Model-Based Reinforcement Learning for Approximate Optimal Control with Temporal Logic Specifications

Max H. Cohen  
cmaxohen@bu.edu  
Boston University  
Boston, MA

Calin Belta  
cbelta@bu.edu  
Boston University  
Boston, MA

ABSTRACT

In this paper we study the problem of synthesizing optimal control policies for uncertain continuous-time nonlinear systems from syntactically co-safe linear temporal logic (scLTL) formulas. We formulate this problem as a sequence of reach-avoid optimal control sub-problems. We show that the resulting hybrid optimal control policy guarantees the satisfaction of a given scLTL formula by constructing a barrier certificate. Since solving each optimal control problem may be computationally intractable, we take a learning-based approach to approximately solve this sequence of optimal control problems online without requiring full knowledge of the system dynamics. Using Lyapunov-based tools, we develop sufficient conditions under which our approximate solution maintains correctness. Finally, we demonstrate the efficacy of the developed method with a numerical example.

CCS CONCEPTS

• Computing methodologies → Reinforcement learning; Online learning settings; Computational control theory; • Theory of computation → Modal and temporal logics.

KEYWORDS

Optimal control, Hybrid systems, Safe reinforcement learning

INTRODUCTION

The intersection of hybrid systems [18], formal methods [3], and control theory provides a pathway towards controlling dynamical systems to achieve objectives richer than stability [5]. These objectives can range from simple reachability and safety specifications [31, 33] to more complex specifications such as temporal logic (TL) formulas. Classical approaches to designing controllers for dynamical systems subject to TL specifications involve abstraction-based techniques [5, 46]. In this setting, the dynamical system is abstracted into a finite transition system (FTS) whereas the specification, often represented as a linear TL (LTL) formula, is captured by a finite-state automaton (FSA). This FTS and FSA can then be combined to form a product automaton and the problem of synthesizing a correct (with respect to the specification) control policy for the original system is reduced to solving a Büchi or Rabin game over the product automaton. These approaches have been extensively studied in the literature [25, 47]; however, a well-documented limitation is the computational burden associated with the abstraction process.

To address the limitations of abstraction-based control design, more recent work has focused on optimization-based control with TL constraints [4]. Here, TLs with quantitative semantics, such as signal temporal logic (STL) [32], are leveraged to form optimization problems with the goal of maximizing the satisfaction of the desired specification. These approaches often rely on encoding Boolean satisfaction of STL formulas as mixed-integer constraints, leading to mixed-integer linear/quadratic programs (MILP/MIQP) that can be solved within model predictive control (MPC) frameworks [37, 40]. Although these approaches avoid the computationally prohibitive abstraction process, solving MILPs/MIQPs in a receding horizon fashion is still computationally intensive and may prevent real-time implementation. To address this limitation, other approaches have explored using control barrier functions (CBFs) [2] to encode subsets of LTL or STL formulas within more computationally efficient quadratic programs [30, 44].

Recently, optimization-based control from TL specifications has also been explored within more traditional Hamilton-Jacobi-Bellman (HJB)-based optimal control frameworks [17, 35, 45]. Similar to abstraction-based approaches, these methods rely on translating a TL formula into an FSA but circumvent the computational expense of abstracting the dynamical system. The authors in [17, 35] augment a continuous dynamical system with the discrete behavior of an FSA corresponding to a syntactically co-safe LTL (scLTL) formula to form a hybrid dynamical system. The authors then solve a mixed continuous-discrete-time HJB equation corresponding to an optimal control problem to obtain a correct-by-construction hybrid feedback controller. For linear systems, this problem can be solved through semi-definite programming; however, for nonlinear systems this same approach results in a semi-infinite program, which is difficult to solve in general. Other recent work in this domain includes [45], where the authors propose an approximate dynamic programming (ADP) approach to solve a sequence of two-point boundary value problems, whose solution yields a control policy capable of satisfying an scLTL formula.
With the exception of accounting for bounded external disturbances, the aforementioned approaches rely on exact model knowledge. In practice, however, models often exhibit some degree of uncertainty, which motivates the use of learning-based control methods that can handle uncertainty while preserving formal guarantees. One way to address this challenge is to take a completely model-free approach as in [1, 39]; however, it is generally difficult to maintain formal guarantees in a model-free setting [42]. An alternative approach is to take a model-based approach wherein an approximate model of the system is identified, which is then used to establish formal guarantees. Along these lines, the authors in [41] leverage techniques from adaptive control to control discrete-time systems with uncertain parameters from STL specifications; however, the approach is abstraction-based. On the other hand, the authors in [42] use input-output data to construct a set-valued piecewise affine model of an uncertain system that is then used in a MPC framework to synthesize a controller from an STL formula. Although this approach can provide formal guarantees for uncertain systems, identifying the uncertain system is computationally intensive and cannot be performed during run-time.

In this paper, we seek to combine optimality, correctness, and learning to synthesize approximately optimal controllers for uncertain systems with formal guarantees online. Specifically, we consider deterministic continuous-time nonlinear control-affine systems with unknown drift terms subject to specifications expressed as scLTL formulas defined over a finite set of observations. To link the continuous behavior of the system with the discrete behavior of the scLTL formula’s FSA, we form a hybrid system [18] and associate with each observation of the formula a subset of the state-space, referred to as regions of interest (ROI) throughout this paper. Leveraging the approach from [10], we construct a sequence of reach-avoid optimal control problems to regulate the system from ROI to ROI while avoiding “bad” regions in the state-space. We then show that the value functions obtained from solving the corresponding sequence of HJB equations can be used to form a hybrid barrier certificate [6] to establish formal guarantees. However, solving these HJB equations can be computationally intractable and requires perfect knowledge of the system dynamics. To address this challenge, we leverage a computationally efficient model-based reinforcement learning approach [20, 21, 38] to learn the unknown dynamics and value functions simultaneously online. We show that, under suitable conditions, the resulting approximate optimal control policy provides formal guarantees of correctness with respect to the given specification.

The remainder of this paper is organized as follows. Section 2 covers preliminaries related to scLTL and hybrid systems [18]. Section 3 formulates the problem. Sections 4 and 5 present our optimal control framework. Section 6 outlines our reinforcement learning-based solution approach. Section 7 provides numerical examples that support the theoretical findings and Section 8 contains concluding marks and a discussion on future research directions.

2 PRELIMINARIES

2.1 Notation

For a set $C$, the notations $\partial C$, Int($C$), cl($C$), $2^C$ denote the boundary, interior, closure, and power set of $C$, respectively. A continuous function $\alpha : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ is a class $K$ function if $\alpha(0) = 0$ and it is strictly increasing. The operator $||\cdot||$ denotes the 2-norm and $\lambda_{\text{min}}(\cdot)$, $\lambda_{\text{max}}(\cdot)$ denote the minimum and maximum eigenvalues of a matrix, respectively. The partial derivative of a continuously differentiable function $f(x_1, x_2, \ldots)$ with respect to its $i$th argument $x_i$ is denoted by $\nabla_{x_i} f(x_1, x_2, \ldots) = \frac{df}{dx_i}(x_1, x_2, \ldots)$. We use $F : \mathbb{R}^n \Rightarrow \mathbb{R}^n$ to indicate that $F(x)$ is a set-valued mapping.

2.2 Syntactically co-safe Linear Temporal Logic

In this paper, we consider control specifications in the form of syntactically co-safe linear temporal logic (scLTL) formulas:

Definition 1 (scLTL syntax [5]). Syntactically co-safe LTL formulas are recursively defined over a set of observations $O$ as

$$\phi = T | o | \neg o | \phi_1 \land \phi_2 | \bigcirc \phi | \phi_1 U \phi_2, \quad (1)$$

where $o \in O$ is an observation, $T$ is the Boolean constant True, $\neg$ and $\land$ stand for Boolean negation and conjunction, $\bigcirc$ and $U$ are the “next” and “until” temporal operators, and $\phi, \phi_1$, and $\phi_2$ are scLTL formulas.

Other temporal operators, such as the eventually operator $\bigcirc$, can be derived using (1) (see [5, Ch. 2] for details). scLTL formulas can be used to specify rich behaviors for dynamical systems such as reachability and avoidance through formulas such as $\phi = (\diamond o_1 \lor \diamond o_2) \land (\neg o_3 U (o_1 \lor o_2))$, which requires a system to eventually reach $o_1$ or $o_2$ and never visit $o_3$. Traditionally, the semantics of scLTL formulas are defined over infinite words over $2^O$; however, in this paper, we adopt the approach taken in [5] and consider formulas defined over infinite words over $O$. As it will become clear later, this is not restrictive, as we will use $O$ to label non-overlapping regions in the state space of a dynamical system, and at most one observation can be satisfied at any time. Although the semantics of scLTL formulas are interpreted over infinite words, their satisfaction can be decided over finite prefixes of such words. Specifically, any infinite word that satisfies an scLTL formula $\phi$ defined over $O$ also contains a finite prefix such that all infinite words that contain this prefix satisfy $\phi$. The language of $\phi$, denoted by $L(\phi)$, is the set of all finite prefixes of infinite words that satisfy $\phi$. For any scLTL formula $\phi$ over $O$, there exists a finite state automaton $A$ (see Def. 2 below) that accepts $L(\phi)$ [26], which can be constructed using tools such as check2 [27].

Definition 2 (Finite state automaton [5]). A finite state automaton (FSA) is a tuple $A = (S, s_0, O, p, S_f)$ where $S$ is a finite set of states, $s_0 \in S$ is the initial state, $O$ is the input alphabet, $p : S \times O \rightarrow S$ is a transition function, and $S_f \subseteq S$ is the set of accepting states.

The semantics of an FSA are defined over finite words over $O$. An input word $w_O = w_O(1)w_O(2), \ldots, w_O(N)\in O$, $i = 1, \ldots, N$ produces a run $w_S = w_S(1)w_S(2), \ldots, w_S(N+1)\in S$, $i = 1, \ldots, N + 1$, such that $w_S(1) = s_0$ and $p(w_S(i), w_O(i)) =$

\footnote{In this paper, all FSAs are assumed to be deterministic. Note that any non-deterministic FSA can be easily determinized.}
w_2(i+1) for all i = 1, 2, . . . , N. We say that a word w_2 is accepted by the FSA if and only if the run w_2 produced by w_2 ends in an accepting state (w_2(N + 1) ∈ S_f). Throughout this paper, we assume that a scLTL formula is feasible, in the sense that, in the corresponding FSA, there exists a path from the initial state s_0 to the set of accepting states S_f [6, Assumption 1].

2.3 Hybrid Dynamical Systems

Following [18], a hybrid system with state x ∈ R^n is a tuple H = (F, G, C, D), where F : R^n → R^n is the flow map, G : R^n → R^n is the jump map, C ⊂ R^n is the flow set, and D ⊂ R^n is the jump set. The semantics of H are given by

\[
\begin{align*}
    H : \left\{ \begin{array}{l}
        \dot{x} \in F(x), \quad x \in C, \\
        x^+ \in G(x), \quad x \in D.
    \end{array} \right.
\end{align*}
\]

In essence, when the state x belongs to the flow set C, it continuously evolves according to the differential inclusion in (2a) and when it belongs to the jump set D it changes, discretely, according to the difference inclusion in (2b). A hybrid time domain, denoted by E ⊂ R≥0 × N, is the union of a (possibly infinite) sequence of intervals of the form \{t_j, t_{j+1}\} × \{j\} \[18, Def. 2.3\]. A hybrid arc is a function ξ : E → R^n such that for each j ∈ N, t → ξ(t, j) is locally absolutely continuous on the interval I_j := \{t : (t, j) ∈ E\} \[18, Def. 2.4\]. If a hybrid arc ξ(t, j) satisfies additional conditions outlined in \[18, Def. 2.6\] then ξ is said to be a solution to (2). A solution to (2) is said to be maximal, denoted by ξ ∈ S_H, if it cannot be extended \[18, Def. 2.7\]. A complete exposition on the class of hybrid systems defined in (2) is beyond the scope of this paper and we refer the unfamiliar reader to \[18, Ch. 2\] for an excellent introduction. Similar to \[6, 7\], in this paper, we leverage this hybrid modeling framework to combine the continuous dynamics of the plants under consideration with the discrete dynamics of the FSA corresponding to a given scLTL formula.

Definition 3 (Eventuality [6]). Consider a hybrid system as in (2) and a closed set R ⊂ C ∪ D. The eventualty property for H with respect to R holds if for each maximal solution ξ ∈ S_H there exists some finite time T ≥ 0 and finite number of jumps \(j ≥ 0\) such that:

1. For t + j < T + j we have ξ(t, j) ∈ C ∪ D,
2. For t = T and j = 0 we have ξ(t, 0) ∈ R.

Intuitively, the eventualty property for a hybrid system H with respect to a set R holds if the solutions to H reach R in finite time and remain within the flow and jumps sets for all times beforehand. We will use this property in conjunction with the acceptance condition of a FSA to enforce a temporal logic specification. As shown in [6], the eventualty property can be certified by finding a hybrid barrier certificate, defined as follows:

Definition 4 (Hybrid barrier certificate [6]). Let R ⊂ C ∪ D be a closed set satisfying G(cl(D|R)) ⊂ C ∪ D. A function \(V_B\) is a hybrid barrier certificate for a hybrid system H with respect to R if it is continuous on cl(C|R) ∪ cl(D|R), differentiable on an open neighborhood of cl(C|R), and there exists a positive constant \(\epsilon_B > 0\) such that the following conditions hold:

1. \(V_B\) is bounded from below on cl(C|R) ∪ cl(D|R),
2. \(V_B\) is radially unbounded on cl(C|R) ∪ cl(D|R),
3. \(\nabla_x V_B(x) F(x) < -\epsilon_B\) for all x ∈ cl(C|R) and F ∈ F(x),
4. \(V_B(G(x)) - V_B(x) < -\epsilon_B\) for all x ∈ cl(D|R) and G ∈ G(x).

The following theorem shows that the existence of a hybrid barrier certificate for H enforces the eventualty property:

Theorem 1 ([6]). Consider a hybrid system (2) that satisfies the hybrid basic conditions \[18, Assumption 6.5\] and let R ⊂ C ∪ D be a closed set that satisfies the properties detailed in Definition 4. If there exists a hybrid barrier certificate for H with respect to R, then the eventualty property holds for H with respect to R.

3 PROBLEM FORMULATION

Consider a continuous-time nonlinear affine control system

\[
\dot{x} = f(x) + g(x)u, \quad x_0 = x(0),
\]

where \(x(t) \in \mathbb{R}^n\) is the system state, \(u(t) \in \mathbb{R}^m\) is the control input, \(f : \mathbb{R}^n → \mathbb{R}^n\) models the unknown system drift, and \(g : \mathbb{R}^n → \mathbb{R}^{n×m}\) captures the known control directions. The functions \(f\) and \(g\) are assumed to be locally Lipschitz with \(f(0) = 0\) and \(0 < ||g|| ≤ \bar{g}\). Let \(O\) be a finite set of observations where each observation \(o ∈ O\) corresponds to a region of interest (ROI) of the form

\[
R_o := \{x ∈ \mathbb{R}^n : h_o(x) ≥ 0\},
\]

for all \(o ∈ O\), where \(h_o : \mathbb{R}^n → \mathbb{R}\) is a continuously differentiable function.

Assumption 1. Each ROI \(R_o\) is non-empty and has no isolated points, i.e., \(\text{Int}(R_o) ≠ \emptyset\) and \(\text{cl}(\text{Int}(R_o)) = R_o\) for all \(o ∈ O\). Additionally, all ROI are disjoint in the sense that \(R_o ∩ R_{o'} = \emptyset, \forall o ≠ o'\).

In this paper, we consider the following problem:

Problem 1. Given a dynamical system (3) and an scLTL formula \(φ\) defined over a finite set of observations \(O\) labeling ROIs as in (4), find a control policy \(u\) such that the trajectory of (3) satisfies \(φ\).

To address Problem 1, we first translate the scLTL formula \(φ\) into an FSA \(A\) that accepts the language satisfying \(φ\). Then we compose the discrete behavior of \(A\) with the continuous behavior of (3) to form a hybrid system \(H\) as in (2). Given this formulation, Problem 1 becomes equivalent to certifying the eventualty property for \(H\) with respect to the set of accepting FSA states \(S_f\). To synthesize a correct control policy, we construct a sequence of optimal control problems (OCPs) corresponding to the sequence of FSA states in the accepting run \(w_2\) with the goal of driving the continuous trajectory of the plant through the ROIs so as to generate the accepting input word \(w_2\). As a main contribution of this paper, we show that the sequence of optimal value functions obtained from solving these OCPs in combination with a distance to acceptance function defined over the FSA forms a hybrid barrier certificate to establish correctness with respect to the given scLTL formula. This formulation is related to those seen in \[17, 35, 45\], in the sense that we translate an scLTL formula into a corresponding FSA and then...
use the FSA’s accepting run to construct a sequence of OCPs. The computational demands of solving these OCPs, as well as the uncertainty associated with the system drift \( f \), motivates us in Sec. 6 to introduce an approximate solution approach based on ADP and RL that does not require full knowledge of the system dynamics.

4 HYBRID SYSTEMS FORMULATION

To formalize Problem 1 we turn to the setting of hybrid systems as detailed in Sec. 2.3. To this end, consider the FSA \( A = (S, s_0, O, p, S_f) \) corresponding to a \( \text{scLTL} \) formula \( \phi \) over \( O \) as a discrete-time dynamical system of the form

\[
    s(j + 1) = p(s(j), o(j)),
\]

where \( j \in \mathbb{N}, s(j) \in S \) is the FSA state, and \( o(j) \in O \) acts as an input to (5). Following the approach in [6], we then combine the continuous-time dynamics in (3) with the discrete-time dynamics in (5) to obtain a hybrid system with state \( x_{H} \triangleq [x^T, s, o]^T \) of the form

\[
\begin{align}
    \dot{x}(t, j) &= f(x(t, j)) + g(x(t, j))u, \\
    \dot{s}(t, j) &= 0, \\
    \dot{o}(t, j) &= 0, \\
    x(t, j + 1) &= x(t, j), \\
    s(t, j + 1) &= p(s(t, j), o(t, j)), \\
    o(t, j + 1) &= v(t, j),
\end{align}
\]

Here, \( x(t, j) \) represents the current state of the plant, \( s(t, j) \) is the current state of the FSA, \( o(t, j) \) is the next observation that should appear in the accepting word of the FSA, and \( o(t, j) \) is a policy to be defined that selects the subsequent observation to be generated. Before explicitly defining \( C, \mathcal{D} \) we introduce the set \( O_\Delta \subseteq O \), which is indexed by \( s \) and is defined as

\[
    O_\Delta = \{ o \in O : p(s, o) \neq \emptyset \}, \quad \forall s \in S.
\]

In essence, \( O_\Delta \) is the set of all admissible observations for a given \( s \in S \), in the sense that, when \( o \in O_\Delta \) is input to the FSA in state \( s \) there exists an outgoing transition from \( s \) to \( p(s, o) \). Using (7) we define \( C \) and \( \mathcal{D} \) for (6) as

\[
\begin{align}
    C &= \{(x, s, o) : \exists x \in \text{cl}(\mathbb{R}^n \setminus \mathcal{R}_o), s \in S, o \in O_\Delta\}, \\
    \mathcal{D} &= \{(x, s, o) : x \in \mathcal{R}_o, s \in S, o \in O_\Delta\},
\end{align}
\]

which implies that for some \( o \in O \) jumps can only occur if \( x \in \mathcal{R}_o \); however, \( x \) can flow through \( \mathcal{R}_o \) for some \( o' \neq o \) without inducing a jump or generating observation \( o' \). An exception to this is when the \( \text{scLTL} \) formula specifies that a certain observation \( o' \) is not to be generated at a given \( s \in S \). In this situation we can impose active avoidance of \( \mathcal{R}_o \) by constructing a barrier function [2] for \( \mathcal{R}_o \) which is then incorporated into the subsequent control development outlined in Sec. 5. Given the above discussion, we now consider the more formal problem statement:

**Problem 2.** Given the hybrid system in (6), find a set of control policies \((u, o)\) such that the eventuality property for (6) holds with respect to \( \mathcal{R}_f \triangleq \{(x, s, o) : x \in \mathbb{R}^n, s \in S_f, o \in O_\Delta\} \). 

Our first step to solving Problem 2 is to specify the discrete-time policy \( v \), which is designed to select subsequent observations that produce an accepting word of the FSA. To this end, consider the FSA \( A \) as a digraph where each state \( s \in S \) represents a node and each observation \( o \in O \) labels an edge from \( s \) to \( p(s, o) \). The policy \( v \) is then selected so as to minimize the distance to acceptance (DTA) metric [15, 43], which, for some \( s \in S \), represents the minimum number of transitions required to reach the set of accepting states \( S_f \). Before defining the DTA, we define the distance between any two states in the FSA as [43]

\[
    d(s, s') \triangleq \begin{cases} 
    \min_{\varphi \in \Phi(s, s')} N(\varphi), & \text{if } \Upsilon(s, s') \neq \emptyset, \\
    \infty, & \text{if } \Upsilon(s, s') = \emptyset,
    \end{cases}
\]

where \( \Upsilon \) is the set of states between \( s \) and \( s' \), \( \varphi \) is a single path through \( \Upsilon \), and \( N(\varphi) \) is the number of states in \( \varphi \). The DTA can then be computed as

\[
    V_d(s) \triangleq \min_{s' \in S_f} d(s, s').
\]

Note that \( V_d \) is positive definite with respect to \( S_f \) (i.e., \( V_d(s) > 0, \forall s \in S_f \) and \( V_d(s) = 0, \forall s \in S_f \)) and radially unbounded in the sense that \( V_d(s) \to \infty \) as \( d(s, s') \to \infty, \forall s \in S_f \). Using the DTA, the discrete-time control policy \( v \) is selected as any element of \( O_{p(s, o)} \) such that the DTA strictly decreases across jumps as [6]

\[
    v(t, j) \in O_{p(s, o)}^d \triangleq \{ o' : v' \in O_{p(s, o)} \}.
\]

where \( V_d(p(s, o), v') < V_d(p(s, o)) \) if \( p(s, o) \in S_f \). (12)

where \( V_d(p(s, o)) \) is the DTA resulting from transitioning from the current state \( s \) to \( p(s, o) \) by inputting observation \( o \in O_s \) and \( V_d(p(s, o), v') \) is the DTA resulting from transitioning from \( p(s, o) \) to \( p(p(s, o), v') \) by inputting observation \( o' \in O_{p(s, o)} \). In the context of this paper, the DTA can be shown to satisfy the properties of a barrier certificate for a hybrid system consisting of the purely discrete behavior of the FSA with respect to \( S_f \) [7, Lemma 3]. Importantly, this implies that the sequence of observations selected by the policy in (12) produces an accepting word \( w_{\text{O}} \) for a given scLTL formula.

5 REACH-AVOID OPTIMAL CONTROL

We now transition to the problem of synthesizing a continuous control strategy \( u \) for the hybrid system such that the continuous trajectory of the plant \( x(t, j) \) produces an accepting word for the FSA. In this section, we show how this can be accomplished by associating with each \( j \in \mathbb{N} \) a suitably constructed optimal control problem. Importantly, we show that the value function of each optimal control problem can be combined with the DTA from the previous section to construct a barrier certificate for the hybrid system in (6) with respect to the set in (9).

5.1 System Transformation

Consider the current state \( x_{H}(t, j) \) of the hybrid system in (6) and recall that \( o(t, j) \) represents the observation that should be generated next so as to satisfy the given scLTL formula. For ease of exposition, we drop the explicit dependence on \( t, j \) unless otherwise needed for clarity. To ensure the state of the plant \( x \) can be regulated to the ROI \( \mathcal{R}_o \) corresponding to \( o \), we perform a system
transformation so that \( R_o \) contains an equilibrium point for (3). To this end, consider some point \( x_d \in \text{Int}(R_o) \) and define the regulation error \( e = x - x_d \). Additionally, define the invertible mapping \( z : \mathbb{R}^n \rightarrow \mathbb{R}^n \) as \( z(e) = e + x_d \), where \( z^{-1}(x) = x - x_d \). To ensure the system transformation can be performed we make the following assumption:

**Assumption 2** ([19]). The input dynamics matrix \( g(x) \) has full column rank and the function \( g^T(g(x)^{-1}g(x))^T \) is bounded and locally Lipschitz.

Following the formulation in [19], we decompose the control policy into two terms: a feedforward term \( u_d \in \mathbb{R}^m \) that ensures \( x \rightarrow x_d \) as \( u \rightarrow u_d + \mu \). The steady-state control policy corresponding to \( x_d \) is then \( u_d = -g^T(x_d)f(x_d) \) [19], which is a constant value for each \( o \in O \). With this steady-state policy, we define an auxiliary dynamical system as

\[
\dot{e} = \hat{x} = f(z(e)) + g(z(e)) u_d + g(z(e)) \mu,
\]

where \( F(e) = 0 \) and \( F(e) \) is locally Lipschitz.

### 5.2 Constructing the Cost Function

Given the system transformation in the previous subsection, the present goal is to determine a feedback control policy \( \mu : \mathbb{R}^n \rightarrow \mathbb{R}^m \) for the auxiliary system (13) such that \( e \rightarrow 0 \Rightarrow x \rightarrow R_o \). Simultaneously, for a given FSA state \( s \), this policy must ensure that any negated observations \( o \) are not generated (i.e., certain \( R_o \) must be avoided while in state \( s \)). To address this challenge, we associate with each state of the FSA a reach-avoid optimal control problem, where the objective is to find an optimal feedback policy \( \mu^* : \mathbb{R}^n \rightarrow \mathbb{R}^m \) that minimizes the following cost functional

\[
V(e, \mu) \triangleq \int_0^\infty r(e(\tau), \mu(\tau)) d\tau,
\]

where \( r : \mathbb{R}^n \times \mathbb{R}^m \rightarrow [0, \infty) \) denotes the instantaneous positive definite cost. To facilitate this approach, let \( \partial, O \) denote the set of all observations that are negated at the FSA state \( s \). Now, for each \( o \in \partial, \) consider the set \( X_o \triangleq \{ x \in \mathbb{R}^n : h_o(x) \leq 0 \} \) and associate with each \( X_o \) a barrier function [2] of the form \( b_o(x) = -1/h_o(x) \), which satisfies \( \inf_{x \in X_o} b_o(x) \geq 0 \) and \( \lim_{x \rightarrow \partial X_o} b_o(x) = \infty \). To ensure the optimal control problem is well-posed \(^5\) we then encode each \( b_o \), \( Yo \in \partial, \) within a positive definite centered barrier function (RBF) [34, 49] as

\[
B(e) \triangleq \sum_{o \in \partial, O} (b_o(z(e)) - b_o(x_d) - \nabla b_o(x_d)e)^2,
\]

which satisfies \( B(0) = 0, B(e) > 0, \forall e \neq 0, \) and \( B(e) \rightarrow \infty \) as \( z(e) \rightarrow \partial X_o \) for any \( o \in \partial, \). The instantaneous cost from (14) is then selected as

\[
r(e, \mu) = Q(e) + \mu^T R \mu + B(e),
\]

where \( Q : \mathbb{R}^n \rightarrow [0, \infty) \) is a positive definite and radially unbounded function, \( R \in \mathbb{R}^{m \times m} \) is a symmetric positive definite matrix, and \( B(e) \) is from (15). The optimal value function (cost-to-go)

\[
V^* : \mathbb{R}^n \rightarrow [0, \infty)
\]

is defined as

\[
V^*(e) \triangleq \inf_{\mu} \int_0^\infty r(e(\tau), \mu(\tau)) d\tau,
\]

and is assumed to be continuously differentiable \(^6\) on an open neighborhood of \( \text{cl}(C\setminus R_f) \). Assuming that an optimal feedback policy \( \mu^* : \mathbb{R}^n \rightarrow \mathbb{R}^m \) exists, the value function is characterized by the Hamilton-Jacobi-Bellman (HJB) equation

\[
0 = \min_{\mu} H(e, \nabla eV^*, \mu) = \nabla eV^*(e)(F(e) + G(e) \mu^*) + r(e, \mu^*),
\]

with a boundary condition of \( V^*(0) = 0 \), where \( H : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R} \) is the Hamiltonian. The optimal policy \( \mu^*(e) \) is obtained from the Hamiltonian minimization condition \( \partial H / \partial \mu = 0 \) as

\[
\mu^*(e) = -\frac{1}{2} R^{-1} G(e)^T \nabla eV^*(e)^T.
\]

### 5.3 Hybrid Barrier Certificate

The following proposition shows that for each \( j \in \mathbb{N} \), the optimal policy in (19) regulates \( x \rightarrow x_d \) while avoiding any ROI that should not be visited.

**Proposition 1.** If Assumptions 1-2 hold and there exists a positive definite and continuously differentiable function \( V^* \) satisfying (18), then the control policy in (19) renders the origin asymptotically stable for the auxiliary dynamical system (13). Furthermore, the set \( \text{Int}(X_o) \) is forward invariant.

**Proof.** The proof is based on the fact that the derivative of \( V^* \) along the trajectory of (13) can be upper bounded as \( \dot{V}^* \leq -\psi(|e|) \), where \( \psi \) is a class \( K \) function. It follows the lines of the proof of Prop. 1 in [10], and it is omitted. \( \square \)

The following theorem illustrates that the DTA from the previous section and the optimal value function from this section can be used to construct a barrier certificate to address Problem 2.

**Theorem 2.** Provided Assumptions 1-2 and the hypothesis of Proposition 1 hold, for some positive constant \( \lambda \in \mathbb{R}_{>0} \) the function

\[
V^*_B(z^{-1}(x), s, o) \triangleq V^*_B(s) + \lambda V^*(z^{-1}(x)),
\]

where \( V^*_B \) is the DTA from (11), is a barrier certificate for (6) with respect to (9).

**Proof.** Recall that a function \( V^*_B \) is a barrier certificate if it satisfies the criteria of Definition 4. The fact that the flow map of (6) is continuous on the flow set along with [7, Lemma 2] can be used to show that (6) satisfies the hybrid basic conditions [18, Assumption 8.6.5]. The continuity of \( V^*_B \) on \( \text{cl}(C \setminus R_f) \cup \text{cl}(\bar{D} \setminus R_f) \) follows from the continuity of \( V_d \) and \( V^* \) on \( \text{cl}(C \setminus R_f) \cup \text{cl}(\bar{D} \setminus R_f) \). Since we’ve assumed \( V^* \) is continuously differentiable on an open neighborhood of \( \text{cl}(C \setminus R_f) \), then \( V_B \) is continuously differentiable on an open neighborhood of \( \text{cl}(C \setminus R_f) \). Furthermore, \( V_d \) and \( V^* \) are both

\(^5\)Assuming continuous differentiability of the value function is standard throughout the literature (see [22] and references therein).
positive definite and radially unbounded, thus $V_B$ satisfies the first two conditions of Definition 4. For $(x, s, o) \in \text{cl}(C\backslash R_f)$ we have

$$\begin{align*}
\frac{dV_B}{dt} &= \lambda V^*(z^{-1}(x)) \left( F(z^{-1}(x)) + G(z^{-1}(x)) \mu^*(z^{-1}(x)) \right) \\
&= -\lambda \left( Q(z^{-1}(x)) + B(z^{-1}(x)) \right) \leq -\lambda \psi \|z^{-1}(x)\|,
\end{align*}$$

(21)

where the second equality follows by substituting in the HJB equation and the subsequent inequality from noting that $Q(z^{-1}(x)) + B(z^{-1}(x))$ is positive definite and can hence be lower bounded by a class $\mathcal{K}$ function $\psi : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ [23, Lemma 4.3]. Since $\psi(0) = 0$ is only attained at $x = x_d$ and $x_d \in \text{int}(R_u)$, then by the definition of $C$ in (8a), there exists some positive constant $\epsilon_6 \in \mathbb{R}_{>0}$ such that $\psi(z^{-1}(x)) > \epsilon_6$ for all $(x, s, o) \in \text{cl}(C\backslash R_f)$. Hence, the third condition of Definition 4 holds. For $(x, s, o) \in \text{cl}(D\backslash R_f)$ we have

$$\begin{align*}
V_B(x(t, j + 1), s(t, j + 1), o(t, j + 1)) - V_B(x(t, j), s(t, j), o(t, j)) &= (V_B(s(t, j + 1)) - V_B(s(t, j))) \\
&+ \lambda \left( V^*(z^{-1}(x(t, j + 1))) - V^*(z^{-1}(x(t, j))) \right). 
\end{align*}$$

(22)

Since $V_d$ takes only integer values and strictly decreases at jumps by (12) we have $V_d(s(t, j + 1)) - V_d(s(t, j)) \leq -1$. We must now distinguish between two cases: either $V^*(z^{-1}(x(t, j + 1))) - V^*(z^{-1}(x(t, j))) < 1$ or $V^*(z^{-1}(x(t, j + 1))) - V^*(z^{-1}(x(t, j))) \geq 1$. For the former, the final condition of Definition 4 is trivially satisfied by letting $\lambda = 1$. For the latter, pick $\lambda$ as any sufficiently small positive constant such that

$$\begin{align*}
V_d(s(t, j + 1)) - V_d(s(t, j)) > \lambda \left( V^*(z^{-1}(x(t, j + 1))) - V^*(z^{-1}(x(t, j))) \right). 
\end{align*}$$

(23)

By Proposition 1, $V^*$ is bounded for all $j \in \mathbb{N}$ and the existence of such a $\lambda$ is guaranteed. Hence the fourth condition of Definition 4 holds. Since $V_B$ satisfies the conditions of Definition 4, $V_B$ is a barrier certificate and Theorem 1 can be invoked to conclude that the eventuality property for (6) holds with respect to (9).

**Remark 1.** Even if $V^*(z^{-1}(x(t, j + 1))) - V^*(z^{-1}(x(t, j))) \Rightarrow 1$, $\lambda$ can always be taken to be some arbitrarily small positive constant such that the inequality in (23) holds, and hence the conditions of Definition 4 hold. Note that the actual value of $\lambda$ is not required for any part of implementing the method developed in this paper.

## 6 REINFORCEMENT LEARNING-BASED SOLUTION

To circumvent the computational demands of solving a sequence of OCPs and to alleviate the need to have full knowledge of the system dynamics, in this section we introduce an approximate solution approach based on ADP and RL (see [24, 29] for surveys of these approaches). Here, we take a model-based RL approach based on the work in [20, 21] wherein the value functions, optimal control policies, and system drift dynamics are all simultaneously learned online. We provide a proof of convergence of this approximation scheme and develop sufficient conditions under which formal guarantees of correctness with respect to the given specification are maintained.

### 6.1 System Identification

Using the universal function approximation property of neural networks (NNs), on a compact set $\mathcal{X} \subset \mathbb{R}^n$ containing the origin the unknown drift dynamics $f(x)$ can be represented as $f(x) = \theta^T Y(x) + e_\theta(x)$ where $Y : \mathbb{R}^n \rightarrow \mathbb{R}^{p_1}$ is a vector of user-defined basis function, $\theta \in \mathbb{R}^{p_1 \times n}$ are the unknown ideal NN weights, and $e_\theta : \mathbb{R}^n \rightarrow \mathbb{R}^n$ are the unknown function reconstruction errors$^6$. Given the NN representation of $f$, the unknown NN weights are replaced with estimates $\hat{\theta} \in \mathbb{R}^{p_1 \times n}$ to yield the estimated dynamics $f(x, \hat{\theta}) \approx \hat{\theta}^T Y(x)$. To update the weights so as to minimize the approximation error $\theta - \hat{\theta}$, we take a concurrent learning [8, 36] approach. In this setting, input-output data is recorded in a history stack that is used to update the weights in real-time. Provided the recorded data is rich enough, convergence of the weight estimates to a neighborhood their ideal values can be established. Specifically, we leverage an integral concurrent learning (ICL) approach [36], which alleviates the need to measure (or compute) state derivatives, and select the update law as [12]

$$\dot{\hat{\theta}}(t) = \text{proj} \left\{ k_0 \Gamma_{\theta}(t) \sum_{i=1}^{M} \Gamma_i \left[ x(t_i) - (x(t_i - \Delta t) - \mathcal{U}_i - \hat{\theta}(t)^T Y_i) \right] \right\}$$

$$\Gamma_{\theta}(t) = \beta_0 \Gamma_{\theta}(t) - k_0 \Gamma_{\theta}(t) \left( \sum_{i=1}^{M} \gamma_i Y_i^T \right) \Gamma_{\theta}(t),$$

(24a)

(24b)

where $k_0 \in \mathbb{R}_{>0}$ is a user-defined gain, $\beta_0 \in \mathbb{R}_{>0}$ is a forgetting factor, and $\text{proj} \{ \cdot \}$ is a smooth projection operator that bounds the weight estimates [16, §4.4]. In (24) $M \in \mathbb{N}$ is the size of the history stack, $t_i \in [0, t]$ is some previous time, and the terms $Y_i \in \mathbb{R}^{n \times n}$ and $\mathcal{U}_i \in \mathbb{R}^{n \times n}$ are defined as $Y_i(t) \triangleq \int_{t_0}^{t_i} \frac{\partial f}{\partial x} \left( \int_{t_0}^{\tau} \frac{\partial f}{\partial x} \right) d\tau$ and $\mathcal{U}_i(t) \triangleq \int_{t_0}^{t_i} \frac{\partial f}{\partial x} \left( \int_{t_0}^{\tau} \frac{\partial f}{\partial x} \right) d\tau$. The integration window. To ensure identification of the unknown dynamics, the history stack is required to be sufficiently rich.

**Assumption 3** ([12, 36]). There exist constants $T_0, \lambda_0 \in \mathbb{R}_{>0}$ such that for all $t \geq T_0$, $\lambda_0 \min \left\{ \sum_{i=1}^{M} \gamma_i \right\} \geq \lambda_0$.

Provided the above assumption is satisfied and $\lambda_{\text{min}} \Gamma_{\theta}(0) > 0$, $\Gamma_{\theta}(t)$ can be shown to satisfy $\Gamma_{\theta}(t) \leq \Gamma_{\theta}(0) \leq \Gamma_{\theta}(0), \forall t \in [0, t]$ where $\Gamma_{\theta}(t) = \mathbb{R}_{>0}$ and $\mathcal{U}_i$ denotes a $p_1 \times p_1$ identity matrix. Convergence of the weight estimates to a neighborhood of their ideal values can be established using $\Gamma_{\theta}(t) \leq \mathcal{U}(t)$ as a Lyapunov function, which can be shown to satisfy $\dot{\Gamma}_{\theta} \leq -k_0 \| \hat{\theta} \|^2 + D_{\theta} \| \hat{\theta} \|$ where $k_0 \in \mathbb{R}_{>0}$ is a constant that depends on the gains and $D_{\theta} \in \mathbb{R}_{>0}$ is a residual bound that depends on $\epsilon_\theta$ [12, Thm. 1].

$^6$ Similar to works such as [19, 20] it is assumed that $Y$ provides a proper basis for function approximation and hence the NN satisfies $\| \theta \| < \theta < \infty, \sup_{x \in \mathcal{X}} \| e_\theta(x) \| \leq \epsilon_\theta, \sup_{x \in \mathcal{X}} \| e_{\text{eq}}(x) \| \leq \epsilon_{\text{eq}}$, where $\theta, \epsilon_\theta, \epsilon_{\text{eq}} \in \mathbb{R}_{>0}$ are positive constants. See works such as [20] for additional details.
6.2 Value Function Approximation

To estimate the value function online we leverage computationally efficient state-following kernel (StaF) functions introduced in [20, 38], which provide a local representation of the value function within a small compact set that travels with the state of the system. To facilitate this approach let \( \chi \subseteq \mathbb{R}^n \) be a compact set with \( e \in \text{Int}(\chi) \) and let \( B_d(e) \subseteq \chi \) be a closed ball of radius \( d \in \mathbb{R}_{>0} \) centered at \( e \). Furthermore, let \( \hat{B} : \mathbb{R}^n \rightarrow \mathbb{R}_{\geq 0} \) be a bounded positive definite function defined as \( \hat{B}(y) \equiv \frac{B_d(y)}{1 + B_d(y)} \). Then, after adding and subtracting \( \hat{B}(y) \), the value function can be represented at points \( y \in B_d(e) \) as [20, 38]

\[
V^*(y) = W(e)^T \sigma(y, c(e)) + \epsilon(e, y) + \hat{B}(y),
\]

where \( W : \chi \rightarrow \mathbb{R}^n \) is the continuously differentiable ideal weight function, \( \sigma : \chi \times \mathbb{Y} \rightarrow \mathbb{R}^n \) is a vector of continuously differentiable kernel functions, \( \epsilon \equiv \{ c_1(e), \ldots, c_L(e) \}^T \) is a set of distinct centers, and \( e : \chi \times Y \rightarrow \mathbb{R} \) is the continuously differentiable function reconstruction error. Since the ideal weight function is unknown, it is replaced with estimates \( \hat{W}_c, \hat{W}_d : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^L \) yielding the estimated value function and control policy as

\[
\hat{V}(y, e, \hat{W}_c) = \hat{W}_c^T \sigma(y, c(e)) + \hat{B}(y) \tag{26a}
\]

\[
\hat{\mu}(y, e, \hat{W}_d) = -\frac{1}{2} \hat{R}^{-1} \hat{G}(y)^T (\hat{V}_y \sigma(y, c(e)) + \hat{G}(y) \hat{\mu}(y, e, \hat{W}_d)) \tag{26b}
\]

The objective is then to determine values for the estimated weights \( \hat{W}_c(t), \hat{W}_d(t) \) such that \( \hat{V} \) and \( \hat{\mu} \) closely approximate \( V^* \) and \( \mu^* \). To this end, the approximations from (26a) and (26b), as well as the estimated dynamics \( \hat{f} = \hat{\theta}^T Z(e) \), are substituted into the HJB equation (18) to yield the Bellman error (BE) \( \delta \)

\[
\delta(y, e, \hat{W}_c, \hat{W}_d, \hat{\theta}) \equiv r(y, \hat{\mu}(y, e, \hat{W}_d)) + \nabla_y \hat{V}(y, e, \hat{W}_c)(\hat{f}(y, \hat{\theta}) + \hat{G}(y) \hat{\mu}(y, e, \hat{W}_d)), \tag{27}
\]

where \( \hat{f}(y, \hat{\theta}) \equiv \hat{\theta}^T Z(y) - g(z(y)) \sigma_1(x_d) \hat{\theta}^T Y(x_d) \). Since the HJB equation in (18) is equal to zero under optimal conditions, the goal of the subsequent learning strategy is to update \( \hat{W}_c(t), \hat{W}_d(t) \) so as to drive \( \delta \) to zero.

6.3 Weight Update Laws

To update the weight estimates, at every time \( t \in \mathbb{R}_{\geq 0} \) the BE is evaluated as \( \delta_t \equiv \delta(e, \epsilon, \hat{W}_c(t), \hat{W}_d(t), \hat{\theta}) \), which implies that the control policy that influences the auxiliary dynamics in (13) is given by

\[
\mu = \hat{\mu}(e, \hat{W}_d),
\]

where \( \hat{\mu} \) is defined as in (26b). Hence, the controller that influences the original system in (3) is

\[
u = -g^T(x_d) \hat{\theta}^T Y(x) + \mu,
\]

where \( \mu \) is given by (28). To relax the restrictive persistence of excitation (PE) condition that is traditionally required to identify the value function, at every time \( t \in \mathbb{R}_{\geq 0} \) off-policy trajectories are sampled using the family of functions \( \{ e_i : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R} \} i \geq 1 \), where each function \( e_i(t, \epsilon) \in B_d(e(t)) \) defines a mapping from the current state \( e \) to an unexplored point in \( B_d(e(t)) \). The BE is then evaluated at each sampled point as \( \delta_{e_i} \equiv \delta(e_i, e, \hat{W}_c(t), \hat{W}_d(t), \hat{\theta}) \). Using both \( \delta_t \) and \( \delta_{e_i} \) the value function weights are then updated using a recursive least-squares update law with a forgetting factor as

\[
\dot{\hat{W}}_c = -k_{c1} \frac{\hat{\theta}^T}{\rho^2} \delta_t - \frac{k_{c2}}{N} \sum_{i=1}^{N} \frac{\hat{\theta}^T}{\rho_i^2} \delta_{e_i},
\]

where \( \Gamma(t) \in \mathbb{R}^{L \times L} \) is a dynamic gain matrix, \( \rho(t) \equiv 1 + \gamma_1 \omega(t)^T \omega(t) \) is a normalization term, \( k_{c1}, k_{c2}, \gamma_1 \in \mathbb{R}_{>0} \) are positive constant gains and \( \omega(t) \) is a regressor vector defined as

\[
\omega(t) \equiv \nabla_{e_i} \sigma(e(t), e(c(e))) \left( \hat{f}(e_i, \hat{\theta}(t)) + G(e(t)) \mu(t) \right).
\]

The condition in (34c) asks for the regressor \( \omega \), which depends on the trajectory of the system, to be PE which is a strong condition that is notoriously difficult to verify online. On the other hand, the conditions in (34a) and (34b) are placed on \( \omega_i \), which depends on the sampled trajectories and can be verified online [13]. Hence, one can select the family of functions \( \{ e_i : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R} \} i \geq 1 \), so that \( \omega_i \) satisfies (34a) or (34b) by either sampling highly oscillatory extrapolated trajectories, for (34a), or by sampling a large

---

All terms with the subscript \( i \) denote that a function is evaluated at the sampled state \( e_i \). For example, \( \omega_i \equiv \nabla_{e_i} \sigma(e_i, c(e)) \left( \hat{f}(e_i, \hat{\theta}(t)) + G(e(t)) \mu_i(e_i, e, \hat{W}_d) \right) \).
number of extrapolated trajectories, for (34b). Indeed, multiple results [11, 13, 20] have demonstrated that Assumption 4 can be satisfied with as few as one time-varying extrapolation point. Moreover, if Assumption 4 is satisfied and $\lambda_{\text{min}}(\Gamma^{-1}(0)) > 0$, the update law in (32) can be used to show that the matrix $\Gamma$ satisfies $\Gamma H \leq \Gamma$ for all $t \geq 0$, where $\Gamma, \Gamma \in \mathbb{R}_{>0}$ [20, Lemma 1].

Now consider the following Lyapunov function candidate

$$V_L(Z_L, t) = V^*(e) + \frac{1}{2} \bar{\gamma}^2 \Gamma^{-1}(t) \bar{W}_c + \frac{1}{2} \bar{W}_a + V_0(\bar{e}, t).$$

(35)

where $Z_L \doteq \{e^T, \bar{W}_c, \bar{W}_a, \bar{\gamma}^2 \}^T$ is a composite state vector, $W_c \doteq W - W_c, \bar{W}_a \doteq W - W_c$ denote the weight estimation errors, and $V_0$ is from Sec. 6.1. Since $V^*$ is positive definite, $V_L$ is positive definite and can hence be bounded as $p_1(\|Z_L\|) \leq V_L(Z_L, t) \leq \bar{\gamma}(\|Z_L\|)$ where $\bar{\gamma}, \bar{\gamma} : \mathbb{R}_{>0} \rightarrow \mathbb{R}_{>0}$ are class $\mathcal{K}$ functions [23, Lemma 4.3]. Moreover, let $B_Z \subset \mathbb{R}^{2 \times n_p}$ be a closed ball of radius $\bar{\gamma}$ in $\mathbb{R}_{>0}$ centered at the origin and define the operator $\|\cdot\| \doteq \sup_{x \in B_Z} \|x\|$. The following proposition illustrates that the developed learning strategy guarantees uniformly ultimately bounded convergence to the optimal solution.

**Proposition 2.** Provided Assumptions 2-4 hold and the following sufficient conditions are satisfied

$$\mathcal{L} > \frac{\varphi_{\theta} + \varphi_{\phi}}{k_2},$$

$$k_{a1} + k_{a2} > 2\varphi_u + \varphi_{ac},$$

$$\bar{\gamma}_1(1) > \tilde{\gamma}_1(1),$$

where $\mathcal{L} > \frac{\varphi_{\theta} + \varphi_{\phi}}{k_2}$, and $\varphi_u, \varphi_{ac}, \varphi_{\phi}$, $\tilde{\gamma}_1 : \mathbb{R}_{>0} \rightarrow \mathbb{R}_{>0}$ are positive constants and $\tilde{\gamma}_1 : \mathbb{R}_{>0} \rightarrow \mathbb{R}_{>0}$ is a class $\mathcal{K}$ function developed in the subsequent proof, then the control policy in (28) and the update laws in (30), (32), (33) guarantee that $Z_L(t)$ is uniformly ultimately bounded in the sense that $\limsup_{t \rightarrow \infty} \|Z_L(t)\| \leq \tilde{\gamma}_1(1)$.

Moreover, the set Int$(\mathbb{R}_{>0})$ is forward invariant.

By the definition of uniform ultimate boundedness [23, Def. 4.6], the above proposition implies that $t$ reaches some neighborhood of $x_d$ in finite time, the size of which depends on the parameters associated with the learning scheme (see Remark 2).

**Proof.** The derivative of $V_L$ along the trajectory of the auxiliary system in (13) is

$$\dot{V}_L = V^* + \bar{W}_c^2 \Gamma^{-1}(W - \bar{W}_c) + \bar{W}_a^2 (W - \bar{W}_a) - k_{\phi} + \bar{\gamma}_1^2(1).$$

(37)

Using the definition of the weight estimation errors, the BE can be expressed as $\Delta_t = \bar{\gamma}(e^T \bar{W}_c + \frac{1}{2} \bar{W}_a, \bar{\gamma}_1^2(1), \varphi_u, \varphi_{ac}, \varphi_{\phi}, k_{a1}, k_{a2})$ and $A$ and $\Delta_t$ are of size that are uniformly bounded over $\chi$. Using $W = \bar{W}_c$, $F$ and $G_\phi$, the HJB (18), substituting in the update laws (30), (32), (33), and upper bounding yields

$$\dot{V}_L \leq -\gamma(e^T) - \frac{\gamma(e^T)}{2} \bar{W}_c^2 + \frac{1}{2} \bar{W}_a^2 + k_{\phi} - \bar{\gamma}_1^2(1).$$

(38)

where $\varphi_u, \varphi_{\phi}, \varphi_{ac}, k_{a1}, k_{a2} \in \mathbb{R}_{>0}$ are positive constants.

Provided the gains are selected according to (36), applying Young’s inequality, segregating terms, and completing the square allows (38) to be further bounded as

$$\dot{V}_L \leq -\gamma(e^T) - \frac{\gamma(e^T)}{2} \bar{W}_c^2 + \frac{1}{2} \bar{W}_a^2 + k_{\phi} - \bar{\gamma}_1^2(1).$$

(39)

which implies that $V_L$ is bounded and decreasing outside of a small compact set centered at the origin such that

$$\dot{V}_L \leq -\gamma(e^T) \quad \forall \|Z_L\| > \gamma^{-1},$$

(40)

Moreover, selecting the update laws (30), (32), (33), and upper bounding

$$\dot{V}_L \leq -\gamma(e^T) \quad \forall \|Z_L\| > \gamma^{-1},$$

(41)

(42)

The bound in (40) and the fact that $V_L$ is bounded as $\mathcal{K}$ $(\|Z_L\|)$ can be used with [23, Thm. 4.18] to conclude that $Z_L$ is uniformly ultimately bounded. Similarly to Prop. 1, the boundedness of $V_L$ implies that system trajectories cannot reach any ROI that are negated at the current FSA state. Since $Z_L \in \mathbb{R}_{>0}$, then $W, \bar{W}_c$ and thus $W \in \mathbb{R}_{>0}$, which implies that $W, \bar{W}_c$ and thus $\mu \in \mathbb{R}_{>0}$. □

**Remark 2.** The sufficient conditions from (36) can be satisfied by increasing the number of basis functions for value function approximation $L$, which reduces the residual error $\varepsilon$, and by sampling off-policy trajectories that result in large values of $\mathcal{L}$. Similarly, increasing the number of basis function used to estimate the unknown dynamics $p_1$ and size of the history stack $M$ will help satisfy (36). Additionally, selecting $R$ with a large minimum eigenvalue and increasing $k_{a2}$ and $\gamma$ can help satisfy (36).

Now consider an augmented hybrid system $\tilde{H} = (\tilde{F}, \tilde{G}, \tilde{C}, \tilde{D})$ with state $x_{\tilde{H}} = \{x^T, \tilde{W}_c, \tilde{W}_a, \bar{\gamma}^2 \}^T$ that captures the dynamics of the plant, automaton, and learning strategy, where

$$\tilde{F}(x_{\tilde{H}}) \doteq \begin{bmatrix} x^T \\ \tilde{W}_c^T \tilde{W}_a^T \tilde{\gamma}^2 \end{bmatrix},$$

$$\tilde{G}(x_{\tilde{H}}) \doteq \begin{bmatrix} x^T \\ \tilde{W}_c^T \tilde{W}_a^T \tilde{\gamma}^2 \end{bmatrix},$$

$$\tilde{C}(x_{\tilde{H}}, \tilde{W}_c, \tilde{W}_a, s, o) \doteq \begin{bmatrix} r(s, o) \end{bmatrix},$$

(43)

$$\tilde{D} \doteq \{ (x, \tilde{W}_c, \tilde{W}_a, s, o) : x \in \text{s}cl(\mathbb{R}^n \setminus \mathcal{R}_0), \tilde{W}_c \in \mathbb{R}^L, \tilde{W}_a \in \mathbb{R}^L \},$$

(43c)

The constants used in the proof of Proposition 2 are defined as follows:

$$\varphi_{ac} \doteq \frac{\gamma(e^T)}{2} \bar{W}_c^2 + \frac{1}{2} \bar{W}_a^2 + k_{\phi} - \bar{\gamma}_1^2(1).$$

(43)

$$\varphi_{\phi} \doteq \frac{\gamma(e^T)}{2} \bar{W}_c^2 + \frac{1}{2} \bar{W}_a^2 + k_{\phi} - \bar{\gamma}_1^2(1).$$

(43)

$$\varphi_u \doteq k_{a1} + k_{a2} + \frac{\gamma(e^T)}{2} \bar{W}_c^2 + \frac{1}{2} \bar{W}_a^2 + k_{\phi} - \bar{\gamma}_1^2(1).$$

(43)
Remark 3. The condition that \( \tau \) is sufficiently small is somewhat restrictive as it depends on terms such as the residual value function approximation error \( \epsilon \), which are unknown and infeasible to compute in general [20, Footnote 6]. Despite the fact that it may be difficult to obtain a reasonable estimate of \( \tau \), the bound \( \gamma^{-1}(\tau) \) is a class \( \mathcal{K} \) function of \( \tau \) and hence decreases with decreasing \( \tau \). Based on the definition of \( \tau \) in (42), the bound can be decreased by selecting the parameters for the learning scheme as outlined in Remark 2. Similar to previous works that have leveraged the model-based RL framework used in this paper [10, 11, 13, 14, 20], we'll demonstrate numerically in the following section that a sufficient approximation can generally be achieved using as few as \( L = n + 1 \) basis functions for value function approximation.

7 NUMERICAL EXAMPLES

We consider a system as in [22, Ch. 5.2], [13, §VI-C] with \( x = [x_1, x_2]^T \) and

\[
f(x) = \begin{bmatrix} -x_1 + \frac{1}{2} x_2 \\ -\frac{1}{2} x_1 - \frac{1}{2} x_2 (1 - \cos(2x_1) + 2) \end{bmatrix}
\]  

(46)

\( \dot{\mathcal{D}} \triangleq \{(x, \widehat{W}_c, \widetilde{W}_d, \widetilde{\theta}, s, o) : x \in \mathbb{R}^n, \widehat{W}_c \in \mathbb{R}^L, \widetilde{W}_d \in \mathbb{R}^L, \widetilde{\theta} \in \mathbb{R}^{P \times n}, s \in S, o \in O \} \). 

(43d)

In (43b), the terms \( \widehat{W}_c^*, \widetilde{W}_d^* \) denote the jump in the weight estimation errors when switching from one OCP to the next as the ideal weights corresponding to the subsequent optimal value function and control policy may be different from the previous OCP. The following theorem shows that if the ultimate bound in Proposition 2 is sufficiently small, then the Lyapunov function \( V_d \) and the DTA \( V_d \) from Sec. 4 can be used construct a barrier certificate for (43).

**Theorem 3.** Suppose Assumptions 1-4 and the hypothesis of Proposition 2 hold and the positive constant \( \kappa \) from (42) is sufficiently small. Then, for some \( \lambda \in \mathbb{R}_{>0} \) the function

\[
\dot{V}_B \triangleq V_d(s) + \lambda V_L(Z_L, i),
\]  

(44)

where \( V_d \) is from (11) and \( V_L \) is from (35), is a barrier certificate for (43) with respect to

\[
\mathcal{R}_f \triangleq \{(x, \widehat{W}_c, \widetilde{W}_d, \widetilde{\theta}, s, o) : x \in \mathbb{R}^n, \widehat{W}_c \in \mathbb{R}^L, \widetilde{W}_d \in \mathbb{R}^L, \widetilde{\theta} \in \mathbb{R}^{P \times n}, s \in S_f, o \in O \}.
\]  

(45)

Proof sketch. The proof follows the same steps as that of Theorem 2; however, more attention must be given to the third condition of Definition 4. Following the same steps as in Theorem 2, for \( (x, \widehat{W}_c, \widetilde{W}_d, \widetilde{\theta}, s, o) \in \mathcal{C}(\mathcal{C}(\mathcal{R}_f)) \) we have \( \nabla V_B \mathcal{F} \leq -\lambda_0 \| z_L \| \) as long as \( \| z_L \| > \gamma^{-1}(\tau) \), where \( \gamma \) is the class \( \mathcal{K} \) function and \( \tau \) is the constant defined in (41) and (42), respectively. Hence, the third condition of Definition 4 holds provided the bound \( \gamma^{-1}(\tau) \) is sufficiently small, which can be achieved by appropriately selecting the parameters associated with the learning scheme (see Remark 3). Verifying the first, second, and fourth condition of Definition 4 follows closely to that of Theorem 2 and is omitted. Given that \( V_B \) can be shown to satisfy the properties of a barrier certificate outlined in Definition 4, Theorem 1 can be invoked to show that the eventuality property holds for \( \mathcal{H} \) with respect to \( \mathcal{R} \).

Figure 1: FSA corresponding to the scLTL formula in (48). The double circle denotes the final state. A transition labeled by a Boolean logic formula stands for a set of transitions, each labeled by a singleton, e.g., the self transition labeled by \( \neg o_1 \wedge \neg o_5 \) at \( s_1 \) represents 3 self transitions labeled by \( o_1, o_2 \), and \( o_3 \), respectively.

\[
g(x) = \begin{bmatrix} \sin(2x_1) + 2 \cos(2x_1) + 2 \end{bmatrix}.
\]  

(47)

To define a specification for the system, we consider non-overlapping ROI of the form \( \mathcal{R}_b \triangleq \{x \in \mathbb{R}^2 : \| x - x_{di} \| \leq r_i \} \), where \( x_{di} \in \mathbb{R}^2 \) and \( r_i \in \mathbb{R}_{>0} \) denote the center and radius of the \( i \)th ROI, respectively, and \( h_0(x) = r_i - \| x - x_{di} \| \), where \( i \in \{1, \ldots, 5\} \). Using these ROI we construct a specification as an scLTL formula of the form [17, §3.2]

\[
\phi = \bigcirc (\bigcirc (o_1 \wedge o_2) \circ (o_2 \wedge o_3) \circ (o_3 \wedge o_4) \circ (o_4 \wedge o_5) \circ (\neg o_4 \wedge \neg o_5))
\]  

(48)

which states "visit \( \mathcal{R}_b \) then \( \mathcal{R}_b \) or visit \( \mathcal{R}_b \) then \( \mathcal{R}_b \) and eventually visit \( \mathcal{R}_b \) and don’t visit \( \mathcal{R}_b \) until visiting \( \mathcal{R}_b \)." The FSA corresponding to the scLTL formula in (48) is provided in Fig. 1. Using the DTA from Sec. 4 we compute \( w_{o_1} = o_1 \neg o_2 o_3 \) and \( w_{o_2} = o_2 o_3 \) as accepting words, which can be used to construct a sequence of optimal control problems as outlined in Section 5.

For each optimal control problem we select the cost function as in (16) with \( Q = \| e \|^2 \) and \( R = 1 \). To ensure the trajectory never reaches \( \mathcal{R}_{o_1} \) or \( \mathcal{R}_{o_5} \) we construct a RBF as in (15) with \( b_{o_1}(x) = h_{o_1}(x) / h_{o_1}(x) \) and then scale the RBF as \( 0.001 \times B(x) \). To perform system identification we parameterize the drift dynamics as \( f(x) = \theta^T Y(x) + \epsilon \phi(x) \) where the vector of basis functions is selected as \( Y(x) = [x_1, x_2, x_2(1 - \cos(2x_1) + 2)^2] \) and thus the ideal weights are

\[
\theta = \begin{bmatrix} \theta_1 & \theta_4 \\ \theta_2 & \theta_5 \\ \theta_3 & \theta_6 \end{bmatrix}, \quad \theta_1 = -1, \theta_2 = 1, \theta_3 = 0, \theta_4, \theta_5 = -\frac{1}{2}.
\]  

(46)

Since an exact basis is used to represent \( f(x) \), \( \epsilon \phi(x) = 0 \) and the weight estimates can be compared to their true values. The unknown dynamics are learned by using a pre-populated history stack, which is then used to estimate the unknown weights online using the update laws in (24) with \( h_{o_1} = 15, \beta_6 = 10 \), and \( \Gamma_0(0) = 20I_3 \) where the initial weight estimates \( \theta(0) \) are drawn
randomly from a uniform distribution between -5 and 5. The parameters for value function approximation are selected to be the same as in [14, §VII-A], where \( T(0) = I_5 \) and the initial weights for the value function \( \hat{W}_0(0) \) and control policy \( \hat{W}_u(0) \) are drawn from a uniform distribution between 0 and 1. The closed-loop system with an initial condition of \( x(0) = [-2, 2]^T \) is simulated for both accepting words in MATLAB R2019b for 10 seconds using Euler integration with a step size of 0.002 seconds on a 2019 MacBook Pro with a 1.4 GHz Quad-Core Intel i5 processor and 8 GB of memory. The total computation time for the simulations are 1.6458 seconds and 1.7844 seconds, respectively.

Results. The results of the simulation are illustrated in Fig. 2-3. In each figure the solid lines correspond to the simulation for \( \omega_2 = o_1 o_2 o_3 \) whereas the simulation corresponding to \( \omega_2 = o_2 o_1 o_3 \) is represented by dashed lines. In Fig. 2 the yellow, purple, and green disks denote \( R_{v_1}, R_{v_2}, R_{v_3} \), respectively, and the red disks denote \( R_{v_4}, R_{v_5} \) where the initial condition is represented as a blue circle and the terminal condition is represented as a blue star. The trajectory for both runs can be seen to successfully satisfy the scLTL specification with visiting the ROI in the correct order without ever entering \( R_{v_4}, R_{v_5} \). The evolution of the FSA state over time is provided in Fig. 3 (top right), where the accepting words \( \omega_2 = o_1 o_2 o_3 \) and \( \omega_2 = o_2 o_1 o_3 \) are produced in around 4 seconds for each simulation. Fig. 3 (bottom right) illustrates the evolution of the estimated value function and control policy weights. Because the optimal value function and policy are unknown the estimated weights cannot be compared to their true values. Moreover, this implies that the explicit evolution of the barrier certificate \( \hat{V}_B \) is unknown since \( V_L \) contains unknown terms. The evolution of the system identification weight estimates are shown in Fig. 3 (bottom left), where the estimates are shown to rapidly converge to their true values. Note that the history stack was populated with pre-recorded data; however, the weights are updated online during run time. The use of a pre-recorded history stack is made for ease of implementation and to improve computational efficiency. As previously mentioned, this data can be recorded online using algorithms such as those mentioned in [9] at the cost of additional computation.

8 CONCLUSION AND FUTURE WORK

We presented a hybrid systems approach to synthesizing optimal control policies for uncertain systems from scLTL formulas. By adopting tools from hybrid systems and optimal control theory, we provided Lyapunov-like conditions for scLTL formula satisfaction and showed that the value function in conjunction with a DTA metric forms a barrier certificate to provide guarantees of scLTL satisfaction. To circumvent the computational demands of solving a sequence of HJB equations, we take a learning-based approach to approximately solve a sequence of reach-avoid optimal control problems online without requiring full knowledge of the system dynamics. Furthermore, we developed sufficient conditions for which this approximate solution approach maintains formal guarantees and supported these claims with a numerical example.

A limitation of our approach that will be explored in future work is the inability to formally verify if the sufficient conditions of Theorem 3 are satisfied for a given set of learning parameters. This technical challenge stems from the uniformly ultimately bounded convergence results of Proposition 2, hence establishing an asymptotic convergence result, as in [48], may provide a pathway towards eliminating this complication. Despite this, we illustrated through numerical examples that this condition is not overly restrictive in practice. Similar to most Lyapunov-based approaches, the conditions for each result are only sufficient and thus our approach is not complete; however, our approach is sound in the...
sense that satisfaction of the sufficient conditions implies satisfaction of the given specification. Directions for future research include extensions to generic LTL formulas and decentralized systems where distributed teams of agents must act locally to satisfy global objectives.

9 ACKNOWLEDGEMENTS

This work was partially supported by the NSF under grants IIS-1723995, IIS-2024606, and GRFP DGE-1840990. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF. We thank the anonymous reviewers for their helpful comments and suggestions which have improved the quality of this paper.

REFERENCES

[1] D. Askaray, A. Jones, Z. Kong, M. Schwager, and C. Belta. 2016. Q-learning for robust satisfaction of signal temporal logic specifications. In Proc. Conf. Decis. Control. 6565–6570.

[2] A. D. Ames, X. Xue, J. W. Grizzle, and P. Tabuada. 2017. Control barrier function based quadratic programs for safety critical systems. IEEE Trans. Autom. Control 62, 8 (2017), 3861–3876.

[3] C. Baier and J. P. Katoen. 2008. Principles of model checking. MIT Press.

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

[5] C. Bela, B. Yooyanov, and E. A. Gol. 2017. Formal methods for discrete-time dynamical systems. Springer.

[6] A. Bisoffi and D. V. Dimarogonas. 2018. A hybrid barrier certificate approach to satisfy linear temporal logic specifications. In Proc. Amer. Control Conf. 634–639.

[7] A. Bisoffi and D. V. Dimarogonas. 2020. Satisfaction of linear temporal logic specifications through recurrence tools for hybrid systems. IEEE Trans. Autom. Control (2020).

[8] G. Chowdhary. 2010. Concurrent learning for convergence in adaptive control without persistence of excitation. Ph.D. Dissertation. Georgia Institute of Technology, Atlanta, GA.

[9] G. Chowdhary and E. Johnson. 2011. A singular value maximizing data recording algorithm for concurrent learning. In Proc. Amer. Control Conf. 3547–3552.

[10] M. H. Cohen and C. Belta. 2020. Approximate Optimal Control for Safety-Critical Systems with Control Barrier Functions. In Proc. Conf. Decis. Control. 2062–2067.

[11] P. Deptula, Z. I. Bell, E. A. Doucette, J. W. Curtis, and W. E. Dixon. 2020. Data-based reinforcement learning approximate optimal control for an uncertain nonlinear system with control effectiveness faults. Automatica 116 (2020).

[12] P. Deptula, Z. I. Bell, M. F. Zegera, R. A. Lietzra, and W. E. Dixon. 2018. Single agent indirect herding via approximate dynamic programming. In Proc. Conf. Decis. Control. 7136–7141.

[13] P. Deptula, H.Y. Chen, R. Lietzra, J. A. Rosenfeld, and W. E. Dixon. 2020. Approximate optimal motion planning to avoid unknown moving avoidance regions. IEEE Trans. Robot. 32, 2 (2020), 414–430.

[14] P. Deptula, J. A. Rosenfeld, R. Kamalapurkar, and W. E. Dixon. 2018. Approximate dynamic programming: combining regional and local state following approximations. IEEE Trans. Neural Netw. Learn. Syst. 29, 6 (2018), 2154–2166.

[15] X. Ding, M. Lazar, and C. Bela. 2014. LTL receding horizon control for finite deterministic systems. Automatica 50 (2014), 399–408.

[16] W. E. Dixon, A. Behal, D. M. Dawson, and S. Nagarkatti. 2003. Nonlinear Control of Engineering Systems: A Lyapunov-Based Approach. Birkhauser: Boston.

[17] J. Fu, I. Papusha, and U. Topcu. 2017. Sampling-based Approximate Optimal Control Under Temporal Logic Constraints. In Proceedings of the 46th International Conference on Hybrid Systems: Computation and Control. 227–235.

[18] R. Goebel, R. G. Sanfelice, and A. R. Teel. 2012. Hybrid dynamical systems: modeling, stability, robustness. Princeton University Press.

[19] R. Kamalapurkar, L. Andrews, P. Walters, and W. E. Dixon. 2016. Model-based reinforcement learning for infinite-horizon approximate optimal tracking. IEEE Trans. Neural Netw. Learn. Syst. 28, 3 (2016), 753–758.

[20] R. Kamalapurkar, J. A. Rosenfeld, and W. E. Dixon. 2016. Efficient model-based reinforcement learning for approximate online optimal control. Automatica 74 (2016), 247–258.

[21] R. Kamalapurkar, P. Walters, and W. E. Dixon. 2016. Model-based reinforcement learning for approximate optimal regulation. Automatica 64 (2016), 94–104.

[22] R. Kamalapurkar, P. Walters, J. A. Rosenfeld, and W. E. Dixon. 2018. Reinforcement Learning for Optimal Feedback Control: A Lyapunov-Based Approach. Springer.