Algorithm development for Non-Intrusive Load Monitoring for Verification and Diagnostics

Christopher E. Reeg, Thomas J. Overbye

***Abstract*--Non-intrusive load monitoring (NILM) is a method of detecting the current energy consumption of a building using a single set of sensors on the main building supply. This is in contrast to intrusive monitoring where end use devices are sensed. Building on techniques of previous works, it will be shown that NILM can be implemented on currently available commercial devices and provide meaningful feedback both to the user and supplying utility. Limitations of inexpensive commercial devices such as resolution and measurement accuracy are a challenge. An algorithm currently under development, specifically developed for the low resolution of inexpensive commercial sensor packages, will be discussed and demonstrated. This information can then be used to verify the effectiveness of various smart grid initiatives in addition to other energy saving protocols such as weather proofing and energy efficient appliances. In addition, this information can be used to select devices for distributed control by analyzing not only the type of device but also the real world operating characteristics.**

***Index Terms*— load management, pattern matching, power system measurements, power system monitoring**

I. Introduction

NON-INTRUSTIVE load monitoring (NILM) has been explored over the past few decades as an alternative to invasive end use sensing techniques. Instead of placing several usage sensors on the consuming devices themselves, a single sensor set is placed on the electrical service entrance of a building. This has been performed successfully using laboratory equipment with a relatively high degree of accuracy [1] [2] [3].

The standard NILM configuration includes a sensor set to measure current and voltage and a processing algorithm for determining the status of various devices. Data is usually collected at or above line frequency in order to detect transient events or for performing harmonic analysis on current waveforms [4]. Algorithms are then used to isolate transients in the data to detect a device turning on, off or otherwise changing state. Data can then be collected by analyzing the transient events such as the size of the event, frequency and the time relation to other events. One of the primary means of analyzing collected data is with clustering algorithms. A preliminary application of such an algorithm will be detailed later.

This material is based upon work supported by the Department of Energy under Award Number DE-OE0000097.

Christopher E. Reeg is a second year masters student at the University of Illinois, Urbana, IL 61801 USA (email: creeg2@illinois.edu)

Thomas J. Overbye is the Fox Family Professor of Electrical and Computer Engineering at University of Illinois, Urbana, IL 61801 USA (email: overbye@illinois.edu)

There are three main types of loads that are seen in the load profile of a home. The first is a simple on/off load. This is kind of behavior is shown by lights, resistance heaters and uncontrolled motors (ie: refrigerator compressor motors). The second type of load is a load that has several states. This behavior is exhibited by dishwashers which may have several motors and heaters which become active in several different combinations throughout a cycle. A simpler example is a three way light bulb which transitions from low to medium to high and then off again. The last type of load is a variable load. This type is becoming more common with the adoption of variable frequency drives; specifically on compressor and fan motors.

Two state loads are relatively easy to detect as usually the increase in load when the devices is turned on is matched by an equal decrease when it is turned off. Three or more state loads are slightly more difficult to detect but still present edges in the measurement signal. By using the time relation between these edges and not just the magnitude, multi-state loads can be characterized. Variable frequency drives (VFD) offer a particular challenge to NILM as a result of continually changing the load over a relatively long time period. As suggested in [5], VFDs may be detected by using harmonic analysis. In the application of NILM in the typical domestic environment, it is not expected that many of the loads would be VFD controlled; therefore, one or two end use sensors on these drives may be justified in these cases. Modern power supplies present additional challenges by adding noise into the measurement. For instance, a desktop computer may have an average of 300 W but fluctuate more than 20 W in a very fast time frame. Filtering in the detection algorithm is necessary to prevent these small disturbances from crowding the clustering algorithm.

By analyzing the behavior of devices using NILM, data on the effectiveness of smart grid initiatives such as controlled load shedding and intelligent appliances can be evaluated on a very granular level. In addition, the impact of adding high efficiency windows, additional insulation and weather sealing can be viewed in terms of real performance improvements, such as the decrease in the duty cycle of an HVAC unit. This type of analysis also applies to evaluating energy saving appliances in real world applications to determine if laboratory efficiency ratings hold up in actual usage situations.

This type of monitoring is particularly relevant in identifying which devices in a system are available to be controlled by a demand side load management scheme. Without this type of monitoring, a survey of each connected device would be necessary to determine which would be the best candidates for distributed control implementations. With NILM, a quick survey of the system can be completed which

will identify various devices available for control by analyzing the specific load characteristics (size, timing, duty cycle). The end goal of this technology is to reduce the total electric load by micromanaging devices indentified by NILM as well as lower individual electric costs to the consumer. By knowing not only the type of device connected to the system but also the behavior of the device in day-to-day operation, an optimal selection of devices to be controlled can be made.

II. Hardware Overview

Recently, inexpensive commercial devices have been developed using off-the-shelf parts to create a sensor suite which can be used to measure electrical usage in the home. This measurement setup usually includes current clamps and voltage probes to measure the A and B phases in the typical North American domestic power connection. These measurements are then multiplied to obtain real and apparent power readouts. This package is commonly available for between $200 and $300 USD.

In this particular application, the device sampled the real and apparent power every one second. Unfortunately, only real power was archived in the device's internal memory so a computer was setup to poll the device every second to record the real time usage data outside of the device. This data was stored in a database to be later processed by test algorithms. In a production application, the data analysis algorithm will be able to process incoming data in near real time to determine the current load characteristics.

The hardware used had a maximum resolution of one watt but this often introduced noise into the signal which later had to be processed out. In practice, no device of interest produced a meaningful signal on the order of less than 25 W. This device also showed a tendency to have an offset in the power measurement signals. Since feature detection in this implementation was based on signal differential rather than absolute value, this did not have an effect on the quality of device detection.

These installations were setup in three different homes to collect data in varied environments. This included a small apartment installation, a 200 A service installation and a 400 A service installation. A wide range of appliances, lighting and heating was encountered.

III. Data Recording

Three weeks worth of usage data was collected for design and testing of feature recognition algorithms. One problem observed was the detection of events that occur in a small time period. The physical measuring device takes several measurements per second and then averages the result to be recorded. Therefore, if a load turns on at some point in the middle of the 1 second sampling time, the actual load on the system may not be fully realized until the next result is returned, in which the load was on for the entire sampling time. This causes the false perception that a load is taking several seconds to come up to full power when in fact it turns on in only a few cycles.

In one of the test environments, an unknown load was causing a large power spike for three seconds every 60 to 90 seconds in a periodic manner. This load was previously unknown to the homeowner but with the help of this data, the energy robbing source was discovered as an "instant on" coffee maker and is now being properly controlled. Figure 1 shows an example of this periodic load. Note that the exact same event was recorded a different way each time by the measurement device. This is the result of the long sampling period rather than the behavior of the device itself. While events in figure 1 suggest the device is taking one to two seconds to turn on, in fact the device is coming to full power in only a few cycles.

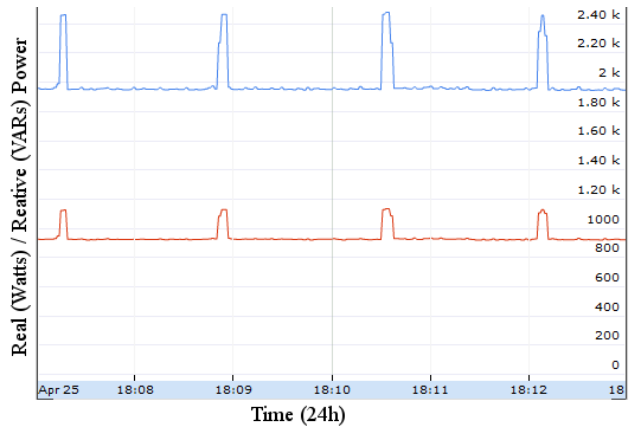

Fig. 1 – Six minute time sample of real (top trace) and reactive (bottom trace) power data. This specific signature is being generated by an always-on coffee maker. This is an example of the different sampling results of the same electrical event

When a sharp transient occurs in the system, this measurement device may show the transient lasting much long that it actually does. Due to the averaging effect, the magnitude of the transient will also be distorted. Figure 2 shows an induction motor startup. Note the spike in real and reactive power when the motor first turns on. While this transient only lasts a few cycles as the motor comes up to speed, the sample rate of the measurement device shows the effect for a whole second.

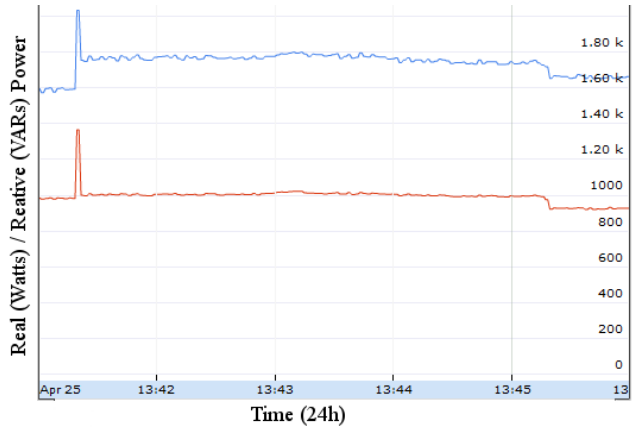

Fig. 2 – Example of detection of an induction motor startup. Real (top trace) and reactive (bottom trace) power spikes sharply when the motor first turns on and then reduces to a nominal level two samples (two seconds) later. The relatively long length of this transient is due to the low sampling frequency.

Electronic noise was also detected by the measurement device. This noise was the result of electronic loads which have rapidly changing demand based on the power usage of digital processors and other electronics. This effect on the measurement can be seen in figure 3 where in the first half of the window, a personal computer is sitting idle (with a refrigerator cycling as the primary waveform). In the second half, the computer is being used. The effect of the processor becoming active is clearly seen overlaid on the base waveform as random, fast changing spikes.

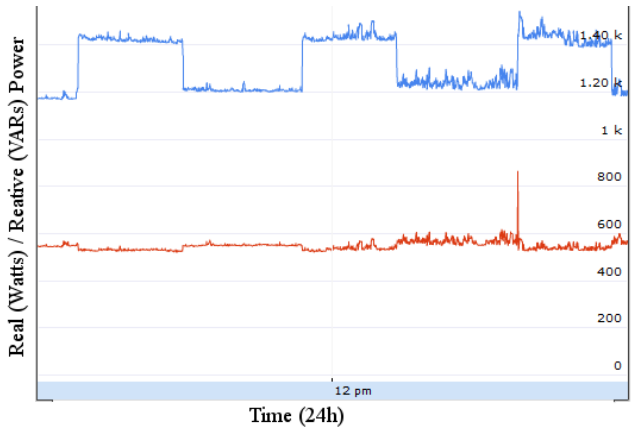

Fig. 3 – One hour time sample of real (top trace) and reactive (bottom trace) power signals generated by a refrigerator in addition to a personal computer which becomes active near 12:00pm. This shows the effect of digital electronic load noise on the base waveform

IV. Feature Detection Algorithm

In order to characterize the behavior of devices connected to the home electrical service, a feature detection algorithm was developed which could produce similar results even with non-identical input, as in figure 1. The algorithm also had to be immune to noise as in figure 3 but still be able to cleanly detect the magnitude and placement of edges in the signal.

To accomplish these objectives, a windowed derivative method was implemented. In this method, a "window" of samples (4 to 7 consecutive samples) is taken from the signal to be analyzed. A linear fit line of this window is then taken and recorded as the derivative value for the sample in the middle of the window. The window is then shifted to the next sample and the process repeated until all samples have been processed. This technique results in the edges in the original signal showing up as local maxima or minima in the recorded derivative signal.

This derivative signal is then processed so that anything below a certain threshold is ignored to remove noise and "uninteresting" features (typically features below 30 Watts), resulting in a "tresholded derivative". The windowed derivative has an inherent low pass filter characteristic so this final step ensures that only features of interest are processed.

As the value of derivative signal was set to zero for all samples below a certain threshold, the signal is now examined to determine where the local maxima/minima begin and end. These points indicate where to examine the original, unprocessed signal to determine the power usage before and after an edge. For example, with positive edges, the original signal is sampled when the derivative first changes from zero to a positive value. This point is taken as the starting power value for the edge. The ending power value is taken when the derivative returns to zero, indicating the end of the event. The difference between the beginning and ending power value determines the magnitude of the event while the timestamp of the local maxima/minima determines the placement of the event.

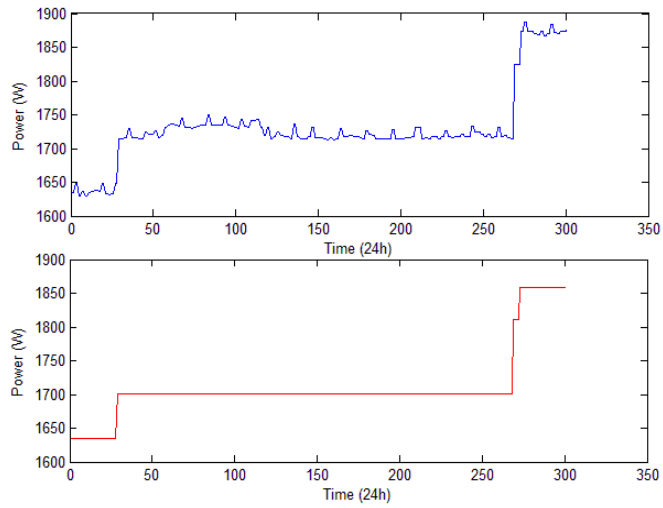

Fig 4. – Results of windowed derivative algorithm. Top trace shows the real power sampled from a house. The bottom trace shows the reconstructed real power signal based on the windowed derivative feature detection method.

Figure 4 shows an example of the results of this windowed derivative algorithm. Using information about the edges detected in the top trace, the original signal was recreated with a high degree of accuracy. By comparing the resulting reconstruction, it can be determined if the level of granularity is sufficient for gathering accurate system information. Note in this case that a transient from 60 to 120 seconds was not detected as it was below the noise threshold. This event did not significantly affect the accuracy of the reconstruction so it can be concluded that the threshold level and window size have been set correctly.

The ability of the algorithm to produce similar results under varied input is demonstrated in figure 5. In this case, varied input data, similar to that in figure 1, is transformed into nearly identical electrical events. The noise is filtered out and the edges of the events easily compared to one another. Processing the signal edges to be similar to each other will increase the probability that differently measured events will be assigned to the same electrical device. Without this processing, events actually coming from the same device may be characterized as coming from separate devices, which decreases the accuracy of the system model.

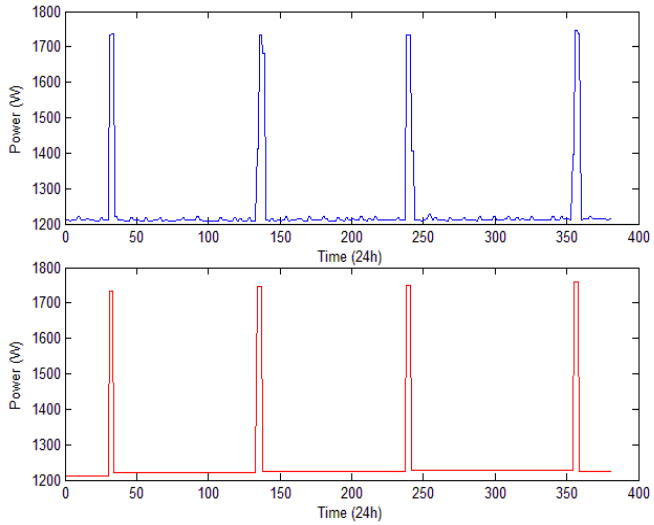

Fig 5 – Windowed derivative algorithm applied to a six minute series of equal electrical events. Events were each sampled by the measurement hardware (top trace) in a slightly different manner each time but produced reconstructed signals which are near exact copies of each other (bottom trace).

This algorithm still has room for improvement as it does not handle the case where positive and negative edges are very close to each other as the low pass filter characteristic of the algorithm cancels out the two events. While implemented on static data collected in a database, there is nothing fundamental preventing this algorithm from being implemented in real time to detect events as they happen on the system. This algorithm requires minimal computing resources as the calculation need only be performed once for every incoming sample (in this case once a second). A small embedded microprocessor with floating point capability could easily handle this task.

V. Feature Processing

After features have been collected, a classification process is used to group features together into a device profile. This operation can be performed by many different methods but for this particular application, the k-means clustering algorithm was used [6].

Only the real (P) and reactive (Q) power features are used for cluster analysis as the voltage events were often uncorrelated to devices turning on or off. The k-means clustering method categorizes points by their distance to *k* user specified centroids. Groupings are determined by first randomly placing *k* centroids in the sample space. Data points are then assigned to the closest centroid. The mean of all the samples belonging to a centroid is then calculated and the location of the centroid is then moved to the mean point. The process is then repeated: Data points are assigned to the closest centroid and then a new mean is calculated. This process is iterated until there is no change in the members of every specified grouping.

In this specific case, real and reactive power points were categorized into 15 groups. The number of groups was chosen to minimize the total distance of the samples to the centroids while maintaining distinct groupings. Figure 6 shows the result of applying this clustering algorithm to an hour of data collected during a particularly active time of day.

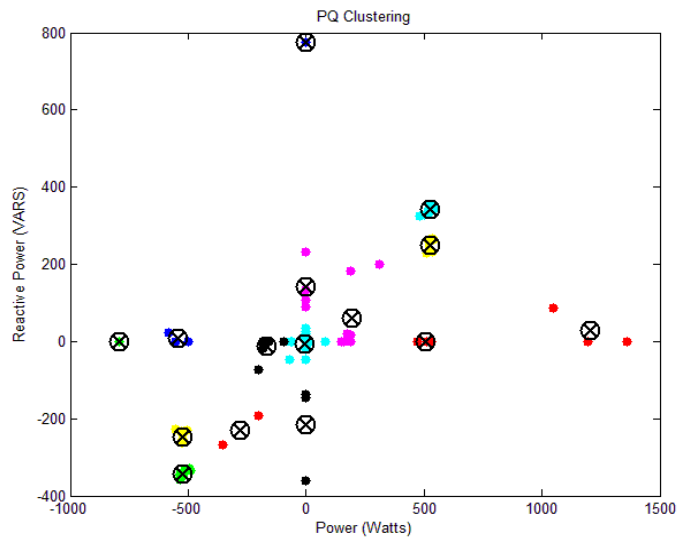

Fig. 6 – Results of k-means clustering on an hour long sample window. The centroid of each cluster is marked by an X. Note that a few of the clusters do not represent particular devices of interest but several of the centroids do accurately represent the turn on and turn off characteristics of a specific device. (Colors are repeated.)

As can be seen in figure 6, there are no events in the in quadrants II or IV, this is expected as there were no capacitive loads on the system (there were no devices that produced reactive power when consuming real power). Note that there is also a general symmetry line mirroring quadrant I and III. This is expected as the turn-on characteristic of a device is expected to match the negative turn-off behavior. Improvements in the feature detection algorithm will help reduce the variance of these clusters and produce a more accurate system model.

The k-means process does have a few downsides. For one, the process is not repeatable. While each result may not necessarily be unique, there may be several results which are valid under the specified ending criteria. This algorithm may also result in empty clusters; the initial seeding of centroids may result in some centroids which are not assigned any of the initial samples. This results in dead clusters that cannot move and grow. Therefore, the algorithm will not complete and a new seeding has to be attempted. In practice, the k-means algorithm is often repeated several times and the result with the minimum sum of the distance from the centroids to their associated samples is chosen as the clustering result.

The results of this clustering process are used to classify newly detected P,Q changes from the feature detection algorithm. After the initial clustering has been completed, any new sample is classified by finding the closest centroid. Unfortunately, this requires evaluating the distance to each centroid; therefore, the more devices which are desired to be sensed, the longer this process will take (O(n) operation). In the home environment, this is not expected to impede the detection process as these calculations can be made very quickly and the number of devices is not expected to be very large.

The initial clustering will require manual intervention on the part of the user for "training". This process helps determines which of the identified clusters belong to a particular device and which are simply noise. A mathematical

evaluation of this is to classify the "usefulness" of a cluster by maximizing the number of samples belonging to a cluster while minimizing the variance of these samples.

VI. Conclusions

This combination of windowed derivative and k-means clustering algorithms was able to successfully identify which and classify load events that were happening on the system. Currently this method is implemented using Matlab on static data held in a database. In the future, the detected cluster pattern can be uploaded to a device that detects events in real time.

The k-means clustering algorithm, while effective does have a few downsides that could be rectified by using a different clustering method. While not currently implemented, quality threshold [7] or density based clustering may be a better algorithm as results have greater reproducibility and handle the variance in measurement better than the k-means method.

The next step in the investigation of devices would be to analyze the state and time dependence between events. Currently, this algorithm does not handle complex cycle devices such as dish washing or clothes washing machines. The load in these devices changes in steps throughout the operational cycle. This type of device could be detected by observing the time interdependence of the P,Q steps and recording the estimated state of the device. Collecting time based data such as duty cycle and time of use information for these devices will allow devices to be optimally chosen for demand side load control.

The work presented here details the preliminary steps to verify smart grid initiatives and energy saving methods. From this implementation, further work on integrating analysis with real time measurements can progress in addition to increasing the consistency and accuracy of detections.

VII. References

[1] L.K. Norford, and S.B. Leeb, "Non-intrusive electrical load monitoring in commercial buildings based on steady-state and transient load-detection algorithms." Energy and Buildings, vol. 24, Cambridge: Elsevier Science S.A., pp. 51-64, 1995.

[2] G.W. Hart, "Nonintrusive Appliance Load Monitoring," Proceedings of the IEEE, Vol. 80, 12, pp. 1870-1891, December 1992.

[3] S.R. Shaw, C.B. Abler, R.F. Lepard, D. Luo, S.B. Leeb and L.K. Norford, "Instrumentation for High Performance Nonintrusive Electrical Load Monitoring," Journal of Solar Energy Engineering, Vol. 120, Cambridge : American Society of Mechanical Engineers, pp 224-229, 1998.

[4] S.B. Leeb, S.R. Shaw, and J.L. Kirley, "Transient Event Detection in Spectral Envelope Estimates for Nonintrusive Load Monitoring," IEEE Transactions on Power Delivery, Vol. 10, Cambridge, pp. 1200-1211, July 1995.

[5] C. Laughman, K. Lee., R. Cox., S. Shaw , S. Leeb, L. Norford, and P. Armstrong, "Power Signature Analysis," IEEE Power & Energy Magazine, pp. 56-63, March/April 2003.

[6] J. MacQueen, "Some methods for classification and analysis of multivariate observations", Proc. Fifth Berkeley Symp. on Math, Statistics and Probability, Vol. 1, Los Angeles: University of California Press, pp. 281-297, 1967.

[7] L. Heyer, S. Kruglyak, and S. Yooseph, "Exploring Expression Data: Identification and Analysis of Coexpressed Genes," Genome Research, vol. 9, Los Angeles: Cold Spring Harbor Laboratory Press, pp 1106-1115, 1999.