SPEED CONTROL OF THREE-PHASE INDUCTION MOTOR

PRESENTED BY:
GOKUL PRASANTH [PTA19EE008]
ISEN JACOB GEORGE [PTA19EE009]
RAKESH HARI [PTA19EE010]
SACHIN THOMAS [PTA19EE012]
COLLEGE OF ENGINEERING KALLOOPPARA

GUIDED BY:
MR. DEEPU V.S
ASST. PROFESSOR &
MR. JAISON JAMES
HEAD OF DEPARTMENT
COLLEGE OF ENGINEERING KALLOOPPARA.

CONTENTS

INTRODUCTION

- Induction motors are widely used in various industries for their reliability, durability, and low maintenance requirements.
- Precise control over motor speed is essential to ensure process stability, improve efficiency, and maintain safe operation.
- Traditional methods of speed control may not be sufficient for meeting the requirements of modern industrial processes.
- V/F control is a popular method for speed control of induction motors, which involves adjusting the voltage and frequency supplied to the motor to control its speed.
- V/F control is a cost-effective and efficient way to achieve precise control over motor speed, and it can be customized to meet the specific requirements of different applications.
- V/F control is widely used in various industrial applications, such as conveyor systems, pumps, fans, and more.
- V/F control can help to reduce energy consumption, extend motor lifespan, and improve overall process efficiency.

V/F Speed control- Torque-Slip Chara

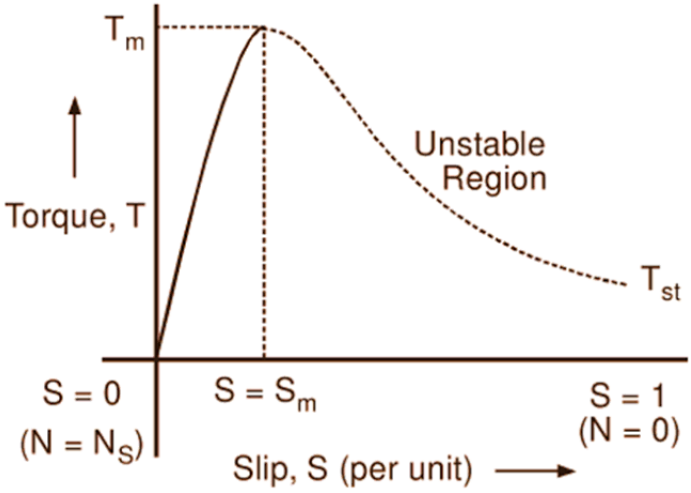


Figure: 1.Torque-Slip Characteristics.

Torque-Slip Characteristics Contd..

- Torque slip characteristics refers to the relationship between the torque output of an AC motor and its slip speed.
- At low slip speeds, the torque output of the motor is high, and it decreases as the slip speed increases.
- The maximum torque output of the motor occurs at the point where the torque slip curve intersects with the line of maximum power.
- As the load on the motor increases, the slip speed also increases, which causes the torque output to decrease.
- The torque slip characteristics of a motor are important for determining its performance in various applications, such as pumps, fans, and conveyor systems.

V/F Speed control-Graph

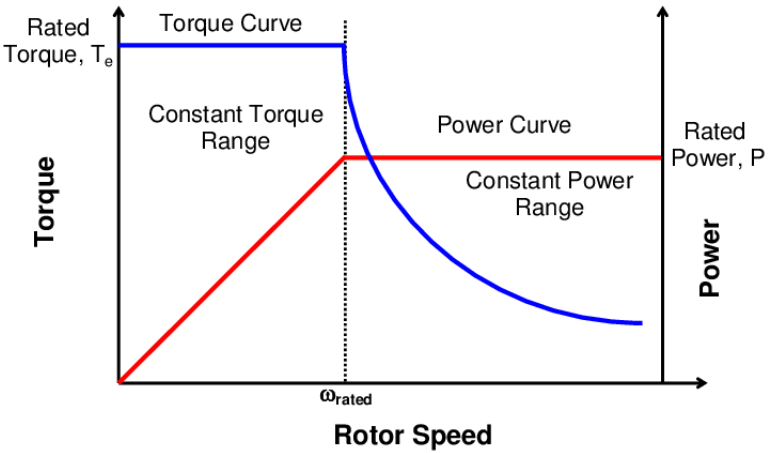


Figure: 2. V/F control - Graph

V/F Speed control - Graph Contd..

- The V/f (voltage-to-frequency) graph of a three-phase induction motor represents the relationship between the motor's voltage and frequency.
- The V/f graph is a straight line with a positive slope. The slope is proportional up-to the ratio of the motor's rated voltage to its rated frequency
- The V/f graph has a maximum frequency limit beyond which the motor's operation becomes unstable. This limit is determined by the motor's construction and design, and it is typically around 2-2.5 times the rated frequency
- The V/f graph also has a minimum frequency limit below which the motor's torque production decreases rapidly. It is typically around 30-40% of the rated frequency.
- The V/f graph can be used to determine the optimal voltage and frequency settings for the motor under different operating conditions.

Block diagram of VFD

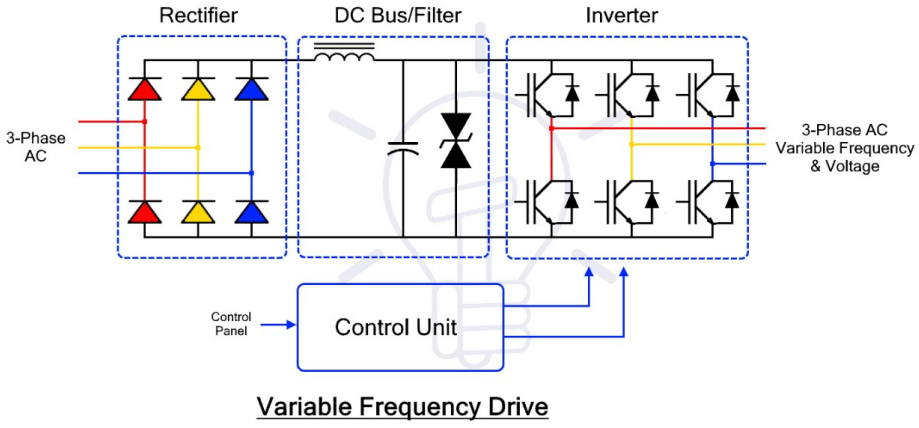


Figure: 3. Block diagram of VFD drive

Block diagram of VFD Contd..

- A VFD (Variable Frequency Drive) is an electronic device used to control the speed and torque of an AC motor by varying the frequency and voltage supplied to it.
- The block diagram of a VFD typically consists of four main components: rectifier, DC bus, inverter, and control circuitry.
- The rectifier converts the incoming AC power to DC power, which is then filtered and smoothed to provide a stable DC bus voltage.
- The inverter converts the DC power back to AC power at variable frequency and voltage, which is then supplied to the motor.
- The control circuitry includes microprocessors and other electronic components that monitor and adjust the output frequency and voltage of the VFD based on various input signals such as speed commands and feedback from the motor.

Simulation Of VFD

- Simulations are done in MATLAB Software(2019 Version)
- In simulation Squirrel-Cage induction motor is used with the following specifications:
  - 5.4HP(4kw),400V,1430RPM
  - Stater Resistance and inductance:1.405ohm and 0.005839H.
  - Rotor resistance and inductance:1.395ohm and 0.005839H.
  - Mutual Inductance: 0.1722H
  - Frequency=50Hz,Poles=2

Simulation Block Diagram

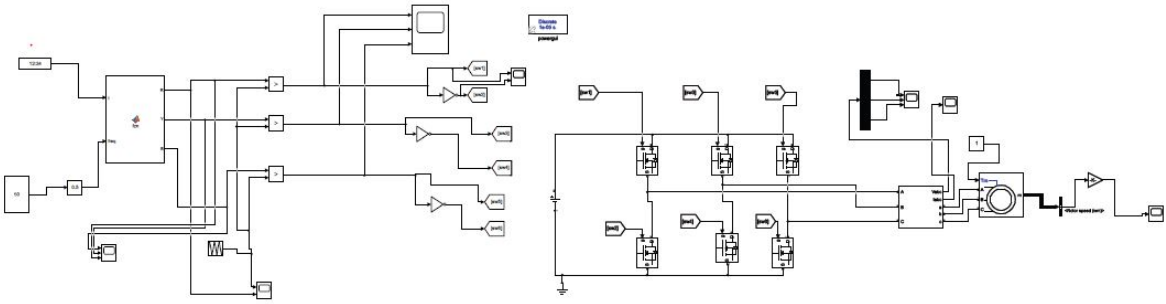


Figure: 4. Simulation block of VFD drive

Simulation Block Diagram contd..

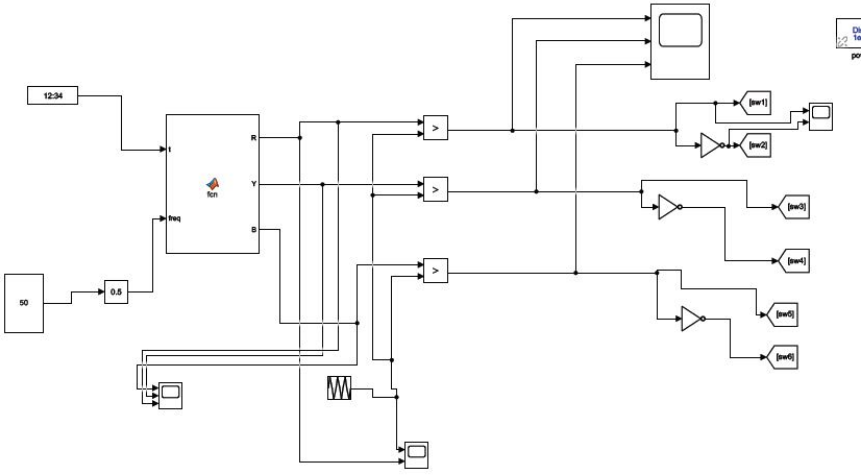


Figure: 5. Pulse Generator Block for the Inverter circuit

Simulation Block Diagram contd..

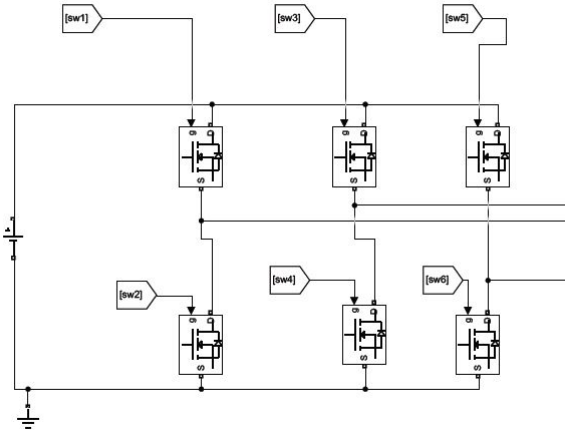


Figure: 6. Inverter circuit

Simulation Block Diagram contd..

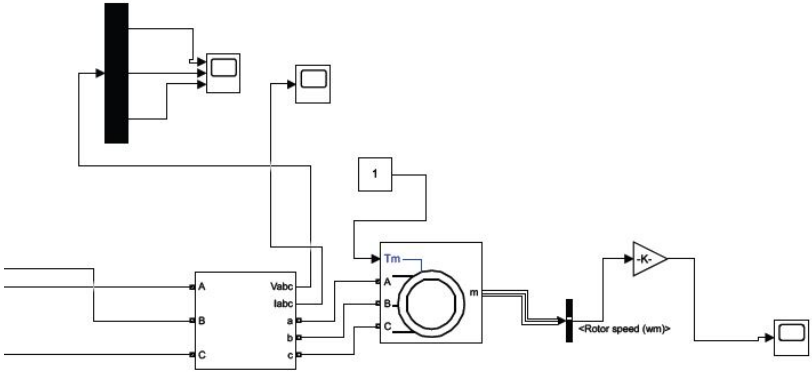


Figure: 7. Three-Phase Induction motor side

Simulation Results

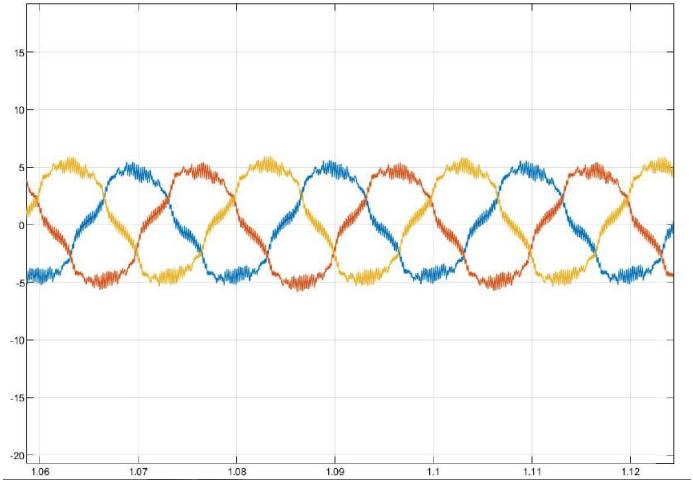


Figure: 8. Three-Phase Inverter Output Currents

Simulation Results

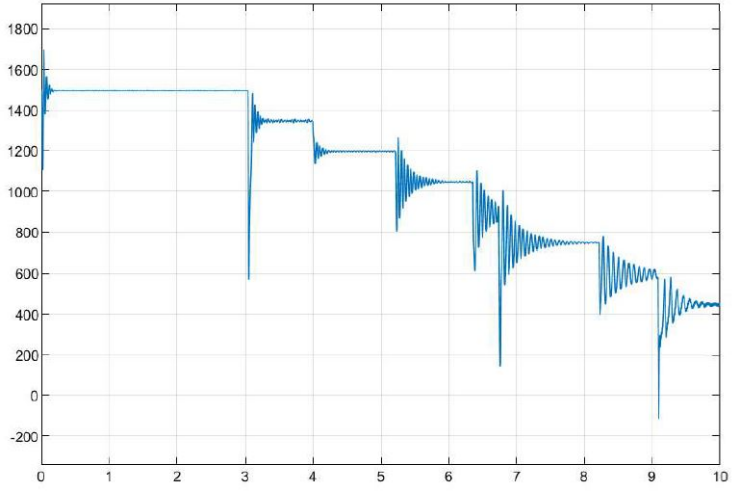


Figure: 9. Variation of speed under different frequencies

Simulation Results Contd..

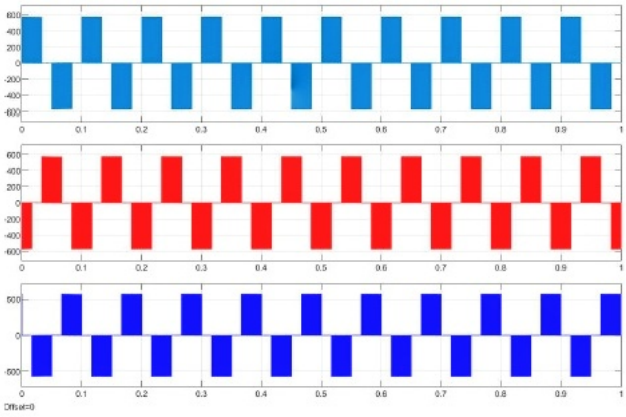


Figure: 10. Inverter Output Voltage

Hardware Implementation - Schematic Representation

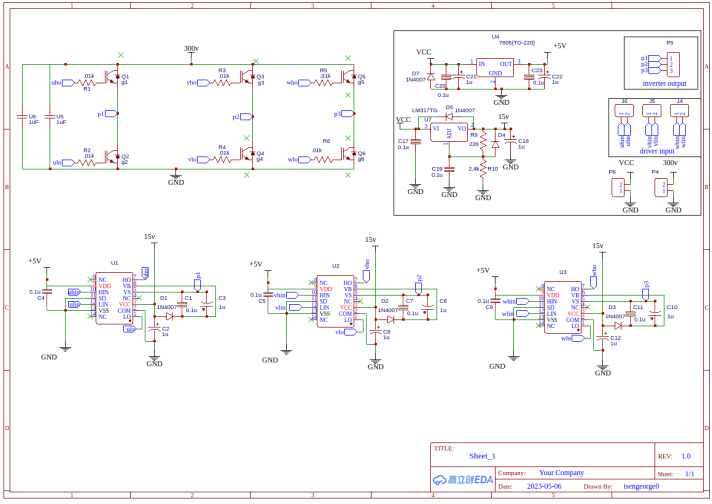


Figure: 11. Schematic Representation-Inverter and Drivers

Hardware Implementation - Schematic Representation

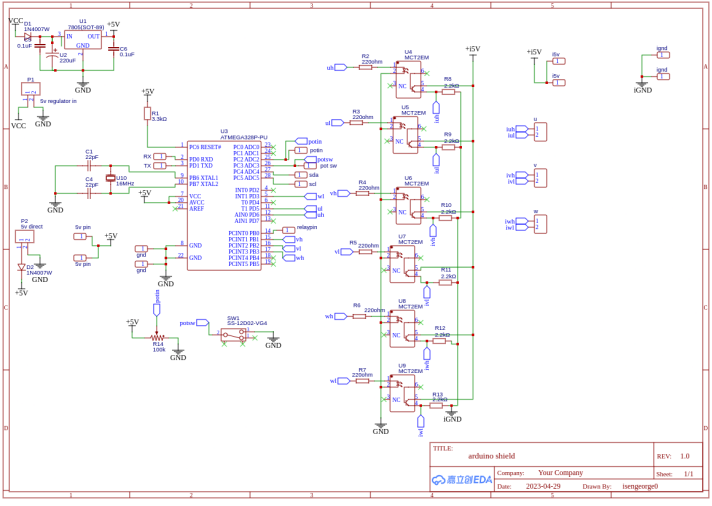


Figure: 12. Schematic Representation-Control Units

Hardware Implementation - Major Components

* Rectifier

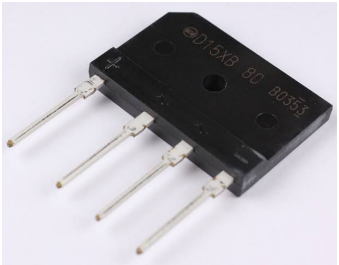


Figure: Rectifier Component

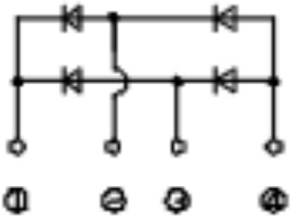


Figure: Internal Diagram

Rectifier-D15XB80
-800V,15A Rectifier

Hardware Implementation - Major Components

* Inverter

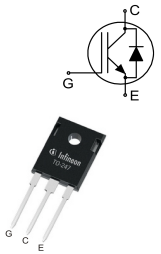


Figure: IKW40N60 IGBT

Specifications
- 600V,40A
- Vge=+/-20V
- VCEsat=1.85V

Hardware Implementation - Major Components

* Gate Driver for IGBT

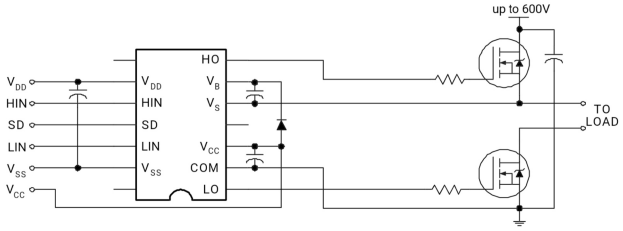


Figure: Typical Circuit

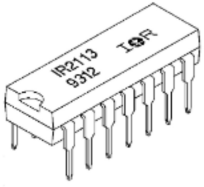


Figure: IR2113 Gate Driver

- Voffset-600V max
- Io - +/-2A
- Vout - 10-20V

Hardware Implementation - Major Components

* Isolation -Optocoupler

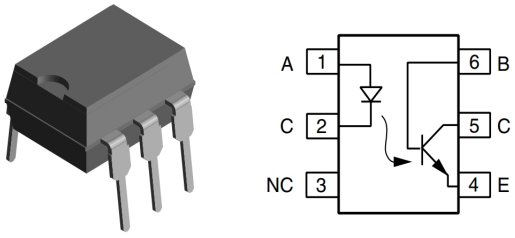


Figure: MCT2E Optocoupler

Hardware Implementation - Major Components

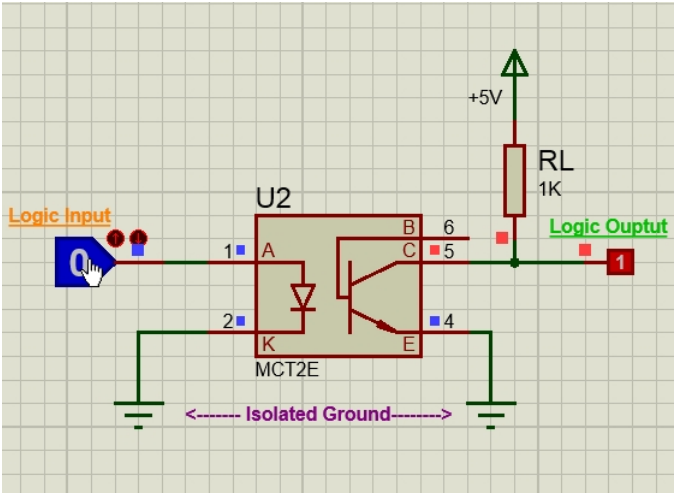


Figure: MCT2E Optocoupler-Working

Hardware Implementation - Major Components

* ATmega328p

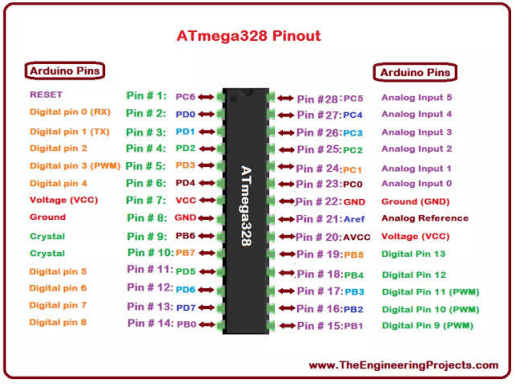
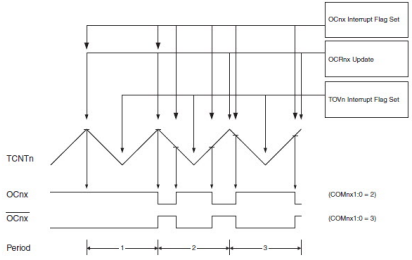


Figure: ATmega328p

Hardware Implementation - Major Components

* Phase Correct PWM



$$DutyCycle_{normal} = \frac{OCR0A}{255} \cdot 100$$

$$DutyCycle_{inverted} = \frac{255 - OCR0A}{255} \cdot 100$$

$$Output\ Frequency = \frac{Fclk_{I/O}}{2 * N * TOP(ICR1)}$$

$$TOP(ICR1) = \frac{Fclk_{I/O}}{2 * N * Output\ Frequency}$$

$$TOP(ICR1) = \frac{16Mhz}{2 * 8 * 100Hz}$$

$$TOP(ICR1) = 10000$$

Hardware Implementation - Major Components

Atmega328p Specifications:

Architecture: 8-bit AVR
Operating Voltage: 1.8 - 5.5V
Flash Memory: 32KB
EEPROM: 1KB
SRAM: 2KB
Clock Speed: 20MHz maximum
GPIO pins: 23
Analog Input Pins: 6
PWM Channels: 6
10-bit ADC (Analog to Digital Converter)
Power Consumption: Active mode: 1.5mA at 1MHz, 5V; Idle mode: 0.5mA at 1MHz, 5V Two 8-bit timers and one 16-bit timer/counters with PWM capability

Hardwares

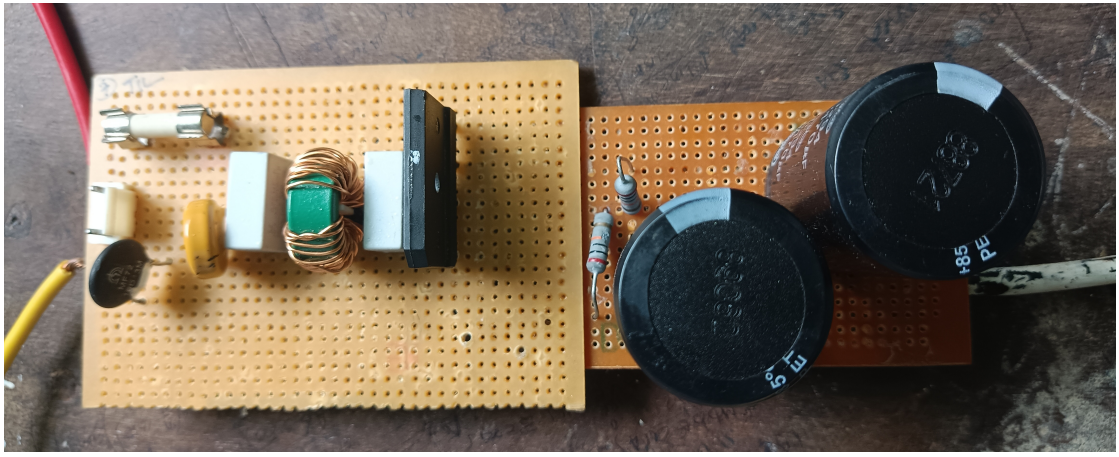


Figure: Rectifier

Hardwares

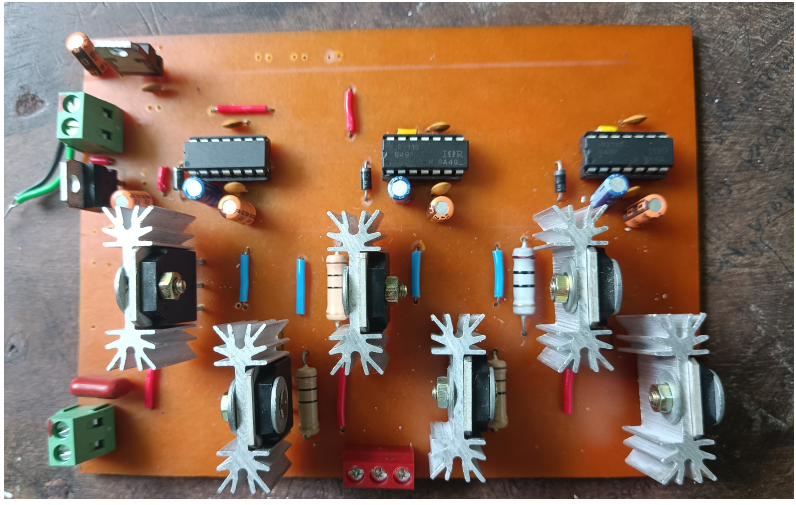


Figure: Inverter/Driver

Hardwares

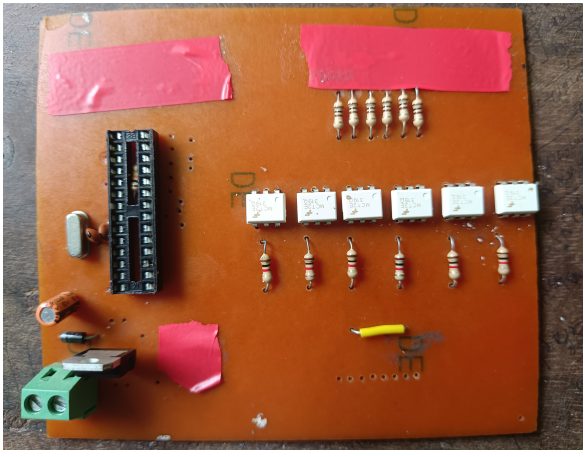
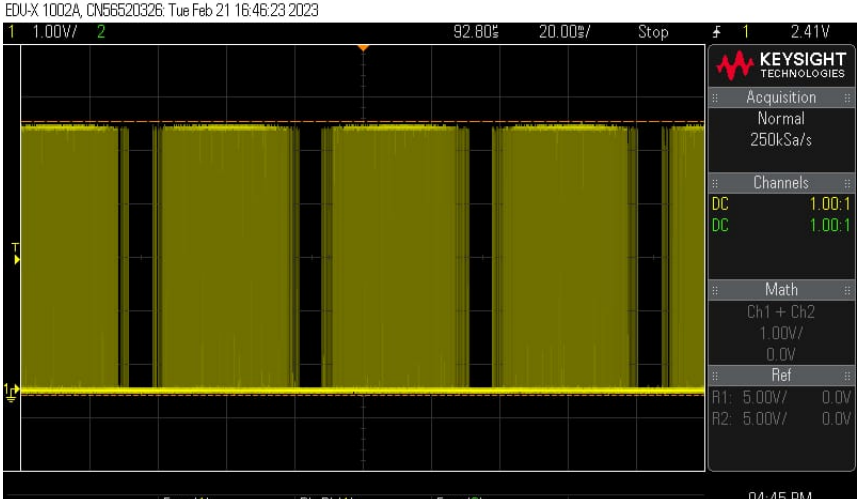


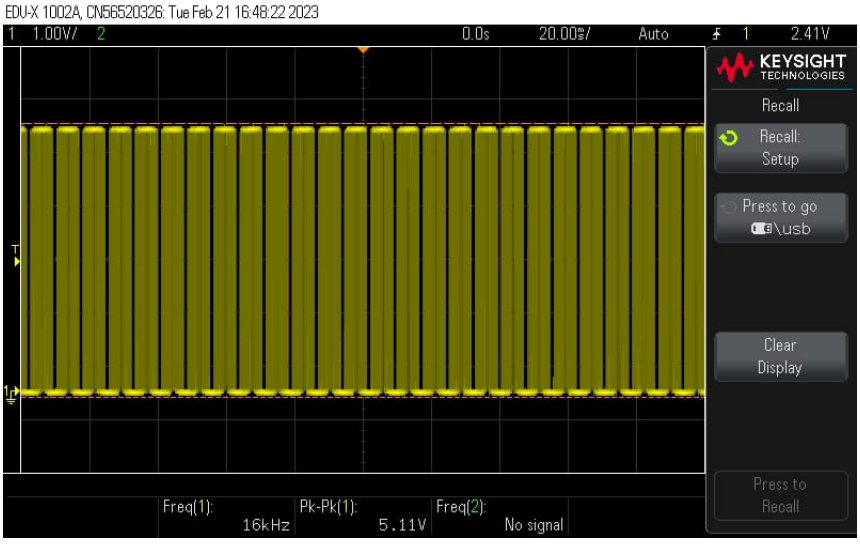
Figure: Controller/Isolation

Hardware Implementation - Outputs[Pulse waveforms]

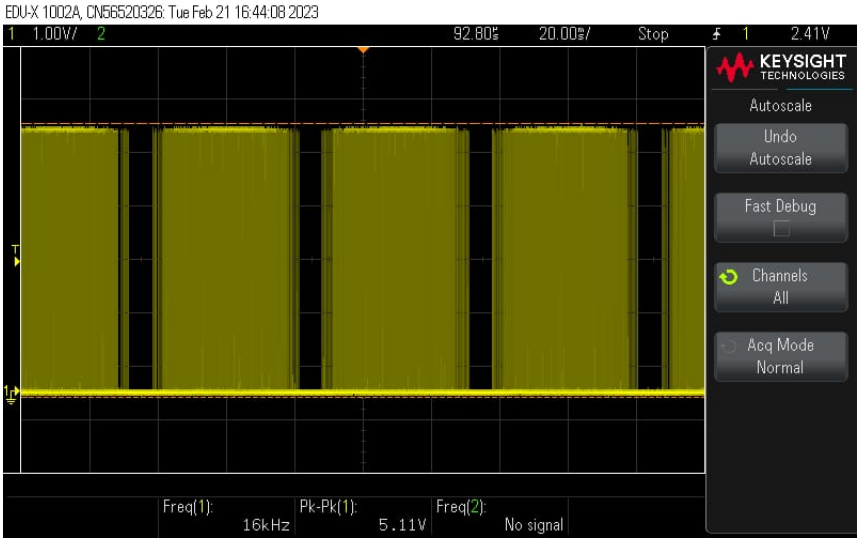
The figures given below shows the variation in pulses due to changes in frequency done by varying the 40kOhm pot.



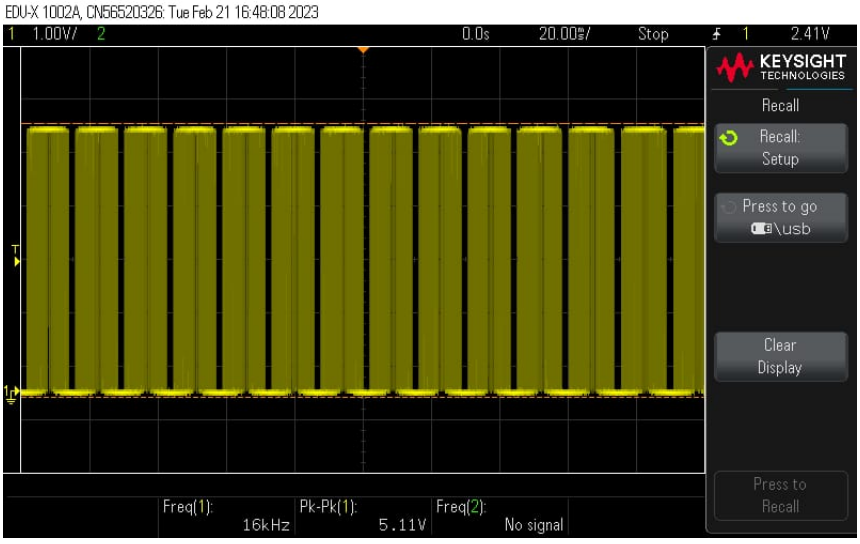
Pulse waveform contd..



Pulse waveform contd..



Pulse waveform contd..



Output Voltage

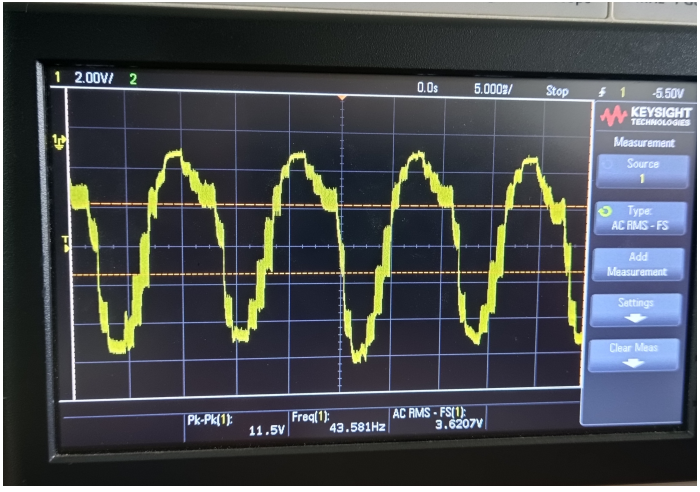


Figure: Output Voltage

Conclusions

- We were able to control the speed of an Three-phase Induction motor.
- Outputs were verified in Rectifier,Inverter And Gate driver Circuits.
- Isolation and protective measures were also taken under consideration during designing and implementation.
- This drive is capable of driving any three phase induction motor under 3HP.
- Improvements
  - IOT based controller(ESP8266 MODULE)
  - Overcurrent Protection
  - Using 16/32 bit Microcontroller

References

[1]. Bharti, R., Kumar, M., Prasad, B. . (2019). V/F Control of Three Phase Induction Motor. 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN). doi:10.1109/vitecon.2019.8899420

[2]. Harsha, A. R., Pranupa, S., Kiran Kumar, B. M., Nagaraja Rao, S., Indira, M. S. (2020). Arduino based V/f Drive for a Three Phase Induction Motor using Single Phase Supply. 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE).doi:10.1109/icstcee49637.2020.9277093

[3]. P.Enjeti, A. Rahman, and R. jakkli,"economic single phase to three phase converter topologies for fixed and variable frequency output," IEEE Trans. Power E"lectron. Vol. 8 no. 3 pp.88-94 Jul.

[4]. P. V. Patil and S. A. Naveed, "Implementation of VFD Application for Speed Control of Induction Motor," 2020 International Conference on Smart Innovations, 2020, pp. 168-170, doi:10.1109/ICSIDEMPC49020.2020.9299636.

[5]. A. Z. Latt and N. N. Win, "Variable Speed Drive of Single Phase Induction Motor Using Frequency Control Method," 2009 International Conference on Education Technology and Computer, 2009, pp. 30-34, doi: 10.1109/ICETC.2009.72.

THANK YOU

APPENDIX-ARDUINO CODE

```
{
  PWM_Running = PWM_NOT_RUNNING;
  cli();
  Init_PWM_Counter = 0;
  TCCR0A = 0;
  TCCR0B = 0;
  TCCR1A = 0;
  TCCR1B = 0;
  TCCR2A = 0;
  TCCR2B = 0;
  sei();
}

void Pwm_Config()
{
  //***Check in scope. Need to make sure the pins are LOW prior to and after setting them to
outputs so don't accidentally cause short in IPM.
  if (Phase_Config == THREE_PH)
  {
     cli();                    //Disable interrupts
     PWM_Running = PWM_RUNNING;
```

```
    GTCCR = (1<<TSM)|(1<<PSRASY)|(1<<PSRSYNC); //Halt all timers
    //Timer 0
    TCCR0A = (1 << COM0A1) | (1 << COM0B1) | (1 << COM0B0) | (1 << WGM00); //
Clear OC0A and set OC0B counting up. Waveform mode 1 (Table 14-8)
    TCCR0B = (1 << CS00);       //No prescaler
    TIMSK0 = (1 << TOIE0);      //Timer/Counter0 Overflow Interrupt Enable
    OCR0A = 0;     //Sign determined by set or clear at count-up. High-side IGBT OFF.
    OCR0B = 127;   //Sign determined by set or clear at count-up. Low-side IGBT 50% duty
cycle to charge bootstrap cap.
    // Timer 1
    TCCR1A = (1 << COM1A1) | (1 << COM1B1) | (1 << COM1B0) | (1 << WGM10); //
Clear OC1A and set OC1B counting up. Waveform mode 1 (Table 14-8)
    TCCR1B = (1 << CS10);       //No prescaler
    OCR1A = 0;     //Sign determined by set or clear at count-up. High-side IGBT OFF.
    OCR1B = 127;   //Sign determined by set or clear at count-up. Low-side IGBT 50% duty
cycle to charge bootstrap cap.
    // Timer 2
    TCCR2A = (1 << COM2A1) | (1 << COM2B1) | (1 << COM2B0) | (1 << WGM20); //
Clear OC0A and set OC0B counting up. Waveform mode 1 (Table 14-8)
    TCCR2B = (1 << CS20);       //No prescaler
    OCR2A = 0;     //Sign determined by set or clear at count-up. High-side IGBT OFF.
    OCR2B = 127;   //Sign determined by set or clear at count-up. Low-side IGBT 50% duty
cycle to charge bootstrap cap.
```

```
    TCNT0 = 0;     //Set timer0 to 0
    TCNT1H = 0;    //Set timer1 high byte to 0
    TCNT1L = 0;    //Set timer1 low byte to 0
    TCNT2 = 0;     //Set timer2 to 0
    GTCCR = 0;     //Release all timers
    Init_PWM_Counter = 0;
    Wait_A_Bit(BOOT_CAP_CHARGE_TIME);
    sei();
  }
  else if (Phase_Config == ONE_PH)
  {
    PORTB &= 0xF5;                          //Clear the 2nd (OC1A) and 4th (OC2A) bits. Set
High-Side transistors LOW.
    PORTB |= (1 << PORTB2);                 //Set the 3rd (OC1B) bit. Set Low-Side
transistor HIGH. Should be connected to GND.
    PORTD |= (1 << PORTD3);                 //Set the 4th bit (OC2B). Set Low-Side
transistor HIGH. Should be connected to GND.
    cli();                  //Disable interrupts
    PWM_Running = PWM_RUNNING;
```

```
    GTCCR = (1<<TSM)|(1<<PSRASY)|(1<<PSRSYNC); //Halt all timers
    //Timer 0
    TCCR0A = (1 << COM0A1) | (1 << COM0B1) | (1 << COM0B0) | (1 << WGM00); //
Clear OC0A and set OC0B counting up. Waveform mode 1 (Table 14-8)
    TCCR0B = (1 << CS00);       //No prescaler
    TIMSK0 = (1 << TOIE0);      //Timer/Counter0 Overflow Interrupt Enable
    OCR0A = 0;     //Sign determined by set or clear at count-up. High-side IGBT OFF.
    OCR0B = 127;   //Sign determined by set or clear at count-up. Low-side IGBT 50% duty
cycle to charge bootstrap cap.
    // Timer 1 - Disabled
    TCCR1A = 0;
    TCCR1B = 0;
    OCR1A = 0;     //Sign determined by set or clear at count-up. High-side IGBT OFF.
    OCR1B = 127;   //Sign determined by set or clear at count-up. Low-side IGBT 50% duty
cycle to charge bootstrap cap.
    // Timer 2 - Disabled
    TCCR2A = 0;
    TCCR2B = 0;
    OCR2A = 0;     //Sign determined by set or clear at count-up. High-side IGBT OFF.
    OCR2B = 127;   //Sign determined by set or clear at count-up. Low-side IGBT 50% duty
cycle to charge bootstrap cap.
```

```
    TCNT0 = 0;     //Set timer0 to 0
    TCNT1H = 0;    //Set timer1 high byte to 0
    TCNT1L = 0;    //Set timer1 low byte to 0
    TCNT2 = 0;     //Set timer2 to 0
```

```
    GTCCR = 0;     //Release all timers
    Init_PWM_Counter = 0;
    Wait_A_Bit(BOOT_CAP_CHARGE_TIME);
    sei();
  }
}
```

```
ISR (TIMER0_OVF_vect)
{
  OVF_Counter++;
  if (OVF_Counter >= OVF_Counter_Compare)
  {
    if (Sine_Index == Sine_Len) Sine_Index = 0;
    if (Sine_Index_120 == Sine_Len) Sine_Index_120 = 0;
    if (Sine_Index_240 == Sine_Len) Sine_Index_240 = 0;
    //
    if ((Sine_Used[Sine_Index] - DEADTIME_SUB) < MIN_PWM_VAL)
    {
      OCR0A = 0;
    }
    else
    {
      OCR0A = uint8_t(Sine_Used[Sine_Index] - DEADTIME_SUB);
    }
    OCR0B = OCR0A + DEADTIME_ADD;

    if ((Sine_Used[Sine_Index_120] - DEADTIME_SUB) < MIN_PWM_VAL)
    {
      OCR1A = 0;
    }
```

```
else
{
  OCR1A = uint8_t(Sine_Used[Sine_Index_120] - DEADTIME_SUB);
}
OCR1B = OCR1A + DEADTIME_ADD;

if ((Sine_Used[Sine_Index_240] - DEADTIME_SUB) < MIN_PWM_VAL)
{
  OCR2A = 0;
}
else
{
  OCR2A = uint8_t(Sine_Used[Sine_Index_240] - DEADTIME_SUB);
}
OCR2B = OCR2A + DEADTIME_ADD;
```

```
    OVF_Counter = 0;
    Sine_Index++;
    Sine_Index_120++;
    Sine_Index_240++;
  }
}
```