The Sinefit Spectral Envelope Preprocessor

James Paris, John S. Donnal, Zachary Remscrim, Steven B. Leeb, *Fellow, IEEE*, and Steven R. Shaw

Abstract—This paper presents a new spectral envelope preprocessor based on sinusoid fitting and the discrete Fourier transform. This preprocessor is well-suited for nonintrusive condition monitoring and diagnostics due to its high noise resiliency and flexibility. It reduces data storage, transfer, and processing requirements by extracting only relevant harmonic signatures. This paper analyzes the resolution and accuracy benefits of spectral envelopes, including the effects of additive white Gaussian noise and presence of higher frequency spectral harmonics.

Index Terms—Energy monitoring, power monitoring, power computation, nonintrusive load monitoring, digital metering.

I. Introduction

FOR many electrical systems driven by ac sources, the "spectral envelope" representation of observed current and voltage signals has proven to be a widely useful and powerful metric for classification, diagnostics, and power quality measurement [1]–[5]. Spectral envelopes describe the harmonic content of the measured signals at integer multiples of the ac line frequency driving the monitored loads. Such loads exhibit behaviors that are synchronous with the line frequency. By extracting spectral envelopes, the preprocessor facilitates physically based analysis of power and current consumption. Line harmonics flow though utility impedances creating voltage distortions that form a class of power quality problems. Tracking these line harmonics can permit estimation of likely voltage disturbances [6].

The spectral envelope preprocessor has two primary tasks: phase and frequency estimation, and harmonic coefficient calculation. Existing versions of the spectral envelope preprocessor vary in their implementations. For phase and frequency estimation, common techniques include phase-locked loops [4], weighted least-squares estimators, and Kalman filters [2]. To reduce computation load, existing implementations have utilized techniques such as analog multipliers and precomputed basis vectors [2], [4].
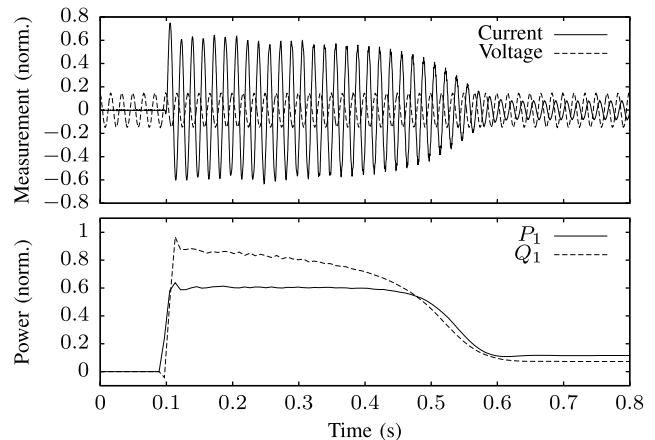
Fig. 1. Raw ac voltage and current measurements (top) and computed spectral envelopes (bottom).

This paper presents the Sinefit spectral envelope preprocessor, based on non-linear least-squares sinusoid fitting combined with the DFT. These techniques focus on accuracy and implementation flexibility, reflecting the growing availability of high performance computing resources.

In an ideal environment the spectral harmonics could be extracted directly without error by any preprocessing algorithm. In environments with sensor noise and quantization error, simple single-point algorithms, such as peak and zero-crossing detection, are subject to erroneous output. Sinefit improves on these algorithms by evaluating many input data points over multiple periods of the input waveform. Furthermore, Sinefit is a stateless algorithm that does not rely on prior data, and is therefore robust to rapid changes in line frequency and voltage. By comparison, similar preprocessing tools such as the Kalman filter may take hundreds of line cycles to adjust to step changes in the input [2].

II. Spectral Envelopes

Spectral envelopes $a_k(t)$ and $b_k(t)$ are short-term averages of harmonic content, calculated over sliding windows of an input signal. Fig. 1 demonstrates spectral envelopes as computed for an ac load. The first plot is the raw sampled input from a data acquisition board. The second shows the in-phase and quadrature components of the first harmonic (60 Hz) envelopes.

A. Definition

For the Non-Intrusive Load Monitor (NILM), we assume that the voltage and current signals $v(t)$ and $i(t)$ are *locally* periodic over one ac line cycle. For a discrete input $i[n]$

sampled at rate f_s, one period is of length $N = f_s/f_0$ samples, and we can compute harmonic coefficients as:

$$a_k = \frac{2}{N} \sum_{n=0}^{N-1} i[n] \cdot \sin(k(2\pi n/N)) \tag{1}$$

$$b_k = \frac{2}{N} \sum_{n=0}^{N-1} i[n] \cdot \cos(k(2\pi n/N)) \tag{2}$$

Here, k denotes the multiple of the line frequency to which a particular coefficient corresponds; for example, $k = 1$ corresponds to the 60 Hz component and $k = 3$ to the 180 Hz component.

Harmonic coefficients can also be calculated in complex form using the DFT, which is defined as:

$$X_k = \mathcal{F}(x[n]) = \sum_{n=0}^{N-1} x[n] \cdot e^{-jk2\pi \eta/N} \tag{3}$$

Using Euler's formula, we can extract a_k and b_k in terms of the DFT as:

$$e^{-jw} = \cos(\omega t) - j\sin(\omega t) \tag{4}$$

$$a_k = -\frac{2}{N} \operatorname{imag}(X_k) \tag{5}$$

$$b_k = \frac{2}{N} \operatorname{real}(X_k) \tag{6}$$

where, as before, $N = f_s/f_0$.

The values of these coefficients are calculated for successive or sliding windows of the input signal, and the resulting time-varying harmonics $a_k[m]$ and $b_k[m]$ are the spectral envelopes.

Spectral envelopes can be extracted from any periodic or quasiperiodic input signal, and are typically computed for the current, using the voltage as the phase and frequency reference. When computed separately for voltage $v[n]$ and current $i[n]$, we denote the coefficients as a_{vk}, b_{vk}, a_{ik}, and b_{ik}. Then, first harmonic real and reactive power are:

$$P_1 = a_{v1} \cdot a_{i1} \qquad Q_1 = b_{v1} \cdot b_{i1} \tag{7}$$

For relatively "stiff" and harmonic-free utility voltage, a_{v1} and b_{v1} can be assumed to be constant, in which case:

$$P_1 \propto a_{i1} \qquad Q_1 \propto -b_{i1} \tag{8}$$

B. Phase Rotation

The coefficients a_k and b_k in (5) and (6) are calculated with sliding or successive windows over the input data. These windows must be phase-aligned with a reference, typically the utility voltage, such that $n = 0$ corresponds to, for example, the zero crossing. Then, a_k and b_k will refer to the "in-phase" and "quadrature" spectral components, respectively.

In the more general case, the spectral coefficients can be computed over any window $[n, n + N]$, where the reference phase corresponding to sample n is $\phi[n] = \phi_0 \neq 0$. Then, a correcting rotation of $-k\phi_0$ can be applied to the complex DFT coefficient X_k:

$$X'_k = X_k \cdot e^{\phi_0 \cdot jk} \tag{9}$$

There are four common cases that require this phase rotation:

1) When calculating harmonic coefficients with a sliding window that is shifted by a non-integer number of periods, the start of each successive window will have a different phase ϕ_0, which must be accounted for by a rotation of X_k on a per-window basis.
2) For three-phase ac systems, it is common to use a single voltage $V_{\varphi A}$ as a phase reference. When computing spectral envelopes corresponding to $I_{\varphi A}$, $I_{\varphi B}$, and $I_{\varphi C}$ using this reference, phase rotations of 0°, 120°, and 240° should be applied, respectively.
3) Current transformers or transducers may introduce a fixed phase offset in their measurement, typically 0-1°. Correcting this offset with phase rotation of the preprocessor output can significantly reduce error at higher harmonics.
4) Multi-channel data acquisition cards that sample sequentially rather than simultaneously will introduce a similar phase offset between samples. For example, evenly-spaced, 6 channel, 8 kHz sampling of a 60 Hz signal inserts a phase rotation of 0.45° between each channel.

In subsequent discussion, the first case is referred to as ϕ_{shift}, and the others are collectively referred to as ϕ_{extra}.

III. IMPLEMENTATION

A variety of approaches have been used to extract spectral envelope information for non-intrusive monitoring, including implementations of the Short-Time Fourier Transform [3], the S-transform, [7], wavelet transforms [8], hardware PLL's for tracking utility frequency, and recursive esitmators like the Kalman filter [9], which essentially implement a software PLL for tracking utility frequency. Sinefit provides an especially useful approach for trading off the two key concerns in computing spectral envelopes: noise sensitivity versus localization in time. Responsive methods that quickly adapt to frequency changes may be highly sensitive to noise. More noise-robust methods use larger windows of information to compute frequency for spectral content estimation, but respond sluggishly to changes. Specific results necessarily depend on the monitoring environment and certain applications may have characteristics more suitable to a particular pre-processors. That is, a blanket characterization of preprocessing approaches is usually not possible. However, Sinefit has proven an especially flexible approach for optimizing the tradeoff between noise sensitivity and dynamic response.

Environments with noisy voltage waveforms, such as the one shown in Fig 2, make harmonic envelope extraction especially challenging. Figure 3 compares Sinefit to a Kalman filter preprocessor for a load on this utility. Both preprocessors capture the steady increase in power, but the Kalman filter introduces a time delay and rapid oscillations while Sinefit accurately tracks the true harmonic envelope.

Sinefit has a modular, transparent design that allows for easy replacement and reuse of components, such as the phase alignment or harmonic coefficient calculation. The preprocessor uses a phase and frequency estimation algorithm
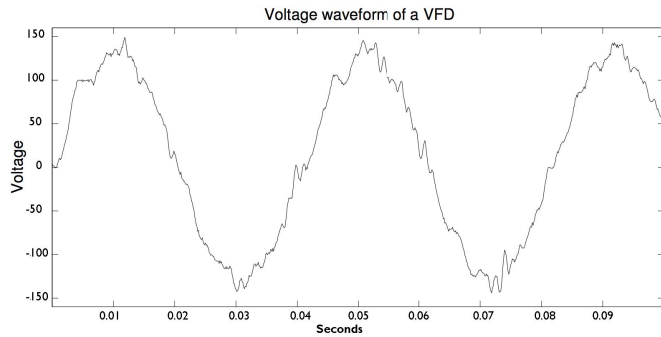
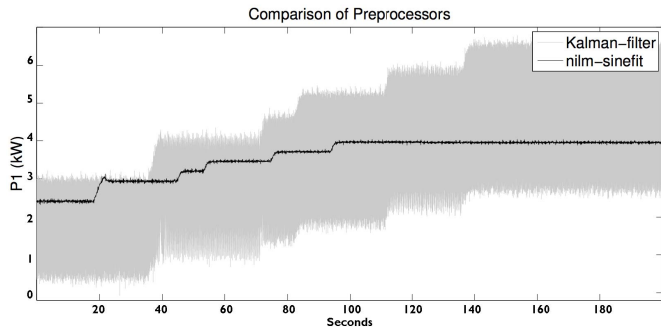Fig. 2. Noisy voltage waveforms complicate the extraction of harmonic envelopes.

Fig. 3. Harmonic envelope for a noisy voltage system as calculated by a Kalman-filter preprocessor and Sinefit.

that is robust when presented with highly variable or truncated voltage waveforms. Finally, it focuses on correctness and implementation simplicity by taking advantage of the significant advancements in computing power since the existing preprocessors were developed. Fig. 3 compares the new preprocessor to the previous Kalman-filter preprocessor for a variable frequency drive system. The new preprocessor computes accurate harmonic envelopes despite the distorted voltage and it does not introduce time delays like the Kalman-filter approach.

Here, the spectral envelope preprocessor is split into two independent components. The first, an algorithm and code module called "nilm-sinefit", performs least-squares fits of sinusoids to successive windows of the input waveform in order to mark zero crossings, frequency, and amplitude. The second, "nilm-prep", performs spectral envelope extraction using (5), (6) and (9) over sliding windows of the input and sine fit data. The signal flow in the new spectral envelope preprocessor is shown in Fig. 4, and the implementations of these components are detailed below.

A. 4-Parameter Sinusoid Fitting

In order to accurately estimate the unknown frequency and phase angle from a voltage waveform, the preprocessor finds the best fit of a sinusoid to the data, using the method identified in IEEE Std 1241-2010 Annex B.2 [10]. Fig. 5 demonstrates this optimal fit for two representative waveforms sampled by an Analog to Digital Converter (ADC).

Mathematically, given an arbitrary waveform vector v of length N sampled at frequency f_s, we wish to calculate the
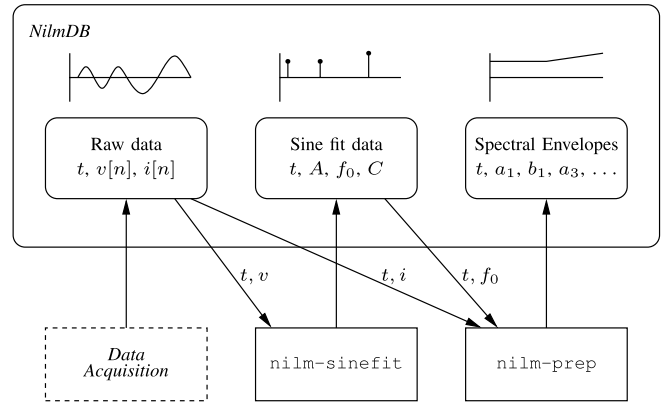

Fig. 4. Block diagram of signal flow in the Sinefit spectral envelope preprocessor.
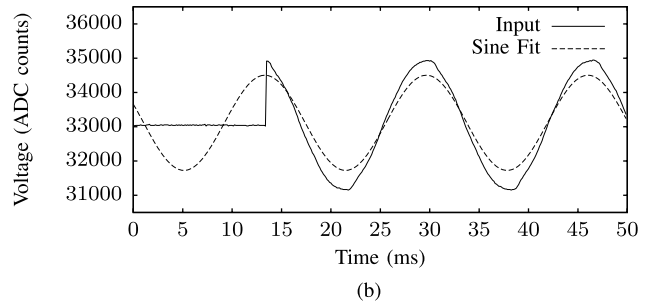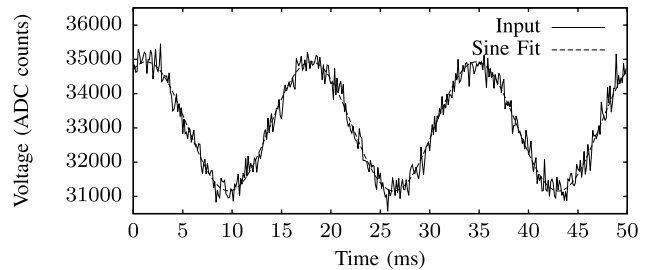

(a)

(b)

Fig. 5. Results of 4-parameter sinusoid fitting. (a) Sine fit on a noisy waveform. (b) Sine fit on a voltage that is zero for the first 1/4 of the window. Frequency and phase are still identified with reasonable accuracy.

four parameters A', B', C, and f_0 that best satisfy:

$$v' = A' \cos\left(2\pi n \frac{f_0}{f_s}\right) + B' \sin\left(2\pi n \frac{f_0}{f_s}\right) + C \qquad (10)$$

such that the least-squares residual $\sum (v[n] - v'[n])^2$ is minimized. Note that this system is non-linear.

The algorithm that estimates these parameters is as follows. First, an initial estimate of f_0 is found by calculating the DFT of v with a window size N calculated in (26), locating the DFT index L with the largest magnitude, and computing $f_{0,\text{est}} = L \cdot f_s / N$. This frequency is accurate to within the resolution of the DFT. Other estimation techniques can be used; for example, the implementation in Section III-B uses the more accurate interpolated DFT described in [11], which may improve convergence of the sinusoid fit for small N.

Now, perform the iterative fit of (10). Define the starting conditions as:

$$A_0 = B_0 = C_0 = 0 \qquad (11)$$

$$\omega_0 = 2\pi \cdot f_{0,\text{est}} \qquad (12)$$

For each iteration $i = \{0, 1, 2, \ldots, m\}$, perform a least-squares fit on a linearization of the system. Given an assumed small deviation from ω_i of $\Delta\omega_i$, the Taylor series expansion of Section III-A around the current estimated parameters gives the linear equation:

$$v'_i = D_i \cdot \begin{bmatrix} A_i & B_i & C_i & \Delta\omega_i \end{bmatrix}^T \quad (13)$$

where D_i is defined as:

$$D_i \equiv \begin{bmatrix} \cos(\omega_i t) & \sin(\omega_i t) & 1 & d_i \end{bmatrix} \quad (14)$$

and terms d_i and t are defined as:

$$d_i \equiv B_i t \circ \cos(\omega_i t) - A_i t \circ \sin(\omega_i t) \quad (15)$$

$$t \equiv \begin{bmatrix} 0, 1, \ldots, (N-1) \end{bmatrix} / f_s \quad (16)$$

The least-squares solution to this system gives the updated estimates for the next iteration as:

$$\begin{bmatrix} A_{i+1} \\ B_{i+1} \\ C_{i+1} \\ \Delta\omega_{i+1} \end{bmatrix} = \left(D_i^T D_i \right)^{-1} \left(D_i^T v \right) \quad (17)$$

$$\omega_{i+1} = \omega_i + \Delta\omega_{i+1} \quad (18)$$

Note that the least-squares fit in Section III-A can be computed with a more numerically-stable method such as Q-R decomposition [10]. The solution converges rapidly, and the preprocessor stops after a fixed number $m = 7$ iterations. The fitted parameters for Section III-A are:

$$A' = A_m \quad B' = B_m \quad (19)$$

$$C = C_m \quad f_0 = \frac{\omega_m}{2\pi} \quad (20)$$

Finally, we convert this fit into the equivalent polar form:

$$v[n] = A \cdot \sin\left(2\pi n \frac{f_0}{f_s} + \phi_0\right) + C \quad (21)$$

by computing:

$$A = \sqrt{A_m^2 + B_m^2} \quad (22)$$

$$f_0 = \frac{\omega_m}{2\pi} \quad (23)$$

$$\phi_0 = \text{atan2}(A_m, B_m) \quad (24)$$

$$C = C_m \quad (25)$$

This form is preferable, because the parameters $[A, f_0, \phi_0, C]$ are more directly applicable to computing spectral envelopes.

B. Implementation of `nilm-sinefit`

The `nilm-sinefit` tool, implemented in Python, uses successive 4-parameter sine wave fits to find and mark every positive zero crossing ($\phi = 0$) of an input voltage waveform. Each mark includes the amplitude A, frequency f_0, and offset C of the following period. Given the expected line frequency $f_{\exp}$ and sampling rate f_s, the fits are calculated over sliding windows of

$$N = 3.5 \cdot f_s / f_{\exp} \quad (26)$$

samples of voltage. This corresponds to approximately 3.5 periods, although the actual number will vary with f_0.
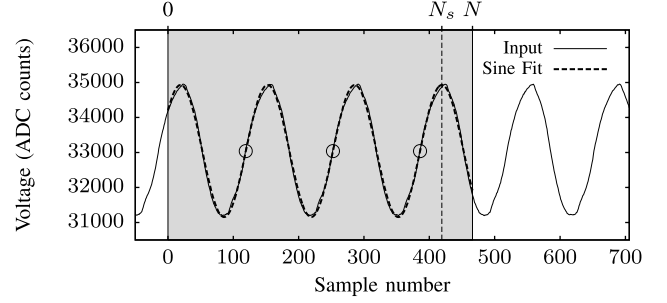

Fig. 6. Windowing algorithm of `nilm-sinefit`. Sine fit is performed over the window $[0, N]$, and positive zero crossings are marked as shown by the circles. The next window starts at N_s, the last point with phase angle $\frac{\pi}{2}$.

TABLE I
PARAMETERS FOR `nilm-prep`

Parameter	Default	Description
N_{harm}	4	Number of odd harmonics to store.
N_{shift}	1	Number of shifted windows for which to compute coefficients, per zero crossing.
ϕ_{extra}	0	Extra phase rotation to apply, to correct for known phase offsets.

If f_0 falls outside predetermined bounds, or the amplitude A is too low, the window is skipped to avoid spurious marks.

Fig. 6 demonstrates the fit and marking over a window of length N. To avoid potentially double-marking a zero crossing that occurs near window boundaries, the point N_s is calculated as the last point within $[0, N]$ with phase angle $\frac{\pi}{2}$. Only zero crossings prior to N_s are marked, and the next window is shifted to $[N_s, N_s + N]$. This ensures that any particular zero crossing will only fall squarely within one window, even as sine fit parameters change.

The `nilm-sinefit` processing tool takes as input a single timestamped data stream, and marks zero crossings using data from a user-specified column in the input stream. Output marks are written to a new stream consisting of a timestamp t and three floating-point values A, f_0, and C. These values correspond to the parameters from Section III-A, as calculated for the crossing detected at time t.

C. Implementation of `nilm-prep`

The `nilm-prep` tool, implemented in Python, reads two data streams: an input waveform $i[n]$, and the zero crossing and f_0 estimates for each period as marked by `nilm-sinefit`. Other parameters that control its behavior are shown in Table I. For each identified zero crossing at time t with frequency f_0, the spectral envelopes are calculated with the following algorithm:

1) Initialize $N = f_s / f_0$ and $\phi_{\text{shift}} = 0$.
2) Extract N samples starting at time $t + (\phi_{\text{shift}}/(2\pi f_0))$.
3) Use the Fast Fourier Transform to calculate X_k of $i[n]$, as defined in (3).
4) Apply rotation from Section II-B using $\phi_0 = \phi_{\text{extra}} - \phi_{\text{shift}}$ to obtain X'_k.
5) Calculate and store the first N_{harm} odd harmonic coefficients from X'_k using (5), (6).
6) Increment ϕ_{shift} by $2\pi / N_{\text{shift}}$.
7) Repeat from step 2 until $\phi_{\text{shift}} \geq 2\pi$.

This results in N_{shift} sets of coefficients per period of the input waveform, by applying successive overlapping DFTs. This sliding-window approach may be useful in some cases, as it can help retain additional information about energy content at harmonics that are not otherwise stored by this implementation, such as the even harmonics. For waveforms known to consist primarily of the low-numbered odd harmonics, $N_{\text{shift}} = 1$ is sufficient and results in the most space savings.

Output from `nilm-prep` is stored in a new stream where each row contains a timestamp and $2 * N_{\text{harm}}$ floating-point values numbers, in order $\{a_1, b_1, a_3, b_3, \ldots\}$.

IV. COMPRESSION BENEFITS

Physical modeling of ac loads shows that there are often useful bounds on the number and types of harmonics that are present in the load current. By omitting these higher harmonics from stored data streams, the spectral envelope preprocessor performs a domain-specific form of compression that greatly reduces data storage and transfer requirements. A typical NILM data acquisition is 16-bit samples at a rate of 8 kHz. For a three-phase system, 6 channels are recorded. Assuming one 64-bit timestamp is also stored with each sample, the storage requirements for raw data are:

$$8 \text{ kHz} \cdot (64 + 16 \cdot 6) \text{ bits} = 13.8 \text{ GB/day}. \tag{27}$$

Preprocessor output is harmonics a_k and b_k for $k = 1, 3, 5, 7$ as 32-bit floating point values with a 64-bit timestamp, for each of the three phases. One set of coefficients is calculated per 60 Hz period, so the total storage requirements for preprocessed data are:

$$60 \text{ Hz} \cdot (64 + 32 \cdot 8) \text{ bits} \cdot 3 = 0.62 \text{ GB/day}. \tag{28}$$

Thus, the preprocessor reduces the data to only 4.5% of its original size, while preserving all information about three-phase power usage up to the 7th harmonic. Even doubling this and storing up to the 15th harmonic still uses only 8.1% of the original raw data space.

Other reductions of the data could instead be applied, for example, simply recording aggregate real power averaged over every period or second. However, such data would not reflect the detailed short term variations that would occur in real power, nor would it reflect any of the behavior of the higher harmonics. Time-varying spectral envelope coefficients strike a balance between the usefulness of the retained data and the storage and transmission requirements of that data.

V. RESOLUTION AND ACCURACY

The original digital samples of the current and voltage waveforms have limited precision due to quantization. As shown in Fig. 7, the continuous input signal is divided into B discrete regions, and each region is mapped to a unique digital code during the sampling process. This quantization can be stated explicitly as

$$i[n] = \left\lfloor 2^B \cdot i(t) + 0.5 \tfrac{1}{2} \right\rfloor \tag{29}$$

where $i(t)$ is normalized to the range [0], [1]. The preprocessor input consists of N values of $i[n]$ over one period of the line voltage.
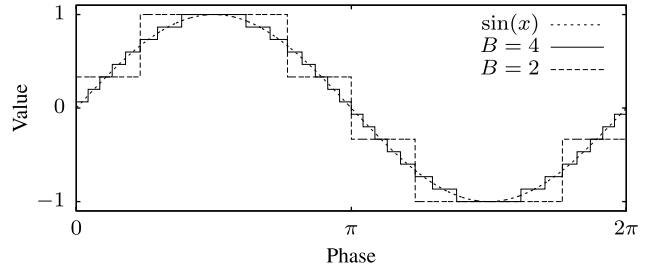

Fig. 7. Quantization error, as would be introduced by a linear ADC. The continuous function $\sin(x)$ is compared to the same function represented discretely with only 4 and 2 bits of resolution.

For power estimation and load identification purposes, it is desirable to find the average power of the input signal, but the quantization error affects this measurement. Using raw data to directly estimate power is less accurate than using preprocessed spectral envelopes to do the same. To compare the accuracy, we begin by considering estimation of power when the current waveform is a simple sinusoid of the form

$$i(t) = A \sin(\omega t) \tag{30}$$

Here, total power is proportional to the amplitude A of the sinusoid. More complex input waveforms are addressed in Section V-D.

A. Resolution of Power From Preprocessor Output

The preprocessor improves effective power resolution by averaging over time. Specifically, nilm-prep uses all N sampled data points over one or more periods. For example, the preprocessor discards all but the low, odd-numbered harmonics; the sinusoid in Section V contains no harmonics other than for $k = 1$. Thus, the nilm-prep output encapsulates all information about the original signal. Since the analogous DFT is invertible, this means that every unique sampled input has a unique corresponding preprocessor output.

We can use this to determine how many discernible outputs the preprocessor will produce as the amplitude A of the input sinusoid changes. The sampling process that generates $i[n]$ is limited in resolution, and so there are necessarily a finite number U of unique sampled $i[n]$ waveforms. Because the DFT operation maps each unique input to a corresponding unique output, there are the same number U of unique preprocessor outputs. Thus, the preprocessor output can be used to discern U different values of A.

To calculate U, consider sweeping A from $0 \rightarrow 1$, as demonstrated in Fig. 8. By symmetry, let us consider only the first $N/4$ samples. At $A = 0$, all samples are zero. As A increases, there will be some transition where a single quantized sample $i[n]$ will increase by one. More specifically, as A increases from 0 to 1, samples will monotonically increase according to the following pattern:

- The first sample at $n = 0$ never increases.
- The peak sample at $n = N/4$ increases $2^{(B-1)} - 1$ times.
- For each sample between these extremes, the number of "increases", or effective quantization steps, is given by:

$$U_n = \left\lfloor (2^{(B-1)} - 1) \sin\left(\tfrac{2\pi n}{N}\right) + \tfrac{1}{2} \right\rfloor \tag{31}$$
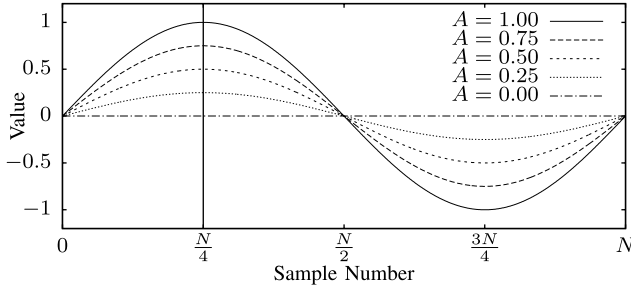
Fig. 8. Sweeping through all input amplitudes of a single fundamental frequency from 0 to 1, in order to enumerate the number of possible quantized inputs. By symmetry, only the first $N/4$ samples need to be considered.
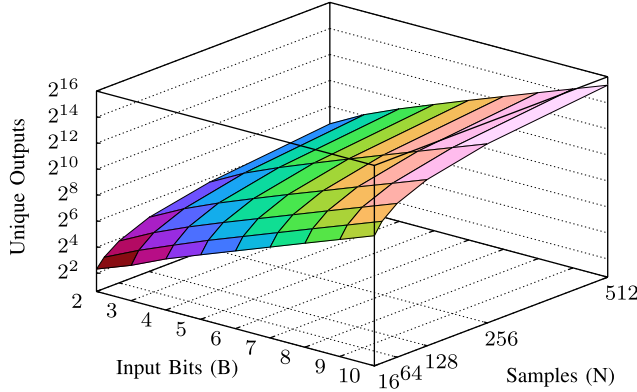

Fig. 9. Number of unique preprocessor outputs, as a function of the number of samples N and the number of input bits B, for an input signal consisting of a single sinusoid of varying amplitude.

Fig. 10. Unique preprocessor outputs as a function of input amplitude of a sinusoid. For each amplitude level A_x, the graph plots the total number of unique outputs seen while the input sinusoid amplitude is swept from 0 to A_x.

Fig. 11. Effective resolution, in bits, as a function of input amplitude of a sinusoid. An effective resolution of β bits means that a change of $1/(2^\beta)$ in input amplitude can be resolved.

where the symbol $\lfloor x \rfloor$ denotes the largest integer not greater than x.

Each increase creates a new input to the preprocessor, which results in a new output. Thus, the total number of unique outputs from the preprocessor is one corresponding to the initial case ($i[n] = 0$), plus one for each time a sample in $i[n]$ increases.

$$U = 1 + 0 + \left(\sum_{n=1}^{\frac{N}{4}-1} U_n \right) + \left(2^{(B-1)} - 1 \right) \tag{32}$$

$$= 2^{(B-1)} + \sum_{n=1}^{\frac{N}{4}-1} \left\lfloor (2^{(B-1)} - 1) \sin \left(\frac{2\pi n}{N} \right) + \frac{1}{2} \right\rfloor \tag{33}$$

Fig. 9 shows this number of unique outputs U as a function of the input samples N and input quantizer bits B. The number of output bits are approximately linear with the number of input bits, and increase logarithmically with N. As a representative data point, sampling at values $N = 128$ and $B = 10$ gives an overall output resolution of about $B = 13$ bits.

Note, however, that the unique outputs of the preprocessor are not linearly distributed among the input amplitudes. For example, for amplitudes less than the first quantized bit level of $1/(2^B - 1)$, all quantized input samples are zero, and so there is only one unique preprocessor output. Fig. 10 shows the number of unique outputs as a function of the input amplitude, as enumerated computationally via binary search (described later in Algorithm 1). As the input amplitude is swept along
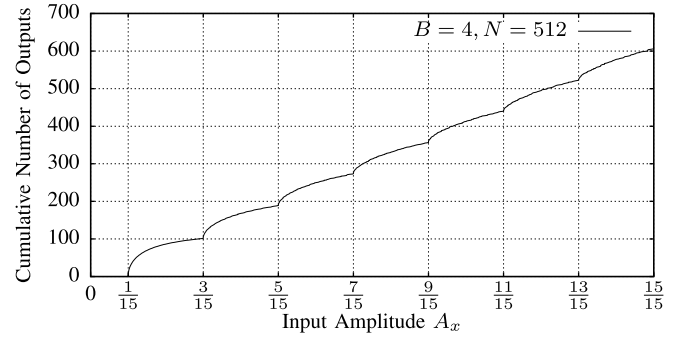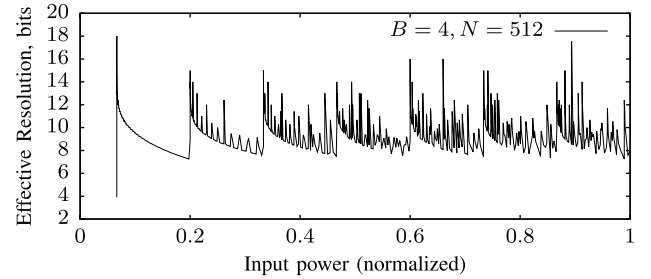
the x-axis from left to right, the cumulative number of unique outputs are counted and plotted along the y-axis. Inflection points in the curve correspond to the quantization levels. This is because small changes in amplitude affect more samples when the peak of the sinusoid is near a quantization level, causing relatively more unique outputs in these areas.

From Fig. 10, a measure of effective resolution as a function of input amplitude can be developed. If a change of amplitude from A to $(A + \Delta A)$ causes the preprocessor to output a new unique value, then discerning these values corresponds to discriminating between input amplitudes with a resolution of $\beta = \log_2 (1/\Delta A)$ bits around operating point A. Fig. 11 shows this resolution as a function of input amplitude. Here, the input bits $B = 4$ and samples $N = 512$, and the output resolution varies between $\beta = 4$ and 18 bits, with an average around 10 bits.

B. Accuracy of Preprocessor Output

As shown, the preprocessor can provide a high resolution estimate of input waveform amplitude. For load identification this may be sufficient to discern between different inputs. In some cases, such as in power level measurement and energy scorekeeping, high accuracy is also desired.

When the characteristics of the input waveform are known, the accuracy can often be directly determined. To continue the previous example, consider a single sinusoidal waveform at the fundamental line frequency and amplitude A, corresponding to power P. This waveform is sampled, quantized, and passed through the spectral envelope preprocessor. Fig. 12 shows a
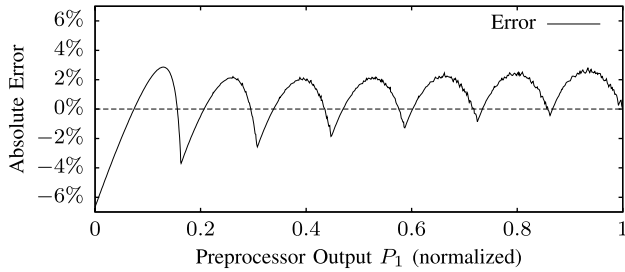
Fig. 12. Accuracy of preprocessor output when provided with quantized input data from a single sinusoid. The absolute error, as a percentage of the full power range, is plotted as a function of the P_1 output. Here, $B = 4$, $N = 512$, and the maximum error of any single sample due to quantization alone would be $1/(2^B) = 6.25\%$.
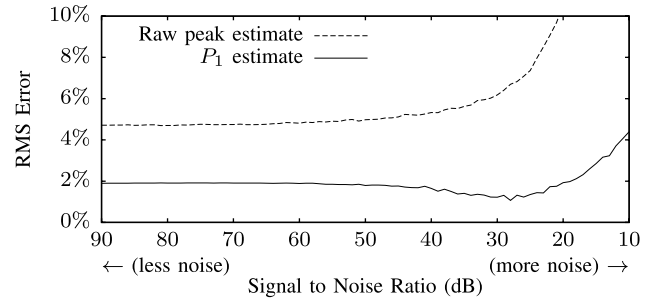

Fig. 13. Preprocessor error in the presence of additive white Gaussian noise. At some levels, white noise can improve preprocessor accuracy by introducing dither into the quantization process. The sampling parameters are $B = 4$, $N = 512$.

plot of the error between the preprocessor output P_1 and the actual power P, as a function of P. Like the resolution, the error varies with signal amplitude. Note that the maximum absolute error corresponds to approximately one part in 2^B, where B is the quantization bits. Thus, the plot shows that the error of the preprocessor output is always less than or equal to the quantization error of any individual sample.

C. Accuracy in the Presence of Noise

The previous analysis has been calculated and simulated based on ideal waveforms. In practice, there are many potential sources of noise or interference in non-intrusive load monitoring. This noise can be broadly split into two categories, correlated and uncorrelated signals.

Correlated interference refers to any introduced signal or distortion that is related to the loads being monitored or is otherwise nonrandom. Examples include magnetic coupling between adjacent current transducers, pickup of stray 60 Hz electric fields from other shielded wiring or lighting fixtures, and aliasing effects in the data acquisition process. This sort of coupling can potentially be complex and relate closely to specific installation details, which means that its effect on preprocessor accuracy is highly variable. In general, the net effect of correlated noise has been extremely low in observed systems [5].

Uncorrelated noise is statistically independent from the input waveforms. One such type of noise is additive white Gaussian noise (AWGN), which is normally-distributed around zero and might be expected to appear as a result of thermal noise in the sensors or data acquisition. This form of noise often sets the limit of sampling resolution in the monitoring system.

The preprocessor is particularly well-suited to handle such disturbances. To analyze the effect of AWGN on preprocessor accuracy, we again consider the example of a single sinusoid, with added noise $\mathcal{N}(t)$:

$$i(t) = A\sin(\omega t) + \mathcal{N}(t) \qquad (34)$$

By sweeping A from $0 \rightarrow 1$, we can generate data in the same manner as Fig. 12, simulating the sampling process and calculating the preprocessor power estimate $P_1(A)$ corresponding to each actual power input A. Define the overall root-mean-

square error of this data as:

$$E_{\text{RMS}} = \sqrt{\int_0^1 (P_1(A) - A)^2 \, dA} \qquad (35)$$

This RMS error will vary with the amount of noise $\mathcal{N}(t)$ that is being added to the sinusoid. We describe this amount using the signal-to-noise ratio (SNR), the ratio of power in the full-amplitude sinusoid to the power of the added white Gaussian noise.

Fig. 13 shows the calculated overall RMS error versus the SNR of injected noise. For each SNR value, the amount of noise is held constant as A is swept to produce one point of each line in Fig. 13. The minimum error is zero and the maximum error is bounded by the random process describing the noise channel. Two approaches to estimating power are shown: the preprocessor P_1 estimate, and the simple peak-estimation technique. As expected, the preprocessor estimate offers an improvement over the raw peak estimate in all cases, reducing overall error by more than half.

In some cases, the addition of noise will actually *reduce* error. Here, noise levels around 30 dB provide a slight accuracy improvement. This is due the dithering effect of the white Gaussian noise on the sampling quantization which, combined with the averaging effect of the spectral envelope calculation, serves to decouple the quantization error from the input. The preprocessor can therefore be seen as even more useful in the presence of noise.

D. Resolution and Accuracy With Complex Waveforms

Real-world systems often draw energy content at more harmonics than just the fundamental line frequency. The analysis of preprocessor resolution and accuracy can be extended past single sinusoids to take these harmonics into account. For such waveforms, closed-form solutions to find the number of discernable power levels like Section V-A may not necessarily exist.

Instead, we develop a model $f(A)$ that describes the current waveform shape as a function of power level $A = 0 \rightarrow 1$. This approach is suitable for loads that exhibit correlations between fundamental and higher harmonic components, such as a_3 and a_5. For example, one such model is described in [13] for variable-speed drive (VSD) systems, where harmonic contents are related to the total apparent power A by functions

Algorithm 1 Binary Search to Enumerate Unique Preprocessor Outputs as Power A Varies, Given Waveform Model $f(A)$

```
outputs ← [ ]
function ENUMERATE(f, A_low, A_high)
    P_low  ← PREPROCESS(SAMPLE(f(A_low)))
    P_high ← PREPROCESS(SAMPLE(f(A_high)))
    if P_low ≠ P_high then
        A_mid ← (A_high + A_low)/2
        ENUMERATE(f, A_low, A_mid)          ▷ Search 1st half
        ENUMERATE(f, A_mid, A_high)         ▷ Search 2nd half
    else if P_low not in outputs then
        outputs ← outputs + P_low           ▷ Add if unique
    ENUMERATE(f, 0, 1)
return SORT(outputs)
```

of the form:

$$\sqrt{a_k^2 + b_k^2} = bA^p \qquad (36)$$

In many systems, including VSDs, these relationships are nearly linear. Here, we consider a similar class of waveforms where the ratios between harmonics are fixed in proportions A_3 and A_5:

$$f(A, t) = A \cdot (\sin(\omega t) + A_3 \sin(3\omega t) + A_5 \sin(5\omega t)) \qquad (37)$$

We wish to determine the effective resolution of the preprocessor for systems following this model as we vary A. As with the single sinusoid in Section V-A, we do this by counting the number of unique preprocessor outputs as we sweep $A = 0 \rightarrow 1$. For every value of A, we use the model $f(A)$ to find the waveform for this power level, numerically simulate the sampling and quantization, and compute the preprocessor output. To efficiently enumerate all potential outputs, we apply the binary search algorithm described in Algorithm 1, and count the resulting number of unique outputs U to determine the effective number of bits as $\log_2(U)$.

Note that the binary search in Algorithm 1 assumes a monotonically increasing preprocessor output as A increases. For more complex models $f(A)$ where this does not hold true, the algorithm can be augmented or replaced with a slower linear search that steps through A by sufficiently small ΔA and counts the total number of unique outputs seen.

The enumeration process was performed for varying combinations of harmonic content ratios A_3 and A_5 in Section V-D. Since varying harmonic content can change the peak amplitude of the current waveform, the sampling process was scaled such that the bits-per-amp ratio is fixed for all trials, to maintain consistency. Fig. 14 shows a plot of the calculated effective bits of resolution. The base resolution is 9.8 bits for $A_3 = A_5 = 0$.

These results demonstrate that adding harmonic content often increases the effective resolution of the preprocessor. Some particular combinations of A_3 and A_5 also reduce it, to a smaller extent. Fig. 15 shows the waveforms corresponding to the minimum (9.6 bits) and maximum (10.6 bits) resolutions over this parameter range. Informally, the cases where resolution is reduced are those where the total signal amplitude is lowered, or where the waveform has "flat" regions that do not vary much as the parameter A is scaled. Cases where the
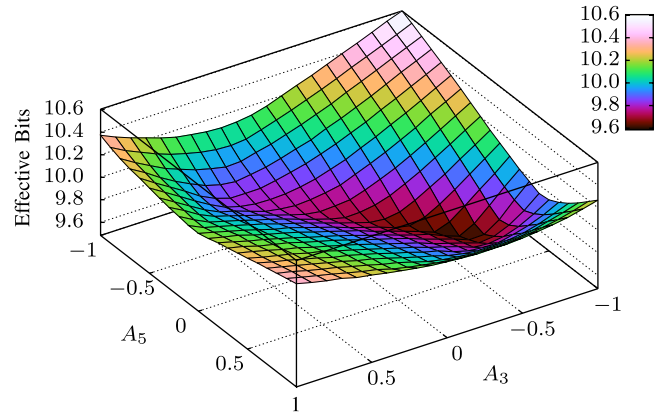

Fig. 14. Effective bits of preprocessor resolution as relative amounts of harmonic content A_3 and A_5 are varied in Section V-D. The harmonics are held at a fixed ratio while the overall waveform scaling is swept from $A = 0 \rightarrow 1$. Sampling parameters are $B = 8$, $N = 128$ and the base case $A_3 = A_5 = 0$ corresponds to 9.8 bits.
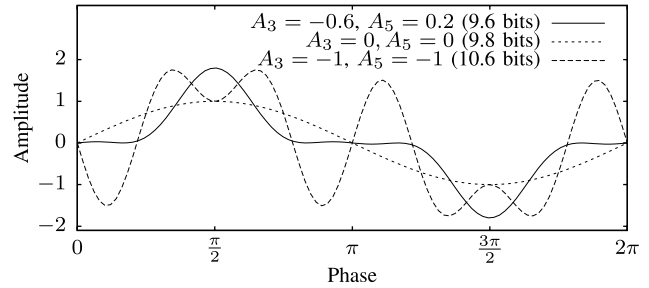

Fig. 15. Full-amplitude waveforms corresponding to the minimum and maximum resolution from Fig. 14. In general, increasing complexity increases the effective resolution, although some combinations of harmonics reduce it slightly due to "flat" regions that do not vary much with power scaling.

final waveform exhibits more complexity will generally see improvements in preprocessor output resolution.

VI. EXPERIMENTAL RESULTS

To assess the practicallity of non-intrusive load monitoring in a shipboard environment two NILM's were installed on the the USCGC ESCANABA (Boston, MA) in Febraury 2009. One device was placed "upstream" near the generator facility- a typical installation point non-intrusive monitoring. The currents measured by this device represent an aggregate of multiple loads. To determine the ability of this single device to effectively isolate and monitor individual loads, a second NILM was placed "downstream" for comparison. The experimental setup is shown in Fig. 16 and fully detailed in [5]. In this configuration, the effective resolution of the sensor and ADC for "Load 3" at the upstream NILM was 2.3 bits, and the resolution of the downstream NILM was 8.3 bits.

A single run of "Load 3" (a collection-hold-transfer (CHT) vacuum pump) was captured simultaneously by upstream and downstream NILM systems in the described configurations, with no additional loads active at the time. A portion of the pump's transient at shutoff is shown in Figure 17. As calculated, the upstream data shows significantly less resolution and accuracy compared to the downstream data. In the detailed view of Figure 17(b), smaller changes in the raw current
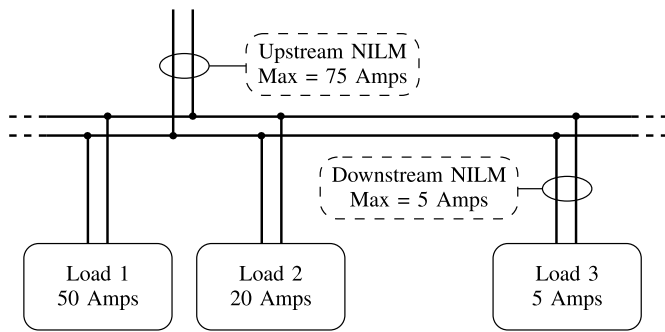
Fig. 16. NILM installation locations on the USCGC Escanaba. The "upstream" NILM monitors all three loads, while the "downstream" NILM monitors only one load.
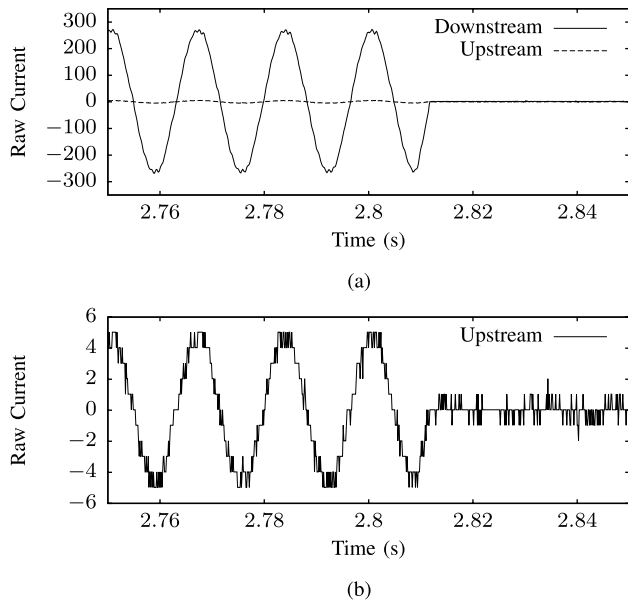

Fig. 17. Two NILM configurations recording the same transient event. The upstream configuration is scaled to handle much larger loads, but cannot resolve the finer details of a single load due to quantization. (a) Raw data from both configurations on the same scale. (b) Raw upstream data, enlarged to show detail.

are indistinguishable from the quantization noise due to the reduced accuracy.

In general, the accuracy of the raw current data acquisition for any particular load will linearly decrease as the maximum load capacity of the NILM increases, given that the ADC parameters are fixed. For example, if the NILM is sized to handle twice as much aggregate current, then the power usage of any individual load will be measured with half the resolution. As the NILM scales to more loads, the raw resolution may be severely reduced.

Fortunately, preprocessing the data using an algorithm such as Sinefit, can recover most of this lost resolution. As discussed in Section V-C, quantization error in the raw data will then have a limited impact, as long as the mean value still approaches the actual power usage of the load. In effect, the preprocessor uses the high-speed, quantized sampled data from the data acquisition to produce good quality output at a lower rate.
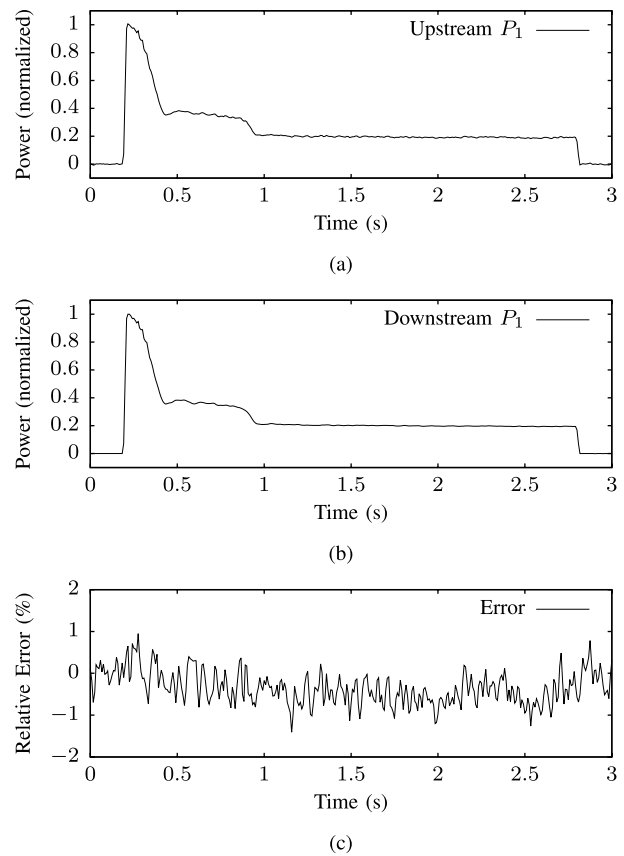

Fig. 18. Comparison of preprocessor spectral envelope output for the raw waveforms from Figure 17. (a) Spectral envelope of the downstream measurement. (b) Spectral envelope of the upstream measurement. (c) Difference as a percentage of the full transient amplitude.

Figure 18 demonstrates this effect of the preprocessor on the two NILM configurations tested on the USCGC ESCANABA. Although the upstream quantization error in Figure 18 is readily apparent, the preprocessor's spectral envelope output for downstream and upstream data in Figures 18 and 18 are very similar. Both show fine details of the pump's power usage, particularly during the startup transient. Figure 18 demonstrates this accuracy by plotting the error as a percentage of the full amplitude of the transient; the error is generally within $\pm 1\%$ for the duration of the pump run.

This result demonstrates that moving the NILM upstream does not cause significant loss of accuracy due to quantization and measurement error, in the common case that spectral envelopes are the metric used for identification and diagnostics.

VII. CONCLUSION

The spectral envelope preprocessor accurately extracts relevant harmonic information while providing data storage reduction. The applicability of the preprocessor to complex systems such as multi-phase systems and variable speed drives has been improved by the development of a more flexible and modular Sinefit preprocessor design with improved phase and frequency estimation [14], [15].

REFERENCES

[1] S. B. Leeb, S. R. Shaw, and J. L. Kirtley, Jr., "Transient event detection in spectral envelope estimates for nonintrusive load monitoring," *IEEE Trans. Power Del.*, vol. 10, no. 3, pp. 1200–1210, Jul. 1995.

[2] S. R. Shaw and C. R. Laughman, "A Kalman-filter spectral envelope preprocessor," *IEEE Trans. Instrum. Meas.*, vol. 56, no. 5, pp. 2010–2017, Oct. 2007.

[3] C. R. Laughman *et al.*, "Power signature analysis," *IEEE Power Energy Mag.*, vol. 1, no. 2, pp. 56–63, Mar. 2003.

[4] Z. Remscrim *et al.*, "FPGA-based spectral envelope preprocessor for power monitoring and control," in *Proc. Appl. Power Electron. Conf.*, Palm Springs, CA, USA, Feb. 2010, pp. 2194–2201.

[5] J. Paris *et al.*, "Scalability of non-intrusive load monitoring for shipboard applications," in *Proc. Amer. Soc. Naval Eng. Day*, National Harbor, MD, USA, Apr. 2009.

[6] S. R. Shaw, C. R. Laughman, S. B. Leeb, and R. F. Lepard, "A power quality prediction system," *IEEE Trans. Ind. Electron.*, vol. 47, no. 3, pp. 511–517, Jun. 2000.

[7] Y. Lin and M. Tsai, "Development of an improved time-frequency analysis-based nonintrusive load monitor for load demand identification," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 6, pp. 1470–1483, Jun. 2014.

[8] C. Duarte, P. Delmar, K. W. Goossen, K. Barner, and E. Gomez-Luna, "Non-intrusive load monitoring based on switching voltage transients and wavelet transforms," in *Proc. Future Instrum. Int. Workshop (FIIW)*, Oct. 2012, pp. 1–4.

[9] A. Routray, A. Pradhan, and K. Rao, "A novel Kalman filter for frequency estimation of distorted signals in power systems," *IEEE Trans. Instrum. Meas.*, vol. 51, no. 3, pp. 469–479, Jun. 2002.

[10] *IEEE Standard for Terminology and Test Methods for Analog-to-Digital Converters*, IEEE Standard 1241-2010, pp. 1–139, 2011.

[11] J. Schoukens, R. Pintelon, and H. Van Hamme, "The interpolated fast Fourier transform: A comparative study," *IEEE Trans. Instrum. Meas.*, vol. 41, no. 2, pp. 226–232, Apr. 1992.

[12] L. Schuchman, "Dither signals and their effect on quantization noise," *IEEE Trans. Commun. Technol.*, vol. 12, no. 4, pp. 162–165, Dec. 1964.

[13] K. D. Lee, S. B. Leeb, L. K. Norford, P. R. Armstrong, J. Holloway, and S. R. Shaw, "Estimation of variable-speed-drive power consumption from harmonic content," *IEEE Trans. Energy Convers.*, vol. 20, no. 3, pp. 566–574, Sep. 2005.

[14] J. Paris, "A comprehensive system for non-intrusive load monitoring and diagnostics," Ph.D. dissertation, Dept. Electr. Eng. Comput. Sci., Massachusetts Inst. Technology, Cambridge, MA, USA, Sep. 2013.

[15] C. Schantz and S. Leeb, "Non-intrusive fault detection in reciprocating compressors," in *Proc. Int. Refrig. Air Conditioning Conf.*, Jul. 2012.

John S. Donnal received the B.S. degree in electrical engineering from Princeton University, Princeton, NJ, USA, in 2007, and the M.S. degree in electrical engineering from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2013, where he is currently pursuing the Ph.D. degree. His current research interests include nonintrusive load monitoring synthesis, energy harvesting, and communications systems.

Zachary Remscrim is working towards the Ph.D. degree at the Massachusetts Institute of Technology, Cambridge, MA, USA.

Steven B. Leeb (F'07) received the Ph.D. degree from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 1993. He has been a member of the faculty of the Department of Electrical Engineering and Computer Science at MIT since 1993. He also holds a joint appointment with the Department of Mechanical Engineering at MIT. He is concerned with the development of signal processing algorithms for energy and real-time control applications.

James Paris received the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2013.

Steven R. Shaw received the Ph.D. degree from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2000. He is currently an Associate Professor with Montana State University, Bozeman, MT, USA. He is interested in system identification and controls, energy and fuel cell systems, and instrumentation.