REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
الجمهورية الجزائرية الديمقراطية الشعبية

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
HIGHER SCHOOL IN APPLIED SCIENCES
–T L E M C E N–

المدرسة العليا في العلوم التطبيقية
École Supérieure en Sciences Appliquées

وزارة التعليم العالي والبحث العلمي
المدرسة العليا في العلوم التطبيقية
- تلمسان -

Mémoire de fin d'étude

Pour l'obtention du diplôme de Master

Filière : Électrotechnique
Spécialité : Énergie et environnement

Présenté par : **Mohamed ADDA**

Thème

Intelligent Control of Doubly-Fed Induction Generators.

Soutenu publiquement, le 01/07/2024 , devant le jury composé de:

M. S. BELAROUCI	MCB	ESSA. Tlemcen	Président
M. M. AIT-AHMED	Professeur	POLYTECH Nantes	Examinateur 01
M. A. TAHOUR	Professeur	ESSA. Tlemcen	Examinateur 02
M. A.K. CHEMIDI	MCA	ESSA. Tlemcen	Encadrant
M. Lo. MERAD	Professeur	ESSA. Tlemcen	Co-Encadrant

Année universitaire : **2023 /2024**

Dedication

“

À mon père et ma mère, pour leur amour infini, leur soutien indéfectible et leurs sacrifices incessants qui m'ont permis de rêver et d'atteindre mes objectifs. Votre foi en moi a toujours été ma lumière dans les moments de doute,

À mon cher frère et ma précieuse sœur, votre présence a toujours été ma source d'inspiration. Que notre lien fraternel demeure un phare éclatant, nous guidant vers un avenir radieux,

À mes amis de toujours, et à mes estimés collègues de la Promotion Electrotechnique 2021. Vous avez été des compagnons indispensables dans ce voyage, offrant soutien, camaraderie, et de précieux souvenirs,

À tous ceux qui me sont chers, à vous tous,

Merci.

”

- M.

Acknowledgments

First and foremost, I am profoundly grateful to Almighty God for granting me the strength, perseverance, and patience to navigate through my years of study successfully, culminating in the completion of this thesis.

I extend my heartfelt thanks to my thesis supervisor, **Mr. CHEMIDI Abdelkarim**, and co-supervisor, **Mr. MERAD Lotfi**, for their invaluable assistance, patience, and encouragement. Their insightful feedback and critical perspectives were essential in shaping this thesis and enhancing its quality. Their mentorship, professional advice, and unwavering support have been truly remarkable.

I am deeply grateful to the members of the jury for honoring me by taking the time to read and evaluate this thesis. My sincere thanks go to **Mr. BELAROUCI Salim**, Assistant Lecturer at ESSA Tlemcen, for accepting the presidency of the jury. I extend my sincere gratitude to **Mr. AIT-AHMED Mourad**, Professor at POLYTECH Nantes - FRANCE . His willingness to join us and evaluate my work is a privilege that I greatly appreciate. I am also very thankful to **Mr. TAHOUR Ahmed**, Professor at ESSA Tlemcen, for his interest in this work and for agreeing to serve on the jury.

I would also like to express my appreciation to the entire professorial team at **ESSA Tlemcen** for their dedicated efforts in providing us with a quality education.

Finally, I wish to thank everyone who contributed, directly or indirectly, to the realization of this work. Your support and encouragement have been invaluable throughout this journey.

Abstract

This thesis investigates the intelligent flux-oriented control of active and reactive powers in a Doubly Fed Induction Generator (DFIG) to optimize its performance. The study begins with a comprehensive review of the state-of-the-art in Doubly-fed induction machines, focusing specifically on generators. It then gets into Artificial Neural Networks, first as a concept and then as an advanced control strategy for our system. After that, it also explores the Adaptive Neuro-Fuzzy Inference System (ANFIS), theoretically, and then its application to our control system in a similar manner.

Various tests were conducted to evaluate the efficiency and robustness of both control strategies. At the end, a comparative analysis was performed to highlight the strengths and weaknesses of each approach. Simulation results in the MATLAB/SIMULINK environment demonstrate that both controllers deliver excellent performance in term of robustness. However, ANFIS exhibits a slight edge regarding response time and robustness.

The findings underscore the potential of intelligent control techniques in optimizing DFIG-based wind turbines; This improvement is attributed to the adaptive and flexible nature of intelligent controllers, which better handle the complexities and uncertainties inherent in wind energy systems.

Keywords : Doubly-fed induction generator (DFIG), flux-oriented control, Artificial Neural Networks, MLP, Neuro-fuzzy, ANFIS.

Résumé

Cette thèse étudie la commande vectorielle intelligente des puissances active et réactive dans une Génératrice Asynchrone à Double Alimentation (GADA) pour optimiser ses performances. L'étude débute par une revue complète de l'état de l'art des machines asynchrones à double alimentation, en se concentrant spécifiquement sur les géneratrices. Elle aborde ensuite les réseaux de neurones artificiels, d'abord en tant que concept, puis en tant que stratégie de contrôle avancée pour notre système. Après cela, elle explore également le système d'inférence neuro-flou adaptatif (ANFIS), théoriquement, puis son application à notre système de contrôle de manière similaire.

Divers tests ont été menés pour évaluer l'efficacité et la robustesse des deux stratégies de contrôle. Une analyse comparative a été réalisée à la fin pour mettre en évidence les forces et les faiblesses de chaque approche. Les résultats de simulation dans l'environnement MATLAB/SIMULINK montrent que les deux contrôleurs offrent d'excellentes performances en termes de robustesse. Cependant, l'ANFIS présente un léger avantage, notamment en ce qui concerne le temps de réponse et la robustesse.

Les résultats soulignent le potentiel des techniques de contrôle intelligent pour optimiser les éoliennes à base de DFIG, Cette amélioration est attribuée à la nature adaptative et flexible des contrôleurs intelligents, qui permettent une meilleure gestion des complexités et des non-linéarités inhérentes aux systèmes d'énergie éolienne.

Mots clés : Generatrice asynchrone à double alimentation (GADA), Commande vectorielle, Réseaux de neurones artificiels, MLP, Neuro-flou, ANFIS.

ملخص

تتناول هذه الأطروحة التحكم الشعاعي الذكي للقدرة الفعالة والقدرة غير الفعالة للمولد اللاتزامني مزدوج التغذية (DFIG) بهدف تحسين أدائه. تبدأ الدراسة بمراجعة شاملة لأحدث التطورات في آلات التحريض ذات التغذية المزدوجة، مع التركيز بشكل خاص على المولدات. ثم تتناول الشبكات العصبية الاصطناعية، أولاً كمفهوم ثم كاستراتيجية تحكم متقدمة لنظامنا. بعد ذلك، تستكشف نظام الاستدلال العصبي الضبابي التكيفي (ANFIS)، نظريًا ثم بتطبيقه على نظام التحكم لدينا بطريقة مماثلة.

تم إجراء اختبارات متنوعة لتقييم كفاءة و صلابة استراتيجيات التحكم المذكورة. وفي النهاية، قمنا بدراسة مقارنة لتسليط الضوء على نقاط القوة والضعف في كل منهج. تُظهر نتائج المحاكاة في بيئة MATLAB/SIMULINK أن كلا من وحدات التحكم تقدم أداءً ممتازًا. ومع ذلك، يُظهر ANFIS ميزة طفيفة من حيث زمن الاستجابة والصلابة.

تؤكد النتائج على إمكانات تقنيات التحكم الذكية في تحسين توربينات الرياح القائمة على المولد اللاتزامني مزدوج التغذية، يرجع هذا التحسن إلى الطبيعة المرنة والقابلة للتكيف للمتحكمات الذكية، التي تُمكن من التعامل الأفضل مع التعقيدات والعوامل غير الخطية الطبيعية في أنظمة الطاقة الرياحية.

كلمات مفتاحية: مولد التحريض ذو التغذية المزدوجة (DFIG)، التحكم الشعاعي ، الشبكات العصبونية الاصطناعية، النظام العصبي الضبابي،MLP ، ANFIS.

Table of Contents

List of Figures

List of Tables

List of Abbreviations and Terms

DFIM	*Doubly-fed Induction Machine*
DFIG	*Doubly-fed Induction Generator*
ANN	*Artificial Neural Network*
MLP	*Multilayer Perceptron*
FIS	*Fuzzy Inference System*
MF	*Membership Function*
ANFIS	*Adaptive Neuro-Fuzzy Inference System*
RMSE	*Root Mean Square Error*
R	*Regression Coefficient*
θ	*Electrical Position Angle*
R_r	*Rotor Resistance*
R_s	*Stator Resistance*
L_r	*Rotor Inductance*
L_s	*Stator Inductance*
s	*Slip*
V_r	*Rotor Voltage*
V_s	*Stator Voltage*

M	*Mutual Inductance*
I_r	*Rotor Current*
I_s	*Stator Current*
P_s	*Stator Active Power*
Q_s	*Stator Reactive Power*
ϕ_r	*Rotor Flux*
ϕ_s	*Stator Flux*
p	*Number of Pole Pairs*
T_e	*Electromagnetic Torque*
T_r	*Resistant Torque*
f	*Friction Coefficient*
Ω_m	*Mechanical Rotation Speed*
J	*Machine Inertia*

General Introduction

Context

The global shift towards renewable energy sources has highlighted the importance of advanced technologies for improving efficiency and adaptability. Among these, the Doubly Fed Induction Generator **(DFIG)** stands out, particularly in wind energy conversion. Renowned for its versatility in operating across a broad range of wind speeds, the DFIG is a cornerstone in modern variable-speed wind turbines, enhancing energy conversion processes and promoting sustainable solutions.

In recent years, the adoption of DFIGs has surged due to their ability to manage varying operational speeds and conditions efficiently. This adaptability not only boosts the efficiency of wind energy systems but also reduces operational complexities and costs by minimizing reliance on heavy energy converters. This context focuses on advanced control strategies for DFIGs, aiming to streamline their operation and maximize efficiency in wind energy conversion.

Problem Statement

Precisely controlling the Doubly Fed Induction Generator (DFIG) is crucial for optimizing wind turbine performance. Despite its advantages, such as reduced operational costs and enhanced energy output, the DFIG's dynamic and complex nature requires sophisticated control strategies. Active power control directly impacts the turbine's energy output and efficiency, while reactive power control is essential for maintaining voltage stability and power quality in the electrical grid. The interactions between active and reactive power flows in DFIGs present significant challenges, especially under varying wind conditions and grid disturbances. These issues highlight the need for advanced control strategies to adjust the powers dynamically, ensuring efficient energy conversion, enhanced grid support, and extended generator lifespan.

Objectives

This thesis aims to enhance the control of Doubly Fed Induction Generators (DFIGs) in variable-speed wind turbines by developing and validating advanced intelligent control strategies. Focusing on Artificial Neural Networks (ANN) and Adaptive Neuro-Fuzzy Inference Systems (ANFIS), the study seeks to address the operational complexities of DFIGs, improving their dynamic performance, robustness, and overall efficiency. Through extensive simulation in the MATLAB/SIMULINK environment, in the aim of contributing to optimizing wind energy conversion and advancing renewable energy technologies.

Structure & Organization

Organized into three main chapters, this thesis thoroughly examines various aspects of DFIG control, contrasting classical approaches with advanced methods such as Artificial Neural Networks (ANN) and Adaptive Neuro-Fuzzy Inference Systems (ANFIS):

Chapter 1 establishes the foundation by providing a comprehensive review of the state-of-the-art in doubly-fed induction machines, focusing specifically on generators. It covers configurations of DFIGs, their operating modes, and working principles. Additionally, it discusses the applications of DFIGs and their modeling, including simplifications and the ABC model.

Chapter 2 provides an in-depth analysis of artificial neural networks for controlling DFIGs. It covers the basics of neural networks, including historical context, network architectures, and learning methods. The chapter then focuses on the development and implementation of ANN-based controllers, presenting simulation results and performance evaluations.

Chapter 3 explores the ANFIS approach for controlling DFIGs. It begins with an introduction to fuzzy logic and neural networks, followed by a detailed explanation of the ANFIS architecture and learning algorithms. The chapter then discusses the application of ANFIS to DFIG control, presenting simulation results and comparing the performance of ANFIS with the previous ANN controller and the traditional PI also.

At the end, a general conclusion summarizes the work conducted throughout the thesis and offers insights into potential future research directions.

Chapter 1

Doubly-Fed Induction Generator (DFIG): State-of-the-art.

1.1 Introduction

The Doubly-Fed Induction Machine (DFIM) has been the subject of extensive research, particularly in its operation as a generator (DFIG) for renewable energy applications. Its ability to efficiently harness wind energy and convert it into electrical Power has made it a cornerstone of modern wind turbine technology.

In this chapter, we will present the state-of-the-art DFIG systems used in various applications, configurations, and operational modes in wind turbines. We introduce a comprehensive dynamic model of its components, starting with the main mathematical equations that define the relationship between voltage and fluxes within the generator. The Park transformation and the selection of reference frames are also discussed, as they are crucial for simplifying the machine's dynamic equations, making the analysis and control of the DFIG more intuitive and efficient. Then, we will explore control strategies to optimize the efficiency and reliability of wind energy generation.

1.2 Description of the Doubly-fed induction generator (DFIG)

A doubly-fed induction machine (DFIM) is nothing more than a wound rotor induction machine. The term "doubly-fed" indicates that the machine is connected to the grid through its stator and rotor (Figure 1.1), with this connection being made either without or with static converters, leading to various structures. It was used for years in the past for applications requiring speed control. However, working as a generator, this machine enables an important feature: it can produce Power at both subsynchronous and supersynchronous speeds. Because of this trait, wind turbines can operate efficiently in various velocities. [1]

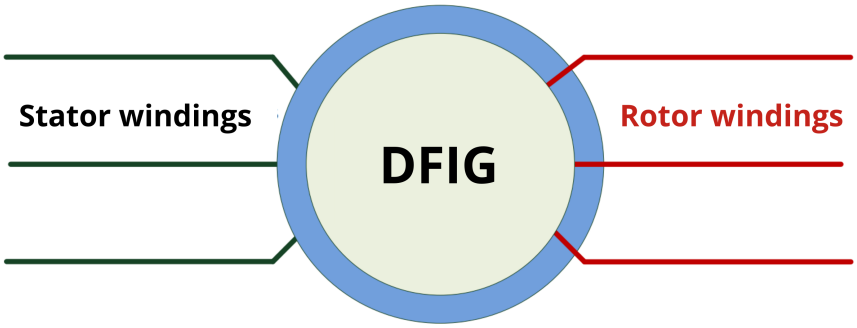

Figure 1.1: Symbol of the Doubly-fed induction machine.

A DFIG has a stator identical to a squirrel cage induction or synchronous machine. The rotor differs significantly, as it is not composed of magnets or a squirrel cage but of three-phase windings arranged similarly to the stator windings. (Figure 1.2)

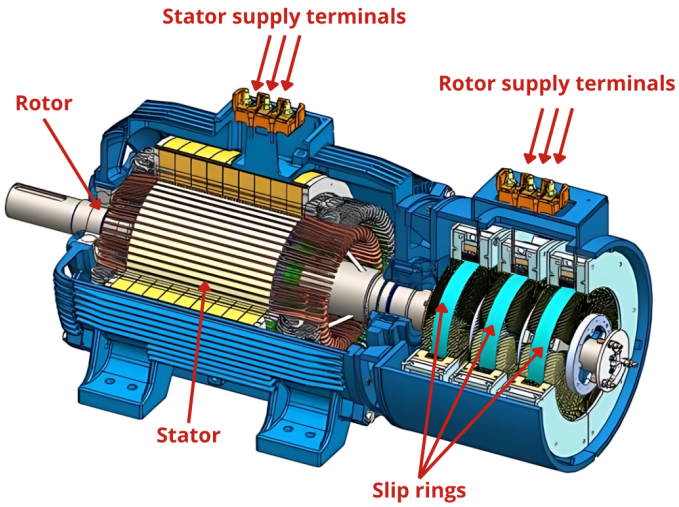

Figure 1.2: Structure of the Doubly-fed induction machine. [2]

As shown in Figure (1.3), the rotor windings are connected in a star configuration, and the three phases are connected to a system of sliding contacts (brushes and collector rings) that allow access to the rotor voltages and currents.

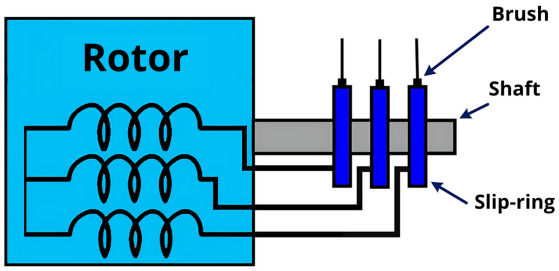

Figure 1.3: Wound rotor principal.

1.3 Configurations of Doubly-Fed Induction Machines

Today, the Doubly-Fed Induction Machine (DFIM) is the most commonly used variable-speed machine in wind energy production due to its characteristics. The DFIM with a wound rotor has a three-phase stator identical to conventional induction machines but with a different rotor design. In this design, the rotor windings are connected in a star configuration, and the ends are connected to slip-rings. Brushes rub against these rings during machine operation. While the stator windings are directly connected to the grid, the rotor windings pass through bidirectional power converters to control slip variation.

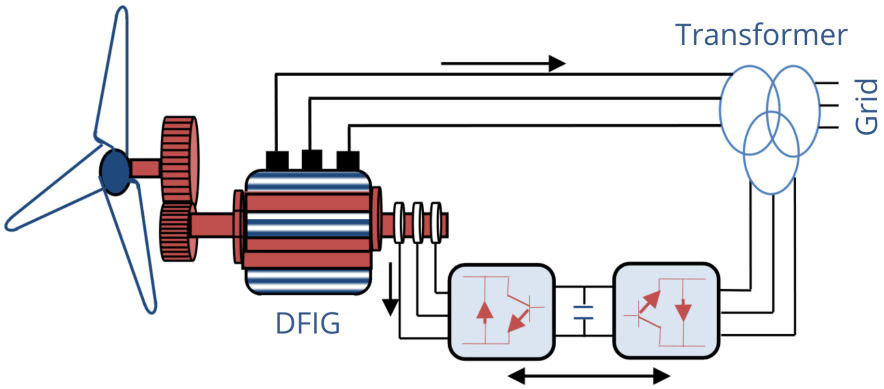

Figure 1.4: Standard wounded rotor DFIG for Wind turbine. [3]

In this system, the stator is directly connected to the power grid, while the rotor is connected to a voltage-source converter (back-to-back), serving as a frequency converter. The term "Doubly-fed" refers to the stator voltage drawn from the grid and the rotor voltage provided by the converter. This setup enables variable-speed operation within a specific operating range. The converter compensates for the difference between mechanical and electrical frequencies by injecting a variable-frequency current into the rotor. (Figure 1.4)

1.3.1 DFIG with Rotor Energy Dissipation

Figure (1.5) illustrates the technology that allows for a limited speed variation of approximately 10% around the synchronous speed by changing the rotor resistance. The stator is directly connected to the grid, and the rotor is connected to a rectifier. [4] Besides the limited speed variation range, this solution's drawback is the dissipation of rotor power in the rotor resistance in the event of a high slip.

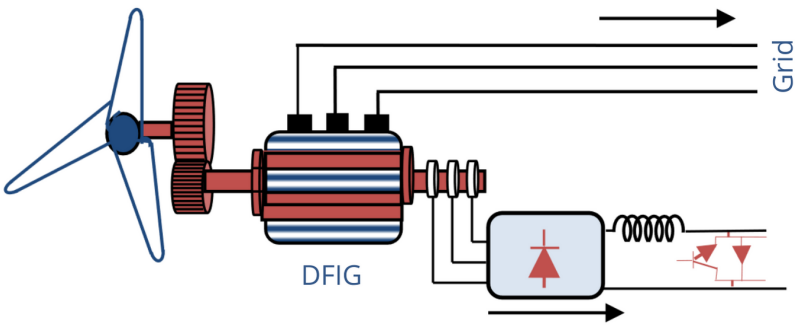

Figure 1.5: DFIG with slip control through dissipation of rotor energy. [3]

1.3.2 DFIG - Krammer structure

To enhance the efficiency of the previous system configuration, the structure (Figure 1.6) employs both a diode bridge and a thyristor bridge. The diode bridge rectifies the voltages between rings. The thyristor inverter applies a variable voltage to this rectifier by controlling the firing angle of the thyristors. This setup allows for adjusting the conduction range of the diodes, thereby varying the Power extracted from the rotor circuit and, consequently, the slip of the asynchronous generator. However, a drawback of this configuration is that the power supply does not allow for the electrical speed control of the machine. Additionally, the three-phase inverter used in this structure injects significant amplitude low-frequency harmonic currents. [5]

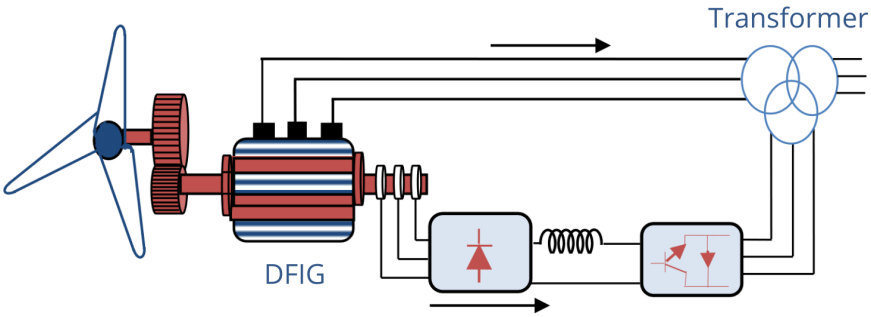

Figure 1.6: DFIG Krammer structure. [3]

1.3.3 DFIG - Scherbius Structure with Cycloconverter

To enable bidirectional energy flow between the rotor and the grid, this topology replaces the rectifier-inverter combination with a cycloconverter. The speed variation range is doubled compared to the Kramer structure. [5]

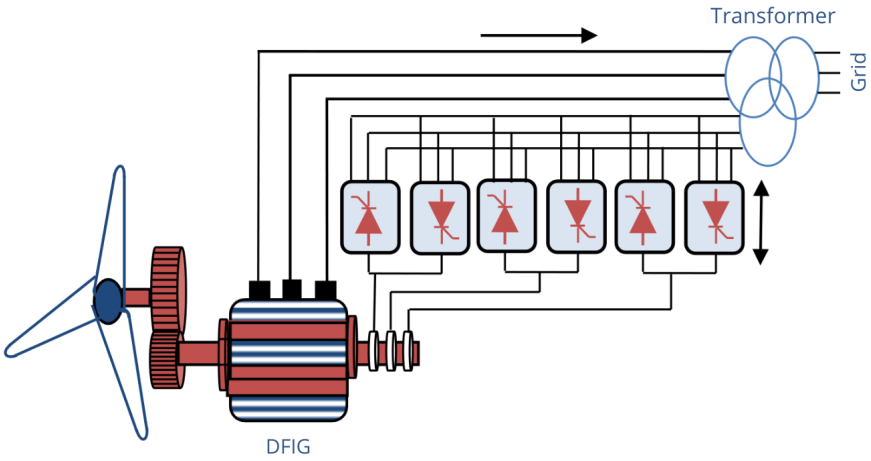

Figure 1.7: DFIG Scherbius Structure with Cycloconverter. [3]

This setup is also known as the Scherbius static topology. Formally, the Scherbius principle is based on the use of rotating machines instead of power converters. (Figure 1.7)

1.3.4 DFIG - Scherbius Structure with PWM converters

This configuration involves using two IGBT-based three-phase bridges controlled by pulse width modulation (PWM). This choice allows for two degrees of freedom for each converter: control over the flux and rotational speed of the asynchronous generator on the machine side and control over the active and reactive powers transferred on the grid side. [5] [4] (Figure 1.8)

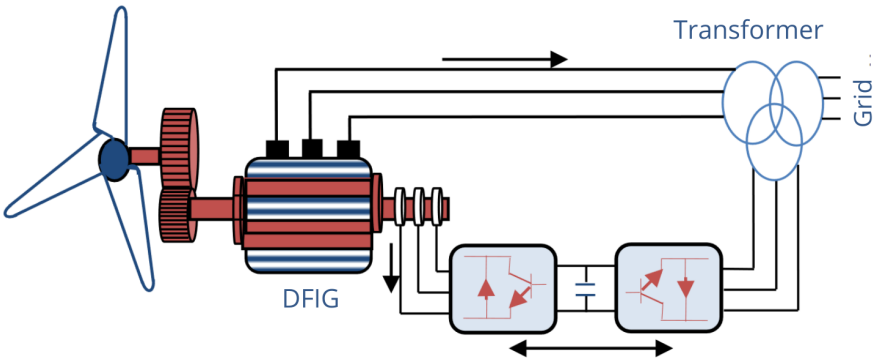

Figure 1.8: DFIG Scherbius Structure with PWM. [3]

However, it is noteworthy that operating the grid-side inverter with PWM allows for a more precise extraction of currents.

There are several configurations of doubly-fed induction machines, with the most prominent ones outlined above. Table (1.1) summarizes the primary distinctions between these configurations.

Table 1.1: Main Characteristics of Different Structures of the Doubly Fed Induction Machine (DFIM). [5]

Technology	**Type of DFIM**	**Converter Used**	**Power Transfer**	**Rotor Speed Variation**
Brushless machine with double fed induction	Double winding on the stator	PWM converter on the stator and grid side	Bidirectional transfer of slip energy	Variable 25%
Double fed by rotor and stator, with energy dissipation	Single winding on the stator and wound rotor	Rectifier for slip control with chopper connected to a resistive load	Unidirectional transfer of slip energy	Variable 25%
Double fed by rotor and stator, with slip energy recovery (Kramer system)	Single winding on the stator and wound rotor	Diode rectifier and thyristor inverter	Unidirectional transfer of slip energy	Variable 25%
Double fed by rotor and stator, with slip energy recovery	Single winding on the stator and wound rotor	Scherbius cycloconverter	Bidirectional transfer of slip energy	Variable 50%
Double fed by rotor and stator, with slip energy recovery	Single winding on the stator and wound rotor	Dual PWM converter	Bidirectional transfer of slip energy	Variable 50%

1.4 Operating Modes of DFIG

The slip s is defined as the ratio of the difference between the synchronous angular frequency ω_s and the electrical angular frequency of the rotor ω_r, to ω_s:

$$s = \frac{\omega_s - \omega_r}{\omega_s} \tag{1.1}$$

According to the slip s, three different operating modes are possible for the DFIG [6]:

- Sub-synchronous operation: $\omega_r < \omega_s \Rightarrow s > 0$
- Synchronous operation: $\omega_r = \omega_s \Rightarrow s = 0$
- Super-synchronous operation: $\omega_r > \omega_s \Rightarrow s < 0$

1.4.1 Subsynchronous Motor Operation

In subsynchronous motor operation, Power is supplied by the grid to the stabilizer, while slip power passes through the rotor and is reintroduced to the grid. Thus, the operation occurs below the synchronous speed, as depicted in Figure (1.9). While the conventional squirrel-cage asynchronous machine can function in this mode, slip power is dissipated as Joule losses in the rotor.

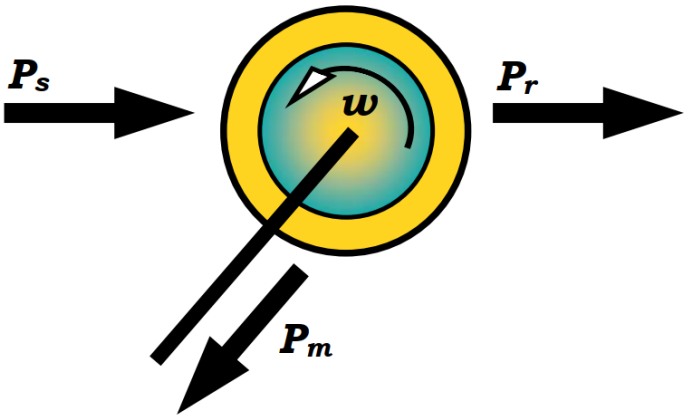

Figure 1.9: Subsynchronous Motor Operation of the DFIG

1.4.2 Supersynchronous Motor Operation

In Supersynchronous motor operation, Power is supplied by the grid to the stator, and the grid also supplies slip power to the rotor. Thus, the operation occurs above the synchronous speed, as illustrated in Figure (1.10).

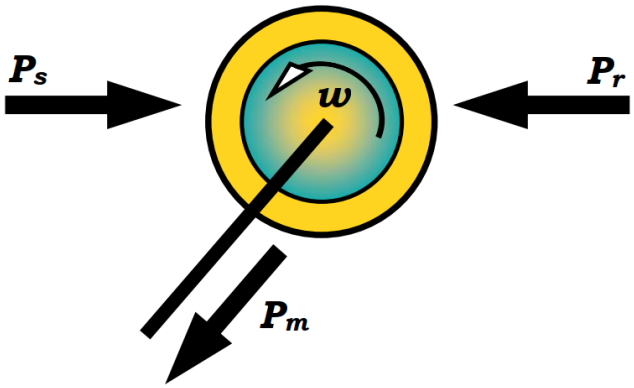

Figure 1.10: Supersynchronous Motor Operation of the DFIG

1.4.3 Subsynchronous Generator Operation

In subsynchronous generator operation, Power is supplied to the grid by the stator. The stator also supplies slip power. The rotor absorbs the slip power, and the direction of the magnetic field is the same as that of the stator field. Thus, generator operation occurs below the synchronous speed, as depicted in Figure (1.11).

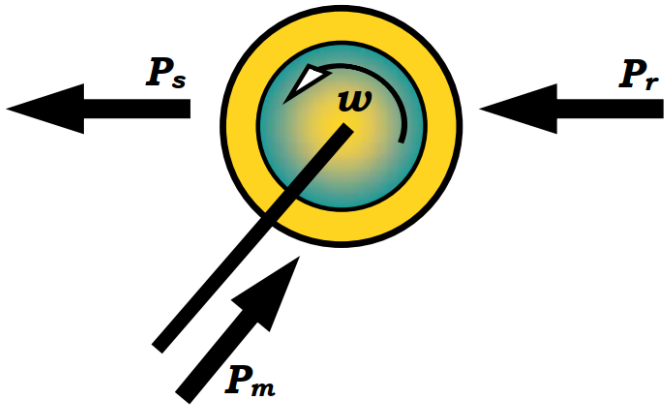

Figure 1.11: Subsynchronous Generator Operation of the DFIG

1.4.4 Supersynchronous Generator Operation

In supersynchronous generator operation, Power is supplied to the grid by the stator, and slip power is recovered via the rotor and fed back into the grid. Thus, generator operation occurs above the synchronous speed, as depicted in Figure (1.12).

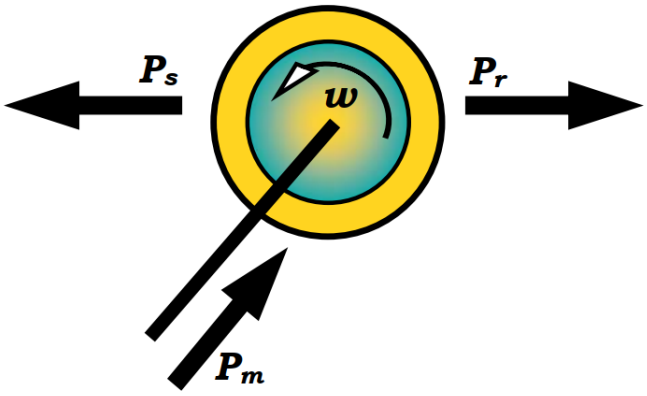

Figure 1.12: Supersynchronous Generator Operation of the DFIG

So, to recapitulate:

- For **Motor** Operation ($P_m > 0$), the machine is delivering mechanical Power and receiving electric Power from the grid, as shown in the table below:

Table 1.2: Motor Operational Modes of DFIG

Mode	P_m	P_s	P_R
Subsynchronous: $s > 0$ and $\omega_m < \omega_s$	$P_m > 0$	$P_s > 0$	$P_R > 0$
Supersynchronous: $s < 0$ and $\omega_m > \omega_s$	$P_m > 0$	$P_s > 0$	$P_R < 0$

- For **Generator** Operation ($P_m < 0$), the machine is receiving mechanical Power and delivering electric Power to the grid, as shown in the table below:

Table 1.3: Generator Operational Modes of DFIG

Mode	P_m	P_s	P_R
Subsynchronous: $s > 0$ and $\omega_m < \omega_s$	$P_m < 0$	$P_s < 0$	$P_R > 0$
Supersynchronous: $s < 0$ and $\omega_m > \omega_s$	$P_m < 0$	$P_s < 0$	$P_R < 0$

1.5 Working principals

An induction generator consists of a stator and a rotor. In a Doubly Fed Induction Generator (DFIG), both the stator and rotor have three sinusoidally distributed windings. These windings correspond to three phases, labelled a, b, and c, and are separated by an angle of 120°. The stator includes p pairs of poles.

The rotor is connected to the grid through converters. A three-winding transformer gives different voltage levels for stator and rotor side. A schematic of such a system is presented in Figure (1.13). When the machine produces energy, only a small part of the generated power flows from the rotor to the grid. The converters can then be chosen in accordance with this small rotor power. This means smaller converters compared to fully rated converters, and this decreases costs. [7]

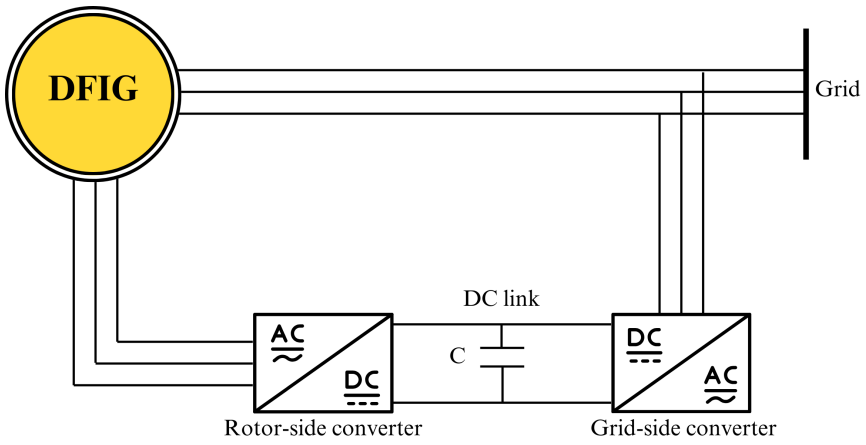

Figure 1.13: DFIG associated with its converters.

The stator windings are connected to the power grid, which sets the frequency of the stator current f_s. This results in the stator currents creating a rotating magnetic field within the air gap. The speed of this rotating field, ω_s, is directly proportional to f_s:

$$\omega_s = 2\pi f_s \tag{1.2}$$

When the rotor rotates at a speed different than the synchronous speed ω_s, a change in magnetic flux occurs. According to Faraday's law of induction, this change generates currents in the rotor windings. The electrical speed of the rotor, ω_r, is related to the mechanical speed of the rotor ω_m by the number of pole pairs P:

$$\omega_r = P \cdot \omega_m \tag{1.3}$$

The flux linked by the rotor windings changes with time if $\omega_r \neq \omega_s$. The machine operates usually as a generator if $\omega_r > \omega_s$ and as a motor otherwise. In the case of the DFIG, however, it can operate in sub-synchronous mode as a generator [7] , The slip, s, represents the speed difference between the rotor and the synchronous speed ω_s:

$$s = \frac{\omega_s - \omega_r}{\omega_s} \tag{1.4}$$

Equation (1.4) defines slip s in terms of the rotor's electrical speed ω_r and the synchronous speed ω_s.

For a generator, the slip is usually negative, while it is positive for a motor. The currents induced in the rotor windings oscillate at an angular velocity given by the difference between the synchronous speed and the rotor speed. The rotor current frequency, f_r, is:

$$f_r = s \cdot f_s \tag{1.5}$$

If the rotor rotated at synchronous speed ω_s, there would be no change in magnetic flux, and hence, no electrical currents would be induced in the windings. Therefore, the machine always operates at a speed different from the synchronous speed.

The rotor-side inverter regulates the rotor currents. As shown in Equation (1.5), by controlling the rotor currents, the slip s and thus the speed of the machine can be managed [7].

1.6 Applications of the DFIM

The Doubly Fed Induction Machine (DFIM) occupies a significant place in industrial applications due to its numerous advantages. Indeed, the DFIM is widely used as a generator in renewable energy applications, particularly in wind power systems [8]. Additionally, its operation as a generator makes the DFIM a serious alternative to conventional synchronous machines in many decentralized power generation systems, such as

- On-board power generators for ships or aircraft;
- Hydraulic power plants with variable flow and speed;
- Generator sets where reducing speed during low consumption significantly reduces fuel consumption.

The DFIM can also be used in other significant applications requiring high starting torque, such as [6]:

- Metallurgy with coil winders and unwinders;
- Traction, including urban transport or maritime propulsion applications;
- Lifting applications such as elevators and hoists.

It should be noted that the DFIM's applications as a motor are relatively limited. The primary applications are in electric traction and pumping systems.

1.6.1 Advantages and Disadvantages of the DFIG

In this section, we briefly introduce the main advantages and disadvantages of the Doubly Fed Induction Generator (DFIG) when used in the field of variable speed applications [9]

1.6.1.1 Advantages

Among its numerous advantages, we can mention:

- The ability to modify the characteristics of the rotor winding of the machine, notably by connecting rheostats to limit the current, increase torque during startup, and extend the speed variation range.
- The accessibility to the stator and rotor offers the opportunity to have great flexibility and precision in controlling the flux and the electromagnetic torque, and consequently for well controlling the transfer of active and reactive Power.
- The possibility of operating the DFIG as an active filter for current harmonics thanks to the rotor currents' indirect control of active and reactive powers.
- The converter on the rotor side is sized for a third of the DFIG's nominal Power, which effectively divides their price by three compared to converters sized for nominal power, thus improving the efficiency of the conversion system.

- In generator mode, the powering of the rotor circuit with variable frequency allows delivering a fixed frequency to the stator even in case of speed variation.
- The ability to operate at constant torque beyond the nominal speed.
- Operation in a degraded mode, if one of the two inverters fails, is more flexible than that of a single-feed machine.
- The DFIG has a slightly higher power density than other large power machines, making it competitive for high powers ($>$ 1MW).

1.6.1.2 Disadvantages

Compared to other machines, the DFIG presents disadvantages mainly related to the slip-ring system:

- It is more significant than a squirrel-cage Induction Motor (IM) of equivalent power due to the presence of the slip-ring system.
- The additional cost incurred by maintenance due to the use of the gearbox and the slip-ring system of the DFIG, compared to other machines such as the permanent magnet synchronous machine and the switched reluctance machine.
- The multi-converter aspect increases the number of converters (rectifiers and two inverters or one rectifier and two inverters) and consequently the price, although some studies claim the opposite.

1.7 Modeling of Doubly Fed Induction Generators

As introduced earlier, the doubly fed induction machine (DFIM) is an induction wound rotor machine with a construction very similar to that of the traditional squirrel-cage machine. It consists of two sets of three-phase windings: a stator and a rotor.

1.7.1 Simplifying Assumptions for DFIG Modeling

The DFIG system is characterized by its high order, nonlinearity, and time-varying nature. To simplify the process of dynamic analysis, we introduce the following assumption: [10]

- Three-phase generator windings are balanced with a Y connection, with 120° phase shift between each other in space. The induced magneto-motive force is distributed in sinusoidal form along the air gap.

- The magnetic saturation of the stator and the rotor core is neglected.
- The iron loss of both the stator and the rotor core is neglected.
- The stator or rotor winding parasitic resistances do not change with temperature or frequency.

To simplify the analysis, all electrical variables on the rotor side are converted to the stator side [10], and the positive reference direction is chosen to be the direction in which Power or current enters the stator or rotor of the DFIG.

1.7.2 ABC (abc) Model

Based on the previously stated assumptions, the DFIG can be modeled as three static windings positioned on the stator (ABC) and three windings positioned on the rotor (abc) rotating at an angular frequency of (ω_r), as illustrated in Figure (1.14).

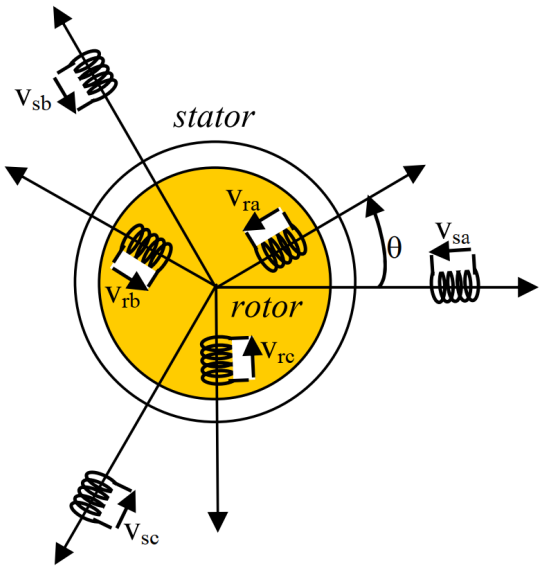

Figure 1.14: DFIG stator and rotor windings.

In Figure (1.14), v_{sa}, v_{sb}, v_{sc} and v_{ra}, v_{rb}, v_{rc} are three-phase stator voltages and three-phase rotor voltages, respectively.

The following equations describe a three-phase symmetrical doubly fed induction generator.

1.7.2.1 Electrical relations

The voltage relations on rotor and stator sides are:

$$\begin{cases} [V_s] = [R_s] \cdot [I_s] + \dfrac{d}{dt}[\phi_s] \\ [V_r] = [R_r] \cdot [I_r] + \dfrac{d}{dt}[\phi_r] \end{cases} \tag{1.6}$$

With :

$$[V_s] = \begin{bmatrix} V_{as} \\ V_{bs} \\ V_{cs} \end{bmatrix} \quad [I_s] = \begin{bmatrix} I_{as} \\ I_{bs} \\ I_{cs} \end{bmatrix} \quad [\phi_s] = \begin{bmatrix} \phi_{as} \\ \phi_{bs} \\ \phi_{cs} \end{bmatrix} \quad [R_s] = \begin{bmatrix} R_s & 0 & 0 \\ 0 & R_s & 0 \\ 0 & 0 & R_s \end{bmatrix}$$

$$[V_r] = \begin{bmatrix} V_{ar} \\ V_{br} \\ V_{cr} \end{bmatrix} \quad [I_r] = \begin{bmatrix} I_{ar} \\ I_{br} \\ I_{cr} \end{bmatrix} \quad [\phi_r] = \begin{bmatrix} \phi_{ar} \\ \phi_{br} \\ \phi_{cr} \end{bmatrix} \quad [R_r] = \begin{bmatrix} R_r & 0 & 0 \\ 0 & R_r & 0 \\ 0 & 0 & R_r \end{bmatrix}$$

The subscripts r and s denote rotor and stator quantities, respectively.

The subscripts a,b and c are used for phases a, b and c quantities,respectively.

The symbols v and i are for voltages and currents and ϕ represents flux linkages.

The stator and rotor winding resistances are Rs and Rr. (They are assumed to be equal for all phase windings).

1.7.2.2 Magnetic relations

The flux linkages are coupled to the currents by the inductances:

$$\begin{cases} [\phi_s] = [L_{ss}] \cdot [I_s] + [M_{sr}] \cdot [I_r] \\ [\phi_r] = [L_{rr}] \cdot [I_r] + [M_{rs}] \cdot [I_s] \end{cases} \tag{1.7}$$

With:

$$[L_{ss}] = \begin{bmatrix} l_s & M_s & M_s \\ M_s & l_s & M_s \\ M_s & M_s & l_s \end{bmatrix} \quad ; \quad [L_{rr}] = \begin{bmatrix} l_r & M_r & M_r \\ M_r & l_r & M_r \\ M_r & M_r & l_r \end{bmatrix}$$

Where L_{ss}, L_{rr} are respectively the self-inductances of the stator and rotor phases.
M_s, M_r - the mutual inductances between the stator and rotor phases.
$[M_{sr}]$ is the transpose of $[M_{rs}]$ - the matrix of mutual inductances or stator-rotor coupling matrix, which is given by the formula (1.8):

$$[M] = [M_{sr}] = [M_{rs}]^T = M_{\max} \begin{bmatrix} \cos(\theta) & \cos\left(\theta + \frac{2\pi}{3}\right) & \cos\left(\theta - \frac{2\pi}{3}\right) \\ \cos\left(\theta - \frac{2\pi}{3}\right) & \cos(\theta) & \cos\left(\theta + \frac{2\pi}{3}\right) \\ \cos\left(\theta + \frac{2\pi}{3}\right) & \cos\left(\theta - \frac{2\pi}{3}\right) & \cos(\theta) \end{bmatrix} \tag{1.8}$$

Where $M_{\max}$ is the maximum mutual inductance between a stator phase and a rotor phase.

By replacing (1.7) in (1.6) , we get the following :

$$\begin{cases} [V_s] = [R_s] \cdot [I_s] + \dfrac{d}{dt}\left([L_{ss}] \cdot [I_s] + [M_{sr}] \cdot [I_r]\right) \\ [V_r] = [R_r] \cdot [I_r] + \dfrac{d}{dt}\left([L_{rr}] \cdot [I_r] + [M_{rs}] \cdot [I_s]\right) \end{cases} \tag{1.9}$$

It is noticed that both the stator flux and the rotor flux change with rotor angle θ_r. Therefore, DFIG model in ABC (*abc*) frame is a time-dependent system. Thus, analyzing its dynamics in *abc*-reference frame is more complicated. We will employ the *dq0*-transformation to eliminate the time dependency.

1.7.2.3 Mechanical relations

In the above section, the electrical dynamics of the DFIG have been developed in the stator reference frame. A model of the mechanical dynamics is given here to complete the model. The dynamics of the generator shaft relate to the rotor speed and the electromagnetic torque: [11]

$$T_e = T_r + f_r \cdot \Omega + J \cdot \frac{d\Omega}{d\theta} \tag{1.10}$$

With:

- T_e - the electromagnetic torque of the machine;
- T_r - the resisting torque;
- f_r - the coefficient of viscous friction of the DFIG;
- Ω - the rotation speed of the axis of the DFIG;
- J - the inertia of the rotating parts.

The following relation gives the expression of the electromagnetic torque:

$$T_e = P \cdot [I_s]^T \cdot \frac{d}{d\theta}[M] \cdot [I_r] \tag{1.11}$$

Where P is the number of pole pairs of the DFIG.

1.7.3 dq-reference frame

Switching to a more suitable reference frame is convenient for generating a simpler model. One such reference frame is the so-called *dq*-reference frame.

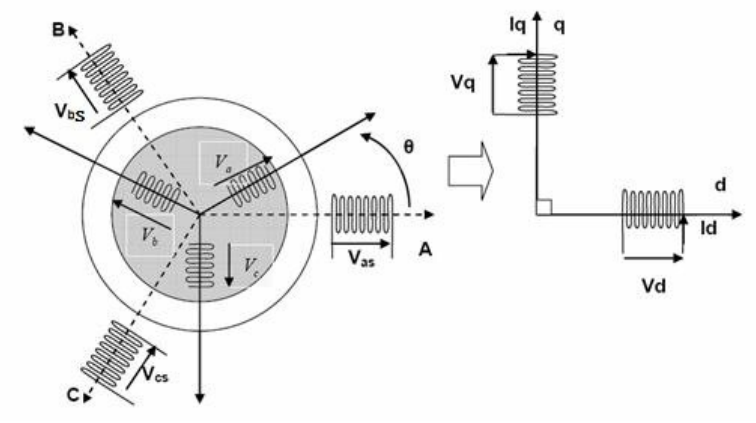

Figure 1.15: abc to dq transformation

The original ABC variables must be transformed into d, q variables (Figure 1.15), but this transformation depends on the speed of rotation of the D, Q coils. Hence, each reference frame has its own transformation. [12]
In general, for any arbitrary value of θ, the transformation of stator ABC phase variables $[F_{ABC}]$ to d,q stator variables $[F_{dq}]$ is carried out through Park's transform as follows:

$$[F_{dq}] = [P_\theta][F_{ABC}] \tag{1.12}$$

where $P(\theta)$ is the rotation matrix which transforms the abc-quantities into the dq-reference frame:

$$P(\theta) = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos(\theta) & \cos\left(\theta - \frac{2\pi}{3}\right) & \cos\left(\theta + \frac{2\pi}{3}\right) \\ -\sin(\theta) & -\sin\left(\theta - \frac{2\pi}{3}\right) & -\sin\left(\theta + \frac{2\pi}{3}\right) \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}, \tag{1.13}$$

with

- $\theta = \theta_s$ when the stator quantities are of interest,
- $\theta = \theta_r$ when the rotor quantities are interesting.

The notation $[F_{ABC}]$ represents a vector quantity, which may be a current, voltage, or flux.

1.7.4 dq Model

The DFIG (Doubly Fed Induction Generator) modeling is fundamentally the same as that of the traditional asynchronous machine. However, the distinction lies in that the rotor windings are not short-circuited, which means that the rotor voltages are not zero. The mathematical model is created by applying Park's transformation, which involves transforming the three-phase winding system of ABC axes into an equivalent two-phase winding system of dq axes, making the same magneto-motive force rotate at the same speed as the stator field. (Figure 1.16)

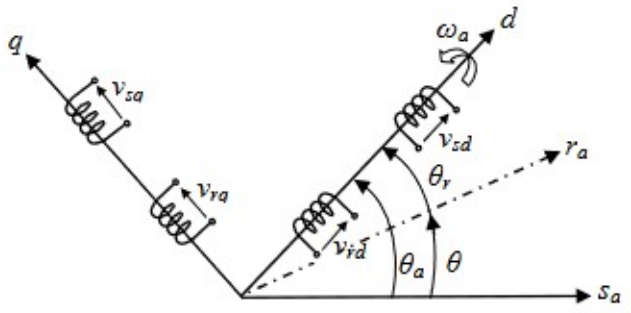

Figure 1.16: DFIG model after PARK transform

As introduced earlier, the Park equation to pass from ABC reference to dq reference and vice versa is :

$$[F_{dq}] = [P_\theta][F_{ABC}] \hookrightarrow [F_{ABC}] = [P_\theta]^{-1}[F_{dq}]$$

Park's transformation, applied to the stator equations, gives us:

$$\begin{aligned}
[V_{s_{dq0}}] = [P(\theta_s)] \cdot [V_{s_{abc}}] \quad &\Rightarrow \quad [V_{s_{abc}}] = [P(\theta_s)]^{-1} \cdot [V_{s_{dq0}}] \\
[I_{s_{dq0}}] = [P(\theta_s)] \cdot [I_{s_{abc}}] \quad &\Rightarrow \quad [I_{s_{abc}}] = [P(\theta_s)]^{-1} \cdot [I_{s_{dq0}}] \\
[\phi_{s_{dq0}}] = [P(\theta_s)] \cdot [\phi_{s_{abc}}] \quad &\Rightarrow \quad [\phi_{s_{abc}}] = [P(\theta_s)]^{-1} \cdot [\phi_{s_{dq0}}]
\end{aligned}$$

Park's transformation, applied to the rotor equations, gives us:

$$\begin{aligned}
[V_{r_{dq0}}] = [P(\theta_r)] \cdot [V_{r_{abc}}] \quad &\Rightarrow \quad [V_{r_{abc}}] = [P(\theta_r)]^{-1} \cdot [V_{r_{dq0}}] \\
[I_{r_{dq0}}] = [P(\theta_r)] \cdot [I_{r_{abc}}] \quad &\Rightarrow \quad [I_{r_{abc}}] = [P(\theta_r)]^{-1} \cdot [I_{r_{dq0}}] \\
[\phi_{r_{dq0}}] = [P(\theta_r)] \cdot [\phi_{r_{abc}}] \quad &\Rightarrow \quad [\phi_{r_{abc}}] = [P(\theta_r)]^{-1} \cdot [\phi_{r_{dq0}}]
\end{aligned}$$

1.7.4.1 Electrical relations

By applying Park transform to the equation (1.6) , we get :

$$\begin{aligned}
[P(\theta_s)]^{-1} \cdot [V_{s_{dq0}}] &= [R_s] \cdot [P(\theta_s)]^{-1} \cdot [I_{s_{dq0}}] + \frac{d}{dt}[P(\theta_s)]^{-1} \cdot [\phi_{s_{dq0}}] \\
[P(\theta_r)]^{-1} \cdot [V_{r_{dq0}}] &= [R_r] \cdot [P(\theta_r)]^{-1} \cdot [I_{r_{dq0}}] + \frac{d}{dt}[P(\theta_r)]^{-1} \cdot [\phi_{r_{dq0}}]
\end{aligned} \tag{1.14}$$

$$\begin{aligned}
[V_{s_{dq0}}] &= [R_s] \cdot [I_{s_{dq0}}] + [P(\theta_s)] \cdot \frac{d}{dt}[P(\theta_s)]^{-1} \cdot [\phi_{s_{dq0}}] \\
[V_{r_{dq0}}] &= [R_r] \cdot [I_{r_{dq0}}] + [P(\theta_r)] \cdot \frac{d}{dt}[P(\theta_r)]^{-1} \cdot [\phi_{r_{dq0}}]
\end{aligned} \tag{1.15}$$

With:

$$\frac{d}{dt}[P(\theta_s)] = \frac{d\theta_s}{dt}\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad ; \quad \frac{d}{dt}[P(\theta_r)] = \frac{d\theta_r}{dt}\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{1.16}$$

By consequence, the voltage equations of the stator and rotor of the DFIG are given by: [11]

$$\begin{cases} V_{sd} = R_s I_{sd} + \dfrac{d\phi_{sd}}{dt} - \phi_{sq}\dfrac{d\theta_s}{dt} \\ V_{sq} = R_s I_{sq} + \dfrac{d\phi_{sq}}{dt} + \phi_{sd}\dfrac{d\theta_s}{dt} \\ V_{rd} = R_r I_{rd} + \dfrac{d\phi_{rd}}{dt} - \phi_{rq}\dfrac{d\theta_r}{dt} \\ V_{rq} = R_r I_{rq} + \dfrac{d\phi_{rq}}{dt} + \phi_{rd}\dfrac{d\theta_r}{dt} \end{cases} \tag{1.17}$$

V_{sd}, V_{sq}, V_{rd} and V_{rq} are respectively the direct and quadrature stator and rotor voltages;

I_{sd}, I_{sq}, I_{rd} and I_{rq} are respectively the direct and quadrature stator and rotor currents;

$\phi_{sd}, \phi_{sq}, \phi_{rd}$ and ϕ_{rq} are, respectively, the direct and quadrature stator and rotor fluxes of the system;

θ_s, θ_r are the electrical angles that form the axes of the stator and the rotor with the direct axis of Park's two-phase reference frame.

1.7.4.2 Magnetic relations

By applying the same approach, we get :

$$\begin{aligned} [P(\theta_s)]^{-1} \cdot [\phi_{s_{dq0}}] &= [L_{ss}] \cdot [P(\theta_s)]^{-1} \cdot [I_{s_{dq0}}] + [M_{sr}] \cdot [P(\theta_s)]^{-1} \cdot [I_{r_{dq0}}] \\ [P(\theta_r)]^{-1} \cdot [\phi_{r_{dq0}}] &= [L_{rr}] \cdot [P(\theta_r)]^{-1} \cdot [I_{r_{dq0}}] + [M_{rs}] \cdot [P(\theta_r)]^{-1} \cdot [I_{s_{dq0}}] \end{aligned} \tag{1.18}$$

As a consequence, the flux equations of the stator and rotor of DFIG are given by:

$$\begin{cases} \phi_{sd} = L_s I_{sd} + M I_{rd} \\ \phi_{sq} = L_s I_{sq} + M I_{rq} \\ \phi_{rd} = L_r I_{rd} + M I_{sd} \\ \phi_{rq} = L_r I_{rq} + M I_{sq} \end{cases} \tag{1.19}$$

L_s, L_r are, respectively, the cyclic stator and rotor inductances;

M is the mutual inductance.

1.7.4.3 Electromagnetic torque relation

The electromagnetic torque can be obtained, from several equal expressions, from the stator flux and currents by:

$$T_e = P \cdot (\phi_{sd} i_{sq} - \phi_{sq} i_{sd}) \tag{1.20}$$

It can also be expressed in terms of the rotor currents and stator fluxes by:

$$T_e = P \cdot \frac{M}{L_s} (I_{rd}\phi_{sq} - I_{rq}\phi_{sd}) = P \cdot \frac{M}{L_s}(\phi_{qs} i_{dr} - \phi_{ds} i_{qr}) \tag{1.21}$$

This equation is used in field-oriented (vector) control of active and reactive stator powers with the orientation of the stator flux, which will be studied in the following chapter.

The active and reactive powers of the DFIG of the stator and the rotor, respectively, are defined by: [13]

$$\begin{cases} P_s = V_{sd}I_{sd} + V_{sq}I_{sq} \\ Q_s = V_{sq}I_{sd} - V_{ds}I_{sq} \\ P_r = V_{rd}I_{rd} + V_{rq}I_{rq} \\ Q_r = V_{rq}I_{rd} - V_{rd}I_{rq} \end{cases} \tag{1.22}$$

1.7.4.4 Choice of reference

The analysis of the DFIG based on Park's transformation leads to relationships related to the chosen frame of reference. (Figure 1.17)

Depending on the objective of the application considered, the following three possibilities appear:

- ***The (d-q) axis reference frame fixed to the stator :*** This reference frame is used to analyze transient states with large speed variations or study rotor quantities regardless of the supply frequency. $(\theta_s = 0, \theta_r = \theta)$
- ***The (d-q) axis reference frame fixed to the rotor:*** This reference frame is employed to study transient states, especially when studying rotor faults such as rotor bar breakages or imbalances of rotor windings. $(\theta_r = 0, \theta_s = \theta)$
- ***The (d-q) axis reference frame fixed to the rotating field:*** This reference frame is primarily used for machine control. In this frame, the model quantities are constant in steady state because the frequency of the rotating field equals that of the stator quantities; hence:

  $$\frac{d\theta_s}{dt} = \omega_s, \quad \frac{d\theta}{dt} = \omega_s - \omega_r \tag{1.23}$$

  where ω_s is the synchronous angular speed of the stator's rotating magnetic field.

  The rotor angular speed (rotational speed) is expressed by:

  $$\omega_r = \omega_s - \omega'_r \Rightarrow \theta_r = \theta_s - \theta \tag{1.24}$$

  where ω'_r is the angular speed in the (d-q) reference frame fixed to the rotor.

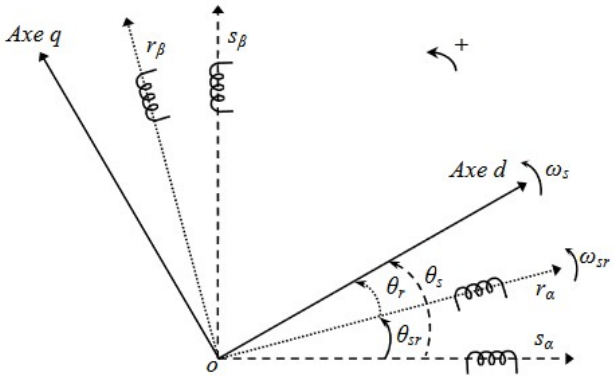

Figure 1.17: Illustration of Park's Reference Frames: Stator ($s_\alpha - s_\beta$), Rotor ($r_\alpha - r_\beta$), and Rotating ($d - q$) frames.

The reference frame fixed to the rotating field will be considered in the following work, as it is ideally suited to synthesizing the various control strategies considered in this study.

1.8 Flux-Oriented Control of the DFIG

1.8.1 Model of the DFIG with Stator flux-oriented control

To achieve decoupled stator reactive power and rotor speed control, we choose in the *dq0*-reference-frame the stator flux phi_s aligned with the d-axis, so we have: [14]

$$\begin{cases} \phi_{sd} = \phi_s \\ \phi_{sq} = 0 \end{cases} \tag{1.25}$$

The equation system can be simplified to the following form:

$$\begin{cases} V_{sd} = R_s i_{sd} \\ V_{sq} = R_s i_{sq} + \omega_s \phi_s \\ V_{rd} = R_r i_{rd} + \dfrac{d\phi_{rd}}{dt} - \omega_r \phi_{rq} \\ V_{rq} = R_r i_{rq} + \dfrac{d\phi_{rq}}{dt} + \omega_r \phi_{rd} \end{cases} \tag{1.26}$$

If we neglect the stator winding resistance, an assumption often accepted for large power machines used in wind energy generation, the voltage equations of the machine reduce to the following form:

$$\begin{cases} V_{sd} = 0 \\ V_{sq} = V_s = \omega_s \phi_s \\ V_{rd} = R_r i_{rd} + \dfrac{d\phi_{rd}}{dt} - \omega_r \phi_{rq} \\ V_{rq} = R_r i_{rq} + \dfrac{d\phi_{rq}}{dt} + \omega_r \phi_{rd} \end{cases} \tag{1.27}$$

The flux equations become:

$$\begin{cases} \phi_{sd} = \phi_s = L_s i_{sd} + M_s i_{rd} \\ 0 = L_s i_{sq} + M_s i_{rq} \\ \phi_{rd} = L_r i_{rd} + M_s i_{sd} \\ \phi_{rq} = L_r i_{rq} + M_s i_{sq} \end{cases} \tag{1.28}$$

The expression for the electromagnetic torque then becomes:

$$T_e = -P \frac{M}{L_s} \phi_s i_{rq} \tag{1.29}$$

Assuming that the electrical grid is stable leads to a constant stator flux ϕ_s. This shows that the electromagnetic torque T_e is directly proportional to the rotor current i_{rq} in quadrature.

The active and reactive stator powers in the orthogonal reference frame are given by:

$$\begin{cases} P = V_s i_{sq} \\ Q = V_s i_{sd} \end{cases} \tag{1.30}$$

Therefore, the active Power P and reactive Power Q are functions of the stator currents i_{sd} and i_{sq} respectively, over which we have direct control.

From the expressions of the stator fluxes, we can write:

$$\begin{cases} i_{sd} = \dfrac{V_s}{L_s \omega_s} - \dfrac{M}{L_s} i_{rd} \\ i_{sq} = -\dfrac{M}{L_s} i_{rq} \end{cases} \tag{1.31}$$

By substituting the direct and quadrature stator currents with their expressions from the flux equations into the equations for active and reactive Power, we obtain:

$$\begin{cases} P = -\dfrac{V_s M_s}{L_s} i_{rq} \\ Q = -\dfrac{V_s M_s}{L_s} i_{rd} + \dfrac{V_s^2}{L_s \omega_s} \end{cases} \tag{1.32}$$

The control of active and reactive stator power is decoupled. The active Power is directly proportional to the quadrature rotor current, and the reactive Power is proportional to the direct rotor current to a preset constant ($\frac{V_s{}^2}{L_s\omega_s}$) imposed by the grid.

1.8.2 Expressions of Rotor Voltages

To control the machine, it is necessary to establish the correlation between the currents and rotor voltages applied to it [15]. By substituting the stator equation (1.31) into the flux equations to obtain the rotor fluxes, then the result into the rotor voltage equation, we find :

$$\begin{cases} V_{rd} = R_r i_{rd} + \left(L_r - \dfrac{M^2}{L_s}\right)\dfrac{di_{rd}}{dt} - s\omega_s\left(L_r - \dfrac{M^2}{L_s}\right) i_{rq} \\ V_{rq} = R_r i_{rq} + \left(L_r - \dfrac{M^2}{L_s}\right)\dfrac{di_{rq}}{dt} + s\omega_s\left(L_r - \dfrac{M^2}{L_s}\right) i_{rd} + s\dfrac{MV_s}{L_s} \end{cases} \tag{1.33}$$

In steady state, the terms involving the derivatives of the phase-shifted rotor currents disappear. Thus, we can write:

$$\begin{cases} V_{rd} = R_r i_{rd} - s\omega_s\left(L_r - \dfrac{M^2}{L_s}\right) i_{rq} \\ V_{rq} = R_r i_{rq} + s\omega_s\left(L_r - \dfrac{M^2}{L_s}\right) i_{rd} + s\dfrac{MV_s}{L_s} \end{cases} \tag{1.34}$$

The rotor voltages are linked to the stator's active and reactive powers. This also means that we can implement vector control since, close coupling effects aside, each axis can be independently controlled, each with its own controller. (Figure 1.18)

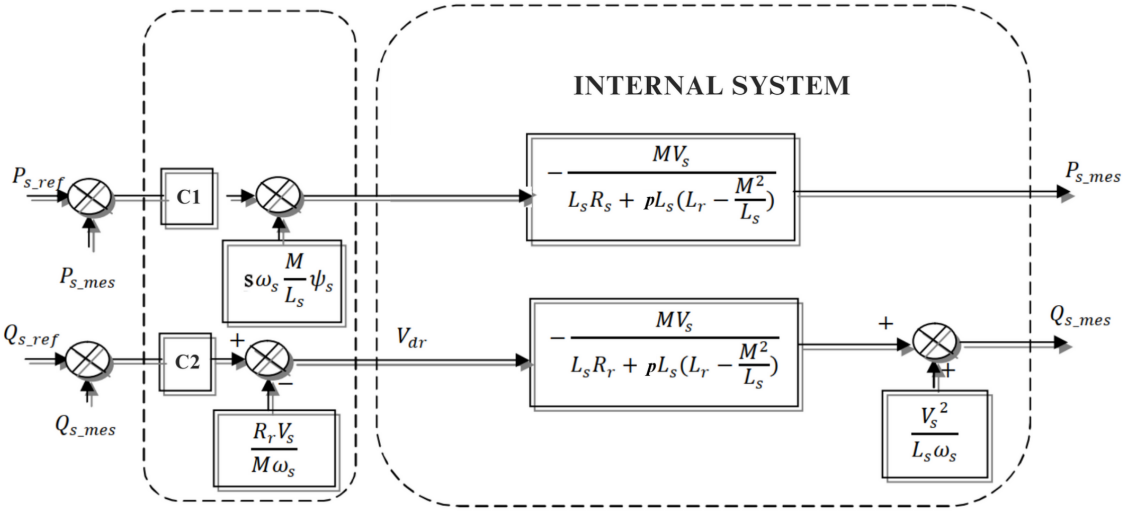

Figure 1.18: Direct vector control of DFIG [16]

The blocks $C1$ and $C2$ in Figure (1.19) represent the active and reactive power controllers. Their nature will be discussed in the upcoming chapters.

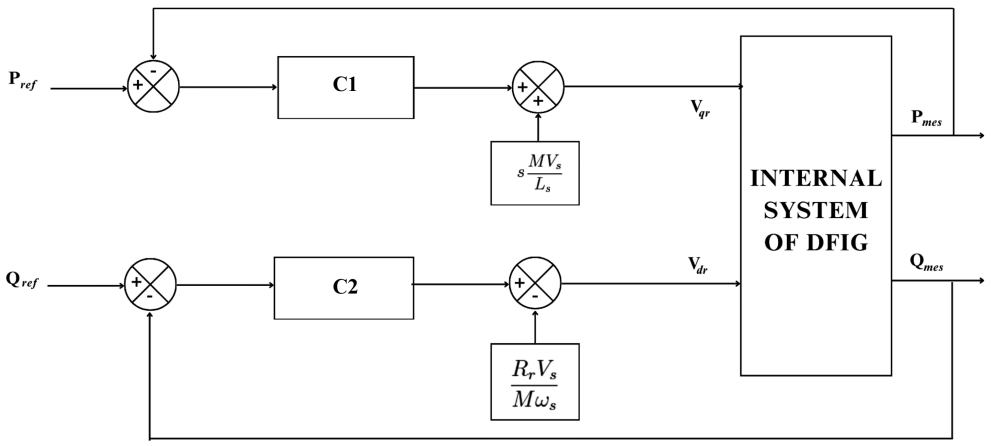

Figure 1.19: Direct vector control of DFIG

This method is referred to as the **direct** method because the power controllers directly control the rotor voltages of the DFIG.

1.9 Conclusion

In this chapter, we've covered the basics of the Doubly Fed Induction Generator (DFIG). We explained how it works and described its mathematical model. This model hinges on the electrical and mechanical equations derived from Park's transformation within the rotating reference frame linked to the rotating field. Additionally, we provided an overview of field-oriented control, which will be utilized throughout the rest of this thesis.

In the forthcoming chapter, we will explore intelligent neural control techniques to regulate active and reactive stator powers, improving the DFIG's performance in real-world applications regarding time response, steady-state error, and overall system stability.

Chapter 2

Artificial Neural Networks (ANN) Control of DFIG Powers.

2.1 Introduction

Artificial neural networks (ANNs) represent a powerful computational technique that emphasizes numerical learning over symbolic learning and arithmetic operations over logic-based rules. Traditionally, ANNs have been employed in various domains, such as pattern recognition, speech recognition, and optimization. However, their exceptional learning capabilities make them highly suitable for process regulation and control.

This chapter focuses on the application of neural control to regulate the stator powers of the Doubly-Fed Induction Generator (DFIG). The objective is to fine-tune the active power to align with the turbine's reference output, thereby maximizing the efficiency of the wind energy system while dynamically adjusting the reactive power to maintain optimal performance and achieve a desirable power factor on the stator side.

After introducing the neural control approach and detailing the structure and properties of neural networks, we present simulation results and robustness tests to evaluate the effectiveness of the neural control method. The results demonstrate the potential of ANNs in enhancing the performance and reliability of DFIG-based wind energy systems.

2.2 Principle of Artificial Neural Networks

Artificial neural networks (ANNs) constitute a family of functions that enable the construction of various models and controllers through learning. Inspired by the behavior of a biological neural network, ANNs are composed of systems of interconnected nonlinear operators organized in layers. (Figure 2.1) Each neuron receives one or more inputs, performs a weighted sum of the inputs, adds a bias, and then applies an activation function to produce an output.

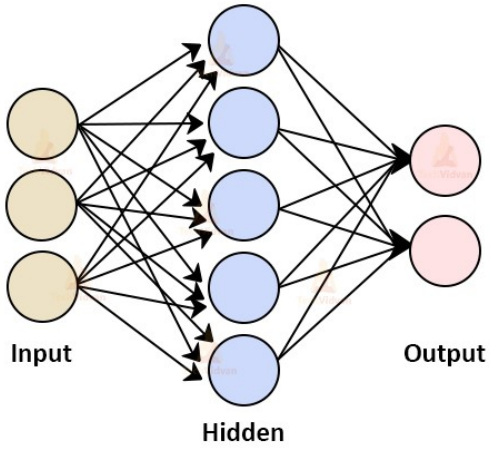

Figure 2.1: Neural Networks structure.

This method is highly valued in fields where it is necessary to model complex, robust

functions, requiring high adaptability and management of unstructured data.
Several application domains are feasible for ANNs, for example:

- Pattern recognition.
- Computer vision.
- Machine translation.
- Time series prediction.
- Dynamic system modeling.
- Image classification.
- Machine learning.

2.2.1 Brief History [17]

- **1943**: McCulloch and Pitts present the first formal neuron.
- **1949**: Hebb proposes a learning mechanism.
- **1958**: Rosenblatt presents the first artificial neural network: the Perceptron. The visual system inspires it and has two layers of neurons: perceptive and decision-making. Widrow presented the ADALINE (ADAptive LINear Element) model during the same period. This will be the basic model for multilayer networks.
- **1969**: Minsky and Papert publish a critique of perceptrons, showing their limitations, which decreases research on the subject.
- **1972**: Kohonen presents his work on associative memories.
- **1982**: Hopfield demonstrates the importance of using recurrent networks for understanding and modeling memory functions.
- **1986**: Werbos's backpropagation algorithm, which enables training the hidden layers of multilayer networks, becomes popular thanks to Rumelhart.

Neural networks have since been extensively studied, and numerous applications have been found.

2.2.2 Biological Neuron Topology

The human brain consists of two lateral hemispheres connected by the corpus callosum and other axonal bridges. It weighs less than two kilograms and contains a trillion cells, of which 100 billion are neurons organized into networks. Neurons are nerve cells that can be broken down into four main parts (Figure 2.2) :

- **Dendrites**: These are the branches where other cells make synaptic contact and signal reception occurs through the dendrites.
- **Cell Body (Soma)**: This is the processing unit of the neuron.
- **Axon**: Messages accumulated in the cell body travel along the axon; information transmission occurs via the axon.
- **Synapses**: These are the points of connection through which the cell communicates with other cells, allowing signals to pass from one cell to another.

When stimulated, a neuron sends electrical impulses, or action potentials, to other neurons. These impulses travel along the neuron's single axon and are converted into chemical signals at the contact points between neurons and synapses. When the accumulation of excitations reaches a certain threshold, the neuron delivers an action potential of approximately 100 millivolts and a duration of 1 millisecond. [18]

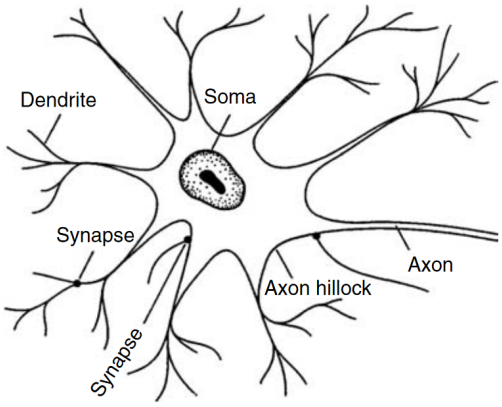

Figure 2.2: Neuron anatomy. [19]

2.2.3 Formal Neuron

A formal/artificial neuron is a very simple processor (simulated on a computer or implemented on an integrated circuit) that roughly mimics the structure and function of a biological neuron. The first version of the formal neuron, [20], developed by McCulloch and Pitts, appeared in 1943.

It is a binary automaton that performs a weighted sum of its inputs, known as the potential, and compares this potential to a threshold (zero): if it is higher, the output is +1, and the neuron is active; if it is lower, the output is -1, and the neuron is inactive. [20] These days, different kinds of neurons work better for tasks like signal processing or classification. Their output is not the sign of their potential but a nonlinear differentiable function f of this potential, like a hyperbolic tangent.
The function in question is referred to as the activation function of the neuron (Figure 2.3).

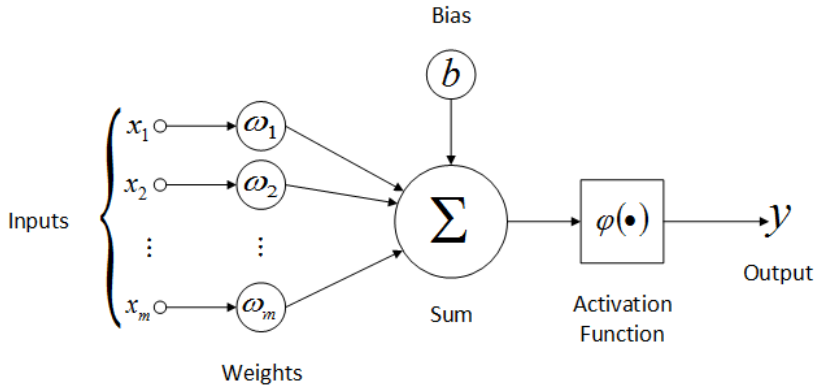

Figure 2.3: Sketch of an artificial neuron. [21]

- The neuron's inputs are $(x_1, x_2, \ldots, x_n)$.
- Along with the inputs, the neuron receives a bias; typically, its value is set to 1.
- The weights are represented by $(w_0, w_1, w_2, \ldots, w_n)$.
- The product of weight and input gives the signal strength. [22]
- A neuron receives multiple inputs from various sources and produces a single output. [22]

2.2.4 Activation Functions

Different transfer functions that can be used as neuron activation functions are listed in Table (2.1).

Table 2.1: Different transfer functions used as activation functions. [21]

Function	**Equation**	**Description**
Threshold (Hard Limit)	$f(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$	Outputs 0 or 1 based on the input threshold
Linear	$f(x) = x$	Outputs the input directly
Sigmoid	$f(x) = \frac{1}{1+e^{-x}}$	Outputs a value between 0 and 1

The most commonly used activation functions are the "threshold" (also known as "hard limit"), "linear," and "sigmoid" functions. (Figure 2.4)

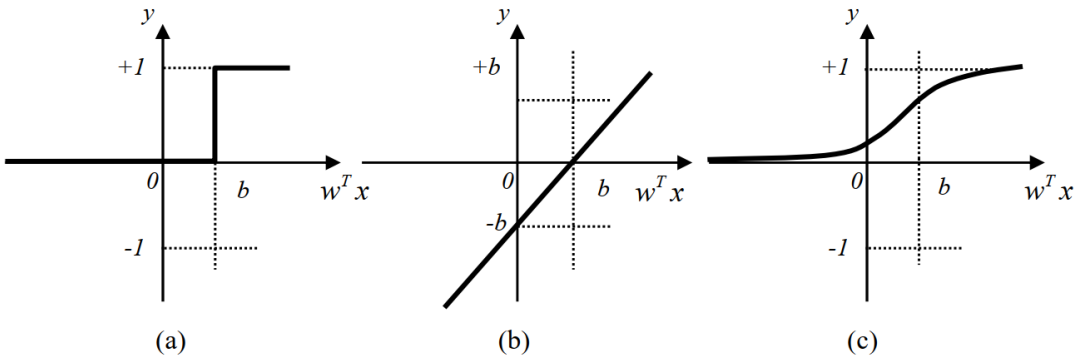

Figure 2.4: Activation functions: (a)-threshold; (b)-linear; (c)-sigmoid.

In our case, multilayer networks often use the log-sigmoid transfer function (*logsig*) illustrated in Figure (2.5). [23]

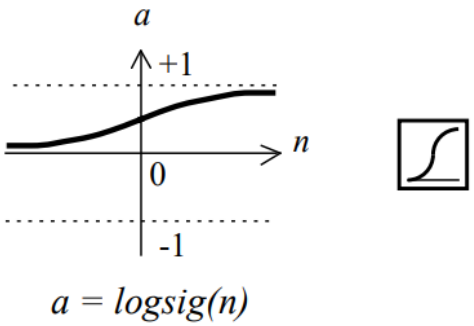

Figure 2.5: sigmoid activation function. [23]

2.3 Neural Network Architectures

The architecture of a neural network depends on the task to be learned. A neural network generally consists of multiple layers of neurons, from inputs to outputs. There are two main types of neural network architectures: feedforward and recurrent neural networks.

2.3.1 Feedforward Neural Networks

The feedforward neural network is one of the most basic artificial neural networks. In this ANN, the data or the input provided travels in a single direction. It enters the ANN through the input layer and exits through the output layer, while hidden layers may or may not exist. So, the feedforward neural network only has a front-propagated wave and usually does not have backpropagation. [24]

2.3.1.1 Single-Layer Neural Networks

The structure of a single-layer network is such that neurons organized in the input are fully connected to other neurons organized in the output by a modifiable layer of weights (Figure 2.6).

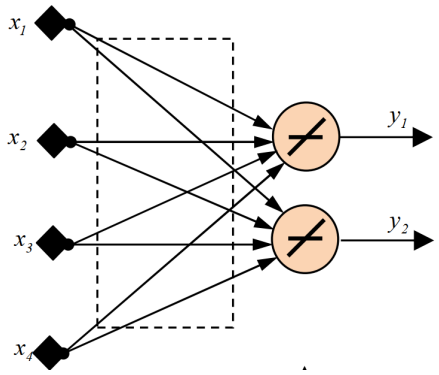

Figure 2.6: Single-Layer Neural Network.

2.3.1.2 Multi-Layer Neural Networks

Neurons are arranged by layer. There are no connections between neurons in the same layer; connections are made only with neurons in subsequent layers. Typically, each neuron in a layer is only connected to all neurons in the next layer. This allows us to introduce the concept of the direction of information (activation) flow within a network and thus define the concepts of input and output neurons. By extension, the set of input neurons is called the input layer, and the set of output neurons is called the output layer.

Intermediate layers with no contact with the outside are called hidden layers. [25]
Figure (2.7) represents a feedforward neural network with a specific structure that is very frequently used: it includes inputs, two hidden layers of neurons, and output neurons. The neurons in the hidden layer are not connected to each other.
This structure is called a Multilayer Perceptron (MLP).

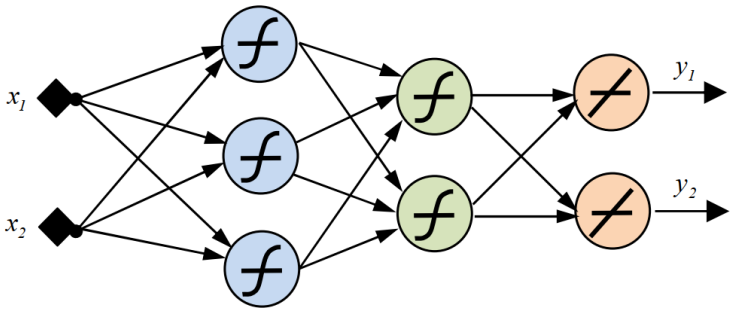

Figure 2.7: Multilayer Neural Network.

2.3.2 Recurrent Neural Networks

Unlike feedforward neural networks, where the connection graph is acyclic, recurrent neural networks can have any connection topology. The Recurrent Neural Network saves the output of a layer and feeds this output back to the input to better predict the outcome of the layer. The first layer in the RNN is quite similar to the feedforward neural network, and the recurrent neural network starts once the output of the first layer is computed. After this layer, each unit will remember some information from the previous step to perform computation as a memory cell. [24] (Figure 2.8).

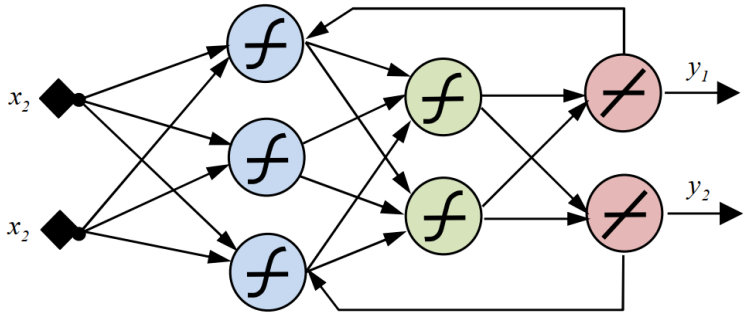

Figure 2.8: Multilayer Neural Network.

In this thesis, we will use the Multilayer Perceptron (MLP) neural network architecture.

2.4 Learning in Neural Networks

Learning in a neural network means changing its behavior to allow it to approach a defined goal. This goal is typically the approximation of a set of examples or the optimization of the network's state based on its weights to achieve the optimum of a predefined economic function. There are three main types of learning: supervised learning, unsupervised learning, and reinforcement learning [26].

2.4.1 Supervised Learning

Supervised learning is the adaptation of a network's synaptic coefficients (weights) so that, for each example, the network's output corresponds to the desired output.

2.4.2 Unsupervised Learning

Learning is unsupervised when the weights' adaptation depends only on the network's internal criteria. Adaptation is made solely with input signals, with no error signal or desired output being considered.

2.4.3 Reinforcement Learning

Learning is reinforcement-based when the neural network interacts with the environment. The environment rewards a satisfactory response from the network and assigns a penalty otherwise. The network must discover the reactions that give it the maximum rewards.

The choice of using a particular neural network architecture or learning type depends on the application and the processing capabilities of the system on which these architectures will be implemented. In our case, MultiLayer Perceptron (MLP) is a supervised learning neural network.

2.5 Backpropagation

Backpropagation is currently the most commonly used rule for supervised learning in neural networks; it is a technique for calculating derivatives that can be applied to any different function structure. It is generally used for multilayer neural networks, also called perceptrons. [27]

2.5.1 Principle of Backpropagation

Consider a non-recurrent multilayer network with m inputs and n outputs, composed of l layers ($l-1$ hidden layers + output layer). The following equations give the states of the neurons in layer k: [27]

$$O_i^k(t) = f_k[S_i^k(t)] \quad \text{for } i = 1, \ldots, n_k \tag{2.1}$$

$$S_i^k(t) = \sum_{j=0}^{n_{k-1}} W_{ij}^k O_j^{k-1}(t) \quad \text{for } k = 1, 2, \ldots, l \tag{2.2}$$

Where for layer k:

- f_k: the activation function;
- n_k: the number of neurons;
- O_i^k: the output of neuron i;
- W_{ij}^k: the synaptic weight of the relation between neuron i in layer k and neuron j in the previous layer $k-1$. (Figure 2.9)

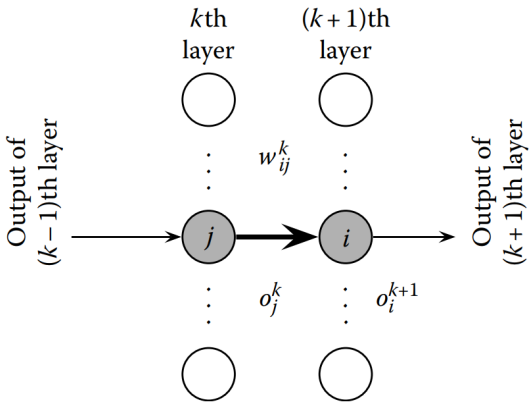

Figure 2.9: Sub/superscription of variables in feedforward NNs. [19]

A representative set of learning samples is available: a set of input/output pairs. The objective is to adapt the weights W to minimize the mean value of the global quadratic error over the entire learning set, expressed by: [27]

$$E = \frac{1}{2}\sum_{t=1}^{T} E(t)E(t)^t = \frac{1}{2}\sum_{t=1}^{T}[y^d(t) - y(t)][y^d(t) - y(t)]^t \tag{2.3}$$

Where:

- y^d: is the desired output vector;
- y: is the output vector of the neural network;
- T: is the length of the learning set.

The learning operation begins with a random choice of the initial values of the weights. At each step, the samples are presented to the network input. After propagation, the network output and the corresponding global error are available.

By backpropagation of the global error, the error gradients for all weights are calculated, and the parameters are adjusted in the direction opposite to these error gradients, for example, by applying the following adaptation rule:

$$W_{ij}^k(n) = W_{ij}^k(n-1) + \Delta W_{ij}^k(n) \tag{2.4}$$

$$\Delta W_{ij}^k(n) = -\mu \frac{\partial E}{\partial W_{ij}^k(n)} \tag{2.5}$$

which is known as the MIT rule, steepest or gradient descent. [27]

The adjustment of biases is done in the same manner as the weights.

μ is the learning rate (positive constant), and n is the iteration number.

2.5.2 Learning Algorithm

Step 1:

- Initialize the weights W_{ij}^k with small values, typically in the range $[-1, 1]$.

Step 2:

- Present an example and calculate the output and the corresponding error using equations (2.1), (2.2), (2.3), (2.4), and (2.5).

Step 3:

- Compute the partial derivatives of the error for each weight.
  - For the first technique:
    - Set $\Delta W_{ij}^k(n) = [\Delta W_{ij}^k(n)]_P$
    - Go to Step 4
  - For the second technique:

* If $P \neq T$, return to Step 2
* Otherwise, $\Delta W_{ij}^{k}(n) = \sum_{t=1}^{T}[\Delta W_{ij}^{k}(n)]$

Step 4:

- Adjust the parameters using equation (2.4).

Repeat Steps 2 to 4 until the maximum number of iterations is reached or the error threshold is achieved. [19]

In its basic form, the backpropagation algorithm uses the technique of gradient descent, which is among the simplest but is generally inefficient because it uses little information about the error surface. In the literature, there are a large number of more sophisticated techniques derived from it, some of which can be mentioned:

- Gradient descent with variable learning rate;
- Resilient backpropagation;
- Fletcher-Reeves algorithm;
- Quasi-Newton algorithm;
- Levenberg-Marquardt algorithm.

Among these, the ***Levenberg-Marquardt*** (LM) algorithm stands out for its computational speed compared to the basic backpropagation method. In our work, we opted for the LM algorithm to exploit its capability to leverage second-order information, such as the Hessian matrix or its approximation, thereby enhancing the training process efficiency.

2.6 Development of the Neural Controller

Before designing the neural networks, it is essential to construct the training dataset. We collected several samples of input and output observations over a total duration of 5 seconds. The input and output variables were stored in input and output matrices. This step allows the neural networks to use these matrices in the supervised learning process to mimic the control law closely.

The two input signals and the target output are exported to the workspace. In our application, we wrote a small code (see Appendix B) and employed the "Neural Networks" toolbox available in MATLAB [23] to create and train the MLP (Multilayer Perceptron) neural network using the previously defined dataset.
Figure (2.10) illustrates our MLP network topology.

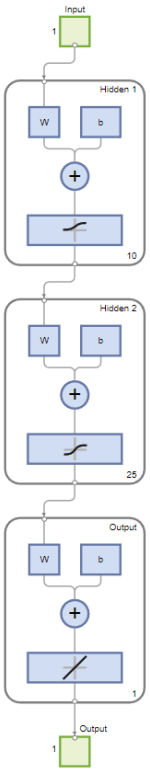

Figure 2.10: Topology of the neural network.

Each neural network fulfils a well-defined function depending on its selected architecture, such as the number of hidden layers and neurons in each hidden layer. The challenge lies in finding the architecture that yields the best results. We conducted several tests to determine the optimal network architecture to address this. We found that the most effective choice was a neural network with one (1) input layer using *'logsig'*, one (1) output layer using *'purelin'*, and two hidden layers containing ten (10) and twenty-five (25) neurons respectively also using *'logsig'* and *'tansig'* sigmoid functions for activation (Figure 2.11).

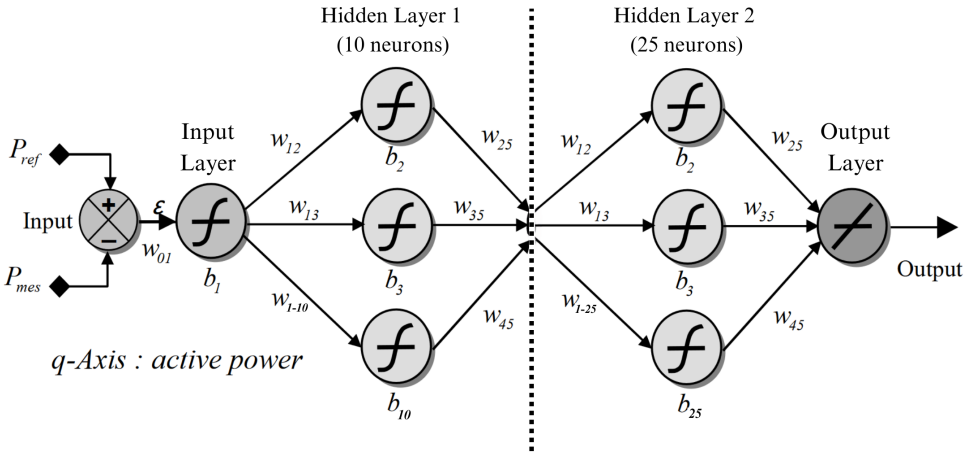

Figure 2.11: MLP structure (1-10-25-1).

It is also essential to determine other parameters of the neural network. Training was achieved by presenting 564.888 examples to the network with a maximum error of 10^{-12} and a maximum iteration count of 1500. The training method was backpropagation, specifically the Levenberg-Marquardt (LM) algorithm. This algorithm is known for its speed and excellent convergence towards a minimum quadratic error, especially for function approximation problems with fewer network weights than a few hundred.

Additionally, a cross-validation approach was employed, where the dataset was divided into three subsets: 70% for training, 15% for validation, and 15% for testing. This allowed for a thorough evaluation of the model's performance and generalization ability

Figure (2.12) and Figure (2.13) show the error evolution over the iterations for both Active and reactive power's MLP.

The results are pretty good considering the large size of the dataset (564888x1 double) for both the input and the output of each. The training process achieves an error of $0,0086$ at the 200th iteration for Active Power and $0,0027$ at the 480th iteration for Reactive Power, indicating good learning performance.

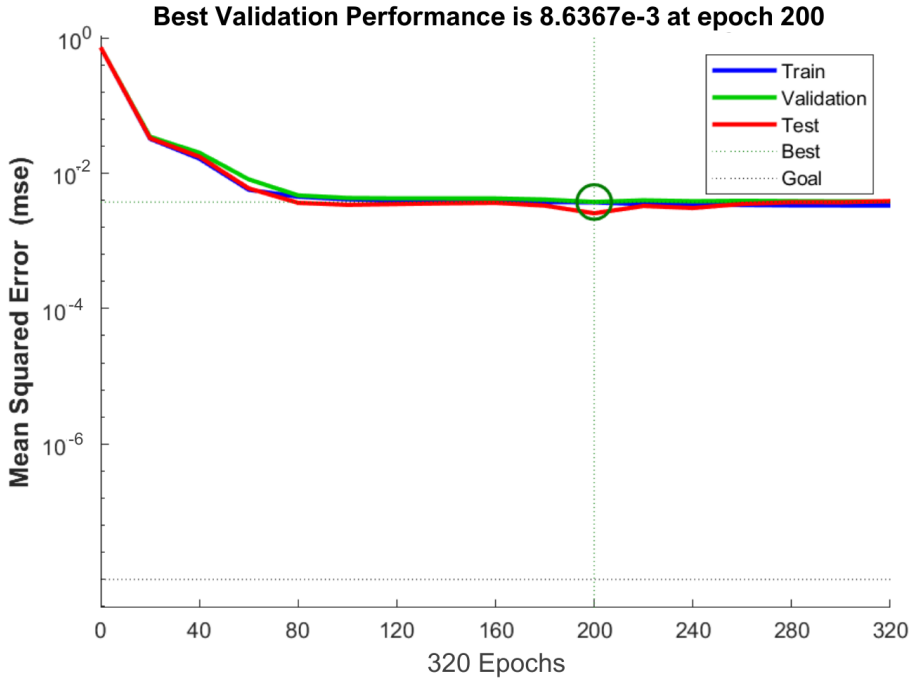

Figure 2.12: MLP learning for Active Power.

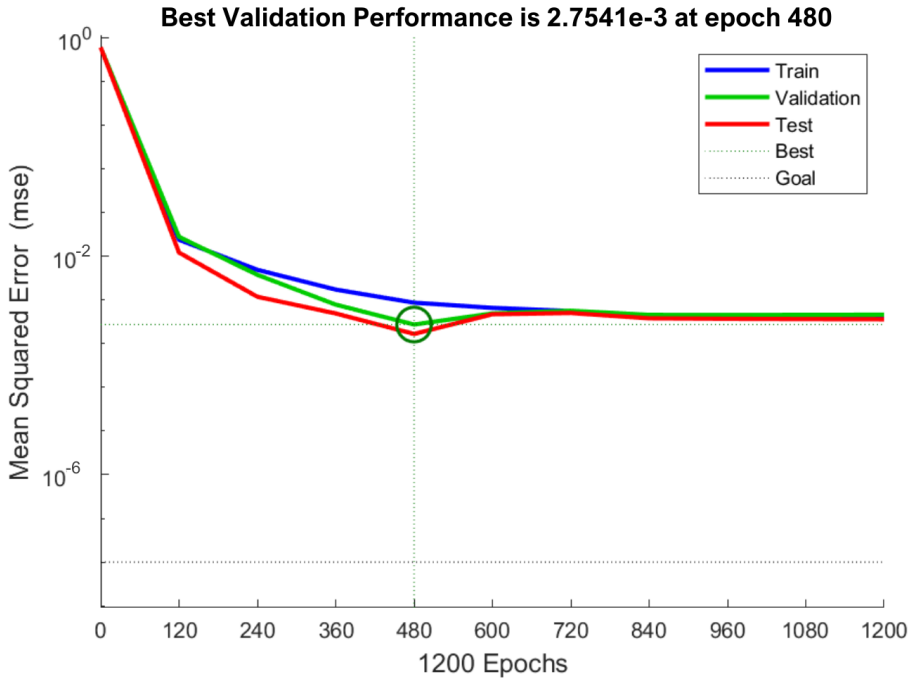

Figure 2.13: MLP learning for Reactive Power.

Discussion

- The best validation performance for both models is remarkably low, indicating a robust mean squared error (MSE) in the order of (10^{-3}); This low MSE signifies strong performance on the validation set, reflecting the network's ability to generalize effectively.
- The training error, represented by the blue line, shows a consistent decrease, stabilizing at a minimal value. This trend highlights the effectiveness of the LM algorithm in minimizing the error during the training phase.
- The validation error (green line) closely follows the training error, also stabilizing at a low value. This behavior suggests that the network maintains a high level of generalization, avoiding overfitting to the training data.
- The test error (red line) remains low, further corroborating the model's robustness and ability to perform well on unseen data.
- Both models terminated training after reaching the optimal error rates at 200 and 480 epochs, respectively. This early stopping criterion prevents overfitting and conserves computational resources, ensuring efficient training processes

To complete the analysis, we also present the regression plots for both Active Power (Figure 2.14) and Reactive Power (Figure 2.15) . The regression coefficients (R) values indicate the goodness of fit of our regression model to the data.

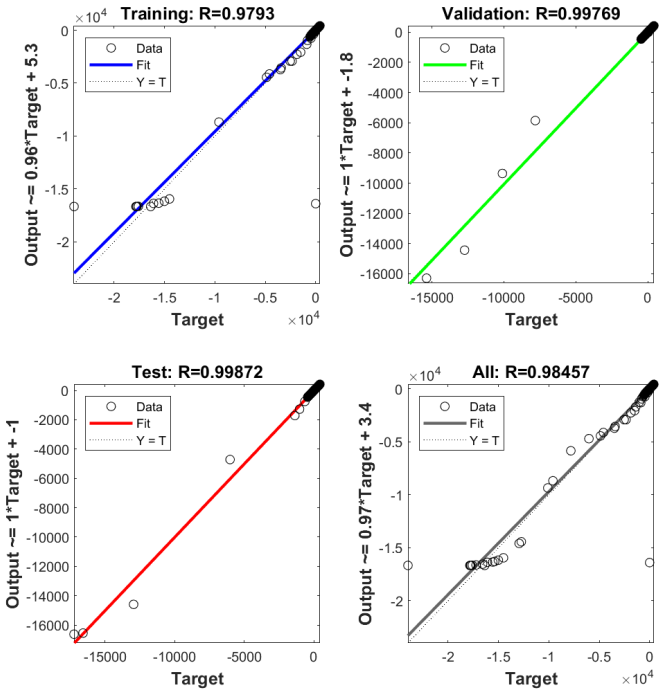

Figure 2.14: Regression plot for Active Power.

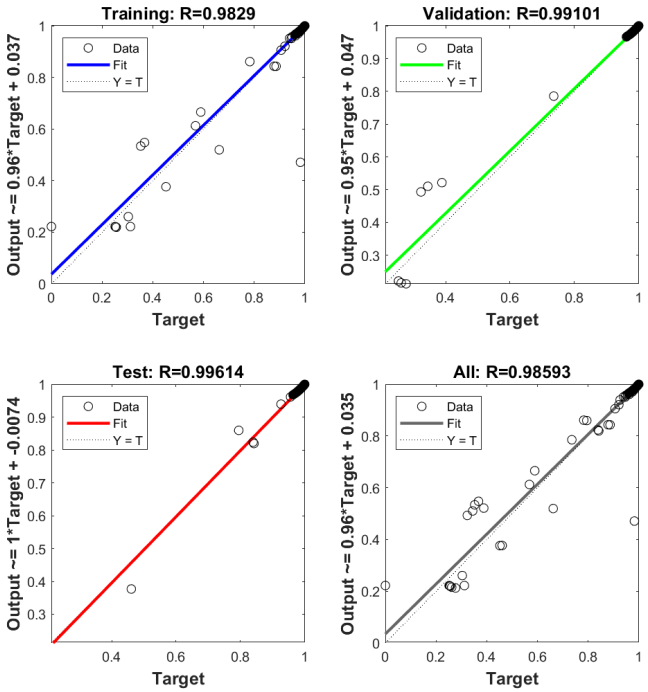

Figure 2.15: Regression plot for Reactive Power.

These values indicate the goodness of fit of our regression model to the data. For Active Power, the training R value is 0.9793, and for Reactive Power, it is 0.9829.

These high R values suggest that our regression model provides a solid fit for the training data, indicating the effectiveness of our ANN controller in capturing the relationship between the input and output variables.

To summarize, Figure (2.16) illustrates a flowchart of the general principles and steps in designing the ANN controller.

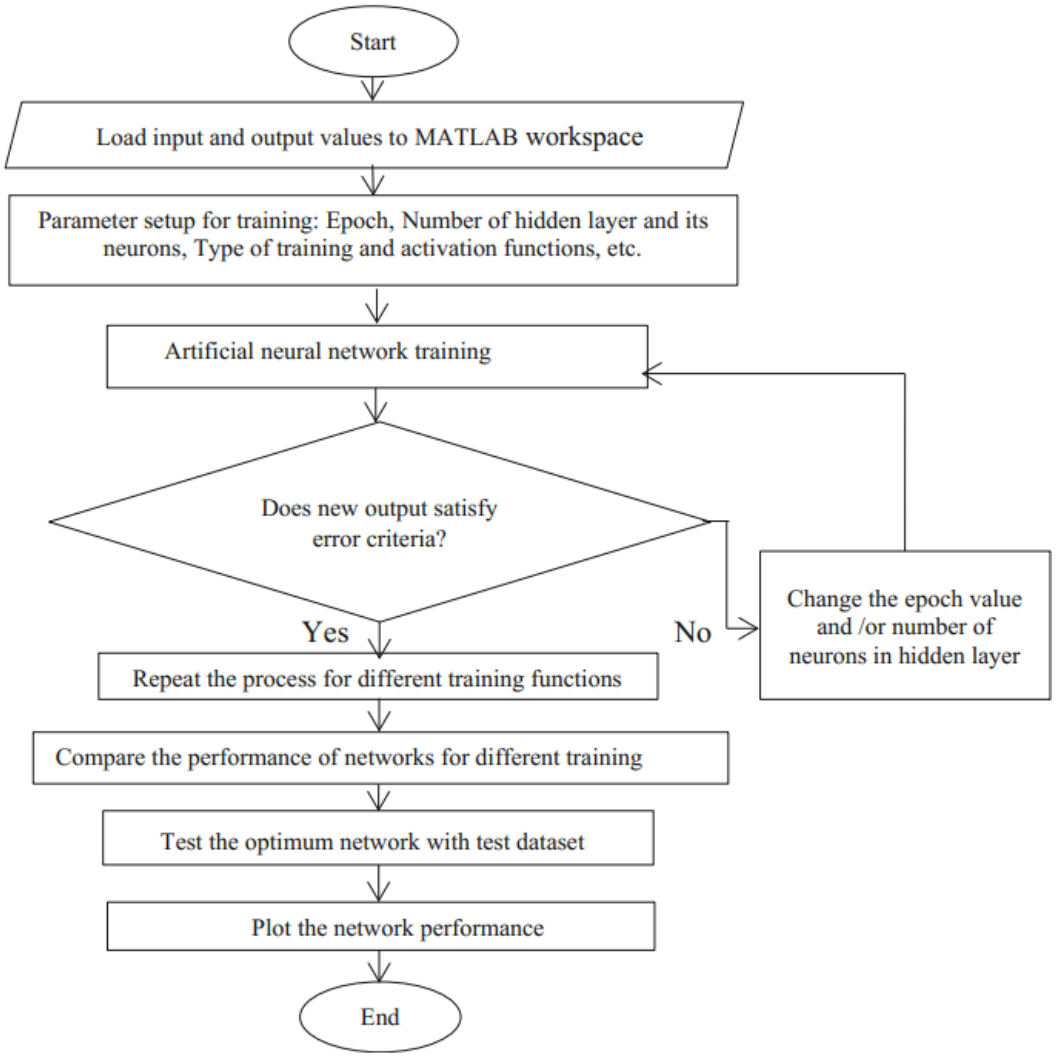

Figure 2.16: ANN Design Flowchart.

2.6.1 Simulation Results

To validate the direct control technique for active and reactive powers of the DFIG using ANN controllers, a simulation study was conducted in the SIMULINK/MATLAB environment. (Figure 2.17)

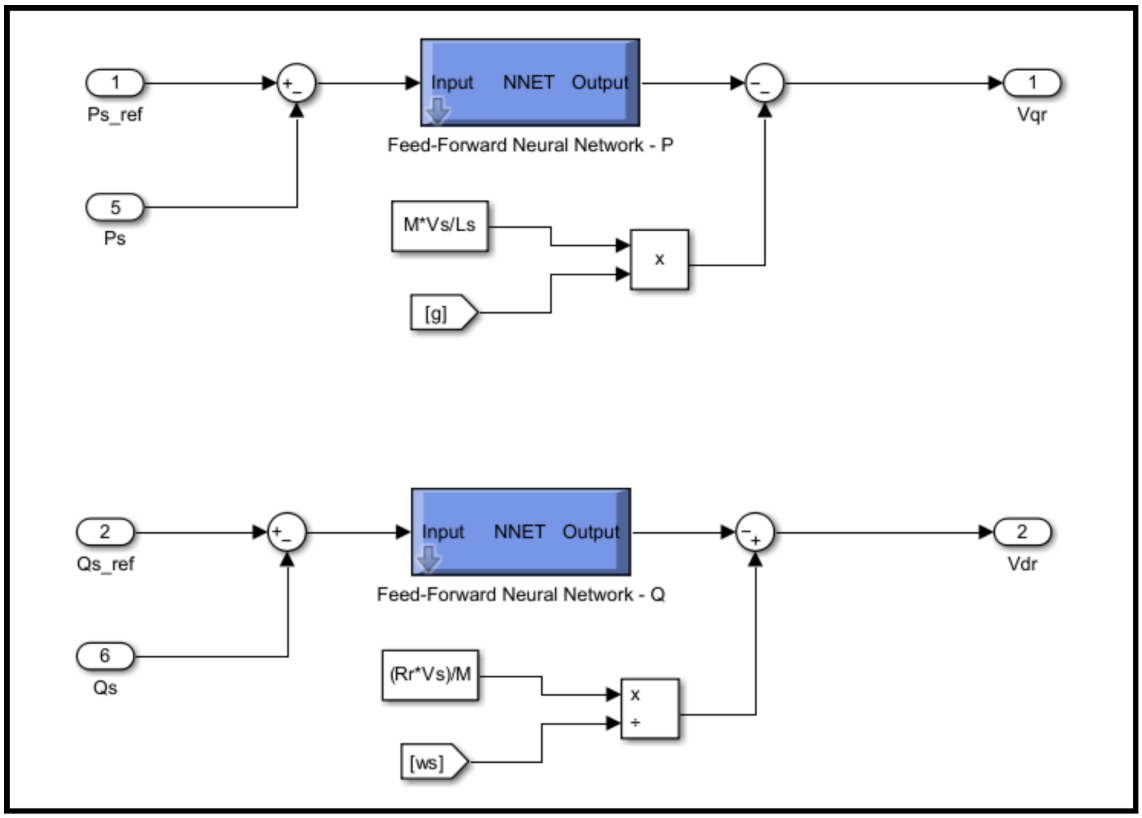

Figure 2.17: Simulink diagram of DFIG powers control using ANN controller.

To evaluate our control chain, the regulation of this system uses reference power values shown in the table below:

Table 2.2: Active and reactive power references.

Time Interval (s)	$P_{\mathbf{ref}}$ **(W)**	$Q_{\mathbf{ref}}$ **(Var)**
[0s – 3s]	-4000	0
[3s – 4s]	-7500	0
[4s – 5s]	-7500	-2000

The parameters of the DFIG used for the simulation are detailed in Appendix A. Several tests were performed to assess the influence of the chosen neural control on the system's dynamic and static performance. The tests aimed to evaluate if the ANN controller meets key requirements such as:

- Good system response without overshoot;
- Minimal steady-state error with effective disturbance rejection;
- Robustness against parametric variations of the DFIG.

2.6.1.1 Setpoint Tracking Test

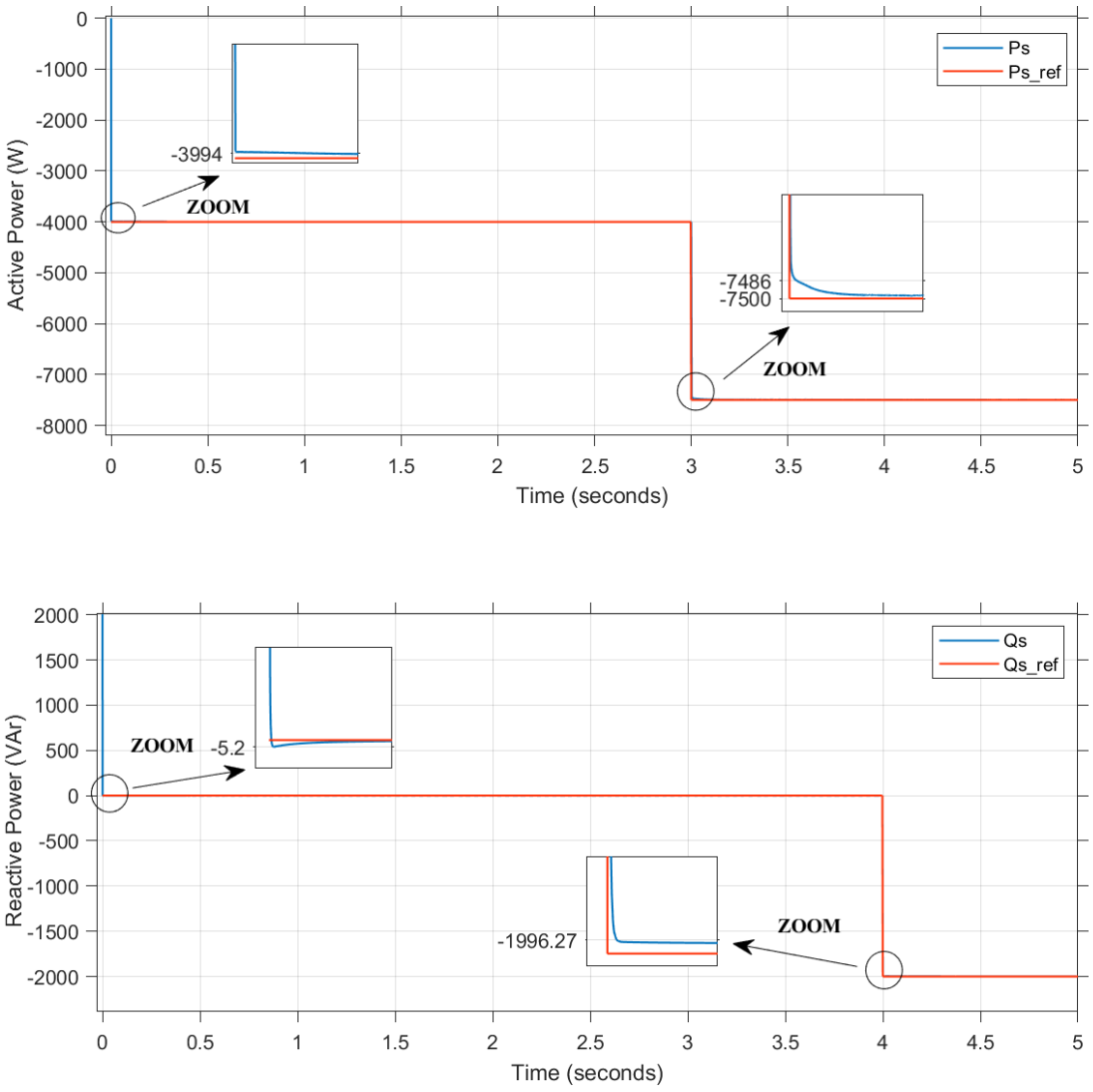

Figure 2.18: Active and Reactive powers direct control with ANN Controller.

Interpretation:

The simulation results depicted in Figure (2.18) demonstrate that the DFIG system, controlled by the ANN controller, successfully tracks the setpoint references for both active power (P_s) and reactive power (Q_s).

- The active power (P_s) quickly aligns with the reference (P_{ref}) with a slight initial deviation. This initial error is rapidly corrected, indicating a quick response time.
- When the reference power changes at 3 seconds from -4000 W to -7500 W, the system adapts promptly. The detailed zoom in the plot highlights a small transient response, which is swiftly damped out, showcasing the system's ability to handle setpoint changes with minimal overshoot and negligible steady-state error.

- Similarly, the reactive power (Q_s) follows its reference (Q_{ref}) effectively from the beginning. The initial response shows a minor deviation, which is corrected swiftly.
- When the reactive power setpoint changes at 4 seconds to -2000 Var, the system responds quickly and accurately. The zoomed-in section illustrates a brief transient that is quickly dampened, demonstrating the controller's ability to handle reactive power adjustments efficiently.
- The results also highlight the decoupling between the d-axis and q-axis controls. Changes in active power do not induce oscillations in reactive power and vice versa. This is crucial for the stability and performance of the DFIG system, ensuring that the control of one power component does not adversely affect the other.
- The system shows a minimal response time, rapidly reaching the setpoints without significant delays; both active and reactive power settle closely to their respective references, with almost negligible steady-state error; the plots show an absence of significant overshoot during setpoint changes, contributing to the stability and reliability of the control system.

In conclusion, the ANN controller demonstrates efficient control over the DFIG system, providing precise setpoint tracking for both active and reactive powers with minimal transient effects, excellent decoupling of control axes, and rapid adaptation to setpoint changes.

2.6.1.2 Robustness Test

a) To evaluate the system's ability to reject disturbances and track reference signals, a disturbance was introduced at ($t = 2.5\,\text{s}$). This disturbance corresponds to a sudden step change in the rotational speed of the DFIG shaft, increasing from 140 rad/s to 185 rad/s. The objective was to verify if the system could compensate for this disturbance and maintain the reference value trajectory.

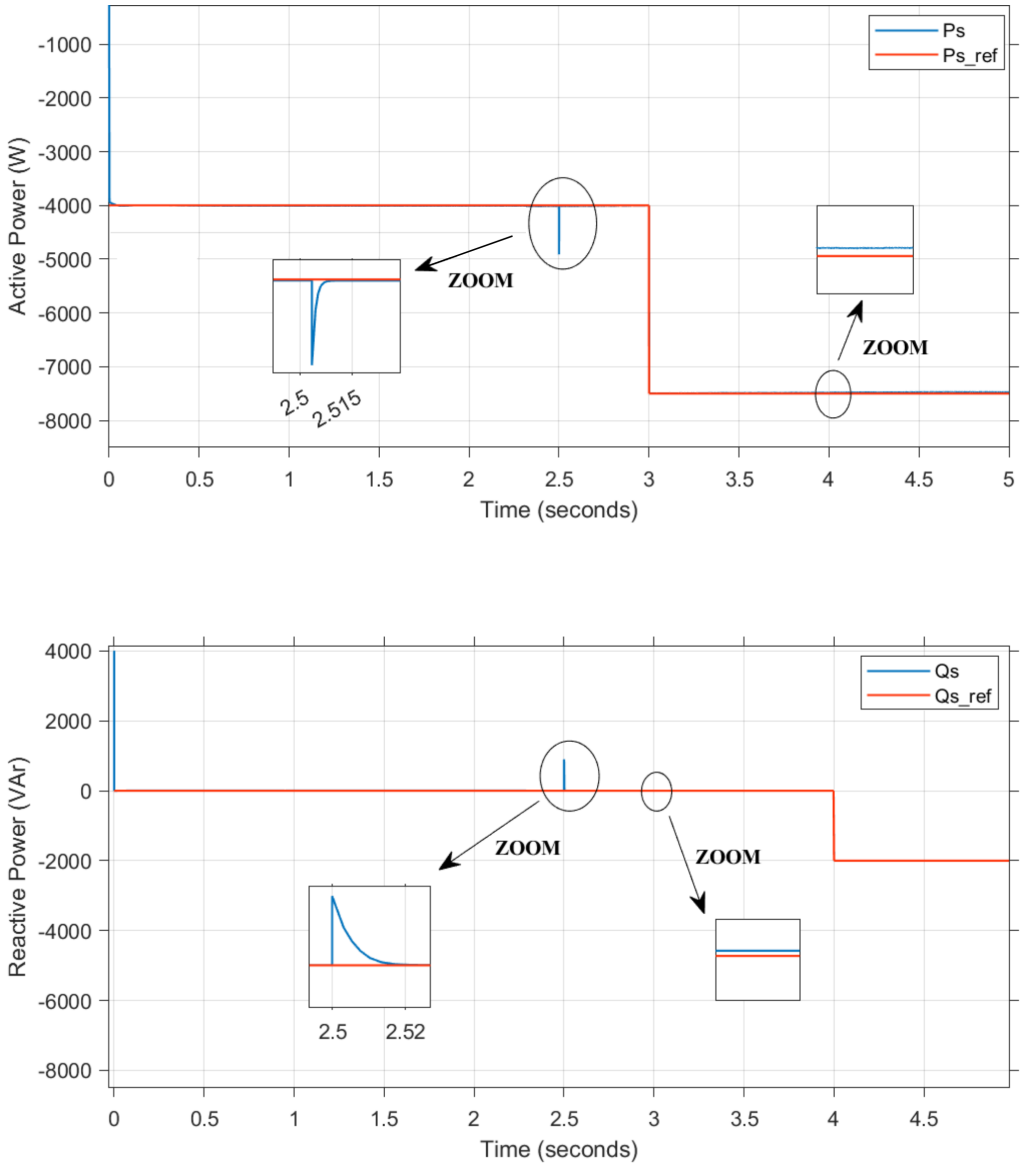

Figure 2.19: Active and Reactive powers after applying a disturbance (ANN).

Interpretation: The results demonstrate the DFIG system's ability to rapidly track reference signals for both active and reactive powers, even under disturbance conditions. At $t = 2.5\,\text{s}$, a sudden step change in the rotational speed of the DFIG shaft induces a temporary deviation in stator power outputs. However, the system quickly compensates for this disturbance, with both active (P_s) and reactive (Q_s) powers realigning with their respective reference signals within milliseconds. This quick realignment, without additional overshoot, underscores the robustness of the ANN controller.

Furthermore, the test reveals effective decoupling between the d and q control axes. In other controllers, changes in active power minimally influence reactive power and vice versa, but here, this wasn't the case, indicating that the controller successfully isolated the two power components. This robustness in maintaining decoupled con-

trol actions is crucial for ensuring stable and accurate power regulation in varying operational conditions.

Overall, the system's ability to handle disturbances and swiftly return to the desired performance metrics highlights the efficacy of the ANN controller in maintaining robust and reliable control over the DFIG's power outputs.

b) Given that the machine's parameters are not always precisely known and can vary with operating conditions (e.g., heating, load variation, magnetic saturation), another robustness test included varying the parameters of the DFIG model. Figure (2.20) shows the behavior of the direct control when the rotor resistance R_r is increased by 100% of its nominal value, simulating overheating of the DFIG.

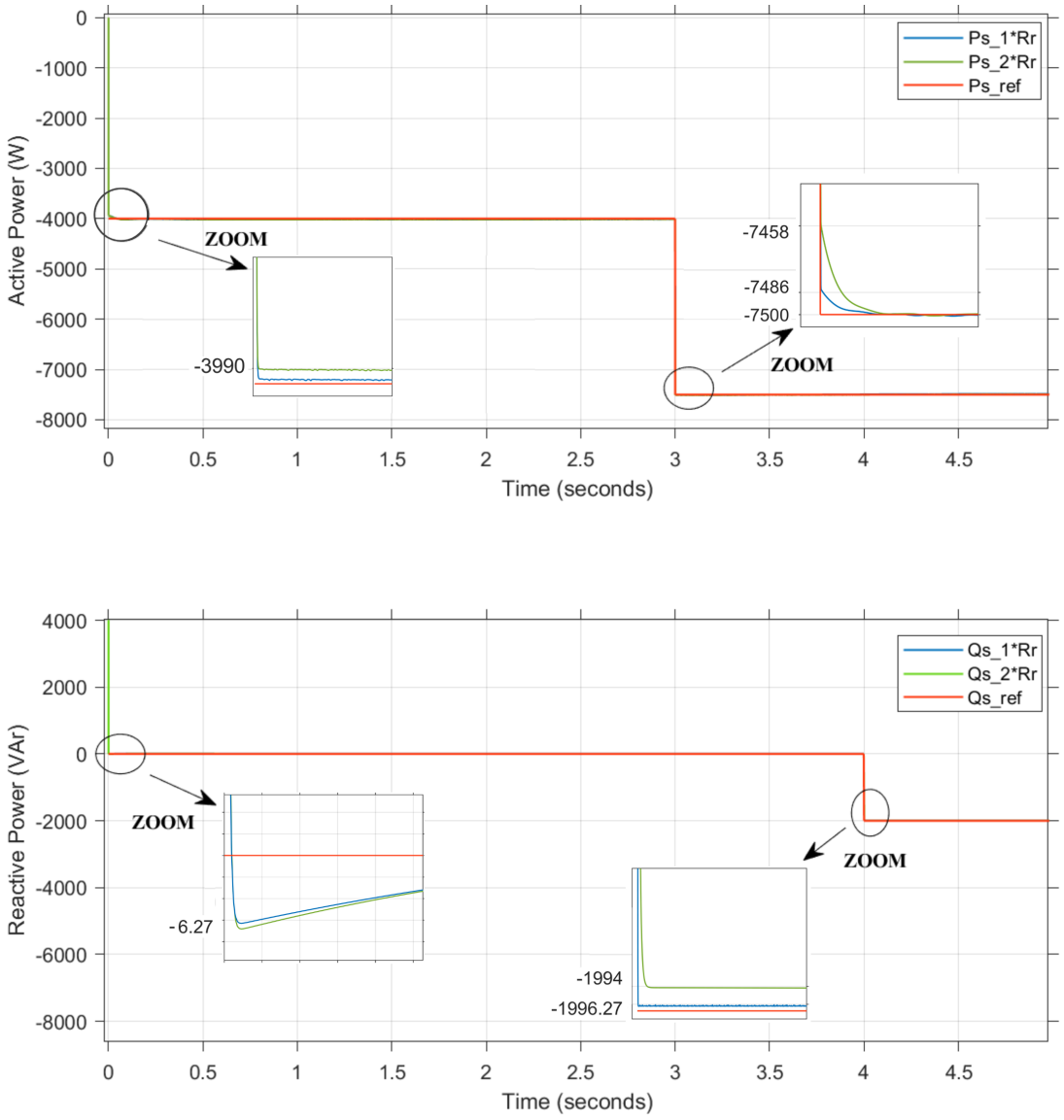

Figure 2.20: Active and Reactive powers after changing the parameter Rr (ANN).

Interpretation: The simulation results reveal that a 100% increase in rotor resistance R_r caused only a slight increase in response time and error for both active

and reactive powers. Despite the significant parametric variation, the ANN controller effectively maintained control, demonstrating its good robustness. The active power (P_s) and reactive power (Q_s) continued to follow their reference signals with minimal performance degradation closely.

The slight increase in response time and error indicates that while the system is sensitive to extreme changes in rotor resistance, the overall impact on performance is minimal. This resilience is critical in practical applications where parameter variations are inevitable due to factors like overheating or wear and tear. The ability of the ANN controller to adapt and maintain effective control under these conditions underscores its robustness and reliability, making it a suitable choice for the DFIG power control in real-world scenarios.

2.7 Conclusion

In this chapter, we introduced artificial neural networks, their principles, structure, and various architectures. We designed a Multilayer perceptron Artifical neural network controller for the field-oriented control of active and reactive powers of the DFIG. The results were very promising, demonstrating the controller's effectiveness in terms of response time, steady-state error, and robustness against disturbances and parametric variations. However, some disadvantages of the ANN controller were identified, such as the sensitivity to the quality and quantity of training data (in our case, the data set is relatively too large.) and the potential for over-fitting if not adequately regularized.

To counter these issues, the next chapter will explore the use of an ANFIS controller, a control system based on neural networks and fuzzy logic, aiming to leverage the strengths of both approaches for improved performance.

Chapter 3

Adaptive Neuro-Fuzzy Inference System (ANFIS) Control of DFIG Powers.

3.1 Introduction

The need for advanced control techniques in wind energy conversion systems, particularly those utilizing Doubly-Fed Induction Generators (DFIGs), has become increasingly evident in the quest to enhance performance and reliability. While Artificial Neural Networks (ANNs) have shown considerable promise in controlling the active and reactive powers of DFIGs, their limitations, such as sensitivity to initial conditions and dependency on training data quality, necessitate the exploration of alternative control strategies. This chapter introduces an alternative approach: the Adaptive Neuro-Fuzzy Inference System (ANFIS) as a robust and versatile control methodology for DFIGs.

In this chapter, we begin by discussing the fundamental principles of ANFIS, including its structure, functioning, and advantages over traditional control methods. We then provide a comprehensive description of the design and implementation of an ANFIS-based controller specifically tailored for the direct control of active and reactive powers in DFIGs. Following this, simulation results are presented to evaluate the performance of the ANFIS controller under various operating conditions, disturbances, and parametric variations. A comparative analysis with the previously studied ANN controller is also included to highlight the relative merits and potential enhancements offered by the ANFIS approach.

Through this exploration, we aim to demonstrate that ANFIS can significantly improve the robustness, adaptability, and overall control quality of DFIG systems, thereby contributing to more efficient and reliable wind energy generation.

3.2 Definition

Adaptive Neuro-Fuzzy Inference System (ANFIS) combines the strengths of artificial neural networks (ANN) and fuzzy logic to create a powerful hybrid approach for handling complex, non-linear systems (Jang 1993). Fuzzy logic excels at transforming qualitative human knowledge into precise quantitative analysis. However, it lacks a structured method for transforming human thought processes into a rule-based fuzzy inference system (FIS) and requires considerable time to fine-tune membership functions (MFs) (Jang 1993). On the other hand, ANN demonstrates superior learning capabilities, allowing it to adapt effectively to its environment [28]. By integrating these two methodologies, ANFIS employs ANN to automatically adjust the MFs, thereby enhancing the efficiency and accuracy of the fuzzy logic rule determination process.

ANFIS leverages the adaptive learning capabilities of neural networks and the transparent, rule-based reasoning of fuzzy logic systems. This hybrid approach offers a robust control strategy capable of managing the non-linearities, uncertainties, and complex dynamics inherent in the operations of Doubly-Fed Induction Generators (DFIGs).

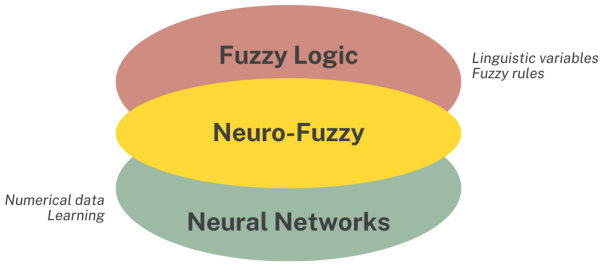

Figure 3.1: Principle of Neuro-fuzzy.

3.3 Advantages and Disadvantages of Fuzzy Logic & Neural Networks

The simultaneous use of neural networks and fuzzy logic allows us to leverage the advantages of both methods: the learning capabilities of the former and the readability and flexibility of the latter.

To summarize the contributions of neuro-fuzzy systems, Table (3.1) presents the advantages and disadvantages of fuzzy logic and neural networks.

Neuro-fuzzy systems are created to synthesize the advantages and overcome the disadvantages of neural networks and fuzzy systems. Learning algorithms can be employed to determine the parameters of fuzzy systems. This effectively automates the creation or improvement of a fuzzy system using methods specific to neural networks. An important aspect is that the system remains interpretable in terms of fuzzy rules, as it is based on a fuzzy system.

Table 3.1: Comparison between Fuzzy Logic and Neural Networks

Neural Networks	**Fuzzy Logic**
Advantages	
No rule-based knowledge is required	Prior knowledge of rules can be used
No mathematical model required	No mathematical model required
Various learning algorithms available	Simple interpretation and implementation
Disadvantages	
Black box (lack of traceability)	Cannot learn
Adaptation to different environments is difficult and requires relearning	Rules must be available
Internal knowledge cannot be used (learning from scratch),	Difficult adaptation to changing environments
No guarantee of learning convergence	No formal methods for adjustment

Neuro-fuzzy systems combine the strengths of neural networks and fuzzy logic to create a powerful and flexible approach to modeling and control. They can learn and adapt like neural networks while maintaining the interpretability and ease of use associated with fuzzy logic systems.

3.4 Fuzzy Inference System

A Fuzzy Inference System (FIS) is built on three main components: basic rules (which consist of "If-Then" fuzzy logic rules), fuzzy set membership functions, and reasoning mechanisms that use fuzzy inference techniques to produce outputs from the basic rules [28]. Figure (3.2) illustrates the structure of the fuzzy inference system.

FIS operates by converting the input containing actual values into fuzzy values through the process of fuzzification. The membership function used in this process assigns fuzzy values ranging between 0 and 1.

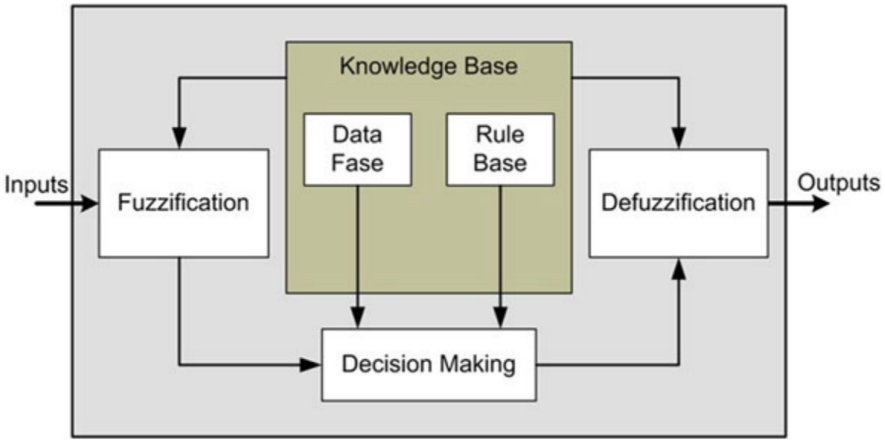

Figure 3.2: Fuzzy Inference System

The basic rules and databases together form the knowledge base, which is crucial for decision-making. The database typically includes definitions such as parameters of fuzzy sets for every linguistic variable. Developing a database involves defining a universe, determining the number of linguistic values for each variable, and establishing membership functions. Based on the rules, it contains fuzzy logic operators and conditional "If-Then" statements.

Basic rules can be derived either from human expertise or through automatic generation using numerical input-output data. There are several types of FIS, with the Takagi–Sugeno and Mamdani models being the most common. The Takagi–Sugeno model is particularly prevalent in the application of ANFIS methods [28].

3.5 Adaptive Network

An adaptive network is a type of feed-forward neural network with multiple layers, as shown in Figure (3.3).

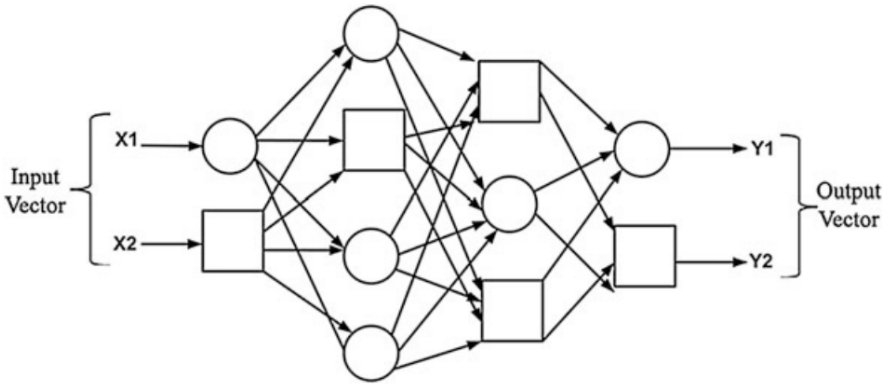

Figure 3.3: Adaptive Network. [27]

These networks often use supervised learning algorithms during the learning process. An adaptive network's architecture consists of several adaptive nodes interconnected directly without associated weight values. Each node performs different functions and tasks, and the output depends on the incoming signals and parameters available at the node. The learning rule used affects the node parameters, which in turn helps minimize errors in the adaptive network's output [28].

3.6 ANFIS Architecture

As introduced in the definition, Adaptive Neuro-Fuzzy Inference System (ANFIS) integrates the strengths of Artificial Neural Networks (ANN) and Fuzzy Inference Systems (FIS). ANFIS operates within the framework of the Sugeno fuzzy inference system. Its architecture resembles that of a multilayer feedforward neural network, with the primary distinction being that the links in ANFIS indicate the flow of signals without associated weights [28].

To illustrate the basic concept of ANFIS, consider two input variables, x and y, and one output variable, f. The system is governed by two fuzzy if-then rules as follows:

- **Rule 1**: If x is A_1 and y is B_1, Then $f_1 = p_1x + q_1y + r_1$
- **Rule 2**: If x is A_2 and y is B_2, Then $f_2 = p_2x + q_2y + r_2$

Here, A_1, A_2, B_1, B_2 are the membership functions for inputs x and y, respectively, and $p_1, q_1, r_1, p_2, q_2, r_2$ are the parameters identified through training [29].

3.6.1 ANFIS Layers

The ANFIS architecture is composed of five layers, each performing a distinct function in the fuzzy inference process. These layers are illustrated in Figure (3.4).

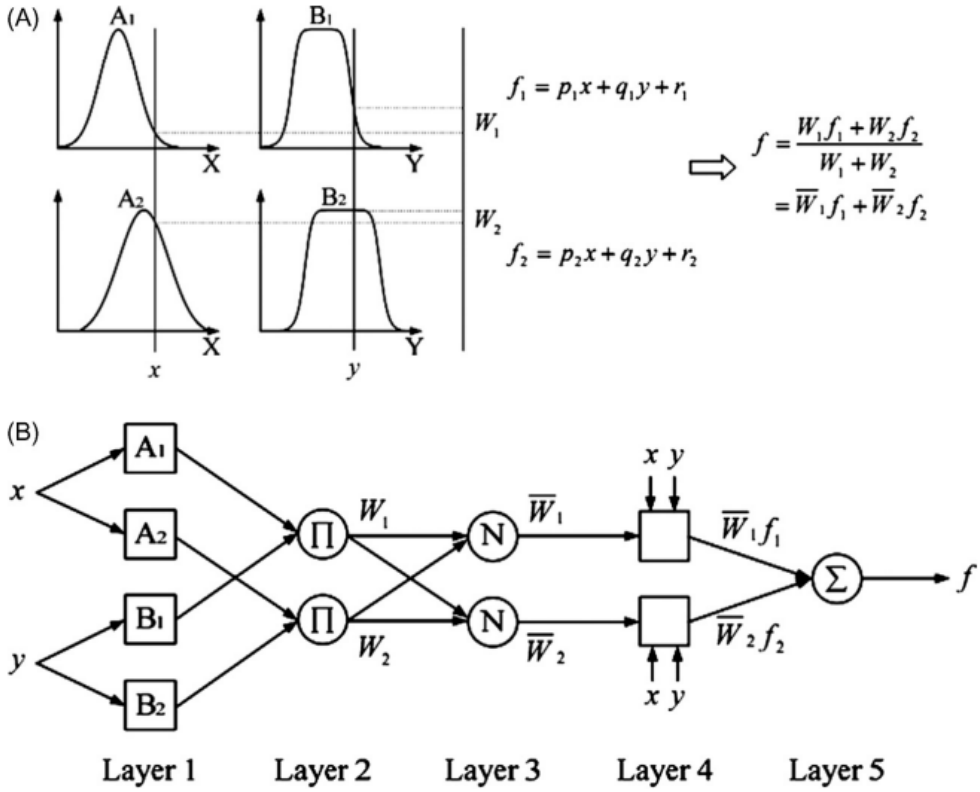

Figure 3.4: Sugeno fuzzy model and the corresponding ANFIS architecture. [27]

3.6.1.1 Layer 1: Fuzzification

In this layer, each node represents a membership function that outputs the degree of membership of the input variable. The membership functions can be of various types, such as Gaussian or bell-shaped functions. The output of each node is calculated as: [28]

$$O_i^1 = \mu_{A_i}(x), \quad i = 1, 2 \tag{3.1}$$

$$O_i^1 = \mu_{B_{i-2}}(y), \quad i = 3, 4 \tag{3.2}$$

Where μ_{A_i} and $\mu_{B_{i-2}}$ represent the membership functions for the fuzzy sets A_i and B_i .

3.6.1.2 Layer 2: Rule Firing Strengths

This layer contains fixed nodes that calculate the firing strength of each rule by multiplying the corresponding membership values from Layer 1. The output of each node is given by: [29]

$$O_i^2 = w_i = \mu_{A_i}(x) \cdot \mu_{B_i}(y), \quad i = 1, 2 \tag{3.3}$$

Where w_i represents the firing strength of the i-th rule.

3.6.1.3 Layer 3: Normalization of Firing Strengths

The nodes in this layer are also fixed. Each node normalizes the firing strength by dividing the firing strength of each rule by the sum of all firing strengths: [28]

$$O_i^3 = \bar{w}_i = \frac{w_i}{\sum_j w_j} \tag{3.4}$$

This normalization ensures that the firing strengths are proportionate.

3.6.1.4 Layer 4: Defuzzification

This layer contains adaptive nodes that calculate the contribution of each rule to the overall output. The output of each node is determined by: [29]

$$O_i^4 = \bar{w}_i f_i = \bar{w}_i(p_i x + q_i y + r_i) \tag{3.5}$$

Where $\bar{w}_i$ is the normalized firing strength from Layer 3, and p_i, q_i, r_i are the consequent parameters.

3.6.1.5 Layer 5: Output Layer

The single node in this layer is fixed and computes the final output by summing the contributions from all rules: [28]

$$O_i^5 = \sum_i \bar{w}_i f_i \tag{3.6}$$

This sum provides the overall output of the ANFIS model.

ANFIS's layered approach combines fuzzy logic's interpretability with neural networks' learning capabilities, making it a robust tool for modeling and controlling complex systems.

3.7 Learning Algorithm of ANFIS

The learning algorithm of the Adaptive Neuro-Fuzzy Inference System (ANFIS) uses neuro-adaptive learning techniques to model data sets effectively. These techniques adjust the membership function parameters to match the input/output data best, enhancing the model's accuracy during the learning process [30].

To address real-world problems efficiently, the ANFIS learning algorithm tunes all adjustable parameters to make the ANFIS output match the training data. A hybrid learning algorithm, which combines the least squares method (LSM) and the gradient descent method (GDM), is used to improve convergence. In the forward pass, LSM identifies the best values for the consequent parameters in Layer 4 while keeping the premise parameters fixed. The gradient vector helps measure how well the fuzzy inference system models the input/output data [31].

Once the optimal consequent parameters are determined, the backward pass begins. In this phase, errors are propagated back through the network, and the premise parameters are updated using the gradient descent method. This two-pass hybrid algorithm combines LSM for consequent parameter optimization and GDM for premise parameter tuning, effectively merging aspects of backpropagation and least squares estimation [30].

The table below summarizes the two passes in the hybrid learning algorithm for ANFIS:

	Forward Pass	**Backward Pass**
Premise Parameters	Fixed	Gradient Descent
Consequent Parameters	Least Squares	Fixed
Signals	Node Outputs	Error Signals

Table 3.2: Passes of the hybrid learning algorithm for ANFIS [30].

In this **batch training** process, the output error is used to adjust the premise parameters using a standard backpropagation algorithm, minimizing the mean square error function:

$$E(\theta) = \sum_{i=1}^{m} (z_i - a_i^T \theta)^2 = e^T e = (z - A\theta)^T (z - A\theta) \tag{3.7}$$

where $e = z - A\theta$ represents the error vector for a specific choice of θ. Minimizing this squared error using the least squares estimator (LSE) is fundamental to the hybrid algorithm, ensuring efficient training of the ANFIS [30].

Overall, the hybrid learning algorithm distinguishes between linear and nonlinear parameters, iteratively updating nonlinear parameters through gradient descent and optimizing linear parameters through least squares. This approach facilitates faster convergence and improves the accuracy of the ANFIS model.

3.8 Application of ANFIS Controller in DFIG

An Adaptive Neuro-Fuzzy Inference System (ANFIS) controller was implemented to enhance the control of the Doubly-Fed Induction Generator (DFIG). The ANFIS approach combines the learning capabilities of neural networks with the fuzzy logic inference system, providing a suitable control strategy for the active and reactive powers in our DFIG system.

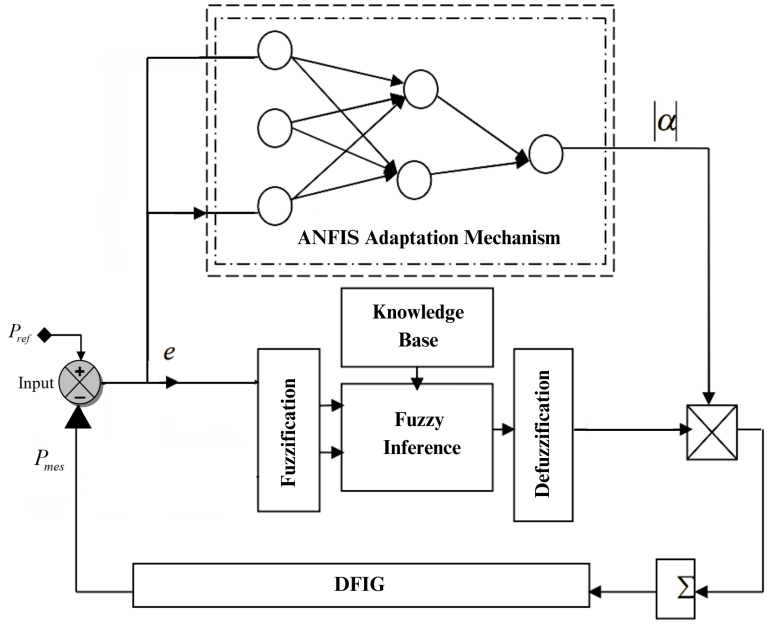

Figure 3.5: ANFIS Controller structure for the DFIG system.

3.8.1 Data Collection and Preprocessing

Similar to the ANN controller development, the first step in designing the ANFIS controller involved constructing the training dataset. From the same first controller used to train ANN previously, we collected several samples of input and output observations over a total duration of 5 seconds (564888x1 double). The input and output variables were stored in input and output matrices, respectively, which were then used in the supervised learning process of the ANFIS model.

3.8.2 Initial FIS Structure Generation

To generate the initial FIS structure, a grid partitioning method was employed. This method divides the input space into fuzzy regions defined by membership functions. And a rule is created for every possible combination of these MFs. Each rule's output corresponds to one output MF.

The structure of ANFIS consists of a single input (error e) and a single output. The number of epochs (iterations) used in this experiment is 100, with an error tolerance of 0.0001. The input was given ten (10) membership functions, providing a detailed representation of the input space. (Figure 3.6)

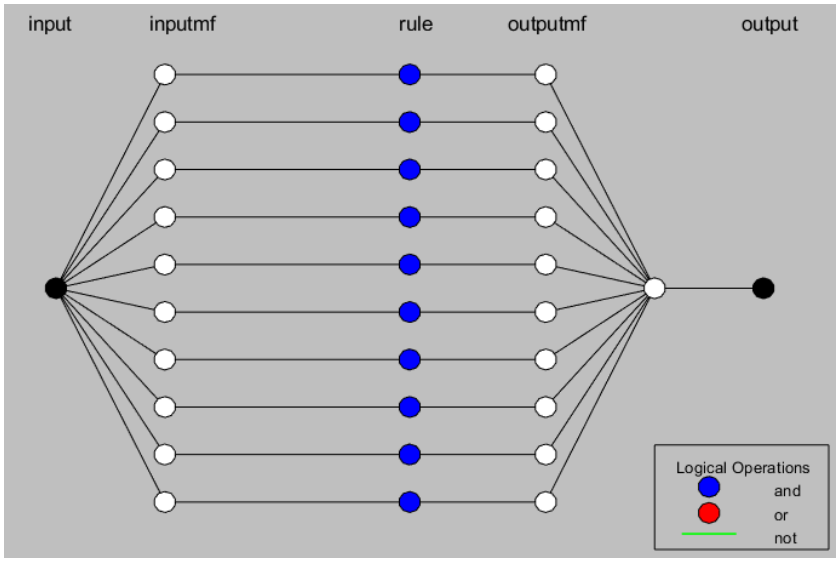

Figure 3.6: ANFIS Structure.

3.8.3 ANFIS Training in MATLAB

The ANFIS controller was designed and trained using a script that we've written (see Appendix B) and the ANFIS toolbox available in MATLAB. (Figure 3.7)

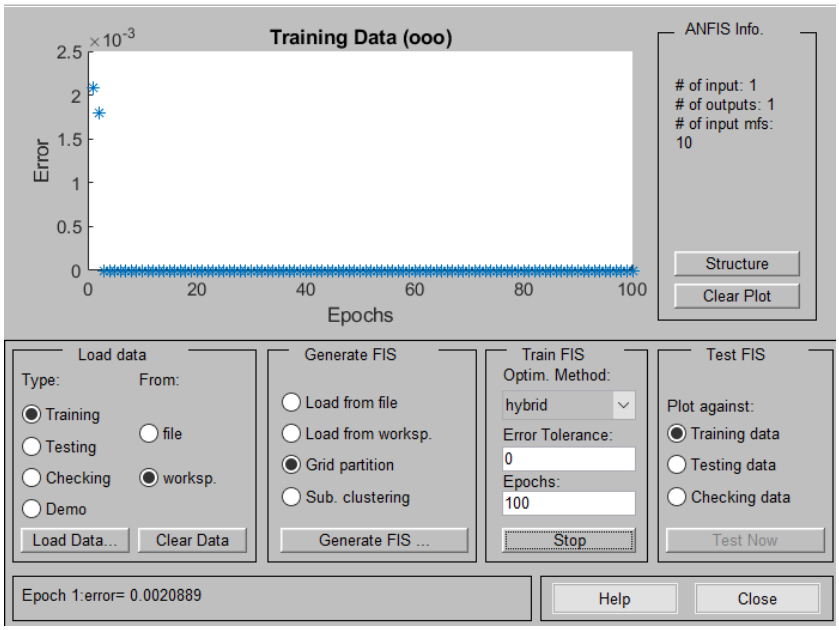

Figure 3.7: ANFIS GUI 'anfisedit' in MATLAB.

The training process involved adjusting the parameters of the membership functions and the fuzzy rules to minimize the error between the predicted and actual outputs. For this purpose, the hybrid learning algorithm, combining the least squares method and gradient descent (backpropagation), was used. The least squares method optimizes the consequent (output) parameters, while backpropagation fine-tunes the premise (input) parameters.

During training, the ANFIS model iteratively adjusted its parameters to reduce the mean squared error (MSE) between the predicted and actual outputs. The process continued until the error converged to a minimal value, indicating a well-trained model.

After training, the ANFIS produced a new FIS, which was tested using the *'evalfis'* function to predict the outputs based on the input data. The evaluation process involved feeding the input data into the trained ANFIS model and comparing the predicted outputs with the actual outputs. This step is crucial for assessing the model's performance and its ability to generalize to new, unseen data.

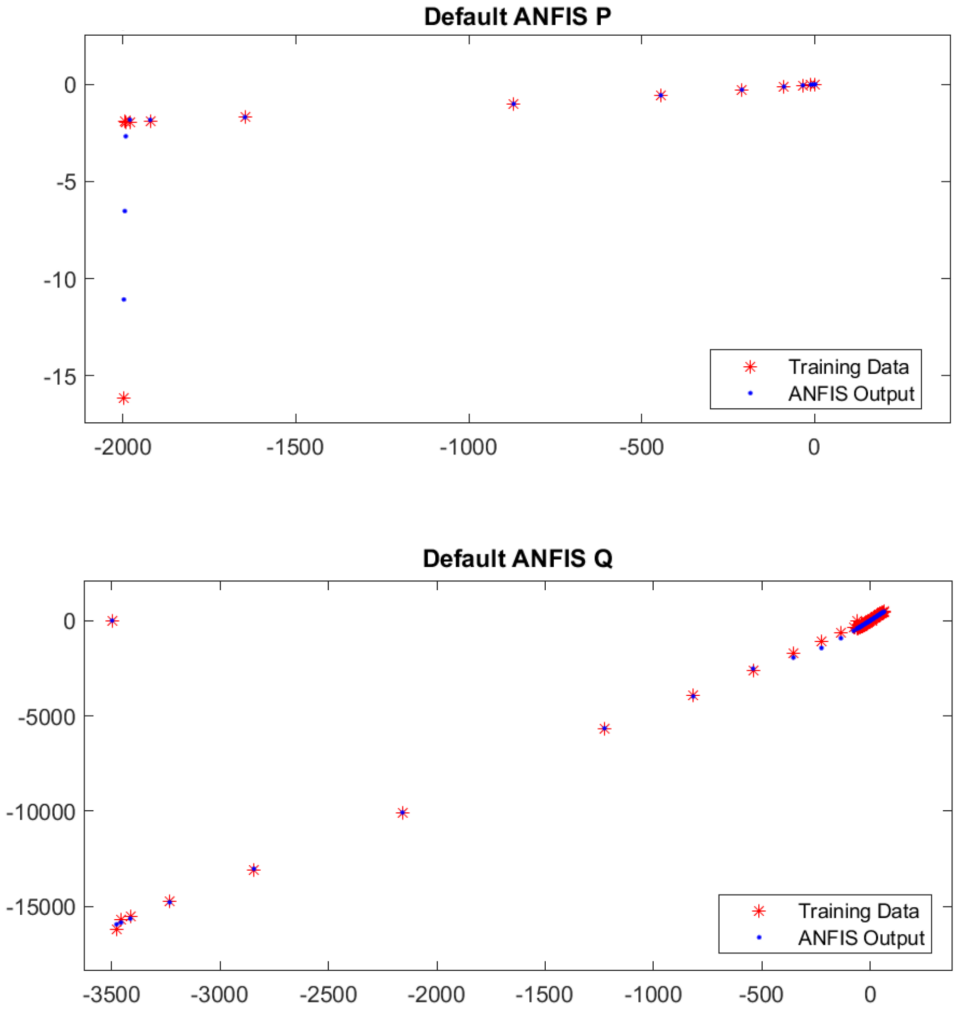

Figure 3.8: Training Data vs ANFIS Output.

Figure (3.8) illustrates this comparison: training data points are represented by circles, and the new FIS output points are depicted by stars. The circles and stars appear almost superimposed, indicating minimal adjustments made by the ANFIS during training. While these adjustments are not easily visible in the graph, they are evident in the new FIS's membership functions, where subtle modifications were made to the MFs and rules during the training process.

The validation performance of the ANFIS model indicated a low mean squared error (MSE), 0.00035 for Active Power and 0.00057 for Reactive Power, suggesting strong generalization to the validation set. (Figure 3.9)

```
Designated epoch number reached. ANFIS training completed at epoch 100.

Minimal training RMSE = 0.000353475
  sugfis with properties:                                        (a)

                      Name: "fis"
                 AndMethod: "prod"
                  OrMethod: "max"
           ImplicationMethod: "prod"
           AggregationMethod: "sum"
       DefuzzificationMethod: "wtaver"
     DisableStructuralChecks: 0
                    Inputs: [1×1 fisvar]
                   Outputs: [1×1 fisvar]
                     Rules: [1×10 fisrule]

   See 'getTunableSettings' method for parameter optimization.

ANFIS training completed successfully.
Elapsed time is 458.022151 seconds.
```

```
Designated epoch number reached. ANFIS training completed at epoch 100.

Minimal training RMSE = 0.000570635
  sugfis with properties:                                        (b)

                      Name: "fis"
                 AndMethod: "prod"
                  OrMethod: "max"
           ImplicationMethod: "prod"
           AggregationMethod: "sum"
       DefuzzificationMethod: "wtaver"
     DisableStructuralChecks: 0
                    Inputs: [1×1 fisvar]
                   Outputs: [1×1 fisvar]
                     Rules: [1×10 fisrule]

   See 'getTunableSettings' method for parameter optimization.

ANFIS training completed successfully.
Elapsed time is 893.269784 seconds.
```

Figure 3.9: ANFIS controller training results : (a)- Active Power, (b)-Reactive Power.

To complete the analysis, we also present the regression plots for both active and reactive powers. The regression coefficients (R) indicate the goodness of fit of our regression model to the data. (Figure 3.10)

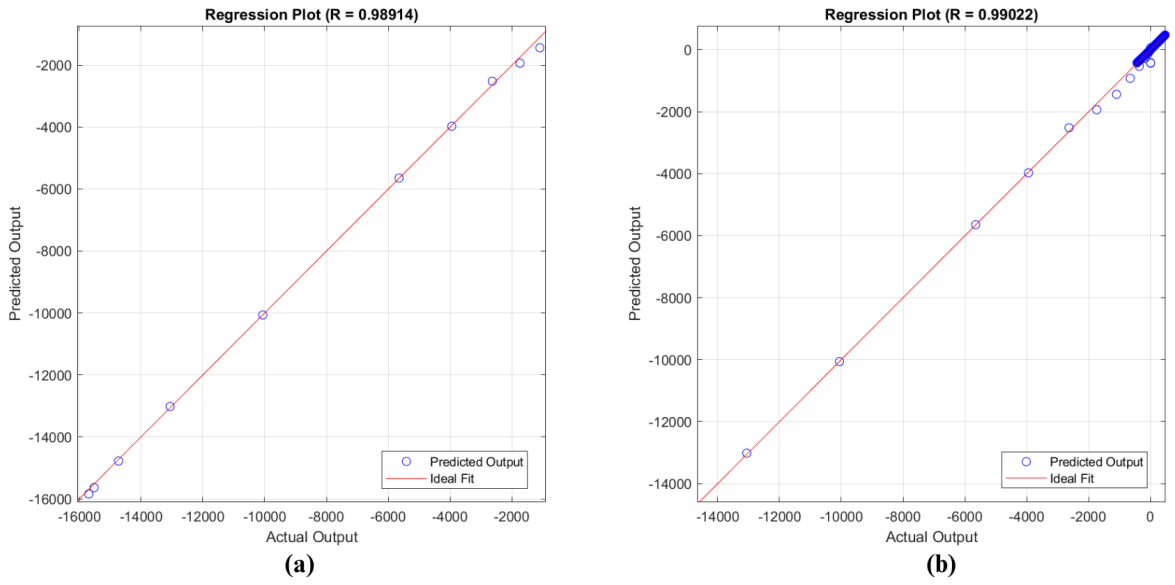

Figure 3.10: Regression plots: (a) Power, (b)-Reactive Power.

These values indicate the goodness of fit of our regression model to the data.

For Active Power, the training R value is 0.989, and for Reactive Power, it is 0.99. These high R values suggest that our regression model provides a solid fit for the training data, indicating the effectiveness of our ANFIS controller in capturing the relationship between the input and output variables.

To summarize, Figure (3.11) illustrates a flowchart of the general principles and steps in designing the ANFIS controller.

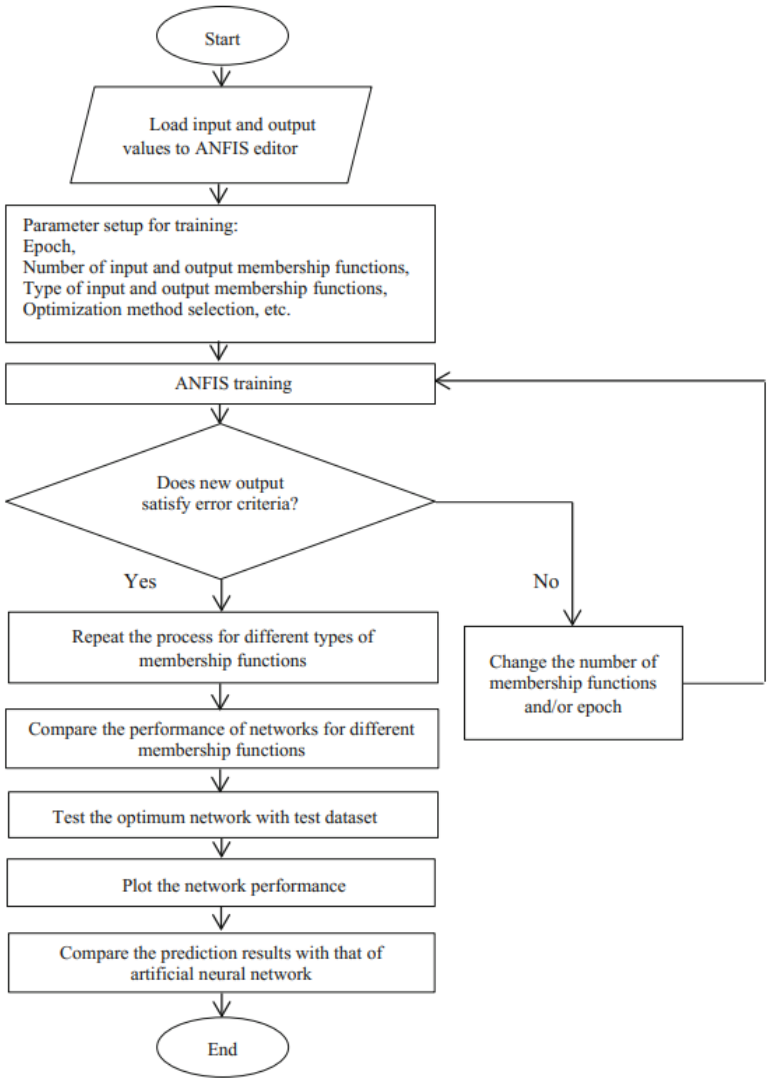

Figure 3.11: ANFIS Algorithm Flowchart.

3.8.4 Simulation results

The numerical simulation of our model was conducted in the MATLAB/SIMULINK environment, with the parameters of the DFIG detailed in Appendix A.
Below, in Figure (3.12), we present our Simulink implementation of active and reactive powers control of the DFIG using ANFIS controllers.

We use two neuro-fuzzy controllers, one for active power control and the other for reactive power control. Both networks have a single input, which is the difference between the reference power and the measured power. The outputs of the two networks represent the control applied to the DFIG, which is V_{qr} (for the first network) and V_{dr} for the second network.

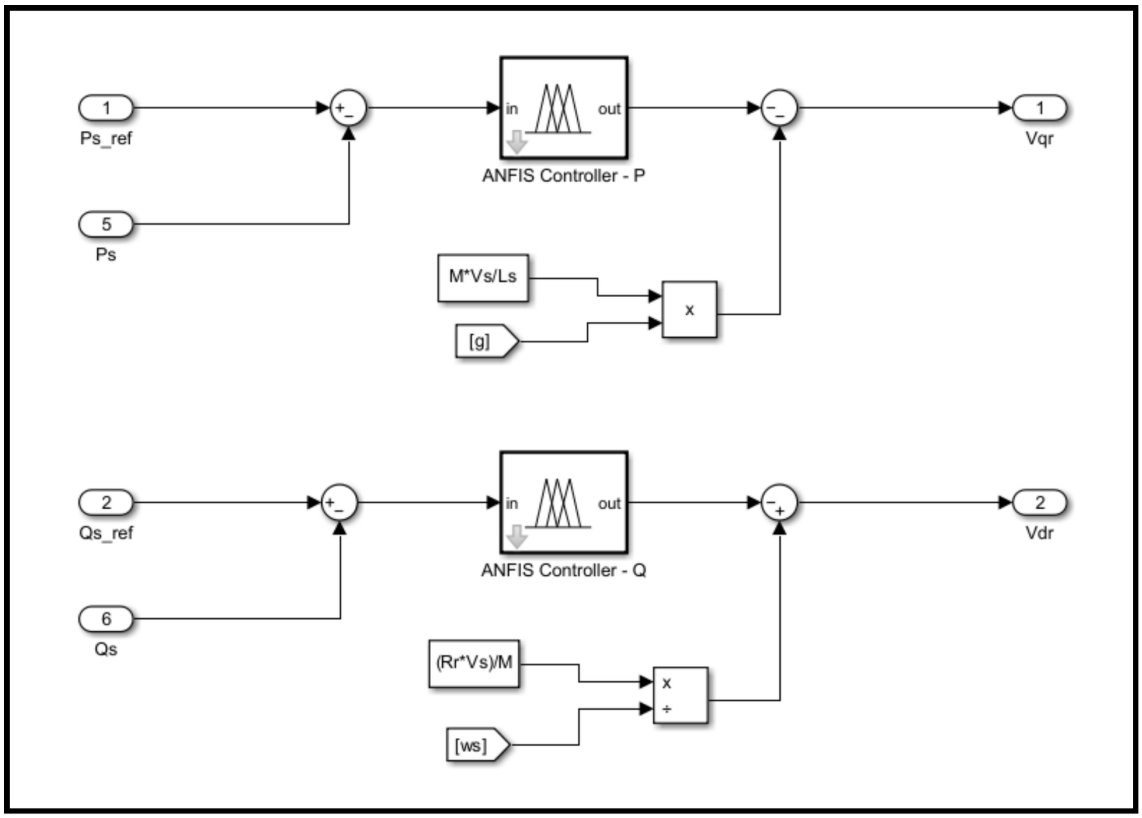

Figure 3.12: Simulink diagram of DFIG powers control using ANFIS.

This step aims to control the doubly-fed induction generator using neuro-fuzzy controllers. Various tests will be applied to demonstrate the performance of this control method.

3.8.4.1 Setpoint Tracking Test

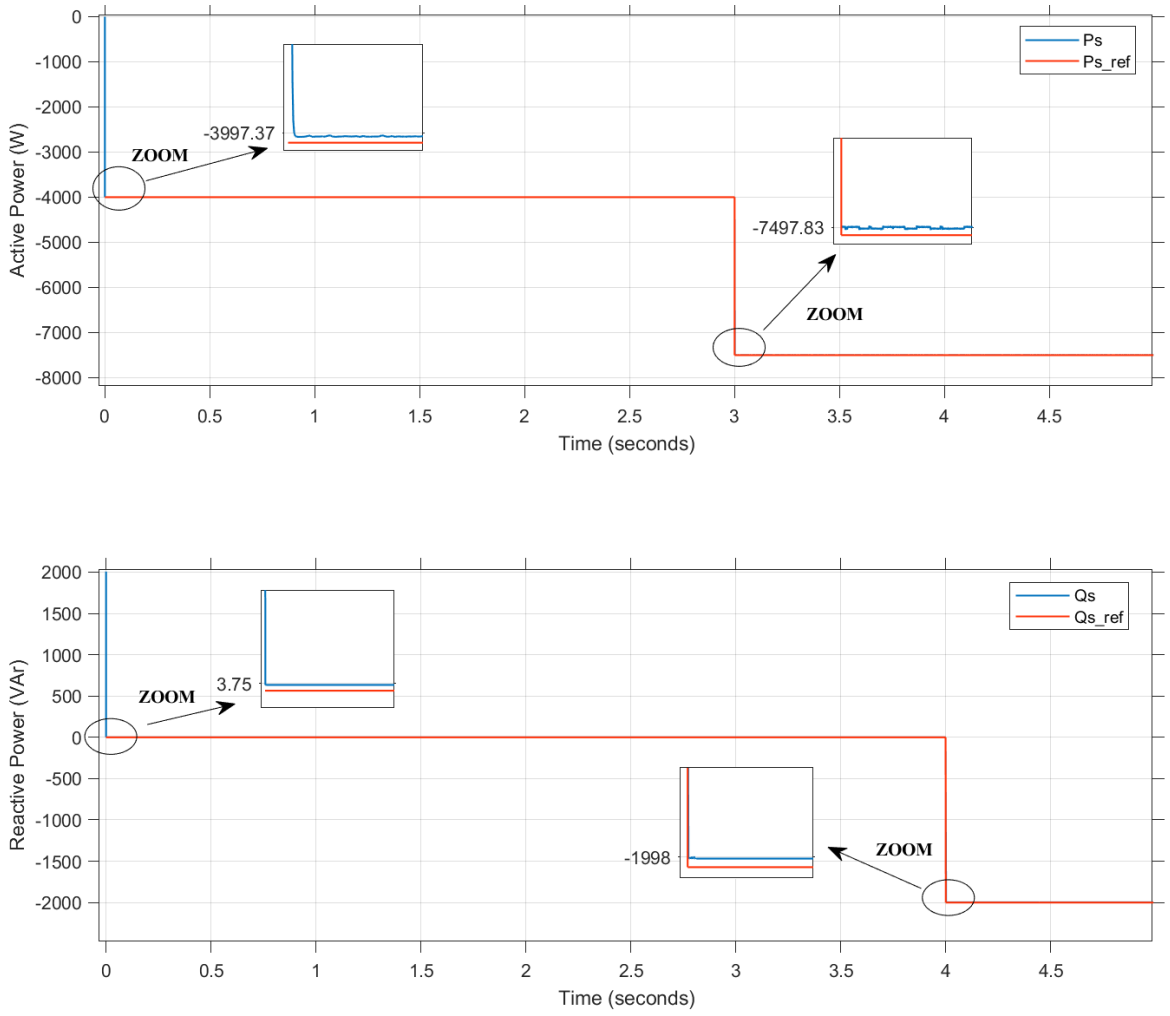

Figure 3.13: Active and Reactive powers direct control with ANFIS Controller.

Interpretation: Based on the curves in Figure (3.13), it is evident that the ANFIS controller achieves excellent decoupling between the active and reactive stator power components, maintaining both at their desired values. This controller demonstrates a small but significant improvement over the ANN controller introduced in the previous chapter. The enhancements include rapid response during transient states and setpoint changes, no overshoot, and a near-zero error between the setpoint values and the measured values.

3.8.4.2 Robustness Test

a) To evaluate the system's ability to reject disturbances and track reference signals, a disturbance was introduced at ($t = 2.5\,\mathrm{s}$). This disturbance corresponds to a sudden change in the rotational speed of the DFIG shaft, increasing from 140 rad/s to 185 rad/s. The objective was to verify if the system could compensate for this disturbance and maintain the reference value trajectory.

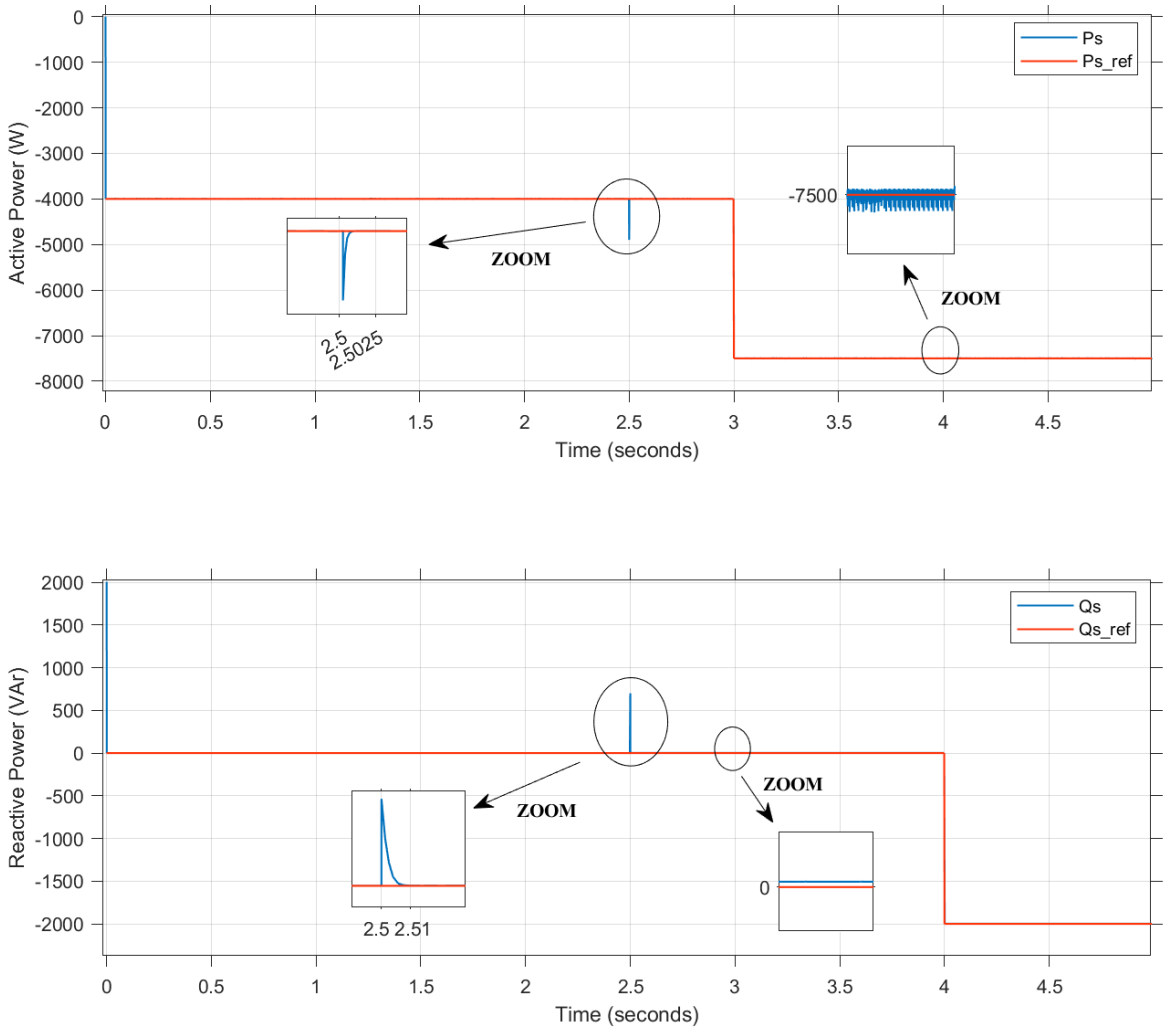

Figure 3.14: Active and Reactive powers after applying a disturbance (ANFIS).

Interpretation: The active and reactive powers exhibit perfect superposition with their reference signals without any additional overshoot. Notably, the decoupling between the two control axes (d and q) remains unaffected by the disturbance, highlighting the robustness of the controller and its convenience for direct vector control.

At (t = 2.5 , s), the disturbance impacts the stator powers. Still, both active and reactive powers swiftly realign with the reference signal trajectory within milliseconds, demonstrating the ANFIS controller's superior disturbance rejection capabilities.

b) To verify the robustness of the ANFIS controller against parameter variations, we conducted the same simulation for power control with a 100% variation in rotor resistance ($2\times R_r$).

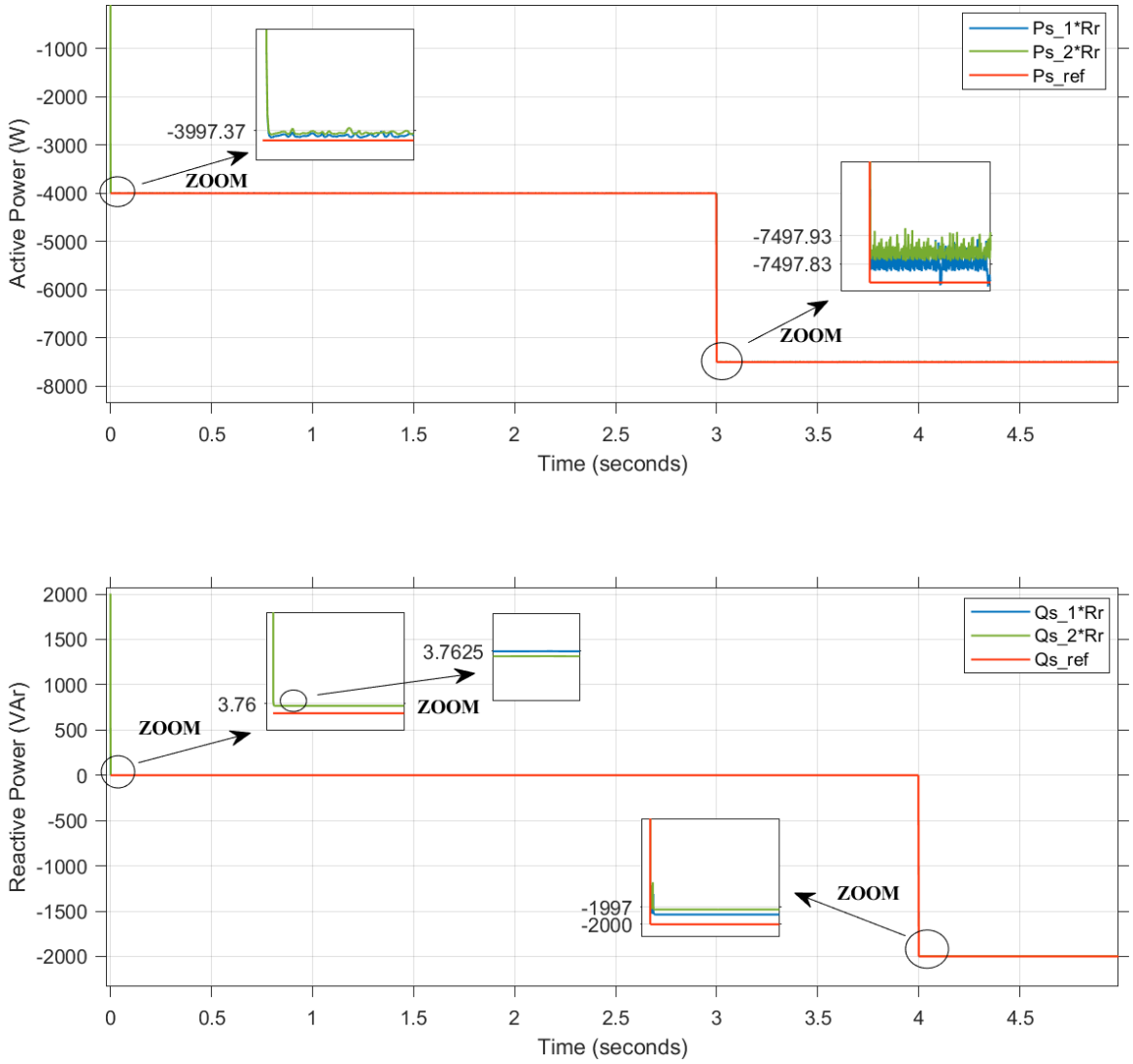

Figure 3.15: Active and Reactive powers after changing the parameter Rr (ANFIS).

Interpretation: The results in Figure (3.15) clearly indicate no undesirable effects on the dynamic responses during parameter variations of the DFIG. The plots for the original R_r and the variation $2{\times}R_r$ are almost identical. This observation underscores the robustness of the ANFIS controller in handling parametric variations. The controller effectively maintains system stability and performance despite changes in machine parameters. Moreover, we note that the decoupling between control variables also remains unaffected by these variations. Such robustness is essential for ensuring reliable and efficient operation in practical applications where parameter fluctuations are inevitable.

3.9 Comparative Study

3.9.1 Controllers Performance

In this section, we compare the performance of Artificial Neural Network (ANN) and Adaptive Neuro-Fuzzy Inference System (ANFIS) controllers in controlling the active and reactive powers of a Doubly Fed Induction Generator (DFIG). The comparison is based on several key metrics, including training speed, number of epochs required, regression coefficient (R), and Root Mean Square Error (RMSE).

Table (3.3) and Table (3.4), respectively, present the comparison of ANN and ANFIS controllers based on the metrics for active and reactive power control.

Table 3.3: Comparison of ANN and ANFIS Controllers for Active Power Control

Metric	**ANN**	**ANFIS**
Training Speed (s)	95	458
Number of Epochs	200	100
Regression Coefficient (R)	0.979	0.98
RMSE	0.0086	0.00035

Table 3.4: Comparison of ANN and ANFIS Controllers for Reactive Power Control

Metric	**ANN**	**ANFIS**
Training Speed (s)	142	893
Number of Epochs	480	100
Regression Coefficient (R)	0.982	0.99
RMSE	0.0027	0.00057

Discussion

For active power control, the ANN controller requires 200 epochs to converge, while the ANFIS controller requires only 100 epochs. This indicates that ANFIS is more efficient in terms of the learning process, converging faster than ANN. However, this efficiency comes at the cost of training speed, with ANN completing the training in 95 seconds compared to 458 seconds for ANFIS. This significant difference suggests that ANN is computationally more efficient per epoch.

Despite the faster training speed of ANN, the performance in terms of accuracy shows a nuanced picture. The regression coefficient for ANN is 0.979, slightly lower than the 0.98 achieved by ANFIS. Similarly, the RMSE for ANN is 0.0086, higher than the 0.00035

for ANFIS. These results indicate that while ANN trains faster, ANFIS provides slightly better accuracy and lower prediction error for active power control.

For reactive power control, the contrast between ANN and ANFIS becomes more pronounced. The ANN controller requires 480 epochs to converge, significantly more than the 100 epochs required by the ANFIS controller. In terms of training speed, ANN again demonstrates superior computational efficiency, completing training in 142 seconds compared to 893 seconds for ANFIS.

Accuracy metrics further underscore the differences: ANN achieves a regression coefficient of 0.982 and an RMSE of 0.0027, while ANFIS achieves a regression coefficient of 0.99 and an RMSE of 0.00057. This illustrates that ANFIS not only converges faster but also provides substantially better accuracy and lower error for reactive power control.

The choice between ANN and ANFIS controllers involves balancing training speed, convergence efficiency, and accuracy. ANN's faster training speed is advantageous in scenarios where computational resources or time are limited. This speed advantage, however, is offset by the need for more epochs and slightly lower accuracy compared to ANFIS. ANFIS, on the other hand, excels in accuracy and efficiency of convergence, making it better suited for applications where precision is paramount. However, its longer training times indicate a higher computational burden, which might be a limiting factor in real-time or resource-constrained environments.

3.9.2 Control Results

As shown in the previous chapter, we initially implemented an Artificial Neural Network (ANN) controller to enhance the performance of the direct vector control system. The ANN controller's learning capabilities allow it to effectively manage non-linearities, resulting in improved system performance in terms of regulation accuracy and response time.

To optimize the control strategy further, we subsequently implemented an Adaptive Neuro-Fuzzy Inference System (ANFIS) controller in this chapter. The ANFIS controller integrates the strengths of both neural networks and fuzzy logic, providing superior performance through adaptive learning and precise real-time adjustments.

The results obtained from the ANN and ANFIS controllers will be compared with those of the conventional PI controller to demonstrate their advantages better.

Figure (3.16) illustrates the comparison between the control results of active and reactive powers; it highlights a substantial decrease in the response time of active and reactive powers, as well as an absence or quasi-absence of overshoot and a nearly negligible error when employing the intelligent approaches.

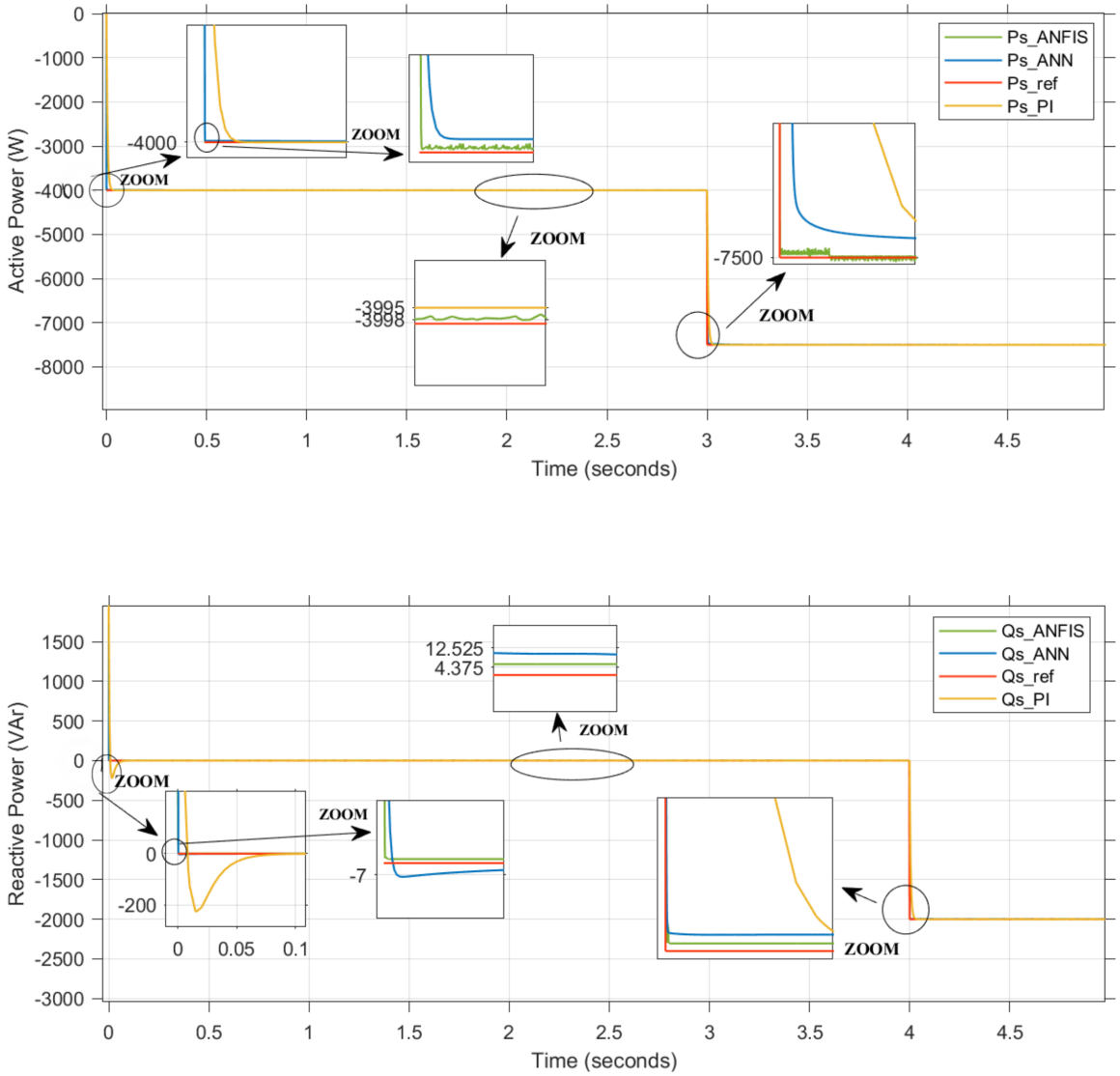

Figure 3.16: Comparison between PI, ANN, and ANFIS for Active and Reactive powers control.

Tables (3.5) and (3.6) present a comparison of the performance metrics for Classical PI, ANN, and ANFIS controllers in controlling the active and reactive powers, respectively.

Controller	Response Time	Overshoot	Steady-state Error
Classical PI	40 ms	2%	0.1%
ANN	10 ms	0%	0.09%
ANFIS	5 ms	0%	0.05%

Table 3.5: Control results with Classical PI, ANN, and ANFIS Controllers for Active Power

Controller	Response Time	Overshoot	Steady-state Error
Classical PI	60 ms	11%	0.125%
ANN	10 ms	0.35%	0.1%
ANFIS	3 ms	0%	0.04%

Table 3.6: Control results with Classical PI, ANN, and ANFIS Controllers for Reactive Power

Discussion

The comparative analysis of the control strategies for the DFIG, as illustrated in Tables (3.5) and (3.6), underscores the superior performance of the ANFIS controller in terms of response time, overshoot, and steady-state error for both active and reactive power control. The ANFIS controller demonstrates the fastest response times, with 5 ms for active power and 3 ms for reactive power, significantly outperforming both the ANN and PI controllers. This rapid response is crucial for adapting to dynamic changes and maintaining system stability.

Moreover, the ANFIS controller achieves zero overshoot in both active and reactive power control, a critical factor for preventing system stress and ensuring stable operation. In contrast, the PI controller shows significant overshoot, particularly in reactive power control (11%), which indicates a less precise handling of dynamic conditions.

In terms of steady-state error, the ANFIS controller again leads with the lowest values, highlighting its precision in maintaining desired outputs. While the ANN controller also shows improvements over the PI controller, it is still slightly less accurate than ANFIS, particularly in reactive power control.

Overall, the ANFIS controller's combination of neural network learning capabilities and fuzzy logic's adaptability allows it to effectively manage the complexities of DFIG control, outperforming both the classical PI and ANN controllers in various critical performance metrics.

3.10 Conclusion

In this chapter, we introduced an advanced intelligent vector control approach for the doubly fed induction generator (DFIG) using a neuro-fuzzy controller. Our designed control system demonstrated superior performance in tracking reference signals accurately. The results indicate that the ANFIS controller not only achieves highly satisfactory reference tracking but also significantly enhances system dynamics compared to the ANN controller.

Additionally, a comprehensive comparative study was conducted at the end of this chapter to evaluate the controllers' performance. This study compared the number of epochs necessary, training time, and regression coefficients between the ANN and ANFIS controllers. The results demonstrated that ANFIS requires fewer epochs and less training time while achieving higher regression coefficients, indicating better training efficiency and accuracy.

In terms of control results, the comparative analysis showed that both ANN and ANFIS controllers significantly outperformed the conventional PI controller regarding overshoot, response time, and steady-state error. Among the intelligent controllers, ANFIS demonstrated slightly better performance than ANN, particularly in minimizing overshoot and achieving faster settling times.

In conclusion, the proposed ANFIS controller offers a highly effective and robust control strategy for vector-controlled generators. Its ability to deliver high dynamic performance and adapt to varying parameters makes it a valuable asset for modern power systems, providing a viable and advanced alternative to traditional control methods.

General Conclusion

General Conclusion

This thesis aimed to develop and validate advanced control strategies for a doubly-fed induction generator (DFIG) used in wind energy conversion systems. The research encompassed the application of two intelligent control techniques to enhance the performance and reliability of DFIG-based wind turbines.

In Chapter 1, we provided an in-depth review of DFIG configurations, operating modes, and working principles, setting the stage for the control strategies explored in the subsequent chapters.

Chapter 2 focused on the implementation and analysis of artificial neural network (ANN) control for DFIGs. We detailed the structure and operation of ANN controllers, their training processes, and their application to DFIG control. The results indicated that ANN control offers good dynamic performance and robustness compared to known conventional methods.

Chapter 3 introduced the adaptive neuro-fuzzy inference system (ANFIS) control for DFIGs. This chapter detailed the ANFIS architecture, learning algorithms, and application to DFIG control. The simulation results demonstrated that ANFIS control outperforms the ANN controller, providing superior tracking accuracy and robustness under varying operational conditions.

Perspectives

The successful implementation of advanced control strategies for DFIGs in this study opens several avenues for future research and development:

- Using deep learning control strategies that can be more suitable for such large datasets typically available in wind energy systems.
- Exploration of hybrid control systems combining multiple intelligent techniques to leverage their complementary strengths.
- Experimental validation of the proposed control strategies using hardware implementations to bridge the gap between simulation and real-world application.
- Investigation of the integration of these advanced control techniques with emerging technologies such as IoT and smart grids to optimize overall system performance and reliability.

This research contributes to the field of wind energy by demonstrating the potential of advanced control strategies to improve the efficiency and reliability of DFIG-based wind turbines, paving the way for more robust and sustainable renewable energy solutions.

Appendices

Appendix A

Parameters of the DFIG and Wind Turbine

In this appendix, we provide a detailed listing of the parameters used in our study for the Doubly Fed Induction Generator (DFIG) and the attached wind turbine.

Parameter	**Symbol**	**Value**
DFIG Parameters		
Nominal Power	P_n	8.5 kW
Supply Voltage	V_s	220V/380V
Supply Frequency	f	50 Hz
Number of Pole Pairs	p	2
Stator Phase Resistance	R_s	0.455 Ω
Rotor Phase Resistance	R_r	0.62 Ω
Stator Phase Inductance	L_s	0.084 H
Rotor Phase Inductance	L_r	0.081 H
Mutual Inductance	M_{sr}	0.078 H
Rotor Inertia Moment	J_r	0.3125 Kg.m^2
Friction Coefficient	f_g	6.73×10^{-3} N.m.s^{-1}
Wind Turbine Parameters		
Nominal Power	P_n	10 kW
Number of Blades	P	3
Blade Diameter	R	3 m
Gearbox Ratio	G	5.4
Turbine Inertia Moment	$J_{turbine}$	3.1959 Kg.m^2
Viscosity Coefficient	$f_{turbine}$	0.0073 N.m.s^{-1}

Table A.1: Parameters of the DFIG and Wind Turbine

Appendix B

Matlab Scripts

```
% REACTIVE POWER ANN TRAINING (same for Active Power)

% Define the network architecture
hiddenLayerSizes = [10,25]; %  10 neurons in the first hidden layer, 25 in
    the second
net = feedforwardnet(hiddenLayerSizes, 'trainlm'); %Levenberg-Marquardt
    backpropagation

% Setting activation function for hidden layers
net.layers{1}.transferFcn = 'logsig';
net.layers{2}.transferFcn = 'tansig';

% Setting activation function for output layer
net.layers{3}.transferFcn = 'purelin';

% Setting up training parameters
net.trainParam.show = 1;          % result of error (mse) at each iteration
net.trainParam.epochs = 1000;     % maximum number of training epochs
net.trainParam.goal = 1e-12;      % Stopping criterion based on (mse) goal

% Dividing the data into training, validation, and test sets
net.divideParam.trainRatio = 70/100;  % 70% of data for training
net.divideParam.valRatio = 15/100;    % 15% of data for validation
net.divideParam.testRatio = 15/100;   % 15% of data for testing

% Training the network
[net, tr] = train(net, in_Q', out_Q');

% Exporting the trained network to Simulink
gensim(net, -1);
```

Listing B.1: MATLAB script for training an ANN for reactive power control.

```
1
2  % ACTIVE POWER ANFIS (same for Reactive power)
3  tic
4  % Extract input and output data
5  data_P = [in_P, out_P];
6  inP = data_P(:, 1);
7  outP = data_P(:, 2);
8
9  % Create a grid partitioning option
10 opt_P = genfisOptions('GridPartition');
11
12 % Setting the number of MFs
13 opt_P.NumMembershipFunctions = 10;
14
15 % Generate the initial FIS structure for active power
16 fis_P = genfis(inP, outP, opt_P);
17
18 % Train the ANFIS model for active power with more epochs
19 trainOptions = anfisOptions('InitialFIS', fis_P, 'EpochNumber', 100, '
       DisplayANFISInformation', 1, 'DisplayErrorValues', 1, 'DisplayStepSize',
        1, 'DisplayFinalResults', 1);
20
21 fis_P = anfis(data_P, trainOptions);
22
23 % Evaluate the ANFIS model for active power
24 anfisOut_P = evalfis(fis_P, inP);
25
26 % Calculate the correlation coefficient (R) between actual and predicted
       outputs
27 R_P = corrcoef(outP, anfisOut_P);
28 R_value_P = R_P(1,2);  % Extract the correlation coefficient
29
30 disp(fis_P);
31 disp('ANFIS training completed successfully.');
32 disp(['Correlation coefficient (R) for active power: ', num2str(R_value_P)
       ]);
33
34 % Plot the initial ANFIS results for active power
35 subplot(2,1,1);
36 plot(inP, outP, '*r', inP, anfisOut_P, '.b');
37 title('Default ANFIS P');
38 legend('Training Data', 'ANFIS Output');
39
40 toc
```

Listing B.2: MATLAB script for ANFIS training for Active power control.

Bibliography

[1] I. Yahyaoui. *Advances in Renewable Energies and Power Technologies*. Amsterdam: Elsevier, 2018.

[2] M. Ruviaro, F. Rüncos, and N. Sadowski. "Wound rotor doubly fed induction machine with radial rotary transformer". In: *Journal of Microwaves, Optoelectronics and Electromagnetic Applications* 12 (2013), pp. 411–426.

[3] M. Loucif. "Synthèse de lois de commande non-linéaires pour le contrôle d'une machine asynchrone à double alimentation dédiée à un système aérogénérateur". Université Aboubakr Belkaïd Tlemcen, 2016.

[4] A. Chemidi. "Analyse, modélisation et commande avancée d'une éolienne utilisée dans une ferme". Université de Tlemcen, 2015.

[5] S. El Aimani. "Modèlisation de différentes technologies d'éoliennes intégrées dans un réseau moyenne tension". Ecole centrale de Lille, 2004.

[6] D. S. Simonetti and F. D. C. Oliveira. "Chapter 15 - Doubly Fed Induction Generator in Wind Energy Conversion Systems". In: *Advances in Renewable Energies and Power Technologies*. Elsevier, 2018, pp. 461–490.

[7] C. Hamon, K. Elkington, and M. Ghandhari. "Doubly-fed induction generator modeling and control in DigSilent PowerFactory". Zhejiang, China, 2010.

[8] Lab-Volt. *Principles of Doubly-fed Induction Generators (DFIG): Instructor Guide*. Québec, 2011.

[9] Y. Djeriri. "Commande directe du couple et des puissances d'une MADA associée à un système éolien par les techniques de l'intelligence artificielle". Université Djillali Liabès de Sidi Bel-Abbès, 2015.

[10] F. Blaabjerg et al. *Advanced Control of Doubly Fed Induction Generator for Wind Power Systems*. Wiley, 2018.

[11] A. Ourici. "Double flux orientation control for a doubly fed induction machine". In: *International Journal of Electrical Power & Energy Systems* (2012), pp. 617–620.

[12] F. Boumaraf. "Commande D'un Aérogénérateur- Apport Des Techniques de L'intelligence Artificielle". Université de Batna 2, 2017.

[13] S. Abdeddaim and A. Betka. "Optimal tracking and robust power control of the DFIG wind turbine". In: *International Journal of Electrical Power & Energy Systems* (2013), pp. 234–242.

[14] A. Kerboua and M. Abid. "Hybrid fuzzy sliding mode control of a doubly-fed induction generator speed in wind turbines". In: *IEEE Transactions on Energy Conversion* (2015), pp. 1101–1110.

[15] A. Chemidi, H. Mohamed, and M. Bourouis. "A New Robust RST Controller Based on PSO Optimization for DFIG Wind Turbine". In: *European Journal of Electrical Engineering* (2022), pp. 13–20.

[16] I. Bendeddouche and W. T. Belharazem. "Etude, modélisation et commande robuste d'une MADA". Master thesis. HIGHER SCHOOL IN APPLIED SCIENCES TLEMCEN, 2023.

[17] M. Ayad. "Stabilisation d'un pendule inversé rotatif par approche neuronale". Master thesis. HIGHER SCHOOL IN APPLIED SCIENCES TLEMCEN, 2023.

[18] D. Graupe. *Principles of Artificial Neural Networks*. 3rd ed. Vol. 7. Advanced Series on Circuits and Systems. World Scientific, 2013.

[19] J. Sarangapani. *Neural Network Control of Nonlinear Discrete-Time Systems*. Boca Raton: CRC Press, 2017.

[20] F. Benmessaoud et al. "Multi-Level Direct Torque Control of Induction Motor using Fuzzy-Genetic Speed Regulation". In: *2019 International Conference on Power Generation Systems and Renewable Energy Technologies (PGSRET)*. 2019, pp. 1–5.

[21] C. L. Perera, M. Fernando, and N. Fernando. "Comparison of Multiple Linear Regression and Artificial Neural Network Models for the Prediction of Solid Waste Generation in Sri Lanka". In: 3 (2020), pp. 3–25.

[22] A. M. Toms. "Frequency Control of Microgrids using Intelligent Techniques – ANN, PSO and ANFIS". Rochester Institute of Technology, 2023.

[23] MathWorks Inc. *Neural Network Toolbox User's Guide*.

[24] GeeksforGeeks. *Difference Between Feed-Forward Neural Networks and Recurrent Neural Networks*. 2023. URL: https://www.geeksforgeeks.org/difference-between-feed-forward-neural-networks-and-recurrent-neural-networks (visited on 05/30/2024).

[25] C. Tan. "Artificial Neural Networks: A Financial Tool As Applied in the Australian Market". Bond University, Australia, 1997.

[26] I. Mohammad Jaber, H. A. R. Akkar, and H. R. Hatem. "OFDM Channel Estimation Based on Intelligent Systems". In: *Engineering and Technology Journal* (2014), pp. 305–324.

[27] B. M. Wilamowski and J. D. Irwin, eds. *Intelligent Systems*. 2nd ed. The Industrial Electronics Handbook. CRC Press, 2011.

[28] W. Suparta and K. M. Alhasa. *Modeling of Tropospheric Delays Using ANFIS*. SpringerBriefs in Meteorology. Springer International Publishing, 2016.

[29] M. Imran and S. A. Alsuhaibani. "A Neuro-Fuzzy Inference Model for Diabetic Retinopathy Classification". In: *Intelligent Data Analysis for Biomedical Applications*. Elsevier, 2019, pp. 147–172.

[30] N. K. Walia, H. Singh, and A. Sharma. "ANFIS: Adaptive Neuro-Fuzzy Inference System - A Survey". In: *International Journal of Computer Applications* 123 (2015), pp. 32–38.

[31] M. E. Boussag and A. Bakri. "Direct Power Control of three-phase PWM Rectifier based on Neuro-Fuzzy controller". University of M'sila, 2023.