Advanced Regular-Sampled PWM Control Techniques for Drives and Static Power Converters

Sidney R. Bowes, Senior Member, IEEE

Abstract—Regular-sampled PWM techniques have been developed to reproduce the harmonic-elimination and harmonic minimization PWM characteristics. These new regular-sampled PWM control strategies significantly reduce the computational requirements for real-time microprocessor-based PWM implementation. This results in simplified and more efficient microprocessor software/hardware requirements, leading to real-time PWM generation with minimized harmonics, suitable for drives and uninterruptible power supplies.

I. INTRODUCTION

The control of modern adjustable speed drives and static power converters is increasingly based on the real-time digital generation of pulse-width modulated (PWM) waveforms using either microprocessors or application-specific integrated circuits (ASIC's).

Microprocessor-based PWM generation techniques offer significant advantages, particularly for optimizing the PWM waveform. However, the maximum switching frequency of the inverter can be severely limited by the computation time needed to generate the PWM switching times in the microprocessor, particularly when advanced PWM strategies are used; for example, the harmonic minimization [1] or harmonic elimination [2], [3] PWM strategies which involve considerable "on-line," "real-time" computing. Past development of these more advanced PWM strategies has involved considerable off-line computation, using a mainframe computer to determine the PWM switching angle characteristics. This has been necessary because the equations relating the PWM switching angles to voltage are transcendental and cannot be solved on-line by a microprocessor-based controller.

Significant off-line computation is therefore involved in developing these strategies [4]-[6], and this has delayed their use by industry in many applications. In contrast, the conventional natural-sampled [7]-[9] and regular-sampled [9], [10] PWM techniques are based on well-established and definable modulation processes, which can be readily implemented using standard hardware or microprocessor software implementations [10]-[19].

Recently developed novel harmonic elimination and harmonic minimized PWM control strategies, based on regular-sampling techniques [8], can be used to reduce significantly the computational requirements and provide more efficient real-time microprocessor-based PWM implementation [19].

II. REGULAR-SAMPLED PWM

Traditionally PWM control was accomplished using an analog technique known as natural sampling [7]-[9]. The equations describing the natural sampled switching angles are transcendental and therefore not suitable for microprocessor software implementation. This led to the development of a regular sampling technique in the early 1970's [8]. The basic principles of regular-sampled PWM techniques are shown in Fig. 1 for asymmetric regular-sampled PWM. Regular-sampling is a digital process of sampling a sinusoidal modulating wave "a," at regularly spaced intervals to produce sinusoidally weighted digital samples of the modulating wave, represented by "b," in Fig. 1.

As shown in Fig. 1, two samples of "b" per carrier cycle are generated. The first sample in the carrier cycle is used to sinusoidally weight digital samples of the modulating wave, represented by "b," in Fig. 1.

As shown in Fig. 1, two samples of "b" per carrier cycle are generated. The first sample in the carrier cycle is used to sinusoidally weight digital samples of the modulating wave, represented by "b," in Fig. 1.

$$r_k = T/2 + \left( MT/4 \right) \left[ \sin \left( kT/2 \right) + \sin \left( (k + 1)T/2 \right) \right]$$

0278-0046/95$04.00 © 1995 IEEE
Fig. 2. Four-timer PWM implementation.

where \( kT/2 \), and \((k + 1)T/2\), etc., are the sampling time instants, \( T \) is the sampling or carrier period, and \( M \) is the modulation index. Equation (1) can be used directly to generate regular-sampled PWM in “real-time” using an extremely efficient microprocessor-based software algorithm [10], [11].

The derivation of the switching equations describing asymmetric regular-sampled PWM has been given earlier [8], [9], and the main results are given in (2).

\[
\begin{align*}
\delta_k &= T/4 - MT/4 \sin(kT/2) \\
\delta_{k+1} &= T/4 + MT/4 \sin((k + 1)T/2) 
\end{align*}
\]  

These equations are executed in a software algorithm which simulates the modulation processes shown in Fig. 1, using switching angles \( \delta_k \) and \( \delta_{k+1} \) defined in (2). Phase A uses a modulating function \( F(T_k) = \sin(kT/2) \) and phases B and C use \( F(T_k) = \sin(kT/2 - 2\pi/3) \) and \( \sin((kT/2 - 4\pi/3) \) respectively.

A. Microprocessor Implementation

Four-Timer Carrier Cycle PWM: Both three-timer and four-timer microprocessor regular-sampled PWM implementations have been presented earlier [10], [11], and therefore only a brief resume of the four-timer regular-sampled PWM implementation is provided as a basis for appreciating the optimized PWM implementations to follow.

Fig. 2 illustrates the process of PWM waveform generation using asymmetric regular-sampling and a four-timer implementation [11]. The pulse-widths are calculated in the microprocessor using (2), and are then converted into time intervals using four programmable timers, one for each phase, and the fourth is used to generate interrupt signals to the microprocessor.

As illustrated in Fig. 2, at the beginning of each carrier cycle the microprocessor loads the phase \( A \), \( B \), and \( C \) timers with the prepulse times \( \delta_A(k) \), \( \delta_B(k) \), and \( \delta_C(k) \) respectively, and the fourth timer with \( T/2 \). When the timers for each phase finish timing, their corresponding outputs clock three J-K flip-flops which generate the PWM inverter gating signals.

When the fourth-timer completes timing out \( T/2 \), the microprocessor is interrupted and the three-phase timers are loaded with the post-pulse times \( \delta_A(k+1) \), \( \delta_B(k+1) \), and \( \delta_C(k+1) \), and the fourth-timer times out \( T/2 \) to complete the carrier cycle. This process is repeated for the following carrier cycles to generate the complete PWM wavelength.

As shown earlier [11], [15], the four-timer approach is more efficient compared with the alternative earlier three-timer implementation [10], requiring only two interrupts per carrier cycle rather than seven interrupts for the three-timer implementation, with a corresponding improvement in timing accuracy.

The calculation procedure, based on (2) for each phase, requires a total of six look-up table (LUT) samples, six multiplications, three additions, and three subtractions, a saving of six subtractions compared with the corresponding three-timer implementations.

Single-Timer Carrier Cycle PWM: The four-timer microprocessor implementation is relatively simple due to the extensive use of timers. If a single timer is used, then it is necessary to link the timer vector to the three phases that are switching [15]–[19]. The single-timer version reduces the hardware requirements at the expense of increased calculation; although, as will be shown, using the techniques developed in this paper, these calculations can be significantly reduced.

As shown in Fig. 3, the three phases are considered together and a timing vector calculated that holds the values \( t_0 \), \( t_A \), \( t_B \), and \( t_7 \) and a switching vector that defines the PWM output during these periods. As shown, the switching times at the beginning and end of the half carrier cycle are defined by times “\( t_0 \)” and “\( t_7 \)” respectively. “\( t_0 \)” refers to a period when all three inverter phases are low (vector 000 = 0) and \( t_7 \) corresponds to all three inverter phases being high (111 = 7). These “null vectors” represent nonswitching or an inactive inverter state. The time periods “\( t_A \)” and “\( t_B \)” correspond to one of the six possible inverter active states.

The output of the PWM gating signals to the inverter is via a memory-port mapped peripheral interface adapter (PIA), connected to the inverter gating circuitry [19].

The microprocessor implementation procedure loads the timer with a particular timing period and the PIA with the PWM output for that period. The timer interrupts the processor at the end of the period and the interrupt service routine loads the next time and PWM output to the respective devices.
The basic calculations to generate the prepulse and postpulse angles use (2) and the "difference angles" calculated by subtracting neighboring angles. A total of six look-up tables, six multiplications, three additions, and nine subtractions are required. It is possible to reduce the calculations using prior knowledge of the sampling function to determine the phase switching sequence directly, as illustrated in Fig. 4, and thus eliminate the need to sort the switching angles prior to calculating the timing vector. The benefits of this technique are most pronounced for the optimized regular sampled strategies, to be considered in Section III-C.

III. HARMONIC MINIMIZED REGULAR-SAMPLED PWM

Previous research [13] has determined the optimal modulating wave \( F(T_k)^* \), which minimizes the "Total Harmonic Current Distortion" (THD). The details of the procedures and techniques used in these investigations have been extensively reported earlier [13] and therefore only the main results are reported here.

The results showed [13] that the optimal modulating wave, \( F(T_k)^* \), consists of a fundamental, and a third harmonic which remained approximately constant at 25% for all frequency ratios and modulated depths. This provided a simple means of approximating the optimal modulating wave, \( F(T_k)^* \), using a sampled modulating signal of the form

\[
F(T_k) = \sin (T_k) + 1/4 \sin (3T_k)
\]

where \( T_k = kT/2 \).

In addition, it was shown [13] that using (3) a linear relationship exists between the fundamental of the PWM voltage and the modulation depth \( M \), up to approximately \( M = 1.2 \) p.u., thereby extending the PWM voltage range by approximately 20% before over-modulation and pulse-dropping occurs. These techniques have been further extended to permit the smooth transition from PWM to Quasi-Square Wave (QSW) operation [14].

It is important to note that the addition of the 25% third harmonic minimizes the THD, while maximizing the fundamental voltage [13]. The alternative approach suggested by some authors [20]-[22] of adding a 1/6 third harmonic only maximizes the fundamental and does not minimize the THD [13], [19], thereby producing an inferior harmonic performance.

Based on (3) the optimal PWM pulse-widths can be defined as:

Prepulse angle defining leading edge:

\[
\delta_k = T/4 - (MT/4)F(T_k).
\]

Postpulse angle defining trailing edge:

\[
\delta_{k+1} = T/4 + (MT/4)F(T_{k+1}).
\]

Equations (4) and (5) can be used as the basis for a simple and efficient microprocessor software algorithm for real-time generation of optimal PWM. Indeed, only minor modifications to the previously described (see Section II) PWM microprocessor implementation (10) and (11), involving only slight modifications to the modulating samples stored in ROM are required.

The three-phase modulating waveforms including the third harmonic are shown in Fig. 4.

During any particular 30º section, one of the three phases is at its most positive amplitude while another is at its most negative and the third is in between. For a particular carrier period, the two samples associated with each phase when applied in (4) and (5) will produce the widest pulse for the phase with the most positive sample, and a narrow pulse for that phase with the most negative sample. The characteristic of particular interest is that the two phases with the most positive and negative samples exhibit a significant degree of similarity within this 30º range.

It has been shown previously [19] that the magnitude of the sum of the most positive and most negative phase samples is less than 0.0962 (corresponding to a difference of 12%). The importance of this observation has been shown [19] to greatly simplify the microprocessor real-time calculations, as demonstrated in the following section.

A. Microprocessor Implementation

Four-Timer Optimized PWM: It has been shown previously [19] that the sum of the most positive and most negative phase samples is less than 0.096; this simplifies the calculations required and hence minimizes the pulse-width computation time.

For example, from (4)

\[
\delta_B(k) = T/4 - (MT/4)F_B(T_k).
\]

However, during the 0–30º period, as shown in Fig. 4, \( F_B(T_k) \) is approximately equal to \( -F_C(T_k) \), hence (6) becomes

\[
\delta_B(k) = T/4 + MT/4F_C(T_k)
\]

or

\[
\delta_B(k) = T/2 - \delta_C(T_k).
\]
The maximum angle error using this approximation is less than 3% as shown in Section III-B. Thus using (7) has eliminated the need to use a multiplication to calculate $\delta B(k)$, and therefore only two multiplications are now required for the three-phase prepulse angle calculations instead of the three multiplications required earlier. Note the same applies to the postpulse angle calculations.

This result can be generalized for all 60° intervals, or sextants, by observing the general form of the three-phase modulating waveforms shown in Fig. 4. For example, from inspection of (4) and (5), it can be seen from (4) that the first phase to switch in a particular carrier period will be the one with the most positive sample value of $F(T_k)$. The largest of the prepulse times, and hence last phase to be set, will be due to the most negative sample. The situation is reversed for the postpulse case, due to the positive sign in (3, 4), with the largest sample causing that phase to be reset last. The order of the phase switching can be determined by comparing the function values according to the observations above, together with the function values $F(T_k)$ of Fig. 4. The result is that the phase switching order is the same for samples taken within each sextant, as illustrated in Fig. 4. Thus, by determining where a particular sample occurs, the Look Up Table (LUT) holding the phase switching order can be accessed. Using this approach, only the prepulse times $\delta_X(k)$, $\delta_Y(k)$, for example, for the first and second phases to switch need to be calculated using (4), and the last phase to switch can be quickly calculated from $\delta_Z(k) = T/2 - \delta_X(k)$. (Note that $\delta_X$, $\delta_Y$ alternate between phases A, B, and C depending on which sextant is under consideration). A typical example for the first 30° interval is illustrated in Fig. 5.

The basic operations now required to calculate the three-phase pulse-widths have therefore been reduced to four table lookups, four multiplications, four subtractions, and two additions; thus providing a significant reduction in computation time.

**B. Single-Timer Optimized PWM**

It was shown earlier [19] that the sum of the most positive and most negative phase samples, shown in Fig. 4, is less than

0.0960. The importance of this observation can be seen from Fig. 3, where from inspection of this figure and (4) we deduce

\[
\begin{align*}
t_0 &= \delta_c(k) \\
    &= T/4 - (MT/4)F_c(T_k). \\
t_7 &= T/2 - \delta_B(k) \\
    &= T/4 + (MT/4)F_B(T_k).
\end{align*}
\]

Defining a percentage error $\varepsilon = t_7 - t_0$ normalized to the carrier period $T$ gives, from (8) and (9)

\[
\varepsilon = M/4(F_B(T_k) + F_C(T_k)) \times 100%.
\]

Since $[F_B(T_k)] + [F_C(T_k)]$ has a maximum value of 0.0962, then (10) becomes $\varepsilon = 2.41M\%$, which gives an error of $0 < \varepsilon < 3.0\%$ for $0 \leq M \leq 1.273(\text{QSW})$. Thus if a third harmonic component is added to the modulating function then the times, $t_0$ and $t_7$, become very nearly equal. This strategy needs only sample two phases and the third phase modulation is achieved by equating $t_0 = t_7$. The knowledge of the switching sequence or "phase order," shown in Fig. 4 is used to select the first two switching phases and thus calculate $t_0$ and $t_A$, from which $t_B$ and $t_7$ can be quickly calculated with appropriate subtractions.

The fundamental operations required have therefore been reduced to four table lookups and four multiplications, emphasizing the computational savings using the $t_0 = t_7$ approximation.

The harmonic performance using either the single-timer $t_0 = t_7$ approximation, or the equivalent four-timer optimized PWM implementation, is shown in Fig. 6. Fig. 6 gives typical experimental voltage and current waveforms and spectra for typical high-voltage operating conditions corresponding to the number of switchings per quarter cycle $N = 4$ and 2. As shown in these figures, the excellent low harmonic current spectrum confirms the effectiveness of the new PWM strategy for applications requiring harmonic minimization.
BOWES: ADVANCED REGULAR-SAMPLED PWM CONTROL TECHNIQUES FOR DRIVES AND STATIC POWER CONVERTERS

IV. SIMPLIFIED HARMONIC ELIMINATED REGULAR-SAMPLED PWM

Whereas the optimal PWM technique is based on the minimization of THD, the harmonic-elimination PWM technique is used to eliminate selected harmonics [2], [3]. Extensive investigations have shown [17] that it is possible to fit a regular-sampled PWM modulation process to the harmonic-elimination switching characteristics. To fully appreciate these developments it is necessary to refer to earlier papers [17], [18] and only the main results are summarized below.

It is possible to closely reproduce the harmonic-elimination PWM using a modified version of regular-sampling [17] as shown in Fig. 7. This figure illustrates how the regular-sampling process can be applied to reproduce the harmonic-elimination switching angle characteristic, using three switching angles per quarter cycle, \( N = 3 \), as a typical example. Noting that the switching-angle against fundamental voltage characteristics are produced using computer-based numerical techniques to eliminate \((N - 1)\) harmonics, as described previously, [2], [3], [17].

Fig. 7 shows the switching angles in the first quarter-cycle of the PWM waveform, with the center of a pulse (which corresponds to the regular-sampling instant) coincident with the angular separation given by \( t_k = kT \) at zero voltage, which for \( N = 3 \), from (11), gives \( T = 30^\circ \). The regular-sampling process can be applied and the degree of modulation \( \Delta \alpha_k \) is sinusoidal, for both odd and even \( k \) and all odd \( N \geq 3 \), up to approximately 0.8 p.u. fundamental voltage [17].

Therefore, two sinewaves suitably phase displaced can be regularly sampled, as shown in Fig. 7(b) and (c), to produce the degree of sinusoidal modulation required \( \Delta \alpha_k \). This is subsequently used to asymmetrically modulate each edge of the pulse to produce the quarter-wave symmetric PWM waveform shown in Fig. 7(d). This results in the following switching angle [17]:

For the leading edge with \( k \) odd

\[
\alpha_k = (k + 1)T/2 - MT/2 \sin[(k + 1)/2 + \phi_2] \tag{11}
\]

and for trailing-edge with \( k \) even

\[
\alpha_k = kT/2 + MT/2 \sin[kT/2 + \phi_2] \tag{12}
\]

where

\[
\phi_1 = NT/2 \quad \text{for} \quad k \text{ odd} \\
\phi_2 = T/4 \quad \text{for} \quad k \text{ even} \tag{13}
\]

Investigations [17] also confirmed that the modulation index, \( M \), was equal, in per unit terms, to the fundamental of the PWM voltage, thus providing a simple direct means of voltage control.

The switching angle errors are less than 0.2° and harmonics less than 4% with the majority less than 2%, and decrease with increasing \( N \) as the regular sampling process more accurately reproduces the harmonic elimination process [17]. This is illustrated in Fig. 8 for \( N = 5 \) and 7 using constant \( \phi_1 \) and \( \phi_2 \) as defined in (13). It is also possible to further “fine-tune” phase-angles \( \phi_1 \) and \( \phi_2 \) to more accurately reproduce the “exact” harmonic-elimination switching angles [17] and reduce the “ideally eliminated” harmonics to less than 2% with the majority less than 1%. These techniques have been further extended to \( 0.8 < M < 1.273 \) p.u. and permit the smooth transition from PWM to quasi-square wave operation [17].

A. Microprocessor Implementation

Harmonic elimination regular-sampled PWM can be implemented on a carrier-cycle basis using both four-timer and single-timer microprocessor implementations of the kind described earlier [18] in Section II-A.

An alternative approach uses the special characteristics of the harmonic elimination PWM three-phase waveforms to simplify the microprocessor implementation [18]. A typical harmonic elimination PWM waveform is shown in Fig. 7, using \( N = 3 \) as a simple illustrative example. As illustrated in the figure, the complete PWM cycle (phase A) can be subdivided into six 0–60° periods (sextants), with PWM switching

Fig. 7. Regular-sampled harmonic elimination PWM.

Fig. 8. Harmonics for constant \( \phi_1 \), \( \phi_2 \), and \( M = V_1 \).
in the 0–60° and 120–180° periods and no switching in the 60–120° period. Due to the half-wave symmetry, this switching sequence is repeated in the second half cycle of the PWM waveform.

Close inspection of Fig. 9 shows that within the first sextant phase A is switching the 0–60° switching pattern, phase B the 120–180° switching pattern, and phase C the 60–120° switching pattern. These patterns are rotated between each of the phases in subsequent sextants.

The two-timer and single-timer sextant approach [18] requires the three phases to be considered simultaneously rather than individually. Using the switching angle (11) and (12), the $N$ switches per quarter cycle can be calculated in the 0–60° range and, because of the quarter-wave symmetry, can be used to generate the complete three-phase PWM waveforms. However, because the $N$ switchings only take place in the 0–60° range and the complete PWM waveform is symmetrical, it is only necessary to consider the first one-sixth of a cycle of the three-phase PWM waveform, as shown in Fig. 9.

For the three-phase system with $N$ switches per quarter cycle (per phase), there will be a total of $2N + 1$ switches in any one-sixth cycle. These $2N + 1$ switching angles will generate switching on only two phases with the third phase switched “on” (or off) for a one-sixth cycle. This switching pattern is repeated for the complete PWM cycle, with each
phase switching alternately from one switching pattern to each of the others in each consecutive one-sixth cycle period. Thus, to implement this PWM strategy using a single-timer implementation requires only \(2N + 1\) timing intervals \(t_k\) (timing vector), as shown in Fig. 9, consisting of the 0–60° and 120–180° switching patterns to be combined into one LUT.

Fig. 10 shows typical experimental voltage, current waveforms, and spectra for harmonic elimination regular-sampled PWM corresponding to figures, the resulting harmonics are extremely low, confirming generation of harmonic elimination and harmonic minimized sampled control, thus providing for wider industrial application.

While this paper has concentrated on advanced regular sampled control technique for voltage source inverter applications, it is important to note that similar developments have taken place for current source inverter applications [23–25].

V. CONCLUSIONS

All the problems associated with the development and generation of harmonic elimination and harmonic minimized PWM can be eliminated using well-established regular-sampled PWM techniques. The new PWM strategies can be used for a wide range of PWM controlled inverter applications, including drives, uninterruptible power supplies, static frequency changers, etc. Using the new PWM strategies, efficient four-timer and single-timer microprocessor implementation can be developed which allow PWM to be achieved in real-time with the minimum of on-line calculation. The regular-sampled PWM microprocessor approach has eliminated the need for any “off-line” calculations or extensive use of LUT’s and interpolation between LUT’s for voltage control, thus providing for wider industrial application.

While this paper has concentrated on advanced regular sampled control technique for voltage source inverter applications, it is important to note that similar developments have taken place for current source inverter applications [23–25].

REFERENCES