The Benefits of Dynamically Resizing Residential Storage

Jan Deller
University of Cambridge
Cambridge, United Kingdom
jfd37@cam.ac.uk

Srinivasan Keshav
University of Cambridge
Cambridge, United Kingdom
sk818@cam.ac.uk

ABSTRACT

Today's solar and storage systems are typically sized at the time of deployment to match historical loads. However, loads do change over time, which may make one-shot sizing sub-optimal. Modular storage systems allow storage capacity to change over time to better match changing loads. We investigate how a modular battery system, in combination with a solar photovoltaic (PV) in a grid-connected system, can be dynamically resized to reduce overall costs. We develop an algorithm for dynamic resizing and evaluate it in multiple scenarios using both real and synthetic traces of solar and load. We find that our algorithm is able to closely approximate the offline optimal algorithm. Moreover, in our case study, we find that overall costs are not significantly reduced due to resizing of storage, and this conclusion is insensitive to types of load changes and the choice of resizing interval.

CCS CONCEPTS

• **Hardware** → *Batteries*.

KEYWORDS

Energy storage, dynamic sizing, renewable energy

1 INTRODUCTION

Rapidly declining prices of PV systems and storage have made them increasingly popular, even in residential settings. By adding storage (also called battery in the remainder of this paper), PV systems can supply energy on cloudy days or at night, allowing home-owners to reduce their dependence on the electrical grid. However, due to substantial capital costs for PVs and batteries, their sizes need to be chosen carefully.

Typically, PV and storage are sized at the time of installation based on historical traces of load and solar generation [6]. If the load deviates from its historical patterns, for example due to the purchase of an electric vehicle (EV), the addition or removal of a pool pump, or the addition or departure of a family member, this sizing may be incorrect. Unfortunately, residential PVs are usually

roof-mounted and cannot be easily accessed to allow a change in its size. In contrast, a modular battery system, which consists of a set of bays into which storage modules can be self-installed by the owner, much like a hard drive, is easy to reconfigure [3]. Such a system gives the owner the opportunity to dynamically change its size to reflect changes in load patterns.

It seems intuitively obvious that dynamically changing the battery size to match the load would prevent oversizing or undersizing at the time of initial deployment, reducing overall costs. To validate the correctness of this insight, this paper investigates how a modular storage system's size can be dynamically adapted to changes in the load or generated electricity to minimize total costs, which are the sum of PV costs, battery costs, and unmet energy costs. We propose a stochastic model-predictive control (MPC) approach to the resizing algorithm that reduces total costs, and evaluate how well this algorithm performs. We also study how often resizing should take place to balance between cost minimization and responsiveness to changes in load.

Note that we assume that the PV system is grid-connected, so that load not met from this system is purchased from the grid. For simplicity, we assume that this price is constant and not subject to time-of-use pricing; we also assume that the grid does not pay for an power transferred to it. Our approach could be adapted to model more complex scenarios with appropriate parameter changes. Note also that the PV size is difficult to change, so in this work it is regarded as fixed, and the cost of PV is modeled as a sunk cost. Finally, we size the storage only for energy, not for power, since a typical 10 kWh residential store can provide nearly 100A at 110A, which meets most domestic power requirements.

We make three key contributions:

- We present an algorithm based on model-predictive control to periodically resize the storage system.
- We show that our algorithm (slightly) reduces total costs compared to a static sizing and that its performance is very close to that of an offline optimal algorithm.
- However, contrary to our expectations, in our case study, the reduction in costs from resizing is not significant. Moreover, this conclusion is robust to changes in load type and resizing interval.

The organization of the rest of the paper is as follows. Section 2 provides an overview of related research, before section 3 introduces the approach taken to resize the battery. Section 4 evaluates this approach and answers questions like whether or not dynamic resizing can reduce the total costs of a PV/battery system and how often the resizing should take place. Finally, section 5 concludes this paper.

2 RELATED WORK

This section outlines related work on PV and storage sizing as well as work on synthetic trace generation.

Sizing solar and storage systems is an important practical problem that has been well studied in the literature, as recently surveyed in Reference [5]. Therefore, in the interests of space, we focus here only on the most closely related work.

Kazhamiaka *et al* [6] present and evaluate three different robust sizing approaches: mathematical programming, simulation, and an analytical method based on the stochastic network calculus. Load and solar traces are used to express a residence's electricity demand and solar activity that can be used for electricity generation, respectively. Their goal is to minimize the costs of the PV and battery while guaranteeing a specified Quality of Service (QoS), which is either measured as loss of load probability or expected unserved energy for a specified period.

The present work differs from this work in three aspects. First, we allow storage to be resized. Specifically, the PV size is assumed to be fixed and we explore how costs might be reduced by storage owners periodically self-resizing their storage.

Second, their goal of minimizing PV and battery costs that guarantee a certain QoS makes sense in an off-grid situation. We assume that the residence is grid-connected, so the cost minimization is not subject to a QoS constraint. Instead we explicitly model the costs of unserved energy, i.e., additional electricity costs with a fixed-cost tariff. Thus, the overall costs are minimized by choosing a sensible trade-off between higher investments into the battery and a higher level of unserved energy.

Finally, Kazhamiaka *et al* use a Chebyshev bound to compute robust sizings of PV and storage. They need to do so because of the considerable degree of uncertainty in future loads. In our work, because we allow resizing, each resizing choice itself need not be overly conservative, allowing us to (mostly) dispense with the use of this bound (the exception is that we use the bound for the initial sizing, as discussed in more detail below).

The second strand of related work is concerned with synthetic trace generation. As historical data that can be used for the sizing of a system is often rare, synthetic traces can be used to generate more traces from the same distribution. Sun *et al* [8] evaluate how well data generated by different generative models can be used to find a sizing that meets a predefined QoS for a PV/storage system. The goal is to make the system robust to future variations in generation or load, which are unknown when the sizing decision is made. The ARMA model was found to be the most effective. Given the uncertainty of future traces, an ARMA model can also be used in a dynamic resizing approach to model what future traces might look like, to make the sizing to a certain extend robust to this uncertainty. We discuss this in more detail later in this paper.

3 DYNAMIC RESIZING APPROACH

As discussed, prior work demonstrates how to size storage and PV given historical traces of solar generation and load [6]. We use this approach to compute the initial PV size, using the simulation approach discussed there and one year of prior load and solar data, which is assumed to exist. We then periodically re-calculate the cost-minimizing battery size. This section describes how the cost-minimizing battery size is computed.

Our approach mimics the control procedures used in stochastic Model Predictive Control [4]. Specifically, we periodically minimize the total cost:

$$\underset{B}{\operatorname{argmin}} \, c_{bat} + \mathbb{E}(c_{ue}) \qquad (1)$$

where B represents the size of the storage, the battery cost is c_{bat}, and the expected cost of unserved energy, which must be obtained from the grid, is $\mathbb{E}(c_{ue})$. Both costs are evaluated from the time of resizing to the end of the lifetime of the battery[1], as discussed in more detail below. While the battery costs are certain, the expected unserved energy depends on the future behaviour of solar and load and is therefore uncertain.

We now modify Equation 1 to better model resizing costs. Specifically, it will be convenient to compute the cost of a *change* in battery size, denoted ΔB, rather than the battery size itself. Also, we include the labour cost of changing the battery size by ΔB, denoted $c_{inst}(\Delta B)$ (this cost is expected to be realatively low because the installation would be by the owner of the storage).

We thus calculate the cost-minimizing change in battery size ΔB^* at time t in days from the start of the system deployment as:

$$\Delta B^*(t) = \underset{\Delta B}{\operatorname{argmin}} \, c_{bat}(t, \Delta B) + c_{inst}(\Delta B) + \mathbb{E}(c_{ue}(t, \Delta B)) \qquad (2)$$

The term t in this equation emphasizes that the time span over which the costs are calculated is from the time of resizing t to the end of the lifetime of the initial battery. The lifetime of the initial battery in days is denoted as T.

We make the simplifying assumption that modules added to the storage system have the same lifetime as the remainder of the system[2]. We also assume linear depreciation in module price. The battery costs can then be computed with either a buying or a selling price for modules, depending on the sign of ΔB:

$$c_{bat}(t, \Delta B) = \left\{ \begin{array}{ll} \pi_B^{buy}(t) * \Delta B = \pi_B^{buy} * \frac{T-t}{T} * \Delta B, & \text{for } \Delta B \geq 0 \\ \pi_B^{sell}(t) * \Delta B = \pi_B^{sell} * \frac{T-t}{T} * \Delta B, & \text{for } \Delta B < 0 \end{array} \right\}$$

where π_B^{buy} is the price for a battery module with lifetime T, and thus $\pi_B^{buy} * \frac{T-t}{T}$ is the price for a battery module with remaining lifetime $T - t$. Respectively, π_B^{sell} is the price for selling such a module.

The installation costs can be computed according to

$$c_{inst}(\Delta B) = \left\{ \begin{array}{ll} \pi_{inst}, & \text{for } \Delta B \neq 0 \\ 0, & \text{for } \Delta B = 0 \end{array} \right\}$$

with the penalty for changing the battery size π_{inst}. The installation costs are not dependent on t or ΔB.

The final component from Equation 2 to be calculated is $\mathbb{E}(c_{ue})$, the expected cost of unserved energy. This, of course, depends on the future solar generation and load, which is unknown. We make the pragmatic assumption that the load trace from the year before the resizing time can be used to compute the expected load

---

[1]Battery lifetime depends both on calendar age, about 8-10 years, as well as the total energy flux through the store. In our work, we conservatively choose a maximum calendar age of 8 years and, for simplicity, ignore aging due to energy flux.
[2]This models the fact that the system owner has no motivation to purchase a module whose lifetime exceeds that of the remainder of the system.

trace for the next year. An entire year of data is used to ensure that the seasonal component is captured[3]. Specifically, the ARMA model from [8] is used to generate ten one-year load traces based on the last year's load trace. Interestingly, we found that using the same approach to generate solar traces (i.e., using the past year to generate ARMA traces for the next year) decreased the accuracy of the resizing. Thus, in this work, the actual solar data from the last year is used for computing the resizing, rather than ARMA-generated traces. The resulting ten one-year load and solar trace pairs are then simulated for different battery size changes ΔB, and the mean unserved energy $UE_{mean,\Delta B}$ is noted for each ΔB. Then, $\mathbb{E}(c_{ue}(t, \Delta B))$ is calculated for the remainder of the time period T using the price of electricity per kWh in USD π_{UE} as:

$$\mathbb{E}(c_{ue}(t, \Delta B)) = \frac{UE_{mean,\Delta B}}{365} * (T - t) * \pi_{UE}$$

Thus, our algorithm is the following:

(1) Use the past year's load and solar trace to compute the initial PV sizing.
(2) At each resizing point, carry out steps 3-7.
(3) Use the prior year's (known) load to compute optimal ARMA parameters.
(4) Use the ARMA model to generate ten traces for the next year's expected load.
(5) Use the prior year's solar trace to form ten one-year load and solar trace pairs.
(6) For each possible battery size change, simulate the ten trace pairs to compute the unserved energy *for the next year*. Use this to calculate the extrapolated battery costs and expected unserved energy costs for the *remainder of the lifetime of the battery*, i.e., 8 years.
(7) Determine the battery size change that minimizes the total cost in Equation 2.

Note that the sizing decision models the trade-off between battery costs and unserved energy costs.

Also, note that the choice of resizing interval balances two costs. Resizing too often can lead to higher installation costs, whereas resizing too seldom can also lead to a cost increase as it takes longer to react to changes in load. The choice of resizing interval that minimizes the total cost is discussed in Section 4.

Finally, note that this algorithm can be viewed as a stochastic Model-Predictive Controller, where the ARMA-derived load traces constitute the set of forecasts, the remaining lifetime of the battery is the control horizon, and the resizing interval is the control period. The re-optimization of the expected future cost over the control horizon by tuning the change in battery size then maps directly to the stochastic MPC paradigm [4].

4 NUMERICAL EVALUATION

In this section, we numerically evaluate our algorithm in the context of a case study based on the Pecan Street Project in Austin, TX [1]. Specifically, we choose solar and load traces from seven homes in this project and compute the gains from resizing storage in response to changes in the load. We begin by describing the parameters used in our evaluation.

4.1 Parameters used in the evaluation

4.1.1 Battery parameters. We used battery model 1* from Reference [7]. As in [6], the battery's size is expressed in number of cells, where one cell holds 0.011 kWh, and its parameters were taken from Reference [6]. The battery buying price is also taken from Reference [6]. The battery selling price parameter expresses the selling price relative to the buying price: 0.9 means that the selling price is $\pi_B^{sell} = 0.9 * \pi_B^{sell}$. These parameters can be found in Table 1.

Table 1: Battery parameters used for the evaluation

Cell size (kWh)	0.011
Battery buying price (USD/kWh)	500
Battery selling price parameter	0.9
Battery lifetime (years)	8
Installation costs (USD)	15

4.1.2 Other parameters. We assume that the installed cost of a PV system is $2,000/kW [4]. As the battery lifetime is set to 8 years and PVs frequently offer a warranty for 25 years, this PV price is depreciated by a $\frac{16}{25}$ factor, for a resulting cost of $640/kW.

We did not choose the cost of unserved energy using electricity prices from the US, since the price there, which is around 0.12 USD per kWh, would have led the optimal battery size to be 0, as it would not be worth investing in a battery in the first place. Instead, unserved energy costs are set to 0.3 USD per kWh, which is typical for many countries (for example, the cost of energy for households in Germany is 0.31 USD [2]). A higher value leads to a larger battery, and with a cost around 0.3 USD per kWh, the battery is reasonably sized. This enables us to evaluate how well the resizing algorithm works.

For the initial PV sizing, the simulation approach from Reference [6] was used with the target level of expected unmet energy of 0.05 and confidence level 0.95.

We obtained solar and load traces from the Pecan Street Project [1]. For all simulations, the length of the solar and load trace to be simulated was the lifetime of a battery, i.e., 8 years. Additionally, one more year was used at the start of the traces for the initial sizing.

Finally, although only load changes are analyzed, we also tested our resizing algorithm assuming changes in solar generation, and the results are essentially identical to that with changes in load, and thus are not presented here.

The rest of this section is structured as follows. First, it is investigated how well the algorithm computes the battery size. Then, we study how well it reacts to changes in the load trace. Finally, we look at the impact of resizing interval on total costs.

---

[3]For more information on seasonality and the impact of non-stationarity in load traces, see Section 4 in Reference [6].

[4]This is midway between the cost in the US ($3000/kW) https://news.energysage.com/how-much-does-the-average-solar-panel-installation-cost-in-the-u-s/ and the cost in Germany ($1000/kw) https://www.pv-magazine.com/2018/03/20/pv-has-the-lowest-lcoe-in-germany-finds-fraunhofer-ise/

Table 2: Other parameters

PV price (USD/kW)	640
Unserved energy costs (USD per kWh)	0.30
Length of evaluation (years)	8

4.2 How well does our algorithm compute the optimal battery size with one-shot sizing?

Although one-shot sizing is not the focus of our work, we first investigate the quality of our algorithm in this simple setting. We selected seven different year-long trace pairs from different houses in the Pecan Street data set [1]. Since we did not have traces of length 8 years, we selected one year of traces and concatenated these to produce 8-year long traces. For each house, the battery size computed by one run of our sizing algorithm (i.e., steps 3-7) that only knows the past one-year's history is compared to the optimal battery size computed by an oracle that knows the future load profile. In addition to the chosen battery size, the total costs corresponding to the chosen battery size are also compared.

Table 3: Battery size in cells

Algorithm: Size	Oracle: Size	% Difference
2450	2300	6.5%
2450	2300	6.5%
4650	4700	-1.1%
5500	5500	0%
5700	5750	-0.9%
2250	2200	2.3%

Table 3 shows the chosen and optimal battery size in cells, and Table 4 compares the sum of battery and unserved energy costs. Each row represents one of the seven houses. We see that the optimal size as computed by the sizing algorithm is quite close to the oracle-based optimal size. In the second part of this question, we will discuss what effect these sub-optimal (slightly oversized) values have on the total costs.

Table 4: Battery and unserved energy costs with algorithm's optimal size and oracle-based optimal size, in USD

Algorithm Cost	Oracle Cost	Cost Difference	Cost Difference (%)
$23,463	$23,450	13.1	0.06%
$26,704	$26,681	23.4	0.09%
$70,129	$70,125	4	0.006%
$59,497	$59,497	0	0.0%
$74,114	$74,113	0.7	0.001%
$23,577	$23,575	1.4	0.006%
$24,963	$24,938	25.1	0.1%

Table 4 shows the sum of battery and unserved energy costs for an eight-year simulation of the seven trace pairs for the battery size as determined by the sizing algorithm and the optimal oracle-based battery size. Additionally, it shows the cost differences. We find

that even when the battery size is not optimal, its total costs are very close to the optimal total costs. The maximum cost increase is close to 0.1 percent, and in all but two cases, it is significantly below 0.1 percent.

Given the general uncertainty about the future load, we conclude that the sizing computed by our algorithm is close to optimal in a one-shot setting.

4.3 Does resizing reduce costs?

We now compare the total costs as computed by the resizing algorithm to the total costs if no resizing takes place and the battery keeps its initial size.

We first generated synthetic traces with four different change patterns, which are meant to model charging of an electric vehicle. As before, we concatenated traces to create 8-year traces. Additionally, for each trace, the change was applied after the first two years of the simulation. The first two traces are subject to a load increase by 8 kWh per day, which is the amount an EV would typically charge in one day, assuming that it drives 46km/day[5] and has an efficiency of 16 kWh/100km [6], which are typical today. In trace 1, this is a night-only increase, leading to 8 hours per night with an increase by 1 kWh. This models an EV charging overnight. In trace 2, the increase is equally distributed across the entire day, with an hourly increase of $\frac{1}{3}$ kWh. This models an EV charging throughout the day. The last two traces are subject to a load decrease by the same amount, where the changes are applied in the same fashion for trace 3 as for trace 1 and trace 4 as for trace 2. This models the sale of an EV. We ensured that even when the load decreased, it never became negative, and that the cumulative load change stayed the same.

Table 5: Total costs with and without battery resizing, in USD. Trace 1: night load increase, trace 2: all-day load increase, trace 3: night load decrease, trace 4: all-day load decrease

Trace	$c_{dynamic}^{52}$	$c_{dynamic}^{8}$	c_{static}
1	$56,132	$56,094	$56,469
2	$54,748	$54,738	$54,809
3	$49,232	$49,264	$49,518
4	$50,419	$50,431	$50,569

Table 5 shows the total costs, i.e. PV costs, battery costs, and unserved energy costs, of the resizing intervals 52 weeks ($c_{dynamic}^{52}$) and 8 weeks ($c_{dynamic}^{8}$) as well as those without resizing c_{static}. For the no resizing option, the battery keeps the size that is initially computed to be optimal throughout the simulation. While the costs are lower in all four traces with the resizing of the battery, the savings are relatively small, e.g. smaller than 0.7 percent for the 8-week resizer.

The saving is significantly larger for the traces 1 and 3, which had nightly changes. This change pattern has a stronger effect on the optimal battery size than the change pattern used for traces 2

---

[5]The typical driver in Texas drives 10,500 miles/year, which is about 46km per working day https://www.carinsurance.com/Articles/average-miles-driven-per-year-by-state.aspx .
[6]https://pushevs.com/electric-car-range-efficiency-epa/

and 4. For trace 1 for example, the optimal battery size increases from 2450 cells to 3050 cells after the change, whereas it is 2800 cells for trace 2. Therefore, the costs that can are saved by resizing are larger for these two traces.

Taking into account that the PV costs of 28928 USD cannot be changed, the relative saving increases, peaking at around 1.4 percent for the 8-week resizer. While it does make sense to resize the battery as it reduces the total costs, the cost difference is surprisingly small. This is because the role of storage in our setting is essentially only to displace the cost of unmet load. If this cost is relatively low, due to the low price of grid electricity, then modifying storage size only marginally affects the overall cost.

4.4 How close is our resizing algorithm to the offline optimal?

We now compare total costs incurred by our resizing algorithm to that of an an oracle-based resizer (c_{opt}) to see how close our algorithm is to optimal. This allows us to investigate if the limited improvement in performance is due to our resizing algorithm or due to a limited possible savings even in the optimal case.

Table 6: Total costs with resizing and with oracle-based resizing, in USD. Trace numbers as in Table 5

Trace	$c^8_{dynamic}$	c_{opt}	c_{static} $-c_{opt}$	c_{static} $-c^8_{dynamic}$	Ratio (%)
1	$56,094	$56,055	$414	$375	90%
2	$54,738	$54,713	$97	$71	73%
3	$49,264	$48,790	$729	$254	35%
4	$50,430	$50,168	$401	$138	34%

We compute the costs for the optimal battery sizes before the change in the load pattern and after the change in the load pattern, c_{before} and c_{after}, respectively. The optimal costs can be computed as $c_{opt} = \frac{2}{8} * c_{before} + \frac{6}{8} * c_{after} + 15$, as the installation costs also have to be taken into account. Table 6 shows the results.

The second column shows the cost achieved when dynamically resizing storage every 8 weeks and the third column shows the optimal cost. The maximum achievable cost reduction, which is the difference between the static costs c_{static} and the optimal costs c_{opt}, is shown in the fourth column. The fifth column displays the actually achieved cost reduction, i.e. the static costs c_{static} minus the costs of the dynamic resizing algorithm $c^8_{dynamic}$, and the last column is the ratio of the reduction in costs from our algorithm to the optimal.

The table provides three main insights. First, the fourth column shows that the amount of money that can be saved by resizing, even in the optimal case, is rather limited. It amounts to only about $100 to $700 for a system that costs several tens of thousands of dollars. The takeaway point is that resizing helps, but not by much.

Second, given that c_{opt} is an unachievable lower bound, the dynamic resizing algorithm's performance is quite good. On average, more than 55 percent of the achievable cost reduction was achieved, with a maximum of 90 percent for the first trace.

Finally, the resizing algorithm performs better when the optimal battery size increases due to an increase in load. On average, more

than 80 percent of the maximum cost reduction is achieved when the optimal battery size increases, as opposed to an average of only around 35 percent when the optimal battery size decreases. This is mostly likely due to the slight oversizing of the battery by our algorithm, as shown in Table 3. If the initial size is a little too large, it matches load changes that lead to an increasing battery size, i.e., traces 1 and 2, better than changes that lead to a decreasing battery size, i.e., traces 3 and 4.
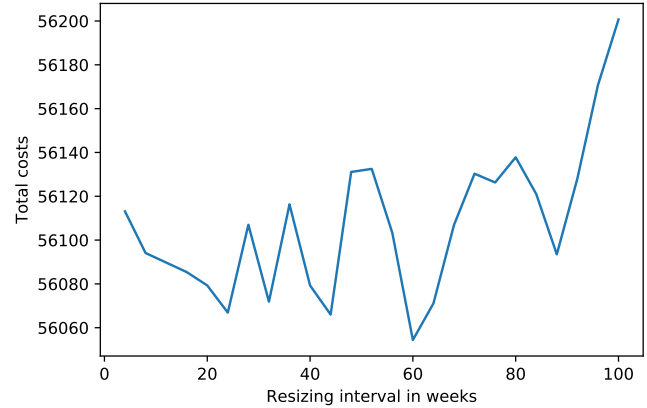
4.5 What is the optimal resizing interval?

Figure 1: Total costs of different resizing intervals

Figure 1 shows the total costs for trace 1 from the previous question for different resizing intervals. Three features are apparent, as discussed next.

The first insight is that the graph is spiky instead of smooth. Table 2 shows how the battery sizes are adapted in the simulation period for various different resizing intervals. The vertical red line at week 104 signifies that the change happened at this point. We
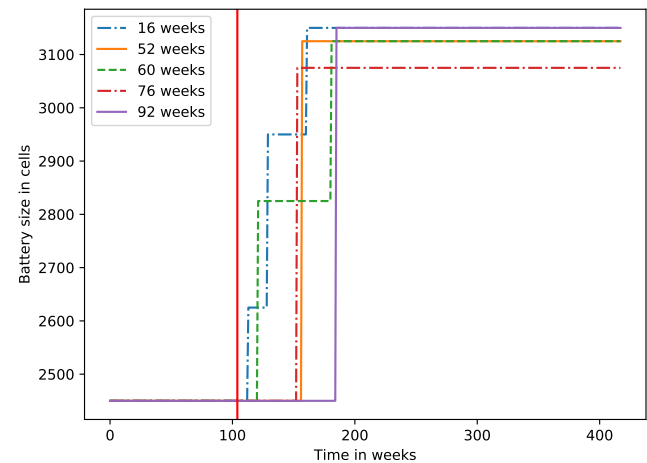

Figure 2: Battery sizes over simulation period for different resizing intervals

can see that depending on when the resizing happens, the chosen

battery size is different. The greater the gap between week 104 and the time of resizing, the more of the data used for the resizing comes from a trace that reflects the new load pattern. As the new trace has a larger optimal battery, this means that the later after week 104 the resizing is, the larger the battery size. This is true for resizings that take place before the end of week 156, as after this point, the entire data used for the resizing comes from the new trace. For some resizing intervals, the battery is resized multiple times, which leads to higher installation costs, but a lower sum of battery and unserved energy costs. How fast the resizer reacts to the change does not only depend on the resizing interval, but also on when the change happens. For this trace, the 60-week resizer can react by increasing the battery size after 120 weeks. The 52-week resizer on the other hand can only react after 156 weeks, as the change happens after 104 weeks. Therefore, the total costs of the 60-week resizer are lower than those of the 52-week resizer. This randomness of when the resizer reacts and by how much it reacts leads to a spiky Figure 1.

The second insight is that with increasing resizing interval, the total costs also increase. One would expect there to be such a trend, as it takes longer until change is detected with a larger resizing interval. While this trend is quite obscured by the randomness described above, it holds true globally, which can for example be seen for the resizing intervals of 24 and 96 weeks. Figure 3 shows the
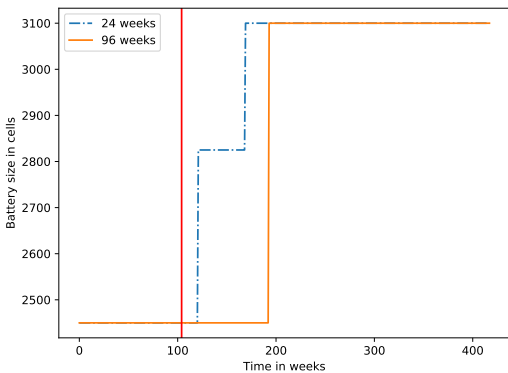

Figure 3: Battery sizes over simulation period for the resizing intervals of 24 and 96 weeks

development of the battery sizes for these two resizing intervals. As can be seen, the 96-week resizer takes longer to detect the change, while the 24-week resizer reacts comparably fast. Although the double resizing leads to higher installation costs, the saving that comes from the better-fitting battery size outweighs these additional costs. The total costs for the 24-week resizer are 56,067 USD, while those of the 96-week resizer are 56,171 USD. The pattern that longer resizing intervals take longer to react can also be seen when comparing the costs of the resizing interval of 88 to 100 weeks. However, as noted above, this is clouded by the spikiness of the graph.

The third observation is that most of the total costs in Figure 1 are quite close together. The costs for all resizing intervals between 4 and 92 weeks are between 56,060 USD and 56,140 USD, which is a range of 80 USD. As discussed above, the sizing's timing plays a

crucial role in the sizing decision's outcome. The more of the new trace is used in the sizing decision, the closer the outcome is to the cost-optimal size. Smaller resizing intervals usually take multiple steps before reaching the final battery size. However, battery sizes that are not perfectly suited, e.g., an intermediate sizing around 2800 cells, do not significantly increase the sum of battery and unserved energy costs. In comparison to the sum of these two costs with the optimal battery size after the load change of 3050 cells, a battery size of 2800 cells increases this sum by about 12 USD per year. As these large sizing differences rarely last for a longer period, the cost difference is quite small.

The data indicates that there is no clear answer to the question which resizing interval minimizes the costs. While there is a slight trend of a cost increase for increasing resizing interval, this trend is especially clear for long resizing intervals of more than around 70 weeks. Moreover, the total costs show a large volatility that is connected to the randomness of when a load trace change occurs and when the next resizing is. While the 60-week resizer performs well in this scenario, it could perform worse if the load change is after 120 weeks. Thus, the total costs of these relatively large resizing intervals can vary depending on the exact trace. While there is no clear cut-off value, the data for other load change times, i.e., after 1 year, after 3 years and after 4 years, shows that resizing intervals below 40 weeks perform well irrespective of when the change occurs. Those resizing intervals therefore seem to be a good choice to reduce costs as much as possible.

5 DISCUSSION AND CONCLUSIONS

This paper is a preliminary exploration of the benefits from periodically resizing storage. Nevertheless, we find that, contrary to our prior intuition, resizing storage does not appear to significantly reduce costs. Indeed, in our setting, we find that even an optimal resizing algorithm that fully knows the future reduces overall costs by less than 1%, so even a highly sophisticated resizing algorithm is not likely make a substantial impact on overall costs. However, our results depend on the following strong assumptions:

- We assume that the load and solar traces from the Pecan Street Project in Austin, TX, are representative of residential buildings worldwide.
- We study only four types of load changes.
- Electricity costs are assumed to only include the cost of energy and do not have a connection cost or a time-of-use component; nor is there a feed-in tariff. In a real-world setting, these more complex situations would apply and would vary from country to country.
- We model the effect of battery degradation with a linear decline in cost of storage over time, which ignores degradation due to energy flux.

We are aware that changing these assumptions may change our results. Therefore, we believe that there would considerable merit in a more thorough analysis of the problem.

ACKNOWLEDGMENTS

REFERENCES

[1] Pecan Street Inc. Dataport. 2019. https://www.pecanstreet.org/

[2] Eurostat. 2020. *Electricity prices for household consumers - bi-annual data (from 2007 onwards)*. Eurostat. Retrieved April 30, 2020 from https://appsso.eurostat.ec.europa.eu/nui/show.do?dataset=nrg_pc_204&lang=en

[3] Liang He, Linghe Kong, Yu Gu, Cong Liu, Tian He, and Kang G Shin. 2019. Extending Battery System Operation via Adaptive Reconfiguration. *ACM Transactions on Sensor Networks (TOSN)* 15, 1 (2019), 1–21.

[4] Tor Aksel N Heirung, Joel A Paulson, Jared O'Leary, and Ali Mesbah. 2018. Stochastic model predictive control—how does it work? *Computers & Chemical Engineering* 114 (2018), 158–170.

[5] Fiodar Kazhamiaka. 2019. *Modelling, Design, and Control of Energy Systems: A Data-Driven Approach*. Ph.D. Dissertation. University of Waterloo, 200, University Ave. W, Waterloo, ON N2L3G1, Canada.

[6] Fiodar Kazhamiaka, Yashar Ghiassi-Farrokhfal, Srinivasan Keshav, and Catherine Rosenberg. 2018. Robust and Practical Approaches for Solar PV and Storage Sizing. In *Proceedings of the Ninth International Conference on Future Energy Systems* (Karlsruhe, Germany) *(e-Energy '18)*. Association for Computing Machinery, New York, NY, USA, 146–156. https://doi.org/10.1145/3208903.3208935

[7] Fiodar Kazhamiaka, Catherine Rosenberg, Srinivasan Keshav, and Karl-Heinz Pettinger. 2016. Li-Ion Storage Models for Energy System Optimization: The Accuracy-Tractability Tradeoff. In *Proceedings of the Seventh International Conference on Future Energy Systems* (Waterloo, Ontario, Canada) *(e-Energy '16)*. Association for Computing Machinery, New York, NY, USA, Article 17, 12 pages. https://doi.org/10.1145/2934328.2934345

[8] Sun Sun, Fiodar Kazhamiaka, Srinivasan Keshav, and Catherine Rosenberg. 2019. Using Synthetic Traces for Robust Energy System Sizing. In *Proceedings of the Tenth ACM International Conference on Future Energy Systems* (Phoenix, AZ, USA) *(e-Energy '19)*. Association for Computing Machinery, New York, NY, USA, 251–262. https://doi.org/10.1145/3307772.3328306