FAULT DETECTION IN MULTILEVEL H-BRIDGE INVERTER USING MACHINE LEARNING ALGORITHM

A PROJECT REPORT

Submitted by

BAASKAR R	**715519105301**
GOWTHAMAN K	**715519105302**
NICK SALTER S	**715519105307**
SANJAY S	**715519105311**

In partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

ELECTRICAL AND ELECTRONICS ENGINEERING

PSG INSTITUTE OF TECHNOLOGY AND APPLIED RESEARCH

COIMBATORE - 641 062

MAY 2023

ANNA UNIVERSITY CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report "**FAULT DETECTION IN MULTILEVEL H-BRIDGE INVERTER USING MACHINE LEARNING ALGORITHM"** is the bonafide work of **BAASKAR R (715519105301), GOWTHAMAN K (715519105302), NICK SALTER S (715519105307), SANJAY S (715519105311**), who carried out the project work under my supervision.

HEAD OF THE DEPARTMENT	**FACULTY GUIDE**
Dr. C. L. Vasu	Dr. M. Sathiyanathan
Professor and HOD	Associate professor
Electrical and Electronics	Electrical and Electronics
Engineering	Engineering
PSG Institute of Technology	PSG Institute of Technology
and Applied Research	and Applied Research

Certified that candidate was examined in the viva-voice examination held on

Internal Examiner External Examiner

ABSTRACT

The development of power electronic converter, especially multilevel converters, are used in high power applications for several decades. The complex switching and increased power of semiconductor devices are prime reasons for faults in multilevel inverters and have raised question about reliability. To improve the reliability, a cost-effective solution in terms of fault diagnosis is essential. In this context, this study proposed an Open Circuit Fault (OCF) diagnosis technique for a switching device in a five-level cascaded H-bridge multilevel inverter using Machine Learning (ML). The aim of the project is to design a classifier for fault detection in a 5-level cascaded H- bridge inverter which has various applications in the real world. This work estimates the various algorithms in Machine Learning which best suits for this work with various permutations and combinations of the dataset employed for building the model. The MATLAB/Simulink results and prototype results are the evidence to support the proposed fault diagnosis technique. This work is a small try for modelling the complete fault detection and prognosis of the several elements and factors affecting the healthy state nature of an electronic system.

ACKNOWLEDGEMENT

We would like to express our gratitude to **Dr. G. Chandramohan,** Principal and management of PSG Institute of Technology and Applied Research, Coimbatore-641062, for providing the opportunity to carry out our project with the best infrastructure and facilities.

We extend our sincere thanks to **Dr. C. L. Vasu**, Professor and Head of the Department of Electrical and Electronics Engineering for providing consistent support throughout the project.

We express our sincere thanks to our project guide **Dr. M. Sathiyanathan**, Associate Professor, for giving his unfailing support throughout the project.

We would also like to thank all the faculty members and supporting staff members of Department of Electrical and Electronics Engineering, PSG Institute of Technology and Applied Research, Coimbatore for providing assistance in carrying out this project work. Finally, we express our deep sense of gratitude to my parents, friends and all others who were indirectly involved with this project for their invaluable help. The technical information derived from various research papers is gratefully acknowledged.

TABLE OF CONTENT

CHAPTER NO.		TITLE	PAGE NO.
		ABSTRACT	**iii**
		ACKNOWLEDGEMENT	**iv**
		LIST OF FIGURES	**ix**
		LIST OF TABLES	**x**
1		**INTRODUCTION**	**1**
	1.1	FIVE LEVEL H-BRIDGE INVERTER	1
	1.2	PROBLEM IDENTIFICATION	2
	1.3	OBJECTIVES OF THE PROJECT	2
	1.4	PROPOSED SOLUTION	3
	1.5	ORGANISATION OF REPORT	3
2		**LITERATURE SURVEY**	**5**
3		**ANALYSIS OF H-BRIDGE INVERTER**	**14**
	3.1	3.1 CASCADED H-BRIDGE MULTILEVEL INVERTER	14
	3.2	3.2 MODULATION TECHNIQUES FOR MULTILEVEL INVERTER	15
		3.2.1 Pulse width modulation techniques	15
		3.2.2 Phase disposition pulse width modulation	15

	3.3	OPERATING MODES OF FIVE-LEVEL CASCADED H-BRIDGE MULTILEVEL INVERTER	16
	3.4	DESIGN OF FIVE-LEVEL H-BRIDGE INVERTER SIMULATION IN MATLAB	18
	3.5	OUTPUT WAVEFORM	20
		3.5.1 Output of cascaded h-bridge inverter	20
		3.5.2 Output of h-bridge inverter under fault state	21
4		**IMPLEMENTATION OF MACHINE LEARNING ALGORITHM**	**24**
	4.1	INTRODUCTION TO MACHINE LEARNING	24
		4.1.1 Need for Machine Learning	25
		4.1.2 Classification of Machine Learning	25
		4.1.3 Supervised Learning	26
		4.1.4 Unsupervised Learning	26
		4.1.5 Reinforcement Learning	27
		4.2 TYPES OF ALGORITHMS INVOLVED	27
		4.2.1 Decision Tree Algorithm	27
		4.2.2 Naive Bayes Algorithm	30
		4.2.3 Random Forest Algorithm	32
		4.2.4 K-Nearest Neighbour Algorithm	33

5		**HARDWARE ARRANGEMENT AND FAULT DETECTION SETUP**	**36**
	5.1	HARDWARE SETUP	36
	5.2	TROUBLESHOOTING SETUP	37
	5.3	HARDWARE USED	42
		5.3.1 Arduino Uno	42
		5.3.2 MOSFET	44
		5.3.3 Optocoupler (PC817)	49
6		**RESULTS AND DISCUSSION**	**51**
	6.1	IMPLEMENTATION OF MACHINE LEARNING	51
		6.1.1 Cleaning the Data	51
		6.1.2 Importing the Data	52
		6.1.3 Split the Data into Training and Testing Sets	54
		6.1.4 Deciding the Appropriate Model and the Appropriate Split	55
		6.1.5 Building the Classifier	58
	6.2	MAKING PREDICTIONS	61
	6.3	INPUT FROM USER	61
	6.4	OUTPUT OF CLASSIFIER	61
	6.5	HARDWARE DESIGN	62
	6.6	HARDWARE OUTPUT WAVEFORM	62

6.6.1 Output of Cascaded H-Bridge Inverter Under Fault State 63

6.7 COMPARISON OF ALGORITHM IN SIMULATION MODEL 64

6.8 COMPARISON OF SIMULATION AND HARDWARE MODEL 65

7 CONCLUSION 66

7.1 FUTURE WORK 66

REFERENCES 67

LIST OF TABLES

TABLE NO.	TABLE NAME	PAGE NO.
3.1	Switching Sequence	19
3.2	Pulse width modulation with delay	19
6.1	Comparison of algorithm in simulation model	64
6.2	Comparison of simulation and hardware model	65

LIST OF FIGURES

FIG NO.	NAME OF THE FIGURES	PAGE NO.
3.1	Operating mode for getting output voltage of $+2V_{dc}$	17
3.2	Operating mode for getting output voltage of $+ V_{dc}$	17
3.3	Operating mode for getting output voltage of Zero	18
3.4	Operating mode for getting output voltage of V_{dc}	18
3.5	Operating mode for getting output voltage of $-2 V_{dc}$	19
3.6	Five-level H-bridge inverter	19
3.7	Output Waveform without fault	21
3.8	Output Waveform under switch S_1 in fault condition	22
3.9	Output Waveform under switches S_1 , S_3 in fault condition	23
4.1	Machine Learning process	24
4.2	Decision Tree classification	28
4.3	Bayes Theorem	31
4.4	Naive Bayes Classifier	31
4.5	Random Forest algorithm	32
4.6	K-NN process	34
5.1	Hardware setup	36
5.2	Arduino UNO Micro-controller board	42
5.3	Circuit diagram of optocoupler	49

5.4	Configuration of optocoupler	50
6.1	Complete Dataset after extraction	51
6.2	Combination of faulted switches	52
6.3	Data frame construction	53
6.4	Mapping outputs as targets	54
6.5	Splitting Data into training and testing sets	54
6.6	Selecting Correct algorithm and split	55
6.7	Accuracy for 80% train and 20% test dataset split	56
6.8	Accuracy for 70% train and 30% test dataset split	57
6.9	Accuracy for 60% train and 40% test dataset split	57
6.10	Accuracy for 50% train and 50% test dataset split	58
6.11	Mapping the Output of the separate models	58
6.12	Concatenating into the data frame	59
6.13	Output of building the classifier	60
6.14	Getting the input from user	61
6.15	Output for the classifier	61
6.16	Hardware Design	62
6.17	Hardware output waveform without fault	62
6.18	Output waveform under switch S1 in fault condition	63
6.19	Output waveform under switches S1, S3 in fault condition	63

CHAPTER 1

INTRODUCTION

1.1 FIVE-LEVEL H-BRIDGE INVERTER

The multilevel inverter has been implemented in various applications ranging from medium to high-power levels, such as motor drives, power conditioning devices, also conventional or renewable energy generation and distribution**.** The different multilevel inverter structures are cascaded H-bridge, diode clamped and flying capacitor multilevel inverter. Among the three topologies, the cascaded multilevel inverter has the potential to be the most reliable and achieve the best fault tolerance owing to its modularity. A five-level cascaded H-bridge multilevel inverter. The converter consists of two series connected H-bridge cells which are fed by independent voltage sources. The outputs of the H-bridge cells are connected in series such that the synthesized voltage waveform is the sum of all of the individual cell outputs.

The output voltage is given by

$$V=V_1+V_2 \tag{1.1}$$

Where the output voltage of the first cell is labelled V_1 and the output voltage of the second cell is denoted by V_2. There are five level of output voltage i.e. 2V, V, 0, -V, - 2V. The main advantages of cascaded H-bridge inverter is that it requires least number of components, modularized circuit and soft switching can be employed. But the main disadvantage is that when the voltage level increases, the number of switches increases and also the sources, this in effect increases the cost and weight. The cascaded H-bridge multilevel inverters have been applied where high power and power quality are essential, for example, static synchronous compensators, active filter and reactive power compensation applications, photo voltaic power conversion, uninterruptible power supplies, and magnetic resonance

imaging. Furthermore, one of the growing applications for multilevel motor drive is electric and hybrid power trains.

1.2 PROBLEM IDENTIFICATION

Electronics are increasingly used in safety, and infrastructure-critical systems. Unexpected failures in such electronic systems during field operation can have severe implications. Failures could be prevented and unexpected system downtime could be eliminated if an appropriate prognostic method is incorporated to determine the advent of failure and mitigate system risk. Electronic system failures can result majorly from its elements rather than the faults in the circuit board (e.g., traces), connectors. The detection of the fault in an efficient way is very much essential for this electronic era which consists of many complex electronic systems.

1.3 OBJECTIVES OF THE PROJECT

- The objective of this project is to detect the fault in an efficient way by creating a classifier model using artificial intelligence algorithms for a 5-level cascaded H-bridge inverter.
- To detect the open-circuit fault conditions for the 8-switches which forms the 5-level H-bridge system.
- To employ intelligent algorithms which comprises of Neural Networks and other machine learning schemes.

This project helps to diagnose the fault in the complex circuits under switch failure It saves time for trouble-shooting circuit, which intern improves usage of the circuit in the application.

1.4 PROPOSED SOLUTION

The major part of solving any problem is the kind of approach that is adopted. These strategies or methods that are used make a lot of difference in obtaining an appropriate solution. The solution can be called an appropriate one, when it satisfies all the requirements, considered from various perspectives that tends to solve the intended issue. In the context of this project, initially the industries who makes the complex circuits can be aware of how much time they waste, if they had fault in circuits and thus motivate them to use the resources properly. But this method has a major disadvantage that many of them may not take this seriously and the issue is not resolved properly. This pushes us to adopt better strategies. The project involves the artificial intelligence in the fault detection to provide desired output and data. By this way employing the fault detection technique for all complex circuit we can easily diagnosed fault state . Thus, a lot of strategies are decided depending on the probability of success and technological factors used to implement these features and reduce the wastage of time.

1.5 ORGANISATION OF REPORT

CHAPTER 1: INTRODUCTION

This chapter gives a brief introduction about the energy management monitoring in practice. The objective and need of the project are explained.

CHAPTER 2: LITRERATURE SURVEY

The various works carried out in the context of energy management in required area are explained. The scope and limitations of the papers published are given.

CHAPTER 3: ANALYSIS OF H-BRIDGE INVERTER

This chapter presents simulation of H-bridge inverter using MATLAB software and the 5-level waveforms of voltage and current are analysed.

CHAPTER 4: IMPLEMENTATION OF MACHINE LEARNING ALGORITHM

This chapter gives the detailed information about machine learning algorithm and shows the implementation of algorithm

CHAPTER 5: HARDWARE ARRANGEMENT FOR FAULT DETECTION

This chapter present the hardware circuit arrangement of the proposed 5-level H-bridge inverter and fault cases created for the analysis.

CHAPTER 6: RESULT AND DISCUSSION

The results achieved in each stage of the project implementation are explained.

CHAPTER 7: CONCLUSION

The scope of the project and future recommendation are specified

CHAPTER 2

LITERATURE SURVEY

2.1 STUDY OF CASCADED H-BRIDGE MULTILEVEL INVERTER

In [1], cascaded H-bridge multilevel inverter with SPWM technique is presented. Multilevel inverter has high power application & low harmonics due to these applications it is used widely in the area of control and energy distribution. This paper includes performance of 3-Level, 5-Level & 7-Level cascaded H-Bridge multilevel inverter with respect to number of switches, total harmonic distortion, waveform pattern, harmonic spectrum, output voltage, voltage stress across the switch & input DC voltage with the help of simulation by using MTLAB/SIMULINK and implementation of Five level cascaded H-bridge multilevel inverter.

2.2 OPEN-CIRCUIT FAULT DIAGNOSIS FOR CASCADED H-BRIDGE MULTILEVEL INVERTER BASED ON LS-PWM TECHNIQUE

In this context, an open-circuit fault diagnosis method for cascaded H-bridge multilevel inverter (CHBMI) based on the level-shifted pulse width modulation (LS-PWM) technique is proposed. Particularly [2] tells that fault is diagnosed based on the algorithm, the proposed method only needs a certain logical judgment to diagnose the fault based on subtle fault characteristics. The output voltage of the H-bridge and load current are used for fault diagnosis.

2.3 FIVE-LEVEL CASCADED H-BRIDGE MULTILEVEL INVERTER FED INDUCTION MOTOR DRIVE USING SINUSOIDAL PWM

In [3] a simple simulation model of three-phase conventional inverter used for speed control of induction motor (two-level) and cascade H-bridge multilevel inverter(five-level) used for speed control of induction motor (closed loop) using

sinusoidal PWM. Hardware implementation of above mention topology is done and compare simulation results and hardware results.

2.4 MODIFIED H-BRIDGE INVERTER WITH REDUCED NUMBER OF SWITCHING DEVICES

In this paper [4], a modified topology for developing H-bridge inverter is proposed which the number of components is lower than other inverters as well as the proposed inverter generates a high number of levels with low total harmonics distortions. In the first, the new developed h-bridge inverter is proposed and then a cascaded connection of this inverter to increase the number of levels is presented. Three different methods are proposed to determine the DC power sources magnitudes and among them, the best algorithm is chosen for comparison the proposed cascaded inverter with other cascaded inverters. Finally, the proposed cascaded inverter consists of two proposed basic inverters to generate thirty-three voltage levels is simulated to evaluate the performance of the proposal.

2.5 FIVE-LEVEL CASCADED H-BRIDGE MULTILEVEL INVERTER USING MULTICARRIER PULSE WIDTH MODULATION TECHNIQUE

The multilevel inverter utilization has been increased since the last decade. These new types of inverters are suitable in various high voltage and high-power applications due to their ability to synthesize waveforms with better harmonic spectrum and faithful output. In [5] presents an asymmetrical five level cascaded H-bridge multilevel inverter, using multicarrier pulse width modulation technique. And also, comparison is made between multicarrier pulse width modulation and the embedded MATLAB function. The Simulation results are presented to prove that THD is reduced with the multicarrier modulation. This topology also reduces the number of switches and also the cost. From the results,

the proposed inverter provides higher output quality with relatively lower power loss as compared to the other conventional inverters with the same output quality.

2.6 FAULT DETECTION AND DIAGNOSIS IN VOLTAGE SOURCE INVERTERS USING PRINCIPLE COMPONENT ANALYSIS

In this paper, the fault detection and diagnosis in a three-phase voltage source inverter (VSI) is investigated. The open-circuit of power switches is considered as the faulty conditions. The proposed algorithm in [6] is the principle component analysis (PCA) that determines the failed switch and the fault starting time based on the inverter output currents. In order to validate the performance of the proposed fault detection method, one of the most commonly used methods presented in the literature entitled “3D Current Trajectory” is also implemented and its results are compared to those of PCA. Under the circumstances where the open-circuit fault occurs at a specific time or the fault time interval is very small, the conventional methods cannot detect the faults. Using the software simulations in MATLAB, it is demonstrated that the proposed method can detect any faults under any circumstances and identify the origin of them.

2.7 CASCADED H-BRIDGE MULTILEVEL INVERTER SYSTEM FAULT DIAGNOSIS USING A PCA AND MULTICLASS RELEVANCE VECTOR MACHINE APPROACH

In this context and to improve fault diagnosis accuracy and efficiency of a cascaded H-bridge multilevel inverter system (CHMLIS), a fault diagnosis strategy based on the principle component analysis and the multiclass relevance vector machine (PCA-mRVM), is elaborated and proposed in paper [7]. First, CHMLIS output voltage signals are selected as input fault classification characteristic signals. Then, a fast Fourier transform is used to pre-process these signals. PCA is used to extract fault signals features and to reduce samples dimensions. Finally, an mRVM model is used to classify faulty samples.

Compared to traditional approaches, the proposed PCA-mRVM strategy not only achieves higher model sparsity and shorter diagnosis time, but also provides probabilistic outputs for every class membership. Experimental tests are carried out to highlight the proposed PCA-mRVM diagnosis performances.

2.8 FAULT DETECTION AND LOCALIZATION FOR CASCADED H-BRIDGE MULTILEVEL CONVERTER WITH MODEL PREDICTIVE CONTROL

In this article, a model predictive control (MPC)-based fault detection and localization (FDL) scheme is proposed for the CHB multilevel converter that utilizes the already-installed ac current measurement and does not require additional sensors. In [8]the proposed FDL scheme can detect single or multiple open-circuit switch faults in the CHB multilevel converter by comparing the predicted states of the MPC controller with the actual measured states. The detected fault(s) is(are) further localized through a fault localization matrix which narrows down the faulty switch using historical fault data. The proposed scheme is shown to complete the FDL process for both single or multiple open-switch faults within a fundamental cycle period, even during low-load conditions. It has also been shown that the proposed scheme is immune to system dynamics such as load and voltage variations, and has been validated experimentally for various open-circuit fault scenarios

2.9 A GENERALIZED SWITCH FAULT DIAGNOSIS FOR CASCADED H-BRIDGE MULTILEVEL INVERTERS USING MEAN VOLTAGE PREDICTION

This article presents a novel generalized open-switch fault-diagnostic approach for an N-level cascaded H-bridge multilevel inverter. In [9] the detection technique, half-cycle mean values of bridge voltages, which are calculated for positive and negative half cycles individually, are used as fault

identification features. These means under open-switch fault are predicted from the reference half-cycle means and compared with those measured values to locate the open-switch fault. This quick detection scheme can identify the faulty switch within one fundamental period of output voltage for the cascaded inverters having different number of voltage levels. The computation requirement is minimum, since this approach does not need complex calculations or domain transformations. This strategy can efficiently locate the faulty switch of the cascaded inverter working with various level-shifted pulse width modulation schemes, different loading conditions, modulation indexes, and switching frequencies. The potency of the fault classification algorithm is verified through simulation and experimentation.

2.10 A FAULT DETECTION METHOD IN CASCADED H-BRIDGE MULTILEVEL INVERTER

This paper presents the switch-fault detection method in a cascaded H-bridge multilevel inverter (CHBMI). When the open-fault occurred on the H-bridge inverter switch of single-phase CHBMI, the body diode of fault switch can provide a freewheeling current path. However, in [10] the output current distortion of short-fault is different from that of the open-fault because disconnection of fuse cannot provide a freewheeling current path. The fault detection method is based on the zero current time analysis according to zero-voltage switching states. The proposed method is available in the rotating level-shifted pulse width modulation (RLS-PWM) method. Using the proposed method, it is possible to detect not only the fault location of cell but the fault location of switch accurately. The PSIM simulation results show the effectiveness of proposed fault detection method.

2.11 OPEN SWITCH FAULT DETECTION IN CASCADED H-BRIDGE MULTILEVEL INVERTER USING NORMALISED MEAN VOLTAGES

In [11] the paper presents a simple and fast diagnostic method for single open switch fault in Cascaded H-Bridge Multi-Level Inverters. The proposed fault diagnostic scheme utilises inverter output voltage and voltage across each H-bridge. The variations in the mean values of the measured parameters are used as fault features. The classification algorithm is simple and the fault location identification can be done accurately within one cycle of the output voltage. A Modified Phase Disposition Pulse Width Modulation method is employed for controlling the inverter switches. MATLAB/Simulink simulation of the 5-level Cascaded Multilevel inverter is presented. The simulation results show that proposed fault detection method is performing efficiently under open fault of different switches. This diagnostic method can be extended to higher level inverters.

2.12 A FAULT-TOLERANT HYBRID CASCADED H-BRIDGE MULTILEVEL INVERTER

The cascaded H-bridge (CHB) inverter is one of the most attractive multilevel topologies for renewable energy applications. Due to the fact that CHB inverters employ a large number of components, they suffer from a higher probability of fault, which reduces the system reliability. A fault-tolerant operation for a CHB inverter is described in this article. New features ensure reliable and robust operation of the converter in the event of a fault. Especially in [12]the proposed strategy uses an additional cross-coupled CHB (X-CHB) unit in companion with the existing CHB units to support the output voltage and ensure continuity of operation in the event of an open/short-circuit fault. The operation of the proposed X-CHB inverter is described in detail. Simulation and

experimental verification of the proposed concept are demonstrated using a seven-level CHB. Both simulation and experimental results confirm the fault-tolerant operation of the X-CHB for a battery energy storage system in case of switch faults.

2.13 FAST FAULT DETECTION OF OPEN POWER SWITCH IN CASCADED H-BRIDGE MULTILEVEL INVERTERS

The Cascaded H-Bridge topology has in recent years become popular for use in medium voltage drives. The reliability of this topology however is of concern due to the large number of semiconductor switches. The natural redundancy possessed by the CHB topology can only be realized if failed devices may be identified and removed from the system. In [13] a fault diagnostic method for open switch fault is proposed. The method is based on the prediction of the active and zero voltage states of each module. With the proposed method, the exact location of the switch under fault in an H-bridge module can be determined quickly. The performance of the proposed technique has been verified through simulation.

2.14 OPEN CIRCUIT SWITCH FAULT DETECTION IN MULTILEVEL INVERTER TOPOLOGY USING MACHINE LEARNING TECHNIQUES

Multilevel inverter is mostly used in high voltage and high-power applications in industry. The possibility of faults raises with an increment in the number of switches in multilevel inverter. In power industries, the reliability of multilevel inverters is one of the main concerns. Hence methods for detecting switch faults are required to improve in the reliability. This paper [14] is mainly focused on open circuit switch fault detection for multilevel inverter. The proposed scheme identifies failed switches by monitoring capacitor current and switches current data. The diagnosis techniques are Artificial Neural Network

(ANN), k-Nearest Neighbours (KNN), Support Vector Machines (SVM) and Decision Tree (DT). These methods are only capable for diagnosing failed switches. On identification of the faulty switch, switching sequence has to be reconfigured such that the output voltage is restored to its healthy operating conditions.

2.15 DIAGNOSIS AND LOCATION OF THE OPEN-CIRCUIT FAULT IN MODULAR MULTILEVEL CONVERTERS: AN IMPROVED MACHINE LEARNING METHOD

In this paper [15], the fault diagnosis and location (FDL) problem of the open circuit fault for modular multilevel converter (MMC) is investigated. A mixed kernel support tensor machine (MKSTM) is provided, and it's employed to improve the support tensor machine which is an important algorithm of machine learning. By extracting the characteristic data of ac current and internal circulation current in either normal operation or open-circuit fault, then training and classifying the obtained samples with MKSTM, FDL of MMC can be realized with the supplied algorithm. Finally, experimental results show that the classification accuracy of MKSTM algorithm is improved observably than single kernel function STM such as linear, Radial Basis function (RBF), sigmoid and polynomial types. Synchronously, the open-circuit fault can be effectively diagnosed and located with the proposed method.

2.16 A FAULT-DIAGNOSIS AND TOLERANT CONTROL TECHNIQUE FOR FIVE-LEVEL CASCADED H-BRIDGE INVERTERS

A fault-detection technique based on total harmonic distortion and a normalized output voltage factor is presented. Also discussed is a fault-isolation technique based on reduction of amplitude modulation, which results in a fault-tolerant inverter. In [16] the level-shifted pulse-width modulation (LSPWM) technique is used for switching operation. LSPWM is most suitable for MLIs that

require less computational effort. The presented fault-diagnosis technique can be implemented on MLI-based drives, grid-connected operations etc. without any major changes. The simulation and hardware results are presented to validate the proposed fault-detection and fault-tolerant control technique at different power factors.

2.17 A DETECTION METHOD FOR AN OPEN-SWITCH FAULT IN CASCADED H-BRIDGE MULTILEVEL INVERTERS

This paper [17] presents a detecting method for an open-switch fault in cascaded H-bridge multilevel inverters. Recently, cascaded multilevel inverters are widely used in high voltage and power systems. Therefore, the reliability of this topology has become an important factor. To improve the reliability, detecting methods for an open-switch fault are required. The proposed method is based on the current path analysis according to zero voltage switching states when the open-switch fault occurs. The location of both faulty cells and switches can be detected by the current waveform and zero voltage switching states. This method is applicable when using the level-shifted PWM that is commonly used for a modulation method in cascaded H-bridge multilevel inverters. The performance of proposed fault detecting method is verified by simulation results.

2.18 OPEN- AND SHORT-CIRCUIT SWITCH FAULT DIAGNOSIS FOR NON-ISOLATED DC-DC CONVERTERS USING FIELD PROGRAMMABLE GATE ARRAY

The aim of this study is to present a fast yet robust method for fault diagnosis in non-isolated DC-DC converters. Fault detection is based on time and current criteria which observe the slope of the inductor current over the time. Particularly in [18], it is realized by using a hybrid structure via coordinated operation of two fault detection subsystems that work in parallel. No additional sensors, which increase system cost and reduce reliability, are required for this

detection method. For validation, computer simulations are first carried out. The proposed detection scheme is validated on a boost converter. Effects of input disturbances and the closed-loop control are also considered. In the experimental set-up, a Field Programmable Gate Array (FPGA) digital target is used for the implementation of the proposed method, to perform very fast switch fault detection. Results show that with the presented method, fault detection is robust, and can be done in a few microseconds.

CHAPTER 3

ANALYSIS OF H-BRIDGE INVERTER

3.1 CASCADED H-BRIDGE MULTILEVEL INVERTER

The concept of multilevel inverter is based on connecting H-bridge inverters in series to get a sinusoidal voltage output. One full-bridge is itself a three-level cascaded H-bridge multilevel inverter and every module added in cascade which extends the inverter with two voltage levels. Each full-bridge inverter can create three voltages V_{dc}, 0 and $-V_{dc}$. To change one level of voltage cascaded H-bridge multilevel inverter turns one switch ON and another switch OFF in one full bridgeinverter.

Switches S_1 and S_2 are turned ON, for $-V_{dc}$, the switches S_3 and S_4 are turned OFF. When there is no current following through the full-bridge, then 0 voltage level is achieved [3, 4]. The output voltage in each bridge is the summation of the voltage that is generated by each cell. The number of output voltage levels are 2n + 1, where n is the number of cells. The cascaded H-bridge multilevel inverter is capable of producing the total voltage source magnitude in both positive and negative half cycles, while many other topologies can only produce half the total DC-bus voltage source magnitude. Full-bridge inverter that is connected in series can contribute with the same voltage, thus meets topology. There is possibility to charge every module in a cascaded H-bridge multilevel inverter with different voltages. There are two full-bridge inverters connected in series for obtaining five different output voltage levels, $-2\ V_{dc}$, V_{dc}, $0 - V_{dc}$ and $+2\ V_{dc}$. The advantages of this type of multilevel inverter are that it needs less number of components comparative to the diode clamped or the flying capacitor. However, the number of sources is higher, for the phase-leg to be able to create a number if m voltage level and switches 2 * (m − 1) [1, 2, 4].

3.2 MODULATION TECHNIQUES FOR MULTILEVEL INVERTER

Multilevel inverters have different modulation techniques for obtaining a better output voltage response with minimum harmonic distortions. There are basically two groups of methods: modulation with fundamental switching frequency or highswitching frequency pulse width modulation (PWM).

3.2.1 Pulse Width Modulation Techniques

A multilevel pulse width modulation method uses high switching frequency carrier waves in comparison to the reference waves to generate asinusoidal output wave as such in the two-level PWM case. To reduce harmonic distortions in the output voltage waveform, phase-shifting techniques are used thecarrier-based pulse width modulation techniques can be broadly classified into:

- Phase-shifted modulation
- Level-shifted modulation

In both modulation techniques, for an m-level inverter, (m-1) triangular carrier waves are required and all the carrier waves should have the same frequency and same peak-to-peak magnitude.

3.2.2 Phase Disposition Pulse Width Modulation:

In phase disposition modulation technique, all the triangular carriers are in phase and are arranged one over the other. These arranged triangular carriers are compared with reference wave to obtain the pulses for the multilevel inverter switches. This technique is generally accepted as the method that creates the lowest harmonic distortion in line-to-line voltage.

3.3 OPERATING MODES OF FIVE-LEVEL CASCADED H-BRIDGE MULTILEVEL INVERTER

- **Mode 1**

The operating mode for getting output voltage of +2 V_{dc}. In this mode, switches S_1, S_2, S_5 and S_6 are ON and all the other switches S_3, S_4, S_7 and S_8 are OFF. Fig. 3.1 shows the operating mode for getting output voltage of +2 V_{dc}.

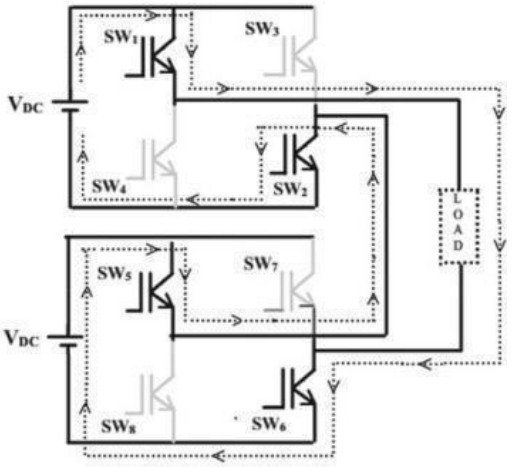


Fig. 3.1 Operating mode for getting output voltage of +2 V_{dc}

- **Mode 2**

The operating mode for getting output voltage of + V_{dc} In this mode, switches S_1, S_2, S_8 and S_6 are ON and all the other switches S_3, S_4, S_7 and S_5 are OFF. Fig. 3.2 shows operating mode for getting output voltage of + $V_{dc.}$

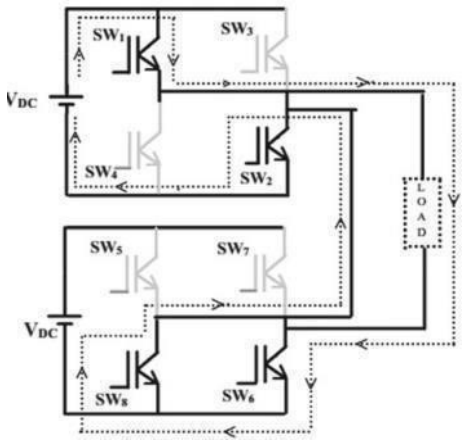


Fig. 3.2 Operating mode for getting output voltage of + V_{dc}

· **Mode 3**

The operating mode for getting output voltage of zero. The lower-leg switches are triggered; hence, there will no flow of current in the power circuit. Fig. 3.3 shows operating mode for getting output voltage of zero.

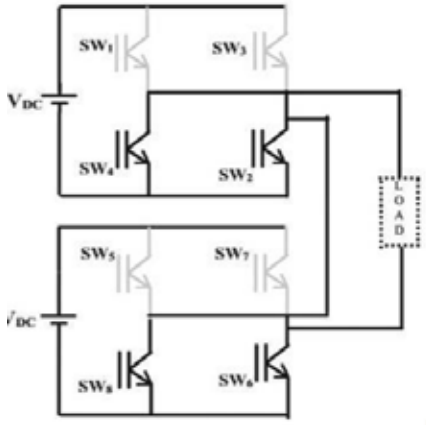


Fig. 3.3 Operating mode for getting output voltage of zero

· **Mode 4**

The operating mode for getting output voltage of − V_{dc} In this mode,switches S_3, S_4, S_8 and S_6 are ON and all the other switches S_1, S_2, S_7 and S_5 are OFF. The flow of current is opposite to the load current. Fig. 3.4 shows operating mode for getting output voltage of -V_{dc}.

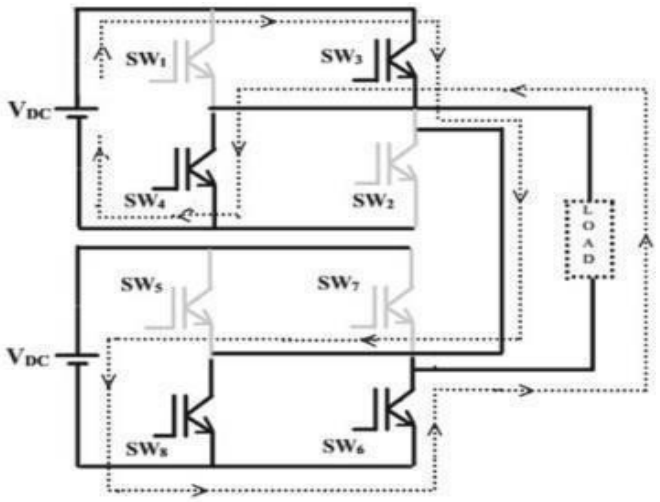


Fig. 3.4 Operating mode for getting output voltage of – V_{dc}

- **Mode 5**

The operating mode for getting output voltage of $-2V_{dc}$ in this mode, switches S_3, S_4, S_8 and S_7 are ON and all the other switches S_1, S_2, S_6 and S_5 are OFF. The flow of current is opposite to the load current. Fig. 3.5 shows the operating mode for getting output voltage of $-2V_{dc}$.

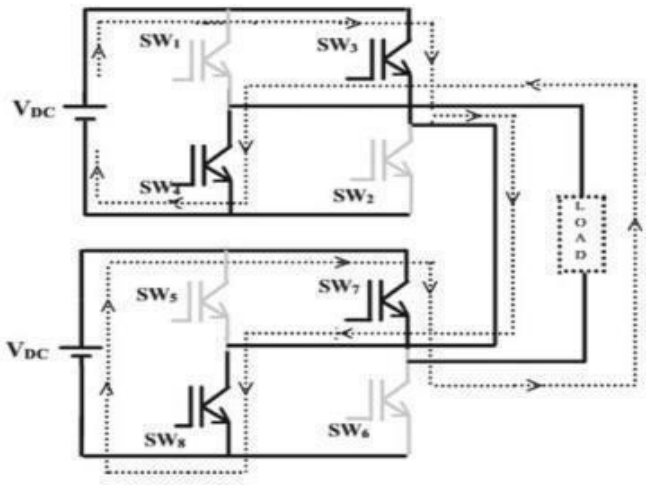


Fig. 3.5 Operating mode for getting output voltage of −2Vdc

3.4 SIMULATION MODEL OF FIVE-LEVEL H-BRIDGE INVERTER USING MATLAB

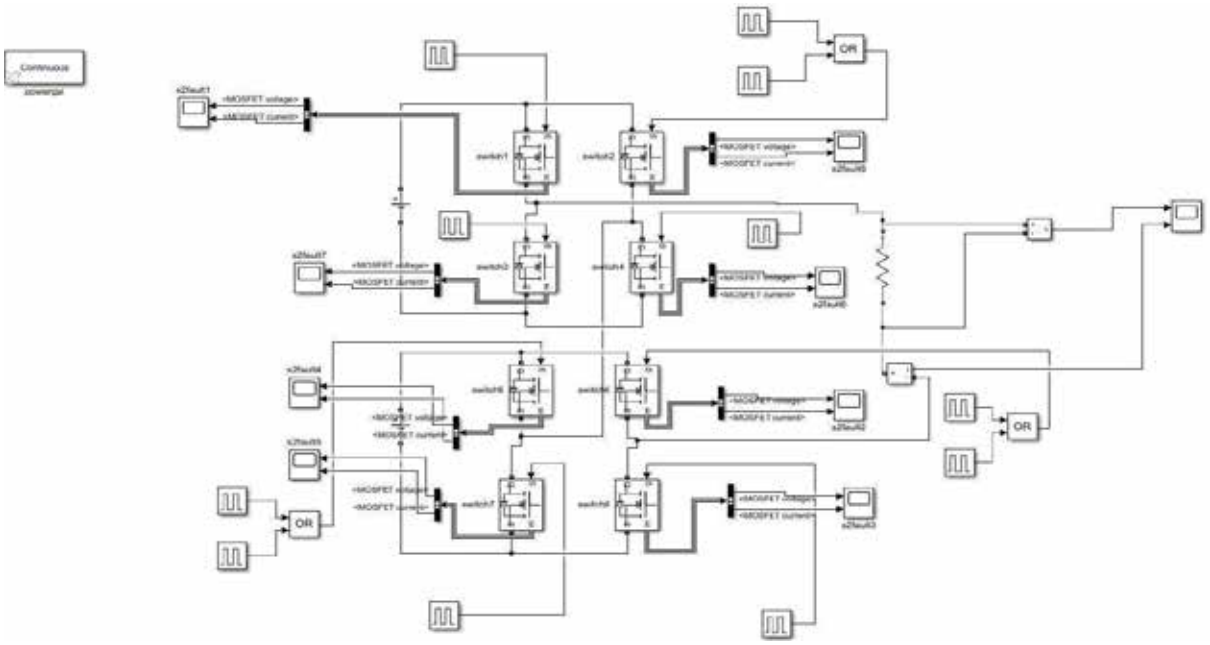


Fig. 3.6 Five-level H-bridge inverter

Fig. 3.6 shows the simulation diagram of five-level cascaded H-bridge inverter

Table 3.1 Switching Sequence

Voltage Levels / Switches	0	V	2V	V	0	-V	-2V	-V
S_1	1	1	1	1	1	0	0	0
S_2	1	0	0	0	1	1	1	1
S_3	0	0	0	0	0	1	1	1
S_4	0	1	1	1	0	0	0	0
S_5	1	1	1	1	1	1	0	1
S_6	1	1	0	1	1	1	1	1
S_7	0	0	0	0	0	0	1	0
S_8	0	0	1	0	0	0	0	0

Table 3.2 Pulse width modulation with delay

Switches	Delay	Pulse Width (P_w)	OR Operation	Delay	Pulse Width
S_1	(0.02/8) *0	(5/8) *100	No	NA	NA
S_2	(0.02/8) *0	(1/8) *100	Yes	(0.02/8) *4	P_w= (4/8) *100
S_3	(0.02/8) *5	(3/8) *100	No	NA	NA
S_4	(0.02/8) *1	(3/8) *100	No	NA	NA
S_5	(0.02/8) *0	(6/8) *100	Yes	(0.02/8) *7	P_w= (1/8) *100
S_6	(0.02/8) *0	(2/8) *100	Yes	(0.02/8) *3	P_w= (5/8) *100
S_7	(0.02/8) *6	(1/8) *100	No	NA	NA

The Table 3.1 and Table 3.2 presents the switching sequence and pulse width modulation with delay.

3.5 OUTPUT WAVEFORM

3.5.1 Output of Cascaded H-Bridge Inverter

The below Fig. 3.7 presents the output waveform of five-level cascaded H-bridge inverter without any open switch fault under normal condition.

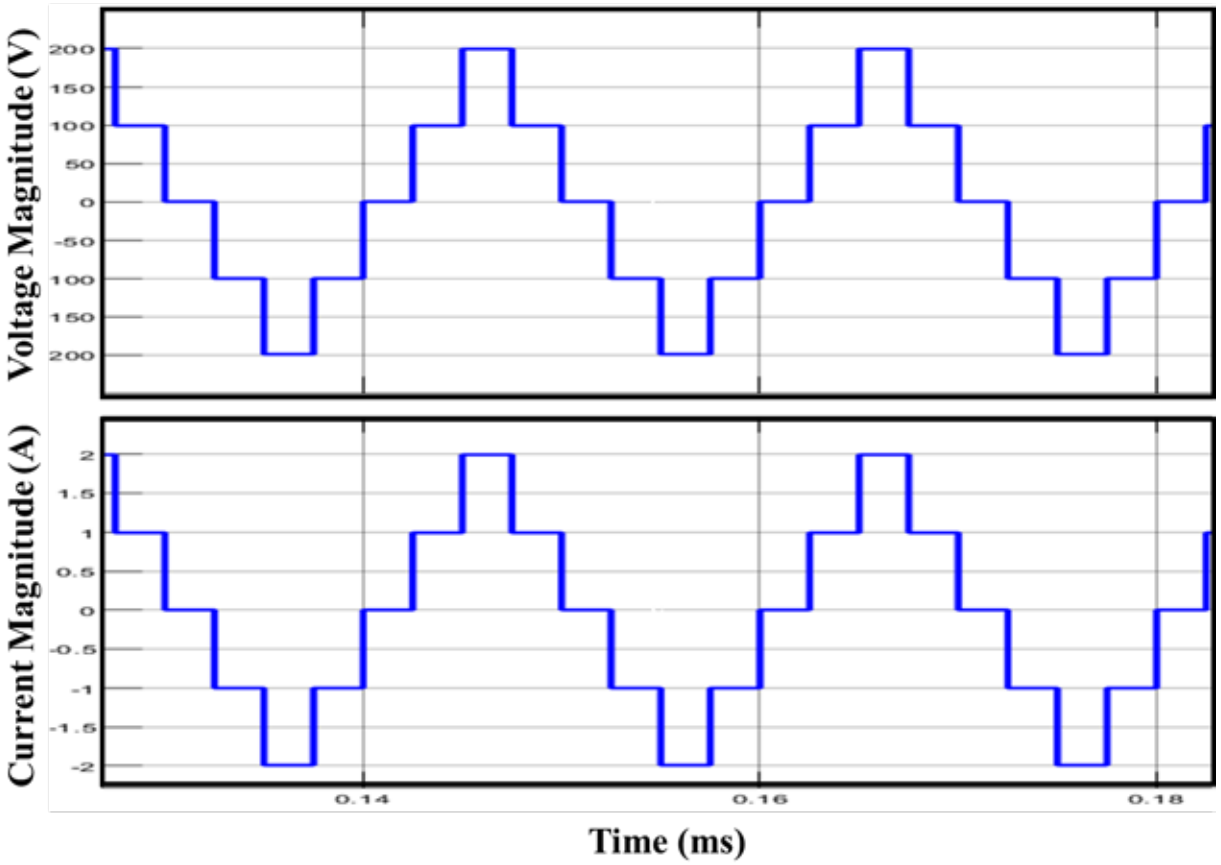


Fig. 3.7 Output waveform without fault

3.5.2 Output of Cascaded H-Bridge Inverter Under Fault State

Here manually switch S_1 is under fault condition and Fig. 3.8 show the output waveform of the cascaded h-bridge inverter this waveform makes it simple to identify the faulted switch

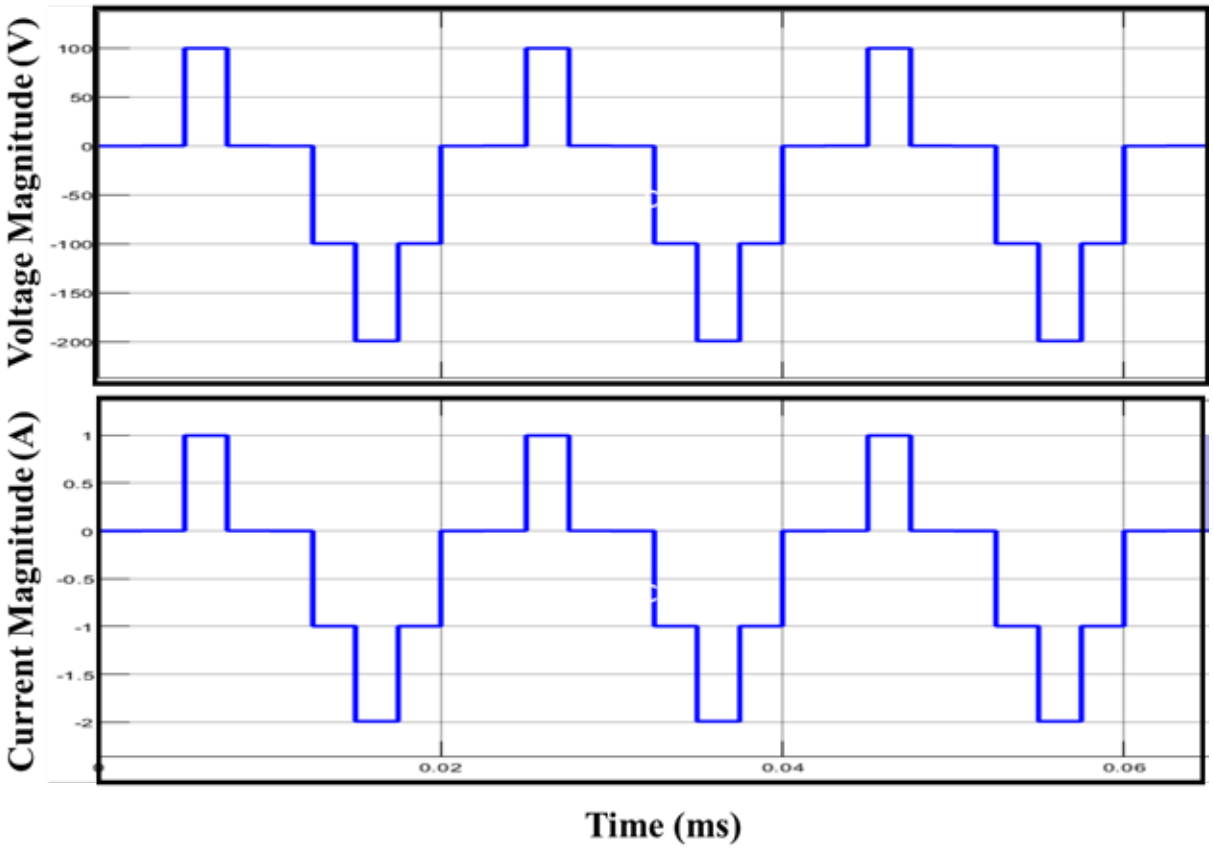


Fig. 3.8 Output waveform under switch S_1 in fault condition

Here manually switch S_1, S_3 is under fault condition and Fig. 3.9 show the output waveform.

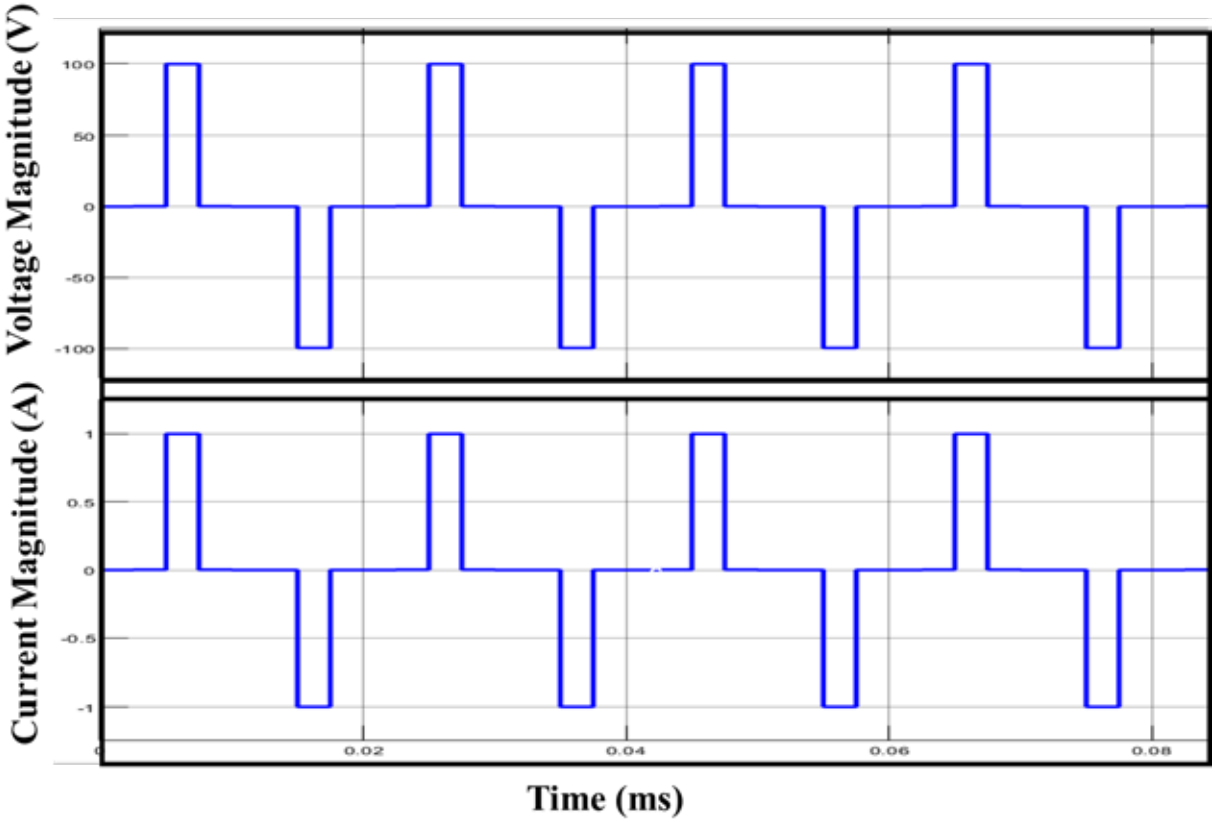


Fig. 3.9 Output waveform under switches S_1, S_3 in fault condition

- A typical Output of a five-level H-Bridge inverter is presented in Fig. 3.7.

- This waveform resembles the correct working of all the switches in the H- Bridge. The variations in this structure are observed and given as the input for the classifier to build a network structural framework to predict the given output.

CHAPTER 4
IMPLEMENTATION OF MACHINE LEARNING ALGORITHM

4.1 INTRODUCTION TO MACHINE LEARNING

The term Machine Learning was coined by Arthur Samuel 1959, an American pioneer in the field of computer gaming and artificial intelligence, and stated that “it gives computers the ability to learn without being explicitly programmed”. And in 1997,Tom Mitchell gave a “well-posed” mathematical and relational definition as shown in Fig. 4.1.

Machine learning is a growing technology which enables computers to learn automatically from past data. Machine learning uses various algorithms for building mathematical models and making predictions using historical data or information**.** Currently, it is being used for various tasks such as image recognition**,** speech recognition**,** email filtering**,** Facebook auto-tagging**,** recommender system, and many more. Machine Learning is said as a subset of artificial intelligence that is mainly concerned with the development of algorithms which allow a computer to learn from the data and past experiences on their own.

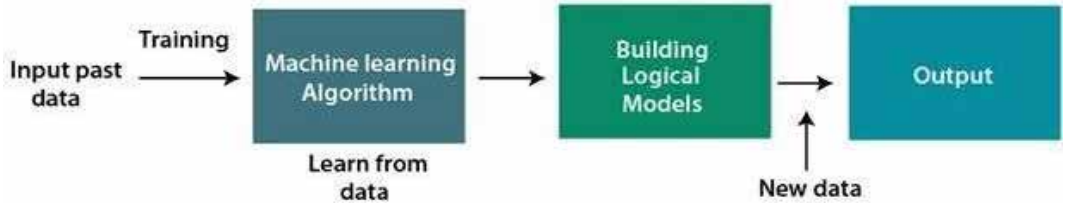


Fig. 4.1 Machine Learning process

A machine learning system learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it. The accuracy of predicted output depends upon the amount of data, as the huge

amount of data helps to build a better model which predicts the output more accurately. where we need to perform some predictions, so instead of writing a code for it, we justneed to feed the data to generic algorithms, and with the help of these algorithms, machine builds the logic as per the data and predict the output.

4.1.1 Need for Machine Learning

The need for machine learning is increasing day by day. The reason behind the need for machine learning is that it is capable of doing tasks that are too complex for a person to implement directly. As a human, we have some limitations as we cannotaccess the huge amount of data manually, so for this, we need some computer systems and here comes the machine learning to make things easy for us.

Machine learning algorithms by providing them the huge amount of data and let them explore the data, construct the models, and predict the required output automatically. The performance of the machine learning algorithm depends on the amount of data, and it can be determined by the cost function. With the help of machine learning, we can save both time and money.

The importance of machine learning can be easily understood by its use cases, Currently, machine learning is used in self-driving cars**,** cyber fraud detection**,** face recognition**,** and friend suggestion by Facebook, etc.

4.1.2 Classification of Machine Learning

- Supervised learning
- Unsupervised learning
- Reinforcement learning

4.1.3 Supervised Learning

Supervised learning is a type of machine learning method in which provided sample labelled data to the machine learning system in order to train it, and on that basis, it predicts the output.

The system creates a model using labelled data to understand the datasets and learn about each data, once the training and processing are done then we test the model by providing a sample data to check whether it is predicting the exact output or not.

The goal of supervised learning is to map input data with the output data. The supervised learning is based on supervision, and it is the same as when a student learns things in the supervision of the teacher. The example of supervised learning is spam filtering.

Supervised learning can be grouped further in two categories of algorithms:

- Classification
- Regression

4.1.4 Unsupervised Learning

The training is provided to the machine with the set of data that has not been labeled, classified, or categorized, and the algorithm needs to act on that data without any supervision. The goal of unsupervised learning is to restructure the input data into new features or a group of objects with similar pattern. The machine tries to find useful insights from the huge amount of data.

4.1.5 Reinforcement Learning

Reinforcement learning is a feedback-based learning method, in which a learning agent gets a reward for each right action and gets a penalty for each wrong action. The agent learns automatically with these feedbacks and improves its performance. In reinforcement learning, the agent interacts with the environment and explores it. The goal of an agent is to get the most reward points, and hence, it improvesits performance.

4.2 TYPES OF ALGORITHMS

- Decision Tree Algorithm
- Naive Bayes Algorithm
- K-NN Algorithm
- K-Nearest Neighbor Algorithm

4.2.1 Decision Tree Algorithm

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where rules internal nodes represent the features of a dataset, branches represent the decision and each leaf node represents the outcome.

In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches The decisions or the test are performed on the basis of

features of the given dataset. It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure. In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm as shown in Fig. 4.2.

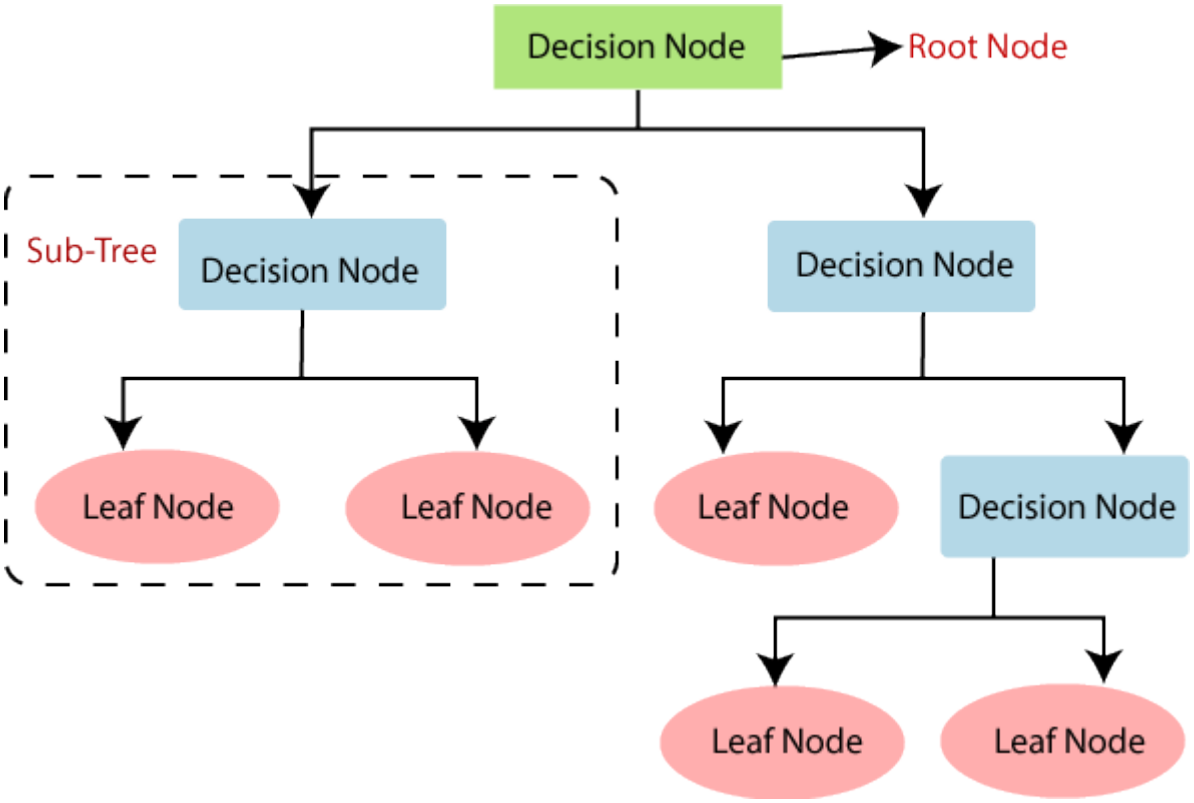


Fig. 4.2 Decision Tree Classification

There are various algorithms in Machine learning, so choosing the best algorithm for the given dataset and problem is the main point to remember while creating a machine learning model. Below are the two reasons for using the Decision tree:

Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand.

The logic behind the decision tree can be easily understood because it shows a tree-like structure.

Decision Tree Terminologies

- **Root Node:** Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneoussets.
- **Leaf Node:** Leaf nodes are the final output node, and the tree cannot besegregated further after getting a leaf node.
- **Splitting:** Splitting is the process of dividing the decision node/root nodeinto sub-nodes according to the given conditions.
- **Branch/Sub Tree:** A tree formed by splitting the tree.
- **Pruning:** Pruning is the process of removing the unwanted branches fromthe tree.
- **Parent/Child node:** The root node of the tree is called the parent node, andother nodes are called the child nodes.

Implementation Steps of Decision Tree Algorithm

In a decision tree, for predicting the class of the given dataset, the algorithm starts from the root node of the tree. This algorithm compares the values of root attribute with the record (real dataset) attribute and, based on the comparison, follows the branchand jumps to the next node.

For the next node, the algorithm again compares the attribute value with the other sub-nodes and move further. It continues the process until it reaches the leaf node of the tree.

Step-1: Begin the tree with the root node, says S, which contains the complete dataset.

Step-2: Find the best attribute in the dataset using **Attribute Selection Measure (ASM).**

Step-3: Divide the S into subsets that contains possible values for the best attributes.

Step-4: Generate the decision tree node, which contains the best attribute.

Step-5: Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

4.2.2 Naive Bayes Algorithm

Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems. It is mainly used in text classification that includes a high-dimensional training dataset. Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions. It is a probabilistic classifier, which means it predicts on the basis of the probability of an object. Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

Baye's Theorem

Bayes' theorem is also known as **Bayes' Rule** or **Bayes' law**, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability. The formula for Bayes' theorem is given in Fig. 4.3. The Navies bayes classifier is shown in Fig. 4.4.

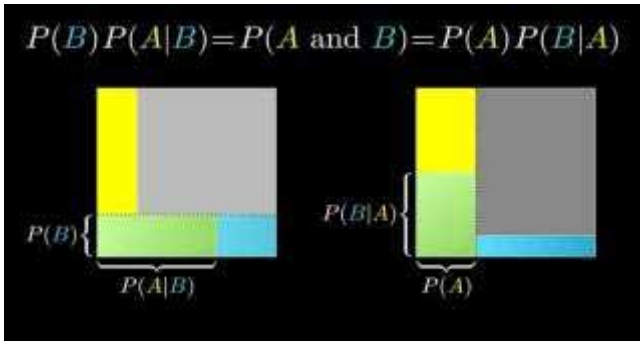


Fig. 4.3 Bayes Theorem

P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.

P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.

P(A) is Prior Probability: Probability of hypothesis before observing the evidence.

P(B) is Marginal Probability: Probability of Evidence.

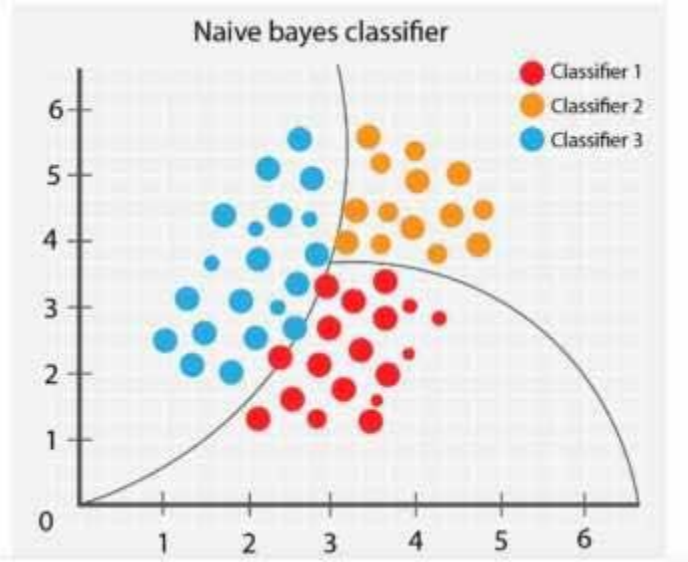


Fig. 4.4 Navies bayes classifier

4.2.3 K-NN Algorithm

K-NN is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process ofcombining multiple classifiers to solve a complex problem and to improve the performance of the model.

"Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." The Random Forest algorithm is shown in Fig. 4.5. Instead of relying on one decision tree, the Random Forest takes the prediction from each tree and based on the majority votes of predictions, andit predicts the final output.

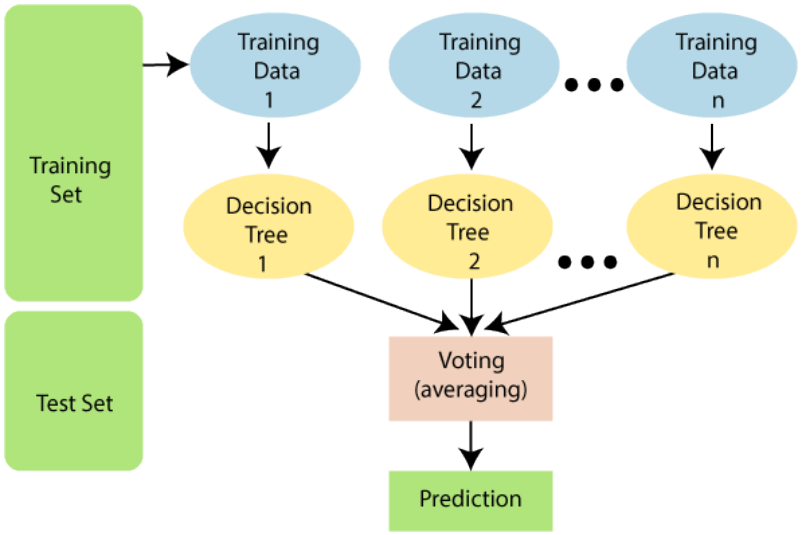


Fig. 4.5 Random Forest Algorithm

The Random Forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output. Therefore, below are two assumptions for a better Random Forest classifier:

Implementation steps of Random Forest

Random Forest works in two-phase first is to create the Random Forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

The working process can be explained in the below steps and diagram:

Step-1: Select random K data points from the training set.

Step-2: Build the decision trees associated with the selected data points (Subsets).

Step-3: Choose the number N for decision trees that you want to build.

Step-4: Repeat Step 1 & 2.

Step-5: For new data points, find the predictions of each decision tree, and assign thenew data points to the category that wins the majority votes.

4.2.4 K-Nearest Neighbor Algorithm

- K-Nearest Neighbor is one of the simplest Machine Learning algorithms basedon Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and availablecases and put the new case into the category that is most similar to the availablecategories.
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K-NN algorithm.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.

- K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.
- It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- K-NN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

There are two categories, i.e., Category A and Category B, and we have a newdata point x1, so this data point will lie in which of these categories shown in Fig. 4.6. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset.

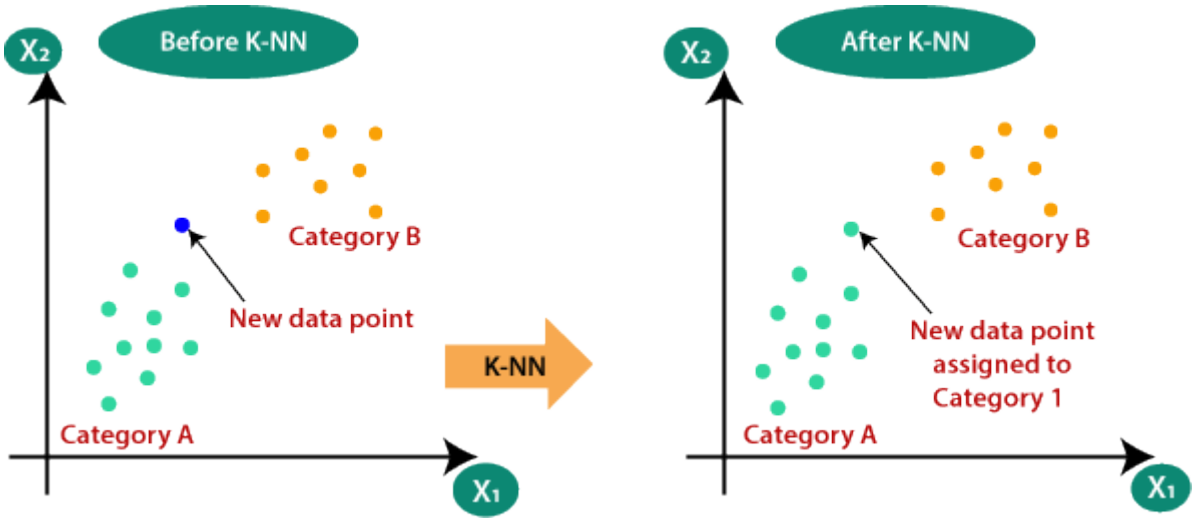


Fig. 4.6 K-NN Process

Implementation steps of K-NN

The K-NN working can be explained on the basis of the below algorithm:

Step-1: Select the number K of the neighbors

Step-2: Calculate the Euclidean distance of **K number of neighbors**

Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.

Step-4: Among these k neighbors, count the number of the data points in eachcategory.

Step-5: Assign the new data points to that category for which the number of theneighbor is maximum.

Step-6: Our model is ready.

CHAPTER 5

HARDWARE ARRANGEMENT FOR FAULT DETECTION SETUP

5.1 HARDWARE SETUP

The hardware setup includes regulated power supply, DSO, Arduino board with software program then switches of N-channel and P-channel MOSFET and optocoupler is act as driver circuit this shown Fig. 5.1.

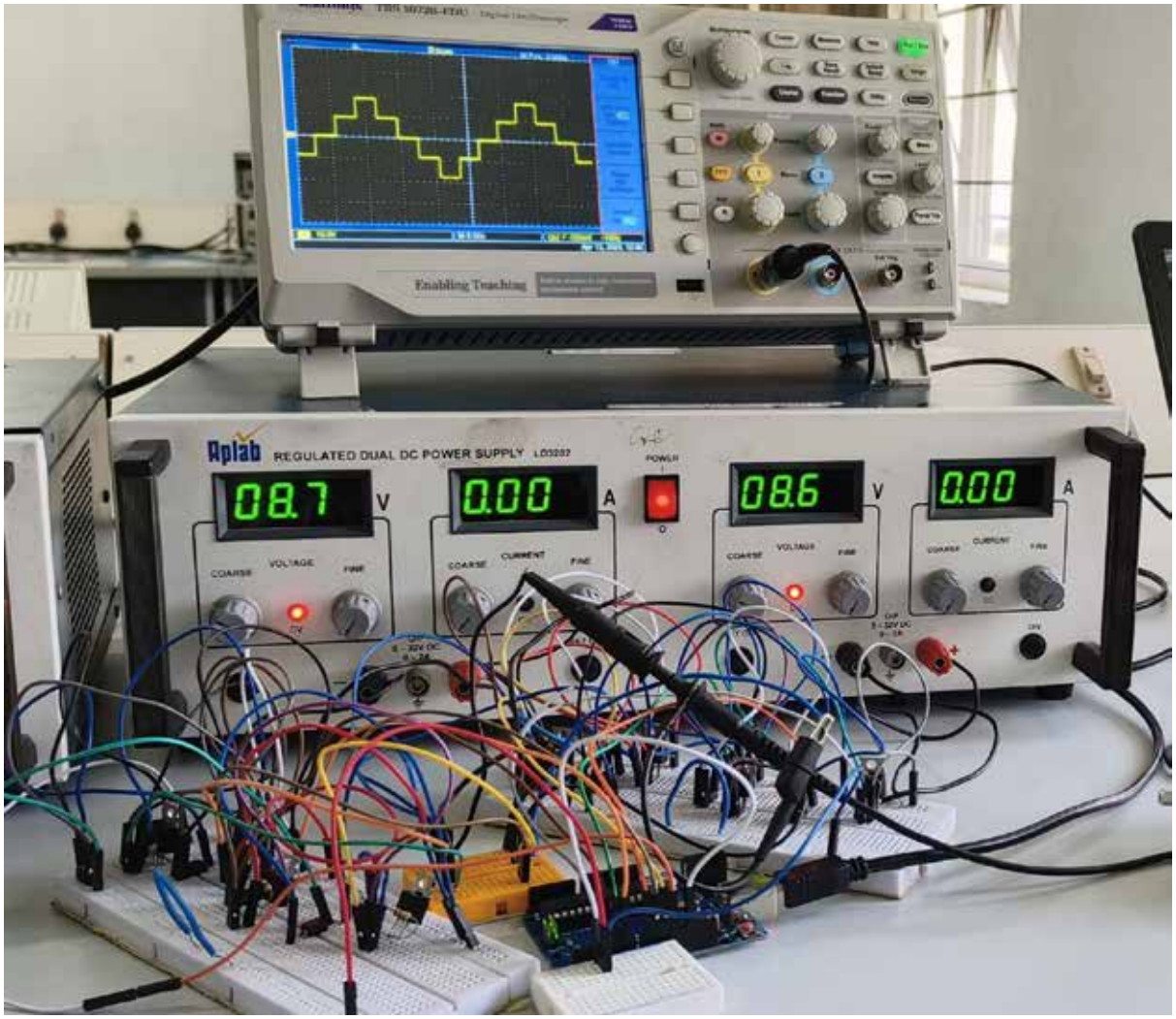


Fig. 5.1 Hardware Setup

5.2 TROUBLESHOOTING SETUP

In real time fault detection, there will be controlling signals given tothe gate of all the switches being used. The way of gate signals given in the particular system is based on the design and also there may be fault in the gate signals. So, the fault in the system is detached from the controller which provides the gate signal. So, the fault detection algorithm is developed using a symmetric type of cascaded H-bridge inverter the voltage levels of the 2 H- bridges are verified to be the same. Hence, a protection diode is attached near thenegative terminal of voltage supply of the two supplies to prevent from short circuit. Therefore, with respect to the pcb tracks the first thing can be done are,

Visual Inspection: The first step in detecting PCB track faults is to conduct a visual inspection. A visual inspection can detect damage to the PCB tracks caused by mechanical stress, corrosion, or other environmental factors. followed by the **Continuity Testing:** Continuity testing is a common method for detecting open circuits and short circuits in PCB tracks. A continuity tester, such as a multimeter, can be used to check if there is a continuous path between two points on the PCB tracks.

If the switches are all correct then the pcb board components like resistors are checked for the faults using the multimeter.

Then the fault detection will be done on the driver circuit as the There are several faults that can occur in the driver circuit of a MOSFET, including:

1. Driver IC failure: If the driver IC fails, the MOSFET may not receive the necessary gate voltage, which can lead to a loss of control or short-circuit. This can be identified by checking the output voltage of the driver IC using a multimeter or oscilloscope.

2. Resistor failure: If a resistor in the driver circuit fails, the MOSFET may not receive the correct voltage, which can lead to a loss of control or short-circuit. This can be identified by checking the resistance of the resistor using a multimeter.

3. Capacitor failure: If a capacitor in the driver circuit fails, the MOSFET may not receive the correct voltage or may receive a voltage spike, which can lead to a loss of control or short-circuit. This can be identified by checking the capacitance of the capacitor using a capacitance meter or oscilloscope.

4. Wiring faults: If there are any wiring faults in the driver circuit, such as broken or loose connections, the MOSFET may not receive the correct voltage, which can lead to a loss of control or short-circuit. This can be identified by checking the continuity of the wiring using a multimeter.

To track and rectify faults in the driver circuit of a MOSFET, the following steps can be taken:

1. Identify the fault: Use the methods mentioned above to identify the specific fault in the driver circuit.

2. Replace the faulty component: Once the fault has been identified, replace the faulty component. This may involve desoldering the faulty component and soldering in a new component.

3. Check the circuit: After replacing the faulty component, check the circuit to ensure that the fault has been rectified. This can be done by checking the output

voltage using a multimeter or oscilloscope, or by testing the MOSFET in a test circuit.

4. Test the system: Once the driver circuit has been repaired, test the system to ensure that it is functioning correctly and that the MOSFET is receiving the correct gate voltage. This can be done by connecting the system to a load and measuring the output voltage and current using a multimeter or oscilloscope. It is important to follow proper safety procedures when working with MOSFETs and driver circuits, and to ensure that the equipment being used is appropriate for the task.

Choosing the right heat sink for a MOSFET is important to ensure that the MOSFET operates within its safe operating temperature range. Here are some general guidelines for selecting a heat sink for a MOSFET:

1. Determine the power dissipation: Calculate the power dissipation of the MOSFET, which is the product of the drain-source voltage drop, the drain current, and the on-state resistance. This will give you an idea of the amount of heat that needs to be dissipated.

2. Choose a heat sink with appropriate thermal resistance: The thermal resistance of a heat sink is a measure of how effectively it can dissipate heat. Choose a heat sink with a thermal resistance that is appropriate for the power dissipation of the MOSFET. A lower thermal resistance will provide better heat dissipation, but may be more expensive.

3. Check the physical dimensions: Ensure that the heat sink is of an appropriate size to fit the MOSFET package and the PCB layout. The heat sink should have sufficient surface area to allow for efficient heat dissipation.

4. Consider the airflow: If the MOSFET is in an enclosed space with limited airflow, consider choosing a heat sink with fins or other features that promote

airflow. Alternatively, you may need to add a fan or other cooling mechanism to ensure adequate heat dissipation.

5. Mounting and attachment: Ensure that the heat sink can be securely attached to the MOSFET and the PCB. A good thermal contact between the MOSFET and the heat sink is essential for effective heat dissipation.

Some common types of heat sinks that are used for MOSFETs include extruded aluminum heat sinks, bonded fin heat sinks, and folded fin heat sinks. The choice of heat sink will depend on the specific requirements of the application and the available space and budget.

Power electronics is a crucial technology that plays a significant role in modern society. It involves the conversion, control, and conditioning of electrical power for various applications, such as renewable energy systems, electric vehicles, and industrial automation. One of the most important aspects of power electronics is the ability to efficiently and reliably convert DC power into AC power of different voltages and frequencies using power inverters. Multilevel inverters are a type of power inverter that can generate AC waveforms with high quality and low distortion. They use multiple power semiconductor switches to produce voltage steps that approximate sinusoidal waveforms. Multilevel inverters have several advantages over conventional two-level inverters, such as reduced harmonic content, improved voltage waveform quality, and lower electromagnetic interference. Among the various multilevel inverter topologies, the cascaded H-bridge multilevel inverter is a popular and promising design. It consists of several H-bridge inverter cells connected in series to produce a staircase waveform. The cascaded H-bridge multilevel inverter offers high modularity, scalability, and fault tolerance, making it suitable for a wide range of applications, such as renewable energy systems, motor drives, and power quality control. This project aims to design, simulate, and experimentally validate a five-level cascaded H-bridge multilevel inverter.

are to study the operating principles and characteristics of cascaded H-bridge multilevel inverters, to design a practical and efficient five-level cascaded H-bridge inverter system, to evaluate the performance of the system using simulation and experimental methods, and to analyze the results and draw conclusions. The project is expected to contribute to the advancement of multilevel inverter technology and its applications in the field of power electronics.

Multilevel inverters have gained significant attention in recent years due to their advantages over conventional two-level inverters. Various multilevel inverter topologies have been proposed in the literature, such as diode-clamped, flying capacitor, and cascaded H-bridge. Among these, cascaded H-bridge multilevel inverters have attracted considerable interest due to their modularity, scalability, and fault tolerance. The cascaded H-bridge multilevel inverter consists of several H-bridge inverter cells connected in series to produce a staircase waveform. The number of levels in the output waveform is determined by the number of H-bridge cells. The cascaded H-bridge topology has several advantages over other multilevel inverter topologies, such as reduced component count, easy control, and high efficiency. However, it also has some limitations, such as high cost and complexity of the control algorithm. The control of cascaded H-bridge multilevel inverters is a challenging task due to the large number of switches and the complex interaction between the cells. Several control techniques have been proposed in the literature, such as pulse-width modulation (PWM), space vector modulation (SVM), and model predictive control (MPC). PWM is the most commonly used technique, which involves switching the H- bridge cells in a synchronized manner to produce the desired output waveform. The performance of cascaded H-bridge multilevel inverters has been extensively studied in the literature. Several research works have investigated the harmonic content, voltage waveform quality, and efficiency of cascaded H-bridge

inverters under various operating conditions. It has been found that cascaded H-bridge multilevel inverters can achieve high-quality output waveforms with low harmonic distortion and high efficiency. However, their performance is affected by various factors, such as modulation strategy, DC voltage sources, and load characteristics. The applications of cascaded H-bridge multilevel inverters are diverse and widespread. They are used in various fields, such as renewable energy systems, motor drives, power quality control, and HVDC transmission. Several research works have investigated the use of cascaded H-bridge multilevel inverters in these applications and demonstrated their advantages over other multilevel inverter topologies. However, the practical implementation of cascaded H-bridge multilevel inverters still faces some challenges, such as the cost of the power semiconductor switches and the complexity of the control algorithm.

5.3 HARDWARE USED

5.3.1 ARDUINO UNO

The Arduino UNO is a micro-controller board based on the ATmega328. It has 14 digital Input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an

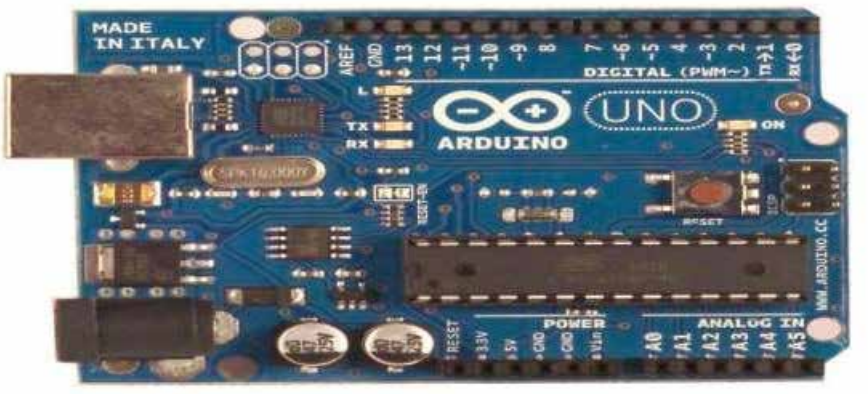


Fig. 5.2 Arduino UNO Micro-controller Board

ICSP header, and a reset button. It contains everything needed to support the micro-controller simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. The UNO and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions. Fig. 5.2 shows Arduino UNO micro-controller board.

TECHNICAL SPECIFICATION

· Microcontroller AT mega: 328

· Operating Voltage: 5 V

· Input Voltage (recommended): 7 - 12 V

· Input Voltage (limits): 6 - 20 V

· Digital I/O Pins: 14

· Analog Input Pins: 6

· DC Current per I/O Pin: 40 mA

· DC Current for 3.3 V Pin: 50 mA

· Flash Memory: 32 KB

· KB used: Boot loader

· SRAM: 2 KB

· EEPROM: 1 KB

· Clock Speed: 16 MHz

5.3.2 MOSFET

The metal–oxide–semiconductor field-effect transistor (MOSFET, MOS- FET, or MOS FET) is a type of field-effect transistor (FET), most commonly fabricated by the controlled oxidation of silicon. It has an insulated gate, the voltage of which determines the conductivity of the device. This ability to change conductivity with the amount of applied voltage can be used for amplifying or switching electronic signals. A metal-insulator-semiconductor field-effect transistor (MISFET) is a term almost synonymous with MOSFET. Another synonym is IGFET for Insulated-Gate Field-Effect Transistor. Two power MOSFETs in D2PAK surface-mount packages. Operating as switches, each of these components can sustain a blocking voltage of 120 V in the off state, and can conduct a continuous current of 30 A in the on state, dissipating up to about 100 W and controlling a load of over 2000 W. The main advantage of a MOSFET is that it requires almost no input current to control the load current, when compared with bipolar transistors (bipolar junction transistors/BJTs). Inan enhancement mode MOSFET, voltage applied to the gate terminal increases the conductivity of the device. In depletion mode transistors, voltage applied at the gate reduces the conductivity. The "metal" in the name MOSFET is sometimes a misnomer, because the gate material can be a layerof polysilicon (polycrystalline silicon). Similarly, "oxide" in the name can also be a misnomer, as different dielectric materials are used with the aim of obtainingstrong channels with smaller applied voltages. The MOSFET is by far the most common transistor in digital circuits, as billions may be included in a memory chip or microprocessor. Since MOSFETs can be made with either p-type or n- type semiconductors, complementary pairs of MOS transistors can be used to make switching circuits with very low power consumption, in the form of CMOSlogic.

Gate Drive Voltage: MOSFETs require a sufficient gate voltage to turn on fully. If the gate voltage is too low, the MOSFET may not turn on completely, which can result in increased power dissipation and reduced efficiency. If the gate voltage is too high, it can damage the MOSFET. Therefore, it is essential to ensure that the gate voltage is within the specified range for the MOSFET being used.

Gate Driver Circuit: In order to provide the necessary gate drive voltage and current to the MOSFET, a gate driver circuit is often required. A gate driver circuit must be carefully designed to ensure that it can provide the necessary voltage and current while minimizing switching losses and EMI.

Parasitic Capacitances: MOSFETs have inherent capacitances between the gate, drain, and source terminals. These parasitic capacitances can cause problems in high-frequency circuits, such as switching power supplies, by slowing down the switching speed and reducing efficiency.

Protection Circuitry: MOSFETs can be sensitive to overvoltage, overcurrent, and electrostatic discharge (ESD). It is important to include appropriate protection circuitry in the circuit design to prevent damage to the MOSFET.

In summary, connecting MOSFETs requires careful consideration of several factors, including gate drive voltage, gate driver circuit design, parasitic capacitances, thermal management, and protection circuitry. Proper attention to these factors can help ensure that the MOSFET operates reliably and efficiently in the intended circuit.

MODES OF OPERATION

The operation of a MOSFET can be separated into three different modes, depending on the voltages at the terminals. In the following discussion, a simplified algebraic model is used. Modern MOSFET characteristics are more complex than the algebraic model presented here. For an enhancement-mode, n-channel MOSFET, the three operational modes are:

Cutoff, subthreshold, and weak-inversion mode

When $V_{GS} < V_{th}$:

where is gate-to-source bias and is the threshold voltage of the device.

N-CHANNEL MOSFET(IRFZ44N)

The IRFZ44N is an N-channel MOSFET (Metal-Oxide-Semiconductor Field-Effect Transistor) that can be used in a variety of applications such as switching power supplies, motor control circuits, and lighting control circuits. It is designed to handle high power and voltage levels, and has a maximum drain-source voltage of 55 V and a maximum drain current of 49 A.The N-channel MOSFET is a type of transistor that is controlled by a voltage applied to its gate terminal. The IRFZ44N has three terminals: the source, the drain, and the gate. The source terminal is connected to the negative or ground side of the circuit, while the drain terminal is connected to the positive or load side of the circuit. The gate terminal is connected to a control signal that determines the on/off state of the MOSFET. When a voltage is applied to the gate terminal, an electric field is created in the MOSFET's channel, which controls the flow of current between the source and drain terminals. In the on-state, the MOSFET has a low resistance, allowing current to flow through the load. In the off-state, the MOSFET has a high resistance, effectively blocking current flow. The IRFZ44N MOSFET has a low gate threshold voltage, which means it can be easily driven by microcontrollers and other digital logic circuits. However, it also has a relatively

high gate capacitance, which can cause delays and ringing in high-frequency switching applications. To mitigate this, it is important to choose a gate driver that is capable of providing sufficient current and voltage to drive the MOSFET quickly and reliably. It is important to refer to the manufacturer's datasheet for theIRFZ44N MOSFET to ensure proper use and performance of the device. The datasheet provides information on the device's electrical characteristics, recommended operating conditions, and application circuit examples.

P-CHANNEL MOSFET(IRF9540)

The IRF9540 is a P-channel MOSFET (Metal-Oxide-Semiconductor Field-Effect Transistor) that is commonly used in power management applications such as motor control, power supplies, and audio amplifiers. It is designed to handle high power and voltage levels, and has a maximum drain-source voltage of 100V and a maximum drain current of 23 A. The P-channel MOSFET is a type of transistor that is controlled by a voltage applied to its gate terminal. The IRF9540 has three terminals: the source, the drain, and the gate. The source terminal is connected to the positive or load side of the circuit, while the drain terminal is connected to the negative or ground side of the circuit. The gate terminal is connected to a control signal that determines the on/off state of the MOSFET.

In the off-state, the MOSFET has a low resistance, allowing current to flow through the load. In the on-state, the MOSFET has a high resistance, effectively blocking current flow. The IRF9540 MOSFET has a low gate threshold voltage, which means it can be easily driven by microcontrollers and other digital logic circuits. However, it also has a relatively high gate capacitance, which can cause delays and ringing in high-frequency switching applications. To mitigate this, it is important to choose a gate driver that is

capable of providing sufficient current and voltage to drive the MOSFET quickly and reliably. One advantage of P-channel MOSFETslike the IRF9540 is that they can be used in high-side switching applications, where the MOSFET is connected between the positive supply voltage and the load. In contrast, N-channel MOSFETs are typically used in low-side switching applications, where the MOSFET is connected between the negative supply voltage and ground. It is important to refer to the manufacturer's datasheet for the IRF9540 MOSFET to ensure proper use and performance of the device. The datasheet provides information on the device's electrical characteristics, recommended operating conditions, and application circuit examples.

5.3.3 OPTOCOUPLER (PC817)

An optocoupler, such as the PC817, is a device that electrically isolates two circuits by using light to transfer a signal between them. It consists of an LED and a phototransistor or photodiode, both of which are enclosed in a single package. The LED is connected to the input circuit and emits light when a signalis applied. This light is then detected by the phototransistor or photodiode, whichis connected to the output circuit as shown in Fig. 5.3 The output circuit responds to the input signal based on the amount of light received.

The PC817 is a popular optocoupler that can be used in a variety of applications, such as isolating high voltage signals from low voltage control circuits. It has a typical input current of 10 mA and a maximum output current of 50 mA. It can operate at a maximum voltage of 80V and has a maximum isolation voltage of 5,000 V_{rms}.

When using the PC817 in a circuit, it is important to follow the manufacturer's datasheet for proper operation and performance. This datasheet provides information on the device's electrical characteristics, recommended

operating conditions, and application circuit examples.

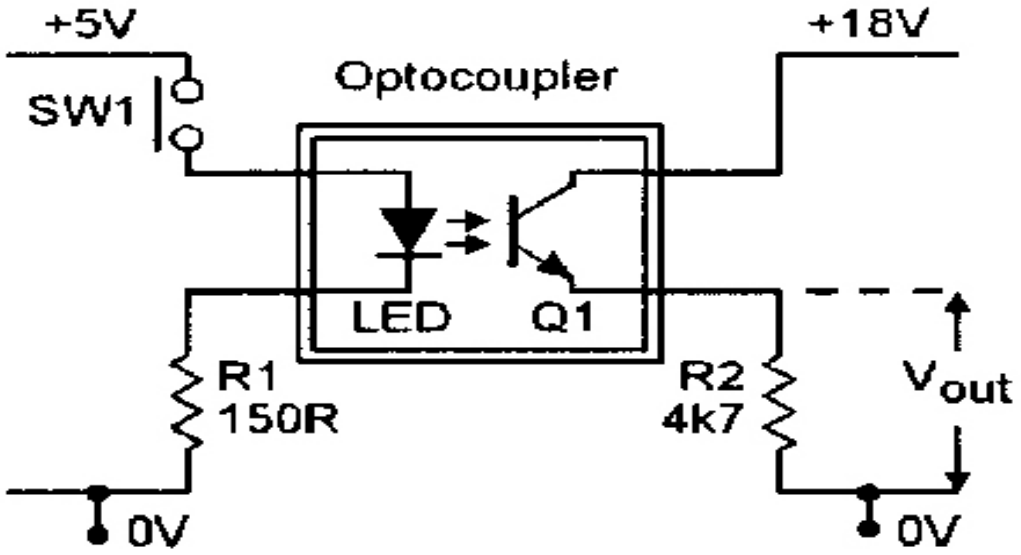


Fig. 5.3 Circuit diagram of Optocoupler

The PC817 optocoupler consists of an LED (Light Emitting Diode) and a phototransistor inside a single package. The LED is connected to the input circuit while the phototransistor is connected to the output circuit as shown in Fig. 5.4. When a voltage is applied to the input circuit, current flows through the resistor and the LED. The LED emits light that falls on the base of the phototransistor. The amount of lightreceived by the phototransistor depends on the input voltage level. If the input voltage is high, the LED emits more light, which results in a larger current flowing through the phototransistor. If the input voltage is low, the LED emits less light, resulting in a smaller current flowing through the phototransistor.

There are four configurations of optocouplers, the difference being the photosensitive device used. Photo-transistor and photo-Darlington are typically used in dc circuits, and photo-SCR and photo-TRIAC are used to control ac circuits. In the photo-transistor optocoupler, the transistor could either be PNP or NPN.

The term optocoupler and optoisolator are often used interchangeably, but there is a slight difference between the two. The distinguishing factor is the voltage difference expected between the input and the output.

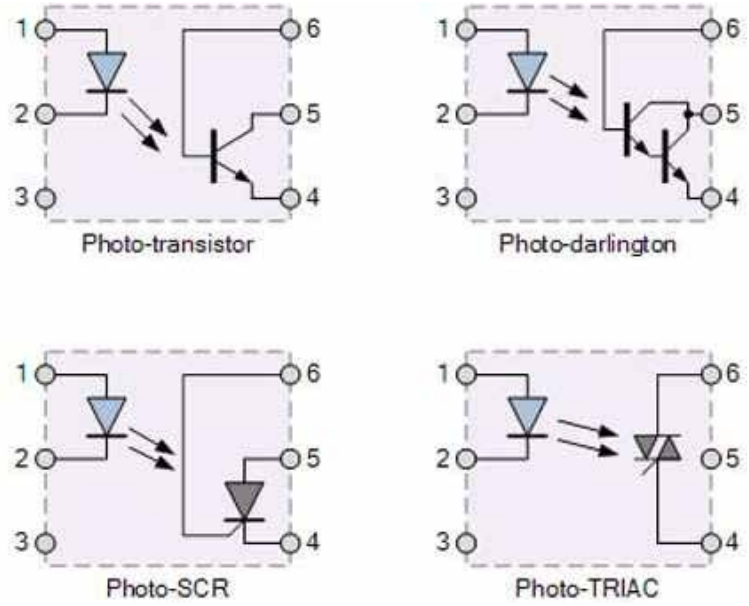


Fig. 5.4 Configuration of Optocoupler

CHAPTER 6

RESULT AND DISCUSSION

6.1 IMPLEMENTATION OF MACHINE LEARNING

This chapter explains the data collection and implementation of ML algorithms for fault detection in MLI. The details of the implementation are presented as follows;

6.1.1 Cleaning the Data

The values of various switches are Collected from MATLAB simulation and plotted in an Excel Sheet as shown in the Fig. 6.1. The blank spaces and redundancy data are Cleaned and the various (256) combinations Faulted Switch conditions are plotted.

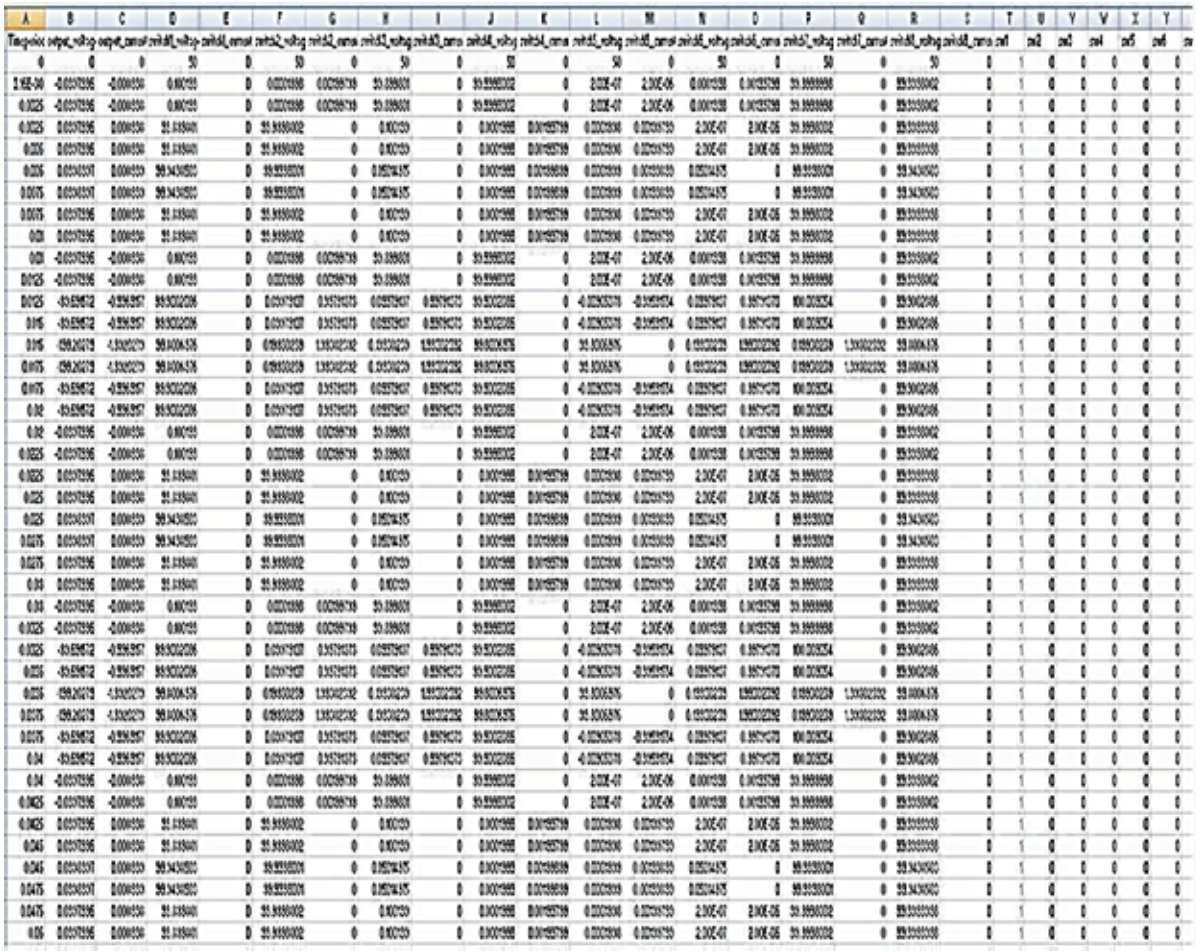
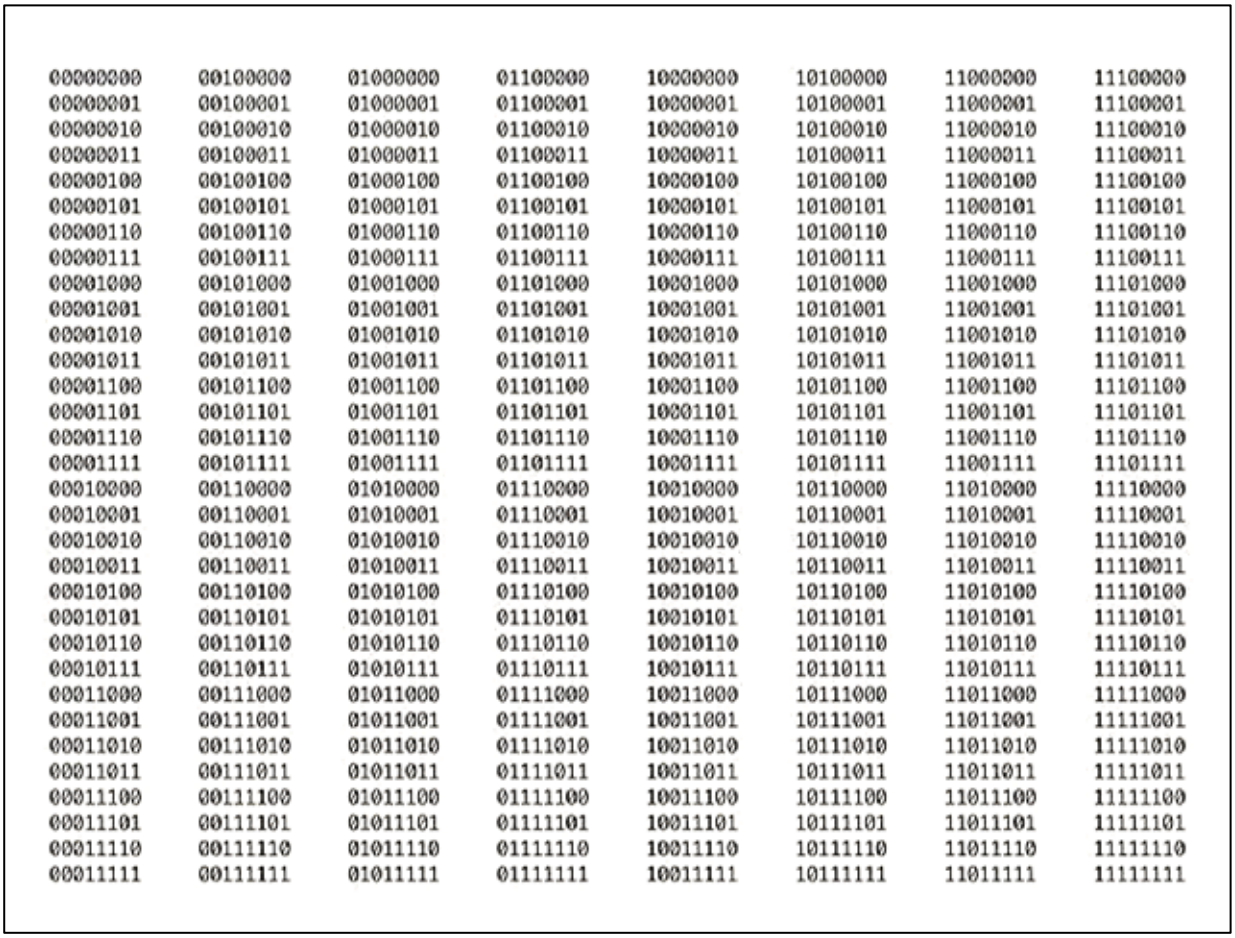


Fig. 6.1 Complete dataset for the fault detection



00000000	00100000	01000000	01100000	10000000	10100000	11000000	11100000
00000001	00100001	01000001	01100001	10000001	10100001	11000001	11100001
00000010	00100010	01000010	01100010	10000010	10100010	11000010	11100010
00000011	00100011	01000011	01100011	10000011	10100011	11000011	11100011
00000100	00100100	01000100	01100100	10000100	10100100	11000100	11100100
00000101	00100101	01000101	01100101	10000101	10100101	11000101	11100101
00000110	00100110	01000110	01100110	10000110	10100110	11000110	11100110
00000111	00100111	01000111	01100111	10000111	10100111	11000111	11100111
00001000	00101000	01001000	01101000	10001000	10101000	11001000	11101000
00001001	00101001	01001001	01101001	10001001	10101001	11001001	11101001
00001010	00101010	01001010	01101010	10001010	10101010	11001010	11101010
00001011	00101011	01001011	01101011	10001011	10101011	11001011	11101011
00001100	00101100	01001100	01101100	10001100	10101100	11001100	11101100
00001101	00101101	01001101	01101101	10001101	10101101	11001101	11101101
00001110	00101110	01001110	01101110	10001110	10101110	11001110	11101110
00001111	00101111	01001111	01101111	10001111	10101111	11001111	11101111
00010000	00110000	01010000	01110000	10010000	10110000	11010000	11110000
00010001	00110001	01010001	01110001	10010001	10110001	11010001	11110001
00010010	00110010	01010010	01110010	10010010	10110010	11010010	11110010
00010011	00110011	01010011	01110011	10010011	10110011	11010011	11110011
00010100	00110100	01010100	01110100	10010100	10110100	11010100	11110100
00010101	00110101	01010101	01110101	10010101	10110101	11010101	11110101
00010110	00110110	01010110	01110110	10010110	10110110	11010110	11110110
00010111	00110111	01010111	01110111	10010111	10110111	11010111	11110111
00011000	00111000	01011000	01111000	10011000	10111000	11011000	11111000
00011001	00111001	01011001	01111001	10011001	10111001	11011001	11111001
00011010	00111010	01011010	01111010	10011010	10111010	11011010	11111010
00011011	00111011	01011011	01111011	10011011	10111011	11011011	11111011
00011100	00111100	01011100	01111100	10011100	10111100	11011100	11111100
00011101	00111101	01011101	01111101	10011101	10111101	11011101	11111101
00011110	00111110	01011110	01111110	10011110	10111110	11011110	11111110
00011111	00111111	01011111	01111111	10011111	10111111	11011111	11111111

Fig. 6.2 Combination of switch failure

The Fig. 6.2 shows the combination of switches, each digit represents the switch number 1 to 8 from left to right and there are (256) combinations of data that has been extracted.

6.1.2 Importing the data

The Extracted features are cleaned and collected in a single Comma Separated Value (CSV) file. This CSV file is imported using “pd.read_csv('dataset.csv')” and stored in a Data frame “df”. The “df.head()” represents the sample of the data frame and the “df.columns” shows the columns present in the Data Frame. The type in the columns represents the nature of fault as “Open Circuit Fault” and the columns 'S_1', 'S_2', '$S_{3'}$,'S_4', 'S_5', 'S_6', 'S_7' and 'S_8' are given the value ‘1’ if there is a fault and value‘0’ if there is no fault which Exhibits the nature of the respected switch as shown in Fig. 6.3.

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

```
df = pd.read_csv('dataset.csv')
df.head()
```

	time	output_voltage	output_current	type	sw1	sw2	sw3	sw4	sw5	sw6	sw7	sw8	output
0	0.0000	0.0000	0.000000	op	1	0	0	0	0	0	0	1	op10000001
1	0.0000	-0.0998	-0.000998	op	1	0	0	0	0	0	0	1	op10000001
2	0.0025	-0.0998	-0.000998	op	1	0	0	0	0	0	0	1	op10000001
3	0.0025	0.0998	0.000998	op	1	0	0	0	0	0	0	1	op10000001
4	0.0050	0.0998	0.000998	op	1	0	0	0	0	0	0	1	op10000001

```
[ ] df.columns
```

```
Index(['time', 'output_voltage', 'output_current', 'type', 'sw1', 'sw2', 'sw3',
       'sw4', 'sw5', 'sw6', 'sw7', 'sw8', 'output'],
      dtype='object')
```

```
[ ] df_input=df[['time', 'output_voltage', 'output_current']]
```

Fig. 6.3 Data frame construction

The “output” column in the dataset represents the final output of the given 'time', 'output voltage' and 'output current' where the represents the time period, Output voltage and output current. import NumPy as np import pandas as pd

These two are basically used for implementing the Data frame accessing actions. The NumPy library contains multidimensional array and matrix data structures. It provides **ND array**, a homogeneous n-dimensional array object, with methods to efficiently operate on it. NumPy can be used to perform a wide variety of mathematical operations on arrays.

It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices and it supplies enormous library of high-level mathematical functions that operate on these arraysand matrices.

NumPy arrays are faster and more compact than Python lists. An array consumes less memory and is convenient to use. NumPy uses much less memory to store data and it provides a mechanism of specifying the data types.

Pandas Data Frame is two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns). A Data frame is a two-dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns. Pandas Data Frame consists of three principal components, the data, rows, and columns as shown in Fig. 6.4.

The Time period, Output voltage and output current are chosen as the inputparameters.

```
target1=df['sw1']
target2=df['sw2']
target3=df['sw3']
target4=df['sw4']
target5=df['sw5']
target6=df['sw6']
target7=df['sw7']
target8=df['sw8']
```

Fig. 6.4 Mapping outputs as targets

Separate models are build for Separate Switches predictions. To build the model thepredetermined outputs are passed to the classifier for mapping.

6.1.3 Split the data into training and testing sets

```
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(df_input,target1, test_size=0.2, random_state=42)
```

Fig. 6.5 Splitting Data into training and testing sets

The sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction as shown in Fig. 6.5. Here it is used for sorting the Data frame into TRAININGDATA and TESTING DATA where the x train and y_train are the Training dataset and x_test and y_test represents the Testing dataset. “target1“is the output representing variable.

6.1.4 Deciding the appropriate model and the appropriate split

It is mandatory to decide the perfect split at which the model gives the accuracy to the higher extent. the decision of choosing a model based on the accuracy is also very much important to achieve the right as shown in Fig. 6.6.

```
from sklearn.model_selection import train_test_split
i=0
for i in range(1,10):
  n=i*10
  print('\n\n ********** \n Train test split',100-n,'%',n,'%')
  x_train, x_test, y_train, y_test = train_test_split(df_input,target1, test_size=n, random_state=42)
  print('\n size of the data:',y_train.value_counts())
  from sklearn.tree import DecisionTreeClassifier
  from sklearn.naive_bayes import GaussianNB
  from sklearn.ensemble import RandomForestClassifier
  from sklearn.neighbors import KNeighborsClassifier
  from sklearn.neural_network import MLPClassifier
  print('\n\n 1. Decision tree*****')
  classifier=DecisionTreeClassifier()
  classifier.fit(x_train, y_train)
  print('Training_accuracy:',classifier.score(x_train,y_train))
  print('Testing_accuracy:',classifier.score(x_test,y_test))
  print('\n\n2. Naive Bayes*****')
  classifier=GaussianNB()
  classifier.fit(x_train, y_train)
  print('Training_accuracy:',classifier.score(x_train,y_train))
  print('Testing_accuracy:',classifier.score(x_test,y_test))
```

Fig. 6.6 Selecting correct algorithm and split

Using a for loop the Training accuracy and Testing accuracy are found for various models. If the Training accuracy and Testing accuracy are moreover close then it is not considered as the best split. The data has been split into 80% for training and 20% for testing. The accuracy of the classifier is shown in Fig. 6.7.

```
 Train test split 80 % 20 %

 size of the data: 1    89740
0    29760
Name: sw1, dtype: int64


 1. Decision tree*****
Training_accuracy: 0.8018744769874477
Testing_accuracy: 0.75


2. Naive Bayes*****
Training_accuracy: 0.7339916317991632
Testing_accuracy: 0.75


3.Random Forest *****
Training_accuracy: 0.8018744769874477
Testing_accuracy: 0.75


4. K-Nearest Neighbour *****
Training_accuracy: 0.7902092050209205
Testing_accuracy: 0.7


5. Multi-Layered Perceptron Neural Network *****
Training_accuracy: 0.78218410041841
Testing_accuracy: 0.75
```

Fig. 6.7 Accuracy for 80% train and 20% test dataset split

The data has been split into 70% for training and 30% for testing. The accuracy of the classifier is shown in Fig. 6.8.

```
 Train test split 70 % 30 %

 size of the data: 1    89733
0    29757
Name: sw1, dtype: int64


 1. Decision tree*****
Training_accuracy: 0.8018746338605741
Testing_accuracy: 0.7666666666666667


2. Naive Bayes*****
Training_accuracy: 0.7339944765252322
Testing_accuracy: 0.7333333333333333


3.Random Forest *****
Training_accuracy: 0.8018746338605741
Testing_accuracy: 0.7666666666666667


4. K-Nearest Neighbour *****
Training_accuracy: 0.7872206879236756
Testing_accuracy: 0.8


5. Multi-Layered Perceptron Neural Network *****
Training_accuracy: 0.782190978324546
Testing_accuracy: 0.7333333333333333
```

Fig. 6.8 Accuracy for 70% train and 30% test dataset split

The data has been split into 60% for training and 40% for testing. The accuracy of the classifier is shown in Fig. 6.9.

```
 Train test split 60 % 40 %

 size of the data: 1    89724
0    29756
Name: sw1, dtype: int64


 1. Decision tree*****
Training_accuracy: 0.8018664211583528
Testing_accuracy: 0.8


2. Naive Bayes*****
Training_accuracy: 0.7339805825242719
Testing_accuracy: 0.775


3.Random Forest *****
Training_accuracy: 0.8018664211583528
Testing_accuracy: 0.8


4. K-Nearest Neighbour *****
Training_accuracy: 0.7892115835286241
Testing_accuracy: 0.875


5. Multi-Layered Perceptron Neural Network *****
Training_accuracy: 0.7821811181787747
Testing_accuracy: 0.775
```

Fig. 6.9 Accuracy for 60% train and 40% test data split

The data has been split into 50% for training and 50% for testing. The accuracy of the classifier is shown in Fig. 6.10

```
 Train test split 50 % 50 %

 size of the data: 1    89718
0    29752
Name: sw1, dtype: int64


 1. Decision tree*****
Training_accuracy: 0.8018833179877793
Testing_accuracy: 0.76


2. Naive Bayes*****
Training_accuracy: 0.7339917971038754
Testing_accuracy: 0.74


3.Random Forest *****
Training_accuracy: 0.8018833179877793
Testing_accuracy: 0.76


4. K-Nearest Neighbour *****
Training_accuracy: 0.790332300996066
Testing_accuracy: 0.76


5. Multi-Layered Perceptron Neural Network *****
Training_accuracy: 0.7821963672888591
Testing_accuracy: 0.74
```

Fig. 6.10 Accuracy for 50% train and 50% test data split

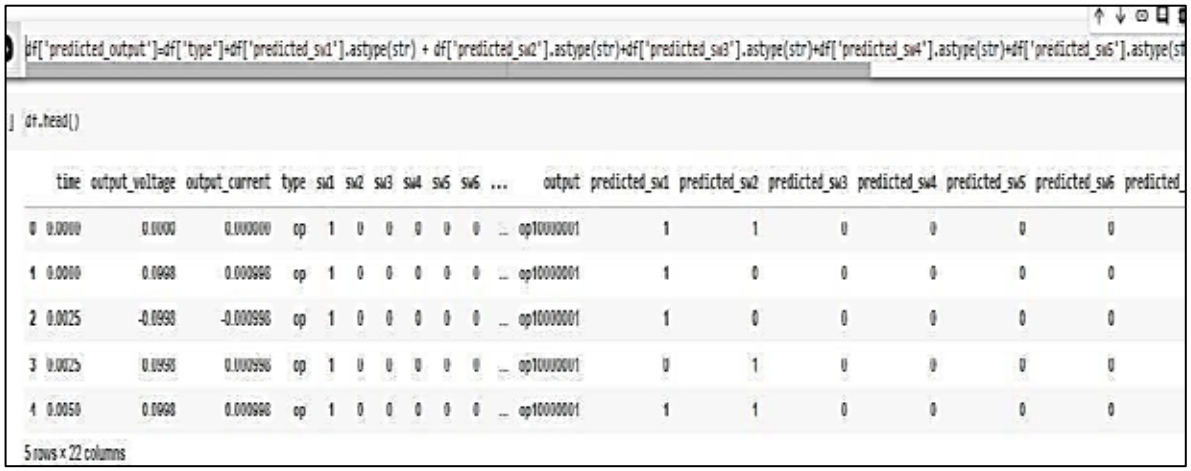
6.1.5 BUILDING THE CLASSIFIER

Fig. 6.11 presents eight Separate models and Build for eight Separate switches and their output is plottedas predicted switch number.

```
df['predicted_sw1']=model1.predict(df_input)
df['predicted_sw2']=model2.predict(df_input)
df['predicted_sw3']=model3.predict(df_input)
df['predicted_sw4']=model4.predict(df_input)
df['predicted_sw5']=model5.predict(df_input)
df['predicted_sw6']=model6.predict(df_input)
df['predicted_sw7']=model7.predict(df_input)
df['predicted_sw8']=model8.predict(df_input)
```

Fig. 6.11 Mapping the output of the separate models

These are then concatenated to the Data frame and the final output is described as “predicted output”



```
df['predicted_output']=df['type']+df['predicted_sw1'].astype(str) + df['predicted_sw2'].astype(str)+df['predicted_sw3'].astype(str)+df['predicted_sw4'].astype(str)+df['predicted_sw5'].astype(st

df.head()
```

	time	output_voltage	output_current	type	sw1	sw2	sw3	sw4	sw5	sw6	...	output	predicted_sw1	predicted_sw2	predicted_sw3	predicted_sw4	predicted_sw5	predicted_sw6	predicted_
0	0.0000	0.0000	0.000000	op	1	0	0	0	0	0	...	op10000001	1	1	0	0	0	0	
1	0.0000	0.0998	0.000998	op	1	0	0	0	0	0	...	op10000001	1	0	0	0	0	0	
2	0.0025	-0.0998	-0.000998	op	1	0	0	0	0	0	...	op10000001	1	0	0	0	0	0	
3	0.0025	0.0998	0.000998	op	1	0	0	0	0	0	...	op10000001	0	1	0	0	0	0	
4	0.0050	0.0998	0.000998	op	1	0	0	0	0	0	...	op10000001	1	1	0	0	0	0	

5 rows × 22 columns

Fig. 6.12 Concatenating into the data frame

Fig. 6.12 shows the output of building different classifiers. It shows the faulted switches and the normal conditioned switches.

```
#model 1
classifier=DecisionTreeClassifier()
x train, x test, y train, y test = train test split(df input,target1, test size=30, random state=42)
classifier=DecisionTreeClassifier()
model1=classifier.fit(x_train, y_train)
print('Training_accuracy1:',model1.score(x_train,y_train))
print('Testing_accuracy1:',model1.score(x_test,y_test))

#model 2
classifier=DecisionTreeClassifier()
x_train, x_test, y_train, y_test = train_test_split(df_input,target2, test_size=30, random_state=42)
classifier=DecisionTreeClassifier()
model2=classifier.fit(x_train, y_train)
print('Training_accuracy2:',model2.score(x_train,y_train))
print('Testing_accuracy2:',model2.score(x_test,y_test))

#model 3
classifier=DecisionTreeClassifier()
x train, x test, y train, y test = train test split(df input,target3, test size=30, random state=42)
classifier=DecisionTreeClassifier()
model3=classifier.fit(x_train, y_train)
print('Training_accuracy3:',model3.score(x_train,y_train))
print('Testing_accuracy3:',model3.score(x_test,y_test))

#model 4
classifier=DecisionTreeClassifier()
x_train, x_test, y_train, y_test = train_test_split(df_input,target4, test_size=30, random_state=42)
classifier=DecisionTreeClassifier()
model4=classifier.fit(x_train, y_train)
print('Training_accuracy4:',model4.score(x train,y train))
print('Testing_accuracy4:',model4.score(x_test,y_test))

#model 5
classifier=DecisionTreeClassifier()
x train, x test, y train, y test = train test split(df input,target5, test size=30, random state=42)
classifier=DecisionTreeClassifier()
model5=classifier.fit(x_train, y_train)
print('Training_accuracy5:',model5.score(x_train,y_train))
print('Testing_accuracy5:',model5.score(x_test,y_test))

#model 6
classifier=DecisionTreeClassifier()
x_train, x_test, y_train, y_test = train_test_split(df_input,target6, test_size=30, random_state=42)
classifier=DecisionTreeClassifier()
model6=classifier.fit(x_train, y_train)
print('Training_accuracy6:',model6.score(x_train,y_train))
print('Testing_accuracy6:',model6.score(x_test,y_test))

#model 7
classifier=DecisionTreeClassifier()
x train, x test, y train, y test = train test split(df input,target7, test size=30, random state=42)
classifier=DecisionTreeClassifier()
model7=classifier.fit(x_train, y_train)
print('Training_accuracy7:',model7.score(x_train,y_train))
print('Testing_accuracy7:',model7.score(x_test,y_test))

#model 8
classifier=DecisionTreeClassifier()
x_train, x_test, y_train, y_test = train_test_split(df_input,target8, test_size=30, random_state=42)
classifier=DecisionTreeClassifier()
model8=classifier.fit(x_train, y_train)
print('Training_accuracy8:',model8.score(x_train,y_train))
print('Testing_accuracy8:',model8.score(x_test,y_test))
```

Fig. 6.13 Output of building the classifier

6.2 MAKING PREDICTIONS

The input to be predicted is received from the user (Time period, Output voltageand output current) and the prediction is done by the model and the results are shown.

6.3 INPUT FROM USER

```
x = input("Enter a value for time: ")
y = input("Enter a value for output_voltage: ")
z = input("Enter a value for output current: ")
```

```
Enter a value for time: 0.0025
Enter a value for output_voltage: -0.0998
Enter a value for output current: -0.000998
```

Fig. 6.14 Getting the input from the user

6.5 OUTPUT OF CLASSIFIERS

Fig. 6.15 Output for the classifier

In Fig. 6.15, it is predicted that the switch 1,7 and 8 are considered as faulted switches.

6.6 HARDWARE DESIGN

The hardware design includes regulated power supply, DSO, Arduino board with software program then switches of N-channel and P-channel MOSFET and optocoupler is act as driver circuit this shown Fig. 6.16.

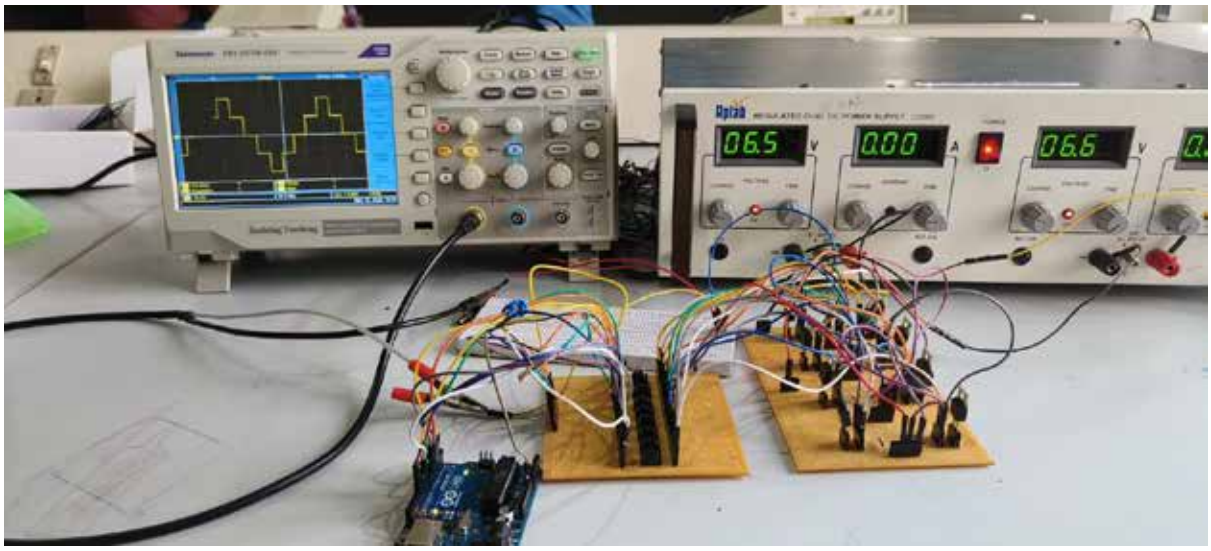


Fig. 6.16 Hardware Design

6.7 HARDWARE OUTPUT WAVEFORM

The Fig. 6.17 presents the output waveform of five-level cascaded h-bridge inverter without any open switch fault under normal condition.

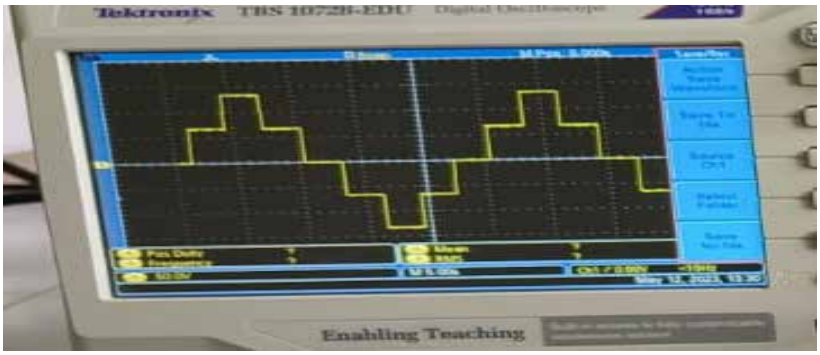


Fig. 6.17 Hardware output waveform without fault

6.7.1 Output of Cascaded H-Bridge Inverter Under Fault State

Here manually switch S_1 is under fault condition and the Fig. 6.18 show the output waveform of the cascaded h-bridge inverter this waveform makes it simple to identify the faulted switch

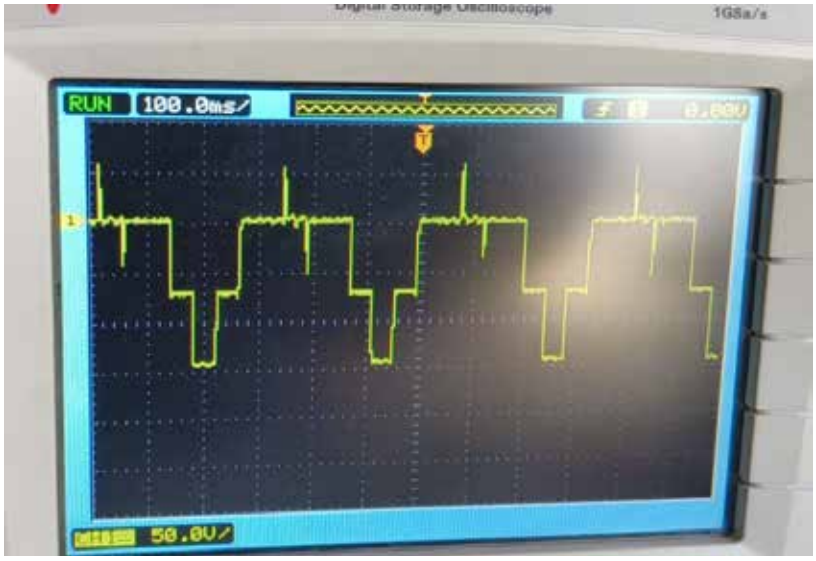


Fig. 6.18 Output waveform under switch S_1 in fault condition

Here manually switch S_1, S_3 is under fault condition and the Fig. 6.19 show the output waveform.

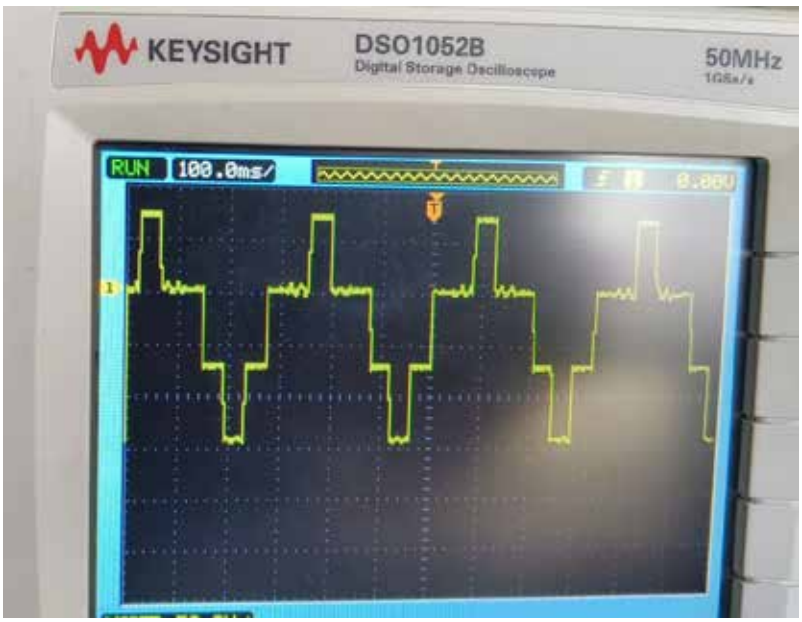


Fig. 6.19 Output waveform under switches S_1, S_3 in fault condition

6.8 COMPARISON OF ALGORITHM IN SIMULATION MODEL

The algorithms are compared in the simulation for different fault condition and the accuracy of each algorithm are obtained

Table 6.1 Comparison of algorithm in simulation model

ALGORITHMS	TEST CASES	ACCURACY (%)
K-NN	Without fault	8.0
	S_1 Fault	7.6
	S_1, S_3 Fault	6.8
RANDOM FOREST	Without fault	7.8
	S_5, S_6 Fault	7.2
	S_7, S_8 Fault	6.6
DECISION TREE	Without fault	6.8
	S_4, S_5, S_6 Fault	6.2
	S_1, S_2, S_3 Fault	6.4
NAÏVE BAYES	Without fault	6.9
	S_1, S_2, S_3, S_4, S_5 Fault	5.9
	S_1, S_2, S_3, S_4 Fault	6.3

The results shown in Table 6.1 shows that the K-NN algorithm has more accuracy than the other algorithm.

6.9 COMPARISON OF SIMULATION AND HARDWARE MODEL

Table 6.2 presents that the K-NN algorithm in the software model gives more accuracy when compared to the hardware model. This is because there are more noise and distortions.

Table 6.2 Comparison of simulation and hardware model

ALGORITHMS	TEST CASES	ACCURACY (%)
K-NN (Software Model)	Without fault	8.0
	S_1 Fault	7.6
	S_1, S_3 Fault	6.8
K-NN (Hardware Model)	Without fault	7.8
	S_3, S_4 Fault	7.2
	S_6, S_7, S_8 Fault	6.6

CHAPTER 7

CONCLUSION

This chapter concludes the project by summarizing the procedures that were carried out to execute this project. The project work involves successful implementations of fault detection in five-level H-bridge inverter. The combination of 256 datasets have been extracted from the simulation model. These datasets have been used in the machine learning algorithms such as K-NN, Decision tree, Random Forest and Naive Bayes. The datasets extracted from the hardware model have been used in machine learning algorithm such as K-NN to detect the fault. The results of the simulation model and hardware model has been compared.

7.1 FUTURE WORK

The current algorithm is capable of providing the high accuracy of “True positive” and “True negative” and less amount of “false positive” and “false negative”. It can reduce the training dataset and correcting class imbalances. The further fault possibilities can also be included for the modelling. The signal processing techniques can also be implemented for precise modelling and better detection. The prognosis of the particular switch which is detected as the fault by the model can be estimated for Remaining Useful Life (RUL) which can be implemented in real-time circuits.

REFERENCES

[1] A. Arbune, “Study of cascaded H-Bridge multilevel inverter,,” International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT), vol. 10.1109/ICACDOT, no. 2016.7877574, pp. 179-182, 2016.

[2] G. Zhang, “Open-circuit fault diagnosis for cascaded H-bridge multilevel inverter based on LS-PWM technique,” CPSS Transactions on Power Electronics and Applications, vol. 6, no. 3, pp. 201-208, 2021.

[3] D. Kore, “5-level cascade H bridge multilevel inverter fed induction motor drive using sinusoidal PWM,” Intelligent Computing, Instrumentation and Control Technologies, vol. 1, no. 10.1109/ICICICT46008.2019.8993170, pp. 1074-1076, 2019.

[4] W. Mohammad Ali Hosseinzadeh, “Modified H-Bridge Inverter with Reduced Numberof Switching Devices,” IEEE International Conference on Industrial Technology, vol. 10.1109, no. 8755220, pp. 1521-1526, 2019.

[5] R. Divya Subramanian, “Five Level Cascaded H-Bridge Multilevel,” International Journal of Engineering and Innovative Technology, vol. 3, no. 1, 2013.

[6] S. Shadlu, “{Fault detection and diagnosis in voltage source inverters using principle component analysis,” Conference on Knowledge-Based Engineering and Innovation , pp. 0509-0515, 2017.

[7] T. Wang, “Cascaded H-bridge multilevel inverter system fault diagnosis using a PCA and multiclass relevance vector machine approach,” IEEE Transactions on Power Electronics , pp. 7006-7018, 2015.

[8] M. Chai, “Fault Detection and Localization for Cascaded H-Bridge Multilevel Converter With Model Predictive Control,” IEEE Transactions on Power Electronics, vol. 35, no. 10, pp. 10109-10120, 2020.

[9] A. Anand, “A Generalized Switch Fault Diagnosis for Cascaded H-Bridge Multilevel Inverters Using Mean Voltage Prediction,” IEEE Transactions on Industry Applications, vol. 56, no. 2, pp. 1563-1574, 2020.

[10] H. Lee, “A fault detection method in cascaded H-bridge multilevel inverter,” Conference on Power and Energy, pp. 473-478, 2016.

[11] A. Anand, “Open switch fault detection in Cascaded H-Bridge Multilevel Inverter using normalised mean voltages,” IEEE 6th International Conference on Power Systems, pp. 1-6, 2016.

[12] H. Mhiesan, “A Fault-Tolerant Hybrid Cascaded H-Bridge Multilevel Inverter,” IEEE Transactions on Power Electronics, vol. 35, no. 12, pp. 12705-12715, 2020.

[13] S. Mukherjee, “Fast fault detection of open power switch in cascaded H-bridge multilevel inverters,” IEEE Transportation Electrification Conference and Expo , pp. 1-5, 2016.

[14] K. Sahu, “Open Circuit Switch Fault Detection in Multilevel Inverter Topology using Machine Learning Techniques,” IEEE 9th Power India International Conference, pp. 1-6, 2020.

[15] X. Cui Li, “Diagnosis and location of the open-circuit fault in modular multilevel converters: An improved machine learning method,” vol. 331, pp. 58-66, 2019.

[16] K. Pavan Mehta, “A fault-diagnosis and tolerant control technique for five-level cascaded H-bridge inverters,” vol. 15, no. 4, pp. 366-376, 2021.

[17] W. Sim, “A detection method for an open-switch fault in cascaded H-bridge multilevel inverters,” IEEE Energy Conversion Congress and Exposition, pp. 2101-2106, 2014.

[18] Z. Mahmoud Shahbazi, “Open- and Short-Circuit Switch Fault Diagnosis for Nonisolated DC-DC Converters Using Field Programable Gate Array,” IEEE Transactions on Industrial Electronics, vol. 60, no. 9, pp. 4136-4146, 2013.