Circuit-Level Load Monitoring for Household Energy Management

Circuit-level nonintrusive load monitoring can overcome some of the basic problems relative to whole-house NILM, enabling sophisticated energy-monitoring applications. The two proposed approaches demonstrate the effectiveness of this technique.

Alan Marchiori,
Douglas Hakkarinen, and Qi Han
Colorado School of Mines

Lieko Earle
National Renewable Energy Laboratory

Nonintrusive load monitoring (NILM) first appeared in the 1980s, but there's currently a renewed interest in understanding and reducing building energy use. Commercial interest in NILM includes Google PowerMeter (www.google.org/powermeter), a system developed by Greenbox (http://getgreenbox.com), and the established NILM products from Enetics (www.enetics.com).

A reasonable prerequisite for reducing buildings' energy consumption is a practical method to monitor consumption. There are two approaches to monitoring household energy. The first uses a single NILM meter to measure aggregate energy use as power enters the home. NILM is attractive because it's easy, requiring only one meter to monitor whole-house energy consumption. This approach analyzes the whole-house data and matches step changes in power use to a load database—for example, a 500-watt step might represent a refrigerator turning on. This works remarkably well for large (more than 150-W) loads that operate as simple on/off devices or have simple operating states, such as high, medium, and low.[1]

However, low-power loads and devices with numerous states, such as a dishwasher, or continuously variable energy use, such as an electric stove, are difficult to extract from whole-house measurements. Hence, the second approach measures each load in the home separately, using complex instrumentation systems that individually meter each device's energy consumption[2–4] (see also www.tendrilinc.com). Although this approach provides accurate data for each device, it requires a significant investment in monitoring equipment.

Here, we describe two approaches for circuit-level NILM that enable detailed, practical household energy monitoring: a heuristic-based approach and a Bayesian approach. Both address the limitations of previous NILM algorithms by considering steady-state power use in addition to step changes in steady-state power use and by using circuit-level energy measurements. Using the historical steady-state energy use pattern for each device lets us more precisely classify each edge and eliminate some cases of indistinguishable step changes. By measuring energy use at the circuit level, we can overcome the inability of whole-house NILM to monitor small or variable-power devices. This is because there are fewer devices on each circuit and high-powered devices (such as a stove, hot-water heater, air conditioner, and clothes dryer) each receive dedicated circuits and won't overshadow lower-powered devices (such as TVs, radios, and cell phone chargers).

Our Heuristic Approach

This approach builds on the basic ideas in the original NILM work (see the "Related Work in

Published by the IEEE CS ■ 1536-1268/11/$26.00

Related Work in Nonintrusive Load Monitoring

Fred Schweppe and George Hart originally developed nonintrusive load monitoring (NILM) in the 1980s at the Massachusetts Institute of Technology. Hart later described the NILM eight-step process:[1]

1. measure the power and voltage,
2. normalize,
3. edge detection,
4. cluster analysis,
5. build an appliance model,
6. track behavior in terms of models,
7. tabulate statistics, and
8. appliance naming.

In recent commercial solutions, a smart meter reports whole-home energy use to a server for analysis. These solutions use an NILM algorithm to process the information, and then they present the information to the user through a Web-based portal. The NILM approach that companies such as Greenbox (http://getgreenbox.com) and Enetics (www.enetics.com) use isn't publicly available. In fact, exact statistics on the accuracy of most NILM systems are scarce. However, according to one study, a three-home NILM system reported "75 percent to 90 percent of on/off events."[2] Hart's preliminary results suggest that NILM "usually reports energy consumption within ±10% of the independent sensors."[1] Researchers have identified that the original NILM research had three limiting assumptions: loads are distinguishable, loads are steady state, and data is batch processed.[3]

Other researchers are also considering novel ways to extend the basic NILM process. These approaches complement ours because each approach uniquely extends NILM's capabilities. ViridiScope couples whole-house power measurements with indirect sensors in the home.[4] The indirect sensors capture additional information (using magnetic, light, and acoustic sensors) that can be used during disaggregation. For example, if a light sensor in the kitchen detects an abrupt increase at the same time as the whole-house power consumption increases, we can deduce that the kitchen light has turned on and the observed power increase is attributable to that light. Another idea is to detect transient noise caused by devices turning on or off.[5] This approach, which requires high-frequency sampling (from 100 Hz to 100 MHz), recognizes devices by their spectral fingerprint.

Although both these approaches are promising, no single approach has yet been proven universally effective (including our own). The most effective systems will likely incorporate ideas from each of these approaches.

REFERENCES

1. G. Hart, "Nonintrusive Appliance Load Monitoring," *Proc. IEEE*, vol. 80, no. 12, 1992, pp. 1870–1891.

2. G. Hart, "Residential Energy Monitoring and Computerized Surveillance via Utility Power Flows," *IEEE Technology and Society Magazine*, vol. 8, no. 2, 1989, pp. 12–16.

3. C. Laughman et al., "Power Signature Analysis," *IEEE Power and Energy Magazine*, vol. 1, no. 2, 2003, pp. 56–63.

4. Y. Kim et al., "ViridiScope: Design and Implementation of a Fine Grained Power Monitoring System for Homes," *Proc. 11th Int'l Conf. Ubiquitous Computing* (UbiComp 09), ACM Press, 2009, pp. 245–254.

5. S.N. Patel et al., "At the Flick of a Switch: Detecting and Classifying Unique Electrical Events on the Residential Power Line," *UbiComp 2007: Ubiquitous Computing*, Springer, 2007, pp. 271–288.

NILM" sidebar for more details), while addressing its weaknesses through several steps to enable more complete device-level energy monitoring. First, as we mentioned before, we use circuit-level energy measurements to simplify the analysis. Because only a handful of devices are on each circuit, we expect a lower occurrence of indistinguishable devices.

Second, we use a probabilistic level-based disaggregation algorithm rather than an edge-based algorithm. Because this algorithm doesn't require a step change in energy use to identify devices, we're better able to monitor devices with a complex state or continuously variable power use. Although the level-based approach might not perform well with whole-home energy measurements, it's effective here because of the reduced number of devices in each circuit-level energy measurement.

We train the system by detecting each device as the user turns it on and off; we can also do this automatically by using a control system to switch individual devices on and off and measure the effect on power use. Both training methods are autonomous and generate samples of each device's power use.

Figure 1 shows the steps in our heuristic algorithm for circuit-level measurements.

Training

Step 1 (see Figure 1a) measures the real (P) and reactive (Q) power, line voltage, and frequency once per second. The sampling rate we use is the fastest supported by our energy meter (see www.ccontrolsys.com/products/wattnode_modbus.html). Step 2 normalizes the power measurements to a 120-V line voltage, as in previous research.[1] Step 3 constructs a 2D histogram of the measured power for each device in $P-Q$ space collected during training.

Step 4 identifies clusters corresponding to the different device states. Our clustering approach uses histogram thinning to compute the location and

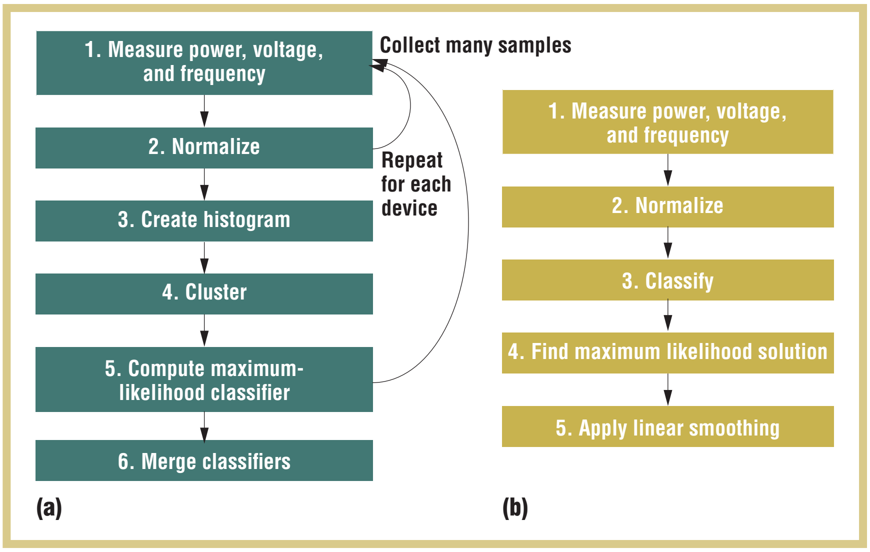


Figure 1. Our heuristic nonintrusive load monitoring (NILM) algorithm. (a) Training and (b) processing begin by measuring the real (P) and reactive (Q) power, line voltage, and frequency once per second. This algorithms uses circuit-level energy measurements to simplify the analysis.

number of clusters. The thinning process first discards outliers with less than 0.1 percent of the data. The next step is to find local peaks, where all the adjacent cells have a lower value. It then increases the peak's value while decreasing the neighboring cells' value. The peak-thinning process repeats until every neighboring cell of the local peak has neighbors that are all zero. The number of peaks determines the number of states K. The (nonthinned) histogram value at each peak is the height of the peak.

To classify a point, we compute the cost to move in a straight line from the unknown point in $P - Q$ space to each peak in the histogram. The cost is the sum of the steps needed to reach the peak. If the height decreases, we know a different peak is closer, so we can eliminate the current class as a solution. If we reach a point with zero height, we immediately disregard that peak. The lowest-cost solution determines which cluster the unknown point belongs to. If no peaks are reachable from the current location, the point doesn't belong to any of the device's classes. In this way, we treat changes in both P and Q equally.

Step 5 classifies each histogram cell and computes the probability mass function (PMF) for each of the K classes. The PMF has the same dimensions as the histogram and represents the probability of observing the $P - Q$ measurement conditioned on the device being in the state identified in step 4. We use the PMF in the next step to compute the maximum-likelihood merged classifier.

Finally, in step 6, the merging process creates a classifier by computing all possible combinations of individual device states and their probability. When overlap occurs in the merged classifier, we use the set of classes that maximizes the joint probability. The merged classifier appears blurry compared to the individual classifiers owing to a quantization error in the combined measurements. For example, if we have two classifiers with a 1-W^2 cell size, the points in the rectangle with corners at (0, 0) and (1, 1) go in the first histogram cell. When we merge the classifier, we must account for this possible range of values, so the resulting merged range includes the four cells contained by the square with corners at (0, 0) and (2, 2).

Processing

We use the combined classifier to decompose the circuit-level energy measurement into device-level energy estimates, using the process in Figure 1b. After measuring and normalizing the power data (steps 1 and 2), we use the combined classifier to classify the most likely state for each device (step 3). Next, we go back to the individual devices' histograms and find the most likely energy use of each device that results in the same total energy (measured in step 3). The result is limited to the histograms' resolution, so we linearly smooth the result so that the sum equals the measured value (step 5).

An Example

Figure 2 shows the measurements and computational steps in creating a classifier for a PC. Figure 2a shows the raw power data plotted in $P - Q$ space. The few samples between the two clusters represent the transient created by switching between standby and active mode. The outlier detection step in the thinning process successfully identified and removed these samples. Figure 2b shows the histogram of the raw data using a bin size of 1 W in each dimension. The histogram is then thinned to determine the number of device states. Figure 2c shows the computed peaks; each peak represents the most likely measurement in each state. We repeat this process for each device and then compute each maximum-likelihood classifier.

We can use the classifier's output to determine which state the device is in. Once the state is known, we use the original histogram as an estimate of the PMF in each state. We could use the histogram as an estimate of the PMF without constraining the state, but this would bias the estimate to favor the most common state and thus make uncommon states highly unlikely.

The next step generates a classifier for each device and combines them into a single classifier for the combined energy consumption. We do this by computing every possible combination of each device. We sum the resulting power and use each device's PMF to compute the joint probability of the combined solution. We save the set of classes that maximize the joint probability as the final classification for each cell in the combined classifier.

Our Bayesian Approach

The NILM heuristic method uses probability estimates based on training data and combines them to disaggregate data. This approach has elements of Bayesian probability. To better understand how this process works, we've developed a formalizable purely Bayesian approach. This second approach uses a naive Bayes classifier to compute each device's most likely state given a measured aggregate total and detected state change. Our approach is naive because we assume each device's state is completely independent of the other devices. This is a fair assumption in general, but devices such as TVs and DVD players can have highly correlated operation.

In this approach, the measured real power P is the sole input because we want to independently train devices and then combine them. Because the reactive power Q doesn't combine linearly, we can't train each device independently, which results in less information. To compensate, we added steady-state changes as an additional information source.

Formalizing the Problem

We first define the disaggregated state S:

$$S = \{D_1 = s_1, ..., D_n = s_N\}.$$

S is the set of states for each device D_n, where each device is in some known state s_N. To avoid manually labeling, we represent each state as the steady-state power rounded to the nearest watt. Then Ω is the state space consisting of all valid states. To disaggregate a power measurement p with detected steady-state change (edge) e, we must solve this problem:

$$\arg\max_{s \in \Omega} Pr\left(S \middle| \sum_{n=1}^{N} D_n = p \cap E = e\right).$$

Now we apply Bayes' theorem to this probability:

$$Pr\left(S \middle| \sum_{n=1}^{N} D_n = p \cap E = e\right) = \frac{Pr\left(\sum_{n=1}^{N} D_n = p \cap E = e \middle| S\right) Pr(S)}{Pr\left(\sum_{n=1}^{N} D_n = p \cap E = e\right)}.$$

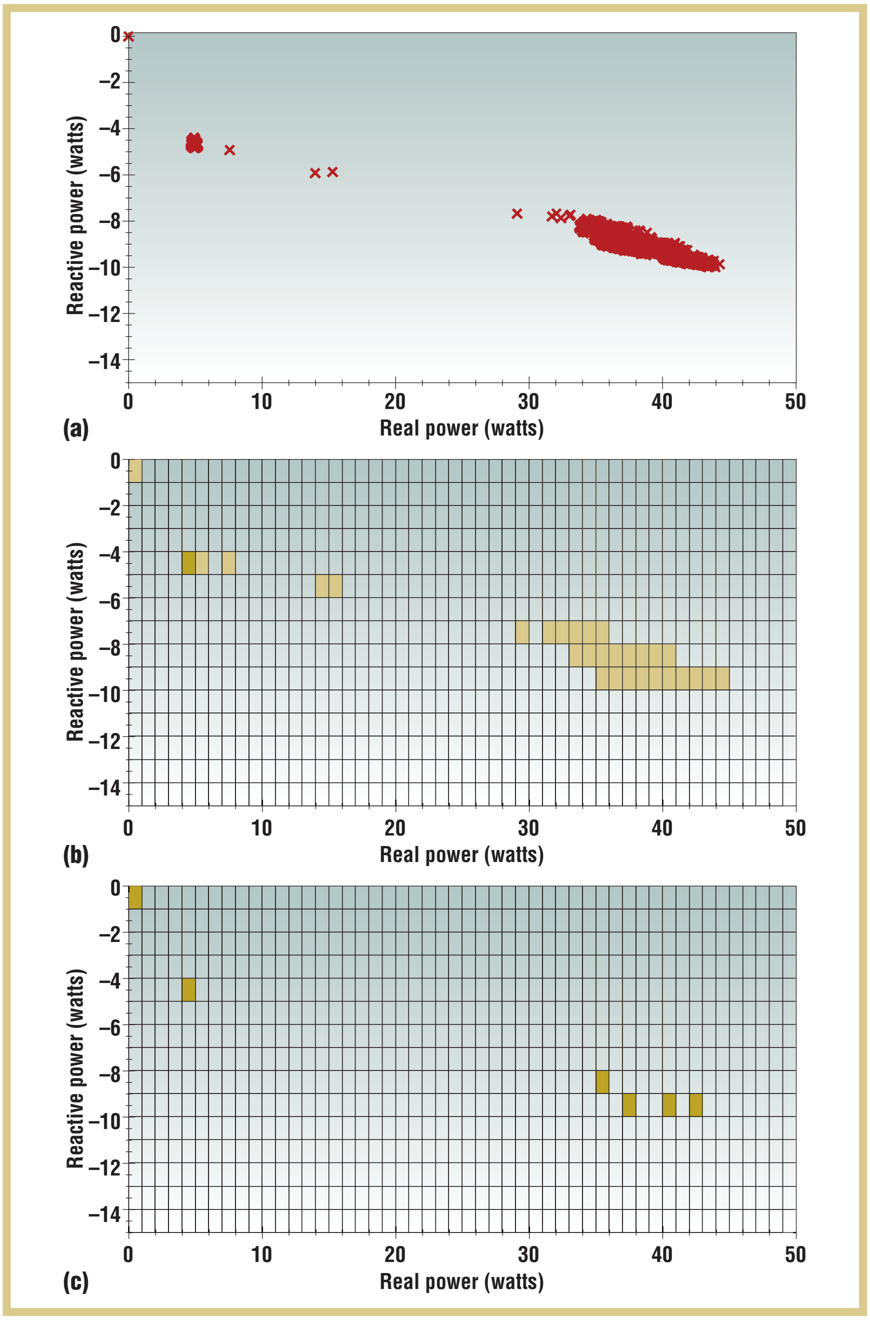


Figure 2. Training data for a low-power PC. We collected (a) the raw power data over several hours. We first created (b) a histogram of the raw data using a bin size of 1 W in each dimension and then (c) a thinned histogram to determine the number of device states.

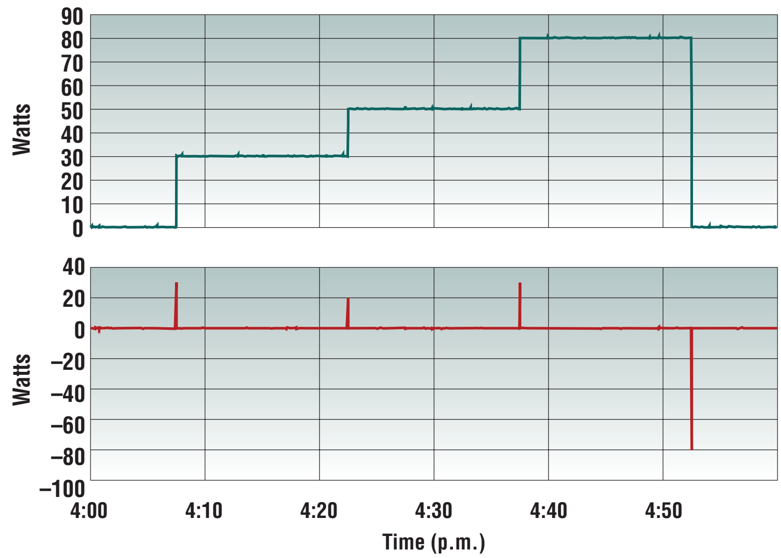


Figure 3. Sample 30-50-80-watt lightbulb measurements of power and detected transitions. The light is in each of the four states for an equal amount of time.

We can then apply the fact that $\sum_{n=1}^{N} D_n = p$ to constrain the search space. Accordingly, this term then is simply 1 and can be eliminated, so we can rewrite the problem as

$$\arg \max_{S \in \Omega : \sum_{n=1}^{N} D_n = p} \frac{Pr\left(E = e \middle| S\right) Pr\left(S\right)}{Pr\left(E = e\right)}.$$

We compute these probabilities to find the most likely solution. Because the denominator doesn't depend on S, we can remove it because it's common to all terms, leaving the final classification problem as

$$\arg \max_{S \in \Omega : \sum_{n=1}^{N} D_n = p} Pr\left(E = e \middle| S\right) Pr\left(S\right).$$

These two terms can be independently computed from training data. We consider the second term, $Pr(S)$, first because it's simply the observed probability of being in a particular state S. Because S actually represents each device's particular state as $S = \{D_1 = s_1, \ldots, D_n = s_N\}$, where we have devices 1 … N, and each device is independent, we can compute this as a product of each device separately as $\prod_{n=1}^{N} Pr\left(D_n = s_n\right)$. The first term, $Pr(E = e|S)$, is the probability of seeing a steady-state change (or edge) of size e and ending in state S.

If we assume that only one device is changing at a time, $Pr(E = e|S)$ is the probability that a device has changed state by e to end up in S. This probability can be estimated from the training data by looking at how many times a change of e ended up in S relative to all changes that ended in S. Let E_s be the set of edges in the training data that resulted in state s. Let I be a function that returns 1 if a condition is true, and 0 otherwise. Under this set of assumptions, we can formalize $Pr(E = e|S)$ as

$$Pr\left(E = e \middle| S\right) \approx \frac{\sum_{i \in E_S} I\left(e = i\right)}{\sum_{i \in E_S} 1}.$$

This analysis shows that, in both cases, we can train the classifier independently on each device and then use the trained classifiers together to disaggregate a set of devices. This lets us create a database of known devices and then select the particular subset contained in the aggregate measurement of interest.

Although not the core topic of this article, we can also consider relaxing the assumption that only one device is changing at a time. In this case, we define a state transition as

$$T = \{t_1, t_2, \ldots, t_N\},$$

where T is a transition consisting of the set of power transitions for each of the N devices. For a single device i changing state, only one t_i will be nonzero. The magnitude of an edge defined by a particular transition T will be

$$e_T = \sum_{i=1}^{N} t_i .$$

To find the probability $Pr(E = e|S)$, we find all transitions with an edge of e. For each transition, we then want to find the probability that the transition resulted in S. Because we trained each device individually, we don't have data on the probability of a transition. Assuming device independence, we can construct this probability by considering all nonzero device state changes that add up to e. For example, if T has two nonzero device state changes a and b, with power changes e_a and e_b, we can find the probability of T_S:

$$Pr(T|S) = Pr(a|S) \times Pr(b|S - e_a) + Pr(b|S) \times Pr(a|S - e_b).$$

This isn't commutative because $S - e_a$ isn't necessarily $S - e_b$. For the generic case, we must consider every ordering of the individual device transitions of that transition. In other words, if a transition has n nonzero transitions, then we must consider $n!$ orderings.

The exhaustive approach to doing this while still being able to train the devices independently requires significant computation. Specifically, we must find every transition combination that has a magnitude of e. In the simplest case, the transition for a specific device will have only three possible values: turning on (positive), staying on or off (zero), or turning off (negative). Reducing the problem further to only consider transitions of turning on or retaining state, this problem effectively is the subset-sum problem, a known NP-complete problem. Because solving this problem will also solve the subset-sum problem, this problem is NP-hard.

Adding additional transitions per device and combinatorics of the devices, this problem will have a worse complexity than just the subset-sum problem. Hence, for anything but a small number of devices that change state, this approach is intractable. You could argue for trying both assumptions of two state changes rather than just one, but the generic case is computationally prohibitive.

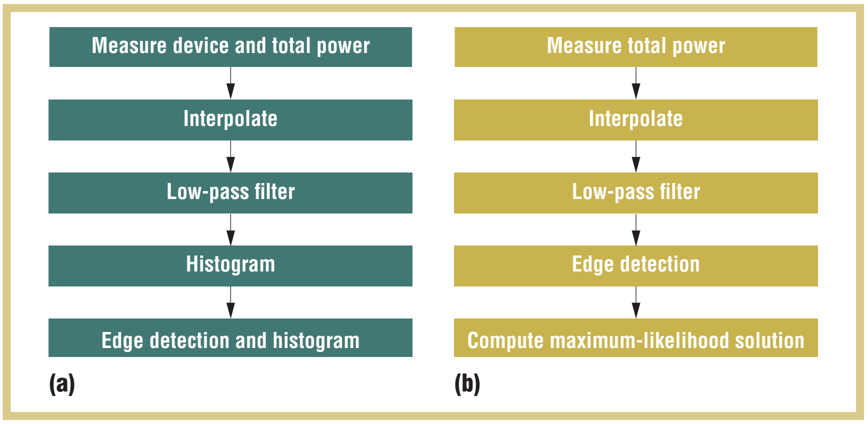


Figure 4. Our Bayesian nonintrusive load monitoring (NILM) algorithm. The (a) training and (b) processing begin by measuring the aggregate or device-level power.

An Example

Imagine a three-way lightbulb. It has two filaments and operates by turning on one filament, the other, and then both. This device has four states: off, low, medium, and high. The light might also actually be off for a brief period between two states. However, because we sample power relatively slowly, we don't measure this period.

Figure 3 shows the measured power and edges for a 30-50-80-W lightbulb. In this trace, the light is in each of the four states for an equal amount of time. To estimate $Pr(S)$, we construct a histogram of the instantaneous power samples. Because the device spends an equal amount of time in each measured power state—0, 30, 50, and 80 W—the estimated probability of being in any one state is 1/4.

To estimate $Pr(E = e|S)$, we must compute a probability estimate for all observed state changes (edges) e_M. Because this is conditioned on a particular device state S, we compute a histogram for each device state. We detected four edges and four distinct steady-state states. In this case, the probability is 1 for each edge because we observed only one distinct edge for each steady-state value. For example, the only edge that resulted in the 50-W state was the +20-W edge. There were two +30-W edges, but they resulted in two distinct states: 30 and 80 W. The final edge is –80 W, which ended in state 0 W.

This example involves a device with obvious states and transitions and a precise training procedure. With more complex devices, this is impossible and not necessary for this process to work properly. If we monitor a PC, for example, we see a wide variation over a 10-W range owing to the changing power states of the various peripherals and the CPU's changing power requirements. These complex interactions are difficult to fully understand and control for training purposes. However, as long as the training period can provide a statistically accurate device usage pattern covering all possible device states, this approach will work.

Figure 4 shows the final algorithm. Both training and processing begin by measuring the aggregate or device-level power. We sample the power at 1 Hz, but it's possible to miss some samples. To correct this problem, we use linear interpolation to estimate missing samples. The data then passes through a low-pass filter to remove high-frequency noise. During training, we generate a histogram of the power samples using a 1-W histogram bin. This generates an estimate of the probability density function (PDF) for the device's instantaneous power use. A histogram is generated for each possible state change in the power measurements. For each detected edge, we store the device's resulting power level. This result lets us estimate the probability that the device will switch to a given power level if the process observes a particular edge.

Evaluation

To explore our approach for circuit-level NILM, we conducted several experiments. In each case, we used the WattNode, a commercially available power meter from Continental Control Systems. Each WattNode measures power, voltage, and frequency once per second with ±1 percent accuracy. One WattNode acts as the circuit-level power meter monitoring the aggregate use of the three devices under test. Each device is then individually monitored by another WattNode. The measured values don't include the power used by the WattNodes (less than 2 W). These evaluations demonstrate that both our NILM algorithms can disaggregate small (10- to 30-W) devices using circuit-level measurements. This is a significant improvement over existing approaches that only detect devices greater than 150 W.

Experiment 1: Long-Term Stability

First, we monitored a PC, LCD, and desk lamp for 24 hours to evaluate each approach's long-term reliability under stable conditions (see Figure 5). Each device was on for the entire experiment, but we used a random screensaver on the PC. As the name implies, the random screensaver selects a different screensaver after random delays. This

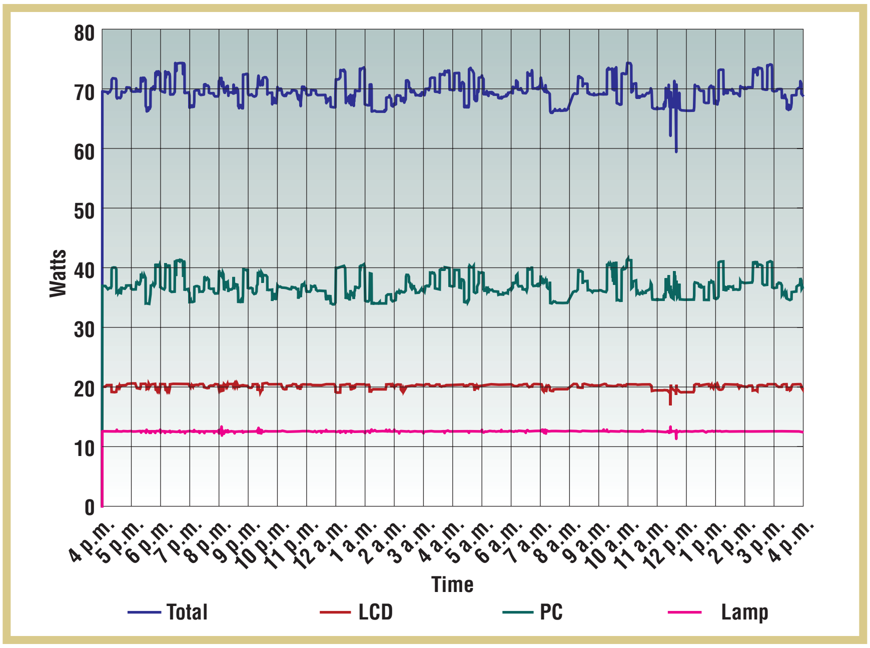


Figure 5. Measured power for a 24-hour period. We monitored a PC, LCD, and desk lamp, using the first two hours for training.

causes the CPU load to vary from 0 to 50 percent and the measured power to vary by more than 10 W. The LCD and lamp show fairly constant power consumption. Because the PC oscillates between many similar states, it's more difficult to properly disaggregate than a device with large step changes such as a refrigerator. We used the first two hours of the 24-hour trace for training (and then excluded this data from the evaluation). Figure 6a shows the results.

We used the true device-level power measurements to compute the error of the disaggregation for each method. Both methods achieved good accuracy over the entire experiment—much better than the ±10 percent error cited by previous NILM approaches. Overall, the Bayesian approach was superior. Because the heuristic approach didn't consider edges, it had some difficulty distinguishing between the PC at high power and the lamp turned off, and the PC at low power and the lamp turned on. In both cases, these two devices used approximately 44 W. In this case, the state-change information was sufficient to disaggregate these two states.

Table 1 shows the resulting error rates, illustrating that circuit-level NILM can successfully disaggregate small steady-state loads with high accuracy.[5]

Experiment 2: Changing States

Because it's also important to accurately detect and classify devices as they change state, we evaluated the Bayesian approach under changing device states. We replaced the PC from the previous example with a three-speed fan to increase the number of distinct device states. We collected six minutes of training data (see Figure 7), in which we manually cycled each device through all possible operating modes.

A challenging feature of this data was that the LCD in the active state is nearly identical to the fan on low speed. In addition, when the LCD transitions from active to standby, it generates transients similar to the lamp turning off or on. As a result, neither the steady-state values nor the state changes alone were sufficient for disaggregation.

We then collected data for a one-hour period in which we manually changed device states in a different order than we used when collecting the training data. Figure 8 shows the measured powers and the Bayesian algorithm's output. The experiment started with the LCD in standby and the fan and lamp turned off. We see a small error because the total energy consumption of 1 W is below our meter's creep limit (the minimum nonzero power). This experiment demonstrates that our algorithms quickly and accurately detect device state changes from the circuit-level aggregate measurements. Table 1 summarizes this experiment's resulting error rates.

Although these results are promising, we look forward to deploying our solution in a real household environ-

TABLE 1
Experimental error rates.

	Long-term stability		Changing states	
Device	Heuristic (%)	Bayesian (%)	Heuristic (%)	Bayesian (%)
PC and fan	0.13	–0.67	–0.69	0.74
LCD	3.10	0.79	1.13	–2.31
Lamp	–5.35	2.91	1.72	0.73

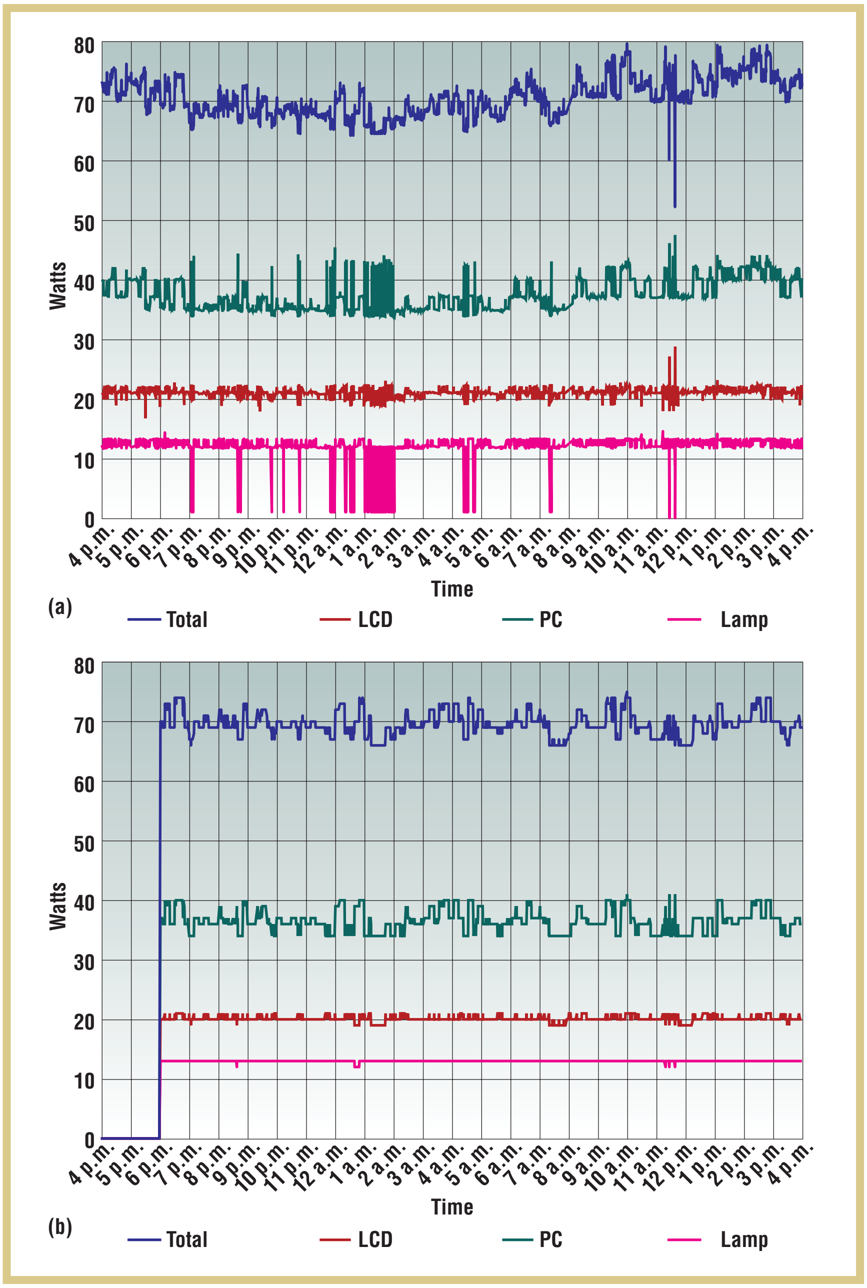


Figure 6. Long-term stability test results. Using the data we collected from monitoring a PC, LCD, and desk lamp, we tried to disaggregate the devices using the (a) heuristic and (b) Bayesian approaches.

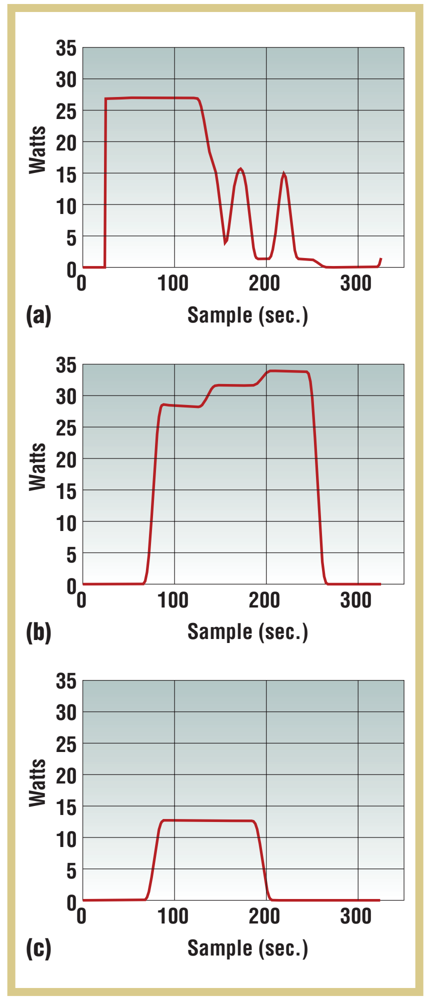


Figure 7. Training data for (a) a three-speed fan, (b) an LCD, and (c) a desk lamp. We manually cycled the devices through all possible operating modes.

ment with many more devices. This will allow us to validate the basic assumptions we've explored here.

Our long-term goal is to develop a complete household energy management system that integrates measurement and control functions. For meaningful control, the obvious requirement is to integrate an AC-line switch, but we believe controlling every device isn't practical or cost-effective. For example, the current Energy Star criteria for televisions require that they draw no more than 1 W in sleep mode.[6] So, actively disabling an Energy Star-compliant TV would yield only modest energy savings, unlike with older TVs, the most egregious of which can draw 10 to 20 W in off mode. We believe that an energy management system could use circuit-level NILM to identify those devices that should be controlled to maximize efficiency while minimizing cost. In addition, if the AC-line switch also included energy monitoring, we could use energy-monitoring data to effectively remove those devices from the unknown aggregate measurements, increasing the accuracy for the remaining devices.

Numerous studies have shown that household energy consumption can be lowered by simply providing real-time

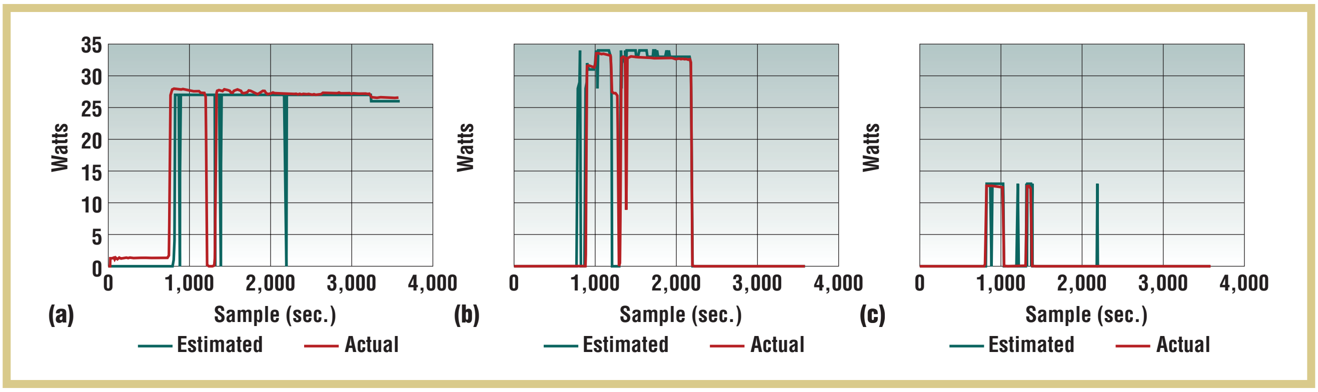
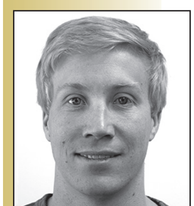


Figure 8. Estimated and actual power for the (a) LCD, (b) fan, and (c) lamp. We were able to detect device state changes from the circuit-level aggregate measurements.

aggregate energy use information.[7] We believe that device-level energy information will provide useful information, letting occupants identify wasteful devices and further reduce their consumption. Our research represents a small step toward the overall long-term goal of automated energy saving, in which buildings themselves detect wasteful or unnecessary devices and then disable them on the basis of the sensed occupant behavior.

the AUTHORS

Alan Marchiori is a PhD candidate in computer science at the Colorado School of Mines. His research interests include cyberphysical systems, ubiquitous computing, distributed systems, and wireless sensor networks. Marchiori has an MS in electrical and computer engineering from Purdue University. He is a member of IEEE and the ACM. Contact him at amarchio@mines.edu.



Douglas Hakkarinen is a PhD student in computer science at the Colorado School of Mines. His research focuses on using wireless sensor networks and machine learning to optimize environmental remediation techniques. Hakkarinen has an MS in computer science from the Colorado School of Mines. He is a student member of IEEE. Contact him at dhakkari@mines.edu.

Qi Han is an assistant professor in the Department of Mathematical and Computer Sciences at the Colorado School of Mines. Her research interests include distributed systems, middleware, mobile and pervasive computing, dynamic data management, and cyberphysical systems. Han has a PhD in computer science from the University of California, Irvine. She is a member of IEEE and the ACM. Contact her at qhan@mines.edu.

Lieko Earle is a researcher in the Electricity, Resources, and Building Systems Integration Center at the National Renewable Energy Laboratory in Golden, Colorado. Her research centers around laboratory- and field-based investigations of automated home energy management systems and advanced monitoring and load-disaggregation techniques. Earle has a PhD in physics from the University of Colorado at Boulder. She is a member of IEEE and the American Physical Society. Contact her at lieko.earle@nrel.gov.

REFERENCES

1. G. Hart, "Nonintrusive Appliance Load Monitoring," *Proc. IEEE*, vol. 80, no. 12, 1992, pp. 1870–1891.
2. Y. Agarwal, T. Weng, and R.K. Gupta, "The Energy Dashboard: Improving the Visibility of Energy Consumption at a Campus-Wide Scale," *Proc. 1st ACM Workshop Embedded Sensing Systems for Energy-Efficiency in Buildings* (BuildSys 09), ACM Press, 2009, pp. 55–60.
3. X. Jiang et al., "Experiences with a High-Fidelity Wireless Building Energy Auditing Network," *Proc. 7th ACM Conf. Embedded Networked Sensor Systems* (SenSys 09), ACM Press, 2009, pp. 113–126.
4. M. Kazandjieva et al., "PowerNet: A Magnifying Glass for Computing System Energy," *Proc. Stanford Energy & Feedback Workshop: End-Use Energy Reductions through Monitoring, Feedback, and Behavior Modification*, 2008.
5. A. Marchiori and Q. Han, "Using Circuit-Level Power Measurements in Household Energy Management Systems," *Proc. 1st ACM Workshop Embedded Sensing Systems for Energy-Efficiency in Buildings*, (BuildSys 09), ACM Press, 2009, pp. 7–12.
6. "TVs Key Energy Star Product Criteria," US Environmental Protection Agency, 2010; www.energystar.gov/index.cfm?c=tv_vcr.pr_crit_tv_vcr.
7. A. Faruqui, S. Sergici, and A. Sharif, "The Impact of Informational Feedback on Energy Consumption: A Survey of the Experimental Evidence," Brattle Group, 2009.