VARIABLE SPEED DRIVE FOR THREE-PHASE INDUCTION MOTOR IN COOLING APPLICATIONS

By

LEE TENG HONG

Dissertation submitted in partial fulfillment of

the requirements for the

Bachelor of Engineering (Hons)

(Electrical & Electronics Engineering)

Universiti Teknologi PETRONAS

Bandar Seri Iskandar

31750 Tronoh

Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

VARIABLE SPEED DRIVE FOR THREE-PHASE INDUCTION MOTOR IN COOLING APPLICATIONS

by

Lee Teng Hong

A project dissertation submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Approved:

_____

Assoc. Prof. Kondapalli Siva Rama Rao
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

December 2010

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

_____
Lee Teng Hong

ACKNOWLEDGEMENTS

ABSTRACT

Induction motors (IM) are the most widely used type of motor in the industry mainly because they can be directly connected to the AC supply. However the motor can only run at a small speed range close to the rated speed with constant frequency supply. This results in energy wastage as the motors need not run at full speed all the time in most of the applications. A drive is required to vary the motor speed in order to reduce power consumption and to improve process control. This project intends to complete the designing and prototyping of an electronic variable speed drive (VSD) for three-phase IM. The VSD is designed for cooling applications where the IM is coupled to a cooling fan. The drive receives temperature feedback from objects to be cooled and output a corresponding frequency of voltage to IM. The VSD prototype is constructed for applications on a 175 W, 4-pole, squirrel cage three-phase IM. The heart of the control circuit is a low-cost Microchip's PIC16F777 microcontroller where it is programmed using C language to generate variable frequency sinusoidal pulse width modulation (SPWM) switching signals. These switching signals are fed to an IGBT inverter. The design of the control circuit and the generation of the SPWM signals are explained in detail in this report. The experimental results of applying the VSD prototype to an IM show that the motor is able to run at desired speed based on the feedback temperature as well as user setting. Cost savings using the prototype are also being demonstrated.

TABLE OF CONTENTS

vii

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

IM	Induction motor
PWM	Pulse width modulation
VSD	Variable speed drive
AC	Alternating current
DC	Direct current
PIC	Programmable integrated circuit
DSP	Digital signal processing
RMS	Root mean square
SCR	Silicon controlled rectifier

CHAPTER 1
INTRODUCTION

1.1 Background of Study

Electric motors use around 70 % of all electricity in industry [1]. Therefore any chance of reducing the electricity consumption of electric motors is highly significant especially when energy conservation and green technology have become the world's agenda. Most of the motors used are induction motors and one way of reducing induction motors' power consumption is through the introduction of power electronics controller to control the speed of induction motors (IM). The power electronic controller or Variable Speed Drive (VSD) act as a driver for IM, changing the frequency of supply before feeding it to the motor. VSD is very useful for IMs which are used in cooling application because the motor is often not required to run at full speed. VSD is able to optimize motor speed depending on the temperature of objects to be cooled.

1.2 Problem Statement

In most of the cases IMs are consuming more electricity than required. Most induction motors without speed control drive are running close to full speed. This is wasteful because the motor keep running faster than the requirement [1]. Furthermore it is a common practice for engineers to oversize the motor to cater for future expansion which waste unnecessary energy. Oversizing also occurs as motors are available only at certain rating of speed and engineers normally opt for the next bigger size relative to process requirement. Beside energy wastage, motors running at higher speed than required often results in premature wearing off of the motor, increasing maintenance costs and possibly reducing the lifespan of the motors.

1.3 Objectives

The objectives of the project are:

1. To design a VSD that continuously regulates the speed of a 3-phase IM based on temperature feedback. The speed should also be able to be controlled manually by user.
2. To construct a prototype of the VSD and to obtain experimental results from the prototype
3. To prove reduction in power consumption with the implementation of the prototype

1.4 Scope of Study

The main components of the VSD prototype include:

i. An inverter
ii. A frequency control circuit that provides controlled switching signal to the inverter to produce variable frequency AC.

The features of the VSD prototype include:

i. User is able to select between two modes of speed control: the automatic Temperature Control Mode and the manual User Control Mode
ii. Under Temperature Control Mode, speed of 3-phase IM is based on temperature feedback. The temperature is displayed in a LCD. The higher the temperature, the faster the speed
iii. Under User Control Mode, user is able to control the speed manually using a controller. Desired synchronous speed is displayed in a LCD.

CHAPTER 2

LITERATURE REVIEW

2.1 Reviews on Electronic VSD Technology

The induction motor (IM) was invented by Nikola Tesla in 1924 and it was highly desirable as it can be directly connected to single-phase or three-phase supply [2]. The motor speed is directly proportional to the applied frequency and is determined by the formula N = 120 f / P where N is the synchronous speed, f is the frequency of applied power and P is the number of pole on the motor. Therefore a 4-pole IM with applied frequency of 50 Hz runs at 1500 RPM synchronous speed less the slip required to produce enough torque at full load. The slip for IM under NEMA design B is 3 to 5 % making a typical 4-pole IM running at around 1425 RPM at 50 Hz under full load.

Soon after the IM was developed, the need for varying the speed of the motor is almost unavoidable. Edward C. Lee (2008) states that "the only practical way of providing a variable frequency at that time was to use a DC motor to turn an AC alternator before supplying to an induction motor" (p.4). This method was widely used from the 50s until the 70s. In the 80s many industries start to use the static variable speed drive where motors are allowed to run at different steps of speed, using the variable voltage design. In the mid 90s, the variable voltage, variable frequency (VVVF) drives became common due to its improved control and reduction in power consumption. Up until now, this type of drives has grown to the 21st Century drive of choice. It uses the sine coded pulse width modulation (PWM) method in generating variable frequency using power electronics converters. This chapter explains the relevant characteristics of IM and the architecture of a generic variable speed drive for AC motors.

2.2 Torque-Speed Characteristics of Induction Motors

The understanding on the relationship between torque and speed of an induction motor is essential as part of the investigation on methods of varying the speed of induction motors. Figure 1 shows the torque-speed characteristics of an IM where the X axis represents speed and slip while the Y axis represents torque and current. Motor usually draws up to seven times of the rated current during start-up because at rotor speed = 0, inductive reactance in the rotor winding is very low due to low rotor frequency, thus drawing high current ($X_L = 2\pi fL$). Furthermore, high starting current is needed to overcome the losses in stator and rotor and produce the momentum to rotate the rotor.
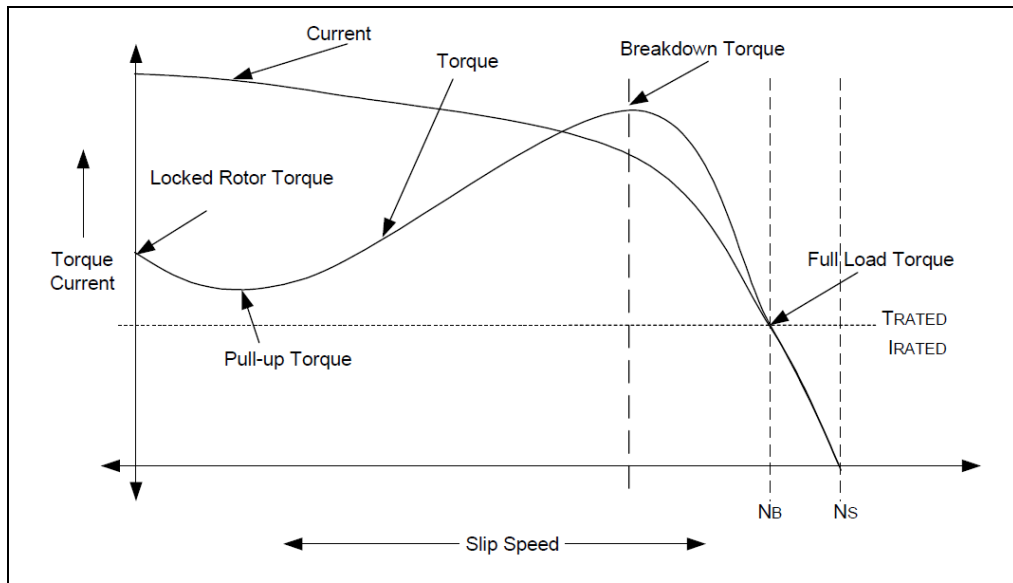
Figure 1: Torque-speed characteristics of induction motor *(Graph extracted from Padmaraja Yedamale (2002). Speed Control of 3-Phase Induction Motor Using PIC18 Microcontroller. Microchip. Pp3.)*

As the speed increases, the current drawn by the motor reduces and at rated speed, the motor draws the rated current and produces the rated torque. At rated speed, if the motor load exceeded its rated torque, the speed starts to drop and slip increases until the load reached a point called breakdown torque where motor will not be able to take any further load and the motor will stall if the load exceeded this point [4]. In

addition, the load current increases quite rapidly when the load is above the rated torque, as shown in Figure 1. This causes winding to heat up and damages its insulation as well as bringing down motor efficiency due to higher losses.

Another important point observed from the torque-speed characteristic is that torque is highly nonlinear with varying speed which is something undesirable in motor speed varying application. "So the variable speed drive must be able to keep the torque constant throughout the speed range" (Yedamale, 2002, p.3). The proposed method of performing such task is to apply the variable voltage variable frequency (VVVF or V/f) method which is discussed in the following section.

2.3 Variable Voltage Variable Frequency (VVVF or V/f) Operations

It is known that motor speed can be controlled by varying the supply frequency. However as frequency reduces, impedance of motor circuit reduces as well drawing higher current thus producing greater magnetic flux. This increases the torque since torque is proportional to magnetic field in the air gap. Therefore applied voltage needs to be reduced to lower down the current so that torque can remain. In order to keep torque constant throughout the speed range, applied voltage and frequency need to be varied at a constant ratio which is what the VVVF or constant V/f control method about. Considering an IM rated at 400 V, 50 Hz; the V/f ratio at rated speed is 8 (400/50). If the motor has to run at 25 Hz, the voltage should be adjusted to 200 V.

As shown in Figure 2, applied voltage and frequency are varied at constant ratio producing constant torque. At rated frequency, applied voltage is adjusted to be at rated voltage. However at above rated frequency, applied voltage can no longer be further increased. Thus any increase in frequency will cause field weakening and reduction in torque due to reduction of line current. The nonlinear reduction of torque is due to the significant increase in windage and frictional losses of the motor at higher speed [3]. IM may not be functioning under certain frequency (around 10 % of

rated frequency) as sufficient power need to be supplied to the motor to overcome motor losses (core, frictional, and windage loss).
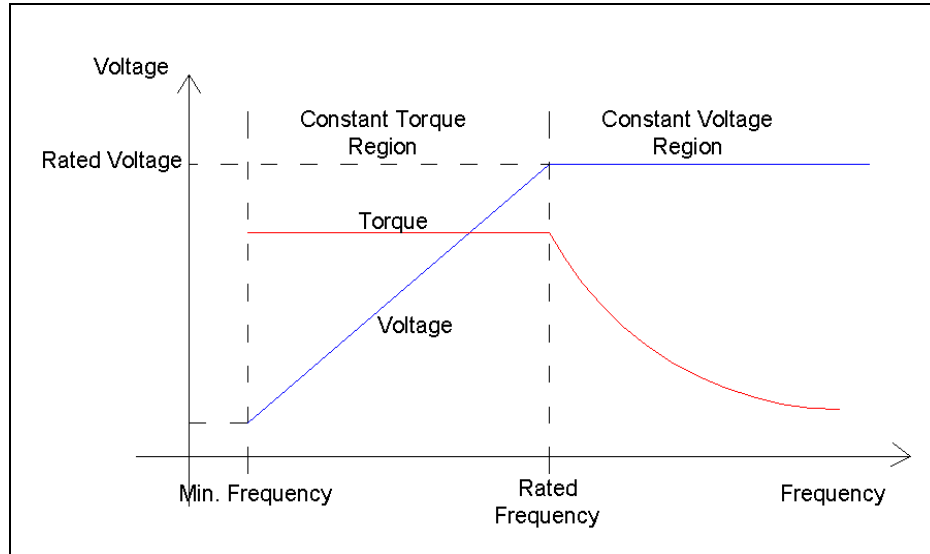
Figure 2: Voltage-frequency relationship of induction motors

2.4 Generic Architecture of an Electronic VSD

Figure 3 shows a 3-phase motor system that incorporates a VSD where the VSD is fixed in between an AC supply and an IM. The three main components of a common VSD include of a rectifier, an inverter, and a control circuit. The constructed VSD prototype for this project include an inverter and a control circuit but not the rectifier as the project focuses only on generating variable frequency output.
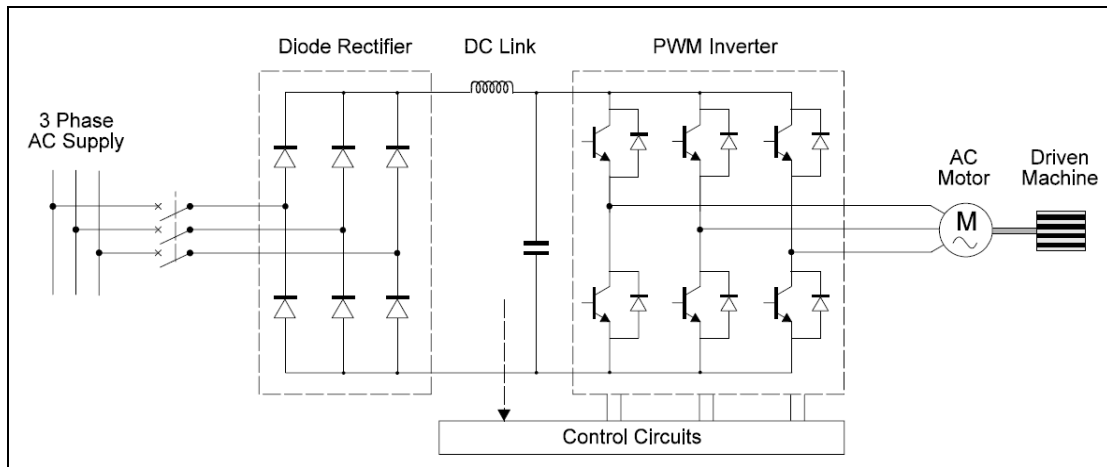
Figure 3: Generic architecture of an electronic VSD (*extracted from Malcolm Barnes (2003). Practical Variable Speed Drives and Power Electronics. Elsevier. p. 156*)

The 3-phase rectifier is used to convert the incoming 3-phase supply into DC voltage which is required by the 3-phase inverter. Voltage ripples are reduced by a DC link that consists of inductors and capacitors. The inverter then converts the DC voltage into AC voltage at variable frequency and supply to a 3-phase IM. The frequency of the output AC is controlled by the PWM switching signals from the control circuit.

2.4.1 The Rectifier

As mentioned, magnitude of applied voltage to the motor needs to be varied simultaneously with applied frequency to maintain V/f ratio. One method of varying the AC output voltage is to vary the magnitude of the DC voltage supplied the inverter. This can be done using a controlled rectifier with power thyristors. The DC output of the rectifier should be smoothened by a filter before being fed into the inverter. Figure 4 shows the circuit of the controlled rectifier with a capacitor filter. The control circuit sends the switching signals to the six thyristors at the required firing delay angle, α, to produce the desired DC voltage magnitude.

The relationship between α and DC output voltage is given by Equation (2.1):

$$V_D = 1.35 \times V_{RMS} \times \cos \alpha \qquad (2.1)$$
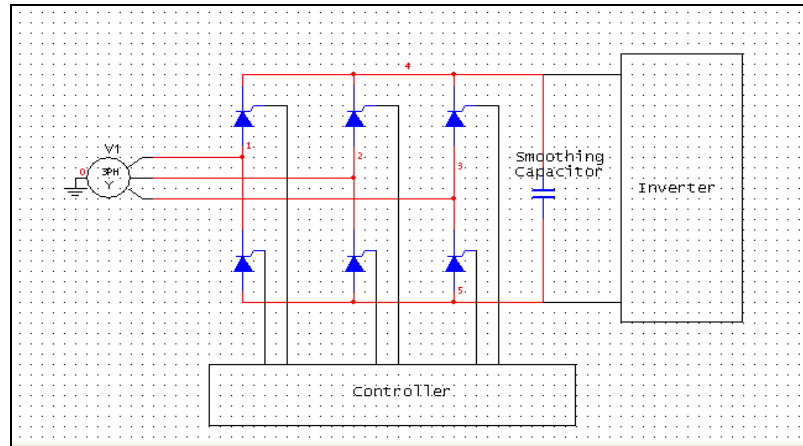
Figure 4: Power thyristor rectifier with smoothing capacitor

The voltage ripple factor of rectifier output is given by Equation (2.2):

$$RF = \frac{\sqrt{V_{RMS}^2 - V_{AV}^2}}{V_{AV}} \qquad (2.2)$$

Normally the ripple must not exceed certain percentage limit and the smoothing capacitor value is calculated [4] from Equation (2.3):

$$RF = \frac{1}{4\sqrt{3}fCR} \qquad (2.3)$$

Where,

f is frequency of AC supply

C is the value of smoothing capacitor

R is the resistance of the load

2.4.2 The Inverter

A three-phase inverter converts DC voltage into AC voltage at variable frequency based on the control circuit. Figure 5 shows an IGBT inverter with six IGBT switches. Lab-Volt's 8837-A three-phase IGBT inverter is used for this VSD prototype.

The inverter switches receive sine pulse width modulated (SPWM) signals from control circuit where a basic SPWM signal is shown in Figure 6. Switches IGBTH1 through IGBTH3 are at the upper side of the inverter while switches IGBTL1 through IGBTL3 are at the lower side. The three upper switches receive the original three-phase SPWM signals while the three lower switches receive the complemented signals of the upper switches.

To each switch a diode is connected in parallel because when the switches are turned on, current flows from the DC supply to the highly inductive motor winding which store current. When the switches are turned off, the diodes provide a path for the current to dissipate off. This ensures that the switches are protected from the feedback current.
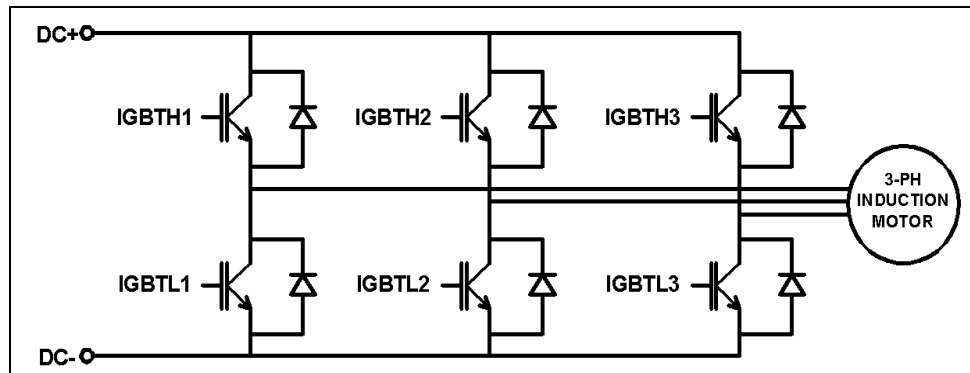
Figure 5: Lab-Volt's 8837-A three-phase IGBT inverter

Figure 6 shows the basic pattern of SPWM signal that need to be generated and fed to the IGBT switches of the inverter to produce a complete sinusoidal cycle at the inverter output terminals. A full duty cycle PWM pulse gives positive peak voltage while a zero duty cycle PWM pulse gives negative peak voltage. Three signals of SPWM are required with each signal being 120° apart from each other.
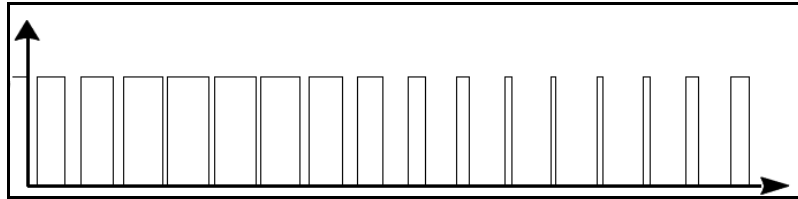
Figure 6: Basic pattern of SPWM signal to produce a complete output cycle

2.4.3 The Control Circuits

The control circuit is the most important component in a VSD as it controls the overall system. The main function of the control system for modern PWM-type AC VVVF drives is to control the semiconductor switches of the PWM inverter by generating and feeding the SPWM signals to the switches. Besides, the control circuit should be able to read and interpret feedback inputs such as speed and process inputs including flow rate and temperature. Most of the drive's control circuits also include external interfacing which includes parameter setting by user and status information.

Some of the common technologies used for the control system include microprocessors and DSP processors. In this project, the author uses PIC microcontroller to perform the control for the VSD prototype as it is low-cost and is suitable for motor control applications.

2.5 Microchip's PIC16F777 Microcontroller

Several models of microcontrollers from Microchip have been investigated to determine one which suits the application of speed control of a three-phase induction motor. The 40-pin PIC16F777 had been selected as it provides three on-chip hardware PWM modules, making it suitable for three-phase induction motor application as three-phase PWM signals are required [5]. Only PIC16F7X7 series of the Microchip microcontrollers have such feature. Most of the other microcontrollers have either two or five PWM modules. Those with two PWM modules require complicated coding to generate the third PWM signal while those with five PWM modules have additional of 24 to 40 pins which are redundant. Figure 7 shows the pin diagram of PIC16F777.

Besides, the PIC16F777 has the 10-bit analog-to-digital-converter (ADC) modules which are necessary in processing the temperature feedback signal which is in the form of an analog signal. With the 10-bit ADC module, temperature analog signal can be converted into digital signal in high precision before using the converted digital signal to vary the frequency of PWM signal in a programming environment.
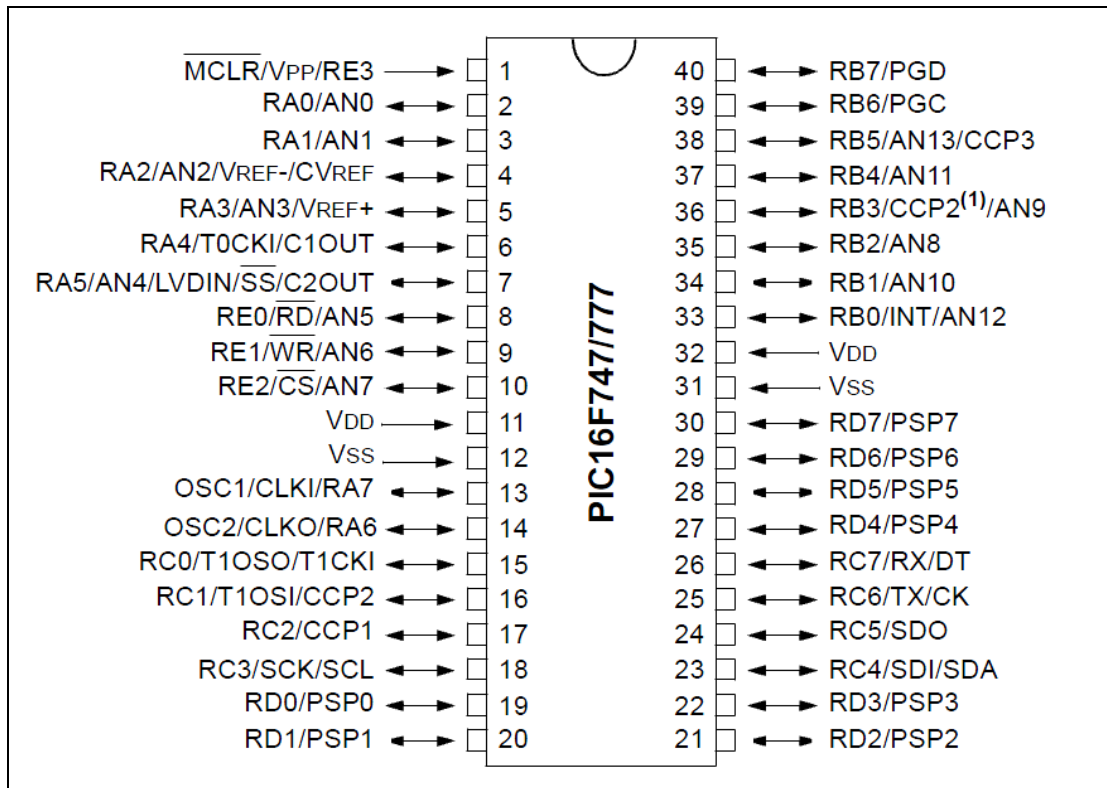
Figure 7: Pin diagram of PIC16F777 microcontroller

CHAPTER 3

METHODOLOGY

Many activities are accomplished in completing this project. This chapter intends to capture the main activities performed by the author. Project work breakdown, Gantt chart, and tools used shall be presented.

3.1 Project Work Breakdown

The project work is divided into five major activities which are (1) front-end activities, (2) research and designing of the VSD, (3) acquisition of components and equipments, (4) simulation on control circuit, and (5) prototyping and experimentation of the VSD.

1. *Front-end Activities*
   1.1 Feasibility study on project
   1.2 Title proposal to supervisor
   1.3 Approval on title

2. *Research and Designing*
   2.1 Overall system of VSD
      2.1.1 Understanding how VSD works
      2.1.2 Evaluation of different method of performing VVVF operation
   2.2 Rectifier with DC link
      2.2.1 Calculation on three-phase rectifier and sizing of DC link
   2.3 Inverter
      2.3.1 Research on switching signal required by three-phase inverter to generate three-phase AC output

5.2 Overall system

    5.2.1 Integrating the control circuit to the inverter and induction motor

    5.2.2 Performance test on VSD prototype

3.2 Gantt Chart

Upon listing out the project work breakdown, a level 2 Gantt chart was outlined to ensure all activities are to be completed within time frame. The Gantt chart is available in the following page.

The project was started in the first week of January 2010 while the final report is to be submitted on 12th of November 2010. As shown in the Gantt chart, FYP part 1 supposed to complete the front-end activities, research and designing, and simulation on control circuit while acquisition of equipments and components; and prototyping and experimentation works were covered in both FYP part 1 and part 2. The key milestones of the project in FYP part 1 includes the completion of simulation works and the completion of a control circuit with ability to generate three-phase variable frequency SPWM signals, while the key milestone in FYP part 2 is the completion of the VSD prototype equipped with all features proposed.

Throughout the project, the author managed to work accordingly with the Gantt chart with minor deviations.

3.3 Equipments and Softwares used

The equipments used throughout the project include:

i.   Three-phase, four-pole induction motor (175W)

ii. AC and DC supply

iii. Lab-Volt's 8837-A three-phase IGBT inverter

iv. Bizchip's Universal PIC Microcontroller Programmer

- To burn programs into the PIC chip which is the PIC16F777

v.   Tektronix's digital oscilloscope

- To monitor SPWM signals

vi. Data Acquisition Interface module

vii. Speed tachometer

viii. Power thyristor rectifier module

ix. Digital multimeter

The softwares used include:

i.   PIC C Compiler

- To compile program written in C language into hex file

ii. Onshon's PIC Simulator IDE

- To simulate the program written

iii. Eagle 5.1.0 Light

- To design printed circuit board

Figure 8: Bizchip's Universal USB PIC Programmer

Figure 9: Tektronix's digital oscilloscope
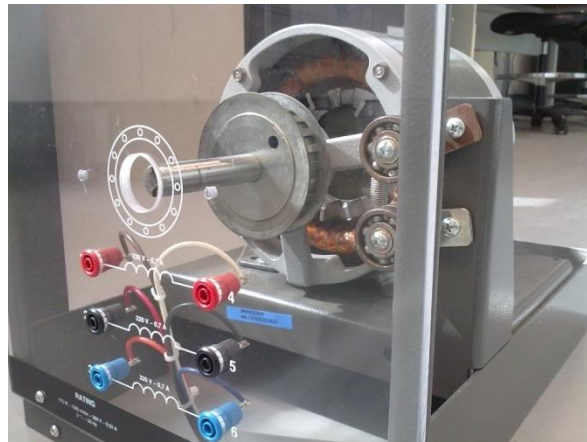
Figure 10: Three-phase, four-pole induction motor

Figure 11: IGBT Inverter module
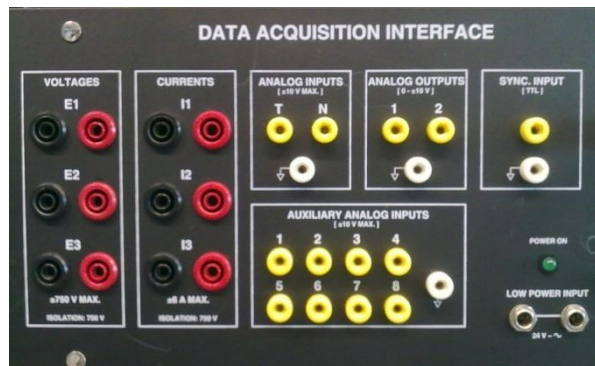
Figure 12: Data acquisition module
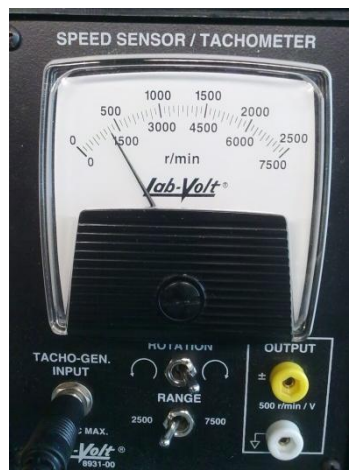
Figure 13: Speed tachometer

CHAPTER 4

RESULTS AND DISCUSSION

This chapter explains in detail on the designing and prototyping of the variable speed drive (VSD) for three-phase induction motor. The VSD is to be used in cooling application where the motor is coupled to a cooling fan to cool certain object. A temperature indicator is used to provide temperature feedback to the VSD. At high temperature, motor speed increases and vice versa. Motor speed can also be manually controlled by user. The system architecture, hardware arrangement, and operation process will be explained. It is necessary to look into the control circuit and explain how three-phase variable frequency SPWM signals are produced. This is followed by the demonstration of simulation and experimental results where actual relationship between temperature sensed and motor speed are shown. Lastly the cost savings of using the VSD prototype are presented where payback period is calculated to be less than three years.

4.1 System Architecture

Figure 14 shows the block diagram of overall system consisting of a VSD and a three-phase IM coupled with a cooling fan. The VSD converts incoming DC voltage to a variable frequency AC voltage based on the feedback temperature of object to be cooled. The relationship between frequency generated and temperature is linearly proportional with frequency ranges from 15 Hz to 50 Hz while temperature ranges from 0 ^{0}C to 100 ^{0}C; meaning that at 100 ^{0}C, the applied frequency will be at 50 Hz. The control circuit can be re-programmed to cater for other ranges as long as the frequency does not exceed motor requirement.

Two modes of operation available for speed control is implemented: automatic temperature control mode and manual user control mode. This allows a better control on the motor speed.

The constructed VSD is designed based on the drive motor, a 175 W, 4-pole, squirrel cage three-phase IM. The VSD includes a three-phase inverter and a frequency control circuit. The Lab-Volt's 8837-A three-phase IGBT inverter is used while the control circuit mainly consists of a PIC16F777 microcontroller, a PWM driver, and a 16-pin LCD display unit. The PWM driver is simply a DM74LS04N Not-gate. Figures 15 and 16 show the main hardware components and the experimental setup.
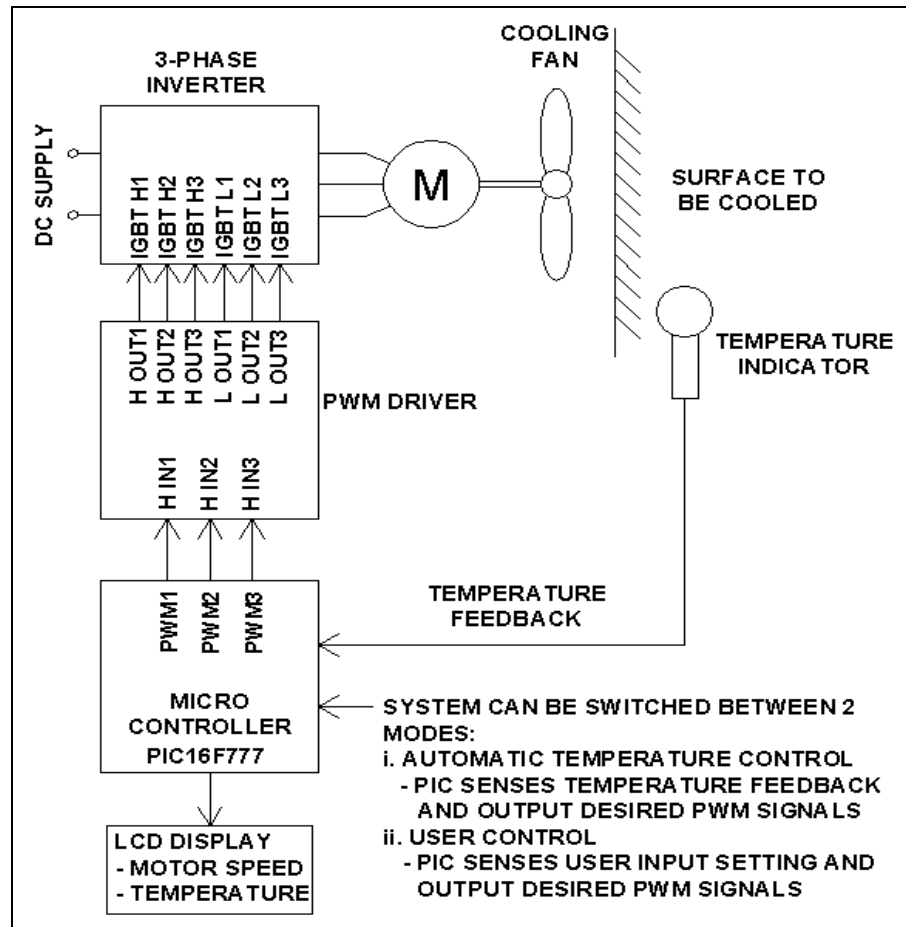
Figure 14: System overview

Control circuit controls output frequency based on feedback temperature and user setting

DC Supply

IGBT Inverter for DC-AC Inverting

Supply to three-phase induction motor
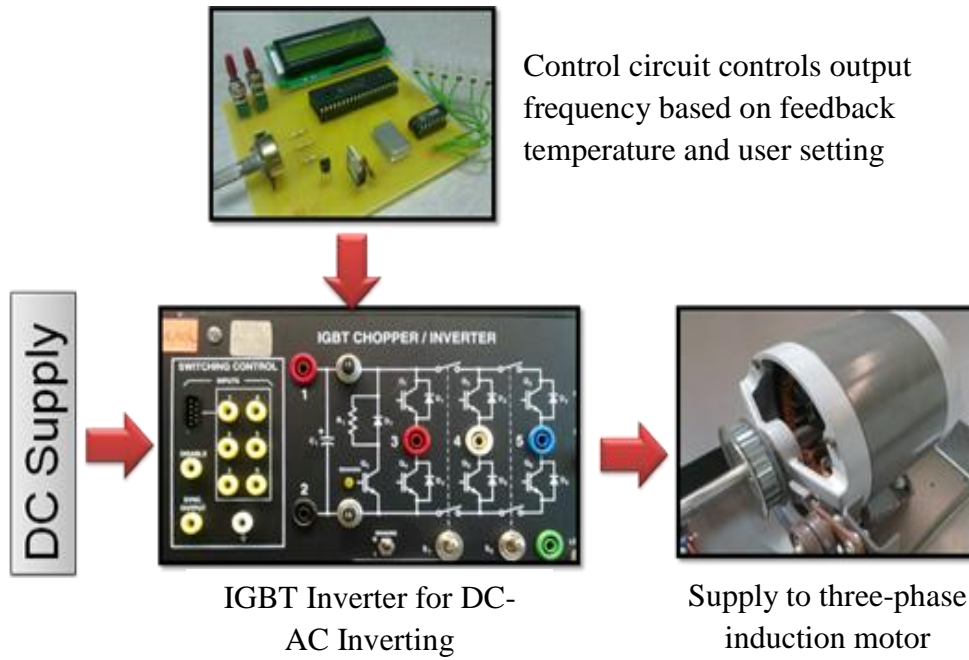
Figure 15: Connection of main hardware components reflecting the block diagram in Figure 14

Figure 16: Experimental setup

4.1.1 The Inverter

A three-phase IGBT inverter as shown in Figure 17 receives SPWM signals from frequency control circuit to produce variable frequency AC. Switches IGBT H1 through IGBT H3 are at the upper side of the inverter while switches IGBT L1 through IGBT L3 are at the lower side. The three upper switches receive the original three-phase SPWM signals while the three lower switches receive the complemented signals of the upper switches. The diodes protect the switches from feedback currents.
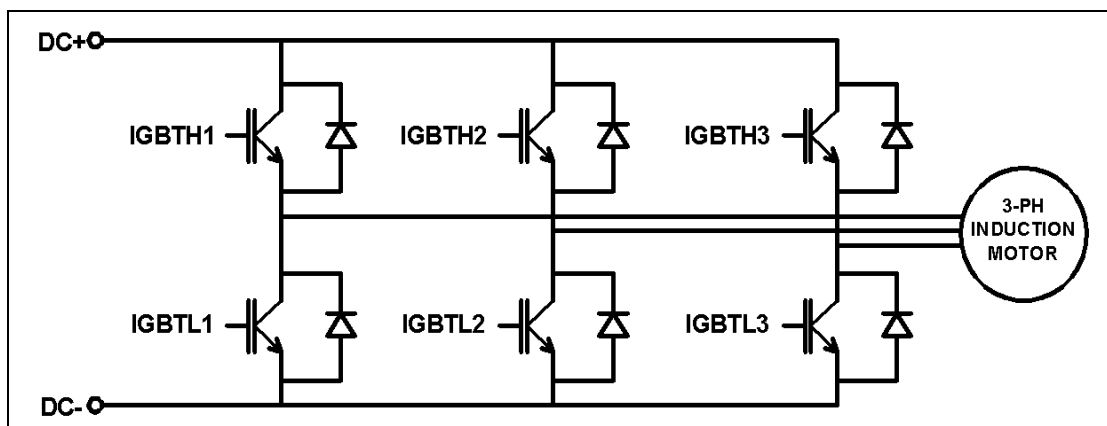
Figure 17: Lab-Volt's 8837-A three-phase IGBT inverter

4.1.2 The Frequency Control Circuit

Figure 18 shows the block diagram and pin assignments of the constructed control circuit. The 40-pin PIC16F777 is chosen as the controller because it is low-cost and it has three built-in PWM modules which are required for a three-phase application. The PIC is programmed using C language. A maximum clock speed of 20 MHz is used to minimize delay in executing the instructions. IC LM35 is used to sense the temperature in this prototype while a variable resistor is used as the speed controller. The temperature and frequency are displayed on the LCD display. Three-phase SPWM signals are generated from the microcontroller and are inverted using a not-gate. The three original signals are fed to IGBT H1 through IGBT H3 while the complimented signals are fed to IGBT L1 through IGBT L3. Figure 19 shows the control circuit built on a breadboard while Figure 20 presents the circuit schematics.

23
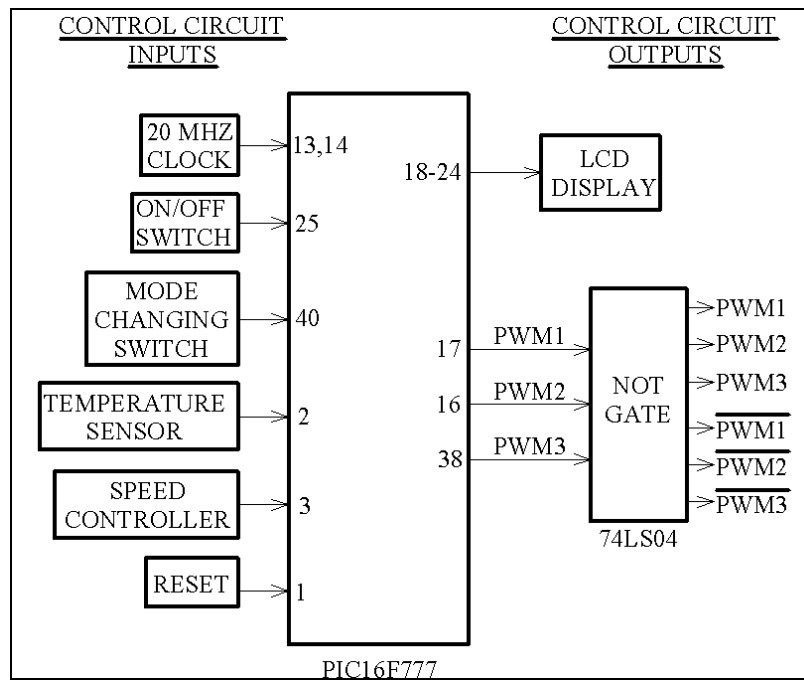
Figure 18: Block diagram of and pin assignment of the control circuit indicating all inputs and outputs
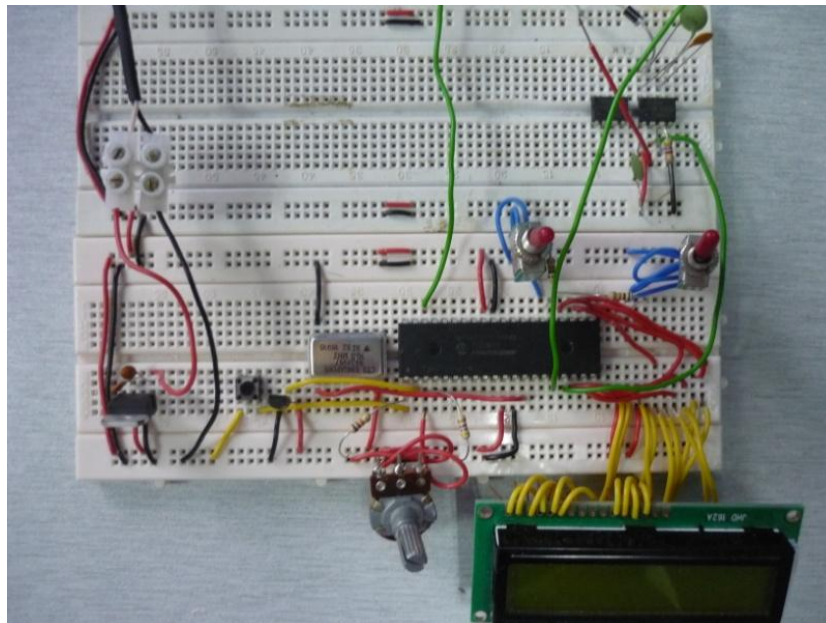
Figure 19: Constructed control circuit on breadboard
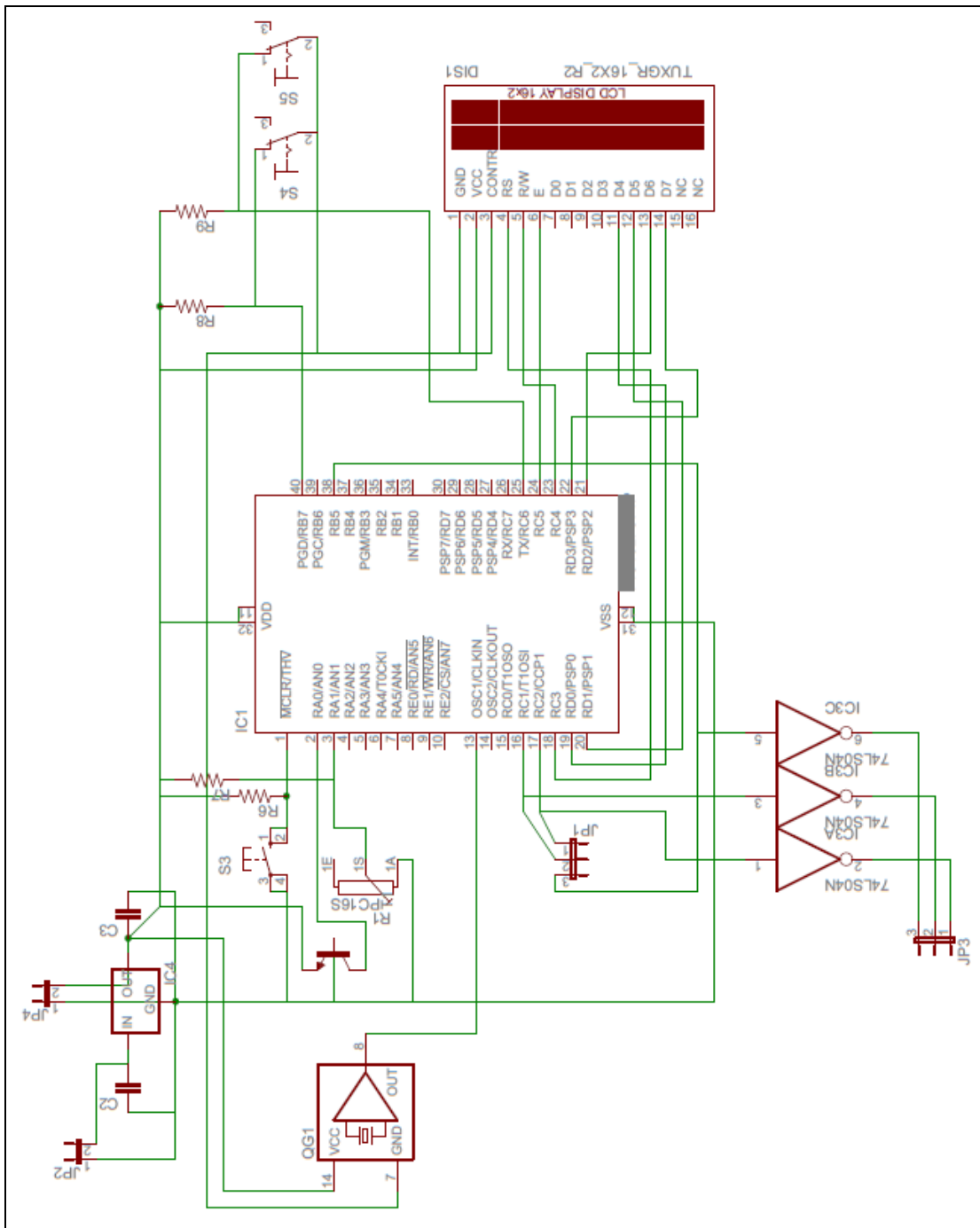
Figure 20: Schematic of the control circuit

25

Printed circuit board (PCB) of the control circuit

The design of PCB layout of the circuit as shown in Figure 21 was made using EAGLE 5.1.0 Light software. Figure 22 shows the fabricated PCB. The fabrication of the PCB was performed by one of the technicians in the PCB's lab.
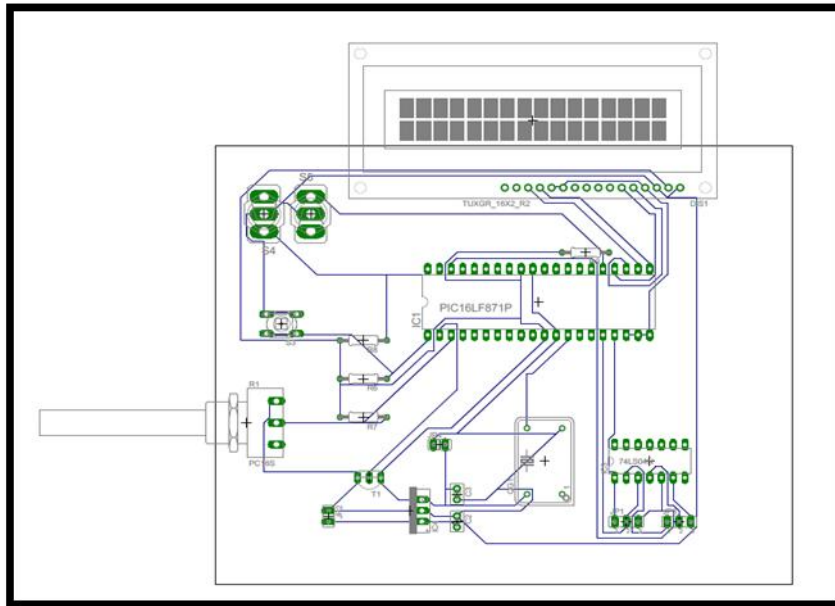
Figure 21: PCB design layout of the controlcircuit
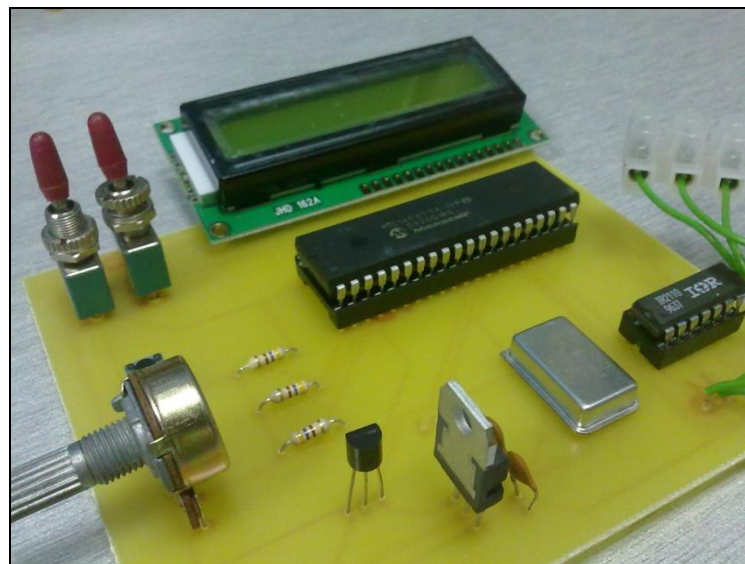
Figure 22: PCB of the control circuit with components soldered

4.2 Programming the microcontroller using C language

4.2.1 The operation flow of the control circuit

Figure 23 shows the flowchart of the control circuit which is reflected in the program coding of the "main" function shown in Figure 24. The sine look-up table is defined in the beginning. This pre-defined table will be used in the program to generate SPWM signals later on. After the system is turned on, PIC reads the Mode Select switch. There are two modes available as mentioned. For temperature control mode, temperature is sensed and displayed on a LCD screen. Then the value of PR2 register is calculated based on the temperature. PR2 is an 8-bit register in the PIC memory whereby *its value determines the period of each PWM cycle* which then control the output frequency produced by the SPWM signals. Further explanations will be given on this statement. Similarly for user control mode, period of PWM pulses is determined by the rheostat value. Synchronous speed of the motor is displayed on the LCD screen under this mode.

This is followed by the generation of three-phase SPWM signals from the PIC using the sine look-up table based on the PWM period determined earlier. At this point, the PIC continuously produces the same SPWM signals until there is any change in any inputs i.e. mode select switch, temperature sensed etc. This allows the PIC to bypass complicated arithmetic instructions which results in a more stable output.

The following sections shall explain in detail on the generation of the variable frequency SPWM signals as well as both the temperature control and user control modes.
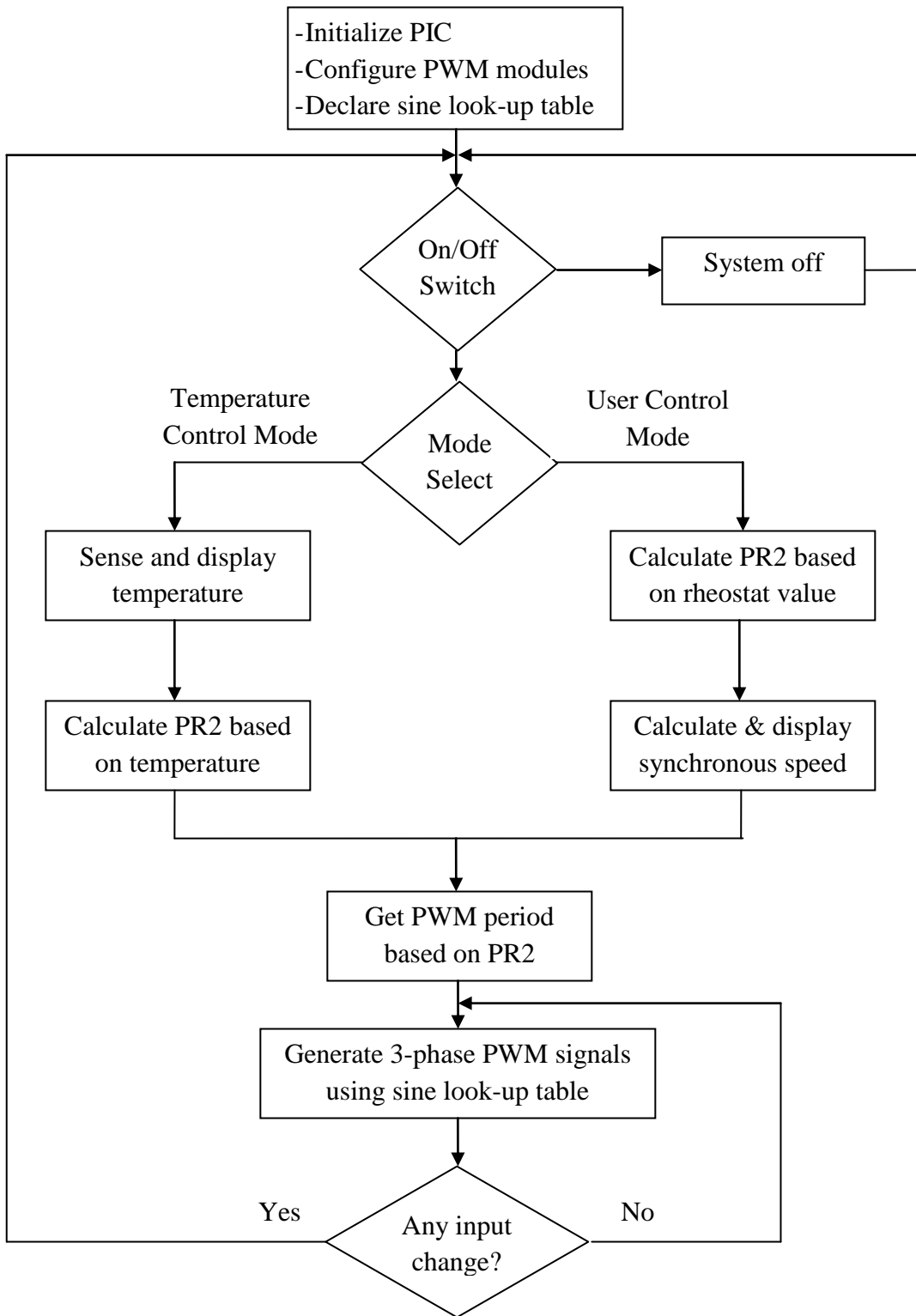
Figure 23: The control circuit flowchart

```
long sin_table[36]={512 601 687 768 841 904 955 993 1016 1024 1016
                    993 955 904 841 768 687 601 512 423 337 256
                    183 120 69 31 8 0 8 31 69 120 183 256 337 423};
void main()
{
    int16 PR2, rheostat_adc, temp_adc;
    int16 delay1;
    float temp;
    int on_off, control_mode;

    setup_ccp1(CCP_PWM);                // Configure CCP1 as PWM module
    setup_ccp2(CCP_PWM);                // Configure CCP2 as PWM module
    setup_ccp3(CCP_PWM);                // Configure CCP3 as PWM module

    setup_adc(ADC_CLOCK_INTERNAL);      //Configure ADC hardware
    setup_adc_ports(ALL_ANALOG);        //pin A0-A7 set as ADC

    lcd_init();
    lcd_putc("\fWelcome to LTH\nTech"); //display welcoming message for 3second
    delay_ms(3000);
    set_tris_c(0x11000000);             //set pin C6 and C7 as inputs

    while(true)
    {
        on_off=input(PIN_C6);           //On/Off switch
        if(on_off==0)                   //system on
        {
            control_mode=input(PIN_B7);   //Mode Select switch
            if(control_mode==0)           //if temperature control mode
            {
            get_temp(&temp, &temp_adc);   //get temperature and display temp
            PR2 = get_PR2(temp);          //get PR2 based on temperature
            }
            else                          //if user control mode
            {
            control_mode_PR2(&PR2,&rheostat_adc); //get PR2 based on rheostat
            }
            delay1 = get_delay(PR2);      //get delay value based on PR2
            setup_timer_2(T2_DIV_BY_16, PR2, 1); //get PWM period based on PR2
            pwm_3phase(delay1,PR2,control_mode,rheostat_adc,temp_adc);//generate 3-
phase SPWM signals
        }
        else                            //system off
        {
        set_pwm1_duty(0);               //set duty cycle of PWMs to zero
        set_pwm2_duty(0);
        set_pwm3_duty(0);

        while(on_off==1)
        {
            on_off=input(PIN_C6);       //check on/off switch
            lcd_putc("\fOFF");
            delay_us(1000);
        }
        }
    }
}
```

Figure 24: Code fraction for the "main" function

4.2.2 Generation of variable frequency SPWM signals

Figure 25 shows the basic pattern of SPWM signal that need to be generated and fed to the IGBT switches of the inverter to produce a complete sinusoidal cycle at the inverter output terminals. A full duty cycle PWM pulse gives positive peak voltage while a zero duty cycle PWM pulse gives negative peak voltage. Three signals of SPWM are required with each signal being 120^0 apart from each other.
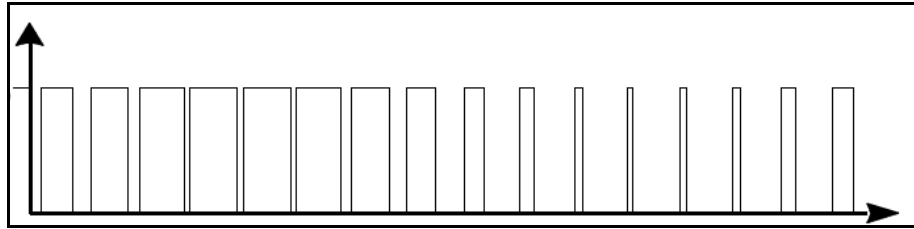
Figure 25: Basic pattern of a SPWM signal to produce a complete output cycle

Since voltage level depends on the duty cycle of PWM pulses, the strategy used in this prototype is to divide a sinusoidal cycle into 36 steps of duty cycle. Five PWM pulses are repeated for each step of duty cycle. Thus, a total of 180 (36 x 5) pulses are available in a complete output AC cycle. The PIC16F777 provides a built-in programming function for the setting of duty cycle of PWM pulses as in (4.1).

$$\text{set\_pwm1\_duty(CCPR1L:CCP1CON<5:4>)} \tag{4.1}$$

CCPR1L:CCP1CON<5:4> is a 10-bit register in the PIC memory and its value determines the duty cycle of PWM pulses generated [5]. Its maximum value is 1024 (2^{10}) which produce a full duty cycle while its minimum value is 0, producing zero duty cycle. The relationship between the angles of a sine wave, their corresponding duty cycles and value of memory CCPR1L:CCP1CON<5:4> is presented in Table 4.1.

TABLE 1: Relationship between angle, duty cycle, and CCPR1L:CCP1CON <5:4>

Angle	Duty Cycle	*CCPR1L:CCP1CON <5:4>	Angle	Duty Cycle	*CCPR1L:CCP1CON <5:4>
0	0.5	512	180	0.500	512
10	0.587	601	423	0.413	423
20	0.671	687	337	0.329	337
30	0.75	768	256	0.25	256
40	0.821	841	183	0.179	183
50	0.883	904	120	0.117	120
60	0.933	955	69	0.067	69
70	0.97	993	31	0.03	31
80	0.992	1016	8	0.008	8
90	1	1024	0	0	0

*CCPR1L:CCP1CON<5:4> = [sin (angle) + 1] x 1024 / 2

Table 4.1 demonstrates how the sine look-up table is formed based on the values of register CCPR1L:CCP1CON<5:4>. The table is used as part of the program in generating sine waves.

As for the realization of variable frequency AC output, the strategy used here is to control the period of each PWM pulse and at the same time maintaining a total of 180 pulses in a complete cycle. When the period of each PWM pulse is reduced, the total time taken to generate 180 pulses is also reduced. Thus the period of one cycle of AC output is reduced and hence a higher output frequency is realized. Reversely, when the period of each pulse is increased, a lower output frequency is generated.

The PIC16F777 provides a built-in function that allows setting of period of PWM pulses, as in (4.2).

$$setup\_timer\_2(TMR2\_prescaler\_value, PR2, 1) \qquad (4.2)$$

The period of each PWM pulse is calculated as in (4.3).

$$PWM\ Period = (PR2+1) \times 4 \times T_{OSC} \times TMR2\ prescaler\ value \qquad (4.3)$$

31

The period of oscillator, T_{OSC} is fixed at 1 / (20 MHz) while the timer 2 prescaler value is limited to 1, 2, 4, or 16 [5]. Thus the period of PWM pulses depends very much on the value of register PR2. It is an 8-bit register and it can be set from 0 to 255. Increasing value of PR2 results in reduction of AC output frequency, and vice versa.

The program coding in the function as shown in Figure 26 is used to generate three-phase SPWM signals at certain output frequency. As mentioned before, the output frequency depends on the period of each PWM pulse and the period is determined by the value of PR2 register. This function receives the PR2 value and produce SPWM signals at desired output frequency. The PR2 value is being calculated in other functions based on the temperature sensed or user setting before being sent to this function. Besides, this function also receives other variables which are used to track whether there is any change in the inputs. This is to enable the program to bypass the complicated program instruction if there is no change in the inputs.

```
void pwm_3phase(int16 delay1,int16 PR2,int control_mode,int16
rheostat_adc,int16 temp_adc)
{
   int8 sin_counter1, sin_counter2, sin_counter3, index1, index2, cycle;
   float scale_value;
   long new_sin_table[36];
   int new_control_mode,input_variation=0,on_off=0;
   int16 new_rheostat_adc, new_temp_adc;

   scale_value = PR2 * 0.0039;

   new_sin_table[0] = 512 * scale_value;

   if (PR2>63)
   cycle=25;
   else if (PR2<=63 && PR2>45)
   cycle=40;
   else
   cycle=55;

// output_high(PIN_B1);
   while ((on_off==0) && (input_variation==0))
   {
   for(index2=0; index2<cycle; index2++)
   {
      for(sin_counter1=0; sin_counter1<=35; sin_counter1++) //to have 36
steps of duty cycle over 360degree of each phase
      {
         set_pwm1_duty(new_sin_table[sin_counter1]); //PWM phase 1
         sin_counter2 = (sin_counter1+24)%36;       //120degree betw. phase
```

```
        set_pwm2_duty(new_sin_table[sin_counter2]); //PWM phase 2
        sin_counter3 = (sin_counter1+12)%36;          //240degree betw. phase
1 & 3
        set_pwm3_duty(new_sin_table[sin_counter3]); //PWM phase 3

        index1 = sin_counter1 + 1;
        new_sin_table[index1] = sin_table[index1] * scale_value;
        delay_us(delay1);
    }
  }
/*The following check whether there is any changes in inputs
  The program will remain in this function and SPWM signals will be generated
continuously
  Only when input variation is sensed, the program will go back to the main
function*/
  on_off = input(PIN_C6);
  new_control_mode = input(PIN_B7);
  if (new_control_mode == control_mode)  //no change in mode
  {
     if (control_mode == 1) //user control mode
     {
        set_adc_channel(1);  //set pin A1 as adc
        delay_us(20);
        new_rheostat_adc = read_adc();
        if ((new_rheostat_adc >= (rheostat_adc + 39))||(new_rheostat_adc <=
(rheostat_adc - 39)))
        input_variation = 1;
        else
        input_variation = 0;
     }
     else
     {
        set_adc_channel(0);  //set pin A0 as adc
        delay_us(20);
        new_temp_adc = read_adc();
        if ((new_temp_adc >= temp_adc + 2)||(new_temp_adc <= temp_adc - 2))
        input_variation = 1;
        else
        input_variation = 0;
     }
  }
  else
  input_variation = 1;
  }
}
```

Figure 26: Program coding for the function that generate 3-phase SPWM signals

Figure 27 shows the waveforms of phase 1 and phase 2 of the three-phase SPWM signals generated from the PIC at the output frequency of 45 Hz. A Tektronix digital oscilloscope is used to display the waveforms. Phase 3 is not displayed as the oscilloscope has only two channel inputs. It can be observed that the signals follow a sine wave pattern which is similar to Figure 25.

Figure 27: SPWM signals (phase 1 and phase 2) at output frequency of 45 Hz

Figure 28 shows the enlarged waveforms to have a clearer view on each PWM pulse. It can be seen that five PWM pulses are repeated for one step of duty cycle, and there are 36 steps of duty cycle that produce the sinusoidal PWM signals. This means that there are 180 (5 x 36) PWM pulses in one full cycle of sinusoidal SPWM signal. These signals control the output frequency of the PWM signals thus controlling the period of each PWM pulse.
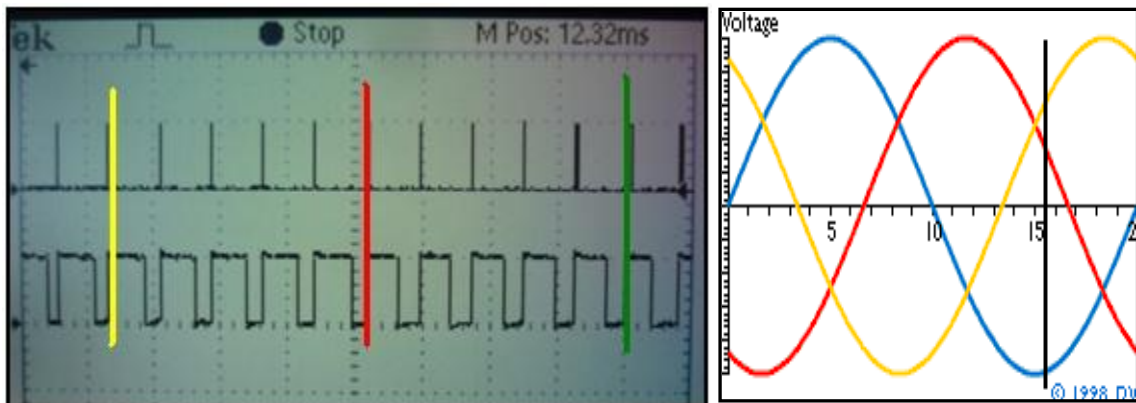

Figure 28: Enlarged SPWM signals; and comparison between SPWM signals and sine waves

Figure 28 also verifies that the SPWM signals for both phases are 120^0 apart. The vertical black line on the sine waves shows the instance where the SPWM signals correspond with the sine waves. Phase 1 of the SPWM signals (upper waveforms) has a duty cycle of almost zero which corresponds to the near negative peak of the blue color sine wave (phase 1 of the sine waves) while phase 2 of the SPWM signals has a duty cycle of around 70 % which corresponds to level of yellow color sine wave (phase 2 of the sine waves).

4.2.3 Temperature Control Mode

The previous section has shown how the frequency of output voltage of the VSD is varied using the PWM method where the output frequency is determined by the value of PR2 register. The PR2 value varies with the inputs of the system. If the system is operating under temperature control mode, the temperature sensed determines the PR2 value. Similarly under user control mode, the value of rheostat controller determines the PR2 value. This section explains the first mode which involves temperature sensing and how the temperature affects the PR2 value.

The device used to sense temperature in the circuit is a LM35 chip, shown in Figure 29 The output of which is an analog signal is fed into pin 2 or PIN_A0. All 8 pins of port A are compatible for analog to digital conversion (ADC).
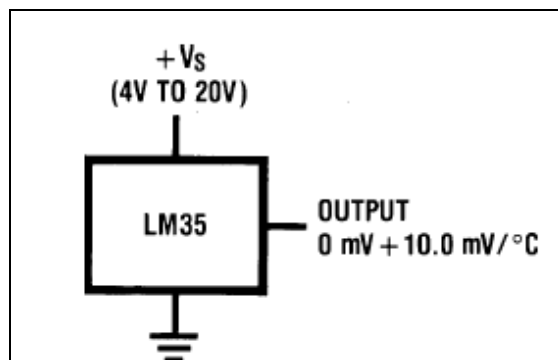
Figure 29: LM35 Temperature Sensor

Figure 30 shows a part of coding which is the function of temperature sensing and displaying. The temperature is obtained as in (4.4):

$$temp = \frac{\dfrac{temp\_adc}{1024 levels} \times 5Volts}{10m{V}\big/{_0 C}} \qquad (4.4)$$

The 1024-level is derived from the fact that the ADC is of 10 bits. The variable *temp_adc* is the reading from the temperature sensor which is converted from analog to digital 10-bit signal. Comments are provided to explain the command functions. A total of seven pins of the PIC are connected to a LCD screen to send digital signals to the LCD.

```
void get_temp(float *temp, unsigned int16 *temp_adc)
{
    lcd_putc("\fTemp Control");
    set_adc_channel(0);                      //set A0 as ADC
    delay_us(20);
    *temp_adc = read_adc();                  //read A0 analog value
    *temp = *temp_adc * 0.4888;              //get temperature
    lcd_gotoxy(1,2);
    printf(lcd_putc,"Temp:%.1f",*temp);      //display temperature
}
```

Figure 30: Coding fractions on obtaining ambient temperature

Figure 31: LCD screen displaying ambient temperature in degree Celsius

36

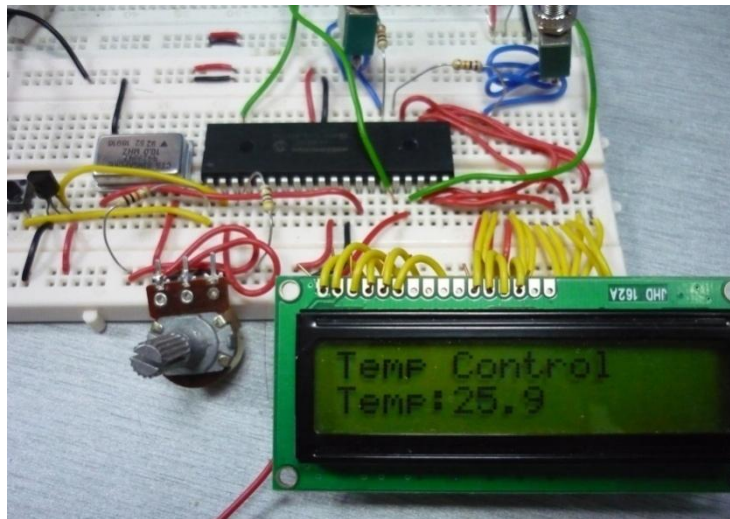After obtaining the temperature, value of PR2 register needs to be calculated. Figure 32 shows the program coding of the function used to convert temperature sensed into desired value of PR2. The relationship between temperature and frequency is linear. The range of temperature is from 0 to 100 ^{0}C while the range of frequency is from 15 Hz to 50 Hz. The temperature 100 ^{0}C correspond to 50 Hz (maximum speed). The relationships between temperature (T), frequency (f), and PR2 value are shown as in (4.5) to (4.8).

$$f = 0.35T + 15 \qquad (4.5)$$

$$Period_{PWM} = 4 \times 16 \times \frac{1}{20MHz} \times PR2 \qquad (4.6)$$

$$Period_{PWM} = \frac{1}{180 \times f} \qquad (4.7)$$

Solving the equations (4.5) to (4.7), the following relationship is obtained as in (4.8).

$$PR2 = \frac{1736}{0.35T + 15} \qquad (4.8)$$

```
int16 get_PR2(float temp)
{
   int16 PR2;
   float store;

   if(temp>100.00)   //PR2 is set to lowest value if temp > 100 celsius
   {
      PR2=33;
   }
   else if (temp<0.00) //PR2 is set to highest value if temp < 0 celsius
   {
      PR2=115;
   }
   else  //if 0<temp<100, use formula to calculate PR2
   {
      store = 1736.00 / (0.35 * temp + 15.00); //formula to get PR2
      PR2 = (int)(store);  //change float to int type
   }
   return (PR2);  //return PR2 to main function
}
```

Figure 32: Function to obtain PR2 value based on input temperature

The PIC is programmed in such a way that the output frequency will not go beyond the limits which are 15 Hz and 50 Hz when the temperature exceeds the upper and lower limits which are 100 ^{0}C and 0 ^{0}C. Figure 33 shows the SPWM signals generated when the temperature equals to 25.9 ^{0}C.
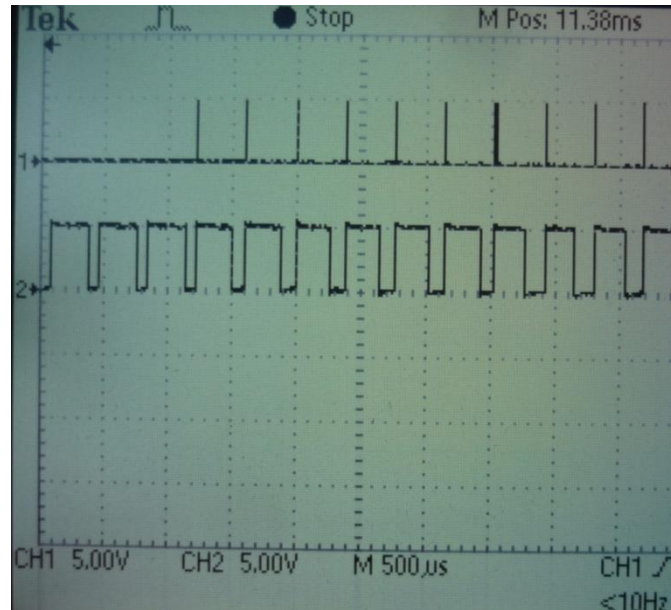

Figure 33: Resulted SPWM waveforms when temperature is at 25.9 degree Celsius

4.2.4 User Control Mode

In this mode user can control the output frequency by adjusting the rheostat. Output frequency will in turn control the speed of motor. Figure 34 shows the connection of rheostat to produce a variable analog signal into pin 2 or pin_A1 which is a 10-bit ADC module. As user adjusts the rheostat, the PIC interprets the analog signal and changes the output frequency according to the analog input. Figure 35 shows a part of the code for obtaining PR2 value and corresponding RPM value as well as displaying the RPM value.
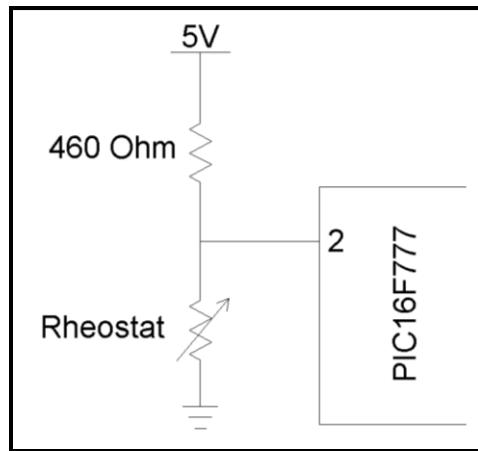
Figure 34: Rheostat connection

```
int16 control_mode_PR2()
{
    int16 rheostat_adc; float storePR2,storeRPM; int16 PR2; double RPM;

    lcd_putc("\fUser Control");
    set_adc_channel(1);                         //set pin A1 as adc
    delay_us(50);
    rheostat_adc = read_adc();                  //read analog input at pin A1
    storePR2 = 0.264 * rheostat_adc + 68.00;    //formula to get PR2
    PR2 = (int16)(storePR2);                     //change PR2 from float to integer
    storeRPM = -5.845 * PR2 + 1897.45;          //get RPM based on PR2
    RPM = (double)(storeRPM);                    //change RPM from float to double
    lcd_gotoxy(1,2);
    printf(lcd_putc,"RPM:%.3e",RPM);            //display RPM
    return(PR2);
}
```

Figure 35: Code function to obtain PR2 and synchronous speed based on rheostat analog value
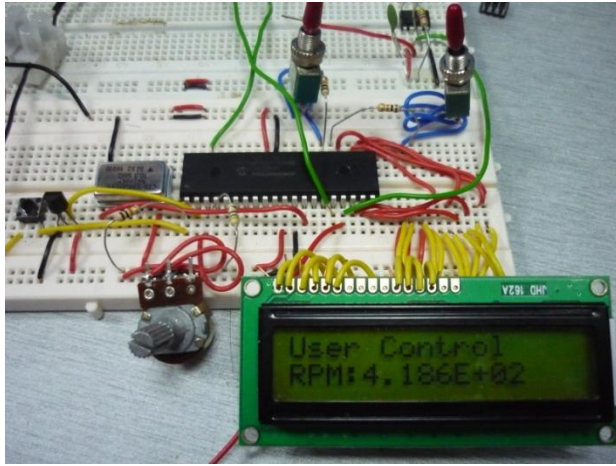
<center>(i)</center>
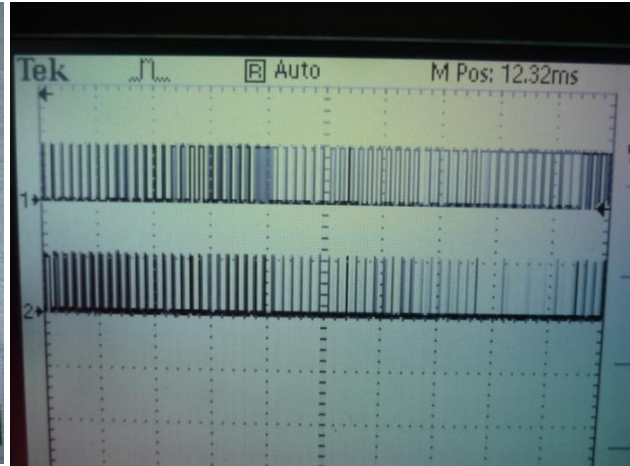
<center>(ii)</center>

<center>(iii)</center>
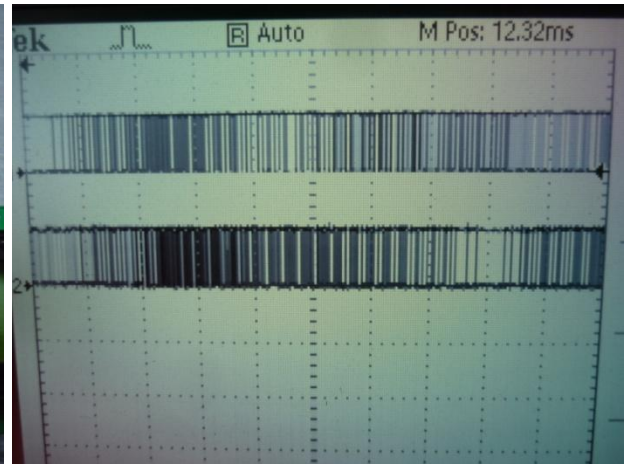
<center>(iv)</center>

<center>Figure 36: Results for User Control Mode</center>

Figure 36 (i) and (iii) show the display on LCD screen where each of them is for different rheostat adjustments while Figure 36 (ii) and (iv) are the corresponding PWM signal waveforms. It is observed that the frequency of waveform for 1266 RPM is higher than the waveform for 418.6 RPM and the exact period of each PWM cycle can be calculated using the equations and methods explained above.

Thus the above results and discussion highlights on the generation of three-phase SPWM signals at variable frequency based temperature feedback or controller setting depending on which mode the system is operating at.

4.3 Experimental setup: Integrating the control circuit to the motor

The previous sections have shown how the control circuit generates the SPWM signals. This section proceeds with the integration of overall system which include the control circuit, the three-phase inverter, and the three-phase induction motor (IM).

It is intended to show the result of an experiment where the VSD prototype is used to control the speed of a 3-phase, 4-pole, 50 Hz, 175 W, 1360 RPM, 380 V, 0.52 Amp induction motor. The connection of the circuit is based on Figure 37. The line current and phase voltage of the motor are measured. The motor is being run under no-load condition. The experimental setup is shown in Figure 16 in the early part of this chapter.
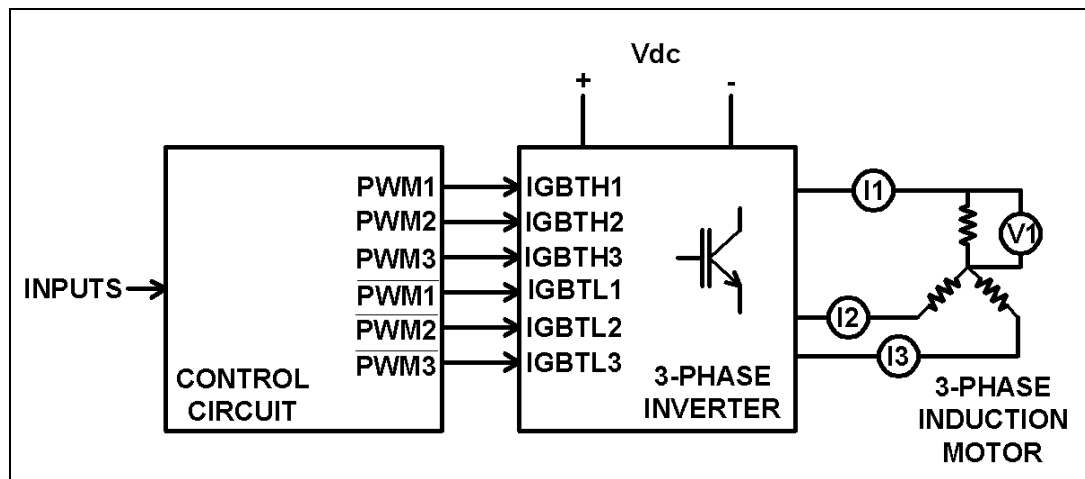

Figure 37: Connection used in the experiment

4.3.1 Temperature Control Mode

Under this operating mode, it is difficult to run the experiment under the required temperature in real condition. Thus the temperature sensed need to be simulated by assigning the temperature reading in the program itself. Table 4.2 shows the measured parameters against the set value.

Table 4.2 Experimental Result for Temperature Control Mode

Simulated temperature (Celsius)	Set Value		Measured				
	Freq (Hz)	Synchronous speed (RPM)	Freq. (Hz)	Speed (RPM)	Speed regulation (%)	Line voltage (V)	Line current (A)
0	15	450	15.04	425	-5.56	114	0.11
14.3	20	600	20.57	570	-5.00	152	0.11
28.6	25	750	25.73	725	-3.33	190	0.12
42.9	30	900	29.77	880	-2.22	228	0.12
57.1	35	1050	35.89	1020	-2.86	266	0.12
71.4	40	1200	40.15	1175	-2.08	304	0.12
85.7	45	1350	45.21	1325	-1.85	342	0.12
100	50	1500	49.67	1470	-2.00	380	0.12

Table 4.2 shows that as temperature increases, frequency and synchronous speed increase. The motor actual speed follows the generated synchronous speed with average speed regulation of around -2 % which is believed to be the slip of the IM. The line current remains at around 20 % of rated current throughout the speed range as the motor is operating at no-load. The line voltage is increased to maintain the V/F ratio for any increase in frequency. Figures 38(a) to (g) show the line current and phase voltage of the motor which were captured using computerized oscilloscope.
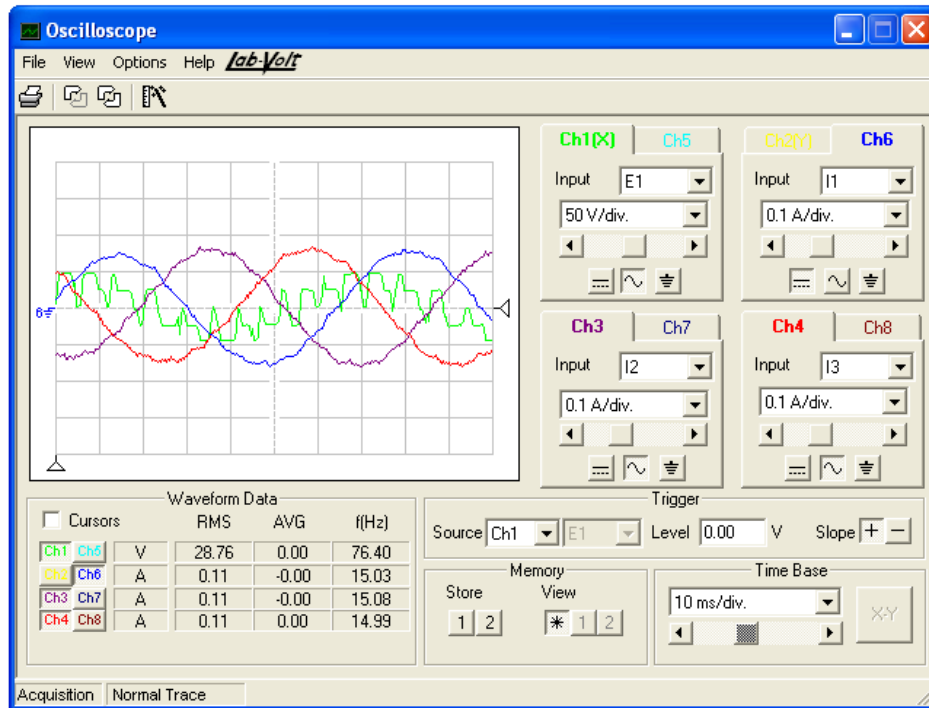
Figure 38 (a): Temperature = 0 ^{0}C; freq. = 15.04 Hz

Figure 38 (b): Temperature = 14.3 ^{0}C; freq. = 20.57 Hz

43

Figure 38 (c): Temperature = 28.6 ^{0}C; freq. = 25.73 Hz

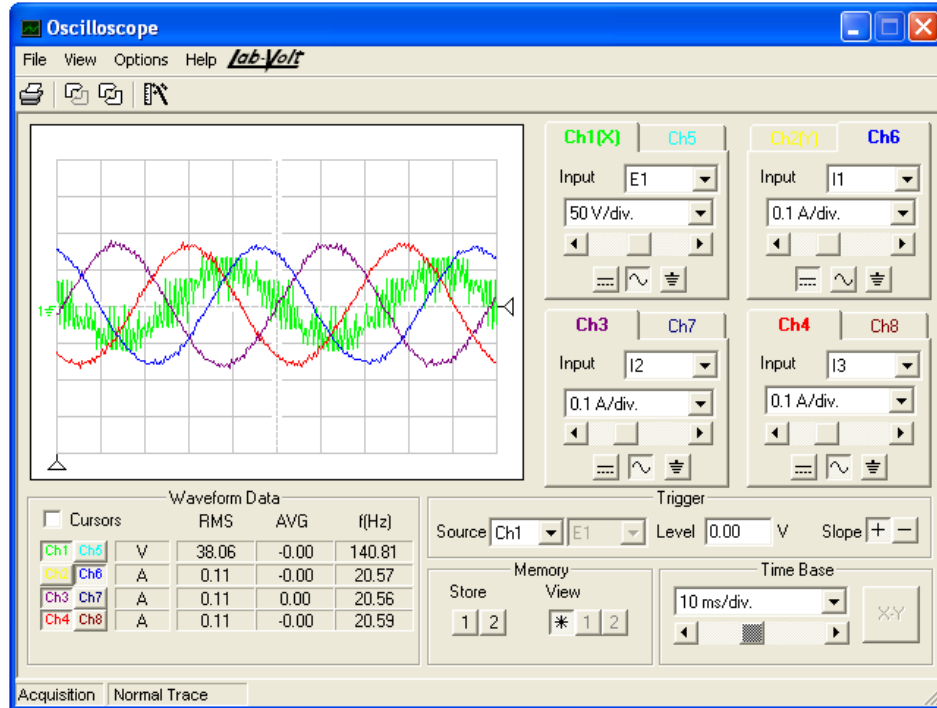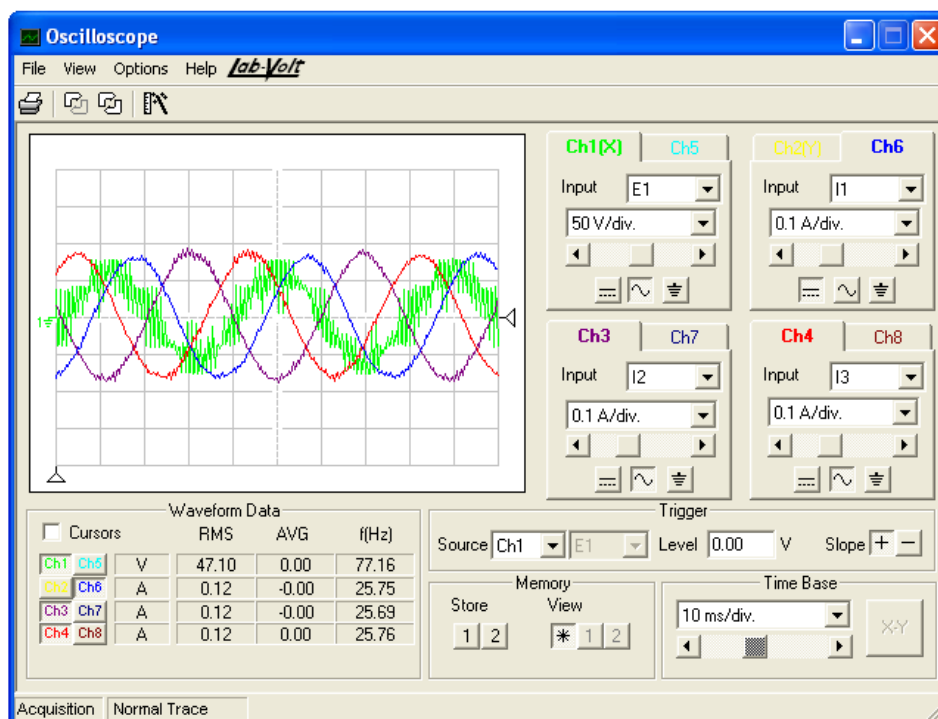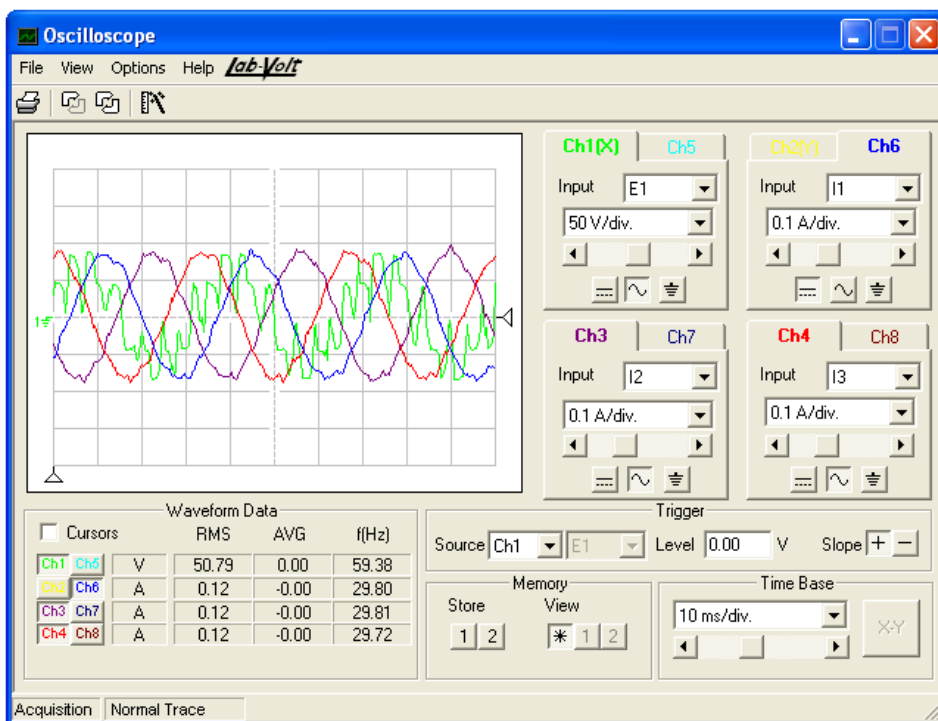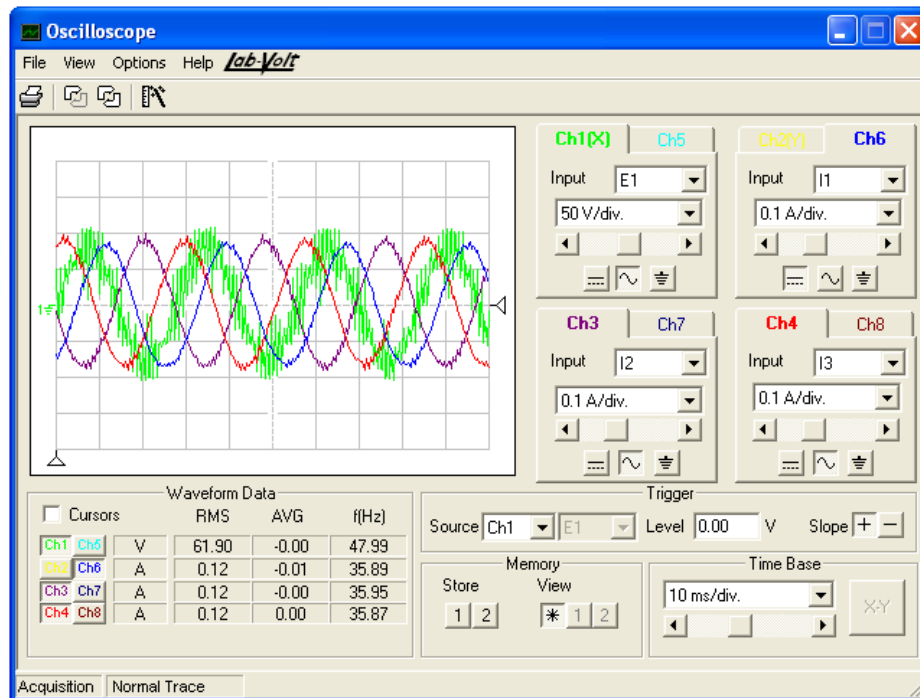Figure 38 (d): Temperature = 42.9 ^{0}C; freq. = 29.77 Hz

44

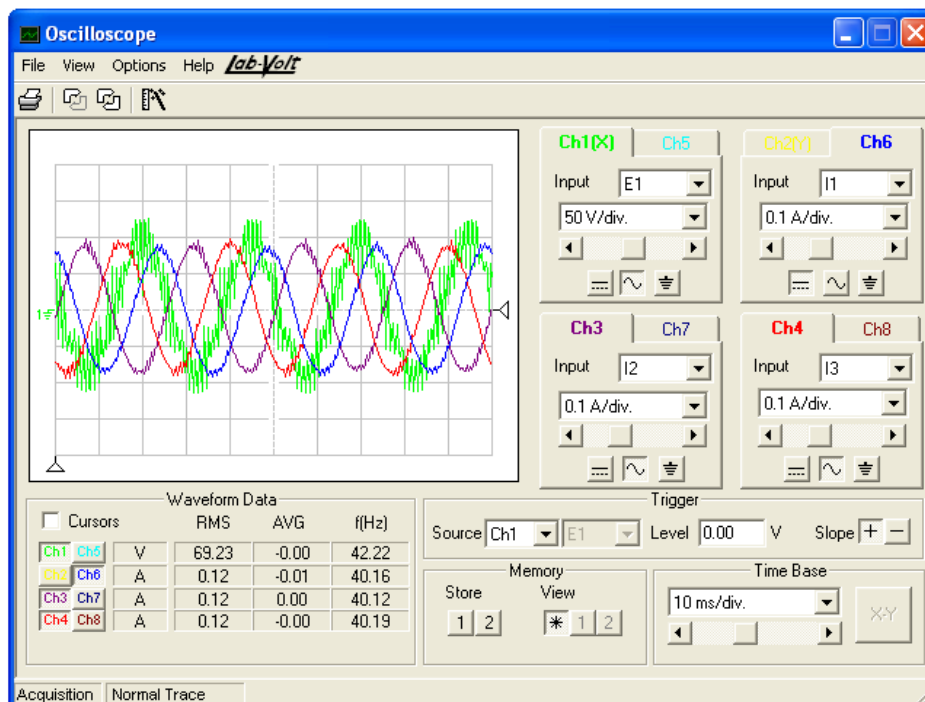Figure 38 (e): Temperature = 57.1 ^{0}C; freq. = 35.89 Hz

Figure 38 (f): Temperature = 71.4 ^{0}C; freq. = 40.15 Hz
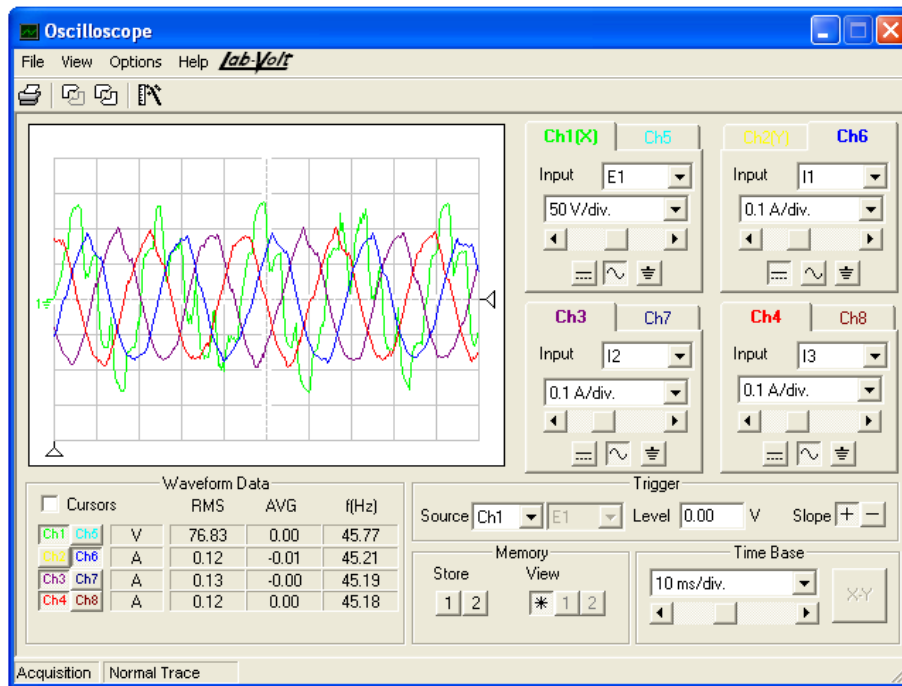
Figure 38 (g): Temperature = 85.7 ^{0}C; freq. = 45.21 Hz

4.3.2 User Control Mode

Similar experiments have been carried out for user control mode as well. Under this mode, the rheostat controller is used to adjust the synchronous speed manually and the results are being tabulated as in Table 4.3.

It is clearly shown that the result for this user control mode is almost the same as compared to the temperature control mode. The measured frequencies are close to the set frequencies and speed regulations are around 6 to 9 % which are believed to be the slip of the induction motor. The waveforms for this experiment are not presented as they are similar to the waveforms generated from the temperature control mode.

Table 3: Experiment Result for User Control Mode

Set Value		Measured				
Synchronous speed (RPM)	Freq. (Hz)	Freq. (Hz)	Speed (RPM)	Speed regulation (%)	Line voltage (V)	Line current (A)
466	15.5	15.38	450	-3.43	114	0.11
612	20.4	20.65	590	-3.59	152	0.11
751	25	25.59	725	-3.46	190	0.12
893	29.8	29.61	875	-2.02	228	0.12
1038	34.6	35.25	1020	-1.73	266	0.12
1215	40.5	40.72	1190	-2.06	304	0.12
1345	44.8	44.46	1325	-1.49	342	0.12
1499	50	49.55	1475	-1.60	380	0.12

4.4 Cost Savings Generated from the Application of the VSD Prototype

Tables 4.2 and 4.3 show that line voltage is lower at lower frequency (or speed) while line current does not change. Thus power consumption of the motor reduces linearly with decreasing speed. The developed VSD prototype uses an inverter with rated current of 1.5 amps and it can support a 3-phase IM up to 500 W rating (150 % overcurrent tolerance). To compute the savings in cost, consider applying the VSD to a 500 W IM with the following assumptions:

i.   Motor operates at 85 % efficiency and 6,000 hrs per annum,
ii.   Average speed is reduced by 20 %,
iii. Tariff rate is 0.378 RM/kWh, and
iv.   VSD's starting cost is around RM 700.

The savings in cost and payback period are computed as follows:

Annual saving

$= $ Input power $\times$ speed reduction factor $\times$ running hour $\times$ tariff rate

$= (500W/0.85) \times 0.2 \times 6000$ hrs $\times 0.378$ RM/kWh

$= $ **RM 267**

Payback period

$= $ RM 700 / RM 267

$= $ **2.62 years**

For the applications of motors with higher ratings, the main component which has to be upgraded is the inverter. A higher current rating of the IGBT switches and feedback diodes is required. The same designed control circuit can be used for any motor size.

As motor size increases, the starting cost of the VSD increase. However annual savings will increase as well as bigger motors draw higher currents. The payback period should be around the same.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

The designing and prototyping of the VSD is successfully demonstrated with proven experimental results and cost savings. The prototype is able to run an induction motor at desired speed based on temperature feedback and user setting. The VSD prototype mainly consists of a 3-phase inverter and a low-cost frequency control circuit which includes the PIC16F777 microcontroller. The control circuit is able to sense temperature and user setting before generating variable frequency 3-phase SPWM signals which results in variable frequency AC output voltage. A pre-defined sine look-up table is used to come out with sine waveforms while variable output frequency is realized through varying the period of each PWM pulse. With payback period of less than three years, the prototype is believed to be marketable as the industry is focusing on sustainability.

This project was selected as one of the finalist in the Engineering Design Exhibition 26 (EDX 26) held in UTP, October 2010 and it won the Bronze Medal Award in the exhibition.

5.2 Recommendations

This basic workability of the VSD prototype has been proven in this project. However, additional features need to be incorporated before the product can be marketed. The following are some of the features recommended:

i. Protective circuit such as fuse or circuit breaker should be connected at the incoming side to protect the drive from electrical faults.

ii. A casing should be constructed to house the components. Proper ingress protection (IP) rating should be considered based on applications. Cooling fan may be included. The casing should be earthed.

iii. A more user-friendly interfacing can be implemented which include a wider LCD screen and LED indication lights that indicate motor status such as "Run", "Ready", and "Fault".

REFERENCE

[1]   Jukka Tolvanen (2008). *Saving Energy With Variable Speed Drives.* ABB. World Pump June 2008 Press.

[2]   Rakesh Parekh (2004). VF Control of 3-Phase Induction Motors Using PIC16F7X7 Microcontrollers. Microchip. Pp3-8.

[3]   Wikipedia – The Free Encyclopedia, Ripple (Electrical). 16 March 2010. Retrieved from: < http://en.wikipedia.org/wiki/Ripple_(electrical)>

[4]   Aníbal T. De Almeida, Fernando J. T. E. Ferreira, Paula Fonseca. *VSDs For Electric Motor Systems*. ISR – University of Coimbra. Chapter 1, pp 7-8.

[5]   Debbie Hobbs, Chris Monson (2003). Variable Speed Drives. InsideENERGY May 2003 Press.

[6]   Padmaraja Yedamale (2002). *Speed Control of 3-Phase Induction Motor Using PIC18 Microcontroller*. Microchip. Pp3.

[7]   Rakesh Parekh (2003). *Induction Motor Fundamental*. Microchip. Pp16.

[8]   Jin-Woo Jung (2005). *Sine PWM Inverter*. The Ohio State University.

[9]   Jianming Yao (2000). *Single Phase Induction Motor Adjustable Speed Control Using DSP and Microcontroller*. UW-Madison.

[10]   Brandon Upchurch (2005). *Motor Speed Sensing with PIC Microcontroller*. Olivet Nazarene University.

APPENDICES

APPENDIX A

FULL PROGRAM CODING

```c
#include <16F777.h>
#device ADC=10
#fuses HS,NOWDT,NOPROTECT
#use Delay(clock=20000000)
#include "C:\Documents and Settings\User\Desktop\FYP\Deli\pwx2\brum\Flexlcd2.c"

long sin_table[36]={512 601 687 768 841 904 955 993 1016 1023 1016
                    993 955 904 841 768 687 601 512 423 337 256
                    183 120 69 31 8 0 8 31 69 120 183 256 337 423};


void pwm_3phase(int16 delay1,int16 PR2,int control_mode,int16 rheostat_adc,int16
temp_adc)
{
   int8 sin_counter1, sin_counter2, sin_counter3, index1, index2, cycle;
   float scale_value;
   long new_sin_table[36];
   int new_control_mode,input_variation=0,on_off=0;
   int16 new_rheostat_adc, new_temp_adc;

   scale_value = PR2 * 0.0039;

   new_sin_table[0] = 512 * scale_value;

   if (PR2>63)
   cycle=25;
   else if (PR2<=63 && PR2>45)
   cycle=40;
   else
   cycle=55;

   while ((on_off==0) && (input_variation==0))
   {
   for(index2=0; index2<cycle; index2++)
   {
      for(sin_counter1=0; sin_counter1<=35; sin_counter1++)
{
         set_pwm1_duty(new_sin_table[sin_counter1]);
         sin_counter2 = (sin_counter1+24)%36;
         set_pwm2_duty(new_sin_table[sin_counter2]);
         sin_counter3 = (sin_counter1+12)%36;
         set_pwm3_duty(new_sin_table[sin_counter3]);

         index1 = sin_counter1 + 1;
         new_sin_table[index1] = sin_table[index1] * scale_value;
         delay_us(delay1);
      }
   }
   on_off = input(PIN_C6);
   new_control_mode = input(PIN_B7);
   if (new_control_mode == control_mode)
   {
      if (control_mode == 1)
      {
         set_adc_channel(1);
         delay_us(20);
         new_rheostat_adc = read_adc();
         if ((new_rheostat_adc >= (rheostat_adc + 39))||(new_rheostat_adc <=
(rheostat_adc - 39)))
         input_variation = 1;
         else
         input_variation = 0;
      }
      else
      {
         set_adc_channel(0);
```

53

```
            delay_us(20);
            new_temp_adc = read_adc();
            if ((new_temp_adc >= temp_adc + 2)||(new_temp_adc <= temp_adc - 2))
            input_variation = 1;
            else
            input_variation = 0;
        }
    }
    else
    input_variation = 1;
    }
}

int16 get_PR2(float temp)
{
    int16 PR2;
    float store;

    if(temp>100.00)
    {
        PR2=33;
    }
    else if (temp<0.00)
    {
        PR2=115;
    }
    else  //if 0<temp<100, use formula to calculate PR2
    {
        store = -0.82 * temp + 115.0;
        PR2 = (int)(store);
    }
    return (PR2);
}

int16 get_delay(int16 PR2)
{
    int16 delay1;

    delay1 = (PR2+1) * 16 - 140;
    return (delay1);
}

void get_temp(float temp, unsigned int16 temp_adc)
{
    lcd_putc("\fTemp Control");
    set_adc_channel(0);
    delay_us(20);
    temp_adc = read_adc();
    temp = temp_adc * 0.4888;
    lcd_gotoxy(1,2);
    printf(lcd_putc,"Temp:%.1f",temp);
}

void control_mode_PR2(int16 PR2, int16 rheostat_adc)
{
    float storePR2,storeRPM,sync_speed,display_speed;
    double RPM;

    lcd_putc("\fUser Control");
    set_adc_channel(1);
    delay_us(20);
    rheostat_adc = read_adc();

    sync_speed = 1.527 * rheostat_adc + 450.00;
    storePR2 = 52132.00 / sync_speed - 1.00;
    PR2 = (int16)(storePR2);
//   display_speed = sync_speed - 50.00;
    lcd_gotoxy(1,2);
    printf(lcd_putc,"Nsync:%.3e",sync_speed);
}
```

```
void main()
{
    int16 PR2, rheostat_adc, temp_adc;
    int16 delay1;
    float temp;
    int on_off, control_mode;

    setup_ccp1(CCP_PWM);                // Configure CCP1 as a PWM
    setup_ccp2(CCP_PWM);                // Configure CCP2 as a PWM
    setup_ccp3(CCP_PWM);                // Configure CCP3 as a PWM

    setup_adc(ADC_CLOCK_INTERNAL);      //Configure ADC hardware
    setup_adc_ports(ALL_ANALOG);        //pin A0-A7 set as ADC

    lcd_init();
    lcd_putc("\fWelcome to LTH\nTech"); //display welcoming message for 3second
    delay_ms(3000);
    set_tris_c(0x11000000);

    while(true)
    {
//      output_high(PIN_B0);
        on_off=input(PIN_C6);
        if(on_off==0)
        {
            control_mode=input(PIN_B7);
            if(control_mode==0)
            {
            get_temp(temp, temp_adc);
            PR2 = get_PR2(temp);
            }
            else
            {
            control_mode_PR2(PR2,rheostat_adc);     //get PR2 based on rheostat
            }
            delay1 = get_delay(PR2);
            setup_timer_2(T2_DIV_BY_16, PR2, 1);
            pwm_3phase(delay1,PR2,control_mode,rheostat_adc,temp_adc);              }
        else                            //system off
        {
        set_pwm1_duty(0);               //set duty cycle of PWMs to zero
        set_pwm2_duty(0);
        set_pwm3_duty(0);

        while(on_off==1)
        {
            on_off=input(PIN_C6);       //check on/off switch
            lcd_putc("\fOFF");
            delay_us(1000);
        }
        }
    }
}
//*/
```

55