Iterative Optimization-based Control of a Class of Mixed Logical Dynamical Systems with STL Specifications: A Case Study on Microgrids

Yoshinari Takayama, Adnane Saoud, Alessio Iovine

Iterative Optimization-based Control of a Class of Mixed Logical Dynamical Systems with STL Specifications: A Case Study on Microgrids

Yoshinari Takayama * **Adnane Saoud** **,* **Alessio Iovine** *

* *Laboratory of Signals and Systems (L2S), CNRS, CentraleSupelec, Paris-Saclay University 3, rue Joliot Curie, 91190 Gif-sur-Yvette, France* {*name.surname*}*@centralesupelec.fr*
** *College of Computing, University Mohammed VI Polytechnic, Benguerir, Morocco adnane.saoud@um6p.ma*

Abstract: Motivated by the interest in modeling real-world systems as mixed logical dynamical (MLD) systems with signal temporal logic (STL) specifications, this paper introduces an efficient iterative optimization scheme for this class of systems. We focus on a microgrid system with energy losses, providing two formulations: a mixed-integer-based model and a model tailored to the proposed technique. A numerical case study using model predictive control (MPC) demonstrates that the proposed method significantly improves computation time compared to the classical mixed-integer linear programming (MILP) approach for longer time horizons, while also addressing discontinuity and volatility issues sometimes observed in the MILP approach.

Keywords: optimal control, signal temporal logic, mixed logical dynamical systems, microgrids

1. INTRODUCTION

Over the past decades, there has been a growing recognition of the efficacy of utilizing high-level languages in the design and verification of controllers. Among several high-level languages, Signal Temporal Logic (STL) stands out as it was introduced to facilitate the reasoning about temporal properties of continuous-time dynamical systems. It offers a useful quantitative semantics called *robustness*, which quantifies how robustly a formula is satisfied (Donzé and Maler (2010)). By maximizing this score, control input sequences can be synthesized robustly, and the resulting trajectories can formally satisfy the specification. Initially, Karaman et al. (2008) proposed formulating it as a mixed-integer program (MIP). However, this MILP formulation tends to be computationally expensive when the horizon increases, due to the exponential increase of the binary variables (see Karaman et al. (2008); Sadraddini and Belta (2019)). To avoid this issue, recent work has focused on formulating the problem as nonlinear programs (NLP) by a smooth approximation of max and min operators in the robustness function (e.g., Pant et al. (2017); Gilpin et al. (2021)). This NLP is then solved *naively* through a sequential quadratic programming (SQP) method (or other gradient-based methods). However, the major drawback in such approaches is that they can easily become infeasible and may not find the global optima due to the iterative approximations.

To tackle these issues, Takayama et al. (2023a,b) propose a novel iterative optimization framework that exploits STL structures utilizing a heuristic method called convex-concave procedure (CCP) (see Lipp and Boyd (2016)). This scheme has shown superior performance with respect to the aforementioned algorithms. However, it does not fully consider cases where the dynamics include logical components. With the aim of implementing these algorithms in practical contexts, this paper seeks to expand upon the concepts introduced in Takayama et al. (2023a,b) for such hybrid systems. We focus on a class of Mixed Logic Dynamical (MLD) systems that are of interest for modeling, among others, power systems. Therefore, we extend the basic algorithm to this class of MLD systems with STL specifications. As a first step, we consider the power management of an electrical microgrid that takes into account *power losses* and consequently necessitates logical constraints (see Iovine et al. (2019)).

The proposed framework first transforms these logical parts of the MLD systems into certain inequalities with min functions. These min-type constraints have nice structures that can be leveraged with the simultaneous use of the STL decomposition algorithm in Takayama et al. (2023a,b). Then, the proposed framework only approximates the disjunctive part of the system and temporal specifications, which is different from the simple application of sequential quadratic programming (SQP). To the best of the authors' knowledge, no iterative algorithms exploiting problem structures have been proposed particularly for MLD systems with STL specifications. The main contribution of this paper is the aforementioned framework. A secondary, albeit minor, contribution concerns the novel transformation of the microgrid MLD model into the suggested preferable form and the solution of the power management problem with an STL specification.

A comparison of the considered problem between the suggested methodology and the classical MILP one is performed in numerical simulations via a Model Predictive Control (MPC) approach. It demonstrates the better performance of the proposed methodology with the increase of the prediction horizon while addressing discontinuity and volatility issues, sometimes observed in the MILP approach.

⋆ This work has received funding from the Agence Nationale de la Recherche (ANR) via grant ESTHER ANR-22-CE05-0016. Y. Takayama acknowledges the support of the Watanabe Foundation International Fellowship.

The rest of this paper is organized as follows. Section 2 introduces preliminaries, while Section 3 models the power management of a direct current microgrid as a linear MLD system and formulates the problem as a MILP. Section 4.1 demonstrates how the problem can be recast as a Difference of Convex (DC) program, enabling the proposed approach, which is described in Section 4.2. Section 5 then compares the performance of the proposed methodology against the MILP-based framework commonly used in the existing literature.

2. PRELIMINARIES

Notations Let $\mathbb{R}$, $\mathbb{N}$, and $\mathbb{N}_{>0}$ be the set of real numbers, integers, and positive integers, respectively. Let $\mathbb{P}$ denote the probability measure. For $x \in \mathbb{R}$, $|x|$ denotes its absolute value. The superscript $\top$ indicates transpose. A signal is defined as an infinite sequence $\boldsymbol{x} = x_0 x_1 \cdots$, where $x_t \in \mathbb{R}, t \in \mathbb{N}$. Signals starting at timestep t, i.e., $x_t x_{t+1} \cdots$ are denoted as $(\boldsymbol{x}, t)$. For $x_i \in \mathbb{R}, i \in \{1, 2, ..., n\}$, $Diag(x_1, x_2, ..., x_n)$ denote the diagonal matrix of order n where the elements on the main diagonal are given by $x_1, x_2, ..., x_n$.

2.1 Mixed Logical Dynamical (MLD) Systems

MLD systems are a class of systems involving both continuous and boolean-valued states, constraints, and logic statements (Bemporad and Morari (1999)):

$$\begin{cases} x_{t+1} = A_1 x_t + A_2 d_t + B_1 u_t + B_2 \delta_t + B_3 z_t, & \text{(1a)} \\ E_2 \delta_t + E_3 z_t \leq E_1 u_t + E_4 x_t + E_5 d_t + E_6, & \text{(1b)} \end{cases}$$

where $x_t = \left[x_{t,c}^\mathsf{T}, x_{t,l}^\mathsf{T} \right]^\mathsf{T}, x_{t,c} \in \mathcal{X} \subseteq \mathbb{R}^n$ is the state, $u_t = \left[u_{t,c}^\mathsf{T} \; u_{t,l}^\mathsf{T} \right]^\mathsf{T}, u_{t,c} \in \mathcal{U} \subseteq \mathbb{R}^m$ is the control input, and $d_t \in \mathbb{R}^l$ is the exogenous input. $\delta_t \in \{0,1\}^{r_l}$ is a vector of discrete auxiliary variables, $z \in \mathbb{R}^{r_c}$ is a vector of continuous auxiliary variables, and $A_1, A_2, B_1, B_2, B_3, E_1, E_2, E_3, E_4, E_5$, and E_6 are the system matrices. The subindex c denotes the continuous-valued components while the subindex l denotes the discrete-valued ones. Given an initial state x_0, a horizon $\kappa \in \mathbb{N}_{>0}$ and a sequence of control inputs $\boldsymbol{u} = (u_0, \ldots, u_{\kappa-1})$ and exogenous input $\boldsymbol{d} = (d_0, \ldots, d_{\kappa-1})$, the trajectory $\boldsymbol{x} = (x_0, \ldots, x_\kappa)$ is uniquely generated.

2.2 Signal Temporal Logic (STL)

STL is a predicate logic that specifies continuous signal properties (see Fainekos and Pappas (2009)). Predicates, or atomic propositions, are a part of STL and take either True (1 or $\top$) or False (0 or $\bot$). A predicate μ can be acquired through function $g^\mu : \mathbb{R}^n \to \mathbb{R}$ as $\mu = (g^\mu(x_t) \leq 0)$. In addition to the standard boolean operators $\wedge$ and $\vee$, the STL also incorporates temporal operators $\Box$ (*always*), $\Diamond$ (*eventually*), and $\boldsymbol{U}$ (*until*). The semantics of STL is defined as follows (Baier and Katoen (2008)):

$$\varphi := \mu | \vee \varphi | \wedge \varphi | \Box_{[t_1,t_2]} \varphi | \Diamond_{[t_1,t_2]} \varphi \mid \varphi_1 \boldsymbol{U}_{[t_1,t_2]} \varphi_2. \quad (2)$$

The symbol | stands for OR and the definition is recursive. Each temporal operator has associated bounded time interval $[t_1, t_2]$ where $0 \leq t_1 < t_2$ and $t_2 < \infty$. The temporal operator always $\Box_{[t_1,t_2]}\varphi$ is satisfied at time t if φ is True at all times in $t + [t_1, t_2]$ while eventually $\Diamond_{[t_1,t_2]}\varphi$ is satisfied at time t if φ is True at some time in $t + [t_1, t_2]$. As an instance, formula $\varphi = \Box_{[0,2]}(x_t > 0)$ evaluated at time 0 specifies that for all times between 0 and 2, $x_t > 0$ is satisfied, and the formula $\varphi = \Diamond_{[0,2]}(x_t > 0)$ evaluated at time 0 specifies that there exists a time instant t between 0 and 2 such that $x_t > 0$.

The concept of *robustness* is a significant semantics defined for STL formulas, which is a real-valued function that characterizes how well a trajectory satisfies an STL formula. The robustness of an STL formula φ to a trajectory $\boldsymbol{x}$ and a time t can be obtained recursively according to the following quantitative semantics:

$$\begin{aligned}
&\rho^\mu((\boldsymbol{x}, t)) = -g^\mu(x_t) \\
&\rho^{\varphi_1 \wedge \varphi_2}((\boldsymbol{x}, t)) = \min(\rho^{\varphi_1}((\boldsymbol{x}, t)), \rho^{\varphi_2}((\boldsymbol{x}, t))) \\
&\rho^{\varphi_1 \vee \varphi_2}((\boldsymbol{x}, t)) = \max(\rho^{\varphi_1}((\boldsymbol{x}, t)), \rho^{\varphi_2}((\boldsymbol{x}, t))) \\
&\rho^{\Box_{[t_1,t_2]}\varphi}((\boldsymbol{x}, t)) = \min_{t' \in [t+t_1, t+t_2]} (\rho^\varphi((\boldsymbol{x}, t'))) \\
&\rho^{\Diamond_{[t_1,t_2]}\varphi}((\boldsymbol{x}, t)) = \max_{t' \in [t+t_1, t+t_2]} (\rho^\varphi((\boldsymbol{x}, t'))) \\
&\rho^{\varphi_1 \boldsymbol{U}_{[t_1,t_2]} \varphi_2}((\boldsymbol{x}, t)) = \min_{t' \in [t+t_1, t+t_2]} \Bigg(\max\Big(\Big[\rho^{\varphi_1}((\boldsymbol{x}, t')), \\
&\max_{t'' \in [t+t_1, t']} \left(\rho^{\varphi_2}((\boldsymbol{x}, t'')) \right) \Big] \Big) \Bigg)
\end{aligned} \quad (3)$$

Trajectory $\boldsymbol{x}$ satisfies formula φ, denoted as $\boldsymbol{x} \vDash \varphi$, if and only if $\rho^\varphi((\boldsymbol{x}, t)) \geq 0$. By maximizing the robustness score ρ^ϕ that results from (3), a robust control input sequence can be synthesized. The trajectory length N has to be chosen so that it is longer than *the formula length* of φ, which is the horizon needed to calculate the robustness of a formula (see Sadraddini and Belta (2015)).

3. PROBLEM FORMULATION

3.1 Microgrid Model

In this paper, we consider the microgrid model in Iovine et al. (2019) to demonstrate our main idea. The considered microgrid consists of a battery, a supercapacitor, a renewable generator from the photovoltaic panel (PV), a load, and a direct current bus connecting the aforementioned components, as shown in Fig. 1. The dynamical model of the microgrid given by

$$\begin{cases}
E_{DC}(t+1) = E_{DC}(t) \\
\quad + \gamma \left[\eta_{PV}(O_{PV}(t) - P_{PV}(t)) - \dfrac{1}{\eta_L}(O_L(t) - P_L(t)) \right] \\
\quad + \gamma \left[\eta_B^d P_B^+(t) - \dfrac{1}{\eta_B^c} P_B^-(t) + \eta_S^d P_S^+(t) - \dfrac{1}{\eta_S^c} P_S^-(t) \right] \\
E_B(t+1) = E_B(t) + \gamma \left[-P_B^+(t) + P_B^-(t) \right] \\
E_S(t+1) = (1 - \gamma \alpha_S) E_S(t) + \gamma \left[-P_S^+(t) + P_S^-(t) \right] + w(t)
\end{cases} \quad (4)$$

where $E_B, E_S, E_{DC}(t) \in \mathbb{R}$ are the energies stored in the battery, supercapacitor, and microgrid, respectively. $O_{PV} - P_{PV}$ is the power produced by the PV array, where $O_{PV}(t) \in \mathbb{R}$ is the current available power and $P_{PV}(t) \in \mathbb{R}$ is the amount of power to be cut off for the stability of the entire grid; according to the same reasoning, $O_L - P_L$ is the power demanded by the load, with $O_L \in \mathbb{R}$ the current demanded power and $P_L \in \mathbb{R}$ the amount of power to be cut off. The terms $P_B^+, P_B^-, P_S^+, P_S^- \in \mathbb{R}$ are the powers exchanged by the battery and the supercapacitor, respectively, where the power absorbed by the storages are P_B^- and P_S^-, while the ones provided are P_B^+ and P_S^+. We introduced these different variables for charging and discharging to take into account the losses due to the physical characteristics, which could result in different values in the charge or discharge case. The parameters

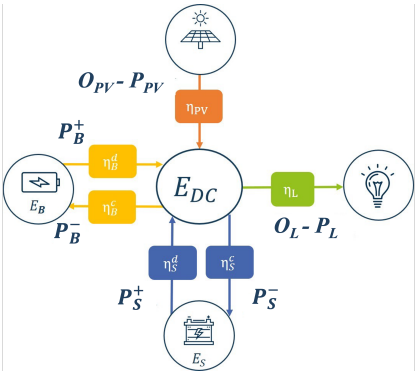

Fig. 1. The considered microgrid framework

η_{PV}, $\frac{1}{\eta_L}$, η_B^d, $\frac{1}{\eta_B^c}$, η_S^d, $\frac{1}{\eta_S^c}$ $\in \mathbb{R}$ describe the loss in efficiency due to $O_{PV} - P_{PV}$, $O_L - P_L$, P_B^+, P_B^-, P_S^+, P_S^-, respectively. The parameter $\alpha_S \in \mathbb{R}$ is the self-discharge ratio of the supercapacitor with respect to the considered sampling time $\gamma \in \mathbb{R}$.

Differently from Iovine et al. (2019), a Gaussian disturbance $w(t)$ is considered in (4). It is a white noise with the variance being 0.01, i.e., $w(t) \sim \mathcal{N}(0, 0.01)$. By focusing exclusively on the secondary control layer of the hierarchical power system, we introduce a stochastic disturbance to accurately reflect the discrepancies between this layer and the unconsidered primary layer. As highlighted in Iovine et al. (2019), the supercapacitor is particularly sensitive to these mismatches, striving to address rapid disturbances from lower levels that the secondary layer does not fully capture. This leads to variations between the supercapacitor's target value and its actual state in each iteration. To address this, this study introduces a white noise specifically tailored for the supercapacitor.

Next, we define the constraints on the dynamics. For state constraints of the microgrid and battery, we impose each energy to be kept between an interval to ensure a certain level of power quality and not damage the devices as follows:

$$E_B^m \leq E_B(t) \leq E_B^M, E_{DC}^m \leq E_{DC}(t) \leq E_{DC}^M, \forall t \in \mathbb{N} \quad (5)$$

where the constants $E_B^m, E_B^M, E_{DC}^m, E_{DC}^M \geq 0$. For the supercapacitor's state, however, a similar hard constraint $E_S^m \leq E_S(t) \leq E_S^M$ can make the optimization problem infeasible due to the disturbance $w(t) \in \mathcal{N}(0, 0.01)$ in the simulated model. To avoid this issue, the classical approach is to introduce the chance constraints or the robust tubes by stochastic/robust optimizations. However, we avoid these approaches by introducing a temporal specification in the next section.

For control inputs, we introduce the following constraints.

- The inputs of the PV and the load must not exceed the provided energies themselves;

$$P_{PV}(t) \geq 0, O_{PV}(t) - P_{PV}(t) \geq 0, \forall t \in \mathbb{N}, \quad (6)$$

$$P_L(t) \geq 0, O_L(t) - P_L(t) \geq 0, \forall t \in \mathbb{N}. \quad (7)$$

- The charging power and the discharging power by the battery and the supercapacitor are bounded;

$$0 \leq P_B^+(t) \leq \overline{P}_B^+, 0 \leq P_B^-(t) \leq \overline{P}_B^-, \forall t \in \mathbb{N}, \quad (8)$$

$$0 \leq P_S^+(t) \leq \overline{P}_S^+, 0 \leq P_S^-(t) \leq \overline{P}_S^-, \forall t \in \mathbb{N}, \quad (9)$$

where $\overline{P}_B^+ \geq 0$, $\overline{P}_B^- \geq 0$, $\overline{P}_S^+ \geq 0$, $\overline{P}_S^- \geq 0$.

- The battery power variation should be restricted to save the battery life;

$$|P_B^+(t+1) - P_B^+(t)| \leq \Delta\overline{P}_B^+, \forall t \in \mathbb{N} \quad (10)$$

$$|P_B^-(t+1) - P_B^-(t)| \leq \Delta\overline{P}_B^-, \forall t \in \mathbb{N} \quad (11)$$

where $\Delta\overline{P}_B^+ \geq 0$, $\Delta\overline{P}_B^- \geq 0$;

- To avoid the simultaneous charging and discharging of the battery and the supercapacitor, either $P_B^+(t) = 0$ or $P_B^-(t) = 0$ (resp. $P_S^+(t) = 0$ or $P_S^-(t) = 0$) must be satisfied for all $t \in \mathbb{N}$.

$$\text{if } P_B^+(t) \neq 0 \text{ then } P_B^-(t) = 0 \quad (12)$$

$$\text{if } P_S^+(t) \neq 0 \text{ then } P_S^-(t) = 0 \quad (13)$$

The last condition includes logical components. We introduce two formulations for these constraints in Subsections 3.2 and 4.1.

3.2 MLD Formulation

The classical formulation of (12) and (13) is the MLD system using binary variables. We introduce binary variables $S_B, S_S \in \{0, 1\}$ and constraints as

$$P_B^+(t) \leq S_B(t) \cdot \overline{P}_B^+, \; P_B^-(t) \leq (1 - S_B(t)) \cdot \overline{P}_B^-, \quad (14)$$

$$P_S^+(t) \leq S_S(t) \cdot \overline{P}_S^+, \; P_S^-(t) \leq (1 - S_S(t)) \cdot \overline{P}_S^-, \quad (15)$$

Adding the constraints (14) and (15) directly into the optimization problem results in a mixed integer nonlinear program, which is extremely difficult to solve. Therefore, the nonlinearity of these constraints (14), (15) are transformed into a linear constraint with binary variables in the literature. Due to page constraints, we refer readers to Bemporad and Morari (1999); Pham et al. (2022) for full details. Let us denote the state, input, and exogenous input as

$$x_t = [E_{DC}(t) \;\; E_B(t) \;\; E_S(t)]^\mathsf{T} \quad (16)$$

$$u_t = [P_{PV}(t) \;\; P_L(t) \;\; P_B^+(t) \;\; P_B^-(t) \;\; P_S^+(t) \;\; P_S^-(t) \;\;]^\mathsf{T} \quad (17)$$

$$d_t = [O_{PV}(t) \;\; O_L(t)]^\mathsf{T} \quad (18)$$

The resulting system is an MLD system of the form (1), which is denoted as $\mathcal{S}_{\text{MLD}}$ below.

3.3 Problem Statement

Given the hybrid system $\mathcal{S}_{\text{MLD}}$, an estimation $\widetilde{\boldsymbol{d}}$ of exogenous inputs $\boldsymbol{d} = (d_0, \ldots, d_{N-1})$, and constraint sets $\mathcal{X}$ and $\mathcal{U}$ defined in (5)–(11), this paper considers the control problem with STL specifications.

Problem 1. Given an initial state $x_0 \in \mathcal{X}$, an STL specification φ, and a prediction horizon $N \in \mathbb{N}_{>0}$, compute the control inputs $\boldsymbol{u} = (u_0, \ldots, u_{N-1}) \in \mathcal{U}$ and the resulting trajectories $\boldsymbol{x} = (x_0, \ldots, x_N)$ that satisfies φ robustly by solving the following optimization problem:

$$\min_{\boldsymbol{u}} \; -\rho^\varphi(\boldsymbol{x}) \quad (19a)$$

$$\text{s.t. system } \mathcal{S}_{\text{MLD}} \quad (19b)$$

$$x_t \in \mathcal{X}, u_t \in \mathcal{U} \quad (19c)$$

$$\rho^\varphi(\boldsymbol{x}) \geq 0 \quad (19d)$$

The STL specification part of the problem, i.e., (19a) and (19d) can be encoded as mixed integer constraints, which results in a mixed-integer linear program (MILP). This MILP approaches tend to be computationally expensive when the horizon increases. Therefore, we provide an alternative approach in the next section.

4. PROPOSED FRAMEWORK

The proposed algorithm extends an STL encoding framework in Takayama et al. (2023a,b) to a class of hybrid systems by combining it with a specific system reformulation. In Takayama et al. (2023a,b), an STL encoding framework was proposed, which decomposes the part of STL specification (19a) and (19d) into a set of constraints. This algorithm can be directly applied to our problem (19) as well, as the formulation of the STL specification part is the same. The main difference between them is the approximation of the system parts at each iteration. The algorithm in Takayama et al. (2023a,b) only approximates the concave parts of the STL specification, while this paper proposes to approximate also the concave parts of the system in the same manner.

4.1 Proposed Formulation

The proposed approach transforms the system $\mathcal{S}_{\text{MLD}}$ into a particular form. In particular, constraints in (12) and (13) are formulated as constraints with min functions *without* introducing binary variables:

$$\min(P_B^+(t), P_B^-(t)) \leq 0, \tag{20}$$
$$\min(P_S^+(t), P_S^-(t)) \leq 0. \tag{21}$$

Importantly, although these constraints themselves are *not equivalent* to the inequalities of (14) and (15) in general, we demonstrate their equivalence under certain additional conditions in the following result:

Proposition 1. Constraints (8) and (14) (respectively (9) and (15)) are equivalent to the inequalities in (8) and (20) (respectively, (9) and (21)).

Proof. We only provide a proof for the first case, the second case can be derived similarly. From (8), one has that $P_B^+(t)$ and $P_B^-(t)$ are positive for all $t \in \mathbb{N}$. Therefore, (20) is equivalent to

$$\min(P_B^+(t), P_B^-(t)) = 0, \quad \forall t \in \mathbb{N}.$$

These equality constraints state that for all $t \in \mathbb{N}$, either $P_B^+(t) = 0$ or $P_B^-(t) = 0$, which is the same statement as in (14). This concludes the proof.

By this interpretation of the energy loss constraints, the system model (4) is transformed into

$$\begin{cases} x_{t+1} = A_1 x_t + B_1 u_t + C_1 d_t, & \text{(22a)} \\ \min(D_1 x_t + G_1 u_t + F_1 d_t, ..., D_h x_t + G_h u_t + F_h d_t) \\ \qquad \leq E_1 u_t + E_4 x_t + E_5 d_t + E_6. & \text{(22b)} \end{cases}$$

The matrices $A_1, D_j, E_4 \in \mathbb{R}^{n\times n}, B_1, G_j, E_1 \in \mathbb{R}^{n\times m}$, $C_1, E_5, F_j \in \mathbb{R}^{n\times l}$, $j \in \{0, ..., h\}$, represent the dynamics of the system, while $E_6 \in \mathbb{R}^n$ is a constant vector.

4.2 STL Decomposition

In this subsection, we briefly describe the basic idea of the reformulation. Due to page limitation, we changed some notations from Takayama et al. (2023b) and have omitted or simplified many explanations. We refer to the paper for further details.

The first procedure is to decompose the robustness function $-\rho^{\varphi}$ in the cost function (19a) into a set of constraints. For simplicity, we consider the case where the outermost operator of the robustness function ρ^{φ} is min, i.e., $\rho^{\varphi} = \min(\rho^{\varphi_1}, \rho^{\varphi_2}, ..., \rho^{\varphi_r})$ (see the definition of STL robustness in (3)). We introduce a new variable s_ξ, and reformulate the program as follows.

$$\min_{\boldsymbol{x},\boldsymbol{u},s_\xi} \ s_\xi \tag{23a}$$
$$\text{s.t. system (22)} \tag{23b}$$
$$x_t \in \mathcal{X}, u_t \in \mathcal{U} \tag{23c}$$
$$s_\xi \leq 0 \tag{23d}$$
$$-\rho^{\varphi_1}(\boldsymbol{x}) \leq s_\xi, \ldots, -\rho^{\varphi_r}(\boldsymbol{x}) \leq s_\xi \tag{23e}$$

As each $-\rho^{\varphi_i}$ for $i = 1, \ldots, r$ in (23e) is a robustness function, each inequality can be restated as a constraint in one of the following two forms, depending on whether the outermost operator is max or min:

$$\max(-\rho^{\Phi_1}, ..., -\rho^{\Phi_{y_{\max}}}) \leq s_\xi, \tag{24}$$
$$\min(-\rho^{\Psi_1}, ..., -\rho^{\Psi_{y_{\min}}}) \leq s_\xi, \tag{25}$$

where functions $-\rho^{\Phi_j}(j \in \{1, ..., y_{\max}\})$ (resp. $-\rho^{\Psi_j}(j \in \{1, ..., y_{\min}\})$) are robustness functions associated with Φ_j (resp. Ψ_j), which are the subformulas of $\varphi_i(i \in \{1, ..., r\})$. We continue decomposing these max and min functions until all the arguments in each function become the predicates, which is the bottom of the STL specification. When these repetitive operations finish, the program becomes a Difference of Convex (DC) program, and can be written as

$$\min_{\boldsymbol{z}} \ s_\xi \tag{26a}$$
$$\text{s.t. system (22a)} \tag{26b}$$
$$x_t \in \mathcal{X}, u_t \in \mathcal{U} \tag{26c}$$
$$s_\xi \leq 0, \tag{26d}$$
$$\boldsymbol{\rho}^{\mu_{\max}} \leq \boldsymbol{s}_{\max}, \tag{26e}$$
$$\left.\begin{array}{l} \text{system (22b)} \\ \min(-\rho^{\mu_1^{(1)}}, ..., -\rho^{\mu_{y_{\min}}^{(1)}}) \leq s_{\min}^{(1)} \\ \quad \vdots \\ \min(-\rho^{\mu_1^{(w)}}, ..., -\rho^{\mu_{y_{\min}}^{(w)}}) \leq s_{\min}^{(w)} \end{array}\right\} \text{ concave parts} \tag{26f}$$

where $\boldsymbol{z} = [\boldsymbol{x}^\mathsf{T}, \boldsymbol{u}^\mathsf{T}, s_\xi, \boldsymbol{s}_{\max}, s_{\min}^{(1)}, ..., s_{\min}^{(w)}]^\mathsf{T} \in \mathbb{R}^{n+m+1+v+w}$ denotes the vector of optimization variables. $\boldsymbol{s}_{\max} = [s_{\max}^{(1)}, ..., s_{\max}^{(v)}]^\mathsf{T}$ denotes all the introduced variables during the decomposition procedures of constraint-type (24), in addition to variable s_ξ. Similarly, $[s_{\min}^{(1)}, ..., s_{\min}^{(v)}]^\mathsf{T}$ denotes all the introduced variables during the decomposition procedures of constraint-type (25). $\boldsymbol{\rho}^{\mu_{\max}} = [\rho^{\mu_{\max}^{(1)}}, ..., \rho^{\mu_{\max}^{(v)}}]^\mathsf{T}$ denotes all the predicates whose parent node is a conjunctive-type node. Similarly, $\boldsymbol{\rho}^{\mu_{\min}} = [\rho^{\mu_1^{(1)}}, ..., \rho^{\mu_{y_{\min}}^{(1)}}, \rho^{\mu_1^{(w)}}, ..., \rho^{\mu_{y_{\min}}^{(w)}}]^\mathsf{T}$ denotes all the predicates whose parent node is a disjunctive-type node. Note that some slight abuse of notations are made such that some of the variables $s_{\max}^{(\cdot)}, s_{\min}^{(\cdot)}$ in constraints (26e),(26f) can also be variables s_ξ, and some of the arguments of the min functions (26f) are variables $s_{\max}^{(\cdot)}$. Despite the several *non-equivalent* transformations, this transformation is proven to retain important properties of the original program. Specifically, building upon the findings of Theorem 1 in Takayama et al. (2023a), one can easily show that the optimal solutions of both programs (19) and (26) are identical.

The resulting problem (26) has a particular structure: First, (26) belongs to a class of programs called Difference of Convex (DC) Programs (see Shen et al. (2016)), where the program only consists of convex parts and concave

parts. Moreover, the concave parts are all in the form of min functions in (26f). These structures enable us to efficiently solve the problem.

4.3 Smooth Approximations

Next, we provide a smooth approximation for all the min functions in (26f) using the log-sum-exp (LSE) function, as $\overline{\min}_k := -\overline{\max}_k(-\boldsymbol{a})$, where $\overline{\max}_k(-\boldsymbol{a}) := \frac{1}{k}\ln\sum_{i=1}^{r} e^{ka_i}$ (e.g., Pant et al. (2017)), and $k \in \mathbb{R}$ is the smoothing parameter. Note that the proposed approach uniquely focuses on smoothing only minimum functions due to the aforementioned structure. This differs from the literature where typically both max and min functions in the robustness function (3) are smoothed.

4.4 Iterative Optimization

Finally, we solve the smoothed version of (26) by sequentially solving a quadratic program: At each iteration, we linearize the concave parts, i.e., the $\overline{\min}$ functions in (26f), which corresponds to the disjunctive node of the robustness tree of the specification φ. Then, we solve the resulting efficient quadratic program, given the solution of the previous iteration as the initial guess. We operate this procedure sequentially until it converges to an optimum. The following proposition is a direct consequence of the propositions in Takayama et al. (2023b). Note that sets $\mathcal{X}$ and $\mathcal{U}$ defined in (5)–(11) are restricted to intersections of polyhedra.

Proposition 2. Let all predicates g^μ of the STL specifications φ be restricted to linear functions. Then, the proposed algorithm solves a (convex) *quadratic* program at each iteration, while it only approximates logical parts (26f) of Problem 1 by their linearizations, that is, the disjunctive parts of specification φ and the logical parts of system (22b).

This proposition demonstrates the benefits of our iterative optimization approach. Firstly, it eliminates the need for binary variables, simplifying the problem structure. Secondly, it preserves complete information of the problem's convex components. Additionally, all concave parts share the same $\overline{\min}$ form, which can enhance the algorithm's efficiency.

5. NUMERICAL EXPERIMENT

We demonstrate the effectiveness of the proposed method through a numerical experiment in a MPC scenario with prediction horizon N. We compare the proposed method described in Section 4 with the MILP formulation mentioned in Subsection 3.2. All experiments were conducted on a MacBook Air 2020 with an Apple M1 processor (Maximum CPU clock rate: 3.2 GHz) and 8GB of RAM. All parameter settings for the state and control constraints are the same as the ones in Iovine et al. (2019) with the sampling time γ being one second. We set $\tilde{E}_S^m = 3.0$ and $\tilde{E}_S^M = 6.0$ for STL specification in (28) inside the range of $E_S^m = 1.5$ to $E_S^m = 8.0$, and the estimated exogenous inputs $\widetilde{\boldsymbol{d}} = (\widetilde{d}_0, \ldots, \widetilde{d}_{N-1}) = ([O_{PV}(0) \quad O_L(0)]^\mathsf{T}, \ldots, ([O_{PV}(0)\ O_L(N-1)]^\mathsf{T})$, assuming that $O_{PV}(t)$ is constant throughout the horizon while O_L is known.

Both programs are solved using the QP-solver and the mixed integer solver of GUROBI in Python with the default options. For the proposed approach, the parameters of the STL encoding algorithm, `STLCCP` in Takayama et al. (2023b) are set as follows: the weight on penalty variables in the cost function at the outset to 5.0 (denoted as p_τ), the rate at which p_τ increases to 5.0, maximum values on variables for the terminal condition to $1e^2$. Other parameter variables for the algorithm are set as the defaults. For more details on these parameters and their default values, please refer to `https://github.com/yotakayama/STLCCP`. For the compared MILP, STL specification φ is transformed with the method in Belta and Sadraddini (2019) with the default setting of the parameter values.

For the STL specification φ, we consider maintaining a specific energy level in a microgrid component (the supercapacitor) against the added disturbance while avoiding too much conservativeness. We require $E_S(t)$ to come back to the safe range between $\tilde{E}_S^m$ to $\tilde{E}_S^M$ for τ seconds in the horizon N, where the unsafe regions is both 1.5 to 3.0 and 6.0 to 8.0. The specification of interest, φ, is defined as follows using predicates ϕ_1 and ϕ_2 as

$$\phi_1 = [E_S(t) - \tilde{E}_S^M \leq 0], \quad \phi_2 = [\tilde{E}_S^m - E_S(t) \leq 0], \tag{27}$$

$$\phi = [\phi_1 \wedge \phi_2], \quad \varphi = [\Box_{[-\tau, N-\tau]} \Diamond_{[0,\tau]} \phi]. \tag{28}$$

During each iteration of the receding horizon computation, we explore a finite trajectory with a horizon length of N, a part of which is dedicated to assessing the satisfaction of the *past* trajectory. The reason for assessing the past trajectory is to satisfy the formula in the actual trajectory, not only in the predicted trajectory.

In addition to the STL specifications, we also incorporate state tracking as our objective. We add a standard quadratic cost $\frac{1}{2}\left[\tilde{x}_N^\mathsf{T} P \tilde{x}_N + \sum_{i=0}^{N-1} \tilde{x}_i^\mathsf{T} Q \tilde{x}_i + u_i^\mathsf{T} R u_i)\right]$, where $\tilde{x}_t = x_t - x_t^{\mathrm{ref}}$. We require the state x_t to track the reference $x_t^{\mathrm{ref}} = [1.39e^{-5}, 310, 4.5]^\mathsf{T}$, while also requiring each element of the control inputs to track 0. The weight matrix is determined as $Q = Diag(e^{17}, e^8, e^9)$, $R = Diag(e^5, e^7, 5e^2, 5e^2, 1, 1)$, so that we put more importance on P_{PV} and P_L than P_B and P_S. This is because we prefer to charge or discharge the power to curtail the supply and demand in practice

We first compared the two methods with a fixed prediction horizon $N = 15$, with the STL specification parameter being $\tau = 5$. The results are presented in Fig. 2. The left-hand sides of all figures represent the results of the MILP approach, while the right-hand side represents that of the proposed approach. We can see in each comparison that the trajectories of both approaches are similar in all states and control variables. It is also worth noting that the MILP approach sometimes exhibits rapid variations in the supercapacitor's power depending on the parameters E_B^{ref} and $\overline{E}_B$ during the experiments. In contrast, the proposed approach successfully avoids this behavior. One possible reason for this behavior could be that the MILP formulation does not satisfy the complementary condition, meaning that the optimal solution from the previous iteration is not always a good solution for the current iteration due to the influence of added white noise. The proposed approach successfully avoids this behavior as it uses no boolean variables.

Next, we compare the computational time of the two approaches with fixed $\tau = 5$ over different horizons from $N = 15$ to 50. The result is summarized in Table 1. All the units are seconds. The computational time is divided into two parts: one is the required time for creating the program, which is called "Formulation", and the other is the required time for solving the program, which is called "Solve". The values in the table represent the average values up to the first 20 MPC simulation time, and the

values in parentheses represent the median values. We add the median value in the parenthesis because the computational time of the MIP method is greatly volatile depending on the initial values of variables. The MIP method took 1122 seconds at horizon $N = 35$ and exceeded 5000 seconds at horizons $N \geq 40$ to solve the initial simulation program. Therefore, we did not proceed with subsequent simulations for these horizons.

While results are comparable to MIP mthod with short horizons, the proposed method is more efficient than the MIP method for longer horizons. Moreover, the MIP method exhibits higher volatility, meaning values can vary significantly across different simulation times. This volatility is evident in the difference between the average and median values for the MIP method. This is in contrast with the proposed method, which is more consistent.

Table 1. Comparison of computational time

Horizon [N]	Formulation [s]		Solve [s]	
	MIP	Proposed	MIP	Proposed
15	0.10	0.08	0.17 (0.08)	2.23 (2.26)
20	0.14	0.10	1.06 (0.41)	3.38 (3.30)
25	0.14	0.12	16.85 (1.23)	4.00 (4.00)
30	0.23	0.14	329.40 (20.57)	4.84 (4.70)
35	0.24	0.16	>1000	5.55 (5.50)
40	0.25	0.17	>5000	6.32 (6.25)
50	0.30	0.29	>5000	9.22 (9.20)

6. CONCLUSIONS

This paper proposes an efficient optimization scheme dedicated to a class of MLD systems with STL specifications, avoiding the utilization of binary variables. The power management of an electrical microgrid with complex specifications is considered as a case study to show the effectiveness of the proposed approach. One future direction is to generalize the proposed framework and construct an efficient framework applicable to a wider class of hybrid systems, not limited to the MLD system that can be transformed into the form (22).

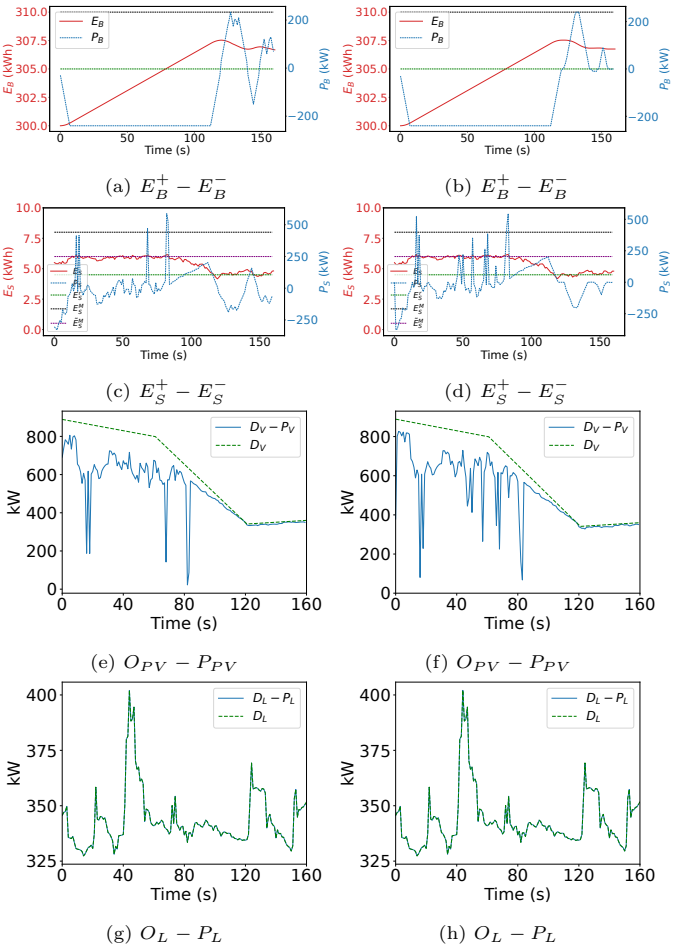

Fig. 2. Comparison of the resulted trajectory with $N = 15$ (lefts: MILP, rights: Proposed)

REFERENCES

Baier, C. and Katoen, J.P. (2008). *Principles of Model Checking*. MIT Press.

Belta, C. and Sadraddini, S. (2019). Formal methods for control synthesis: An optimization perspective. *Annual Review of Control, Robotics, and Autonomous Systems*, 2, 115–140.

Bemporad, A. and Morari, M. (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3), 407–427.

Donzé, A. and Maler, O. (2010). Robust satisfaction of temporal logic over Real-Valued signals. *in Formal Modeling and Analysis of Timed Systems*, 92–106.

Fainekos, G.E. and Pappas, G.J. (2009). Robustness of temporal logic specifications for continuous-time signals. *Theoretical Computer Science*, 410(42), 4262–4291.

Gilpin, Y., Kurtz, V., and Lin, H. (2021). A smooth robustness measure of signal temporal logic for symbolic control. *IEEE Control Systems Letters*, 5(1), 241–246.

Iovine, A., Rigaut, T., Damm, G., De Santis, E., and Di Benedetto, M.D. (2019). Power management for a DC microgrid integrating renewables and storages. *Control Engineering Practice*, 85, 59–79.

Karaman, S., Sanfelice, R.G., and Frazzoli, E. (2008). Optimal control of mixed logical dynamical systems with linear temporal logic specifications. *in IEEE Conference on Decision and Control*, 2117–2122.

Lipp, T. and Boyd, S. (2016). Variations and extension of the convex–concave procedure. *Optimization and Engineering*, 17(2), 263–287.

Pant, Y.V., Abbas, H., and Mangharam, R. (2017). Smooth operator: Control using the smooth robustness of temporal logic. *in IEEE Conference on Control Technology and Applications (CCTA)*, 1235–1240.

Pham, T.H., Iovine, A., Olaru, S., Maeght, J., Panciatici, P., and Ruiz, M. (2022). Nonlinearity handling in MPC for Power Congestion management in sub-transmission areas. In *18th IFAC Workshop on Control Applications of Optimization (CAO)*.

Sadraddini, S. and Belta, C. (2015). Robust temporal logic model predictive control. In *53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 772–779.

Sadraddini, S. and Belta, C. (2019). Formal synthesis of control strategies for positive monotone systems. *IEEE Transactions on Automatic Control*, 64(2), 480–495.

Shen, X., Diamond, S., Gu, Y., and Boyd, S. (2016). Disciplined convex-concave programming. *in IEEE Conference on Decision and Control (CDC)*, 1009–1014.

Takayama, Y., Hashimoto, K., and Ohtsuka, T. (2023a). Signal temporal logic meets convex-concave programming: A structure-exploiting SQP algorithm for STL specifications. *in IEEE Conference on Decision and Control (CDC)*.

Takayama, Y., Hashimoto, K., and Ohtsuka, T. (2023b). STLCCP: An efficient convex optimization-based framework for signal temporal logic specifications. Preprint available at `https://arxiv.org/abs/2305.09441`.