Multi-resolution Dynamic Programming for the Receding Horizon Control of Energy Storage

Khalid Abdulla, *Graduate Student Member, IEEE,* Julian De Hoog, *Senior Member, IEEE,*
Kent Steer, Andrew Wirth, Saman Halgamuge, *Fellow, IEEE*

***Abstract*—A multi-resolution approach to dynamic programming is presented which reduces the computational effort of solving multi-stage optimization problems with long horizons and short decision intervals. The approach divides an optimization horizon into a series of sub-horizons, discretized at different state space and temporal resolutions, enabling a reduced computational complexity compared to a single-resolution approach. The method is applied to optimizing the operation of a residential energy storage system, using real 1-minute demand and rooftop PV generation data. The multi-resolution approach reduces the required computation time, allowing optimization to be re-run more frequently, increasing the robustness of the receding-horizon-control approach to forecast errors. In an empirical study this increases the cost-saving offered by a 2kWh behind-the-meter battery energy storage system by 120% on average, compared to an approach using a single fine-grained resolution.**

***Index Terms*—Energy storage, optimal operation, temporal resolution.**

I. Introduction

THE increasing penetration of non-dispatchable distributed renewable generation reduces the operational flexibility of electrical power systems. One approach to regain some operational flexibility is the introduction of distributed energy storage systems (ESSs).

The time-series dynamics associated with ESS make optimizing their operation challenging, especially when these dynamics, and the value-propositions of ESSs, exist over different timescales. For example, frequency regulation services are typically settled in intervals of seconds to minutes, energy is traded on an interval of several minutes, shifting of renewables requires forecasting ahead several hours, and battery degradation occurs over a time-scale of months to years. For many practical operational optimization problems receding horizon control is an attractive approach, as forecasts are only available a finite distance into the future, and are subject to forecasting errors.

Within such a receding horizon control framework, dynamic programming (DP) is a powerful and flexible approach to optimization, especially for the operation of ESSs, where the state-of-charge provides a natural *state* variable. DP is straightforward to implement, usually quick to solve, and flexible with regards to form of the objective and constraints of the problem. However, a key decision in the set up of a DP is the choice of discretization interval, both regarding the state space (state-of-charge) and the temporal resolution (number of stages in a horizon). Small intervals lead to more accurate solutions, but take longer to compute.

In fact, an increase in temporal resolution means that an increased state space resolution is also necessary (to maintain resolution of the power at which the battery is operated), so this also means an increased decision-space. If the horizon length is kept the same, an n-fold increase in temporal resolution results in an order n^3 increase in computational effort per horizon. Additionally, if we seek to implement only a single interval decision before re-solving the optimization (as is typical with receding-horizon-control), the time available to find a single horizon solution is reduced by a factor n. Therefore, the computational difficulty increases with the fourth power of the inverse of the interval length.

Moving to higher temporal resolutions represents a scalability challenge to the discretized DP approach. This is addressed by considering a horizon which is made up of multiple linked DPs, of variable temporal and state space resolution. The problems are linked by interpolating the 'minimum-cost-to-go' (*i.e.* value function) from the initial stage of one program, onto the state space of the final stage of the preceding program. This is illustrated in Fig. 1.

This work addresses some shortcomings of existing methods proposed to optimize the operation of ESSs. The contributions of this paper are to:

(i) introduce an approach to solving the scalability problem of discretized DP optimization of ESS operation at high temporal resolutions (Section III);
(ii) apply this approach to an empirical application of behind-the-meter residential ESSs using real 1-minute household demand and PV generation data [1] (Section IV);
(iii) carry out sensitivity studies to determine the required horizon-length and state space and temporal resolutions for this application (Section VI);
(iv) empirically evaluate the effectiveness of the multi-resolution approach in a realistic computationally-constrained setting, (Section VI-D).

K. Abdulla, A. Wirth, K. Steer and J. de Hoog are with the Melbourne School of Engineering, University of Melbourne, Melbourne VIC 3010 Australia. Contact: kabdulla@student.unimelb.edu.au.

K. Abdulla and J. de Hoog are also with IBM Research – Australia, 19/60 City Road, Melbourne VIC 3006 Australia

S. Halgamuge is with the Research School of Engineering, Australian National University, Canberra, ACT 2601, Australia.

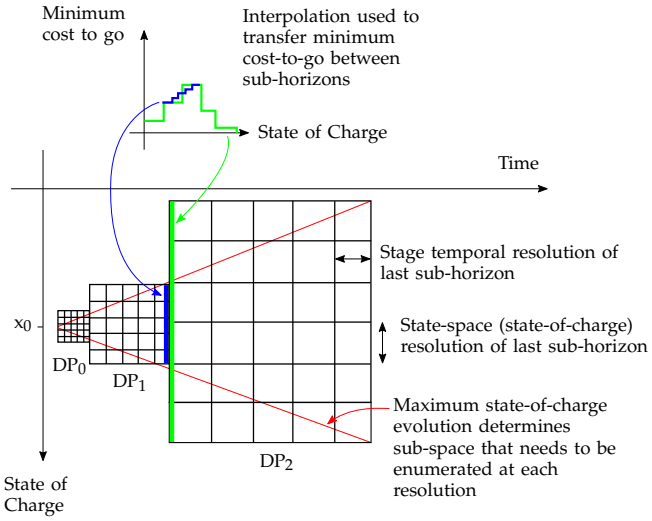

Fig. 1: Illustration of Multi-resolution Dynamic Program for a single horizon, with three DPs (DP_0, DP_1, DP_2) each representing a sub-horizon

II. Background

Dynamic programming has been used extensively for the operational optimization of energy storage systems – for example a recent review describes ten DP-based approaches to the operational optimization of microgrids [2]. However many of the approaches require the battery state-of-charge (which is effectively a continuous quantity) to be discretized into a number of discrete intervals. An accurate representation of the underlying problem requires the use of a large number of small intervals, but this increases the computational complexity of solving the resulting DP problem.

In [3] Codemo *et al*. present four algorithms of varying complexity and their performance when optimizing the operation of an energy storage system subject to demand with a known statistical distribution, and a convex time-invariant cost of imports. They illustrate that under certain conditions the optimal policy can be described as a simple function of the charge level, and the current local net load. Unfortunately, such a simplification is unlikely to be possible for a system with time-varying costs, a complex battery degradation model, and consideration of multiple markets into which a battery can bid its services. For such scenarios being able to directly implement a DP method in an online setting is desirable.

Kamyar and Peet [4] apply multi-objective dynamic programming to optimize the operation of a residential energy storage system considering households with a time-of-use tariff and a peak-demand charge, whilst also considering the degradation of the battery energy storage system. Their approach to dealing with the non-separable peak-demand term is valuable; however their results assume perfect foresight of demand / PV profiles, and their simulations are run at a relatively coarse time interval of 1-hour.

[5] develops a stochastic DP approach which can optimize the operation of distributed energy storage for up to four uses: energy arbitrage, ancillary services (frequency regulation), backup energy, and relief of distribution constraints. This modeling is performed on an hourly time-step, and makes use of a two-step solution procedure; in the first step a discretized state space dynamic program is solved, and in the second step a mixed integer linear program is solved to find near-optimal decisions on a continuous state space. As noted in [6], resolving the full value potential of a residential energy storage system requires simulation and optimization to be performed on an interval of at most a few minutes.

Song *et al*. [7] use a DP approach to optimize the operation of a hybrid energy storage system consisting of a battery and super-capacitor for powering an electric bus. They find that the optimal charging profile determined by DP can be well approximated by a simple rule-based controller, however this varies between the two driving cycles considered, and would also be altered if a different battery degradation model were considered. Song *et al*. explore a rule-based controller for online operation due to the computational tractability challenges of DP, however in the present paper we propose an approach to making DP more computationally tractable so that it can be directly applied online.

Computational tractability is a well-known challenge when solving a discretized approximation of a problem with an underlying continuous state space. This has motivated two broad approaches; (i) Approximate Dynamic Programming (ADP, see for example [8]) and related approaches which do not attempt to evaluate the value function over the entire $\{state, stage\}$ space, but rather sample a large number of paths through this space and approximate the value of a decision in a Monte Carlo style approach, and (ii) intelligent approaches to discretization.

Intelligent approaches to discretization includes a range of methods seeking to find the most effective means of discretizing the state space in a particular problem. For example in [9] Munos and Moore examine various methods of selecting which areas of a state space should be refined when using a variable-resolution approach to solving optimal control problems. They consider top-down approaches to choosing which cells of their state space representation to split (*i.e.* which areas to refine), and focus on a variety of possible refinement criteria. They find that for low-dimension problems refinement based on the value-function alone perform well, but for higher-dimension problems it was necessary to have refinement criteria which depend on novel non-local measures which they term *influence* and *variance*.

Liang *et al*. [10] use a multi-scale DP approach to classify the boundary between tissue layers in images of arteries, during which a coarse-scale image is first annotated with an approximate location of the boundaries, before a finer-scale image is then passed through another DP. In a similar vein, Lo *et al*. [11] use multi-pass DP to

optimize the operation of an ESS. Each pass refines the previous by a factor of two, but searches only in the vicinity of the optimal found from the previous pass, so only local optimality is guaranteed.

In [12] temporal-aggregation techniques are considered to reduce the computational complexity of mixed integer linear programming scheduling problems. In these methods multiple consecutive time intervals are combined into a single interval to reduce the number of decision variables of the problem.

In many existing studies a multi-scale approach is proposed because it is demonstrated that some areas of a problem's state space need to be described at a finer resolution in order to achieve near-optimal control. However, in this study we consider a multi-resolution approach because we are solving this DP within a receding-horizon-controller, and therefore only the first decisions, made at the finest resolution considered, are implemented without recourse. Our approach can be considered analogous to a project Gantt chart, which has greatest detail for those activities due to take place in the near-term. It is not that activities further into the future are less important, or can be decided more coarsely, it is rather that describing them in greater detail can wait until further in the future.

The multi-resolution dynamic programming approach proposed in this paper can be considered a form of "adaptive data processing" as discussed briefly in Section III.D of [13], a classical paper reviewing the procedures available for improving the computational tractability of DP methods.

There are also a similar set of methods in the literature, sometimes referred to as *multi-scale* optimization; [14] provides an example of how such a method can be applied to the optimization of home energy systems. In a multi-scale approach an outer/strategic optimization problem is solved on a coarser time-scale (for example hours) and produces outputs/decisions which are provided as inputs to an inner optimization problem which is solved on a finer time-scale (of minutes or seconds). This approach is distinct to the multi-resolution process presented here, where a single optimization problem, which has stages of differing lengths, is solved.

III. Multi-Resolution Dynamic Program (MRDP)

This formulation could be applied in any setting where discretized state space DP presents tractability issues, but is intended to be applied within a receding-horizon-controller, wherein decisions further into the horizon are subject to recourse, motivating their description at coarser time and state space intervals. The approach is illustrated graphically in Fig. 1, and formalized in Algorithm 1. For now we consider the number of DPs within a horizon, as well as the length and interval-length of each of these programs to be given. An approach to selecting these parameters is presented in Section VI.

Algorithm 1 Multi-Resolution Dynamic Programming

Notation:

N	No. of sub-horizons (dynamic-programs) in horizon;
T_n	No. of intervals in n^{th} sub-horizon;
Δt_n	Duration of intervals in n^{th} sub-horizon;
x, x_0	state variable, and its value at start of the horizon;
Δx	State space discretization at finest temporal resolution;
z_t	Time-series value of interest during interval t;
$\mathbb{X}_n$	Set of reachable state space values in sub-horizon n;
$V^*_{n,t,x}$	Optimal cost-to-go from stage t of sub-horizon n, in state x;
$u^*_{n,t,x}$	Optimal action from stage t, of sub-horizon n, in state x.

Require: $N, \quad x_0, \quad \Delta x$

$\mathcal{N} \leftarrow \{0, \dots, N-1\}$

Require: $T_n, \forall n \in \mathcal{N}, \quad \Delta t_n, \forall n \in \mathcal{N}, \quad z_t, \forall t \in \{0, \dots, T_n - 1\}, \forall n \in \mathcal{N}$

$V^*_{N-1,T_n,x} \leftarrow 0, \ \forall x \in \mathbb{X}_{N-1}$ ▷ Zero cost-to-go from end of MRDP

$V^*_{N-1,0,x}, \forall x \in \mathbb{X}_{N-1} \leftarrow solve(DP_{N-1})$ ▷ Solve last DP

for $n = N-2, \dots, 0$ **do**

  $V^*_{n,T_n,x}, \forall x \in \mathbb{X}_n \leftarrow interp(V^*_{n+1,0,x})$ ▷ Interpolate cost-to-go

  $V^*_{n,0,x}, \forall x \in \mathbb{X}_n \leftarrow solve(DP_n)$ ▷ Solve sub-horizon

return $V^*_{0,0,x_0}, u^*_{0,0,x_0}$ ▷ Cost-to-go & optimal action for the 1st stage of 1st DP.

In Algorithm 1 the time-series of interest, z_t, may be a vector of values over the horizon, and may be forecasts; for example in the case study (Section IV) we consider forecasts of rooftop PV generation and household demand, and the known values of future import/export tariffs. Before being solved, each dynamic program, DP_n, is initialized; in particular we find the set of sets of reachable states by the t^{th} stage of the program, $\mathbb{X}_{n,t}$. Note that if the rate of state change is limited in a particular problem (as is the case with rate-of-charge constraints for an ESS) the full state space does not need to be enumerated at the finest resolution as illustrated in Fig. 1. $solve(DP_n)$ solves the n^{th} DP using backwards induction from a cost-to-go assigned to the terminal states of that program, $V^*_{n,T_n,x}$. Finally, $interp()$ linearly interpolates the cost-to-go from the initial stage of one DP to the ending stage of the previous DP, as illustrated at the top of Fig. 1.

For the case-study we consider, the cost-to-go is naturally a smooth function of the battery's state-of-charge, so linear interpolation was appropriate. Problems with other cost-functions might have less well behaved cost-to-go curves, and may require different approaches towards interpolation. One example would be a use case having a large penalty for being at less than a particular required state-of-charge (*e.g.* in an application where the battery is to be used for satisfying critical loads during a black-out). In such scenarios, the transition from the initial stage of one DP to the final stage of the previous DP needs to be considered differently.

In Algorithm 1 we have maintained generality of any DP which can be solved by backwards induction. In the next section we formulate a MRDP for the specific case of optimizing the operation of a residential ESS.

IV. Optimizing Residential ESS Operation

Multi-resolution dynamic programming is now applied to the operation of a behind-the-meter residential ESS, building on previous work [15]. The tariff structure for our case study (Table I)[1] is typical of those available to residential customers in several states of Australia. It suggests two ways in which energy storage might offer a household cost-savings: by minimizing export of roof-top PV generation to the grid for minimal return, and by minimizing imports from the grid during peak-price times. These savings need to be traded off against the increased degradation of the battery ESS, which has associated with it a fraction of the replacement cost of the battery.

TABLE I: Tariff Structure

Export Tariff, r_t^e	0.05 \$/kWh	flat rate
Import Tariff, r_t^i	0.40 \$/kWh	7:00AM – 10:00PM,
	0.20 \$/kWh	at other times

A. Problem Formulation

Optimizing the operation of a residential ESS to minimize total cost to the householder is a stochastic sequential decision making problem, and we base our presentation on the canonical model in Powell and Meisel [16]. For these types of problems, it is necessary to specify five components; the *state variable*: S_t, the *decision variable*: u_t, any *exogenous information*: w_t revealed as time progresses, the *transition function*: $S_{t+1} = S^M(S_t, u_t, w_{t+1})$ defining the system dynamical model, and the *objective function*.

The objective function of the base problem is to minimize the total cost of electricity paid by the household over a period of consideration, which ends at time index P (P needs to be sufficiently large so that the average cost over the period is representative – for a residential electricity customer a few years would likely be sufficient).

$$\min_{\pi \in \Pi} \mathbf{E}^{\pi} \sum_{t=0}^{P} C(S_t, U_t^{\pi}(S_t)) \tag{1}$$

where Π is the set of implementable policies, and $C(S_t, U_t^{\pi}(S_t))$ gives the cost of electricity in interval t given that the system is in state S_t and we implement the decision $u_t = U_t^{\pi}(S_t)$, under a particular policy π.

a) State variable: our state variable consists of the physical state-of-charge of the battery q_t at the start of interval t, and two measured state variables: the per-interval energy output from a roof-top PV system p_{t-1}, and the per-interval local energy demand d_{t-1} (at time t these energy levels are only known from the previous interval $t-1$). Finally it also consists of the forecasts of energy demand ($\tilde{p}_t$) and PV output ($\tilde{d}_t$) into the future as required by any look-ahead policy (which in the present study are univariate, so based on the time-history of previous states only). This gives us a state variable[2]:

$$S_t = (q_t, p_{t-1}, d_{t-1}, \tilde{p}_t, \tilde{d}_t) \tag{2}$$

b) Decision variable: there is a single decision variable, b_t: the amount of energy to discharge from the battery during interval t, in kWh. We assume that this decision is made in terms of the kWh state-of-charge of the battery, with $\hat{b}$ giving the energy exchanged with the rest of the system accounting for charge/discharge efficiencies:

$$\hat{b_t} := \begin{cases} b_t/\eta_c & b_t < 0 \\ b_t \eta_d & b_t \geq 0 \end{cases} \tag{3}$$

where $\eta_c, \eta_d \in (0, 1]$ are charging and discharging efficiencies of the battery (assumed fixed).

Any excess/shortfall of energy given the flows from the PV system and from/to the battery is made up by grid imports, g_t. The decision b_t is subject to rate-of-charge and state-of-charge constraints of the battery:

$$b_t \leq q_t - \underline{q} \tag{4}$$

$$b_t \geq \bar{q} - q_t \tag{5}$$

$$b_t \leq \bar{b}\Delta t \tag{6}$$

$$b_t \geq -\underline{b}\Delta t \tag{7}$$

where $\underline{q}, \bar{q}$ are the minimum and maximum charge-levels for the battery (in kWh), $\bar{b}, \underline{b}$ are the maximum discharge and charge rates of the battery (in kW) and Δt is the duration of an interval (in hours).

c) Exogenous information: there are two pieces of stochastic information, the change in PV output and demand between interval $t-1$ and t: Δp_t, Δd_t.

d) Transition function: the transition function models the state-of-charge dynamics of the battery, and the evolution of the PV output and demand levels:

$$q_{t+1} = q_t - b_t \tag{8}$$

$$p_t = p_{t-1} + \Delta p_t \tag{9}$$

$$d_t = d_{t-1} + \Delta d_t \tag{10}$$

e) Objective function: the total cost is made up of the costs for any grid imports, less reward given for any grid exports, plus the cost of battery degradation:

$$C(q_t, b_t) = r_t^i [g_t]^+ - r_t^e [g_t]^- + RB\mathcal{D}(q_t, b_t) \tag{11}$$

where $[\cdot]^+$ and $[\cdot]^-$ represent taking the positive and negative component of the argument, $g_t = d_t - p_t - \hat{b}_t$ is the import of energy from the grid during interval t, and r_t^i, r_t^e are the import and export price of energy during interval t (see Table I). R is the per-kWh replacement cost of the battery, B is the capacity of the battery, and $\mathcal{D}$ is the fractional degradation of the battery resulting from discharging b_t kWh from the battery from an initial state-of-charge of q_t (see Appendix B for details of the degradation model used).

[1] \$ symbol refers to Australian Dollars throughout this paper

[2] The forecast variables consist of forecasts over the horizon required by the look-ahead policy

This completes the formulation of the underlying problem. Due to the continuous and stochastic nature of the variables and quantities involved (battery state of charge, expected generation, expected demand, etc.), an exact solution to the problem is intractable. However, [16] provides some discussion of various classes of implementable policies which can be used to find good solutions to these kinds of problems.

In the present study we compare three approaches. This first, used for benchmarking purposes, is a simple rule-based controller which charges the battery with any excess PV output (subject to rate-of-charge and state-of-charge constraints) in order to avoid exporting energy to the grid at unfavorable rates. The second, also for benchmarking, is a receding horizon control approach in which the cost over a finite horizon is to be minimized and a small number of intervals of horizon-optimal decisions are implemented before re-solving the model. We solve an individual horizon using dynamic programming, in which both the temporal horizon and the battery's state-of-charge are discretized into intervals. The third, which is one of the main contributions of this paper, is a receding horizon control approach similar to the one above, but with a novel multi-resolution approach to the solution of the discretized dynamic program, which allows a longer horizon problem to be solved without running into computational tractability issues.

B. Solution using MRDP

We replace the full problem specified in Section IV-A with a sequential series of analogous optimization problems which only consider T intervals, and determine a profile of T optimal charging decisions. For each of these, the first one or more decisions are implemented, and subsequently a revised optimization is solved for a future start time with an updated starting state-of-charge and with updated forecasts. For each individual optimization problem, the point forecasts for that horizon are treated as if they are realized values (*i.e.* the solution is deterministic). However, in the evaluation of the solution we use the true realized values of demand and generation. In other words, forecast error is included as part of the evaluation.

We define the following additional notation, with [units] given where appropriate. The superscript d is used to distinguish between the discretized decision variables b^d, and the underlying continuous decision variable b.

- $x_{n,t}$ integer representing the charge-level of ESS in sub-horizon n, at the start of stage t;
- $\mathbb{X}_{n,t}$ set of reachable integer charge-level states by the start of stage t, in sub-horizon n;
- x^0 the starting integer charge-level for the horizon;
- Δq_n the charge-increment of sub-horizon n [kWh];
- b^d the decision variable: No. of integer charge-levels to drop;
- $\mathbb{B}_{n,t}(x)$ set of feasible discharge decisions during stage t, in sub-horizon n, from state x;
- $\mathcal{C}(x_{n,t}, b^d)$ immediate cost of making discharge decision b^d from state/stage $x_{n,t}$ [$];
- $\mathcal{V}(x_{n,t})$ optimal horizon cost-to-go of state/stage $x_{n,t}$ [$].

Algorithm 2 Solving the n^{th} DP

Require: $\mathbb{X}_{n,t}, \quad \forall t \in \{0 \dots T_n - 1\}$
Require: $\mathcal{V}(x_{n,T_n}), \quad \forall x_{n,T_n} \in \mathbb{X}_{n,T_n}$
for $t \in \{T_n - 1, \dots, 0\}$ **do**
  for $x_{n,t} \in \mathbb{X}_{n,t}$ **do**
    $v^*(x_{n,t}) \leftarrow \infty$
    for $b^d \in \mathbb{B}_{n,t}(x)$ **do**
      $v \leftarrow \mathcal{C}(x_{n,t}, b^d) + \mathcal{V}(x_{n,t} - b^d)$
      if $v < v^*(x_{n,t})$ **then**
        $v^*(x_{n,t}) \leftarrow v$
        $b^{d*}(x_{n,t}) \leftarrow b^d$
return $v^*(x_{n,0}), b^{d*}(x_{n,0}), \quad \forall x_{n,0} \in \mathbb{X}_{n,0}$

The method for solving the n^{th} DP is then given in Algorithm 2. For each stage of the sub-horizon, and each reachable state to be in by that stage, we exhaustively search for the feasible decision with the lowest cost-to-go. Cost-to-go is the immediate cost for making a decision during that stage, plus the cost of being in the resultant state at the start of the following stage. The immediate cost of making discharge decision b^d, from state/stage $x_{n,t}$ is analogous to the base problem cost function, (11):

$$\mathcal{C}(x_{n,t}, b^d) := r^i_{n,t}[\tilde{d}_t - \tilde{p}_t - \hat{b}]^+ - r^e_{n,t}[\tilde{d}_t - \tilde{p}_t - \hat{b}]^- + RB\mathcal{D}(x_{n,t}, b^d) \tag{12}$$

where the battery fractional degradation, $\mathcal{D}$, can depend upon the battery's state of charge, $\underline{q} + x_{n,t}\Delta q_n$, and the battery discharge decision, b^d, (see Appendix B for details).

$\hat{b}_t$ can be defined as follows:

$$\hat{b} := \begin{cases} b^d \Delta q_n / \eta_c & b^d < 0 \\ b^d \Delta q_n \eta_d & b^d \geq 0 \end{cases} \tag{13}$$

where b^d is the number of intervals to discharge the battery by, and Δq_n is the charge-level interval for the sub-horizon.

C. Case-Study Details

1) Battery Properties: are given in Table II. The degradation model is based on that presented in [17], with modifications made to suit the high temporal resolution of this study (see Appendix B).

2) Tariff Structure: is given in Table I, and is typical of those available in parts of New South Wales, Australia.

TABLE II: Battery Properties

Parameter	Symbol	Value	Units
Nominal Capacity	B	2	kWh
Usable Charge Range	$[\underline{q}, \bar{q}]$	[0.3, 1.9]	kWh
(Dis)Charging efficiency	η_c, η_d	0.948	-
Maximum charge current	$I_{ch,\max} = -\underline{b}/B$	2.0	C
Maximum discharge current	$I_{d,\max} = \bar{b}/B$	2.0	C
Nominal charge current	$I_{ch,\text{nom}}$	1.0	C
Nominal discharge current	$I_{d,\text{nom}}$	1.0	C
Nominal Cycle Life	CL_{nom}	3650	No.
Nominal State of Charge	$SoC_{av,\text{nom}}$	55	%
Nominal Depth of Discharge	DoD_{nom}	80	%
Maximum battery life		15	years
Battery Replacement Cost	R	600	\$/kWh
Initial State-of-Charge		50	%
Degradation parameters	See [17] & Appendix B		

TABLE III: Horizon Properties

Parameter	Value
No. of sub-horizons	3
Finest state space resolution	512 states/kWh
Total Horizon Length	1440 min
Simulation Interval	1 min

3) Demand and PV Generation Data: for this study are from [1]. This dataset contains 1-minute household demand and roof-top PV generation for a few hundred homes over several years. For the period 2013-2014, complete data was available for 71 houses, of which 16 were randomly selected for inclusion in the results presented in Section VI-D.

4) Horizon Properties: The baseline properties of the horizons considered in this study are given in Table III – chosen via sensitivity studies (Sections VI-A and VI-B).

5) Benchmark Controller: Performance of the MRDP approach is compared to a single-resolution DP, and a simple set-point controller. The set-point controller seeks to maximize the PV-self-consumption of the household by charging the battery by $p_t - d_t$ kWh during interval t (subject to rate- and state-of-charge constraints).

6) Forecasts: Two forecast models are considered. The first is a Perfect Foresight (PF) forecast with zero error. This is not implementable, but is useful in setting a prescient upper-bound on the cost-saving potential of a battery. The second is a standard auto-regressive forecast, commonly used in time-series forecasting (details in Appendix C).

V. Computation Time for MRDP

In this section we analyze the time-complexity of MRDP, given the number of sub-horizons, and for each the number (and length) of intervals within it.

The computation time is directly proportional to the total number of feasible decisions considered. In Algorithm 2 this is the number of times statement 5 is reached. The number of feasible decisions $\mathcal{O}$ is:

$$\mathcal{O} := \sum_{n\in\{0,\ldots,N-1\}} \sum_{t\in\{0,\ldots,T_n-1\}} \sum_{x\in\mathbb{X}_{n,t}} |\mathbb{B}_{n,t}(x)| \qquad (14)$$

Working from right-to-left we are summing over the number of feasible discharge decisions (the cardinality of the set $\mathbb{B}$), for each state x to be in, for each stage t of each sub-horizon n. The total number of decisions is dependent upon the starting state of charge, because this affects which states are reachable by each stage of each sub-horizon (i.e. the sets $\mathbb{X}_{n,t}$, see Appendix A for details). We are interested in the maximum computation time, so consider the worst-case where the starting state-of-charge is as close as possible to a 50% state-of-charge[3]. The formal definitions of sets $\mathbb{X}_{n,t}$ and $\mathbb{B}_{n,t}$ are consigned to the Appendix, they are simply confined by a battery's rate-of-charge and state-of-charge constraints.

The computation time required as a function $\mathcal{O}$ was determined by solving many MRDP problems of different sizes, and fitting a straight line to the computation times as a function of $\mathcal{O}$. This determined that on average each decision required 1.21μs, when solved on a single core of an Intel© Core™ i7-6700HQ CPU. When simulating the real-time operation of the battery, and the associated optimization problem, we assume a 16-fold increase on this computation time to allow for:

i) More modest computational resources being available in a dedicated embedded controller;
ii) Some values available directly during simulation will need to be estimated from measured data which will have associated over-heads;
iii) There may be communication delays between different pieces of embedded equipment;
iv) It is necessary to ensure the computation is always complete within the required time-frame.

For long horizons with many intervals the maximum optimization time for an embedded controller exceeds a single interval (1-minute in the present study). In these cases it is necessary to implement multiple solutions from a single horizon optimization in an 'open-loop' fashion (*i.e.* without incorporating updated forecasts). The advantage to the MRDP approach is that a long horizon can be considered at a substantially reduced computational cost.

The benefits of lower computation time are two-fold, and are illustrated in Fig. 2, the center of which shows the maximum computation time for a single horizon optimization. During this computation time optimal decisions from the previously-computed horizon are implemented in an open-loop fashion (*i.e.* multiple decisions are implemented without feedback of realized outcomes of the forecast variables). As the computation time increases there are two impacts: firstly, the number of intervals which need to be implemented open-loop increases, reducing the robustness offered by the

[3]The rate-of-charge and rate-of-discharge constraints are the same

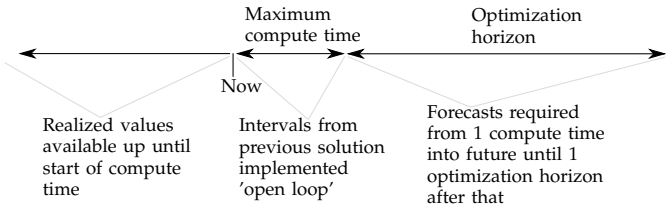

Fig. 2: Illustration of forecast and optimization timeline

recourse of receding-horizon-control. Secondly, there is an increasing time-lag between the most-recent realized value of a stochastic variable which is available, and the first time-interval for which a forecast is required. Both these effects mean that as the computational time of an optimization increases the performance of receding-horizon-control will be reduced. However, given the horizon-myopia of our formulation it is necessary to have a long enough horizon that operation of the battery is optimized effectively. There is a trade-off between computational tractability, and the approximation error associated with formulating our problem as a discrete state space DP. The nature of this trade-off will depend on the specific problem; in particular it will be sensitive to how well the stochastic variables can be forecast.

VI. Results

Referring to Fig. 1, the horizon length, the number of sub-horizons, and the state space and temporal resolution of each sub-horizon need to be chosen. The following empirical procedure is summarized in this section:

i) Determine the horizon length required to achieve an acceptable optimality gap;
ii) At this horizon length, determine the state space and temporal resolution to achieve an acceptable optimality gap;
iii) Split the horizon into three sub-horizons, with a two-fold factor reduction in state space and temporal resolution between sub-horizons;
iv) Increase the factor reduction in step iii) until the solution time is less than a single interval.

In the following sections when we refer to a net cost saving, we mean the reduction in operational cost relative to a case in which no battery is present (*i.e.* excess PV generation/demand is met immediately from grid imports/exports), net of the degradation cost associated with operating the battery. The sensitivity studies are completed assuming unconstrained computational resources (*i.e.* a single interval of the optimal solution is implemented regardless of how long the optimization takes).

A. Selecting Horizon Length

We first decide how far into the future an individual optimization horizon should look. To the best of the authors' knowledge there is no general-purpose approach to selecting an appropriate horizon length for a receding-horizon-control problem. Several theoretical results have been established which under certain assumptions can guarantee the stability and optimality gap of a finite horizon receding-horizon-control solution to an infinite horizon problem, provided the horizon is 'sufficiently large', see [18] for example.

For the ESS operational optimization, there are two value-propositions which must be traded-off; maximizing solar-self-consumption, and minimizing imports from the grid during peak price times. Typically household peak demand will occur around 6-8 PM, and peak PV generation at about midday. Peak- and off-peak tariffs occur on a 24-hour period. The finite capacity of the battery needs to be divided between these two value streams, so we might expect a horizon length of at least 12-hours is required to achieve near-optimal performance.

The top panel of Fig. 3 summarizes an empirical study in which we increase the horizon length, and assess the net cost saved for four households during a 7-day simulation using real demand and PV data. Also plotted are best-fit curves of the form $y = A - Be^{-Cx}$. The parameters of this fit were used to estimate the cost-saving for an infinitely-long horizon, and the horizon length required to achieve a saving within 0.5% of this was found. For the four houses considered this varied between a 800-minute and 1260-minute horizon. Based on these results a horizon length of 1-day (1440-minutes) was chosen for the remainder of our analysis.

B. Selecting State Space & Temporal Resolution

The underlying state-of-charge of a battery is an essentially continuous variable. However, to allow the use of a discretized state space DP approach to optimization, it is necessary to chose some discretization resolution. The results of an empirical study are summarized in the middle panel of Fig. 3, which shows the net cost-saved during a 7-day simulation of four houses with an increasingly fine state space resolution. Also plotted are fits of the form $y = A - Be^{-Cx}$. The parameters of this fit were used to estimate the cost-saving which would be achieved with an infinitely-fine resolution, and the resolution required to achieve a cost-saving within 0.5% of this was found. For the four houses considered this varied between 275 and 580 states per kWh. Based on this a resolution of 512 states per kWh was used for the remainder of our analysis.

An analogous process was used to determine the required temporal resolution, the results of which are summarized in the bottom panel of Fig. 3. Based on the parameters for the plotted fits we determined a resolution of 60-minutes per hour (*i.e.* using the full resolution of the 1-minute dataset) was necessary.

C. Generalising to Other Datasets

The horizon parameters chosen from the sensitivity studies above are specific to the present case-study and

data-set. To understand what drives the sensitivities in general, we looked at the time-series demand and PV data for the customers with the most and least sensitivity to each of horizon length, state space resolution, and temporal resolution.

No particular pattern was spotted when looking at customers who were most/least sensitive to the horizon length considered; indeed the four customers included in the sensitivity study had rather similar sensitivities to horizon length (Fig. 3, top panel). We surmise that for this case-study horizon length sensitivity is driven primarily by non customer specific features such as the tariff structure.

We found that the customer whose net cost saving was most sensitive to state space resolution had a demand profile which varied over only a small range of values; as a result the impact of a coarse kWh-interval control of the battery was more significant.

The customer whose net cost saving was most sensitive to temporal resolution had the most rapid and frequent changes in their demand profile; as a result controllers which implement a single control action for multiple intervals impacted their cost-saving more significantly (this is dealt with in more detail in [6]).

In general, the choice of horizon parameters for other datasets and applications will therefore be somewhat dependant on the particular demand and generation profiles in question (especially their volatility), as well as the underlying tariff structure (or any other factors influencing the objective).

D. Performance Improvement using MRDP

The above sensitivity studies assume the availability of unlimited computational resources for the optimization (*i.e.* if the optimization takes more than one interval to solve, the simulation waits for the result before proceeding). This approach is clearly unsuitable for application in real-time. We now consider simulations which allow for the computational capabilities of an embedded controller (see Section V for details).

There is a fundamental trade-off between optimization complexity (and the associated computation time) and the error resulting from the discretized state space approximation, and the finite-horizon approximation of the underlying continuous state space and infinite horizon objective function respectively. The main thesis of this paper is that a simple multi-resolution approach (illustrated in Fig. 1, and detailed in Algorithm 1) allows for more effective control by choosing a coarser discretization for the state space for intervals further into the future. Results suggesting this are illustrated in Fig. 4, where the net cost-saved over a 7-day simulation is plotted for a number of different methods. Also plotted are the number of 1-minute intervals which have been implemented open loop during the simulation to allow for finite computational resources. From Fig. 4 we see that using a MRDP approach allows fewer optimal control actions to be implemented open loop (due to the reduced computation time) which results in improved performance. For the coarsest resolution horizon, made up of 16 intervals of 1-minute length, 18 intervals of 8-minute length and 20 intervals of 64-minute length, the estimated embedded computation time is less than one interval so only a single interval of the optimized charging profile needs to be implemented without recourse, providing the highest possible robustness against forecast errors.

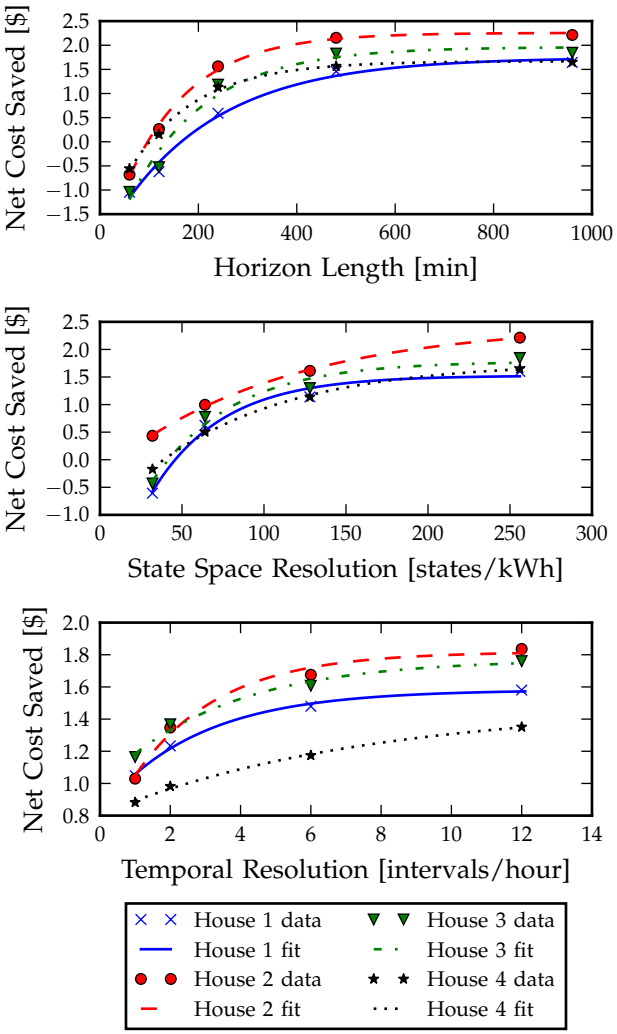

Fig. 3: Net cost-saved for four customers using a 2kWh battery operated using receding-horizon-control (assuming a perfect foresight forecast) plotted against horizon length (top), state space resolution (middle) and temporal resolution (bottom panel).

Results for the perfect foresight forecast were also produced at different horizon resolutions, however the performance was similar because there are no forecast errors necessitating recourse. There was a very slight reduction in performance for coarser horizons due to errors when interpolating the cost-to-go from one sub-horizon to the next. From Fig. 4, we can deduce that a battery with a replacement cost of $600/kWh would be a

borderline economic investment for the households and tariff structure considered. The best-performing method, not requiring a perfect foresight forecast, would deliver $56 of net savings over the year on average (assuming the simulated week is typical), representing a net return-on-investment of 4.7%.

E. Sensitivity to Battery Degradation Model

To assess the sensitivity of these results to the choice of battery degradation model, we repeated the simulations whilst assuming a fixed degradation model in which each kWh of energy charged to or discharged from the battery causes the same amount of degradation. Using this simple degradation model we found: (i) The simple set-point controller was able to achieve performance almost as good as a perfect foresight controller; and (ii) MRDP-based methods using realistic forecasts had a negative net cost saving for horizons with a large number of short stages (due to their long computation times which necessitated the use of more open-loop control). Both of these results underline the importance of taking battery degradation into account in these kinds of studies; and the second observation further confirms that using a multi-resolution approach can offer a performance advantage in computationally-constrained settings.

F. Implementation and Limitations

The proposed multi-resolution dynamic programming approach is relatively straightforward to implement in practice, and the results presented in Fig. 4 are from simulations in which realistic forecasts are used from forecast models only receiving input data available up-until the point in time at which the optimization computation would need to start. These are subject to a restricted computational constraint to simulate the reduced capability of embedded hardware. In general, DP-based methods are typically straightforward to code, test, and implement, and often easier to understand and debug than other optimisation methods – some of which may rely on highly optimized third party solvers that make deployment on embedded hardware less straightforward.

A limitation of the work is that it assumes the degradation model for the battery is known and deterministic (subject to known battery operating parameters), *i.e.* the degradation model assumed for the optimization matches that used for the simulation. In practice battery degradation is a stochastic process, and there are likely to be non-negligible modeling errors in any proposed degradation model. Modelling battery degradation is in general a difficult problem and improved models are the subject of ongoing work. However, one advantage of the type of solution presented here is that new and updated degradation models are typically straightforward to plug into the DP-based problem formulation.

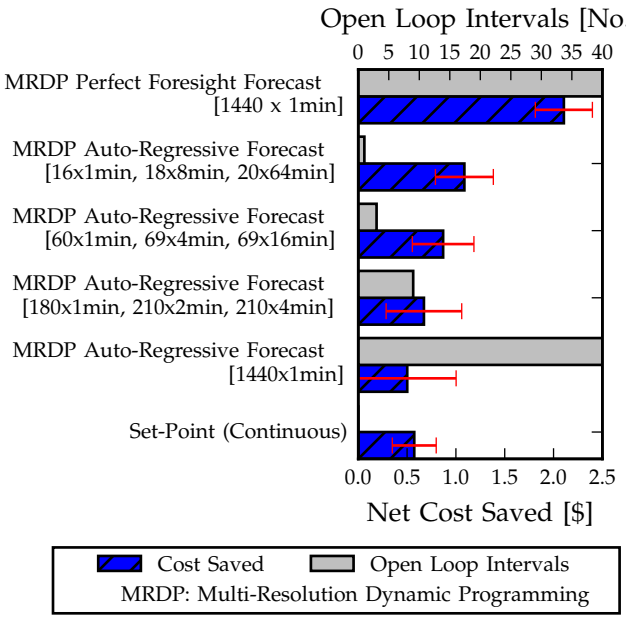

Fig. 4: Net Cost Saved from operating battery for 7 days using various methods. Plots show the average and ± 1 standard deviation over 16 customers considered. The use of MRDP allows fewer open loop intervals to be implemented, resulting in improved performance.

VII. Conclusions & Further Work

A multi-resolution approach to solving discretized state space DPs has been introduced, and its benefits for applications in which optimization is carried out via receding-horizon-control were discussed. The use of a multi-resolution approach allows optimization over a longer horizon than would be possible if the entire horizon were enumerated at the highest temporal and state space resolution, whilst keeping the computation times manageable. As a result optimizations can be re-run more frequently, minimizing the number of control intervals which need to be implemented in an open loop fashion. This delivers the full benefit of the recourse offered by receding-horizon-control. The performance benefit offered was demonstrated empirically in the application of operating a small battery ESS in a home, with simulations based on real 1-min demand and rooftop PV generation data for several homes [1]. If the full 1-day horizon is enumerated at 1-minute resolution, the large computation time necessitates the open-loop (recourse-free) implementation of 40-intervals of the optimal charging profile. The loss of robustness to forecast errors results in poor performance. However if this one-day horizon is divided into a number of sub-horizons of reducing state space and temporal resolution, a 1-day optimization can be solved fast enough, so that it can be re-run every interval. This resulted in an approximately 120% improvement in closed-loop performance, on average, over a 1-week period for the 16 houses simulated.

There are alternative approaches to dealing with the

tractability problem of a discretized state space DP as the control interval gets smaller. A common approach is to have a layered optimization/control approach wherein a long-term optimization solved at a relatively coarse interval determines set-points or other parameters for a much simpler real-time controller. Further work could compare such an approach to the multi-resolution approach presented here. In the present study we have used a simple approach to aggregating a long multi-stage horizon (with 1440 1-minute stages) into a series of sub-horizons of different resolutions. Further work could examine more principled approaches for choosing the timing and nature of the resolution steps, perhaps making use of some of the criteria presented in [9]. Once a series of coarsened sub-horizons have been chosen, note that the original problem is replaced by an approximation of it in which a single decision is applied/implemented for the duration of the aggregated stages.

APPENDIX

A. Definition of Sets $\mathbb{X}_{n,t}, \mathbb{B}_{n,t}$

The sets $\mathbb{X}_{n,t}$ enumerate the states which can be reached by each stage, t, of sub-horizon n. The generation of these sets for a particular problem instance is described below. For convenience a zero-indexing convention is used, *i.e.* the first stage and sub-horizon have index 0. The starting charge-level of the first stage of the first sub-horizon is known to be exactly equal to the starting charge-level state, $\mathbb{X}_{0,0} := \{x^0\}$.

We then define upper and lower bounds for subsequent stages and sub-horizons as follows:

$$\bar{x}_{n,t} := \left\lfloor \frac{\min(\bar{q} - \underline{q}, x^0\Delta q_n - \underline{b}(\sum_{i=0}^{t}\Delta t_n + \sum_{j=0}^{n-1} T_j \Delta t_j))}{\Delta q_n} \right\rfloor,$$
$$\forall t \in \{0,\dots,T_n\},\ n \in \{0,\dots,N-1\}$$

$$\underline{x}_{n,t} := \left\lfloor \frac{\max(0,\ x^0\Delta q_n - \bar{b}(\sum_{i=0}^{t}\Delta t_n + \sum_{j=0}^{n-1} T_j \Delta t_j))}{\Delta q_n} \right\rfloor,$$
$$\forall t \in \{0,\dots,T_n\},\ n \in \{0,\dots,N-1\}$$

where $\lfloor\cdot\rfloor$ returns the largest integer smaller than its argument – in other words the maximum and minimum charge-levels which can be reached if we make the most charging and discharging decisions respectively. We then define the charge levels which can be reached by the start of a particular stage, t, of a sub-horizon, n as:

$$\mathbb{X}_{n,t} := \{x \in \mathbb{N} \mid \underline{x}_{n,t} \le x \le \bar{x}_{n,t}\}$$
$$\forall t \in \{0,\dots,T_n\},\ n \in \{0,\dots,N-1\}$$

where $\mathbb{N}$ is the set of non-negative integers. The sets $\mathbb{B}_{n,t}(x)$ enumerate the feasible battery discharge decisions which can be made given that we are in state x at the start of stage t in sub-horizon n. The generation of these sets is similar and is omitted for brevity.

B. Throughput-based degradation model

Previous work [15] made use of a cycle-based degradation model (first proposed in [17]). A simplifying assumption made to allow that model to be used in a real-time operational optimization was to assume that each discrete (dis)charge decision represented a (dis)charge half-cycle. This was an acceptable approximation when considering the 30-minute resolution of that study, but significantly distorts degradation when applied to 1-minute data.

To resolve this we consider instead a throughput-based degradation model, in which it is assumed that kWh-throughput is the primary factor driving battery degradation. The degradation due to an individual (dis)charge decision is then computed as:

$$\mathcal{D}(q_t, b^d) = \frac{|b^d \Delta q_n|}{2q^{\text{nom}} CL^{\text{nom}} nCL(I_d) nCL(I_{ch})} \qquad (15)$$

where $q_t = x_{n,t}\Delta q_n + \underline{q}$ is the initial battery state-of-charge [kWh] in interval t (of sub-horizon n being considered), $b^d\Delta q_n$ is the amount of energy [kWh] to discharge from the battery during interval t, q^{nom} is the charge/discharge associated with the nominal cycle ($q^{\text{nom}} = DoD^{\text{nom}}B$, where B is the nominal capacity of the battery [kWh] and DoD^{nom} is the depth-of-discharge associated with the nominal cycle [%]), CL^{nom} is the number of cycles to failure under nominal conditions, nCL are the normalized cycle-life functions detailed in [17] and $\mathcal{D}$ is the fractional degradation of the battery. I_d and I_{ch} are the charging and discharging currents implied by charge decision b^d; for discharge decisions ($b^d > 0$) the charging current is assumed to take its nominal value I_{ch}^{nom} and the discharge current is computed as $I_d = |b^d\Delta q_n|/(B\Delta t_n)$, where Δt_n is the interval in hours, and I_d is the discharging current measured in C.

The numerator of (15) is the charge/discharge energy [in kWh] associated with decision b^d, and the denominator is the total energy throughput which is available over the battery's life, modified by the nCL functions to allow for operation at non-nominal conditions (i.e. fewer kWh-throughput are available when operating at higher charge/discharge currents).

This model is similar to that described in [15] and first presented in [17], with two changes. Firstly, we use energy throughput as the dominant determinant of degradation in place of charge/discharge cycles. Secondly, the influence of the cycle-dependent factors (SoC_{av}; the average state-of-charge of a cycle, and DoD the depth-of-discharge of a cycle) has been ignored. This is a simplification of the degradation model (it is assumed that the lifetime energy throughput of a battery is independent of the amplitude and central state-of-charge of operational cycles). Modification of the DP formulation to accommodate cycle-based degradation is an area of further work which is being explored. In addition we consider a minimum per-interval degradation to model calendar battery degradation (as was done in [15]).

TABLE IV: Mean forecast errors for day-long forecasts over 7-days for four households

Forecast Method	Demand RMSE [kW]	PV RMSE [kW]
R forecast package	0.69	0.58
Six-Parameter	0.48	0.25

C. Auto-Regressive Forecast Model

To forecast future values of demand and PV generation we consider a simple auto-regressive forecast model. Demand (or PV output) is assumed to be some linear combination of the following regressors:

i) Realized value in the previous interval;
ii) Realized value two intervals ago;
iii) Realized value from 1-day ago;
iv) Realized value from 1-week ago;
v) The historic average value for this interval-of-day;
vi) The historic average value for this interval-of-week.

The model is applied as a single-interval forecast model in a recursive fashion, *i.e.* if realized values are not available for any of the regressors they are replaced by their forecasts in a recursive fashion.

The forecast is trained as follows; for every interval in the training data-set (in the case study considered above this is the first year of data, with the second year of data used for simulations) for which the regressors can be formed, and the predicted output can be evaluated, a vector of predictor variables and responses (realized demand or PV generation) are formed. These are then used to train a linear regression model to determine the importance of each of the six parameters (using least-squares linear regression). Future values are then predicted by composing a vector of predictor variables for a target interval (and forecasts of those variables where a realized value is not available), and these are multiplied by the coefficients from the training dataset to produce a forecast for that interval.

Performance of this forecast model was compared to a fully automated univariate time-series forecasting tool [19], [20] and was found to perform well. See Table IV for performance of the two methods on average when producing forecasts for 1-day horizons over 7-days of residential demand and PV output for four households from the considered data-set [1]. The relatively poor performance of the fully-automated forecasting approach is thought to be due to the high temporal resolution of the data, and the resulting long seasonal periods (daily period has 24x60=1440 intervals, and weekly period has 7x1440=10080 intervals).

REFERENCES

[1] "Source: Pecan Street Inc. Dataport 2016," https://dataport.cloud/, September 2016, accessed: 25th May 2017.

[2] A. A. Khan, M. Naeem, M. Iqbal, S. Qaisar, and A. Anpalagan, "A compendium of optimization objectives, constraints, tools and algorithms for energy management in microgrids," *Renewable and Sustainable Energy Reviews*, vol. 58, pp. 1664 – 1683, 2016, https://doi.org/10.1016/j.rser.2015.12.259. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1364032115016421

[3] C. G. Codemo, T. Erseghe, and A. Zanella, "Energy storage optimization strategies for smart grids," in *2013 IEEE International Conference on Communications (ICC)*, June 2013, pp. 4089–4093.

[4] R. Kamyar and M. M. Peet, "Multi-objective dynamic programming for constrained optimization of non-separable objective functions with application in energy storage," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, Dec 2016, pp. 5348–5353.

[5] X. Xi, R. Sioshansi, and V. Marano, "A stochastic dynamic programming model for co-optimization of distributed energy storage," *Energy Systems*, vol. 5, no. 3, pp. 475–505, 2014. [Online]. Available: http://dx.doi.org/10.1007/s12667-013-0100-6

[6] K. Abdulla, K. Steer, A. Wirth, J. de Hoog, and S. Halgamuge, "The importance of temporal resolution in evaluating residential energy storage," in *2017 IEEE PES General Meeting (to appear)*, 16-20 July 2017.

[7] Z. Song, H. Hofmann, J. Li, X. Han, and M. Ouyang, "Optimization for a hybrid energy storage system in electric vehicles using dynamic programing approach," *Applied Energy*, vol. 139, pp. 151 – 162, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306261914011696

[8] J. Gönsch and M. Hassler, "Sell or store? an adp approach to marketing renewable energy," *OR Spectrum*, vol. 38, no. 3, pp. 633–660, 2016. [Online]. Available: http://dx.doi.org/10.1007/s00291-016-0439-x

[9] R. Munos and A. Moore, "Variable resolution discretization in optimal control," *Machine Learning*, vol. 49, no. 2, pp. 291 – 323, 2002, http://doi.org/10.1023/A:1017992615625. [Online]. Available: http://dx.doi.org/10.1023/A:1017992615625

[10] Q. Liang, I. Wendelhag, J. Wikstrand, and T. Gustavsson, "A multiscale dynamic programming procedure for boundary detection in ultrasonic artery images," *IEEE Transactions on Medical Imaging*, vol. 19, no. 2, pp. 127–142, Feb 2000, http://doi.org/10.1109/42.836372.

[11] C. H. Lo and M. D. Anderson, "Economic dispatch and optimal sizing of battery energy storage systems in utility load-leveling operations," *IEEE Transactions on Energy Conversion*, vol. 14, no. 3, pp. 824–829, Sep 1999, https://doi.org/10.1109/60.790960.

[12] M. H. Bassett, J. F. Pekny, and G. V. Reklaitis, "Decomposition techniques for the solution of large-scale scheduling problems," *AIChE Journal*, vol. 42, no. 12, pp. 3373–3387, 1996, http://doi.org/10.1002/aic.690421209. [Online]. Available: http://dx.doi.org/10.1002/aic.690421209

[13] R. Larson, "A survey of dynamic programming computational procedures," *IEEE Transactions on Automatic Control*, vol. 12, no. 6, pp. 767–774, December 1967.

[14] L. Jia, Z. Yu, M. C. Murphy-Hoye, A. Pratt, E. G. Piccioli, and L. Tong, "Multi-scale stochastic optimization for home energy management," in *2011 4th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Dec 2011, pp. 113–116.

[15] K. Abdulla, J. de Hoog, V. Muenzel, F. Suits, K. Steer, A. Wirth, and S. Halgamuge, "Optimal operation of energy storage systems considering forecasts and battery degradation," *IEEE Transactions on Smart Grid*, vol. PP, no. 99, pp. 1–1, 2017, https://doi.org/10.1109/TSG.2016.2606490.

[16] W. B. Powell and S. Meisel, "Tutorial on stochastic optimization in energy – part i: Modeling and policies," *IEEE Transactions on Power Systems*, vol. 31, no. 2, pp. 1459–1467, March 2016.

[17] V. Muenzel, J. de Hoog, M. Brazil, A. Vishwanath, and S. Kalyanaraman, "A multi-factor battery cycle life prediction methodology for optimal battery management," in *Proceedings of the 2015 ACM Sixth International Conference on Future Energy Systems*, ser. e-Energy '15. New York, NY, USA: ACM, 2015, pp. 57–66, http://doi.acm.org/10.1145/2768510.2768532. [Online]. Available: http://doi.acm.org/10.1145/2768510.2768532

[18] K. Worthmann, "Estimates on the prediction horizon length in model predictive control," in *20th International Symposium on Mathematical Theory of Networks and Systems (MTNS2012)*, 2012, http://num.math.uni-bayreuth.de/en/publications/2012/worthmann_mtns_2012/worthmann_mtns_2012.pdf, Accessed: 6th February 2017. [Online]. Available: http://num.math.uni-bayreuth.de/en/publications/2012/worthmann_mtns_2012/worthmann_mtns_2012.pdf

[19] R. J. Hyndman, *forecast: Forecasting functions for time series and linear models*, 2016, r package version 8.0, http://github.com/robjhyndman/forecast, Accessed: 10th January 2017. [Online]. Available: \url{http://github.com/robjhyndman/forecast}

[20] R. J. Hyndman and Y. Khandakar, "Automatic time series forecasting: the forecast package for R," *Journal of Statistical Software*, vol. 26, no. 3, pp. 1–22, 2008, http://dx.doi.org/10.18637/jss.v027.i03. [Online]. Available: http://www.jstatsoft.org/article/view/v027i03

Khalid Abdulla (GS'15) is currently pursuing a Ph.D. degree with the University of Melbourne. His research considers the optimal operation of energy storage systems colocated with renewable generation sources, subject to the inherent associated uncertainties.

Julian de Hoog (SM'17) is a Research Staff Member at IBM Research – Australia and an Honorary Research Fellow at the University of Melbourne. His current research interests include optimal operation of distributed energy resources, data-driven forecasting, and Internet-of-Things.

Kent Steer is with Amazon.com Inc. and an Honorary Research Fellow with the University of Melbourne. His research includes distributed systems and deep learning networks..

Andrew Wirth is an Associate Professor with the Department of Mechanical Engineering, University of Melbourne. His recent work has focused on online scheduling problems.

Saman Halgamuge (SM'11) is a Professor and the Director of the Research School of Engineering, Australian National University. His research interests are in Big Data analytics and optimization and in particular bio-inspired methods focusing on applications in mechanical engineering and bioengineering.