**NEURO FUZZY BASED IoT ASSISTED POWER MONITORING SYSTEM FOR SMART GRID**

**PROJECT REPORT**

**Submitted by**

|  |  |
| --- | --- |
| **SABARI VIGNESH N** | **922318105039** |
| **SATHISH KUMAR R**  **DHARUN M**  **KARTHICK S** | **922318105043**  **922318105301**  **922318105501** |

***in partial fulfilment for the award of the* *degree***

***of***

**BACHELOR OF ENGINEERING**

**in**

**ELECTRICAL AND ELECTRONICS ENGINEERING**



**UNIVERSITY COLLEGE OF ENGINEERING-DINDIGUL**

**ANNA UNIVERSITY::CHENNAI 600025**

**JUNE 2022**

**ANNA UNIVERSITY::CHENNAI 600025**

**BONAFIDE CERTIFICATE**

Certified that this project report titled “**NEURO-FUZZY BASED IoT ASSISTED POWER MONITORING SYSTEM FOR SMART GRID”** is the bonafide work of **“DHARUN.M, KARTHICK.S, SATHISH KUMAR.R, and SABARI VIGNESH.N”** who carried out the project work under my supervision.

**SIGNATURE SIGNATURE**

**Dr. S.VASANTHI, M.E., Ph.D., Mrs. S. KIRUBADEVI M.E.,**

**HEAD OF THE DEPARTMENT,** S**UPERVISOR,**

**ASSISTANT PROFESSOR ASSISTANT PROFESSOR**

Department of EEE Department of EEE

University College of Engineering, University College of Engineering,

Dindigul-624 622 Dindigul-624 622

Submitted for the project and viva voice examination held on

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**ACKNOWLEDGEMENT**

The success and final outcome of this project required a lot of guidance and assistant from many people and we are extremely privileged to have all along the completion of our project. All that we have done is only due to the supervision and assistance and we would not forget to thank them. We would like to express our sincere and grateful thanks to our Dean, **Dr. S. SUTHA, M.E., Ph.D.,** of University College of Engineering Dindigul, for forwarding us to do our project and offering adequate support in completing our project.

We thank our respected Head of the Department and our gratitude to **Dr. S.VASANTHI, M.E., Ph.D.,** Electrical and Electronics Engineering for her valuable suggestion and encouragement in the completion of the project work.

With warm hearts and immense pleasure, we thank almighty for his grace and blessings, which drove us to the successful completion of the project work. With full heartedness we take this opportunity to express our sincere thanks to the respectable project guide and coordinator **Mrs.,** **S.KIRUBADEVI, M.E.**, Assistant Professor University College of Engineering Dindigul, Who is the guiding light for all activities in our college.

Finally, we would be honored to thank our parents and other staff members and friends of the department for their continued support behind this greater achievement during the course of our successful.

**ABSTRACT**

The Internet of Things (IoT) is commonly utilized for intelligent energy control, industrial automation, and a host of other applications. IoT sensors are installed in various stages of the smart grid (SG) to track and manage network statistics for safe and efficient power delivery. The challenges in the integration of IoT-SG must be overcome for the network to function efficiently. An IoT-based smart grid energy monitoring system depending on neuro-fuzzy is proposed in this project. At the core of the operator, a wireless sensor network (WSN) is employed to compute and transfer the necessary parameters for the prediction model. This project revolves around an IoT-based energy monitor, which can track and analyze electrical parameters, including current, voltage, active power, and load power consumption. Based on this data, consumers and electric power companies in the SG model can better control their usage to minimize billing costs. The results obtained show that the performance of hybridized solar/wind power plants will be improved with the help of ANFIS controller to a great extent.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO** | **TITLE** | **PAGE NO** |
|  | **ABSTRACT** | IV |
|  | **LIST OF FIGURES** | VII |
|  | **LIST OF TABLES** | IX |
|  | **LIST OF ABBREVIATION** | X |
| **1.** | **INTRODUCTION** | 1 |
|  | 1.1 EXISTING SYSTEM | 1 |
|  | 1.2 PROPOSED SYSTEM | 2 |
|  | 1.3 LITERATURE SURVEY | 3 |
|  | 1.4 OBJECTIVE OF THE WORK | 10 |
| **2.** | **METHODOLOGY** | 11 |
|  | 2.1 BLOCK DIAGRAM | 11 |
|  | 2.2 BLOCK DIAGRAM DESCRIPTION | 12 |
|  | 2.2.1 PV module | 12 |
|  | 2.2.2 Wind turbine | 15 |
|  | 2.2.3 Adaptive neuro fuzzy  inference system | 17 |
|  | 2.2.4 Internet of things | 20 |
|  | 2.2.5 Thingspeak | 22 |
|  | 2.3 SOFTWARE DESCRIPTION | 24 |
|  | 2.3.1 Matlab | 24 |
| **3.** | **SIMULATION RESULTS AND DISCUSSION** | 33 |
|  | 3.1 OVERALL IMPLEMENTATION | 33 |
|  | 3.2 IMPLEMENTATION OF MICRO GRID | 35 |
|  | 3.2.1 Implementation of solar PV  system | 36 |
|  | 3.2.2 Implementation of wind energy  conversion system | 38 |
|  | 3.3 IMPLEMENTATION OF MICRO GRID  WITH ANFIS CONTROLLER AND  BATTERY BACKUP | 41 |
|  | 3.3.1 Implementation of anfis and battery  Backup | 42 |
|  | 3.4 IMPLEMENTATION OF IOT  DEVICE | 50 |
|  | 3.5 SIMULATION RESULTS | 53 |
| **4.** | **CONCLUSION** | 58 |
|  | 4.1 FUTURE SCOPE | 58 |
|  | **REFERENCES** | 59 |

**LIST OF FIGURES**

**FIGURE TITLE PAGE NO**

**NO**

2.1 Block diagram 11

2.2 PV Array Block 13

2.3 I-V and P-V Characteristics Curve 14

2.4 ANFIS with 5 Layers for Fuzzy Model of the 18

Prediction Model

2.5 Internet of Things 20

2.6 ThingSpeak 23

3.1 Overall Implementation 33

3.2 Three-Phase Transformer Inductance 34

Matrix Type (Two Windings)

3.3 Three-Phase Breaker 35

3.4 Implementation of Micro Grid 35

3.5 Implementation of Solar PV System 36

3.6 PV Array 37

3.7 Implementation of Wind Energy Conversion System 38

3.8 Universal Bridge Converter 40

3.9 Implementation of Micro Grid with Anfis Controller and 41

Battery Backup

3.10 VSC Main Controller 43

3.11 Anfis Controller 44

3.12 Centroid Defuzzification Waveform 47

3.13 Subsystem for Battery Connection 50

3.14 ThingSpeak Output 50

3.15 ThingSpeak Configuration 51

3.16 Implementation of ThingSpeak to DC Voltage Measurement 51

3.17 Implementation of ThingSpeak to AC V-I Measurement 52

3.18 Micro Grid DC Output Voltage 53

3.19 Micro Grid DC Output Current 54

3.20 AC Output Voltage 55

3.21 AC Output Current 55

3.22 Battery Output 56

3.23 Data Monitoring Graph 56

3.24 Relationship Graph for AC Voltage and Current 57

**LIST OF TABLES**

**TABLE NO TITLE PAGE NO**

2.1 Formula Description 15

3.1 Reference Power Rule 48

**LIST OF ABBREVIATIONS**

IoT Internet of Things.

SG Smart Grid.

ANFIS Adaptive Neuro-Fuzzy Interface System.

WT Wind Turbine.

ICT Information and Communication Technology.

DES Distributed Energy Storage.

MG Micro Grid. .

PV Photo-Voltaic.

ANN Artificial Neural Network.

FLC Fuzzy Logic Controls

**CHAPTER 1**

**INTRODUCTION**

The Internet of Things (IoT), which can trade and share information, is a significant arrangement related to the communication system. Sensors, programming, and other devices are all part of the system. IoT provides protection and data processing. The IoT will connect objects and people from all over the world. Smart cities, rapid emergency services, acute stress, and smart buildings are examples of IoT applications. The internet is now a massive network capable of connecting various devices all over the world. Many sensors have been used for a long time to electrical power machines for domestic automation in recent days. Many sensors have been used in home automation to control electrical equipment for a long time. As a result of the use of large sensors, it is not cost- effective. The power consumption and cost of each device that needs a sensor will rise as the number of devices grows. Many sensors can be replaced by a small number of sensors in current IoT systems. IoT devices can be connected to a single network and consume energy and power.

* 1. **EXISTING SYSTEM**

In Existing system the smart grid consist of wind and solar power plant is controlled by fuzzy logic controller and WG/PV/FC/battery and WG/PV/FC/ultra-capacitor have been investigated during the last decades. The control of power flow in the system is not so much efficient. There is no absolute chance to meet the load requirement and there is no battery to store the excessive. The system mainly operates based on the reference value from load side and control the power flow from the smart grid to meet the load demand. There is no monitoring in the existing system.

**Disadvantages**

* Power loss
* Less predicting efficiency
* There is no power management control
  1. **PROPOSED SYSTEM**

In the proposed system neuro fuzzy logic based algorithm is implemented for power management and IoT sensors are used for monitoring solar and wind system with grid parallel power system. To control the power, an adaptive neuro-fuzzy inference system (ANFIS) is used in the smart grid and the battery backup is provided. The Anfis controller considers the input power from the renewable energy sources and the output power required by the consumers and then it allows power to meet the load demand, When the load demand is met it stores the excessive power in battery otherwise power is supplied from battery to grid to meet the load demand. The input values like irradiation, temperature and the output values like voltage and current value of the hybrid power plant, ac grid voltage and current values are passed to cloud platform through IoT devices, using this measured values monitoring of the system, data analysis to improve the system and predict the future development of the system can be done.

**Advantages**

* Power management
* Power flow control

**1.3 LITERATURE SURVEY**

FadiAl-Turjman and Mohammad Abujubbeh (2019)discussed about Power quality and reliability issues are big challenges to both service provider and consumers in conventional power grids. The ongoing technological advancements in the Internet of Things (IoT) era provide better solutions to enhance the management of these challenges and enforce the measures of a Smart Grid (SG). Advanced Metering Infrastructure (AMI) and Smart Metering (SM) technologies are enabler technologies that can modernize the conventional power grid through exposing the hidden details of electrical power by introducing two-way communication scheme during power transaction process between utilities and consumers. Throughout literature, AMI and SM technologies are widely discussed. However, few studies discuss the role of SM in power quality and reliability monitoring in IoT-enabled SGs. Hence, the paper aims to comprehensively review the feasibility of employing SM for power quality and reliability monitoring. First, we provide a detailed overview about the SMs, wireless communication technologies, and routing algorithms as enabling technologies in AMI. Then, we categorize the existing literature works that target power quality and reliability monitoring. Finally, open research issues are outlined based on shortages in the existing literature.

Kenneth Kimani and Vitalice Oduol (2019) discussed about The energy needs of the 21st century are growing rapidly due to the population growth and considerable efforts are being made to make the electricity grid more intelligent in order to make it more responsive to the energy needs of the consumers and to provide improved efficiency and reliability of power systems. Internet of Things (IoT) has emerged as one of the enabling technologies for a smart grid network. As the IoT connected devices continue to grow at a rapid pace, one of the major challenges is security since the devices are online hence making the smart grid vulnerable to significant attacks. Since an IoT based smart grid would consist of potentially millions of nodes, it has the largest attack surface for an IoT focused cyber-attack. A cyber-attack on a smart grid would have devastating effects on reliability of widespread infrastructure given the potential cascade effects of shutting down the electricity grid since most of the devices in our homes, offices, hospitals and trains require electricity to run. Once a single device is compromised, then the whole grid becomes vulnerable to cyber attacks. Such attacks on electricity supply can grind entire cities to a halt thereby causing huge financial and economic losses. This makes security a critical factor to consider before large scale deployment of IoT based smart grid networks. In this paper we review and explore the major challenges and security issues stunting the growth of IoT-based smart grid networks.

Muhammad Babar and Muhammad UsmanTariq (2020) discussed about the national security, economy, and healthcare heavily rely on the reliable distribution of electricity. The incorporation of communication technologies and sensors in the power structures, recognized as the smart grid which revolutionizes the model of the production, distribution, monitoring, and control of the electricity. To realize the applicability of smart grid, several issues need to be addressed. Securing the smart grid is a very challenging task and a pressing issue. In this article, a secure demand-side management (DSM) engine is proposed using machine learning (ML) for the Internet of Things (IoT)-enabled grid. The proposed DSM engine is responsible to preserve the efficient utilization of energy based on priorities. A specific resilient model is proposed to control intrusions in the smart grid. The resilient agent predicts the dishonest entities using the ML classifier. Advanced energy management and interface controlling agents are proposed to process energy information to optimize energy utilization. The efficient simulation is executed to test the efficiency of the proposed scheme. The analysis results reveal that the projected DSM engine is less vulnerable to the intrusion and effective enough to reduce the power utilization of the smart grid.

Zhuzhu Wang, Yang Liu and Zhuo Ma (2020) discussed about as the largest Internet-of-Things (IoT) deployment in the world, the smart grid implements extremely reduction in the energy dissipation for the operation of the smart city. However, the electricity data produced by the smart grid contain massive sensitive information, such as dispatching instructions and bills. The data are always revealed to cloud servers in the plaintext format for the Q -learning-based energy strategy making, which gives the chance for the adversary to abuse the user data. Therefore, in this article, we propose a lightweight privacy-preserving Q -learning framework (LiPSG) for the energy management strategy making of the smart grid. Before being sent to the control centre, the electricity data of each power supply region in LiPSG are first split into uniformly random secret shares. During completion of the computation task of Q -learning, the data are kept in the random share format all the time to avoid the data privacy disclosure. The computation feature is implemented by the newly proposed additive secret-sharing protocols. The edge computing technology is also deployed to further improve efficiency. Moreover, comprehensive theoretic analysis and experiments are given to prove the security and efficiency of LiPSG. Compared with the existing privacy-preserving schemes of the smart grid, LiPSG first provides a general Q-learning-based privacy-preserving power strategy making architecture with high efficiency and low-performance loss.

Vilmar Abreu and Altair OlivoSantin (2020) discussed about A smart grid (SG) is a complex system that comprises distributed servers and Internet-of-Things (IoT) devices. IoT devices are resource-constrained and are unable to cope with traditional communication and security protocols. In light of this limitation, this work proposes a novel method for end-to-end secure communication between the elements in the SG. Our proposal enables an authenticated user to transport her Internet credentials to the IoT context. We provide high efficiency in the message exchanges by adopting multicast communication without compromising the SG security. However, even though this process provides secure communication, it cannot enforce fine-grained access control over protected resources. Therefore, we propose a new two-step lightweight access control mechanism that leverages the established configuration to provide role-based authorization in the IoT context. The prototype evaluation shows that our proposal is more flexible demanding less manual configuration, while also requires only 23% of message exchanges compared to other approaches in the literature.

Muhammad Usman Ali Khan and ZeeshanShafiq (2020) discussed about There will be a dearth of electrical energy in the prospective world due to exponential increase in electrical energy demand of rapidly growing world population. With the development of internet-of-things (IoT), more smart devices will be integrated into residential buildings in smart cities that actively participate in electricity market via demand response (DR) programs to efficiently manage energy in order to meet this increasing energy demand. Thus, with this incitement, an energy management strategy using price-based DR program is developed for IoT-enabled residential buildings. We propose a wind-driven bacterial foraging algorithm (WBFA), which is a hybrid of wind-driven optimization (WDO) and bacterial foraging optimization (BFO) algorithms. Subsequently, we devised a strategy based on our proposed WBFA to systematically manage the power usage of IoT-enabled residential building smart appliances by scheduling to alleviate peak-to-average ratio (PAR), minimize cost of electricity, and maximize user comfort (UC). To endorse productiveness and effectiveness of the proposed WBFA-based strategy, substantial simulations are carried out. Furthermore, the proposed WBFA-based strategy is compared with benchmark strategies including binary particle swarm optimization (BPSO) algorithm, genetic algorithm (GA), genetic wind driven optimization (GWDO) algorithm, and genetic binary particle swarm optimization (GBPSO) algorithm in terms of energy consumption, cost of electricity, PAR, and UC. Simulation results show that the proposed WBFA-based strategy outperforms the benchmark strategies in terms of performance metrics.

PrakashPawar, PandurangaVittal K M. (2019) discussed about the day-to-day increased usage of power appliance by consumers is a growing concern in the energy sector, which creates an imbalance in the ratio of demand and supply. Demand-side energy management is an imperative tool to avoid significant deficiency from the supply end and improve energy efficiency. The trend in energy management lays focus on reducing the overall cost of electricity without limiting the consumption counterpart by instead choosing to reduce the power consumption during peak hours. The above issue seeks for design and development of a flexible and portable system to cover a wide variety of consumers for balancing the overall system. The design of smart energy management system is intended to replace the scenario of a complete power outage in a region with partial load shedding in a controlled manner as per the consumer’s preference. Demonstration of experimental work is carried out assuming demand response event and also, considering the maximum demand limit constraint with different cases and changing the order of priority assigned to an appliance. Cost optimization algorithms based on time of usage and user comfort level with sensory information features are embedded within SEMS. Reliable ZigBee communication for home area network is established and also, an IoT environment is developed for data storage and analytics.

YasirSaleem and Noel Crespi (2019)discussed about traditional power grids are being transformed into smart grids (SGs) to address the issues in the existing power system due to unidirectional information flow, energy wastage, growing energy demand, reliability, and security. SGs offer bi-directional energy flow between service providers and consumers, involving power generation, transmission, distribution, and utilization systems. SGs employ various devices for the monitoring, analysis, and control of the grid, deployed at power plants, distribution centres, and in consumers’ premises in a very large number. Hence, an SG requires connectivity, automation, and the tracking of such devices. This is achieved with the help of the Internet of Things (IoT). The IoT helps SG systems to support various network functions throughout the generation, transmission, distribution, and consumption of energy by incorporating the IoT devices (such as sensors, actuators, and smart meters), as well as by providing the connectivity, automation, and tracking for such devices. In this paper, we provide a comprehensive survey on the IoT-aided SG systems, which includes the existing architectures, applications, and prototypes of the IoT-aided SG systems. This survey also highlights the open issues, challenges, and future research directions for the IoT-aided SG systems.

Ahmed Ibrahim Saleh, AsmaahamdyRabie (2021) discussed about the smart electrical grid (SEG) that utilizes information for creating a widely distributed automated energy delivery network, is considered as an advanced digital 2-way power flow power system. Under different uncertainties, SEG is capable of self-healing, adaptive, resilient, and sustainable with foresight for prediction. Hence, SEG is considered as the next generation power grid. In this paper, a comprehensive survey on SEG as a new technology and operating models which will affect performance of distribution networks in the future are explored in detail. Most of the basic concepts affect such new technology like (Internet of Things (IoT), fog, cloud computing, and big data analysis) are discussed. A brief overview of IoT technologies is provided. It will explore the architectural structure of a typical IoT, cloud computing system, and different levels of the system. Furthermore, many classification methods and then electrical load forecasting (ELF) strategy that includes the pre-processing phase and the prediction phase have been discussed. Additionally, the different techniques used to manage big data generated by sensors and meters for application processing are explored. Feature selection and outlier rejection are discussed as a pre-processing process to filter the data, and then the load prediction process is explained. Finally, this paper covers the analysis of the load prediction phase in ELF strategy in which the prediction techniques will be reviewed.

Aihua Jiang, Huihong Yuan, Delong Li and JunyangTian (2019) discussed about ubiquitous Internet of Things refer to the interconnection and interaction of information at any time, any place, anyone, and anything. The ubiquitous power Internet of Things (UPIoT) refers to the application of ubiquitous IoT technology in power systems. Its implementation has the following advantages for power grids: connecting the devices that should have been connected; sharing the data that should have been shared and the value of data can be used more efficiently. Different from the smart grid, which is designed to build a multi energy integrated network and distributed management intelligent system with intelligent judgment and adaptive adjustment ability, the essence of UPIoT is to realize holistic perception and ubiquitous connection of energy. This paper introduces a technical architecture proposed by China State Grid Corporation, based on which the key communication and information technologies of UPIoT are analyzed in detail from the four elements of UPIoT. The challenges of implementing UPIoT are also introduced.

**1.4 OBJECTIVES OF THE WORK**

* To consider the hybrid power plant output and load requirement for fuzzy logic controller.
* To control the flow of power, load requirement and improve the power quality in the system using fuzzy logic rules and the power electronic devices embedded in the power system.
* To provide battery backup and add fuzzy logic controller to control the battery operations like charging and discharging.
* To use IoT sensor for reading the necessary parameters like voltage, current, real power, reactive power.
* To transmit parameters to analytics platform through IoT sensors for analysis and monitoring purpose.

**CHAPTER 2**

**METHODOLOGY**

**2.1 BLOCK DIAGRAM**

****

**Fig.2.1 Block diagram**

**2.2 BLOCK DIAGRAM DESCRIPTION**

The system consists of controlling and monitoring units. The controlling units are comprised of an ANFIS, which controls the power from both the solar-based power plant and wind power plant for consumer and communication with distributed generation in the smart grid. The monitoring system consists of a wireless sensor network, which is implemented in the overall framework.

**2.2.1 PV module**

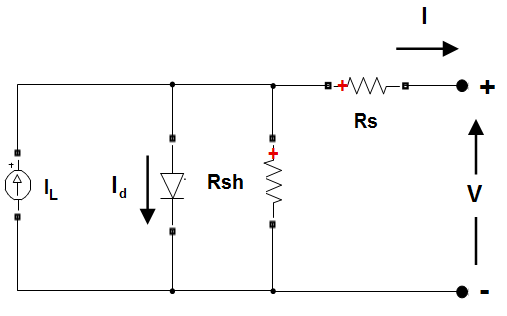
A single [solar cell](https://www.electrical4u.com/solar-cell/) cannot provide required useful output. So to increase output power level of a PV system, it is required to connect number of such **PV solar cells**. A solar module is normally series connected sufficient number of solar cells to provide required standard output [voltage](https://www.electrical4u.com/voltage-or-electric-potential-difference/) and power. One solar module can be rated from 3 watts to 300 watts. The solar modules or PV modules are commercially available basic building block of a solar electric power generation system.

Actually a single solar [PV cell](https://www.electrical4u.com/working-principle-of-photovoltaic-cell-or-solar-cell/) generates very tiny amount that is around 0.1 watt to 2 watts and the normal output voltage of a solar cell is approximately 0.5 V. But it is not practical to use such low power unit as building block of a system. So required number of such cells are combined together to form a practical commercially available solar unit which is known as **solar module or PV module**.

In a solar module the solar cells are connected in same fashion as the battery cell units in a battery bank system. That means positive terminals of one cell connected to negative terminal voltage of solar module is simple sum of the voltage of individual cells connected in series in the module.

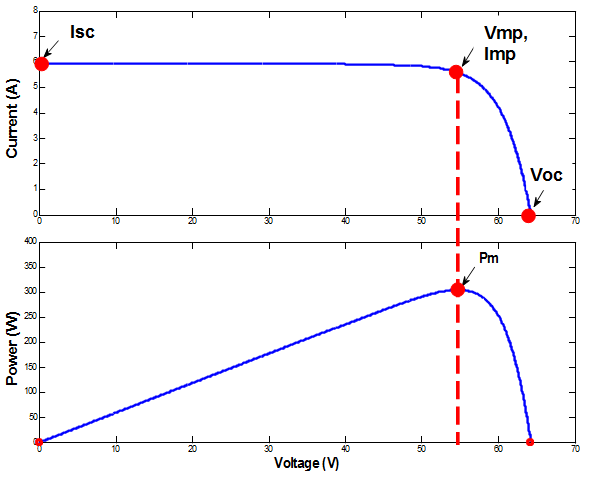
The PV Array block implements an array of photovoltaic (PV) modules. The array is built of strings of modules connected in parallel, each string consisting of modules connected in series. This block allows you to model preset PV modules from the National Renewable Energy Laboratory (NREL) System Advisor Model (2018) as well as PV modules that you define.

The PV Array block is a five-parameter model using a light-generated current source (IL), diode, series resistance (Rs), and shunt resistance (Rsh) to represent the irradiance- and temperature-dependent I-V characteristics of the modules.



**Fig.2.2 PV array block**

The below Figure 2.3 shows the I-V and P-V characteristics of PV array model



**Fig.2.3 I-V and P-V characteristics curve**

The diode I-V characteristics for a single module are defined by the equations

 (2.1)



(2.2)

Where

**Table.2.1 Formula description**

|  |  |
| --- | --- |
| I d | Diode current (A) |
| V d | Diode voltage (V) |
| I 0 | Diode saturation current (A) |
| Ni | Diode ideality factor, a number close to 1.0 |
| K | Boltzman constant = 1.3806e-23 J.K-1 |
| Q | Electron charge = 1.6022e-19 C |
| T | Cell temperature (K) |
| Ncell | Number of cells connected in series in a module |

**2.2.2 Wind turbine**

A wind turbine is a device that converts the kinetic energy of wind into electrical energy. Hundreds of thousands of large turbines, in installations known as wind farms, now generate over 650 giga watts of power, with 60 GW added each year. They are an increasingly important source of intermittent renewable energy, and are used in many countries to lower energy costs and reduce reliance on fossil fuels. One study claimed that, as of 2009, wind had the "lowest relative greenhouse gas emissions, the least water consumption demands and... the most favorable social impacts" compared to photovoltaic, hydro, geothermal, coal and gas energy sources. Smaller wind turbines are used for applications such as battery charging for auxiliary power for boats or caravans, and to power traffic warning signs. Larger turbines can contribute to a domestic power supply while selling unused power back to the utility supplier via the electrical grid. Wind turbines are manufactured in a wide range of sizes, with either horizontal or vertical axes.

Wind turbines work on a simple principle: instead of using electricity to make wind—like a fan—wind turbines use wind to make electricity. Wind turns the propeller-like blades of a turbine around a rotor, which spins a generator, which creates electricity.

Wind is a form of solar energy caused by a combination of three concurrent events. The sun unevenly heating the atmosphere Irregularities of the earths surface the rotation of the earth. Wind flow patterns and speeds vary greatly across the United States and are modified by bodies of water, vegetation, and differences in terrain. Humans use this wind flow, or motion energy, for many purposes: sailing, flying a kite, and even generating electricity. The terms "wind energy" and "wind power" both describe the process by which the wind is used to generate mechanical power or electricity. This mechanical power can be used for specific tasks (such as grinding grain or pumping water) or a generator can convert this mechanical power into electricity. A wind turbine turns wind energy into electricity using the aerodynamic force from the rotor blades, which work like an airplane wing or helicopter rotor blade. When wind flows across the blade, air pressure on one side of the blade decreases. The difference in air pressure across the two sides of the blade creates both lift and drag. The force of the lift is stronger than the drag and this causes the rotor to spin. The rotor connects to the generator, either directly (if it’s a direct drive turbine) or through a shaft and a series of gears (a gearbox) that speed up the rotation and allow for a physically smaller generator. This translation of aerodynamic force to rotation of a generator creates electricity.

## Types of wind turbines

The majority of wind turbines fall into two basic types:

#### Horizontal-Axis Turbines

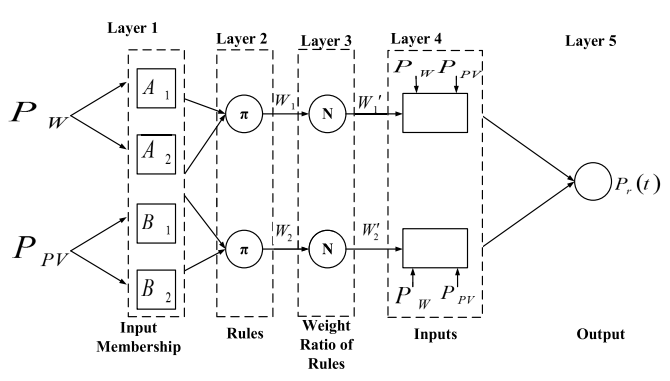
#### Vertical-Axis Turbines

Wind turbines can be built on land or offshore in large bodies of water like oceans and lakes. The U.S. Department of Energy is currently funding projects to facilitate offshore wind deployment in U.S. waters.

**2.2.3 Adaptive neuro-fuzzy inference system**

An adaptive neuro-fuzzy inference system or adaptive network-based fuzzy inference system (ANFIS) is a kind of artificial neural network that is based on Takagi–Sugeno fuzzy inference system. The technique was developed in the early 1990s. Since it integrates both neural networks and fuzzy logic principles, it has potential to capture the benefits of both in a single framework. Its inference system corresponds to a set of fuzzy IF–THEN rules that have learning capability to approximate nonlinear functions. Hence, ANFIS is considered to be a universal estimator. For using the ANFIS in a more efficient and optimal way, one can use the best parameters obtained by genetic algorithm. It has uses in intelligent situational aware energy management system.

It is possible to identify two parts in the network structure, namely premise and consequence parts. In more details, the architecture is composed by five layers. The first layer takes the input values and determines the [membership functions](https://en.wikipedia.org/wiki/Membership_function_(mathematics)) belonging to them. It is commonly called fuzzification layer. The membership degrees of each function are computed by using the premise parameter set, namely {a, b, c}. The second layer is responsible of generating the firing strengths for the rules. Due to its task, the second layer is denoted as "rule layer". The role of the third layer is to normalize the computed firing strengths, by dividing each value for the total firing strength. The fourth layer takes as input the normalized values and the consequence parameter set {p, q, r}. The values returned by this layer are the defuzzificated ones and those values are passed to the last layer to return the final output.



**Fig.2.4 Anfis with 5 layer for fuzzy model of the prediction model**

**Fuzzification layer**

The first layer of an ANFIS network describes the difference to a vanilla neural network. Neural networks in general are operating with a [data pre-processing](https://en.wikipedia.org/wiki/Data_pre-processing) step, in which the [features](https://en.wikipedia.org/wiki/Feature_(machine_learning)) are converted into normalized values between 0 and 1. An ANFIS neural network doesn't need a [sigmoid function](https://en.wikipedia.org/wiki/Sigmoid_function), but it's doing the pre-processing step by converting numeric values into fuzzy values.

Here is an example: Suppose, the network gets as input the distance between two points in the 2d space. The distance is measured in pixels and it can have values from 0 up to 500 pixels. Converting the numerical values into [Fuzzy numbers](https://en.wikipedia.org/wiki/Fuzzy_number) is done with the membership function which consists of [semantic descriptions](https://en.wikipedia.org/wiki/T-norm_fuzzy_logics) like near, middle and far. Each possible linguistic value is given by an individual [neuron](https://en.wikipedia.org/wiki/Artificial_neuron). The neuron “near” fires with a value from 0 until 1, if the distance is located within the category "near". While the neuron “middle” fires, if the distance in that category. The input value “distance in pixels” is split into three different neurons for near, middle and far.

**Product layer**

The input membership function’s logical ‘‘and’’ or product is completed by the product layer. The product layer output represents the next node’s input weight function.

**Normalization layer**

The normalized layer is the third layer, with each node representing a permanent node. It is capable of performing the fuzzy ‘‘and’’ process as well as efficiently normalizing the input weights.

**Defuzzification layer**

The role of this layer is to run an adaptive function that generates membership functions as output based on pre-defined fuzzy rules.

**Total output layer**

This is last and final layer where total output can be generated and is represented as f.

**2.2.4 Internet of things**



**Fig.2.5 Internet of things**

The Internet of things (IoT) describes physical objects (or groups of such objects) with sensors, processing ability, software, and other technologies that connect and exchange data with other devices and systems over the Internet or other communications networks.

Internet of things has been considered a misnomer because devices do not need to be connected to the public internet, they only need to be connected to a network and be individually addressable. The field has evolved due to the convergence of multiple technologies, including ubiquitous computing, commodity sensors, increasingly powerful embedded systems, and machine learning. Traditional fields of embedded systems, wireless sensor networks, control systems, automation (including home and building automation), independently and collectively enable the Internet of things. In the consumer market, IoT technology is most synonymous with products pertaining to the concept of the "smart home", including devices and appliances (such as lighting fixtures, thermostats, home security systems, cameras, and other home appliances) that support one or more common ecosystems, and can be controlled via devices associated with that ecosystem, such as smart phones and smart speakers.

IoT is also used in healthcare systems. There are a number of concerns about the risks in the growth of IoT technologies and products, especially in the areas of privacy and security, and consequently, industry and governmental moves to address these concerns have begun, including the development of international and local standards, guidelines, and regulatory frameworks.

In the simplest terms, the Internet of Things (IoT) is how we describe the digitally connected universe of everyday physical devices. These devices are embedded with internet connectivity, sensors and other hardware that allow communication and control via the web.

IoT makes once "dumb" devices "smarter" by giving them the ability to send data over the internet, allowing the device to communicate with people and other IoT-enabled things. The connected "smart home" is a good example of IoT in action. Internet-enabled thermostats, doorbells, smoke detectors and security alarms create a connected hub where data is shared between physical devices and users can remotely control the "things" in that hub (i.e., adjusting temperature settings, unlocking doors, etc.) via a mobile app or website. Far from being restricted to just the home, the Internet of Things can be found in an array of devices, industries and settings. From smart blackboards in school classrooms to medical devices that can detect signs of Parkinson's disease, IoT is rapidly making the world smarter by connecting the physical and the digital.

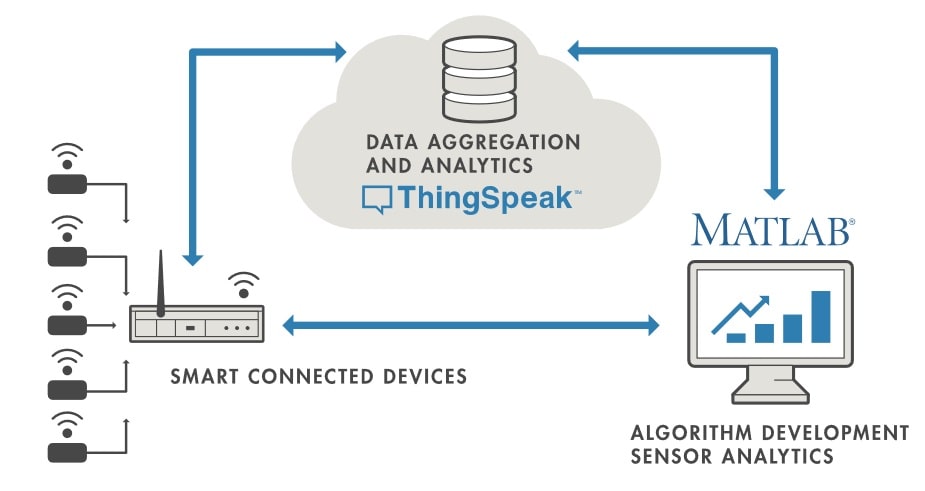
**2.2.5 ThingSpeak**

ThingSpeak is an IoT analytics platform service that allows you to aggregate, visualize and analyze live data streams in the cloud. ThingSpeak provides instant visualizations of data posted by your devices to ThingSpeak. With the ability to execute MATLAB® code in ThingSpeak you can perform online analysis and processing of the data as it comes in. ThingSpeak is often used for prototyping and proof of concept IoT systems that require analytics.

Internet of Things (IoT) describes an emerging trend where a large number of embedded devices (things) are connected to the Internet. These connected devices communicate with people and other things and often provide sensor data to cloud storage and cloud computing resources where the data is processed and analyzed to gain important insights. Cheap cloud computing power and increased device connectivity is enabling this trend.

IoT solutions are built for many vertical applications such as environmental monitoring and control, health monitoring, vehicle fleet monitoring industrial monitoring and control, and home automation.

At a high level, many IoT systems can be described using the figure 2.6



**Fig. 2.6 Thingspeak**

On the left, we have the smart devices (the "things" in IoT) that live at the edge of the network. These devices collect data and include things like wearable devices, wireless temperatures sensors, heart rate monitors, and hydraulic pressure sensors, and machines on the factory floor.

In the middle, we have the cloud where data from many sources is aggregated and analyzed in real time, often by an IoT analytics platform designed for this purpose.

The right side of the diagram depicts the algorithm development associated with the IoT application. Here an engineer or data scientist tries to gain insight into the collected data by performing historical analysis on the data. In this case, the data is pulled from the IoT platform into a desktop software environment to enable the engineer or scientist to prototype algorithms that may eventually execute in the cloud or on the smart device itself.

An IoT system includes all these elements. ThingSpeak fits in the cloud part of the diagram and provides a platform to quickly collect and analyze data from internet connected sensors.

**Thingspeak key features**

ThingSpeak allows you to aggregate, visualize and analyze live data streams in be cloud. Some of the key capabilities of ThingSpeak include the ability to:

* Easily configure devices to send data to ThingSpeak using popular IoT protocols.
* Visualize your sensor data in real-time.
* Aggregate data on-demand from third-party sources.
* Use the power of MATLAB to make sense of your IoT data.
* Run your IoT analytics automatically based on schedules or events.
* Prototype and build IoT systems without setting up servers or developing web software.
* Automatically act on your data and communicate using third-party services like Twilio® or Twitter®.

**2.3 SOFTWARE DESCRIPTION**

**2.3.1 Matlab**

**MATLAB** (matrix laboratory) is a multi-paradigm numerical computing environment and proprietary programming language developed by MathWorks. MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages.

Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine allowing access to symbolic computing abilities. An additional package, Simulink, adds graphical multi-domain simulation and model-based design for dynamic and embedded systems.

The MATLAB application is built around the MATLAB programming language. Common usage of the MATLAB application involves using the "Command Window" as an interactive mathematical shell or executing text files containing MATLAB code

* Interactive figure updates in the Live Editor, including title, labels, legend, and other annotations, as well as the ability to copy live script outputs to other applications​
* heatmap chart functions for visualizing data
* More functions for operating on tall arrays, including is member, sort, conv, and moving statistics functions​​​​​

**Matlab online**

* Use MATLAB through your web browser
* Ideal for teaching, learning, and convenient, lightweight access
* Available with most licenses

**Matlab’s power of computational mathematics**

MATLAB is used in every facet of computational mathematics. Following are some commonly used mathematical calculations where it is used most commonly:

* Dealing with Matrices and Arrays
* 2-D and 3-D Plotting and graphics
* Linear Algebra
* Algebraic Equations
* Non-linear Functions
* Statistics
* Data Analysis
* Calculus and Differential Equations
* Numerical Calculations
* Integration
* Transforms
* Curve Fitting
* Various other special functions

**The current directory and defined path**

It is necessary to declare a current directory before saving a file, loading a file, or running an M-file. By default, unless you edit the MATLAB shortcut, the current directory will be .../MATLAB/work. After you start MATLAB, change the current directory by either using the toolbar at the left-hand side of the screen, or entering the path in the bar at the top.

The current directory is the directory MATLAB will look in first for a function you try to call. Therefore if you have multiple folders and each of them has an M-file of the same name, there will not be a discrepancy if you set the current directory beforehand. The current directory is also the directory in which MATLAB will first look for a data file.

If you still want to call a function but it is not part of the current directory, you must define it using MATLAB's 'set path' utility. To access this utility, follow the path: file > set path... > add folder...

You could also go to "add folder with subfolders...", if you're adding an entire group, as you would if you were installing a toolbox. Then look for and select the folder you want. If you forget to do this and attempt to access a file that is not part of your defined path list, you will get an 'undefined function' error.

**Saving files**

There are many ways to save to files in MATLAB.

* save - saves data to files, \*.mat by default
* uisave - includes user interface
* hgsave - saves figures to files, \*.fig by default
* diary [filename] - saves all the text input in the command window to a text file.

**Loading files**

Likewise, there are many ways to load files into the workspace. One way is to use the "file" menu. To open a .m file click "open", whereas to import data from a data file select "import data..." and follow the wizard's instructions. He file must be in a recognized directory (usually your current directory, but at least one for which the path has been set).The data in the .mat file is stored with the same name as the variable originally had when it was saved. To get the name of this and all other environment variables, type "who".

To open an .m file, you can use file -> open, or type

**File naming constraints**

You can name files whatever you want (usually simpler is better though), with a few exceptions:

* MATLAB for Windows retains the file naming constraints set by DOS. The following characters cannot be used in filenames: (” \* \ |?)
* You're not allowed to use the name of a reserved word as the name of a file. For example, while.m is not a valid file name because while is one of MATLAB's reserved words.
* When you declare an m-file function, the m-file must be the same name as the function or MATLAB will not be able to run it.
* You must save it as "factorial.m" in order to use it. MATLAB will name it for you if you save it after typing the function declaration, but if you change the name of the function you must change the name of the file manually, and vice versa.

**Variables**

Variables are defined using the assignment operator, =. MATLAB is a weakly typed programming language because types are implicitly converted. It is an inferred typed language because variables can be assigned without declaring their type, except if they are to be treated as symbolic objects, and that their type can change. Values can come from constants, from computation involving values of other variables, or from the output of a function

**Structures**

MATLAB supports structure data types. Since all variables in MATLAB are arrays, a more adequate name is "structure array", where each element of the array has the same field names. In addition, MATLAB supports dynamic field names (field look-ups by name, field manipulations, etc.).

**Functions**

When creating a MATLAB function, the name of the file should match the name of the first function in the file. Valid function names begin with an alphabetic character, and can contain letters, numbers, or underscores. Variables and functions are case sensitive.

**Function handles**

MATLAB supports elements of [lambda calculus](https://en.wikipedia.org/wiki/Lambda_calculus) by introducing function handles, or function references, which are implemented either in .m files or anonymous/nested functions.

**Classes and object-oriented programming**

MATLAB supports object-oriented programming including classes, inheritance, virtual dispatch, packages, pass-by-value semantics, and pass-by-reference semantics. However, the syntax and calling conventions are significantly different from other languages. MATLAB has value classes and reference classes, depending on whether the class has handle as a super-class (for reference classes) or not (for value classes). Method call behaviour is different between value and reference classes.

**Interfacing with other languages**

MATLAB can call functions and subroutines written in the programming languages C or Fortran. A wrapper function is created allowing MATLAB data types to be passed and returned. MEX files (MATLAB executables) are the dynamically loadable object files created by compiling such functions. Since 2014 increasing two-way interfacing with Python was being added.

Libraries written in Perl, Java, ActiveX or .NET can be directly called from MATLAB, and many MATLAB libraries (for example XML or SQL support) are implemented as wrappers around Java or ActiveX libraries. Calling MATLAB from Java is more complicated, but can be done with a MATLAB toolbox which is sold separately by MathWorks, or using an undocumented mechanism called JMI (Java-to-MATLAB Interface), (which should not be confused with the unrelated Java Metadata Interface that is also called JMI). Official MATLAB API for Java was added in 2016. As alternatives to the MuPAD based Symbolic Math Toolbox available from MathWorks, MATLAB can be connected to Maple or Mathematica. Libraries also exist to import and export MathML.

**Image processing toolbox**

Image Acquisition Toolbox is a very valuable collection of functions that handles receiving image and video signal directly from computer to the Matlab environment. This toolbox recognizes video cameras from multiple hardware vendors. Specially designed interface leads through possible transformations of images and videos, acquired thanks to mechanisms of Image Acquisition Toolbox.

Image Processing Toolbox is a wide set of functions and algorithms that deal with graphics. It supports almost any type of image file. It gives the user unlimited options 17 for pre- and post- processing of pictures. There are functions responsible for image enhancement, de blurring, filtering, noise reduction, spatial transformations, creating histograms, changing the threshold, hue and saturation, also for adjustment of colour balance, contrast, and detection of objects and analysis of shapes.

**Econometrics toolbox**

Bayesian linear regression model for analyzing the relationship between a response and a set of predictor variables

Vector autoregressive model for analyzing multivariate time series data including exogenous predictors

**MATLAB production server**

Web-based server management dashboard for IT configuration and control

**Neural network toolbox**

Deep learning algorithms for training convolutional neural networks (CNNs) for regression tasks using multiple GPUs on PCs, on clusters, and in the cloud

Deep learning visualization for the features a CNN model has learned using image optimization

Functions for transferring weights from pre trained CNN models (AlexNet, VGG-16, and VGG-19) and models from Caffe Model Zoo

**Statistics and machine learning toolbox**

Regression Learner app for training regression models using supervised machine learning tall array algorithms for support vector machine (SVM) and Naïve Bayes classification, bagged decision trees, and lasso regression

**Computer vision system toolbox**

Deep learning for detecting objects using Fast R-CNN and Faster R-CNN

**Features of matlab**

**Following are the basic features of matlab:**

* It is a high-level language for numerical computation, visualization and application development.
* It also provides an interactive environment for iterative exploration, design and problem solving.
* It provides vast library of mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, numerical integration and solving ordinary differential equations.
* It provides built-in graphics for visualizing data and tools for creating custom plots.
* MATLAB's programming interface gives development tools for improving code quality and maintainability and maximizing performance.
* It provides tools for building applications with custom graphical interfaces.
* It provides functions for integrating MATLAB based algorithms with external applications and languages such as C, Java, .NET and Microsoft Excel.

**Uses of matlab**

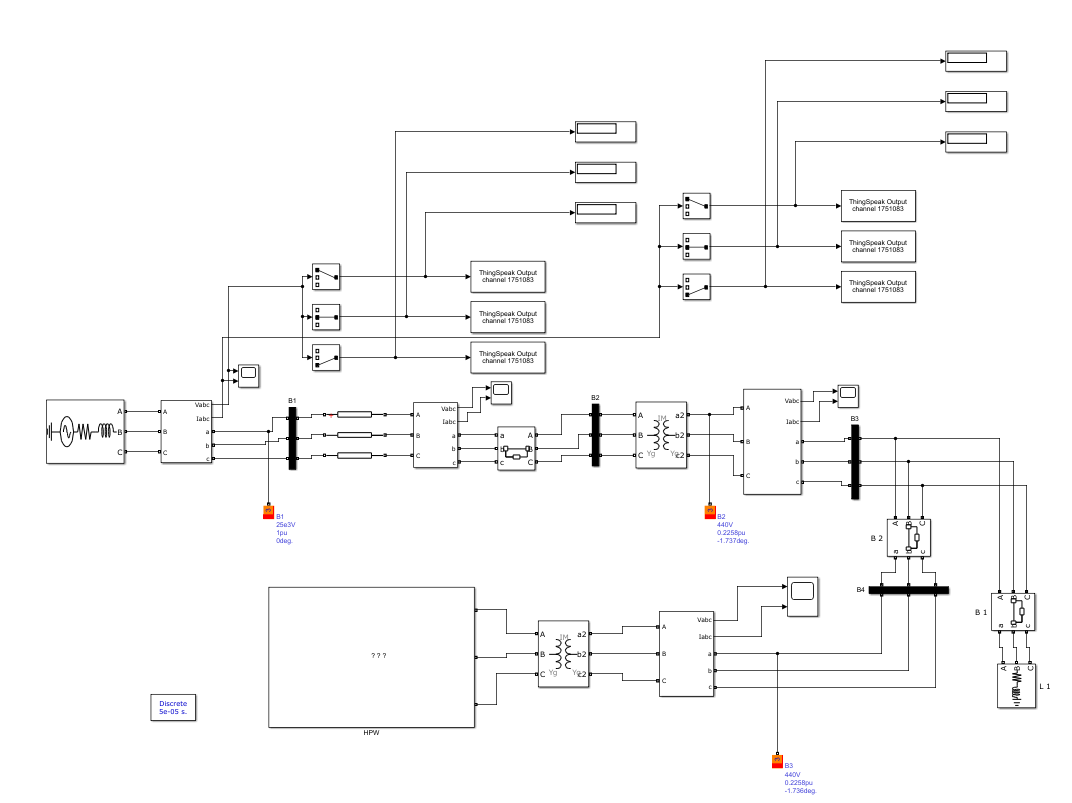
MATLAB is widely used as a computational tool in science and engineering encompassing the fields of physics, chemistry, math and all engineering streams. It is used in a range of applications including:

* Signal Processing and Communications
* Image and Video Processing
* Control Systems
* Test and Measurement
* Computational Finance
* Computational Biology

**CHAPTER 3**

**SIMULATION RESULT AND DISCUSSION**

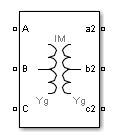
**3.1 OVERALL IMPLEMENTATION**

****

**Fig.3.1 Overall implementation**

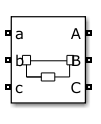
The Figure 3.1 shows the overall implementation of the project. In HPW subsystem the implementation of micro grid consist of solar and wind power plant and neuro fuzzy logic controller to control the power flow and battery with fuzzy logic controller to provide backup take places and the controlled AC power from subsystem is passed to AC grid consist of load, 3 phase transformer and 3 phase circuit breaker.

Here the transformer is a Three-Phase Transformer Inductance Matrix Type (Two Windings) used to step up or step down voltage without changing its frequency and the transformer core type is Three-limb or five-limb. When "Three-limb or five-limb" core type is specified, the transformer is modelled by 6 coupled windings. The winding connections are set as default Yg.



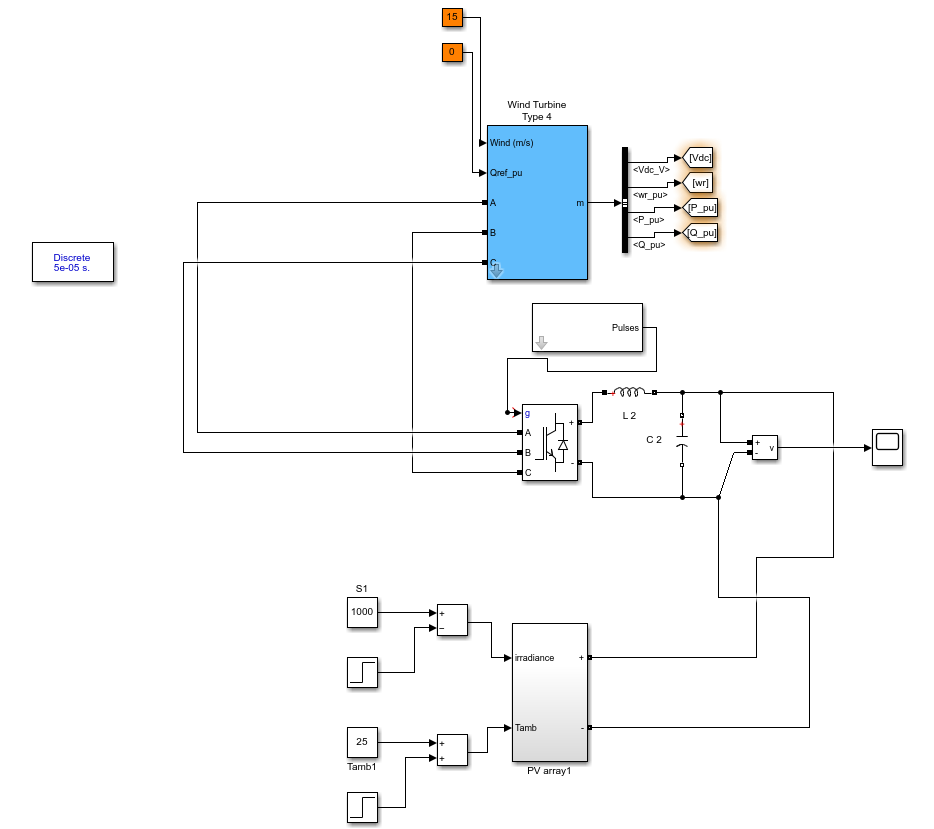
**Fig.3.2 Three-phase transformer inductance matrix type (two windings)**

The Three-Phase Breaker block implements a three-phase circuit breaker where the opening and closing times can be controlled either from an external Simulink® signal (external control mode), or from an internal control timer (internal control mode).The Three-Phase Breaker block uses three Breaker blocks connected between the inputs and the outputs of the block. You can use this block in series with the three-phase element you want to switch. The arc extinction process of the Three-Phase Fault block is the same as for the Breaker block. If the Three-Phase Breaker block is set in external control mode, a control input appears in the block icon. The control signal connected to the Simulink input must be either 0, which opens the breakers, or any positive value, which closes the breakers.



**Fig.3.3 Three-phase breaker**

**3.2 IMPLEMENTATION OF MICRO GRID**

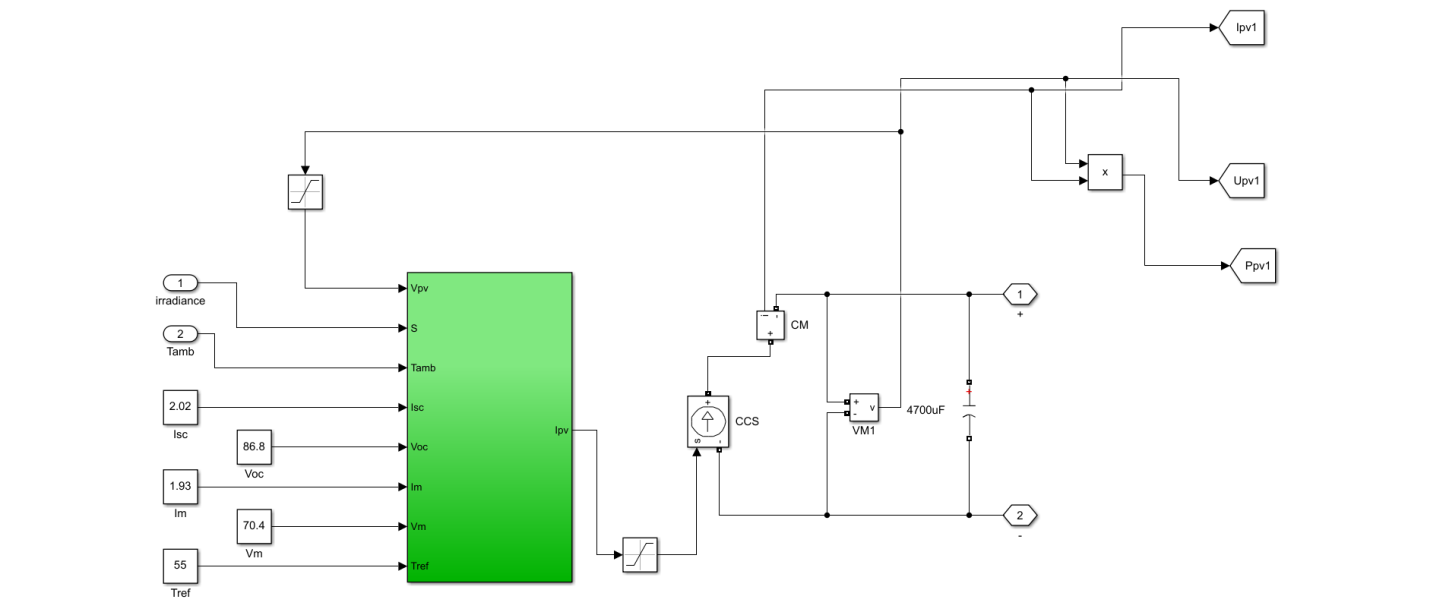
****

**Fig.3.4 Implementation of micro grid**

The Figure 3.4 shows implementation micro grid to produce electricity micro grid is nothing but group of electric resources are installed to produce electricity. There are two renewable resources present in the micro gird. They are

* Solar PV system
* Wind energy conversion system

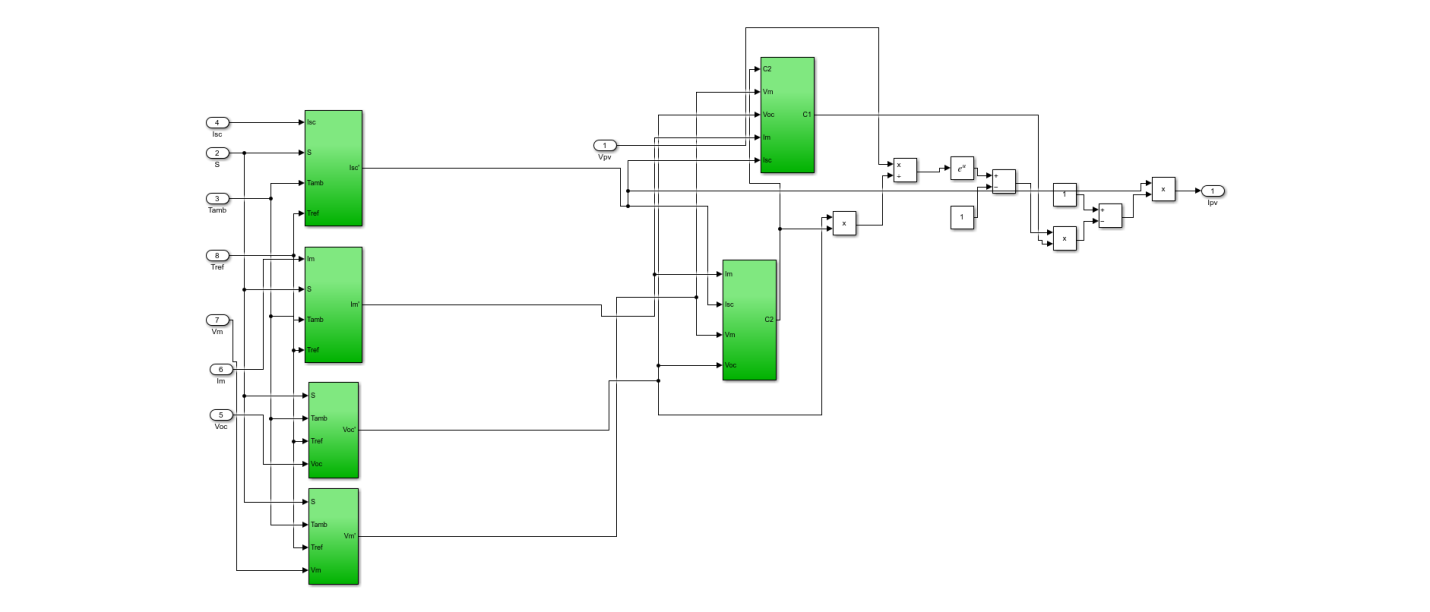
**3.2.1 Implementation of solar PV system**



**Fig.3.5 Implementation of solar PV system**

The Figure 3.5 shows the implementation of solar panel and the solar panel takes irradiation and temperature as input and produce corresponding output voltage and current.

The solar plant subsystem models a solar plant that contains parallel-connected strings of solar panels. The solar panel is modelled using the Solar Cell block from the Simscape™ Electrical™ library. Given the specified DC bus voltage, solar cell characteristics and specified power rating, a calculation is made of the solar panel string length and number of parallel-connected strings. Connecting multiple panels can slow the simulation because it increases the number of elements in a model. By assuming uniform irradiance and temperature across all the solar panels, it is possible to reduce the number of solar elements by using the controlled current and voltage sources as shown in the above solar panel subsystem.

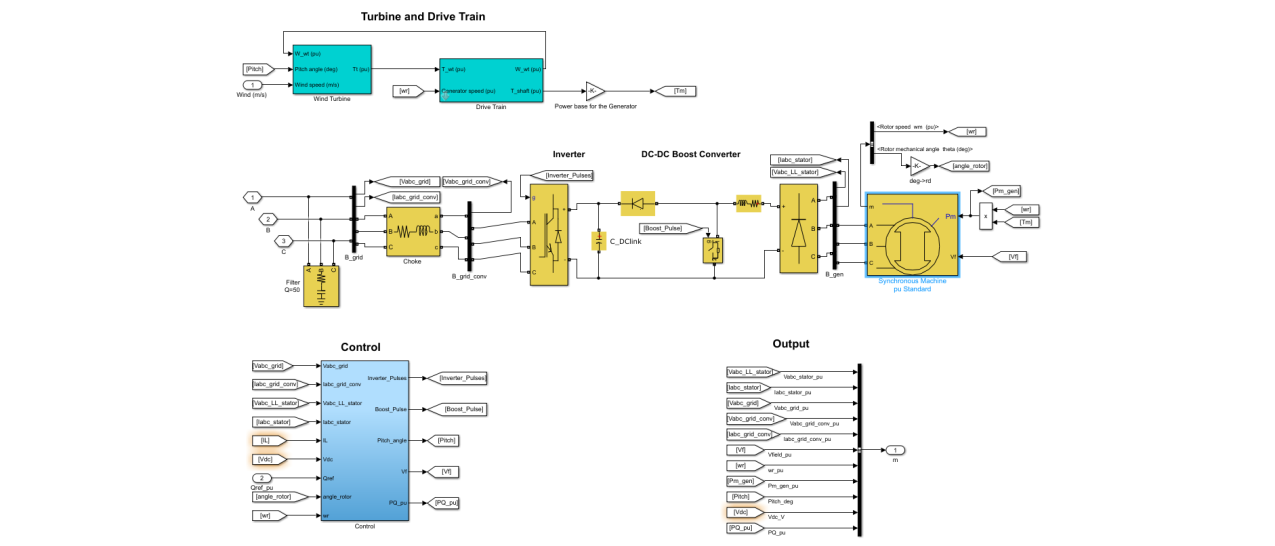


**Fig.3.6 PV array**

The Figure 3.6 shows that numbers of panels are connected in parallel and series with each other to generate respective voltage and current.

The output of the solar panel array is variable DC in nature so we use capacitor of particular rating shown in Fig.3.5 to produce constant DC voltage and current.

**3.2.2 Implementation of wind energy conversion system**

****

**Fig.3.7 Implementation of wind power plant**

The Figure 3.7 shows the implementation of wind power plant. The WT model is focused on wind speed versus power characteristics of WT output. The wind turbines output power is defined by

 (3.1)

Where, the wind turbines mechanical output power can be represented as PW. The performance coefficient of the turbine is CP; υ is the rotor blades tip speed ratio. ϕ and ρ are the pitch and of blade and air density respectively. The turbine swept area and wind speed can be represented as A and VWIND respectively. The performance coefficient model can be represented by.

 (3.2)

The parameters that depend on WT’s blade design and rotor structure can be represented as C1 to C6. The parameter υi can be expressed by.

 (3.3)

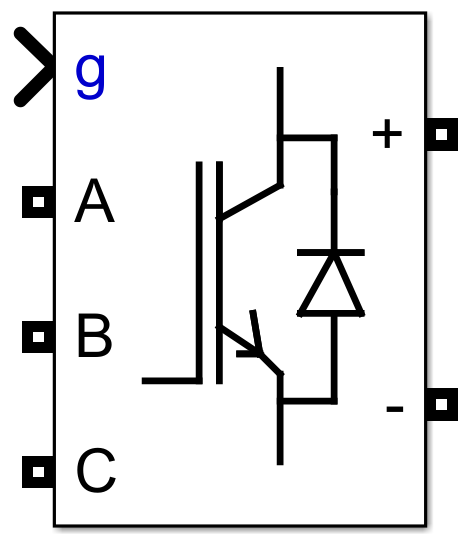
Furthermore, for unique values of ρ and A, can be simplified and normalized, as in.

 (3.4)

where, PW(p.u.) is the power in per unit (p.u) of nominal power for different values of A and ρ, CP(p.u.) denotes the p.u. value of output coefficient CP, kp defines power gain, and VWIND(p.u.) defines the p.u. value of the base wind speed. The average predicted wind speed in metres per second (m/s) is known to be base wind speed.

The inputs are the generator and wind speed, and the torque given to the generator shaft is the output. The generator’s torque is determined by the generator’s power and rpm. The WT drives the rotor shaft and generates mechanical torque based on the values of generator and wind speed.

The power generated from wind power plant is AC in nature so we use universal bridge converter to convert AC to variable DC here same as solar power plant use capacitor of particular rating to produce constant DC voltage and current.



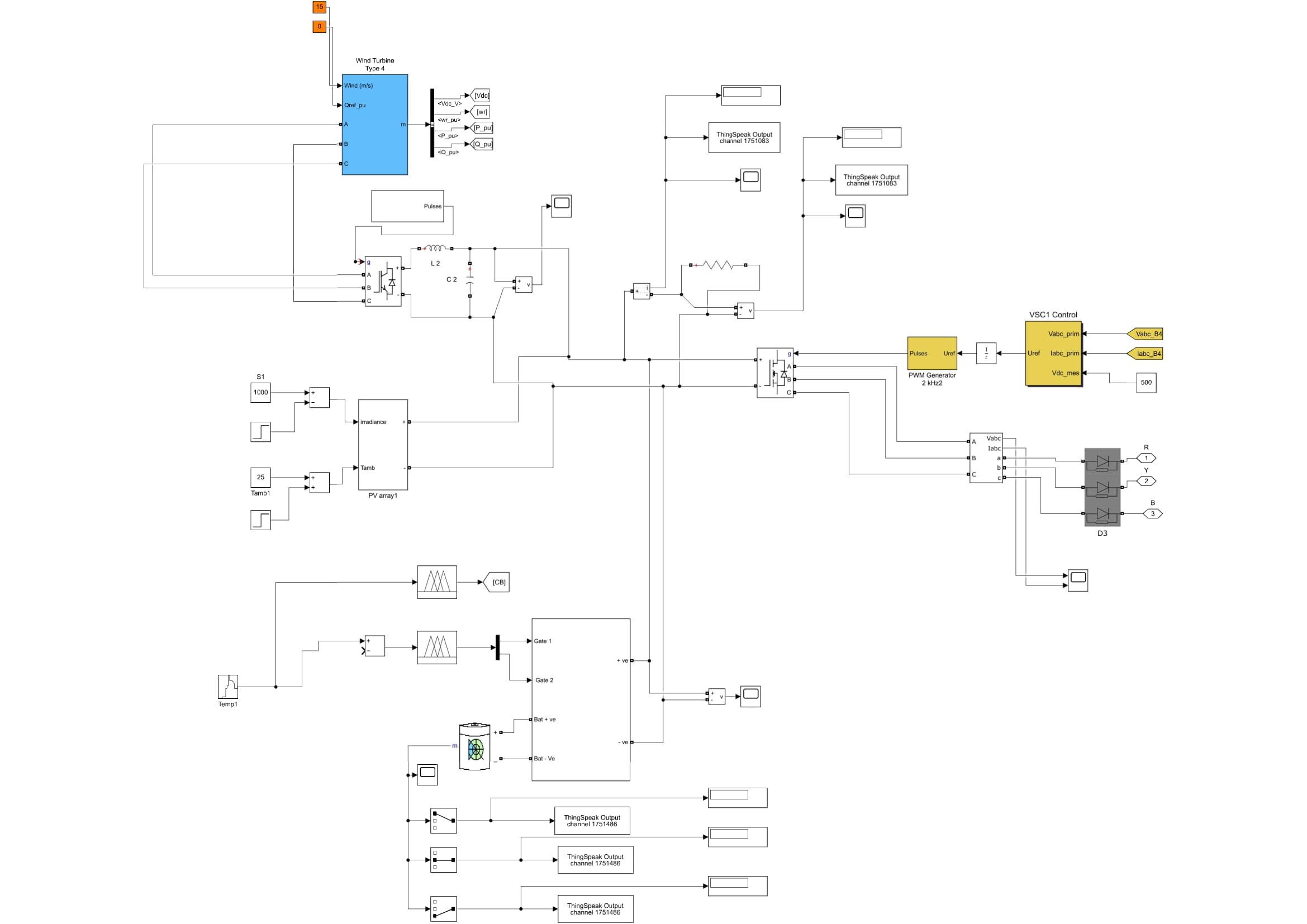
**Fig.3.8 Universal bridge converter**

The output from the micro grid is constant DC but we need AC supply to AC grid so we use universal bridge inverter to convert DC to AC voltage and current.

The inverter gate pulse is triggered by pwm pulse generator which is triggered by adaptive neuro fuzzy inference system controller to control the power flow to meet load demand.

**3.3 IMPLEMENTATION OF MICRO GRID WITH ANFIS CONTROLLER**

**AND BATTERY BACKUP**

****

**Fig.3.9 Implementation of micro grid with anfis controller and battery backup**

The Figure 3.9 shows the implementation of micro grid with anfis controller and battery backup the output of the micro grid is DC in nature and required power supply is AC so we use inverter to convert DC to AC the inverter is controlled by anfis controller by triggering gate pulse. The gate pulse is triggered until the load demand is satisfied and the excessive power is stored in battery if the power generation is not enough for satisfy the load demand the power from the battery is supplied to inverter to meet the load demand

Here the ANFIS controller role is triggered the gate pulse until the load demand is satisfy and the other two Sugeno type fuzzy logic controllers are used to operate the battery for charging and discharging.

**3.3.1 Implementation of anfis controller and battery backup**

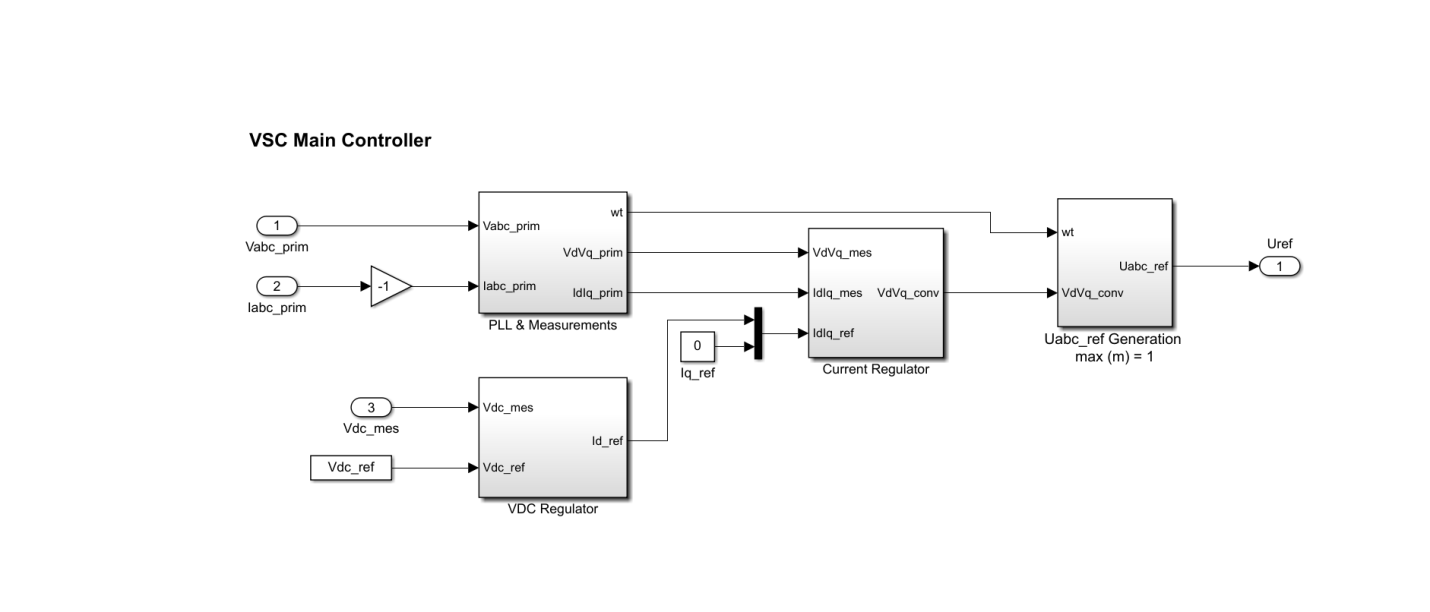
The ANFIS is a hybrid model that combines ANN with fuzzy systems to achieve the advantages of both systems. It implements using a Takagi-Sugeno or mamdani based fuzzy method.

Artificial neural network (ANN) is a network of efficient computing systems the central theme of which is borrowed from the analogy of biological neural networks. ANNs are also named as “artificial neural systems,” parallel distributed processing systems,” “connectionist systems.” ANN acquires large collection of units that are interconnected in some pattern to allow communications between units. These units, also referred to as nodes or neurons, are simple processors which operate in parallel.

Every neuron is connected with other neuron through a connection link. Each connection link is associated with a weight having the information about the input signal. This is the most useful information for neurons to solve a particular problem because the weight usually inhibits the signal that is being communicated.

Following are some reasons to use fuzzy logic in neural networks.

* Fuzzy logic is largely used to define the weights, from fuzzy sets, in neural networks.
* When crisp values are not possible to apply, then fuzzy values are used.
* We have already studied that training and learning help neural networks perform better in unexpected situations. At that time fuzzy values would be more applicable than crisp values.
* When we use fuzzy logic in neural networks then the values must not be crisp and the processing can be done in parallel

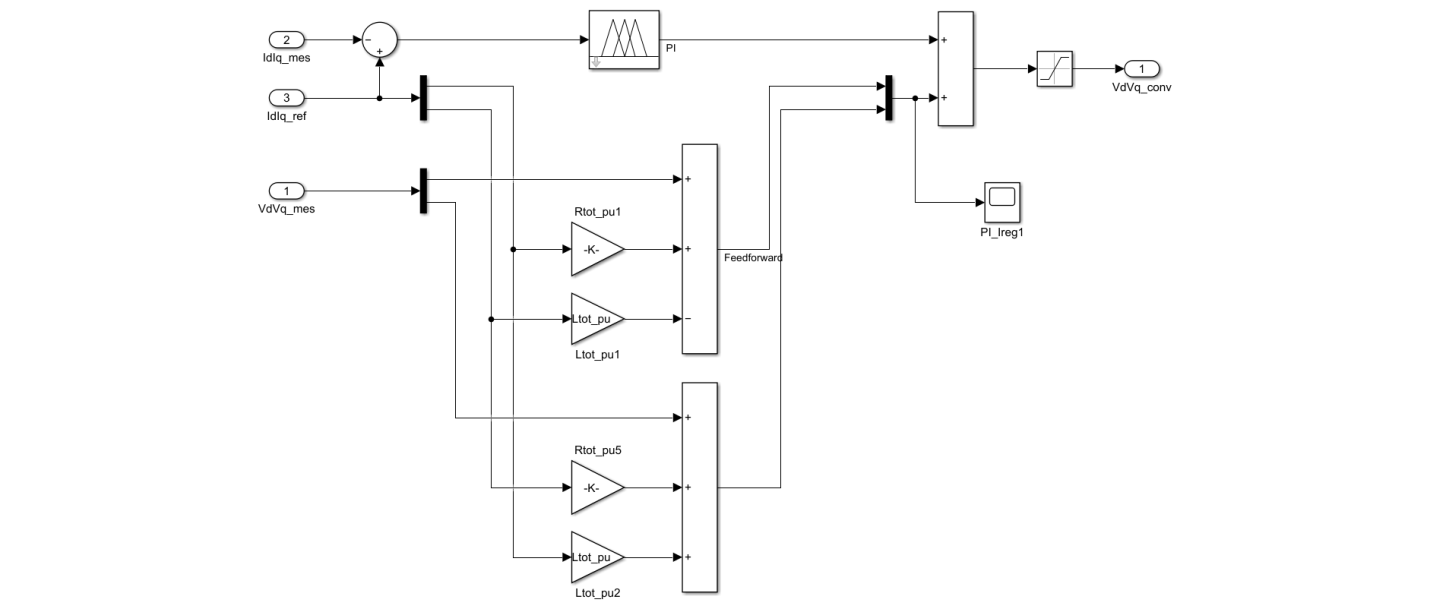


**Fig.3.10 Vsc main controller**

The above Figure 3.10 shows units, also referred to as nodes or neurons are simple processors which operate in parallel.

Each unit are connected with other unit through a connection link and each connection link is associated with a weight having the information about the input signal.

This information helps to solve a problem more efficiently.



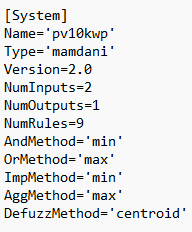
**Fig.3.11 Anfis controller**

Adaptive Fuzzy Controller is designed with some adjustable parameters along with an embedded mechanism for adjusting them. Adaptive controller has been used for improving the performance of controller. Design of a controller is based on an assumed mathematical model that resembles a real system. The error between actual system and its mathematical representation is calculated and if it is relatively insignificant than the model is assumed to work effectively.

A threshold constant that sets a boundary for the effectiveness of a controller also exists. The control input is fed into both the real system and mathematical model.

After designing the adaptive fuzzy logic controller with neural network. We have to write fuzzy logic rules to define the working of the system. Here we have to control the power flow based on the load requirement we have two inputs voltage and current for load demand it was fed into neural network (VSC main controller) to attain a weight having information about input signals after that the output from neural network is passes to fuzzy logic controller with mathematical model. The input processed according to the fuzzy rule written in ANFIS and the output is produced and triggering the gate pulse.

In order to write fuzzy logic rule we have to define the system parameters like name, type, number of inputs, outputs and rules, defuzzification method declaration and operator replacement.

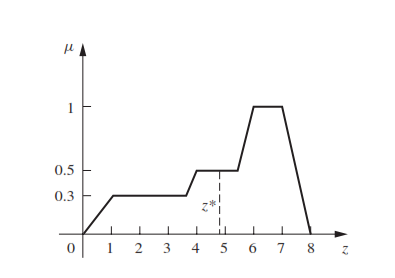


The above fuzzy logic system is Mamdani based rule type it has the following rules

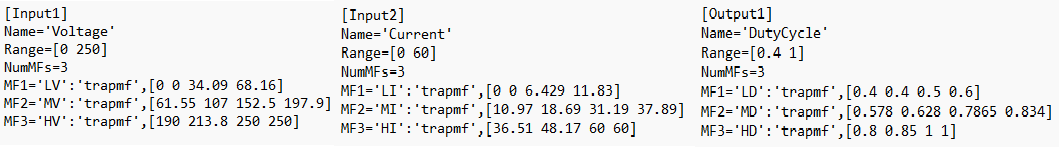
* Fuzzify all input values into fuzzy membership functions.
* Execute all applicable rules in the rulebase to compute the fuzzy output functions.
* De-fuzzify the fuzzy output functions to get "crisp" output values.

And it has total two number of input and produce one output after execute all the nine rules present in the system the operators or replaced by constituents of minimum and maximum. A fuzzy logic function represents a disjunction of constituents of minimum, where a constituent of minimum is a conjunction of variables of the current area greater than or equal to the function value in this area (to the right of the function value in the inequality, including the function value).After that we have to specify defuzzification method. Defuzzification is the process of producing a quantifiable result in Crisp logic, given fuzzy sets and corresponding membership degrees. It is the process that maps a fuzzy set to a crisp set. It is typically needed in fuzzy control systems. These will have a number of rules that transform a number of variables into a fuzzy result, that is, the result is described in terms of membership in fuzzy sets. Defuzzification is the conversion of a fuzzy quantity to a precise quantity, just as fuzzification is the conversion of a precise quantity to a fuzzy quantity µ. Here we use centroid defuzzification method. This method is also known as centre of mass, centre of area or centre of gravity. It is the most commonly used defuzzification method. The defuzzified output z\* is given by

 (3.5)



**Fig.3.12 Centroid defuzzification waveform**



After that we have to specify input and output details and its membership. Here we have 3 membership functions for each input and output the membership function is classified as low, medium, high respectively. We use Matlab inbuilt membership function “tramf” to computes fuzzy membership values.

y = tramf(x, params) returns fuzzy membership values computed using the following trapezoidal membership function is given by

 (3.6)

To define the membership function parameters, specify params as the vector

[a b c d] membership values are computed for each input values in x.

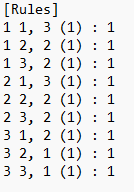
Now we have to define rules. The reference power rules are given by

**Table.3.1 Reference Power Rule**

|  |  |  |
| --- | --- | --- |
| PGen (t-1) | PL (t) | Reference Power |
| High | High | Low |
| High | Low | High |
| Low | High | Very Low |
| Low | Low | Medium |

Here PGen(t-1) is an generated power and PL(t) is a load power requirement

Based on the above table we have 3 membership functions so the rules can be written as



**Fuzzy rules** are used within fuzzy logic systems to infer an output based on input variables. For example

Premise: *x is A*

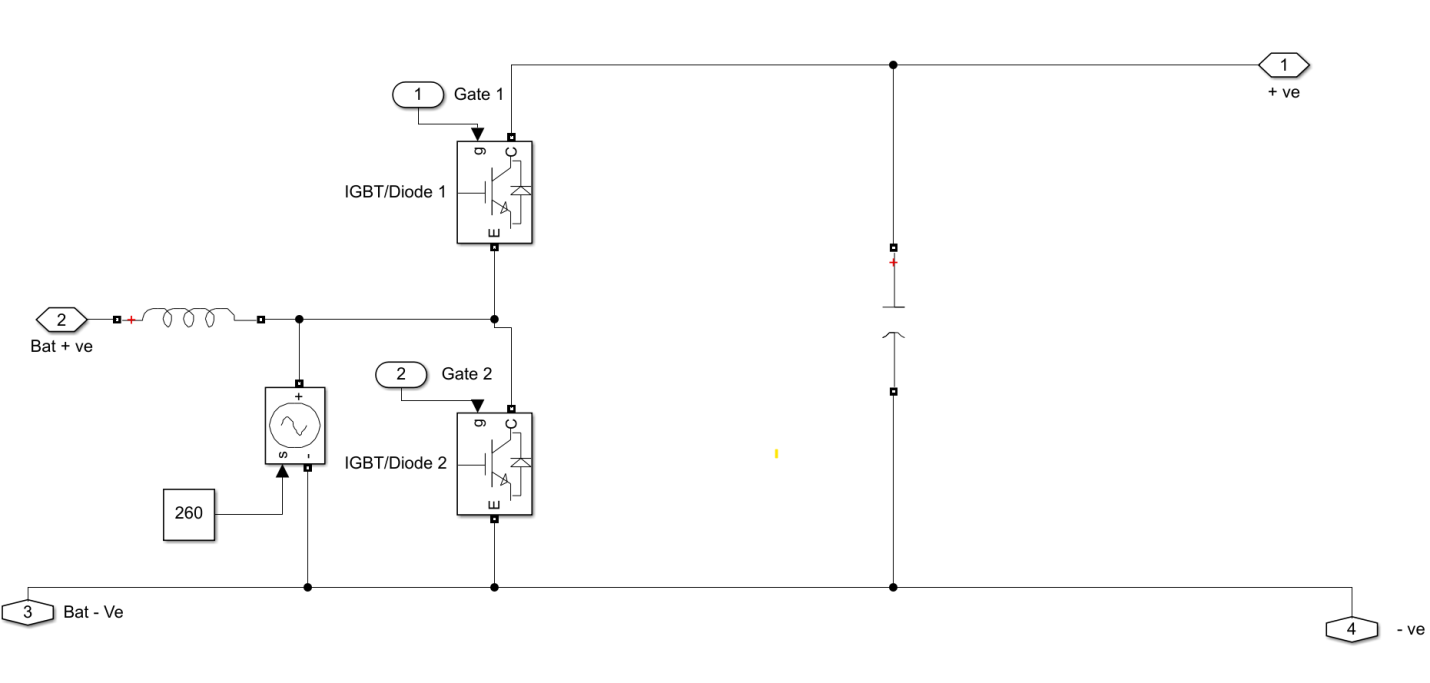
Implication: ***IF****x is A****THEN****y is B*

Consequent: *y is B*

Now the fuzzy logic rule is written the mamdani based fuzzy logic controller fuzzify all input values into fuzzy membership functions and execute all applicable rules in the rulebase to compute the fuzzy output functions and de-fuzzify the fuzzy output functions to get "crisp" output values.

The output from the ANFIS controller is fed into PWM pulse generator through unit delay and the output from PWM pulse generator is used to trigger the gate pulse of the inverter to control power flow.

Similar to ANFIS controller another two fuzzy logic controller without neural network and mathematical model is implemented to operate battery charging and discharging. The batter is connected to the micro grid through a subsystem consist of IGBT devices which act as a switch to control the battery operations. The IGBT devices are controlled by two fuzzy logic controller one for battery charging and other for battery discharging. The type of the fuzzy logic controller is Sugeno type. It is similar to Mamdani, but the defuzzification process is included in the execution of the fuzzy rules. These are also adapted, so that instead the consequent of the rule is represented through a polynomial function (usually constant or linear). Similar to ANFIS the fuzzy logic rule is written to control the IGBT devices.



**Fig.3.13 Subsystem for battery connection**

The Figure 3.13 shows the subsystem for battery connection.

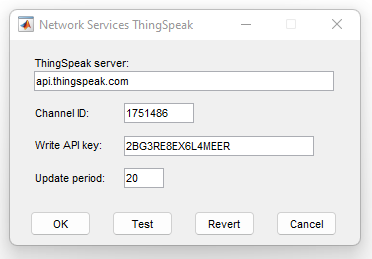
**3.4 IMPLEMENTATION OF IoT DEVICE**

In this project IoT is used for monitoring purposes to collect data from the system in real time and store the data for monitoring and data analysis purpose. Here we use ThingSpeak an Matlab inbuilt real time component



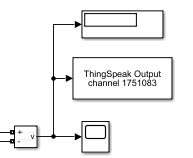
**Fig.3.14 ThingSpeak Output**

In order to use ThingSpeak we have to open an account in thingspeak website for writing or reading data (<https://thingspeak.com/>). The thingspeak cloud has channel to write or read data. Each channel has many fields in which data is written or readied. Each channel has a unique ID and respective write are read API to connect with Matlab component. In Matlab component ThingSpeak we configure the thingspeak cloud account. In our case we need to write the data to cloud so we use ThingSpeak output component and configure with thingspeak cloud account through write API key. The configuration is shown below



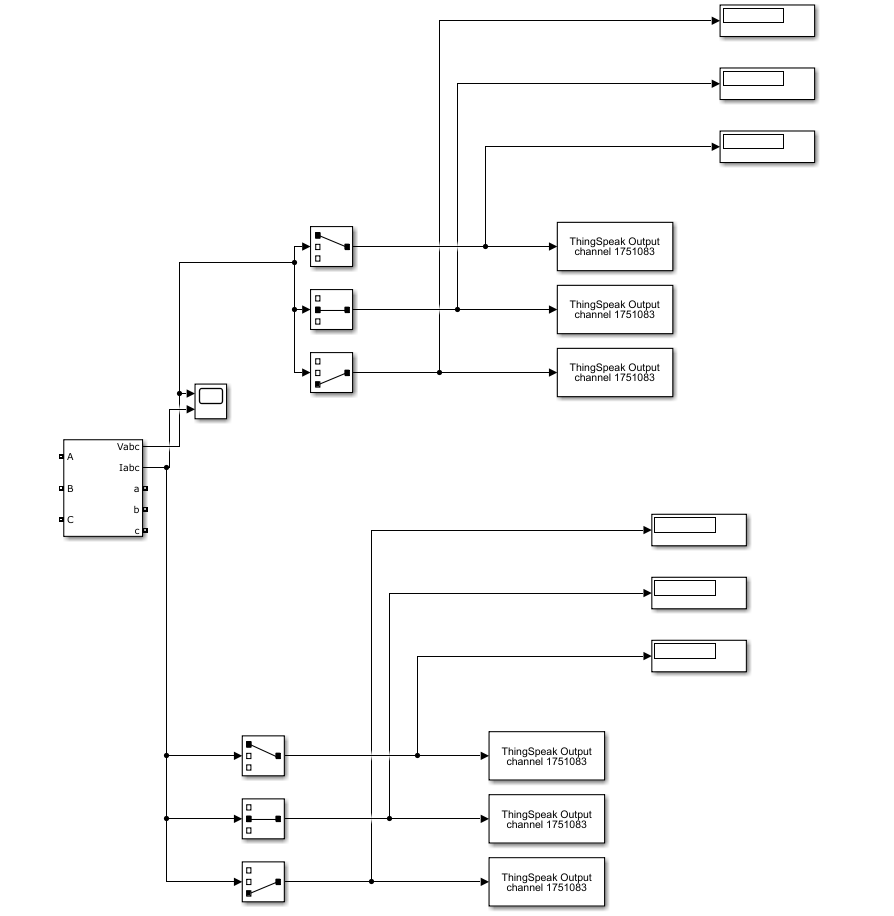
**Fig.3.15 Thingspeak configuration**

In our case we have to write DC voltage and current as well as 3 phase AC voltage and current the three phase value has vector in nature of size 3. So we use selector to select each index value and then write to the cloud channel through ThingSpeak output and for DC we can directly write data to cloud channel through ThingSpeak output. The implementation of ThingSpeak is show below



**Fig.3.16 Implementation of thingspeak to dc voltage measurement**

The Figure 3.16 shows the implementation of ThingSpeak to DC voltage measurement

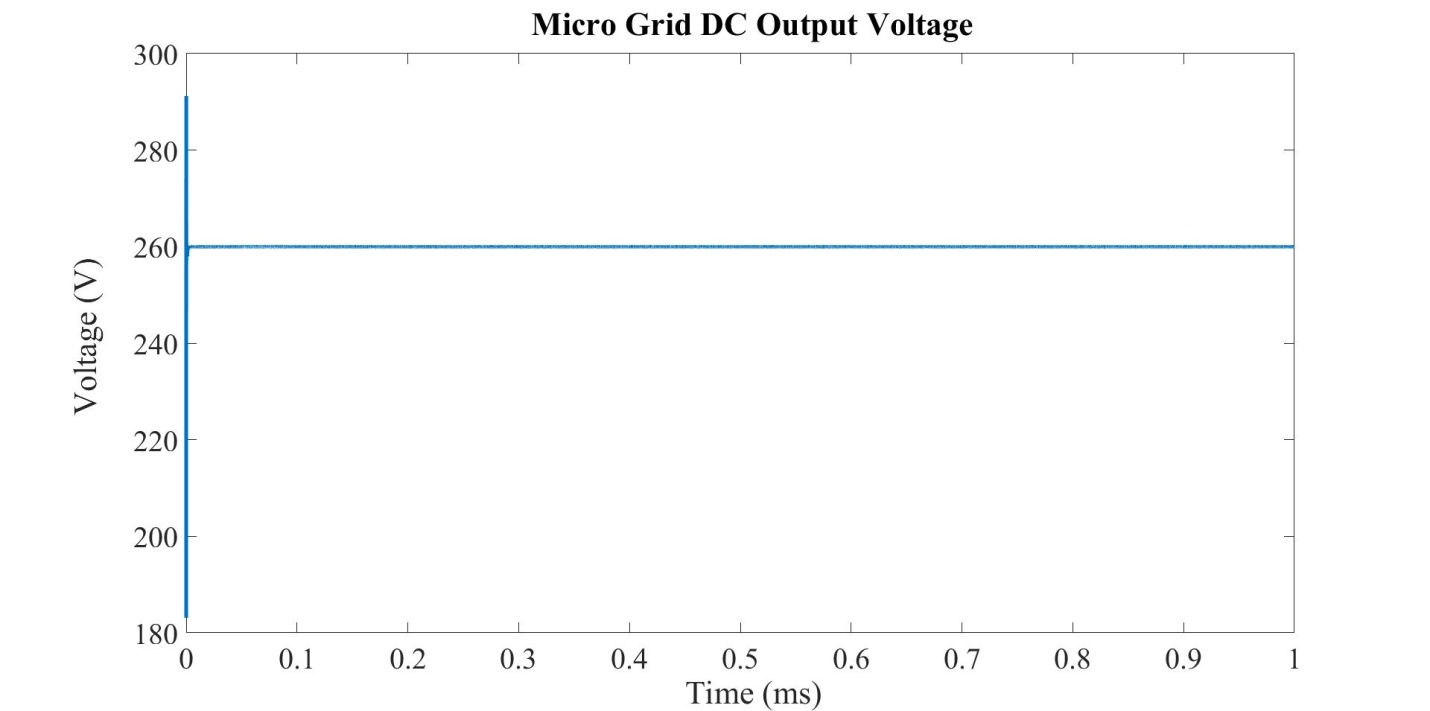
****

**Fig.3.17 Implementation of thingspeak to ac v-i measurement**

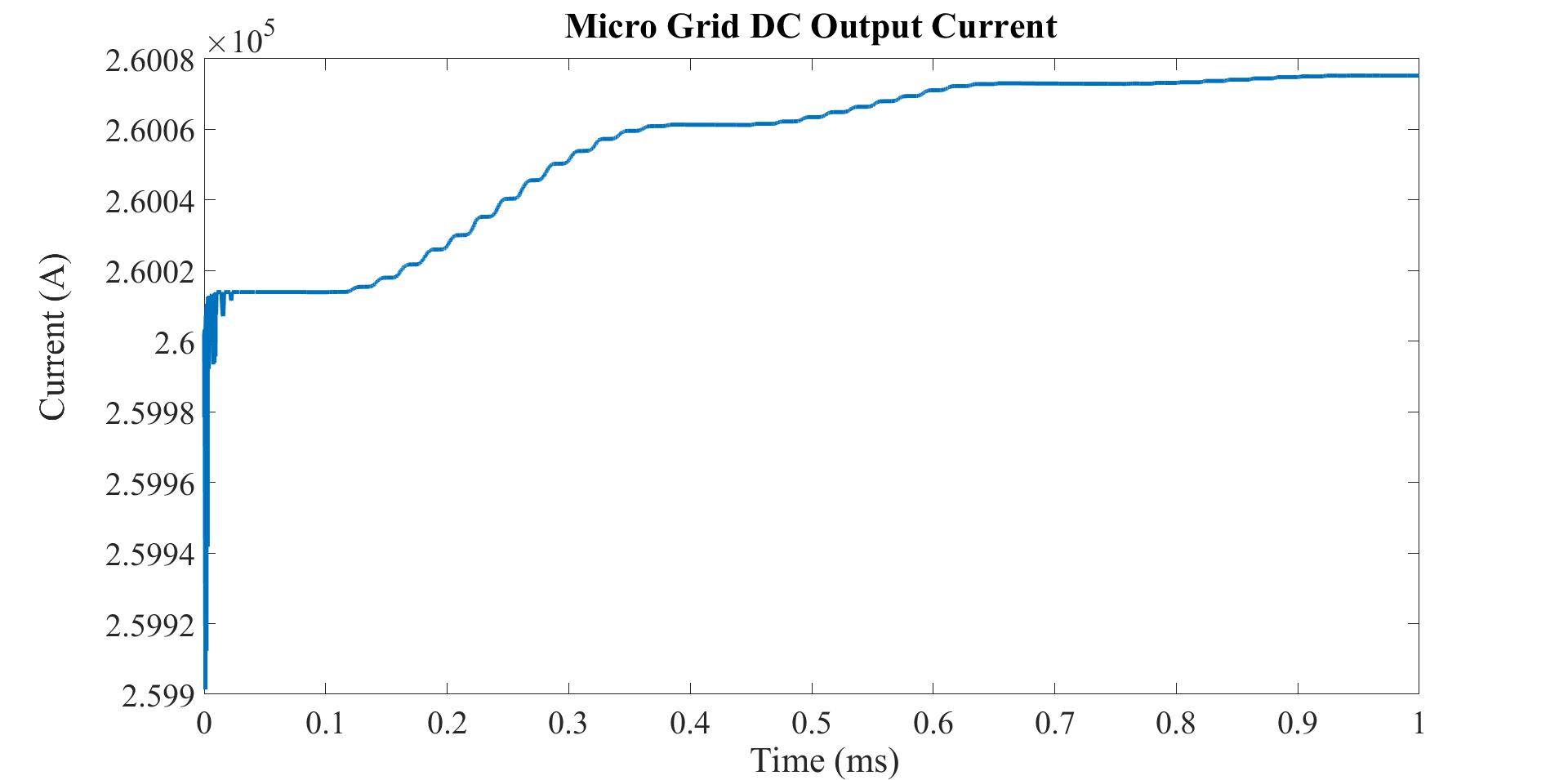
The Figure 3.17 shows the implementation of ThingSpeak to AC voltage-current measurement

**3.5 SIMULATION RESULTS**

After the implementation of model in the Matlab save the file and upload fuzzy logic rule file by typing fuzzy command in command prompt after that save the file and run the model. Based on the irradiation, temperature to PV panel array and wind speed to wind turbine the total DC output voltage and current generated from the micro grid is shown in below graph.

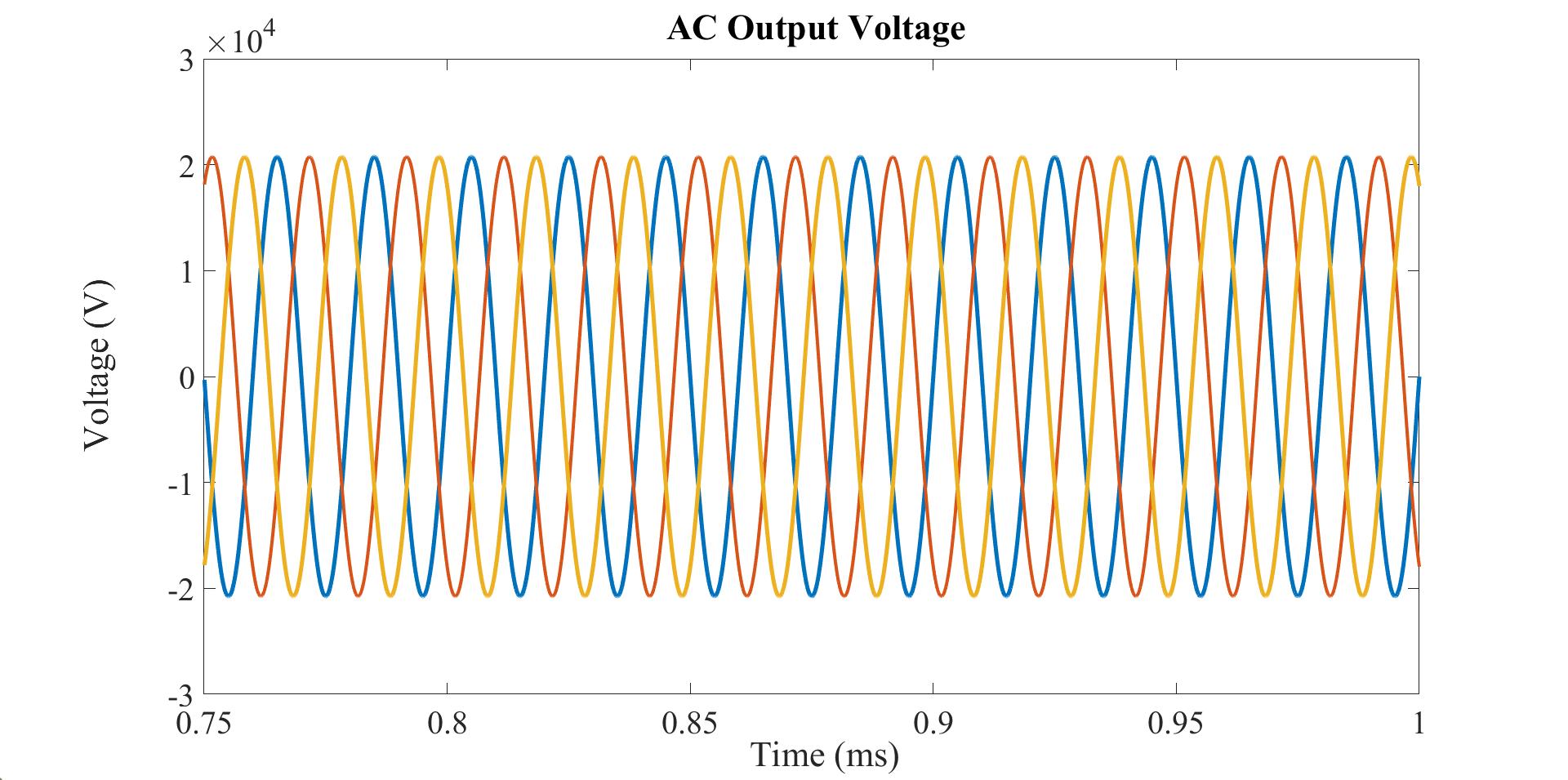
****

**Fig.3.18 Micro grid dc output voltage**

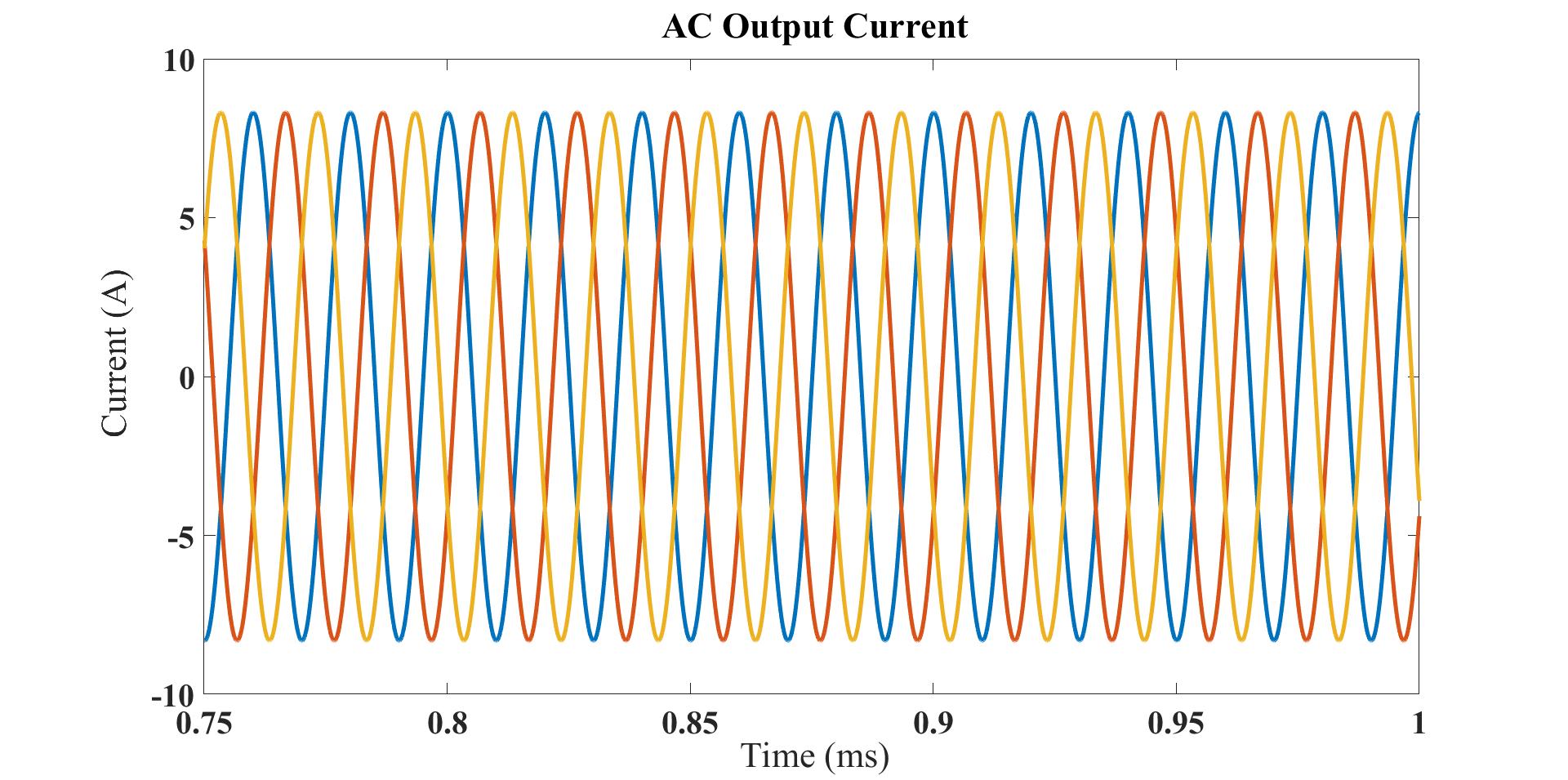
****

**Fig.3.19 Micro grid dc output current**

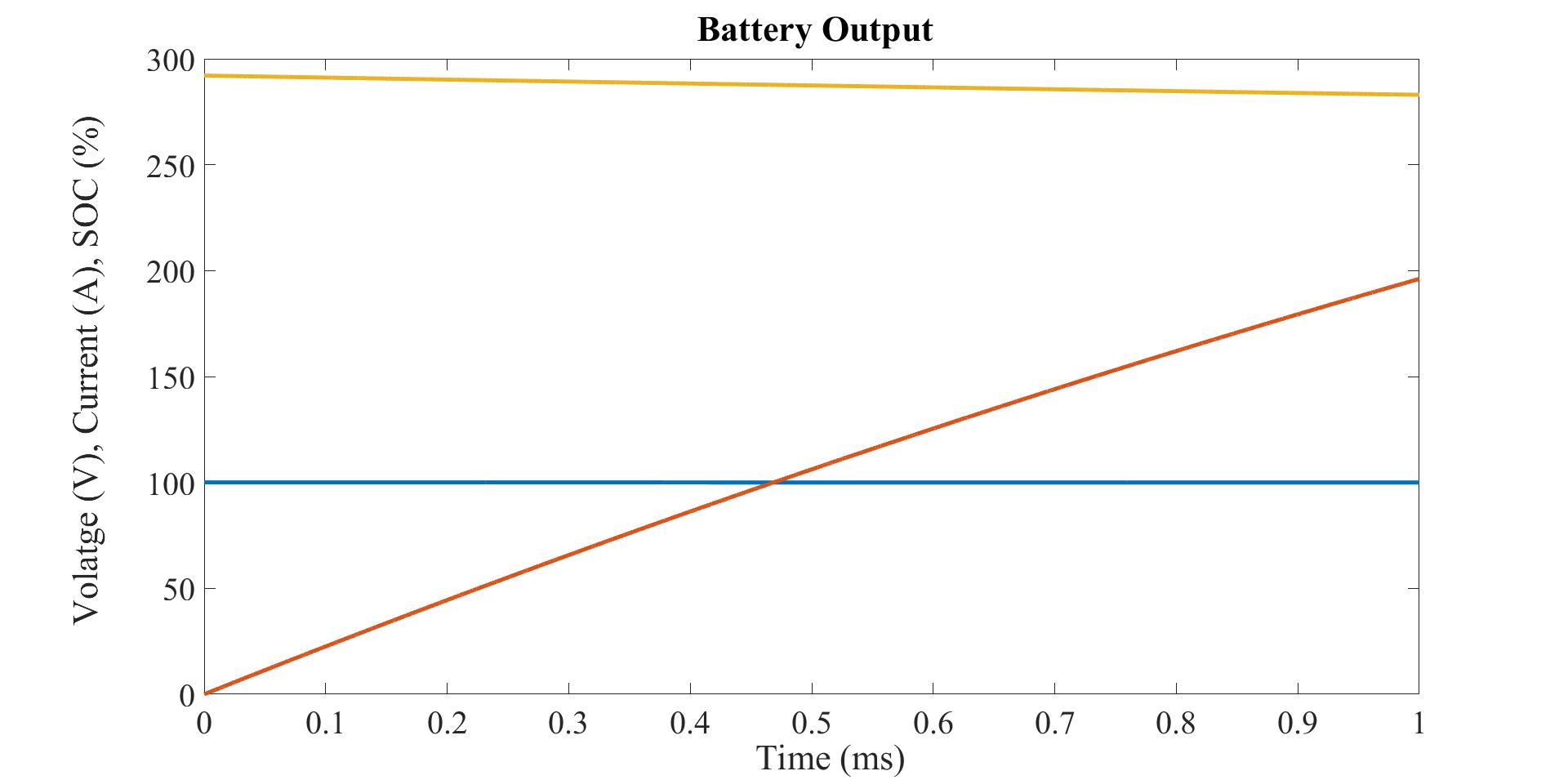
Based on the load requirement the controlled AC output voltage and current with battery charging and discharging operation is shown in the below graph.

****

**Fig.3.20 AC output voltage**

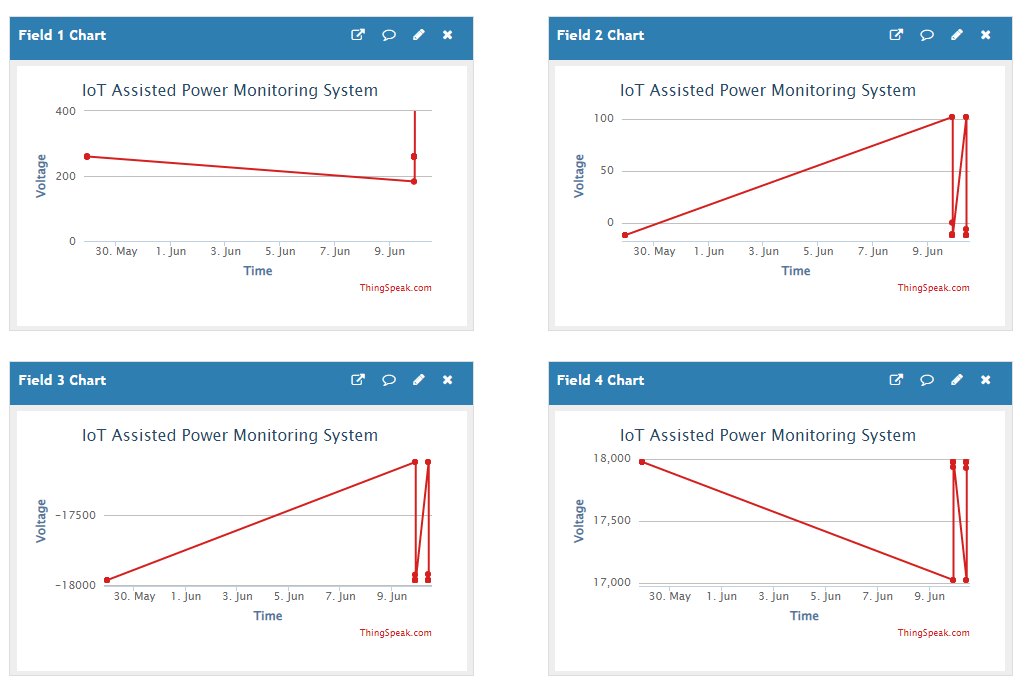
****

**Fig.3.21 AC output current**

****

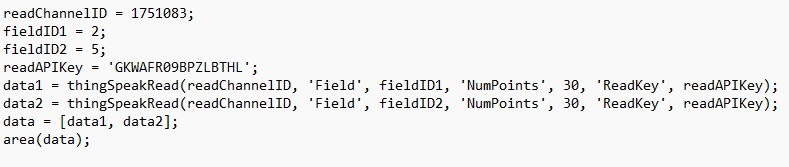
**Fig.3.22 Battery output**

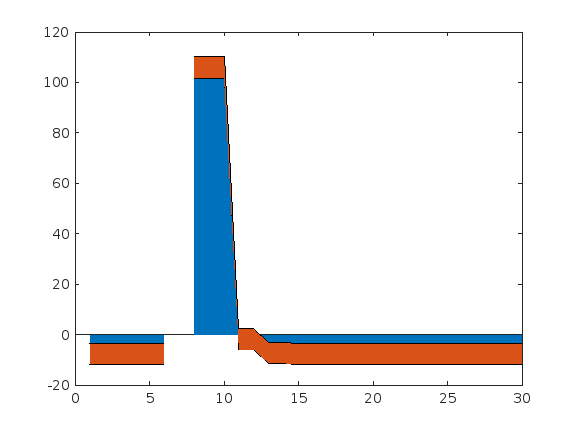
The data from this model is collected and transmitted over internet to cloud platform using ThingSpeak and the data collected can be used for monitoring and for data analysis purpose. The data monitoring graph is shown below

****

**Fig.3.23 Data monitoring graph**

The sample analysis of relation between AC voltage and current is shown below

****



**Fig.3.24 Relationship graph of ac voltage and current**

**CHAPTER 4**

CONCLUSION

Modern grid systems need an intelligent operation in power system infrastructure. SG is a more efficient and reliable grid that addresses several issues that trouble conventional grids. The implementation of a Neuro-Fuzzy-based IoT-assisted power monitoring setup is described in this project. In the smart grid, a controller called ANFIS is used to control hybrid solar and wind power plants and battery is provided for backup purpose. The ANFIS based power management improved the power outcome of the wind and solar power plants to a great extent. The implemented IoT based Neuro-Fuzzy concept for power monitoring using Simulink software and the parameters like power, current, and voltage of the load are taken into consideration. The proposed design would help in the production of low-cost power monitoring and sensing devices that are easy to incorporate into user environments.

**4.1 FUTURE SCOPE**

In the proposed system the IoT devices continuously transfer all the data’s to analytics with the help of IoT sensors. We can do various exploratory data analysis to improve the system further and apply machine learning algorithms to predict the future working and error values that may arise in the give system.

**REFERENCES**

1. F. Al-Turjman and M. Abujubbeh, ‘‘IoT-enabled smart grid via SM: An overview,’’ Future Gener. Comput. Syst., vol. 96, pp. 579–590, Jul. 2019.

2. M. Babar, M. U. Tariq, and M. A. Jan, ‘‘Secure and resilient demand side management engine using machine learning for IoT-enabled smart grid,’’ Sustain. Cities Soc., vol. 62, Nov. 2020, Art. No. 102370.

3. Z. Wang, Y. Liu, Z. Ma, X. Liu, and J. Ma, ‘‘LiPSG: Lightweight privacy preserving Q-learning-based energy management for the IoT-enabled smart grid,’’ IEEE Internet Things J., vol. 7, no. 5, pp. 3935–3947, May 2020.

4. A. H. Bagdadee, M. Z. Hoque, and L. Zhang, ‘‘IoT based wireless sensor network for power quality control in smart grid,’’ Proc. Comput. Sci., vol. 167, pp. 1148–1160, Jan. 2020

5. V. Abreu, A. O. Santin, E. K. Viegas, and V. V. Cogo, ``Identity and access management for IoT in smart grid,'' in Proc. Int. Conf. Adv. Inf. Netw. Appl. Cham, Switzerland: Springer, Apr. 2020, pp. 12151226.

6. G. Hafeez, Z. Wadud, I. U. Khan, I. Khan, Z. Shaq, M. Usman, and M. U. A. Khan, ``Efcient energy management of IoT-enabled smart homes under price-based demand response program in smart grid,'' Sen- sors, vol. 20, no. 11, p. 3155, Jun. 2020.

7. P. Pawar and K. P. Vittal, ``Design and development of advanced smart energy management system integrated with IoT framework in smart grid environment,'' J. Energy Storage, vol. 25, Oct. 2019, Art. no. 100846.

8. Y. Ylmaz and S. Uludag, ``Timely detection and mitigation of IoT-based cyberattacks in the smart grid,'' J. Franklin Inst., vol. 358, no. 1, pp. 172192, Jan. 2021.

9. Y. Saleem, N. Crespi, M. H. Rehmani, and R. Copeland, ``Inter- net of Things-aided smart grid: Technologies, architectures, applica- tions, prototypes, and future research directions,'' IEEE Access, vol. 7, pp. 6296263003, 2019.

10. A. H. Rabie, A. I. Saleh, and H. A. Ali, ``Smart electrical grids based on cloud, IoT, and big data technologies: State of the art,'' J. Ambient Intell. Hum. Comput., vol. 12, no. 10, pp. 94499480, Oct.

11. D. Amarnath and S. Sujatha, ``Internet-of-Things-aided energy man- agement in smart grid environment,'' J. Supercomput., vol. 76, no. 4, pp. 23022314, Apr. 2020.

12. A. Jiang, H. Yuan, D. Li, and J. Tian, ``Key technologies of ubiquitous power Internet of Things-aided smart grid,'' J. Renew. Sustain. Energy, vol. 11, no. 6, Nov. 2019, Art. no. 062702.

13. A. M. Fadel, A. M. Ibrahim, A. K. Tamazin, R. A. Hamdy, and A. S. Abdel-Khalik, ``IoT-based power management for DC microgrids,'' in Proc. 21st Int. Middle East Power Syst. Conf. (MEPCON), Dec. 2019, pp. 11071111.