Digital Implementation of Different PWM Control Techniques for Voltage Source Inverters Using a Microcontroller

التطبيق الرقمي لأنواع مختلفة من تقنيات التحكم في عواكس مصدر الجهد عن طريق تغيير عرض النبضة باستخدام متحكم متناهي في الصغر

W.A.M. Ghoneim and A. Lotfy
Arab Academy for Science, Technology and Maritime Transport.
walidghoneim1970@yahoo.co.uk, Alotfy@aast.edu

K.Maher
Alexandria Higher Institute of Engineering and Technology.
Khaled_mecha@yahoo.com

تقدم هذه الورقة البحثية مجموعة من الخوارزميات الرقمية والتي تمكن المتحكمات المتناهية الصغر من إنتاج أنواع مختلفة من تقنيات "العرض المتغير للنبضة". هذه الخوارزميات - و التي تتميز بضآلة الحيز الذي تشغله من الذاكرة - يمكنها أن تستخدم في أنظمة التحكم الرقمية للمواتير وعمليات التنظيم التلقائي للجهد و غير ذلك من تطبيقات تقنية العرض المتغير للنبضة. تم إجراء محاكاه لأنواع مختلفة من تقنيات العرض المتغير للنبضة مثل: الخطي الجيبي و الجيبي المعدل و الجيبي المحقون بالتوافقية الثالثة وفضاء المتجه وذلك باستخدام برنامج الماتلاب / سيمولينك . بعد ذلك تم إختبار الخوارزميات المذكورة معملياً للتأكد من فاعليتها كما أجريت أيضا مقارنة بين الأنواع المذكورة آنفا من حيث حجم البرنامج والوقت اللازم للتنفيذ و تردد التبديل/التقطيع و الامكانيات المستخدمة في المتحكم و التوافقيات الموجودة في جهد الخرج.

Abstract: **This paper presents efficient, code-optimized and low memory sized algorithms that enable the usage of general purpose affordable microcontrollers in the generation of various types of PWM control techniques. The integration of both forms cost-efficient and reliable digital control systems to be used in drivers, automatic voltage regulators and other PWM-based applications. Simulations of various types of discretized PWM techniques, such as stepped sinusoidal PWM, modified sinusoidal PWM, 3rd harmonic injection sinusoidal PWM, and space vector PWM, were developed using MATLAB/SIMULINK. New digital algorithms for implementing such PWM techniques were developed, then practical implementation of a complete microcontroller-based, digitally-controlled voltage source inverter (VSI) was carried out. A comparison between algorithms implementing various PWM techniques is presented based on code size, time of execution, switching frequency and utilized features of the controller and output voltage harmonic components.**

Index Terms **– SPWM, SVPWM, Microcontroller, VSI.**

1. INTRODUCTION

Inverters convert the DC input voltage to an AC one with the desired magnitude and frequency. If the DC input voltage is constant, variable output voltage could be obtained by adjusting the inverter gain. This is achievable by pulse width modulation (**PWM**) signals applied to the gates of the power transistors of the inverter [1].

PWM techniques are generally based on comparing two signals, reference signal and carrier signal. The Intersection points of the two signals show commutation time of power electronic switches of the inverter [2], as shown in figure 1. The output is also mathematically represented in equation 1.

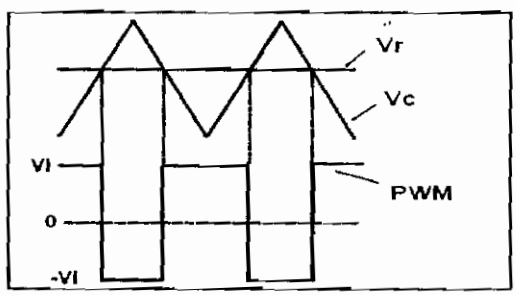

Fig.1 PWM principle.

$$\text{If } V_r > V_c \text{ then } V_o = +V_l \text{ else } V_o = -V_l \qquad (1)$$

The frequency of the carrier signal must be much higher than that of the reference signal so that the dominant harmonics are shifted to a higher frequency region, but this will eventually lead to an increase in the switching losses. One method to overcome such problem - as suggested in [3] - is to reduce the commutation number of the inverter by stopping the switching operation during a part of the each cycle. Keeping in mind that to obtain a sinusoidal line-to-line voltage with a PWM inverter, each phase voltage does not necessarily need to have a sinusoidal waveform as shown in figure 2.

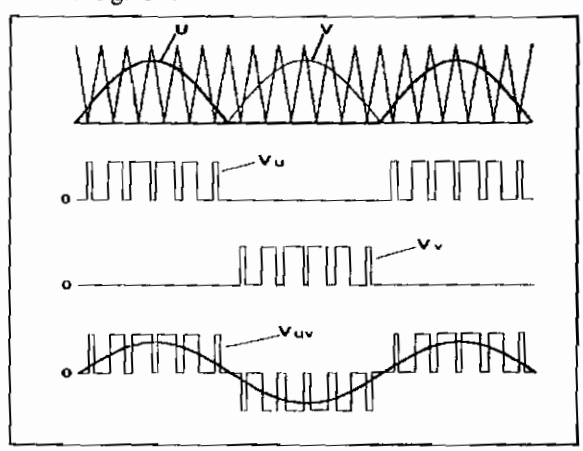

Fig.2 Minimizing switching losses by stopping the switching operation during a part of each cycle

Accepted August 23, 2011.

3. PWM TECHNIQUES ALGORITHMS

Mainly, two types of algorithms have been designed. The first is used for implementing the three sinusoidal-based PWM techniques (SPWM), while the second for implementing the space vector PWM technique (SVPWM).

3.1. SINUSOIDAL PWM TECHNIQUES ALGORITHM

In SPWM techniques, the reference signal (sine wave) will be sampled over a specific number of times (N), producing a known value at each sample. As the number of samples increases, the quality of performance also increases as we approach to the analog (continuous) situation. With the **AVR** microcontroller capabilities, the maximum number of samples that could be reached was **60** samples per cycle; this gives a **6°** interval between each sample.

The first step in implementing this algorithm was to determine the values of the **60** samples that represent the discretized reference signal. Actually, determining the values of **30** samples of the first half of the cycle only is just enough, as the inverter stops the switching action for the second-half of the cycle, see Figure 2. A *Simulink* model was used to calculate the discretized values of the reference signal for various SPWM-based techniques and then store them in a look up table to be used by the controller. The peak value of the reference signal was chosen to be (1024).

The AVR timer/counter will be programmed to work in *phase correct PWM* mode to generate the discrete carrier signal, which is a traingular wave with variable peak and frequency (f_c). Unlike analog signals, discrete signals depend on the constant timer clock frequency. Thus, the only way to alter the frequency of the carrier signal is to change its peak accordingly, which when reached the timer/counter will start to down-counting untill it resets, as shown in figure 8 and equation 2.

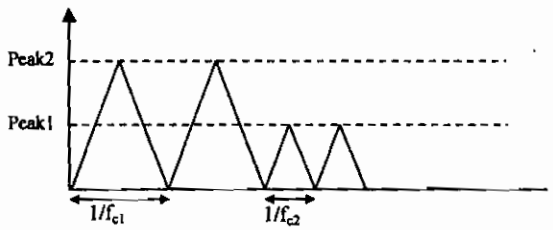

Fig8. Carrier signals with different frequencies

$$peak\ value = \frac{f_{timer}}{2 \times f_c} \qquad (2)$$

where, f_{timer} is timer clock frequency and chosen to be 2MHz. Since the carrier frequency (f_c) depends on the desired modulating signal frequency (f_m) according to:

$$f_c = N \times f_m = 60 \times f_m \qquad (3)$$

where N is the number of samples per cycle, choosen to be **60**, then for f_m ranging from 0 to 50 Hz, fc will range from 0 to 3000 Hz and accordingly the timer peak value ranges from 0 to 1666 counts.

When a timer underflow occurs (carrier signal value reaches zero), an interrupt routine is automatically called. In this sub-routine, a predefined pointer called (Pntr) will be incremented by one to refer to the next value of the discretized reference signal in the lookup table, and then loaded it into the compare register. The latter's value is continuously compared to the carrier value, and in case that the compare value was larger than the carrier value, the output is set, otherwise it is reset according to figure 9.

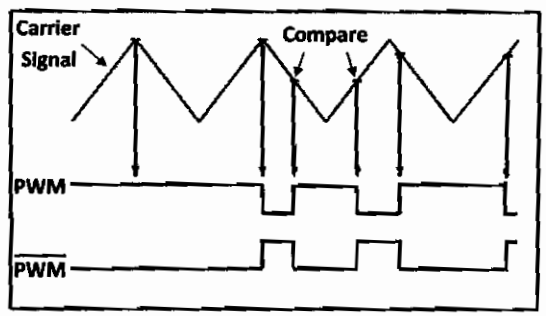

Fig 9. Inverting and non-inverting output of the timer in *phase correct PWM* mode.

Since the analog amplitude of the modulating index (M_a) ranges from 0 to 1. Hence, to avoid floating-point arithmetic, a discretized modulating index (M_{ad}) is chosen to range from 0 to 1024 according to equation 4:

$$M_{ad} = M_a \times 1024 \qquad (4)$$

The reference value, stored in the look up table, is weighted by M_{ad} to generate the modulated reference value (MRV), according to:

$$MRV = \frac{M_{ad} \times ref.signal\ value\ from\ the\ table}{1024} \qquad (5)$$

Finally, the compare value will be as follow:

$$Compare\ value = \frac{MRV \times carrier\ signal\ peak\ value}{1024} \qquad (6)$$

As known, the three phase PWM signals have to be generated with a phase shift of **120°**. This phase shift is established by shifting the pointer of the reference signal values by a specific number of steps (n) in the table, calculated as follow:

$$n = \frac{120}{360} \times N \qquad (7)$$

In this work, since N is **60** samples per cycle, the number of steps to be jumped (n) is 20.

Figure **10** shows the flowchart of the SPWM implemented code.

3.2. SPACE VECTOR PWM TECHNIQUE ALGORITHM

Since the upper power switches of the three phase inverter can only be On or Off, and since the lower ones are supposed to always be in the opposed state, there are only eight possible switching states. Six of them lead to non-zero phase voltages, and two interchangeable states lead to zero phase voltages [7].

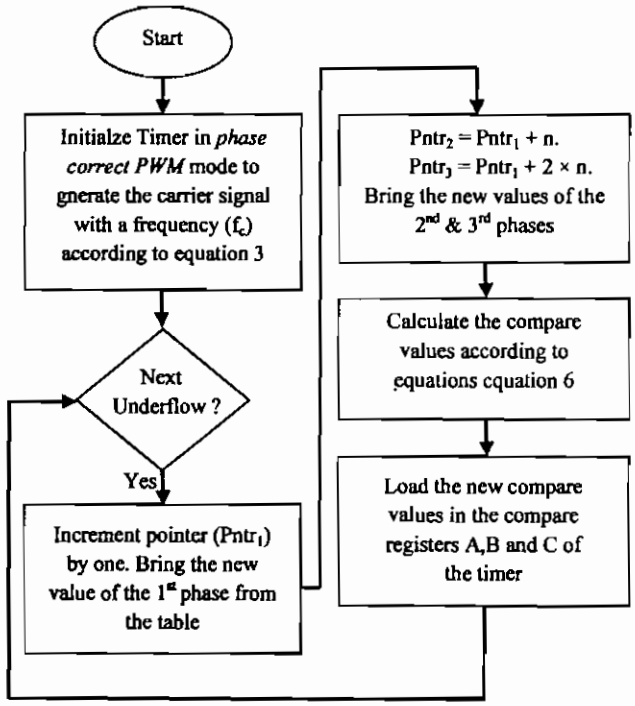

Fig10. Flow chart of SPWM algorithm.

Mapping the phase voltages corresponding to the eight combinations onto the d-q plane by performing a d-q transformation, results in six non-zero vectors and two zero vectors. The non-zero vectors form the axes of a hexagonal as shown in Figure 11. The angle between any two adjacent non-zero vectors is 60 degrees. The two zero vectors are at the origin and apply zero voltage to a load. The eight vectors are called the basic space vectors and are denoted by V_0, V_1, V_2, V_3, V_4, V_5, V_6 and V_7.The same transformation is applied to the desired output voltage vector to get thc desired voltage vector V_{out} in the d-q [8, 9].

The objective of space vector PWM technique is to approximate the output voltage vector V_{out} by a combination of the eight switching patterns as shown in Eq. 8. Eq. 8 says that for every PWM period T_s, the desired output voltagc V_{out} can be approximated by having the power inverter in switching pattern V_x and V_{x+60} for T_1 and T_2 duration of time respectively. Since the sum of T_1 and T_2 may be less than or equal to T_s the power inverter need to be put in '0' pattern for the rest of the period (T_0). Therefore Eq. 8 becomes Eq. 9.

$$\int_t^{t+Ts} V_{out} = T_s\, V_{out} = T_1 V_x + T_2 V_{x+60} \qquad (8)$$

$$T_s\, V_{out} = T_1 V_x + T_2 V_{x+60} + T_0(V_0 \text{ or } V_7) \qquad (9)$$

Where $T_1 + T_2 + T_0 = T_s$. Equations 10 and 11 show the evaluation of the time duration for each space vector T_1 and T_2.

$$T1 = 0.5 \times f_{system} \times Ts \times M_a \times \sin(60 - \theta) \qquad (10)$$

$$T2 = 0.5 \times f_{system} \times Ts \times M_a \times \sin(60) \qquad (11)$$

Where f_{system} is the system clock frequency in Hz.

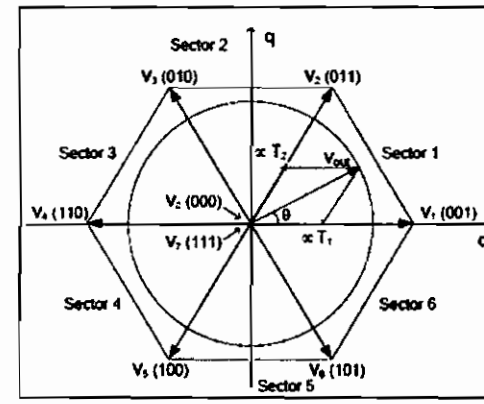

Fig11. Representation of the 8 possible switching configurations in d-q plane.

As shown in these equations the evaluation of the time duration for each space vector (T_1 and T_2) includes a calculation of the 'sine' of the present output vector angle (θ), which presents a challenge to the AVR microcontroller working on fixed-point arithmetic basis and in relatively low-processing speeds.

In order to overcome such a problem, the sine of angles ranging from 0° to 60° with a step angle of 0.05° are pre-calculated to be stored in a look up table. But, for more simplification, these float values will be multiplied by the fixed values of 0.5, f_{system} and Ts, as shown in equation 10 and 11, and then rounding the result to the nearest integer value. Hence, equations 10 and 11 are converted to equations 17 and 18 respectively and no floating-point arithmetic is needed any more.

The PWM period (Ts) is choosed to be 139 us, this gives a switching frequency of 7200 Hz which is close to the maximum allowable switching frequency of the Integrated Power inverter Module used in this work (which is 10 KHz), as will be shown in scction 4.

The AVR timer/counter will be programmed to work in *phase correct PWM* mode to generate the carrier signal, which is a traingular wave with a fixed frequency (f_c) of 7200 Hz. The timer also will be programmed to generate an interrupt routine whcn a timer underflow occurs (every T_s). In this sub-routine, the output vector angle (θ) will be incremented by:

$$\Delta\theta = 360° \times T_s \times f_m \quad \text{(in degree)} \qquad (12)$$

The previous formula would be more convenient when it is converted from degrees to number of steps in the look-up table. Thus θ will be incremented as follow:

$$\Delta\theta = \frac{360° \times T_s \times f_m}{\text{step angle of the stored table}} \quad \text{(in steps)} \qquad (13)$$

Substituting with the chosen values, we get:

$$\Delta\theta = \frac{360° \times 139 \times 10^{-6} * fm}{0.05°} = fm \quad \text{(in steps)} \qquad (14)$$