Title
Distributed Control of Plug Loads for Building Energy Management

Author
Giroud, Mathieu Lee

Publication Date
2022

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA SAN DIEGO

Distributed Control of Plug Loads for Building Energy Management

A Thesis submitted in partial satisfaction of the
requirements for the degree Master of Science

in

Engineering Sciences (Mechanical Engineering)

by

Mathieu Lee Giroud

Committee in charge:

Professor Jan Kleissl, Chair
Professor Patricia Hidalgo-Gonzalez
Professor Sonia Martinez

2022

The Thesis of Mathieu Lee Giroud is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2022

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to acknowledge Professor Jan Kleissl for his support as the chair of my committee. His multiple reviews and comments on my written work were very valuable to improve its quality. Moreover, he was always available to answer questions and provide feedback. I would also like to thank my committee members for letting my defense be an enjoyable moment.

I would also like to give special thanks to Adil Khurram, a postdoctoral scholar working under Professor Jan Kleissl, with whom I met on a weekly basis. He guided me throughout the research process for all three quarters and his help was invaluable. His help was invaluable and I am very grateful to have worked with.

I would also like to thank Keaton Chia, who helped me set up the hardware required for my research. It was always a pleasure to discuss this project with him.

Finally, I would like to acknowledge Scott Brown for his previous work on developing the software to run the DANA algorithm, which served as a good reference for my research.

ABSTRACT OF THE THESIS

Distributed Control of Plug Loads for Building Energy Management

by

Mathieu Lee Giroud

Master of Science in Engineering Sciences (Mechanical Engineering)

University of California San Diego, 2022

Professor Jan Kleissl, Chair

Management of demand side resources can play a significant role in enabling renewable energy integration and decarbonizing the electric grid. Plug loads constitute a major portion of energy consumption of commercial buildings. They can be turned off using smart plugs when not in use. Smart plugs can also infer building occupancy from energy consumption measurements of plug loads. Therefore, plug load control can conserve energy and provide load flexibility to the grid for frequency regulation. This work focuses on a distributed control algorithm, distributed approximate Newton algorithm (DANA). DANA uses local information to optimize a network of nodes to track a reference signal using only agent-to-agent communication. Moreover, the power consumption of

loads can be used as inputs to the algorithm in the form of box constraints to account for load usage. In this work, DANA is implemented on a real-life system that consists of computers, a TV monitor, and a printer. The objective of this work was to show that DANA could be used on a system of plug loads to track a reference signal while conserving energy. Experimental results show that DANA can be used to switch off idle plug loads to conserve energy. In an office space with high occupancy the DANA implementation reduced energy consumption by 33% over one hour. Including a battery in the system can reduce the average tracking percent error to less that 1% and can therefore be used to provide frequency regulation services.

1 Introduction

1.1 Motivation

In order to reduce carbon emissions to mitigate climate change, a large amount of renewable energy resources are being integrated into the power grid. However, the variability of renewables, such as solar and wind, creates challenges for the grid operators to balance supply and demand. Therefore, new control methods that can account for that variability need to be implemented. Distributed Energy Resources (DERs) have become a topic of interest in the context of renewable energy integration in the smart grid because of their ability to accommodate the high variability of wind and solar power.

DERs allow for flexible generation and load management and can be coordinated via centralized or distributed load coordination schemes to provide ancillary services such as frequency regulation. In centralized schemes, control signals are generated by a central entity based on the current DER states. This type of scheme requires large amounts of data to be exchanged between the controller and DERs which further raises privacy concerns. In distributed and decentralized control schemes, on the other hand, control decisions are made locally by each DER based on its local state information and data is shared only with its neighbors. This reduces the burden on communication infrastructure and allows for parallel computation which is beneficial when the network size increases [1]. Therefore, this work focuses on a distributed control algorithm for coordinating DERs.

1.2 Literature review

In distributed algorithms, the network optimization problems are solved in a distributed manner using only the local information (e.g. the On/Off state) at each node. A ratio-consensus has been proposed in [2] in which controllable nodes can both inject and consume power from the grid within their operating capacities. Each node seeks to achieve consensus on the ratio of operating capacity so that the aggregate tracks a power reference signal. Similarly, a distributed algorithm is proposed in [3] based on the primal-dual dynamics of augmented Lagrangian. However, these algorithms have mostly been validated on DERs such as batteries, thermostatically controlled loads etc. Implementation of plug loads is not generally considered.

This work uses the distributed approximation Newton algorithm (DANA) developed in [4] to coordinate plug loads. DANA has previously been validated on flexible loads and battery energy storage [6] and incorporates constraints on the maximum and minimum power limits of DERs called box constraints. However, [6] considers flexible loads and batteries whose output power can vary continuously between power limits.

Plug loads are usually controlled based on occupancy information. The authors of [7][8] used sensors such as cameras or magnets placed on doors to determine occupancy. Loads will then be switched On if the room is occupied and vice versa. Using sensors not only requires additional hardware, but could result in higher energy consumption when multiple devices are present in the same area while only one of them may actually be in use. Therefore, smart plugs are preferable, as they can both measure power and actuate devices.

Demand response using smart plugs in buildings have been considered in [9]-[12]. While [11] collected extensive data and was able to predict occupancy with an accuracy of 80%, the authors did not use their findings to actuate loads and save energy. Smart plugs are used to control loads in [12], however, the authors studied the effects of switching On and Off groups of loads rather than individual devices. This work takes a more granular approach than [12] and considers variable usage schedules of individual plugs loads as explained next.

1.3 Statement of contributions

This work considers the control of plug loads using the DANA distributed control algorithm to provide grid services while conserving energy. The implementation of DANA is in real-time and uses box constraints. The algorithm is implemented on a test system using Raspberry Pis, Best Energy Reduction Technology (BERT) smart plugs, and actual plug loads. Each BERT smart plug measures voltage, current, power, and energy of the load connected to it. Furthermore, each smart plug is equipped with a relay that can switch the load On or Off based on the control signal received by DANA. The novelties of this work are:

- Smart plugs loads under DANA can contribute to building energy management [13] by switching Off loads that are idle. Feedback from load power measurements is used to determine idle loads.
- The tracking and curtailment benefits of including a battery in a plug load system is evaluated.

1.4 Paper Overview

This paper first presents some background on frequency regulation and energy savings in Section 2. It then describes the methodology in Section 3 including the DANA algorithm and its adaptation to the constrained formulation with box constraints. Furthermore, the implementation of DANA paired with energy saving measures and the topology of the system is also discussed. Section 4.1 then illustrates the test scenarios as well as the results showing the ability of the proposed system to track reference signals and save energy. Finally, section 5 concludes the paper and discusses possible future work.

2 Problem setting

This works focuses on evaluating the potential of plug loads to provide frequency regulation services to the grid, with and without energy storage. Furthermore, the potential energy savings, that can be achieved by turning Off plug loads when not in use, are also quantified.

2.1 Frequency regulation

Power grid systems must maintain a constant frequency, typically 50 or 60 Hz, in order to remain stable [14]. When the system frequency becomes too high or too low, grid operators use frequency regulation techniques to ensure that it does not deviate too far from its set point. In traditional power grids containing legacy equipment, generators can regulate frequency by tuning their power output [14]. The uncertainty associated with the increase of renewable energy sources in the power grid increases the need for frequency regulation. While DERs cannot regulate frequency using traditional methods because of their lack of inertia-based generators, they can participate in frequency regulation using flexibility from loads and storage resources [6].

2.2 Energy savings

The energy usage of plug loads can be reduced by implementing simple control methods. Consider the energy consumption of a computer, TV and a printer, as shown in Figure 2.1 for a single day in a building on the UCSD campus. Figure 2.1(a) demonstrates energy saving measures that could be taken on a computer. The computer is considered to be in use if its power consumption is above the threshold as shown within the red shaded area in Figure 2.1(a). Outside of these time periods, the computer is not expected to be in use. Therefore, the computer could be turned off after 10 minutes of inactivity to conserve energy. The potential energy savings by switching off the computer are obtained by taking the ratio of the total energy consumed within the shaded area in Figure 2.1(a) to the total energy consumed during the day. From 3/9/2022 to 3/28/2022, for this computer, the potential energy savings come out to be 93.94%. The same idea applied to the printer in Figure 2.1(b) would have resulted in 91.25% less energy consumption.

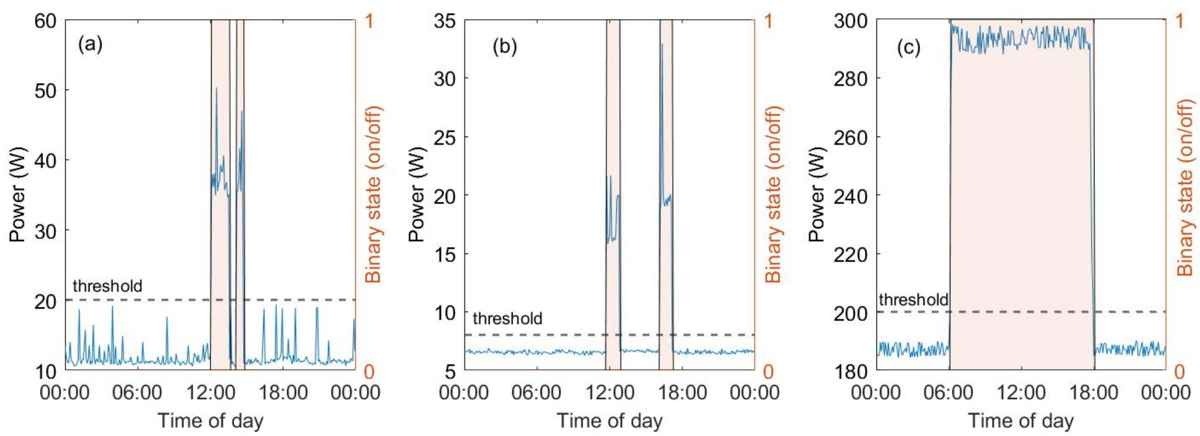


Figure 2.1. Daily energy consumption profile measured at 5 min intervals for: (a) Computer, (b) Printer, (c) TV. Red background represents the On state of the loads if the control scheme discussed in this section is implemented

Figure 2.1(c) represents another source of potential savings for a TV that could be achieved by usage-based smart plugs. In the current system, the TV is set to a pre-determined schedule and is turned on from 6:00 to 18:00 h while it is set to stand-by from 18:00 to 6:00 h, but still uses a significant amount of energy. Two energy savings measures could be taken: (i) Turn Off instead of stand by: TV could be turned off from 18:00 to 6:00 following the same criteria used for the computer and printer. During this time span, the TV would be consuming no energy rather than the 190 W it is consuming in the current control scheme. This method alone would have reduced energy consumption of the TV by 33.52% from 3/9/2022 to 3/28/2022. (ii) Occupancy-based: The state of the TV could be controlled in relation to the usage of nearby loads such as computers, which would indicate that a user is in the same space and could benefit from the TV being on. However, if none of these loads are in use, then we could turn Off the TV because no one would be able to view it. Even greater savings could be achieved by incorporating the occupancy and usage of other loads in addition to the straightforward method presented above and is the focus of this work. The next section formulates the distributed algorithm DANA with the objective to track a reference signal while minimizing energy consumption.

3 Methodology

This section first describes the DANA algorithm as well as the addition of box constraints. It then explains how the box constraints are determined using knowledge of the load power consumption and how the box constraints are used as input into the DANA algorithm. Finally, we describe the test setup.

3.1 Distributed approximate Newton algorithm (DANA)

The energy distribution in the system is optimized using a distributed approximate Newton algorithm that was developed in [4]. To understand how the algorithm works, preliminaries on graph theory are presented next.

3.1.1 Unconstrained formulation

Let a network of agents be represented by an undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N} = \{1, \ldots, n\}$ is the set of nodes and edges and $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ is the set of edges. If the edge set $\mathcal{E}$ has elements $(i, j) \in \mathcal{E}$ then $j \in \mathcal{N}_i$ are the one-hop neighbors of i where $\mathcal{N}_i \subset \mathcal{N}$. Similarly, the union of the neighbors of agent $j \in \mathcal{N}_i$ is the set two-hop neighbors of agent i and is represented by $\mathcal{N}_i^2$. Each node of the network possesses a local cost function of the form,

$$f_i = \frac{1}{2} a x_i^2 + b x_i$$

where x_i is the decision variable and (a, b) are constant coefficients of the quadratic function. DANA aims to optimize the aggregate of the cost functions f_i at each node $i = \{1, \ldots, n\}$ so that the total demand is d. This optimization problem can be written as,

$$\min_{x} \quad f(x) = \sum_{i=1}^{n} f_i(x_i) \tag{3.1a}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} x_i = d \tag{3.1b}$$

The DANA algorithm for this unconstrained problem was formulated in [4] and is the basis for Algorithm 1. To solve this problem, a descent method in x is used which can be written as,

$$x^+ = x + \alpha L \tilde{z}_{nt} \tag{3.2}$$

where L is the graph Laplacian, $\tilde{z}_{nt}$ is an approximate Newton step and α is a fixed step size. The Newton step can be written as,

$$L\tilde{z}_{nt} = -L \sum_{p=0}^{q} (I_n - LH(x)L)^p L \nabla_x f(x) \tag{3.3}$$

where $\nabla_x f(x)$ is the gradient of f(x) with respect to x and $H(x)$ is the Hessian matrix with respect to x. For brevity, we do not present the steps required to obtain Equation (3.3), which are described in [4]. The algorithm that implements the unconstrained problem described above is presented in [4]. The method described above solves the unconstrained problem defined in Equations (3.1a) and (3.1b). If we wish to set limitations on the range of the decision variable x_i, it can be subjected to the following box constraints:

$$\underline{x_i} \leq x_i \leq \overline{x_i} \quad i = \{1, ...n\} \tag{3.4}$$

3.1.2 Box constraints

Box constraints are required for our application since loads can only take on positive power values. Lower and upper bounds constrain the decision variables ensuring that loads that are in use remain On and loads that are not in use remain Off. In [4], box constraints are added to the algorithm by modifying the problem statement defined in Equations (3.1a), (3.1b) and (3.4). Knowing that $\mathbf{1}_n$ is an eigenvector of L corresponding to the eigenvalue 0, we can write $\mathbf{1}_n^T(x^0 + Lz) = d$ and define a new objective function and box constraints using z as the new decision variable:

$$\min_{z} \quad f(x^0 + Lz) = \sum_{i=1}^{n} f_i(x_i^0 + L_i z) \tag{3.5a}$$

$$\text{s.t.} \quad \underline{x_i} - x_i^0 - Lz \preceq \mathbf{0}_n \tag{3.5b}$$

$$x_i^0 + Lz - \overline{x_i} \preceq \mathbf{0}_n \tag{3.5c}$$

where L is the Laplacian of the graph $\mathcal{G}$ and $\mathbf{0}_n$ is a zero-vector of length n.

This new objective function in (3.5a) constrained by (5b) and (5c) can be optimized by finding the saddle point of its Lagrangian. To compute the Lagrangian, we first define the matrix $P(z)$:

$$P(z) = \begin{bmatrix} \underline{x} - x^0 - Lz \\ x^0 + Lz - \overline{x} \end{bmatrix} \tag{3.6}$$

The Lagrangian $\mathcal{L}$ can be written as a function of the primal variable z and the dual variable λ,

$$\mathcal{L}(z, \lambda) = g(z) + \lambda^T P(z)$$

Furthermore, the gradient of the Lagrangian with respect to z and λ are:

$$\nabla_z \mathcal{L}(z, \lambda) = \nabla_z g(z) + [-L \quad L]\lambda, \tag{3.7}$$

$$\nabla_\lambda \mathcal{L}(z, \lambda) = P(z). \tag{3.8}$$

Algorithm 1. DANA with box constraints

procedure DANABOX($d_{new}, L_i, g_i, \underline{x_i}, \overline{x_i}, q$)
  if New demand d from reference node **then**
    $x_i^0 \leftarrow x_i^0 + (d_{new} - d_{old})/n_{nodes}$
    $z_i \leftarrow x_i^0$
    $d_{old} \leftarrow d_{new}$
    loop
      $P(z) = \begin{bmatrix} \underline{x}_i - x_i^0 - z_i L_{ii} - \sum_{j \in N_i} z_j L_{ij} \\ x_i^0 + z_i L_{ii} + \sum_{j \in N_i} z_j L_{ij} - \overline{x}_i \end{bmatrix}$
      $\lambda_i = \lambda_i + [P(z)]_\lambda^+$
      Communicate λ_i to one-hop neighbors
      Communicate z_i to one-hop neighbors
      $\dfrac{\partial g_i}{\partial z_i} = a_i L_{ii} z_i + b_i$
      Communicate $\dfrac{\partial g_i}{\partial z_i}$ to one-hop neighbors
      $y_i \leftarrow L_{ii} \dfrac{\partial g_i}{\partial z_i} + \sum_{j \in N_i} L_{ij} \dfrac{\partial g_i}{\partial z_j} + [-L_{ii} \; L_{ii}]\lambda_i + \sum_{j \in N_i} [-L_{ij} \; L_{ij}]\lambda_j$
      $s_i \leftarrow -y_i$
      $p_i \leftarrow 1$
      while $p_i \leq q$ **do**
        Communicate y_i to two-hop neighbors
        $w_i \leftarrow (I - LHL)_{ii} y_i + \sum_{j \in N_i^2} (I - LHL)_{ij} y_j$
        $y_i \leftarrow w_i$
        $s_i \leftarrow s_i - y_i$
        $p_i \leftarrow p + 1$
      end while
      Communicate s_i to one-hop neighbors
      $z_i \leftarrow z_i + \alpha(L_{ii} s_i + \sum_{j \in N_i^2} L_{ij} s_j)$
    end loop
    $x_i \leftarrow x_i^0 + L_{ii} z_i + \sum_{j \in N_i} L_{ij} z_j$
  end if
end procedure

To find the saddle point of this equation, we use Newton-like ascent dynamics in λ and descent dynamics in z, which can be respectively written in continuous time as:

$$\dot{\lambda} = [\nabla_\lambda \mathcal{L}(z,\lambda)]^+_\lambda = \begin{cases} \nabla_\lambda \mathcal{L}(z,\lambda), & \lambda > 0 \\ max(0, \nabla_\lambda \mathcal{L}(z,\lambda)) & \lambda \leq 0 \end{cases} \tag{3.9}$$

$$\dot{z} = -L \sum_{p=0}^{q} (I_n - LH(x^0 + Lz)L)^p L \nabla_z \mathcal{L}(z,\lambda) \tag{3.10}$$

Using the gradients of the Lagrangian defined in Equation (3.7) and Equation (3.8) and converting the continuous dynamics to discrete using the Euler method, we can finally write (3.9) and (3.10) as:

$$\lambda^+ = \lambda + [P(z)]^+_\lambda \tag{3.11}$$

$$\tilde{z}_{nt} \sum_{p=0}^{q} (I_n - LH(x^0 + Lz)L)^p L(\nabla_z g(z) + [-L \;\; L]\lambda) \tag{3.12}$$

Finally, the decision variable that the original optimization problem was solving is:

$$x = x^0 + Lz \tag{3.13}$$

We can now write Algorithm 1 for the new optimization problem defined in Equation (3.5a) constrained by Equations (3.5b) and (3.5c). We are now optimizing the decision variable z, which is initialized on line 4. In the iterative process starting on line 6, the algorithm first computes Equations (3.6) and (3.11) on lines 7 and 8 respectively. Then, the algorithm calculates Equation (3.12) in three steps: First, $L\nabla_z g(z)$ on the right side of the equation is computed in line 13 by computing the gradient in the outer loop. Second, the exponential term is calculated in the while loop at line 16. Finally, the approximate Newton step is obtained by multiplication with the Laplacian in line 24 which computes

Equation (3.2). Once the loop is terminated, we solve for the original decision variable x_i on line 26.

3.2 Load types and usage

Office buildings house a large variety of plug loads, including computers, TVs, copiers, printers, scanners, coffee makers, water dispensers etc. For simplicity, we focused on the most common types of plug loads found on the UCSD campus, which are laptops/computers, TVs/monitors, and printers/scanners/copiers. Based on energy usage data from the UCSD campus, copiers, printers and scanners exhibit very similar behaviors, similar to what is shown Figure 2.1(b). While desktop and laptop computers behave similarly, their power consumption vary in magnitude, as can be seen in Figure 3.1. Finally, TVs are significantly different from other types of load because of their strictly binary power consumption profile, as shown in Figure 2.1(c).

In order to manage these loads optimally, we must first understand how to determine their optimal state. In the implementation, the box constraints defined in Section 3.1.2 are used to dictate which loads must remain On because they are in use and which loads which should be turned Off because they are not. To decide which loads can be turned Off to save energy, we must know whether the load is being used. A load can exhibit 3 modes of operation, that is, in use, idle and Off. From a usage perceptive, plug loads can further be divided into independent and dependent loads. The mode of independent loads, such as computers, can be determined by their current power consumption and a threshold.

The value of the threshold is determined before executing the algorithm by inspecting the load power consumption when in use or in an idle mode of operation. Figure 3.1 shows the threshold for four different devices. Using this data, we visually selected a threshold to distinguish in use and idle modes. As Figure 3.1 shows, laptop (a) and desktop computer (d) show clearer differences between the two states.Unlike computers, TVs and printers are dependent plug loads. For simplicity, we assume that the need of dependent plug loads to be On can be inferred from the usage of other plug loads in their vicinity even though occupancy may not always be determined by plug load power consumption only. In other words, a printer or TV only need to be switched On if a computer is being used in the same area. While e.g. a TV in a conference room should be turned On whenever someone is in the room, the printer use can vary depending on which user is present as some users might never use printers.

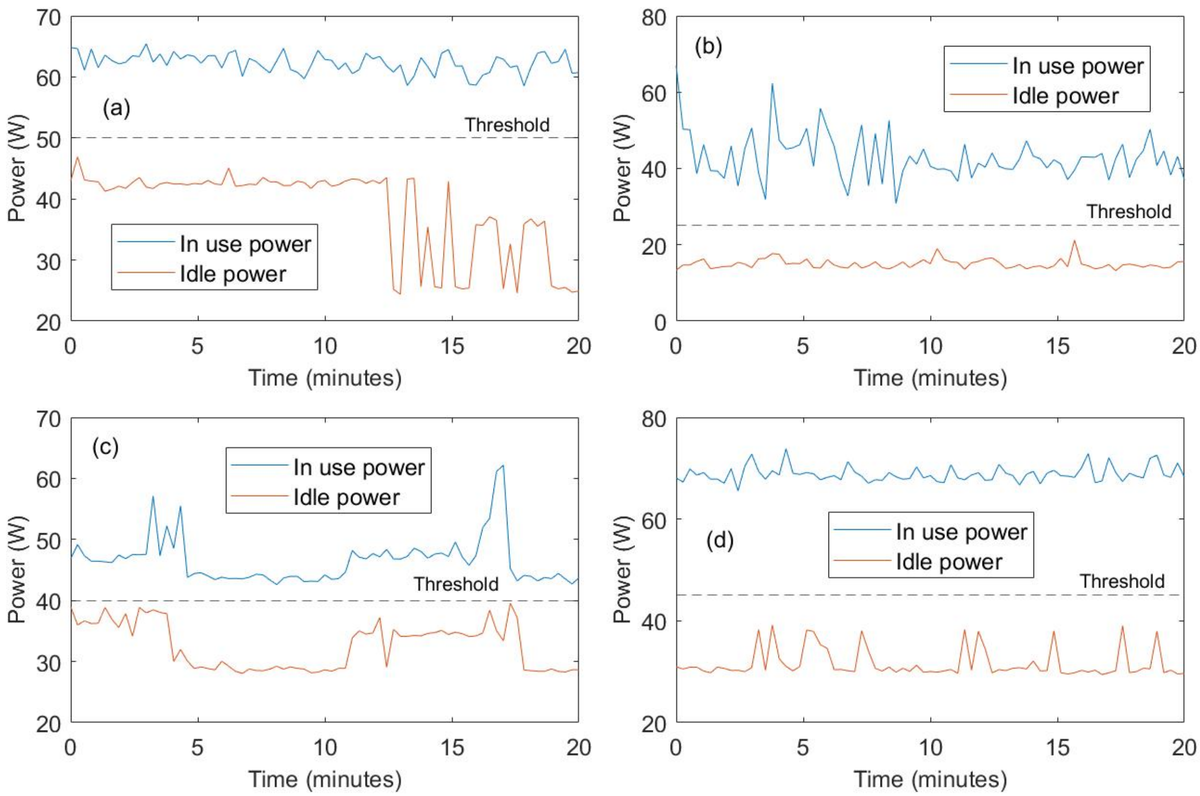


Figure 3.1. Power profiles for (a)-(b) laptops and (c)-(d) tower computers with threshold used to determine the mode of operation (in use versus idle) of the loads

We assumed relationships between the different loads for simplification purposes. First, we assumed that if any computer was in use, then the TV must be turned On. This relationship can be observed for TVs that display information for users in the same area or TVs that are used as a larger monitor for laptops and computers. Second, we considered that if a chosen subset of devices was in use, then the printer should be turned On. In contrast, the state of the printer was independent of the usage of devices outside of that subset. In the future, a prediction model will be employed to determine the dependencies between loads.

3.3 Implementation

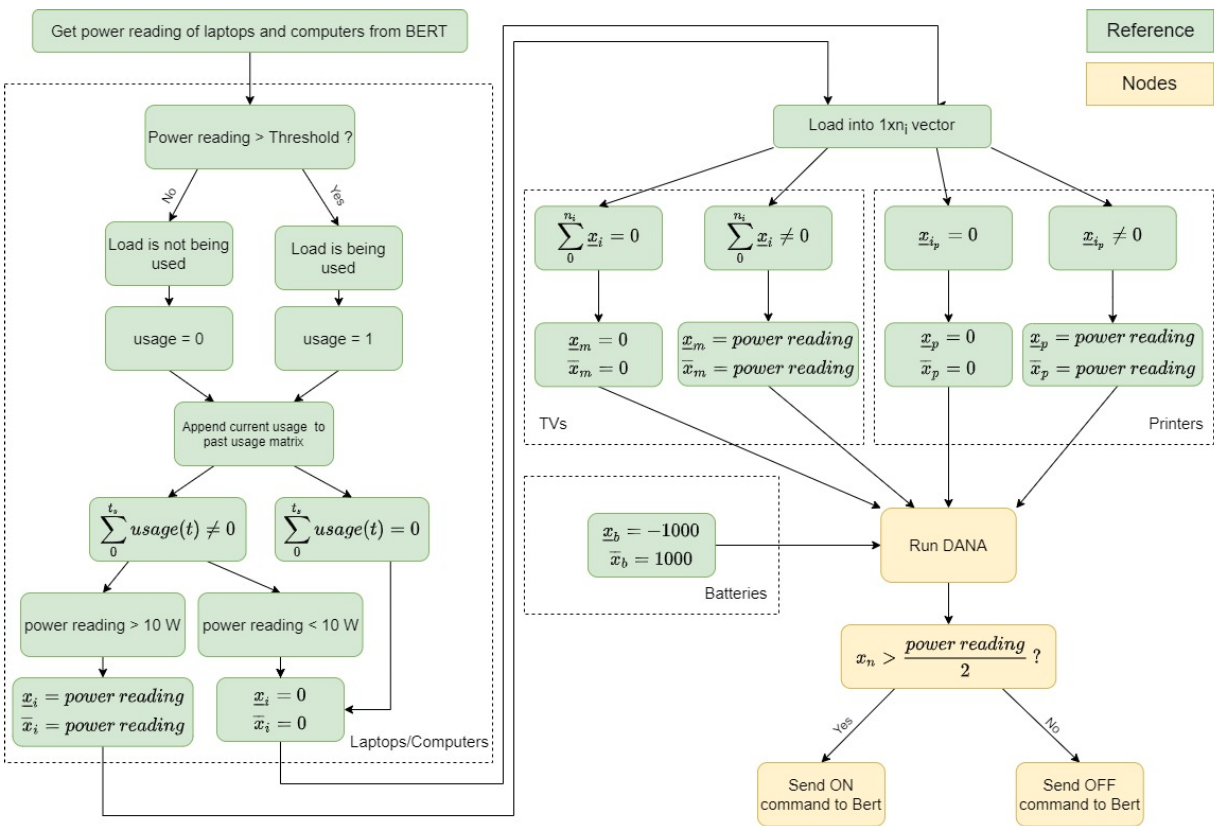


Figure 3.2. Implementation flowchart for system with battery

The system operates as a closed loop as shown in Figure 3.2. The reference node is responsible for retrieving power data from all devices. By comparing the power consumption of the laptop and desktop computers to their pre-determined thresholds, the reference node determines whether those devices are idle or in use. If the power reading is greater than the threshold, a value of 1 is assigned to the usage variable. Otherwise, a value of 0 is assigned. The values are loaded into a vector whose indexes n_i correspond to each computer or laptop device. The $1 \times n_i$ vector is then appended to a $s \times n_i$ matrix containing usage data for the past s iterations with $s = \dfrac{t_s}{\Delta t}$, where t_s is the time in idle mode before a load is switched off and Δt denotes the time step. At every iteration, the first row of that matrix is removed and the new usage value is appended, resulting in an updated $s \times n_i$ matrix with usage data for the past s iterations. To determine whether a device is not in use and can remain off or turned off, each column of the $s \times n_i$ matrix is summed. If the sum of a column is 0, then the device has not been used for that period of time and we can assume it can be turned off. In that case, the lower box constraint $\underline{x}_i$ and the upper box constraint $\overline{x}_i$ are both set to 0 to force the algorithm to converge to 0. For the system without a battery, these constraints were relaxed and $\overline{x}_i$ was not enforced if a load was unused, which reduced the number of unfeasible scenarios and allowed loads that were not in use to be turned on. On the other hand, if the sum is not equal to 0, then the device was in use in the past t_s minutes and we assume that it needs to remain on. To ensure that the device is in fact still in idle mode, we check that its power reading exceeds 10 W, which would indicate that the device has not been turned off by the user. In the case where the device was used in the previous t_s minutes and is still On, the lower box constraint $\underline{x}_i$ and the upper box constraint $\overline{x}_i$ are both set to the latest power reading of the device to force the algorithm to converge to the actual power consumption of the load.

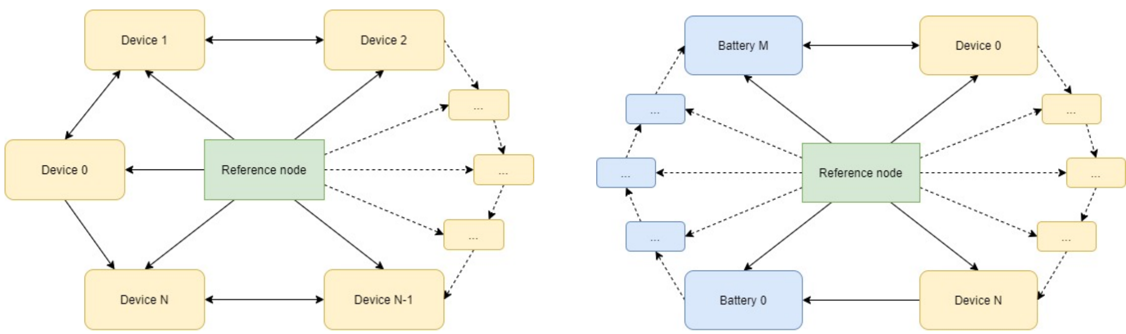
The usage of the remaining dependent nodes n_d is determined using the relationship described earlier in this section. The $\underline{x}_i$ and $\overline{x}_i$ values of the laptops and computers are loaded into a vector. If the sum of the lower constraints $\underline{x}_i$ for the laptops and computers is not equal to 0, then at least one of these devices is in use. In this case, the $\underline{x}_m$ and $\overline{x}_m$ for the TV are set to its latest power reading. As described earlier in this section, if the $\underline{x}_i$ for a subset of nodes $n_p \in n_i$ (denoted x_{i_p} in Figure 3.2), is not equal to 0, then at least one of these devices is in use and the $\underline{x}_p$ and $\overline{x}_p$ are set to its latest power reading. On the other hand, if the the sum of the $\underline{x}_i$ values is equal to 0, then none of the devices are in use and the TV can remain off or be turned off. Similarly, if $\underline{x}_{i_p}$ is equal to 0, then the devices dictating the state of printer are not being used and the printer can be turned Off. The constraints for the TV and printers are then appended to the previous vectors, creating $1 \times n$ vectors containing lower and upper bounds for all devices n. If the system contains a battery, then a last set of constraints are added to those vectors. For simplification purposes, the simulated battery does not have power and energy capacity limits. For those reasons, the lower and upper bounds for the battery, $\underline{x}_b$ and $\overline{x}_p$, are set to -1000 W and 1000 W respectively.

To implement these constraints in the algorithm, the vectors containing lower and upper bounds for all devices n, along with the power readings of each device, are sent to all Raspberry Pi controllers. The controllers then extract the constraint values and power measurements for their respective index defined in Figure 3.3 and use it as an input to the DANA algorithm. If the output of the DANA algorithm x_n on a device is greater than half of its latest power reading, then the BERT linked to that device is set to an On state. Otherwise, the BERT is set to an Off state. This condition is a consequence of the binary state (On/Off) of plug loads. Unlike flexible loads, a plug load's power consumption cannot be set to the algorithm's solution. Therefore, an approximation must be used to determine the optimal state of the load.

3.4 Setup

As Figure 3.3 shows, the devices were set up in a closed ring topology. In a closed ring topology each node communicate with exactly 2 other nodes. This setup was used (a) without a battery or (b) with a battery, since the battery is treated as a node. For simplicity, batteries were placed in between the last and first device nodes. The reference node can send data to all nodes, but the devices do not send back data to the reference node, as indicated by the direction of the arrows. However, device nodes can communicate with their 1-hop ring neighbors, from which they can also obtain information about their

2-hop neighbors. The data exchanged between neighbors is given by DANA and described in section 3.1. Each plug load is controlled by a Raspberry Pi which is responsible for the execution of DANA, data exchange with other Raspberry Pis and for switching between the On/Off states of the load. BERT smart plugs are used to track power usage at each node and communicate this data to the reference node. All Raspberry Pis and smart plugs are connected to the same TCP/IP network.



(a) System without storage

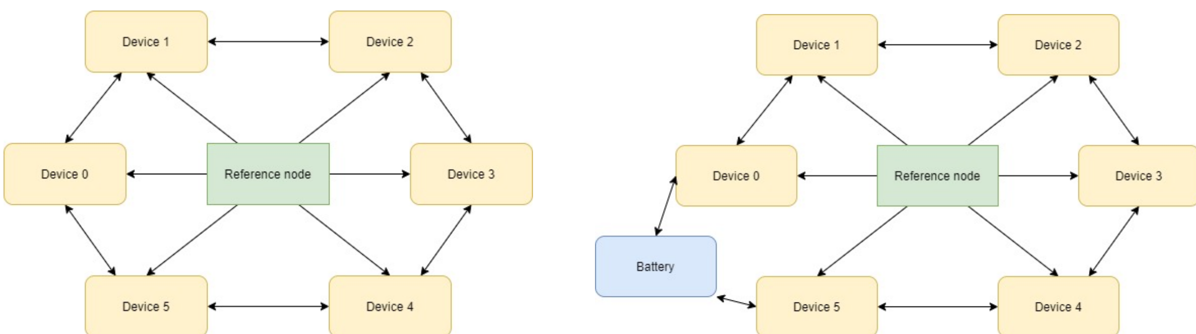
(b) System with storage

Figure 3.3. General layout for systems with and without storage

4 Results and Discussion

This section describes the results and discussion of the two different types of tests (constant set point and decreasing set point) that were executed on two systems, one with and one without the battery).

4.1 Tests Scenarios and Setups



(a) System without storage **(b)** System with storage

Figure 4.1. Layout for systems with and without storage used in the experiment

To demonstrate the algorithm ability to control heterogeneous plug loads, we designed a test system composed of 2 computer towers, 2 laptops, a small TV screen and a printer. This setup emulates a small office space. As shown in Figure 4.1(a), the devices are set up in a ring and are connected to a reference node. Devices 0 and 1 are the laptop computers displayed in Figure 3.1(a)-(b), devices 2 and 3 are the computer towers shown in 3.1(c)-(d), device 4 is a TV and device 5 is a printer. In the second setup, a virtual battery was added between node 0 and 5 to improve signal tracking and limit energy curtailment (see Figure 4.1(b)).

The thresholds for the laptops and computers were determined by inspecting the power consumption profiles from Figure 3.1. These thresholds were recorded in Table 4.1 and used for all tests.

Table 4.1. Load types and threshold to distinguish On and Idle modes. Devices 4 (TV) and 5 (printer) are dependent devices and do not require a threshold.

Device	0	1	2	3	4	5
Load type	laptop	laptop	computer tower	computer tower	TV	printer
Threshold (W)	50	25	40	45	N/A	N/A

The algorithm constants shown in Table 4.2 were used for all nodes and tests. The coefficients a and b were made uniform for all nodes to simplify the overall problem. To pick the remaining constants, α, k_{max} and q, several tests were run using different values. The combination of values chosen here yielded the best convergence rate for the 2-minute time step we were using.

Table 4.2. Constants used for testing

a	b	α	k_{max}	q	Δt (min)	t_s (min)
0.2	0	0.01	100	5	2	10

The setups with and without a battery were tested with the same two tracking signal scenarios:

1. Their ability to handle changes in usage for some devices, as if people were coming and leaving an office, while tracking a constant reference signal.

2. Their ability to track a slow reduction in available power, similar to the end of a day when solar power output decreases.

The tests were performed in a controlled environment in which user behaviors were mimicked. During scenario 1, the independent loads, i.e. the laptops and computers (devices 0-3), followed a usage pattern similar to the ones found in Figure 5.1 in the appendix. The actual schedule used in scenario 1 can be found in Table 4.3. For scenario 1, the reference signal was set to 150 W throughout. Scenario 1 was designed to test whether a plug load system (without and with storage) could track a constant reference signal within the constraints determined by the usage of the individual plugs loads. In other words, scenario 1 is designed to test whether a plug load system can provide frequency regulation without overriding the states defined by the schedule in Figure 5.1. This scenario also measured the efficacy of the energy savings control scheme presented in Section 3.3, i.e. whether a load was turned off after 10 minutes of idle mode and if the relationships between independent and dependent loads were maintained.

For scenario 2, the system tracked a slowly decreasing reference signal, while devices 0 and 2 were in use while devices 1 and 3 were not in use. The reference signal was chosen to simulate the power output of a solar-dominated power grid, where solar production would diminish in the afternoon. The reference signal is actual solar power output from 13:00 at 168 W decreasing to 106 W at 14:00 h [5]. The purposed of this scenario was to measure the plug load system's ability track a changing reference signal given a constant usage of loads and limited ability for energy curtailment.

Table 4.3. Scenario 1 schedule for the independent loads.

Time (minutes)	device 0	device 1	device 2	device 3
0	OFF	IN USE	OFF	OFF
10	OFF	IN USE	IN USE	OFF
20	OFF	IN USE	IN USE	IN USE
30	IN USE	IN USE	IN USE	IN USE
40	IN USE	OFF	IDLE	IN USE
50	OFF	OFF	OFF	OFF

4.2 Scenario results and implications

The following results were evaluated according to these criteria: (1) How well did the system track the reference signal? (2) Did the system turn Off devices that were in use? (3) Was energy saved either by turning Off loads in idle mode or dependent loads or by storing excess energy?

4.2.1 Scenario 1: Constant reference

Consider first the system without storage tracking a constant reference signal of 150 W. In Figure 4.2, DANA is used to control plugs loads as described in Section 3.2. Figure 4.2(b) shows that DANA attempts to schedule the plug loads to track the reference signal, but the tracking error is large. There are 3 possible explanations for these errors. (i) When the reference signal is higher than the total power consumption from all loads and the system is able to switch On loads not in use ($t = 0$ min to $t = 10$ min),the system fails to track the signal because of a lack of resolution. As Figure 4.2(a) shows, even though only device 1 was in use, the system switched on devices 0 and 3 in an effort to track the reference signal. Nonetheless, each of these loads account for a significant portion of the reference signal and therefore cannot help track the load precisely. In a system with more loads, each load would represent a smaller fraction of the reference signal, resulting in more granular tracking. (ii) When the reference signal is lower than the total power consumption from all loads ($t = 20$ min to $t = 50$ min), then the system is infeasible. When this occurs, we can also observe that loads in use are switched off, as it can seen with device 1 at $t = 28$ min and device 2 at $t = 32$ min (see Figure 4.2(b). Therefore, in a system without storage, the plug loads have to be selected to provide sufficient flexibility or tracking errors have to be accepted. In this scenario, the reference signal was purposely chosen to be low to show these limitations. (iii) When the reference signal is higher than the total power consumption from all loads and the system is unable to switch On loads not in use. Some loads, such as laptops and computers, can be switched Off by users and turning On smart plugs will not change the power consumption of the device. This issue can observed from $t = 50$ min to $t = 60$ min when none of the loads are in use. In this situation, the algorithm switched On the smart plug connected to device 0, which had been turned off and therefore could not draw power.

The results in Figure 4.2 show the limitations in the system without storage scenario when precise tracking is required. In contrast to plug loads with only On/Off states, flexible loads such as HVAC or advanced lighting systems [15][16] can vary their output power and would make the system tracking more granular. The tracking performance would therefore improve. However, the system operation would still be limited by the magnitude of the reference signal, i.e. if the sum of the power consumption of all loads in use exceeds the magnitude of the reference signal, then the problem is infeasible. A storage system, on the other hand, would increase the granularity of the system and allow for devices to operate normally even when their total power consumption exceeds the demand signal.

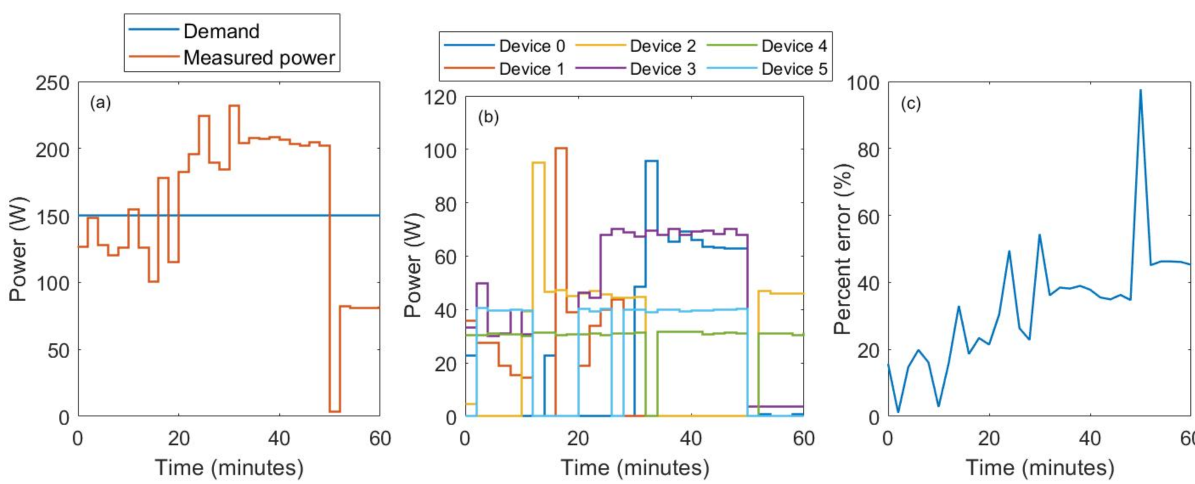


Figure 4.2. Scenario 1 for the system without a battery. Timeseries of a) individual device power, b) reference signal and measured total power, and c) percent error with respect to demand signal.

The system with storage performed much better. Figure 4.3 demonstrates the ability of the system to track a constant reference signal when loads are being turned on and off. In this scenario, the system was able to track the signal with errors of less than 2% as is shown on Figure 4.3(c). As expected, the battery charged when few loads were in use (from t = 0 min to t = 20 min and t = 50 min to t = 60 min) and discharged when the system was drawing more power (from t = 20 min to t = 50 min).

This experiment also shows successful energy savings.For independent plug loads, Figure 4.5(a) shows that device 3 went into idle mode at t = 40 min and was turned off 10 minutes later at t = 50 min. Depending on device's sleep schedule or idle mode power consumption, this measure could reduce energy consumption by up to 94%, as discussed in Section 1.1. In this experiment, the system consumed 32.81% less energy than its uncontrolled version by switching off loads that were not in use and storing excess energy (see Table 4.5.For dependent plug loads, device 5 (printer) followed the same usage pattern as device 3. Both of them turned on at t = 30 min and turned off at t = 50 min. In this experiment, this method reduced the energy consumption by 50%, assuming the printer would have been on during the entire simulation if it were uncontrolled. Device 4 (TV) was on when any of devices 1-4 were in use but was turned off when all other devices were off. Dependent plug load operation is more energy efficient than the current schedule-based operation, which keeps TVs on even when there is no occupancy. In this experiment, the dependent relationship reduced energy consumption by 16.67%.

Scenario 1 shows how adding storage improves tracking and greatly reduces error. The battery ensures that all loads in use can remain on even when the total power consumption is greater than the reference signal. Therefore, the battery is necessary to satisfy the first evaluation criterion. This test demonstrates energy savings by turning off loads in idle mode or switching off devices, such as TVs and printers, whose usage can be determined from laptops and computers in the same area.

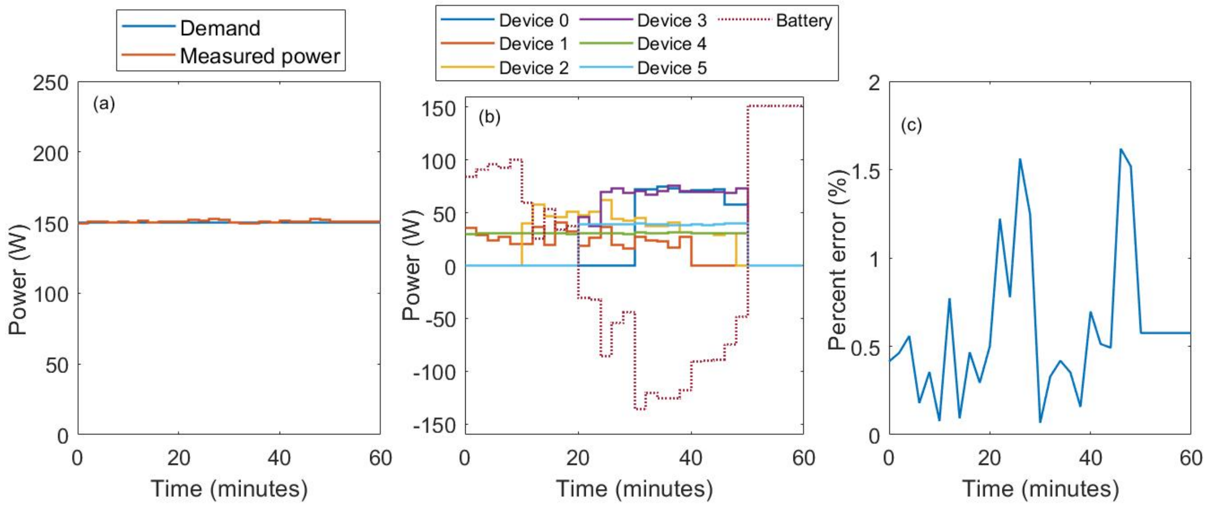


Figure 4.3. Scenario 1 for the system with a battery. Timeseries of a) individual device power, b) reference signal and measured total power, and c) percent error with respect to demand signal.

Table 4.4. Summary of results for scenario 1

Battery	Devices in use switched off	Average tracking error	Energy savings
No	1,2	33.03 %	N/A
Yes	None	0.60 %	32.81 %

4.2.2 Scenario 2: Incorporating variability in solar

For scenario 2, the system without storage also performed poorly. Figure 4.4(b) shows that there was significant overshoot when tracking the reducing reference signal from scenario 2 as reflected by the percent error displayed in Figure 4.4(c). In its attempt to track the reference signal, the system had to turn on device 3 even though they were not in use resulting in energy waste. The error is caused by a lack of continuously flexible loads or batteries in the system. In other words, the resolution of the system tracking is limited by the power consumption of each device. Due to the nature of the condition set in Figure 3.2 ($x > \frac{power\ reading}{2}$), the algorithm may converge to a value slightly higher than than half of a device's power reading and therefore turn on the device, but the device's actual power will be significantly higher, as shown in Figure 3.1. While the resolution of the system would improve on a system with more devices, it would still need to turn on unused devices to track the signal and therefore waste energy.

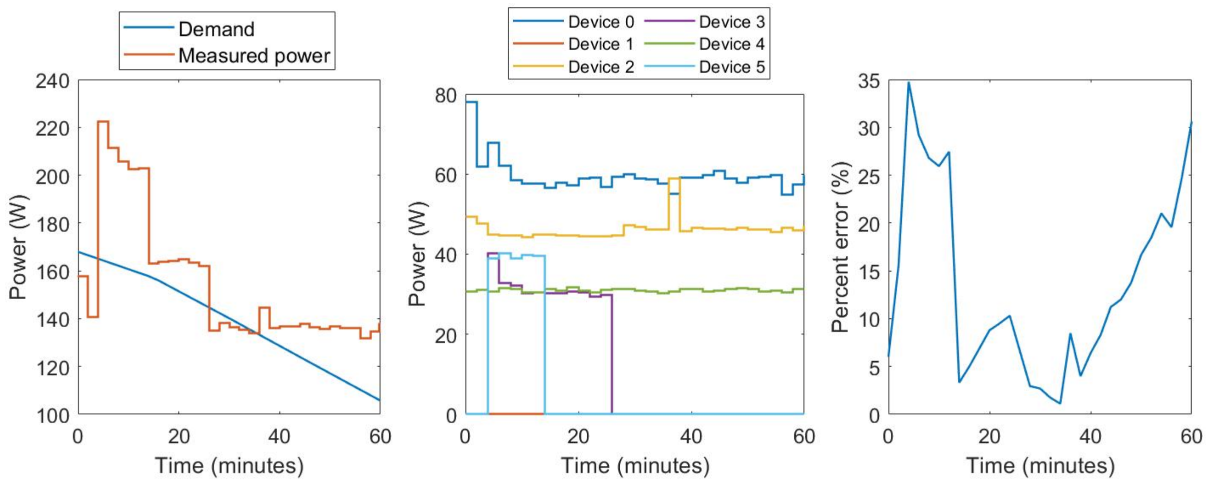


Figure 4.4. Scenario 2 for the system without a battery. Timeseries of a) individual device power, b) reference signal and measured total power, and c) percent error with respect to demand signal.

As in scenario 1, the system with storage performed significantly better than the system without storage. Figure 4.5 displays that the system with a battery can track a slow decrease in generation with a maximum error below 1% (see Figure 4.5(c). The increase in accuracy is provided by the battery, which can charge when the generation is greater than the total power of the loads and discharge when the power of the loads exceeds the generation. This trend can be seen in Figure 4.5(a), where the battery charges from t = 0 min to t = 42 min, when the reference signal is greater than the total power consumption, and discharges for the rest of the experiment, when the total power consumption exceeds the reference signal. Moreover, the battery is able to compensate for the variability in power consumption of the loads, which can also be observed in Figure 4.5b where the battery charge varies with respect to the power consumption of device 0.

In summary, including a battery in this end-of-day scenario greatly increases the system's ability to track a reference signal and contributes to the integration of solar energy in the energy mix. The battery reduces solar energy curtailment during the day, when

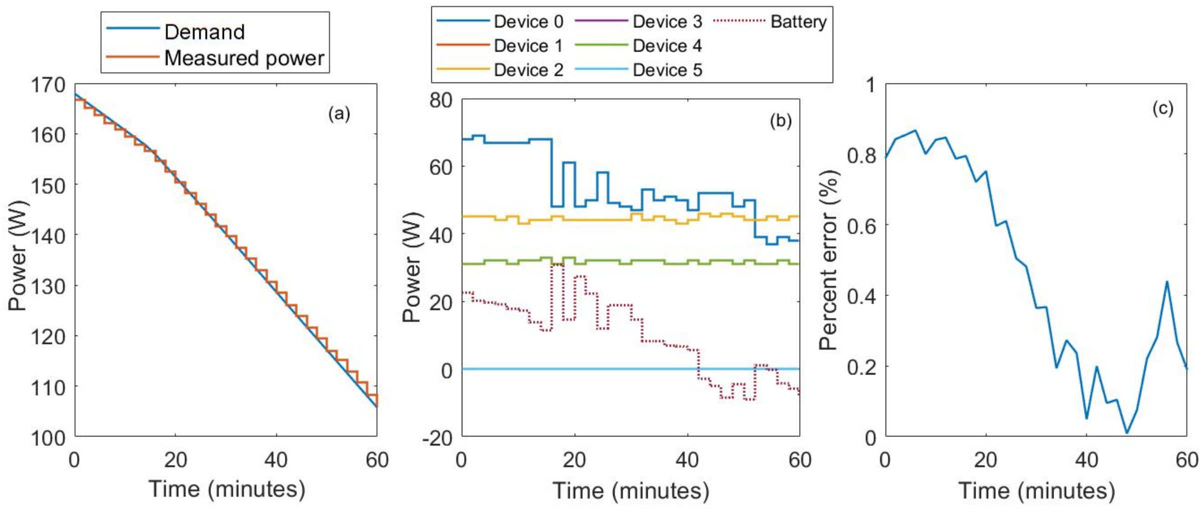


Figure 4.5. Scenario 2 for the system with a battery. Timeseries of a) individual device power, b) reference signal and measured total power, and c) percent error with respect to demand signal.

solar outputs the most power. At the end of the day, when solar output has decreased, the battery supplies energy to the grid and therefore reduce the amount of power needed to be produced fossil generators and limit the required overbuilding of solar energy.

Table 4.5. Summary of results for scenario 2

Battery	Devices in use switched off	Average tracking error	Energy savings
No	None	13.54 %	N/A
Yes	None	0.47 %	7.30 %

5 Conclusions and future work

This thesis presented the first implementation of the Distributed Approximate Newton Algorithm (DANA) on (binary) plug load control. To apply DANA to a test system with plug loads, we added box constraints to the previously developed unconstrained algorithm. Moreover, we added feedback to the system by inputting the power consumption of the devices to DANA. Communications with smart plugs facilitated adding this feedback into DANA by providing energy measurements used in the algorithm. Real-time measurements also contribute to energy savings by switching off unused devices.

The system without storage showed poor performance against the criteria laid out in Section 4.2. The poor tracking performance can be attributed to resolution and infeasibilities. Infeasibilities result from too large or too small of a magnitude of the reference signal with respect to the total power consumption of the loads in use. Adding energy storage helps with tracking the reference signal by storing excess energy and supplying power when the total power consumption is greater than the reference signal. As an alternative to energy storage, resolution issues could be mitigated / signal tracking would improve with the addition of continuously flexible loads, as excess energy could be consumed rather than stored.

While this work combined signal tracking and energy savings, each of these goals could be achieved individually. If one was solely interested in signal tracking, they could run the DANA algorithm with box constraints on system without switching off loads in an idle state. On the other hand, if one was interested in energy savings only, they could implement the control scheme developed in Section 3.3 and actuate loads directly without running the DANA algorithm.

To further this work, a more realistic battery with power and energy capacity limits could be substituted into the system. Flexible loads such as HVAC could be included in the simulations to provide more granularity and flexibility as well as another type of load. Furthermore, priority levels could be implemented, which would dictate which loads to turn Off first when trying to reduce energy consumption [12][17]. Specialized software utilities could be developed that put computers in sleep mode rather than switching them Off, which would ensure a graceful shut down and not result in any data loss [10]. We hope that this work can serve as a resource for the implementation of the DANA and other distributed algorithms on a larger scale.

Bibliography

[1] Molzahn, Daniel K.,Florian Dörfler, Henrik Sandberg, Steven H.Low, Sambuddha Chakrabarti, Ross Baldick, and Javad Lavaei. “A Survey of Distributed Optimization and Control Algorithms for Electric Power Systems.” IEEE Transactions on Smart Grid, vol. 8, no. 6, Nov. 2017, pp. 2941–2962., https://doi.org/10.1109/tsg.2017.2720471.

[2] Dominguez-Garcia, Alejandro D., Christoforos N. Hadjicostis, and Nitin H. Vaidya. ”Resilient Networked Control of Distributed Energy Resources,” in IEEE Journal on Selected Areas in Communications, vol. 30, no. 6, pp. 1137-1148, July 2012, doi: 10.1109/JSAC.2012.120711.

[3] Cherukuri, Ashish, and Jorge Cortés. ”Distributed algorithms for convex network optimization under non-sparse equality constraints,” 2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton), 2016, pp. 452-459, doi: 10.1109/ALLERTON.2016.7852266.

[4] Anderson, Tor, Chin-Yao Chang, and Sonia Martinez. “Distributed Approximate Newton Algorithms and Weight Design for Constrained Optimization.” Automatica, vol. 109, 2019, p. 108538., https://doi.org/10.1016/j.automatica.2019.108538.

[5] Blair, Nate, Nicholas DiOrio, Janine Freeman, Paul Gilman, Steven Janzou, Ty Neises, and Michael Wagner. “System Advisor Model (SAM) General Description”, (Version 2017.9.5) Golden, CO: National Renewable Energy Laboratory.

[6] Anderson, Tor, Manasa Muralidharan, Priyank Srivastava, Hamed Valizadeh Haghi, Jorge Cortés, Jan Kleissl, Sonia Martinez, and Byron Washom. “Frequency Regulation with Heterogeneous Energy Resources: A Realization Using Distributed Control.” IEEE Transactions on Smart Grid, vol. 12, no. 5, 4 Feb. 2021, pp. 4126–4136., https://doi.org/10.1109/tsg.2021.3071778.

[7] Agarwal, Yuvraj, Bharathan Balaji, Rajesh Gupta, Jacob Lyles, Michael Wei, andThomas Weng. “Occupancy-Driven Energy Management for Smart Building Automation.” Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building - BuildSys ’10, 2010, https://doi.org/10.1145/1878431.1878433.

[8] Wang, Zhe, Tianzhen Hong, and Mary A. Piette. “Predicting Plug Loads with Occupant Count Data through a Deep Learning Approach.” Energy, vol. 181, 2019, pp. 29–42., https://doi.org/10.1016/j.energy.2019.05.138.

[9] Kamilaris, Andreas, Balaji Kalluri Sekhar Kondepudi, and ThamKwok Wai. “A Literature Survey on Measuring Energy Usage for Miscellaneous Electric Loads in Offices and Commercial Buildings.” Renewable and Sustainable Energy Reviews, vol. 34, 3 Apr. 2014, pp. 536–550., https://doi.org/10.1016/j.rser.2014.03.037.

[10] Weng, Thomas and Yuvraj Agarwal. ”From Buildings to Smart Buildings—Sensing and Actuation to Improve Energy Efficiency,” in IEEE Design & Test of Computers, vol. 29, no. 4, pp. 36-44, Aug. 2012, doi: 10.1109/MDT.2012.2211855.

[11] Kleiminger, Wilhelm, Thorsten Staake, and Silvia Santini. “Occupancy Detection from Electricity Consumption Data.” Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings, 2013, https://doi.org/10.1145/2528282.2528295.

[12] Weng, Thomas, Bharathan Balaji, Seemanta Dutta, Rajesh Gupta, and Yuvraj Agarwal. “Managing Plug-Loads for Demand Response within Buildings.” Proceedings of the Third ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings - BuildSys ’11, 2011, https://doi.org/10.1145/2434020.2434024.

[13] “California Energy Commission Awards Plug Load Grant to the University.” PRWeb, 30 Mar. 2021, https://www.prweb.com/releases/california_energy_commission_awards_plug_load_grant_to_the_university_of_california_san_diego_johnson_controls_and_bert/prweb17817006.htm.

[14] “EBF 483.” EBF 483 — EBF 483: Introduction to Electricity Markets, https://www.e-education.psu.edu/ebf483/node/508.

[15] Bonomolo, Marina, Marco Beccali, Gaetano Zizzo, Liliana Mineo, Biagio Di Pietra, and Alessandro Buscemi. ”Experimental Set up of Advanced Lighting Systems for Load Shifting Strategies,” 2021 IEEE International Conference on Environment and Electrical Engineering and 2021 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe), 2021, pp. 1-5, doi: 10.1109/EEEIC/ICPSEurope51590.2021.9665407.

[16] Liu, Jingyu, Wen Zhang, and Yutian Liu. ”Primary Frequency Response From the Control of LED Lighting Loads in Commercial Buildings,” in IEEE Transactions on Smart Grid, vol. 8, no. 6, pp. 2880-2889, Nov. 2017, doi: 10.1109/TSG.2016.2632127.

[17] Pipattanasomporn, Manisa, Murat Kuzlu, and Saifur Rahman. ”An Algorithm for Intelligent Home Energy Management and Demand Response Analysis,” in IEEE Transactions on Smart Grid, vol. 3, no. 4, pp. 2166-2173, Dec. 2012, doi: 10.1109/TSG.2012.2201182.

Appendix

Figure 5.1 presents the power consumption patters of 4 different computers in a same office space. This illustrates the idea that devices within a same space can be used a different times of the day and for different lengths of time.

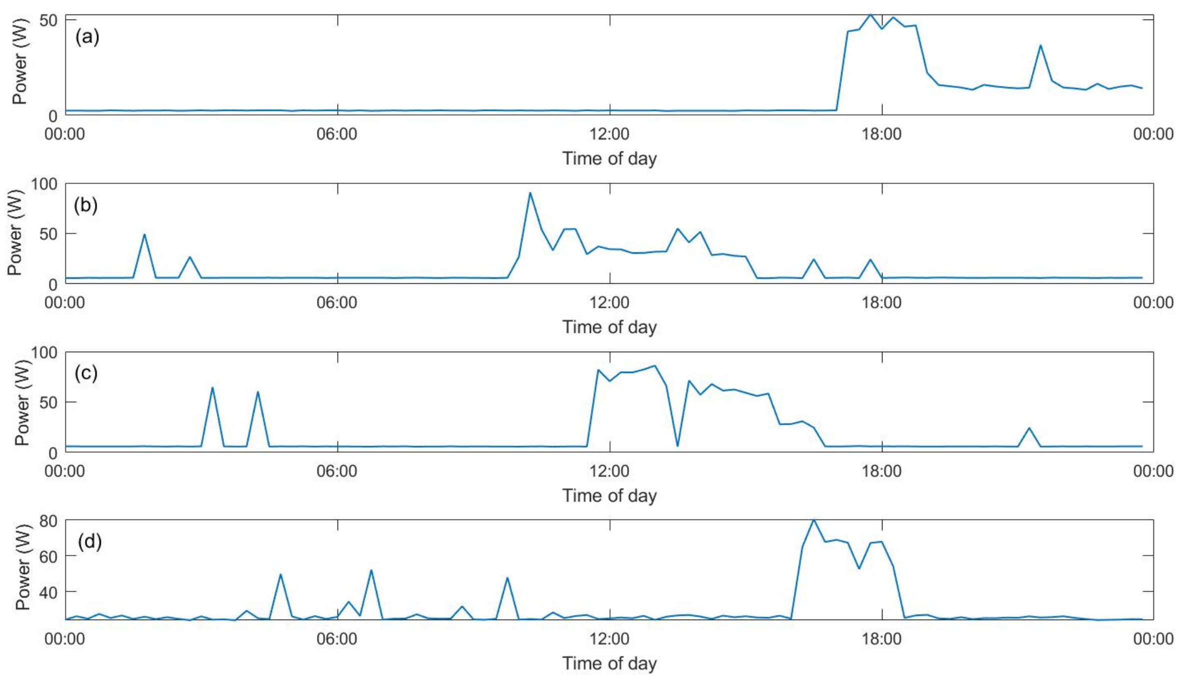


Figure 5.1. Daily energy consumption profile for 4 different computers in the UCSD Students Service Center