Faculty of Technical Sciences in Čačak
University of Kragujevac

Scalar Control of Three-Phase Induction Motor Characteristics

Bachelor Thesis

Aleksandar Brakocevic

1220/17

Thesis Supervisor

PhD Milan Dobricic

SUMMARY

Alternating current (AC) motors are usually controlled by variable frequency drives (VFD), which regulate motor speed by adjusting the frequency of the supply voltage. Depending on the application and the required level of speed regulation, VFDs can be controlled using scalar or vector methods. The most common type of VFD control is scalar, which applies the volts-per-hertz (V/Hz) or volts-per-frequency (V/F) method.

The terms *variable frequency drive (VFD)* and *variable speed drive (VSD)* are often used interchangeably, but there is a difference between the two. A variable speed drive (VSD) is any drive that can control the speed of a piece of equipment, including both AC and DC motors. VSDs may operate through mechanical, hydraulic, or electrical drives. A variable frequency drive (VFD), on the other hand, is specifically used to control the speed of AC motors by varying the frequency of the motor supply voltage.

How does V/Hz control work?

AC motors are designed to operate with a constant magnetic field (flux). The strength of this magnetic field is proportional to the ratio of voltage (V) to frequency (Hz), or V/Hz. The VFD controls the motor speed by varying the frequency of the applied voltage, according to the synchronous speed equation:

$$N_s = \frac{120 * f}{p}$$

where:

- ns= synchronous speed (in revolutions per minute, rpm)
- f = supply frequency (in hertz, Hz)
- P = number of motor poles

Changing the frequency of the voltage affects both the motor speed and the strength of the magnetic field. When the frequency is reduced (for lower motor speeds), the magnetic field increases, creating excessive heat. When the frequency is increased (for higher motor speeds), the magnetic field decreases, resulting in lower torque. To maintain a constant magnetic flux, the V/Hz ratio must remain constant. The V/Hz method keeps torque generation stable regardless of frequency. V/Hz control of a VFD avoids changes in the strength of the magnetic field by adjusting the voltage together with the frequency so that the V/Hz ratio remains constant.

The appropriate V/Hz ratio is determined by the rated voltage and frequency of the motor. For example, a motor rated for delta connection at 230 V and 50 Hz will always operate best with a V/Hz ratio of 4.6 (230/50 = 4.6).

Open-loop (V/Hz) control does not use feedback and only adjusts the voltage and frequency of the motor based on an external speed command.

Closed-loop V/Hz control can incorporate encoder feedback to measure the actual motor speed. An error signal is generated based on the difference between the actual speed and the reference speed, and the controller produces a new frequency command to compensate for the error. Although closed-loop V/Hz improves speed regulation, it is not commonly used because of the additional cost and complexity of the feedback hardware.

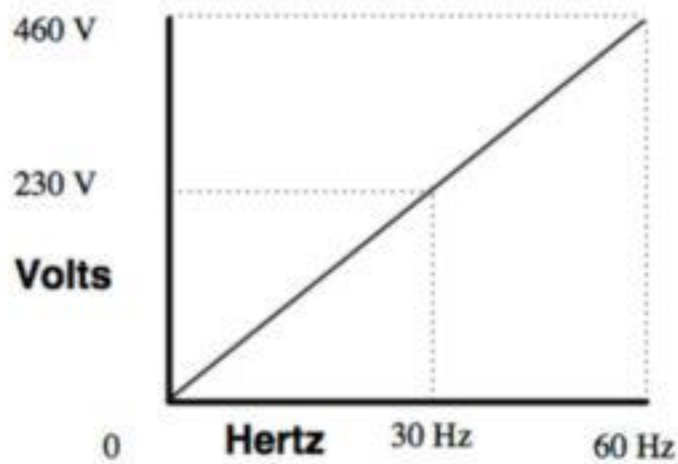


Figure 1.
V/Hz regulation maintains a constant ratio between voltage (V) and frequency (Hz).

V/Hz control is a simple and low-cost method for operating variable frequency drives and is generally considered the most common VFD control scheme. It is suitable for both constant-torque and variable-torque applications and can provide up to 150 percent of rated torque at zero speed for starting and peak loads. Speed regulation is typically within 2 to 3 percent of the maximum rated frequency, making this method unsuitable for applications where precise speed control is critical.

The most common use of V/Hz control is for driving industrial equipment such as fans and compressors. A unique advantage of V/Hz control over other methods is that it allows multiple motors to be operated from a single VFD. All motors will start and stop simultaneously and run at the same speed, which is beneficial in some process applications such as heating and cooling.

Contents:

1. PREFACE

The final thesis represents a seal on the previously acquired knowledge that was selflessly imparted to us by our professors during lectures. Likewise, this work I have chosen to pursue is an attempt to leave a positive impression on the professors and to justify the time they dedicated to us, striving to bring us closer to the subject matter and prepare us for future endeavors.
I have spent a great deal of time working on this thesis, but that time is nothing compared to the knowledge I have gained, both theoretical and especially practical, during its realization. Each stage of the work brought new positive experiences as well as disappointments, particularly when things did not go the way I had hoped. As the saying goes: *"Electronic components run on white smoke—when the smoke escapes, the component no longer works."* I too have witnessed plenty of white smoke, but perseverance paid off, and I successfully completed the project. The feeling and satisfaction of having accomplished something independently and with full understanding is unparalleled.
I would like to express my deep gratitude to Professor Dr. Milan Dobričić, Professor Dr. Dragan Brajović, and Professor Dr. Božimir Mišković.

Aleksandar Brakočević
September 2021

1.1 INTRODUCTION

Induction motors are the most commonly used motors. They are robust, reliable, and durable. When voltage is applied to an induction motor according to the specifications on its nameplate, the motor runs at its rated speed. However, in many applications, variable speed operation is required. For example, a washing machine may use different speeds for each wash cycle.

Historically, mechanical gear and belt systems were used to achieve variable speeds. Thanks to the rapid advancement of electronics, power and control systems have matured to the point where these components are now used to control motors instead of mechanical gears. Electronics not only control motor speed but can also improve the dynamic performance of the motor. Additionally, electronics can reduce the system's average power consumption and lower motor noise.

Controlling an induction motor is complex because of its nonlinear characteristics. Although different control methods exist, the **variable voltage variable frequency (VV/VF)** or **V/F** method is the most common speed control method in open-loop systems. This method is most suitable for applications that do not require position control or high-precision speed regulation. Examples include heating, air conditioning, fans, and compressors.

V/F control can be implemented using inexpensive microcontrollers instead of costly processors and digital signal processors (DSP). Many microcontrollers have built-in hardware PWM (pulse width modulation) signal generators. The **Arduino Mega** development board is equipped with an ATmega2560 microcontroller, which contains six timers and fifteen PWM pins that can be configured in the corresponding registers for different signal values. In this work, we use three PWM pins to generate three SPWM (sine pulse width modulation) signals phase-shifted by 120°.

This thesis covers the basics of scalar control of three-phase induction motors.

1.2 Nameplate

A typical nameplate (Figure 1) of an induction motor contains the following parameters:

- Rated supply voltage at the terminals in volts (V)
- Supply frequency in hertz (Hz)
- Rated current in amperes (A)
- Base speed in revolutions per minute (RPM)
- Power in watts or horsepower (kW–HP)
- Rated torque in newton-meters (Nm)
- Slip speed in RPM, or slip frequency in hertz (Hz)
- Type of winding insulation – class A, B, F, or H
- Efficiency
- Type of stator connection (for three-phase motors only), star (Y) or delta (Δ)

KONČAR	made in Croatia		
Code 276684	Nr 528011		
3 ≈	Type 5AZ 112M4	B3	
Δ / Y	230 / 400 V	5 / 2.9 A	
2,2 kW	cosφ=0,82		
50 Hz	1420 min^{-1}		
t_o °C	Isol.F	IP 54	S1
IEC34-1VDE0530			

Figure 1. Motor Nameplate

1.3 Operating Principle of the Induction Motor

When rated alternating current supply is applied to the stator windings, a magnetic flux of constant magnitude is generated, rotating at synchronous speed. The rotating field moves in the air gap and closes through the stator and rotor, which induces corresponding electromotive forces in the conductors. In the stator winding, a counter electromotive force EsE is generated, which balances the applied stator voltage U. An electromotive force is also induced in the rotor winding. If the rotor circuit is closed, a current Ir will flow, whose active component has the same direction as the induced electromotive force. Since the conductor carrying the current IrI is located in a magnetic field, an electromagnetic force F acts on it, causing the rotor to turn in the direction of the rotating magnetic field. The sum of all the forces acting on the rotor conductors represents the electromagnetic torque of the motor.
As energy is transferred from the stator to the rotor by electromagnetic induction, asynchronous machines are often called induction machines. The windings are of resistive-inductive nature. Reactive energy is required to magnetize the magnetic material and the air gap between the stator and rotor. Since the asynchronous machine cannot generate reactive energy, it must draw it from the network. The current driven through the windings by the supply voltage will always be inductive. Therefore, the asynchronous machine, both in motor and generator mode, consumes reactive energy, which is one of the main reasons why it is used primarily as a motor. In generator mode, the asynchronous machine is used within autonomous power systems, where the reactive energy is supplied by a capacitor bank. In large industrial consumers with many high-power asynchronous motors, static compensators (usually capacitor banks) are often installed to improve the power factor so that reactive energy is not drawn from the network.
To reduce the relative speed, the rotor begins to rotate in the same direction as the flux and attempts to catch up. But in practice, the rotor can never "reach" the stator field. Thus, the rotor always runs slower than the speed of the stator field. This difference in speed is called slip speed, and it depends on the mechanical load on the motor shaft.
The frequency and speed of the motor, with respect to the input supply, are referred to as the synchronous frequency and synchronous speed. The synchronous speed is directly proportional to the ratio of the supply frequency to the number of poles in the motor. The synchronous speed of the induction motor is shown in Equation (1.1).

synchronous speed $N_s = \frac{120 * f}{p}$
f – rated supply frequency **(1.1)**
p – number of poles

Synchronous speed is the speed at which the stator flux rotates. The rotor flux rotates slower than the stator flux. This difference is called slip speed. The speed indicated on the motor nameplate is the base speed. Some manufacturers specify it as a percentage of synchronous speed, as shown in Equation (1.2):

$$Nk\% = \frac{(Ns - Nn) * 100}{Ns} \quad \textbf{(1.2)}$$

where:

- Nn_= base speed
- Ns = synchronous speed
- Nk = slip speed

The slip percentage depends on the motor load: the higher the load, the greater the slip.

.

1.4 Types of asynchronous motors

Based on rotor construction, induction motors are classified into two categories:

- squirrel-cage motors
- slip-ring motors

The stator construction is the same for both types.

1.5 Squirrel-cage rotor motor

Almost 90% of asynchronous motors are squirrel-cage motors. This is because the squirrel-cage motor has a simple and robust construction. The rotor consists of a cylindrical laminated core with axially placed parallel slots to hold the conductors. Each slot is filled with copper, aluminum, or an alloy. The rotor bars are short-circuited by end rings on both sides, as shown in Figure 2. The rotor slots are not parallel to the shaft; they are skewed for two main reasons:

a) To ensure quiet motor operation by reducing magnetic noise.
b) To help reduce the tendency of the rotor to lock..

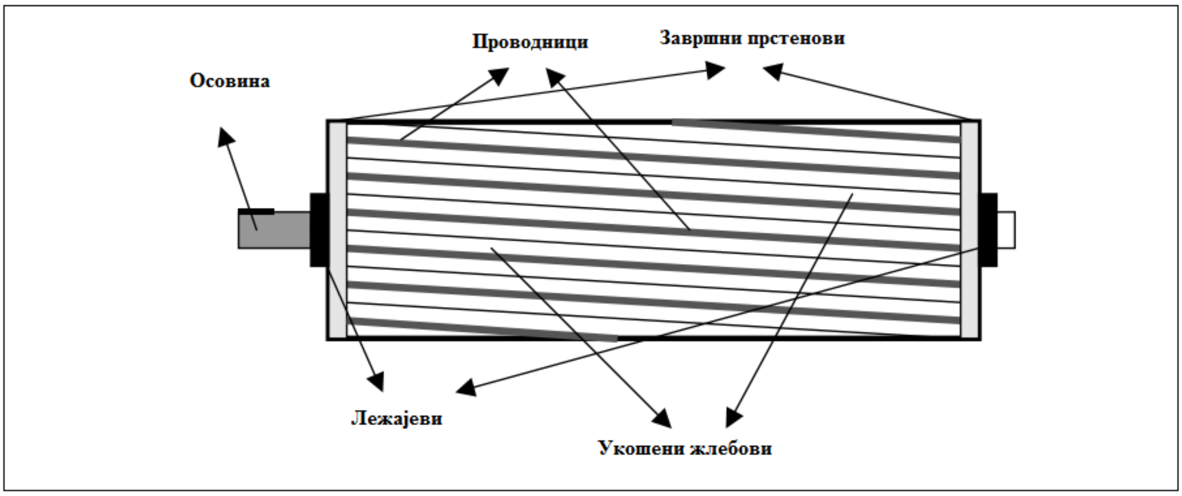


Figure 2. View of a squirrel-cage rotor

1.6 Slip-ring motor

The windings on the rotor are connected to three insulated slip rings on which brushes are mounted (Figure 3). This allows an external resistor to be introduced into the rotor winding. The external resistor can be used to increase the motor torque and modify the torque characteristics. Under normal operating conditions, the slip rings are short-circuited by an external switch. In normal conditions, the slip-ring motor operates like a squirrel-cage motor.

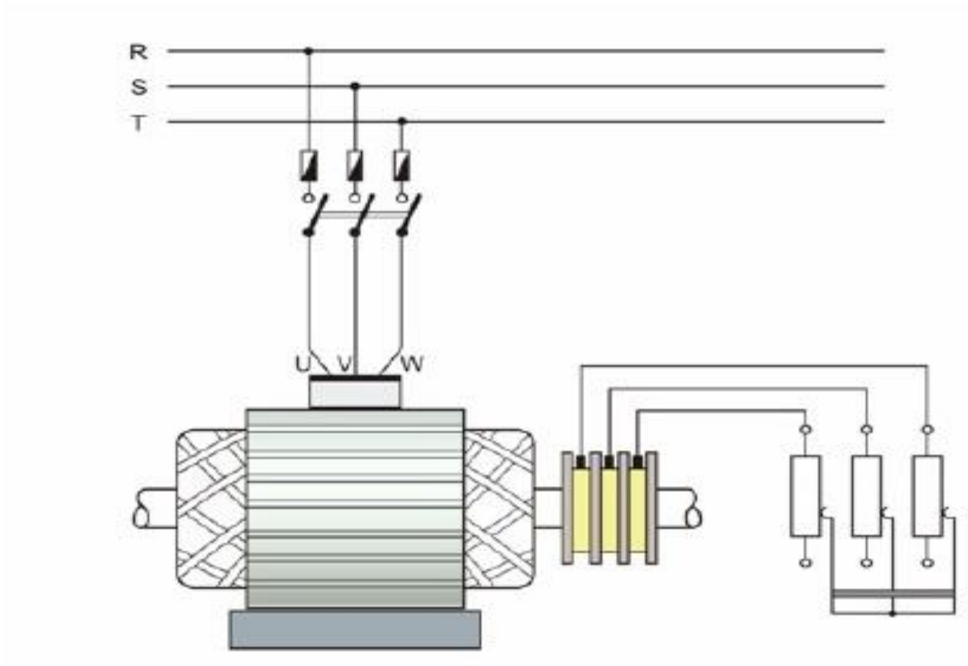


Figure 3. Slip-ring motor

1.7 Equivalent circuit of an asynchronous motor

The phasor model of the asynchronous motor, considering the steady-state motor quantities, is often used in the scalar control method. Before deriving the motor model, several assumptions are made:

- the stator windings are identical and sinusoidally distributed,
- the magnetic system is linear.

The voltage equations (2.1 and 2.2), with respect to the machine variables, can be expressed as:

$$V_s = R_s I_s + j\omega_e \lambda_s \tag{2.1}$$

$$0 = R_r I_r + j\omega_s \lambda_r$$

Where V_s is the stator voltage vector, I_s is the stator current vector, λ_s is the stator leakage flux vector, I_r is the instantaneous rotor current vector, λ_r is the rotor leakage flux vector, ω_e is the synchronous speed, ω_r is the rotor electrical speed, and $\omega_s = \omega_e - \omega_r$
is the slip speed. **(2.2)**

For a magnetically linear system, the leakage flux vector is calculated as follows (equations 2.3 and 2.4).

$\lambda_s = L_s I_s + L_m I_r$ Stator leakage flux **(2.3)**

$\lambda_r = L_m I_s + L_r I_r$ Rotor leakage flux **(2.4)**

In the above equations, R_s is the stator resistance, R_r is the rotor resistance, L_m is the mutual inductance, L_s is the stator inductance, and L_r is the rotor inductance.

In Figure 4, the equivalent circuit of the asynchronous motor is shown. $L_{ls} = L_s - L_m$ is the stator leakage inductance, $L_{lr} = L_r - L_m$ is the rotor leakage inductance,, a $s = \omega s / \omega e$ $s = \omega_s / \omega_e$ is the slip ratio.

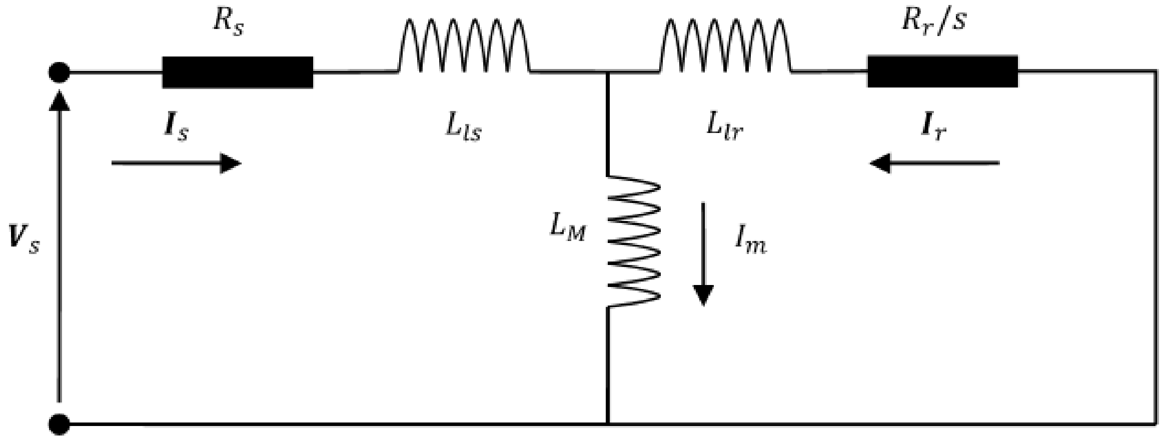


Figure 4. Equivalent circuit of the asynchronous motor

According to equations (3.1) ~ (3.4), the stator voltage and current are related by the equation:

$$V_s = \left(R_s + \frac{\omega_s \omega_e L_M^2 R_r}{R_r^2 + (\omega_s L_r)^2} \right) I_s + j \frac{L_s R_r^2 + \omega_s^2 \left(L_s L_r^2 - L_r L_M^2 \right)}{R_r^2 + (\omega_s L_r)^2} \omega_e I_s \quad \textbf{(2.5)}$$

And the stator leakage flux can be expressed as:

$$\lambda_s = \frac{L_s R_r + \omega_s^2 \left(L_s L_r^2 - L_r L_M^2 \right) - j \omega_s R_r L_M^2}{R_r^2 + (L_r \omega_s)^2} I_s \quad \textbf{(2.6)}$$

Figure 5 shows the vector diagram of the motor quantities when the asynchronous motor operates in motoring mode..

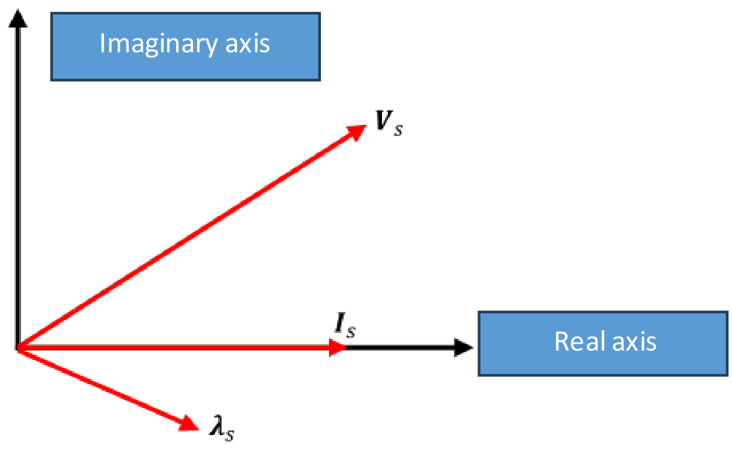


Figure 5. Vector diagram of motor quantities

To describe an induction motor with P (number of) pole pairs, the generated torque can be expressed in terms of current and voltage as shown in equations (2.7) and (8.8).

$$T_e = \frac{3P}{2} * \frac{L_M^2 R_r \omega_s}{R_r^2 + L_r^2 \omega_s^2} |I_S|^2$$

(2.7)

$$T_e = \frac{3P}{2} * \frac{L_M^2 R_r \omega_s}{\left[R_s R_r + \omega_s \omega_e \left(L_M^2 - L_s L_r \right)^2 + R_r^2 L_s^2 \right]} \left| \frac{V_s}{\omega_e} \right|^2 \qquad \textbf{(2.8)}$$

Where |x| denotes the magnitude of a vector. By neglecting the stator resistance, equation (2.8) can be simplified and expressed by the following relation (2.9).

$$T_e = \frac{3P}{2} * \frac{L_M^2 R_r \omega_s}{\omega_s^2 (L_M^2 - L_s L_r)^2 + R_r^2 L_s^2} \left| \frac{V_s}{\omega_e} \right|^2 \qquad \textbf{(2.9)}$$

1.8 Torque–speed characteristic of asynchronous motors

Figure 6 shows the typical torque characteristics of an asynchronous motor. The X-axis represents speed and slip, while the Y-axis shows torque and current. The characteristics are drawn at the rated voltage and frequency supplied to the stator.
During startup, the motor usually draws up to seven times the rated current. This high current is the result of the stator and rotor flux, stator and rotor losses, winding losses, and bearing friction losses. This large starting current overcomes these losses and produces torque on the rotor, causing it to begin rotating.
At startup, the motor delivers about 1.5 times the rated torque of the motor. This starting torque is also called the locked-rotor torque (LRT). As the speed increases, the current drawn by the motor decreases slightly (see Figure 6).

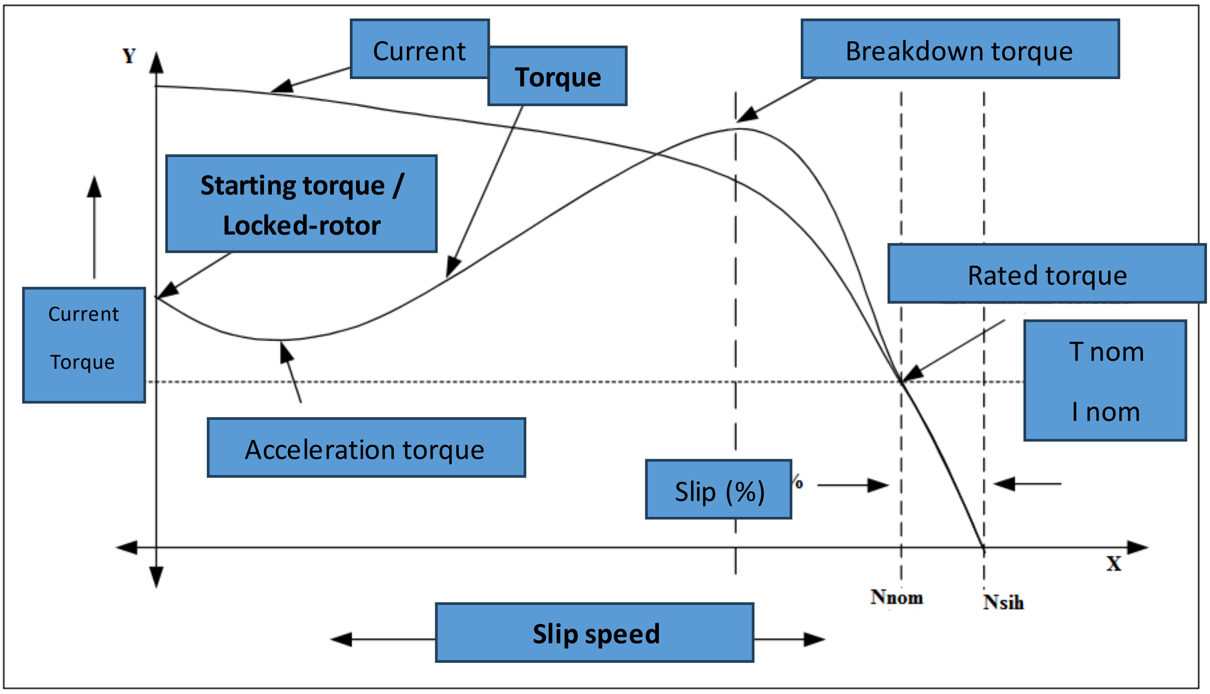


Figure 6. Torque–speed characteristic of an asynchronous motor

The current decreases significantly as the motor speed approaches ~80% of the rated speed. At the rated speed, the motor draws the rated current and delivers the rated torque.

At rated speed, if the load on the motor shaft is increased beyond the rated torque, the speed begins to drop and the slip increases. When the motor operates at approximately 80% of synchronous speed, the load can be increased up to 2.5 times the rated torque. This torque is called the breakdown torque. If the load on the motor continues to increase, the motor will stall.

Furthermore, when the load exceeds the rated load, the load current increases. Due to this higher current flow in the windings, the winding losses also increase, which leads to a rise in the motor winding temperature. Motor windings can withstand different temperatures depending on the insulation class and the motor cooling system. Some motor manufacturers provide overload and duty cycle data. If the motor is overloaded longer than recommended, it may burn out.

As can be seen from the torque characteristics, the torque is highly nonlinear as the speed varies. In many applications, the speed needs to be varied, which makes the torque variable. In this work, we will address a simple speed control of the asynchronous motor using the V/F method in open loop, also known as scalar control.

2. Scalar control

As can be seen from the torque characteristics, the asynchronous motor draws the rated current and delivers the rated torque at the base speed. When the load increases (overload), the speed decreases and the slip increases. As mentioned in the earlier section, the motor can take up to 2.5 times the rated torque with a speed drop of about 20%. Any further increase in the shaft load can stall the motor.

The torque developed by the motor is directly proportional to the magnetic field produced by the stator. Therefore, the voltage applied to the stator is directly proportional to the product of the stator flux and the angular speed. This makes the flux produced by the stator proportional to the ratio of the applied voltage to the supply frequency.

By varying the frequency, the motor speed can be changed. Hence, by varying both the voltage and the frequency in the same ratio, the flux—and therefore the torque—can be kept constant over the entire speed range. (3.1)

Stator voltage (V) ∝ [Stator flux (ϕ)] * [Angular speed (ω)]

$$V \propto \phi * 2\pi F \qquad \textbf{(3.1)}$$

$$\phi \propto \frac{V}{f}$$

In Figure 7, the relationship between voltage and torque with respect to frequency is shown. The Y-axis represents voltage, and on the X-axis the frequency increases up to the rated speed. At the rated speed, both voltage and frequency reach their rated values specified on the nameplate.

We can force the motor to operate above its rated speed by increasing the frequency. However, the voltage cannot be increased above the rated voltage. Therefore, only the frequency can be increased, which results in field weakening and a reduction of the available torque.

Above the rated speed, losses increase significantly, and thus the torque curve becomes nonlinear with respect to speed or frequency.

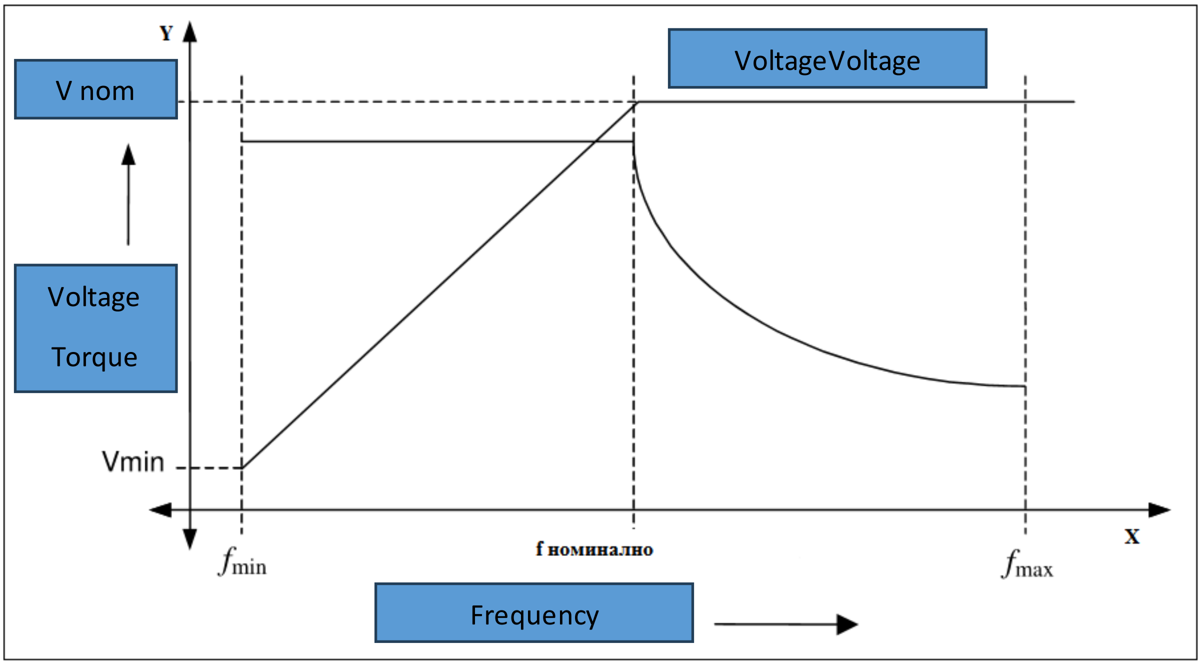


Figure 7. Relationship of voltage and torque with respect to frequency

3. Implementation

3.1 Output IGBT stage

A 3-phase asynchronous motor is connected to a 3-phase bridge inverter as shown in Figure 8. The power inverter consists of six IGBT transistors, controlled by a microcontroller to generate a three-phase AC output from a DC source. PWM signals generated by the microcontroller control these six transistors.
The transistors IGBTH1 to IGBTH3, connected to DC+, are called the upper switches. The transistors IGBTL1 to IGBTL3, connected to DC–, are called the lower switches. The amplitude of the phase voltage is determined by the duty cycle of the PWM signals. While the motor is running, three out of the six switches will be on at any given time—either one upper and two lower switches, or one lower and two upper switches.
By alternately switching certain groups of transistors, a rectangular voltage waveform rich in harmonics (caused by the PWM switching frequency) is generated. The inductive nature of the motor stator windings filters this current to produce a three-phase sinusoidal waveform with negligible harmonics.
When the transistors are turned off, the inductive nature of the windings resists any sudden change in current flow until all the energy stored in the windings is released. To facilitate this, ultra-fast diodes are provided across each transistor. These diodes are known as freewheeling diodes or flyback diodes.
To prevent a short circuit of the DC source, the upper and lower transistors of the same half-bridge must never be turned on simultaneously. This is achieved by inserting a dead time when switching transistors in the same half-bridge branch, either through software programming of the PWM registers or by using specially designed driver circuits with built-in dead time. The typical dead time ranges from 100–500 ns. One such circuit, used in this project, is the IR2109 driver IC from Infineon Technologies.

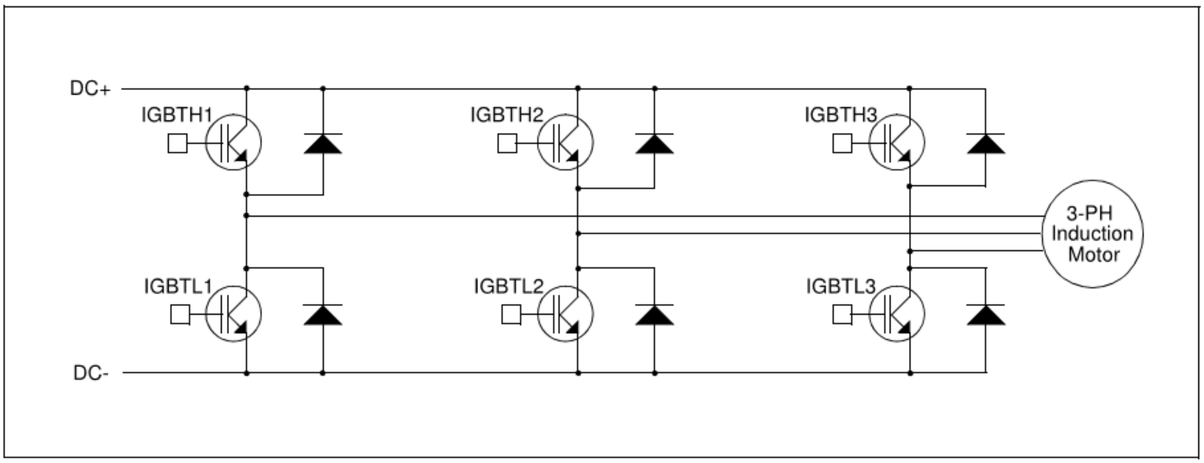


Figure 8. Drive output stage with IGBT transistors

3.2 Control of the output stage

As mentioned earlier, the Atmega 2560 microcontroller has 15 PWM outputs, each of which can be controlled independently to generate three independent PWM signals phase-shifted by 120° using the SPWM technique, which will be described later. In this example, we use Timer 5 of the microcontroller, which is configured to operate in PWM mode. The frequency is set by writing a specific combination into the PWM registers of the microcontroller. A detailed description of the configuration is provided in the appendix of the application program, written in the “ARDUINO” development environment, and included on the CD attached to this work.

To generate a variable three-phase AC voltage from a DC source, six PWM signals are required to control the output transistors individually. Since we are using only three PWM pins of the microcontroller, with the help of IR2109 IGBT drivers we generate all six signals to drive the IGBT transistors. The IR2109 IC has only one input, to which a single PWM pin of the microcontroller is connected. Inside the IC, its control logic generates complementary output signals for the upper and lower switches of the half-bridge, with built-in dead time. Thus, with one input signal, the IR2109 drives both transistors of the half-bridge, and the same applies to the remaining two half-bridge legs. In total, three IR2109 IGBT drivers are required.

3.3 BOOTSTRAP supply of the upper output transistors

One of the most commonly used methods for powering the current circuit of the upper output driver transistors is the **BOOTSTRAP supply**. The BOOTSTRAP supply consists of a BOOTSTRAP diode D1 and a BOOTSTRAP capacitor C1 (Figure 9). The advantage of this method is its simplicity and low cost. The maximum voltage that the bootstrap capacitor (VBS) can reach depends on the components used in the bootstrap supply circuit.

3.4 Bootstrap capacitor calculation

The BOOTSTRAP capacitor must be chosen so that it delivers enough energy to reliably turn on the upper output transistor without discharging excessively. In practice, the value of the bootstrap capacitor is taken to be about 10 times the gate capacitance of the upper output transistor (Cg) to ensure sufficient supply and a steady current during one PWM cycle.

The calculation of Cg is based on the gate drive supply voltage for the high-side IGBT—i.e., the gate-emitter voltage (Vge)—and the gate charge time (Qg).

In this work, the approximate voltage (Vge) can be expressed by the following equation (3.2).

$$VGE \approx VDD - VDboot = 13.3 - 1 \approx 12.3V \quad (3.2)$$

VDD- Supply voltage of the driver stage
VDboot- Voltage drop across the bootstrap diode

The IGBT transistor HGTG20N60 has Qg = 142 nC according to the datasheet, so the equivalent gate capacitance would be, according to equation (3.3)

$$\begin{aligned} C_g &= Q_{g*}V_{GE} = 142nC * 12.3V = 142C * 10^{-7} * 12.3V = 1.74\mu F \\ C_1 &\geq 10 * C_g = 17.4\mu F \end{aligned} \quad (3.3)$$

3.5 Gate protection and discharge circuit

The gate discharge circuit is designed with fast diodes 1N4148 (D2–D4), resistors R1–R4, and resistors R3–R6. The gate charging is carried out through resistors R2–R5. Gate overvoltage protection (Vge) is provided by Zener diodes D3–D5 rated at 15 V.

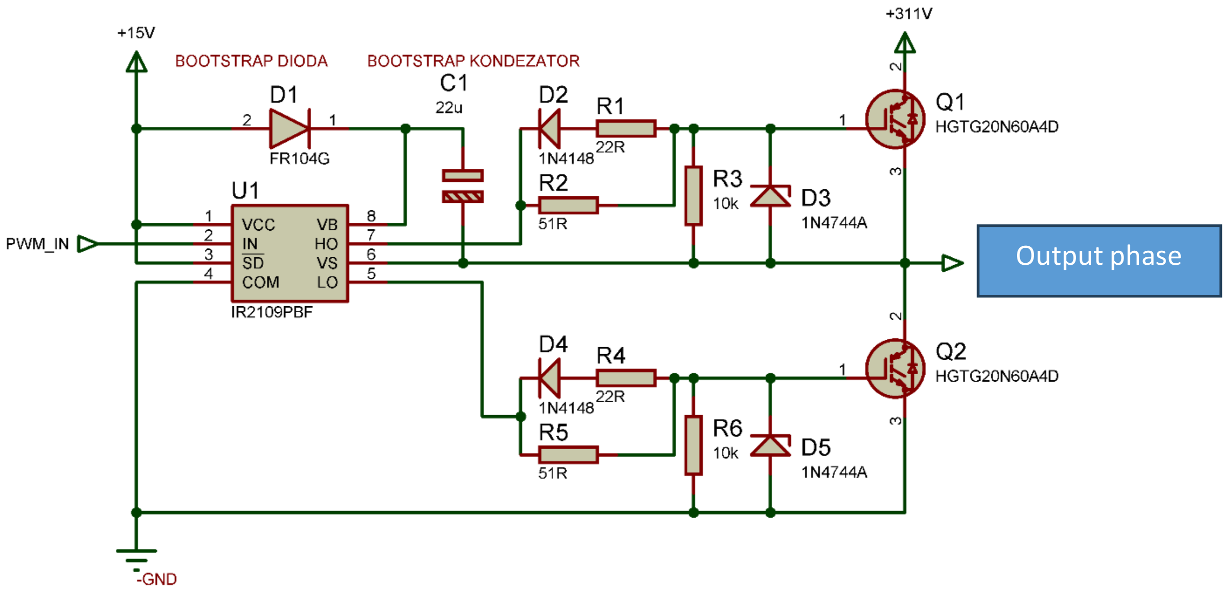


Figure 9. Driver and output stage

3.5 Power supply

The power supply of the frequency converter is 220 V–50 Hz. The voltage is rectified by a diode bridge in a Graetz connection, with a resistor limiting the initial charging current of the capacitor. After a time delay defined in the microcontroller program, a relay switches on, bypassing the resistor and applying full voltage to a 470 µF capacitor, which filters the rectified voltage and forms a DC voltage of 311 V for the power stage.
Additionally, there is another rectifier 220 V / 13.3 V / 5 V for supplying the microcontroller (+5 V) and the output stage (+13.3 V) (Figure 10).

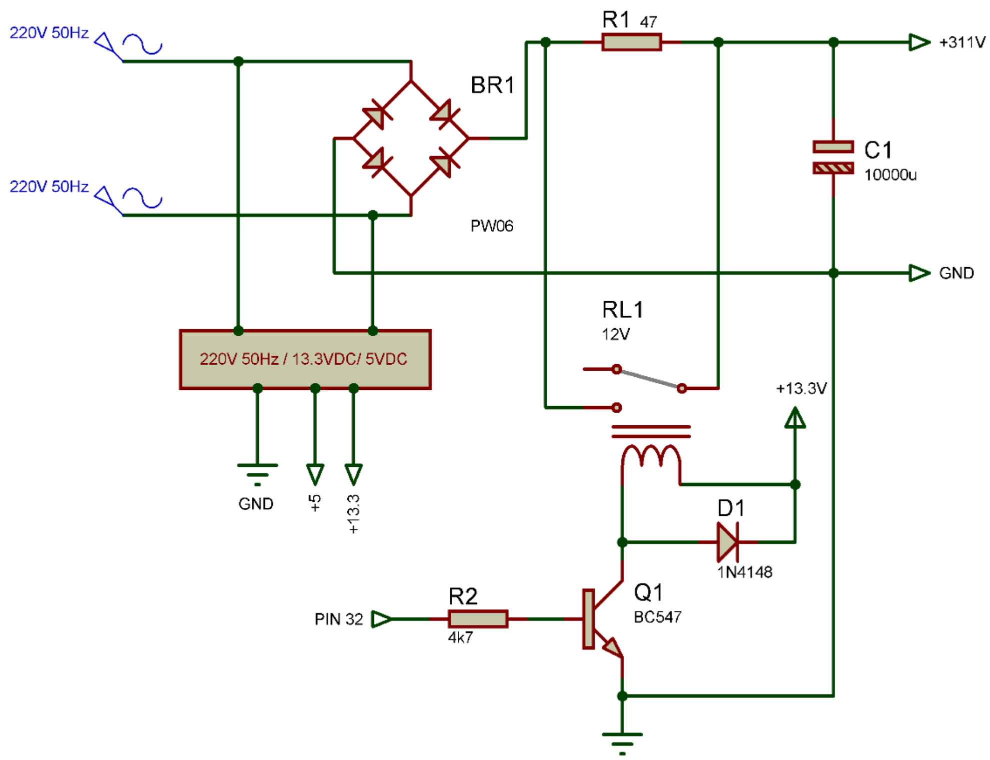


Figure 10. Power supply

3.6 Voltage measurement

We measure the voltage on the DC bus using a voltage divider, as shown in Figure 11.

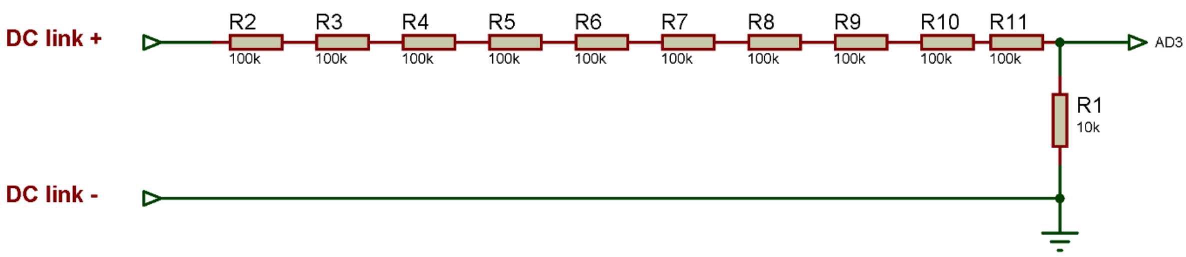


Figure 11. Voltage divider

In the microcontroller program, voltage conversion is performed on the AD3 pin using the subroutine **"void merenje_DCnapon()"**. The formula for calculating the voltage is given in equation (3.4).

$$U_{DC\_link} = \frac{U_{out} * V_{ref}}{ADC_{rezolucija}} * \frac{U_{in\_izmereno}}{U_{out\_izmereno}} \qquad 3.4$$

3.7 Current measurement

Current measurement is carried out using the Arduino ACS712 current sensor module. In this work, only one phase of the motor is measured, and the current value is sampled every 20 ms on the AD2 pin of the microcontroller. If the current exceeds the preset value, the motor is stopped. The circuit diagram is shown in Figure 12, and the module is shown in Figure 13.

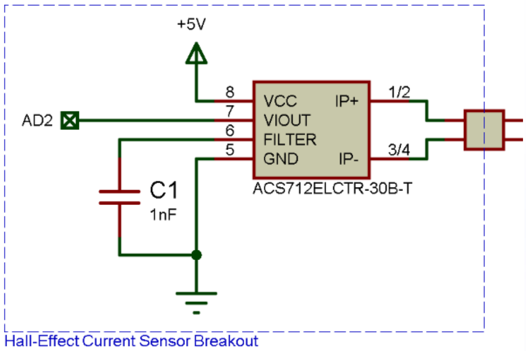


Figure 12. ACS712 circuit diagram Figure 13. Arduino ACS712 module

The subroutine that performs current measurement uses a preinstalled library specifically designed for the ACS712 module and is included with this work on the attached CD. The library is called <ACS712.h>.

4. ARDUINO MEGA – microcontroller

4.1 Schematic of the Arduino Mega development board (Figures 13, 14, 15)

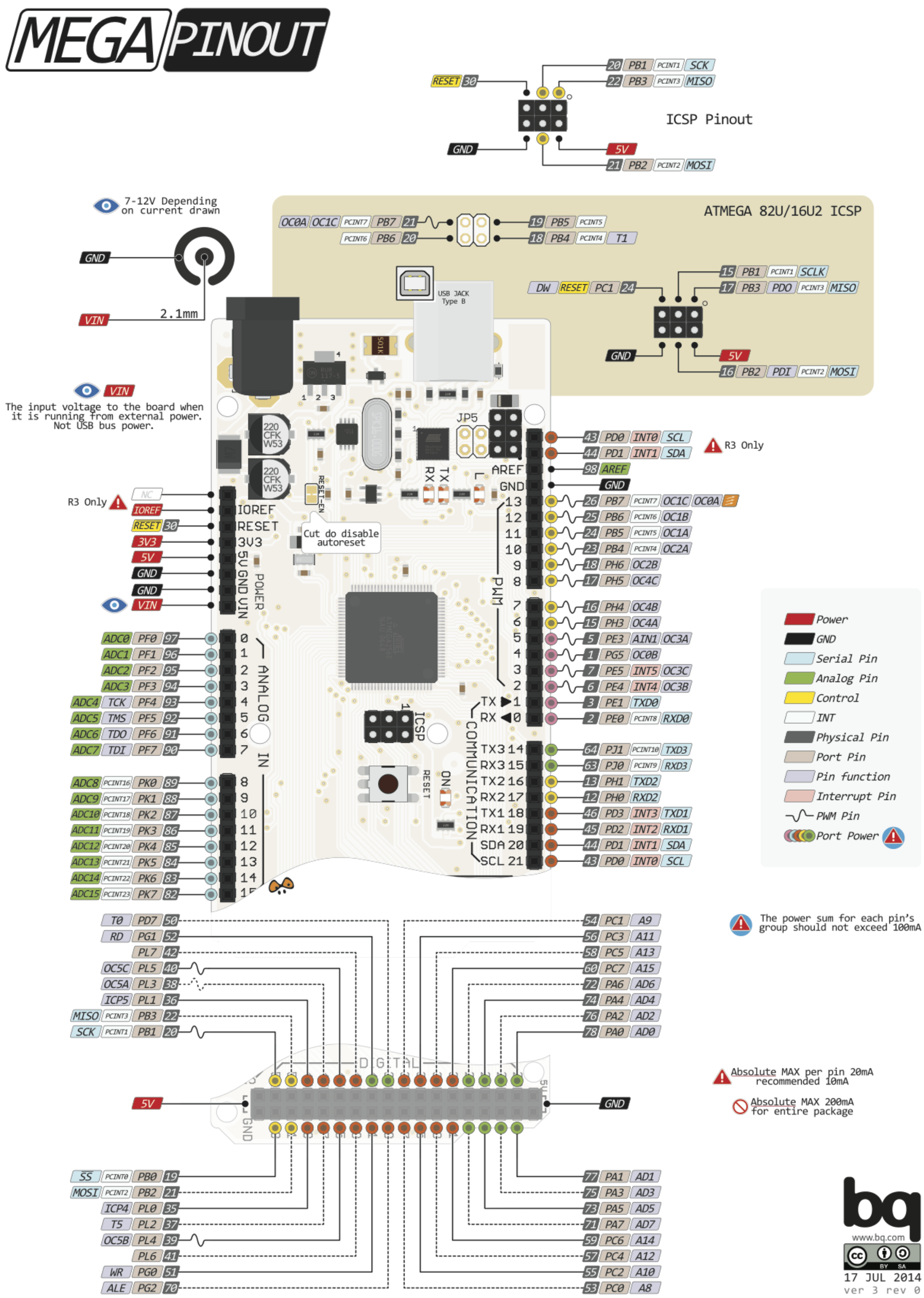


Figure 13.

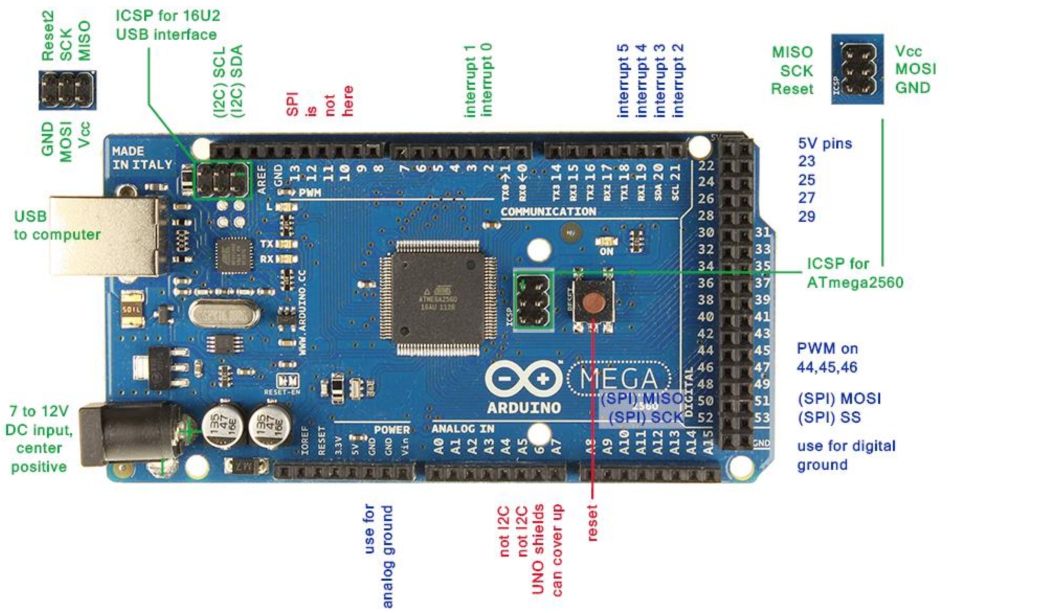


Figure 14.

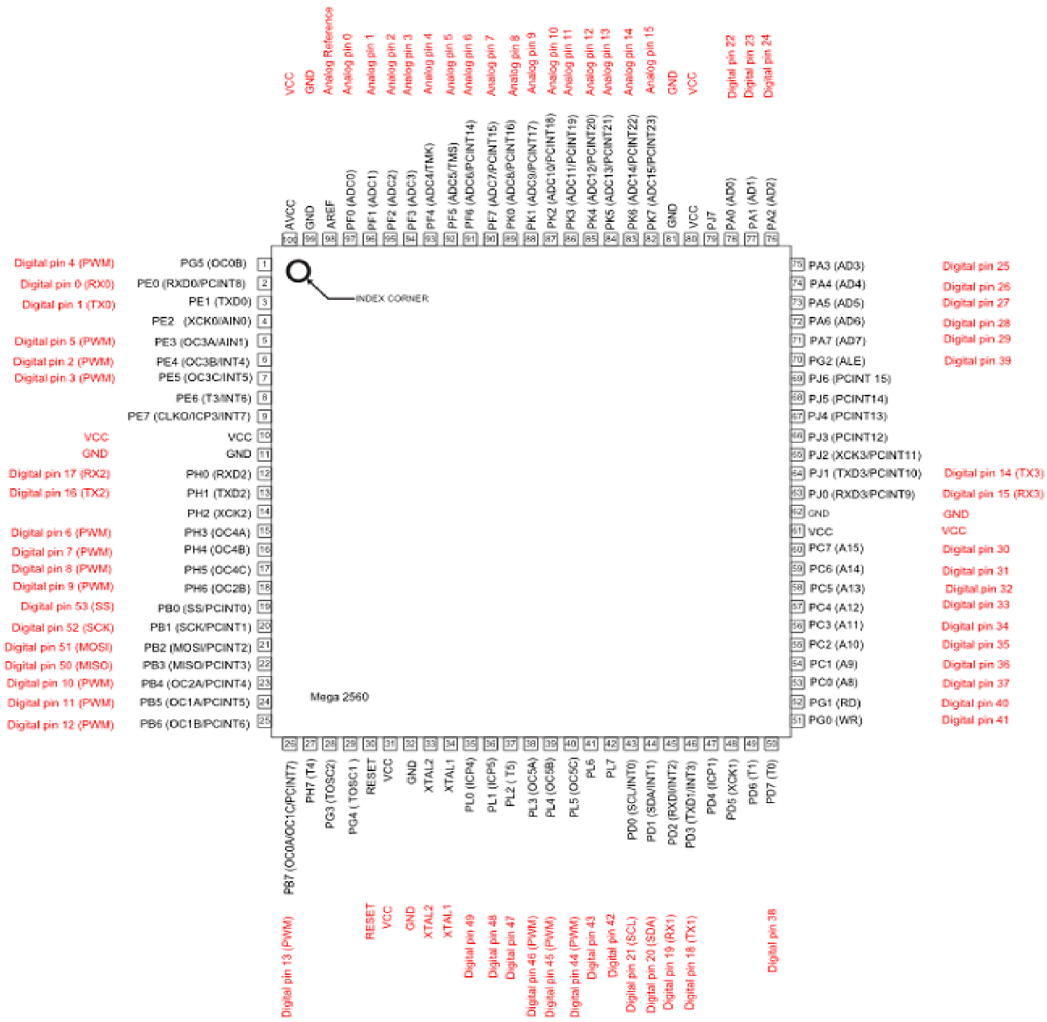


Figure 15.

4.2 Controller Specifications

Microcontroller-----------------------------ATmega2560
Operating Voltage--------------------------5V
Input Voltage (recommended)-----------------7-12V
Input Voltage (limits)----------------------6-20V
Digital I/O---------------------------------54 (of which 15 can be used as PWM outputs)
Analog Inputs-------------------------------16
DC Current per I/O Pin----------------------40 mA
DC Current for 3.3V Pin---------------------50 mA
Flash Memory--------------------------------256 KB (8 KB used by bootloader)
SRAM--------------------------------------8 KB
EEPROM-------------------------------------4 KB
Clock Speed---------------------------------16 MHz

4.3 Digital pins

Pin
Digital Pin 0-------------------------------RXD0 (Serial Port 0)
Digital Pin 1-------------------------------TXD0 (Serial Port 0)
Digital Pin 2-------------------------------External Interrupt 0 - PWM
Digital Pin 3-------------------------------External Interrupt 1 - PWM
Digital Pin 4-------------------------------PWM
Digital Pin 5-------------------------------PWM
Digital Pin 6-------------------------------PWM
Digital Pin 7-------------------------------PWM
Digital Pin 8-------------------------------PWM
Digital Pin 9-------------------------------PWM
Digital Pin 10------------------------------PWM
Digital Pin 11------------------------------PWM
Digital Pin 12------------------------------PWM
Digital Pin 13------------------------------Built-in LED - PWM
Digital Pin 14------------------------------TXD (Serial Port 3)
Digital Pin 15------------------------------RXD (Serial Port 3)
Digital Pin 16------------------------------TXD (Serial Port 2)
Digital Pin 17------------------------------RXD (Serial Port 2)
Digital Pin 18------------------------------TXD (Serial Port 1) - External Interrupt 5
Digital Pin 19------------------------------RXD (Serial Port 1) - External Interrupt 4
Digital Pin 20------------------------------I2C - SDA - External Interrupt 3
Digital Pin 21------------------------------I2C - SCL - External Interrupt 2
Digital Pin 22–43--------------------------General Purpose I/O pins
Digital Pin 44------------------------------PWM
Digital Pin 45------------------------------PWM
Digital Pin 46------------------------------PWM
Digital Pin 50------------------------------MISO
Digital Pin 51------------------------------MOSI
Digital Pin 52------------------------------SCK
Digital Pin 53------------------------------SS

4.4 Arduino Mega Expansion Board (Figures 16, 17)

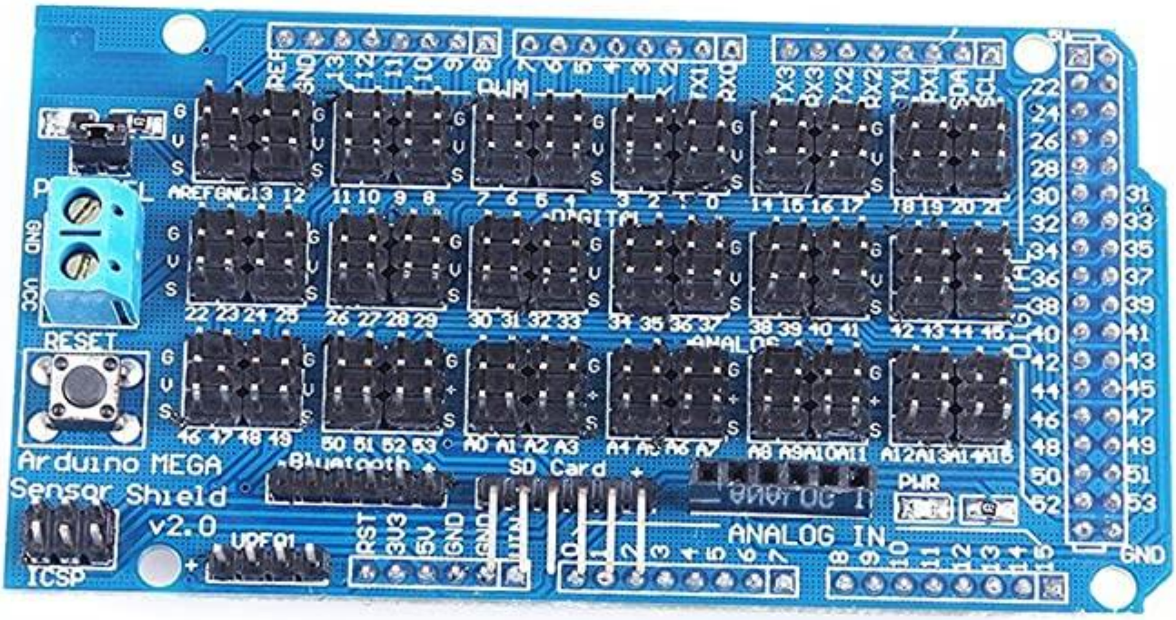


Figure 16.

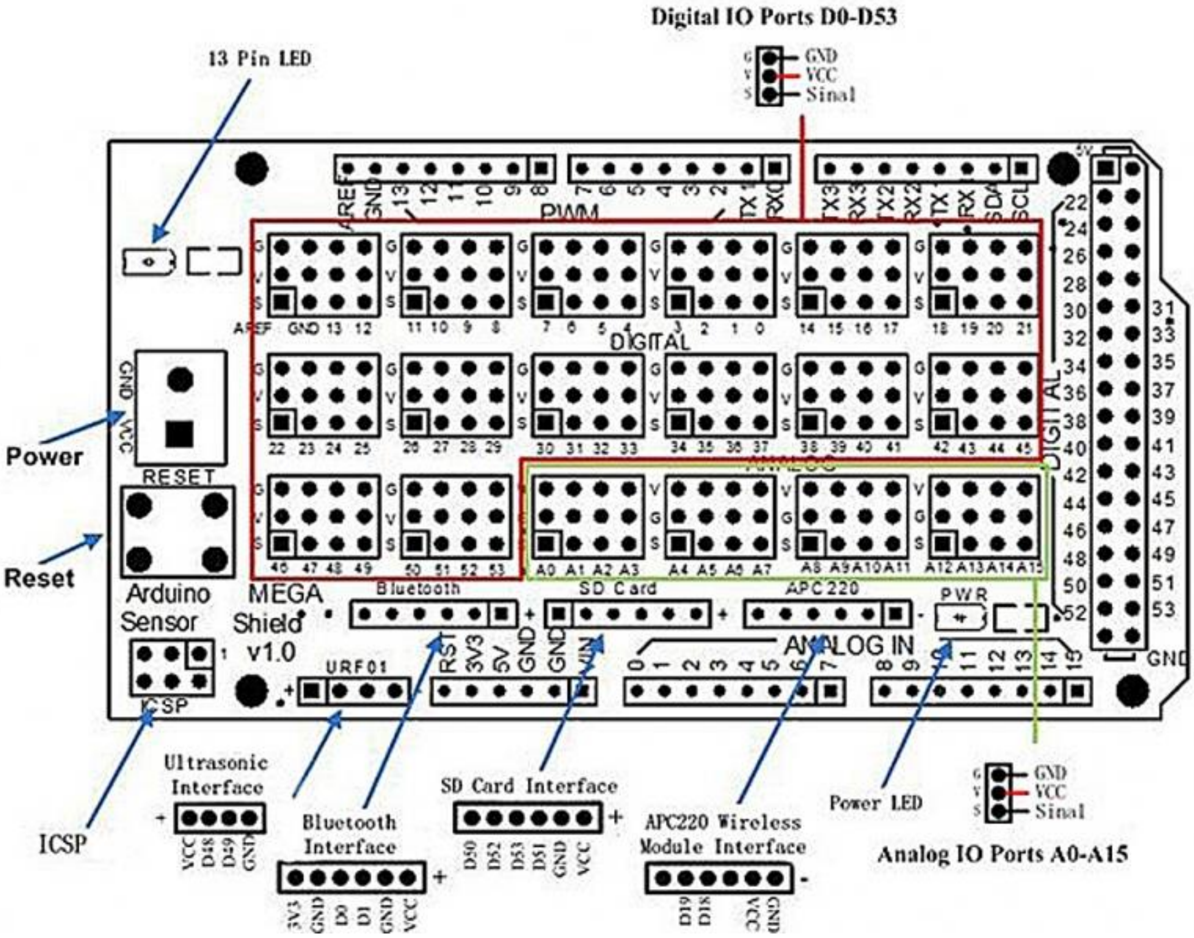


Figure 17.

5. SPWM (Sinusoidal Pulse Width Modulation) technique

By using Timer 5 of the microcontroller, we control the hardware PWM module to generate three independent PWM signals based on a sine table stored in the program memory of the microcontroller. After initialization at startup, the sine table is copied into the working memory of the microcontroller. In this way, the access time to the working registers of the microcontroller is reduced, and the operating characteristics of the three-phase motor are not affected. Each of the three PWM pins of the microcontroller has its own registers in which an offset is defined, and each of these registers points to a value in the table, such that they always maintain a phase shift of 120° with respect to each other.

Figure 18

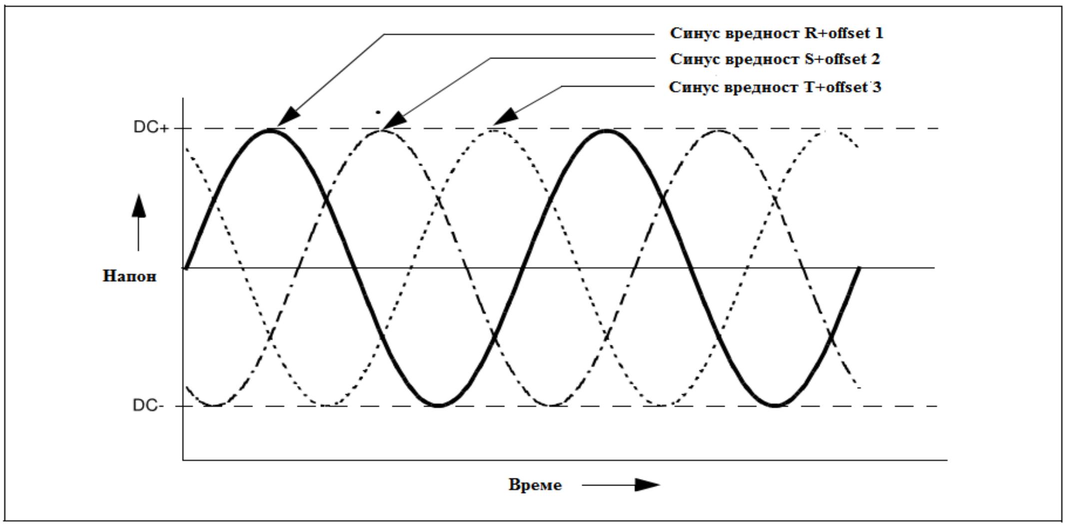


Figure 18. Генерисање трофазног синусног напона

A potentiometer connected to the 10-bit Analog-to-Digital Converter channel (AN0) determines the motor frequency. The microcontroller uses the converter results to calculate the duration of the PWM pulses, and consequently the amplitude and frequency of the motor. To ensure a smooth transition of frequency from higher to lower values and vice versa, the set frequency is read during each Timer 3 interrupt. The motor frequency value is controlled by Timer 1, which uses the result of the ADC to provide the preloaded value (preload value of the OCR1 register) at which the interrupt routine will occur, defined by the following relation: The required speed value in [Hz] is obtained by reading the Analog-to-Digital Converter channel (AN0), and depending on the potentiometer position, the corresponding frequency value is calculated (Equation 4.1).

freq = map(analogRead(FREQ_POT), POT_MIN, POT_MAX, FREQ_MIN, FREQ_MAX); **(4.1)**

where is:

freq = value of the set (calculated) frequency;
FREQ_POT = ADC channel on the controller (in this case AN0);
POT_MIN = minimum ADC conversion value (0);
POT_MAX = maximum ADC conversion value (1023);
FREQ_MIN = minimum motor rotation frequency (7 Hz);
FREQ_MAX = maximum motor rotation frequency (7–70 Hz);
map = Arduino function for calculating the value within the previously defined limits;

Example of the map function for a frequency value of 50 Hz, equation (4.2)

$$map = \frac{(\mathrm{FREQ\_POT} - \mathrm{POT\_MIN}) * (\mathrm{FREQ\_MAX} - \mathrm{FREQ\_MIN})}{(\mathrm{POT\_MAX} - \mathrm{POT\_MIN})} + FREQ\_MIN$$

$$map = \frac{(1023-0)*(50-7)}{(1023-0)} + 7 = 50 \qquad \textbf{(4.2)}$$

Timer 1 Settings:

register **TCCR1A- (refer to Table 5 and Table 6)**

Table 1. Register**TCCR1A**

Bit	7	6	5	4	3	2	1	0
(0x80)	COM1A1	COM1A0	COM1B1	COM1B0	-	-	WGM11	WGM10
READ/WRITE	R/W	R/W	R/W	R/W	R	R	R/W	R/W
Подешено у програму	0	0	0	0	0	0	0	0

Register **TCCR1B- (refer to Table 5 and Table 6)**

Table 2. Register**TCCR1B**

Bit	7	6	5	4	3	2	1	0
(0x81)	ICNC5	ICES5	-	WGM13	WGM12	CS12	CS11	CS10
READ/WRITE	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
Подешено у програму	0	0	0	0	1	0	0	0

The required value for the Timer1 OCR1A register in order to obtain a frequency of 50 Hz is calculated using the following equation (4.3). Since the sine table contains 39 predefined values, the time at which the interrupt routine will occur is calculated as follows:

$$t = \frac{1}{Hz} \rightarrow t = \frac{1}{50} = 0.02ms$$

$$t_{\mathrm{int}} = \frac{t}{Sine\_uzoraka\_po\_ciklusu} \rightarrow \frac{0.02ms}{39} = 512\mu s$$

$$f_{\mathrm{int}} = \frac{1}{t_{\mathrm{int}}} = \frac{1}{512} = 1950Hz \qquad \textbf{(4.3)}$$

$$OCR1A = \frac{f\_osc}{prescaler * \mathrm{f_{int}}} = \frac{16000000MHz}{1*1950} = 8205$$

-f_osc = 16 MHz system clock of the microcontroller
-prescaler = in this example it is 1 (no division of the base frequency)

-map_result = value calculated by the microcontroller

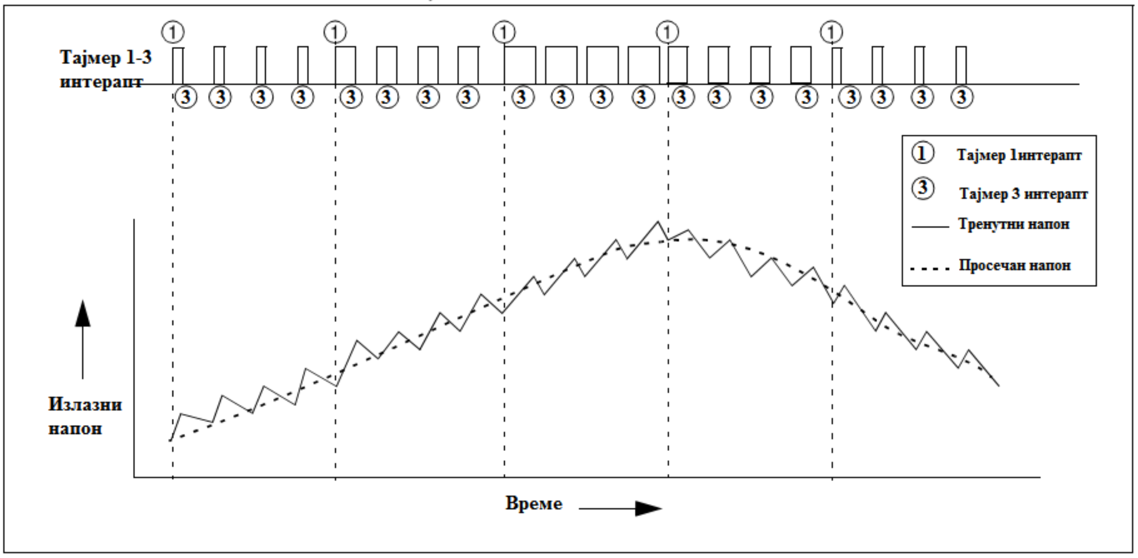
-Sine_samples_per_cycle = number of predefined values in the sine table (39)

-tint = required interrupt time for the given frequency

-fint = interrupt frequency

-OCR1A = timer register responsible for the interrupt routine

After each Timer 1 interrupt (Figure 17), a value is read from three sine tables – one for each phase. The tables are formatted so that each has a phase shift of 120°. The value read from the tables is scaled based on the motor frequency. The sine table values are then multiplied by the amplitude register, which defines the PWM signal duration, and the calculated result is written into the OCR5A, OCR5B, and OCR5C registers of Timer 5, i.e., the PWM registers



Слика 17. Timer 1 interrupt, PWM signal duration and voltage

The amplitude value of the PWM signal is calculated using equation (4.4).

$$Amp = \frac{zadata\_frekfencija}{nom\_frekfencija\_motora} \qquad (4.4)$$

In each interrupt routine, the amplitude value is checked, and if it is greater than 1, it is limited to 1.
After that, the PWM registers are updated for the next reading. If the button for changing the motor direction is pressed, the positions of the PWM_Soft_x registers are swapped, thereby achieving a phase shift, and the motor starts rotating in the opposite direction.

Direction 1:	Direction 2:
$FAZA\_U_{(OCR3A)}$ = (PWM_Soft_A);	$FAZA\_U_{(OCR3A)}$ = (PWM_Soft_A);
$FAZA\_V_{(OCR3B)}$ = (PWM_Soft_B);	$FAZA\_V_{(OCR3B)}$ = (PWM_Soft_C);
$FAZA\_W_{(OCR3C)}$ = (PWM_Soft_C);	$FAZA\_W_{(OCR3C)}$ = (PWM_Soft_B);

Timer 5 is set to a frequency of 6 kHz, which is also the PWM frequency, i.e., the switching frequency of the controller.
Timer 5 Settings:
Register TCCR5A: (see Table 5 and 6)

Table 3. Регистар **TCCR5A**

Bit	7	6	5	4	3	2	1	0
(0x80)	COM5A1	COM5A0	COM5B1	COM5B0	COM5C1	COM5C0	WGM51	WGM50
READ/WRITE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Подешено у прогтраму	1	0	1	0	1	0	1	0

Register TCCR5B: (see Table 5 and 6)

Table 4. Регистар **TCCR5B**

Bit	7	6	5	4	3	2	1	0
(0x81)	ICNC5	ICES5	-	WGM53	WGM52	CS52	CS51	CS50
READ/WRITE	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
Подешено у прогтраму	0	0	0	1	0	0	1	0

Timer 5 is configured to operate at a clock of 6 kHz.

The duration of a single pulse is $t = \frac{1}{f} = \frac{1}{6000} = 166\mu s$

Since Timer 5 operates in *phase correct mode*, i.e., it counts from 0 up to the set value and then from the set value back down to 0, the required value of the ICR5 register is calculated according to equation (4.5).

$$ICR5 = \frac{f\_osc}{prescaler * \text{switchig\_freq} * 2} = \frac{16000000Mhz}{8 * 6000 * 2} = 166 \qquad (4.5)$$

Accordingly, the maximum value in the sine table can be 166, as shown in Program Code 1.

```
const byte PROGMEM sine[] [39]={
{1,4,9,16,24,34,45,57,70,83,96,109,121,132,142,150,157,162,165,166,165,162,157,150,142,132,121,109,96,83,70,57,45,34,24,16,9,4,1},
{132,142,150,157,162,165,166,165,162,157,150,142,132,121,109,96,83,70,57,45,34,24,16,9,4,1,1,4,9,16,24,34,45,57,70,83,96,109,121},
{121,109,96,83,70,57,45,34,24,16,9,4,1,1,4,9,16,24,34,45,57,70,83,96,109,121,132,142,150,157,162,165,166,165,162,157,150,142,132},
};
```

Program Code Excerpt 1. (Sine table values for three PWM signals)

Table 5. Timer Prescaler Setting

CSx2	CSx1	CSx0	Опис
0	0	0	Нема извора такта(тајмер/бројач заустављен)
0	0	1	Такт/1 (без прескалера)
0	1	0	Такт/8 (од прескалера)
0	1	1	Такт/64 (од прескалера)
1	0	0	Такт/256 (од прескалера)
1	0	1	Такт/1024 (од прескалера)
1	1	0	Спољни извор такта на Т1 пин. Clock on faling edge
1	1	1	Спољни извор такта на Т1 пин. Clock on rising edge

Table 6. Description of bits for different timer operating modes

Мод	WGMx3	WGMx2 (CTC1)	WGMx1 (PWMx1)	WGMx0 (PWMx0)	Режим рада	TOP	Тајмер OCRxx	TOV1 Flag Set on
0	0	0	0	0	Normal	0xFFFF	Immediate	MAX
1	0	0	0	1	PWM, Phase Correct, 8-bit	0x00FF	TOP	BOTTOM
2	0	0	1	0	PWM, Phase Correct, 9-bit	0x01FF	TOP	BOTTOM
3	0	0	1	1	PWM, Phase Correct, 10-bit	0x03FF	TOP	BOTTOM
4	0	1	0	0	CTC	OCR1x	Immediate	MAX
5	0	1	0	1	Fast PWM, 8-bit	0x00FF	BOTTOM	TOP
6	0	1	1	0	Fast PWM, 9-bit	0x01FF	BOTTOM	TOP
7	0	1	1	1	Fast PWM, 10-bit	0x03FF	BOTTOM	TOP
8	1	0	0	0	PWM, Phase and Frequency Correct	ICRx	BOTTOM	BOTTOM
9	1	0	0	1	PWM, Phase and Frequency Correct	OCRxA	BOTTOM	BOTTOM
10	1	0	1	0	PWM, Phase Correct	ICRx	TOP	BOTTOM
11	1	0	1	1	PWM, Phase Correct	OCRxA	TOP	BOTTOM
12	1	1	0	0	CTC	ICRx	Immediate	MAX
13	1	1	0	1	(Reserved)	-	-	-
14	1	1	1	0	FAST PWM	ICRx	BOTTOM	TOP
15	1	1	1	1	FAST PWM	OCRxA	BOTTOM	TOP

5.1 Timing diagrams

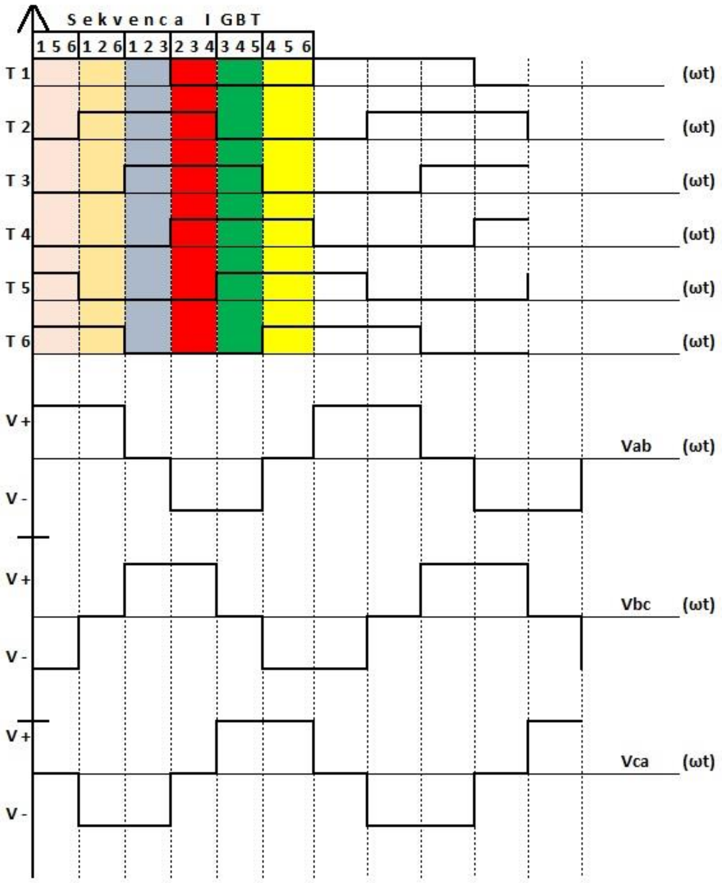


Figure 18. Conduction timing diagram at 180°

Table 7. Sequences of transistor triggering

Sekv/Igbt	IGBT	IGBT	IGBT	IGBT	IGBT	IGBT	Napon	Napon	Napon
	T1	T2	T3	T4	T5	T6	Vab	Vbc	Vca
156	1	0	0	0	1	1	DC+	DC-	O
126	1	1	0	0	0	1	DC+	O	DC-
123	1	1	1	0	0	0	O	DC+	DC-
234	0	1	1	1	0	0	DC-	DC+	O
345	0	0	1	1	1	0	DC-	O	DC+
456	0	0	0	1	1	1	O	DC-	DC+

6.3 Main program, subroutines, and interrupt routines

6.4 Main Program (void loop())

Here, subroutines are executed and called, key presses are checked, and the current values are displayed on the screen. The DC voltage value is monitored against predefined undervoltage and overvoltage limits, and if the voltage is outside the set range, the protection is triggered, stopping the motor and disconnecting the supply. The motor current is also monitored against the predefined nominal current value, and if the current exceeds the set value, the protection is activated and the motor is shut down. Auxiliary variables are created to generate timers for monitoring voltage and current, as well as for generating the animation that indicates the motor rotation direction. Timers are started to create interrupt routines for generating three-phase voltage using the PWM technique. The ADC0 value of the microcontroller is read to define the motor output frequency.

6.5 Subroutines

6.5.1 void citanje_tastera()

In this subroutine, the value of the microcontroller's ADC1 is read, and by using buttons that change the resistance at the converter input, the button functions are defined, which are used to call various inverter functions. The buttons are defined in the following order:

-Тастер_Лево – button left
-Тастер_Десно – button right
-Тастер_Горе – button UP
-Тастер_Доле – button Down
-Тастер_Стоп/Подешавање – utton Setup

6.5.2 void prikaz()

Here, the current and set values of frequency, voltage, current, etc. are displayed.

6.5.3 void Motor_stop()

It stops the motor, halts the timers and registers responsible for the controller operation, and sets the microcontroller registers to predefined values for a new controller start.

6.5.4 void podesavanje()

The condition for this subroutine to be called is that the controller is enabled and the motor is stopped.
The screen displays “STOP”, and by holding the Stop/Settings button for more than 2 seconds, the menu “SETTINGS” is entered.

Here, the following parameters are defined:

- **FREQ MAX OUT** – maximum output frequency (30–70 Hz)
- **MOTOR CURRENT nom.** – nominal motor current (0–20 A)
- **ACCELERATION RAMP (sec)** – motor run-up time in seconds (1–20 s)
- **DECELERATION RAMP (sec)** – motor stop time in seconds (1–20 s)
- **DC overvoltage** – maximum DC voltage value
- **DC undervoltage** – minimum DC voltage value
- **Motor frequency nom.** – motor frequency from the nameplate (50 or 60 Hz)

By holding the Stop/Settings button for more than 2 seconds, the program returns to the main subroutine.

6.5.5 void brzina()

It calculates the required value for the Timer1 OCR1A register, which generates the interrupt routine responsible for the motor’s output frequency.

6.5.6 void merenje_DCnapon()

It is used for measuring the DC voltage (DC link voltage).

6.5.7 void pisanje__citanje_INT_eeprom

his subroutine is used for reading from and writing to the microcontroller’s EEPROM, and it is called during the controller startup and when exiting the “SETTINGS” subroutine.

6.6 Interrupt routines

6.6.1 ISR(TIMER1_COMPA_vect)

This is the Timer 1 interrupt routine, which is responsible for the output frequency value. It checks whether the stop button is pressed, verifies the rotation direction status, and updates the PWM registers by changing the positions of the sine table readings. The current sine table values, scaled by the amplitude register value, are written into the PWM registers of the microcontroller. At the end of the interrupt routine, the OCR1A register value of Timer 1 is always refreshed, and if necessary, the output frequency of the controller is increased or decreased.

6.6.2 ISR(TIMER3_COMPA_vect)

This Timer 3 interrupt routine is used for smooth deceleration or acceleration of the motor. During each Timer 3 interrupt, the controller frequency is increased or decreased (depending on the position of the frequency potentiometer) by 0.25 Hz until the set frequency is reached.

7. Operation with the controller

7.1 Power-up and System Initialization

Here, the initial values for the controller start are defined. After power is applied, the microcontroller sets the pins as inputs or outputs depending on the function of each pin defined in the program.

Input pins:

- **AD0** – reading the potentiometer for output frequency
- **AD1** – reading the buttons used for controller operation
- **AD2** – current measurement
- **AD3** – voltage measurement

Output pins:

- **Pin 32 (releykon)** – enables the mains capacitor relay
- **Pin 35 (releynapR)** – enables the controller supply relay
- **Pin 34 (releynap0)** – enables the controller supply relay
- **Pin 33 (enableIGBT)** – controls the switching of output transistors
- **Pin 36 (kontrola)** – checks the interrupt frequency of Timer 1
- **Pin 46 (FAZA_U)** – PWM output pin
- **Pin 45 (FAZA_V)** – PWM output pin
- **Pin 44 (FAZA_W)** – PWM output pin

7.2 Main loop

After the system initialization, the program enters the loop and the controller is in stop mode (Figure 18). By pressing one of the tasteri, **Taster_Levo** or **Taster_Desno**, the motor starts rotating in the direction corresponding to the selected taster (Figures 19 and 20).

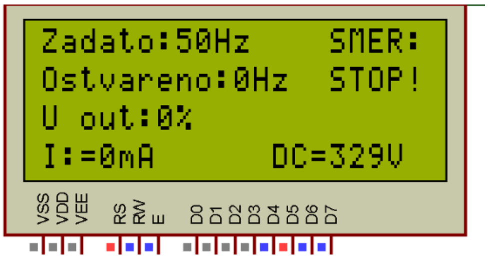


Figure 18. Stop

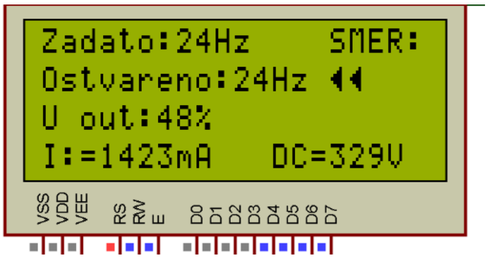


Figure 19. Left direction

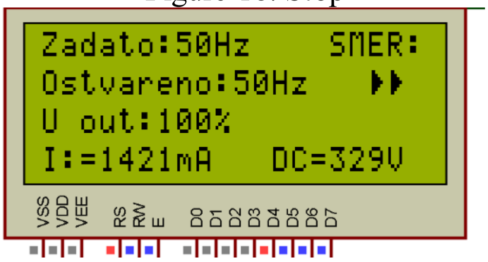


Figure 20. Right direction

When the controller is in stop mode, by pressing **Taster_Stop/Podešavanje** for more than two seconds, the program enters the settings menu (Figure 21). Here we adjust the limits for overvoltage and undervoltage protection, current limits, motor output frequency, acceleration and deceleration ramps, and the nominal motor frequency. With **Taster_Gore** and **Taster_Dole** we change the values of the selected parameters, while with **Taster_Levo** and **Taster_Desno** we select the parameter we want to modify. By pressing **Taster_Stop/Podešavanje** again for more than two seconds, the parameters are written into the EEPROM and the program returns to the main loop.

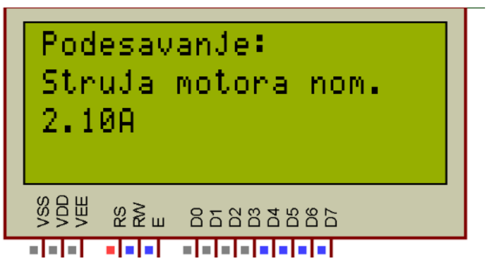


Figure 21. Settings menu

7.3 Taster functions

Figure 22 shows the description of the tasteri used for controller operation.

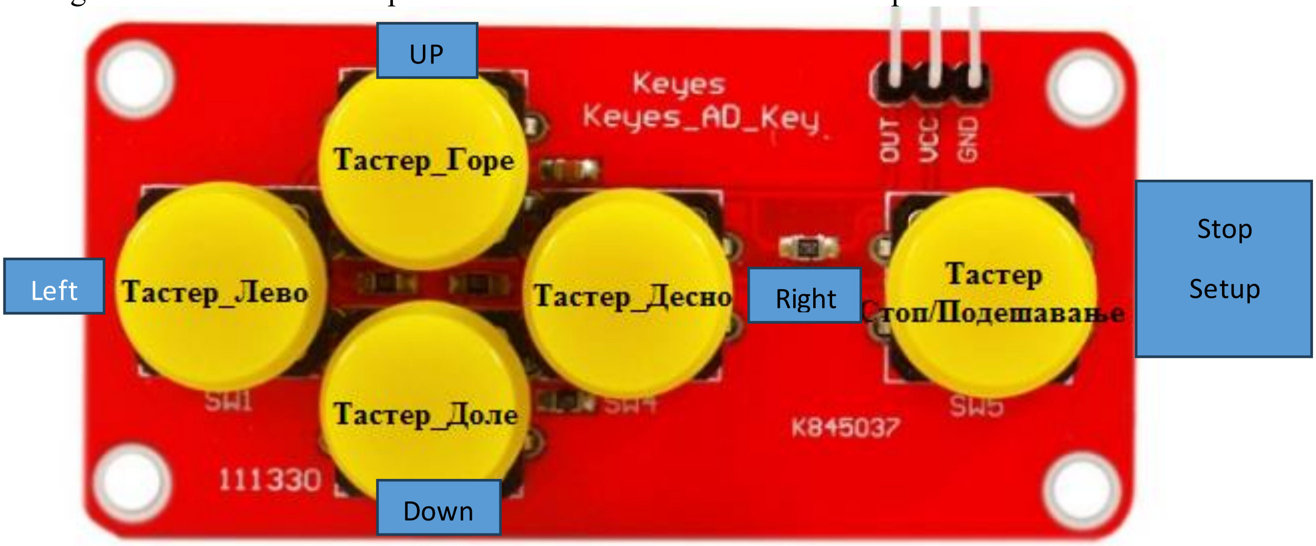


Figure 22. Taster functions of the controller

8. Simulation in MATLAB

Since a high-quality oscilloscope was not available for this work, a simulation was carried out in MATLAB with approximate values of voltage, frequency, and the switching frequency of the PWM signal. The project is included on the CD attached to this thesis.

The simulation schematic is shown in Figures 23 and 24.

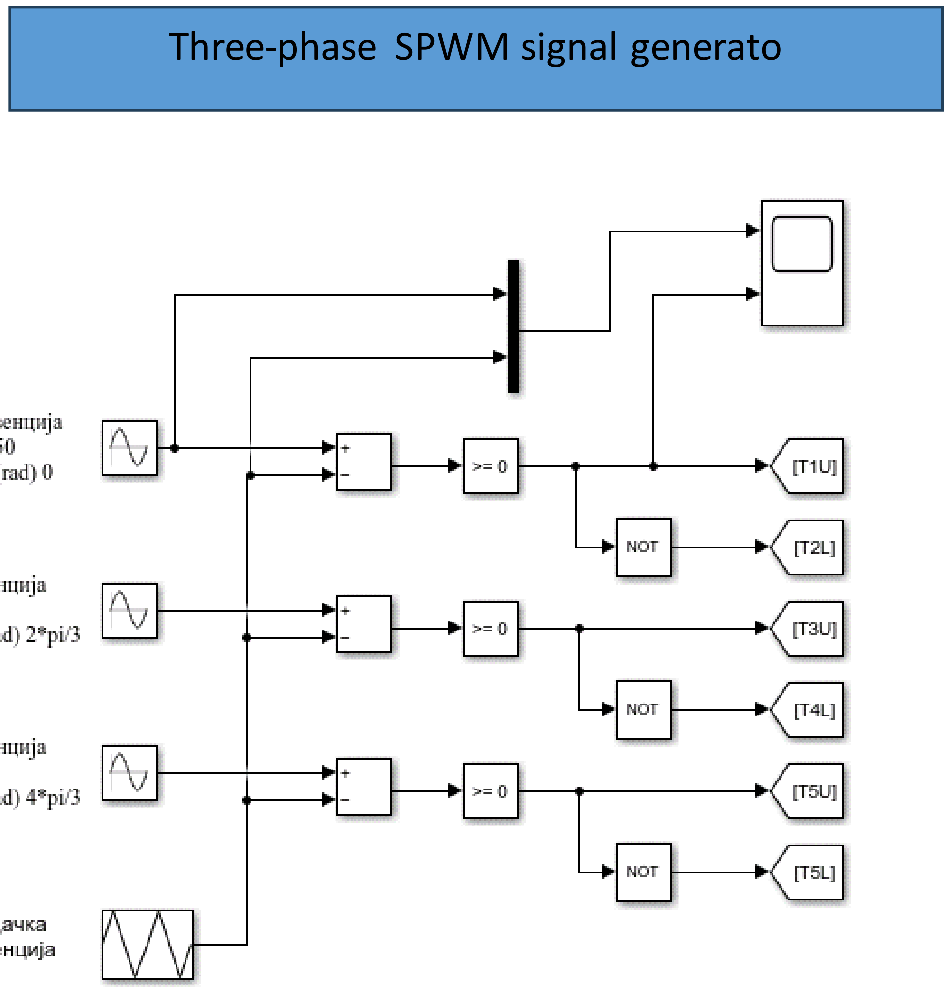


Figure 23. PWM signal generation in MATLAB

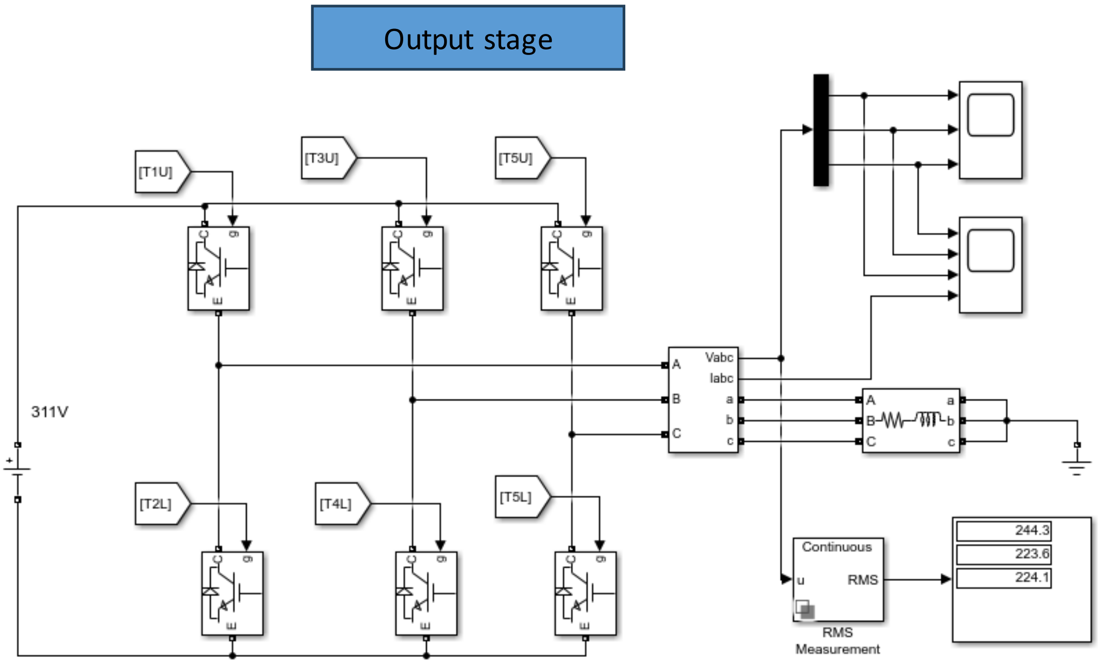
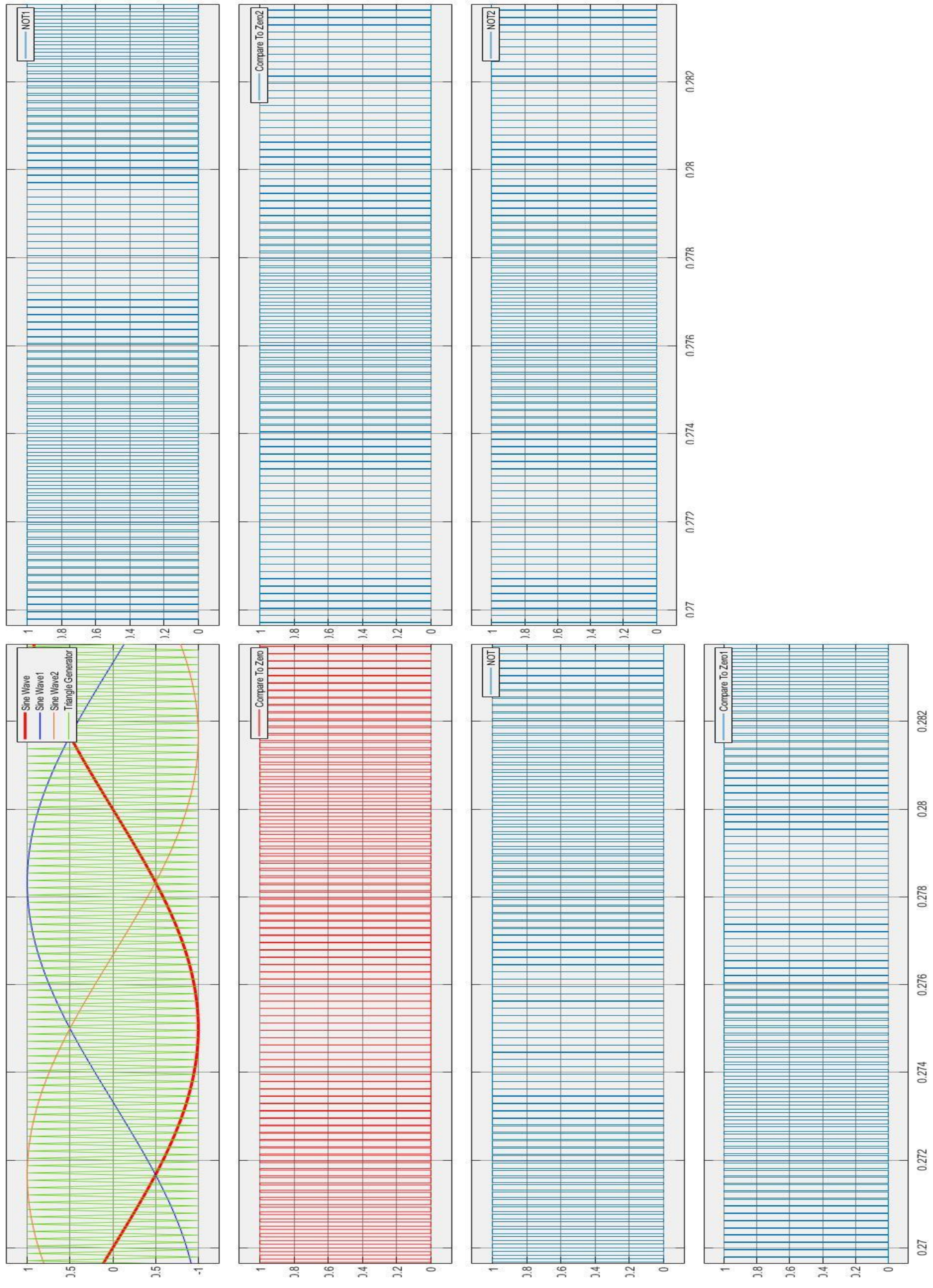


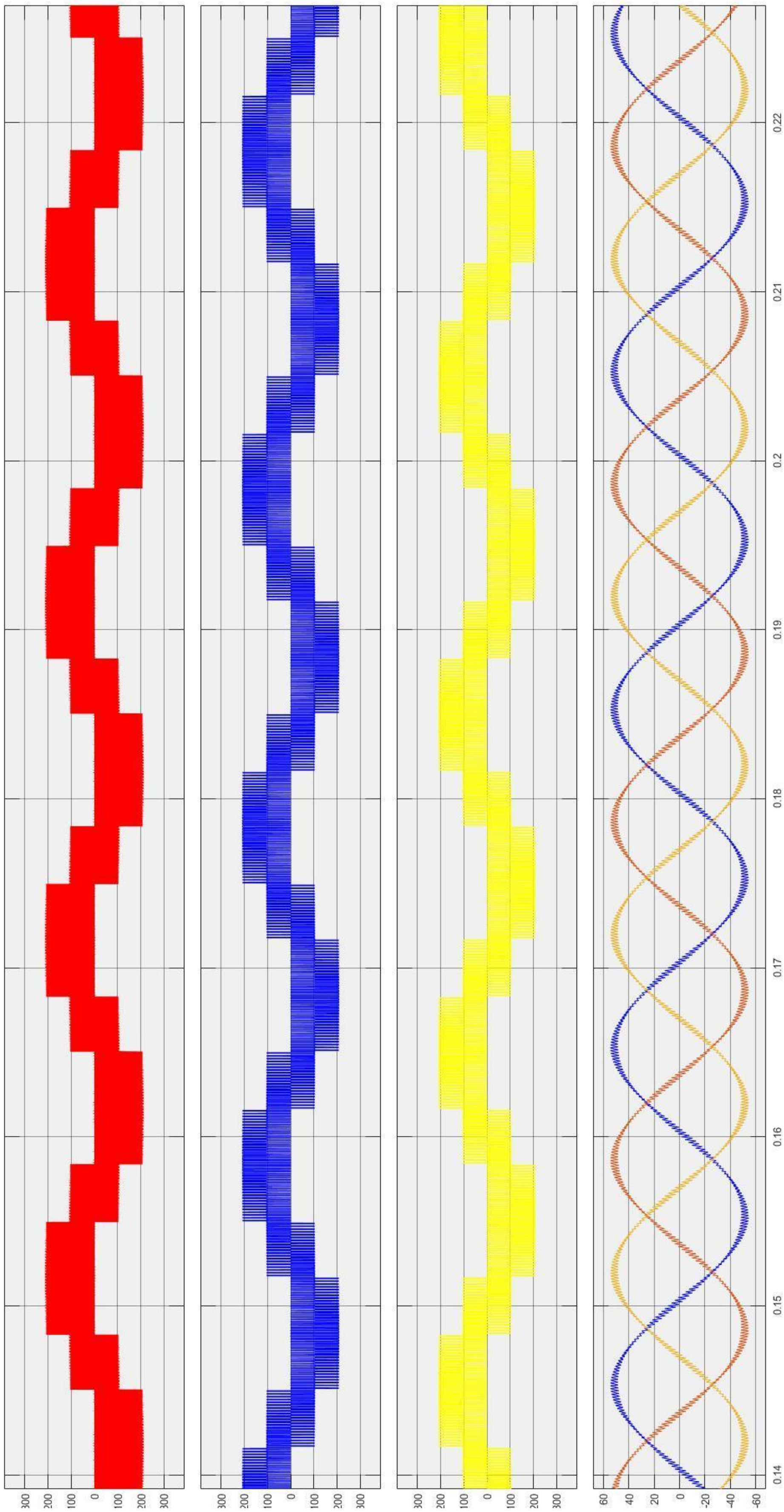
Figure 24. Output stage in MATLAB

8.1 Waveforms of PWM signals

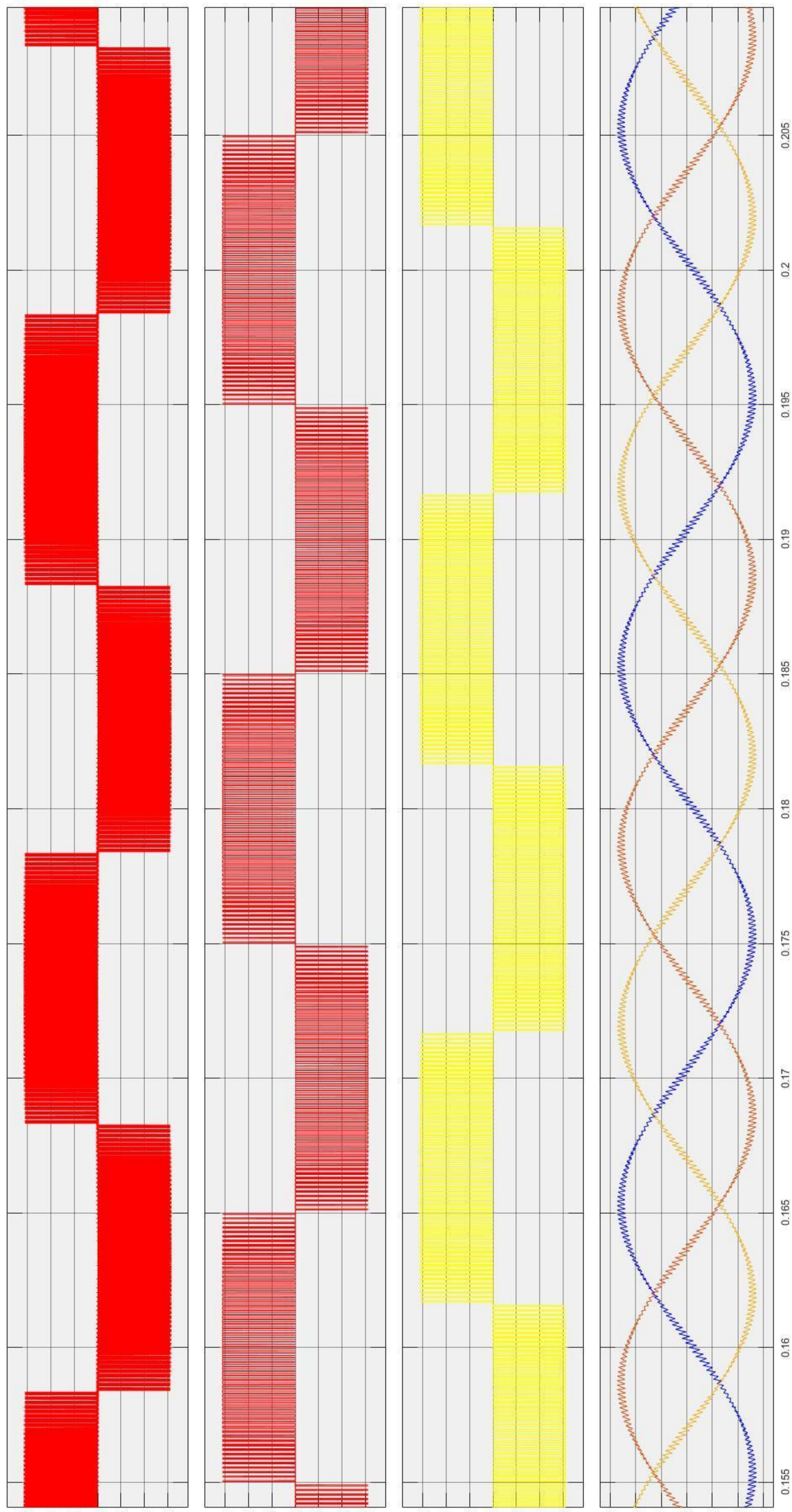
The following graphs show the waveforms of the PWM signal generator as well as the output voltages and currents simulated in MATLAB.



Graph 1. PWM signal waveforms in MATLAB



Graph 2. Output voltage and current waveforms (Phase-to-Ground measurement)



Graph 2. Output voltage and current waveforms (Phase-to-Phase measurement)

9. Conclusion

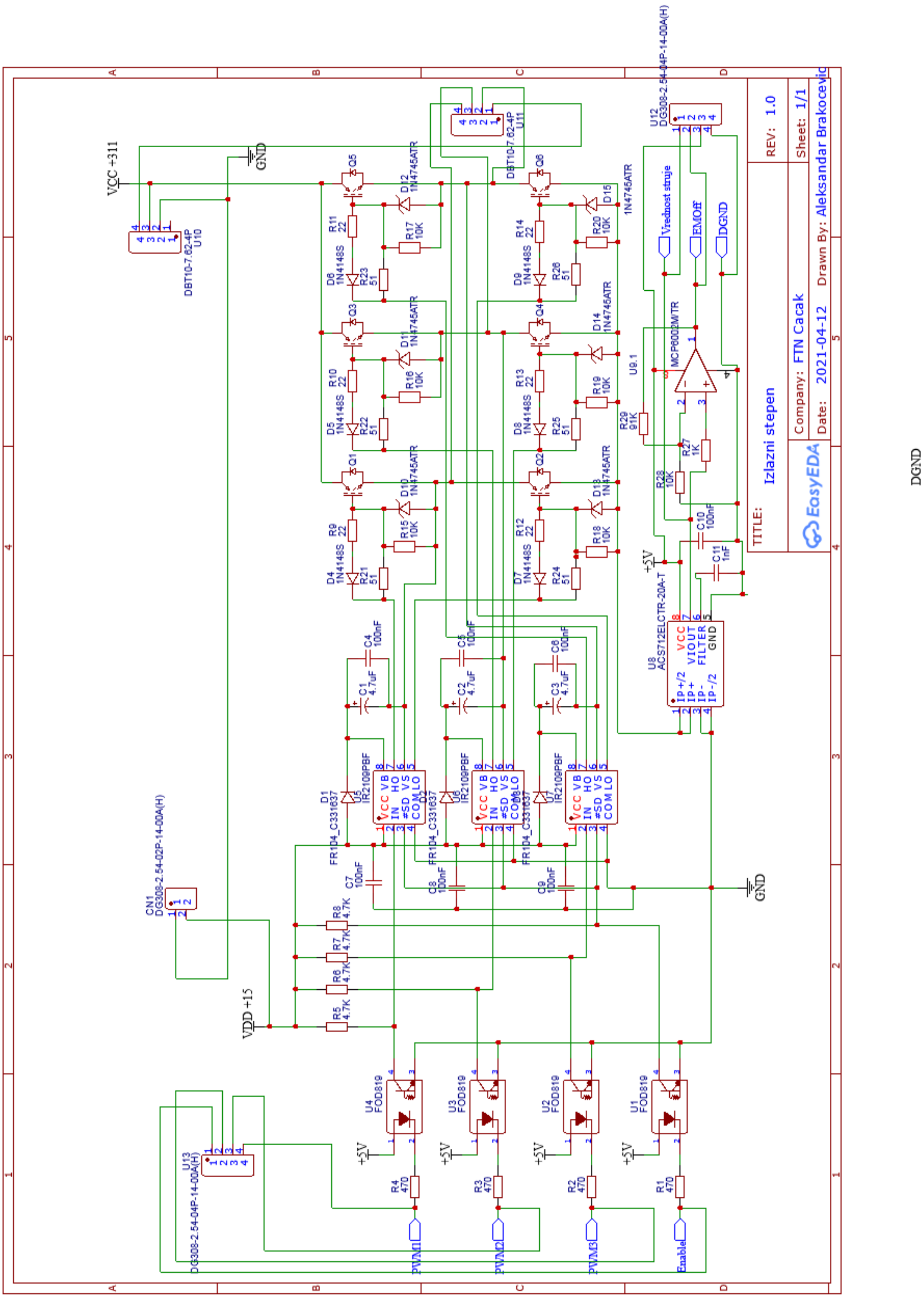
With this work, we have covered only a small part of the vast field of induction machine control. This is the simplest method of controlling a three-phase asynchronous motor. Scalar control of an asynchronous motor using the V/F technique is very easy to implement in practice, inexpensive, and reliable, but it has limitations regarding the applications where the motor can be used. It is most commonly applied in drives such as pumps, fans, and compressors, i.e., where motors operate with a constant torque, and several motors can be connected in parallel and regulated with a single controller. Thanks to the rapid advancement of electronic components and their relatively low cost, it is now much easier to implement more complex control methods of asynchronous motors, such as Space Vector Modulation (SVPWM) or Direct Torque Control (DTC), but these will be discussed in more detail in another work.

10. References

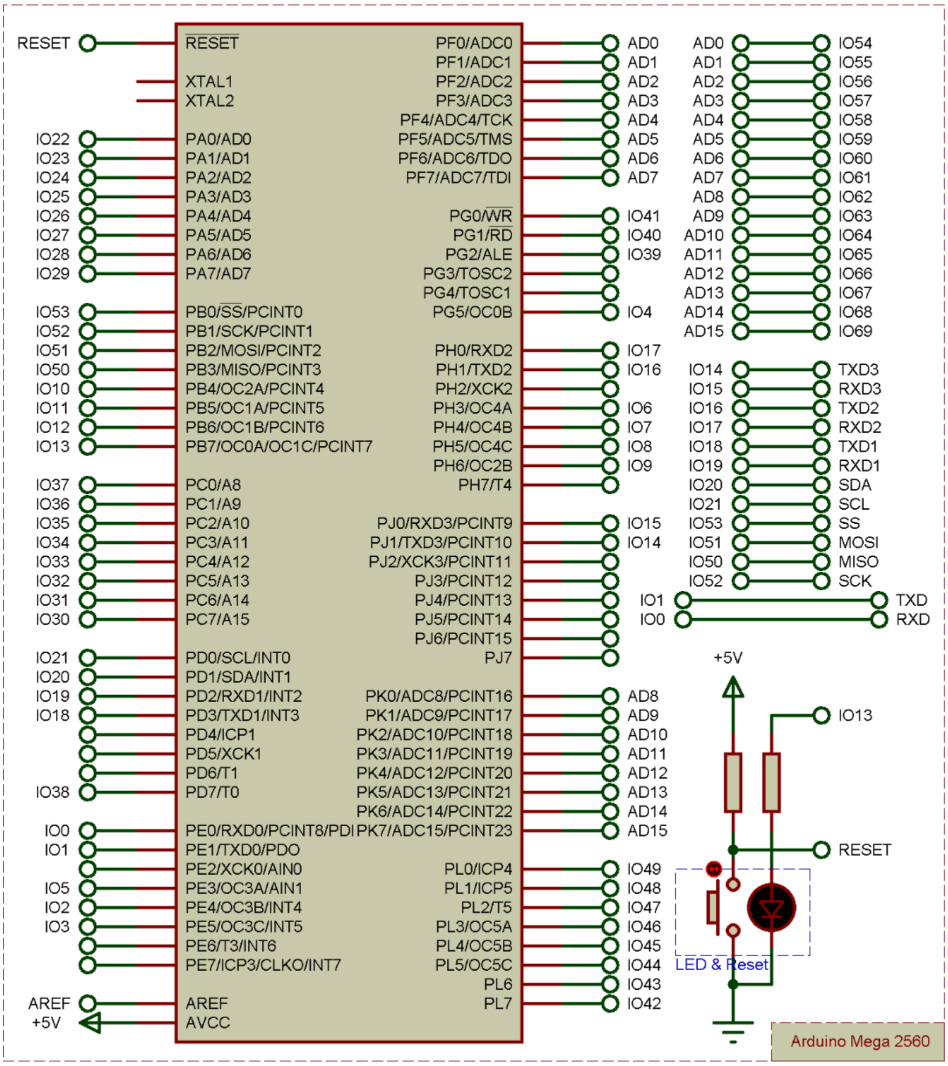
- Predrag B. Petrović – *Power Electronics*
- Radojle Radetić – *Transistor Converters*
- Martin Brown – *Electronics for Motor Drive and Control*
- *Language Reference* – https://www.arduino.cc/reference/en/
- Vladimir Petrović – *Electrical Machines 2*
- www.google.com

11. Appendices

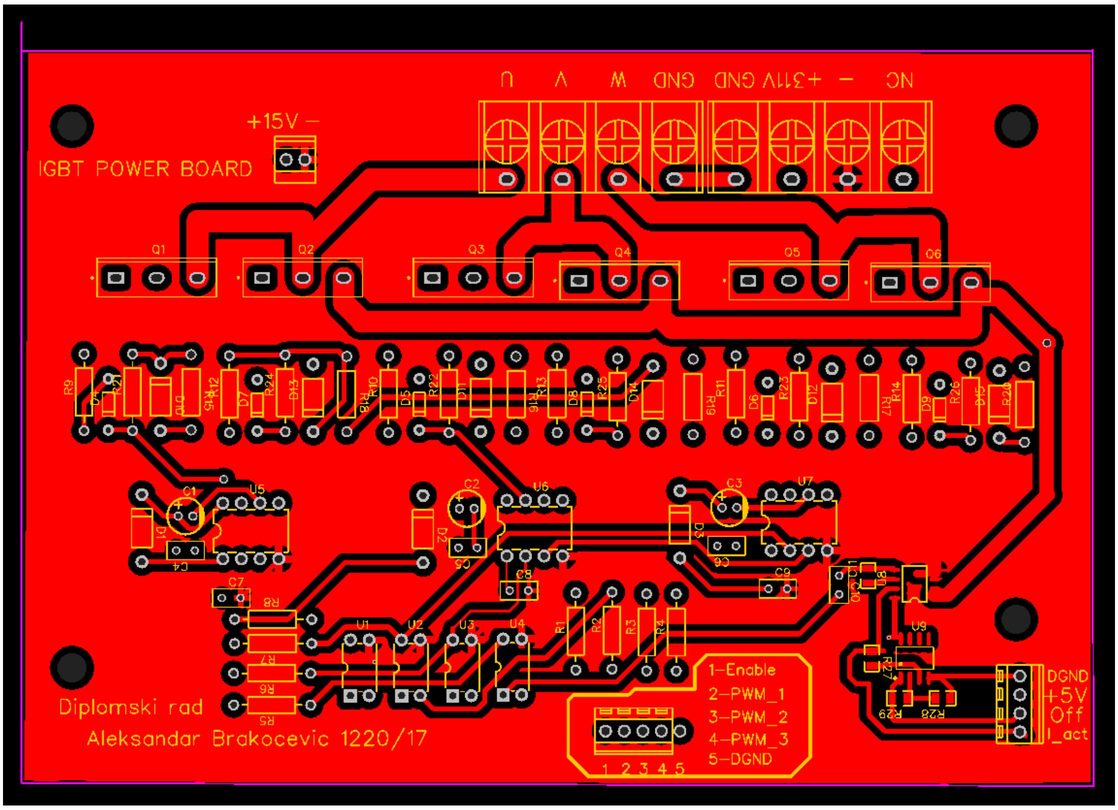
Appendix 1. Output stage schematic



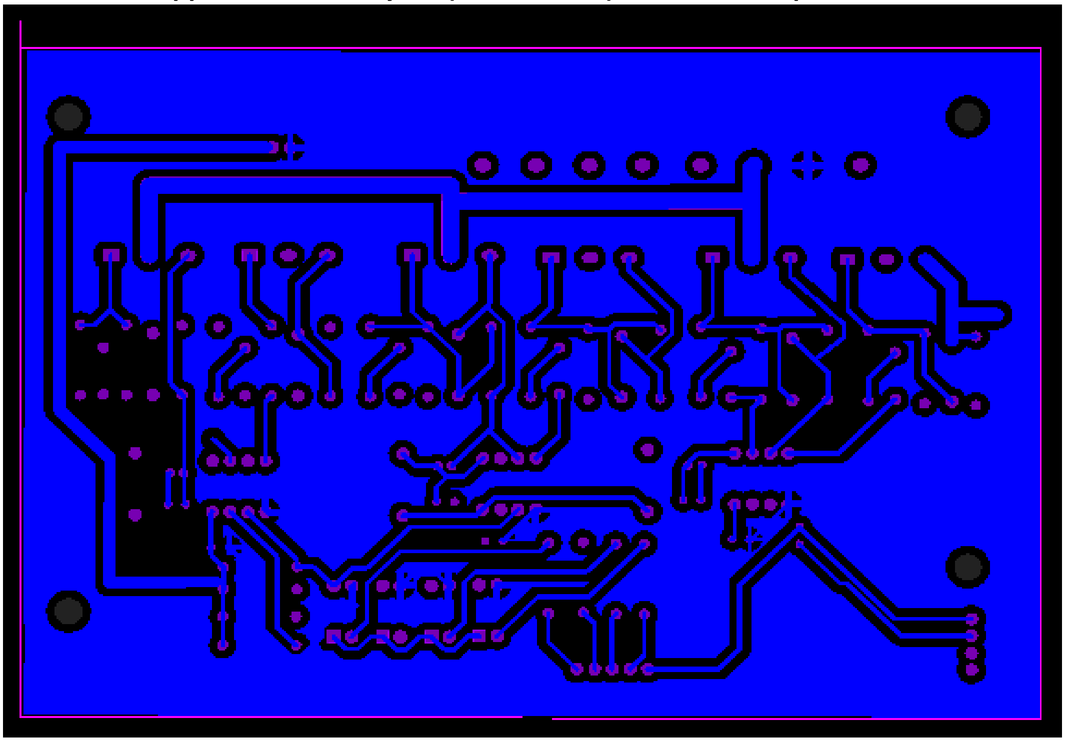
Appendix 2. Arduino Mega 2560



Appendix 4. PCB layout (top side) – inverter output board

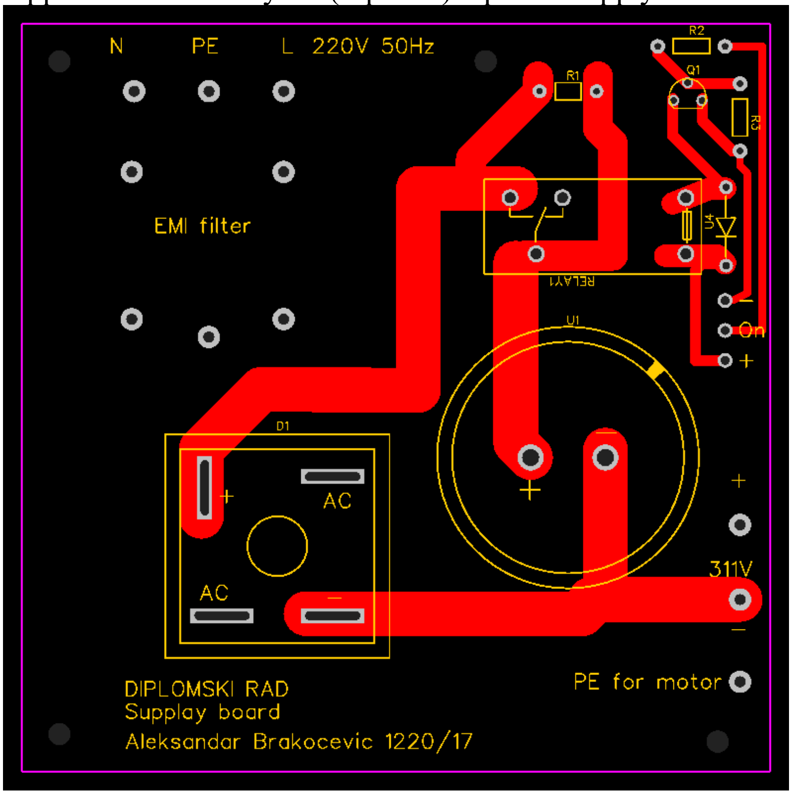


Appendix 4. PCB layout (bottom side) – inverter output board

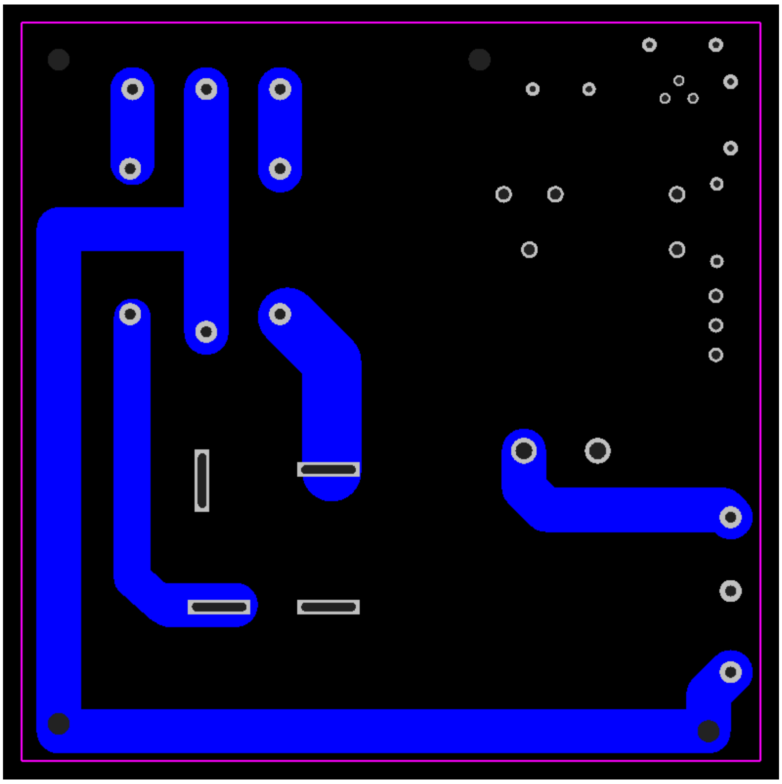


*The Gerber file of the printed circuit boards is included on the attached CD.

Appendix 6. PCB layout (top side) – power supply board*



Appendix 7 . PCB layout (bottom side) – power supply board*



*The Gerber file of the printed circuit boards is included on the attached CD.

Appendix 8. Arduino program listing and Arduino code

It is included on the disk attached to the thesis.

Appendix 9. Technical specifications of Arduino Mega 2560

It is included on the disk attached to the thesis.

Appendix 10. Bill of Materials (BOM)

It is included on the disk attached to the thesis.