CHAPTER 4

FPGA IMPLEMENTATION OF SIX SWITCH AND FOUR SWITCH INVERTER FOR INDIRECT POSITION DETECTION OF PERMANENT MAGNET BRUSHLESS DC MOTOR USING VIRTUAL INSTRUMENTATION

4.1 INTRODUCTION

An Electric motor is defined as a transducer that converts electrical energy into mechanical energy. Electrical motors are an integral part of industrial plants, around five billion motors built worldwide every year. Residential and commercial applications mostly use conventional motor drive technologies. Typically, machines found in these appliances are single-phase induction motors or brushed DC machines which are characterized by low efficiency and high maintenance, respectively. Single-phase induction motors are less efficient because of the ohmic loss in the rotor and due to the phase angle displacement between the stator current and back-EMF. In the case of DC machines, they require more maintenance due to the presence of brushes. Replacing these inefficient motors with more efficient Brushless DC (BLDC) motors will result in substantial energy savings.

The trapezoidal BLDC motors fed with ideal rectangular current generate smooth instantaneous torque. However, ideal rectangular current shapes cannot be realized in practice due to the phase inductance and finite inverter voltage. Because of this, non-commutated phase current undulates due to phase commutation and this pulsating current generates undesirable
torque ripple. The commutation torque ripple varies with speed and may reach 50% of the average torque. To apply the BLDC motors to home appliances, it is required to minimize the commutation torque ripple, because the torque ripple generates noise, vibrations and causes errors in sensorless motor drives. Two different methods for reducing the commutation torque ripple when a single current sensor is used in the DC link of the inverter to regulate the current flowing through two motor phases in series. These methods are inappropriate to apply to various trapezoidal BLDC motors, because they are open loop in nature.

The introduction of independent current sensors in the motor phases provides a powerful approach for reducing commutation torque ripple, because the current regulation of each phase is possible during commutation periods. Carlson has demonstrated that the commutation torque ripple can be minimized during low speed by the introduction of direct current sensing with hysteresis current controller.

Hysteresis current control and Pulse Width Modulation (PWM) control coupled with continuous control theory have produced the most widely used BLDC motor control techniques. Sinusoidal pulse width modulation is used to control the inverter output voltage and maintain a good performance of the drive in the entire range of operation between zero and 78% of the value that would be reached by square operation as mentioned by Rahman et al 1985.

The commutation torque ripple can be reduced by the introduction of feed-forward term in current controller to compensate for the average voltage variation of the non-commutated phase due to commutation in bipolar pulse width modulation (PWM) inverter-fed BLDC motor drives stated by Berendsen et al (1993).
For low cost application unipolar PWM is not suitable, because the voltage between the neutral point of the inverter and the neutral point of the machine varies with the on–off status of the switching device of the inverter stated by Dae-Kyong Kim et al (2006).

With the growing complexity of motor and motion control applications, it becomes apparent that a Field Programmable Gate Array (FPGA) offers significant advantage over the shelf Application Specific Standard Product (ASSP) solutions in the areas of performance, flexibility and inventory control. With an FPGA, calculations that would normally consume large amounts of CPU time when implemented in software may be hardware accelerated. Using hardware acceleration allows for more functionality within the system software. Custom motor drive interfaces such as PWM can be developed easily, quickly and at low cost. Additionally, because of full configurability, the same FPGA can be used in various product ranges, reducing the need to maintain inventory for multiple devices.

PWM is easily developed design using the Spartan™-3 device based FPGA Motor Control, that utilizes the MicroBlaze™ 32-bit CPU soft core, floating point unit, and associated memory subsystems, SPI communications interface, Hall sensor output and IP specifically designed to control brushless DC (BLDC) stated by Craig Hackney et al (2005).


Since the concept of multilevel PWM inverter was introduced, various modulation strategies have been developed and studied in detail such as multilevel PWM and multilevel selective harmonic elimination. A standard three-phase voltage source inverter utilizes three legs (six-switch three-phase voltage source inverter or SSTPI), with a pair of complementary power switches per phase. A reduced switch voltage source inverter (four switch three-phase voltage source inverter or FSTPI) uses only two legs, with four switches. The Several papers report on FSTPI structure. The advantages of FSTPI is to reduce the cost of inverter, lesser switching losses, lower EMI and less complexity of the control algorithms and interface circuits to generate PWM logic signals.

An asymmetric PWM scheme for a four-switch three-phase BLDC motor drive is to make six commutations and produce four floating phases to detect back electromotive force. The position information of the rotor can be acquired based on the crossing points of the voltage of controllable phases stated by Lin and Hung et al (2008). Virtual Hall sensor signals are made by detecting the zero crossing points of the stator terminal voltages, and there is no need to build a 30° phase shift, which is prevalent in most of the sensorless algorithms given by Niasar and Moghbeli et al (2007).

The pulsating current can be suppressed by using the voltage disturbance rejection method and unipolar PWM. The direct phase current control method can suppress the pulsating current only in the region where DC bus voltage is larger than four times that of phase back electromotive force. In the voltage disturbance rejection method, the input for compensation must be applied to the inverter only during the commutation. For this reason, the voltage disturbance rejection method uses a commutation interval that can
be measured or estimated from phase current. However, current sensors increase the cost of a motor driver.

![Symmetrical PWM Waveform](image)

**Figure 4.1 Symmetrical PWM Waveform**

This work proposes a novel strategy for reducing commutation torque ripple in a position sensorless BLDC motor drive. Since the proposed method directly measures commutation interval from motor terminal voltage waveforms, it does not require a current sensor and current control loop. Therefore, the proposed method is suitable for a low cost BLDC motor driver. In addition, the proposed method synchronizes the commutation points at the starting point of a PWM carrier signal. The experimental result shows that the proposed method considerably reduces not only current ripples but also the vibrations. Speed control of BLDC motors is often achieved by using high frequency PWM control within the overall ON time for device as specified by
the rotor position. When high frequency switching is employed, the terminal voltages need to be filtered before zero crossing detection can be attempted. However this can be achieved without a filter and consequent delays by sampling the terminal voltage waveform during the PWM ON time at a suitable instant.

A symmetrical PWM pattern is proposed in this work as shown in Figure 4.1, which shows the advantage of lower THD without increasing the switching losses. Thus this work demonstrates that a more efficient and faster solution is the use of FPGA, it investigates how to generate a variable PWM waveform based on Xilinx FPGA and the proposed design is tested by functional / timing simulation and experiments.

A voltage source inverter is commonly used to supply a three phase permanent magnet brushless DC motor. By varying the voltage across the motor, the speed of the motor can be controlled. When using PWM outputs to control the six switches of the three phase bridge, variation of the motor voltage can be obtained by varying the duty cycle of the PWM signal.

4.2 PROPOSED VI BASED INDIRECT POSITION DETECTION USING SIX SWITCH INVERTER AND FPGA

Feedback control loops are implemented to increase dynamic performance or precision of scientific and industrial equipment. The basic principle of such loops is to taken into account actual measurements in order to compute appropriate actuations that adjust the operational conditions to meet the given requirements. Motion control and process control are two major application areas of this paradigm. Due to this broad application field and its interdisciplinary nature, Automatic control is a fundamental subject usually taught in many engineering discipline, such as electrical, mechanical and chemical engineering. Acquisition of measurements and modification of
actuations are the usual tasks carried out by LabVIEW and DAQ boards. With the rapid progress in microelectronics, FPGA is more and more flexible, programmable and lower in cost, is therefore more and more widely used by recent researchers. In order to achieve a low cost BLDC control, the Xilinx 3S100E FPGA is used to replace the microprocessor or DSP to implement the sensorless six switch inverter scheme.

The implementation of this paradigm based on personal computer and standard operating systems constitute a new trend in automation. It allows the user to avoid such aged, specialized or expensive solutions as analog PID loops, Programmable Logic Controllers (PLC) or dedicated hardware based on Digital Signal Processors (DSP). To stress the advantage of the above mentioned open paradigm, the researchers are provided with an integrated framework versatile enough to be quickly adapted to their different needs and backgrounds. This solution reduces the additional knowledge required to proceed with the implementation and helps them to focus on essential concepts.

To be attractive the framework needs to be highly interactive and offer a well-designed graphical user interface. The scalability and the ease in designing a user interface provided with LabVIEW, as well as its multi-platform implementation, make this package well suited to develop didactic tools. Moreover, the possibility to compile standalone virtual instruments, which can be freely distributed, is a big advantage. Moreover, the proposed paradigm for real time control implementation is not only limited to education. In research and industry, its ease of use also represents an interesting opportunity to meet the growing needs of scientists for fast prototyping. This paradigm enables teachers to implement real time control solutions in a really efficient manner, both from a time and resources perspective.
The total work is a novelty in following ways.

1. The control scheme is simple in architecture

2. Position is detected using line voltage difference instead of back-EMF using LABVIEW.

3. The PWM control has been realized using a single FPGA.

4. Such a wide frequency control with very high frequency-switching is only possible by utilizing the state of art VLSI digital circuit design approach.

5. The whole system is implemented in a single chip and the waveforms are easily monitored using LabVIEW which makes the circuit very compact.

6. Systems of FPGA chip are more reliable because they do not need any control software.

7. Faster design and verification time, design change without penalty.

A three phase bridge was inverter fabricated using n – channel MOSFET and is operated in 120 degree mode to provide square wave current excitation to the stator windings. MOSFET’s are used to control the speed of the motor by varying the supply voltage to the motor. It can be switched on to a very high speed with the help of PWM waves. The PWM waves are generated by the FPGA.

The PWM time period and duty cycle is controlled by the LabVIEW software. The MOSFET has advantages of faster switching, lesser drive power requirement and absence of secondary breakdown phenomenon.
4.2.1 Realization of ZCD and Waveforms in LabVIEW

Acquisition of measurements and modification of actuations are the usual tasks carried out by LabVIEW and DAQ boards. Firing pulses to the inverter are provided by detecting the zero crossings of the back-EMF waveform using LabVIEW. Normally DAQ boards are used to measure the current and voltage signals. But in our proposed method serial communication is implemented to reduce the total cost. In telecommunications, RS-232 is a standard for serial binary data interconnection between a Data Terminal Equipment (DTE) and a Data Circuit-Terminating Equipment (DCE). It is commonly used in computer serial ports.

The line voltage difference is measured using simple resister divider circuit and it is filtered to eliminate the ripples present in the wave form. The filtered waveform is compared in a comparator and the comparator output is converted into digital and given to LabVIEW through RS232. In LabVIEW Stacked Sequence structure consists of one or more sub diagrams, or frames, that execute sequentially. The frames can be added or deleted and to create sequence locals to pass data between frames. Using the Stacked Sequence structure used to ensure a sub diagram executes before or after another sub diagram. The Block Diagram of VI based Indirect position detection using six switch inverter is shown in Figure 4.2.
Figure 4.2   Block Diagram of VI based indirect position detection using Six switch Inverter and FPGA

The Real time monitoring is carried out using LabVIEW in virtual Instrumentation by adopting certain procedure in step by step manner. The steps are as follows

1. The data is read by using VISA READ function in Block Diagram of VI Programme.
2. Initially there is a syntax checking which checks whether the data is received or not.

3. The Voltage, Current and Speed Data’s are received serially as specified in byte count (is the number of bytes to be read)

4. Data are stored in one frame of stacked sequence.

5. Using the data the Speed waveform and voltage waveform is displayed in Front Panel of the VI Program.

6. SET speed is given to the Block diagram of VI is converted into string which is compared with actual speed.

7. The error information and the position information’s are sent to FPGA through serial communication

VISA Read function placed in stacked sequence structure used to Read the specified number of bytes from the device or interface specified by VISA resource name and returns the data in read buffer.
Case 0: SYNTAX CHECKING

VISA read reads the specified number of bytes from the device or interface specified by VISA resource name and returns the data in read buffer. This is clearly shown in Figure 4.3. This consist of three main blocks.

1) VISA input
2) VISA serial
3) VISA read

Figure 4.3 Case 0: Syntax Checking
VISA input get the specified mode of communication from the user, either serial or parallel. This program is for serial communication, and the user selection mode COM (serial communication) enables the input block.

VISA serial initializes the serial port specified by VISA resource name to the specified settings. This polymorphic initializes the serial port using VISA class. Its default value 9600 bps with 8 bit data rate and zero parity.

Case 1: SERIAL DATA INPUT

Stacked sequence 1 checks the data if its start bit is * and if so its true, it enters a case structure which is shown in Figure 4.4.

Figure 4.4 Case 1: Serial Data Input
The value wired to the selector terminal of this case structure determines which case to execute and can be Boolean, string, integer or enumerated type.

Case 2: STORING VOLTAGE AND SPEED VALUES

Stacked sequence 2 contains an VISA input block to get the set speed which is given to the string subset block. This returns the sub string of the input string beginning at offset which is shown in Figure 4.5.

Figure 4.5 Case 2: Storing Voltage and speed values
Case 3:  SENDING DATA THROUGH SERIAL PORT TO FPGA

Stacked sequence 3 concatenates the sub string using concatenation string block and converts the numeric characters in string starting at offset and to a decimal integer and returns it in number. This case 3 structure shown in Figure 4.6.

Figure 4.6 Sending data through serial port to FPGA

From Figures  4.7 and 4.8, it is very clear that the wave forms of each phase is clearly displayed and the actual speed is also measured. The
information from VISA READ function is directly given to FPGA for producing proper speed control.

Figure 4.7  Voltage waveforms for set speed 4500 rpm
Figure 4.8 Voltage waveforms for set speed 7000rpm

4.2.2 Generation of Synchronous PWM

From the sensed terminal voltages with respect to negative DC bus ($V_a$, $V_b$, $V_c$), line voltages and subsequently their differences ($V_{ca}$, $V_{ab}$, $V_{bc}$, $V_{bca}$) are determined. It can be seen that the waveform contains voltage
spikes. These appear at the commutation instants and are the result of conduction of free-wheeling diodes at the phase commutation instants. They are to be filtered out, otherwise they cause unwanted zero crossings in the line voltage difference waveform and the spikes would cause spurious zero crossing detection. In order to eliminate this, the line voltage differences are fed to simple filter circuits.

The Filter circuits are activated just prior to a device switching so that they hold the previous value of the voltage waveform during the free-wheeling diode conduction. The voltage spikes are there by filtered out. Such a filtered line voltage difference waveform ($V_{abce}$) is also shown in Figure 4.9.

![Figure 4.9 Measured Line voltage difference waveform](image)

The filtered voltages are fed to zero crossing detector circuit to detect the zero crossing instants of the line voltage difference waveform.
4.2.3  Sensorless Starting Mode

At standstill the back-EMF induced is zero. Therefore the rotor is aligned to a predetermined position by triggering two devices for a predetermined time. After alignment, switching signals 120 degrees interval away from the pre-positioning switching signals are applied to the devices in order to cause a rotor movement. This rotor movement is sufficient enough for the proposed algorithm to detect the zero crossing instants of the back-EMF. Then the subsequent switching signals are applied at the zero crossing instants without any delay which develops positive torque to rotate the motor. Thus the motor is started in sensorless mode so as to develop sufficient back-EMF for the sensorless running to takeover. The speed waveform during sensorless starting and running mode are shown in Figure 4.10.

![Graph showing speed waveform with labels for starting and running modes](image)

**Figure 4.10** Speed wave form for 7000 rpm in Front panel of VI Programme
During the pre-positioning period (200 ms) the rotor moves from an unknown position to a predefined position. This movement is reflected in the speed waveform shown in Figure 4.9. The rotor oscillates before settling to a predefined position. Upon starting the machine using the proposed algorithm, the speed starts increasing without any oscillations. Unlike in many applications, this approach for start-up does not require prior tuning or ramping up the switching frequency in a predetermined manner. While this approach to switch the inverter is sufficient even to achieve continuous operation of the motor, estimation of the correct triggering instants based on the zero crossing detection improves the operation.

4.2.4 Sensorless Running Mode

Once the motor is started in the sensorless starting mode then the correct commutation instants are to be estimated from the ZCP in order to switch over to running mode. When zero crossing information becomes available, the switching is done and the motor enters the running mode according to the sequence given in Table 4.1. Subsequent to this changeover, 60° interval detection and the delay of 30° is performed on a continuous basis to maintain operation in the running mode. Speed control of BLDC motors is often achieved by using high frequency PWM control within the overall ON time for device as specified by the rotor position. When high frequency switching is employed, the terminal voltages need to be filtered before zero crossing detection can be attempted. This can be achieved without a filter and consequent delays by sampling the terminal voltage waveform during the PWM ON time at a suitable instant.
Table 4.1 Sequence for 60 Degrees Counter Operation

<table>
<thead>
<tr>
<th>Device switching Signal</th>
<th>Every 60° period</th>
<th>Start at</th>
<th>Stop at</th>
</tr>
</thead>
<tbody>
<tr>
<td>T_A^H</td>
<td>+ve going ZCP of V_{abbc}</td>
<td>-ve going ZCP of V_{bcca}</td>
<td></td>
</tr>
<tr>
<td>T_B^H</td>
<td>+ve going ZCP of V_{bcca}</td>
<td>-ve going ZCP of V_{abbc}</td>
<td></td>
</tr>
<tr>
<td>T_A^L</td>
<td>-ve going ZCP of V_{abbc}</td>
<td>+ve going ZCP of V_{caab}</td>
<td></td>
</tr>
<tr>
<td>T_C^H</td>
<td>+ve going ZCP of V_{caab}</td>
<td>-ve going ZCP of V_{bcca}</td>
<td></td>
</tr>
<tr>
<td>T_C^L</td>
<td>-ve going ZCP of V_{bcca}</td>
<td>+ve going ZCP of V_{abbc}</td>
<td></td>
</tr>
<tr>
<td>T_C^L</td>
<td>-ve going ZCP of V_{caab}</td>
<td>+ve going ZCP of V_{caab}</td>
<td></td>
</tr>
</tbody>
</table>

4.2.5 VHDL Coding for Duty ratio calculation and PWM generation

The duty ratio is calculated from the actual speed and position details from VISA data output function from LabVIEW programming.

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity phase_3 is
port(clk : in std_logic;
      pulse : in std_logic;
      i1,i2,i3 : in std_logic;
      rpm : out std_logic_vector(15 downto 0):="0000000000000000";
      s1,s2,s3,s4,s5,s6 : out std_logic);
end phase_3;
architecture df of phase_3 is

  signal count1 : integer:=0;
  signal count : std_logic_vector(15 downto 0):="0000000000000000";
  signal a : std_logic:="0";
  signal dcount : std_logic_vector(7 downto 0):="00000000";
  signal e : std_logic:="0";

begin

  process(clk,pulse,i1,i2,i3,a,count,count1,dcount,e)

    variable temp1,temp2 :std_logic;

  begin

    if clk'event and clk='1' then

      --s4<="0";
      --s5<="0";
      --s6<="0";

      count1 <= count1 + 1;

      dcount<=dcount+1';

      if i1='0' and i3='1' and i2='0' and E='0' then

        s1 <= '1';
        s2 <= '1';

        s3 <= pulse;

        -- s3 <= '0';

        ---s4 <= pulse;

        s4 <= '0';

        s5 <= '1';

        s6 <= '1';

    end if;

  end process;

end df;
elsif i1='1' and i2='0' and i3='0' and E='0' then
    s1 <= pulse;
    -- s1 <= '0';
    s2 <= '1';
    s3 <= '1';
    s4 <= '1';
    -- s5 <= pulse;
    s5 <= '0';
    s6 <= '1';

    -- elsif i1='1' then

elsif i1='1' and i2='1' and i3='0' and E='0' then

    s1 <= pulse;
    --- s1 <= '0';
    s2 <= '1';
    s3 <= '1';
    s4 <= '1';
    s5 <= '1';
    s6 <= '0';
    --- s6 <= pulse;---

    --- elsif i2='1' and i3='1' then
elsif i1='0' and i2='1' and i3='0' and E='0' then
    s1 <= '1';
s2 <= pulse;  --- s2 <= '0';
    s3 <= '1';
    s4 <= '1';
    s5 <= '1';
    s6 <= '0';
    --- s6 <= pulse;

elsif i1='0' and i2='1' and i3='1' and E='0' then

    s1 <= '1';
s2 <= pulse;
    --- s2 <= '0';
    s3 <= '1';
    --- s4 <= pulse;

    s4 <= '0';
    s5 <= '1';
    s6 <= '1';

--elsif i1='1' and i2='1' then

elsif i1='0' and i2='0' and i3='1' and E='0' then

    s1 <= '1';
s2 <= '1';
s3 <= pulse;
    s3 <= '0';
s4 <= '0';
    --- s4 <= pulse;
    s5 <= '1';
s6 <= '1';

elsif i1='1' and i2='0' and i3='1' and E='0' then

    s1 <= '1';
    s2 <= '1';
    s3 <= pulse;
        --- s3 <= '0';
        s4 <= '1';
        --- s5 <= pulse;
        s5 <= '0';
    s6 <= '1';
    count <= count + "10";
end if;

    if dcount <= "00111100" then
        E <='1';
    else
        E<='0';
    end if;
end if;
   if i1='1' and a='0' then
      --- count <= count + '1';
      a <= '1';
   elsif i1='0' and a='1' then
      a <= '0';
   end if;
   if count1=25000 then
      rpm <= count;
   end if;
   if count1=25100 then
      count <= "0000000000000000";
      count1 <=0;
   end if;
end process;
end if;

Switching waveforms of each 60 operation is produced in FPGA are given as triggering pulses to the voltage source inverter. At starting the back-EMF is zero, so the current drawn by the motor has to be regulated. With high frequency switching, duty ratio of the PWM can be adjusted so as to regulate the phase current drawn. In this work, a 10 KHz PWM signal is applied only to the top MOSFET devices of the inverter. Phases B and C are excited for pre-positioning the rotor (gating signals TB+ and TC- are switched ON). After pre-positioning the phases C and A are excited for the initial rotor movement.
Thus the motor reliably starts in the sensorless mode. The motor operates in the starting mode till 1.023s and then changes over to running mode.

Simple novel technique to detect back-EMF zero crossings for a BLDC motor using the line voltage difference is proposed. It is shown that the line voltage difference provides an amplified version of the back-EMF in the zero crossing region. A simple and reliable method of starting and running the machine in sensorless mode is then proposed in this work making use of this novel zero crossing detection algorithm. A novel starting algorithm is proposed that switches the devices at the zero crossings detected using the proposed algorithm.

4.3 PROPOSED VI BASED INDIRECT POSITION DETECTION USING FOUR SWITCH INVERTER AND FPGA

Conventionally, BLDC motors are excited by a six-switch inverter as explained in chapter 4.1. However, cost-effective design is becoming one of the most important concerns for the modern motor control research. Some researchers developed new power inverters with reduced losses and costs. Among these developments, the three-phase voltage source inverters with only four switches, as shown in Figure 4.11, is an attractive solution. In comparison with the usual three-phase voltage-source inverter with six switches, the main features of this inverter are twofold: the first is the reduction of switches and freewheeling diode count; the second is the reduction of conduction losses.

Lee et al and Kuk Lee et al developed BLDC motor drives with trapezoidal BEMF using the four-switch three-phase (FSTP) inverter. The authors used four-space-vector scheme and the six commutation modes based on current control. They used position sensors to achieve commutation control of BLDC motors. However, position sensors make the total system
more expensive, larger in volume, and less reliable. On the other hand, sensorless control for six-switch three-phase BLDC motors has had many successful applications.

![Diagram of a 3-phase BLDC motor with an FSTP inverter](image)

**Figure 4.11 Conventional FSTP Inverter fed PMBLDC motor**

Almost all sensorless control schemes for six-switch three-phase BLDC motors detect the zero-crossing point of voltage waveforms from unexcited windings to estimate the rotor position, but it is impossible to achieve sensorless control schemes for four-switch three-phase BLDC motors by using the conventional four-space-vector strategy, because there is no floating winding. In contrast, if six commutation modes are used in the four-switch inverter, then there are four floating phases during the operating period.
is obtained, and the position information can be detected from the floating line.

In this proposed work a position sensorless BLDC motor drive with a new algorithm for sensorless operation and sensorless control without signal injection is proposed. The rotor position is sensed using line voltage difference of the motor. In order to achieve a low cost BLDC control, the Xilinx 3S100E FPGA is used to replace the microprocessor or DSP to implement the sensorless FSTP inverter scheme and to make the effective control Virtual Instrumentation is used. Using Virtual Instrumentation the waveforms can be seen, and the speed control is very easy. The block diagram for FPGA Based FSTP inverter fed Three Phase Brushless DC Motor using Virtual instrumentation is shown in Figure 4.12.

![Diagram of FPGA-based sensorless FSTP BLDC motor configuration using LabVIEW](image-url)
4.3.1 Novel PWM Scheme for FSTP BLDC Motor Drives

For BLDC motors with a trapezoidal back-EMF, rectangular stator currents are required to produce a constant electric torque. The proposed voltage PWM scheme for FSTP inverter requires six commutation modes which are (X,0), (1,0), (1,X), (X,1), (0,1) and (0,X), as shown in Figure 4.13(a) to 4.13(f) respectively.

![Diagram of six commutating modes of voltage PWM scheme for FSTP inverter- Mode 1 (X,0)](image)

Figure 4.13(a) Six commutating modes of voltage PWM scheme for FSTP inverter- Mode 1 (X,0)

The symbols in parenthesis denote the switch ON/OFF states of $T_A^H$, $T_A^L$, $T_B^H$, and $T_B^L$ (phases A and B). “X” denotes the OFF state for both high side and low-side switching devices in the same leg, “1” denotes the ON state for the high-side switching device, and “0” denotes the ON state for the low-side switching device.
Figure 4.13(b) Six commutating modes of voltage PWM scheme for FSTP inverter - Mode II (1,0)

Figure 4.13(c) Six commutating modes of voltage PWM scheme for FSTP inverter - Mode III (1,X)
Figure 4.13(d) Six commutating modes of voltage PWM scheme for FSTP inverter - Mode IV (X,1)

Figure 4.13(e) Six commutating modes of voltage PWM scheme for FSTP inverter – Mode V (0,1)
Figure 4.13(f) Six commutating modes of voltage PWM scheme for FSTP inverter - Mode VI (0,X)

In Mode II, the FSTP BLDC motor drive uses the conventional voltage PWM scheme as shown in Figure 4.14. In Mode II, this conventional voltage PWM scheme provides a discharging loop between the capacitor and the low-side switch as shown in Figure 4.15(a) and (b) which causes non-rectangular stator current waveforms which are harmful for constant torque.
Figure 4.14 Conventional voltage PWM scheme for FSTP BLDC motor
Figure 4.15 (a) Operation stages of FSTP inverter using conventional PWM scheme in Mode II - stage (1,0)

Figure 4.15(b) Operation stages of FSTP inverter using conventional PWM scheme in Mode II - stage (X,0)
To produce the rectangular current waveform, additional stage is added in Mode II. The corresponding three stages are (1,0), (X,0), and (X,X) as shown in Figure 4.16 (a)-(c) respectively. This proposed FSTP BLDC motor drive uses the Novel voltage PWM scheme as shown in Figure 4.17.

Figure 4.16(a) Operation stages of FSTP using novel PWM scheme in Mode II - stage (1,0)

Figure 4.16(b) Operation stages of FSTP using novel PWM scheme in Mode II - stage (X,0)
Figure 4.16(c) Operation stages of FSTP using novel PWM scheme in Mode II - stage (X,X)

Similar situations are applied to Mode V. The new stage (X, X) of this novel PWM scheme in Modes II and V is introduced to turn off all power devices to prevent the capacitor discharging from the low-side switch. Furthermore, the supply voltages in Modes II and V are doubled and hence other four Modes I, III, IV and VI the PWM duty cycle double. This novel voltage PWM scheme is called as the asymmetric PWM scheme for FSTP BLDC motor drives. The Switching sequences are shown in Table 4.2.
Table 4.2  Switching Sequence of the Novel Asymmetric Voltage PWM Scheme

<table>
<thead>
<tr>
<th>Modes</th>
<th>Active Phase</th>
<th>Floating phase</th>
<th>Switching devices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode I</td>
<td>Phase B and C</td>
<td>Phase A</td>
<td>$T_B^L$</td>
</tr>
<tr>
<td>Mode II</td>
<td>Phase A and B</td>
<td>Phase C</td>
<td>$T_A^H$ and $T_B^L$</td>
</tr>
<tr>
<td>Mode III</td>
<td>Phase A and C</td>
<td>Phase B</td>
<td>$T_A^H$</td>
</tr>
<tr>
<td>Mode IV</td>
<td>Phase B and C</td>
<td>Phase A</td>
<td>$T_B^H$</td>
</tr>
<tr>
<td>Mode V</td>
<td>Phase A and B</td>
<td>Phase C</td>
<td>$T_A^L$ and $T_B^H$</td>
</tr>
<tr>
<td>Mode VI</td>
<td>Phase Aand C</td>
<td>Phase B</td>
<td>$T_A^L$</td>
</tr>
</tbody>
</table>

Figure 4.17  Novel voltage PWM scheme for FSTP BLDC motor
4.3.2 Starting Technique

The first step to start the sensorless drive is to get the initial rotor position. Since only in Modes II and V the BLDC motor is supplied by whole DC bus, the inverter could supply enough power to drive the rotor to an expected position. Therefore for starting, simply excite the motor in Mode II or Mode V to force rotor to rotate in the specified direction.

4.3.3 Novel Sensorless Control Scheme

If rotor position sensors (Hall sensors) are installed into BLDC motors, then from the voltage waveforms of phases A and B, it is observed that two waveform crossings matched the two Hall signals (101 and 010) at the same time, respectively, as shown in Figure 4.18. Therefore, we propose to use the two crossings for rotor position estimation for sensorless commutation purposes. The time difference between the two crossings can be estimated.

![Figure 4.18 Voltage waveforms for BLDC motor using FSTP inverter and the relationship between waveform crossings](image-url)
The time difference between the two crossings is equal to the crossing counter (N) multiplied by the period of the timing counter, which is \(10^{-6}\) s.

In summary, we have time difference between commutation is given from equation (4.1) to (4.4)

$$T = (N-0) \times 10^{-6}$$  \hspace{1cm} (4.1)

$$T_c^1 = T/3$$  \hspace{1cm} (4.2)

$$T_c^2 = 2T/3$$  \hspace{1cm} (4.3)

$$\omega = \pi / 2T$$  \hspace{1cm} (4.4)

### 4.3.4 Implementation of FSTP

The line voltage difference is filtered by Low Pass Filters (LPF) and fed to the LabVIEW programme to find the actual speed and to measure the voltage. The set speed is given in LabVIEW which compares the actual and set speed and then the error is also given to FPGA. Figure 4.19 shows the switching pulses produced by FPGA.

![Switching pulses produced by FPGA](image)
By writing the VHDL programming the proposed PWM is produced for four switches alone. The proposed algorithm is implemented with the Xilinx.

Figure 4.20 Entire circuit diagram of FSTP BLDC Motor
4.3.5 Hardware implementation

Figure 4.20 shows the entire circuit diagram and the hardware setup of Xilinx Spartan Kit with interface is shown in Figure 4.21 and whole experimental system configuration is shown in Figure 4.22.

![Hardware setup of Xilinx Spartan Kit](image)

Figure 4.21 Hardware setup of Xilinx Spartan Kit

The split capacitor bank must be large enough that it can be treated as a voltage source. The voltage across capacitors and the voltage ripple are applied across the switch. It is reasonable to allow 5% voltage ripple in the voltages across $C_1$ and $C_2$. The relationship between the ripple voltage of the capacitors’ and the current in the capacitors is
\[ i_c = C \frac{\Delta V_c}{\Delta C} \]

\[ C = \frac{i_c \Delta t}{\Delta V_c} = \frac{\Delta t}{\Delta V_c} \]

Figure 4.22 Experimental system configuration

The detailed schematic diagram of the sensorless control is shown in Figure 4.23. Table 4.3 shows how much logic resource of FPGA is used to implement the whole system, and as shown in the table every item is utilized below 40%. It means one can select a smaller and cheaper FPGA to further reduce the cost, or one can also build up a microcontroller Intellectual Properties (IP) into FPGA to implement more sophisticated control algorithm.
Figure 4.23  Detailed schematic diagram of the sensorless control

The speed, voltage and current waveform of the FPGA-based sensorless control for FSTP BLDC motor drives for various speeds is shown in Figures 4.24 and 4.25. From the figure we can observe that the rotor speed is accelerated to the specified speed (2000 rpm) because the novel sensorless scheme can estimate the correct rotor position. As demonstrated the motor runs stably at both high and low speeds under sensor less control.
### Table 4.3 Device utilizations and Logic Resources of FPGA

<table>
<thead>
<tr>
<th>Logic Utilization</th>
<th>Used</th>
<th>Available</th>
<th>Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Total Number Slice Registers</strong></td>
<td>260</td>
<td>1,920</td>
<td>13%</td>
</tr>
<tr>
<td>Number used as Flip Flops</td>
<td>257</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number used as Latches</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of 4 input LUTs</td>
<td>523</td>
<td>1,920</td>
<td>27%</td>
</tr>
<tr>
<td>Number of occupied Slices</td>
<td>377</td>
<td>960</td>
<td>39%</td>
</tr>
<tr>
<td><strong>Total Number 4 input LUTs</strong></td>
<td>623</td>
<td>1,920</td>
<td>32%</td>
</tr>
<tr>
<td>Number used as logic</td>
<td>523</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number used as a route-thru</td>
<td>99</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number used as Shift registers</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of bonded IOBs</td>
<td>28</td>
<td>108</td>
<td>25%</td>
</tr>
<tr>
<td>Number of GCLKs</td>
<td>2</td>
<td>24</td>
<td>8%</td>
</tr>
<tr>
<td><strong>Total equivalent gate count for design</strong></td>
<td>7,318</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![BLDC MOTOR CONTROL](image)

**Figure 4.24** The speed and voltage waveform of the FPGA-based sensorless control for FSTP BLDC motor drive for the set speed of 2000 rpm
Figure 4.25 The speed and voltage waveform of the FPGA-based sensorless control for FSTP BLDC motor drive for the set speed of 5000 rpm

4.4 CONCLUSION

This research work provides a novel FPGA-based sensorless control scheme for six switch and four-switch three-phase brushless DC motor drive. In this scheme, a novel asymmetric PWM scheme using six commutation modes in the FSTP inverter is proposed. It is shown that the line voltage difference provides an amplified version of the back-EMF in the zero crossing region. Hence the position information is estimated from the crossings of line voltage difference waveforms, and a low cost FPGA is utilized to implement the algorithm.
The stator current waveforms of the FSTP inverter using this novel voltage PWM scheme are rectangular hence the motor will operate smoothly. A simple and reliable method of starting and running the machine in sensorless mode is then proposed in this work making use of this novel zero crossing detection algorithm. A novel starting algorithm is proposed, which switches the devices at the detected zero crossings. Continuous running is achieved by realizing the correct commutation instants 30° delay from the zero crossings. Simulation and experimental results are shown validate the suitability of the proposed method. The experimental results show that the scheme works very well. With the developed control scheme and the lowest cost implementation, the proposed scheme is suitable for commercial applications.

Field Programmable Gate Array (FPGA) has better advantages compared to microprocessor and DSP control, this sensorless technique based on line voltage difference is implemented in FPGA with the help of LabVIEW Programming. To control the speed of the BLDC motor a PWM gate signals are generated for the six switch inverter based on the zero crossing instants using FPGA. The FPGA programming makes it easier in designing the pulse width modulation pattern generator using field programmable array. Logic resource of FPGA is used to implement the whole system, and every item is utilized below 40%. It means one can select a smaller and cheaper FPGA to further reduce the cost, or one can also build up a microcontroller Intellectual Properties (IP) into FPGA to implement more sophisticated control algorithm.

The driving system of a reduced parts BLDC motor drive which can be used in low cost applications. The main benefits are increased system reliability and cost reduction of the overall system. Cost reduction is achieved via reducing number of power switches and related circuits such as power
supplies and drivers and also elimination of position hall effect sensors. Switches are replaced by simple capacitors to reduce the cost by 33% and thereby the switching losses of the switches.

Table 4.4 gives the comparison of performance and device utilization and cost with the work published by Lin et al. (2008).

**Table 4.4 Comparison of proposed work with Lin et al.**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Resources</td>
<td>Four Switch FPGA, Conventional Position detection circuit and low pass filter.</td>
<td>Four switch FPGA, Virtually created position detecting circuit and low pass filter.</td>
</tr>
<tr>
<td>Device Utilization</td>
<td>40%</td>
<td>Less than 40%</td>
</tr>
<tr>
<td>Speed control</td>
<td>Up to 2000 rpm</td>
<td>Up to 5000 rpm</td>
</tr>
<tr>
<td>Settling time</td>
<td>2 sec</td>
<td>0.5 sec</td>
</tr>
</tbody>
</table>

From the table it is clear that the proposed sensorless speed control of PMBLDC motor using FSTP inverter can be suitable for low cost high speed commercial applications.