A Lagrangian Policy for Optimal Energy Storage Control

Bolun Xu, Magnus Korpås, Audun Botterud, Francis O’Sullivan

Abstract—This paper presents a millisecond-level look-ahead control algorithm for energy storage. The algorithm connects the optimal control with the Lagrangian multiplier associated with the state-of-charge constraint. It is compared to solving look-ahead control using a state-of-the-art convex optimization solver. The paper include discussions on sufficient conditions for including the non-convex simultaneous charging and discharging constraint, and provide upper and lower bounds for the primal and dual results under such conditions. Simulation results show that both methods obtain the same control result, while the proposed algorithm runs up to 100,000 times faster and solves most problems within one millisecond. The theoretical results from developing this algorithm also provide key insights into designing optimal energy storage control schemes at the centralized system level as well as under distributed settings.

Index Terms—Energy systems, Numerical algorithm, Predictive control for nonlinear systems

I. INTRODUCTION

Energy storage devices such as batteries are key resources in future energy systems due to their flexibility and fast response speed, and their convenient installations as either large-scale bulk units or as distributed resources. Most real-time energy storage operations are optimized using predictive control, with applications such as economic dispatch [1], frequency control [2], voltage control [3], renewable integration [4], energy arbitrage [5], peak shaving [6], electric vehicle charging [7], or a combination of several aforementioned applications [8]. These predictive control strategies solve a multi-period optimization problem over a look-ahead horizon at each control step, obtaining the control and state profile over the entire horizon but only applies the first control result, the problem is then updated with a new horizon and state information for the next control step.

The challenge of using look-ahead control in practice is trading off optimality with computational tractability, as a longer look-ahead horizon incorporates more future information and thus improves solution optimality, but increases the computational challenge significantly. For example, real-time economic dispatches in power systems are typically solved over a single period or with a look-ahead horizon less than one hour [9]. However, power system operations have strong daily patterns due to load and weather variations, such as charging storage from solar power during the day and discharge during the night. Thus, being able to incorporate a look-ahead horizon over one day or even longer is crucial for the future power system, but solving such problems over the scale of a realistic power system is extremely computationally challenging [10], especially binary variables must be introduced in certain application to prevent simultaneous charging and discharging, making the problem non-convex [11]. In addition, future uncertainties in power systems are often modeled with scenarios [12], and modeling uncertainties from multiple sources can easily lead to hundreds of scenarios that makes it almost impossible to solve look-ahead economic dispatch with conventional optimization solvers. While methods such as stochastic dual dynamic programming [13] reduce the solution complexity by introducing inter-temporal and scenario decomposition, the computation is still difficult and requires significant memory usage. On the other hand, the optimal control problem must be solved within a reasonable timescale to fully utilize the fast response speed of energy storage devices. For example, a battery ramps from zero to full discharge power within milliseconds [14] thus, a scheme that takes seconds or even minutes to update the control decision is not appropriate for controlling batteries.

Solving storage control from the dual problem is more effective than dealing with the primal problem directly since the storage has only a single state variable with upper and lower bounds. Cruise et al [15] concluded the storage control problem can be solved using a search algorithm based on the binding conditions on the state-of-charge, and Hashimi et al [16] has developed an algorithm for energy storage price arbitrage with quadratic time complexity, based on solving the dual problem. Comparably, the technical contributions of this paper and the main advantages of the proposed algorithm is summarized as follows:

1) We show that energy storage control with a generalized time-varying objective functions can be solved in worst-case linear time complexity and constant space complexity, with respect to the look-ahead horizon.

2) We conclude the optimal control condition for energy storage without having to go through the full look-ahead horizon, i.e., the current control is optimal with respect to any future realizations that may not be included in the current look-ahead window.

3) We derive a sufficient condition for the occurrence of simultaneous charging and discharging, and provide upper and lower bounds for the prime and dual results under such non-convex conditions.

The rest of this paper is organized as follows: Section II formulates the problem; Section III presents main analytical results and the algorithm; Section IV demonstrates numerical results; and Section V concludes the paper.

B. Xu, A. Botterud, and F. O’Sullivan are with Massachusetts Institute of Technology, MA, USA. M. Korpås is with Norwegian University of Science and Technology, Trondheim, Norway. Contact: \{xubolun, audunb,frankie\}@mit.edu, magnus.korpaas@ntnu.no.
II. Formulation and Preliminaries

A. Problem Formulation

We consider a time period \( t \in T = \{1, \ldots, T\} \) where \( t = 1 \) is the current control step and \( t = 2 \) to \( T \) is the look-ahead horizon. The optimal control profile \( p_t^* \) is a minimizer to the following multi-period optimization problem

\[
p_t^* \in \arg\min_{p_t} \sum_{t=1}^{T} O_t(p_t) + C_T(e_T) \quad (1a)
\]

subject to

\[
p_t = p_t^+ - p_t^- \quad (1b)
\]

\[
p_t^+ = 0 \text{ or } p_t^- = 0 \forall t \in T \quad (1c)
\]

\[
e_t - e_{t-1} = -\frac{p_t^+}{\eta} + p_t^- \quad (1d)
\]

\[
p_t^+ \geq 0 : \mu^+, \quad p_t^- \leq P : \nu^+ \quad (1e)
\]

\[
e_t \geq 0 : \lambda, \quad e_t \leq E : \lambda_t \quad (1f)
\]

where

1) \( O_t(\cdot) : \mathbb{R} \to \mathbb{R} \) is a scalar time-varying convex objective function. Its derivative is denoted as \( o_t = \dot{O}_t \).

2) \( C_T(\cdot) : \mathbb{R} \to \mathbb{R} \) is the terminal cost function of the end state of charge \( e_T \). \( C_T \) is also convex and its derivative is denoted as \( e_T = \dot{C}_T \). Note that \( C_T \) can also be used to model the operation beyond \( T \) via dynamic programming [17].

3) \( p_t \) is the control decision variable and it is the energy dispatched from the storage during the time period \( t \).

4) \( p_t^+ \) is the positive (discharge) component of \( p_t \).

5) \( p_t^- \) is the negative (charge) component of \( p_t \).

6) \( e_t \) is the state of charge (SoC) at the end of time period \( t \), subject to an initial value of \( e_0 \).

7) \( \eta \in (0, 1] \) is the storage charge and discharge efficiency.

8) \( P \in \mathbb{R}^+ \) is the maximum energy that can be charged or discharged into the storage during a single period.

9) \( E \in \mathbb{R}^+ \) is the maximum energy that can be stored in the storage.

10) \( p_t^* \) is a set of minimizers to the optimization problem.

11) \( \theta_t \) is the Lagrangian multiplier associated with the SoC dynamic, its physical meaning is the marginal value of SoC at the end of time \( t \) over the future operation \([t + 1, T]\).

12) \( \mu^+, \nu^+, \lambda, \lambda_t, \nu_t \) are positive dual variables associated with inequality constraints.

The objective function \((1a)\) minimizes the total operating cost over the period \([1, T]\). Constraint \((1b)\) divides the control \( p_t \) into a positive component \( p_t^+ \) and a negative component \( p_t^- \) in order to model the efficiency difference during charge and discharge in the SoC evolution constraint. \((1c)\) is the non-simultaneous charging and discharging constraint that enforces the storage to only charge or discharge at any given time point. \((1d)\) models the SoC evolution subjects to efficiencies. Power and energy ratings are modeled in \((1e)\).

B. Karush-Kuhn-Tucker conditions

The results in this paper are primarily based on the use of the Karush-Kuhn-Tucker (KKT) conditions [18], which are listed below for \((\text{I})\) (the non-simultaneous charging and discharging constraint \((\text{Ic})\) is non-convex and is excluded from the KKT condition below, treatment of this constraint will be discussed later):

\[
o_t(p_t^*) + \mu^+ + \theta_{t-1}/\eta = 0 : p_t^+ \quad (2a)
\]

\[
-o_t(p_t^*) + \nu^+ - \theta_{t-1}/\eta = 0 : p_t^- \quad (2b)
\]

\[
\theta_{t-1} + \nu - \theta_t = 0 : e_t \quad (2c)
\]

\[
\theta_T + e_T(e_T^*) = 0 : e_T \quad (2d)
\]

and the complimentary slackness conditions associated with the inequality dual variables:

\[
\mu^+[p_t^+] = 0, \quad \nu^+([-p_t^+]^+) = 0, \quad \mu^+([-p_t^-]^+) = 0, \quad \nu^+(P - [p_t^+]^+) = 0, \quad \nu^+(P - [-p_t^-]^+) = 0,
\]

\[
\mu^+ e_t^* = 0, \quad \nu^+(E - e_t^*) = 0,
\]

and \( \mu^+, \nu^+, \lambda, \lambda_t, \nu_t \geq 0 \). Note that we replaced the use of \( p_t^+ \) and \( p_t^- \) with \( p_t \) since \( \partial p_t^+ = \partial p_t \) and \( \partial p_t^- = -\partial p_t \).

Then we discuss how we can bound the result when it is necessary to incorporate the non-convex non-simultaneous charging and discharging constraint \((\text{Ic})\) using the proposed algorithm.

A. Optimal Control Policy

We define the policy \( \pi \) that calculates a storage control decision \( p_t^\pi(x) \) for time \( t \) from an input \( x \in \mathbb{R} \) as

\[
p_t^\pi(x) = \left[ \frac{\varphi_t(x/\eta)}{o_t} \right]_0^P, \quad p_t^\pi(x) = \left[ -\varphi_t(-x\eta) \right]_0^P
\]

\[
p_t^\pi(x) = p_t^+(x) - p_t^-(x)
\]

where \( [x]^z = \max\{\min\{x, y\}, z\} \) saturates \( x \) between \( y \) and \( z \) (\( z \leq y \)), and \( \varphi_t(x) : \mathbb{R} \to \mathbb{R} \) is the inverse function of \( o_t \) (derivative of \( O_t \)) as

\[
\varphi_t(x) = \sup\{y \in \mathbb{R} | o_t(y) \leq x\}.
\]

Note that \( \varphi_t \) is an alternative definition of the inverse function to \( o_t \) while compatible with a piecewise linear \( O_t \).

The following proposition states that we can obtain the optimal control \( p_t^* \) by using the Lagrangian multiplier \( \theta_{t-1} \) as input to policy \( \pi \):

**Proposition 1.** Policy \( \pi \) is a minimizer to problem \((\text{I})\) when using the Lagrangian \( \theta_{t-1} \) as the input, i.e., \( p_t^* = p_t^\pi(\theta_{t-1}) \).
Proof. We start by rewriting the KKT condition associated with $p_t^+$ as

\begin{align}
\alpha(t) p_t^+ + \theta_{t-1}/\eta &< 0 \quad \text{if} \quad p_t^+ = P \\
\alpha(t) p_t^+ + \theta_{t-1}/\eta &> 0 \quad \text{if} \quad p_t^+ = 0 \\
\alpha(t) p_t^+ + \theta_{t-1}/\eta &= 0 \quad \text{if} \quad 0 \leq \varphi_t(-\theta_{t-1}/\eta) \leq P
\end{align}

where we substitute the complementary slackness condition into (2a) that replaces $\mu_t^+$ and $\mu_t^-$. It is now trivial to see that we can calculate $p_t^+$ as $\varphi_t(-\theta_{t-1}/\eta)$ and limiting the result between 0 and $P$, hence

\begin{equation}
\begin{aligned}
p_t^+ = \max\{0, \min\{P, \varphi_t(-\theta_{t-1}/\eta)\}\}.
\end{aligned}
\end{equation}

We repeat the similar process for $p_t^-$ with (2b), and use $p_t = p_t^+ - p_t^-$ which gives us the result in Proposition 1.

The following corollary supplements that with $\theta_0$ we can obtain $p_t^+$ as well as a series of consecutive optimal control decisions by recording the accumulated sum of the control results $\sigma_t$ defined as

\begin{equation}
\sigma_t(x) = \sigma_{t-1}(x) - [p_t^-(x)]^+/\eta - [p_t^+(x)]^-\eta,
\end{equation}

with the initial value $\sigma_0(x) = \epsilon_0$, where $[x]^+$ is the positive value function, and $[x]^-$ is the negative value function. $\sigma_t(x)$ emulates the SoC evolution but using the control result $p_t^+(x)$ which may not be optimal. Another difference is that $\sigma_t(x)$ is not limited between $[0, E]$, instead, whether any $\sigma_t(x)$ falls above $E$ or below 0 is an indicator on the optimality of $p_t^+(x)$, as defined by the following corollary:

**Corollary 2.** $p_t^+ = p_t^+(\theta_0)$ if $0 \leq \sigma_t(\theta_0) \leq E \forall \tau \in [1, t]$.

Corollary 2 means that we can maintain optimal control by using $\theta_0$ as the input to $\varphi_t$ for control steps beyond $t = 1$ if all previous $\sigma_t(\theta_0)$ are within the SoC constraint. Corollary 2 is based on Proposition 1 and the KKT condition associated with $e_t$ in (2) that the $\theta_t$ value will not change if both $\tau_t$ and $\nu_t$ are zeros, indicating $0 \leq e_t^+ \leq E$ and $\sigma_t(x) = e_t^+$. This corollary is thus proved.

**B. Main Theorem on Finding Lagrangian Dual**

**Theorem 3.** Given $x \in \mathbb{R}$, its equality relationship with respect to $\theta_0$ can be determined as

1) If a) $\sigma_t(x)$ reaches upper bound first, i.e., $\exists \tau \in T \ s.t. \ \sigma_t(x) > E$ and $0 \leq \sigma_t(x) \leq E \ \forall \gamma \in [1, \tau]$; or b) $\sigma_t(x)$ reached neither bound and $x > -c_{tr}(\sigma_t(x))$; then $x \geq \theta_0$;

2) If a) $\sigma_t(x)$ reaches lower bound first, i.e., $\exists \tau \in T \ s.t. \ \sigma_t(x) < 0$ and $0 \leq \sigma_t(x) \leq E \ \forall \gamma \in [1, \tau]$; or
b) \( \sigma_t(x) \) reached neither bound and \( x < -c_T(\sigma_T(x)) \); then \( x = \theta_0 \).

3) If \( \sigma_t(x) \) reached neither bound and \( x = -c_T(\sigma_T(x)) \), then \( x = \theta_0 \).

Proof of this theorem is deferred to Appendix. The intuition is that the Lagrangian dual is the price of the stored energy, its value does not change despite the change with the SoC evolution, except reaching either the upper or lower SoC bound. The SoC series driven by the optimal dual value should never exceed the SoC bounds as the dual value itself reflects the constrained storage capacity. If the SoC exceed the upper SoC bound, it means SoC value is over estimated as the storage does not have enough capacity to store the excessive energy, hence we picked an \( x \) that is higher than the optimal Lagrangian dual value. Vice versa, if the SoC exceed the lower bound, mean we under estimated the dual value.

C. Solution Algorithm

We design a binary search algorithm that finds \( \theta_0 \) according to Theorem 3, thus we find \( p_t = p_t^*(\theta_t) \) (Proposition 1) as well as some consecutive optimal control actions (Corollary 2) without needing to explicitly solve Problem (1). The algorithm requires a preset search accuracy \( \epsilon \) and is described as follows:

1) Initialize a search range \( L \) and \( R \) with which we are confident that \( L \leq \theta_0 \leq R \);
2) Set \( x \rightarrow (L+R)/2 \). If \( R - L < \epsilon \), return \( x \) as the optimal Lagrangian dual value and \( p_t^*(x) \) as the optimal storage control up to time step \( t \);
3) Run the following iterative simulation
   a) Set \( t \rightarrow t+1 \) and \( e_0 \rightarrow \sigma_0(x) \);
   b) Calculate \( p_t^*(x) \) using Eq. (5);
   c) Calculate \( \sigma_t(x) \) using Eq. (3);
   d) If \( \sigma_t(x) \geq T \), set \( x \rightarrow R \), go to Step 2);
   e) If \( \sigma_t(x) \leq 0 \), set \( x \rightarrow L \), go to Step 2);
   f) If \( t < T \), set \( t + 1 \rightarrow t \) and go to Step b);
   g) If \( x \geq -c_T(\sigma_T(x)) \), set \( x \rightarrow R \);
   h) If \( x < -c_T(\sigma_T(x)) \), set \( x \rightarrow L \);
4) Go to Step 2.

An example of a confident search range is that we can assume stored energy always has a positive value and choose \( L = 0 \) and \( R = \max\{a_t(p)/\eta| t \in T, p \in [-P, P]\} \). This algorithm achieves the following complexity results:

1) Constant space complexity: The algorithm achieves \( O(1) \) space complexity with respect to the search range and the look-ahead duration \( T \), because the equality relationship between \( x \) and \( \theta_0 \) can be identified using only the current simulation result \( \sigma_t(x) \) so that previous simulation results are not required to be stored.
2) Worst-case linear run-time complexity: The algorithm achieves a worst-case \( O(n) \) complexity with respect to the look-ahead horizon \( T \) since the worst-case scenario is to simulate all operations steps from \( t = 1 \) to \( T \) during each search, but may terminate before reaching \( T \) as stated in step 3-d and 3-e. It also achieves \( O(\log n) \) time complexity with respect to the search range for using a binary search algorithm.

D. Non-simultaneous Charging and Discharging

By far we have concluded the optimal storage control when relaxing the non-simultaneous charging and discharging constraint (1b). A sufficient condition for relaxing this constraint without sacrificing result optimality is illustrated in the following proposition:

Proposition 4. A sufficient condition for simultaneous charging and discharging to happen is the Lagrangian dual being negative, i.e., if \( \theta_{t-1} \geq 0 \) then \( p_t^+ p_t^- = 0 \).

Proof. From (5), it is trivial to see that for simultaneous charging and discharge to happen, both terms in (5) must take non-zero values, hence

\[
\varphi_t(-x\eta) < 0 < \varphi_t(-x/\eta)
\]

and since \( \varphi_t \) is a monotonic increasing function and \( 0 < \eta \leq 1 \), it follows

\[
-x\eta < -x/\eta
\]

hence \( x \) must be less than zero. \( \square \)

Recall that \( \theta_t \) is the value of the stored energy, hence \( \theta_t \) being negative indicates the stored energy has a negative value, i.e., we have an intention to store as less energy as possible. Thus, when (1b), the storage can charging and discharging at the same time and use round-trip efficiency loss to consume excessive energy even when the storage is full and has no more storage space. This intuition is useful when deciding whether simultaneous charging and discharging should be considered when formulating the problem, for example, this constraint should be considered in price arbitrage for markets with frequent negative prices.

A common method for enforcing non-simultaneous charging and discharging is to add auxiliary binary variables \( v = \{v_t\} \) such that the storage can only charge or discharge at one time, as

\[
p_t^+ <= P v_t, \quad p_t^- <= P (1 - v_t) \quad v_t \in \{0, 1\}
\]

making (1) a mixed-integer programming problem, the Lagrangian dual can thus be calculated given a set of fixed \( v \).

We assume set \( V \) as the set of all reasonable charging status results, in which \( v_t \) may be either 0 or 1 during all periods when simultaneous charging and discharge occur, i.e., when (1c) must be enforced. It is worth noting that the optimal result for \( v \) must be in \( V \). We denote \( \theta_0(v) \) as the resulting Lagrangian dual associated with the charging status \( v \), then the following proposition stands:

Proposition 5. For all \( v \in V, \theta_0 \leq \theta_0(v) \leq \theta_0 \), where \( \theta_0(v) \) is calculated using Algorithm 1 by replacing (5) with

\[
p_t^-(x) = p_t^+(x) 1_{[p_t^-(x) = 0]} - p_t^-(x)
\]
and \( \theta_0 \) is calculated using Algorithm 1 by replacing (5) with
\[
p_i^\tau(x) = p_i^\tau(x) - p_i^{-}(x) \mathbf{1}_{[p_j^\tau(x) = 0]} \tag{13b}
\]
where \( \mathbf{1}_{[x]} = \{0 \text{ if } x \text{ is true and 1 otherwise}\} \) is the indicator function.

**Proof.** First note that compared to (5), (13b) enforces the storage to charge whenever charging and discharging components are both non-zero. Thus when using (13b) to simulate the battery operation in Algorithm 1, the resulting SoC must always be lower than using any \( v \in \mathcal{V} \). This accuracy of the search algorithm is set to \( \epsilon = 10^{-3} \).

Hence Proposition 5 and Proposition 6 provide bounds for the primal and dual result of the non-convex battery control problem using the proposed algorithm that has worst-case linear time complexity.

**IV. NUMERICAL SIMULATION**

We use randomly generated data sets to compare the proposed algorithm to solving Problem [1] with different objectives using Gurobi [19] (model generated using CVX [20]). All simulations are performed in Matlab [21] on a 2.3 GHz machine with 16GB memory. The storage parameter is set as \( P = 1 \text{ p.u., } E = 4 \text{ p.u., } c_0 = 4 \text{ p.u., } \eta = 0.92 \), and the terminal cost function is set to \( C_T(v(T)) = (E - c_T)^2 / 2 \). The accuracy of the search algorithm is set to \( \epsilon = 10^{-3} \).

**A. Piece-wise linear objectives**

In this section the proposed algorithm is compared with Gurobi using piecewise linear objective function inspired by the supply curves in power system economic dispatches [22]. The derivative of the objective function \( o_i \) is written as
\[
o_i(p) = c_j \text{ if } q_{j-1} \leq p < q_j \tag{15}\]
where \( j \in [1, J] \) is the piecewise segment index, \( J \) is the number of segments, \( c_j \) is the marginal cost (calculated below of the system when

**TABLE I**

| Trials | CVX+Gurobi Proposed
|-------|-------------------------|
|       | \( \theta_0 \) | \( p_i^\tau \) | \( \text{cpu [ms]} \) | \( \theta_0 \) | \( p_i^\tau \) | \( \text{cpu [ms]} \) |
|-------|-------|-------|-------|-------|-------|-------|
| \( T = 10, J = 100 \) |       |       |       |       |       |       |
| 1     | 17.53 | 0.6   | 408.8 | 17.53 | 0.6   | 0.1   |
| 2     | 16.82 | 0.1   | 278.1 | 16.82 | 0.0   | 0.1   |
| 3     | 14.59 | 0.3   | 274.5 | 14.59 | 0.3   | 0.1   |
| 4     | 14.21 | 0.1   | 281.7 | 14.21 | 0.1   | 0.1   |
| 5     | 7.27  | -0.5  | 283.7 | 7.27  | -0.5  | 0.1   |
| \( T = 10, J = 1000 \) |       |       |       |       |       |       |
| 6     | 18.67 | 0.0   | 1851.5| 18.66 | 0.0   | 0.1   |
| 7     | 20.28 | 0.8   | 1791.5| 20.28 | 0.8   | 0.2   |
| 8     | 19.50 | 0.0   | 1788.2| 19.50 | 0.0   | 0.1   |
| 9     | 17.44 | -0.7  | 1868.4| 17.44 | -0.7  | 0.2   |
| 10    | 19.00 | 0.0   | 1868.7| 19.00 | 0.0   | 0.2   |
| \( T = 100, J = 1000 \) |       |       |       |       |       |       |
| 11    | 19.09 | 0.6   | 18497.7| 19.09 | 0.6   | 0.1   |
| 12    | 19.95 | 0.8   | 19108.2| 19.95 | 0.8   | 0.2   |
| 13    | 19.44 | 0.0   | 18786.1| 19.44 | 0.0   | 0.1   |
| 14    | 18.91 | -1.0  | 19263.2| 18.91 | -1.0  | 0.2   |
| 15    | 19.55 | 0.0   | 19080.4| 19.55 | 0.0   | 0.2   |

Fig. 2. Three examples of generated piecewise linear cost curves plotted as derivative of the objective function \( o_i \).
minutes to solve the problem depending on the problem size
the benchmark method using Gurobi may need up to several
constraint (1c). In all test trails, the primal and dual result fall
solving (1) using the integer constraint (12) for enforcing
the proposed method all solves less than 1 millisecond, while
within the calculated range. In terms of computation speed,
control, where
tails including upper and lower bounds on the dual and
constrained. Table II shows the simulation result for five
simultaneous charging and discharging will occur if not
This will result in negative dual prices for the storage and
the storage wishes to absorb as much energy as possible.

B. Negative Lagrangian dual example

We consider the following quadratic objective function
\[ O_t = \frac{\alpha_t}{2} (\beta_t - p)^2 \]  
(16)
where \( \alpha_t \) are randomly generated between \([0, 10]\), and \( \beta_t \)
between \([-10, 0]\). Recall that negative sign is for charging
the battery, hence this is a generation tracking problem where
the storage wishes to absorb as much energy as possible.
This will result in negative dual prices for the storage and
simultaneous charging and discharging will occur if not
constrained. Table I shows the simulation result for five
trails including upper and lower bounds on the dual and
control, where \( \theta_t^0 \) and \( p_t^1 \) is calculated using Gurobi with
mixed-integer quadratic programming under default settings,
solving (1) using the integer constraint (12) for enforcing
constraint (1c). In all test trails, the primal and dual result fall
within the calculated range. In terms of computation speed,
the proposed method all solves less than 1 millisecond, while
the benchmark method using Gurobi may need up to several
minutes to solve the problem depending on the problem size
(thousands of steps), due to solving a mixed-integer quadratic
programming problem.

V. CONCLUSION

This paper proposed a novel algorithm for solving look-
ahead control for energy storage. The numerical results
illustrate that the algorithm provides computation speed in
milliseconds for controlling a single energy storage device
over an extended planning period. In future research, we
plan on expanding this method to controlling multiple energy
storage devices subject to network constraints. Moreover,
using the generalized terminal state function we plan on
incorporating this algorithm into scenario-based stochastic
programming or dynamic programming. In addition, our
results connects the optimal control with the Lagrangian
multiplier associated with the state-of-charge constraint, which
we will further explore to provide key insights into designing
future electricity pricing and distributed control schemes.

REFERENCES

[1] M. Korpas and A. T. Holen, “Operation planning of hydrogen storage
connected to wind power operating in a power market,” IEEE Trans-
actions on Energy Conversion, vol. 21, no. 3, pp. 742–749, 2006.
[2] N. Li, C. Zhao, and L. Chen, “Connecting automatic generation control
and economic dispatch from an optimization view,” IEEE Transactions
on Control of Network Systems, vol. 3, no. 3, pp. 254–264, 2016.
[3] B. Zhang, A. Y. Lam, A. D. Domínguez-García, and D. Tse, “An optimal
distributed method for voltage regulation in power distribution
systems,” IEEE Transactions on Power Systems, vol. 30, no. 4,
pp. 1714–1726, 2015.
[4] M. Khalid and A. Savkin, “A model predictive control approach to the
problem of wind power smoothing with controlled battery storage,”
Renewable Energy, vol. 35, no. 7, pp. 1520–1526, 2010.
[5] D. Krishnamurthy, C. Uckun, Z. Zhou, P. R. Thimmapuram, and
A. Botterud, “Energy storage arbitrage under day-ahead and real-time
price uncertainty,” IEEE Transactions on Power Systems, vol. 33, no. 1,
pp. 84–93, 2018.
[6] Y. Shi, B. Xu, D. Wang, and B. Zhang, “Using battery storage for peak
shaving and frequency regulation: Joint optimization for superlinear
gains,” IEEE Transactions on Power Systems, vol. 33, no. 3, pp. 2882–
2894, 2018.
[7] Z. Xu, W. Su, Z. Hu, Y. Song, and H. Zhang, “A hierarchical
framework for coordinated charging of plug-in electric vehicles in
china,” IEEE Transactions on Smart Grid, vol. 7, no. 1, pp. 428–438,
2016.
[8] O. Mégel, J. L. Mathieu, and G. Andersson, “Scheduling distributed
energy storage units to provide multiple services,” in Power Systems
Computation Conference (PSCC), 2014. IEEE, 2014, pp. 1–7.
[9] EPRI, “Wholesale electricity market design initiatives in the united
states: Survey and research needs,” 2016, [Online]. Available:
https://www.epri.com/#/pages/product/3002009273/
[10] J. Zhao, T. Zheng, and E. Litvinov, “A multi-period market design
for markets with intertemporal constraints,” arXiv preprint
arXiv:1812.07034, 2018.
[11] A. Castillo and D. F. Gayme, “Profit maximizing storage allocation
in power grids,” in 52nd IEEE Conference on Decision and Control.
IEEE, 2013, pp. 429–435.
[12] J. Wang, A. Botterud, R. Bessa, H. Keiko, L. Carvalho, D. Issicaba,
J. Sumaili, and V. Miranda, “Wind power forecasting uncertainty and
unit commitment,” Applied Energy, vol. 88, no. 11, pp. 4014–4023,
2011.
[13] A. Papavasiliou, Y. Mou, L. Cambrier, and D. Scieur, “Application of
stochastic dual dynamic programming to the real-time dispatch of
storage under renewable supply uncertainty,” IEEE Transactions on
Sustainable Energy, vol. 9, no. 2, pp. 547–558, 2018.
[14] L. Gao, S. Liu, and R. A. Dougal, “Dynamic lithium-ion battery
model for system simulation,” IEEE transactions on components and
packaging technologies, vol. 25, no. 3, pp. 495–505, 2002.
We start by showing that when moving to the next control step, the value of the Lagrangian \( \theta_t \) only changes after \( e^*_t \) is reaching the upper or lower SoC bound, or more specifically:

\[
\begin{align*}
\theta_{t-1} &= \theta_t & \text{if } \mathbf{p}_t = \mathbf{\kappa}_t = 0 \\
\theta_{t-1} &< \theta_t & \text{if } \mathbf{p}_t > 0 \\
\theta_{t-1} &> \theta_t & \text{if } \mathbf{\kappa}_t < 0
\end{align*}
\]

Hence, it is trivial to see that if \( 0 \leq \sigma_t(\theta_0) \leq E \forall t \in [1, T] \), then \( \theta_{t-1} = \theta_0 \), leading \( p^*_t(\theta_{t-1}) = p^*_t(\theta_0) = p^*_t \) according to Proposition 1.

We will do the proof separately for three possible cases of \( e^*_t \): 1) \( e^*_t \) never reaches upper or lower bound with all \( \mathbf{p}_t \) and \( \mathbf{\kappa}_t \) equal to zero; 2) \( e^*_t \) reached upper bound first; 3) \( e^*_t \) reached lower bound first. These three cases are illustrated in Fig. 1.

1) \( 0 < e^*_t < E \forall t \in [1, T] \): This covers the cases when \( e^*_t \) never reached the upper or lower bound, which from Proposition 1 we know \( \theta_0 = \theta_1 = \ldots = \theta_T \), hence \( p^*_t = p^*_t(\theta_0) \) and \( e^*_t = \sigma_t(\theta_0) \) for all \( t \in [1, T] \), and in particular for \( T = T \) we have

\[
\theta_0 = \theta_T = -c_T(e^*_T) = -c_T(\sigma_T(\theta_0))
\]

according to (20d) and the aforementioned result. Since \( O_t \) and \( c_T \) are convex, \( \sigma_T \), and \( \mathbf{p}_t \) (inverse of \( \sigma_T \)) are monotonic increasing functions, it follows

\[
\begin{align*}
x \geq \theta_0 && (19a) \\
p^*_t(x) &\leq p^*_t(\theta_0) : \text{convexity, see 1} && (19b) \\
\sigma_t(x) &\geq \sigma_t(\theta_0) : \text{see 9} && (19c) \\
c_T(\sigma_t(x)) &\geq c_T(\sigma_t(\theta_0)) : c_T(\cdot) \text{ both sides} && (19d) \\
c_T(\sigma_T(x)) &\geq \theta_0 \geq -\theta_0 && (19e) \\
c_T(\sigma_T(x)) &\geq -x : -\theta_0 \geq -x && (19f) \\
x &\geq c_T(\sigma_T(x)) && (19g)
\end{align*}
\]

meaning if \( x \geq -c_T(\sigma_T(\theta_0)) \) then \( x \geq \theta_0 \), and vice versa for \( \sigma_t(x) \) goes below 0, hence we proved condition 1-a and 2-a. It is also trivial to see that if \( x = c_T(\sigma_T(x)) \), then the KKT condition is satisfied and \( x = \theta_0 \), which proves condition 3. Thus all conditions in this theorem are proved for this case.

2) \( \exists \tau \in [1, T] \) s.t. \( \mathbf{p}_\tau > 0 \) and \( 0 < e^*_\tau < E \forall t \in [1, \tau] \): This covers the cases when \( e^*_\tau \) reached the upper bound first. An example of this case can be shown in Fig. 4. Now from (19d)

\[
\begin{align*}
\sigma_t(x) &> \sigma_t(\theta_0) \text{ hence } x > \theta_0 \text{, and vice versa for } \sigma_t(x) \text{ goes below 0, hence we proved condition 1-a and 2-a. It is also trivial to see that if } x = c_T(\sigma_T(x)) \text{, then the KKT condition is satisfied and } x = \theta_0, \text{ which proves condition 3. Thus all conditions in this theorem are proved for this case.}
\end{align*}
\]

we can conclude if \( \exists \tau \in [1, T] \) s.t. \( \mathbf{p}_\tau > 0 \) and \( 0 < e^*_\tau < E \forall t \in [1, \tau] \), then the same condition must be satisfied for all \( x \geq \theta_0 \), hence condition 1-a is proved. And from (17) we know after reaching the upper bound, all the following \( \theta \) values will be greater than \( \theta_0 \) until \( e^*_t \) reaches the lower bound or till the end of the operation \( T \) (i.e., \( e^*_t \) never reaches the lower bound). Without loss of generality, let \( \gamma \) be the time that \( e^*_\gamma \) first reaches the lower bound or the end of the operation period, i.e., \( e^*_t > 0 \forall t \in [1, \gamma] \), it follows

\[
\begin{align*}
x &\leq \theta_0 \quad (20a) \\
x &\leq \theta_t \forall t \in [1, \gamma] \quad (20b) \\
p^*_t(x) &\geq p^*_t(\theta_{t-1}) : \text{see 1} \quad (20c) \\
p^*_t(x) &\geq p^*_t : \text{Proposition 1} \quad (20d) \\
\sigma_t(x) &\leq e^*_t \quad (20e)
\end{align*}
\]

which means \( \sigma_t(x) \) either will go below 0 (condition 2-a) or \( \sigma_T(x) \leq e^*_T \) which leads to \( x \leq c_T(\sigma_T(x)) \) (condition 2-b) according to (19), hence we proved this theorem for this case.

3) \( \exists \tau \in [1, T] \) s.t. \( \mathbf{\kappa}_\tau > 0 \) and \( 0 < e^*_\tau < E \forall t \in [1, \tau] \): This covers the case when \( e^*_\tau \) reaches the lower bound first. This is a mirror proof to the previous case while inverting the upper and lower bound logic, hence this proof is omitted.