Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

ARTICLE TYPE

# Controller Synthesis against Omega-Regular Specifications: A Funnel-based Control Approach

Pushpak Jagtap*[1] | Dimos V. Dimarogonas[2]

[1]Robert Bosch Center for Cyber Physical Systems, Indian Institute of Science, Bengaluru, India
[2]Division of Decision and Control Systems, KTH Royal Institute of technology, Stockholm, Sweden

**Correspondence**
*Pushpak Jagtap, Robert Bosch Center for Cyber-Physical Systems, Indian Institute of Science, Bengaluru, India.
Email: pushpak@iisc.ac.in

**Summary**

The paper focuses on the problem of formal synthesis of controllers for control-affine nonlinear systems against complex properties. Our goal is to design a closed-form control policy that guarantees the satisfaction of complex properties that are expressed using ($\omega$)-regular languages and equivalently recognized by Non-deterministic Büchi Automata (NBA). We propose leveraging a funnel-based control approach to provide a closed-form solution to the problem. Our approach decomposes the specification represented by NBA into a sequence of reachability problems which we solve using a funnel-based control approach. Controllers associated with each reachability problem are then combined to design a hybrid control policy enforcing the desired ($\omega$)-regular property. We demonstrate the effectiveness of the proposed results on room temperature control and mobile robot motion control case studies.

**KEYWORDS:**
Formal controller synthesis, Omega-regular specifications, Funnel-based control

## 1 | INTRODUCTION

The formal synthesis of controllers for dynamical systems against complex logic specifications has gained remarkable attention in recent years. These specifications are usually expressed using temporal logic formulae[1] or (in)finite strings over automata[2]. In the literature, abstraction-based approaches[3] are popular for solving such synthesis problems. However, since the abstraction-based approaches usually require discretization of the state and input sets of physical systems, the synthesis problem becomes very intractable for control systems with higher state-space dimensions. To address this scalability issue, state-space discretization-free abstractions[4,5] and compositional abstraction-based techniques[6,7] have been proposed under suitable assumptions on the system dynamics (e.g., Lipschitz continuity or incremental input-to-state stability).

The discretization-free approaches based on control barrier functions (CBF) have shown a potential to solve the formal synthesis problem for various classes of complex logic specifications. The results in [8,9,10] search for parametric control barrier functions to synthesize controllers to enforce specifications given by deterministic forms of Automata. However, there are several limitations over the applicability of these results; for example, (i) the techniques rely on a parametric form of control barrier function (such as polynomial) that will solve the problem, which is very difficult to determine for a given system and (ii) the techniques rely on a numerical search of the unknown parameters by using sum-of-squares or counter-example guided inductive synthesis approaches which are computationally heavy and most of the time suffer from numerical errors. There are a few works that use another version of the control barrier function methodology for temporal-logic motion planning [11,12], and which rely on the online computation of the controller using optimization tools. Nevertheless, all these approaches do not provide closed-form controllers and rely on, very often numerically intractable and without feasibility guarantees, computational techniques.

To address these issues, this work proposes the use of a funnel-based control approach [13] to provide a closed-form hybrid control policy that enforces complex specifications recognized by (omega)-regular languages [2] (or equivalently provided by non-deterministic Büchi automata (NBA) [14]). There have been few attempts to utilize funnel-based control approaches for enforcing a class of complex specifications. For example, the authors in [15,16] provide results enforcing a fragment of signal temporal logic specifications, and the work in [17] presents results enforcing a metric interval temporal logic for cooperative manipulation by discretizing state-space. However, to the best of our knowledge, this is the first paper to provide closed-form solutions enforcing specifications that are recognized by ($\omega$)-regular languages. In particular, first, we provide funnel-based closed-form controllers that solve simple reachability problems for control-affine nonlinear systems. Then, we decompose given specifications as a sequence of reachability problems generated with the help of NBA recognizing the given ($\omega$)-regular property. Finally, we combine the obtained controllers corresponding to each reachability task to design a hybrid control policy that enforces the given ($\omega$)-regular property.

In the following, we summarize the key contributions of this work.

- Unlike the abstraction-based approaches [3,4], reachable set computations-based approaches [18], and the CBF-based discretization-free approaches [8,9], we provide a closed-form solution that does not require any numerical computations to design a controller enforcing complex logic specifications.
- The proposed result can handle a general class of complex specifications that are recognized by ($\omega$)-regular languages (which is capable of capturing the full class of linear temporal logic (LTL) specifications).
- The results in [8,9] work only with deterministic forms of automata and need to compute the complement of automata representing original specifications. In contrast, the proposed results work directly with NBA that recognizes $\omega$-regular languages.

The remainder of this paper is structured as follows. In section 2, we introduce system dynamics and the complex specifications given by the $\omega$-regular property. Then, we formally define the problem considered in this paper. Section 3 provides a result on solving a reachability specification by utilizing a funnel-based control approach. In Section 4, we discuss the design of a closed-form hybrid control policy by utilizing the results obtained in Section 3 to enforce the $\omega$-regular property. Section 5 demonstrates the effectiveness of the results on two case studies: $(i)$ temperature control in a room and $(ii)$ mobile robot motion control under complex tasks. Finally, Section 6 concludes the paper.

## 2 | PRELIMINARIES AND PROBLEM STATEMENT

### 2.1 | Notations

The symbols $\mathbb{N}$, $\mathbb{N}_0$, $\mathbb{Z}$, $\mathbb{R}$, $\mathbb{R}^+$, and $\mathbb{R}_0^+$ denote the set of natural, nonnegative integer, integer, real, positive real, and nonnegative real numbers, respectively. We use $\mathbb{R}^{n\times m}$ to denote a vector space of real matrices with $n$ rows and $m$ columns. We use $\|\cdot\|$ to represent the Euclidean norm. For $a, b \in \mathbb{R}$ and $a < b$, we use $(a, b)$ to represent open interval in $\mathbb{R}$. For $a, b \in \mathbb{N}$ and $a \leq b$, we use $[a; b]$ to denote close interval in $\mathbb{N}$. We use $I_n$ and $0_{n\times m}$ to denote identity matrix in $\mathbb{R}^{n\times n}$ and zero matrix in $\mathbb{R}^{n\times m}$, respectively. A diagonal matrix in $\mathbb{R}^{n\times n}$ with diagonal entries $d_1, \dots, d_n$ is denoted by $diag\{d_1, \dots, d_n\}$. Given a matrix $M \in \mathbb{R}^{n\times m}$, $M^T$ represents transpose of matrix $M$. Given a matrix $P \in \mathbb{R}^{n\times n}$, $\mathrm{Tr}(P)$ represents trace of matrix $P$. Given a set $A$, we use $|A|$ to represent the cardinality of the set $A$. $x_i$, $i \in [1; n]$ denotes $i$-th element of vector $x \in \mathbb{R}^n$. Consider $N$ sets $A_i$, $i \in \{1, \dots, N\}$, the Cartesian product of the sets is given by $A = \prod_{i\in\{1,\dots,N\}} A_i := \{(a_1, \dots, a_N)|a_i \in A_i, i \in \{1, \dots, N\}\}$. Consider a set $X_a \subset \mathbb{R}^n$, its projection on $i$th dimension, where $i \in [1; n]$, is given by an interval $[\underline{X}_{ai}, \overline{X}_{ai}] \subset \mathbb{R}$, where $\underline{X}_{ai} := \min\{x_i \in \mathbb{R} \mid [x_1, x_2, \dots, x_n] \in X_a\}$ and $\overline{X}_{ai} := \max\{x_i \in \mathbb{R} \mid [x_1, x_2, \dots, x_n] \in X_a\}$; and $Int(X_a)$ represents the interior of set $X_a$. We denote the empty set by $\emptyset$. Given a set $S$, the notation $|S|$ denotes the cardinality of $S$; $S^*$ and $S^\omega$ denote the set of all finite and infinite strings over $S$, respectively. Given sets $U$ and $S \subset U$, the complement of $S$ with respect to $U$ is defined as $U\backslash S = \{x : x \in U, x \notin S\}$.

### 2.2 | System Description

In this work, we consider control-affine systems $\mathcal{S}$ given by ordinary differential equations of the form

$$\mathcal{S} : \dot{x} = f(x) + g(x)u, \tag{2.1}$$

where $x(t) = [x_1(t), x_2(t), \dots, x_n(t)]^T \in X \subseteq \mathbb{R}^n$ and $u(t) \in \mathbb{R}^m$ are state and input vectors, respectively. We assume that $g(x(t))g(x(t))^T$ is positive definite for all $x(t) \in X$. We use $x_{x_0 u}$ to denote a trajectory of $\mathcal{S}$ starting from initial state $x_0 \in X$ under a control signal $u$.

## 2.3 | Class of Specifications

The main goal in this work is to synthesize controllers for systems (2.1) ensuring the satisfaction of $\omega$-regular properties[2]. Such specifications can be expressed by $\omega$-automata that can recognize infinite words, such as non-deterministic Büchi automata[14], deterministic Rabin automata[19], deterministic Streett automata[20], parity automata or Muller automata[21]. While the above-mentioned automata have different acceptance conditions, they have the same expressive power and all of them recognize $\omega$-regular languages. Here, we use non-deterministic Büchi automata (NBA) to describe $\omega$-regular properties, which are formally defined next.

**Definition 1.** A non-deterministic Büchi automaton (NBA) is a tuple $\mathcal{A} = (Q, Q_0, \Sigma, \delta, F)$, where $Q$ is a finite set of states, $Q_0 \subseteq Q$ is a set of initial states, $\Sigma$ is a finite set of alphabets, $\delta : Q \times \Sigma \rightarrow 2^Q$ is a transition function, and $F \subseteq Q$ is a set of accepting states.

We use notation $q \xrightarrow{\sigma}_{\mathcal{A}} q'$ to denote transition $(q, \sigma, q') \in \delta$ in an NBA $\mathcal{A}$. Consider an infinite state-run $\mathsf{q} = (q_0, q_1, \dots) \in Q^\omega$, an infinite word (a.k.a. trace) $\sigma = (\sigma_0, \sigma_1, \dots) \in \Sigma^\omega$ such that $q_0 \in Q_0$, $q_i \xrightarrow{\sigma_i}_{\mathcal{A}} q_{i+1}$ for all $i \in \mathbb{N}_0$, and let $\mathsf{Inf}(\mathsf{q})$ be the set of states that occur infinitely many times in $\mathsf{q}$. An infinite word $\sigma = (\sigma_0, \sigma_1, \dots) \in \Sigma^\omega$ is accepted by NBA $\mathcal{A}$ if there exists an infinite state run $\mathsf{q}$ corresponding to $\sigma$ such that $\mathsf{Inf}(\mathsf{q}) \cap F \neq \emptyset$. The set of words accepted by $\mathcal{A}$ is called the accepting language of $\mathcal{A}$ and is denoted by $\mathcal{L}(\mathcal{A})$.

We consider specifications expressed by accepting languages of NBA $\mathcal{A}$ when input symbols are defined over a set of atomic propositions $\Pi$ as the alphabet, i.e., $\Sigma = 2^\Pi$. We should highlight that the full class of linear temporal logic (LTL) properties can be represented using NBA and the conversion can be done by the existing tools such as SPOT[22] and LTL2BA[23].

*Remark 1.* The approach proposed in this work can also be used for $\omega$-regular languages represented using deterministic Streett automata and regular languages represented by (non)deterministic finite automata[24]. From a temporal logic perspective, the full class of LTL can be represented by deterministic Streett automata (one can use tool ltl2dstar[25] for conversions) and LTL over finite traces (LTL$_f$)[26] can be represented using deterministic finite automata (one can utilize toolbox MONA[27] for conversions).

## 2.4 | Satisfaction of Specification by Systems $\mathcal{S}$

A given system $\mathcal{S}$ in (2.1) is connected to the specification given by the accepting language of an NBA $\mathcal{A}$ defined over the set of atomic propositions $\Pi$, with the help of a labeling function $L : X \rightarrow \Pi$ as described in the next definition which is adapted from[28, Definition 2].

**Definition 2.** For a system $\mathcal{S}$ in (2.1) and a labeling function $L : X \to \Pi$, an infinite sequence $\sigma(x_{x_0 u}) = (\sigma_0, \sigma_1, \ldots) \in \Pi^\omega$ is an infinite trace of the trajectory $x_{x_0 u}$ of $\mathcal{S}$ if there exists an associated timing sequence $t_0, t_1, \ldots$ such that $t_0 = 0$, $t_r \to \infty$ as $r \to \infty$, and for all $j \in \mathbb{N}$, $t_j \in \mathbb{R}_0^+$, and the following conditions hold

- $t_j < t_{j+1}$;
- $x_{x_0 u}(t_j) \in L^{-1}(\sigma_j)$;
- If $\sigma_j \neq \sigma_{j+1}$, then for some $t'_j \in [t_j, t_{j+1}]$, $x_{x_0 u}(t) \in L^{-1}(\sigma_j)$ for all $t \in (t_j, t'_j)$; $x_{x_0 u}(t) \in L^{-1}(\sigma_{j+1})$ for all $t \in (t'_j, t_{j+1})$; and either $x_{x_0 u}(t'_j) \in L^{-1}(\sigma_j)$ or $x_{x_0 u}(t'_j) \in L^{-1}(\sigma_{j+1})$.

Next, we define the satisfaction of specification given by the language of an NBA $\mathcal{A}$.

**Definition 3.** Consider a system $\mathcal{S}$ in (2.1), a specification given by the accepting language of an NBA $\mathcal{A}$, and $\sigma(x_{x_0 u})$ (i.e., an infinite trace of trajectory $x_{x_0 u}$) as in Definition 2. We say that the trajectory of $\mathcal{S}$ starting from initial state $x_0 \in X$ under input signal $u$ satisfies specification given by $\mathcal{A}$, denoted by $\sigma(x_{x_0 u}) \vDash \mathcal{A}$, if $\sigma(x_{x_0 u}) \in \mathcal{L}(\mathcal{A})$.

### 2.5 | Problem Definition

The main controller synthesis problem considered in this work is formally defined next.

**Problem 1.** Consider a system $\mathcal{S}$ in (2.1), a specification given by the accepting language of an NBA $\mathcal{A} = (Q, Q_0, \Pi, \delta, F)$ over a set of atomic propositions $\Pi = \{p_0, p_1, \ldots, p_M\}$, and a labeling function $L : X \to \Pi$. We aim at computing a closed-form hybrid controller $u$ such that $\sigma(x_{x_0 u}) \vDash \mathcal{A}$ for all $x_0 \in L^{-1}(p_i)$ and some $i \in [1; M]$ (i.e., we will have different controllers based on different starting regions associated with each initial atomic propositions).

To solve the aforementioned problem, we first utilize a funnel-based control approach which is discussed in the next section to solve reachability specification.

## 3 | REACHABILITY USING FUNNEL-BASED CONTROL

In this section, we propose the use of a funnel-based control approach [13] to solve the reachability problem, which will later serve as a key element to solve Problem 1. Consider a funnel representing time-varying bounds for the trajectory $x_i, i \in [1; n]$ given as follows

$$-c_i \rho_i(t) < x_i(t) - \eta_i < d_i \rho_i(t) \tag{3.1}$$

for all $t \in \mathbb{R}_0^+$, where $\rho_i : \mathbb{R}_0^+ \to \mathbb{R}^+$, $i \in [1; n]$ are positive, smooth, and strictly decreasing funnel functions, $c_i, d_i \in \mathbb{R}_0^+$ and $\eta_i \in \mathbb{R}$ are some constants. In this work, we consider the following form of funnel function

$$\rho_i(t) = \rho_{i0} \mathrm{e}^{-\epsilon_i t} + \rho_{i\infty}, \tag{3.2}$$

where $\rho_{i0}$, $\rho_{i\infty}$, $\epsilon_i \in \mathbb{R}^+$ are positive constants and $\rho_{i\infty} = \lim_{t\to\infty} \rho_i(t)$. Now, by normalizing $x_i(t) - \eta_i$ with respect to the performance function $\rho_i(t)$, we define the modulating error as $\hat{x}_i(t) := \frac{x_i(t)-\eta_i}{\rho_i(t)}$ and the corresponding performance region $\hat{\mathcal{D}}_i := \{\hat{x}_i \mid \hat{x}_i \in (-c_i, d_i)\}$. Then, the modulated error is transformed through a transformation function $T_i : \hat{\mathcal{D}}_i \to \mathbb{R}$ such that $T_i(0) = 0$ and is chosen as

$$T_i(\hat{x}_i) = \ln\left(\frac{d_i(c_i + \hat{x}_i)}{c_i(d_i - \hat{x}_i)}\right). \tag{3.3}$$

The transformed error is then defined as $\xi_i(x_i(t), \rho_i(t)) := T_i(\hat{x}_i)$. It can be verified that if the transformed error is bounded, then the modulated error $\hat{x}_i$ is constrained within the region $\hat{\mathcal{D}}_i$. This also implies that $x_i(t) - \eta_i$ evolves within the bounds given in (3.1). Differentiating $\xi_i$ with respect to time, we obtain transformed error dynamics for $i$th dimension as

$$\dot{\xi}_i = \phi_i(\hat{x}_i, t)[\dot{x}_i + \alpha_i(t)(x_i - \eta_i)], \tag{3.4}$$

where $\phi_i(\hat{x}_i, t) := \frac{1}{\rho_i(t)} \frac{\partial T_i(\hat{x}_i)}{\partial \hat{x}_i} > 0$ for all $\hat{x}_i \in (-c_i, d_i)$ and $\alpha_i(t) := -\frac{\dot{\rho}_i(t)}{\rho_i(t)} > 0$ for all $t \in \mathbb{R}_0^+$ are the normalized Jacobian of the transformation function $T_i$ and the normalized derivative of the performance function $\rho_i$, respectively. Now, by stacking all the transformed error dynamics, one gets

$$\dot{\xi} = \Phi_t(\dot{x} + \alpha_t(x - \eta)), \tag{3.5}$$

where $\xi = [\xi_1, \dots, \xi_n]^T$, $\Phi_t = diag\{\phi_1(\hat{x}_1, t), \dots, \phi_n(\hat{x}_n, t)\}$, $\alpha_t = diag\{\alpha_1(t), \dots, \alpha_n(t)\}$, and $\eta = [\eta_1, \dots, \eta_n]$. In the next theorem, we provide the result for enforcing reachability specification by utilizing the funnel approach.

**Theorem 1.** Consider system $\mathcal{S}$ in (2.1), any arbitrary sets $X_a, X_b \subset X$, $\Xi_i := [\underline{X}_{ai}, \overline{X}_{ai}] \cap [\underline{X}_{bi}, \overline{X}_{bi}]$, $\underline{X}_i := \min\{\underline{X}_{ai}, \underline{X}_{bi}\}$, $\overline{X}_i := \max\{\overline{X}_{ai}, \overline{X}_{bi}\}$, an arbitrarily chosen state $\eta = [\eta_1, \eta_2, \dots, \eta_n]^T \in Int(X_b)$ satisfying

$$\eta_i \in \begin{cases} \Xi_i & \text{if } \Xi_i \neq \emptyset \\ [\underline{X}_{bi}, \overline{X}_{bi}] & \text{if } \Xi_i = \emptyset, \end{cases}$$

$i \in [1; n]$, and funnel function (3.1) with $\epsilon_i \in \mathbb{R}^+$,

$$\rho_{i0} = \begin{cases} \max\{|\eta_i - \underline{X}_{ai}|, |\eta_i - \overline{X}_{ai}|\} & \text{if } \Xi_i \neq \emptyset \\ \max\{|\eta_i - \underline{X}_i|, |\eta_i - \overline{X}_i|\} & \text{if } \Xi_i = \emptyset, \end{cases}$$

constants $c_i, d_i$ as follows:

$$
\begin{aligned}
c_i &= \frac{|\eta_i - \underline{X}_{ai}|}{\rho_{i0}}, d_i = 1, && \text{if } \Xi_i \neq \emptyset \text{ and } |\eta_i - \underline{X}_{ai}| \leq |\eta_i - \overline{X}_{ai}| \\
c_i &= 1, d_i = \frac{|\eta_i - \overline{X}_{ai}|}{\rho_{i0}}, && \text{if } \Xi_i \neq \emptyset \text{ and } |\eta_i - \underline{X}_{ai}| > |\eta_i - \overline{X}_{ai}| \\
c_i &= \frac{|\eta_i - \underline{X}_i|}{\rho_{i0}}, d_i = 1, && \text{if } \Xi_i = \emptyset \text{ and } |\eta_i - \underline{X}_i| \leq |\eta_i - \overline{X}_i| \\
c_i &= 1, d_i = \frac{|\eta_i - \overline{X}_i|}{\rho_{i0}}, && \text{if } \Xi_i = \emptyset \text{ and } |\eta_i - \underline{X}_i| > |\eta_i - \overline{X}_i|,
\end{aligned}
$$

and $\rho_{i\infty}$ is such that $\prod_{i\in[1;n]} \eta_i + [-c_i\rho_{i\infty}, d_i\rho_{i\infty}] \subset X_b$.

Then under time-varying control law:

$$
u(x,\rho) = -g(x)^T(g(x)g(x)^T)^{-1}(f(x) + \xi(x,\rho) + \overline{\epsilon}(x-\eta)), \tag{3.6}
$$

where $\xi(x,\rho) = [\xi_1(x_1,\rho_1), \ldots, \xi_n(x_n,\rho_n)]^T := \left[\ln\left(\frac{d_1\left(c_1 + \frac{x_1-\eta_1}{\rho_1}\right)}{c_1\left(d_1 - \frac{x_1-\eta_1}{\rho_1}\right)}\right), \ldots, \ln\left(\frac{d_n\left(c_n + \frac{x_n-\eta_n}{\rho_n}\right)}{c_n\left(d_n - \frac{x_n-\eta_n}{\rho_n}\right)}\right)\right]^T$ is a transformation error in (3.3) as discussed in Section 3 and $\overline{\epsilon} := \max_{i\in[1;n]} \epsilon_i$, one can ensure that $\exists t \in \mathbb{R}_0^+$ such that $x_{x_0u}(t) \cap X_b \neq \emptyset$ for all $x_0 \in X_a$. In other words, the trajectory starting for any initial point in $X_a$ will reach $X_b$ in a finite time under the control law (3.6).

*Proof.* Consider Lyapunov like function $V = \frac{1}{2}\xi^T\xi$ and

$$
\begin{aligned}
\dot{V} &= \xi^T\Phi_t(f(x) + g(x)u + \alpha_t(x-\eta)) \\
&= \xi^T\Phi_t(f(x) - g(x)g(x)^T(g(x)g(x)^T)^{-1}(f(x) + \xi + \overline{\epsilon}(x-\eta)) + \alpha_t(x-\eta)).
\end{aligned}
$$

By following the facts that $\Phi_t$ and $\alpha_t$ are positive definite matrices, $\alpha_t < \overline{\epsilon} := \max_{i\in[1;n]} \epsilon_i$ and $\xi^T(x-\eta) \geq 0$ (this is due to $\xi_i(\hat{x}_i)$ is strictly increasing and $\xi_i(0) = 0$), one obtains $\dot{V} \leq -\xi^T\Phi_t\xi$. This implies that $\xi(t)$ is bounded for all $t \in \mathbb{R}_0^+$ and hence we guarantee (3.1) that is $-c_i\rho_i(t) + \eta_i < x_i(t) < d_i\rho_i(t) + \eta_i$. From the choice of $\eta \in Int(X_b)$ and constants $\rho_{i0}, \rho_{i\infty}, c_i, d_i, \eta_i$ for all $i \in [1;n]$, one can readily ensure that $X_a \subseteq \prod_{i\in[1;n]} [-c_i\rho_i(0\}) + \eta_i, -d_i\rho_i(0) + \eta_i]$ and as $\lim_{t\to\infty} \prod_{i\in[1;n]} [-c_i\rho_i(t) + \eta_i, -d_i\rho_i(t) + \eta_i] = \prod_{i\in[1;n]} \eta_i + [-c_i\rho_{i\infty}, d_i\rho_{i\infty}] \subset X_b$. This implies that there exist $t \in \mathbb{R}_0^+$ such that $x_{x_0u}(t) \cap X_b \neq \emptyset$ for all $x_0 \in X_a$. □

*Remark 2.* It is worth emphasizing that the sets $X_a$ and $X_b$ considered in Theorem 1 are arbitrary sets. The projection intervals of those arbitrary sets on the $i$th dimensions are used to design the corresponding funnel parameters.

# 4 | EXTENSION TO COMPLEX SPECIFICATIONS

In this section, we first provide an approach to decompose specifications given by the language of an NBA into a sequence of reachability tasks. Then by utilizing the result of Theorem 1, we provide a construction of hybrid control policy solving Problem 1.

## 4.1 | Sequential Reachability Decomposition

Consider an NBA $\mathcal{A} = (Q, Q_0, \Pi, \delta, F)$ corresponding to an $\omega$-regular language expressing the properties of interest for the system $\mathcal{S}$.

For a given accepting state-run $\mathsf{q}$ of $\mathcal{A}$, we denote the corresponding infinite words by $\sigma(\mathsf{q}) \subseteq \Pi^{\omega}$. We also use a similar notation to denote finite words corresponding to finite state runs (i.e., $\sigma(\bar{\mathsf{q}}) \in \Pi^n$ for $\bar{\mathsf{q}} \in Q^{n+1}, n \in \mathbb{N}$). It is known that there exists a word $\sigma \in \Pi^{\omega}$ accepted by $\mathcal{A}$ if there exists a state run of $\mathcal{A}$ of the form $\mathsf{q} = (q_0^r, q_1^r, \ldots, q_{\mathsf{m}_r}^r, (q_0^s, q_1^s, \ldots, q_{\mathsf{m}_s}^s)^{\omega}) \in Q^{\omega}$, where $\mathsf{m}_r, \mathsf{m}_s \in \mathbb{N}$, $q_0^r \in Q_0$ and $q_0^s \in F$. Let $\bar{\mathsf{q}}$ be a finite state run fragment of an accepting run $\mathsf{q}$ constructed by considering infinite sequence $(q_0^s, q_1^s, \ldots, q_{\mathsf{m}_s}^s)$ only once and is given by $\bar{\mathsf{q}} = (q_0^r, q_1^r, \ldots, q_{\mathsf{m}_r}^r, q_0^s, q_1^s, \ldots, q_{\mathsf{m}_s}^s, q_0^s, q_1^s) \in Q^*$.

Let $\mathcal{R}$ be the set of all such finite state-run fragments excluding self-loops,

$$\mathcal{R} := \{\bar{\mathsf{q}} = (q_0^r, q_1^r, \ldots, q_{\mathsf{m}_r}^r, q_0^s, q_1^s, \ldots, q_{\mathsf{m}_s}^s, q_0^s, q_1^s) \,|\, q_0^r \in Q_0, q_0^s \in F, q_i^r \neq q_{i+1}^r, \forall i < \mathsf{m}_r, \text{and } q_j^s \neq q_{j+1}^s, \forall j < \mathsf{m}_s\}. \tag{4.1}$$

Computation of $\mathcal{R}$ can be done algorithmically by viewing $\mathcal{A}$ as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with vertices $\mathcal{V} = Q$ and edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ such that $(q, q') \in \mathcal{E}$ if and only if $q' \neq q$ and there exist $p \in \Pi$ such that $q \xrightarrow{p}_{\mathcal{A}} q'$. We call a finite sequence of states $(q_0, q_1, \ldots, q_{\tilde{n}}) \in Q^{\tilde{n}}$, $\tilde{n} \in \mathbb{N}$, satisfying $(q_i, q_{i+1}) \in \mathcal{E}$, for all $i \in [0, \ldots, \tilde{n} - 1]$ a path in the graph $\mathcal{G}$. For any $(q, q') \in \mathcal{E}$, we denote the atomic proposition associated with the edge $(q, q')$ by $\sigma(q, q')$. Now, one can easily compute $\mathcal{R}$ using variants of depth-first search algorithm[29] over $\mathcal{G}$. For each $p \in \Pi$, we define a set $\mathcal{R}^p$ as

$$\mathcal{R}^p := \{\bar{\mathsf{q}} = (q_0^r, q_1^r, \ldots, q_{\mathsf{m}_r}^r, q_0^s, q_1^s, \ldots, q_{\mathsf{m}_s}^s, q_0^s, q_1^s) \in \mathcal{R} \mid \sigma(q_0^r, q_1^r) = p\}. \tag{4.2}$$

The above notation with superscript $p$ makes partitions of set $\mathcal{R}$ according to the initial atomic proposition of finite state-run fragments in (4.1). In particular, it will help in defining hybrid controllers corresponding to each initial state set $L^{-1}(p_i)$, $i \in \{1, 2, \ldots, M\}$. Decomposition into sequential reachability is performed as follows. For any $\bar{\mathsf{q}} = (q_0, q_1, \ldots, q_{\mathsf{m}_r+\mathsf{m}_s+3}) \in \mathcal{R}^p$, we define $\mathcal{P}^p(\bar{\mathsf{q}})$ as a set of all state runs of length 3 (referred to as triplets),

$$\mathcal{P}^p(\bar{\mathsf{q}}) := \{(q_i, q_{i+1}, q_{i+2},) \mid 0 \leq i \leq \mathsf{m}_r + \mathsf{m}_s + 1\}. \tag{4.3}$$

For a better understanding, the decomposition into sequential reachability is demonstrated below with the following example.

**Example 1.** Consider an NBA $\mathcal{A}$ as shown in Figure 1, where $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}$, $Q_0 = \{q_0\}$, $\Pi = \{p_0, p_1, p_2, p_3\}$,

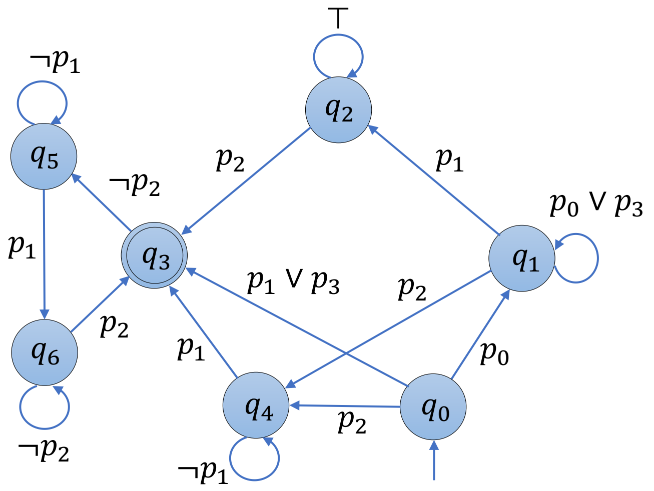


**Figure 1** NBA $\mathcal{A}$ used in Example 1.

and $F = \{q_3\}$.

The set of accepting state runs is $\{(q_0, q_1^*, q_2^*, (q_3, q_5^*, q_6^*)^\omega), (q_0, q_1^*, q_4^*, (q_3, q_5^*, q_6^*)^\omega), (q_0, q_4^*, (q_3, q_5^*, q_6^*)^\omega), (q_0, (q_3, q_5^*, q_6^*)^\omega)\}$. The set of finite state-run fragments $\mathcal{R}$ in (4.1) is obtained as follow:

$$\mathcal{R} = \{(q_0, q_1, q_2, q_3, q_5, q_6, q_3, q_5), (q_0, q_1, q_4, q_3, q_5, q_6, q_3, q_5), (q_0, q_4, q_3, q_5, q_6, q_3, q_5), (q_0, q_3, q_5, q_6, q_3, q_5)\}.$$

The sets $\mathcal{R}^p$ for $p \in \Pi$ are as follows:

$$\mathcal{R}^{p_0} = \{(q_0, q_1, q_2, q_3, q_5, q_6, q_3, q_5), (q_0, q_1, q_4, q_3, q_5, q_6, q_3, q_5)\}, \mathcal{R}^{p_1} = \{((q_0, q_3, q_5, q_6, q_3, q_5)\},$$

$$\mathcal{R}^{p_2} = \{(q_0, q_4, q_3, q_5, q_6, q_3, q_5)\}, \mathcal{R}^{p_3} = \{(q_0, q_3, q_5, q_6, q_3, q_5)\}.$$

The sets $\mathcal{P}^p(\overline{\mathsf{q}})$ for $\overline{\mathsf{q}} \in \mathcal{R}^p$ are as follows:

$\mathcal{P}^{p_0}(q_0, q_1, q_2, q_3, q_5, q_6, q_3, q_5) = \{(q_0, q_1, q_2),(q_1, q_2, q_3),(q_2, q_3, q_5),(q_3, q_5, q_6), (q_5, q_6, q_3),(q_6, q_3, q_5)\}$,

$\mathcal{P}^{p_0}(q_0, q_1, q_4, q_3, q_5, q_6, q_3, q_5) = \{(q_0, q_1, q_4),(q_1, q_4, q_3), (q_4, q_3, q_5),(q_3, q_5, q_6),(q_5, q_6, q_3),(q_6, q_3, q_5)\}$,

$\mathcal{P}^{p_2}(q_0, q_4, q_3, q_5, q_6, q_3, q_5) = \{(q_0, q_4, q_3), (q_4, q_3, q_5),(q_3, q_5, q_6),(q_5, q_6, q_3),(q_6, q_3, q_5)\}$,

$\mathcal{P}^{p_1}(q_0, q_3, q_5, q_6, q_3, q_5) = \mathcal{P}^{p_3}(q_0, q_3, q_5, q_6, q_3, q_5) = \{(q_0, q_3, q_5), (q_3, q_5, q_6), (q_5, q_6, q_3), (q_6, q_3, q_5)\}$.

## 4.2 | Hybrid Control Policy

Each triplet $\nu = (q, q', q'') \in \mathcal{P}^p(\overline{\mathsf{q}})$, $p \in \Pi$ can be viewed as reach while avoid specification. In particular, the system trajectory should reach region $L^{-1}(\sigma(q', q''))$ while avoiding region $X \setminus L^{-1}(\sigma(q', q'))$ starting from some initial region $L^{-1}(\sigma(q, q'))$. Next, we raise an Assumption 1 on the locations of these sets under which we can use the proposed approach to solve Problem 1.

**Assumption 1.** We assume that there exist an accepting state-run $\mathsf{q}$ such that for all $\nu = (q, q', q'') \in \mathcal{P}^p(\overline{\mathsf{q}})$, $p \in \Pi$, either of the following hold:

- $X \setminus L^{-1}(\sigma(q', q')) = L^{-1}(\sigma(q', q''))$ or
- $X \setminus L^{-1}(\sigma(q', q')) \cap \prod_{i \in [1;n]} [-c_i \rho_i(0) + \eta_i, -d_i \rho_i(0) + \eta_i] = \emptyset$,

where $\rho_i(0)$, $c_i$, $d_i$, and $\eta_i$ are as defined in Theorem 1 with $X_a = L^{-1}(\sigma(q, q'))$ and $X_b = L^{-1}(\sigma(q', q''))$.

*Remark 3.* Assumption 1 says that either the avoid region should be the complement of the reach region or the intersection of the avoid region with the region given by the funnel should be empty. In general, NBA have more than one accepting state-run, so the possibility of having an accepting state-run that does not satisfy Assumption 1 is much smaller. Even in the unlikely case that there is no accepting state-run satisfying the above assumption, one can introduce dummy atomic propositions that ensure that the assumption is satisfied with the newly constructed NBA without affecting the original specification. One can readily observe it for the 2-dimensional case. However, providing a generalized and automated algorithm is challenging and will be considered as a future research direction. Another advantage of Assumption 1 is to filter out practically infeasible accepting state runs (for example, those that do not have continuity). For better understanding, consider the temperature control case study (given in Subsection 5.1) where a triplet $(q_1, q_3, q_1)$ in Figure 3 is associated with a reachability task defined as "reach $L^{-1}(p_2)$ from $L^{-1}(p_1)$ without entering any other region". Then one can readily see from Figure 4 that the task is not feasible.

Next, we provide a Lemma that correlates a particular segment in specification automata NBA $\mathcal{A}$ (referred to as triplet) with the result of Theorem 1.

**Lemma 1.** For a triplet $\nu = (q, q', q'') \in \mathcal{P}^p(\overline{\mathsf{q}})$, where $\overline{\mathsf{q}} \in \mathcal{R}^p$ for some $p \in \Pi$ satisfying Assumption 1, if we use control policy $u_\nu(x, \rho_\nu)$ as in (3.6) in Theorem 1 with $X_a = L^{-1}(\sigma(q, q'))$ and $X_b = L^{-1}(\sigma(q', q''))$, then there exist $t \in \mathbb{R}_0^+$ such that the trajectory $\xi_{x_0 u_\nu}$ of $\mathcal{S}$ starting from any initial state $x_0 \in X_a$ under policy $u_\nu$ satisfies $x_{x_0 u_\nu} \cap X_b \neq \emptyset$.

*Proof.* The proof follows similar to that of Theorem 1. □

Given an NBA $\mathcal{A} = (Q, Q_0, \Pi, \delta, F)$ expressing the properties of interest, an accepting run $\mathsf{q}$, and corresponding finite state-run fragment $\overline{\mathsf{q}}$ satisfying Assumption 1, next we define a finite state transition system which provides a switching mechanism for a hybrid control policy. The switching mechanism is given by a finite state transition system $\mathfrak{S} = (Q_s, Q_{0s}, \Pi, \delta_s)$, where $Q_{0s} = Q_0$, $Q_s = Q_{0s} \cup \mathcal{P}^p(\overline{\mathsf{q}})$, and transition relation $(q_s, \sigma, q_s') \in \delta_s$ (also denoted by $q_s \xrightarrow{\sigma}_{\mathfrak{S}} q_s'$) is defined as:

- for all $q_s = q_0 \in Q_{0s}$,
  - $q_0 \xrightarrow{\sigma(q_0, q')}_{\mathfrak{S}} (q_0, q', q'')$, where $q_0 \xrightarrow{\sigma(q_0, q')}_{\mathcal{A}} q'$;

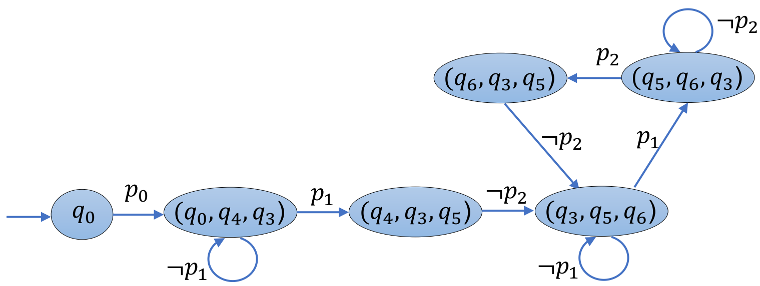


**Figure 2** Switching mechanism given by state transition system

- for all $q_s = (q, q', q'') \in Q_s \setminus Q_{0s}$,
  - $(q, q', q'') \xrightarrow{\sigma(q',q'')}_{\mathfrak{S}} (q', q'', q''')$, such that $q, q', q'', q''' \in Q$, $q' \xrightarrow{\sigma(q',q'')}_{\mathcal{A}} q''$, $q'' \xrightarrow{\sigma(q'',q''')}_{\mathcal{A}} q'''$.

The hybrid controller defined over the augmented state-space $X \times Q_s$ that is a candidate for solving Problem 1 is given by

$$\tilde{\mathbf{u}}(x(t), q_s) := u_{q'_s}(x(t), \rho_{q'_s}(t - t_{q_s})), \quad \forall (q_s, L(x), q'_s) \in \delta_s, \ \forall t \in \mathbb{R}_0^+, \tag{4.4}$$

where $t_{q_s} \in \mathbb{R}_0^+$ is the time instance defined as $t_{q_s} := \min\{t \in \mathbb{R}_0^+ \mid x_{x_0\tilde{\mathbf{u}}}(t) \in \sigma(q', q'')$ for $q_s = (q, q', q'')\}$. The next theorem shows that under the proposed hybrid controller in (4.4), one can ensure the satisfaction of the specification given by the language of an NBA.

**Theorem 2.** Consider a system $\mathcal{S}$ (2.1) and an NBA $\mathcal{A}$ representing a specification with a finite state-run fragment $\bar{\mathsf{q}} \in \mathcal{R}^p$ for some $p \in \Pi$ corresponding to the accepting run $\mathsf{q}$ satisfying Assumption 1. Then the state trajectory $x_{x_0\tilde{\mathbf{u}}}$ of $\mathcal{S}$ starting from any initial state $x_0 \in L^{-1}(p)$ under the hybrid controller $\tilde{\mathbf{u}}$ satisfies the language of NBA $\mathcal{A}$, i.e., $\sigma(x_{x_o\tilde{\mathbf{u}}}) \vDash \mathcal{A}$.

*Proof.* Consider $p \in \Pi$ and an accepting state run $\mathsf{q} = (q_0^r, q_1^r, \dots, q_{\mathsf{m}_r}^r, (q_0^s, q_1^s, \dots, q_{\mathsf{m}_s}^s)^\omega) \in Q^\omega$ in $\mathcal{A}$ with $\sigma(q_0^r, q_1^r) = p$. Let the corresponding finite state-run be $\bar{\mathbf{q}} \in \mathcal{R}^p$ as defined in Subsection 4.1 satisfying Assumption 1. If we utilize controller (4.4) that provides a switching strategy among the reachability control laws $u_\nu(x)$ among $\nu = (q, q', q'') \in \mathcal{P}^p(\bar{\mathbf{q}})$ as given in Lemma 1, one can conclude $\sigma(\mathsf{q}) \in \mathcal{L}(\mathcal{A})$. By utilizing the definition of labeling function $L$, this implies that the state run $x_{x_0,\tilde{\mathbf{u}}}$ of $\mathcal{S}$ starting from any initial state $x_0 \in L^{-1}(p)$ under policy $\tilde{\mathbf{u}}$ given in (4.4) satisfies $\sigma(x_{x_0\tilde{\mathbf{u}}}) \in \mathcal{L}(\mathcal{A})$. This concludes the proof. □

## 5 | CASE STUDIES

In this section, we consider two case studies to demonstrate the effectiveness of our results.

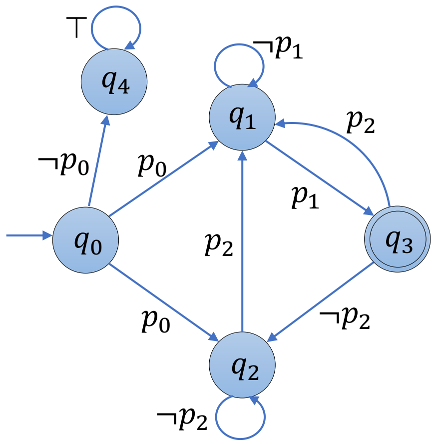


**Figure 3** An NBA representing specification for the room temperature control example

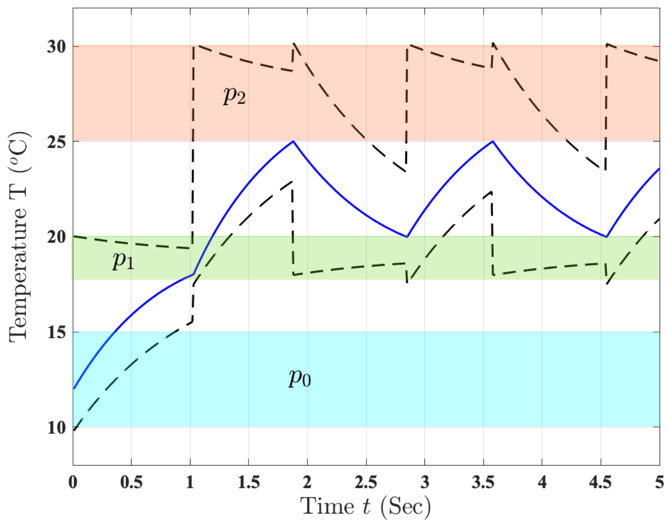


**Figure 4** A trajectory of the system (5.1) under hybrid control policy (4.4). The black dotted lines show funnels in which the trajectory is evolving.

## 5.1 | Temperature control of a room

We consider the evolution of a room temperature given by

$$\dot{T} = \alpha_e(T_e - T) + \alpha_H(T_h - T)u, \tag{5.1}$$

where $T(t)$ denotes the temperature of the room, $u(t)$ represents control input, $T_h = 55°C$ is the heater temperature, $T_e = 15°C$ is the ambient temperature, and $\alpha_e = 8 \times 10^{-3}$ and $\alpha_H = 3.6 \times 10^{-3}$ are heat exchange coefficients. All the parameters are adopted from[5].

The state set of the system is $X = [10, 30]$. We consider regions of interest $X_0 = [10, 15]$, $X_1 = [18, 20]$, $X_2 = [25, 30]$, and $X_3 = X \setminus (X_0 \cup X_1 \cup X_2)$. The set of atomic propositions is given by $\Pi = \{p_0, p_1, p_2, p_3\}$ with labeling function $L(x_i) = p_i$ for all $x_i \in X_i, i \in \{0, 1, 2, 3\}$. The objective is to compute a hybrid control policy ensuring the satisfaction of the specification given by LTL formula $\varphi = p_0 \wedge (\Box\Diamond p_1 \wedge \Box\Diamond p_2)$ or equivalently by the accepting language of the NBA $\mathcal{A}$ in Figure 3. Note that $\Box$ and $\Diamond$ are temporal operators in LTL specification representing always and eventually operators, see[24] for more details. From Figure 3, one can readily identify that there are four possible structures of accepting state runs: $(q_0, q_1^*, (q_3, q_1^*)^\omega)$, $(q_0, q_2^*, q_1^*(q_3, q_1^*)^\omega)$, $(q_0, q_1^*, (q_3, q_2^*, q_1^*)^\omega)$, and $(q_0, q_2^*, q_1^*, (q_3, q_2 *, q_1^*)^\omega)$ with corresponding sets

$$\mathcal{R}^{p_0} = \{(q_0, q_1, q_3, q_1, q_3, q_1), (q_0, q_2, q_1, q_3, q_1, q_3, q_1), (q_0, q_1, q_3, q_2, q_1, q_3, q_2), (q_0, q_2, q_1, q_3, q_2, q_1, q_3, q_2)\},$$

$$\mathcal{R}^{p_1} = \mathcal{R}^{p_2} = \mathcal{R}^{p_3} = \emptyset, \mathcal{P}^{p_0}(q_0, q_1, q_3, q_1, q_3, q_1) = \{(q_0, q_1, q_3), (q_1, q_3, q_1), (q_3, q_1, q_3), (q_1, q_3, q_1)\},$$

$$\mathcal{P}^{p_0}(q_0, q_2, q_1, q_3, q_2, q_1, q_3, q_2) = \{(q_0, q_2, q_1), (q_2, q_1, q_3), (q_1, q_3, q_2), (q_3, q_2, q_1), (q_2, q_1, q_3), (q_1, q_3, q_2)\},$$

$$\mathcal{P}^{p_0}(q_0, q_2, q_1, q_3, q_1, q_3, q_1) = \{(q_0, q_2, q_1), (q_2, q_1, q_3), (q_1, q_3, q_1), (q_3, q_1, q_3), (q_1, q_3, q_1)\},$$

$$\text{and } \mathcal{P}^{p_0}(q_0, q_1, q_3, q_2, q_1, q_3, q_2) = \{(q_0, q_1, q_3), (q_1, q_3, q_2), (q_3, q_2, q_1), (q_2, q_1, q_3), (q_1, q_3, q_2)\}.$$

One can readily observe that $(q_1, q_3, q_1)$ in $\mathcal{P}^{p_0}(q_0, q_2, q_1, q_3, q_1, q_3, q_1)$ and $\mathcal{P}^{p_0}(q_0, q_1, q_3, q_1, q_3, q_1)$ does not satisfy Assumption 1. Therefore, we choose $\mathcal{P}^{p_0}(q_0, q_1, q_3, q_2, q_1, q_3, q_2)$ to design a hybrid controller in (4.4). Figure 4 shows a trajectory of the system (5.1) under a proposed hybrid control policy (4.4). The black dotted lines show the funnel constructed to design the controller as discussed in Theorem 1. One can readily see that the trajectory is evolving inside the bounds given by constructed funnel functions and hence satisfying the desired specification.

## 5.2 | Mobile Robot Motion Control

For the second case study, we consider a three-wheeled omnidirectional mobile robot[30] given by differential equations as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} \cos(x_3) & -\sin(x_3) & 0 \\ \sin(x_3) & \cos(x_3) & 0 \\ 0 & 0 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} 0 & -1 & L \\ \cos\frac{\pi}{6} & \sin\frac{\pi}{6} & L \\ -\cos\frac{\pi}{6} & \sin\frac{\pi}{6} & L \end{bmatrix}}_{B}{}^{-1} Ru,$$

where two states $x_1$ and $x_2$ indicates robot's position, and $x_3$ indicates robot's orientation with respect to the $x_1$-axis. $R = 0.02$ is the wheel radius, $B$ describes geometric constraints with $L = 0.2$ (radius of the robot body), and $u(t) \in \mathbb{R}^3$ are control inputs. The state set of the system is $X = [0, 100] \times [0, 100] \times [0, 2\pi]$. We consider regions of interest $X_0 = [40, 70] \times [40, 70] \times [0, 2\pi]$, $X_1 = [10, 35] \times [10, 35] \times [0, 2\pi]$, $X_2 = [10, 30] \times [70, 90] \times [0, 2\pi]$, $X_3 = [70, 90] \times [10, 30] \times [0, 2\pi]$, and $X_4 = X \setminus (X_0 \cup X_1 \cup X_2 \cup X_3)$ (see Figure 6). The set of atomic propositions is given by $\Pi = \{p_0, p_1, p_2, p_3, p_4\}$ with labeling function $L(x_i) = p_i$ for all $x_i \in X_i, i \in \{0, 1, 2, 3, 4\}$. The objective is to compute a hybrid control policy ensuring the satisfaction of the specification given by LTL formula $\varphi = \Box\neg p_0 \wedge p3 \wedge \Box\Diamond p_2 \wedge \Diamond p_1$ or equivalently by the accepting language of the NBA $\mathcal{A}$ in Figure 5. From Figure 5, one can readily identify that there

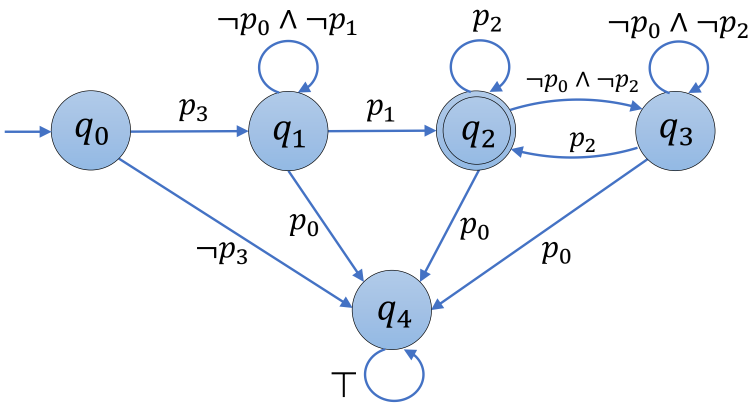


**Figure 5** An NBA representing specification for the mobile robot example

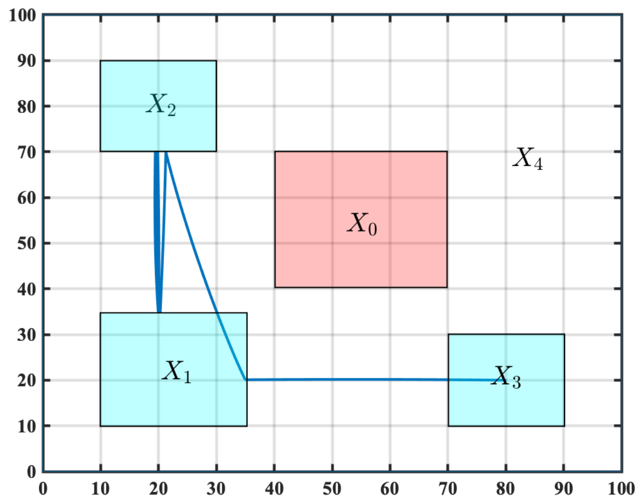


**Figure 6** Simulation under hybrid control policy (4.4)

are three possible structures of accepting state runs: $(q_0, q_1^*, q_2^\omega)$, $(q_0, q_1^*(q_2^*, q_3^*)^\omega)$, and $(q_0, q_1^*, q_2^*, q_3^*, q_2^\omega)$ with corresponding sets $\mathcal{R}^{p_3} = \{(q_0, q_1, q_2, q_2), (q_0, q_1, q_2, q_3, q_2, q_3), (q_0, q_1, q_2, q_3, q_2, q_2)\}$, $\mathcal{R}^{p_0} = \mathcal{R}^{p_1} = \mathcal{R}^{p_2} = \mathcal{R}^{p_4} = \emptyset$; $\mathcal{P}^{p_3}(q_0, q_1, q_2, q_2) = \{(q_0, q_1, q_2), (q_1, q_2, q_2)\}$, $\mathcal{P}^{p_3}(q_0, q_1, q_2, q_3, q_2, q_3) = \{(q_0, q_1, q_2), (q_1, q_2, q_3), (q_2, q_3, q_2), (q_3, q_2, q_3)\}$, $\mathcal{P}^{p_3}(q_0, q_1, q_2, q_3, q_2, q_2) = \{(q_0, q_1, q_2), (q_1, q_2, q_3), (q_2, q_3, q_2), (q_3, q_2, q_2)\}$. One can readily observe that $(q_1, q_2, q_2) \in \mathcal{P}^{p_3}(q_0, q_1, q_2, q_2)$ do not satisfy the Assumption 1. Thus, we choose $\mathcal{P}^{p_3}(q_0, q_1, q_2, q_3, q_2, q_3)$ to design a hybrid controller in (4.4). Figure 6 shows a trajectory of the systems (5.1) under a proposed hybrid control policy (4.4). One can readily see that the trajectory is satisfying the specification given in Figure 5.

# 6 | CONCLUSION

In this paper, we proposed a discretization-free approach for the formal synthesis of controllers for control-affine systems to enforce complex properties expressed by $\omega$-regular languages (or by the language of an NBA). The approach utilizes a funnel-based control approach to provide a closed-form solution to solve the problem. As a future direction, we would like to extend the approach to a more general class of nonlinear systems.

# ACKNOWLEDGMENTS

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation, the Swedish Foundation for Strategic Research (SSF), the Swedish Research Council (VR), the ERC CoG LEAFHOUND project and the Robert Bosch Center for Cyber-Physical Systems.

# References

1. Pnueli A. The temporal logic of programs. In: IEEE. ; 1977: 46–57.

2. Thomas W. Automata on infinite objects. In: Elsevier. 1990 (pp. 133–191).

3. Tabuada P. *Verification and control of hybrid systems: A symbolic approach*. Springer Science & Business Media . 2009.

4. Girard A. Approximately bisimilar abstractions of incrementally stable finite or infinite dimensional systems. In: IEEE. ; 2014: 824–829.

5. Jagtap P, Zamani M. QUEST: A tool for state-space quantization-free synthesis of symbolic controllers. In: Springer. ; 2017: 309–313.

6. Rungger M, Zamani M. Compositional construction of approximate abstractions. In: ACM. ; 2015: 68–77.

7. Saoud A, Jagtap P, Zamani M, Girard A. Compositional abstraction-based synthesis for interconnected systems: An approximate composition approach. *IEEE Transactions on Control of Network Systems* 2021.

8. Jagtap P. *Formal Synthesis of Controllers for Complex Dynamical Systems: State-Space Discretization-free Approaches*. PhD thesis. Universitätsbibliothek der TU München, 2020.

9. Anand M, Lavaei A, Zamani M. Compositional Synthesis of Control Barrier Certificates for Networks of Stochastic Systems against $\omega$-Regular Specifications. *arXiv preprint arXiv:2103.02226* 2021.

10. Ramasubramanian B, Niu L, Clark A, Bushnell L, Poovendran R. Linear temporal logic satisfaction in adversarial environments using secure control barrier certificates. In: Springer. ; 2019: 385–403.

11. He B, Lee J, Topcu U, Sentis L. BP-RRT: Barrier pair synthesis for temporal logic motion planning. In: IEEE. ; 2020: 1404–1409.

12. Nilsson P, Ames AD. Barrier functions: Bridging the gap between planning from specifications and safety-critical control. In: IEEE. ; 2018: 765–772.

13. Bechlioulis CP, Rovithakis GA. A low-complexity global approximation-free control scheme with prescribed performance for unknown pure feedback systems. *Automatica* 2014; 50(4): 1217–1226.

14. Büchi JR. On a decision method in restricted second order arithmetic. In: Springer. 1990 (pp. 425–435).

15. Lindemann L, Verginis CK, Dimarogonas DV. Prescribed performance control for signal temporal logic specifications. In: IEEE. ; 2017: 2997–3002.

16. Lindemann L, Dimarogonas DV. Feedback control strategies for multi-agent systems under a fragment of signal temporal logic tasks. *Automatica* 2019; 106: 284–293.

17. Verginis CK, Dimarogonas DV. Timed abstractions for distributed cooperative manipulation. *Autonomous Robots* 2018; 42(4): 781–799.

18. Schürmann B, Althoff M. Optimizing sets of solutions for controlling constrained nonlinear systems. *IEEE Transactions on Automatic Control* 2020; 66(3): 981–994.

19. Rabin MO. Decidability of second-order theories and automata on infinite trees. *Transactions of the american Mathematical Society* 1969; 141: 1–35.

20. Streett RS. Propositional dynamic logic of looping and converse is elementarily decidable. *Information and control* 1982; 54(1-2): 121–141.

21. Muller DE. Infinite sequences and finite machines. In: IEEE Computer Society. ; 1963: 3–16.

22. Duret-Lutz A, Lewkowicz A, Fauchille A, Michaud T, Renault E, Xu L. Spot 2.0—a framework for LTL and omega-automata manipulation. In: Springer. ; 2016: 122–129.

23. Gastin P, Oddoux D. Fast LTL to Büchi automata translation. In: Springer. ; 2001: 53–65.

24. Baier C, Katoen JP, Larsen KG. *Principles of model checking*. MIT press . 2008.

25. Klein J. ltl2dstar-LTL to deterministic Streett and Rabin automata. 2007.

26. De Giacomo G, Vardi MY. Linear temporal logic and linear dynamic logic on finite traces. In: Association for Computing Machinery. ; 2013: 854–860.

27. Henriksen JG, Jensen J, Jørgensen M, et al. Mona: Monadic second-order logic in practice. In: Springer. ; 1995: 89–110.

28. Wongpiromsarn T, Topcu A. Automata theory meets barrier certificates: Temporal logic verification of nonlinear systems. *IEEE Transactions on Automatic Control* 2015; 61(11): 3344–3355.

29. Russell SJ, Norvig P. *Artificial Intelligence: A Modern Approach*. Pearson Education. 2 ed. 2003.

30. Liu Y, Zhu JJ, Williams II RL, Wu J. Omni-directional mobile robot controller based on trajectory linearization. *Robotics and autonomous systems* 2008; 56(5): 461–479.

□