Energy-Agile Laptops: Demand Response of Mobile Plug Loads Using Sensor/Actuator Networks

Nathan Murthy[†^], Jay Taneja[‡], Kamil Bojanczyk[⋆], David Auslander[^], and David Culler[‡]

[†]Cloud-based Solutions Innovation Group, Fujitsu Laboratories of America, Sunnyvale, California 94085
[‡]Computer Science Division, University of California, Berkeley, Berkeley, California 94720
[⋆]College of Engineering, Cornell University, Ithaca, New York 14850
[^]Department of Mechanical Engineering, University of California, Berkeley, Berkeley, California 94720

Email: nathan.murthy@us.fujitsu.com, {taneja,culler}@cs.berkeley.edu, ksb37@cornell.edu, dma@me.berkeley.edu

***Abstract*—This paper explores demand response techniques for managing mobile, distributed loads with on-board electrochemical energy storage over a plug-level sensing/actuating wireless mesh network. We target laptops and construct a power consumption and battery charging model from measurements obtained across a variety of such devices. We then build simulations of charging patterns using the most general cases we observed. Our first simulation study explores a *classic demand response* scenario in which a large number of loads participate in a typical pre-scheduled demand response (DR) event. We show that we can achieve load curtailments in the range of 30-90% of aggregate baseline load as a function of the duration of a DR event by managing the charging schedules of laptops that randomly enter and leave the control jurisdiction of a DR event participant. In a second simulation study, we investigate a *continuous demand response* scenario in which charging schedules respond to a fluctuating renewable electricity supply (*e.g.* wind or solar) and show that we can reduce grid dependence by 26.8-33.8% compared to oblivious charging.**

I. Introduction and Background

The motivation for creating demand response (DR) solutions for buildings at the plug-load level comes at a moment when growing interest in building energy science research [1], [2] coincides with increasing maturity in load-monitoring wireless sensor network deployments [3]. Research in built environment sciences has been making significant strides in energy efficiency as well as in DR. In some cases, whole buildings have been transformed into state-of-the art laboratories for experimenting DR strategies and for collecting submetered energy usage data from building-level to plug-level resolution [4]. Progress in large-scale, long-term wireless sensor/actuator networks makes it possible to monitor and manage plug loads in very dynamic building environments [5], [6]. Moreover, microgrid projects around the world in remote areas present emerging challenges to fulfilling the energy needs of whole communities moving towards energy independence or of those who depend almost entirely on intermittent generation [7], [8] as their only source of electricity.

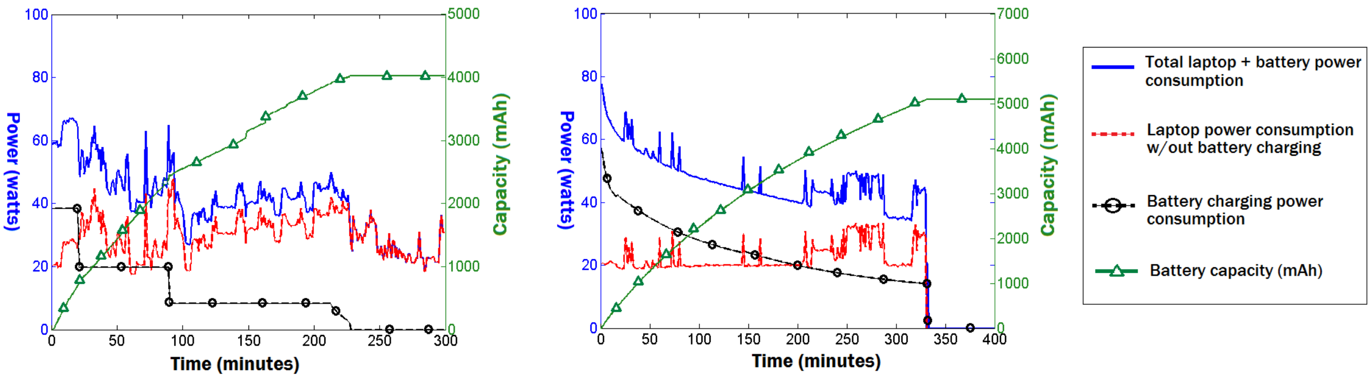
As harbingers of DR-ready design, we investigate laptops in this paper because of their mobile energy storage features and because of their ubiquity in daily life. Energy storage gives them unique load deferment properties; but since laptop battery charging is random and mobile (it is hard to predict when and where someone will charge a laptop from an outlet), managing several of these devices as one aggregate load from a DR perspective poses unique challenges.

The work we present herein has many parallels with the electric vehicle (EV) charging problem. As utilities and grid operators anticipate more EVs hitting markets, the use case of DR for laptops serves as a proxy for the looming question of how to manage EV charging schedules in a DR setting. We build on the existing literature on EV charging [11], [12], [13] and address some of the burgeoning challenges facing consumer groups who are becoming empowered and interconnected through new energy production technologies.

II. Laptop Charging Model

Our models fundamentally rely on understanding the power consumption profiles of laptops as a function of their capacities or states of charge (SoC). It is already well known that as the charge in a lithium-ion battery is depleted, the resistance within the battery increases and therefore the amount of power delivered to a battery in a low SoC is higher than the amount of power delivered to a battery in a high SoC. It is also well known that the charging dynamics and capacity limits of rechargeable batteries vary over their lifetimes depending on the nature of their charge and discharge cycles. Without even introducing vendor-specific laptop designs, these simple facts about laptop batteries alone imply that our problem cuts across a great deal of heterogeneity which makes this problem particularly interesting from a characterization standpoint.

In general, we can classify laptops as either having or lacking charge control mechanisms. These built-in charge controllers, or lack thereof, affect the individual power consumption patterns aggregated over each subcomponent of the said laptop. The power traces in Figure 1 reveal this classification empirically. The graph on the left (Figure 1(a)) depicts power and capacity traces over one full charging cycle of a laptop



(a) MacBook 2006 (b) Dell Latitude P1L Notebook

Fig. 1. On the left we observe one full charging cycle of an Apple MacBook 2006, and on the right we observe a Dell Latitude P1L. The solid blue curve in each graph represents the total power going into each laptop and their batteries. The dashed red curve represents the power delivered to just the laptops, and the circle-dashed black curve is the power delivered to their batteries alone; and thus, the sum of the circle-dashed black and dashed red curves equal the solid blue curve in each figure. The triangle green curves depict the battery capacity of each laptop varying over time as they charge from 0% to 100%.

possessing a charge controller that produces a decreasing stair-step power trace. In Figure 1(b), we depict a charging cycle of a laptop that lacks such a controller, and so we observe, rather, an exponentially decaying power trace profile typical of lithium-ion batteries.

Modeling the total power consumption of laptops with all their various interconnected subcomponents is more complex than modeling just a battery. The dynamic nature of operating systems and software means that the electronic subcomponents on which they run also behave dynamically (as demonstrated by the seemingly stochastic high-frequency variations of the above power traces), and thus the complexity of managing the total power consumption of a large number of these devices is compounded as we include for every laptop a load management policy for each of their subcomponents in addition to their batteries.

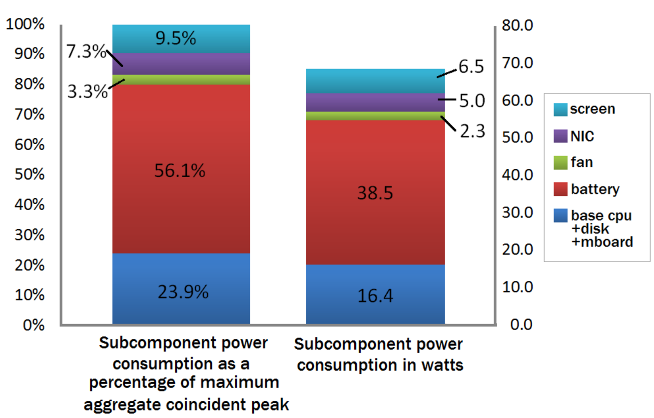


Fig. 2. Coincident subcomponent power disaggregation of an Apple MacBook 2006.

In order to understand how each measurable subcomponent contributes to the electrical power consumption of the laptop as a whole, we use the same laptop that produced the results in Figure 1(a) to estimate which subcomponents contribute to the bulk of this power consumption. Although some software tools [9], [10] provide insights into how application-level activity affects power consumption of software programs, we use empirical methods that reveal how low-level hardware activity, as opposed to application-level activity, affects power consumption. We disaggregate the peak power consumption of the subcomponents in Figure 2 using the same measurement techniques used to isolate the power delivered to a charging MacBook lithium-ion battery. For components such as the network interface card (NIC), LCD screen, and built-in cooling fan, we throttle parameters – NIC bandwidth caps, screen brightness, and fan speed, respectively – to get a clear sense of how much power is delivered to these components from their minimum to maximum operation thresholds. The subcomponent disaggregation figure shows that power delivered to this particular laptop battery in low-capacity states accounts for up to 56.1% of total laptop power consumption during charging.

Given the above considerations and in order to reduce the complexity of the problem, we tailor our algorithms around developing load management policies that manage the charging schedules of only laptop batteries since they account for the bulk of power delivered to laptops when drawing power from an outlet. More importantly, knowing the functional relationship revealed in Figure 1 between the power consumption of the laptop and its battery capacity state allows us to project in advance how much power a laptop is likely to consume given the SoC of its battery even if it is not presently drawing power from an outlet.

III. Problem Formulation

We investigate two model-predictive control problems and develop algorithms which attempt to solve each of these distinct but related problems. In each of the problems, we simulate the scenario of laptops entering and leaving a building. The first control problem, which we call the *classic demand response problem*, involves reducing absolute power consumption from a baseline during a demand response event.

A demand response event (DR event) is a period of time during which the available supply is below the reserve margins set forth by a balancing authority, and so the total load on the grid could exceed supply if action is not taken to curtail demand. With respect to our first scenario, we develop a scheduling algorithm whose objective is to minimize the aggregate load of laptops randomly entering and leaving a building that has been issued a DR event in order to reduce the building's overall peak load during the event.

The second control problem, which we call the *continuous demand response problem*, involves fitting a random load with a random supply over a continuous rather than discrete period of time. Just as in the previous scenario, the random load is the power consumption of laptops entering and leaving a building. But rather than being issued DR events as in the first scenario, this building comes locally equipped with a variable energy resource (e.g. rooftop solar). The solution to this problem relies on techniques that fit load to supply in a manner that optimizes the use of this variable resources both in times of scarcity and excess by minimizing reliance on external power.

Irrespective of either control problem, we wish to have an accurate model of the load we are simulating in each demand response scenario. To achieve this, in addition to building an accurate battery model, we also construct a weighted Poisson arrival and departure process to simulate laptops entering and leaving our control area based on building occupancy inferences from real load profile data.

In both DR scenarios we begin with the initial set $\mathcal{S}_0$ of N laptops at t=0 on the domain [0, T]. For each laptop $\ell_1, .., \ell_i, ..., \ell_N \in \mathcal{S}_0$ we seed their capacity traces so they are randomly chosen to either have one or the other charge profile observed in Figure 1 to introduce some reasonable load heterogeneity in our model and parameterize the battery capacity X_i of the i^{th} laptop as

$$X_i(t+1) = X_i(t) + u_i \mathcal{C}_i(X_i(t))\delta t + (1-u_i)\mathcal{D}_i \delta t \quad (1)$$

where $u_i \in \{1, 0\}$ represents whether the i^{th} laptop battery is charging ($u_i = 1$) at a rate $\mathcal{C}_i(X_i(t))$ (*e.g.* the derivative of either of the capacity traces from Figure 1) or discharging ($u_i = 0$) at a rate $\mathcal{D}_i$ as it transitions from t to $t+1$ with a step size of δt. Unlike the charge rate which is a function of the capacity of the battery, we observe (also from empirical measurements) that the discharge rate of laptop batteries is mostly constant in both high and idle activity states. Furthermore, each laptop in $\mathcal{S}_0$ has the initial power traces

$$u_1 P_1(X_1), ..., u_i P_i(X_i), ..., u_N P_N(X_N).$$

In our models, we initialize the capacity states, and hence the power states, of each laptop so that they are uniformly random on $[0, X_{i,max}]$.

We then introduce a weighted Poisson arrival and departure process to simulate laptops entering and leaving the control area (or zone) with arrival and departure rates defined as

$$\begin{aligned} \lambda_{arrival} &= w_t \lambda \\ \lambda_{departure} &= (1-w_t)\lambda \end{aligned} \quad (2)$$

for a fixed λ whereby $w_t \sim \mathcal{L}_{building}$ and where $\mathcal{L}_{building}$ is the inferred occupancy of the building based on its load profile which roughly provides an estimate of laptops in the zone. In this way, as building occupancy increases, the rate of laptops arriving increases, and the rate of laptops leaving decreases. And conversely, as the occupancy decreases, the rate of laptops arriving decreases, and the rate of laptops leaving increases. So then the set of laptops in subsequent time steps increases or decreases according

$$\mathcal{S}_{t+1} = \mathcal{S}_t + arrivals_{t+1} - departures_{t+1}, \quad (3)$$

and thus over each time step on the domain, the total number of laptops in each set $\mathcal{S}_t$ varies across the chain of step transitions

$$\mathcal{S}_0 \to \cdots \to \mathcal{S}_t \to \cdots \to \mathcal{S}_T$$

as laptops enter and leave the zone in this aleatoric fashion.

A. Classic DR Model

In the classic DR problem, the goal is to reduce peak demand by some percentage of baseline power consumption during a scheduled DR event that lasts for some known duration. We call this percent reduction the *curtailment coefficient* of the DR event which occurs on the interval $[t+k, T-m]$ for $t+k < T-m$. In real world grid operations, the event typically lasts between 3-6 hours.

For this scenario, our model tries to find a combination of laptops $\ell_{1'}, ..., \ell_{k'} \in \mathcal{S}_t$ at each time step t that minimizes the aggregate peak power consumption over the entire event duration. We construct a *time-varying bounded knapsack* algorithm to accomplish this whereby for each $t \in [t+k, T-m]$ we try to find a subset $s_t \subseteq \mathcal{S}_t$ that optimizes the curtailment coefficient c_t thereby minimizing the peak power consumption in the interval.

Since laptop arrivals and departures are random, we can, at best, only try to predict the curtailed load during the demand response event. In real implementations, we would take historical power data of a building with densely deployed networks of laptops paired with outlet sensor-actuators and then perform machine learning operations on the data to extract the patterns needed to make baseline and curtailment load projections. For our purposes, it is sufficient to simulate arrival and departure rates described in (2) to produce load projections. We make a curtailed load projection $\sum_{\ell_i \in s_t}^{s_t} u_i P_i$ on $[t+k, T-m]$ with $u_i \in \vec{\mathbf{u}}_t$ where $\vec{\mathbf{u}}_t$ is a vector whose elements correspond to whether the i^{th} laptop is charging or not. The elements of $\vec{\mathbf{u}}_t$ are assigned either 0 or 1 (in real deployments, entries of $\vec{\mathbf{u}}_t$ correspond to the actuation state of sensor/actuator nodes) by choosing the optimal $c_t \in (0, 1]$ as $t : t+k \to T-m$ by carrying out the following knapsack formulation:

$$\text{objective} : \mathit{maximize}\ c_t \in (0, 1]$$

$$\mathit{minimize}\ \mathit{max} \left\{ \sum_{\ell_i \in s_t}^{s_t} u_i P_i \right\}$$

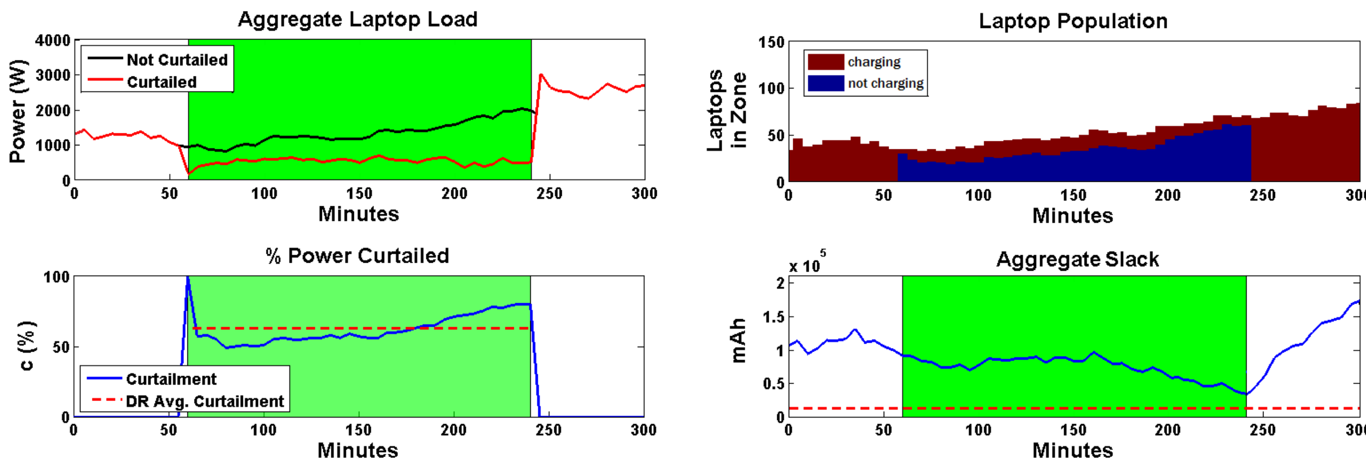


Fig. 3. Simulation run of initially 50 laptops in 5-minute time steps over a total of 5 hours interrupted by a 3-hour DR event (green shade) that begins at $t = 60$min and ends at $t = 240$min. The dashed red line in the slack figure represents the sum of acceptance thresholds of the initial 50 laptops. As the total slack of the load approaches this aggregate acceptance threshold, the deferrability of the entire system approaches zero.

such that

$$\sum_{\ell_i \in s_t}^{s_t} u_i P_i \quad \leq \quad (1 - c_t) \sum_{\ell_i \in \mathcal{S}_t}^{\mathcal{S}_t} P_i \tag{4}$$

Unfortunately, the search space for assigning elements in $\vec{\mathbf{u}}_t$ with this algorithm is very large. To reduce this space, we introduce a method of prioritizing laptops weighted against their power and capacity states which we call a *z-score* defined as

$$z_i(X_i(t), P_i(t)) = \alpha\left(1 - \frac{P_i(t)}{P_{i,max}}\right) + \beta exp\left(-\kappa \frac{X_i(t)}{X_{i,max}}\right), \tag{5}$$

where α, β, and κ are chosen so that the X_i term dominates z_i for small values of $X_i(t)$ (*i.e.* low battery capacity states). In this equation, $P_{i,max}$ and $X_{i,max}$ are the largest observed values of laptop ℓ_i. The z-score represents a function that increases as the SoC of a laptop battery diminishes or as the total power draw of the laptop decreases. The higher the z-score, the higher the laptop's charging priority in the scheduler. But since laptops in low capacity states will also likely draw more power than laptops in higher capacity states, we pick α and β so that $\alpha < \beta$ in order to weight their their capacities over their power consumption levels (κ must also be positive to ensure that z is monotonically decreasing on $[0, 1]$). Therefore, the laptop in the lowest capacity state and lowest consumption level has the highest charging priority, and the laptop with the highest capacity and highest consumption level has the lowest priority.

To shrink the search space, we sort laptops in $\mathcal{S}_t$ according to their z-scores in descending order. We then iterate over this ordered set and pick the ℓ_i's that satisfy the above constraint. We summarize the sort-and-assign procedure as follows:

1) Sort $(\ell_1, ..., \ell_N) \in \mathcal{S}_t$ from $z_{max} \to z_{min}$ which yields an ordered set $(\ell_{1'}, ..., \ell_{N'}) \in \mathcal{S}'_t$
2) Pick $(\ell_{1'}, ..., \ell_{k'}) = s_t \subseteq \mathcal{S}'_t$ that solves the time-varying bounded knapsack problem in (4) by assigning the corresponding elements of $\vec{\mathbf{u}}_t$ to their appropriate boolean values so that each laptop with $u_i = 1$ is allowed to charge (and each with $u_i = 0$ is disallowed from charging) in $\mathcal{S}_{t+1}$

Last, we may prefer a policy that does not allow any laptop battery to die. If this is the case, as in our model, we introduce a capacity acceptance margin $X_{i,accept}$ whereby any laptop ℓ_i that satisfies $X_i(t) \leq X_{i,accept}$ is allowed to charge. By introducing an acceptance threshold, we would not want to have laptops marginally break above $X_{i,accept}$ immediately after time δt of charging only to fall below $X_{i,accept}$ again after discharging for δt in the next time step. To mitigate this, we also introduce a rejection threshold $X_{i,reject}$ that will only disallow laptops from charging once they have reached $X_i(t) \geq X_{i,reject}$. This will prevent oscillations across capacity thresholds. Alternatively, we might have z-score acceptance and rejection thresholds derived from $X_{i,accept}$ and $X_{i,reject}$ so that we can use the same sort-and-assign procedure as above. We implement the latter policy in both the classic and continuous DR control problems in simulations we run.

We show an example of a simulation run of this model in Figure 3 that achieves a more than 60% average reduction of load from the baseline during a 3-hour DR event. We quantify the deferrability of the whole cluster of laptops as a single load during the event by tracking the *energy slack* [6] of the system under control in the bottom-right chart of the figure.

B. Continuous DR Model

In the continuous DR scenario we try to fit a time-varying random load to a random renewable supply. Unlike the classic DR problem where the goal is to reduce peak power consumption over the duration of a DR event, the continuous scenario involves matching load with supply in times of excess and scarcity over the entire domain $[0, T]$ during which some intermittent resource is generating electricity albeit variably.

The classic DR model assumes that all power delivered to each load originates from the grid. However, in the continuous model we assume that the intermittent supply is at the site of consumption with the grid available as an external resource. The goal, then, is to minimize dependence on any external or supplemental power supply.

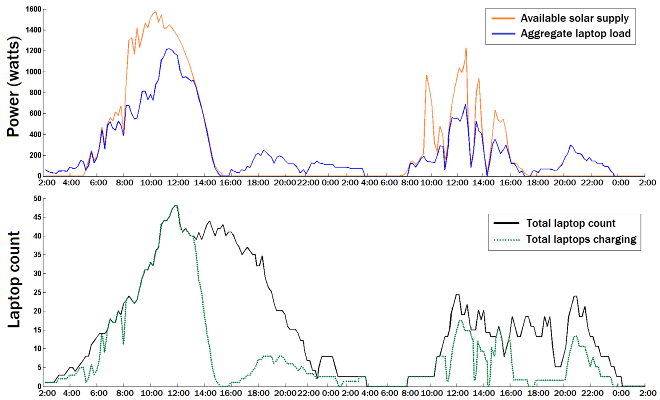


Fig. 4. Simulation run of continuous DR over two days in 15 minute time steps.

We construct the continuous model using the same techniques as the classic model with a nearly identical problem formulation. In this time-varying bounded knapsack algorithm, there exists only one objective function, namely, the absolute error of aggregate load and total supply over each time step $t \in [0, T]$ under the constraint that load does not exceed supply. The formal problem statement is

$$\text{objective}: \; minimize \Big| \sum_{\ell_i \in s_t}^{s_t} u_i P_i - \hat{P}_{supply}(t) \Big|$$

$$\text{such that} \sum_{\ell_i \in s_t}^{s_t} u_i P_i \; \leq \; \hat{P}_{supply}(t) \qquad (6)$$

where $u_i \in \vec{\mathbf{u}}_t$ and $\hat{P}_{supply}(t)$ is the predicted supply of power at time t. In this way, part of the performance of this supply-following algorithm depends on the accuracy of the predictor. We quantify the error between load and supply as the *grid dependence* which is defined on a real-valued scale of [0,1] with 0 being no dependence on external power whatsoever and 1 being total dependence.

The continuous scenario employs the same sort-and-assign procedure to populate the entries of $\vec{\mathbf{u}}_t$ by using the z-score of each laptop to prioritize them for charge scheduling in each time step but with only a single objective function to fulfill as opposed to two objective functions in the first scenario. When no renewable power supply is available (*e.g.* night time in the case of solar), the z-score acceptance and rejection thresholds are put into effect to ensure that no laptop dies at the expense of grid independence.

IV. Results from Simulation Studies

We use the same z-score function (with $\alpha = 0.25$, $\beta = 0.75$, and $\kappa = 3.0$) to prioritize laptop charging schedules for both the classic and continuous DR simulations. A key metric we test in the classic DR scenario is how well our curtailment algorithm performs over varying DR event durations since the available aggregate energy slack of our loads towards the end of every DR event approaches zero as the battery capacity of each laptop depletes (recall Figure 3). Less slack implies less deferrability and therefore less curtailment feasibility. We summarize our trials in Figure 5 in order to demonstrate this relationship.

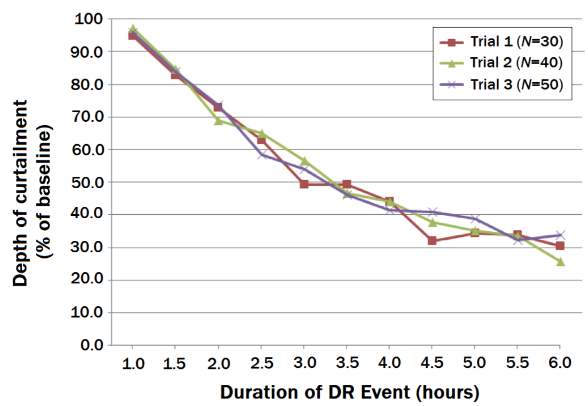


Fig. 5. Depth of curtailment as percent of baseline load with respect to DR event duration.

To produce the results above, we executed a total of 330 simulation runs of the classic DR model in three separate trials. We adjusted the length of the DR event from 1 to 6 hours in half-hour increments and did this 10 times in each trial for a total of 110 runs per trial. In each trial we picked large values for N_0 (N_0 = 30, 40, and 50 for trials 1, 2, and 3, respectively) since DR events typically occur in times of high demand. And so each curve in Figure 5 is the mean curtailment ratio over 110 runs in each of these trials. We see that the depth of curtailment over varying DR event durations provides a proxy for quantifying the deferrability of the loads we are modeling without directly knowing the energy slack of these loads. Therefore, the deferrability of the system in this scenario varies indirectly with the length of a DR event.

We then assess the performance of our algorithm tailored to the continuous DR scenario using public solar PV data from rooftops across 12 locations around the United States for 5 days of summer (7/8/2010-7/12/2010) and 5 days of winter (12/19/2010-12/23/2010) for a total of 120 day-location pairs of solar power traces [14]. Each day-location corresponds to a unique simulation run, all of which are then used to build performance trials based on different supply prediction techniques.

In these trials we wish to determine how much reliance on external power supply (perhaps either in the form of utility-provided electricity, on-site energy storage, or backup generation) is needed to buffer demand as the renewable supply fluctuates. To quantify this fluctuation, we invoke the notion of *scaled incremental mean volatility* (scaled IMV) [15] which is simply the absolute difference between the moving average of the supply signal and the actual observed supply, averaged over the time domain of interest and scaled according the ratio of total energy demanded over total energy supplied in that domain. The scaled IMV gives us insights into the high-frequency fluctuations of our supply signal which could be the result of sporadic cloud cover (in the case of solar) or microbursts (in the case of wind) without bias to energy imbalances in supply and demand.

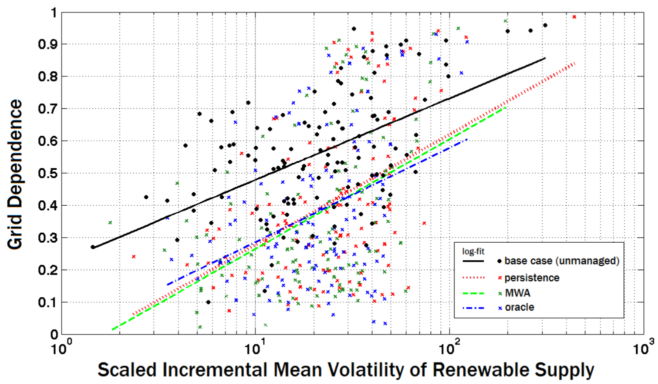


Fig. 6. Semilogarithmic scatter plot of grid dependence as a function of the scaled IMV of the supply signal for a specific day (*i.e.* simulation run) of all three supply prediction trials compared to the unmanaged base case trial without any charge scheduling algorithms applied.

In Figure 6 we present the results of four distinct trials corresponding to one unmanaged base case scenario in which our algorithm is not applied, and three applied scenarios employing three distinct supply prediction models for performance comparison. The base case (solid black) assumes loads obliviously consume power regardless of the renewable supply, in which case we do not apply any load control. In one algorithm-applied trial we use a persistence model (dotted red) in which we assume that the power supply in the current time step will be the same in the next time step; in the next trial we use a 1-hour moving window average (dashed green) of historical data to predict supply in the next time step; and in the final trial we use an oracle (dash-dot blue) in which we know exactly how much solar will be available in every time step. The average base case grid dependence of oblivious charging is 0.561. The average grid dependencies over each algorithm trial of 0.411, 0.380, and 0.371, respectively, reflect a 26.8-33.8% improvement to the base case when our sort-and-assign knapsack algorithms are applied to laptop charging schedules.

V. Conclusion

These results demonstrate that it is possible to leverage sensor/actuator networks to manage in real-time a large number of distributed battery-powered mobile devices for a classic DR scenario in a way that could meet the curtailment objectives of a demand response program participant, or for a continuous DR scenario that optimizes available variable energy resources like wind or solar by reducing dependence on an external power grid. By examining a single category of devices ubiquitous and dispersed within modern buildings and elsewhere, we develop strategies that treat all of these devices as one load in the context of an energy management system (EMS) and anticipate similar DR approaches to the plug-in electric and hybrid vehicle problem. In a real-world EMS, our online algorithm could work in conjunction with a whole suite of load management controls and algorithms for buildings or microgrids.

Acknowledgements

This work is supported by the U.S. Department of Energy (DI-ADR DE-EE0003847) and National Science Foundation (CPS:LoCal CNS-0932209 and CPS:ActionWebs CNS-0931843). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Department of Energy or National Science Foundation.

References

[1] D.B. Crawley, L.K. Lawrie, F.C. Winkelmann, et al., "EnergyPlus: creating a new-generation building energy simulation program," *Energy and Buildings*, Elsevier, Vol.22, Issue 4, pp. 319-331, April 2001.

[2] Y. Lu, P. Mukka, T. Gruenwald, "Building Management System and OpenADR Integration," *Distributed Intelligent Automated Demand Response (DIADR)*, Siemens Corporate Research, Nov 2010.

[3] X. Jiang, M.V. Ly, J. Taneja, et al., "Experiences with a High-Fidelity Wireless Building Energy Auditing Network," *SenSys '09*, ACM, Nov 2009.

[4] D.B. Arnold, N.A Murthy, M. Sankur, et al., "DIADR Local Control Testing in a Laboratory Environment," *A Distributed Intelligent Automated Demand Response Buliding Management System*, DOE Award Number: DE-EE0003847, University of California, Berkeley, Sept 2011.

[5] S. Dawson-Haggerty, S. Lanzisera, J. Taneja, et al., "@scale: Insights from a Large, Long-Lived Applicance Energy WSN," *Proceedings of the 11th ACME/IEEE Conference on Information Processing in Sensor Networks*, SPOTS Track (IPSN/SPOTS '12), April 2012.

[6] J. Taneja, D. Culler, P. Dutta, "Towards Cooperative Grids: Sensor/Actuator Networks for Renewables Integration." *IEEE SmartGridComm '10*, Oct 2010.

[7] I. Schinca, I. Amigo, "Using renewable energy to include off-grid rural schools into the national equity project Plan Ceibal," *International Conference on Biosciences*, pp. 130-134, March 2010.

[8] D. Soto, E. Adkins, M. Basinger, et al., "A prepaid architecture for solar electricity delivery in rural areas," *ICTD '12*, pp. 130-138, March 2012.

[9] J. Flinn and M. Satyanarayanan, "PowerScope: A Tool for Profiling the Energy Usage of Mobile Applications," *IEEE Workshop on Mobile Computing Systems and Applications*, pp.2-10, Feb 1999.

[10] L. Zhang, B. Tiwana, Z. Qian, et al., "Accurate online power estimation and automatic battery behavior based power model generation for smartphones," *CODES/ISSS '10*, pp.105-114. Oct 2010.

[11] A. Brooks, E. Lu, D. Reicher, et al., "Demand Dispatch: Using Real-Time Control of Demand to Help Balance Generation and Load," *IEEE Power and Energy Magazine*, Vol.8, Issue 3, pp.20-29, May 2010.

[12] J. Lee, A Varuttamaseni, F. Rahman, "Optimization of the Integrated Electric Grid," *DTE Energy MPSC PHEV Pilot Project*, University of Michigan, Ann Arbor, January 2011.

[13] S. Studli, E. Crisostomi, et al., "AIMD-like algorithms for charging electric and plug-in hybrid vehicles," *IEEE Electric Vehicle Conference*, March 2012.

[14] Rooftop solar traces available from http://view2.fatspaniel.net/SSH/MainView.jsp

[15] M. Roozbehani, M. Dahleh, S. Mitter, "Volatility of Power Grids under Real-Time Pricing," *LIDS Report*, MIT. June 2011.