AN AT89C52 MICROCONTROLLER BASED HIGH RESOLUTION PWM CONTROLLER FOR 3-PHASE VOLTAGE SOURCE INVERTERS

K. M. RAHMAN AND S. J. M. IDRUS

Department of Mechatronics Engineering

Faculty of Engineering, International Islamic University Malaysia, P.O. Box 10, 50728, Malaysia

kazi@iiu.edu.my, junita@iiu.edu.my

Abstract: A novel technique for generating real time high resolution pulse width modulated patterns for a three phase voltage source inverter is presented in this paper. An AT89C52 microcontroller computes the PWM pulse widths on carrier cycle basis based on the input frequency. The microcontroller send the pulse width information for the three phases to a memory minimized ROM lookup table, which is scanned by a binary counter to generate the real time PWM patterns. The on board timer of AT89C52 is utilized to generate programmed clock for the scanning binary counter. The overall design is hardware minimized and compact that makes it feasible for low cost high performance ac drive applications. The proposed PWM controller supports variable frequency variable voltage operation for wide range and hence is ideally suited for industrial drives requiring wide speed variations.

Keywords: *Pulse width modulation, Microcontroller application, Voltage source inverter, variable speed ac drives.*

1. INTRODUCTION

Three phase voltage source inverters are controlled by pulse width modulation (PWM) techniques. The prime goal of a PWM technique is to minimize the harmonics in the phase voltage and phase current spectrum. Different PWM strategies are available in the literature for controlling three phase inverters using per phase modulation such as regular sampled PWM (RSPWM) or voltage space vector (SVPWM) technique. Both RSPWM [1-4] and SVPWM [5-6] have well defined switching equations for computing PWM pulse widths and are usually implemented using timers and a microprocessor. The SVPWM needs more computations than the RSPWM, however, SVPWM have linear over modulation capability. Harmonic elimination PWM (HEPWM) is another technique that computes switching points strategically eliminating some selective harmonics and is the most computation intensive PWM technique that cannot be implemented directly in real time. Switching patterns close to HEPWM are obtainable using regular sampled harmonic elimination PWM (RSHEPWM), where, the leading edge of a PWM switching pulse for any phase is calculated from the sine modulating reference and the trailing edge is computed from a shifted sine reference [7-8]. Considering harmonic behaviour, the RSPWM is the most efficient PWM technique for feed forward inverter control applications. The RSPWM is one of the mostly used PWM techniques and have a number of derivatives. The symmetric time averaged RSPWM [9] is very efficient in respect of both performance and speed. The performance of a PWM scheme depends on how the PWM control patterns are calculated and also on the implementation strategy. Even the most efficient PWM technique may offer worst performance if the resolution of the PWM pattern is not maintained properly. This happens because of the limitations of the digital hardware through which the PWM strategy is implemented in real time.

2. PULSE WIDTH CALCULATIONS

The PWM pulse widths are to be calculated such that the inverter output has the lowest order harmonic near the carrier switching frequency. Amongst the various PWM strategies, the regular sampled PWM is found to give better output at lesser computation complexity.

2.1 Regular Sampled PWM

The modulating signal is a sinusoid at the inverter fundamental frequency (f). The modulating signal is sampled at regular intervals at the carrier frequency (f_s). The number of samples in a fundamental period is N_{max} given by,

$$N_{\max} = {}^{f_s}\!/\!_{f} \tag{1}$$

The three phase modulating signals are,

$$\begin{aligned} v_a &= V_m \sin(\omega t) \\ v_b &= V_m \sin(\omega t - 2\pi/3) \\ v_c &= V_m \sin(\omega t - 4\pi/3) \end{aligned} \tag{2}$$

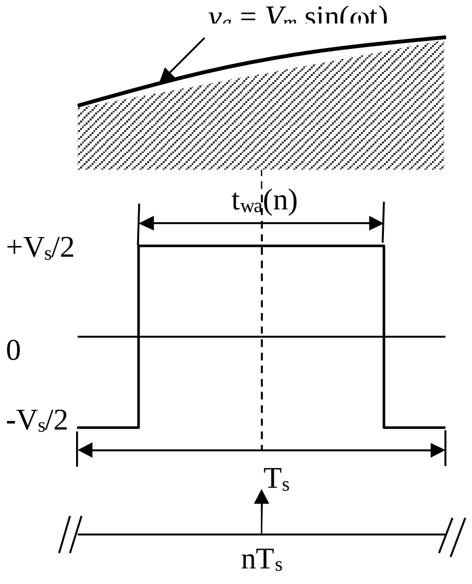

Fig. 1: Pulse width calculations for two-level PWM, the area under the modulating signal for nth carrier period is equated to be same as the area of the two-level PWM pulse.

For the nth PWM pulse, the area of the two-level PWM pulse should be equal to the shaded area under the modulating signal as illustrated in Fig. 1,

$$\frac{V_s}{2}\times t_{wa}(n)-\frac{V_s}{2}\times\{T_s-t_{wa}(n)\}=V_m\sin(n\omega T_s)\times T_s \tag{3}$$

Solving (3), the PWM pulse width is given by,

$$t_{wa}(n)=\frac{T_s}{2}\left[1+\frac{2V_m}{V_s}\sin(n\omega T_s)\right] \tag{4}$$

$$t_{wa}(n)=\frac{T_s}{2}\left[1+M\sin(n\omega T_s)\right] \tag{5}$$

where, M is the index of modulation, $M = 2V_m/V_s$. Equation (5) gives the PWM pulse width for phase A. For phases B and C, the PWM pulse widths are,

$$t_{wb}(n)=\frac{T_s}{2}\left[1+M\sin(n\omega T_s-2\pi/3)\right] \tag{6}$$

$$t_{wc}(n)=\frac{T_s}{2}\left[1+M\sin(n\omega T_s-4\pi/3)\right] \tag{7}$$

Adding (5), (6) and (7) results,

$$t_{wa}(n)+t_{wb}(n)+t_{wc}(n)=\frac{3T_s}{2} \tag{8}$$

From (8), the PWM pulse width for phase C can be obtained as,

$$t_{wc}(n)=\frac{3T_s}{2}-\{t_{wa}(n)+t_{wb}(n)\} \tag{9}$$

2.2 PWM Pulse Width in Sampled Domain

For generating real time PWM waveforms for the three phase inverter using a digital hardware, the PWM pulse widths are transformed into sampled domain. Considering K_{max} samples in a carrier period T_s, the sampling time ΔT_s is given by,

$$\Delta T_s = {}^{T_s}\!/\!{}_{K_{\max}} \tag{10}$$

If $K_a(n)$, $K_b(n)$, and $K_c(n)$ are the numbers of 'HIGH' samples for the phases A, B, and C respectively in sampled domain, their magnitudes are given by,

$$K_a(n)=\frac{t_{wa}(n)}{\Delta T_s}=\frac{K_{\max}}{2}+\frac{K_{\max}}{2}M\sin(n\omega T_s) \tag{11}$$

$$K_b(n)=\frac{t_{wb}(n)}{\Delta T_s}=\frac{K_{\max}}{2}+\frac{K_{\max}}{2}M\sin(n\omega T_s-2\pi/3) \tag{12}$$

$$K_c(n)=\frac{t_{wc}(n)}{\Delta T_s}=\frac{3K_{\max}}{2}-\{K_a(n)+K_b(n)\} \tag{13}$$

3. IMPLEMENTATION SCHEME

The PWM implementation scheme is shown in Fig. 2. The core processor is an AT89C52 microcontroller, where, the desired frequency/speed is set by an analogue potentiometer. An 8-bit ADC (ADC0804) converts the frequency/speed setting into digital binary code that is given as input to port 2 of AT89C52.

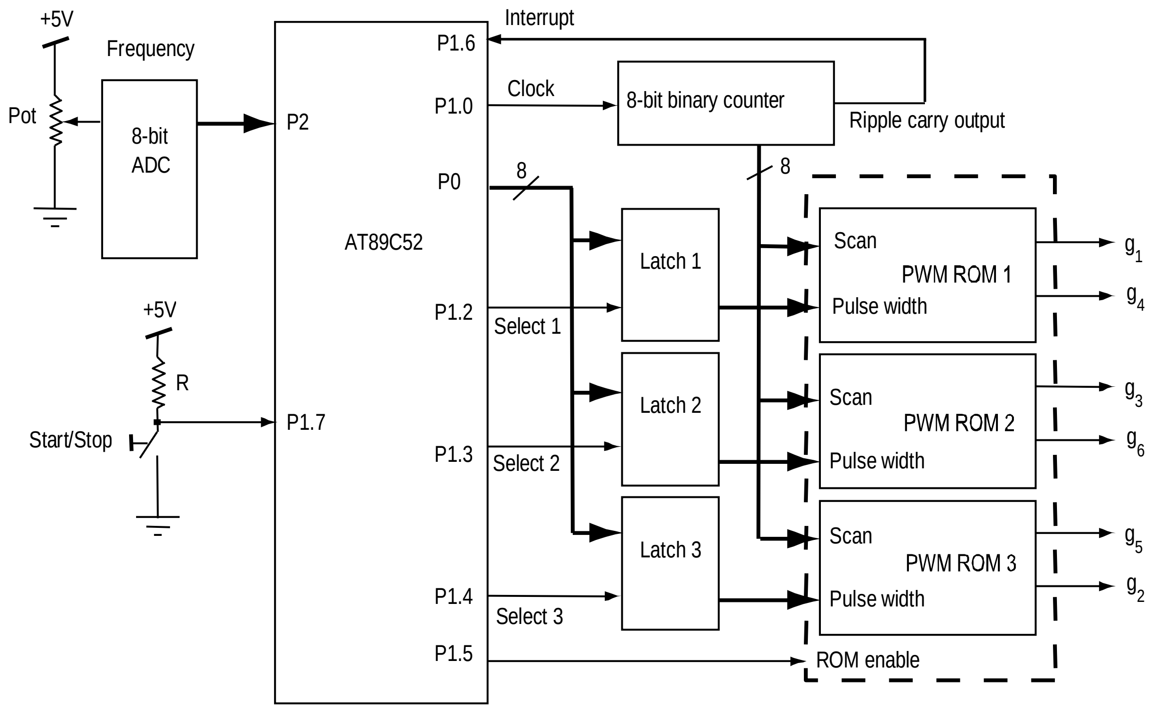

Fig. 2. An AT89C52 microcontroller based scheme for three-phase PWM pattern generation using three ROM lookup tables and a scan counter.

The microcontroller calculates the scan clock frequency and sets the corresponding RCAP2L and RCAP2H values for the onboard Timer 2. The Timer 2 of AT89C52 is configured in clock generator mode that gives the programmed clock output at P1.1. The scan counter is clocked from P1.1 and the ripple carry out from the scan counter is made to interrupt the microcontroller after every 256 counts that correspond to the total number samples in a carrier period. The PWM waveform pattern samples are sent to ROM lookup tables through port P0. The PWM patterns for three phases are sent through the same port P0 using multiplexing and latch principle. For each phase there is a set of latch and ROM lookup table, so that they can generate the PWM patterns independently.

3.1 Frequency Setting

The fundamental frequency of the PWM output patterns are dependent on the microcontroller clock frequency (f_{CPU}), counter clock frequency (f_{CLK}) and the number of carrier pulses per fundamental cycle (N_{max}). Since the counter clock (f_{CLK}) is derived from the microcontroller clock frequency (f_{CPU}) by a divide by counter (Timer 2 of the microcontroller), the output frequency (f) is given by

$$f = \frac{1}{N_{\max}} \frac{f_{CLK}}{256} \tag{14}$$

$$f_{CLK} = \frac{f_{CPU}}{4}\frac{1}{65536 - \text{DIVN}}$$

(15)

Combining (1) and (2), the DIVN value is given by

$$\text{DIVN} = 65536 - \frac{f_{CPU}}{1024 fN_{\max}} \tag{16}$$

$$\text{RCAP2H} = floor\left(\frac{\text{DIVN}}{256}\right) \tag{17}$$

$$\text{RCAP2L} = \text{DIVN - RCAP2H} \tag{18}$$

3.2 Integer Arithmetic Models for PWM Patterns

The microcontroller does not have floating point computation support. Hence, it can not compute sine functions directly. Rather, it has to be calculated using tailor series expansion method that require huge computation time. To reduce the computation complexity, (11) and (12) are simplified having only integer manipulations.

$$\begin{aligned} K_a(n) &= \frac{K_{\max}}{2} + M\left[\frac{K_{\max}}{2}\sin(n\omega T_s)\right] = \frac{K_{\max}}{2} + MS_a(n) = \frac{K_{\max}}{2} + \frac{M}{K_{\max}/2}\frac{K_{\max}}{2}S_a(n) \\ &= \frac{K_{\max}}{2} + \frac{2M_{tag}}{K_{\max}}S_a(n) \end{aligned} \tag{19}$$

where, $S_a(n) = \frac{K_{\max}}{2}\sin(n\omega T_s)$ and $M_{tag} = M\frac{K_{\max}}{2}$. For phase B,

$$K_b(n) = \frac{K_{\max}}{2} + \frac{2M_{tag}}{K_{\max}}S_b(n) \tag{20}$$

where, $S_b(n) = \frac{K_{\max}}{2}\sin(n\omega T_s - 2\pi/3)$ and $M_{tag} = M\frac{K_{\max}}{2}$.

The values of $S_a(n)$ and $S_b(n)$ are computed off line, rounded to integer numbers and stored in lookup table inside the program code of the AT89C52. The M_{tag} value is chosen from the ADC output that is always an integer. Hence, the computations of $K_a(n)$, $K_b(n)$ using (19) and (20) involves only integer arithmetic that can be done efficiently with AT89C52 in real time.

3.3 Real Time PWM Pattern Generation

PWM patterns having 256 different duties are stored in the PWM ROMs as a lookup table. The microcontroller computes the PWM patterns on carrier cycle basis and the real time PWM patterns are generated in the ROM data bus using the following algorithm:

Algorithm for PWM Pattern Generation

```
Step 1.   Set P1.5 = 1 to disable the PWM ROMs.
Step 2.   Initialize n = 1.
Step 3.   Read P2 and set $f_{tag}$ = P2.
Step 4.   Choose $N_{max}$ and $f_{clock}$, set RCAP2L and RCAP2H.
Step 5.   Compute $K_a(n)$, $K_b(n)$ and $K_c(n)$.
Step 6.   Set P0 = $K_a(n)$ and P1.2 = 0 ($K_a(n)$ data to Latch 1).
```

Step 7. Set P0 = $K_b(n)$ and P1.3 = 0 ($K_b(n)$ data to Latch 2).
Step 8. Set P0 = $K_c(n)$ and P1.4 = 0 ($K_c(n)$ data to Latch 3).
Step 9. Wait for interrupt at P1.6 and goto step 10 if interrupt occurs, otherwise loop to
Step 10. Enable the data ROMs by setting P1.5 = 0.
Step 11. Read the shut down status at P1.7.
Step 12. If P1.7 = 0, disable the data ROMs (set P1.5 = 1) and goto step 16, else goto step .
Step 13. Increment n, n = n+1.
Step 14. If n>Nmax, set n = 1.
Step 15. Goto step 3.
Step 16. Halt operation.

3.4 Performance simulation

The performances of the proposed PWM scheme are studied through simulation with a three phase voltage source inverter. The inverter is shown in Fig. 3 and drives a star connected inductive load with insulated neutral. The switching functions for phases A, B and C are S_a, S_b and S_c respectively. The switching function and associated transistor status are shown in Table 1. Each switching function has two level statuses; either +1 or -1, where, +1 drives the top transistor and -1 drives the bottom transistor of a leg of the inverter.

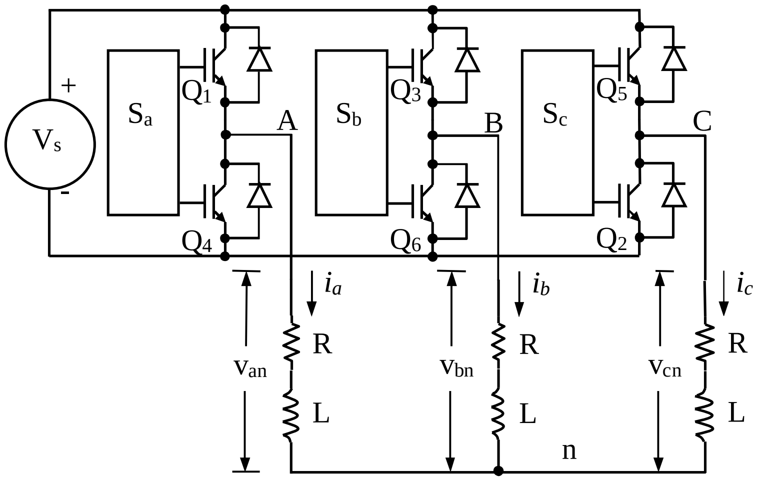

Fig. 3: Three phase voltage source inverter connected to star connected R-L load with insulated neutral (Sa, Sb and Sc are two level switching functions having magnitudes +1 and -1, for driving the associate top and bottom transistors in a leg).

The PWM switching patterns for three phases and inverter output phase voltages are shown in Fig. 4 for supply voltage V_s = 600V (1 pu), modulation index M = 1 at f = 50 Hz, and carrier frequency f_s = 1.2 kHz. The corresponding Fourier frequency spectrum of the PWM switching function (S_a) and line to neutral (phase) voltage v_{an} are shown in Fig. 5. The noticeable point is that although the fundamental frequency amplitude in S_a is 1 pu (equal to M), the fundamental phase voltage output is 0.5 pu, that confirms the analysis described earlier. The inverter output currents (three phases) and Fourier frequency spectrum are shown in Fig. 6 for an R-L load of 10 ohms at 0.8 pf lagging. Due to low pass action in the output current, the harmonics die out in the current spectrum (Fig. 6) although they are noticeable in the phase voltage spectrum (Fig. 5).

Table 1: Switching functions and transistor operating status for the three phase voltage source inverter.

Switching function	Switching status	Transistor status	
S_a (Leg a)	+1	Q_1 = ON	Q_4 = OFF
	-1	Q_1 = OFF	Q_4 = ON
S_b (Leg b)	+1	Q_3 = ON	Q_6 = OFF
	-1	Q_3 = OFF	Q_6 = ON
S_c (Leg c)	+1	Q_5 = ON	Q_2 = OFF
	-1	Q_5 = OFF	Q_2 = ON

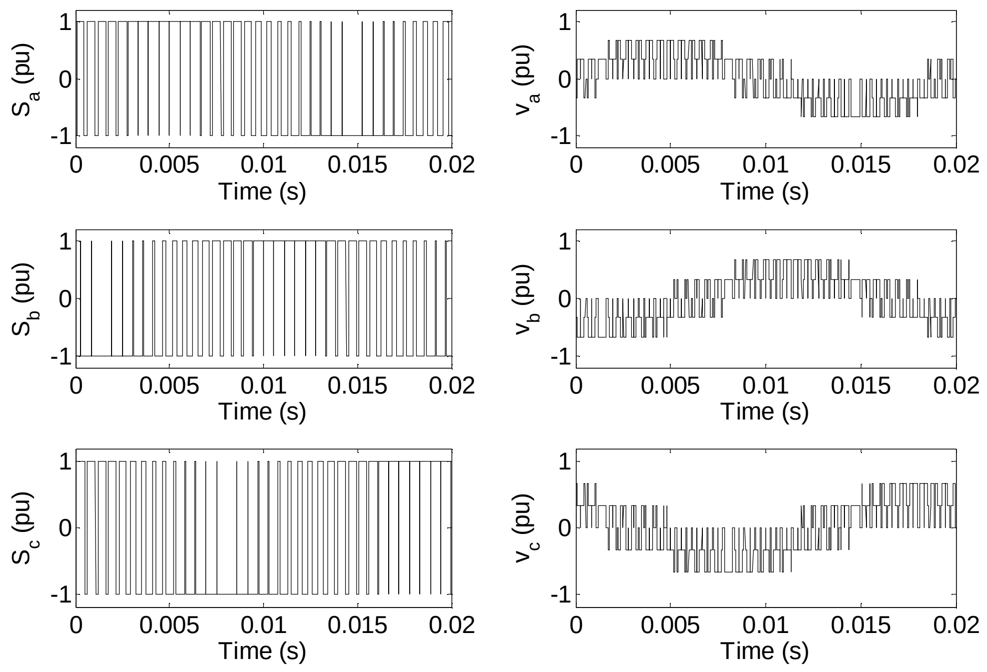

Fig. 4: PWM switching patterns at f = 50 Hz, f_s = 1.2 kHz, M = 1 for the three phase legs of the inverter and the phase voltages for star connected load with insulated neutral.

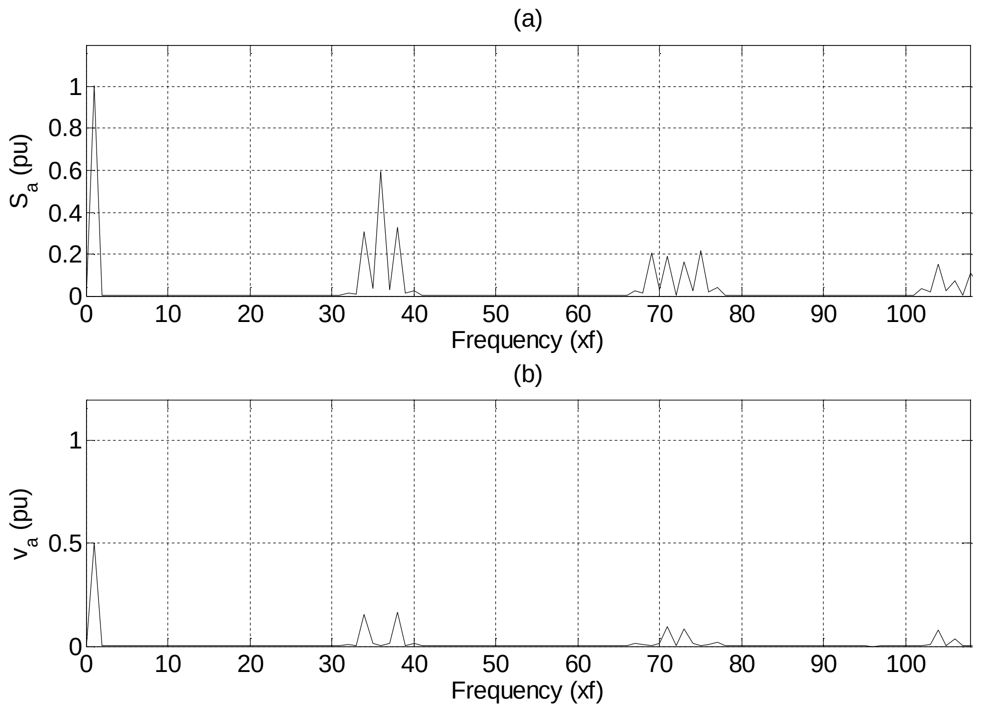

Fig. 5: Voltage spectrum of (a) switching function S_a and (b) the corresponding phase voltage v_a for the three phase inverter.

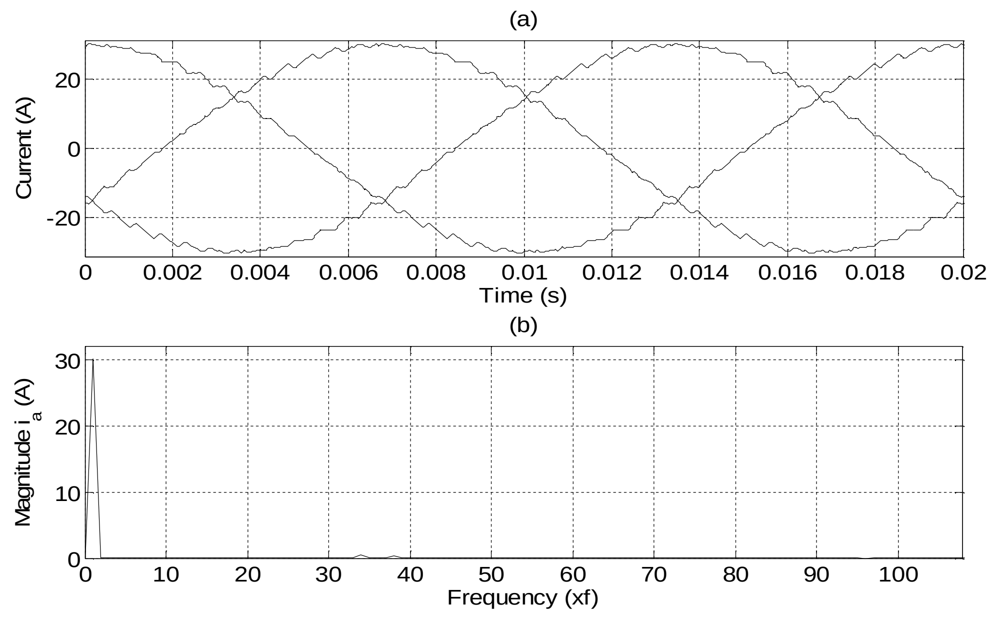

Fig. 6: Inverter output currents (three phases, a, b and c) and Fourier frequency spectrum at f = 50 Hz, M = 1, f_s = 1.2 kHz, V_s = 600V, Z = 10 ohm at 0.8 pf lagging.

4. CONCLUSIONS

A simplified technique of implementing regular sampled sine PWM is presented. The simplification involves integer arithmetic only that can be implemented with a low cost microcontroller at moderate speed. The simulation results show no deterioration in performance, although computations are all transformed from floating point into integer arithmetic. The hardware implementation scheme is compact, employing a memory minimized real time PWM pattern generator having less burden on the microcontroller. Due to low cost and high performance operating characteristics, the proposed PWM scheme is suitable for household, commercial and industrial applications for efficient control of voltage source inverters.

REFERENCES

[1] K. M. Rahman and M. Quamruzzaman, “A new hybrid PWM scheme for voltage source inverters,” in Proc. ICECE 2002, Dhaka, Bangladesh, pp. 332-335, 26-28 December 2002.

[2] S. R. Bowes, “Novel real-time harmonic minimized PWM control for drives and static power converters,” IEEE Trans. Power Electron., vol. 9, no. 3, pp. 256-262, May 1994.

[3] S. R. Bowes, “Advanced regular sampled PWM control techniques for drives and static power converters,” IEEE Trans. Ind. Electron., vol. 42, no. 4, pp. 367-373, August 1995.

[4] S. R. Bowes, “Regular sampled harmonic elimination PWM control of inverter drives,” IEEE Trans. Power Electron., vol. 10, no. 5, pp. 521-531, September 1995.

[5] K. M. Rahman, “PC controlled look-up table based PWM scheme for voltage source inverters and ac drives,” in Proc. ICCIT 99, Dhaka, Bangladesh, pp. 84-88, 3-5 December 1999.

[6] K. M. Rahman, “PC based regular sampled pulse width modulator for inverters and drives,” in Proc. ICCIT 99, Dhaka, Bangladesh, pp. 89-93, 3-5 December 1999.

[7] K. M. Rahman, “Analysis and implementation of PC based space vector PWM controller for VSI inverters and ac drives,” in Proc. ICCIT 2000, Dhaka, Bangladesh, pp. 49-54, 25-26 January 2001.

[8] K. M. Rahman, K. M. Z. Shams and A. H. M. Z. Alam, “FPGA implementation of space vector PWM controller for three phase voltage source inverters,” in Proc. ICCIT 2001, Dhaka, Bangladesh, pp. 154159, 28-29 December 2001.

[9] S. R. Bowes and S. S. Grewal, “Novel space-vector harmonic elimination inverter control,” IEEE Trans. Ind. Applicat., vol. 36, no. 2, pp. 549-557, March 2000.

[10] Y. S. Lai and S. R. Bowes, “A new suboptimal pulse-width modulation technique for per-phase modulation and space vector modulation,” IEEE Trans. Energy Convert., vol. 12, no. 4, pp. 310-316, December 1997.