Space Vector Modulated PWM Generation for Motor Control Systems

Zarzisur Rahman Rony*, Shuvangkar Chandra Das and Md. Ziaur Rahman Khan

Department of EEE, Bangladesh University of Engineering and Technology, Dhaka-1205, Bangladesh

*zarzis.rony@gmail.com

Abstract—**Microcontroller based sample Space Vector Modulation for two-level three-phase voltage source inverter is discussed here. The system is designed in ATmega64, which is an 8-bit low cost microcontroller. A fixed number of sample is used to produce a full wave. Here, Timer overflow interrupt is used to generate every new sample. All the calculation is done in interrupt subroutine. With simple modification, the system can be integrated to a variable frequency drive (VFD) of induction motor. Simulations are performed in Matlab/Simulink and Proteus Design Suite. Similar results are found in both simulation tools. The simulation results show that the proposed modified method results in reduction of calculation cycle numbers in interrupt routine.**

***Index Terms*—Space Vector Modulation, Scalar Control of Induction Motor, Three Phase Inverter, Arduino Uno.**

I. INTRODUCTION

Three-phase voltage source inverter has a wide application in power electronics such as industrial motor drive, electric vehicle, alternate energy interfacing to grid etc. It can produce variable voltage with variable frequency, which is widely used in scalar (V/f) control of induction motor. Many modulation techniques such as sine triangle PWM (SPWM), third harmonic injected PWM and space vector modulated PWM (SVM) are available. SVM shows superior performance over SPWM in terms of DC bus utilization. 90.7 percent of the fundamental at the square wave (Line to neutral) is available in the linear region, compared to 78.55 percent in the sinusoidal PWM [1]. In addition, SVM has lower switching loss and less harmonic distortion than SPWM [3]. Digital implementation of SVM is also easy relative to other strategies. A significant amount of research work has been done on SVM modulation techniques and it`s applications [5-6][10][12-14]. Most of the time SVM is implemented using DSP/DSC controller. Some works are also done on Arduino based platform such as Arduino Mega, Arduino DUE board [7-8]. In this work, SVM is implemented using a low cost ATmega64 microcontroller. The program algorithm is such that it can be implemented also using ATmega328p which is used in Arduino Uno board. A modified method is proposed for systems where frequency doesn`t change frequently. A Matlab/Simulink simulation is done and hardware system implementation is checked by Proteus Design Suite software.

II. PRINCIPLE OF OPERATION OF SPACE VECTOR MODULATED INVERTER

Three Phase voltage V_{an}, V_{bn}, V_{cn} can be visualized as a reference vector, V_{ref} rotating in a voltage space.Using Clarke`s Transformation (α, β) three phase to two phase model achieved with equations:

$$V_\alpha = \frac{2}{3}(V_{an} - 0.5V_{bn} - 0.5V_{cn}) \quad (1)$$

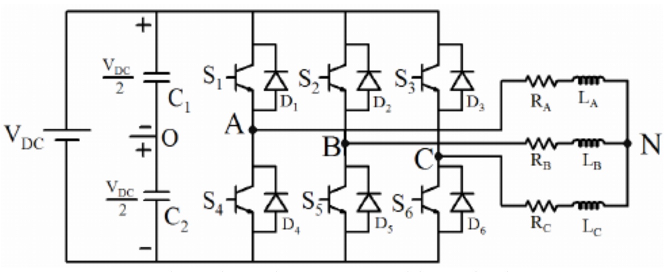

Fig.1 Three Phase Inverter with R-L load

$$V_\beta = \frac{2}{3}\left(\frac{\sqrt{3}}{2}V_{bn} - \frac{\sqrt{3}}{2}V_{cn}\right) \quad (2)$$

$$V_{ref} = \sqrt{(V_\alpha^2 + V_\beta^2)} \text{ and } \theta = \tan^{-1}\frac{V_\beta}{V_\alpha} \quad (3)$$

Three-leg two-level inverter in Fig.1 has $2^3 = 8$ switching states, which act as active voltage vectors, and among them, 2 are zero voltage vector. Table 1 shows line to neutral voltages for different vectors. The voltage vectors divide the voltage space into 6 sectors as shown in Fig.2. In each sector reference voltage vector is produced with help of two adjacent active vector`s and one zero vector`s dwell time (T_l, T_r, T_0). T_l, T_r can be imagined as time associated to left sided and right sided active vector in a sector. Using volt sec balance in any sector,

$$V_{ref} * T_s = V_r * T_r + V_l * T_l + (V_7 or V_0) * T_0$$

$$T_r = m * T_s * \sin(60 - \theta) \quad (4)$$

$$T_l = m * T_s * \sin\theta \quad (5)$$

$$\text{And } \ T_0 = T_s - T_r - T_l \quad (6)$$

Where T_s =Sample Time, m=modulation index=$\frac{\sqrt{3}*V_{ref}}{V_{DC}}$, in linear region m can vary from 0 to 1 [2].

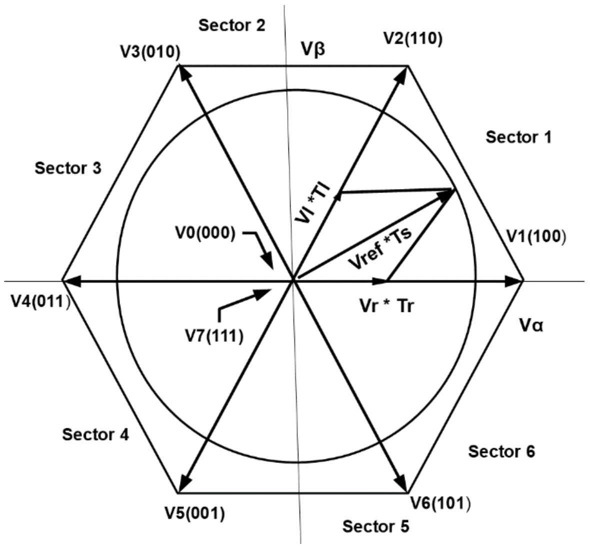

Fig.2 Voltage Space Vectors

TABLE 1

LINE TO NEUTRAL VOLTAGE FOR DIFFERENT VECTORS

Vectors	[S1,S2,S3]	V_{an}	V_{bn}	V_{cn}	θ
V1	[1,0,0]	$\frac{2V_{DC}}{3}$	$\frac{-V_{DC}}{3}$	$\frac{-V_{DC}}{3}$	0
V2	[1,1,0]	$\frac{V_{DC}}{3}$	$\frac{V_{DC}}{3}$	$\frac{-2V_{DC}}{3}$	$\frac{\pi}{3}$
V3	[0,1,0]	$\frac{-V_{DC}}{3}$	$\frac{2V_{DC}}{3}$	$\frac{-V_{DC}}{3}$	$\frac{2\pi}{3}$
V4	[0,1,1]	$\frac{-2V_{DC}}{3}$	$\frac{V_{DC}}{3}$	$\frac{V_{DC}}{3}$	π
V5	[0,0,1]	$\frac{-V_{DC}}{3}$	$\frac{-V_{DC}}{3}$	$\frac{2V_{DC}}{3}$	$\frac{4\pi}{3}$
V6	[1,0,1]	$\frac{V_{DC}}{3}$	$\frac{-2V_{DC}}{3}$	$\frac{V_{DC}}{3}$	$\frac{5\pi}{3}$
V7	[1,1,1]	0	0	0	-
V0	[0,0,0]	0	0	0	-

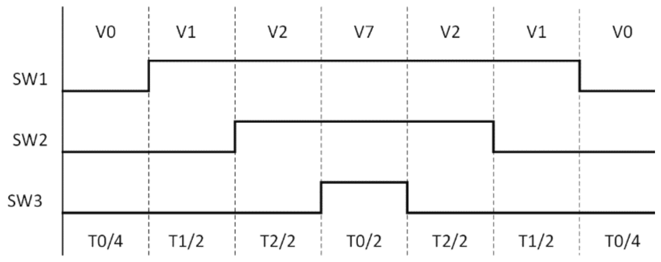

Fig.3 Symmetrical Switching sequence in sector 1

TABLE 2

DUTY CYCLES IN SECTORS

	On Times Of Inverter Upper Switch		
Sector	Switch 1	Switch 2	Switch 3
1	$T_r+T_l+0.5T_0$	$T_l+0.5T_0$	$0.5T_0$
2	$T_r+0.5T_0$	$T_r+T_l+0.5T_0$	$0.5T_0$
3	$0.5T_0$	$T_r+T_l+0.5T_0$	$T_l+0.5T_0$
4	$0.5T_0$	$T_r+0.5T_0$	$T_r+T_l+0.5T_0$
5	$T_l+0.5T_0$	$0.5T_0$	$T_r+T_l+0.5T_0$
6	$T_r+T_l+0.5T_0$	$0.5T_0$	$T_r+0.5T_0$

Symmetric strategy gives better performance than right aligned, left aligned, and alternate switching strategy in terms of THD parameter [4]. In symmetrical method, switching frequency is twice of sample frequency. Here active vectors are arranged in such a way that minimizes switching loss. Fig.3 shows switching sequence in sector 1. Table 2 shows duty cycles of upper three legs in different sectors.

III. SYSTEM DESIGN

In this work, simulation is performed for Load Resistance=10Ω, Load Inductance=0.1H, V_{DC}=100Volt

A. Matlab Simulation and results

The block diagram of the system in MATLAB Simulink environment is shown in Fig 4. Reference time signals (Fig.5) are generated through Clarke transformation-sector identification-adjacent vector time calculation and vector selection. The reference signals are then compared with triangular wave to generate PWM. Fig.6 to Fig.8 show the PWM signals, line to neutral voltage of phase A and line current waveforms, respectively. All these waveforms are of command frequency of 50 Hz and 0.9 modulation index.

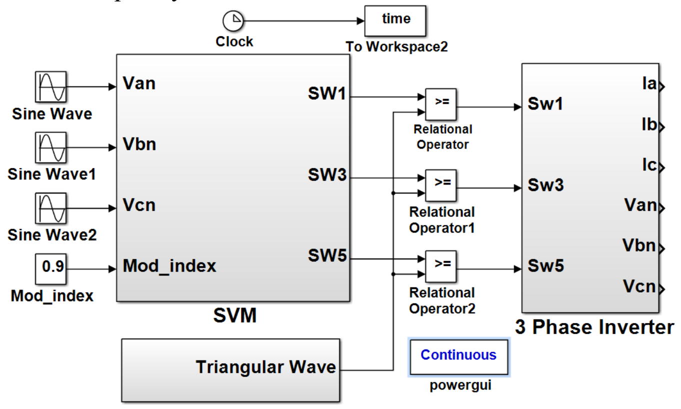

Fig.4 System design in MATLAB Simulink.

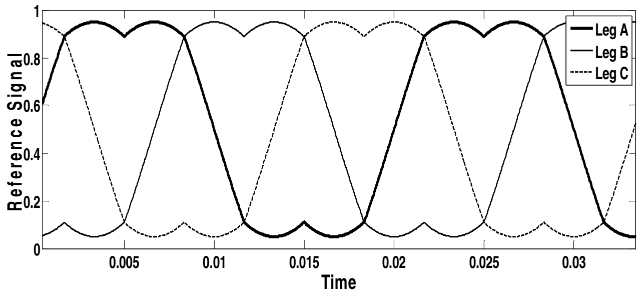

Fig.5 Reference Time Value for Upper Legs

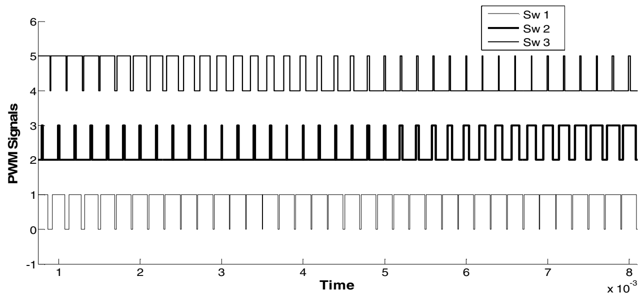

Fig.6 PWM signals for upper three switches.

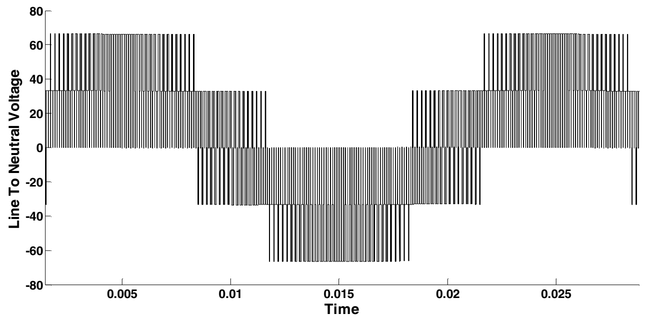

Fig.7 Line to neutral voltage of Phase A.

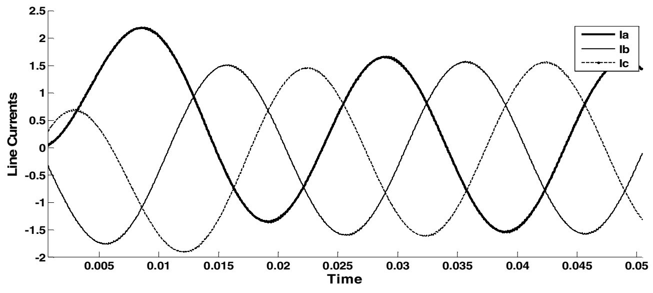

Fig.8 Three phase line current (for f=50 Hz and m=0.9)

B. Algorithm

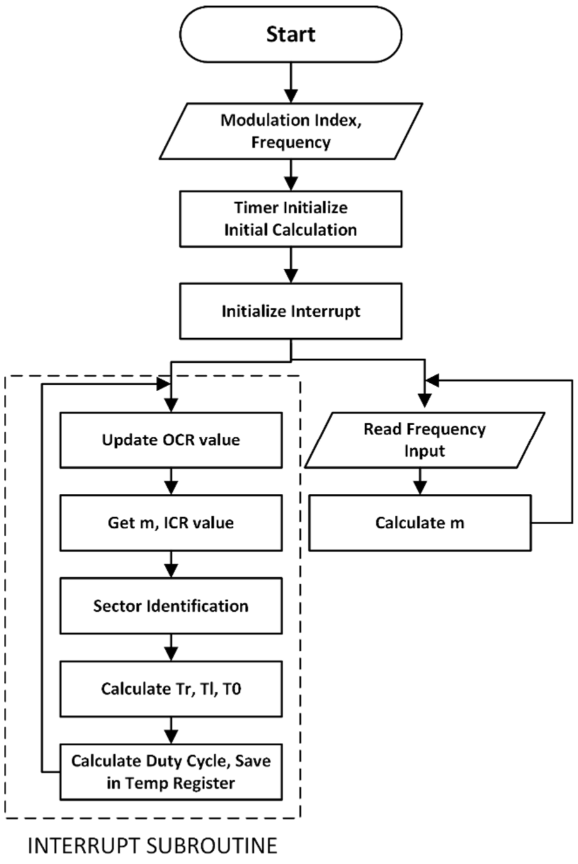

Fig.9 Algorithm of space vector modulation.

The algorithm is designed considering the architecture of ATmega64 microcontroller. ATmega64 offers phase corrected PWM through which symmetric PWM is generated. When timer counter register`s (TCNT) value matches input capture register`s (ICR) value it generates a timer overflow interrupt. In interrupts service routine every calculation needed to generate duty cycle for each leg is done. Fig.9 shows algorithm for calculation of duty cycle.

Reference vector`s angle calculation is a vital part of space vector modulation. However, angle calculation involves at least two multiplication so to optimize, a fixed number of samples are generated. For example, if 48 samples are used to create a full cycle then every sector contains $\frac{48}{6}$ =8 samples and every new sample will be shifted from the previous sample by $\frac{60}{8}$ =7.5 degree. For each sample, there is 2 adjutant vector time calculation $(Tr,\ Tl)$ necessity which involves trigonometric value (4) (5). In the beginning, the system is initialized. Total sample number, starting frequency and modulation index (m) are specified. To get trigonometric value, two sine look-up table is formed, one for angle θ and other for $(60^o - \theta)$. Next, the Timer is initialized, during when 1st sample`s duty cycle value is calculated and timer overflow interrupt is set. In the very beginning of interrupt routine duty cycle value is fed to output compare register (OCR) and then next sample`s value is calculated. To implement VFD, only ICR value and modulation index (m) need to be changed. To reduce very high starting current of induction motor, frequency is gradually increased. To do this, in interrupt service routine a starting algorithm is implemented. Fig.10 shows the result of changing frequency gradually. After getting ICR and m values, $Tr, Tl\ and\ T0$ are calculated. Then from table 2 duty cycle values for the upper three legs are estimated in different sectors. These duty cycle values are fed to OCR registers at the beginning of next interrupt subroutine with the help of temporary registers.

As there is only two variable (ICR, m) associated with frequency and voltage, with the help of ADC user`s choice of frequency can be synthesized. When a user presses a button or rotates potentiometer, ADC will sense and calculate ICR and m to implement variable frequency drive (VFD). Therefore, this algorithm is more generalized in nature.

Calculation time in interrupt routine can be further reduced. Solving $T_l = m * T_s * \sin\theta$ involves multiplying two floating-point number and one integer, which takes many cycles. To optimize, sine table value can be modified according to $sin\theta = m * T_s * sin\theta = K * sin\theta$, K`s value is decided according to frequency outside of interrupt routine. This modification decision is taken assuming that when the system goes steady state, frequency doesn`t change for a significant amount of time. Then each sine look-up table value can be modified once and be used instead of repeated multiplication. After modification, this equation, $T_l = \sin\theta$, can be used which is very fast. Fig.11 is generated to compare between generalized and modified method. This figure shows the performance of the system when frequency and modulation index is selected arbitrarily (without using V/f). Table 3 shows the total calculations of principle calculation block between generalized and modified method.

TABLE 3
CALCULATION NUMBERS AT STEADY STATE IN INTERRUPT ROUTINE

	Generalized	Modified
Multiplication	4	0
Addition	8	8
Switch Case	1	1
Bitwise Shifting	2	2
Total Cycle Taken	870	190

The three PWM signals are generated in Atmega 64 using one 16-bit timer. This is also possible in Arduino Uno (ATmega 328p) which has one 16-bit timer and two 8-bit timer. However, in ATmega328p only two PWM generation is possible with every timer. So two of the three Timers should be synchronized using GTCCR register to get three PWM.

C. Results from Proteus Simulation

The system with the microcontroller is simulated in Proteus. Fig.12 to Fig.14 shows PWM signals, V_{AB}, V_{AN} and current waveforms of 50 Hz and 0.9 modulation index. Fig.10 shows line currents for changing frequency gradually from 5 to 50 Hz with modulation index 0.1 to 0.9 respectively. Fig.15 shows V_{AB}, V_{AN} waveform of this process. Fig.11 shows gradually generating 5, 10,20,30,40 Hz line currents with 0.1, 0.5, 0.4, 0.2, 0.7 modulation index respectively.

A comparison of total calculation time between generalized and modified method is observed during generation of Fig.11. 8 MHz internal RC oscillator is used and the Timer`s frequency is made to 1Mhz. It has been noted that the average time needed by the modified method is around 60 microseconds while the generalized method needs around 120 microseconds. Also in experiments 48 sample is used. Fig.3 shows that in every sample generation there are two switchings. Therefore, to generate 50 Hz sine wave, switching frequency is $50 * 2 * 48 = 4.8\ Khz.$

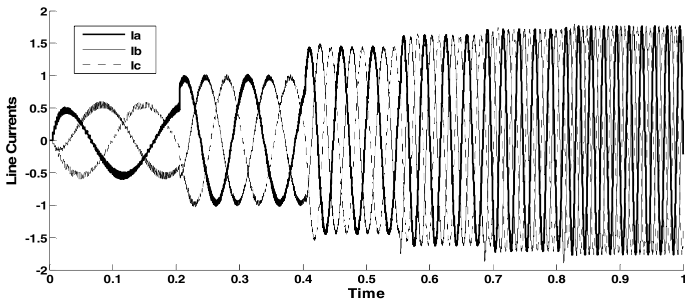

Fig.10 Generation of 5, 10, 20,30,40,50 Hz frequency of 0.1, 0.2, 0.4, 0.6, 0.8 and 0.9 modulation index respectively.

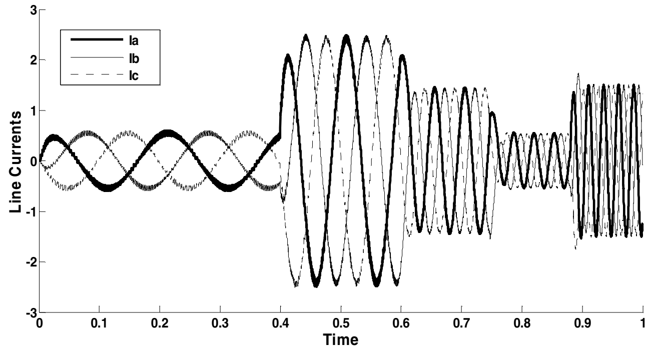

Fig.11 Generation of 5, 10,20,30,40 Hz frequency of 0.1, 0.5, 0.4, 0.2 and 0.7 modulation index respectively.

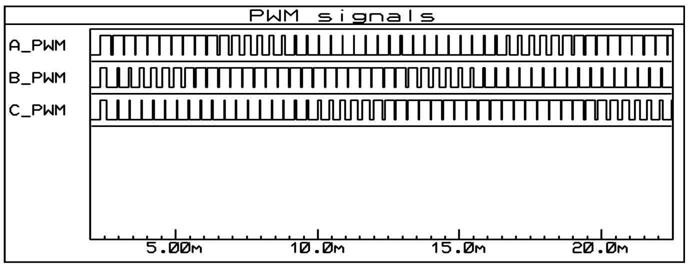

Fig.12 PWM signals.

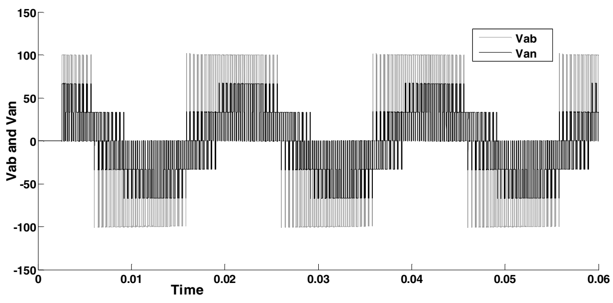

Fig.13 Line to Line and Line to neutral Voltage.

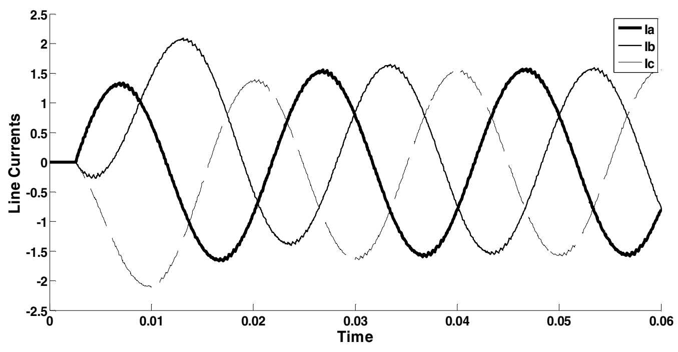

Fig.14 Line currents for f=50, m=0.9.

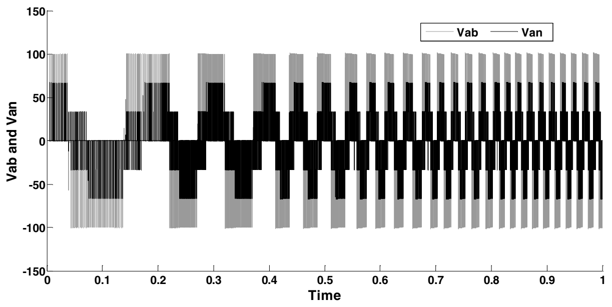

Fig.15 Line to Line and Line to neutral voltage during gradual increase of modulation index.

Each sample period is $\frac{1}{48*50} = 416$ microseconds. So when implementing generalized algorithm $(416 - 120)$ =296 microseconds is available to implement other processes and interfacing. If 16 MHz crystal is used then calculation time should reduce to half. More sample can be generated in a cycle and switching frequency will increase.

IV. CONCLUSIONS

Space vector modulation for three-phase two-level inverter is simulated in Matlab/Simulink environment and implementation with ATmega 64 microcontroller is observed in Proteus. Program design is done in two ways: generalized and modified method. It has been shown modified method needs less computation time then generalized method. The same algorithm can be implemented on Arduino Uno. The generated PWM signal can be made to use for variable frequency drive.

REFERENCES

[1] "Modern Power Electronics and AC drive",Bimel.K.Bose

[2] "High Power Converters and AC drives", Bin Wu

[3] "Inverter control: Comparative study between SVM and PWM"Omessaad ELBEJI, Zina BOUSSADA, Mouna BEN HAMED(2017)

[4] "Comparison of State Space Vector PWM Schemes For a Voltage Source Inverter" N. H. Saad, Abd El meged and A. M. A. Mahmoud (2017)

[5] "A Survey on Space Vector Pulse Width Modulation Technique for a Two-Level Inverter "Anup Kumar, Debashis Chatterjee (2017).

[6] "A Random Forest Regression Based SpaceVectorPWM Inverter Controller for the Induction Motor Drive"M A Hannan,M N Uddin (2016)

[7] Ansal, V., V. K. Remya, and Subhin Antony. "Realization of an arduino-mega based space vector modulation controller." Power Electronics (IICPE), 2016 7th India International Conference on. IEEE, 2016.

[8] Kherroubi, Zine El Abidine, et al. "Real time implementation of space vector pulse width modulation using arduino due board." Industrial Electronics Society, IECON 2016-422

[9] Maha, Zoghlami, and Bacha Faouzi. "Implementation of Space Vector modulation using DSP." Electrical Engineering and Software Applications (ICEESA), 2013 International Conference on. IEEE, 2013

[10] Akin, Bilal, and Nishant Garg. "Scalar (V/f) control of 3-phase induction motors." Texas Instruments Incorporated, Texas (2013).

[11] Rathnakumar, D., J. LakshmanaPerumal, and T. Srinivasan. "A new software implementation of space vector PWM." SoutheastCon, 2005. Proceedings. IEEE. IEEE, 2005.

[12] Jabbar, M. A., Ashwin M. Khambadkone, and Zhang Yanfeng. "Space-vector modulation in a two-phase induction motor drive for constant-power operation." IEEE Transactions on Industrial Electronics 51.5 (2004): 1081-1088.

[13] Shireen, Wajiha, Mohammed S. Arefeen, and David Figoli. "Controlling multiple motors utilizing a single DSP controller." IEEE Transactions on power electronics 18.1 (2003): 124-130.

[14] Van Der Broeck, Heinz Willi, H-C. Skudelny, and Georg Viktor Stanke. "Analysis and realization of a pulsewidth modulator based on voltage space vectors." IEEE transactions on industry applications 24.1 (1988): 142-150.