Computer-Aided Design and Application of Sinusoidal Switching Patterns

Steven R. Shaw Deron K. Jackson Timothy A. Denison Steven B. Leeb

Abstract— **This paper discusses the design of discrete switching sequences for synthesizing sinewaves in power-electronic circuits and drives. The discrete-level sinewave approximations described in this paper can be implemented with reduced switching losses, in comparison to typical pulse-width modulation patterns, and with user-selectable harmonic content. We present two design algorithms, including an algorithm based on simulated annealing. The algorithms allow the engineer to select switching sequences that meet desired objectives or characteristics in almost any context. In addition, we demonstrate the use of these sequences experimentally in a practical application.**

I. Background

The energy conversion efficiencies achievable with switching power circuits have been touted for decades [1]. One common switching strategy for power-electronic circuits is fixed-frequency pulse-width modulation (PWM). With PWM, a duty-ratio command is varied to adjust features of the switched waveform. In an inverter circuit for a motor drive, for example, the duty ratio might be varied periodically to create a switched waveform whose short-time average value closely follows the value of a reference sinewave. Many switch cycles will occur during one cycle of the reference sinewave. Because the frequency of the PWM approach is fixed, a set number of switch transitions occur during each cycle of the fundamental.

Switching transitions contribute to conversion losses and inefficiency. The losses result from switch dissipation – particularly in circuits that do not employ resonant, soft-switched transitions – and also from gate- or base-drive losses. In addition, the PWM waveform will have significant harmonic energy around and above the switching frequency. In some cases the switching-frequency harmonic may be larger in amplitude than the component at the frequency of the sinusoidal PWM reference. A low-pass filter can be used to attenuate switching-frequency harmonics, but low-frequency harmonics in the passband still affect the waveform quality.

Several methods have been proposed to minimize the low-frequency harmonic content of PWM circuits [2], [3], [4]. The methods use analytical techniques, such as Fourier or Walsh function analysis, to express the harmonic content in terms of the PWM duty ratio. The resulting equations are then solved to minimize distortion. In many cases the resulting fixed-frequency PWM waveforms are discretized in time so that they may be implemented using digital circuitry [5]. These "programmed" PWM waveforms are essentially long, repeating, discrete-time binary switching sequences which serve as approximations to a sinewave.

In general, switching sequences can be created with switch transitions that do not occur at a fixed frequency. A technique in [1], which eliminates harmonics by zeroing or "notching" a square wave, is one example. Other possibilities and design criteria are described in [6] and [7]. For example, a fixed-length sequence can be used to construct a discrete approximate to a reference sinewave. The number and position of the "ones" and "zeroes" can be selected to minimize the distortion without regard to fixed-frequency constraints. In general, it is possible to create low distortion sinewave approximations that use fewer switch transitions per cycle than a comparable PWM patterns [5]. It is also possible to tailor or place the harmonic content of a switching sequence to suit particular applications. The bit patterns of the binary switching sequence can be stored in a memory and clocked out at a fixed rate or generated in real time using a simple state machine or microcontroller.

The achievable amplitude of the target fundamental is proportional to the percentage of ones in the binary sequence. Therefore, it is possible in many cases to simply search all M-point sequences with the desired number of ones. A simple algorithm is described in the next section to determine the harmonic content of a specified bit sequence. Analyzing long switching sequences, however, quickly becomes unwieldy; the multiplicity of possible bit patterns grows exponentially with M. A simulated annealing technique is presented which allows the designer to quickly satisfy requirements such as the number of transitions, harmonic content, or other metrics of interest. The application of annealed switching sequences in an AC inverter is presented.

II. Pattern Design

A switching sequence could in principle be created by assuming any number of discrete output levels. For typical power-electronic inverter circuits, two convenient assumptions regarding output discretization are shown in Fig. 1. The tri-level waveform could be produced by a full-bridge inverter circuit, and the bi-level waveform could be created by a half-bridge inverter. In practice, the tri-level waveform is assumed to have half- or quarter-wave symmetry. The symmetry simplifies inverter construction and improves efficiency by allowing one leg of the inverter to switch only once per half-cycle. The symmetry assumptions also reduce the design of a tri-level waveform to one of a bi-level subset. The complete tri-level sequence can be constructed from the bi-level subset. For example, the second half-period of a sequence with half-wave symmetry is a replication of the

The authors are with the Laboratory for Electromagnetic and Electronic Systems, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

0-7803-4856-7-8/98/$10.00 © 1998 IEEE

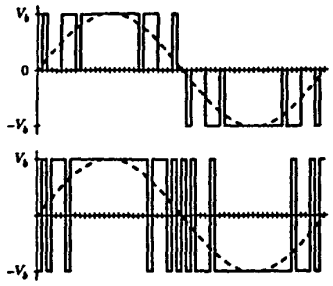

Fig. 1. Bi-level and tri-level sinewave approximations.

first half-period with a sign inversion. This symmetry has the added benefit that all even harmonics are eliminated from the final sequence. Quarter-wave symmetry additionally eliminates any cosinusoidal content from the switching sequence. The algorithms presented here will assume quarter-wave symmetry. Although we make symmetry assumptions in this paper, the techniques can be extended to more general sequences.

For illustration purposes, we will focus on the design of a tri-level sequence x with quarter-wave symmetry and a length $M = 4N$. Because the waveform is symmetric, the problem is to find the first quarter-period of length N, which consists only of ones and zeros, with fixed energy

$$E = \sum_{i=0}^{N-1} x_i \tag{1}$$

that minimizes some loss function $V(x)$. The loss function $V(x)$ is typically a weighted function of the harmonic content of the sequence x and the number of transitions in the sequence. Other criteria may also be included in the loss function.

A candidate switch sequence can be analyzed as a superposition of M identical square bits shifted in time. In the frequency domain, the harmonic content of the sequence is a superposition of sinc functions [8],

$$\mathcal{F}(f) = e^{-j\pi fT} \sum_{i=0}^{M-1} x_i e^{-j2\pi fTi} \frac{\sin(\pi fT)}{\pi f}, \tag{2}$$

multiplied by a phase associated with each bit's relative time offset. The factor x_i is the trinary amplitude that determines whether the i-th pulse amplitude is one, zero, or minus one.

The switching sequence x will be periodic in time with a period MT, where T is the duration of one bit in the sequence. The harmonic content is, therefore, multiplied by a series of delta functions,

$$\mathcal{F}(f) = \tag{3}$$
$$e^{-j\pi fT} \sum_{i=0}^{M-1} x_i e^{-j2\pi fTi} \frac{\sin(\pi fT)}{\pi f} \frac{1}{MT} \sum_k \delta(f - \frac{k}{MT}).$$

For any particular harmonic frequency $f = \frac{k}{MT}$ of the fundamental frequency $\frac{1}{MT}$, both the sinc and phase term $e^{-j\pi fT}$ are common to all terms in the summation over i. For purposes of determining the presence or absence of frequency content at a given harmonic frequency, these terms can be removed from further consideration. Assuming a periodic sequence, the analysis of the presence or absence of the higher harmonic content of a periodic switching sequence reduces to the discrete-time Fourier series:

$$X[k] = \frac{1}{M} \sum_{i=0}^{M-1} x_i e^{-\frac{j2\pi ki}{M}}, \tag{4}$$

where k is the harmonic frequency index and i is the index of a bit in the sequence. To eliminate the presence of a particular harmonic k, the x_i must be chosen such that the sum of the phases adds to zero.

Two other conditions can simplify the analysis. First, if the sequence has half-wave symmetry, then all even harmonics are zero. Second, if the sequence has quarter-wave symmetry, then only sinusoidal components are present, and only the first N points of the M-point sequence must be considered in the frequency content summation, as follows:

$$X[k] = \frac{1}{M} \sum_{i=0}^{N-1} x_i \sin(\frac{2\pi ki}{M}) \tag{5}$$

for *odd values* of k. Designing the symmetric switching sequence is, therefore, reduced to selecting the first N values of x_i to suppress higher-order harmonics of concern.

The analysis of the phase summations is compactly solved using matrix multiplication. A matrix,

$$B = \begin{pmatrix} 0 & 0 & 0 & \cdots \\ 0 & 0 & 1 & \cdots \\ 0 & 1 & 1 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \tag{6}$$

represents all possible N-length sequences containing a specified energy: each row is one possible quarter-period of a full switching sequence. A second matrix,

$$P = \frac{1}{M} \begin{pmatrix} \sin(0) & \sin(0) & \sin(0) & \cdots \\ \sin(1\Phi) & \sin(3\Phi) & \sin(5\Phi) & \cdots \\ \sin(2\Phi) & \sin(6\Phi) & \sin(10\Phi) & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \tag{7}$$

represents the phase content from each bit i in the switching sequence, where $\Phi = 2\pi/M$. Each row in P represents the phase offset of a single bit for a series of harmonics, and each column represents the phase shift of all N bits for a particular harmonic. The total number of columns in P is determined by the number of harmonics of interest.

Multiplication of these two matrices

$$Z = B \cdot P \tag{8}$$

gives the net phase coefficient for each possible bit pattern at the harmonic frequencies of interest. Each column in the resulting matrix identifies the net phase summation for a

particular harmonic; the rows correspond to the harmonic series for a specific bit pattern. To choose a switching sequence, a row in the matrix Z is chosen that has zeros in the appropriate columns (harmonics) of interest. Alternatively, a vector W of weights can be designed. This vector might correspond to magnitude samples of the transfer characteristic of the output filter in an inverter circuit. A switching sequence could be selected by finding the minimum entry of the vector $Z \cdot W$ to minimize harmonics passed by the output filter.

A computational challenge arises in actually specifying each row of the matrix B. For an N-point quarter-wave sequence, there are 2^N possible binary combinations. That is, with the worst case of no particular energy constraint, the matrix B has N columns and 2^N rows! For a particular energy constraint, i.e., if we limit B to include only rows with at least E ones, the number of possible rows r of the matrix B as a function of N and E is

$$r(N,E) = \frac{N!}{(N-E)!E!}. \tag{9}$$

Using the Stirling approximation [9],

$$r(N,E) \approx \sqrt{\frac{2}{\pi N}} 2^N e^{-2(E-N/2)^2/N}. \tag{10}$$

As N increases, and $E = \alpha N$, i.e., the normalized energy is constant, the multiplicity of candidate sequences increases exponentially with N. The "direct" pattern generation approach outlined in this section is superb for short length, e.g., 30 point, sequences on fast computers. For large sequences, e.g., $M = 1000$ or more, it quickly becomes necessary to find another design approach.

III. Simulated Annealing

The method of simulated annealing can be used to find an approximate solution to this exponentially increasing problem in sub-exponential time. The annealing strategy is analogous in some sense to the physical process of annealing metals, but the goal here is to minimize the loss function. The process begins by heating the system, making random perturbation in the loss function, until the movement is largely random. The system is then cooled, favoring downhill movements in the loss function toward a towards a minimum. Local minima are avoided by allowing positive movement depending on the "temperature" of the system.

A. Algorithm

In this application simulated annealing is used to minimize the harmonic content and the number of switch transitions for a proposed switching sequence. The strategy begins by generating an initial guess for the sequence x, given the length $M = 4N$ and the number of ones E. Although the approach is general, quarter-wave symmetry is assumed. Next, a change in the configuration of x is proposed, and the loss function for the changed configuration is evaluated. The change is either accepted or rejected based on a probabilistic criterion.

For the purposes of this paper, the loss function is the sum of two factors,

$$V(x) = V_D(x) + V_T(x) \tag{11}$$

where V_D is a distortion factor and V_T is a transition factor. The distortion factor V_D is defined as

$$V_D(x) = 100\sqrt{\frac{\sum_{k=2}^{2N-1} |X[k]|^2 |W[k]|^2}{|X[1]|^2}} \tag{12}$$

where the $X[k]$ are the discrete-time Fourier-series components of the periodic sequence x, and $W[k]$ is a user-selectable harmonic weighting function. The loss factor V_D represents the *weighted* total harmonic distortion (THD) of the sequence. The second loss factor V_T is defined as

$$V_T(x) = \begin{cases} W_T(T(x) - T_T) & \text{if } (T(x) - T_T) > 0 \\ 0 & \text{if } (T(x) - T_T) \leq 0 \end{cases} \tag{13}$$

where $T(x)$ is the number of up/down switch transitions in x, and T_T is a user-defined target for the number of transitions. If the number of transitions exceeds the target, the error is weighted linearly by W_T. However, if the number of transitions is less than the target V_T is set to zero.

Since the object is to find a sequence with E ones in its first quarter-period that minimizes V, only configuration changes that conserve E are considered. One such reconfiguration operator is

$$C(x): \begin{cases} x_{i1} = 1 - x_{i1} \\ x_{i0} = 1 - x_{i0} \end{cases} \tag{14}$$

where $i1$ indexes the l'th 1 and $i0$ indexes the m'th 0, and l, m are uniform random deviates. Simply put, the reconfiguration operator swaps the locations of a randomly selected zero and one. The new configuration, which leads to a loss function change ΔV, is accepted or rejected according to the Metropolis criterion $M(\Delta V, T)$ below:

$$M(\Delta V, T) = \begin{cases} \text{accept} & \text{if } \Delta V < 0 \text{ or } q < e^{-\frac{\Delta V}{T}} \\ \text{reject} & \text{otherwise} \end{cases} \tag{15}$$

where $q \in [0,1]$ is a uniform random deviate and T is the temperature of the system. The Metropolis criterion will accept all configuration changes that cause the loss to decrease, i.e., $\Delta V < 0$. However, the criterion will also accept a positive ΔV with a probability that increases with the temperature T. This probabilistic acceptance of positive ΔV helps the algorithm "jump out" of local minima. When a configuration change is accepted, the new configuration is compared against the previous "best" configuration. The new configuration replaces the "best" configuration if the loss has decreased.

The above steps are repeated S times at a given temperature. If at least one configuration change is accepted during S iterations, the temperature is decreased by a constant factor. However, if no changes are accepted, the temperature is "reheated" according to a prescribed schedule.

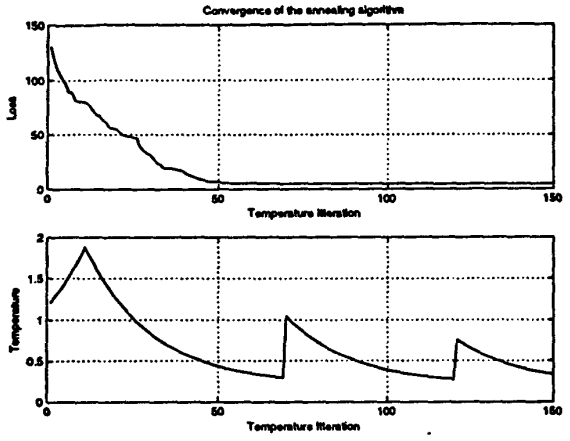

Fig. 2. Convergence of the simulated annealing algorithm.

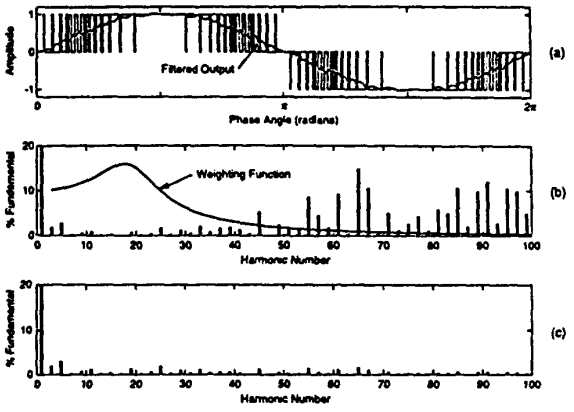

Fig. 3. Annealing result using second-order low-pass weighting. (a) The 1024-bit switching sequence. (b) Unweighted harmonic amplitude. (c) Weighted harmonic amplitude. (Note that the fundamental is clipped in (b) and (c).)

The process is illustrated in Fig. 2. The figure shows a typical convergence plot for a 1024-bit switching sequence The x-axis indicates the number of the temperature iteration. In this case, $S = 400$ configuration changes occur at each temperature. A startup procedure is employed to determine a reasonable initial temperature. The temperature T is increased by a constant factor until positive ΔV's, larger than about 5%, are accepted. This ensures that the initial temperature is high enough to induce large random deviations. After startup the temperature decreases by about 1% each step. When the number of acceptances reaches zero, the temperature is reheated to 0.5 and then 0.33 times its initial value.

The upper trace in Fig. 2 plots the decrease in the "best" value of the loss function as the algorithm progresses. The figure illustrates that the loss improves substantially during the first cooling cycle. In this case, the reheat cycles provide little additional decrease. Note, the convergence rate of the algorithm will vary depending on the loss function weighting and the length M of the sequence.

The annealing algorithm was implemented in the C programming language for speed. A MATLAB script `ANNEAL.M`, which is functionally equivalent to the C code, is provided in the Appendix. The output of the algorithm is shown in Figs. 3 and 4 for two example cases. These cases demonstrate the performance and flexibility of the algorithm. In both cases the algorithm searched for optimal 1024-bit switching sequences that approximate a sinewave with a fundamental amplitude of 1.0. This corresponds to a quarter-period length of $N = 256$ and $E = 163$.

In Fig. 3, the parameters of the loss factor V_T were set to $T_T = 108$ and $W_T = 400$. The transition weighting is high enough so that, effectively, only sequences with 108 transitions or less were considered. The harmonic weighting $W[k]$ was selected to match the roll-off of a second-order low-pass filter with a loaded Q of approximately 1.6. The algorithm produced the switching sequence in (a) with a final loss of 5.7%, which corresponds to the THD of the output waveform after the second-order filter. The design parameters for this waveform were selected to allow a direct comparison with similar PWM patterns presented in [4]. The PWM patterns in [4] were derived using Walsh function analytic harmonic elimination techniques. The published patterns switch at a fixed frequency with 108 switch transitions per cycle and a similar second-order output filter was assumed. A comparable discrete 1024-bit PWM pattern yields a filtered THD of 7.8%. The annealing result is slightly better in this case, probably due to the fact that the shape and Q of the filter were taken into account during the annealing process.

A second example in Fig. 4, demonstrates the flexibility of the annealing algorithm. In this case the the transition parameters were set at $T_T = 300$, and $W_T = 100$. The harmonic weighting $W[k]$ was set to unity for frequency harmonics at $k = 3$–9, 20–29, and 40–49. Outside of this range $W[k]$ was set to zero. As a result, the peak harmonic amplitude in the weighted frequency range is less than 1% of the fundamental. This demonstrates the ability to tailor the frequency content of the switching sequence almost arbitrarily.

B. AC Inverter Motor Drive

The simulated annealing algorithm was used to produce a 1024-bit switching sequence for use with a 3-kW H-Bridge inverter circuit constructed for these experiments and shown in Fig. 5. The MOSFET switches Q1 through Q4 are modulated to produce a tri-level voltage waveform measured across the load. The switching sequence was generated with $N = 256$ and $E = 96$, which yields a peak amplitude for the fundamental sinewave of about 59% of the DC input voltage level. This corresponds to approximately 110 VAC RMS given our DC input voltage of 265 volts. The frequency weighting $W[k]$ was selected to minimize the harmonic content up to $k = 40$. The resulting switching sequence was coded into a PAL-based finite-state machine. This yielded a simple digital circuit that interfaced easily

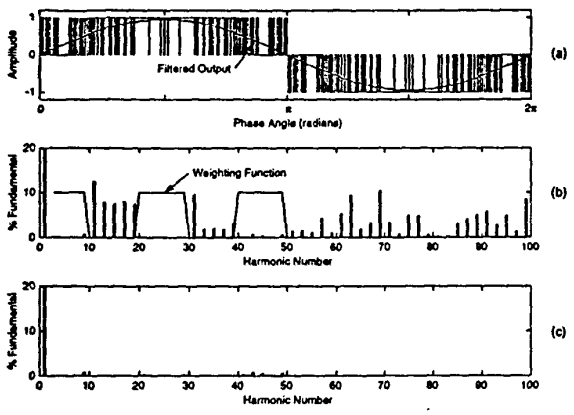

Fig. 4. Annealing result using a windowed frequency weighting. (a) The 1024-bit switching sequence. (b) Unweighted harmonic amplitude. (c) Weighted harmonic amplitude. (Note that the fundamental is clipped in (b) and (c).)

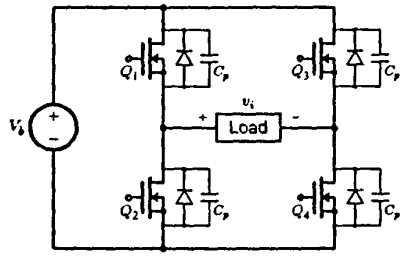

Fig. 5. A full-bridge single-phase inverter.

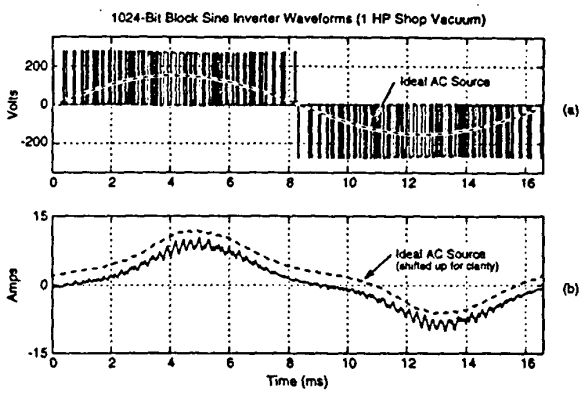

Fig. 6. Measured waveforms with the converter driving a 1 HP shop vacuum.

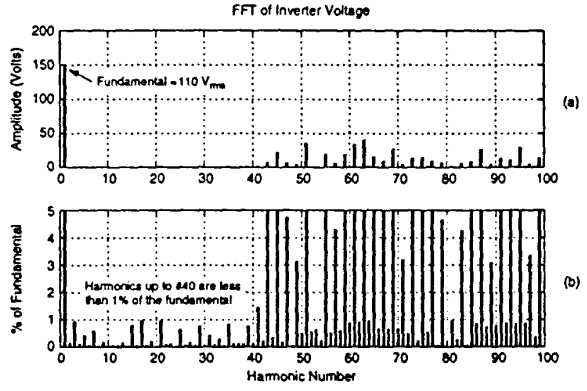

Fig. 7. Fourier transform of the inverter voltage. (a) FFT with amplitude in volts. (b) FFT showing harmonics as a percent of the fundamental. (Please note that the y-axis has been magnified.)

with the gate-drive signals for all four MOSFET switches in the inverter. Small, several hundred nano-second delays were built into the MOSFET gate-drive circuitry to prevent the possibility of shoot-through due to overlapping edges.

The experimental output of the system is shown in Fig. 6. A 1-HP shop vacuum was used as a load for the experiment, and a 2-mH inductor was wired in series with the motor to provide minimal external filtering. The top axis in Fig. 6 shows the inverter output voltage measured across the load. The dashed line is the fundamental component of the voltage waveform. The lower axis shows the current drawn by the shop vacuum. The solid line is the current measured at the inverter output. The dashed line is the ideal current, which was generated using an HP 6834 AC source to provide a near perfect sinusoidal voltage.

A discrete-Fourier-transform (DFT) analysis of the experimental voltage waveform harmonics is provided in Fig. 7. The figure demonstrates the extremely low harmonic content of the 1024-bit switching pattern. The first 40 harmonics have an amplitude smaller than 1% of the fundamental, and the higher harmonics are well distributed. The THD of the unfiltered voltage waveform is approximately 113%. However, the small amount of filtering provided by the inverter inductor and the load inductance results in only about 10% distortion in the current from its ideal. A capacitor could be added to the filter circuit to provide second-order filtering. A second-order low-pass filter centered at 2 kHz would reduce the voltage THD to approximately 5%.

IV. Discussion

A wide range of techniques have been proposed over the last thirty years for designing a variety of switched waveform sinewave approximations [2]-[7]. Many of the methods presented in these papers rely on analytic formulations. For example, Walsh basis functions have been applied to optimize specific criteria, such as low-frequency harmonic content, in switched sinewave approximations. This paper has reviewed two different methods for designing switching sequences that approximate sinewaves. Either method could be used to develop approximations with nearly arbitrary selection of the loss function. The direct method provides a global minimum or optimal solution for a given loss function, but is computationally intractable for large sequences. Large sequences with substantial freedom in pulse placement offer the most flexibility for meeting com-

plex optimization criteria, which might include constraints in both the frequency and time domains. To design these large sequences, we have found that the simulated annealing approach provides excellent optimization in very reasonable computation times. The simulated annealing algorithm has been shown capable of producing sequences with competitive or superior low-frequency distortion to other published switched sinewave approximations.

Switched sequences produced by the methods described in this paper have been applied to a variety of practical problems. The experiments shown here demonstrate an AC inverter with good low-frequency harmonic elimination. This was achieved using a 1024-bit switching sequence, and the distortion is comparable to, or better than, traditional PWM techniques. The experimental sequence chosen has only 300 switch transitions per 60-Hz cycle, which makes it efficient. A practical system might further refine the techniques by improving the optimization of the switching sequences and/or using a significantly longer sequence. It might, for example, be desirable in an inverter to push the significant frequency harmonics above the audible range, thereby relaxing the constraints on the output low-pass filter.

We have found that the methods described here are useful in lower power and signal processing systems. For example, textbook synchronous detection schemes generally modulate the signal of interest as a sinusoid and demodulate the received signal by linear multiplication with a second sinusoid. Practically, however, the multiplication is often implemented with a binary switching circuit [10], i.e., square-wave detection. Binary switching is inexpensive to implement and avoids the need for a high-quality analog multiplier. However, if the sinusoidal signal being detected is at the fundamental of the square wave, then the frequency content of the square wave at higher harmonics could result in high-frequency noise being reflected back into the measurement. We have selected switching sequences for demodulation that have very low harmonic frequency content in the neighborhood of the fundamental frequency. Demodulated high-frequency artifacts are significantly diminished in comparison to square-wave demodulation. This serves to reduce the sensitivity of the demodulation to noise and other high-frequency disturbances and/or allow the tracking bandwidth of the demodulation low-pass filter to be expanded, improving the sensor bandwidth.

References

[1] J. G. Kassakian, M. F. Schlecht, and G. C. Verghese, *Principles of Power Electronics*. New York: Addison-Wesley, 1991.

[2] H.S. Patel and R.G. Hoft, "Generalized technique of harmonic elimination and voltage control in thyristor inverters: Part I harmonic elimination," *IEEE Trans. on Industry Applications*, vol. IA-9, no. 3, pp. 310–317, May/June 1973.

[3] J. A. Asumada and R. G Hoft, "Microprocessor-based sinusoidal waveform synthesis using Walsh and related orthogonal functions," *IEEE Trans. on Power Electronics*, vol. 4, no. 2, pp 234–241, 1989.

[4] T.-J. Liang, R.M. O'Connell, and R. G. Hoft, "Inverter harmonic reduction using Walsh function harmonic elimination method," *IEEE Trans. on Power Electronics*, vol. 12, no. 6, pp. 971–982, November 1997.

[5] P. N. Enjeti, P. D. Ziogas, and J. F. Lindsay, "Programmed PWM techniques to eliminate harmonics: A critical evaluation," *IEEE Trans. on Industry Applications*, vol. 26, no. 2, pp. 302–316, March/April 1990.

[6] P. Eliams and A. D. Mansell, "Simulated annealing in the analysis of resonant DC link inverters," *IEE Proceedings on Electric Power Applications*, vol. 141, no. 3, May 1994.

[7] D. Lancaster, "Tech Musings," *Electronics Now*, vol. 66, no. 5, May 1995.

[8] W. Siebert, *Circuits, Signals, and Systems*, New York: McGraw-Hill, 1986.

[9] C. Kittel and H. Kroemer, *Thermal Physics*, New York: W.H. Freeman and Company, 1980, pp. 14–19.

[10] Linear Technology, "LTC 1043 Data Sheet," *Linear Databook*, 1990, pp. 11-5–11-30.

V. Appendix

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ANNEAL.M                                                                      %
%                                                                              %
% This script implements the simulated annealing                               %
% algorithm on MATLAB.  By adjusting the algorithm                             %
% parameters, the user can optimize both the harmonic                          %
% content and the number of switch transitions.  The                           %
% sequences are assumed to have quarter-wave symmetry.                         %
% The variable 'full_x' contains the final switching                           %
% sequence.                                                                    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Quarter-Wave Sequence Parameters
%---------------------------------
N=256;              % Sequence quarter-period length
Amp=1.0;            % Amplitude of the fundamental

% Transition Loss Function Parameters
%------------------------------------
TT=108;             % Target number of switch transitions
WT=400;             % Linear transition weighting factor

% Harmonic Loss Function Parameters
%----------------------------------
% A second-order filter is used here. The user may select
% the vector W as desired for different results.
R=100;
C=2e-6;
L=8.8e-3;
w=120*pi*[0:(2*N-1)];          % W(k) is |H(j120pik)|
W=freqs([R],[R*L*C L R],w);    % for the 2nd-order filter

% Simulated Annealing Parameters
%-------------------------------
Tinitial=1;           % Initial temperature
Tcool=0.99;           % Cooling factor
Theat=1.2;            % Heating factor
Melted=0;             % Startup flag
Melt_point=5;         % Delta_V/V percentage for "melted"
Max_T=400;            % MAX number of temperature cycles

% Generate an initial guess for the quarter-wave of x
E=round(Amp*2*N/pi);   % Amplitude in # of 1's
x_new=[ones(1,E) zeros(1,N-E)];
% Initialize annealing variables
index=1:N;
V_best=1e6;
V_last=1e6;
T_schedule=[1.0 1.0 0.5 0.33];

% Main annealing LOOP
for h=1:length(T_schedule)       % Reheat loop
  T=Tinitial*T_schedule(h);
  for j=1:Max_T                    % Temp itteration loop
    num_accepts=0;
    for i=1:N                        % Bit swapping loop
      % Reconfigure the bits in x_new
      x_new=x_now;
```

```
    z=floor(rand(1)*(N-E))+1;
    o=floor(rand(1)*E)+1;
    i_ones=index(x_new);
    i_zeros=index(1-x_new);
    x_new(i_ones(o))=0;
    x_new(i_zeros(z))=1;

    % Assemble a full period of the sequence
    full_x=[x_new fliplr(x_new) -x_new -fliplr(x_new)];

    % Find loss factor due to # of transitions
    Tx=sum(abs(diff(full_x)));
    Terror=(Tx-TT)/TT;
    if Terror < 0
       VT=0;
    else
       VT=WT*Terror;
    end
    % Find loss due to filtered harmonics
    X=fft(full_x)/(2*N);          % DFS magnitude coeffs
    VD=sum((abs(X(3:(2*N))).*abs(W(3:(2*N)))).^2);
    VD=100*sqrt(VD/(abs(X(2))^2));
    V = VT + VD;      % Add for the total loss function

    % Compare V to V_last and look for improvement
    Delta_V = V - V_last;
    if (Delta_V < 0)                % Accept -dV always
       V_last=V;
       x_now=x_new;
       num_accepts=num_accepts+1;
    end
    if (rand(1) < exp(-Delta_V/T)) % Accept +dV w/prob.
       V_last=V;
       x_now=x_new;
       num_accepts=num_accepts+1;
       if (~Melted & ((Delta_V/V) > Melt_point/100))
          Melted=1;                 % Check if melted
          Tinitial=T;
       end
    end
    if V < V_best          % Keep a copy of the best
       x_best=x_new;
       V_best=V;
       VD_best=VD;
       VT_best=VT;
       Tx_best=Tx;
    end
  end                      % End the bit swapping loop

  if(Melted==1) Melted=2;break;end  % Break if melted
  if (num_accepts > 0)     % Cool or heat
     if(h==1)
        T=T*Theat;          % Heat if not melted yet
     else
        T=T*Tcool;          % Cool if already melted
     end;
  else
     break;                 % Break out if #accepts = 0
  end

  % Display progress
  if(h==1)
     str=sprintf('Melting=%1.0f Tcyc=%3.0f',h,j);
  else
     str=sprintf('Coolcyc=%1.0f Tcyc=%3.0f',h,j);
  end
  str=sprintf('%s Acpts=%4.1f%%',str,num_accepts/N*100);
  str=sprintf('%s T=%5.2f%',str,T);
  str=sprintf('%s *** BEST-> V=%5.1f%%',str,V_best);
  str=sprintf('%s (VD=%5.1f%%',str,VD_best);
  str=sprintf('%s VT=%4.1f)',str,VT_best);
  str=sprintf('%s Tx=%3.0f',str,Tx_best);
  disp(str);
 end                         % End Temperature loop
end                          % End the Reheat loop
```

```
% Graph the result
disp('-----------');
full_x=[x_best fliplr(x_best) -x_best -fliplr(x_best)];
X=abs(fft(full_x))/(2*N);

subplot(211)
stairs([0:(4*N-1)],full_x);
axis([0 4*N -1.1 1.1]);
xlabel('Index (i)');
ylabel('Amplitude');
title('Final Switching Sequence');
subplot(212)
bar([0:(4*N-1)],abs(X)*100);
hold on
plot([0:(2*N-1)],abs(W)/max(abs(W))*100,'r');
hold off
axis([0 100 0 100]);
xlabel('Harmonic Number');
ylabel('% of Fundamental');
title('Final Harmonic Magnitudes');
```