2025 IEEE 64th Conference on Decision and Control (CDC)
December 10-12, 2025. Rio de Janeiro, Brazil

Control Synthesis for Multiple Reach-Avoid Tasks via Hamilton-Jacobi Reachability Analysis

Yu Chen, Shaoyuan Li and Xiang Yin

2025 IEEE 64th Conference on Decision and Control (CDC) | 979-8-3315-2627-6/25/$31.00 ©2025 IEEE | DOI: 10.1109/CDC57313.2025.11312352

***Abstract*— We investigate the control synthesis problem for continuous-time control-affine systems under a class of *multiple reach-avoid* (MRA) tasks. Specifically, the MRA task requires the system to reach a series of target regions in a specified order while satisfying state constraints between each pair of target arrivals. This problem is more challenging than standard reach-avoid tasks, as it requires considering the feasibility of future reach-avoid tasks during the planning process. To solve this problem, we define a series of value functions by solving a cascade of time-varying reach-avoid problems characterized by Hamilton-Jacobi variational inequalities. We prove that the super-level set of the final value function computed is exactly the feasible set of the MRA task. Additionally, we demonstrate that the control law can be effectively synthesized by ensuring the non-negativeness of the value functions over time. The effectiveness of the proposed approach is illustrated through two case studies on robot planning problems.**

I. Introduction

Formal controller synthesis is a fundamental problem in autonomous systems, including autonomous driving [1], mobile robots [2], [3] and industrial manufacturing systems [4]. The objective is to algorithmically design a controller that guarantees the satisfaction of a given specification. Over the years, formal controller synthesis with provable guarantees has been extensively studied for various system classes and requirement types, resulting in a wide array of methodologies [5]–[7]. As autonomous systems grow increasingly complex, ensuring their safe and efficient operation demands computationally efficient synthesis techniques.

Among formal specifications, *reachability* is one of the most fundamental tasks. It requires the system to reach a target state either eventually or within a predefined time horizon. In many applications, additional constraints are imposed, such as avoiding obstacles or remaining within the target region once reached. These tasks are referred to as reach-avoid [8]–[13] or reach-avoid-stay problems [14]–[16]. Particularly, reach-avoid tasks are not only important on their own but also serve as fundamental building blocks for more complex specifications. For example, in linear temporal logic specifications, a system must sequentially reach specific labeled regions according to automaton states in order to satisfy desired temporal-spatial behavior [17].

In formal controller synthesis for reachability-based specifications, the key challenge lies in analyzing reachability based on system dynamics. This problem can be addressed using Hamilton-Jacobi Reachability (HJR), which formulates it as a Hamilton-Jacobi partial differential equation (PDE) [18] and represents the set of interest as the level set of the PDE solution. The HJR method provides a theoretical foundation for synthesizing controllers for reachability [19] and reach-avoid [9] tasks in dynamic systems under disturbances. In recent years, it has been applied to a wide range of complex reach-avoid problems such as multiplayer reach-avoid games [20] and multi-vehicle path planning [21].

In this paper, we address the control synthesis problem for control-affine systems under a new class of tasks called *multiple reach-avoid* (MRA) task. An MRA task requires the system to reach a series of targets in a predefined sequence while satisfying state constraints between each pair of consecutive arrivals. This problem is more challenging than standard reach-avoid tasks, as it necessitates ensuring the feasibility of future reach-avoid sub-tasks during the planning process. To solve this problem, we first prove that the feasible set of the MRA task can be exactly characterized as the super-level set of a specific function. This function is computed by solving a sequence of time-varying reach-avoid HJR problems, where the feasible set of future sub-tasks is treated as a dynamic target. We then propose a control synthesis procedure for MRA tasks by enforcing the non-negativity of these value functions throughout the system's evolution. Finally, we demonstrate the effectiveness of our method through two robot planning case studies.

Our work is related to solving control synthesis problems under complex temporal requirements using HJR methods [22]–[26]. For instance, [22] computes the feasible sets of signal temporal logic tasks by recursively handling each temporal operator, while [23]–[26] introduce a temporal logic tree structure to heuristically guide the feasible set computation for linear temporal logic tasks. These approaches employ heuristic algorithms to account for temporal task dependencies, yielding only conservative approximations of the feasible set. In contrast, our work provides an exact functional representation of the proposed MRA task by treating the feasible set of future sub-tasks as a time-varying dynamic target. This approach offers new insights into precisely characterizing complex temporal dependencies through HJR methods.

II. Preliminaries

Notations: We denote by $\mathbb{R}$, $\mathbb{R}_{\geq 0}$ and $\mathbb{R}^n$ the set of all real numbers, non-negative real numbers and n-dimensional

This work was supported by the National Natural Science Foundation of China (62573291,62173226) and Science Center Program of National Natural Science Foundation of China under Grant 62188101.

Yu Chen, Shaoyuan Li and Xiang Yin are with School of Automation and Intelligent Sensing, Shanghai Jiao Tong University, Shanghai 200240, China. {yuchen26, syli, yinxiang}@sjtu.edu.cn.

real vectors, respectively. For a positive integer i, we define $[i] = \{1, 2, \ldots, i\}$.

A. System and trajectories

We consider a control-affine system described by

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

where $x \in \mathbb{R}^n$ is the system state, $u \in \mathcal{U} \subseteq \mathbb{R}^m$ is the control input with compact input space $\mathcal{U}$, and $f : \mathbb{R}^n \to \mathbb{R}^n$, $g : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ are bounded and Lipschitz continuous.

The set of admissible control functions over the interval $[t_0, t_1]$ is defined as $\mathbb{U}_{[t_0,t_1]} := \{\mathbf{u} : [t_0, t_1] \to \mathcal{U} \mid \mathbf{u}(\cdot) \text{ is measurable}\}$. For an initial state $x \in \mathbb{R}^n$ and time t, under a control function $\mathbf{u} \in \mathbb{U}_{[t,s]}$, the system's evolution is determined by the unique continuous trajectory $\xi^{\mathbf{u}}_{x,t} : [t, s] \to \mathbb{R}^n$ satisfying $\xi^{\mathbf{u}}_{x,t}(t) = x$ and $\dot{\xi}^{\mathbf{u}}_{x,t}(\tau) = f(\xi^{\mathbf{u}}_{x,t}(\tau)) + g(\xi^{\mathbf{u}}_{x,t}(\tau))\mathbf{u}(\tau)$, a.e. $\tau \in [t, s]$, where a.e. (almost everywhere) means the differential equation holds except on a set of Lebesgue measure zero.

B. Time-Varying Reachability

In a reach-avoid task, the system needs to reach a target region while remaining within a safe region throughout its trajectory. Formally, a (time-varying) reach-avoid task is defined as a tuple $(\mathcal{T}, \mathcal{G})$, where $\mathcal{T}, \mathcal{G} \subseteq \mathbb{R}^n \times [t_0, t_1]$ are time-augmented sets representing the target region and the safe region, respectively. For each $\star \in \{\mathcal{T}, \mathcal{G}\}$ and time instant $t \in [t_0, t_1]$, we denote the state set at time t by $\star(t) = \{x \in \mathbb{R}^n \mid (x, t) \in \star\}$. Given $t_0 \le t \le t_1$, the *feasible set* of the reach-avoid task $(\mathcal{T}, \mathcal{G})$ is defined as

$$\mathsf{RA}(t, t_1, \mathcal{T}, \mathcal{G}) = \tag{2}$$
$$\left\{ x \in \mathbb{R}^n \;\middle|\; \begin{array}{c} \exists \mathbf{u} \in \mathbb{U}_{[t,t_1]},\, \exists s \in [t, t_1], \xi^{\mathbf{u}}_{x,t}(s) \in \mathcal{T}(s), \\ \forall s' \in [t, s],\, \xi^{\mathbf{u}}_{x,t}(s') \in \mathcal{G}(s') \end{array} \right\}.$$

For each $\star \in \{\mathcal{T}, \mathcal{G}\}$, we assume that there is Lipschitz continuous function $h_\star : \mathbb{R}^n \times [t_0, t_1] \to \mathbb{R}$ such that $h_\star(x, t) \ge 0$ iff $x \in \star(t)$. Then we define a value function by: $\forall x \in \mathbb{R}^n, t \in [t_0, t_1]$, we have

$$h_{\mathsf{RA}}(x, t, h_{\mathcal{T}}, h_{\mathcal{G}}) = \tag{3}$$
$$\sup_{\mathbf{u}\in\mathbb{U}_{[t,t_1]}} \max_{s\in[t,t_1]} \min \left\{ h_{\mathcal{T}}(\xi^{\mathbf{u}}_{x,t}(s), s),\, \min_{s'\in[t,s]} h_{\mathcal{G}}(\xi^{\mathbf{u}}_{x,t}(s'), s') \right\}.$$

According to [9], we know that $x \in \mathsf{RA}(t, t_1, \mathcal{T}, \mathcal{G})$ iff $h_{\mathsf{RA}}(x, t, h_{\mathcal{T}}, h_{\mathcal{G}}) \ge 0$. Moreover, the value function h_{RA} is the viscosity solution of following Hamilton-Jacobi variational inequality (VI):

$$\min \left\{ \begin{array}{c} \max \left\{ \frac{\partial h_{\mathsf{RA}}(x,t)}{\partial t} + \mathsf{Ham}(x, t), h_{\mathcal{T}}(x, t) - h_{\mathsf{RA}}(x, t) \right\}, \\ h_{\mathcal{G}}(x, t) - h_{\mathsf{RA}}(x, t) \end{array} \right\}$$
$$= 0, \tag{4}$$

where $h_{\mathsf{RA}}(x, t_1) = \min\{h_{\mathcal{T}}(x, t_1), h_{\mathcal{G}}(x, t_1)\}$ is the boundary condition and

$$\mathsf{Ham}(x, t) = \max_{u\in\mathcal{U}} \frac{\partial h_{\mathsf{RA}}(x, t)}{\partial x}(f(x) + g(x)u). \tag{5}$$

The reader is referred to [9] for more details on solving time-varying reachability problem by HJR method.

III. Problem Formulation

In this work, we introduce a new type of reach-avoid task called the *multiple reach-avoid* (MRA) task. In an MRA task, the system needs to visit *a sequence of* target regions in a prescribed order while satisfying state constraints between consecutive target arrivals.

Formally, let $\mathbb{T} = (\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_N)$ and $\mathbb{G} = (\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_N)$ denote sequences of N target regions and N safe regions, respectively, where for each $i \in [N]$, the sets satisfy $\mathcal{T}_i, \mathcal{G}_i \subseteq \mathbb{R}^n$. Let $t_0, t_1 \in [0, T]$ represent the start time and end time of the entire task. The multiple reach-avoid task is then defined by the 4-tuple $\Phi = (t_0, t_1, \mathbb{T}, \mathbb{G})$. For simplicity, we denote an MRA task by $\Phi^{[t_0,t_1]}$ when the target and safe regions are clear from the context.

Given an initial state $x_0 \in \mathbb{R}^n$ and a control function $\mathbf{u} \in \mathbb{U}_{[t_0,t_1]}$, the generated trajectory $\xi^{\mathbf{u}}_{x_0,t_0}$ is said to satisfy the MRA task $\Phi^{[t_0,t_1]}$, denoted by $\xi^{\mathbf{u}}_{x_0,t_0} \models \Phi^{[t_0,t_1]}$, if there exists a sequence of time instants $t_0 = \tau_0 \le \tau_1 \le \tau_2 \le \cdots \le \tau_N \le t_1$ such that

$$\left[\forall i \in [N] : \xi^{\mathbf{u}}_{x_0,t_0}(\tau_i) \in \mathcal{T}_i\right] \wedge \left[\forall \tau \in [\tau_{i-1}, \tau_i] : \xi^{\mathbf{u}}_{x_0,t_0}(\tau) \in \mathcal{G}_i\right].$$

The feasible set of the MRA task Φ is defined as

$$\mathsf{MRA}(t_0, t_1, \mathbb{T}, \mathbb{G}) = \left\{ x \in \mathbb{R}^n \mid \exists \mathbf{u} \in \mathbb{U}_{[t_0,t_1]}, \xi^{\mathbf{u}}_{x,t_0} \models \Phi^{[t_0,t_1]} \right\}.$$

The MRA task control synthesis problem is stated as follow.

Problem 1 (MRA Control Synthesis Problem): Given the control-affine system (1), an initial state $x_0 \in \mathbb{R}^n$, and an MRA task $\Phi = (0, T, \mathbb{T}, \mathbb{G})$:

(1) Decide whether $x_0 \in \mathsf{MRA}(0, T, \mathbb{T}, \mathbb{G})$.
(2) If feasible, synthesize a control function $\mathbf{u} \in \mathbb{U}_{[0,T]}$ such that the resulting trajectory $\xi^{\mathbf{u}}_{x_0,0}$ satisfies $\xi^{\mathbf{u}}_{x_0,0} \models \Phi$.

Remark 1: We conclude this section by discussing the motivation behind the MRA task framework. From a theoretical perspective, the MRA task represents a natural generalization of the standard reach-avoid problem. This extension introduces significant new challenges, as it requires ensuring the feasibility of future reach-avoid objectives during the current planning phase. Such temporal dependency issue does not exist in the single-target case. Furthermore, the MRA task is closely related to control synthesis for high-level specifications, such as Linear Temporal Logic (LTL) tasks. For instance, given an LTL specification φ, we can convert it into a finite-state automaton A_φ such that a system trajectory satisfies φ if and only if it induces an accepting state sequence in the automaton [27]. To achieve this, the system must sequentially reach specific labeled regions, which effectively reduces the problem to the proposed MRA task. This connection suggests that MRA tasks serve as a fundamental building block for solving more complex temporal logic control synthesis problem.

IV. Value function computation

In this section, we adopt the HJR method to compute the value function, whose super-level set represents the feasible set of the MRA task $\Phi^{[0,T]}$. If the initial state lies within

this feasible set, then the value function will later be used to ensure task satisfaction (as discussed in the next section). Note that the standard HJR method computes the feasible set only for a single reach-avoid task. We extend this approach to the MRA task by treating the feasible set of future tasks as a dynamic target. First, we assume that the target and safe regions can be expressed in terms of value functions, as formalized below.

Assumption 1: Let $\mathbb{T} = (\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_N)$ and $\mathbb{G} = (\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_N)$ be sequences of N (time-invariant) target regions and safe regions, respectively, where $\mathcal{T}_i, \mathcal{G}_i \subseteq \mathbb{R}^n$ for each $i \in [N]$. For each $i \in [N]$, we assume there exist Lipschitz continuous (with respect to x) functions

$$h_{\mathcal{T}_i}, h_{\mathcal{G}_i} \colon \mathbb{R}^n \to \mathbb{R}$$

that characterize the target region $\mathcal{T}_i$ and safe region $\mathcal{G}_i$, respectively, such that:

$$\big[x \in \mathcal{T}_i \Leftrightarrow h_{\mathcal{T}_i}(x) \geq 0\big] \ \wedge \ \big[x \in \mathcal{G}_i \Leftrightarrow h_{\mathcal{G}_i}(x) \geq 0\big]. \quad (6)$$

Our approach for computing the overall value function for the MRA task consists of the following steps:

1) **Initial Feasible Set Computation**: First, we compute the feasible set for a single-target reach-avoid task with target function $h_{\mathcal{T}_N}$ and safe function $h_{\mathcal{G}_N}$, leveraging existing results from [9] (formally stated in Lemma 1).
2) **Recursive Value Function Construction**: Next, we compute a new value function for a single-target reach-avoid task by treating: (i) the value function from the previous step and $h_{\mathcal{T}_{N-1}}$ as a combined *time-varying* target function, and (ii) $h_{\mathcal{G}_{N-1}}$ as the safe function. This value function represents the feasible set for first reaching $\mathcal{T}_{N-1}$ and then $\mathcal{T}_N$, while remaining in $\mathcal{G}_{N-1}$ and $\mathcal{G}_N$ before arriving at $\mathcal{T}_{N-1}$ and $\mathcal{T}_N$, respectively. A formal proof is provided in Proposition 1.
3) **Iterative Extension to Full MRA Task**: We repeat the above step recursively, treating the value function computed at each iteration as a time-varying target for the next. The remaining target $h_{\mathcal{T}_i}$ and safe function $h_{\mathcal{G}_i}$ (not yet considered) are incorporated as the new target and safe regions, respectively. Upon including all targets, the super-level set of the final value function yields the feasible set of the MRA task, as proven in Theorem 1.

To formally establish our results, we define $\mathbb{T}_i = (\mathcal{T}_{N-i+1}, \ldots, \mathcal{T}_N)$ as the sequence of the last i target sets, i.e., $\mathbb{T}_1$ means that one only needs to achieve the last task $\mathcal{T}_N$. Similarly, we define $\mathbb{G}_i = (\mathcal{G}_{N-i+1}, \ldots, \mathcal{G}_N)$. Given an MRA task $\Phi^{[0,T]} = (0, T, \mathbb{T}, \mathbb{G})$, for any $0 \leq t_0 \leq T$, we define the truncated MRA task as $\Phi_i^{[t_0,T]} = (t_0, T, \mathbb{T}_i, \mathbb{G}_i)$, which considers only the last i target and safe regions with the start time shifted to t_0.

We first directly use result in [9] to compute the feasible set of the task $\Phi_1^{[0,T]} = (0, T, \mathbb{T}_1, \mathbb{G}_1)$.

Lemma 1: Let $h_{\mathsf{RA}}^{\Phi_1}(x,t)$ be the viscosity solution of HJ-VI in (4) for target function $h_{\mathcal{T}}^{\Phi_1}$ and safe function $h_{\mathcal{G}}^{\Phi_1}$ defined by: for any$(x,t) \in \mathbb{R}^n \times [0,T]$, we have

$$\begin{cases} h_{\mathcal{T}}^{\Phi_1}(x,t) = h_{\mathcal{T}_N}(x) \\ h_{\mathcal{G}}^{\Phi_1}(x,t) = h_{\mathcal{G}_N}(x) \end{cases}.$$

Then for any $(x,t) \in \mathbb{R}^n \times [0,T]$, it holds that

$$h_{\mathsf{RA}}^{\Phi_1}(x,t) \geq 0 \ \Leftrightarrow \ (\exists \mathbf{u} \in \mathbb{U}_{[t,T]})[\xi_{x,t}^{\mathbf{u}} \models \Phi_1^{[t,T]}]. \quad (7)$$

Next, we prove that the feasible set of MRA task Φ_{i+1} can be computed by regarding the feasible set of Φ_i as an additional time-varying target region.

Proposition 1: For each $i \in [N-1]$, let $h_{\mathsf{RA}}^{\Phi_i}$ be a function such that, for any $(x,t) \in \mathbb{R}^n \times [0,T]$, we have

$$h_{\mathsf{RA}}^{\Phi_i}(x,t) \geq 0 \Leftrightarrow \exists \mathbf{u} \in \mathbb{U}_{[t,T]}, \xi_{x,t}^{\mathbf{u}} \models \Phi_i^{[t,T]}.$$

Let $h_{\mathsf{RA}}^{\Phi_{i+1}}(x,t)$ be the viscosity solution of HJ-VI in (4) for target function $h_{\mathcal{T}}^{\Phi_{i+1}}$ and safe function $h_{\mathcal{G}}^{\Phi_{i+1}}(x,t)$ defined by: for any$(x,t) \in \mathbb{R}^n \times [0,T]$, we have

$$\begin{cases} h_{\mathcal{T}}^{\Phi_{i+1}}(x,t) = \min\{h_{\mathcal{T}_{N-i}}(x), h_{\mathsf{RA}}^{\Phi_i}(x,t)\} \\ h_{\mathcal{G}}^{\Phi_{i+1}}(x,t) = h_{\mathcal{G}_{N-i}}(x) \end{cases}.$$

Then for any $(x,t) \in \mathbb{R}^n \times [0,T]$, it holds that

$$h_{\mathsf{RA}}^{\Phi_{i+1}}(x,t) \geq 0 \ \Leftrightarrow \ (\exists \mathbf{u} \in \mathbb{U}_{[t,T]})[\xi_{x,t}^{\mathbf{u}} \models \Phi_{i+1}^{[t,T]}]. \quad (8)$$

Based on Proposition 1, we immediately have the following result for the feasible set of the entire MRA task.

Theorem 1: Let $h_{\mathsf{RA}}^{\Phi_N}(x,t)$ be the function satisfying (8). Then we have

$$h_{\mathsf{RA}}^{\Phi_N}(x,0) \geq 0 \Leftrightarrow x \in \mathsf{MRA}(0, T, \mathbb{T}, \mathbb{G}). \quad (9)$$

Remark 2: In fact, the result of Theorem 1 applies to general nonlinear systems without requiring the control-affine assumption. However, the control synthesis discussed in the next section involves computing control inputs in real time, which may become impractical for non-control-affine systems. Thus, when focusing solely on the feasible set of the MRA task, the system dynamics in (1) can be relaxed to $\dot{x} = f(x,u)$, where $f : \mathbb{R}^n \times \mathcal{U} \to \mathbb{R}^n$ is Lipschitz continuous in x for a fixed u.

V. Control synthesis procedure

Suppose that for the initial state $x_0 \in \mathbb{R}^n$, the MRA task is feasible, i.e., $x_0 \in \mathsf{MRA}(0, T, \mathbb{T}, \mathbb{G})$. The objective of this section is to show how to explicitly compute control function $\mathbf{u} \in \mathbb{U}_{[0,T]}$ such that $\xi_{x_0,0}^{\mathbf{u}} \models \Phi^{[0,T]}$.

Before delving into technical details, we outline our control synthesis procedure, presented in Algorithm 1. The approach consists of the following key steps. First, in the offline computation stage, we compute the value functions $h_{\mathsf{RA}}^{\Phi_i}$ for each subtask Φ_i, where $i = 1, 2, \ldots, N$. Then in the online execution stage, suppose that the controller is during the execution of the i-th reach-avoid task, it ensures feasibility for subsequent tasks as follows:

- **Current Functions Selection:** The current value function is set to $\mathtt{b} = h_{\mathsf{RA}}^{\Phi_{N-i+1}}$, which is the solution to HJ-VI defined in Proposition 1 (Line 3). Then we obtain the current target function $h_{\mathcal{T}}^{\mathtt{b}}$ which is the dynamic target function of HJ-VI computing the current value function $\mathtt{b}$ (Line 4). When this target function value is non-negative, the system is in target set and feasible set of future MRA task.

Algorithm 1: Control Synthesis Procedure

Input: Initial state $x_0 \in \mathbb{R}^n$ and value functions $h^{\Phi_i}_{\mathsf{RA}}$ for each $i = 1, 2, \ldots, N$

1 $x \leftarrow x_0$, $t \leftarrow 0$;
2 **for** $i = 1, 2, \ldots, N$ **do**
3 set current value function by $\mathtt{b}(x,t) \leftarrow h^{\Phi_{N-i+1}}_{\mathsf{RA}}(x,t)$;
4 set current target function $h^{\mathtt{b}}_{\mathcal{T}}(x,t)$ by (10);
5 set state-feedback control law $\mathbf{u}_{\mathtt{b}}(x,t)$ by (14);
6 **while** $h^{\mathtt{b}}_{\mathcal{T}}(x,t) < 0$ **do**
7 applied control input $\mathbf{u}_{\mathtt{b}}(x,t)$ and record new state x and time t;

- **Current Control Law Derivation:** A time-varying state-feedback control law $\mathbf{u}_{\mathtt{b}}(x,t)$ is derived from $\mathtt{b}$, ensuring $\mathtt{b}(x,t)$ remains non-negative over time.
- **Termination and Transition:** The control input $\mathbf{u}_{\mathtt{b}}(x,t)$ is applied until $h^{\mathtt{b}}_{\mathcal{T}}(x,t) \geq 0$, indicating the current target is achieved and future task is feasible (Lines 6–7). Once $h^{\mathtt{b}}_{\mathcal{T}}(x,t) \geq 0$ is satisfied, the target index i is updated in the for-loop, and the process repeats with the next value function until all targets are successfully reached.

We first introduce the target function $h^{\mathtt{b}}_{\mathcal{T}}$ in line 4. As mentioned above, the current value function $\mathtt{b}$ is the solution of one of HJ-VIs computed in last section. Then the current target function $h^{\mathtt{b}}_{\mathcal{T}}$ is the dynamic target function of this HJ-VI. Specifically, for $\mathtt{b} = h^{\Phi_i}_{\mathsf{RA}}$ and $(x,t) \in \mathbb{R}^n \times [0,T]$,

$$h^{\mathtt{b}}_{\mathcal{T}}(x,t) = \begin{cases} h_{\mathcal{T}_{N-i+1}}(x) & \text{if } i = 1 \\ \min\{h_{\mathcal{T}_{N-i+1}}(x), h^{\Phi_{i-1}}_{\mathsf{RA}}(x,t)\} & \text{otherwise} \end{cases}. \tag{10}$$

Now we explain how to derive the state-feedback control function $\mathbf{u}_{\mathtt{b}}$ in line 5. We require the assumption as below.

Assumption 2: The value function $h^{\Phi_i}_{\mathsf{RA}}$ is differentiable over $\mathbb{R}^n \times [0,T]$ for $i \in [N]$.

Remark 3: When the value functions associated with target $\mathcal{T}_i$ and constraint $\mathcal{G}_i$ are Lipschitz continuous, the reach-avoid value functions $h^{\Phi_i}_{\mathsf{RA}}(x,t)$ for $i = 1, 2, \ldots, N$ inherit this Lipschitz continuity and are consequently differentiable almost everywhere. In practical implementations where the differential of $\mathtt{b}(x,t)$ may not exist at certain points, we follow the approach in [28] by replacing the standard derivative in (11) with either: the *superdifferential* (for non-smooth maximization problems), or the *subdifferential* (for non-smooth minimization problems), as formally defined in [29, Chapter 3.2.5]. This generalization enables practical control synthesis even at non-differentiable points.

For any value function $\mathtt{b} : \mathbb{R}^n \times [0,T] \to \mathbb{R}$, we define

$$\mathtt{S}_{\mathtt{b}}(x,t) = \left\{ u \in \mathcal{U} \,\middle|\, \frac{\partial \mathtt{b}(x,t)}{\partial x}(f(x) + g(x)u) + \frac{\partial \mathtt{b}(x,t)}{\partial t} \geq 0 \right\}. \tag{11}$$

Clearly, if we apply control input from (11), then the time derivative of $\mathtt{b}$ satisfies

$$\dot{\mathtt{b}}(t) = \frac{\partial \mathtt{b}(x,t)}{\partial x}(f(x) + g(x)u(t)) + \frac{\partial \mathtt{b}(x,t)}{\partial t} \geq 0, \tag{12}$$

meaning $\mathtt{b}$ never decreases over time. For technical purposes, we further define the *feasible control input set* by

$$\tilde{\mathtt{S}}_{\mathtt{b}}(x,t) = \begin{cases} \mathtt{S}_{\mathtt{b}}(x,t) & \text{if } h^{\mathtt{b}}_{\mathcal{T}}(x,t) < \mathtt{b}(x,t) \\ \mathcal{U} & \text{otherwise} \end{cases}. \tag{13}$$

Now, let us consider the case when the current target value function $h^{\mathtt{b}}_{\mathcal{T}}$ is negative and the current value function $\mathtt{b}$ is non-negative. We have the following two key observation:

- By adopting control input in $\mathtt{S}_{\mathtt{b}}$ defined in (11), we can ensure that the value of $\mathtt{b}$ never decreases over time.
- Furthermore, the current target function will be non-negative at least at time T since we have $h^{\mathtt{b}}_{\mathcal{T}}(x,T) = \mathtt{b}(x,T) \geq 0$ by boundary condition of HJ-VI.

By combining the above two observations together, we know that the desired control objective, i.e., current target is reached and future task is feasible, can be achieved by adopting feasible control input set in Eq. (13). This intuition above is formally stated as follow.

Proposition 2: Given current value function $\mathtt{b}$ and current target function $h^{\mathtt{b}}_{\mathcal{T}}$, suppose that for $t_0 \in [0,T]$ and $x_0 \in \mathbb{R}^n$, we have $\mathtt{b}(x_0,t_0) \geq 0 > h^{\mathtt{b}}_{\mathcal{T}}(x_0,t_0)$. Let $\mathbf{u}_{\mathtt{b}} : \mathbb{R}^n \times [0,T] \to \mathcal{U}$ be a function such that $\mathbf{u}_{\mathtt{b}}$ is Lipschitz continuous in x and piecewise continuous in t, and

$$\mathbf{u}_{\mathtt{b}}(x,t) \in \tilde{\mathtt{S}}_{\mathtt{b}}(x,t), \forall x \in \mathbb{R}^n, t \in [0,T]. \tag{14}$$

Then there exists $\mathbf{u} \in \mathbb{U}_{[t_0,T]}$ s.t. $\mathbf{u}(\tau) = \mathbf{u}_{\mathtt{b}}(\xi^{\mathbf{u}}_{x_0,t_0}(\tau), \tau)$ for $\tau \in [t_0,T]$. Moreover, for such control function $\mathbf{u}$,

(a) $\exists t_1 \in [t_0,T] : h^{\mathtt{b}}_{\mathcal{T}}(\xi^{\mathbf{u}}_{x_0,t_0}(t_1), t_1) \geq \mathtt{b}(\xi^{\mathbf{u}}_{x_0,t_0}(t_1), t_1)$; and
(b) $\forall \tau \in [t_0,t_1] : \mathtt{b}(\xi^{\mathbf{u}}_{x_0,t_0}(\tau), \tau) \geq 0$.

One issue of feasible control input set $\tilde{\mathtt{S}}_{\mathtt{b}}$ in (13) is that $\mathtt{S}_{\mathtt{b}}$ may be empty, which makes it impossible to construct the state-feedback control function $\mathbf{u}_{\mathtt{b}}$. The following result guarantees that such situation never happens.

Proposition 3: Given the current value function $\mathtt{b} : \mathbb{R}^n \times [0,T] \to \mathbb{R}$, then for any $x \in \mathbb{R}^n, t \in [0,T]$, we have $\tilde{\mathtt{S}}_{\mathtt{b}}(x,t) \neq \emptyset$ where $\tilde{\mathtt{S}}_{\mathtt{b}}$ is defined in (13).

Remark 4: In practice, given value function $\mathtt{b}$, time instant $t \in [0,T]$, and state $x \in \mathbb{R}^n$, when the control constraint set $\mathcal{U}$ is a polytope, the following quadratic programming (QP) problem can be solved to select the control input:

$$\begin{aligned} \min_{u \in \mathcal{U}} \quad & u^\top Q(x,t) u + F(x,t)^\top u \\ \text{s.t.} \quad & \frac{\partial \mathtt{b}(x,t)}{\partial x}(f(x) + g(x)u) + \frac{\partial \mathtt{b}(x,t)}{\partial t} \geq 0, \end{aligned} \tag{15}$$

where $Q(x,t) \in \mathbb{R}^{m \times m}$ is a positive semi-definite matrix and $F(x,t) \in \mathbb{R}^m$. Under certain assumptions, as discussed in [30], [31], the solution $u^\star(x,t)$ of QP (15) can be Lipschitz continuous in x and piecewise continuous in t. Also, in many applications, a reference controller $u_{ref}(x,t) : \mathbb{R}^n \times [0,T] \to \mathcal{U}$ is already provided, which may perform well on other criteria, such as energy efficiency. In such cases, the QP objective in (15) can be modified to $(u - u_{ref}(x,t))^\top Q(u -$

$u_{ref}(x,t)$), ensuring minimal deviation from the reference controller while still satisfying the safety constraints. This approach allows the control input to meet the MRA task requirements while preserving the original performance as much as possible.

Proposition 2 shows that each target can be visited under corresponding control law in line 5. Therefore, by an inductive argument, we show that Algorithm 1 solves Problem 1 as formal statement below.

Theorem 2: Given the system in (1), the MRA task $\Phi = (0, T, \mathbb{T}, \mathbb{G})$, and initial state x_0 with $x_0 \in \mathsf{MRA}(0, T, \mathbb{T}, \mathbb{G})$, assume that $\mathbf{u}_{\mathtt{b}}$ in Proposition 2 can be found for each value function $\mathtt{b}$. Then there exists $0 = \tau_0 \leq \tau_1 \leq \tau_2 \leq \cdots \leq \tau_N \leq T$ such that the Algorithm 1 comes into i-th for-loop at time τ_{i-1} and the MRA task Φ is finished at time τ_N.

Remark 5: The function $\mathtt{b}$ represents the robustness of reaching a target while satisfying constraints. As previously discussed in (12), the value of function $\mathtt{b}$ will never decrease over time. However, to allow for a larger admissible control set, we may relax this condition and only require $\mathtt{b}$ to remain above a threshold $\beta > 0$. In this case, the control set (11) can be modified as

$$\mathtt{S}^m_{\mathtt{b}}(x,t) = \Big\{ u \in \mathcal{U} \;\Big|\; \frac{\partial \mathtt{b}(x,t)}{\partial x}(f(x)+g(x)u) + \frac{\partial \mathtt{b}(x,t)}{\partial t} \geq -\alpha(\mathtt{b}(x,t)-\beta) \Big\}, \tag{16}$$

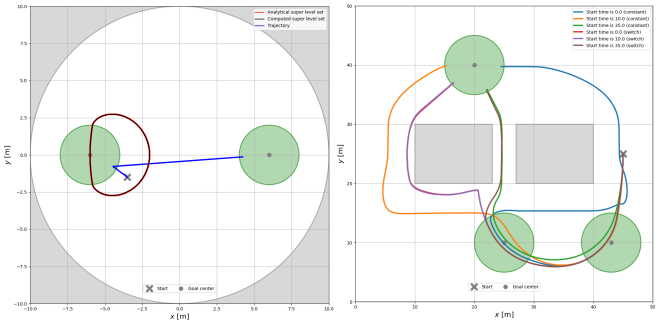
where $\alpha : \mathbb{R} \to \mathbb{R}$ is a strictly increasing, continuous function with $\alpha(0) = 0$. If $\mathtt{b}(x,t) \geq \beta$, then $-\alpha(\mathtt{b}(x,t)-\beta) \leq 0$ and $\mathtt{S}_{\mathtt{b}}(x,t) \subseteq \mathtt{S}^m_{\mathtt{b}}(x,t)$. Moreover, from [30], [31], the control law derived from (16) ensures $\mathtt{b}(x,t) \geq \beta$ for all time.

VI. Case Studies

The theoretical results for computing value functions and synthesizing controls have been implemented in Python. The HJR PDE is solved using a dynamic programming approach, which can leverage GPU acceleration to mitigate the curse of dimensionality. In this section, we demonstrate our algorithm through two case studies: a single integrator and a kinematic unicycle robot. For both examples, the offline value function computation takes only a few seconds on a desktop equipped with a single NVIDIA GeForce RTX 3090 GPU.

A. Single Integrator

We first consider a mobile robot model by a single-integrator as $\dot{x} = u$, where $x = (x_1, x_2) \in \mathbb{R}^2$ and $u = (u_1, u_2) \in \mathcal{U} \subseteq \mathbb{R}^2$ such that $\mathcal{U} = \{u \in \mathbb{R}^2 \mid \|u\|_2 \leq 1\}$. The regions of interest are defined by $R_1 = \{x \in \mathbb{R}^2 \mid (x_1+6)^2 + x_2^2 \leq 2^2\}$, $R_2 = \{x \in \mathbb{R}^2 \mid (x_1-6)^2 + x_2^2 \leq 2^2\}$, and $R_3 = \{x \in \mathbb{R}^2 \mid x_1^2 + x_2^2 \leq 10^2\}$. The MRA task is described by $\Phi = (0, 10, \mathbb{T} = (R_1, R_2), \mathbb{G} = (R_3, R_3))$, i.e., the robot should reach regions R_1 and R_2 in order while always staying in region R_3. We use signed distance function as value function of each target region and safe region. For example, for target R_1 and $t \in [0, 10]$, we define $h_{R_1}(x,t) = \sqrt{2^2 - (x_1+6)^2} - x_2^2$ if $x \in R_1$ and $h_{R_1}(x,t) = -\sqrt{(x_1+6)^2 + x_2^2} - 2^2$ if $x \notin R_1$.



(a) Case study 1. (b) Case study 2.

Fig. 1: Result of case studies.

The workspace for this case study is shown in Figure 1(a). The numerically computed feasible set of the MRA task, derived from Theorem 1, is represented by the black line. Due to the simplicity of the system dynamics and the MRA task, we also analytically compute the feasible set, depicted by the red line in Figure 1(a). The two sets overlap precisely, which further validates the correctness of Theorem 1.

For both case studies in this section, we use a system sampling time of 0.01s and the control input is computed by solving Program (15) with zero-order hold and a control update period of 0.1s. In this case study, the average solving time for (15) is 0.009s, making it sufficiently fast for online implementation. The robot's initial state is set to $x_0 = (-3.5, -1.5)$, and Figure 1(a) displays the trajectory, demonstrating successful completion of the MRA task.

B. Kinematic Unicycles

In this case, we consider a mobile robot modelled by kinematic unicycles dynamic. Specifically, the state $[x, y, \theta]^\top$ denotes x-position, y-position and angle, respectively. The control $[v, \omega]^\top \in \mathcal{U}$ denotes speed and angular velocity, respectively. The dynamic equation is given by $\dot{x} = v\cos\theta$, $\dot{y} = v\sin\theta$, $\dot{\theta} = \omega$ such that $\mathcal{U} = [0, 3] \times [-0.3, 0.3]$. There are three circle target regions with centers $(20, 40), (25, 10), (43, 10)$ and radius 5 denoted by R_1, R_2, R_3 respectively. Two rectangle obstacles G_1 and G_2 are defined by their lower left and upper right points $p_{11} = (10, 20)$, $p_{12} = (23, 30)$ and $p_{21} = (27, 20)$, $p_{22} = (40, 30)$, respectively. Targets and Obstacles are illustrated in Figure 1(b). The workspace of the robot is $G_f = [0, 50] \times [0, 50] \times [0, 2\pi]$. Then $G = G_f \setminus (G_1 \cup G_2)$ be the collision free state space of robot. The robot is required to reach R_3, R_2, R_1 in order with no collision within 60s, which can be expressed as the MRA task $\Phi = (0, 60, (R_3, R_2, R_1), (G, G, G))$.

While the MRA task itself only requires the robot to reach its targets before the final time, we may want to complete it as quickly as possible. To achieve this, we can modify the value function $\mathtt{b}$ from Algorithm 1 by defining a translated version $\mathtt{b}'$ such that $\mathtt{b}'(x,t) = \mathtt{b}(x, t+t_0)$ for some $t_0 \geq 0$. This translation effectively reduces the latest arrival time by t_0. In our case study, we evaluate three scenarios with $t_0 = 0$,

10, and 35. Additionally, we implement the control input set defined in (16) with $\beta = 1.2$. The control input sets in (11) and (16) are denoted by "*constant*" and "*switch*" respectively in Figure 1(b). We also consider a reference controller (v_{ref}, θ_{ref}) where $v_{ref} = k_v d_o(x)$ scales linearly with the distance $d_o(x)$ to obstacles and $\theta_{ref} = k_\theta \theta_e(x)$ adjusts based on the angle $\theta_e(x)$ to the current target center. Here, both k_v and k_θ are constant gains.

The robot has initial state $x_0 = (45, 25, 1.5\pi)$ and its trajectories under different control laws and parameters are shown in Figure 1(b). Compared with different t_0 under same control input set, the robot will choose a shorter path when using higher t_0. Compared with different control input sets under same t_0, when using control input set (16), robot will stay closer to obstacle. The reason is that when the distance between obstacles and robot is larger than pre-defined value $\beta = 1.2$, control input set (16) allows robot to approach obstacles if reference controller suggests so.

VII. Conclusion

In this paper, we addressed multiple reach-avoid task control synthesis problem. We illustrated how to accurately verify the feasibility of a multiple reach-avoid task by a series of value functions computed by existing Hamilton-Jacobi Reachability method. Moreover, we proposed a procedure to utilize the computed value functions for finishing the multiple reach-avoid task. The experimental results indicated that the proposed method can be applied to robot planning efficiently. Our work illustrated a new view on how to accurately characterize the complex temporal task by Hamilton-Jacobi Reachability method. In the future, we will explore the connection between multiple reach-avoid tasks and temporal logic task, e.g., LTL, and extend proposed method to control synthesis of more complex temporal logic tasks.

References

[1] N. Mehdipour, M. Althoff, R. D. Tebbens, and C. Belta, "Formal methods to comply with rules of the road in autonomous driving: State of the art and grand challenges," *Automatica*, vol. 152, p. 110692, 2023.

[2] B. Cui, F. Huang, S. Li, and X. Yin, "Robust temporal logic task planning for multirobot systems under permanent robot failures," *IEEE Trans. Control Systems Technology*, 2025.

[3] J. Zhao, K. Zhu, M. Feng, S. Li, and X. Yin, "No-regret path planning for temporal logic tasks in partially-known environments," *The International Journal of Robotics Research*, p. 02783649251315758, 2025.

[4] J. Campos, C. Seatzu, and X. Xie, *Formal Methods in Manufacturing*. CRC press, 2018.

[5] C. Belta and S. Sadraddini, "Formal Methods for Control Synthesis: An Optimization Perspective," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 2, no. 1, pp. 115–140, 2019.

[6] S. Liu, A. Trivedi, X. Yin, and M. Zamani, "Secure-by-construction synthesis of cyber-physical systems," *Annual Reviews in Control*, vol. 53, pp. 30–50, 2022.

[7] X. Yin, B. Gao, and X. Yu, "Formal synthesis of controllers for safety-critical autonomous systems: Developments and challenges," *Annual Reviews in Control*, vol. 57, p. 100940, 2024.

[8] K. Margellos and J. Lygeros, "Hamilton–jacobi formulation for reach–avoid differential games," *IEEE Trans. Automatic Control*, vol. 56, no. 8, pp. 1849–1861, 2011.

[9] J. F. Fisac, M. Chen, C. J. Tomlin, and S. S. Sastry, "Reach-avoid problems with time-varying dynamics, targets and constraints," in *HSCC*, pp. 11–20, 2015.

[10] Z. Zhou, J. Ding, H. Huang, R. Takei, and C. Tomlin, "Efficient path planning algorithms in reach-avoid problems," *Automatica*, vol. 89, pp. 28–36, 2018.

[11] B. Xue, N. Zhan, M. Fränzle, J. Wang, and W. Liu, "Reach-avoid verification based on convex optimization," *IEEE Trans. Automatic Control*, vol. 69, no. 1, pp. 598–605, 2024.

[12] Y. Chen, Y. Li, S. Li, and X. Yin, "Distributionally robust control synthesis for stochastic systems with safety and reach-avoid specifications," *arXiv:2501.03137*, 2025.

[13] Y. Chen, S. Li, and X. Yin, "On the construction of barrier certificate: A dynamic programming perspective," *arXiv:2507.17222*, 2025.

[14] Y. Meng, Y. Li, M. Fitzsimmons, and J. Liu, "Smooth converse lyapunov-barrier theorems for asymptotic stability with safety constraints and reach-avoid-stay specifications," *Automatica*, vol. 144, p. 110478, 2022.

[15] Y. Meng and J. Liu, "Stochastic lyapunov-barrier functions for robust probabilistic reach-avoid-stay specifications," *IEEE Trans. Automatic Control*, vol. 69, no. 8, pp. 5470–5477, 2024.

[16] R. Das and P. Jagtap, "Prescribed-time reach-avoid-stay specifications for unknown systems: A spatiotemporal tubes approach," *IEEE Control Systems Letters*, vol. 8, pp. 946–951, 2024.

[17] B. Zhong, M. Zamani, and M. Caccamo, "Formal synthesis of controllers for uncertain linear systems against-regular properties: A set-based approach," *IEEE Trans. Automatic Control*, vol. 69, no. 1, pp. 214–229, 2024.

[18] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, "Hamilton-jacobi reachability: A brief overview and recent advances," in *IEEE Conf. Decision and Control*, pp. 2242–2253, IEEE, 2017.

[19] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games," *IEEE Trans. Automatic Control*, vol. 50, no. 7, pp. 947–957, 2005.

[20] H. Huang, J. Ding, W. Zhang, and C. J. Tomlin, "A differential game approach to planning in adversarial scenarios: A case study on capture-the-flag," in *IEEE International Conf. Robotics and Automation*, pp. 1451–1456, 2011.

[21] M. Chen, S. Bansal, J. F. Fisac, and C. J. Tomlin, "Robust sequential trajectory planning under disturbances and adversarial intruder," *IEEE Trans. Control Systems Technology*, vol. 27, no. 4, pp. 1566–1582, 2018.

[22] M. Chen, Q. Tam, S. C. Livingston, and M. Pavone, "Signal temporal logic meets reachability: Connections and applications," in *International Workshop on the Algorithmic Foundations of Robotics*, pp. 581–601, Springer, 2018.

[23] Y. Gao, A. Abate, F. J. Jiang, M. Giacobbe, L. Xie, and K. H. Johansson, "Temporal logic trees for model checking and control synthesis of uncertain discrete-time systems," *IEEE Trans. Automatic Control*, vol. 67, no. 10, pp. 5071–5086, 2021.

[24] P. Yu, X. Tan, and D. V. Dimarogonas, "Continuous-time control synthesis under nested signal temporal logic specifications," *IEEE Trans. Robotics*, 2024.

[25] J. Verhagen, L. Lindemann, and J. Tumova, "Robust stl control synthesis under maximal disturbance sets," *arXiv:2404.05535*, 2024.

[26] F. J. Jiang, K. M. Arfvidsson, C. He, M. Chen, and K. H. Johansson, "Guaranteed completion of complex tasks via temporal logic trees and hamilton-jacobi reachability," *arXiv:2404.08334*, 2024.

[27] C. Belta, B. Yordanov, and E. A. Gol, *Formal methods for discrete-time dynamical systems*, vol. 89. Springer, 2017.

[28] J. J. Choi, D. Lee, K. Sreenath, C. J. Tomlin, and S. L. Herbert, "Robust control barrier–value functions for safety-critical control," in *IEEE Conf. Decision and Control (CDC)*, pp. 6814–6821, 2021.

[29] M. Bardi, I. C. Dolcetta, *et al.*, *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations*, vol. 12. Springer, 1997.

[30] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Trans. Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.

[31] L. Lindemann and D. V. Dimarogonas, "Control barrier functions for signal temporal logic tasks," *IEEE Control Systems Letters*, vol. 3, no. 1, pp. 96–101, 2018.