International Journal of Frontier Technology and Engineering (IJFTE)
Vol. 03, No. 01, December 2024, pp. 30-43

DC to AC Converter Frequency Speed Drive Mosfet H-Bridge Topology with Fixed Frequency PWM Sine

Henky Kurnia Dhany[1], Budhy Setiawan[1], and Sapto Wibowo[1]

[1] *Departement of Electrical Engineering, Politeknik Negeri Malang, Malang, Indonesia*

Corresponding Author: Henky Kurnia Dhany, henkydhany@gmail.com

Received Date: 27-10-2024 Revised Date: 09-11-2024 Accepted Date: 03-12-2024

Abstract

One way to regulate the rotational speed of an induction motor is to determine the value of its input sine wave frequency, where increasing the frequency of the voltage source given to the induction motor is expected to increase the rotational speed of a single-phase induction motor. This paper focuses on the efforts of researchers to simulate Frequency Speed Drive (FSD) which can regulate changes in sine wave frequency from 2 to 50 Hz with a voltage of 220 Volt RMS. And from the results of the study on the simulator, a frequency of 1.90 to 45.25 Hz was obtained. This study was conducted by simulating electronic circuits and creating firmware that can regulate the speed of the sine frequency using the PWM (Pulse Width Modulation) Chopping technique with a fixed frequency. The chopping method used to produce the SPWM signal is not as usual by comparing the modulation signal with the carrier signal, but simply by changing the sinusoidal voltage value every 1 millisecond to the percentage of the pulse width every 1 millisecond. Because the PWM Chopping frequency is fixed, the lower the sinusoidal frequency, the more PWM Chopping will be produced for one wave. On the other hand, the higher the sinusoidal frequency, the less PWM Chopping will be generated for one wave. With this technique, SPWM waves will be generated with varying frequencies and effective voltage values approaching pure sine waves.

Keywords : Frequency Speed Drive, Fixed Frequency Chopping, Sinusoidal Pulse Width Modulation

1 Introduction

One type of electric motor that is widely used is the 1 phase AC motor. This is because induction motors have advantages compared to other types of electric motors. The advantages of induction motors are that they have relatively high efficiency, they are simple to construct, they have strong resistance, they are easy to maintain and relatively cheap. However, behind the advantages of the induction motor, there are disadvantages to it, one of which is that the rotation speed of the motor cannot be regulated directly, but certain efforts must be made [1], [2] [3], [4]. These include determining the number of poles and regulating the voltage or frequency speed of the AC supply to the motor.

Techniques for controlling the rotational speed of induction motors, both single- and three-phase, have been carried out by several researchers. The researcher [1] applied the Frequency Control method to regulated speed variations in a single phase induction motor, namely, using a PWM IC (SG3525A) to control the required PWM signal frequency. The sinusoidal signal generated from the PWM IC was compared with the carrier signal to generate an SPWM signal. The results of his research showed that by changing the PWM frequency from 16 to 56 Hz, the motor rotated at 480 to 1680 rpm. The researcher [2] used an ATmega328 microcontroller to generate SPWM signals. The sinusoidal reference signal was compared with a triangular carrier signal to generate a PWM signal with a varying pulse width that was identical to the amplitude of the sinusoidal signal. The result obtained was that the 42 V single-phase induction motor tested produced different rpms for the frequency range of 10-50 Hz. The researchers [3] generated modulation waves digitally using a microcontroller. The frequency of the modulation wave could be changed by changing the parameters

in the algorithm that controls the ratio between the carrier wave and the modulation wave. The researcher [4] had generated PWM (Pulse Width Modulation) and SPWM (Sinusoidal Pulse Width Modulation) signals from Arduino ATmega2560 that were directly connected to Matlab / Simulink. The modulation process was carried out by comparing the amplitude of the sinusoidal reference signal with the triangular carrier signal in the MATLAB/Simulink environment. The researcher [5] compared the results of laboratory practice and simulations to control the speed of a single-phase induction motor of a capacitor starter, where there was a possibility that the simulation parameters did not match the real conditions in the laboratory, so several different results were obtained.

To produce alternating voltage, an inverter is needed [6], [7], [8]. An inverter is needed to change DC to AC voltage, with frequency and voltage according to the logic signal from the controller [9], [10]. The researcher tested the effectiveness of single-phase Half-Bridge and Full-Bridge voltage source inverters (VSI) using Arduino Uno and Ni-MyRIO. Where Half-Bridge was generally used in simpler applications with fewer variable loads, Full-Bridge was used in applications that require high performance such as controlling induction motors in variable drive systems. The study showed that full-bridge is more suitable for industrial applications that require high-quality AC output and better control.

The aim of the research is to produce a Frequency Speed Drive (FSD) to regulate the sine voltage frequency. The higher the frequency of the induction motor input voltage source, the faster the rotation of the single phase induction motor will. Based on the background above, the author conducted this research by designing firmware and simulating electronic circuits to control the speed of SPWM wave frequency. The signal is generated by programming the MCU simulator and ESP32 to compare the results. The chopping method to generate SPWM signals in this study is not like the usual one by comparing the modulation signal with the carrier signal, but simply by changing the sinusoidal voltage value every 1 millisecond to the percentage of the pulse width in every 1 millisecond. Because the PWM Chopping frequency is fixed, the lower the sinus frequency produced, the more PWM Chopping occurs for one wave, and vice versa.

2 Method

2.1 System Block Diagram

The system that will be created has a way of working by reading a voltage of 0-5 volts at the input of the MCU (Microcontroller Unit) to convert it into a frequency of 2-50 Hz. In the MCU program, the sine wave produced comes from PWM chopping starting with 20 sections for an SPWM frequency of 50 Hz. The amount of PWM chopping increases as the Sinusoidal Pulse Width Modulation (SPWM) frequency decreases. Control logic to determine clockwise (CW) and counterclockwise (CCW) signals via the H-Bridge Mosfet circuit.

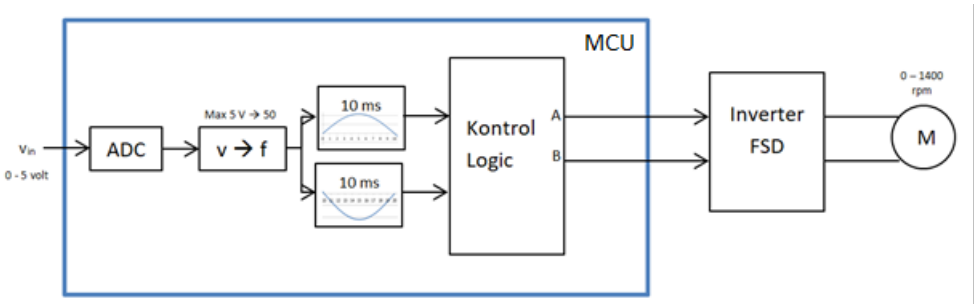


Figure 1: Block Diagram of the Frequency Speed Drive (FSD) System

In Figure 1, when the Vin voltage is 5 volts, the SPWM frequency of 50 Hz is equal to a period of 20 ms, 10 ms for the positive cycle, and 10 ms for the negative cycle. One 50 Hz sine wave will be chopped in 20 parts, so the chopping time is 1 ms (PWM frequency 1 kHz). If the SPWM frequency decreases, the PWM chopping frequency is kept constant, so the amount of chopping will be greater. The Frequency Speed Drive inverter is an H-Bridge Mosfet circuit.

2.2 FSD Inverter Topology

The input of the system is a voltage of 0-5 Volts through a potentiometer. The MCU (Microcontroller Unit) system forms a Sinusoidal Pulse Width Modulation (SPWM) signal with a frequency of 2-50 Hz on pins 5 and 6 which is linear with changes in the input voltage. The SPWM signal from the MCU is useful for triggering 4 Mosfet H-Bridge gates via the optocoupler. The SPWM signal with a voltage of 310 Volts is generated by the H-Bridge Mosfet to control the rotation of a single-phase induction motor. The Frequency Speed Drive (FSD) circuit drawn using the Proteus program is shown in Figure 2.

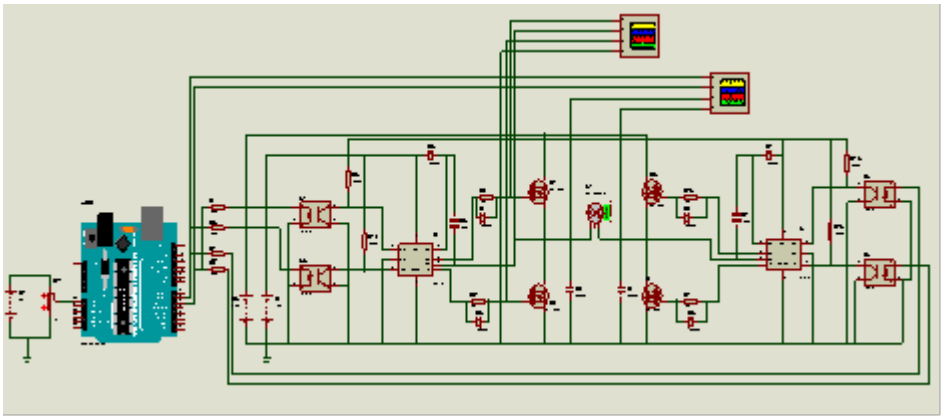


Figure 2: Frequency Speed Drive (FSD) Test Circuit

2.3 SPWM Chopping

If the maximum operating frequency of the motor is 50 Hz, then the minimum period is 0.02 seconds (20 ms). Because,

$$T = \frac{1}{f} \tag{1}$$

$T = \frac{1}{50} = 0.02 seconds = 20 ms$

The positive cycle of the AC signal is at least 10 ms and the negative cycle is also 10 ms. The chopping time is of the period , or kHz. This chopping time is at a frequency of 50 Hz, and will remain the same when the frequency is lowered, so the amount of chopping will increase. The ac voltage value at any time can be formulated in Equation 2.

$$v_t = V_{max} sin(\omega t) \tag{2}$$

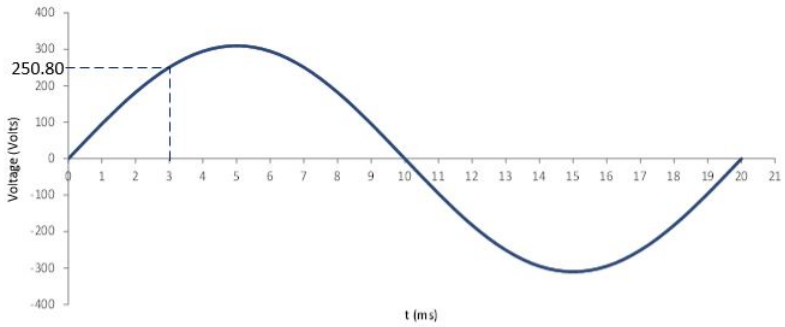
The voltage is the voltage value at time dan is the maximum voltage of the sinusoidal wave. If the maximum voltage is 310 volts, then
$v_t = 310 sin(\omega t) = 310 sin(2\pi f t) = 310 sin(\frac{2\pi}{T} t$
Because T = 20 ms, then
$v_t = 310 sin(\frac{2\pi}{20} t) = 310 sin(2\pi(0.05)t) = 310 sin(18.t)$
Where t in milliseconds and π radians equal to 180° And $\omega = 18$ degrees/millisecond.

If the chopping period is 1 ms or the PWM chopping frequency is 1 kHz, then a table can be made to form a wave with a frequency of 50 Hz as shown in Table I.

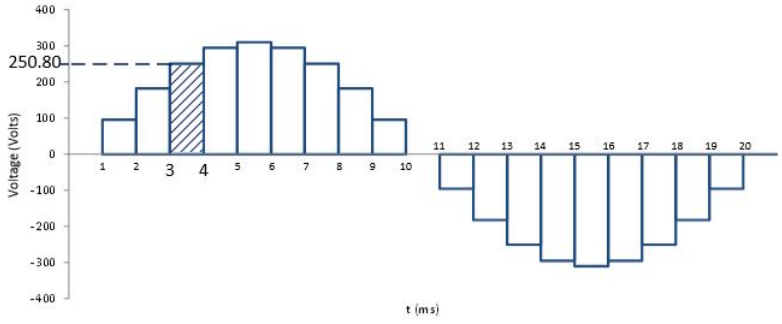
Table 1: Sinusoidal Wave Chopping with a Frequency of 50 Hz

2π(degrees)	f(Hz)	ω	t (ms)	ω.t (degrees)	V(t)(volts)	PWM duty cycle (%)
			0	0	0.00	0
			1	18	95.80	31
			2	36	182.21	59
			3	54	250.80	81
			4	72	294.83	95
			5	90	310.00	100
			6	108	294.83	95
			7	126	250.80	81
			8	144	182.21	59
			9	162	95.80	31
360	50	18	10	180	0.00	0
			11	198	-95.80	-31
			12	216	-182.21	-59
			13	234	-250.80	-81
			14	252	-294.83	-95
			15	270	-310.00	-100
			16	288	-294.83	-95
			17	306	-250.80	-81
			18	324	-182.21	-59
			19	342	-95.80	-31
			20	360	0.00	0

In Table 1, it is found that the higher the sinusoid value (v_t), the duty cycle of the PWM (Pulse Width Modulation) signal is also wider. Because the duty cycle width changes according to the sinusoid value, the wave is called SPWM (Sinusoidal Pulse Width Modulation). The minus sign in column 7 (PWM duty cycle) indicates a negative cycle of SPWM.



(a)



(b)

Figure 3: (a) Graph of the sine function, (b) Chopping every 1 ms

In Figure 3a, the sinusoidal voltage value in the third millisecond is 250.80 volts. In Figure 3b, the voltage value at the third millisecond is 250.80 volts until the fourth millisecond. This can be made into a rectangular area. Based on Equation 2, which is a basic sinusoidal equation, we can determine the area of pulse width modulation (PWM) in the form of an integral equation for each cutting, namely every 1 ms. The form of the integral equation can be written in Equation (3).

$$pwm_t = (\int_t^{(t+\frac{v_t}{V_{max}}} V_{max}, dt) \tag{3}$$

From the integral equation, it can be seen that PWM (pulse width modulation) is a right-angled rectangular area with a height of . The length of the right-angled rectangle is the value of the duty cycle where the value is the same as . The area of the right-angled rectangular from equation 3 is shown in Figure 4.

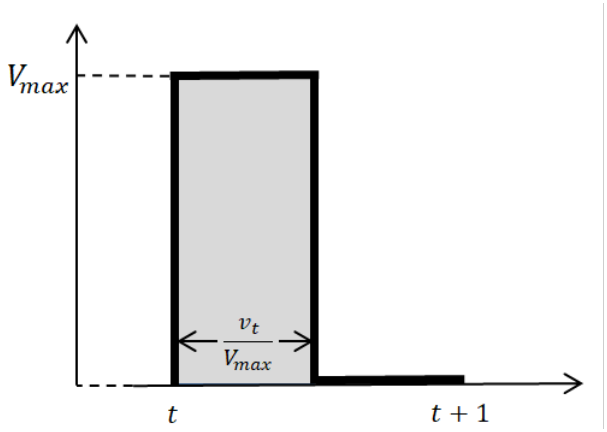


Figure 4: Area of a right-angled rectangular from the integral equation

If there is a sinusoidal wave with a maximum amplitude
$V_{max} = 310volt, v_t = 310sin(2\pi.(0.05)t)$.
If t=3, then the v_t value is 250.80 volts, with the following explanation.
$v_3 = 310sin(2\pi(0.05)(3))$
$= 310sin(2\pi(0.15)) = 310sin(0.3.\pi) = 310x0.81 = 250.80volts$
The v_t value is equal to 250.80 volts, which is the voltage value of the sinusoidal wave when t=3.

If the sinusoid equation above is inserted into equation 3 which in this study is stated as the PWM (pulse width modulation) equation, it can be explained as follows.

$$pwm_3 = (\int_3^{(3+\frac{250.80}{310}} 310dt) = (\int_3^{3.81} 310, dt) = 310x(3.81-3) = 250.80volts \tag{4}$$

The value is equal to 250.80 volts, this is the area of a right-angled rectangle with a height of 310 and a length of 0.81 or a duty cycle of 81%. This value is obtained when , as in the shaded graph in Figure 6. And so on for every t-th chopping. Obtained the similarity of the voltage value (v_t) from the sinusoid equation (equation 2) with the area value from equation (equation 3). From the information above, the calculation results to obtain a 50 Hz Sinusoidal Pulse Width Modulation (SPWM) signal with chopping every 1 ms are presented in Table II below.

Table 2: SPWM generator with 1 ms chopping time

pulse to	duty cycle (%)	t_{on}(ms)	T_{off} (ms)	t start ON	t finish ON /t start OFF	t finish OFF
0	0	0.00	1.00	0.00	0.00	1.00
1	31	0.31	0.69	1.00	1.31	2.00
2	59	0.59	0.41	2.00	2.59	3.00
3	81	0.81	0.19	3.00	3.81	4.00
4	95	0.95	0.05	4.00	4.95	5.00
5	100	1.00	0.00	5.00	6.00	6.00
6	95	0.95	0.05	6.00	6.95	7.00
7	81	0.81	0.19	7.00	7.81	8.00
8	59	0.59	0.41	8.00	8.59	9.00
9	31	0.31	0.69	9.00	9.31	10.00
10	0	0.00	1.00	10.00	10.00	11.00
11	-31	0.31	0.69	11.00	11.31	12.00
12	-59	0.59	0.41	12.00	12.59	13.00
13	-81	0.81	0.19	13.00	13.81	14.00
14	-95	0.95	0.05	14.00	14.95	15.00
15	-100	1.00	0.00	15.00	16.00	16.00
16	-95	0.95	0.05	16.00	16.95	17.00
17	-81	0.81	0.19	17.00	17.81	18.00
18	-59	0.59	0.41	18.00	18.59	19.00
19	-31	0.31	0.69	19.00	19.31	20.00
20	0	0.00	1.00	20.00	20.00	21.00

In Table 2, when the pulse increases to 0, the duty cycle = 0%, so the pwm pulse is OFF for 1 ms. At the first pulse, duty cycle = 31%, the pulse of pwm is ON for 0.31 ms, where it is ON when t = 1 ms to t = 1.31 ms and OFF when t = 1.31 ms to t = 2 ms. During the second pulse, duty cycle = 59%, the pwm pulse is ON for 0.59 ms, where it is ON when t = 2 ms to t = 2.59 ms and OFF when t = 2.59 ms to t = 3 ms. And so on, so that a graph can be produced as in Figure 5 below.

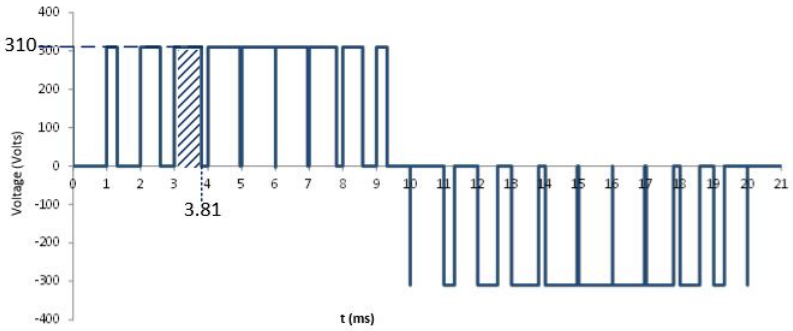


Figure 5: SPWM wave calculation results

2.4 Effective Voltage of Sinusoidal Pulse Width Modulation (SPWM)

Effective voltage is a voltage that provides an energy value equal to the energy value of the direct voltage, which is formulated in Equation 4.

$$V_{RMS} = \sqrt{\frac{1}{T}\int_0^T v_t^2 dt} = \frac{V_{max}}{\sqrt{2}} \tag{5}$$

If there is a sinusoidal wave with a maximum amplitude 310 volts, then 219.20 volts. The calculation of the effective voltage value for SPWM frequency 50 Hz with a PWM chopping frequency of 1 kHz (chopping period 1 ms) is shown in Table 3.

Table 3: Calculation of effective voltage SPWM 50 Hz

T_{PWM}	d(On)	310.d(On)	$(310.d(On))^2$
0	0.00	0.00	0.00
1	0.31	95.80	9176.73
2	0.59	182.21	33201.73
3	0.81	250.80	62898.27
4	0.95	294.83	86923.27
5	1.00	310.00	96100.00
6	0.95	294.83	86923.27
7	0.81	250.80	62898.27
8	0.59	182.21	33201.73
9	0.31	95.80	9176.73
10	0.00	0.00	0.00
11	0.31	95.80	9176.73
12	0.59	182.21	33201.73
13	0.81	250.80	62898.27
14	0.95	294.83	86923.27
15	1.00	310.00	96100.00
16	0.95	294.83	86923.27
17	0.81	250.80	62898.27
18	0.59	182.21	33201.73
19	0.31	95.80	9176.73
20	0.00	0.00	0.00
$\sum_0^\infty (310 - d(ON))^2 =$			45761.90
$V_{RMS}=$			213.92

Effective voltage values for SPWM frequencies of 50, 40, 30, 20, and 10 Hz are shown in Table 4 below.

Table 4: SPWM effective voltage 10 to 50 Hz

Frequency (Hz)	V_{RMS} (volt)
50	213.92
40	214.95
30	214.02
20	217.04
10	218.12

Table 4 shows that the lower the SPWM frequency, the closer the effective voltage value is to the calculated pure sinusoidal voltage value, namely 219.20 volts.

2.5 Design of the Sinusoidal Pulse Width Modulation (SPWM) Program

The steps in the method for forming sinusoidal pulse width modulation (SPWM) can be seen in the flow chart in Figure 6. The flow chart in Figure 6 explains how to change the sinusoidal wave equation into Pulse Width Modulation (PWM) form with a duty cycle that is linear to the amplitude of the sinusoidal wave, which is chopped every 1 ms. The voltage value of the input given to the microcontroller is used to determine the period value (T). The sinusoid value (y) changes with changes in time (t) every 1 ms (chopping time) or at a frequency of 1 kHz. If the value of $y > 1$ is satisfied, then output pinA (or, as shown in Figure 3, pin 5 of the MCU) will be activated (active LOW) for y microseconds, then inactive (HIGH) for 1000 - y microseconds, etc. The value of y will change as long as t increases to t + 1. However, if the value of $y < -0.1$, then output pinB (or if in Figure 3 pin 6 of the MCU is shown) will be active (low active) for y microseconds and then inactive (HIGH) for 1000 - y microseconds. And if $-0.1 \leq y \leq 0.1$ pinA and pinB are inactive for 5 microseconds. And then when time exceeds the period value T ($t > T$), it is reset to zero again. This branching of the program aims to allow the system to complete generating one SPWM signal cycle even if the value changes before one sinusoid cycle is completed. This rule is expected to convert one sinusoidal wave cycle into a sinusoidal pulse width modulation (SPWM) signal.

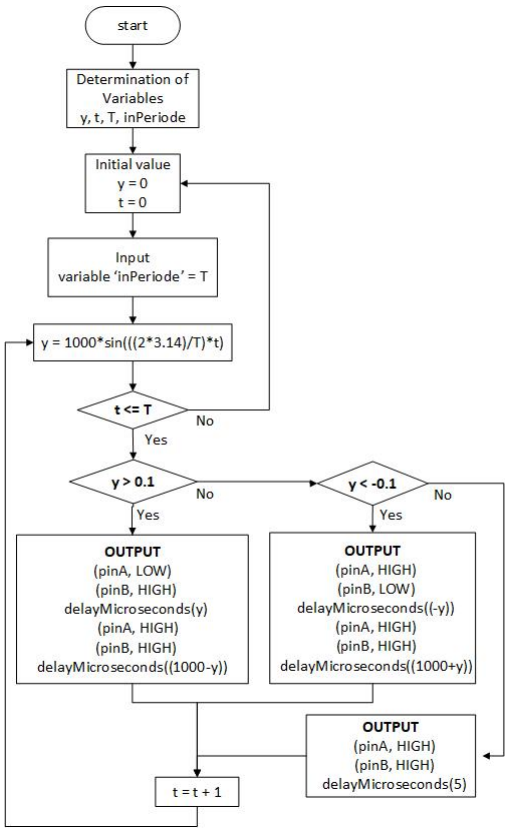


Figure 6: Flow chart for creating the SPWM program

3 Results and Discussion

The circuit is simulated using Simulation Software to see the output signal in several parts.

3.1 MCU output logic signal

The output of the SPWM logic signal from the MCU converter circuit in the simulator has a voltage of 0 to 3.95 volts, active LOW logic, and a period of 22.10 milliseconds.

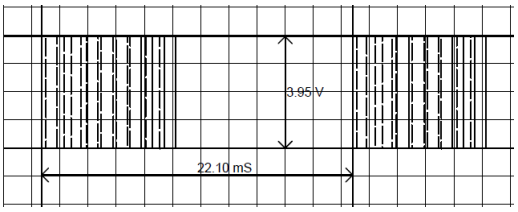


Figure 7: MCU output logic signal

Figure 7. This signal is the output of MCU pin 5, with a period setting in the firmware of 20 milliseconds (or a frequency of 50 Hz). There is a difference of 2.10 milliseconds between the settings in the firmware and the MCU output, this is because the simulator requires execution time to run the program. The SPWM logic signal from the ESP32 has a voltage of 0 to 3.40 volts, and a period of 21.20 milliseconds. Shown in Figure 8.

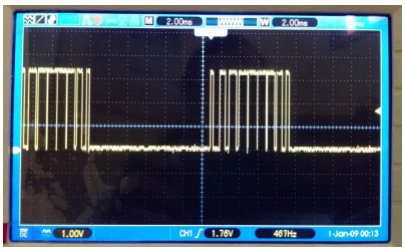


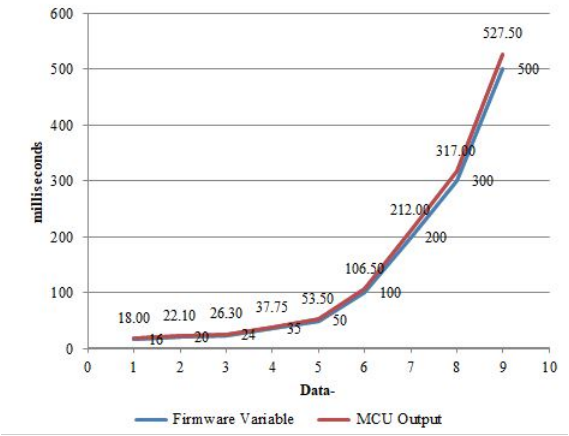
Figure 8: ESP32 output logic signal

The SPWM logic signal is generated by the firmware on the MCU. In the test, the period and frequency of the SPWM output from the MCU output pin were observed. The test results using simulator software are shown in Table 5 by comparing the period size of a variable in the firmware with the oscilloscope display on the MCU output side.

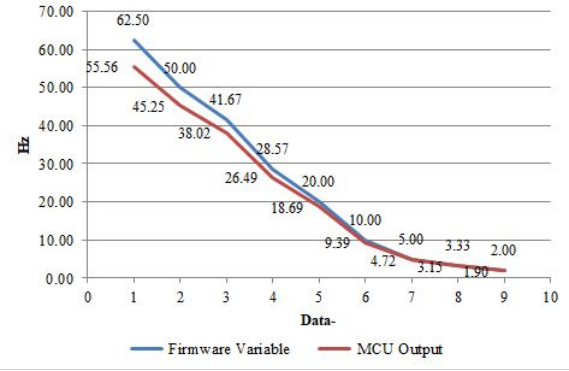
Table 5: Comparison of period and frequency between firmware variables and MCU output from software simulation results.

No	Period (milliseconds)			Frequency (Hz)		
	Firmware Variables	MCU outputs	Error (%)	Firmware Variables	MCU outputs	Error (%)
1	16	18.00	12.50	62.50	55.56	12.50
2	20	22.10	10.50	50.00	45.25	10.50
3	24	26.30	9.58	41.67	38.02	9.58
4	35	37.75	7.86	28.57	26.49	7.86
5	50	53.50	7.00	20.00	18.69	7.00
6	100	106.50	6.50	10.00	9.39	6.50
7	200	212.00	6.00	5.00	4.72	6.00
8	300	317.00	5.67	3.33	3.15	5.67
9	500	527.50	5.50	2.00	1.90	5.50

Table 5 shows the test results by changing the period or frequency value in the firmware variable and then observing the period or frequency of the MCU output in the simulator. There is a difference (error) of 5. 50% to 12. 50%, where the longer the signal period, the smaller the error level. The lower the signal frequency, the smaller the error level. Figure 10 is a graph of the period and frequency of the firmware variables and the MCU output.



(a)



(b)

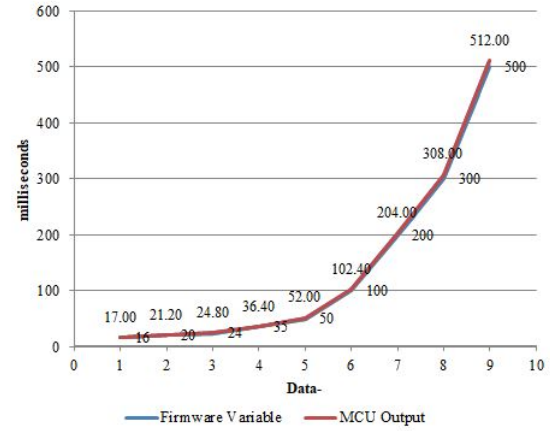
Figure 9: (a) SPWM period and (b) SPWM frequency in the simulator

In Figure 9, the test results show comparable changes in period (Figure 9a) and frequency (Figure 9b) between the firmware variables and the output of the MCU.

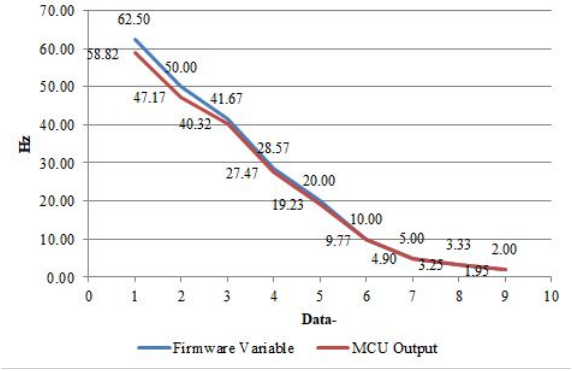
Table 6: Comparison of period and frequency between firmware variables and MCU output from ESP32.

No	Period (milliseconds)			Frequency (Hz)		
	Firmware Variables	MCU outputs	Error (%)	Firmware Variables	MCU outputs	Error (%)
1	16	17.00	6.25	62.50	58.82	6.25
2	20	21.20	6.00	50.00	47.17	6.00
3	24	24.80	3.33	41.67	40.32	3.33
4	35	36.40	4.00	28.57	27.47	4.00
5	50	52.00	4.00	20.00	19.23	4.00
6	100	102.40	2.40	10.00	9.77	2.40
7	200	204.00	2.00	5.00	4.90	2.00
8	300	308.00	2.67	3.33	3.25	2.67
9	500	512.00	2.40	2.00	1.95	2.40

The test results using ESP32 are shown in Table 6. Table 6 also shows the test results by changing the period or frequency value in the firmware variable and then observing the period or frequency in the ESP32 output. There is a difference (error) of 2. 40% to 6. 25%, where the longer the signal period, the smaller the error level, or the smaller the signal frequency, the smaller the error level. Figure 10 is a graph of the firmware variable period and ESP32 output.



(a)



(b)

Figure 10: (a) SPWM period and (b) SPWM frequency output from ESP32

3.2 H-Bridge Mosfet Output SPWM Signal

The H-Bridge Mosfet output SPWM signal without load from the Mosfet H-Bridge is presented in Figure 11. The SPWM signal has a voltage of 310 volts according to the dc voltage source.

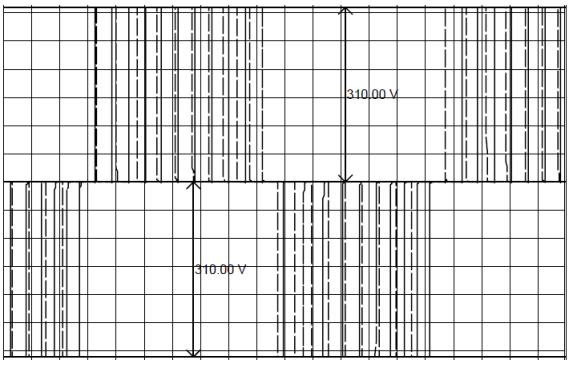


Figure 11: H-Bridge Mosfet output SPWM signal

3.3 H-Bridge Mosfet output signal

The output signal of the H-Bridge Mosfet, which is loaded with an induction motor and capacitor filters, is presented in Figure 12. The sinusoidal signal formed at the input of the induction motor has a voltage of 592.50 V_{pp} (or 296.25 V_p). There is a voltage drop for the N-Mosfet on the drain source side, so that the maximum output voltage is below the source voltage (310 V_{dc}).

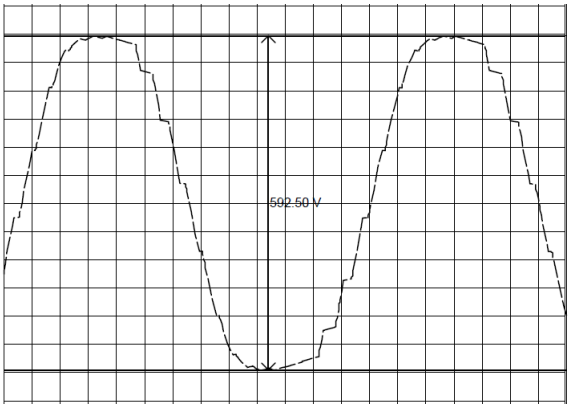


Figure 12: Signal at the input of an induction motor

3.4 PWM Chopping Period

In figure 13, the PWM chopping period in testing shows 1.05 milliseconds. In the firmware, this period is expected to be 1 millisecond (1 kHz frequency), so there is a difference of 0.05 milliseconds (5% error). This is because the simulator requires execution time to run the program. Here, the MCU clock frequency in the simulator is 16 MHz.

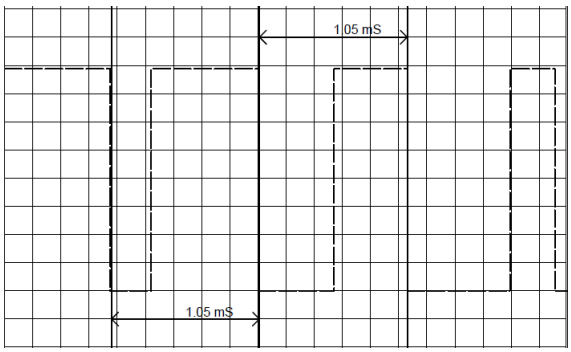


Figure 13: Chopping period in the simulator

In the ESP32 shown in Figure 14, the PWM chopping period in the test shows 1040 micro seconds (1.04 milli seconds). In the firmware, this period is expected to be 1 millisecond (1 kHz frequency), so there is a difference of 0.04 milliseconds (4% error).

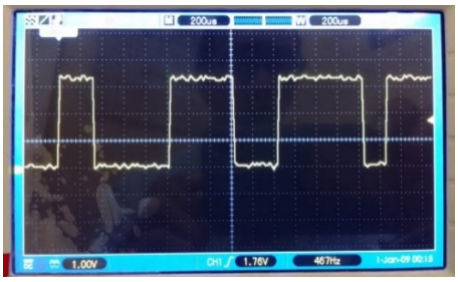


Figure 14: Chopping period on ESP32

4 CONCLUSION

Based on the calculation results in the Research Method, it is shown that the lower the SPWM frequency, the closer the effective voltage value is to the calculation of pure sinusoidal voltage, namely 219.20 volts. This is because if the SPWM frequency is lower, the amount of chopping will increase.

Based on the test results, the firmware can produce an SPWM logic signal with a chopping period of 1.05 milliseconds on the MCU simulator output pin where the firmware setting is 1 millisecond (1 kHz frequency); this is because the simulator requires execution time to run the program. Meanwhile, ESP32 produces almost the same chopping period, namely 1.04 milliseconds. There were also comparable changes in period and frequency between the firmware variables and the output of the MCU simulator with a difference of 5.50% to 12.50%. Meanwhile, using ESP32 there is a difference of 2.40% to 6.25%. In the simulator, the SPWM signal without load from the H-Bridge MOSFET produces a voltage of 310 Vp. If the system uses an induction motor load with a capacitor filter, it produces a sinusoidal wave with a voltage of 592.5 Vpp or 296.25 Vp. Since research is limited to simulation only, it is expected that further research can be tested by making hardware. The use of ESP32 in this study is only limited to firmware testing.

References

[1] A. Z. Latt and N. N. Win, "Variable speed drive of single phase induction motor using frequency control method," in *2009 International Conference on Education Technology and Computer (ICETC 2009)*, 2009, pp. 30–34.

[2] B. Kafle, P. Basnet, A. Ghimire, and A. Lamichhane, "Speed control of a single phase induction motor using the v/f method," in *2019 Innovations in Power and Advanced Computing Technologies (i-PACT 2019)*, 2019, pp. 1–4.

[3] M. Naveenkumar, A. Munjal, S. Srinivasan, and D. Prasad, "Design and implementation of a variable frequency drive for single-phase induction motor," in *2015 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE 2015)*, 2016, pp. 239–242.

[4] S. P. Biswas, M. K. Hosain, M. R. I. Sheikh, M. F. Kibria, F. Hasan, and M. Y. Y. U. Haque, "A noble approach for generating real time firing pulse for inverter using arduino and matlab/simulink," in *4th International Conference on Electrical Engineering and Information Communication Technology (iCEEiCT 2018)*, 2018, pp. 662–665.

[5] N. A. Daw, "Comparison of lab work and simulation results for speed control of single phase induction motor capacitor starting," in *2016 17th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, 2017, pp. 397–402.

[6] S. Mahmodicherati, N. Ganesan, L. Ravi, and R. Tallam, "Application of active gate driver in variable frequency drives," in *2018 IEEE Energy Conversion Congress and Exposition (ECCE)*, 2018, pp. 1796–1799.

[7] V. K. Awaar, P. Jugge, and S. T. Kalyani, "Field test of cost effective voltage source inverter for driving an induction motor," in *12th IEEE International Conference on Electronics, Energy, Environment, Communication, Computer and Control (INDICON 2015)*, 2016, pp. 1–6.

[8] S. K. Dalai and D. K. Dash, "Analysis of single phase matrix converter with regenerative capabilities of single phase induction motor," in *2017 International Conference on Smart Technologies for Smart Nation (SmartTechCon 2017)*, 2018, pp. 465–469.

[9] F. Ronilaya, S. Ilmawati, M. Huda, W. Anistia, I. N. Syamsiana, and M. N. Hidayat, "A development of an arduino pure sine wave inverter for a small scale off-grid solar pv system," *IOP Conference Series: Materials Science and Engineering*, vol. 1073, no. 1, p. 012043, 2021.

[10] E. L. Blanco, "Design and implementation of an inverter with spwm modulation," in *2017 IEEE Central America and Panama Student Conference (CONESCAPAN 2017)*, 2017, pp. 1–6.