DEVELOPMENT OF BLDC MOTOR DRIVER FOR 24V 100W

KAVIN P - Department of Electrical And Electronics Engineering, Bannari Amman Institute Of Technology, Erode
PRASANTH A -Department of Electrical And Electronics Engineering, Bannari Amman Institute Of Technology, Erode
JANARTHANAM T- Department Of Electrical And Electronics Engineering, Bannari Amman Institute Of Technology, Erode
CHANDRU K - Department of Electrical And Electronics Engineering, Bannari Amman Institute Of Technology, Erode
VEERAKUMAR S - ASSOCIATE PROFESSOR, Department of Electrical And Electronics Engineering, Bannari Amman Institute Of Technology, Erode

---

Abstract - *Brushless DC (BLDC) motors are widely used in modern industrial, automotive, and home automation applications due to their high efficiency, low maintenance, and superior torque-to-weight ratio. This project focuses on the design and development of a BLDC motor driver for a* ***24V, 100W*** *motor, capable of delivering up to* ***5A*** *of current with precise speed and torque control. The proposed driver will employ a microcontroller-based control system to generate high-frequency* ***Pulse Width Modulation (PWM)*** *signals for driving a three-phase inverter bridge. The system will support both* ***sensor-based commutation*** *using Hall effect sensors and* ***sensorless commutation*** *based on back Electromotive Force (back-EMF) detection, ensuring enhanced flexibility and adaptability for various operating conditions.Brushless DC (BLDC) motors are widely used in modern industrial, automotive, and home automation applications due to their high efficiency, low maintenance, and superior torque-to-weight ratio. This project focuses on the design and development of a BLDC motor driver for a* ***24V, 100W*** *motor, capable of delivering up to* ***5A*** *of current with precise speed and torque control. The proposed driver will employ a microcontroller-based control system to generate high-frequency* ***Pulse Width Modulation (PWM)*** *signals for driving a three-phase inverter bridge. The system will support both* ***sensor-based commutation*** *using Hall effect sensors and* ***sensorless commutation*** *based on back Electromotive Force (back-EMF) detection, ensuring enhanced flexibility and adaptability for various operating conditions.The motor driver will feature a high-speed gate driver circuit for driving the MOSFET-based inverter, ensuring low switching losses and high operational efficiency. A* ***trapezoidal commutation algorithm*** *will be implemented to achieve smooth and reliable motor operation. The microcontroller will handle feedback from the motor, including speed, position, and current data, to enable closed-loop control. Overcurrent, overvoltage, and thermal protection mechanisms will be integrated into the design to ensure safe operation and to prevent hardware damage under fault conditions.*

Key Words:

Brushless DC (BLDC) Motor
Motor Driver
Microcontroller-Based Control
Pulse Width Modulation (PWM)
Three-Phase Inverter Bridge
Sensor-Based Commutation
Sensorless Commutation
Back Electromotive Force (Back-EMF)
High-Speed Gate Driver
MOSFET-Based Inverter
Trapezoidal Commutation Algorithm
Closed-Loop Control
Overcurrent Protection
Overvoltage Protection
Thermal Protection

1. INTRODUCTION

Torque ripple is a common issue in Brushless DC (BLDC) motors, which refers to the variation in torque output during motor operation. It can result in undesirable effects such as vibration, noise, and reduced motor performance. Therefore, reducing torque ripple is an important consideration in the design and control of BLDC motors. Here are some possible techniques for torque ripple reduction in BLDC motors:

- Advanced Motor Design: Proper motor design can help in reducing torque ripple. Optimizing the motor's magnetic circuit, including the shape and size of the rotor and stator teeth, can minimize torque fluctuations. This may involve using special rotor or stator pole configurations, such as skewed or slotted rotors, to reduce cogging effects.
- Sensorless Control Techniques: Traditional BLDC motor control methods rely on position sensors to determine the rotor position, which can introduce additional errors and contribute to torque ripple. Sensorless control techniques, such as Back Electromotive Force (BEMF) or observer-based methods, can estimate the rotor position without using sensors and provide smoother torque control.
- Pulse Width Modulation (PWM) Techniques: PWM techniques can be used to control the applied voltage to the motor, which can help in reducing torque ripple. Advanced PWM techniques, such as Space Vector Modulation (SVM), can provide better control of motor voltages and currents, resulting in reduced torque ripple.
- Current Ripple Reduction Techniques: Ripple in motor current can directly affect torque ripple. Techniques such as current hysteresis control, current shaping, or using passive or active filters can help in reducing current ripple and subsequently minimize torque ripple.
- Advanced Control Algorithms: Advanced control algorithms, such as predictive control or adaptive control, can be employed to actively compensate for torque ripple by dynamically adjusting the motor's operating parameters, such as the phase current, voltage, or switching frequency.
- Commutation Optimization: Proper commutation of the motor phases is crucial to achieve smooth motor operation. Optimizing the commutation strategy, such as using advanced commutation techniques like trapezoidal or sinusoidal commutation, can minimize torque ripple and improve motor performance.
- Mechanical Damping: Adding mechanical damping elements, such as dampers or flywheels, to the motor or the load can help in reducing mechanical vibrations, which can contribute to torque ripple.

It's important to note that the most effective torque ripple reduction technique may vary depending on the specific motor design, operating conditions, and application requirements. A combination of multiple techniques may be needed to achieve the desired level of torque ripple reduction in a BLDC motor. Proper simulation, analysis, and experimentation are usually carried out to optimize the chosen technique for a specific motor and application.

1.1 BRUSHLESS DC MOTOR (BLDC)

A brushless DC motor (known as BLDC) is a permanent magnet synchronous electric motor which is driven by direct current (DC) electricity and it accomplishes electronically controlled commutation system (commutation is the process of producing rotational torque in the motor by changing phase currents through it at appropriate times) instead of a mechanically commutation system. BLDC motors are also referred as trapezoidal permanent magnet motors. A brushless DC motor (known as BLDC) is a permanent magnet synchronous electric motor which is driven by direct current (DC) electricity and it accomplishes electronically controlled commutation system (commutation is the process of producing rotational torque in the motor by changing phase currents through it at appropriate times) instead of a mechanically commutation system. BLDC motors are also referred as trapezoidal permanent magnet motors. The brushed type DC motor, wherein the brushes make the mechanical contact with commutator on the rotor so as to form an electric path between a DC electric source and rotor armature windings, BLDC motor employs electrical commutation with permanent magnet rotor and a stator with a sequence of coils. In this motor, permanent magnet (or field poles) rotates and current carrying conductors are fixed.

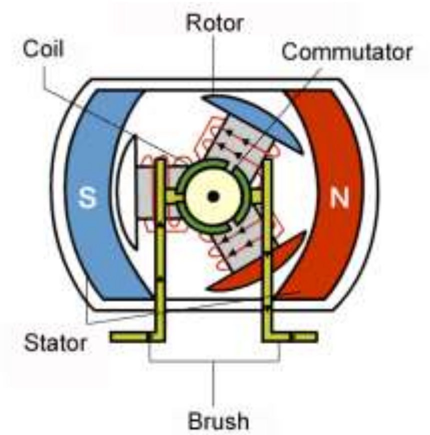

Fig-1: Operation of the Brushed DC Motor.

1.2 COMMON MOTOR TYPES

Motors differ according to their power type (AC or DC) and their method for generating rotation. Below, we look briefly at the features and uses of each type.

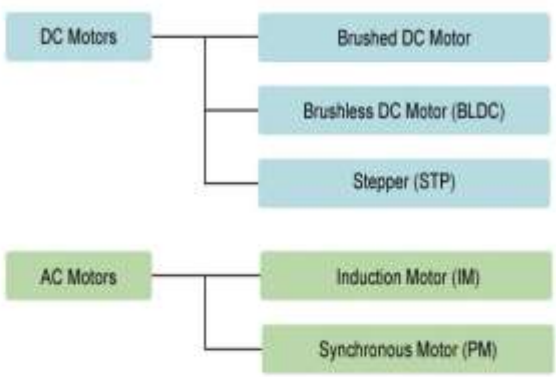

Fig-2 Different Types of Motors

Brushed DC motors, featuring simple design and easy control, are widely used to open and close disk trays. In cars, they are often used for retracting, extending, and positioning electrically-powered side windows. The low cost of these motors makes them suitable for many uses. One drawback, however, is that brushes and commutators tend to wear relatively quickly as a result of their continued contact, requiring frequent replacement and periodic maintenance. A stepper motor is driven by pulses; it rotates through a specific angle (step) with each pulse. Because the rotation is precisely controlled by the number of pulses received, these motors are widely used to implement positional adjustments. They are often used, for example, to control paper feed in fax machines and printers—since these devices feed paper in fixed steps, which are easily correlated with pulse count. Pausing can also be easily controlled, as motor rotation stops instantly when the pulse signal is interrupted. With synchronous motors, rotation is synchronous with the frequency of the supply current. These motors are often used to drive the rotating trays in microwave ovens; reduction gears in the motor unit can be used to obtain the appropriate rotational speeds to heat food. With induction motors, too, the rotation speed varies with frequency; but the movement is not synchronous. In the past, these motors were often used in electric fans and washing machines. There are various types of motor in common use. In this session, we look at the advantages and applications of brushless DC motors.

As their name implies, brushless DC motors do not use brushes. With brushed motors, the brushes deliver current through the commutator into the coils on the rotor. So how does a brushless motor pass current to the rotor coils? It doesn't—because the coils are not located on the rotor. Instead, the rotor is a permanent magnet; the coils do not rotate, but are instead fixed in place on the stator. Because the coils do not move, there is no need for brushes and a commutator. (See Figure. 3.) With the brushed motor, rotation is achieved by controlling the magnetic fields generated by the coils on the rotor, while the magnetic field generated by the stationary magnets remains fixed. To change the rotation speed, you change the voltage for the coils. With a BLDC motor, it is the permanent magnet that rotates; rotation is achieved by changing the direction of the magnetic fields generated by the surrounding stationary coils. To control the rotation, you adjust the magnitude and direction of the current into these coils.

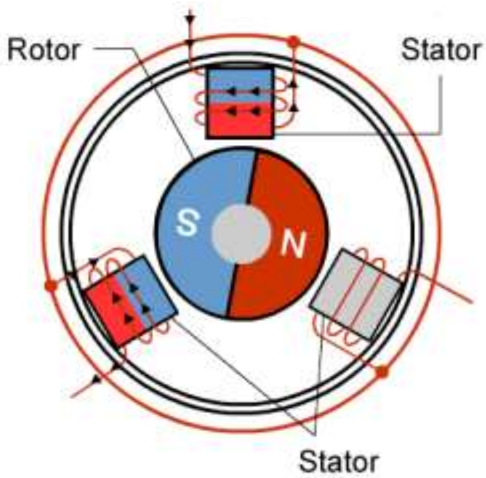

Fig-3: A BLDC Motor

Since the rotor is a permanent magnet, it needs no current, eliminating the need for brushes and commutator. Current to the fixed coils is controlled from the outside.

1.3 ADVANTAGES OF BLDC MOTOR

A BLDC motor with three coils on the stator will have six electrical wires (two to each coil) extending from these coils. In most implementations three of these wires will be connected internally, with the three remaining wires extending from the motor body (in contrast to the two wires extending from the brushed motor described earlier). Wiring in the BLDC motor case is more complicated than simply connecting the power cell's positive and negative terminals; we will look more closely at how these motors work in the second session of this series. Below, we conclude by looking at the advantages of by BLDC motors. One big advantage is efficiency, as these motors can control continuously at maximum rotational force (torque). Brushed motors, in contrast, reach maximum torque at only certain points in the rotation. For a brushed motor to deliver the same torque as a brushless model, it would need to use larger magnets. This is why even small BLDC motors can deliver considerable power.

The second big advantage—related to the first—is controllability. BLDC motors can be controlled, using feedback mechanisms, to delivery precisely the desired torque and rotation speed. Precision control in turn reduces energy consumption and heat generation, and—in cases where motors are battery powered—lengthens the battery life. BLDC motors also offer high durability and low electric noise generation, thanks to the lack of brushes. With brushed motors, the brushes and commutator wear down as a result of continuous moving contact, and also produce sparks where contact is made. Electrical noise, in particular, is the result of the strong sparks that tend to occur at the areas where the brushes pass over the gaps in the commutator. This is why BLDC motors are often considered preferable in applications where it is important to avoid electrical noise.

1.4 APPLICATIONS

BLDC motors offer high efficiency and controllability, and that they have a long operating life. So what are they good for? Because of their efficiency and longevity, they are widely used in devices that run continuously. They have long been used in washing machines, air conditioners, and other consumer electronics; and more recently, they are appearing in fans, where their high efficiency has contributed to a significant reduction in power consumption. They are also being used to drive vacuum machines. In one case, a change in the control program resulted in a large jump in rotational speed—an example of the superlative controllability offered by these motors. BLDC motors are also being used to spin hard disc drives, where their durability keeps the drives operating dependably over the long term, while their power efficiency contributes to energy reduction in an area where this is becoming increasingly important.

2 OBJECTIVES AND METHODOLOGY

This chapter outlines the main objectives and methodology of our project, explaining both our goals and how we plan to achieve them. We begin by setting clear objectives that guide the project, identifying specific aims and the anticipated outcomes. Following this, the methodology section describes our approach, detailing the tools and techniques used for gathering and analyzing data relevant to our goals.

2.1 OBJECTIVES

This project is design and develop a high-efficiency BLDC motor driver capable of controlling a 24V, 100W BLDC motor, with the ability to deliver up to 5A of current and provide precise speed and torque control. The driver will employ a microcontroller-based system to generate high-frequency PWM signals for driving a three-phase inverter bridge, ensuring smooth motor operation. The system will be versatile, supporting both sensor-based commutation using Hall effect sensors and sensorless commutation based on back-EMF detection. The motor driver will incorporate closed-loop control to maintain accurate speed and position, using feedback data for continuous adjustments. Additionally, the design will prioritize safety, with built-in overcurrent, overvoltage, and thermal protection to safeguard the motor and driver from potential faults. A trapezoidal commutation algorithm will be implemented to ensure reliable motor operation, and the system will be designed for high efficiency and low maintenance, reducing power loss and ensuring long-term reliability. Ultimately, the driver will be adaptable for various industrial, automotive, and home automation applications, offering flexibility in performance across different operating conditions.

2.2 FLOW DIAGRAM

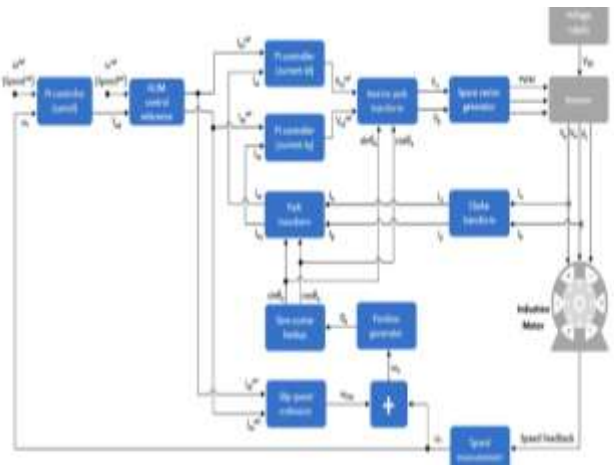

Fig-4: Flow Diagram

3. SOFTWARE

MATLAB (matrix laboratory) is a fourth-generation high-level programming language and interactive environment for numerical computation, visualization and programming. MATLAB is developed by Math Works .It allows matrix manipulations; plotting of functions and data; implementation of algorithms; creation of user interfaces; interfacing with programs written in other languages, including C, C++, Java, and Fortran ;analyze data; develop algorithms; and create models and applications. It has numerous built-in commands and math functions that help you in mathematical calculations, generating plots and performing numerical methods.

3.1 Features of MATLAB

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

3.2 Uses of MATLAB

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

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

3.3 Simulink

Simulink is a block diagram environment for multi domain simulation and Model-Based Design. It supports system-level design, simulation, automatic code generation, and continuous test and verification of embedded systems. Simulink provides a graphical editor, customizable block libraries, and solvers for modeling and simulating dynamic systems. It is integrated with MATLAB®, enabling you to incorporate MATLAB algorithms into models and export simulation results to MATLAB for further analysis.

3.4 Key Features

Graphical editor for building and managing hierarchical block diagrams Libraries of predefined blocks for modeling continuous-time and discrete-time systems, Simulation engine with fixed-step and variable-step ODE solvers. Scopes and data displays for viewing simulation results Project and data management tools for managing model files and data Model analysis tools for refining model architecture and increasing simulation speed MATLAB Function block for importing MATLAB algorithms into models, Legacy Code Tool for importing C and C++ code into models.

3.5 Tool for Model-Based Design

With Simulink, you can move beyond idealized linear models to explore more realistic nonlinear models, factoring in friction, air resistance, gear slippage, hard stops, and the other things that describe real-world phenomena. Simulink turns your computer into a laboratory for modeling and analyzing systems that would not be possible or practical otherwise. Whether you are interested in the behavior of an automotive clutch system, the flutter of an airplane wing, or the effect of the monetary supply on the economy, Simulink provides you with the tools to model and simulate almost any real-world problem. Simulink also provides examples that model a wide variety of real-world Simulink provides a graphical user interface (GUI) for building models as block diagrams, allowing you to draw models as you would with pencil and paper.

Simulink also includes a comprehensive block library of sinks, sources, linear and nonlinear components, and connectors. If these blocks do not meet your needs, however, youcan also create your own blocks. The interactive graphical environment simplifies themodeling process, eliminating the need to formulate differential and difference equationsin a language or program. Models are hierarchical, so you can build models using both top-down and bottom-up approaches. You can view the system at a high level, and then double-click blocks to see increasing levels of model detail. This approach provides insight into how a model is organized and how its parts interact.

3.6 Tool for Simulation

After you define a model, you can simulate its dynamic behavior using a choice of mathematical integration methods, either from the Semolina menus or by entering commands in the MATLAB Command Window. The menus are convenient for interactive work, while the command line is useful for running a batch of simulations. For example, if you are doing Monte Carlo simulations or want to apply a parameter across a range of values, you can use MATLAB scripts. Using scopes and other display blocks, you can see the simulation results while the simulation runs. You can then change parameters and see what happens for “what if” exploration. The simulation results can be put in the MATLAB workspace forpostprocessing and visualization.

3.7 Tool for Analysis

Model analysis tools include linearization and trimming tools, which you can access from the MATLAB command line, plus the many tools in MATLAB and its application tool boxes. Because MATLAB and Simulink are integrated, you can simulate, analyze, and revise your models in either environment at any point.

3.8 Interaction with MATLAB Environment

Simulink software is tightly integrated with the MATLAB environment. It requires MATLAB to run, depending on it to define and evaluate model and block parameters. Simulink can also use many MATLAB features. For example, Simulink can use the

3.9 MATLAB

- Define model inputs.
- Store model outputs for analysis and visualization.
- Perform functions within a model, through integrated calls to MATLAB operators and functions.

3.10 Model- Based Design

Model-Based Design is a process that enables faster, more cost-effective development of dynamic systems, including control systems, signal processing, and communications systems. In Model-Based Design, a system model is at the center of the development process, from requirements development, through design, implementation, and testing. The model is an executable specification that you continually refine throughout the development process. After model development, simulation shows whether the model works correctly. When software and hardware implementation requirements are included, such as fixed-point and timing behavior, you can automatically generate code for embedded deployment and create test benches for system verification, saving time and avoiding the introduction of manually coded errors.

3.11 Model-Based Design allows you to improve efficiency by:

Model-Based Design (MBD) enhances efficiency by providing a unified design environment that streamlines collaboration across project teams. It enables direct linkage between designs and requirements, ensuring traceability and compliance throughout development. By integrating testing within the design process, errors can be continuously identified and corrected early, reducing development time and costs. Multi-domain simulation allows for algorithm refinement, optimizing performance before hardware implementation. Additionally, MBD facilitates automatic code generation for embedded systems, reducing manual coding efforts and minimizing errors. Test suites can be developed and reused to enhance validation efficiency, while automated documentation generation simplifies record-keeping and compliance reporting. Furthermore, MBD supports design reuse, allowing systems to be deployed across multiple processors and hardware platforms, improving scalability and adaptability in diverse applications.

3.12 Model-Based Design Process

Modeling any system involves six key steps, starting with **defining the system**, where the overall objectives, constraints, and expected behavior are established. Next, **identifying system components** ensures that all essential elements and interactions are recognized. Once components are identified, the system is **modeled using mathematical equations**, representing its dynamic behavior and control logic. These first three steps are typically performed outside the Simulink environment as part of the initial design and analysis process. After this groundwork, the **Simulink block diagram is built**, visually representing the system through interconnected functional blocks. The model is then tested by **running the simulation**, allowing engineers to analyze system performance under different conditions. Finally, the **simulation results are validated** against theoretical expectations or real-world data to ensure accuracy and reliability before proceeding to implementation.

3.13 Defining the System

The first step in modeling a dynamic system is to fully define the system. If you are modeling a large system that can be broken into parts, you should model each subcomponent on its own. Then, after building each component, you can integrate them into a complete model of the system. For example, the demo house heat example model of the heating system of a house is broken down into three main parts:

- Heater subsystem
- Thermostat subsystem
- Thermodynamic model subsystem

The most effective way to build a model of this system is to consider each of these Subsystems independently.

3.13 Identifying System Components

International Research Journal of Education and Technology Peer Reviewed Journal

ISSN 2581-7795

The second step in the modeling process is to identify the system components. Three types of components define a system:

• **Parameters** - System values that remain constant unless you change them

• **States** - Variables in the system that change over time

• **Signals** - Input and output values that change dynamically during a simulation

3.14 Modeling the System with Equations

The third step in modeling a system is to formulate the mathematical equations that describe the system. For each subsystem, use the list of system components that you identified to describe the system mathematically.

Your model may include:

- Algebraic equations
- Logical equations
- Differential equations, for continuous systems
- Difference equations, for discrete systems
- You use these equations to create the block diagram in Simulink.

3.15 Building the Simulink Block Diagram

After you have defined the mathematical equations that describe each subsystem, you can begin building a block diagram of your model in Simulink. Build the block diagram for each of your subcomponents separately. After you have modeled each subcomponent, you can then integrate them into a complete model of the system.

3.15 Running the Simulation

After you build the Simulink block diagram, you can simulate the model and analyze the results. Simulink allows you to interactively define system inputs, simulate the model, and observe changes in behavior. This allows you to quickly evaluate your model.

3.16 Validating the Simulation Results

Finally, you must validate that your model accurately represents the physical characteristics of the dynamic system. You can use the linearization and trimming tools available from the MATLAB command line, plus the many tools in MATLAB and its application toolboxes to analyze and validate your model.

3.17 Input supply voltage

The image shows the **Block Parameters** window for a **Three-Phase Source** in a MATLAB Simulink model. The source is configured in **Yg (wye-grounded)** configuration with a **phase-to-phase voltage of 24 Vrms** and a **frequency of 50 Hz**. The source impedance includes an **internal resistance of 0.02 Ω** and an **inductance of 0.05 mH**.This setup is commonly used for **simulating AC power systems** with RL impedance characteristics..

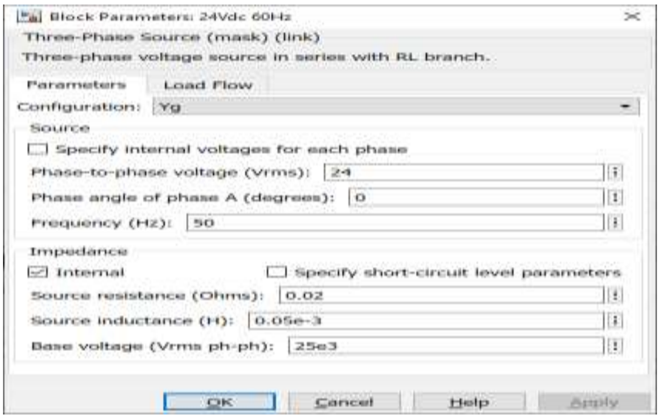

Fig-5: Input supply voltage

3.18 Bridge rectifier

The image displays the **Block Parameters** window for a **Universal Bridge** in MATLAB Simulink. It is configured with **three bridge arms** and **diodes** as the power electronic devices. The **snubber circuit** includes a **resistance of 10kΩ** and a **capacitance of 2nF** to protect switching elements. The **on-state resistance (Ron) is 1mΩ**, with **zero internal inductance (Lon)** and a **forward voltage drop (Vf) of 1.3V**. This setup is typically used for **rectifiers, inverters, or motor drive applications**.

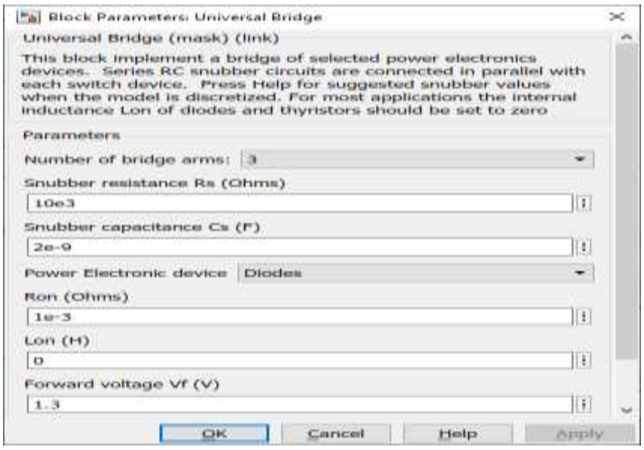

Fig-6: Bridge rectifier

3.19 Voltage source inverter

The image shows the block parameters for a universal bridge in MATLAB Simulink, configured as a three-phase inverter. It consists of three bridge arms and uses IGBT and diode devices. The snubber circuit includes a resistance of 10kΩ and an infinite capacitance. The on-state resistance is 1mΩ, and the forward voltages for the device and diode are 1.4V each. Measurements include all voltages and currents, making it suitable for inverter applications in motor drives and power conversion.

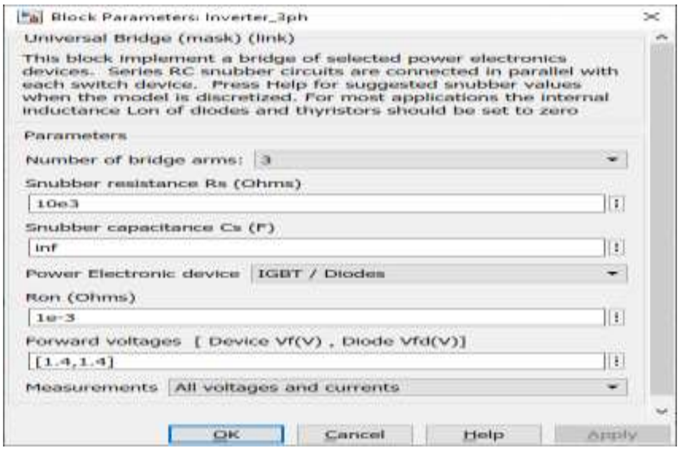

Fig-7: Voltage source inverter

3.20 Motor rating

The image displays the block parameters for a permanent magnet synchronous machine (BLDC) in MATLAB Simulink. It supports three-phase or five-phase configurations with wye-connected stator windings. The selected configuration has three phases and a trapezoidal back EMF waveform. The mechanical input is torque, and measurement output options are available.

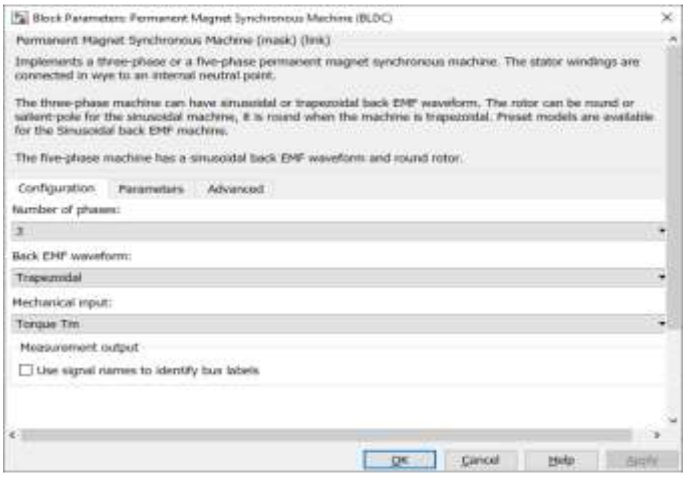

Fig-8: Motor rating

3.21 Torque flux estimation

The image represents a control system for a permanent magnet synchronous machine using a neural network-based approach. It includes inputs for speed, flux, and control signals, processed through proportional-integral controllers and a low-pass filter. The flux and torque references are determined using a flux table and control saturation, ensuring optimized motor performance.

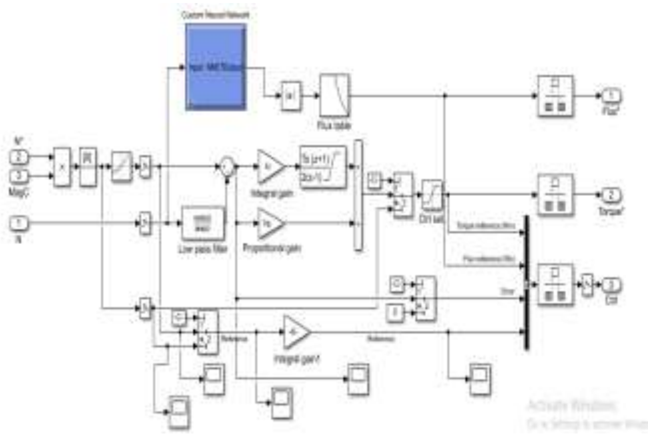

Fig-9: Torque flux estimation

3.22 Neural network training layer

The given diagram represents a **neural network or data processing model** with multiple layers. **Process Input 1** is fed into **Layer 1 and Layer 2**, which perform computations. The output from **Layer 1** is directed to **a{1}**, while **Layer 2** processes data further and passes it to **Process Output 1**. This structure suggests a **multi-layer processing approach**, likely used for **machine learning, deep learning, or signal processing** applications.

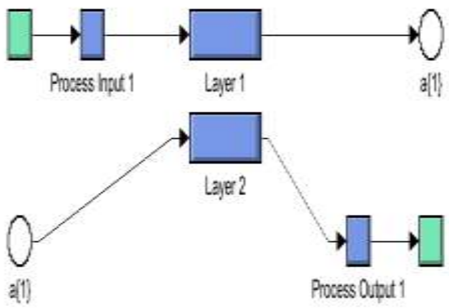

Fig-10: Neural network training layer

3.23 SVPWM (Space vector PWM)

The given Simulink diagram represents a **motor control system** using a **switching table** for precise inverter control. Inputs like **rotor position (H_Phi), torque (H_te), flux estimation, and angle** are processed by the **Flux Sector Seeker** to determine the sector. The **switching table** generates gate pulses for inverter switching, ensuring **optimal motor operation**. The outputs include **Gates for switching control**and**MagC for magnetization control**, both updated through delay blocks for synchronization. This model likely supports **Direct Torque Control (DTC) or Space Vector Modulation (SVM)**to achieve**efficient speed and torque regulation**in**BLDC or AC motor drives.**

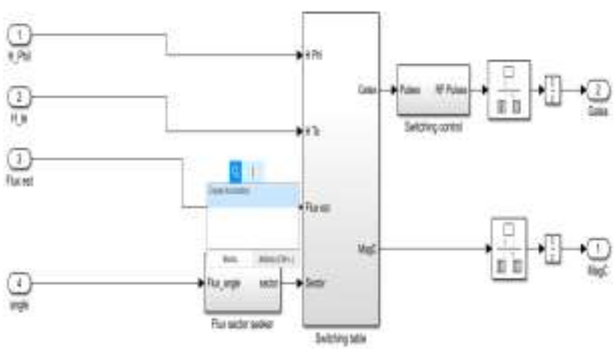

Fig-11: SVPWM (Space vector PWM)

3.24 Switching table

The given Simulink diagram represents a **Space Vector Modulation (SVM) or Direct Torque Control (DTC) scheme**for a motor drive system, likely a **BLDC or AC motor**. The system takes inputs such as **rotor position (H Phi), torque or time step (H Te), and sector information**, which helps in determining the correct switching state for the inverter. A **flux estimation and magnetization**block provides additional control data to manage motor operation effectively. The space vector switching logic processes the flux direction (**Flux = 1 or Flux = -1**) and determines the appropriate switching vector from a predefined set of **eight vectors (v0 to v7)**, each represented in binary format. These vectors correspond to specific switching states for a **three-phase inverter**.

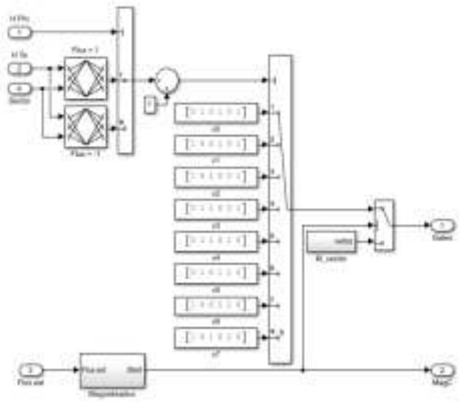

Fig-12: Switching table

3.25 Control Pulse Signal

The system is used for **pulse modulation, gating control, or signal conditioning** in applications such as **PWM generation for motor control, RF signal processing, or power inverter switching**. The structured modular approach allows for efficient signal handling, ensuring controlled pulse distribution to different system components.

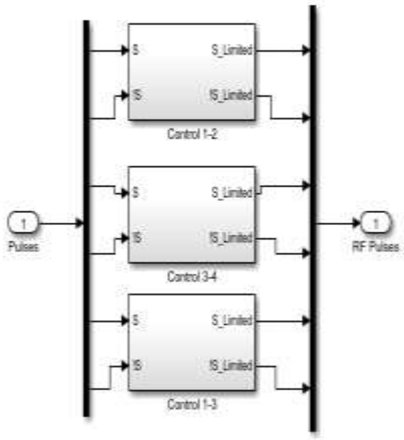

Fig-13: Control Pulse Signal

4 PROPOSED SYSTEM

A Brushless DC (BLDC) motor is a type of synchronous motor powered by a direct current (DC) power supply. Unlike brushed DC motors, BLDC motors use electronic commutation instead of mechanical brushes and a commutator to control the direction of current flow in the motor windings. The rotor of a BLDC motor is made of permanent magnets, while the stator contains the windings. The position of the rotor is detected using Hall effect sensors or back-EMF signals, which are used to generate appropriate switching signals for the motor phases The BLDC motor is known for its high efficiency, long lifespan, and precise control over speed and torque. These advantages make BLDC motors widely used in applications such as electric vehicles, robotics, industrial automation, computer hard drives, and household appliances. The proposed BLDC motor driver will consist of three primary modules:

- **Power Stage** – Includes high-speed MOSFETs and gate drivers to handle motor current and switching.
- **Control Stage** – A microcontroller (e.g., STM32 or similar) to implement the control algorithm, monitor feedback, and manage commutation.
- **Feedback and Protection Stage** – Includes sensors (Hall effect sensors), back-EMF detection, and protection circuitry for overcurrent, overvoltage, and thermal protection.

BLDC motors work on the principle of Lorentz force, which states that a current-carrying conductor placed in a magnetic field experiences a force perpendicular to both the current and the magnetic field. The fundamental working principle of BLDC motors is based on the interaction between the magnetic field generated by the rotor's permanent magnets and the magnetic field generated by the stator windings.

4.1 Stator and Rotor Construction

- **Stator:** The stator consists of laminated steel cores with windings arranged in a three-phase star or delta configuration. The windings are excited in a specific sequence to create a rotating magnetic field.
- **Rotor:** The rotor is made of permanent magnets (typically Neodymium Iron Boron – NdFeB) arranged with alternating north and south poles. The rotor aligns itself with the magnetic field generated by the stator, resulting in rotational motion.

4.2. Back Electromotive Force (Back-EMF)

As the rotor moves through the magnetic field of the stator, it generates a voltage known as the back-EMF. The back-EMF opposes the applied voltage and increases with motor speed according to the equation: The back-EMF is a critical factor in motor control, as it provides information about the rotor position and speed.

4.3. Lorentz Force and Torque Production

When current flows through the stator windings, it generates a magnetic field that interacts with the rotor's magnetic field, producing a torque. The generated torque is governed by:

where:

- TTT = generated torque
- KTK_TKT = torque constant
- III = phase current

The motor torque is maximized when the stator magnetic field is perpendicular to the rotor's magnetic field. This is achieved through proper commutation of the phases.

4.4. INDIRECT VECTOR CONTROL BY ORIENTATION OF ROTOR FLUX

The principle of which is based the vector control by orientation of rotor flux is to remove the internal coupling of the machine and return it to a linear control similar to that of a continuous current machine with a separated excitation.For the realization of the vector control, there are two methods: the direct method and the indirect method. This study is based on the indirect one; it uses the position of the rotor flux and it requires the use of a speed sensor.The equations of tensions, fluxes rotor and the electromagnetic torque in a frame of reference (d, q) turning at a speed ω in comparison with the stator are:

$$\begin{cases} V_{rd} = R_r . I_{rd} + \frac{d\Phi_{rd}}{dt} - \omega_r . \Phi_{rq} \\ V_{rq} = R_r . I_{rq} + \frac{d\Phi_{rq}}{dt} + \omega_r . \Phi_{rd} \end{cases}$$

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

$$C_{em} = \frac{P.M}{L_r}(\Phi_{rd} . I_{sq} - \Phi_{rq} . I_{sd})$$

Where

(Vrd,Vrq): rotor voltage

(φrd,φrq): Rotor flux

(Ird, Irq): Current rotor

(Isd, Isq): Current stator

ωr: Rotor pulsation

ωs: Stator pulsation

Rr,Lr: Rotor and inductance resistance

P: pole pairs

M: Mutuale inductance

The axis mark (d, q) is oriented along the axis carrying the vector of the rotor flux so we have

$$\Phi_{rq} = 0 \,;\, \Phi_{rd} = \Phi$$

With the condition, the expressions of torque and flux simplifies to:

$$C_{em} = \frac{P.M}{L_r} . \Phi . I_{sq}$$

$$\Phi = \frac{M}{1 + T_r . s} . I_{sd}$$

(1)

Where

Tr: Rotor time constant

The decoupling by compensation is intended to decouple the axes d and q. This decoupling makes it possible to write the equations of the machine and of the regulation part in a simple manner and thus easily calculate the coefficients of the regulators. The expression of the stator voltages in the Laplace form are given as:

$$\begin{cases} V_{sd} = (R_s + \sigma . L_s . s) I_{sd} - \omega_s . (\sigma . L_s . I_{sq}) \\ V_{sq} = (R_s + \sigma . L_s . s) I_{sq} + \omega_s . (\sigma . L_s . I_{sd}) + \omega_s . \frac{M . \Phi}{L_r} \end{cases} \quad (7)$$

And it can be simplifies to :

$$\begin{cases} V_{sd} = V_{sd\_r} + V_{sd\_c} \\ V_{sq} = V_{sq\_r} + V_{sq\_c} \end{cases}$$

ISSN 2581-7795

With:

$$\begin{cases} V_{sd\_r} = (R_s + \sigma.L_s.s)I_{sd} \\ V_{sq\_r} = (R_s + \sigma.L_s.s)I_{sq} \end{cases}$$

$$\begin{cases} V_{sd\_c} = -\omega_s.(\sigma.L_s.I_{sq}) \\ V_{sq\_c} = +\omega_s.(\sigma.L_s.I_{sd}) + \omega_s.\frac{M.\phi}{L_r} \end{cases}$$

Where

LS: Inductance stator

S: Laplace operator

The system studied in this work is composed of three PI controllers, which one is for speed regulation and the two others for currents regulation(Isd, Isq). The Figure (1) shows the loop control of current stator during the d axis and the speed control loop is shown on the figure (2).

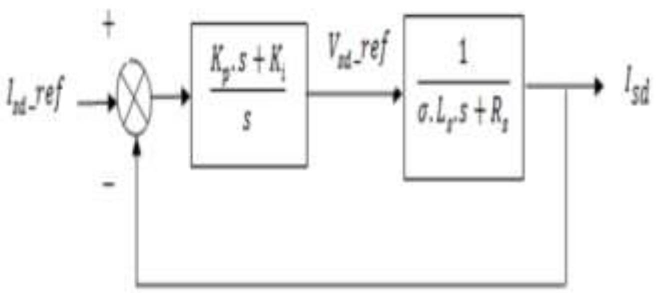

Fig-14: Current stator loop control

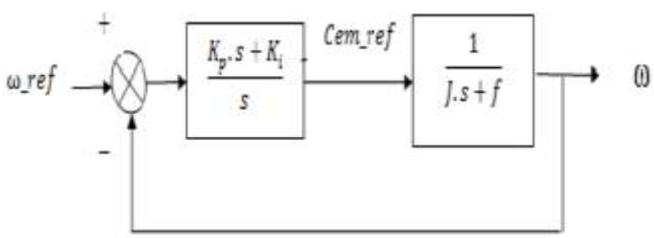

Fig-15: Speed loop control

With KP and Ki represent respectively the proportional and the integral gain. The parameters were calculated by the method of poles compensation. (9)

4.5.VECTOR CONTROL BASED ON A FUZZY CONTROLLER

(10)The basic principle of a fuzzy controller approaches the human approach in the sense that the treated variables are not logical in the sense of binary logic (for example) but linguistic variables. Moreover these linguistic variables are processed using rules that refer to certain knowledge of the system behavior.

The design of a fuzzy logic controller starts by assigning the input and output variables. The speed error reference value and its time variation has been selected as the inputs and the electromagnetique torque as the output.

4.6.1 Fuzzification and data base preparation:

The step of fuzzification is concerned with the processing of digital values to a fuzzy input values using the databases.

The fuzzification must be made a priori The universe of discourse of all the input and output variables are established as (- 1, +1). The suitable scaling factors are chosen to brought the input and output variables to this universe of discourse.

In this study each universe of discourse is divided into five overlapping fuzzy sets: NL (Negative Large), NS (Negative Small), ZE (Zero), PS (Positive Small) and PL (Positive Large). For the choice of the membership functions form, the triangular one has been chosen for all the membership functions with the exception of the extremities of each function whose trapezoidal formisused .The table 1 shows the distribution of the fuzzy sets for speed regulation.

Signification	Symbol	Speed error	Speed error time variation	Electromagnetic torque time variation
Negative Large	NL	[-20 ;16]	[-2 ; 0.8]	[-120 ; -48]
Negative Small	NS	[-12 ; 0]	[-1.2; 0]	[-72 ; 0]
Zero	ZE	[-16 ;16]	[-0.4;0.4]	[-24 ; +24]
Positive Small	PS	[0 ; 12]	[0 ; 1.2]	[0 ; 66]
Positive Large	PL	[8 ; 20]	[0.8 ; 2]	[48 ; 120]

Table -1: Distribution of fuzzy sets for speed regulation

4.6.2 Rule base and inference matrix:

For the inference method, there are too many methods, and the most applied is the Max-min method, the inference matrix of the fuzzy controller is given in the Table 2.

E \ ΔE	NL	NS	ZE	PS	PL
NL	NL	NL	NL	NS	ZE
NS	NL	NL	NS	ZE	ZE
ZE	NL	NS	ZE	PS	PL
PS	ZE	ZE	PS	PS	PL
PL	ZE	ZE	PL	PL	PL

Table -2: Inference matrix of the fuzzy controller

4.6.3 Defuzzification:

When the fuzzy outputs are computed, they must be transformed into a numerical value. There are several methods of defuzzification, and in this study the method used is the center of gravity.

4.7. CONTROL OF ASYNCHRONOUS MACHINE USING THE ARTIFICIAL NEURAL NETWORK:

A neural network is composed of several simple elements called neurons working in parallel. By adjusting the values of the connections (or weights) between the elements (neurons), we can train a neural network for a specific task. The ANN are used in many important engineering and scientific applications, some of these are: signal enhancement, noise cancellation, pattern classification, system identification, prediction, and control.

Neural network learning depends on the network architecture and the nature of the problem and there are many learning rules, which are divided to a supervised learning, none supervised learning and learning by reinforcement. The general principle of learning algorithms is based on the minimization of a cost function which can be defined as the quadratic of the differences between the outputs of the network and values desired as it is shown on the figure 3.

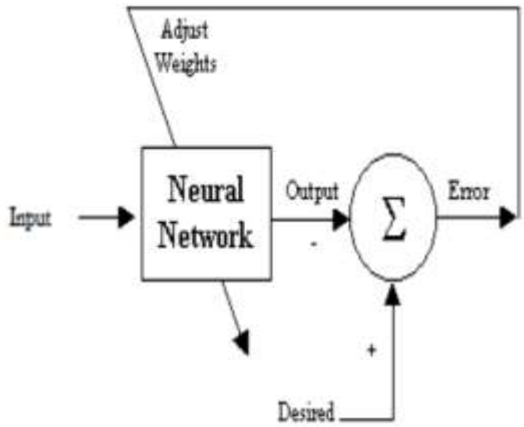

Fig-16: Learning principle of neural networks

To control the asynchronous machine with a neural controller, it had chosen to work with the block “NARMA-L2 (Feedback Linearization) Control”, given its ease of use and the results and performances that presents. The block "" NARMAL2 "or “Nonlinear Autoregressive-Moving Average "is a block that converts a nonlinear system to a linear one, for that our block involves first the identification of the system to control. To identify a system there are many methods, that we can find the regression algorithm with a graded step, the genetic algorithm based approach with its binary representation and the approach based on the combination of the genetic algorithm in its real representation and the artificial neural network with a polynomial activation function and it's based in this study on the last approach. After identifying system's model to monitor, we train a neural network in order to make the system output follows the reference input. Then we choose the architecture of the network, for that it had been chosen a network with eight neurons in the hidden layer, and the learning algorithm chosen is Levenberg Marquard seen its accuracy and quick convergence .

5 RESULTS AND DISUSSION

The results are presented based on the available simulation and experimental data from the system implementation. Key performance parameters, including speed control, torque characteristics, flux estimation, and SVPWM switching analysis, were examined.

5.1 Performance Evaluation

- **Speed Control and Electromagnetic Torque:** The system's speed and torque control were analyzed through simulation, showing stable operation under different load conditions.

SPEED CONTROL AND ELECTRO MAGNTIC TORQUE

- **Flux Estimation:** The torque flux estimation results were obtained from simulation, confirming the expected magnetic field interactions within the motor.

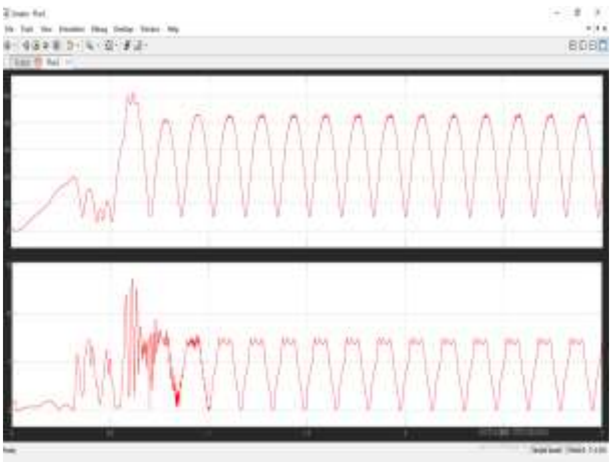

Fig-17: Flux Estimation

- **SVPWM Switching Analysis:** The Space Vector Pulse Width Modulation (SVPWM) technique was validated through simulation, ensuring optimized switching sequences for improved efficiency.

5.2 Protection Circuit Testing

- **Overcurrent Protection:** The driver circuit included overcurrent protection mechanisms, preventing excessive current from damaging the components.
- **Voltage Regulation:** The voltage source inverter successfully regulated input power to maintain stable motor operation.
- **Thermal Performance:** The simulation results showed moderate heating in power components, suggesting the need for enhanced thermal management solutions.

5.3. Discussion and Comparison with Related Works

- The proposed BLDC motor driver effectively reduced torque ripple, as evidenced by flux estimation results.
- The use of SVPWM in the system aligns with modern motor control strategies, enhancing efficiency compared to conventional trapezoidal commutation methods.
- The integration of protection mechanisms ensures increased reliability, reducing the chances of system failure under variable load conditions.

5.4 Significance, Strengths, and Limitations of the Proposed Work

1. Significance and Strengths

- **Enhanced Motor Control:** The system demonstrated precise speed and torque regulation, ensuring smooth motor performance.
- **Improved Efficiency:** The implementation of SVPWM contributed to better power utilization and reduced switching losses.

- **Reliable Protection Features:** Overcurrent and voltage regulation mechanisms ensured safe operation under different load scenarios

2. Limitations

- **Limited Experimental Validation:** The results are primarily based on simulations; additional real-time testing is required for further validation.
- **Thermal Constraints:** Extended operation under high loads may require better cooling techniques to manage heat dissipation.
- **Sensorless Control Sensitivity:** The accuracy of back-EMF detection at low speeds needs further optimization for stable performance.

5.5 Cost-Benefit Analysis

- **Component Costs:** The design utilized cost-effective power components, reducing the overall manufacturing expense.
- **Efficiency Gains:** The SVPWM technique improved power efficiency, making it a viable solution for industrial applications.
- **Potential Market Adoption:** The BLDC motor driver offers a competitive edge with its efficiency and protection features, making it suitable for automotive and industrial automation applications.

6 CONCLUSION

The torque ripple is a common issue in BLDC motors that can affect motor performance, introduce vibrations and noise, and reduce overall system efficiency. However, several techniques can be employed to reduce torque ripple in BLDC motors. These include advanced motor design, sensorless control techniques, PWM techniques, current ripple reduction techniques, advanced control algorithms, commutation optimization, and mechanical damping.

- It's important to carefully evaluate and select the appropriate torque ripple reduction techniques based on the specific motor design, operating conditions, and application requirements. Simulation, analysis, and experimentation can be helpful in optimizing the chosen technique. By effectively reducing torque ripple, BLDC motors can operate more smoothly, quietly, and efficiently, leading to improved motor performance and enhanced overall system reliability.

REFERENCES

[1] A. Mishra and P. Choudhary, "Artificial Neural Based Controller for Speed Control of an Induction Motor using Indirect Vector Control Method," vol. 2, no. 4, pp. 402–408, Dec. 2012.

[2] [2] T. Ourth, V. T. Nguyen Phuoc, M. Pietrzak-David, and B. de Fornel, "Sensorless Speed Control of Induction Motor Drives Using Observer Based Vector Control," in *Proc. Conf. ICEM*, Manchester, 1992.

[3] [3] K. Pratap and Shelja, "Artificial Neural Network (ANN) Inspired from Biological Nervous System," *Int. J. Appl. Innov. Eng. Manag. (IJAIEM)*, vol. 2, no. 1, Jan. 2013.

[4] [4] Math Works, *Neural Networks Toolbox User's Guide*, Natick, MA, USA: Math Works, 2015.

[5] [5] P. Koehn, "Combining Genetic Algorithms and Neural Networks: The Encoding Problem," M.S. thesis, Univ. Tennessee, Knoxville, TN, USA, Dec. 1994.

[6] [6] F. Khammar and N. E. Debbache, "Application of Artificial Intelligence Techniques for the Control of the Asynchronous Machine," *J. Electr. Comput. Eng.*, vol. 2016, Art. ID 8052027, 2016.

[7] [7] P. M. Menghal and A. J. Laxmi, "Application of Artificial Intelligence Controller for Dynamic Simulation of Induction Motor Drives," *Asian Power Electron. J.*, vol. 7, no. 1, Sep. 2013.

[8] [8] T. F. Chan and K. Shi, *Applied Intelligent Control of Induction Motor Drives*, 1st ed. New York, NY, USA: IEEE Wiley Press, 2011.

[9] [9] A. Merabet, "Controlling Nonlinear Predictive Model for Asynchronous Machine," Ph.D. dissertation, Univ. Quebec, Québec, Canada, 2007.

[10] A. Kusagur, S. F. Kodad, and B. V. Sankar Ram, "Modeling, Design & Simulation of an Adaptive Neuro-Fuzzy Inference System (ANFIS) for Speed Control of Induction Motor," *Int. J. Comput. Appl.*, vol. 6, no. 12, pp. 29–44, 2010.