A Multiprocessor for Transient Event Detection

Umair A. Khan Steven B. Leeb, Member, IEEE Mark C. Lee

Laboratory for Electromagnetic and Electronic Systems
Massachusetts Institute of Technology
Cambridge, MA 02139, USA

Abstract: *The multiscale transient event detector (TED) introduced in [1] uses measurements of current and voltage made at the utility service entry of a building to help determine the operating schedule of all of the significant loads in the building. This paper describes a TED algorithm implemented on a custom multiprocessing machine with several inexpensive processors performing tasks in parallel. A parallel version of the serial TED algorithm is developed, and the hardware design for a multiprocessing TED is presented. The performance of the algorithm is illustrated with results from a prototype.*

I. Monitoring Nonintrusively

The Nonintrusive Load Monitor (NILM) determines the operating schedules of the individual loads at a target site strictly from measurements made at the electric utility service entry [2], [1]. Applications of this technology have provided utilities with inexpensive, highly detailed load usage surveys. Nonintrusive monitoring provides easier installation, simpler data collection, and simpler data analysis than many conventional monitoring schemes that involve separate sub-metering of every load of interest. In [1] and [3], load turn-on transients were used to develop a transient event detector (TED) for a NILM. With the incorporation of the transient event detector, the applicability of the NILM is expanded to commercial and industrial environments where significant efforts, e.g., power factor correction and load balancing, are made to homogenize the steady-state behavior of different loads.

Research is under way to extend the capabilities of the NILM for power quality monitoring and diagnostic monitoring of critical loads, such as HVAC components, at residential, commercial, and industrial facilities. For example, with the TED, the NILM can track down power quality offenders, i.e., loads which draw extremely distorted, non-sinusoidal input current waveforms, by correlating the introduction of undesired harmonics with the operation of specific loads at a target site. Also, since the NILM can associate observed pieces of a waveform created by an aggregate of loads with a particular load of interest, we suspect it is possible to use state and model-based parameter estimation algorithms to extract model parameters for diagnostic "trending" of important loads.

The basis for the TED is the observation that the transient behavior of most important load classes is distinctive. This allows reliable recognition of individual loads from observed transients. However, direct examination of a current waveform at the service entry, or a closely related waveform such as instantaneous power, may not satisfactorily reveal key features for transient identification. A preprocessor, described in [4], in the prototype event detector eliminates carrier frequency artifacts from input data by averaging over at least one carrier wave period to generate a short time estimate of spectral content which we call a *spectral envelope*.

For example, Figure 1 shows the turn-on transient or *event* of a rapid start fluorescent lamp bank on a single phase electrical service. The figure shows the spectral envelope of fundamental frequency current in phase with the voltage waveform. This spectral envelope corresponds, in steady state, to a measure of real power. The waveform has been segmented into regions of significant variation, or *v-sections*, marked by ellipses A–C. The TED makes a complete transient identification by searching for a precise time pattern of v-sections corresponding to a particular load. As long as each of the v-section shapes overlaps with no more than a near-constant region (e.g., regions like those in between v-sections in Figure 1), the event detector will be able to identify a pattern of v-sections and therefore the transient. Since some degree of overlap is tolerable, the v-section set recognition technique will generally operate successfully in an environment with a higher rate of event generation than

96 SM 520-7 PWRD A paper recommended and approved by the IEEE Power System Instrumentation & Measurements Committee of the IEEE Power Engineering Society for presentation at the 1996 IEEE/PES Summer Meeting, July 28 - August 1, 1996, in Denver, Colorado. Manuscript submitted January 2, 1996; made available for printing May 2, 1996.

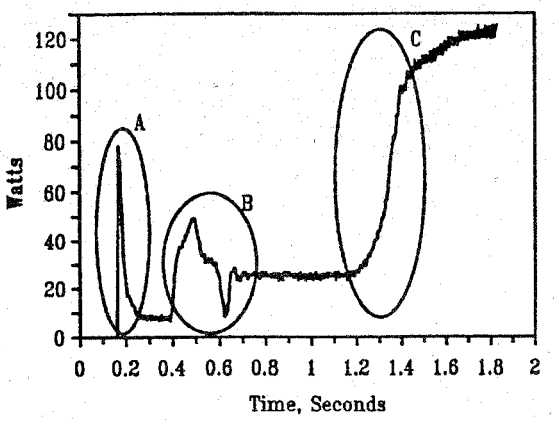

Figure 1: Rapid Start Lamp Bank Transient in Real Power

would a detector searching for whole, undisturbed transient shapes [3].

Experimental results from the prototype event detector presented in [3] indicate that the detection algorithm is capable of successfully determining the operating schedule of different loads, even in environments with relatively high rates of event generation. The detection algorithm as implemented in this prototype is essentially serial. A single digital signal processor searches input spectral envelopes for v-sections of every load of interest. Searching for more loads in real-time places greater demands on available processing power, as the processor must complete its search before the next window of sample points is available in order to avoid missing events of interest.

Fortunately, substantial components of the detection algorithm could be executed in parallel. Taken individually or in small groups, these operations are relatively undemanding in terms of required computational effort. This paper describes the development and testing of a parallel or multiprocessing algorithm and platform, the multiprocessing load monitor (MLM), for multiscale transient event detection. In the MLM prototype, an array of 16 inexpensive processors replaces the single digital signal processor employed in [3], providing tremendously improved recognition capacity. Experimental results from the prototype are presented.

II. MLM Algorithm

The "serial" transient event detection algorithm introduced in [1] is shown in Figure 2. We will see that this algorithm exhibits a high degree of potentially parallel execution. The next section reviews the TED algorithm used in [1]. The following section examines this algorithm for data dependencies and develops a parallel implementation, the MLM algorithm.

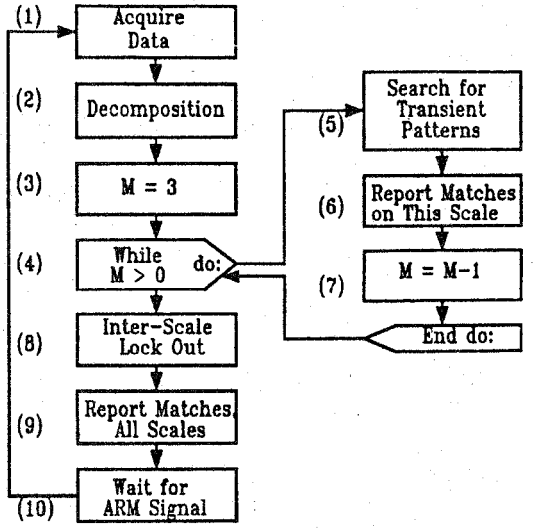

Figure 2: Transient Event Detection Algorithm

Transient Event Detection

The v-sections of interesting loads are collected and stored as templates before the TED algorithm in Figure 2 is activated. Once armed, the TED samples the spectral envelopes computed by the spectral envelope preprocessor, searching for known transient patterns. Within a particular class of loads, e.g., induction motors, a small number of template shapes, appropriately scaled in amplitude and duration, may serve to represent all of the loads in the class over a certain power range. For this reason, event detection occurs over several time scales. For instance, the input envelope may be downsampled by factors of two and analyzed for v-sections. This multiple time scale search procedure minimizes template storage and processing time. A tree-structured decomposition, performed at Step 2 in Figure 2, alters the time scale of the spectral envelopes for as many time scales as are of interest. Steps 3 and 4 then set up a loop to perform a v-section pattern search on each time scale. In [1], transversal filters detect v-sections in the input envelopes (Step 5).

If all the v-sections in a complex v-section set of a certain load are found, it is very likely that the transient associated with the load is present in the spectral envelopes. An intra-scale v-section lock out is performed in Step 5 to minimize the possibility of false positive identifications. If a complex transient is detected, the locations of its v-sections are noted as "locked out". A simpler transient will not be identified if its v-sections are detected at locked out locations.

Once all v-sections for a load on a time scale are found, a report is made of all events detected on that time scale

(Step 6). The TED then repeats event detection and lock out on the other time scales (Step 7). When all the patterns on all the scales have been searched, all events detected are further scrutinized by inter-scale v-section lock out (Step 8). This guarantees that v-sections from a complex pattern on a coarse time scale were not used to match simpler v-sections on a finer time scale. Finally, a report of all loads identified is made in Step 9.

Parallel Algorithm

The TED monitors load activity by searching for transients in several data signals (e.g., real and reactive power) over several time scales. Transient event detection consists of the following sequence of operations:

1. Sample data from each spectral envelope.
2. Scale each envelope in time, to get new envelopes for each original input envelope.
3. For each data envelope (on each time scale), search for v-sections of interest.
4. Collate results and perform intra-scale v-section lockout on results for each time scale.
5. Perform inter-scale v-section lockout on results from all time scales.

Many of these operations are not constrained to precede or follow one another. As a first step in bringing out the parallelism in the transient event detection algorithm, we must determine the data dependencies for the functions performed in the algorithm.

Figure 3 shows the data-dependency graph for the TED operations, for a sample case examining at least two time scales. The following data dependencies are present in the TED algorithm: For each envelope independently, data acquisition must precede decomposition and the search for v-sections. For each new time scale independently, time scale decomposition must precede v-section recognition, which in turn must precede intra-scale lockout. Finally, over all time scales and all envelopes, intra-scale lockouts precede the inter-scale lockout.

The data-dependency graph brings out the following *independencies*: The operations of data acquisition, time scale decomposition, and v-section recognition can be performed on each spectral envelope independently from the performance of these operations on other envelopes. For a given time scale, all result collation and intra-scale lockouts may be performed independently from these operations on other time scales. For a given spectral envelope, all time scale alteration or decomposition operations can be performed independently from one another,

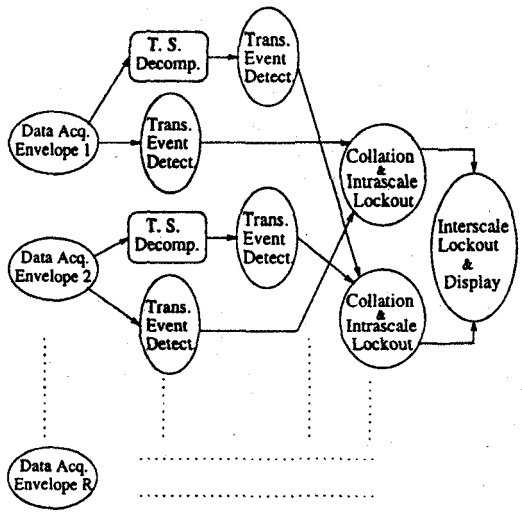

Figure 3: Data-Dependency Graph for TED Operations

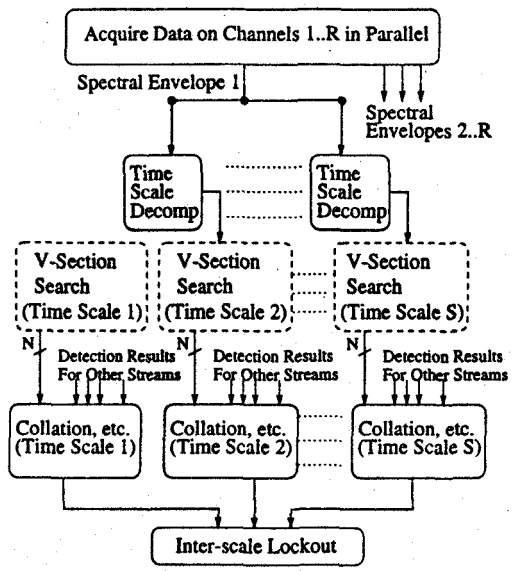

Figure 4: Parallel TED Algorithm

and all operations (v-section search, collation, and intra-scale lockout) on different time scales may be performed independently from one another.

Using these data dependencies and independencies we can organize a parallel transient event detection algorithm. One possible implementation is shown in Figure 4. The main steps of this algorithm are listed below:

1. Data acquisition is performed on all input spectral envelopes in parallel.
2. The following operations are performed in parallel:
   (a) Every envelope is searched in parallel for each load's transients on the original time scale. In

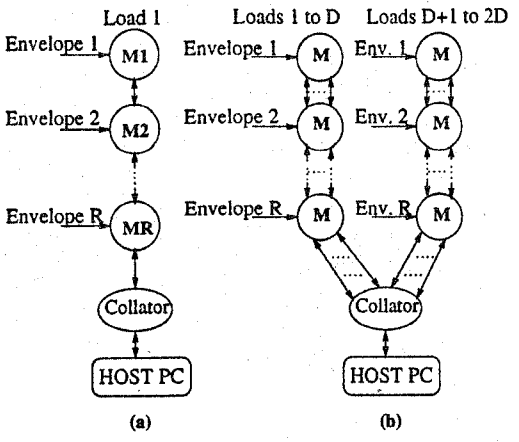

Figure 5: Identifying (a) One Load (b) Several Loads

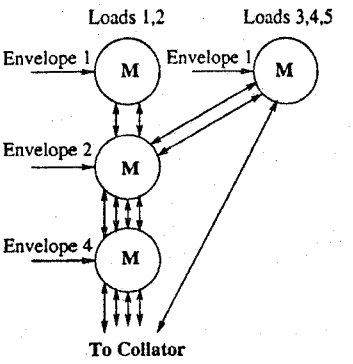

Figure 6: Compact Load Detection

addition, the MLM could search for the v-sections of a given load on a given envelope in parallel.

(b) Time scale decomposition is performed on all the envelopes in parallel.

3. V-section search on all the time-scaled envelopes is performed in parallel in the same manner as for the original envelopes.

4. Result collation (including intra-scale lockout) is performed on each time scale in parallel.

5. Inter-scale lockout is performed on the results of the collators.

In the prototype, we choose to search for all v-sections of a load on one envelope at one time scale sequentially on a single processor. The v-section patterns for multiple loads on this time scale and envelope may also be assigned to this processor, up to the limit of the available computation capability. This choice was not motivated by precedence constraints. In the prototype, this choice provided a reasonable trade-off between hardware (interconnect) complexity and computational power available per individual processing unit. For other hardware architectures, different allocations of computational effort might be more appropriate.

Figure 5(a) shows a configuration of computation modules appropriate for identifying a single load on one time scale with transient v-sections in a total of R spectral envelopes. Processor module M1 searches Envelope 1 for the v-sections from the load that could be present in that envelope. Upon detecting all v-sections, it signals processor M2, which has been searching for v-sections in its own input envelope. The detection results are passed down this linear processor chain, or *load chain*. When all modules, including the final processor MR, have identified their v-sections, the load turn-on event has been identified on the given time scale. A notification message is passed on to a collator processor for that time scale.

Figure 5(b) shows a possible, but not necessarily efficient, processing module configuration for identifying multiple loads on a particular time scale. Each processor searches for v-sections of D loads on their respective envelopes, and passes detection results to the next processor in the load chain. One reason why this arrangement is not necessarily efficient is the fact most loads will *not* have v-sections in all envelopes, e.g., an incandescent lamp might exhibit interesting behavior only in the real power envelope, while an induction motor might exhibit interesting v-sections in both the real and reactive power envelopes. Also, loads will not generally have the same number of v-sections or similarly sized v-sections in different envelopes. Thus, a more compact configuration is employed in the prototype.

Figure 6 shows a "custom" distribution of loads and v-sections across processing modules. Loads 1 and 2 have a large number of v-sections in Envelope 1 compared to Loads 3, 4, and 5. Hence, one processor is dedicated to v-sections of Loads 1 and 2 in Envelope 1. Another processor searches Envelope 1 for the v-sections of Loads 3, 4, and 5. Loads 1 through 4 also have v-sections in Envelopes 2 and 4. Load 5 has no v-sections in any envelope other than Envelope 1. All five loads can be identified with the processor load chains shown in Figure 6. In practice, we have found it convenient to preconfigure the prototype with a collection of "custom" load chains commonly exhibited by many loads. For example, the prototype contains, among others, processor chains for identifying loads with v-sections in envelopes of real power only (e.g., some heating and lighting loads), real and reactive power (e.g., induction motors), and real power and third harmonic current content (e.g., personal computers). Careful *a priori* selection of load chains has allowed us to avoid reconfiguring the prototype during field testing.

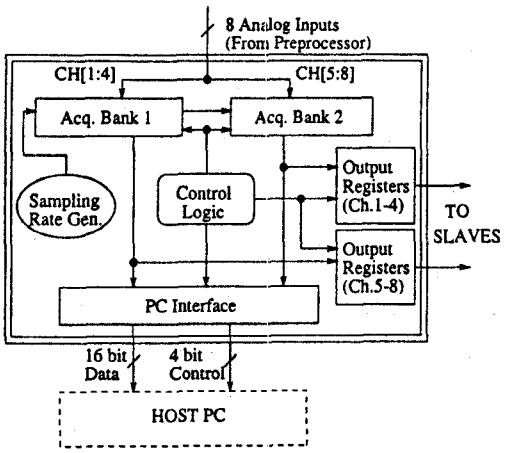

Figure 7: MLM Master Board

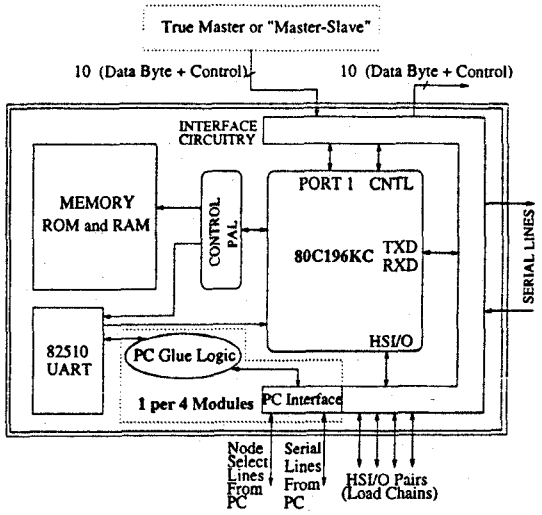

Figure 8: MLM Slave Module

III. Prototype Hardware

The MLM prototype consists of two subsystems: a data acquisition front-end and an array of computational units. The data acquisition front-end includes an analog spectral envelope preprocessor and a *master board* to sample the incoming spectral envelopes. It buffers the digitized data and periodically ships data blocks to the array of processing modules. Each processing module, or *slave module*, performs one of the three TED functions: v-section search, time scale decomposition, or result collation.

Data Acquisition Front-End

The analog preprocessor is the interface between the digital world of the MLM and the electric utility service entry [4]. The preprocessor computes the envelopes of real (P) and reactive (Q) power, as well as in-phase and quadrature third harmonic (3P, 3Q) contents of current. It also calculates the envelopes of in-phase and quadrature higher current harmonics (e.g., 5P, 5Q, 6P, etc.), providing a total of 16 analog channels for each utility phase. [1]

The master board accepts eight channels from the analog preprocessor. It digitizes and stores the incoming analog data at a selectable sample rate (200 Hz per channel in the prototype), periodically transfering blocks to the slave modules. Note that the spectral envelope decomposition effectively demodulates the envelope of each harmonic, up to 16th or higher, to near-DC [4]. Hence, each spectral envelope may be sampled at a relatively low rate in comparison to the actual harmonic frequency. In a very cost-sensitive implementation, the sample rate could drop as low as 80 Hz per channel for the analog preprocessor described in [4]. The master board employs an interleaved memory buffer to prevent data loss during transfers to the slave modules. It also separately stores large windows of data that, upon request, are transferred to the host PC.

Figure 7 shows the basic blocks of the master board. The eight analog input channels are sampled by two 4-channel, analog-to-digital converters that provide 12 bits per channel. Each converter transmits its conversion results to dedicated memory. A converter, associated memory, and address counters comprise an *acquisition bank*. A sampling rate generator and appropriate control logic coordinate the functions of the acquisition banks. The basic data path of the master board consists of an 8-bit data bus, a 12-bit address bus, and various control and status signals. After each analog-to-digital conversion, the control logic is informed that new sampled data is ready. The control logic then reads the data from the converters and writes it into memory. All microcontrol is implemented by finite state machines (see [5]) programmed into six Programmable Array Logic (PAL) ICs [8].

The acquired data must be periodically transferred to the slave processors. Transfer begins when the banks of memory on the master board are full. The master board issues an interrupt signal to all appropriate slave processors. Data is read sequentially from the acquisition banks, and transmitted to the slaves via output registers. The acquired data can also be shipped to the host PC.

Slave Module

The master board provides data to an array of slave

[1] Note that in-phase and quadrature harmonics are referred to with the same P and Q notation as "real power" and "reactive power". This nomenclature, adopted from [4], is simply a "shorthand" notation for describing in-phase and quadrature harmonics.

modules that operate in parallel and can communicate with one another as needed. Figure 8 shows a block-level diagram of a slave module based on the Intel 80C196KC microcontroller [6]. Both Read Only Memory (ROM) and Random Access Memory (RAM) are provided. The microcontrol of all data paths is performed by a control PAL. The controller's internal serial port is used for inter-slave communication. An additional, external serial port (Intel 82510) provides communication between each slave module and the host PC.

The host PC communicates over a serial link with the slave boards. It downloads program code into each slave's RAM along with v-section templates needed for transient event detection. It retrieves results of event detection and collation from the collators. Since an MLM will typically consist of several slave modules, hardware support is provided to allow the PC to select and communicate with individual processors. Each processor is assigned a unique 8-bit ID. The host PC sends the selected slave's ID byte via a PC I/O card to each slave board. Processor selection is achieved by special circuitry, called the "glue logic", on the slave boards. The glue logic compares the provided slave ID byte with switch settings to determine if its board is selected and, if so, which processor is selected. It then connects the desired slave processor's serial lines to the PC serial lines.

Each slave receives input data on its parallel port 1. An interrupt signal from the master goes to each slave's non-maskable interrupt line, signalling the availability of fresh sampled data. Slave processors searching for transient events on time scales different from that of the originally sampled data acquire input data from other slaves performing time scale alteration. The master interface circuitry and the data transfer protocol ensure that, for slaves searching for v-sections, it does not matter whether their "master" is the master board or another slave.

The high speed input and high speed output ports are used for inter-slave communication. In the prototype, inter-slave communication was accomplished, for simplicity, entirely through flags communicated on the high speed lines. In a prototype identifying more loads, serial inter-slave communication would become desirable.

IV. Slave Functions

In the MLM computational array, a slave module may serve to recognize v-sections, perform time scale decomposition, or collate detected events. On start-up, the host PC configures each slave module with appropriate software for the function it will perform.

V-section Recognition

The PC also loads v-section templates into the slave modules performing v-section searches. The slave modules could employ any of a variety of pattern discrimination techniques to search sampled spectral envelopes for known v-section patterns [7]. In the TED discussed in [3], for example, the prototype was implemented on a floating point digital signal processor, and transversal filters were used to search for v-sections. To minimize expense in the prototype, the processors in the slave modules provide hardware support for integer arithmetic only. The prototype employs a pattern discriminator that computes the l_1 norm of the error between an observed data vector i and a known template vector t:

$$e = \sum_{n=0}^{n=N-1} |i_n - t_n|$$

The observed data vector i is an N-point windowed segment of the data block provided by the master board. The template vector t is stored in memory for use by the event detector during a one-time training period when the detector is installed at the target site.

The computation of the distance measure e, the aggregate of the point-wise absolute difference between two N-vectors i and t, requires no multiply or divide operations and is, therefore, relatively computationally inexpensive. Both the v-section templates and the observed data vectors are "AC-coupled" before e is computed [3]. Removing short term DC levels in the input envelope is essential for detecting v-sections occuring on top of the quasi-static regions of other loads' transients or steady state operating characteristics. If the distance measure e between t and i is within a specified error threshold, the v-section t is presumed present in the input data.

Once a slave module identifies all of the v-sections in the precise time pattern expected for a particular load on the envelope under examination, it so informs the other processors in the load chain through the high speed lines. The processor at the head of this chain is designated the *first processor* and the one at the tail is the *last processor*. Each processor on the load chain is aware as to its being the first, last or an intermediate processor on the chain. A full transient turn-on event for a particular load is detected when all the v-sections for that load have been identified.

Communication follows a *Generate and Propagate* scheme. If the first processor identifies all of its v-sections for a given load, it generates an identification message for the next processor on the chain. This message is propagated by each intermediate processor upon detection of the v-section(s) for which it is searching, until the last processor receives this message. After the last processor also identifies its v-section(s), it compiles a record of all relevant information (the associated load, the event time, etc.) and serially transmits this information as an event detection notification to a collator.

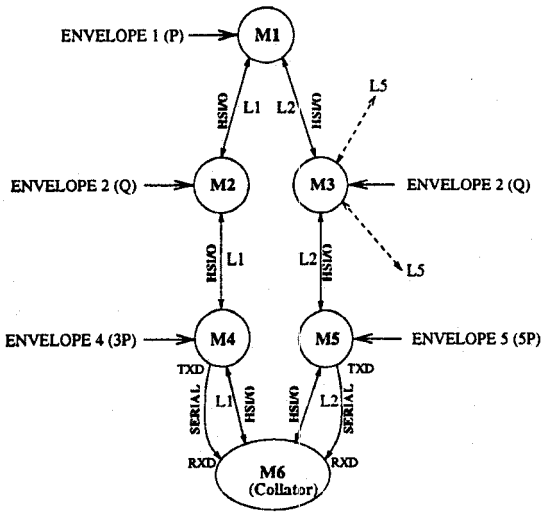

Figure 9: Load Chains in the MLM

As an example, consider Figure 9, displaying six processors M1, M2,..., M6. Modules M1 through M5 are configured for transient event detection. Processor M1 searches the spectral envelope corresponding to real power, P. Processors M2 and M3 examine the reactive power envelope, Q, M4 searches 3P, and M5 examines 5Q. Processor M6 is a collator for this time scale. In this example, the load chains are searching for two loads, L1 and L2. The load L1 exhibits one or more v-sections in each of the envelopes P, Q and 3P. Identification of the transient event associated with L1 will be conducted by M1, M2, and M4, i.e., the load chain for load L1 is:

$$M1 \rightarrow M2 \rightarrow M4$$

Load L2 has one or more v-sections in each of the envelopes P, Q and 5Q. The chain for load L2 is:

$$M1 \rightarrow M3 \rightarrow M5$$

Processor M2, an intermediate processor for L1, waits for an identification message from M1. When M1 detects its v-sections in P, it sends an identification message to M2. When M2 identifies its v-sections in Q, this message is propagated to M4. Module M4, the last processor in the chain, transmits a load identification record to the collator M6 if and only if it too detects all its v-sections in its input envelope, 3P. Note that any processor, e.g. M1 or M3 (which is part of the unshown load chain for load L5), may be part of more than one load chain.

Time Scale Alteration

For each required time scale in the prototype, one processor is dedicated to altering the sample rate of the input data. In the prototype, the scale alteration is conducted by first digitally low pass filtering and then decimating the data in time [1]. The resulting block is then transmitted to the slaves searching for v-sections on the coarser, down-sampled time scale, using the protocol of the regular master-to-slave transmission.

Result Collation

There is one collator per time scale. Each collator performs intra-scale lock-out on identified transient events and relays all geniune load identifications on its time scale to the host PC. The collator polls each "last" processor in the load chains on a round-robin basis to see if a load identification record awaits. If so, the collator grants the processor access to its serial port for identification record transmission.

The host PC communicates with the collators on a round-robin basis, checking periodically for positive identifications. The final steps (inter-scale lockout and result display/storage) in the multiscale event detection algorithm are assigned to the host PC, which has access to the event detection results on *all* of the time scales.

V. Prototype Performance

The MLM prototype contains a master board and four quad-slave boards, providing a total of 16 processors. The five boards are stacked in a rack-mount chassis installed in a cabinet that holds the host PC. Slave board 1 houses slave modules M0 through M3. These perform v-section detection on the original time scale on envelopes P, Q and 3P (two processors watch 3P). The second slave board provides M4 and M6, which also serve as v-section detectors, and M5 and M7, the collators for the two time scales implemented in the prototype. Slave board 3 holds slave modules M8 and M9, performing time scale alteration decomposition on P and Q respectively. Modules M10 and M11 are the event detectors on the coarse scale that receive data, downsampled by 4, from M8 and M9. The fourth slave board consists of slave modules M12 through M15, which are unused, "spare" modules in the prototype. The host PC retrieves the most recently acquired envelope data and transient identifications from the master board for display and possible analysis.

The following six loads' activities were monitored by the MLM (shorthand names are included in [ ]'s):

1/4 Hp Induction Motor [MOTOR]
Rapid Start Fluorescent Lamp Bank [RAPID]
Instant Start Fluorescent Lamp Bank [INSTANT]
Incandescent Light Bulbs [LIGHT]
Personal Computer [COMP]
1/3 Hp Induction Motor [BIG_MTR]

Load chains implemented in the prototype are listed below, with the collator for each chain shown at the end,

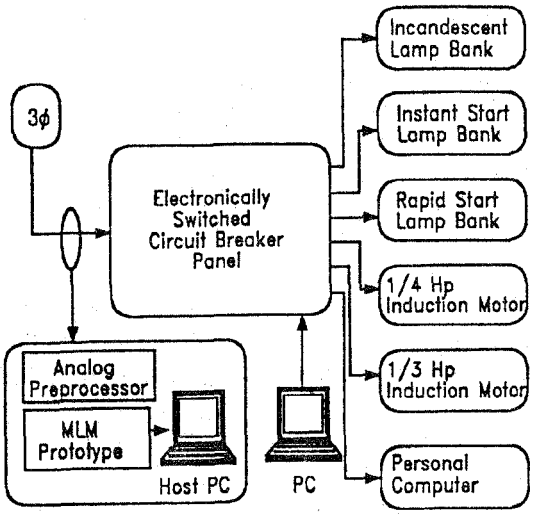

Figure 10: Prototype Test Facility

in [ ]'s:

INSTANT START: M0(P)→ M2(3P) [→ M5]
RAPID START: M0(P)→ M1(Q)→ M2(3P) [→ M5]
COMPUTER: M0(P)→ M3(3P) [→ M5]
SMALL MOTOR: M4(P)→ M6(Q) [→ M5]
INCANDESCENT LIGHT: M4(P) [→ M5]
BIG MOTOR: M10(P)→ M11(Q) [→ M7]

Additional loads may, of course, be identified.

The test stand used for experimentation is schematically illustrated in Figure 10 [1]. The prototype monitors the "service entry" to an electronically switched circuit breaker panel which provides the electrical hookup to the loads. A three phase, 208/120 volt electrical service powered the loads, which were chosen as representative of important load classes in commercial and industrial buildings. A dedicated computer controls the operation of each load connected to the circuit breaker panel. It can be programmed to simulate a variety of possible end-use scenarios. Template v-sections for each of the loads were captured during a one-time "walk-through" of the test stand. As in [3], the templates for the big motor were not collected from raw data. Instead, the v-sections for the small motor were simply scaled in time and amplitude, and used to identify the activity of the big motor. Because the big and small induction motors are members of the same load class, a single transient template, appropriately scaled in amplitude and duration, was expected to prove satisfactory for identifying both motors.

Figures 11 through 16 show screen prints from the host PC during seven tests conducted with the stand (many additional experiments with the MLM can be found in [8]). During normal operation, the prototype continuously monitors the service entry and reports any recognized transient events. Each screen print shows four plots of the envelopes of P in Watts, Q in VAr, and 2P and 3P harmonic contents of current. Loads identified as turning on are shown in the *MLM Console* window along with the time of the event. Event detection was carried out on two time scales: the original, fine scale (scale 1 in the figures), and a coarse scale (scale 2 in the figures) derived by downsampling the original data by 4.

Figures 11 and 12 show the prototype detecting the activity of some sample single load transients on the fine scale. Figure 13 shows the turn-on transient for the big motor on the coarse scale. Refer to these figures when studying the following results of monitoring multiple load activity.

In Figure 14, the MLM has correctly identified the small motor on the fine time scale and the big motor on the coarse scale. The big motor is turned on first, followed closely by the small motor. Figures 15 and 16 show four loads turning on. In Figure 15, the big motor turns on first, followed by the rapid start lamps, the small motor, and the instant start lamps. The transients of the latter three overlap, but all four loads are correctly identified. In Figure 16, the small motor follows the start up of the big motor and the transients for the two overlap. This pair of transients is followed by the transients for the rapid start lamps and the instant start lamps, with the instant start lamps' transient occuring in the middle of the rapid start lamps' transient. All four loads are identified correctly.

VI. Conclusions

With little tuning, the MLM prototype was consistently able to recognize the turn-on transients of the test loads, providing a continuous, real-time indication of active loads in the test stand. In [3], we observed that the robustness of a TED could be enhanced by working with more complicated v-section pattern sets for each load. Experiments with the MLM prototype lend credence to this statement. For example, the detector described in [1] examined only spectral envelopes P and Q, i.e., real and reactive power. The instant start lamps used in the test stand (for both the experiments described in this paper and in [1]) exhibit a single v-section in P and essentially no consistently repeatable v-section in Q. The v-section in P, furthermore, exhibits a significant variation with the precise instant where the lamps are activated in the line cycle. The MLM prototype had no trouble reliably identifying the instant start lamp bank because it searched for distinctive v-sections in *both* P and 3P.

The prototype had no difficulty even with loads employing active power factor correction. The rapid start lamp bank, for example, contains solid state active wave-

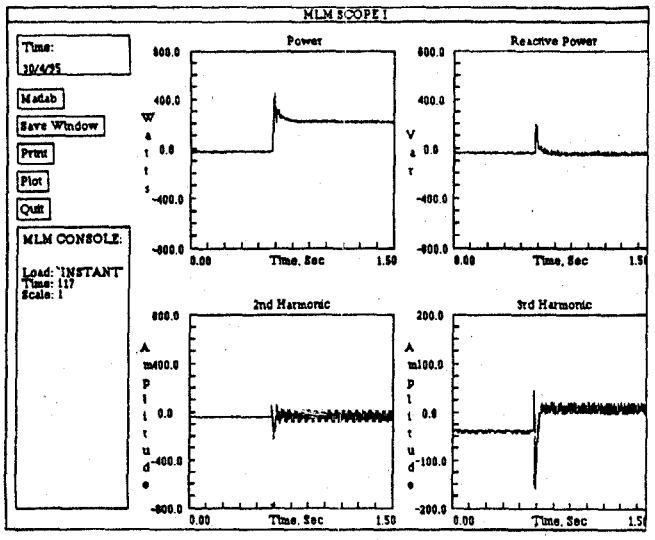

Figure 11: MLMscope Report: Instant Start Lamps

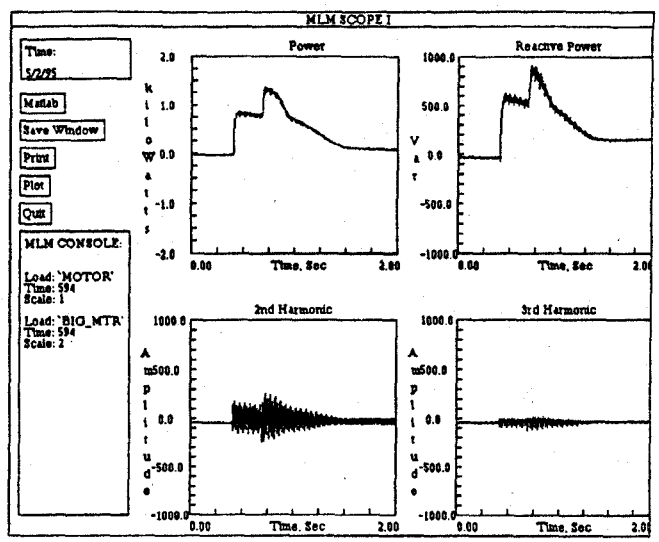

Figure 14: MLMscope Report: Big Motor, Small Motor

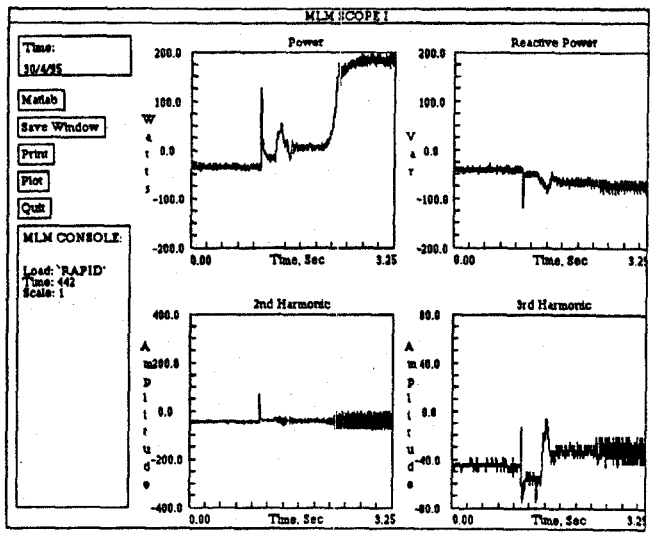

Figure 12: MLMscope Report: Rapid Start Lamps

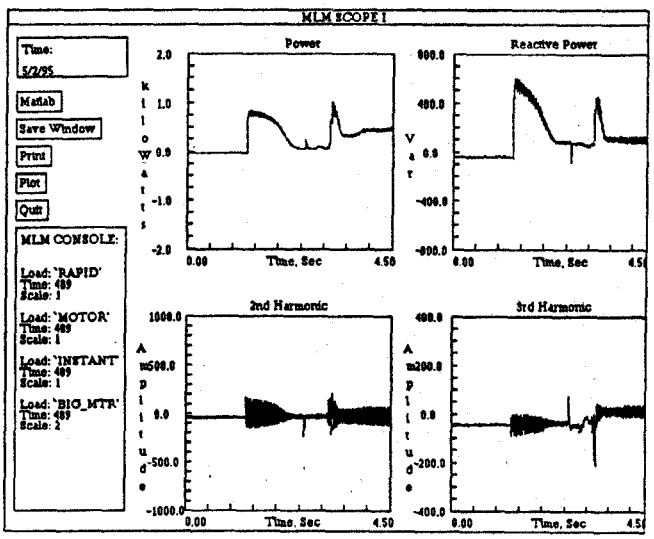

Figure 15: MLMscope Report: Motors, Rapid, Instant

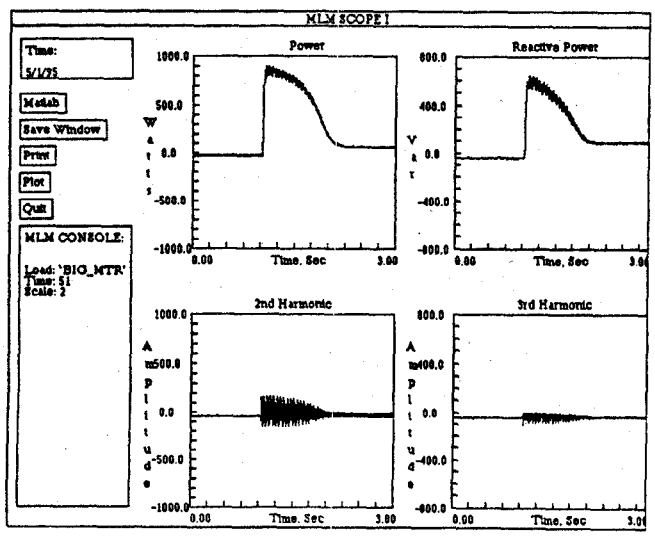

Figure 13: MLMscope Report: Big Motor

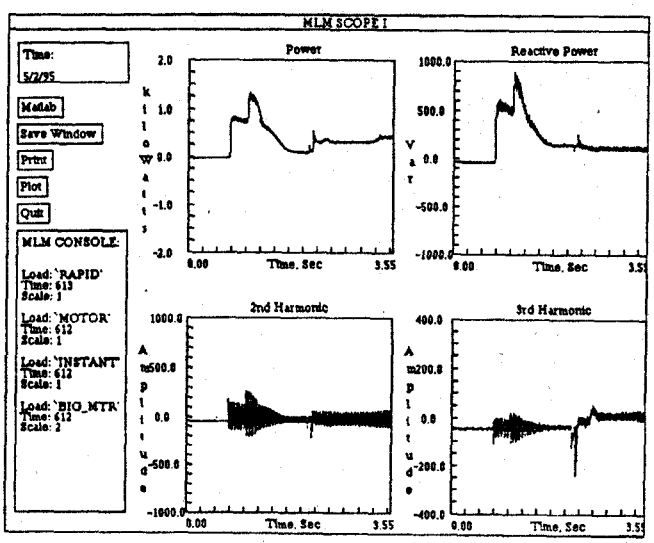

Figure 16: MLMscope Report: Motors, Rapid, Instant

shaping circuitry. This circuitry experiences a brief turn-on transient, characteristic of most power factor correction circuits. During this transient, a significant, repeatable pulse of harmonics appears in the line input current to the lamps, as shown in Figure 12. This pulse of harmonic content provides useful v-sections for transient event detection. Distinctive *transient* profiles in harmonics tend to persist even in loads which employ active waveshaping to correct *steady state* power factor.

Each slave module used in the prototype can identify between 15 and 20 typical v-sections (about 20 points each). The use of inexpensive microcontrollers provided results substantially improved, both in quantity of loads identifiable and also in identification accuracy, in comparison to the relatively expensive serial digital signal processor employed in [1] and [3]. The optimal distribution of v-sections per processor, points per v-section, and time scale assignment for different loads to maximize the number and accuracy of load identifications made by the MLM remains an active area of research.

Power quality monitoring is possible using the MLM. At present the prototype examines the 2nd, 3rd, and 5th harmonics of current from the analog preprocessor. There are sufficient (unused for our preliminary experiments) acquisition channels in the MLM prototype to facilitate the monitoring of 2nd through 16th harmonic. For complete power quality monitoring, the preprocessor already computes all harmonics up to and including 8th harmonic, and could easily be configured beyond this range. Together with its load identification capability, this harmonic content information permits the MLM to track down power quality offenders by associating the introduction of undesirable harmonics with the operation of specific loads. The MLM is also a potentially valuable platform for nonintrusive diagnostic evaluations of loads. The MLM allows raw data to be shipped — in bulk — directly to the host PC, while reporting all events detected. We suspect that, for specific critical loads, e.g. HVAC components, this wealth of data could be analyzed with state and parameter estimation algorithms to monitor load performance, enabling the scheduling of needed maintenance before a complete failure occurs.

Acknowledgements

This research was funded by ESEERCo. and EPRI. The authors gratefully acknowledge the valuable advice and support of Professors James L. Kirtley, Jr. and Les K. Norford, Dr. Richard Tabors, Frank Porretto, Laurence Carmichael, Kurt Levens, Steven Shaw, and Shreekrishna Pandey. Essential hardware for this project was made available through generous donations from the Intel Corporation and Tektronix.

References

[1] S.B. Leeb, "A Conjoint Pattern Recognition Approach to Nonintrusive Load Monitoring," M.I.T. Ph.D. Thesis, Department Of Electrical Engineering and Computer Science, February 1993.

[2] G.W. Hart, "Nonintrusive Appliance Load Monitoring," *Proceedings of the IEEE*, Vol. 80, No. 12, pp. 1870–1891, December 1992.

[3] S.B. Leeb, S.R. Shaw, J.L. Kirtley, Jr., "Transient Event Detection in Spectral Envelope Estimates for Nonintrusive Load Monitoring," *IEEE PES Winter Meeting*, Paper 95 WM 042-2-PWRD New York, NY, January, 1995.

[4] S.B. Leeb, S.R. Shaw "Harmonic Estimates for Transient Event Detection," Universities Power Engineering Conference (UPEC), Galway, Ireland, 1994.

[5] S.A. Ward and R.H. Halstead, *Computation Structures*, The MIT Press, 1990.

[6] Intel Corporation, *80C196KC: User's Manual*, 1992.

[7] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*, John-Wiley and Sons, 1973.

[8] U. A. Khan, "A Multiprocessing Platform for Transient Event Detection," M.I.T. S.M. Thesis, Department Of Electrical Engineering and Computer Science, May 1995.

[9] S.B. Leeb, and J.L. Kirtley, Jr., "A Transient Event Detector for Nonintrusive Load Monitoring," U.S. Patent Application, filed December 1993.

Umair A. Khan is a graduate research assistant in the Laboratory for Electromagnetic and Electronic Systems at MIT. He received his S.M. degree in Electrical Engineering from MIT's Department of Electrical Engineering and Computer Science in June of 1995.

Steven B. Leeb (S'89-M'93) received his Bachelor of Science, Master of Science, Electrical Engineer, and Doctoral degrees from the Massachusetts Institute of Technology in 1987, 1989, 1990, and 1993, respectively. He currently serves as the Carl Richard Soderberg Assistant Professor of Power Engineering at MIT's Laboratory for Electromagnetic and Electronic Systems. Dr. Leeb is concerned with the design, analysis, development, and maintenance processes for all kinds of machinery with electrical actuators, sensors, or power electronic drives. He is a member of the IEEE Power Engineering Society, a Fellow of M.I.T.'s Leader's for Manufacturing Program, and a member of Tau Beta Pi and Eta Kappa Nu.

Mark C. Lee is an undergraduate research assistant in the Laboratory for Electromagnetic and Electronic Systems at MIT. He is a junior in the Department of Electrical Engineering and Computer Science.