Maximum income resulting from energy arbitrage by battery systems subject to cycle aging and price uncertainty from a dynamic programming perspective[☆]

Guzmán Díaz[a,*], Javier Gómez-Aleixandre[a], José Coto[a], Olga Conejero[b]

[a] *Dep. of Electrical Engineering, University of Oviedo, Campus de Viesques, s/n, 33204 Spain*
[b] *ITMA Materials Technology, Parque Empresarial Principado de Asturias, Calafates, 11, 33417 Spain*

Abstract

This paper describes an approach to compute the maximum value of energy storage systems (ESS) in grid applications under uncertain energy prices. The value obtained is based on an optimal operation (consisting of charge/discharge sequences) of the ESS. In other words, it is the maximum value that may be obtained when the ESS charge/discharge sequence is adapted to the expected operational conditions.

To obtain that optimal value, this paper describes a dynamic program approach, with the particularity that the switching decisions are optimized considering an uncertain price evolution and a dynamic calculation of the aging cost. A practical implementation of this approach is proposed, in which the problem is conveniently sliced into matrices corresponding to single decisions. It is shown that such an arrangement, combined with shift and re-indexing operators, provides a fast solution to the optimization problem consisting of a huge number of decision evaluations. The algorithm is then applied to a number of European electricity markets, with a particular focus on arbitrage. The particularities of the algorithm solutions are analyzed, and it is shown that not considering the imperfect foresight and the aging impacts leads to considerable errors in valuing an ESS.

Keywords: economic value, battery aging, dynamic programming, optimization, stochastic analysis

1. Introduction

1.1. Motivation

In developing a perspective on the viability of energy storage systems (ESS) in grid applications, it is necessary to calculate a figure of their cost. In this line, Lazard has recently introduced the concept of levelized cost of storage (LCoS) in an attempt to assimilate ESS to other generation technologies that have been conventionally valued by means of the well-known levelized cost of energy (LCoE). At the time of writing this paper, Lazard has published its second report on LCoS, where it articulates a cost framework with similar implications to those of the LCoE [1]. The interpretation of the LCoS is as simple as for LCoE: it is the cost of generating one kWh. However, it is particularly stressed in Lazard's report that the LCoS is

[☆]This work was supported in part by the Spanish Ministry of Economics, Industry, and Competitivity under Grant ENE2016-80053-R.

[*]Corresponding author
Email addresses: guzman@uniovi.es (Guzmán Díaz), jgomez@uniovi.es (Javier Gómez-Aleixandre), jcoto@uniovi.es (José Coto), o.conejero@itma.es (Olga Conejero)

not as straightforwardly calculated as the LCoE, because it depends on one or several "revenue streams," such as frequency regulation, demand response, arbitrage, etc.

Different operating (charge/discharge) policies responding to those revenue streams lead to notably different economic values [1, 2]. The charge/discharge pattern affects the operating expenses included in the ESS value by affecting (i) the cost associated to the purchase of energy from the grid, instead of the cost of purchasing primary energy or fuel [3], (ii) the opportunity cost incurred by not selling generated energy in case that the ESS is associated to a distributed generator [4], and (iii) the replacement costs because of the different aging caused to the ESS through is activation. It also affects the revenues, because differently from generation technologies, in grid-connected ESS positive and negative cash-flow terms cannot generally occur at the same time.

Lazard's calculations rely on the classical components of the LCoE: CAPEX, OPEX, and incomes. But apparently Lazard's report does not include the cost of aging derived from different charge/discharge patterns into its calculations; though that cost would directly modify the CAPEX over the calculation period by means of the replacement cost. This omission is in striking conflict with the fact that the ESS life cycle can be appreciably altered by the way in which it operates. In energy arbitrage in spot markets, the ESS must follow an uncertain electricity price evolution [5]. In supporting distributed generation, the ESS additionally follows the stochastic evolution of generation and load [6]. Ultimately it is the grid application—the revenue stream—that defines the ESS idle times and cycling pattern, hence ultimately altering the value of the CAPEX. However, the aging issue is notably left out of many comparative economic analysis [7, 8, 9, 10].

1.2. Literature review

Aging of a battery ESS (BESS) can be defined as the modification of its properties—essentially the available energy and power, and the mechanical integrity of cells—with time and use. Two types of aging mechanisms are usually differentiated: during use (on cycling) and on storage (idle). Cycling generally damages the reversibility of materials, whereas storage aging, due to the interactions between active materials and the electrolyte, depends mostly on time and temperature. Storing aging determines the "calendar life" of the cell. Both mechanisms are usually considered as additive, but they can also interact between them [11]. Particularly the cycle-life of batteries, of chief interest in the generation of revenue streams, is affected by the depth of discharge (DoD) and the state of charge (SoC), as well as the operating temperature and the BESS chemistry. For instance, the anodes of Li-ion batteries undergo mechanical strain during cycling at high C-rate and high SoC. This is due to the insertion and de-insertion of the lithium ions, which produces cracks and fissures, and splits the graphite particles, making them less oriented compared to the original platelets. Eventually, it is the nature and orientation of the graphite particles that influences the reversibility of the anode [12]. Also it has been found that overcharging in long-term cycling induces a cumulative damage on $LiCoO_2$ cathodes in rechargeable lithium batteries, producing severe strain, high defect densities, and occasional fracture of particles [13]. Cation disorder, microcracks of the $LiCoO_2$, particles in the cathode, and the increase in thickness of the passive film on the anode due to the reduction of the electrolyte have been linked to the capacity fade of the battery during cycling as well [14]. When the DoD is increased during cycling, also positive active mass degradation quickens [15] and provokes a non-linear

loss of lithium mass [16]. On the whole, cycle aging as a function of the battery operating characteristics is a complex mechanism.

Notably, these electrochemical mechanisms are of difficult implementation in an algorithm devoted to finding the BESS value in grid applications. Necessarily these degradation mechanisms must be translated into mathematical models for efficient integration into an optimization program. Those models must calculate the inflicted damage as a function of the exogenous variables that drive the ESS in its interaction with the electrical system. The alternatives in the literature differ amply in their complexity. In some cases, the figure of degradation is directly approximated as a percentage of the annual use of ESS, regardless of the cycle information [17]. On the opposite side, proposals such as [18] are comprehensive, though remarkably too complex to be applied in economic optimization programs. Other models are simpler, such as [19] for lead-acid batteries, which estimates the capacity degradation of the ESS following a current throughput profile. Wang and colleague's model is still much simpler, opting for a multiparametric exponential formulations that can be regressed against laboratory data [16]. However, it must be stressed that these models still appear to be too focused on the BESS maximum capacity reduction. In applied cost analysis it is actually an "approximate" figure of cost that is needed to evaluate the optimality of a decision. In this respect, Xu *et al.* proposed a formulation also based on stress factors as in [19]. But alternatively in their model, the degradation of a Li-ion battery is a function of the DoD and the average state of charge following a load profile [20]. In some sense, the approach is altogether similar to that in [19], but it concentrates on the exogenous aspects of the BESS operation, producing a simpler model that can be calibrated from degradation test data, as provided by the BESS manufacturer. Moreover, their model decomposes the degradation into a sum of stress factors (again as in [19]), which allows separate analysis of calendar and cycling degradation.

The above considerations about aging would allow incorporating the variable replacement cost into the computation of a BESS value. Indeed, some models in the line of Xu and colleagues' only require cycle information (number of cycles, DoD, and/or SoC) to produce an estimate of the BESS damage [20]. They are remarkably simpler to implement. For instance, the model in [20] has been used in [21] to calculate the economic value of exchanging energy using a Li-ion battery, by means of Matlab/Simulink simulations. But in the field of optimization—regarding the valuation of the BESS based on an optimal operation under electricity prices—only recently some authors have incorporated the cost of aging; revealing a more involved problem. A relevant and complete example was provided in [22]. In a regulation market framework, the authors considered that the cost of cycling at different depths of discharge (DoD) can be prorated and included into a constrained optimization program. The process needs to be simplified, nevertheless, because each time that the optimization program works out a prospective non-optimal solution, the aging value has to be re-assessed by means of new iterations. So the authors resorted to an equivalent cycling count to reduce the computational burden. A similar example, in which BESS was employed to avoid imbalance costs, can be found in [23]. In [24, 25], again the cost derived from aging was calculated ex-post, as a function of daily operation. That is, ex-post calculation of aging seems a requisite of constrained optimization programs.

Incorporating aging into an optimization program is further complicated if the stochasticity of the state

variables is considered. Most of the times, unique historical time series have been employed (instances are [26, 4, 27, 28]), or single realizations of time-of-use tariffs [29]. However, this procedure entails that there is a perfect knowledge about the evolution of electricity price; which is a particularly inaccurate assumption in spot electricity markets. In other cases where the uncertainty in parameters and price are explicitly incorporated in the optimization procedure, the aging is not considered in turn [30]. Some exceptions are [25], employing Monte Carlo samples to recurrently run an optimization routine with ex-post aging calculation; [31, 32] using a constrained optimization program where energy demand is randomly obtained and aging is computed through BESS total energy throughput; [33] proposing a dynamic program that incorporates calendar aging in a multiobjective function, where the uncertainty in loads is accounted for by using two representative days; and [34] where a stochastic dynamic program is, solved iteratively to account for uncertainty.

1.3. Aim and contributions

Our approach to incorporating stochasticity and aging into the valuation problem is different to the approaches based on constrained optimization problems. This paper proposes using a stochastic dynamic program approach to calculate the value of grid-connected ESS, operating under uncertain electricity prices.

Approached in this way, the problem can be sequentially solved as a set of subproblems. It does not require specialized optimization libraries, because it is entirely based on matrix operations. The decision of charging/discharging is made at each time step by observing (i) the expected payoff from switching and (ii) the aging cost derived from the decision.

Our approach is close to that in [34]. But particularly this paper introduces two distinctive contributions:

- First, we considered the stochasticity of prices by conditioning the charge/discharge decisions on the regressed value of the state variable (the price). This negates assuming a perfect foresight of electricity price at the time of the decision, as explained in [35, 36]. Particularly, each charge/discharge decision is made observing ex-ante modeled uncertainty.
- We also integrated the computation of aging dynamically. Whereas in [34] the authors employed a history of decisions to obtain the cycle information (ex-post assessment) we integrated the aging calculation *at the time of the decision*, by using a recursive cycle count based on the approach in [37]. In other words, the degradation induced by a charge/discharge decision is taken into account at the time of valuing the decision (again, ex-ante assessment).

A second approach related to this paper can be found in [38], where the authors provided an interesting comparison of several BESS technologies, based specifically on aging estimates. Particularly, Ciez *et al.* provided a detailed procedure on the valuation of BESS under optimal operation. Our paper also approaches the economic value of BESS under optimal operation but, differently, it details a penalization system of the ESS cycling, depending on the cycling characteristics. This is different to the proposal in [38], which is based on the number of cycles.

Table 1: Summary of citations

Aging models	Optimization programs	Economic valuation	Programming details
[17], [18], [19], [16], [42], [43], [20], [41], [44]	[43], [25], [24], [26], [4], [27], [28], [30], [33], [34], [22]	[39], [40], [7], [8], [21], [22], [23], [24], [25], [38], [9], [10]	[35], [36], [37], [45], [46], [47], [48], [49]

Another two recent papers dealing with the same problem—the valuation of ESS employed in energy arbitrage considering cycle aging—are [39, 40]. Our paper shows a different approach again because we integrated Musallam's model to categorize the damage caused by different switching policies rather than using the number of yearly cycles [40] or the transferred energy [39].This integration is made seamlessly in the dynamic program.

In addition, other contributions of this paper are:

- The approach is based on simple rules to evaluate the optimality of a decision. The different payoff terms are considered additive and can be reformulated easily. In this sense, adaptation to the different revenue streams is simple. We employed an arbitrage setting to illustrate our approach in this paper—the payoff is the product of the charging/discharging decision times the electricity price—but because the market operation payoff is separated from aging, it can be readily reformulated as any other function of the state variable. Also, the separated aging payoff can be reformulated by using a number of degradation functions based on damage by cycling at high SoC as in [41] or other models based on cycling or calendar aging [20].
- Finally, this paper proposes a solution to the management of the huge amount of decision evaluations that must be conducted at each time step. It is based on a re-ordering procedure conditional on the decision indices. In our approach the decisions are classified by types and evaluated in indexed blocks. This provides a structured and fast way of obtaining the solution.

1.4. Document organization

Section 2 details the theoretical background behind the two main component blocks: the stochastic dynamic program and the aging payoff dynamic calculation. Next, Section 3 proposes a practical implementation through an algorithm based on slicing the problem into matrices that evaluate single decisions. In Section 4, the validity of the algorithm results are discussed, and the practical implications resulting from considering (i) the stochasticity of the problem and (ii) the aging of the ESS through cycling are discussed. We particularly employed an analysis of energy arbitrage in several European electricity markets. Finally, Section 5 summarizes the conclusions.

A summary of citations, split according to the use we have made of them in the paper, is shown in Table 1.

2. Methodology

This section describes the employed methodology. A summary is represented in the chart in

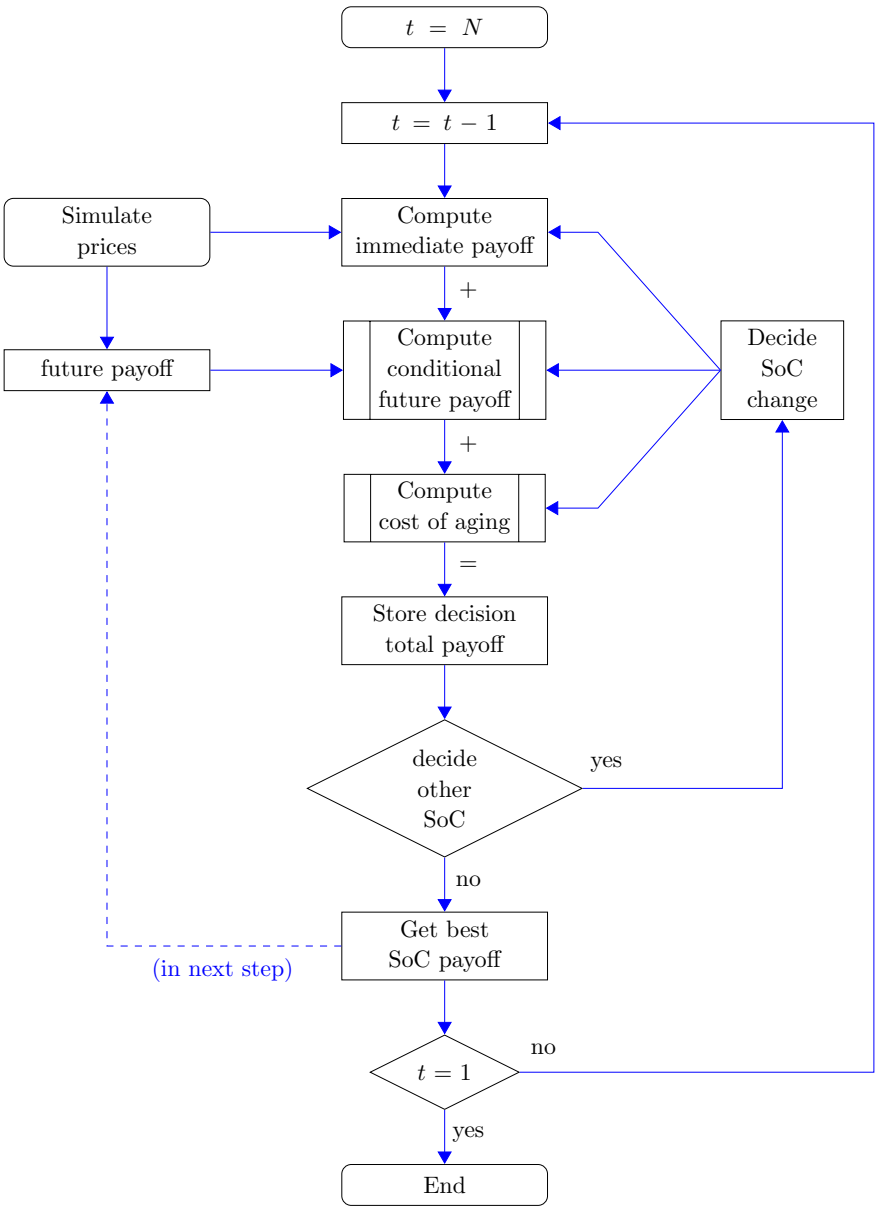

Figure 1: Procedure to compute the optimal operation of the ESS.

2.1. Operation without ESS aging

The problem of computing the economic value of a BESS influenced by exogenous inputs such as electricity price, load demand or power generation, requires defining its optimal operating sequence. The reason is that the optimal operation sequence will produce the best cost-benefit result.

We propose here a dynamic programming approach. It requires the discretization of the problem to obtain a dynamic step-by-step and level-by-level evaluation of payoffs and, by extension, to a definition of the optimal operating policy. The discretization drastically reduces the number of decision variables, naturally concentrating on obtaining the optimal decision at each step, employing a sort of memory to keep track of the influence of results derived in previous steps.

Let $\mathcal{I}_L = \{1, \dots, L\}$ be the discretization of the ESS capacity into L levels, and $d_t = (t, \Delta\ell_t)$ a decision at time t to reach a level $\ell_t \in \mathcal{I}_L$ from the current SoC. The decision is made under the uncertain evolution of a state variable X_t (market price, power generation,...), which can be defined by means of K samples. Associated to that decision is a payoff function $\Pi(X_t, t, \ell_t, d_t)$, which accounts for the profit made by observing X_t and changing the level ℓ_t.

The process of driving the ESS SoC from an initial value ℓ_{τ_1} at time τ_1 to a final value ℓ_{τ_N} can be summarized by a decision vector $\mathbf{d} = (\tau_n, \ell_n)_{n=1,\dots,N}$. The goal of the optimization program is to define an optimal operation sequence $\mathbf{d}^*$ that maximizes the total profit

$$J(X_{\tau_1}, \tau_1, \mathbf{u}) = \mathbb{E}\left[\sum_{t=\tau_1}^{\tau_N} \Pi(X_t, t, \ell_t, d_t) + \zeta(X_{\tau_N}, \tau_N, \ell_{\tau_N})|\mathcal{F}_t\right], \tag{1}$$

where $\zeta(X_{\tau_N}, \tau_N, \ell_{\tau_N})$ is the residual value at the end of the ESS operation trajectory, and the operator $\mathbb{E}[\cdot|\mathcal{F}_t]$ is the expectation conditional on the flirtation $\mathcal{F}_t$. In other words, $\mathbb{E}[\cdot|\mathcal{F}_t]$ it is the expectation conditional on all the available information until time t [45, §5.1], which is but an estimate of the realization of the payoff based on that information. This conditioning, thoroughly explained in [35], is what makes it possible to avoid the perfect foresight of the payoff evolution in the stochastic framework.

To find the optimal value, a dynamic program splits the problem into subproblems. Rather than finding the optimal sequence $\mathbf{d}^*$ of dimension N that maximizes (1) by "testing" different values of $\mathbf{d}$, it divides the problem into N decision problems. At each of these steps, it finds the optimal value

$$V(X_t, t, \Delta t, \ell_t) = \arg\max_{d_t} \left\{\mathbb{E}\left[\Pi(X_t, t, \ell_t, d_t) + \zeta(X_{t+\Delta t}, t + \Delta t, \ell_{t+\Delta t})|\mathcal{F}_t\right]\right\}. \tag{2}$$

arising from an optimal decision d_t^*. That is, it operates by analyzing what is the optimal decision in each interval $[t, t + \Delta t]$.

The ESS valuation problem has a distinct feature. Optimal values cannot be obtained independently in each $[t, t + \Delta t]$. A decision taken at step t necessarily affects the decision at $t + 1$. For instance, completely discharging the ESS at step t makes it impossible to allow for further discharging at $t + 1$. Bellman's principle of optimality provides a way of solving this problem by stating that an optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision [46, Ch. 3]. This means that the problem can be started at the ending step τ_N and be solved backwards, with the functional $\zeta(\cdot)$ accounting

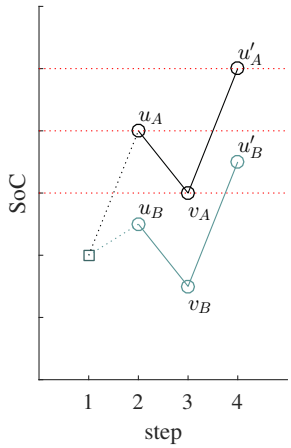

Figure 2: Sample of two operation sequences.

for the optimal decisions ahead. This functional will be the *continuation* value, which will progressively accumulate the optimal values obtained previously in the steps ahead. Bellman's principle ensures that in this way the obtained policy will be optimal.

2.2. *Incorporating ESS aging*

The value of $\Pi(X_t, t, \ell_t, d_t)$ in (2) represents the immediate payoff obtained from exercising a decision d_t at time t from a SoC level ℓ_t, after a state realization X_t is obtained. It must incorporate the incomes (energy sale, for instance) and costs (energy purchase) deriving from the decision. But we claim in this paper that it also has to incorporate the prorated cost of replacement arising from the aging of the ESS. Next, we discuss a possible way of dynamically incorporating the cost of cycling aging. For simplicity, we do not include calendar aging, mindful that it would be easy to incorporate by valuing the ESS idle times.

Most of the previous works on valuing the degradation of ESS by counting the number of switching cycles can be related to the procedure summarized in [44]. This is a standard procedure that makes use of the information provided by a complete time series to find the number of cycles and half-cycles. It servers well to the purpose of counting cycles ex-post, when the entire operation series is available. But the limitation to apply this procedure to our approach is immediately evident, however, precisely because it requires the complete time series. But by contrast, our DP approach builds the SoC series backwards progressively as it selects the optimal decisions, which in turn depend on the cycle counting.

Musallam and Johnson proposed in [37] a dynamic alternative, different to the "static" procedure described in [44]. Their approach consists of progressively counting and canceling out cycles and half-cycles as new points are supplied to a rolling time series. To that end, they introduce two memory stacks in which the maximum and minimum values of the time series are dynamically stored and updated. Their paper gives a detailed example of application, with which the rules for cycle counting can be readily understood. In what follows, we build on that algorithm to propose a backwards-counting version that accounts for the calculation of the number of cycles and their depth, using also the same memory stack idea that allowed the dynamical counting in [37].

To provide a simplified explanation about how Musallam and Johnson's paper translates into our approach, we shall use in what follows the schematic shown in Fig. 2. It represents two possible decisions that are evaluated at step 1. We assume that the two continuation paths, A and B, have been already computed (the algorithm proceeds backwards) from step 2 onwards, and of course they are optimal. Both paths might have merged in the future, but never crossed because of Bellman's principle. We also assume that the memory stack of maxima for path A stores the maxima u_A (the latest that has been recorded proceeding backwards) and u'_A. The stack for minima stores v_A. The same would be of application to path B. That is, each optimal path would have two memory stacks.

A memory stack may be used to split the decision space into four intervals. In the case of path A, its information defines the decision space as $[0, v_A) \cup [v_A, u_A) \cup u_A \cup (u_A, u'_A] \cup (u'_A, L]$, where L is the maximum SoC level. By doing this, we can propose the following counting rules:

a. If $\ell_1 = u_A$, where ℓ_1 is the level investigated at step 1 (denoted by a square in Fig. 2), then it means that the decision investigated is to remain idle. Neither the cycle count nor the memory stacks would be modified.
b. If $\ell_1 \in (v_A, u_A]$, the investigated decision at step 1 is a charge. (Note importantly that the action is reversed from what would be a forward calculation like in [37].) However, it is not a change in the SoC level large enough to trigger a count calculation. But being a variation in sign with respect to the next decision (discharge), it must modify the memory stack of minima.
c. If $\ell_1 \in (0, v_A]$, the change would be significant. This is particularly the case represented by path A in Fig. 2. Translating the procedure in [37], it means that the cycle count must be updated. Looking into the future, from ℓ_1's perspective, a successive maximum u_A would occur, followed by a minimum v_A; and possibly another maximum u'_A. So, reverting the procedure by Musallam and Johnson, this case means that a *charge* decision is investigated, with sufficient magnitude so as to amount to a half-cycle of height $u_A - v_A$. Alternatively, as it is the case represented in Fig. 2, the existence of u'_A indicates that the decision investigated constitutes a full-cycle of the same height. In both cases, the minimum v_A must be removed from its stack, which would be updated with ℓ_1. In the second case, where u'_A exists, the maximum u_A must be similarly removed from its stack.
d. If $\ell_1 \in (u_A, u'_A]$ the decision that is investigated from ℓ_1 is a future discharge. According to [37], and again reverting the reasoning, the change is not significant. It is similar to the case in which $\ell_1 \in (v_A, u_A]$, but in this case the updating of the maximum stack amounts to substituting ℓ_1 with u_A.
e. Finally, if $\ell_1 \in (u'_A, L]$, a half-cycle of depth $u'_A - v_A$ must be declared, with ℓ_1 subsequently replacing u'_A.

The above procedure systematically adapts the forward, one-path dynamic counting methodology in [37] to deal with the backward, multi-path framework addressed in this paper. When a SoC level is investigated to see how possible decisions will result in different cycle counts, it suffices to examine the content of the two stacks of each level and determine which is the interval in which they fit best. Thus for instance in Fig. 2, the decision to increase the level from ℓ_1 to u_A would be case c. The decision to charge the ESS up to level u_B would be case b. Note that only the two last values of both stacks, which importantly are

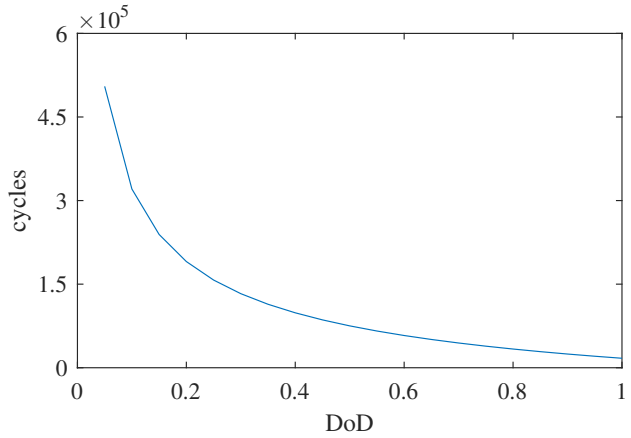

Figure 3: Relationship between maximum estimate of cycles to failure and DoD [20].

LIFO lists, are needed. With longer stacks, further delving into the future decisions makes it necessary to evaluate recursively their contents, as in [37]. But the classification proposed above would still hold. The DoD corresponding to each cycle or half-cycle event would be computed as the minimum level occurring in that event.

At each step, this dynamic rainflow counting method yields the number of cycles and half-cycles, along with the additional information regarding their depth and amplitude, ahead of each decision. This information must be translated into a figure of cost. The procedure is straightforward. We use the relationship between DoD and expected maximum cycle number represented in Fig. 3 [20]:

$$c_{\mathrm{F}}(\mathrm{DoD}) = (1.40 \times \mathrm{DoD}^{-0.501} - 1.23) \times 10^5. \tag{3}$$

By application of Palmgren-Miner's rule (see for instance [47, 48]), the equivalent total damage or spent life can be obtained as the sum of damage at each DoD as a fraction of the total life estimated by the manufacturer. This damage is but a reduction of the lifetime of the BESS. So the cost of following a decision can be calculated as a portion of the CAPEX given by that number of consumed cycles:

$$p_{\mathrm{a}} = \left(\sum_{\mathrm{DoD}=0}^{L} \frac{c(\mathrm{DoD})}{c_{\mathrm{F}}(\mathrm{DoD})} \right) \times \mathrm{CAPEX}, \tag{4}$$

where $c(\mathrm{DoD})$ is the equivalent number of cycles obtained by the rainflow count at a given DoD.

3. Practical implementation

Essentially, the practical implementation must provide a means to assess the payoff of any possible decision at a given step by observing a given state (for instance, price) realization. The payoff is indeed a combination of the instant payoff from exercising a decision and the uncertain future payoff from continuing at the decided new level. Moreover, the instant payoff from exercising a decision is also a combination of income and aging cost.

In what follows, we propose a program that combines these payoff calculations by means of simple matrix operations. The program uses payoff matrices at time step t that store the payoff from a decision

taken after a given price realization (matrix row) at a given starting SoC level (matrix column). There will be one payoff matrix for each possible charge/discharge decision and, eventually, it will suffice to compare all these matrices to obtain the optimal decision. Each of these payoff matrices will additionally be calculated as the summation of instant payoffs (aging and charge/discharge) and the expected continuation payoff, which will be obtained from a regression of the optimized payoff in the previous step $t+1$. Key to all these operations will be the pre-conditioning of matrices by re-indexing them using a distinctive decision code.

3.1. Main dynamic program

The algorithm must operate in a three-dimensional framework, where the time path of a stochastic variable—mostly a price signal—and the SoC of the ESS dictate the decisions to follow to reach optimality. That space can be discretized into N time steps, K realizations of the price, and L levels of SoC; so that each "cell" of the $(K \times N \times L)$-order array will represent a unique SoC, which observes a price realization in a given instant. It is in that setting that the complexity of the practical implementation arises from the problem formulation: How, for each and every cell of that three-dimensional array, is it decided which decision to make (charge, discharge, or stand by), if optimality depends on the SoC at the time of deciding? Moreover: How can it be assessed the aging cost at each and every of those cells, when this also depends on the evolution of the SoC (which turns out to depend on the decisions taken)?

To simplify this problem, we followed an approach based on operating at each time step with a kind of step-ahead memory, either for the optimality of the decisions ahead in the Bellman's sense or for the accumulation of aging through ESS operation. Regardless, although this reduces the complexity, the amount of decisions to check at any time step still remains huge. The space is of dimension $K \times L$, and looping through each element to evaluate the cost of each and every possible decisions would render the procedure cumbersome and computationally expensive.

But by working with *stacks* of cost matrices computed from plain decision arrays, we reach a solution that actually simplifies the task of evaluating the huge amount of options. Particularly for each possible decision, we define a $(K \times L)$-order array, in which we code the charge/discharge action by means of an integer representing the number of levels that the SoC trajectory would move up or down. For instance, if the ESS capacity is discretized in a number of levels so that each one means a charge/discharge of 10 kWh, and if the maximum allowed rate of change is 100 kWh, then the possible decisions are encoded in a vector of $M = 21$ entries as $\mathbf{d} = (-10, -9, \ldots, 0, \ldots, +9, +10) = (d_1, \ldots, d_M)$.

Each element of this vector eventually produces an array of order $K \times L$ with all its entries equal to that element. Thus we have M decision arrays of the type

$$\mathbf{D}^{(i)} = \begin{pmatrix} d_i & \ldots & d_i \\ \vdots & \ddots & \vdots \\ d_i & \ldots & d_i \end{pmatrix}, \quad i = 1, \ldots, M, \tag{5}$$

which must be evaluated with regard to the their effect on the payoff. If the decision is to buy/sell energy

at price samples $\mathbf{x} = (x_1, \ldots, x_K)$, then the corresponding payoff is

$$\mathbf{P}_{\mathrm{e}}^{(i)} = \begin{pmatrix} d_i \times x_1 & \ldots & d_i \times x_1 \\ \vdots & \ddots & \vdots \\ d_i \times x_K & \ldots & d_i \times x_K \end{pmatrix}, \quad i = 1, \ldots, M, \tag{6}$$

Similarly, payoff arrays $\mathbf{P}_{\mathrm{a}}^{(i)}$, also of order $K \times L$, are obtained by submitting each $\mathbf{D}^{(i)}$ to the aging calculation algorithm (later explained). Eventually, a set of M payoff arrays $\mathbf{P}_{\mathrm{e}}^{(i)}$ and $\mathbf{P}_{\mathrm{a}}^{(i)}$ would be available, each one corresponding to a unique decision; tested for every price and SoC.

These immediate payoffs—derived from the decision to change the SoC at the current time step—must be supplemented with those payoffs that will follow in the future as a consequence of the change now decided. It is not acceptable an immediate large payoff if the decision leads the SoC to a level that will entail a continuation path of low value. So the optimal decision must also take into account the accumulated payoff from the SoC following an optimal trajectory in the future; which depends on the current SoC, the price observation, and the decision taken. To again avoid evaluating such a huge number of possible trajectories, we employed the concept of continuation value in the sense explained by Longstaff and Schwartz in [35]. The continuation value accumulates the payoff obtained from following optimal policies from the step ahead up to the end. It is based on Bellman's Principle of Optimality, which states that whatever the initial state (price) and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision [46, Ch. 3]. So the continuation values can be regarded as a kind of summary about the payoffs ensuing from taking only optimal decisions in the future. Longstaff and Schwartz employed continuation values for each price path. We simply expand that concept to a two-dimensional framework to incorporate the current SoC levels.

The matrix of continuation values cannot be exactly the data stored from the previous optimization during the step ahead, however. This would mean that we know what is going to happen in the next step—meaning a perfect foresight flaw [35, 36]. We employed two three-dimensional arrays, of order $K \times N \times L$, to store the continuation values (array $\boldsymbol{\mathcal{V}}$) and the optimal decisions (array $\boldsymbol{\mathcal{D}}$). These are filled backwards, as the optimization process progress. So the *certain* step-ahead continuation value is the slice matrix $\mathbf{V} = \boldsymbol{\mathcal{V}}_{:t+1:}$ (this notation is detailed in [49]), which is a matrix of order $K \times L$. But because we want to avoid the perfect foresight, matrix $\mathbf{V}$ cannot be directly used, and instead we use a matrix $\mathbf{W}$, which is a regression of the columns of $\mathbf{V}$ and the state vector $\mathbf{x}$ [35, 36].

An additional practical problem arises when the M immediate payoff matrices ($\mathbf{P}_{\mathrm{e}}^{(m)}$ and $\mathbf{P}_{\mathrm{a}}^{(m)}$, with $m = 1, \ldots, M$) have to be compared to the continuation value in the step ahead. By construction, the element $p_{\mathrm{e},k\ell}^{(m)}$ of matrix $\mathbf{P}_{\mathrm{e}}^{(m)}$ is the payoff from exercising the decision m under a price realization k when the SoC is at level ℓ. $p_{\mathrm{e},k\ell}^{(m)}$ is the element of $\mathbf{P}_{\mathrm{e}}^{(m)}$ in the k-th row and ℓ-th column. In our algorithm, the continuation value is also an array of order $K \times L$ that stores the sum of payoffs obtained from exercising an optimal policy. Its element $w_{k\ell}$ refers to the continuation of the SoC trajectory starting at level ℓ (using price path k). But importantly, $\mathbf{P}_{\mathrm{e}}^{(m)}$ and $\mathbf{W}$ refer to different time steps: $\mathbf{P}_{\mathrm{e}}^{(m)}$ to the current step and $\mathbf{W}$ to the step ahead. The former is the payoff from exercising now, whereas the latter is the payoff from following an optimal trajectory in the future. To be congruent, therefore, the total payoff from the decision must

be evaluated as $p^{(m)}_{\mathrm{e},k\ell} + p^{(m)}_{\mathrm{a},k\ell} + w_{k(\ell-m)}$, because taking the decision to change the SoC m levels makes the continuation to start m levels apart. It is evident, therefore, that this prevents the algorithm from evaluating the total payoff as $\mathbf{P}^{(m)}_{\mathrm{e}} + \mathbf{P}^{(m)}_{\mathrm{a}} + \mathbf{W}$.

The solution that we employ takes advantage of having split the problem into M matrices, each one containing the result of a unique decision at the current time step. Each matrix informs about one unique charge/discharge decision, of application to every level and path. This is relevant because if the decision is to change the SoC m units, then it is readily observed that it will suffice to *horizontally shift* the entire matrix $\mathbf{W}$ by an amount $-m$; completing the unavailable entries after the shift with $-\infty$ to indicate that the decisions lying out of bounds are of infinite cost. For instance, for a two-observation sample of prices, (x_1, x_2), with three levels of SoC, the payoff (immediate and future, combined) from reducing one level the SoC when $\mathbf{d} = (-2, -1, 0, +1, +2)$ is represented as:

$$\mathbf{C}^{(2)} = \underbrace{\begin{pmatrix} 2x_1 & 2x_1 & 2x_1 \\ 2x_2 & 2x_2 & 2x_2 \end{pmatrix}}_{\mathbf{P}^{(2)}_{\mathrm{e}}} + \underbrace{\begin{pmatrix} p^{(2)}_{\mathrm{a},11} & p^{(2)}_{\mathrm{a},12} & p^{(2)}_{\mathrm{a},13} \\ p^{(2)}_{\mathrm{a},21} & p^{(2)}_{\mathrm{a},22} & p^{(2)}_{\mathrm{a},23} \end{pmatrix}}_{\mathbf{P}^{(2)}_{\mathrm{a}}} + \underbrace{\begin{pmatrix} w_{13} & -\infty & -\infty \\ w_{23} & -\infty & -\infty \end{pmatrix}}_{\text{shifted } \left[\mathbf{W}=\left(\begin{smallmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \end{smallmatrix}\right)\right]} \tag{7}$$

This procedure ends up producing a set of matrices $\mathbf{C}^{(m)}$, $m = 1, \ldots, M$, which must be investigated to finally extract the optimal decisions as those that maximize the results. Again to avoid costly loops, we devised a simple procedure consisting on stacking the M matrices. This yields an array of order $K \times M \times L$. And because each component of the stack corresponds by design to an only decision, it suffices to obtain the index of the stack in the second dimension to get the optimal decision. Indeed, this procedure can be conducted in bulk, extracting a $(K \times L)$-sized array that contains the indices. These can be finally mapped into the optimal decision matrix, $\mathbf{D}^*$, by means of the vector $\mathbf{d}$. (For instance in the previous example where $M = 21$, if for the $k\ell$ entry of price realization and SoC, the index of the maximum in the stack analysis is $m = 2$, it means that the optimal decision is -9, which is the second entry of $\mathbf{d}$, i.e. discharging 9 units or 90 kWh.)

Once the optimal decisions for all prices and SoCs are embedded in the matrix $\mathbf{D}^*$, the step is abandoned after repeating the previous procedure of immediate payoff calculation to accumulate to the continuation value and store it in $\boldsymbol{\mathcal{V}}$.

The procedure is summarized in Algorithm 1.

3.2. Aging computation

The calculation of the immediate payoff in Algorithm 1 requires a computation of the cost related to the ESS aging. Cycling the ESS through different DoDs will entail a degradation that a supplementary algorithm to Algorithm 1 should compute in an efficient way to not overload the calculation procedure. This must be conducted in a dynamic way, because the decisions are influenced at each step by the amount of degradation that they can cause. In other words, the traditional ways of counting cycles reported in [44] that require the whole loading cycle are of not application in this setting, because precisely the loading cycle will be decided at each step regarding the degradation caused and the profit obtained. Our implementation of a rainflow counting algorithm based on the work by Musallan and Johnson makes the computation in

Algorithm 1: Dynamic program to compute the optimal switching

Function `OptimalSwitch`
  Input : a time series (state variable)
  Output : optimal decision and related cost
  Init : $\boldsymbol{\mathcal{V}} \leftarrow -\infty$; $\boldsymbol{\mathcal{V}}_{:N\ell_N} \leftarrow 0$;
  for $t \leftarrow N$ **to** 1 **step** -1 **do**
    $\mathbf{V} \leftarrow \boldsymbol{\mathcal{V}}_{:t+1:}$;
    $\mathbf{D}' \leftarrow \boldsymbol{\mathcal{D}}_{:t+1:}$;
    Define : $\mathbf{W} \leftarrow \mathrm{E}[\mathbf{V}|\mathbf{x}]$, the continuation values conditional on the state vector
    foreach $d = d_{\min}, \ldots, d_{\max}$ **do**
      Define : $\mathbf{D} = [d_{ij}] \leftarrow [d]$, an all-$d$ matrix of order $K \times L$
      $\mathbf{P}_\mathrm{e} \leftarrow [d_{ij} \times x_i]$;
      $\mathbf{P}_\mathrm{a} \leftarrow$ `RainflowCount(`$\mathbf{D}$, $\mathbf{D}'$`)`;
      $\mathbf{W}' = [w'_{ij}] \leftarrow [w_{(i-d)j}]$;
      Pile $\mathbf{P}_\mathrm{e} + \mathbf{P}_\mathrm{a} + \mathbf{W}'$ on the decision stack;
    Define : $\mathbf{D}^* = [d^*_{ij}]$, with the position of maxima in the stack of $\mathbf{P}_\mathrm{e} + \mathbf{P}_\mathrm{a} + \mathbf{W}'$
    $\mathbf{P}^*_\mathrm{e} \leftarrow [d^*_{ij} \times x_i]$;
    $\mathbf{P}^*_\mathrm{a} \leftarrow$ `RainflowCount(`$\mathbf{D}^*$, $\mathbf{D}'$`)`;
    $\mathbf{W}'^* = [w'^*_{ij}] \leftarrow [w_{(i-d^*)j}]$;
    $\boldsymbol{\mathcal{V}}_{:t:} \leftarrow \mathbf{P}^*_\mathrm{e} + \mathbf{P}^*_\mathrm{a} + \mathbf{W}'^*$;
    $\boldsymbol{\mathcal{D}}_{:t:} \leftarrow \mathbf{D}^*$;
  return $\boldsymbol{\mathcal{V}}$, $\boldsymbol{\mathcal{D}}$

Algorithm 2: Dynamic, multilevel rainflow counting algorithm

Function `RainflowCount`
  Input : $\mathbf{D} = [d_{k\ell}]$, $\mathbf{D}' = [d'_{k\ell}]$, maxSOCStack, minSOCStack
  Output : cycle count for minimum and mean SoCs
  Get : preconditioned reindexed versions of $\mathbf{D}'$, maxSOCStack, and minSOCStack, using $\mathbf{D}$
  foreach *level* (ℓ) *and path* (k) **do**
    $\mathbf{u} \leftarrow$ maxSOCStack; $\mathbf{v} \leftarrow$ minSOCStack;
    $\Delta\ell_k \leftarrow d_{k\ell}$;
    $\Delta\ell'_k \leftarrow d'_{k\ell}$;
    if $\Delta\ell_k > 0$ **then**
      if $\Delta\ell'_k > 0$ **then**
        remove last entry from $\mathbf{u}$;
      `CountCharging(`ℓ_k, $\Delta\ell_k$, $\mathbf{u}$, $\mathbf{v}$`)`;
    if $\Delta\ell_k < 0$ **then**
      if $\Delta\ell'_k < 0$ **then**
        remove last entry from $\mathbf{v}$;
      `CountDischarging(`ℓ_k, $\Delta\ell_k$, $\mathbf{u}$, $\mathbf{v}$`)`;
  Compute : ESS degradation, $\sum_{\mathrm{DoD}=0}^{L} \frac{c(\mathrm{DoD})}{c_\mathrm{F}(\mathrm{DoD})}$, using Palmgren-Miner's rule
  return *spent CAPEX*
Function `CountCharging`
  Input : ℓ_k, $\Delta\ell_k$, $\mathbf{u}$, $\mathbf{v}$
  Output : charging cycle count for minimum and mean SoCs
  Compute : min SoC (related to DoD) is the last recorded min SoC
  if $\ell_k >$ *last recorded maximum SoC in* $\mathbf{u}$ **then**
    remove that last entry from $\mathbf{u}$;
    if *there is more than one recorded event in* $\mathbf{v}$ **then**
      remove last entry from $\mathbf{v}$;
      declare full cycle;
    else
      declare half cycle;
    return `CountCharging(`ℓ_k, $\Delta\ell_k$, $\mathbf{u}$, $\mathbf{v}$`)`
  else
    $U \leftarrow U \cup \{\ell_k\}$;
    return *updated* $\mathbf{u}$ *and* $\mathbf{v}$, *and number and composition of cycles*

a dynamic way. Again, the main challenge is to manage the sizable combination of decisions, levels, and paths.

We refer to Algorithm 2 for details. The inputs to the algorithm are $K \times L$ matrices of current decision and ahead-decisions—respectively $\mathbf{D} = \boldsymbol{\mathcal{D}}_{:t:}$ and $\mathbf{D}' = \boldsymbol{\mathcal{D}}_{:t+1:}$. Also inputs are the stacks of maximum and minimum SOCs. These stacks are L vectors of variable dimension, depending on the history of the SoC trajectory. For each level, each of this vectors or sets will be here denoted as $\mathbf{u}$ and $\mathbf{v}$; the maximum (peaks) and minimum (valleys) recorded SoCs, respectively.

The algorithm starts by determining which levels and paths will behave as charging or discharging events. This is decided after consulting the value of the current decisions in the $K \times L$ decision array (see line 8). A decision to discharge the ESS is revealed by $\Delta\ell_k > 0$, and it would trigger the charging count. Before proceeding, however, the step-ahead decision would be checked. Were the ahead decision also charging—which is tracked on the preconditioned step-ahead optimal decisions—then the last element of the vector of maximum recorded SoC $\mathbf{u}$ would be removed (line 11). This amounts to the peak-valley filtering described in [44, §5.4.4].

We divide the counting into charging and discharging counts, resulting in dedicated algorithms; which are quite similar, but operate on different SoC stacks. See for instance the charging count function starting in line 19 of Algorithm 2. The SoC amplitude and mean value, and the DoD (or the related minimum SoC) are computed before modifying the $\mathbf{u}$ and $\mathbf{v}$ stacks. Thereafter, if the investigated storage level exceeds the last recorded maximum, the count is triggered, following two possible scenarios. If there was a previous minimum SoC, more than one so that a charge-discharge action can be declared, a full cycle is counted. Conversely if the minimum SoC stack was empty or had one only value, a (discharging) half cycle would have been counted, without emptying the stack.

The algorithm works recursively (see line 2). Thus, before updating the SoC maximum level, the evaluation is recalled, with now the last maximum (and possibly the last minimum) removed, to investigate if more updating of the stacks and further cycle counting is necessary. Eventually, if the analyzed storage level ℓ_k does not trigger more cycle counts, its value is stored in the maximum SoC stack. This level will hence become the last recorded maximum.

It is easy to see that the counting of a discharging decision follows the opposite pattern, but it has the same structure. The level ℓ_k would be tested to be *lower* than the last recorded minimum SoC. And the roles of the stacks $\mathbf{u}$ and $\mathbf{v}$ would be exchanged.

To speed up the computation, our algorithm employs reindexed versions of the "continuation" data. The reindexed versions consist in a reordering of the data, taking as indices the current decision codes. This preconditioning can be efficiently and easily obtained by array indexing in implementations in Matlab, Python or R, for instance. As a result, for each storage level and path the current and step-ahead values will be aligned. We have found that this preconditioning of the arrays notably improves speed and simplicity, because it later avoids complex looping to discover which will be the decisions and stack contents one step ahead of each level and path. Indeed, because current and step-ahead decisions become aligned, matrix operations can be easily performed between current and step-ahead conditions.

This algorithm is called several times during the computation of each step optimal decision. A first set of calls are made during the building of the payoff stack regarding the evaluation of the prospective decisions (line 12 of Algorithm 1). But it is important to realize that these calls *must not modify* the SoC stacks. They

are queries meant to assess the aging provoked by potential decisions. It is only after obtaining the optimal decision matrix that in a second call (line 17 of Algorithm 1) the SoC stacks must be modified, so that in the next optimization step the record about the cycle count is available.

Eventually, at each step the optimal decision is based on the sum of all the three constituent payoffs or costs (see line 19 in Algorithm 1): future expected payoff, aging cost, and payoff from charging/discharging. Interestingly, this disaggregation leads to think that the model may be upgraded by adding other payoffs or penalties, which could represent some other operational constraints (see for instance in [50] an interest selection and detailed explanation of some). For instance, high charge rates could be easily penalized by affecting the decision vector with a non-linear cost function.

4. Results and discussion

In this Section we provide an account of numerical tests conducted to clarify and validate the performance of the algorithm. We start with a simple deterministic analysis, just to show the optimal operation policy foundation and how the incorporation of the aging subroutine affects the decision process. Then, we expand the analysis to a more generic stochastic case, examining the reasons why the deterministic analysis might offer inaccurate results in some scenarios of high ESS CAPEX. Finally, we elaborate on a wider analysis of how the algorithm produces different policies under different input series with varied structural composition.

In all the numerical analyses we employed an energy arbitrage focus, because of its simplicity compared to other ESS applications. Based on the simple policy "buy low sell high," it allowed us to concentrate on the numerical performance of the algorithm. The payoff function that was evaluated at every cell was simply defined as the energy traded times the price, with a sign depending on the decision code (meaning sale or purchase). We modeled the electricity prices of several European spot markets using the approach detailed in [36].

Importantly, the aim of this section is not to provide a detailed account of the impact that using BESS has on the long-term profitability of operation in different markets. Neither this section is intended to constitute an assessment of the effect that different price structures provoke on the full life cycle of the BESS. Its aim is to illustrate the methodology and compare the different results that are obtained when the aging is incorporated into the decision analysis under different price signals.

4.1. Deterministic optimal operation policy

Fig. 4 serves to clarify the meaning of optimal operation in an arbitrage scenario, as well as to verify the correctness of the algorithm decisions. The analysis shown is based on the Fingrid spot price (January, 2017), where remarkable spikes happened in two occasions, along with a string of spikes in the last week. These events make the series ideal for arbitraging purposes.

The results of arbitraging without taking the cost of aging into account show some particular, recognizable features. First, the largest discharges are placed at moments that exhibit the largest price variations—the spikes. Obviously, the charges happen before those events. Also, when there is a string of spikes such as that starting on January 23, the discharge is organized in a sawtooth pattern, to take advantage of the

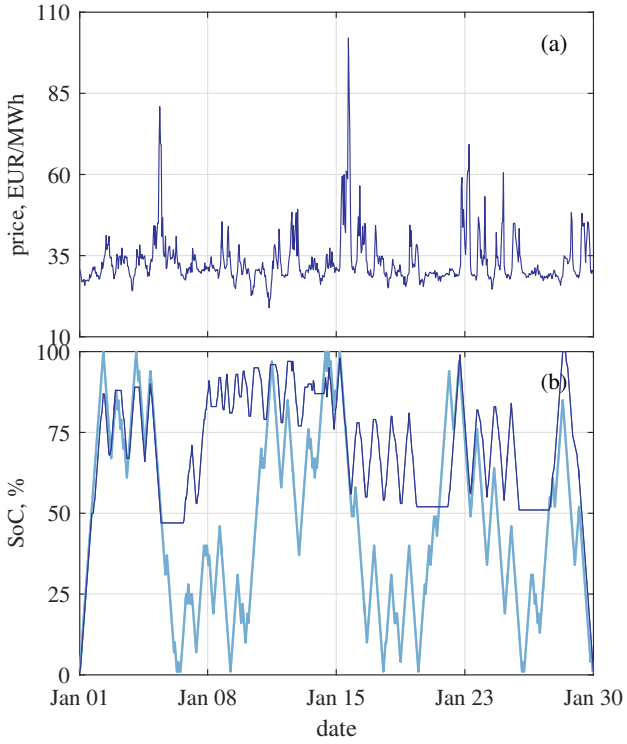

Figure 4: Switching analysis from a deterministic point of view. (a) Spot price in January, 2017, of Fingrid market. (b) Switching policy for optimal energy arbitrage with (dark, thin line) and without (light, thick line) aging cost. ESS CAPEX was 125 €/MWh.

lower prices in between spikes. But on the whole, a discharging trend can be observed. Finally, the ESS uses the whole SoC range—from fully charged to fully discharged—regardless of the damaging large DoD events. This is further corroborated by the histogram in Fig. 5a, where the prevalence of high DoD cycles is readily observed.

The inclusion of the cycle count routine and the associated cost visibly changes the arbitrage policy. Still the computed policy tries to get the most benefit from the price variations during the spikes. But differently now, the payoff contains the information about the ESS reposition due to cycling at large DoDs. Thus, the payoff obtained from lowering the SoC below 50% does not make up for the prorated cost of reposition, which increases when the SoC is reduced. This is in clear contrast with the policy disregarding the cost of reposition derived from the aging count. Also, because higher operation frequencies are less damaging at lower DoDs, larger number of cycles are observed in the high-SoC trajectories. In Fig. 5b, the operation is concentrated to the left of the panel, meaning a significant attempt to avoid a damaging high DoD operation. Note that this result obtained by the algorithm can be related to the shape of Fig. 3.

Definitely, the optimal operation of the BESS is a compromise solution between two conflicting policies. If aging is neglected, the BESS operation will directly follow an income-maximization policy. This is the operation reflected in Fig. 5a, where the algorithm proposal is to "vigorously" cycle the BESS, preferring large charge/discharge cycles that take advantages of price spikes. But this policy would not have into account the cost derived from the BESS degradation. On the other hand, if the aging is to be minimized,

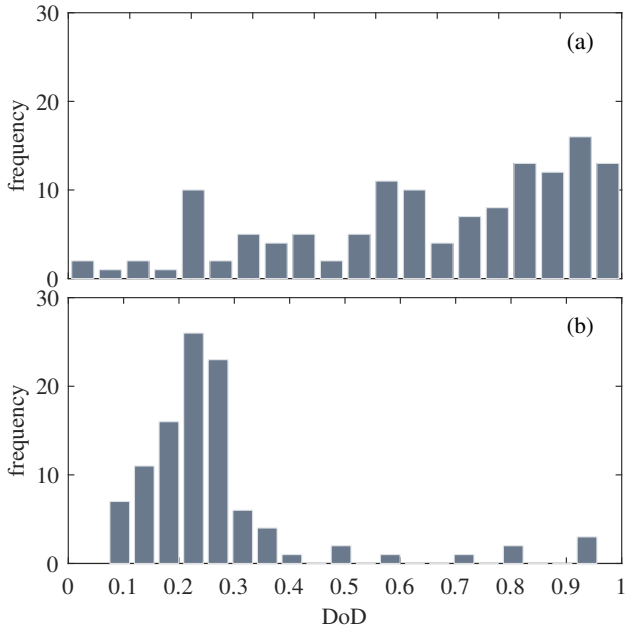

Figure 5: Cycling frequencies for optimal operation following the spot price in Fig. 4. (a) Without considering aging costs. (b) Considering aging costs with ESS CAPEX equal to 125 €/kWh.

the optimal solution would be not to cycle the BESS at all. But in that case the income would be obviously null. In between these two extreme options, the aging routine produces a cost figure that is subtracted dynamically from the profits of the first case. As a result, see Fig. 5b, the algorithm produces a sequence that allows cycling the BESS in an optimal "region," which mostly produces cycles between 10% and 40% DoD. That is, the algorithm proposes a cycling that would optimize the use of the curve represented in Fig. 3, without excessively reducing the BESS profitability from arbitraging.

4.2. Need for a stochastic approach

The results shown in Fig. 4 were based on a deterministic analysis, because we employed one only price realization. This provided clarity in analyzing the operation regime. However, in what follows we show that that simplified approach is not adequate when we consider the aging mechanism under relatively high ESS CAPEX prices.

Fig. 6 shows the optimal operation policies for energy arbitrage following the Swissgrid spot prices of January, 2017. (January 1, 2017, was a Sunday.) We conducted the analysis under different CAPEX assumptions, ranging from null to a relatively high CAPEX. We employed 50 samples of simulated prices.

First, when the CAPEX of the ESS is not considered—or in other words the aging is neglected because the cost of reposition is not introduced into the analysis—the arbitrage is carried on continuously. The algorithm proposes vigorous charges of energy on Saturdays and Sundays, to proceed with progressive discharges during the weekdays, punctuated by small intermediate charges at night lower prices (see Figs. 6a and 6b). The algorithm finds no reason to change the pattern of weekly arbitrage, even in the presence of lower price volatility and level spotted during the first two weeks, because the ESS aging is supposed not to be affected by the continuous operation.

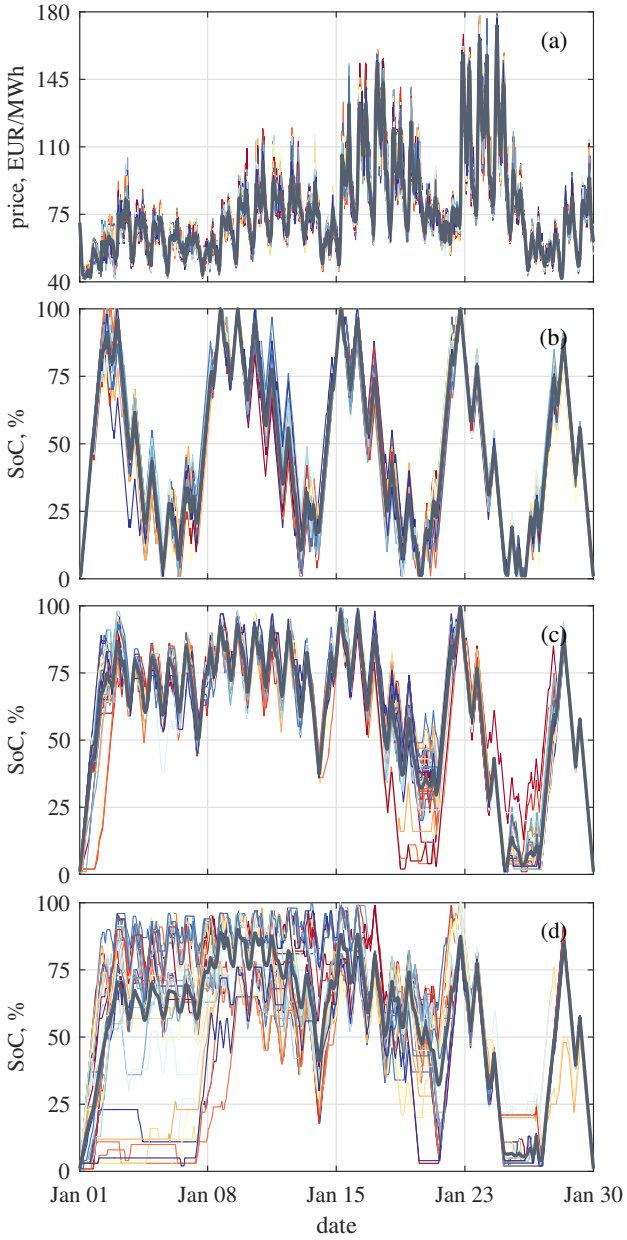

Figure 6: Impact of ESS CAPEX value on the operation decisions under uncertainty. $K = 50$ samples. (a) Swissgrid spot price in January, 2017. (b) Null CAPEX. (c) CAPEX equal to 50 €/kWh. (d) CAPEX equal to 100 €/kWh

In Fig. 6c the situation is visibly different, however. The degradation of the ESS does influence the definition of the optimal policy in the form of a price penalty. As a result, a preference for arbitrage through the last two weeks is observed; mostly visible over the last week, of largest price volatility. In those final weeks lower SoCs are tolerated, despite the reduction in the life expectancy that would ensue from such deep discharges. The expected profit seems to worth it. But differently, the first two weeks do not offer as much a profit as to make up for the ESS degradation, and thus the operation is kept above 50% SoC. Compared to the optimal policy shown in Fig. 6b, now in Fig. 6c the total payoff is reduced because of the

reduction of the amount of exchanged energy.

Whereas the difference between the optimal policies depicted in Figs. 6b and 6c lies mainly in the SoC level, in Fig. 6d the difference is more related to the spread of possible solutions. This latter is a case of relatively high CAPEX, meaning that the operation of the ESS is appreciably sensitive to the price trajectories. This is particularly evident in the left part of the panel, corresponding with the two less attractive weeks from the point of view of arbitrage. The proposed operation implies in some cases maintaining levels above the 75% SoC to evade a costly degradation. In other cases, the ESS is barely used over the first week, and it remains almost discharged, without cycling. This is in evident contrast with the the case depicted in Fig. 6b, where it might have been argued that the complications introduced by a stochastic formulation of the problem might not be worth the accuracy achieved. Prices might have been treated deterministically, using only the original series and not the 50 different price paths, because (at least visually) it does not seem to exist much difference between different trajectories. But our point here is that such a deterministic approach would be inaccurate when the ESS profitability come closer to the break even. A twofold increase of the CAPEX with respect to the value in Fig. 6c, would make the optimal path trajectory more uncertain, as shown in Fig. 6d. Some price paths that in practice were considered to be indistinguishable from others in the cases Fig. 6b and Fig. 6c now have a large impact on the decisions taken in Fig. 6d. Eventually it means that selecting one unique price path—the deterministic approach—could entail a high valuation error.

4.3. Overall performance of the algorithm, including aging impact

The previous two analyses showed different arbitrage opportunities, which the optimization algorithm solves step by step. This Section shows a comparison of those two cases, along with four additional scenarios. We try to cover different European markets to show how the algorithm adapts the solution to different structural contents in the input series. The analysis concentrates on the characteristic operation policy identified by the algorithm, but does not try to infer definite conclusions about the arbitrage chances in European electricity market. Indeed, these are one-month data, which cannot be considered representative of long-term operation.

The first series analyzed in Fig. 4a, corresponding to Fingrid, exhibits a reduced trend component, a relatively reduced seasonality, and some large price spikes. It is similar to NO1 (Norway) and Litgrid, for instance. As we discussed previously, the arbitrage in this case is based on capitalizing the relatively strong fluctuations in the price. The results are given under the labels Fingrid 1 and 2 in Fig. 8a. Noticeably in this case, the introduction of the cost of aging approximately halves the mean payoffs. This is a characteristic result of this series, and it does not occur in the remaining cases. The reason, as a detailed observation of the algorithm results attests, is the uncertainty about the peak locations. They do not seem to occur at predictable dates. This is accounted for by the regression over different simulated paths (line 8, Algorithm 1), which estimates the conditional expectation of the multiple continuation options before calculating the decisions about operation. As a result, the payoff is reduced more importantly than in the other cases. Indeed, if we consider the deterministic simulation analyzed in Fig. 4, the payoff reduces only about 30% compared to the case in which the cost of aging is not introduced. In this case, the uncertainty would have

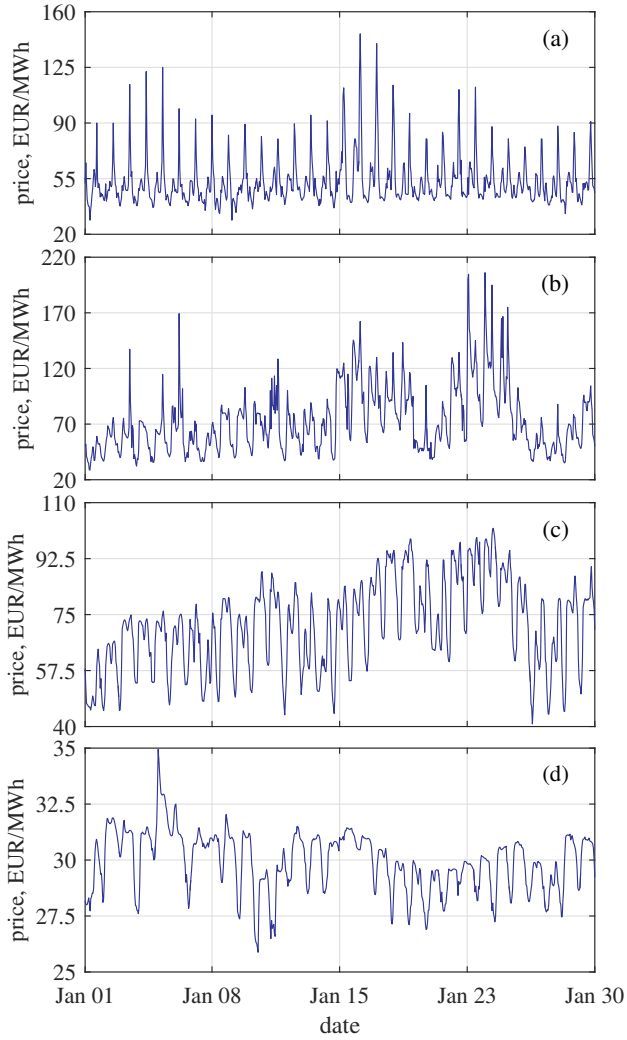

Figure 7: From top to bottom, National Grid, Elia, REN, and NO5 (Norway).

been removed, with a precise location of the spikes, giving an inaccurate valuation of the ESS arbitrage performance. This further serves to demonstrate the need for a stochastic approach.

The second series, Swissgrid, was already used in Fig. 6, and it is an example of strong daily and weekly seasonality, as well as a more defined level trend. IT-North and RTE are other similar examples in Europe. The results are more favorable now than for Fingrid, from the arbitrage point of view. This is a consequence of the noticeably larger number of fluctuations. These oscillations, compared to Fingrid, make the payoff solutions spread over a wider range (Fig. 8a). This is an expected consequence of the similarly wider range of prices with frequent repetition. Here the algorithm finds more variety in arbitrage opportunities and produces a higher mean payoff.

The case of National Grid, Fig. 7a, is in some aspects similar to Fingrid regarding the large price variations. But differently, it shows a strong seasonality in the price peaks. This makes the algorithm yield a distinct solution. First, the idle time, Fig. 8b, is reduced. From 123 out of 720 hours in Fingrid to 70 now

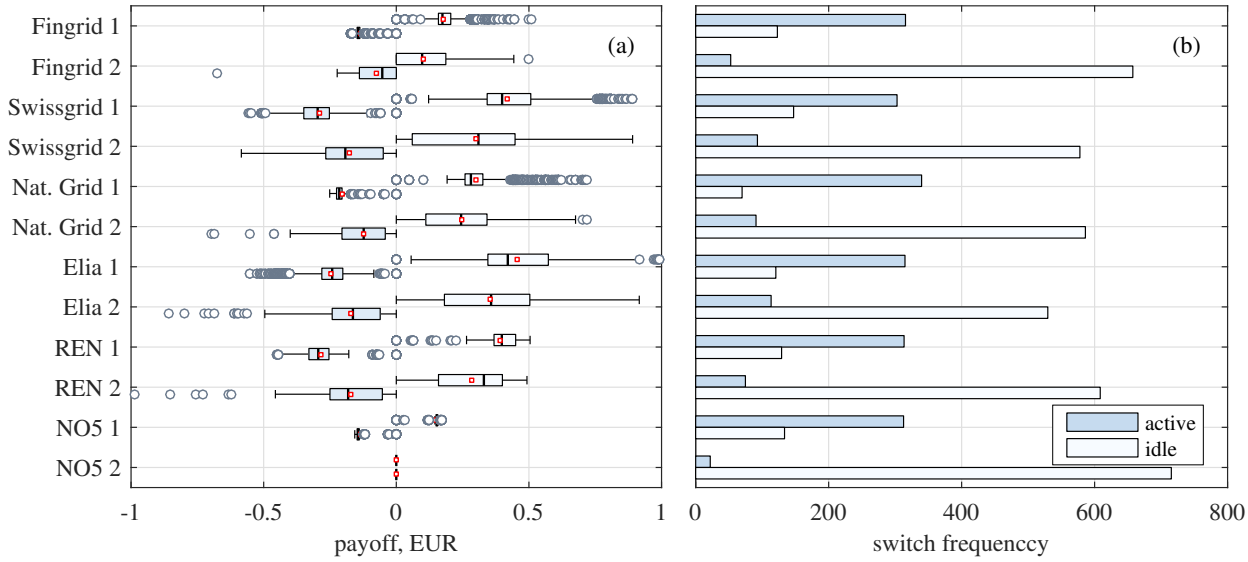

Figure 8: Results from the analysis of 50-sample simulations of the spot electricity prices shown in Figs. 4a, 6a, and 7. For each spot market, option 1 means *without* using the aging penalization and 2 *with* it. The size of the ESS was 100 kWh and the CAPEX 50 €/kWh. (a) Distribution of payoffs. Positive and negative payoffs are sparated into light and dark boxplots, respectively. The red square in each box stands for the distribution mean. (b) Frequency of active (either charge or discharge, because the number of events must be the same) or idle operation.

in National Grid: almost a 50% reduction. This is an obvious, direct consequence of the larger amount of spikes in National Grid. Indeed, the Fingrid's case demonstrates exceptional high idle times when the cost of aging is incorporated, as a consequence that only the few large price spikes make up for the ESS activation. Secondly, the reduction of the mean payoff when the cost of aging is incorporated is not so prominent (Fig. 8a). Here the reduced uncertainty in the spike locations plays a key role.

Elia's time series in Fig. 7b is similar to Swissgrid's. However, its calibration returns a slightly higher volatility in its error component. Or in other words, the simulated paths differ more from the original series than in Swissgrid's case. As a consequence, the payoff results are actually similar, when the algorithm cancels out most of the deviations through the conditional expectation analysis at the decision points.

The series plotted in Fig. 7c, corresponding to REN and similar to REE and Transelectrica, is characterized by a distinct local level variation. Also a clear daily seasonality defines its structure. Unlike Swissgrid and Fingrid, however, its weekly seasonality is less perceptibly. Finally, there are no spikes, but more sustained prices at high levels. Again the solution provided by the algorithm agrees with the impressions obtained after a visual inspection of the input series. Except for some exceptions, the differences between high and low prices are fairly constant every week day, around 20 €. This actually makes the positive and negative payoff distributions concentrated toward the means, if aging is not considered (Fig. 8a). (Only Fingrid, which has a remarkably different price structure, has a similar payoff concentration.) In that case, the solution proposed by the algorithm clearly agrees with the characteristics of the time series. Differently, when the cost of aging is considered, the optimal payoffs describe a lower, more even distribution due to the DoD and cycle number limitations. In this case, the algorithm identifies the level trend as a major player in the arbitrage process, with the evenly 20 € signals somehow distorted by those limitations.

The last case, NO5 of Norway in Fig. 7d, is a particular situation with constant prices. They follow variations of about 5 €. This means that these prices hardly qualify for arbitrage purposes. Indeed, it is a case in which even without considering the ESS aging the profits obtained are comparatively reduced: the mean discharging (sale) payoff is $+1.40$ c€, and the charging (purchase) payoff is -1.34 c€. The values are the lowest when compared to any other case. And as expected, when the aging is introduced, the small variations in prices makes it almost impossible to compensate for the negative payoff related to the cost of aging; effectively resulting in almost null activity.

Elia's data produced good arbitrage results, compared to the other analyzed series. Without considering aging, its mean payoffs were $+46$ and -24 c€, with 314 activations over the 744 hours. A quick calculation indicates that the mean total payoff would be $314 \times (46 - 24) = 69$ €/month. A similar calculation when the cost of aging is included would give $114 \times (36 - 17) = 22$ €. (This is an approximated computation from the results shown in Fig. 8, which nonetheless approximates well the results obtained from the algorithm: 65.8 and 21.0 €, respectively.) On the cost side, for all the analyzed cases the CAPEX was 50 €/kWh, and the ESS capacity was 100 kWh. Assuming an arbitrary WACC equal to 6% over 10 years, the capital recovery factor would be 0.136. Therefore, the equivalent annuity would be 679.5 €/year, which could be prorated to 56.6 €/month, making it clear that the introduction of replacement costs reduces the profitability below the initial CAPEX.

Of course this is but a rough calculation because it implies arbitrary assumptions, such as the WACC, the CAPEX, and the investment duration. They are reasonable assumptions, but they are not grounded on a detailed assessment of prices. Also, it is assumed that the price structure will remain invariable over time, which is arguable as well. But the main point here is to show how the introduction of aging costs dramatically reduces the chances of profitability. The above simple calculation shows that the error due to not introducing the aging costs in an investment analysis may be large, especially if we consider that the above CAPEX figure is particularly low by actual standards.

5. Conclusions

We have proposed and discussed a dynamic programming approach for valuing ESS, with a particular focus on grid applications, which shows distinctive features with respect to other works published on the subject. A first feature is that this approach addresses the optimization of charge/discharge decisions under aging penalties. The difference with respect to other works lies in that the aging cost from a decision is calculated dynamically (ex-ante), thus being incorporated into the decision evaluation. Second, this approach tackles the imperfect payoff foresight that ensues from operating the ESS in a stochastic framework, in which prices and energy demand are uncertain. Again, this feature is ex-ante integrated into the decision evaluation. Finally, the practical implementation is also singular, consisting in slicing the problem into unique decision matrices that are conveniently managed by means of shifting and re-indexing operators, leading to a solution that can cope with the huge amount of decision evaluations in an efficient way.

The program is remarkably flexible, because it is compartmentalized into several blocks. One of them is the payoff calculation, which is also split into several dedicated functions. Such functions must compute the payoff from operating in the market, the cost of aging, and indeed any other additional term that may

well have an impact into the payoff formation. The only requisite they have is that they return a payoff value when queried about unique operation decisions. We have employed for simplicity in the discussion an energy arbitrage policy, where for instance the function must return the payoff from a decision to sell/purchase energy subject to the electricity instant price. But other stream revenues can be analyzed as well, where more sophisticated payoff calculations are employed based not only on electricity prices but on load/generation profiles or EV driving results. Similarly, the aging payoff calculation responds to the same structure. We have employed a model based on maximum life as a function of the number of cycles at different DoD. Yet other models are acceptable as long as they yield the consumed life as a function of the number of cycles and idle time; which again the dedicated proposed algorithm computes as a function of the operation decision.

Our analysis of energy arbitrage revenues in different European markets, with different structural components, shows on the whole the need for including uncertainty and aging in the valuation of ESS. This is particularly important when the CAPEX value is increased. In fact, we have found that under some circumstances of maximum life cycle and market price structure, the errors from not considering uncertainty and aging can be readily perceptible for CAPEX above 50 €/kWh, which is a relatively low figure by current CAPEX values. These results must be interpreted with caution, nonetheless, because they are limited in scope (one month data of several selected European markets). As we have emphasized before, the analyses have been performed to illustrate the methodology and demonstrate the changes in the optimal operation of the BESS when aging is considered.

References

[1] Lazard, Levelized Cost of Storage Analysis 2.0, Technical Report, 2016.

[2] J. A. Taylor, D. S. Callaway, K. Poolla, Competitive energy storage in the presence of renewables, IEEE Transactions on Power Systems 28 (2013) 985–996.

[3] I. Pawel, The cost of storage - How to calculate the levelized cost of stored energy (LCOE) and applications to renewable energy generation, in: Energy Procedia, volume 46, pp. 68–77.

[4] C. S. Lai, M. D. McCulloch, Levelized cost of electricity for solar photovoltaic and electrical energy storage, Applied Energy 190 (2017) 191–203.

[5] K. Bradbury, L. Pratson, D. Patiño-Echeverri, Economic viability of energy storage systems based on price arbitrage potential in real-time U.S. electricity markets, Applied Energy 114 (2014) 512–519.

[6] H. Zhao, Q. Wu, S. Hu, H. Xu, C. N. Rasmussen, Review of energy storage system for wind power integration support, Applied Energy 137 (2014) 545–553.

[7] V. Jülch, Comparison of electricity storage options using levelized cost of storage (LCOS) method, Applied Energy 183 (2016) 1594–1606.

[8] K. Divya, J. Østergaard, Battery energy storage technology for power systems—An overview, Electric Power Systems Research 79 (2009) 511–520.

[9] R. S. Go, F. D. Munoz, J. P. Watson, Assessing the economic value of co-optimized grid-scale energy storage investments in supporting high renewable portfolio standards, Applied Energy 183 (2016) 902–913.

[10] A. Sumper, Impact of operation strategies of large scale battery systems on distribution grid planning in Germany, 2017.

[11] M. Broussely, Aging Mechanisms and Calendar-Life Predictions in Lithium-Ion Batteries, Advances in Lithium-Ion Batteries (2002) 293–432.

[12] V. Agubra, J. Fergus, Lithium ion battery anode aging mechanisms, 2013.

[13] H. Wang, Y.-i. Jang, B. Huang, D. R. Sadoway, Y.-m. Chiang, TEM Study of Electrochemical Cycling-Induced Damage and Disorder in LiCoO2 Cathodes for Rechargeable Lithium Batteries, Journal of The Electrochemical Society 146 (1999) 473–480.

[14] J. Li, E. Murphy, J. Winnick, P. A. Kohl, Studies on the cycle life of commercial lithium ion batteries during rapid charge-discharge cycling, Journal of Power Sources 102 (2001) 294–301.

[15] P. Ruetschi, Aging mechanisms and service life of lead-acid batteries, Journal of Power Sources 127 (2004) 33–44.

[16] J. Wang, J. Purewal, J. Graetz, S. Soukiazian, H. Tataria, M. W. Verbrugge, Degradation of lithium ion batteries employing graphite negatives and nickel-cobalt-manganese oxide + spinel manganese oxide positives: Part 2, chemical-mechanical degradation model, Journal of Power Sources 272 (2014) 1154–1161.

[17] K. R. Khalilpour, A. Vassallo, A generic framework for distributed multi-generation and multi-storage energy systems, Energy 114 (2016) 798–813.
[18] R. Ahmed, M. El Sayed, I. Arasaratnam, J. Tjong, S. Habibi, Reduced-Order Electrochemical Model Parameters Identification and SOC Estimation for Healthy and Aged Li-Ion Batteries. Part I: Parameterization Model Development for Healthy Batteryies, IEEE Journal of Emerging and Selected Topics in Power Electronics 2 (2014) 659–677.
[19] J. Schiffer, D. U. Sauer, H. Bindner, T. Cronin, P. Lundsager, R. Kaiser, Model prediction for ranking lead-acid batteries according to expected lifetime in renewable energy systems and autonomous power-supply systems, Journal of Power Sources 168 (2007) 66–78.
[20] B. Xu, A. Oudalov, A. Ulbig, G. Andersson, D. Kirschen, Modeling of Lithium-Ion Battery Degradation for Cell Life Assessment, IEEE Transactions on Smart Grid (2016) 1–1.
[21] A. Assunção, P. S. Moura, A. T. Almeida, Technical and economic assessment of the secondary use of repurposed electric vehicle batteries in the residential sector to support solar energy, Applied Energy 181 (2016) 120–131.
[22] G. He, Q. Chen, C. Kang, P. Pinson, Q. Xia, Optimal Bidding Strategy of Battery Storage in Power Markets Considering Performance-Based Regulation and Battery Cycle Life, IEEE Transactions on Smart Grid 7 (2016) 2359–2367.
[23] X. Ke, N. Lu, C. Jin, Control and Size Energy Storage Systems for Managing Energy Imbalance of Variable Generation Resources, IEEE Transactions on Sustainable Energy 6 (2015) 70–78.
[24] A. Stoppato, G. Cavazzini, G. Ardizzon, A. Rossetti, A PSO (particle swarm optimization)-based model for the optimal management of a small PV(Photovoltaic)-pump hydro energy storage in a rural dry area, Energy 76 (2014) 168–174.
[25] R. Dufo-López, E. Pérez-Cebollada, J. L. Bernal-Agustín, I. Martínez-Ruiz, Optimisation of energy supply at off-grid healthcare facilities using Monte Carlo simulation, Energy Conversion and Management 113 (2016) 321–330.
[26] M. Naumann, R. C. Karl, C. N. Truong, A. Jossen, H. C. Hesse, Lithium-ion battery cost analysis in PV-household application, Energy Procedia 73 (2015) 37–47.
[27] S. Moazeni, W. B. Powell, A. H. Hajimiragha, Mean-Conditional Value-at-Risk Optimal Energy Storage Operation in the Presence of Transaction Costs, IEEE Transactions on Power Systems 30 (2015) 1222–1232.
[28] W. W. Kim, J. S. Shin, S. Y. Kim, J. O. Kim, Operation scheduling for an energy storage system considering reliability and aging, Energy 141 (2017) 389–397.
[29] R. Dufo-López, J. L. Bernal-Agustín, Techno-economic analysis of grid-connected battery storage, Energy Conversion and Management 91 (2015) 394–404.
[30] M. Majidi, S. Nojavan, K. Zare, Optimal stochastic short-term thermal and electrical operation of fuel cell/photovoltaic/battery/grid hybrid energy system in the presence of demand response program, Energy Conversion and Management 144 (2017) 132–142.
[31] M. Zheng, C. J. Meinrenken, K. S. Lackner, Smart households: Dispatch strategies and economic analysis of distributed energy storage for residential peak shaving, Applied Energy 147 (2015) 246–257.
[32] X. Chen, T. Wei, S. Hu, Uncertainty-aware household appliance scheduling considering dynamic electricity pricing in smart home, IEEE Transactions on Smart Grid 4 (2013) 932–941.
[33] J. Li, M. A. Danzer, Optimal charge control strategies for stationary photovoltaic battery systems, Journal of Power Sources 258 (2014) 365–373.
[34] K. Abdulla, J. de Hoog, V. Muenzel, F. Suits, K. Steer, A. Wirth, S. Halgamuge, Optimal Operation of Energy Storage Systems Considering Forecasts and Battery Degradation, IEEE Transactions on Smart Grid (2016) 1–1.
[35] F. A. Longstaff, E. S. Schwartz, Valuing American options by simulation: a simple least-squares approach, Review of Financial Studies 14 (2001) 113–147.
[36] G. Díaz, B. Moreno, J. Coto, J. Gómez-Aleixandre, Valuation of wind power distributed generation by using Longstaff-Schwartz option pricing method, Applied Energy 145 (2015) 223–233.
[37] M. Musallam, C. M. Johnson, An efficient implementation of the rainflow counting algorithm for life consumption estimation, IEEE Transactions on Reliability 61 (2012) 978–986.
[38] R. E. Ciez, J. Whitacre, Comparative techno-economic analysis of hybrid micro-grid systems utilizing different battery types, Energy Conversion and Management 112 (2016) 435–444.
[39] F. Wankmüller, P. R. Thimmapuram, K. G. Gallagher, A. Botterud, Impact of battery degradation on energy arbitrage revenue of grid-level energy storage, Journal of Energy Storage 10 (2017) 56–66.
[40] R. L. Fares, M. E. Webber, What are the tradeoffs between battery energy storage cycle life and calendar life in the energy arbitrage application?, Journal of Energy Storage 16 (2018) 37–45.
[41] S. Bashash, S. J. Moura, J. C. Forman, H. K. Fathy, Plug-in hybrid electric vehicle charge pattern optimization for energy cost and battery longevity, Journal of Power Sources 196 (2011) 541–549.
[42] J. Wu, X. Wang, L. Li, C. Qin, Y. Du, Hierarchical control strategy with battery aging consideration for hybrid electric vehicle regenerative braking control, Energy 145 (2018) 301–312.
[43] A. Forough, R. Roshandel, Lifetime optimization framework for a hybrid renewable energy system based on receding horizon optimization, Energy 150 (2018).
[44] ASTM, E1049-85 Standard Practices for Cycle Counting in Fatigue Analysis, Technical Report, ASTM, 2011.
[45] S. T. Rachev, Y. S. Kim, M. L. Bianchi, F. J. Fabozzi, CFA, Financial Models with Levy Processes and Volatility Clustering, John Wiley & Sons, 2011.
[46] R. Bellman, Dynamic Programming, Courier Corporation, 2013.
[47] M. A. Tankari, M. B. Camara, B. Dakyo, G. Lefebvre, Use of Ultracapacitors and Batteries for Efficient Energy Management in Wind–Diesel Hybrid System, IEEE Transactions on Sustainable Energy 4 (2013) 414–424.
[48] A. Jaafar, C. Akli, B. Sareni, X. Roboam, A. Jeunesse, Sizing and Energy Management of a Hybrid Locomotive Based on Flywheel and Accumulators, IEEE Transactions on Vehicular Technology 58 (2009) 3947–3958.

[49] T. G. Kolda, B. W. Bader, Tensor Decompositions and Applications, SIAM Review 51 (2009) 455–500.
[50] K. Liu, K. Li, H. Ma, J. Zhang, Q. Peng, Multi-objective optimization of charging patterns for lithium-ion battery management, Energy Conversion and Management 159 (2018) 151–162.