A BL-CSC converter-fed BLDC motor drive with power factor correction using sliding mode controller

Design, simulation, and performance evaluation for a broad spectrum of low- and medium-power applications

[1]B.Prasanna vidya, [2]G.Kiran Kumar

[1]PG Student, [2]Assistant Professor
[1]Department of Electrical and Electronics Engineering,
[1]University College of Engineering, Osmania University, Hyderabad, India
[1]prasannavidyabeera1996@gmail.com, [2]kirankumargollapelli@gmail.com

***Abstract*—This document details a power factor correction (PFC)-based bridgeless canonical switching cell (BL-CSC) converter that is used to power a brushless DC (BLDC) motor drive. The BL-CSC converter functions in discontinuous inductor current mode to maintain a unity power factor in the AC main, using only one voltage sensor. The speed regulation of the BLDC motor is achieved by altering the DC bus voltage of the voltage source inverter (VSI), which powers the BLDC motor through a PFC converter. Consequently, the BLDC motor is electronically commuted, allowing the VSI to operate at fundamental frequency switching, thereby minimizing switching losses. Additionally, the bridgeless design of the CSC converter reduces conduction losses by partially removing the diode bridge rectifier at the input stage. This proposed setup demonstrates a significant efficiency improvement compared to traditional methods. The performance of the proposed drive system is simulated and the results are gathered using MATLAB. Improved power quality is maintained on the AC mains over a wide spectrum of control speeds and input voltages.**

***Index Terms*—Bridgeless canonical switching cell converter, brushless dc (BLDC) motor, discontinuous inductor current mode(DICM), Sliding mode control(SMC), power factor correction(PFC), power quality.**

I. Introduction

Over the past several years, Brushless DC (BLDC) motors have emerged as a focal point of interest, largely because they offer markedly better operating characteristics and deliver higher-quality electrical output when compared to conventional DC motor drive systems. Brushless DC (BLDC) motors are extensively used across both industrial systems and consumer products, largely due to their superior energy efficiency, strong operational reliability, extended service lifetime, and the relatively low levels of electromagnetic interference (EMI) they generate during operation. These motors feature a rotor equipped with permanent magnets and a stator containing three-phase windings. Because the current is commutated electronically rather than by mechanical brushes, they avoid problems typical of brushed motors, including electrical sparking, elevated acoustic noise, and the need for frequent maintenance and brush replacement. The use of Hall-effect sensors for electronic commutation delivers highly accurate detection of rotor position, which in turn allows for finely tuned control of motor operation. This precise feedback makes it possible to maintain smooth, stable, and accurately regulated speeds across an extensive operating range.Because of these benefits, BLDC motors are now widely regarded as the technology of choice for a broad spectrum of low- and medium-power applications. Typical use cases span household appliances (such as refrigerators, washing machines, and fans), medical devices and instrumentation, precision position actuators, heating, ventilation, and air-conditioning (HVAC) units, a variety of motion-control systems in industrial and commercial settings, as well as modern electric transportation platforms, including electric vehicles, scooters, and other battery-powered mobility solutions.

II. System Modelling and Design

This paper focuses on creating a bridgeless design of a CSC converter, with the aim of partially eliminating the DBR at the front end to decrease conduction losses. The paper also examines how this converter can be used to operate a BLDC motor drive, offering a cost-effective approach for low-power use. Figure1 shows the VSI based on the BL-CSC converter designed to drive a BLDC motor. This setup omits the DBR from the BL-CSC converter, thus reducing the usual conduction losses. The BL-CSC converter operates in discontinuous inductor current mode (DICM), making the currents through the Li1 and Li2 inductors discontinuous, while ensuring that the voltage across the capacitors C1 and C2 stays continuous throughout every switching cycle. The DC link voltage is adjusted to control the speed of the BLDC motor, using electronic commutation to reduce VSI switching losses. The following sections explain the functioning, architecture, and management of the BLDC motor drive powered by a BL-CSC converter. The test results from a prototype validate the effectiveness of the proposed drive, showing improved power quality on the AC mains at various speeds and voltages. Figure2 provides a brief comparison between the proposed setup and current bridgeless converter designs. This figure details the total number of components (Switch—Sw, Diode—D, Inductor—L, and Capacitor—C) and specifies which components conduct during each half-cycle of the supply voltage. Bridgeless buck and boost converter configurations are not suitable for the intended application due to their need for a high voltage conversion ratio to manage speed in a wide range. In contrast to the various bridgeless configurations of Cuk, SEPIC, and Zeta converters, the proposed BL-CSC converter utilizes a comparatively smaller number of components and has the fewest conducting devices during each half-cycle of the supply voltage. Consequently, the proposed configuration achieves the minimum conduction losses due to the minimal number of components conducting during each half line cycle.

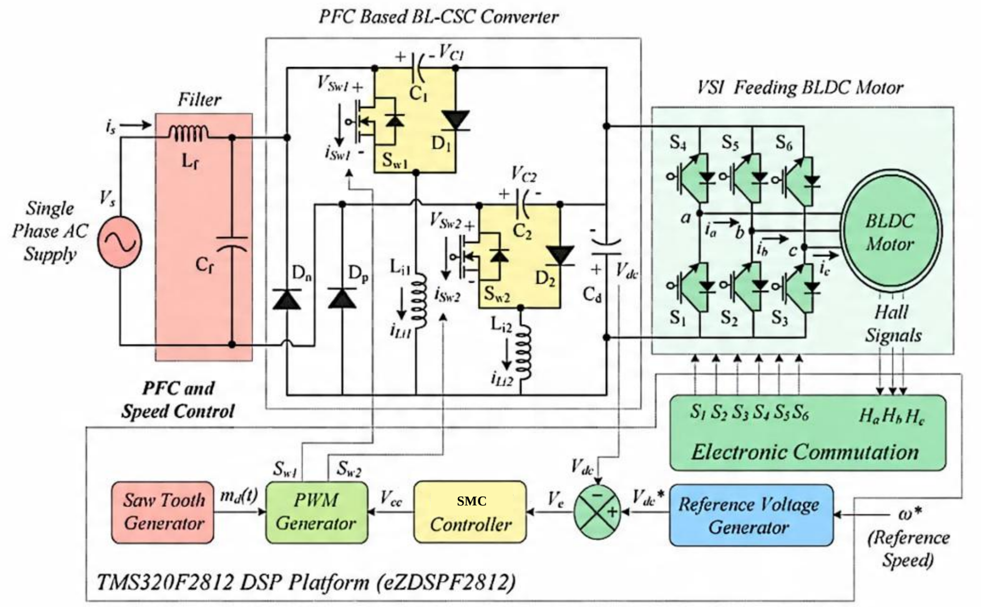


Figure 1: A BL-CSC Converter-Fed BLDC Motor Drive With Power Factor Correction Using SMC Controller

The functions of the BL-CSC converter are split into two main categories:

The bridgeless converter is designed so that two switches operate in both the positive and negative cycles of the supply voltage. Figure3 a to f depict the functioning of the suggested BL-CSC converter in these cycles. In the positive half-cycle, the input current flows through Sw1, Li1, and the fast recovery diode Dp. Conversely, the negative half-cycle conduction path uses switch Sw2, inductor Li2, and diode Dn, as shown in Fig3.The waveforms display the supply voltage, the inductor currents (iLi1 and iLi2), and the intermediate capacitor voltages (VC1 and VC2). The converter operates in Discontinuous Inductor Current Mode (DICM), with inductor currents (iLi1 and iLi2) not continuous, while the voltages between intermediate capacitors (VC1 and VC2) remain stable with minimal voltage ripple during a full switching cycle[1]. The proposed BL-CSC converter is designed to operate in Discontinuous Inductor Current Mode (DICM), making the current through inductors Li1 and Li2 intermittent during a switching cycle. Below, the operational modes across a full switching cycle for both positive and negative halves of the input voltage are shown.

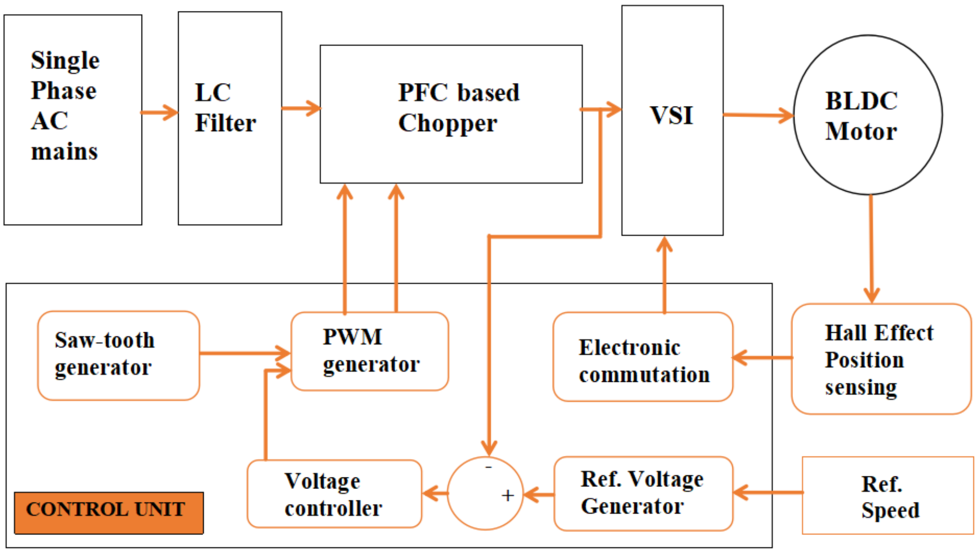


Figure 2: Block diagram of proposed converter scheme

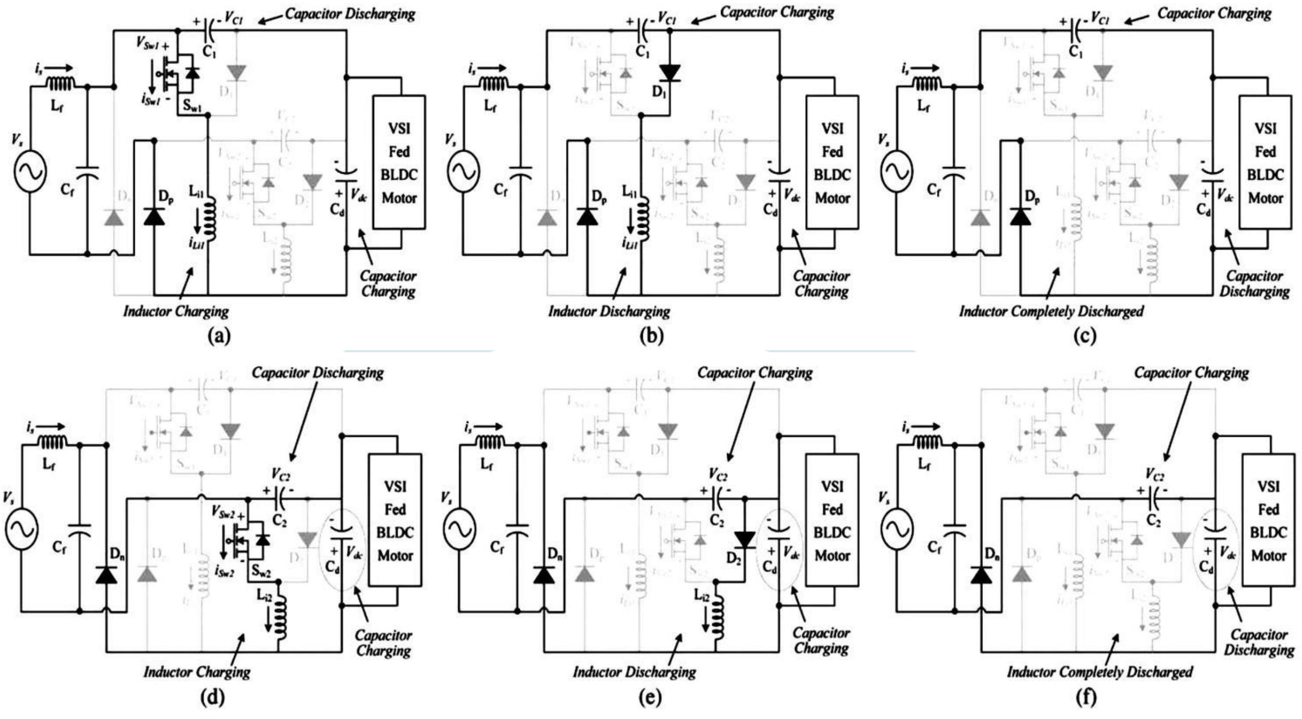


Figure 3: Supply Modes in Positive and Negative Half-Cycles

In Mode I-A: When switch Sw1 is turned on, the input inductor Li1 charges via diode Dp, causing current iLi to rise. At the same time, capacitor C1 releases its charge via switch Sw1, powering the dc link capacitor Cd. As a result, VC1 voltage drops, and Vdc voltage increases.

Mode I-B: Turning off switch Sw1 releases the energy stored in inductor Li1 to the dc link capacitor Cd through diode D1, as shown in Fig3. In this phase, iLi reduces as the dc link voltage increases, while capacitor C1 starts charging, leading to a rise in voltage VC1.

Mode I-C indicates the DCM operation where the current in the input inductor Li1 drops to zero, as shown in Fig3. In this mode, capacitor C1 retains its charge and stores energy, as capacitor Cd supplies energy to the load. The converter exhibits similar behavior during the supply voltage's negative half-cycle. Here, inductor Li2, capacitor C2, and diodes Dn and D2 function similarly.

In Mode II-A: When switch Sw2 is turned on, the input-side inductor Li2 starts charging via diode Dp, resulting in an increase in current iL2. Concurrently, the intermediate capacitor C2 is discharged through the Sw2 switch to charge the dc link capacitor Cd. As a result, the voltage across the intermediate capacitor VC2 decreases, while the dc link voltage Vdc increases.

Mode II-B: When Sw2 is turned off, energy from the Li2 inductor is discharged to the dc link capacitor Cd through diode D2, as shown in Fig3. During this operational phase, the current iL2 decreases, while the voltage across the dc link keeps rising. The intermediate capacitor C2 begins to charge, resulting in an increase in voltage VC.

Mode II-C is known as Discontinuous Conduction Mode (DCM), where the input inductor Li2's current drops to zero, as shown in Fig3. At this point, C2 retains its energy and charge as Cd supplies energy to the load. Electronic devices require power supplies to convert grid AC voltage to electronics-compatible DC voltage. Linear power supplies, including those with passive filters, often exhibit a low power factor and may generate harmonic currents in the system. The influence of one power supply might be minor, but when multiplied by millions, these devices can notably impact power quality. Using power supplies with power factor correction circuits can improve power factor and reduce harmonic currents. The boost power factor correction converter is one such circuit that can be incorporated into power supplies to significantly enhance their power quality[2].

A. Design of Bl-Csc Converter:

The BLDCM drive proposed incorporates a PFC-based CSC converter functioning in DICM[3]. This front-end PFC-based CSC converter, with a power output of 400W, is designed for a 314W BLDCM (complete specifications can be found in the appendix). The DC link voltage needs to be managed within a range of 50V (Vdcmin) to 200V (Vdcmax). This is designed for a supply voltage (Vs) of 220V.

$$\mathrm{Vin}(t) = |V\mathrm{msin}(2\pi f t)| = |220(\sqrt{2}) * \sin 314t|$$

$$Vdc = \frac{D}{(1-D)} * Vin$$

$$D(t) = \frac{Vdc}{Vin(t) + Vdc} = \frac{Vdc}{|V\mathrm{Sin}(wt)| + Vdc}$$

$$Pi = \left(\frac{P\mathrm{max}}{Vdc\mathrm{max}}\right) * Vdc$$

- **Inductance Calculation:**

At the minimum supply voltage of 85 V, the minimum critical input inductance (Lic) is set for universal AC mains operation (85–270 V). Lic min is determined by the inductor's critical input value.

Lic is expressed as

$$Lic = \frac{\mathrm{Vin}(t) * D(t)}{2\mathrm{Iin}(t) * fs} = \frac{\mathrm{Rin} * D(t)}{2fs}$$

$$= \left(\frac{Vs^2}{Pi}\right) * \frac{D(t)}{2 * fs}$$

$$= \frac{85^2}{500} * \frac{0.7206}{2 * 20000}$$

$$\approx 260\mu H$$

In order to achieve discontinuous current conduction, the input inductors Li1 and Li2 should be chosen with values less than the minimum critical inductance, Lic min[4]. Consequently, Li1 and Li2 are both set to approximately one third of Lic min, specifically 70 micro Henry, to ensure discontinuous current conduction. The intermediate capacitor's value is determined at the peak of its ripple, which takes place at the rated DC link voltage of 310V and the maximum supply voltage of 270V, i.e.,

- **Capacitor Calculations:**

The expression for intermediate capacitance (C1 and C2)[4] is given as

$$C1 = C2 = \frac{Vdc * D(t)}{\Delta Vc(t) * fs * RL}$$

$$= \frac{Vdc * D(t)}{\eta * \{Vin(t) + Vdc\} * (fs * RL)}$$

$$= \frac{310 * 0.4481}{(0.1\{270\sqrt{2} + 310\} * 20000 * 192.2)}$$

$$= 0.522\mu\ \mathrm{F}$$

The value of dc link capacitor is calculated as

$$Cd = \frac{P\mathrm{min}}{Vdc\mathrm{min}} * \frac{1}{(2 * \omega * \Delta Vdc\mathrm{min})}$$

$$= \left(\frac{113}{70}\right) * \left(\frac{1}{(2 * 314 * 0.02 * 70)}\right)$$

$$\approx 1836\mu F$$

- **Low pass filter calculations:**

A low-pass LC filter is used to prevent the reflection of higher order harmonics in the supply system[4]. The filter capacitance's maximum value is specified as

$$C\mathrm{max} = \frac{Im}{\omega L * Vm} * \tan(\theta)$$

$$= \left(\frac{Po\sqrt{2}Vs}{\omega L * Vm}\right) * \tan(\theta)$$

$$= \left(\frac{500\sqrt{2}/220}{314 * 220\sqrt{2}}\right) * \tan(1^\circ)$$

$$= 574.27\mathrm{nF}$$

The value of the filter inductor is determined by taking into account the source impedance (LS), which is 4 to 5 percent of the base impedance. Therefore, the extra inductance needed is provided as

$$Lf = Lreq + Ls \Longrightarrow \frac{1}{4\pi^2(fc)^2 * cf} = Lreq + 0.05\left(\frac{1}{\omega L}\right) * \left(\frac{(Vs)^2}{Po}\right)$$

$$\mathrm{Lreq} = \left(\frac{1}{4\pi^2 * (2000)^2 * 330 * (10)^{-9}}\right) - 0.05 * \left(\frac{1}{314}\right) * \left(\frac{220^2}{500}\right)$$

$$= 3.77\mu H$$

- **BLDC Motor Rating**

4 pole, P_{rated} (Rated Power) = 424.11 W(0.5hp), V_{rated} (Rated DC Link Voltage) = 310 V, T_{rated} (Rated Torque) = 1.35 N · m, ω_{rated} (Rated Speed) = 3000rpm, K_b (Back EMF Constant) = 78 V/krpm, K_t (Torque Constant) = 0.74 N · m/A, R_{ph} (Phase Resistance) = 14.56Ω, L_{ph}(Phase Inductance) = 25.71mH, J (Moment of Inertia) = 1.3×10^{-4} N · m/s^2.

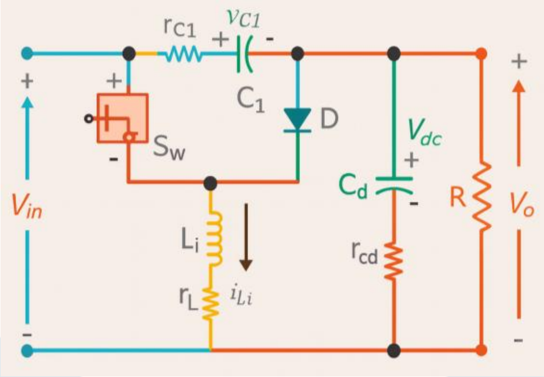


Figure 4: Circuit configuration of the CSC converter with parasitic resistances

- **Modeling of the Proposed Converter**
  The CSC converter is modeled in DCM of operation. Fig.4 shows the circuit configuration of the CSC converter with parasitic resistances as r_L, r_{c1}, and r_{cd} of the input inductor, intermediate capacitor, and dc link capacitor, respectively. The state-space model of the proposed converter is developed where the state-space parameters are the inductor current i_{Li}, the intermediate capacitor voltage V_{C1}, and the dc link voltage V_{dc}. The state-space equations are given as

$$\dot{X} = A_n X + B_n V_{\text{in}}, V_o = CX$$

where X is the state vector, A_n is the state matrix, B_n is the input matrix, and C is the output matrix.

The equations corresponding to the three modes of operation viz., switch turn on ($X' = A_1 X + B_1 V_{\text{in}}$), switch turn off ($X' = A_2 X + B_2 V_{\text{in}}$), and the DCM of operation ($X' = A_3 X + B_3 V_{\text{in}}$), are respectively given in

$$\begin{bmatrix} i_{Li} \\ \dot{V}_{C1} \\ \dot{V}_{\text{dc}} \end{bmatrix} = \begin{bmatrix} -\frac{r_L}{L_i} & 0 & 0 \\ 0 & b_1\left(1+\frac{r_{\text{cd}}}{R}\right) & b_1 k_o \\ 0 & \frac{1}{a_1} & \frac{R-r_{c1}}{Ra_1} \end{bmatrix} \begin{bmatrix} i_{Li} \\ V_{C1} \\ V_{\text{dc}} \end{bmatrix} + \begin{bmatrix} \frac{1}{L_i} \\ -b_1\left(1+\frac{r_{\text{cd}}}{R}\right) \\ -\frac{1}{a_1} \end{bmatrix} V_{\text{in}}$$

$$\begin{bmatrix} i_{Li} \\ \dot{V}_{C1} \\ \dot{V}_{\text{dc}} \end{bmatrix} = \begin{bmatrix} k_3 & k_4 & k_5 \\ k_1 & b_2 & k_2 \\ \frac{r_{c1}}{a_2} & \frac{1}{a_2} & \frac{1}{a_2}\left(1+\frac{r_{c1}}{R}\right) \end{bmatrix} \begin{bmatrix} i_{Li} \\ V_{C1} \\ V_{\text{dc}} \end{bmatrix} + \begin{bmatrix} \frac{1}{L_i} \\ 0 \\ 0 \end{bmatrix} V_{\text{in}}$$

$$\begin{bmatrix} \dot{i}_{Li} \\ \dot{V}_{C1} \\ \dot{V}_{\text{dc}} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{C_d}{a_2 C_1} & k_6 \\ 0 & \frac{1}{a_2} & \frac{1}{a_2}\left(1+\frac{r_{c1}}{R}\right) \end{bmatrix} \begin{bmatrix} i_{Li} \\ V_{C1} \\ V_{\text{dc}} \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{C_d}{a_2 C_1} \\ -\frac{1}{a_2} \end{bmatrix} V_{\text{in}}$$

where various parameters, i.e., a_1, a_2, and $k_o - k_6$, are defined as

$$a_1 = -C_d\left(r_{c1} + r_{\text{cd}} + \frac{r_{c1} r_{\text{cd}}}{R}\right)\ b_1 = \frac{C_d}{a_1 C_1}$$

$$a_2 = -C_d(r_{c1} + r_{\text{cd}})\ b_2 = \frac{C_d}{a_2 C_1}$$

$$k_o = \left(1 - \frac{r_{c1}}{R}\right)\left(1 + \frac{r_{c2}}{R}\right) + \frac{1}{R \cdot b_1}$$

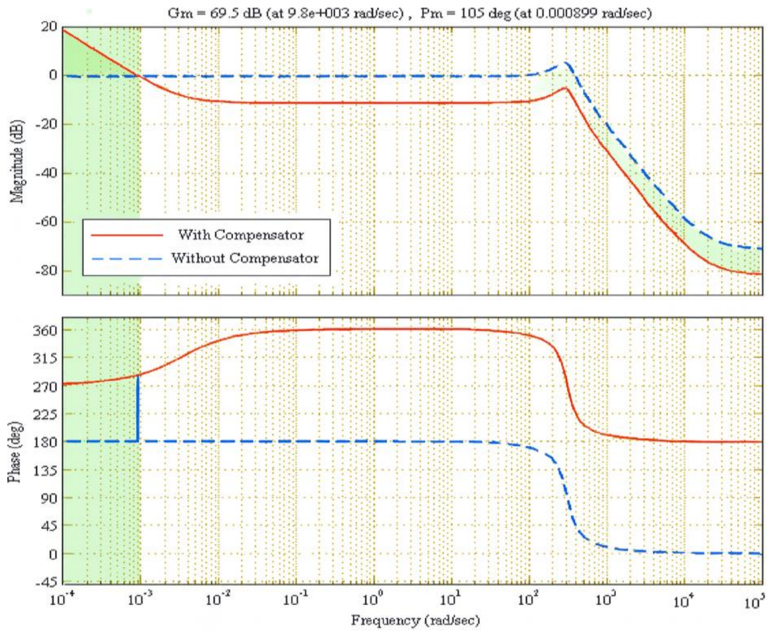


Figure 5: Bode plots of the CSC converter with and without compensator

$$k_1 = \frac{1}{C_1} + b_2 r_{c1} \; k_2 = \frac{1}{RC_1} + \left(1 + \frac{r_{c1}}{R}\right) b_2$$
$$k_3 = \frac{-C_1 r_{c1} k_1 - r_L}{L} \; k_4 = \frac{-C_1 r_{c1} b_2 - 1}{L}$$
$$k_5 = \frac{-C_1 r_{c1} k_2}{L} \; k_6 = \frac{C_d}{C_1}\left\{\frac{1}{a_3}\left(1 + \frac{r_{c1}}{R}\right) + \frac{1}{RC_d}\right\}$$

Moreover, the output equation is given as

$$V_o = [0 \quad 0 \quad 1]\begin{bmatrix} i_{Li} \\ V_{C1} \\ V_{\mathrm{dc}} \end{bmatrix}.$$

The state-space model for a complete switching cycle is derived using the state and input matrices, which are given as

$$A = A_1 d_1 + A_2 d_2 + A_3(1 - d_1 - d_2)$$
$$B = B_1 d_1 + B_2 d_2 + B_3(1 - d_1 - d_2)$$

where d_1 and d_2 are the duty ratios corresponding to the conduction period of the switch (S_w) and diode (D), respectively. Moreover, the parameter $(1 - d_1 - d_2)$ denotes the period during the DCM of operation.

The transfer function (audio susceptibility) of the CSC converter is calculated as

$$G(s) = \frac{V_o}{V_{\mathrm{in}}}(s) = C[sI - A]^{-1}B.$$

For a converter of 500 W(P_o) with output voltage of 310 V (V_o) and input voltage of 198 V$(V_{\mathrm{in}}$), the output emulated resistance is calculated as $192.2\Omega(R)$. The values of other designed parameters are given as follows: $d_1 = 0.08, d_2 = 0.12, L_i = 70\mu\mathrm{H}, C_1 = 0.66\mu$ F$, C_2 = 2200\mu$ F$, r_L = 0.05\Omega, r_{c1} = 3.2$ m $\cdot\ \Omega$, and $r_{c2} = 30$ m $\cdot\ \Omega$.

By putting these values of different parameters in (30), the plant transfer function is obtained as

$$G(s) = \frac{1.205 \times 10^4 s^2 + 7.799 \times 10^6 s - 3.718 \times 10^{12}}{s^3 + 4.565 \times 10^7 s^2 + 7.102 \times 10^9 s + 4.282 \times 10^{12}}.$$

The transfer function of the PI controller is given as

$$G_c(s) = K_p + \frac{K_i}{s}$$

where K_p and K_i represent the proportional and integral gains of the PI controller and their values selected as 0.3 and 0.001 , respectively.

Using the plant and compensator transfer functions as given, the Bode plots of the complete system with and without compensator have been plotted, as shown in Fig.5. A positive and a high value of gain and phase margins are obtained on the order of 70 dB and 105°, respectively, which shows good stability of the proposed system.

B. Sliding mode controller:

Sliding Mode Control (SMC) is an effective control strategy applied to non-linear systems to maintain stability and performance in the face of uncertainties and external disturbances. This method, a type of variable structure control (VSC), forces the system to "slide" along a designated path known as the sliding surface.

- **Basic Concept**

The SMC operates in two phases:

1. **Reaching Phase:** The controller guides the system's state to the sliding surface.
2. **Sliding Phase:** When the system reaches the sliding surface, it is confined to a predetermined path, guaranteeing robustness to disturbances.

- **Key Features:**

1. **Robustness:** ability to handle system uncertainties and external disturbances efficiently.
2. **Time Convergence:** Ensures a rapid response by forcing the system to reach the sliding surface in finite time.
3. **Chattering Phenomenon:** The discontinuous nature of the control signal results in a high-frequency oscillation, which can be mitigated through modifications such as boundary layer control or higher-order SMC.

- **Mathematical Formulation:**

A typical system is given by the following:

$$\dot{x} = f(x) + g(x)u$$

Here, x represents the state, u denotes the control input, and f(x), g(x) signify the system's dynamics.
The sliding surface is expressed as:

$$s(x) = 0$$

where s(x) represents a function related to the state of the system.
The control law is typically selected as:

$$U = Ueq + Usw$$

where: Ueq serves as the equivalent control that maintains the system on the sliding surface.
Usw is the switching control responsible for guiding the system to the sliding surface.

- **Applications:**

SMC is widely used in the following fields:

1. Robotics (trajectory tracking)
2. Power electronics (DC-DC converters)
3. Aerospace (attitude control)
4. Automotive (ABS, traction control)

In this scheme, the sliding plane S is defined based on the tracking speed error e(t),

its integral $\int$ e.dt, and its rate of change $\dot{e}(t)$ and is given by,

$$S = \dot{e} + s\lambda 1e + \lambda 2 \int edt$$

Lambda 1 and lambda 2 are surface parameters that are positive real numbers, and these gain parameters establish the slope of the sliding surface.

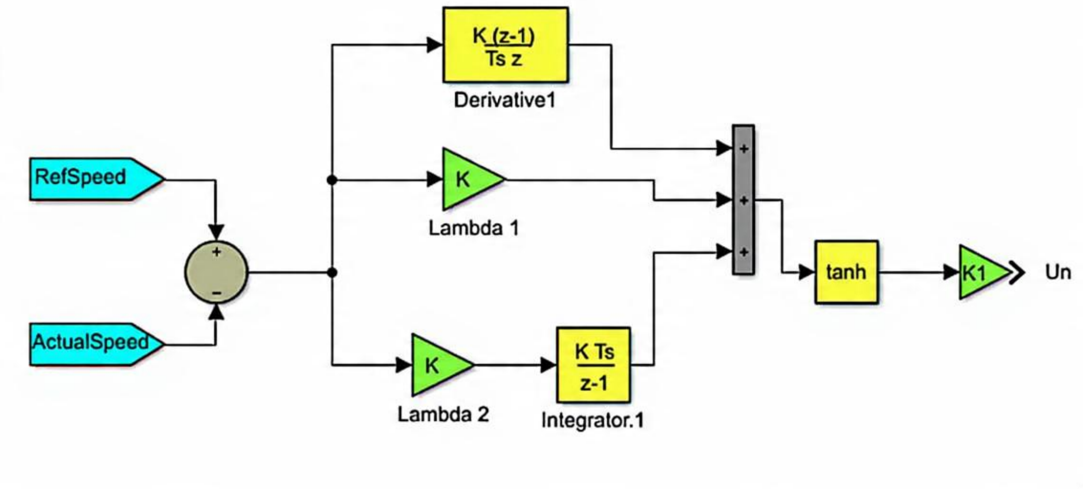


Figure 6: SMC internal structure

Sliding Mode Control (SMC) is a powerful nonlinear control technique famous for ensuring system stability and exceptional performance despite uncertainties and external disruptions. The SMC, by establishing a sliding surface, guaranties that the system maintains a particular dynamic behavior, which makes it very suitable for use in robotics, aerospace, power electronics, and automotive systems. The primary benefits of SMC include its robustness, the ability to achieve finite-time convergence, and the resistance to parameter changes. Nevertheless, the main disadvantage of SMC is the chattering effect, which can lead to mechanical wear or increased energy consumption in some applications[5].

To address this problem, a variety of sophisticated methods have been created, including higher-order SMC and boundary layer techniques. SMC is generally a popular control strategy for dealing with complex and uncertain systems, providing a good compromise between accuracy, robustness, and flexibility. Ongoing research is improving its performance, making it increasingly suitable for contemporary engineering applications[5].

- **Specifying all terms**

i. Inputs

Initially, the control system relies on two primary inputs: target speed and measurement Speed. Both signals are crucial for aligning a control system's desired output with actual measurements. the target speed for the system, typically established by a higher-level controller or user input. It represents the speed the system needs to reach. measurement Refers to the present speed of the system, generally gauged by sensors like encoders that monitor the motor or system's speed. The control system aims to reduce the discrepancy between two signals. The error signal shows the gap between the desired speed and the measured speed, guiding the controller to modify the system's performance.

ii. Subtractor

The error block subtracts the target Speed from the measured Speed and outputs the error signal. It is important to understand that the error signal represents the difference between the state of the system and the target state. Specifically, a positive error means that the system is slower than desired, and a negative error means that the system is moving too fast. The control output attempts to eliminate the error.

iii. Branching of the Signal

After the error occurs, it splits into three separate routes for different control actions: While the branches look like Derivative, Proportional, and Integral each handle the error in their own way to achieve good control. In modern control systems, branching is common and essential if you want to use a mixture of control methods such as PID control. These different methods work together to ensure quick response, stability, and zero steady-state error.

iv. Derivative

At the top, the error is sent to a block that estimates its derivative. This is used via a discrete-time derivative approximation with transfer function:

Here's what this means:

K: A constant multiplier that is multiplied onto the derivative result.

Ts: The time interval at which this system must refresh its control signal. An important aspect of discrete control.

Z: In digital signal processing, the discrete-time variable is written as 'z' in the z-domain.

This section calculates the rate of change of the error, creating the derivative term. The derivative action on fast fluctuations of the error signal allows the controller to quickly handle disturbances and changes in system behavior. Derivative action can help prevent overshoot of the speed set point or slow response.

v. Lambda 1

The central route passes the error through a proportional gain block Lambda 1, a basic gain, K: This is the proportional part of the control signal. This term corrects the error by multiplying it by K. It makes big corrections for big errors and small corrections for small errors. This is the main corrective force. However, it cannot eliminate steady-state error. This is practically equivalent to saying that the system will continually "push" against the setpoint with a force given by the amount of error. In other words, the system might not reach the target speed without other terms (derivative, integral) which help eliminate steady state error.

vi. Combining Lambda 2 and Integrator

Lambda 2 is, as Lambda 1, a gain multiple for the error. Integrator Block: The error u is multiplied by a discrete-time integrator block. This looks like: Multiplication with the error adds up the error over time. As long as the error is not zero, the integrator is adding the error to remove steady state error. The control action of the integrator is piling up while the error is present (guaranteeing that the error will go to zero). We need the integral component because there might be a residual error, a bias. For example, there might be some friction or some other continuous effect that makes the system stick to a speed outside of the intended value. The integral component multiplies the error and therefore increases the control output until the error is zero.

vii. Vector Addition

The outputs of the derivative, proportional and integral blocks are fed into a summation block. The effect is that the derivative, proportional and integral block signals combine into one signal that represents the total control action obtained from each of the paths which in this case represent different aspects of system performance. The derivative signal is responsive to rapid fluctuations of the error signal, the proportional signal is responsive to moderate fluctuations in the error signal, and the integral component removes a persistent error.

viii. Nonlinearity

After the control signal has been calculated it passes through a tanh (hyperbolic tangent) function, which is the non-linearity: The tanh function is a smooth limiting function that limits the output to a range of -1 to +1. By saturating the control signal, the tanh function prevents the system from overdriving the actuator, such as a motor, which would stop instability and damage occurring to the system if the control signal is too strong. It is important that the tanh function is used instead of a hard limit, as shown below, because it prevents sudden clipping that occurs with the hard limit and the tanh function limits the control action, preventing sudden system actions, which would be undesirable in a system with a mechanical actuator.

ix. Output K1

The signal passes through a tanh activation, then a scaling gain block called K1. Last modification: corrects the control signal to account for what is physically required to move the actuator (i.e., from motor to power, etc). K1 keeps the control signal suitable for the actuator. Last modification: corrects the control output of the controller to account for what is physically required to move the actuator (i.e. to account for voltage, current, power, etc.

x. Final Output

Un serves as the control signal directed to the actuator. The actuator deciphers this signal to modify system behavior, like regulating motor speed or power. The control loop frequently adjusts this output to align the system's behavior and reduce the gap between the target speed and the measured speed.

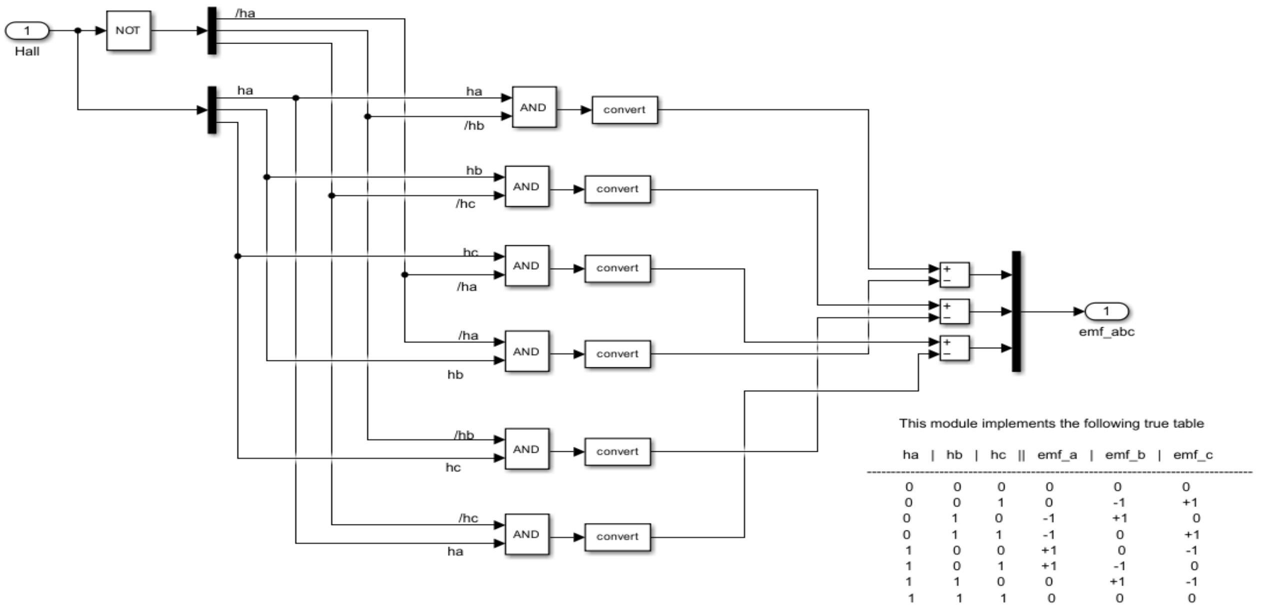
- **Power Factor and its Correction**

Passive power factor correction (PFC) involves using a component that balances the system's reactive power with an equal and opposite reactive power. For example, to offset an inductive load with a reactive power of 1.754 kVAR, a capacitive load of the same reactive power is required. One approach to power factor correction is to employ a large set of capacitors that can be linked to the circuit when necessary. This method of power factor correction is suitable for extensive linear loads, where its cost can be warranted by the system's size and expense. But for individual power supplies power factor is still important. It is not because any individual power supply has a great impact, but because there are many power supplies. Additionally, these power supplies are problematic as nonlinear loads, not just adding capacitors or inductors will cure the power factor problem. As shown in previous article simple passive filters won’t help to significantly increase the power factor or reduce the harmonic distortion. Active power factor circuit is required to make sure that the AC current follows the AC voltage[6].

III. Control Strategy

mechanism behind electronic commutation in a BLDC motor

A Brushless DC (BLDC) motor uses electronic commutation to alternate the winding current, ensuring the rotor spins continuously. Unlike classic brushed DC motors, BLDC motors use electronic circuits and Hall sensors for changing current direction, instead of a mechanical commutator and brushes. As shown in the Fig.5 Hall sensors determine the rotor's position by sensing the magnetic field of its permanent magnets. Based on these signals, the electronic controller (inverter) selects and sequences the stator windings to energize. A BLDC motor features six Hall sensor combinations, each corresponding to one of six rotor positions per electrical cycle. The controller manages the inverter's transistors (MOSFETs or IGBTs) to consistently power two phases: one linked to the positive DC bus and another to the negative, with the third phase inactive. Switching sequences create a rotating magnetic field in the stator that interacts with the rotor's permanent magnets to generate torque and maintain rotation. Electronic commutation, or six-step/120-degree commutation, uses steps of 60 electrical degrees, with just two phases active at once. This method ensures efficiency, smooth operation, and minimal upkeep, as it lacks brushes and mechanical parts. Electronic commutation precisely controls motor speed and torque, making BLDC motors widely utilized in electric vehicles, drones, robotics, and industrial automation.



This module implements the following true table

ha	hb	hc	emf_a	emf_b	emf_c
0	0	0	0	0	0
0	0	1	0	-1	+1
0	1	0	-1	+1	0
0	1	1	-1	0	+1
1	0	0	+1	0	-1
1	0	1	+1	-1	0
1	1	0	0	+1	-1
1	1	1	0	0	0

Figure 7: Decoder

A. Decoder Functionality and Tasks

Hall-sensor decoding and phase EMF production in a three-phase BLDC motor. It reads the three digital Hall signals (ha, hb, hc), identifies the current rotor position among the six valid ones, and produces a numerical sign for each stator phase (emf_a, emf_b, emf_c) to show whether the phase should be positive (+1), negative (−1), or idle (0) for that commutation step. These outputs illustrate the back-EMF polarity pattern or serve as a guide for inverter switching logic, ensuring the correct two phases are on while the third remains off. Thus, the module applies electronic commutation (120°/six-step) with basic digital logic.

i. Inputs: Hall signals and their meaning

The three digital Hall sensor inputs are on the left, each reading either 1 or 0 based on the rotor magnet's position. A set of three bits creates a 3-bit code indicating the rotor's electrical position. An electrical cycle typically involves six distinct 3-bit sequences: 100, 110, 010, 011, 001, 101, and then returns to 100. Every distinct combination equates to a commutation step of 60 electrical degrees, indicating to the controller which phase to energize positively, which to energize negatively, and which to remain unenergized.

ii. NOT / inverted signals

Immediately following the Hall inputs, there are NOT/inverter paths generating the complements ha, hb, and hc (logical opposites). These inverted signals are essential as the logic conditions to recognize a specific Hall pattern often need a combination of true and inverse inputs (e.g., detect ha=1 AND hb=0 AND hc=1). The circuit can create all necessary minterms by offering both true and inverted Hall bits using basic AND gates.

iii. AND gates — interpreting Hall combinations

The vertical wire series supplies AND gates, with each gate connected to specific configurations of ha, hb, hc, or their inverses. The AND gate produces a logic 1 when the Hall inputs are correct, while the other AND gates produce a 0. Put simply, each AND gate identifies one of the six permissible commutation states. The diagram shows six AND→convert paths, each aligning with an active Hall state in typical six-step commutation. (The diagram seems to show AND gates for every significant combination, and the truth table below verifies which combinations are utilized.)

iv. Transform blocks — correlate logic with numeric EMF impact

The output from each AND gate feeds into a block marked convert. The convert block changes the identified Hall combination into numerical values for the three-phase EMFs. Instead of generating binary 0/1 logic, the convert blocks yield outputs such as +1, −1, or 0, indicating the phase polarity at a specific step:

+1 indicates that the phase is the positive (source) phase, linked to +Vdc,

A value of −1 indicates the negative (sink) phase, linked to either 0 or −Vdc based on the topology, and

0 indicates the phase is undriven and floating.

In this design, each convert block usually creates a three-part output, but here, each convert output is directed to the adders of the correct phase channel, allowing the three-phase outcomes to be combined.

v. Adders and creating emf_a / emf_b / emf_c

To the right, small adder blocks merge the converted contributions. Every adder combines inputs from varying AND/convert routes, which make that phase either positive, negative, or zero across different Hall states. Since just one AND gate operates at any given moment, the adders pick the number aligned with the active commutation state. The last three outputs from these adders are the signals emf_a, emf_b, and emf_c, representing the instantaneous EMF sign for phases A, B, and C numerically.

vi. Truth table — correlating Hall codes with EMF outputs

The module's mapping is clearly displayed in the truth table on the lower right. Each row shows a Hall combination (ha, hb, hc) alongside the respective emf_a, emf_b, emf_c values. Sample entries (format: ha hb hc → emf_a emf_b emf_c):

0 0 0 → 0 0 0 : invalid or no magnet state — all outputs are zero.

0 0 1 → 0 −1 +1 : Phase A is zero, phase B is negative, and phase C is positive.

0 1 0 → −1 +1 0 : Phase A is negative, Phase B is positive, and Phase C is neutral.

0 1 1 → −1 0 +1 : A minus, B neutral, C plus.

1 0 0 → +1 0 −1 : A is positive, B is neutral, C is negative.

1 0 1 → +1 −1 0 : A is positive, B is negative, C is zero.

1 1 0 → 0 +1 −1 : Type A: 0, Type B: +, Type C: -.

1 1 1 → 0 0 0 : sensor state is invalid or overlapping — all zero.

Rows entirely filled with zeros or ones represent invalid states or transitions, often indicating no-drive or fault/idle conditions.

A truth table confirms that in any valid state, one phase is +1, another is −1, and the third is 0, defining the signature of 120° commutation where two phases are active and the third is inactive.

vii. Why are just two phases active and one left floating?

In a BLDC motor, six-step commutation powers two phases simultaneously, with one high and one low, while the third phase is left unpowered. This generates a unipolar torque-inducing current path between two phases, resulting in stepped quasi-trapezoidal phase currents/back-EMF aligned with the rotor magnet design. Control strategies may either measure or disregard the back-EMF during the floating phase, with the latter simplifying commutation and cutting switching losses.

viii. Electrical degrees, timing, and sequence

Hall sensors are usually positioned 120 mechanical degrees (or 60 electrical degrees, based on pole pairs) apart to ensure the digital code sequence progresses through the six commutation steps evenly. When the rotor rotates, the Hall code alters, activating the relevant AND→convert path and advancing the emf outputs to the subsequent combination. This creates a cyclical six-step sequence each electrical cycle and generates a rotating magnetic field in the stator that pulls the rotor with it.

ix. Managing improper or intermediate conditions

In the truth table, both 000 and 111 result in zeros for all EMFs, often seen in sensor faults or at mechanical limits when two sensors switch states almost simultaneously. Setting them to all-zero ensures safety by preventing phase shorting or wrong torque generation. In practice, debounce, brief delays, or interpolation are commonly used during Hall transitions to prevent glitches from misaligned sensor edges.

x. Implementation details (Simulink/FPGA/logic)

This diagram represents a typical Simulink/HDL setup: Hall signals feed into logic units; AND gates identify minterms; conversion units generate numeric values; and adders output numeric phase results. In hardware, this logic can be executed using combinational gates or a compact lookup table (LUT) to convert the 3-bit Hall code into a 3-value vector (emf_a, emf_b, emf_c). Using a LUT, or truth-table lookup, is simpler and reduces errors compared to numerous discrete gates, which is a typical method for microcontrollers or FPGAs to perform commutation decoding.

xi. How the EMF signals are used downstream

The drive controller uses the numeric emf outputs to: (a) determine which inverter transistors to activate (whether the drive phase should be high or low), (b) create PWM gating with respect to the DC link, and (c) optionally perform back-EMF sensing or zero-cross detection during sensorless cycle portions. They can be combined with current references to produce real modulation signals for the inverter.

xii. Potential improvements and protective measures

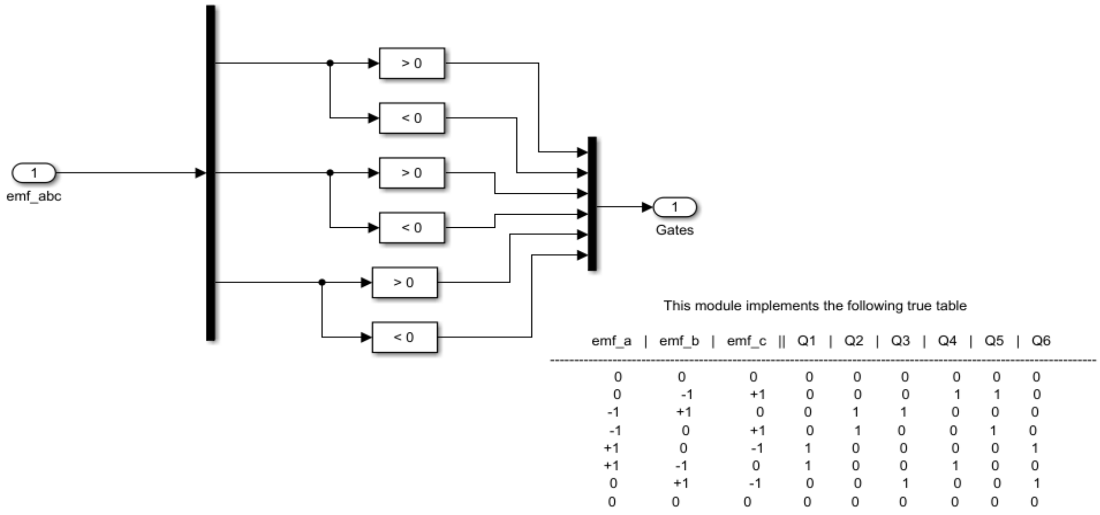
Actual designs incorporate noise filtering for Hall inputs, transition-edge detection to avoid mid-switch commutation, diagnostic detection to report impossible or stuck Hall patterns, dead-time management in power switches, and protection logic to avert shoot-through during transistor pair changes. Using soft-start techniques minimizes mechanical jerk at startup.

xiii. Example walkthrough of one Hall state

Consider ha=1, hb=0, hc=1 as a typical valid state. The AND gate configured in that manner outputs a 1, with its conversion block yielding values (emf_a=+1, emf_b=−1, emf_c=0) that pass through the adders to the outputs. The inverter supplies phase A, draws phase B, and phase C remains inactive. With each 60° rotation of the rotor, Hall signals alter, triggering a new AND path and advancing the output to the subsequent +1/−1/0 configuration, thus perpetuating the commutation cycle.

B. Operating of gate switches

This module transforms the three numerical phase polarity signals emf_a, emf_b, and emf_c from the Hall-decoder stage (each valued at +1, −1, or 0) into six distinct gate enable signals S1...S6 that activate the inverter switches. In essence, it converts the phase EMF pattern (indicating which phase is positive, negative, or floating) into ON/OFF signals for the six power transistors in the three-phase bridge. The system uses six-step (120°) commutation, where in each step one phase is positive, another is negative, and the third remains floating, while the module sends out the pertinent transistor gate signals.



This module implements the following true table

emf_a	emf_b	emf_c	Q1	Q2	Q3	Q4	Q5	Q6
0	0	0	0	0	0	0	0	0
0	-1	+1	0	0	0	1	1	0
-1	+1	0	0	1	1	0	0	0
-1	0	+1	0	1	0	0	1	0
+1	0	-1	1	0	0	0	0	1
+1	-1	0	1	0	0	1	0	0
0	+1	-1	0	0	1	0	0	1
0	0	0	0	0	0	0	0	0

Figure 8: Gate operation

i. Inputs: emf_abc vector

The emf_abc block on the left is an input vector with three elements: emf_a, emf_b, and emf_c. All elements are numeric and represent phase states: +1 indicates the phase connects to the positive DC bus (source), −1 indicates connection to the negative/return (sink), and 0 represents a floating phase (not driven). These figures, derived from either the Hall-decoder or a lookup table, indicate the desired polarity for each phase during the current commutation interval.

ii. Comparators (>0 and<0)

The module includes two basic comparators for each phase: one checks if it's greater than 0, the other checks if it's less than 0. In other words:

emf_phase > 0 holds true solely when the phase needs to be set to high (+1). The condition emf_phase < 0 is met precisely when the phase is intended to be reduced to −1. Comparators change numeric polarity values into boolean states. When the phase value is zero, both comparators yield false; a value of +1 makes only the >0 comparator true, while −1 makes only the <0 comparator true.

iii. Arrangement into gate vector

The six comparators' boolean outputs (two for each phase) are combined into a vector called Gates. The standard sequence, as indicated by the truth table, is:[A_high, A_low, B_high, B_low, C_high, C_low] — where A_high indicates emf_a > 0, and A_low signifies emf_a < 0, etc. Every boolean within this vector corresponds directly to the activation of an inverter transistor gate: upper and lower transistors of phase A, then for phases B and C.

iv. Understanding the insight provided by a truth table

The accompanying truth table displays every possible combination of (emf_a, emf_b, emf_c) and their respective binary outputs S1 to S6. It lists the six valid commutation states and includes the neutral or invalid state of all zeros. For instance, as demonstrated by a typical mapping:

When emf_a is 0, emf_b is -1, and emf_c is +1, it indicates A is floating, B is sinking, and C is sourcing. The comparators will assign B_low and C_high to 1, while all other gates are set to 0. This generates the gate vector with the transistor linking phase B to negative and the one linking phase C to positive both activated.

emf_a = +1, emf_b = -1, emf_c = 0 implies A sources, B sinks, C floats, resulting in A_high=1, B_low=1, others 0.

Every truth-table row guarantees that two gate signals are active (one is high and the other is low) for the two conducting phases, with the remaining four gate signals staying inactive. The table indicates that emf_abc values of 000 and 111 result in all gates being zero (safe/idle).

v. How gates correspond to real transistors

A typical three-phase half-bridge inverter consists of six switches: S1 and S4 for phase A, S2 and S5 for phase B, and S3 and S6 for phase C, though the naming can differ. The *_high comparators activate the upper (high-side) switches, while the *_low comparators activate the lower (low-side) switches. In hardware, incorporate dead-time and interlock logic to ensure that both devices of the same phase are never activated together, preventing shoot-through. The module generates the primary selection, while a downstream gate driver layer needs to handle PWM, dead-time insertion, and protection.

vi. Managing invalid or zero combinations

When emf_phase is 0 in each phase (0 0 0), no comparators activate, resulting in S1 to S6 being zero. The same applies to 1 1 1 (invalid overlap); these secure defaults avert incorrect switching amid faults or sensor errors. In practice, you incorporate diagnostic indicators and apply edge-debounce or brief blanking intervals during Hall transitions, as two sensors might shift nearly at the same time, causing temporary invalid codes.

vii. Sample sequence illustrating a single electrical cycle)

Begin by commuting with (emf_a, emf_b, emf_c) set to (+1, -1, 0). The comparators set A_high and B_low to 1, with others at 0; the gate vector activates the upper A and lower B, allowing current to flow from +Vdc through A to the motor, then to B, and back to DC− via lower B. Approximately 60 electrical degrees later, the Hall decoder transitions to the subsequent state, for instance, (0, +1, -1). With B_high=1 and C_low=1 now true, the gates adjust, redirecting the current to B→C. This cycle of six states creates the two-phase conduction pattern that rotates the rotor.

viii. Implementation notes and alternatives

Even though the diagram employs comparators and a gate vector, in practice, a compact lookup table (LUT) indexed by the 3-bit Hall code typically outputs the 6-bit gate vector directly. This is quicker and more straightforward in microcontrollers/FPGAs. The diagram's comparators suggest emf_* values are numeric; if they're 3-state encoded values (+1/0/−1), you can reduce it to simple equality checks or bit masks.

ix. Safety, PWM and modulation

This logic solely determines the transistors that can be activated, without executing PWM modulation. Typically, the chosen upper or lower device is modulated using a PWM waveform, like carrier comparison, to regulate current amplitude. Final gate signals need to include dead-time to prevent shoot-through, fault masking for overcurrent and undervoltage, and synchronization to ensure seamless commutation without overlap or glitches during Hall transitions.

x. Pros and cons of this method

Benefits: the method is predictable, simple to check, demands minimal processing, and matches standard six-step BLDC commutation.

Constraints: relies on clean, debounced Hall inputs and assumes immediate switching; lacks built-in capabilities for advanced modulation (FOC) or sensorless back-EMF detection. In noisier environments or sensorless setups, advanced observers and enhanced smoothing techniques are necessary.

C. Functioning of control system

This explains the process of transforming a control reference (speed/torque) into an inverter's switching command via the "SMC" (Sliding Mode Controller). It details the conversion of this command into PWM/gate signals and the measurement/display of DC-bus voltage. Essentially, the sequence is: reference → controller → PWM → gate logic → inverter switches, with Vdc being measured and fed back for monitoring.

i. Torque map block reference (1-D T(u) with N = 2000, V1 = 310)*

A 1-D lookup block marked T(u) is located near the bottom center. This is a torque or current reference generator designed as a one-dimensional table: It transforms an input reference, such as a speed reference N* (set at 2000 rpm), into a torque or control setpoint. The numeric block N* 2000 inputs to the lookup, yielding the required torque/current value. V1 310 probably signifies a standard DC link voltage of 310 V utilized by the controller or for reference scaling. This section defines the objectives for the controller, such as the ideal torque or current in relation to speed.

ii. SMC block (Sliding Mode Controller) — functions and inputs

The SMC, identified by the central red block, represents the sliding-mode controller. Its function is to quickly adjust the plant (whether motor current/torque or Vdc, based on the model's configuration) so the measured output matches the reference. The SMC generally calculates the error (difference between reference and measured values) and determines a control action to drive the system state toward a specified sliding surface. The diagram shows the SMC input receiving a signal from a nearby summing junction, with its output serving as a control signal for the PWM generator. Sliding mode control is selected for its ability to handle parameter changes and reject disturbances.

iii. Summing block feeding SMC — reference less feedback

To the immediate left of the SMC, there is a small circular summing node (±). This subtractor generates the control error by deducting a feedback measurement (the returning measured quantity from the plant) from a reference (either from the T(u) torque/speed lookup or another source). The SMC receives the error. For DC-link regulation, the summing block calculates Vdc_ref minus Vdc_measured, whereas for torque/current tracking, it computes Torque_ref minus Torque_meas. The diagram illustrates Vdc routed close by, indicating that the SMC might either control Vdc or utilize it as an input scaling factor. Regardless, the subtractor delivers the necessary error signal for the SMC.

iv. Vdc assessment and Vdc_SMC module

To the right, a DC-link node from the inverter/DC bus provides input to both a block marked Vdc_SMC and the display. Data is captured from the DC bus and delivered to both the SMC for feedback or scaling, and a display/monitoring unit. The Vdc_SMC block might process the raw Vdc signal, such as using a low-pass filter to eliminate switching ripple, before the controller uses it.

v. SMC output → PWM generator (modulating carrier)

The SMC produces a continuous control value, typically a duty cycle ranging from −1 to +1 or 0 to 1, which is then sent to the PWM Generator block. The PWM Generator generates switching waveforms, labeled c and d, by comparing the controller command with a carrier waveform, such as a triangle or sawtooth. The generator produces PWM pulses in the time domain, with pulse widths corresponding to the SMC output. When SMC outputs a positive duty, the PWM maintains the high-side active for that portion of the carrier cycle, while a negative output might activate the complementary device. The PWM outputs serve as the main signals to be transformed into gate signals.

vi. Logic gates: NOT gate, square wave and X gates, generating complements

To the left of PWM, there's a small square, presumably the carrier comparator output, along with a NOT block. These logic blocks generate complementary and inverse signals needed for half-bridge gate pairs. The diagram additionally illustrates x blocks adjacent to [S1] and [S2] resembling multiplexers or logic gates that merge PWM output with gating permission signals, such as channel enabling/disabling. Standard gate logic performs three functions:

(1) decide the active phase/pair using decoder/emf signals,

(2) produce a complementary PWM for the pair, where the upper transistor receives the PWM signal and the lower one receives its inverse, and

(3) add delay to avoid shoot-through.

The NOT produces a complementary waveform from PWM; The x blocks integrate the selection, determining which switch pair is enabled, with the PWM to generate the final gate line for S1 and S2. Although only two gate outputs might be displayed, this pattern is consistent across all three phases.

vii. Output from the last gate: [S1], [S2], and others implied

On the extreme left, outputs labeled [S1] and [S2] are displayed. These signals control specific inverter switches, such as S1 for phase A's high side and S2 for its low side. A complete setup would feature six gate outputs (S1 to S6). The diagram demonstrates how PWM signals merge with gating permissions to create the outputs sent to gate driver hardware. The x blocks probably affect the PWM through a selection signal, ensuring only specific phases are modulated.

viii. Green and magenta blocks for display monitoring

A green square and a magenta display block are situated next to the Vdc measurement on the right. These display units present Vdc readings and potentially the controller output or torque calculations on the model interface. They offer live visualization of crucial signals like Vdc and SMC output. this allows the operator to monitor DC bus voltage, duty cycle, or other variables during simulations or tests.

ix. Timing and modulation specifics (dead-time, carrier, complementary outputs)

A PWM generator requires a triangle wave carrier and must include dead-time between complementary pulses to prevent half-bridge shoot-through. The NOT block generates the opposite PWM for low-side switches; x/multiplexer blocks control the complementary PWM using phase enable. Dead-time must be added between switching off a transistor and activating its complement. Although the diagram omits a dead-time block, it is typically incorporated within the PWM Generator or gate-combination blocks in real systems.

x. Illustration of signal transmission (single operation step)

Example: T(u) from the lookup provides a torque request.; The subtractor computes the error using a measured value, such as Vdc or current; the SMC calculates a duty directive (e.g., 0.6); A PWM generator contrasts 0.6 with a carrier, producing a PWM wave with a 60\% duty cycle; The NOT gate outputs the opposite PWM signal; By integrating the multiplexer x with phase select, PWM can control S1, while its inverted form governs S2; The inverter receives the final gate signals, adjusts the motor's voltage, and alters Vdc/torque; Vdc is tracked and returned to the SMC. The screen indicates Vdc and other internal metrics.

xi. Rationale for control and design selection (why use SMC here?)

Sliding Mode Control is frequently selected in power electronics due to its robustness and quick disturbance rejection, managing parameter uncertainty (changes in motor inductance/resistance) and external disturbances (load torque variations) effectively. This model employs SMC to precisely track the selected variable, whether it's Vdc or torque/current. SMC outperforms a PI controller in terms of speed and robustness, though it demands attention to prevent high-frequency chattering, usually mitigated by employing a boundary layer or PWM for actuation.

xii. Practical implementation and improvements

In practice, these abstract blocks translate to code on an MCU or logic in an FPGA: The lookup table provides torque/current references, the controller uses discrete sampling, the PWM generator is timer-based, and the NOT/multiplex logic employs output gating with safety interlocks. Planned improvements: add dead-time insertion, overcurrent/undervoltage safeguards, soft-start, filter feedback from Hall sensors or encoders, and diagnostic flags for erroneous states. To prevent chattering and minimize switching stress with SMC, incorporate a boundary layer or a high-frequency filter.

D. how to derive reference voltage from reference speed

essentially, there are three varieties;

i. by mathematical calculation

ii. by mat lab simulation

iii. by physical measurements

i. by mathematical calculation

Converting reference speed to reference voltage assumes a direct linear relationship between motor speed and its corresponding voltage. The motor in your situation has a 3000 rpm rated speed and a 310-volt rated voltage. Thus, at peak motor speed, the system anticipates a 310 V voltage, whereas at 0 rpm, it aligns with a reference of 0 V. Due to the linear relationship, a speed from 0 to 3000 rpm corresponds to a voltage from 0 to 310 volts in a proportional manner. The reference speed is divided by the maximum speed to calculate the requested fraction of the full speed range. The reference voltage is calculated by multiplying that fraction with the rated voltage. This leads to the formula where n_{ref} represents the target speed. For example, 1500 rpm is half of the maximum speed, so the voltage half of the maximum (155 volts). This way you are sure that the voltage command and the desired motor speed is either half of the range throughout the whole range. The table contains the calculated values for speeds from 0 to 3000 rpm increased by 100 rpm. In each row is the speed with the respective reference voltage, which is obtained by the same linear formula. The lookup tables is often used by controllers, PLCs and particularly by the embedded devices to increase the speed and simplify the implementation by the precalculated values. A controller can use the table to quickly convert any speed request into the right voltage, eliminating the need for continuous formula calculations during operation.

ii. by mat lab simulation

In a MATLAB/Simulink simulation, the speed-to-voltage conversion starts by creating a model with reference speed as the input. The input is processed using either a formula block (like a Gain or Divide block) or a Lookup Table block with speed-to-voltage mapping. During the simulation, Simulink calculates the reference voltage at every step. To document these voltages, Simulink offers various logging options. A basic method is linking the reference voltage output to a 'To Workspace' block. Once the simulation concludes, this block automatically transfers the voltage signal to the MATLAB workspace as a time-series variable. After the variable is in the workspace, you have the options to display, plot, save, or transform it into a table. A frequently used approach is employing a 'Scope' block in simulations to display the voltage signal live. Once the simulation ends, access the Scope to directly save the recorded data. Simulink enables signal logging via the Data Inspector, capturing chosen signals in a simulation without additional blocks. Just right-click a signal line and choose “Log Selected Signal”. Upon concluding the simulation, the recorded reference voltage is displayed in the Simulation Data Inspector for viewing or exporting to the workspace. When using a lookup table for conversion, the 1-D Lookup Table block in Simulink can accommodate your full 0-3000 rpm table. During the simulation, Simulink autonomously selects the reference voltage based on the input speed. Utilizing a To Workspace block or signal logging enables continuous capture and saving of the output. This simplifies the validation of conversion logic, plot generation, and the comparison of simulated and theoretical voltages. MATLAB/Simulink offers versatile tools for capturing reference voltages during simulations, simplifying analysis and documentation.

What does the "D" represent in a 1-D Lookup Table?

In Simulink, a 1-D Lookup Table block associates an input variable with an output value through a set of predefined data points. Rather than computing the output with a formula at each simulation step, the system directly retrieves it from a table using the input. When the input doesn't precisely align with any table values, Simulink interpolates between the closest points for a smooth, precise output. Lookup tables are especially beneficial in control systems that require fast response, accuracy, or dealing with non-linear dynamics.

In 1-D Lookup Table, "D" signifies "dimension." Here, a dimension is the count of independent inputs that influence the result. A one-dimensional (1-D) lookup table features a single input. In your case, target speed of motor is the input and output is the corresponding reference voltage. We can see in the table there are two vectors, one for explaining input breakpoints(speeds) and one for explaining output values(voltages). In simulink, it reads both vectors and calculates the output voltage for any input speed, even if the input speed is in between the breakpoints. Yes, you can certainly use a 1-D lookup table for converting speed to voltage since your reference voltage depends only on speed. With this table you can compute and store the voltage values for the speed range of 0 to 3000 rpm, in 100 rpm increments. When you simulate, the block will jump instantly to providing the correct voltage when the target speed changes. This simplifies the model, reduces the computational burden and ensures accurate speed to voltage conversion, rather than having to keep recalculating the formula. Conversely, if the output relies on two or more input variables, a 2-D or 3-D lookup table is required. For instance, if the reference voltage was influenced by speed and motor load, a 2-D lookup table would apply. Likewise, with three influencing factors, a 3-D table is needed. The "dimension" indicates the number of independent variables influencing the output.

iii. by physical measurements

To physically ascertain a motor's reference voltage, note that motor voltage and speed are directly proportional during standard operation. In a DC motor, speed is directly linked to the voltage applied when the load is constant, as the back EMF and terminal voltage are related. In AC motors using variable frequency drives (VFDs), voltage must adjust in direct proportion to frequency to maintain constant flux, ensuring a linear speed-to-voltage link. The main concept is that each speed corresponds to a specific voltage needed for the motor to operate, measurable with suitable tools. You will need a variable voltage source (DC power supply for a DC motor or VFD for an AC motor), a voltmeter or multimeter to measure the motor terminal voltage, a tachometer or RPM meter to measure the motor rotational speed. The power source will allow you to incrementally increase the voltage into the motor, the voltmeter will show you what voltage you are currently at at the motor terminals, and the tachometer shows you the motor speed. Link one end of the power cord to the motor and connect the other end to the variable power source or VFD. Be sure the voltmeter is hooked up to the motor terminals to measure the voltage. The voltmeter will allow you to incrementally increase the voltage to the motor. The tachometer will be attached to the motor shaft/rotor to accurately measure the motor speed. Start the motor at 0 volts, the speed should be 0. Increase the voltage slowly and watch the tachometer. The voltage meter will show the voltage at which the motor reaches 1500 rpm. This is the starting voltage for the 1500 rpm speed. You can repeat this process for other speed values to create a speed to voltage reference table. To create a reference table for use in simulations or for creating a reference table for a controller, you could physically increase the speed in 100 rpm increments from 0 to 3000 rpm and measure the voltage increment by increment. You would have to assume that the motor was either lightly loaded or not loaded at all, because the load will affect the speed-voltage correlation. I would think you would need a VFD for AC motors as it takes care of both voltage and frequency to keep them correlated. A Variable Frequency Drive or VFD (also called a Variable Speed Drive) is an electronic device that controls the speed and torque of an AC motor by controlling the supply frequency and voltage to the motor. Essentially it allows an AC motor to speed up or slow down beyond the normal operating speed without changing hardware. Essentially it is just measuring what the voltage is. You will incrementally increase the voltage to the motor while incrementally increasing the speed. The engineers were using this method to tune up their motor controllers and create their own reference tables. They want to be able to measure speed and voltage directly without relying on equations or simulations.

IV. Simulation and Results

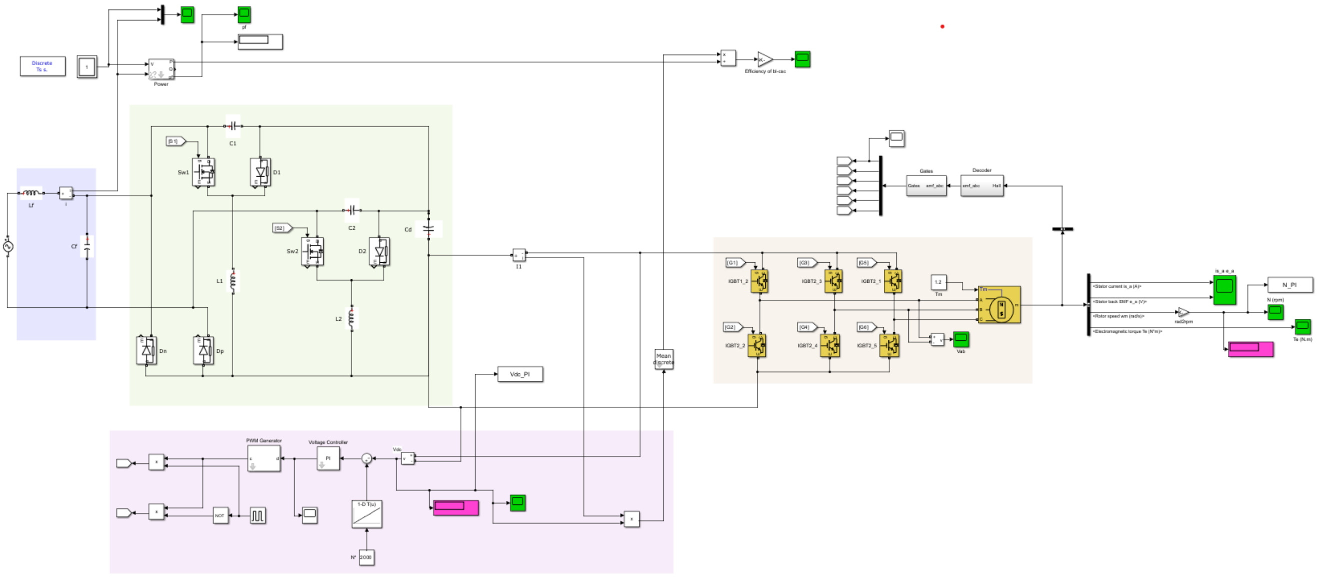


Figure 9: Simulink design of a BL-CSC PFC converter linked to a BLDC motor via a PI controller

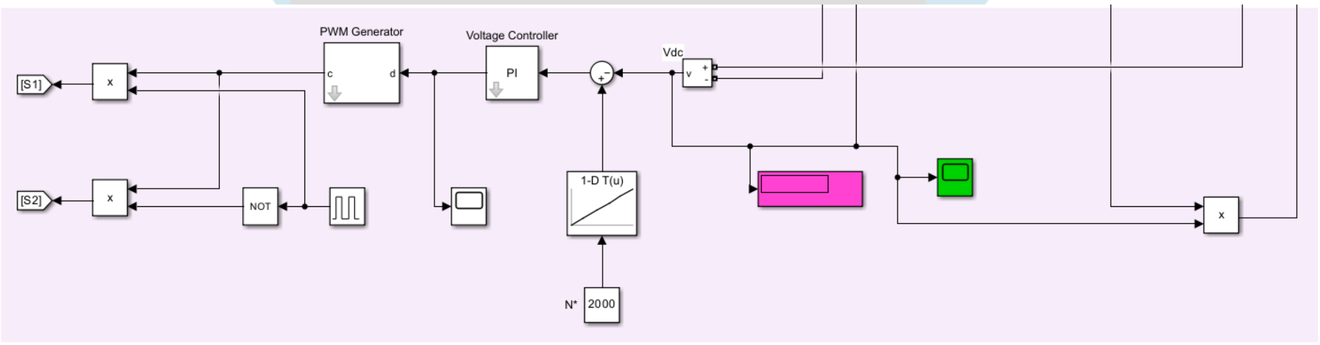


Figure 10: PI voltage regulator for the BL-CSC PFC converter

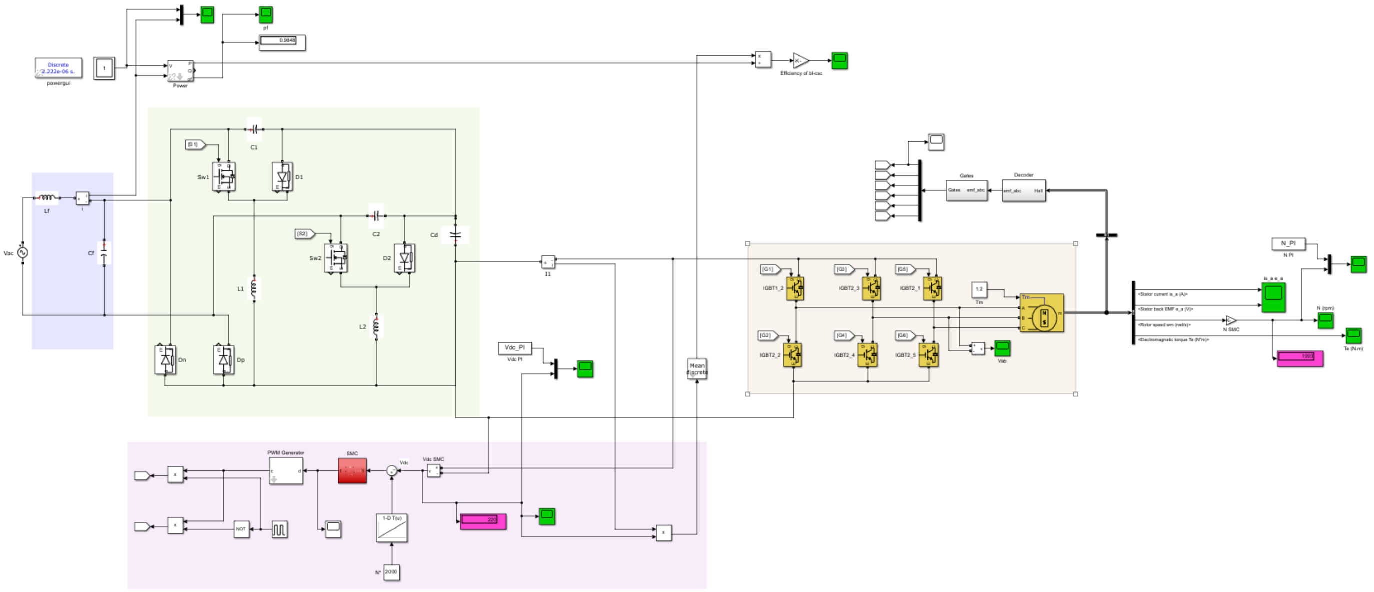


Figure 11: Simulink model of a BL-CSC PFC converter linked to a BLDC motor using an SMC controller

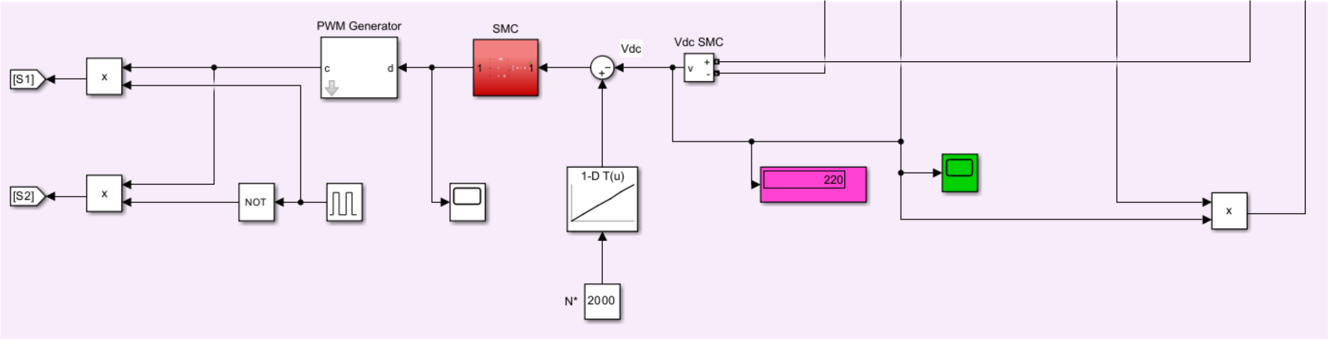


Figure 12: SMC Voltage regulator for the BL-CSC PFC converter

A. Analyzing motor performance

There are two ways to evaluate the performance of an electric motor:

static and transient performance. With these, you can know how the motor performs when it is in a steady state and what happens when there are changes in load, speed, or input voltage. It is essential to understand electric motors both in design, management, and evaluation in industrial, automobile, and other engineering applications. steady state performance is how a motor performs when it reaches steady state. In this state, electrical and mechanical factors, such as current, torque, speed, and power, do not change with time. In other words, it is the state in which a motor operates in a steady state. When a motor reaches a steady speed from initiation and continues to run at that speed, it is in a steady state. The static properties of the motor are often represented with torque-speed, efficiency, and power factor graphs. Engineers often use these graphs to visualize motor performance under different loads.Key aspects analyzed in steady-state are efficiency, torque, speed regulation, and current use. When evaluated, these aspects will show the motor's ability to efficiently convert electrical energy into mechanical energy, how much load the motor can carry, and how effectively it can operate under certain conditions. Dynamic performance, on the other hand, is the response of a motor to changes or disturbances such as a change in load, speed directive, or input voltage. It is the behavior of a motor during a short period of time while it is shifting to a new steady state. For example, when a motor starts from a stationary position, changes its velocity, and encounters a sudden increase in load, it goes through a dynamic phase before reaching a new steady state. Generally, dynamic performance is defined with parameters such as acceleration, deceleration, rise time, settling time, and overshoot. These parameters define how fast and how efficiently a motor achieves its new speed. A motor with better dynamic performance reaches its new speed quickly and steadily without big oscillations or too much delay. This aspect is important in fields such as robotics, electric cars, and industrial automation, where quick and accurate motion control is critical.

B. The Significance and Applications of Dynamic and Steady-State Testing

Testing a motor in both steady state and dynamic operation will ensure a motor that is reliable, safe, and efficient in real-world service. The efficiency, power, current, and torque ratings of a motor are verified with steady state testing. This ensures quality, design accuracy, and compliance with standards. This test ensures that the motor can perform its duties under normal operating conditions without overheating or wasted energy. Dynamic testing measures the performance of a motor when starting, stopping, or changing loads. This information helps engineers design effective control systems, choose appropriate drive electronics, and prevent machine vibrations and stresses. For electric vehicles, dynamic tests determine the motor's rate of acceleration and its ability to switch between speeds smoothly.

C. Why We Use These Tests

Testing both steady-state and dynamic performance ensures that the motor will operate as expected in both steady-state and changing conditions. Every motor intended for industrial use must run efficiently and be able to handle changes in process load. Testing identifies design problems, validates control algorithms, and improves efficiency. Testing both static and dynamic properties allows engineers to more accurately represent the electrical and mechanical behavior of the motor. Modeling these characteristics is necessary for the development of parameters to change the motor behavior instantaneously, such as for PID or vector control. Tests in this category help achieve precision in aerospace, fast response in robotics, and longevity in industrial automation. In short, steady-state performance is an indication of efficiency in a steady condition and dynamic performance is an indication of behavior with changes and disturbances. To understand how a motor will behave while in operation, both characteristics must be understood. Testing to determine both provides information necessary to accurately define a design, develop appropriate controls, and ensure performance. All of this helps motors operate efficiently while changing to meet the various demands of the modern world of electrical and mechanical systems.

D. Based on our simulation

The behavior in steady-state for proposed BLDC motor drive with 220 V AC supply and 310 V DC link voltage which is being handled by both PI and SMC. The next two graphs show the waveform Vs which is the supply voltage to the motor marked Vs it is a neat sine wave since the AC is constant. The Is waveform is the motor stator current. The current waveform is a sinusoidal form similar to the voltage, so the motor is running well without any distortion. The next graph shows the Vdc plot which is the voltage across the DC link that handles the inverter. It is constantly about 310 V. The system is working well and the rectifier or DC link is working fine. The Ia waveform at the bottom is the phase-A current of BLDC motor. Unlike the sinusoidal form of the current in the stator, the current in the phases of the BLDC motor with step-like switching pattern is controlled in trapezoidal. The stepped current is verified to be the motor commutation is working fine and the current is in rated range. The motor speed is 2927.53 rpm and 2963.71 rpm. It means that motor is working near its rated speed with these steady electrical conditions. maybe PI and SMC results.

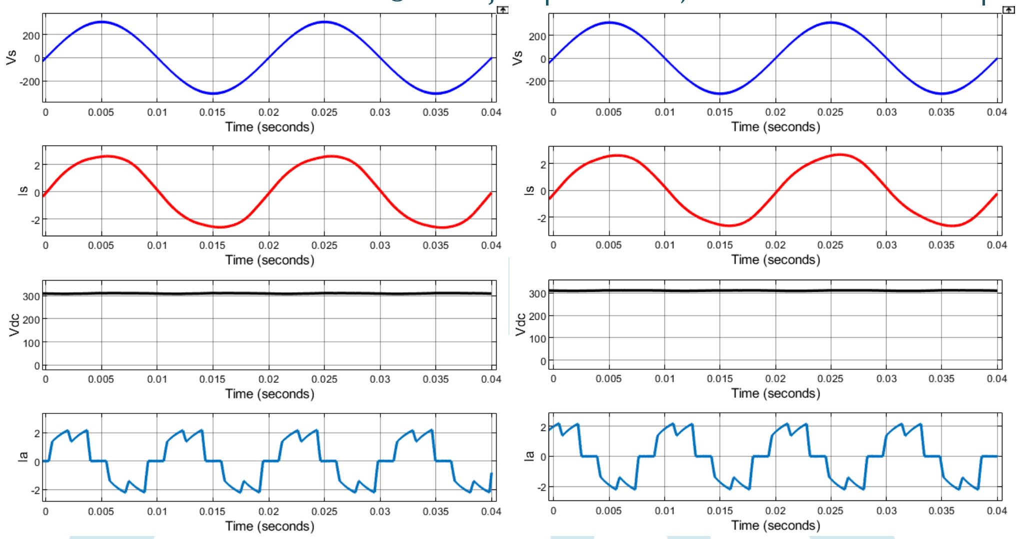


Figure 13: The drive’s performance under rated conditions with a 220V supply and a 310V DC link, utilizing a PI & SMC controller

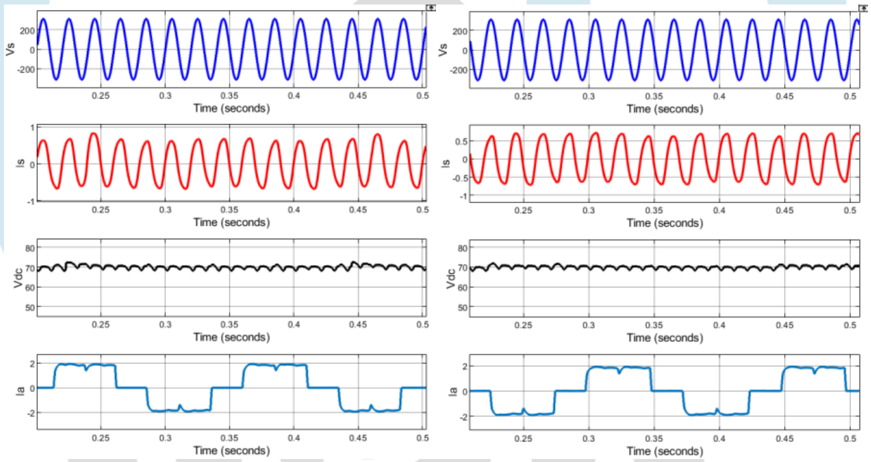


Figure 14: The drive’s operation under the specified condition, utilizing a 220V supply voltage and a 70V DC link voltage with a PI & SMC controller, was tested.

As shown in the Fig.14 steady-state performance with constant 220 V AC supply and reduced DC voltage across the link (70 V). The Vs waveform is the supply voltage to the motor marked Vs it is a neat sine wave since the AC is constant. The Is waveform is the motor stator current, it is a neat sine wave since the AC is constant also. The Vdc waveform is the voltage across the DC link that handle the inverter. It is oscillate around 70 V and it has a ripple because of the reduced energy in the DC link capacitor across both sides. The reduced voltage cause the inverter not able to supply enough excitation to the BLDC motor. It is seen in the Ia current waveform at the bottom. The phase-A current has more irregularity and more sharp change. It means the current and so the torque is decreased and the control is not sharp enough. So the speed of the motor will be decreased to around 200 rpm. It is normal when the DC link voltage is too low to reach the speed of the motor’s rated speed. This entry shows the same drive with much lower DC power. Therefore, it is obvious that the DC link voltage should be in an appropriate range in order to get the desired speed and torque from the BLDC.

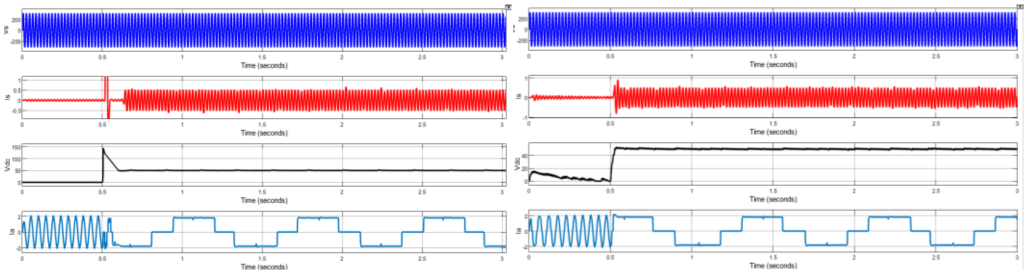


Figure 15: The proposed drive’s startup performance with a 50V DC supply on the BLDC motor using a PI & SMC controller at rated load.

As shown in the Fig.15 the requested drive's dynamic behaviour under rated load with a PI controller, the motor starts from 50 V DC link voltage. The Vs waveform plots show the sinusoidal behaviour of the continuous AC supply during the 3 seconds of simulation. The Is waveform plot shows a large transient at start-up. This is to be expected as a PI controller makes a powerful correction when the motor starts from speed zero. The Vdc waveform plot shows a large change: the DC voltage spikes initially before settling, meaning that the continuous supply, rectifier and load all interact as the motor starts to take current. During start-up, the Ia phase current waveform plot shows intense oscillations. These oscillate a lot with large spikes. These spikes show that the PI controller takes some time to decrease the current and stabilise the system, especially with a large voltage DC voltage at the DC link. With time, the oscillations die down and the motor current stabilises to show that the PI controller can always stabilise the system and motor operation eventually, although the response is slow and shows overshoot. This is a known disadvantage of using a PI controller in any system that is classed as non-linear or changing rapidly, such as BLDC drives.

the dynamic operation with a Sliding Mode Controller (SMC) starting from a DC voltage of 50 V. The Vs waveform plot shows the sinusoidal behaviour of the continuous AC supply. The Is waveform plot shows quicker stabilisation than the PI controller, but still shows a transient. The Vdc plot shows the DC link voltage increasing rapidly and stabilising sooner than the previous PI scenario. This shows the SMC can provide much more control and stronger behaviour even at low DC voltage. The Ia waveform plot shows much smoother behaviour than the PI start. There are oscillations, but they are less and stabilise quicker. One of the advantages of using an SMC is that it can guarantee a smooth behaviour by forcing the system to follow a certain path. It makes the system follow the desired path by forcing the system to behave as it should, meaning that the nonlinear behaviour is overcome by the SMC. The behaviour of the current shows that the SMC improves the dynamic response and reduces both overshoot and settling time.

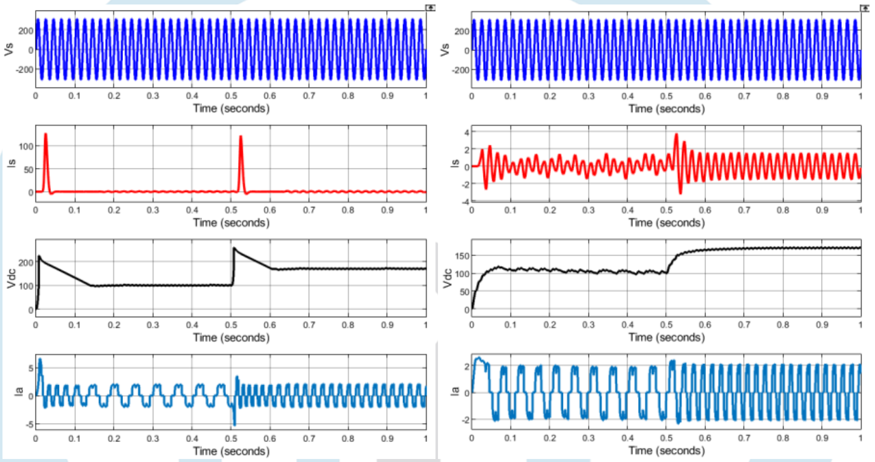


Figure 16: The proposed drive's dynamic performance was evaluated for speed control on a BLDC motor, with DC link voltage varying from 100V to 170V, employing a PI & SMC controller.

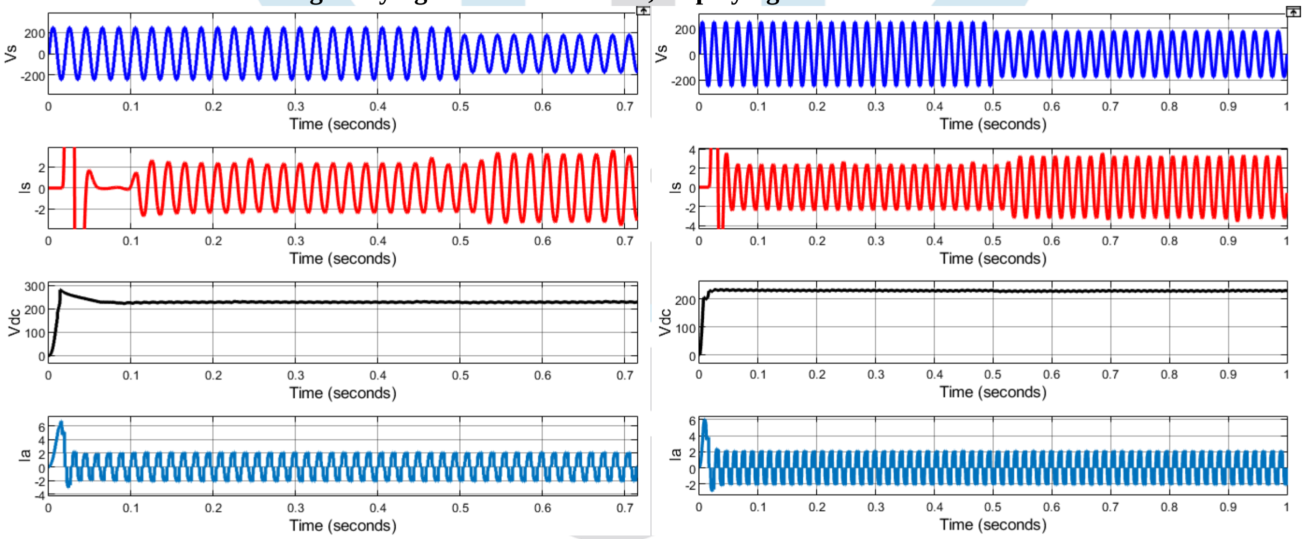


Figure 17: The proposed drive's dynamic response at full load on the BLDC motor when the supply voltage suddenly drops from 250V to 180V using a PI & SMC controller.

On the PI results on the left, we can still see sinusoidal Vs waveform. At some certain times, there are distinct large spikes in the Is waveform appeared, especially when the DC link voltage is changing, it shows that the PI controller can not properly and smoothly change the current of motor when DC link voltage changing quickly. Vdc shows that the DC link is increasing from about 100V to 170V midway of the simulation. Since the PI is very slow to respond to the sudden change of abrupt input shift, we can see that the current commands are very random as shown in the Ia below. The current phase varies a lot, which means the PI control gives rise to a ripple and instability that should be eliminated in the future. The motor reaches the target speed more slowly in the case of PI control as shown in 500 rpm and 1350 rpm.

the voltage change happens similarly, yet more precise. The spikes on the Is are less distinct and flatter. In the case of SMC, as shown in the above figure, the DC link voltage increases a little more smoothly, and the Ia is much more stable. The SMC controller is able to handle a sudden shift in voltage from 100V to 170V with relatively little disruption in the current through the motor. As a result, the motor reaches its speed target (600 rpm and 1350 rpm) more consistently with less transient instability.

From the above analysis, we can see that compared with the PI control, SMC is far better in dynamic, robustness, and current regulation, especially when the voltage or load changes suddenly.

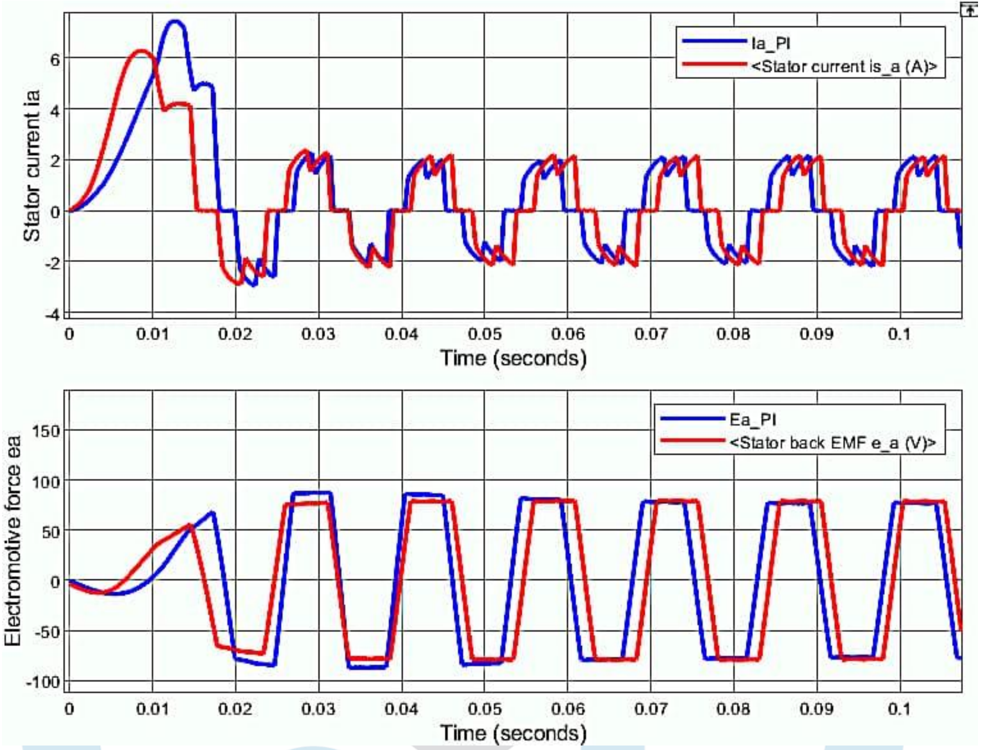


Figure 18: Stator current and back EMF in BLDC motor

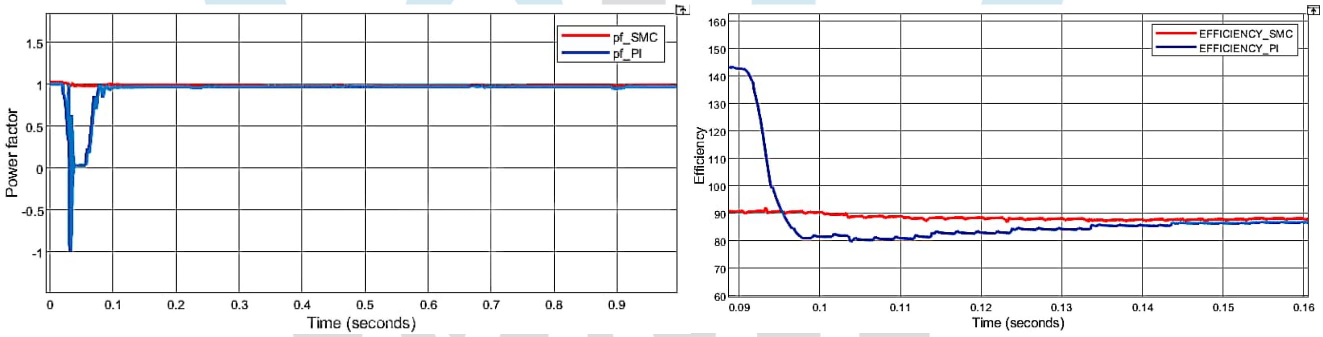


Figure 19: Source’s power factor and Efficiency of the BL-CSC PFC converter

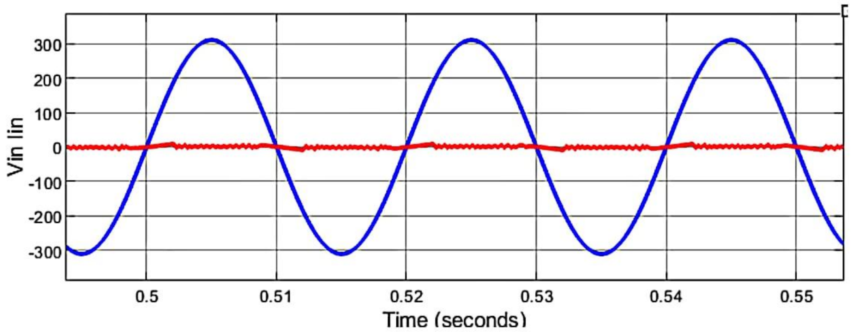


Figure 20: Input voltage and current

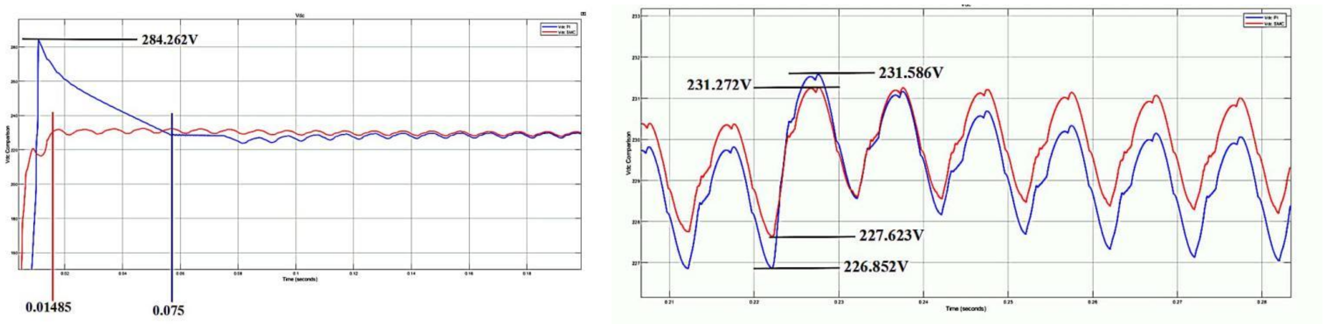


Figure 21:DC link voltage comparison

$$\text{Ripple}(\%) = \frac{\text{Difference}}{\text{average}} * 100$$

$$\text{PI regulator of voltage} = \frac{\frac{231.586-226.852}{\frac{231.586+226.852}{2}}}{} * 100 = \frac{4.732}{229.219} * 100 = 2.06\%$$

$$\text{SMC regulator of voltage} = \frac{\frac{231.272-227.623}{\frac{231.272+227.623}{2}}}{} * 100 = \frac{3.649}{229.4475} * 100 = 1.59\%$$

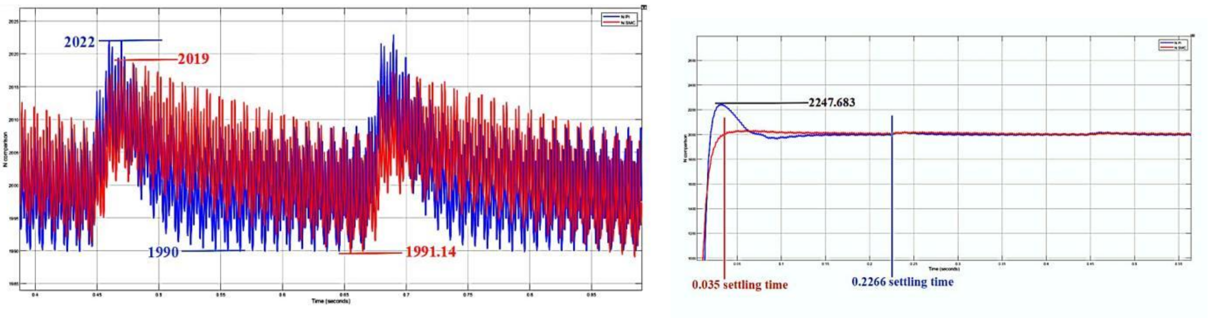


Figure 22: BLDC motor speed comparison

$$\text{PI regulator of speed} = \frac{\frac{2022-1990}{\frac{2022+1990}{2}}}{} * 100 = \frac{32}{2007.5} * 100 = 1.7\%$$

$$\text{SMC regulator of speed} = \frac{\frac{2019-1991.14}{\frac{2019+1991.14}{2}}}{} * 100 = \frac{27.86}{2005.07} * 100 = 1.3\%$$

Name of the parameter	*PI*	*SMC*
DC peak	284.262 V	No peak
DC ripple	2.06%	1.59%
DC settling time	0.075 s	0.0148 s
Speed peak	2247.683 rpm	No peak
Speed ripple	1.7%	1.3%
Speed settling time	0.226 s	0.035 s

Table 1: Comparison table

V. Conclusion

The system includes a BLDC motor drive with a PFC-based BL-CSC converter to improve AC mains power quality. Using a bridgeless setup in the CSC converter reduces conduction losses in the PFC converter. Furthermore, one voltage sensor is used to regulate both the BLDC motor's speed and the PFC at the AC mains. Minimizing switching losses in an VSI is achieved by fundamental frequency switching via electronic commutation in the BLDC motor. The speed of the BLDC motor is controlled by varying the dc link voltage of the VSI's. The suggested drive shows improved power quality at the AC mains over a wide range of control and supply voltages. The suggested drive shows good results and is recommended for low-power uses. Improving the SMC voltage regulator greatly improves the performance of the BLDC motor without changing the system rating. The DC voltage and the BLDC motor speed are halved, and the settle time is greatly reduced.

References

[1] AT Sankara Subramanian, P Sabarish, and A Nazar Ali. A power factor cor rection based canonical switching cell converter for vsi fed bldc motor by using voltage follower technique. In 2017 IEEE International Conference on Elec trical, Instrumentation and Communication Engineering (ICEICE), pages 1–8. IEEE, 2017.

[2] Bhim Singh and Vashist Bist. A bl-csc converter-fed bldc motor drive with power factor correction. IEEE Transactions on Industrial Electronics, 62(1):172–183, 2014.

[3] Jitendra Gupta, Radha Kushwaha, and Bhim Singh. Improved power quality charger based on bridgeless canonical switching cell converter for a light electric vehicle. IEEE Transactions on Industry Applications, 59(4):4610–4619, 2023.

[4] ASS Subramanian, P Sabarish, and R Jai Ganesh. An improved voltage follower canonical switching cell converter with pfc for vsi fed bldc motor. Journal of Science & Technology (JST), 3(3):01–11, 2018.

[5] PJ Shaija and Asha Elizabeth Daniel. Robust sliding mode control strategy applied to ifoc induction motor drive. In 2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT), pages 1–6. IEEE, 2021.

[6] L Pattathurani, Rajat Kumar Dwibedi, and P Sivachidambaranathan. A bl-csc converter fed bldc motor drive with power factor correction. IOSR J Electr Electron Eng, 10(6):89–97, 2015.