Self-Improving Online Storage Control for Stable Wind Power Commitment

Chenbei Lu, *Graduate Student Member, IEEE*, Hongyu Yi, *Graduate Student Member, IEEE*, Jiahao Zhang, and Chenye Wu, *Senior Member, IEEE*

***Abstract*—The integration of distributed energy resources, particularly wind energy, presents both opportunities and challenges for the modern electrical grid. On the supply side, wind farms frequently encounter penalties due to wind power's intermittency and variability. The incorporation of energy storage systems can mitigate these penalties through real-time power adjustments. However, the uncertainties in future renewable generation significantly impede optimal storage control, and existing algorithms either lack theoretical guarantees, or fail to effectively leverage data to attain better performance. This paper effectively addresses this dichotomy by bridging the gap between data utilization and theoretical guarantees based on the Markov decision process. Specifically, we first introduce a one-shot online storage control algorithm that utilizes historical data to make near-optimal decisions with theoretical performance guarantees. To further enable continuous learning from new data, we develop an online learning-based self-improving storage control algorithm, underscoring its asymptotic optimality. The numerical study using field data demonstrates the efficacy of the proposed approach.**

***Index Terms*—Distributed energy resource, wind power, storage control, online optimization.**

I. Introduction

With the widespread grid integration of distributed energy resources, especially renewable energies, both the power shortage in the grid and the carbon-related ecological challenges are well alleviated. However, the inherent intermittency and unpredictability of renewable sources [1] present considerable threats to the grid's stability and reliability, leading to heightened operational costs for the power system.

Wind farms, as wind power suppliers integrated into the power grid, are striving to ensure a stable and predictable power supply. Specifically, wind farms often undertake power supply contracts with the bulk grid's independent system operator (ISO) [2]. The contract specifies the timing and quantity of electricity that should be delivered to the grid, which is usually determined based on wind power forecasts. However, real-time wind power generation frequently diverges from these forecasts, resulting in a disparity between the committed and actual power generation. Consequently, wind farms are subjected to penalty costs for this deviation, which are paid to the grid for the procurement of real-time energy balancing services.

To alleviate these penalties, wind farms can employ energy storage systems (ESS) to stabilize the wind power output. Specifically, at any given time, the ESS can either be charged by wind power or discharged to the grid, which can effectively regulate the amount of the delivered wind power. However, in practice, the uncertainties in both wind power generation and real-time penalty costs considerably complicate effective storage control. Moreover, the physical constraints of the ESS, i.e., the storage capacity, and the charging and discharging power limits, prevent the complete containment of energy imbalances.

To tackle these challenges, in this paper, we propose an online storage control algorithm based on the Markov decision process (MDP), which can effectively learn from historical data and make the nearly-optimal storage control decisions with theoretical guarantees. To facilitate continuous learning from new data, we further design a self-improving online storage control algorithm based on the online learning scheme, and we theoretically demonstrate its asymptotical optimality.

A. Related Works

The problem of storage system operation has attracted increasing interest in the power community. Most recent works mainly focus on utilizing storage systems to improve power system operation. For example, Mahmoodi et al. introduce a distributed economic dispatch strategy for microgrids with multiple ESS in [3]. Shi et al. design a control algorithm for a battery storage system for simultaneous peak shaving and frequency regulation through a joint optimization framework in [4]. Uddin et al. propose a decision-tree-based algorithm for generation scheduling and storage control in the islanded microgrids [5]. These works mostly focus on day-ahead storage scheduling, whereas our study addresses online storage control for stable wind power commitment.

Manuscript received 21 July 2023; revised 26 October 2023; accepted 1 January 2024. Date of publication 8 January 2024; date of current version 21 June 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 72271213, and in part by the Shenzhen Science and Technology Program under Grant JCYJ20220530143800001, Grant RCYX20221008092927070, and Grant ZDSYS20220606100601002. Paper no. TSG-01112-2023. *(Corresponding author: Chenye Wu.)*

Chenbei Lu is with the Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing 100084, China.

Hongyu Yi, Jiahao Zhang, and Chenye Wu are with the School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen 518172, Guangdong, China (e-mail: chenyewu@yeah.net).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSG.2024.3350895.

Digital Object Identifier 10.1109/TSG.2024.3350895

The online storage control problem has garnered substantial research interest lately due to its complexity and significance for the modern power grid. Among notable studies, Malysz et al. present an online storage control method based on model predictive control (MPC) with a mixed-integer-linear-programming (MILP) optimization formulation in [6]. Dabbagh et al. design an efficient storage control strategy based on MPC assuming limited-horizon future knowledge in [7]. Besides the MPC-based approach, Wu et al. design a threshold-based storage control policy integrating the forecast of future information in [8], [9]. These works aim at integrating wind power prediction to improve the performance of storage control. However, they often lack theoretical performance guarantees and are highly sensitive to prediction accuracy. In contrast, our approach does not rely on any future prediction and provides asymptotic theoretical guarantees.

There are also various works targeted to design the storage control algorithm with theoretical performance guarantees. For example, Koutsopoulos et al. propose a threshold-based control policy for storage control assuming that the power demand arrival and service processes follow the Poisson process in [10]. Chau et al. further design a provable threshold-based online storage control algorithm without additional information about future demand in [11]. Besides the threshold-based online algorithm, Lyapunov optimization is often leveraged for storage control. For example, Huang et al. design a joint demand response and ESS management algorithm for a power-consuming entity based on Lyapunov drift-plus-penalty optimization in [12]. Qin et al. propose an online modified greedy algorithm based on the Lyapunov optimization for storage control on the demand side with sub-optimality bound analysis in [13], [14]. Zhong et al. design an online control approach for real-time distributed ESS sharing based on the Lyapunov optimization framework in [15]. However, these works mainly focus on the storage control problem for the demand side, and do not utilize historical data. In practice, effective exploitation of historical data helps design more powerful online algorithms. To this end, in this paper, we propose a data-driven online storage control algorithm based on MDP. To the best of our knowledge, only very limited works focus on the storage control based on MDP [16], [17]. And these works often have strict requirements on the time length of the storage control process, and lack theoretical guarantees. In contrast, we design a one-shot decision algorithm without the time length requirement, and include an adjustable parameter to trade-off the short-term and long-term interests. Also, our algorithm can leverage both the historical data and the continuously collected new data to improve the performance with theoretical guarantees.

B. Our Contributions

Our major contributions can be summarized as follows:

- *One-shot Decision-making Algorithm Design in Ideal Condition*: We propose a computationally efficient one-shot decision framework for online storage control. We also utilize MDP to optimally design the storage control algorithm with perfect historical information.

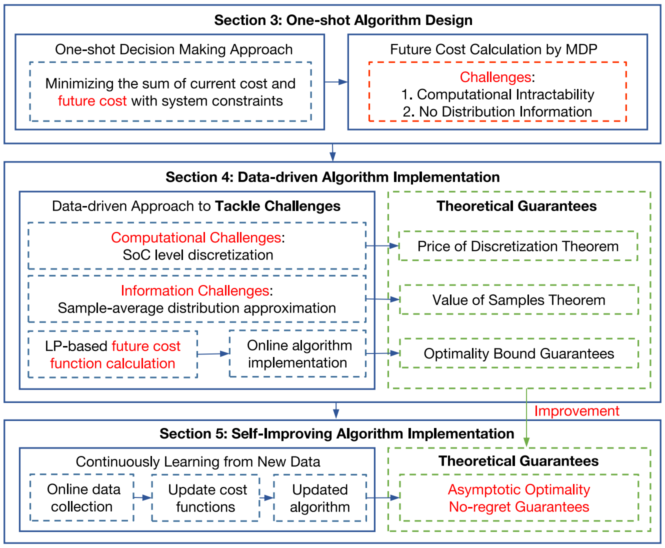

Section 3: One-shot Algorithm Design
One-shot Decision Making Approach
Minimizing the sum of current cost and future cost with system constraints
Future Cost Calculation by MDP
Challenges:
1. Computational Intractability
2. No Distribution Information
Section 4: Data-driven Algorithm Implementation
Data-driven Approach to Tackle Challenges
Computational Challenges:
SoC level discretization
Information Challenges:
Sample-average distribution approximation
LP-based future cost function calculation
Online algorithm implementation
Theoretical Guarantees
Price of Discretization Theorem
Value of Samples Theorem
Optimality Bound Guarantees
Improvement
Section 5: Self-Improving Algorithm Implementation
Continuously Learning from New Data
Online data collection
Update cost functions
Updated algorithm
Theoretical Guarantees
Asymptotic Optimality
No-regret Guarantees

Fig. 1. Structural Diagram.

- *Data-driven Algorithm Implementation with Theoretical Guarantees*: We propose a practical approach to implementing the one-shot decision algorithm with limited data and computational resources. We also characterize how limited data and discretization will influence the accuracy of our algorithm, respectively. In addition, the regret bound of our algorithm is derived.
- *Self-improving Algorithm Design with Theoretical Guarantees*: We propose a self-improving one-shot decision algorithm, which can continuously utilize the new data during the control process to improve the algorithm performance. We prove that the self-improving algorithm can achieve a sub-linear regret with asymptotic optimality.

The remainder of this paper is organized as follows: Section II introduces the storage control problem. Section III proposes the one-shot decision algorithm based on MDP. Section IV implements the data-driven one-shot decision algorithm and provides theoretical guarantees for the proposed algorithm. Section V further adopts the notion of online learning to improve the one-shot decision algorithm with a no-regret performance guarantee. Section VI evaluates the performance of our proposed approaches, and Section VII concludes our paper. The structural diagram of our proposed algorithm is visualized in Fig. 1. All necessary proof sketches are provided in the Appendix.

II. System Models

Consider a wind farm that delivers power to the grid based on an electricity supply contract. Specifically, the committed wind power supply amount at time t is $\hat{w}_t$. The committed wind power supply is determined by the wind power forecast from the wind farm or the ISO. Such forecast can be done from minute-ahead to day-ahead [18], [19], [20]. And in real-time, the actual wind power generation w_t sequentially reveals at each time t. With the equipment of storage, the wind farm can either charge the storage with amount u_t^+ by wind power,

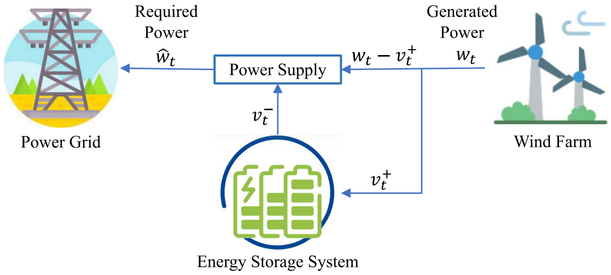


Fig. 2. System Structure.

or discharge the storage with amount u_t^- to the grid at time t. Consequently, the eventually delivered power g_t satisfies:

$$g_t = w_t + v_t^- - v_t^+. \tag{1}$$

When the mismatch between commitment $\hat{w}_t$ and the delivered power g_t exists, the wind farm will be charged for a penalty cost[1] $c_t(\hat{w}_t, g_t)$ as follows:

$$c_t\big(\hat{w}_t, g_t\big) = p_t^+ \max\big(g_t - \hat{w}_t, 0\big) + p_t^- \max\big(\hat{w}_t - g_t, 0\big), \tag{2}$$

where p_t^+ and p_t^- denote the unit penalty prices for wind power generation surplus and shortage at time t, respectively. The system model is visualized in Fig. 2.

Remark: With the growing development of wind power prediction technologies [21], the real-time energy mismatch between prediction (the day-ahead commitment power) and the real-time wind power generation can be well reduced. And the performance of most storage control algorithms will be promoted. In contrast, we focus on the case that the prediction error still exists and leads to the energy mismatch between commitment power and real generation power.

The wind farm aims to minimize the accumulated mismatch penalty across all T time slots by reasonably utilizing the storage system. Mathematically, the storage control problem can be formulated as follows:

$$\begin{aligned}
\textbf{(P1)} \quad \min_{v_t^+, v_t^-, \forall t} \quad & \sum_{t=1}^{T} c_t\big(\hat{w}_t, g_t\big) && (3)\\
\text{s.t.} \quad & g_t = w_t + v_t^- - v_t^+, \forall t, && (4)\\
& SoC_1 = \frac{C}{2}, && (5)\\
& SoC_{t+1} = SoC_t + \eta^+ v_t^+ - \eta^- v_t^-, \forall t, && (6)\\
& \eta^+ v_t^+ \le C - SoC_t, \forall t, && (7)\\
& \eta^- v_t^- \le SoC_t, \forall t, && (8)\\
& v_t^+ \le w_t, \forall t, && (9)\\
& v_t^+, v_t^- \ge 0, \forall t, && (10)\\
& v_t^+ v_t^- = 0, \forall t. && (11)
\end{aligned}$$

In the optimization problem, the decision variables at time t include:

- v_t^+: generated wind power that is charged to the energy storage;
- v_t^-: discharged energy from the energy storage to the grid;

And the other functions, latent variables, and system parameters include:

- $c_t(\cdot)$: penalty cost at time t;
- $\hat{w}_t$: committed wind power supply at time t;
- w_t: wind power generation at time t;
- T: duration for storage control decisions;
- p_t^+, p_t^-: unit grid penalty prices for power generation shortage and surplus at time t, respectively;[2]
- g_t: actual supplied power at time t;
- SoC_t: state-of-charge (SOC) of storage at time t;
- C: energy storage capacity;
- η^+, η^-: charging and discharging efficiencies of storage;

Constraint (4) characterizes the delivered power; constraints (5) and (6) describe the dynamics of storage; and constraints (7) and (8) represent the storage capacity limits. Constraint (9) and (10) indicate the upper and lower limits of storage control actions, and constraint (11) ensures that the storage cannot be charged and discharged simultaneously.

The challenges for tackling this problem are twofold. First, the optimization problem implicitly involves integer decision variables in constraint (11), making our problem an MILP. In practice, the time length T can be very large, and solving such a large-scale MILP may face significant computational burdens [22]. The second challenge comes from the uncertainties of the future parameters. Specifically, at any time t_0, the unit penalty prices p_t^+, p_t^-, and wind power generation w_t for all $t > t_0$ in the future are unknown, which hinders effective storage control.

To tackle these issues, in the next section, we introduce an online algorithm with one-shot decision-making in a sequential manner.

III. Online Algorithm Design: The Basics

In this section, we first formulate the one-shot online decision-making problem of storage control. Then, we introduce the notion of the storage value function and propose the framework of the one-shot decision-making algorithm. Finally, we implement the algorithm by calculating the storage value function based on MDP.

A. One-Shot Decision-Making Problem

Due to the inherent uncertainties associated with penalty prices and renewable energy generation, it is impractical to obtain all future optimal storage control decisions. Therefore, in practice, we often resort to sequential storage control. Specifically, at each time t, we determine the current storage control actions v_t^+, v_t^-, c_t based on the available information. Consequently, we establish the following one-shot storage control problem at time t as follows:

$$\begin{aligned}
\textbf{(P2)} \quad \min_{v_t^+, v_t^-} \quad & c_t\big(\hat{w}_t, g_t\big) + \sum_{\tau=t+1}^{\infty} \gamma^{\tau-t}\mathbb{E}\big(c_\tau\big(\hat{w}_\tau, g_\tau\big)\big) && (12)\\
\text{s.t.} \quad & g_t = w_t + v_t^- - v_t^+, && (13)\\
& SoC_{t+1} = SoC_t + \eta^+ v_t^+ - \eta^- v_t^-, && (14)\\
& \eta^+ v_t^+ \le C - SoC_t, && (15)
\end{aligned}$$

[1] The penalty cost of wind power mismatch can be generalized to different cases. Please refer to the Appendix for more details.

[2] They are latent variables in $c_t(\cdot)$.

$$\eta^- v_t^- \leq SoC_t, \tag{16}$$

$$v_t^+ \leq w_t, \tag{17}$$

$$v_t^+, v_t^- \geq 0, \tag{18}$$

$$v_t^+ v_t^- = 0. \tag{19}$$

We discern that, compared with **(P1)**, the constraints (13)–(19) in **(P2)** only involve one-shot decision variables at time t, which indicates a much less computational burden.

Function $c_t(\hat{w}_t, g_t)$ in the objective denotes the penalty cost at time t, and $\sum_{\tau=t+1}^{\infty} \gamma^{\tau-t}\mathbb{E}(c_t(\hat{w}_\tau, g_\tau))$ denotes the cumulative discounted penalty in the future. Note that, we adopt the temporal discount ratio $\gamma \in (0, 1)$ to characterize the wind farm's preference for short-term revenue or long-term revenue, which is a commonly adopted notion for long-term decision makings [23]. Specifically, γ is an adjustable parameter, and when γ is small, the future penalty term also becomes smaller, which means the wind farm operator is myopic. On the other hand, when γ is large, the wind farm has a long-term vision, and tends to treat the penalties of all time equally.

The one-shot decision problem **(P2)** seems to be a linear programming[3] with a significantly smaller problem scale compared with **(P1)**. However, **(P2)** is fundamentally nonlinear. This is because complex correlations exist between the current decisions at time t and the future penalty costs for $\tau > t$. Intuitively, the current decision affects the storage's SoC, which then influences the storage control decisions in the future. Additionally, the future decision problems involve the uncertain parameters w_τ, p_τ^+, and p_τ^-, which makes directly solving **(P2)** extremely challenging.

B. Reformulation Based on Storage Value Function Modeling

Despite the challenges, we can discern that, the SoC of storage bridges the one-shot decisions across different time t, which accounts for the temporal correlation. This observation allows us to reformulate the original problem **(P2)** into the following much simpler form:

$$\textbf{(P3)} \min_{v_t^+, v_t^-} \quad c_t\big(\hat{w}_t, g_t\big) + \gamma F_{t+1}^{\pi}(SoC_{t+1}) \tag{20}$$

$$F_{t+1}^{\pi}(SoC_{t+1}) = \sum_{\tau=t+1}^{\infty} \gamma^{\tau-t}\mathbb{E}\big(c_\tau\big(\hat{w}_\tau, g_\tau\big)\big), \tag{21}$$

$$\text{Constraints} \quad (13)-(19). \tag{22}$$

It can be observed that, in this problem, the objective function only consists of the current cost $c_t(\hat{w}_t, g_t)$ and future cost $\gamma F_{t+1}^{\pi}(SoC_{t+1})$, which are influenced by g_t and SoC_{t+1}, respectively. Specifically, we term F_{t+1}^{π} as the storage value function, which represents the accumulated future cost given a specific SoC. Note that, the storage value function $F_{t+1}^{\pi}(SoC_{t+1})$ is affected by both the adopted storage control policy π in the future and SoC_{t+1}. The expectation in (21) is taken over all possible p_τ^+, p_τ^-, and w_τ in the future.

Algorithm 1 One-Shot Decision Algorithm π

```
Input: Storage capacity $C$; function $F_t^{\pi}(x)$ for all $t$.
Output: Storage control policy $v_t^+$, $v_t^-$ at time $t$;
for $t = 1, 2, \ldots$ do
    Obtain parameters $w_t$, $\hat{w}_t$, $p_t^+$, $p_t^-$ in real time.
    Solve the problem (P3);
    Return the solved storage control policy at time $t$;
end for
```

Essentially, **(P3)** trades off between the current cost and the future cost by deciding g_t and SoC_{t+1}. If we accurately knew the value function $F_t^{\pi}(x)$ for all t and for all $x \in [0, C]$, we could have solved problem **(P3)** to obtain the optimal one-shot solution that minimizes the expected total cost. Algorithm 1 illustrates this one-shot decision process.

C. MDP-Based Value Function Estimation

The remaining hurdle is to determine the value function $F_t^{\pi}(SoC)$. In this part, we propose an approach to derive the value function $F_t^{\pi}(SoC)$ utilizing the tools from MDP. For simplicity, we assume the function $F_t(SoC)$ to be homogeneous for all t. This makes our algorithm time-independent, which is a common choice for many online algorithms [11], [13]. We also show how to generalize this assumption by considering heterogenous $F_t(SoC)$ for different t at the end of this section.

Before solving the value function, it is demonstrated that the one-shot decision problem **(P3)** can be formally transformed into MDP in the following manner:

Markov Decision Process $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$:

- States $\mathcal{S}$: Any state $s \in \mathcal{S}$ is composed of the penalty prices p^+, p^-, the committed and real wind power generation $\hat{w}$ and w, and SoC. Formally, $s = (p^+, p^-, \hat{w}, w, SoC)$;
- Actions $\mathcal{A}$: Any action $a \in \mathcal{A}$ is composed of two components: the charge amount v^+ and discharge amount v^-. Formally, $a = (v^+, v^-)$;
- Transition probability $\mathcal{P}$: $\mathcal{P}$ is the transiting probability matrix $\mathcal{P}_a = \{\mathbf{Pr}(s_{t+1} = s' | s_t = s, a_t = a), \forall s, s' \in \mathcal{S}, a \in \mathcal{A}\}$, which includes the probability of transiting from state s to s' with action a for all s, s' and a;
- Reward $\mathcal{R}$: $\mathcal{R}$ is the immediate reward (penalty in our case) after transiting from state s to state s' due to action a, i.e., $\mathcal{R} = \{r(s, a), \forall s, a\}$. Specifically, in the one-shot decision problem, the penalty $r(s, a)$ equals the negative of the penalty, i.e.,

$$r(s, a) = p^+ \max\big(w - \hat{w}, 0\big) + p^- \max\big(\hat{w} - w, 0\big).$$

We can observe that, for the storage control problem, the state space $\mathcal{S}$ and the action space $\mathcal{A}$ are known. The reward $\mathcal{R}$ is also known once the state s and action a are decided. The only unknown comes from the transition probability $\mathcal{P}$.

However, some important observations for $\mathcal{P}$ can simplify the problem. Specifically, we can divide the state variables $s = (p^+, p^-, \hat{w}, w, SoC)$ into one deterministic state and several random states. The deterministic state is SoC, which can be determined following Eq. (14) without any uncertainty.

[3] It should be noted that, **(P2)** still contains a nonlinear constraint in (19). However, it can be easily transformed into linear constraints by discussing the two cases $v_t^+ = 0$ and $v_t^- = 0$ separately.

And the random states include $p^+, p^-, \hat{w}, w$, which are fully random.[4] In fact, they are independent of SoC and the control decisions. The observations enable us to derive the following conditions for the value function $F(SoC)$:

$$F(SoC) = \int_{q \in \mathcal{Q}} R^*(q, SoC) f(q) dq, \forall SoC \in [0, C], \quad (23)$$

where we define q as the random state variables with $q = \{p^+, p^-, \hat{w}, w\}$. The set $\mathcal{Q}$ contains all possible q, and $f(q)$ denotes the probability density function (*pdf*) of the state q. $R^*(q, SoC)$ denotes the cumulative cost with SoC under state q, satisfying:

$$\begin{aligned} R^*(q, SoC) = \min_{v^-, v^+} c(q, g) \ + \gamma F(SoC + \Delta) \quad (24)\\ s.t. \quad \Delta = \eta^+ v^+ - \eta^- v^-, \quad (25)\\ \text{Constraints } (13)-(19), \quad (26) \end{aligned}$$

where Δ denotes the variation of SoC after storage control actions, and function $c(q, g)$ satisfies:

$$c(q, g) \ = p^+ \max\big(g - \hat{w}, 0\big) + p^- \max\big(\hat{w} - g, 0\big). \quad (27)$$

We can observe that, $R^*(q, SoC)$ bridges $F(SoC)$ with different SoC through problem (24)–(26). Note that, this estimation methodology can be easily extended to the case considering heterogeneous $F_t(s)$. We only need to include t as a state parameter into $\mathcal{S}$, which enables time-varying storage value function estimation.

However, this system (23)–(26) is intractable for two reasons. Firstly, the equations are with infinite dimensions and variables due to the infinite possible values of SoC, which is computationally intractable. Second, the *pdf* $f(q)$ in Eq. (23) is unknown, but can be potentially estimated from data. To address these challenges, in the next section, we design a data-driven approach to estimate the storage value function $F(SoC)$ with theoretical accuracy guarantees.

IV. Online Algorithm Design: Data-Driven Implementations

In this section, we formally introduce how to implement the online storage control algorithm based on limited samples. Specifically, we first illustrate how to estimate $F(SoC)$ based on samples. Then, we implement the storage control algorithm based on the estimation.

Suppose we have a set of historical data Q with amount N_s, i.e., $\mathcal{Q} = \{q_1, q_2, \ldots, q_{N_s}\}$. Each piece of data q_i consists of the random state parameters at a single time slot, i.e., $q_i = (p_i^+, q_i^-, \hat{w}_i, w_i)$. These historical data samples will be employed to estimate the storage value function.

A. Sample-Based Algorithm Design

As we mentioned in Section III, calculating $F(SoC)$ poses two key challenges. The first is associated with the continuity of $F(s)$, and the second originates from the unknown nature of $f(p)$. We address these two challenges separately as follows:

1) SoC State Discretization: We discretize the continuous SoC into finite discrete points, i.e., $SoC \in \{0, \frac{C}{M}, \frac{2C}{M}, \ldots, \frac{(M-1)C}{M}, C\}$. And we only need to estimate the corresponding finite function values $F(0)$, $F(\frac{C}{M})$, $F(\frac{2C}{M})$, ..., $F(\frac{(M-1)C}{M})$, $F(C)$. At each time t, the resulting SoC can only be one of the above values.

2) Data-Driven Sample Average Estimation: Due to the difficulty of obtaining the true *pdf* of $f(q)$, where $q = (p^+, p^-, \hat{w}, w)$, we use the sample average approximation to estimate the value function in (23). Specifically, given a set of data $\mathcal{Q}$ with sample size N_s, the value function $F(SoC)$ satisfies:

$$F(SoC) = \frac{1}{N_s} \sum_{i=1}^{N_s} R^*(q_i, SoC). \quad (28)$$

Based on the above two methods, we design the data-driven estimation as follows:

$$F\left(\frac{kC}{M}\right) = \frac{1}{N_s} \sum_{i=1}^{N_s} R^*\left(q_i, \frac{kC}{M}\right), \forall k \le M, \quad (29)$$

where $R^*(q_i, \frac{kC}{M})$ denotes the cumulative cost with SoC $\frac{kC}{M}$ under state q_i, which is defined as follows:

$$\begin{aligned} R^*\left(q_i, \frac{kC}{M}\right) = \min_{v^-, v^+} c(q_i, g) \ + \gamma F\left(\frac{kC}{M} + \Delta\right) \quad (30)\\ \text{s.t. } g = w_i + v^- - v^+, \quad (31)\\ \text{Constraints } (25)–(26), \quad (32)\\ \Delta + \frac{kC}{M} \in \left[0, \frac{C}{M}, \frac{2C}{M}, \ldots, C\right]. \quad (33) \end{aligned}$$

Although the above optimization contains integer variables, all the variables only have finite possible values due to discretization. This helps us to make the following simplification:

Proposition 1: The constrained problem (30)–(33) can be equivalently transformed into the following unconstrained form:

$$R^*\left(q_i, \frac{kC}{M}\right) = \min_{0 \le j \le M} \left(h(q_i, k, j) + \gamma F\left(\frac{jC}{M}\right)\right), \quad (34)$$

where $h(q_i, k, j)$ is a deterministic constant and can be calculated in advance for all i, k, and j satisfying the following:

$$h(q_i, k, j) = \begin{cases} p_i^+ \max\Big(w_i - \hat{w}_i - \frac{(j-k)C}{\eta^+ M}, 0\Big) \\ \quad +p_i^- \max\Big(\hat{w}_i - w_i + \frac{(j-k)C}{\eta^+ M}, 0\Big), \quad j \ge k, \\ p_i^+ \max\Big(w_i - \hat{w}_i - \frac{\eta^-(j-k)C}{M}, 0\Big) \\ \quad +p_i^- \max\Big(\hat{w}_i - w_i + \frac{\eta^-(j-k)C}{M}, 0\Big), j < k. \end{cases}$$

Then Eq. (29) can be transformed into a much simpler form:

$$F\left(\frac{kC}{M}\right) = \frac{1}{N_s} \sum_{i=1}^{N_s} \min_{0 \le j \le M} \left(h(q_i, k, j) + \gamma F\left(\frac{jC}{M}\right)\right), \forall k. \quad (35)$$

Conventionally, solving equations with the min operator needs either an iteration approach or the integer programming.

[4] Note that, the committed wind power $\hat{w}$ is essentially not random. However, only the generation mismatch $w - \hat{w}$ exists in our problem, and such mismatch can be regarded as a random variable.

Algorithm 2 Online Storage Control (OSC) Algorithm

```
Input: Historical data $\mathcal{Q} = \{q_1, q_2, ..., q_{N_s}\}$; Size of dataset $N_s$; Discretization level $M$; Storage capacity $C$; Charging and discharging efficiency $\eta_+, \eta_-$;
Output: Storage control policy $v_t^+$, $v_t^-$ at each time $t$;
Value Function Construction:
for $i = 1, 2, ..., N_s$ do
    for $k = 0, 1, ..., M$ do
        for $j = 0, 1, ..., M$ do
            Solve $h(q_i, k, j)$ according to data $q_i$.
        end for
    end for
end for
Solve problem (LP) in Eqs. (36)-(38) to obtain the value function $F\left(\frac{kC}{M}\right)$ for $k = 0, 1, ..., M$;
Linear interpolation of $F\left(\frac{kC}{M}\right)$ for different $k$'s to get a continuous function $F(SoC)$;
Online Optimization:
for $t = 1, 2, ...$ do
    Obtain parameters $w_t$, $\hat{w}_t$, $p_t^+$, $p_t^-$ in real time.
    Solve the problem (P4);
    Return the solved storage control policy at time $t$;
end for
```

In contrast, we propose an equivalent linear programming to accurately solve it:

$$\textbf{(LP)} \quad \min_{y_k, x_{k,i}} \quad \sum_{k=0}^{M} y_k \tag{36}$$

$$\text{s.t.} \quad y_k \geq \frac{1}{N_s} \sum_{i=1}^{N_s} x_{k,i}, \forall k, \tag{37}$$

$$x_{k,i} \geq h(q_i, k, j) + \gamma y_j, \forall k, \forall i, \forall j. \tag{38}$$

The following theorem illustrates the equivalence:

Theorem 1 [24, Th. 1]: The optimal solution y_k^* of **(LP)** equals the solution of Eq. (35), which satisfies:

$$y_k^* = F\left(\frac{kC}{M}\right), \forall k. \tag{39}$$

By solving **(LP)**, we can obtain the value function $F(\frac{kC}{M})$ and implement the online storage control problem. We can observe that, with a growing discretization level M, the problem scale increases in $\mathcal{O}(M^2)$. The amount of data N_s also contributes to the problem scale in $\mathcal{O}(N_s)$, though, even with a large M and N_s (e.g., $M = 50$ and $N_s = 100$), linear programming is one of the most well-investigated optimization problems, and can be solved very efficiently based on commercial solvers. After obtaining $F(\frac{kC}{M})$ for different k's, we can simply make linear interpolation to get a piecewise linear approximation of the real value function $F(SoC)$.

Based on the value function estimation, we can implement the conceptual Algorithm 1 into a practical data-driven form in Algorithm 2. Specifically, after obtained the value function $F(SoC)$, then at each time t, we first collect the real-time parameters w_t, $\hat{w}_t$, p_t^+, p_t^-. Parameters w_t and $\hat{w}_t$ denote the real and committed wind power generations at time t, respectively. p_t^+ and p_t^- denote the unit energy surplus and shortage penalty costs. Based on these parameters, we can solve **(P4)** to calculate the optimal storage control policy v_t^+ and v_t^- at time t:

$$\textbf{(P4)} \quad \min_{v_t^+, v_t^-} \quad \underbrace{c_t(\hat{w}_t, g_t)}_{\text{Current Cost}} + \underbrace{F(SoC_{t+1})}_{\text{Expected Future Cost}} \tag{40}$$

$$\text{s.t.} \quad g_t = w_t + v_t^- - v_t^+, \tag{41}$$

$$SoC_{t+1} = SoC_t + \eta^+ v_t^+ - \eta^- v_t^-, \tag{42}$$

$$\eta^+ v_t^+ \leq C - SoC_t, \tag{43}$$

$$\eta^- v_t^- \leq SoC_t, \tag{44}$$

$$v_t^+ \leq w_t, \tag{45}$$

$$v_t^+, v_t^- \geq 0, \tag{46}$$

$$v_t^+ v_t^- = 0. \tag{47}$$

Specifically, the two decision variables of **(P4)** are v_t^+ and v_t^-, which denote the charging and discharging powers of storage at time t. The objective in (40) is the sum of current cost $c_t(\hat{w}_t, g_t)$ at time t and the expected future cost $F_{t+1}(SoC_{t+1})$. **(P4)** minimizes the total costs of the current time and the future.

Constraint (41) characterizes the total wind power supply; constraint (42) describes the dynamics of storage; and constraints (43) and (44) represent the storage capacity limits. Constraints (45) and (46) indicate the upper and lower limits of storage control actions; constraint (47) ensures that the storage cannot be charged and discharged simultaneously. Since both $c_t(\hat{w}_t, g_t)$ and $F(SoC)$ are piecewise linear convex functions, **(P4)** can be efficiently solved.

Remark: The whole algorithm is highly computationally efficient. Specifically, the value function construction procedure is only required to be done once before the online optimization procedure begins, which solves a linear program within several minutes. For the online optimization procedure, we only need to solve a tiny-scale piecewise linear problem **(P4)**[5] with two variables v_t^-, v_t^+ and seven constraints, which can be solved very efficiently within 0.01 s.

Remark: Our approach can be easily extended to consider the charging and discharging rate constraints. Specifically, we only need to add linear constraints $v_t^+ \leq UR$ and $v_t^- \leq DR$ into **(P1)-(P4)**, where UR and DR denote the maximal charging and discharging rates. Such modification only introduces additional linear constraints and does not influence the problem structure and the analysis.

B. Theoretical Guarantees

The discretization and limited data will inevitably impact the accuracy of the estimation, and subsequently influence the algorithm's performance. In this part, we theoretically characterize the accuracy of value function estimation regarding the discretization and limited data. Then, we derive the

[5] The piecewise linearity of the objective is due to the piecewise linearity of both $c_t(\hat{w}_t, g_t)$ and $F(SoC_{t+1})$. The binary constraint in (47) can be released by considering $v_t^+ = 0$ and $v_t^- = 0$ separately.

performance gap between our algorithm and the optimal online algorithm in terms of regret.

Denote $\hat{F}(\frac{kC}{M})$ and $F^*(\frac{kC}{M})$ as the estimated and actual value function with SoC $\frac{kC}{M}$, respectively. We first analyze how samples influence the accuracy of the value function estimation, which is provided in the following theorem:

Theorem 2 (Value of Samples): Given a randomly sampled dataset $\mathcal{Q}$ with sample size N_s, for any given k and error bound θ, the estimated value function $\hat{F}(\frac{kC}{M})$ with discretization level M satisfies:

$$\mathbf{Pr}\left(\left|\hat{F}\left(\frac{kC}{M}\right)-F^*\left(\frac{kC}{M}\right)\right|\geq\theta\right)$$
$$\leq 2\exp\left(\frac{-2N_s(1-\gamma)^4\theta^2}{p_{\max}^2(C+\Delta w_{\max})^2}\right), \tag{48}$$

where $p_{\max}$ denotes the maximal penalty price, i.e.,

$$p_{\max}=\max\bigl(\max_{q\in\mathcal{Q}}p^+,\max_{q\in\mathcal{Q}}p^-\bigr), \tag{49}$$

and $\Delta w_{\max}$ denotes the maximal gap between wind power forecast $\hat{w}$ and real generation w, i.e.,

$$\Delta w_{\max}=\max_{q\in\mathcal{Q}}|\hat{w}-w|. \tag{50}$$

This theorem elucidates the enhancement of value function estimation accuracy via samples. Specifically, for any given error range θ, when sample N_s increases linearly, the probability of violating the error range decreases rapidly in $\mathcal{O}(e^{-N_s})$, which demonstrates the value of data for storage control. Also, a large penalty price $p_{\max}$, a large storage capacity C, and a large forecast error $\Delta w_{\max}$ all contribute to a significant estimation error, which is consistent with our intuition.

We also show how discretization influences the accuracy of value function estimation by the following theorem:

Theorem 3 (Price of Discretization): Given a randomly sampled dataset $\mathcal{Q}$ with sample size N_s, for any given k, the estimated value function $\hat{F}(\frac{kC}{M})$ with discretization level M satisfies:

$$\left|\hat{F}\left(\frac{kC}{M}\right)-F^*\left(\frac{kC}{M}\right)\right|\leq\frac{LC^2+2MCp_{\max}}{M^2(1-\gamma)}, \tag{51}$$

where L denotes the Lipschitz constant of the value function F^*'s gradient.

This theorem indicates that higher discretization level M results in the more accurate approximation of $F^*(\cdot)$ at the rate of $\mathcal{O}(1/M)$. Also, a large penalty price $p_{\max}$ and storage capacity C will make the estimation less accurate.

Based on these results, we finally characterize the gap in performance between our algorithm and the optimal online algorithm. We first introduce a comparative metric as follows:

Definition 1 (Expected Regret [23]): The expected regret of an online algorithm π is defined as:

$$\mathcal{R}_\pi=\sum_{t=1}^{T}\mathbb{E}_{q^t\in\mathcal{Q}}\left(c_t\bigl(\hat{w}_t,g_t^\pi\bigr)-\min_{g_t}c_t\bigl(\hat{w}_t,g_t\bigr)\right), \tag{52}$$

where g_t^π denotes the decisions of algorithm π at time t.

The expectation is taken over all possible states $q^t\in\mathcal{Q}$. Intuitively, the regret $\mathcal{R}_\pi$ represents the additional cost induced by an algorithm π compared with the optimal online algorithm.

Based on the definition, we can derive the following regret bound for our proposed online storage control algorithm:

Theorem 4 (Linear Regret Bound): Given a randomly sampled dataset $\mathcal{Q}$ with sample size N_s, the expected regret $\mathcal{R}_\pi$ of our algorithm with discretization level M satisfies:

$$\mathcal{R}_\pi=T\cdot\mathcal{O}\left(\frac{Cp_{\max}}{M}+\sqrt[4]{\frac{p_{\max}^6(C+\Delta w_{\max})^2}{N_s}}\right). \tag{53}$$

We discern that, our online algorithm exhibits a linear regret regarding T. The discretization level M and sample size N_s jointly influence the coefficient of the linear term. Specifically, the increasing discretization level M can mitigate the regret at the rate of $\mathcal{O}(M^{-1})$, while an increasing sample size N_s reduces the regret in $\mathcal{O}(M^{-\frac{1}{4}})$. However, the discretization process is independent of data, so in practice, we can choose a large enough M, and solve the value function offline before starting the storage control. Consequently, the sample size N_s becomes the bottleneck of improving the algorithm's performance. In the next section, we will address this issue by offering our algorithm the ability to continuously learn from the new data during the control process, which can break the linear-regret bound and achieves asymptotic optimality.

V. Online Algorithm Design: Self-Improving Approach

In this section, we further enhance the online storage control algorithm by enabling it to continuously learn from new data and improve performance during the storage control process. We also emphasize that the online learning algorithm can break the linear regret bound and achieves an $\mathcal{O}(T^{\frac{3}{4}})$ regret performance.

A. Self-Improving Algorithm Design

Recall that, our proposed data-driven online algorithm makes decisions by the estimated value function. If the estimation is perfect, then the corresponding one-shot decision can minimize the expected cost in the future. Therefore, in order to improve the online algorithm, the crucial element is continuously collecting new data during the control process to refine the value function estimation, which is a simple and famous philosophy known as online learning [25]. Specifically, at each time t, we can include the newly collected data $q_t=(p_t^+,p_t^-,\hat{w}_t,w_t)$ into the dataset $\mathcal{Q}$, and conduct the value function estimation again. As time passes, the estimation with a growing number of data can approach the accurate value function.

However, the value function estimation requires solving the linear programming in **(LP)**. As time progresses, the problem scale will continuously increase due to the expanding dataset, and brings the computational burden. To tackle the problem, we have a simple observation that, when the collected data already possesses substantial volume, an additional data sample has negligible impact on the estimated value function,

Algorithm 3 Policy Iteration Algorithm

```
Input  Discretization level $M$; storage capacity $C$; initial value
    function $F\left(\frac{kC}{M}\right), \forall k$; historical dataset $\mathcal{Q}$; dataset size $N_s$;
    newly obtained data $q_t = (w_t, \hat{w}_t, p_t^+, p_t^-)$;
Output  The optimized value function $F\left(\frac{kC}{M}\right), \forall k$.
Initial Policy Generation:
$\mathcal{Q} = \mathcal{Q} \cup q_t$;
$N_s = N_s + 1$;
for $i = 1, 2, ..., N_s$ do
    for $k = 0, 1, ..., M$ do
        $A_{i,k} = \arg\min_{0 \le j \le M} \left(h(q_i, k, j) + \gamma F\left(\frac{jC}{M}\right)\right), \forall k.$
    end for
end for
Policy Iteration:
while Policy $A_{i,k}$ for all $i, k$ do not change do
    Update the value function $F\left(\frac{kC}{M}\right)$ for all $k$ based on
        estimated action $A_{i,k}, \forall i, \forall k$;
    Update the policy $A_{i,k}$ for all $i, j$ based on the updated
        value function $F\left(\frac{kC}{M}\right), \forall k$;
end while
Return value function $F\left(\frac{kC}{M}\right), \forall k$;
```

Algorithm 4 Self-Improving Online Storage Control (SOSC) Algorithm

```
Input: Historical data $\mathcal{Q} = \{q_1, q_2, ..., q_{N_s}\}$; size of dataset
    $N_s$; discretization level $M$; storage Capacity $C$; charging
    and discharging efficiency $\eta_+$, $\eta_-$;
Output: Storage control policy $v_t^+$, $v_t^-$ at each time $t$;
Value Function Construction:
for $i = 1, 2, ..., N_s$ do
    for $k = 0, 1, ..., M$ do
        for $j = 0, 1, ..., M$ do
            Solve $h(q_i, k, j)$ according to data $q_i$.
        end for
    end for
end for
Solve problem (LP) in Eqs. (36)-(38) to obtain the value
    function $F(\frac{kC}{M})$ for $k = 0, 1, ..., M$;
Linear interpolation of $F\left(\frac{kC}{M}\right)$ for different $k$'s to get a
    continuous function $F(SoC)$;
Online Optimization:
for $t = 1, 2, ...$ do
    Obtain parameters $q_t = (w_t, \hat{w}_t, p_t^+, p_t^-)$ in real time.
    Update dataset $\mathcal{Q} = \mathcal{Q} \cup q_t$;
    Update the value function $F(\frac{kC}{M})$ based on dataset $\mathcal{Q}$
        and the policy iteration algorithm;
    Solve the problem (P3);
    Return the solved storage control policy at time $t$;
end for
```

and the resulting control policy alters minutely. Therefore, we can leverage such consistency and accelerate the value function estimation update process by incorporating the policy iteration method [23].

Algorithm 3 summarizes the policy iteration algorithm. Specifically, the policy iteration algorithm iteratively determines the optimal control policy for each possible scenario based on the estimated value function from the previous round. Then, following the estimated control policy, the algorithm updates the value function based on the resolved control policy. If the resolved control policies in two consecutive iterations are identical, then the corresponding value function is accurately updated. The policy iteration algorithm is frequently employed in reinforcement learning due to its rapid update advantage, which can also expedite our online estimation update process.

Leveraging the policy iteration algorithm, we can eventually implement our self-improving online storage control algorithm, which is presented in Algorithm 4.

B. Performance Analysis

In this part, we emphasize the asymptotic optimality of our proposed self-improving algorithm. The following theorem indicates the regret performance of our algorithm:

Theorem 5 (Sub-Linear Regret Bound): Given the dataset $\mathcal{Q}$ with sample size N_s, the expected regret $\mathcal{R}_\pi$ of the self-improving online algorithm satisfies:

$$\mathcal{R}_\pi = \mathcal{O}\left(\frac{T}{\sqrt[4]{N_s + T}}\right). \tag{54}$$

This theorem indicates that, our designed self-improving algorithm can achieve an expected regret at the rate of $\mathcal{O}(T^{\frac{3}{4}})$,[6] which is sub-linear in T. It reveals the asymptotic optimality of our algorithm:

$$\lim_{T \to \infty} \frac{\hat{W}_T}{W_T^*} = 1, \tag{55}$$

where $\hat{W}_T$ and W_T^* denote the accumulated costs of our algorithm and the optimal online algorithm across T time slots, respectively.

VI. Numerical Studies

In this section, we evaluate the performance of our proposed online storage control algorithms (OSC and SOSC) and the other benchmark algorithms with field data.

A. Simulation Settings

To comprehensively validate the performance of our algorithm under diverse system scales and scenarios, we have undertaken a numerical study using three distinct datasets, each representing a different system level:

- *Region Aggregation Level*: For this level, we utilized the California aggregate wind power generation dataset from

[6] The discretization level M essentially influences the rate of the regret. However, the discretization process is independent of data, and we can choose a very large M in practice to eliminate the impact of discretization.

CAISO [20]. This dataset comprises real wind power generation data with a 5-minute resolution spanning from January 2020 to December 2020.

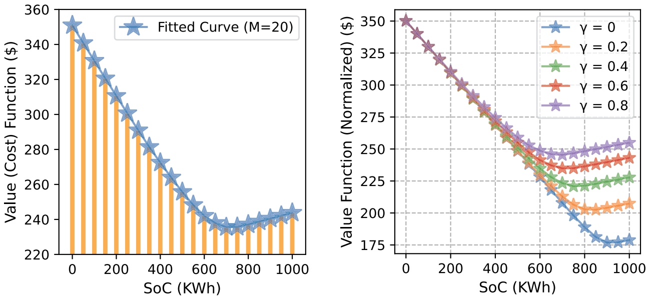
- *Wind Farm Level*: At the wind farm level, we employed the SDWPF wind farm generation dataset [26] from Longyuan Power Group Corp. Ltd, China. This dataset contains real wind farm power generation data with a 10-minute resolution covering the same period from January 2020 to December 2020.
- *Wind Turbine Level*: At the finest granularity, we utilized the SCADA wind turbine-level generation dataset [27] from Turkey. This dataset includes power generation data for a single wind turbine at 10-minute intervals spanning from January 2018 to December 2018.

To ensure consistency and comparability, we normalized all data to the same order of magnitude and processed it into 10-minute resolution data. The real wind power generation data w are directly obtained from the dataset, and the committed wind power generation data $\hat{w}$ are based on the 3-hour-ahead forecasts. We generated wind power forecasts using a Long Short-term Memory (LSTM) model [28] with one hidden layer comprising 128 units. The model took the wind power generation data from the previous two days as inputs and produced power generation forecasts for the next 3 hours at a 10-minute resolution. The shortage penalty price equals the average electricity price of CASIO [20] with the matching resolution and periods. The surplus penalty price is set to be much smaller than the shortage penalty, conforming to the Gaussian distribution with a mean of \$10 and a standard deviation of \$1. Each storage control action occurred in 10-minute time slots. Additionally, we set $C = 1000$ KWh, $M = 20$,[7] $\eta^{+} = 0.9$, $\eta^{-} = 1.1$, $\gamma = 0.6$. Moreover, we use the data from the first 3 months to train the algorithms, and utilize data from the subsequent 9 months for evaluation.

B. Competing Methods

We compare our algorithm with the following 6 benchmark approaches:

- *Greedy Algorithm (GA):* GA charges and discharges the storage greedily. Specifically, whenever the generation shortage or surplus exists, GA charges or discharges the storage as much as possible to serve the grid until the storage is empty or reaches capacity.
- *Lyapunov Optimization-based Online Algorithm (LYA):* The Lyapunov algorithm decides the control actions by optimizing the upper bound of the Lyapunov drift-plus-penalty function [13].
- *Model Predictive Control (MPC):* The MPC predicts the future information $(p_t^{+}, p_t^{-}, \hat{w}_t, w_t)$ in advance, and then solves the optimization based on prediction and conducts the storage control policy at the current time slot [6]. We adopt the fully connected neural network (FCNN) for the price and generation forecasts.



(a) Shape of Value Function (b) Impact of γ

Fig. 3. Value Function Estimation.

- *Threshold-based Online Algorithm (TOA):* TOA decides whether and how much to charge and discharge according to whether the electricity price is above a threshold [11].
- *Deep Q Learning Algorithm (DQN):* DQN is an advanced reinforcement learning algorithm that trains a three-layered neural Q-network to learn the optimal storage control strategy based on the continuous state parameter inputs [29].
- *Offline Optimal Approach (OPT):* OPT is assumed to know all future information in advance and can solve the offline optimal solution with the optimal cost.

For the hyper-parameters setting of these benchmarks, we use a random search approach [30] in the training set to determine the optimal hyper-parameters. These hyper-parameters include the stable SoC level and weight of cost for LYA [13]; the prediction window size for MPC [6]; the control threshold for TOA [11]; the learning rate, number of neurons, batch size, discount ratio and ϵ-greedy parameters for DQN [29].

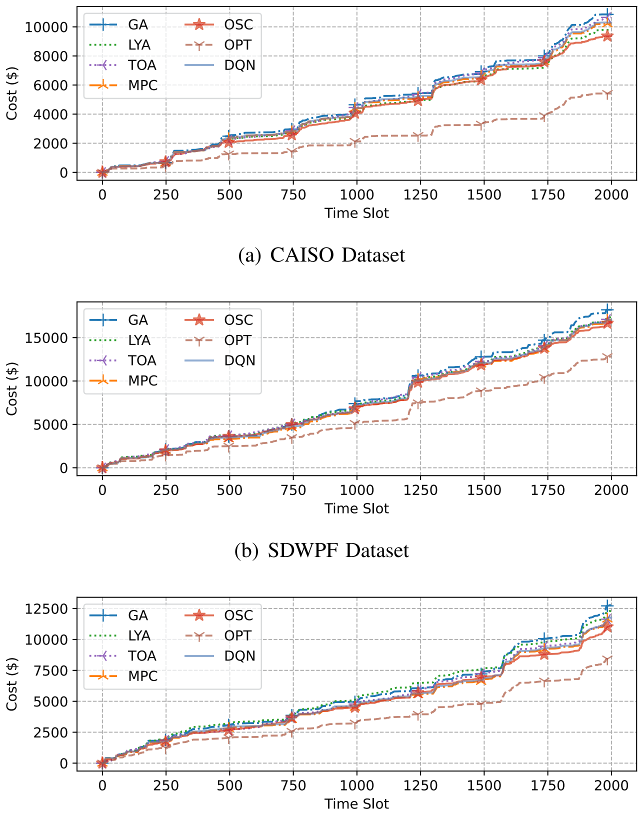
The numerical study is performed by CVXPY 1.3.1 [31] and COPT solver 6.5.5 [32] on a desktop with Intel Core i5-11400F CPU and 16G RAM.

C. Performance Evaluation

Fig. 3 visualizes the estimated value function. Notably, in Fig. 3(a) where $\gamma = 0.6$, the value function is convex and reaches the minimum when SoC is approximately 700 KWh. This suggests that our method aims to maintain the SoC around 700 KWh to balance risks associated with electricity shortage and surplus. The curve's gradient when SoC is small is steeper than that when SoC is large, which indicates the electricity shortage risk outweighs the surplus risk. Fig. 3(b) further illustrates the impact of parameter γ on the value function estimation. We can discern that with $\gamma = 0$, the value function is minimized when SoC is roughly 900 KWh, indicating a large SoC is preferred to avoid shortage cost. However, as γ increases, the SoC minimizing the value function decreases. It indicates that when our algorithm gives greater consideration to future costs (reflected by a higher γ), a more symmetrical value function curve is better for balancing the risks of electricity shortage and surplus.

Figure 4 provides a comprehensive performance evaluation across various datasets. In particular, Figs. 4(a), 4(b), and 4(c) showcase the accumulated costs incurred over 2,000 time steps by seven distinct methodologies. Notably, the GA method

[7] In practice, setting $M = 20$ to 30 is often enough to approximate the value function curve with enough accuracy and short computation time (typically less than 5 minutes).



(a) CAISO Dataset

(b) SDWPF Dataset

(c) SCADA Dataset

Fig. 4. Cost Evolution (C = 1000KWh).

TABLE I
PERFORMANCE EVALUATION WITH CAISO DATASET

C (KWh)	Costs of Approaches ($)						
	GA	LYA	MPC	TOA	DQN	OSC	OPT
500	13782	12379	12945	12822	13007	**11898**	9091
750	11943	11125	11229	11394	11339	**10794**	7398
1000	10855	9785	10205	10672	10297	**9364**	5419

TABLE II
PERFORMANCE EVALUATION WITH SDWPF DATASET

C (KWh)	Costs of Approaches ($)						
	GA	LYA	MPC	TOA	DQN	OSC	OPT
500	22051	20861	**20562**	20874	20625	20585	17668
750	19767	18826	18407	18807	18475	**18298**	14771
1000	18209	17340	17025	17144	17052	**16788**	12808

TABLE III
PERFORMANCE EVALUATION WITH SCADA DATASET

C (KWh)	Costs of Approaches ($)						
	GA	LYA	MPC	TOA	DQN	OSC	OPT
500	14416	13318	13247	13146	13214	**13089**	10700
750	13298	12669	12203	12465	12272	**11790**	9571
1000	12768	12293	11696	11685	11606	**11109**	8428

TABLE IV
PERFORMANCE EVALUATION FOR DIFFERENT MONTHS

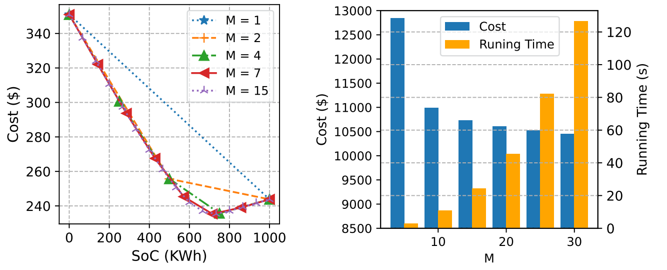
Month	Costs of Approaches ($)						
	GA	LYA	MPC	TOA	DQN	OSC	OPT
Apr.	10855	9785	10205	10672	10297	**9364**	5419
May	14128	13094	13072	13325	13153	**12882**	9249
Jun.	9576	9007	9047	9002	9116	**8979**	5490
Jul.	7844	7246	7204	7255	7239	**6716**	4354
Aug.	8721	7455	8054	7651	7994	**6887**	4336
Sep.	14955	12844	13704	13256	13695	**12402**	9014

incurs the highest cost across all three datasets. This suboptimal performance can be attributed to GA's inability to effectively harness price information. Using current price data, both the TOA and Lyapunov techniques manage to reduce cumulative costs. In contrast, our proposed OSC method, which utilizes value function estimation from historical data, not only outperforms these but even exceeds the capabilities of the MPC method. Compared with DQN, a large model with considerably more parameters to be trained, our approach can learn the optimal control policy more efficiently with limited data, leading to superior performance outcomes.

To provide a deeper quantitative insight, we evaluate the performance of various methods with different storage capacities. As demonstrated by Table I, within the CAISO dataset, our method surpasses the top-performing benchmark, LYA, by achieving an average cost reduction of 3.7%. In the SDWPF dataset, denoted by Table II, our method stands out particularly at storage capacities of 750 KWh and 1,000 KWh, realizing a 1% average cost decrement. Even with 500 KWh storage, our approach's cost is marginally greater than the top benchmark MPC, with a mere 0.1% difference. Further, Table III emphasizes that, in comparison to the leading benchmark DQN, our method boasts an average performance enhancement of 2.8%, underscoring its exceptional efficacy with different storage sizes and wind power scenarios.

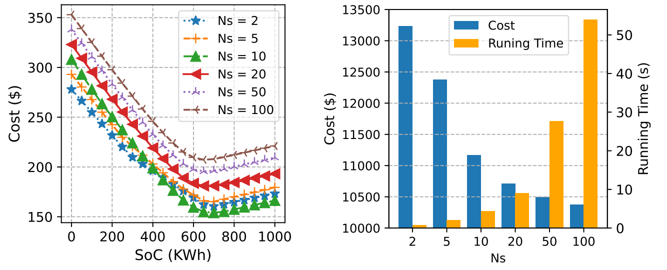
We further assess the temporal robustness of our algorithm. Specifically, we conducted evaluations across varied time spans at a storage size of 1,000 KWh within the CAISO dataset. Table IV reveals that during May, June, and September, our approach outstrips the best benchmarks by margins ranging from 2% to 3%. In April, it surpassed the best LYA benchmark with a notable 4.3% cost improvement. Even more impressively, during the months of July and August, we observed performance enhancements exceeding 7%. These results affirm the consistent and superior performance of our algorithm across diverse time periods.

Fig. 5 depicts the effects of the discretization level M. Specifically, Fig. 5(a) presents the estimated value function curves with different M. It can be observed that, when $M = 1$, the value function is just a linear function, and as M increases, the value function approaches the optimal curve fast in a piecewise linear fashion. Fig. 5(b) illustrates how M influences the algorithm's economic and computational performance. As M increases, the resulting cost gradually reduces and converges, and the computation time increases swiftly.



(a) Value Function Estimation (b) Cost-Time Trade-off

Fig. 5. Price of Discretization.



(a) Value Function Estimation (b) Cost-Time Trade-off

Fig. 6. Value of Self-improving Algorithm.

Fig. 6 showcases the effectiveness of the self-improving algorithm. In particular, Fig. 6(a) indicates that, the self-improving algorithm can continuously collect data during the control process and iteratively refine the value function estimation. Meanwhile, a small amount of data with $N_s = 100$ (which can be collected approximately in 8 hours) can render the value function estimation rather accurate, which demonstrates the effectiveness of our approach. Fig. 6(b) further indicates that the cost can be well reduced with more collected data.

VII. Conclusion

In this paper, we propose a one-shot online storage control algorithm based on the MDP theory. To tackle the challenges from computational intractability and the limitations of distribution information, we implement the online algorithm leveraging limited data with theoretical performance guarantees. To promote continuous learning from new data, we further design a self-improving online algorithm based on the online learning scheme. Theoretical results highlight the asymptotic optimality of our approach. Numerical study based on field data further verifies the remarkable performance of our algorithm.

Our work opens up avenues for extension in various directions. In terms of methodology enhancement, there is room for developing improved value function approximation techniques that offer lower approximation errors and enhanced regret performance. As for the practical application of our algorithm, it would be very interesting and meaningful to implement this approach for smart grid problems with more complex decision spaces, such as distributed storage control, voltage control in distribution networks, and electric vehicle charging scheduling.

Appendix

A. Modeling Details of Wind Power Mismatch Costs

In this section, we introduce the detailed structure of mismatch costs to justify our cost modeling. Specifically, the mismatch cost c_t at time t consists of the shortage cost c_t^- and excess cost c_t^+ as follows:

$$c_t = c_t^- + c_t^+. \tag{56}$$

We introduce these two costs as follows:

- *Shortage Cost c_t^-:* It consists of two parts: the shortage energy payment $c_t^{-,e}$ and shortage penalty $c_t^{-,p}$. The shortage energy payment is charged since the supplied energy is below the required amount. Therefore, the wind farm cannot get all payment of full supply which is stipulated in the contract, but to subtract the shortage energy payment. Specifically, such payment is linear to the shortage amount and satisfies:

$$c_t^{-,e} = p^{-,e}\left(\hat{w}_t - g_t\right)^+, \tag{57}$$

where $p^{-,e}$ is unit electricity price at time t; $\hat{w}_t$ and g_t denote the required wind power supply and actual wind power supply (after storage control), respectively. Also, an additional penalty cost $c_t^{-,p}$ of deviation will be charged as follows:

$$c_t^{-,p} = p^{-,p}\left(\left(1-\alpha^-\right)\hat{w}_t - g_t\right)^+, \tag{58}$$

where $p^{-,p}$ denotes the unit penalty price of shortage, α^- is the allowed shortage ratio ($0 \le \alpha^- \le 1$). It indicates that the penalty cost will be charged when the real supply is below $1-\alpha^-$ of requirement. Therefore, the shortage cost c_t^- satisfies:

$$c_t^- = p^{-,e}\left(\hat{w}_t - g_t\right)^+ + p^{-,p}\left(\left(1-\alpha^-\right)\hat{w}_t - g_t\right)^+. \tag{59}$$

If the unit energy cost $p^{-,e}$ is significantly larger than penalty cost $p^{-,p}$, or the allowed shortage ratio α^- is small, c_t^- can be approximated by the following one-side linear form:

$$c_t^- \approx \left(p^{-,e} + p^{-,p}\right)\left(\hat{w}_t - g_t\right)^+ = p^-\left(\hat{w}_t - g_t\right)^+, \tag{60}$$

which is consistent with the form in our paper. We also demonstrate that this is not a strong assumption. For comparing the unit energy cost $p^{-,e}$ and the unit penalty cost $p^{-,p}$, [33] and [34] indicates the penalty price is much lower than the energy price. Also, for the value of α^-, [33] assumes $\alpha^- = 0.04$, and [35] assumes $\alpha = 0$. It means that α is relatively small in practice. This is particularly true because of the improving accuracy of power prediction, the deviation of wind power prediction is significantly reduced [21]. Hence a higher wind power stability is preferred and required.

- *Excess Cost c_t^+:* This cost is dependent on the setting of whether wind curtailment is allowed. If it is allowed, then the excess cost is referred to the wind curtailment cost [2], which is linear to the excess amount with the following form:

$$c_t^+ = p_t^{+,w}\left(g_t - \hat{w}_t\right)^+, \tag{61}$$

where $p_t^{+,w}$ denotes the unit wind curtailment cost. Then, it is exactly the same one-side linear form as our model. Otherwise, the power grid will accept the abundant wind power but charges the excess penalty cost. The excess cost c_t^+ then satisfies:

$$c_t^+ = -p^{+,e}\big(g_t - \hat{w}_t\big)^+ + p^{+,p}\big(g_t - \big(1+\alpha^+\big)\hat{w}_t\big)^+, \quad (62)$$

where $p^{+,e}$ denotes the unit energy price of the excess wind power, $p^{+,p}$ characterizes the unit excess penalty cost, and α^+ is the allowed excess ratio $\alpha \geq 1$, which indicates that the penalty cost will be charged when the real supply is above $1+\alpha^+$ of requirement. If the unit energy price $p^{+,e}$ is significantly larger than the penalty price $p^{+,p}$, or α^+ is small, then c_t^+ can be approximated by:

$$c_t^+ \approx \big(-p^{+,e} + p^{+,p}\big)\big(g_t - \hat{w}_t\big)^+ = p^+\big(g_t - \hat{w}_t\big)^+. \quad (63)$$

Similarly, according to [2] and [35], such approximation holds without strong assumption.

By combining the shortage cost and the excess cost, we can derive the total cost with the same form in our model:

$$c_t = p^-\big(\hat{w}_t - g_t\big)^+ + p^+\big(g_t - \hat{w}_t\big)^+. \quad (64)$$

B. Proof Sketch for Theorem 2

We define the estimation error $\delta_k = \tilde{F}(\frac{kC}{M}) - F^*(\frac{kC}{M})$. The notations $\tilde{F}(\frac{kC}{M})$ and $F^*(\frac{kC}{M})$ are represented by the simplified forms $\tilde{F}(k)$ and $F^*(k)$, respectively. Then for any $k \leq M$, we can derive the following condition:

$$\begin{aligned}
&|\tilde{F}(k) - F^*(k)| \\
&\quad\leq \Bigg|\frac{1}{N_s}\sum_{i=1}^{N_s}\min_{\Delta_i}\bigg(c(q_i,g_i) + \gamma F^*\bigg(\frac{kC}{M} + \Delta_i\bigg)\bigg) \\
&\qquad - \sum_{i=1}^{W}\alpha_i \min_{\Delta_i}\bigg(c(q_i,g_i) + \gamma F^*\bigg(\frac{kC}{M} + \Delta_i\bigg)\bigg)\Bigg| \\
&\qquad + \gamma\big|\tilde{F}(k) - F^*(k)\big|. \quad (65)
\end{aligned}$$

Given any fixed k, $\min_{\Delta_i}((c(q_i,g_i) + \gamma F^*(\frac{kC}{M} + \Delta_i)))$ is essentially a function of q_i. We denote it as y_i for simplicity. Then for each k, $|\tilde{F}(k) - F^*(k)|$ can be transformed into:

$$|\tilde{F}(k) - F^*(k)| \leq \frac{1}{1-\gamma}\left|\frac{1}{N_s}\sum_{i=1}^{N_s} y_i - \bar{y}\right|. \quad (66)$$

Since all q_i's are *i.i.d.*, for each k, we can derive the following condition based on the Hoeffding's inequality [36]:

$$\mathbf{P}\left(\left|\frac{1}{N_s}\sum_{i=1}^{N_s} y_i - \bar{y}\right| \geq \epsilon\right) \leq 2\exp\left(\frac{-2N_s\epsilon^2}{b^2}\right), \quad (67)$$

where $b = \frac{p_{\max}(C + \max_i(\hat{w}_i - w_i))}{1-\gamma}$, which is the derived upper bound of y_i following standard mathematical manipulations. The constant $p_{\max} = \max(\max_{q\in\mathcal{Q}} p^+, \max_{q\in\mathcal{Q}} p^-)$.

Combining conditions (67) and (66) yields our results. ■

C. Proof Sketch for Theorem 3

For the simplicity of symbolic representation, we define $[M] = \{0, 1, \ldots, M-1, M\}$. For any $k < M$, the following condition holds:

$$\begin{aligned}
&|\tilde{F}(k) - F^*(k)| \\
&\leq \left|\frac{1}{N_s}\sum\nolimits_{i=1}^{N_s}\left[\min_{\Delta_i\in[M]}\left(c(q_i,g_i) + \gamma\tilde{F}\left(\frac{kC}{M} + \Delta_i\right)\right)\right.\right. \\
&\quad \left.\left. - \min_{\Delta_i\in[M]}\left(c(q_i,g_i) + \gamma F^*\left(\frac{kC}{M} + \Delta_i\right)\right)\right]\right| \\
&\quad + \left|\frac{1}{N_s}\sum_{i=1}^{N_s}\left[\min_{\Delta_i\in[M]}\left(c(q_i,g_i) + \gamma F^*\left(\frac{kC}{M} + \Delta_i\right)\right)\right.\right. \\
&\quad \left.\left. - \sum_{i=1}^{N_s}\min_{\Delta_i\in[0,C]}\left(c(q_i,g_i) + \gamma F^*\left(\frac{kC}{M} + \Delta_i\right)\right)\right]\right|,
\end{aligned} \quad (68)$$

where each Δ_i satisfies condition (33).

In (68), the first absolute term can be bounded following the routine in the proof for Theorem 2, and the second term can be bounded by checking the Lipschitz continuity condition of the value function F^*'s gradient. Combining the two terms yields:

$$\begin{aligned}
&\big|\tilde{F}(k) - F^*(k)\big| \\
&\quad\leq \gamma\big|\tilde{F}(k^*) - F^*(k^*)\big| + \frac{C}{M}\left(2p_{\max} + \frac{LC}{M}\right), \quad (69)
\end{aligned}$$

where L denotes the Lipschitz constant of the value function F^*'s gradient. Based on simple substitution, we can derive the final result. ■

D. Proof Sketch for Theorem 4

We first derive the one-step regret bound, then conclude the cumulative regret bound. Specifically, for each i, we make the following definition:

$$\Delta_i^1 = \arg\min_{\Delta\in[0,C]}\big(c_i(\Delta) + \gamma F^*(SoC_t + \Delta)\big), \quad (70)$$

$$\Delta_i^2 = \arg\min_{\Delta\in[M]}\big(c_i(\Delta) + \gamma F^*(SoC_t + \Delta)\big), \quad (71)$$

$$\Delta_i^3 = \arg\min_{\Delta\in[M]}\big(c_i(\Delta) + \gamma\tilde{F}(SoC_t + \Delta)\big), \quad (72)$$

where $c_i(\cdot)$ is the penalty cost function with state q_i.

We characterize the upper bound of $\mathbb{E}[|c(\Delta_i^1) - c(\Delta_i^3)|]$ as follow:

$$\begin{aligned}
&\mathbb{E}\Big[|c\big(\Delta_i^1\big) - c\big(\Delta_i^3\big)|\Big] \\
&\quad\leq p_{\max}\mathbb{E}\Big(|\Delta_i^1 - \Delta_i^2| + |\Delta_i^3 - \Delta_i^2|\Big). \quad (73)
\end{aligned}$$

Specifically, we first derive the bound of $|\Delta_i^1 - \Delta_i^3|$. Due to the discretization, the following condition holds:

$$|\Delta_i^1 - \Delta_i^2| \leq \frac{C}{M}. \quad (74)$$

Since the estimated value function $\tilde{F}(k)$ depends on random samples. Therefore, $\tilde{F}(k)$ is also a random variable. Based on Theorem 2, we can derive the following condition:

$$\mathbb{E}\big[|\tilde{F}(k) - F^*(k)|\big] \leq \sqrt{\frac{\pi b^2}{2N_s(1-\gamma)^2}}, \forall k \in [M]. \quad (75)$$

Based on condition (75), we can derive the upper bound of $\mathbb{E}[|\Delta_i^3 - \Delta_i^2|]$ by splitting different situations as follows:

$$\mathbb{E}\Big[|\Delta_i^3 - \Delta_i^2|\Big] \le \sqrt{\frac{4\gamma}{\mu}\sqrt[4]{\frac{\pi b^2}{2N_s(1-\gamma)^2}}} + \frac{2C}{M}. \quad (76)$$

Combining (74), (76), and the physical limits of function $c_i(\cdot)$ yields the one-step result. Taking the expectation on all possible q_t, and conducting standard mathematical manipulation across all time yield our result. ■

E. Proof Sketch for Theorem 5

Eliminating the terms involving M, the one-step regret given sample size N_s satisfies:

$$\mathbb{E}\Big[|c\Big(\Delta_i^1\Big) - c\Big(\Delta_i^3\Big)|\Big] \le p_{\max}\left(\sqrt{\frac{4\gamma}{\mu}\sqrt[4]{\frac{\pi b^2}{2N_s(1-\gamma)^2}}}\right). \quad (77)$$

At $t = 1$, the sample size is N_s. During the control process, the available sample size at time t is $N_s + t - 1$. Thus, the one-step regret at time t equals $\mathcal{O}(\sqrt[4]{\frac{1}{N+t-1}})$.

Though the resulting SoCs of the online algorithm and the optimal algorithm during different times t are different, we can prove that such effects are minor compared with the one-step regret. By standard mathematical manipulation and eliminating minor terms, we can derive the cumulative regret as follows:

$$\mathcal{R}_\pi \le \sum_{t=1}^{T}\mathcal{O}\left(\sqrt[4]{\frac{1}{N+t-1}}\right) = \mathcal{O}\left(\frac{T}{\sqrt[4]{\sqrt[4]{N_s}+T}}\right). \quad (78)$$

This concludes our proof. ■

F. Guidelines for Storage Sizing

Storage sizing is a pivotal concern within power system infrastructure investment. The judicious determination of energy storage size holds the key to striking a balance between short-term revenue from storage operations and the long-term amortized investment costs, ultimately leading to enhanced system efficiency. Due to the physical property of energy storage, we also take the storage degradation into consideration. We first introduce how to consider the storage degradation in our model, and then introduce a storage sizing algorithm to choose the optimal storage size minimizing the average storage life-circle cost.

Specifically, there are two ways to consider the degradation cost in the literature [37]:

- *Amortized Investment Cost:* The classical and straightforward method to model degradation cost involves utilizing the amortized investment costs over the battery's entire life cycle [37]. Specifically, with any storage size C, the system first decides the expected battery life $T(C)$ and the battery investment cost $Q(C)$. Then, the amortized investment cost $Q(C)/T(C)$ is adopted to be the degradation cost. If we include this modeling into our algorithm, we only need to include the unit degradation cost into the original cost function c_t, i.e.,

$$c_t\big(\hat{w}_t, g_t\big) = p_t^+ \max\big(g_t - \hat{w}_t, 0\big) + p_t^- \max\big(\hat{w}_t - g_t, 0\big) + \frac{Q(C)}{T(C)}. \quad (79)$$

Adding this constant does not influence the optimization structure, hence has no impact on our algorithm. However, it's worth noting that this model simplifies the degradation cost and doesn't consider the impact of different storage control policies on the battery's life cycle, For example, a policy with more frequent charging/discharging behaviors will make battery life shorter, while a less frequent charging/discharging policy can potentially increase the battery life.

- *Refined Mileage Degradation Cost:* To capture the physical property of storage degradation, recent literature often uses the mileage-styled degradation cost [38], [39]. These modeling assumes there is a fixed usage mileage $M(C)$ of the storage and an investment cost $Q(C)$ regarding the storage size C. At each time slot t, the degradation cost m_t satisfies:

$$m_t = \frac{Q(C)\big(\alpha_1 v_t^+ + \alpha_2 v_t^- + \alpha_3\big)\Delta t}{M(C)}, \quad (80)$$

where v_t^+ and v_t^- denote the charging and discharging rates of storage at time t. Parameters α_1, α_2 represent the degradation ratios of charging and discharging, respectively. α_3 denotes the time-degradation ratio, and Δt is the length of a single time slot. In [39], a more refined degradation cost is proposed by setting α_1, α_2 and α_3 to be time-dependent. We can follow a similar way to incorporate the mileage degradation cost into our model. Specifically, the cost function c_t at time t should be modified into:

$$c_t\big(\hat{w}_t, g_t\big) = p_t^+ \max\big(g_t - \hat{w}_t, 0\big) + p_t^- \max\big(\hat{w}_t - g_t, 0\big) + \frac{Q(C)\big(\alpha_1 v_t^+ + \alpha_2 v_t^- + \alpha_3\big)\Delta t}{M(C)}. \quad (81)$$

It just slightly changes the form of the cost function, and does not influence the piecewise linearity of the objective function, as well as our whole algorithm.

For the storage capacity degradation, we only need to set the storage capacity C to be a time-dependent C_t. Since the storage degradation process is much slower than the storage control process, we can assume the storage capacity is consistent in a single time slot. Only when the storage capacity is significantly changed, we then adopt the new capacity C_t and run the whole algorithm to get the updated control policy, which does not influence the computational efficiency.

By the investment and degradation cost modeling, we can propose an optimal storage sizing approach. The target of storage sizing is to minimize the total cost considering both the investment/degradation costs and the revenue of storage control. However, calculating the storage control revenue often requires simulating a lot of future scenarios, which is time-consuming [38], [39]. We highlight that we can use the concept of storage value function in our paper to efficiently evaluate such revenue.

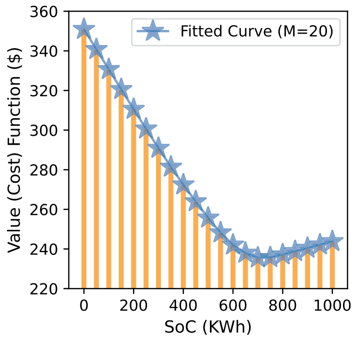


Fig. 7. Example of the cost function $F(SoC)$.

Specifically, by considering the cost function Eq. (81) incorporating the degradation cost, the cost value function $F(SoC, C)$ in our paper characterizes the future discounted penalty cost with SoC of value SoC and storage size C. An example of $F(SoC, C)$ is provided in Fig. 7. Specifically, the x-coordinate is the SoC of storage, and the y-coordinate is the corresponding discounted penalty cost accumulated in the future. We can observe that, the value function is convex and reaches the minimum when SoC is approximately 700 KWh. This suggests that it is beneficial to maintain the SoC around 700 KWh to balance risks associated with electricity shortage and surplus. And the value at such point denotes the optimal cost when equipped with the storage with size C. We use $P(C)$ to represent such cost, specifically:

$$P(C) = \min_{SoC} F(SoC, C). \tag{82}$$

When $C \to \infty$, we know $P(C) = 0$, which means the infinite size storage can entirely contain all generation mismatches and avoid all penalty costs. And $C \to 0$ denotes the case without storage, and $P(C)$ is maximized. By evaluating different C's, we can get different $P(C)$'s and finally obtain the whole continuous cost function curve $P(C)$ in terms of C.

Therefore, for a single operation time slot t, we can define the average cost function $\overline{P}(C_t)$ equals:

$$\overline{P}(C_t) = (1-\gamma)P(C_t), \tag{83}$$

where $1-\gamma$ is the standard normalization factor [40] for the Markov decision process.

Specifically, denote $T(C)$ as the expected life cycle of the storage, κ is the amortized maintenance cost of a single operation period. And $Q(C)$ is the investment cost of a storage with size C. Then we can solve the following problem to get the optimal size C^*:

$$C^* = \arg\min_{C} \quad \frac{1}{T}\left(\sum_{t=1}^{T} \overline{P}(C_t) + \kappa T(C)\right), \tag{84}$$

$$\text{s.t.} \quad C_1 = C, \tag{85}$$

$$C_t = A_t(C), \tag{86}$$

where $\sum_{t=1}^{T} \overline{P}(C_t)$ denotes the total penalty and investment cost considering storage degradation; $\kappa T(C)$ denotes the total maintenance cost. $A_t(C)$ is the capacity time-degradation curve of storage in terms of time t, which can be obtained by checking the storage's technical parameters. By solving this problem (84)–(86), we can decide the optimal storage size of minimizing the overall cost.

References

[1] I. Dincer, "Renewable energy and sustainable development: A crucial review," *Renew. Sustain. Energy Rev.*, vol. 4, no. 2, pp. 157–175, 2000.

[2] Z. Li, W. Wu, B. Zhang, and B. Wang, "Adjustable robust real-time power dispatch with large-scale wind power integration," *IEEE Trans. Sustain. Energy*, vol. 6, no. 2, pp. 357–368, Apr. 2015.

[3] M. Mahmoodi, P. Shamsi, and B. Fahimi, "Economic dispatch of a hybrid microgrid with distributed energy storage," *IEEE Trans. Smart Grid*, vol. 6, no. 6, pp. 2607–2614, Nov. 2015.

[4] Y. Shi, B. Xu, D. Wang, and B. Zhang, "Using battery storage for peak shaving and frequency regulation: Joint optimization for superlinear gains," *IEEE Trans. Power Syst.*, vol. 33, no. 3, pp. 2882–2894, May 2018.

[5] M. Uddin, M. Romlie, M. Abdullah, C. Tan, G. Shafiullah, and A. H. A. Bakar, "A novel peak shaving algorithm for islanded microgrid using battery energy storage system," *Energy*, vol. 196, Apr. 2020, Art. no. 117084.

[6] P. Malysz, S. Sirouspour, and A. Emadi, "An optimal energy storage control strategy for grid-connected microgrids," *IEEE Trans. Smart Grid*, vol. 5, no. 4, pp. 1785–1796, Jul. 2014.

[7] M. Dabbagh, A. Rayes, B. Hamdaoui, and M. Guizani, "Peak shaving through optimal energy storage control for data centers," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kuala Lumpur, Malaysia, 2016, pp. 1–6.

[8] J. Wu, Z. Wang, C. Wu, K. Wang, and Y. Yu, "A data-driven storage control framework for dynamic pricing," *IEEE Trans. Smart Grid*, vol. 12, no. 1, pp. 737–750, Jan. 2021.

[9] J. Wu, C. Lu, and C. Wu, "Learning-aided framework for storage control facing renewable energy," *IEEE Syst. J.*, vol. 17, no. 1, pp. 652–663, Mar. 2023.

[10] I. Koutsopoulos, V. Hatzi, and L. Tassiulas, "Optimal energy storage control policies for the smart power grid," in *Proc. IEEE Int. Conf. Smart Grid Commun. (SmartGridComm)*, Brussels, Belgium, 2011, pp. 475–480.

[11] C.-K. Chau, G. Zhang, and M. Chen, "Cost minimizing online algorithms for energy storage management with worst-case guarantee," *IEEE Trans. Smart Grid*, vol. 7, no. 6, pp. 2691–2702, Nov. 2016.

[12] L. Huang, J. Walrand, and K. Ramchandran, "Optimal demand response with energy storage management," in *Proc. IEEE Int. Conf. Smart Grid Commun. (SmartGridComm)*, Tainan, Taiwan, 2012, pp. 61–66.

[13] J. Qin, Y. Chow, J. Yang, and R. Rajagopal, "Online modified greedy algorithm for storage control under uncertainty," *IEEE Trans. Power Syst.*, vol. 31, no. 3, pp. 1729–1743, 2015.

[14] J. Qin, Y. Chow, J. Yang, and R. Rajagopal, "Distributed online modified greedy algorithm for networked storage operation under uncertainty," *IEEE Trans. Smart Grid*, vol. 7, no. 2, pp. 1106–1118, Mar. 2016.

[15] W. Zhong, K. Xie, Y. Liu, C. Yang, S. Xie, and Y. Zhang, "Online control and near-optimal algorithm for distributed energy storage sharing in smart grid," *IEEE Trans. Smart Grid*, vol. 11, no. 3, pp. 2552–2562, May 2020.

[16] S. Grillo, A. Pievatolo, and E. Tironi, "Optimal storage scheduling using Markov decision processes," *IEEE Trans. Sustain. Energy*, vol. 7, no. 2, pp. 755–764, Apr. 2016.

[17] S. Dimopoulou, A. Oppermann, E. Boggasch, and A. Rausch, "A Markov decision process for managing a hybrid energy storage system," *J. Energy Stor.*, vol. 19, pp. 160–169, Oct. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352152X17303341

[18] Y.-K. Wu and J.-S. Hong, "A literature review of wind forecasting technology in the world," in *Proc. IEEE Lausanne Power Tech*, Lausanne, Switzerland, 2007, pp. 504–509.

[19] J. Kehler, M. Hu, M. McMullen, and J. Blatchford, "ISO perspective and experience with integrating wind power forecasts into operations," in *Proc. IEEE PES Gener. Meet.*, Minneapolis, MN, USA, 2010, pp. 1–5.

[20] (California Indep. Syst. Oper. Corp., Folsom, CA, USA). *Electricity Price Data*. (2021). Accessed: Dec. 20, 2022. [Online]. Available: https://www.energyonline.com/Data/

[21] P. Lu, L. Ye, Y. Zhao, B. Dai, M. Pei, and Y. Tang, "Review of meta-heuristic algorithms for wind power prediction: Methodologies, applications and challenges," *Appl. Energy*, vol. 301, Nov. 2021, Art. no. 117446.

[22] A. Schrijver, *Theory of Linear and Integer Programming*. Hoboken, NJ, USA: Wiley, 1998.

[23] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[24] E. V. Denardo, "On linear programming in a Markov decision problem," *Manage. Sci.*, vol. 16, no. 5, pp. 281–288, 1970.

[25] S. Shalev-Shwartz, "Online learning and online convex optimization," *Found. Trends® Mach. Learn.*, vol. 4, no. 2, pp. 107–194, 2012, doi: 10.1561/2200000018.

[26] J. Zhou et al., "SDWPF: A dataset for spatial dynamic wind power forecasting challenge at KDD cup 2022," 2022, *arXiv:2208.04360*.

[27] B. Erisen, 2023, "Wind turbine scada dataset," kaggle. [Online]. Available: https://www.kaggle.com/datasets/berkerisen/wind-turbine-scada-dataset/data

[28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[29] V.-H. Bui, A. Hussain, and H.-M. Kim, "Double deep *Q*-learning-based distributed operation of battery energy storage system considering uncertainties," *IEEE Trans. Smart Grid*, vol. 11, no. 1, pp. 457–469, Jan. 2020.

[30] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, no. 2, pp. 281–305, 2012.

[31] S. Diamond and S. Boyd, "CVXPY: A python-embedded modeling language for convex optimization," *J. Mach. Learn. Res.*, vol. 17, no. 83, pp. 1–5, 2016.

[32] D. Ge, Q. Huangfu, Z. Wang, J. Wu, and Y. Ye, "Cardinal Optimizer (COPT) user guide." 2023. [Online]. Available: https://guide.coap.online/copt/en-doc

[33] T. K. Brekken, A. Yokochi, A. Von Jouanne, Z. Z. Yen, H. M. Hapke, and D. A. Halamay, "Optimal energy storage sizing and control for wind power applications," *IEEE Trans. Sustain. Energy*, vol. 2, no. 1, pp. 69–77, Jan. 2011.

[34] A. Giannitrapani, S. Paoletti, A. Vicino, and D. Zarrilli, "Wind power bidding in a soft penalty market," in *Proc. 52nd IEEE Conf. Decis. Control*, Firenze, Italy, 2013, pp. 1013–1018.

[35] E. Y. Bitar, R. Rajagopal, P. P. Khargonekar, K. Poolla, and P. Varaiya, "Bringing wind energy to market," *IEEE Trans. Power Syst.*, vol. 27, no. 3, pp. 1225–1235, Aug. 2012.

[36] R. Vershynin, *High-Dimensional Probability: An Introduction With Applications in Data Science*, vol. 47. Cambridge, U.K.: Cambridge Univ. Press, 2018.

[37] P. L. C. García-Miguel, J. Alonso-Martínez, S. A. Gómez, M. G. Plaza, and A. P. Asensio, "A review on the degradation implementation for the operation of battery energy storage systems," *Batteries*, vol. 8, no. 9, p. 110, 2022.

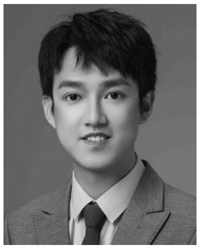
[38] Y. Pu et al., "Optimal sizing for an integrated energy system considering degradation and seasonal hydrogen storage," *Appl. Energy*, vol. 302, Nov. 2021, Art. no. 117542.

[39] G. He, S. Kar, J. Mohammadi, P. Moutis, and J. F. Whitacre, "Power system dispatch with marginal degradation cost of battery storage," *IEEE Trans. Power Syst.*, vol. 36, no. 4, pp. 3552–3562, Jul. 2021.

[40] R. Bellman, "A Markovian decision process," *J. Math. Mech.*, vol. 6, no. 5, pp. 679–684, 1957.

Chenbei Lu (Graduate Student Member, IEEE) received the bachelor's degree from the School of Software Engineering, Huazhong University of Science and Technology. He is currently pursuing the Ph.D. degree in computer science and technology with the Institute for Interdisciplinary Information Sciences, Tsinghua University advised by Prof. C. Wu.

He is currently working on the optimal design and operation of power systems. He has been awarded the National Scholarship in 2017 and the Excellent Graduate of the Huazhong University of Science and Technology in 2020.



Hongyu Yi (Graduate Student Member, IEEE) received the bachelor's degree in applied mathematics from SSE, The Chinese University of Hong Kong (Shenzhen) in 2023, where he is currently pursuing the M.Phil. degree in computer and information engineering with the School of Science and Engineering, advised by Prof. C. Wu.

His research interests include control in power systems, online algorithms, and online learning. He has been awarded Bowen Scholarship in 2019.

Jiahao Zhang received the bachelor's degree from the School of New Energy Science and Engineering, North China Electrical Power University and the master's degree from the Department of Engineering, Imperial College London. He is currently pursuing the Ph.D. degree in computer and information engineering with the School of Science and Engineering, The Chinese University of Hong Kong (Shenzhen) advised by Prof. C. Wu.

His research interests include artificial intelligence, privacy preservation, and optimization in power systems. He has been awarded Excellent Graduate in 2021.

Chenye Wu (Senior Member, IEEE) received the Ph.D. degree from the Institute for Interdisciplinary Information Sciences (IIIS), Tsinghua University in July 2013.

He is an Assistant Professor with the School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen (CUHK Shenzhen). Before joining CUHK Shenzhen, he was an Assistant Professor with IIIS, Tsinghua University. He worked with ETH Zurich as a wiss. Mitarbeiter (Research Scientist), working with Prof. Gabriela Hug in 2016. Before that, Prof. K. Poolla and Prof. P. Varaiya hosted him as a Postdoctoral Researcher with the University of California at Berkeley for two years. From 2013 to 2014, he spent one year with Carnegie Mellon University as a Postdoctoral Fellow, hosted by Prof. G. Hug and Prof. S. Kar. He is currently working on economic analysis, optimal control, and operation of power systems. His Ph.D. advisor is Prof. A. Yao, the Laureate of the A.M. Turing Award in the year of 2000. He was the Best Paper Award Co-Recipients of IEEE SmartGridComm 2012, IEEE PES General Meeting 2013, and IEEE PES General Meeting 2020.