.	A speed (deviation)-based PSS installed on the dominant generator with the greater influence on low-frequency oscillations	The determination of the generator that influences the overall power system stability as well as the low-frequency oscillations through eigenvalue analysis.	The overall stability might not be attained since all local and inter-area oscillations might not be efficiently mitigated.	IEEE 9-bus and 14-bus	Time domain simulation
Baek, S.M, 2014	Improvement of low-frequency oscillation following disturbances.	Coordination control of PSS and FACTS devices.	Small signal stability and transient stability improvement.	Open-loop controller on a Single Machine Infinite-Bus SMIB power system power system	SMIB power system model	Digital simulation (MATLAB)

Decentralized controllers for Power system inter-area oscillations damping cont'd

Author(s)	Aim	Methods	Innovation	Drawback	Power System Model Considered	Simulation/Real-time implementation
Morsali J., et al., 2014.	Power system small-signal stability enhancement with a novel PSS structure using the PID controller and the IEEE type PSS2B.	PID-PSS2B controller used in coordination with thyristor-controlled series capacitor (TSCS) -based power system oscillation damping (POD) controller.	The controller parameters are tuned concurrently via improved swarm optimization (IPSO) algorithm by minimizing the integral of time square error (ITSE) performance index	Time-domain simulations. Furthermore, only two generators are equipped with PSSs	IEEE benchmark two-area system	Digital simulation (MATLAB)
Wei, Q., et al., 2014	Inter-area low frequency suppression	Novel approach of generator governor control based on the absolute rotor angle in lieu of the speed to suppress inter-area oscillations	Turbine's valve opening dependent on whether the measured rotor angle is below a set point.	Time-domain simulation	IEEE benchmark two-area system	Time-domain simulation
Gholinezhad J., et al., 2015	Power system low-frequency oscillations damping with a coordinated PSS and SSC controller	Coordinated design of PSS and SSSC controller.	Sensing and signal transmission delays are considered as effectiveness parameters.	Time-domain simulation. Furthermore, the proposed PSS has a single-input configuration.	Single machine and two-machine	Digital simulation (MATLAB)
Shi L., et al., 2016	Design of an Energy Storage System (ESS) -based power system stabilizer to damp power system oscillations	(ESS)-based stabilizers whose location is dependent on Damping Torque Indices (DTIs) under multi-operational conditions. Its tuning depends on the operational condition of these indices, while a compensation angle is used to determine its parameters.	Utilization of the Flywheel Energy Storage System (FESS) in the power system oscillations damping controller	A dual-input PSS could have been used. Furthermore, the validation of the novel approach suffers from a lack of real-world implementation	1. IEEE benchmark two-area system 2. New England ten-machine power system.	Eigenvalue analysis and nonlinear simulations

Decentralized controllers for Power system inter-area oscillations damping cont'd

Author(s)	Aim	Methods	Innovation	Drawback	Power System Model Considered	Simulation/Real-time implementation
Wei, Q., et al., 2016	Suppress LFEOs and potentially replace the existing Automatic Generator Control (AGC)	Using the absolute rotor angle as an input to a Proportional Derivative (PD) compensator bloc, the turbine valve's opening is controlled hence the mechanical power supplied.	The use of the novel Rotor Angle Droop (RAD) control in ensuring that the power system angle stability is maintained	Time-domain simulation	IEEE 39 New England power system model	Digital simulation (MATLAB)
Wu X., et al., 2016	Use of non-modal tools to analyse and control inter-area oscillations	Input-Output analysis using power spectral density and variance amplification of stochastically forced systems.	Analysis of interarea oscillations by studying their power spectral densities and output covariances	Time-domain simulation. Furthermore, the novel approach may have to be implemented in a real-world environment.	IEEE 39 New England power system model	Time-domain simulations
Farahani and Ganjefar, 2017	Power system oscillations damping through intelligent PSSs	Power system oscillations damping is achieved through Adaptive Fuzzy Sliding Mode Controller (AFSMC) with a PI switching surface.	The use of a hybrid Wavelet-Neural Network sliding-mode control to damp power system oscillations	Small variations in load were not taken into consideration in the validation of the proposed architecture.	IEEE benchmark two-area system	Time-domain simulations
Maherani and Elrich, 2018	Design of a decentralized wide area damping controller for LFEOs damping	Deployment of fixed-order decentralized H_∞ controllers for each area in the (interconnected) study network	Open loop shaping based H_∞ controller	Small variations in load were not taken into consideration in the validation of the proposed architecture.	IEEE benchmark two-area system	Time-domain simulations
Yohanandhan R.V., Srinivasan L., 2018	Low-frequency interarea oscillations damping	Decentralized measurement-based indirect adaptive wide-area controllers	The use of adaptive model predictive controllers with the online identification of plant model parameters achieved through Recursive Polynomial Model Estimator (RPME)	Non-real-world implementation The third order model of a synchronous generator suffice in the design of its controller Possible slowness in the control action with over 20 seconds elapsing before faults are cleared	179-bus wNAPS system	Digital simulation (MATLAB)

Decentralized controllers for Power system inter-area oscillations damping cont'd

Author(s)	Aim	Methods	Innovation	Drawback	Power System Model Considered	Simulation/Real-time implementation
Khosravi-Charmi M., Amraee T., 2018	Power system low frequency oscillations damping	Use of multi-band power system stabilizer to damp Low Frequency Oscillations (LFOs)	Hierarchical control implementation using the multi-band configuration of power system stabilizers	1. The use of subspace for the identification of oscillations parameters may be less robust considering that power systems are highly nonlinear. The use of a signal-based approach would have been more consistent 2. Furthermore, the novel approach may have to be implemented in a real-world environment	IEEE 39 New England power system model	Digital simulation (MATLAB)
Prakash *et al.,* 2019	Design of a Wide-Area Power System Stabilizer (WAPSS) using synchrophasor data.	Using measurement from the IEEE C37.118.1 PMUs' compliant, a global signal is added onto the generator excitation system.	1. Time-Delay Compensator (TDC) to compensate for communication time-delays. 2. Jaya Algorithm (JA) for the location of the WAPSS	1. Rather than a time-domain simulation for the validation of the proposed approach, a real-world implementation could have been preferable. 2. Missing amongst the test conditions are small variations in loads.	IEEE 39 New England power system model	Digital simulation (MATLAB)
Zacharia L., et al., 2020	Compensation of local and inter-area oscillations	Coordinating the excitation signal and the steam valve output, hence coordinating all local controllers to increase small-signal stability of the power system.	The use of Wide Area Control (WAC) signals for the coordination of power system governors or their coordination with PSSs or directly with excitation systems.	Rather than using a speed deviation-based global signal and considering the approach proposed by Wei *et al.* (2016), an angle-based controller could have been better.	1. IEEE 39 New England power system model 2. IEEE benchmark two-area system	1. Digital simulation (MATLAB) 2. OPAL-RT (for real-time implementation)

2.4.4   Review of the Centralized-Based Controllers

In contrast to the decentralized controllers that focus on individual generating units' control to achieve the overall stability, controllers that are classified as *Centralized* aim at adding an additional level of control to the traditional ones (PSSs and excitation systems). These types of controllers are referred to as *Wide Area Controllers* and can be either a mechanical device such as FACTs devices installed at the lie-line or a software-based one.

Wang (2000) investigated the use and integration of a Static Synchronous Series Compensator (SSSC) damping controller in a power system to improve its oscillation stability. This is achieved by superimposing an auxiliary stabilizing signal on the already existing capability of the SSSC to change its reactance characteristics from capacitive to inductive when used in power flow control.

Juan et al., (2008) proposed an optimal control-based coordinated controller to damp power oscillations. The proposed is said to consider effective interactions between variables, while nonlinear dynamics associated with DC-link capacitor voltage is taken into consideration.

Yao et al., (2009) introduced an adaptive control based on Generalized Predictive Control (GPC) and model identification using remote signals from a Wide-Area Measurement System (WAMS). The proposed Wide-Area Damping Controller (WADC)'s output is added onto a selected generator excitation system. The effect of change of operating conditions, model uncertainties and robustness against time delay existing in remote signals feedback are said to be minimized with this controller.

Extending from their previous work, Yao et al., (2010) presented a WADC based on Network Predictive Control (NPC) in attempt to eliminate the impact of communication delays of the wide-area signals from the WAMS. Using this approach, enhancement in interarea oscillations damping under constant and random communication delays …

Hassan and Roy, (2016) investigated power systems' damping performance through eigenvalue analysis and presented a supplementary damping controller to work in conjunction with the Static Synchronous Compensator (STATCOM) which is placed at the (weakest) bus requiring highest reactive power. The supplementary Power

Oscillation Damping (POD) controller uses the real power deviation, and its structure is made of a gain, wash-out and compensator blocks.

Patel et al., (2018) introduced an H$_\infty$ controller with regional pole placement to ensure decent dynamic performance. Using non-synchronized feedback signals whose delays are approximated using the *Padé's model*, the proposed controller is called robust in mitigating inter-area oscillations.

Mahdavian et al., (2017) investigated the effect of SSSC for damping LFEOs. The proposed damping controller's is made of a POD whose output is connected to the SSSC and whose structure is made of a gain block, a washout, and a two-stage phase compensator blocks.

Laverde and Ríos, (2018) proposed the computation of a supplementary H$_\infty$ controller to be deployed at the Voltage Source Converters (VSCs) of the High Voltage Direct Current (HVDC) link in the transmission line.

To Enhance inter-area oscillation damping and mitigate the effect of actuator saturation, a Thyristor Controlled Series Capacitor (TCSC) WADC based is proposed by Maddela and Subudhi, (2019). Actuator saturation is said to have an impact on the controller performance, hence reducing its efficiency. The damping controller, which is formulated in the form of Linear Matrix Inequality (LMI), is designed so that it maximizes the region of attraction. The design of the very controller is solved as an optimization problem in the damping of oscillations.

Joseph et al., (2019) introduce a novel Adaptive-Linear Quadratic Regulator (A-LQR) that utilizes an Artificial Neural Network (ANN) in the generation of optimal feedback gain for the current operating condition. Input-output data obtained from the simulation under various operating conditions are used as training dataset used for the ANN.

Dobrowolski et al., (2019) evaluated three different controllers to damp LFEOs. While the first two made use of proportional gains which considered availability of measurements from different areas of the power system hence resulting in different optimization functions, the third approach is based on the Linear Quadratic Gaussian (LQG) considering the state-space representation of the system under investigation.

Maherani et al., (2020) made use of *nonsmooth optimization method* in designing a fixed order robust controller aimed at suppressing LFEOs in interconnected grid. To overcome some limitations of the H_2, H_∞, loop shaping, and $\mu-$synthesis, a Ricatti-based robust controller is proposed

Prakash et al., (2021) proposed a TCSC based power system stabilizer as a wide area damping controller, with the choice of the wide area feedback signal obtained through Geometric Measure of Observability (GMO).

Centralized methods for power systems inter-area oscillations damping are discussed and evaluated in Table 2.3.

Table 2.3: Centralized controllers for Power system inter-area oscillations damping

Author(s)	Aim	Methods	Innovation	Drawback	Power System Model Considered	Simulation/Real-time implementation
Wang, H.F, 1999	Investigation of the damping control function of a Static Synchronous Series Compensator in power systems.	Static Synchronous Series Compensator (SSSC) based damping controller. Two models were used for validation of the controller with the phase compensation method for the single-machine infinite-bus configuration and an objective function-based searching algorithm for the multi-machine setup	Utilization of the damping function of the SSSC in power systems while using its inherent features for power flow control	Only considered oscillations caused by a 3-phase short circuit on the transmission line while variation in loads are mostly responsible for interarea oscillations	Single machine-infinite bus	Digital simulation (MATLAB)
Juan L., et al., 2008	Development of a coordinated control scheme between the Static Synchronous Series Compensator (SSSC) and the excitation system of the generator.	Coordinated control scheme between SSSC and generator excitation system as well as an optimal control-based coordination controller.	Oscillation damping is achieved but also the reactive power on the tine-line is compensated by means of the SSSC	The equations used to describe a generator are non-realistic for control purposes. Furthermore, the results may need to be validated in a real-time environment.	Single Machine Infinite-Bus power system model	Digital simulation (MATLAB)
Yao W., et al., 2009	Power system wide area damping controller	Adaptive Wide Area Damping Controller (WADC) based on generalized predictive control and model identification whose output is fed into the generator excitation system.	This paper introduces the concept of WADC to mitigate the effects of interarea oscillations in a power system	Single input PSS (Speed-based) applied solely to generators 1 and 3	IEEE benchmark two-area system	Digital simulation (MATLAB)

Centralized controllers for Power system inter-area oscillations damping cont'd

Author(s)	Aim	Methods	Innovation	Drawback	Power System Model Considered	Simulation/Real-time implementation
Yao, W., et al., 2010	Power system wide area damping controller	Generalized Predictive Control (GPC) based WADC with a Network Delay Compensator (NDC) to compensate communication delays	This paper explores and presents ways to mitigate the impact time-delay caused by transmission of remote signals.	Single input PSS (Speed-based) applied solely to generators 1 and 3	IEEE benchmark two-area system	Digital simulation (MATLAB)
Hassan, M., Roy N.K., 2015	Damping performance of power systems through eigenvalue analysis	Use of a Static Synchronous Compensator (STATCOM) and Power Oscillation Damping (POD) controller based on the line active power	Improvement of the overall system damping performance by using a controller equipped with STATCOM	The control system associated to the STATCOM is only made of a gain, wash-out, and compensation blocks which could not be very efficient for a nonlinear system such as a power system	IEEE-14 bus test-system	Eigenvalue analysis
Patel A., et al., 2018	Power system interarea oscillations damping	Wide-Area Power System Stabilizer (WAPSS) controller synthesized using H∞ control with regional placement.	Mitigation of time delay variations of wide-area signals	Single input speed-based PSS	1. IEEE benchmark two-area system 2. IEEE 39 New England power system model	Digital simulation (MATLAB)
Mahdavian M., et al., 2017	Power system oscillations damping	Use of the Static Synchronous Series Compensators (SSSC) alongside a Power Oscillation Damping (POD) controller	Power system damping oscillations as well as the overall system stability.	The generating units aren't equipped with local controllers. This could lead to system instability.	Two generation substations with one major dynamic load	Digital simulation (MATLAB)

Centralized controllers for Power system inter-area oscillations damping cont'd

Author(s)	Aim	Methods	Innovation	Drawback	Power System Model Considered	Simulation/Real-time implementation
Laverde J.S., Ríos M.A., 2018	Electromechanical oscillations damping	H∞ controller on the VSC of HVDC links for electromechanical oscillations damping	The proposed can be applied to smart grids whereby large non-synchronous renewable energy sources are connected to large conventional power sources	Time-domain simulation. Furthermore, the novel approach may have to be implemented in a real-world environment	IEEE benchmark two-area system	Digital simulation (MATLAB)
Joseph et al., 2019	Inter-area low-frequency electromechanical oscillations damping	LQR-based adaptive wide area controller using PMU measurements	The use of an input-output trained Artificial Neural Network in the generation of optimal feedback gain for a given operating point	With the generators' states measured directly and exported via PMUs, the ANN could have been then used for the prediction of the control signal. Non-real-world implementation	5-area 16-generator 68-bus system	Digital simulation (MATLAB)
Dobrowolski et al., 2019	Inter-area oscillations damping in large power systems	Linear Quadratic Gaussian using the state space model of the system under investigation	Overall control of the power system achieved through a single command targeting the most influenced group of machines in the system	The overall control scheme relies entirely on PMUs, hence prone to communication problems	IEEE New England model	Digital simulation (MATLAB)

Centralized controllers for Power system inter-area oscillations damping cont'd

Author(s)	Aim	Methods	Innovation	Drawback	Power System Model Considered	Simulation/Real-time implementation
Maddela and Subudhi, 2019	Damping enhancement of inter-area oscillations through a robust wide-area TCSC controller.	Design of a dynamic output feedback controller incorporating anti-windup compensator to mitigate actuator saturation. Furthermore, quadratic Lyapunov criterion is utilized to ensure asymptotic stability of the closed-loop system in the form of LMIs.	1. Linear Matrix Inequality (LMI)-based controller for oscillation damping improvement 2. Actuation saturation taken into consideration in the controller design	Time-delays in communication of global signals is not taken into consideration	1. 16 machine 68 bus power system 2. IEEE benchmark two-area system	Nonlinear simulation
Maherani et al., 2020	Inter-area oscillations damping in large power systems	Linear Quadratic Gaussian using the state space model of the system under investigation	Overall control of the power system achieved through a single command targeting the most influenced group of machines in the system	The overall control scheme relies entirely of PMUs, hence prone to communication problems	IEEE New England model	Digital simulation (MATLAB)
Prakash et al., 2021	Design of a wide area power system stabilizer for inter-area oscillations damping	Use of a Thyristor-Controlled Series Capacitor (TCSC) based PSS that uses Geometric Measure of Observability (GMO) for the wide-area feedback signal.	The use of Geometric Measure of Observability in identifying the feedback signal that carries sufficient information.	Other than the fact that this was validated solely through digital simulations, the paper did not consider the introduction of Gaussian noise to ensure the robustness of the proposed controller.	IEEE benchmark two-area system	Digital simulation (MATLAB)

2.4.5 Discussion on the Design of the Power System Oscillation Controllers

Various controllers described in the literature and used to damp the power system LFEOs were presented.

Many power system models were investigated, from the Single Machine-Infinite Bus (SMIB) network to the multi-machine configurations; with the SMIB being a simplification of a generator or group of generators that are connected to an infinite (power) system. For the SMIB configuration, Juan et al., (2008) proposed a FACTS device in coordination with the generator excitation system to damp oscillations, Baek (2014) and Morsali et al., (2014) added a FACTS device in conjunction with the PSS, and Kamwa et al., (1999) introduced a two-loops PSS model based on the speed deviation and accelerating power to damp both local and interarea oscillations. Note, this last author validated his method in three (3) different configurations: the SMIB, the IEEE benchmark two-area system, and the Hydro-Quebec power system.

With respect to the multi-machine configurations, some focused on decentralized schemes while others proposed centralized controllers as illustrated in Table 2.2 and 2.3 respectively.

The decentralized PSS-based configuration as illustrated in Machowski et al., (2008) has some advantages as both local and interarea oscillations could be damped. In this model, the local PSS has two inputs, as implemented by both Kamwa et al., (1999) and Morsali et al., (2014). The centralized configuration on the other hand looks at the power system as a whole, while attempting to mitigate external factors such as the delay in communication and Wide-Area Damping Controllers (WADCs) are proposed therein.

However as proposed by most authors a single-input PSS configuration is less consistent in damping both local and interarea oscillations.

Nevertheless, further works on Khosravi-Charmi and Amraee (2018) with the innovative four-bands PSS as well as Laverde and Ríos (2018) with their proposed VSC in an HVDC configuration can be very useful considering the fact that only digital simulations were utilized in those cases.

The use of digital simulations with MATLAB seems to be prominent (Wang, 1999; Juan et al., 2008; Yao et al., 2009; Hussein et al., 2010; Miotto and Covacic, 2010; Yao et al., 2010;

Hashmani and Erlich, 2012; Baek, 2014; Morsali et al., 2014; Gholinezhad et al., 2015; Patel et al., 2017; Mahdavian et al., 2017, Khosravi-Charmi and Amraee, 2018; Laverde and Ríos, 2018; Yohanandhan R.V., Srinivasan L., 2018; Dobrowoslki et al., 2019; Joseph et al., 2019 ; Prakash et al., 2019; Maherani et al., 2020, Prakash et al., 2021). Time-domain simulations are presented by (Kamwa, 1999; Xianzhang, 2001; Fereidouni et al., 2013; Wei, Q., et al., 2014; Wu et al, 2016; Farahani and Ganjefar, 2017; Maherani and Elrich, 2018), eigenvalues analysis in (Hassan and Roy, 2015), nonlinear simulation (Maddela and Subudhi, 2019), or like Shi and Wu (2016) both time-domain and nonlinear simulations are done to simulate the closed loop system. Real-time implementation of the developed closed loop control systems is not considered in the existing literature except in Zacharia et al., (2020) where an OPAL-RT *real-time* simulator was used in the validation of the controller.

2.4.6 Focus of the Research Work in the Thesis

The research work in the thesis is based on the same approach as in Wei et al., (2018), Wu et al., (2016) and Kamwa et al., (1999) in the design of the decentralized controller. In contrast to what these three proposed, this thesis utilises Model-Reference Adaptive Control (MRAC) methods in the design of the proposed decentralized controller model.

MRAC has not been explored extensively for power system intearea oscillations damping. Amongst the few reported authors in literature that made use of it are Kim (2009) where it is utilized for adaptive control scheme of a Permanent-Magnet (PM) synchronous motor and Bhunia et al., (2021) for outer Photovoltaic (PV) voltage control loop and inner grid current loop.

To minimize the time delays in the transmission of the control signals, the communication between the power system modelled in the Real-Time Digital Simulator (RTDS) and the controller implemented in a Hardware in the Loop (HIL) real device is done via the IEC 61850 protocol

Lastly, in lieu of the time-domain simulation or eigenvalues analysis or digital simulations, the proposed control scheme is validated first through digital simulations then in a real-time HIL implementation with the RTDS, SEL-3555 Real-time Automation Controller (RTAC) and the industrial grade SEL-3555 rugged computer.

2.5 Conclusion

This Chapter introduced the concept of the power system stability problem with the focus on the small signal rotor angle stability. Also presented are their detection and estimation of their characteristics i.e., frequency and damping.

Two main categories of controllers namely *Decentralized* and *Centralized* control schemes are presented and reviewed. However, it has been shown that a few of the proposed algorithms cannot fit the aforementioned classification since utilizing both approaches to some extends for the purpose of damping power system electromechanical oscillations.

This chapter concludes with the specifications of the proposed interarea oscillations damping control algorithm.

Chapter 3 presents the synchronous generator's (rotor) dynamics with an emphasis on the mathematical representation/model suitable for both the control systems of generators, their synthesis and dynamic analysis of the small-signal stability as well as their modelling in the full range of electromechanical oscillations. This is followed by the modelling and simulation of the very derived model in MATLAB®.

DEVELOPMENT OF A NONLINEAR MODEL OF THE SYNCHRONOUS GENERATOR AND THE INTERCONNECTED POWER SYSTEM

3.1 Introduction

The design of a control mechanism requires some knowledge of the system the very control ought to be applied onto, hence the knowledge of its various inputs and outputs is of an uttermost importance. As defined by Burns (2001), a system is a collection of matters, parts, components, or procedures which are included within some specified boundary. The way in which its outputs (system response) respond in changes to its inputs is therefore critical.

In this thesis, the word "system" would refer to the IEEE benchmark two-areas system which is used in the modelling and simulation, and later in the design of the oscillations damping adaptive controller.

After a brief overview of various generator models and their respective usefulness in section 3.2, an in-depth illustration of the mathematical equations describing the 3rd and 4th order models of the synchronous generator's rotor dynamic is presented together with its state-space representation in section 3.3 and 3.4 respectively. The generator's 3rd order representation being the suitable model for stability studies whereas the 4th order is said to be sufficiently accurate to analyse electromechanical dynamics. This later being an extension of the previous.

Lastly, in section 3.5, the simulation of the aforementioned models with their various characteristics are shown.

3.2 Power System Electromechanical Oscillations and Rotor Dynamics

3.2.1 Background

Power system electromechanical oscillation studies deal with the analysis and control of low frequency oscillations that characterize interconnected power systems. Small variations in the system load are mainly responsible for such events.

Contingencies like ringdown oscillations happen often since a change in the system operating condition is likely to cause the transition from a state of stability to an instability state. If therefore these oscillations are not properly damped, they may lead to rapid system collapse (Rogers, 2000).

3.2.2. Rotor Dynamics and Swing Equation

The equation governing the motion of synchronous machines is based on this principle of dynamics:

$$accelerating\ torque\ =\ moment\ of\ inertia\ \times angular\ acceleration \qquad (3.1)$$

The electromechanical torque and the mechanical torque rotate in ~~the~~ opposite directions. The first is developed by the electromagnetic field and stator whereas the second is provided by the prime mover. Also, the direction of rotation of the rotor is the same as that of the mechanical torque.

The dynamics of the generator are described by the following equation (Eremia and Shahidehpour, 2013):

$$J\frac{d^2\vartheta_m}{dt^2} = \ T_a = \ T_m - \ T_e \qquad (3.2)$$

where

J: combined moment of inertia of the rotating mass, in kg.m2,

ϑ_m: rotor angle (mech. rad),

T_a: net accelerating torque,

T_m: mechanical torque supplied by the prime mover less the retarding torque due to rotational losses (Nm.),

T_e: net electrical or electromechanical torque (N.m), and t is the time, in seconds.

$$\text{Yet, } \vartheta_m = \ w_{sn}t + \ \delta_m \qquad (3.3)$$

where

w_{sn} = synchronous speed of the machine (mechanical radians),

δ_m = angular displacement from the synchronously rotating reference axis.

From Equation (3.3), the below can be derived:

$$\frac{d\vartheta_m}{dt} = \ w_{sn} + \ \frac{d\delta_m}{dt} \qquad (3.4)$$

$$\frac{d^2\vartheta_m}{dt^2} = \frac{d^2\delta_m}{dt^2} \tag{3.5}$$

Based on Equation (3.5), Equation (3.2) can be written as:

$$\therefore J\frac{d^2\delta_m}{dt^2} = T_a = T_m - T_e \tag{3.6}$$

With the rotor speed $w_n = \frac{d\vartheta_m}{dt}$, Equation (3.6) can be rewritten as:

$$Jw_n\frac{d^2\delta_m}{dt^2} = w_nT_a = w_nT_m - w_nT_e \tag{3.7}$$

The product of the speed and the torque gives a power, i.e.

$$w_nT_m = P_m \tag{3.8}$$
$$w_nT_e = P_e \tag{3.9}$$

where

P_m = shaft power input to the machine less rotating torque losses,

P_e = electrical power crossing the air gap (air power).

Equation (3.6) becomes:

$$Jw_n\frac{d^2\delta_m}{dt^2} = P_m - P_e \tag{3.10}$$

On steady state, the synchronous generator's rotor's speed is

$$rotor\ speed - synchronous\ speed \cong 0 \tag{3.11}$$
$$\therefore w_n - w_{sn} \cong 0 \tag{3.12}$$

From Equations (3.11) and (3.12) above, the difference between the rotor and synchronous speed becomes large when the machine loses synchronism.

With $Jw_n = M$, where M is the angular momentum at synchronous speed.

Equation (3.9) becomes:

$$M\frac{d^2\delta_m}{dt^2} = P_m - P_e \tag{3.13}$$

Though the rotor speed w_n is not constant; but assuming there is synchronism i.e., $w_n \cong w_{sn}$, M can be said to be constant. However, it varies for different types of generators, as for instance, from hydro generators to turbo generators.

Let H be the ratio between the stored kinetic energy (in mega joules) at synchronous speed over the machine rating MVA

$$H = \frac{\frac{1}{2}Jw_{sn}^2}{S_{mach}} \tag{3.14}$$

$$which\ can\ be\ written\ as\ H = \frac{\frac{1}{2}Mw_{sn}}{S_{mach}} \tag{3.15}$$

The following relationship can be deduced from these two equations:

$$M = \left(\frac{2H}{w_{sn}}\right).S_{mach} \tag{3.16}$$

where S_{mach} is the machine rating MVA

Equation (3.13) becomes:

$$\frac{2H}{w_{sn}}\frac{d^2\delta_m}{dt^2} = \frac{P_m - P_e}{S_{mach}} \tag{3.17}$$

The ratio of P_m over the machine MVA S_{mach} represents the per-unit mechanical power $P_m(pu)$ and the ratio of P_e over the machine MVA S_{mach} represents the per-unit electrical power $P_e(pu)$.

Equation (3.17) becomes:

$$\frac{2H}{w_{sn}}\frac{d^2\delta_m}{dt^2} = P_m(pu) - P_e(pu) \tag{3.18}$$

The above expression is called the **swing equation**.

In most cases, the *pu* does not appear but it is worth to mention that these powers are expressed per-unit. Also, it should be noted that the expression $P_m - P_e$ is applicable for

generators whilst for motors, it is reversed as $P_e - P_m$. That is because for motors, the electromagnetic torque sustains rotation while the mechanical load opposes rotation.

An inspection of the swing curves of all machines in the system indicates whether they remain in synchronism after a disturbance.

Small disturbances such as random variations in loads, for analysis purposes, are sufficiently small for linearization of the system equations.

3.2.3. High Orders Models of Synchronous Generators

The model presented in Equation (3.18) is referred to as the second order model. However, that is not the only representation of synchronous generator dynamics.

Electromechanical oscillations are inherent to all power systems. Henceforth, the analysis developed has the fundamental aim to understand these phenomena in qualitative terms. Consequently, reference can be made to the simplest model of a generator (or area) connected to an infinite system; this scheme is valid in case of both local and interarea oscillations (Eremia and Shahidehpour, 2013). Its representation is shown in Figure 3.3 below.
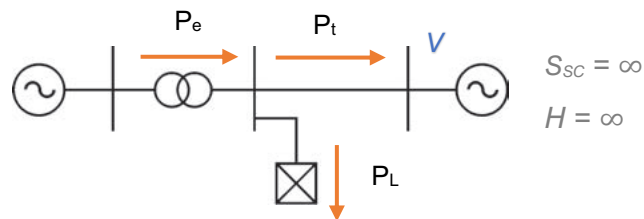
Figure 3.1: Generator or area connected to an infinite system

It is this very representation that is used to derive various orders of the synchronous generator; the word order referring to the set of differential equations used to characterize it.

Higher orders such as the 3rd, 4th, 5th, and 7th could also be used to describe it and details on these equations as well as fields of study each one is best suited for could be found in (Kundur, 1994; Machowski, 2008; Eremia and Shahidehpour, 2013).

The 3rd order model is said to be suitable for studying control systems of generators and their synthesis as well as in the dynamic analysis of the small-signal stability (Eremia and Shahidehpour, 2013). As for the 4th order model, it is sufficiently accurate to analyse electromechanical dynamics (Mircea. Eremia and Shahidehpour, 2013; Scott, 1979) and

as emphasized by Eremia and Shahidehpour (2013), it is suitable to model the generator in the full range of (local and interarea) electromechanical oscillations.

Hence, both the 3rd and 4th order models will be presented, with both representations utilized in the controllers' design even though the latter encompasses the first and covers both aspects of control systems of generators and the modelling of the generator in full range of electromechanical oscillations.

3.2.3.1 Third-Order Model of Synchronous Generators

Equations (3.19a) – (3.19c) represent the *third order model* of the synchronous generator, which assumes only the existence of the field winding in the rotor (without damper circuits in the d- and q-axis), is used in the dynamic analysis of the small-signal stability and less in transient stability studies (Eremia and Shahidehpour, 2013). Besides neglecting the effect of the damper windings by assuming the d-axis transient emf $e_d^{'}$ is assumed to remain constant. This model also neglects the damping produced by the rotor body eddy currents, even if an additional coil is used to represent the rotor body (Machowski et al., 2008).

Figure 3.3 shows the model of a synchronous generator connected to an infinite bus through a transmission line. As in (De León-Morales et al., 2001; Sanchez-Orta et al., 2002), this model is very useful for stability analysis and controller design.

The equations describing this model are as in the equations below:

$$\dot{\delta} = \omega \tag{3.19a}$$

$$\dot{\omega} = \frac{1}{J}\left(T_m - P_e - D\omega\right) \tag{3.19b}$$

$$\dot{e}_q^{'} = \frac{1}{T_{do}^{'}}\left(E_{FD} - e_q^{'} - \left(x_d - x_d^{'}\right)i_d\right) \tag{3.19c}$$

where

$$i_d = \frac{e_q^{'} - V\cos\delta}{x_q^{'}}$$

$$i_q = \frac{V\sin\delta}{x_q}$$

$$P_e \cong \frac{V}{x_d'} e_q' \sin \delta + \frac{V^2}{2} \left(\frac{1}{x_q'} - \frac{1}{x_d'} \right) \sin(2\delta)$$

δ : rotor angle

ω : rotor speed

i_d : current in the d-axis

i_q : current in the q-axis

V : voltage on transformer's terminal

E_{FD} : induced emf by the field current (field voltage)

T_{do}' : d-axis open-circuit time constant

$\dot{e}_q'$: q-axis transient emf

D : damping coefficient

P_e = air-gap power of the generator

Let $\underline{X} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \delta \\ \omega \\ e_q' \end{bmatrix}$, and $\underline{U} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} E_{FD} \\ P_m \end{bmatrix}$

Since in this model the influence of magnetic saturation is neglected, x_d, x_q, and x_d' can be assumed to be constant. However, the x_d, x_q, and x_d' used in Equations (3.19a), (3.19b), and (3.19c) are the augmented reactances with the line and transformer reactances added onto them (De Leon-Morales et al., 2001; Sanchez-Orta et al., 2002).
The state-space model is as in Equation (3.20a):

$$\dot{X} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\dfrac{D}{J} & 0 \\ 0 & 0 & \dfrac{1}{T_{do}'} \left(\dfrac{x_d}{x_d'} \right) \end{bmatrix} \begin{bmatrix} \delta \\ \omega \\ e_q' \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & \dfrac{1}{J} \\ \dfrac{1}{T_{do}'} & 0 \end{bmatrix} \begin{bmatrix} E_{FD} \\ P_m \end{bmatrix}$$

$$+ \begin{bmatrix} 0 \\ -\dfrac{1}{J} \left(\dfrac{V}{x_d'} e_q' \sin(\delta) + \dfrac{V^2}{2} \left(\dfrac{1}{x_q} - \dfrac{1}{x_d'} \right) \sin(2\delta) \right) \\ \dfrac{1}{T_{do}'} \left(\dfrac{x_d - x_d'}{x_d'} \right) V\cos(\delta) \end{bmatrix} \tag{3.20a}$$

46

This can be written in the form of $\dot{x} = Ax + Bu + F(\delta)$, where $F(\delta)$ is a vector with nonlinear elements.

Assuming the generator rotor angle and the (active) electrical power as outputs, it can be written:

$$y = \begin{bmatrix} \delta \\ P_e \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ \dfrac{V^2}{2\delta}\left(\dfrac{1}{x'_q} - \dfrac{1}{x'_d}\right)\sin(2\delta) & 0 & \dfrac{V}{x'_d}\sin(\delta) \end{bmatrix} \begin{bmatrix} \delta \\ \omega \\ e'_q \end{bmatrix} \qquad (3.20b)$$

with $y \epsilon R^2$ is the generator output.

3.2.3.2 Fourth-order Model of Synchronous Generators

As illustrated in Zacharia et al., (2020), Ghahremani and Kamwa (2016), Eremia and Shahidehpour (2013), and Machowski et al., (2008), the generator 4th order model representation of the synchronous generator can be seen as an extension of the 3rd order model, with the damper winding in the *q*-axis taken into consideration. This can be written as (Machowski et al., 2008):

$$\underline{X} = \begin{bmatrix} \delta & \omega & e'_d & e'_q \end{bmatrix}^T = [x_1, x_2, x_3, x_4] \qquad (3.21a)$$

$$\dot{x}_1 = \omega_0 x_2 \qquad (3.21b)$$

$$\dot{x}_2 = \frac{1}{J}(T_m - T_e - Dx_2) \qquad (3.21c)$$

$$\dot{x}_3 = \frac{1}{T'_{qo}}\left(-x_4 + \left(x_q - x'_q\right)i_q\right) \qquad (3.21d)$$

$$\dot{x}_4 = \frac{1}{T'_{do}}\left(E_{fd} - x_3 - \left(x_d - x'_d\right)i_d\right) \qquad (3.21e)$$

where

$$i_d = \frac{e'_q - V\cos\delta}{x'_q}$$

$$i_q = \frac{V\sin\delta - e'_d}{x'_q}$$

$$P_e \cong \frac{V}{x_d'} e_q' \sin \delta - \frac{V}{x_q'} e_d' \cos \delta + \frac{V^2}{2} \left(\frac{1}{x_q'} - \frac{1}{x_d'} \right) \sin(2\delta)$$

δ : rotor angle

ω : rotor speed

i_d : current in the d-axis

i_q : current in the q-axis

V : voltage on transformer's terminal

E_{FD} : induced emf by the field current (field voltage)

T_{do}' : d-axis open-circuit time constant

$\dot{e}_q'$: q-axis transient emf

$\dot{e}_d'$: d-axis transient emf

D : damping coefficient

P_e = air-gap power of the generator

Let $\underline{X} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \delta \\ \omega \\ e_d' \\ e_q' \end{bmatrix}$, and $\underline{U} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} E_{FD} \\ P_m \end{bmatrix}$

The state-space representation ca be written as in Equations (3.22a) and (3.22b):

$\underline{\dot{X}}$

$$= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \dfrac{-D}{J} & 0 & 0 \\ 0 & 0 & -\dfrac{1}{x_q' T_{qo}'}(x_q' + x_q - x_d') & 0 \\ 0 & 0 & 0 & -\dfrac{1}{T_{do}'}\left(\dfrac{x_d}{x_d'}\right) \end{bmatrix} \begin{bmatrix} \delta \\ \omega \\ e_d' \\ e_q' \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & -\dfrac{1}{J} \\ 0 & 0 \\ \dfrac{1}{T_{do}'} & 0 \end{bmatrix} \begin{bmatrix} E_{FD} \\ P_m \end{bmatrix}$$

$$+ \begin{bmatrix} 0 \\ \dfrac{1}{J}\left(\dfrac{e_d' V}{x_q'} \cos \delta - \dfrac{e_q' V}{x_d'} \sin \delta - \dfrac{V^2}{2} \sin 2\delta \left(\dfrac{1}{x_q'} - \dfrac{1}{x_d'} \right) \right) \\ \dfrac{V}{T_{qo}'}\left(\dfrac{x_q - x_d'}{x_q'} \right) \sin \delta \\ \dfrac{V}{T_{do}'}\left(\dfrac{x_d}{x_d'} - 1 \right) \cos \delta \end{bmatrix} \qquad (3.22a)$$

Assuming the generator rotor angle and the (active) electrical power as outputs, it can be written:

$$Y = \begin{bmatrix} \delta \\ P_e \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \dfrac{V^2}{2\delta} \left(\dfrac{1}{x'_q} - \dfrac{1}{x'_d} \right) \sin(2\delta) & 0 & \dfrac{-V\cos\delta}{x'_q} & \dfrac{V\sin\delta}{x'_d} \end{bmatrix} \begin{bmatrix} \delta \\ \omega \\ e'_d \\ e'_q \end{bmatrix} \qquad (3.22b)$$

3.2.4   Linearized Model of The Synchronous Generator

3.2.4.1 Overview

The phase plane analysis of a nonlinear system is related to that of its linearized system because the local behaviours of the nonlinear system can be approximated by the behaviours of its linearized systems in the vicinity of an equilibrium point. Even though the trajectories of the nonlinear solution can exhibit unpredictable behaviours due the fact that they have multiple equilibrium points.

In this subsection, derivations of the linearized models of both the 3rd and 4th order models of the synchronous generators are presented.

The linearized block representing a generator connected to an infinite bus is shown in Figure 3.4.
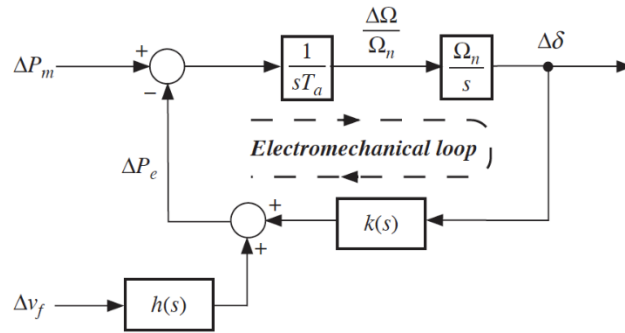
Figure 3.2: Linearized block representing a
synchronous generator connected to an infinite bus
(Eremia and Shahidehpour, 2013)

3.2.4.2 Linearized Third-Order Model

Equation (3.20a) can be linearized using the Jacobian as illustrated below.

To derive the matrices $\bar{A}$ and $\bar{B}$ and of the linearized model, let's define

$$\dot{X} = \bar{f}\left(\delta, \omega, e'_q\right) \equiv \bar{f}(x_1, x_2, x_3,): R^4 \to R^4$$

$$
\begin{cases}
\dot{X}_1 = f_1(x_1, x_2, x_3,) = x_2 \\[2mm]
\dot{X}_2 = f_2(x_1, x_2, x_3,) = \dfrac{-D}{J}x_2 + \dfrac{u_2}{J} - \dfrac{1}{J}\left(\dfrac{V}{x_d'}x_3\sin(x_1) + \dfrac{V^2}{2}\left(\dfrac{1}{x_d} - \dfrac{1}{x_d'}\right)\sin(2x_1)\right) \\[4mm]
\dot{X}_3 = f_3(x_1, x_2, x_3,) = \dfrac{-1}{T_{do}'}\left(\dfrac{x_d}{x_d'}\right)x_3 + \dfrac{1}{T_{do}'}u_1 + \dfrac{1}{T_{do}'}\left(\dfrac{x_d - x_d'}{x_d'}\right)V\cos x_1
\end{cases}
\tag{3.23}
$$

$$
\frac{\partial \bar{f}}{\partial X}\bigg/_{X=X_{ss},\,U=U_{ss}} =
\begin{bmatrix}
\dfrac{\partial f_1}{\partial x_1} & \dfrac{\partial f_1}{\partial x_2} & \dfrac{\partial f_1}{\partial x_3} \\[3mm]
\dfrac{\partial f_2}{\partial x_1} & \dfrac{\partial f_2}{\partial x_2} & \dfrac{\partial f_2}{\partial x_3} \\[3mm]
\dfrac{\partial f_3}{\partial x_1} & \dfrac{\partial f_3}{\partial x_2} & \dfrac{\partial f_3}{\partial x_3}
\end{bmatrix}
=
\begin{bmatrix}
\bar{a}_{11} & \bar{a}_{12} & \bar{a}_{13} \\
\bar{a}_{21} & \bar{a}_{21} & \bar{a}_{23} \\
\bar{a}_{31} & \bar{a}_{32} & \bar{a}_{33}
\end{bmatrix}
= \bar{A}
\tag{3.24}
$$

where X_{SS}, and U_{SS} are the steady state points of the state and control trajectory of the generator and

$\bar{a}_{11} = 0 = \bar{a}_{13} = \bar{a}_{32}$

$\bar{a}_{12} = 1$

$\bar{a}_{21} = \dfrac{-1}{J}\left(\dfrac{V}{x_d'}x_3\cos(x_1) + V^2\left(\dfrac{1}{x_d} - \dfrac{1}{x_d'}\right)\cos(2x_1)\right) /_{\text{Xss, Uss}}$

$\bar{a}_{22} = \dfrac{-D}{J}$

$\bar{a}_{23} = \dfrac{-1}{J}\left(\dfrac{V}{x_d'}\sin(x_1)\right) /_{\text{Xss, Uss}}$

$\bar{a}_{31} = \dfrac{-1}{T_{do}'}\left(\dfrac{x_d - x_d'}{x_d'}\right)V\sin x_1 \ /_{\text{Xss, Uss}}$

$\bar{a}_{31} = \dfrac{-1}{T_{do}'}\left(\dfrac{x_d}{x_d'}\right) /_{\text{Xss, Uss}}$

The $\bar{\bar{B}}$ matrix is obtained as below:

$$
\frac{\partial \bar{f}}{\partial U} =
\begin{bmatrix}
\dfrac{\partial f_1}{\partial u_1} & \dfrac{\partial f_1}{\partial u_2} \\[3mm]
\dfrac{\partial f_2}{\partial u_1} & \dfrac{\partial f_2}{\partial u_2} \\[3mm]
\dfrac{\partial f_3}{\partial u_1} & \dfrac{\partial f_3}{\partial u_2}
\end{bmatrix}
=
\begin{bmatrix}
0 & 0 \\[2mm]
0 & \dfrac{1}{J} \\[3mm]
\dfrac{1}{T_{do}'} & 0
\end{bmatrix}
\tag{3.25}
$$

3.2.4.3 Linearized Fourth-Order Model

Equation (3.25) can be linearized using the Jacobian as illustrated below.

To derive the matrices $\bar{A}$ and $\bar{B}$ and of the linearized model, let define

$$\dot{X} = \bar{f}\left(\delta,\, \omega,\, e_d^{'},\, e_q^{'}\right) \equiv \bar{f}(x_1, x_2, x_3, x_4): R^4 \to R^4$$

$$
\begin{cases}
\dot{X}_1 = f_1(x_1, x_2, x_3, x_4) = x_2 \\[4pt]
\dot{X}_2 = f_2(x_1, x_2, x_3, x_4) = \dfrac{-D}{J} x_2 + \dfrac{u_2}{J} - \dfrac{1}{J}\left(\dfrac{V}{x_d^{'}} x_4 \sin(x_1) - \dfrac{V}{x_q^{'}} x_3 \cos(x_1) + \dfrac{V^2}{2}\left(\dfrac{1}{x_d} - \dfrac{1}{x_d^{'}}\right)\sin(2x_1)\right) \\[10pt]
\dot{X}_3 = f_3(x_1, x_2, x_3, x_4) = \dfrac{-1}{x_q^{'} T_{qo}^{'}}(x_q^{'} + x_q - x_d^{'})x_3 + \dfrac{V}{T_{qo}^{'}}\left(\dfrac{x_q - x_d^{'}}{x_q^{'}}\right)\sin(x_1) \\[10pt]
\dot{X}_4 = f_3(x_1, x_2, x_3, x_4) = \dfrac{-1}{T_{do}^{'}}\left(\dfrac{x_d}{x_d^{'}}\right)x_3 + \dfrac{1}{T_{do}^{'}}u_1 + \dfrac{1}{T_{do}^{'}}\left(\dfrac{x_d}{x_d^{'}} - 1\right)V\cos x_1
\end{cases}
\quad (3.26)
$$

$$
\left.\dfrac{\partial \bar{f}}{\partial X}\right/ X = X_{ss}, U = U_{ss} =
\begin{bmatrix}
\dfrac{\partial f_1}{\partial x_1} & \dfrac{\partial f_1}{\partial x_2} & \dfrac{\partial f_1}{\partial x_3} & \dfrac{\partial f_1}{\partial x_4} \\[8pt]
\dfrac{\partial f_2}{\partial x_1} & \dfrac{\partial f_2}{\partial x_2} & \dfrac{\partial f_2}{\partial x_3} & \dfrac{\partial f_2}{\partial x_4} \\[8pt]
\dfrac{\partial f_3}{\partial x_1} & \dfrac{\partial f_3}{\partial x_2} & \dfrac{\partial f_3}{\partial x_3} & \dfrac{\partial f_3}{\partial x_4} \\[8pt]
\dfrac{\partial f_4}{\partial x_1} & \dfrac{\partial f_4}{\partial x_2} & \dfrac{\partial f_4}{\partial x_3} & \dfrac{\partial f_4}{\partial x_4}
\end{bmatrix}
=
\begin{bmatrix}
\bar{a}_{11} & \bar{a}_{12} & \bar{a}_{13} & \bar{a}_{14} \\
\bar{a}_{21} & \bar{a}_{22} & \bar{a}_{23} & \bar{a}_{24} \\
\bar{a}_{31} & \bar{a}_{32} & \bar{a}_{33} & \bar{a}_{34} \\
\bar{a}_{41} & \bar{a}_{42} & \bar{a}_{43} & \bar{a}_{44}
\end{bmatrix}
= \bar{A}
\quad (3.27)
$$

where X_{SS}, and U_{SS} are the steady state points of the state and control trajectory of the generator and

$\bar{a}_{11} = 0 = \bar{a}_{13} = \bar{a}_{14} = \bar{a}_{41} = \bar{a}_{32} = \bar{a}_{34} = \bar{a}_{42} = \bar{a}_{43}$

$\bar{a}_{12} = 1$

$\bar{a}_{21} = \dfrac{-1}{J}\left(\dfrac{V}{x_d} x_4 \cos(x_1) + \dfrac{V}{x_d} x_3 \sin(x_1) + V^2\left(\dfrac{1}{x_d} - \dfrac{1}{x_d^{'}}\right)\cos(2x_1)\right) \Big/ \text{Xss, Uss}$

$\bar{a}_{22} = \dfrac{-D}{J}$

$\bar{a}_{23} = \dfrac{1}{J}\left(\dfrac{V}{x_q^{'}}\cos(x_1)\right) \Big/ \text{Xss, Uss}$

$\bar{a}_{24} = \dfrac{-1}{J}\left(\dfrac{V}{x_d^{'}}\sin(x_1)\right) \Big/ \text{Xss, Uss}$

$\bar{a}_{31} = \dfrac{V}{T_{qo}^{'}}\left(\dfrac{x_d - x_d^{'}}{x_q^{'}}\right)\cos x_1 \Big/ \text{Xss, Uss}$

$\bar{a}_{33} = \dfrac{-1}{x_q^{'} T_{qo}^{'}}\left(x_q^{'} + x_q - x_d^{'}\right) \Big/ \text{Xss, Uss}$

$\bar{a}_{41} = \dfrac{V}{T_{do}^{'}}\left(\dfrac{x_d}{x_d^{'}} - 1\right) \Big/ \text{Xss, Uss}$

$\bar{a}_{44} = \dfrac{-1}{T_{do}^{'}}\left(\dfrac{x_d}{x_d^{'}}\right) \Big/ \text{Xss, Uss}$

The $\bar{B}$ matrix is obtained as below:

$$\frac{\partial \bar{f}}{\partial U} = \begin{bmatrix} \dfrac{\partial f_1}{\partial u_1} & \dfrac{\partial f_1}{\partial u_2} \\ \dfrac{\partial f_2}{\partial u_1} & \dfrac{\partial f_2}{\partial u_2} \\ \dfrac{\partial f_3}{\partial u_1} & \dfrac{\partial f_3}{\partial u_2} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & \dfrac{1}{J} \\ \dfrac{1}{T'_{do}} & 0 \end{bmatrix} \tag{3.28}$$

3.3   Modelling and Simulation of the Synchronous Generator

3.3.1   Synchronous Generator and Interarea Oscillations

Inter-area oscillation refers to the oscillations in which a coherent group of generators in one part of the system swing against other generators in the other part of the system (Kundur, 1994; Pal and Chaudhuri, 2005; Turunen, 2011). They are observed in systems where two or more groups of generators of closely coupled machines are interconnected by weak ties which implies high effective impedance between oscillating groups of generators (Kundur, 1994; Turunen, 2011).

Ambient inter-area oscillations occur in power systems due to poor damping and are excited mainly by varying loads.

"Inter-area oscillation is a complex and nonlinear phenomenon, and its damping characteristic is dictated by the strength of the transmission path, the nature of loads, the power flow through interconnection and the interaction of the loads with the dynamics of generators and their associated controls" (Pal and Chaudhuri, 2005; Messina 2009; Turunen, 2011).

Though generally stable, if the system is stressed too much, the oscillations may cause the Hopf bifurcation[1] to occur where the real parts of the complex conjugate eigenvalue pair cross the imaginary axis causing the system to become unstable (Mithulananthan et al., 2003; Turunen, 2011).

The high impedance causes the amortisseur windings of the generator to lose their effect on the inter-area oscillation damping. The same applies for the adverse interactions among the automatic controls, especially the AVRs. Irrespective of the adverse effects among the automatic controls, the uncontrolled system damping for such oscillations is commonly poor when the transmission path is weak. Furthermore, when the loading of the interconnecting lines grows, the damping decreases. This is because the angle difference

---

[1] Hopf or Poincaré–Andronov–Hopf bifurcation, named after Henri Poincaré, Eberhard Hopf, and Aleksandr Andronov. It is local bifurcation in which a fixed point of a dynamical system loses stability as a pair of complex conjugate eigenvalues of the linearization around the fixed point cross the imaginary axis of the complex plane.

between oscillating groups of generators grows and thus the voltage oscillations at each generator terminal; causing therefore the AVRs to act, producing negative damping (Turunen 2011).

To mitigate the effect of such oscillations, knowledge of the system characteristics is of ~~an~~ uttermost importance. Hence, system characteristics and simulation results of both the 3rd and 4th order models of the synchronous generators are presented in the next two subsections.

3.3.2. Third-Order Synchronous Generator
3.3.2.1 Representation
Parameters used in the modelling can be found in Table 3.1.

Table 3.1: Synchronous generator parameters (Eremia and Shahidehpour, 2013)

Acronym	Value
x_d	1.8 p.u.
$x_d^{'}$	0.3 p.u.
x_q	1.7 p.u.
x_q'	0.55 p.u.
x_q''	0.25 p.u.
x_d''	0.25 p.u.
D	0
H	6.5
S	900 MVA
T_{qo}'	0.4 s
T_{do}'	8 s
T_{do}''	0.03 s
T_{qo}''	0.05 s

Figure 3.5 is a representation of Equations (3.20a) and (3.20b).
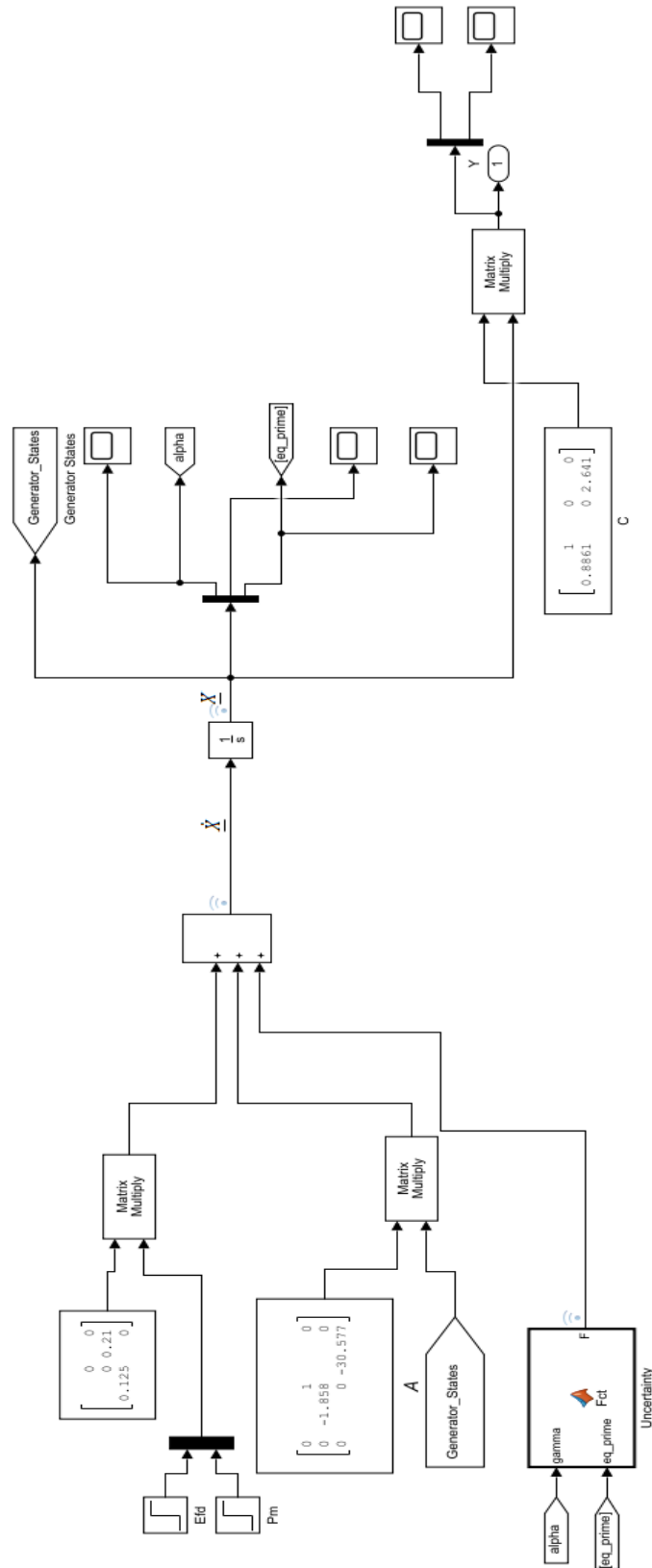
Figure 3.5: Third-order model representation of a synchronous generator

As mentioned earlier, the aforementioned Equations (3.20a) can be written in the form of $\dot{x} = Ax + Bu + F(\delta)$, with $F(\delta)$ expressed as:

MATLAB script 3.1

```
V = 1;
xd = 1.8;
xd_prime = 0.3;
xq_prime = 0.55;
J = 4.774;
Td_prime = 8;

J = 4.774;
Td_prime = 8;

elt2 = (V/J) * (sin(gamma)/xd_prime) * eq_prime;
elt3 = ((V^2) / J) * ((1/xq_prime) - (1/xd_prime)) * sin(gamma) * cos(gamma);
row2 = - elt2 - elt3;

row3 = (V/Td_prime) * ((xd/xd_prime) - 1) * cos(gamma);
F = [0; row2; row3];

End
```

3.3.2.1 Simulation Results

Modelled are the system output i.e., the electrical power and rotor angle but also the rotor (angular) speed as in Figures 3.6, 3.7, and 3.8.
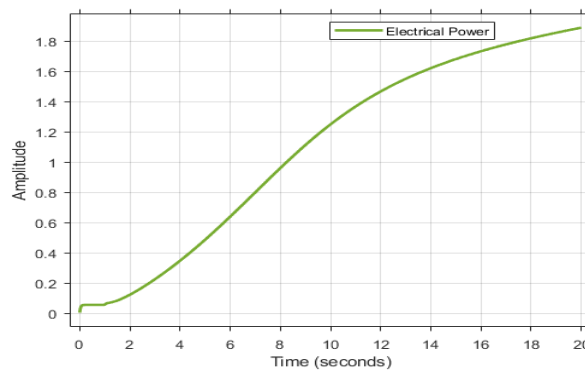
Figure 3.6: Electrical Power output by the generator
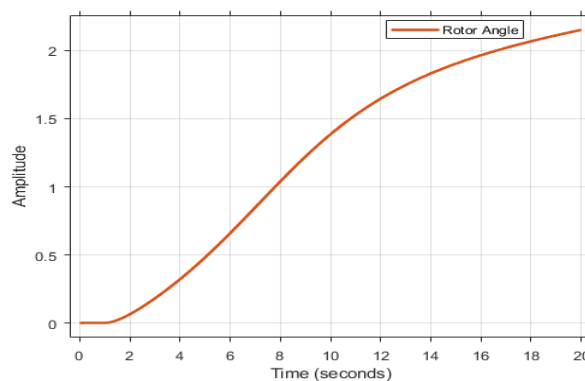
Figure 3.7: Generator Rotor Angle

55
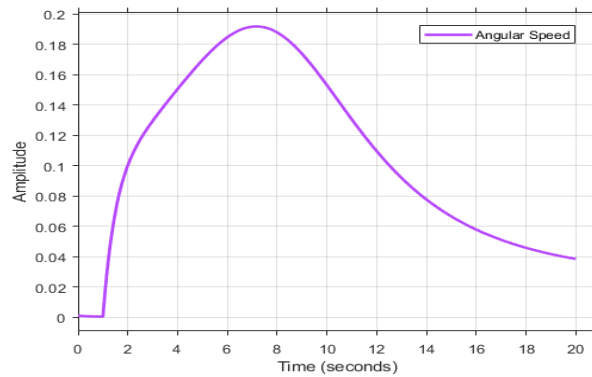
Figure 3.8: Generator Rotor (Angular) Speed

3.3.3. Fourth-Order Synchronous Generator

3.3.3.1 Representation

Parameters used in the modelling are the same as in in Table 3.1.

Figure 3.9 is a representation of Equations (3.25a) and (3.25b).

Similar to the 3rd order representation, the aforementioned Equation (3.25a) can be written in the form of $\dot{X} = AX + BU + F(\delta)$, with $F(\delta)$ expressed as:

MATLAB script 3.2

```
Function F = Fct(gamma, ed_prime,eq_prime)

V = 1;
xd = 1.8;
xq = 1.7;
xd_prime = 0.3;
xq_prime = 0.55;
J = 4.774;
Td_prime = 8;
Tq_prime = 0.4;

elt1 = (1/J) * (V*cos(gamma)/xq_prime) * ed_prime;
elt2 = (V/J) * (sin(gamma)/xd_prime) * eq_prime;
elt3 = ((V^2) / J) * ((1/xq_prime) - (1/xd_prime)) * sin(gamma) *
cos(gamma);
row2 = elt1 - elt2 - elt3;
row3 = (V/Tq_prime) * ((xq-xd_prime) / xq_prime) * sin(gamma);
row4 = (V/Td_prime) * ((xd/xd_prime) - 1) * cos(gamma);
F = [0; row2; row3; row4];

end
```

Figure 3.9: 4th order representation of a Synchronous Generator

3.3.3.1 Simulation Results

The behaviour of the system outputs i.e., the electrical power and rotor angle but also the rotor (angular) speed are shown in Figures 3.10, 3.11, and 3.12 respectively.
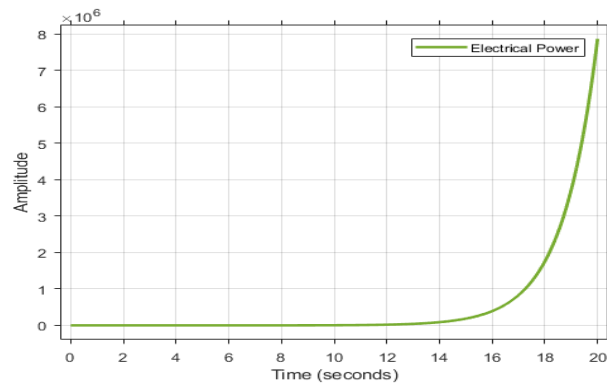
Figure 3.10: Generator Electrical Power
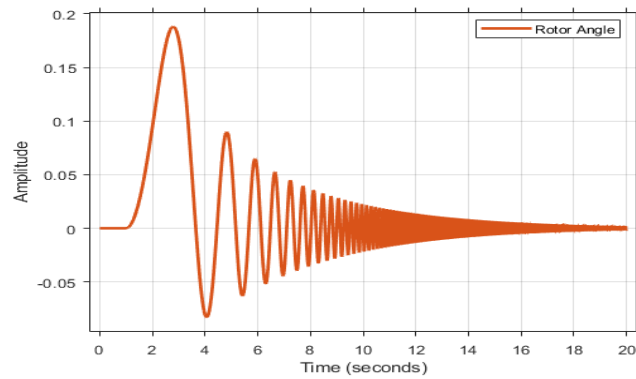
Figure 3.11: Generator Rotor Angle
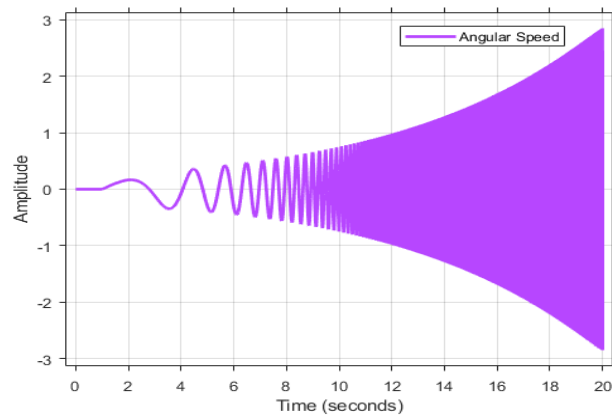
Figure 3.12: Synchronous Generator Rotor (Angular) Speed

3.3.4. Discussions

Though useful for controller design, the third-order representation of a synchronous generator falls short in providing sufficient information pertaining to a power system's oscillations as illustrated in Figures 3.6, 3.7 and 3.8.

As stated by Eremia and Shahidehpour (2013) and Machowski et al. (2008), the 4th order model of a synchronous generator is suitable to model the generator in full range of electromechanical oscillations, and this can be seen from Figures 3.10, 3.11, and 3.12. Inter-area oscillations are inherent to a given power system, and excessive generation as in Figure 3.10 causes acceleration of a given generator's rotor speed (Figure 3.12) which may lead to angular instability and loss of synchronism.

The synchronous generator being a nonlinear system, linear controllers would not suffice for their control. Thus, the need for an appropriate nonlinear controller.

3.4   Conclusion

This chapter introduced the dynamics of the synchronous generator. Both the 3rd and 4th order representations were presented.

Though the latter is an extension of the first, both have revealed through their modelling how unstable a synchronous generator is.

Next, Chapter 4 will present a comprehensive background on nonlinear control theory and introduce both the concept of a Lyapunov based nonlinear controller and Model-Reference Adaptive Controller (MRAC) used in the design of the control algorithm for power system inter-area oscillations damping.

CHAPTER FOUR

DESIGN OF A NONLINEAR REFERENCE MODEL BASED CONTROLLER AND MODEL-REFERENCE ADAPTIVE CONTROLLER (MRAC) FOR POWER SYSTEM LOW FREQUENCY ELECTROMECHANICAL OSCILLATIONS DAMPING

4.1   Introduction

As for any dynamical system with feedback control, stability is an important consideration, and methods such as eigenvalue analysis, root locus, phase margins, etc. are suitable for those classified as Linear Time Invariant (LTI) systems (Nguyen, 2018).

Alexander Mikhailovich Lyapunov, through his work "The general problem of motion" defined two methods of stability namely linearization (quantitative) and direct (qualitative) and introduced the general and integral approaches for the stability of nonlinear systems (Slotine and Li, 1991; Grayson, 1965).

The preponderance of real-world systems are inherently nonlinear and do possess many complex behaviours not observed in linear systems which are viewed as idealization of non-linear systems in some subspaces of their nonlinear solutions (Nguyen, 2018). Characteristics such as multiple equilibrium points, limit cycle, finite escape time, or chaos illustrate their complex behaviour (Nguyen, 2018).

The Stability theory developed by Lyapunov is very important for the design of the nonlinear adaptive controllers (Nguyen, 2018; Bosworth and Williams-Hayes, 2007; Williams-Hayes, 2005; Calise and Rysdyk, 1998; Narendra and Annaswamy, 1987). Therefore, since power systems are highly nonlinear, Lyapunov stability theory would be most suitable as it provides the necessary technique for stability analysis and design of nonlinear controllers for the nonlinear systems.

This chapter is structured as follows:

First, the Lyapunov stability theory is introduced in sections 4.2 and 4.3. Thereafter the synchronous generator dynamics as derived in Chapter 3 are used in the design of a Lyapunov-based nonlinear controller in section 4.4. Lastly, the Model-reference Adaptive Controller (MRAC) for power system oscillations damping is presented in section 4.5.

4.2   Stability Concepts

4.2.1   Background

Due to the difficulty to assess global stability of a nonlinear system over its entire solution domain, linearization in a region about an equilibrium point can provide information on the local stability (Nguyen, 2018).

In contrast to LTI systems that have *the origin* as their sole equilibrium point, a linearized system's origin corresponds to the equilibrium point about which the nonlinear system's linearization is performed (Nguyen, 2018).

4.2.2. Concept of Local Stability

Definition 4.1: *An LTI system is absolutely stable if the eigenvalues of the transition matrix A (Hurwitz matrix) all have negative real part as:*

$$\Re\big(\lambda(A)\big) < 0 \tag{4.1}$$

Definition 4.2: *Phase portraits are plots of trajectories of the solution that can be useful for studying the behaviours of second-order nonlinear systems.*

As for nonlinear systems, phase portraits can also be useful in the study of linearized systems' behaviours from which local stability of nonlinear systems can be learned (Nguyen, 2018).

4.2.3. Highlight of Nonlinear Systems Behaviours

Since they can exhibit many complex behaviours, nonlinear systems do differ greatly to linear systems (Nguyen, 2018; Khalil, 2001).

Unlike LTI systems, a nonlinear system can have multiple equilibrium points, and the phase plane analysis of such a system is similar to that of its linearized version because its local behaviours can be approximated by that of the linearized one in the vicinity of the equilibrium points (Nguyen, 2018).

Other than the multiple isolated equilibrium points, two other prominent behaviours of such system are mentioned by Nguyen (2018):

- *Finite escape time*: The ability of a nonlinear system to become unbounded in a finite interval of time; phenomenon only found in unstable linear system as time approaches infinity.

- **_Limit cycle_**: Described as _a periodic nonlinear solution represented by a closed trajectory in the phase plane such that all trajectories in its vicinity either converge to it or diverge from it_ and depending on the trajectories of the solution in its very vicinity, a limit cycle can either be stable, unstable, or neutral.

4.2.4. Stability Definitions

Let

$$\dot{x} = f(t, x), \quad x(t_0) = x_0 \tag{4.2}$$

where $x \in \mathbb{R}^n, t \geq 0$ is a nonlinear system.

Definition 4.3: _$\dot{x}$ is said to be autonomous or time-invariant if it does not explicitly depend on time, and non-autonomous if otherwise_ (Slotine, 1991; Sastry and Bodson, 1989).

However, it is worth emphasizing that all known physical systems can be classified as non-autonomous in a sense that they have a time-variant dynamic characteristic.

While autonomous systems do not depend on an initial time, non-autonomous ones do depend on it (Slotine, 1991).

Definition 4.4: _The equilibrium point $x = 0$ of a non-autonomous system is said to be stable in the sense of Lyapunov if_

$$\forall t_0 \geq 0 \ and \ \varepsilon > 0, \ \exists \ \delta(t_0, \varepsilon) : \ \|x_0\| < \delta(t_0, \varepsilon) \implies \|x(t)\| < \varepsilon, \ \forall \ t \geq t_0 \tag{4.3}$$

Otherwise, the equilibrium point is said to be unstable (Nguyen, 2018; Slotine, 1991; Sastry and Bodson, 1989). Lyapunov stability implies therefore that, given a system with an initial condition close to the origin, its trajectory can be kept arbitrarily close to it as illustrated in Figure 4.1 (Nguyen, 2018).
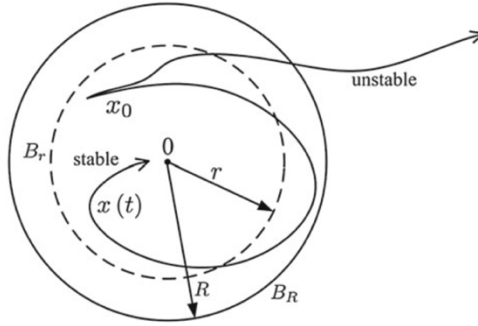
Figure 4.1: Stability concept (Nguyen, 2018)

Unlike linear systems whereby instability implies that the solution grows exponentially as $t \to \infty$ due to unstable poles in the right half plane which result in unbounded signals, nonlinear systems' instability does not always lead to unbounded signals (Nguyen, 2018, Slotine, 1991).

4.2.4.1 Uniform Stability

Definition 4.5: *The equilibrium point $x = 0$ is called a uniformly stable equilibrium point of Equation 4.2 if δ can be chosen to be independent of t_0* (Sastry and Bodson, 1989).

From the above, Stability can be said to be a mild requirement for an equilibrium point in a sense that it does not require that trajectories starting to the origin tend to the origin asymptotically (Sastry and Bodson, 1989).

4.2.4.2 Asymptotic Stability

Such for an ideal spring-mass system without friction that displays a sinusoidal motion forever if subjected to a disturbance, a system can be said to be stable in the Lyapunov sense without converging to the origin (Nguyen, 2018). Hence, as Nguyen (2018) stated, "the stability concept in the Lyapunov sense does not explicitly imply that the trajectory of a nonlinear system will eventually converge to the origin."

Definition 4.6: *The equilibrium point $x = 0$ is called an asymptotically stable equilibrium point of Equation (4.2), if* (Sastry and Bodson, 1989):

- $x = 0$ *is a stable equilibrium point of Equation (4.2),*
- $x = 0$ *is attractive i.e.*

$$\forall t_0 \geq 0 \; , \; \exists \, \delta(t_0) : \|x_0\| < \delta(t_0) \implies \lim_{t \to \infty} \|x(t)\| = 0 \tag{4.4}$$

63

4.2.4.3 Uniform Asymptotic Stability

Definition 4.7: *The equilibrium point $x = 0$ is called a uniformly asymptotically stable equilibrium point of Equation (4.2), if (*Sastry and Bodson, 1989):

- $x = 0$ *is a uniformly stable equilibrium point of Equation (4.2),*

- *the trajectory $x(t)$ converges to 0 uniformly in t_0 i.e.*

$$\exists \ \delta > 0, and \ \gamma(\tau, x_0): \ \mathbb{R}_+ \times \mathbb{R}^n \longrightarrow \mathbb{R}_+$$

$$\lim_{r \to \infty} \gamma(\tau, x_0) = 0 \ \forall \ x_0, \ and \ \|x_0\| < \delta \implies \|x(t)\| \ \leq \gamma(t - t_0, x_0) \ \forall \ t \geq t_0 \geq 0 \qquad (4.5)$$

4.2.4.4 Global Asymptotic Stability

Definition 4.8: *The equilibrium point $x = 0$ is called a globally asymptotically stable equilibrium point of Equation (4.2), if it is asymptotically stable and* $\lim_{t \to \infty} \|x(t)\| = 0 \ \forall \ x_0 \in$ $\mathbb{R}^n$ (Sastry and Bodson, 1989).

It is worth mentioning that the same definition applies to *Global uniform asymptotic stability* and that the speed to convergence is not quantified in all three types of asymptotic stability as here defined (Sastry and Bodson, 1989).

4.2.4.5 Exponential Stability

By comparing the solution of a nonlinear differential equation to an exponential decay function, its rate of convergence can be estimated (Nguyen, 2018; Khalil, 2001; Slotine and Li, 1991).

Definition 4.9: *The equilibrium point $x = 0$ is called an exponentially stable equilibrium point of Equation (4.2), if (*Sastry and Bodson, 1989):

$$\exists \ \alpha, \beta > 0 \colon \|x(t)\| \ \leq \alpha \|x_0\| e^{-\beta(t - t_0)}, \forall \ x_0 \in B_R, t \geq t_0 \geq 0 \qquad (4.6)$$

where B_R is the close ball of radius R centered at 0 in $\mathbb{R}^n$, and β is the rate of convergence.

Exponential stability is assumed to be uniform with respect to t_0; so is assumed uniform asymptotic stability to be equivalent to exponential stability for linear systems (Nguyen, 2018; Sastry and Bodson, 1989).

4.3 Lyapunov Stability Theory
4.3.1 Motivation

Inspired by a physical phenomenon that associates a given mechanical (spring-mass-damper) system equilibrium to the decreasing of its "measure of energy", the stability of a dynamic system can be examined by the variation of a single scalar (energy) function (Slotine and Li, 1991; Sastry and Bodson, 1989).

In contrast with the physical systems, whereby this energy function is unique, Lyapunov function can be any positive-defined function that satisfies the (semi)-definiteness of its time derivative (Nguyen, 2018).

It is similar to the energy concept of a given mechanical system where the following observations can be made:

- the positivity of the energy function,
- the stability of the equilibrium is related to the negative semi-definite nature of the time rate of the energy function.

Alexander Mikhailovich Lyapunov recognized that, using a class of positive-definite (Lyapunov) functions, the stability of a given system can be proven without developing a true knowledge of the system energy (Nguyen, 2018).

Definition 4.10: *A function $\alpha(\epsilon): \mathbb{R}_+ \longrightarrow \mathbb{R}_+$ belongs to a class $\mathcal{K}$ i.e., $\alpha(.) \in \mathcal{K}$, if it is continuous, strictly increasing, and $\alpha(0) = 0$* (Sastry and Bodson, 1989).

Definition 4.11: *A continuous function $v(t, x): \mathbb{R}_+ \times \mathbb{R}^n \longrightarrow \mathbb{R}_+$ is called a locally positive definite function (l.p.d.f) if* (Sastry and Bodson, 1989)*:*

$$\exists \ h > 0, and \ \alpha(.) \in \mathcal{K} :$$
$$v(t,0) = 0 \ and \ v(t,x) \geq \alpha(\|x\|), \forall x \in B_R, t \geq 0 \tag{4.7}$$

Unlike positive definite functions which are globally like *energy functions*, l.p.d.f only resembles them locally (Sastry and Bodson, 1989).

Definition 4.12: *A continuous function $v(t,x)\colon \mathbb{R}_+ \times \mathbb{R}^n \longrightarrow \mathbb{R}_+$ is called a positive definite function (p.d.f), if* (Sastry and Bodson, 1989):

$$\exists\ \alpha(.) \in \mathcal{K}:$$
$$v(t,0) = 0 \ and \ v(t,x) \geq \alpha(\|x\|) \ \forall x \in \mathbb{R}^n, t \geq 0, and \ \lim_{p \to \infty} \alpha(p) = \infty \tag{4.8}$$

It is worth emphasizing that for both the l.p.d.f and p.d.f, the energy functions are not bounded as t varies (Sastry and Bodson, 1989).

Definition 4.13: *The function $v(t,x)$ is called decrescent if* (Sastry and Bodson, 1989):

$$\exists \beta(.) \in \mathcal{K}:$$
$$v(t,x) \leq \beta(\|x\|), \forall x \in B_R, t \geq 0 \tag{4.9}$$

Sastry and Bodson (1989) presented the following examples to illustrate the aforementioned functions:

- $v(t,x) = \|x\|^2$: p.d.f, decrescent
- $v(t,x) = x^T P x$, with $P > 0$: p.d.f, decrescent
- $v(t,x) = (t+1)\|x\|^2$: p.d.f
- $v(t,x) = e^{-t}\|x\|^2$: decrescent
- $v(t,x) = \sin^2(\|x\|^2)$: l.p.d.f, decrescent

Definition 4.14: *A function $V(x)$ is said to be a Lyapunov function if the following conditions are satisfied* (Nguyen, 2018)*:*

- *$V(x)$ is a p.d.f and has a continuous first partial derivative.*
- *$\dot{V}(x)$ is at least negative semi-definite:*

$$\dot{V}(x) = \frac{\partial V}{\partial x}\dot{x} = \frac{\partial V}{\partial x}f(x) \leq 0 \tag{4.10}$$

Figure 4.2 illustrates this using a bowl-shape surface.
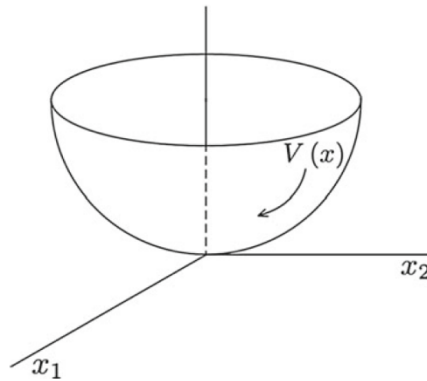
Figure 4.2: Illustration of a Lyapunov function (Nguyen, 2018)

Definition 4.15: *A continuous, positive-valued $\varphi(x) \in \mathbb{R}^+$ is said to belong to class $\mathcal{KR}$; i.e., $\varphi(x) \in \mathcal{KR}$, if (Nguyen, 2018):*

- $\varphi(0) = 0$
- $\varphi(x)$ *is strictly increasing* $\forall \varphi(x) < \infty$
- $\lim\limits_{x \to \infty} \varphi(x) = \infty$

Definition 4.16: *A continuous, positive-valued $V(x) \in \mathbb{R}^+$ with $V(0) = 0$ is said to be radially bounded if (Nguyen, 2018):*

$$\exists \varphi(\|x\|) \in \mathcal{KR} : V(x) \geq \varphi(\|x\|), \forall x(t) \in \mathbb{R}^n \tag{4.11}$$
$$i.e., \ V(x) \to \infty \ as \ \|x\| \to \infty$$

4.3.2. Lyapunov Stability Theorems
4.3.2.1 Lyapunov Theorem for Local Stability
Theorem 4.1: *Let $v(t, x)$ be continuous differentiable, then:*

Table 4.1: Lyapunov Stability for a continuous differentiable function

Conditions on $v(t, x)$	*Conditions on $-\dot{v}(t, x)$*	*Conclusions*
l.p.d.f.	≥ 0 *locally*	*stable*
l.p.d.f., decrescent	≥ 0 *locally*	*Uniformly stable*
l.p.d.f.	*l.p.d.f.*	*Asymptotically stable*
l.p.d.f., decrescent	*l.p.d.f.*	*Uniformly asymptotically stable*

p.d.f., decrescent	p.d.f.	Globally uniformly asymptotically stable

Proof of the theorem can be found in (Vidyasagar, 1978).

An interesting fact about this theorem is that it gives sufficient conditions guaranteeing the stability of Equation (4.2) in a sense that, if an equilibrium point is stable, there exits an l.p.d.f $v(t,x)$ with $\dot{v}(t,x) \leq 0$ (Sastry and Bodson, 1989).

It is worth mentioning that the Lyapunov's direct method only gives a sufficient condition for stability, and a failure of a (Lyapunov) candidate function to satisfy the stability condition does not necessarily imply that the equilibrium is unstable but rather that a good candidate has not been identified (Nguyen, 2018).

4.3.2.2 Lyapunov Theorem for Exponential Stability

Exponential stability is very important, and as stated by Sastry and Bodson (1989), "when considering the convergence of adaptive algorithms, exponential stability means convergence, and the rate of convergence is a useful measure of how fast estimates converge to their nominal values."

Theorem 4.2: *Let $x = 0$ be an equilibrium point and if for a given Lyapunov function $V(x)$:*

$$V(x) > 0, \forall x(t) \in B_R: \dot{V}(x) < 0, \forall x(t) \in B_R,$$

$$\exists \eta, \beta > 0: V(x) \leq \eta \|x\|^2 \ and \ \dot{V}(x) \leq -\beta V(x) \tag{4.12}$$

then the equilibrium is locally exponentially stable (Nguyen, 2018).

The very theorem can be found in (Hahn, 1967), and Sastry and Bodson (1989), with its proof in the later.

4.3.2.3 Barbarshin-Krasovkii Theorem for Global Asymptotic Stability

As stated in Theorem 4.1, the asymptotic stability concept in the Lyapunov sense of an equilibrium point is a local concept and from definition 4.12 where asymptotic stability in the large is said to be a global concept requiring the region of attraction to extend to the entire Euclidian space $\mathbb{R}^n$, the global stability Lyapunov condition is defined following Barbarshin-Krasovskii theorem (Nguyen, 2018; Khalil, 2001; Slotine and Li, 1991):

Theorem 4.3: *The equilibrium point $x = 0$ is said to be asymptotically stable in the large if there exists a radially bounded Lyapunov function $V(x) > 0, \forall x(t) \in \mathbb{R}^n : \dot{V}(x) < 0, \forall x(t) \in \mathbb{R}^n$* (Nguyen, 2018).

4.3.2.4 Lasalle's Invariant Set Theorem

Considering the spring-mass-damper system with the energy function chosen as the Lyapunov function, then $\dot{V}(x) \leq 0$, which conforms to Theorem 4.2. To resolve this apparent contradiction to Theorem 4.3 when an asymptotically stable equilibrium point of an autonomous system only satisfies the condition $\dot{V}(x) \leq 0$, LaSalle introduced the invariant theorem which is defined as (Nguyen, 2018):

Definition 4.17: Given an autonomous system, *a set $\mathcal{M}$ is said to be invariant if every trajectory starting on a point in $\mathcal{M}$ will remain in it for all future time* (Nguyen, 2018; Khalil, 2001; Slotine and Li, 1991), i.e.:

$$x(0) \in \mathcal{M} \implies x(t) \in \mathcal{M}, \forall t \geq t_0 \tag{4.13}$$

The invariant set theorem is stated as:

Theorem 4.4: *Given an autonomous system, let $V(x) > 0$ be a p.d.f with a continuous first partial derivative such that $\dot{V}(x) \leq 0$ in some finite region $B_R \subset \mathcal{D}$. Let $\mathcal{R}$ be a set of all points where $\dot{V}(x) = 0$, and $\mathcal{M}$ the largest invariant set in $\mathcal{R}$. Then, every solution $x(t)$ starting in B_R approaches $\mathcal{M}$ as $t \longrightarrow \infty$*

Considering the spring-mass-damper system whose invariant set $\mathcal{M} \subset \mathcal{R}$ is a set that contains only the origin (Nguyen, 2018: 66), according to Theorem 4.4 all trajectories will converge to the origin as $t \longrightarrow \infty$; making the origin asymptotically stable.

From the above, the following corollary of Theorem 4.4 can be stated (Nguyen, 2018):

Corollary 4.1: *Let $V(x) > 0$ be a p.d.f with a continuous first partial derivative such that $\dot{V}(x) \leq 0$ in some finite region $B_R \subset \mathcal{D}$ and $\mathcal{R} = \{x(t) \in B_R : \dot{V}(x) = 0\}$. Assuming that no solution other than $x = 0$ can stay in $\mathcal{R}$, the origin is then said to be asymptotically stable.*

Furthermore, if $V(x) > 0$ is a positive definite radially unbounded function and $\mathscr{R} = \{x(t) \in \mathbb{R}^n : \dot{V}(x) = 0\}$, the origin is then said to be asymptotically stable at large.

4.3.2.5 Barbalat's Lemma

In contrast to LaSalle's invariant theorem that deals with the asymptotic stability of autonomous system, Barbalat's lemma addresses the challenges of assessing asymptotic stability of non-autonomous systems to some extent (Nguyen, 2018; Slotine and Li, 1991).

Definition 4.18: A function $f(t) \in \mathbb{R}$ is said to be uniformly continuous on a set $\mathcal{D}$ if, $\forall \varepsilon > 0, \exists \delta(\varepsilon) > 0$:

$$|t_2 - t_1| < \delta \implies |f(t_2) - f(t_1)| < \varepsilon, \forall t_1, t_2 \tag{4.14}$$

Extending definition 4.18, the uniform continuity of a differentiable function $f(t)$ can be said to require its derivative $\dot{f}(t)$ to exist and be bounded (Nguyen, 2018).

Lemma 4.1: *if $f(t)$ is a uniformly continuous function, such that $\lim_{t \to \infty} \int_0^t f(\tau) d\tau$ exists and is finite, then $f(t) \longrightarrow 0$ as $t \longrightarrow \infty$*

Proof of the above lemma can be found in Popov (1973).

The above does not always hold true as highlighted in (Nguyen, 2018: 73), with functions such as $f(t) = \sin(\ln t)$ whose derivative $\dot{f}(t) = \frac{1}{t}\cos(\ln t)$ tends to zero but $f(t)$ does not have a finite limit as $t \longrightarrow \infty$.

Hence, an extension of the Barbalat's lemma to the Lyapunov's direct method to examine asymptotic stability of non-autonomous system by the following Lyapunov-like lemma (Nguyen, 2018; Slotine and Li, 1991):

Lemma 4.2: *Considering a p.d.f $V(x, t)$ with a finite limit as $t \longrightarrow \infty$, and if $\dot{V}(x, t)$ is negative semi-definite and uniformly continuous $\forall t \in [0, \infty)$, then $V(x, t) \longrightarrow 0$ as $t \longrightarrow \infty$.*

As stated by Nguyen (2018), "The Lyapunov stability theory is the foundation of nonlinear systems and adaptive control theory," and based on the aforementioned theorems and lemmas, the following can be said:

i.   Barbashin-Krasovskii theorem provides a method for global stability analysis.

ii. LaSalle's invariant set theorem provides a complementary tool for analysing systems with invariant sets.

iii. Assessing the stability of non-autonomous systems involves the concept of uniform stability, uniform boundness, and uniform ultimate boundness.

iv. The stability of adaptive control systems in connection with the uniform continuity of a real-valued function can be analysed better through Barbalat's Theorem.

4.4   Design of a Nonlinear Reference Model Based Controller for Power System Oscillations Damping

4.4.1   Overview

As stated in Chapter 3, two representations of the synchronous generator are utilized in the design of the control algorithm i.e., the 3rd and 4th orders.

In this section, a nonlinear servo-based reference model controller is presented.

After a brief review of the generator's dynamics, theories around the design of the reference model are introduced. Thereafter, the actual structure of the nonlinear servo-based reference model controller is presented.

4.4.2   Third-order Synchronous Generator Dynamics

From Equations 3.20a and 3.20b the state and output of the synchronous generator are represented as:

$$
\underline{\dot{x}} =
\begin{bmatrix}
0 & 1 & 0 \\
0 & -\dfrac{D}{J} & 0 \\
0 & 0 & \dfrac{1}{T'_{do}}\left(\dfrac{x_d}{x'_d}\right)
\end{bmatrix}
\begin{bmatrix}
\delta \\
\omega \\
e'_q
\end{bmatrix}
+
\begin{bmatrix}
0 & 0 \\
0 & \dfrac{1}{J} \\
\dfrac{1}{T'_{do}} & 0
\end{bmatrix}
\begin{bmatrix}
E_{FD} \\
P_m
\end{bmatrix}
$$

$$
+
\begin{bmatrix}
0 \\
-\dfrac{1}{J}\left(\dfrac{V}{x'_d}e'_q \sin(\delta) + \dfrac{V^2}{2}\left(\dfrac{1}{x_q} - \dfrac{1}{x'_d}\right)\sin(2\delta)\right) \\
\dfrac{1}{T'_{do}}\left(\dfrac{x_d - x'_d}{x'_d}\right)Vcos(\delta)
\end{bmatrix}
$$

$$
\underline{y} = \begin{bmatrix} \delta \\ P_e \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ \dfrac{V^2}{2\delta}\left(\dfrac{1}{x'_q} - \dfrac{1}{x'_d}\right)\sin(2\delta) & 0 & \dfrac{V}{x'_d}\sin(\delta) \end{bmatrix} \begin{bmatrix} \delta \\ \omega \\ e'_q \end{bmatrix}
$$

where:

δ : rotor angle

ω : rotor speed

V : voltage on transformer's terminal

E_{FD} : induced emf by the field current (field voltage)

T'_{do} : d-axis open-circuit time constant

e'_q : q-axis transient emf

D : damping coefficient

P_e : air-gap power of the generator

x_d, x_q, and x'_d : augmented reactances with the line and transformer reactances added onto them (Leon-Morales et al., 2001; Sanchez-Orta et al., 2002)

The above equations can be written in the form of

$$\dot{\underline{x}} = A\underline{x} + B\underline{u} + \underline{F}(\delta) \tag{4.15a}$$

$$\underline{y} = C\underline{x} \tag{4.15b}$$

where $\underline{F}(\delta)$ is a vector of the nonlinear elements.

4.4.3 Reference Model-Based Closed Loop Control System

When used as the target to ensure that a given system's output is consistent with the same target, the reference model is said to determine the performance of the very given system (Su et al., 2019).

The design of the generator's controller is based on the above concept.

The generator closed-loop system's output behaviour must follow that of a reference model's output. Hence, the design of its controller has to be done in such a way that it produces a signal that makes the system's output mimics that of the reference model as illustrated in Figure 4.3.
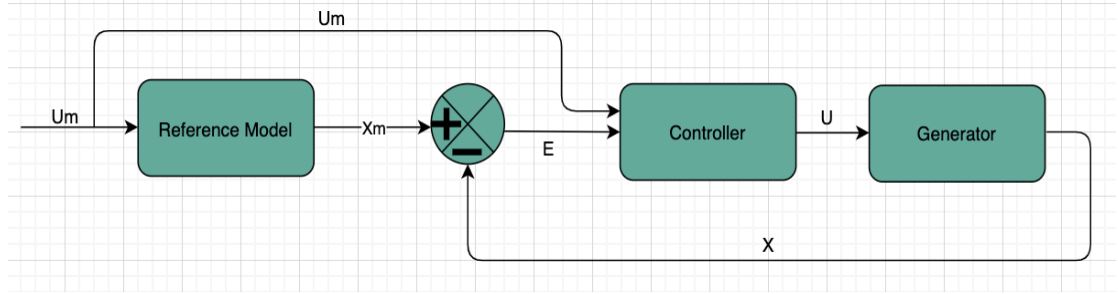
Figure 4.3: Proposed control architecture overview

where $u_m \in \mathbb{R}^m$ is the reference model's control vector input, $u \in \mathbb{R}^m$ the controller's output, and $e \in \mathbb{R}^n$ the error between the desired and the generator's state vectors.

Assumed to be linear in this thesis, the reference model is expressed by the following equations:

$$\dot{\underline{x}}_m = A_m \underline{x}_m + B_m \underline{u}_m \qquad (4.16a)$$

$$\underline{y}_m = C\underline{x}_m \qquad (4.16b)$$

where:

$\underline{x}_m \in \mathbb{R}^n$ is the reference model's state vector.

$\underline{u}_m \in \mathbb{R}^m$ is the reference model's control vector.

$A_m \in \mathbb{R}^{nxn}$ is the reference's model state matrix.

$B_m \in \mathbb{R}^{nxm}$ is the reference's model control matrix.

while A_m is assumed to be a $Hurwitz\ matrix$ so that the reference model system has an asymptotically stable equilibrium state, the control signal u_m is selected such that $\underline{x}_m$ follows a desired trajectory, and based on Figure 4.3, the generator's state will thereafter follow the same behaviour.

Equation (4.3) determines that the reference model is a linear system. Therefore, ensuring its stability mean applying linear control tools.

Chosen to be linear, the reference model requires linear control techniques to ensure its stability. Without having any integrator in its structure and to meet the overall robustness needed, the type 1 servo for type-0 plant regulator structure is selected for the computation of the control signal u_m

4.4.4   PI Controller Design for the Reference Model Closed Loop System

Specifying a given system's performance does often imply utilizing a model that produces the desired output for a given input signal. This is achieved by comparing the desired output with that of the system to control to adjust the control signal. In some cases, however, it is the error dynamics between the very system and the reference model that is used.

The reference model is chosen such that it matches some predefined design specifications as well as the structure of the system matrix. Furthermore, Linear Time Invariant (LTI) are often considered as good candidates.

Considering that the reference model dictates the behaviour of our system's output, classical control techniques such as pole placement, frequency response, …. can be used in ensuring it produces the desired output.

For this thesis, the pole-placement method is used via the servo system concept as in Figure 4.4.
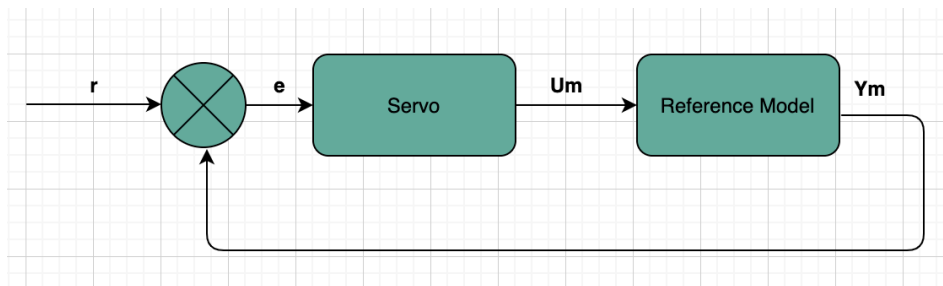
Figure 4.4: Linear controller for reference model

The concept of *servomechanism* was introduced by Hazen, and as early as in 1934 after defining the above concept, he discussed the design of a relay servomechanism that could closely follow a changing input (Ogata, 2010).

As explained in Ogata (2010), with modern control theory[2], the *pole-placement* method is the preferred method for stability of closed-loop linear system. From its very name, the method referred to as $pole\text{-}placement$ requires that all closed-loops poles be placed at desired locations to ensure the stability of a given (linear) system.

Stability using the servo system concept can be achieved in two possible ways depending on the given plant's structure, with $type\text{-}1$ plants referring to plants containing an integrator and $type\text{-}0$ plants, those without one.

Throughout the rest of the thesis, $type\ 1\ servo\ system$ would refer to a linear system whereby the servo system concept is utilized for its stability through pole-placement method for the design of the controller.
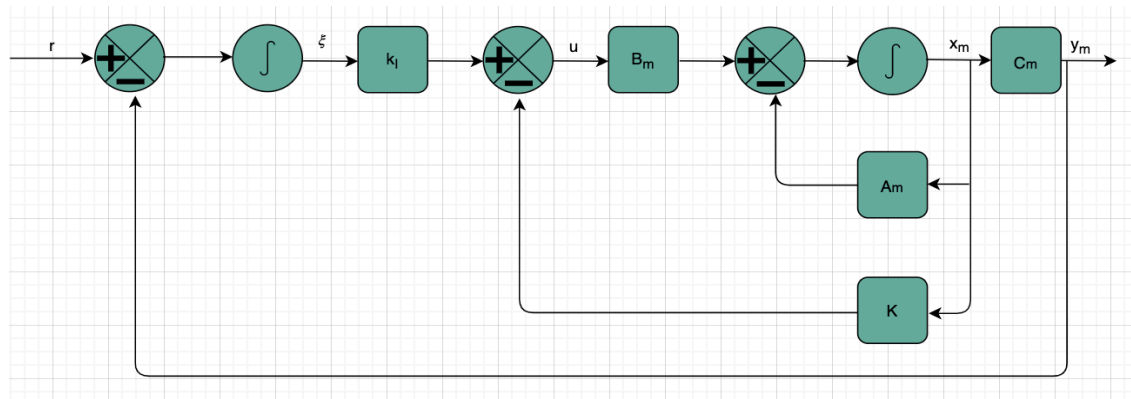
Figure 4.5: Type 1 servo for type-0 plant

Two assumptions are critical here:

- The system transfer function in Equation (4.17) has no zero at the origin:

$$G(S) = C_m(SI - A_m)^{-1}B_m \tag{4.17}$$

- The reference model given by Equation (4.18) needs to be completely state controllable:

$$\dot{x}_m = A_m x_m + B_m u_m \tag{4.18}$$

---

[2] Modern control theory is based on the description of system equations in terms of n first-order differential equations, which may be combined into a first-order vector-matrix differential equation (Ogata, 2010).

From Figure 4.5, it can be written the following systems of equations, as follows:

$$y_m = C_m x_m \tag{4.19}$$

$$u_m = -Kx_m + k_1\xi \tag{4.20}$$

$$\dot{\xi} = r - y_m = r - C_m x_m \tag{4.21}$$

where:

u_m is the reference model control signal,

y_m, the system output

ξ, the integrator output (system's state variable)

r , the reference input (step function)

$A_m = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -18 & -15 & -2 \end{bmatrix}$, 3× 3 constant matrix

$B_m = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$, 3× 1 constant matrix

$C_m = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$, 1 × 3 constant matrix

The closed loop system dynamics can be described by a combination of Equations (4.18) – (4.21) (Ogata, 2010):

$$\begin{bmatrix} \dot{x}(t) \\ \dot{\xi}(t) \end{bmatrix} = \begin{bmatrix} A_m & 0 \\ -C_m & 0 \end{bmatrix} \begin{bmatrix} x_m(t) \\ \xi(t) \end{bmatrix} + \begin{bmatrix} B_m \\ 0 \end{bmatrix} u_m(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r(t) \tag{4.22}$$

An asymptotically stable system with $x_m(\infty)$, $\xi(\infty)$, and $u_m(\infty)$ approaching constant values respectively such that at steady state, $\dot{\xi}(t) = 0$ and $y_m(\infty) = r$ is designed (Ogata, 2010):

$$\begin{bmatrix} \dot{x}(\infty) \\ \dot{\xi}(\infty) \end{bmatrix} = \begin{bmatrix} A_m & 0 \\ -C_m & 0 \end{bmatrix} \begin{bmatrix} x_m(\infty) \\ \xi(\infty) \end{bmatrix} + \begin{bmatrix} B_m \\ 0 \end{bmatrix} u_m(\infty) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r(\infty) \tag{4.23}$$

Since $r(\infty) = r$, for $t > 0$, subtracting Equation (4.23) from Equation (4.22), it is obtained:

$$\begin{bmatrix} \dot{x}_m(t) - \dot{x}_m(\infty) \\ \dot{\xi}(t) - \dot{\xi}(\infty) \end{bmatrix} = \begin{bmatrix} A_m & 0 \\ -C_m & 0 \end{bmatrix} \begin{bmatrix} x_m(t) - x_m(\infty) \\ \xi(t) - \xi(\infty) \end{bmatrix} + \begin{bmatrix} B_m \\ 0 \end{bmatrix} [u_m(t) - u_m(\infty)] \tag{4.24}$$

with

$$x_e(t) = x_m(t) - x_m(\infty)$$

$$\xi_e(t) = \xi(t) - \xi(\infty)$$

$$u_e(t) = u_m(t) - u_m(\infty) = -Kx_e(t) + k_1\xi_e(t)$$

Equation (4.25) can be rewritten as:

$$\begin{bmatrix} \dot{\boldsymbol{x}}_e(t) \\ \dot{\boldsymbol{\xi}}_e(t) \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_m & 0 \\ -\boldsymbol{C}_m & 0 \end{bmatrix} \begin{bmatrix} x_e(t) \\ \xi_e(t) \end{bmatrix} + \begin{bmatrix} \boldsymbol{B}_m \\ 0 \end{bmatrix} \boldsymbol{u}_e(t) \tag{4.25}$$

The above equation can further be simplified by defining a new $(n+1)th - order$ error vector $e(t)$ to produce:

$$e(t) = \begin{bmatrix} \dot{\boldsymbol{x}}_e(t) \\ \dot{\boldsymbol{\xi}}_e(t) \end{bmatrix} \implies \dot{e}(t) = \widehat{A}e + \widehat{B}u_e \tag{4.26a}$$

$$\therefore \dot{e}(t) = (\widehat{A} - \widehat{B}\widehat{K})e \tag{4.26b}$$

where:

$$\widehat{A} = \begin{bmatrix} A_m & 0 \\ -C_m & 0 \end{bmatrix},$$

$$\widehat{B} = \begin{bmatrix} B_m \\ 0 \end{bmatrix},$$

$$u_e = -\widehat{K}e,$$

$$\widehat{K} = [K \vdots - k_1],$$

While the desired closed-loop poles are obtained by finding the eigenvalues of $\widehat{A} - \widehat{B}\widehat{K}$, pole-placement techniques are utilized for the computation of the state-feedback matrix K and the integral gain constant k_1.

Furthermore, the controllability of the system described in Equation (4.26b) can be asserted provided that $\begin{bmatrix} A_m & 0 \\ -C_m & 0 \end{bmatrix}$ has a rank $(n+1)$.

4.4.5 Determination of the Error Dynamics (E) Between the States of the Reference Model and the Generator

From Equations (4.15) and (4.16), the state error between the reference model and the generator model can be written as:

$$\underline{e} = \underline{x}_m - \underline{x}, \underline{e} \in \mathbb{R}^n \tag{4.27}$$

The state error must be minimized by a suitable control vector u.

Differentiating Equation (4.27), we have:

$$\underline{\dot{e}} = \underline{\dot{x}}_m - \underline{\dot{x}}$$

$$\Leftrightarrow \underline{\dot{e}} = \underline{A}_m \underline{x}_m + \underline{B}_m \underline{u}_m - \underline{h}(x, u, \delta)$$

$$\Leftrightarrow \underline{\dot{e}} = A_m \underline{x}_m + A_m \underline{x} - A_m \underline{x} + B_m \underline{u}_m - \underline{h}(x, u, \delta)$$

$$\Leftrightarrow \underline{\dot{e}} = A_m(\underline{x}_m - \underline{x}) + A_m \underline{x} + B_m \underline{u}_m - \underline{h}(x, u, \delta)$$

where $\underline{h}(x, u, \delta)$ represents the generator dynamics i.e., $Ax + Bu + F(\delta)$.

Finally, the expression of the derivative of the error signal can be presented as in Equation (4.28):

$$\underline{\dot{e}} = A_m \underline{e} + A_m \underline{x} + B_m \underline{u}_m - \underline{h}(X, U, \delta) \tag{4.28}$$

The nonlinear controller design problem is therefore that of ensuring that $\underline{x}_m = \underline{x}$ to make the error $\underline{e}$ zero.

4.4.6 Design of the Nonlinear Controller
4.4.6.1 Overview

This section covers the method used for the design of the nonlinear linearizing controller based on a reference model and the Lyapunov stability (direct method) theory together with a linear controller aimed at improving the performance of the closed-loop system. While the linear controller described in section 4.4.4 is used for the stability of the reference model, the nonlinear controller is meant to ensuring that the Single Machine Infinite Bus (SMIB) system remains stable when subjected to disturbances.

Figure 4.6 shows the steps used in the design of the nonlinear controller.
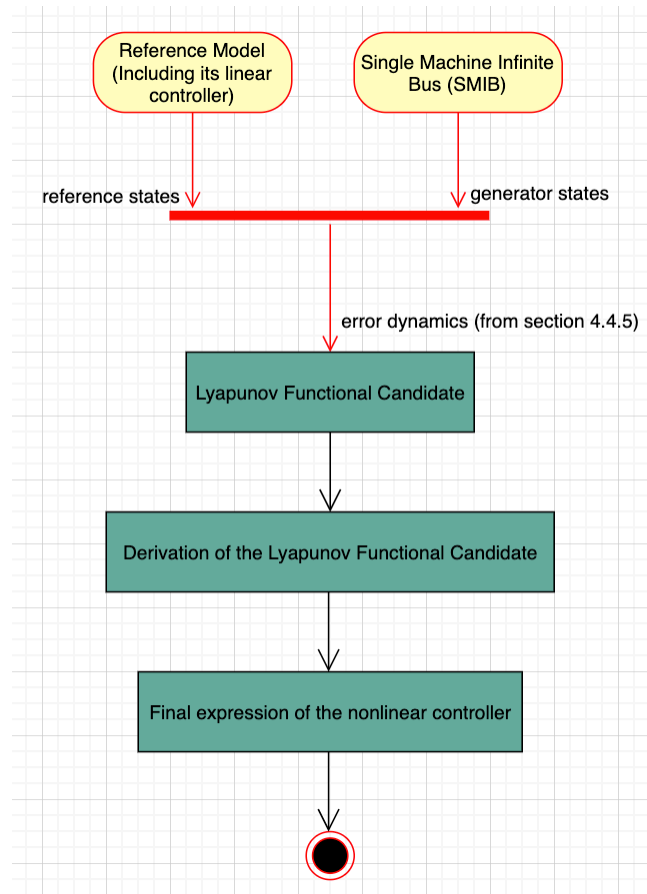
Figure 4.6: Steps describing the process used in the derivation of the control signal for the 3rd order model of the synchronous generator.

Details around the reference model and its control together with the expression of the error dynamics being presented in section 4.4.4 and 4.4.5, the choice of the Lyapunov functional candidate, its derivation and final structure of the nonlinear controller will be presented in the following subsections.

4.4.6.1 Lyapunov Functional Candidate for the Generator Controller

Let V, a p.d.f and decrescent function, be used as the Lyapunov candidate:

$$V(t) = \underline{E}^T P \underline{E} \tag{4.29}$$

where $P \in \mathbb{R}^{n \times n}$ is a positive definite real symmetrical matrix.

4.4.6.2 Derivation of the Lyapunov Function

The first derivative of the Lyapunov function described in Equation 4.29 is:

$$\dot{V}(t) = \dot{E}^T PE + E^T P\dot{E} \tag{4.30}$$

After substitution of the expression from Equation (4.28), the following two expressions for the Lyapunov derivative are obtained

$$\Leftrightarrow \dot{V}(t) = (A_m E + A_m X + B_m U_m - AX - BU - F(\delta))^T PE$$
$$+ E^T P(A_m E + A_m X + B_m U_m - AX - BU - F(\delta)) \tag{4.31}$$

$$\Leftrightarrow \dot{V}(t) = E^T A_m{}^T PE + X^T A_m{}^T PE + U_m{}^T B_m{}^T PE - X^T A^T PE - U^T B^T PE - F(\delta)^T PE$$
$$+ E^T P A_m E + E^T P A_m X + E^T P B_m U_m - E^T P A X - E^T P B U$$
$$- E^T P F(\delta) \tag{4.32}$$

Considering $E^T A_m{}^T PE + E^T P A_m E = E^T (A_m{}^T P + P A_m) E$, then the following expression is obtained:

$$\dot{V}(t) = E^T P (A_m{}^T P + P A_m) E + 2\Gamma \tag{4.32}$$

where:

$$2\Gamma = (X^T A_m{}^T PE + U_m{}^T B_m{}^T PE - X^T A^T PE - U^T B^T PE - F(\delta)^T PE) + (E^T P A_m X$$
$$+ E^T P B_m U_m - E^T P A X - E^T P B U - E^T P F(\delta)) \tag{4.33}$$

In order to have $E^T P$ at the beginning of the expression (4.33), the expression $(X^T A_m{}^T PE + U_m{}^T B_m{}^T PE - X^T A^T PE - U^T B^T PE - F(\delta)^T PE)$ is transposed. This results in the following:

$$2\Gamma = (E^T P A_m X + E^T P B_m U_m - E^T P A X - E^T P B U - E^T P F(\delta)) + (E^T P A_m X$$
$$+ E^T P B_m U_m - E^T P A X - E^T P B U - E^T P F(\delta)) \tag{4.34}$$

$$\implies 2\Gamma = 2E^T P (A_m X + B_m U_m - AX - BU - F(\delta)) \tag{4.35}$$

Finally,

$$\Gamma = E^T P (A_m X + B_m U_m - AX - BU - F(\delta)) \tag{4.36}$$

4.4.6.3 Determination of the Nonlinear Controller Expression

Based on Barbalat's lemma, $\dot{V} < 0$ and from Equation (4.32) it can be written:

$$E^T P \left(A_m{}^T P + P A_m \right) E + 2\Gamma < 0 \tag{4.37}$$

Considering that the reference model is selected to be stable i.e., $E^T \left(A_m{}^T P + P A_m \right) E < 0$, then $\Gamma < 0$ would ensure that the generator is asymptotically stable in large.

To make the generator's output follow the desired behaviour of the reference model, the control signal U must be computed in such a way that $\Gamma < 0$ i.e.:

$$E^T P \left(A_m X + B_m U_m - AX - BU - F(\delta) \right) < 0$$
$$\iff E^T P (A_m X + B_m U_m - AX - F(\delta)) < E^T PBU \tag{4.38}$$

To derive the expression of U, $E^T PB$ needs to be made an identity matrix. Since $E^T PB \in \mathbb{R}^{1x2}$, a transformation is needed to make it quadratic for further processing. This is achieved by multiplying Equation (4.38) with the transpose of $E^T PB$:

$$(E^T PB)^T E^T P (A_m X + B_m U_m - AX - F(\delta)) = (E^T PB)^T (E^T PB) U \tag{4.39}$$

The matrix $(E^T P)^T E^T P \in \mathbb{R}^{2x2}$, multiplying it by its inverse would produce an identity matrix. This is shown through the derivations below.

$$[(E^T PB)^T E^T PB]^{-1} (E^T PB)^T E^T P (A_m X + B_m U_m - AX - F(\delta))$$
$$< [(E^T PB)^T E^T PB]^{-1} (E^T PB)^T (E^T PB) U \tag{4.40}$$

Finally,

$$U > [(E^T PB)^T E^T PB]^{-1} (E^T PB)^T E^T P (A_m X + B_m U_m - AX - F(\delta)) \tag{4.41}$$

Figure 4.7 which is an extension of Figure 4.3 illustrates the overall structure of the proposed control scheme. Note, the aspect of data exchange to and from the synchronous generator is covered in Chapter 6.
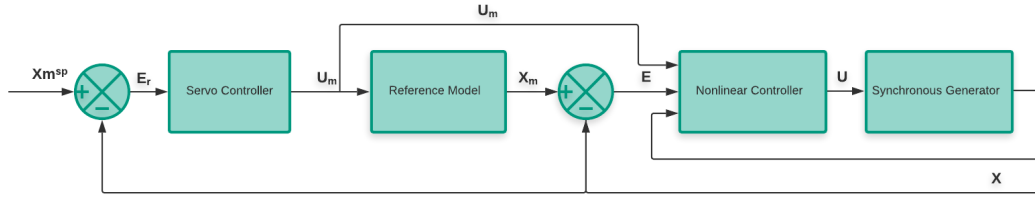
Figure 4.7: Overall structure of the proposed power system interarea oscillations damping control scheme considering the 3rd order representation of the synchronous generator.

4.5 Design of a Model-Reference Based Adaptive Controller for the Synchronous Generator

4.5.1 Overview

In this section, a Linear Quadratic Regulator (LQR)-based reference model adaptive controller is presented.

In contrast with the previous section whereby the 3rd order representation was utilized, the 4th order is used.

Unlike the 3rd order model where a servo could be used as the reference model's controller, with the design of the adaptive controller, no feedback nor feedforward gains could be obtained such that:

$$A - BK_x = A_m \tag{4.42a}$$

$$BK_r = B_m \tag{4.42b}$$

where:

A_m is the reference model system matrix

B_m is the reference input matrix

K_x is the feedback gain matrix

K_r is the feedforward gain matrix

That is due to the structure of the linearized 4th order model's system matrix that has uncontrollable subspaces. Controlling the reference model implies therefore the use of a different type of controller. Thus, the use of LQR for its choice and control.

After a brief review of the generator's dynamics, theories around the design of the optimal regulator systems are introduced. Thereafter, the actual structure of adaptive controller is presented.

4.5.2 Fourth-order Synchronous Generator Dynamics

From Equation 3.22a the state of the synchronous generator is represented as:

$$\underline{\dot{x}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \dfrac{-D}{J} & 0 & 0 \\ 0 & 0 & -\dfrac{1}{x_q'\,T_{qo}'}\left(x_q' + x_q - x_d'\right) & 0 \\ 0 & 0 & 0 & -\dfrac{1}{T_{do}'}\left(\dfrac{x_d}{x_d'}\right) \end{bmatrix} \begin{bmatrix} \delta \\ \omega \\ e_d' \\ e_q' \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & -\dfrac{1}{J} \\ 0 & 0 \\ \dfrac{1}{T_{do}'} & 0 \end{bmatrix} \begin{bmatrix} E_{FD} \\ P_m \end{bmatrix}$$

$$+ \begin{bmatrix} 0 \\ \dfrac{1}{J}\left(\dfrac{e_d'\,V}{x_q'}\cos\delta - \dfrac{e_q'\,V}{x_d'}\sin\delta - \dfrac{V^2}{2}\sin 2\delta\left(\dfrac{1}{x_q'} - \dfrac{1}{x_d'}\right)\right) \\ \dfrac{V}{T_{qo}'}\left(\dfrac{x_q - x_d'}{x_q'}\right)\sin\delta \\ \dfrac{V}{T_{do}'}\left(\dfrac{x_d}{x_d'} - 1\right)\cos\delta \end{bmatrix}$$

where:

δ : rotor angle

ω : rotor speed

i_d : current in the d-axis

i_q : current in the q-axis

V : voltage on transformer's terminal

E_{FD} : induced emf by the field current (field voltage)

T_{do}' : d-axis open-circuit time constant

$\dot{e}_q'$: q-axis transient emf

$\dot{e}_d'$: d-axis transient emf

D : damping coefficient

P_e = air-gap power of the generator

The above equations can be written in the form of:

$$\dot{\underline{x}} = A\underline{x} + B\underline{u} + \underline{F}(\delta) \tag{4.43}$$

where $\underline{F}(\delta)$ is a vector of the nonlinear elements also referred to as the $unmatched$ $uncertainty$. This concept will be further developed in section 4.5.4.

Furthermore, Equation (3.22b), the output has the following expression:

$$Y = \begin{bmatrix} \delta \\ P_e \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \dfrac{V^2}{2\delta}\left(\dfrac{1}{x'_q} - \dfrac{1}{x'_d}\right)\sin(2\delta) & 0 & \dfrac{-V\cos\delta}{x'_q} & \dfrac{V\sin\delta}{x'_d} \end{bmatrix} \begin{bmatrix} \delta \\ \omega \\ e'_d \\ e'_q \end{bmatrix}$$

4.5.3   Quadratic Optimal Regulator Systems Design
4.5.3.1 Optimal Control Theory Review
4.5.3.1.1 Overview

As stated by Burns (2001), "An optimal control system seeks to maximize the return from a system for the minimum cost." Hence finding a suitable control signal that makes a given system to follow an optimal trajectory while minimizing the performance criterion makes the optimal control problem.

This problem is said to be one of the constrained functional minimizations (Burns, 2001), and various methods can be used in solving it. They are described in Table 4.2.

Table 4.2: Methods employed in solving the optimal control problem

Author(s)	Method	Objective/Application
Dreyfus (1962)	Variational calculus to obtain the Euler-Lagrange equations	- Provide boundary conditions - Applicable to systems with nonlinear, time-varying state equations as well as non-quadratic, time-varying performance criteria
Pontryagin (1962)	Maximum Principle	- Provide boundary conditions via a Hamiltonian function - Applicable to systems with nonlinear, time-varying state equations as well as non-quadratic, time-varying performance criteria
Bellman (1957)	Dynamic programming method based on the principle of optimality and the imbedding approach.	- Optimal control policy - Linear Time Invariant (LTI) plants with performance index which takes the form of the matrix Ricatti equation

Among advantages of having quadratic optimal control over pole-placement methods is the fact that they provide a systematic way of computing the state feedback control gain (Ogata, 2010).

4.5.3.1.2 Types of Optimal Control

From the aforementioned methods, various types of optimal control problems are presented (Burns, 2001):

a. *Terminal control problem*:

Objective: Bring a given system as close as possible to the terminal state within a predefined period of time.

Application: Automatic aircraft landing system with the optimum control policy focusing on minimizing errors in the state vector at landing point.

b. *Minimum-time control problem*:

Objective: Reach the terminal state within the shortest time period.

Application: "Bang-Bang" control policy where the control is set to its maximum initially and switched to it minimum at specific times.

c. *Minimum energy control problem*:

Objective: With minimum expenditure of control energy, transfer a given system from an initial to its final state.

Application: Satellite control.

d. *Regulator control problem*:

Objective: Return a given system to its equilibrium state after it has been displaced. This is achieved through the minimization of the performance index.

e. *Tracking control problem*:

Objective: This is a generalization of the regulator control problem whereby the system states are to track some desired state time history while minimizing the performance index.

4.5.3.2 Linear Quadratic Regulator

4.5.3.2.1 Introduction

Linear Quadratic Regulator is in fact a regulator control problem; and as stated in the previous section, it is used to provide an optimal control law for a linear system by minimizing the quadratic performance index. The selection of the latter depending upon the nature of control problem.

4.5.3.2.2 Structure

Given a system $\dot{x} = Ax + Bu$, the quadratic optimal regulator problem implies finding the matrix **K** of the optimal control vector $u(t) = -Kx(t)$ to minimize the performance index. This optimal configuration is illustrated in Figure 4.8.
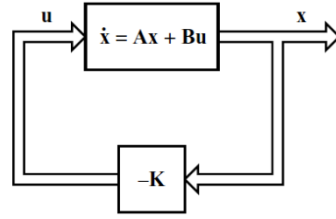
Figure 4.8: Optimal regulator (Ogata, 2010)

From Figure 4.8, let $\dot{x}$ be a given system equation, and the optimal control vector $u(t)$, the performance index can be expressed as (Ogata, 2010):

$$J = \int_0^\infty (x^T Q x + u^T R u) dt \tag{4.44}$$

where:

Q: can either be a positive-definite Hermitian, a positive semi-definite Hermitian or real symmetric matrix.

R: positive-definite Hermitian or real symmetric matrix.

From Equation (4.44), matrices Q and R, which are also referred to as state-cost weighted matrix and control weighted matrix respectively, determine the relative importance of the error and the expenditure of the energy of the control signals which is represented by $u^T R u$ therein. Furthermore, it is assumed that the control vector is unconstrained.

Rewriting Equation (4.44) to include the expression of the control signal produces:

$$J = \int_0^\infty x^T (Q + K^T R K) x \, dt \tag{4.45}$$

Let **P** be a positive-definite Hermitian or real symmetric matrix such that:

$$x^T (Q + K^T R K) x = -\frac{d}{dt}(x^T P x) \tag{4.46}$$

Yet,

$$\frac{d}{dt}(x^T P x) = \dot{x}^T P x + x^T P \dot{x}$$

Equation (4.46) can now be expressed as:

$$x^T(Q + K^T R K)x = -\dot{x}^T P x - x^T P \dot{x} \tag{4.47}$$

Given the system $\dot{x} = Ax + Bu$, and the control vector $u(t) = -Kx(t)$, Equation (4.47) becomes:

$$x^T(Q + K^T R K)x = -x^T[(A - BK)P + P(A - BK)]x \tag{4.48}$$

For the expression in Equation (4.48) to hold true,

$$[(A - BK)P + P(A - BK)] = -(Q + K^T R K) \tag{4.49}$$

$(A - BK)$ being stable, the equality in Equation (4.49) holds if there exist a positive definite matrix P that satisfy the very equation.

From the above, the integral in Equation (4.45) can be solved as:

$$J = \int_0^\infty x^T(Q + K^T R K)x \, dt = -x^T P x \Big|_0^\infty = -(x^T(\infty)Px(\infty) - x^T(0)Px(0) \tag{4.50}$$

Considering that all eigenvalues of $(A - BK)$ have negative real parts, $x(\infty) = 0$, the expression of the performance index is therefore dependent on both the initial condition $x(0)$ and P, and can thus be expressed as:

$$J = x^T(0)Px(0) \tag{4.51}$$

R being assumed to be a positive-definite Hermitian or real symmetric matrix, the following equality holds true where T is a nonsingular matrix:

$$R = T^T T \tag{4.52}$$

Using the above expression, Equation (4.49) can be rewritten as:

$$(A^T - K^T B^T)P + P(A - BK) + Q + K^T T^T TK = 0 \tag{4.53}$$

This can be further rewritten as:

$$A^T P + PA + [TK - (T^T)^{-1}B^T P] * [TK - (T^T)^{-1}B^T P] - PBR^{-1}B^T P + Q = 0 \tag{4.54}$$

Minimizing the performance index J in Equation (4.51) implies the minimization of $x^T[TK - (T^T)^{-1}B^T P] * [TK - (T^T)^{-1}B^T P]x$ with respect to K. This is valid if and only:

$$TK = (T^T)^{-1}B^T P \tag{4.55}$$

Finally, the optimal matrix K of the optimal control vector can be expressed as:

$$K = (T^T)^{-1}B^T P = R^{-1}B^T P, with\ R = T^T T \tag{4.56}$$

The optimal control law $u(t) = -Kx(t)$ can thus be given by

$$u(t) = -R^{-1}B^T Px(t) \tag{4.57}$$

While R can be obtained from Equation (4.52), P can be derived from Equation (4.49) or the following expression which can be extracted Equation (4.54) after minimizing the performance index i.e.

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \tag{4.58}$$

The expression in Equation (4.58) is also referred to as the *reduced-matrix Ricatti equation*. Note, the performance index can also be given in terms of the output vector rather than the state vector. The same procedure described above will still be followed in obtaining the optimal feedback gain K.

4.5.4 Model-Reference Adaptive Control

4.5.4.1 Overview

Designing a controller for a given system implies getting to know how that very system behaves physically, and this is oftentimes achieved through its mathematical representation. As shown in Chapter 3, the synchronous generator dynamics do have parameter variations that are due to its very nonlinear structure. Hence, reducing the system uncertainty as much as it can be practical is of the uttermost importance. The control strategy used in this thesis is the Model-Reference Adaptive Control (MRAC).

First, an overview of the composition of such a controller is introduced. Thereafter, the structure of the MRAC for the synchronous generator is presented.

Figure 4.9 depicts the steps followed in the design of this controller.
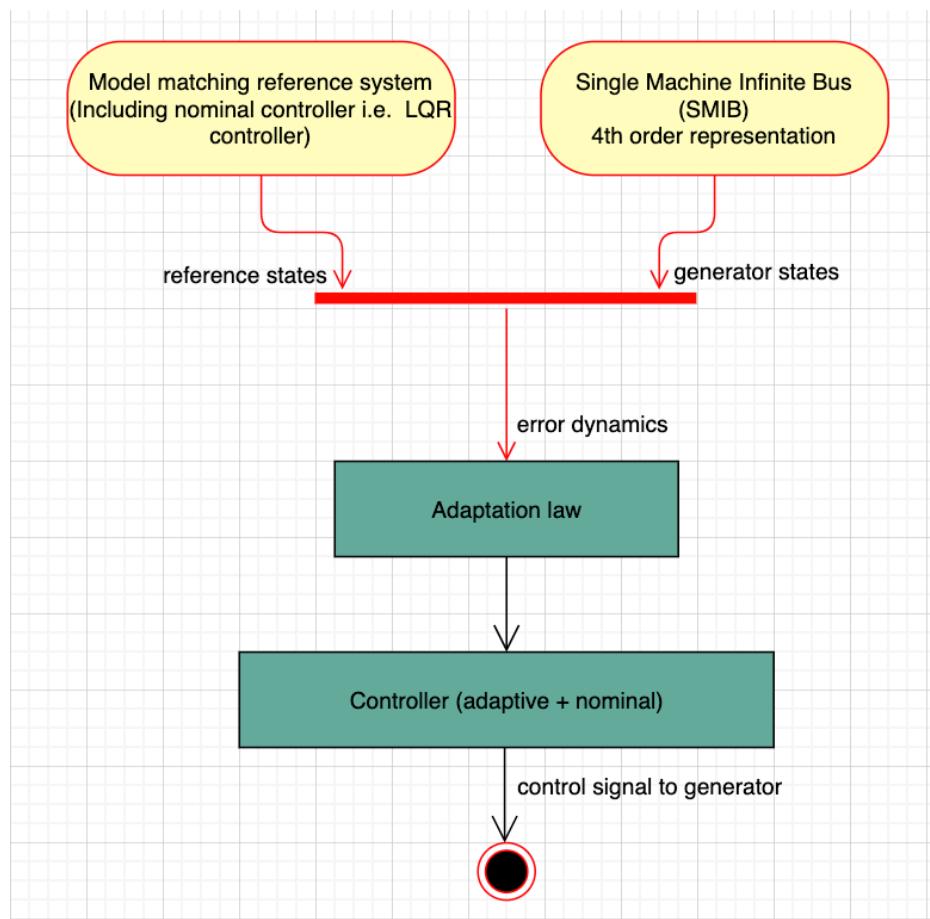
Figure 4.9: MRAC design architecture for power system interarea oscillations damping

4.5.4.2 Composition

Two classes of adaptive control are generally identified, the direct and indirect methods (Nguyen, 2018; Astrom and Wittenmark, 2008; Ioannou and Sun, 1996). Though either one of the classes are used in adaptive control architectures, oftentimes, they are combined and referred to as composite (Nguyen, 2018; Lavretsky, 2009), combined, or hybrid-direct adaptive control (Nguyen, 2018).

Figure 4.10 shows a typical structure of the MRAC system bloc.
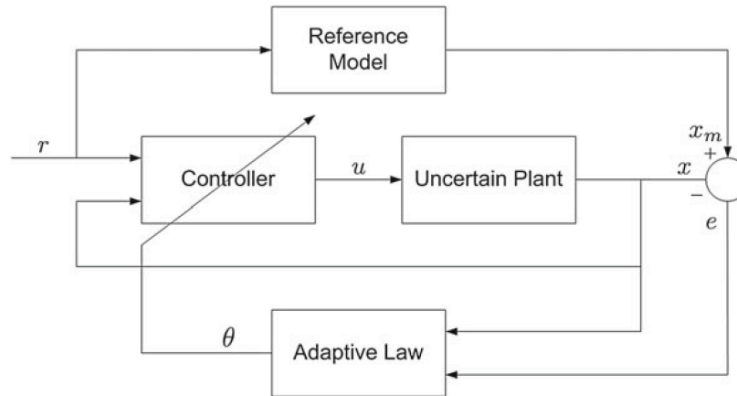
Figure 4.10: **Typical model-reference adaptive controller structure (Nguyen, 2018)**

Equation 4.40 shows a typical direct adaptive controller expression. In the direct method, the feedback control mechanism is directly adjusted to cancel any unwanted system uncertainty so to allow the performance of a given system to be regained thereafter (Nguyen, 2018).

$$u = k_x(t)x + k_r(t)r \tag{4.59}$$

where $k_x(t)$ and $k_r(t)$ are adjustable control gains.

As for the indirect class, the performance of the system is regained by adjusting the control gains indirectly as in Equation 4.41.

$$u = k_x(p(t))x + k_r(p(t))r \tag{4.60}$$

where $p(t)$ are online estimated system parameters used to update the control gains.

4.5.4.2.1 Uncertain Plant

As its name implies, uncertain plants are those that have some source of uncertainty within their structure. These uncertainties can either be structured, unstructured, or unmodelled (Nguyen, 2018).

Uncertain plant in this context is the 4th order representation of the synchronous generator.

Structured uncertainty: Referred to as parametric uncertainty, this type a source of uncertainty that has uncertain parameters, yet its functional characteristics are known. A typical example being a linear spring-mass damper system with an uncertain constant.

Unstructured uncertainty: In contrast with the structured uncertainty, neither its parameters nor its functional characteristics are certain. A system with similar features is a spring-mass-damper with uncertain spring characteristics.

Unmodelled dynamics: Unlike the first two, this source of uncertainty pertains to a given system internal or external dynamics not included in a plant model because they are either unmeasurable, unobservable, or incorrectly assumed to be negligeable.

Matched uncertainty: Given a Multiple Input Multiple Output (MIMO) linear affine-in-control system as in Equation 4.42, this type of uncertainty is in fact a structured uncertainty that can be matched by the control input of such as system.

$$\dot{x} = f(x) + B\left[u + \Theta^{*T}\,\Phi(x)\right] \tag{4.61}$$

where

$x(t) \in \mathbb{R}^n$ is the state vector

$u(t) \in \mathbb{R}^m$ is the control vector

$B \in \mathbb{R}^n x\ \mathbb{R}^m$ is the control input matrix

$\Theta^* \in \mathbb{R}^p x\ \mathbb{R}^m$ is a matrix of uncertain parameters

$\Phi(x) \in \mathbb{R}^p$ is a known bounded regressor function.

$\Theta^{*T}\Phi(x)$ is referred to as the parametric matched uncertainty.

Unmatched uncertainty: Some MIMO linear affine-in-control system with the structure as in Equation 4.43 and whose control input matrices are non-square or are rank-deficient such that their inverse does not exist, do have a source of uncertainty that cannot be matched by these very control input matrices.

$$\dot{x} = f(x) + Bu + \Theta^{*T}\Phi(x) \tag{4.62}$$

Since the control input cannot completely cancel the uncertainties by adaptive control, the (unmatched) uncertainty can be cast as a matched uncertainty via the pseudo-inverse transformation in Equation 4.44 below:

$$\dot{x} = f(x) + B\left[u + B^T(BB^T)^{-1}\Theta^{*T}\Phi(x)\right] \tag{4.63}$$

where

$B \in \mathbb{R}^n x\ \mathbb{R}^m$ is either a full-rank non-square wide matrix with $n < m$ and $rank(B) = n$, or a full-rank square matrix.

$B^T(BB^T)^{-1}$ is the right pseudo-inverse of such as matrix.

Control input uncertainty: Given a MIMO linear affine-in-control system with a structure as in Equation 4.45, this uncertainty is one that exists in the control input matrix.

$$\dot{x} = f(x) + B\Lambda u \tag{4.64}$$

where Λ is a positive diagonal matrix, whose diagonal elements represent the control input effectiveness uncertainty which can be in the amplitude or in the sign or both.

4.5.4.2.2 Reference Model

The adaptive control being formulated as a tracking control problem, the adaptation is aimed at tracking the error between a given reference model and the system output. The former being in a sense a shaping filter used to achieve the desired behaviour This error can either be based on system states or outputs error.

From the above, it is therefore important that it is well designed. Moreover, Linear Time Invariant (LTI) systems are proposed as they can be designed such that key performance specifications are met (rise time, settling time, …).

Considering that the objective of an adaptive control system is to adapt a given system's uncertainty so that the tracking error is minimized $(e(t) \rightarrow 0)$, the states of this very system must follow the reference model perfectly i.e. $x(t) \rightarrow x_m(t)$.

4.5.4.2.3 Controller

This is the baseline as its structure is aimed at ensuring the overall system performance and stability are provided for a nominal plant without uncertainty.

Though dependent on the objective of the control, this controller can either be a nominal controller augmented with an adaptive controller or a fully adaptive controller. The adaptive augmentation being more prominent and deemed more robust than their fully adaptive counterpart (Nguyen, 2018).

4.5.4.2.4 Adaptive Law

This is a mathematical relationship that illustrates how adaptive parameters should be adjusted to keep the tracking error as small as possible. This law can either be linear time-varying or nonlinear even though the stability of the adaptive control system is often analysed using the Lyapunov stability theory. Furthermore, the selection of the suitable adaptive law together with the set of tunning parameters built therein is often dictated by the trade-off between performance and robustness of the proposed adaptive control system.

4.5.4.3 Model-Reference Adaptive Controller Design for the Synchronous Generator

Equation 4.43 describing the 4th order representation of the synchronous generator's dynamics corresponds to that of a system with unmatched source of uncertainty as described in Equation 4.62.

If $\exists K_x$ and K_r that satisfy the model matching conditions as described in Equations 4.43a and 4.43b:

$$A + BK_x = A_m$$
$$BK_r = B_m$$

an LQR control is employed to find the optimal feedback matrix that will ensure the stability of the chosen reference-model.

Considering Equation 4.63, the adaptive controller is therefore designed as:

$$u = u_n + u_a \qquad (4.65)$$

where

$$u_n = \boldsymbol{K}_x x + \boldsymbol{K}_r r \text{ is the nominal controller}$$

$$u_a = -\boldsymbol{B}^T (\boldsymbol{B}\boldsymbol{B}^T)^{-1} \boldsymbol{\Theta}^T \Phi(x) \text{ the adaptation component}$$

4.5.4.3.1 Nominal Controller for the Synchronous Generator

It was mentioned that matrices **R** and **Q** determine the relative importance of the error and the expenditure of the energy of the control signals.

Various methods can be employed to determine their suitable structure based on a given application. In this thesis, output weighting is chosen for the Q matrix whereas R is chosen to be an identity matrix.

Let $R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and $Q = \tau * C^T C$, where the C is obtained from Equation 4.60, and τ the weighting factor. Increasing or decreasing this weighting factor has an influence on the balance between the importance of the error and control effort. Higher values for τ implies tuning the linearized generator's response to be faster while focusing on the importance of the error rather than the control effort.

Table 4.3: **Synchronous generator parameters (Eremia and Shahidehpour, 2013)**

Acronym	Value
x_d	1.8 p.u.
x_d'	0.3 p.u.
x_q	1.7 p.u.
x_q'	0.55 p.u.
x_q''	0.25 p.u.
x_d''	0.25 p.u.
D	0
H	6.5
S	900 MVA
T_{qo}'	0.4 s
T_{do}'	8 s
T_{do}''	0.03 s
T_{qo}''	0.05 s

Considering that the aim is to find matrices K_1 and K_2 such that $A - BK_1 = A_m$ and $BK_2 = B_m$,

    where:

        K_1 is the feedback gain

        K_2 is the feedforward gain

        A_m and B_m, the reference system matrix and reference control matrix respectively.

The quadratic optimal control problem can be solved with the following MATLAB script considering the generator parameters in Table 4.3.

MATLAB script 4.1

```
A =[0 0 0 0;1 0 0 0;0 0 -6.375 0;0 0 0 0.75];
B = [0 0;0 0.21;0 0;0.125 0];
C = [1 0 0 0;2.458 0 -1.326 2.28];
R = eye(2);
Q = 100 * Cᵀ * C;                         % moderately higher value of tau
[K,P,E] = lqr(A,B,C,Q,R);
```

The above script gives the values of the real symmetric matrix **P** and optimal control feedback gain **K**.

$$P = \begin{bmatrix} 290.5799 & 83.6609 & -37.0418 & 159.1003 \\ 83.6606 & 59.0966 & -2.4773 & 29.1826 \\ -37.0418 & -2.4773 & 12.4794 & -32.4421 \\ 159.1003 & 29.1826 & -32.4421 & 230.1267 \end{bmatrix}$$

$$K = \begin{bmatrix} 19.8875 & 3.6478 & -4.0553 & 28.7658 \\ 17.5688 & 12.4103 & -0.5202 & 6.1283 \end{bmatrix}$$

Yet, this linear controller's closed-loop response as in Figure 4.11 and 4.12 has quite an important steady-state error. This error has an amplitude greater than 0.9 unit.

Input = *Unit step*
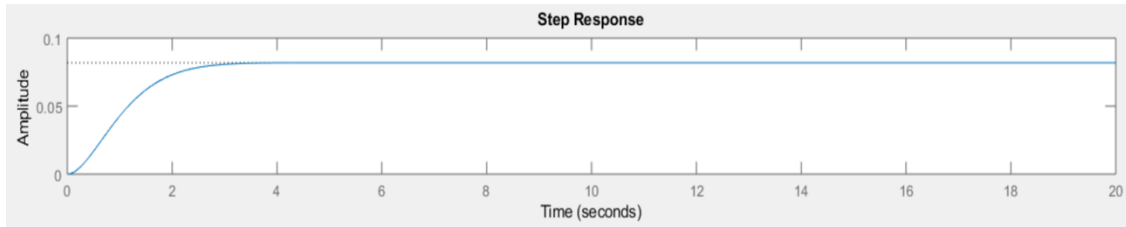
Figure 4.11: Output 1 - Rotor angle with $E_{FD} = 1$, $P_m = 1$

Input = *Unit step*
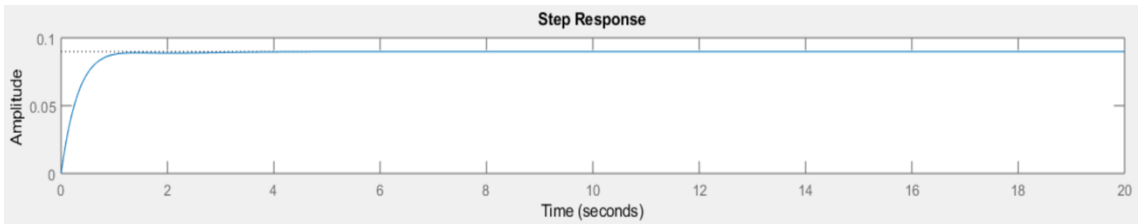
Figure 4.12: Output 2 - Electrical Power with $E_{FD} = 1$, $P_m = 1$

Therefore, changes to script 4.1 are required to address this as shown below:

MATLAB script 4.2

```
A =[0 0 0 0;1 0 0 0;0 0 -6.375 0;0 0 0 0.75];
B = [0 0;0 0.21;0 0;0.125 0];
C = [1 0 0 0;2.458 0 -1.326 2.28];
R = eye(2);
Q = 100 * Cᵀ * C;                          % moderately higher value of tau
[K₁,P,E]  = lqr(A,B,C,Q,R);

% Introduce a feedforward gain that cancels the steady-state error
K₂ = -inv(C * (inv(A - B*K) * B)
```

With the above changes, the system response has been improved and the structure of our linear controller has been found with $K_1 \equiv K$ the feedback gain and K_2 the feedforward gain.

Figures 4.13 and 4.14 shows the desired system response.

Input = Unit step
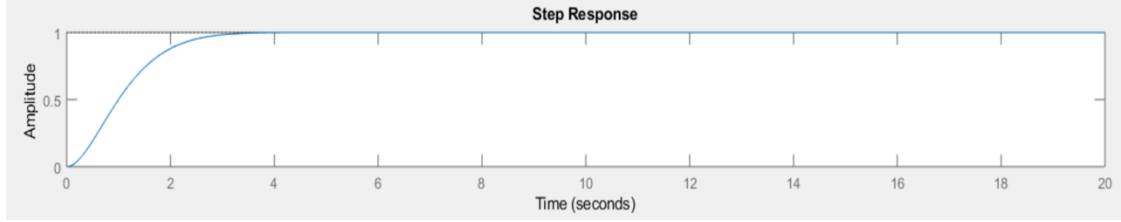
Figure 4.13: Output 1 - Rotor angle (No steady state error) with $E_{FD} = 1, P_m = 1$

Input = Unit step
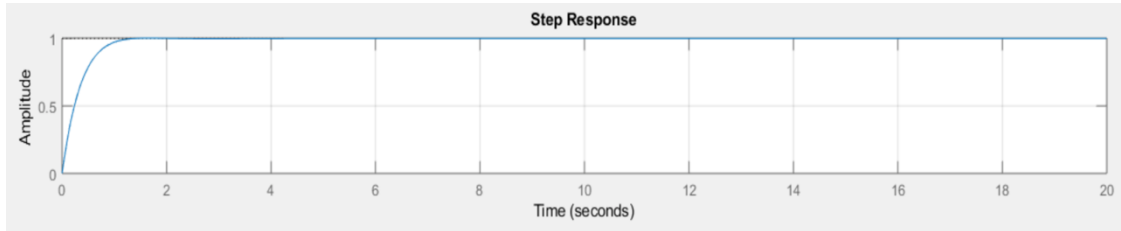
Figure 4.14: Output 2 - Electrical Power (No steady state error) with $E_{FD} = 1, P_m = 1$

4.5.4.3.2 Choice of the Adaptation Law

Defining $\tilde{\Theta}(t) = \Theta(t) - \Theta^*$ as the estimation error, the closed-loop synchronous generator model is expressed as

$$\dot{x} = (A + BK_x)x + BK_r r - B^T(BB^T)^{-1}\tilde{\Theta}^T\Phi(x) \tag{4.66}$$

Thus, the closed-loop tracking error is described as in Equation 4.67 below:

$$\dot{e} = \dot{x}_m - \dot{x} = A_m e + B^T(BB^T)^{-1}\tilde{\Theta}^T\Phi(x) \tag{4.67}$$

Choosing the following Lyapunov candidate

$$\dot{V} = -\Gamma\Phi(x)e^T PB^T(BB^T)^{-1} \tag{4.68}$$

and from the Barbalat's lemma, the tracking error can be shown to be asymptotically stable with $e(t) \to 0, \forall t \to \infty$.

4.6   Conclusion

This chapter introduced the Lyapunov stability theory and its application in the design of both the Lyapunov-based nonlinear controller as well as the Model-Reference Adaptive Controller (MRAC) for the 3rd and 4th order representation of a synchronous generator respectively in a Single-Machine Infinite Bus (SMIB) configuration.

The 3rd order representation is said to be useful in power system control studies whereas the 4th order is said to be suitable to model a given synchronous generator in full range of electromechanical oscillations i.e., both local and interarea oscillations (Eremia and Shahidehpour, 2013).

The theory behind using two different controllers for these generator's representations are provided together with their respective structures.

In contrast with the Lyapunov-based nonlinear controller whereby the type-1 servo system was used for the control of the reference model, optimal control techniques and most specifically the Linear Quadratic Regulator (LQR) is utilized for the control of the reference model of the MRAC used for the 4th order representation. This is due to the fact that its system matrix had unstable subspaces.

While Chapter 3 showed how nonlinear the synchronous generator's dynamics are, this chapter presented control techniques to be used to improve its stability.

Chapter 5 will showcase the performance of each one of these control schemes through simulations in the MATLAB environment. Various case studies for the validation of the performance of these controllers are described and analysed.

MATLAB IMPLEMENTATION OF THE PROPOSED POWER SYSTEM INTERAREA OSCILLATIONS DAMPING CONTROLLER ALGORITHM

5.1 Introduction

As stated in Chapter 4, two approaches are explored in the design of the best suited decentralized controller to damp power system interarea oscillations. The first being a reference-model based nonlinear controller based on Lyapunov theory, the other a model-reference adaptive controller.

In this this chapter, the performance of each of these methods are presented through digital simulations in MATLAB.

First, a revisit of the generator dynamics for the 3rd and 4th order representations is introduced in section 5.2. This is presented in the form of simulation results. Thereafter, the results of the performance of the reference-model based nonlinear controller based on Lyapunov theory are presented in section 5.3. Section 5.4 illustrates the performance of the proposed model-reference adaptive controller. Lastly, section 5.5 concludes this chapter.

5.2 Synchronous Generator Dynamics
5.2.1 Introduction

While detailed explanations on the structure and derivation of the linearized models of the 3rd and 4th order representations of the synchronous generator are given in Chapter 3, parameters used for the modelling can be found in Table 4.3.

The next subsections show the dynamics of these respective models in form of simulations results.

5.2.2 Third- Order Model Representation of the Synchronous Generator
5.2.2.1 System Modelling

From Equations 3.20a and 3.20b the state and output of the synchronous generator are represented as:

$$\underline{\dot{x}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\dfrac{D}{J} & 0 \\ 0 & 0 & \dfrac{1}{T_{do}'}\left(\dfrac{x_d}{x_d'}\right) \end{bmatrix} \begin{bmatrix} \delta \\ \omega \\ e_q' \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & \dfrac{1}{J} \\ \dfrac{1}{T_{do}'} & 0 \end{bmatrix} \begin{bmatrix} E_{FD} \\ P_m \end{bmatrix}$$

$$+ \begin{bmatrix} 0 \\ -\dfrac{1}{J}\left(\dfrac{V}{x_d'}e_q' \sin(\delta) + \dfrac{V^2}{2}\left(\dfrac{1}{x_q} - \dfrac{1}{x_d'}\right)\sin(2\delta)\right) \\ \dfrac{1}{T_{do}'}\left(\dfrac{x_d - x_d'}{x_d'}\right)V\cos(\delta) \end{bmatrix}$$

$$\underline{y} = \begin{bmatrix} \delta \\ P_e \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ V^2\left(\dfrac{1}{x_q} - \dfrac{1}{x_d'}\right)\cos(2\delta) & 0 & \dfrac{V}{x_d'}\sin(\delta) \end{bmatrix} \begin{bmatrix} \delta \\ \omega \\ e_q' \end{bmatrix}$$

where:

δ : rotor angle

ω : rotor speed

V : voltage on transformer's terminal

E_{FD} : induced emf by the field current (field voltage)

T_{do}' : d-axis open-circuit time constant

e_q' : q-axis transient emf

D : damping coefficient

P_e : air-gap power of the generator

x_d, x_q, and x_d' : augmented reactances with the line and transformer reactances added onto them (Leon-Morales et al., 2001; Sanchez-Orta et al., 2002)

The aforementioned equations could be re-written as in Equations 4.15a and 4.15b

$$\underline{\dot{x}} = A\underline{x} + B\underline{u} + \underline{F}(\delta)$$

$$\underline{y} = C\underline{x}$$

Using the values in Table 4.3, Equations (4.15a) and (4.15b) can be modelled as in Figure 5.1, with $\underline{F}(\delta)$ expressed by the MATLAB script 5.1
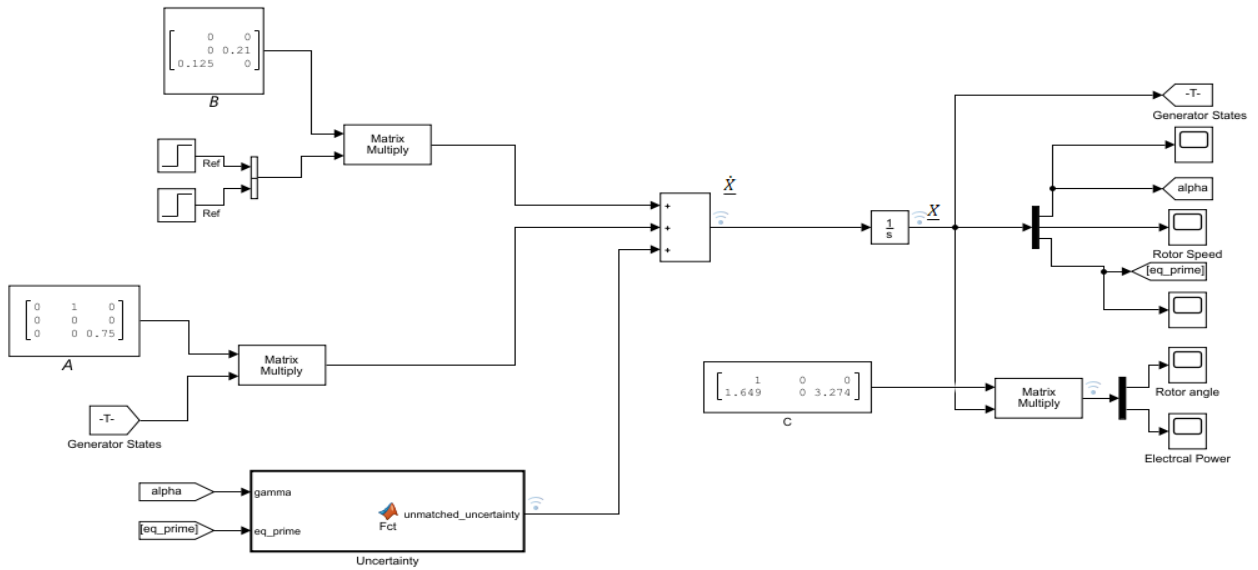
Figure 5.1: Synchronous generator 3rd order representation

MATLAB script 5.1

```matlab
function unmatched_uncertainty = Fct(gamma, eq_prime)


V = 1;
xd = 1.8;
xq = 1.7;
xd_prime = 0.3;
J = 4.774;
Td_prime = 8;


elt2 = (-V/J) * (sin(gamma)/xd_prime) * eq_prime;
elt3 = (-(V^2) / J) * ((1/xq) - (1/xd_prime)) * sin(gamma) *
cos(gamma);
row2 = elt2 + elt3;
row3 = (V/Td_prime) * ((xd/xd_prime) - 1) * cos(gamma);
F = [0;row2;row3];
unmatched_uncertainty = F;


end
```

5.2.2.2 Simulation Results

Over a 20 second window with a step input and no initial conditions, Figures 5.2-5.5 shows the response of the 3rd order model system representation of the synchronous generator.
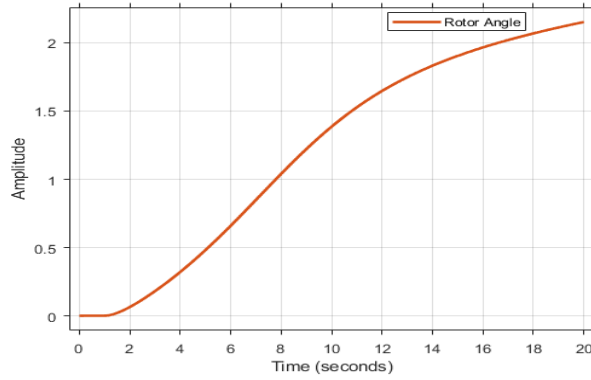
Figure 5.2: Generator rotor angle dynamics when
$E_{FD} = 1$ and $P_m = 1$
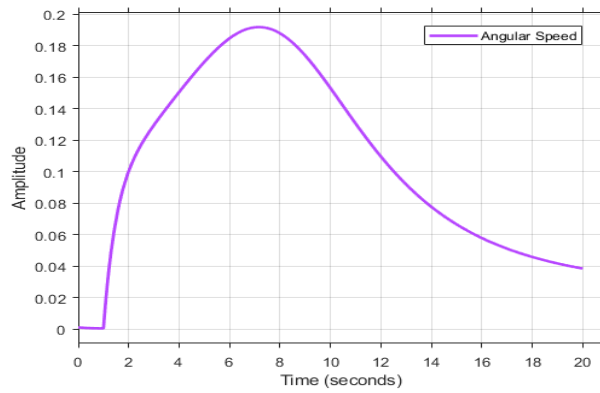
Figure 5.3: Generator rotor speed dynamics when
$E_{FD} = 1$ and $P_m = 1$
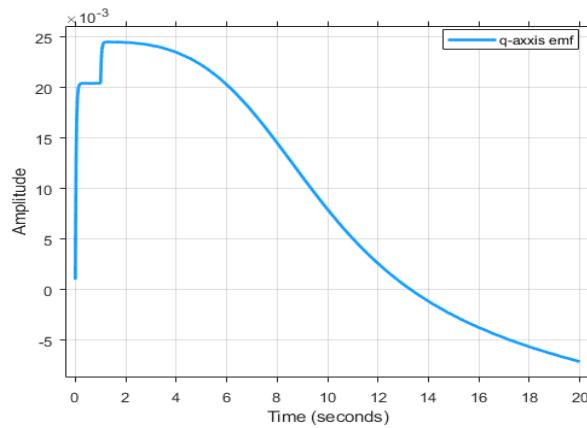
Figure 5.4: Generator q-axis transient emf dynamics
when $E_{FD} = 1$ and $P_m = 1$
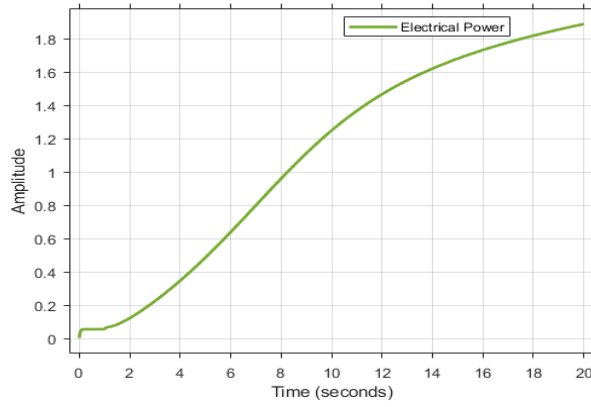
Figure 5.5: Generator electrical power dynamics
when $E_{FD} = 1$ and $P_m = 1$

5.2.2.3 Discussions

The 3rd order representation of the synchronous generator has revealed how unstable a synchronous generator's dynamics could be.

Though the angle's dynamics seems to show oscillations that tend to decrease, the speed, the electrical power and q-axis emf illustrates the instability as they have increased quite drastically from the 8th second for the first and 14th second for the last two.

Ensuring that this instability is mitigated is therefore critical.

5.2.3   Fourth-Order Model Representation of the Synchronous Generator

5.2.3.1 System Modelling

From Equation 3.22a the state of the synchronous generator is represented as:

$$
\underline{\dot{x}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \dfrac{-D}{J} & 0 & 0 \\ 0 & 0 & -\dfrac{1}{x_q' T_{qo}'}\left(x_q' + x_q - x_d'\right) & 0 \\ 0 & 0 & 0 & -\dfrac{1}{T_{do}'}\left(\dfrac{x_d}{x_d'}\right) \end{bmatrix} \begin{bmatrix} \delta \\ \omega \\ e_d' \\ e_q' \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & -\dfrac{1}{J} \\ 0 & 0 \\ \dfrac{1}{T_{do}'} & 0 \end{bmatrix} \begin{bmatrix} E_{FD} \\ P_m \end{bmatrix}
$$

$$
+ \begin{bmatrix} 0 \\ \dfrac{1}{J}\left(\dfrac{e_d' V}{x_q'}\cos\delta - \dfrac{e_q' V}{x_d'}\sin\delta - \dfrac{V^2}{2}\sin 2\delta\left(\dfrac{1}{x_q'} - \dfrac{1}{x_d'}\right)\right) \\ \dfrac{V}{T_{qo}'}\left(\dfrac{x_q - x_d'}{x_q'}\right)\sin\delta \\ \dfrac{V}{T_{do}'}\left(\dfrac{x_d}{x_d'} - 1\right)\cos\delta \end{bmatrix}
$$

103

where:

δ : rotor angle

ω : rotor speed

i_d : current in the d-axis

i_q : current in the q-axis

V : voltage on transformer's terminal

E_{FD} : induced emf by the field current (field voltage)

T_{do}' : d-axis open-circuit time constant

$\dot{e}_q'$: q-axis transient emf

$\dot{e}_d'$: d-axis transient emf

D : damping coefficient

P_e = air-gap power of the generator

As in Equation (4.43), this can be re-written as:

$$\dot{\underline{x}} = A\underline{x} + B\underline{u} + \underline{F}(\delta)$$

where $\underline{F}(\delta)$ is a vector of the nonlinear elements also referred to as the *unmatched uncertainty*.

Furthermore, Equation (3.22b), the output has the following expression:

$$Y = \begin{bmatrix} \delta \\ P_e \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \dfrac{V^2}{2\delta}\left(\dfrac{1}{x'_q} - \dfrac{1}{x'_d}\right)\sin(2\delta) & 0 & \dfrac{-V\cos\delta}{x'_q} & \dfrac{V\sin\delta}{x'_d} \end{bmatrix} \begin{bmatrix} \delta \\ \omega \\ e'_d \\ e'_q \end{bmatrix}$$

Using the values in Table 4.3, the above can be modelled as in Figure 5.6, with $\underline{F}(\delta)$ expressed by the MATLAB script 5.2.
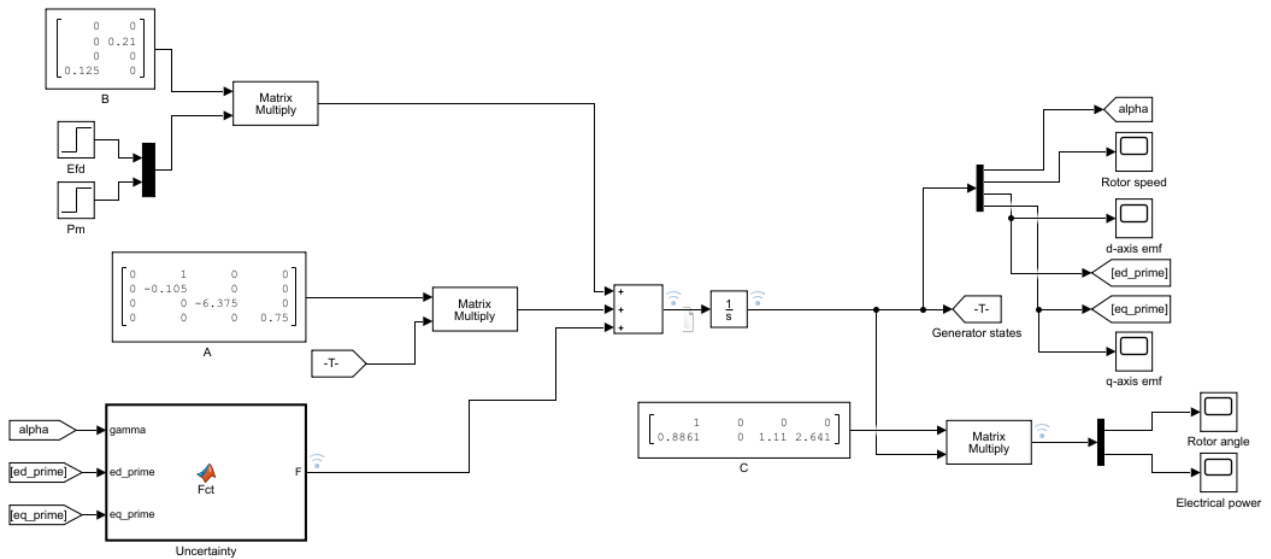
Figure 5.6: **4**[th] **order representation of the synchronous generator**

```
MATLAB script 5.2

function F = Fct(gamma, ed_prime, eq_prime)


V = 1;
xd = 1.8;
xq = 1.7;
xd_prime = 0.3;
xq_prime = 0.55;
J = 4.774;
Td_prime = 8;
Tq_prime = 0.4;


elt1 = (1/J) * (V*cos(gamma)/xq_prime) * ed_prime;
elt2 = (V/J) * (sin(gamma)/xd_prime) * eq_prime;
elt3 = ((V^2) / J) * ((1/xq_prime) - (1/xd_prime)) * sin(gamma) * cos(gamma);
row2 = elt1 - elt2 - elt3;
row3 = (V/Tq_prime) * ((xq-xd_prime) / xq_prime) * sin(gamma);
row4 = (V/Td_prime) * ((xd/xd_prime) - 1) * cos(gamma);
F = [0;row2;row3;row4];


end
```

5.2.3.2 Simulation Results

With no initial conditions and over a 20 second window with a step input, Figures 5.7-5.11 show the response of the 4th order model system representation of the synchronous generator.
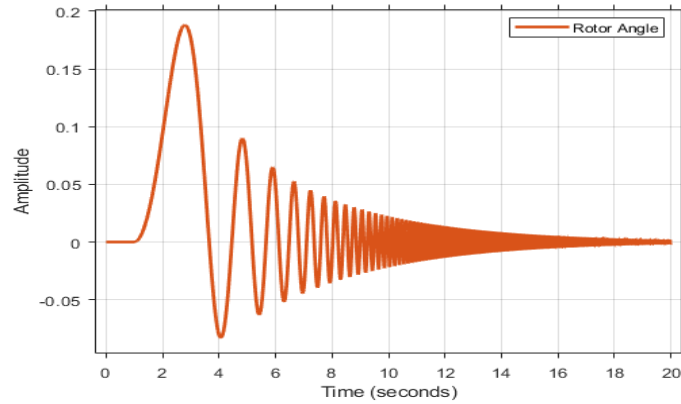
Figure 5.7: Generator rotor angle dynamics when $E_{FD} = 1$ and $P_m = 1$
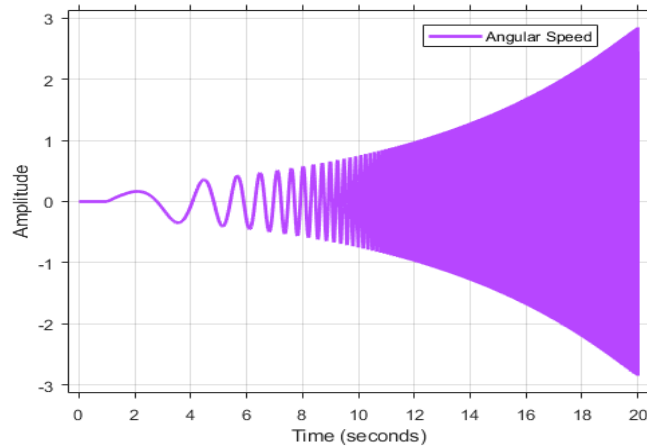
Figure 5.8: Generator rotor speed dynamics when $E_{FD} = 1$ and $P_m = 1$
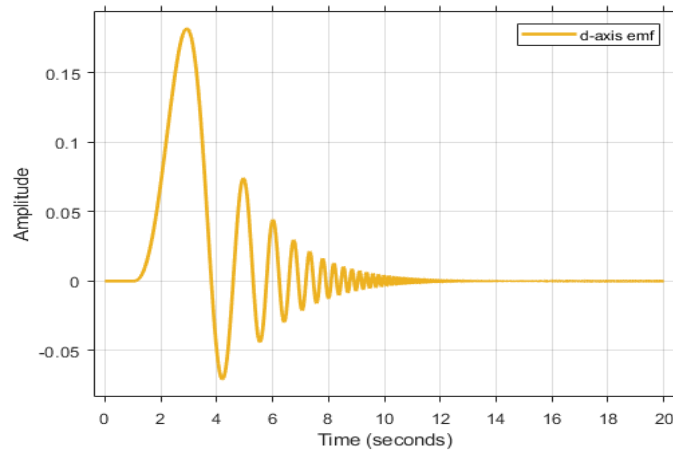
Figure 5.9: Generator d-axis dynamics when $E_{FD} = 1$ and $P_m = 1$

106
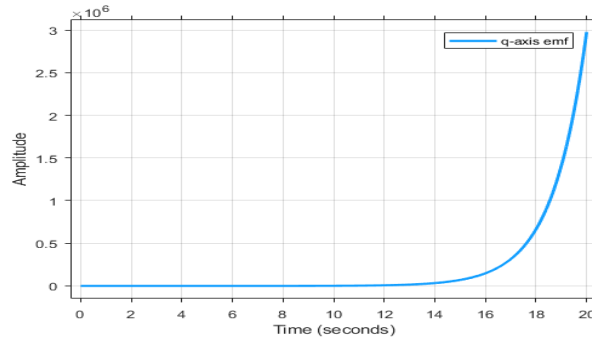
Figure 5.10: Generator q-axis dynamics when $E_{FD} = 1$ and $P_m = 1$
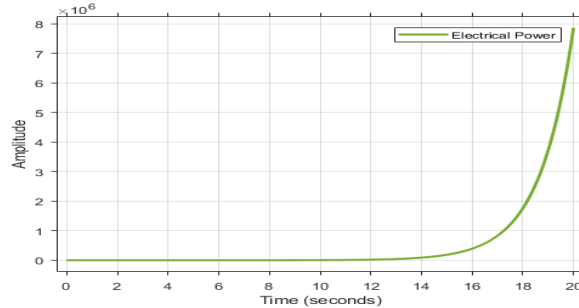
Figure 5.11: Generator electrical power dynamics when $E_{FD} = 1$ and $P_m = 1$

5.2.3.3 Discussions

Similar to the 3rd order representation of the synchronous generator, this model also revealed how unstable a synchronous generator's dynamics could be.

Though the d-axis emf and the angle's dynamics seems to show oscillations that tend to decrease, the speed, the electrical power and q-axis emf illustrates the instability as they have increased quite drastically from the 8th second for the first and 14th second for the last two. Eremia and Shahidehpour (2013) emphasized that the time of interest for Low Frequency Electromechanical Oscillations (LFEOs) is 3-5 seconds and could extend to 10 seconds for large power systems with weak interconnections between remote generators. Thus, ensuring that this instability is mitigated is very critical.

5.3   Performance of the Nonlinear Controller Based on Lyapunov Stability Theory for the Third-Order Representation of the Synchronous Generator
5.3.1   Introduction

A Detailed explanation on the design of the controller is presented in Chapter 4; and as stated therein, a *type-1* servo will be used for the control of the reference model.

A Lyapunov reference-model based nonlinear controller is therefore proposed for stability improvement of the system modelled in section 5.2.2.

First, the modelling and simulation of the reference model is introduced. Then, the Lyapunov nonlinear controller's MATLAB translation is given. Thereafter, the decentralized interarea oscillations damping control scheme is presented.

5.3.2 Closed-loop Modelling and Simulation of the reference model

The reference model used is the linearized model of the 3rd order system. The closed-loop representation and response of this model including its controller are given in Figures 5.12, 5.13, and 5.14. The system is modelled over a 20 second window with no initial conditions. The set points are the field voltage and mechanical power, each having unity as amplitude (step inputs).
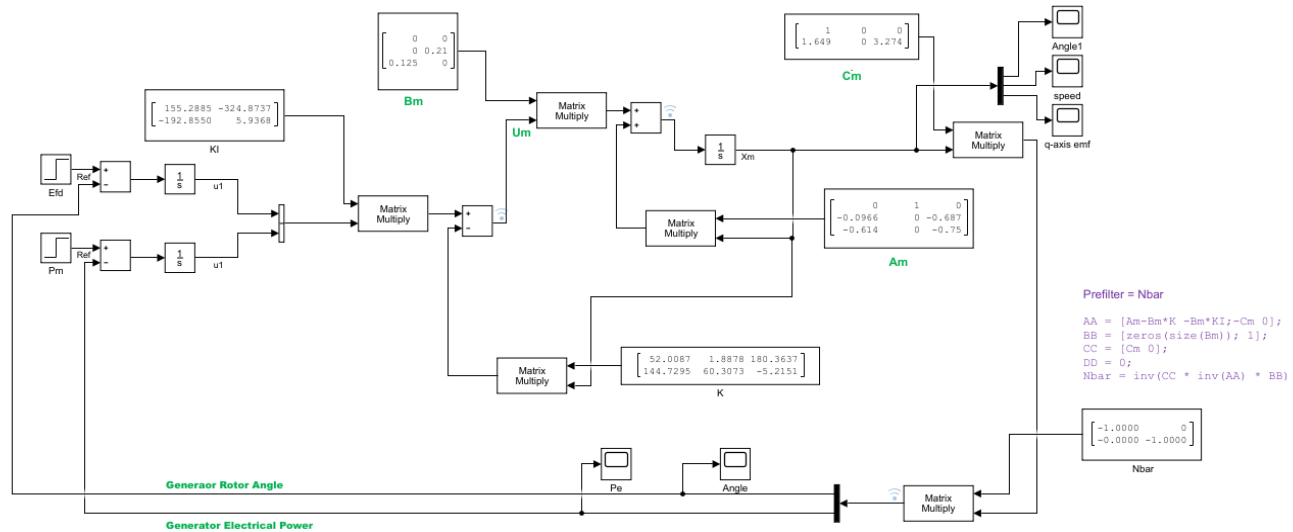
Figure 5.12: Linearized model of the 3rd order representation of the synchronous generator with a *type-1* servo controller when $E_{FD} = 1$ and $P_m = 1$
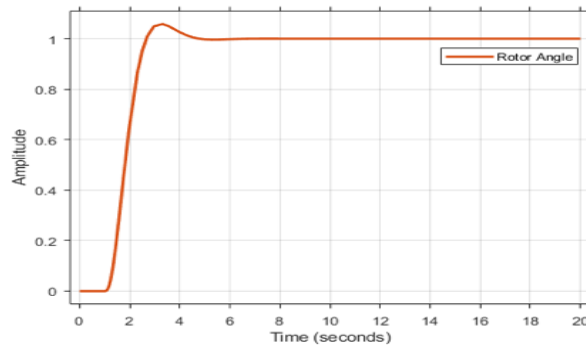
Figure 5.13: Generator rotor angle dynamics when *type-1* servo is applied onto it, and with $E_{FD} = 1$ and $P_m = 1$
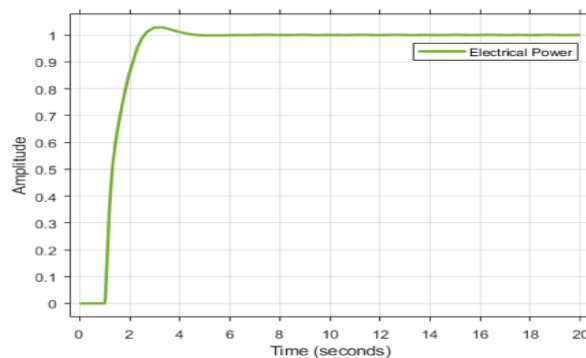
Figure 5.14: Generator electrical power dynamics when *type-1* servo is applied onto it, and with $E_{FD} = 1$ and $P_m = 1$

5.3.3 Lyapunov-based Nonlinear Controller

As in Equation (4.41), the expression of the nonlinear controller is given by:

$$U > [(E^T PB)^T E^T PB]^{-1} (E^T PB)^T E^T P(A_m X + B_m U_m - AX - F(\delta))$$

The above equation can be rewritten as:

$$U = \mathbf{Z} * \lambda \tag{5.1}$$

Where:

$$\mathbf{Z} = [(E^T PB)^T E^T PB]^{-1} (E^T PB)^T E^T P(A_m X + B_m U_m - AX - F(\delta))$$

$\lambda > 1$: a scalar gain

This can be translated as in the MATLAB script 5.3 below.

MATLAB script 5.3

```matlab
function U = adaptive_lyapunov( Am, Bm, Um, A, B, X, F, Error )
U = zeros(2,1);

% Positive definite matrix P
P = [15.5833    12.1389     0.0278;
     12.1389    11.7269     0.8426;
      0.0278     0.8426     0.6713];

df = isequal(Error,[0;0;0]);
if df
   U = [0;0];
else
    Uno = Error' * P * B;
    Dos = Uno' * Uno;
    Tres = Error'*P;
    Quatro = Tres *(Am * X);
    Quinto =  Tres *(Bm * Um);
    Sexto = Tres * (A * X);
    Septo = Tres * F;
    temp1 = Dos\Uno';
    temp2 = (Quatro + Quinto - Sexto - Septo);
    temp3 = temp1 * temp2;
    control = temp3 * λ;
    if isfinite(control) == 0
        U = [0;0];
    else
        U = control;
    end
 end

 end
```

5.3.4 Interarea Oscillations Damping Controller Scheme

5.3.4.1 System Modelling

The decentralized oscillations damping scheme in Figure 5.15 comprises 3 components:

- (1): The reference model whose structure has been presented in sub-section 5.3.2

- (2): The nonlinear controller introduced in sub-section 5.3.3

- (3): The 3rd order of the synchronous generator modelled in sub-section 5.2.2.
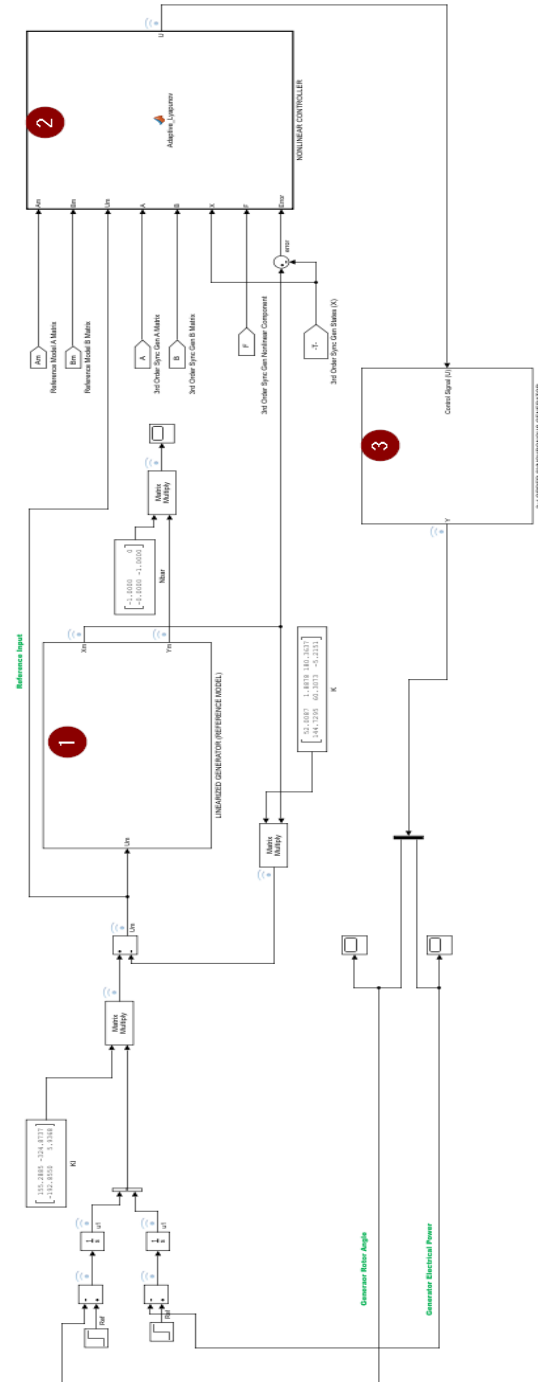
Figure 5.15: Power system interarea oscillation decentralized controller scheme based on Lyapunov theory when $E_{FD} = 1$ and $P_m = 1$

Figure 5.15 can be simplified as in Figure 5.16 below.
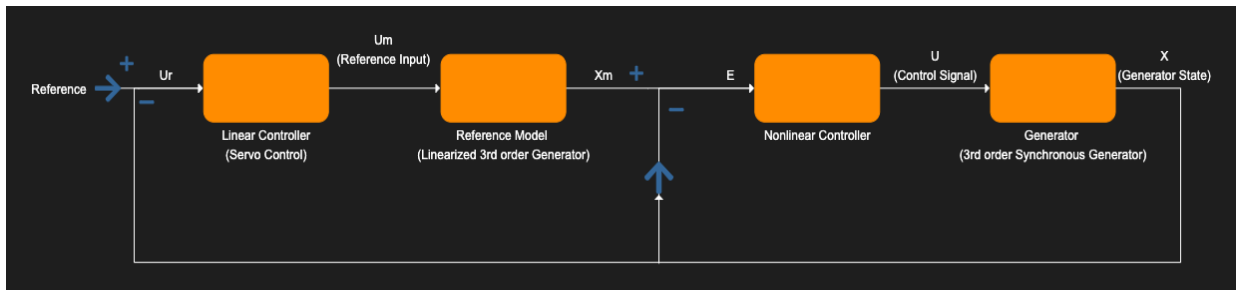
Figure 5.16: Power system interarea oscillation decentralized controller architecture based on Lyapunov theory

5.3.4.2 Simulation Results

From Equation (5.1), it can be seen that the control signal is dependent on the value of the gain λ. Hence, for various values of the very gain, different behaviors can be expected with the ultimate gain to find the most suitable one that ensures system stability and damps oscillations.

It is worth noting that finding the suitable gain that would improve the stability was somehow challenging as there are no known reference in literature for this specific type of controller. While this section only contains the value that produced acceptable results, the remaining ones together with the generator's output behaviours are listed in the appendices.

Empirical methods were used to obtain the value of λ that provided good system response. The performance of this control scheme is assessed by change of set-points which were initially $E_{FD} = 1$, $P_m = 1$. The parameters used in the simulation are shown in Table 5.1, and initial conditions are assumed to be null.

Table 5.1: Synchronous generator parameters (Eremia and Shahidehpour, 2013)

Acronym	Value
x_d	1.8 p.u.
x_d'	0.3 p.u.
x_q	1.7 p.u.
x_q'	0.55 p.u.
x_q''	0.25 p.u.
x_d''	0.25 p.u.
D	0
H	6.5
S	900 MVA
T_{qo}'	0.4 s
T_{do}'	8 s
T_{do}''	0.03 s
T_{qo}''	0.05 s

Case Study 1: Steady State - No initial conditions and $E_{FD} = 1$, $P_m = 1$ as setpoints
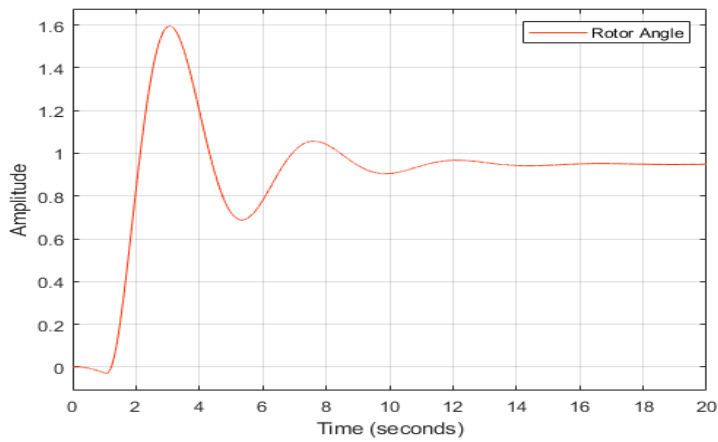
Figure 5.17: Synchronous generator rotor angle when $E_{FD} = 1$, $P_m = 1$ and $\lambda = 4.5$
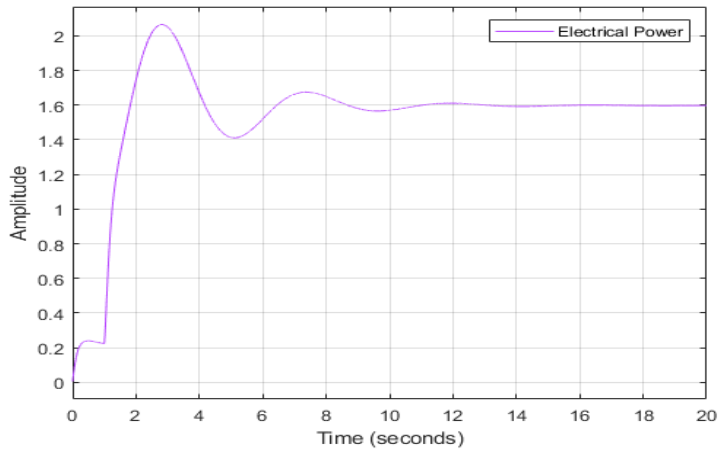
Figure 5.18: Synchronous generator electrical power when $E_{FD} = 1$, $P_m = 1$ and $\lambda = 4.5$

Case Study 2: No initial conditions and change of setpoint - $E_{FD} = 2.395$, $P_m = 1$
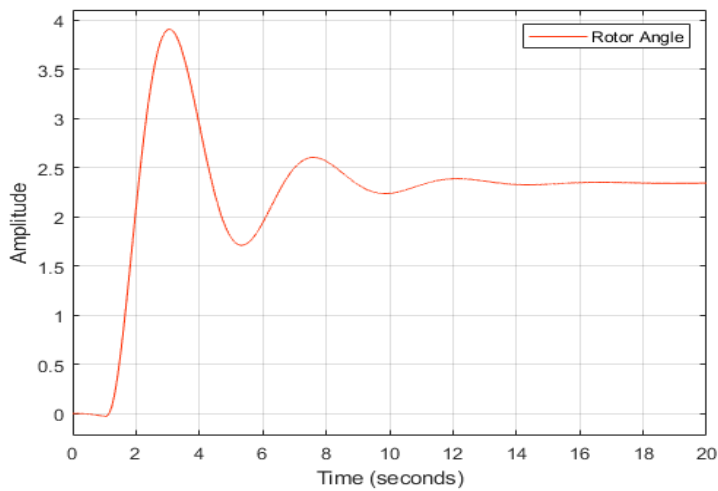
Figure 5.19: Synchronous generator rotor angle when $E_{FD} = 2.395$, $P_m = 1$ and $\lambda = 4.5$
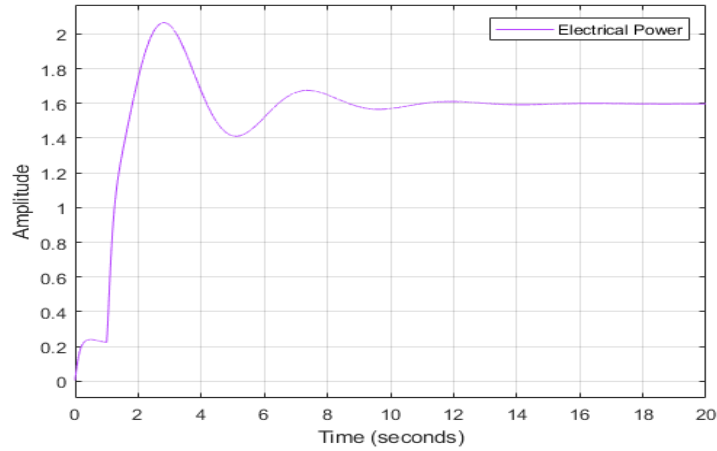
Figure 5.20: Synchronous generator electrical power when $E_{FD} = 2.395$, $P_m = 1$ and $\lambda = 4.5$

Case Study 3: No initial conditions and change of setpoint - $E_{FD} = 1$, $P_m = 0.77778$

Figure 5.21: Synchronous generator rotor angle when $E_{FD} = 1$, $P_m = 0.77778$ and $\lambda = 4.5$
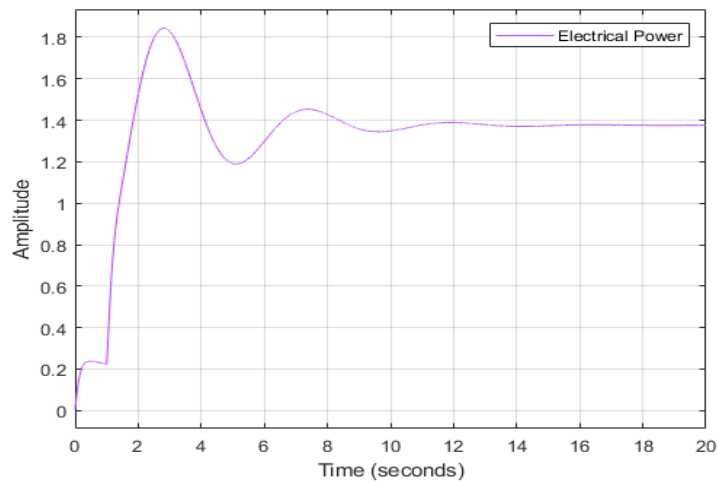
Figure 5.22: Synchronous generator electrical power when $E_{FD} = 1$, $P_m = 0.77778$ and $\lambda = 4.5$

Case Study 4: No initial conditions and change of setpoint - $E_{FD} = 2.395, P_m = 0.77778$
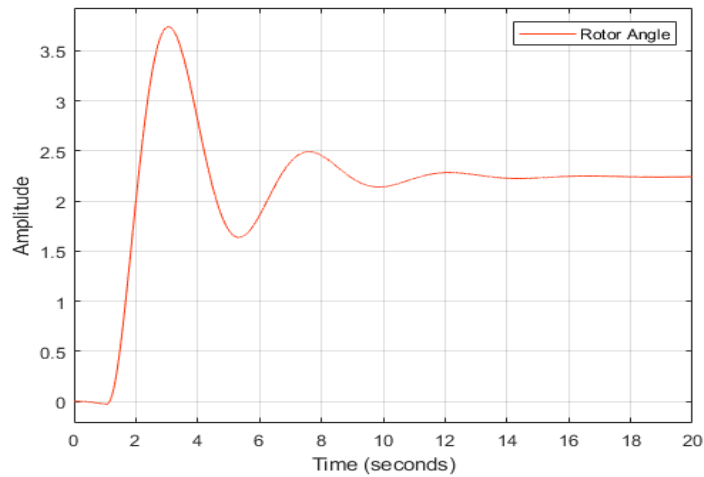
Figure 5.23: Synchronous generator rotor angle when $E_{FD} = 2.395, P_m = 0.77778$ and $\lambda = 4.5$
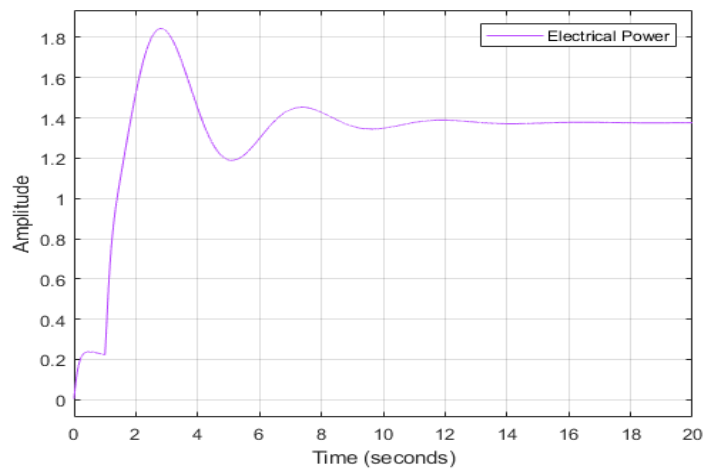
Figure 5.24: Synchronous generator electrical power when $E_{FD} = 2.395, P_m = 0.77778$ and $\lambda = 4.5$

From the above, it can be seen that changes in the field voltage input affect the rotor angle alone whereas the electrical power is only affected by changes in the mechanical input power. This is illustrated in Figures 5.25 and 5.26.
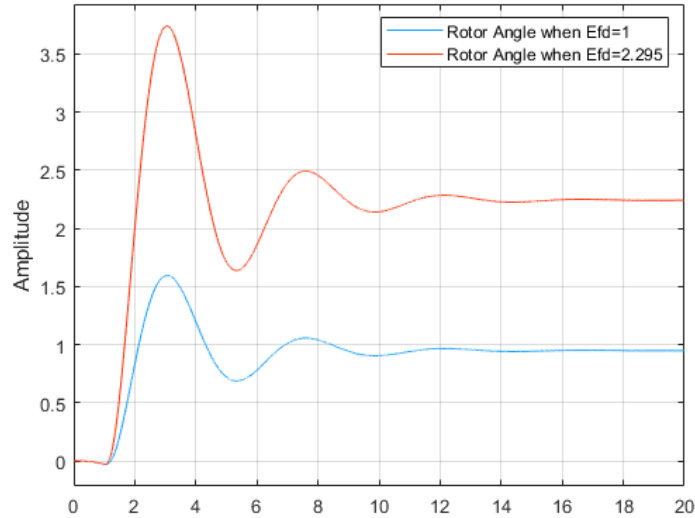
Figure 5.25: Synchronous generator rotor angle characteristics as E_{FD} changes
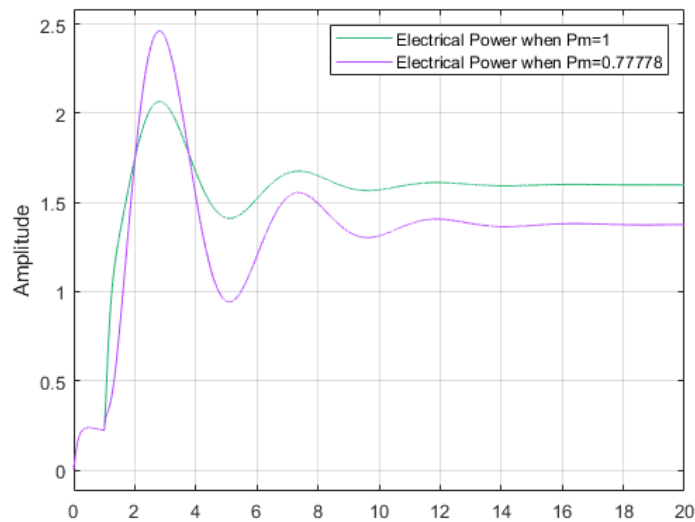
Figure 5.26: Synchronous generator electrical power characteristics as P_m changes

The introduction of an internal disturbance in the form of Gaussian noise as in Figure 5.28 to emulate low frequency oscillations caused by small variations in loads produced no good results. Also investigated are the impact faults could have on the Lyapunov controller. The results of the above are presented in Case 5-8. Though in the previous test cases the both the rotor angle and electrical power's responses were shown, in the subsequent cases only the first is studied. The aim here is the illustrate how sensitive a Lyapunov-based nonlinear controller could be, and the rotor angle' behaviour alone is sufficient to determine a synchronous generator's stability.
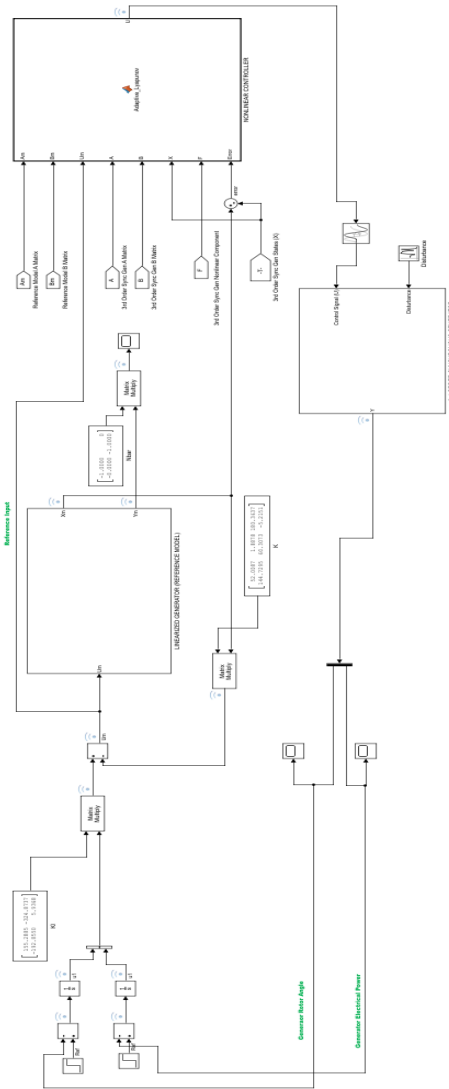
Figure 5.27: Power system interarea oscillation decentralized controller scheme based on Lyapunov theory when $E_{FD} = 1$, $P_m = 1$ and added disturbance

Case Study 5: Added Gaussian noise and - $E_{FD} = 1$, $P_m = 1$
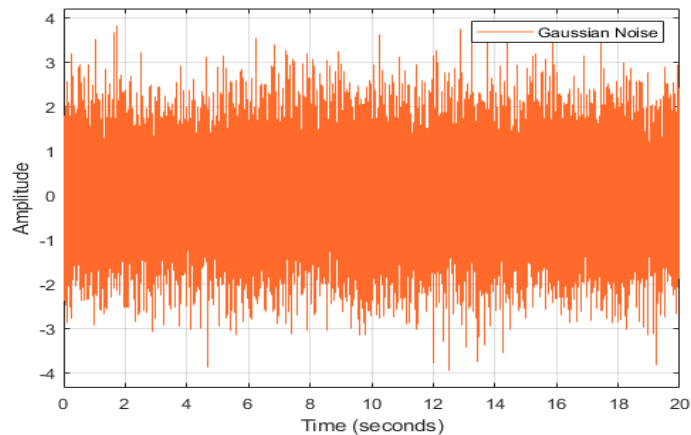
Figure 5.28: Gaussian noise characteristics
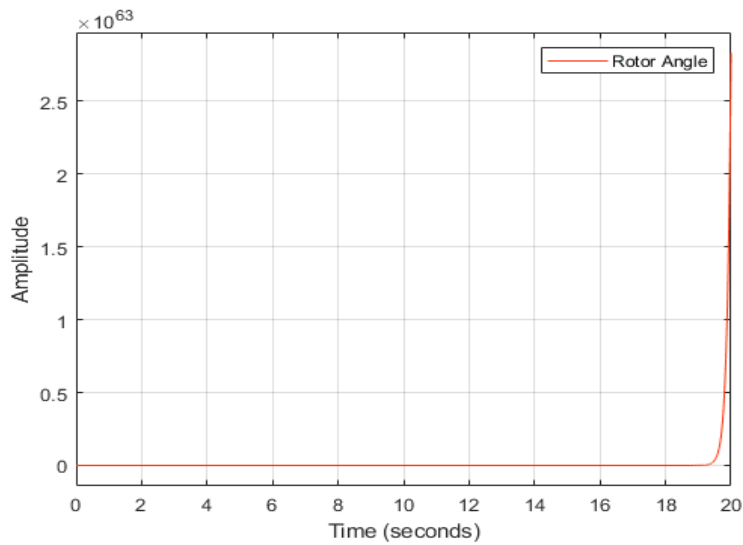
Figure 5.29: Synchronous generator rotor angle when internal disturbances are included to the generator structure and $E_{FD} = 1$, $P_m = 1$

Case Study 6: **Disturbances added to input but no change of setpoint -** $E_{FD} = 1$, $P_m = 1$ **nor initial conditions**

The disturbances added onto the input have the same characteristics as those shown in Figure 5.28.

As illustrated in Figure 5.30, the synchronous generator behaves in a opposite manner than it did when subjected to internal disturbances i.e., the angle grows bigger abruptly after the nineteenth (19 s) but towards negative values. The values reached by the rotor angle are in the range of $\geq 10^{55}$
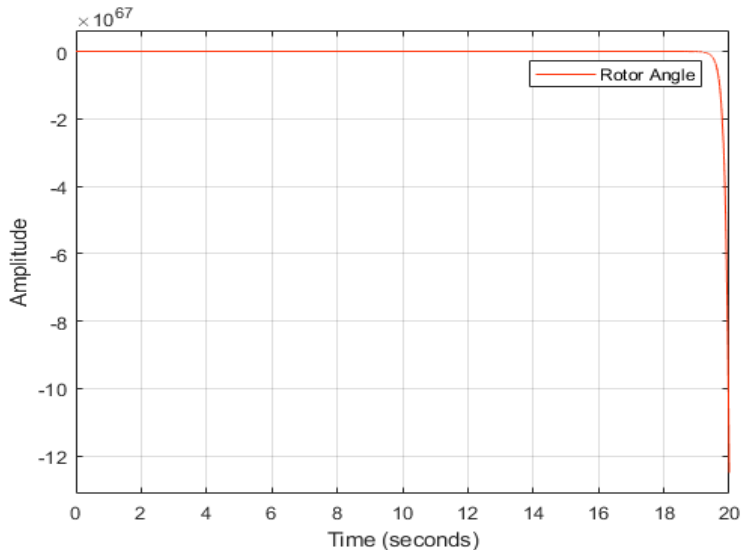
Figure 5.30: Synchronous generator rotor angle when disturbances are added onto the input and $E_{FD} = 1$, $P_m = 1$

Case Study 7: Added fault at t = 9s - no change of setpoint - $E_{FD} = 1$, $P_m = 1$ and no initial conditions
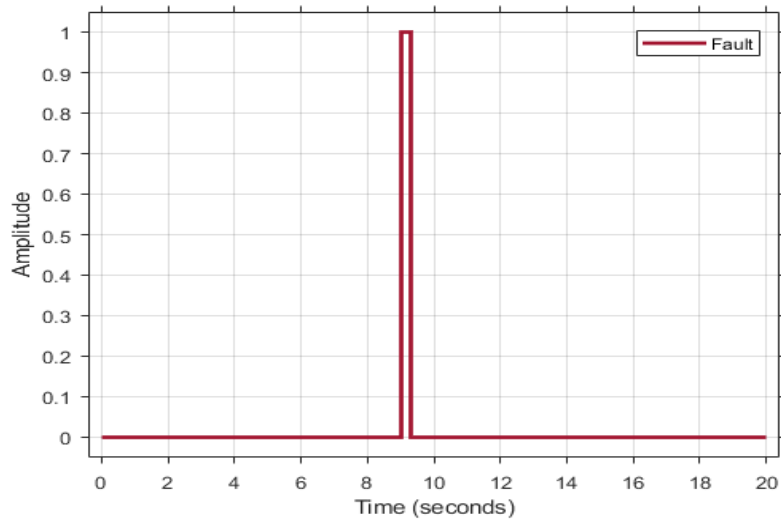
Figure 5.31: Fault characteristics - introduced at t = 9 s for a duration of 300ms
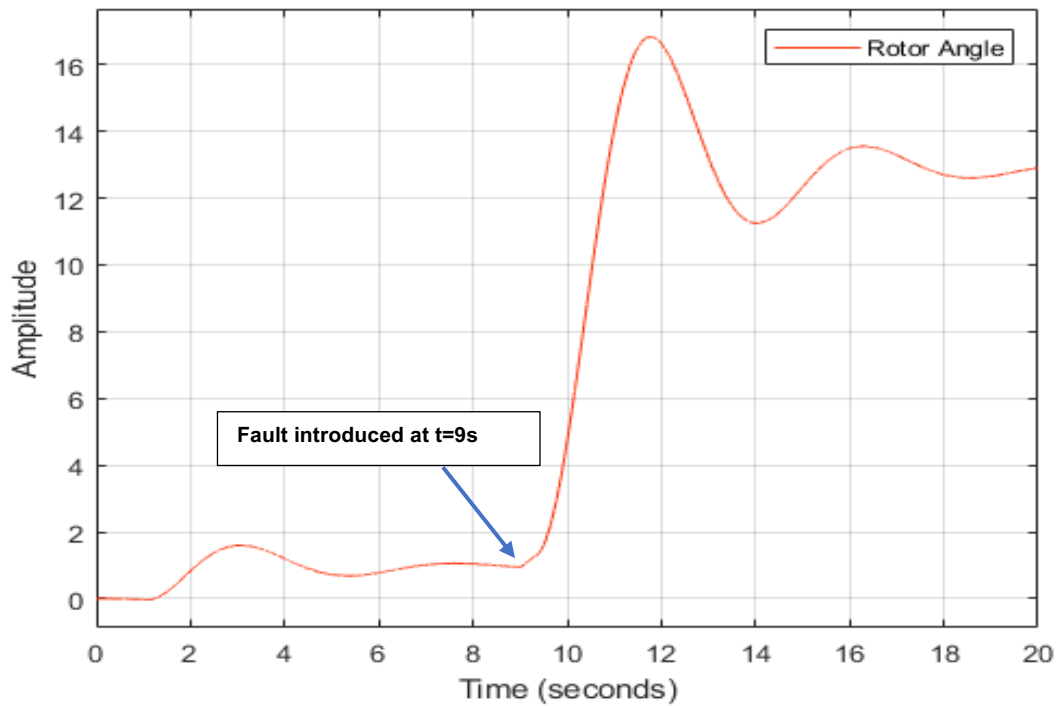
Figure 5.32: Synchronous generator rotor angle when fault is introduced at t = 9s and $E_{FD} = 1$, $P_m = 1$

Case Study 8: Added fault at t = 3s - no change of setpoint - $E_{FD} = 1$, $P_m = 1$ and no initial conditions
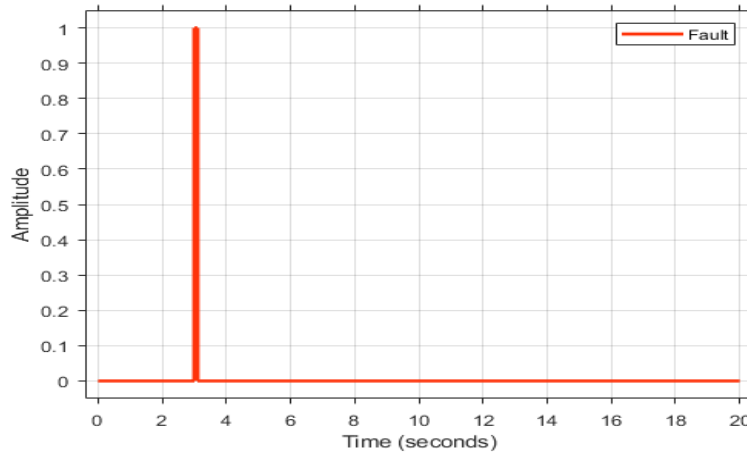
Figure 5.33: Fault characteristics - introduced at t = 3 s for a duration of 100ms
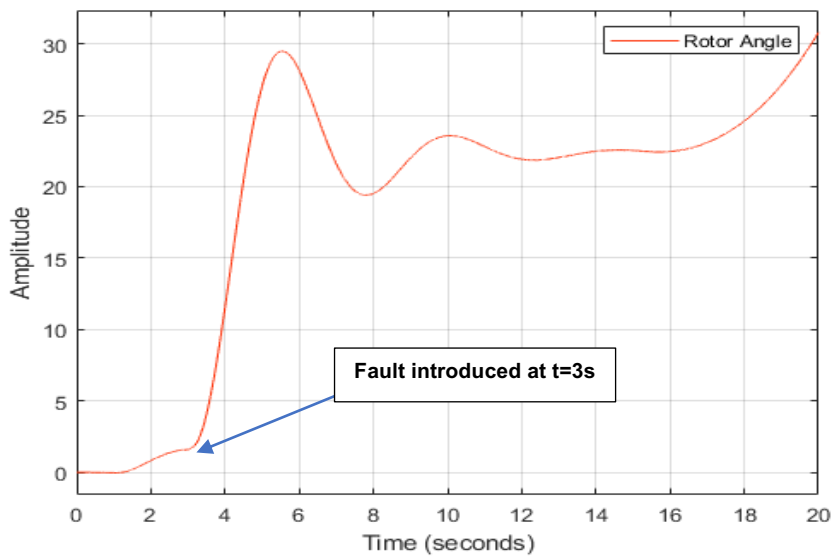
Figure 5.34: Synchronous generator rotor angle when fault is introduced at t = 3s and $E_{FD} = 1, P_m = 1$

5.3.4.3 Discussions

Most of the challenges arose from the selection of the suitable gain that would make the system follow the behaviour of the reference model due to limited literature in that regard. Though successfully used for wastewater control (Paseka, 2009), this controller did not yield good results for the synchronous generator control.

With a gain of 4.5, the rotor angle's response was acceptable under steady-state conditions. However, changes in setpoints as well as the introduction of disturbances in the system showcased the limitations of such controller for power system oscillations

119

damping. These disturbances were in form of gaussian noise to emulate LFEOs. Faults were also introduced to assess the robustness of the controller.

The sensitivity of this controller can easily be identified as the overall stability is lost once a change in setpoint or disturbances are introduced into the system.

Figures 5.35 and 5.36 illustrates the system whose tests results are shown in Case studies 6, 7, and 8.
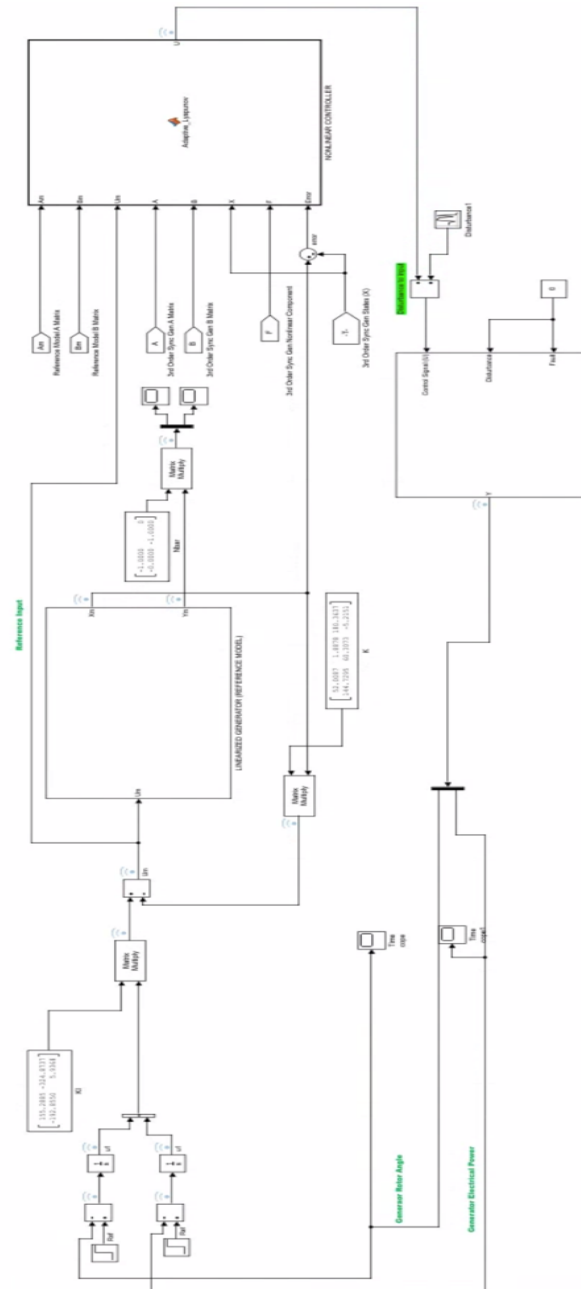
Figure 5.35: Power system interarea oscillation decentralized controller scheme based on Lyapunov theory with added input disturbances

Figure 5.36: Power system interarea oscillation decentralized controller scheme based on Lyapunov theory with added input disturbances

121

5.4   Performance of the Model-Reference Adaptive Control (MRAC) Scheme for the Fourth-Order Representation of the Synchronous Generator

5.4.1   Introduction

Detailed explanation on the design of the controller is presented in Chapter 4; and as stated therein, a Linear Quadratic Regulator (LQR) will be used for the control of the reference model. This was due to the structure of the state matrix of the 4th order representation of the synchronous generator which has non controllable states.

This controller is aimed at improving the stability of the system modelled in section 5.2.3. First, the modelling and simulation of the reference model is introduced. Then, the adaptation law's MATLAB translation is given. Thereafter, the decentralized interarea oscillations damping control scheme based on MRAC is presented.

5.4.2   Reference Model

The optimal control problem that produced the feedback gain K_1 and feedforward gain K_2 in Figure 5.37 is solved as in the MATLAB script 5.4, considering the generator parameters in Table 5.1.
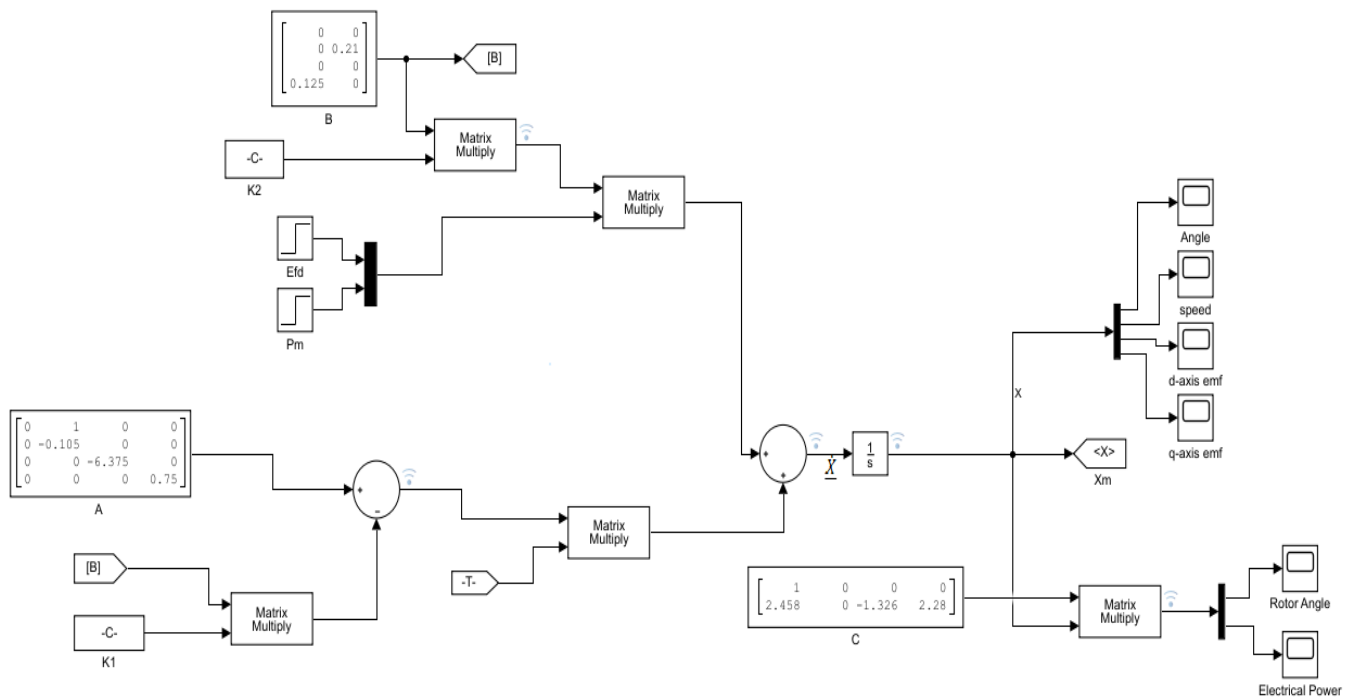
Figure 5.37: Reference model when $E_{FD} = 1$ and $P_m = 1$

MATLAB script 5.4

```
A =[0 0 0 0;1 0 0 0;0 0 -6.375 0;0 0 0 0.75];
```

```
B = [0 0;0 0.21;0 0;0.125 0];

C = [1 0 0 0;2.458 0 -1.326 2.28];

R = eye(2);

Q = 1e7 * C^T * C;              % very higher value of the weighting factor

[K_1,P,E] = lqr(A,B,C,Q,R);


% Introduce a feedforward gain that cancels any steady-state error

K_2 = -inv(C * (inv(A - B*K) * B)
```

A very higher value for the weighting factor is explained by the design objectives i.e., a faster response with a focus on the importance of the error rather than the control effort. Figures 5.38 and 5.39 show the reference model's output response with no initial conditions and $E_{FD} = 1, P_m = 1$ as set points (step inputs).
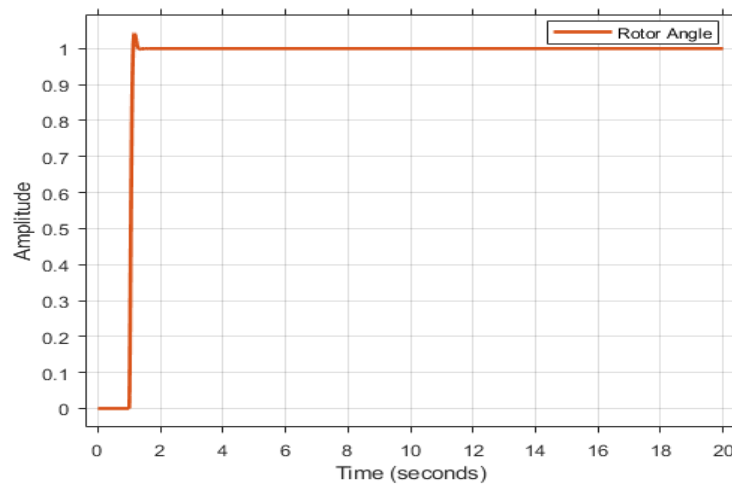
Figure 5.38: Reference model output1 when $E_{FD} = 1$ and $P_m = 1$
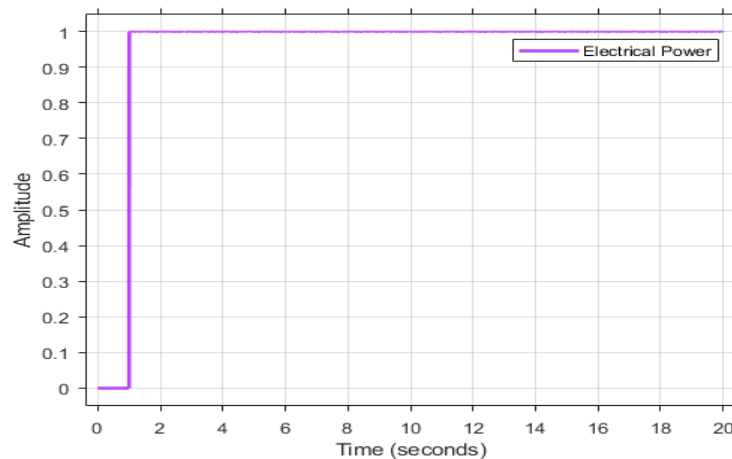
Figure 5.39: Reference model output2 when $E_{FD} = 1$ and $P_m = 1$

5.4.3 Adaptation Law

From Equation (4.68), the adaptation law is given by:

$$\dot{V} = -\Gamma\Phi(x)e^T PB$$

This translates to the MATLAB script below

MATLAB script 5.5

```
function phi = parameter_estimation (B, X, Error)
n = size(B);
phi = zeros(n);

P = [2.8966    2.1881    1.1965    2.1551;
     2.1881    1.9966    0.6827    1.8861;
     1.1965    0.6827    0.7590    0.5348;
     J];

phi = X * Error' * P * B;
end
```

5.4.4 Interarea Oscillations Damping Controller Scheme based on MRAC

5.4.4.1 System Modelling

The proposed control architecture as presented in Figure 5.40 comprises three (3) components:

- The reference model: modelled in the sub-section 5.4.2
- The parameter estimation: this block contains the adaptation law presented in sub-section 5.4.3
- The 4th order model of the synchronous generator: modelled in sub-section 5.2.3.
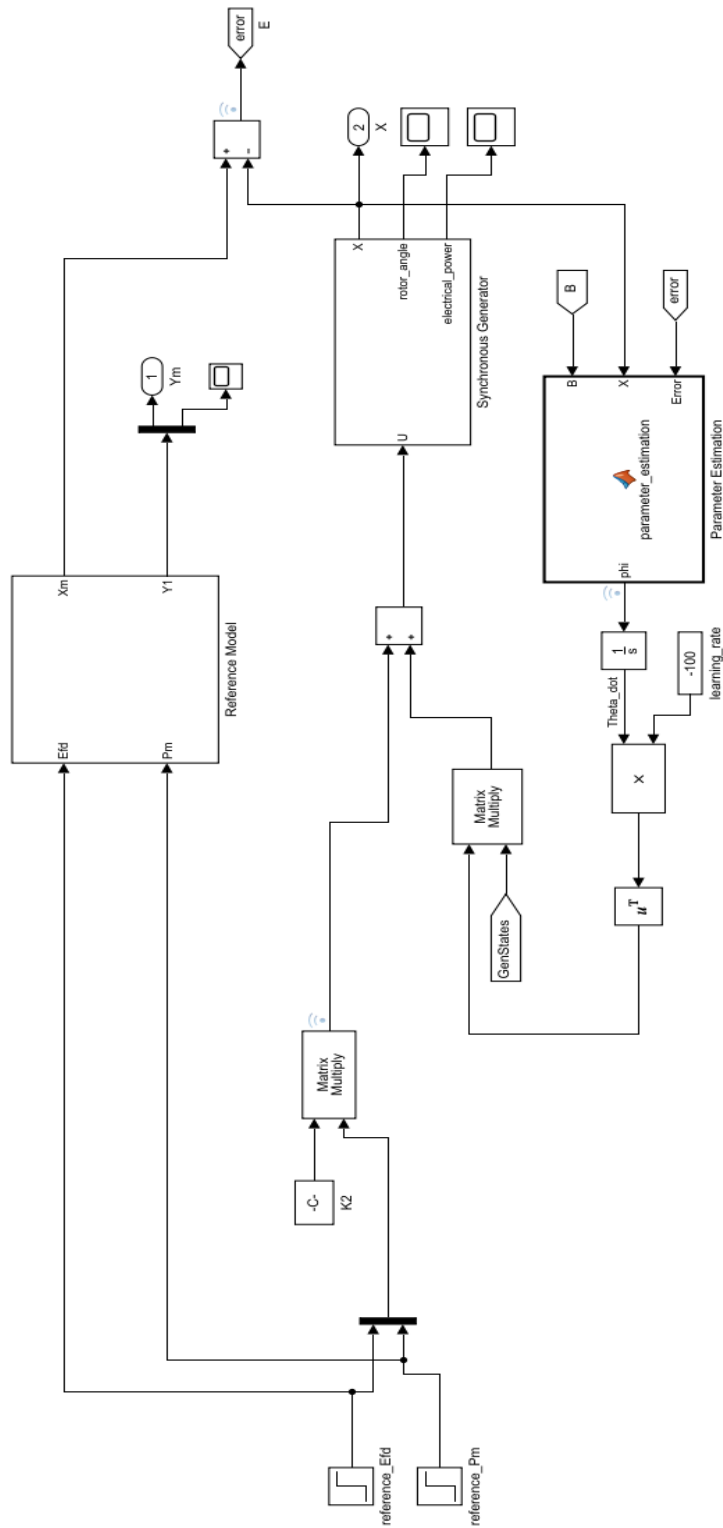
Figure 5.40: Proposed MRAC-based decentralized power system interarea oscillation controller scheme when $E_{FD} = 1$, $P_m = 1$ and learning rate =100

Since the system is of type *unmatched uncertainty* i.e., the control input matrix $B \in \mathbb{R}^2 x \, \mathbb{R}^4$ is a full-rank non-square wide matrix with $rank(B) = 2$, some transformations are needed to ensure that the control input cancels out the uncertainty by adaptive control.

The MATLAB script 5.2 can be modified as:

MATLAB script 5.6

```
function unmatched_uncertainty = Fct(gamma, ed_prime, eq_prime, X, B)

V = 1;
xd = 1.8;
xq = 1.7;
xd_prime = 0.3;
xq_prime = 0.55;
J = 4.774;
Td_prime = 8;
Tq_prime = 0.4;

elt1 = (1/J) * (V*cos(gamma)/xq_prime) * ed_prime;
elt2 = (V/J) * (sin(gamma)/xd_prime) * eq_prime;
% elt3 = ((V^2) / J) * ((1/xq_prime) - (1/xd_prime)) * sin(gamma) *
cos(gamma);
elt3 = ((V^2) / J) * ((1/xq) - (1/xd_prime)) * sin(gamma) *
cos(gamma);
row2 = elt1 - elt2 - elt3;
row3 = (V/Tq_prime) * ((xq-xd_prime) / xq_prime) * sin(gamma);
row4 = (V/Td_prime) * ((xd/xd_prime) - 1) * cos(gamma);
F = [0;row2;row3;row4];

pseudo_inverse = (B'*B)\ B';
if sum(isnan(pseudo_inverse(:)))
    unmatched_uncertainty = [0;0];
else
    unmatched_uncertainty = pseudo_inverse * F; % B' * ((B*B')^-1) * F
* X' * (X*X')^-1
end

end
```

5.4.4.2 Simulation Results

The results will be presented in form of cases studies starting from the steady state condition.

The performance of the proposed algorithm will be assessed by change of setpoints and the introduction of internal disturbances in the form of gaussian noise. The later to mimic low-frequency interarea oscillations caused by small variations in loads. With no initial conditions and the initial set-points being $E_{FD} = 1$, $P_m = 1$, the Gaussian noise characteristics are given in Figure 5.48.

Figure 5.41 illustrates the system used in case study 5 onwards.
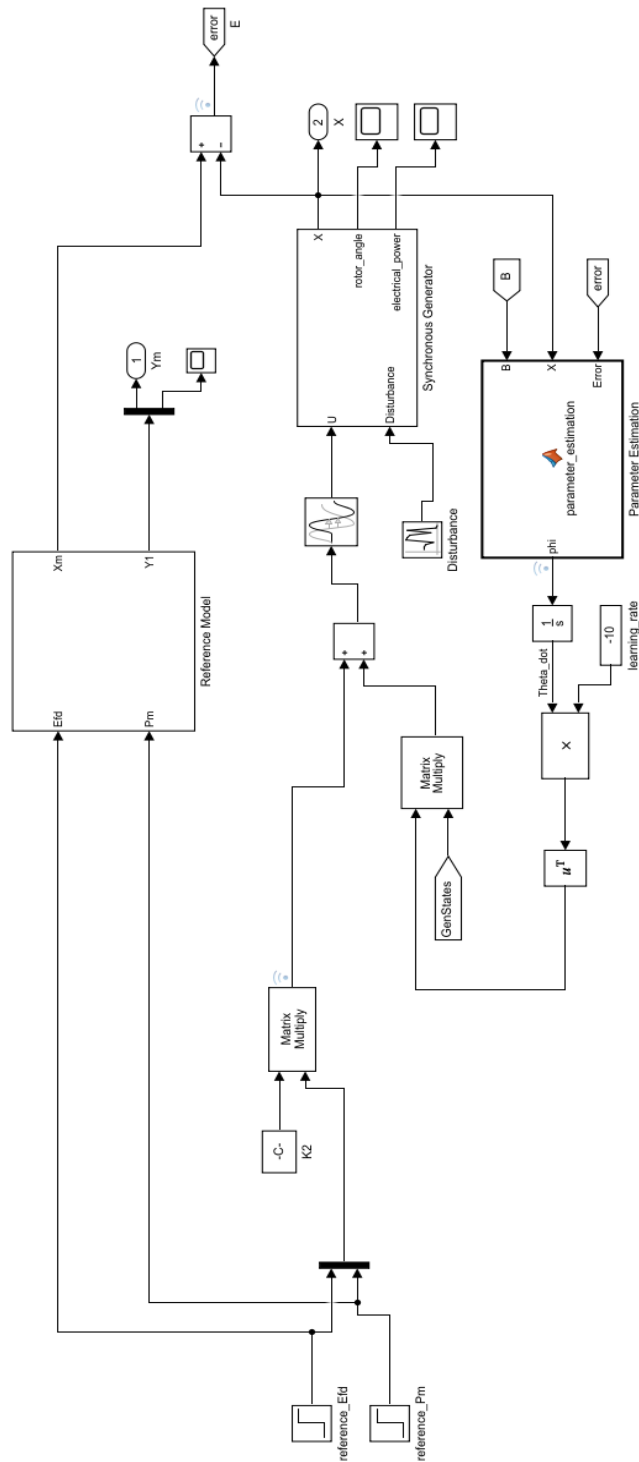
Figure 5.41: Proposed MRAC-based decentralized power system
interarea oscillation controller scheme with added internal disturbance.
Transport delay = 1/60 = maximum simulation step size

Case Study 1: Steady State with no initial conditions and set points $E_{FD} = 1, P_m = 1$
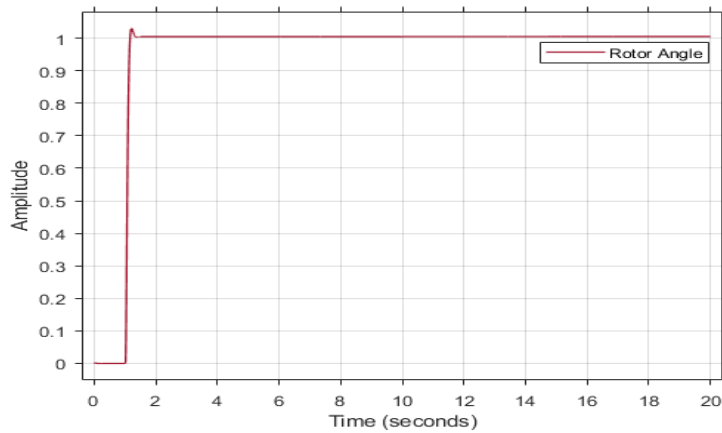
Figure 5.42: Synchronous generator rotor angle when $E_{FD} = 1, P_m = 1$ and learning rate = 10
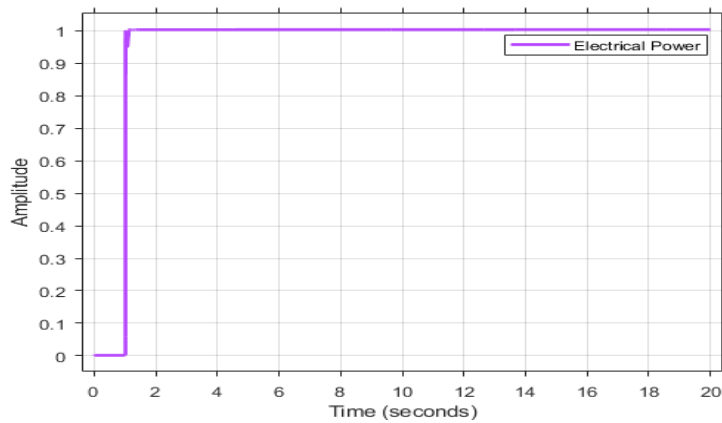
Figure 5.43: Synchronous generator electrical power when $E_{FD} = 1, P_m = 1$ and learning rate = 10

Case Study 2: Setpoint Change - $E_{FD} = 2.395$ and $P_m = 1$ and no initial conditions
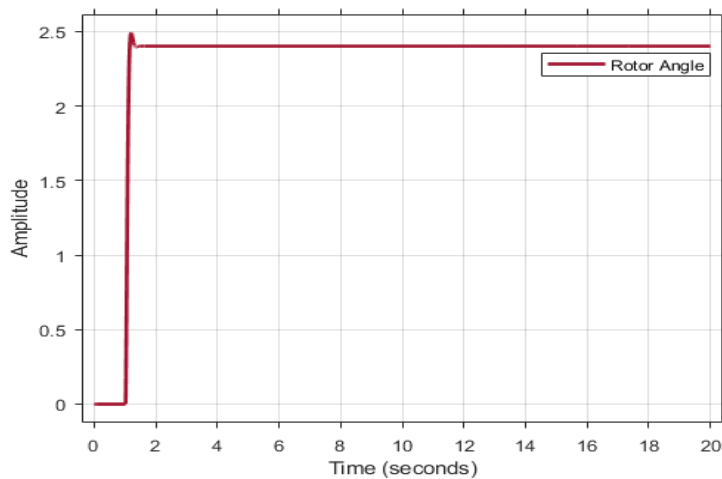
Figure 5.44: Synchronous generator rotor angle when $E_{FD} = 2.395, P_m = 1$ and learning rate = 10
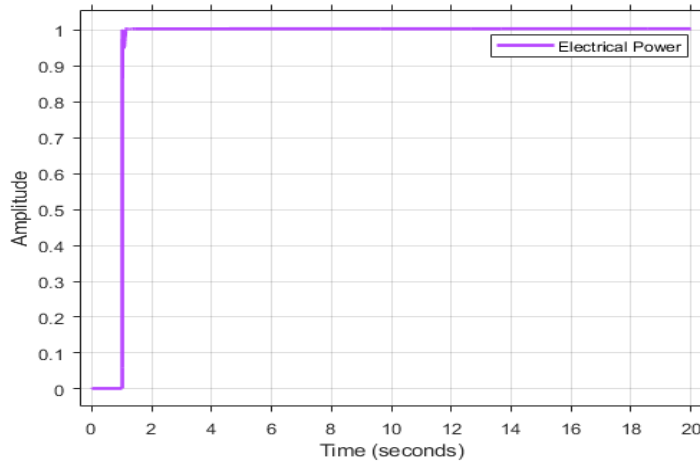
Figure 5.45: Synchronous generator electrical power when $E_{FD} = 2.395$, $P_m = 1$ and learning rate = 10

Case Study 3: Setpoint Change - $E_{FD} = 1$ and $P_m = 0.77778$, and no initial conditions
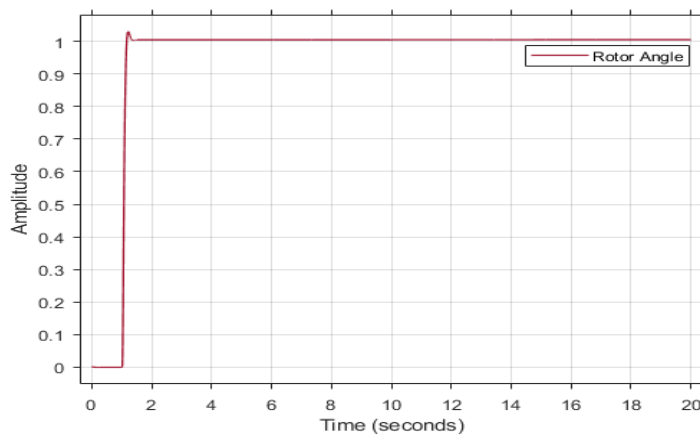
Figure 5.46: Synchronous generator rotor angle when $E_{FD} = 1$, $P_m = 0.77778$ and learning rate = 10
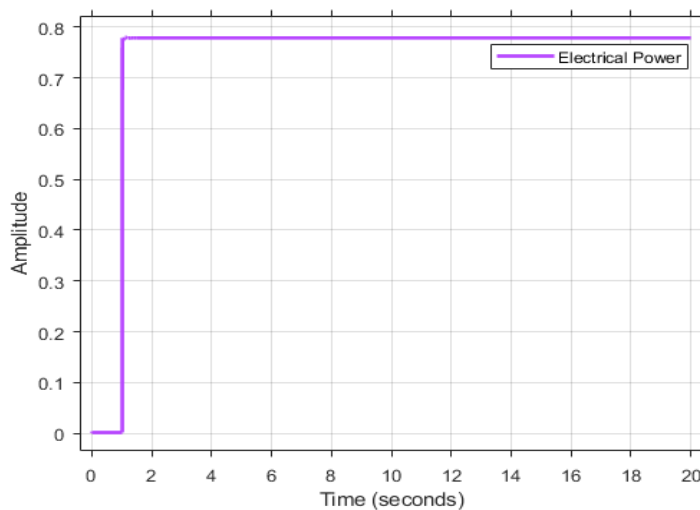
Figure 5.47: Synchronous generator electrical power when $E_{FD} = 1$, $P_m = 0.77778$ and learning rate = 10

Case Study 4: Setpoint Change - $E_{FD} = 2.395$ and $P_m = 0.77778$, and no initial conditions
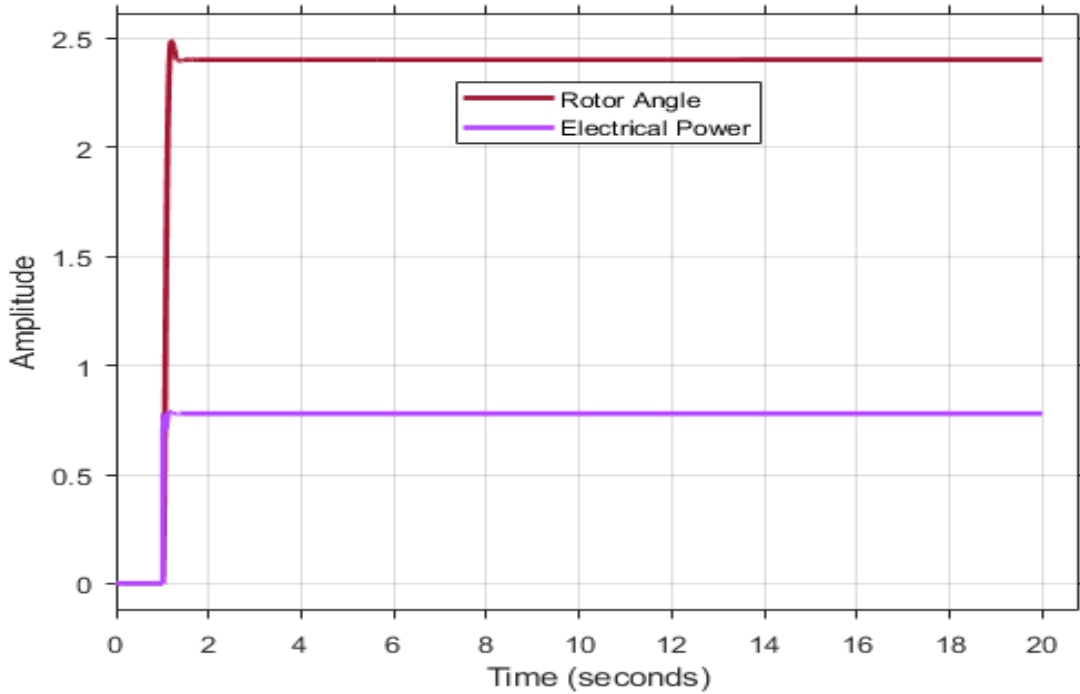
Figure 5.48: Synchronous generator rotor angle and electrical power when $E_{FD} = 2.395$, $P_m = 0.77778$ and learning rate = 10

Case Study 5: Added internal disturbance at t = 0s - $E_{FD} = 1$ and $P_m = 1$, and no initial conditions

A normally (Gaussian) distributed random signal with a variance of 1 and 0.001 sample time is added onto the system.

In all subsequent test cases, and whenever an internal or external noise is added onto the system, the signal with the characteristics illustrated in Figure 5.48 is the one utilized.

Furthermore, to better emulate inter-area oscillations which are inherent to a given power system, this added noise is added at the very beginning of the simulation.

While in this thesis external disturbances refers to noise added onto the input signal, internal disturbances pertain to those that makes the very system.
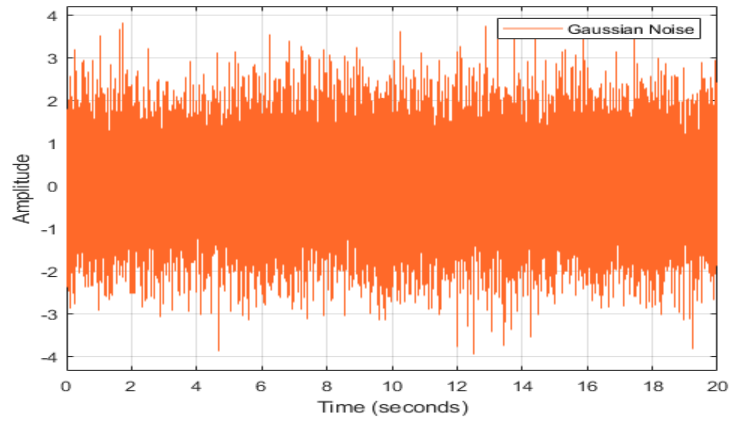
Figure 5.49: Gaussian noise characteristics
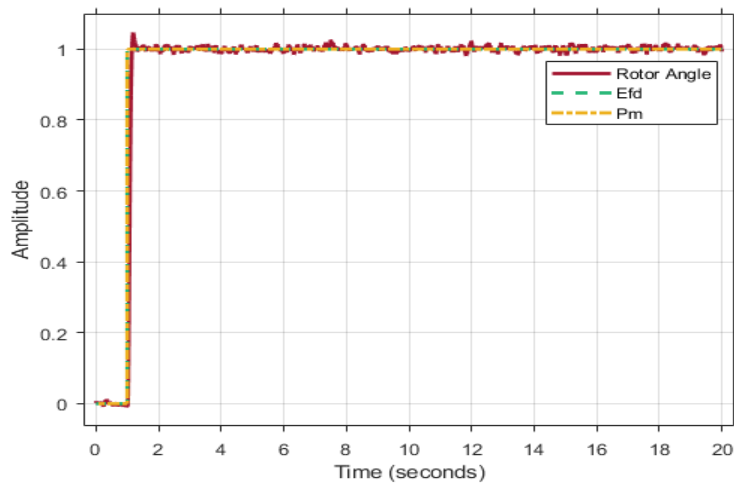
Figure 5.50: Synchronous generator rotor angle when $E_{FD} = 1$, $P_m = 1$, learning rate = 10, and added disturbance
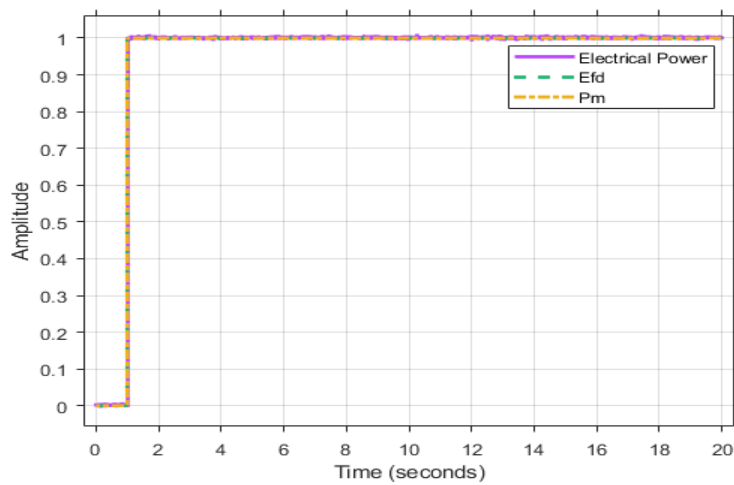
Figure 5.51: Synchronous generator electrical power when $E_{FD} = 1$, $P_m = 1$, learning rate = 100, and added disturbance
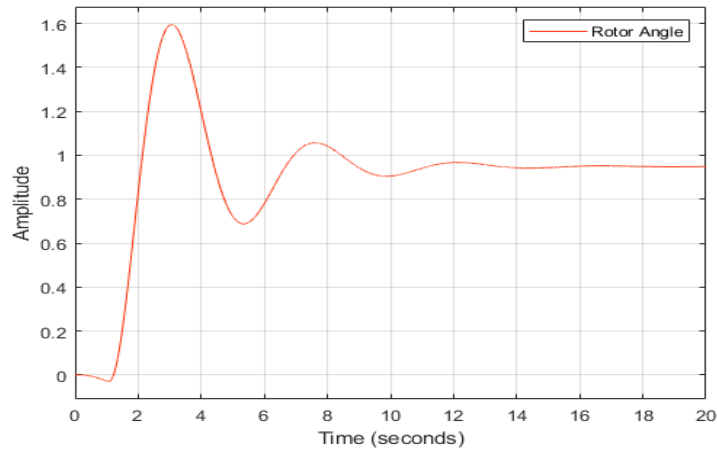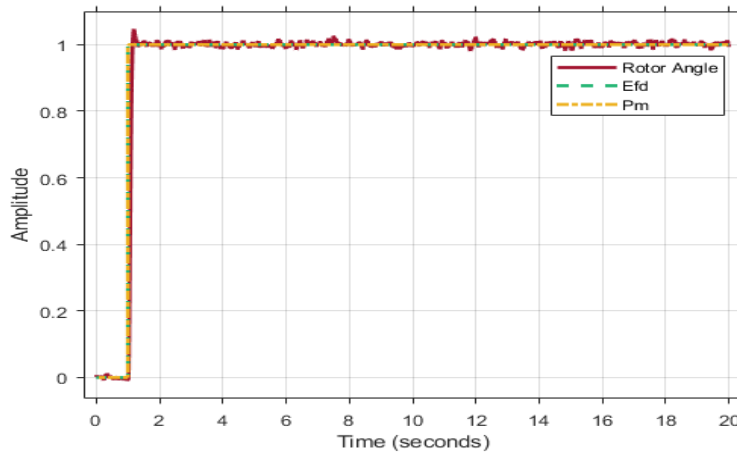
Figure 5.52: Synchronous generator rotor angle when $E_{FD} = 1$, $P_m = 0.77778$, learning rate = 10 and added disturbance
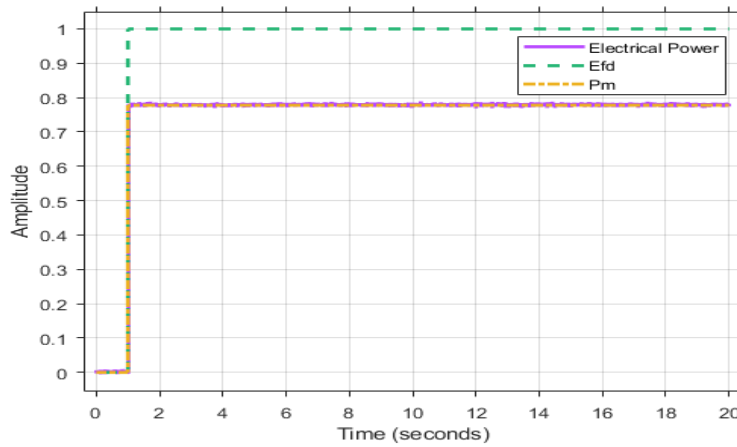
Figure 5.53: Synchronous generator electrical power when $E_{FD} = 1$, $P_m = 0.77778$, learning rate = 10 and added disturbance

Case Study 7: Added internal disturbance at t = 0s and setpoint change - $E_{FD} = 2.395$, $P_m = 1$
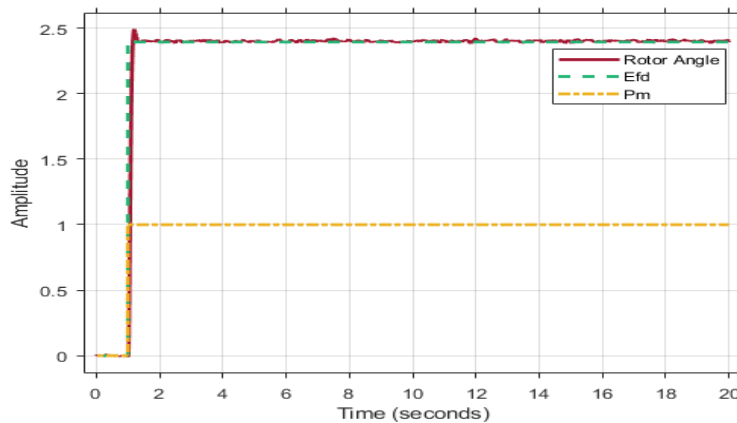
Figure 5.54: Synchronous generator rotor angle when $E_{FD} = 2.395$, $P_m = 1$, learning rate = 10, and added disturbance
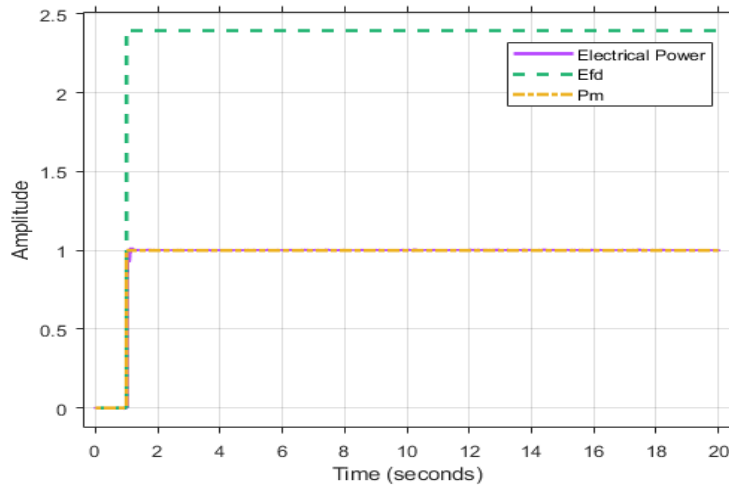
Figure 5.55: Synchronous generator electrical power when $E_{FD} = 2.395$, $P_m = 1$, learning rate = 10, and added disturbance

Case Study 8: Added internal disturbance at t = 0s and setpoint change -$E_{FD} = 2.395$, $P_m = 0.77778$
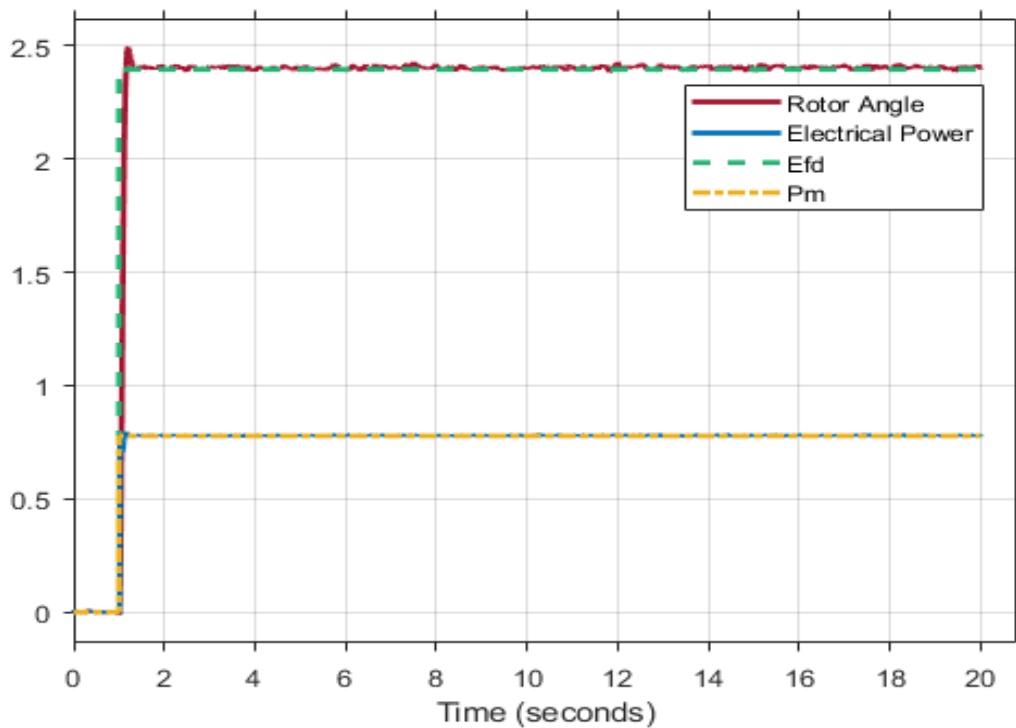
Figure 5.56: Synchronous generator rotor angle and electrical power when $E_{FD} = 2.395$, $P_m = 0.77778$, learning rate = 10, and added disturbance

Also considered are disturbances to the input signal. This is shown in Figure 5.57 and case studies 9-11, 15. For these cases, the learning rate is kept at 10 and there are no initial conditions. The characteristic of the noise is shown in Figure 5.48
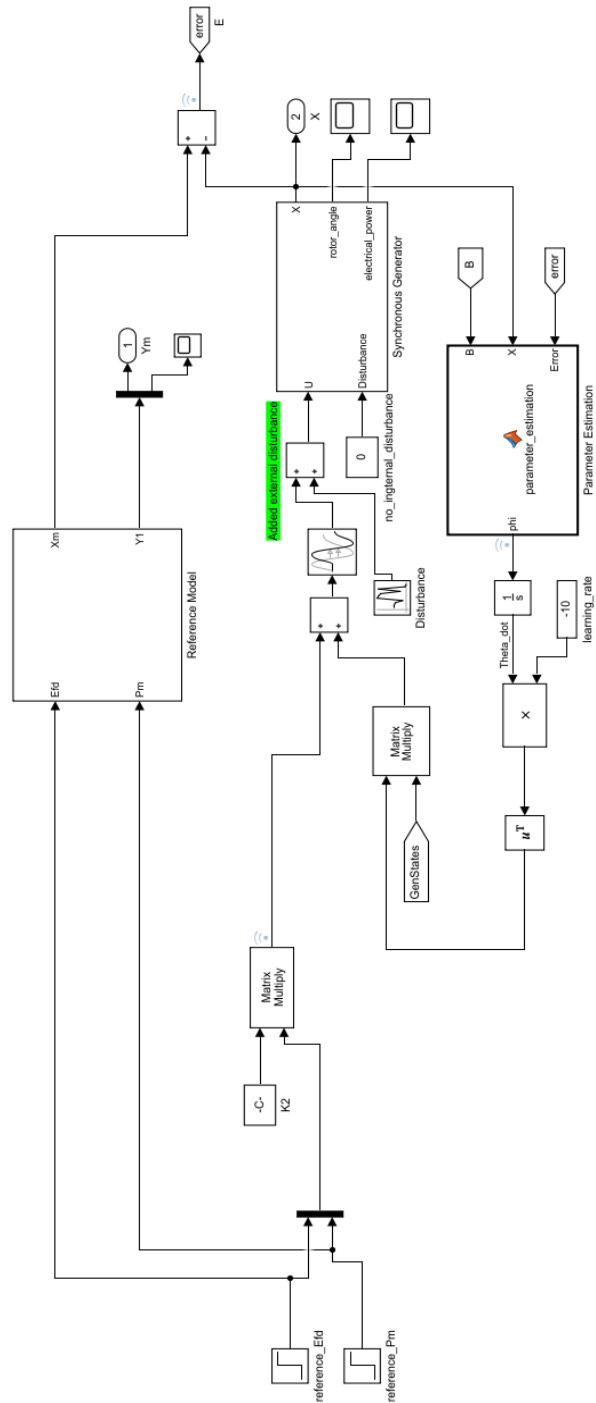
Figure 5.57: Proposed MRAC-based decentralized power system interarea oscillation controller scheme when $E_{FD} = 1, P_m = 1$, without internal disturbances but with external disturbances. Transport delay = 1/60 = maximum simulation step size

Case Study 9: Added disturbance to input signal at t = 0s and $E_{FD} = 1$, $P_m = 1$
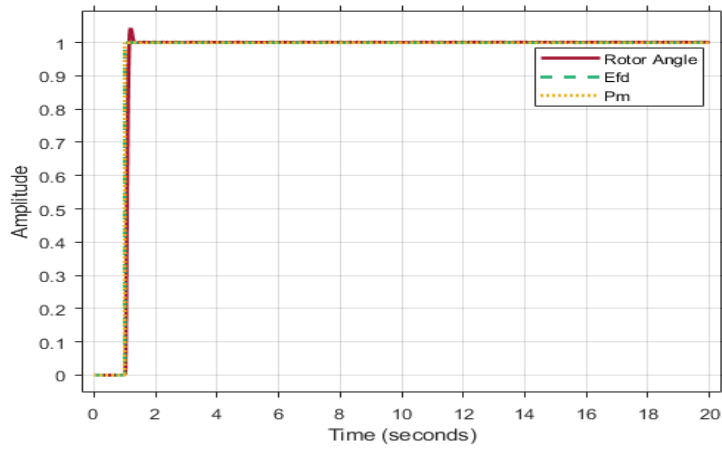
Figure 5.58: Synchronous generator rotor angle when $E_{FD} = 1$, $P_m = 1$, learning rate = 10, and added external disturbance
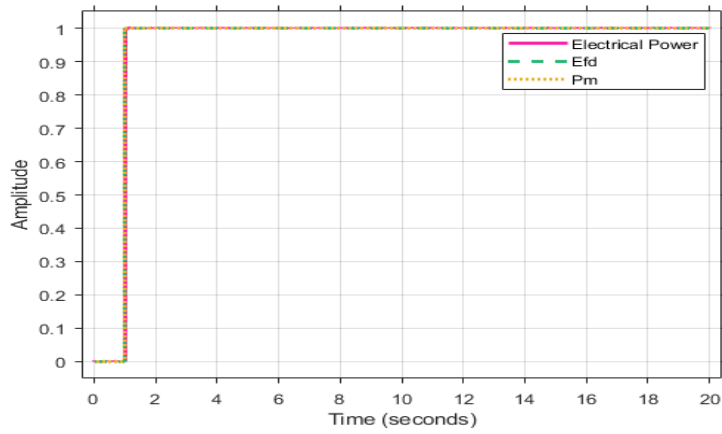
Figure 5.59: Synchronous generator electrical power when $E_{FD} = 1$, $P_m = 1$, learning rate = 10, and added external disturbance

Case Study 10: Added external disturbance at t = 0s and setpoint change - $E_{FD} = 2.395$, $P_m = 1$
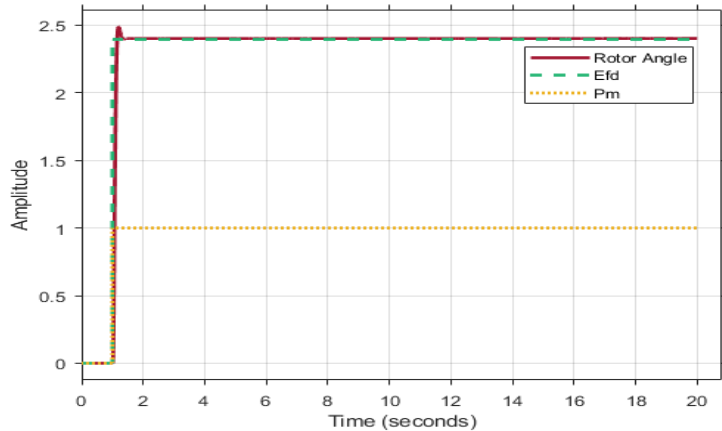
Figure 5.60: Synchronous generator rotor angle when $E_{FD} = 2.395$, $P_m = 1$, learning rate = 10, and added external disturbance
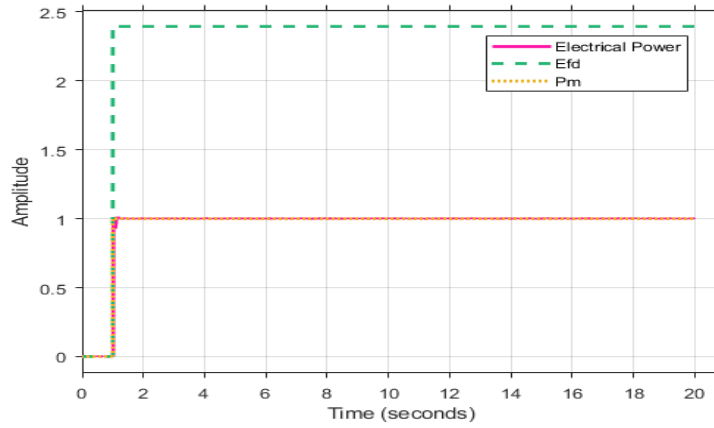
Figure 5.61: Synchronous generator electrical power when $E_{FD} = 2.395$, $P_m = 1$, learning rate = 10, and added external disturbance

Case Study 11: Added external disturbance at t = 0s and setpoint change - $E_{FD} = 1$, $P_m = 0.77778$
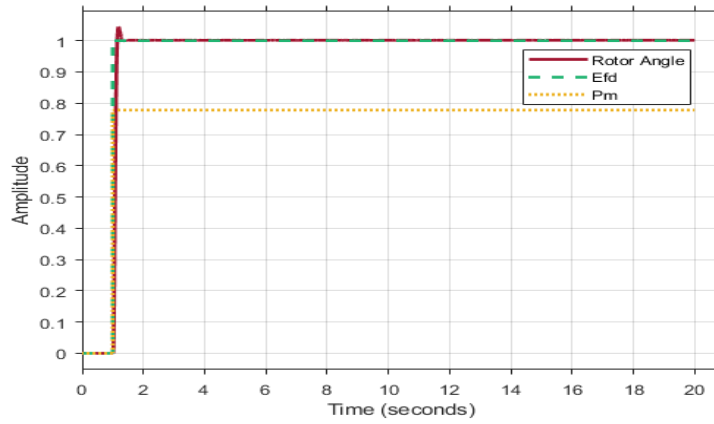
Figure 5.62: Synchronous generator rotor angle when $E_{FD} = 1$, $P_m = 0.77778$, learning rate = 10, and added external disturbance
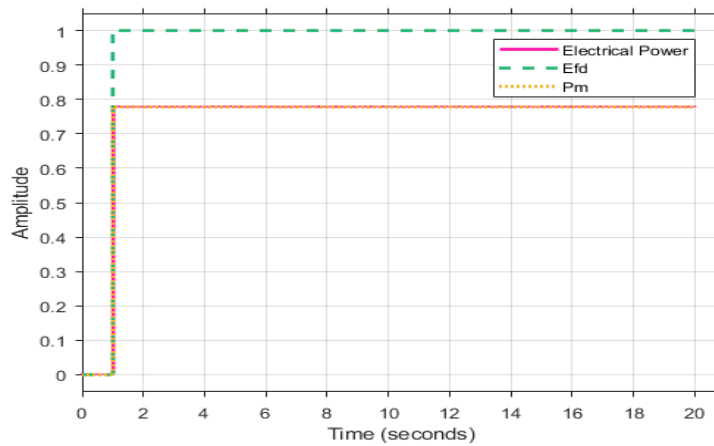
Figure 5.63: Synchronous generator electrical power when $E_{FD} = 1$, $P_m = 0.77778$, learning rate = 10, and added external disturbance

Figure 5.64 shows a modified version of the proposed decentralized control architecture that incorporates both internal and external disturbances at the very beginning of the simulation.

Figure 5.64: Proposed MRAC-based decentralized power system interarea oscillation controller scheme when $E_{FD} = 1, P_m = 1$, with both internal and external disturbances.
Transport delay = 1/60 = maximum simulation step size
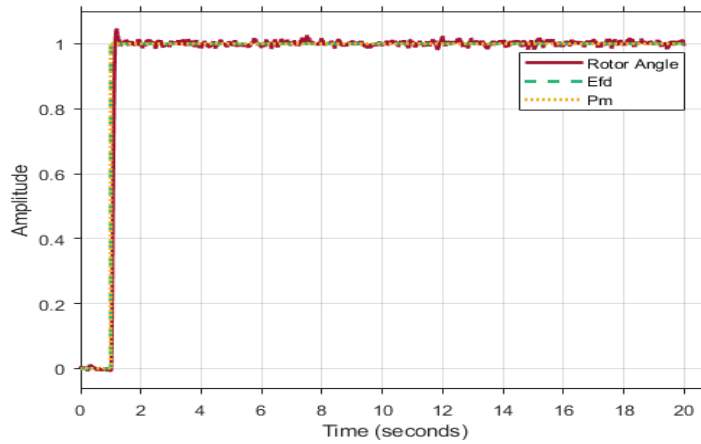
Figure 5.65: Synchronous generator rotor angle when $E_{FD} = 1$, $P_m = 1$, learning rate = 10, and added external and internal disturbances
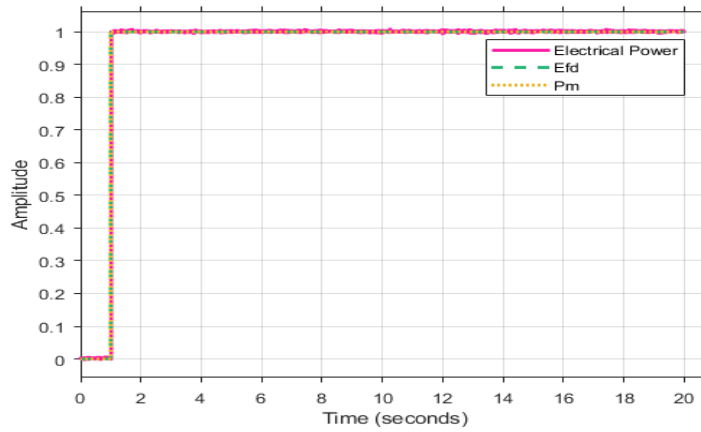
Figure 5.66: Synchronous generator electrical power when $E_{FD} = 1$, $P_m = 1$, learning rate = 10, and added external and internal disturbances

Case Study 13: Added external and internal disturbances at t = 0s - $E_{FD} = 2.395$ and $P_m = 1$
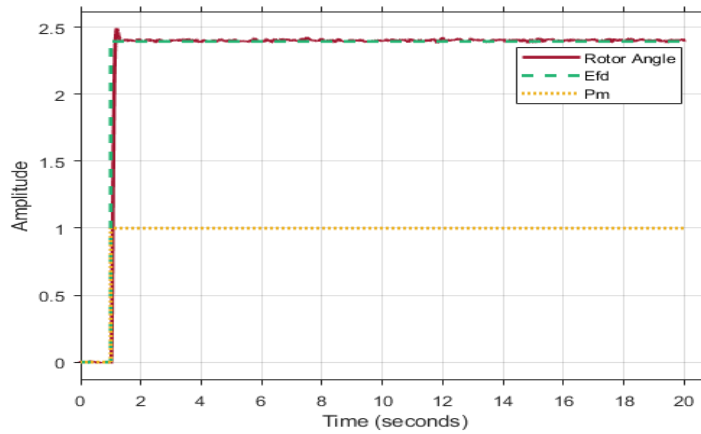
Figure 5.67: Synchronous generator rotor angle when $E_{FD} = 2.395$, $P_m = 1$, learning rate = 10, and added external and internal disturbances
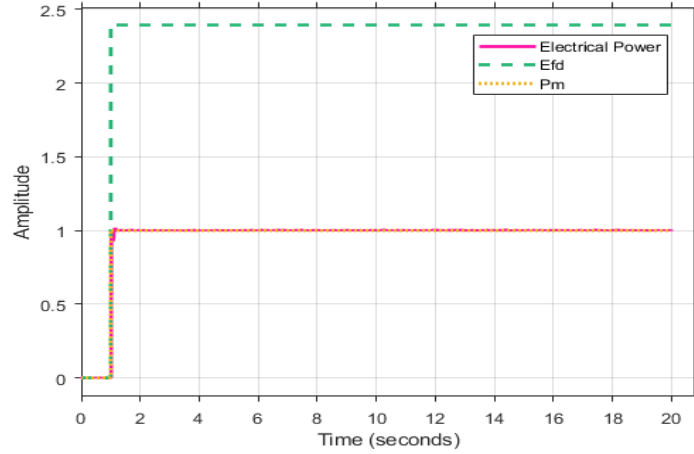
Figure 5.68: Synchronous generator electrical power when $E_{FD} = 2.395$, $P_m = 1$, learning rate = 10, and added external and internal disturbances

Case Study 14: Added external and internal disturbances at t = 0s - $E_{FD} = 1$ and $P_m = 0.77778$
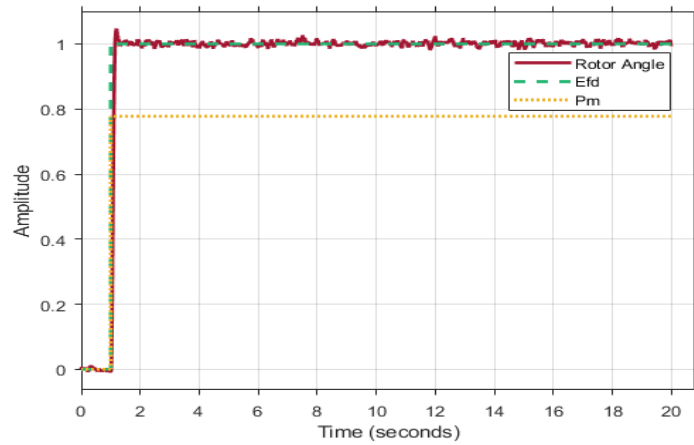
Figure 5.69: Synchronous generator rotor angle when $E_{FD} = 1$, $P_m = 1$, learning rate = 10, and added external and internal disturbances
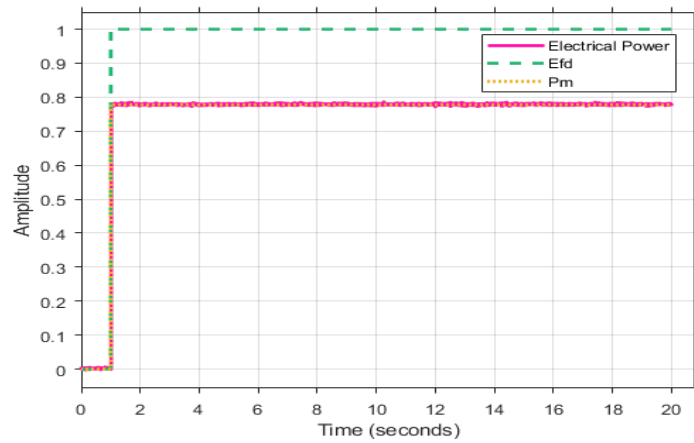
Figure 5.70: Synchronous generator electrical power when $E_{FD} = 1$, $P_m = 1$, learning rate = 10, and added external and internal disturbances

Case Study 15: Added external disturbance at t = 0s and setpoint change $E_{FD} = 2.395\ P_m = 0.77778$
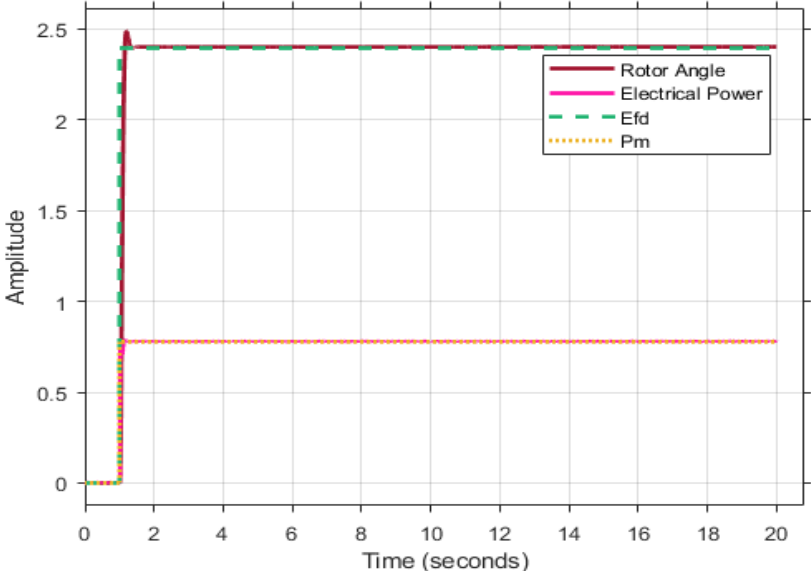
Figure 5.71: Synchronous generator rotor angle and generator electrical power when $E_{FD} = 2.395, P_m = 0.77778$, learning rate = 10, and added external disturbance

Case Study 16: Added external and internal disturbances at t = 0s and - $E_{FD} = 2.395, P_m = 0.77778$
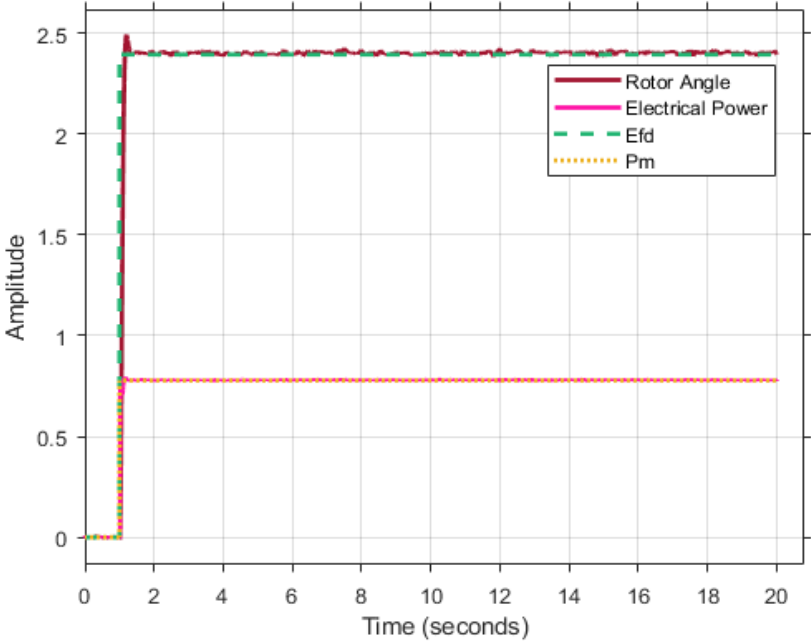
Figure 5.72: Synchronous generator rotor angle and electrical power when $E_{FD} = 2.395, P_m = 0.77778$, learning rate = 10, and added external and internal disturbances

Other than cases that matches best the type of disturbances linked to LFEOS highlighted until now, the robustness of the proposed algorithm was also tested by modifying the structure in Figure 5.48 to include a fault in the system through an impulse. This is illustrated in Figure 5.73, 5.74, and 5.75 while the system's response is shown in case study 17.
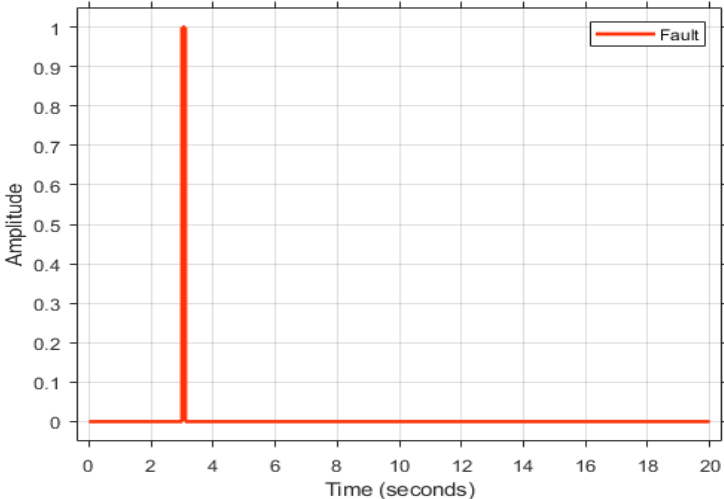
Figure 5.75: Fault 1 – 100ms duration at t=3 s
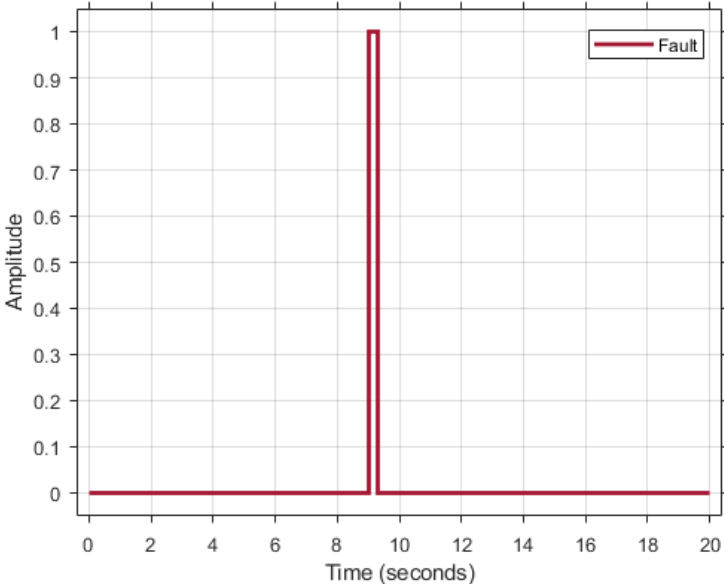
Figure 5.76: Fault 2 – 300ms duration at t=9 s
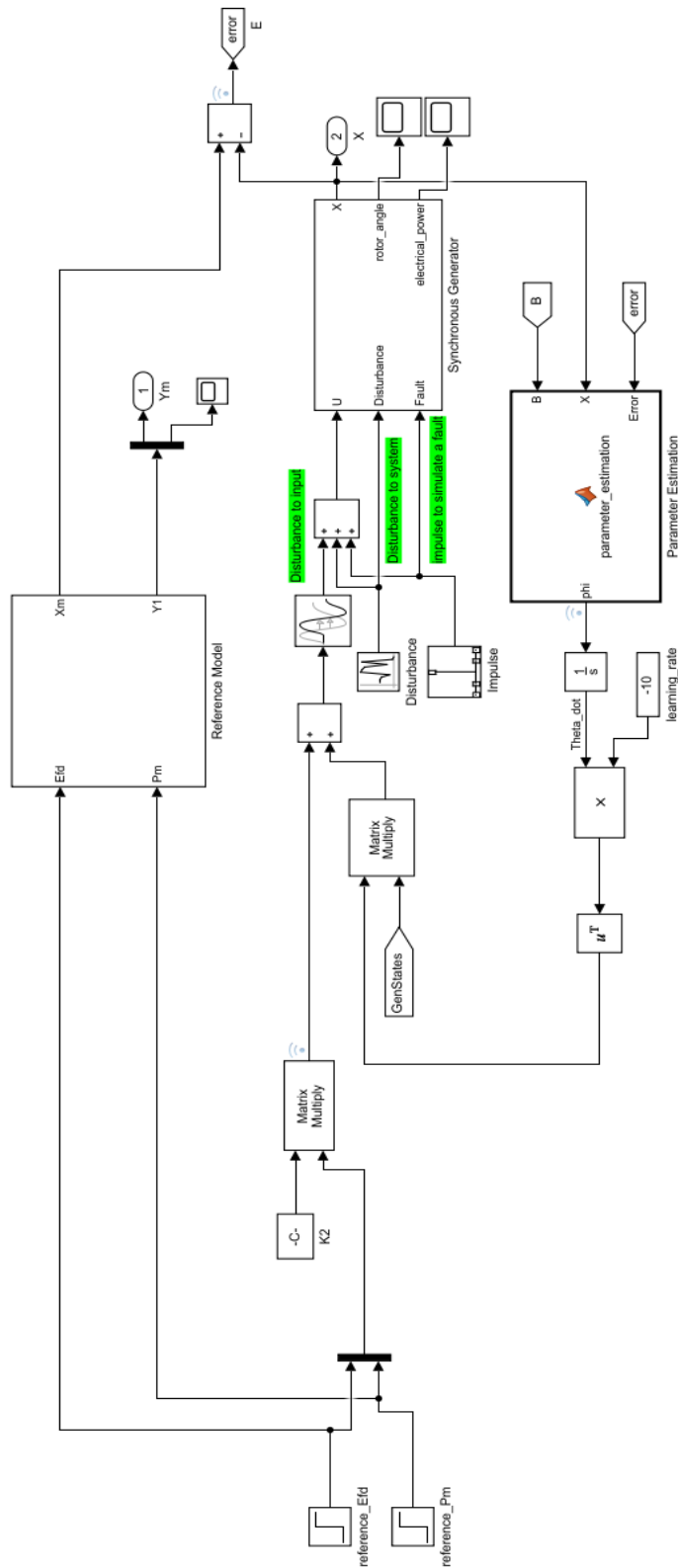
Figure 5.75: Proposed MRAC-based decentralized power system interarea oscillation controller scheme when $E_{FD} = 2.395, P_m = 0.77778$, and added disturbances as well as a fault. Transport delay = 1/60 = maximum simulation step size

Case Study 17: **Added disturbances at t = 0s and fault in system as well as setpoint change.**

This case study includes both internal disturbances as well as disturbances added onto the generator input signals. While the disturbances characteristics are shown in Figure 5.49, two faults are introduced at the third (3s) and the nineth second (9s) for a duration of 100ms and 300ms respectively. These are illustrated in Figures 5.76 and 5.77.
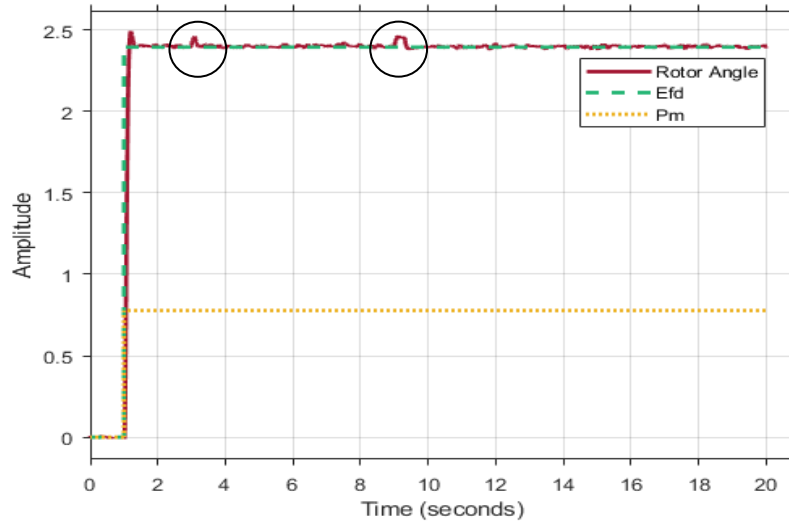
Figure 5.76: Synchronous generator rotor angle when $E_{FD} = 2.395$, $P_m = 0.77778$, learning rate = 10, and added external and internal disturbances as well as a fault
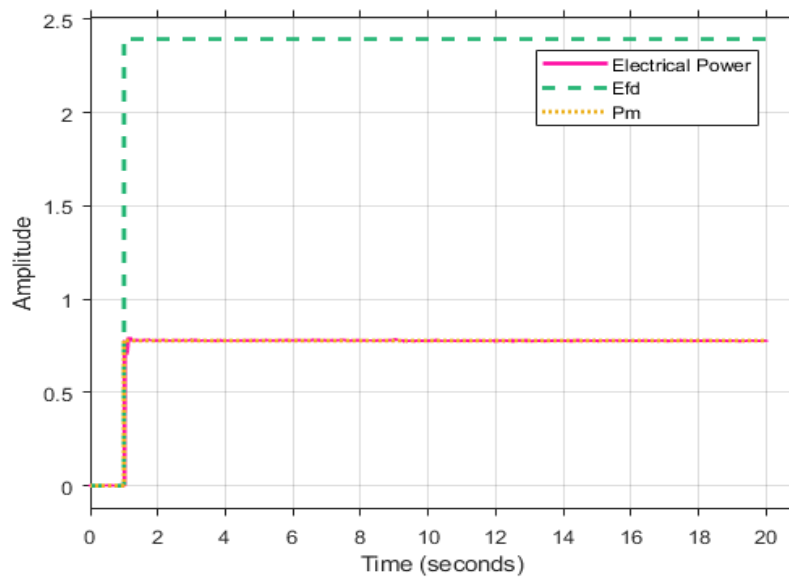
Figure 5.77: Synchronous generator electrical power when $E_{FD} = 2.395$, $P_m = 0.77778$, learning rate = 10, and added external and internal disturbances as well as a fault

Detailed summary of the various case studies is given in Table 5.2 below.

Table 5.2: Simulation test cases for the proposed MRAC

	Setpoints		Disturbances			Rotor angle characteristics				
	E_{FD}	P_m	Internal	External	Fault	Rise time (ms)	Slew rate (/s)	Overshoot (%)	Steady-state error (%)	Recovery time (ms)
Case 1	1	1	0	0	0	83.308	9.524	4.737	N/A	N/A
Case 2	2.395	1	0	0	0	88.852	21.528	3.646	N/A	N/A
Case 3	1	0.77778	0	0	0	83.314	9.525	4.737	N/A	N/A
Case 4	2.395	0.77778	0	0	0	88.857	21.529	3.646	N/A	N/A
Case 5	1	1	Gaussian noise	0	0	86.434	9.301	4.737	N/A	N/A
Case 6	2.395	1	Gaussian noise	0	0	90.534	21.243	3.646	N/A	N/A
Case 7	1	0.77778	Gaussian noise	0	0	90.534	21.243	3.646	N/A	N/A
Case 8	2.395	0.77778	Gaussian noise	0	0	90.541	21.243	3.646	N/A	N/A
Case 9	1	1	0	Gaussian noise	0	83.311	9.524	4.737	N/A	N/A
Case 10	2.395	1	0	Gaussian noise	0	88.854	21.528	3.646	N/A	N/A
Case 11	1	0.77778	0	Gaussian noise	0	83.317	9.524	4.737	N/A	N/A
Case 12	1	1	Gaussian noise	Gaussian noise	0	86.439	9.801	4.737	N/A	N/A
Case 13	2.395	0.77778	0	Gaussian noise	0	88.854	21.528	3.646	N/A	N/A
Case 14	2.395	1	Gaussian noise	Gaussian noise	0	90.537	21.243	3.646	N/A	N/A
Case 15	1	0.77778	Gaussian noise	Gaussian noise	0	83.317	9.524	4.737	N/A	N/A
Case 16	2.395	0.77778	Gaussian noise	Gaussian noise	0	90.544	21.243	3.646	N/A	N/A
Case 17	2.395	0.77778	Gaussian noise	Gaussian noise	Yes	90.544	21.243	3.646	N/A	~ 54

5.4.4.3 Discussions

The Synchronous generator having two inputs; both are varied first independently and then together while assessing the performance of the proposed algorithm. Furthermore, in order to emulate small variations in loads that are mainly responsible for interarea oscillations, internal disturbances were added. Also added were external disturbances that did not affect the system performance as illustrated in Figures 5.48-5.64.

Irrespective of the disturbances introduced, the rotor angle kept a considerably stable response, with an overshoot of less than 5 %, a rise time less than 100 ms, non-existent steady-state error and an impressive recovery time of 54 ms. This recovery time was achieved when the generator was subjected to disturbances in form of noise as well as a 300 ms fault at the nineth second.

The performance of the proposed decentralized architecture is compared with that of the Lyapunov-based nonlinear controller presented in subsection 5.3.

Unlike the latter, the MRAC produced good results and can be deployed efficiently as part of decentralized control architecture in an aim to enhance the (rotor) angle stability thus mitigating LFEOs in power systems. A comparison of the performance of both controllers is shown in Figure 5.78a, 5.78b, 5.79a, and 5.79b.
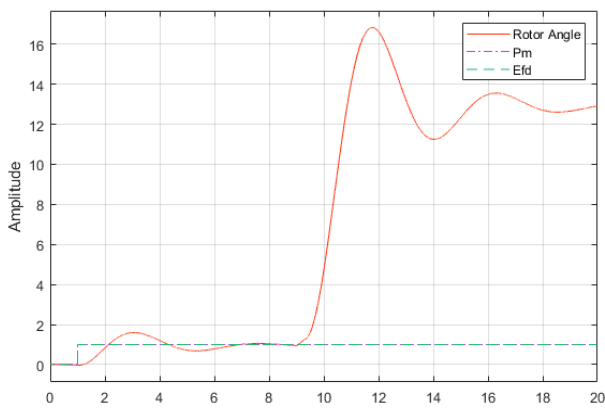
Figure 5.78a: Synchronous generator rotor angle when $E_{FD} = 1$, $P_m = 1$, $\lambda = 4.5$. A 300ms fault is introduced at t=9s while the Lyapunov-based controller is applied to the system
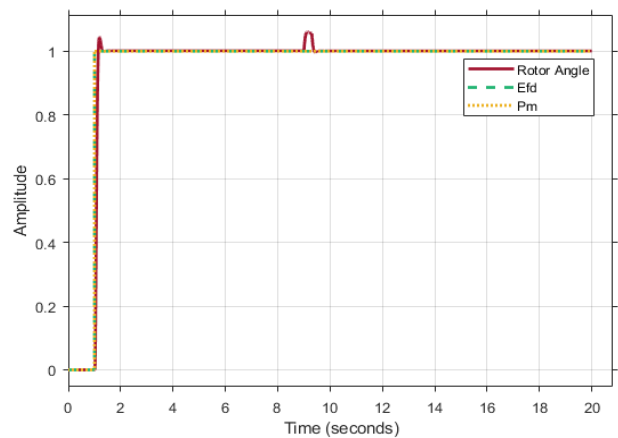
Figure 5.78b: Synchronous generator rotor angle when $E_{FD} = 1$, $P_m = 1$, learning rate =10. A 300ms fault is introduced at t=9s while the proposed MRAC controller is applied to the system
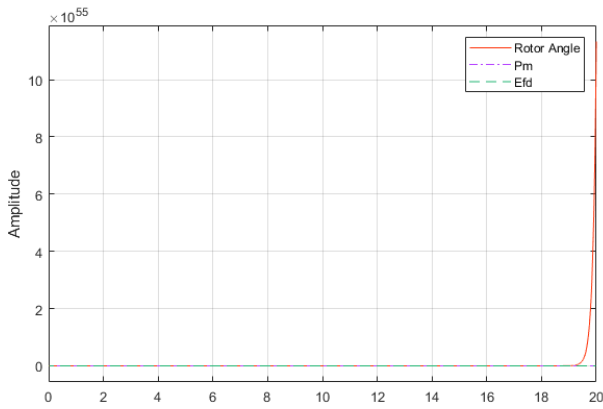
Figure 5.79a: Synchronous generator rotor angle when $E_{FD} = 1$, $P_m = 1$, $\lambda = 4.5$. With the Lyapunov-based controller applied to the system, two contingencies namely a 300ms fault introduced at t=9s as well as a normally (Gaussian) distributed random signal with a variance of 1 and a sample time of 0.001 are added.
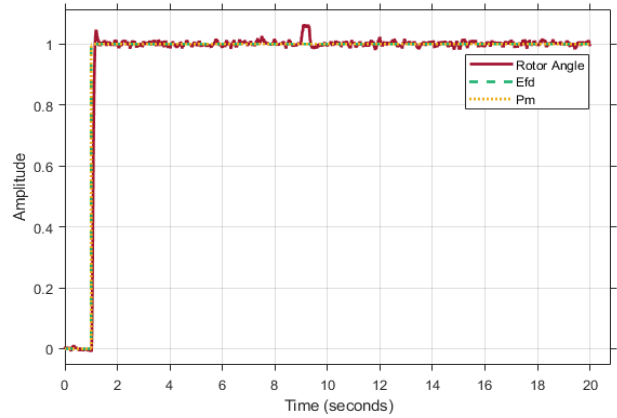
Figure 5.79b: Synchronous generator rotor angle when $E_{FD} = 1$, $P_m = 1$, learning rate = 10. With the proposed MRAC applied to the system, two contingencies namely a 300ms fault introduced at t=9s as well as a normally (Gaussian) distributed random signal with a variance of 1 and a sample time of 0.001 are added.

The results as presented in this sub-section are satisfactory as the generator, through its rotor angle remained stable irrespective of the contingencies. Furthermore, considering the time of interest 3-5 seconds for standard power systems and 10 seconds for large ones with weak interconnections, the proposed novel scheme is very robust.

5.5 Conclusions

The performance of the Lyapunov reference-model based nonlinear controller together with the MRAC are presented in this chapter.

While the first showed a poor performance, the results with the later were satisfactory for small signal stability improvement.

Chapter 6 presents the testbench for its real-time implementation.

CHAPTER SIX

DESIGN AND IMPLEMENTATION OF A LAB-SCALE TESTBED FOR THE POWER SYSTEM LOW-FREQUENCY ELECTROMECHANICAL OSCILLATIONS (LFEOs) DAMPING SYSTEM

6.1 Introduction

The design of a control mechanism requires some knowledge of the system the control ought to be applied onto, hence the importance of the knowledge of its various inputs and outputs. As defined by Burns (2001), a system is a collection of matters, parts, components, or procedures which are included within some specified boundary. The way in which its outputs (system response) respond in changes to its inputs and other disturbances is therefore critical.

An electrical power system can be seen as an aggregate of generating units together with their respective controls. This was illustrated in Chapter 3 where the dynamics of a given generator were shown to be highly nonlinear and required some control.

The generator (rotor) angle instability being responsible for Low-Frequency Electromechanical Oscillations (LFEOs) in power systems, ensuring that each generator's rotor angle remains stable before and after system disturbance is of an uttermost importance. Therefore, this thesis proposes the use of Model-Reference Adaptive Controllers (MRACs) to ensure the small signal stability is preserved.

The proposed control scheme for small signal stability enhancement is validated using a novel decentralized MRAC scheme testbed.

In this thesis, the word "system" would refer to the Single Machine Infinite Bus (SMIB) system which is used in the design of the proposed decentralized interarea oscillations damping algorithm in Chapter 4 as well as the modelling and simulations presented thereafter in Chapter 5.

This chapter is structured as follows: Section 6.2 introduces the various devices used in the development of the testbed for the Hardware-in-the-Loop (HIL). Section 6.3 gives an overview of the existing communication protocols used in power system oscillations damping monitoring and damping studies, with the focus on the IEC 61850 Standard. Section 6.4 describes the implementation of the proposed MRAC-based power system interarea oscillations damping algorithm and Section 6.5 concludes the chapter.

6.2   Description of the Testbed Components

The proposed HIL testbed is illustrated in Figure 6.1, and relevant equipment and software used in the implementation of the proposed power system oscillations MRAC are listed in Table 6.1 and described in the following subsections.

Table 6.1: Hardware and software used in the proposed MRAC power system low-frequency electromechanical oscillations damping

Equipment/Software	Configuration Software
Real-Time Digital Simulator® (RTDS)	RSCAD V5.005.1
GTNET-PMU Card	Telnet
SEL-3355 computer	N/A
SEL Real-Time Automation Controller (RTAC)	AcSELerator Architect
SEL-2725 Ethernet Switch	N/A
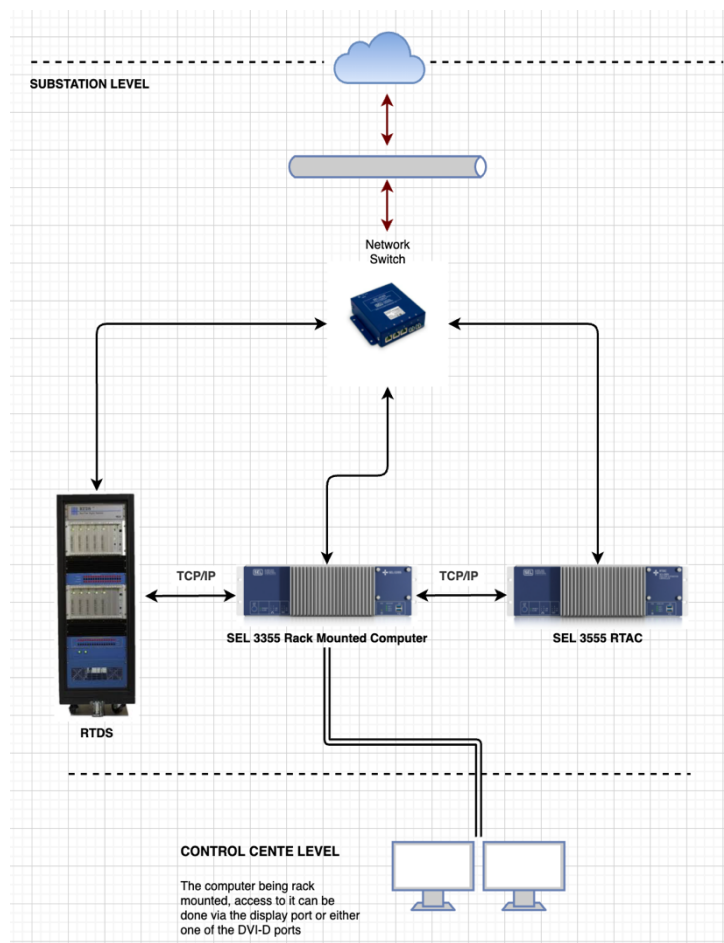
Figure 6.1: Overview of the proposed MRAC power systems interarea oscillations damping testbed.

148

While the network switch is utilized to connect each one of these devices to the substation network, sub-sections 6.2.1-6.2.3 provide a comprehensive summary around the features and usefulness of each one of the devices in Figure 6.1 in the implementation of the MRAC based interarea oscillations damping algorithm. Furthermore, a throughout explanation on how they communicate and interact in the completed testbed is presented therein.

6.2.1   The Real-Time Digital Simulator (RTDS) And RSCAD Software
6.2.1.1 Real-Time Digital Simulator

The Real-Time Digital Simulator$^®$ (RTDS) as its name implies is a power system simulator that makes use of advanced parallel processing techniques with its built-in 50 μs time-step to achieve the computation speeds to maintain real-time operation (RTDS manual, 2014). From smart grid and distributed generation studies to closed loop testing of power system controllers, the RTDS$^®$ has found applications in many fields of power systems. A detailed summary of these applications is listed in Table 6.2 (RTDS hardware manual, 2014).

Table 6.2: Real-time Digital Simulator Applications (RTDS hardware manual, 2014)

RTDS$^®$ applications
Closed loop testing of power system controllers
Communication protocols for substation automation using IEC 61850, IEC 60870-5-104, and IEEE C37.118 standards respectively
Hardware-in-the-loop (HIL) testing of protection relays
High speed power system simulations
Power system stability studies
Power Hardware-In-the-Loop (PHIL) testing of renewable energy devices, electric vehicles, motors, and loads
Smart grid and distributed generation studies

Amongst the available cards in the RTDS$^®$ racks at the Centre for Substation Automation and Energy Management Systems (CSAEMS) are the processor cards (PB5 and GPC cards) used in solving the equations representing the components modelled in the RSCAD software and the network solution of the modelled power system. Also available are the GTNET and GTSYNC cards. The first is responsible for providing various network protocols to the communication network whereas the second ensures that the RTDS$^®$ simulator time-step for Phasor Measurement Unit (PMU) and Sampled Value (SV) applications is locked

to an external accurate time reference such as the Global Positioning System (GPS). More on these cards can be found in Adewole (2016).

6.2.1.2 RSCAD Software

The RSCAD software suite is the user's main interface to the Real-Time Digital Simulator (RTDS) hardware and has three main modules: The Draft, the Transmission Line (T-Line), and the Runtime environments (RTDS manuals, 2014).

The Draft module allows for the designing of the power system models and entering of the parameter settings. The T-Line is used in the modelling of transmission lines while the Runtime module contains the user interface with controls and outputs linked to the model created in the Draft module.

Through the Runtime module, users can control actions such as starting and stopping the simulation, initiate disturbance, online monitoring of system quantities, trigger data acquisition system, etc. (RTDS manuals, 2014).

One requirement to establish communication between the RSCAD software and the RTDS is that the workstation whereby it is installed and the RTDS be in the same network (RTDS manuals, 2014).

Upon successful completion of the software installation, two directories, RSCAD and RTDS_USER, are created. The first contains the installation files while the second serves as the user project's directory.

This is illustrated in Figure 6.2.
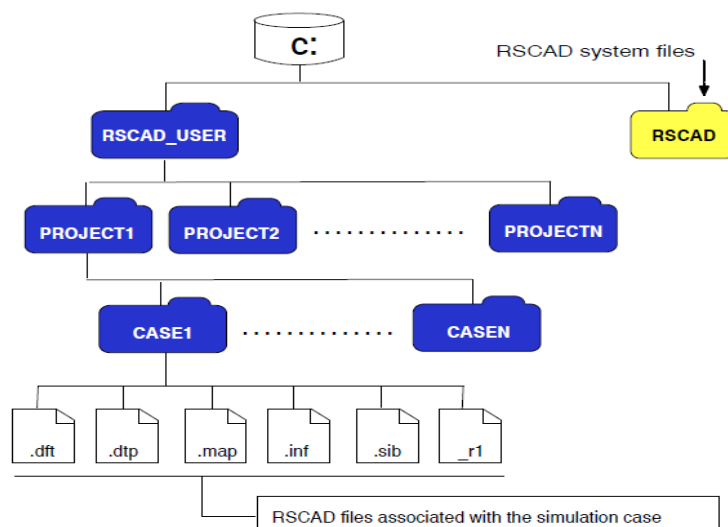
Figure 6.2: RSCAD system and user files directories (RSCAD manuals, March 2020)

6.2.2 The SEL-3355 Rack-Mounted Computer

The Schweitzer Engineering Laboratories (SEL)-3355 hardened industrial substation computer is used for modelling and computation. Built-in for industrial automation, this device comes with a Xeon quad-core 2.8 GHz processor. It can be used as a reliable computing platform, supporting soft programmable logic controller but also as an embedded device supporting object-oriented programming languages. This computer is also very useful for HMI visualization, monitoring, and control while meeting industry standards (NERC CIP, …). This is achieved by leveraging its Centre for Internet Security (CIS) Benchmarks settings.

A detailed summary of its applications is listed in Table 6.3.
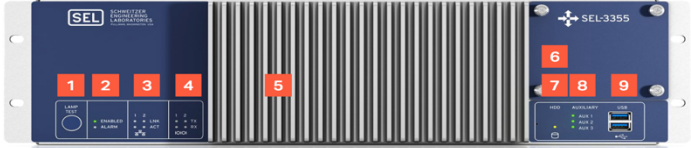
Table 6.3: **Summary of the Schweitzer Engineering Laboratories-3355 rack-mount computer applications (Schweitzer Engineering Laboratories, 2021)**

Application	Use case
Industrial Automation Platform	Deployment of a complete automation control and operator station functionality in a single package directly to a given plant without being concerned about possible harsh environment conditions
HMI Visualization, Monitoring, and Control	Secure system visualization, monitoring and control point for substations or plants
Information Processor: Data Concentrator/Protocol Converter	Collection of legacy Remote Terminal Units (RTUs) to be sent to SCADA through legacy protocols. These data could then be serviced to any given client via any supported protocol.
Centralized Authentication Server	Role-based access control can be set up. Furthermore, high-availability services can be setup for always-on-service.
Original Equipment Manufacturer (OEM) Software Appliance	Software deployment as a dedicated appliance in harsh environmental conditions.
Synchrophasor Archiver, Server, and Console	With its built-in RAID capability that provides fast and reliable database storage, the concentration, archiving, and display of synchrophasor data from a wide area with a SEL-5073 SYNCHROWAVE software is made more efficient.
Engineering Workstation	Other than being fully compliant with the IEEE 1613 and protective relay specifications for harsh environments, remote access is possible via Microsoft Windows Remote Desktop tunnelled over IPsec or through Intel Active Management Technology (AMT) with remote KVM over IP.
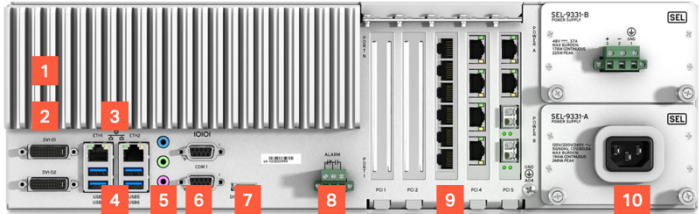
Tables 6.4 and 6.5 present the features of a typical rackless SEL-3355 computer.

Table 6.4: **SEL-3355 Features (front)**

Description	
1. LED lamp test button	
2. Alarm and operational LED indicators	
3. Ethernet port link status and network activity LEDs	
4. Serial port transmits and receives LEDs	
5. Front heat sink and no fans or moving parts	**Figure 6.3**: **Schweitzer Engineering Laboratories-3355 computer - front (Schweitzer Engineering Laboratories, 2021)**
6. Up to four hot-swappable SSDs	
7. Hard disk drive activity LED	
8. Three programmable bicolour LEDs	
9. Two front-panel USB 3.1 ports	

Table 6.5: **SEL-3355 Features (back)**

Description	
1. Rear heat sink	
2. Two DVI-D ports	
3. Two high-speed Gigabit Ethernet ports	
4. Four USB 3.1 ports	
5. Line-in, line-out, and microphone jacks	
6. Two built-in BIOS-configurable EIA-232 ports with +5V on Pin 1	
7. DisplayPort monitor technology	**Figure 6.4**: **Schweitzer Engineering Laboratories-3355 computer – back (Schweitzer Engineering Laboratories, 2021)**
8. From C alarm contact output	
9. Up to five expansion slots: one legacy PCI, two x1 PCIe, and two x4 PCIe	
10. Dual hot-swappable power supplies	

6.2.3 The SEL-3555 Real-Time Automation Controller (RTAC)

Combining the best features of the x86-64 architectures, embedded microcomputers, embedded real-time operating systems, as well as secure communications framework, the SEL-3555 RTAC is reputed to be one of the best automation platforms. This motivates its usefulness in the current research considering the criticality of the $50\,\mu s$ computation timestep of the RTDS. The latter being utilised for the modelling and real-time simulation of the Single-Machine Infinite Bus (SMIB) power system model.

Another important characteristic of this device is its security features. It can be configured such that central authentication is enforced via Lightweight Directory Access Protocol (LDAP). In addition, there is a possibility to assign individual and role-based account authentication.

Table 6.6 presents a summary of key features of this automation controller, Figure 6.5 its functional diagram, and Table 6.7-6.8 its features.

Table 6.6: **Summary of the Schweitzer Engineering Laboratories-3555 Real-time automation controller applications (Schweitzer Engineering Laboratories, 2021)**

Application	Use case
Integrate Power Management with Industrial Control	Can be utilized simultaneously as a protocol gateway, Remote Terminal Unit (RTU), logic processor, and system-wide data logger (or viewer). Withal, it can be used to exchange critical data for real-time monitoring, process control, and power system integration. Lastly, by means of either a single serial port, external modem, or high-speed network connection, one may access various station IEDs.
Data Concentration and Protocol Conversion	Can be deployed as a data concentrator either through modern or legacy protocols. This can be achieved through data points mapping and scaling as well as IED data normalisation into common data types, timestamp formats, and time zones. Moreover, high-speed control schemes between the RTAC, RTDS and/or relays with the GOOSE peer-to-peer (P2P) messaging is also possible.
Distribution Automation or Microgrid Controller	Considering the secure, self-healing network capabilities of the SEL ICON® Integrated Communications Optical Network and accurate time distribution to IEDs, all aspects of a microgrid can be monitored. Furthermore, it can integrate synchrophasor messages from relays or PMUs in control logic or SCADA usage.
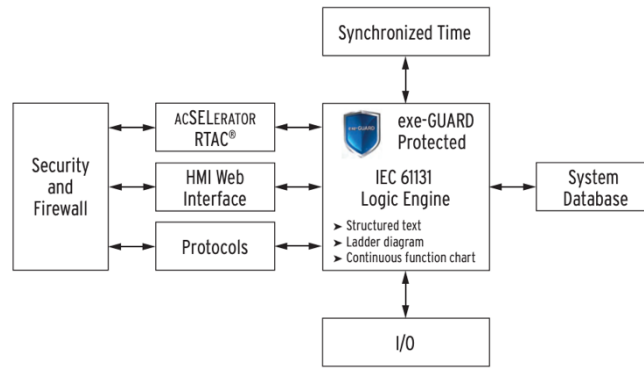
Figure 6.5: Schweitzer Engineering Laboratories-3355 computer – back (Schweitzer Engineering Laboratories, 2021)

Table 6.7: SEL-3555 Features (front)

Description	
1. LED lamp test button	
2. Ethernet activity LEDs	
3. Serial port activity LEDs	
4. SSD activity LED	
5. Programmable bicolour LEDs	**Figure 6.6**: Schweitzer Engineering Laboratories-3555 RTAC - front (Schweitzer Engineering Laboratories, 2021)
6. 2 front-panel USB 3.1 ports	
7. SSD with storage capacity up to 480 GB	

Table 6.8: SEL-3555 Features (back)

Description	
1. DVI-D ports	
2. 2 high-speed Gigabit Ethernet ports	
3. 4 USB 3.1 ports	
4. 2 built-in EIA-232 ports	
5. DisplayPort	**Figure 6.7**: Schweitzer Engineering Laboratories-3555 RTAC - back (Schweitzer Engineering Laboratories 2021)
6. Form C alarm outputs	

7. PCIe expansion slots	
8. 6 EIA-232/422/485 ports	
9. Dual hot-swappable power supplies	

6.3 Communication Protocols used in Power System Oscillations Monitoring, Protection And Control

6.3.1 Synchrophasor for Power Systems

6.3.1.1 Overview

Phasors are used in many protections and data acquisition functions. Synchrophasors are phasors' value estimated from voltage or current waveforms referenced to a common time base (IEEE Std C37.118, 2011). Simultaneous measurement sets derived from synchronized Phasors provide a vastly improved method for tracking power system dynamic phenomena for improved power system monitoring, protection, operation, and control (IEEE Std C37.118, 2011). Thus the need for a standard that would ensure that consistency is taken into consideration in the design of phasor measurement devices and protocols used in data collection.

Since both the voltage and current are *alternating current*, the phasors values are estimated using an interval over which the samples are taken and used to make the estimate (IEC/IEEE 60255-118-1-2018). Changes in the waveform parameters during the estimation window are ineluctable therefore, Synchrophasors represent the average value for the sinusoid during that interval. The choice of the appropriate filtering is very critical depending on the application (e.g., a longer interval reduces interferences but average out more dynamics) and following this, two classes are defined for specific applications (IEC/IEEE 60255-118-1-2018):

- M-class - more accurate reporting but can have delays (*for Measurement)*
- P-class – minimal delay without filtering (*for Protection)*

The IEEE standard C37.118 defines synchronized phasor measurements used in power system applications by providing methods to quantify the measurement, tests to be sure the measurement conforms to the definition, and error limits for the test (IEEE Std C37.118, 2005). Furthermore, it also defines a data communication protocol including message formats that allows various parts of a given real-time system communicating using the very data (IEEE Std C37.118, 2005).

As from its 2011 revision, this standard is divided into two parts: IEEE Std C37.118.1™-2011 for synchrophasor measurement requirements inclusing dynamic performance and the IEEE Std C37.118.2™-2011 for the data transfer requirements.

6.3.1.2 Synchrophasor Protocols

The introduction of concept of synchronized phasors within the power system took place in the 1980s; this was standardized for the first time in the IEEE 1344 standard (Martin, 2011). This was further developed into the widely used IEEE C37.118 in 2005. Then, it was further improved to include the definition of measurements in dynamic changes in the power system in its 2011 revision. Martin (2011) presented an exhaustive summary on the evolution of these protocols starting from the IEEE 1344 to the harmonization of the IEEE C37.118 -2011 with the IEC 61850 standard.

Though a simple structure of a synchrophasor network consists of the Phasor Measurement Unit (PMU) and the Phasor Data Concentrator (PDC), typically, many PMUs located at various key substations gather data and send it in real time to a PDC at the utility location where the data is aggregated (IEEE Std C37.118.2™, 2011). The collected data by various PDCs may be used to support many applications, ranging from visualization of information and alarms for awareness, to ones that provide sophisticated analytical, control, or protection functionality (IEEE Std C37.118.2™, 2011).

6.3.2   The IEC 61850 Standard

The early 1900s saw the rise in need to standardize the transfer of information between various devices within power system networks.

Though conducted somehow independently, work conducted in North America and Europe on substation communication bus defined the specifications of the UCA 2.0 substation bus peer-to-peer messaging service also referred to as Generic Object-Oriented Substation Events (GOOSE).

The proposed protocol was widely accepted by various manufacturers, and this led to further development to produce the current IEC 61850 standard. Amongst its benefits are the interchangeability or the ability to replace a given device with another from a different manufacturer without altering the entire network, interoperability, future proofing, etc. (IEC TR 61850-1, 2013)

The various parts that constitute the standard as in 2020 are illustrated in Table 6.9.

Table 6.9: Detailed Summary of the IEC 61850 Standard parts

General Description	Part Description
IEC61850-1: Communication networks and systems for power utility automation	Part 1: Introduction and overview
IEC61850-2: Communication networks and systems in substations	Part 2: Glossary
IEC61850-3: Communication networks and systems for power utility automation	Part 3: General requirements
IEC61850-4: Communication networks and systems for power utility automation	Part 4: System and project management
IEC61850-5: Communication networks and systems for power utility automation	Part 5: Communication requirements for functions and device models
IEC61850-6: Communication networks and systems for power utility automation	Part 6: Configuration description language for communication in power utility automation systems related to IEDs
IEC61850-7-1: Communication networks and systems for power utility automation	Part 7-1: Basic communication structure - Principles and models
IEC61850-7-2: Communication networks and systems for power utility automation	Part 7-2: Basic communication structure - Abstract communication service interface (ACSI)
IEC61850-7-3: Communication networks and systems for power utility automation	Part 7-3: Basic communication structure - Common data classes
IEC61850-7-4: Communication networks and systems for power utility automation	Part 7-4: Basic communication structure - Compatible logical node classes and data object classes
IEC61850-7-410: Communication networks and systems for power utility automation	Part 7-410: Basic communication structure - Hydroelectric power plants - Communication for monitoring and control
IEC61850-7-420: Communication networks and systems for power utility automation	Part 7-420: Basic communication structure - Distributed energy resources logical nodes
IEC61850-7-500: Communication networks and systems for power utility automation	Part 7-500: Basic communication structure - Use of logical nodes for modelling application functions and related concepts and guidelines for substations
IEC61850-7-510: Communication networks and systems for power utility automation	Part 7-510: Basic communication structure - Hydroelectric power plants - Modelling concepts and guidelines
IEC61850-7-6: Communication networks and systems for power utility automation	Part 7-6: Guideline for definition of Basic Application Profiles (BAPs) using IEC61850

Table 6.9: Detailed Summary of the IEC 61850 Standard parts cont'd

General Description	Part Description
IEC61850-7-7: Communication networks and systems for power utility automation	Part 7-7: Machine-processable format of IEC 61850-related data models for tools
IEC61850-8-1: Communication networks and systems for power utility automation	Part 8-1: Specific communication service mapping (SCSM) - Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3
IEC61850-8-2: Communication networks and systems for power utility automation	Part 8-2: Specific communication service mapping (SCSM) - Mapping to extensible Messaging Presence Protocol (XMPP)
IEC61850-9-1: Communication networks and systems in substations	Part 9-1: Specific communication service mapping (SCSM) - Sampled values over serial unidirectional multidrop point link
IEC61850-9-2: Communication networks and systems for power utility automation	Part 9-2: Specific communication service mapping (SCSM) - Sampled values over ISO/IEC 8802-3
IEC61850-9-3: Communication networks and systems for power utility automation	Part 9-3: Precision time protocol profile for power utility automation
IEC61850-80-1: Communication networks and systems for power utility automation	Part 80-1: Guideline to exchanging information from a CDC-based data model Using IEC 60870-5-101 or IEC 60870-5-104
IEC61850-80-2: Communication networks and systems for power utility automation	Part 8-2: Specific communication service mapping (SCSM) - Mapping to Extensible Messaging Presence Protocol (XMPP)
IEC61850-80-3: Communication networks and systems for power utility automation	Part 80-3: Mapping to web protocols - Requirements and technical choices
IEC61850-80-4: Communication networks and systems for power utility automation	Part 80-4: Translation from COSEM object model (IEC 62056) to the IEC 61850 data model
IEC61850-90-1: Communication networks and systems for power utility automation	Part 90-1: Use of IEC61850 for the communication between substations
IEC61850-90-2: Communication networks and systems for power utility automation	Part 90-2: Use of IEC61850 for the communication between substations and control centres
IEC61850-90-3: Communication networks and systems for power utility automation	Part 90-3: Using IEC6850 for condition monitoring diagnosis and analysis
IEC61850-90-4: Communication networks and systems for power utility automation	Part 90-4: Network engineering guidelines
IEC61850-90-5: Communication networks and systems for power utility automation	Part 90-5: Use of IEC61850 to transmit synchrophasor information according to IEEE C37.118

Table 6.9: Detailed Summary of the IEC 61850 Standard parts cont'd

General Description	Part Description
IEC61850-90-6: Communication networks and systems for power utility automation	Part 90-6: Use of IEC61850 for Distribution Automation Systems
IEC61850-90-7: Communication networks and systems for power utility automation	Part 90-7: Object models for power converters in distributed energy resources (DER) systems
IEC61850-90-8: Communication networks and systems for power utility automation	Part 90-8: Object model for E-mobility
IEC61850-90-9: Communication networks and systems for power utility automation	Part 90-9: Use of IEC 61850 for Electrical Energy Storage Systems
IEC61850-90-10: Communication networks and systems for power utility automation	Part 90-10: Models for scheduling
IEC61850-90-11: Communication networks and systems for power utility automation	Part 90-11: Methodologies for modelling of logics for IEC 61850 based applications
IEC61850-90-12: Communication networks and systems for power utility automation	Part 90-12: Wide area network engineering guidelines
IEC61850-90-17: Communication networks and systems for power utility automation	Part 90-17: Using IEC 61850 to transmit power quality data

6.3.3 Discussion

As mentioned in sub-section 6.3.1, synchrophasor data as measured and calculated by PMUs are very useful in assessing the condition of a power system. However, though useful, the IEEE Std C37.118.2™, 2011 is not compliant to the concepts presented within the IEC 61850 standard.

The development of methods within the IEC 61850 that would integrate synchrophasor transport became a necessity, and this led to the IEC Technical Report (TR) 61850-90-5 (Naspi, 2012; IEC 61850-90-5).

The IEC/TR 61850-90-5:2012(E) specifies ways of exchanging synchrophasor data between PMUs, PDCs WAMPAC (Wide Area Monitoring, Protection, and Control), and between control centre applications. Other than synchrophasor data as described in IEEE C37.118 that are transported in an IEC 61850 compliant way, the IEC/TR 61850-90-5:2012(E) also provides routable profiles for IEC 61850-8-1 GOOSE and IEC 61850-9-2 SV packets. These routable packets can thus be utilized to transport general IEC 61850 data as well as synchrophasor data.

This part of the IEC 61850 standard can be used for:

- Wide area applications utilizing synchrophasors

- Synchro-check

- Adaptive relaying

- Out-of-step (OOS) protection

- Situation awareness

- State estimation and online security assessment

- Archive data (event and continuous)

- Wide area controls

The communication protocol chosen in this thesis is the IEC 61850 standard, and this is illustrated by Figure 6.8 below.
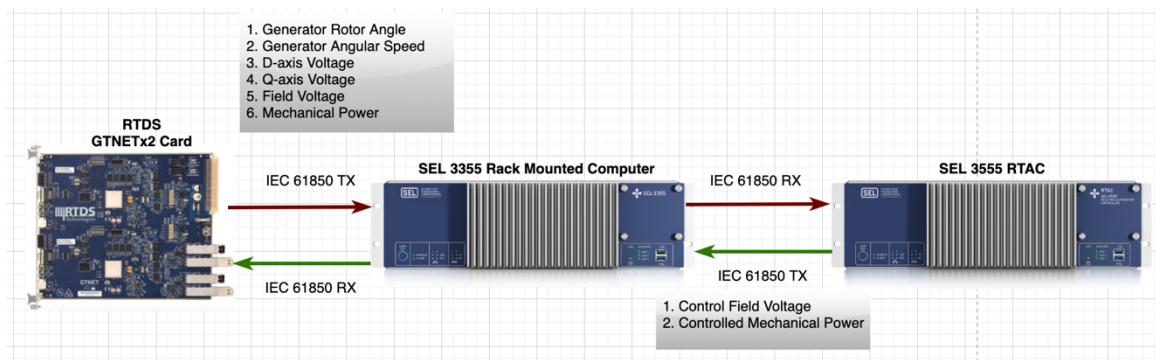
Figure 6.8: Overview of communication between various devices used in the implementation of the proposed MRAC-based power system interarea oscillations damping control scheme

6.4   Implementation of the Proposed MRAC Control Algorithm

6.4.1   Overview

In this penultimate part, details around the configuration of various components and associated software as well as communication protocol are presented. This ranges from the configuration of the SEL 3355 computer to the RTDS GTNETx2 card via the RSCAD software. Moreover, details around the implementation of the control algorithm in the RTAC is also presented.

Figure 6.9 describes the steps followed.
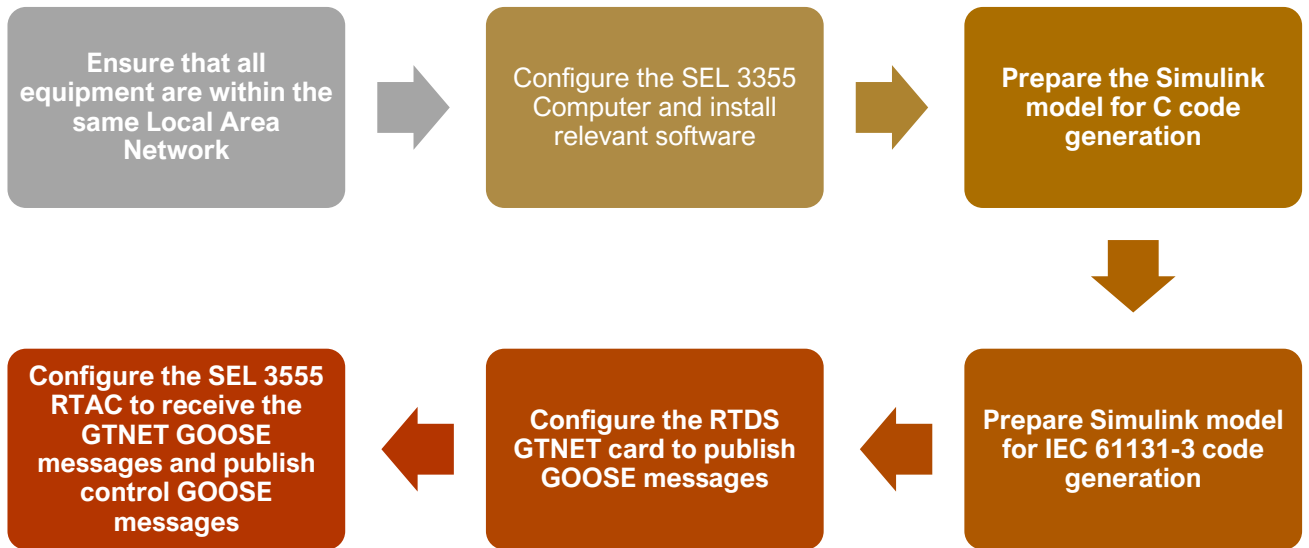
Figure 6.9: Overview of the steps followed in configuring the testbed for the proposed MRAC interarea oscillations damping algorithm

Each step described in Figure 6.9 is discussed in the following sub-sections.

6.4.2. SEL 3355 Computer Configuration

This industrial substation computer is central to the HIL because all the software used for configuring the GTNETx2 card and the SEL RTAC as well as the MRAC implementation and the conversation of the 4th order model of the synchronous generator from SIMULINK to RSCAD through the CBuilder.

Pre-configured with *Windows 10 IoT Enterprise*, it had the following software installed:

- *AcSELerator Architect*: Used for importing published Goose by the GTETx2 card of the RTDS and defining analogue tags that would contain controlled signal to be sent back to the RTDS via GOOSE.

- *AcSELerator RTAC*: This is the computing platform where runs the code that implements the oscillations damping algorithm.

- *MATLAB(SIMULINK)*: This is where the 4th order model of the generator is modelled and exported to run on a real-time platform i.e., the RTDS.

- *RSCAD*: Used for modelling the converted Simulink model but also for the configuration of the GTNETx2 card for GOOSE publishing.

To establish communication between this device and the other two in Figure 6.8, it is necessary to ensure they are all within the same IP range i.e., 192.168.1.xxx.

6.4.3. Simulink to RSCAD CBuilder
6.4.3.1. Simulink model and C/C++ Code Generation
6.4.3.1.1 Overview of the SIMULINK® Model

It is worth highlighting that the words machine and synchronous generator will be interchanged throughout the remainder of the chapter.

The power system model exported to the RSCAD software is presented here. As mentioned in Chapters 3 and 5, it is a configuration of a single machine or groups of machines connected to an infinite bus and is used in the derivation of the equations describing the dynamics of the synchronous generators. Furthermore, and as emphasized in the aforementioned chapters, the 4th order model which encompasses the 3rd is chosen in the design of the controller.

Figure 6.10 shows the Simulink model.
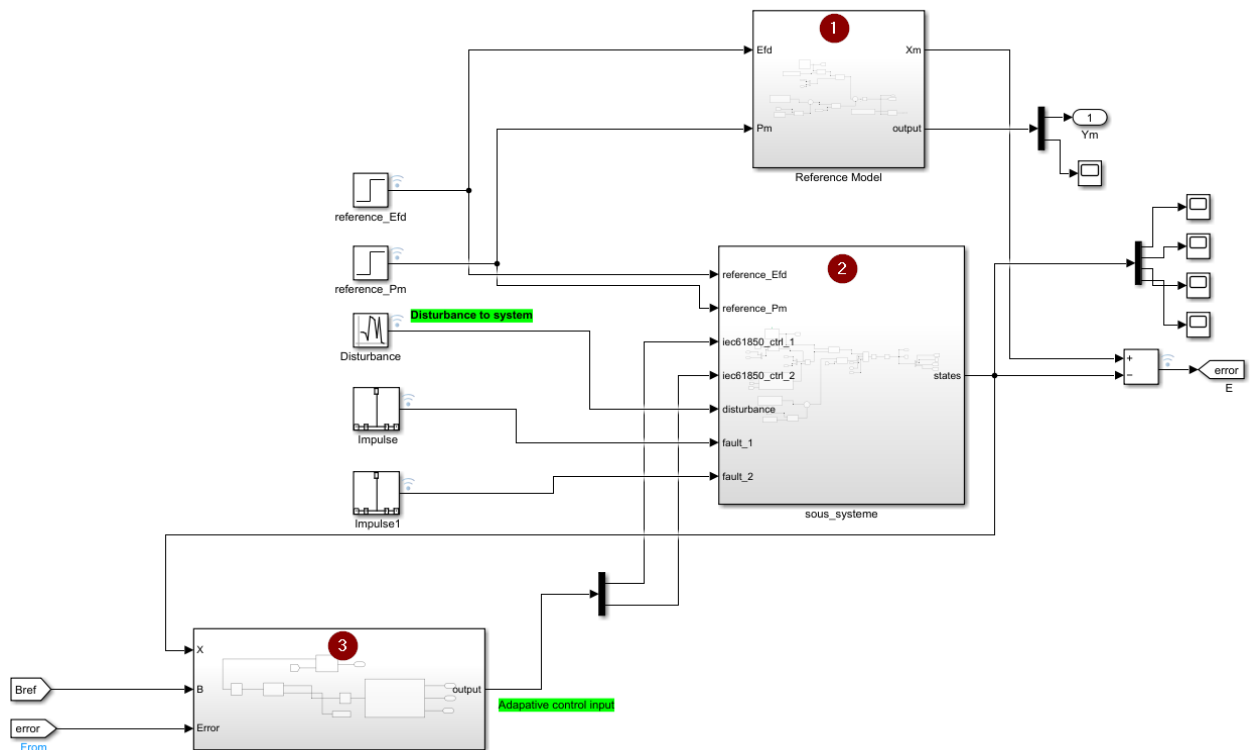
Figure 6.10: Overview of the Complete MRAC Simulink Model

The subsystem (1) contains the reference model with its LQR controller, subsystem (2), the 4th order representation of the synchronous generator with the nominal (LQR) controller and the subsystem (3) MRAC *Lyapunov* candidate. The subsystem of interest which is an expansion of the model of synchronous generator with the nominal LQR controller is shown in Figure 6.11.
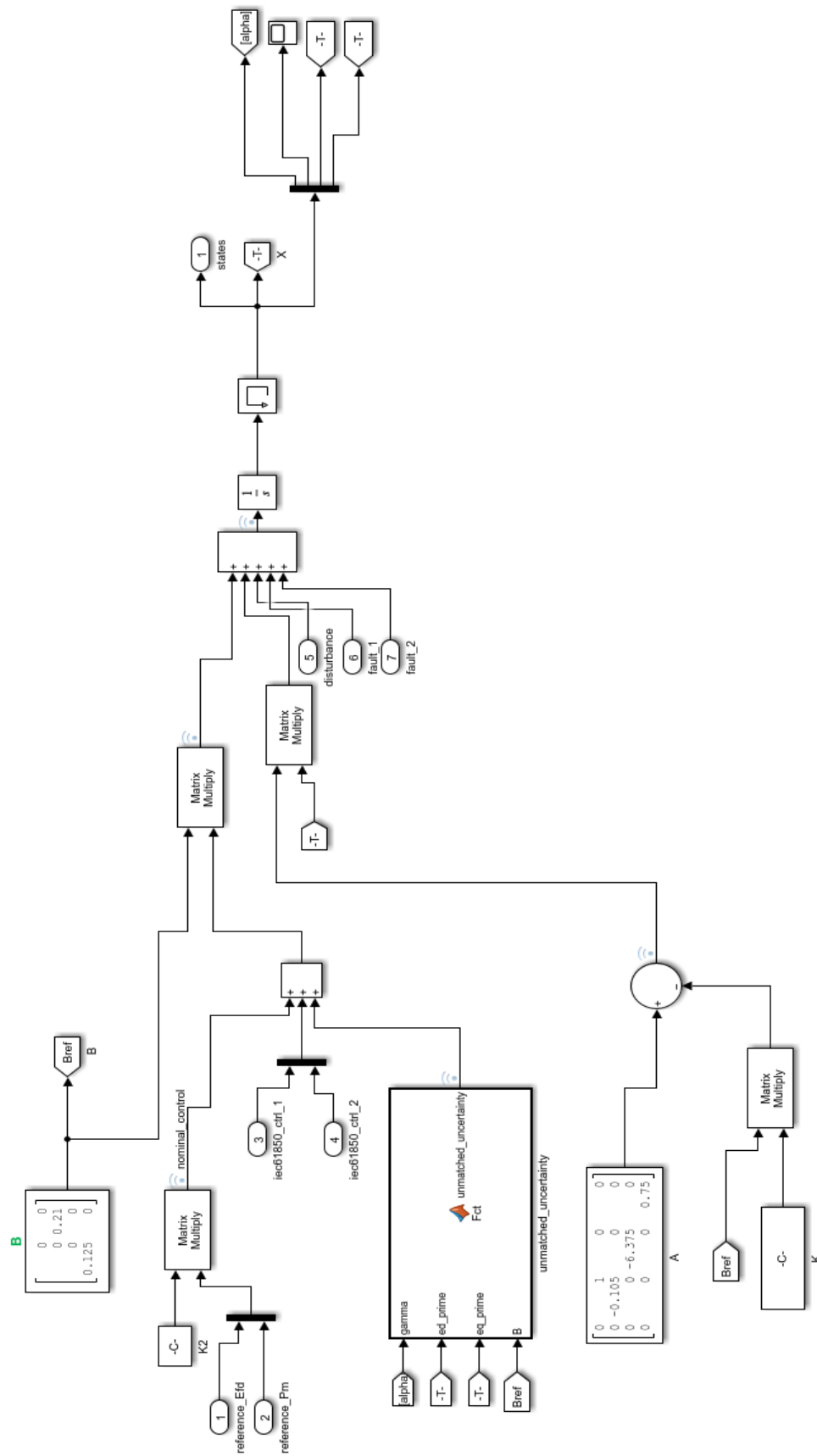
Figure 6.11: Fourth-order Order model of the synchronous generator with a nominal controller

163

6.4.3.1.2 Preparation of the SIMULINK® Model for C Code Generation

Certain prerequisites are to be adhered to for a successful Simulink to RSCAD conversion. In this sub-section, the minimum MATLAB and Simulink packages required as well as considerations around the very Simulink model to be converted are described and discussed. Also highlighted are the model design considerations and restrictions.

Table 6.10 details the minimum MATLAB and Simulink packages requirements.

Table 6.10: Detailed Summary of the IEC 61850 Standard parts

Package	Version	MATLAB Release
MATLAB	7.4	(R2012a)
MATLAB Coder	2.2	(R2012a)
Simulink	7.9	(R2012a)
Simulink Coder	8.14	(R2012a)
Simulink Control Design	3.5	(R2012a)
Control System Toolbox	9.3	(R2012a)
Embedded Coder	6.2	(R2012a)

The version of MATLAB and Simulink used in this project are (R2021a – academic use).

Besides the versioning, the instructions below also need to be taken into consideration. Failure to do so, the import to RSCAD would fail (RSCAD manuals, 2021).

- Simulink control models only.

- Fixed-step solver to be used for the model.

- If any, only inline S-function blocks are accepted.

- ANSI C code of the MATLAB's embedded code is the required tool to be used for conversion.

- Only control models are supported. No MATLAB/Simulink power system models, or components are allowed at this stage.

- Avoid allocating large amount of memory for component's buffer.

- Avoid the $uint8\_T$ or $uint8$ data types due to the difference in handling of unsigned chars by Simulink and RSCAD.

- Multi-rate whereby certain components do run at different time steps owe to be avoided.

- The naming of I/O ports and signal lines should be unique to avoid potential referencing compiling errors in RSCAD when compiling the Simulink-RSCAD CBuilder component.

These restrictions may affect the design of a given Simulink model:

- As stated earlier, the variable-step solver would not just prevent the conversion of Simulink models to RSCAD but also prevent Simulink from generating the C code.
- Algebraic loops are to be avoided. Add a memory block after the integrator for instance. Feedback signals are a typical example of these loops.
- The solver fundamental sample time must be clearly defined.
- The existence of any libraries linked to a given component is required, else Simulink would not generate the C code.

With the above adhered to, the steps described below are followed for the Simulink C code generation of the model in Figure 6.11.

Step 1 – Solver setup: As recommended by (RSCAD manuals, 2020), one of the fixed steps solvers must be chosen with a fundamental sample time corresponding to that of the RTDS simulation time-step of $50 \, \mu s$.

Step 2 –Code generation: Configuration of the Simulink code generator followed the steps described in Figure 6.12. Detail explanation of each process are provided in the following lines.
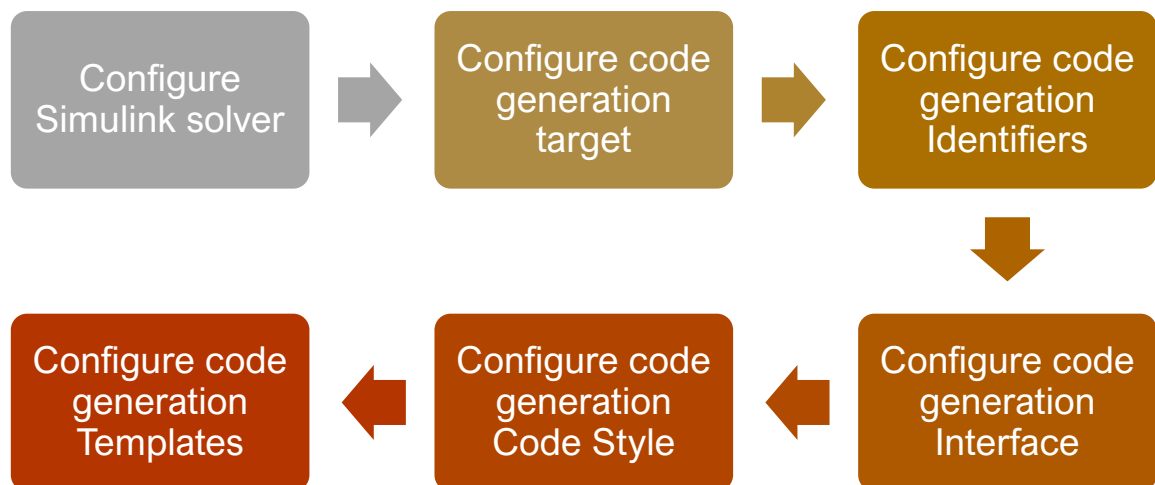
Figure 6.12: Overview of the Simulink C code generation process

i.   *Under Simulation, select Model Settings:*
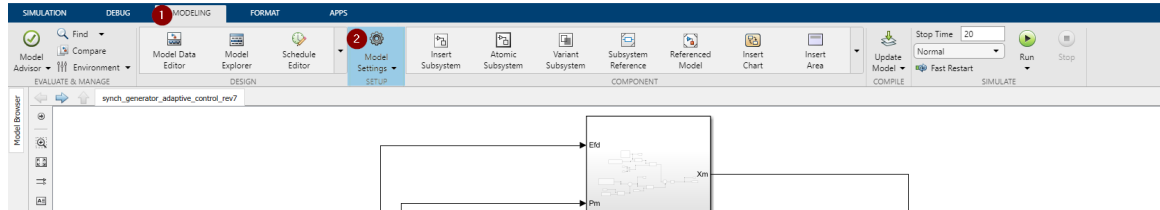
Figure 6.13: Simulink model settings

ii.   *Solver Detail: The stop time is set to 2000 sec. so that the simulation runs longer in RSCAD.*
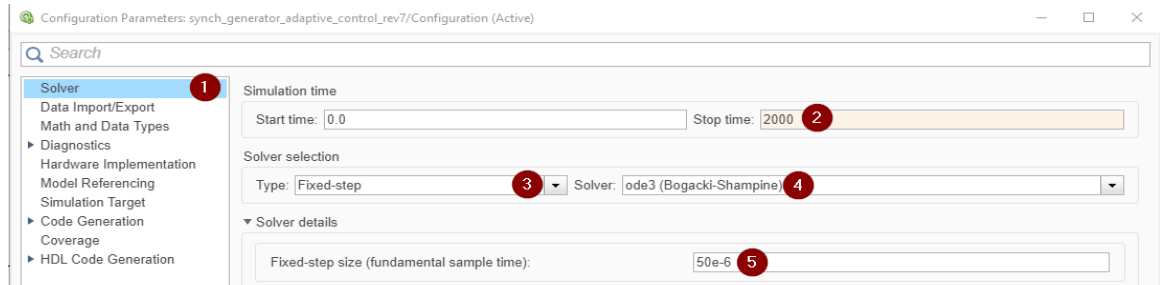
Figure 6.14: Simulink solver information

iii.   *Code Generation: Under Target Selection, browse and select the ert.tlc system target file.*
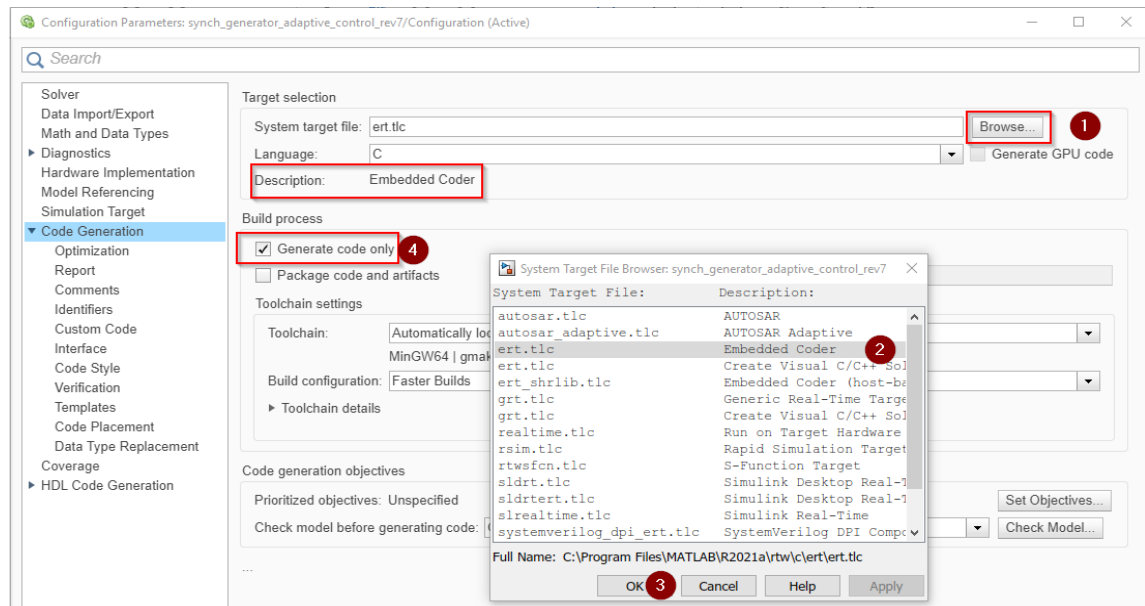
Figure 6.15: Configuration parameter window

iv.   *Under identifier, amend the settings to match those in Figure 6.16*
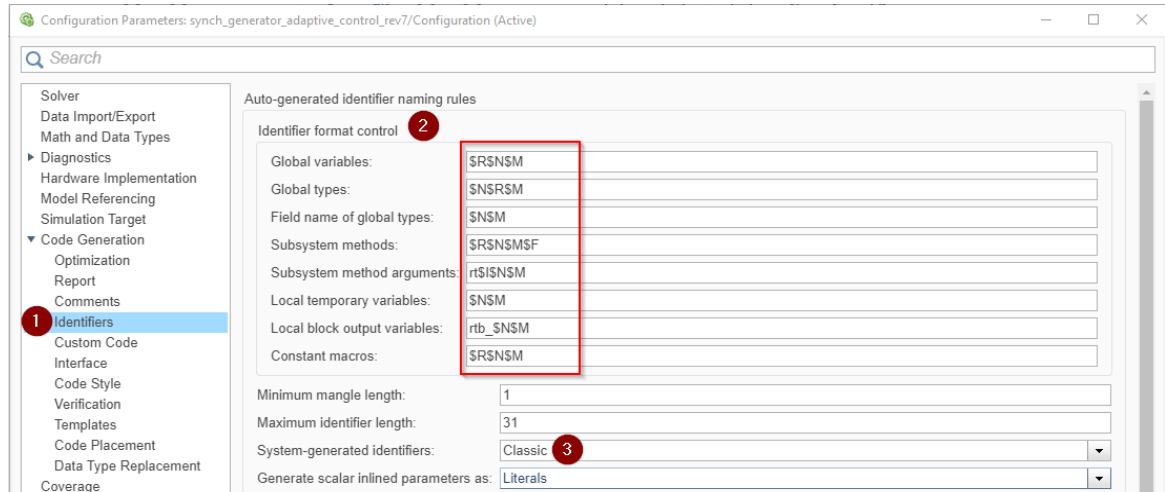
Figure 6.16: Symbols panel of the configuration parameter window

v.   *Under interface, amend the settings to conform with those in Figures 6.17a and 6.17b*
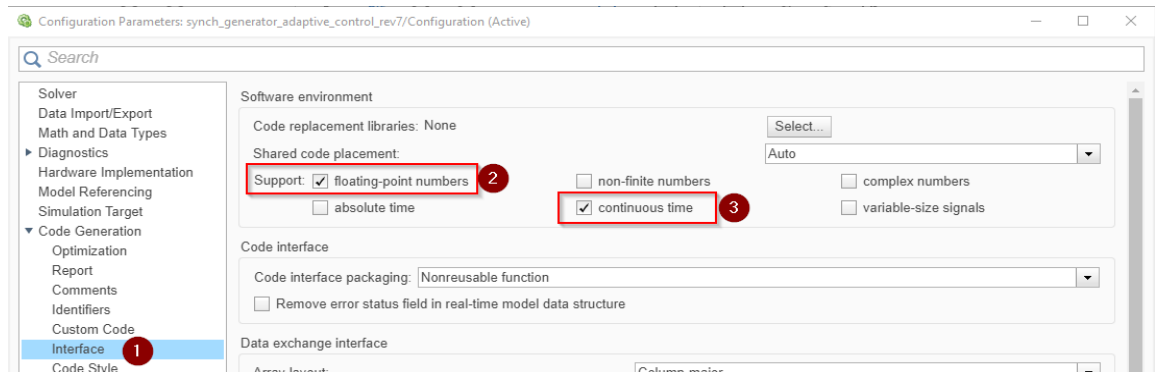
Figure 6.17a: Interface panel of the configuration parameter window (part 1)
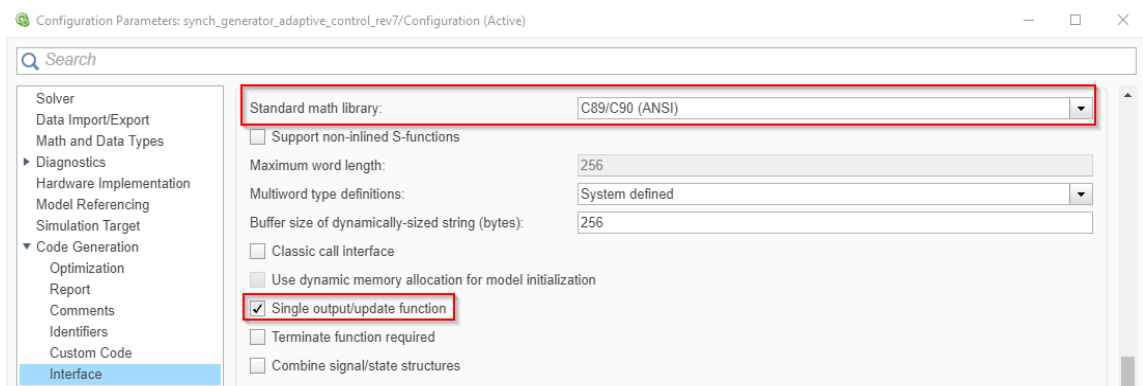
Figure 6.17b: Interface panel of the configuration parameter window (part 2)

vi. *Under code style, select preserve extern keyword as below.*
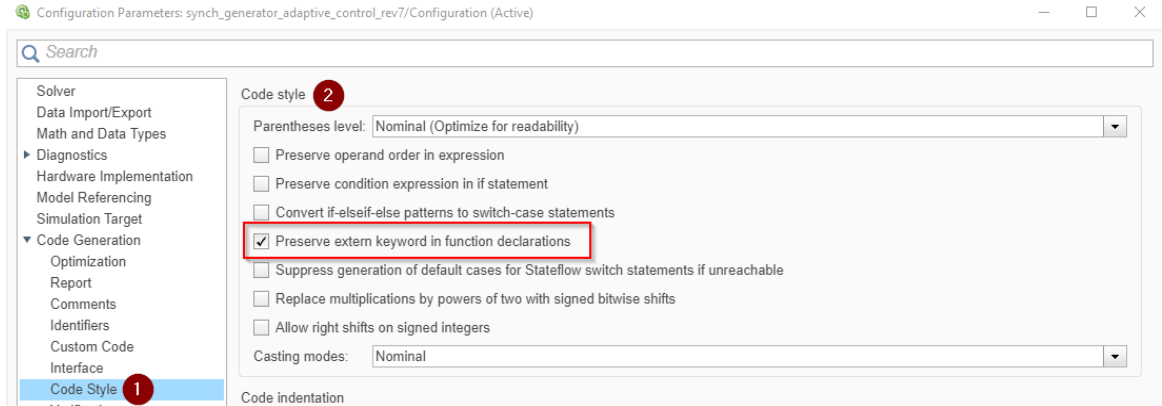
Figure 6.18: Code Style panel of the configuration parameter window

vii. *Under templates, ensure the generation of the main program is selected.*
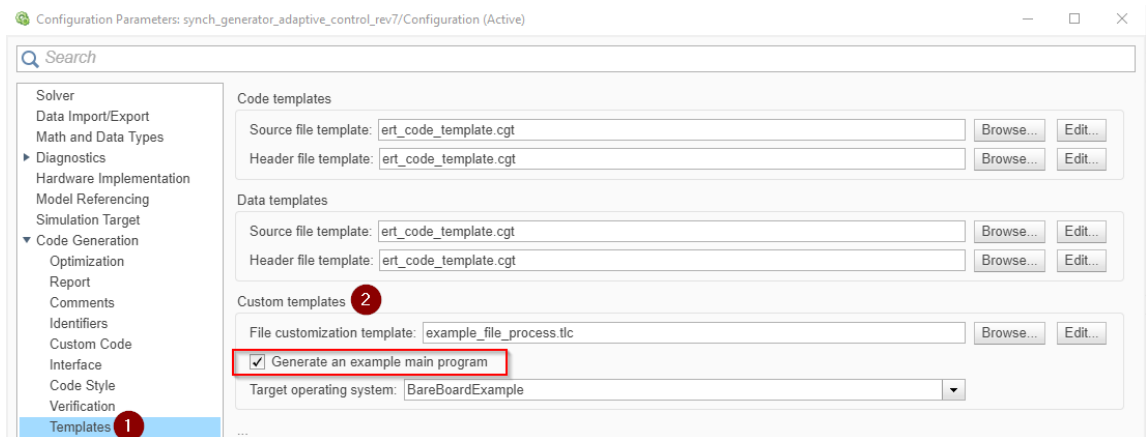
Figure 6.19: Template panel of the configuration parameter window

Step 3 – Generate Code: Under C-Code, select Embedded C Code, then Embedded C Code again as below.
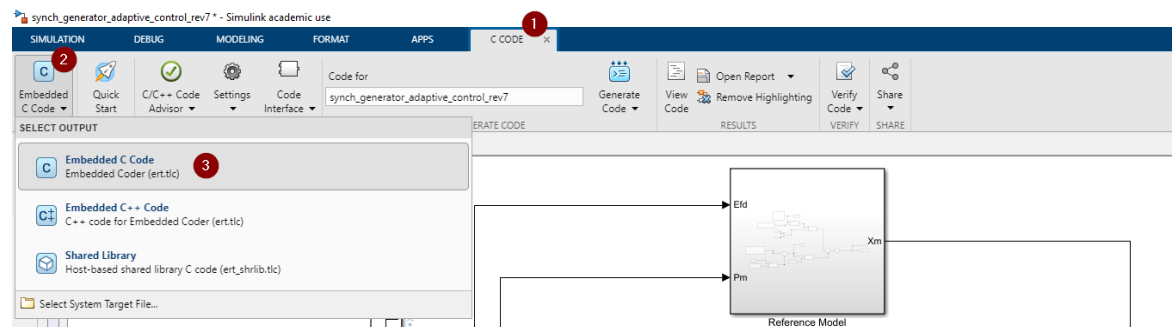
Figure 6.20: C code generation process

Should all go well, and after a right click on the relevant subsystem and selecting Build This Subsystem as in Figure 6.20, a screen such as the one in Figure 6.21 will be returned.
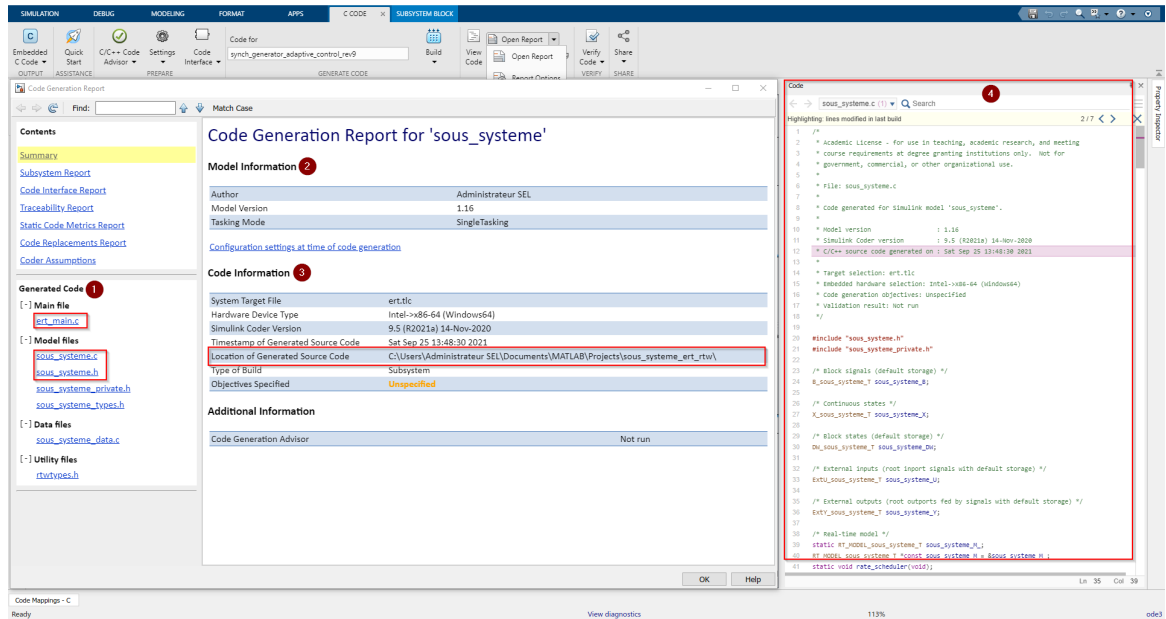
Figure 6.21: Generated C code report

From the above Figure, (1) shows the various files generated in the process with Table 6.11 providing detail explanation of their meaning, (2) provides information about the model, (3) the information about the generated code with the emphasis on its location, and (4) a snippet of the C code.

Table 6.11: Detailed Summary of the IEC 61850 Standard parts (Adapted from RTDS Manual, 2021)

Naming Convention	Actual File Name	File Description
modelName.c*	sous_system.c	This file contains all the major function to implement the Simulink model
modelName.h*	sous_system.h	This file contains the interface for all the major functions and data structure declarations
modelName_data.c	sous_system _data.c	These files contain data structures specific to the model
ModelName_private.h	sous_system _private.h	
modelName_types.h	sous_system _types.h	
ert_main.c*	ert_main.c	An example of the main program generated by the Embedded Coder

Important C file generated by the embedded coder

6.4.3.2. Importing Generated C Code Into RSCAD

This is done through the RSCAD CBuilder following the steps below.

Step 1 – C code import: After Launching the RSCAD CBuilder from the main software page, select Tools >> Import Simulink Model and a screen similar to Figure 6.22 will be presented.

 Step 2: Browse to the path shown in Figure 6.21 and select the sous_system.c file, type in the name while leaving the rest to their default settings as in Figure 6.22. Pressing OK would save the imported file and show the details of the imported file as shown in Figure 6.23.
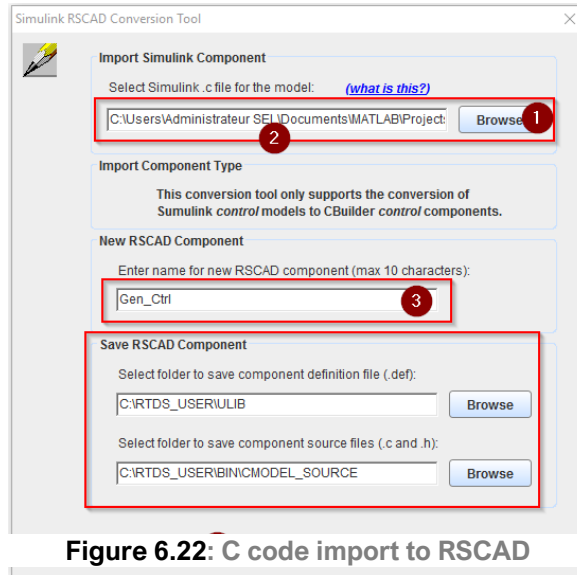
Figure 6.22: C code import to RSCAD
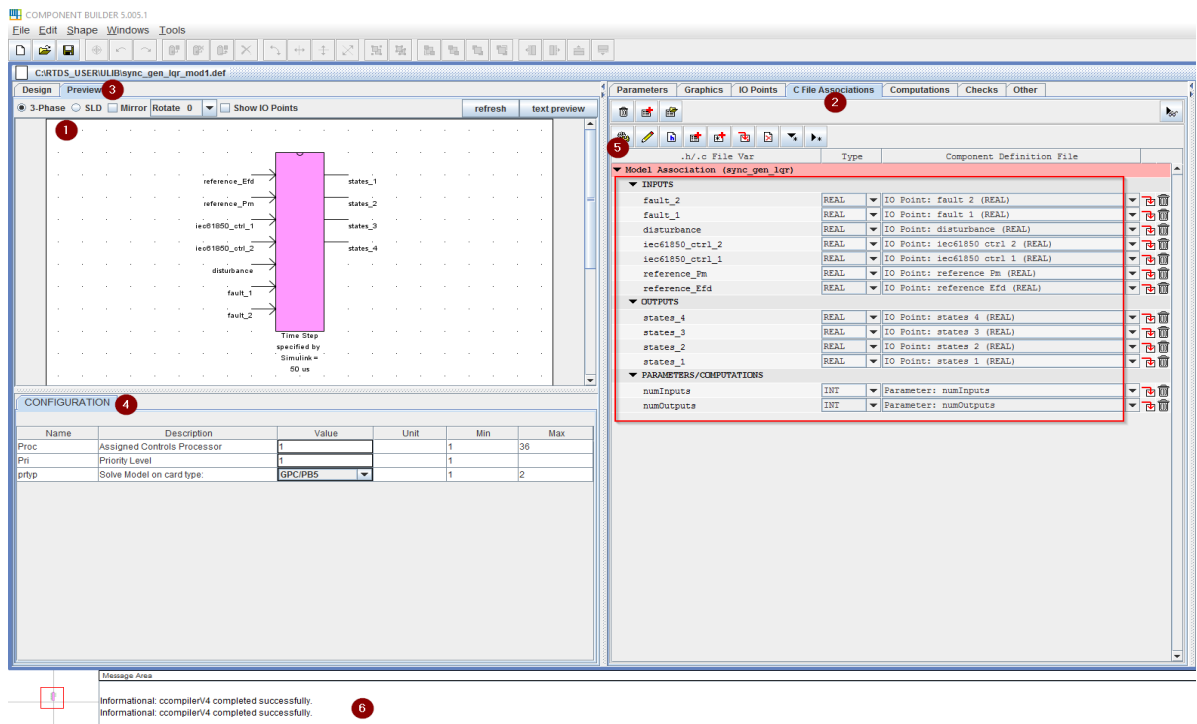
Figure 6.23: CBuilder view of the imported Simulink C code

170

Figure 6.23 shows the design pane containing the generated component based on the imported Simulink C code. On the right pane, under C File Associations (2), various inputs and outputs as defined in Simulink are listed as in. On the left pane, under Preview (3), detail around the default assigned processor is listed together with the processor card type. To import the generated model in the RSCAD Draft for simulations, it is compiled (5) and result of the compilation shown in (6).

Step 3: Check that the time step specified in CBuilder by the newly created component matches the one specified in Simulink as in (1).

6.4.4. GOOSE Messages Publishing and Subscription
6.4.4.1. Adding the Newly Added Component to the RSCAD Draft

The process described in Figure 6.24 is to be followed to use the Simulink-CBuilder component in the RSCAD Draft:

Figure 6.24: Process of adding the newly created Simulink-CBuilder component in a new RSCAD project

Once imported, various library components to be associated to the case can be added. Figure 6.25 shows the modelled system in the RSCAD software.
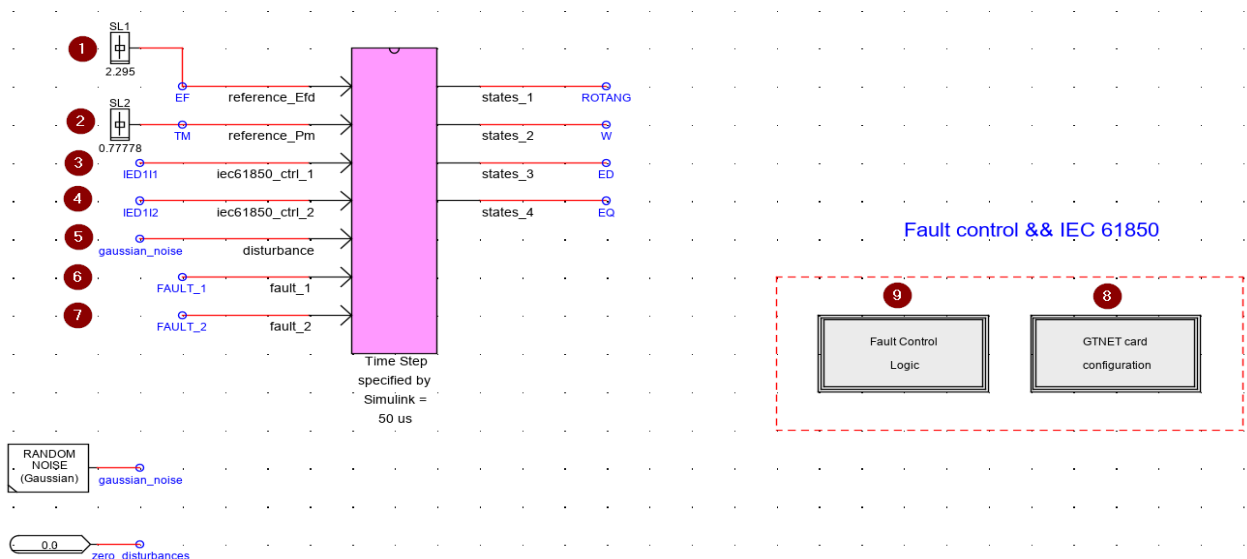
Figure 6.25: Overview of the thesis RSCAD Draft

Inputs (1) and (2) are the setpoints i.e., the field voltage E_{FD} and mechanical power P_m. The use of sliders for those inputs is motivated by the fact that those changes in setpoints will form part of test cases in Chapter 7. Inputs (3) and (4) represent the GOOSE MRAC control signals coming from the RTAC as illustrated in Figure 6.8, while input (5) is fed with a Gaussian noise to emulate small variations in load that are inherent to a power system. As mentioned in Chapter 5, they are caused by small variations in the system load and are mainly responsible for Low-Frequency Electromechanical Oscillations (LFEOs). Lastly, inputs (6) and (7) are the faults added onto the system.

As for the outputs, though Figure 6.9 refers to one output which is in fact the system states, the model being a 4th order one, that would translate in $state\_i$, with $i \in [1,4]$.

Figures 6.26a and 6.26b show the contents of the fault logic expansion (8) and (9).

<div style="text-align:center">
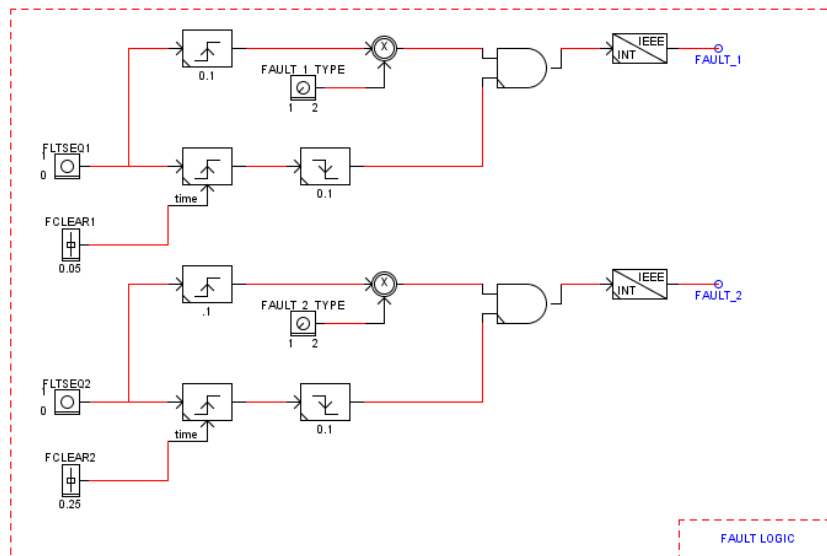
Figure 6.26a: Fault logic   **Figure 6.26b**: Overview of the GTNET card configuration tool
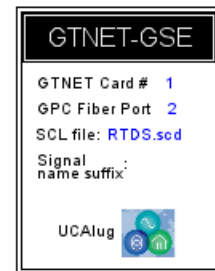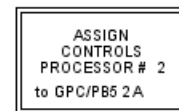
</div>

6.4.4.2. GTNET Configuration for GOOSE Publishing

Two (2) main steps are to be followed to publish variables of interest via GOOSE to the RTAC. These are the two setpoints mentioned in the previous section (E_{FD} and P_m) together with the states i.e., the rotor angle (ROTANG), rotor speed (W), d-axis voltage (ED), q-axis voltage (EQ).

These steps are - the GTNET card configuration and SCL file creation. Figure 6.27 highlights the steps one must follow to configure the GTNET card.

Figure 6.27: GTNET card configuration process

Figures 6.28a, 6.28b, 6.28c, 6.28d illustrate the above process.

Figure 6.28a: Configuration tab of the GTNET card configuration

Figure 6.28b: GOOSE configuration tab of the GTNET card configuration

Figure 6.28c: Output dead band configuration tab of the GTNET card configuration

Figure 6.28d: Output signal names configuration tab of the GTNET card configuration

6.4.4.3. Substation Configuration Language (SCL) File Creation

To allow the configured signals i.e., the rotor angle ($ROTANG$), rotor speed (W), d-axis voltage (ED), q-axis voltage (EQ), mechanical power and field voltage to be published over the network, an SCL file needs to be created. Unless one had already been created, one must create it from a built-in template by right clicking on the GTNET component and selecting Edit >> IEC 61850 SCD File from Template as in Figure 6.29.
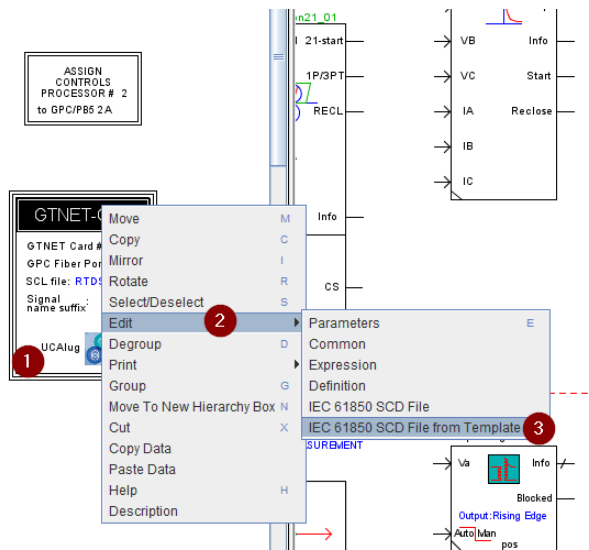
Figure 6.29: SCD file creation

The opened template will contain detailed information pertaining to the GTNET card configuration as set in sub-section 6.4.4.2. The next step would be to save the file and assign a name to it.

6.4.5. MRAC Algorithm Implementation in the SEL RTAC 3555

In this subsection are provided detailed steps followed to implement the MRAC algorithm in the SEL RTAC 3555. This is illustrated in Figure 6.30 below.

Configure Simulink for PLC code generation

Generate code for the relevant subsystems

Create a new RTAC project and choose the applicable RTAC based on its firmware

Create Function Blocks that would contain the generated reference model code as well as the adaptation part

Create a program that would make use of those function blocks in conjunction with the received IEC 61850 GOOSE messages and publish control sgnal to the RTDS

Figure 6.30: Process describing the MRAC implementation in the SELC RTAC

174

6.4.5.1 Simulink PLC Coder for IEC 61131-1 Code Generation

Unlike the process for generating C code for embedded platforms as described in subsection 6.4.3, the generation of a hardware-independent IEC 61131-3 Structured Text necessitates that no continuous state component exists within any of the subsystems that would be used to generate the IEC 61131-1 code. Component such as continuous integrators must be replaced with their discrete counterparts. Furthermore, for all feedback loops, introduce a discrete unit delay.

The following lines will describe the steps to follow for the generation of a Structured Text file code for the SEL RTAC 3555.

Step 1 – Solver Configuration: A fixed step solver is required with a fundamental sample time that corresponds to that of the RTDS $50\ \mu s$ as in Figure 6.31 below.
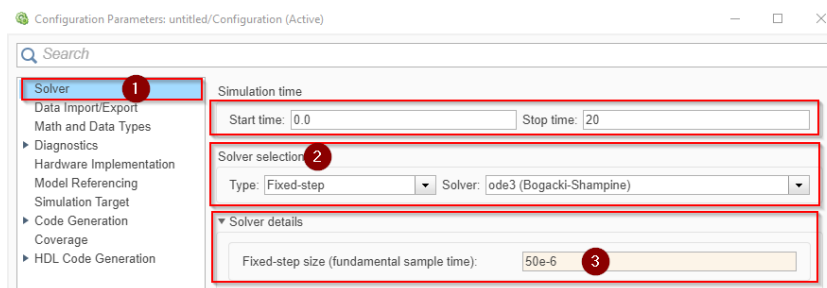
Figure 6.31: Solver configuration for Structured Text Generation

Step 2 – Simulink PLC Coder app: This feature might not be enabled by default even with the current R2021 a version. Figure 6.32 shows how to make it available for the project.
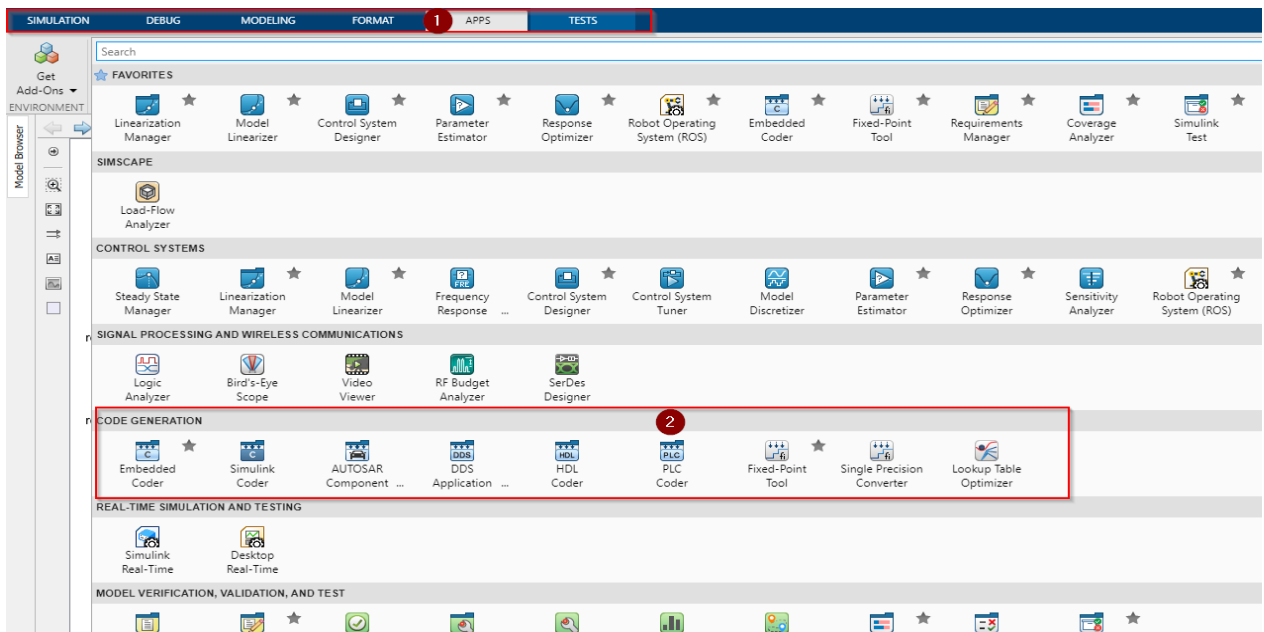
Figure 6.32: Simulink PLC Coder enabling process

Step 3 – PLC Code Generation Configuration*:* Under Target IDE, select "Generic" since AcSELerator RTAC is not listed amongst supported IDEs, and ensure that "Show full target list" is selected as in Figure 6.33. One may opt to allow Simulink to generate a testbench for subsystems; the corresponding checkbox would have to be selected.
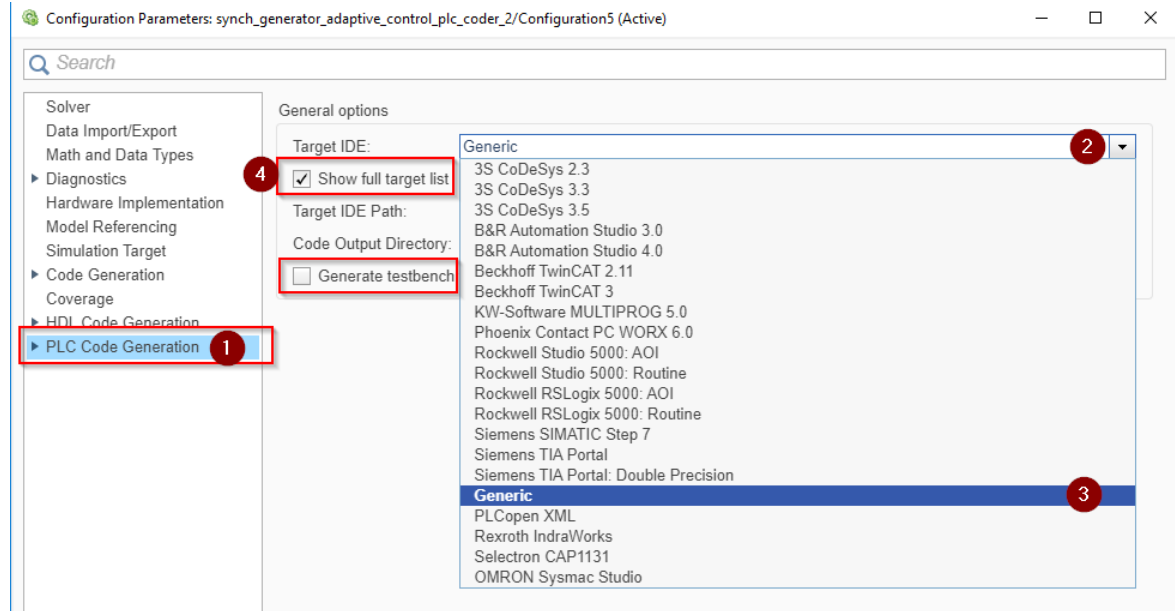
Figure 6.33: Simulink PLC target selection for Structured Text generation

Step 4 – PLC Code Generation*:* Figure 6.34 shows the system being modelled for the PLC code generation. In contrast with Figure 6.9 where quite a substantial number of inputs are added, only a few are needed since the subsystems of interest are those in (1) and (2).
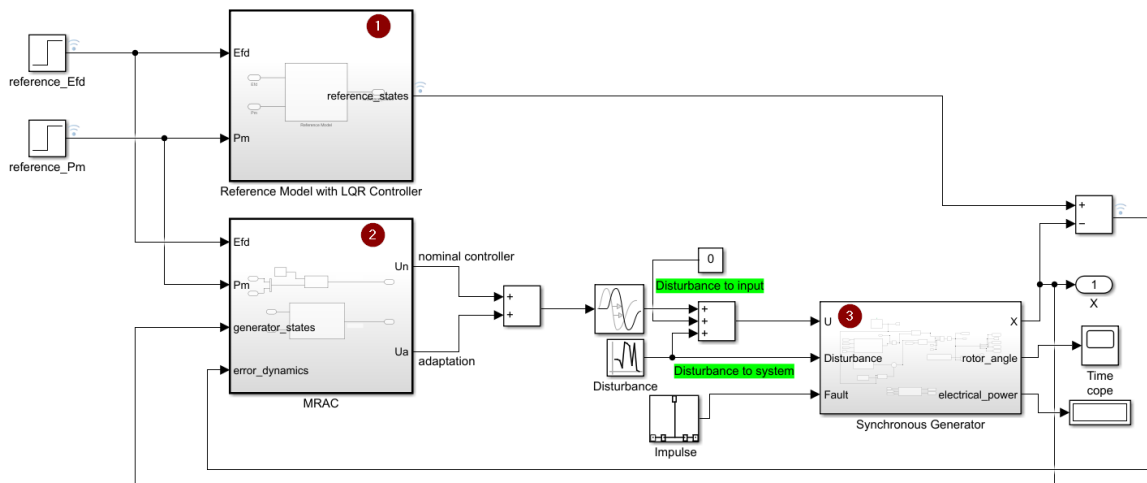
Figure 6.34: Structure of the MRAC for the 4th order model of the synchronous generator

176

For each of the subsystems of interest, right-click and select "*PLC Code*" >> "*Generate code for subsystem*" as illustrated in Figure 6.35 below.
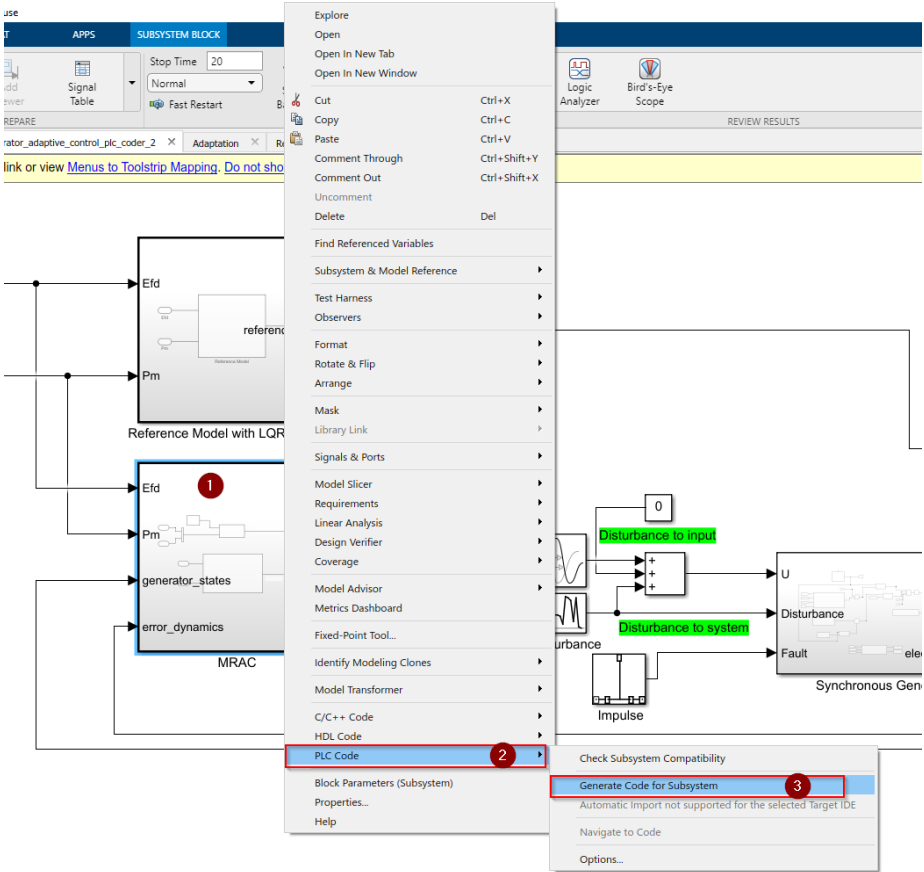
Figure 6.35: Process of generating the Structured Text code for subsystems

Step 5: Upon successful generation, a report similar to the one in Figure 6.36 is returned.
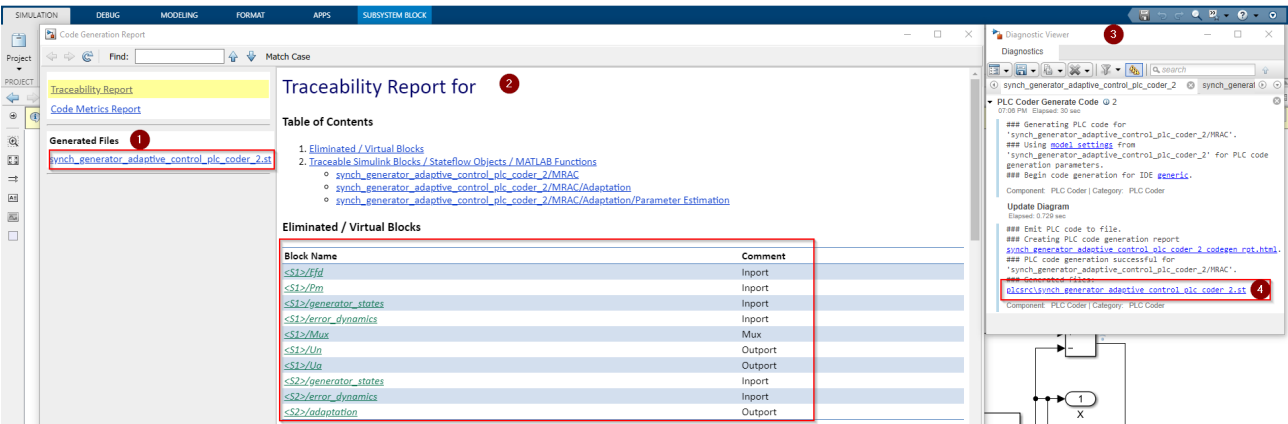
Figure 6.36: Generated report after PLC code generation.

(1) shows the file name, while (2) contains details around the various inputs and outputs found in the generated code. Inside the Diagnostic Viewer (3), the path to the code can be found (4).

6.4.5.2 AcSELerator Architect Configuration

To allow communication between the RTDS GTNET card and the SEL-3555 RTAC, the AcSELerator Architect software is configured such that the defined GOOSE messages as defined in the SCD file in Section 6.4.4.3 can be imported and utilized.

The process to achieve that is summarized below.

Step 1: Launch the AcSELerator Architect software, and in the IED Palette pane import the created GTNET SCD file created in sub-section 6.4.4.3 as in Figure 6.37.
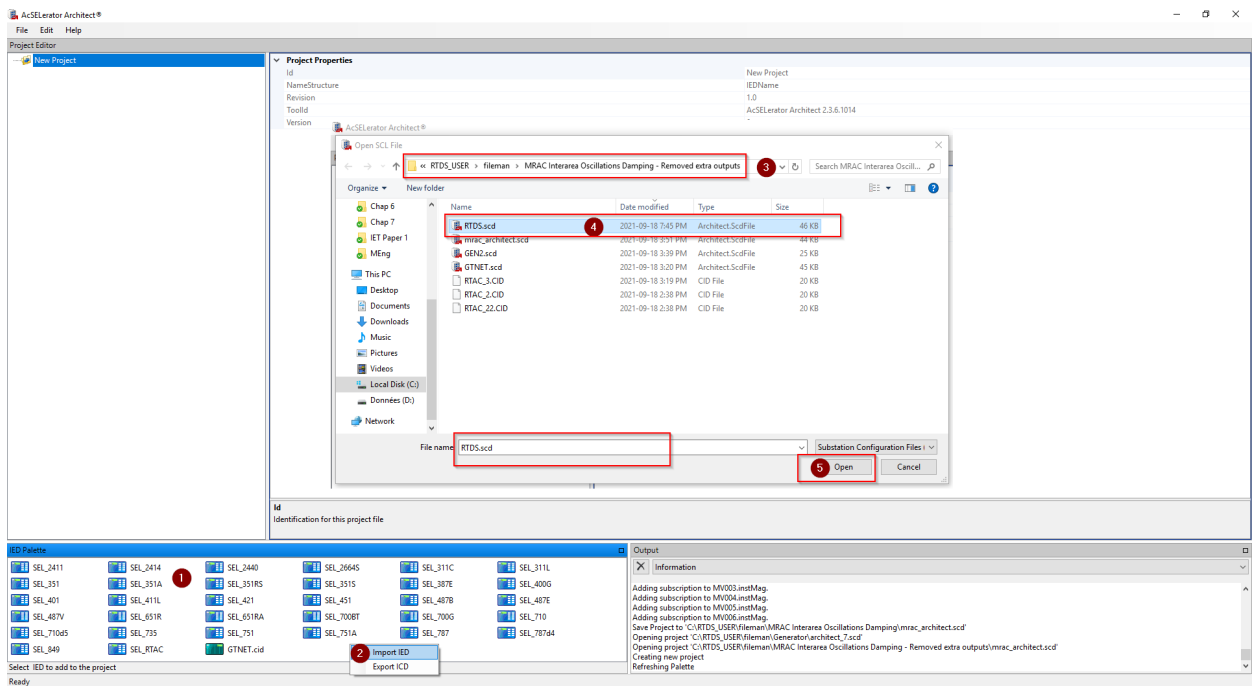
Figure 6.37: Importing of the saved GTNET SCD file to AcSELerator Architect.

Step 2: By drag and drop, the SEL_RTAC and the imported RTDS.scd file are added into the Project Editor pane. At the end of the process, the editor resembled Figure 6.38.
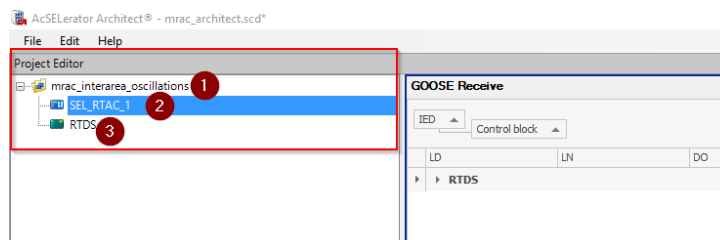
Figure 6.38: AcSELerator Architect. Project Pane after IEDs are added

178

Step 3: Add the network configuration of the RTDS IED, with the IP Address and Subnet Mask being most critical. This is illustrated in Figure 6.39.
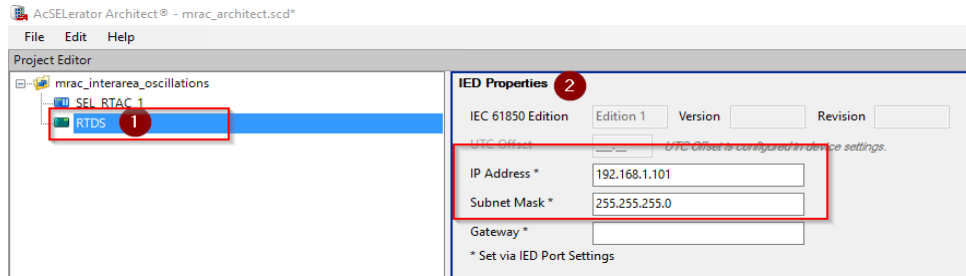
Figure 6.39: RTDS IED network configuration

Step 4 SEL_RTAC_1 Configuration: Here is where the bulk of the configuration takes place. As seen in the menus at the bottom of the IED Properties pane in Figure 6.39a, besides the network configuration, parameters around the receiving of the published GOOSE messages by the RTDS GTNET card or those pertaining to the publishing of the control inputs by the SEL RTAC are configured. This is illustrated in Figures 6.40b-6.40h.

    i.   Add the correct network information
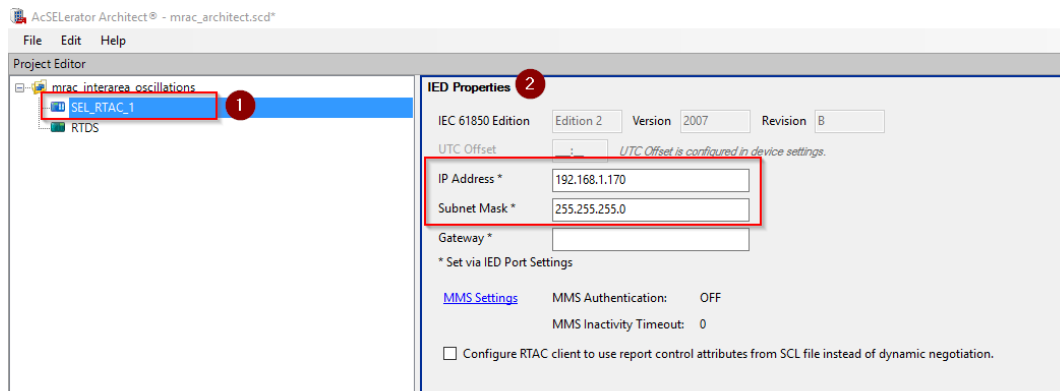
Figure 6.40a: SEL_RTAC_1 network configuration

    ii.   Under *Server Model,* click on the blank space to start with the process of adding logical devices.
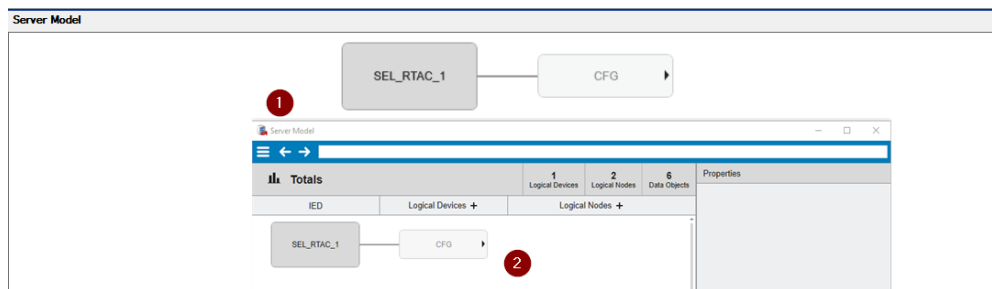
Figure 6.40b: SEL RTAC_1 server model

iii.   Then, click on the (+) sign in the *Logical Devices* tab to add one
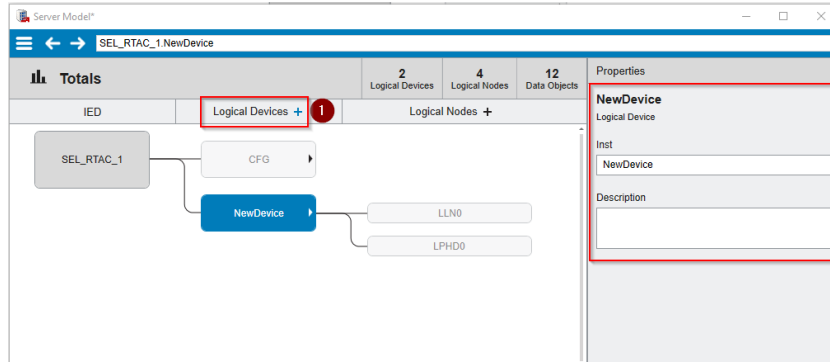
Figure 6.40c: SEL_RTAC_1 adding a new logical device

iv.   Add a new logical node by clicking on the (+) sign in the *Logical Nodes*' tab.
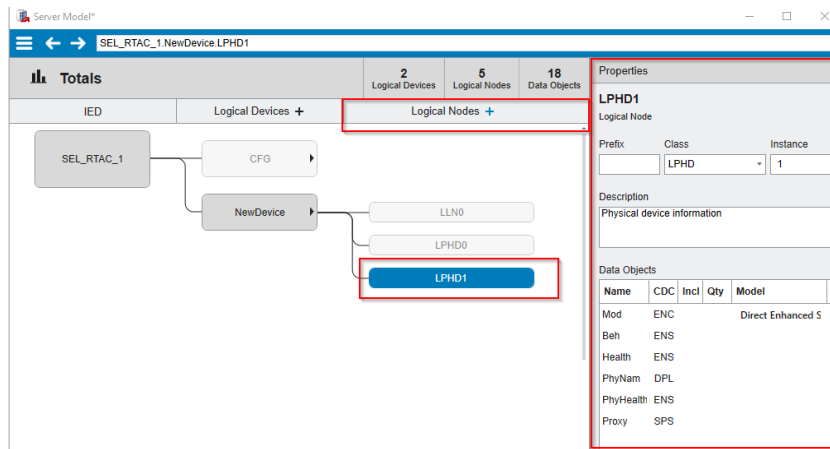
Figure 6.40d: SEL_RTAC_1 adding a new logical node

v.   Modify the default properties of the newly added logical node to correspond to the project's requirements i.e., configure to analogue outputs to be utilized by the SEL_RTAC_1.
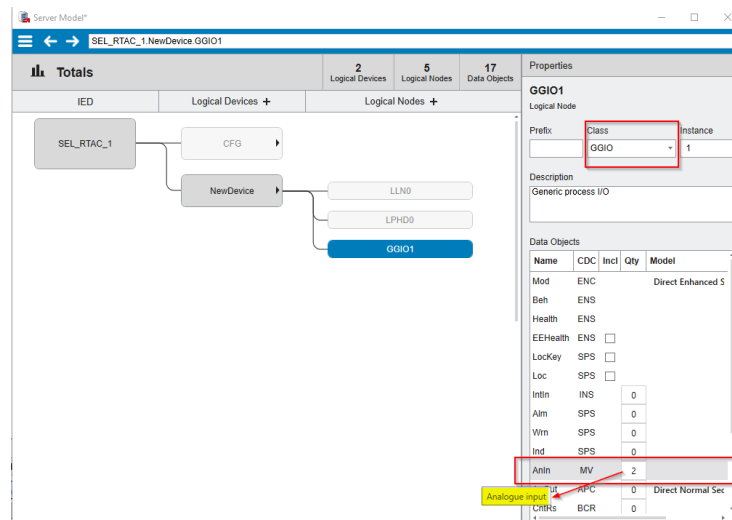
Figure 6.40e: SEL_RTAC_1 logical node configuration

vi.   Press the OK button to save the new configurations.

vii.   Under *GOOSE Receive*, drag-and-drop all 6 messages to the right-hand panel. Upon completion of the process, LEDs on the left pane will turn green.
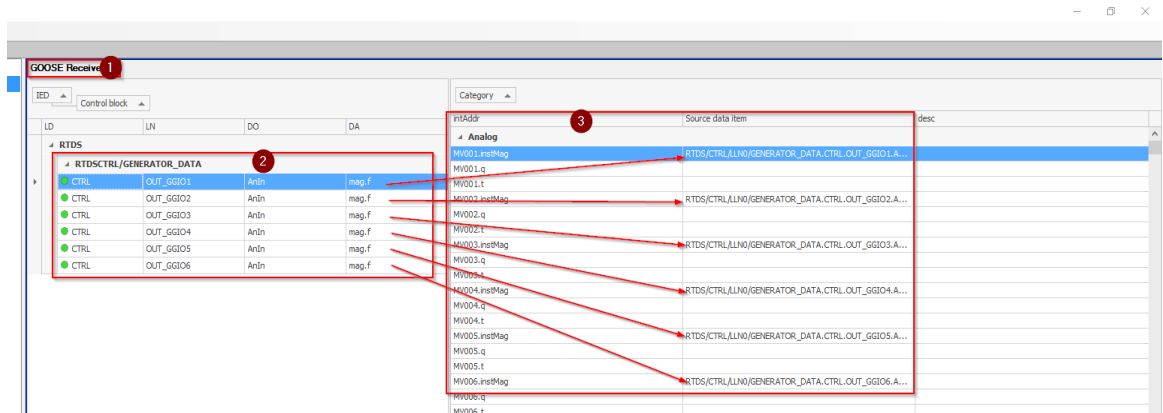
Figure 6.40f: **SEL_RTAC_1 mapping of received GOOSE messages**

viii. Under *Dataset*, select the applicable *Functional Constraint* (FC), then drag the two instantaneous analogue inputs to the right-hand pane.
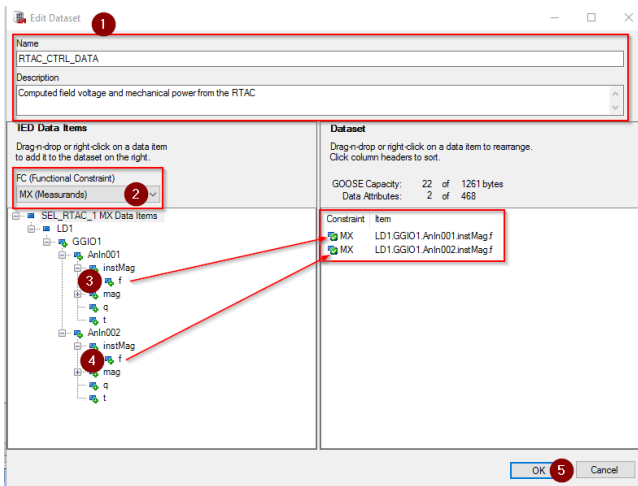
Figure 6.40g: **SEL_RTAC_1 Datasets configuration**

ix.   Under GOOSE Transmit, click New and associate with the dataset created earlier
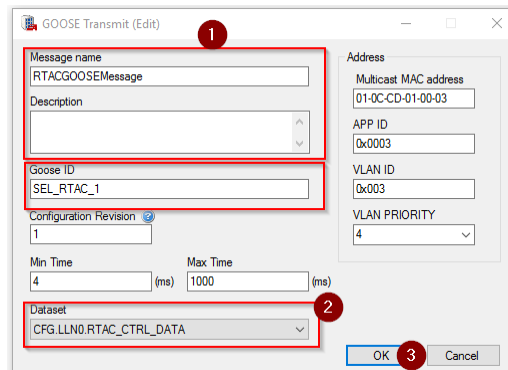
Figure 6.40h: **SEL_RTAC_1 GOOSE transmit configuration**

181

Step 5*: Export the SEL_RTAC_1 configuration as an CID file so that it can be imported by the RTDS SCL editor.*
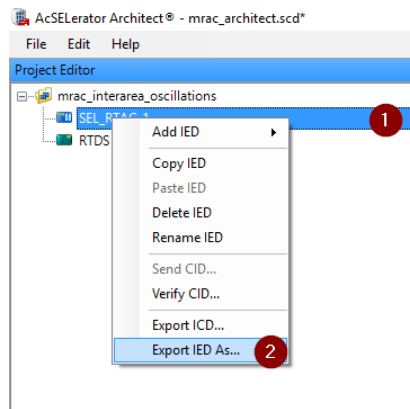
Figure 6.41: Exporting the SEL_RTAC_1 IED as an SCD file

Step 6*: Save the project as an SCD file so that it can be imported in the AcSELerator RTAC Software.*
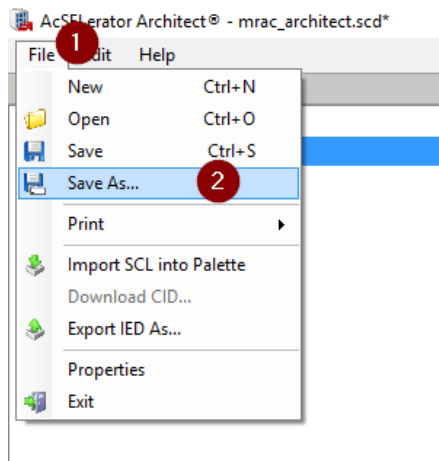
Figure 6.42: Saving the project as an SCD file

6.4.5.3 AcSELerator RTAC Setup and IEC 61131-1 Code import

A new project is created in the AcSELerator RTAC software. The language chosen is Structured Text to conform with the generated IEC 61131-1 code generated by Simulink. Two function blocks and one main program files are created under the User Logic section, with the first two containing converted Simulink code and the latter MRAC implementation. Also imported is the saved project in ***Step 6*** *of the previous section. It is imported as an IEC 61850 Configuration: Insert >> Set IEC 61850 Configuration described in Figure 6.42.* Figure 6.43 shows a snippet of the overall code structure.

Figure 6.42: Importing of the AcSELerator Architect project into the AcSELerator RTAC project
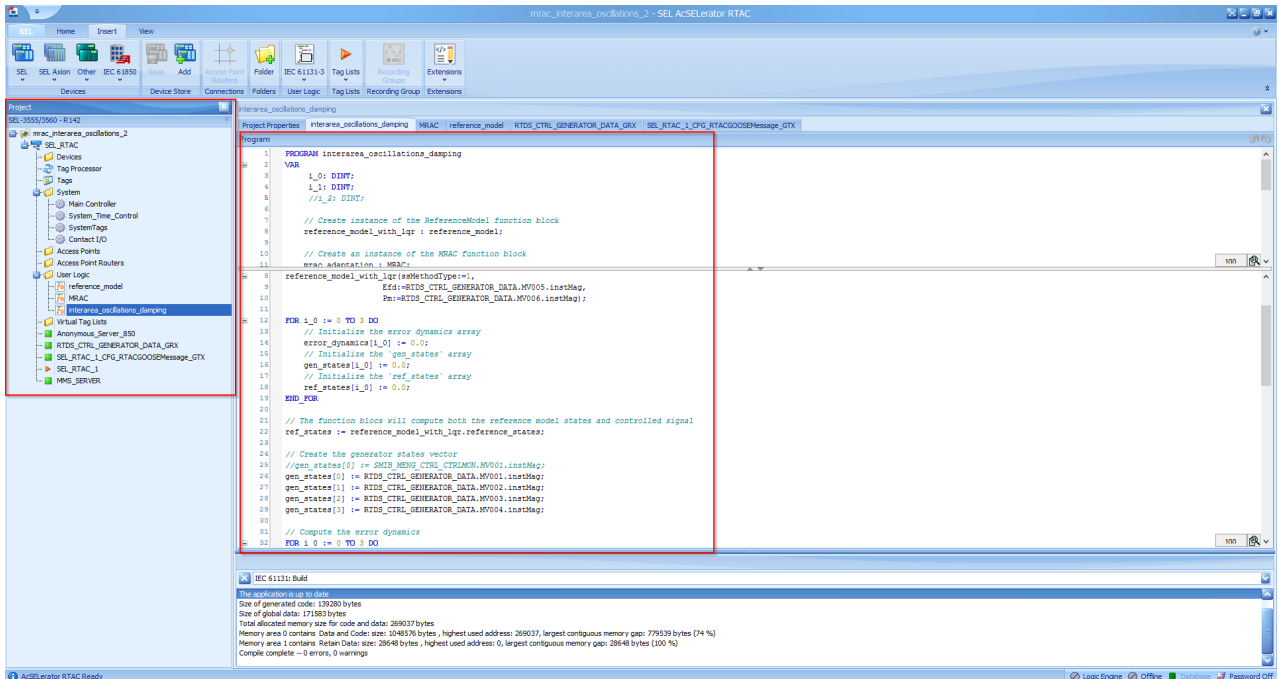
Figure 6.43: Overview of the AcSELerator Project's structure

6.4.6. RTAC SCD File Import to the RSCAD via the SCL Editor

The process of importing the SCD file generated in *Step 5 of sub-section 6.4.5.2 is described in Figure 6.44.*

Figure 6.44: Importing the SEL_RTAC_1 SCD file into the RSCAD SCD Editor

Once imported, map the two analogue inputs to any of the analogue inputs of the GTNET card. Inputs 1, and 2 are chosen.

Figure 6.44 illustrates the process.

183
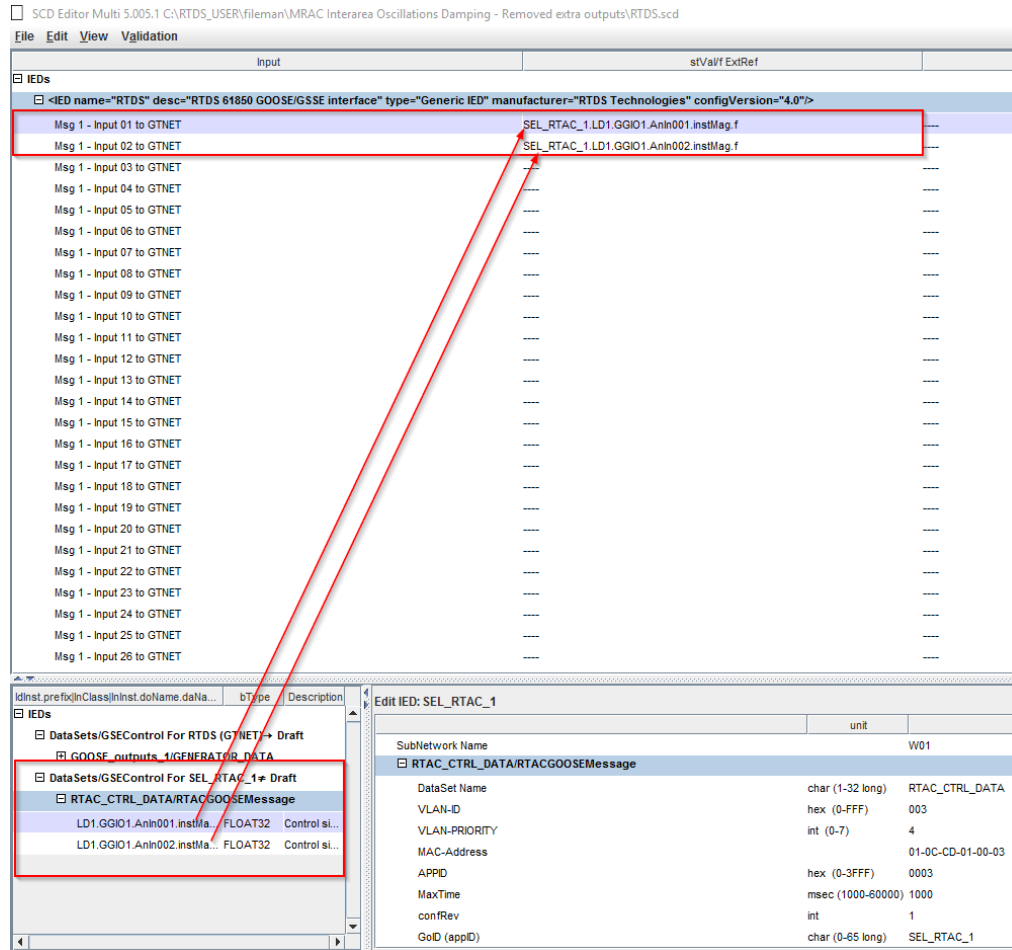
Figure 6.44: **Mapping of the SEL_RTAC_1 analogue GOOSE messages to the RTDS GTNET card**

6.5   Conclusion

Presented and discussed in this chapter are the details of the lab-scale implementation of Model-Reference Adaptive Control (MRAC) testbed for the proposed decentralized power system interarea oscillations damping architecture.

Also presented are the equipment used for the real-time implementation of the scheme. Moreover, various aspect around the configuration of each of these equipment and associated software are presented.

Moreover, the procedure followed in the development of the Hardware-in-the-Loop (HIL) testbed using the IEC 61850 standard is presented and discussed thoroughly.

Chapter 7 presents the results obtained during the application of the proposed control architecture.

CHAPTER SEVEN
HARDWARE IN THE LOOP (HIL) REAL-TIME SIMULATION OF THE CLOSED-LOOP SYSTEM

7.1   Introduction

The implementation of the proposed interarea oscillations damping algorithm presented in the preceding chapters is done in the developed lab-scale testbed for real-time testing and validation. The Model-Reference Adaptive Control (MRAC) presented in Chapter Four is applied to the system as a remedy to the small signal rotor angle stability problem.

The angle instability is a result of Low Frequency Electromechanical Oscillations (LFEOs) and ensuring that countermeasures are in place to damp their effects is highly critical. Moreover, a drift in the angle value between a group of generators in a given area against those in another one leads to interarea oscillations which in turn have devastating consequences within a power system such as voltage collapse, cascade overload, frequency collapse, loss of synchronism, and system separation. While oscillations involving groups of generators within a given area are generally damped accurately by the standard Power System Stabilizers (PSSs) using generators' speed or speed deviation as input, those involving groups of generators in one area swinging against the ones in another area require a control mechanism that insures the stability of the rotor angle.

In this chapter, the real-time implementation of the proposed decentralized MRAC algorithm is presented. The use of Analog Generic Object-Oriented Substation Event (GOOSE) communication in lieu of or Energy Management System or Supervisory Control and Data Acquisition (SCADA) systems is motivated by the 1-5 sec(s) measurement intervals that these systems have which make them unsuitable for any real-time control. The results pertaining to the (real-time) testing of the proposed MRAC interarea oscillations algorithm are presented and discussed henceforward.

The rest of the chapter is structured as follows: Section 7.2 introduces the model used in the real-time implementation of the proposed algorithm. This is the model presented in Figure 6.25 of sub-section 6.4.4 of Chapter 6. Section 7.3 presents the various test cases used for its validation, and Section 7.4 concludes the chapter.

7.2 The Study Network

7.2.1 Overview

As highlighted in Chapter 6, the study network is the Single Machine Infinite Bus power system network modelled in its fourth order model representation. Moreover, it is emphasized in. Chapter 6 that the exported Simulink to RSCAD C-Builder system is modelled to include the nominal LQR controller.

This subsection is structured as follows: after a brief overview of the study network in sub-section 7.2.1, the RSCAD runtime and the monitoring of published GOOSE messages are presented in sub-sections 7.2.2 and 7.2.3. Lastly, sub-section 7.2.4 shows data received and published by the SEL-3555 Real-Time Automation Controller (RTAC).

Figure 7.1 shows the modelled system, and Table 7.1 presents detailed explanations of various components that constitute it.
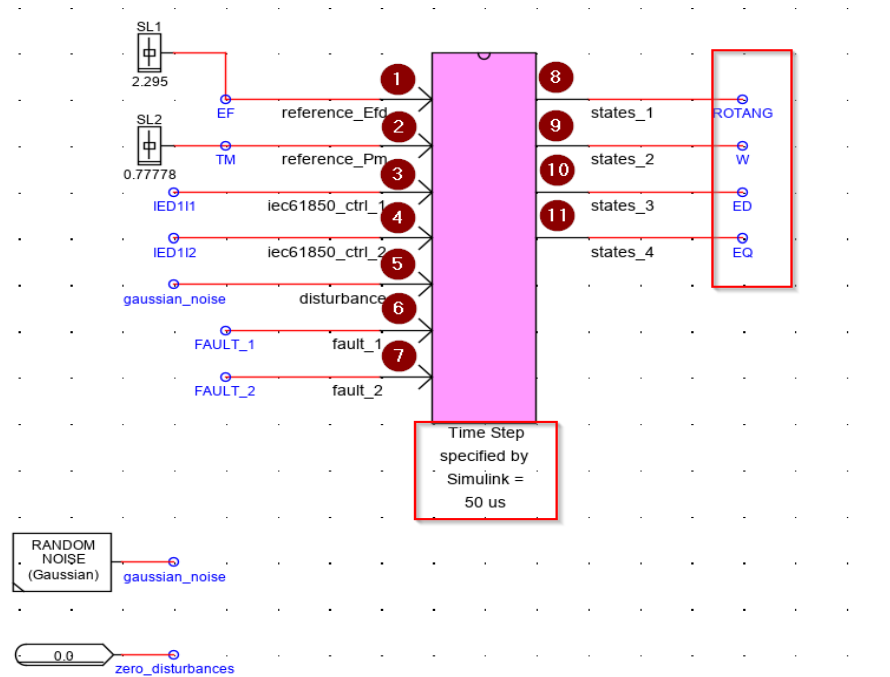
Figure 7.1: Overview of the SMIB study network

Table 7.1: Study network components description

No	Description
1	Reference field voltage. The use of a slider is motivated by the need to be able to change its value on runtime.
2	Reference mechanical power/torque. The use of a slider is motivated by the need to be able to change its value on runtime.
3-4	Computed adaptive control input.
5	Disturbance in the form of Gaussian noise.
6-7	Faults.
8-11	Synchronous generator states i.e., rotor angle, speed, d-axis voltage, and q-axis voltage.

7.2.2 RSCAD Runtime and GOOSE Monitoring of Published Signals

From the study network described in Figure 7.1, the Real-Time Digital Simulator (RTDS) GTNET card is configured such that the reference field voltage, reference mechanical power, as well as the generator states are published as analogue GOOSE messages. The rotor angle which is one of the synchronous generator's states is further configured to be monitored in the front panel of the RSCAD Runtime software environment. Figures 7.2 and 7.3 show the published GOOSE messages from within the GOOSE Inspector environment and an overview of the RSCAD Runtime respectively.
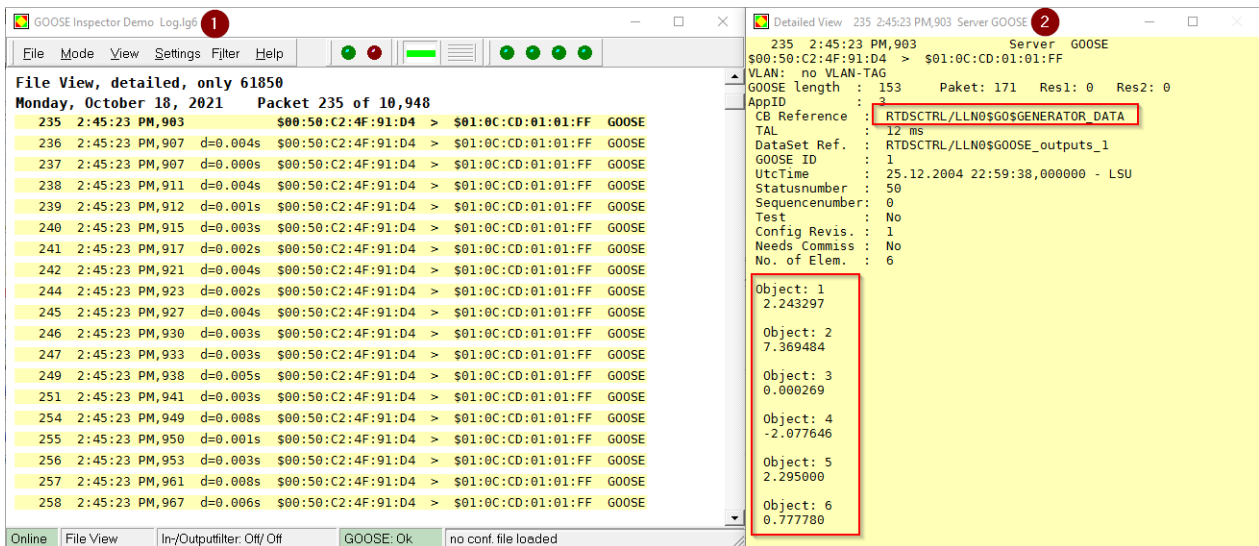
Figure 7.2a: GOOSE Inspector - Overview of published GOOSE messages by the RTDS GTNET card **Figure 7.2b**: Detailed view of published GOOSE messages

From Figure 7.2b, objects 1-6 in the detailed view of the GOOSE inspector software the published signals from the GTNET card are shown. This monitoring can also be achieved using other software packages such as Wireshark.

In order to read and visualize in real-time the rotor angle as the network is subjected to various contingencies but also its behaviour as the field voltage and mechanical power is varied, a plot component is introduced in the RSCAD Runtime environment as shown in Figure 7.3.
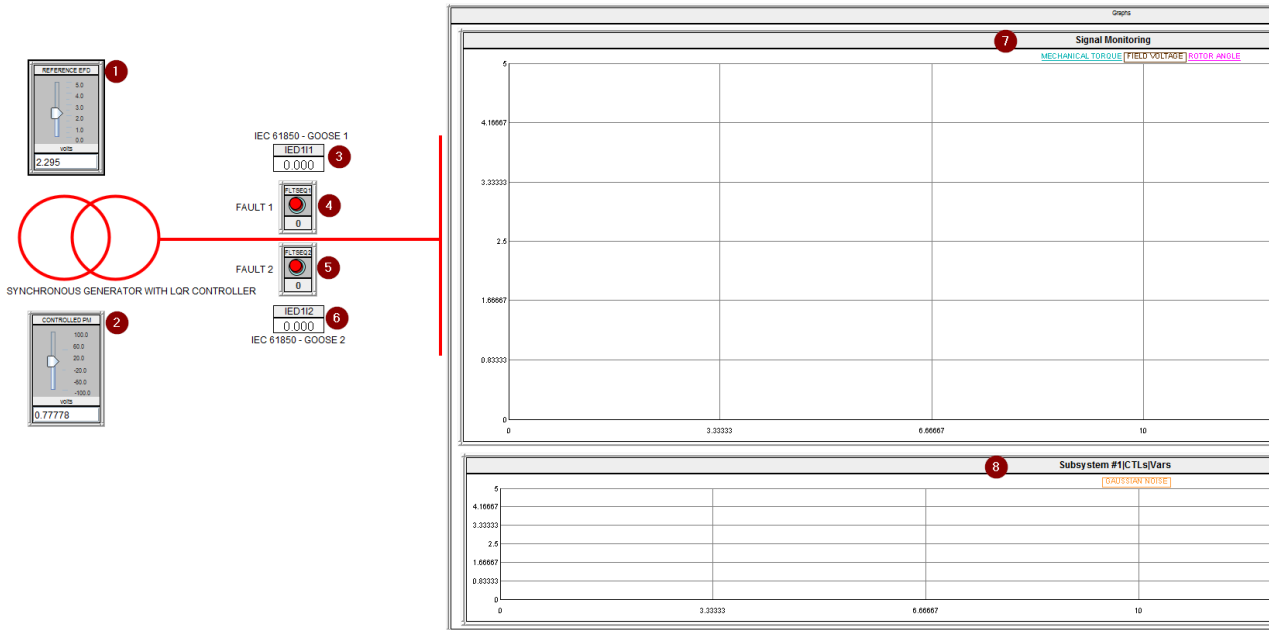
Figure 7.3: RSCAD Runtime – Detailed signal monitoring view

In this figure, (1) and (2) are the equivalent of the sliders introduced in Figure 7.1, (4) and (5) the faults logic presented in Figure 7.4, (3) and (6) the controlled GOOSE messages published by the RTAC as shown in Figure 7.4. The component in (8) is the plot that contains the plots of the mechanical torque, field voltage, and rotor angle as well as the Gaussian noise introduced to emulate Low-Frequency Electromechanical Oscillations (LFEOs) responsible for interarea oscillations.
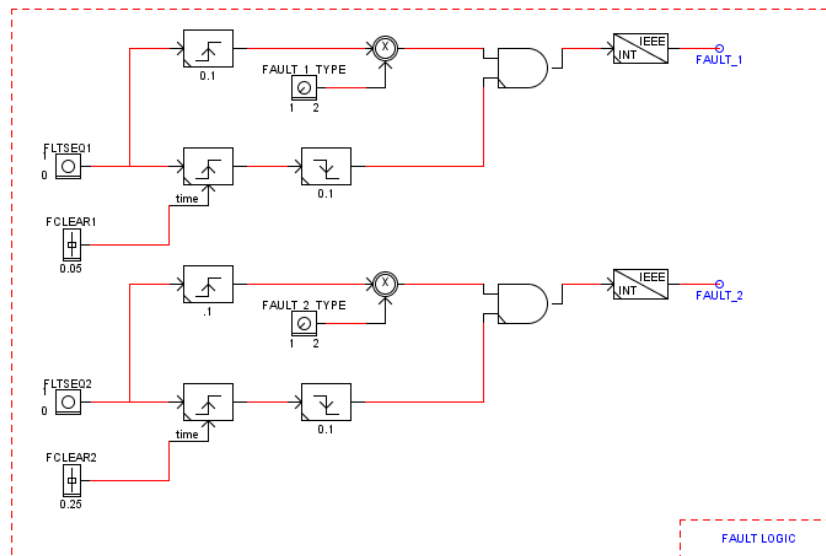
Figure 7.4: Fault logic

188

Table 7.2 gives an insight on components 1-6 from Figure 7.3 while Figure 7.5, which is similar to Figure 6.8 in Chapter 6 shows the analogue GOOSE messages configured to be published by and subscribed from the RTDS GTNET card and the SEL-35555 RTAC and vice-versa.

Table 7.2: Detailed explanation on common components between Figure 7.3a and 7.3b

Component Number	Description
1 - 2	Field voltage and Mechanical Torque/Power inputs. As a result of the use of sliders for both of this input signals, their Runtime versions is also utilised to change their value during simulation as illustrated in Test Cases 2-5, 7-9 and 11.
3	The computed adaptive field voltage sent by the SEL-3555 RTAC using the IEC 61850 standard.
4-5	Rising edge faults introduced to the system to assess the performance of the algorithm for transient stability.
6	The computed adaptive mechanical torque/power sent by the SEL-3555 RTAC using the IEC 61850 standard.
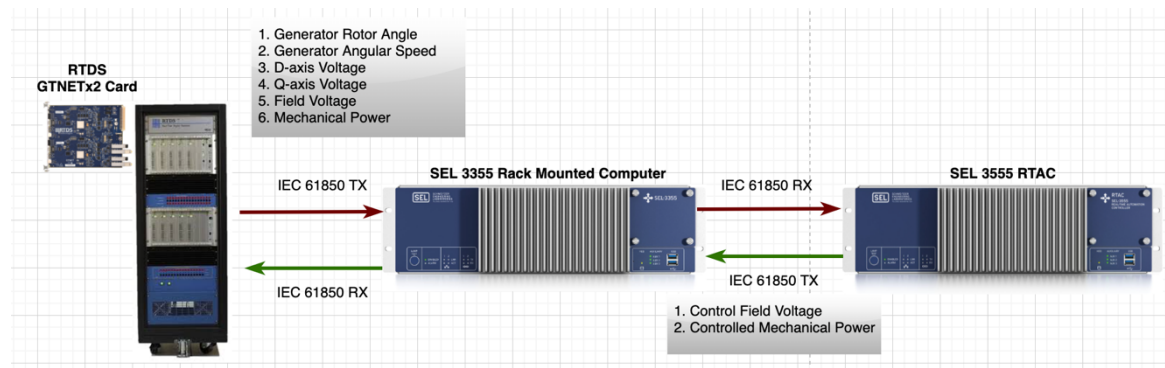
Figure 7.5: Overview of analogue GOOSE messages configured to be published by the RTDS GTNET card and those by the SEL-3555 RTAC.

7.2.3 SEL-3555 RTAC GOOSE Subscription and Publishing

As explained in sub-section 6.4.4 of Chapter 6, the process of subscribing to GOOSE messages and publishing of the controlled signal by the SEL-3555 RTAC follows certain procedures described therein. The AcSELerator RTAC Integrated Development Environment (IDE) is the dedicated computing platform where the control algorithm is implemented. The AcSELerator Architect is configured to allow the SEL-3555 RTAC to subscribe to the published GOOSE messages and publish control signal back to the RTDS. Similar to the GOOSE Inspector, and post completing the AcSELerator Architect software installation and configuration, published data from the RTDS can also be monitored as shown in Figure 7.6.
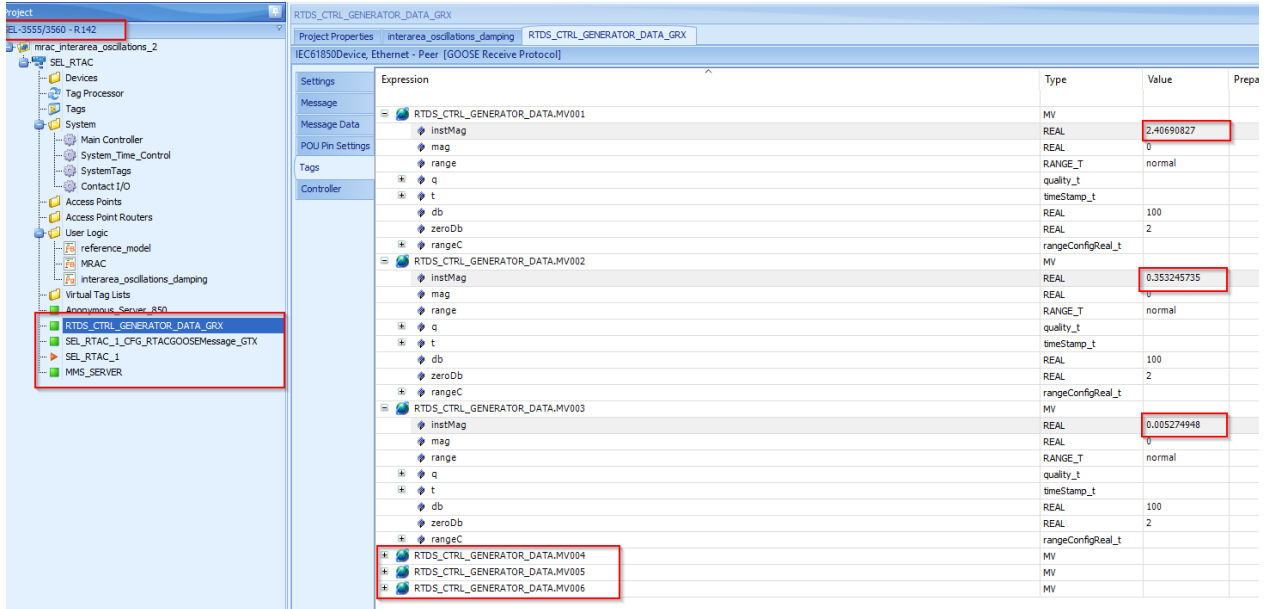
Figure 7.6: SEL-3555 RTAC view of published data from the RTDS

From the above figure, the instantaneous values mapped to tags correspond to the ones shown in Figure 7.2b. These values are the ones used in the development of the control algorithm.

7.3 Implementation of the IEC 61850 Standard-Based MRAC Algorithm
7.3.1 SEL-3555 RTAC and the IEC 61131-3 Standard.

The control algorithm is developed using the IEC 61131-3 standard, with the Structured Text (**ST**) as the programming language. Two functions block namely the *reference_model* and *adaptation* contain, as their names indicate, the code that controls the reference model and the adaptive control component respectively. Furthermore, the *interarea_oscillations_damping* program utilizes both functions to complete the control architecture. Though Appendix A shows the content of each of the aforementioned function blocks, the content of the *interarea_oscillations_damping* is provided below.

```
         IEC 61131-3 STRUCTURED CODE BASED IMPLEMENTATION OF THE MRAC ALGORITHM


         PROGRAM interarea_oscillations_damping
         VAR
                 i_0: DINT;

                 i_1: DINT;


                 // Create instance of the Reference Model function block
```

190

```
            reference_model_with_lqr : reference_model;


        // Create an instance of the MRAC function block
        mrac_adaptation : MRAC;


        // Output arrays of both the reference model and the synchronous generator
model in the RTDS
         ref_states: ARRAY [0..3] OF LREAL;
         gen_states: ARRAY [0..3] OF LREAL;


        // Error dynamics
        error_dynamics: ARRAY [0..3] OF LREAL;


        //Control signal to be sent back to the RTDS via IEC61850
        adaptation: ARRAY [0..1] OF LREAL;
        nominal_controller: ARRAY [0..1] OF LREAL;
        iec61850_control_signal: ARRAY [0..1] OF LREAL;


END_VAR


VAR_TEMP
    i_4: DINT;
END_VAR
```

```
// Call the reference model function block and assign inputs.
reference_model_with_lqr(ssMethodType:=1,
                                Efd:=RTDS_CTRL_GENERATOR_DATA.MV005.instMag,
                                Pm:=RTDS_CTRL_GENERATOR_DATA.MV006.instMag);


FOR i_0 := 0 TO 3 DO
        // Initialize the error dynamics array
        error_dynamics[i_0] := 0.0;
        // Initialize the `gen_states` array
        gen_states[i_0] := 0.0;
        // Initialize the `ref_states` array
        ref_states[i_0] := 0.0;
END_FOR


// The function blocs will compute both the reference model states and controlled
signal
ref_states := reference_model_with_lqr.reference_states;


// Create the generator states vector
gen_states[0] := RTDS_CTRL_GENERATOR_DATA.MV001.instMag;
```

```
gen_states[1] := RTDS_CTRL_GENERATOR_DATA.MV002.instMag;

gen_states[2] := RTDS_CTRL_GENERATOR_DATA.MV003.instMag;

gen_states[3] := RTDS_CTRL_GENERATOR_DATA.MV004.instMag;


// Compute the error dynamics

FOR i_0 := 0 TO 3 DO

        error_dynamics[i_0] := ref_states[i_0] - gen_states[i_0];

END_FOR


// Call the adaptation function block and assign inputs

FOR i_1 := 0 TO 1 DO

        mrac_adaptation(ssMethodType:=i_1,

                                Efd:=RTDS_CTRL_GENERATOR_DATA.MV005.instMag,

                                Pm:=RTDS_CTRL_GENERATOR_DATA.MV006.instMag,

                                generator_states:=gen_states,

                                error_dynamics:=error_dynamics);

END_FOR


// Get the two components of the controlled signal

adaptation := mrac_adaptation.Ua;

nominal_controller := mrac_adaptation.Un;


// Assign controlled data to GOOSE TX tags

SEL_RTAC_1.LD1.GGIO1.AnIn001.instMag := adaptation[0];

SEL_RTAC_1.LD1.GGIO1.AnIn002.instMag := adaptation[1];
```

In the above code snippet, the GOOSE messages published by the RTDS GTNET card as shown in Figure 7.2b are read in real-time through their respective tag names and used while instantiating the function block containing the code that ensures the reference model is optimally controlled as well and the one that contains controller. After initialising the arrays that hold the values of the reference model states, the generator states and the error dynamics, the latter is computed as a difference between the received generator states GOOSE messages and the computed reference model ones. With the value of the error dynamics obtained, the adaptive control function block (*mrac_adaptation*) is called to compute the controlled signals which are in turn mapped to the correct tags. Figure 7.7a and 7.7b show the tag names of the subscribed GOOSE messages as well the ones configured for publishing via the SEL-3555 RTAC.
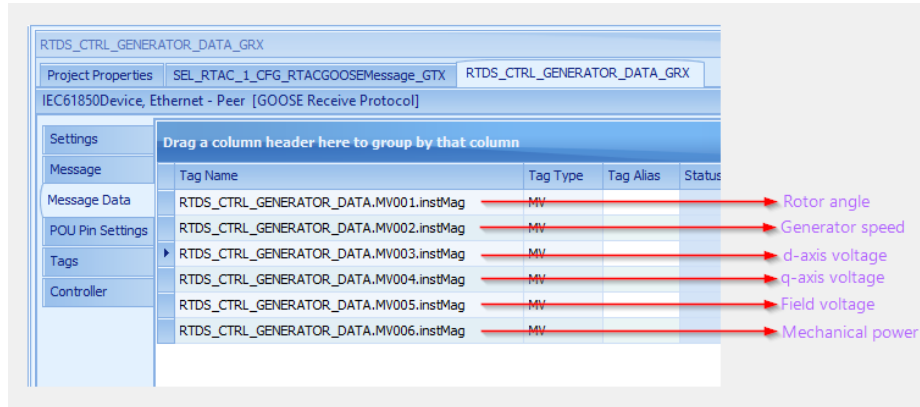
Figure 7.7a: SEL-3555 RTAC Subscribed Analogue GOOSE Messages Tags
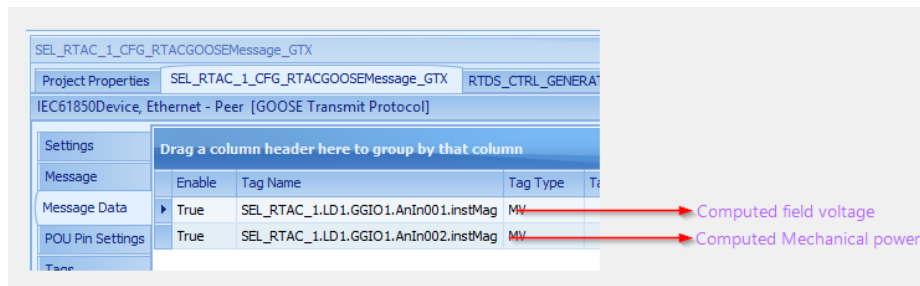
Figure 7.7b: SEL-3555 RTAC published analogue GOOSE message tag names

7.3.2   Practical Implementation of the MRAC Algorithm Using the SEL-3555 RTAC and the RTDS

7.3.2.1 Overview

To validate the proposed MRAC algorithm, the Single Machine Infinite Bus (SMIB) power system network modelled in its 4th order which include an optimal Linear Quadratic Regulator (LQR) as shown in Figure 7.1 is simulated in real-time in the RTDS through the RSCAD software. Using the HIL configuration illustrated in Figure 7.4, its performance is assessed and validated through the following test cases.

7.3.2.2 Case Study 1

The first case study carried out involves introducing a 100ms fault. The initial values for the mechanical torque and field voltage are 0.77778 pu and 2.295 pu respectively. The rotor behaviour value over a 20 second window of observation is shown in Figure 7.8.
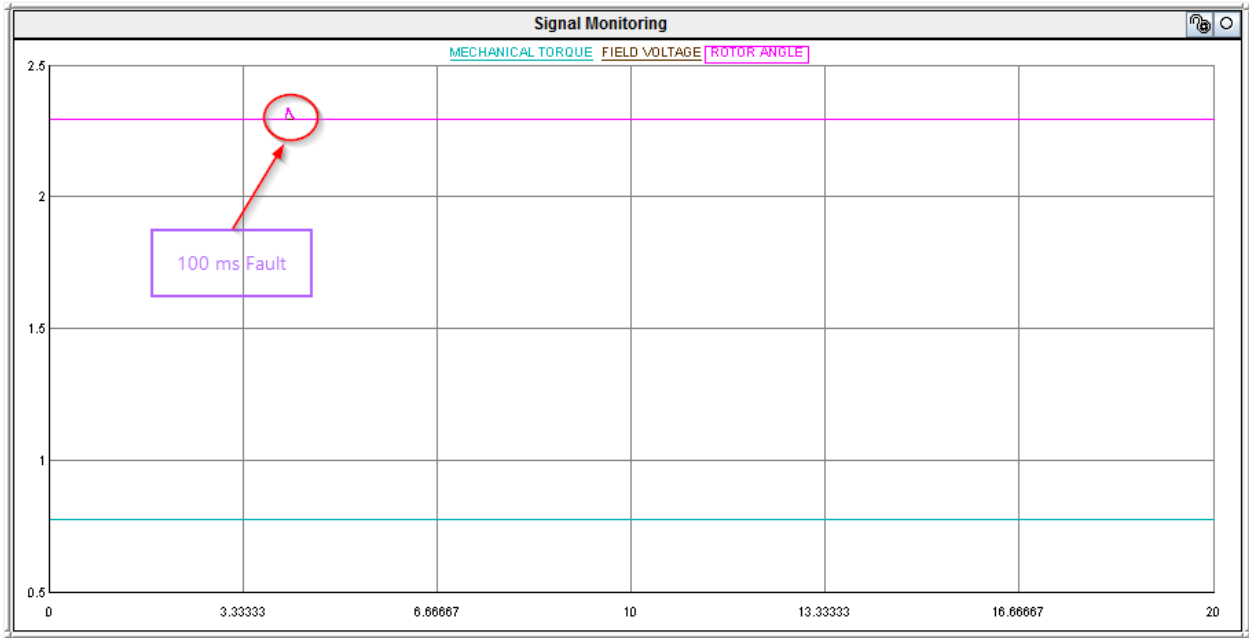
Figure 7.8: Synchronous generator rotor angle when the network is subjected to a 100s fault

7.3.2.3 Case Study 2

In the second case study, the field voltage's initial value is dropped by 5% while the mechanical torque remains at 0.77778 pu. The rotor angle behaviour over a 20 second window of observation is shown in Figure 7.9.
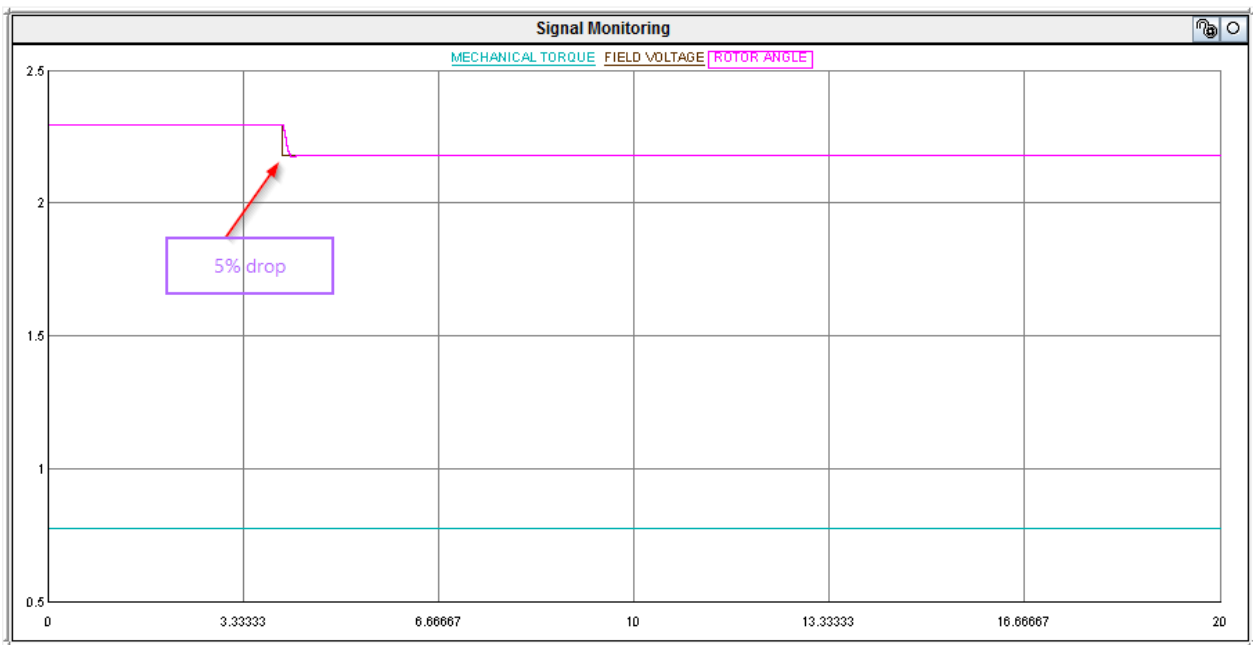
Figure 7.9: Synchronous generator rotor angle when the field voltage value is drop by 5%

7.3.2.4 Case Study 3

From the previous 5% drop, the field voltage value is increased by 10% in this case study, with the mechanical torque remains at 0.77778 pu. The rotor angle behaviour over a 20 second window of observation is shown in Figure 7.10
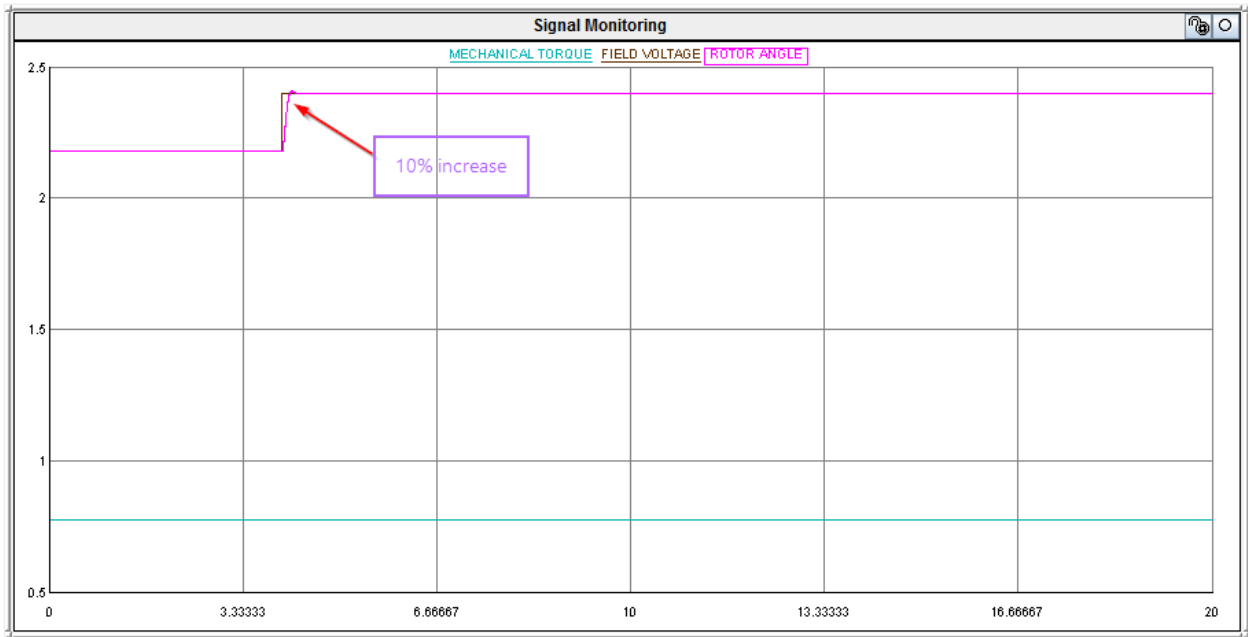
Figure 7.10: Synchronous generator rotor angle when the field voltage value is increased by 10% from the previous 5% drop

7.3.2.5 Case Study 4

In this case study, the field voltage is further increase by another 3% while the mechanical torque remains at 0.77778 pu. The rotor angle's behaviour over a 20 second window of observation is shown in Figure 7.11.
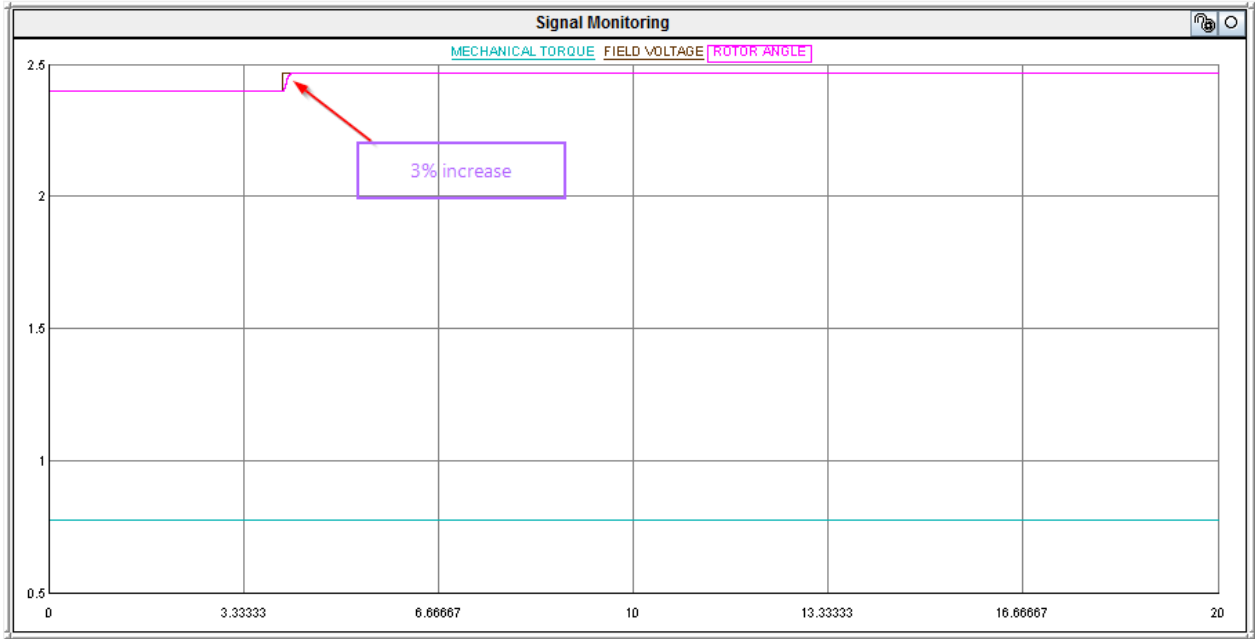
Figure 7.11: Synchronous generator rotor angle when the field voltage value is increased by 3% from the previous 10% increase

7.3.2.6 Case Study 5

In contrast with the first 4 test cases, this one focused on analysing the behaviour of the synchronous generator rotor angle as the mechanical torque input varies. Similar to the digital simulation results presented in Chapter 5, the mechanical torque did impact the rotor angle as it remained unchanged from the 3% increase from case study 5.

The rotor angle behaviour over a 20 second window of observation when the mechanical torque is increased by 10% from its initial value is shown in Figure 7.12.
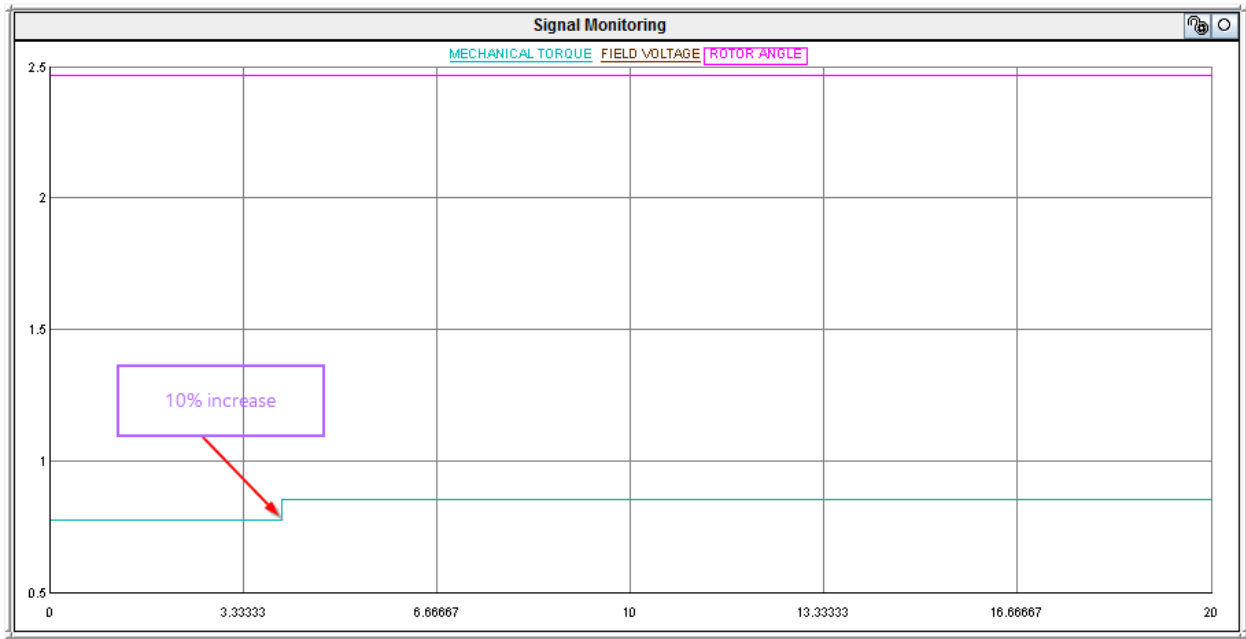
Figure 7.12: Synchronous generator rotor angle when the mechanical torque's value is increased by 10% from its initial 0.77778 pu value

7.3.2.7 Case Study 6

Small variations in the system loads within a power system are the root cause of Low-Frequency Electromechanical Oscillations (LFEOs). These inherent oscillations, with a frequency ranging between 0.2-0.8 Hz, are responsible for the small signal rotor angle stability problem. This is because they cause the rotor angle to oscillate thus the use of Power System Stabilizers (PSSs) to mitigate their impact. However, with weak tie-line and long distances between generating units and consumers because of the ever-growing electricity market, they may result in a group of generators in one area oscillating against those in another area group if not well damped.

From this case study to the last, the network will be subjected to Gaussian noise while the performance of the MRAC algorithm is assessed by observing the synchronous generator rotor angle over a 20 second window.

Figure 7.13 shows the rotor angle as it is monitored over a 20 second window while the network is subjected to a Gaussian noise.
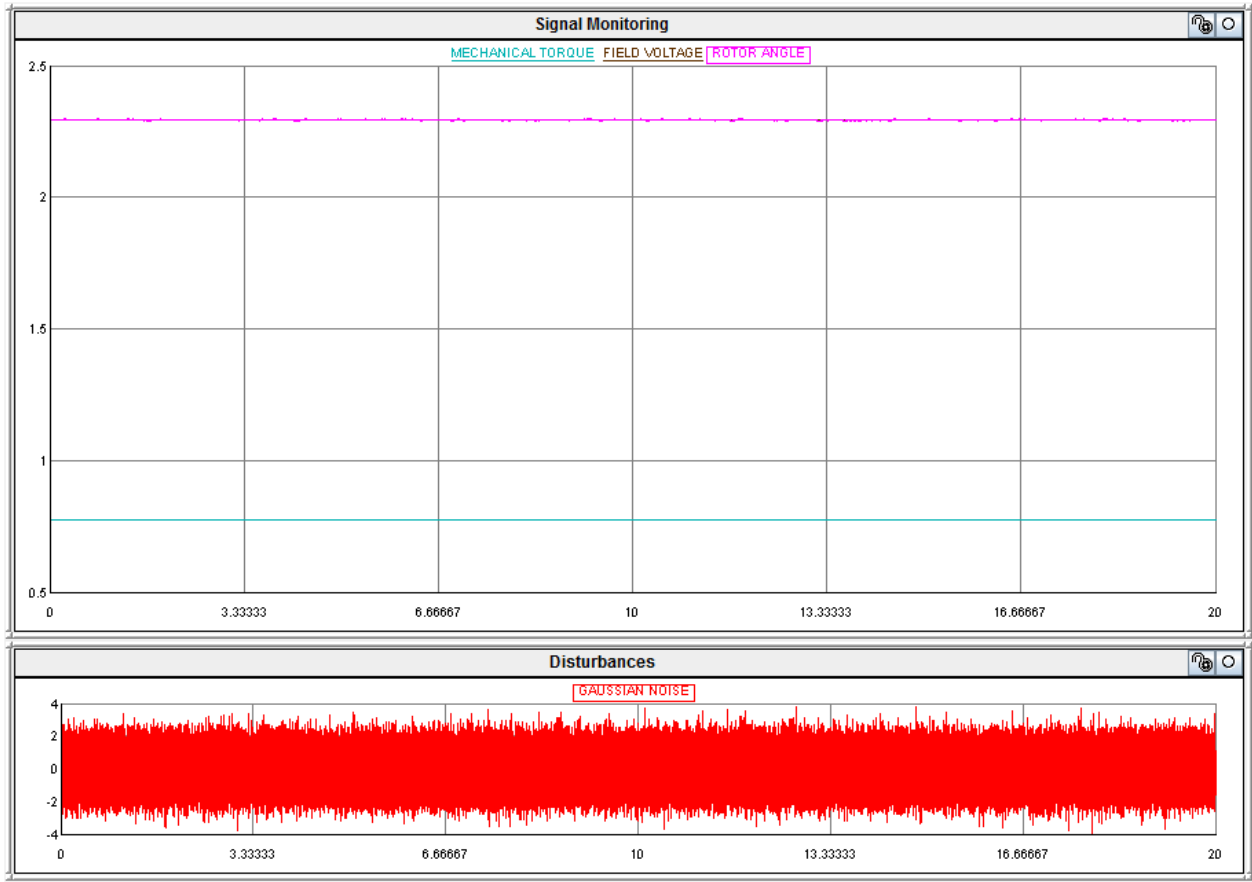
Figure 7.13: Synchronous generator rotor angle when system is subjected to Gaussian noise, and with both the field voltage and mechanical torque to their initial 2.295 pu and 0.77778 pu values.

7.3.2.8 Case Study 7

In this test case, the field voltage is dropped by 5% while the system is subjected to Gaussian noise. The mechanical torque is kept at its initial 0.77778 pu value. The rotor angle's behaviour as it is monitored over a 20 second window of observation is shown in Figure 7.14.
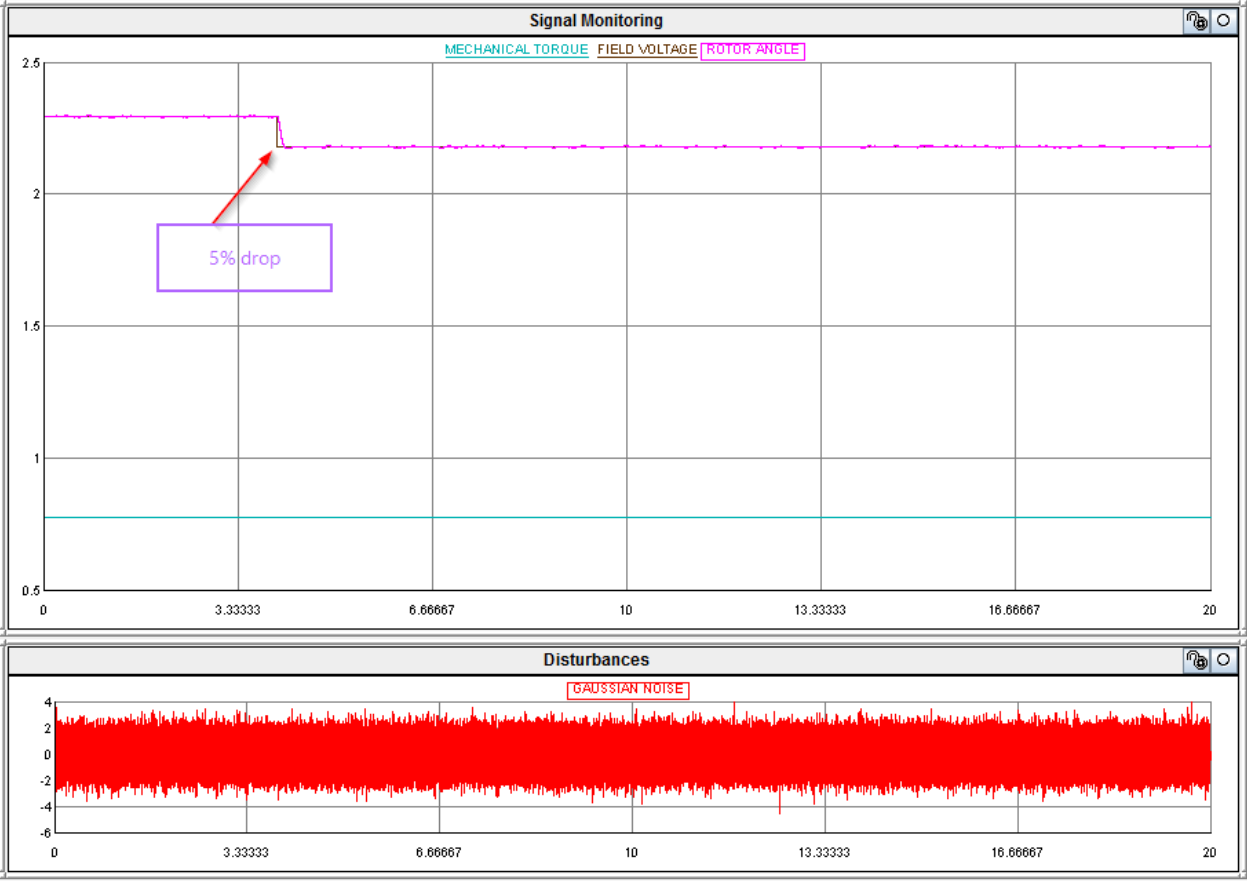
Figure 7.14: Synchronous generator rotor angle when system is subjected to Gaussian noise, and a 5% drop of the field voltage initial value. The mechanical torque is kept at its initial value of 0.77778 pu

7.3.2.9 Case Study 8

While still subjected to Gaussian noise, the field voltage is increased by 10% from the previous 5% drop. The rotor angle's behaviour as it is monitored over a 20 second window of observation is shown in Figure 7.15.
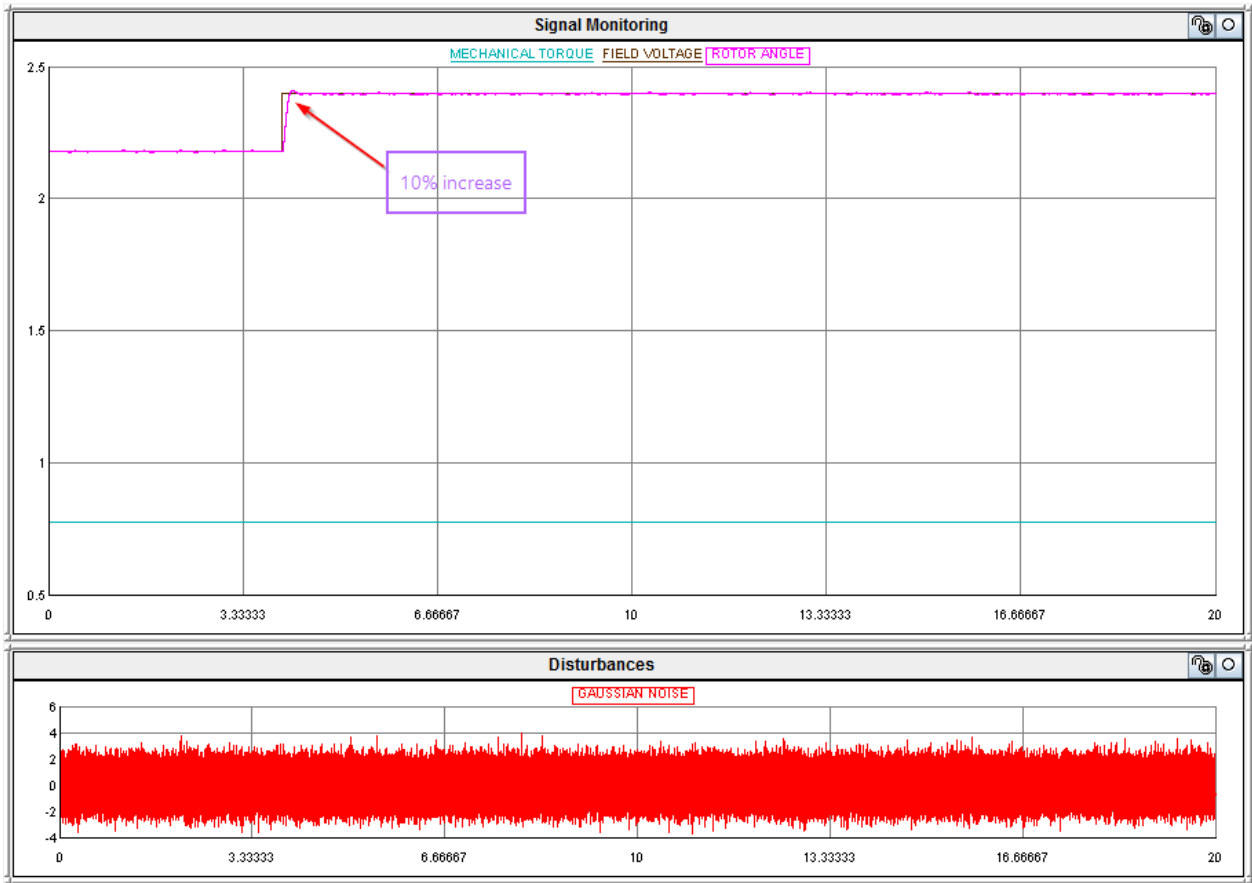
Figure 7.15: Synchronous generator rotor angle when system is subjected to Gaussian noise, and a 10% increase of the field voltage from the previous 5% drop. The mechanical torque is kept at its initial value of 0.77778 pu

7.3.2.10 Case Study 9

In this case study, a further 3% increase is applied to the field voltage with the system still subjected to Gaussian noise and the mechanical torque unchanged. The rotor angle's behaviour as it is monitored over a 20 second window of observation is shown in Figure 7.16.
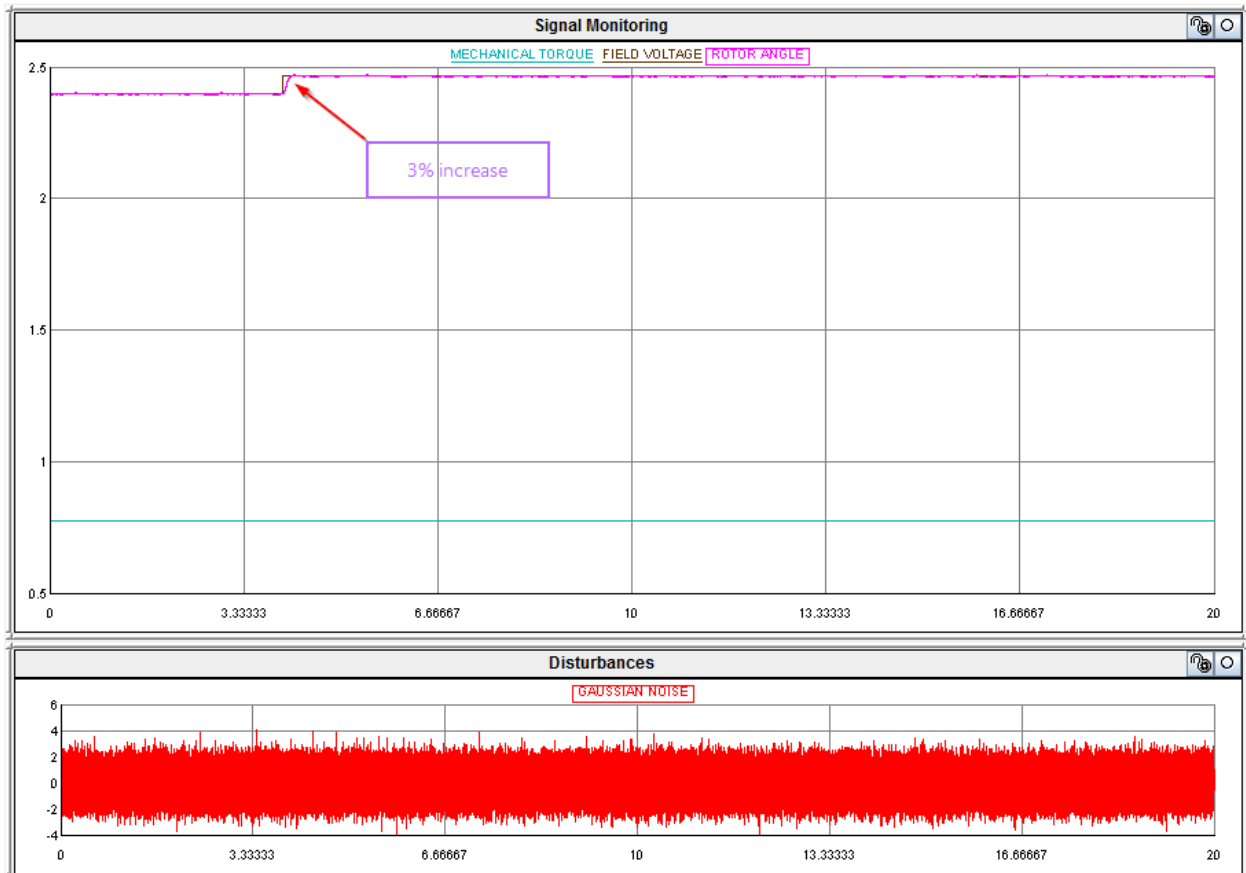
Figure 7.16: Synchronous generator rotor angle when the system is subjected to Gaussian noise, and a further 3% increase of the field voltage from the previous 10% increase. The mechanical torque is kept at its initial value of 0.77778 pu

7.3.2.11 Case Study 10

With the system still subjected to Gaussian noise, a 100 ms fault is further applied. The field voltage is kept to its value from test case 9 and the mechanical torque still unchanged. The rotor angle's behaviour as it is monitored over a 20 second window of observation is shown in Figure 7.17
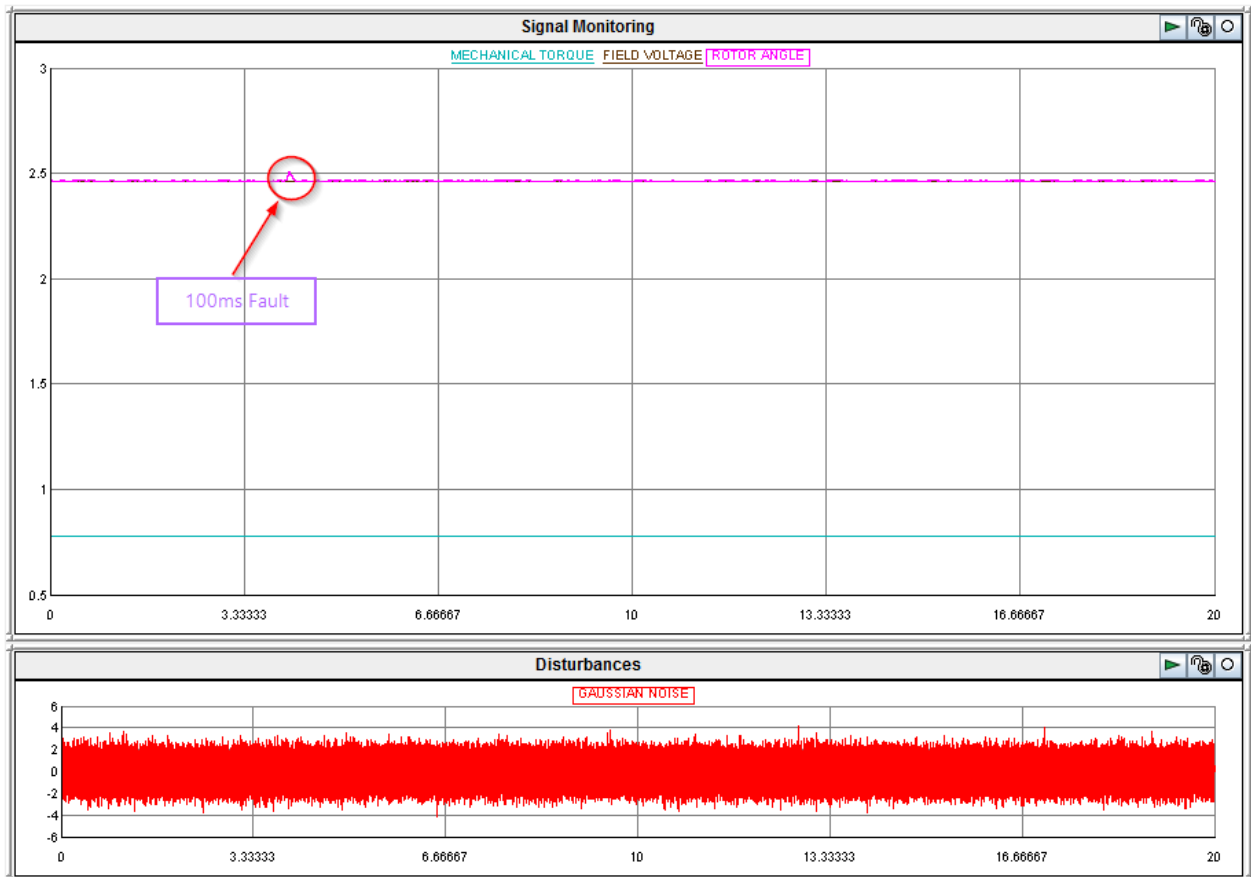
Figure 7.17: Synchronous generator rotor angle when the system is subjected to Gaussian noise, and a 100 ms fault. The field voltage is kept at its value from test case 9 while the mechanical torque is kept at its initial value of 0.77778 pu

7.3.2.12 Case Study 11

Similar to case study 5, the mechanical torque is increased by 10% from its initial value while the system is still subjected to Gaussian noise. The rotor angle's behaviour as it is monitored over a 20 second window of observation when the mechanical torque is increased by 10% from its initial value is shown in Figure 7.18.
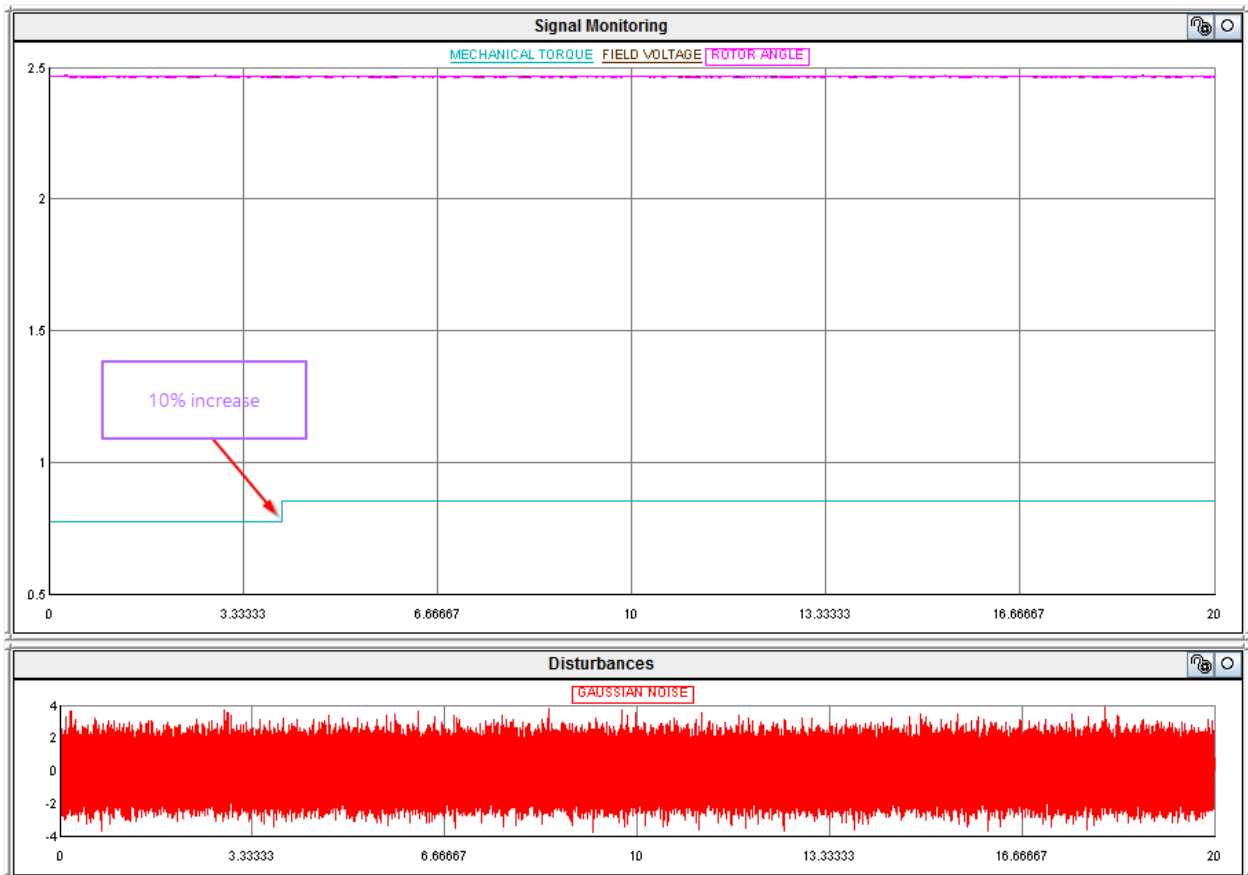
Figure 7.18: **Synchronous generator rotor angle when the system is subjected to Gaussian noise, and the mechanical torque increased by 10% from its initial 0.77778 pu value. The field voltage is kept at its value from test case 9.**

7.3.3 Discussions

The MRAC based algorithm proposed in Chapter 4 to mitigate the effects of interarea oscillations is tested and validated in a HIL testbed consisting of the SEL-3555 RTAC, and the RTDS and the SEL-3355 rugged industrial computer. Using the IEC 61850 standard, GOOSE messages are exchanged between the real-time controller (RTAC) and the RTDS GTNET card. In the RTAC, the computation is done using the IEC 61131-3 standard, with the Structured Text (ST) used as the preferred programming language.

Similar to the results presented during the digital simulation with MATLAB in Chapter 5, the real-time implementation of the proposed algorithm has confirmed its robustness. Though initially aimed at mitigating the effects of LFEOs, this adaptive controller has shown prospect of being utilized and perhaps expanded to be used for transient stability enhancement as well. This is illustrated in test cases 1 and 10 respectively. Moreover, irrespective of the disturbance introduced, the rotor angle remained stable.

The performance of the MRAC algorithm is further verified by comparing the response, of the digital simulations with those of the real-time implementation under similar conditions. Two case studies were selected: one where both are subjected to Gaussian noise solely and the other when a fault is added onto the current disturbance as in Figures 7.19a, 7.19b, 7.20a, and 7.20b. A summary of the results in Figures 7.19 and 7.20 are presented in Table 7.3.
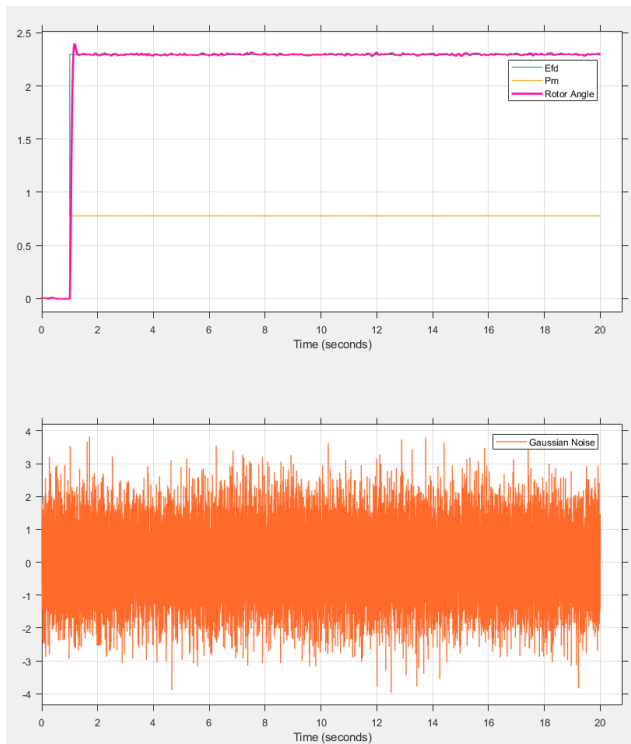
Figure 7.19a: Rotor angle's response during simulation with SIMULINK when the synchronous generator is subjected to Gaussian noise from t=0s, for the duration of the simulation. The field voltage when $E_{FD} = 1$ and the mechanical power, $P_m = 0.77778$
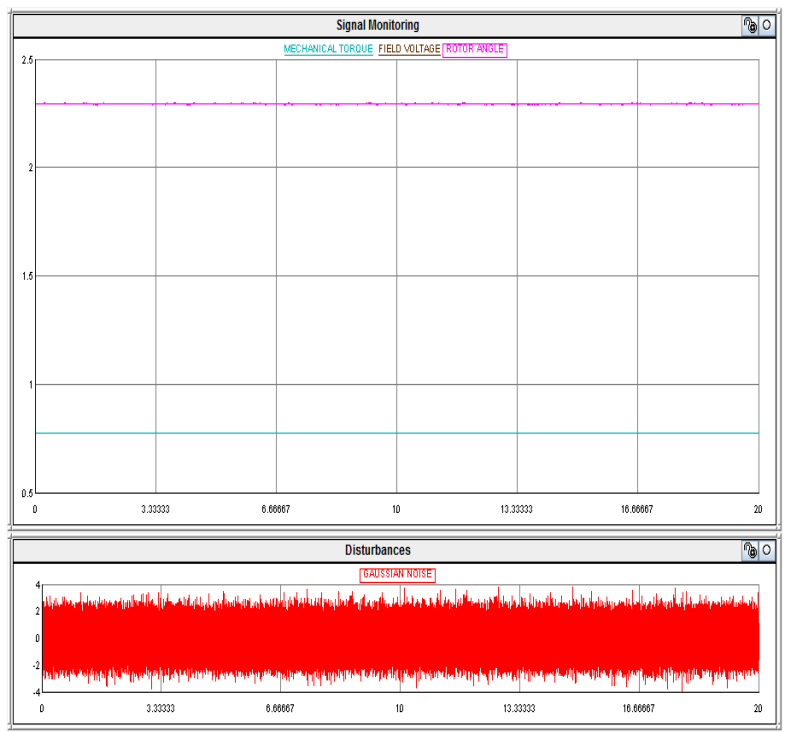
Figure 7.19b: Real-time rotor angle's response when the synchronous generator is subjected to Gaussian noise. The field voltage 2.295 pu and the mechanical torque is at 0.77778 pu
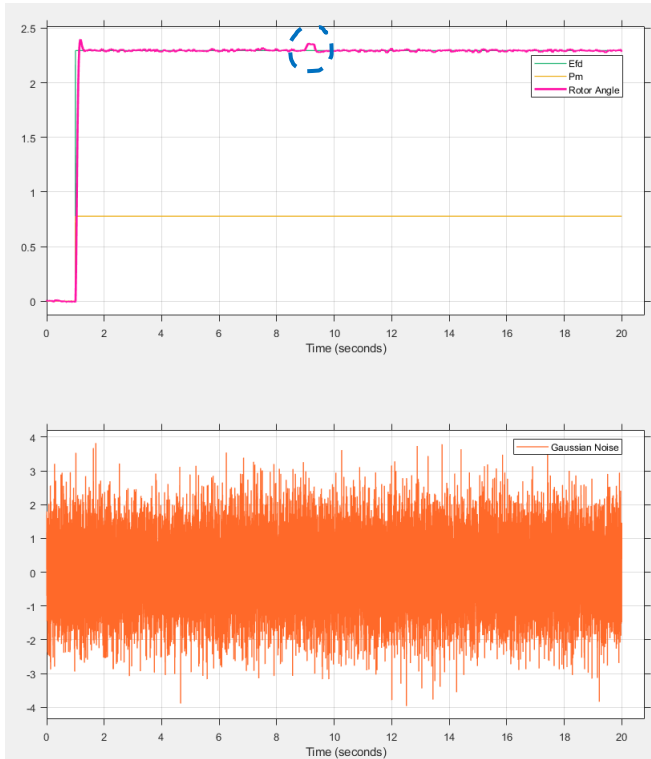
Figure 7.20a: Rotor angle's response during simulation with SIMULINK when the synchronous generator is subjected to Gaussian noise from t=0s for the duration of the simulation as well as a 100ms fault in the form of impulse. The field voltage when $E_{FD} = 1$ and the mechanical power, $P_m = 0.77778$
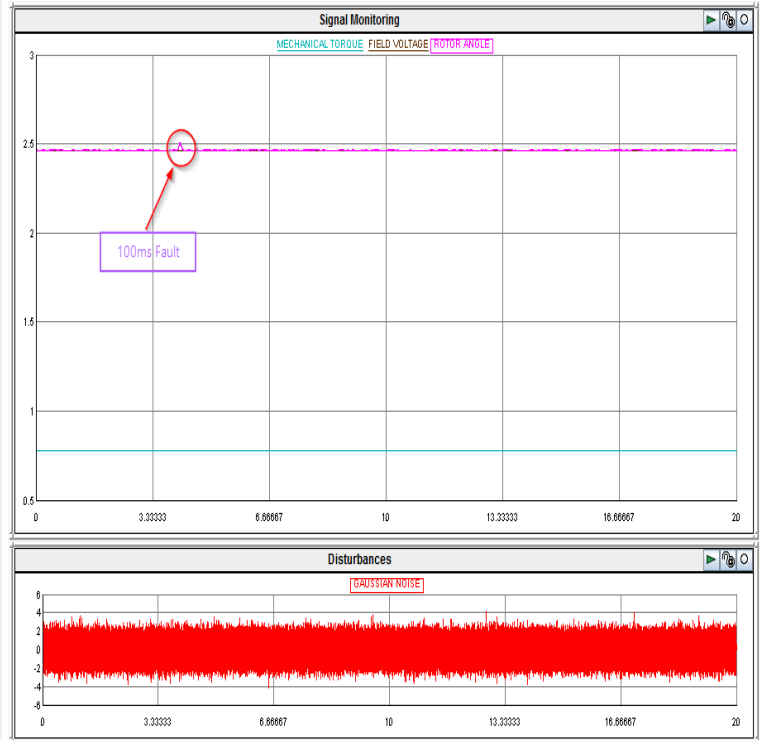
Figure 7.20b: Real-time rotor angle's response when the synchronous generator is subjected to Gaussian noise and a 100 ms fault. The field voltage 2.295 pu and the mechanical torque is at 0.77778 pu

Table 7.3: Performance evaluation of the MRAC with SIMULINK vs real-time implementation

| | Setpoints | | Disturbances | | Rotor angle characteristics | |
	E_{FD}	P_m	Inherent	Fault	Steady-state error (%)	Recovery time (ms)
Figure 7.19a	2.295	0.77778	Gaussian noise	0	N/A	N/A
Figure 7.19b	2.295 pu	0.77778 pu	Gaussian noise	0	N/A	N/A
Figure 7.20a	2.295	0.77778	Gaussian	100 ms	N/A	~ 54
Figure 7.20b	2.295 pu	0.77778 pu	Gaussian noise	100ms	N/A	~ 42

As shown in Table 7.3, the results obtained in a real-time HIL setup align with those presented earlier in chapter 5.

The development and implementation of the IEC 61850 HIL testbed is achieved while its performance validated the theoretical developments presented in Chapter 4 as well as the results presented in Chapter 5 based on digital simulations in MATLAB.

7.6   Conclusions

The proposed Model-Reference Adaptive Control algorithm for power system interarea oscillations is tested using real-time Hardware-in-the-Loop (HIL) simulations with the developed testbed. The results presented for the real-time interarea oscillations damping through the rotor angle stability enhancement demonstrated the robustness of the proposed algorithm

Irrespective of the contingencies, drop in the value of the field voltage, its increase, increase in the mechanical torque value, nor the introduction of Gaussian noise did affect the performance of this algorithm. Furthermore, its application for transient stability is also investigated. This is shown in case studies 1 and 10.

The rotor angle remaining stable, the proposed IEC 61850 based MRAC algorithm is shown to be suitable for real-time application.

Chapter 8 concludes this thesis and presents its deliverables. Moreover, it highlights a few recommendations for future work. Lastly, the publication outputs related to this thesis are presented.

CHAPTER EIGHT
CONCLUSIONS AND RECOMMENDATIONS

8.1 Introduction

Power system networks operate closer to their capacity as a consequence of current liberalized electricity framework as well as constrained transmission networks. Considering the long distances that separate generating units as well as the weak tie-lines that link them, they become prone to the small signal stability problem because of Low-Frequency Electromechanical Oscillations (LFEOs) that are often not well damped. Thus, the need for mitigating strategies to ensure the synchronous generator rotor angle remains stable hence the overall power system network. This is essential for preventing blackout or system collapse when these oscillations involve groups of generators in one area oscillating against those in another area.

This thesis proposes an IEC 61850 based Model Reference Adaptive Control (MRAC) algorithm that ensures the synchronous generator rotor angle remains stable when subjected to such disturbances (0.2-0.8 Hz). Moreover, it has been shown that the proposed scheme can also be used to some extent for transient stability enhancement.

The deliverables and conclusions of this thesis are presented herein, with section 8.2 presenting the aim and objectives as introduced in Chapter 1. Section 8.3 provides the deliverables and achieved objectives, and Section 8.4 its application in academia and industry. Recommendations for future work as well as publications around this research are presented in Section 8.5 and 8.6 respectively. Lastly, Section 8.7 concludes this work.

8.2 Aim And Objectives
8.2.1 Aim of the Thesis

The aim of this research is to design, develop and implement an adaptive algorithm for power system interarea oscillations.

8.2.2 Objectives

The aim is attained through theoretical derivations and practical implementation

8.2.2.1 Objectives – Theoretical Analysis

- To review methods used for power system interarea oscillations analysis, and oscillations damping.

- To review on existing methods and algorithms for interarea damping controllers and stability improvement.

- To formulate a Model Reference Adaptive Control (MRAC) scheme based on the error dynamics to enhance the stability of the synchronous generator rotor angle.

- To design an MRAC algorithm using the IEC 61850-8-1 Generic Object-Oriented Substation Event (GOOSES) messages.

8.2.2.2 Objectives – Practical Real-Time Implementation

- Using digital simulation in the MATLAB environment, to design and implement the proposed interarea oscillations damping algorithm.

- Validate the performance of the control scheme through case studies.

- Conversion of the 4th order Single-Machine Infinite Bus (SMIB) Simulink model to C/C++ Code.

- Modelling of the converted model in the RSCAD software.

- Conversion of the reference model as well as the adaptation and controller to an IEC 61131-3 compatible code via the SIMULINK PLC coder.

- Implementation of the proposed algorithm for real-time interarea oscillations damping using external hardware interfaced to the Real-Time Digital Simulator (RTDS).

- Development of the proposed algorithm in the Schweitzer Engineering Laboratories (SEL) 3555 Real-Time Automation Controller (RTAC).

- Real-time lab-scale implementation of an IEC 61850 based MRAC algorithm using the RTDS and SEL-3555 RTAC in a Hardware-in-the-Loop (HIL).

8.3   Thesis Deliverables
This thesis deliverables are further elucidated in the following sections.

8.3.1   Literature Review
An ample literature review was conducted on power system stability with the focus on rotor angle stability. The review also covered aspects of estimation methods for oscillations characteristics as well as approaches to mitigate instability resulting from poorly damped Low-Frequency Electromechanical Oscillations (LFEOs); with the latter being the root cause of interarea oscillations. Two different type architectures are discussed namely the centralized and decentralized ones in Chapter 2.

This thesis considered the latter in ensuring that the stability of the rotor angle is maintained irrespective of the contingencies.

8.3.2 Synchronous Generator Dynamics

The equations describing the dynamics of a synchronous generator were presented, specifically the 3rd and 4th order models. The latter was preferred in the design of the Model-Reference Adaptive Control algorithm as it encompasses the first and is suitable to model the synchronous generator in full range of electromechanical oscillations i.e., local, and interarea oscillations.

Both equations are derived from the Single-Machine Infinite Bus (SMIB) setup where a synchronous generator is connected to an infinite bus.

Moreover, the word "system" mentioned in the subsequent sub-sections refers to the SMIB power system network

8.3.3 Theoretical Aspects of the Nonlinear Reference Model Based Controller Small Signal Rotor Angle Stability Enhancement

A comprehensive overview of stability concepts in general and the Lyapunov stability theory were presented. Using these concepts, a nonlinear servo-based reference-model controller was proposed for the synchronous generator represented by its 3rd order model to enhance the stability of the rotor angle.

8.3.4 Theoretical Aspects of the Model-Reference Adaptive Controller (MRAC) and its Application for Small Signal Rotor Angle Stability Enhancement

Using the Quadratic Optimal Regulator Systems designs concepts together with the Model-Reference Adaptive Control (MRAC) theories, an MRAC-based controller was developed and proposed for the 4th order model of the synchronous generator to enhance the rotor angle stability.

8.3.5 Digital Simulations of the Nonlinear Reference Model Based Controller in the MATLAB Environment

The performance of the controller mentioned in 8.3.3 was evaluated through digital Simulation in the MATLAB/SIMULINK environment. Various case studies were considered, and it was shown that the rotor angle stability could not be guaranteed when subjected to contingencies such as setpoints changes for instance.

8.3.6 Digital Simulations of the Model-Reference Adaptive Controller (MRAC) in the MATLAB Environment

Similar to the nonlinear servo-based reference-model controller, the MRAC controller highlighted in 8.3.4 was evaluated through digital simulation in the MATLAB/SIMULINK environment. But unlike the first, contingencies such as setpoints change did not impact the stability of the rotor angle. Furthermore, the introduction of disturbances such as Gaussian noise to emulate Low-Frequency Electromechanical Oscillations or impulses to emulate faults did not perturb its stability either.

8.3.7 Real-Time Implementation of the Model-Reference Adaptive Controller (MRAC) Algorithm in a Hardware-in-the-Loop (HIL) Testbed

For real-time modelling in a Hardware-in-the-system, certain components of the system modelled in 8.3.6 had to be used for C/C++ and IEC 61131-3 code generation. The first pertains to the synchronous generator which was imported into the RSCAD C-Builder, while the latter comprised of the reference model with its controller and the adaptive component of the MRAC used in the implementation of the algorithm in the SEL-3555 Real-Time Automation Controller (RTAC).

8.3.7.1 C/C++ Code Generation for the Synchronous Generator

SIMULINK was configured such that C/C++ code generation becomes possible as illustrated in sub-section 6.4.3.1 of Chapter 6. As explained therein, certain aspects/procedures had to be followed for a successful code generation but also to ensure that it can be imported and compiled in the RSCAD C-Builder.

8.3.7.2 IEC 61131-3 Compatible Code Generation of the Reference Model and Controller

Similar to the process in 8.3.7.1, SIMULINK had to be configured to allow Structured Text (ST) code generation for the reference model together with its LQR controller, as well as the adaptation component of the MRAC.

8.3.7.3 RSCAD Modelling of the Imported C-Builder Model

The SIMULINK to C-Builder model of the system mentioned in 8.3.7.1 was modelled in real-time using the RSCAD software.

8.3.7.4 RTDS GTNET Card Configuration for GOOSE Publishing

Communication between the SEL-3555 RTAC and the RTDS was achieved through Generic Object-Oriented Substation Events (GOOSE) messages; with the rugged SEL-

3355 industrial computer as the computing platform. Therefore, the RTDS GTNET card had to be configured for GOOSE message publishing with parameters of interest such as the synchronous generator's states and setpoints (or inputs) published over the network using the IEC 61850 communication standard.

8.3.7.5 SEL-3555 RTAC Configuration For GOOSE Subscription and Publishing

The real-time implementation of the control algorithm required an industrial-grade computing device that is suitable for the task. Combining the best features of the x86-64 architectures, embedded microcomputers, embedded real-time operating systems, as well as secure communications framework, the SEL-3555 RTAC is reputed to be one of the best automation platforms. Hence its use for this thesis. The AcSELerator Architect software was configured and utilized such that published GOOSE messages from the RTDS GTNET card are subscribed by the SEL-3555 RTAC and those by the latter are subscribed by the first.

8.3.7.6 MRAC Algorithm Implementation in the SEL-3555 RTAC

Two Function Blocks namely *reference_model.st* and *adaptation.st* together with a Program (*interarea_oscillations_damping.st*) were setup for the real-time implementation of the proposed algorithm. While the main program was introduced in sub-section 7.3.1 of Chapter 7, the two function blocks' content are given in APPENDIX 2 and 3 respectively.

8.3.7.7 Real-Time Lab-Scale Model-Reference Adaptive Controller (MRAC) Algorithm Testbed

The proposed IEC 61850 based MRAC algorithm was tested and validated in real-time through a lab-scale Hardware-in-the-Loop (HIL) testbed presented in Chapter 6 and implemented in Chapter 7. Industrial-grade equipment such as the RTDS®, SEL-2725 Ethernet Switch, SEL-3355 Rugged Computer, and the SEL-3555 RTAC were utilized in the implementation of the proposed algorithm.

The power system model used in the validation of the proposed algorithm was introduced and utilised for real-time HIL simulations in Chapter 6 and 7 respectively. Its performance as the network was subjected to pervasive disturbance in the form of Gaussian noise as well as contingencies such as setpoint changes and faults was investigated and evaluated. The results are presented in Chapter 7 in the form of test cases.

8.4   Software Development

A summary of various codes developed and used for the implementation of the IEC 61850 based MRAC algorithm for power system interarea oscillations damping is presented in this section. These software codes are grouped in two categories that corresponds to the two approaches taken in the thesis for the validation of the MRAC algorithm as illustrated in Table 8.1 and 8.2 respectively.

Table 8.1: Summary of the software developed for the digital simulation in the MATLAB® environment

Number	Filename	Application	Appendix
1	fourth_order_rev_final.m	Analyse the dynamics of the 4th order model of the synchronous generator.	Appendix A
2	fourth_order_servo.m	Testing of the closed loop with the 4th order model of the synchronous generator and a servo controller to determine its suitability for the model matching reference model.	Appendix B
3	third_order_servo_rev3	Testing of closed loop system with the 3rd order model of the synchronous generator and a servo controller.	Appendix C
4	refmodel_servo.m	Testing of closed loop system with a 3rd order reference model with a servo controller.	Appendix D
5	fourth_order_sync_gen_mimo_lqr.m	Testing of closed loop system with the 4th order model of the synchronous generator and an LQR and derivation of the model-matching 4th order reference model for the MRAC algorithm.	Appendix E
6	lqr_fourth_order.m	Testing file used as a guide for designing a suitable LQR for a given system.	Appendix F
7	synchronous_generator.slx	Simulink model of the closed loop system with the 3rd order model of the synchronous generator and a servo controller.	Appendix G
8	synchronous_generator_adaptive_control_1.slx	Simulink model of the closed loop system with the 4th order model of the synchronous generator and the MRAC. This model was also configured for C/C++ code generation.	Appendix H
9	synchronous_generator_adaptive_control_plc_coder.slx	Simulink model of the closed loop system with the 4th order model of the	Appendix I

| | | synchronous generator and the MRAC. This slightly modified version of the one in (8) was configured for PLC code generation. | |

Table 8.2: Summary of the software developed for the real-time implementation of the MRAC algorithm with the RTDS, SEL-3355 Rugged Computer, and SEL-3555 RTAC

Number	Filename	Application	Appendix
1	unmatched_uncertainty.m	Simulink function that contains the unmatched uncertainty component of the MRAC.	Appendix J
2	sous_systeme.slx	Simulink subsystem containing the 4th order model of the synchronous generator with a nominal controller which was exported for real-time simulation into the RSCAD C-Builder.	Appendix K
3	ert_main.c	An example of the main program generated by the Simulink Embedded Coder.	Appendix L
4	sous_system.c	This file contains all the major function to implement the Simulink model.	Appendix M
5	sous_system.h	This file contains the interface for all the major functions and data structure declarations.	Appendix N
6	sous_system _data.c sous_system _private.h sous_system _types.h	These files contain data structures specific to the model.	Appendix O
10	MRAC.st	Contains a copy of the implementation of the MRAC.	Appendix P
11	reference_model.st	Contains a copy of the code that ensures that the reference mode remains stable.	Appendix Q
12	interarea_ocsillations_damping.st	Software code that that utilises both the reference_model.st and the MARC.st to compute the controlled signal.	Appendix R

8.5   Applications of the Developed Algorithm

8.5.1   Industrial Applications

The proposed and developed algorithm can be utilized in power system utilities. Additionally, it can also be considered in the development of new control devices and incorporated therein for application in smart grids.

The points below summarize a few of the aforementioned applications:

- Power system planning and operations.
- Power system monitoring, stability analysis and control.
- Research and development at power system utility companies.
- Development of control and automation devices for smart grids by Original Equipment Manufacturers (OEMs).

- The proposed algorithm can be extended and applied in power system networks for real-time small signal rotor angle stability enhancement.

8.5.2 Academic Applications

- The control system theories and concepts presented herein together with their usefulness in the development of the proposed interarea oscillations damping can assist postgraduate students both in the control and power system streams.
- The developed algorithm can be utilised for further studies and extended by researchers.
- This thesis can serve as a use case for those studying the application of the IEC 61860 standard in real-time control system.

8.6 Future Work

Below are highlighted a few of the areas the proposed algorithm can be extended to:

- The control algorithm presented in this thesis can further be refined so that robustness is ensured with a fast adaptation.
- With the application of $\mathcal{L}_1$ Adaptive Control theory, the adaptation can be decoupled from robustness while the transient performance together with the said robustness can be guaranteed in the presence of fast adaptation
- For application in large power system with complex structures, Hierarchical Control Theory can be explored. With this approach, the proposed decentralized architecture can be utilised together with a coordinating controller so that the relationship between subsystems is constantly adjusted to meet the correlation constraints thus achieving the global optimal control.

8.7 Publications Related to this Thesis

8.7.1 Journal Articles

Tswa-wen PP Banga-Banga, Yohan Darcy Mfoumboulou, Carl Kriger (2021). Novel Decentralized Model-Reference Adaptive Control Based Algorithm for Power System Inter-Area Oscillations Damping. *IET Journal of Engineering*, (under review). ID: JOE-2021-08-0160.

Tswa-wen PP Banga-Banga, Yohan Darcy Mfoumboulou, Carl Kriger (2021). Lyapunov-Based Trajectory Tracking Controller and Model-Reference Adaptive Control: Application

for Power System Angle Stability Enhancement. (submitted to IET Control Theory & Applications).

Tswa-wen PP Banga-Banga, Yohan Darcy Mfoumboulou, Carl Kriger (2021). Model-Reference Adaptive Control IEC 61850-Based Algorithm For Power System Small Signal Rotor Angle Stability Enhancement. (submitted to IEEE Transactions on Power Systems).

8.7.2 Conference Publications

Banga-Banga T.P, Adewole, A.C, Tzoneva R. (2017). Oscillation Mode Estimation Using Spectrum Analysis of Synchronized Phasor Measurements. *25th Southern African Universities Power Engineering Conference (SAUPEC 2017)*, pp 818-823.

8.8   Conclusion

The Thesis deliverables are achieved, and applications areas discussed. Additionally, future research directions and publications related to this research are also presented.

BIBLIOGRAPHY

Adewole, A.C, 2016. "Voltage Stability Assessment and Wide Area Protection/Control using Synchrophasor Measurements". Doctor Engineering Thesis, Cape Peninsula University of Technology, CPUT.

Anon. 1997. System Disturbance Stability Studies for Western Systems Coordinating Council (WSCC).

Astrom, K.J. and Wittenmark, Dr.B. 2008. Adaptive Control: Second Edition. Dover. Dover Publications.

Avdakovic, S., Nuhanovic, A., Kusljugic, M. and Becirovic, E. 2014. Applications of wavelets and neural networks for classification of power system dynamics events. Turkish Journal of Electrical Engineering and Computer Sciences, 22(2): 327–340.

Baek, S.-M. 2014. Design of PSS and SCRC Controllers to Damp out Low-Frequency Oscillations in Power System. International Journal of Control and Automation, 7(10): 283–292.

Banga-Banga, T.P., Adewole, A.C. and Tzoneva, R. 2017. Oscillation Mode Estimation Using Spectrum Analysis Of Synchronized Phasor Measurements. *25th Southern African Universities Power Engineering Conference (SAUPEC 2017)*, pp 818-823.

Bellman, R. 1957. *Dynamic programming*. Princeton University Press.

Bhunia, M., Subudhi, B. and Ray, P.K. 2021. Design and Real-Time Implementation of Cascaded Model Reference Adaptive Controllers for a Three-Phase Grid-Connected PV System. *IEEE Journal of Photovoltaics*, 11(5): 1319–1331.

Bosworth, J.T. and Williams-Hayes, P.S. 2007. Flight test results from the NF-15B intelligent flight control system (IFCS) project with adaptation to a simulated stabilator failure. *Collection of Technical Papers - 2007 AIAA InfoTech at Aerospace Conference*, 2: 1080–1097.

Burns, R.S. 2001. Advanced control engineering.

Calise, A.J. and Rysdyk, R.T. 1998. Nonlinear Adaptive Flight Control Using Neural Networks. *IEEE Control Systems*, 18(6): 14–25.

Cutsem, T. and Vournas, C. 2001. *Voltage stability of electric power systems*. 2nd print. Boston ;;London ;;Dordrecht: Kluwer Academic Publishers.

Dobrowolski, J., Korba, P., Rafael, F., Sevilla, S. and Sattinger, W. 2019. *Centralized Wide Area Damping Controller for Power System Oscillation Problems*.

Dreyfus, S. 1962. Variational problems with inequality constraints. *Journal of Mathematical Analysis and Applications*, 4(2): 297–308.

Eremia, M., Bulac, C., Bulac, A.I., Tristiu, I. and Trecat, J. 2000. *Voltage Stability In Power System: Assessment And Control By Artificial Intelligence Techniques*. Available: http://www.cigre.org.

Eremia, M. and Shahidehpour, M. 2013. *Handbook of Electrical Power System Dynamics*.

Eremia, Mircea. and Shahidehpour, M. 2013. Handbook of electrical power system dynamics: modelling, stability, and control.

Fan, R., Wang, S., Huang, R., Lian, J. and Huang, Z. 2020. Wide-area measurement-based modal decoupling for power system oscillation damping. *Electric Power Systems Research*, 178: 106022.

Farahani, M. and Ganjefar, S. 2017. Intelligent power system stabilizer design using adaptive fuzzy sliding mode controller. *Neurocomputing*, 226: 135–144.

Fereidouni, A.R., Vahidi, B., Hoseini Mehr, T. and Tahmasbi, M. 2013. Improvement of low frequency oscillation damping by allocation and design of power system stabilizers in the multi-machine power system. *International Journal of Electrical Power and Energy Systems*, 52(1): 207–220.

Gholinezhad, J., Ebadian, M. and Aghaebrahimi, M.R. 2017. Coordinated design of PSS and SSSC damping controller considering time delays using biogeography-based optimization algorithm. *30th Power System Conference, PSC 2015*: 1–7.

Grayson, L.P. 1965. The status of synthesis using Lyapunov's method. *Automatica (Journal of IFAC)*, 3(2): 91–121.

Hahn, W. 1967. The Direct Method of Lyapunov. *Stability of Motion*: 93–165.

Hashmani, A.A. and Erlich, I. 2012. Mode selective damping of power system electromechanical oscillations for large power systems using supplementary remote signals. *International Journal of Electrical Power and Energy Systems*, 42(1): 605–613.

Hassan, M. and Roy, N.K. 2016. Damping of power system oscillations using STATCOM. *2nd International Conference on Electrical Information and Communication Technologies, EICT 2015*: 545–548.

Hussein, T., Saad, M.S., Elshafei, A.L. and Bahgat, A. 2010. Damping inter-area modes of oscillation using an adaptive fuzzy power system stabilizer. *Electric Power Systems Research*, 80(12): 1428–1436.

IEC 61850-8-1:2012. Communication Networks and Systems for Power Utility Automation-Part8-1. Specific communication service mapping (SCSM) – Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3

IEC 61850-90-5:2012. Communication networks and systems for power utility automation-Part 90-5: Use of IEC 61850 to transmit synchrophasor information according to IEEE C37.118

IEC/IEEE 60255-118-1-2018. Measuring relays and protection equipment – Part 118-1: Synchrophasor for power systems - Measurements

IEC-TR 61850-1-2013. Communication networks and systems for power utility automation – Part 1: Introduction and overview.

IEEE Standard 1344[TM]-1995, IEEE Standard for Synchrophasors for Power Systems

IEEE Standard C37.118[TM]-2005, IEEE Standard for Synchrophasors for Power Systems, 2005

IEEE Standard C37.118.1[TM]-2011, IEEE Standard for Synchrophasors for Power Systems, 2011

IEEE Standard C37.118.2™-2011, IEEE Standard for Synchrophasors for Power Systems, 2011

IEEE Standard for Synchrophasor Measurements for Power Systems – Amendment 1: C37.118.1a-2014 Modification of Selected Performance Requirements.

Ioannou, P.A. and Sun, J. 1996. *ROBUST ADAPTIVE CONTROL*. New Jersey: Prentice-Hall. Inc.

Joseph, T., Tyagi, B. and Kumar, V. 2019. Adaptive optimal wide-area controller for multi-area power system using phasor measurement unit measurements. *IET Generation, Transmission and Distribution*, 13(23): 5284–5293.

Juan, L., Sheng, D. and Xingfu, Z. 2008. A nonlinear control approach to increase power oscillations damping by SSSC. In *Proceedings of the 2008 International Conference on Computer and Electrical Engineering, ICCEE 2008*. 734–738.

Kamwa, I., Trudel, G. and Gerin-Lajoie, L. 1999. Robust design and coordination of multiple damping controllers using nonlinear constrained optimization. *IEEE Power Industry Computer Applications Conference*: 87–94.
Khalil, H.K. 2001. *Nonlinear systems*.

Khosravi-Charmi, M. and Amraee, T. 2018. Wide area damping of electromechanical low frequency oscillations using phasor measurement data. *International Journal of Electrical Power and Energy Systems*, 99: 183–191.

Kim, K.H. 2009. Model reference adaptive control-based adaptive current control scheme of a PM synchronous motor with an improved servo performance. *IET Electric Power Applications*, 3(1): 8–18.

Kundur, P. 1994. Power System Stability and Control - Prabha Kundur - McGraw-Hill Education.

Kundur, P., Klein, M., Rogers, G.J. and Zywno, M.S. 1989. Application of power system stabilizers for enhancement of overall system stability. *IEEE Transactions on Power Systems*, 4(2): 614–626.

Kundur, P., Paserba, J., Ajjarapu, V., Andersson, G., Bose, A., Canizares, C., Hatziargyriou, N., Hill, D., Stankovic, A., Taylor, C., van Cursem, T. and Vittal, V. 2004. Definition and classification of power system stability. *IEEE Transactions on Power Systems*, 19(3): 1387–1401.

Laverde, J.S. and Ríos, M.A. 2018. Damping electromechanical oscillations using supplementary controls at VSC-HVDC stations based on reduced low order models. *Control Engineering Practice*, 79: 195–208.

Lavretsky, E. 2009. Combined/composite model reference adaptive control. *IEEE Transactions on Automatic Control*, 54(11): 2692–2697.

Lei, X., Lerch, E.N. and Povh, D. 2001. Optimization and coordination of damping controls for improving system dynamic performance. *IEEE Transactions on Power Systems*, 16(3): 473–480.

De León-Morales, J., Busawon, K., Acosta-Villarreal, G. and Acha-Daza, S. 2001. Nonlinear control for small synchronous generator. *International Journal of Electrical Power and Energy System*, 23(1): 1–11.

Machowski, Jan., Bialek, J.W. and Bumby, J.R. (James R. 2008. Power system dynamics : stability and control.

Maddela, C.O. and Subudhi, B. 2019. Robust wide-area TCSC controller for damping enhancement of inter-area oscillations in an interconnected power system with actuator saturation. *International Journal of Electrical Power and Energy Systems*, 105: 478–487.

Mahdavian, M., Janghorbani, M., Eshaghpour, I., Ganji, E. and Movahedi, A. 2017. Impacts of the SSSC on damping power system oscillations. *ECTI-CON 2017 - 2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*: 54–57.

Maherani, M. and Erlich, I. 2018. Robust Decentralized Fixed Order Wide Area Damping Controller. *IFAC-PapersOnLine*, 51(28): 438–443.

Maherani, M., Erlich, I. and Krost, G. 2020. Fixed order non-smooth robust H∞ wide area damping controller considering load uncertainties. *International Journal of Electrical Power and Energy Systems*, 115: 105423.

Martin, K.E. 2011. Synchrophasor standards development - IEEE C37.118 and IEC 61850. *Proceedings of the Annual Hawaii International Conference on System Sciences*.

Messina, A.R. 2009. *Inter-area oscillations in power systems: a nonlinear and nonstationary perspective*. 1st ed. A. R. Messina, ed. Springer.

Miotto, E.L. and Covacic, M.R. 2011. Analysis of impacts of PSS controllers and TCSC FACTS devices at dynamic stability of a multimachine system power. *2010 IEEE/PES Transmission and Distribution Conference and Exposition: Latin America, T and D-LA 2010*: 359–365.

Morison, G.K., Gao, B. and Kundur, P. 1993. Voltage stability analysis using static and dynamic approaches. *IEEE Transactions on Power Systems*, 8(3): 1159–1171.

Morison, K., Wang, L. and Kundur, P. 2004. Power system security assessment. *IEEE Power and Energy Magazine*, 2(5): 30–39.

Morsali, J., Kazemzadeh, R., Azizian, M.R. and Parhizkar, A. 2014. Introducing PID-based PSS2B stabilizer in coordination with TCSC damping controller to improve power system dynamic stability. *22nd Iranian Conference on Electrical Engineering, ICEE 2014*: 836–841.

Narendra, K.S. and Annaswamy, A.M. 1987. Persistent excitation in adaptive systems. *Int. J. of Contr.*, 45(1): 127–160.

Naspi. 2012. *NASPI Synchrophasor Technical Report NASPI IEC 61850-90-5 Tutorial October 16, 2012 Summary and Update*.

Nguyen, N.T. 2018. *Advanced Textbooks in Control and Signal Processing Model-Reference Adaptive Control A Primer*. 1st ed. Springer International Publishing.

Pal, B. and Chaudhuri, B. 2005. Robust Control in Power Systems. *Robust Control in Power Systems*.

Patel, A., Ghosh, S. and Folly, K.A. 2018. Inter-area oscillation damping with non-synchronised wide-area power system stabiliser. *IET Generation, Transmission and Distribution*, 12(12): 3070–3078.

Pontryagin, L.S. (Lev S. 1962. *The mathematical theory of optimal processes*. Wiley-Interscience.

Prakash, A., Singh, P., Kumar, K. and Parida, S.K. 2021. Design of TCSC Based Optimal Wide Area Power System Stabilizer for Low-Frequency Oscillation. In *2021 IEEE 4th International Conference on Computing, Power and Communication Technologies (GUCON)*. IEEE: 1–6.

Prakash, T., Singh, V.P. and Mohanty, S.R. 2019. A synchrophasor measurement based wide-area power system stabilizer design for inter-area oscillation damping considering variable time-delays. *International Journal of Electrical Power and Energy Systems*, 105: 131–141.

Rogers, G. 2000. Power System Oscillations. *Power System Oscillations*.

RTDS hardware manual, 2014

RTDS manual, 2014

Rueda, J.L., Juárez, C.A. and Erlich, I. 2011. Wavelet-based analysis of power system low-frequency electromechanical oscillations. *IEEE Transactions on Power Systems*, 26(3): 1733–1743.

Sanchez-Orta, A.E., de León-Morales, J. and Lopez-Toledo, E. 2002. Discrete-time nonlinear control scheme for small synchronous generator. *International Journal of Electrical Power and Energy Systems*, 24(9): 751–764.

Sastry, S. and Bodson, M. 1989. *Adaptive Control- Stability, Convergence, and Robustness*. Thomas Kailath, ed. New Jersey: Prentice Hall.

SEL-3355 Rack-Mounted Computer Instruction Manual, 2021. Schweitzer Engineering Laboratories

SEL-3555 Real-Time Automation Controller (RTAC) Instruction Manual, 2021. Schweitzer Engineering Laboratories

Scott, B. 1979. Power System Dynamics Response Calculations. *Proceeding of the IEEE*, 67(2).

Shi, L., Lee, K.Y. and Wu, F. 2016. Robust ESS-Based Stabilizer Design for Damping Inter-Area Oscillations in Multimachine Power Systems. *IEEE Transactions on Power Systems*, 31(2): 1395–1406.

Slotine, J.-J.E. (Jean-J.E.) and Li, W. 1991. *Applied nonlinear control*. Prentice Hall.

Su, S.J., Zhu, Y.Y., Wang, H.R. and Yun, C. 2019. A method to construct a reference model for model reference adaptive control. *Advances in Mechanical Engineering*, 11(11).

Taylor, C.W., Balu, N.J. and Maratukulam, Dominic. 1994. Power system voltage stability.

Thambirajah, J., Thornhill, N.F. and Pal, B.C. 2011. A multivariate approach towards interarea oscillation damping estimation under ambient conditions via independent component analysis and random decrement. *IEEE Transactions on Power Systems*, 26(1): 315–322.

Tse, C.T. and Tso, S.K. 1993. Refinement of conventional pss design in multimachine system by modal analysis. *IEEE Transactions on Power Systems*, 8(2): 598–605.

Turunen, J. 2011. *A wavelet-based method for estimating damping in power systems*. Doctorate dissertation, Alto University. Available: https://aaltodoc.aalto.fi/handle/123456789/4930.

Turunen, J., Larsson, M., Korba, P., Jyrinsalo, J. and Haarla, L. 2008. Experiences and future plans in monitoring the inter-area power oscillation damping. *IEEE Power and Energy Society 2008 General Meeting: Conversion and Delivery of Electrical Energy in the 21st Century, PES*.

Vanfretti, L., Bengtsson, S., Peric, V.S. and Gjerde, J.O. 2012. Effects of forced oscillations in power system damping estimation. *2012 IEEE International Workshop on Applied Measurements for Power Systems, AMPS 2012 Proceedings*: 59–64.

Vidyasagar, M. (Mathukumalli). 1978. *Nonlinear systems analysis*. Prentice-Hall.

Vittal, V. 2000. Consequence and impact of electric utility industry restructuring on transient stability and small-signal stabilty analysis. *Proceedings of the IEEE*, 88(2): 196–207.

Wang, H.F. 2000. Static synchronous series compensator to damp power system oscillations. *Electric Power Systems Research*, 54(2): 113–119.

Wei, Q., Han, X., Guo, W., Yang, M. and He, N. 2014. The principle of absolute rotor angle control and its effect on suppressing inter-area low frequency oscillations. *International Journal of Electrical Power and Energy Systems*, 63: 1039–1046.

Wei, Q., Guo, W., Han, X. and Guo, M. 2016. Generator rotor angle droop control and its load-following characteristics. *2016 IEEE International Conference on Power System Technology, POWERCON 2016*.

Williams-Hayes, P.S. 2005. *Flight Test Implementation of a Second Generation Intelligent Flight Control System*. Available: http://www.sti.nasa.gov.

Wu, X., Dorfler, F. and Jovanovic, M.R. 2016. Input-Output Analysis and Decentralized Optimal Control of Inter-Area Oscillations in Power Systems. *IEEE Transactions on Power Systems*, 31(3): 2434–2444.

Xianzhang L., Lerch E.N., Povh D. 2001. Optimization and coordination of damping controls for improving system dynamic performance. *IEEE Transactions on Power Systems*. Vol: 16, Issue: 3

Xu, W., Mansour, Y. and Hydro, B.C. 1994. Voltage stability analysis using generic dynamic load models. *IEEE Transactions on Power Systems*, 9(1): 479–493.

Yao, W., Jiang, L., Wen, J.Y., Cheng, S.J. and Wu, Q.H. 2009. An adaptive wide-area damping controller based on generalized predictive control and model identification. *2009 IEEE Power and Energy Society General Meeting, PES '09*.

Yao, W., Jiang, L., Wu, Q.H., Wen, J.Y. and Cheng, S.J. 2010. Design of wide-area damping controllers based on networked predictive control considering communication delays. *IEEE PES General Meeting, PES 2010*.

Yohanandhan, R.V. and Srinivasan, L. 2018. Decentralized Measurement based Adaptive Wide-Area Damping Controller for a Large-scale Power System. *Proceedings of 2018 IEEE International Conference on Power Electronics, Drives and Energy Systems, PEDES 2018*.

Zacharia, L., Asprou, M. and Kyriakides, E. 2020. Wide area control of governors and power system stabilizers with an adaptive tuning of coordination signals. *IEEE Open Access Journal of Power and Energy*, 7(1): 70–81.

Zhu, C., Khammash, M., Vittal, V. and Qiu, W. 2003. Robust power system stabilizer design using H∞ loop shaping approach. *IEEE Transactions on Power Systems*, 18(2): 810–818.

Khairudin, F. 2016. *Synchrophasor Measurement Based Mode Detection And Damping Estimation In Power System Using FFT-Continuous Wavelet Transform Approach*. Doctorate dissertation, Kyushu Institute of Technology.

APPENDICES

APPENDIX A: Fourth-order model of the synchronous generator dynamics

```
MATLAB SCRIPT 1
A = [0          1.0000     0          0;
     -51.969  -1.858     -54.474    -30.576;
      0          0         -6.3750    0;
     -5.0798   -17.828    -16.1779  -22.2500];
B = [0 0;0 0.21;0 0;0.125 0];
C = [1 0 0 0;0.8861 0  1.11 2.641];
D = 0;

sys = ss(A,B,C,D);
sys_order = order(sys);

poles = eig(A);
B_all = isstable(sys)
ctrl = ctrb(A,B);

p1 = -2+2*j;
p2 = -2-2*j;
p3 = -3;
p4 = -2;

is_true = ctrb(A,B);

rank(is_true)

% Controllability - if = 0, the system is controllable
riditelnostSys1 = rank(A) - rank(ctrb(sys))

% Observability, if = 0, the system is fully observable
observabilitySys1 = rank ( A ) - rank ( obsv ( sys ) )

% stability = 0, system not stable
stabilitySys1 = isstable(sys)

p = [-5*2, -8*2, - 2.0 + 3.3119598956507447096246704188697*1i, - 2.0 -
3.3119598956507447096246704188697*1i];

Tc = ctrb(A,B); rank(Tc);
[Abar,Bbar,Cbar,T,k]=ctrbf(A,B,C)

At = Abar(2:4, 2:4)
Bt = Bbar(2:4, :)
Ct = Cbar(:, 2:4)
rank(C)

desired=[-sqrt(3)+1i*sqrt(3) -sqrt(3)-1i*sqrt(3) -10];

temp2 = k * inv(T)

rank(is_true)

pote = place(At,Bt,desired)
zc = zeros (size (pote, 1), 1);
newmatrix = [pote, zc]
km = newmatrix * inv(T)
Br = [0 0 0 -1]';

KI = [192.4000  255.6736;
      -2.6429 -428.7005];
Astar = A-B*km
t = 0:0.02:10;
```

223

```
sys_cl = ss(A-B*km,B,C,D)
Nbar = -inv(C*(inv(A - B*km))*B) %Reduce steady-state error

step(sys_cl*Nbar);

grid on
```

APPENDIX B: Fourth-order model of the synchronous generator with a servo controller

MATLAB Script 2
```
A = [0 1 0 0;0 -0.105 0 0;0 0 -6.375 0; 0 0 0 0.75];
B = [0 0;0 0.21;0 0;0.125 0];
C = [1 0 0 0;0.8861 0  1.11 2.641];
D = [0 0;0 0];

K1 = [16.4901    1.7201    11.2586    59.0811;
      21.0273   13.5553     0.2772     2.8898];

Abar = A - B*K1
K2 = -inv(C*(inv(Abar))*B)

Bbar = B

sys_cl = ss(Abar,Bbar,C,D);
t = 0:0.02:20;

step(sys_cl);
grid on
```

APPENDIX C: Third-order model of the synchronous generator with a servo controller

MATLAB Script 3
```
Am = [0 1 0;-55.1891 -1.8453 -54.4788;-18.46952 0 -30.579];
Bm = [0 0 7.6]';
Cm = [1 0 0];
Dm = 0;
sys = ss(Am,Bm,Cm,Dm);

sys_order = order(sys)
determinant = det(ctrb(Am,Bm))

poles = eig(Am)

[p,z] = pzmap(sys);
grid on

% Closed-loop with compensator
Ahat = [Am zeros(3,1); -Cm 0]
Bhat = [Bm;0]
P = [Am Bm;-Cm 0]
rank(P)
poles_2 = eig(Ahat)

p1 = -1.33+1.49i;
p2 = -1.33-1.49i;
p3 = -13.3;
p4 = -10;
Khat = place(Ahat,Bhat,[p1,p2,p3,p4]);

K = Khat(:,1:3)
KI = Khat(:,4)
AA = [Am-Bm*K -Bm*KI;-Cm 0];
```

```
BB = [zeros(size(Bm)); 1];
CC = [Cm 0];
DD = [0];

% Remove steady-state error
Nbar = inv(CC * inv(AA) * BB)

% ***** Enter step command and plot command *****
t = 0:0.001:10;
y = step(AA,BB,CC,DD,1,t);
plot(t,y)
grid

title('Unit-Step Response')
xlabel('t Sec')
ylabel('Rotor Angle')
```

APPENDIX D: Third-order model of the synchronous generator's reference model with a servo controller

```
MATLAB SCRIPT 4
Am = [0 1 0;0 0 1;-18 -15 -2];
Bm = [0 0 1]';
Cm = [1 0 0];
Dm = 0;
sys = ss(Am,Bm,Cm,Dm);

% Closed-loop with compensator
Ahat = [Am zeros(3,1); -Cm 0]
Bhat = [Bm;0]
P = [Am Bm;-Cm 0];
rank(P)

J = [-sqrt(5)+1i*sqrt(5) -sqrt(5)-1i*sqrt(5) -13.3 -29.5];

[Khat,prec,message] = place(Ahat,Bhat,J)

K = Khat(:,1:3)
KI = Khat(:,4)
AA = [Am-Bm*K -Bm*KI;-Cm 0];
BB = [zeros(size(Bm)); 1];
CC = [Cm 0];
DD = 0;

% Eliminate steady-state error
Nbar = inv(CC * inv(AA) * BB)

S_i = ss(Ahat - Bhat * Khat, [Bm * Nbar; 1], [Cm 0], Dm)

t = 0:0.01:5;
y = step(AA,BB,CC,DD,1,t);

plot(t,y)
grid
title('Unit-Step Response')
xlabel('t Sec')
ylabel('Output y')

grid
```

APPENDIX E: Fourth-order model of the synchronous generator with an LQR controller

MATLAB SCRIPT 5

```
A = [0 1 0 0;0 0 0 0;0 0 -6.375 0; 0 0 0 0.75];
B = [0 0;0 0.21;0 0;0.125 0];
C = [1 0 0 0;2.458 0  -1.326 2.28];
D = [0 0;0 0];

Q=1e7*transpose(C)*C;
test = C'
R = eye(2);
N = 0;
[K1,P,e]=lqr(A,B,Q,R,N)

Am = A - B*K1 % Model matching
K2 = -inv(C*(inv(Am))*B)
Bm = B*K2

sys_cl = ss(Am,Bm,C,D);
t = 0:0.02:3;

subplot(311)
step(sys_cl(1,1), t);

subplot(312)
step(sys_cl(2,2), t);

grid on
```

APPENDIX F: Test file for designing suitable LQRs

MATLAB SCRIPT 6

```
% CREDIT MATHWORKS STACK


clear all;

clc;

display('------------Linear Quadratic Regulator-----------------')

A=input('enter the A matrix = ');

B=input('enter the B matrix = ');

C=input('enter the C matrix = ');

D=input('enter the D matrix = ');


SYS = ss(A,B,C,D);

sys1=tf(SYS);

[MSYS,U] = minreal(SYS)

W=input('if want to enter value of Q manually enter 1 else 2 = ')

if W==1

    Q=input('enter value of q = ')

else

    Q=400*transpose(C)*C

end
```

```
R=input('enter the matrix of R(no. of columns must be equal to B) = ');

Y=input('if want to enter value of N manually enter 1 else 2 = ')

if Y==1

    N=input('enter value of N = ')

else

    N=0

End


[K,S,e]=lqr(A,B,Q,R,N)

n=length(K)

AA=A - B * K

for i=1:n

%     BB(:,i)=B * K(i);

End


CC=C

DD=D


% Remove steady-state error

Nbar = inv(CC * inv(AA) * B);

Nbar2 = -inv(CC*(inv(AA))*B)

BB=B*Nbar2


sys=ss(AA,BB,CC,DD);


step(sys)
```

APPENDIX G: Simulink model of the third-order model of the synchronous generator with a servo controller.
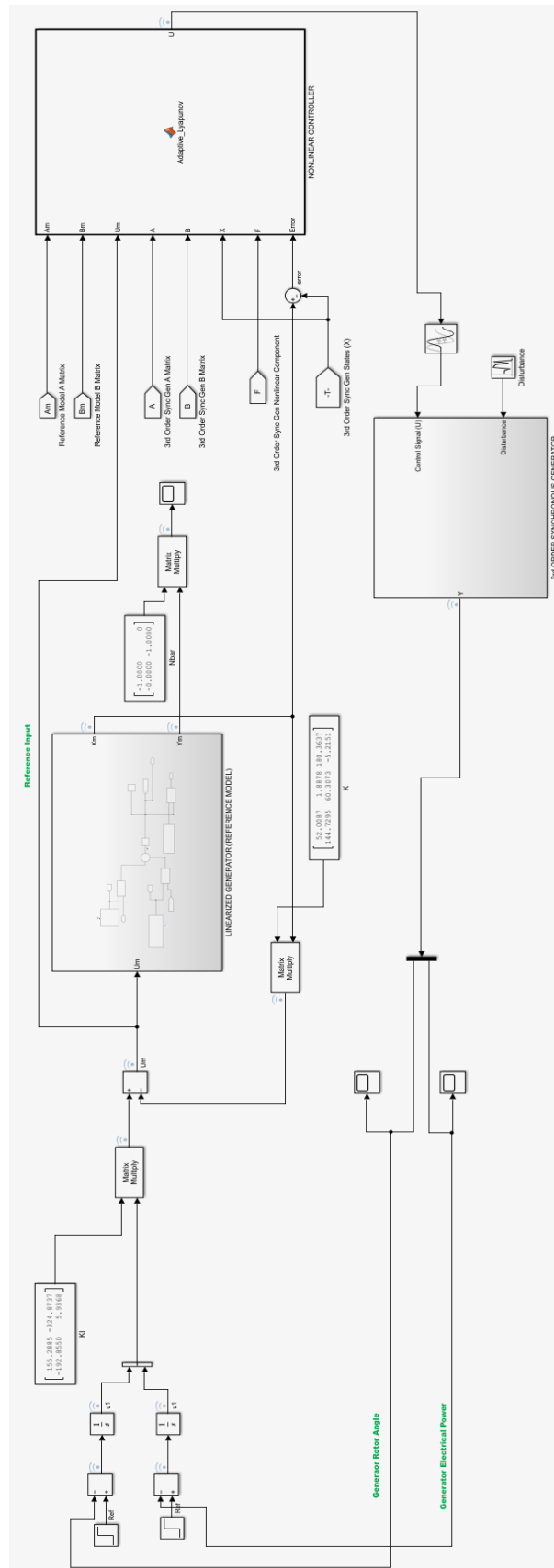
APPENDIX H: Simulink model of the fourth-order model of the synchronous generator with the proposed MRAC - 1
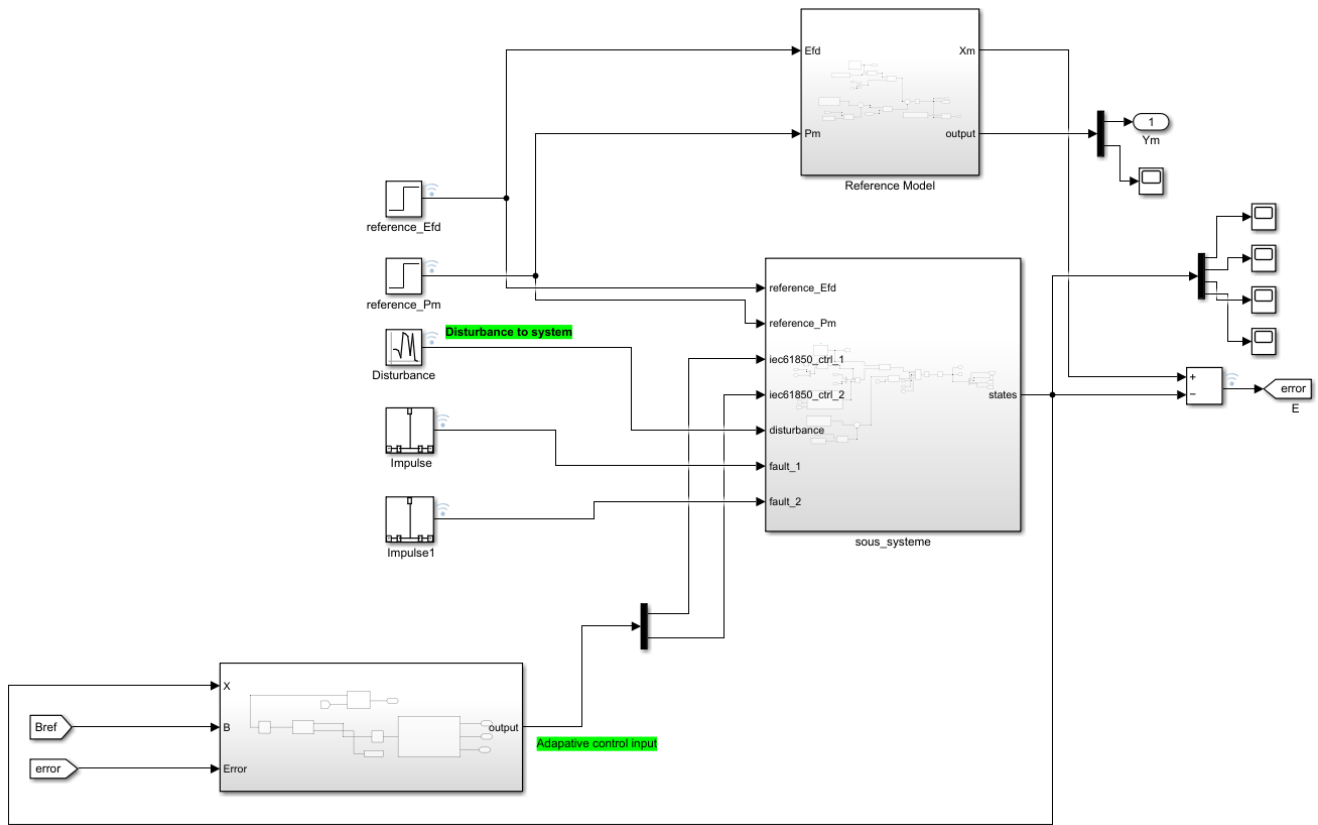
APPENDIX I: Simulink model of the fourth-order model of the synchronous generator with the proposed MRAC – 2
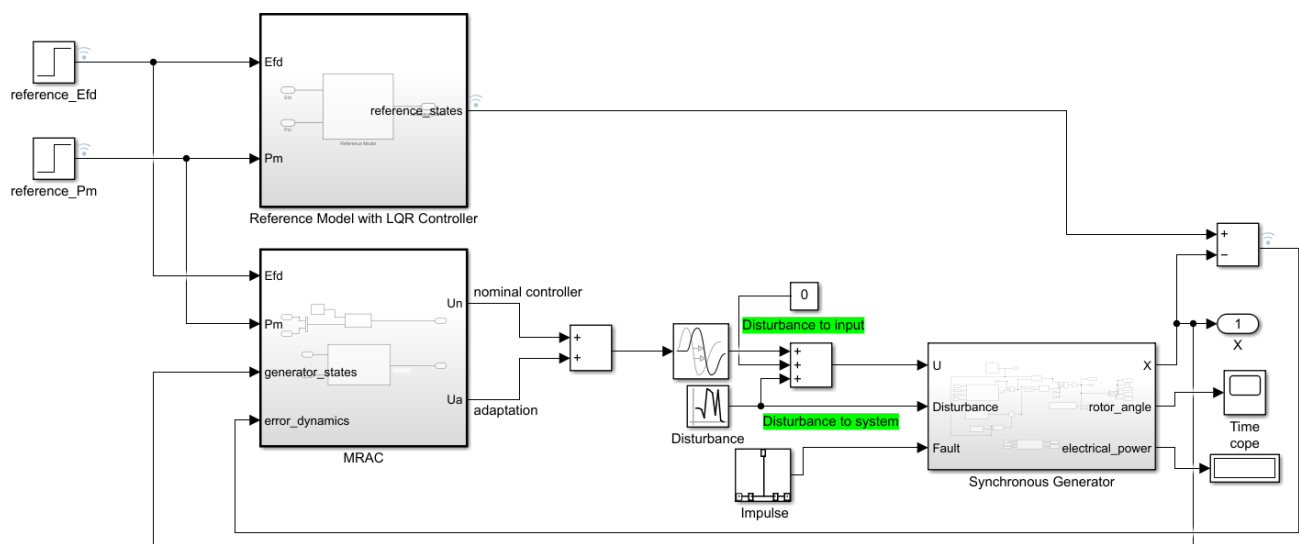
APPENDIX J: Unmatched uncertainty component of the MRAC

```
function unmatched_uncertainty = Fct(gamma,ed_prime,eq_prime,B)

V = 1;
xd = 1.8;
xq = 1.7;
xd_prime = 0.3;
xq_prime = 0.55;
J = 4.774;
Td_prime = 8;
Tq_prime = 0.4;

elt1 = (1/J) * (V*cos(gamma)/xq_prime) * ed_prime;
elt2 = (V/J) * (sin(gamma)/xd_prime) * eq_prime;
% elt3 = ((V^2) / J) * ((1/xq_prime) - (1/xd_prime)) * sin(gamma) * cos(gamma);
elt3 = ((V^2) / J) * ((1/xq) - (1/xd_prime)) * sin(gamma) * cos(gamma);
row2 = elt1 - elt2 - elt3;
row3 = (V/Tq_prime) * ((xq-xd_prime) / xq_prime) * sin(gamma);
row4 = (V/Td_prime) * ((xd/xd_prime) - 1) * cos(gamma);
F = [0;row2;row3;row4];

pseudo_inverse = (transpose(B)*B)\ transpose(B);
if sum(isnan(pseudo_inverse(:)))
    unmatched_uncertainty = [0;0];
else
    unmatched_uncertainty = pseudo_inverse * F; % B' * ((B*B')^-1) * F * X' *
(X*X')^-1
end

end
```

APPENDIX K: Simulink subsystem that contains the 4th order model of the synchronous generator with a nominal controller.

APPENDIX L: C code generated by the Simulink Embedded coder

```
/*
 * File: ert_main.c
 *
 * Code generated for Simulink model 'sous_systeme'.
 *
 * Model version                  : 1.16
 * Simulink Coder version         : 9.5 (R2021a) 14-Nov-2020
 * C/C++ source code generated on : Fri Oct  1 12:20:25 2021
 *
 * Target selection: ert.tlc
 * Embedded hardware selection: Intel->x86-64 (Windows64)
 * Code generation objectives: Unspecified
 * Validation result: Not run
 */

#include <stddef.h>
#include <stdio.h>                   /* This ert_main.c example uses printf/fflush */
#include "sous_systeme.h"              /* Model's header file */
#include "rtwtypes.h"

/*
 * Associating rt_OneStep with a real-time clock or interrupt service routine
 * is what makes the generated code "real-time".  The function rt_OneStep is
 * always associated with the base rate of the model.  Subrates are managed
 * by the base rate from inside the generated code.  Enabling/disabling
 * interrupts and floating point context switches are target specific.  This
 * example code indicates where these should take place relative to executing
 * the generated code step function.  Overrun behavior should be tailored to
 * your application needs.  This example simply sets an error status in the
 * real-time model and returns from rt_OneStep.
 */
void rt_OneStep(void);
```

```
void rt_OneStep(void)
{
  static boolean_T OverrunFlag = false;

  /* Disable interrupts here */

  /* Check for overrun */
  if (OverrunFlag) {
    rtmSetErrorStatus(sous_systeme_M, "Overrun");
    return;
  }

  OverrunFlag = true;

  /* Save FPU context here (if necessary) */
  /* Re-enable timer or interrupt here */
  /* Set model inputs here */

  /* Step the model for base rate */
  sous_systeme_step();

  /* Get model outputs here */

  /* Indicate task complete */
  OverrunFlag = false;

  /* Disable interrupts here */
  /* Restore FPU context here (if necessary) */
  /* Enable interrupts here */
}

/*
 * The example "main" function illustrates what is required by your
 * application code to initialize, execute, and terminate the generated code.
 * Attaching rt_OneStep to a real-time clock is target specific.  This example
 * illustrates how you do this relative to initializing the model.
 */
int_T main(int_T argc, const char *argv[])
{
  /* Unused arguments */
  (void)(argc);
  (void)(argv);

  /* Initialize model */
  sous_systeme_initialize();

  /* Simulating the model step behavior (in non real-time) to
   *  simulate model behavior at stop time.
   */
  while ((rtmGetErrorStatus(sous_systeme_M) == (NULL)) && !rtmGetStopRequested
         (sous_systeme_M)) {
    rt_OneStep();
  }

  /* Disable rt_OneStep() here */

  /* Terminate model */
  sous_systeme_terminate();
  return 0;
}
```

APPENDIX M: Functions to implement in the RSCAD C-Builder

```c
/*
 * File: sous_systeme.c
 *
 * Code generated for Simulink model 'sous_systeme'.
 *
 * Model version                  : 1.16
 * Simulink Coder version         : 9.5 (R2021a) 14-Nov-2020
 * C/C++ source code generated on : Fri Oct  1 12:20:25 2021
 *
 * Target selection: ert.tlc
 * Embedded hardware selection: Intel->x86-64 (Windows64)
 * Code generation objectives: Unspecified
 * Validation result: Not run
 */

#include "sous_systeme.h"
#include "sous_systeme_private.h"

/* Block signals (default storage) */
B_sous_systeme_T sous_systeme_B;

/* Continuous states */
X_sous_systeme_T sous_systeme_X;

/* Block states (default storage) */
DW_sous_systeme_T sous_systeme_DW;

/* External inputs (root inport signals with default storage) */
ExtU_sous_systeme_T sous_systeme_U;

/* External outputs (root outports fed by signals with default storage) */
ExtY_sous_systeme_T sous_systeme_Y;

/* Real-time model */
static RT_MODEL_sous_systeme_T sous_systeme_M_;
RT_MODEL_sous_systeme_T *const sous_systeme_M = &sous_systeme_M_;
static void rate_scheduler(void);

/*
 *    This function updates active task flag for each subrate.
 * The function is called at model base rate, hence the
 * generated code self-manages all its subrates.
 */
static void rate_scheduler(void)
{
  /* Compute which subrates run during the next base time step.  Subrates
   * are an integer multiple of the base rate counter.  Therefore, the subtask
   * counter is reset when it reaches its limit (zero means run).
   */
  (sous_systeme_M->Timing.TaskCounters.TID[2])++;
  if ((sous_systeme_M->Timing.TaskCounters.TID[2]) > 19) {/* Sample time: [0.001s,
0.0s] */
    sous_systeme_M->Timing.TaskCounters.TID[2] = 0;
  }

  (sous_systeme_M->Timing.TaskCounters.TID[3])++;
  if ((sous_systeme_M->Timing.TaskCounters.TID[3]) > 1999) {/* Sample time: [0.1s,
0.0s] */
    sous_systeme_M->Timing.TaskCounters.TID[3] = 0;
  }

  (sous_systeme_M->Timing.TaskCounters.TID[4])++;
  if ((sous_systeme_M->Timing.TaskCounters.TID[4]) > 5999) {/* Sample time: [0.3s,
0.0s] */
```

233

```
      sous_systeme_M->Timing.TaskCounters.TID[4] = 0;
  }
}

/*
 * This function updates continuous states using the ODE3 fixed-step
 * solver algorithm
 */
static void rt_ertODEUpdateContinuousStates(RTWSolverInfo *si )
{
  /* Solver Matrices */
  static const real_T rt_ODE3_A[3] = {
    1.0/2.0, 3.0/4.0, 1.0
  };

  static const real_T rt_ODE3_B[3][3] = {
    { 1.0/2.0, 0.0, 0.0 },

    { 0.0, 3.0/4.0, 0.0 },

    { 2.0/9.0, 1.0/3.0, 4.0/9.0 }
  };

  time_T t = rtsiGetT(si);
  time_T tnew = rtsiGetSolverStopTime(si);
  time_T h = rtsiGetStepSize(si);
  real_T *x = rtsiGetContStates(si);
  ODE3_IntgData *id = (ODE3_IntgData *)rtsiGetSolverData(si);
  real_T *y = id->y;
  real_T *f0 = id->f[0];
  real_T *f1 = id->f[1];
  real_T *f2 = id->f[2];
  real_T hB[3];
  int_T i;
  int_T nXc = 4;
  rtsiSetSimTimeStep(si,MINOR_TIME_STEP);

  /* Save the state values at time t in y, we'll use x as ynew. */
  (void) memcpy(y, x,
                (uint_T)nXc*sizeof(real_T));

  /* Assumes that rtsiSetT and ModelOutputs are up-to-date */
  /* f0 = f(t,y) */
  rtsiSetdX(si, f0);
  sous_systeme_derivatives();

  /* f(:,2) = feval(odefile, t + hA(1), y + f*hB(:,1), args(:)(*)); */
  hB[0] = h * rt_ODE3_B[0][0];
  for (i = 0; i < nXc; i++) {
    x[i] = y[i] + (f0[i]*hB[0]);
  }

  rtsiSetT(si, t + h*rt_ODE3_A[0]);
  rtsiSetdX(si, f1);
  sous_systeme_step();
  sous_systeme_derivatives();

  /* f(:,3) = feval(odefile, t + hA(2), y + f*hB(:,2), args(:)(*)); */
  for (i = 0; i <= 1; i++) {
    hB[i] = h * rt_ODE3_B[1][i];
  }

  for (i = 0; i < nXc; i++) {
    x[i] = y[i] + (f0[i]*hB[0] + f1[i]*hB[1]);
  }
```

```
  rtsiSetT(si, t + h*rt_ODE3_A[1]);
  rtsiSetdX(si, f2);
  sous_systeme_step();
  sous_systeme_derivatives();

  /* tnew = t + hA(3);
     ynew = y + f*hB(:,3); */
  for (i = 0; i <= 2; i++) {
    hB[i] = h * rt_ODE3_B[2][i];
  }

  for (i = 0; i < nXc; i++) {
    x[i] = y[i] + (f0[i]*hB[0] + f1[i]*hB[1] + f2[i]*hB[2]);
  }

  rtsiSetT(si, tnew);
  rtsiSetSimTimeStep(si,MAJOR_TIME_STEP);
}

/* Model step function */
void sous_systeme_step(void)
{
  real_T pseudo_inverse[8];
  real_T y_tmp[8];
  real_T y[4];
  real_T a21;
  real_T a22;
  real_T a22_tmp;
  real_T pseudo_inverse_tmp;
  int32_T r1;
  int32_T r2;
  int32_T y_tmp_0;
  int32_T y_tmp_1;
  if (rtmIsMajorTimeStep(sous_systeme_M)) {
    /* set solver stop time */
    rtsiSetSolverStopTime(&sous_systeme_M->solverInfo,
                          ((sous_systeme_M->Timing.clockTick0+1)*
      sous_systeme_M->Timing.stepSize0));
  }                                    /* end MajorTimeStep */

  /* Update absolute time of base rate at minor time step */
  if (rtmIsMinorTimeStep(sous_systeme_M)) {
    sous_systeme_M->Timing.t[0] = rtsiGetT(&sous_systeme_M->solverInfo);
  }

  if (rtmIsMajorTimeStep(sous_systeme_M) &&
      sous_systeme_M->Timing.TaskCounters.TID[1] == 0) {
    for (r1 = 0; r1 < 4; r1++) {
      /* Product: '<S1>/Product2' incorporates:
       *  Sum: '<S1>/Sum1'
       */
      sous_systeme_B.Product2[r1] = 0.0;
      sous_systeme_B.Product2[r1] += sous_systeme_ConstB.Sum1[r1] *
        sous_systeme_DW.Memory_PreviousInput[0];
      sous_systeme_B.Product2[r1] += sous_systeme_ConstB.Sum1[r1 + 4] *
        sous_systeme_DW.Memory_PreviousInput[1];
      sous_systeme_B.Product2[r1] += sous_systeme_ConstB.Sum1[r1 + 8] *
        sous_systeme_DW.Memory_PreviousInput[2];
      sous_systeme_B.Product2[r1] += sous_systeme_ConstB.Sum1[r1 + 12] *
        sous_systeme_DW.Memory_PreviousInput[3];

      /* MATLAB Function: '<S1>/unmatched_uncertainty' incorporates:
       *  Constant: '<S1>/          '
       */
      r2 = r1 * 2;
      y_tmp[r2] = sous_systeme_ConstP._Value[r1];
```

```
      y_tmp[r2 + 1] = sous_systeme_ConstP._Value[r1 + 4];
    }

    /* MATLAB Function: '<S1>/unmatched_uncertainty' incorporates:
     *  Constant: '<S1>/            '
     */
    for (r1 = 0; r1 < 2; r1++) {
      for (r2 = 0; r2 < 2; r2++) {
        y_tmp_0 = r2 + r1 * 2;
        y[y_tmp_0] = 0.0;
        y_tmp_1 = r1 * 4;
        y[y_tmp_0] += sous_systeme_ConstP._Value[y_tmp_1] * y_tmp[r2];
        y[y_tmp_0] += sous_systeme_ConstP._Value[y_tmp_1 + 1] * y_tmp[r2 + 2];
        y[y_tmp_0] += sous_systeme_ConstP._Value[y_tmp_1 + 2] * y_tmp[r2 + 4];
        y[y_tmp_0] += sous_systeme_ConstP._Value[y_tmp_1 + 3] * y_tmp[r2 + 6];
      }
    }

    if (fabs(y[1]) > fabs(y[0])) {
      r1 = 1;
      r2 = 0;
    } else {
      r1 = 0;
      r2 = 1;
    }

    a21 = y[r2] / y[r1];
    a22_tmp = y[r1 + 2];
    a22 = y[r2 + 2] - a22_tmp * a21;
    pseudo_inverse[1] = (y_tmp[r2] - y_tmp[r1] * a21) / a22;
    pseudo_inverse[0] = (y_tmp[r1] - a22_tmp * pseudo_inverse[1]) / y[r1];
    pseudo_inverse_tmp = y_tmp[r1 + 2];
    pseudo_inverse[3] = (y_tmp[r2 + 2] - pseudo_inverse_tmp * a21) / a22;
    pseudo_inverse[2] = (pseudo_inverse_tmp - a22_tmp * pseudo_inverse[3]) /
      y[r1];
    pseudo_inverse_tmp = y_tmp[r1 + 4];
    pseudo_inverse[5] = (y_tmp[r2 + 4] - pseudo_inverse_tmp * a21) / a22;
    pseudo_inverse[4] = (pseudo_inverse_tmp - a22_tmp * pseudo_inverse[5]) /
      y[r1];
    pseudo_inverse_tmp = y_tmp[r1 + 6];
    pseudo_inverse[7] = (y_tmp[r2 + 6] - pseudo_inverse_tmp * a21) / a22;
    pseudo_inverse[6] = (pseudo_inverse_tmp - a22_tmp * pseudo_inverse[7]) /
      y[r1];
    a22 = sin(sous_systeme_DW.Memory_PreviousInput[0]);
    a22_tmp = cos(sous_systeme_DW.Memory_PreviousInput[0]);
    a21 = (a22_tmp / 0.55 * 0.20946795140343527 *
           sous_systeme_DW.Memory_PreviousInput[2] - a22 / 0.3 *
           0.20946795140343527 * sous_systeme_DW.Memory_PreviousInput[3]) -
      -0.57501006267609678 * a22 * a22_tmp;
    a22 *= 6.3636363636363624;
    a22_tmp *= 0.625;
    for (r1 = 0; r1 < 2; r1++) {
      sous_systeme_B.unmatched_uncertainty[r1] = 0.0;
      sous_systeme_B.unmatched_uncertainty[r1] += pseudo_inverse[r1 + 2] * a21;
      sous_systeme_B.unmatched_uncertainty[r1] += pseudo_inverse[r1 + 4] * a22;
      sous_systeme_B.unmatched_uncertainty[r1] += pseudo_inverse[r1 + 6] *
        a22_tmp;
    }
  }

  /* Sum: '<S1>/Sum4' incorporates:
   *  Inport: '<Root>/iec61850_ctrl_1'
   *  Inport: '<Root>/iec61850_ctrl_2'
   *  Inport: '<Root>/reference_Efd'
   *  Inport: '<Root>/reference_Pm'
   *  Product: '<S1>/Product4'
```

```
 */
a21 = ((-6.1000000000000005 * sous_systeme_U.reference_Efd + 3162.3 *
        sous_systeme_U.reference_Pm) + sous_systeme_U.iec61850_ctrl_1) +
  sous_systeme_B.unmatched_uncertainty[0];
a22_tmp = ((3162.3 * sous_systeme_U.reference_Efd + 6.1000000000000005 *
            sous_systeme_U.reference_Pm) + sous_systeme_U.iec61850_ctrl_2) +
  sous_systeme_B.unmatched_uncertainty[1];
for (r1 = 0; r1 < 4; r1++) {
  /* Sum: '<S1>/Sum2' incorporates:
   *  Constant: '<S1>/             '
   *  Inport: '<Root>/disturbance'
   *  Inport: '<Root>/fault_1'
   *  Inport: '<Root>/fault_2'
   *  Product: '<S1>/Product3'
   */
  sous_systeme_B.Sum2[r1] = ((((sous_systeme_ConstP._Value[r1 + 4] * a22_tmp +
    sous_systeme_ConstP._Value[r1] * a21) + sous_systeme_B.Product2[r1]) +
    sous_systeme_U.disturbance) + sous_systeme_U.fault_1) +
    sous_systeme_U.fault_2;
}

if (rtmIsMajorTimeStep(sous_systeme_M) &&
    sous_systeme_M->Timing.TaskCounters.TID[1] == 0) {
  /* Outport: '<Root>/states' */
  sous_systeme_Y.states[0] = sous_systeme_DW.Memory_PreviousInput[0];

  /* Outport: '<Root>/Outport2' */
  sous_systeme_Y.Outport2[0] = sous_systeme_DW.Memory_PreviousInput[0];

  /* Outport: '<Root>/states' */
  sous_systeme_Y.states[1] = sous_systeme_DW.Memory_PreviousInput[1];

  /* Outport: '<Root>/Outport2' */
  sous_systeme_Y.Outport2[1] = sous_systeme_DW.Memory_PreviousInput[1];

  /* Outport: '<Root>/states' */
  sous_systeme_Y.states[2] = sous_systeme_DW.Memory_PreviousInput[2];

  /* Outport: '<Root>/Outport2' */
  sous_systeme_Y.Outport2[2] = sous_systeme_DW.Memory_PreviousInput[2];

  /* Outport: '<Root>/states' */
  sous_systeme_Y.states[3] = sous_systeme_DW.Memory_PreviousInput[3];

  /* Outport: '<Root>/Outport2' */
  sous_systeme_Y.Outport2[3] = sous_systeme_DW.Memory_PreviousInput[3];
}

/* Integrator: '<S1>/Integrator5' */
sous_systeme_B.Integrator5[0] = sous_systeme_X.Integrator5_CSTATE[0];
sous_systeme_B.Integrator5[1] = sous_systeme_X.Integrator5_CSTATE[1];
sous_systeme_B.Integrator5[2] = sous_systeme_X.Integrator5_CSTATE[2];
sous_systeme_B.Integrator5[3] = sous_systeme_X.Integrator5_CSTATE[3];
if (rtmIsMajorTimeStep(sous_systeme_M)) {
  if (rtmIsMajorTimeStep(sous_systeme_M) &&
      sous_systeme_M->Timing.TaskCounters.TID[1] == 0) {
    /* Update for Memory: '<S1>/Memory' */
    sous_systeme_DW.Memory_PreviousInput[0] = sous_systeme_B.Integrator5[0];
    sous_systeme_DW.Memory_PreviousInput[1] = sous_systeme_B.Integrator5[1];
    sous_systeme_DW.Memory_PreviousInput[2] = sous_systeme_B.Integrator5[2];
    sous_systeme_DW.Memory_PreviousInput[3] = sous_systeme_B.Integrator5[3];
  }
}                                    /* end MajorTimeStep */

if (rtmIsMajorTimeStep(sous_systeme_M)) {
  rt_ertODEUpdateContinuousStates(&sous_systeme_M->solverInfo);
```

```c
    /* Update absolute time for base rate */
    /* The "clockTick0" counts the number of times the code of this task has
     * been executed. The absolute time is the multiplication of "clockTick0"
     * and "Timing.stepSize0". Size of "clockTick0" ensures timer will not
     * overflow during the application lifespan selected.
     */
    ++sous_systeme_M->Timing.clockTick0;
    sous_systeme_M->Timing.t[0] = rtsiGetSolverStopTime
      (&sous_systeme_M->solverInfo);

    {
      /* Update absolute timer for sample time: [5.0E-5s, 0.0s] */
      /* The "clockTick1" counts the number of times the code of this task has
       * been executed. The resolution of this integer timer is 5.0E-5, which is the
step size
       * of the task. Size of "clockTick1" ensures timer will not overflow during
the
       * application lifespan selected.
       */
      sous_systeme_M->Timing.clockTick1++;
    }

    rate_scheduler();
  }                                    /* end MajorTimeStep */
}

/* Derivatives for root system: '<Root>' */
void sous_systeme_derivatives(void)
{
  XDot_sous_systeme_T *_rtXdot;
  _rtXdot = ((XDot_sous_systeme_T *) sous_systeme_M->derivs);

  /* Derivatives for Integrator: '<S1>/Integrator5' */
  _rtXdot->Integrator5_CSTATE[0] = sous_systeme_B.Sum2[0];
  _rtXdot->Integrator5_CSTATE[1] = sous_systeme_B.Sum2[1];
  _rtXdot->Integrator5_CSTATE[2] = sous_systeme_B.Sum2[2];
  _rtXdot->Integrator5_CSTATE[3] = sous_systeme_B.Sum2[3];
}

/* Model initialize function */
void sous_systeme_initialize(void)
{
  /* Registration code */
  {
    /* Setup solver object */
    rtsiSetSimTimeStepPtr(&sous_systeme_M->solverInfo,
                          &sous_systeme_M->Timing.simTimeStep);
    rtsiSetTPtr(&sous_systeme_M->solverInfo, &rtmGetTPtr(sous_systeme_M));
    rtsiSetStepSizePtr(&sous_systeme_M->solverInfo,
                       &sous_systeme_M->Timing.stepSize0);
    rtsiSetdXPtr(&sous_systeme_M->solverInfo, &sous_systeme_M->derivs);
    rtsiSetContStatesPtr(&sous_systeme_M->solverInfo, (real_T **)
                         &sous_systeme_M->contStates);
    rtsiSetNumContStatesPtr(&sous_systeme_M->solverInfo,
      &sous_systeme_M->Sizes.numContStates);
    rtsiSetNumPeriodicContStatesPtr(&sous_systeme_M->solverInfo,
      &sous_systeme_M->Sizes.numPeriodicContStates);
    rtsiSetPeriodicContStateIndicesPtr(&sous_systeme_M->solverInfo,
      &sous_systeme_M->periodicContStateIndices);
    rtsiSetPeriodicContStateRangesPtr(&sous_systeme_M->solverInfo,
      &sous_systeme_M->periodicContStateRanges);
    rtsiSetErrorStatusPtr(&sous_systeme_M->solverInfo, (&rtmGetErrorStatus
      (sous_systeme_M)));
    rtsiSetRTModelPtr(&sous_systeme_M->solverInfo, sous_systeme_M);
  }
```

```
   rtsiSetSimTimeStep(&sous_systeme_M->solverInfo, MAJOR_TIME_STEP);
   sous_systeme_M->intgData.y = sous_systeme_M->odeY;
   sous_systeme_M->intgData.f[0] = sous_systeme_M->odeF[0];
   sous_systeme_M->intgData.f[1] = sous_systeme_M->odeF[1];
   sous_systeme_M->intgData.f[2] = sous_systeme_M->odeF[2];
   sous_systeme_M->contStates = ((X_sous_systeme_T *) &sous_systeme_X);
   rtsiSetSolverData(&sous_systeme_M->solverInfo, (void *)
                     &sous_systeme_M->intgData);
   rtsiSetSolverName(&sous_systeme_M->solverInfo,"ode3");
   rtmSetTPtr(sous_systeme_M, &sous_systeme_M->Timing.tArray[0]);
   sous_systeme_M->Timing.stepSize0 = 5.0E-5;

   /* InitializeConditions for Integrator: '<S1>/Integrator5' */
   sous_systeme_X.Integrator5_CSTATE[0] = 0.001;
   sous_systeme_X.Integrator5_CSTATE[1] = 0.001;
   sous_systeme_X.Integrator5_CSTATE[2] = 0.001;
   sous_systeme_X.Integrator5_CSTATE[3] = 0.001;

   /* ConstCode for Outport: '<Root>/Outport1' incorporates:
    *  Constant: '<S1>/            '
    */
   memcpy(&sous_systeme_Y.Outport1[0], &sous_systeme_ConstP._Value[0], sizeof
          (real_T) << 3U);
}
```

APPENDIX N: Interface for the functions in APPENDIX L

```
/*
 * File: sous_systeme.h
 *
 * Code generated for Simulink model 'sous_systeme'.
 *
 * Model version                  : 1.16
 * Simulink Coder version         : 9.5 (R2021a) 14-Nov-2020
 * C/C++ source code generated on : Fri Oct  1 12:20:25 2021
 *
 * Target selection: ert.tlc
 * Embedded hardware selection: Intel->x86-64 (Windows64)
 * Code generation objectives: Unspecified
 * Validation result: Not run
 */

#ifndef RTW_HEADER_sous_systeme_h_
#define RTW_HEADER_sous_systeme_h_
#include <math.h>
#include <string.h>
#ifndef sous_systeme_COMMON_INCLUDES_
#define sous_systeme_COMMON_INCLUDES_
#include "rtwtypes.h"
#include "rtw_continuous.h"
#include "rtw_solver.h"
#endif                                  /* sous_systeme_COMMON_INCLUDES_ */

#include "sous_systeme_types.h"

/* Macros for accessing real-time model data structure */
#ifndef rtmGetErrorStatus
#define rtmGetErrorStatus(rtm)          ((rtm)->errorStatus)
#endif

#ifndef rtmSetErrorStatus
```

```c
#define rtmSetErrorStatus(rtm, val)      ((rtm)->errorStatus = (val))
#endif

#ifndef rtmGetStopRequested
#define rtmGetStopRequested(rtm)         ((rtm)->Timing.stopRequestedFlag)
#endif

#ifndef rtmSetStopRequested
#define rtmSetStopRequested(rtm, val)  ((rtm)->Timing.stopRequestedFlag = (val))
#endif

#ifndef rtmGetStopRequestedPtr
#define rtmGetStopRequestedPtr(rtm)     (&((rtm)->Timing.stopRequestedFlag))
#endif

#ifndef rtmGetT
#define rtmGetT(rtm)                     (rtmGetTPtr((rtm))[0])
#endif

#ifndef rtmGetTPtr
#define rtmGetTPtr(rtm)                  ((rtm)->Timing.t)
#endif

/* Block signals (default storage) */
typedef struct {
  real_T Product2[4];                    /* '<S1>/Product2' */
  real_T Sum2[4];                        /* '<S1>/Sum2' */
  real_T Integrator5[4];                 /* '<S1>/Integrator5' */
  real_T unmatched_uncertainty[2];       /* '<S1>/unmatched_uncertainty' */
} B_sous_systeme_T;

/* Block states (default storage) for system '<Root>' */
typedef struct {
  real_T Memory_PreviousInput[4];        /* '<S1>/Memory' */
} DW_sous_systeme_T;

/* Continuous states (default storage) */
typedef struct {
  real_T Integrator5_CSTATE[4];          /* '<S1>/Integrator5' */
} X_sous_systeme_T;

/* State derivatives (default storage) */
typedef struct {
  real_T Integrator5_CSTATE[4];          /* '<S1>/Integrator5' */
} XDot_sous_systeme_T;

/* State disabled  */
typedef struct {
  boolean_T Integrator5_CSTATE[4];       /* '<S1>/Integrator5' */
} XDis_sous_systeme_T;

/* Invariant block signals (default storage) */
typedef struct {
  const real_T Product1[16];             /* '<S1>/Product1' */
  const real_T Sum1[16];                 /* '<S1>/Sum1' */
} ConstB_sous_systeme_T;

#ifndef ODE3_INTG
#define ODE3_INTG

/* ODE3 Integration Data */
typedef struct {
  real_T *y;                             /* output */
  real_T *f[3];                          /* derivatives */
} ODE3_IntgData;
```

```
#endif

/* Constant parameters (default storage) */
typedef struct {
  /* Expression: [0 0;0 0.21;0 0;0.125 0]
   * Referenced by: '<S1>/           '
   */
  real_T _Value[8];
} ConstP_sous_systeme_T;

/* External inputs (root inport signals with default storage) */
typedef struct {
  real_T reference_Efd;                 /* '<Root>/reference_Efd' */
  real_T reference_Pm;                  /* '<Root>/reference_Pm' */
  real_T iec61850_ctrl_1;               /* '<Root>/iec61850_ctrl_1' */
  real_T iec61850_ctrl_2;               /* '<Root>/iec61850_ctrl_2' */
  real_T disturbance;                   /* '<Root>/disturbance' */
  real_T fault_1;                       /* '<Root>/fault_1' */
  real_T fault_2;                       /* '<Root>/fault_2' */
} ExtU_sous_systeme_T;

/* External outputs (root outports fed by signals with default storage) */
typedef struct {
  real_T states[4];                     /* '<Root>/states' */
  real_T Outport1[8];                   /* '<Root>/Outport1' */
  real_T Outport2[4];                   /* '<Root>/Outport2' */
} ExtY_sous_systeme_T;

/* Real-time Model Data Structure */
struct tag_RTM_sous_systeme_T {
  const char_T *errorStatus;
  RTWSolverInfo solverInfo;
  X_sous_systeme_T *contStates;
  int_T *periodicContStateIndices;
  real_T *periodicContStateRanges;
  real_T *derivs;
  boolean_T *contStateDisabled;
  boolean_T zCCacheNeedsReset;
  boolean_T derivCacheNeedsReset;
  boolean_T CTOutputIncnstWithState;
  real_T odeY[4];
  real_T odeF[3][4];
  ODE3_IntgData intgData;

  /*
   * Sizes:
   * The following substructure contains sizes information
   * for many of the model attributes such as inputs, outputs,
   * dwork, sample times, etc.
   */
  struct {
    int_T numContStates;
    int_T numPeriodicContStates;
    int_T numSampTimes;
  } Sizes;

  /*
   * Timing:
   * The following substructure contains information regarding
   * the timing information for the model.
   */
  struct {
    uint32_T clockTick0;
    time_T stepSize0;
    uint32_T clockTick1;
    struct {
```

```
      uint16_T TID[5];
    } TaskCounters;

    SimTimeStep simTimeStep;
    boolean_T stopRequestedFlag;
    time_T *t;
    time_T tArray[5];
  } Timing;
};

/* Block signals (default storage) */
extern B_sous_systeme_T sous_systeme_B;

/* Continuous states (default storage) */
extern X_sous_systeme_T sous_systeme_X;

/* Block states (default storage) */
extern DW_sous_systeme_T sous_systeme_DW;

/* External inputs (root inport signals with default storage) */
extern ExtU_sous_systeme_T sous_systeme_U;

/* External outputs (root outports fed by signals with default storage) */
extern ExtY_sous_systeme_T sous_systeme_Y;
extern const ConstB_sous_systeme_T sous_systeme_ConstB;/* constant block i/o */

/* Constant parameters (default storage) */
extern const ConstP_sous_systeme_T sous_systeme_ConstP;

/* Model entry point functions */
extern void sous_systeme_initialize(void);
extern void sous_systeme_step(void);
extern void sous_systeme_terminate(void);

/* Real-time Model object */
extern RT_MODEL_sous_systeme_T *const sous_systeme_M;

/*-
 * These blocks were eliminated from the model due to optimizations:
 *
 * Block '<S1>/Scope1' : Unused code path elimination
 */

/*-
 * The generated code includes comments that allow you to trace directly
 * back to the appropriate location in the model.  The basic format
 * is <system>/block_name, where system is the system number (uniquely
 * assigned by Simulink) and block_name is the name of the block.
 *
 * Note that this particular code originates from a subsystem build,
 * and has its own system numbers different from the parent model.
 * Refer to the system hierarchy for this subsystem below, and use the
 * MATLAB hilite_system command to trace the generated code back
 * to the parent model.  For example,
 *
 * hilite_system('synch_generator_adaptive_control_rev9/sous_systeme')    - opens
subsystem synch_generator_adaptive_control_rev9/sous_systeme
 * hilite_system('synch_generator_adaptive_control_rev9/sous_systeme/Kp') - opens
and selects block Kp
 *
 * Here is the system hierarchy for this model
 *
 * '<Root>' : 'synch_generator_adaptive_control_rev9'
 * '<S1>'   : 'synch_generator_adaptive_control_rev9/sous_systeme'
 * '<S2>'   :
'synch_generator_adaptive_control_rev9/sous_systeme/unmatched_uncertainty'
```

```
 */
#endif                                      /* RTW_HEADER_sous_systeme_h_ */
```

APPENDIX O: Data structures specific to the SIMULINK subsystem containing the synchronous
generator with a nominal controller

```
/*
 * File: sous_systeme_data.c
 *
 * Code generated for Simulink model 'sous_systeme'.
 *
 * Model version                  : 1.16
 * Simulink Coder version         : 9.5 (R2021a) 14-Nov-2020
 * C/C++ source code generated on : Fri Oct  1 12:20:25 2021
 *
 * Target selection: ert.tlc
 * Embedded hardware selection: Intel->x86-64 (Windows64)
 */

#include "sous_systeme.h"
#include "sous_systeme_private.h"

/* Invariant block signals (default storage) */
const ConstB_sous_systeme_T sous_systeme_ConstB = {
  { 0.0, 667.233, 0.0, 970.85, 0.0, 36.393, 0.0, 1.0375, 0.0, -1.617, 0.0,
    -520.46250000000009, 0.0, 2.9189999999999996, 0.0, 902.0 },/*
'<S1>/Product1' */

  { 0.0, -667.233, 0.0, -970.85, 1.0, -36.498, 0.0, -1.0375, 0.0, 1.617, -
6.375,
    520.46250000000009, 0.0, -2.9189999999999996, 0.0, -901.25 }/*
'<S1>/Sum1' */
};

/* Constant parameters (default storage) */
const ConstP_sous_systeme_T sous_systeme_ConstP = {
  /* Expression: [0 0;0 0.21;0 0;0.125 0]
   * Referenced by: '<S1>/            '
   */
  { 0.0, 0.0, 0.0, 0.125, 0.0, 0.21, 0.0, 0.0 }
};
```

```
/*
 * File: sous_systeme_private.h
 *
 * Code generated for Simulink model 'sous_systeme'.
 *
 * Model version                  : 1.16
 * Simulink Coder version         : 9.5 (R2021a) 14-Nov-2020
 * C/C++ source code generated on : Fri Oct  1 12:20:25 2021
 *
 * Target selection: ert.tlc
 * Embedded hardware selection: Intel->x86-64 (Windows64)
 */

#ifndef RTW_HEADER_sous_systeme_private_h_
```

```
#define RTW_HEADER_sous_systeme_private_h_
#include "rtwtypes.h"

/* Private macros used by the generated code to access rtModel */
#ifndef rtmIsMajorTimeStep
#define rtmIsMajorTimeStep(rtm)          (((rtm)->Timing.simTimeStep) ==
MAJOR_TIME_STEP)
#endif

#ifndef rtmIsMinorTimeStep
#define rtmIsMinorTimeStep(rtm)          (((rtm)->Timing.simTimeStep) ==
MINOR_TIME_STEP)
#endif

#ifndef rtmSetTPtr
#define rtmSetTPtr(rtm, val)             ((rtm)->Timing.t = (val))
#endif

/* private model entry point functions */
extern void sous_systeme_derivatives(void);

#endif                                   /* RTW_HEADER_sous_systeme_private_h_
*/
```

```
/*
 * File: sous_systeme_types.h
 *
 * Code generated for Simulink model 'sous_systeme'.
 *
 * Model version                  : 1.16
 * Simulink Coder version         : 9.5 (R2021a) 14-Nov-2020
 * C/C++ source code generated on : Fri Oct  1 12:20:25 2021
 *
 * Target selection: ert.tlc
 * Embedded hardware selection: Intel->x86-64 (Windows64)
 */

#ifndef RTW_HEADER_sous_systeme_types_h_
#define RTW_HEADER_sous_systeme_types_h_

/* Model Code Variants */

/* Forward declaration for rtModel */
typedef struct tag_RTM_sous_systeme_T RT_MODEL_sous_systeme_T;

#endif                                   /* RTW_HEADER_sous_systeme_types_h_
*/
```

```
(*
 *
 * File: synch_generator_adaptive_control_plc_coder.st
 *
 * IEC 61131-3 Structured Text (ST) code generated for subsystem
"synch_generator_adaptive_control_plc_coder/Reference Model with LQR Controller"
 *
 * Model name                        : Reference Model with LQR Controller
 * Model version                     : 1.12
 * Model creator                     : pierr
 * Model last modified by            : Administrateur SEL
 * Model last modified on            : Sat Jun 26 13:51:50 2021
 * Model sample time                 : 5e-05s
 * Subsystem name                    :
synch_generator_adaptive_control_plc_coder/Reference Model with LQR Controller
 * Subsystem sample time             : 5e-05s
 * Simulink PLC Coder version        : 2.6 (R2018b) 24-May-2018
 * ST code generated on              : Sat Jun 26 13:53:06 2021
 *
 * Target IDE selection              : Generic
 * Test Bench included               : No
 *
 *)
FUNCTION_BLOCK reference_model
VAR
    Product3: ARRAY [0..15] OF LREAL;
    c_DiscreteTimeIntegrator_DS: ARRAY [0..3] OF LREAL;
    Sum2: ARRAY [0..15] OF LREAL;
    UnitDelay_DSTATE: ARRAY [0..3] OF LREAL;
    Product1: ARRAY [0..7] OF LREAL;
    A_Value: ARRAY [0..15] OF LREAL := [0.0,0.0,0.0,0.0,1.0,-0.105,0.0,0.0,0.0,0.0,-
6.375,0.0,0.0,0.0,0.0,0.75];
    B_Value: ARRAY [0..7] OF LREAL := [0.0,0.0,0.0,0.125,0.0,0.21,0.0,0.0];
    K_Value: ARRAY [0..7] OF LREAL := [7766.8,3177.2999999999997,8.3,173.3,-
4163.7000000000007,-7.7,7216.0,
        13.899999999999999];
END_VAR
VAR_TEMP
    rtb_DiscreteTimeIntegrator: ARRAY [0..3] OF LREAL;
    i: DINT;
    i_0: DINT;
    tmp: LREAL;
    Product3_tmp: DINT;
    Product3_tmp_0: DINT;
    Product3_tmp_1: DINT;
    Product1_tmp: LREAL;
END_VAR

VAR_INPUT
    ssMethodType: DINT;
    Efd: LREAL;
    Pm: LREAL;
END_VAR

VAR_OUTPUT
```

245

```
     reference_states: ARRAY [0..3] OF LREAL;
END_VAR
CASE ssMethodType OF
    0:
         (* InitializeConditions for DiscreteIntegrator: '<S2>/Discrete-Time Integrator'
*)
         c_DiscreteTimeIntegrator_DS[0] := 0.0;
         (* InitializeConditions for UnitDelay: '<S2>/Unit Delay' *)
         UnitDelay_DSTATE[0] := 0.0;
         (* InitializeConditions for DiscreteIntegrator: '<S2>/Discrete-Time Integrator'
*)
         c_DiscreteTimeIntegrator_DS[1] := 0.0;
         (* InitializeConditions for UnitDelay: '<S2>/Unit Delay' *)
         UnitDelay_DSTATE[1] := 0.0;
         (* InitializeConditions for DiscreteIntegrator: '<S2>/Discrete-Time Integrator'
*)
         c_DiscreteTimeIntegrator_DS[2] := 0.0;
         (* InitializeConditions for UnitDelay: '<S2>/Unit Delay' *)
         UnitDelay_DSTATE[2] := 0.0;
         (* InitializeConditions for DiscreteIntegrator: '<S2>/Discrete-Time Integrator'
*)
         c_DiscreteTimeIntegrator_DS[3] := 0.0;
         (* InitializeConditions for UnitDelay: '<S2>/Unit Delay' *)
         UnitDelay_DSTATE[3] := 0.0;
    1:
             // Get the values of the field voltage and mechanical power (GOOSE)
             Efd:= RTDS_CTRL_GENERATOR_DATA.MV005.instMag;
             Pm := RTDS_CTRL_GENERATOR_DATA.MV006.instMag;

         FOR i := 0 TO 3 DO
             (* DiscreteIntegrator: '<S2>/Discrete-Time Integrator' *)
             rtb_DiscreteTimeIntegrator[i] := c_DiscreteTimeIntegrator_DS[i];
             (* Product: '<S2>/Product3' incorporates:
              *  Constant: '<S2>/        B   '
              *  Constant: '<S2>/K' *)
             FOR i_0 := 0 TO 3 DO
                 Product3_tmp := i_0 * 4;
                 Product3_tmp_0 := i + Product3_tmp;
                 Product3[Product3_tmp_0] := 0.0;
                 Product3_tmp_1 := i_0 * 2;
                 Product3_tmp := Product3_tmp + i;
                 Product3[Product3_tmp_0] := Product3[Product3_tmp] +
(K_Value[Product3_tmp_1] * B_Value[i]);
                 Product3[Product3_tmp_0] := (K_Value[Product3_tmp_1 + 1] * B_Value[i +
4]) + Product3[Product3_tmp];
             END_FOR;
             (* End of Product: '<S2>/Product3' *)
         END_FOR;
         (* Sum: '<S2>/Sum2' incorporates:
          *  Constant: '<S2>/            A            ' *)
         FOR i := 0 TO 15 DO
             Sum2[i] := A_Value[i] - Product3[i];
         END_FOR;
         (* End of Sum: '<S2>/Sum2' *)
         FOR i := 0 TO 3 DO
             (* Outport: '<Root>/reference_states' incorporates:
              *  UnitDelay: '<S2>/Unit Delay' *)
             reference_states[i] := UnitDelay_DSTATE[i];
             (* Product: '<S2>/Product1' incorporates:
```

```
               *  Constant: '<S2>/      B   ' *)
           Product1[i] := 0.0;
           Product1[i] := (B_Value[i] * -6.1000000000000005) + Product1[i];
           Product1_tmp := B_Value[i + 4];
           Product1[i] := (Product1_tmp * 3162.3) + Product1[i];
           (* Product: '<S2>/Product4' incorporates:
            *  SignalConversion: '<S2>/TmpSignal ConversionAtProduct4Inport2' *)
           tmp := Product1[i] * Efd;
           (* Product: '<S2>/Product1' incorporates:
            *  Constant: '<S2>/      B   '
            *  Constant: '<S2>/K2' *)
           Product1[i + 4] := 0.0;
           Product1[i + 4] := Product1[i + 4] + (B_Value[i] * 3162.3);
           Product1[i + 4] := (Product1_tmp * 6.1000000000000005) + Product1[i + 4];
           (* Product: '<S2>/Product4' incorporates:
            *  SignalConversion: '<S2>/TmpSignal ConversionAtProduct4Inport2' *)
           tmp := (Product1[i + 4] * Pm) + tmp;
           (* Product: '<S2>/Product' incorporates:
            *  UnitDelay: '<S2>/Unit Delay' *)
           Product1_tmp := (Sum2[i + 12] * UnitDelay_DSTATE[3]) + ((Sum2[i + 8] *
UnitDelay_DSTATE[2]) + ((Sum2[i + 4] * UnitDelay_DSTATE[1]) + (Sum2[i] *
UnitDelay_DSTATE[0])));
           (* Update for DiscreteIntegrator: '<S2>/Discrete-Time Integrator'
incorporates:
            *  Sum: '<S2>/Sum1' *)
           c_DiscreteTimeIntegrator_DS[i] := ((tmp + Product1_tmp) * 5.0E-5) +
c_DiscreteTimeIntegrator_DS[i];
       END_FOR;
       (* Update for UnitDelay: '<S2>/Unit Delay' *)
       UnitDelay_DSTATE[0] := rtb_DiscreteTimeIntegrator[0];
       UnitDelay_DSTATE[1] := rtb_DiscreteTimeIntegrator[1];
       UnitDelay_DSTATE[2] := rtb_DiscreteTimeIntegrator[2];
       UnitDelay_DSTATE[3] := rtb_DiscreteTimeIntegrator[3];
END_CASE;
```

APPENDIX Q: Modified generated Structured Text for the SIMULINK MRAC component

```
(*
 *
 * File: synch_generator_adaptive_control_plc_coder_2.st
 *
 * IEC 61131-3 Structured Text (ST) code generated for subsystem
"synch_generator_adaptive_control_plc_coder_2/MRAC"
 *
 * Model name                        : synch_generator_adaptive_control_plc_coder_2
 * Model version                     : 1.25
 * Model creator                     : pierr
 * Model last modified by            : Administrateur SEL
 * Model last modified on            : Sun Aug 15 16:14:51 2021
 * Model sample time                 : 5e-05s
 * Subsystem name                    : synch_generator_adaptive_control_plc_coder_2/MRAC
 * Subsystem sample time             : 5e-05s
 * Simulink PLC Coder version        : 2.6 (R2018b) 24-May-2018
 * ST code generated on              : Sun Aug 15 16:17:29 2021
 *
 * Target IDE selection              : Generic
 * Test Bench included               : No
 *
 *)
FUNCTION_BLOCK MRAC
VAR_INPUT
```

```
    ssMethodType: DINT;
    Efd: LREAL;
    Pm: LREAL;
    generator_states: ARRAY [0..3] OF LREAL;
    error_dynamics: ARRAY [0..3] OF LREAL;
END_VAR
VAR_OUTPUT
    Un: ARRAY [0..1] OF LREAL;
    Ua: ARRAY [0..1] OF LREAL;
END_VAR
VAR
    c_DiscreteTimeIntegrator_DS: ARRAY [0..7] OF LREAL;
    u_Value: ARRAY [0..7] OF LREAL := [0.0,0.0,0.0,0.125,0.0,0.21,0.0,0.0];
    b: ARRAY [0..15] OF LREAL :=
[2.8966,2.1881,1.1965,2.1551,2.1881,1.9966,0.6827,1.8861,1.1965,0.6827,0.759,0.5348,
      2.1551,1.8861,0.5348,2.0955];
      // Create instance of the ReferenceModel function block
      reference_model_with_lqr : reference_model;
END_VAR
VAR_TEMP
    rtb_Transpose2: ARRAY [0..7] OF LREAL;
    i: DINT;
      i_1: SINT;
    generator_states_0: ARRAY [0..15] OF LREAL;
    generator_states_1: ARRAY [0..15] OF LREAL;
    generator_states_2: ARRAY [0..7] OF LREAL;
    Ua_0: LREAL;
    rtb_Transpose2_tmp: DINT;
    generator_states_tmp: DINT;
    generator_states_tmp_0: DINT;
    generator_states_tmp_1: DINT;
END_VAR
```

```
CASE ssMethodType OF
    0:
        (* SystemInitialize for Atomic SubSystem: '<S1>/Adaptation' *)
        (* InitializeConditions for DiscreteIntegrator: '<S2>/Discrete-Time Integrator' *)
        FOR i := 0 TO 7 DO
            c_DiscreteTimeIntegrator_DS[i] := 0.0;
        END_FOR;
        (* End of InitializeConditions for DiscreteIntegrator: '<S2>/Discrete-Time Integrator' *)

            FOR i := 0 TO 1 DO
                    // Initialize adaptive outputs
                    Ua[i] := 0.0;
                    // Initialize nominal outputs
                    Un[i] := 0.0;
            END_FOR

            FOR i := 0 TO 3 DO
                    // Initialize the `generator_states` array
                    generator_states[i] := 0.0;
                    // Initialize the error dynamics `error_dynamics`
                    error_dynamics[i] := 0.0;
            END_FOR

        (* End of SystemInitialize for SubSystem: '<S1>/Adaptation' *)
    1:
        (* Outport: '<Root>/Un' incorporates:
         *  Constant: '<S1>/K2'
         *  Product: '<S1>/Product4'
         *  SignalConversion: '<S1>/TmpSignal ConversionAtProduct4Inport2' *)

            // Create the generator states vector
            generator_states[0] := RTDS_CTRL_GENERATOR_DATA.MV001.instMag;
            generator_states[1] := RTDS_CTRL_GENERATOR_DATA.MV002.instMag;
            generator_states[2] := RTDS_CTRL_GENERATOR_DATA.MV003.instMag;
```

```
                generator_states[3] := RTDS_CTRL_GENERATOR_DATA.MV004.instMag;

                // Call the reference model function block and assign inputs.
                reference_model_with_lqr(ssMethodType:=1,
                                                    Efd:= RTDS_CTRL_GENERATOR_DATA.MV005.instMag,
                                                    Pm:= RTDS_CTRL_GENERATOR_DATA.MV006.instMag);

                // Compute the error dynamics
                FOR i := 0 TO 3 DO
                    error_dynamics[i] := reference_model_with_lqr.reference_states[i] -
generator_states[i];
                END_FOR

        Un[0] := (-6.1000000000000005 * Efd) + (3162.3 * Pm);
        Un[1] := 3162.3 * Efd;
        Un[1] := (6.1000000000000005 * Pm) + Un[1];
        (* Outputs for Atomic SubSystem: '<S1>/Adaptation' *)
        (* MATLAB Function: '<S2>/Parameter Estimation' *)
        (* MATLAB Function 'MRAC/Adaptation/Parameter Estimation': '<S3>:1' *)
        (* '<S3>:1:2' n = size(B); *)
        (* '<S3>:1:3' phi = zeros(n); *)
        (* '<S3>:1:5' P = [2.8966    2.1881    1.1965    2.1551; *)
        (* '<S3>:1:6'      2.1881    1.9966    0.6827    1.8861; *)
        (* '<S3>:1:7'      1.1965    0.6827    0.7590    0.5348; *)
        (* '<S3>:1:8'      2.1551    1.8861    0.5348    2.0955]; *)
        (* '<S3>:1:10' phi = X * Error' * P * B; *)
        (* phi =  B' * (B*B')^-1 * phi1; *)
        (* phi = X * Error' * P * B' * (B*B')^-1;  *)
        (* End of Outputs for SubSystem: '<S1>/Adaptation' *)
        (* Outputs for Atomic SubSystem: '<S1>/Adaptation' *)
        FOR i := 0 TO 3 DO
            (* Math: '<S2>/Transpose2' incorporates:
             *  Constant: '<S2>/learning_rate'
             *  DiscreteIntegrator: '<S2>/Discrete-Time Integrator'
             *  Product: '<S2>/Product' *)
            rtb_Transpose2_tmp := i * 2;
            rtb_Transpose2[rtb_Transpose2_tmp] := c_DiscreteTimeIntegrator_DS[i] * -100.0;
            rtb_Transpose2[1 + rtb_Transpose2_tmp] := c_DiscreteTimeIntegrator_DS[i + 4] * -
100.0;
            (* MATLAB Function: '<S2>/Parameter Estimation' incorporates:
             *  Constant: '<S2>/          1' *)
            generator_states_0[i] := generator_states[i] * error_dynamics[0];
            generator_states_0[i + 4] := generator_states[i] * error_dynamics[1];
            generator_states_0[i + 8] := generator_states[i] * error_dynamics[2];
            generator_states_0[i + 12] := generator_states[i] * error_dynamics[3];
            FOR rtb_Transpose2_tmp := 0 TO 3 DO
                generator_states_tmp := rtb_Transpose2_tmp * 4;
                generator_states_tmp_0 := i + generator_states_tmp;
                generator_states_1[generator_states_tmp_0] := 0.0;
                generator_states_tmp_1 := generator_states_tmp + i;
                generator_states_1[generator_states_tmp_0] :=
generator_states_1[generator_states_tmp_1] + (b[generator_states_tmp] * generator_states_0[i]);
                generator_states_1[generator_states_tmp_0] := (b[generator_states_tmp + 1] *
generator_states_0[i + 4]) + generator_states_1[generator_states_tmp_1];
                generator_states_1[generator_states_tmp_0] := (b[generator_states_tmp + 2] *
generator_states_0[i + 8]) + generator_states_1[generator_states_tmp_1];
                generator_states_1[generator_states_tmp_0] := (b[generator_states_tmp + 3] *
generator_states_0[i + 12]) + generator_states_1[generator_states_tmp_1];
            END_FOR;
            FOR rtb_Transpose2_tmp := 0 TO 1 DO
                generator_states_tmp := rtb_Transpose2_tmp * 4;
                generator_states_tmp_0 := i + generator_states_tmp;
                generator_states_2[generator_states_tmp_0] := 0.0;
                generator_states_tmp_1 := generator_states_tmp + i;
```

249

```
                generator_states_2[generator_states_tmp_0] :=
generator_states_2[generator_states_tmp_1] + (u_Value[generator_states_tmp] *
generator_states_1[i]);
                generator_states_2[generator_states_tmp_0] := (u_Value[generator_states_tmp + 1]
* generator_states_1[i + 4]) + generator_states_2[generator_states_tmp_1];
                generator_states_2[generator_states_tmp_0] := (u_Value[generator_states_tmp + 2]
* generator_states_1[i + 8]) + generator_states_2[generator_states_tmp_1];
                generator_states_2[generator_states_tmp_0] := (u_Value[generator_states_tmp + 3]
* generator_states_1[i + 12]) + generator_states_2[generator_states_tmp_1];
            END_FOR;
        END_FOR;
        (* Update for DiscreteIntegrator: '<S2>/Discrete-Time Integrator' *)
        FOR i := 0 TO 7 DO
            c_DiscreteTimeIntegrator_DS[i] := (5.0E-5 * generator_states_2[i]) +
c_DiscreteTimeIntegrator_DS[i];
        END_FOR;
        (* End of Update for DiscreteIntegrator: '<S2>/Discrete-Time Integrator' *)
        (* End of Outputs for SubSystem: '<S1>/Adaptation' *)
        (* Outport: '<Root>/Ua' incorporates:
         *  Inport: '<Root>/generator_states'
         *  Product: '<S2>/Product1' *)
        FOR i := 0 TO 1 DO
            (* Outputs for Atomic SubSystem: '<S1>/Adaptation' *)
            Ua_0 := (rtb_Transpose2[i + 6] * generator_states[3]) + ((rtb_Transpose2[i + 4] *
generator_states[2]) + ((rtb_Transpose2[i + 2] * generator_states[1]) + (rtb_Transpose2[i] *
generator_states[0])));
            Ua[i] := Ua_0;
            (* End of Outputs for SubSystem: '<S1>/Adaptation' *)
        END_FOR;
        (* End of Outport: '<Root>/Ua' *)
END_CASE;
```

APPENDIX R: Structured Text code of the MRAC implementation

```
PROGRAM interarea_oscillations_damping
VAR
        i_0: DINT;
        i_1: DINT;
        //i_2: DINT;

        // Create instance of the ReferenceModel function block
        reference_model_with_lqr : reference_model;

        // Create an instance of the MRAC function block
        mrac_adaptation : MRAC;

        // Output arrays of both the reference model and the synchronous generator model in the
RTDS
        ref_states: ARRAY [0..3] OF LREAL;
        gen_states: ARRAY [0..3] OF LREAL;

        // Error dynamics
        error_dynamics: ARRAY [0..3] OF LREAL;

        // Control signal to be sent back to the RTDS via IEC61850
        adaptation: ARRAY [0..1] OF LREAL;
        nominal_controller: ARRAY [0..1] OF LREAL;
        iec61850_control_signal: ARRAY [0..1] OF LREAL;

END_VAR

VAR_TEMP
    i_4: DINT;
END_VAR
```

```
// Call the reference model function block and assign inputs.
reference_model_with_lqr(ssMethodType:=1,
                                    Efd:=RTDS_CTRL_GENERATOR_DATA.MV005.instMag,
                                    Pm:=RTDS_CTRL_GENERATOR_DATA.MV006.instMag);


FOR i_0 := 0 TO 3 DO
      // Initialize the error dynamics array
      error_dynamics[i_0] := 0.0;
      // Initialize the `gen_states` array
      gen_states[i_0] := 0.0;
      // Initialize the `ref_states` array
      ref_states[i_0] := 0.0;
END_FOR

// The function blocs will compute both the reference model states and controlled signal
ref_states := reference_model_with_lqr.reference_states;

// Create the generator states vector
gen_states[0] := RTDS_CTRL_GENERATOR_DATA.MV001.instMag;
gen_states[1] := RTDS_CTRL_GENERATOR_DATA.MV002.instMag;
gen_states[2] := RTDS_CTRL_GENERATOR_DATA.MV003.instMag;
gen_states[3] := RTDS_CTRL_GENERATOR_DATA.MV004.instMag;

// Compute the error dynamics
FOR i_0 := 0 TO 3 DO
      error_dynamics[i_0] := ref_states[i_0] - gen_states[i_0];
END_FOR

// Call the adaptation function block and assign inputs
FOR i_1 := 0 TO 1 DO
      mrac_adaptation(ssMethodType:=i_1,
                                    Efd:=RTDS_CTRL_GENERATOR_DATA.MV005.instMag,
                                    Pm:=RTDS_CTRL_GENERATOR_DATA.MV006.instMag,
                                    generator_states:=gen_states,
                                    error_dynamics:=error_dynamics);
END_FOR

mrac_adaptation(ssMethodType:=1,
                                    Efd:=RTDS_CTRL_GENERATOR_DATA.MV005.instMag,
                                    Pm:=RTDS_CTRL_GENERATOR_DATA.MV006.instMag,
                                    generator_states:=gen_states,
                                    error_dynamics:=error_dynamics);

// Get the two components of the control signal
adaptation := mrac_adaptation.Ua;
nominal_controller := mrac_adaptation.Un;

// Assign control data to relevant tags for GOOSE publishing
SEL_RTAC_1.LD1.GGIO1.AnIn001.instMag := adaptation[0];
SEL_RTAC_1.LD1.GGIO1.AnIn002.instMag := adaptation[1];
```