The Sinefit Spectral Envelope Preprocessor

James Paris, Zachary Remscrim, Dr. Steven B. Leeb, Dr. Steven R. Shaw

***Abstract*—This paper presents a new spectral envelope preprocessor based on sinusoid fitting and the Discrete Fourier Transform (DFT). This preprocessor is well-suited for non-intrusive 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 the presence of higher frequency spectral harmonics.**

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.

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].

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. The preprocessor is implemented within the NilmDB framework [6], allowing reuse and replacement of computation components for related or optimized calculations. Sinefit solves problems with existing preprocessors by providing more robust phase and frequency detection, accurate timestamping of spectral envelopes, and improved and quantifiable accuracy.

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.

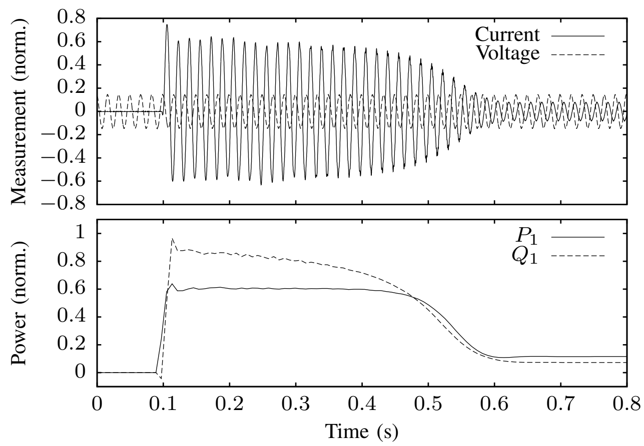


Fig. 1: Raw ac voltage and current measurements (top) and computed spectral envelopes (bottom).

A. Definition

For the 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:

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

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

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

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:

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

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

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

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 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}.

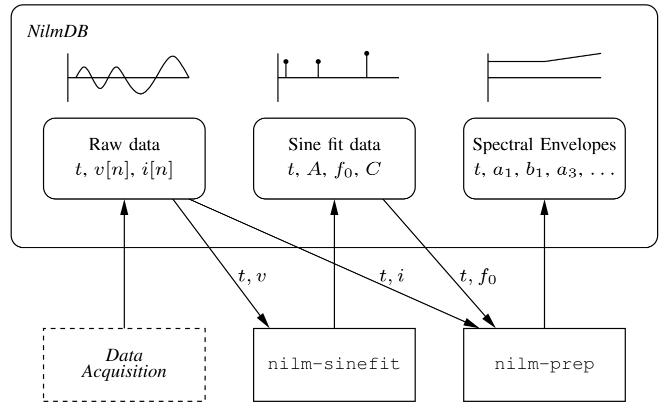


Fig. 2: Block diagram of signal flow in the Sinefit spectral envelope preprocessor.

III. Implementation

Existing preprocessors such as the Kalman-filter spectral envelope preprocessor, described in [2], have been heavily used in non-intrusive load monitoring and diagnostics [3], [7]–[9]. However, a number of practical shortcomings of this preprocessor have been identified [6]. To address these, a new approach to spectral envelope preprocessing has been developed within the NilmDB framework, a unified system for managing and processing NILM data [6].

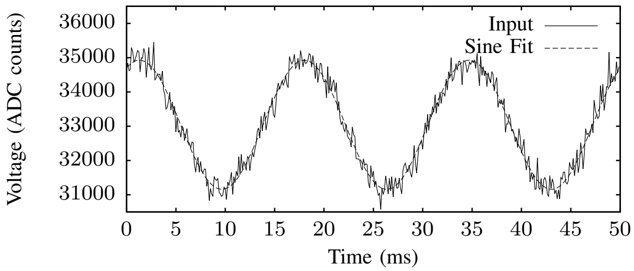
The new preprocessor has a modular, transparent design that allows for easy replacement and reuse of components, such as the phase alignment or harmonic coefficient calculation. It also fully supports the capabilities of NilmDB by utilizing its stream and metadata metaphors, and by incorporating accurate timestamping for all data. The preprocessor uses a phase and frequency estimation algorithm 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.

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. 2, and the implementations of these components are detailed below.

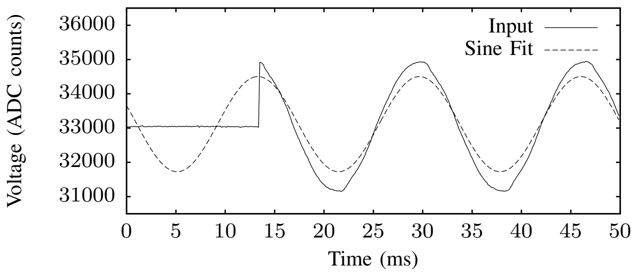
A. 4-Parameter Sinusoid Fitting

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. 3 demonstrates this optimal fit for two representative waveforms.

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



(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.

Fig. 3: Results of 4-parameter sinusoid fitting.

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

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

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 $\mathbf{v}$, 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 estimatation techniques can be used; for example, the implementation in §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:

$$A_0 = B_0 = C_0 = 0 \tag{11}$$

$$\omega_0 = 2\pi \cdot f_{0,\text{est}} \tag{12}$$

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 (10) around the current estimated parameters gives the linear equation:

$$\mathbf{v_i'} = \mathbf{D_i} \cdot \begin{bmatrix} A_i & B_i & C_i & \Delta\omega_i \end{bmatrix}^T \tag{13}$$

where

$$\mathbf{D_i} = \begin{bmatrix} \cos(\omega_i \mathbf{t}) & \sin(\omega_i \mathbf{t}) & 1 & \mathbf{d_i} \end{bmatrix} \tag{14}$$

$$\mathbf{d_i} = B_i \mathbf{t} \circ \cos(\omega_i \mathbf{t}) - A_i \mathbf{t} \circ \sin(\omega_i \mathbf{t}) \tag{15}$$

$$\mathbf{t} = \begin{bmatrix} 0, & 1, & \ldots, & (N-1) \end{bmatrix} / f_s \tag{16}$$

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

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

$$\omega_{i+1} = \omega_i + \Delta\omega_{i+1} \tag{18}$$

Note that the least-squares fit in (17) 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 (10) are:

$$A' = A_m \qquad B' = B_m \tag{19}$$

$$C = C_m \qquad f_0 = \frac{\omega_m}{2\pi} \tag{20}$$

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

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

by computing:

$$A = \sqrt{A_m^2 + B_m^2} \tag{22}$$

$$f_0 = \frac{\omega_m}{2\pi} \tag{23}$$

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

$$C = C_m \tag{25}$$

This form is preferable because the parameters $[A, f_0, \phi_0, C]$ are more directly applicable for 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

$$N = 3.5 \cdot f_s / f_{\text{exp}} \tag{26}$$

samples of voltage. This corresponds to approximately 3.5 periods, although the actual number will vary with f_0. If the fitted f_0 falls outside predetermined bounds, or the amplitude A is too low, the window is skipped to avoid spurious marks.

Fig. 4 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 NilmDB 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

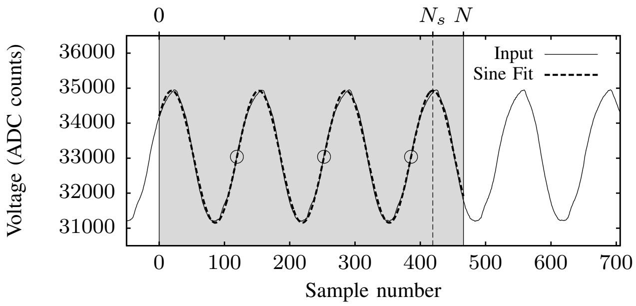


Fig. 4: Windowing algorithm of `nilm-sinefit`. Sine fit is performed over the window $[0, N]$, and positive zero crossings are marked as shown by the circles. The next window starts at N_s, the last point with phase angle $\frac{\pi}{2}$.

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.

Table 1: Parameters for `nilm-prep`.

floating-point values A, f_0, and C. These values correspond to the parameters from (21), 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 1. 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 (9) 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) and (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 NilmDB stream where each row contains a timestamp and $2 * N_{\text{harm}}$ floating-point values numbers, in order $\{a_1, b_1, a_3, b_3, \ldots\}$.

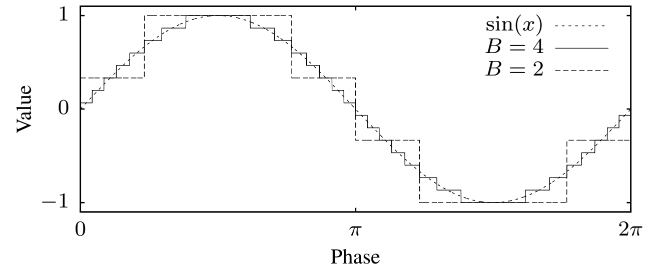


Fig. 5: Quantization error, as would be introduced by a linear ADC. The continuous function $\sin(x)$ is compared to the same function represented discretely with only 4 and 2 bits of resolution.

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. 5, the continuous input signal is divided into B discrete regions, and each region is mapped to a unique

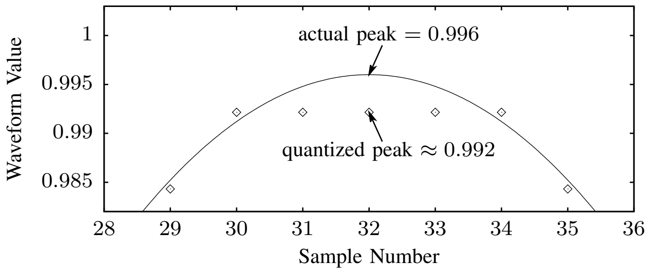


Fig. 6: Example of the error caused by estimating signal amplitude using the peak quantized sample value alone. Here, $A = 0.996$, $N = 128$, $B = 8$.

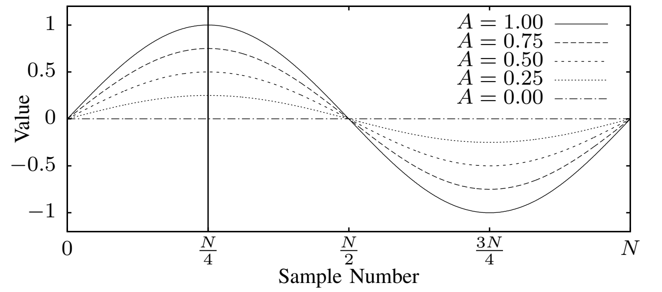


Fig. 7: Sweeping through all input amplitudes of a single fundamental frequency from 0 to 1, in order to enumerate the number of possible quantized inputs. By symmetry, only the first $N/4$ samples need to be considered.

digital code during the sampling process. This quantization can be stated explicitly as

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

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.

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 §V-E.

A. Resolution of Power From Raw Data

A straightforward approach to estimating A from raw sampled data is to locate the peak of the waveform. However, as shown in Fig. 6, finding the amplitude in this manner leads to measurement error, because the single point at the peak is subject to the same quantization as any other point. Of the 2^B possible quantized sample values, the positive peak will reside in the upper half, resulting in a total of $2^{(B-1)}$ discernable amplitudes. This corresponds to an overall resolution of

$$\log_2 2^{(B-1)} = B - 1 \tag{31}$$

bits of precision, slightly less than the original sampling precision of B bits.

B. Resolution of Power From Preprocessor Output

The preprocessor improves effective power reesolution by averaging over time. Specifically, `nilm-prep` uses not only the quantized value at the peak, but 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 (30) 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 of the 1:1 relationship of the DFT, there are the same number U of unique preprocessor outputs, and we can therefore discern U different values of A.

To calculate U, consider sweeping A from $0 \to 1$, as demonstrated in Fig. 7. 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:

$$U_n = \left\lfloor (2^{(B-1)} - 1) \sin\left(\frac{2\pi n}{N}\right) + \frac{1}{2} \right\rfloor \tag{32}$$

where $\lfloor x \rfloor$ denotes $\text{floor}(x)$.

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.

$$U = 1 + 0 + \left(\sum_{n=1}^{\frac{N}{4}-1} U_n \right) + \left(2^{(B-1)} - 1 \right) \tag{33}$$

$$= 2^{(B-1)} + \sum_{n=1}^{\frac{N}{4}-1} \left\lfloor (2^{(B-1)} - 1) \sin\left(\frac{2\pi n}{N}\right) + \frac{1}{2} \right\rfloor \tag{34}$$

Fig. 8 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

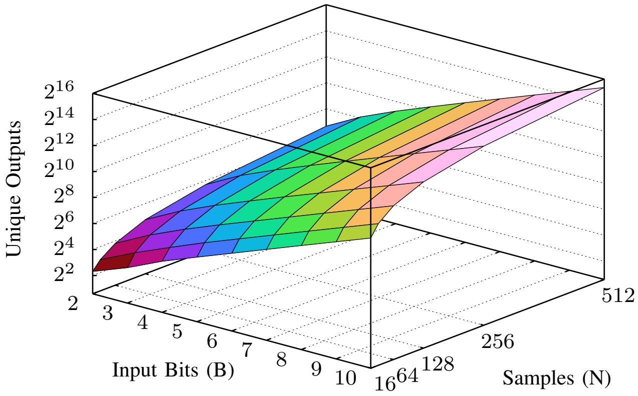


Fig. 8: Number of unique preprocessor outputs, as a function of the number of samples N and the number of input bits B, for an input signal consisting of a single sinusoid of varying amplitude.

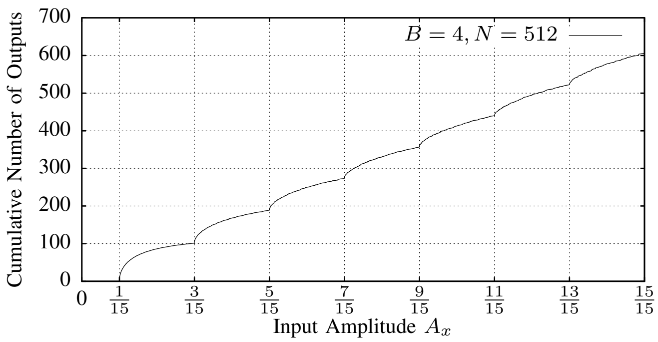


Fig. 9: Unique preprocessor outputs as a function of input amplitude of a sinusoid. For each amplitude level A_x, the graph plots the total number of unique outputs seen while the input sinusoid amplitude is swept from 0 to A_x.

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. 9 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. 9, 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. 10 shows this resolution as a function of input amplitude. Here,

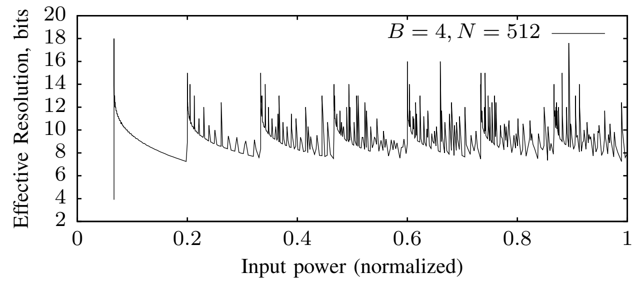


Fig. 10: Effective resolution, in bits, as a function of input amplitude of a sinusoid. An effective resolution of β bits means that a change of $1/(2^\beta)$ in input amplitude can be resolved.

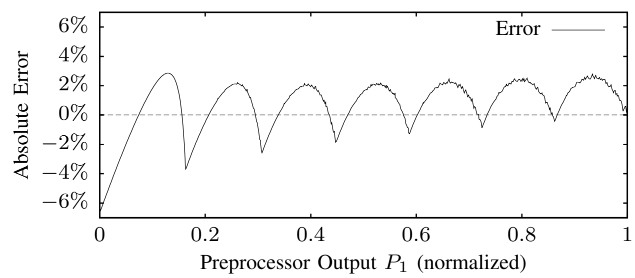


Fig. 11: Accuracy of preprocessor output when provided with quantized input data from a single sinusoid. The absolute error, as a percentage of the full power range, is plotted as a function of the P_1 output. Here, $B = 4, N = 512$, and the maximum error of any single sample due to quantization alone would be $1/(2^B) = 6.25\%$.

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.

C. Accuracy of Preprocessor Output

For load identification, the ability to simply discern inputs may be sufficient. As shown, the preprocessor can provide a high resolution estimate of input waveform amplitude. 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. 11 shows a 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.

D. 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 $\mathcal{N}(t)$:

$$i(t) = A\sin(\omega t) + \mathcal{N}(t) \tag{35}$$

By sweeping A from $0 \to 1$, we can generate data in the same manner as Fig. 11, 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:

$$E_{\text{RMS}} = \sqrt{\int_0^1 \left(P_1(A) - A\right)^2 dA} \tag{36}$$

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. Note that the amount of noise is held constant as A is swept.

Fig. 12 shows the calculated overall RMS error versus the SNR of injected noise. Two approaches to estimating power are shown: the preprocessor P_1 estimate, and the simple peak-estimation technique described in §V-A. 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 waveform [12]. The preprocessor can therefore be seen as even more useful in the presence of noise.

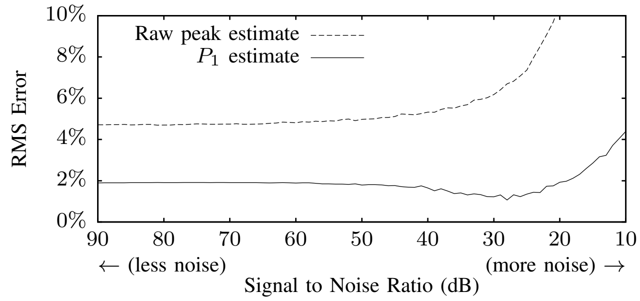


Fig. 12: Total RMS error of power estimates for a single sinusoid over all amplitudes $A = 0 \to 1$, with simulated additive Gaussian white noise injected at the specified signal-to-noise ratio. At some levels, white noise can improve preprocessor accuracy by introducing dither into the quantization process. The sampling parameters are $B = 4, N = 512$.

E. 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 (34) may not necessarily exist.

Instead, we develop a model $f(A)$ that describes the current waveform shape as a function of power level $A = 0 \to 1$. 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 of the form:

$$\sqrt{a_k^2 + b_k^2} = bA^p \tag{37}$$

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:

$$f(A, t) = A \cdot \left(\sin(\omega t) + A_3\sin(3\omega t) + A_5\sin(5\omega t)\right) \tag{38}$$

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 §V-B, we do this by counting the number of unique preprocessor outputs as we sweep $A = 0 \to 1$. 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.

outputs ← [ ]
function ENUMERATE($f, A_{\text{low}}, A_{\text{high}}$)
  $P_{\text{low}} \leftarrow$ PREPROCESS(SAMPLE($f(A_{\text{low}})$))
  $P_{\text{high}} \leftarrow$ PREPROCESS(SAMPLE($f(A_{\text{high}})$))
  if $P_{\text{low}} \neq P_{\text{high}}$ **then**
    $A_{\text{mid}} \leftarrow (A_{\text{high}} + A_{\text{low}})/2$
    ENUMERATE($f, A_{\text{low}}, A_{\text{mid}}$) ▷ Search 1st half
    ENUMERATE($f, A_{\text{mid}}, A_{\text{high}}$) ▷ 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)

Algorithm 1: Binary search to enumerate unique preprocessor outputs as power A varies, given waveform model $f(A)$.

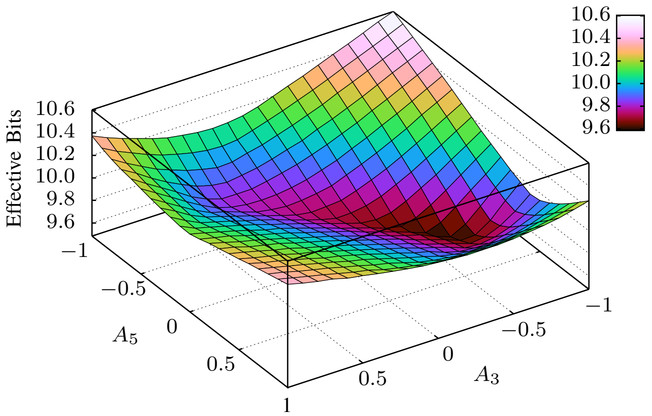


Fig. 13: Effective bits of preprocessor resolution as relative amounts of harmonic content A_3 and A_5 are varied in (38). The harmonics are held at a fixed ratio while the overall waveform scaling is swept from $A = 0 \to 1$. Sampling parameters are $B = 8$, $N = 128$.

The enumeration process was performed for varying combinations of harmonic content ratios A_3 and A_5 in (38). 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. 13 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. 14 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 final waveform exhibits more complexity will generally see improvements in preprocessor output resolution.

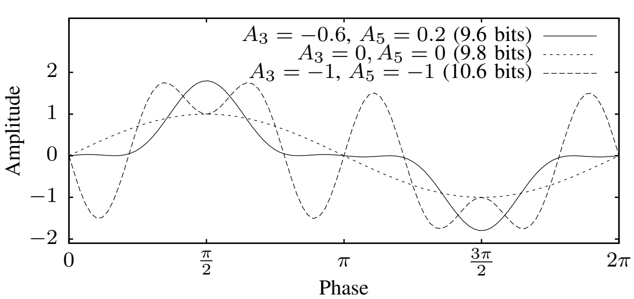


Fig. 14: Full-amplitude waveforms corresponding to the minimum and maximum resolution from Fig. 13. In general, increasing complexity increases the effective resolution, although some combinations of harmonics reduce it slightly due to "flat" regions that do not vary much with power scaling.

VI. CONCLUSIONS

The spectral envelope preprocessor has 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. The new preprocessor is designed to integrate with the NilmDB framework and builds upon its ability to correlate, manipulate, and retrieve interrelated data streams.

ACKNOWLEDGEMENTS

This research was funded by the BP-MIT Research Alliance, the ONR Structural Acoustics Program, and The Grainger Foundation.

REFERENCES

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

[2] S. R. Shaw and C. R. Laughman, "A kalman-filter spectral envelope preprocessor," *IEEE Transactions on Instrumentation and Measurement*, vol. 56, no. 5, pp. 2010–2017, October 2007.

[3] C. R. Laughman, S. R. Shaw, S. B. Leeb, L. K. Norford, R. W. Cox, K. D. Lee, and P. Armstrong, "Power signature analysis," *IEEE Power and Energy Magazine*, pp. 56–63, March 2003.

[4] Z. Remscrim, J. Paris, S. B. Leeb, S. R. Shaw, S. Neuman, C. Schantz, S. Muller, and S. Page, "FPGA-Based Spectral Envelope Preprocessor for Power Monitoring and Control," in *Applied Power Electronics Conference*, Palm Springs, CA, February 2010.

[5] J. Paris, Z. Remscrim, K. Douglas, S. B. Leeb, R. W. Cox, S. T. Gavin, S. G. Coe, J. R. Haag, and A. Goshorn, "Scalability of non-intrusive load monitoring for shipboard applications," in *American Society of Naval Engineers Day 2009*, National Harbor, Maryland, April 2009.

[6] J. Paris, "A comprehensive system for non-intrusive load monitoring and diagnostics," Ph.D. dissertation, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, September 2013.

[7] W. Greene, J. S. Ramsey, S. B. Leeb, T. DeNucci, J. Paris, M. Obar, R. Cox, C. Laughman, and T. J. McCoy, "Non-intrusive monitoring for condition-based maintenance," in *American Society of Naval Engineers Reconfigurability and Survivability Symposium*, Atlantic Beach, Florida, February 2005.

[8] T. DeNucci, R. Cox, S. B. Leeb, J. Paris, T. J. McCoy, C. Laughman, and W. Greene, "Diagnostic indicators for shipboard systems using non-intrusive load monitoring," in *IEEE Electric Ship Technologies Symposium*, Philadelphia, Pennsylvania, July 2005.

[9] R. W. Cox, P. Bennett, D. McKay, J. Paris, and S. B. Leeb, "Using the non-intrusive load monitor for shipboard supervisory control," in *IEEE Electric Ship Technologies Symposium*, Arlington, VA, May 2007.

[10] "IEEE Standard for Terminology and Test Methods for Analog-to-Digital Converters," *IEEE Std 1241-2010 (Revision of IEEE Std 1241-2000)*, pp. 1–139, 2011.

[11] J. Schoukens, R. Pintelon, and H. Van Hamme, "The interpolated fast fourier transform: a comparative study," *Instrumentation and Measurement, IEEE Transactions on*, vol. 41, no. 2, pp. 226–232, 1992.

[12] L. Schuchman, "Dither signals and their effect on quantization noise," *Communication Technology, IEEE Transactions on*, vol. 12, no. 4, pp. 162–165, 1964.

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