

Generation of Space Vector PWM by Using Arduino Uno

Nur Ashida Salim^{1*}, Muhammad Azizi Kaprowi² and Ahmad Asri Abd Samat³

¹*Faculty of Electrical Engineering, Universiti Teknologi Mara, 40450 Shah Alam, Selangor, Malaysia*

²*Universiti Teknologi Mara Pulau Pinang (UiTM), 13500 Pulau Pinang, Malaysia*

ABSTRACT

Space Vector Pulse Width Modulation (SVPWM) method is widely used as a modulation technique to drive a three-phase inverter. It is an advanced computational intensive method used in pulse width modulation (PWM) algorithm for the three-phase voltage source inverter. Compared with the other PWM techniques, SVPWM is easier to implement, thus, it is the most preferred technique among others. Mathematical model for SVPWM was developed using MATLAB/ Simulink software. In this paper, the interface between MATLAB Simulink with the three-phase inverter by using Arduino Uno microcontroller is proposed. Arduino Uno generates the SVPWM signals for Permanent Magnet Synchronous Motor (PMSM) and is described in this paper. This work consists of software and hardware implementations. Simulation was done via Matlab/Simulink software to verify the effectiveness of the system and to measure the percentage of Total Harmonic Distortion (THD). The results show that SVPWM technique is able to drive the three-phase inverter with the Arduino UNO.

Keywords: SVPWM, Arduino UNO, PMSM, Matlab/Simulink

INTRODUCTION

Space Vector Pulse Width Modulation (SVPWM) technique is an advanced computational intensive PWM algorithm for

voltage source converter. This paper describes the digital implementation of SVPWM method using Arduino Uno microcontroller. The main focus of the study was to design and develop the mathematical model of SVPWM to drive the three-phase inverter. The parameters that are discussed in this paper are SVPWM switching signal, dead band and Total Harmonic Distortion (THD).

In the last couple of decades, Pulse Width Modulation (PWM) technique has been used to achieve variable voltage and frequency in power converters. Originally, Space Vector Modulation (SVM) was developed as a vector

ARTICLE INFO

Article history:

Received: 24 August 2016

Accepted: 03 Jun 2017

E-mail addresses:

nurashida606@salam.uitm.edu.my (Nur Ashida Salim),

muhdazizi19@yahoo.com (Muhammad Azizi Kaprowi)

ahmadasri759@ppinang.uitm.edu.my (Ahmad Asri Abd Samat)

*Corresponding Author

approach to PWM for three-phase inverters. This technique is more convenient to obtain higher voltage to the motor with lower THD for generating sine wave. The SVPWM technique can be implemented to generate switching signal into the three-phase inverter to drive permanent magnet synchronous motor (PMSM). The benefits of using PMSM in the system are: high efficiency, high power density, higher power factor, free maintenance operation and it can be used in various types of applications (Harahap et al., 2014). Vipin and George (2014) stated that a high response system is needed by a high-performance motor control system to act immediately when a motor experiences any disturbance.

DSP, PIC and Arduino are types of processors that can be used to interface between Simulink and hardware. However, these processors have their limitations and problems such as complex implementations, moderate processing, and higher switching losses (Naik et al., 2014). In this project, Arduino UNO microcontroller is used to interface the hardware with MATLAB Simulink to overcome these limitations. The advantages of using Arduino Uno are: the board is inexpensive, simple and has a clear programming environment (Zulkifli et al., 2015). Designing a signal generator of SVPWM and interfacing it with a microcontroller have an advantage, namely it is easy to be programmed (Slamet. 2013). This paper looks at both software and hardware implementations

The SVPWM is a digital modulating technique that is designed to generate PWM load line voltage that are in average equals to a given reference load line voltage. Compared with the SPWM technique, SVPWM is easier to be implemented because it has higher DC voltage utilisation ratio (Kumar et al., 2010). In SVPWM technique, the voltage reference is presented by using a revolving reference vector. The SVPWM uses only one reference space vector to generate three-phase sine wave (Badran et al., 2013).

In addition, microcontroller technologies and the power electronics device have been studied and developed to efficiently support power drive systems in designing the inverter (Quach et al., 2012). The SVPWM algorithm could improve the quality of AC motor by reducing the harmonic and adjusting the amplitude and frequency of output voltage.

The SVPWM refers to a switching scheme of the six power switches of a three-phase inverter. Six of the voltage vectors ($V_1 - V_6$) are working states that form stationary vectors in the $\alpha\beta$ frame and divide the plane into six sectors with each having an angle of 60 degree as shown in Figure 1 (Iqbal et al. 2006). The SVPWM generate a voltage vector which is close

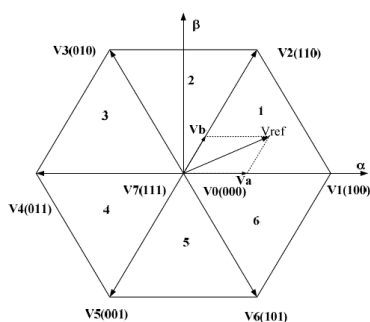


Figure 1. Voltage vectors and sectors in $\alpha\beta$ frame

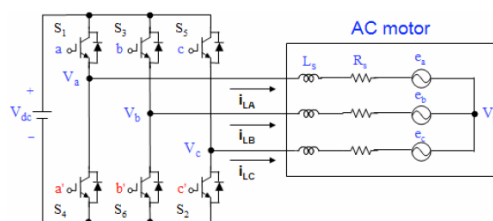


Figure 2. Circuit model for three phase-inverter

to the reference circle through the various switching modes of inverter (Wang et al., 2008). The switches of the inverter are controlled according to the voltage vector at the given time with respect to the switching period (Nazlee et al., 2010).

METHODOLOGY

The switching technique in this project is used to drive the three-phase inverter by using SVPWM technique. The switching technique is designed and simulated by using the MATLAB/Simulink software. The Arduino Uno microcontroller is used to interface the switching signal designed in Matlab/Simulink to three-phase inverter. The switching signal that has been generated from Simulink software is uploaded into the Arduino Uno microcontroller. Matlab/Simulink software was used to design the mathematical model of SVPWM and the results were measured through the scope. The Arduino UNO was used to interface between the Simulink model and the three-phase inverter. The output waveforms were measured by using oscilloscope. This project was able to generate SVPWM signal by using simple interfacing platform, which is, Arduino UNO. The interfacing becomes simple without using any code of programming to generate SVPWM into three-phase inverter.

Principle of SVPWM

The SVPWM works based on the principle that when the upper leg of inverter is switched ON, the equivalent of the lower leg is switched OFF. Figure 2 shows the circuit model of the three-phase inverter which consists of six power switches that form the output. The switching signal are controlled by the switching variable of a , a' , b , b' , c and c' . From the circuit model, when a , b or c is 1, the equivalent a' , b' or c' is 0. Thus, the ON and OFF states of the upper leg of an inverter S_1 , S_3 and S_5 can be used to determine the output voltages. The generation of SVPWM signal consists of the sector calculation model, XYZ calculation model, T_1T_2 calculation model, T_{abcon} calculation model and V_{abc} calculation model.

Sector calculation model is used to determine which sector of the voltage vector is within its limit. The XYZ calculation model and T_1T_2 calculation model are sectors that determine the operation time of the fundamental vectors. T_1 is the main vector operating time in the current work sector and T_2 is the vice-vector of operating time. Another sector is T_{abcon} calculation model which is used to generate SVPWM waveform by using value T , T_1 , and T_2 . The last part for SVPWM model is V_{abc} calculation model which is the switch operation time. The generation of symmetrical SVPWM can be performed by comparing the calculated value of T_{cm1} , T_{cm2} , and T_{cm3} with the equilateral triangle diagram.

Switching time duration of sector 1 can be calculated as follows:

$$\int_0^T V_{REF} = \int_0^{T_1} V_1 + \int_{T_1}^{T_1+T_2} V_2 + \int_{T_1+T_2}^T V_o \quad (1)$$

$$TV_{REF} = (T_1V_1) + (T_2V_2) \quad (2)$$

$$TV_{REF} \begin{bmatrix} \cos \alpha \\ \sin \alpha \end{bmatrix} = T_1 \frac{2}{3} V_{DC} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + T_2 \frac{2}{3} V_{DC} \begin{bmatrix} \cos \frac{\pi}{3} \\ \sin \frac{\pi}{3} \end{bmatrix} \quad (3)$$

where T_1 , T_2 , and T are times during at which V_1 , V_2 , and zero-vector are applied, V_{REF} is reference voltage of voltage vector and α is phase angle of output vector, $0 \leq \alpha \leq 60$.

From equation (3)

$$TV_{REF} [\cos \alpha] = T_1 \frac{2}{3} V_{DC} [1] + T_2 \frac{2}{3} V_{DC} \left[\cos \frac{\pi}{3} \right] \quad (4)$$

$$TV_{REF} [\sin \alpha] = T_2 \frac{2}{3} V_{DC} \left[\sin \frac{\pi}{3} \right] \quad (5)$$

From equation (4) and (5), switching time duration for sector 1 is:

$$T_1 = T \frac{|V_{REF}|}{\frac{2}{3} V_{DC}} \sin \left[\left(\frac{\pi}{2} - \alpha \right) \right] \quad (6)$$

$$T_2 = T \frac{|V_{REF}|}{\frac{2}{3} V_{DC}} \sin \left[\frac{\alpha}{\frac{\pi}{3}} \right] \quad (7)$$

$$T_0 = T + T_1 + T_2 \quad (8)$$

where T_0 is time of the zero vector is applied. Thus, switching time duration for other sector can be gained from equation (6), (7) and (8).

Equation (9) and (10) represent the T_1 and T_2 respectively.

$$T_1 = \frac{\sqrt{3}T|V_{REF}|}{V_{DC}} \left(\sin \frac{n}{3} \pi \cos \alpha - \cos \frac{n}{3} \pi \sin \alpha \right) \quad (9)$$

$$T_2 = \frac{\sqrt{3}T|V_{REF}|}{V_{DC}} \left(-\cos \alpha \sin \frac{n-1}{3} \pi - \sin \alpha \cos \frac{n-1}{3} \pi \right) \quad (10)$$

where n is number of sector (1 to 6).

After all the mathematical equations have been derived, then all the equations are modelled in Matlab/Simulink to generate the switching signal. Figure 3 shows the block diagram that contains all the mathematical equation developed in Matlab/Simulink.

Hardware Implementation

The Arduino UNO microcontroller functions as interfacing component between the software and hardware parts. Figure 4 shows the simulation setup for interfacing Simulink with Arduino

UNO. The selected pins from Arduino UNO to generate the output are pin 3, 5, 6, 9, 10 and 11. These pins are selected because it can produce the PWM output signal.

The Arduino UNO needs a supply of 5V to turn ON. The signal from Arduino UNO is an output of SVPWM which is an ON or OFF signal that functioned as a switch for the inverter. The unit delay block is used to give the delay time between ON time and OFF time for each signal. Each of the selected pin of Arduino UNO is connected to the three-phase inverter with the gate driver. Then, the output of three-phase inverter is connected to the PMSM.

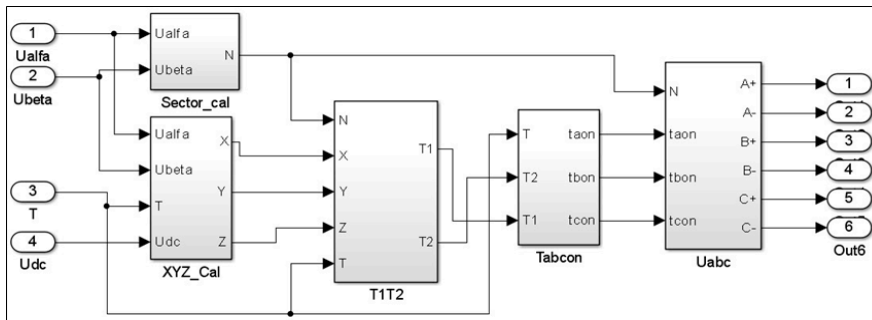


Figure 3. Simulation model of SVPWM

RESULT AND DISCUSSION

This project was simulated in Matlab/ Simulink software to test the effectiveness of the system. The mathematical model for SVPWM was developed first and then the Arduino software was designed to interface between the Matlab/Simulink and the three-phase inverter. The responses of the speed, voltage and current were observed by changing the value of the modulation index.

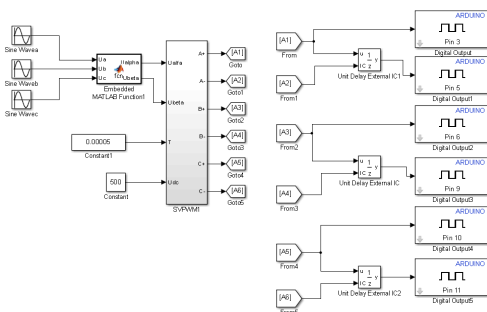


Figure 4. Interfacing model between Simulink and Arduino UNO



Figure 5. Hardware setup for Arduino UNO

Total Harmonic Distortion (THD)

Total Harmonic Distortion (THD) is the existing amount of harmonic distortion in the waveform. In this analysis, the following parameters were selected to observe the percentage of THD in the system:

$$V_{dc} = 250V$$

Fundamental frequency = 50 Hz

Modulation index = 0.5, 0.7 and 0.9

Table 1 shows the reading of THD based on the value of modulation index. Equation (11) is used to calculate the U_{dc} . Table 1 shows that by increasing the value of modulation index, it will give the less value of THD.

$$U_{dc} = m_i \cdot \sqrt{3} \cdot \frac{V_{dc}}{2} \tag{11}$$

Table 1
Modulation index and corresponding total harmonic distortion

| Modulation Index | Udc (V) | THD for current (%) |
|------------------|---------|---------------------|
| 0.5 | 108 | 20.74 |
| 0.7 | 152 | 20.71 |
| 0.9 | 195 | 18.72 |

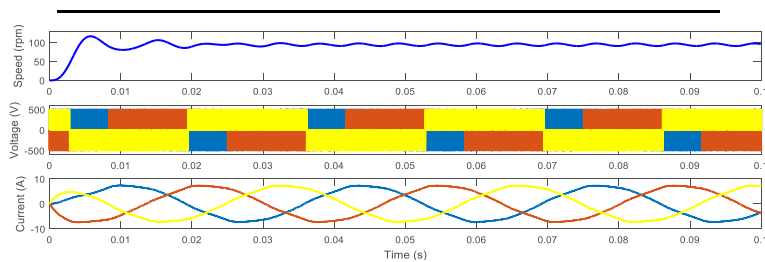


Figure 6. Simulation results with modulation index of 0.5

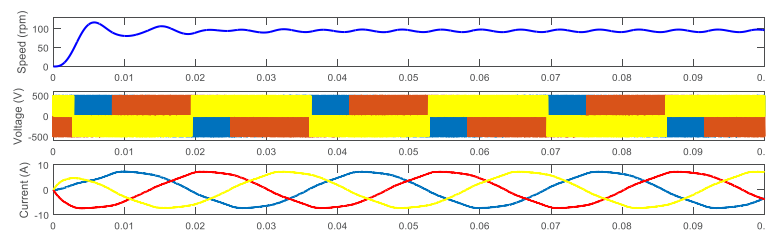


Figure 7. Simulation results with modulation index of 0.7

Generation of Space Vector PWM by Using Arduino Uno

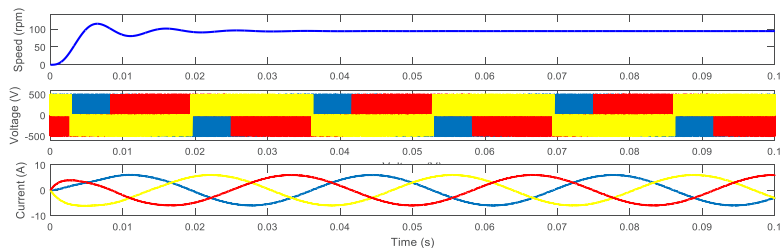


Figure 8. Simulation results with modulation index of 0.9

Figures 6,7 and 8 show the response of speed, voltage and current of the PMSM with modulation index of 0.5, 0.7 and 0.9 respectively. Results show that the higher number of THD will give more harmonics in current and cause more ripples in speed as depicted in Figure 6 and Figure 7.

In addition, as depicted in Figure 8, the speed ripple is near to zero because the harmonics contents in current are less. As tabulated in Table 1, the value of the THD in this case is 18.72%.

Switching signal

The switching signal from SVPWM that was designed in Matlab/Simulink was injected to the three-phase inverter via Arduino UNO. The interfacing between Matlab/Simulink and Arduino UNO is as shown in Figure 4 and Figure 5. Table 2 shows the pin connection at Arduino UNO board that represents the switching signal of the inverter.

Table 2
Switching signal and Arduino UNO pin setup

| Switching Signal | Arduino UNO Pin |
|------------------|-----------------|
| A+ | 3 |
| A- | 5 |
| B+ | 6 |
| B- | 9 |
| C+ | 10 |
| C- | 11 |

Table 3
Amplitude and period of switching signal before and after entering gate driver

| Switching Signal | Before Gate Driver | | After Gate Driver | |
|------------------|--------------------|-------------|-------------------|-------------|
| | Pk-Pk (V) | Period (ms) | Pk-Pk (V) | Period (ms) |
| A+ | 7.6 | 6.638 | 15.1 | 6.615 |
| A- | 6.4 | 6.613 | 14.5 | 6.357 |
| B+ | 7.6 | 6.493 | 14.7 | 6.424 |
| B- | 7.4 | 6.607 | 14.7 | 6.607 |
| C+ | 7.6 | 6.641 | 14.9 | 6.616 |
| C- | 6.4 | 6.677 | 14.6 | 6.617 |

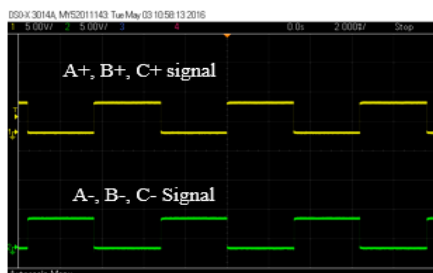


Figure 9. Switching signal from Arduino UNO pin

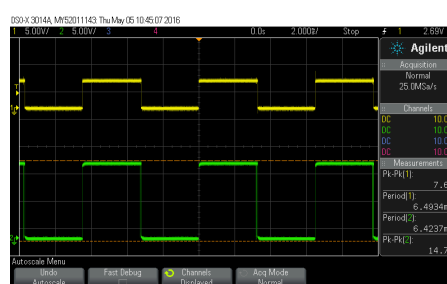


Figure 10. Output waveform after entering gate driver for signal B+

The SVPWM has 6 outputs (A+, A-, B+, B-, C+ and C-) which represent the six sector. The value of output is either 0 or 1. Figure 9 shows the switching signal for A+, A-, B+, B-, C+ and C- from the Arduino UNO. The switching signal will inject the gate driver before it triggers the switch of the three-phase inverter. The gate driver is a power amplifier that receives a low power input and produces a high voltage and current gate drive to switch ON and OFF the MOSFET. Table 3 shows the result of the amplitudes of switching signal from Arduino UNO before and after entering the gate driver and the period is for one cycle. The results show that the amplitude has been amplified by the gate driver. Figure 10 shows the output waveform after entering the gate driver for signal B+.

CONCLUSION

The basic principle and the mathematical model of SVPWM were developed using Matlab/Simulink. The Arduino UNO microcontroller was used to interface between the SVPWM modelled in Matlab/Simulink and the three-phase inverter. The results show that SVPWM switching signal is able to generate the three-phase inverter with the Arduino UNO microcontroller. The value of THD was also measured in simulation and the results shows that the modulation index of 0.9 gives the lowest percentage of THD. It validated that higher value of the modulation index will deliver lower harmonics contains in the system.

ACKNOWLEDGEMENT

The authors would like to thank the Research Management Institute (RMI), Universiti Teknologi Mara, Malaysia and the Ministry of Higher Education (MOHE), Malaysia for research grant 600-RMI/DANA 5/3/LESTARI (77/2015).

REFERENCES

- Asma, N. R., & Jangamshetti Suresh, H. (2014). Implementation of Space Vector Pulse Width Modulation using Arduino. *International Journal of Science and Research*, 3(7), 1706-1710.
- Badran, M. A., Tahir, A. M., & Faris, W. F. (2013). Digital Implementation of Space Vector Pulse Width Modulation Technique Using 8-bit Microcontroller. *World Applied Sciences Journal*, 21, 21-28.

- Harahap, C. R., Saito, R., Yamada, H., & Hanamoto, T. (2014). Speed control of permanent magnet synchronous motor using FPGA for high frequency SiC MOSFET Inverter. *Journal Engineering Science and Technology, October*(2014), 11-20.
- Iqbal, A., Lamine, A., & Ashraf, I. (2006, September). MATLAB/SIMULINK model of space vector pwm for three-phase voltage source inverter. In *Proceedings of the 41st International Universities Power Engineering Conference* (Vol. 3, pp. 1096-1100). IEEE.
- Kumar, K. V., Michael, P. A., John, J. P., & Kumar, D. S. S. (2010). Simulation and comparison of SPWM and SVPWM control for three phase inverter. *ARPJN Journal of Engineering and Applied Sciences*, 5(7), 61-74.
- Nazlee, A. M., Hamid, N. H., Hussin, F. A., & Ali, N. B. Z. (2010, December). Space Vector PWM for PMSM simulation using Matlab Simulink. In *Circuits and Systems (APCCAS), 2010 IEEE Asia Pacific Conference on* (pp. 1127-1130). IEEE.
- Quach, D. C., Yin, Q., Shi, Y. F., & Zhou, C. J. (2012, December). Design and Implementation of Three-phase SVPWM Inverter with 16-bit dsPIC. In *2012 12th International Conference on Control Automation Robotics and Vision (ICARCV)*.
- Slamet. (2013). Generation of Space Vector PWM Using Microcontroller Atmega 16. *International Journal of Scientific & Engineering Research*, 4(3), 1-5.
- Vipin, A. M., & George, S. (2014, July). Hardware implementation of space vector PWM control of Permanent Magnet Synchronous Motor. In *Annual International Conference on Emerging Research Areas: Magnetism, Machines and Drives (AICERA/iCMMD)*, 2014 (pp. 1-5). IEEE.
- Wang, Z. G., Jin, J. X., Guo, Y. G., & Zhu, J. G. (2008). SVPWM techniques and applications in HTS PMSM machines control. *Journal of Electronic Science and Technology of China*, 6(2), 191-197.
- Zulkifli, S. A., Hussin, M. N., & Saad, A. S. (2014, December). MATLAB-Arduino as a low cost microcontroller for 3 phase inverter. In *IEEE Student Conference on Research and Development (SCOReD), 2014* (pp. 1-5). IEEE.

