# Viterbi Algorithm with Sparse Transitions (VAST) for Nonintrusive Load Monitoring

Michael Zeifman and Kurt Roth
Building Energy Efficiency Group
Fraunhofer Center for Sustainable Energy Systems
Cambridge, Massachusetts 02141
{mzeifman, kroth}@fraunhofer.org

***Abstract*—Implementation of smart grid provides an opportunity for concurrent implementation of nonintrusive appliance load monitoring (NIALM), which disaggregates the total household electricity data into data on individual appliances. This paper introduces a new disaggregation algorithm for NIALM based on a modified Viterbi algorithm. This modification takes advantage of the sparsity of transitions between appliances' states to decompose the main algorithm, thus making the algorithm complexity linearly proportional to the number of appliances. By consideration of a series of data and integrating *a priori* information, such as the frequency of use and time on/time off statistics, the algorithm dramatically improves NIALM accuracy as compared to the accuracy of established NIALM algorithms.**

***Keywords-electric appliance; energy efficiency; statistical distribution; clustering***

## I. Introduction

Buildings consume almost 40% of America's energy and 73% of its electricity [1]. Prior studies indicate that building electricity consumption can be reduced by up to 10-15% using better energy management [2], which may benefit from appliance-specific data. One way to obtain the appliance-specific information is by disaggregation of total electricity consumption data acquired at the main breaker level [3]. Such nonintrusive appliance load monitoring (NIALM) method requires both hardware (data acquisition) and software (signal processing algorithms) components. Smart grid implementation can greatly facilitate the data acquisition process, yet the accuracy of available algorithms does not reach the level sufficient for massive implementation of NIALM [4].

For the data sampling rates attainable with smart grids (up to 1 Hz), the established NIALM algorithms are based on matching the observed step-wise power changes with appliances being turned on or off [3], [5]. This matching is prone to both measurement and algorithmic errors and to the ambiguity related to a simultaneous start or end of multiple appliances. The main reason, nonetheless, for the monitoring accuracy being low is an overlap in the power draw between different appliances [4].

In this paper, we propose a new NIALM algorithm capable of dramatic improvement of disaggregation accuracy. In the proposed algorithm, the matching between the measurable power observations and the appliances is made by consideration of a series of transitions among the appliance states which maximizes the likelihood of not only the given series of power changes but also the time-of-use statistics. The appliances' states are recovered using a modification of Viterbi algorithm. In this modification, the advantage of the sparsity of transitions between appliances' states is used to decompose the main algorithm, thus reducing the number of states and transitions by orders of magnitude and reaching feasible computational efficiency.

## II. Established NIALM Algorithms

A comprehensive review of NIALM algorithms is given in Ref. [4]. Here, we briefly discuss the established NIALM methods, appropriate for data sampled at a rate of about 1 Hz.

### A. *Conventional NIALM Algorithms with one-by-one matching*

The established NIALM algorithm [3], [5] includes five steps. First, an edge detector identifies changes in steady-state power-draw levels. Second, a cluster analysis algorithm locates these changes in a two-dimensional signature space of real and reactive power. Third, positive and negative clusters of similar magnitude are paired or matched. In the fourth step, known as anomaly resolution, unmatched clusters and events are paired or associated with existing or new clusters according to a best likelihood algorithm. In the final step, pairs of clusters are associated with known power draw levels for different loads to determine their operating schedule. This disaggregation method can be considered as an optimization algorithm.

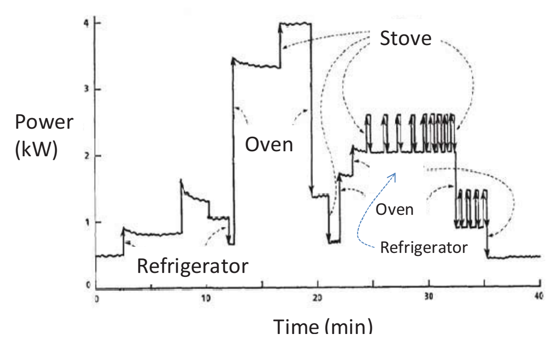


Fig. 1 Power vs. time and corresponding appliances, from Ref. [3]

Fig. 1 illustrates step 1 of this method. For the appliances considered, the 2D clusters are quite distinct; however, there is no guarantee that the clusters do not overlap in a general case. Also, it is unlikely that power-grid meters will provide both

978-1-4244-9894-9/11/$26.00 ©2011 IEEE

real and reactive power components. In case of only real power component available, the separation of appliances is significantly worse [6].

### B. NIALM algorithms with simultaneous matching of observation series and numerous appliance turns

In order to improve algorithmic performance, Baranski and Voss have proposed a method that uses an optimization algorithm to optimally match a large set of detected real power changes to appliance presences in time. The use of the simultaneous matching apparently improves the detection and monitoring accuracy of the method so that its performance is comparable to that of the 2D NIALM algorithm [3], [5]. However, the genetic algorithm proposed as the optimization algorithm is a simulation method, which may or may not provide the optimal solution.

Another optimization algorithm was offered in [7], where integer programming was used to match the presence of appliances in time to the observations. This technique, however, requires signal waveforms as input and accordingly will not work with mere changes of power.

Additional information about the appliances, e.g., the statistics of appliance usage in time, could potentially improve the algorithmic performance. In principle, the optimization algorithms can be complemented with the distributions of time on or time off for the appliances in question. The calculation of time on or off of a given appliance requires consideration of the appliance states prior to the given state, which poses a challenge of exponential complexity with regard to the number of appliances. In other words, optimization algorithms with time of use statistics become computationally intractable for a realistic number of household appliances.

## III. Proposed Algorithm

### A. Algorithm Assumptions

A set of household appliances with binary states "on" and "off" can be considered as a system with various states. Whenever an appliance changes its state, the system transits to a different state, too. If the current system state depends only on the previous state, the system is completely characterized by an initial state and a set of transition probabilities. The available measurable information on the system, e.g., the observed power changes, the distribution of power changes for a given appliance and the time on/ off statistics for a given appliance, can be incorporated into the transition probabilities. In this case, the system states can be optimally decoded using the Viterbi algorithm [8].

The Viterbi algorithm obtains the maximum likelihood estimation $\{\hat{s}_t\}$ of the state sequence $\{s_t\}$, given the transition observations $\{\omega_t\}$, where $t$ is the series of discrete transition times [9]:

$$\{\hat{s}_t\} = \underset{\{s(t)\}}{\arg\max}[\{s_t\} \mid \{\omega_t\}] \tag{1}$$

The algorithm calculates the maximum probability of observing a state after each transition step. In the last step, the state with maximum probability is selected and traced back to recover the most likely transition sequence [10].

However, for $N$ household appliances, the system will have $2^N$ states, and the number of possible transitions will be $2^{2N}$. The sheer number of states and state transitions for a typical number of household appliances of about 30 [3] makes this system not amenable to a computational solution. Fortunately, under simple yet realistic assumptions, the entire system and the Viterbi algorithm can be decomposed in such a way that the total number of transitions (and algorithm complexity) is *linearly* proportional to $N$.

Let the appliances be ordered by their power draw. Because of the measurement errors and natural variability, there is scatter in changes of power when a given appliance is turned on or off. The dependence of the probability of appliance $i$ to be turned on or off on the power change $\Delta P$ can be expressed through the probability density function (PDF), conditional for a given appliance. For a given $\Delta P$, the probability that appliance $i$ has been turned on is proportional to PDF($\Delta P|i$).

The first required assumption is that an overlap between the PDFs of appliance $i$ and appliance $(i + 2)$, $i = 1, 2, .., N - 2$ is negligible (see Fig. 2). This assumption is central for further development and requires justification. For the established NIALM algorithms considered earlier, the level of accuracy is about 80% [4]. The best available NIALM algorithms that work with waveform data sampled at 20-500 kHz can attain the level of accuracy of up to 90% [4]. Therefore, for the 30 household appliances, the probability overlap between appliance $i$ and appliance $(i + 2)$ can be considered to be negligible if it is less than 0.1/30 = 0.0033.

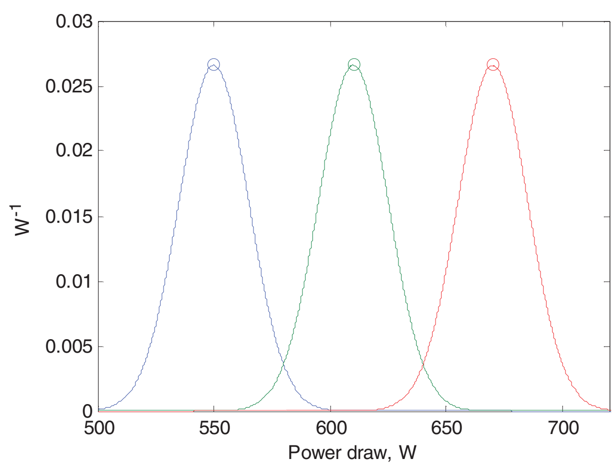


Fig. 2 PDFs of three neighboring by power draw appliances. Distribution means (circles) are located at a distance of 4 standard deviations apart.

As it was mentioned earlier, the established NIALM algorithms use cluster analysis to, at least preliminary, relate the power changes with the presence of appliances in time. For a 1D cluster analysis, it can be reasonably assumed that the distance between two adjacent cluster centers is at least four standard deviations [11]. The distance between clusters $i$ and $(i + 2)$ will be eight standard deviations, and, by Chebyshev's theorem, the probability overlap between these two clusters will be no more than 1/16. Chebyshev's theorem, however, is a very general estimation which is applicable to any distribution function with finite variance. A realistic distribution model of a

Gaussian mixture, for example, yields a probability overlap of about 0.00003, which is two orders of magnitude less than the threshold of 0.0033. Even if the distance between two adjacent cluster centers is three standard deviations, the probability overlap is about 0.0013, which is still less than the threshold.

In light of the above assumption, the overlap between the PDF-s of appliance $i$ and appliance $(i + k)$, $k > 2$ is also negligible. Therefore, for a given value of a measured power change $\Delta P$, only the states of the appliance with the distribution mean closest to $\Delta P$ and of its two neighbors shall be considered. In this case, the transition probability matrix is sparse with zeros for any transition but for those between the states of the above-mentioned three appliances. If it can also be assumed that the appliances are mutually independent, then the only three mentioned appliances need to be included in the system at a time.

*B. Algorithm Realization*

A straightforward realization of the algorithm on the basis of the above assumptions could be done as follows. Consider clusters of, e.g., positive power changes, cluster $i$ being corresponding to household appliance $i$. Since appliance $i$ can only be confused with appliance $(i - 1)$ and $(i + 1)$, these three appliances need to be considered in order to reconstruct the states of appliance $i$ in time. Correspondingly, the measured power changes within the boundaries of clusters $(i - 1)$, $i$, and $(i + 1)$, need to be considered as input. In order to reconstruct the states of appliance $(i + 1)$, appliances $i$ and $(i + 2)$ need to be considered, and so on. In this consideration, it is assumed that two or more appliances do not start or stop simultaneously. However, this assumption is not essential (see Section III.D).

It is easy to see that in this algorithm, almost each appliance is considered three times, and generally three different solutions for each appliance can be obtained. Also, the transition probability matrix size is 8·8, which may slow down the computations.

An alternative and more elegant algorithm realization works with appliance pairs. Under the first assumption, there is no overlap between the power draws of appliances $(i - 1)$ and $(i + 1)$. Therefore, it is possible to further decompose the triple $(i - 1)$, $i$, $(i + 1)$ into two pairs, $(i - 1)$, $i$ and $i$, $(i + 1)$. Within each pair, the transition probability matrix is 4·4 in size, which greatly simplifies the computations.

In order to consider all appliances, this algorithm needs to be also applied to pair $(i + 1)$, $(i + 2)$, to pair $(i + 2)$, $(i + 3)$, .., so that every appliance, but 1st and $N$th, will be considered twice. Because of this duality, two different solutions can be obtained for the same appliance. The proposed resolution of this conflict is considered in Section III.D. In this Section, we consider the basic two-appliance system in detail.

Table 1 lists the states of a two-appliance system, and Tables 2 (positive change ΔP) and 3 (negative change -ΔP) list the transition probabilities among these states. In Tables 2-3, $w_i$ is a fraction of number of switches of appliance $i$ among the overall number of appliance switches over a prolonged period of time, i.e., the probability of appliance $i$ being turned on/off, $P_{i(-)}$ and $P_{i(+)}$ are PDFs of the negative and positive power changes of appliance $i$, and $C_{\mathrm{on},i}$ and $C_{\mathrm{off},i}$ are cumulative distribution functions (CDFs) of the duration of time on and time off of appliance $i$.

TABLE 1 STATES OF TWO-APPLIANCE SYSTEM

| *State* | *Appliance i* | *Appliance i + 1* |
|---|---|---|
| 1 | off | off |
| 2 | on | off |
| 3 | off | on |
| 4 | on | on |

Tables 2 and 3 suggest that the proposed algorithm differs from the established algorithms (see Section II) not only by consideration of a series of observations in time but also by consideration of the turn-on/off probabilities, distributions of the power changes and distributions of the times on and off. In this way, the proposed algorithm utilizes prior data on the appliances in question. Prior data on additional features, e.g., the distributions of amplitude and duration of power surges, characteristic of many appliances, can be naturally included in the transition probabilities listed. Moreover, these prior data can be further enhanced by consideration of the probabilities/distributions separately for, e.g., day and night hours, or summer and winter seasons.

TABLE 2 TRANSITION PROBABILITIES FOR A POSITIVE CHANGE OF POWER $\Delta P$. ONLY NONZERO PROBABILITIES ARE LISTED.

| *Transition between states* | *Probability* |
|---|---|
| 1 ⟶ 2 | $\propto w_i P_{i(+)}(\Delta P) C_{off,i}(t_{12})[1 - C_{off,i+1}(t_{13})]$ |
| 1 ⟶ 3 | $\propto w_{i+1} P_{i+1(+)}(\Delta P) C_{off,i+1}(t_{13})[1 - C_{off,i}(t_{12})]$ |
| 2 ⟶ 4 | 1 |
| 3 ⟶ 4 | 1 |

TABLE 3 TRANSITION PROBABILITIES FOR A NEGATIVE CHANGE OF POWER $-\Delta P$. ONLY NONZERO PROBABILITIES ARE LISTED.

| *Transition between states* | *Probability* |
|---|---|
| 2 ⟶ 1 | 1 |
| 3 ⟶ 1 | 1 |
| 4 ⟶ 2 | $\propto w_{i+1} P_{i(-)}(-\Delta P) C_{on,i+1}(t_{42})[1 - C_{on,i}(t_{43})]$ |
| 4 ⟶ 3 | $\propto w_i P_{i(-)}(-\Delta P) C_{on,i}(t_{43})[1 - C_{on,i+1}(t_{42})]$ |

Tables 2 and 3 also suggest that there are two peculiarities in the considered system that may hinder the implementation of the Viterbi-type algorithm. First, several transitions between the states are forbidden, which may render the algorithm unsolvable. Second, the transition probabilities of this system are time-dependent, which calls for additional calculations of the time intervals $t_{12}$, $t_{13}$, $t_{42}$ and $t_{43}$. The time-dependent probabilities also make the current system state to be dependent on several previous states and not on just one previous state.

The algorithm will be unsolvable, e.g., in case of a missing power change or in case of a wrong power change, which in turn can be, e.g., the result of a simultaneous start/end of two or more appliances or a measurement/ processing error. The missing power change can result, e.g., from a ramp at appliance start so that the power change gets split. In case of such insolvency, the algorithm can be programmed in such a way as to yield a special state, e.g., 0, each time the insolvency occurs. After the algorithm has processed the data, these special state occurrences can be found and the corresponding power changes can be separated and excluded for further consideration. The algorithm then shall be reapplied to the remaining data. The above-described procedure can be repeated several times until the number of the insolvencies gets below a pre-defined threshold. See also Section III.D for additional remedies.

## IV. ILLUSTRATIVE EXAMPLE

The main novelty of the proposed algorithm is the division of the appliances into overlapping pairs. Therefore, in order to compare the performance of the proposed algorithm to the performance of the established algorithm, simulated data on two appliances with a heavy overlap between the power draw distributions have been generated. The simulation data include 300 positive and 300 negative changes of power accumulated over 10 hours of sampling with 1 Hz frequency. The distributions of the power draw are Gaussian with mean μ and standard deviation σ. The distributions of time-on durations are uniform with boundaries $t_{start}$ and $t_{stop}$. The time-off distributions are also uniform, complementing the time-on distributions and the frequency of the power changes. The characteristics of the simulation data are listed in Table 4 and the PDFs of the power changes are shown in Fig. 3. Note that the distributions of positive and negative power changes of each appliance do not match exactly. This is often the case for real appliances.

The degree of the overlap in this simulated data set significantly exceeds that underlying the first assumption about household appliances (see Section III.A). Therefore, the algorithms will experience more severe conditions with this data set than under normal circumstances. It is believed here that if one algorithm exhibits a significant performance advantage over the other on this data set, this performance gap will also hold for any expected data, too.

TABLE 4 CHARACTERISTICS OF SIMULATED DATA

| ***Parameters*** | ***Appliance 1*** | ***Appliance 2*** |
|---|---|---|
| $w_i$ | 2/3 | 1/3 |
| μ, σ for +ΔP | 130, 15 | 150, 20 |
| μ, σ for -ΔP | -135, 15 | -160, 20 |
| [tstart tstop] for time on | [30 80] | [60 100] |

Fig. 4 shows a fragment of the data generated along with the indication of appliance states. The indicator takes value 1 if appliance 1 is turned on, value 2 if appliance 2 is turned on, value -1 if appliance 1 is turned off, and value -2 if appliance 2 is turned off. It can be seen in the Figure that the heavy overlap between the power draws of the two appliances makes the reconstruction of appliance states a challenging problem.

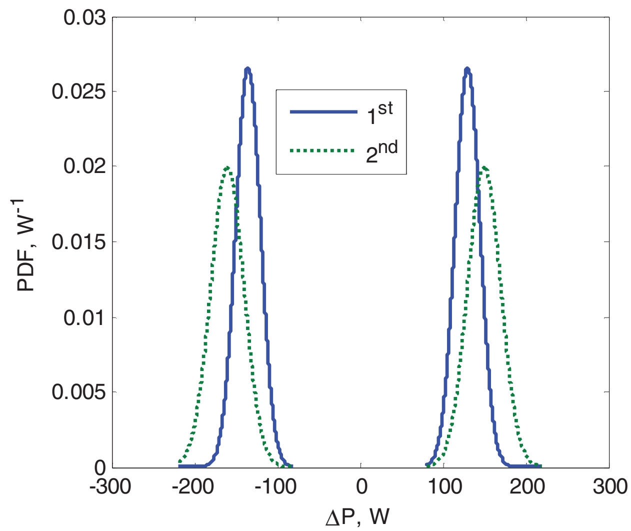


Fig. 3 PDFs of change of power used in simulation of two appliances.

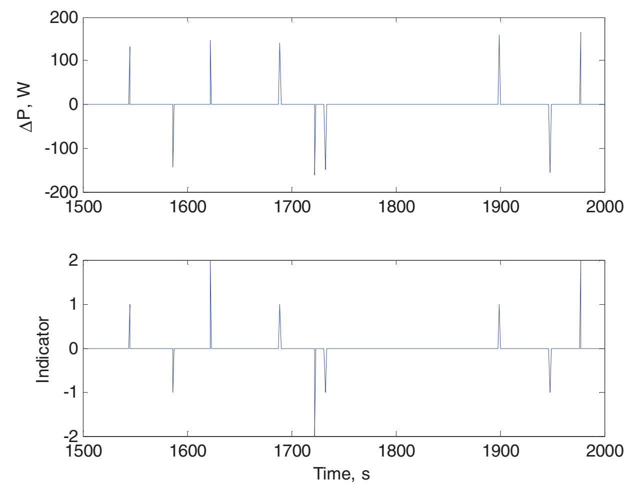


Fig. 4 Fragment of simulated data. Total data set includes 600 power changes over 10 hours.

First, the established NIALM algorithm was applied to the simulated data. Portion of the results that correspond to the data fragment of Fig. 4 are shown in Fig. 5. The indicator takes value 0 whenever no matching between the positive and negative power changes of the same appliance can be found. It is seen in the Figure that the established algorithm performs poorly. The overall fraction of correctly reconstructed appliance states is 54.5%.

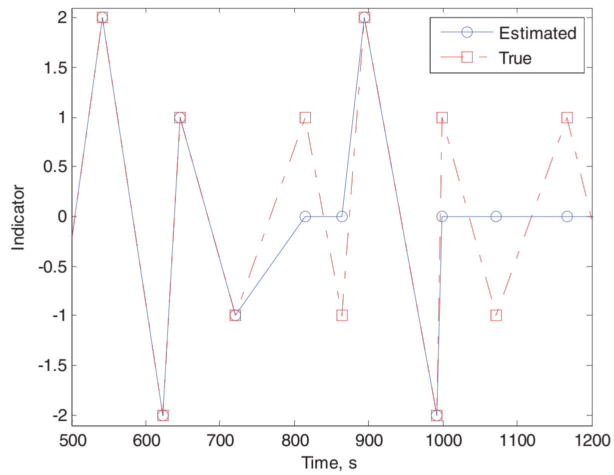


Fig. 5 Fragment of results of established NIALM algorithm, applied to simulation data. Overall correct reconstruction is 54.5%.

It is interesting to compare this success metric to the theoretical performance result of a Bayesian classifier [11] applied to the clusters of power draw with PDFs listed in Table 4. If equal cluster probabilities are used, which is the case for the established NIALM algorithm, the Bayesian classifier uses the decision boundaries calculated by the intersections of the power draw PDFs. In this case, for the positive power changes, the decision boundary is 142.64 W. For the negative power changes, the decision boundary is -149.04 W. If a positive power change is observed, it is classified as appliance 1 being turned on if $\Delta P$ < 142.64 and as appliance 2 being turned on otherwise. If a negative power change is observed, it is classified as appliance 1 being turned off if $\Delta P$ > -149.04 and as appliance 2 being turned off otherwise. The error probability can then easily be calculated by integrating of the corresponding PDFs. In this particular example, the estimated error probability is 0.5344, which is very close to the error rate of the established algorithm.

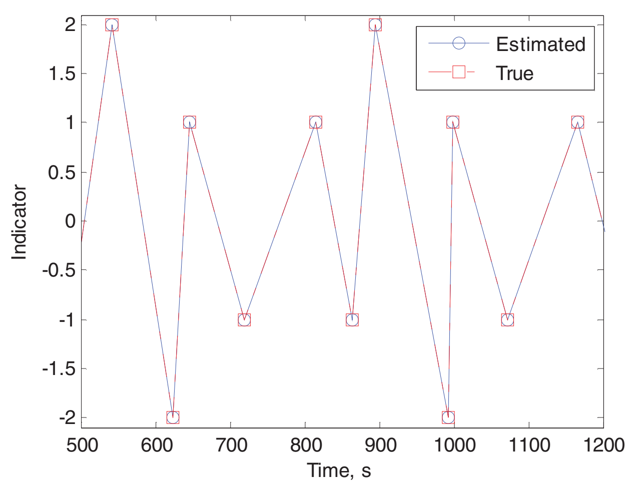


Fig. 6 Fragment of results of proposed algorithm, applied to simulation data. Overall correct reconstruction is 97.3%.

In order to apply the proposed algorithm to the simulated data, the empirical CDFs of the time on/time off durations for each of the appliance have been estimated. The PDF-s of the power draw and the probabilities of appliance switches were used in accordance with Table 4. The algorithm was coded in MATLAB 7. The results of this Viterbi-type algorithm, applied to the simulated data, are shown in Fig. 6. This particular portion of results indicates a perfect restoration of the appliance states. The overall fraction of correctly reconstructed appliance states for the proposed algorithm is 97.3%. This is a dramatic improvement over the conventional method. The overall computational time required to process all the data (36,000 data points of which 600 are then used in the algorithm) on a 3.0 GHz Pentium machine is about 2 s, which is acceptable for a MATLAB program.

In the proposed algorithm, a significant portion of computational time is being spent on calculations of the time periods required for the transitional probabilities (see Tables 2 and 3). Therefore, it is of interest to verify that the statistics of time on/off indeed improve the algorithm accuracy. To this end, a simplified version of the algorithm with no time statistics included, was developed and coded. The results of the application of this simplified algorithm to the simulated data are shown in Fig. 7.

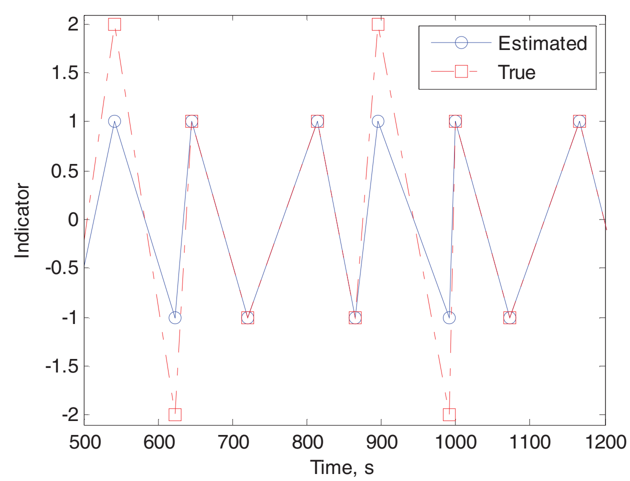


Fig. 7 Fragment of results of simplified algorithm (no time statistics), applied to simulation data. Overall correct reconstruction is 65.3%.

The results of this algorithm are somewhat better than the results of the established algorithm shown in Fig. 5. The overall fraction of correctly reconstructed appliance states for the simplified algorithm is 65.3%. However, this improvement is minimal as compared to the dramatic improvement obtained with the time statistics (97.3%).

## V. Real-Life Effects

### A. Influence of Missing Data

The simulated data may not realistically simulate real data, even though the degree of the overlap used is actually larger than that underlying the first assumption (see Section III.A). In reality, some of the data may be missing. If, e.g., a negative power change is missing, then the algorithm will not be able to properly match the previously obtained positive power change, and this error may accumulate with time.

In order to test how missing data affect the algorithm performance, two new data sets were generated from the original simulation data. In the first data set, 1% of the power changes have been removed at random. In the second data set, 5% have been removed. The results are shown in Figs. 8-9.

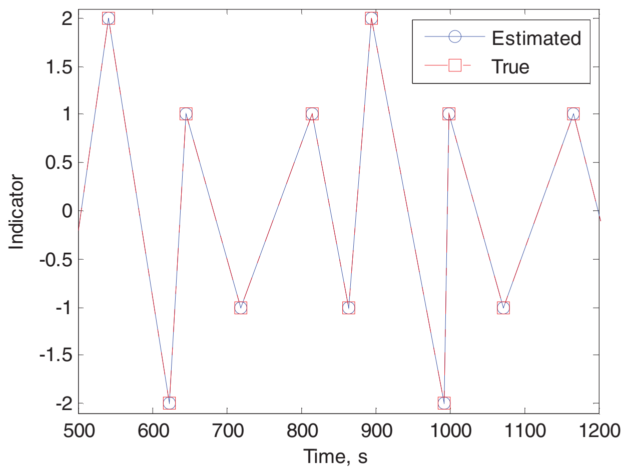


Fig. 8 Fragment of results of proposed algorithm, applied to simulation data with missing 1% of power changes. Overall correct reconstruction is 90.0%.

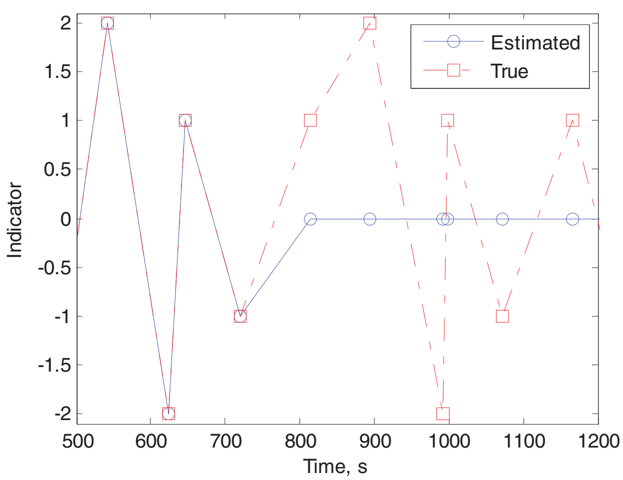


Fig. 9 Fragment of results of proposed algorithm, applied to simulation data with missing 5% of power changes. Overall correct reconstruction is 67.0%.

It is seen in Figs. 8-9 that, while 1% of missing data has small effect on the algorithmic performance, 5% of missing power changes results in a dramatic deterioration of algorithm accuracy. The missing points apparently render the algorithm unsolvable for portions of data.

At least two different strategies can be used to work with the corrupted data. The simplest solution is to identify the starting points of the prolonged insolvency periods, i.e., the periods with indicator value being zero. Then the data points corresponding to the starting points of the prolonged periods are excluded from consideration, and the algorithm is applied to the remaining data. The process is then repeated until the maximum length of the prolonged periods of insolvency reached a desired low level.

Fig. 10 shows the results of the application of this simple strategy to the data set with 5% missing points. The accuracy of the modified "robust" algorithm has dramatically increased to 81%. This value is intermediate between the 97.3% accuracy of the proposed algorithm and the 54.5% accuracy of the established algorithm, as applied to the uncorrupted data. Note that the accuracy of the established algorithm, applied to the data set with 5% missing points, decreases to 50%.

Another strategy to deal with the missing points can be consideration of non-zero probability of the system to remain in the same state. That is, if the probabilities of transitions 1→1, 2→2, 3→3, 4→4 (see Tables 2, 3) are non-zero, then a missing data point will no longer cause the system to remain in the previous state. Since this strategy can also be used for conflict resolution, it is considered in more detail in Section III.D.

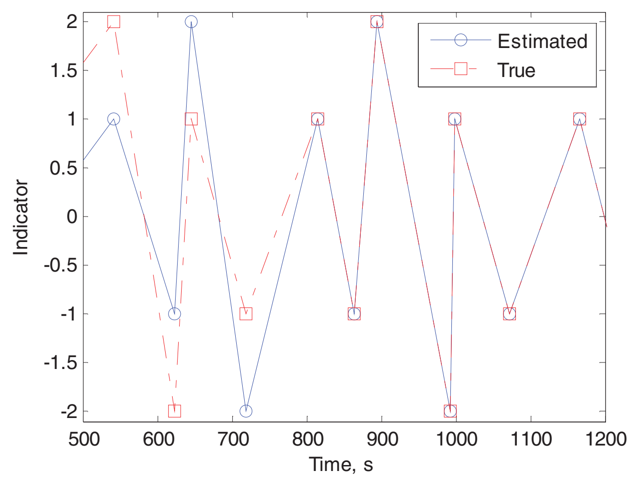


Fig. 10 Fragment of results of proposed algorithm, applied to simulation data with missing 5% of power changes. Overall correct reconstruction is 81.0%.

### B. *Conflict Resolution*

The proposed algorithm considers pairs of appliances, adjacent in power draw. Therefore, each appliance, but first and last, is considered two times. Because of this duality, two different solutions can be obtained for the same appliance. Suppose the two solutions for appliance $i$ coincide over $k$ consecutive data points, but differ for the $k + 1$ data point. This occurs, e.g., whenever a power change $\Delta P$ observed at the $(k + 1)^{\text{th}}$ data point is caused either by appliance $(i - 1)$ when a pair $i$, $(i + 1)$ is considered, or by appliance $(i + 1)$ when a pair $(i - 1)$, $i$ is considered. Since the transition probabilities listed in Tables 2 and 3 do not permit remaining at the same state, the algorithm is enforced to accept a wrong transition.

One way to resolve this problem is to consider a non-zero probability of a system of two appliances to remain in the same state. This probability shall be proportional to the probability of an appliance, external to the two considered appliances, to be turned on or off with the observed power change $\Delta P$ or larger. Therefore, the transition probability matrix shall be modified to account for the "external" transitions.

The transition probability matrix with the possibility of such external transitions is listed in Table 5 for negative changes of power. A matrix for the positive power changes with the external transitions can be obtained similarly.

The considered possibility of the power change being caused by an appliance external yet adjacent to the considered couple of appliances, is not the only possibility for the conflict. Other possibilities could be a measurement error, a simultaneous turning on or off of two appliances, and a ramp at appliance start so that the power change gets split. A proposed solution to this problem is to run the algorithm in two stages. At the first stage, the conflicting points are excluded from consideration, and the algorithm is run for all pairs of appliances. At the second stage, the power changes corresponding to the excluded states are considered. Various strategies can be applied to processing these power changes.

TABLE 5 TRANSITION PROBABILITIES FOR A NEGATIVE CHANGE OF POWER -$\Delta P$ WITH POSSIBILITY OF EXTERNAL TRANSITIONS. ONLY NONZERO PROBABILITIES ARE LISTED.

| ***Transition between states*** | ***Probability*** |
|---|---|
| $1 \rightarrow 1$ | $\propto [w_{i-1}P_{i-1(-)}(-\Delta P) + w_{i+2}P_{i+2(-)}(-\Delta P)]$ |
| $2 \rightarrow 1$ | $\propto w_{i+1}P_{i(-)}(-\Delta P)C_{on,i}(t_{21})[1 - C_{off,i+1}(t_{24})]$ |
| $2 \rightarrow 2$ | $\propto [w_{i-1}P_{i-1(-)}(-\Delta P) + w_{i+2}P_{i+2(-)}(-\Delta P)]$ |
| $3 \rightarrow 1$ | $\propto [w_{i+1}P_{i(-)}(-\Delta P)[1 - C_{off,i}(t_{34})]C_{on,i+1}(t_{31})]$ |
| $3 \rightarrow 3$ | $\propto [w_{i-1}P_{i-1(-)}(-\Delta P) + w_{i+2}P_{i+2(-)}(-\Delta P)]$ |
| $4 \rightarrow 2$ | $\propto w_{i+1}P_{i(-)}(-\Delta P)C_{on,i+1}(t_{42})[1 - C_{on,i}(t_{43})]$ |
| $4 \rightarrow 3$ | $\propto w_{i}P_{i(-)}(-\Delta P)C_{on,i}(t_{43})[1 - C_{on,i+1}(t_{42})]$ |
| $4 \rightarrow 4$ | $\propto [w_{i-1}P_{i-1(-)}(-\Delta P)] + w_{i+2}P_{i+2(-)}(-\Delta P)]$ |

For example, if several excluded power changes occur within a given time interval, they can be merged together. Isolated in time power changes can be split: $\Delta P = \Delta P_1 \pm \Delta P_2$, where $\Delta P_1$ and $\Delta P_2$ are within the boundaries of any two clusters. The algorithm is then reapplied to the power change data modified in this way.

## VI. DETAILED IMPLEMENTATION

The algorithm starts with analyzing and modeling of historical data, i.e., the data collected by a sensor over, e.g., a two-week period. Stepwise negative changes of power are firstly identified. These negative changes usually correspond to an appliance being turned off. Because of the measurement errors and natural variability, there is scatter in negative changes of power when a given appliance is turned off. Therefore, in order to identify appliances, the identified negative changes of power are grouped into clusters, using, e.g., the ISODATA algorithm [11]. Additional information, e.g., the time of use statistics, can be used to optimally merge and/or split clusters [12]. Each cluster may correspond to a separate appliance being turned off.

Similarly, positive changes of power are identified. However, since some of the appliances exhibit surge at start, the post-surge value is used for the positive power change. The surge features in terms of its magnitude and duration can be statistically characterized and used to complement the transition probabilities listed in Table 2 [12].

In each cluster, the negative power changes are characterized statistically in parametric form, e.g., by fitting their empirical distribution to a Gaussian mixture model. Our experience suggests that a two-component Gaussian mixture model of the probability density function (PDF) is sufficient in most cases, however, other parametric distribution models can be used.

Preliminary time statistics of appliances being at state "on" and "off" are then estimated. To this end, each identified negative power change $j$ of magnitude -$\Delta P_{ij}$ that occurred at time $t_j$ and came from cluster $i$, $i$ = 1, 2., ., $N$, is matched with a positive power change $k$ of magnitude $\Delta P_{ik}$ that occurred earlier, at time $t_k$. An exact equality between the positive and negative power changes for a match is not required, instead, a tolerance $\delta$ is used:

$$\Delta P_{ik} \in \Delta P_{ij}(1 \pm \delta). \tag{2}$$

At this stage, the match is considered to be the first $\Delta P_{ik}$ satisfying Eq. (2) when going backward from the -$\Delta P_{ij}$ in time. In this way, for each cluster $i$, a sample of times "on" $\{t_{\text{on}}\}_i$ is constructed by calculating $t_{\text{on}} = t_j - t_k$ for each available matching pair. Similarly, a sample of times "off" $\{t_{\text{off}}\}_i$ is constructed by calculating $t_{\text{off}} = t_{k+1} - t_j$. Once both samples are available for each cluster, the cumulative distribution functions (CDF) of $t_{\text{on}}$ and $t_{\text{off}}$ for each cluster are calculated empirically. Other time-dependent statistics, e.g., the clock-time probability of use, can also be implemented for cluster characterization.

A collection of the positive power changes that match the negative power changes from cluster $i$ is considered to be cluster $i_{\text{plus}}$. In each such cluster, the statistical distribution of the positive power changes is parametrically characterized, e.g., by fitting the empirical distribution to a two-component Gaussian mixture.

Consider adjacent clusters of negative power changes $i$ and $i$ + 1. Each of these clusters includes the detected negative power changes that range from -$\Delta P_{i\_max}$ to -$\Delta P_{i\_min}$ and from -$\Delta P_{i+1\_max}$ to -$\Delta P_{i+1\_min}$ for clusters $i$ and $i$ + 1 correspondingly. Consider the detected positive power changes. The positive power changes $\Delta P_k$ are considered to be candidates for matching with the clusters $i$ or $i$ + 1 if they are within the matching boundaries plus the tolerance:

$$\Delta P_{i+1,min}(1 - \delta) \leq \Delta P_k \leq \Delta P_{i,max}(1 + \delta). \tag{3}$$

Clusters $i$ and $i_{\text{plus}}$ presumably correspond to appliance $i$, whereas clusters $i$ + 1 and $(i + 1)_{\text{plus}}$ correspond to appliance $i$ + 1. The main algorithm (Sections III B-D) is then applied to the consecutive pairs of clusters. After the algorithm solution is obtained, the cluster and time on/off statistics are updated.

The real-time version of the algorithm is similar to the version for the historical data. Instead of consideration of a lengthy time period, a data window of a reasonable size, e.g., the recent 24 hours is used. This size, the so-called truncation depth [8] depends on the algorithm convergence and can be estimated experimentally. The algorithm is applied each time a new power change is detected. Since the algorithm is intended to resolve the likeliest state path, i.e., the most probable series of appliances' states, the appliance state estimated at a given time will be re-estimated as soon as new data are obtained and processed, and therefore may change.

This comprehensive algorithm is currently being tested, and the results will be reported elsewhere.

## VII. Discussion

To the best of our knowledge, this is the first attempt to apply a Viterbi algorithm for a plurality of two-state appliances. Attempts to use a Viterbi algorithm in NIALM applications were made in early 1990-s [3], and early 2000-s [6]. However, these works considered appliances with several definitive states, e.g., a washing machine, and never attempted to model several two-state appliances.

The proposed Viterbi-type algorithm naturally integrates various data on appliances, such as the frequency of use, time on/time off statistics, surge magnitude and duration. Moreover, the dependence of these characteristics on time of the day, day of the week, and on the season can easily be included, too. The ease of inclusion of such *a priori* data into the existing NIALM algorithms is questionable. The resultant dramatic accuracy improvement of the proposed algorithm over the established NIALM algorithm has been demonstrated.

The main assumption underlying the sparsity of the transition probability matrix and the corresponding pairwise decomposition is quite realistic. However, even if this assumption is violated, i.e., significant overlap occurs between non-adjacent by power draw appliances, the decomposition can be based on processing of appliances by triples or quadruples. Even though the transition probability matrix will be 9·9 or 16·16 in size, the algorithm complexity will be proportional to the matrix size and linearly proportional to the number of appliances. A brute force application of the Viterbi algorithm to a typical set of household appliances would require about $2^{30} \cdot 2^{30}$ transition probability matrix which would render it computationally unsolvable.

## References


[1] DOE Internet site: http://buildingsdatabook.eere.energy.gov/

[2] Ehrhardt-Martinez, K., Donnelly, K.A., and Laitner, J.A., "Advanced Metering Initiatives and Residential Feedback Programs: a Meta-Review for Household Electricity-Saving Opportunities," Report E105, ACEEE 2010.

[3] Hart, G.W. , "Nonintrusive Appliance Load Monitoring," IEEE proceedings, 80, p. 1870 (1992).

[4] Zeifman, M., Roth, K., "Nonintrusive Appliance Load Monitoring: Review and Outlook," IEEE Transactions on Consumer Electronics, vol. 57, in press (2011).

[5] Drenker, S., Kader, A.. "Nonintrusive Monitoring of Electric Loads," IEEE Computer Applications in Power, 47 (1999).

[6] Baranski, M., J. Voss. "Genetic Algorithm for Pattern Detection in NIALM Systems," IEEE Conference on Systems, Man and Cybernetics, p. 3462 (2004).

[7] Suzuki, K., et al. "Nonintrusive Appliance Load Monitoring Based on Integer Programming," SICE Annual Conference, (2008), p. 2742.

[8] Forney, G. (1973), "The Viterbi Algorithm," Proceedings of the IEEE , 268-278.

[9] Hocaoglu, F.O., Gerek, O.N., Kurban, M., "A novel wind speed modeling approach using atmospheric pressure observations and hidden Markov models," J. Wind Eng. Ind. Aerodyn. 98, p. 472 (2010).

[10] Rabiner, L.R., Juang,B.H., "An introduction to hidden Markov models," IEEE ASSP Magazine 3, p. 4 (1986)

[11] Tou, J. T., Gonzalez, R. C. (1974). Pattern Recognition Principles. New York: Addison-Wesley.

[12] Zeifman, M., Roth, K., "Suite of algorithms for nonintrusive load monitoring," Provisional patent application, USPTO (2010).