International Journal of Robotics Research
XX(X):1–34

DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

Motion Planning for Hybrid Dynamical Systems: Framework, Algorithm Template, and a Sampling-based Approach

Nan Wang and Ricardo G. Sanfelice[1]

Abstract

This paper focuses on the motion planning problem for the systems exhibiting both continuous and discrete behaviors, which we refer to as hybrid dynamical systems. Firstly, the motion planning problem for hybrid systems is formulated using the hybrid equation framework, which is general to capture most hybrid systems. Secondly, a propagation algorithm template is proposed that describes a general framework to solve the motion planning problem for hybrid systems. Thirdly, a rapidly-exploring random trees (RRT) implementation of the proposed algorithm template is designed to solve the motion planning problem for hybrid systems. At each iteration, the proposed algorithm, called HyRRT, randomly picks a state sample and extends the search tree by flow or jump, which is also chosen randomly when both regimes are possible. Through a definition of concatenation of functions defined on hybrid time domains, we show that HyRRT is probabilistically complete, namely, the probability of failing to find a motion plan approaches zero as the number of iterations of the algorithm increases. This property is guaranteed under mild conditions on the data defining the motion plan, which include a relaxation of the usual positive clearance assumption imposed in the literature of classical systems. The motion plan is computed through the solution of two optimization problems, one associated with the flow and the other with the jumps of the system. The proposed algorithm is applied to an actuated bouncing ball system and a walking robot system so as to highlight its generality and computational features.

Keywords

Hybrid Systems, Motion Planning, RRT Algorithm, Probabilistic Completeness

1 Introduction

Motion planning consists of finding a state trajectory and associated inputs, connecting the initial and final state while satisfying the system dynamics and a given safety criterion. The motion planning technology has been widely used to help robotics applications, such as autonomous driving systems Teng et al. (2023), satellites systems Rybus (2020), biped robots Huang et al. (2001), and quadrotors Liu et al. (2017), to complete complicated tasks. Particularly, the system dynamics considered in the motion planning not only depends on the mechanical design of the robot, but also depends on the internal logic/timer to resolve the task, and the interaction between the robot systems and the environment, et.al. The former usually leads to a continuous dynamics while the later usually leads to a discrete dynamics. For example, the *continuous* evolution of the states of the wheeled vehicle can be modeled by either its kinematic model Rajamani (2011) or its dynamic model Gillespie (2021). On the other hand, if a motion planner is designed for a robot to move to the highest floor in a building, then a logic variable representing the floor number should be employed and updated in a *discrete* manner Branicky et al. (2003b). In addition, a motion planner to plan the motion of consecutive steps of a biped robot should consider *discrete* changes over the foot's speed when an impact between the foot and the ground occurs.

However, in certain motion planning tasks, the states can both evolve continuously and, from time to time, execute some discrete changes. For instance, in the context of a collision-resilient multicopter system in Zha and Mueller (2021), a RRT-type motion planner is devised. The multicopter's position and velocity states evolve continuously in open space, yet exhibit discrete changes upon collision with a wall. In such instances, neither a purely continuous-time nor discrete-time model suffices to accurately capture system behavior. Instead, a hybrid system model proves essential, encompassing not only purely continuous or discrete-time systems but also those manifesting both continuous and discrete behaviors. Consequently, employing a general hybrid system model renders motion planning problems more comprehensive than those confined to purely continuous-time or discrete-time systems.

Research has been conducted to study motion planning for some specific classes of hybrid systems. Notably, in

[1]Hybrid Systems, Laboratory, Department of Electrical and Computer Engineering, University of California, Santa Cruz, USA

Corresponding author:
[1]Nan Wang, Department of Electrical and Computer Engineering, University of California, Santa Cruz, USA.
Email: nanwang@ucsc.edu

Wu et al. (2020) and Branicky et al. (2003a), two RRT-type motion planning algorithms are introduced to address motion planning problems within the realms of continuous-time hybrid systems. These systems feature continuous states evolving continuously alongside discrete states (modes) that transition among a finite set of feasible modes. However, they do not encompass hybrid systems where states evolve continuously and intermittently execute jumps. This paper focuses on motion planning problems for hybrid systems modeled as hybrid equations Sanfelice (2021). In this modeling framework, differential and difference equations with constraints are used to describe the continuous and discrete behavior of the hybrid system, respectively. This general hybrid system framework can capture most hybrid systems emerging in robotic applications, not only the class of hybrid systems considered in Wu et al. (2020) and Branicky et al. (2003a), but also systems with memory states, timers, impulses, and constraints. From the authors' best knowledge, it is the first time that the motion planning problem is formulated for this general hybrid system framework.

Motion planning problems for purely continuous-time systems (known as *kinodynamic planning*) and purely discrete-time systems have been extensively explored in the literature; see, e.g., LaValle (2006). Most existing motion planning algorithms, such as RRT algorithm LaValle and Kuffner Jr (2001), incrementally construct a search tree in the state space and seek for a path in the search tree connecting the initial and final states. Operations like concatenation, well-defined for purely continuous-time and purely discrete-time systems, are crucial in these algorithms and are extensively documented in the literature; see, for instance, (LaValle 2006, Chapter 14.3). However, defining such operations for trajectories of hybrid systems poses a significant challenge because of their considerably more complicated domain structure. In fact, trajectories of hybrid systems may exhibit various behaviors: 1) evolves purely continuously, 2) exhibits jumps all the time, 3) evolves continuous and exhibits one or multiple jumps at times, or 4) exhibits Zeno behavior. To the authors' best knowledge, there is no existing work formulating such operations for hybrid systems, let alone their properties. Within this general framework, mathematical definitions of operations for hybrid systems are formulated, along with theoretical analysis of those operations for motion planning algorithms for hybrid systems. In addition, a high-level motion planning algorithm template that utilizes those operations is proposed to help design and validate the motion planning algorithms to solve the motion planning problems for hybrid systems.

Various planning algorithms have been developed to solve motion planning problems, from graph search algorithms Likhachev et al. (2005) to artificial potential field methods Khatib (1986). A main drawback of graph search algorithms is that the number of vertices grows exponentially as the dimension of states grows, which makes computing motion plans inefficient for high-dimensional systems. The artificial potential field method suffers from getting stuck at local minimum. Arguably, the most successful algorithm to solve motion planning problems for purely continuous-time systems and purely discrete-time systems is the sampling-based RRT algorithm LaValle (1998). This algorithm incrementally constructs a tree of state trajectories toward random samples in the state space. Similar to graph search algorithms, RRT suffers from the curse of dimensionality, but, in practice, achieves rapid exploration in solving high-dimensional motion planning problems Cheng (2005). Compared with the artificial potential field method, RRT is probabilistically complete LaValle and Kuffner Jr (2001), which means that the probability of failing to find a motion plan converges to zero, as the number of samples approaches infinity. While RRT searches for a feasible motion plan without considering its quality, an asymptotically optimal RRT* algorithm is introduced in Karaman and Frazzoli (2011), assuming the existence of a steering function that connects two states while satisfying differential constraints. However, finding such a steering function is challenging in practice. On the other hand, the stable sparse RRT algorithm, introduced in Li et al. (2016), does not require a steering function and is guaranteed to be asymptotically near-optimal. In recent years, machine learning technology has made significant strides in the motion planning field. In Qureshi et al. (2020), a learning-based neural planner is proposed, which incorporates existing RRT-type algorithms to generate samples in the subspace that is most likely to contain motion plans.

While RRT algorithms have been used to solve motion planning problems for purely continuous-time systems LaValle and Kuffner Jr (2001) and purely discrete-time systems Branicky et al. (2003b), fewer efforts have been devoted to applying RRT-type algorithms to solve motion planning problems for systems with combined continuous and discrete behavior. In Branicky et al. (2003a), a hybrid RRT algorithm is designed for a specific class of hybrid systems. While Branicky et al. (2003a) uses graph vertices as samples, Solovey et al. (2016) proposes an RRT algorithm for a geometrically embedded discrete graph. Following the algorithm template proposed in this paper, for this broad class of hybrid systems, an RRT-type motion planning algorithm is designed. The proposed algorithm, called HyRRT, incrementally constructs search trees, rooted in the initial state set and toward the random samples. At first, HyRRT draws samples from the state space. Then, it selects the vertex such that the state associated with this vertex has minimal distance to the sample. Next, HyRRT propagates the state trajectory from the state associated with the selected vertex. It is established that, under mild assumptions, HyRRT is probabilistically complete. To the authors' best knowledge, HyRRT is the first RRT-type algorithm for systems with hybrid dynamics that is proved to be probabilistically complete. The proposed algorithm is applied to an actuated bouncing ball system and a walking robot system so as to assess its capabilities.

The remainder of the paper is structured as follows. Section 2 presents notation and preliminaries. Section 3 presents the problem statement and introduction of applications. Section 4 defines the operations heavily used in the algorithms and presents a general framework template for the motion planning algorithms for hybrid systems. Section 5 presents the HyRRT algorithm. Section 6 presents the analysis of the probabilistic completeness of HyRRT algorithm. Section 7 presents the illustration of HyRRT in examples.

2 Notation and Preliminaries

2.1 Notation

The real numbers are denoted as $\mathbb{R}$ and its nonnegative subset is denoted as $\mathbb{R}_{\geq 0}$. The set of nonnegative integers is denoted as $\mathbb{N}$. The notation $\operatorname{int} I$ denotes the interior of the interval I. The notation $\overline{S}$ denotes the closure of the set S. The notation ∂S denotes the boundary of the set S. Given sets $P \subset \mathbb{R}^n$ and $Q \subset \mathbb{R}^n$, the Minkowski sum of P and Q, denoted as $P+Q$, is the set $\{p+q : p \in P, q \in Q\}$. The notation $|\cdot|$ denotes the Euclidean norm. Given a point $x \in \mathbb{R}^n$ and a subset $S \subset \mathbb{R}^n$, the distance between x and S is denoted $|x|_S := \inf_{s \in S} |x-s|$. The notation $\mathbb{B}$ denotes the closed unit ball of appropriate dimension in the Euclidean norm. The probability of the probabilistic event[1] M is denoted as $\text{Prob}(M)$. Given a set S, the notation $\mu(S)$ denotes its Lebesgue measure. The Lebesgue measure of the n-th dimensional unit ball, denoted ζ_n, is such that

$$\zeta_n := \begin{cases} \dfrac{\pi^k}{k!} & \text{if } n = 2k, k \in \mathbb{N} \\ \dfrac{2(k!)(4\pi)^k}{(2k+1)!} & \text{if } n = 2k+1, k \in \mathbb{N}; \end{cases} \tag{1}$$

see Gipple (2014).

2.2 Preliminaries

Following Sanfelice (2021), a hybrid system $\mathcal{H}$ with inputs is modeled as

$$\mathcal{H} : \begin{cases} \dot{x} = f(x,u) & (x,u) \in C \\ x^+ = g(x,u) & (x,u) \in D \end{cases} \tag{2}$$

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ is the input, $C \subset \mathbb{R}^n \times \mathbb{R}^m$ represents the flow set, $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ represents the flow map, $D \subset \mathbb{R}^n \times \mathbb{R}^m$ represents the jump set, and $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ represents the jump map. The continuous evolution of x is captured by the flow map f. The discrete evolution of x is captured by the jump map g. The flow set C collects the points where the state can evolve continuously. The jump set D collects the points where jumps can occur.

Given a flow set C, the set

$$U_C := \{u \in \mathbb{R}^m : \exists x \in \mathbb{R}^n \text{ such that } (x,u) \in C\} \tag{3}$$

includes all possible input values that can be applied during flows. Similarly, given a jump set D, the set

$$U_D := \{u \in \mathbb{R}^m : \exists x \in \mathbb{R}^n \text{ such that } (x,u) \in D\} \tag{4}$$

includes all possible input values that can be applied at jumps. These sets satisfy $C \subset \mathbb{R}^n \times U_C$ and $D \subset \mathbb{R}^n \times U_D$. Given a set $K \subset \mathbb{R}^n \times U_\star$, where $\star$ is either C or D, we define

$$\Pi_\star(K) := \{x : \exists u \in U_\star \text{ s.t. } (x,u) \in K\} \tag{5}$$

as the projection of K onto $\mathbb{R}^n$, and define

$$C' := \Pi_C(C) \tag{6}$$

and

$$D' := \Pi_D(D). \tag{7}$$

In addition to ordinary time $t \in \mathbb{R}_{\geq 0}$, we employ $j \in \mathbb{N}$ to denote the number of jumps of the evolution of x and u for $\mathcal{H}$ in (2), leading to hybrid time (t,j) for the parameterization of its solutions and inputs. The domain of a solution to $\mathcal{H}$ is given by a hybrid time domain as is defined as follows.

Definition 2.1. (Hybrid time domain). *A set $E \subset \mathbb{R}_{\geq 0} \times \mathbb{N}$ is a hybrid time domain if, for each $(T,J) \in E$, the set*

$$E \cap ([0,T] \times \{0,1,...,J\})$$

can be written in the form

$$\bigcup_{j=0}^{J} ([t_j, t_{j+1}] \times \{j\})$$

for some finite sequence of times $\{t_j\}_{j=0}^{J+1}$ satisfying $0 = t_0 \leq t_1 \leq t_2 \leq ... \leq t_{J+1} = T$.

Remark 2.2. *Note that this ordered sequence $\{t_j\}_{j=0}^{J}$ is only employed to characterized the interval $[t_j, t_{j+1}]$, ensuring that the left endpoint of each interval is always less than the right endpoint, and that the union of these intervals covers the entire interval $[0,T]$ on the positive real line.*

Given a compact hybrid time domain E, we denote the maximum of t and j coordinates of points in E as

$$\max_t E := \max\{t \in \mathbb{R}_{\geq 0} : \exists j \in \mathbb{N} \text{ such that } (t,j) \in E\} \tag{8}$$

and

$$\max_j E := \max\{j \in \mathbb{N} : \exists t \in \mathbb{R}_{\geq 0} \text{ such that } (t,j) \in E\}. \tag{9}$$

We denote the maximum of a hybrid time domain E as

$$\max E := (\max_t E, \max_j E).$$

The input to the hybrid systems is defined as a function on a hybrid time domain as follows.

Definition 2.3. (Hybrid input). *A function $\upsilon : \operatorname{dom} \upsilon \to \mathbb{R}^m$ is a hybrid input if $\operatorname{dom} \upsilon$ is a hybrid time domain and if, for each $j \in \mathbb{N}$, $t \mapsto \upsilon(t,j)$ is Lebesgue measurable and locally essentially bounded on the interval $I_\upsilon^j := \{t : (t,j) \in \operatorname{dom} \upsilon\}$.*

A hybrid arc describes the state trajectory of the system as is defined as follows.

Definition 2.4. (Hybrid arc). *A function $\phi : \operatorname{dom} \phi \to \mathbb{R}^n$ is a hybrid arc if $\operatorname{dom} \phi$ is a hybrid time domain and if, for each $j \in \mathbb{N}$, $t \mapsto \phi(t,j)$ is locally absolutely continuous on the interval $I_\phi^j := \{t : (t,j) \in \operatorname{dom} \phi\}$.*

Definition 2.5. (Types of hybrid arcs). *A hybrid arc ϕ is called*

- *compact if $\operatorname{dom} \phi$ is compact;*
- *nontrivial if $\operatorname{dom} \phi$ contains at least two points;*
- *purely discrete if nontrivial and $\operatorname{dom} \phi \subset \{0\} \times \mathbb{N}$;*
- *purely continuous if nontrivial and $\operatorname{dom} \phi \subset \mathbb{R}_{\geq 0} \times \{0\}$.*

The definition of solution pair to a hybrid system is given as follows.

Definition 2.6. (Solution pair to a hybrid system). *A hybrid input υ and a hybrid arc ϕ define a solution pair (ϕ, υ) to the hybrid system $\mathcal{H} = (C, f, D, g)$ if*

1) $(\phi(0,0), \upsilon(0,0)) \in \overline{C} \cup D$ *and* $\operatorname{dom}\phi = \operatorname{dom}\upsilon (= \operatorname{dom}(\phi, \upsilon))$.

2) For each $j \in \mathbb{N}$ *such that* I_ϕ^j *has nonempty interior* $\operatorname{int}(I_\phi^j)$, (ϕ, υ) *satisfies*

$$(\phi(t,j), \upsilon(t,j)) \in C$$

for all $t \in \operatorname{int} I_\phi^j$, *and*

$$\frac{d}{dt}\phi(t,j) = f(\phi(t,j), \upsilon(t,j))$$

for almost all $t \in I_\phi^j$.

3) For all $(t,j) \in \operatorname{dom}(\phi, \upsilon)$ *such that* $(t, j+1) \in \operatorname{dom}(\phi, \upsilon)$,

$$\begin{aligned} (\phi(t,j), \upsilon(t,j)) &\in D \\ \phi(t, j+1) &= g(\phi(t,j), \upsilon(t,j)). \end{aligned} \tag{10}$$

4) $(\phi(t,j), \upsilon(t,j)) \notin X_u$ *for each* $(t,j) \in \operatorname{dom}(\phi, \upsilon)$ *such that* $t + j \leq T + J$, *namely, the solution pair does not intersect with the unsafe set before its state trajectory reaches the final state set.*

Therefore, given sets X_0, X_f, *and* X_u, *and a hybrid system* $\mathcal{H}$ *as in (2) with data* (C, f, D, g), *a motion planning problem* $\mathcal{P}$ *is formulated as* $\mathcal{P} = (X_0, X_f, X_u, (C, f, D, g))$.

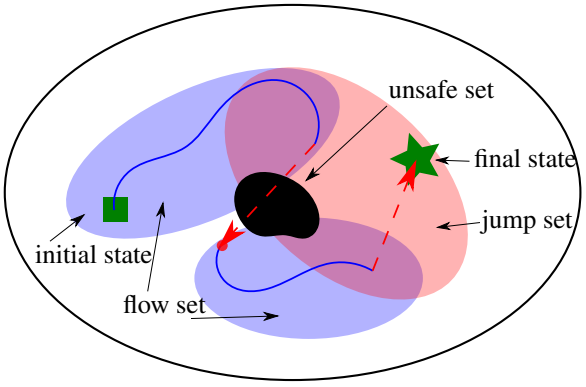


Figure 1. A motion plan to Problem 1. The green square denote the initial state set. The green star denotes the final state set. The blue region denotes the flow set. The red region denotes the jump set. The solid blue lines denote flow and the dotted red lines denote jumps in the motion plan.

Remark 2.7. *Definition 2.6 introduces the pair (ϕ, υ) as a solution to $\mathcal{H}$. It first requires that both ϕ and υ are defined on hybrid time domains. Item 1) specifies that the initial state-input pair must be in the union of the flow set and the jump set so either flow within C or jump from D can occur. Item 1) also imposes that ϕ and υ share the same hybrid time domain. Item 2) requires that, during flow, the state-input pair satisfies the continuous dynamics defined by f and the corresponding constraints captured by C. Item 3) requires that, at a jump, the new value of the state is given by the map g evaluated at the state-input pair from D. This definition is used in the next section to define the motion planning problem for hybrid systems studied in this paper.*

3 Problem Statement

The motion planning problem for hybrid systems studied in this paper is formulated as follows.

Problem 1. *Given a hybrid system $\mathcal{H}$ as in (2) with input $u \in \mathbb{R}^m$ and state $x \in \mathbb{R}^n$, the initial state set $X_0 \subset \mathbb{R}^n$, the final state set $X_f \subset \mathbb{R}^n$, and the unsafe set[2] $X_u \subset \mathbb{R}^n \times \mathbb{R}^m$, find a pair $(\phi, \upsilon) : \operatorname{dom}(\phi, \upsilon) \to \mathbb{R}^n \times \mathbb{R}^m$, namely,* a motion plan, *such that for some $(T, J) \in \operatorname{dom}(\phi, \upsilon)$, the following hold:*

1) $\phi(0,0) \in X_0$, namely, the initial state of the solution belongs to the given initial state set X_0;

2) (ϕ, υ) is a solution pair to $\mathcal{H}$ as defined in Definition 2.6;

3) (T, J) is such that $\phi(T, J) \in X_f$, namely, the solution belongs to the final state set at hybrid time (T, J);

There are some interesting special cases of Problem 1. For example, when $D = \emptyset$ ($C = \emptyset$) and C (respectively, D) is nonempty, $\mathcal{P}$ denotes the motion planning problem for purely continuous-time (respectively, discrete-time) systems with constraints. In this way, Problem 1 subsumes the motion planning problems for purely continuous-time and purely discrete-time system studied in LaValle and Kuffner Jr (2001) and LaValle (2006).

Example 3.1. (Actuated bouncing ball system). *Consider a ball bouncing on a fixed horizontal surface as is shown in Figure 2(a). The surface is located at the origin and, through control actions, is capable of affecting the velocity of the ball after the impact. The dynamics of the ball while in the air is given by*

$$\dot{x} = \begin{bmatrix} x_2 \\ -\gamma \end{bmatrix} =: f(x,u) \qquad (x,u) \in C \tag{11}$$

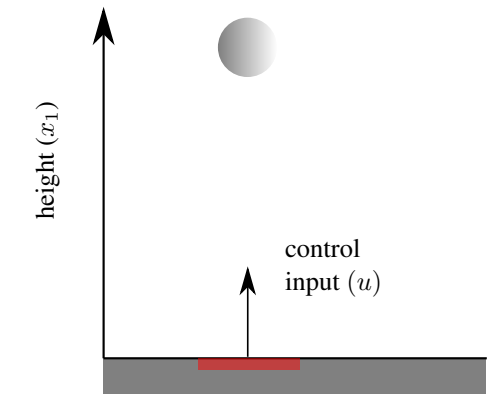
where $x := (x_1, x_2) \in \mathbb{R}^2$. The height of the ball is denoted by x_1. The velocity of the ball is denoted by x_2. The gravity constant is denoted by γ. Flow is allowed when the ball is above the surface. Hence, the flow set is

$$C := \{(x,u) \in \mathbb{R}^2 \times \mathbb{R} : x_1 \geq 0\}. \tag{12}$$

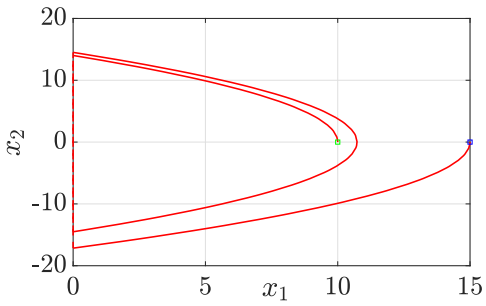
At every impact, and with control input equal to zero, the velocity of the ball changes from negative to positive while the height remains the same. The dynamics at jumps of the actuated bouncing ball system is given as

$$x^+ = \begin{bmatrix} x_1 \\ -\lambda x_2 + u \end{bmatrix} =: g(x,u) \qquad (x,u) \in D \tag{13}$$

where $u \geq 0$ is the input and $\lambda \in (0,1)$ is the coefficient of restitution. Jumps are allowed when the ball is on the surface



(a) The actuated bouncing ball system



(b) A motion plan to the sample motion planning problem for actuated bouncing ball system.

Figure 2. The actuated bouncing ball system in Example 3.1.

with nonpositive velocity. Hence, the jump set is

$$D := \{(x,u) \in \mathbb{R}^2 \times \mathbb{R} : x_1 = 0, x_2 \leq 0, u \geq 0\}. \quad (14)$$

The hybrid model of the actuated bouncing ball system is given by (2), where the flow map f is given in (11), the flow set C is given in (12), the jump map g is given in (13), and the jump set D is given in (14).

An example of a motion planning problem for the actuated bouncing ball system is as follows: using a bounded input signal, find a solution pair to (2) when the bouncing ball is released at a certain height with zero velocity and such that it reaches a given target height with zero velocity. To complete this task, not only the values of the input, but also the hybrid time domain of the input need to be planned properly such that the ball can reach the desired target. One such motion planning problem is given by defining the initial state set as $X_0 = \{(15,0)\}$, the final state set as $X_f = \{(10,0)\}$, the unsafe set as $X_u = \{(x,u) \in \mathbb{R}^2 \times \mathbb{R} : u \in (-\infty, 0] \cup [5,\infty)\}$. The motion planning problem $\mathcal{P}$ is given as $\mathcal{P} = (X_0, X_f, X_u, (C, f, D, g))$. We solve this motion planning problem later in this paper.

Example 3.2. (Walking robot). *The state x of the compass model of a walking robot is composed of the angle vector θ and the velocity vector ω Grizzle et al. (2001). The angle vector θ contains the planted leg angle θ_p, the swing leg angle θ_s, and the torso angle θ_t. The velocity vector ω contains the planted leg angular velocity ω_p, the swing leg angular velocity ω_s, and the torso angular velocity ω_t. The input u is the input torque, where u_p is the torque applied on the planted leg from the ankle, u_s is the torque applied on the swing leg from the hip, and u_t is the torque applied on the torso from the hip. The continuous dynamics of $x = (\theta, \omega)$ are obtained from the Lagrangian method and are given by $\dot{\theta} = \omega, \dot{\omega} = D_f(\theta)^{-1}(-C_f(\theta,\omega)\omega - G_f(\theta) + Bu) =: \alpha(x,u)$, where D_f and C_f are the inertial and Coriolis matrices, respectively, and B is the actuator relationship matrix. In Short and Sanfelice (2018), the input*

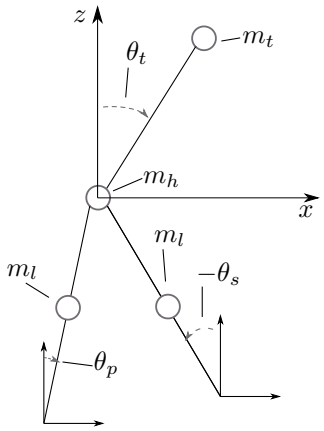


Figure 3. The biped system in Example 3.2. The angle vector θ contains the planted leg angle θ_p, the swing leg angle θ_s, and the torso angle θ_t. The velocity vector ω contains the planted leg angular velocity ω_p, the swing leg angular velocity ω_s, and the torso angular velocity ω_t. The input u is the input torque, where u_p is the torque applied on the planted leg from the ankle, u_s is the torque applied on the swing leg from the hip, and u_t is the torque applied on the torso from the hip.

torques that produce an acceleration a for a special state x are determined by a function μ, defined as $\mu(x,a) := B^{-1}(D_f(\theta)a + C_f(\theta,\omega)\omega + G_f(\theta))$. By applying $u = \mu(x,a)$ to $\dot{\omega} = \alpha(x,u)$, we obtain $\dot{\omega} = a$. Then, the flow map f is defined as

$$f(x,a) := \begin{bmatrix} \omega \\ a \end{bmatrix} \quad (x,a) \in C.$$

Flow is allowed when only one leg is in contact with the ground. To determine if the biped has reached the end of a step, we define $h(x) := \phi_s - \theta_p$ for all $x \in \mathbb{R}^6$ where ϕ_s denotes the step angle. The condition $h(x) \geq 0$ indicates that only one leg is in contact with the ground. Thus, the flow set is defined as

$$C := \{(x,a) \in \mathbb{R}^6 \times \mathbb{R}^3 : h(x) \geq 0\}.$$

Furthermore, a step occurs when the change of h is such that θ_p is approaching ϕ_s, and h equals zero. Thus, the jump set D is defined as

$$D := \{(x,a) \in \mathbb{R}^6 \times \mathbb{R}^3 : h(x) = 0, \omega_p \geq 0\}.$$

Following Grizzle et al. (2001), when a step occurs, the swing leg becomes the planted leg, and the planted leg becomes the swing leg. The function Γ is defined to swap angles and velocity variables as $\theta^+ = \Gamma(\theta)$. The angular velocities after a step are determined by a contact model denoted as $\Omega(x) := (\Omega_p(x), \Omega_s(x), \Omega_t(x))$, where Ω_p, Ω_s, and Ω_t are the angular velocity of the planted leg, swing leg,

and torso, respectively. Then, the jump map g is defined as

$$g(x,a) := \begin{bmatrix} \Gamma(\theta) \\ \Omega(x) \end{bmatrix} \quad \forall (x,a) \in D. \tag{15}$$

For more information about the contact model, see (Grizzle et al. 2001, Appendix A).

An example of a motion planning problem for the walking robot system is as follows: using a bounded input signal, find a solution pair to (2) associated to the walking robot system completing a step of a walking cycle. One way to characterize a walking cycle is to define the final state and the initial state as the states before and after a jump occurs, respectively. One such motion planning problem is given by defining the final state set as $X_f = \{(\phi_s, -\phi_s, 0, 0.1, 0.1, 0)\}$, the initial state set as $X_0 = \{x_0 \in \mathbb{R}^6 : x_0 = g(x_f, 0), x_f \in X_f\}$ where the input argument of g can be set arbitrarily because input does not affect the value of g; see (15), and the unsafe set as $X_u = \{(x,a) \in \mathbb{R}^6 \times \mathbb{R}^3 : a_1 \notin [a_1^{\min}, a_1^{\max}]$ or $a_2 \notin [a_2^{\min}, a_2^{\max}]$ or $a_3 \notin [a_3^{\min}, a_3^{\max}]$ or $(x,a) \in D\}$, where $a_1^{\min}$, $a_2^{\min}$, and $a_3^{\min}$ are the lower bounds of a_1, a_2, and a_3, respectively, and $a_1^{\max}$, $a_2^{\max}$, and $a_3^{\max}$ are the upper bounds of a_1, a_2, and a_3, respectively. We also solve this motion planning problem later in this paper.

4 A Forward Propagation Algorithm Template

In this section, an algorithm template, which we refer to as *forward propagation algorithm template*, is proposed for the design of motion planning algorithms to solve the motion planning problem $\mathcal{P} = (X_0, X_f, X_u, (C, f, D, g))$. The main steps of the forward propagation algorithm template are as follows.

Step 1: Propagate the state from the initial state set X_0 forward in hybrid time without intersecting with the unsafe set X_u.

Step 2: During the propagation, if a solution pair to the hybrid system (C, f, D, g) ending in the final state set X_f is found, then return it as a motion plan.

In the remainder of this section, first, we formalize the propagation operation used in the algorithm template and show that the result constructed by the propagation operation satisfies the requirements in Problem 1. Then, we present the algorithm template framework and analyze its soundness and exactness properties. A sampling-based motion planning algorithm, which we call HyRRT, that is designed using the proposed algorithm template is presented in the forthcoming Section 5.

4.1 Propagation Operation

The proposed template requires an operation to collect all solution pairs that start from the initial state set, without reaching the unsafe set. The definition of the forward propagation operation is formalized as follows.

Definition 4.1. *(Forward propagation operation) Given a hybrid system $\mathcal{H} = (C, f, D, g)$, an initial state set X_0, and an unsafe set X_u, the forward propagation operation with parameters $(X_0, X_u, (C, f, D, g))$ constructs a set $\mathcal{S}$ that collects all the solution pairs $\psi = (\phi, u)$ that satisfy the following conditions:*

1. *$\psi = (\phi, u)$ is a solution pair to $\mathcal{H}$ (see Definition 2.6);*
2. *$\phi(0,0) \in X_0$;*
3. *$\psi(t,j) = (\phi(t,j), \upsilon(t,j)) \notin X_u$ for all $(t,j) \in \operatorname{dom} \psi$.*

In other words,

$$\mathcal{S} := \{\psi = (\phi, \upsilon) : (\phi, \upsilon) \in \mathcal{S}_{\mathcal{H}}, \phi(0,0) \in X_0, \\ (\phi(t,j), \upsilon(t,j)) \notin X_u \; \forall (t,j) \in \operatorname{dom} \psi\}$$

where $\mathcal{S}_{\mathcal{H}}$ denotes the set of all solution pairs to $\mathcal{H}$.

Remark 4.2. *Note that Definition 4.1 describes an exact searching process since it constructs a set that includes all possible solution pairs. The implementation of an algorithm constructing such set is difficult in practice because there can be infinite many possible values for the input at each hybrid time instance. Additionally, the solution pairs collected in $\mathcal{S}$ are not limited to maximal solutions, which means that any truncation of a solution pair in $\mathcal{S}$ should also be collected in $\mathcal{S}$.*

Although it is practically difficult to implement the propagation operation exactly, however, this operation can be implemented incrementally and numerically. The search trees constructed in the classic motion planning algorithms, such as RRT, can be seen as an approximation of the set $\mathcal{S}$ because any path in the search trees starting from X_0 can be seen as an element in $\mathcal{S}$. An RRT-type implementation of the propagation operation is presented in the forthcoming Section 5. ▲

4.2 Forward Propagation Algorithm Template and Property Analysis

The forward propagation algorithm is given in Algorithm 1. The inputs of the proposed algorithm are as follows:

- The motion planning problem is given as $\mathcal{P} = (X_0, X_f, X_u, (C, f, D, g))$ (defined in Problem 1).
- The set of solution pairs $\mathcal{S}$ to store forward propagation operation result.

The algorithm template collects all collision-free solution pairs to $\mathcal{H}$ starting from X_0 and stores them in the set $\mathcal{S}$ (Line 1). Then, the algorithm template searches for a solution pair in the set $\mathcal{S}$ that ends in X_f (Line 2). If such a solution pair is found, then the algorithm returns this solution pair as a motion plan (Line 3). If such a solution pair is not found, then the algorithm template returns failure (Line 5).

Soundness guarantees the correctness of the returned solution, as defined next.

Algorithm 1 Forward propagation algorithm template

Input: $X_0, X_f, X_u, \mathcal{H} = (C, f, D, g)$
Output: motion plan ψ or `Failure` signal

1: Forward propagate with parameters $(X_0, X_u, (C, f, D, g))$ (see Definition 4.1) and store results in $\mathcal{S}$.
2: **if** $\exists \psi = (\phi, u) \in \mathcal{S}$ such that $\phi(T, J) \in X_f$ where $(T, J) = \max \operatorname{dom} \psi$ **then**
3: **return** ψ.
4: **else**
5: **return** `Failure`.
6: **end if**

Definition 4.3. (Soundness Fink and Veloso (1994)). *An algorithm is said to be sound for a problem if for any possible data defining the problem (e.g., $(X_0, X_f, X_u, (C, f, D, g))$ in Problem 1), the result returned by the algorithm is always a correct solution to the problem.*

The next result shows that the proposed forward propagation algorithm template is sound for the motion planning problem in Problem 1.

Theorem 4.4. *The forward propagation algorithm in Algorithm 1 is sound for the motion planning problem $\mathcal{P}$ in Problem 1.*

Proof. If non failure is returned (Line 3), from Definition 4.1, the solution ψ is a solution pair to $\mathcal{H}$ that starts from X_0, and does not intersect with X_u. In addition, Line 2 in Algorithm 1 guarantees that ψ ends in X_f. Therefore, the returned result is a motion plan to Problem 1.

If the failure signal is returned (Line 5), then there does not exist an element in $\mathcal{S}$ that starts from X_0, ends in X_f, and does not intersect with X_u (Line 2). This means that there does not exist a motion plan to Problem 1.

In conclusion, the returned result is a correct solution to Problem 1.

Exactness guarantees that the algorithm will always return a solution when one exists, as defined next.

Definition 4.5. (Exactness Goldberg (1994)). *An algorithm is said to be exact for a problem if for any possible data defining the problem (e.g., $(X_0, X_f, X_u, (C, f, D, g))$ in Problem 1), it finds a solution when one exists.*

The next result shows that the proposed forward propagation algorithm template is exact for the motion planning problem in Problem 1.

Theorem 4.6. *The forward propagation algorithm in Algorithm 1 is an exact algorithm for the motion planning problem $\mathcal{P}$ in Problem 1.*

Proof. If a solution ψ to the given motion planning problem exists, then from Problem 1, ψ is a solution pair to $\mathcal{H}$ that starts from X_0, ends at X_f, and does not have intersection with X_u. This means that ψ should be collected by the forward propagation operation with parameters $(X_0, X_u, (C, f, D, g))$, be stored in $\mathcal{S}$, and the condition in Line 2 is `true`. Therefore, ψ is returned by Algorithm 1.

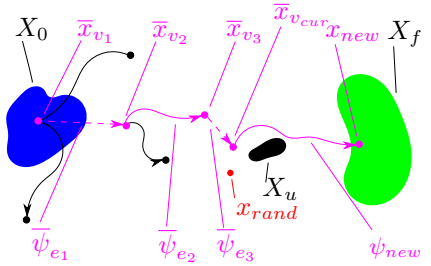
Remark 4.7. *The exactness property guarantees that the proposed algorithm returns a solution when one exists. However, when no solution exists, the proposed algorithm is not guaranteed to return a failure signal in finite time because the forward propagation operations are not always guaranteed to be completed within finite time. This is not a unique problem of our algorithm template, but rather a general issue in motion planning algorithms; see Karaman et al. (2011).* ▲

5 HyRRT: A Sampling-based Motion Planning Algorithm for Hybrid Systems

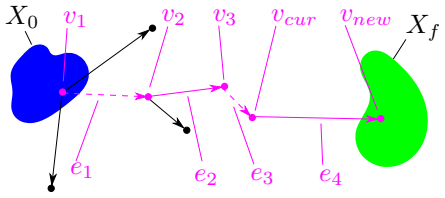
In this section, an RRT-type motion planning algorithm for hybrid systems, called HyRRT, is proposed. This algorithm implements the propagation operation in Algorithm 1 by incrementally constructing a search tree.

5.1 Overview

HyRRT searches for a motion plan by incrementally constructing a search tree. The search tree is modeled by a directed tree. A directed tree $\mathcal{T}$ is a pair $\mathcal{T} = (V, E)$, where V is a set whose elements are called vertices and E is a set of paired vertices whose elements are called edges. The edges in the directed tree are directed, which means the pairs of vertices that represent edges are ordered. The set of edges E is defined as $E \subseteq \{(v_1, v_2) : v_1 \in V, v_2 \in V, v_1 \neq v_2\}$. The edge $e = (v_1, v_2) \in E$ represents an edge from v_1 to v_2. A path in $\mathcal{T} = (V, E)$ is a sequence of vertices $p = (v_1, v_2, ..., v_k)$ such that $(v_i, v_{i+1}) \in E$ for all $i = 1, 2, ..., k - 1$.



(a) States and solution pairs.



(b) Search tree associated with the states and solution pairs in Figure 4(a).

Figure 4. The association between states/solution pairs and the vertices/edges in the search tree. The blue region denotes X_0, the green region denotes X_f, and the black region denotes X_u. The dots and lines between dots in Figure 4(b) denote the vertices and edges associated with the states and solution pairs in Figure 4(a). The path $p = (v_1, v_2, v_3, v_{cur}, v_{new})$ in the search graph in Figure 4(b) represents the solution pair $\tilde{\psi}_p = \overline{\psi}_{e_1} | \overline{\psi}_{e_2} | \overline{\psi}_{e_3} | \psi_{new}$ in Figure 4(a).

Each vertex in the search tree $\mathcal{T}$ is associated with a state value of $\mathcal{H}$. Each edge in the search tree is associated with a

solution pair to $\mathcal{H}$ that connects the state values associated with their endpoint vertices. The state value associated with vertex $v \in V$ is denoted as $\overline{x}_v$ and the solution pair associated with edge $e \in E$ is denoted as $\overline{\psi}_e$, as shown in Figure 4.

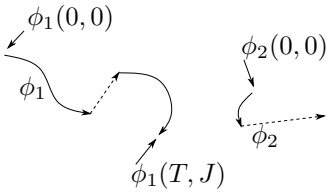
The concatenation operation is formulated as follows and heavily used in constructing the search tree.

Definition 5.1. (Concatenation operation). *Given two functions* $\phi_1 : \operatorname{dom}\phi_1 \to \mathbb{R}^n$ *and* $\phi_2 : \operatorname{dom}\phi_2 \to \mathbb{R}^n$, *where* $\operatorname{dom}\phi_1$ *and* $\operatorname{dom}\phi_2$ *are hybrid time domains,* ϕ_2 *can be concatenated to* ϕ_1 *if* ϕ_1 *is compact and* $\phi : \operatorname{dom}\phi \to \mathbb{R}^n$ *is the concatenation of* ϕ_2 *to* ϕ_1, *denoted* $\phi = \phi_1|\phi_2$, *namely,*

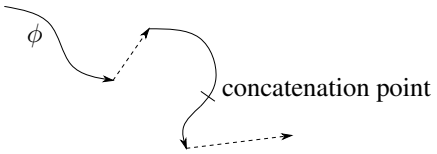
1) $\operatorname{dom}\phi = \operatorname{dom}\phi_1 \cup (\operatorname{dom}\phi_2 + \{(T, J)\})$, *where* $(T, J) = \max \operatorname{dom}\phi_1$ *and the plus sign denotes Minkowski addition;*

2) $\phi(t, j) = \phi_1(t, j)$ *for all* $(t, j) \in \operatorname{dom}\phi_1 \backslash \{(T, J)\}$ *and* $\phi(t, j) = \phi_2(t - T, j - J)$ *for all* $(t, j) \in \operatorname{dom}\phi_2 + \{(T, J)\}$.

The following example in Figure 5 illustrates the concatenation of the functions on hybrid time domains. Let ϕ_1 and ϕ_2 be two functions defined on hybrid time domains and $(T, J) = \max \operatorname{dom}\phi_1$. When ϕ_2 is concatenated to ϕ_1, the domain of the concatenation is constructed by translating the domain of ϕ_2 by (T, J) and concatenating the translated domain to the domain of ϕ_1. Particularly, if ϕ_1 and ϕ_2 are hybrid arcs, their concatenation is not guaranteed to be a hybrid arc. The reason is that the concatenation result may not be continuous at (T, J). The concatenation of two trajectories is illustrated in Figure 5. In the figure, the solid lines denote the flows and the dotted lines denote the jumps. In Figure 5(a), the trajectories ϕ_1 and ϕ_2 denote the original trajectories and the trajectory ϕ in Figure 5(b) denotes the concatenation of ϕ_2 to ϕ_1.



(a) The original trajectory ϕ_1 and ϕ_2.



(b) The trajectory ϕ resulting from concatenating ϕ_2 to ϕ_1.

Figure 5. The concatenation ϕ of trajectory ϕ_2 to trajectory ϕ_1.

Proposition 5.2 below shows that the concatenation of solution pairs satisfies the definition of solution pair in Definition 2.6 under mild conditions.

Proposition 5.2. *Given two solution pairs* $\psi_1 = (\phi_1, \upsilon_1)$ *and* $\psi_2 = (\phi_2, \upsilon_2)$ *to a hybrid system* $\mathcal{H}$, *their concatenation* $\psi = (\phi, \upsilon) = (\phi_1|\phi_2, \upsilon_1|\upsilon_2)$, *denoted* $\psi = \psi_1|\psi_2$, *is a solution pair to* $\mathcal{H}$ *if the following hold:*

1) $\psi_1 = (\phi_1, \upsilon_1)$ *is compact;*

2) $\phi_1(T, J) = \phi_2(0, 0)$, *where* $(T, J) = \max \operatorname{dom}\psi_1$;

3) *If both* $I^J_{\psi_1}$ *and* $I^0_{\psi_2}$ *have nonempty interior, where, for each* $j \in \{0, J\}$, $I^j_\psi = \{t : (t, j) \in \operatorname{dom}\psi\}$ *and* $(T, J) = \max \operatorname{dom}\psi_1$, *then* $\psi_2(0, 0) \in C$.

Proof. See Appendix A.

Remark 5.3. *Item 1 in Proposition 5.2 guarantees that* ψ_2 *can be concatenated to* ψ_1. *Definition 5.1 suggests that if* ψ_2 *can be concatenated to* ψ_1, ψ_1 *is required to be compact. Item 2 in Proposition 5.2 guarantees that the* ϕ *is a hybrid arc or* ψ *satisfies item 3 in Definition 2.6 at hybrid time* (T, J), *where* $(T, J) = \max \operatorname{dom}\psi_1$. *Item 3 in Proposition 5.2 guarantees that the concatenation result* ψ *satisfies item 2 in Definition 2.6 at hybrid time* (T, J). *Note that item 2 therein does not require that* $\psi_1(T, J) \in C$ *and* $\psi_2(0, 0) \in C$ *since* $T \notin \operatorname{int} I^J_{\psi_1}$ *and* $0 \notin \operatorname{int} I^0_{\psi_2}$. *However,* T *may belong to the interior of* I^J_ψ *after concatenation. Hence, item 3 guarantees that if* T *belongs to the interior of* I^J_ψ *after concatenation, then* $\psi(T, J) \in C$. ▲

The solution pair that the path $p = (v_1, v_2, ..., v_k)$ represents is the concatenation of all the solutions associated with the edges therein, namely,

$$\tilde{\psi}_p := \overline{\psi}_{(v_1,v_2)}|\overline{\psi}_{(v_2,v_3)}|\ ...\ |\overline{\psi}_{(v_{k-1},v_k)} \tag{16}$$

where $\tilde{\psi}_p$ denotes the solution pair associated with the path p. An example of the path p and its associated solution pair $\tilde{\psi}_p$ is shown in Figure 4.

The proposed HyRRT algorithm requires a library of possible inputs. The input library $(\mathcal{U}_C, \mathcal{U}_D)$ includes the input signals that can be applied during flows (collected in $\mathcal{U}_C$) and the input values that can be applied at jumps (collected in $\mathcal{U}_D$). More details about the input library are presented in the forthcoming Section 5.2.

Next, we introduce the main steps executed by HyRRT. Given the motion planning problem $\mathcal{P} = (X_0, X_f, X_u, (C, f, D, g))$ and the input library $(\mathcal{U}_C, \mathcal{U}_D)$, HyRRT performs the following steps:

Step 1: Sample a finite number of points from X_0 and initialize a search tree $\mathcal{T} = (V, E)$ by adding vertices associated with each sampling point.

Step 2: Randomly select one regime among flow regime and jump regime for the evolution of $\mathcal{H}$.

Step 3: Randomly select a point x_{rand} from C' (D') if the flow (jump, respectively) regime is selected in Step 2.

Step 4: Find the vertex associated with the state value that has minimal Euclidean distance to x_{rand}, denoted v_{cur}, as is shown in Figure 4(b).

Step 5: Randomly select an input signal (value) from $\mathcal{U}_C$ ($\mathcal{U}_D$) if the flow (jump, respectively) regime is selected. Then, compute a solution pair using continuous (discrete, respectively) dynamics

simulator starting from $\overline{x}_{v_{cur}}$ with the selected input applied, denoted $\psi_{new} = (\phi_{new}, \upsilon_{new})$. Denote the final state of ϕ_{new} as x_{new}, as is shown in Figure 4(a). If ψ_{new} does not intersect with X_u, add a vertex v_{new} associated with x_{new} to V and an edge (v_{cur}, v_{new}) associated with ψ_{new} to E.

Step 6: If a new vertex is added, check whether a path in the search tree can be used to construct a motion plan. If so, return the motion plan; otherwise, proceed to **Step 2**.

In the remainder of this section, first, we formalize the input library for the motion planning problem for hybrid systems. Then, the continuous dynamics simulator and discrete dynamics simulator to compute a new solution pair from the given starting state and input are introduced. After those components being nicely introduced, the HyRRT algorithm to solve the motion planning problem for hybrid systems is presented.

5.2 Hybrid Input Library

HyRRT requires a library of inputs to simulate solution pairs. Note that inputs are constrained by the flow set C and the jump set D of the hybrid system $\mathcal{H}$. Specifically, given C and D, the set of input signals allowed during flows, denoted $\mathcal{U}_C$, and the set of input values at jumps, denoted $\mathcal{U}_D$, are described as follows.

1. The input signal applied during flows is a continuous-time signal, denoted $\tilde{\upsilon}$, that is specified by a function from an interval of time of the form $[0, t^*]$ to U_C in (3), namely, for some $t^* \in \mathbb{R}_{\geq 0}$,

$$\tilde{\upsilon} : [0, t^*] \to U_C.$$

Definition 2.3 also requires that $\tilde{\upsilon}$ is Lebesgue measurable and locally essentially bounded. Then, the set $\mathcal{U}_C$ collects each such $\tilde{\upsilon}$. Given $\tilde{\upsilon} \in \mathcal{U}_C$, the functional $\bar{t} : \mathcal{U}_C \to [0, \infty)$ returns the time duration of $\tilde{\upsilon}$. Namely, given $\tilde{\upsilon} : [0, t^*] \to U_C$, $\bar{t}(\tilde{\upsilon}) = t^*$.

2. The input applied at a jump is specified by the set U_D in (4). The set $\mathcal{U}_D$ collects possible input values that can be applied at jumps, namely,

$$\mathcal{U}_D \subset U_D.$$

The pair of sets $(\mathcal{U}_C, \mathcal{U}_D)$ defines the input library, denoted $\mathcal{U}$, namely,

$$\mathcal{U} := (\mathcal{U}_C, \mathcal{U}_D).$$

Input Library Construction Procedure: A procedure to construct $\mathcal{U}_C$ using constant inputs is given as follows:

Step 1: Set T_m to a positive constant. Construct the safe input set during the flows from U_C as $U_C^s := \{u \in U_C : \exists x \in C', (x, u) \notin X_u\}$.

Step 2: For each point $u^s \in U_C^s$ and $t_m \in (0, T_m]$, construct an input signal $[0, t_m] \to \{u^s\}$ and add it to $\mathcal{U}_C$.

Set $\mathcal{U}_D$ can be constructed as $\mathcal{U}_D := \{u \in U_D : \exists x \in D', (x, u) \notin X_u\}$ from U_D.

Remark 5.4. *The safe input set during flows, U_C^s, is constructed based on the flow set C provided by the hybrid system data and the unsafe set X_u. The set U_C^s includes all possible inputs during flow, characterized by U_C, except those for which, for all possible states during flow, the state-input pair will surely be in the unsafe set X_u.*

The input library construction procedure above does not require discretization. No parameters are needed except the upper bound over the domain of the input signals, T_m.

5.3 Continuous Dynamics Simulator

HyRRT requires a simulator to compute the solution pair starting from a given initial state $x_0 \in C'$ with a given input signal $\tilde{\upsilon} \in \mathcal{U}_C$ applied, following continuous dynamics. The initial state x_0, the flow set C, and the flow map f are used in the simulator.

Note that when the simulated solution enters the intersection between the flow set C and the jump set D, it can either keep flowing or jump. In Sanfelice et al. (2013), the hybrid system simulator HyEQ uses a scalar priority option flag `rule` to show whether the simulator gives priority to jumps (`rule` $= 1$), priority to flows (`rule` $= 2$), or no priority (`rule` $= 3$) when both $x \in C$ and $x \in D$ hold. When no priority is selected, then the simulator randomly selects to flow or jump. In this paper, the case `rule` $= 3$ is not considered[3].

The proposed simulator should be able to solve the following problem.

Problem 2. *Given the flow set C, the flow map f, and the jump set D of a hybrid system $\mathcal{H}$ with input $u \in \mathbb{R}^m$ and state $x \in \mathbb{R}^n$, a priority option flag `rule` $\in \{1, 2\}$, an initial state $x_0 \in \mathbb{R}^n$, and an input signal $\tilde{\upsilon} \in \mathcal{U}_C$ such that $(x_0, \tilde{\upsilon}(0)) \in C$, find a pair $(\phi, \upsilon) : [0, t^*] \times \{0\} \to \mathbb{R}^n \times \mathbb{R}^m$, where $t^* \in [0, \bar{t}(\tilde{\upsilon})]$, such that the following hold:*

1. *$\phi(0, 0) = x_0$;*
2. *For all $t \in [0, t^*]$, $\upsilon(t, 0) = \tilde{\upsilon}(t)$;*
3. *If $[0, t^*]$ has nonempty interior,*
   (a) *the function $t \mapsto \phi(t, 0)$ is locally absolutely continuous,*
   (b) *for all $t \in (0, t^*)$,*

$$\begin{aligned} (\phi(t,0), \upsilon(t,0)) &\in C \backslash D \quad \text{if } \texttt{rule} = 1 \\ (\phi(t,0), \upsilon(t,0)) &\in C \quad \text{if } \texttt{rule} = 2, \end{aligned}$$

   (c) *for almost all $t \in [0, t^*]$,*

$$\frac{d}{dt}\phi(t, 0) = f(\phi(t, 0), \upsilon(t, 0)). \tag{17}$$

Remark 5.5. *In general, a solution pair to $\mathcal{H}$ that solves Problem 2 may not be unique. Note that we can impose assumptions to get uniqueness as in (Sanfelice 2021, Proposition 2.11), as follows:*

1. *for every* $x_0 \in \overline{C}\backslash D$, $T > 0$ *and* $\tilde{\upsilon} \in \mathcal{U}_C$, *if two absolutely continuous* $z_1, z_2 : [0, T] \to \mathbb{R}^n$ *are such that* $z_i(t) = f(z_i(t), \tilde{\upsilon}(t))$ *for almost all* $t \in [0, T]$, $(z_i(t), \tilde{\upsilon}(t)) \in C$ *for all* $t \in (0, T]$, *and* $z_i(0) = x_0$, $i = 1, 2$, *then* $z_1(t) = z_2(t)$ *for all* $t \in [0, T]$;

2. *for every* $x_0 \in C \cap D$, *there does not exist* $\epsilon > 0$ *and an absolutely continuous function* $z : [0, \epsilon] \to \mathbb{R}^n$ *such that* $z(0) = x_0$, $\dot{z}(t) = f(z(t), \upsilon(t))$ *for almost all* $t \in [0, \epsilon]$ *and* $(z_i(t), \upsilon(t)) \in C$ *for all* $t \in (0, \epsilon]$.

In this paper, the simulator is designed to simulate a maximal solution pair (ϕ, υ) solving Problem 2. The definition of maximal solution is given as follows; see Sanfelice (2021).

Definition 5.6. (Maximal solution). *A solution pair* ψ *to* $\mathcal{H}$ *that solves Problem 2 is said to be maximal if there does not exist another solution pair* ψ' *to* $\mathcal{H}$ *that solves Problem 2 such that* $\operatorname{dom} \psi$ *is a proper subset of* $\operatorname{dom} \psi'$ *and* $\psi(t, 0) = \psi'(t, 0)$ *for all* $t \in \operatorname{dom}_t \psi$.

The module to simulate the maximal solution pair to $\mathcal{H}$ that solves Problem 2 is called the *continuous dynamics simulator*. The inputs of this module are the flow set C, the flow map f, the jump set D, the priority option `rule`, the initial state x_0, and the input signal $\tilde{\upsilon} \in \mathcal{U}_C$. The output of this module is the maximal solution pair (ϕ, u) to $\mathcal{H}$ that solves Problem 2. This module is denoted as

$$(\phi, u) \leftarrow \texttt{continuous\_simulator}(C, f, D, \texttt{rule}, x_0, \tilde{\upsilon}). \tag{18}$$

The continuous dynamics simulator performs the following steps. Given the flow set C, the flow map f, the jump set D, the priority option `rule`, the initial state x_0, and the input signal $\tilde{\upsilon} \in \mathcal{U}_C$,

Step 1: Solve for $\hat{\phi} : [0, \bar{t}(\tilde{\upsilon})] \to \mathbb{R}^n$ to satisfy

$$\hat{\phi}(t) = x_0 + \int_0^t f(\hat{\phi}(\tau), \tilde{\upsilon}(\tau)) d\tau \quad t \in [0, \bar{t}(\tilde{\upsilon})]. \tag{19}$$

Step 2: Calculate the largest time t such that over $[0, t)$, (ϕ, υ) is in $C\backslash D$ if $\texttt{rule} = 1$, or, if $\texttt{rule} = 2$, is in C, as follows:

$$\hat{t} := \begin{cases} \max\{t \in [0, \bar{t}(\tilde{\upsilon})] : (\hat{\phi}(t'), \tilde{\upsilon}(t')) \in C\backslash D, \\ \forall t' \in (0, t)\} & \text{if } \texttt{rule} = 1, \\ \max\{t \in [0, \bar{t}(\tilde{\upsilon})] : (\hat{\phi}(t'), \tilde{\upsilon}(t')) \in C, \\ \forall t' \in (0, t)\} & \text{if } \texttt{rule} = 2. \end{cases} \tag{20}$$

Step 3: Construct the solution function pair $(\phi, \upsilon) : [0, \hat{t}] \times \{0\} \to \mathbb{R}^n \times \mathbb{R}^m$ by

$$\phi(t, 0) = \hat{\phi}(t), \upsilon(t, 0) = \tilde{\upsilon}(t) \; \forall t \in [0, \hat{t}]. \tag{21}$$

The solution function pair (ϕ, u) is a maximal solution to $\mathcal{H}$ that solves Problem 2 and, hence, the output of module `continuous_simulator`. The construction of $\hat{\phi}$ in Step 1 can be approximated by employing numerical integration methods. To determine $\hat{t}$ in Step 2, zero-crossing detection algorithms can be used to detect the largest time t such that over $[0, t)$, (ϕ, υ) is in $C\backslash D$ if $\texttt{rule} = 1$, or, if $\texttt{rule} = 2$, is in C. For the computation framework that implements the simulator of continuous dynamics, see Appendix B.

5.4 Discrete Dynamics Simulator

HyRRT algorithm also requires a simulator to compute the solution pair with a single jump, starting from an initial state $x_0 \in D'$ with a given input value $u_D \in \mathcal{U}_D$ applied. Such a simulator only requires evaluating g as the following problem states.

Problem 3. *Given the jump set* D *and jump map* g *of hybrid system* $\mathcal{H}$ *with input* $u \in \mathbb{R}^m$, *state* $x \in \mathbb{R}^n$, *an initial state* $x_0 \in \mathbb{R}^n$, *and an input value* $u_D \in \mathcal{U}_D$ *such that* $(x_0, u_D) \in D$, *find a pair* $(\phi, \upsilon) : \{0\} \times \{0, 1\} \to \mathbb{R}^n \times \mathbb{R}^m$ *such that the following hold:*

1. $\phi(0, 0) = x_0$;
2. $\upsilon(0, 0) = u_D$;
3. $\phi(0, 1) = g(\phi(0, 0), \upsilon(0, 0))$.

Problem 3 can be solved by constructing a function pair $(\phi, \upsilon) : \{0\} \times \{0, 1\} \to \mathbb{R}^n \times \mathbb{R}^m$ in the following way:

$$\begin{aligned} \phi(0, 0) = x_0, \quad \phi(0, 1) = g(x_0, u_D) \\ \upsilon(0, 0) = u_D, \quad \upsilon(0, 1) = p \in \mathbb{R}^m. \end{aligned} \tag{22}$$

In (22), we can implement $\upsilon(0, 1) \leftarrow p \in \mathbb{R}^m$ by selecting an arbitrary point p in $\mathbb{R}^m$ such that $(\phi(0, 1), p) \notin X_u$.

The function pair in (22) can be constructed by a module called *discrete dynamics simulator*. The inputs of this module are the jump set D, the jump map g, the initial state $x_0 \in D'$, and the input $u_D \in \mathcal{U}_D$ such that $(x_0, u_D) \in D$. The output of this module is the solution pair (ϕ, υ) constructed in (22). This module is denoted as

$$(\phi, \upsilon) \leftarrow \texttt{discrete\_simulator}(D, g, x_0, u_D). \tag{23}$$

5.5 HyRRT Algorithm

Following the overview in Section 5.1, the proposed algorithm is given in Algorithm 2. The inputs of Algorithm 2 are the problem $\mathcal{P} = (X_0, X_f, X_u, (C, f, D, g))$, the input library $(\mathcal{U}_C, \mathcal{U}_D)$, a parameter $p_n \in (0, 1)$, which tunes the probability of proceeding with the flow regime or the jump regime, an upper bound $K \in \mathbb{N}_{>0}$ for the number of iterations to execute, and two tunable sets $X_c \supset \overline{C'}$ and $X_d \supset D'$, which act as constraints in finding a closest vertex to x_{rand}. HyRRT is implemented in the following algorithm:

Step 1 in the overview provided in Section 5.1 corresponds to the function call $\mathcal{T}$.`init` in line 1 of Algorithm 2. **Step** 2 is implemented in line 3. **Step** 3 is implemented by the function call `random_state` in lines 5 and 8. **Step** 4 corresponds to the function call `nearest_neighbor` in line 1 of the function call `extend`. **Step** 5 is implemented by the function calls `new_state`, $\mathcal{T}$.`add_vertex`, and $\mathcal{T}$.`add_edge` in lines 3, 5, and 6 of the function call `extend`. **Step** 6 is implemented by the conditional execution of the function call `solution_check` in line 12 of HyRRT.

Each function in Algorithm 2 is defined next.

Algorithm 2 HyRRT algorithm

Input: $X_0, X_f, X_u, \mathcal{H} = (C, f, D, g), (\mathcal{U}_C, \mathcal{U}_D), p_n \in (0,1)$, $K \in \mathbb{N}_{>0}$

Output: motion plan ψ or `Failure` signal.

$\mathcal{T}$.`init`(X_0).
for $k = 1$ to K **do**
  randomly select a real number r from $[0, 1]$.
  if $r \leq p_n$ **then**
    $x_{rand} \leftarrow$ `random_state`$(\overline{C'})$.
    `is_extended` $\leftarrow$ `extend`$(\mathcal{T}, x_{rand}, (\mathcal{U}_C, \mathcal{U}_D), \mathcal{H}, X_u, X_c)$.
  else
    $x_{rand} \leftarrow$ `random_state`(D').
    `is_extended` $\leftarrow$ `extend`$(\mathcal{T}, x_{rand}, (\mathcal{U}_C, \mathcal{U}_D), \mathcal{H}, X_u, X_d)$.
  end if
  if `is_extended` $=$ `Advanced` **then**
    (`is_a_motion_found`, ψ) $\leftarrow$ `solution_check`$(\mathcal{T}, X_0, X_f, C)$.
    if `is_a_motion_found` = `true` **then**
      return ψ.
    end if
  end if
end for
return `Failure`.

Algorithm 3 Extend function

Input: $\mathcal{T}, x, (\mathcal{U}_C, \mathcal{U}_D), \mathcal{H}, X_u, X_*$

Output: `Advanced` or `Trapped` signal.

function EXTEND$((\mathcal{T}, x, (\mathcal{U}_C, \mathcal{U}_D), \mathcal{H}, X_u, X_*))$
  $v_{cur} \leftarrow$ `nearest_neighbor`$(x, \mathcal{T}, \mathcal{H}, X_*)$;
  (`is_a_new_vertex_generated`, x_{new}, ψ_{new}) $\leftarrow$ `new_state`$(v_{cur}, (\mathcal{U}_C, \mathcal{U}_D), \mathcal{H}, X_u)$
  if `is_a_new_vertex_generated` $= true$ **then**
    $v_{new} \leftarrow \mathcal{T}$.`add_vertex`$(x_{new})$;
    $\mathcal{T}$.`add_edge`$(v_{cur}, v_{new}, \psi_{new})$;
    return `Advanced`;
  end if
  return `Trapped`;
end function

5.5.1 $\mathcal{T}$.*init*(X_0)*:* The function call $\mathcal{T}$.`init` is used to initialize a search tree $\mathcal{T} = (V, E)$. It randomly selects a finite number of points from X_0. For each sampling point $x_0 \in X_0$, a vertex v_0 associated with x_0 is added to V. At this step, no edge is added to E.

5.5.2 $x_{rand} \leftarrow$ *random_state*(S)*:* The function call `random_state` randomly selects a point from the set $S \subset \mathbb{R}^n$. Rather than to select from $\overline{C'} \cup D'$, it is designed to select points from $\overline{C'}$ and D' separately depending on the value of r . The reason is that if $\overline{C'}$ (or D') has zero measure while D' (respectively, $\overline{C'}$) does not, the probability that the point selected from $\overline{C'} \cup D'$ lies in $\overline{C'}$ (respectively, D') is zero, which would prevent establishing probabilistic completeness.

5.5.3 $v_{cur} \leftarrow$ *nearest_neighbor*$(x_{rand}, \mathcal{T}, \mathcal{H}, X_*)$*:* The function call `nearest_neighbor` searches for a vertex v_{cur} in the search tree $\mathcal{T} = (V, E)$ such that its associated state value has minimal distance to x_{rand}. This function is implemented as solving the following optimization problem over $X_\star$, where $\star$ is either c or d.

Problem 4. *Given a hybrid system $\mathcal{H} = (C, f, D, g)$, $x_{rand} \in \mathbb{R}^n$, and a search tree $\mathcal{T} = (V, E)$, solve*

$$\begin{aligned} \arg\min_{v \in V} \quad & |\overline{x}_v - x_{rand}| \\ \textit{s.t.} \quad & \overline{x}_v \in X_\star. \end{aligned}$$

The data of Problem 4 comes from the arguments of the `nearest_neighbor` function call. This optimization problem can be solved by traversing all the vertices in $\mathcal{T} = (V, E)$.

5.5.4 (*is_a_new_vertex_generated*, x_{new}, ψ_{new}) $\leftarrow$ *new_state*$(v_{cur}, (\mathcal{U}_C, \mathcal{U}_D), \mathcal{H} = (C, f, D, g), X_u)$*:* Given v_{cur}, if $\overline{x}_{v_{cur}} \in \overline{C'} \backslash D'$, the function call `new_state` randomly selects an input signal $\tilde{v}$ from $\mathcal{U}_C$ such that $(\overline{x}_{v_{cur}}, \tilde{v}(0)) \in \overline{C}$ and generates a new maximal solution pair, denoted $\psi_{new} = (\phi_{new}, \upsilon_{new})$, by

$$\psi_{new} \leftarrow \texttt{continuous\_simulator}(C, f, D, 2, \overline{x}_{v_{cur}}, \tilde{v}) \tag{24}$$

where `continuous_simulator` is formulated as in (18).

If $\overline{x}_{v_{cur}} \in D' \backslash \overline{C'}$, the function call `new_state` randomly selects an input signal u_D from $\mathcal{U}_D$ such that $(\overline{x}_{v_{cur}}, u_D) \in D$ and generates a new solution pair, denoted $\psi_{new} = (\phi_{new}, \upsilon_{new})$, by

$$\psi_{new} \leftarrow \texttt{discrete\_simulator}(D, g, \overline{x}_{v_{cur}}, u_D). \tag{25}$$

where `discrete_simulator` is formulated as in (23).

If $\overline{x}_{v_{cur}} \in \overline{C'} \cap D'$, then this function generates ψ_{new} by randomly selecting flows or jump. This random selection is implemented by randomly selecting a real number r_D from the interval $[0, 1]$ and comparing r_D with a user-defined parameter $p_D \in (0, 1)$. If $r_D \leq p_D$, then the function call `new_state` generates ψ_{new} by flow, otherwise, by jump. The final state of ψ_{new} is denoted as x_{new}.

After ψ_{new} and x_{new} are generated, the function `new_state` checks if ψ_{new} is trivial, as defined in Definition 2.5, item 2. If so, then ψ_{new} does not explore any unexplored space and is not necessary to be added into $\mathcal{T}$. Hence,

`is_a_new_vertex_generated` $\leftarrow$ `false`

and the function call `new_state` is returned. Else, the function `new_state` checks if there exists $(t, j) \in \operatorname{dom} \psi_{new}$ such that $\psi_{new}(t, j) \in X_u$. If so, then ψ_{new} intersects with the unsafe set and

`is_a_new_vertex_generated` $\leftarrow$ `false`.

Otherwise,

`is_a_new_vertex_generated` $\leftarrow$ `true`.

5.5.5 $v_{new} \leftarrow \mathcal{T}$.*add_vertex*$(x_{new})$ *and* $\mathcal{T}$.*add_edge*$(v_{cur}, v_{new}, \psi_{new})$*:* The function call $\mathcal{T}$.`add_vertex` (x_{new}) adds a new vertex v_{new} associated with x_{new} to $\mathcal{T}$ and returns v_{new}. The function call $\mathcal{T}$.`add_edge`$(v_{cur}, v_{new}, \psi_{new})$ adds a new edge $e_{new} = (v_{cur}, v_{new})$ associated with ψ_{new} to $\mathcal{T}$.

5.5.6 $(\texttt{is\_a\_motion\_found}, \psi) \leftarrow \texttt{solution\_check}$ *($\mathcal{T}$, X_0, X_f, C):* When the function call `extend` returns `Advanced`, HyRRT checks if a path in $\mathcal{T}$ can be used to construct a motion plan solving the given motion planning problem. If this function finds a path $p = ((v_0, v_1), (v_1, v_2), ..., (v_{n-1}, v_n)) =: (e_0, e_1, ..., e_{n-1})$ in $\mathcal{T}$ such that

1) $\overline{x}_{v_0} \in X_0$,
2) $\overline{x}_{v_n} \in X_f$,
3) for $i \in \{0, 1, ..., n-2\}$, if $\overline{\psi}_{e_i}$ and $\overline{\psi}_{e_{i+1}}$ are both purely continuous, then $\overline{\psi}_{e_{i+1}}(0,0) \in C$,

then

$$\texttt{is\_a\_motion\_found} \leftarrow \texttt{true}$$

and the solution pair $\tilde{\psi}_p$ associated with path p, defined in (16), is a motion plan to the given motion planning problem, namely,

$$\psi \leftarrow \tilde{\psi}_p.$$

If HyRRT does not find a path satisfying items 1 - 3 above, then

$$\texttt{is\_a\_motion\_found} \leftarrow \texttt{false}.$$

More specifically, the items listed above guarantee that ψ satisfies all the conditions in Problem 1. Items 1 and 2 guarantee that $\tilde{\psi}_p$ starts from X_0 and ends within X_f. Then, we show that each condition in Proposition 5.2 is satisfied such that Proposition 5.2 guarantees that ψ is a solution pair to $\mathcal{H}$ as follows:

1. Because the input signal in $\mathcal{U}_C$ is compact, the solution pairs generated by `new_state` by flow are compact. The solution pairs generated by `new_state` by jump are compact since the domain of the solution pair generated in (22) is $\{0\} \times \{0, 1\}$. Therefore, the first condition in Proposition 5.2 is satisfied.
2. Note that for $i \in \{0, 1, 2, ..., n-2\}$, the final state of $\overline{\psi}_{e_i} = \overline{\psi}_{(v_i, v_{i+1})}$ equals the initial state of $\overline{\psi}_{e_{i+1}} = \overline{\psi}_{(v_{i+1}, v_{i+2})}$ because both of them are $\overline{x}_{v_{i+1}}$. Then, the second condition in Proposition 5.2 is satisfied.
3. Then, item 3 above guarantees that the third condition in Proposition 5.2 is satisfied.

Therefore, Proposition 5.2 guarantees that ψ is a solution pair to $\mathcal{H}$. Note that for each e in p, $\overline{\psi}_e$ does not intersect with unsafe set because the solution pairs that intersect the unsafe set have been excluded by the function `new_state`. Therefore, ψ satisfies all the conditions in Problem 1.

Remark 5.7. *Note that the choices of inputs in the function call* `new_state` *are random. Some RRT variants choose the optimal input that drives x_{new} closest to x_{rand}. However, Kunz and Stilman (2015) proves that such a choice makes the RRT algorithm probabilistically incomplete.* ▲

Remark 5.8. *In practice, item 2 above is too restrictive. Given $\epsilon > 0$ representing the tolerance associated with this condition, we implement item 2 as*

$$|\overline{x}_{v_n}|_{X_f} \leq \epsilon. \tag{26}$$

▲

6 Probabilistic Completeness Analysis

This section establishes probabilistic completeness of the HyRRT algorithm. Probabilistic completeness means that the probability that the planner fails to return a motion plan, if one exists, approaches zero as the number of samples approaches infinity.

6.1 Clearance of Motion Plan and Inflation of a Hybrid System

The clearance of a motion plan captures the distance between the motion plan and the boundary of the constraint sets, which in Problem 1, includes the initial state set X_0, the final state set X_f, the unsafe set X_u, the flow set C, and the jump set D. We propose two different clearances, *safety clearance* and *dynamics clearance*, that capture the distance to the constraint sets (X_0, X_f, X_u) and (C, D), respectively. As defined in Section 2.1, in the remainder of this paper, the notation $\mathbb{B}$ denotes the closed unit ball of appropriate dimension in the Euclidean norm.

Definition 6.1. (Safety clearance of a motion plan). *Given a motion plan $\psi = (\phi, \upsilon)$ to the motion planning problem $\mathcal{P} = (X_0, X_f, X_u, (C, f, D, g))$, the safety clearance of $\psi = (\phi, \upsilon)$ is given by $\delta_s > 0$ if, for each $\delta' \in [0, \delta_s]$, the following conditions are satisfied:*

1) $\{\phi(0,0)\} + \delta'\mathbb{B} \subset X_0$;

2) $\{\phi(T, J)\} + \delta'\mathbb{B} \subset X_f$, where $(T, J) = \max \operatorname{dom} \psi$;

3) For all $(t, j) \in \operatorname{dom} \psi$, $(\{\phi(t, j)\} + \delta'\mathbb{B}, \{\upsilon(t, j)\} + \delta'\mathbb{B}) \cap X_u = \emptyset$.

Definition 6.2. (Dynamics clearance of a motion plan). *Given a motion plan $\psi = (\phi, \upsilon)$ to the motion planning problem $\mathcal{P} = (X_0, X_f, X_u, (C, f, D, g))$, the dynamics clearance of $\psi = (\phi, \upsilon)$ is given by $\delta_d > 0$ if, for each $\delta' \in [0, \delta_d]$, the following conditions are satisfied:*

1) For all $(t, j) \in \operatorname{dom} \psi$ such that I^j has nonempty interior, $(\{\phi(t, j)\} + \delta'\mathbb{B}, \{\upsilon(t, j)\} + \delta'\mathbb{B}) \subset C$;

2) For all $(t, j) \in \operatorname{dom} \psi$ such that $(t, j+1) \in \operatorname{dom} \psi$, $(\{\phi(t, j)\} + \delta'\mathbb{B}, \{\upsilon(t, j)\} + \delta'\mathbb{B}) \subset D$.

Remark 6.3. *The definition of the two types of clearance is analogous to the clearance in Kleinbort et al. (2018), albeit with a specific consideration for the boundaries of different constraint sets. Safety clearance in Definition 6.1 is defined as the minimal positive distance between the motion plan and the nearest boundaries of the initial state set, final state set, and the unsafe set. Dynamics clearance in Definition 6.2, on the other hand, represents the minimal positive distance between the motion plan and the closest boundaries of the flow set and the jump set, depending on whether the motion plan is during a flow or at a jump.*

With both safety clearance and dynamics clearance defined, we are ready to define the clearance of a motion plan.

Definition 6.4. (Clearance of a motion plan). *Given a motion plan $\psi = (\phi, \upsilon)$ to the motion planning problem $\mathcal{P} = (X_0, X_f, X_u, (C, f, D, g))$, the clearance of ψ, denoted δ, is defined as the minimum between its safety clearance δ_s and dynamics clearance δ_d, i.e., $\delta := \min\{\delta_s, \delta_d\}$.*

In the probabilistic completeness result in (Kleinbort et al. 2018, Theorem 2), a motion plan with positive clearance is assumed to exist. However, the assumption that there exists a positive dynamics clearance is restrictive for hybrid systems. Indeed, if the motion plan reaches the boundary of the flow set or of the jump set, then the motion plan has no (dynamics) clearance; see Definition 6.2. Figure 6(a) shows a motion plan to the sample motion planning problem for the actuated bouncing ball system in Example 3.1 without clearance.

To overcome this issue and to assure that HyRRT is probabilistically complete, we inflate the hybrid system $\mathcal{H} = (C, f, D, g)$ as follows.

Definition 6.5. (δ-inflation of a hybrid system). *Given a hybrid system $\mathcal{H} = (C, f, D, g)$ and $\delta > 0$, the δ-inflation of the hybrid system $\mathcal{H}$, denoted $\mathcal{H}_\delta$ with data $(C_\delta, f_\delta, D_\delta, g_\delta)$, is given by*[4]

$$\mathcal{H}_\delta : \begin{cases} \dot{x} = f_\delta(x,u) & (x,u) \in C_\delta \\ x^+ = g_\delta(x,u) & (x,u) \in D_\delta \end{cases} \tag{27}$$

where

1) *The δ-inflation of the flow set is constructed as*

$$C_\delta := \{(x,u) \in \mathbb{R}^n \times \mathbb{R}^m : \exists (y,v) \in C : x \in \{y\} + \delta\mathbb{B}, u \in \{v\} + \delta\mathbb{B}\}, \tag{28}$$

2) *The δ-inflation of the flow map is constructed as*

$$f_\delta(x,u) := f(x,u) \quad \forall (x,u) \in C_\delta, \tag{29}$$

3) *The δ-inflation of the jump set is constructed as*

$$D_\delta := \{(x,u) \in \mathbb{R}^n \times \mathbb{R}^m : \exists (y,v) \in D : x \in \{y\} + \delta\mathbb{B}, u \in \{v\} + \delta\mathbb{B}\}, \tag{30}$$

4) *The δ-inflation of the jump map is constructed as*

$$g_\delta(x,u) := g(x,u) \quad \forall (x,u) \in D_\delta. \tag{31}$$

The above outlines a general method of constructing the δ-inflation of the given hybrid system. Next, this method is exemplified in the actuated bouncing ball system.

Example 6.6. (Actuated bouncing ball system in Example 3.1, revisited). *Given $\delta > 0$, the δ-inflation of hybrid system of the actuated bouncing ball system is constructed as follows.*

- *From (28), the δ-inflation of flow set C, denoted C_δ, is given by*

$$C_\delta = \{(x,u) \in \mathbb{R}^2 \times \mathbb{R} : x_1 \geq -\delta\}. \tag{32}$$

- *From (29), the δ-inflation of flow map f, denoted f_δ, is given by*

$$f_\delta(x,u) = f(x,u) = \begin{bmatrix} x_2 \\ -\gamma \end{bmatrix} \quad \forall (x,u) \in C_\delta \tag{33}$$

- *From (30), the δ-inflation of jump set D, denoted D_δ, is given by*

$$D_\delta := (\{(0,0)\} + \delta\mathbb{B}) \cup \{(x,u) \in \mathbb{R}^2 \times \mathbb{R} : x_2 \leq 0 \text{ and } -\delta \leq x_1 \leq \delta\}. \tag{34}$$

- *From (31), the δ-inflation of jump map g, denoted g_δ, is given by*

$$g_\delta(x,u) := g(x,u) = \begin{bmatrix} x_1 \\ -\lambda x_2 + u \end{bmatrix} \forall (x,u) \in D_\delta. \tag{35}$$

As is shown in Figure 6(b), the inflation of a hybrid system defined above serves to extend the boundary of both the flow and jump sets in $\mathcal{H}$ by δ and to capture the same continuous and discrete dynamics in those extended sets as in $\mathcal{H}$. Consequently, under the assumption that a motion plan to $\mathcal{P}$ with positive safety clearance exists, this methodology facilitates the establishment of such a motion plan with positive dynamics clearance and, in turn, with positive clearance. Next, we show that a motion plan to the original motion planning problem is also a motion plan to the motion planning problem for its δ-inflation.

Proposition 6.7. *Given a motion planning problem $\mathcal{P} = (X_0, X_f, X_u, (C, f, D, g))$ in Problem 1 with positive safety clearance, if ψ is a motion plan to $\mathcal{P}$, then for each $\delta > 0$, ψ is also a motion plan to the motion planning problem $\mathcal{P}_\delta = (X_0, X_f, X_u, (C_\delta, f_\delta, D_\delta, g_\delta))$, where $(C_\delta, f_\delta, D_\delta, g_\delta)$ is the δ-inflation of the hybrid system defined by (C, f, D, g).*

Proof. See Appendix C.

Next we show that the existing motion plan with positive safety clearance has positive clearance for the motion planning problem for the δ-inflation of the original hybrid system $\mathcal{H}$.

Lemma 6.8. *Let ψ be a motion plan to the motion planning problem $\mathcal{P} = (X_0, X_f, X_u, (C, f, D, g))$ formulated as Problem 1 with positive safety clearance $\delta_s > 0$. Then, for each $\delta_f > 0$, ψ is a motion plan to the motion planning problem $\mathcal{P}_{\delta_f} = (X_0, X_f, X_u, (C_{\delta_f}, f_{\delta_f}, D_{\delta_f}, g_{\delta_f}))$ with clearance $\delta = \min\{\delta_s, \delta_f\}$, where $\mathcal{H}_{\delta_f} = (C_{\delta_f}, f_{\delta_f}, D_{\delta_f}, g_{\delta_f})$ is the δ_f-inflation of $\mathcal{H} = (C, f, D, g)$.*

Proof. See Appendix D.

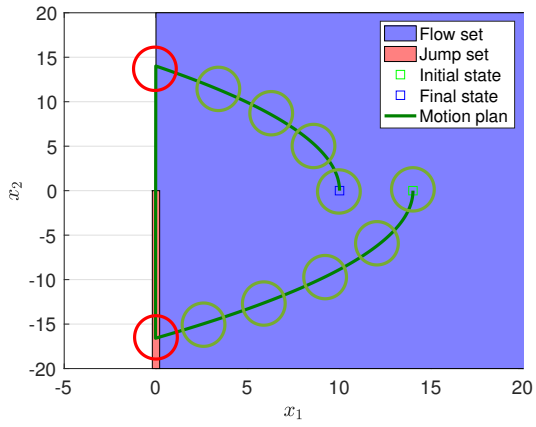
6.2 Assumptions

Similar to (Kleinbort et al. 2018, Definition 2), we consider input functions that are piecewise constant in the following sense.

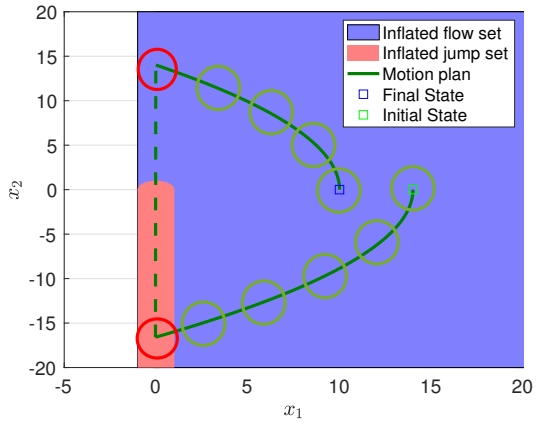
Definition 6.9. (Piecewise-constant function). *A function $\tilde{\upsilon}_c : [0, T] \to U_C$ is said to be a piecewise-constant function for probabilistic completeness if there exists $\Delta t \in \mathbb{R}_{>0}$, called resolution, such that*

1) *$k := \frac{T}{\Delta t} \in \mathbb{N}$;*

2) *for each $i \in \{1, 2, ..., k\}$, $t \mapsto \tilde{\upsilon}_c(t)$ is constant over $[(i-1)\Delta t, i\Delta t)$.*



(a)



(b)

Figure 6. Figure 6(a) shows a sample motion plan for the bouncing ball system in Example 3.1 without clearance. The dark green trajectory shows the motion plan. The blue region denotes the projection of flow set on the state space. The circles denote the boundaries of the balls along the motion plan at specific hybrid time instances. Note that the red balls depicted in Figure 6(a), which encircle the state of the motion plan at the boundary of the flow set, are not subsets of the flow set. In this case, the clearance δ does not exist. However, for the inflated system in (27), the existence of the circles along the motion plan with radius δ, as shown in Figure 6(b), implies the positive clearance δ of the motion plan.

From Definition 6.9, we define a motion plan notion with piecewise-constant inputs as follows.

Definition 6.10. (Motion plan with piecewise-constant input). *Given a motion planning problem $\mathcal{P}$, a motion plan (ϕ, υ) to $\mathcal{P}$ is said to be a* motion plan with piecewise-constant input with resolution Δt *if, for all $j \in \mathbb{N}$ such that I_υ^j has nonempty interior, $t \mapsto \upsilon(t, j)$ is a piecewise constant function with resolution Δt.*

Similar to Li et al. (2016), this paper considers that HyRRT is executed uniformly in relation to the random selection involved in **Step** 1, **Step** 3, and **Step** 5 of HyRRT, in the following sense.

Definition 6.11. (Uniform HyRRT). *HyRRT is said to be executed uniformly if the probability distribution of the random selection in the function calls $\mathcal{T}$.`init`, `random_state`, and `new_state` is the uniform distribution.*

Remark 6.12. *By Definition 6.11, the computation of the probability of randomly selecting a point that lies in a given set is simplified. When randomly selecting a point s from a set S, according to (Billingsley 2017, Page 257 (20.9)), the probability that s belongs to a subset $R \subset S$ is*

$$\text{Prob}[s \in R] = \int_{s \in R} \frac{1}{\mu(S)} ds = \frac{\mu(R)}{\mu(S)} \tag{36}$$

where μ denotes the Lebesgue measure of a set.

Subsequently, we define a complete input library that encompasses each possible input value that can be utilized either during the flow or at a jump.

Definition 6.13. (Complete input library). *The input library $(\mathcal{U}_C, \mathcal{U}_D)$ is said to be a complete input library if*

1) *$\mathcal{U}_C$ is the set of constant input signals and includes all possible input signals such that, for some $T_m > 0$, their time domains are closed subintervals of the interval $[0, T_m]$ including zero and their images belong to U_C. In other words, there exists $T_m > 0$ such that $\mathcal{U}_C = \{\tilde{\upsilon} : \text{dom}\, \tilde{\upsilon} = [0, T] \subset [0, T_m], \tilde{\upsilon}$ is constant, $\tilde{\upsilon} \in U_C\}$;*

2) *$\mathcal{U}_D = U_D$.*

Remark 6.14. *From Definition 6.13, the input signals in $\mathcal{U}_C$ are all constant functions. This property of $\mathcal{U}_C$ allows for the inputs of a motion plan with piecewise-constant input to be constructed by concatenating constant input signals in $\mathcal{U}_C$. The set $\mathcal{U}_C$ collects each possible constant input signal taking values from U_C and with maximal duration $[0, T_m]$. The upper bound T_m on their duration ensures a positive lower bound on the probability of sampling an appropriate input duration by the function call `new_state` in Section 5.5.4, where an input signal $\tilde{\upsilon}$ is randomly selected from $\mathcal{U}_C$. Without this upper bound T_m, according to (36), the probability of selecting any finite subintervals from $[0, \infty)$ is 0 because the Lebesgue measure of $[0, \infty)$ is infinity. The selection process involves a random choice of t_m from the interval $[0, T_m]$ and u_C from U_C, from where a constant input signal is constructed as $\tilde{\upsilon} : [0, t_m] \to \{u_C\}$.* ▲

To ensure that each random process in HyRRT, specially within the function calls $\mathcal{T}$.`init`, `random_state`, and `new_state`, returns a suitable sample with positive probability, it is essential to assume that the Lebesgue measure of the sets being sampled is both nonzero and finite.

Assumption 6.15. *The sets X_0, C', D', U_C, U_D, which HyRRT makes random selection from, have finite and positive Lebesgue measure.*

Remark 6.16. *Under Assumption 6.15, it is guaranteed that $\mu(S) < \infty$, where the set S can be any set among X_0, C', D', U_C, and U_D, which HyRRT makes random selection from. Then, if HyRRT is executed uniformly and $\mu(R)$ is positive, from (36), it follows that* $\text{Prob}[s \in R] \in (0, 1]$. ▲

The following assumption is imposed on the flow map f of the hybrid system $\mathcal{H}$ in (2).

Assumption 6.17. *The flow map f is Lipschitz continuous. In particular, there exist $K_x^f, K_u^f \in \mathbb{R}_{>0}$ such that, for each (x_0, x_1, u_0, u_1) such that $(x_0, u_0) \in C$, $(x_0, u_1) \in C$, and $(x_1, u_0) \in C$,*

$$|f(x_0, u_0) - f(x_1, u_0)| \leq K_x^f |x_0 - x_1|$$
$$|f(x_0, u_0) - f(x_0, u_1)| \leq K_u^f |u_0 - u_1|.$$

Remark 6.18. *Assumption 6.17 adheres to the Lipschitz continuity assumption on differential constraints, as outlined in Kleinbort et al. (2018). Assumption 6.17 guarantees that the flow map is Lipschitz continuous for both state and input arguments. This assumption establishes an explicit upper bound, parameterized by both state and input, on the distance between the motion plan and the simulated purely continuous solution pair in the forthcoming Lemma 6.27. Following the methodology in the proof of (Kleinbort et al. 2018, Lemma 3), by ensuring that this upper bound is less than the motion plan's clearance, a range of the input signals in $\mathcal{U}_C$ with positive Lebesgue measure is determined that are capable of simulating a purely continuous solution pair that stays within the motion plan's clearance. As per (36), there is a guaranteed positive probability of randomly sampling an input signal from $\mathcal{U}_C$ that falls within this specified range. This positive probability is instrumental in ensuring that in each Bernoulli trials in the proof of the upcoming Proposition J.1, there is a positive probability of achieving a successful outcome, which eventually leads to the probabilistic completeness guarantee.* ▲

Example 6.19. (Actuated bouncing ball system in Example 3.1, revisited). *In the bouncing ball system, the flow map is defined as $f(x, u) = \begin{bmatrix} x_2 \\ -\gamma \end{bmatrix}$. The flow set is defined as $C := \{(x, u) \in \mathbb{R}^2 \times \mathbb{R} : x_1 \geq 0\}$. For each (x_0, x_1, u_0) such that $(x_0, u_0) \in C$ and $(x_1, u_0) \in C$, we have*

$$|f(x_0, u_0) - f(x_1, u_0)| = \left| \begin{bmatrix} x_{0,2} - x_{1,2} \\ 0 \end{bmatrix} \right| = |x_{0,2} - x_{1,2}| \leq |x_0 - x_1|$$

where $x_{0,2}$ and $x_{1,2}$ denote the second component of x_0 and x_1, respectively. Therefore, $K_x^f = 1 > 0$ is such that $|f(x_0, u_0) - f(x_1, u_0)| \leq K_x^f |x_0 - x_1|$ for each (x_0, x_1, u_0) such that $(x_0, u_0) \in C$ and $(x_1, u_0) \in C$. Now, for each (x_0, u_0, u_1) such that $(x_0, u_0) \in C$ and $(x_0, u_1) \in C$, we have $|f(x_0, u_0) - f(x_0, u_1)| = 0$. Therefore, any $K_u^f > 0$ is such that $|f(x_0, u_0) - f(x_0, u_1)| \leq K_u^f |u_0 - u_1|$ for each (x_0, u_0, u_1) such that $(x_0, u_0) \in C$ and $(x_0, u_1) \in C$. Therefore, the flow map and flow set in Example 3.1 satisfy Assumption 6.17.

The following assumption is imposed on the jump map g of the hybrid system $\mathcal{H}$ in (2).

Assumption 6.20. *The jump map g is such that there exist $K_x^g, K_u^g \in \mathbb{R}_{>0}$ such that, for each $(x_0, u_0) \in D$ and each $(x_1, u_1) \in D$,*

$$|g(x_0, u_0) - g(x_1, u_1)| \leq K_x^g |x_0 - x_1| + K_u^g |u_0 - u_1|.$$

Remark 6.21. *Assumption 6.20 enables to establish an explicit upper bound for the distance between the motion plan and the simulated purely discrete solution pair, as will be further detailed in the Lemma 6.29, which is forthcoming. Adopting the approach from the proof of (Kleinbort et al. 2018, Lemma 3), by keeping this upper bound beneath the motion plan's clearance, a specific range of the input values in $\mathcal{U}_D$, possessing a positive Lebesgue measure, is identified that are capable of simulating a purely discrete solution pair that stays within the motion plan's clearance. As per (36), there is a guaranteed positive probability of randomly sampling an input value from $\mathcal{U}_D$ that falls within this specified range. This positive probability ensures that, in the Bernoulli trials in the proof of the upcoming Proposition J.1, aside from those already covered by Assumption 6.17, each trial has a positive probability of yielding a successful outcome, eventually ensuring the probabilistic completeness of HyRRT.* ▲

Example 6.22. (Actuated bouncing ball example in Example 3.1, revisited). *In Example 3.1, the jump map is defined as $g(x, u) = \begin{bmatrix} x_1 \\ -\lambda x_2 + u \end{bmatrix}$. The jump set is defined as $D := \{(x, u) \in \mathbb{R}^2 \times \mathbb{R} : x_1 = 0, x_2 \leq 0, u \geq 0\}$. For each pair of $(x_0, u_0) \in D$ and $(x_1, u_1) \in D$, we have*

$$\begin{aligned} |g(x_0, u_0) - g(x_1, u_1)| &= \left| \begin{bmatrix} x_{0,1} - x_{1,1} \\ -\lambda(x_{0,2} - x_{1,2}) + u_0 - u_1 \end{bmatrix} \right| \\ &= \left| \begin{bmatrix} x_{0,1} - x_{1,1} \\ -\lambda(x_{0,2} - x_{1,2}) \end{bmatrix} + \begin{bmatrix} 0 \\ u_0 - u_1 \end{bmatrix} \right| \\ &\leq \left| \begin{bmatrix} x_{0,1} - x_{1,1} \\ -\lambda(x_{0,2} - x_{1,2}) \end{bmatrix} \right| + \left| \begin{bmatrix} 0 \\ u_0 - u_1 \end{bmatrix} \right| \\ &\leq \sqrt{1 + \lambda^2} |x_0 - x_1| + |u_0 - u_1|. \end{aligned}$$

Therefore, $K_x^g = \sqrt{1 + \lambda^2}$ and $K_u^g = 1$ are such that $|g(x_0, u_0) - g(x_1, u_1)| \leq K_x^g |x_0 - x_1| + K_u^g |u_0 - u_1|$ for each pair of $(x_0, u_0) \in D$ and $(x_1, u_1) \in D$.

6.3 Probabilistic Completeness Guarantee

In the main result of this paper, the following definition of closeness between hybrid arcs is employed.

Definition 6.23. ((τ, ϵ)-closeness of hybrid arcs). *Given $\tau, \epsilon > 0$, two hybrid arcs ϕ_1 and ϕ_2 are (τ, ϵ)-close if*

1. *for all $(t, j) \in \operatorname{dom} \phi_1$ with $t + j \leq \tau$, there exists s such that $(s, j) \in \operatorname{dom} \phi_2$, $|t - s| < \epsilon$, and $|\phi_1(t, j) - \phi_2(s, j)| < \epsilon$;*
2. *for all $(t, j) \in \operatorname{dom} \phi_2$ with $t + j \leq \tau$, there exists s such that $(s, j) \in \operatorname{dom} \phi_1$, $|t - s| < \epsilon$, and $|\phi_2(t, j) - \phi_1(s, j)| < \epsilon$.*

Our main result shows that HyRRT is probabilistically complete without assuming positive clearance, achieved through properly exploiting the inflation $\mathcal{H}_{\delta_f}$.

Theorem 6.24. *Given a motion planning problem $\mathcal{P} = (X_0, X_f, X_u, (C, f, D, g))$, suppose that Assumptions 6.15, 6.17, and 6.20 are satisfied, and that there exists a compact motion plan (ϕ, υ) to $\mathcal{P}$ with* safety *clearance*

$\delta_s > 0$ and piecewise-constant input. Then, using a complete input library and when executed uniformly (as defined in Definition 6.11) to solve the problem $\mathcal{P}_{\delta_f} = (X_0, X_f, X_u, (C_{\delta_f}, f_{\delta_f}, D_{\delta_f}, g_{\delta_f}))$, where, for some $\delta_f > 0$, $(C_{\delta_f}, f_{\delta_f}, D_{\delta_f}, g_{\delta_f})$ denotes the δ_f-inflation of (C, f, D, g) in (27), the probability that HyRRT fails to find a motion plan $\psi' = (\phi', \upsilon')$ to $\mathcal{P}_{\delta_f}$ such that ϕ' is $(\tilde{\tau}, \tilde{\delta})$-close to ϕ after k iterations is at most $a\exp(-bk)$, where $a, b \in \mathbb{R}_{>0}$, $\tilde{\tau} = \max\{T + J, T' + J'\}$, $(T, J) = \max \operatorname{dom} \psi$, $(T', J') = \max \operatorname{dom} \psi'$, and $\tilde{\delta} = \min\{\delta_s, \delta_f\}$.

The proof of Theorem 6.24 is established as follows. By the positive safety clearance assumption and exploiting the inflation of $\mathcal{H}$ in (27), we establish that (ϕ, υ) is a motion plan to $\mathcal{P}_\delta$ with positive clearance (see Lemma 6.8 in Section 6.1). Then, given that (ϕ, υ) is a motion plan to $\mathcal{P}_\delta$ with positive clearance, we demonstrate that the probability of HyRRT failing to find a motion plan *with positive clearance* is converging to zero as the number of iterations approaches infinity. To demonstrate this result, we first establish that the probabilities of the following probabilistic events are positive:

E1) The function call `nearest_neighbor` in Section 5.5.3 returns a current vertex in the search tree within the clearance of (ϕ, υ) (see Lemma 6.25 in the forthcoming Section 6.4);

E2) If E1 occurs, the function call `new_state` in Section 5.5.4 adds a new vertex and a new edge within the clearance of (ϕ, υ) to the search tree (see Lemma 6.27 and Lemma 6.29 in the forthcoming Section 6.4).

Therefore, the probability that both $E1$ and $E2$ occur, which is denoted as E, resulting in adding a new vertex and a new edge within the clearance of (ϕ, υ) to the search tree, is positive. By the truncation operation in Definition H.1, the compact motion plan (ϕ, υ) is discretized into a finite number, say $m \in \mathbb{N}$, of segments. Each of those segments can be approximated by a solution pair associated with an edge that was added when the event E occurs. Therefore, if E occurs less than m times, then HyRRT will fail to generate a motion plan that is close to (ϕ, υ). Proposition J.1 in the forthcoming Section J demonstrates that the probability that E occurs less than m times, leading to the failure of HyRRT, approaches zero as the number of iterations increases, thereby establishing the property in Theorem 6.24. A proof of Theorem 6.24 following these steps is given in the forthcoming Section 6.5.

6.4 Probabilistic Guarantees on the Function Calls `nearest_neighbor` and `new_state`

We first characterize the probability that a vertex in the search tree that is close to an existing motion plan is selected as v_{cur} by the function call `nearest_neighbor` in Algorithm 2.

Lemma 6.25. *Given a hybrid system $\mathcal{H} = (C, f, D, g)$ with state $x \in \mathbb{R}^n$ and some $\delta \in \mathbb{R}_{>0}$, let $x_c \in \mathbb{R}^n$ be such that $\{x_c\} + \delta\mathbb{B} \subset S$, where S is either C' or D'. Suppose that there exists a vertex v in the search graph $\mathcal{T} = (V, E)$ such that $\overline{x}_v \in \{x_c\} + (2\delta/5)\mathbb{B}$. Denote by v_{cur} the return of the function call `nearest_neighbor` in Algorithm 3. When HyRRT is executed uniformly as defined in Definition 6.11, the probability that $\overline{x}_{v_{cur}} \in \{x_c\} + \delta\mathbb{B}$ is at least $\frac{\zeta_n(\delta/5)^n}{\mu(S)}$, where ζ_n is given in (1) and $\mu(S)$ denotes the Lebesgue measure of set S.*

Proof. See Appendix E.

Remark 6.26. *Lemma 6.25 shows that given $x_c \in \mathbb{R}^n$, when there exists a vertex v such that $\overline{x}_v \in \{x_c\} + (2\delta/5)\mathbb{B}$, then the probability that the function call `nearest_neighbor` selects a vertex that is close to x_c is bounded from below by a positive constant. This lemma is used to provide a positive lower bound over the probability that a vertex that is close enough to the motion plan is returned by the function `nearest_neighbor` in Algorithm 2.* ▲

The following lemma characterizes the probability that, given an initial state near a specific motion plan as input to the function call `new_state`, the simulated solution pair in the flow regime is within the clearance of this motion plan.

Lemma 6.27. *Given a hybrid system $\mathcal{H}$ with state $x \in \mathbb{R}^n$ and input $u \in \mathbb{R}^m$ that satisfies Assumption 6.17 and an input library that satisfies item 1 of Definition 6.13, let $\psi = (\phi, \upsilon)$ be a purely continuous solution pair to $\mathcal{H}$ with clearance $\delta > 0$, $(\tau, 0) = \max \operatorname{dom} \psi$, and constant input function υ. Suppose that $\tau \in (0, T_m]$, where T_m comes from item 1 in Definition 6.13. Let $\psi_{new} = (\phi_{new}, \upsilon_{new})$ be the purely continuous solution pair generated by (24) in the function call `new_state` and initial state $\overline{x}_{v_{cur}} = \phi_{new}(0, 0) \in \{\phi(0, 0)\} + \kappa_1\delta\mathbb{B}$ for some $\kappa_1 \in (0, 1/2)$. When HyRRT is executed uniformly as defined in Definition 6.11, for each $\kappa_2 \in (2\kappa_1, 1)$ and each $\epsilon \in (0, \frac{\kappa_2\delta}{2})$, there exists $p_t \in (0, 1]$ such that*

$$\operatorname{Prob}[E_1 \& E_2] \geq p_t \frac{\zeta_n \left(\max\left\{ \min\left\{ \frac{\frac{\kappa_2\delta}{2} - \epsilon - \exp(K_x^f \tau)\kappa_1\delta}{K_u^f \tau \exp(K_x^f \tau)}, \delta \right\}, 0 \right\} \right)^m}{\mu(U_C)}. \tag{37}$$

where

1. *E_1 denotes the probabilistic event that ϕ and ϕ_{new} are $(\overline{\tau}, \kappa_2\delta)$-close, where $(\tau', 0) = \max \operatorname{dom} \phi_{new}$ and $\overline{\tau} = \max\{\tau, \tau'\}$;*

2. *E_2 denotes the probabilistic event that $x_{new} = \phi_{new}(\tau', 0) \in \{\phi(\tau, 0)\} + \kappa_2\delta\mathbb{B}$, where x_{new} stores the final state of ϕ_{new} in the function call `new_state` as is introduced in Section 5.5.4,*

ζ_n is given in (1), $\mu(U_C)$ denotes the Lebesgue measure of U_C, and K_x^f and K_u^f come from Assumption 6.17.

Proof. See Appendix F.

Remark 6.28. *Since $\kappa_1 \in (0, 1/2)$, $\kappa_2 \in (2\kappa_1, 1)$ and $\delta > 0$, therefore, we have*

$$\kappa_1\delta < \frac{1}{2}\delta \tag{38a}$$
$$\kappa_2\delta < \delta, \tag{38b}$$

ensuring that there is no intersection between ψ_{new} and X_u, which prevents the function call `new_state` *from returning ψ_{new}, as stated in Section 5.5.4.*

To ensure that Lemma 6.27 provides a positive lower bound, the following additional requirement is imposed on κ_1 and κ_2:

$$\frac{\kappa_2}{2} > \kappa_1 \tag{39}$$

Then, there exists $\epsilon \in (0, \frac{\kappa_2\delta}{2})$ satisfying

$$\epsilon < \frac{\kappa_2\delta}{2} - \exp(K_x^f\tau)\kappa_1\delta \tag{40}$$

to guarantee that the first argument in the min *operator in (37) is positive. From (40) and $\epsilon > 0$, any $\tau \in \mathbb{R}_{>0}$ satisfying the following condition suffices for the existence of ϵ satisfying (40):*

$$\tau < \frac{\ln \frac{\kappa_2}{2\kappa_1}}{K_x^f} \tag{41}$$

where, by (39), $\frac{\ln \frac{\kappa_2}{2\kappa_1}}{K_x^f} > \frac{\ln \frac{2\kappa_1}{2\kappa_1}}{K_x^f} = 0$.

Therefore, if κ_1 and κ_2 adhere to the constraints in (38a), (38b) and (39) and if κ_1, κ_2, and τ satisfy the conditions in (41), then the lower bound in (37) is guaranteed to be positive.

The following result characterizes the probability that, given an initial state near a specific motion plan as input to the function call `new_state`, the simulated solution pair computed by the function call `new_state` in the jump regime is within the clearance of this motion plan.

Lemma 6.29. *Given a hybrid system $\mathcal{H}$ with state $x \in \mathbb{R}^n$ and input $u \in \mathbb{R}^m$ that satisfies Assumption 6.20 and an input library that satisfies item 2 of Definition 6.13, let $\psi = (\phi, \upsilon)$ be a purely discrete solution pair to $\mathcal{H}$ with a single jump, i.e.,* $\max \operatorname{dom} \psi = (0, 1)$ *and clearance $\delta > 0$. Let $\psi_{new} = (\phi_{new}, \upsilon_{new})$ be the purely discrete solution pair generated by (25) in the function call* `new_state` *and initial state $\overline{x}_{v_{cur}} = \phi_{new}(0, 0) \in \{\phi(0, 0)\} + \kappa_1\delta\mathbb{B}$ for some positive $\kappa_1 \in (0, 1)$. When HyRRT is executed uniformly as defined in Definition 6.11, for each $\kappa_2 \in (0, 1)$, we have that*

$$\text{Prob}[E] \geq \frac{\zeta_n \left(\max\left\{\min\left\{\frac{(\kappa_2 - K_x^g\kappa_1)\delta}{K_u^g}, \delta\right\}, 0\right\}\right)^m}{\mu(U_D)} \tag{42}$$

where E denotes the probabilistic event that $x_{new} = \phi_{new}(0, 1) \in \{\phi(0, 1)\} + \kappa_2\delta\mathbb{B}$, x_{new} stores the final state of ϕ_{new} in the function call `new_state` *as is introduced in Section 5.5.4, ζ_n is given in (1), $\mu(U_D)$ denotes the Lebesgue measure of U_D, and K_x^g and K_u^g come from Assumption 6.20.*

Proof. See Appendix I.

Remark 6.30. *Since $\kappa_1 \in (0, 1]$, $\kappa_2 \in (0, 1]$ and $\delta > 0$, therefore, we have*

$$\kappa_1\delta < \delta \tag{43a}$$
$$\kappa_2\delta < \delta, \tag{43b}$$

ensuring that there is no intersection between ψ_{new} and X_u, which prevents the function call `new_state` *from returning ψ_{new}.*

To ensure that the lower bound in (42) is positive, an additional requirement is imposed on κ_1 and κ_2:

$$\frac{\kappa_2}{\kappa_1} > K_x^g, \tag{44}$$

Since this implies that $\frac{(\kappa_2 - K_x^g\kappa_1)\delta}{K_u^g} > 0$, the first argument of the min *operator in (42) is positive. Therefore, if κ_1 and κ_2 adhere to the constraints in (43a), (43b), and (44), then the lower bound in (42) is guaranteed to be positive.*

Note that if $\phi_{new}(0, 0) \in \{\phi(0, 0)\} + \kappa_1\delta\mathbb{B}$ and $\phi_{new}(0, 1) \in \{\phi(0, 1)\} + \kappa_2\delta\mathbb{B}$ are satisfied, ϕ and ϕ_{new} are $(1, \max\{\kappa_1\delta, \kappa_2\delta\})$-close. ▲

6.5 Proof of Theorem 6.24

Given that ψ is a motion plan to $\mathcal{P}$ with safety clearance $\delta_s > 0$ and that $\delta_f > 0$, Lemma 6.8 establishes that ψ is a motion plan to $\mathcal{P}_\delta$ with clearance $\delta = \min\{\delta_s, \delta_f\} > 0$. Furthermore, by Proposition J.1, it follows that the probability that HyRRT fails to find $\psi' = (\phi', \upsilon')$ is at most ae^{-bk} and the generated ϕ' is $(\tilde{\tau}, \tilde{\delta})$-close to ϕ where $\tilde{\tau} = \max\{T + J, T' + J'\}$, $(T, J) = \max \operatorname{dom} \psi$, $(T', J') = \max \operatorname{dom} \psi'$, and $\tilde{\delta} = \min\{\delta_s, \delta_f\}$.

7 HyRRT Software Tool for Motion Planning for Hybrid Systems and Examples

Algorithm 2 leads to a software tool[5] to solve the motion planning problems for hybrid systems. This software only requires the motion planning problem data $(X_0, X_f, X_u, (C, f, D, g))$, an input library $(\mathcal{U}_C, \mathcal{U}_D)$, a tunable parameter $p_n \in (0, 1)$, an upper bound K over the iteration number and two constraint sets X_c and X_d. The tool is illustrated in Examples 3.1 and 3.2.

Example 7.1. (Actuated bouncing ball system in Example 3.1, revisited). *This example serves to demonstrate that the HyRRT algorithm is proficient in solving the specific instance of the motion planning problem as illustrated in Example 3.1. The coefficients of the hybrid model in Example 3.1 are given as follows:*

1. *The gravity constant γ is set to* 9.81.
2. *The coefficient of restitution is set to* 0.8.

The inputs fed to the proposed algorithm are given as follows:

1. *The tune parameter p_n is set as* 0.5.
2. *The upper bound K is set to* 1000.

3. *The construction of the input library* $(\mathcal{U}_C, \mathcal{U}_D)$ *adheres to the methodology delineated in* **Input Library Construction Procedure**, *as referenced in Section 5.2. Here, the upper bound* T_m *is explicitly configured to* 0.1. *From the setting of* X_u, *It is imperative to note that each input falling outside the interval* $(0, 5)$ *is unsafe. As a result, both* U_C^s *and* U_D *are restricted to* $(0, 5)$. *Consequently, both* $\mathcal{U}_C$ *and* $\mathcal{U}_D$ *are constructed in compliance with the specifications outlined in the* **Input Library Construction Procedure**.

4. *The constraint sets* X_c *and* X_d *are set as* $X_c = C'$ *and* $X_d = D'$.

5. *The tolerance* ϵ *in (26) is set to* 0.2.

The simulation result is shown in Figure 7. The simulation is implemented in MATLAB software and processed by a 2.2 *GHz Intel Core i7 processor. On average, over the course of* 20 *runs, the simulation is observed to require approximately* 0.72 *seconds for completion and results in the generation of* 34.2 *vertices during the propagation phase.*

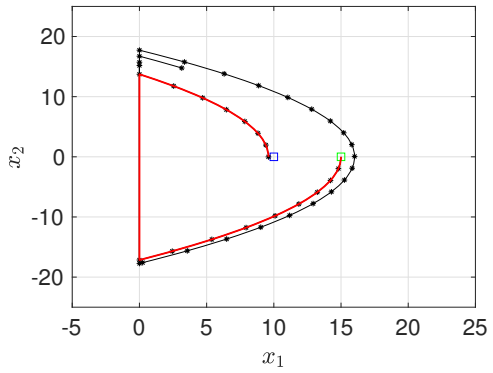


Figure 7. The above shows the search results of HyRRT algorithm to solve the sample motion planning problem in Example 3.1. Green square denotes X_0 and blue square denotes X_f. The states represented by vertices in the search tree are denoted by $\star$'s. The lines between $\star$'s denote the state trajectories of the solution pairs associated with edges in the search tree. The red trajectory denotes the state trajectory of a motion plan to the given motion planning problem.

Example 7.2. (Walking robot system in Example 3.2, revisited). *The coefficients of the hybrid model in Example 3.2 are given as follows:*

1. *The step angle* ϕ_s *is set to* 0.7.

2. $a_1^{\min} = a_2^{\min} = -3$, $a_1^{\max} = a_2^{\max} = 3$, $a_3^{\min} = -0.2$, $a_3^{\max} = 0.2$.

The settings for HyRRT planner are given as follows:

1) The tune parameter p_n *is set as* 0.9 *to encourage the flow regime.*

2) The upper bound K *is set as* 2000.

3) The input library $(\mathcal{U}_C, \mathcal{U}_D)$ *is formulated in accordance with* **Input Library Construction Procedure**. *Specifically, the upper bound* T_m *is set to* 0.4. *Given the constraints imposed by* X_u, *each input deviating from the set* $(-3, 3) \times (-3, 3) \times (-0.2, 0.2)$ *is classified as unsafe. Consequently, both* U_C^s *and* U_D *are restricted to* $(-3, 3) \times (-3, 3) \times (-0.2, 0.2)$. *In light of these constraints, both* $\mathcal{U}_C$ *and* $\mathcal{U}_D$ *are constructed in compliance with the specifications outlined in the* ***Input Library Construction Procedure***.

4) The constraint set X_c *is chosen as* $\{(x, a) \in \mathbb{R}^6 \times \mathbb{R}^3 : h(x) \geq -s\}$ *and* X_d *as* $\{(x, a) \in \mathbb{R}^6 \times \mathbb{R}^3 : h(x) = 0, \omega_p \geq -s\}$ *with a tunable parameter* s *set to* 0, 0.3, 0.5, 1, *and* 2, *such that* $C = X_c|_{s=0} \subsetneq X_c|_{s=0.3} \subsetneq X_c|_{s=0.5} \subsetneq X_c|_{s=1} \subsetneq X_c|_{s=2}$ *and* $D = X_d|_{s=0} \subsetneq X_d|_{s=0.3} \subsetneq X_d|_{s=0.5} \subsetneq X_d|_{s=1} \subsetneq X_d|_{s=2}$.

5) The tolerance ϵ *in (26) is set to* 0.3

The simulation result in Figure 8 shows that HyRRT is able to solve the instance of motion planning problem for the walking robot. The simulation is implemented in MATLAB and processed by a 3.5 *GHz Intel Core i5 processor. Over the course of* 20 *runs, on average, the simulation takes* $63.26/78.3/100.4/183.7/280.8$ *seconds with* s *set to* $0/0.3/0.5/1.0/2.0$, *respectively. Among all the runs conducted, the simulation takes a minimum of* 31.2 *seconds to complete. In contrast, for the same setting, we also developed a brute-force motion planning algorithm that incrementally constructs the search tree in a breadth-first search (BFS) manner. Unlike HyRRT, this algorithm simply discretizes the input library and tries all possible inputs at each vertex. The BFS algorithm takes* 1608.2 *seconds to solve the same problem. The results demonstrate a noteworthy improvement provided by the rapid exploration technique, with average and fastest run computation time improvements of* 96.1% *and* 98.1%, *respectively. It is also observed that as the sets* X_c *and* X_d *grow, HyRRT considers more vertices in solving Problem 4 leading to higher computation time.*

Declaration of conflicting interests

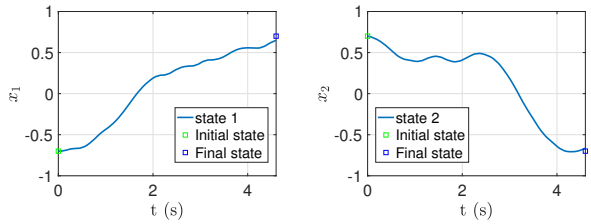
The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The authors disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research is supported by NSF [grants numbers CNS-2039054 and CNS-2111688], by AFOSR [grants numbers FA9550-19-1-0169, FA9550-20- 1-0238, FA9550-23-1-0145, and FA9550-23-1-0313], by AFRL [grant numbers FA8651-22-1-0017 and FA8651-23-1-0004], by ARO [grant number. W911NF-20-1-0253], and by DoD [grant number W911NF-23-1-0158].

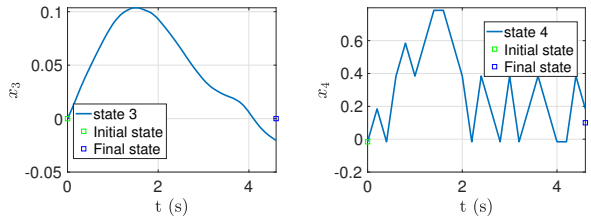
Notes

1. In this paper, we call the event in probability theory *probabilistic event* to distinguish it from the term *event* in the hybrid system theory, which is a synonym for jump. In the probability theory, an event is a set of outcomes of an experiment (a subset of the sample space) to which a probability is assigned Leon-Garcia (1994).



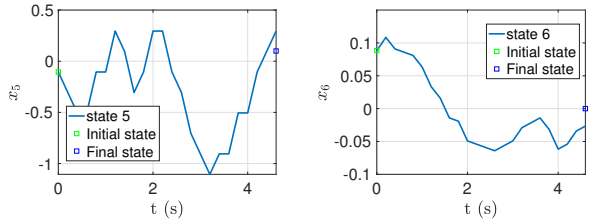
(a) The trajectory of x_1 component of the generated motion plan.

(b) The trajectory of x_2 component of the generated motion plan.



(c) The trajectory of x_3 component of the generated motion plan.

(d) The trajectory of x_4 component of the generated motion plan.



(e) The trajectory of x_5 component of the generated motion plan.

(f) The trajectory of x_6 component of the generated motion plan.

Figure 8. The above shows the trajectories of each state components of the generated motion plan to the sample motion planning problem for biped system. In each figure in Figure 8, green square denotes the corresponding component of X_0 and blue square denotes the corresponding component of X_f.

2. The unsafe set X_u can be used to constrain both the state and the input.
3. In this paper, `rule` $= 3$ is not considered since the input signal $\tilde{v}$ is randomly selected from $\mathcal{U}_C$ which satisfies the forthcoming Assumption 6.13. Therefore, the time duration of the simulation result has been randomized. However, `rule` $= 3$ requires an additional random process to determine whether to proceed with the flow or the jump when the simulation result enters $C \cap D$ leading to a redundant random procedure.
4. The flow set C (and the jump set D) inflates the state x and input u, respectively, yielding their independent ranges. This ensures that the sampling process is a Bernoulli process.
5. Code at https://github.com/HybridSystemsLab/hybridRRT.

References

Billingsley P (2017) *Probability and measure*. John Wiley & Sons.

Branicky MS, Curtiss MM, Levine J and Morgan S (2003a) Sampling-based planning and control. In: *Proceedings of the 12th Yale Workshop on Adaptive and Learning Systems, New Haven, CT*. Citeseer.

Branicky MS, Curtiss MM, Levine JA and Morgan SB (2003b) Rrts for nonlinear, discrete, and hybrid planning and control. In: *42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475)*, volume 1. IEEE, pp. 657–663.

Cheng P (2005) Sampling-based motion planning with differential constraints. Technical report.

Fink E and Veloso M (1994) Prodigy planning algorithm. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF COMPUTER SCIENCE.

Gillespie T (2021) *Fundamentals of vehicle dynamics*. SAE international.

Gipple J (2014) The volume of n-balls. *Rose-Hulman Undergraduate Mathematics Journal* 15(1): 14.

Goldberg K (1994) Completeness in robot motion planning. In: *Workshop on Algorithmic Foundations of Robotics*. pp. 419–429.

Grizzle JW, Abba G and Plestan F (2001) Asymptotically stable walking for biped robots: Analysis via systems with impulse effects. *IEEE Transactions on Automatic Control* 46(1): 51–64.

Huang Q, Yokoi K, Kajita S, Kaneko K, Arai H, Koyachi N and Tanie K (2001) Planning walking patterns for a biped robot. *IEEE Transactions on robotics and automation* 17(3): 280–289.

Karaman S and Frazzoli E (2011) Sampling-based algorithms for optimal motion planning. *The international journal of robotics research* 30(7): 846–894.

Karaman S, Walter MR, Perez A, Frazzoli E and Teller S (2011) Anytime motion planning using the rrt. In: *2011 IEEE International Conference on Robotics and Automation*. IEEE, pp. 1478–1483.

Khatib O (1986) Real-time obstacle avoidance for manipulators and mobile robots. In: *Autonomous Robot Vehicles*. Springer, pp. 396–404.

Kleinbort M, Solovey K, Littlefield Z, Bekris KE and Halperin D (2018) Probabilistic completeness of RRT for geometric and kinodynamic planning with forward propagation. *IEEE Robotics and Automation Letters* 4(2): x–xvi.

Kunz T and Stilman M (2015) Kinodynamic RRTs with fixed time step and best-input extension are not probabilistically complete. In: *Algorithmic Foundations of Robotics XI*. Springer, pp. 233–244.

LaValle SM (1998) Rapidly-exploring random trees: A new tool for path planning .

LaValle SM (2006) *Planning algorithms*. Cambridge University Press.

LaValle SM and Kuffner Jr JJ (2001) Randomized kinodynamic planning. *The international journal of robotics research* 20(5): 378–400.

Leon-Garcia A (1994) *Probability and random processes for electrical engineering*. Pearson Education India.

Li Y, Littlefield Z and Bekris KE (2016) Asymptotically optimal sampling-based kinodynamic planning. *The International Journal of Robotics Research* 35(5): 528–564.

Likhachev M, Ferguson DI, Gordon GJ, Stentz A and Thrun S (2005) Anytime dynamic A*: An anytime, replanning algorithm,. In: *ICAPS*, volume 5. pp. 262–271.

Liu S, Atanasov N, Mohta K and Kumar V (2017) Search-based motion planning for quadrotors using linear quadratic minimum time control. In: *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, pp.

2872–2879.

Qureshi AH, Miao Y, Simeonov A and Yip MC (2020) Motion planning networks: Bridging the gap between learning-based and classical motion planners. *IEEE Transactions on Robotics* 37(1): 48–66.

Rajamani R (2011) *Vehicle dynamics and control*. Springer Science & Business Media.

Rybus T (2020) Point-to-point motion planning of a free-floating space manipulator using the rapidly-exploring random trees (rrt) method. *Robotica* 38(6): 957–982.

Sanfelice R, Copp D and Nanez P (2013) A toolbox for simulation of hybrid systems in matlab/simulink: Hybrid equations (hyeq) toolbox. In: *Proceedings of the 16th international conference on Hybrid systems: computation and control*. pp. 101–106.

Sanfelice RG (2021) Hybrid feedback control .

Short BE and Sanfelice RG (2018) A hybrid predictive control approach to trajectory tracking for a fully actuated biped. In: *2018 Annual American Control Conference (ACC)*. IEEE, pp. 3526–3531.

Solovey K, Salzman O and Halperin D (2016) Finding a needle in an exponential haystack: Discrete rrt for exploration of implicit roadmaps in multi-robot motion planning. *The International Journal of Robotics Research* 35(5): 501–513.

Teng S, Hu X, Deng P, Li B, Li Y, Ai Y, Yang D, Li L, Xuanyuan Z, Zhu F et al. (2023) Motion planning for autonomous driving: The state of the art and future perspectives. *IEEE Transactions on Intelligent Vehicles* .

Wu A, Sadraddini S and Tedrake R (2020) R3t: Rapidly-exploring random reachable set tree for optimal kinodynamic planning of nonlinear hybrid systems. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 4245–4251.

Zha J and Mueller MW (2021) Exploiting collisions for sampling-based multicopter motion planning. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 7943–7949.

A Proof of Proposition 5.2

A.1 Theoretical Tools to Prove Proposition 5.2

Lemma A.1. *Given two functions, denoted ψ_1 and ψ_2 and defined on hybrid time domains, the domain of their concatenation $\psi = \psi_1|\psi_2$, is also a hybrid time domain.*

Proof. Given that $\mathrm{dom}\,\psi_1$ and $\mathrm{dom}\,\psi_2$ are hybrid time domains as per Definition 2.1, for each $(T_1, J_1) \in \mathrm{dom}\,\psi_1$ and each $(T_2, J_2) \in \mathrm{dom}\,\psi_2$, it follows that

$$\mathrm{dom}\,\psi_1 \cap ([0, T_1] \times \{0, 1, ..., J_1\}) = \cup_{j=0}^{J_1}([t_j^1, t_{j+1}^1], j)$$

for some finite sequence of times $0 = t_0^1 \leq t_1^1 \leq t_2^1 \leq ... \leq t_{J_1+1}^1 = T_1$ and

$$\mathrm{dom}\,\psi_2 \cap ([0, T_2] \times \{0, 1, ..., J_2\}) = \cup_{j=0}^{J_2}([t_j^2, t_{j+1}^2], j)$$

for some finite sequence of times $0 = t_0^2 \leq t_1^2 \leq t_2^2 \leq ... \leq t_{J_2+1}^2 = T_2$.

Given that $\mathrm{dom}\,\psi = \mathrm{dom}\,\psi_1 \cup (\mathrm{dom}\,\psi_2 + \{(T, J)\})$ where $(T, J) = \max \mathrm{dom}\,\psi_1$, it follows that for each $(T', J') \in \mathrm{dom}\,\psi$,

$$\begin{aligned}
&\mathrm{dom}\,\psi \cap ([0, T'] \times \{0, 1, ..., J'\}) \\
&= (\mathrm{dom}\,\psi_1 \cup (\mathrm{dom}\,\psi_2 + \{(T, J)\})) \cap ([0, T'] \times \{0, 1, ..., J'\}) \\
&= (\mathrm{dom}\,\psi_1 \cap ([0, T'] \times \{0, 1, ..., J'\})) \\
&\cup (\mathrm{dom}\,\psi_2 + \{(T, J)\}) \cap ([0, T'] \times \{0, 1, ..., J'\})
\end{aligned} \tag{45}$$

Since $(T', J') \in \mathrm{dom}\,\psi$ and $\mathrm{dom}\,\psi = \mathrm{dom}\,\psi_1 \cup (\mathrm{dom}\,\psi_2 + \{(T, J)\})$, (T', J') satisfies one of the following two conditions:

1. $(T', J') \in \mathrm{dom}\,\psi_1$ such that $T' \leq T$ and $J' \leq J$;
2. $(T', J') \in (\mathrm{dom}\,\psi_2 + \{(T, J)\})$ such that $T' \geq T$ and $J' \geq J$.

For the first item, (45) can be written as:

$$\begin{aligned}
&\mathrm{dom}\,\psi \cap ([0, T'] \times \{0, 1, ..., J'\}) \\
&= (\mathrm{dom}\,\psi_1 \cap ([0, T'] \times \{0, 1, ..., J'\})) \\
&\cup (\mathrm{dom}\,\psi_2 + \{(T, J)\}) \cap ([0, T'] \times \{0, 1, ..., J'\}) \\
&= \cup_{j=0}^{J'}([t_j^1, t_{j+1}^1], j) \cup \emptyset = \cup_{j=0}^{J'}([t_j^1, t_{j+1}^1], j)
\end{aligned}$$

for some finite sequence of times $0 = t_0^1 \leq t_1^1 \leq t_2^1 \leq ... \leq t_{J'+1}^1 = T'$.

For the second item, (45) can be written as

$$\begin{aligned}
&\mathrm{dom}\,\psi \cap ([0, T'] \times \{0, 1, ..., J'\}) \\
&= (\mathrm{dom}\,\psi_1 \cap ([0, T'] \times \{0, 1, ..., J'\})) \\
&\cup (\mathrm{dom}\,\psi_2 + \{(T, J)\}) \cap ([0, T'] \times \{0, 1, ..., J'\}) \\
&= (\mathrm{dom}\,\psi_1 \cap ([0, T] \times \{0, 1, ..., J\})) \\
&\cup (\mathrm{dom}\,\psi_2 + \{(T, J)\}) \cap ([T, T'] \times \{J, J+1, ..., J'\}) \\
&= (\mathrm{dom}\,\psi_1 \cap ([0, T] \times \{0, 1, ..., J\})) \\
&\cup (\mathrm{dom}\,\psi_2 \cap ([0, T' - T] \times \{0, 1, ..., J' - J\}) + \{(T, J)\}) \\
&= \cup_{j=0}^{J}([t_j^1, t_{j+1}^1], j) \cup (\cup_{j=0}^{J'-J}([t_j^2, t_{j+1}^2], j) + \{(T, J)\}) \\
&= \cup_{j=0}^{J}([t_j^1, t_{j+1}^1], j) \cup (\cup_{j=0}^{J'-J}([t_j^2 + T, t_{j+1}^2 + T], j + J))
\end{aligned} \tag{46}$$

for some finite sequence of times $0 = t_0^1 \leq t_1^1 \leq t_2^1 \leq ... \leq t_{J+1}^1 = T$ and $T = t_0^2 + T \leq t_1^2 + T \leq t_2^2 + T \leq ... \leq t_{J'-J+1}^2 + T = T'$. Hence, a finite sequence $\{t_i\}_{i=0}^{J'}$ can be constructed as follows such that $0 = t_0 \leq t_1 \leq t_2 \leq ... \leq t_{J'} = T'$.

$$t_i = \begin{cases} t_i^1 & i \leq J \\ t_{i-J}^2 + T & J + 1 \leq i \leq J' \end{cases}$$

Therefore, for each $(T', J') \in \mathrm{dom}\,\psi$, there always exists a finite sequence of $\{t_i\}_{i=0}^{J'}$ such that

$$\mathrm{dom}\,\psi \cap ([0, T'] \times \{0, 1, ..., J'\}) = \cup_{j=0}^{J'}([t_j, t_{j+1}], j).$$

Consequently, this establishes $\mathrm{dom}\,\psi$ as a hybrid time domain. □

Definition A.2. Lebesgue measurable set and Lebesgue measurable function. *A set $S \subset \mathbb{R}$ is said to be* Lebesgue measurable *if it has positive Lebesgue measure $\mu(S)$, where $\mu(S) = \sum_k (b_k - a_k)$ and (a_k, b_k)'s are all of the disjoint open intervals in S. Then $s : S \to \mathbb{R}^{n_s}$ is said to be a* Lebesgue measurable function *if for every open set $\mathcal{U} \subset \mathbb{R}^{n_s}$ then set $\{r \in S : s(r) \in \mathcal{U}\}$ is Lebesgue measurable.*

Lemma A.3. *Given two hybrid inputs, denoted υ_1 and υ_2, their concatenation $\upsilon = \upsilon_1|\upsilon_2$, is such that for each $j \in \mathbb{N}$, $t \mapsto \upsilon(t,j)$ is Lebesgue measurable on the interval $I_\upsilon^j := \{t : (t,j) \in \operatorname{dom}\upsilon\}$.*

Proof. We show that for each $j \in \mathbb{N}$, the function $t \mapsto \upsilon(t,j)$ satisfies conditions outlined in Definition A.2 over the interval $I_\upsilon^j := \{t : (t,j) \in \operatorname{dom}\upsilon\}$.

In line with Definition 5.1, we have $\operatorname{dom}\psi = \operatorname{dom}\psi_1 \cup (\operatorname{dom}\psi_2 + \{(T,J)\})$ where $(T,J) = \max \operatorname{dom}\psi_1$. For each $j \in \mathbb{N}$, j and J satisfy one of the following three conditions:

I $j < J$;

II $j > J$;

III $j = J$.

In each case, we proceed to demonstrate that the function $t \mapsto \upsilon(t,j)$ is Lebesgue measurable.

I In the case where $j < J$, given that υ_1 is a hybrid input, the function $t \mapsto \upsilon_1(t,j)$ is Lebesgue measurable over the interval $I_{\upsilon_1}^j := \{t : (t,j) \in \operatorname{dom}\upsilon_1\}$. According to Definition 5.1, it follows that $\upsilon(t,j) = \upsilon_1(t,j)$ holds for all $(t,j) \in \operatorname{dom}\upsilon_1$ where $j < J$. Consequently, the function $t \mapsto \upsilon(t,j) = \upsilon_1(t,j)$ is Lebesgue measurable on the interval $I_\psi^j = I_{\psi_1}^j$ when $j < J$.

II In the case where $j > J$, given that υ_2 is a hybrid input, the function $t \mapsto \upsilon_2(t,j)$ is Lebesgue measurable on the interval $I_{\psi_2}^j$. According to Definition 5.1, it is established that $\upsilon(t,j) = \upsilon_2(t-T, j-J)$ for each $(t,j) \in \operatorname{dom}\upsilon_2 + \{(T,J)\}$ where $j > J$. Therefore, the function $t \mapsto \upsilon(t,j) = \upsilon_2(t-T, j-J)$ is Lebesgue measurable on the interval $I_\psi^j = I_{\psi_2}^{j-J} + \{T\}$ when $j > J$.

III In the case where $j = J$, according to Definition 5.1, it follows that

$$\upsilon(t,J) = \begin{cases} \upsilon_1(t,J) & t \in I_{\psi_1}^J \backslash \{T\} \\ \upsilon_2(t-T,0) & t \in I_{\psi_2}^0 + \{T\}. \end{cases}$$

Note that $t \mapsto \upsilon_1(t,J)$ is Lebesgue measurable over the interval $I_{\upsilon_1}^J$ and $t \mapsto \upsilon_2(t,0)$ is Lebesgue measurable on the interval $I_{\upsilon_2}^0$. As per Definition A.2, it is established that

(a) for each open set $\mathcal{U}_1 \subset \{\upsilon_1(t,J) \in \mathbb{R}^m : t \in I_{\upsilon_1}^J\}$, the set $\{t \in I_{\upsilon_1}^J : \upsilon_1(t,J) \in \mathcal{U}_1\}$ is Lebesgue measurable;

(b) for each open set $\mathcal{U}_2 \subset \{\upsilon_2(t,0) \in \mathbb{R}^m : t \in I_{\upsilon_2}^0\}$, the set $\{t \in I_{\upsilon_2}^0 : \upsilon_2(t,0) \in \mathcal{U}_2\}$ is Lebesgue measurable.

Consequently, for each open set $\mathcal{U} \subset \{\upsilon(t,J) \in \mathbb{R}^m : t \in I_\upsilon^J\}$, given that

$$\mathcal{U} \subset \{\upsilon(t,J) \in \mathbb{R}^m : t \in I_\upsilon^J\} = \{\upsilon_1(t,J) \in \mathbb{R}^m : t \in I_{\upsilon_1}^J\} \cup \{\upsilon_2(t,0) \in \mathbb{R}^m : t \in I_{\upsilon_2}^0\},$$

therefore, at least one of $\widetilde{\mathcal{U}}_1 := \mathcal{U} \cap \{\upsilon_1(t,J) \in \mathbb{R}^m : t \in I_{\upsilon_1}^J\}$ and $\widetilde{\mathcal{U}}_2 := \mathcal{U} \cap \{\upsilon_2(t,0) \in \mathbb{R}^m : t \in I_{\upsilon_2}^0\}$ is an open set. According to (a) and (b) above, it follows that at least one of the sets $\{t \in I_{\upsilon_1}^J : \upsilon_1(t,J) \in \widetilde{\mathcal{U}}_1\}$ and $\{t \in I_{\upsilon_2}^0 : \upsilon_2(t,0) \in \widetilde{\mathcal{U}}_2\}$ is Lebesgue measurable. Consequently, the set $\{t \in I_\upsilon^J : \upsilon(t,J) \in \mathcal{U}\} = \{t \in I_{\upsilon_1}^J : \upsilon_1(t,J) \in \widetilde{\mathcal{U}}_1\} \cup (\{t \in I_{\upsilon_2}^0 : \upsilon_2(t,0) \in \widetilde{\mathcal{U}}_2\} + \{T\})$ is Lebesgue measurable and, hence, the function $t \mapsto \upsilon(t,J)$ is Lebesgue measurable.

Therefore, $t \mapsto \upsilon(t,j)$ is Lebesgue measurable on the interval I_υ^j for all $j \in \mathbb{N}$. □

Definition A.4. Locally essentially bounded function. *A function $s : S \to \mathbb{R}^{n_s}$ is said to be* locally essentially bounded *if for each $r \in S$ there exists an open neighborhood $\mathcal{U}$ of r such that s is bounded almost everywhere[6] on $\mathcal{U}$; i.e., there exists $c \geq 0$ such that $|s(r)| \leq c$ for almost all $r \in \mathcal{U} \cap S$.*

Lemma A.5. *Given two hybrid inputs, denoted υ_1 and υ_2, their concatenation $\upsilon = \upsilon_1|\upsilon_2$, is such that for each $j \in \mathbb{N}$, $t \mapsto \upsilon(t,j)$ is locally essentially bounded on the interval $I_\upsilon^j := \{t : (t,j) \in \operatorname{dom}\upsilon\}$.*

Proof. We show that for all $j \in \mathbb{N}$, the function $t \mapsto \upsilon(t,j)$ satisfies conditions in Definition A.4 on the interval $I_\upsilon^j := \{t : (t,j) \in \operatorname{dom}\upsilon\}$.

According to Definition 5.1, $\operatorname{dom}\psi = \operatorname{dom}\upsilon_1 \cup (\operatorname{dom}\upsilon_2 + \{(T,J)\})$ where $(T,J) = \max \operatorname{dom}\upsilon_1$. For each $j \in \mathbb{N}$, j and J satisfy one of the following three conditions:

I $j < J$;

II $j > J$;

III $j = J$.

In each case, we proceed to demonstrate that the function $t \mapsto \upsilon(t,j)$ is locally essentially bounded.

I In the case where $j < J$, given that υ_1 is a hybrid input, the function $t \mapsto \upsilon_1(t,j)$ is locally essentially bounded on the interval $I_{\upsilon_1}^j := \{t : (t,j) \in \operatorname{dom}\upsilon_1\}$. As per Definition 5.1, it follows that $\upsilon(t,j) = \upsilon_1(t,j)$ for all $(t,j) \in \operatorname{dom}\upsilon_1$ where $j < J$. Consequently, the function $t \mapsto \upsilon(t,j) = \upsilon_1(t,j)$ is locally essentially bounded on the interval $I_\psi^j = I_{\psi_1}^j$ where $j < J$.

II In the case where $j > J$, given that υ_2 is a hybrid input, the function $t \mapsto \upsilon_2(t,j)$ is locally essentially bounded on the interval $I_{\upsilon_2}^j$. As per Definition 5.1, it follows that $\upsilon(t,j) = \upsilon_2(t-T, j-J)$ for all $(t,j) \in (\operatorname{dom}\upsilon_2 + \{(T,J)\})$ where $j > J$. Consequently, the function $t \mapsto \upsilon(t,j) = \upsilon_2(t-T, j-J)$ is locally essentially bounded on the interval $I_\upsilon^j = I_{\upsilon_2}^{j-J} + \{T\}$ where $j > J$.

III In the case where $j = J$, according to Definition 5.1, it follows that

$$\upsilon(t, J) = \begin{cases} \upsilon_1(t, J) & t \in I^J_{\upsilon_1} \backslash \{T\} \\ \upsilon_2(t - T, 0) & t \in I^0_{\upsilon_2} + \{T\}. \end{cases}$$

Given that $t \mapsto \upsilon_1(t, j)$ is locally bounded on the interval $I^J_{\upsilon_1}$ and $t \mapsto \upsilon_2(t, 0)$ is locally bounded on the interval $I^0_{\upsilon_2}$, according to Definition A.4, it follows that

(a) for each $r \in I^J_{\upsilon_1}$, there exists a neighborhood $\mathcal{U}_1$ of r such that there exists $c_1 \geq 0$ such that $|\upsilon_1(r)| \leq c_1$ for almost all $r \in \mathcal{U}_1 \cap I^J_{\upsilon_1}$;

(b) for each $r \in I^0_{\upsilon_2}$, there exists a neighborhood $\mathcal{U}_2$ of r such that there exists $c_2 \geq 0$ such that $|\upsilon_2(r)| \leq c_2$ for almost all $r \in \mathcal{U}_1 \cap I^0_{\upsilon_2}$.

Then, for each $r \in I^J_\upsilon = I^J_{\upsilon_1} \cup (I^0_{\upsilon_2} + \{T\})$, the following two cases are considered:

(a) If $r \in I^J_{\upsilon_1}$, then there exists a neighborhood $\mathcal{U}_1$ of r such that there exists $c_1 \geq 0$ such that $|\upsilon(r)| \leq c_1$ for almost all $r \in \mathcal{U}_1 \cap I^J_\upsilon$.

(b) If $r \in I^0_{\upsilon_2} + \{T\}$, then there exists a neighborhood $\mathcal{U}_2$ of r such that there exists $c_2 \geq 0$ such that $|\upsilon_2(r)| \leq c_2$ for almost all $r \in \mathcal{U}_1 \cap (I^0_{\upsilon_2} + \{T\})$.

Therefore, the function $t \mapsto \upsilon(t, J)$ is locally essentially bounded on the interval I^J_υ.

Therefore, $t \mapsto \upsilon(t, j)$ is locally bounded on the interval I^j_υ for all $j \in \mathbb{N}$. □

Lemma A.6. *Given two hybrid inputs, denoted υ_1 and υ_2, then their concatenation $\upsilon = \upsilon_1|\upsilon_2$ is also a hybrid input.*

Proof. In this proof, we show that $\upsilon = \upsilon_1|\upsilon_2$ satisfies the conditions in Definition 2.3 as follows:

1. $\operatorname{dom}\upsilon$ is hybrid time domain;
2. for each $j \in \mathbb{N}, t \to \upsilon(t, j)$ is Lebesgue measurable on the interval $I^j_\upsilon := \{t : (t, j) \in \operatorname{dom}\upsilon\}$;
3. for each $j \in \mathbb{N}$, $t \to \upsilon(t, j)$ is locally essentially bounded on the interval $I^j_\upsilon := \{t : (t, j) \in \operatorname{dom}\upsilon\}$.

Given that υ_1 and υ_2 are hybrid inputs, therefore, Lemma A.1 ensures that $\operatorname{dom}\upsilon$ is hybrid time domain. Lemma A.3 guarantees that υ is Lebesgue measurable on the interval I^j_υ. Lemma A.5 establishes that υ is locally essentially bounded on the interval I^j_υ. Consequently, it follows that υ is a hybrid input. □

Definition A.7. Absolutely continuous function and locally absolutely continuous function. *(Sanfelice 2021, Definition A.20) A function $s : [a, b] \to \mathbb{R}^n$ is said to be* absolutely continuous *if for each $\varepsilon > 0$ there exists $\delta > 0$ such that for each countable collection of disjoint subintervals $[a_k, b_k]$ of $[a, b]$ such that $\sum_k (b_k - a_k) \leq \delta$, it follows that $\sum_k |s(b_k) - s(a_k)| \leq \varepsilon$. A function is said to be locally absolutely continuous if $r \mapsto s(r)$ is absolutely continuous on each compact subinterval of $\mathbb{R}_{\geq 0}$.*

Lemma A.8. *Given two hybrid arcs, denoted ϕ_1 and ϕ_2, such that $\phi_1(T, J) = \phi_2(0, 0)$, where $(T, J) = \max\operatorname{dom}\phi_1$, their concatenation $\phi = \phi_1|\phi_2$, is such that for each $j \in \mathbb{N}$, $t \mapsto \phi(t, j)$ is locally absolutely continuous on the interval $I^j_\phi := \{t : (t, j) \in \operatorname{dom}\phi\}$.*

Proof. We show that for all $j \in \mathbb{N}$, the function $t \mapsto \phi(t, j)$ satisfies the conditions in Definition A.7.

According to Definition 5.1, we have $\operatorname{dom}\phi = \operatorname{dom}\phi_1 \cup (\operatorname{dom}\phi_2 + \{(T, J)\})$ where $(T, J) = \max\operatorname{dom}\phi_1$. It follows that

$$I^j_\phi = \begin{cases} I^j_{\phi_1} & \text{if } j < J \\ I^J_{\phi_1} \cup (I^0_{\phi_2} + \{T\}) & \text{if } j = J \\ I^{j-J}_{\phi_2} + T & \text{if } j > J. \end{cases}$$

For each $j \in \mathbb{N}$, j and J satisfy one of the following three conditions:

I $j < J$;

II $j > J$;

III $j = J$.

In each case, we proceed to demonstrate that the function $t \mapsto \phi(t, j)$ is absolutely continuous.

I In the case where $j < J$, given that ϕ_1 is a hybrid arc, according to Definition 2.4, the function $t \mapsto \phi_1(t, j)$ is locally absolutely continuous on the interval $I^j_{\phi_1} = I^j_\phi$ where $j < J$. According to Definition 5.1, it follows that $\phi(t, j) = \phi_1(t, j)$ for all $t \in I^j_{\phi_1} = I^j_\phi$ where $j < J$. Therefore, the function $t \mapsto \phi(t, j) = \phi_1(t, j)$ is locally absolutely continuous on the interval I^j_ϕ where $j < J$.

II In the case where $j > J$, given that ϕ_2 is a hybrid arc, according to Definition 2.4, the function $t \mapsto \phi_2(t - T, j - J)$ is locally absolutely continuous on the interval $I^{j-J}_{\phi_2} + \{T\} = I^j_\phi$ where $j > J$. According to Definition 5.1, it follows that $\phi(t, j) = \phi_2(t - T, j - J)$ for all $t \in I^{j-J}_{\phi_2} + \{T\} = I^j_\phi$ where $j > J$. Therefore, the function $t \mapsto \phi_2(t - T, j - J) = \phi(t, j)$ is locally absolutely continuous on the interval $I^j_\phi = I^{j-J}_{\phi_2} + \{T\}$ where $j > J$.

III In the case where $j = J$, given that $I^j_\phi = I^J_{\phi_1} \cup (I^0_{\phi_2} + \{T\})$ and that the function $t \mapsto \phi(t, J)$ is locally absolutely continuous on the intervals $I^J_{\phi_1}$ and $(I^0_{\phi_2} + \{T\})$, respectively, therefore, it follows the following:

(a) for each $\varepsilon_1 > 0$ there exists $\delta_1 > 0$ such that for each countable collection of disjoint subintervals $[a^1_{k_1}, b^1_{k_1}]$ of $I^J_{\phi_1}$ such that $\sum_{k_1}(b^1_{k_1} - a^1_{k_1}) \leq \delta_1$, it follows that $\sum_{k_1} |\phi(b^1_{k_1}) - \phi(a^1_{k_1})| \leq \varepsilon_1$;

(b) for each $\varepsilon_2 > 0$ there exists $\delta_2 > 0$ such that for each countable collection of disjoint subintervals $[a^2_{k_2}, b^2_{k_2}]$ of $I^0_{\phi_2} + \{T\}$ such that $\sum_{k_2}(b^2_{k_2} - a^2_{k_2}) \leq \delta_2$, it follows that $\sum_{k_2} |\phi(b^2_{k_2}) - \phi(a^2_{k_2})| \leq \varepsilon_2$.

Then for each $\varepsilon > 0$, assume that $\delta > 0$ exists such that for each countable collection of disjoint subintervals $[a_k, b_k]$ of $I^J_{\phi_1} \cup (I^0_{\phi_2} + \{T\})$ such that $\sum_k (b_k - a_k) \leq \delta$, it follows that $\sum_k |\phi(b^2_k) - \phi(a^2_k)| \leq \varepsilon$.

The collections of intervals $\{[a_k, b_k]\}_k$ intersecting with the intervals $I^J_{\phi_1}$ and $(I^0_{\phi_2} + \{T\})$ are defined as follows:

1 $\{[a^1_{k_1}, b^1_{k_1}]\}_{k_1} := \{[a_k, b_k]\}_k \cap I^J_{\phi_1}$;

2 $\{[a^2_{k_2}, b^2_{k_2}]\}_{k_2} := \{[a_k, b_k]\}_k \cap (I^0_{\phi_2} + \{T\})$.

Subsequently, define $\varepsilon_1 := \sum_{k_1} |\phi(b^1_{k_1}) - \phi(a^1_{k_1})|$ and $\varepsilon_2 := \sum_{k_2} |\phi(b^2_{k_2}) - \phi(a^2_{k_2})|$. Given that

$$\begin{aligned}\sum_k |\phi(b^2_k) - \phi(a^2_k)| = &\sum_{k_1} |\phi(b^1_{k_1}) - \phi(a^1_{k_1})| \\ &+ \sum_{k_2} |\phi(b^2_{k_2}) - \phi(a^2_{k_2})| \\ &+ |\phi_2(0,0) - \phi_1(T,J)|,\end{aligned}$$

and that $\phi_2(0,0) = \phi_1(T,J)$, it follows that

$$\sum_k |\phi(b^2_k) - \phi(a^2_k)| = \varepsilon_1 + \varepsilon_2 + 0.$$

According to (a) and (b) above, for each $\varepsilon_1 > 0$ and each $\varepsilon_2 > 0$, there exists $\delta_1 > 0$ and $\delta_2 > 0$ such that $\sum_{k_1}(b^1_{k_1} - a^1_{k_1}) \leq \delta_1$ and $\sum_{k_2}(b^2_{k_2} - a^2_{k_2}) \leq \delta_2$ hold, respectively. Therefore, given that $\sum_k (b_k - a_k) = \sum_{k_1}(b^1_{k_1} - a^1_{k_1}) + \sum_{k_2}(b^2_{k_2} - a^2_{k_2}) \leq \delta_1 + \delta_2 = \delta$ and the existence of δ_1 and δ_2, the existence of $\delta = \delta_1 + \delta_2$ is established.

Therefore, the function $t \mapsto \phi(t,j)$ is locally absolutely continuous on the interval I^j_ϕ for all $j \in \mathbb{N}$. □

Lemma A.9. *Given two hybrid arcs, denoted ϕ_1 and ϕ_2, such that $\phi_1(T,J) = \phi_2(0,0)$, where $(T,J) = \max \operatorname{dom} \phi_1$, then their concatenation $\phi = \phi_1 | \phi_2$ is also a hybrid arc.*

Proof. We show that $\phi = \phi_1|\phi_2$ satisfies the conditions in Definition 2.4 as follows:

1. We show that $\operatorname{dom} \phi$ is hybrid time domain;
2. We show that, for each $j \in \mathbb{N}$, $t \mapsto \phi(t,j)$ is locally absolutely continuous on the interval $I^j_\phi := \{t : (t,j) \in \operatorname{dom} \phi\}$.

Note that ϕ_1 and ϕ_2 are functions defined on hybrid time domains. Therefore, Lemma A.1 guarantees that $\operatorname{dom} \phi$ is hybrid time domain. Lemma A.8 ensures that for each $j \in \mathbb{N}$, $t \mapsto \phi(t,j)$ is locally absolutely continuous on the interval $I^j_\phi := \{t : (t,j) \in \operatorname{dom} \phi\}$. Therefore, it is established that ϕ is a hybrid arc. □

Lemma A.10. *Given two solution pairs, denoted ψ_1 and ψ_2 such that $\psi_2(0,0) \in C$ if both $I^J_{\psi_1}$ and $I^0_{\psi_2}$ have nonempty interior, where, for each $j \in \{0, J\}$, $I^j_\psi = \{t : (t,j) \in \operatorname{dom} \psi\}$ and $(T,J) = \max \operatorname{dom} \psi_1$, then their concatenation $\psi = \psi_1|\psi_2$ satisfies that for each $j \in \mathbb{N}$ such that I^j_ψ has nonempty interior $\operatorname{int}(I^j_\psi)$, $\psi = (\phi, \upsilon)$ satisfies*

$$(\phi(t,j), \upsilon(t,j)) \in C$$

for all $t \in \operatorname{int} I^j_\phi$.

Proof. We show that for all $j \in \mathbb{N}$ such that I^j_ψ has nonempty interior, $\psi(t,j) \in C$ for all $t \in \operatorname{int} I^j_\psi$.

According to Definition 5.1, $\operatorname{dom} \psi = \operatorname{dom} \psi_1 \cup (\operatorname{dom} \psi_2 + \{(T,J)\})$ where $(T,J) = \max \operatorname{dom} \psi_1$. Therefore,

$$I^j_\psi = \begin{cases} I^j_{\psi_1} & \text{if } j < J \\ I^J_{\psi_1} \cup (I^0_{\psi_2} + \{T\}) & \text{if } j = J \\ I^{j-J}_{\psi_2} + \{T\} & \text{if } j > J \end{cases}$$

I In the case of all $j < J$ such that $I^j_{\psi_1}$ has nonempty interior, given that ψ_1 is a solution pair to $\mathcal{H}$, according to Definition 2.6, we have $\psi_1(t,j) \in C$ for all $t \in \operatorname{int} I^j_{\psi_1}$. As per Definition 5.1, it follows that $\psi(t,j) = \psi_1(t,j)$ for all $t \in \operatorname{int} I^j_{\psi_1}$. Therefore, we have $\psi(t,j) = \psi_1(t,j) \in C$ for all $t \in \operatorname{int} I^j_\psi = \operatorname{int} I^j_{\psi_1}$.

II In the case of all $j > J$ such that $I^{j-J}_{\psi_2} + \{T\}$ has nonempty interior, given that ψ_2 is a solution pair to $\mathcal{H}$, according to Definition 2.6, we have $\psi_2(t,j) \in C$ for all $t \in \operatorname{int} I^j_{\psi_2}$. As per Definition 5.1, it follows that $\psi(t,j) = \psi_2(t-T, j-J)$ for all $t \in I^{j-J}_{\psi_2} + \{T\}$. Therefore, we have $\psi(t,j) = \psi_2(t-T, j-J) \in C$ for all $t \in \operatorname{int} I^j_\psi = I^{j-J}_{\psi_2} + \{T\}$.

III In the case where $j = J$ and $I^J_\psi = I^J_{\psi_1} \cup (I^0_{\psi_2} + \{T\})$ has nonempty interior. If one of the intervals $I^J_{\psi_1}$ and $I^0_{\psi_2} + \{T\}$ has nonempty interior while the other does not, the validation process aligns with the previously discussed cases I and II.

If both $I^J_{\psi_1}$ and $(I^0_{\psi_2} + \{T\})$ has nonempty interior, given that

$$\psi(t,j) = \psi_1(t,j) \in C \quad \forall t \in \operatorname{int} I^J_{\psi_1}$$

and that

$$\psi(t,j) = \psi_2(t-T, j-J) \in C \forall t \in \operatorname{int} I^0_{\psi_2} + \{T\},$$

it can be concluded that $\psi(t,j) \in C$ for all $t \in \operatorname{int} I^j_\psi \cup (\operatorname{int} I^0_{\psi_2} + \{T\})$.

In addition, since the point $\psi_2(0,0)$ is assumed to belong to C in this case, then $\psi(t,J) \in C$ holds for all $t \in \operatorname{int} I^J_\psi = \operatorname{int}(I^J_{\psi_1} \cup (I^0_{\psi_2} + \{T\}))$.

Hence, it is established that for all $j \in \mathbb{N}$ such that I^j_ψ has nonempty interior, $\psi(t,j) \in C$ for each $t \in \operatorname{int} I^j_\psi$. □

Lemma A.11. *Given two solution pairs, denoted ψ_1 and ψ_2, then their concatenation $\psi = \psi_1|\psi_2$ satisfies that for each $j \in \mathbb{N}$ such that I^j_ψ has nonempty interior* $\operatorname{int}(I^j_\psi)$, *$\psi = (\phi, \upsilon)$ satisfies*

$$\frac{d}{dt}\phi(t,j) = f(\phi(t,j), \upsilon(t,j))$$

for almost all $t \in I^j_\phi$.

Proof. We show that for all $j \in \mathbb{N}$ such that I^j_ψ has nonempty interior and almost all $t \in I^j_\psi$, $\frac{\mathrm{d}}{\mathrm{d}t}\phi = f(\phi(t,j), \upsilon(t,j))$.

According to Definition 5.1, it follows that $\operatorname{dom}\psi = \operatorname{dom}\psi_1 \cup (\operatorname{dom}\psi_2 + \{(T,J)\})$ where $(T,J) = \max\operatorname{dom}\psi_1$. Therefore, we have

$$I^j_\psi = \begin{cases} I^j_{\psi_1} & \text{if } j < J \\ I^J_{\psi_1} \cup (I^0_{\psi_2} + \{T\}) & \text{if } j = J \\ I^{j-J}_{\psi_2} + \{T\} & \text{if } j > J. \end{cases}$$

I In the case of all $j < J$ such that $I^j_{\psi_1}$ has nonempty interior, given that ψ_1 is a solution pair, it follows that $\frac{\mathrm{d}}{\mathrm{d}t}\phi_1(t,j) = f(\phi_1(t,j), \upsilon_1(t,j))$ for almost all $t \in \operatorname{int} I^j_{\psi_1}$. According to Definition 5.1, we have

$$(\phi(t,j), \upsilon(t,j)) = (\phi_1(t,j), \upsilon_1(t,j))$$

for all $(t,j) \in \operatorname{dom}\psi_1 \backslash (T,J)$. Therefore, it is established that

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\phi(t,j) &= \frac{\mathrm{d}}{\mathrm{d}t}\phi_1(t,j) = f(\phi_1(t,j), \upsilon_1(t,j)) \\ &= f(\phi(t,j), \upsilon(t,j))\end{aligned}$$

for almost all $t \in \operatorname{int} I^j_\psi$.

II In the case of all $j > J$ such that $I^{j-J}_{\psi_2}$ has nonempty interior, given that ψ_2 is a solution pair and $I^{j-J}_{\psi_2}$ has nonempty interior, it follows that $\frac{\mathrm{d}}{\mathrm{d}t}\phi_2(t,j) = f(\phi_2(t-T, j-J), \upsilon_2(t-T, j-J))$ for almost all $t \in \operatorname{int} I^{j-J}_{\psi_2} + \{T\}$. According to Definition 5.1, we have

$$(\phi(t,j), \upsilon(t,j)) = (\phi_2(t-T, j-J), \upsilon_2(t-T, j-J))$$

for all $t \in I^{j-J}_{\psi_2} + \{T\}$. Therefore, it is established that

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\phi(t,j) &= \frac{\mathrm{d}}{\mathrm{d}t}\phi_2(t-T, j-J) \\ &= f(\phi_2(t-T, j-J), \upsilon_2(t-T, j-J)) \\ &= f(\phi(t,j), \upsilon(t,j))\end{aligned}$$

for almost all $t \in \operatorname{int} I^j_\phi$.

III In the case where $j = J$ and $I^J_\psi = I^J_{\psi_1} \cup (I^0_{\psi_2} + \{T\})$ has nonempty interior, If one of the intervals $I^J_{\psi_1}$ and $I^0_{\psi_2} + \{T\}$ has nonempty interior while the other does not, the validation process aligns with the previously discussed cases I and II. If both $I^J_{\psi_1}$ and $(I^0_{\psi_2} + \{T\})$ have nonempty interior, it follows that $I^J_\psi = I^J_{\psi_1} \cup (I^0_{\psi_2} + \{T\})$. Given that $\frac{\mathrm{d}}{\mathrm{d}t}\phi(t,J) = f(\phi(t,J), \upsilon(t,J))$ for almost all $t \in I^J_{\psi_1}$ and $t \in I^0_{\psi_2} + \{T\}$, it is established that $\frac{\mathrm{d}}{\mathrm{d}t}\phi(t,j) = f(\phi(t,j), \upsilon(t,j))$ for $t \in I^J_\psi = I^J_{\psi_1} \cup (I^0_{\psi_2} + \{T\})$.

Therefore, $\frac{\mathrm{d}}{\mathrm{d}t}\phi(t,j) = f(\phi(t,j), \upsilon(t,j))$ holds for almost all $t \in I^j_\psi$. □

Lemma A.12. *Given a hybrid system $\mathcal{H} = (C, f, D, g)$ and two solution pairs to $\mathcal{H}$, denoted ψ_1 and ψ_2, such that $\phi_1(T,J) = \phi_2(0,0)$, where $(T,J) = \max\operatorname{dom}\phi_1$, then their concatenation $\psi = \psi_1|\psi_2 = (\phi, \upsilon)$ satisfies that for all $(t,j) \in \operatorname{dom}(\phi, \upsilon)$ such that $(t, j+1) \in \operatorname{dom}(\phi, \upsilon)$,*

$$\begin{aligned}(\phi(t,j), \upsilon(t,j)) &\in D \\ \phi(t, j+1) &= g(\phi(t,j), \upsilon(t,j)).\end{aligned}$$

Proof. We show that for all $(t,j) \in \operatorname{dom}\psi$ such that $(t, j+1) \in \operatorname{dom}\psi = \operatorname{dom}\psi_1 \cup (\operatorname{dom}\psi_2 + \{(T,J)\})$ where $(T,J) = \max\operatorname{dom}\psi_1$,

$$\begin{aligned}\psi(t,j) &\in D \\ \phi(t, j+1) &= g(\phi(t,j), \upsilon(t,j)).\end{aligned}$$

1. In the case where (t,j) is such that $(t,j) \in \operatorname{dom}\psi_1$ and $(t, j+1) \in \operatorname{dom}\psi_1$, since ψ_1 is a solution pair to $\mathcal{H}$, it follows that

$$\begin{aligned}\psi_1(t,j) &\in D \\ \phi_1(t, j+1) &= g(\phi_1(t,j), \upsilon_1(t,j)).\end{aligned}$$

Given that $(t, j+1) \in \operatorname{dom}\psi_1$, consequently, we have $(t,j) \neq (T,J)$. Since $\psi(t,j) = \psi_1(t,j)$ for all $(t,j) \in \operatorname{dom}\psi_1 \backslash (T,J)$, it is established that

$$\begin{aligned}\psi(t,j) &= \psi_1(t,j) \in D \\ \phi(t, j+1) &= \phi_1(t, j+1) \\ &= g(\phi_1(t,j), \upsilon_1(t,j)) \\ &= g(\phi(t,j), \upsilon(t,j)).\end{aligned}$$

2. In the case where (t,j) is such that $(t,j) \in \operatorname{dom}\psi_2 + \{(T,J)\}$ and $(t, j+1) \in \operatorname{dom}\psi_2 + \{(T,J)\}$, according to Definition 5.1, it follows that $(t-T, j-J) \in \operatorname{dom}\psi_2$ and $(t-T, j-J+1) \in \operatorname{dom}\psi_2$. Therefore, since ψ_2 is a solution pair, it is established that

$$\begin{aligned}&\psi_2(t-T, j-J) \in D \\ &\phi_2(t-T, j-J+1) \\ &\quad = g(\phi_2(t-T, j-J), \upsilon_2(t-T, j-J)).\end{aligned}$$

Note that $\psi(t,j) = \psi_2(t-T, j-J)$ for all $(t-T, j-J) \in \operatorname{dom}\psi_2$. Therefore, the following is satisfied:

$$\begin{aligned}\psi(t,j) &= \psi_2(t-T, j-J) \in D \\ \phi(t, j+1) &= \phi_2(t-T, j-J+1) \\ &= g(\phi_2(t-T, j-J), \upsilon_2(t-T, j-J)) \\ &= g(\phi(t,j), \upsilon(t,j)).\end{aligned}$$

3. In the special case where (t,j) is such that $(t,j) \in \operatorname{dom}\psi_1$ and $(t,j+1) \in \operatorname{dom}\psi_2 + \{(T,J)\}$, this scenario corresponds to the situation where $(t,j) = (T,J-1) \in \operatorname{dom}\psi_1$ and $(t,j+1) = (T,J) \in \operatorname{dom}\psi_2 + (T,J)$. Given that $\phi_1(T,J) = \phi_2(0,0)$, the following relationships can be deduced:

$$\begin{aligned}\phi_1(T,J) &= \phi_2(0,0) = \phi(T,J)\\ \phi_1(T,J-1) &= \phi(T,J-1)\\ \upsilon_1(T,J-1) &= \upsilon(T,J-1)\end{aligned}$$

Given that ψ_1 is a solution pair to $\mathcal{H}$ and considering that

$$(T,J-1) \in \operatorname{dom}\psi_1 \quad (T,J) \in \operatorname{dom}\psi_1,$$

according to Definition 2.6, it follows that

$$\begin{aligned}\phi_1(T,J) &= g(\phi_1(T,J-1), \upsilon_1(T,J-1))\\ \psi_1(T,J-1) &= (\phi_1(T,J-1), \upsilon_1(T,J-1)) \in D.\end{aligned}$$

Therefore, we establish that

$$\begin{aligned}\phi(T,J) &= \phi_2(0,0)\\ &= \phi_1(T,J)\\ &= g(\phi_1(T,J-1), \upsilon_1(T,J-1))\\ &= g(\phi(T,J-1), \upsilon(T,J-1))\end{aligned}$$

and

$$\begin{aligned}&(\phi(T,J-1), \upsilon(T,J-1))\\ &= (\phi_1(T,J-1), \upsilon_1(T,J-1)) \in D.\end{aligned}$$

□

A.2 Proof of Proposition 5.2

Proof. We show that the concatenation result $\psi = (\phi,\upsilon) = (\phi_1|\phi_2, \upsilon_1|\upsilon_2)$ satisfies the conditions in Definition 2.6. Namely, we show in this proof that each of the following items holds:

1. υ is a hybrid input;
2. ϕ is a hybrid arc;
3. $(\phi(0,0), \upsilon(0,0)) \in \overline{C} \cup D$;
4. $\operatorname{dom}\phi = \operatorname{dom}\upsilon (= \operatorname{dom}(\phi,\upsilon))$;
5. for each $j \in \mathbb{N}$ such that I_ψ^j has nonempty interior $\operatorname{int}(I_\psi^j)$, (ϕ,υ) satisfies $(\phi(t,j), \upsilon(t,j)) \in C$ for all $t \in \operatorname{int} I_\psi^j$;
6. for each $j \in \mathbb{N}$ such that I_ψ^j has nonempty interior $\operatorname{int}(I_\psi^j)$, (ϕ,υ) satisfies $\frac{\mathrm{d}}{\mathrm{d}t}\phi(t,j) = f(\phi(t,j), \upsilon(t,j))$ for almost all $t \in I_\psi^j$;
7. for all $(t,j) \in \operatorname{dom}(\phi,\upsilon)$ such that $(t,j+1) \in \operatorname{dom}(\phi,\upsilon)$, $(\phi(t,j), \upsilon(t,j)) \in D$ and $\phi(t,j+1) = g(\phi(t,j), \upsilon(t,j))$.

Given that ψ_1 is compact, therefore, it is feasible to concatenate ψ_2 to ψ_1. Next, we show that each of the items listed above holds as follows:

1. Lemma A.6 guarantees that υ is a hybrid input.
2. Given that $\phi_1(T,J) = \phi_2(0,0)$, therefore, Lemma A.9 guarantees that ϕ is a hybrid arc.
3. Given that ψ_1 is a solution pair to $\mathcal{H}$, it follows that $(\phi_1(0,0), \upsilon_1(0,0)) \in \overline{C} \cup D$. According to Definition 5.1, we have
$$(\phi(t,j), \upsilon(t,j)) = (\phi_1(t,j), \upsilon_1(t,j))$$
for all $(t,j) \in \operatorname{dom}\psi_1$. Since $(0,0) \in \operatorname{dom}\psi_1$, it is established that
$$\begin{aligned}\psi(0,0) &= (\phi(0,0), \upsilon(0,0))\\ &= (\phi_1(0,0), \upsilon_1(0,0)) \in \overline{C} \cup D.\end{aligned}$$
4. Given that ψ_1 and ψ_2 are solution pairs to $\mathcal{H}$, according to Definition 2.6, it follows that
$$\operatorname{dom}\psi_1 = \operatorname{dom}\phi_1 = \operatorname{dom}\upsilon_1,$$
$$\operatorname{dom}\psi_2 = \operatorname{dom}\phi_2 = \operatorname{dom}\upsilon_2.$$
According to Definition 5.1, it follows that
$$\operatorname{dom}\phi = \operatorname{dom}\phi_1 \cup (\operatorname{dom}\phi_2 + \{(T,J)\})$$
and
$$\operatorname{dom}\upsilon = \operatorname{dom}\upsilon_1 \cup (\operatorname{dom}\upsilon_2 + \{(T,J)\}).$$
Therefore, the following is established:
$$\begin{aligned}\operatorname{dom}\phi &= \operatorname{dom}\phi_1 \cup (\operatorname{dom}\phi_2 + \{(T,J)\})\\ &= \operatorname{dom}\upsilon_1 \cup (\operatorname{dom}\upsilon_2 + \{(T,J)\}) = \operatorname{dom}\upsilon.\end{aligned}$$
5. Given that $\psi_2(0,0) \in C$ if both $I_{\psi_1}^J$ and $I_{\psi_2}^0$ have nonempty interior, therefore, Lemma A.10 ensures that for each $j \in \mathbb{N}$ such that I_ψ^j has nonempty interior $\operatorname{int}(I_\psi^j)$, (ϕ,υ) satisfies
$$(\phi(t,j), \upsilon(t,j)) \in C$$
for all $t \in \operatorname{int} I_\psi^j$.
6. Lemma A.11 guarantees that for each $j \in \mathbb{N}$ such that I_ψ^j has nonempty interior $\operatorname{int}(I_\psi^j)$, (ϕ,υ) satisfies
$$\frac{\mathrm{d}}{\mathrm{d}t}\phi(t,j) = f(\phi(t,j), \upsilon(t,j))$$
for almost all $t \in I_\psi^j$.
7. Given that $\phi_1(T,J) = \phi_2(0,0)$, therefore, Lemma A.12 guarantees that for all $(t,j) \in \operatorname{dom}(\phi,\upsilon)$ such that $(t,j+1) \in \operatorname{dom}(\phi,\upsilon)$,
$$\begin{aligned}&(\phi(t,j), \upsilon(t,j)) \in D\\ &\phi(t,j+1) = g(\phi(t,j), \upsilon(t,j)).\end{aligned}$$

In conclusion, the concatenation result ψ satisfies all the conditions in Definition 2.6 and, hence, is a solution pair to $\mathcal{H}$.

B A Computational Framework to Simulate Continuous Dynamics

B.1 Numerical integration scheme model

A general numerical integration scheme can be modeled as

$$F_s(x_k, x_{k+1}, \tilde{v}, f, t) = 0 \tag{47}$$

where s denotes the step size, x_k denotes the state at the kth step, x_{k+1} denotes the approximation of the state at the next step, $\tilde{v}$ denotes the applied input signal, f denotes the flow map, and t denotes the time at current step, namely, $t = ks$. Two examples on how (47) models explicit and implicit numerical integration scheme are given as follows.

Example B.1. *(Forward Euler method) Given the differential equation $\dot{x} = f(x, \tilde{v})$ with initial state x_0 and input signal $\tilde{v} \in \mathcal{U}_C$, the integration scheme using the forward Euler method is*

$$x_{k+1} = x_k + sf(x, \tilde{v}(t)) \tag{48}$$

Following (47), (48) can be modeled as

$$F_s(x_k, x_{k+1}, \tilde{v}, f, t) := x_k + sf(x_k, \tilde{v}(t)) - x_{k+1} \tag{49}$$

and x_{k+1} can be obtained by solving (47).

Example B.2. *(Backward Euler method) Given the differential equation $\dot{x} = f(x, \tilde{v})$ with initial state x_0 and input signal $\tilde{v} \in \mathcal{U}_C$, the integration scheme using backward Euler method is*

$$x_{k+1} = x_k + sf(x_{k+1}, \tilde{v}(t+s)) \tag{50}$$

Following (47), (50) can be modeled as

$$F_s(x_k, x_{k+1}, \tilde{v}, f, t) := x_k + sf(x_{k+1}, \tilde{v}(t+s)) - x_{k+1}$$

and x_{k+1} can be obtained by solving (47).

B.2 Zero-crossing detection model to approximate $\hat{t}$

When the priority option $\texttt{rule} = 1$, following Step 2 above, $\hat{t}$ in (20) is the first time when $(\hat{\phi}, \tilde{v})$ leaves C. When the priority option $\texttt{rule} = 2$, $\hat{t}$ is the first time when $(\hat{\phi}, \tilde{v})$ leaves $C \backslash D$.

Given a set $S \subset \mathbb{R}^n \times \mathbb{R}^m$, a zero-crossing function $h : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ to detect whether the pair (ϕ, υ) is in set S is given by

1. $h(x, u) > 0$ for all $(x, u) \in \operatorname{int} S$,
2. $h(x, u) < 0$ for all $(x, u) \in \operatorname{int}((\mathbb{R}^n \times \mathbb{R}^m) \backslash S)$,
3. $h(x, u) = 0$ for all $(x, u) \in \partial S$,
4. h is continuous over $\mathbb{R}^n \times \mathbb{R}^m$.

When $\texttt{rule} = 1$, this function can be used with $S = C \backslash D$ to determine $\hat{t}$ as the first time when $(\hat{\phi}, \tilde{v})$ leaves $C \backslash D$ by checking zero crossings of $h_f(\hat{\phi}, \tilde{v})$. When $\texttt{rule} = 2$, this function can be used with $S = C$ to determine $\hat{t}$ as the first time when $(\hat{\phi}, \tilde{v})$ leaves C by checking zero crossings of $h(\hat{\phi}, \tilde{v})$. Next, we illustrate this approach by constructing a zero-crossing function for the bouncing ball system in Example 3.1.

Example B.3. (Example 3.1, revisited). *In the bouncing ball system in Example 3.1, the flow set is $C = \{(x, u) \in \mathbb{R}^2 \times \mathbb{R} : x_1 \geq 0\}$ and the jump set is $D = \{(x, u) \in \mathbb{R}^2 \times \mathbb{R} : x_1 = 0, x_2 \leq 0, u \geq 0\}$. Then,*

$$h(x, u) := x_1 \quad \forall (x, u) \in \mathbb{R}^n \times \mathbb{R}^m.$$

is a zero-crossing function for the detection of solutions leaving C. The same function can be used to detect solutions leaving $C \backslash D$ since $D \subset \partial C$ which implies that $\operatorname{int}(C \backslash D) = \operatorname{int} C$ and $\operatorname{int}((\mathbb{R}^n \times \mathbb{R}^m) \backslash (C \backslash D)) = \operatorname{int}((\mathbb{R}^n \times \mathbb{R}^m) \backslash C)$.

The zero-crossing detection algorithm that approximates $\hat{t}$ can be modeled as

$$\hat{t} = t_{zcd}(\hat{\phi}, \tilde{v}, \texttt{rule}, C, D) \tag{51}$$

where the function t_{zcd} employs a zero-crossing detection function for $C \backslash D$ or C, depending on the value of $\texttt{rule}$. We set $\hat{t}$ as -1 when no zero-crossing is detected.

B.3 A computational framework to simulate continuous dynamics

A computational approach to simulate the continuous dynamics of $\mathcal{H}$ is given in Algorithm 4.

Algorithm 4 A computational framework to simulate the continuous dynamics $\texttt{continuous\_simulator}$

function CONTINUOUS_SIMULATOR($C, f, D, \texttt{rule}, s, x_0, \tilde{v}$)
  Set $\phi_0 \leftarrow x_0$, $\hat{t} \leftarrow 0$.
  while $t \leq \bar{t}(\tilde{v}) \& \hat{t} = -1$ **do**
    Compute $\hat{\phi}$ by solving $F_s(\phi_0, \hat{\phi}, \tilde{v}, f, t) = 0$.
    $\hat{t} \leftarrow t_{zcd}(\hat{\phi}, \tilde{v}, \texttt{rule}, C, D)$.
    if $\hat{t} = -1$ **then**
      $\phi \leftarrow \phi | \hat{\phi}$, $\upsilon \leftarrow \upsilon | \hat{v}$.
      $\phi_0 \leftarrow \hat{\phi}(\hat{T}, 0)$ where $(\hat{T}, 0) = \max \operatorname{dom} \hat{\phi}$.
    else
      $\phi \leftarrow \phi | \hat{\phi}([0, \hat{t}], 0)$, $\upsilon \leftarrow \upsilon | \hat{v}([0, \hat{t}], 0)$.
      $\phi_0 \leftarrow \hat{\phi}(\hat{t}, 0)$.
    end if
  end while
  return (ϕ, υ).
end function

C Proof of Proposition 6.7

C.1 Theoretical Tools to Prove Proposition 6.7

We show that $\psi = (\phi, \upsilon)$ in Proposition 6.7 satisfies the conditions in Definition 2.6 for $(C_\delta, f_\delta, D_\delta, g_\delta)$ where $(C_\delta, f_\delta, D_\delta, g_\delta)$ is the δ-inflation of hybrid system of (C, f, D, g) for each $\delta > 0$, as defined in Definition 6.5.

Lemma C.1. *Given a hybrid system $\mathcal{H} = (C, f, D, g)$ and its δ-inflation $\mathcal{H}_\delta = (C_\delta, f_\delta, D_\delta, g_\delta)$ for each $\delta > 0$, if $\psi = (\phi, \upsilon)$ is a solution pair to $\mathcal{H}$, then ψ is also a solution pair to $\mathcal{H}_\delta$.*

Proof. We show that ψ satisfies all the conditions in Definition 2.6 for $(C_\delta, f_\delta, D_\delta, g_\delta)$, namely,

1. ϕ is a hybrid arc and υ is a hybrid input;
2. $(\phi(0,0),\upsilon(0,0)) \in \overline{C_\delta} \cup D_\delta$;
3. For each $j \in \mathbb{N}$ such that I_ϕ^j has nonempty interior $\operatorname{int}(I_\phi^j)$, (ϕ,υ) satisfies
$$(\phi(t,j),\upsilon(t,j)) \in C_\delta$$
for all $t \in \operatorname{int} I_\phi^j$, and
$$\frac{\mathrm{d}}{\mathrm{d}t}\phi(t,j) = f_\delta(\phi(t,j),\upsilon(t,j))$$
for almost all $t \in I_\phi^j$.
4. For all $(t,j) \in \operatorname{dom}(\phi,\upsilon)$ such that $(t,j+1) \in \operatorname{dom}(\phi,\upsilon)$,
$$\begin{aligned}(\phi(t,j),\upsilon(t,j)) &\in D_\delta\\ \phi(t,j+1) &= g_\delta(\phi(t,j),\upsilon(t,j)).\end{aligned} \tag{52}$$

Next, we show that each item above is satisfied.

1. Given that $\psi = (\phi,\upsilon)$ is a solution pair to $\mathcal{H}$, therefore, ϕ is a hybrid arc and υ is a hybrid input.
2. Since $\psi = (\phi,\upsilon)$ is a solution pair to $\mathcal{H}$, then $(\phi(0,0),\upsilon(0,0)) \in \overline{C} \cup D$. Note that $C \subset C_\delta$, $D \subset D_\delta$. Therefore,
$$(\phi(0,0),\upsilon(0,0)) \in \overline{C} \cup D \subset \overline{C_\delta} \cup D_\delta.$$
3. For all $j \in \mathbb{N}$ such that $I^j = \{t : (t,j) \in \operatorname{dom}(\phi,\upsilon)\}$ has nonempty interior,
   (a) $(\phi(t,j),\upsilon(t,j)) \in C \subset C_\delta$ for all $t \in \operatorname{int} I^j$.
   (b) for almost all $t \in I^j$,
$$\dot\phi(t,j) = f(\phi(t,j),\upsilon(t,j)) = f_\delta(\phi(t,j),\upsilon(t,j)).$$
4. For all $(t,j) \in \operatorname{dom}(\phi,\upsilon)$ such that $(t,j+1) \in \operatorname{dom}(\phi,\upsilon)$,
$$\begin{aligned}(\phi(t,j),\upsilon(t,j)) &\in D \subset D_\delta\\ \phi(t,j+1) &= g(\phi(t,j),\upsilon(t,j))\\ &= g_\delta(\phi(t,j),\upsilon(t,j))\end{aligned}$$

Since ψ satisfies all the items in Definition 2.6 for data $(C_\delta, f_\delta, D_\delta, g_\delta)$, it is established that ψ is a solution pair to $(C_\delta, f_\delta, D_\delta, g_\delta)$. □

C.2 Proof of Proposition 6.7

Proof. We show that the motion plan $\psi = (\phi,\upsilon)$ to $\mathcal{P}$ satisfies each condition in Problem 1 for the data $\mathcal{P}_\delta = (X_0, X_f, X_u, (C_\delta, f_\delta, D_\delta, g_\delta))$, namely,

1. $\phi(0,0) \in X_0$;
2. (ϕ,υ) is a solution pair to $\mathcal{H}_\delta$;
3. (T,J) is such that $\phi(T,J) \in X_f$, namely, the solution belongs to the final state set at hybrid time (T,J);
4. $(\phi(t,j),\upsilon(t,j)) \notin X_u$ for each $(t,j) \in \operatorname{dom}(\phi,\upsilon)$ such that $t+j \le T+J$.

Given that the initial state set X_0, the final state set X_f, and the unsafe set X_u are the same in both $\mathcal{P}$ and $\mathcal{P}_\delta$ and that ψ is a motion plan to $\mathcal{P}$, items 1, 3, and 4 are satisfied for free. Note that, by construction, every solution pair to $\mathcal{H}$ is a solution pair to $\mathcal{H}_\delta$. In fact, by Lemma C.1, (ϕ,υ) is a solution pair to $\mathcal{H}_\delta$, namely, item 2 is satisfied. Therefore, all the items are satisfied and (ϕ,υ) is a motion plan to $\mathcal{P}_\delta$. □

D Proof of Lemma 6.8

To prove Lemma 6.8, we first establish the following result.

Lemma D.1. *Given a hybrid system defined as $\mathcal{H} = (C,f,D,g)$ and its δ_f-inflation, denoted $\mathcal{H}_{\delta_f} = (C_{\delta_f}, f_{\delta_f}, D_{\delta_f}, g_{\delta_f})$ and defined in (27), if a state input pair $(y,v) \in \mathbb{R}^n \times \mathbb{R}^m$ is such that $(y,v) \in X$, where X can be either C or D, then for each $\delta' \in [0,\delta_f]$, we have $(\{y\}+\delta'\mathbb{B}, \{v\}+\delta'\mathbb{B}) \subset X_{\delta_f}$.*

Proof. Note that, from (27), C_{δ_f} and D_{δ_f} are captured by the following set by choosing $X = C$ or $X = D$, respectively:

$$\begin{aligned}X_{\delta_f} :=\{(x,u) \in \mathbb{R}^n \times \mathbb{R}^m : \exists (y,v) \in X : x \in \{y\}+\delta_f\mathbb{B},\\ u \in \{v\}+\delta_f\mathbb{B}\}.\end{aligned} \tag{53}$$

Then, to prove $(\{y\}+\delta'\mathbb{B}, \{v\}+\delta'\mathbb{B}) \subset X_{\delta_f}$ where $(y,v) \in X$, we show that for each $(y',v') \in (\{y\}+\delta'\mathbb{B}, \{v\}+\delta'\mathbb{B})$, we have $(y',v') \in X_{\delta_f}$.

Given that $(y',v') \in (\{y\}+\delta'\mathbb{B}, \{v\}+\delta'\mathbb{B})$ where $(y,v) \in X$, by (53), it follows that

$$(y',v') \in X_{\delta'}.$$

Since $\delta' \in [0,\delta_f]$, it follows that $\{y\}+\delta'\mathbb{B} \subset \{y\}+\delta_f\mathbb{B}$ and $\{v\}+\delta'\mathbb{B} \subset \{v\}+\delta_f\mathbb{B}$. Then, by (53), we have that

$$X_{\delta'} \subset X_{\delta_f}.$$

Therefore, for each $(y',v') \in (\{y\}+\delta'\mathbb{B}, \{v\}+\delta'\mathbb{B})$ and each $\delta' \in [0,\delta_f]$ where $(y,v) \in X$, we have

$$(y',v') \in X_{\delta'} \subset X_{\delta_f}.$$

Hence, we have

$$(\{y\}+\delta'\mathbb{B}, \{v\}+\delta'\mathbb{B}) \subset X_{\delta_f}.$$

To establish Lemma 6.8, we proceed as follows. Proposition 6.7 establishes that ψ, which is a motion to problem $\mathcal{P}$, is also a motion plan to problem $\mathcal{P}_{\delta_f}$. We need to show that ψ has clearance $\min\{\delta_s,\delta_f\}$.

Since $\psi = (\phi,\upsilon)$ is a solution pair to $\mathcal{H} = (C,f,D,g)$, then for all $j \in \mathbb{N}$ such that $I^j = \{t : (t,j) \in \operatorname{dom}(\phi,\upsilon)\}$ has nonempty interior, we have

$$(\phi(t,j),\upsilon(t,j)) \in C$$

for all $t \in \operatorname{int} I^j$. Therefore, by Lemma D.1, for each $\delta' \in [0,\delta_f]$ and each $j \in \mathbb{N}$ such that I^j has nonempty interior, it

follows that

$$(\{\phi(t,j)\}+\delta'\mathbb{B},\{v(t,j)\}+\delta'\mathbb{B})\subset C_{\delta_f}.$$

Hence, item 1 in Definition 6.2 is satisfied.

Similarly, since ψ is a solution pair to $\mathcal{H}=(C,f,D,g)$, for each $(t,j)\in\operatorname{dom}\psi$ such that $(t,j+1)\in\operatorname{dom}\psi$, we have

$$(\phi(t,j),v(t,j))\in D.$$

Therefore, by Lemma D.1, for each $\delta'\in[0,\delta_f]$ and each $(t,j)\in\operatorname{dom}\psi$ such that $(t,j+1)\in\operatorname{dom}\psi$, it follows that

$$(\{\phi(t,j)\}+\delta_f\mathbb{B},\{v(t,j)\}+\delta_f\mathbb{B})\subset D_{\delta_f}.$$

Hence, item 2 in Definition 6.2 is also satisfied.

Therefore, δ_f satisfies all the conditions in Definition 6.2 and, hence, is the dynamics clearance. Then, by Definition 6.4, the clearance of ψ is $\min\{\delta_s,\delta_f\}$. □

E Proof of Lemma 6.25

The proof closely parallels that of Lemma 4 as presented in Kleinbort et al. (2018). Suppose there exists a vertex, denoted z, in the search graph $\mathcal{T}=(V,E)$ such that $\overline{x}_z\in S$ and $\overline{x}_z\notin\{x_c\}+\delta\mathbb{B}$, as otherwise it is immediate that $\overline{x}_{v_{cur}}\in\{x_c\}+\delta\mathbb{B}$ because all the vertices in $\mathcal{T}$ belong to $\{x_c\}+\delta\mathbb{B}$. We show that, under the conditions in Lemma 6.25, if the sampling point x_{rand} returned by the function call `random_state` is such that $x_{rand}\in\{x_c\}+(\delta/5)\mathbb{B}$, it follows that $\overline{x}_{v_{cur}}\in\{x_c\}+\delta\mathbb{B}$.

Given that $x_{rand}\in\{x_c\}+(\delta/5)\mathbb{B}$, it follows that

$$|x_{rand}-x_c|\leq\frac{\delta}{5}. \tag{54}$$

Since $\overline{x}_v\in\{x_c\}+(2\delta/5)\mathbb{B}$, then we have

$$|\overline{x}_v-x_c|\leq\frac{2\delta}{5}.$$

Therefore, by the triangle inequality, it follows that

$$|x_{rand}-\overline{x}_v|\leq|x_{rand}-x_c|+|x_c-\overline{x}_v|\leq\frac{3\delta}{5}.$$

Since $\overline{x}_z\notin x_c+\delta\mathbb{B}$, then $|\overline{x}_z-x_c|>\delta$. Using (54), by the triangle inequality, it follows that

$$\begin{aligned}\delta<|\overline{x}_z-x_c|&\leq|\overline{x}_z-x_{rand}|+|x_{rand}-x_c|\\&\leq|\overline{x}_z-x_{rand}|+\frac{\delta}{5}.\end{aligned}$$

namely, $|\overline{x}_z-x_{rand}|>4\delta/5$.

Since $|\overline{x}_v-x_{rand}|\leq3\delta/5$ and $|\overline{x}_z-x_{rand}|>4\delta/5$, we have that x_{rand} is closer to v than to z. This implies that z, which can be any vertex that is not in $\{x_c\}+\delta\mathbb{B}$, will not be reported as v_{cur} by the function call `nearest_neighbor`.

If v is reported as v_{cur}, then, since $\overline{x}_v\in\{x_c\}+2\delta/5\mathbb{B}$, we have that $\overline{x}_v\in\{x_c\}+2\delta/5\mathbb{B}\subset\{x_c\}+\delta\mathbb{B}$. If v is not reported as v_{cur}, then there must exists another vertex $y\in V$ such that y is either closer to or equidistant from x_{rand} as compared to v, i.e.,

$$|\overline{x}_y-x_{rand}|\leq|\overline{x}_v-x_{rand}|\leq\frac{3\delta}{5}.$$

Then, $|\overline{x}_y-x_c|\leq|\overline{x}_y-x_{rand}|+|x_{rand}-x_c|\leq4\delta/5$, which implies that $\overline{x}_y\in x_c+\delta\mathbb{B}$. This implies that if the sampling point x_{rand} is such that $x_{rand}\in x_c+(\delta/5)\mathbb{B}$, no matter v or y is reported as $x_{v_{cur}}$, we have $x_{v_{cur}}\in x_c+\delta\mathbb{B}$.

Note that $x_{rand}\in\{x_c\}+(\delta/5)\mathbb{B}$ implies x_{rand} is sampled from the ball centered at x_c with radius $\delta/5$. Therefore, by Definition 6.11 and (36), the probability of $x_{rand}\in\{x_c\}+(\delta/5)\mathbb{B}$ is $\zeta_n(\delta/5)^n/\mu(S)$, where ζ_n denotes the Lebesgue measure of the unit ball in $\mathbb{R}^n$.

F Proof of Lemma 6.27

F.1 Supporting Lemmas to Prove Lemma 6.27

Lemma F.1. *(Lemma 2 in Kleinbort et al. (2018)) Given a hybrid system $\mathcal{H}$ that satisfies Assumption 6.17, let $\psi=(\phi,v)$ and $\psi'=(\phi',v')$ be two purely continuous solution pairs to $\mathcal{H}$ with $(T,0)=\max\operatorname{dom}\psi$, $(T',0)=\max\operatorname{dom}\psi'$ and $T'\leq T$. The input functions v and v' are assumed to be constant, denoted $v:[0,T]\times\{0\}\to u_C\in U_C$ and $v':[0,T']\times\{0\}\to u'_C\in U_C$, respectively. Suppose initial state $\phi(0,0)\in\{\phi'(0,0)\}+\delta\mathbb{B}$ for some $\delta>0$. Then $|\phi(T')-\phi'(T')|\leq e^{K_x^fT'}\delta+K_u^fT'e^{K_x^fT'}\Delta u$, where $\Delta u=|u_C-u'_C|$, K_x^f and K_u^f are from Assumption 6.17.*

Lemma F.2. *Given a hybrid system $\mathcal{H}$ that satisfies Assumption 6.17, let $\psi=(\phi,v)$ and $\psi'=(\phi',v')$ be two purely continuous solution pairs to $\mathcal{H}$ with $(T,0)=\max\operatorname{dom}\psi$, $(T',0)=\max\operatorname{dom}\psi'$ and $T'\leq T$. The input functions v and v' are assumed to be constant, denoted $v:[0,T]\times\{0\}\to u_C\in U_C$ and $v':[0,T']\times\{0\}\to u'_C\in U_C$, respectively. Suppose initial state $\phi(0,0)\in\{\phi'(0,0)\}+\kappa_1\delta\mathbb{B}$ for some $\delta>0$ and $\kappa_1\in(0,1/2)$. Then, for each $\kappa_2\in(2\kappa_1,1)$ and each $\epsilon\in(0,\frac{\kappa_2\delta}{2})$, ϕ and ϕ' are $(\overline{T},\kappa_2\delta)$-close where $\overline{T}=\max\{T,T'\}$ if the following hold:*

1. *T and T' are such that*

$$\begin{aligned}T'\in T_k=\{t_l\in[\max\{T-\kappa_2\delta,0\},T]:\\\forall t'\in[t_l,T],\{\phi(t',0)\}+(\frac{\kappa_2\delta}{2}-\epsilon)\mathbb{B}\\\subset\{\phi(T,0)\}+\frac{\kappa_2\delta}{2}\mathbb{B}\},\end{aligned} \tag{55}$$

2. *u_C and u'_C are such that*

$$|u_C-u'_C|<\frac{\frac{\kappa_2\delta}{2}-\epsilon-\exp(K_x^fT)\kappa_1\delta}{K_u^fT\exp(K_x^fT)}. \tag{56}$$

Proof. We show that ψ and ψ' satisfy each item in Definition 6.23.

1. This item shows that ψ and ψ' satisfy the first condition in Definition 6.23, namely, for all $(t,j)\in\operatorname{dom}\phi$ with $t+j\leq\overline{T}$, there exists s such that $(s,j)\in\operatorname{dom}\phi'$, $|t-s|<\kappa_2\delta$, and $|\phi(t,j)-\phi'(s,j)|<\kappa_2\delta$.

   Because of Assumption 6.17, according to Lemma F.1, it follows

$$\begin{aligned}&|\phi(t,0)-\phi'(t,0)|\\&\leq\exp(K_x^ft)\kappa_1\delta+K_u^ft\exp(K_x^ft)|u_C-u'_C|\end{aligned} \tag{57}$$

Note that $T' \in T_k \subset [\max\{T-\kappa_2\delta, 0\}, T]$. Therefore, we have $T' \leq T$. Because $T' \leq T$, for all $(t,0) \in \operatorname{dom}\phi'$ with $t+0 \leq T'+0 \leq T+0 = T = \overline{T}$, there exists $s=t$ such that $(s,0) \in \operatorname{dom}\phi$, $|t-s| = 0 < \kappa_2\delta$. Then, by applying (56) to (57), we have

$$\begin{aligned}
&|\phi(t,0) - \phi'(s,0)| = |\phi(t,0) - \phi'(t,0)| \\
&\leq \exp(K_x^f t)\kappa_1\delta + K_u^f t \exp(K_x^f t)\Delta u \\
&\leq \exp(K_x^f T)\kappa_1\delta + K_u^f T \exp(K_x^f T)\Delta u \\
&\leq \frac{\kappa_2\delta}{2} - \epsilon < \kappa_2\delta.
\end{aligned}$$

Hence, item 1 in Definition 6.23 is established.

2. This item shows that ψ and ψ' satisfy the second condition in Definition 6.23, namely, for all $(t,j) \in \operatorname{dom}\phi'$ with $t+j \leq \overline{T} = T$, there exists s such that $(s,j) \in \operatorname{dom}\phi$, $|t-s| < \kappa_2\delta$, and $|\phi'(t,j) - \phi(s,j)| < \kappa_2\delta$.

   We consider the following two cases.

   (a) For all $(t,0) \in \operatorname{dom}\phi$ with $0 \leq t+0 \leq T'+0 = T'$, there exists $s=t$ such that $(s,0) \in \operatorname{dom}\phi'$, $|t-s| = 0 < \kappa_2\delta$ and

   $$|\phi(t,0) - \phi'(s,0)| \leq \frac{\kappa_2\delta}{2} - \epsilon < \kappa_2\delta - \epsilon < \kappa_2\delta$$

   because of (57).

   (b) This item considers the case of $(t,0) \in \operatorname{dom}\phi$ with $T' \leq t+0 \leq T+0 = T$.

   For all $(t,0) \in \operatorname{dom}\phi$ with $T' \leq t+0 \leq T+0 = T$, let $s = T'$. Because of (55), then $s = T' \in [\max\{T-\kappa_2\delta, 0\}, T]$. Since $s \in [\max\{T-\kappa_2\delta, 0\}, T]$ and $t \in [T', T]$, therefore, we have

   $$|t-s| \leq \kappa_2\delta.$$

   Also, because of the definition of T_k in (55), for all $(t,0) \in \operatorname{dom}\phi$ with $T' \leq t \leq T$, we have

   $$\begin{aligned}
   \phi(t,0) &\in \{\phi(t,0)\} + (\frac{\kappa_2\delta}{2} - \epsilon)\mathbb{B} \\
   &\subset \{\phi(T,0)\} + \frac{\kappa_2\delta}{2}\mathbb{B}.
   \end{aligned}$$

   Because of (57) and (55), we also have

   $$\begin{aligned}
   \phi'(s,0) &\in \{\phi(s,0)\} + (\frac{\kappa_2\delta}{2} - \epsilon)\mathbb{B} \\
   &\subset \{\phi(T,0)\} + \frac{\kappa_2\delta}{2}\mathbb{B}.
   \end{aligned}$$

   Therefore, $\phi(t,0)$ and $\phi'(s,0)$ are two points in a ball centered at $\phi(T,0)$ with radius $\frac{\kappa_2\delta}{2}$. Note that the maximum distance between two points within a ball is its diameter. Hence, we have

   $$|\phi(t,0) - \phi'(s,0)| < \kappa_2\delta.$$

Therefore, when both (55) and (56) hold, ϕ and ϕ' are $(\overline{T}, \kappa_2\delta)$-close. □

Lemma F.3. *Given a hybrid system $\mathcal{H}$ that satisfies Assumption 6.17, let $\psi = (\phi, \upsilon)$ and $\psi' = (\phi', \upsilon')$ be two purely continuous solution pairs to $\mathcal{H}$ with $(T,0) = \max \operatorname{dom}\psi$, $(T',0) = \max \operatorname{dom}\psi'$ and $T' \leq T$. The input functions υ and υ' are assumed to be constant, denoted $\upsilon : [0,T] \times \{0\} \to u_C \in U_C$ and $\upsilon' : [0,T'] \times \{0\} \to u_C' \in U_C$, respectively. Suppose initial state $\phi(0,0) \in \{\phi'(0,0)\} + \kappa_1\delta\mathbb{B}$ for some $\delta > 0$ and $\kappa_1 \in (0, 1/2)$. Then, for each $\kappa_2 \in (2\kappa_1, 1)$ and each $\epsilon \in (0, \frac{\kappa_2\delta}{2})$, $\phi'(T',0) \in \{\phi(T,0)\} + \kappa_2\delta\mathbb{B}$ if the following hold:*

1. *T and T' are such that*

$$\begin{aligned}
&T' \in T_k = \{t_l \in [\max\{T-\kappa_2\delta, 0\}, T] : \forall t' \in [t_l, T], \\
&\{\phi(t',0)\} + (\frac{\kappa_2\delta}{2} - \epsilon)\mathbb{B} \subset \{\phi(T,0)\} + \frac{\kappa_2\delta}{2}\mathbb{B}\},
\end{aligned} \tag{58}$$

2. *u_C and u_C' are such that*

$$|u_C - u_C'| < \frac{\frac{\kappa_2\delta}{2} - \epsilon - \exp(K_x^f T)\kappa_1\delta}{K_u^f T \exp(K_x^f T)}. \tag{59}$$

Proof. Because of Assumption 6.17, according to Lemma F.1, we have

$$\begin{aligned}
&|\phi(t,0) - \phi'(t,0)| \leq \\
&\exp(K_x^f t)\kappa_1\delta + K_u^f t \exp(K_x^f t)|u_C - u_C'|
\end{aligned} \tag{60}$$

Note that $T' \in T_k \subset [\max\{T-\kappa_2\delta, 0\}, T]$, therefore, we have $T' \leq T$. Then, by applying $T' \leq T$ and (59) to (60), we have

$$\begin{aligned}
&|\phi(T',0) - \phi'(T',0)| \\
&\leq \exp(K_x^f T')\kappa_1\delta + K_u^f T' \exp(K_x^f T')|u_C - u_C'| \\
&\leq \exp(K_x^f T)\kappa_1\delta + K_u^f T \exp(K_x^f T)|u_C - u_C'| \\
&< (\frac{\kappa_2\delta}{2} - \epsilon).
\end{aligned}$$

Therefore, we have $\phi'(T',0) \in \{\phi(T',0)\} + (\frac{\kappa_2\delta}{2} - \epsilon)\mathbb{B}$.

Since (58) holds, then we have

$$\begin{aligned}
\{\phi(T',0)\} + (\frac{\kappa_2\delta}{2} - \epsilon)\mathbb{B} &\subset \{\phi(T,0)\} + \frac{\kappa_2\delta}{2}\mathbb{B} \\
&\subset \{\phi(T,0)\} + \kappa_2\delta\mathbb{B}.
\end{aligned}$$

Therefore, we have $\phi'(T',0) \in \{\phi(T',0)\} + (\frac{\kappa_2\delta}{2} - \epsilon)\mathbb{B} \subset \{\phi(T,0)\} + \kappa_2\delta\mathbb{B}$. □

F.2 Proof of Lemma 6.27

This proof proceeds as follows. From item 1 in Definition 6.13, we denote the constant input signal that is randomly selected from $\mathcal{U}_C$ as $\tilde{\upsilon}' : [0, t_m'] \to u_C' \in U_C$, where t_m' denotes the time duration of $\tilde{\upsilon}'$. Since in the statement of Lemma 6.27, the input function υ is also constant, denote $\upsilon : [0,\tau] \times \{0\} \to u_C \in U_C$. Under Assumption 6.17, Lemma F.2 and Lemma F.3 guarantee that both E_1 and E_2 occur if t_m' and u_C' satisfy the following conditions:

1. t'_m is such that

$$t'_m \in T_k := \{t_l \in [\max\{\tau - \kappa_2\delta, 0\}, \tau] : \{\phi(t', 0)\} + r'\mathbb{B} \subset \{\phi(\tau, 0)\} + \frac{\kappa_2\delta}{2}\mathbb{B} \quad \forall t' \in [t_l, \tau]\} \tag{61}$$

where $r' = \frac{\kappa_2\delta}{2} - \epsilon$ and the elements in (61) come from (55) and (58).

2. u'_C is such that

$$0 \leq \Delta u \leq \frac{\frac{\kappa_2\delta}{2} - \epsilon - \exp(K_x^f\tau)\kappa_1\delta}{K_u^f\tau\exp(K_x^f\tau)}. \tag{62}$$

where $\Delta u := |u_C - u'_C|$ and the elements in (62) come from (56) and (59).

Then, by the uniform execution of HyRRT as defined in Definition 6.11, we proceed to characterize the probability of selecting $\tilde{v}'$, namely, selecting t'_m and u'_C satisfying (61) and (62), respectively, and provide a lower bound as in (37).

We first show that set T_k has positive Lebesgue measure, which will be used to characterize p_t in (37). Since ϕ is purely continuous, for arbitrary small $\epsilon > 0$, there exists a lower bound $t'_l \in (0, \tau)$ such that

$$|\phi(t', 0) - \phi(\tau, 0)| < \epsilon \tag{63}$$

for each $t' \in [t'_l, \tau]$. For each $t' \in [t'_l, \tau]$ and each point $x_p \in \{\phi(t', 0)\} + r'\mathbb{B}$, by the triangle inequality, it follows that:

$$\begin{aligned} |x_p - \phi(\tau, 0)| &= |x_p - \phi(t', 0) + \phi(t', 0) - \phi(\tau, 0)| \\ &\leq |x_p - \phi(t', 0)| + |\phi(t', 0) - \phi(\tau, 0)|. \end{aligned}$$

From $|x_p - \phi(t', 0)| \leq r' = \frac{\kappa_2\delta}{2} - \epsilon$ and (63), it follows that

$$\begin{aligned} |x_p - \phi(\tau, 0)| &\leq |x_p - \phi(t', 0)| + |\phi(t', 0) - \phi(\tau, 0)| \\ &\leq r' + \epsilon = \frac{\kappa_2\delta}{2}. \end{aligned} \tag{64}$$

From (64), for each point $x_p \in \{\phi(t', 0)\} + r'\mathbb{B}$, it follows $x_p \in \{\phi(\tau, 0)\} + \frac{\kappa_2\delta}{2}\mathbb{B}$. Therefore, we have

$$\{\phi(t', 0)\} + r'\mathbb{B} \subset \{\phi(\tau, 0)\} + \frac{\kappa_2\delta}{2}\mathbb{B}$$

for each $t' \in [t'_l, \tau]$. This leads to the existence of $t'_l < \tau$ such that

$$\begin{aligned} t'_l \in \overline{T}_k :=& \{t_l \in [0, \tau] : \forall t' \in [t_l, \tau], \{\phi(t', 0)\} + r'\mathbb{B} \\ &\subset \{\phi(\tau, 0)\} + \frac{\kappa_2\delta}{2}\mathbb{B}\}, \end{aligned}$$

implying that $\mu(\overline{T}_k) \geq \tau - t'_l > 0$, where $\mu(\overline{T}_k)$ denotes the Lebesgue measure of $\overline{T}_k$. Since the interval $[\max\{\tau - \kappa_2\delta, 0\}, \tau]$ has positive Lebesgue measure and the intervals $\overline{T}_k$ and $[\max\{\tau - \kappa_2\delta, 0\}, \tau]$ are both upper bounded by τ, then $T_k = \overline{T}_k \cap [\max\{\tau - \kappa_2\delta, 0\}, \tau]$ has positive Lebesgue measure. By (36), the probability of selecting the t'_m such that (61) is satisfied, denoted p_t, is computed as follows:

$$p_t = \frac{\mu(T_k)}{T_m} \in (0, 1]$$

where T_m is from Definition 6.13.

Next, we discuss the conditions on u'_C such that both E_1 and E_2 occur. In addition to (62), to ensure that no intersection between ψ_{new} and the unsafe set X_u prevents the return of ψ_{new} in the function call `new_state`, the following condition is also required:

$$\Delta u \leq \delta. \tag{65}$$

Therefore, to ensure that both E_1 and E_2 occur, u'_C need to satisfy both (62) and (65), namely,

$$\Delta u \leq \min\left\{\frac{\frac{\kappa_2\delta}{2} - \epsilon - \exp(K_x^f\tau)\kappa_1\delta}{K_u^f\tau\exp(K_x^f\tau)}, \delta\right\}. \tag{66}$$

Note that the choice of u'_C that satisfies (66) is a ball in $\mathbb{R}^m$ centered at u_C with radius $\max\left\{\min\left\{\frac{\frac{\kappa_2\delta}{2} - \epsilon - \exp(K_x^f\tau)\kappa_1\delta}{K_u^f\tau\exp(K_x^f\tau)}, \delta\right\}, 0\right\}$, where the operator $\max\{\cdot, 0\}$ prevents the negative values for the radius. Therefore, according to (36), the probability of selecting u'_C satisfying (62), denoted p_u, is

$$p_u = \frac{\zeta_n\left(\max\left\{\min\left\{\frac{\frac{\kappa_2\delta}{2} - \epsilon - \exp(K_x^f\tau)\kappa_1\delta}{K_u^f\tau\exp(K_x^f\tau)}, \delta\right\}, 0\right\}\right)^m}{\mu(U_C)}.$$

Hence, we have

$$\begin{aligned} &\text{Prob}[E_1 \& E_2] \geq p_t p_u \\ &= p_t \frac{\zeta_n\left(\max\left\{\min\left\{\frac{\frac{\kappa_2\delta}{2} - \epsilon - \exp(K_x^f\tau)\kappa_1\delta}{K_u^f\tau\exp(K_x^f\tau)}, \delta\right\}, 0\right\}\right)^m}{\mu(U_C)}. \end{aligned}$$

□

G Closeness Guarantee between the Concatenation Results of Hybrid Arcs

G.1 Supporting Lemma

Lemma G.1. *Given two compact hybrid arcs ϕ and ϕ' that are (τ, ϵ)-close, then we have $|T - T'| < \epsilon$ and $J = J'$, where $(T, J) = \max \operatorname{dom} \phi$, $(T', J') = \max \operatorname{dom} \phi'$, and $\tau = \max\{T + J, T' + J'\}$.*

Proof. We prove

$$|T - T'| < \epsilon$$

by contradiction. Suppose

$$T - T' \geq \epsilon. \tag{67}$$

Since ϕ and ϕ' are (τ, ϵ)-close, from Definition 6.23, for $(T, J) \in \operatorname{dom} \phi$ satisfying $T + J \leq \tau = \max\{T + J, T' + J'\}$, there exists s such that $(s, J) \in \operatorname{dom} \phi'$ and

$$|T - s| < \epsilon. \tag{68}$$

By $(s, J) \in \operatorname{dom} \phi'$ and (67), it follows that $s \leq T' < T$. Then, by (67), we have

$$|T - s| \geq T - T' \geq \epsilon,$$

which contradicts (68). Therefore, (67) cannot hold. A similar contradiction can be derived if we suppose $T' - T \geq \epsilon$. Therefore, we have $|T - T'| < \epsilon$.

Similarly, we prove $J = J'$ by contradiction. Suppose

$$J > J'. \tag{69}$$

Then, from Definition 6.23, for $(T, J) \in \operatorname{dom}\phi$ satisfying $T + J \leq \tau = \max\{T + J, T' + J'\}$, there exists s such that $(s, J) \in \operatorname{dom}\phi'$. However, since $(T', J') = \max \operatorname{dom}\phi'$ and (69), such s does not exist. Therefore, (69) cannot hold. A similary contradiction can also be derived if we suppose $J < J'$. Therefore, we have $J = J'$.

G.2 Closeness Guarantee

Next, we demonstrate that the operation of concatenation preserves the closeness between the hybrid arcs.

Proposition G.2. *Given compact hybrid arcs ϕ_1, ϕ_2, ϕ_1', and ϕ_2' such that ϕ_1 and ϕ_1' are (τ_1, ϵ_1)-close and ϕ_2 and ϕ_2' are (τ_2, ϵ_2)-close, where $(T_1, J_1) = \max \operatorname{dom}\phi_1$, $(T_1', J_1') = \max \operatorname{dom}\phi_1'$ and $\tau_1 = \max\{T_1 + J_1, T_1' + J_1'\}$, then $\phi_1|\phi_2$ and $\phi_1'|\phi_2'$ are $(\tau_1 + \tau_2, \epsilon_1 + \epsilon_2)$-close.*

Proof. We show that $\phi = \phi_1|\phi_2$ and $\phi' = \phi_1'|\phi_2'$ satisfy the following, as introduced in Definition 6.23:

C1. for all $(t, j) \in \operatorname{dom}\phi$ with $t + j \leq \tau_1 + \tau_2$, there exists s such that $(s, j) \in \operatorname{dom}\phi'$, $|t - s| < \epsilon_1 + \epsilon_2$, and $|\phi(t, j) - \phi'(s, j)| < \epsilon_1 + \epsilon_2$;

C2. for all $(t, j) \in \operatorname{dom}\phi'$ with $t + j \leq \tau_1 + \tau_2$, there exists s such that $(s, j) \in \operatorname{dom}\phi$, $|t - s| < \epsilon_1 + \epsilon_2$, and $|\phi'(t, j) - \phi(s, j)| < \epsilon_1 + \epsilon_2$.

First, we show that C1 holds. From Definition 5.1, it follows that $\operatorname{dom}\phi = \operatorname{dom}\phi_1 \cup (\operatorname{dom}\phi_2 + \{(T_1, J_1)\})$, where $(T_1, J_1) = \max \operatorname{dom}\phi_1$. Then, for all $(t, j) \in \operatorname{dom}\phi = \operatorname{dom}\phi_1 \cup (\operatorname{dom}\phi_2 + \{(T_1, J_1)\})$ with $t + j \leq \tau_1 + \tau_2$, we show that C1 is satisfied when 1) $(t, j) \in \operatorname{dom}\phi_1$ and when 2) $(t, j) \in \operatorname{dom}\phi_2 + \{(T_1, J_1)\}$, respectively.

1. For all $(t, j) \in \operatorname{dom}\phi$ such that $(t, j) \in \operatorname{dom}\phi_1$, namely, $t + j \leq T_1 + J_1$, given that $\tau_1 = \max\{T_1 + J_1, T_1' + J_1'\}$, it follows that

$$t + j \leq \tau_1 = \max\{T_1 + J_1, T_1' + J_1'\}.$$

Since ϕ_1 and ϕ_1' are (τ_1, ϵ_1)-close, there exists s' such that $(s', j) \in \operatorname{dom}\phi_1'$, $|t - s'| < \epsilon_1$, and $|\phi_1(t, j) - \phi_1'(s', j)| < \epsilon_1$. Therefore, there exists $s = s'$ such that $(s, j) \in \operatorname{dom}\phi_1' \subset \operatorname{dom}\phi'$, $|t - s| < \epsilon_1 < \epsilon_1 + \epsilon_2$, and $|\phi(t, j) - \phi'(s, j)| = |\phi_1(t, j) - \phi_1'(s, j)| < \epsilon_1 < \epsilon_1 + \epsilon_2$. Hence, C1 is established.

2. For all $(t, j) \in \operatorname{dom}\phi$ such that $(t, j) \in (\operatorname{dom}\phi_2 + \{(T_1, J_1)\})$, namely, $t + j \geq T_1 + J_1$, since ϕ_2 and ϕ_2' are (τ_2, ϵ_2)-close, there exists s' such that

$$(s', j - J_1) \in \operatorname{dom}\phi_2' \tag{70}$$

$$|t - T_1 - s'| < \epsilon_2, \tag{71}$$

$$|\phi_2(t - T_1, j - J_1) - \phi_2'(s', j - J_1)| < \epsilon_2. \tag{72}$$

Since ϕ_1 and ϕ_1' are (τ_1, ϵ_1)-close where $\tau_1 = \max\{T_1 + J_1, T_1' + J_1'\}$, by Lemma G.1, we have

$$|T_1 - T_1'| \leq \epsilon_1 \tag{73}$$

and

$$J_1 = J_1'. \tag{74}$$

By (70) and (74), it follows that s' is such that

$$\begin{aligned}(s' + T_1', j) &\in \operatorname{dom}\phi_2' + \{(T_1', J_1)\} \\ &= \operatorname{dom}\phi_2' + \{(T_1', J_1')\} \subset \operatorname{dom}\phi'.\end{aligned}$$

Furthermore, because of (71) and (73), by the triangle inequality, it follows that

$$|t - (s' + T_1')| \leq |t - T_1 - s'| + |T_1 - T_1'| < \epsilon_2 + \epsilon_1.$$

By (72), it follows that

$$\begin{aligned}|\phi(t, j) &- \phi'(s' + T_1', j)| \\ &= |\phi_2(t - T_1, j - J_1) - \phi_2'(s', j - J_1)| \\ &< \epsilon_2 < \epsilon_1 + \epsilon_2.\end{aligned}$$

Therefore, we can find $s = s' + T_1'$ such that $(s, j) \in \operatorname{dom}\phi'$, $|t - s'| < \epsilon_1 + \epsilon_2$, and $|\phi(t, j) - \phi'(s', j)| < \epsilon_1 + \epsilon_2$.

The proof for C2 follows a similar logic to the aforementioned arguments, achieved by swapping ϕ and ϕ'. Therefore, $\phi_1|\phi_2$ and $\phi_1'|\phi_2'$ are $(\tau_1 + \tau_2, \epsilon_1 + \epsilon_2)$-close. □

H Definition of Truncation and Translation Operation

Definition H.1. (Truncation and translation operation). *Given a function $\phi : \operatorname{dom}\phi \to \mathbb{R}^n$, where $\operatorname{dom}\phi$ is hybrid time domain, and pairs of hybrid time $(T_1, J_1) \in \operatorname{dom}\phi$ and $(T_2, J_2) \in \operatorname{dom}\phi$ such that $T_1 \leq T_2$ and $J_1 \leq J_2$, the function $\widetilde{\phi} : \operatorname{dom}\widetilde{\phi} \to \mathbb{R}^n$ is the truncation of ϕ between (T_1, J_1) and (T_2, J_2) and translation by (T_1, J_1) if*

1. *$\operatorname{dom}\widetilde{\phi} = \operatorname{dom}\phi \cap ([T_1, T_2] \times \{J_1, J_1 + 1, ..., J_2\}) \backslash \{(T_1, J_1)\}$, where the minus sign denotes Minkowski difference;*
2. *$\widetilde{\phi}(t, j) = \phi(t + T_1, j + J_1)$ for all $(t, j) \in \operatorname{dom}\widetilde{\phi}$.*

I Proof of Lemma 6.29

Under Assumption 6.20, we have

$$|\phi(0, 1) - \phi_{new}(0, 1)| \leq K_x^g |\phi(0, 0) - \phi_{new}(0, 0)| + K_u^g \Delta u$$

where $\Delta u = |\upsilon(0, 0) - \upsilon_{new}(0, 0)|$. From $\phi_{new}(0, 0) \in \{\phi(0, 0)\} + \kappa_1 \delta \mathbb{B}$, we have

$$|\phi(0, 1) - \phi_{new}(0, 1)| \leq K_x^g \kappa_1 \delta + K_u^g \Delta u \tag{75}$$

with $K_x^g, K_u^g > 0$. We denote the input value that is randomly selected from $\mathcal{U}_D$ in the function call `new_state` as $u_D \in \mathbb{R}^m$. Given that u_D is input to the discrete dynamics simulator (23) to simulate $(\phi_{new}, \upsilon_{new})$, it follows that $\upsilon_{new}(0, 0) = u_D$ and, hence, $\Delta u = |\upsilon(0, 0) - u_D|$.

Then, we show that if

$$\Delta u \leq \frac{(\kappa_2 - K_x^g \kappa_1)\delta}{K_u^g}, \tag{76}$$

then, the probabilistic event E occurs. From (76), it follows from (75) that

$$|\phi(0,1) - \phi_{new}(0,1)| \leq K_x^g \kappa_1 \delta + K_u^g \Delta u \leq \kappa_2 \delta,$$

namely, $\phi_{new}(0,1) \in \{\phi(0,1)\} + \kappa_2 \delta \mathbb{B}$, implying that E occurs.

To ensure that no intersection between ψ_{new} and the unsafe set X_u prevents the return of ψ_{new} in the function call `new_state`, the following condition on the sampling result of u_D is also required:

$$0 \leq \Delta u = |\upsilon(0,0) - u_D| \leq \delta. \tag{77}$$

Therefore, to satisfy both (76) and (77) such that E occurs, we have

$$0 \leq \Delta u \leq \min\left\{\frac{(\kappa_2 - K_x^g \kappa_1)\delta}{K_u^g}, \delta\right\}. \tag{78}$$

By (36), the probability to randomly select an input value from $\mathcal{U}_D$ that satisfies (78) is

$$p_u = \frac{\zeta_n \left(\max\left\{\min\left\{\frac{(\kappa_2 - K_x^g \kappa_1)\delta}{K_u^g}, \delta\right\}, 0\right\}\right)^m}{\mu(U_D)} \tag{79}$$

where ζ_n is the Lebesgue measure of the unit ball in $\mathbb{R}^m$ and $\mu(U_D)$ denotes the Lebesgue measure of U_D. Therefore, we have

$$\text{Prob}[E] \geq p_u = \frac{\zeta_n \left(\max\left\{\min\left\{\frac{(\kappa_2 - K_x^g \kappa_1)\delta}{K_u^g}, \delta\right\}, 0\right\}\right)^m}{\mu(U_D)}. \tag{80}$$

□

J Probabilistic Completeness Guarantee of Finding a Motion Plan with Positive Clearance

We then establish a preliminary result demonstrating that, if there exists a motion plan to $\mathcal{P}$ with positive clearance and all assumptions outlined in Section 6.2 are met, HyRRT is probabilistically complete.

Proposition J.1. *Given a motion planning problem $\mathcal{P} = (X_0, X_f, X_u, (C, f, D, g))$, suppose that Assumptions 6.15, 6.17, and 6.20 are satisfied, and there exists a compact motion plan (ϕ, υ) to $\mathcal{P}$ with clearance $\delta > 0$ and piecewise-constant input. Then, using a complete input library and when executed uniformly as defined in Definition 6.11, the probability that HyRRT fails to find a motion plan $\psi' = (\phi', \upsilon')$ such that ϕ' is $(\tilde{\tau}, \delta)$-close to ϕ after k iterations is at most ae^{-bk}, where $a, b \in \mathbb{R}_{>0}$, $\tilde{\tau} = \max\{T + J, T' + J'\}$, $(T, J) = \max \operatorname{dom} \psi$, and $(T', J') = \max \operatorname{dom} \psi'$.*

Proof. To show the claim, first, given a compact motion plan (ϕ, υ) to $\mathcal{P}$ with clearance $\delta > 0$, we construct a sequence of hybrid time instances

$$\mathbb{T} := \{(T_i, J_i) \in \operatorname{dom}(\phi, \upsilon)\}_{i=0}^m \tag{81}$$

for some $m \in \mathbb{N}\backslash\{0\}$ to be chosen later. Alongside, we construct a corresponding geometric sequence[7] of positive real numbers

$$\mathbb{D} := \{r_i \in \mathbb{R}_{>0}\}_{i=0}^m, \tag{82}$$

with positive common ratio $q \in \mathbb{R}_{>0}$, in which case,

$$r_i := q^i r_0 \quad \forall i \in \{0, 1, ..., m\}.$$

Later in this proof, for each $i \in \{0, 1, ..., m-1\}$, Lemmas 6.25, 6.27, and 6.29 are utilized to provide positive lower bounds for their respective probabilistic events for the balls $\{\phi(T_i, J_i)\} + r_i \mathbb{B}$ and $\{\phi(T_{i+1}, J_{i+1})\} + r_{i+1}\mathbb{B}$, where $(T_i, J_i), (T_{i+1}, J_{i+1}) \in \mathbb{T}$ and $r_i, r_{i+1} \in \mathbb{D}$. This requires that $\{\phi(T_i, J_i)\} + r_i \mathbb{B}$ and $\{\phi(T_{i+1}, J_{i+1})\} + r_{i+1}\mathbb{B}$ meet each condition to establish these positive lower bounds.

The construction of $\mathbb{D}$ starts with selecting a proper common ratio q and the initial term r_0 in $\mathbb{D}$. To ensure that the positive lower bounds stated in Lemmas 6.25, 6.27, and 6.29 hold, the geometric sequence $\mathbb{D}$ must meet the following conditions:

D1) For each $i \in \{0, 1, ..., m-1\}$, $\frac{5}{2} r_i < r_{i+1}$ to meet the requirements for Lemma 6.25;

D2) For each $i \in \{0, 1, ..., m-1\}$, $2r_i < \min\{\delta, r_{i+1}\}$ to meet the conditions for Lemma 6.27 in (38a) and (39) and the condition for Lemma 6.29 in (43). In addition, it is required that $r_m < \delta$ as per (38b);

D3) For each $i \in \{0, 1, ..., m-1\}$, the condition $K_x^g r_i < r_{i+1}$ is essential to comply with the prerequisites for Lemma 6.29 in (44), where K_x^g comes from Assumption 6.20 and is fixed;

D4) $\sum_{i=0}^m r_i < \delta$, thereby ensuring, as per Proposition G.2, that the concatenation of all the ϕ_{new}'s, which are returned by the function call `new_state` in Section 5.5.4 and satisfy the properties introduced in probabilistic events in Lemmas 6.27 and 6.29, remains within the clearance δ.

To satisfy D1 - D3, the common ratio q is selected as any positive number satisfying the following inequality:

$$q > \max\left\{\frac{5}{2}, 2, K_x^g\right\} = \max\left\{\frac{5}{2}, K_x^g\right\}$$

which inherently implies $q > \frac{5}{2}$. Given that $q > \frac{5}{2} > 1$, the sequence $\mathbb{D}$ is, therefore, monotonically increasing. With the selected q, the initial term in $\mathbb{D}$ is selected as

$$r_0 = \frac{\delta}{q^{m+1}},$$

ensuring that, for each $i \in \{0, 1, ..., m-1\}$, $2r_i < 2r_{m-1} = \frac{2\delta}{q^2} < \frac{8\delta}{25} < \delta$ and $r_m = \frac{\delta}{q} < \frac{2\delta}{5} < \delta$. Furthermore, the sum[8] of $\mathbb{D}$, given by

$$\sum_{i=0}^m r_i = \frac{\delta}{q^{m+1}(1-q)} - \frac{\delta}{1-q} = \frac{\delta}{q-1}\left(1 - \frac{1}{q^{m+1}}\right) < \frac{\delta}{q-1}$$

where $q > \frac{5}{2}$, demonstrates that $\sum_{i=0}^{m} r_i < \frac{\delta}{\frac{5}{2}-1} = \frac{2\delta}{3} < \delta$, thereby satisfying D4. Therefore, this selection method satisfies D1 - D4 for each $m \in \mathbb{N}\backslash\{0\}$, with the specific value of m to be selected subsequently.

The sequence $\mathbb{T}$ is constructed such that for each $i \in \{0, 1, ..., m-1\}$, (T_i, J_i) and (T_{i+1}, J_{i+1}) satisfy one of the following two conditions:

T1) $T_i = T_{i+1}$ and $J_{i+1} = J_i + 1$, indicating a jump from (T_i, J_i) to (T_{i+1}, J_{i+1});

T2) $T_{i+1} - T_i = \Delta t'$ and $J_{i+1} = J_i$ for some $\Delta t' = \frac{\Delta t}{k}$, implying a purely continuous evolution from (T_i, J_i) to (T_{i+1}, J_{i+1}), where Δt is the resolution of the piecewise-constant input as in Definition 6.9, T_m is the upper bound for the time duration of the input signals in the complete $\mathcal{U}_C$ as outlined in Definition 6.13, and $k \in \mathbb{N}\backslash\{0\}$ is chosen large enough such that $\Delta t' = \frac{\Delta t}{k}$ is small enough to satisfy $\Delta t' \leq \frac{\ln \frac{q}{2}}{K_x^f}$, thereby satisfying (41), and $\Delta t' \leq T_m$ to establish a positive lower bound in Lemma 6.27.

Since υ is piecewise-constant, as per Definition 6.9, for any $k \in \mathbb{N}\backslash\{0\}$, $\Delta t' = \frac{\Delta t}{k}$ is also a resolution of the piecewise-constant input υ. Furthermore, Definition 6.10 implies that if $(t, j) \in \operatorname{dom} \upsilon$, $(t, j+1) \notin \operatorname{dom} \upsilon$ and $t + \Delta t' \leq \max_t \operatorname{dom} \upsilon$, it follows that $(t + \Delta t', j) \in \operatorname{dom}(\phi, \upsilon)$; see (8) for the definition of $\max_t$. Therefore, it is indeed feasible to construct $\mathbb{T}$ by sweeping through the hybrid time domain of (ϕ, υ) as follows: Starting from the initial assignment $i \leftarrow 0$ and $(T_i, J_i) \leftarrow (0, 0)$,

1. if $(T_i, J_i + 1) \in \operatorname{dom}(\phi, \upsilon)$, it follows that $(T_{i+1}, J_{i+1}) \leftarrow (T_i, J_i + 1)$, which satisfies T1;
2. if $(T_i, J_i + 1) \notin \operatorname{dom}(\phi, \upsilon)$, it follows that $(T_{i+1}, J_{i+1}) \leftarrow (T_i + \Delta t', J_i)$, which satisfies T2.

This process is continued by incrementing $i \leftarrow i + 1$ until $(T_i, J_i) = \max \operatorname{dom}(\phi, \upsilon)$. This construction of $\mathbb{T}$ results in

$$m = \frac{T}{\Delta t'} + J + 1$$

where $(T, J) = \max \operatorname{dom}(\phi, \upsilon)$.

With this construction, the motion plan between (T_i, J_i) and (T_{i+1}, J_{i+1}) is either purely continuous or purely discrete, which allows the use of Lemmas 6.27 and 6.29 in each respective regime. Therefore, the construction of $\mathbb{T}$ and $\mathbb{D}$ adheres to all the conditions required to apply Lemmas 6.25, 6.27, and 6.29.

Next, we partition the compact motion plan into a finite number of segments. By demonstrating that the probability of HyRRT generating ϕ_{new} close to each segment is positive, we establish that the probability of HyRRT failing to find a motion plan converges to zero as the number of iterations approaches infinity. Using the truncation and translation operations defined in Definition H.1, for each $i \in \{0, 1, ..., m-1\}$, let (ϕ_i, υ_i) represent the truncation of (ϕ, υ) between (T_i, J_i) and (T_{i+1}, J_{i+1}) following the translation by (T_i, J_i). Given that (T_i, J_i) and (T_{i+1}, J_{i+1}) satisfy T1 or T2, it follows that each element in $\{\phi_i\}_{i=0}^{m-1}$ is either purely continuous with a constant input or purely discrete with a single jump. Then, with the search tree denoted $\mathcal{T} = (V, E)$, the proof proceeds by showing that the probability of each of the following probabilistic events is positive:

P1) The function call $\mathcal{T}.\texttt{init}(X_0)$ in Section 5.5.1 adds a vertex associated with some $x_0 \in \phi(T_0, J_0) + \frac{2}{5} r_0 \mathbb{B}$ to $\mathcal{T}$.

P2) For each $i \in \{0, 1, , m-1\}$, given the existence of a vertex $v \in V$ such that $x_v \in \phi(T_i, J_i) + \frac{2}{5} r_i \mathbb{B}$, the function call `extend` in Algorithm 3 adds a new vertex to $\mathcal{T}$ associated with x_{new}, along with a corresponding new edge to $\mathcal{T}$ associated with $\psi_{new} = (\phi_{new}, \upsilon_{new})$, such that

PE1) $x_{new} \in \{\phi(T_{i+1}, J_{i+1})\} + \frac{2}{5} r_{i+1} \mathbb{B}$;

PE2) ϕ_{new} is $(\overline{\tau}, r_{i+1})$-close to ϕ_i, where[9]

$$\overline{\tau} = \begin{cases} \max\ \{\max_t \operatorname{dom} \phi_{new}, \max_t \operatorname{dom} \phi_i\} & \text{if } \phi_i \text{ is purely continuous} \\ 1 & \text{if } \phi_i \text{ is purely discrete.} \end{cases}$$

The process of HyRRT in finding a motion plan can be viewed as a Bernoulli trial[10] (P1) following repeated Bernoulli trials (P2). Denote the probability that the probabilistic event P1 occurs as p_{init} and that P2 occurs as p_{extend}. Next, we show that p_{init} and p_{extend} are positive.

1. Given that HyRRT is executed uniformly (see Definition 6.11), by (36) and $\mu(\{\phi(T_0, J_0)\} + \frac{2}{5} r_0 \mathbb{B}) > 0$, where $\mu(\{\phi(T_0, J_0)\} + \frac{2}{5} r_0 \mathbb{B})$ represents the Lebesgue measure of the ball $\{\phi(T_0, J_0)\} + \frac{2}{5} r_0 \mathbb{B}$, it follows that $p_{init} > 0$.
2. The function calls `nearest_neighbor` and `new_state` are executed in the function call `extend`. We first show the probability that each of `nearest_neighbor` and `new_state` contributes to an output of `extend` satisfying PE1 and PE2 is positive:
   (a) Given the existence of a vertex $v \in V$ such that $x_v \in \{\phi(T_i, J_i)\} + \frac{2}{5} r_i \mathbb{B}$, Lemma 6.25 implies that the probability that the function call `nearest_neighbor` returns a vertex v_{cur} such that $\overline{x}_{v_{cur}} \in \{\phi(T_i, J_i)\} + r_i \mathbb{B}$, denoted p_{near}, is positive.
   (b) Under the condition that $\overline{x}_{v_{cur}} \in \{\phi(T_i, J_i)\} + r_i \mathbb{B}$ and $r_i < \frac{2}{5} r_{i+1}$ ensured by D1, the probability that the function call `new_state` generates a new solution pair $\psi_{new} = (\phi_{new}, \upsilon_{new})$ satisfying PE1 and PE2, denoted p_{new}, is positive. This property is guaranteed by Lemma 6.27, if ϕ_i is purely continuous and by Lemma 6.29 if ϕ_i is purely discrete.

   Therefore, since p_{near} and p_{new} are positive, it follows that $p_{extend} := p_{near} p_{new}$ is positive.

Given that the probability of P2 occurring is positive, and denoting the total number of occurrences of P2 trials as X_n, where $n \in \mathbb{N}\backslash\{0\}$, we characterize the probability of HyRRT

failing to identify a motion plan, which is equivalent to P2 occurring fewer than m times, as follows:

$$\lim_{n\to\infty} \text{Prob}[X_n < m] = \lim_{n\to\infty} \sum_{i=0}^{m-1} \binom{n}{i} p_{extend}^i (1-p_{extend})^{n-i} \tag{83}$$

Since $i \in \{0, 1, ..., m-1\}$, we have $0 \le i \le m-1$. As $n \to \infty$, there exists an $N \in \mathbb{N}$ such that for all $n \ge N$, we have $0 \le i \le m-1 \le \frac{n}{2}$. Such an N can be chosen as $2(m-1)$. Due to the increasing monotonicity[11] of $\binom{n}{k}$ for k from 0 to $\frac{n}{2}$ and $0 \le i \le m-1 \le \frac{n}{2}$, we have $\binom{n}{i} \le \binom{n}{m-1}$. Hence,

$$\begin{aligned}\lim_{n\to\infty} \text{Prob}[X_n < m] &\le \lim_{n\to\infty} \sum_{i=0}^{m-1} \binom{n}{m-1} p_{extend}^i (1-p_{extend})^{n-i} \\ &\le \lim_{n\to\infty} \binom{n}{m-1} \sum_{i=0}^{m-1} p_{extend}^i (1-p_{extend})^{n-i}.\end{aligned} \tag{84}$$

Define $p := \max\{p_{extend}, 1-p_{extend}\} \in (0,1)$. It follows that

$$\begin{aligned}\lim_{n\to\infty} \text{Prob}[X_n < m] &\le \lim_{n\to\infty} \binom{n}{m-1} \sum_{i=0}^{m-1} p^i p^{n-i} \\ &\le \lim_{n\to\infty} \binom{n}{m-1} \sum_{i=0}^{m-1} p^n.\end{aligned} \tag{85}$$

Since p^n is not related to i, we have $\sum_{i=0}^{m-1} p^n = mp^n$. It follows that

$$\begin{aligned}\lim_{n\to\infty} \text{Prob}[X_n < m] &\le \lim_{n\to\infty} \binom{n}{m-1} mp^n \\ &\le \lim_{n\to\infty} \frac{n!}{(m-1)!(n-m+1)!} mp^n\end{aligned} \tag{86}$$

Since $\frac{n!}{(n-m+1)!} = \Pi_{i=n-m+2}^{n} i \le \Pi_{i=n-m+2}^{n} n = n^{m-1}$, it follows that

$$\lim_{n\to\infty} \text{Prob}[X_n < m] \le \lim_{n\to\infty} \frac{n^{m-1}}{(m-1)!} mp^n \tag{87}$$

Since $p \in (0,1)$ and m is finite, then $\lim_{n\to\infty} \frac{n^{m-1}}{(m-1)!} mp^n = 0$. Therefore, the probability that HyRRT fails to find a motion plan is converging to zero as the number of iterations approaches infinity.

In addition, Proposition G.2 establishes that the concatenation of all the ϕ_{new}'s satisfying PE2, which is returned by HyRRT as ϕ', is $(\tilde{\tau}, \sum_{i=0}^{m} r_i)$-close to $\phi = \phi_0|\phi_1|...|\phi_m$, where $\sum_{i=0}^{m} r_i < \delta$ is guaranteed by D4.

Notes

6. A property is said to hold almost everywhere or for almost all points in a set S if the set of elements of S at which the property does not hold has zero Lebesgue measure.
7. A geometric sequence is a sequence of non-zero numbers where each term after the first is found by multiplying the previous one by a fixed, non-zero number called the common ratio.
8. The closed-form of the sum of a geometric sequence is $\sum_{i=0}^{m} r_i = \frac{r_0(1-q^{m+1})}{1-q}$
9. Given that ϕ_i is purely continuous, by T2 and Definition H.1, it follows that $\max_t \text{dom}\, \phi_i = \Delta t'$
10. In the theory of probability and statistics, a Bernoulli trial is a random experiment with exactly two possible outcomes, success and failure.
11. The increasing monotonicity of $\binom{n}{i}$ for i over the interval $(0, \frac{n}{2})$ is exploited here. We can prove it by arbitrarily selecting $k_1, k_2 \in (0, \frac{n}{2}) \cap \mathbb{N}$ such that $k_1 > k_2$ and then showing that

$$\begin{aligned}\frac{\binom{n}{k_1}}{\binom{n}{k_2}} &= \frac{\frac{n!}{k_1(n-k_1)!}}{\frac{n!}{k_2(n-k_2)!}} = \frac{\frac{(n-k_2)!}{(n-k_1)!}}{\frac{k_1!}{k_2!}} \\ &= \frac{(n-k_2)(n-k_2+1)\cdots(n-k_1+1)}{k_1(k_1-1)\cdots(k_2+1)} \\ &= \frac{n-k_2}{k_1} \times \frac{n-k_2-1}{k_1-1} \times \cdots \times \frac{n-k_1+1}{k_2+1} > 1,\end{aligned} \tag{88}$$

implying $\binom{n}{k_1} > \binom{n}{k_2}$