

PROTOTYPE NONINTRUSIVE APPLIANCE LOAD MONITOR

000 92 24



PROGRESS REPORT #2

Prototype Nonintrusive Appliance Load Monitor

George W. Hart

SEPTEMBER 1985

MIT Energy Laboratory
Northeast Residential Experiment Station
711 Virginia Road
Concord, Massachusetts 01742

Prepared for Electric Power Research Institute
Under Contract RP 2568-2

ABSTRACT

A method has been developed for carrying out residential load research in a nonintrusive and inexpensive manner. A Nonintrusive Appliance Load Monitor can be installed in the kilowatt-hour meter socket of a residence; it requires no entry or wiring in the house. The microprocessor-based unit analyzes the detailed power flow characteristics of the circuit to identify the nature of the major appliances operating within. It is able to determine what portion of the total energy is consumed by each of the major appliances on the circuit.

This report describes the makeup and operation of this Load Monitor, emphasizing the algorithms which are used to perform the load analysis. It also reports on the success of three field trials of a prototype Load Monitor, and the directions that current research is taking to expand the applicability of the method.

TABLE OF CONTENTS

1.	INTRODUCTION	1
2.	TWO-STATE ALGORITHM.	7
	2.1 Measurement	9
	2.2 Normalization	10
	2.3 Edge Detection.	11
	2.4 Clustering.	13
	2.5 ON/OFF Matching	14
	2.6 Separating Simultaneous Changes	15
	2.7 Transfer to Central Facility.	16
	2.8 Identification.	17
3.	REAL-TIME IMPLEMENTATION	19
	3.1 Program Structure	20
	3.1.1 Architecture and Data Flows.	21
	3.1.2 Dynamic Edge Detection	25
	3.1.3 Dynamic ON/OFF Matching.	28
	3.1.4 Dynamic Clustering	38
	3.2 Results	49
	3.2.1 Natick House	50
	3.2.2 Lincoln House.	64
	3.2.3 Acton House.	70
	3.3 Recommendations	75
4.	MULTI-STATE MACHINES	77
	4.1 Representations	78
	4.2 Algorithms.	79
	4.2.1 Clustering	81
	4.2.2 Cyclic Factor Analysis	82
	4.2.3 Traversal Analysis	83
	4.2.4 Factoring.	84
5.	APPARATUS	86
	5.1 Computational Requirements.	86
	5.2 Mounting and Data Transfer.	89
6.	FUTURE DIRECTIONS.	94
	6.1 Identification Algorithm.	94
	6.2 Multi-house Test.	96
	6.3 Multi-state Appliance Algorithms.	99

Appendicies

References

APPENDICIES

A. On Using Estimates of Variance to Truncate Distributions	.A-1
A.1 Estimation of the Mean of a Truncated Distribution.	.A-2
A.2 Estimation of σ in One Dimension	.A-4
A.3 Estimation of σ in n Dimensions.	.A-7
A.4 Proof that $k = \sqrt{n+2}$.	.A-9
A.5 Implications for Nonintrusive Load Monitor.	.A-11
B. Normalization with Noninteger Exponents.	.B-1
C. User Control Commands.	.C-1
D. Split/Join Test.	.D-1
E. Load Research Results: Appliance Cycles, Energy Profiles, and Timing Plots.	.E-1

LIST OF TABLES

3-1 Appliances Learned at Natick House	51
3-2 Appliances Learned at Lincoln House.	64
3-3 Appliances Learned at Acton House.	70
A-1 Critical Values of k in n Dimensions	.A-8
B-1 Normalizing Exponents for Individual Appliances.	.B-2
B-2 Normalizing Exponents to Minimize Cluster Size	.B-3

LIST OF FIGURES

2-1 Edge Detection	12
2-2 Example of Clusters.	13
3-1 Prototype Load Monitor	19
3-2 Architecture of Prototype Load Monitor	22
3-3 Working Buffer	31
3-4 Buffer Showing Importance of Search Order.	34
3-5 Refrigerator Transitions and Clusters.	41
3-6 Conceptual Organization of Appliance Representation.	44
3-7 Signature Space at Natick House.	52
3-8 Refrigerator Cycles and Energy Profile at Natick House	60
3-9 Refrigerator ON Duration and OFF Duration at Natick House.	63
3-10 Signature Space at Lincoln House	65
3-11 Refrigerator Cycles and Energy Profile at Lincoln House.	69
3-12 Signature Space at Acton House	71
3-13 Refrigerator Cycles and Energy Profile at Acton House.	74
5-1 Computer Architecture.	88
5-2 Collar Mounted Load Monitor.	90
5-3 Pole Mounted Load Monitor for Multi-house Monitoring	92
A-1 Distribution with Infinite Tails	.A-1
A-2 Distribution Where Mean and Maximum Do Not Coincide.	.A-3
A-3 Truncated Normal Distribution.	.A-4
A-4 Standard Deviation of Truncated Normal Distribution.	.A-5
A-5 Solving for Critical Cut-off Parameter	.A-6
B-1 Effect of Normalizing Exponent on Cluster Size	.B-4

LIST OF FIGURES (Continued)

Appendix E contains plots of appliance cycles, energy profiles and duration histograms for the following appliances:

Natick House:	Refrigerator	E-3
	Oil Burner	E-4
	Small Stove Burner	E-5
	Large Stove Burner	E-6
	Iron	E-7
	Bathroom Light	E-8
	Washing Machine.	E-9
	Dryer.	E-10
	Dishwasher Component	E-11
	Toaster Oven (Broil)	E-12
	Toaster Oven (Bake).	E-13
	Hair Dryer	E-14
Lincoln House:	Refrigerator	E-15
	Freezer.	E-16
	Water Bed.	E-17
	Hot Tub Pump	E-18
	Oven (Half 1).	E-19
	Oven (Half 2).	E-20
	Basement Lights.	E-21
	Kitchen Light.	E-22
	Hair Dryer	E-23
	Iron	E-24
	Toaster Oven	E-25
	Dishwasher Component	E-26
	Dishwasher Heating Element	E-27
	Food Processor	E-28
Acton House:	Refrigerator	E-29
	Small Refrigerator	E-30
	Ice Maker.	E-31
	Water Pump	E-32
	Hot Water Heater	E-33
	Dehumidifier	E-34
	Bathroom Light	E-35
	Iron	E-36
	Garage Door Openers.	E-37

Legal Notice

This report was prepared by the Massachusetts Institute of Technology (MIT) as an account of work sponsored by the Electric Power Research Institute, Inc. (EPRI). Neither EPRI, members of EPRI, nor MIT, nor any person working on behalf of either:

a. Makes any warranty or representation, express or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

b. Assumes any liabilities with respect to the use of, or for damages resulting from the use of, any information, apparatus, method, or process disclosed in this report.

1.0 INTRODUCTION

In this report we describe the makeup of a new Nonintrusive Appliance Load Monitor. This is a microprocessor-based device which has been developed by MIT as a low cost and easy-to-use method for utilities to carry out residential appliance load research. In contrast to existing load research equipment, which requires entry into a residence to connect sensors to individual appliances and branch circuits, this new method requires only a single external connection at the kilowatt hour meter or on a utility pole. From this external location, the load monitor carefully measures characteristics of the total residential load and uses sophisticated algorithms to determine what portion of the total load can be allotted to each separate appliance inside the residence. In the course of this operation, it first determines the number of major appliances in the residence, and the electrical nature of each one. It should be emphasized that this process is carried out in a completely automatic and unassisted manner, without the need for any appliance survey or contact with the occupants.

There are two main limitations of the device. The first is that its performance degrades as one considers smaller and smaller appliances. The greater the power consumed by an appliance, the easier it is to recognize its presence in the aggregate load. In the residential settings in which it has been tested, it appears that the threshold of useful operation occurs between 100 and 200 watts. Only appliances which consume more power than this can be accurately detected and reported on. Most of the appliances of interest to load

researchers fortunately fall in this range. Lighting appears to be the single major exception.

The second limitation is that the Load Monitor, at its current state of development, is only capable of learning and reporting on a somewhat restricted class of appliances. This is the class that we call two-state appliances. This refers to appliances which at all times can be completely described as being either ON or OFF. The majority of consumer appliances fall into this class. A second version of the algorithm is under development which is more complex, but capable of operating on the wider class of multi-state appliances, which includes not only the two-state appliances, but also appliances which have more than one type of ON state. This includes appliances such as dishwashers and washing machines which have "wash," "rinse," and "spin" or "dry" states, and fans or air conditioners with "low," "medium" and "high" settings.

Given these two limitations, we recommend that use of the load monitor be targeted towards certain appliances. Electric water heaters and refrigerators, for example, are excellent targets for the monitor. Central air conditioning should be a good target except in the case of heat-pump air conditioners with complex controllers, which are multi-state appliances. The two-state version of the monitor has been operated and tested with successful results which are reported below. When faced with multi-state appliances, the two-state load monitor either ignores them or learns their separate components (e.g. motors and heating elements) as unrelated appliances. The multi-state version is still under development. As it is improved, we believe that the class of appliance targets will be widened.

The key to the load monitor lies in the method by which the total load is analyzed. The bulk of this report describes the details of the method. Real and reactive power measurements are taken at a rate of once or more per second, and examined for increases and decreases which indicate that individual appliances are being turned on or off. The sizes of these increases and decreases are then analyzed statistically to determine which should be associated with the same appliance, and which belong to different appliances, thus giving the number of appliances. (Small changes in power, below a cut-off threshold of approximately 100 W, are ignored.) The times at which they occur are used to determine the energy consumption versus time-of-day characteristics of each of the appliances. This information is then used to identify the nature of each appliance. Power consumption versus time-of-day information is then output at monthly intervals for the use of load researchers.

A prototype load monitor has been developed using a general purpose microcomputer and a power-sensing peripheral. This prototype has been tested for two weeks in three homes. In all three cases, the load monitor successfully learned the electrical characteristics of all the major two-state appliances which operated during the test period. It was able to track their energy consumption with an accuracy between 75 and 90%. Based on these results we are continuing to refine the load monitor in ways which will improve its operation. We are confident that the next generation prototype Load Monitor can achieve an accuracy of approximately 95%.

Sections 2 and 3 of this report describe the two-state algorithm. In Section 2, the essentials of the algorithm necessary to any implementation are discussed in general. In Section 3 we describe the

details of a particular program we have written which realizes this algorithm. This program forms the heart of the prototype load monitor which has been successfully field-tested in three residences under conditions which mimic that of nonintrusive monitoring. The results of these tests are included in Section 3 and Appendix E. A second version of the algorithm, suitable for multi-state appliances, is adumbrated in Section 4. This is a report of ongoing research. The details of the multi-state algorithm have not been specified, and it has not been tested.

In Section 5, the hardware of the nonintrusive appliance load monitor is discussed from two points of view: its computational requirements, and the details of its mounting and installation. Section 6 describes the work which is yet to be completed before the load monitor can be brought forth as a commercial product. This includes developing the multi-state algorithm further, improving its ability to identify appliances by common name, and additional testing.

Before continuing with the technical details of the load monitor, an ethical question must be considered: does the nonintrusive nature of the monitor result in an invasion of privacy to the occupants? An analogy might be made between the manner in which the monitor works and that of tapping a telephone line. To emphasize the problem, consider that the method described herein is capable of detecting and reporting highly detailed information concerning the activities taking place within the house. For example, in two of the field tests described below, the prototype monitor could correctly report all the details of the bathroom light usage, printing out exactly when and for how long (to the second) it was turned on. It does this using only

information extracted from measurements that could be taken outside the house, at the utility pole.

Despite the potential for abuse of the method, we feel there is no real moral problem with the load monitor when we consider how and why it is used. It has been developed for load researchers who are interested in the energy usage characteristics of classes of appliances. For the purposes of planning future generation capacity or transmission and distribution requirements, utility planners are looking for load models which can be built up from individual appliance load models. Public policy makers require data on the energy consumption of appliance classes in order to rationally make decisions affecting the sale of appliances. Utility rate setters also need information concerning typical energy consumption of appliance classes in order to assess the worth to the utilities, and economic consequences, of different metering schemes such as time-of-day metering, demand metering, and water-heater discounts.

Thus there appears to be no legitimate use for the detailed information concerning when each appliance turns on and off. The Load Monitor is therefore programmed not to store or report the detailed information. Only average energy usage over a monthly period for each appliance is kept. The utility, which might have hundreds of Load Monitors in place, then averages this information with data from other houses. In summary, extremely detailed load information does pass through the load monitor--this is essential to the method--but it is processed so that only average information is available from the results, and individuals are not identified.

We should also note in passing that there may be legitimate uses for the detailed switching information in applications of the method

which are not directly related to this project. For example, a load controller could be designed to operate a deferrable load as a function of the on/off state of other, nondeferrable, loads. The nonintrusive technique may be suitable for sensing the state of the nondeferrable loads from a single central sensor, at a lower cost than individual sensors. A second application would be to provide detailed usage information for the homeowner for his/her own purposes. It could be a useful tool for determining exactly where and when energy is used, or for locating appliance failures. In this latter capacity, the load monitor has been used to determine that one of two underground septic pumps at the Acton House (described in Section 3.2.3) has failed. When the operating pump is manually switched, the power consumption of the house increases by an amount compatible with the manufacturer's description of the motors. However, the power consumption of the house increases only minimally when the other pump is switched on manually.

Although this report is designed to be self-contained, it may be helpful to be familiar with progress report [1]. The earlier report contains background material that explains why various options were selected here.

2.0 TWO STATE ALGORITHM

The two-state algorithm is the method by which the nonintrusive load monitor can learn and report on the nature of the two-state appliances in a residence. As defined above, these are the appliances which have only one electrically significant mode of being turned on. They do not have multiple speeds or contain independently switchable elements. The algorithm described below has been designed to learn only the two-state appliances in a residence. When faced with multi-state appliances it is designed to either ignore them altogether, or to learn their separate components as individual two-state appliances. Thus, the algorithm will learn the separate heating elements of an oven or stove as distinct electrical appliances, without realizing that they belong to the same physical assemblage.

There are two different points of view which can be taken when describing the algorithm. Suppose that one wanted to collect a year of load data from a residence. From the first point of view, we imagine that detailed power measurements for the residence for the entire year have been collected and stored in some form, and we now wish to process them. Given this large mass of data, one can in principle describe what operations to carry out in order to arrive at the final results. The calculations can be designed to optimize various parameters (e.g. minimize the error) over the entire year. Although storing and processing a year's worth of detailed data may pose technical problems, it is certainly possible in principle. From the second point of view, we consider the constraints imposed by the finite computing resources available in a small microprocessor-based load monitor. Even if the memory facilities permitted it, we do not

wish to store a complete year of data and then begin to analyze it. Instead, we wish, at each point in time to have analyzed all the data collected up to that point to the maximum extent possible.

The first point of view is a static optimization problem. We are given the data and we ask what is the most likely analysis of it in terms of a set of appliances which would explain the observations. The second point of view gives rise to a dynamic optimization problem. At each point in time we try to formulate the most likely analysis of the data collected up to that point, but we do not keep all of that data around. Instead, we keep only the minimal amount of data--a set of sufficient statistics--with which to solve the problem. As new measurements are recorded, they are not stored, but instead are used to update the statistics to arrive at an improved estimate of the most likely set of appliances given the earlier and new data. In this way, a finite machine with a small amount of memory can continue to operate for an indefinite time period; the first method would eventually lead to a shortage of memory capacity.

In this section we describe the two-state load monitor from the static point of view. This also provides a basic introduction for a later discussion of the dynamic algorithm. Those aspects of the algorithm are postponed until Section 3. In addition, the static problem gives a broader presentation because it is compatible with many different dynamic versions, of which only one is described in this report.

The overall algorithm can be broken up into eight steps, which are described in the following sections. The steps are:

- (1) Measurement
- (2) Normalization
- (3) Edge Detection

- (4) Clustering
- (5) ON/OFF Matching
- (6) Separating Simultaneous Changes
- (7) Transfer to Central Facility
- (8) Identification

Each of these steps might be performed in several alternative manners. The most suitable manner is described first, followed by options which may be appropriate in certain circumstances. (The progress report [1] contains additional alternatives.)

2.1 Measurement

The power and voltage of the residence are measured once per second. By "power" we are referring to four independent quantities to be measured every second: the real and reactive power consumption on each of the two out-of-phase legs entering the residence. These four measurements are grouped into a power vector. The RMS voltage on each of the two legs is also measured every second.

If the residence does not have two separate "legs" at its service entrance, the method simply scales to the number of independent circuits available. The voltage, real power and reactive power of each separate circuit (referenced to ground) is measured. These measurements can be combined into a vector which contains the real and reactive power of each of the separately measured circuits as its elements.

Alternative embodiments of the device could measure only the real part of the power or only the reactive part of the power. This would simplify the device at the cost of reducing its discriminating power, but might be appropriate for some class of target appliances. If it is necessary to increase its discriminating power, other measurements from the following list might be used in addition to, or in place of, the power measurements, but this has not been tested:

Power, current or admittance at the 3rd or 5th Harmonic
Power, current or admittance of sub-harmonics
DC bias current

The exact rate at which sampling occurs is not critical to the method. A slower rate can be used if it is all that the apparatus allows, but this leads to more frequent errors by the device. A slightly higher rate may be preferable, but there is no advantage in exceeding approximately ten measurements per second. The measurements need not be at regular intervals; if, for example, computational requirements necessitated the skipping of occasional samples, the overall accuracy would not be significantly affected.

2.2 Normalization

In the current embodiment of the device, the two circuit voltages are also measured every second. From this data, the real and reactive parts of the power are adjusted every second to correct for the fact that the utility allows the line voltage to vary, using the following formula:

$$\text{Adjusted Power} = \text{Measured Power} * \left(\frac{120}{\text{Voltage}} \right)^2$$

When applying this formula, each component, real or reactive, of each sample is adjusted using the corresponding voltage for that leg (or circuit) at that second.

This normalizes the power to what it would have been if the utility voltage were the nominal 120 volts. By doing this we arrive at more consistent changes in power for each appliance when we perform the edge detection procedure below. It also eliminates changes in power which are caused only by changes in line voltage.

Variations of this procedure which could be used with similar

effect are to:

- (A) Choose a different normalizing voltage instead of 120 in the numerator of the fraction; or
- (B) Choose an exponent other than 2.

Note that if 1 is used as the normalizing voltage, the normalized power becomes equivalent to admittance. The use of 2 as the exponent is a consequence of assuming the power varies as the square of the voltage. This is not exactly the case for most appliances. Exponents other than 2 are considered in Appendix B. It is shown there that it may be preferable to normalize the real part of the power using an exponent of approximately 1.5 and the reactive part with an exponent of approximately 2.5.

2.3 Edge Detection

The third step of the method is to look for changes in power with the following two-step procedure, which is illustrated in Figure 2-1 for hypothetical power measurements:

(A) Divide the sequence of power measurements into time periods in which the power is steady and time periods in which it is changing. A steady period is defined to be one of a certain minimum length in which the load does not vary more than a specified tolerance. The remaining periods, in between the steady periods, are defined to be the periods of change. The current embodiment of the device uses two seconds as the minimum length and 15 watts or VARs as the allowable tolerance in the definition of a steady period, but other values of these parameters could be used with a similar effect.

Note that a time period is defined to be steady if and only if all the measured quantities in the measurement vector remain steady. If any of the components are changing, the period is "changing."

(B) For each time period in which the power is changing, compute the total change in power across the period by subtracting the steady power level before the change begins from the steady power level after the change ends. The current method reduces the effect of noise by averaging all of the measurements (of each vector component) during each steady period to arrive at noise-reduced steady values. The change, or transition, for each period of change is therefore a four-component vector computed by subtracting the average of all the measurement vectors in the previous steady period from the average of the measurement vectors in the subsequent steady period.

Note that this description is appropriate only if it is possible to store a long stream of four-component measurements at one-second intervals. In a small Load Monitor this would be quite impractical. The prototype Load Monitor therefore uses an algorithm for edge detection which is a dynamic version of the above static description. It produces the identical effect by means of a small number of sufficient statistics, without the need for storing a long stream of measurements. Section 3.1.2 describes this dynamic edge detection.

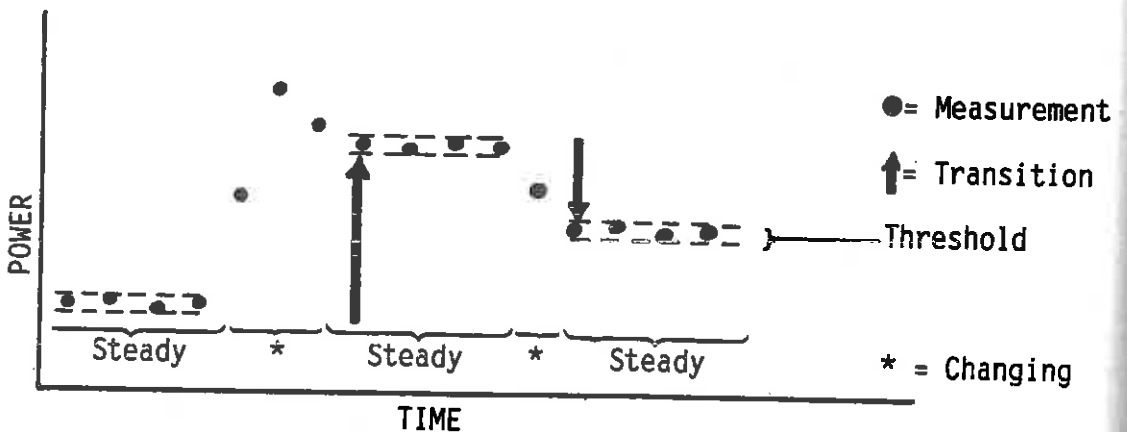


Fig. 2-1. Edge Detection.

2.4 Clustering

The observed changes are then grouped into "clusters." A cluster of changes is simply a set of changes all of which are approximately the same (in all components). For example in Figure 2-2 we show a hypothetical one-dimensional example in which many changes have been observed which can be grouped into four clusters. Each change is approximately 200, 500, -200 or -500 W.



Fig. 2-2. Example of Clusters.

The purpose of the cluster analysis is to allow for a certain variation in the measured change each time an appliance is switched on or off. The data in Figure 2-2 is what one would expect to observe if there were a 200 W appliance and a 500 W appliance in the residence. Each time the 200 W appliance turns on, the total power consumed by the home increases by approximately 200 W, but not necessarily exactly 200 W. Due to variations in the conditions when the appliance is turned on, and measurement noise in the sensors and A/D converter, the observed increase in power will not be exactly 200 W. Similarly, the cluster of changes of approximately 500 W results from the times when the 500 W appliance turns on. The clusters of changes with negative power levels result from the turning off of the appliances.

Actual data from a residence will be more complex than the above example indicates. There are likely to be several dozen clusters, because there are typically dozens of appliances in a residence.

There is more information available however, than in this one-dimensional example. The independent components of the transition vectors allow clustering to be performed in a higher number of dimensions. For example, four-dimensional clustering can be carried out to separate appliances which draw the same real and reactive power, but are on opposite legs.

Many statistical techniques of cluster analysis are well-known and could be used for this purpose (in the static formulation). For example, References [2] and [3] each list dozens of cluster analysis techniques. The clustering technique used in the prototype load monitor has several new features which allow it to function recursively in the dynamic implementation. It is described in Section 3.1.4.

2.5 ON/OFF Matching

Next, the observed changes from the ON and OFF clusters of each appliance are grouped together into pairs according to their time coordinates. Each ON/OFF pair corresponds to a single cycle of appliance usage. For example if there is a change of approximately 200 W at 6:00 and a change of approximately negative 200 W at 9:00 (with no other changes from either of those clusters in the interim), they are grouped together into one appliance cycle. From this we compute that the 200 W appliance was on for the three hours, and consumed 600 watt-hours of energy. Changes which do not fit into an ON/OFF alternation are ignored unless they can be handled by the method of the following section. A detailed account of the method by which the prototype load monitor pairs ON and OFF transitions is given in Section 3.1.3.

2.6 Separating Simultaneous Changes

From time to time, two appliances are turned on or off (or one on and one off) simultaneously, or in rapid succession so that the second appliance is switched before the transient of the first has ended. When this occurs, the change computed by the method of Section 2.3 above will be the sum of the changes that would have been observed if the two appliances were switched at different times. For example, if the 200 W and 500 W appliance are turned on nearly simultaneously, a 700 W increase in total power consumption of the house is observed. This 700 W change is easily interpreted by the facts that:

(A) It rarely happens. (e.g. The cluster of 700 W changes is very small; perhaps containing only one example.)

(B) It occurs, in time, between two ON or two OFF transitions of some appliances, which could not both be paired by the matching procedure above (e.g. The 700 W change occurs between two -200 W changes in a row--a +200 W change is missing--and between two -500 W changes--a +500 W change is missing.)

(C) The observed change is approximately the sum of the two missing changes (e.g. $700 = +200 + 500$).

When all three of these conditions occur, the unusual observed transition is "broken apart" into its two simultaneous components, and the procedure continues as if the two components were available for matching ON's and OFF's as above. Thus the load monitor "understands" that the 700 W change was really two independent appliance transitions which happened to occur at the same moment.

Note that the prototype Load Monitor, as it stands, does not decompose simultaneous transitions. Time has not permitted incorporating the algorithm into the current software. The above technique was developed and tested in the "Recognition Program" described in Reference [1]. We expect that it will also work when implemented here. The next generation prototype Load Monitor, described in

Section 6.2 will incorporate this technique.

2.7 Transfer to Central Facility

The final step of the method that is performed by the micro-processor unit is to output the characteristics of the observed appliances. This includes a description of the clusters in the signature space and parameters specifying their electric power usage such as their total energy consumption. Many such parameters could be selected. In all probability, load researchers would prefer energy to be broken up by hour of the day on weekdays and weekends. (Thus there will be 48 numerical energy values per appliance each month.) Other temporal divisions are, of course, easily arranged if they are of interest to the end users of the data. We expect output to occur at approximately monthly intervals. The energy for any appliance during any given hour is simply the sum of the energy consumed in each of the observed cycles (as calculated in Section 2.5 above) which happened to occur during the specified clock hour. Time-of-day usage plots of this form, which demonstrate the type of results the load monitor is capable of, are given in Section 3.2 and Appendix E.

Another parameter which energy consumption can be correlated with is temperature. We expect the load monitor will contain a temperature sensor so that the sensitivity of the appliance to temperature can be tabulated. (A temperature-humidity index may be even more suitable.) This information should be very useful for identification purposes. For example it should enable space heaters to be clearly distinguished from other large resistive loads. It will also be important for load researchers reconstructing space conditioning loads as a function of weather models. We are not sure at this point what form of energy-temperature data is most useful for this latter

purpose, so we will not propose any particular format here.

The actual data transfer could be performed by telephone link directly to a central computer, or by transfer to an intermediate storage device which is carried from house to house by a meter reader. Various hardware options for data transfer and temperature measurement are considered in Section 5.2.

2.8 Identification

Each ON/OFF pair of clusters (a positive and negative cluster of the same magnitude) represents a separate two-state appliance or appliance component (for example the heater and motor components of a dishwasher may be observed as two separate ON/OFF clusters). The algorithm must examine the properties of the clusters and try to identify the appliance class of each (e.g. "refrigerator," "heater of dishwasher," etc.). To do this, a table of appliance classes and their properties will be provided. The algorithm will check each cluster against the classes in the table to see which item in the table is closest to each observed cluster pair. The properties used will include real and reactive components of the turn-on transitions. For example, refrigerators as a class are expected to exhibit a change of 100-500 W and 100-500 VAR on only a single phase when they turn on, while electric water heaters are expected to be approximately 4000 W balanced on two phases. Weather related correlation factors can also be included. As mentioned above, space heating can be identified by the fact that it on more frequently when it is cold outside. Air conditioners should be identifiable by their positive correlation with temperature.

The table will also contain timing information, such as the

average length of time per ON/OFF cycle of the appliance and the number of cycles per day. Expected time-of-day and time-of-year properties can also be used (e.g. lights are used more often at night, electric lawn mowers are used more often in the day and in the summer). We do not wish to rely too heavily on temporal expectations however, as this could cause the load monitor to fulfill its own prophecies, and only find results which are predictable. For example, if the table erroneously claimed that lights are only used in the evening, and someone ran lighting all day, there would be a danger that the load monitor would misidentify the lights and call them by some other name.

It is important to stress that the identification portion of the method has not yet been developed. We see no major obstacles in doing so however. It is discussed further in Section 6.1.

3.0 REAL TIME IMPLEMENTATION

This section describes the current status of the prototype load monitor. The hardware consists of a Hewlett Packard 9845B desktop scientific computer and a Digital AC Monitor [4] as a sensor instrument, as shown in Figure 3-1. The HP9845B computer runs a program which incorporates a dynamic version of the two-state algorithm. This program consists of approximately 1500 lines of interpreted BASIC code requiring approximately 64 K bytes of memory. Built-in tape drives and printer are used as output devices for the load data. The Digital AC Monitor is a programmable microprocessor-based sensor device which can measure voltage, and real and reactive power on up to eight AC circuits simultaneously. It transmits these measurements to the HP computer over an RS-232 link at one-second intervals.

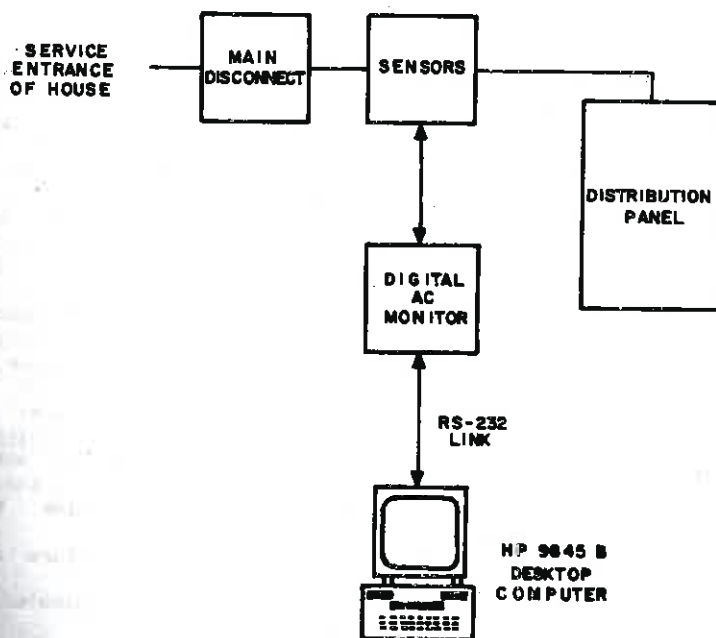


Fig. 3-1. Prototype Load Monitor.

Section 3.1 below describes the architecture of the BASIC program which forms the "brains" of the load monitor. Section 3.2 describes the results of testing the load monitor on three residences. In Section 3.3 we list the improvements to the load monitor which we would like to make in the near future, based on the testing to date.

3.1 Program Structure

Most of our research effort over the past year has gone into designing, implementing, and testing the software which controls the load monitor. The program executes what was termed the dynamic two-state algorithm in Section 2. It therefore contains capabilities which allow it to run continuously in real time, which were not discussed above. Some of these complications consist merely of buffering techniques which allow the processor to be focused on certain computation-intensive aspects of the problem while interrupt-based processes continue to collect and analyze measurements from the sensors at regular one-second intervals. These data-flow aspects of the program are described in Section 3.1.1 below. Section 3.1.2 details the manner in which transition-detection is performed by the load monitor. The static description given above in Section 2.3 is adapted so that it does not require storage and subsequent processing of a potentially unbounded set of measurements. The technique chosen to match ON and OFF transitions is explained in Section 3.1.3. Again, a method was selected which allows a finite memory to be used for pairing transitions over a potentially unbounded time period. Additional complications arise from the need to perform cluster analysis in a dynamic and open-ended manner. These problems have necessitated the development of a new cluster analysis technique which is described in Section 3.1.4.

3.1.1 Architecture and Data Flows

Flow of control and data in the load monitor is shown in Figure 3-2. The general flow of information proceeds from left to right across the center of the diagram. Measurements from the Digital AC Monitor are processed at one-second intervals, and normalized to 120 V as described in Sections 2.1 and 2.2. Real and reactive power on each of the two legs form a four-component measurement vector. An edge detection algorithm is called every second with the new measurement and seeks to locate and quantify step changes in the measurements over time. The edge detection algorithm results in step transitions as described in Section 2.3 above, but this is achieved in a slightly more complex manner. A dynamic edge detection method, described below in Section 3.1.2, allows for edges to be found without storing long streams of measurements.

These three parts of the algorithm, measurement, normalization and edge detection, are performed as an interrupt process. The Digital AC Monitor is programmed to transmit measurements at one-second intervals. When they are received by the HP9845B computer, the processing subroutines are triggered. Generally, no edge is detected, and these subroutines return control to the background process for the remainder of the second until the next measurement arrives. If an edge is detected, then it is converted to a two-and-one-half-dimensional format and buffered in the pre-buffer before the subroutines relinquish control.

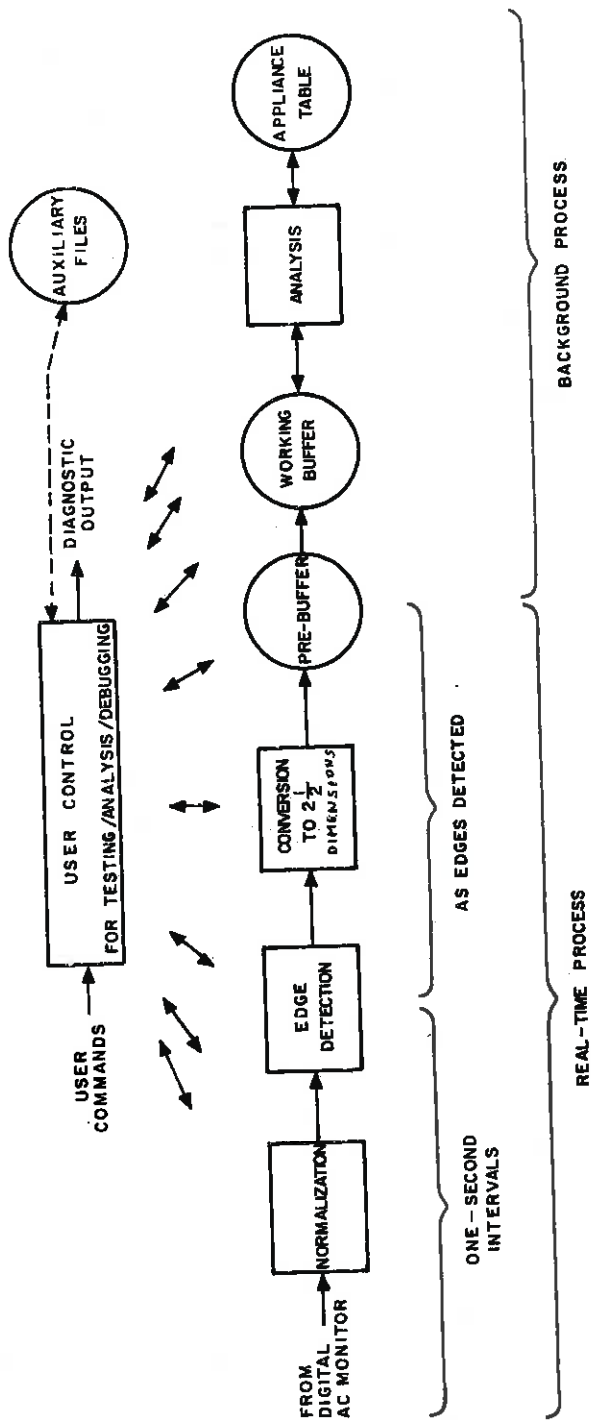


Fig. 3-2. Architecture of Prototype Load Monitor.

The pre-buffer is a circular format buffer which stores transitions until the background process has time to analyze them. The format in which they are stored and later processed is called "two-and-one-half-dimensional" because the four-dimensional measurement vector is compressed to a format with fewer degrees of freedom. The assumption is made that all loads are either 120 V appliances or balanced 240 V appliances. Then the transition observed when an appliance turns on can be expressed as a two-dimensional quantity, the real and reactive power it draws, along with a quantity which takes on one of three discrete values. This latter quantity is a "flag" which indicates which of the two legs the appliance is on, or whether it is wired across both legs. Thus the space can be visualized as three parallel two-dimensional planes.

The two-and-one-half-dimensional format was selected partly because it reduces storage and computation time at later stages of the process, and partly for reasons discussed in Appendix A. The cost of this format is rather high however. The program, as it stands, can not learn about unbalanced 240 V appliances. No provision is made for appliances which draw unequal amounts of power on the two 120 V legs. It was expected that such appliances would be rare. Results from two of the three tests described in Section 3.2 show that this expectation was incorrect. Unbalanced 240 V appliances seem rather common, and therefore we will adopt a four-dimensional representation in future versions of the load monitor. This will increase the range of loads which can be targets.

The right half of Figure 3-2 shows the transition-analysis portion of the load monitor. The transitions which were held in the pre-buffer are moved to a working buffer for analysis to determine how

they relate to each other and the set of appliances already learned. This portion of the algorithm operates as background process, working in the fraction-of-a-second gaps between the time intervals required to process the one-second measurements. As such, this portion of the program typically operates slightly behind real-time. When the load monitor is started up, the appliance table is empty. Initial transitions are analyzed to create entries in the table. Later transitions could be analyzed as new appliances, or used to update the properties of previously-learned appliances. Sometimes two entries in the appliance table need to be fused together into a single appliance in the light of later data. Other times, an entry is split into two distinct entries. In this manner, new data is used to correct and update earlier estimates. Section 3.1.4 below details the dynamic cluster analysis technique used to perform these corrections.

The user control portion of the prototype load monitor is indicated at the top of Figure 3-2. Although only sketched in the figure, this function of the program is by far the largest in the prototype. It permeates and interacts with all of the previously mentioned portions. By means of a simple command language, it allows the user to examine and modify the operation of the load monitor's different functions. It can generate listings or plots of measurements, transitions, or clusters of transitions which define appliances. Output can be sent to the computer screen, a printer, a plotter, or to data files. It also allows for a variety of input sources. Keyboard input of measurements or transitions can be made for testing purposes. Data files of transitions can be "played back" to see the effect of changing parameters. Although the user control

portion of the load monitor consumes over half of the total software, and is essential for testing, analysis and debugging, this much versatility will not be needed in the actual load monitor. Therefore, it is not described in detail in this report. The commands which it accepts are listed in Appendix C.

3.1.2 Dynamic Edge Detection

The edge detection process described above in Section 2.3 has been implemented in a "dynamic" manner. By this we mean that it is programmed in a way which does not require storage and subsequent processing of a long list of measurements as Section 2.3 might seem to imply. Instead, a few parameters are kept and updated every second based on the new measurement. This is done in such a way that the program returns the same results as would be generated if all the measurements were kept. (It may be useful to review Figure 2-1 while reading the following discussion.)

One aspect of edge detection which uses a long stream of measurements is the averaging process which reduces the effects of measurement noise. All of the one-second power measurements which are taken during the relatively steady time period between two successive appliance transitions can be considered to be approximations to the actual power level that the house is at during this period. They are only approximations because small variations due to appliance operation and measurement noise will always be present. It can be shown (given certain reasonable assumptions) that averaging the entire list of measurements will result in the optimal estimate of the actual power level. The actual set of measurements could contain tens of thousands of entries if several hours elapse between appliance transitions. In order to calculate the average of this list of measurements

without actually storing it, the following formulas are applied every second to each new measurement vector. The counter, N, is initially zero, which causes the first measurement to become the first average. Subsequent measurements update the average vector and the counter in a way which continuously reduces the weight of later measurements.

$$\text{Average} = \frac{N}{N + 1} \text{ Average} + \frac{1}{N + 1} \text{ Measurement}$$

$$N = N + 1$$

The algorithm also compares each measurement vector to the previous measurement vector to check for changes beyond a specified threshold. A change larger than the specified size defines the beginning of a time period in which the load is changing. When this occurs, the steady period is over, so the program can gain no more benefit from averaging. It therefore computes the net change from the previous steady period to the one which just ended (by subtraction of vectors) and passes this change along to the routines which convert it to two-and-one-half dimensions and store it in the pre-buffer. This only requires the storage of three vectors: the average value now, the average during the previous steady period, and the previous second's value. Note that the method which is used results in the program always being one transition behind. Only when a transition begins can the previous transition be quantified, because the steady period after each transition is averaged in determining transition size.

Whenever the load is changing faster than the specified threshold, the program sets a flag that indicates a period of change is in progress. This flag may be set for several seconds while passing over a spike in the load or waiting for a start-up transient to decay

If a subsequent pair of measurements differ by less than the threshold, and the flag is still set, then this indicates the beginning of a steady period. Accordingly, the flag is cleared, and averaging to estimate the new steady value begins. The ability to pass over starting spikes and transients, which we have observed to be quite variable, is the key to determining consistent signatures.

The dynamic edge detection algorithm is summarized by the following steps:

- (1) Initialize the following four-component vectors to zero:
 - E -- estimate of the actual power level during this steady period, based on averaging measurements
 - L -- the last power value which was steady for at least two seconds
 - P -- the measurement from the previous second.

Clear the flags:

- A -- set if the power level is changing this second
- C -- set if a change is in progress over a number of seconds.

Select a threshold power level for defining steady-state periods. Repeated measurements of a steady load are expected to vary less than this threshold. (We have used 15 watts as this level.)

Select a noise level for defining significant appliances. Appliance transitions below this level will be ignored. (We have used 70 watts.)

Then repeat steps (2) through (8) every second with the normalized measurement vector, M, for that second.

- (2) Get the measurement M for this second and determine the change in power, M-P, from last second to this second. If any component of M-P exceeds the steady-state threshold, set a flag, A, which indicates the load is active this second (otherwise clear A).
- (3) If A is set and C is clear, a transition is just beginning, so process the previous transition via steps (3A, B and C), (otherwise skip these steps):
 - (3A) Calculate the size of the previous transition as E-L. This value is the output of the edge detection process; if it is larger than the significance threshold, it, and the time held in T, is passed to the routines which buffer transi-

tions for later analysis. (Note the very first such output is ignored because it is the transition from zero to the first steady period.)

- (3B) Set L to E. This stores the current estimate of the steady value for use in computing the next transition.
- (3C) Set T to the current time, for use later when processing the transition which is just beginning this second.
- (4) If A is set, a new steady period may be beginning, so zero the counter N.
- (5) Update the estimate, E, which is the average of all measurements during this steady period, using:
$$E = (N * E + M) / (N + 1)$$
- (6) Add one to the counter N, which notes how many measurements are incorporated in E.
- (7) Set C to the value of A. If set, this records that the load is changing and we are waiting for it to become steady.
- (8) Set P to M. This holds the measurement for comparison next second.
- (9) Go back to Step (2).

The edge-detection method described here appears to operate quite satisfactorily. Refinements might be made, however, to improve its operation in certain circumstances. The updated version of the load monitor, described in Section 6.2, may incorporate an updated edge detector.

3.1.3 Dynamic ON/OFF Matching

As discussed above in Section 2.5, it is necessary to determine the time period that the appliance was on in order to tabulate its energy usage. This ON/OFF matching aspect of the load monitor is fairly independent from that of clustering transitions. There are two orders in which the pairing and clustering can occur. The existing prototype load monitor pairs first and then clusters the pairs. That is, it matches ON transitions with subsequent OFF transitions

determine an appliance cycle before interacting with the table of appliances to see which appliance, if any, it is a cycle of. The pairing takes place based on the size of the transitions. Two transitions which add up to zero, within a small threshold, are considered a pair. The average of the ON transition and (the negative of) the OFF transition is then used for clustering purposes. This method is suitable for the two-state load monitor because we are only targeting appliances in which the ON and OFF transitions are approximately equal.

Note that for the multi-state load monitor this technique will have to be reversed because the proper matching will not be obvious based on size alone. Transitions will be clustered first, and then the ON and OFF relationships between the clusters will be found. This will allow the method to target appliances in which the ON and OFF transitions do not match, but will introduce certain complexities necessary to find the proper cluster pairings. For example, it is possible that just one of the two clusters might be overlapping with a cluster from another appliance.

With this background, the workings of the right side of Figure 3-2 can now be explained. Transition analysis is centered on the interactions between the observed transitions and the table of appliance clusters. The interactions occur in both directions: new transitions force changes in the set of appliances, and the observed clusters guide the analysis of the new transitions. Transitions are held in a data structure we call the working buffer. They are loaded out of the pre-buffer into the working buffer for analysis as space and time permit. The format of the working buffer can be seen in the example of Figure 3-3a and b, which is taken from the test house

described in the following section. In the top half of the example, there are eight transitions (numbered zero through seven) in the working buffer. For each transition the printout records:

Index	The place in the buffer. (The earliest position is numbered zero.)
Leg	"1" or "2" to indicate one of the 120 V legs. "3" signifies a balanced 240 V appliance.
Power	The real power component, in watts, of the ON transition. (If leg "3", the real power on each leg.)
React	The reactive power component, in VARs, of the ON transition.
Time	The time of occurrence (in hours after the load monitor was turned on).
Mark	A flag set to "1" if the transition has been fully analyzed or "0" if unanalyzed.

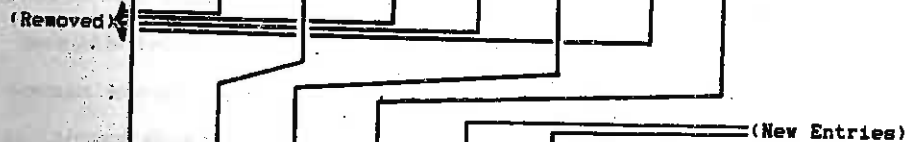
New transitions are inserted at the right side in the working buffer, and are held there while being paired up in ON/OFF pairs. They may be fully analyzed and removed in a fraction of a second, or remain there for hours, depending on circumstances. Note that the transitions are always ordered chronologically; earlier transitions are found at the beginning of the buffer. The number of transitions in the buffer is variable. If all the observed transitions are easily paired and have been fully analyzed, they will have been removed from the buffer and it will be empty. On the other hand, if many appliances have turned on, but the matching OFF transitions have not yet occurred, the buffer can be filled to some specified maximum size with the ON transitions. The oldest transitions which are sitting unmatched in the buffer are removed when space is needed for new transitions. After an ON/OFF pair is found and analyzed, the "mark" bit is set on their entries, which flags them as removable. They can

then be deleted from the buffer by a "cleaning" procedure when space is needed. The remaining unmatched entries are compressed to the left to make room for new entries to the right.

Index:	0	1	2	3	4	5	6	7	(Counter)
Leg:	1	2	1	2	2	3	2	2	(3=240 V)
Power:	-84	259	-120	228	-236	695	-250	72	(Watts)
React:	-6	267	1	562	-259	4	-559	2	(VARs)
Time:	.02	1.22	1.22	1.35	1.39	1.42	1.42	1.47	(Hours)
Mark:	0	1	0	1	1	0	1	0	(1=matched)

Comment: Light OFF Refr ON Light OFF Furn. ON Refr. OFF Burner ON Furn. OFF Light ON

Figure 3-3a Buffer Before Refilling



Index:	0	1	2	3	4	5	(Counter)
Leg:	1	1	3	2	3	2	(3=240 V)
Power:	-84	-120	695	72	-694	-70	(Watts)
React:	-6	1	4	2	-4	-3	(VARs)
Time:	.02	1.22	1.42	1.47	1.49	1.55	(Hours)
Mark:	0	0	0	0	0	0	(1=matched)

Figure 3-3b Buffer After Cleaning and Refilling

Fig. 3-3. Working Buffer.

In the example of Figure 3-3a, entries 1 and 4 form an ON/OFF pair as do entries 3 and 6. In the first case the refrigerator was on for .17 hour and in the second case the furnace (oil burner) was on for .07 hour. The buffer is shown at a point after these entries have been paired and processed, so they are marked for deletion. After the marked items are deleted, the remaining entries are compressed into the four leftmost positions, and two new transitions are placed in the vacancies to the right, as shown in Figure 3-3b. The new transitions had been stored in the pre-buffer during the time that the previous buffer contents were being analyzed. (Note that the two new transitions will form pairs with two of the transitions which remained in the buffer, so after the next buffer cleaning, only the first two entries will remain.)

If the four transitions had not been paired and marked as deletable, and additional items were waiting in the prebuffer for analysis, then several of the oldest transitions would have to be removed. In this way, transitions which can not be paired will eventually be cleaned out of the buffer. Unfortunately, if the maximum buffer size is too small this process will also remove ON transitions for which the pairing OFF transition has not yet occurred. On the other hand, if the buffer size is allowed to grow too large, it will tend to be filled with unpairable transitions generated by multi-state appliances, and much computer time will be consumed attempting to pair them. Buffer size is therefore one of the parameters to be optimized in the design process.

For the two-state load monitor, a pair is defined as two entries which meet the following four conditions:

- (1) They are on the same leg, or are both 240 V,
- (2) They are both unmarked,
- (3) The earlier has a positive real power component, and
- (4) When added together, they result in a vector in which the absolute value of the real power component is less than 35 Watts (or 3.5% of the real power, if the transitions are over 1000 W) and the absolute value of the reactive power component is less than 35 VAR (or 3.5%).

The fourth requirement states that the ON and OFF transitions match within 35 Watts and VARS, or 3.5%, whichever is larger. Thus a 500 W ON transition can be matched to an OFF transition between -465 and -535 W (if the reactive power components also match). The size of the allowable disparity increases to 3.5% for components over 1000 W because large appliances are observed to exhibit proportionally larger mismatches. The parameters 35, 1000 and 3.5% were selected based on measurement of a few individual appliances, and will likely be "tweaked" in the future.

When searching the working buffer for pairs, the order in which the entries are examined is very important. If an appliance has turned on and off several times in succession, there can be many possible pairings between entries in the buffer. The algorithm must not allow an ON transition to match an OFF which occurred at the end of a different cycle, so that only ON/OFF pairs which truly belong together are paired up. Otherwise, the energy consumption of the appliance will be greatly overestimated. The most straightforward search procedures can make errors of this nature when faced with certain types of transition sequences.

The hypothetical buffer in Figure 3-4 shows a situation in which a 1000 W appliance has turned ON and OFF two times in succession. The situation is confused slightly by the fact that a 100 W appliance turned OFF simultaneously with the first OFF transition, to give a -

1100 W transition. As such, the first two transitions do not meet criterion (4) governing the size relationship between members of a pair. If the pairing algorithm simply started with the first element of the buffer, and checked left-to-right for a matching OFF, it would find the second OFF transition, and erroneously conclude that the appliance was on for the entire time period. Instead, the correct way to search the buffer is to start by checking elements which are close together in the buffer, and gradually increase the distance. First, adjacent elements are checked for pairs which meet all four requirements above; if any are found they are processed and marked. Then elements two entries apart are checked, then three, and so on, until the first and last element are checked. By this procedure, the second ON/OFF pair of the sample buffer would be found in the first pass, and the incorrect pairing is avoided. (The first pair, with the obscuring 100 W addition, will remain in the buffer to be decomposed by the methods of Section 2.6.)

Index:	0	1	2	3	(Counter)
Leg:	1	1	1	1	(3=240 V)
Power:	1000	-1100	1000	-1000	(Watts)
React:	0	0	0	0	(VARs)
Time:	.1	.2	.3	.4	(Hours)
Mark:	0	0	0	0	(1=matched)

Fig. 3-4. Buffer Showing Importance of Search Order.

After pairing, an average transition size is defined for clustering purposes. This is just half of the difference between the ON and the OFF transitions. For example if the ON was 500 W, 5 VAR and the OFF was -520 W, 3 VAR, the average is 510 W, 1 VAR. This is done because the current implementation of the load monitor pairs ON and OFF transitions before clustering. As mentioned above, the multi-state algorithm will have to reverse this sequence. The discussion of clustering in the next section is equally applicable to either method.

With this much description of the components, we can now summarize the background process which operates to repeatedly fill, use, and clean the working buffer. The following procedure constitutes the load monitor program:

- (1) Initialization. All variables and tables are zeroed out. The Digital AC Monitor is initialized. An interrupt process is set up to receive the one-second measurements, normalize them, check for edges, and place the edges in the prebuffer.
- (2) Load the Working Buffer. Transitions waiting in the prebuffer are placed into the working buffer until it is full or the prebuffer is empty. If there are no transitions to place in the buffer, the process waits at this step until one is generated.
- (3) Find and Process Pairs. The buffer is searched for pairs in the adjacent-to-far-apart sequence. For each pair found, the following steps are taken. The details of how these steps are performed are given in Section 3.1.4.
 - (3A) The table of known appliances is checked to see if the pair is a cycle of a new or known appliance.
 - (3B) If new, a new entry in the table is made. If old, the entry for that appliance is updated.
 - (3C) The pair of transitions in the working buffer is marked so that neither will be matched again in the future.
- (4) Simultaneous Transitions. Multi-state Machines. At this point, remaining unmarked transitions can be checked by the procedure that analyzes simultaneous transitions and by the procedure that analyzes multi-state machines. Neither of these procedures is implemented in the current prototype load monitor.

- (5) Buffer Cleaning. Marked transitions are removed from the buffer and the remaining unmarked transitions are compressed leftward. If the buffer is full, which only happens if none of the transitions in it could be paired, then the oldest transition is deleted, and the buffer is compressed again.
- (6) Loop Forever. Go back to step (2).

The main virtue of this procedure, over others that might serve the same purpose, is its adaptability with respect to transition analysis. Placing a recent history of transitions in a buffer allows the easiest analysis (ON/OFF pairs) to be found first, and then successively more complex analysis (simultaneous transitions, multi-state machines) to apply to the unanalyzed remainder. The buffer format allows unmatched transitions to be kept "on hold" for a long time period while waiting for their matching transitions.

There is a significant weakness of this method which we plan to remedy in a future version of the Load Monitor. The algorithm sometimes performs an erroneous ON/OFF pairing across a long time interval. This happens when an OFF transition and the immediately following ON transition of an appliance are both "garbled" in some way to make them unpairable. This might be the result of other appliances switching on or off simultaneously for example. When this occurs, the two cycles and the intervening time period in which the appliance was off are erroneously analyzed as a single cycle. The initial ON is matched to the OFF transition of what is actually the following cycle. This is a rare occurrence because it requires two independent coincidences to occur during adjacent transitions of the same appliance, but it has been observed to occur. When it does happen, it can cause a significant error in the monitor's assessment of the appliance's energy consumption. The monitor tabulates energy as if the appliance was on for the entire time period from when the first cycle begins

until the second cycle ends. This may be hours or days. (An enlightening analogy is the problem of a parity error in a computer memory: if one occurs it is detected, but two simultaneous errors within a small memory unit are not detected and may cause serious problems.)

We believe the solution to the problem is fairly simple, although it is as yet untried. The ON/OFF matching method described above relies only on transition size, and never looks at the total power level of the aggregate load. As such it can match a 1 kW ON transition with a subsequent -1 kW OFF transition even though the total power consumption of the house dropped below 1 kW in the interim. Although it is clear in this case that the appliance could not have been operating during the time when the total power was less than 1 kW, the algorithm, as it stands, does not make use of this fact. By adding the following condition to the four listed above on page 33, ON/OFF matching can be made more accurate:

- (5) The unassigned power level between the ON and OFF transitions must never drop below the operating power level of the appliance (with a little leeway given for measurement noise).

To implement this restriction it is necessary that the working buffer contain power level information in addition to transition entries. This is easily handled by alternating transition entries with entries which record the average power level between the transitions. The force of the term unassigned above is understood by considering that the measured power level is the sum of the power consumed by all of the operating appliances. The algorithm, will have to decompose that power measurement into its separate components by subtracting the operating power of each appliance from all power

measurements in the working buffer between a matched pair of ON and OFF transitions. The power level remaining in the buffer will then be what is left over after all the known appliances are accounted for. If the level drops significantly below zero at any time, then one of the pairings made across that time period must be erroneous, of the type described above. Examination of the transitions surrounding the negative power period will determine which appliance was mis-paired. The "garbled" OFF and ON transitions which were not matched will surround the error, and from their size it can be corrected.

An important benefit of keeping track of unassigned power consumption is that it allows the residual power consumption to be tabulated. The residual power consumption is simply the unassigned power after all possible pairings are made. It should correspond to the energy consumption of all the small appliances in use. The average value of the residual will be computed on an hourly time-of-day basis for weekdays and weekends just like that of appliances. This should provide the remaining information necessary for load researchers to generate appliance-based models of residential energy consumption.

3.1.4 Dynamic Clustering

The dynamic clustering technique is the single most complex aspect of the Nonintrusive Appliance Load Monitor. The problem to be solved at each moment in time is to take all of the transitions detected since the load monitor was turned on and arrive at a suitable clustering. The correct clustering will maintain a one-to-one relationship between clusters and appliances: each cluster contains all the transitions resulting from exactly one appliance. As new transitions are observed, they should be incorporated into new or

existing clusters according to whether they were generated by new or previously learned appliances. This must be performed in a way which does not require storage of all the transitions observed since the monitor was turned on (which could be a year or more of transitions). Instead, just enough information must be extracted from the transitions to allow proper updating of the set of clusters when a new transition is processed.

A difficulty comes about because it is extremely unlikely that the exactly correct set of clusters would be found by any clustering technique. Instead, techniques are chosen which attempt to maximize the likelihood of a correct clustering, given certain assumptions about the appliances. Slight changes in the observations can result in large discontinuous changes in the most likely set of clusters. For example, given two clusters which almost merge into a single cluster, a technique must be used to decide if there are two similar appliances generating the observations or a single somewhat inconsistent appliance. Given any such technique, borderline cases can be generated which are sensitive to small changes in any single observation. Thus, all the transitions observed up to a given time could support the hypothesis that there is only one cluster, but when combined with the next transition, would support the hypothesis that there are two distinct appliances. Conversely, the datum might support the hypothesis that there is only one appliance while the earlier transitions suggested two. Thus the dynamic clustering technique must be able to join and split clusters when necessary based on new evidence. Interactions which reorganize groups of three or more clusters can then arise by iteratively splitting and joining pairs as necessary.

In order to determine whether or not to split a cluster, and how to do so should it be necessary, we have developed a representation system in which each appliance is represented by three ellipses in the signature space. One, the main cluster indicates the range of real and reactive power values which have been observed to form a cluster of transitions, and which the program is treating as being associated with the appliance. The other two ellipses associated with an appliance are termed the sub-clusters. These represent the sub-ranges of real and reactive power values which are the best estimate, at any given time, of how the main cluster would be split into two appliances if that were necessary.

Figure 3-5 shows an example of a cluster with its two sub-clusters. The example is that of the refrigerator of the house described in Section 3.2.1 for a one-day period. The asterisks mark the observed transitions; the ellipses indicate the properties of the clusters. The center of the main cluster marks the position in the signature space of the average ON/OFF transition: 265 Watts and 265 VARs. The shape of the main cluster ellipse indicates the range of scatter which has been observed about this average. The elongation to the left and right indicates that the real power tends to vary from the average far more than the reactive power. The slight tilt to the upper left and lower right shows there is a negative correlation between real and reactive components; if the real power is higher than average the reactive power tends to be lower than average, and vice versa. This is typical of induction motors. It is a consequence of the non-linearity of their power consumption with respect to changes in line voltage. The effect is discussed further in Appendix B and in

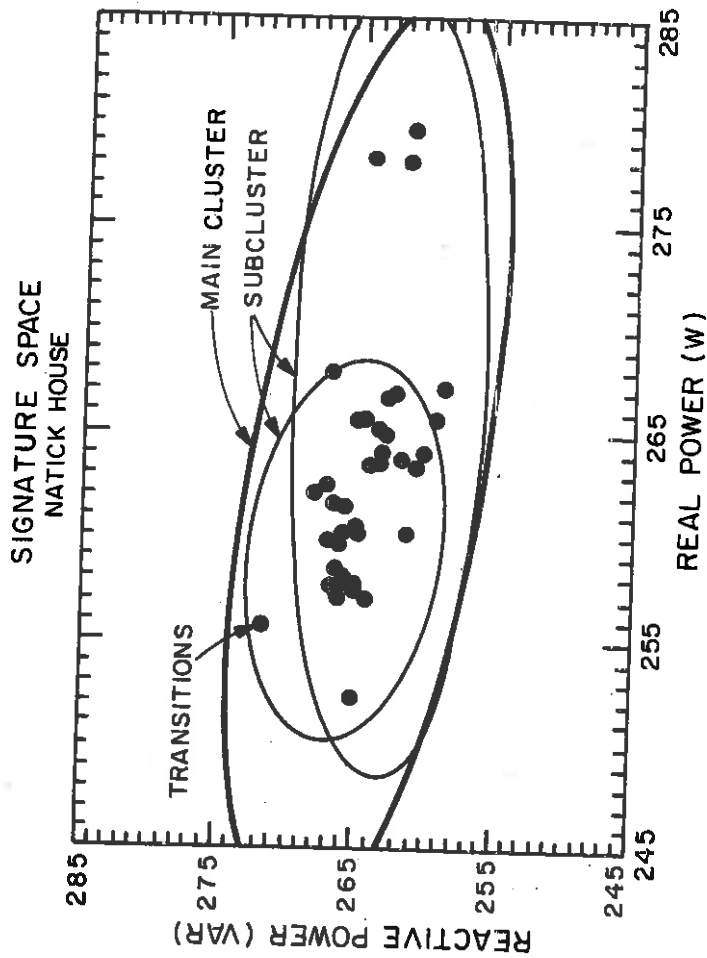


FIG. 3-5. REFRIGERATOR TRANSITIONS AND CLUSTERS.

Reference [1]. The two sub-clusters in the figure cover roughly the same area as the main cluster, without an intervening gap. This is typical of the sub-clusters of a single appliance. If the program had erroneously joined the transitions of two separate but similar appliances into one main cluster, we would expect to find two well-separated tight sub-clusters.

Clusters, whether main clusters or sub-clusters, are represented by elliptical regions of the signature space. The ellipses are actually regions of sufficiently high density in a two-dimensional normal (Gaussian) distribution. Such a distribution is represented by five parameters: the x and y value of its mean, the variances in the x and y directions, and the covariance between x and y. The first two parameters determine the placement of the center of the ellipse in the plane. The latter three determine its size and orientation. Ideally, each appliance would be perfectly consistent, and all measurements would be perfect, so that every transition from any given appliance would be identical, and the ellipses would collapse to zero radius. In fact, this is not the case, so the assumption is made that each appliance will generate a distribution of transitions which is approximated by a two-dimensional normal distribution. A normal distribution will assign a non-zero probability density to any size transition, so a region of the space in which the probability density is no lower than a specified cut-off level is chosen for assigning transitions to the cluster. Such a region will always have an elliptical shape. When new transitions occur, they are checked to see if they fall within the elliptical region associated with any cluster. If so, they are considered to be transitions of the associated appliance. The choice of the probability density cut-off, or equiva-

lently, the number of standard deviations away from the mean to draw the ellipse, is an interesting one, with a subtle complication that is discussed in Appendix A.

Figure 3-6 provides a summary of the conceptual organization of an appliance representation. Beginning at the top, we assume that out in the real world there is a physical appliance which generates some set of transitions as it turns on and off. The set of transitions can be visualized as a scatter plot in the signature space. Based on a set of transitions, an appliance representation is generated with three components: a main component and two sub-components. Each component is mathematically a multivariate normal distribution which is fit to observed transitions. The main use of such a distribution is to define a decision set for use when analyzing future transitions. The decision set is either an ellipse which contains the most probable portion of the distribution, or else, if the properties of the distribution are not sufficiently clear, a circle. The second use of the distributions is to provide a principled basis for splitting and merging appliance representations, as discussed below. Associated with each appliance representation is a set of statistics learned from the timing of the transitions. These include a count of cycles, average cycle duration, and the average energy usage versus time-of-day profile for weekdays and weekends.

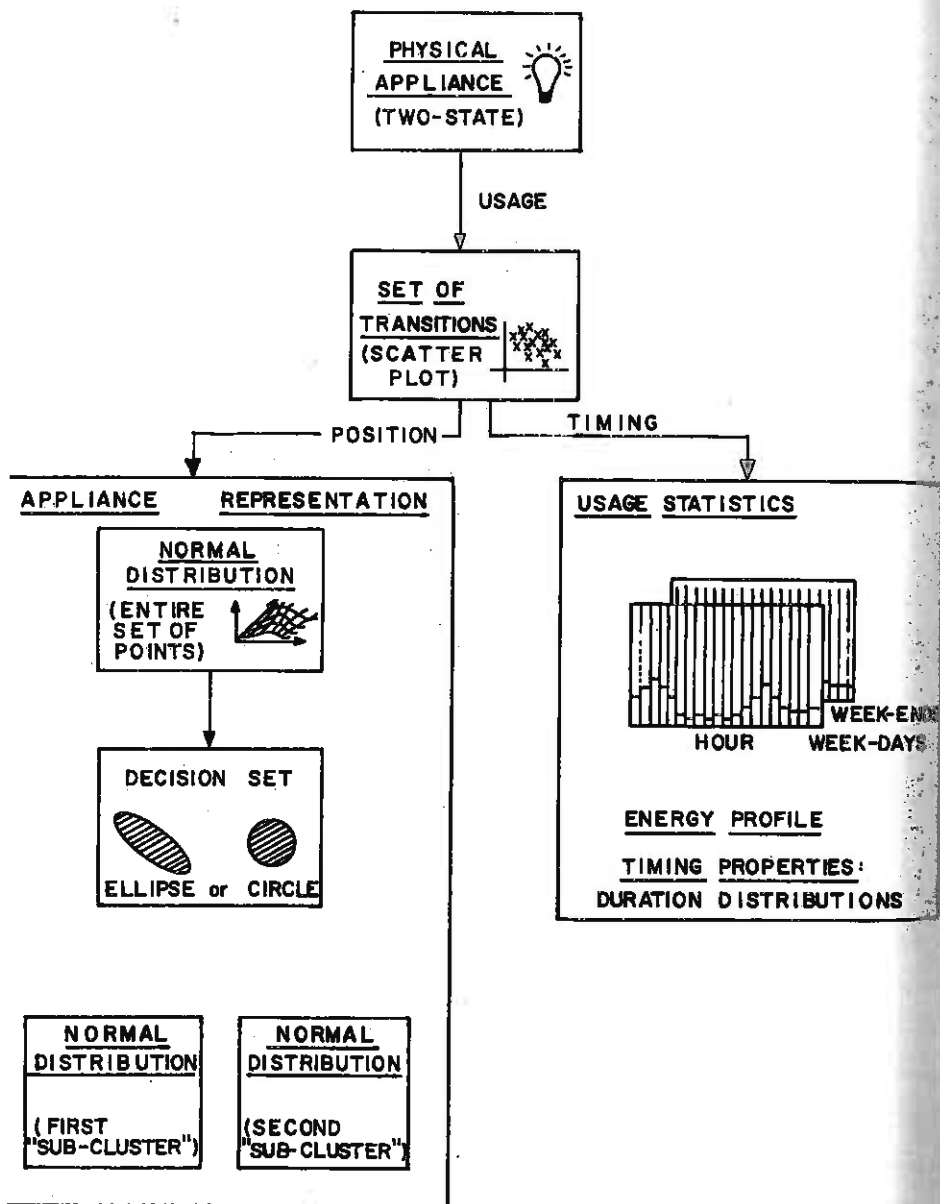


Fig. 3-6. Conceptual Organization of Appliance Representation.

The shape of a cluster is constantly adapted to the transitions observed to fall within it. The coordinates of each new transition are used to update the five parameters which define the cluster. If the new points tend to appear only at the center of the ellipse, the ellipse shrinks in size. If they are frequently near the outskirts of the ellipse, the cluster enlarges. This adaptation to the data is a result of an estimation procedure which chooses the normal distribution which is most likely to have generated the observed points. The update is performed in a recursive manner which does not require storage of all previous transitions. By means of this procedure, initial estimates are constantly refined. The method used involves a finite-memory filter which causes older measurements to be given less weight than recent measurements. By means of this procedure, gradual changes in an appliance's characteristics can be followed.

When a new transition is available to update the representation of an appliance, two independent uses are made of it. The first is that the parameters of the main cluster are updated, as described above. In addition, one of the two sub-clusters is updated in a parallel fashion. In doing this, the load monitor is simultaneously following out the consequences of two mutually exclusive hypotheses. One hypothesis is that there is a single appliance generating all the transitions that have been grouped into the main cluster. The second is that two similar, but statistically separable, appliances are generating the transitions. In following out the consequences of the latter hypothesis, the load monitor tries to update just one of the two appliances. It picks whichever sub-cluster is closer to the new point, and updates the characteristics of only that sub-cluster. In this way, if there really are two appliances, the two sub-clusters

should take on their individual forms unless they overlap excessively.

Initialization of the process begins with a single point. Any observed transition which does not fall into any of the existing appliance clusters is used as the mean of a new cluster. When only one, or even a few, points are in a cluster, it is difficult to estimate its size and shape. The algorithm therefore uses a small fixed radius (20 W/VAR) to define a circle about the mean value. This circle is used as the decision set until there are enough points to use the ellipse method. The first point of a new appliance defines not only the main cluster but one of the sub-clusters. The second point is averaged with the first into the main cluster and also becomes the center of the second sub-cluster. From the third point on, one of the two sub-clusters is selected for updating based on proximity.

An appliance representation can be split into two appliances if the sub-clusters become sufficiently distinct. A split/merge statistical test has been developed which gives a measure of the separation between any pair of clusters. The test is a function of the placement, sizes, and number of observed points in each of the clusters. It determines how likely or unlikely it would be for the two apparent clusters to arise by random processes if a single normal distribution were generating the observed transitions. If the two sub-clusters are rather unlikely to have arisen by chance, they are considered significant, and the cluster is split. The test is repeated on each appliance representation after every twenty-fifth observed cycle. If the test suggests that there are really two appliances present, that appliance entry in the table is cancelled, and instead, two new

appliance entries are created. The main clusters of the new entries are initialized to the sub-clusters of the eliminated entry. New sub-clusters are initialized within each of the new clusters in case they, in turn, need to be split.

The same statistical test is also used when considering whether to join two appliance representations. It is common that an appliance which shows a wide range of starting transitions should initially be learned as two or more separate appliances. As more transitions are observed, the separate clusters grow together into a mass. When the main clusters of two appliance entries overlap, and a transition is observed which is ambiguous because it falls into their common area, they may be joined. The two main clusters are given as argument to the split/join test, and if it reports that they are likely to have been generated by a single normal distribution, the two appliances are merged. Their entries in the appliance table are eliminated, and a new entry is created. The main cluster of the new entry is formed by mathematically combining the main clusters of the two original appliances. These two clusters individually become the sub-clusters of the new appliance in case it should later be decided to split it. The split/join test uses a split threshold and a join threshold with a slight "deadband" between them to provide a "hysteresis" which prevents a borderline appliance from being frequently split and rejoined. The details of the test are given in Appendix D.

The clustering method developed here has a number of advantages over previous methods. The use of "parallel sub-clusters" to allow splitting and merging in a dynamic fashion permits the method to automatically converge upon the appropriate number of clusters. (Many existing clustering methods are only appropriate if the number of

clusters is known.) The split/merge test provides a principled justification for the number of resulting clusters. The fact that the method is implemented in a "dynamic" manner allows it to be applied to large open-ended problems even if only a small amount of computer memory is available. One limitation of the method however, is that the use of the statistical split/merge criterion to guide the splitting and merging is only appropriate where the assumption of a multivariate normal distribution is warranted. Another caution concerning the method is that, unlike most static clustering methods, it is presentation-order sensitive. If the same set of points were presented to the algorithm in a different sequence, a slightly different set of clusters would result.

3.2 Results

We have field tested the current version of the prototype load monitor in three homes. In each of the tests, current transformers were installed on the two service legs inside the home at a point just before the distribution panel. The Digital AC Monitor and HP9845B were placed inside the home, but only had access to information which could be measured outside the home. The three following sections describe the results of testing in these three houses. (The three residences described in the following sections are named after their respective towns.)

Generally speaking, we consider the results to be quite successful. They demonstrate that the nonintrusive appliance load monitor can learn the electrical properties of the major appliances, and keep track of their ON/OFF behavior and energy consumption. Where the load monitor has made errors, they do not result from insurmountable problems, but instead point to ways to make improvements. Our planned refinements to the load monitor, based on the results of the next three sections, are listed in Section 3.3.

In each house, we have performed two types of tests. The primary test is to let the monitor run for a week or two of normal appliance usage and then verify that the appliances it reports on truly exist. We have also manually switched individual appliances on and off to test that the load monitor is indeed processing each transition correctly when we "look over its shoulder." It is difficult to quantify its performance meaningfully over long periods of time however. The problem is that we lack an independent measurement of how much energy is consumed by each appliance during the test period. In order to make a quantified assessment of the load monitor's performance, we

would like to run field tests in homes in which utility load monitoring equipment is already in place. A plan for this additional testing is given in Section 6.2.

The home which has undergone the most extensive testing and in which the highest success rates have occurred is the first house described below, which is the home of the author. It is in this house that primary appliance data was first collected and the algorithms were first developed and tested. Accordingly, there is a concern that those results have come about because the algorithms or parameter settings in the load monitor are in some manner tailored specifically to this house. It is primarily for this reason that the second home was selected. The third house was selected so that the performance of the load monitor could be assessed on two important appliance targets not present in the first two homes: an electric water heater and central heat-pump air conditioning.

3.2.1 Natick House

In the one-week test which we consider most representative, the load monitor learned 45 appliance representations. Of these, only 20 contain a significant number of cycles or a significant amount of energy. These more significant appliance representations are plotted in Figure 3-7 and listed in Table 3-1. Only the main clusters are printed and plotted from each appliance representation. The appliance names labeling each cluster were provided by the author. The load monitor does not yet identify appliances.

The insignificant appliance representations which are omitted from the figure and table are generated by rarely used appliances, simultaneous transitions, and program weaknesses. In some cases we

can identify an appliance, used only once or twice, which was correctly learned. The remaining clusters may be components of appliances which we do not fully understand (such as defrost cycles, etc.) but are most likely attributable to errors. These errors are most likely the occasional inappropriate matches of stray transitions left over from multi-state appliances and simultaneous transitions. The fact that these rarely occur leads to few cycles being counted, with little totalized energy. The clusters selected as significant for discussion below are those which cycled on and off at least four times, or which cycled fewer times, but consumed at least one kWh of energy. Appliance representations which cycled fewer than four times, and consumed less than one kWh of energy are simply ignored.

APPLIANCE	#	LEG	REAL	REAC.	σ_{xx}^2	σ_{yy}^2	σ_{xy}^2	CYCLES	MINUTES	KWH
LIGHTING	1	1	76	-1	28	6	-4	9	36.3	.4
WATER_BURNER	3	2	247	559	64	13	4	134	8.7	4.7
BATHROOM_LIGHT	4	2	144	1	28	6	-2	39	23.5	2.2
AIR_DRIER	6	2	1076	24	34	6	3	10	1.8	.3
WALL_BURNR_ERR	9	1	686	-5	41	5	-5	9	43.5	4.5
UNKNOWN	10	1	753	-32	25	4	4	0	3.9	.4
BURNER_ERROR	15	3	420	-5	20	1	1	14	1.2	.1
WATER_BROIL	17	3	1791	-21	48	2	4	11	1.1	.4
WASTER_OVEN_BR	23	2	763	-3	17	4	2	15	1.1	.2
BRIER_ERROR	30	2	2673	-15	8	0	0	1	36.0	1.6
WASHING_MACHINE	39	2	1099	233	51	4	4	16	1.0	.3
WATER_PON	50	2	993	-7	18	3	-3	45	.5	.4
WALL_BURNER	52	3	685	-2	17	6	1	517	2.0	11.8
WATER_BRIER	55	3	2640	-24	131	10	-4	64	1.7	4.9
LARGE_BRNR_&_BAKE	58	3	1103	-9	49	2	-1	190	.2	.8
FRIGERATOR	63	2	267	265	35	4	-5	308	11.2	15.2
WASTER_OVEN_BK	64	2	1481	-9	80	13	-4	52	.5	.6
WATER_GHTS	65	2	91	-0	83	47	11	10	165.4	2.6
UNKNOWN	66	2	198	-8	0	0	0	1	320.9	1.0
FRIGERATOR	67	2	247	263	38	1	5	2	194.0	1.5

Table 3-1. Appliances Learned at Natick House.

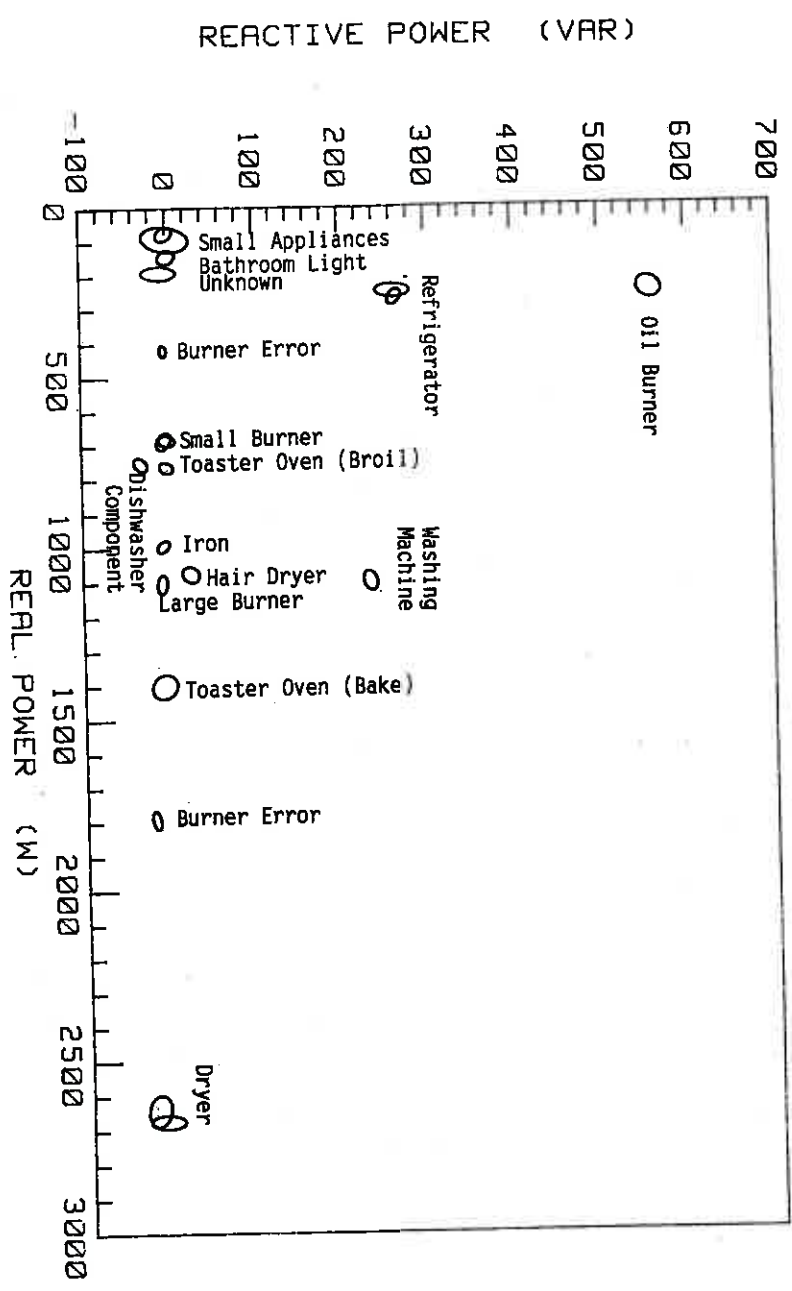


Fig. 3-7. Signature Space at Natick House.

The vast majority of the clusters of Table 3-1 and Figure 3-7 were learned correctly by the load monitor. There is a one-to-one relationship between these clusters and the indicated appliances. As such, there is little to be said about them. The discussion below therefore focuses on the errors in the load monitor's conclusions. From these we see how to continue improving its operation.

Lights and small appliances tend to be fused together into a cluster just above the noise threshold, which is 70 W for this data. The numerous lights and small appliances in the 70 to 120 W range which can not be resolved individually by the load monitor are visible as two clusters (one for each leg) in the lower left of the figure. This is not an error so much as an indication of the limits of the device. It can not discriminate the many small appliances in the load. The threshold for discrimination appears to be approximately 120 W for this residence.

During this one-week test the load monitor detected 3564 transitions which were large enough to process (i.e. with real power greater than 70 W). This is an average of about one every three minutes. Of those, 83.5 percent were matched into pairs. The remaining unpairable transitions were eventually thrown out of the working buffer in the cleaning process. We believe that the majority of the unmatched transitions were the result of multi-state appliances or simultaneous transitions. In many cases however, transitions which should have been paired were not paired because they fell just beyond the 35 W or 3.5% threshold for matching which is described in Section 3.1.3. This occurred mostly with the refrigerator, which shows a wide range of turn-on transitions. Widening the threshold is one way to fix this problem. A more general solution, as discussed in Section 3.1.3, is

to reverse the current sequence of matching and then clustering. If ON and OFF transitions were clustered separately, the outlying ON transitions would be understood to be part of a relatively large cluster, which could be matched to the tight cluster of OFF transitions. This improvement, which will be made in the next version of the load monitor, should improve its performance.

The refrigerator is visible in Figure 3-7 and Table 3-1 as two separate clusters. Note that the second of these contains only two cycles while the first contains 308. The smaller cluster contains two cycles in which the turn-on transition was relatively low, about 20 W less than the other transitions. Eventually these clusters should merge into a single appliance representation. The larger of the two already is the result of the cluster joining procedure. The energy consumption reported for the smaller cluster is completely incorrect however. This is because the cycle duration is too long--over three hours. This error came about by the load monitor matching an ON transition of one cycle with the OFF transition of a later cycle. The forthcoming improvements to the ON/OFF matching algorithm, described above in Section 3.1.3 should eliminate this type of error.

Note that the figure superimposes decision set ellipses from both 120 V legs and from 240 V appliances. In the case of the dryer and the small burner, the two clusters shown for each are actually not overlapping. In both cases, a weakness of the current load monitor algorithm has caused it to erroneously learn a 120 V appliance and a 240 V appliance when in fact there is only a 240 V appliance present. The current version of the software uses a two-and-one-half-dimensional transition representation which is described in Section

3.1.1. This format suffers from the weakness that it can not represent unbalanced loads. The software is currently set to arbitrarily interpret an unbalanced transition as two separate transitions, one on each leg, which happened to occur simultaneously. (When the two-and-one-half-dimensional conversion routine receives an unbalanced transition, it simply splits it into two 120-V transitions which are placed in separate entries of the prebuffer.) The consequence of this procedure is that when small appliances happen to turn on or off simultaneously with the dryer element or small burner, the sum is seen by the load monitor as an unbalanced transition which is erroneously split legwise. Usually these half-appliances can not be matched to anything and eventually are removed from the buffer. But as a result of this happening twice, once to the initial ON transition of the dryer, which includes the 120 V motor, and then a half hour later to an OFF transition, the load monitor found a pair and incorrectly determined that a large 120 V appliance cycled once. This accounts for the 120 V cluster which overlaps the true 240 V dryer cluster in the lower right corner of Figure 3-7. An analogous pair of overlapped clusters were learned for the small burner. Switching to a four-dimensional format in the next generation load monitor should rectify this type of problem. In the case of the drier however, the multi-state machine algorithm will be needed to fully analyze its transitions because the heater and motor are always switched on together initially but then switch separately.

A separate type of error with the small and large burners leads to the 420 W cluster labeled "burner error." The control circuit in the stove causes burners to be independently switched on and off with a period between 4 and 20 seconds, depending on the heat setting.

These frequent transitions provide opportunity for many types of simultaneous transitions. The splitting problem described in the previous paragraph happens when the burner switches simultaneously with a 120 V appliance. A separate problem occurs when two burners happen to switch simultaneously. This will generate a spurious 240 V transition which could be analyzed by the method of Section 2.6. As with the previous problem, if it happens rarely it will have no matching transition and will eventually be removed from the buffer. With two burners however, it has happened frequently enough that matching pairs of simultaneous transitions can sometimes be found. If the large burner turns on as the small burner turns off, a 420 W net transition occurs. This is occasionally matched with a -420 W transition that occurs when the large burner turns off as the small burner turns on. This happened 14 times in the week in question, all within a one-hour interval of cooking. By an analogous process, a 1790 W transition occurs when the burners switch "in phase." There is also a cluster at this power level in the table, but it is not certain to what degree the two burners account for it. This is because the broil element of the stove happens also to generate a 1790 W transition. If it were not for the broil element, the decomposition method for simultaneous transitions would be able to correctly handle both of the above problems. With the broil element present, it is likely to introduce a minor error. It would erroneously think that the broil element was actually two of the burners switching simultaneously. This does not present a major problem in this residence however, because the energy use of all of these elements will be added together under the heading "cooking." The exact allocation of energy to the

separate elements is not crucial.

A second confusion between cooking elements happened with the large burners and the bake element of the oven. The two large burners are 1180 and 1155 W. The bake element is 1125. These are close enough that they all fall within a single cluster. Perhaps the splitting algorithm could eventually separate them but this is not important if cooking is a single load category. If separate clusters were identified, their energy parameters would simply be added together. Similar comments apply to the two small burners which are too similar to resolve.

A somewhat mysterious appliance is the cluster labeled "dishwasher component." The load monitor detected eight cycles of a 120 V appliance drawing 750 W and -30 VAR. We have ascertained that it is some component of the dishwasher, but we do not understand what component would have a capacitive power factor.

In the course of learning the 45 active appliances, the load monitor also generated 22 appliance representations which were later joined in various combinations. This accounts for the enumeration to 67 in the listing of Table 3-1. As far as we can tell, the joining mechanism always operated correctly. Seven of the listed appliance clusters were formed by joining earlier clusters. (Several of those earlier clusters were the result of joining even earlier clusters.) The joining algorithm has been tested repeatedly by feeding it two clusters which are known to be associated with different appliances, and verifying that it does not recommend merging them. Conversely, it has been tested by feeding it the two subclusters of appliance representations which are known to belong to a single appliance and verifying that it does recommend merger.

The properties of the ellipses discussed above, in particular the nearly one-to-one relationship between clusters and major two-state appliances, show that the appliance-learning aspects of the load monitor are performing well. Although this shows that the load monitor is doing a great deal of learning correctly, an analysis of the signature space can not show whether or not the load monitor is tabulating energy properly. The success rate for energy analysis can be estimated by looking at the performance of the load monitor in the time domain, rather than in the signature space domain.

Almost all errors made by the load monitor are of a conservative nature. It may miss a pairing because the ON and OFF transitions are not sufficiently similar, but it rarely introduces an erroneous pair. It therefore underestimates the energy consumption of each appliance. (The erroneous pairings discussed in Section 3.1.3 which occasionally introduce an energy overestimation will, we believe, be corrected by methods discussed there.) Based on time histories presented below, we believe the load monitor is currently matching between 75% and 90% of the ON/OFF pairs that occur. The energy consumption which is reported for each appliance should therefore be, on the average, between 75 and 90% of the correct value. The exact values vary from appliance to appliance. (With the dryer for example the energy is greatly underestimated, because the missed pairs and their on-time are not independent. The initial ON transition includes the motor start, and is never paired with an OFF transition, and so becomes lost. The first cycle is by far the longest however, due to the thermostatic control, so the bulk of the energy is not recognized. The multi-state algorithm will probably be needed to properly analyze the dryer.)

To quantify energy analysis precisely, without having to build an extensive data system to independently monitor each appliance in the house, we would like to compare the nonintrusive load monitor with measurements made where other (intrusive) load monitoring equipment is already in place, as described in Section 6-2. The load monitor to be used will incorporate various improvements which should significantly improve its performance.

We can get a pretty good idea of the load monitor's accuracy by looking at an appliance with a fairly predictable ON/OFF history. We know from direct observation that the refrigerator cycles approximately twice per hour, so if the load monitor indicates otherwise, we can be fairly sure that it is in error. Figure 3-8 shows the load monitor's analysis of the refrigerator, in the time domain. The upper plot shows the load monitor's assessment of the ON/OFF history for the week. The horizontal lines indicate the times when the refrigerator is believed to be on. The ticks above the horizontal line show the ON transitions and the ticks below the line show the OFF transitions. When an isolated tick appears, it means the load monitor did not find its mate to pair into a cycle. From the missing ticks, and occasional missing cycles, we count that the load monitor located over 89% of the refrigerator cycles. Because the missing cycles are almost certainly uncorrelated with the length of the cycle we can also report that the load monitor located about 89% of the energy consumed by the refrigerator. Incidentally, the long cycles that occur at 20-hour intervals are not errors, they are "defrost cycles."

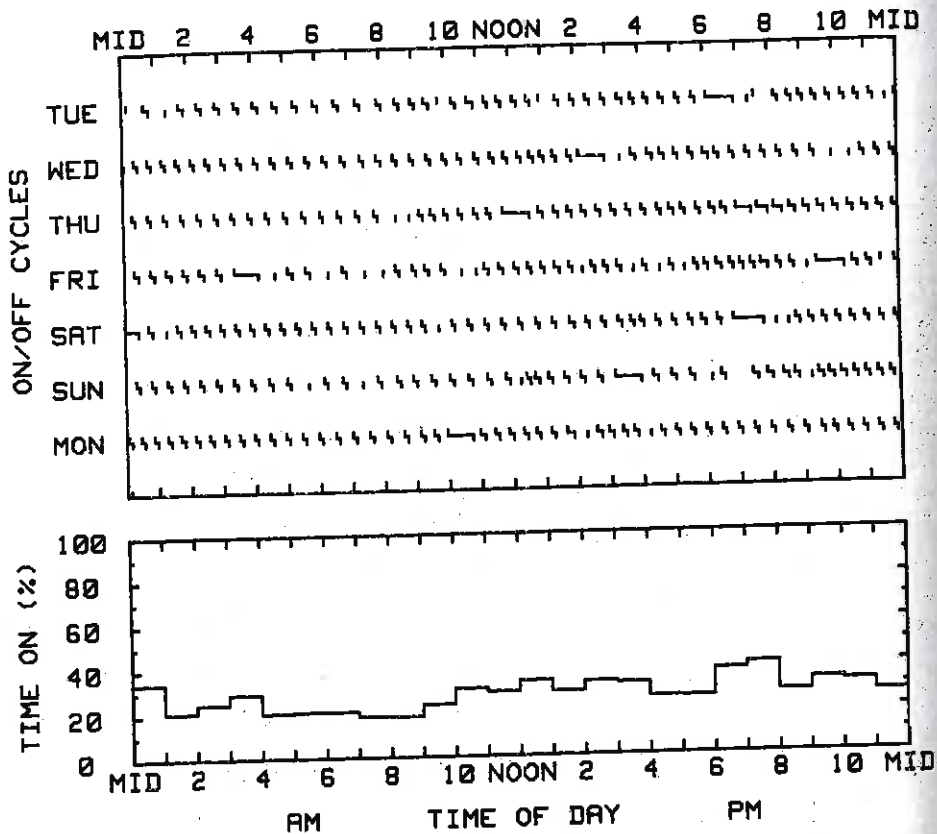


Fig. 3-8. Refrigerator Cycles and Energy Profile at Natick House.

The lower half of Figure 3-8 indicates the energy demand profile versus time-of-day for this one-week period. It actually shows the percentage of time which the appliance is on, which is more directly comparable between appliances than energy. Energy, for load research purposes, is easily calculated from this plot given the operating power level of the appliance. This is the format in which load data will be output at monthly intervals. We expect that the curve would be separated into two components: one for weekday demand and one for weekend demand. Although a week is not a long enough time for the

profile to "smooth out," there is an interesting characteristic that can be seen in this figure. All three refrigerators (this one and the ones to be presented below from the other two monitored houses) show a pattern in which the minimum demand occurs between 4:00 and 6:00 in the morning and the maximum occurs around 6:00 in the evening. This certainly makes sense if one considers when the door is usually opened.

A caveat must be stated concerning Figure 3-8, and others of the same form below. The timing information shown in the upper plot, which indicates the date and time of each transition, is exactly the type of information which is discussed in Section 1 as an invasion of privacy to the occupants. Accordingly, the load monitor does not normally store or output this information. It is reproduced here by a separate program which reconstructs the load monitor's analysis for evaluation purposes. To do this it uses the cluster information that the load monitor learned by the end of the week, and replays the week's transitions to see which fall within the decision ellipse. As such, the decision set used to reconstruct the cycles is static, while the decision set used by the load monitor during its analysis was dynamic. This will result in occasional differences, especially near the beginning of the week when the load monitor did not have much data with which to estimate the ellipses. It would be more proper to state that Figure 3-8 shows what we expect the load monitor would do during its second week of operation if the first week of data happened to be repeated.

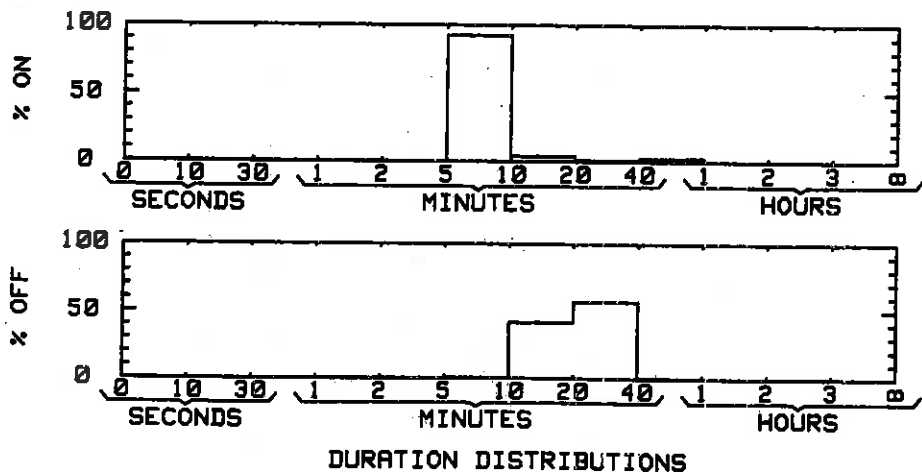
A second, and more subtle, difference between Figure 3-8 and the actual performance of the load monitor involves the finiteness of the working buffer. Old ON transitions in the load monitor's working

buffer are occasionally thrown out to make space before the matching OFF occurs, while the analysis reconstruction program used to generate Figure 3-8 assumes the working buffer of the load monitor is infinitely long and never overflows. We have approximately simulated the finite buffer by not reconstructing cycles of over ten hours duration. The algorithm in the actual load monitor may result in a slightly different matching rate, but we do not think this makes a significant difference in overall performance.

Another view of the load monitor's performance can be seen by looking at its results in the frequency domain. Figure 3-9 indicates the ON/OFF behavior of the refrigerator in terms of the lengths of the observed cycles. The upper plot tabulates, in a histogram form, the length of time that the refrigerator is on. It indicates that 90% of the time it is on for a 5 to 10 minute cycle, while 10% of the time it is on for a period between 10 minutes and 1 hour. It is never on for less than 5 minutes or more than 1 hour at a time. The lower plot of the figure is an analogous histogram of the OFF duration. The refrigerator is never off for more than 40 minutes or less than 10 minutes. These two plots present information which should be very useful for the identification algorithm when it becomes time to name the appliance. The boundary points separating duration intervals were selected somewhat arbitrarily here. Further work on identification may produce a more discriminating set of categorizing intervals.

Appendix E contains ON/OFF cycle plots, energy profiles, and ON/OFF duration histograms, analogous to those given here, for all the major two-state appliances in the house. Some of these deserve special comment. The oil burner provides hot water and usually comes on for a long cycle shortly after 7:00 AM, due to demand. It also

provides space heat, controlled by a day/night thermostat that switches around 7:00 AM. This data was collected in April, so many of the longer cycles are heating cycles. We are not sure what the dishwasher component is that is shown. It is the one mentioned above with a capacitive power factor. The washing machine motor is observed poorly because its start-up transitions are inconsistent. Compare it with the dryer to see how many cycles are missed. The dryer is missing the initial long cycle of each set of cycles for reasons mentioned above. See the iron for an example of thermostatic start-up. The first cycle after a long OFF period is much longer than the average cycle length. This may be a useful identification parameter. The long cycles of the burners are not thermostatic. The duty cycle controller leaves the burner on continuously if the switch is set to "high"; on any other setting it cycles with a period between 4 and 20 seconds. The OFF-duration histogram for the iron and burners show mostly short OFF periods. This indicates the "bunching together" of the cycles, which should be a useful identification parameter.



DURATION DISTRIBUTIONS

Fig. 3-9. Refrigerator ON Duration and OFF Duration at Natick House.

3.2.2 Lincoln House

The second house monitored contains significantly more appliances than the first. It is also a bit more active, with an average of slightly over one transition every three minutes during the 12-day monitoring period. For clarity, the plot of its signature space are separated into two portions. Figure 3-10a shows the appliances on one 120 V leg, and Figure 3-10b, the other. Table 3-2 lists the appliances. The load monitor learned no 240 V appliances for the residence.

APP				σ^2_{xx}	σ^2_{yy}	σ^2_{xy}	JTES		
WATER BED	10	1	347	-0	34	13	3	351	16.9
DISHWASHER COMP.	17	1	710	-70	24	34	4	7	3.6
DISHWASHER COMP.	22	1	684	-74	23	5	-2	7	6.1
DISHWASHER COMP.	24	1	763	-71	48	6	-4	7	6.5
HAIR DRYER	27	1	1245	-8	53	14	8	25	1.6
DISHWASHER COMP.	50	1	731	-71	38	23	6	6	4.4
UNKNOWN	70	1	187	-3	38	9	2	52	5.9
OVEN	73	1	2437	-32	346	106	89	187	1.7
UNKNOWN	77	1	105	3	894	102	195	247	18.9
REFRIGERATOR	79	1	180	241	53	19	11	352	26.8
DISHWASHER HEATER	81	1	1134	23	48	8	-9	8	16.6
FOOD PROCESSOR	90	1	100	-	-	23	16	7	.8
TOASTER OVEN	98	1				10	0	87	1.0
FREEZER	11	2				10	3	307	23.9
1200 W	32	2				30	-8	7	9.3
1200 W	34	2	1280	16	175	109	-74	11	.6
BASEMENT LIGHTS	37	2	408	136	39	99	40	169	3.8
KITCHEN LIGHTS	41	2	146	185	9	52	8	37	13.4
HOT TUB PUMP	43	2	705	464	108	34	-26	7	1.0
UNKNOWN	48	2	435	-27	65	26	4	23	6.5
IRON	60	2	922	-48	37	16	-6	31	.4
HOT TUB PUMP	61	2	688	494	55	12	-10	7	1.0
UNKNOWN	63	2	428	-61	60	17	-13	7	1.0
200 W	66	2	1261	72	100	97	-75	12	.3
UNKNOWN	75	2	82	124	5	44	-1	6	19.3
200 W	82	2	1242	5	34	30	1	7	.6
200 W	83	2	1175	13	11	7	6	8	.3
200 W	84	2	1226	50	259	135	130	11	26.1
200 W	87	2	1145	-11	59	2	2	6	.2
200 W	88	2	1169	-14	16	7	7	13	.2
HOT TUB PUMP	93	2	675	522	310	75	-138	233	1.0
WEN	95	2	1452	-3	150	50	-1	113	2.6
HALL	--	--	--	--	--	--	--	--	14.1

Table 3-2. Appliances Learned at Lincoln House.

SIGNATURE SPACE
LINCOLN HOUSE (1)

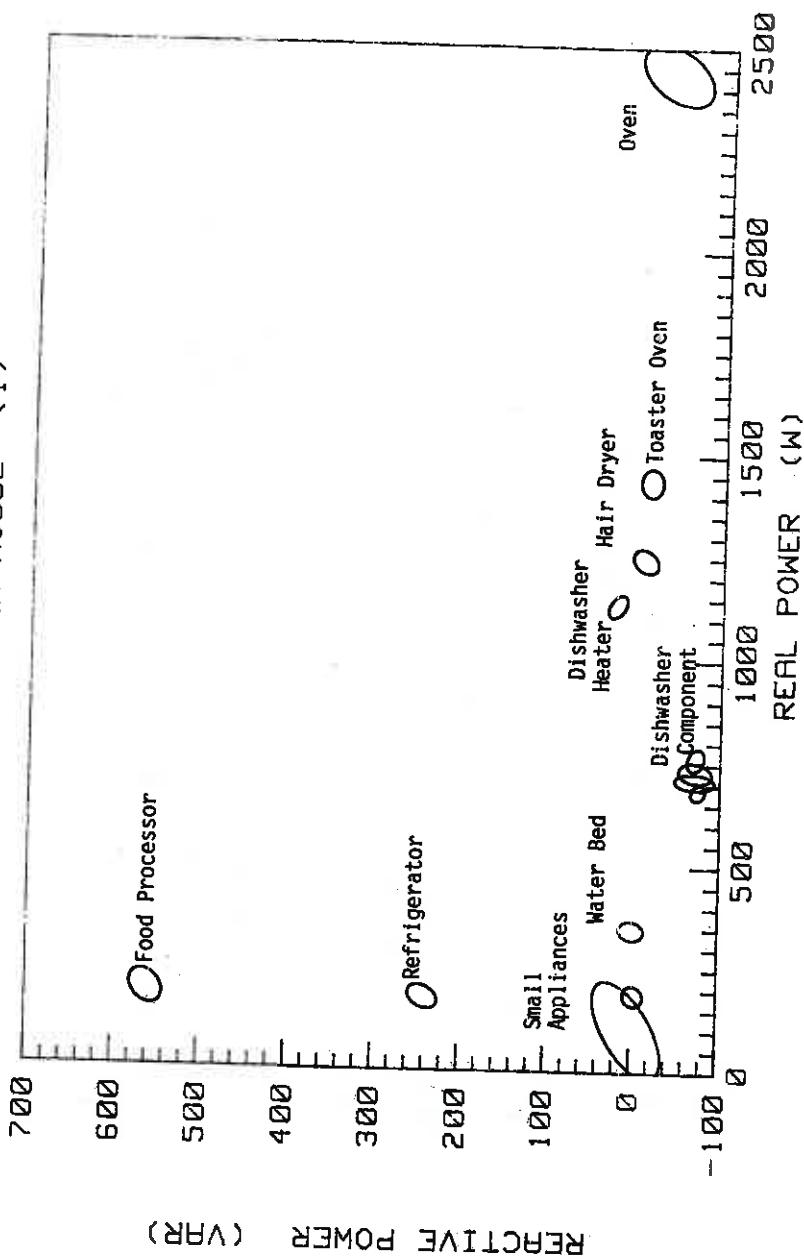


Fig. 3-10a. Signature Space at Lincoln House.

STATIONING OF LINE
LINCOLN HOUSE (2)

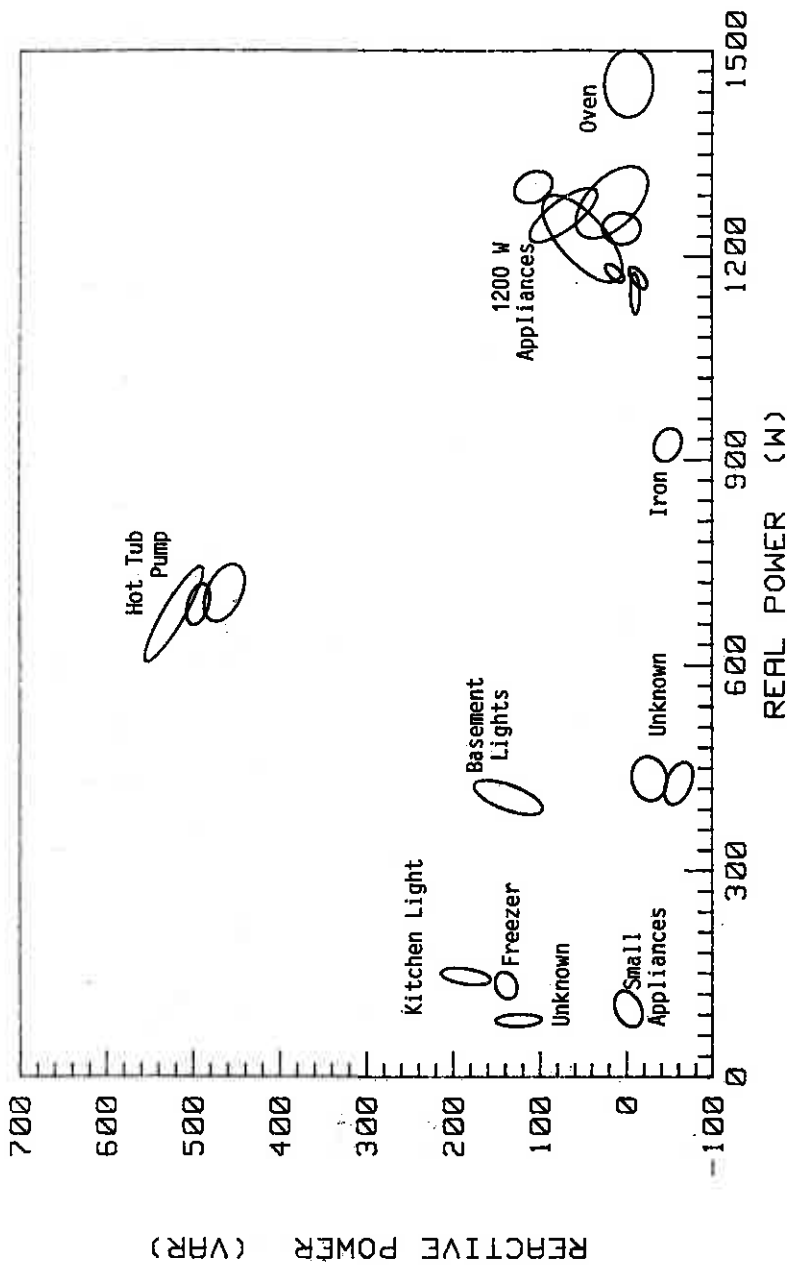


Fig. 3-10b. Signature Space at Lincoln House.

Several of the clusters have not been identified. Most of those that we can name seem to be correctly analyzed into a single cluster. The hot tub pump and "dishwasher component" seem to be well on the way to completion. They each consist of several overlapping clusters which need to be joined together. Based on the degree of overlap between the clusters, this should happen soon.

The clusters comprising the dishwasher component, like the unknown dishwasher component of Natick House above, display a capacitive power factor. The other dishwasher cluster, labeled "dishwasher heating element," probably consists of a heater in parallel with some other component; this would account for its reactive power consumption. The time plots of Appendix E show that in each dishwasher cycle, the capacitive component is on three times and then the heating element is on once. Other aspects of the dishwasher cycles are being missed by the Load Monitor. For example, the power consumption over time ramps up as the dishwasher fills with water. This also happens in the Natick House dishwasher. These are the only two appliances in which we observed gradual rather than step power changes. If there were many such appliances, the edge detection algorithm might need to be reconsidered. As things stand, the multi-state appliance algorithm is needed to fully learn the dishwasher's properties.

The oven is interesting in that it is the only 240 V appliance in the house and it is significantly unbalanced. When on, it consumes 2440 W on one leg and 1450 W on the other. For reasons discussed above, the load monitor misinterprets its transitions as belonging to two separate 120 V appliances.

The large clusters of Figure 3-10b in the 1200 to 1300 W range are likely to include a number of appliances, including a hair dryer.

Apparently they overlap in their power consumption, and will be joined together shortly. This points out a fundamental limitation of the nonintrusive load monitor: it can not distinguish appliances which are very similar electrically. This is likely to be a problem for many houses when loads in the 1200 W range are considered, because there are so many commercially available appliances in this range. Increasing the number of signature components (e.g. including harmonic currents) may improve its performance in this area, but it is not clear to what extent these appliances need to be resolved for load research purposes.

When we examine the refrigerator in the time domain, as in Figure 3-11, we see that almost 90% of the cycles are analyzed correctly. Appendix E contains additional plots of appliance activity. Of these, several are worth special note. The freezer and hot tub pump are also fairly predictable and can be used to estimate the accuracy of the energy tabulations. The hot tub pump is correctly matched over 96% of the time but the freezer is correct only 78% of the time. We are not sure what accounts for this variation. The hot tub pump is controlled by a timer which turns it on for about a minute every hour to run the filter and prevent water in the pipes from freezing. The two long cycles are due to manual control when it was used. Heat loss to ambient is clearly the driving force behind the water bed heater's rather interesting energy profile. The energy profile of the kitchen light is also rather illuminating, showing a three-meal-per-day pattern. The two "oven halves" are actually the same oven, controlled by a single switch. As mentioned above, the lack of a four-dimensional transition representation causes two separate 120 V

appliances to be learned instead. The close similarity of these two independently learned plots is therefore to be expected, and confirms the load monitor's performance.

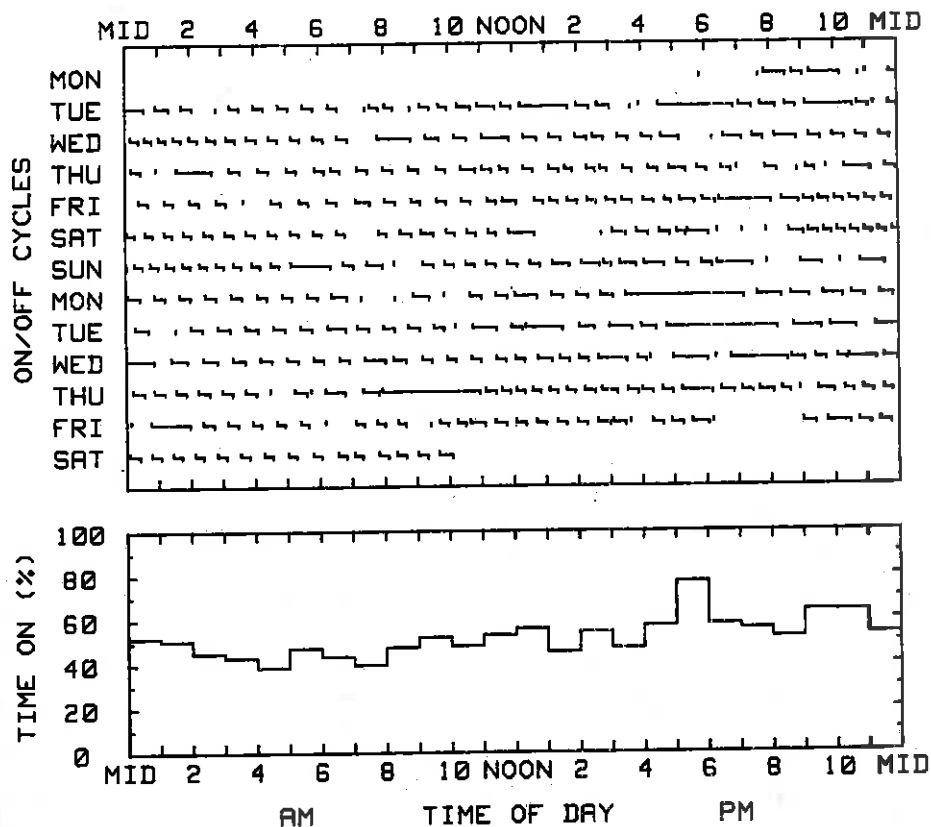


Fig. 3-11. Refrigerator Cycles and Energy Profile at Lincoln House.

3.2.3 Acton House

The third test house is quite large and contains a wide range of appliances. Only half of the house was monitored, however. Two separate 200 A services enter the house and feed two separate distribution panels. Although only one of the two panels was monitored, it is the larger of the two, and services more appliances than the average house. Table 3-3 and Figure 3-12 show the significant appliances learned by the load monitor over a one week period. For clarity, the figure is divided between two plots: 120 V appliances on leg 1 are on the first plot, while 240 V appliances and 120 V appliances from leg 2 are on the other. Again, there are many correct clusterings, in one-to-one relationship with appliances.

APPLIANCE	#	LEG	REAL	REAC.	σ^2_{xx}	σ^2_{yy}	σ^2_{xy}	CYCLES	MINUTES
BEDROOM LIGHT	9	1	183	56	22	4	3	35	2.8
HALL APPLS.	13	1	93	-4	47	20	7	71	6.8
KITCHEN LIGHT	15	2	275	14	13	11	-0	21	5.1
LIVING ROOM	16	1	198	-24	68	30	15	16	2.6
BEDROOM LIGHT	17	1	378	46	13	7	-6	7	4.1
FRIDGE DOOR OPENER	21	2	335	122	38	37	4	7	.2
LIVING ROOM	24	2	137	15	17	11	-10	10	14.2
AIR PUMP (SPLIT)	27	2	685	447	36	23	23	9	7.0
AIR PUMP (SPLIT)	37	2	665	448	52	11	4	6	13.9
BEDROOM LIGHT	46	1	333	17	11	1	1	6	3.5
AIR PUMP (SPLIT)	48	1	788	445	61	15	17	7	13.5
HUMIDIFIER	54	1	690	534	756	19	-66	312	11.5
REFRIGERATOR	57	2	339	448	48	35	-12	108	18.4
BEDROOM I.R.	59	1	284	-4	46	2	-1	30	8.8
CHEST FREEZER	63	1	238	190	78	2	-11	6	50.8
HALL LIGHT	71	1	205	-3	50	3	-0	27	5.9
AIR HEATER	74	3	2224	-20	152	1	-7	23	15.7
HALL	84	1	1841	-8	13	3	1	189	.3
AIR HEATER	88	3	2196	-19	47	1	3	9	10.6
HALL APPLS.	93	2	87	-1	19	47	8	14	-24.0
AIR PUMP	98	3	677	442	71	5	-5	344	.9
REFRIGERATOR	100	1	86	85	29	6	-4	301	9.3

Table 3-3. Appliances Learned at Acton House.

SIGNATURE SPACE ACTON HOUSE (1)

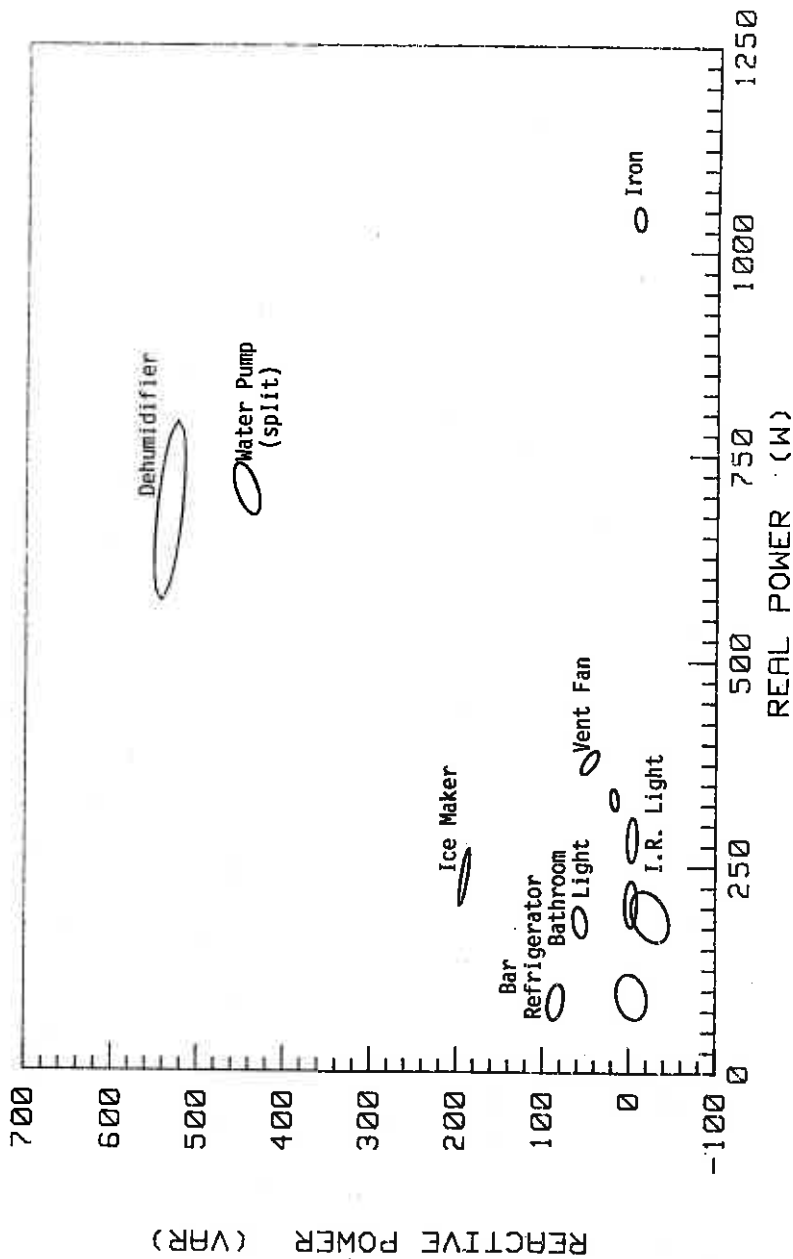


Fig. 3-12a. Signature Space at Acton House.

DIAGNOSTIC SIGNATURE
 ACTON HOUSE (2,3)

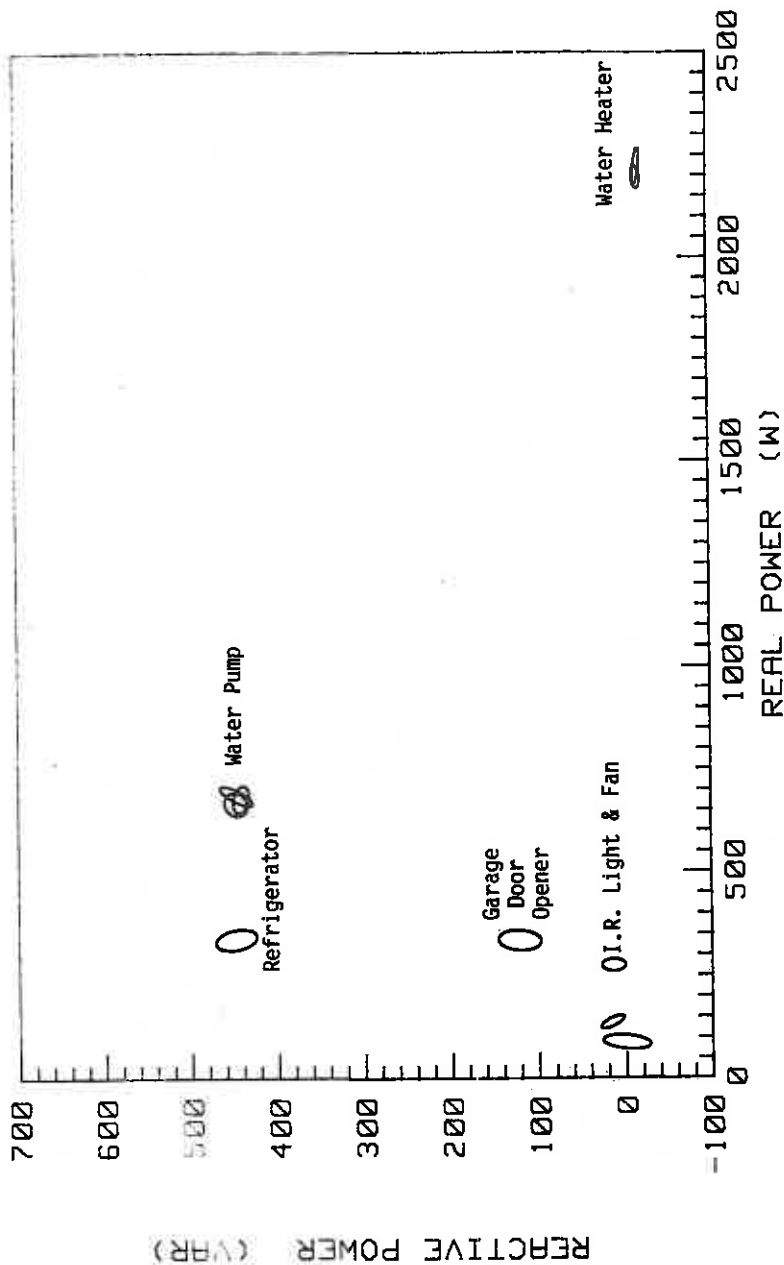


Fig. 3-12b. Signature: Space at Acton House.

The problem of the two-and-one-half dimensional representations has again caused an erroneous splitting of a 240 V appliance into two 120 V appliances. The victim here is the water pump which cycles several times per hour on the average. When it switches on or off simultaneously with a 120 V appliance, an unbalanced transition occurs, analogously to the case of the small burner discussed in Section 3.2.1.

This house was selected in order to test the load monitor on an electric water heater and on central air conditioning. The water heater was learned with great accuracy. It appears that the two clusters of Figure 3-12b and Table 3-3 may correspond to the upper and lower heating elements. We have measured them separately and determined that they differ by approximately 30 W. Because they are so similar it is likely that they will be joined together into a single cluster at some point in the future. They already overlap considerably. It does not appear to matter in this case whether or not this happens however, as their energy consumption will simply be totalled if they are separate.

The load monitor did not have a chance to learn about the heat pump, because it did not turn on during the test period. We are fairly certain however, that it would not have learned it properly. Examination of its operation shows that it is definitely a multi-state machine, with a complex control unit switching a compressor, an indoor fan, an outdoor blower, a backup resistance heater, and relays. Its operation seems well beyond the scope of the two-state algorithms.

Analysis of the refrigerator transitions in the time domain (Figure 3-13) shows an accuracy of about 75 percent. This lower rate

is probably attributable to the greater appliance activity here compared to the two previous houses. The load monitor detected 5364 transitions over 70 W in this week. This comes to about one every two minutes on average. This will result in more simultaneous transitions which can not be analyzed by the current version of the load monitor.

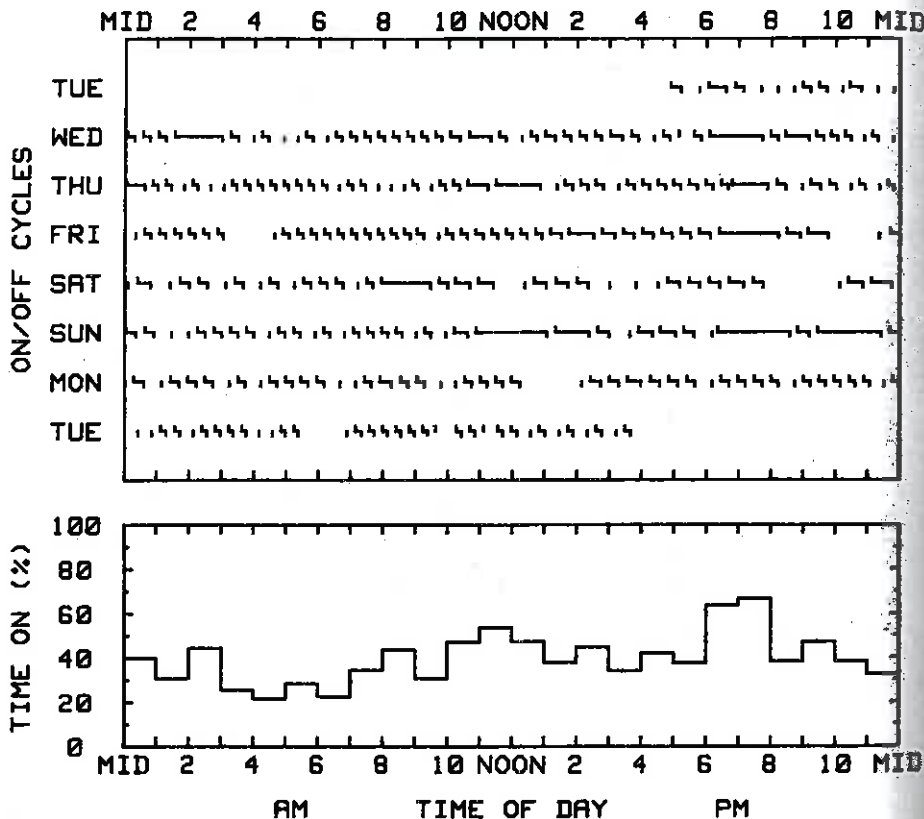


Fig. 3-13. Refrigerator Cycles and Energy Profile at Acton House.

Appendix E contains the load monitor's plots of the major two-state appliances of this house. Most of these should be self-explanatory by this point. The close correlation between the well-

water pump and the hot water heater is worthy of note. Their high usage from 11:00 to noon is attributable to one of the occupants who regularly showers at that time.

3.3 Recommendations

We consider the results of the three field tests above to be quite satisfactory for the current state of development of the load monitor. For most two-state appliances it is performing up to our expectations, but based on these field trials, we have been able to pinpoint certain areas where problems exist. For most of these problems, we feel we know how to adjust its operation and effect improved operation. These modifications will be made and tested in the near future. Specifically, we plan to incorporate the following adjustments before undertaking the multi-house test described in Section 6.2.

- (1) Simultaneous Transition Analysis. As described in Section 2.6, we have already developed and tested algorithms which detect and decompose the spurious transitions which are formed when two appliances change state within the span of two or three measurements. These techniques have not yet been incorporated in the load monitor.
- (2) Faster Sampling. The load monitor currently samples the load at a rate of 1 Hz. By increasing this somewhat (but by less than an order of magnitude) the number of coincidental simultaneous transitions will be reduced. This will ease the burden placed on the simultaneous transition analyzer.
- (3) Four Dimensional Format. Use of true four-vectors to represent transitions will allow unbalanced 240 V appliances to be learned. It will also allow coincidental simultaneous transitions of a 240 V appliance with a 120 V appliance to be analyzed.
- (4) Clustering before Pairing. By clustering transitions before they are matched into ON/OFF pairs, the load monitor will be better able to handle appliances, such as the refrigerator of Section 3.2.1, which show variable transitions. This is also a necessary step towards the multi-state appliance load monitor.
- (5) Unassigned Power Level. As described at the end of Section 3.1.3, it is necessary to keep track of the total power consumption of the residence between transitions, and allocate it to

the appliances which are determined to be operating. With this information, erroneous pairings across intervals when the appliance is off can be avoided.

- (6) Residual Energy Consumption. For load researchers modeling residential energy consumption by totalling the energy from various appliance classes, it will be necessary to know the time-of-day properties of the residual energy due to small appliances, after all the major appliances are accounted for. This is determined from the unassigned power level.
- (7) Multi-State Appliances. We would like the load monitor to learn multi-state appliances properly, or at least not have them cause errors in its analysis of two-state machines. The method by which to do this is not as clear as the above recommendations however. As described in the following section, we continue to address this problem.

The load monitor described in the above sections is limited in that it is not directly applicable to multi-state machines. If a residence contains an appliance such as a dishwasher or electric dryer, with a motor and a heating element, it could not learn its characteristics or energy consumption. The expected behavior of the load monitor depends on the control mechanism of the appliance. If, as is typical with dishwashers, the motor and heating elements generally turn on and off at separate times in the wash cycle, the load monitor should be able to learn their separate characteristics. It would total their energy as two separate appliances, not as one. On the other hand, if, as is typical with dryers, the control mechanism always starts the heating element when the motor starts, the motor will not be learned. The OFF transitions of the motor will never have a matching ON transition. (The heating element will be learned, however, if it cycles independently with a thermostat, but its energy consumption would be underestimated.) These characteristics have been observed at Natick house, described above in Section 3.2.1.

These shortcomings may or may not be fatal to the method, depending on the class of targeted appliances. To improve the performance of future versions of the load monitor, it is necessary to develop methods of automatically learning the properties of multi-state appliances. This is an ongoing research area, as discussed in Section 6.3. In this section we describe the techniques which currently seem to be the most promising. Section 4.1 describes a system for representing multi-state appliances which could be used by a variety of algorithms. Section 4.2 outlines algorithms which can

handle parts of the overall task. We stress that they are still under development and totally untried.

4.1 Representations

In order for an algorithm to learn, manipulate, keep track of, and report on multi-state machines, it must have a mathematical form for representing them. We have spent some time developing a formalism which seems suitable for the purposes of the load monitor. Each appliance can be represented by a finite state machine (FSM) which indicates the possible states that it can be in, and the possible changes of state. Examples of such FSMs are given in Reference [1].

The basic form for representing an n-state FSM is an n-by-n matrix. The ijth element of the array indicates the cluster of transitions which is associated with the transition from state i to state j. As such, it is simply a pointer to an entry in a cluster table similar to that which we currently use for two-state machines. If no transitions have been observed from state i to state j, then the entry would be a "flag" indicating that is is not a legal transition. The cluster table would contain parameters defining the cluster and subclusters along with a list of pointers pointing back to the FSM states which the transitions emanate from. These "back pointers" allow a rapid lookup of all the possible state changes to consider when a transition is to be analyzed. It may also be worthwhile to allow clusters of common simultaneous transitions (such as the 400 W simultaneous burner transition of Section 3.2.1) to be formed with pointers indicating which pair of appliances change. This will allow rapid analysis of simultaneous transitions without having to repeat the search process described in Section 2.6.

One additional function of the representation is to allow only well formed FSMs, which meet various constraints of suitability. For example it must be possible to get from any state of a FSM to any other state in a finite number of transitions; there can be no inaccessible states. This constraint and others are explored in Reference [1]. It is sufficient here to note that the representation which we have in mind allows these constraints to be verified by relatively simple matrix operations. The representation also straightforwardly allows a factoring operation defined below in Section 4.2.4.

We have developed a canonical form which uniquely picks out a single FSM representation matrix from the many which might equivalently describe any given multi-state appliance. Basically, it permutes the row and column entries of the transition matrix into a well defined ordering without affecting the connectivity of the states. In the resulting sequence, the real power component of the states is ordered from lowest to highest. It therefore results in a standard form for each appliance which should simplify the identification procedure. In addition, during the process of canonicalization, additional constraints of the type discussed above are verified.

4.2 Algorithms

The multi-state algorithms, like the two-state algorithms, can be divided conceptually into two parts: the part which learns what appliances are in the house, and the part which keeps track of the appliances after they have been learned. The second part is certainly the simpler of the two. To keep track of appliances it is only necessary to keep track of a "state pointer" for each finite state machine which indicates which state it is in. New transitions are

mined to see which appliance they should be associated with. The responding state pointer is then updated accordingly. Use of a linking buffer analogous to that described in Section 3.1.3 allows a long series of transitions to be stored until a cycle is complete. In the case of multi-state appliances, a cycle will be some path through the machine beginning and ending at the OFF state. (This reduces to an ON/OFF pair in the case of a two-state machine.) By waiting for a complete cycle to be observed before processing the transitions, various types of errors can be reduced. When an ambiguous transition is present, subsequent transitions can be an aid in disambiguation. After keeping track of the state pointer of each appliance, tabulating its energy usage versus time of day is a simple bookkeeping matter.

Although just a sketch, we are confident the ideas above can be expanded into a method which can follow the activities of each appliance, if their structure is known. The more difficult aspects of the multi-state algorithm involve learning the structure of each appliance. We have a number of ideas how to do this, but are less certain of their effectiveness. In particular, we are unsure how they will behave in the presence of "noisy" transition data. We can provide mathematical justification for parts of the algorithms in the ideal situation in which the driving transitions are not "degraded" by the presence of other appliances switching simultaneously. In the less than ideal case that we know will occur, the performance of these algorithms will be affected. It is possible that this will cause the learned appliance representations to grow unstably or otherwise misbehave in an unacceptable manner.

At this point in time, the algorithms have the form of a set of

heuristics that each apply in certain circumstances. Four of these aspects of the overall algorithm seem clear enough to us that we can set them to paper in the following sub-sections. Generally speaking, these four parts of the program would execute in the order in which they are listed below. We repeat that they are totally untried.

4.2.1 Clustering

The transition-forming portion of the multi-state algorithm is identical to that of the two-state algorithm. The same measurements are taken and normalized in the same manner described in Sections 2.1 and 2.2. Transitions are then formed by the edge-detection process of Section 2.3. The first operation which the transitions undergo is clustering. As discussed in Section 3.1.3, clustering must occur before transitions are matched up into ON/OFF pairs, or other more complex cycles, because the proper matching will not be obvious from the transitions alone. Instead, each transition will be examined to see what cluster it belongs to, and associated with that cluster will be a list of appliance representations and states that it may enter into. When a cluster is ambiguous in the sense that its transitions may indicate a state change of more than one appliance, earlier and later transitions in the buffer should be able to provide enough context to determine which appliance changed state if the appliance is already learned. It is less clear at this point how a new ambiguity would be learned.

In what follows below, it is assumed that clustering is carried out optimally, with no ambiguity or error. That is, we assume each cluster of transitions can be part of only one appliance, and there is only one pair of states that the transition can link. Although this will not be the case for some clusters, it is a reasonable beginning

point for developing an algorithm.

4.2.2 Cyclic Factor Analysis

Given a series of observed transitions in the working buffer, the simplest thing which can be done is to perform a two-state analysis. Those transitions which are easily matched into ON/OFF pairs can be processed immediately. Removing them from the buffer then leaves only the more difficult transitions associated with multi-state appliances. Note that the ON/OFF pairing is now an operation on clusters, not individual transitions. Two clusters are sought out whose means are approximately negatives, within some threshold, and which contain approximately the same number of entries. The transitions associated with these clusters are required to be alternating, or nearly so, in the working buffer. If all these conditions are met, the clusters are paired into the ON and OFF transition cluster of a two-state appliance.

By a generalization of this process cyclic finite state machines can be learned. A cyclic finite state machine is one in which the states are linked together in a ring so that every state is entered in sequence in each cycle from OFF to OFF. The ratcheted rotary switch of a three-way light bulb is an example of a cyclic finite state machine. A set of clusters whose mean values add approximately to zero and which contain transitions that occur in the buffer in a cyclic sequence can be sought out. A technique which tabulates the different transitions which occur between successive occurrences of any given transition can identify the cyclic chains of clusters in a relatively simple and rapid manner. If the number of transitions of the clusters match approximately, and the mean values add approxi-

mately to zero, then we can be confident that the cyclic ordering is not merely coincidental. Such clusters are then joined together into a cyclic finite state machine.

4.2.3 Traversal Analysis

The transitions which remain after all cyclic finite state machines are removed from the buffer may belong to the more complex multi-state machines. A method which we call "traversal analysis," because one imagines that one is travelling along the arcs of the machine from state to state, can be used to learn complex appliances with arbitrary state connections. Unfortunately, the method is very sensitive to degraded transition data, so we must assume at this point that transitions have been clustered perfectly with a one-to-one relationship between clusters and actual appliance state changes. Assume further that all the transitions remaining in the working buffer belong to a single appliance. If several multi-state appliances have been used, this can be corrected for by the methods of the next section.

Basically, the traversal analysis procedure is to examine the transitions in sequence, while keeping track of a state pointer that indicates which state the machine is in after each transition. The appliance representation is considered to be partially learned at all times. New data may or may not be used to update it. When a transition occurs, there are two possibilities. Either the state which the algorithm believes the appliance to be in has an arc leading from it that corresponds to the new transition, or it doesn't. If it does, the "recognition" aspect of the algorithm simply updates the state pointer so that the appliance is considered to have changed state. If the transition is unexpected, however, then the "learning" aspect of

the procedure updates the appliance representation so that the new structure is compatible with the entire transition sequence. This will require adding a new state or the merger of two existing states, depending on various conditions which will not be discussed here. Different assumptions regarding the possible structure of appliances lead to slightly different update procedures which will operate differently in certain circumstances. The most suitable conditions have not yet been ascertained.

This portion of the multi-state algorithm is the most crucial, because it appears to be necessary for learning arbitrary finite state machines. It is also the most disconcerting because it seems to be very susceptible to imperfect data. Much further development will be necessary before we have confidence in such a method.

4.2.4 Factoring

If the above procedure is carried out on a set of transitions which arise from two or more independent multi-state machines, a single finite state machine will be learned which must be broken apart into two separate appliance representations. This requires a process which is akin to factoring composite integers into primes. If the two machines operate totally independently, the resulting transitions will be analyzed into a complex FSM which is a product, in an abstract mathematical sense, of the separate FSMs. Reference [1] describes this product/factor relationship in further detail. Given such a product FSM, an algorithm has been developed which can factor it apart into its component FSMs with little difficulty. Complete independence between the component FSMs is unlikely however. Over a limited time span, it is probable that certain combinations of states will not have

occurred. Therefore the factoring process must be developed further so that it can reliably factor composite FSMs even when they change state in a dependent fashion. This work is in progress.

5.0 APPARATUS

The prototype nonintrusive appliance load monitor described in Section 3 is implemented in general purpose hardware which has allowed flexible development and testing, but which is not suitable for a commercial device. This section describes the probable form of a commercial load monitor. Section 5.1 lists the computational requirements of the device while Section 5.2 suggests two alternate physical packages.

5.1 Computational Requirements

Based on our experience with the prototype load monitor in general purpose hardware, we are confident that the necessary algorithms can be satisfactorily implemented in a state-of-the-art microprocessor system design. No unusual speed or memory capability is required. The most difficult aspect of the design might be maintaining reliable operation and sensor accuracy over the extremely wide range of operating temperatures experienced by field equipment.

A likely hardware arrangement is sketched in Figure 5-1. Although the details of the architecture are not worked out, the general characteristics seem fairly clear. The components are described from top to bottom and left to right:

- | | |
|----------------|--|
| Power Supply | Low voltage dc power required by the solid-state electronics is provided by a power supply connected to the utility side of one of the 120 V ac lines. |
| Backup Battery | Key components of the system are powered by a rechargeable battery. In the event of power failure the load monitor does not continue all of its normal functions (because there will be no appliance usage to monitor). Information describing the appliances learned up to that point would be maintained in RAM however, and the clock/calendar would continue to keep time. |
| Sensors | Current transformers provide a signal proportional to the current flow in each of the two leg circuits. One of the leg |

voltages is monitored. Signal conditioning circuitry isolates and scales these ac signals as necessary for interfacing with the digital circuitry. A temperature sensor (not shown in the figure) for measuring ambient temperature or temperature-humidity index will provide data for correlating with appliance usage.

A/D
Conv.

Analog to digital conversion circuitry is used to sample the ac current and voltage waveforms. (The microprocessor logic will calculate the RMS voltage, and real and reactive power of each leg, using the techniques developed for the Digital AC Monitor [1].)

ROM

Read-only memory is provided with all the software necessary for the algorithm. We expect this might require on the order of 10 to 20 K bytes of ROM.

RAM

Random-access memory is used to store the tables of appliance characteristics and energy usage along with all other working quantities. We expect this might require on the order of 100 K bytes of RAM. Battery backup obviates the need to relearn this information in the event of power failure.

Clock/
Calendar

A real-time clock/calendar is necessary so that the energy consumption of each appliance can be categorized by time of day and weekday/weekend. The clock is maintained by battery backup so that the correct time is available after a power failure. (Additional clocking is required, but not shown, for controlling the data sampling intervals.)

Serial
Number

Each load monitor will be provided with a unique serial number, which is made available to the software for tagging its output. Thereby, the information from distinct load monitors is not confused at the central processing facility.

Modem or
I/O Port

Output could be made in one of two manners (see below). If a telephone link is desired, a modem would provide the interface between the load monitor and the telephone line. If the meter-reading option is preferred, an input/output port would be used at monthly intervals. In addition to its output use, the telephone link or I/O port would be used for synchronizing the real-time clock.

Micro-
processor

The main processing functions would be carried out by a microprocessor which controls all of the other hardware in accordance with the program read from ROM. At this point there appears to be a great deal of flexibility in selecting a microprocessor family for designing the system.

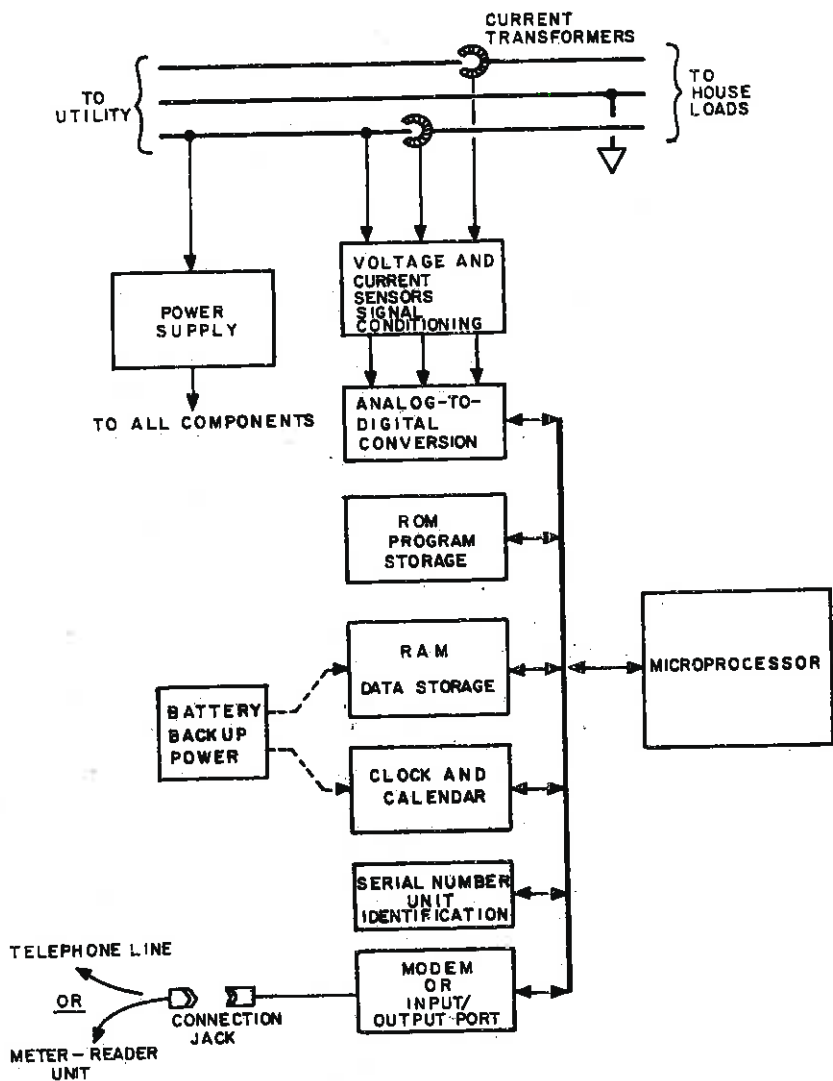


Fig. 5-1. Computer Architecture.

5.2 Mounting

Based on the above computational requirements, we expect that the entire load monitor could be made to fit in a space half the volume of a conventional kilowatt-hour meter. Two mounting scenarios seem likely.

The first mounting technique is to place the hardware in a kWh meter extension collar as shown in Figure 5-2. All of the hardware of Figure 5-1 could be designed to fit in a collar such as those manufactured by the Ekstrom Corporation. The current transformers would be built in, providing the sensor readings. This package would provide the quickest installation and removal time, and allows the kWh meter to remain in place for revenue purposes.

Monthly output of data could easily be arranged in one of two ways. If a telephone line were brought to the meter, a computer at the central load research facility could be programmed to automatically call up the load monitor at monthly intervals to read its accumulated data. This option allows for completely automatic data gathering at the the cost of installing a telephone line out to the house, and monthly phone bills. The second output option is to have a connection plug built into the side of the load monitor which requires monthly access. A portable storage unit could be carried by meter readers from house to house and plugged into the load monitor. In a second or so the month's worth of data could be transferred into the portable unit which would later be fed into the central computer. A briefcase-sized unit could be designed which would hold data from several hundred residences. (We have not pursued a third option for data retrieval, which is to use power line carrier techniques.)

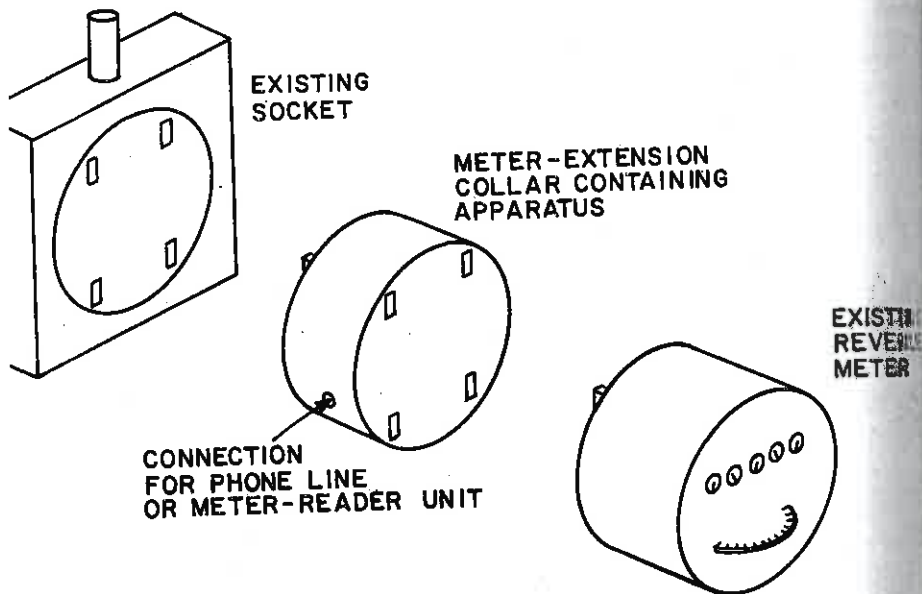


Fig. 5-2. Collar Mounted Load Monitor.

A second mounting option exists which also allows for the above data retrieval techniques. This option is to place the load monitor in a box which would be placed on a utility pole with a pole transformer as in Figure 5-3. From this point there would be ready access to several houses from one instrument. A single load monitor, built as in Section 5.1, but with proportionally more RAM memory, could be designed to multiplex between four or more houses. Current transformers would be placed on the secondaries to each of the houses served by the transformer, and connected, along with a single voltage signal, to the load monitor. Although pole mounting would require the use of a line truck and more installation time than the kWh meter socket technique, this could be offset by the equipment savings brought about by multiplexing. The hardware of Figure 5-1 (with the exception of RAM) would be shared by several residences, greatly reducing the monitoring cost per house. Pole mounting would also provide easier access to a telephone line for data retrieval. If the direct access option were preferred, this could also be arranged at the pole. An additional virtue of the pole mounting is that the monitor could be installed without any service interruption. In contrast, installing the socket mounted monitor causes a short power loss to the occupants.

We are not certain how the temperature sensor would best be mounted in the collar-mount scenario. It seems that any arrangement in which the sensor is attached directly to the monitor will be susceptible to microclimate problems. For example the kWh meter socket, and therefore the sensor, might be shaded at certain times of the day even though the rest of the house is constantly in the sun.

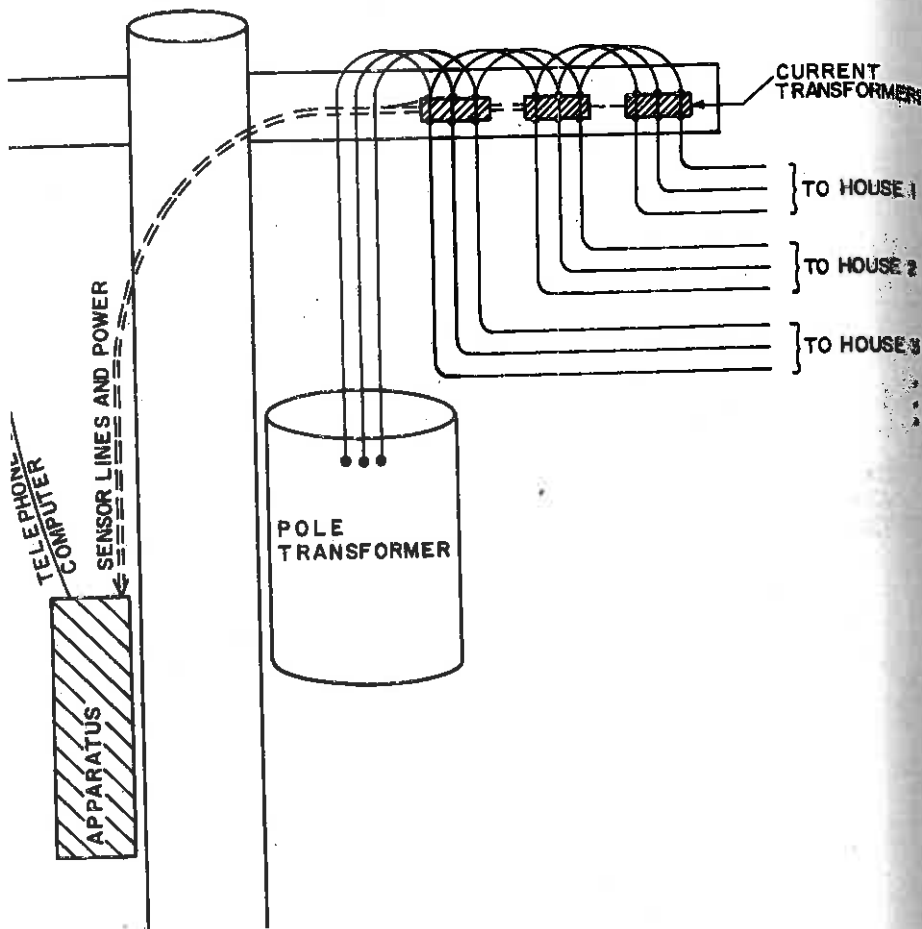


Fig. 5-3. Pole Mounted Load Monitor for Multi-House Monitoring.

If the sensor were attached to the end of a connecting cable, it could be placed in a more favorable location. This would require running a cable along the outside of the house to be monitored, however, which would probably be unacceptable. For the pole-mount scenario, the connecting cable would not present a problem and would most likely be the preferred method.

FUTURE DIRECTIONS

Along with refinements to the algorithms listed in Section 3.3, there are three major aspects of the project yet to be undertaken. We expect that the first two can be completed, and the third can be significantly advanced, in the next year.

Identification Algorithm

The details of the identification algorithm have yet to be worked

This algorithm is the part of the method which assigns common names such as "refrigerator" or "water heater" to the observed appliances. We will initially concentrate on the two-state version of the algorithm (which only attempts to identify two-state appliances), and later see to what extent it is necessary to modify it for multi-state appliances. This task can be broken into three sub-tasks:

(A) Selecting Parameters. The most difficult aspect in the design of the identification algorithm is to select the parameters by which to classify appliances. A possible set of parameters is the following:

- Real Power
- Reactive Power
- 120 Volt, 240 Volt Balanced, or 240 Volt Unbalanced
- Time-of-Day Usage Profile
- Usage versus Temperature Relationship
- On-time and Off-time distributions

Determining the exact set of features to use will require considerable thought into the discriminatory information available in each feature and the discrimination required for the set of targeted appliances. It is desirable that the features be capable of resolving the appliance classes as finely as possible without redundancy. All software should be designed so that the set of parameters could be changed at a later point, if necessary, without losing the use of the data

collected with the earlier parameters. (The relationship between power and applied voltage of each appliance may be useful as an identification parameter as discussed at the end of Appendix B.)

(B) Methodology. A classification and lookup methodology, along with associated algorithms, must be designed. The selected features will be used to define a multi-dimensional feature space. Different regions of the space will be assumed to correspond with different classes of appliances, in either a binary or probabilistic manner. Thus an algorithm can be confronted with the observation that there is some 4000 Watt, 0 VAR (approximately) balanced 240 Volt appliance with given temporal characteristics, and would be expected to conclude something like "There is a 92% probability that it is water heater, 3% probability that it is a space heater, and 5% probability that it is unknown to the data base." The method will presumably accept high probabilities (above some specified threshold) as realities, and declare that the unknown appliance is a water heater, or else request the intervention of a human expert to complete the decision making from that point. We expect that the algorithms to perform this lookup and conclusion task will be relatively straightforward to design once the exact methodology has been decided upon.

(C) Data Base Management System (DBMS). This will be designed to allow load researchers to examine and modify the appliance class data base used by the lookup software. The DBMS will allow new appliance classes to be added as they are encountered, and will allow modification of the parameters of existing classes as required. For example, it might turn out that the refrigerator of some residence is not recognized by the lookup algorithm because it consumes less reactive

ver than was initially expected of any refrigerator. A load researcher confronted with this unrecognized appliance might conclude that it is a refrigerator from properties such as its duty-cycle, and would then modify the data-base so all further refrigerators of this manufacture would be automatically recognized. Through this sort of detective work, which in extremely difficult cases might involve contacting the residents, we expect that the data base will converge to a state which performs virtually automatically.

To facilitate this process, the DBMS will provide graphic capabilities to plot out the regions of the space assigned to given appliance classes, and scatter plots of the location of individual appliance ellipses in the space. It should also provide a means of identifying if there are any large areas of the feature space assigned to an appliance class but which have not been seen to contain any samples. With this information, initially over-large assumptions of the range of an appliance class can be reduced along the required dimensions when necessary.

To the maximum extent possible, the DBMS should be compatible with some existing commercially supported DBMS system(s). This will facilitate user support, provide software maintenance, and greatly reduce the design costs.

MIT will provide an initial data base of appliance classes based on a survey of published information and limited field testing. Only extensive field experience can provide the information necessary to refine the data base.

Multi-house Test

The second major aspect of the development process which we would like to see performed in the next year is a multi-house test. At this

point in our research, we feel the project would benefit greatly by testing the algorithms in approximately ten houses. We propose that MIT shall prepare a prototype appliance load monitor to be installed for three to four months in a set of houses selected by utility load researchers. Instrumentation for collecting concurrent appliance load data should already be in place at the selected houses.

The prototype load monitor will take the form of a portable IBM-compatible computer with a Digital AC Monitor as the sensor device and floppy disks as the output medium. Our experience to date indicates that such a general-purpose computer has sufficient computational power and memory to easily serve the purpose. This hardware is not weather resistant and consequently must be placed internal to the homes, but will only measure the aggregate load which would be available to a nonintrusive load monitor external to the home. Installation should take less than an hour.

Software will be written to implement the two-state algorithms discussed above in Sections 2.1 through 2.7. The software will be written in a high-level compiled language to facilitate its transfer to other hardware. We expect that a considerable portion of the software written for this task may later be directly usable in a special-purpose microprocessor-based commercial product.

The benefits of the multi-house test are as follows:

- (1) It will prove that the method is capable of operating with a variety of residential appliance mixes. The sites can be selected so that a variety of models of each of the targeted appliance classes are observed. Different climates can also be sought out. This should eliminate any concern that the success to date is the result of testing in fortuitous environments.
- (2) It will allow the performance of the algorithm to be quantified precisely. By choosing residences in which (intrusive) appliance load monitoring equipment is already installed and operating, we

will be able to make an accurate comparison of the nonintrusive method's results with more solid figures than we have in our previous field tests.

- 3) It will provide the opportunity to refine the algorithms based on a wider field experience. A commercial version of the Load Monitor designed after the multi-house test is certain to perform more accurately than the best that could be designed without such a test.
- 4) It will provide information necessary for developing and testing the identification algorithm, as discussed in Section 6.1. This effort will be carried out simultaneously with the multi-house test. Ten houses of appliances should be sufficient for proposing an initial appliance-class data base for use by load researchers.
- 5) It will allow us to assess the need for the multi-state appliance algorithms. If the success rate of the device is low in some of the test locations, and the poor performance is attributable to the presence of multi-state appliances, then it will convince us that the multi-state algorithms of Section 4 must be developed further and incorporated in a commercial version of the prototype. Conversely, if the success rate is high in spite of multi-state appliances, it might suggest that a two-state algorithm is sufficient for the purpose.
- 6) It will provide a data-base of transitions for use in developing the multi-state algorithms. By storing the signature transitions observed in the multi-house tests, a broad data-base would be available for "replaying" when testing the multi-state algorithms. The algorithms could be effectively tested instantaneously on several months of data from the ten houses at no additional data collection cost.

Thus we see a definite need for the multi-house test. We at MIT are not comfortable with the administrative problems associated with such a project however. We would prefer to concentrate our resources on the development of algorithms and analysis of test results rather than the travel, installation difficulties and interface with home owners that the tests will necessitate. We therefore propose that the responsibility for the multi-house test be shared with one or more interested utilities. MIT will provide the hardware and software along with instructions for installation and operation. The utility participants would select the sites and take care of the installation

and data-collection tasks. The utilities would compare the results of the test with appliance load-data collected by other means, while MIT would analyze the operation of the algorithms to see how they could be improved. MIT would also use the results in specifying the appliance-class data base, and in evaluating the multi-state algorithms. The test results would, of course, also be available for any other purposes that the participating utilities wished.

We propose the following approximate time table for the experiment:

9/85-3/86	MIT develops software and hardware for installation. EPRI locates utility participants. Utility participants select sites to monitor.
3/86-7/86	Utility installs and operates load monitors. MIT consults while concentrating on multi-state and identification algorithms.
5/86-8/86	MIT analyzes results and incorporates improvements into algorithms. Utility provides comparison statistics.
8/86-9/86	MIT prepares project report.

6.3 Multi-state Appliance Algorithms

The third major task remaining is to develop algorithms for the analysis of multi-state appliances. The degree of analysis which must be performed by these algorithms is as yet undetermined, pending the results of the multi-house test. It may be necessary to put a great deal of effort into algorithms which learn the detailed structure of multi-state appliances by techniques such as those outlined in Section 4, or it may be sufficient that the algorithm simply ignore multi-state appliances, as long as it is not confused by them.

We plan to continue developing the algorithms under the expectation that they will be needed. Even though the two-state

gorithms are sufficient for certain residential targets, the effort will not be wasted because the algorithms will likely be required for other targets. Future commercial and industrial applications of the nonintrusive method will also require the multi-state capability. In addition, a focus on the multi-state case will certainly provide insights useful to ensure that the two-state algorithm, where it is sufficient, will be only minimally confused by the presence of multi-state appliances.

Appendix A. On Using Estimates of Variance to Truncate Distributions

The fact that the signature space contains a number of distributions which each need to be estimated leads to a problem when estimating the individual distributions. The estimates of the individual distributions are necessarily based upon finite regions of the signature space which do not overlap the distributions of appliances which are nearby in the signature space. If the distributions are modeled as normal, or as members of any wide class of distributions which have non-zero tails which extend to infinity, then the fact that the tails are "truncated" and do not enter into the estimate will have an effect on the estimate. It is possible for this to seriously effect the estimate of the means and the covariance matrices. The magnitude of the effect depends upon the nature of the distribution, the extent of the tails which are excluded, and the number of dimensions in the signature space.

A concrete example using a normal distribution in one dimension should clarify the potential enormity of this problem. Suppose measurements of a signature component (e.g. start-up power) of an appliance are in fact normally distributed with a mean μ and a standard deviation σ as in Figure A-1. The learning and recognition algorithms must work together to arrive at an estimate, m , of the mean, and, s , of σ for this distribution. These estimates are used to evaluate new signature measurements in the process of determining if they are instances of this same appliance or a different appliance. If the same, the new transitions are used in updating the estimates m and s . The most obvious way of determining if a new measurement should be associated with this appliance is to pick a constant, k , and see if the new sample falls within k standard deviations of the mean. If k was 3 for example, we would expect 99.7% of the measurements actually associated with a given appliance to be recognized as such. Points beyond this distance in the tail of the distribution (the shaded portion of Figure A-1) would not be associated with the given appliance.

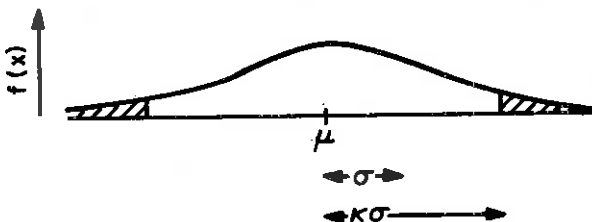


Fig. A-1. Distribution with Infinite Tails

A problem comes about because we do not know the actual standard deviation, σ ; we have only its estimate, s . The usual way of estimating s is to use equation A.1 which requires the estimate of the mean given in equation A.2. Here, N is the number of samples and the x_i are the measured values. Given a large number of samples from the normal distribution, these estimates will converge upon the actual parameters, σ and μ . Note that these equations can be recast in a recursive formulation in which the estimates are continuously updated as new x values are measured. The choice of a static or recursive form does not affect any of the following analysis.

$$s = \sqrt{\frac{\sum (x_i - m)^2}{N - 1}} \quad (\text{A.1})$$

$$m = \frac{\sum x_i}{N} \quad (\text{A.2})$$

To see the effect of truncation on these estimates, suppose that at some point in time, s and m happened to be estimated correctly as σ and μ . As new measurements arrive, those that lie within ks of m will be used to update s and m . There is no systematic effect on m because the discarded tails lie symmetrically about the mean m . There will be a systematic underestimation of s however because the truncation of the tails eliminates those points which have the largest value of $(x-m)^2$ in equation A.1. When s is reduced, the cut-off distance, ks , is proportionally reduced and the portion of measurements in the tails of the distribution increases accordingly. This results in a further underestimation of s and a continuous increase in the portion of the distribution which is ignored as beyond $m \pm ks$.

The key question is whether this underestimation process continues until s is zero and the entire distribution is ignored, or whether s continuously shrinks but approaches a non-zero limit. If a limit is approached, the important question is how close it is to σ and what effect this will have on the overall algorithm. It turns out that the answer to these questions depends on the nature of the actual distribution, the number of dimensions in the signature space and on the value selected for k . We consider the estimation of the mean in section A.1, and then the estimation of the standard deviation in section A.2.

A.1 Estimation of the Mean of a Truncated Distribution

Although the mean and the standard deviation are estimated together, we can separate out the effects of truncation on the two estimates in most cases if we separately consider cases in which ks is very large from cases in which ks becomes very small. Generally speaking, truncation has little effect on the estimate of the mean if ks is large. The tails are then small and are not weighted heavily in the estimation of m . In the case of a symmetric distribution function they balance and have no effect on m . Even if the distribution is not

symmetric, if the tails are both non-zero they will partially balance, reducing their effect on m .

If the estimation of σ is such that ks becomes relatively small however, then m can be systematically effected. In this case, m will move away from μ towards a local maximum. In the case of a normal distribution or any other unimodal distribution in which the mean occurs at the maximum, this does not create an error, because the mean and the maximum coincide. In the general case however this leads to a biased estimate of m .

To see how m is misestimated, consider the skewed one-dimensional distribution of Figure A-2 in which the mean and the maximum do not coincide. If ks was relatively small, as shown, and m happened to be set to its correct value, μ , subsequent estimates of m would move to the right, closer to the value x_{max} at which the distribution function f reaches its maximum. This is because the center of gravity of the shaded trapezoid, which covers the x values used in estimating m , is to the right of μ . The process stabilizes when the estimated m is the center of the region of the distribution under consideration. Equation A.3 must therefore be satisfied by the limiting value of m .

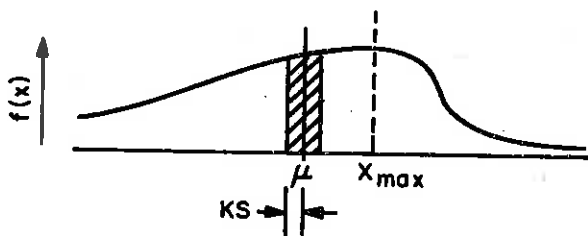


Fig. A-2. Distribution Where Mean and Maximum Do Not Coincide.

$$m = \frac{\int_{m-ks}^{m+ks} x f(x) dx}{\int_{m-ks}^{m+ks} f(x) dx} \quad (A.3)$$

If ks is small enough that $f(x)$ can be approximated by a linear function $ax+b$ within the interval bounded by $\pm ks$, then it is easy to show by direct substitution in equation A.3 and evaluation of the integrals that $a=0$ when the mean stabilizes. Thus the limiting value of m for small values of ks is a point of zero derivative, and not necessarily the mean. In practice this will be a local maximum. In the case of a multimodal distribution, local minima will also satisfy A.3, but stability considerations show that m will eventually move to a maximum. In the case where ks is too large for the linear approximation to hold within the bounds $\pm ks$, yet not so large that the tails can be ignored, m can be expected to reach a limiting value somewhere between the mean and a nearby local maximum.

In the case of a multidimensional estimation problem, the estimate of the mean can be expected to behave analogously. If a large hypervolume is averaged, the true mean will be approximated. If a small enough volume is considered, the estimate of the mean will climb up the gradient of the probability density function and stabilize around a local maximum. An intermediate sized volume should result in an intermediate effect.

A.2 Estimation of σ in one Dimension

The argument of section A.0 shows that s is always an under-estimation of σ if the tails of the distribution are non-zero. In this section we show how the parameter k can be chosen so that the estimate of s does not collapse to zero. We examine first the case of the one-dimensional normal distribution and then generalize to other distributions. Consider the zero-mean unit-variance normal distribution $N(0,1)$, cut off at values $\pm c$ as shown in Figure A-3. Define the standard deviation of the cut-off distribution to be $S(c)$. This function is given by equation A.4 in which the "downstairs" integral normalizes the distribution to unit area and the "upstairs" integral determines the second moment.

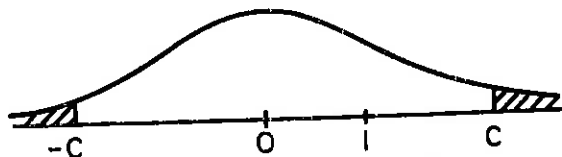


Fig. A-3. Truncated Normal Distribution.

$$S(c) = \sqrt{\frac{\int_{-c}^c x^2 e^{-x^2} dx}{\int_{-c}^c e^{-x^2} dx}} \quad (\text{A.4})$$

Although the function S can not be expressed in closed form, it can be evaluated numerically, and is plotted in Figure A-4. Certain limiting properties of S are readily determined. We know $S(c)$ approaches zero as c approaches zero because the width of the distribution approaches zero. As c increases, $S(c)$ approaches 1 because the distribution approaches the non-truncated $N(0,1)$. Most importantly for what follows, we can also evaluate the limit of the derivative of $S(c)$ as c approaches zero. In this limit, the truncated normal distribution approaches a uniform distribution of 1 within the bounds $\pm c$. The standard deviation of a uniform distribution is easily evaluated (after normalization to unit area) as:

$$\lim_{c \rightarrow 0} S(c) = \sqrt{\frac{\int_{-c}^c x^2 dx}{\int_{-c}^c 1 dx}} = \frac{c}{\sqrt{3}} \quad (\text{A.5})$$

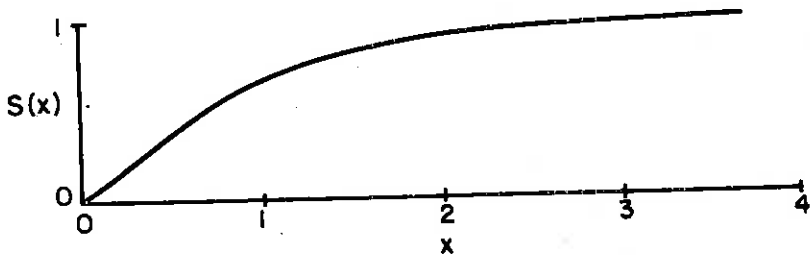


Fig. A-4. Standard Deviation of Truncated Normal Distribution.

By differentiating equation A.5 with respect to c , we see that the slope of Figure A-4 at zero is $1/\sqrt{3}$.

The key equation governing the estimate, s , is A.6. This states that the shrinkage of s , and associated growth of the tails, continues until the standard deviation of the remaining central region agrees with the cut-off point, ks . The estimation process therefore converges to a cut-off point which satisfies equation A.6.

$$c = k S(c) \quad (\text{A.6})$$

Equation A.6 can be restated as A.7 which allows it to be solved graphically (using Figure A-4), to determine the stable value of c and $S(c)$ associated with any k . The graphical solution provides considerable insight into the nature of the problem.

$$S(x) = \frac{x}{k} \quad (\text{A.7})$$

The right-hand side of A.7 is a line with slope $1/k$ which can be superposed on Figure A-4 as in Figure A-5. The point of intersection of the two curves gives the value of c and $S(c)$ which will ultimately be reached by the estimation procedure for any value of the parameter k . As the figure indicates, for relatively large values of k , (i.e. $k > 3$) c is approximately k and $S(c)$ is only slightly less than 1. In other words, the tails start approximately k standard deviations away from the mean as desired, and the standard deviation is estimated as being close to its correct value, 1. For values of k less than $\sqrt{3}$ however, the intersection is at the origin. In this case, the estimate s continues to shrink until it reaches a value of zero and the entire distribution is thrown away in the tails. Intermediate values of k , such as $k=2$, will lead to uncertain results. Although there is a well defined point where the curves intersect, the fact that the curves are almost parallel suggests that there will be a stability problem in their simultaneous solution. In practice s is given by equation A.1, not A.4, and can be represented not by a smooth curve as in Figure A-5, but as a band which takes into account likely variations from the expected value. The convergence criterion will be very sensitive to these variations if k takes a value just above the limiting value of $k = \sqrt{3}$.

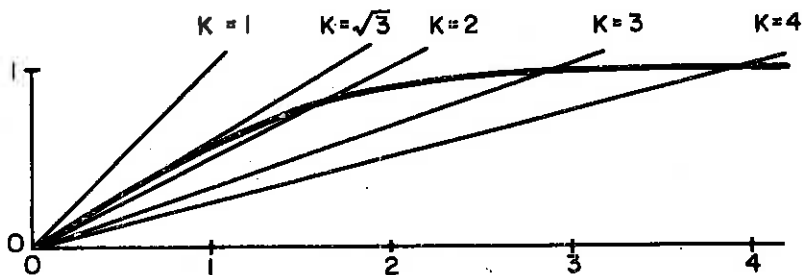


Fig. A-5. Solving for Critical Cutoff Parameter.

Most of this analysis can be shown to apply to non-normal distributions as well. Given any distribution which has a maximum at its mean, the limiting process described above is valid. As k s approaches zero, the portion of the distribution used in estimating s can be approximated as a uniform distribution and the slope of $S(c)$ at zero is $1/\sqrt{3}$. This results again in $\sqrt{3}$ being the critical point in selecting a value of k . The sensitivity of the procedure for values slightly larger than this critical value will vary depending upon the shape of the distribution.

In view of the remarks in section A.1, $\sqrt{3}$ appears to be a critical value of k for any one-dimensional distribution. If k is small enough that s is significantly underestimated, the estimate of the mean will move towards a local maximum. The portion of the distribution surrounding this value can then be approximated as a uniform or trapezoidal distribution and the above analysis again holds. We conclude therefore that in the one-dimensional case, given any possible distribution of signature transitions to estimate, k must be set significantly greater than $\sqrt{3}$.

A.3 Estimation of σ in n Dimensions

In this section we show that as the number of dimensions increases so does the minimal value of k which must be used in order to prevent collapse of the estimates. The n -dimensional case can be approached analogously to the one-dimensional analysis above. Given an n -dimensional normal distribution, a function $S_n(r)$ can be specified which gives the standard deviation along one dimension of the portion of the distribution which lies within a radius r of the mean. This is used because the portion of the distribution beyond some value of r will be discarded as too far from the mean and not be used in updating the estimates. The reciprocal of derivative of this function at zero again gives the critical value of k for the same reasons that it did in the one-dimensional case.

To estimate the derivative of $S_n(r)$ at zero we use the fact that in the limit as r approaches zero, the n -dimensional normal distribution within the n -dimensional hypersphere of radius r approaches a uniform distribution. The function $S_n(r)$, for small values of r , can then be determined from the second moment of the uniform hypersphere. To evaluate this, we first define a series of functions, V_n , which give the hyper-volume of a n -dimensional hypersphere of radius r .

$$V_1(r) = 2r$$

$$V_2(r) = \pi r^2$$

$$V_3(r) = 4/3 \pi r^3$$

These functions can be evaluated recursively using equation A.8.

$$V_n(r) = \int_{-r}^r V_{n-1}(\sqrt{r^2 - x^2}) dx \quad (A.8)$$

The second moment of the hypersphere can be evaluated using equation A.9.

$$T_n(r) = \int_{-r}^r x^2 V_{n-1}(\sqrt{r^2 - x^2}) dx \quad (A.9)$$

The standard deviation of the distribution (for small values of r) is then given by the square root of the ratio $T_n(r)/V_n(r)$.

$$S_n(r) = \sqrt{\frac{T_n(r)}{V_n(r)}}$$

Dimensional analysis shows that this will be directly proportional to r (as was the one-dimensional case in equation A.5) and its derivative is therefore given by equation A.10. We use k_n to denote these critical values of the parameter k for n -dimensional estimation.

$$k_n = \lim_{r \rightarrow 0} \frac{d}{dr} s_n(r) = \frac{1}{r} \sqrt{\frac{T_n(r)}{V_n(r)}} \quad (A.10)$$

Evaluation of equation A.10 for small values of n leads to the following table of values. The integrals A.8 and A.9 quickly become tedious as n increases.

Table A-1. Critical values of k in n dimensions

n	k_n
1	$\sqrt{3}$
2	$\frac{2}{\sqrt{3}}$
3	$\sqrt{\frac{5}{3}}$
4	$\sqrt{\frac{6}{3}}$

Inspection of Table A-1 suggests that general term is given by A.11. This is in fact the case, as is shown in the following section. It should be remembered when selecting a value of k that these are critical values, at or below which the estimation is bound to collapse. In practice we expect that k should be significantly higher than k_n if the estimation procedure is to be robustly stable.

$$k_n = \sqrt{n + 2} \quad (A.11)$$

The comments at the end of section A.2 concerning nonnormal distributions also apply here. These values of k_n , although derived from a normal distribution will apply to a wide variety of distributions because as kr shrinks, the estimate of the mean will simultaneously gravitate to a local maximum, and the distribution will, in the limit, approach uniformity.

When determining distribution inclusion via $(x-m)^t \sum^{-1}(x-m) < k^2$, this requires that $k^2 > n+2$. The consequence of this is that there is a cost to increasing the number of dimensions used in a feature space. Increasing the number of dimensions does not necessarily increase the resolving power of the space.

A.4 Proof that $k_n = \sqrt{n+2}$

This section consists of a proof of equation A.11 which can be skipped by the casual reader. The technique of section A.3 does not provide a result for the general case, only cases for particular dimensions. To derive the general result we proceed indirectly.

In view of the discussion above we need to show that

$$S_n(r) = \frac{r}{\sqrt{n+2}} \quad (\text{A.12})$$

From this it follows that k_n , the reciprocal of the derivative of $s(r)$ at $r=0$, is $\sqrt{n+2}$. Instead of evaluating V_n and T_n in cartesian coordinates as in section A.3, we proceed in spherical coordinates. The "surface area" of a n -dimensional hypersphere of radius r will vary as the n -minus-first power of r and can be given by

$$A_n(r) = a_n r^{n-1}$$

where the a_n are constants yet to be determined. For example $a_2=2\pi$ and $a_3=4\pi$. The volume of the hypersphere can be determined by integrating as follows:

$$V_n(R) = \int_0^R a_n r^{n-1} dr = \frac{a_n}{n} R^n$$

Analogously, the second moment, U_n , of the $n-1$ -dimensional surface of a n -dimensional hypersphere of radius r will vary with r to the n -plus-first power and can be integrated to determine the second-moment function, T_n .

$$U_n(R) = b_n R^{n+1}$$

$$T_n(R) = \int_0^R b_n r^{n+1} dr = \frac{b_n}{n+2} R^{n+2}$$

As in section A.3, the standard deviation can be expressed as the square root of the ratio $T_n(r)/V_n(r)$. We do this with the above spherical formulations of V_n and T_n rather than the Cartesian formulations of section A.3. The constants a_n and b_n remain unevaluated.

$$s_n(r) = \sqrt{\frac{b_n}{a_n} \frac{n}{n+2}} R \quad (\text{A.13})$$

In order to evaluate these constants, consider the n -dimensional normal function e^{-x} , and integrate it across all n -space to determine its "mass" and second moment. We do this first for its "mass" in equation A.14 and then for the second moment in A.15. The left-hand integrals are performed in spherical coordinates using spherical shells as volume units and the right-hand integrals are equivalent but in Cartesian coordinates.

$$\int_0^{\infty} a_n r^{n-1} e^{-r^2} dr = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} e^{-x_1^2 - x_2^2 - \dots - x_n^2} dx_1 dx_2 \dots dx_n \quad (\text{A.14})$$

$$\int_0^{\infty} b_n r^{n-1} e^{-r^2} dr = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} x_1^2 e^{-x_1^2 - x_2^2 - \dots - x_n^2} dx_1 dx_2 \dots dx_n \quad (\text{A.15})$$

The right-hand integrals can be factored into the product of n separate integrals (because the terms of the integrands are non-negative). The left-hand integrals can be seen to be gamma functions with a change of variable $t=r^2$ and $dt=2r dr$. The gamma function is defined by A.16. The substitution $z=n/2$ is used to derive A.17 from A.14, while the substitution $z=1+n/2$ is used to derive A.18 from A.15.

$$\Gamma(z) = \int_0^{\infty} e^{-t} t^{z-1} dt \quad (\text{A.16})$$

$$\frac{1}{2} a_n \Gamma\left(\frac{n}{2}\right) = \left(\int_{-\infty}^{\infty} e^{-x^2} dx \right)^n \quad (\text{A.17})$$

$$\frac{1}{2} b_n \Gamma\left(1 + \frac{n}{2}\right) = \left(\int_{-\infty}^{\infty} x^2 e^{-x^2} dx \right) \left(\int_{-\infty}^{\infty} e^{-x^2} dx \right)^{n-1} \quad (\text{A.18})$$

The definite integral on the right-hand side of equation A.17 is well known to evaluate to $\sqrt{\pi}$. The remaining definite integral of A.18

evaluates to $\sqrt{\pi}/2$. (These can be evaluated if desired by substituting these values for a_2 and b_2 into the spherical formulations of V_2 and T_2 above, and equating them to the easily evaluated integrals A.8 and A.9 respectively.) For our purposes here, we do not need to evaluate a_n and b_n separately, but only the ratio b_n/a_n . The "factorial" property of the gamma function stated as A.19 allows the ratio to be simplified to equation A.20.

$$\Gamma(z+1) = z\Gamma(z) \quad (\text{A.19})$$

$$\frac{b_n}{a_n} = n \quad (\text{A.20})$$

Substitution of A.20 into A.13 gives A.12. QED

A.5 Implications for Nonintrusive Load Monitor

In the design of the Nonintrusive Appliance Load Monitor, cognizance must be taken of two results from the above analysis. The first is that the cutoff parameter, k , must be relatively large in order to ensure there will be no "collapse" of the estimates. One can not attempt to estimate the properties of a truncated cluster of transitions unless all the transitions within several standard deviations of the mean are considered. We have chosen k to be 4 in the prototype load monitor, and this value appears to work satisfactorily. (The most significant problems discussed in Section 3.2 appear to be in the time-domain aspects of the monitor, matching ON/OFF pairs, not in the signature-space domain.) The drawback to such a large k is that it requires a relatively large area about each mean value in the signature space to be devoted to each appliance. Accordingly, fewer appliances can be discriminated in the signature space.

The second consequence of the above analysis concerns the number of dimensions of the signature space. The minimum usable value of k was seen to vary with the dimensionality of the space. As the number of dimensions is increased, k must also be increased. This was a factor in selecting between the four-dimensional and two-and-one-half dimensional appliance representations discussed in Section 3.1.1. Counter-intuitive as it may appear, it is possible that increasing the number of signature components can, in certain circumstances, reduce the discriminating power of the signature space. This happens when pairs of clusters are similar in terms of the existing dimensions and identical in terms of the added dimensions. This is the most common occurrence when we consider the choice between the two formats for the load monitor. If two 120 V appliances on the same leg are similar in terms of their real and reactive power consumption, measurements of the of other leg add no discriminating information, because the power consumption there should be unaffected by a 120 V appliance. But the very fact that four dimensions instead of two are examined requires a larger value of k to be used, and hence a loss of resolution between the two appliances if they should happen to be right at the border of discriminability. This was a factor in the selection of the two-and-one-half dimensional format for the prototype. However, based on the

results of Section 3.2, especially the common occurrences of unbalanced 240 V appliances which can not be represented in the reduced space, we will be using the full four-dimensional representation in the next version of the Load Monitor.

Appendix B. Normalization with Non-Integer Exponents

The normalization process described in Section 2.2 attempts to vitiate the effect of line voltage variation by determining what the power consumption of the house would be if the utility were to provide a constant nominal voltage of 120 V. Estimates stating what would be the case in a counterfactual situation always require some assumptions concerning the structure of the subject matter. The assumption of Section 2.2 is that appliances are linear circuit elements, so that their power consumption varies as the square of the voltage. The fact is that this is not the case for most appliances. A more general model is to assume that a power law relationship of the following form holds for each appliance:

$$\text{Power} = k \text{ Voltage}^{\text{exponent}}$$

The constant k, and the exponent will vary from appliance to appliance and will take on separate values for the real and reactive power.

This was only indirectly considered in Reference [1], in the context of selecting between power, current and admittance as the most appropriate signature components. In this framework, those signature options translate into the question of selecting 0, 1, or 2 as the exponent in the normalization equation:

$$\text{Normalized Power} = \text{Measured Power} \left(\frac{120}{\text{Voltage}} \right)^{\text{exponent}}$$

This normalization equation follows directly from the above power relationship as the appropriate way to normalize any measured power to the power level which would have been measured if the voltage were at 120 V. With the exponent set to its current value of 2, the assumption is being made that the admittance of all appliances is independent of voltage. If the exponent is set to 1, the effect of the formula is the same as assuming that current is independent of voltage. If the exponent were set to zero, the voltage dependence would drop out from the above expression, the result of assuming that the power is independent of voltage and does not need normalization.

Reference [1] claims that admittance is the optimal signature, which is equivalent to claiming that 2 is the optimal exponent for normalization. However, the oversight in Reference [1] is that power, current and admittance are not the only choices to use as signatures. All Reference [1] actually shows is that 2 is the optimal value out of the three integer values considered: 0, 1, and 2. In fact, the exponent can be set to any value, which need not be an integer. When we consider all possible real values between the integers also, the issue becomes less clear cut. An additional possibility not considered in Reference [1] is that the real and reactive parts of the load can be normalized independently, with differing exponents. This adds a new dimension to the normalization question.

To address these questions we consider three arguments. The

first argument is theoretical. Briefly, it is that the linear model predicts that 2 is the optimal exponent for both the real and reactive components of the load. We believe that the linear model is a good first-order approximation to most of the power consuming circuit elements found in appliances, and therefore we conclude that the optimal exponents should not vary far from 2. This argument does not tell us much however because it is not clear how far from 2 we would expect an exponent to be for the degree of nonlinearity that might be found in typical appliances, especially motors. Therefore we must examine measurements of appliance characteristics. We have performed appliance measurements in two ways: in isolation, and as part of an entire household load. These measurements lead to similar conclusions.

First we consider appliances in isolation. We can control the ac voltage in a laboratory environment, and have measured the power consumption versus voltage characteristics of a limited number of appliances. By measuring the real and reactive power consumption at 5 and 125 V, using equipment described in Reference [1], we can determine the optimum exponent in this voltage range. The exponent which makes the power law best fit the measurements will be the optimal exponent with which to normalize power if the load consisted only of appliances of this nature. After measuring the power at 115 and 125 V, we solve for the exponent using:

$$\text{Exponent} = \frac{\log \frac{\text{Power at 125 V}}{\text{Power at 115 V}}}{\log \frac{125}{115}}$$

This relationship is derived straightforwardly by eliminating k from the power law above, after applying it at 115 and 125 V. The optimal exponents for normalizing the real and reactive power, as calculated this way are listed in Table B-1, for four appliances.

	Real	Reactive
Coffee Pot	2.0	-
Light Bulb	1.5	-
Table Fan	1.2	2.4
Refrigerator	0.7	2.9

Table B-1. Normalizing Exponents for Individual Appliances

It is interesting that only the electric coffee pot shows the theoretical value of 2. The water in the coffee pot stabilizes the temperature of the heating element, which keeps its resistance constant. It is therefore well approximated by a linear circuit element. The light bulb, in contrast, shows a distinct non-linearity.

Its power consumption increases slower than quadratically because the filament resistance increases at the higher temperatures that result from higher voltages. The motors of Table B-1 show an even greater departure from linearity. In both cases, the real power increases much slower than quadratically, close to linearly in fact, and the reactive power increases faster than quadratically.

It is not clear, based on these limited measurements, how typical this data is. If these values are representative of a wide range of appliances, then it seems that normalization could be improved, and hence clustering would be tightened, with non-integer exponents. An exponent of approximately 1.5 for the real component and 2.5 for the reactive component might improve the performance of the load monitor.

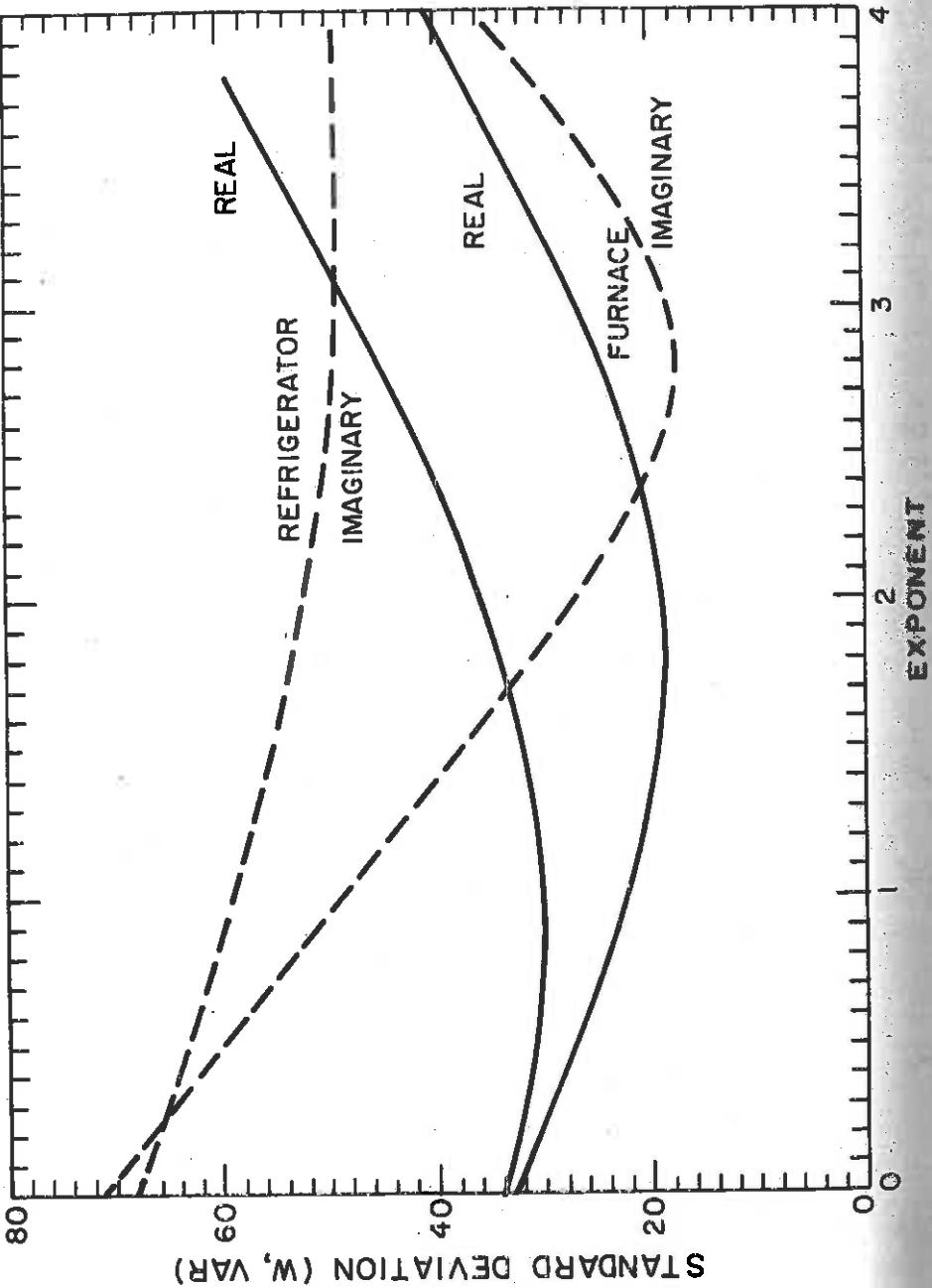
The choice of exponent could be affected somewhat by the target appliances. If water heaters are targeted, they are expected to behave identically to the coffee pot of Table B-1. Accordingly, an exponent of 2 would be used for tightest clustering. If appliances with induction motors are the main targets, and the above data turns out to be representative, then values closer to 1.5 and 2.5 may be preferable.

The third way of addressing the normalization question is to measure the size of transition clusters using different normalization exponents. Many factors enter into the size of a cluster. By isolating a fixed set of transitions for which the line voltage of each is known, we can isolate the effect of the normalization procedure. We quantify cluster size by the standard deviation of the transitions in the real and reactive directions separately. Figure B-1 shows this observed relationship for a week of transitions of the refrigerator and oil burner of the house described in Section 3.2.1. The four curves separate out the standard deviation of the real and reactive power scatter of the two clusters. Table B-2 summarizes the results by indicating the exponent at which the cluster size is minimized for the two appliances.

	Real	Reactive
Refrigerator	1.0	3.2
Oil Burner	1.7	2.7

Table B-2. Normalization Exponents to Minimize Cluster Size

This data supports the above conclusion that an exponent less than 2 for the real power, and greater than 2 for the reactive power will improve load monitor performance. It remains unclear however how far from 2 the values should be to optimize performance over the widest possible range of target appliances. When interpreting this data, it must be remembered that the particular range of voltages present during this week have a primary influence on the data. Long-term measurements, or measurements of a different week may show different minima.



A fourth approach to determining these exponents is simply to ask. It has only recently occurred to us that these exponents are probably known for many large scale electric systems. They must be used for estimating the power reduction benefit that would result from planned brownouts. Studies of tap-changing transformers would also involve them. They might also be measured directly during brownouts. We will explore these areas in the near future.

The choice of normalization method is not believed to be crucial to the load monitor in most situations. The impact of the exponents should only be seen in the size of transition-clusters. This should only have an effect on the load monitor in situations where two similar appliances need to be discriminated. If a household has two refrigerators with slightly differing electrical characteristics, for example, the choice of exponent might make the difference between being able to separate two tight nearby clusters, or finding them irresolvably fused into an ambiguous cluster. It is impossible at this time to estimate how common this type of situation will be in typical applications. The multi-house test described in Section 6.2 will attempt to settle this issue.

Examination of optimum exponents for individual appliances, such as those listed in Table B-1 above suggests a related topic. The question is whether or not the load monitor might determine the optimal real and reactive exponents for each cluster individually, and then use these as parameters for identification purposes. By storing a number of transitions along with the voltage at which each transition occurred, the power versus voltage relationship of each appliance can be readily determined. A cluster which is minimized with an exponent very close to 2 is likely to be a water heater or coffee pot. Other exponent values would indicate induction motors or other load types perhaps. Although we have not ruled this possibility out as yet, it is unclear if this parameter really adds any information to that which is available from other, independently motivated, parameters. Water heaters should be readily identified by their size and lack of reactive power, while induction motors seem to be identifiable by their power factors. It is possible however that there are some appliance targets for which identification would be improved by this parameter. If so, it will surely be included. The resolution of this matter awaits further work on identification, as discussed in Section 6.1.

Appendix C. User Control Commands

This appendix describes the user control features of the current version of the Load Monitor which were passed over at the end of section 3.1.1. The program which controls the prototype load monitor an input data from four sources, and is capable of producing a wide variety of outputs which indicate its status, actions and conclusions. A small command language is implemented which can be used to specify which of these options are desired at any given time. The program requests commands upon start-up and after each main cycle (just after the working buffer has been cleaned and before it is re-filled). Some of the commands produce immediate output regarding the state of the program at that instant. Others set internal flags which control the amount of annotation that the program provides as it makes its decisions during the process of the main loop.

If the program were collecting data from the AC Monitor while waiting for the user to input commands, it would be possible for the input buffer to overflow, so the AC Monitor is paused while the user interacts with the program. In this case it does not normally stop after cleaning the buffer to request input. Key O can be used to force the program to request commands at its next convenience.

Commands can be given as listed below. The CONT key is used to terminate a command. Upper and lower case can be mixed freely. Word order is completely free. For those commands which allow a list of clusters to be specified, the individual clusters can be identified either by number or by name. (The SET NAME command is used to name a numbered cluster.) Thus a cluster list can be something like "1 5 refrigerator 9." If no clusters are specified in such a command, all clusters are affected. In print and plot commands, the order of the entries in the cluster list is used as the order of output.

Commands

- | | |
|------------------------------|--|
| HELP | This causes a list of the legal commands to be printed on the CRT. Adding "TO PAPER" generates a hard copy. |
| AC_MONITOR MODE | This causes the program to process measurements in real-time from the AC Monitor. It also disables the usual input breaks until KEY O is activated. Note that this is the default mode. The command may be useful, however, in order to re-enter this mode after another of the following modes has been in effect. (The four modes are mutually exclusive.) |
| FILE REPLAY MODE
filename | This causes the program to input transitions from the specified stored-data file. |
| TRANSITION MODE | This allows the user to input transitions from the keyboard whenever the buffer is to be filled. |
| MEASUREMENT MODE | This allows the user to input measurements from |

the keyboard (testing the edge-detector and trans-converter).

SAVE number
TRANSITIONS
TO filename This creates and opens a file for storing transitions. It is closed after the specified number of transitions are saved. Note that the HOLD command should be used to close this file if the program is stopped before the file is filled.

PRINT STATUS Prints a brief status message which includes the active mode, the number of clusters, active file status, etc.

TO PAPER This can be added to any print or plot command (or the HELP command) causing output to the internal printer in the case of print commands or to the pen plotter in the case of plot commands. The word "TO" need not appear.

TO SCREEN This can be added to any print or plot command to cause CRT output. Note that this is the default condition and so has no effect except when used to change a previously-set hard-copy option.

PRINT VALUES Prints a list of all the parameter values which may be changed from time to time (by editing the program) to modify the program's operation.

PRINT TYPESCRIPT (Sets a flag which) causes a copy of all commands input by the user to be printed as a record.

CANCEL TYPESCRIPT Undoes the effect of the above command.

PRINT DIFFICULTIES By adding the TO PAPER or TO SCREEN options, this command determines where error messages and "trouble reports" are printed.

PRINT CLUSTERS
<Cluster List> This causes the learned parameters of all clusters in the list to be printed immediately. If the list is empty, all clusters are printed.

PRINT SUBCLUSTERS
<Cluster List> This causes the learned parameters of the specified clusters and their two sub-clusters to be printed immediately.

PRINT DECISIONS
<Cluster List> This causes the classification decision of ON/OFF cycles to be printed as they happen for all specified clusters. (This will be either that the cycle matched an existing cluster or is new.)

CANCEL DECISIONS
<Cluster List> This removes the specified clusters (or all clusters, if the list is empty) from the list of clusters about which to print decisions.

PRINT THOUGHTS This is analogous to PRINT DECISIONS, but includes printout of the factors leading up to the

decisions. CANCEL THOUGHTS is likewise analogous to CANCEL DECISIONS.

RINT BUFFER

This causes the current state of the buffer to be printed immediately. At the point of input, between cleaning and loading, this can only contain unmatched transitions.

RINT
LOADED_BUFFER

This sets a flag which causes the buffer to be printed out (from then on) just after it is filled.

ANCEL
LOADED_BUFFER

This undoes the effect of PRINT LOADED_BUFFER.

RINT or CANCEL
MARKED_BUFFER

These are analogous to the above commands, but cause the buffer to be printed out at a point just before cleaning.

RINT HYPOTHETICAL
<c11 c12> or
<c11>

This causes the hypothetical fusion of two clusters or sub-clusters to be generated and printed out, without actually changing the internal cluster table. (It generates entry zero, which is not otherwise used.) If only one cluster is given, the hypothetical fusion of its two sub-clusters is given. If no arguments are given, The split test is performed on the subclusters of each cluster, and the join test is performed between all pairs of clusters.

LOT CLUSTERS
<Cluster List>

This causes the decision-set ellipses for the specified clusters to be plotted immediately. If the S-Matrix is not sufficiently specified, the decision radius is used to draw a circle.

LOT SUBCLUSTERS
<Cluster List>

This is analogous to PLOT CLUSTERS but includes plots of the two sub-clusters of each cluster.

LOT POINTS
<Cluster List>

This causes a grid to be plotted immediately and the transitions associated with the given clusters to be plotted later as they are observed. the CANCEL POINTS command cancels this point-plotting mode.

SIMULTANEOUSLY

This word can be added to any plot command to cause the plot to be generated over the same set of axes and with the same scale as the previous plot. (The plot must be to the same medium.)

JUMP

This causes whatever plot is currently on the CRT to be "dumped" to the internal printer.

VIEW <n>

This causes the CRT to display the most recent plot generated (instead of the text output). If an argument (n) is given, the plot is shown for

that many seconds.

- SET RANGE
xmin xmax
ymin ymax
- This specifies the bounds of all subsequent plots. Setting either or both of the min-max pairs to 0/0 causes plot-by-plot auto-scaling to resume on that axis. Auto-scaling is the default method for determining plot bounds.
- SET LIMITS
xmin xmax
ymin ymax
- Sets the plot area limits, in millimeters, for all subsequent paper plots.
- SET COLOR
cluster pen
- This causes the specified pen to be used when plotting points or ellipses of the specified cluster on any future paper plots. The cluster can be specified by name or number. The pen number can range from 1 to 8. Initially, all clusters are set to color 2.
- SET CHARACTER
cluster
character
- This specifies the character to use in subsequent point-plots of the given cluster. Initially, all clusters are set to "*".
- SET NAME
cluster name
- This allows a cluster to be identified (or renamed) for future reference and output.
- SET READ_COUNT n
- This allows a specification of the maximum number of transitions to read into the buffer at one time when in file-replay mode. If unspecified, the default condition is that as much data as fits is read in.
- SET CLUSTER
Cluster Leg
Count P Q
Sigma-xx
Sigma-yy
Sigma-xy
- Allows one to modify or create a cluster entry. Add 100 to a cluster index to affect the first subcluster, and 200 for the second. Note that gaps can not be made in the cluster table; only the next highest number can be used when creating a cluster.
- TRANSITION
Leg P Q
- If in transition mode, this command inserts the given transition into the working buffer (if there is room). Time entries are automatically generated (1 hour intervals).
- MEASUREMENT
Power1 Reactive1
Power2 Reactive2
- If in measurement mode, this command simulates a measurement four-vector (allowing testing of the edge-detection and dimension reduction algorithms).
- SPLIT cluster
- Causes the specified cluster to be split into two new appliances corresponding to the sub-clusters of the given cluster.
- JOIN cluster1
cluster2
- Causes the two given clusters to be joined into a new cluster representing their fusion.

- SAVE CLUSTERS TO filename Writes a data file containing the current cluster table.
- GET CLUSTERS filename Reads in a data file containing a cluster table generated by the above command. The new table replaces whatever clusters were previously defined.
- QUIET Causes the program not to beep when difficulties occur.
- CANCEL QUIET Allows the program to beep when it prints "trouble messages".
- HOLD This causes a clean pause to occur. (The AC Monitor is given a Q-command.) The program can be continued with the CONT key. (Note that if data is being SAVED, the last buffer-full may not be written to disk. The command ASSIGN * TO 2 <EXECUTE> should be performed if the partial data file is to be kept.)
- On This allows the program to stop requesting input and proceed with its analysis of the transitions. If n is given, it will skip n-1 input requests, automatically loading and processing the buffer until the nth cycle is complete.

SYNTAX

Note that the following words may optionally appear in any command:

TO AND OF ABOUT THE

Note also that word-order is free, that upper and lower case characters are equivalent, and that command words and cluster names may be abbreviated to two or more characters. Thus the following two commands are equivalent:

Print Clusters 1 2 and 3 to Paper
PA PR CLU 1 2 3

Special Function Keys

- KEY 0 -- Causes an input break after the next buffer cleaning.
- KEY 1 -- Prints status to screen
- KEY 6 -- Copies CRT plot to printer; equivalent to DUMP command.
- KEY 7 -- Toggles between the CRT plot and the CRT printout.
- IFT KEY 0 -- Causes a clean pause; equivalent to the HOLD command.

Appendix D. Split/Join Test

As described in Section 3.1.4, the split/join test is used to determine if two given clusters are likely to belong to a single subsuming cluster. The test considers two hypotheses: that the clusters are unrelated, and that the clusters have arisen by chance from a single distribution. It then determines a relative likelihood ratio for these hypotheses. This ratio is then compared to two thresholds which determine if the clusters should be joined or split apart. As discussed in Section 3.1.4, the same test is used for splitting and joining with a deadband between the thresholds that provides a hysteresis effect.

The input to the test is a set of parameters which describe the two clusters. For each cluster it is necessary to provide:

- (a) The mean of the observations, ie. the center of the ellipse. This is actually a pair of quantities: a real and a reactive power value in the form of a vector, M . It is computed by averaging the observed transitions.
- (b) The number of observations that have been averaged together to form the cluster, N . It is computed by counting the number of transitions that fall within the cluster.
- (c) The scatter matrix which defines the shape of the ellipse, S . This is a symmetric two by two matrix, so it actually contains only three degrees of freedom. It is computed by taking the average value of $(t-M)$ times its transpose, where t takes on the value of each observed transition in turn.

The averaging required to determine the mean, M , and the scatter matrix, S , is performed in a recursive manner which does not require the storage of all previous transitions. There is also a finite memory filter in both averaging processes which creates a moving average effect. This should allow the load monitor to follow gradual changes in an appliance's electrical characteristics. The time constant of this filter is set at 25 transitions.

The arguments M , N , and S are subscripted 1 and 2 for the two clusters. We first compute the corresponding properties of the distribution which would most likely generate the pair of clusters. This combined distribution is given the subscript 3. Its parameters are computed as follows:

$$N_3 = N_1 + N_2$$

$$M_3 = \frac{N_1 M_1 + N_2 M_2}{N_3}$$

$$S_3 = \frac{N_1 S_1 + N_2 S_2}{N_3} + \frac{N_1 N_2}{N_3^2} (M_1 - M_2)(M_1 - M_2)^t$$

The likelihood ratio, L , indicating the relative likelihood of the one cluster hypothesis over the two cluster hypothesis is then computed as:

$$L = \frac{N_1}{N_3} \ln (\det S_1) + \frac{N_2}{N_3} \ln (\det S_2) + \ln (\det S_3)$$

The thresholds for action are -2.5 and -3.5 . The two clusters are joined if L is greater than -2.5 . A single cluster splits if L for its two subclusters is less than -3.5 . These two values were determined empirically, by observing the results of applying the test to clusters which were known to belong together or be separate.

Appendix E. Load Research Results: Appliance Cycles, Energy Consumption, and Timing

This appendix contains a series of figures which indicate the energy and timing characteristics that the Prototype Nonintrusive Load Monitor has learned for the major appliances of the three test houses described in Section 3.2. As such, it shows the type of results that the load monitor is capable of. Comments on many of the appliances are given in the last paragraphs of Sections 3.2.1, 3.2.2, and 3.2.3. For each appliance there are four separate plots:

- (1) ON/OFF Cycles. The first plot indicates the exact time of each ON and OFF occurrence which the load monitor has understood to be caused by the appliance. The horizontal lines indicate the intervals in which the monitor determined the appliance to be operating. An initial tick above a horizontal line indicates the time the cycle began. The tick below the line at the end of each cycle indicates the time when the appliance turned off. When an isolated tick is shown, it means the monitor detected that the appliance turned on or off, but did not find the matching transition for the cycle. The missing transitions generally could not be identified by the current version of the load monitor because they occurred simultaneously with some other appliance transition.

Note that the detailed ON/OFF information of this first figure will not be saved or output by the commercial load monitor for reasons discussed in Section 1. It is examined here for evaluation purposes. The averaged information which is presented in the remaining three plots will be output at monthly intervals.

- (2) On Time Versus Time-Of-Day. The second plot for each appliance indicates what fraction of the time it is operating during each clock hour. For example, a value of 25% between 1:00 and 2:00 means that there is a 25% chance the appliance will be on if one looks at it at some random instant in this hour. By multiplying these time fractions by the operating power level of the appliance, the energy versus time-of-day characteristics can be calculated. We prefer to show time here because it is directly comparable from appliance to appliance. (For multi-state appliances, energy would have to be output directly, because there is no unique operating power level.) The final version of the load monitor will generate two separate profiles for each appliance: one for weekdays and one for weekends. Note that due to various weaknesses of the algorithms that are pointed out in Section 3, the usage profile figures are too low. The current version of the load monitor is only reporting on 75 to 90% of the actual energy usage.
- (3) On Time Distribution. The third plot for each appliance indicates how long the appliance is left on, when it is

used. This is shown in the form of a distribution. For example, a value of 25% between 1 and 2 hours means that there is a 25% chance that the appliance will stay on for at least an hour but turn off in less than 2 hours after it is turned on. The bars shown always add to 100%. This data is generated by use of twelve "bins" which are used to count how many cycles are observed with durations that fall into each of twelve size ranges. The size ranges were selected somewhat arbitrarily. They are organized quasi-logarithmically but with round numbers: 0-10 seconds, 10-30 seconds, ... , 2-3 hours, 3 hours or more.

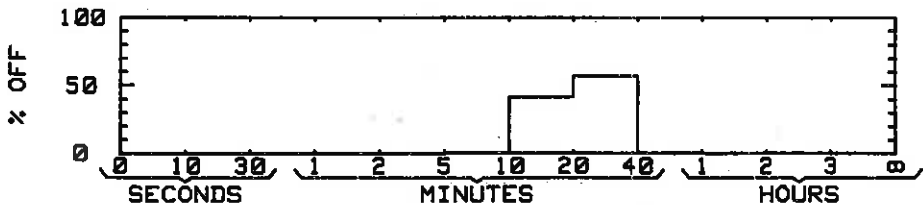
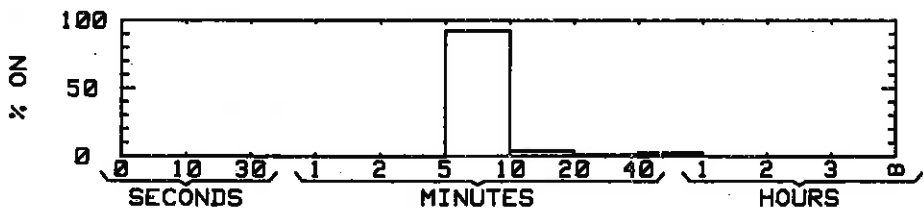
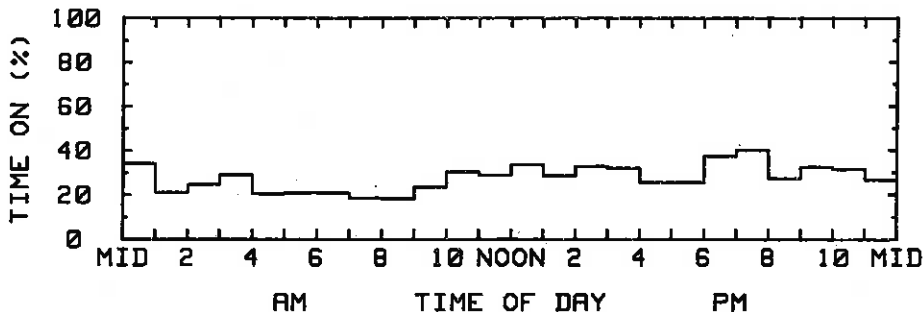
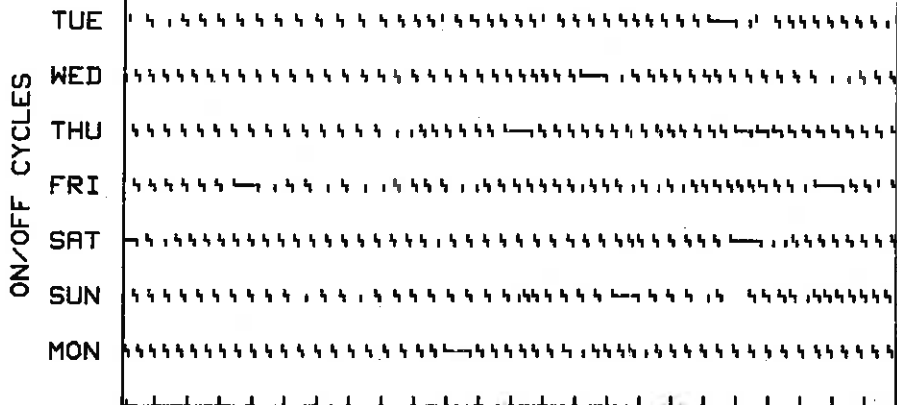
- 4) Off Time Distribution. The fourth plot is analogous to the above but indicates the distribution of the length of time for which the appliance is left off before being turned on again.

We expect that the information contained in the last two figures will be very valuable when identifying appliances, as different appliance classes often display different ranges of characteristic duty cycles, even if their power consumption is similar.

A list of figures is included with the table of contents.

REFRIGERATOR AT NATICK HOUSE

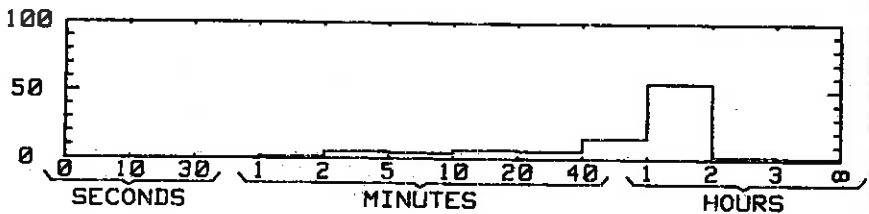
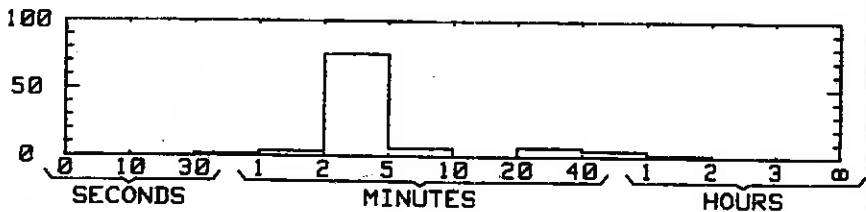
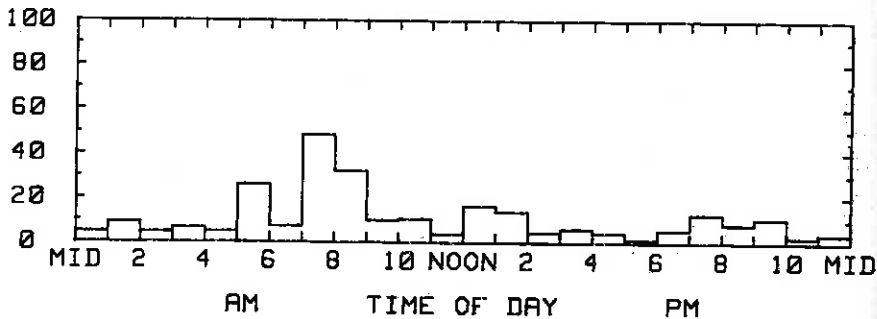
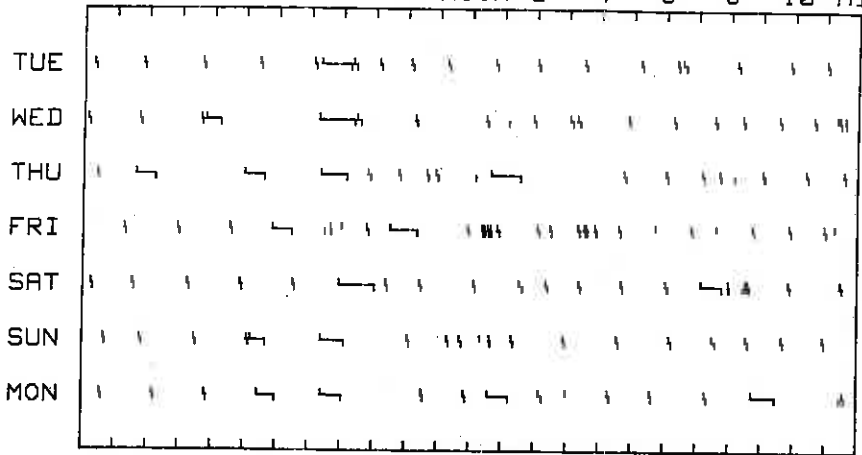
MID 2 4 6 8 10 NOON 2 4 6 8 10 MID



DURATION DISTRIBUTIONS

OIL BURNER AT NATICK HOUSE

MID 2 4 6 8 10 NOON 2 4 6 8 10 MID



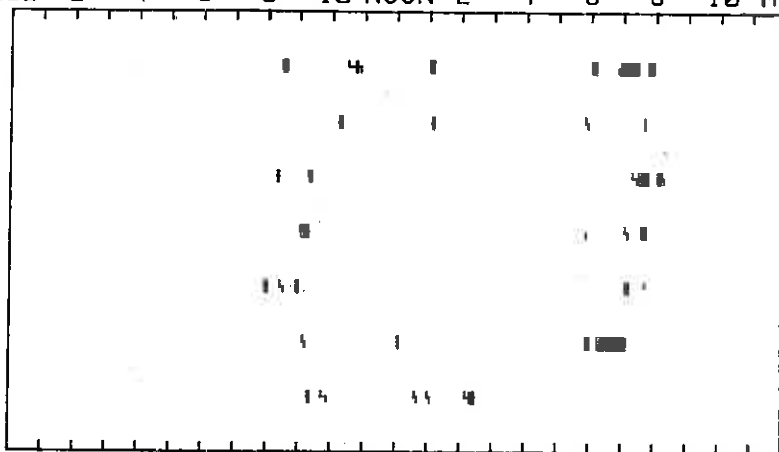
DURATION DISTRIBUTIONS

SMALL BURNER AT NATICK HOUSE

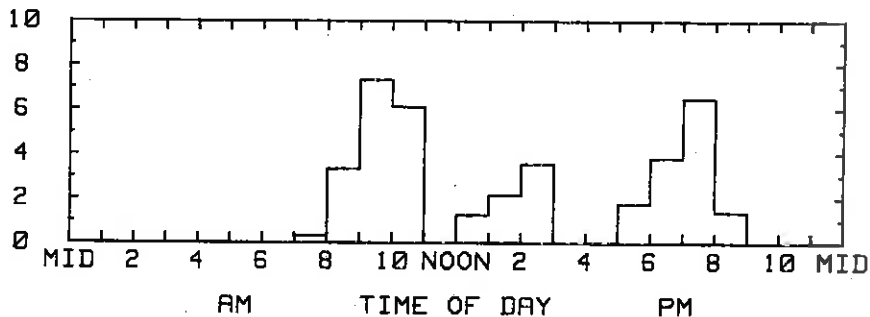
MID 2 4 6 8 10 NOON 2 4 6 8 10 MID

ON/OFF CYCLES

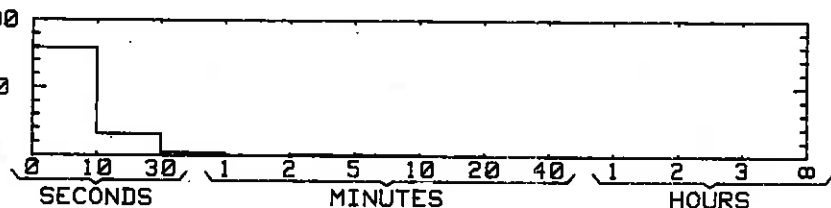
TUE
WED
THU
FRI
SAT
SUN
MON



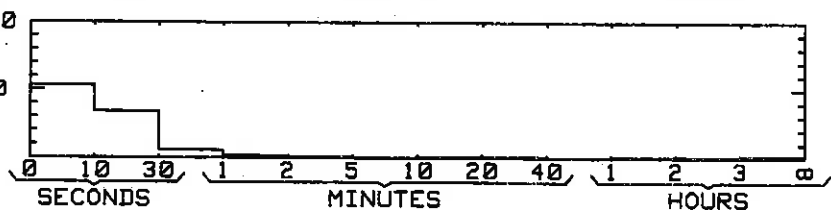
TIME ON (%)



% ON



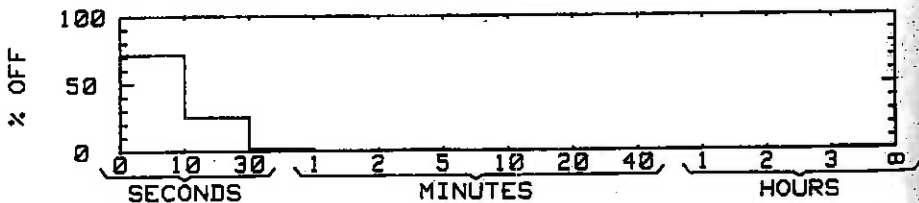
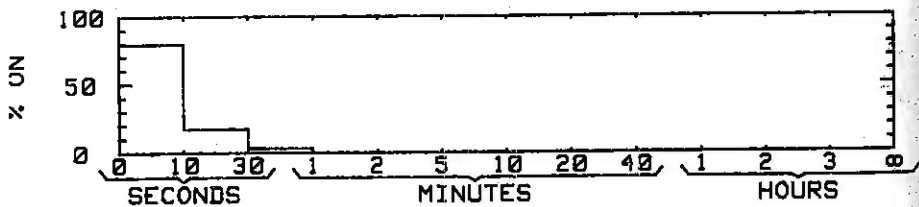
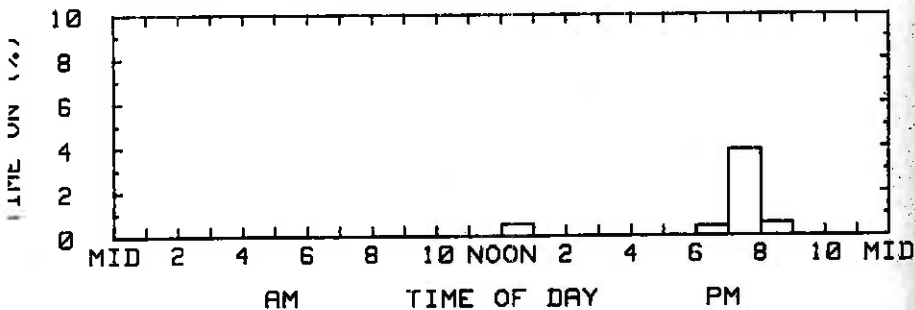
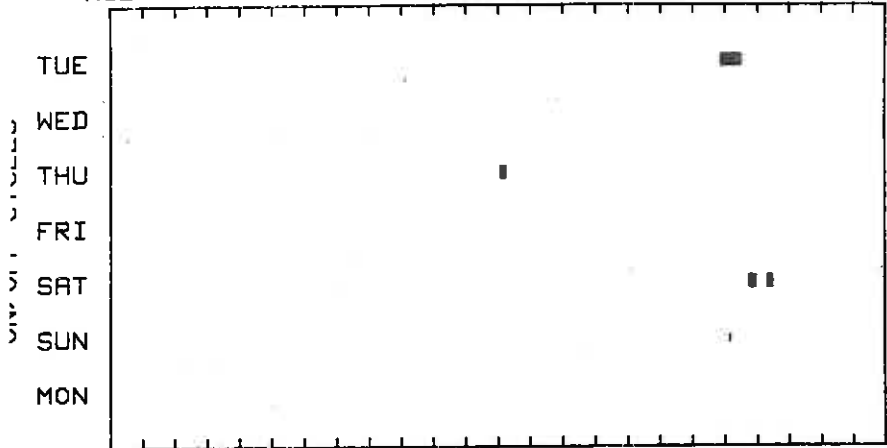
% OFF



DURATION DISTRIBUTIONS

LARGE BURNER AT NATICK HOUSE

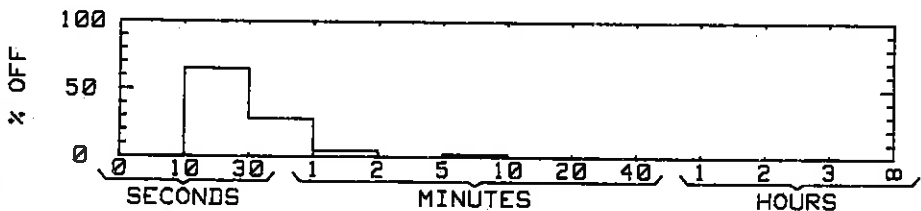
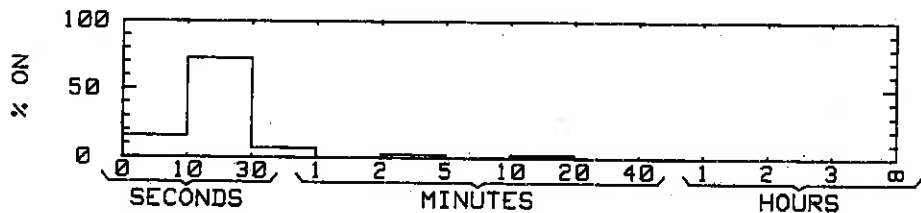
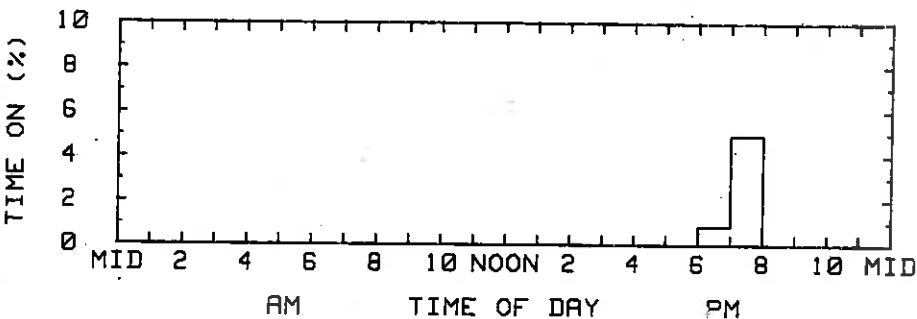
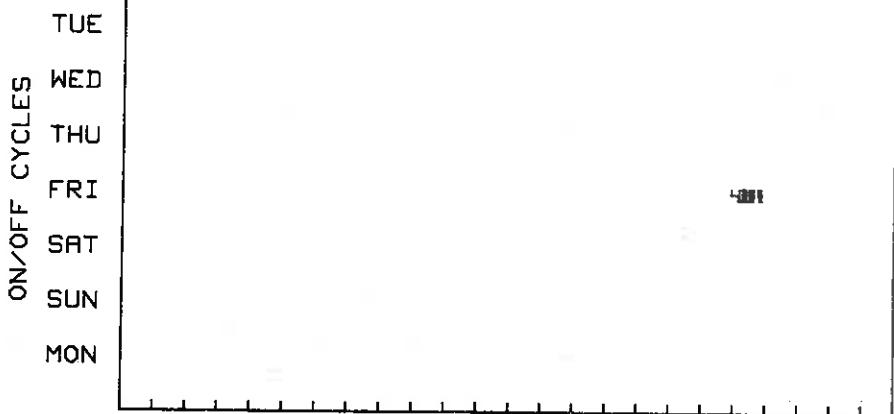
MID 2 4 6 8 10 NOON 2 4 6 8 10 MID



DURATION DISTRIBUTIONS

IRON AT NATICK HOUSE

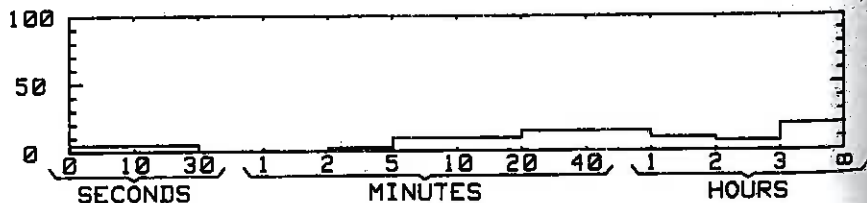
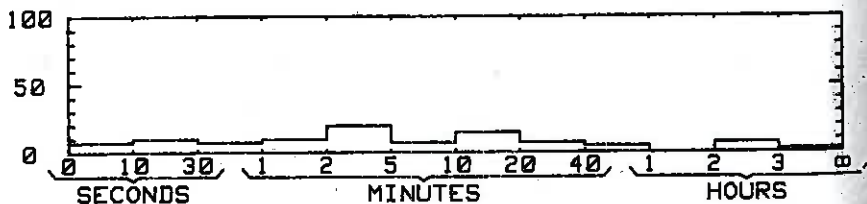
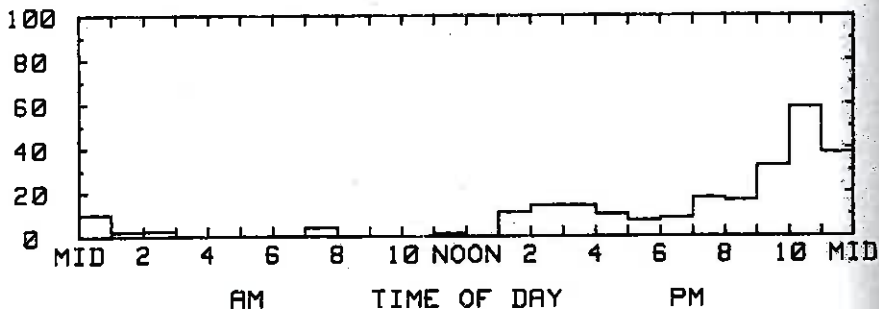
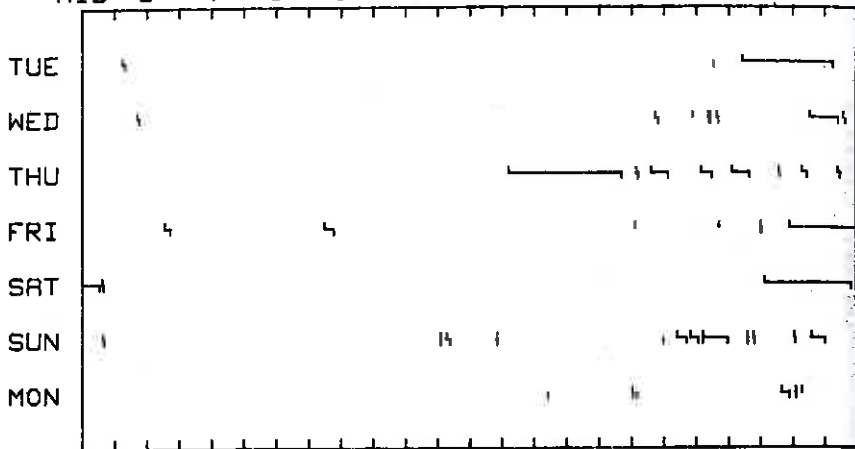
MID 2 4 6 8 10 NOON 2 4 6 8 10 MID



DURATION DISTRIBUTIONS

BATHROOM LIGHT AT NATICK HOUSE

MID 2 4 6 8 10 NOON 2 4 6 8 10 MID



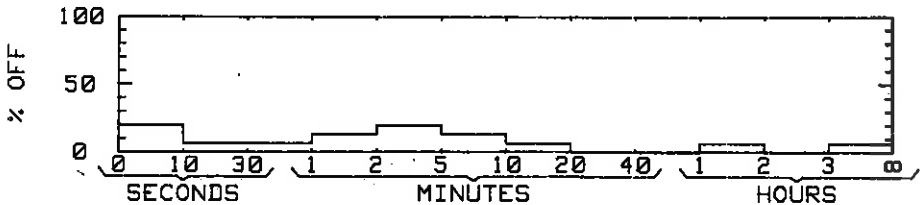
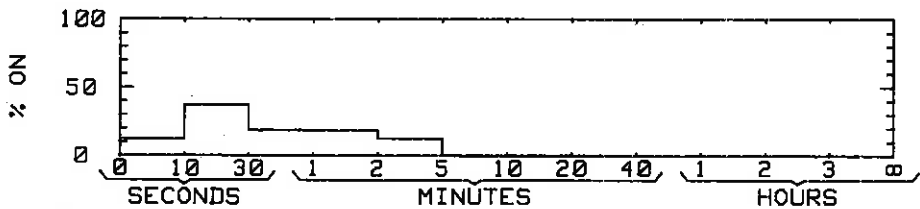
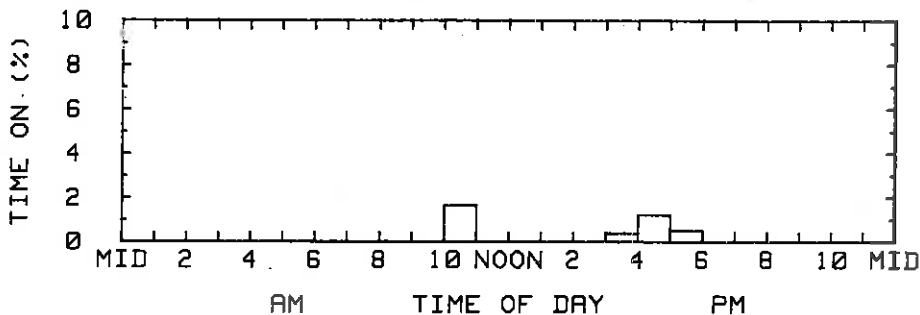
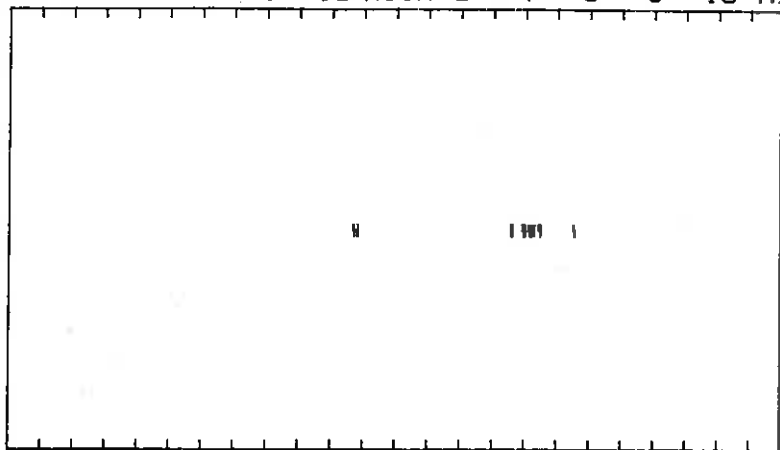
DURATION DISTRIBUTIONS

WASHING MACHINE AT NATICK HOUSE

MID 2 4 6 8 10 NOON 2 4 6 8 10 MID

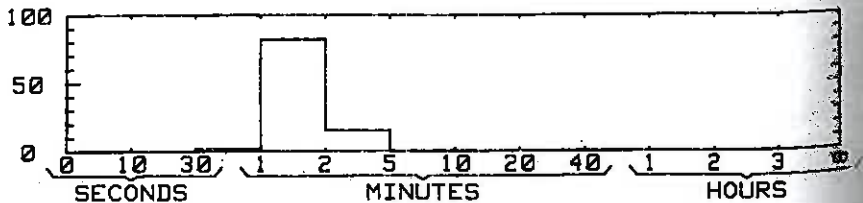
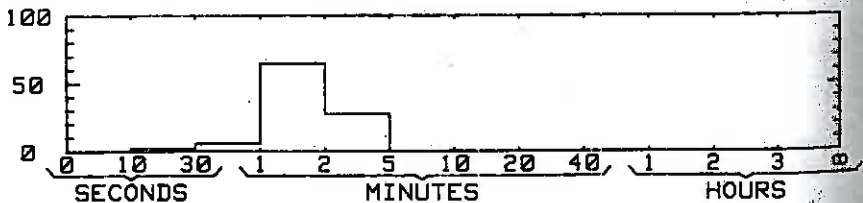
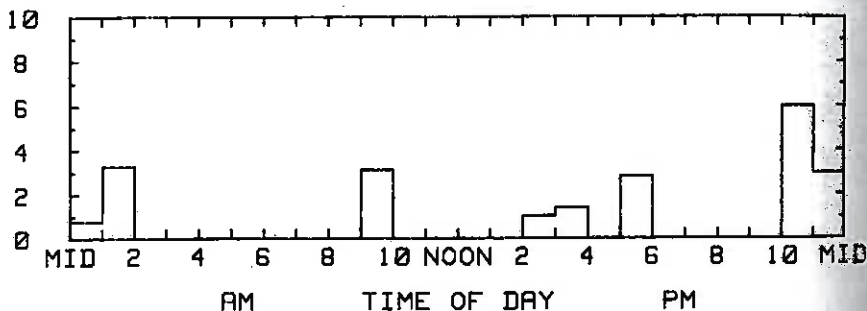
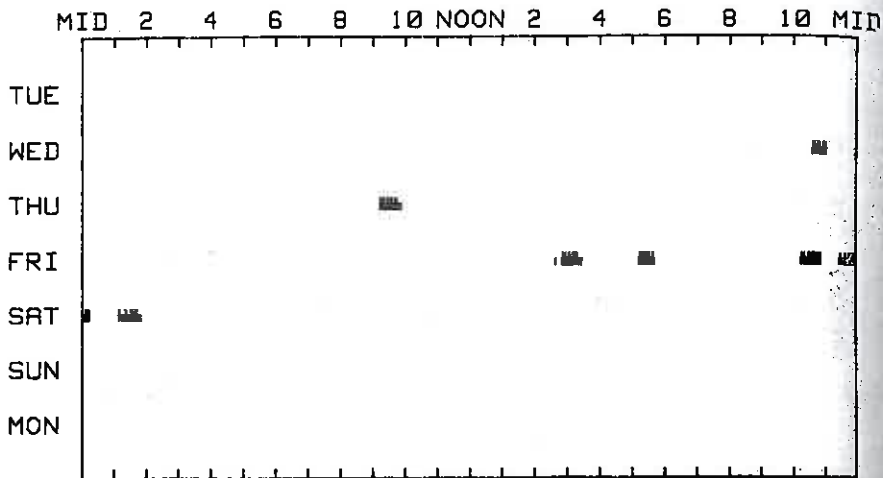
ON/OFF CYCLES

TUE
WED
THU
FRI
SAT
SUN
MON



DURATION DISTRIBUTIONS

DRIER AT NATICK HOUSE

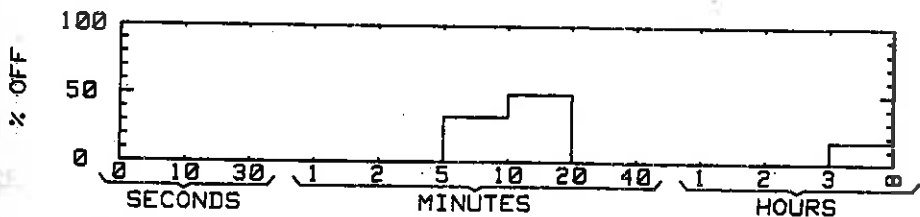
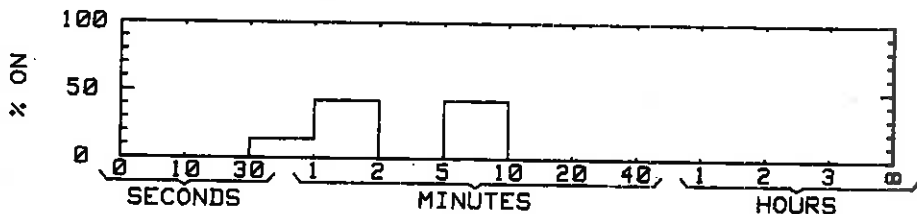
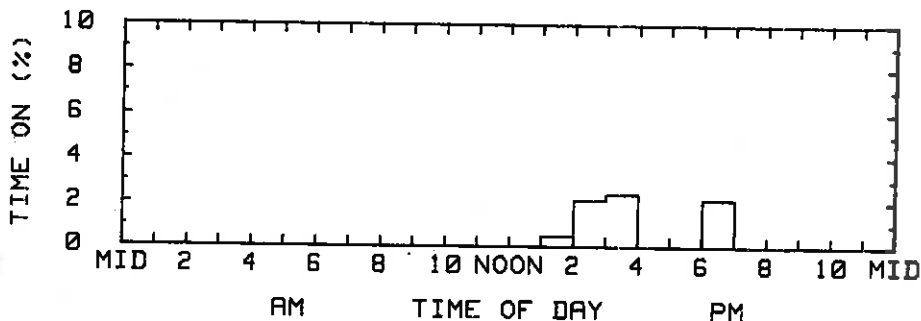
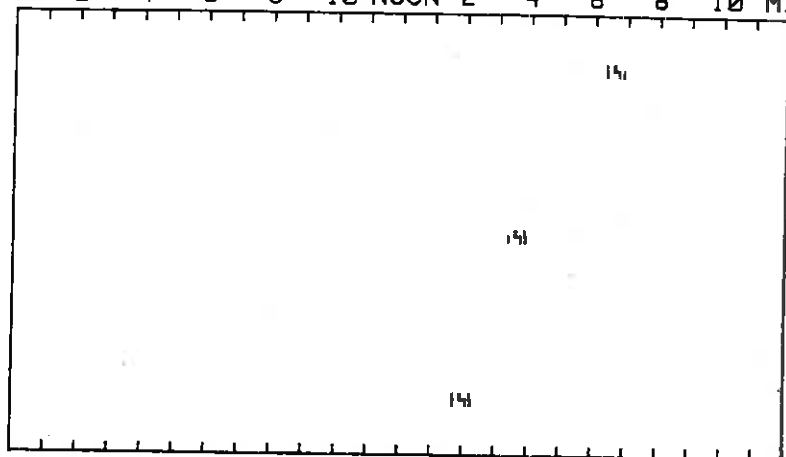


DURATION DISTRIBUTIONS

DISHWASHER COMPONENT AT NATICK HOUSE

MID 2 4 6 8 10 NOON 2 4 6 8 10 MID

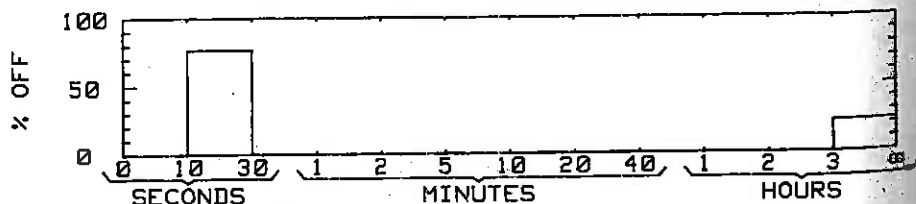
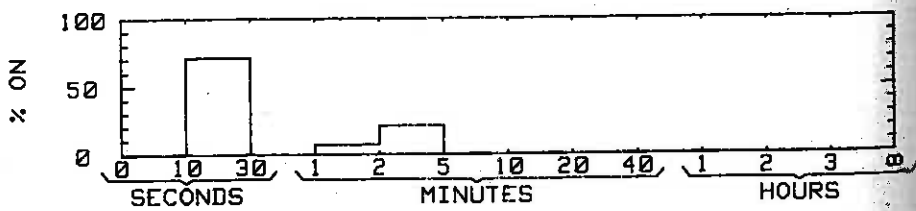
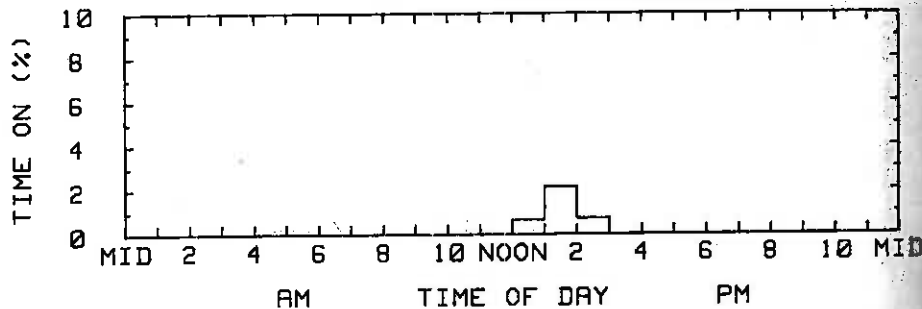
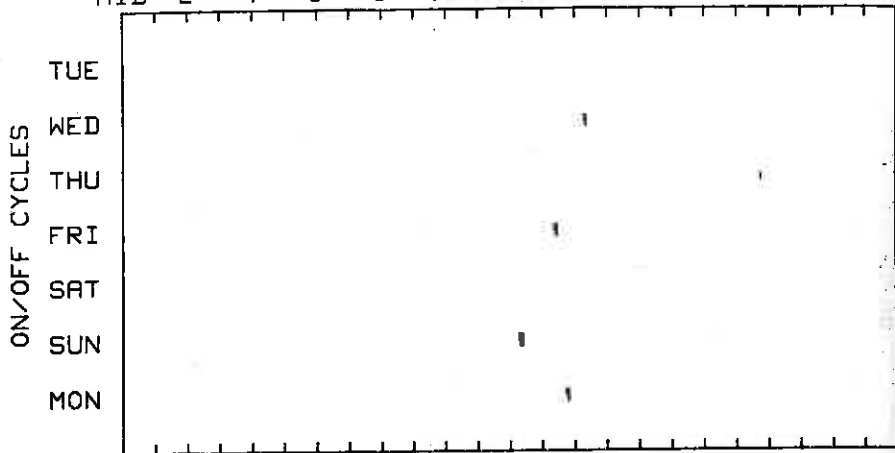
ON/OFF CYCLES
TUE
WED
THU
FRI
SAT
SUN
MON



DURATION DISTRIBUTIONS

TOASTER OVEN (BROIL) AT NATICK HOUSE

MID 2 4 6 8 10 NOON 2 4 6 8 10 MID



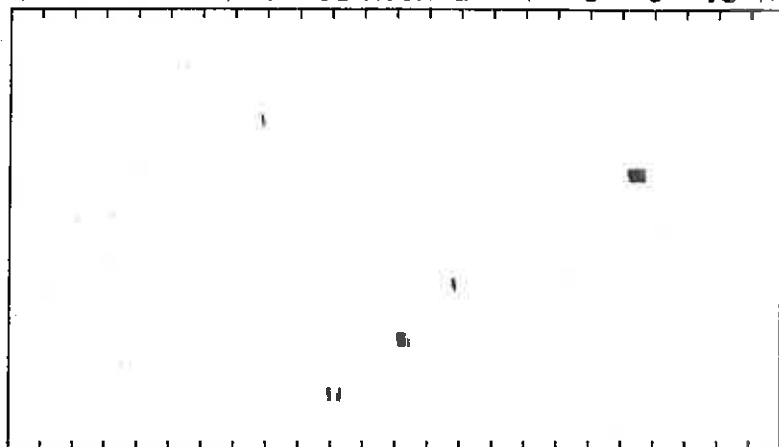
DURATION DISTRIBUTIONS

TOASTER OVEN (BAKE) AT NATICK HOUSE

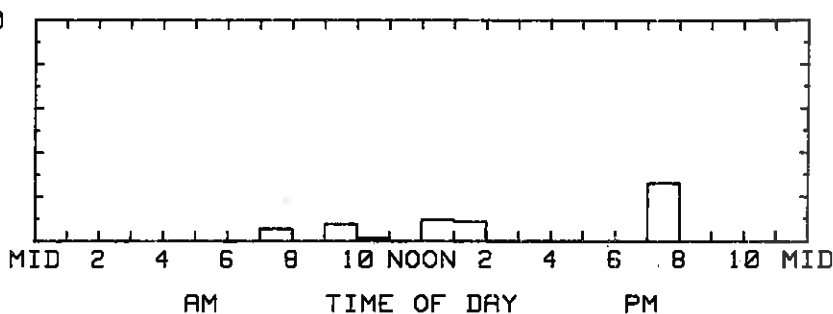
MID 2 4 6 8 10 NOON 2 4 6 8 10 MID

ON/OFF CYCLES

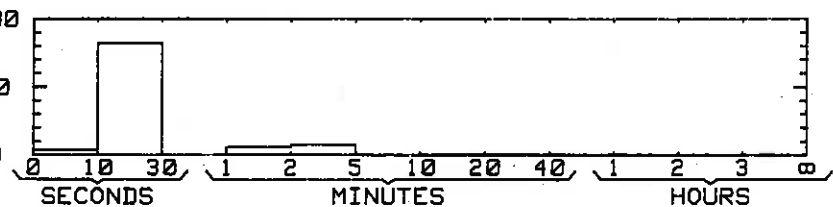
TUE
WED
THU
FRI
SAT
SUN
MON



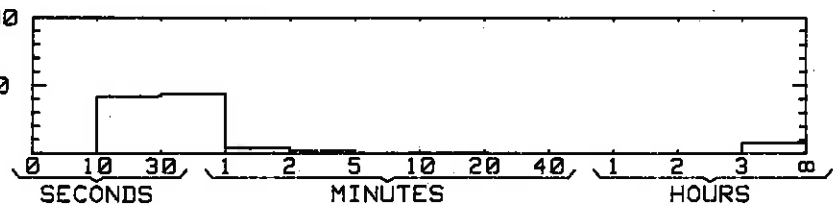
TIME ON (%)



% ON

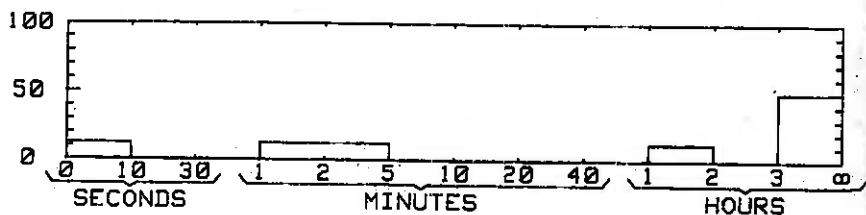
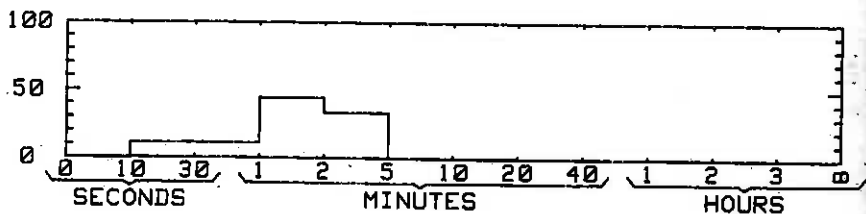
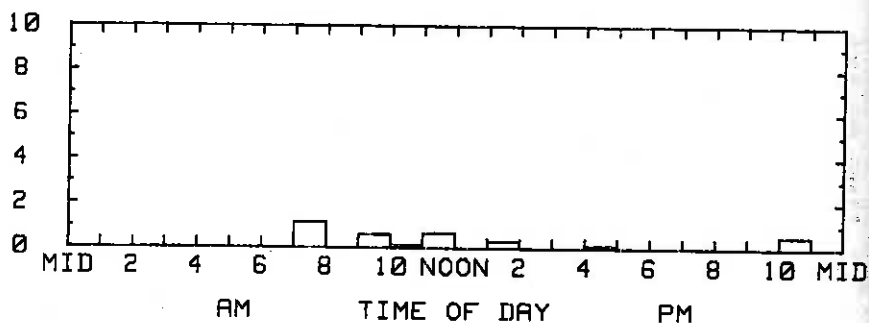
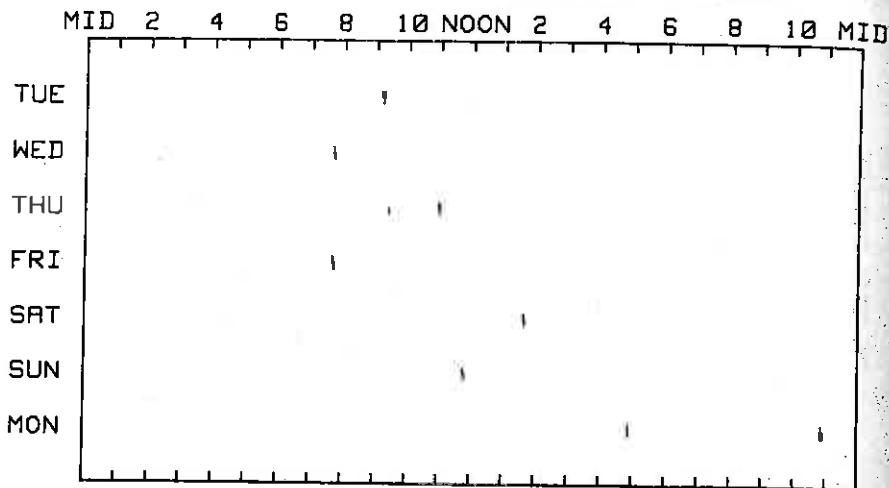


% OFF



DURATION DISTRIBUTIONS

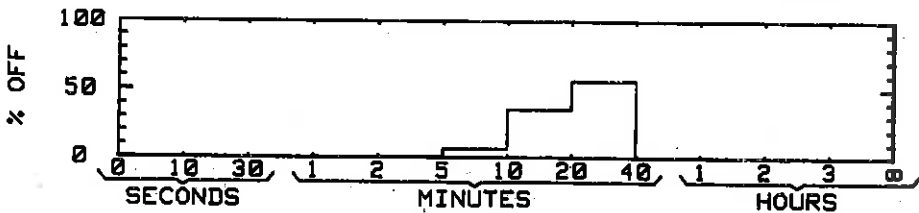
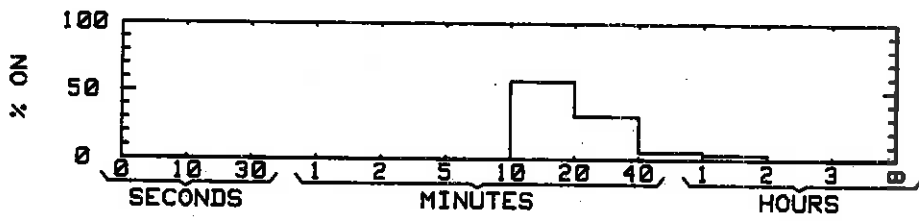
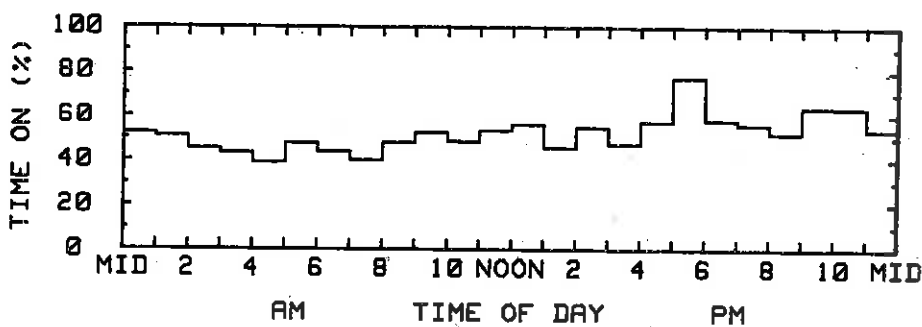
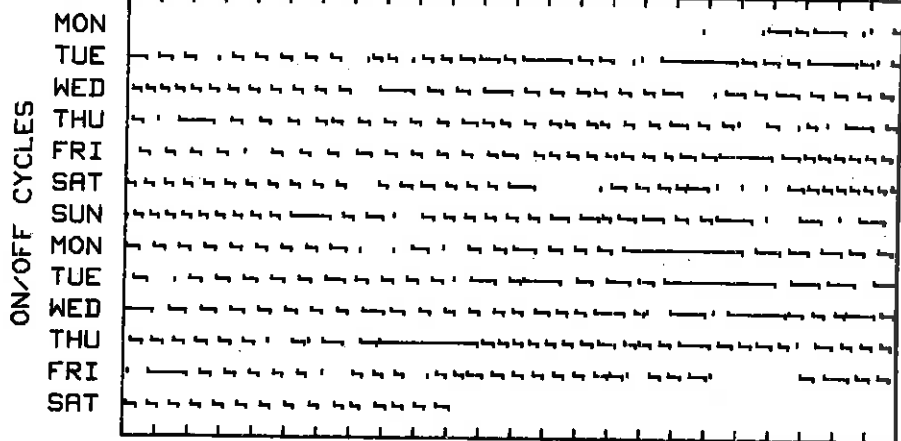
HAIR DRIER AT NATICK HOUSE



DURATION DISTRIBUTIONS

REFRIGERATOR AT LINCOLN HOUSE

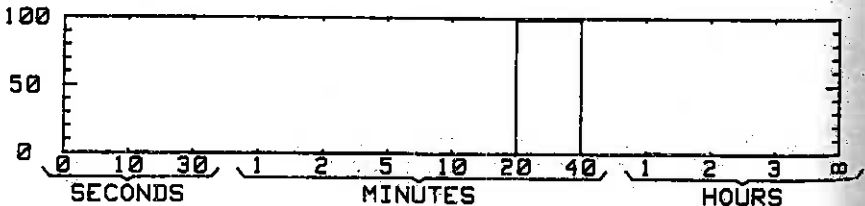
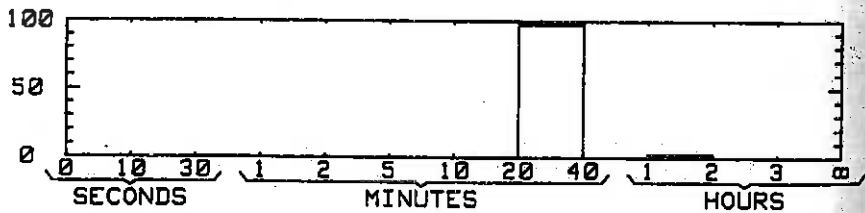
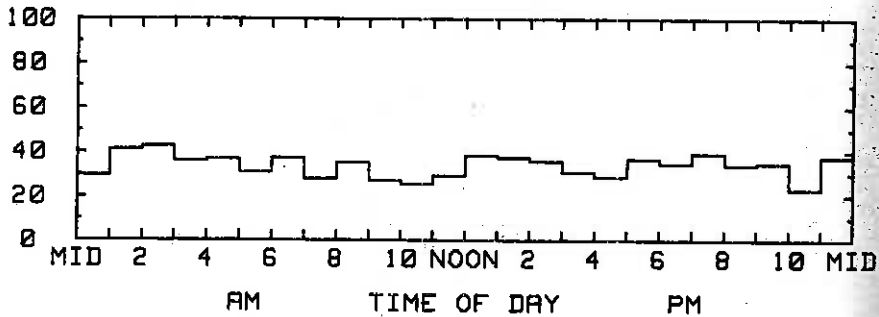
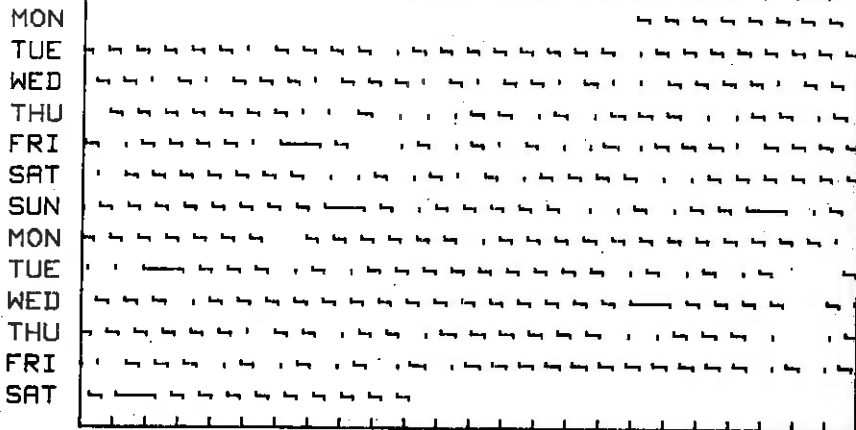
MID 2 4 6 8 10 NOON 2 4 6 8 10 MID



DURATION DISTRIBUTIONS

FREEZER AT LINCOLN HOUSE

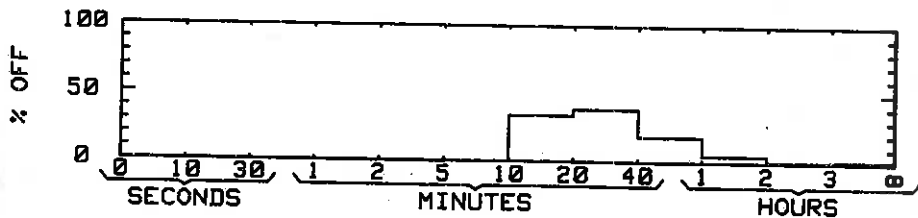
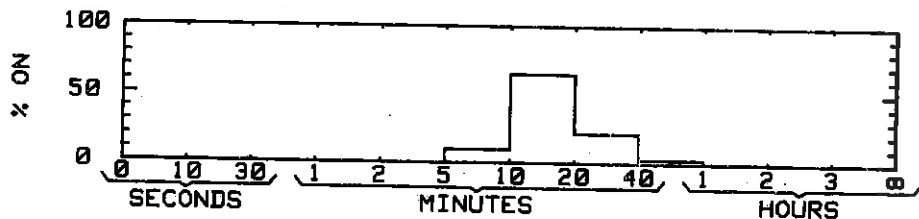
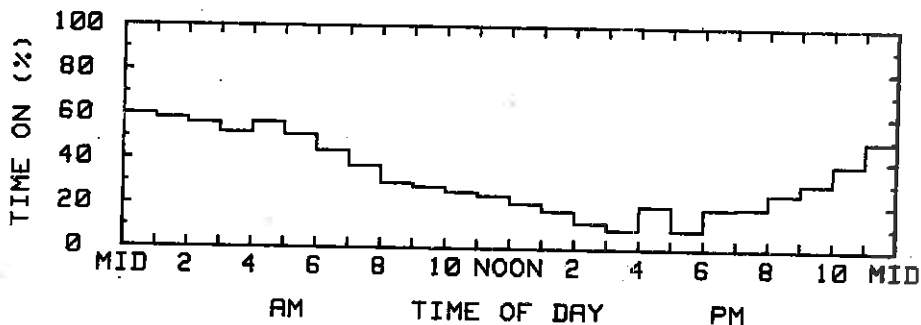
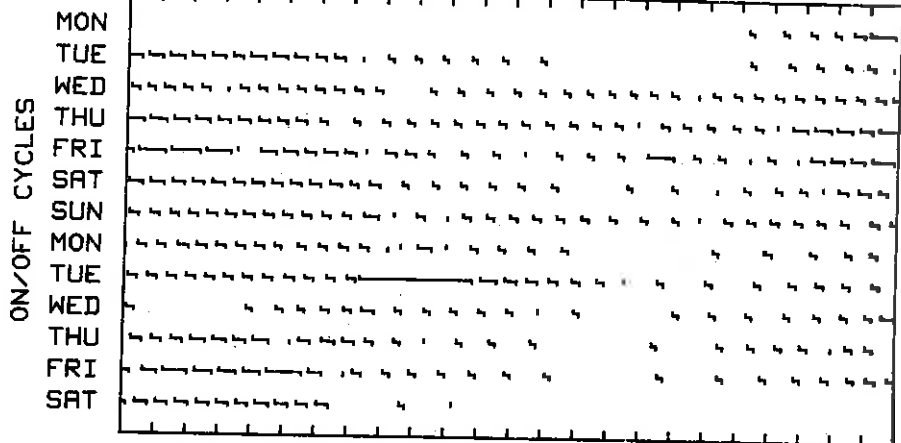
MID 2 4 6 8 10 NOON 2 4 6 8 10 MID



DURATION DISTRIBUTIONS

WATER BED AT LINCOLN HOUSE

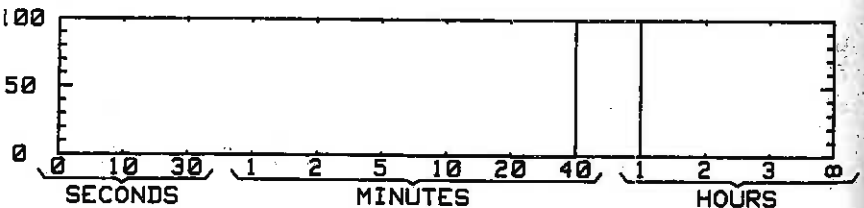
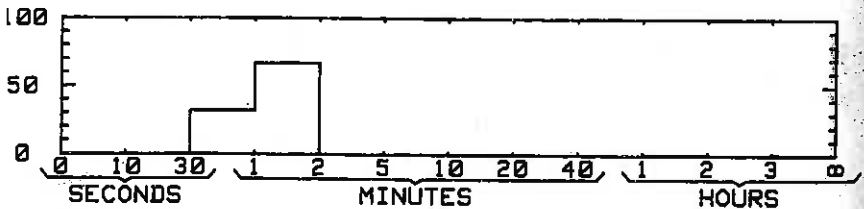
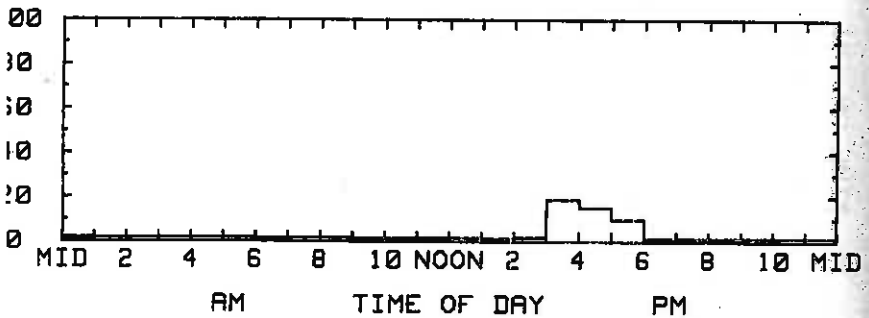
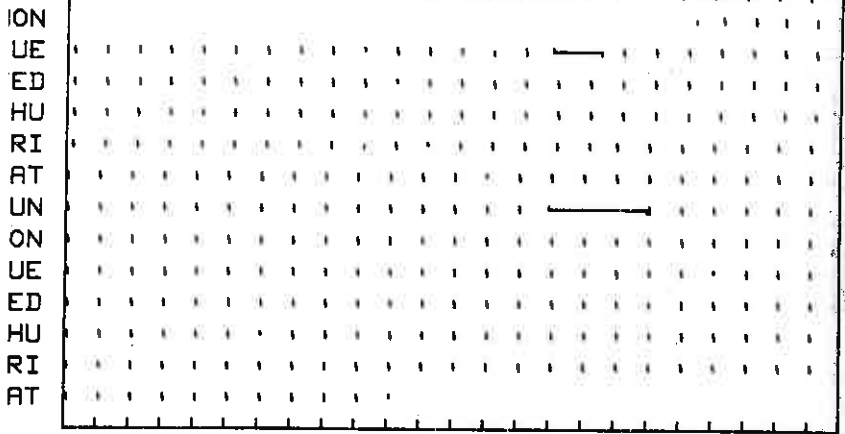
MID 2 4 6 8 10 NOON 2 4 6 8 10 MID



DURATION DISTRIBUTIONS

HOT TUB PUMP AT LINCOLN HOUSE

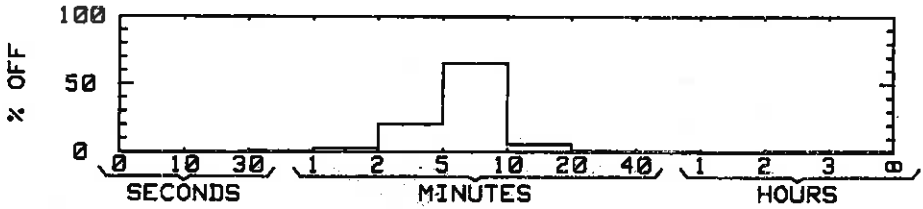
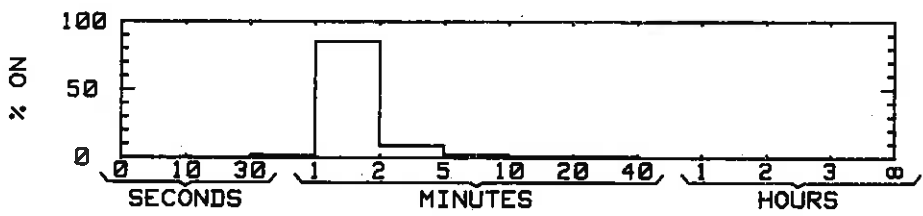
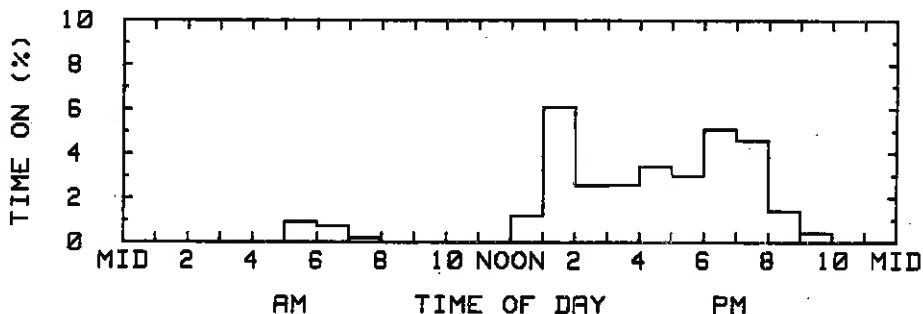
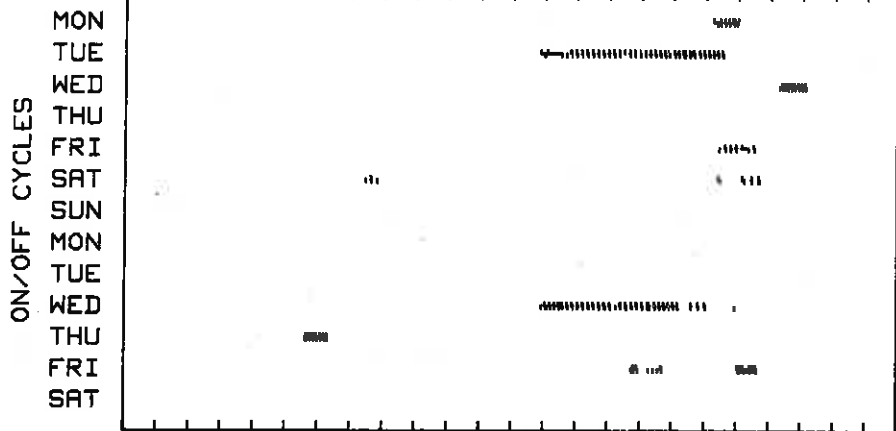
MID 2 4 6 8 10 NOON 2 4 6 8 10 MID



DURATION DISTRIBUTIONS

OVEN 1 AT LINCOLN HOUSE

MID 2 4 6 8 10 NOON 2 4 6 8 10 MID

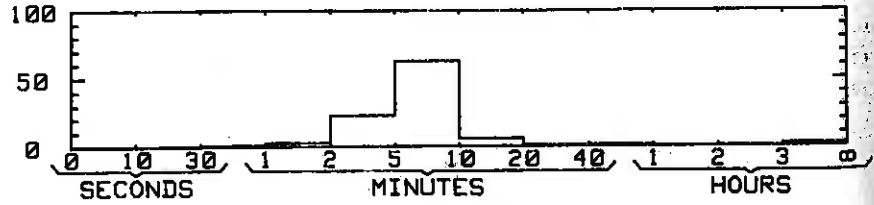
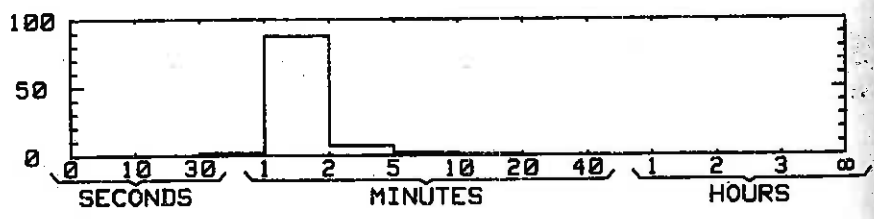
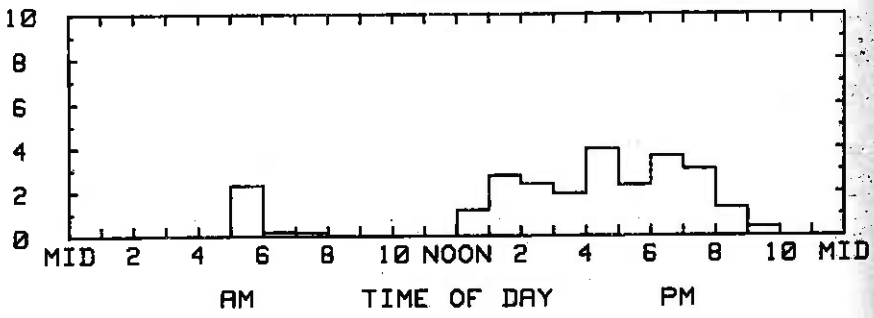
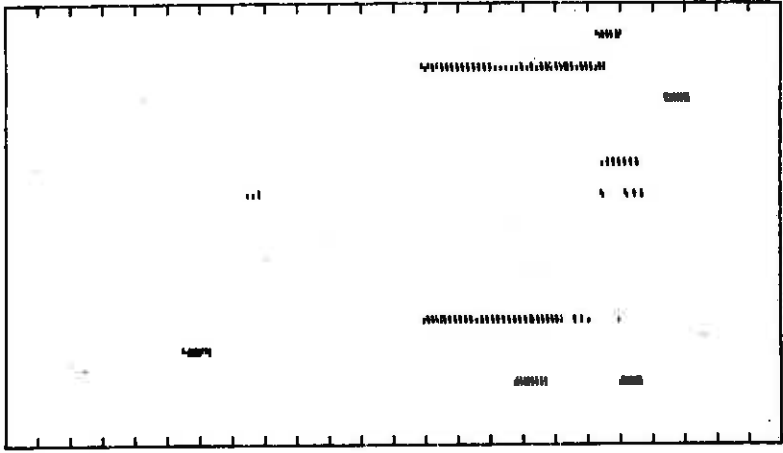


DURATION DISTRIBUTIONS

OVEN 2 AT LINCOLN HOUSE

MID 2 4 6 8 10 NOON 2 4 6 8 10 MID

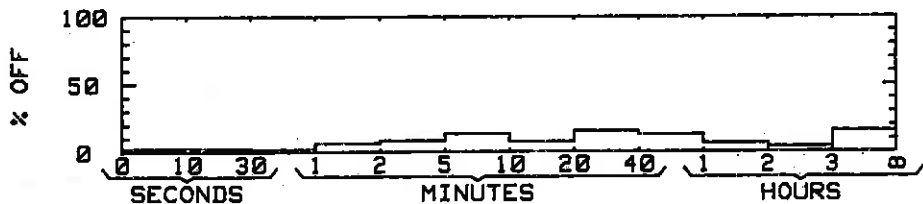
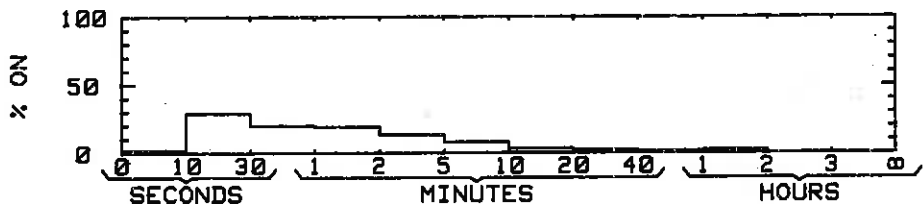
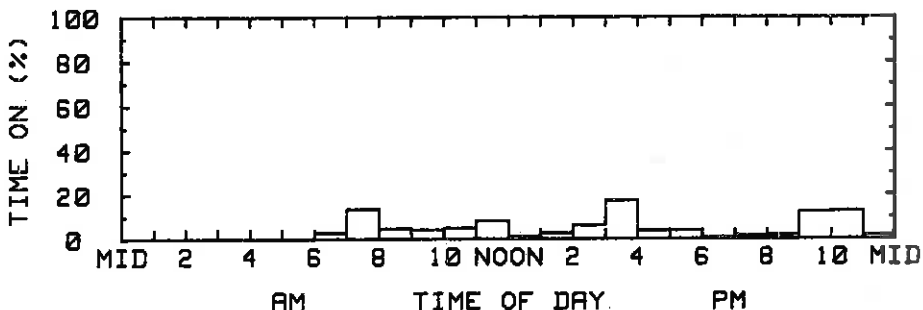
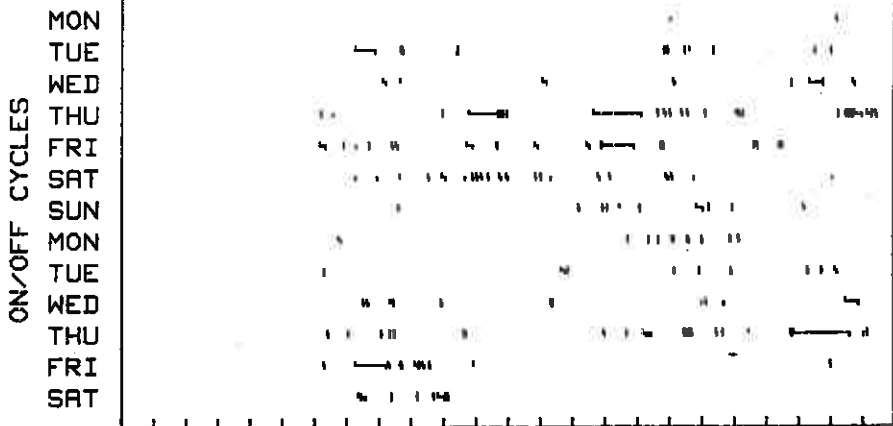
MON
TUE
WED
THU
FRI
SAT
SUN
MON
TUE
WED
THU
FRI
SAT



DURATION DISTRIBUTIONS

BASEMENT LIGHTS AT LINCOLN HOUSE

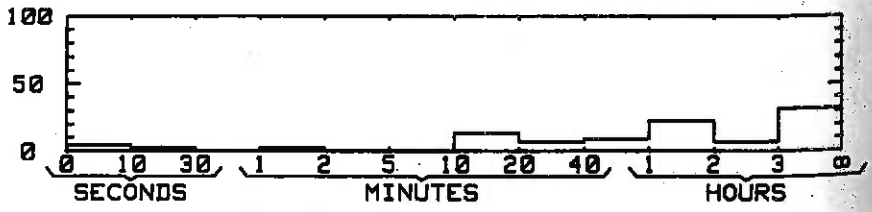
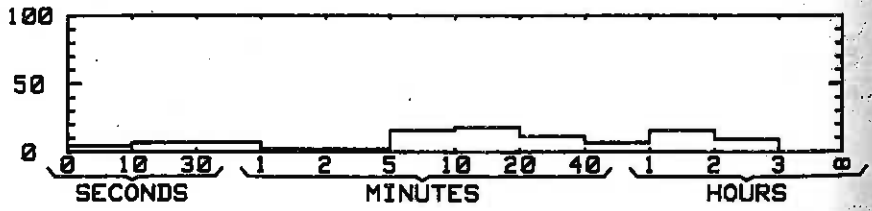
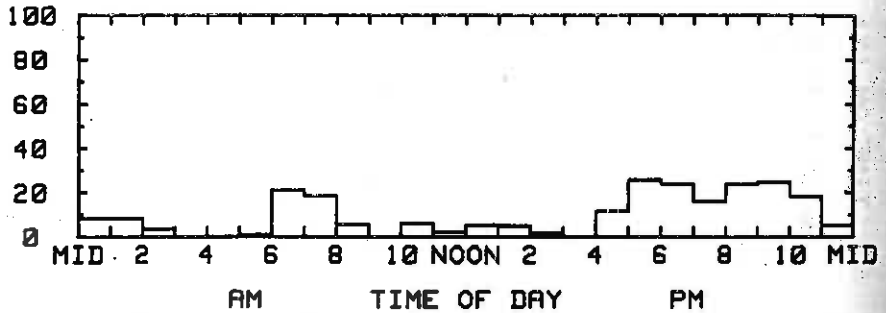
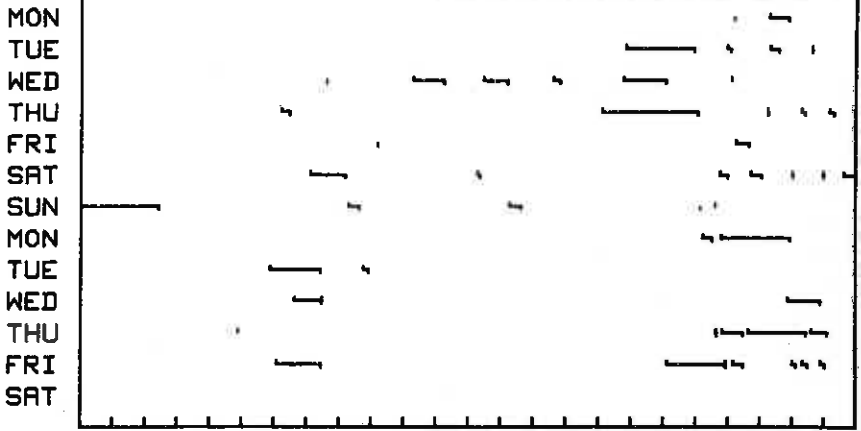
MID 2 4 6 8 10 NOON 2 4 6 8 10 MID



DURATION DISTRIBUTIONS

KITCHEN LIGHT AT LINCOLN HOUSE

MID 2 4 6 8 10 NOON 2 4 6 8 10 MID

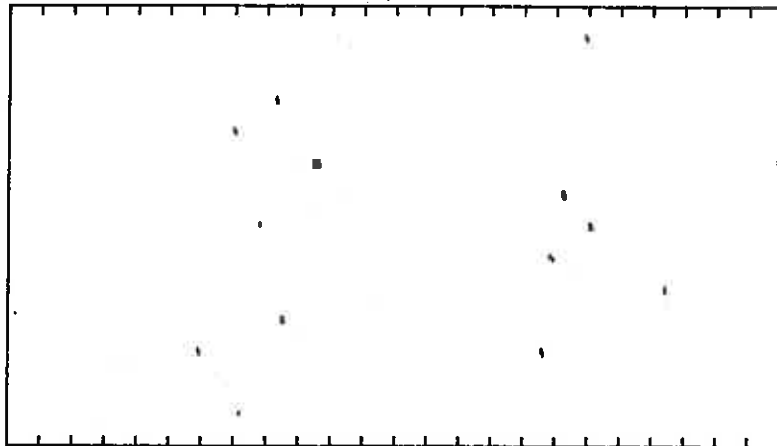


DURATION DISTRIBUTIONS

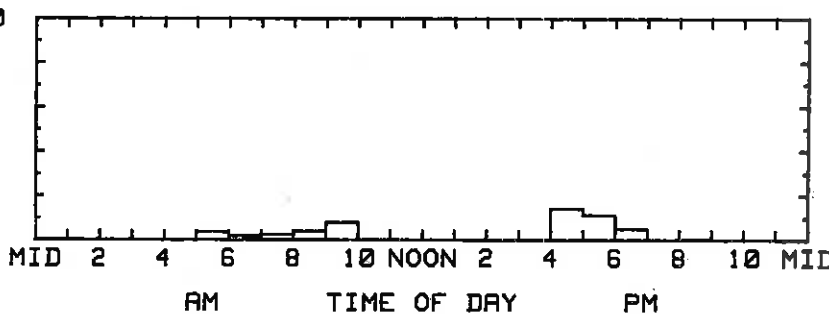
HAIR DRYER AT LINCOLN HOUSE

MID 2 4 6 8 10 NOON 2 4 6 8 10 MID

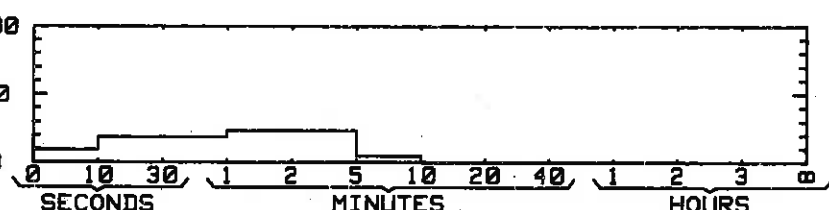
ON/OFF CYCLES
 MON
 TUE
 WED
 THU
 FRI
 SAT
 SUN
 MON
 TUE
 WED
 THU
 FRI
 SAT



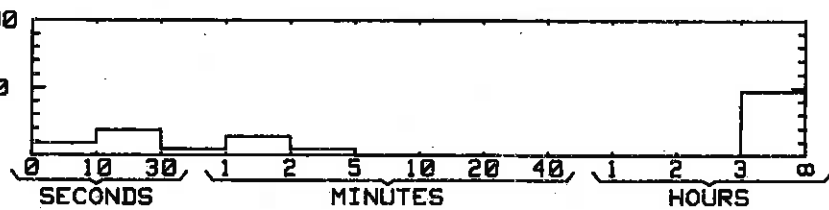
TIME ON (%)



% ON

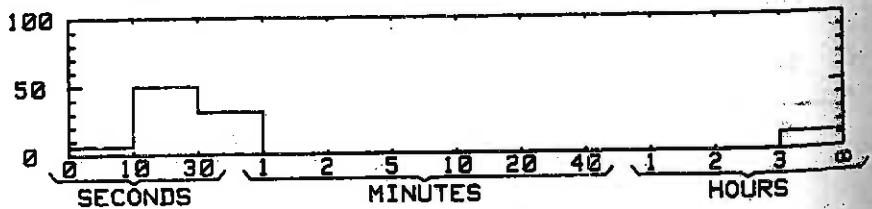
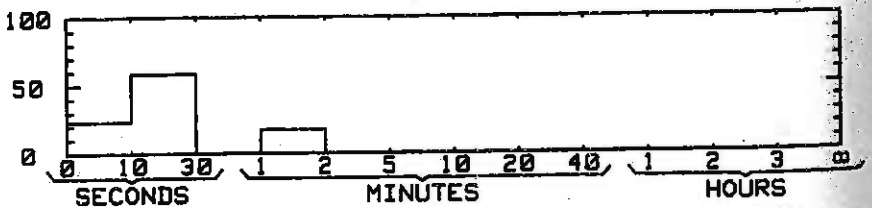
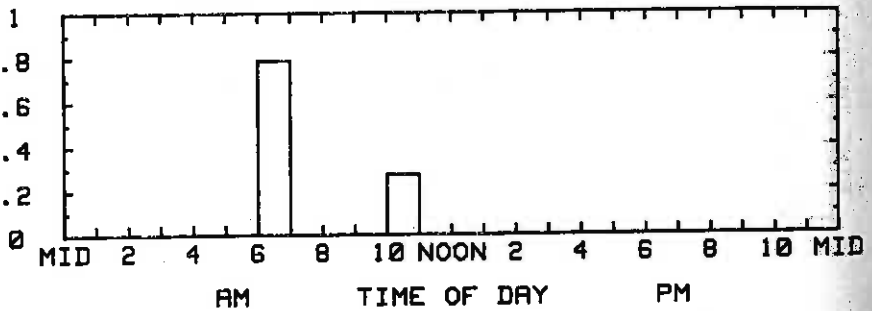
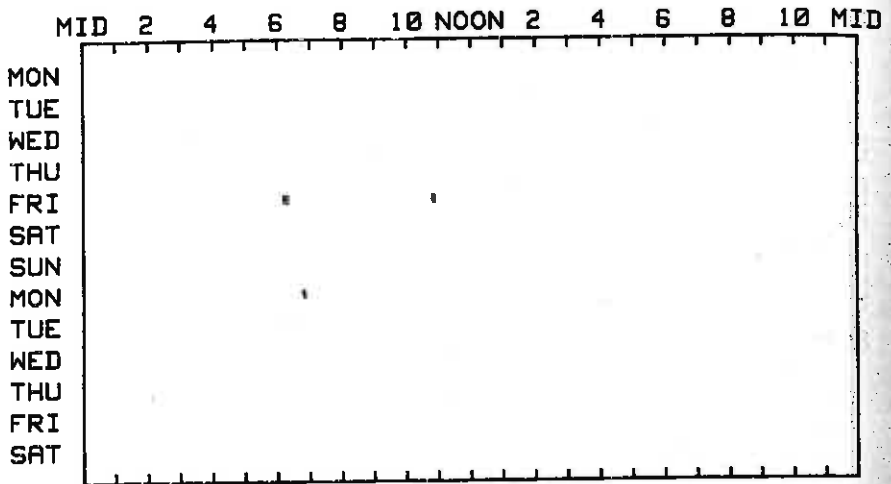


% OFF



DURATION DISTRIBUTIONS

IRON AT LINCOLN HOUSE



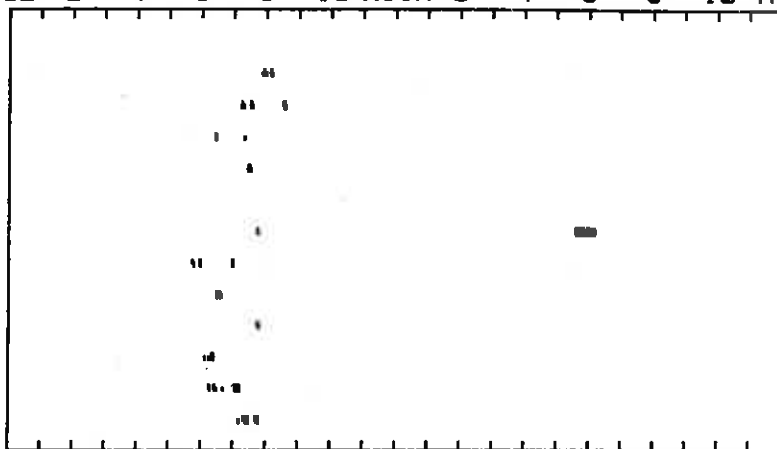
DURATION DISTRIBUTIONS

TOASTER OVEN AT LINCOLN HOUSE

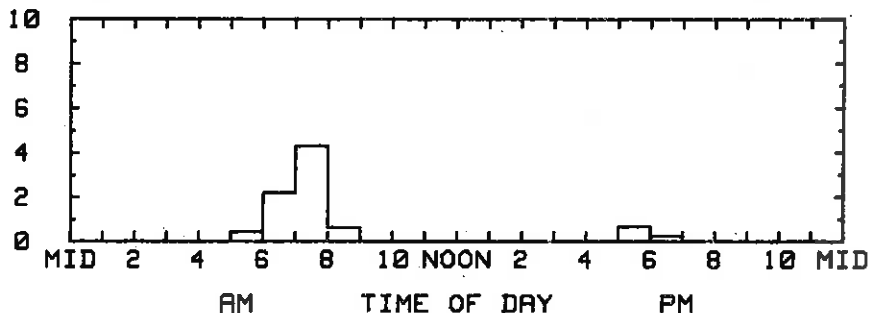
MID 2 4 6 8 10 NOON 2 4 6 8 10 MID

ON/OFF CYCLES

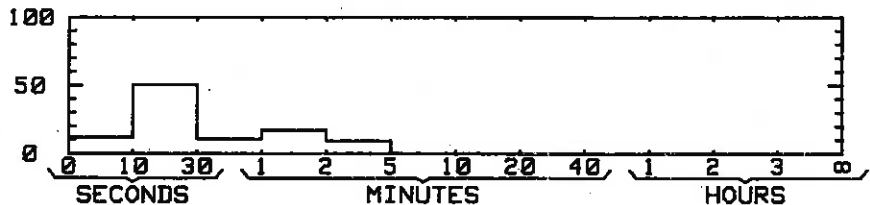
MON
TUE
WED
THU
FRI
SAT
SUN
MON
TUE
WED
THU
FRI
SAT



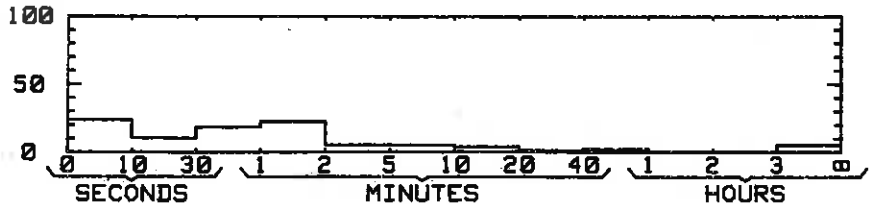
TIME ON (%)



% ON



% OFF

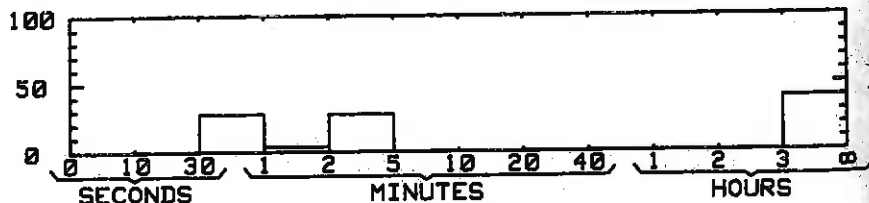
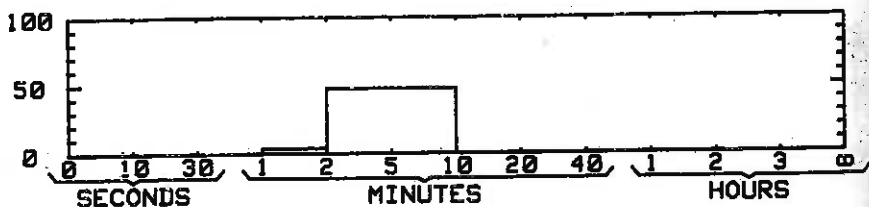
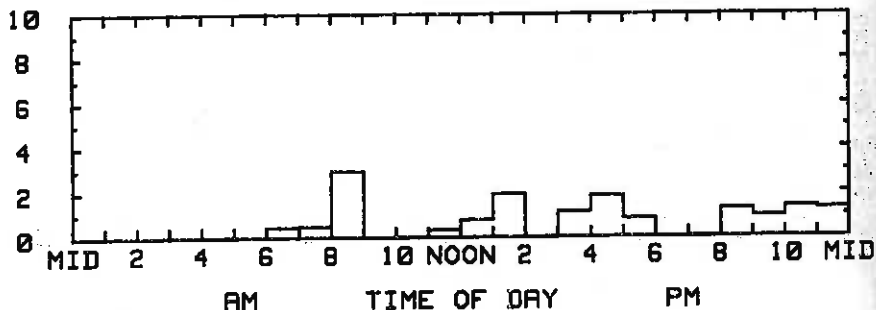
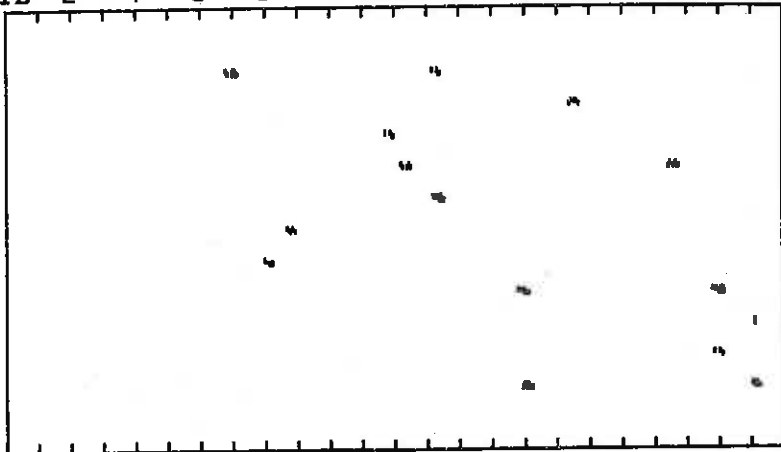


DURATION DISTRIBUTIONS

DISHWASHER COMPONENT AT LINCOLN HOUSE

MID 2 4 6 8 10 NOON 2 4 6 8 10 MID

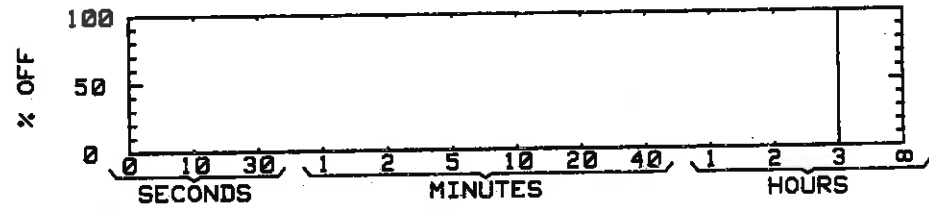
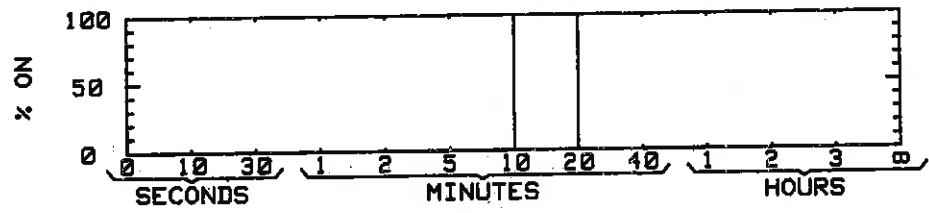
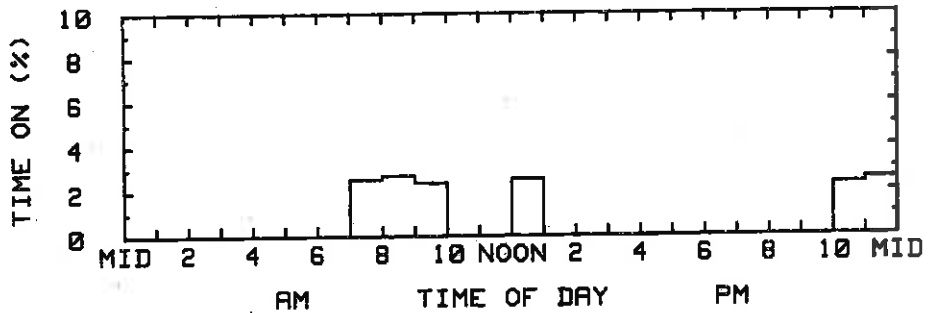
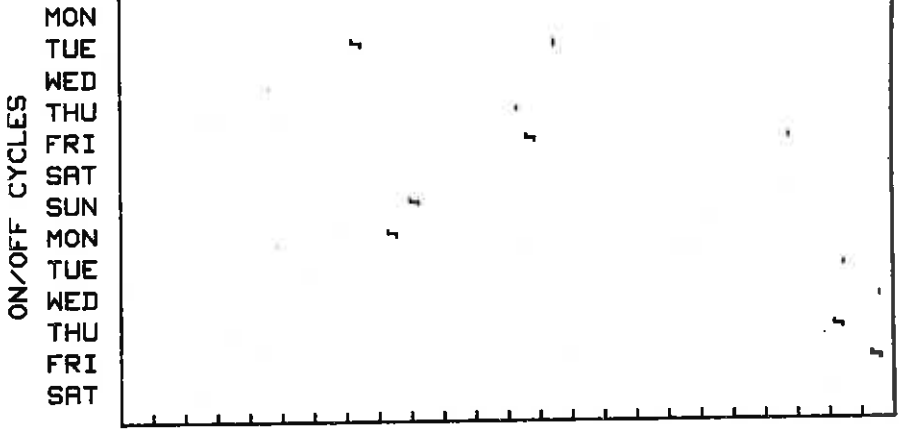
MON
TUE
WED
THU
FRI
SAT
SUN
MON
TUE
WED
THU
FRI
SAT



DURATION DISTRIBUTIONS

DISHWASHER HEATER AT LINCOLN HOUSE

MID 2 4 6 8 10 NOON 2 4 6 8 10 MID

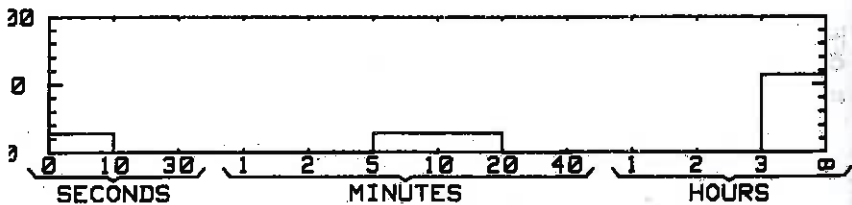
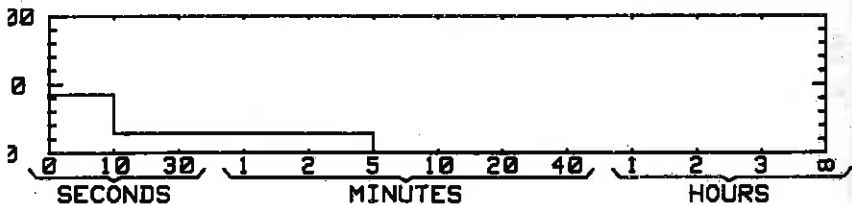
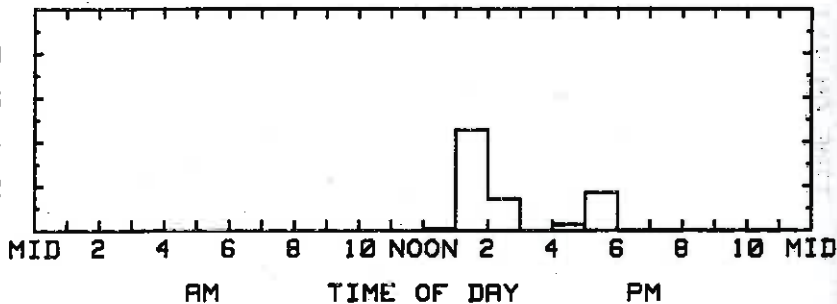
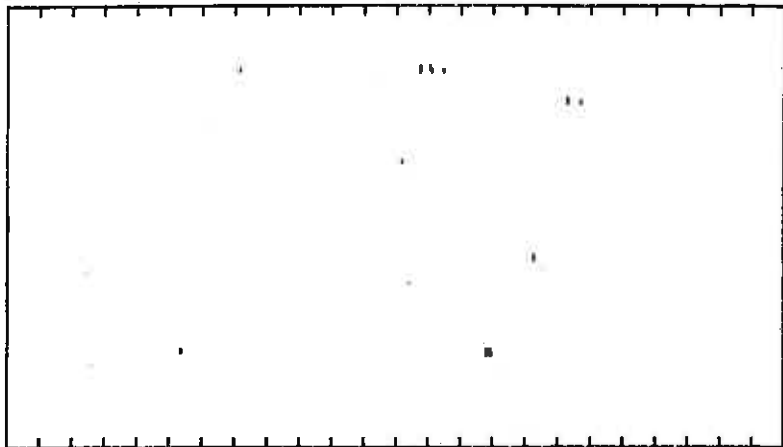


DURATION DISTRIBUTIONS

FOOD PROCESSOR AT LINCOLN HOUSE

MID 2 4 6 8 10 NOON 2 4 6 8 10 MID

ON
 JE
 ED
 HU
 RI
 IT
 IN
 N
 JE
 ED
 HU
 RI
 IT



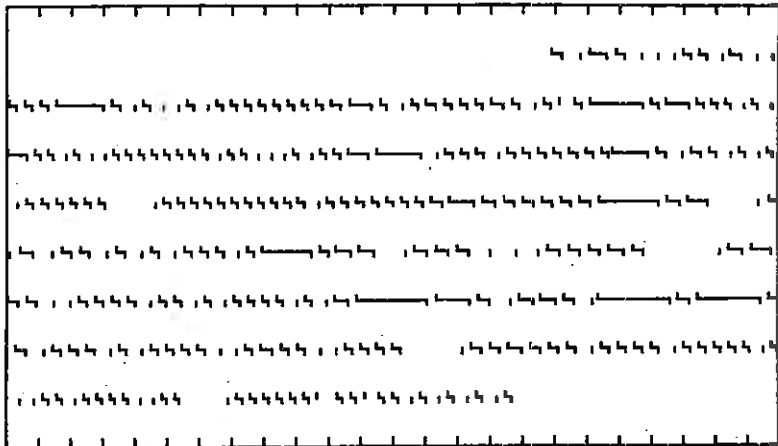
DURATION DISTRIBUTIONS

REFRIGERATOR AT ACTON HOUSE

MID 2 4 6 8 10 NOON 2 4 6 8 10 MID

ON/OFF CYCLES

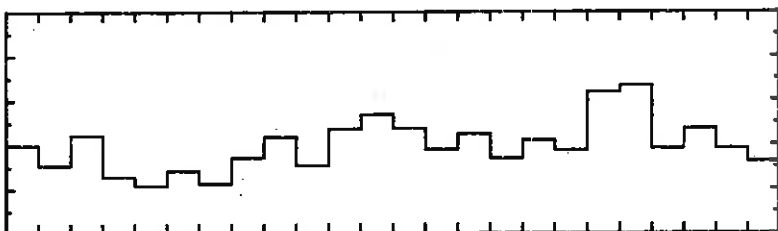
TUE
WED
THU
FRI
SAT
SUN
MON
TUE



TIME ON (%)

100
80
60
40
20
0

MID 2 4 6 8 10 NOON 2 4 6 8 10 MID
AM TIME OF DAY PM



% ON

100
50
0

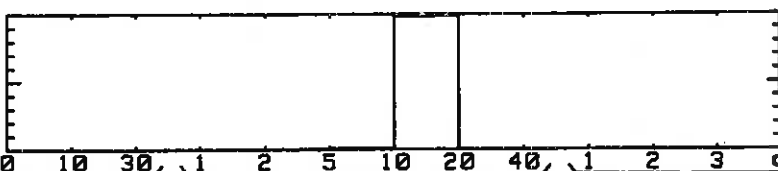
SECONDS MINUTES HOURS



% OFF

100
50
0

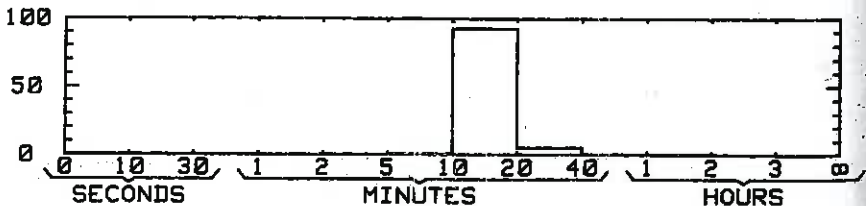
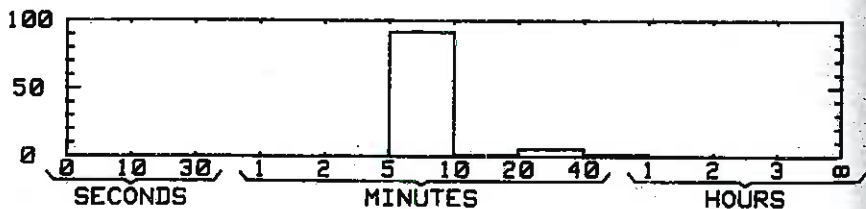
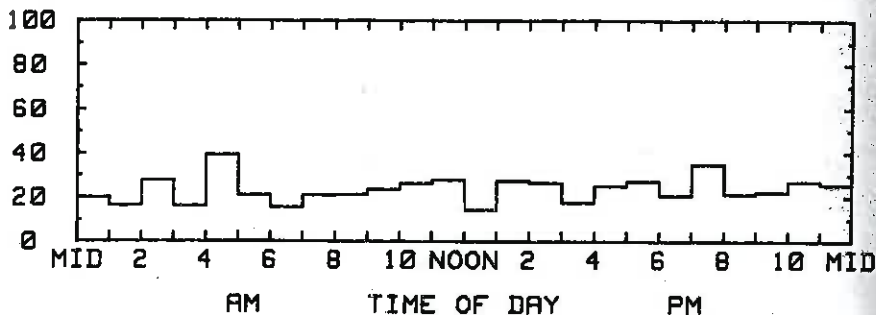
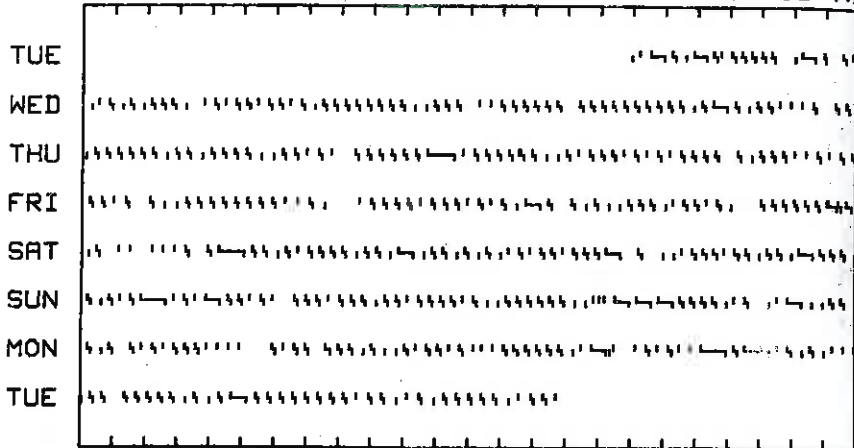
SECONDS MINUTES HOURS



DURATION DISTRIBUTIONS

SMALL REFRIGERATOR AT ACTON HOUSE

MID 2 4 6 8 10 NOON 2 4 6 8 10 MID

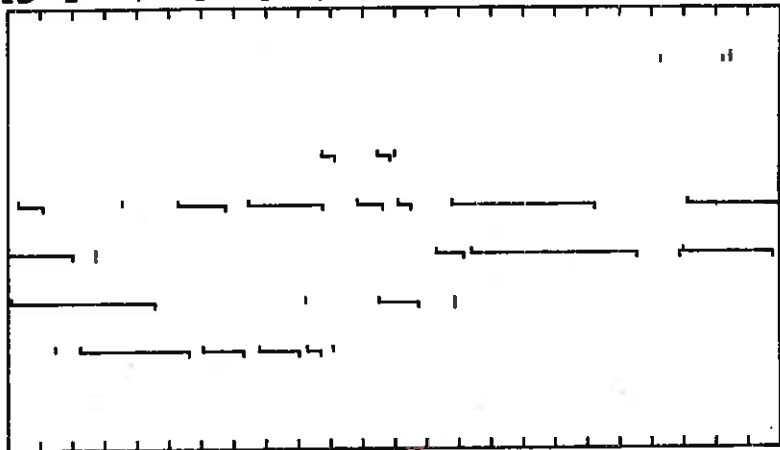


DURATION DISTRIBUTIONS

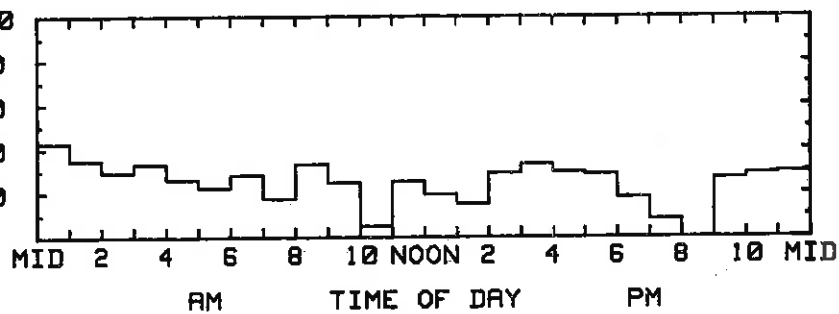
ICE MAKER AT ACTON HOUSE

MID 2 4 6 8 10 NOON 2 4 6 8 10 MID

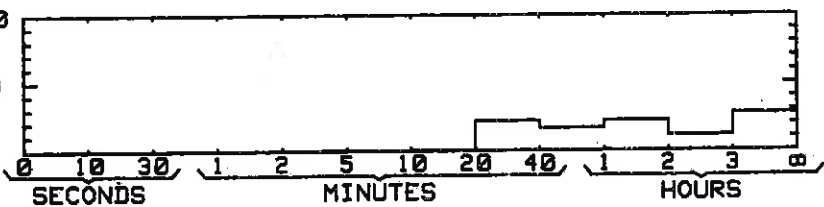
ON/OFF CYCLES



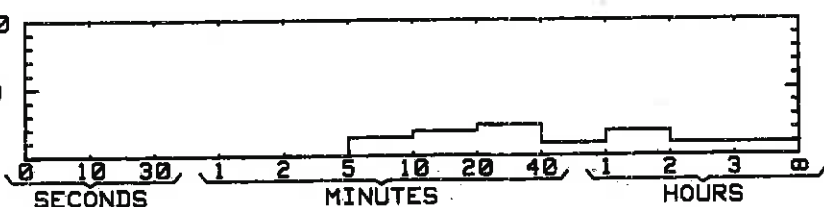
TIME ON (%)



% ON



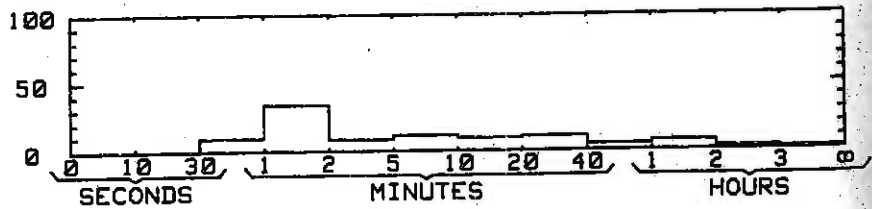
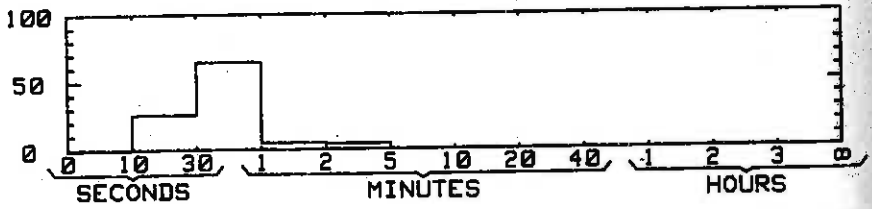
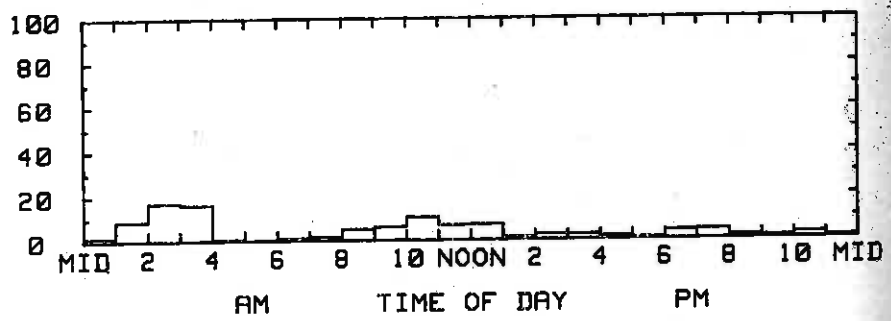
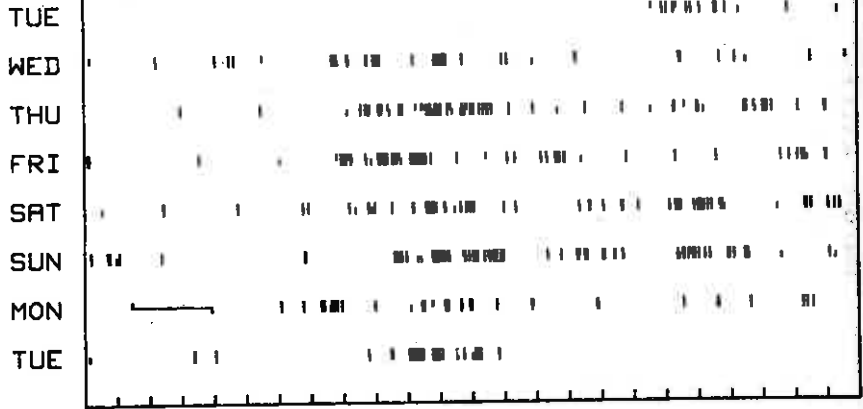
% OFF



DURATION DISTRIBUTIONS

WATER PUMP AT ACTON HOUSE

MID 2 4 6 8 10 NOON 2 4 6 8 10 MID



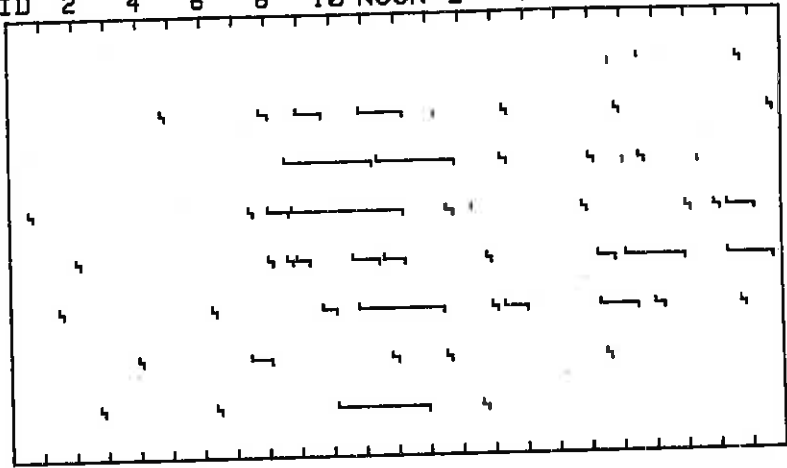
DURATION DISTRIBUTIONS

WATER HEATER AT ACTON HOUSE

MID 2 4 6 8 10 NOON 2 4 6 8 10 MID

ON/OFF CYCLES

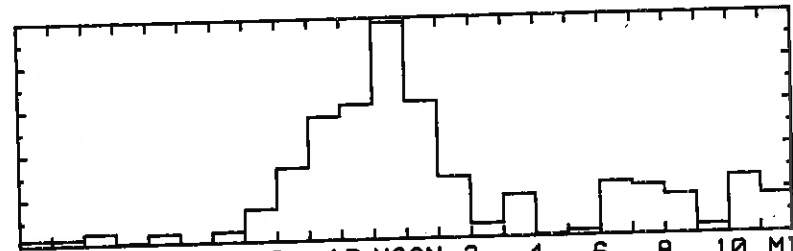
TUE
WED
THU
FRI
SAT
SUN
MON
TUE



TIME ON (%)

100
80
60
40
20
0

MID 2 4 6 8 10 NOON 2 4 6 8 10 MID
AM TIME OF DAY PM



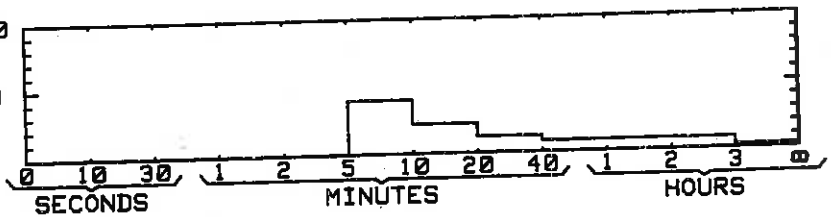
% ON

100
50
0

SECONDS

MINUTES

HOURS



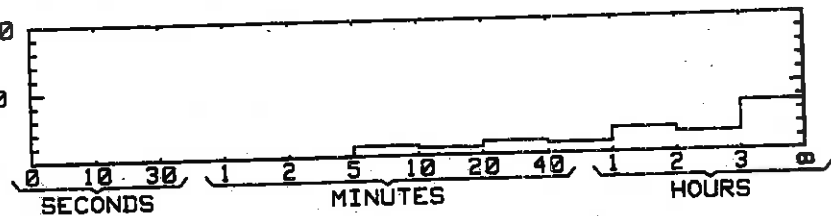
% OFF

100
50
0

SECONDS

MINUTES

HOURS

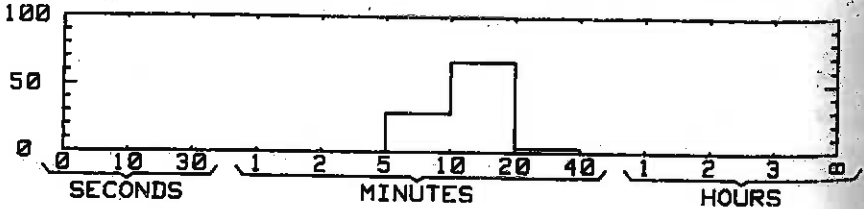
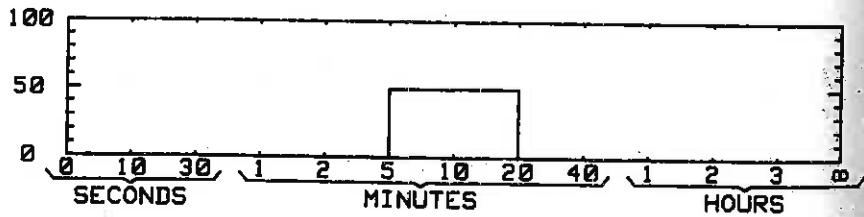
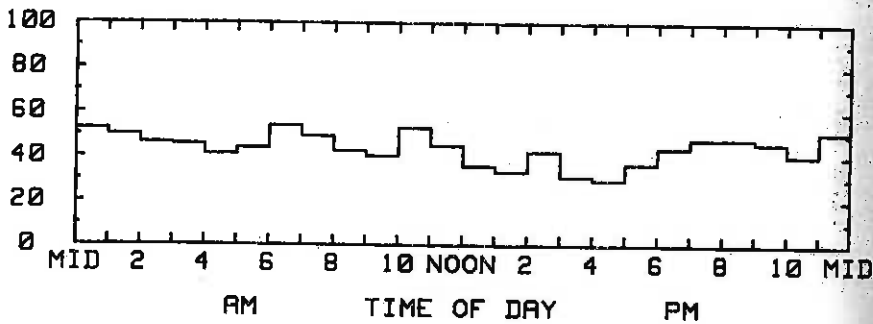
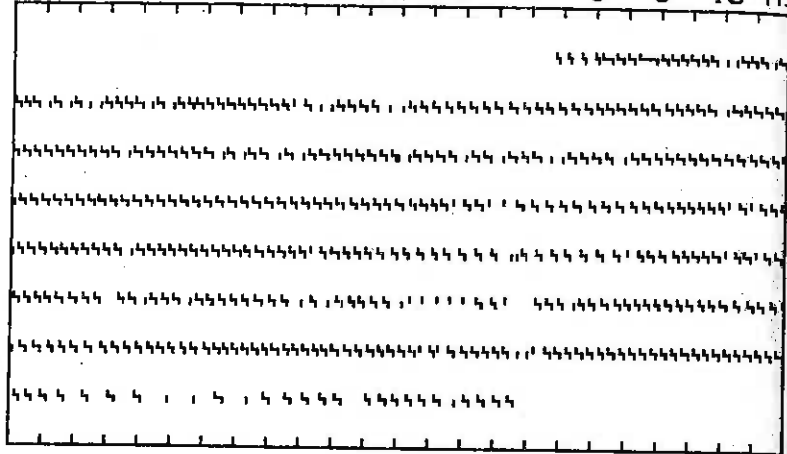


DURATION DISTRIBUTIONS

DEHUMIDIFIER AT ACTON HOUSE

MID 2 4 6 8 10 NOON 2 4 6 8 10 MID

TUE
 WED
 THU
 FRI
 SAT
 SUN
 MON
 TUE

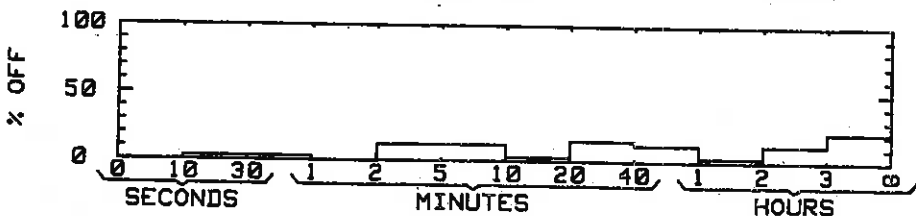
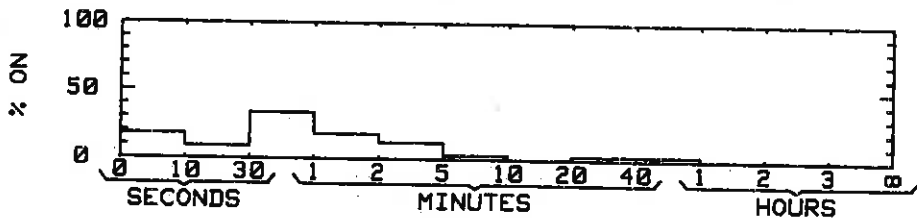
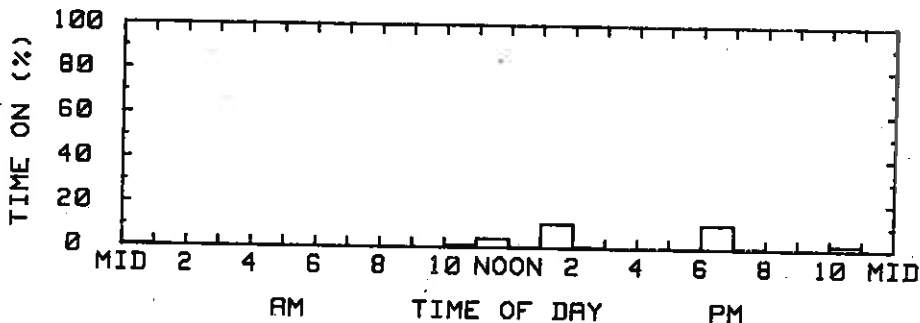
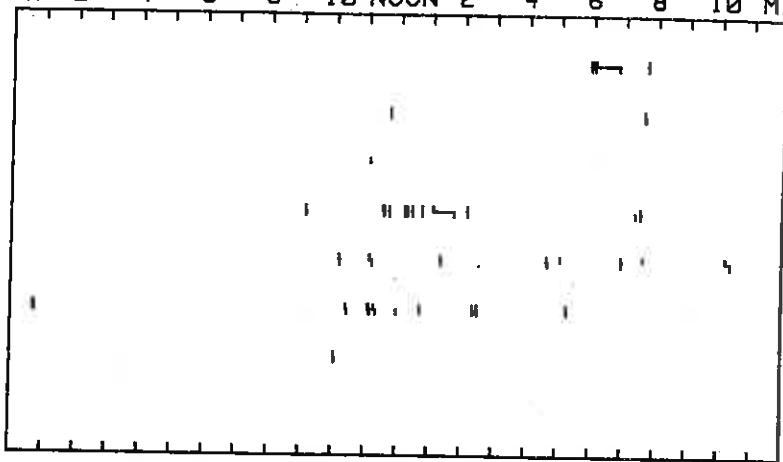


DURATION DISTRIBUTIONS

BASEMENT BATHROOM LIGHT AT ACTON HOUSE

MID 2 4 6 8 10 NOON 2 4 6 8 10 MID

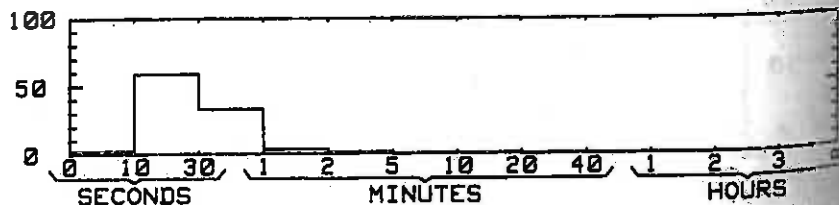
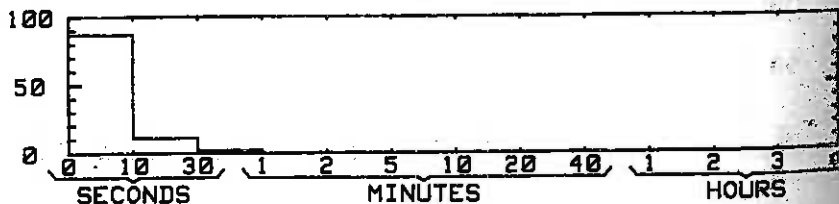
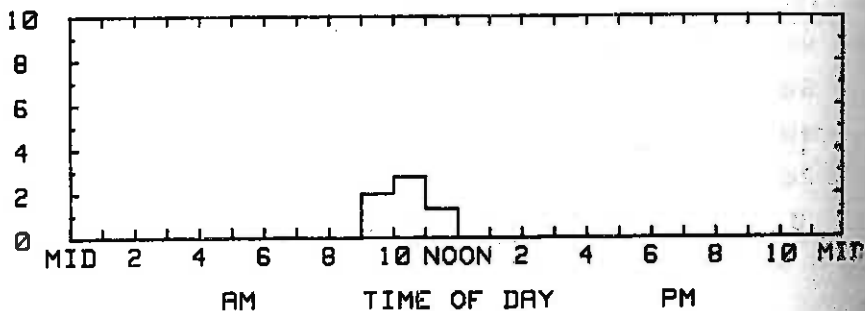
ON/OFF CYCLES
TUE
WED
THU
FRI
SAT
SUN
MON
TUE



DURATION DISTRIBUTIONS

IRON AT ACTON HOUSE

MID 2 4 6 8 10 NOON 2 4 6 8 10 MID



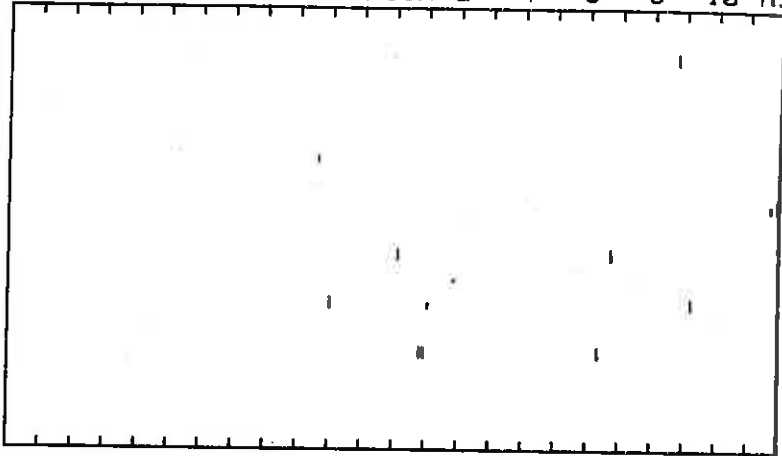
DURATION DISTRIBUTIONS

GARAGE DOOR OPENERS AT ACTON HOUSE

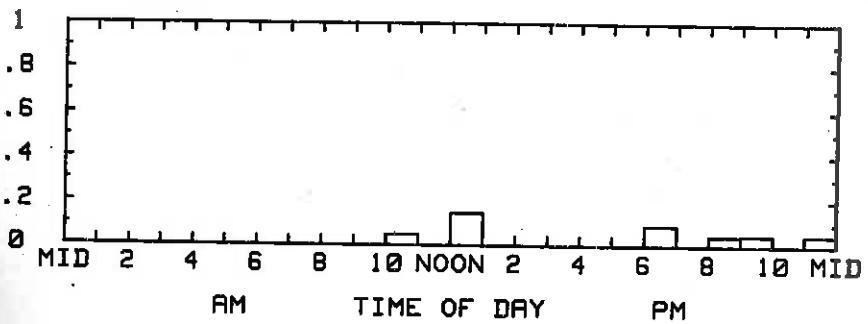
MID 2 4 6 8 10 NOON 2 4 6 8 10 MID

ON/OFF CYCLES

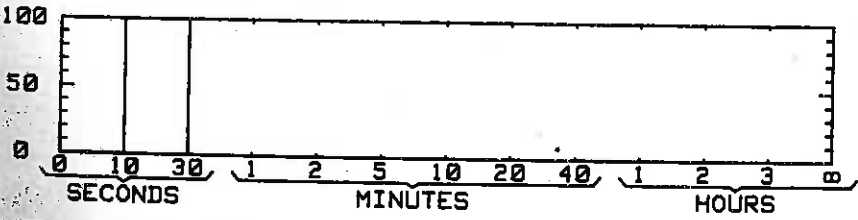
TUE
WED
THU
FRI
SAT
SUN
MON
TUE



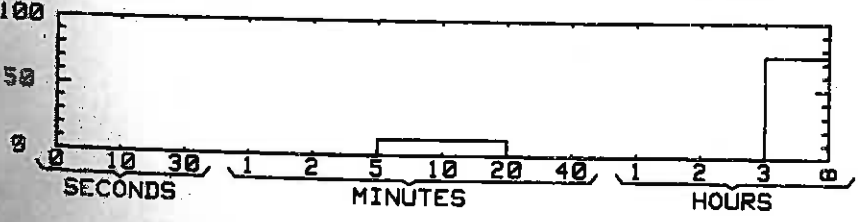
TIME ON (%)



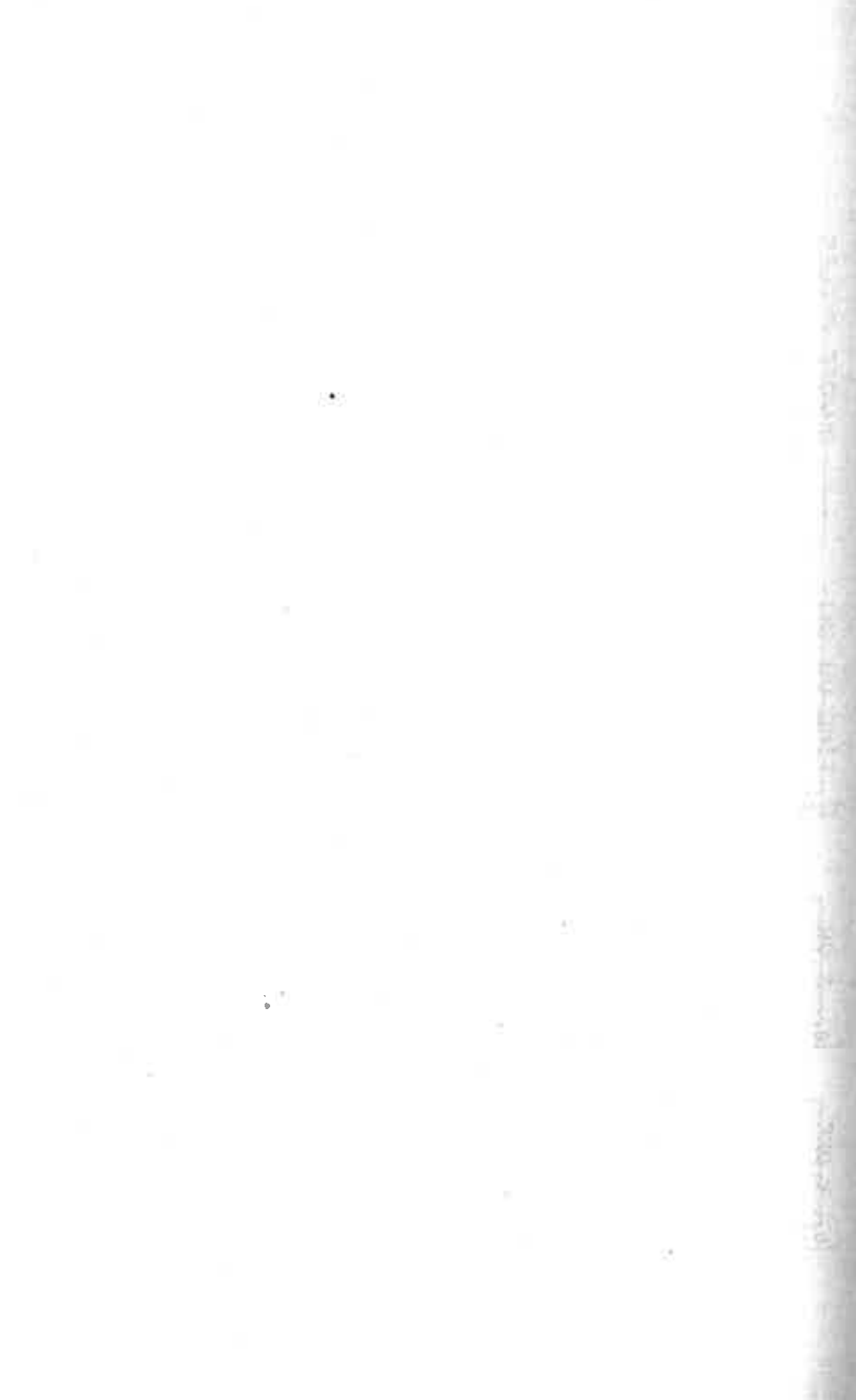
% ON



% OFF



DURATION DISTRIBUTIONS



REFERENCES

- [1] George W.Hart, Nonintrusive Appliance Load Data Acquisition Method, MIT Energy Laboratory Technical Report, September 1984.
- [2] Michael R. Anderberg, Cluster Analysis for Applications, Academic Press, 1973.
- [3] J.A. Hartigan, Clustering Algorithms, John Wiley and Sons, 1975.
- [4] George W. Hart, The Digital AC Monitor, MIT Energy Laboratory Technical Report, August 1985.