Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Continuous Near-Optimal Control of Energy Storage Systems

Julian de Hoog [*,**] **Ramachandra Rao Kolluri** [*,**]
Peter Ilfrich [*]

[*] *IBM Research Australia, Melbourne, Australia*
[**] *Melbourne School of Engineering, The University of Melbourne*

Abstract: Energy storage plays an essential role in enabling greater uptake of renewable generation. In many applications, energy storage must be used for multiple (sometimes competing) purposes in order to provide the maximum possible economic return. A common approach is to find an optimal sequence of charge and discharge rates for a set of discrete time intervals across the horizon of interest. However, calculating optimal solutions at a high temporal resolution can be computationally expensive. In addition, conditions can change instantly as renewable generation fluctuates and loads are switched on and off. What is needed, therefore, is an approach that can both track some long-term optimal trajectory, while still responding to changing conditions in real-time. In this work, we propose two approaches that aim to achieve this. In the first, we calculate a discrete optimal solution, but then convert this into a schedule for simple rule-based controllers that can respond to changes continuously. In the second, we use historical optimal solutions and rule-based schedules to train a neural network that generates a similar schedule. We find that both approaches offer significant advantages over standard discrete optimal solutions: they provide a similar amount of value (and in some cases more), while being 30x less computationally expensive to compute.

Keywords: Energy storage operation and planning, real time optimization and control, continuous control, dynamic programming, adaptive control by neural networks

1. INTRODUCTION

As the shift to renewable generation continues to gain momentum in the energy sector, the role of energy storage is becoming more and more important. Recent years have seen sharp increases in energy storage deployments, both at the large scale and at the small scale, and the International Energy Agency recently reported that behind-the-meter energy storage deployments nearly tripled from 2017 to 2018 (Luis Munuera (2019)). However, the same report indicates that new markets emerge where governments and utilities have created supportive mechanisms, and energy storage can remain an expensive proposition in many parts of the world, with long payback times.

In most cases, energy storage can be used for multiple different opportunities – often called "value streams" – *i.e.* different ways to reduce cost or provide revenue. For example at the large scale, a single energy storage system may be used to time-shift renewable generation, provide ancillary services, and bid into wholesale markets. At the small scale, a single energy storage system may be used to self-consume local renewable generation, conduct tariff optimisation, or (recently) also bid into wholesale markets.

When multiple such opportunities exist, they are often available at the same time and may compete with one another. To get the most value out of an energy storage system, it is therefore essential to calculate the optimal operational strategy of the energy storage system over a given horizon so that it is used for all of the available opportunities in the best possible way. This strategy typically takes the form of charge and discharge decisions for each of several discrete time intervals throughout the horizon – we call this the *optimal charging solution*.

How to calculate such a charging solution is a well studied problem and many approaches have been proposed. For example, several authors have used standard optimisation approaches such as linear programming (Babacan et al. (2017)), quadratic programming (Ratnam et al. (2015)), mixed-integer linear programming (Khalilpour and Vassallo (2016); Hassan et al. (2017)), or mixed-integer quadratic programming (*e.g.* as a benchmark in Procopiou et al. (2019)). When there are significant non-linear components to the problem (for example, battery degradation models), then dynamic programming is a useful way to find a near-optimal solution (Riffonneau et al. (2011); Abdulla et al. (2016a); Latif et al. (2018)). More recently, different forms of machine learning have been applied to the problem – for example, by encoding a model-predictive control policy using neural networks (Kazhamiaka et al. (2019); Henri et al. (2018)).

In some scenarios, the charging solution is calculated in static, regular intervals, for example day-ahead, once a day. In other scenarios, the charging solution is calculated repeatedly, taking into account changing conditions. Often this is implemented in a model-predictive control manner, where the first one (or several) decisions are applied, an updated charging solution for a full horizon is calculated, again the first one (or several) decisions are applied, and so on, in a rolling manner (Abdulla et al. (2016a); Petrou and Ochoa (2019)).

However, the vast majority of approaches make charging decisions across discrete time intervals and are not able to adapt to changes in conditions that may happen at a timescale smaller than the discretisation interval. In one study, it was found that less than 10% of existing work in this area uses a temporal interval smaller than 10 minutes (Abdulla et al. (2016b)). There have been some attempts to address this – for example by solving multiple dynamic programs at multiple resolutions (Abdulla et al. (2017)), or by using a mix of high and low granularity periods in a mixed integer linear program (Petrou and Ochoa (2019)). In many scenarios, however, decisions ideally need to be made instantly: for example, demand can change at the flick of a switch and passing clouds can lead to steep ramps of solar generation over a matter of seconds.

In this work, we try to address this problem using two different approaches. In the first, we calculate an optimal charging solution over a discrete horizon, but then convert this into an *rule-based charging schedule* for rule-based controllers that can respond to changing conditions in real-time. In the second complimentary approach, we choose a rule-based controller by applying machine learning to previously calculated optimal control responses. In both cases, the energy storage system is able to respond instantly to changing conditions, while still following an optimal trajectory over the chosen horizon.

2. USE CASE AND RULE-BASED CONTROLLERS

To demonstrate the performance of our methods we choose a small-scale energy storage use case: a residential home having rooftop solar photovoltaic (PV) and a Li-ion battery. However, our methods are not limited to this type of scenario, and could be applied in the same way to different sets of value streams, including for large scale systems.

We use data collected at a residential home near Melbourne, Australia during the period March - June 2019. This home has a rooftop solar PV system with a peak capacity of 4.23kW, and many large loads such as a pool pump, electric space heating and cooling, electrical hot water, multiple refrigerators, etc. Solar generation and electrical demand are separately metered and are obtained at a resolution of 1-minute intervals.

Electricity is bought and sold using standard retail rates available in this area. During peak consumption (weekdays between 7:00 and 23:00) electricity costs 37.33¢/kWh, and at all other times it costs 20.44¢/kWh (all dollar values are in Australian currency). At any time, energy can be sold back to the grid via a static feed-in tariff of 12¢/kWh.

We further assume that energy can be sold back to the grid at wholesale market prices when these exceed the feed-in tariff. While this is not fully realised yet (home owners can not yet participate in the wholesale market directly) we nevertheless consider this a reasonable and realistic assumption, since retailers and aggregators increasingly operate these systems on owners' behalves, and are able to participate directly in the market. In Australia the wholesale energy market price is published online by the market operator (AEMO (2019)).

As a battery model we use the specifications of a popular existing commercial offering, which has a capacity of 13.5kWh and peak charge/discharge rates of +/- 7kW.

Three possible value streams exist:

(1) **Solar self-consumption (SSC):** Since there is a high peak import tariff, it is advantageous for this owner to store any excess solar generation and use it to offset peak consumption at a later time. Every kWh of energy stored in the battery for purposes of self-consumption represents a value equal to the difference between buying at peak (37.33¢) versus exporting excess to the grid (12¢), in other words, a value of 25.33¢/kWh.

(2) **Tariff optimisation (TO):** Since it is cheaper to import from the grid at off-peak times, the battery can also be used for tariff optimisation: charging when the cost is low, and discharging to offset any local consumption when the price is high. Every kWh used in this way represents a value equal to the difference between buying at peak and off-peak rates, in other words 16.89¢.

(3) **Market participation (MP):** At certain times, the wholesale energy market price can reach levels that are higher than the off-peak cost of importing energy (*i.e.* greater than \$205/MWh). During such periods, it can be advantageous to discharge the battery to the grid. The per-kWh value in this case depends on the market price in the interval under consideration. In Australia's National Electricity Market there are rare occasions where it can spike to \$12,000/MWh or more (representing a value of \$12 per kWh). However most of the time, including in the time period considered here, there are only occasional short periods where discharging to the grid represents a value of 5¢ - 10¢ per kWh.

Each of the value streams can be represented by a simple rule-based controller, as shown in Equations 1–3. A positive value for r indicates that the battery is charging, a negative value indicates that it is discharging.

d	: current demand	g	: current generation
i	: current import tariff	r	: battery rate of (dis-)charge
i^{min}	: min import tariff	r_c^{max}	: max rate of charge
i^{max}	: max import tariff	r_d^{max}	: max rate of discharge
e	: current export tariff		

$$\textbf{SSC:} \qquad r = g - d \tag{1}$$

$$\textbf{TO:} \qquad r = \begin{cases} r_c^{max} & \text{if } i \leq i^{min} \\ -d & \text{if } i \geq i^{max} \end{cases} \tag{2}$$

$$\textbf{MP:} \qquad r = \begin{cases} r_c^{max} & \text{if } i \leq i^{min} \\ -r_d^{max} & \text{if } e > i^{min} \end{cases} \tag{3}$$

Examples of how these rule-based controllers perform over the course of a day are presented in Figure 1b. Many existing inverters already have different "modes" that make it possible to implement these rule-based controllers on real systems today. We note that this not a form of fuzzy logic control: at any time the charge or discharge rate can be determined exactly based on the rules applied, and there is no need to convert any inputs to fuzzy variables.

3. DISCRETE OPTIMAL SOLUTION

In this section, we describe the approach used to generate the *optimal charging solution* for the energy storage system. We use a simplified version of a dynamic programming-based approach that has previously been extensively described and evaluated in Abdulla et al. (2016a). We consider dynamic programming a useful way to approach problems such as these since it can handle cases where the optimal solution depends on non-linear models (such as *e.g.* battery degradation models), and since it can be extended to take stochasticity of forecasts into account. Interested readers are referred to the paper cited above for further information; here we only provide a summary of the approach for completeness.

We assume we have forecasts for demand d_t, generation g_t, import tariff i_t and export tariff e_t over a full discrete future horizon, $t \in [0,\ T]$. We are aiming to determine the optimal charging solution of the battery, which consists of a sequence of charge and discharge decisions r_t over this horizon (where positive values mean charging, negative values mean discharging). We discretise the possible states of charge (in %) that the battery may have into a set $s \in [0,\ S]$. The dimension of state space A, $\dim[A]$ is $(S+1) \times (T+1)$.

Let the battery capacity (in kWh) be Γ. Transition from a state $A_{s_i,t}$ to a state $A_{s_j,t+1}$ is equivalent to a charge rate decision of:

$$r_t = \frac{1}{\Delta t} \times \frac{s_j - s_i}{100}\Gamma \times \mu$$

where $\mu \in (0, 1]$ represents the (dis-) charging efficiency of the energy storage system.

At any time t, the net impact n_t on the grid can be calculated as:

$$n_t = d_t - g_t + r_t$$

Positive values for n_t mean that energy is being imported from the grid; negative values mean that energy is being exported.

The state transition cost C from a state $A_{s_i,t}$ to a state $A_{s_j,t+1}$ can then be calculated as follows:

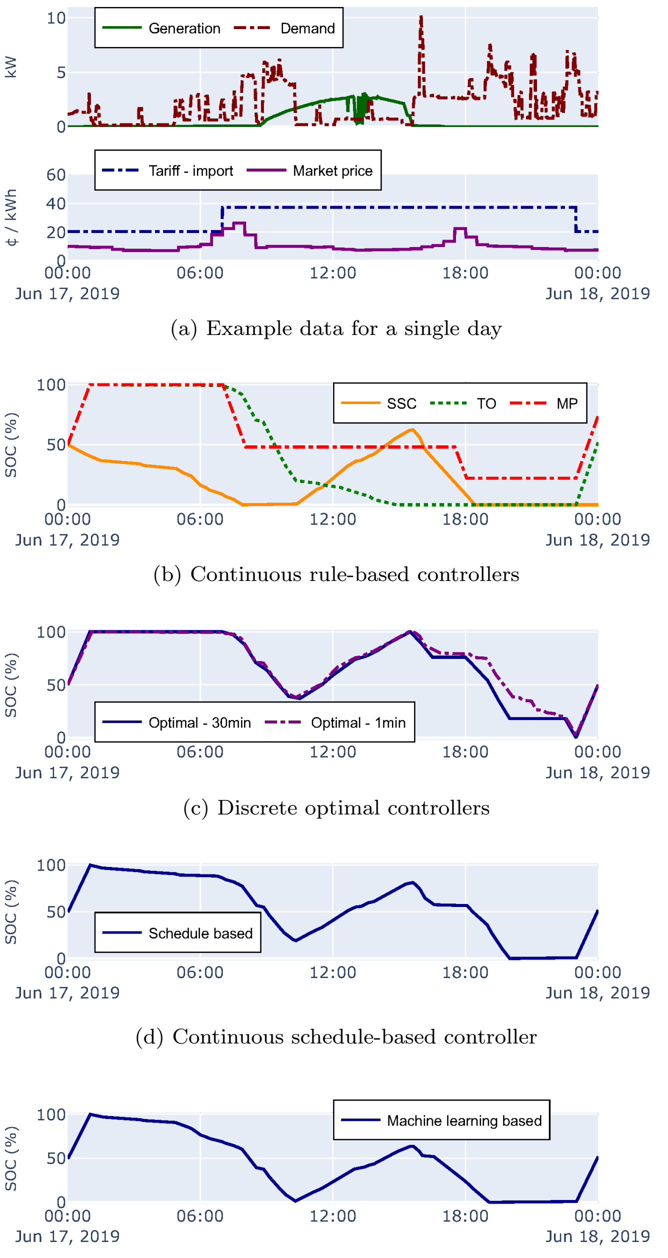
$$C(A_{s_i,t}, A_{s_j,t+1}) = \begin{cases} n_t \times i_t, & \text{if } n_t \geq 0 \\ -1 \times n_t \times e_t, & \text{if } n_t < 0 \end{cases}$$

In words, for any charge decisions leading to a net import from the grid, the relevant import tariff must be paid; for any charge decisions leading to a net export, the relevant export tariff is received.

We determine the "cost to go" CTG for any possible state $A_{s,t}$ as the sum of the state transitions having the lowest joint total cost, using the recursive relationship:

$$CTG(A_{s_i,t}) = \min_{r_t}\{C(A_{s_i,t}, A_{s_j,t+1}) + CTG(A_{s_j,t+1})\}$$

The above is a standard way to set up a dynamic programming problem (Bertsekas (2000)). The dynamic program is solved using backward induction from the final interval T. All final states $A_{s_i,T}$ are initialised to zero. The only exception is if there is a preferred final state of charge s_k, in which case the particular state $A_{s_k,T}$ can be initialised to a very low number (meaning that the solution will finish in this state). The optimal policy from the current state at $t = 0$ is a sequential *charging solution* for our battery that minimises the total cost over the full horizon.



(a) Example data for a single day

(b) Continuous rule-based controllers

(c) Discrete optimal controllers

(d) Continuous schedule-based controller

(e) Continuous machine learning-based controller

Fig. 1. Evolution of battery state of charge across an example day in response to different controllers

This charging solution is truly optimal when evaluated over the discretised intervals used to generate it. However, it is unlikely to be optimal when evaluated at a smaller time resolution. For example, an optimal solution might be calculated across 30-minute intervals over a 24-hour horizon, but when it is applied to smaller intervals (such as 1-minute) it is no longer optimal, due to the many changes in generation and demand that can take place over the course of each 30-minute interval.

Examples of an optimal solution calculated at 30-minute intervals and an optimal solution calculated at 1-minute intervals are presented in Figure 1c.

4. CONVERSION TO CONTINUOUS CONTROL

In reality, circumstances may change faster than 30-minute intervals or even 1-minute intervals; they are changing in real-time as loads are turned on and off and generation ramps up and down (for example, due to passing clouds). Energy storage systems need to be able to respond to such changes instantly. Almost any commercial inverters or charge controllers (which handle the charging and discharging of the battery) are already designed for this. The default mode of most inverters sold with small scale solar-plus-storage systems is to conduct solar self-consumption, and some further have the ability to specify set-point based control to try to take advantage of tariff optimisation (for example, "charge to 70% SOC by 7am").

However, to really get the most out of an energy storage system, it needs to do both: (i) look ahead over a given horizon to anticipate which value streams are available at what times (and what the battery's SOC should ideally be when transitioning from one mode to another), and (ii) respond to changing conditions in real-time.

Much of the existing literature attempts to solve (i), but for most optimisation methods, calculating solutions at very high temporal solutions quickly becomes computationally expensive. Dynamic programming, such as the solution proposed in Section 3, is one such example: since the resolution of the state space must be increased in line with the temporal resolution, an n-fold increase in temporal resolution results in an order n^3 increase in computational effort (Abdulla et al. (2017)).

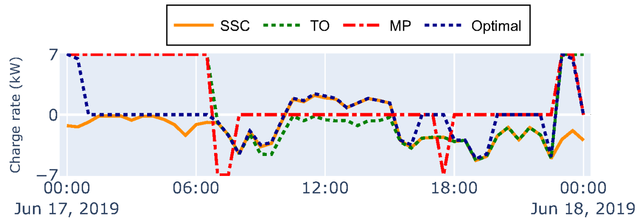
It is necessary, therefore, to focus increasingly on (ii): responding to changing conditions in real-time. Very little work has been done in converting a discrete optimal solution into real-time control. In the remainder of this section, we describe two different approaches that attempt to contribute to filling this research gap.

4.1 Conversion using scheduling

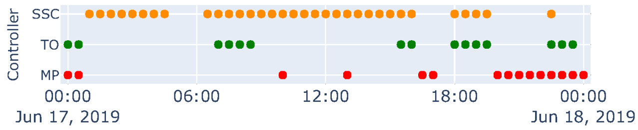
In the first approach, we aim to convert a discrete optimal solution into a schedule for rule-based controllers. The initial idea for this approach has previously been proposed in de Hoog et al. (2018); here we describe a more refined version and provide a more extensive evaluation.

The main motivation for this approach is that we are already able to express the known value streams as very simple rule-based controllers (see Equations 1–3). As mentioned above, most existing inverters already have different "modes" in which they are able to conduct these types of stateless control. At the same time, we can expect that the discrete optimal solution often chooses the same charging rates that one of the known value streams would choose. The key, therefore, is to compare the charging rates of all known value streams with the charging rates specified by the optimal solution: this makes it possible to know which is the best available value stream at which time.

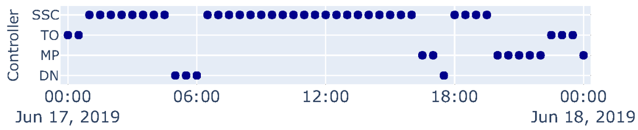
A discrete *optimal charging solution* has the format $[r_t]\ \forall\ t\ \in\ [0,T]$, in other words a choice of a static charge or discharge value for every interval in the horizon. However, following the conversion of the optimal solution into a schedule for rule-based controllers, the *rule-based*



(a) Charging rates for rule-based and optimal controllers



(b) Determining similarity of rule-based controllers to optimal



(c) Resulting rule-based charging schedule

Fig. 2. Generating a rule-based charging schedule

Algorithm 1 Determine similarity matrix

Input:
$[r_t^{OPT}]$, the optimal charging solution;
C, the set of rule-based controllers to compare
s^{max}, the similarity threshold;
Output:
$[s_t^c]$, a matrix indicating binary measure of similarity

// Initialize
$s_t^c = 0 \quad \forall c\ \in C,\ \forall t\ \in [0,T]$

for $t \in [0,T]$ **do**
  for $c \in C$ **do**
    solve r_t^c *// using Equations* (1)–(3)
    if $|r_t^{OPT} - r_t^c| \leq s^{max}$: **then**
      $s_t^c \leftarrow 1$;
    else
      $s_t^c \leftarrow 0$;

Algorithm 2 Generating the rule-based schedule

Input: $[s_t^c]$, the binary measure of similarity
Output: $[c_t]$, the schedule of rule-based controllers

for $t \in [0,T]$ **do**
  // If no controllers are similar, do nothing
  if $\sum_{c \in C} s_t^c == 0$ **then**
    $c_t \leftarrow$ DN
  // If only one controller is similar, use that
  if $\sum_{c \in C} s_t^c == 1$ **then**
    $c_t \leftarrow k$ where $s_t^k == 1$
  // If multiple controllers are similar, find the one having most consecutive intervals of similarity
  if $\sum_{c \in C} s_t^c > 1$ **then**
    $c_t \leftarrow m$ where
      $horizon(m,t) \geq horizon(n,t)\ \forall m,n \in C$

charging schedule has the format $[c_t]\ \forall\ t \in [0, T]$, where c_t specifies which rule-based controller to use in interval t. For example, while the optimal charging solution might choose charging rates of [3.1kW, 2kW, -1.1kW, ...] in the first three intervals, the rule-based charging schedule might choose to use [SSC, TO, SSC, ...] in the first three intervals (see the descriptions of the value streams in Section 2). The resulting behaviour for the discrete optimal solution is to use the same rate of charge or discharge throughout the entire interval, while the behaviour of the rule-based charging schedule is that there can be continuous, instant response to changing conditions throughout the interval.

The conversion of an optimal charging solution into a rule-based controller schedule consists of two main steps:

(1) **Determine similarity matrix of rule-based controllers to optimal solution**. This process is described in Algorithm 1, and shown in Figures 2a and 2b. For every rule-based controller, its charge rates are compared with the charge rates of the optimal solution for every interval within the horizon. If they are similar within a certain threshold, then that rule-based controller is considered a candidate for that interval. The reason that a similarity threshold is required is that the optimal solution uses discretised SOC intervals when calculating the charge rate, and may therefore deviate from the rule-based controller charge rates by a small amount.

(2) **Convert similarity matrix into a schedule**. This process is described in Algorithm 2, and the resulting schedule is shown in Figure 2c. In some cases, no rule-based controller matches the optimal solution – to handle this, an additional rule-based controller, **Do Nothing (DN)**, is introduced (which always has a charge rate of 0). When there is exactly one rule-based controller matching the optimal solution, that controller is chosen. When there are multiple rule-based controllers matching the optimal, the algorithm chooses the one that matches the optimal solution for the largest number of consecutive intervals – to minimise transitions back and forth between different controllers. This is represented in Algorithm 2 by the function *horizon(m,t)*.

The evolution of battery SOC in response to the use of the rule-based schedule is shown in Figure 1d. As can be seen, it closely matches the performance of the optimal solutions.

4.2 Conversion using machine learning

To generate the rule-based schedule described in Section 4.1, we must always previously compute the discrete optimal charging solution. This can be very computationally expensive, particularly when stochastic forecasts and battery degradation models are considered (see Abdulla et al. (2016a)). In addition, if we want to continually take updated information into account (*e.g.* in a model-predictive control-like approach), we have to repeatedly calculate the full optimal solution. As an alternative, we consider here the possibility of applying *machine learning* (ML) to previously computed optimal solutions, in order to develop a model that can generate a schedule of rule-based controllers. Such a model can be instantly applied at every interval without having to recalculate a full optimal solution and without having to go through the process of generating a rule-based schedule.

As the basis for our ML model we explored linear regression, decision trees, extra trees, random forests, support vector regression, and several forms of long short-term memory (LSTM) neural networks. We present here the outcomes of the one that performed the best: an LSTM network with a convolutional layer (CNNLSTM). Such a form of neural network is well-known for fitting sequential data, and for identifying specific patterns in sequential data.

The neural network we ultimately used is presented in Figure 3. It consists of a convolutional long short-term memory layer (`ConvLSTM2D`, cf. Shi et al. (2015)) whose outputs are then flattened to form an activation vector. A number of fully connected layers with added dropout combinations follow the `ConvLSTM2D` layer. The ML schedule is obtained from a soft-max activated fully connected layer at the output. The type of filter, kernel sizes in

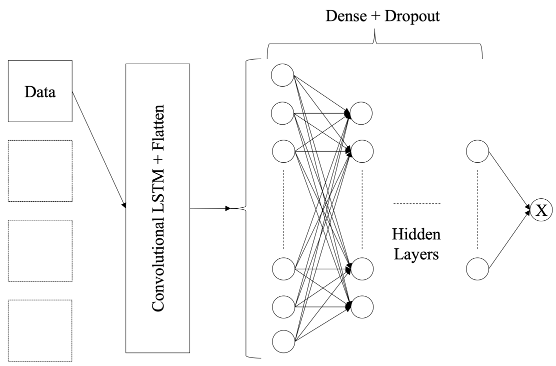


Fig. 3. The neural network used to learn the controller schedule using the historically computed rule-based schedule. Output X is a 'soft-max' activated categorical variable that represents the controller or series of controllers.

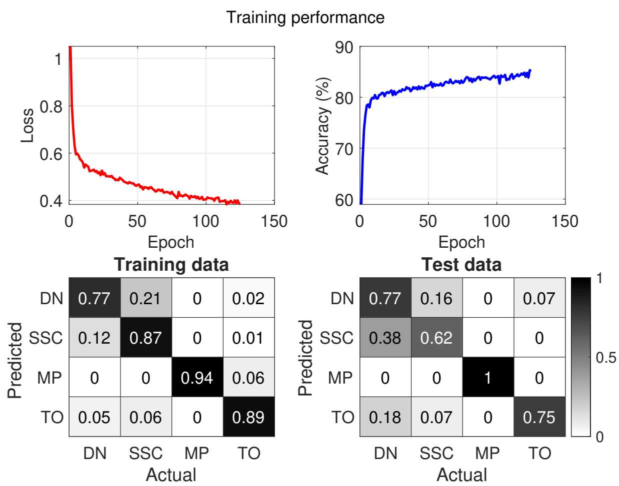


Fig. 4. (Top) Fitting the neural network presented in Figure 3 with training data; (bottom) Normalized confusion matrices representing the neural network's prediction performance over training and test data.

the convolutional layer and the dropout activations in the (hidden) fully connected layers were carefully chosen over an extended period of trial and error to obtain the best training performance. The neural network takes in as inputs:

- x intervals of past generation, demand, tariff, and price information,
- x intervals of past optimal schedule,
- x intervals of past rule-based schedule (if available),
- y intervals of available forecast variables,

and predicts the controller or series of controllers that are to be implemented for the subsequent y intervals. It is important that the past rule-based schedule that is given as input is obtained from the actual conversion from optimal (and not the machine learning prediction) in order to avoid redundancy errors. Given that the neural network is predicting controller schedule, `categorical cross-entropy` has been chosen as the loss parameter that is to be minimized. An adaptive moment estimation based optimizer, `Adam`, (Kingma and Ba (2014)) is used for estimating the weights.

The neural network is trained on three months' worth of data. The test set is 20 days long. Its training performance is shown in Figure 4 (top) and is based on $x = 6$ and $y = 1$. The training reaches an accuracy of 84% in about 125 epochs. The confusion matrices in Figure 4 (bottom) show the categorical prediction accuracy/distribution of the trained neural network on both the training and test data. The diagonal elements in the confusion matrices are relatively larger than off-diagonal elements – indicating high training and test accuracy (≈69%).

There is a slightly lower accuracy for the SSC controller than for the other controllers in the test data. This may be due to the fact that the SSC controller is more likely to recommend charging rates of zero (when generation and demand are similar), leading to misclassifications as DN (do nothing). We consider this unlikely to lead to significantly reduced performance.

The output of the trained neural network for the scenario previously presented in Section 4.1 is shown in Figure 5. As can be seen, it closely matches the rule-based schedule generated in Figure 2c. There are a couple of differences: the ML-based solution does not identify opportunities for market participation and is less likely to suggest *DN* (do nothing) – presumably since there are very few instances of these controllers in the training set. However, when we compare the evolution of battery SOC in response to the use of the ML-based schedule (Figure 1e), we still find that it closely matches the performance of the rule-based schedule and the optimal solutions.

Fig. 5. Machine learning based schedule

5. EVALUATION

We now evaluate both of our continuous control approaches (the rule-based schedule and the machine-learning based schedule), and compare them to the discrete optimal solutions, and to each of the individual rule-based controllers. Since we are using data from a single site, these results can be considered indicative only, and we leave a more extensive evaluation across multiple sites and periods of the year as future work. However, the dataset contains complete 1-minute solar generation and demand values across the full training and testing periods, meaning that the relative benefits of the respective controllers can be evaluated more realistically than other studies using lower resolution.

5.1 Financial cost

Since the ML-based schedule is trained on the first three months of data, we perform a comprehensive cost comparison only on the final 20 days of the dataset that were used for testing – specifically June 1-20. The evolution of total cost incurred for electricity in this household is shown in Figure 6, and a relative comparison of the cost savings provided by each controller is presented in Figure 7.

As can be seen, individual rule-based controllers provide the smallest savings, with tariff optimisation providing the most value. Since June is a winter month, this is due to the low availability of solar generation and high electricity demand due to space heating and cooling; in other months we have found that typically solar self-consumption provides the most value among the available value streams when evaluated individually. In fact, tariff optimisation provides a similar amount of value as the more complex *ML* and *Optimal_30* approaches; however, we anticipate that this would not be the case over the course of a full year, when these would presumably be better at taking into account multiple different value streams available at varying levels at different times of year.

The optimal solution evaluated over 1-minute intervals (*Optimal_1*) clearly provides the greatest savings. Since we are using a 1-minute dataset, this represents the maximum possible value that is available over this time period when evaluating at 1-minute intervals. When the optimal solution is calculated across 30-minute intervals (*Optimal_30*), the savings immediately drop, since the minute-to-minute changes in demand and generation within each interval are not taken into account.

However, the rule-based schedule (*Schedule*), which is derived from the 30-minute optimal solution, provides a high level of value, and is the next best controller after the 1-minute optimal solution. The machine-learning based schedule (*ML*) achieves almost the same amount of savings as a 30-minute optimal solution.

It is worth noting that the financial benefits of energy storage in this case appear not to be that large; however we note again that this is a small case study to demonstrate the feasibility of the proposed approaches. In many other evaluations across broader datasets the value of optimally controlling energy storage has been evaluated more extensively (*e.g.* Abdulla et al. (2016a)).

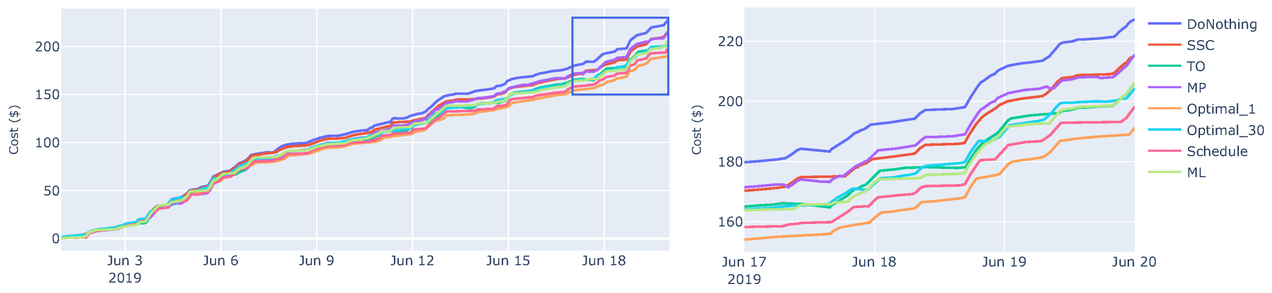


Fig. 6. Comparison of performance across full testing period (left), with zoomed in version of final three days (right)

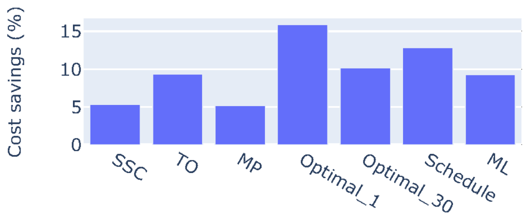


Fig. 7. Financial cost savings attained across full testing period

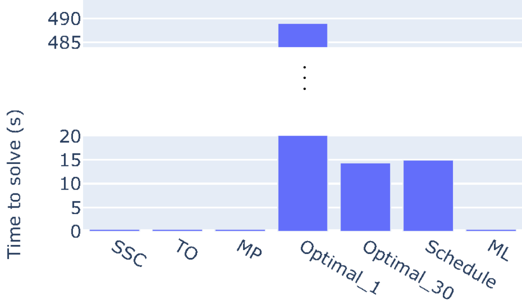


Fig. 8. Time required (in seconds) to generate a complete 24-hour solution on a standard laptop

5.2 Computational cost

When comparing the computational cost, the picture looks quite different. Figure 8 shows the amount of time it takes (in seconds) to calculate a full solution for a 24-hour horizon for each controller, using a standard laptop (2.9 GHz Intel Core i7 processor).

The 1-minute optimal solution takes more than 8 minutes to find a 24-hour solution, while the 30-minute optimal solution requires 15 seconds. The rule-based schedule, which is derived from the 30-minute optimal solution, takes only an additional 0.5s. In other words, the rule-based schedule is more than 30 times faster to compute than the 1-minute optimal schedule.

By comparison, each of the rule-based controllers and the machine learning based schedule are able to generate a full 24-hour solution (at 1-minute intervals) in less than half a second. It should be noted that the time it took to conduct the training of the neural network used in the machine learning model was not included in this comparison, since it is assumed that this can be done offline prior to its implementation.

5.3 Discussion

The above results highlight the importance of converting discrete optimal solutions into some form of continuous control. The majority of the existing literature on optimal operation of energy storage assumes that a solution calculated at discrete time intervals is sufficient. However, as this work has shown, a high-resolution optimal solution can be very computationally expensive. When implemented in a rolling horizon, model-predictive type of way, this means that updated information and changing conditions cannot be taken into account while an updated solution is being computed.

The two methods proposed in this work provide a valuable trade off between a computationally expensive, high-resolution optimal solution, and a fast but myopic rule-based controller. We have shown that it is possible to follow a long-term approximately optimal trajectory, while still being able to respond to real-time changes locally. The rule-based schedule provides more value than the machine learning based schedule, but also takes longer to find a 24-hour solution.

In this work we have not considered the impact of forecasting accuracy on the optimal solutions and rule-based and ML schedules. We anticipate that the scheduling-based approaches are likely to provide further advantages over discrete optimal solutions when inaccurate forecasts affect the results. In future work we hope to explore these impacts in more detail and conduct a more extensive evaluation across multiple sites and times of year. There would also be great value in considering the impacts of operation on battery degradation for each of these approaches, and whether they are able to effectively take this into account.

6. CONCLUSION

Energy storage is playing an increasingly important role in our energy systems as more and more renewable generation is being installed. However, energy storage remains expensive and for such systems to be economically justifiable, typically they need to be used for multiple different purposes. As a result, it is important to operate them in an optimal way, or as close to optimal as possible, to obtain the maximum possible value available.

Much existing work in this area provides methods for calculating an optimal charging solution, consisting of static charge and discharge decisions over discrete time intervals across some horizon. However, for most applications, conditions change in real-time, and both generation and demand can have steep ramps at the minute or sub-minute level. It is necessary therefore to find solutions for operating energy storage that are able to both follow an optimal trajectory over a given horizon, while still being able to respond to changing conditions and sudden events in real-time.

In this work we have proposed two approaches that aim to achieve this. The first uses an optimal solution generated at low resolution (such as e.g. 30-minute intervals), and converts this into a schedule for a set of rule-based controllers that can instantly respond to changing conditions within each interval. The second uses an optimal solution and schedule calculated on previous data to train a neural network that can generate a similar schedule for rule-based controllers. Both the rule-based schedule and machine-learning based schedule provide several advantages. They are considerably faster to compute (30x faster) than a high-resolution discrete optimal solution, and provide more value than any rule-based controller used on its own.

It is hoped that the results presented here contribute to a broader effort to control our energy storage systems more effectively, thus enabling a faster and more cost-effective transition to greater penetration of renewables in our energy generation mix.

REFERENCES

Abdulla, K., Hoog, J.D., Muenzel, V., Suits, F., Steer, K., Wirth, A., and Halgamuge, S. (2016a). Optimal operation of energy storage systems considering forecasts and battery degradation. *IEEE Transactions on Smart Grid*, PP(99), 1–1. doi:10.1109/TSG.2016.2606490.

Abdulla, K., de Hoog, J., Steer, K., Wirth, A., and Halgamuge, S. (2016b). The importance of temporal resolution in evaluating residential energy storage. In *IEEE Power and Energy Society General Meeting*, 1–5.

Abdulla, K., de Hoog, J., Steer, K., Wirth, A., and Halgamuge, S. (2017). Multiresolution dynamic programming for the receding horizon control of energy storage. *IEEE Transactions on Sustainable Energy*. doi:10.1109/TSTE.2017.2754505.

AEMO (2019). Aggregated Price and Demand Data Files. Online: `http://www.aemo.com.au//Electricity/Data`.

Babacan, O., Ratnam, E.L., Disfani, V.R., and Kleissl, J. (2017). Distributed energy storage system scheduling considering tariff structure, energy arbitrage and solar pv penetration. *Applied Energy*, 205, 1384 – 1393. doi:https://doi.org/10.1016/j.apenergy.2017.08.025.

Bertsekas, D.P. (2000). *Dynamic Programming and Optimal Control*. Athena Scientific, 2nd edition.

de Hoog, J., Abdulla, K., Kolluri, R.R., and Karki, P. (2018). Scheduling fast local rule-based controllers for optimal operation of energy storage. In *Proceedings of the Ninth International Conference on Future Energy Systems*, e-Energy '18, 168–172. ACM, New York, NY, USA. doi:10.1145/3208903.3208917.

Hassan, A.S., Cipcigan, L., and Jenkins, N. (2017). Optimal battery storage operation for PV systems with tariff incentives. *Applied Energy*, 203, 422 – 441. doi:https://doi.org/10.1016/j.apenergy.2017.06.043.

Henri, G., Lu, N., and Carreio, C. (2018). A machine learning approach for real-time battery optimal operation mode prediction and control. In *2018 IEEE/PES Transmission and Distribution Conference and Exposition (T D)*, 1–9. doi:10.1109/TDC.2018.8440141.

Kazhamiaka, F., Keshav, S., and Rosenberg, C. (2019). Adaptive battery control with neural networks. In *Proceedings of the Tenth ACM International Conference on Future Energy Systems*, e-Energy '19, 536–543. ACM, New York, NY, USA. doi:10.1145/3307772.3331032.

Khalilpour, R. and Vassallo, A. (2016). Planning and operation scheduling of pv-battery systems: A novel methodology. *Renewable and Sustainable Energy Reviews*, 53, 194 – 208. doi:https://doi.org/10.1016/j.rser.2015.08.015.

Kingma, D.P. and Ba, J. (2014). Adam: A method for stochastic optimization. Cite arxiv:1412.6980. Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

Latif, A., Krishnamurthy, D., and Palmintier, B. (2018). Optimizing storage operation for a probabilistic locational marginal pricing forecast. In *2018 IEEE International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*, 1–6. doi:10.1109/PMAPS.2018.8440374.

Luis Munuera (2019). Energy storage – Tracking Clean Energy Progress. `https://www.iea.org/tcep/energyintegration/energystorage/`. Accessed: 2019-11-02.

Petrou, K. and Ochoa, L.F. (2019). Customer-led operation of residential storage for the provision of energy services. In *2019 IEEE PES Innovative Smart Grid Technologies Conference - Latin America (ISGT Latin America)*, 1–6. doi:10.1109/ISGT-LA.2019.8895319.

Procopiou, A.T., Petrou, K., Ochoa, L.F., Langstaff, T., and Theunissen, J. (2019). Adaptive decentralized control of residential storage in pv-rich mv–lv networks. *IEEE Transactions on Power Systems*, 34(3), 2378–2389. doi:10.1109/TPWRS.2018.2889843.

Ratnam, E.L., Weller, S.R., and Kellett, C.M. (2015). An optimization-based approach to scheduling residential battery storage with solar pv: Assessing customer benefit. *Renewable Energy*, 75, 123 – 134. doi:https://doi.org/10.1016/j.renene.2014.09.008.

Riffonneau, Y., Bacha, S., Barruel, F., and Ploix, S. (2011). Optimal power flow management for grid connected pv systems with batteries. *IEEE Transactions on Sustainable Energy*, 2(3), 309–320. doi:10.1109/TSTE.2011.2114901.

Shi, X., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.k., and WOO, W.c. (2015). Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In C. Cortes, N.D. Lawrence, D.D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 28*, 802–810. Curran Associates, Inc.