A Task Decomposition Approach for Signal Temporal Logic Control

Yangrui Zhang†, Shiheng Zhang†, Shaowen Miao, and Yiding Ji*

Abstract— **This paper develops an optimal control approach for nonlinear dynamical systems with Signal Temporal Logic (STL) specifications. We first decompose the given STL task into disjoint "Eventually" and "Always" fragments. The "Eventually" formula is encoded into the objective function using an Attractive Potential Field (APF), and the "Always" formula is enforced by designing time-varying Control Barrier Functions (CBFs) using nonlinear programming. A smooth approximation technique is employed to handle conjunctions of STL formulas, which facilitates gradient-based optimization for synthesis. Our STL task decomposition-based synthesis framework has improved computational efficiency and feasibility compared with conventional mixed integer linear programming and pure CBF approaches, especially for scenarios involving multiple temporal objectives and safety constraints. The performance of our method is demonstrated through numerical simulations in which a mobile robot aims to achieve sequential reachability and obstacle avoidance tasks under temporal constraints.**

Index Terms— **Signal temporal logic, optimal control, control barrier functions, attractive potential field, nonlinear systems**

I. Introduction

With the increasing complexity of autonomous control systems, there is a growing demand for advanced control strategies capable of handling periodic, sequential, and reactive task specifications [1]. In this context, the integration of formal methods with optimal control has attracted significant attention, enabling the execution of complex, multi-objective tasks, meanwhile provably ensuring safety and correctness. Among formal languages, temporal logics such as linear temporal logic (LTL) and signal temporal logic (STL), provide an expressive framework for specifying complex behaviors of dynamic systems [2]. In particular, STL is adequate for describing continuous-time properties with strict timing constraints [3], and has been successfully applied in both theoretical studies and practical implementations [4]–[8].

Despite its advantages, synthesizing controllers to enforce STL specifications remains challenging, particularly when simultaneously optimizing performance and ensuring safety and stability. A common approach is embedding STL tasks as constraints of Mixed-Integer Linear Programming (MILP) or Mixed-Integer Quadratic Programming (MIQP) frameworks [9], [10]. Although tasks are accomplished, these methods typically suffer from high computational complexity and limited scalability, particularly under dense temporal constraints. To mitigate the issue, time-interval decomposition methods have been proposed [11], however, they often suffer from high approximation errors and computation complexity.

To mitigate these issues, recent works have integrated control barrier functions (CBFs) into quadratic programming (QP) for STL control synthesis [12]–[14]. CBFs preserves real-time constraints by ensuring forward invariance of safe sets [15], and has been widely adopted in safety-critical systems [16]–[21]. However, when applied to STL tasks with multiple conjunctions, the feasible set often becomes overly conservative, potentially leading QP to be infeasible. Moreover, encoding "Eventually" operator using CBFs remains challenging due to two key limitations: (1) limited flexibility in specifying timing constraints, and (2) the inherent myopia of CBFs, which means satisfaction is not guaranteed outside the settled temporal intervals [12]. Although variants of CBF, such as finite-time CBFs [22] and control Lyapunov barrier functions (CLBFs) [23] have been proposed to address these limitations, their synthesis remains skillfully demanding.

An alternative approach incorporates STL robustness into the control objectives using smooth approximations such as the log-sum-exp function, then employs gradient-based optimization to design controllers [24]. This framework has been extended to enforce Metric Temporal Logic (MTL) specifications [25]. However, such formulations often lead to overly complicated nonlinear objectives, which renders it challenging to derive globally optimal solutions. Moreover, when the "Always" operator is active, agents may continue optimizing objective functions even after the formula is satisfied, causing undesirable aggressive behaviors.

Motivated by the above challenges, this paper develops a novel STL task decomposition based optimal control approach. First, we decompose the specifications into "Eventually" and "Always" fragments. The "Eventually" formula is incorporated into the objective function via a properly designed attractive potential field (APF), which guides the system to achieve desired objectives via gradient based optimization. Meanwhile, the "Always" formula is encoded as constraints and then enforced with synthesis of time-varying CBF. This decomposition not only improves the feasibility and efficiency of control synthesis but also enhances the controller's robustness with respect to specification conflicts between "Always" and "Eventually" components. Then we provide extensive simulation results to substantiate the performance of our approach, which demonstrates improved trajectory planning performance and satisfaction of tasks compared to existing methods without STL decomposition.

Authors are all with Robotics and Autonomous Systems Thrust, Systems Hub, the Hong Kong University of Science and Technology (Guangzhou), Guangzhou, Guangdong Province, China. (Emails addresses: yzhang112@connect.hkust-gz.edu.cn, florianzsh7@gmail.com, smiao585@connect.hkust-gz.edu.cn, jiyiding@hkust-gz.edu.cn).

†Contributed equally; *Corresponding author.

Financial support: National Natural Science Foundation of China grants 62303389, 62373289; Guangdong Basic and Applied Basic Research Funding grants 2022A151511076, 2024A1515012586; Guangdong Scientific Research Platform and Project Scheme grant 2024KTSCX039; Guangzhou-HKUST(GZ) Joint Funding Program grants 2024A03J0618, 2024A03J0680.

The rest of this paper is structured as follows. Section II introduces the system model, briefly reviews signal temporal logic and control barrier functions, then formulates the STL control problem for investigation. Section III proposes the control synthesis approach, which is validated through numerical simulations in Section IV. Finally, Section V concludes the paper and outlines future research directions.

II. Preliminaries and Problem Formulation

Let $\mathbb{R}$, $\mathbb{R}_{>0}$, and $\mathbb{R}^n$ be the sets of real numbers, positive real numbers, and real vectors of dimensions n. Denote by $\mathbf{1}$ and $\mathbf{0}$ the vectors of all entries being one and zero, respectively. Consider the nonlinear control-affine dynamical system described by the equation:

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}, \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the system state, and $\boldsymbol{u} \in \mathcal{U} \subseteq \mathbb{R}^m$ is the control input. The functions $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ are assumed to be locally Lipschitz continuous. Given a control signal $\boldsymbol{u} : [t_0, t_1] \to \mathcal{U}$, the state $\boldsymbol{x} : [t_0, t_1] \to \mathbb{R}^n$ is said to be a solution to the system in (1) if it is continuous and satisfies the differential equation (1) for all $t \in [t_0, t_1]$.

A. Control Barrier Functions

In this work, the safety specification is formulated as a set

$$\mathcal{C}(t) := \{\boldsymbol{x} \in \mathbb{R}^n \mid b(\boldsymbol{x}, t) \geq 0\} \tag{2}$$

where $b : \mathbb{R}^n \times \mathbb{R}_{>0} \to \mathbb{R}$ is a continuously differential function. We also let $\partial\mathcal{C}(t) = \{\boldsymbol{x} \in \mathbb{R}^n \mid b(\boldsymbol{x}, t) = 0\}$ be the boundary and $\mathrm{Int}(\mathcal{C}(t)) = \{\boldsymbol{x} \in \mathbb{R}^n \mid b(\boldsymbol{x}, t) > 0\}$ be the interior of $\mathcal{C}(t)$, respectively. For safety critical control of the dynamic system, we introduce the following concepts.

Definition 1 (Extend class $\mathcal{K}_\infty$ functions). *A continuous function α: $\mathbb{R}_{>0} \to \mathbb{R}_{>0}$ is called a extend class $\mathcal{K}_\infty$ function if it is strictly increasing with $\alpha(0) = 0$ and is defined on the entire real line $\mathbb{R} = (-\infty, \infty)$.*

Definition 2 (Forward invariance). *A set $\mathcal{C}(t)$ is forward invariant under control law $\boldsymbol{u}$ if for all initial state $\boldsymbol{x_0} \in \mathcal{C}(t_0)$, there exists a continuous trajectory $\boldsymbol{x} : [t_0, t_1] \to \mathbb{R}^n$ of (1) with $\boldsymbol{x}(t_0) = \boldsymbol{x_0}$ and $\boldsymbol{x}(t) \in \mathcal{C}(t)$ for all $t \in [t_0, t_1]$.*

Definition 3 (Control Barrier Function). *Given the dynamic system (1), then function $b : \mathbb{R}^n \times \mathbb{R}_{>0}$ is a control barrier function (CBF) on the set $\mathcal{C}(t)$ defined in (2) if there exists a extend class $\mathcal{K}_\infty$ function α such that for all $(\boldsymbol{x}, t) \in \mathcal{C}(t) \times [t_0, t_1]$, the following condition holds:*

$$\sup_{\boldsymbol{u}\in\mathcal{U}} \left[\frac{\partial b(\boldsymbol{x}, t)}{\partial \boldsymbol{x}}^T (f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}) + \frac{\partial b(\boldsymbol{x}, t)}{\partial t} \right] \geq -\alpha(b(\boldsymbol{x}, t))$$

B. Signal Temporal Logic

Signal Temporal Logic (STL) is a formal language used to specify temporal properties of continuous-time signals. A STL formula consists of predicates μ, whose truth values are determined based on the evaluation of a predicate function $h : \mathbb{R}^n \to \mathbb{R}$ over a continuous signal $\boldsymbol{x}(t)$, as follows:

$$\mu := \begin{cases} \text{True} & \text{if } h(\boldsymbol{x}(t)) \geq 0, \\ \text{False} & \text{if } h(\boldsymbol{x}(t)) < 0. \end{cases} \tag{3}$$

The syntax of STL defines a formula ϕ by

$$\phi ::= \text{True} \mid \mu \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \, U_{[a,b]} \, \phi_2 \mid F_{[a,b]}\phi \mid G_{[a,b]}\phi \tag{4}$$

where ϕ_1 and ϕ_2 are STL formulas, and $a, b \in \mathbb{R}_{\geq 0}$ with $a \leq b$ are temporal bounds. In this syntax, $\neg$ denotes logical negation, and $\wedge$ represents logical conjunction. The temporal operators are defined as follows: $F_{[a,b]}\phi$ expresses that ϕ eventually holds within the time interval $[a, b]$ (Eventually), $G_{[a,b]}\phi$ requires ϕ to hold throughout the interval (Always), and $\phi_1 U_{[a,b]}\phi_2$ specifies that ϕ_1 must hold continuously until ϕ_2 becomes true within $[a, b]$ (Until).

The satisfaction relation $(\boldsymbol{x}, t) \models \phi$ indicates that the signal $\boldsymbol{x} : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ satisfies the STL formula ϕ at time t. The semantics of STL are defined recursively as follows:

$$\begin{aligned}
&(\boldsymbol{x}, t) \models \mu && \Leftrightarrow h(\boldsymbol{x}(t)) \geq 0 \\
&(\boldsymbol{x}, t) \models \neg\phi && \Leftrightarrow \neg((\boldsymbol{x}, t) \models \phi) \\
&(\boldsymbol{x}, t) \models \phi_1 \wedge \phi_2 && \Leftrightarrow (\boldsymbol{x}, t) \models \phi_1 \wedge (x, t) \models \phi_2 \\
&(\boldsymbol{x}, t) \models \phi_1 \, U_{[a,b]} \, \phi_2 && \Leftrightarrow \exists t' \in [t + a, t + b] \text{ s.t. } (\boldsymbol{x}, t') \models \phi_2 \\
& && \quad \wedge \forall t'' \in [t, t'], (\boldsymbol{x}, t'') \models \phi_1 \\
&(\boldsymbol{x}, t) \models F_{[a,b]}\phi && \Leftrightarrow \exists t' \in [t + a, t + b] \text{ s.t. } (\boldsymbol{x}, t') \models \phi \\
&(\boldsymbol{x}, t) \models G_{[a,b]}\phi && \Leftrightarrow \forall t' \in [t + a, t + b], (\boldsymbol{x}, t') \models \phi.
\end{aligned}$$

Additionally, the robustness measure $\rho(\boldsymbol{x}, t, \phi) \in \mathbb{R}$ quantifies the degree of satisfaction of ϕ by signal $\boldsymbol{x}$ at time t:

$$\begin{aligned}
&\rho(\boldsymbol{x}, t, \text{True}) = 1 \\
&\rho(\boldsymbol{x}, t, \mu) = h(\boldsymbol{x}(t)) \\
&\rho(\boldsymbol{x}, t, \neg\phi) = -\rho(\boldsymbol{x}, t, \phi) \\
&\rho(\boldsymbol{x}, t, \phi_1 \wedge \phi_2) = \min\{\rho(\boldsymbol{x}, t, \phi_1), \rho(\boldsymbol{x}, t, \phi_2)\} \\
&\rho(\boldsymbol{x}, t, \phi_1 U_{[a,b]}\phi_2) = \\
&\qquad \sup_{t'\in[t+a,t+b]} \min\left\{ \rho(\boldsymbol{x}, t', \phi_2), \inf_{t''\in[t+a,t']} \rho(\boldsymbol{x}, t'', \phi_1) \right\} \\
&\rho(\boldsymbol{x}, t, F_{[a,b]}\phi) = \sup_{t'\in[t+a,t+b]} \rho(\boldsymbol{x}, t', \phi) \\
&\rho(\boldsymbol{x}, t, G_{[a,b]}\phi) = \inf_{t'\in[t+a,t+b]} \rho(\boldsymbol{x}, t', \phi).
\end{aligned}$$

C. Problem Formulation

Our goal is to design a controller that enforces the given STL specifications of the dynamical system. Then the key problem of this work is formulated below.

Problem 1 (STL control synthesis). *Given the nonlinear control-affine system described in (1) and an STL formula ϕ in (5b), then consider the following STL fragments:*

$$\psi ::= \mathit{True} \mid \mu \mid \neg\mu \mid \psi_1 \wedge \psi_2, \tag{5a}$$

$$\phi ::= F_{[a,b]}\psi \mid G_{[a,b]}\psi \mid \psi_1 U_{[a,b]}\psi_2 \mid \phi_1 \wedge \phi_2, \tag{5b}$$

where ψ_1, ψ_2 are logical expressions, and ϕ_1, ϕ_2 are STL formulas. Then design a control law $\boldsymbol{u}(\boldsymbol{x}, t)$ such that the trajectory $\boldsymbol{x} : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ satisfies ϕ for all time $t \geq 0$.

III. Control Synthesis Approach

This section presents a task decomposition based signal temporal logic (STL) control strategy that leverages STL semantics to synthesize optimal controllers that provably enforce the given specification. Specifically, the STL formula is decomposed as "eventually" and "always" sub-formulas, then we employ attractive potential fields (APF) and control barrier functions (CBF) to design the respective controllers.

A. Decomposition of STL Formulas

In order to decompose the given STL formula, we begin by reformulating the Until operator.

Lemma 1 ([12], [23])**.** *For an atomic formula $\phi := \psi_1 U_{[a,b]} \psi_2$ as defined in (5b), it can be decomposed into the conjunction: $\phi = G_{[a,b]}\psi_1 \wedge F_{[a,b]}\psi_2$.*

If there exists $t' \in [a,b]$ such that $(\boldsymbol{x}, t') \models \psi_2 \wedge \forall t'' \in [a, t'], (\boldsymbol{x}, t'') \models \psi_1$, then *Lemma 1* is a sufficient but not necessary condition for $\phi = G_{[a,t']}\psi_1 \wedge F_{[a,b]}\psi_2$. Although this decomposition does not strictly follow the original definition of STL semantics, it has been widely adopted in control synthesis [12], [23]. Next, Theorem 1 is established.

Theorem 1 (Decomposition of STL formula)**.** *If the composed Eventually and Always formulas are rewritten as:*

$$\phi_F ::= F_{[a,b]}\psi \mid \phi_{F1} \wedge \phi_{F2}, \tag{6a}$$

$$\phi_G ::= G_{[a,b]}\psi \mid \phi_{G1} \wedge \phi_{G2}, \tag{6b}$$

where ϕ_{F1}, ϕ_{F2} and ϕ_{G1}, ϕ_{G2} are recursive instances of Eventually and Always formulas, respectively. Then any STL formula class ϕ can be represented as:

$$\phi = \phi_F \wedge \phi_G. \tag{7}$$

Proof. Consider a formula ϕ defined as in (5b), which contains the conjunction of k Until atomic formulas:

$$\phi = \phi_{F'} \wedge \phi_{G'} \wedge \bigwedge_{i=1}^{k} \left(\psi_{i_1} U_{[a_i,b_i]} \psi_{i_2} \right).$$

By *Lemma 1*, each Until formula $\psi_{i_1} U_{[a_i,b_i]} \psi_{i_2}$ is conservatively approximated by the conjunction for some $t'_i \in [a_i, b_i]$:

$$\psi_{i_1} U_{[a_i,b_i]} \psi_{i_2} \approx F_{[a_i,b_i]}\psi_{i_2} \wedge G_{[a_i,b_i]}\psi_{i_1}.$$

Substituting these expressions, we obtain:

$$\begin{aligned} \phi &= (\phi_{F'} \wedge \bigwedge_{i=1}^{k} F_{[a_i,b_i]}\psi_{i_2}) \wedge (\phi_{G'} \wedge \bigwedge_{i=1}^{k} G_{[a_i,b_i]}\psi_{i_1}) \\ &= \phi_F \wedge \phi_G. \end{aligned} \tag{8}$$

This concludes the proof. □

B. Optimal Control Design Based on Decomposed STL

This subsection introduces optimization-based controller synthesis based on the decomposed STL formula, expressed as $\phi = \phi_F \wedge \phi_G$. Time-varying CBFs are utilized to enforce the composed "Eventually" formula given in (6a) is integrated into the objective function through an APF, meanwhile the composed"Always" formula described in (6b).

In the context of multiple temporal operators, a smooth approximation of the minimum operator is utilized. Given k functions $h_i(\boldsymbol{x}(t))$, the approximation is defined as:

$$\widetilde{h}(\boldsymbol{x}(t)) := -\frac{1}{\eta} \ln \left(\sum_{i=1}^{k} \exp(-\eta h_i(\boldsymbol{x}(t))) \right), \tag{9}$$

where $i \in \{1, \ldots, k\}$, and $\eta > 0$ is an approximation parameter. As established in [24], the expression in (9) serves as a smooth under-approximation of the minimum function, satisfying $\widetilde{h}(\boldsymbol{x}(t)) \leq \min h_i(\boldsymbol{x}(t))$.

1) Composed Eventually Formula: For the composed Eventually formula ϕ_F defined in (6a), the robustness measure is incorporated into the objective function using an APF. Consider an atomic formula without conjunctions of the form $\phi_{F_i} = F_{[a_i,b_i]}\psi_i$, where the robustness of the predicate ψ_i is given by $\rho(\boldsymbol{x}, t, \psi_i) = h_i(\boldsymbol{x}(t))$. To encode the satisfaction of ϕ_{F_i} into the objective function, we adopt a cumulative robustness formulation that is nspired by [24]:

$$\hat{\rho}(\boldsymbol{x}, t, \phi_{F_i}) = \sum_{t' \in [a_i, b_i]} \rho(\boldsymbol{x}, t', \psi_i). \tag{10}$$

This cumulative measure aggregates robustness over the interval $[a_i, b_i]$, unlike traditional robustness metrics that only consider the critical time instant (e.g., the supremum). The goal is to maximize $\hat{\rho}(\boldsymbol{x}, t, \phi_{F_i})$ over the entire interval, improving the temporal satisfaction of the Eventually operator.

For real-time feedback controller design, since $\hat{\rho}(\boldsymbol{x}, t, \phi_{F_i})$ is accumulated by $\rho(\boldsymbol{x}, t, \psi_i)$, $\rho(\boldsymbol{x}, t, \psi_i)$ is directly set as the stage optimization objective. This formulation induces an APF associated with ϕ_{F_i} for the system described in (1) [26]:

$$A_{\phi_{F_i}}(\boldsymbol{x}, t) = \frac{1}{2} K_F \parallel \rho(\boldsymbol{x}, t, \mathrm{True}) - \hat{\rho}(\boldsymbol{x}, t, \psi_i) \parallel^2, \tag{11}$$

where $K_F > 0$ is a tuning parameter and its gradient with respect to $\rho(\boldsymbol{x}, t, \psi_i)$ is given by:

$$\begin{aligned} \nabla A_{\phi_{Fi}}(\boldsymbol{x}, t) =& \frac{\partial A_{\phi_{Fi}}(\boldsymbol{x}, t)}{\partial \rho(\boldsymbol{x}, t, \psi_i)} \\ =& K_F \parallel \rho(\boldsymbol{x}, t, \mathrm{True}) - \rho(\boldsymbol{x}, t, \psi_i) \parallel . \end{aligned} \tag{12}$$

Since $\nabla A_{\phi_{F_i}}(\boldsymbol{x}, t)$ decreases along the trajectory that satisfies the predicate $\mu = \mathrm{True}$, it is employed as the minimized objective in the optimal control problem.

Next, we extend the composed Eventually formula ϕ_F with conjunctions. For $\phi_F = \bigwedge_{i=1}^{k} \phi_{Fi}$, the cumulative robustness measure is given by:

$$\hat{\rho}(\boldsymbol{x}, t, \phi_F) = \sum_{t' \in \bigcup_{i=1}^{k} [a_i, b_i]} \min \rho(\boldsymbol{x}, t', \psi_i). \tag{13}$$

Similar to the no-conjunction case, the optimization objective for a real-time feedback controller is $\min \rho(\boldsymbol{x}, t, \psi_i)$. To facilitate gradient-based optimization, inspired by (9) and (13), we derive a smooth under-approximation, denoted as the approximate robustness measure of ϕ_F:

$$\widetilde{\rho}(\boldsymbol{x}, t, \phi_F) = -\frac{1}{\eta_1} \ln \left(\sum_{i=1}^{k} \exp(-\eta_1 \rho(\boldsymbol{x}, t, \psi_i)) \right), \tag{14}$$

where $\eta_1 > 0$ is a parameter related to approximation accuracy. Then the composed APF is:

$$\nabla A_{\phi_F}(\boldsymbol{x}, t) = K_F \parallel \rho(\boldsymbol{x}, t, \mathrm{True}) - \widetilde{\rho}(\boldsymbol{x}, t, \phi_F) \parallel . \tag{15}$$

Compared to the CBF-based formulation in [12], the proposed APF-based method avoids enforcing the "Eventually" operator as a hard constraint over conservative time intervals. Instead, it incorporates task satisfaction into the objective function, making the controller more feasible.

2) Composed Always Formula: For ϕ_G as defined in (6b), following the approach proposed in [12], we adopt CBFs to encode the STL constraints within the optimal controller. We begin by considering the Always atomic formula $\phi_{G_i} = G_{[a_i,b_i]}\psi_i$ without conjunctions. The robustness of the predicate ψ_i is defined as $\rho(\boldsymbol{x}, t, \psi_i) = h_i(\boldsymbol{x}(t))$. Assuming that $(\boldsymbol{x}, 0) \models \phi_{G_i}$, a candidate CBF is constructed as

$$b_i(\boldsymbol{x}, t) := \rho(\boldsymbol{x}, t, \psi_i) - \gamma_i(t),$$

where $\gamma_i(t)$ is a safety margin function. For a given signal $\boldsymbol{x}$ such that $b_i(\boldsymbol{x}, t) \geq 0$ for all $t \geq 0$, it follows that $(\boldsymbol{x}, t) \models \phi_{G_i}$ for all $t \geq 0$, which implies $(\boldsymbol{x}, 0) \models \phi_{G_i}$.

We then analyze the composed Eventually formula ϕ_G with conjunctions. Consider $\phi_G = \bigwedge_{i=1}^{k} \phi_{G_i}$, where $\phi_{G_i} = G_{[a_i,b_i]}\psi_i$ corresponds to an atomic Always formula. If CBF $b_i(\boldsymbol{x}, t)$ satisfy the conditions outlined in [12], then a composed CBF $b(\boldsymbol{x}, t)$ can be constructed using a smooth approximation (9) of the minimum operator as follows:

$$b(\boldsymbol{x}, t) = -\frac{1}{\eta_2} \ln \left(\sum_{i=1}^{k} \exp(-\eta_2 b_i(\boldsymbol{x}, t)) \right), \tag{16}$$

where $\eta_2 > 0$ is a tuning parameter related to approximation accuracy. The corresponding constraint, defined in accordance with Definition 3, is given by:

$$\frac{\partial b(\boldsymbol{x}, t)}{\partial \boldsymbol{x}}(f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}) + \frac{\partial b(\boldsymbol{x}, t)}{\partial t} \geq -\alpha(b(\boldsymbol{x}, t)), \tag{17}$$

where α is a class $\mathcal{K}$ function that ensures forward invariance of the safe set defined by the CBF. Satisfaction of the inequality (17) ensures that $(\boldsymbol{x}, t) \models \phi_G$ holds for all $t \geq 0$.

3) Controller Synthesis: Based on (15) and (16), the optimal control input is obtained by combining APFs and CBFs to satisfy the "Eventually" and "Always" formulas, respectively. Then the optimization problem is formulated:

$$J^*(\boldsymbol{x}) = \min_{\boldsymbol{u} \in \mathcal{U}} \boldsymbol{u}^\top Q \boldsymbol{u} + \nabla A_{\phi_F}(\boldsymbol{x}, t), \tag{18a}$$

$$\text{s.t.} \quad \frac{\partial b(\boldsymbol{x}, t)}{\partial \boldsymbol{x}}(f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}) + \frac{\partial b(\boldsymbol{x}, t)}{\partial t} \geq -\alpha(b(\boldsymbol{x}, t)), \tag{18b}$$

where $J^*(\boldsymbol{x})$ is the objective function and Q is a positive definite matrix.

IV. Numerical Simulation

This section provides a numerical case study to evaluate the performance of the proposed STL control approach. The simulations are conducted in Python language using the IPOPT solver [27]. Results demonstrate that our method outperforms several benchmarks under different tasks.

A. Simulation Settings

We consider a mobile robot modeled by a two-dimensional double integrator system. The state vector is defined as $\boldsymbol{x} = [x, y, v_x, v_y]^\top$, where (x, y) represents the position and (v_x, v_y) denotes the velocity in the X and Y directions, respectively. The control input vector $\boldsymbol{u} = [u_x, u_y]^\top$ corresponds to the accelerations in each direction.

The robot operates in a bounded 2-D environment illustrated in Fig. 1. Within this environment, the robot is required to complete a set of temporally constrained tasks specified by an STL formula within 150 seconds. Let the robot position be denoted by $p := (x, y)$. The three target regions, marked in green, are specified by $p_{re1} := (x_{re1}, y_{re1})$, $p_{re2} := (x_{re2}, y_{re2})$ and $p_{re3} := (x_{re3}, y_{re3})$, which the robot must reach sequentially. Additionally, two red-marked regions represent obstacles, located at $p_{ob1} = (x_{ob1}, y_{ob1})$ and $p_{ob2} = (x_{ob2}, y_{ob2})$, which must be avoided.

Fig. 1: Simulation environment with targets and obstacles.

The STL specification is: $\phi := \phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4$, where

$$\phi_1 := F_{[40:60]}(\| p - p_{re1} \| \leq 1.5),$$
$$\phi_2 := F_{[80:100]}(\| p - p_{re2} \| \leq 1.5),$$
$$\phi_3 := G_{[0:150]}(\| p - p_{ob1} \| \geq 2.5),$$
$$\phi_4 := (\| p - p_{ob2} \| \geq 2.5) U_{[100:150]}(\| p - p_{re3} \| \leq 1.5).$$

Note that ϕ imposes the following temporal requirements: the robot must start from the origin, reach Region 1 within $[40, 60]$ seconds, and subsequently reach Region 2 within $[80, 100]$ seconds. Additionally, it must maintain a safe distance of at least 2.5 units from Obstacle 2 during the interval $[100, 150]$ seconds until it successfully reaches Region 3, and avoid Obstacle 1 throughout the entire 150-second time horizon. To facilitate controller synthesis, the Until operator in ϕ_4 is decomposed into two components as follows:

$$\phi_4 = \phi_{F4} \wedge \phi_{G4}$$

where $\phi_{F4} := F_{[100:150]}(\| p - p_{re3} \| \leq 1.5)$ and $\phi_{G4} := G_{[100:150]}(\| p - p_{ob2} \| \geq 2.5)$. This enables the transformation of the original formula ϕ into a conjunction of Eventually and Always components: $\phi_F = \phi_1 \wedge \phi_2 \wedge \phi_{F4}$ and $\phi_G = \phi_3 \wedge \phi_{G4}$. The proposed optimal control method in Section III is then employed to synthesize a feedback controller that satisfies the decomposed STL formula.

B. Simulation Results

We evaluate the performance of our approach and compare it with two existing baselines under the same parameters:

- Approach 1: Encodes the entire STL specification as hard constraints using CBFs.
- Approach 2: Incorporates the cumulative STL robustness metric into the objective function, without decomposition.
- Approach 3 (Our method): Employs a task decomposition-based control strategy that integrates APFs for the Eventually formulas and CBFs for the Always formulas.

To better illustrate the STL task completion, we define a variant of robustness measure denoted by $\rho_o(\boldsymbol{x}, t, \phi)$ for the signal $\boldsymbol{x}$ at time t, which is used in the remainder of the simulation section when there is no confusion.

$$
\begin{aligned}
&\rho_o(\boldsymbol{x}, t, \text{True}) = 1 \\
&\rho_o(\boldsymbol{x}, t, \mu) = \rho(\boldsymbol{x}, t, \mu) \\
&\rho_o(\boldsymbol{x}, t, \neg\phi) = -\rho_o(\boldsymbol{x}, t, \phi) \\
&\rho_o(\boldsymbol{x}, t, \phi_1 \wedge \phi_2) = \min\{\rho_o(\boldsymbol{x}, t, \phi_1), \rho_o(\boldsymbol{x}, t, \phi_2)\} \\
&\rho_o(\boldsymbol{x}, t, F_{[a,b]}\phi) = \rho_o(\boldsymbol{x}, t, \phi) \\
&\rho_o(\boldsymbol{x}, t, G_{[a,b]}\phi) = \rho_o(\boldsymbol{x}, t, \phi) \\
&\rho_o(\boldsymbol{x}, t, \phi_1 U_{[a,b]} \phi_2) = \\
&\qquad\qquad \min\{\rho_o(\boldsymbol{x}, t, F_{[a,b]}\phi_2), \rho_o(\boldsymbol{x}, t, G_{[a,b]}\phi_1)\}.
\end{aligned}
$$

Then for Eventually, Always and Until formulas, we have:

$$
\begin{aligned}
(\boldsymbol{x}, t) \models \phi_1\, U_{[a,b]}\, \phi_2 \quad &\Leftrightarrow \exists t' \in [t+a, t+b] \\
&\quad \text{s.t. } \rho_o(\boldsymbol{x}, t, \phi_1 U_{[a,b]} \phi_2) > 0 \\
(\boldsymbol{x}, t) \models F_{[a,b]}\phi \quad &\Leftrightarrow \exists t' \in [t+a, t+b] \\
&\quad \text{s.t. } \rho_o(\boldsymbol{x}, t, F_{[a,b]}\phi) > 0 \\
(\boldsymbol{x}, t) \models G_{[a,b]}\phi \quad &\Leftrightarrow \forall t' \in [t+a, t+b] \\
&\quad \text{s.t. } \rho_o(\boldsymbol{x}, t, G_{[a,b]}\phi) > 0.
\end{aligned}
$$

The simulation results are illustrated in Figs. 2–4. Fig. 2 shows an overview of the trajectories under each approach. The robot starts from $(0, 0)$ and sequentially attempts to complete task ϕ. Fig. 3 presents more detailed subplots of robot trajectories for each method. The time t_1, t_2, and t_3 indicate when ϕ_1, ϕ_2, and ϕ_4 are satisfied, respectively.

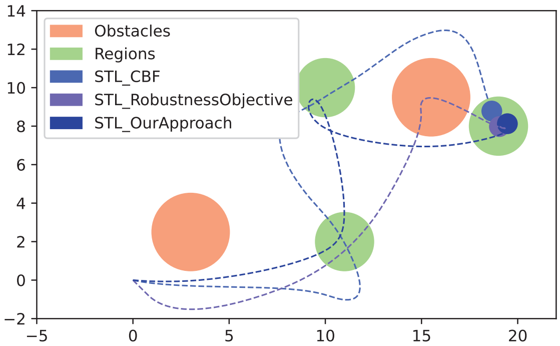


Fig. 2: Comparison of robot trajectories

Fig. 3a (Approach 1) illustrates a conservative path that strictly respects safety constraints via CBFs. Fig. 3b (Approach 2) shows failure to reach the second region and a collision with Obstacle 2, due to conflicting objectives in the robustness formulation. Fig. 3c (Approach 3) demonstrates an efficient trajectory that fulfills all temporal tasks.

The temporal evolution of critical metrics under each approach is shown in Fig. 4. Fig. 4a shows the CBF values in Approach 1, confirming the satisfaction of constraint $b(\boldsymbol{x}, t) \geq 0$. Fig. 4b depicts the robustness for Approach 2, where persistently negative values in the interval $[80, 100]$ indicate violation of ϕ_2. Fig. 4c shows the robustness for Approach 3, which remains positive in sufficient time intervals to signal task completeness. Detailed analysis of each approach's performance is presented below:

- Approach 1 ensures strict satisfaction of the STL task by enforcing CBF constraints. As shown in Fig. 4a, the constraint $b(\boldsymbol{x}, t) \geq 0$ is always satisfied. However, encoding the "Eventually" operator as a hard constraint results in overly conservative behavior and longer trajectories.
- Approach 2 optimizes STL robustness. However, the robot fails to reach Region 2 and finally collides with Obstacle 2 due to conflicting temporal goals between ϕ_2 and ϕ_4,
- Approach 3 effectively balances task satisfaction and safety by integrating robustness-based objectives and safety constraints within a unified optimization framework. The robot successfully completes all given tasks without any violations and demonstrates a more efficient trajectory than the baseline methods that only involve CBFs.

The quantitative results are summarized in Table I, showing that our approach (Approach 3) outperforms the others by achieving total task satisfaction with no violations while maintaining efficiency in trajectory length. Unlike the CBF-only method (Approach 1), it balances safety and goal attainment, and unlike the robustness-only method (Approach 2), it avoids conflicts in temporal objectives.

TABLE I. OBSERVATION DATA

Data	Approach 1	Approach 2	Approach 3
t_1	57	47	41
t_2	96	None	79
t_3	148	105	121
Distance	37.695	26.470	31.041
Violating Times	0	26	0

V. CONCLUSIONS

This paper proposed a task decomposition-based optimal control framework for nonlinear systems to achieve Signal Temporal Logic (STL) specifications. Specifically, the STL task was decomposed into two types of formulas: "Eventually" and "Always". Attractive potential fields (APF) were employed to drive the system towards satisfying the Eventually formula. Time-varying control barrier functions were designed to enforce the Always formula. Then an optimization problem integrating the two formulations is addressed, which generates the desirable controller. Simulation results demonstrated that the proposed approach reliably achieves complex temporal tasks with strict timing and safety requirements, even in dynamic and constrained environments. Future research will extend the framework to multi-agent systems, enhance the scalability and incorporate learning-based techniques adaptive to uncertain environments.

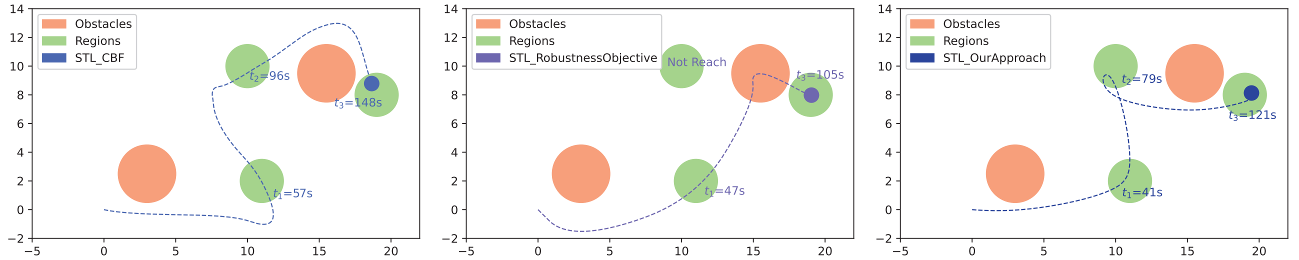
REFERENCES

[1] W. Xiao, C. G. Cassandras, and C. Belta, *Safe Autonomy with Control Barrier Functions*. Springer, 2023.

[2] C. Belta, B. Yordanov, and E. A. Gol, *Formal Methods for Discrete-Time Dynamical Systems*. Springer, 2017.

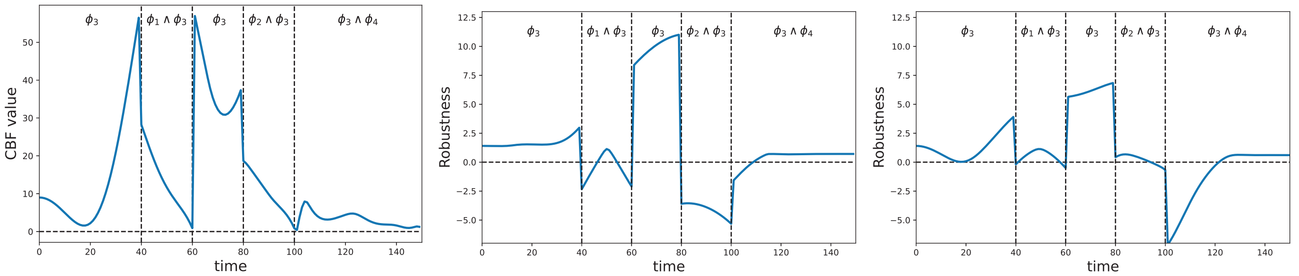
[3] O. Maler and D. Nickovic, "Monitoring temporal properties of continuous signals," in *International Symposium on Formal Techniques in Real-Time and Fault-Tolerant Systems*, pp. 152–166, 2004.

[4] S. Wang, X. Yin, S. Li, and X. Yin, "Tractable reinforcement learning for signal temporal logic tasks with counterfactual experience replay," *IEEE Control Systems Letters*, vol. 8, pp. 616–621, 2024.



(a) (b) (c)

Fig. 3: Robot trajectories generated by different STL control strategies. (a) Approach 1: STL encoded as CBF constraints. (b) Approach 2: STL cumulative robustness used as the optimization objective. (c) Approach 3: Proposed decomposition-based control structure combining APFs and CBFs.



(a) (b) (c)

Fig. 4: Temporal evolution of safety and robustness metrics under each control method. (a) CBF constraint values in Approach 1. (b) Observable STL robustness in Approach 2. (c) Observable STL robustness in Approach 3.

[5] Z. Huang, W. Lan, and X. Yu, "A formal control framework of autonomous vehicle for signal temporal logic tasks and obstacle avoidance," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, p. 1930–1940, 2024.

[6] C. Zhou, J. Yang, S. Li, and W.-H. Chen, "Temporal logic disturbance rejection control of nonlinear systems using control barrier functions," *IEEE Trans. on Cybernetics*, vol. 55, no. 3, pp. 1008–1017, 2025.

[7] X. Mao, X. Li, Y. Huang, J. Shi, and Y. Zhang, "Programmable logic controllers past linear temporal logic for monitoring applications in industrial control systems," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 7, pp. 4393–4405, 2021.

[8] D. Tian, H. Fang, Q. Yang, Z. Guo, J. Cui, W. Liang, and Y. Wu, "Two-phase motion planning under signal temporal logic specifications in partially unknown environments," *IEEE Transactions on Industrial Electronics*, vol. 70, no. 7, pp. 7113–7121, 2022.

[9] V. Raman, M. Maasoumy, and A. Donzé, "Model predictive control with signal temporal logic specifications," in *Intl. Symp. on Formal Techniques in Real-Time and Fault-Tolerant Sys.*, pp. 52–55, 2014.

[10] S. Sadraddini and C. Belta, "Robust temporal logic model predictive control," in *53rd Annual Allerton Conference on Communication, Control and Computing*, p. 772–779, 2015.

[11] X. Yu, C. Wang, D. Yuan, S. Li, and X. Yin, "Model predictive control for signal temporal logic specifications with time interval decomposition," in *IEEE Conf. on Dec. and Con.*, p. 7849–7855, 2023.

[12] L. Lindemann and D. V. Dimarogonas, "Control barrier functions for signal temporal logic tasks," *IEEE Control Systems Letters*, vol. 3, no. 1, p. 96–101, 2019.

[13] X. Huang, L. Li, and J. Chen, "Multi-agent system motion planning under temporal logic specifications and control barrier function," *Control Theory and Technology*, vol. 18, no. 3, pp. 269–278, 2020.

[14] J. Chen, Y. Zou, and S. Li, "Safe reinforcement learning for signal temporal logic tasks using robust control barrier functions," in *2023 42nd Chinese Control Conference (CCC)*, pp. 8627–8632, IEEE, 2023.

[15] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *18th European control conference*, pp. 3420–3431, 2019.

[16] J. Xie, S. Zhao, L. Hu, and H. Gao, "Certificated actor-critic: Hierarchical reinforcement learning with control barrier functions for safe navigation," *arXiv preprint arXiv:2501.17424*, 2025.

[17] B. Li, S. Wen, Z. Yan, G. Wen, and T. Huang, "A survey on the control lyapunov function and control barrier function for nonlinear-affine control systems," *IEEE/CAA Journal of Automatica Sinica*, vol. 10, no. 3, pp. 584–602, 2023.

[18] C. Peng, X. Liu, and J. Ma, "Design of safe optimal guidance with obstacle avoidance using control barrier function-based actor–critic reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 11, pp. 6861–6873, 2023.

[19] J. Sun, J. Yang, and Z. Zeng, "Safety-critical control with control barrier function based on disturbance observer," *IEEE Transactions on Automatic Control*, vol. 69, no. 7, pp. 4750–4756, 2024.

[20] S. Zhang, D.-H. Zhai, Y. Xiong, J. Lin, and Y. Xia, "Safety-critical control for robotic systems with uncertain model via control barrier function," *International Journal of Robust and Nonlinear Control*, vol. 33, no. 6, pp. 3661–3676, 2023.

[21] J. Xu, J. Wang, J. Rao, Y. Zhong, and H. Wang, "Adaptive dynamic programming for optimal control of discrete-time nonlinear system with state constraints based on control barrier function," *Intl. Journal of Robust and Nonlinear Control*, vol. 32, no. 6, pp. 3408–3424, 2022.

[22] M. Srinivasan, S. Coogan, and M. Egerstedt, "Control of multi-agent systems with finite time control barrier certificates and temporal logic," in *IEEE 57th Conf. on Decision and Control*, p. 1991–1996, 2018.

[23] W. Xiao, C. A. Belta, and C. G. Cassandras, "High order control lyapunov-barrier functions for temporal logic specifications," in *American Control Conference*, p. 4886–4891, 2021.

[24] I. Haghighi, N. Mehdipour, E. Bartocci, and C. Belta, "Control from signal temporal logic specifications with smooth cumulative quantitative semantics," in *IEEE 58th Conference on Decision and Control*, pp. 4361–4366, 2019.

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

[26] A. Singletary, K. Klingebiel, J. Bourne, A. Browning, P. Tokumaru, and A. Ames, "Comparative analysis of control barrier functions and artificial potential fields for obstacle avoidance," in *IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, p. 8129–8136, 2021.

[27] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, pp. 25–57, 2006.