HyRRT-Connect: A Bidirectional Rapidly-Exploring Random Trees Motion Planning Algorithm for Hybrid Systems*

Nan Wang* **Ricardo G. Sanfelice***

* University of California, Santa Cruz, Santa Cruz, CA 95064 USA
(e-mail: nanwang, ricardo@ucsc.edu).

Abstract: This paper proposes a bidirectional rapidly-exploring random trees (RRT) algorithm to solve the motion planning problem for hybrid systems. The proposed algorithm, called HyRRT-Connect, propagates in both forward and backward directions in hybrid time until an overlap between the forward and backward propagation results is detected. Then, HyRRT-Connect constructs a motion plan through the reversal and concatenation of functions defined on hybrid time domains, ensuring the motion plan thoroughly satisfies the given hybrid dynamics. To address the potential discontinuity along the flow caused by tolerating some distance between the forward and backward partial motion plans, we reconstruct the backward partial motion plan by a forward-in-hybrid-time simulation from the final state of the forward partial motion plan. By applying the reversed input of the backward partial motion plan, the reconstruction process effectively eliminates the discontinuity and ensures that as the tolerance distance decreases to zero, the distance between the endpoint of the reconstructed motion plan and the final state set approaches zero. The proposed algorithm is applied to an actuated bouncing ball example and a walking robot example so as to highlight its generality and computational improvement.

Keywords: Hybrid systems, Motion planning, RRT, Robotics

1. INTRODUCTION

Motion planning consists of finding a state trajectory and corresponding inputs that connect initial and final state sets, satisfying the system dynamics and specific safety requirements. Motion planning for purely continuous-time systems and purely discrete-time systems has been extensively explored in existing literature; see e.g., LaValle (2006). In recent years, several motion planning algorithms have been developed, including graph search algorithms Wilfong (1988), artificial potential/fluid-flow field method Khatib (1986); Wang et al. (2017); Song et al. (2019) and sampling-based algorithms. The sampling-based algorithms have drawn much attention because of their fast exploration speed for high-dimensional problems and theoretical guarantees; specially, probabilistic completeness, which means that the probability of failing to find a motion plan converges to zero, as the number of samples approaches infinity. Compared with other sampling-based algorithms, such as probabilistic roadmap algorithm, the rapidly-exploring random tree (RRT) algorithm LaValle and Kuffner Jr (2001) is perhaps the most successful algorithm to solve motion planning problems because it does not require a steering function to solve a two point boundary value problem, which is difficult to

solve for most dynamical systems. While the aforementioned motion planning algorithms have been extensively applied to purely continuous-time and purely discrete-time systems, comparatively less effort has been devoted into motion planning for systems with combined continuous and discrete behavior. In our earlier research Wang and Sanfelice (2022), we formulated a motion planning problem for hybrid systems using hybrid equations, as in Sanfelice (2021). This formulation presents a general framework that encompasses a wide range of hybrid systems. In the same paper, we introduced a probabilistically complete RRT algorithm, referred to as HyRRT, specifically designed to address the motion planning problem for hybrid systems. Building on this work, we formulated an optimal motion planning problem for hybrid systems in the same hybrid model framework in Wang and Sanfelice (2023). In this research, we introduce HySST, an asymptotically near-optimal motion planning algorithm for hybrid systems from the Stable Sparse RRT (SST) algorithm, originally introduced in Li et al. (2016).

It is significantly challenging for almost all motion planning algorithms to maintain efficient computation performance, especially in solving high-dimensional problems. Although RRT-type algorithms have demonstrated notable efficiency in rapidly searching for solutions to high-dimensional problems compared to other algorithm types, there remains room for enhancing their computational performance. To improve the computational performance, a modular motion planning system for purely continuous-time systems, named FaSTrack, is designed in Herbert et al. (2017) that simultaneously plans and

tracks a trajectory. This system accelerates the planning process by only considering low-dimensional model of the system dynamics. In Kuffner and LaValle (2000), RRT-Connect algorithm is proposed that propagates both in forward direction and backward direction, where a notable improvement in computational performance is observed. Inspired by this work, we design a bidirectional RRT-type algorithm for hybrid dynamical systems, called HyRRT-Connect, that incrementally constructs two search trees, in which one tree is rooted in the initial state set and constructed forward in hybrid time, while the other is rooted in the final state set and constructed backward in hybrid time. However, the backward propagation is a nontrivial task for hybrid systems. To facilitate the backward propagation, we formally define a backward-in-time hybrid system that approaches the inverse dynamics of the given hybrid system. When HyRRT-Connect detects an overlap between a path in the forward search tree and a path in the backward search tree, it initiates the construction of a motion plan. This construction involves initially reversing the trajectory associated with the path in the backward search tree, followed by concatenating this reversed trajectory with the trajectory associated with the path in the forward search tree. In this paper, we formally define the reversal and concatenation operations and thoroughly validate that the results of both operations satisfy the given hybrid dynamics.

In practice, HyRRT-Connect always tolerates some distance between states in the forward and backward search trees, due to the randomness in state and input selection. However, this tolerance can result in discontinuities along the flow of the constructed motion plan. To address this issue, the trajectory associated with the backward path is reconstructed, involving simulating from the final state of the forward path while applying the reversed input from the backward path. By ensuring the same hybrid time domain as the reversal of the backward path, we guarantee that as tolerance decreases, the reconstructed motion plan's endpoint converges to the final state set. To the best of the authors' knowledge, this is the first bidirectional RRT-type algorithm being applied to systems with hybrid dynamics. The proposed algorithm is illustrated in an actuated bouncing ball system and a walking robot system. In both cases, a significant improvement in computational performance is observed, highlighting the efficiency of this novel approach.

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 presents the HyRRT-Connect algorithm. Section 5 presents the overlap detection, reconstruction process and its theoretical guarantee. Section 6 presents the illustration of HyRRT-Connect in the examples. Section 7 discusses the parallel implementation of HyRRT-Connect Algorithm. Proofs and more details are given in Wang and Sanfelice (2024).

2. NOTATION AND PRELIMINARIES

2.1 Notation

The real numbers are denoted as $\mathbb{R}$, its nonnegative subset is denoted as $\mathbb{R}_{\geq 0}$ and its nonpositive subset is denoted as $\mathbb{R}_{\leq 0}$. 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. The notation $\operatorname{rge} f$ denotes the range of the function f. 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.

2.2 Preliminaries

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

$$\mathcal{H} : \begin{cases} \dot{x} = f(x, u) & (x, u) \in C \\ x^+ = g(x, u) & (x, u) \in D \end{cases} \quad (1)$$

where $x \in \mathbb{R}^n$ represents the state, $u \in \mathbb{R}^m$ represents 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 $U_C := \{u \in \mathbb{R}^m : \exists x \in \mathbb{R}^n \text{ such that } (x, u) \in C\}$ includes all possible input values that can be applied during flows. Similarly, given a jump set D, the set $U_D := \{u \in \mathbb{R}^m : \exists x \in \mathbb{R}^n \text{ such that } (x, u) \in D\}$ 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 $\Pi_\star(K) := \{x : \exists u \in U_\star \text{ s.t. } (x, u) \in K\}$ as the projection of K onto $\mathbb{R}^n$, and define

$$C' := \Pi_C(C), D' := \Pi_D(D). \quad (2)$$

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 (1), leading to hybrid time (t, j) for the parameterization of its solutions and inputs. With the hybrid time domain, the hybrid input and the hybrid arc defined in Wang and Sanfelice (2024), the definition of solution pair to a hybrid system is given as follows.

Definition 1. (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 $\mathcal{H}$ 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 $(t, j + 1) \in \operatorname{dom} (\phi, \upsilon), (\phi(t, j), \upsilon(t, j)) \in D, \phi(t, j+1) = g(\phi(t, j), \upsilon(t, j))$.

The concatenation operation of solution pairs in (Wang and Sanfelice 2022, Definition 2.2) is given next.

Definition 2. (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 $\phi : \mathrm{dom}\ \phi \to \mathbb{R}^n$ is the concatenation of ϕ_2 to ϕ_1, denoted $\phi = \phi_1|\phi_2$, namely,

1) $\mathrm{dom}\ \phi = \mathrm{dom}\ \phi_1 \cup (\mathrm{dom}\ \phi_2 + \{(T, J)\})$, where $(T, J) = \max \mathrm{dom}\ \phi_1$ and the plus sign denotes Minkowski addition;
2) $\phi(t, j) = \phi_1(t, j)$ for all $(t, j) \in \mathrm{dom}\ \phi_1 \backslash \{(T, J)\}$ and $\phi(t, j) = \phi_2(t - T, j - J)$ for all $(t, j) \in \mathrm{dom}\ \phi_2 + \{(T, J)\}$.

3. PROBLEM STATEMENT AND APPLICATIONS

The motion planning problem for hybrid systems studied in this paper is given in (Wang and Sanfelice, 2022, Problem 1) as follows.

Problem 1. (Motion planning problem for hybrid systems). Given a hybrid system $\mathcal{H}$ as in (1) 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 $X_u \subset \mathbb{R}^n \times \mathbb{R}^m$, find $(\phi, \upsilon) : \mathrm{dom}\ (\phi, \upsilon) \to \mathbb{R}^n \times \mathbb{R}^m$, namely, *a motion plan*, such that for some $(T, J) \in \mathrm{dom}\ (\phi, \upsilon)$, 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 1;
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 $(t, j) \in \mathrm{dom}\ (\phi, \upsilon)$ 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 (1) with data (C, f, D, g), a motion planning problem $\mathcal{P}$ is formulated as $\mathcal{P} = (X_0, X_f, X_u, (C, f, D, g))$.

Problem 1 is illustrated in the following examples.

Example 1. (Actuated bouncing ball system). Consider a ball bouncing on a fixed horizontal surface. 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 $\dot{x} = \begin{bmatrix} x_2 \\ -\gamma \end{bmatrix} =: f(x, u) \quad (x, u) \in C$ 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 $C := \{(x, u) \in \mathbb{R}^2 \times \mathbb{R} : x_1 \geq 0\}$. 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 $x^+ = \begin{bmatrix} x_1 \\ -\lambda x_2 + u \end{bmatrix} =: g(x, u) \quad (x, u) \in D$ 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 with nonpositive velocity. Hence, the jump set is $D := \{(x, u) \in \mathbb{R}^2 \times \mathbb{R} : x_1 = 0, x_2 \leq 0, u \geq 0\}$.

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 (1) 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 $X_0 = \{(14, 0)\}$, 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 2. (Walking robot). For the details of the motion planning problem for the walking robot system, see Wang and Sanfelice (2024). We also solve this motion planning problem later in this paper.

4. ALGORITHM DESCRIPTION

4.1 Overview

In this section, a bidirectional RRT-type motion planning algorithm for hybrid systems, called HyRRT-Connect, is proposed. HyRRT-Connect searches for a motion plan by incrementally constructing two search trees: one starts from the initial state set and propagates forward in hybrid time, while the other starts from the final state set and propagates backward in hybrid time. Upon detecting overlaps between the two search trees, a connection is established, subsequently yielding a motion plan, which is elaborated in Section 5. Each 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. 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 $i = \{1, 2, ..., k - 1\}$.

The search tree constructed forward in hybrid time is denoted as $\mathcal{T}^{\mathrm{fw}} = (V^{\mathrm{fw}}, E^{\mathrm{fw}})$ and the search tree constructed backward in hybrid time is denoted as $\mathcal{T}^{\mathrm{bw}} = (V^{\mathrm{bw}}, E^{\mathrm{bw}})$. For the consistency of the notation, we denote $\mathcal{H}$ in (1) as $\mathcal{H}^{\mathrm{fw}} = (C^{\mathrm{fw}}, f^{\mathrm{fw}}, D^{\mathrm{fw}}, g^{\mathrm{fw}})$. Each vertex v in V^{fw} (respectively, V^{bw}) is associated with a state of $\mathcal{H}^{\mathrm{fw}}$ (respectively, the hybrid system that represents inverse dynamics of $\mathcal{H}^{\mathrm{fw}}$, denoted $\mathcal{H}^{\mathrm{bw}}$), denoted $\overline{x}_v$. Each edge e in E^{fw} (respectively, E^{bw}) is associated with a solution pair to $\mathcal{H}^{\mathrm{fw}}$ (respectively, $\mathcal{H}^{\mathrm{bw}}$), denoted $\overline{\psi}_e$, that connects the states associated with their endpoint vertices. 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, $\tilde{\psi}_p := \overline{\psi}_{(v_1, v_2)}|\overline{\psi}_{(v_2, v_3)}| \cdots |\overline{\psi}_{(v_{k-1}, v_k)}$ where $\tilde{\psi}_p$ denotes the solution pair associated with p. For the concatenation notion, see Definition 2.

The proposed HyRRT-Connect algorithm requires a library of possible inputs to construct $\mathcal{T}^{\mathrm{fw}}$, denoted $\mathcal{U}^{\mathrm{fw}} = (\mathcal{U}_C^{\mathrm{fw}}, \mathcal{U}_D^{\mathrm{fw}})$, and to construct $\mathcal{T}^{\mathrm{bw}}$, denoted $\mathcal{U}^{\mathrm{bw}} = (\mathcal{U}_C^{\mathrm{bw}}, \mathcal{U}_D^{\mathrm{bw}})$. The input library $\mathcal{U}^{\mathrm{fw}}$ (respectively, $\mathcal{U}^{\mathrm{bw}}$) includes the input signals for the flows of $\mathcal{H}^{\mathrm{fw}}$ (respectively, $\mathcal{H}^{\mathrm{bw}}$), collected in $\mathcal{U}_C^{\mathrm{fw}}$ (respectively, $\mathcal{U}_C^{\mathrm{bw}}$), and the input values for the jumps of $\mathcal{H}^{\mathrm{fw}}$ (respectively, $\mathcal{H}^{\mathrm{bw}}$), collected in $\mathcal{U}_D^{\mathrm{fw}}$ (respectively, $\mathcal{U}_D^{\mathrm{bw}}$).

HyRRT-Connect addresses the motion planning problem $\mathcal{P} = (X_0, X_f, X_u, (C^{\mathrm{fw}}, f^{\mathrm{fw}}, D^{\mathrm{fw}}, g^{\mathrm{fw}}))$ using input libraries $\mathcal{U}^{\mathrm{fw}}$ and $\mathcal{U}^{\mathrm{bw}}$ through the following steps:

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

Step 2: Incrementally construct $\mathcal{T}^{\mathrm{fw}}$ forward in hybrid time and $\mathcal{T}^{\mathrm{bw}}$ backward in hybrid time, executing both procedures in an interleaved manner [1].

Step 3: If an appropriate overlap between $\mathcal{T}^{\mathrm{fw}}$ and $\mathcal{T}^{\mathrm{bw}}$ is found, reverse the solution pair in $\mathcal{T}^{\mathrm{bw}}$, concatenate it to the solution pair in $\mathcal{T}^{\mathrm{fw}}$ and return the concatenation result.

4.2 Backward-in-time Hybrid System

In the HyRRT-Connect algorithm, a hybrid system that represents backward-in-time dynamics of $\mathcal{H}^{\mathrm{fw}} = (C^{\mathrm{fw}}, f^{\mathrm{fw}}, D^{\mathrm{fw}}, g^{\mathrm{fw}})$, denoted $\mathcal{H}^{\mathrm{bw}} = (C^{\mathrm{bw}}, f^{\mathrm{bw}}, D^{\mathrm{bw}}, g^{\mathrm{bw}})$, is required when propagating trajectories from X_f. The construction of $\mathcal{H}^{\mathrm{bw}}$ is as follows.

Definition 3. (Backward-in-time hybrid system). Given a hybrid system $\mathcal{H}^{\mathrm{fw}} = (C^{\mathrm{fw}}, f^{\mathrm{fw}}, D^{\mathrm{fw}}, g^{\mathrm{fw}})$, the backward-in-time hybrid system of $\mathcal{H}^{\mathrm{fw}}$, denoted $\mathcal{H}^{\mathrm{bw}}$, is the hybrid system

$$\mathcal{H}^{\mathrm{bw}} : \begin{cases} \dot{x} = f^{\mathrm{bw}}(x, u) & (x, u) \in C^{\mathrm{bw}} \\ x^+ \in g^{\mathrm{bw}}(x, u) & (x, u) \in D^{\mathrm{bw}} \end{cases} \quad (3)$$

where

1) The backward-in-time flow set is constructed as $C^{\mathrm{bw}} := C^{\mathrm{fw}}$.
2) The backward-in-time flow map is constructed as $f^{\mathrm{bw}}(x, u) := -f^{\mathrm{fw}}(x, u)$ for all $(x, u) \in C^{\mathrm{bw}}$.
3) The backward-in-time jump map is constructed as $g^{\mathrm{bw}}(x, u) := \{z \in \mathbb{R}^n : x = g^{\mathrm{fw}}(z, u), (z, u) \in D^{\mathrm{fw}}\}$ for all $(x, u) \in \mathbb{R}^n \times \mathbb{R}^m$.
4) The backward-in-time jump set is constructed as $D^{\mathrm{bw}} := \{(x, u) \in \mathbb{R}^n \times \mathbb{R}^m : \exists z \in \mathbb{R}^n : x = g^{\mathrm{fw}}(z, u), (z, u) \in D^{\mathrm{fw}}\}$.

While the jump map g^{fw} of the forward-in-time system $\mathcal{H}^{\mathrm{fw}}$ is single-valued, the corresponding map g^{bw} in $\mathcal{H}^{\mathrm{bw}}$ may not be, especially if g^{bw} is not invertible. Therefore, a difference inclusion is used in (3). For an example illustrating Definition 3 in Example 1, see Wang and Sanfelice (2024).

4.3 Construction of Motion Plans

To construct a motion plan, HyRRT-Connect reverses a solution pair associated with a path detected in $\mathcal{T}^{\mathrm{bw}}$ and concatenates it with a solution pair associated with a path detected in $\mathcal{T}^{\mathrm{fw}}$. The concatenation operation is defined in Definition 2 and the reversal operation is introduced below. Next, Proposition 2 shows that the concatenation result is a solution pair to $\mathcal{H}^{\mathrm{fw}}$ under mild conditions.

Proposition 2. Given two solution pairs $\psi_1 = (\phi_1, u_1)$ and $\psi_2 = (\phi_2, u_2)$ to a hybrid system $\mathcal{H}^{\mathrm{fw}}$, their concatenation $\psi = (\phi, u) = (\phi_1|\phi_2, u_1|u_2)$, denoted $\psi = \psi_1|\psi_2$, is a solution pair to $\mathcal{H}^{\mathrm{fw}}$ if the following hold:

1) $\psi_1 = (\phi_1, u_1)$ is compact;
2) $\phi_1(T, J) = \phi_2(0, 0)$, where $(T, J) = \max \mathrm{dom}\, \psi_1$;

---

[1] The parallel implementation is discussed in Section 7.

3) If both $I_{\psi_1}^J$ and $I_{\psi_2}^0$ have nonempty interior, where $I_\psi^j = \{t : (t, j) \in \mathrm{dom}\, \psi\}$ and $(T, J) = \max \mathrm{dom}\, \psi_1$, then $\psi_2(0, 0) \in C$.

Remark 3. Item 1 in Proposition 2 guarantees that ψ_2 can be concatenated to ψ_1. The concatenation operation defined in Definition 2 suggests that if ψ_2 can be concatenated to ψ_1, ψ_1 is required to be compact. Item 2 in Proposition 2 guarantees that the concatenation ψ satisfies the requirement of being absolutely continuous in the definition of hybrid arc; see Wang and Sanfelice (2024), at hybrid time (T, J), where $(T, J) = \max \mathrm{dom}\, \psi_1$. Item 3 in Proposition 2 guarantees that the concatenation result ψ satisfies item 2 in Definition 1 at hybrid time (T, J), where $(T, J) = \max \mathrm{dom}\, \psi_1$. Note that item 2 therein does not require that $\psi_1(T, J) \in C$ and $\psi_2(0, 0) \in C$ since $T \notin \mathrm{int}\, I_{\psi_1}^J$ and $0 \notin \mathrm{int}\, I_{\psi_2}^0$. However, T may belong to the interior of I_ψ^J after concatenation. Hence, item 3 guarantees that if T belongs to the interior of I_ψ^J after concatenation, ψ still satisfies item 2 in Definition 1. ▲

The reversal operation to reverse the solution pair to $\mathcal{H}^{\mathrm{bw}}$, for its concatenation to a solution to $\mathcal{H}^{\mathrm{fw}}$, is defined next.

Definition 4. (Reversal of a solution pair). Given a compact solution pair (ϕ, u) to $\mathcal{H}^{\mathrm{fw}} = (C^{\mathrm{fw}}, f^{\mathrm{fw}}, D^{\mathrm{fw}}, g^{\mathrm{fw}})$, where $\phi : \mathrm{dom}\, \phi \to \mathbb{R}^n$, $u : \mathrm{dom}\, u \to \mathbb{R}^m$, and $(T, J) = \max \mathrm{dom}\, (\phi, u)$, the pair (ϕ', u') is the reversal of (ϕ, u), where $\phi' : \mathrm{dom}\, \phi' \to \mathbb{R}^n$ with $\mathrm{dom}\, \phi' \subset \mathbb{R}_{\geq 0} \times \mathbb{N}$ and $u' : \mathrm{dom}\, u' \to \mathbb{R}^m$ with $\mathrm{dom}\, u' = \mathrm{dom}\, \phi'$, if the following hold:

1) The function ϕ' is defined as
   a) $\mathrm{dom}\, \phi' = \{(T, J)\} - \mathrm{dom}\, \phi$, where the minus sign denotes Minkowski difference;
   b) $\phi'(t, j) = \phi(T - t, J - j)$ for all $(t, j) \in \mathrm{dom}\, \phi'$.
2) The function u' is defined as
   a) $\mathrm{dom}\, u' = \{(T, J)\} - \mathrm{dom}\, u$, where the minus sign denotes Minkowski difference;
   b) For all $j \in \mathbb{N}$ such that $I^j = \{t : (t, j) \in \mathrm{dom}\, u'\}$ has nonempty interior,
      i) For all $t \in \mathrm{int}\, I^j$, $u'(t, j) = u(T - t, J - j)$;
      ii) If I^0 has nonempty interior, then $u'(0, 0) \in \mathbb{R}^m$ is such that $(\phi'(0, 0), u'(0, 0)) \in \overline{C^{\mathrm{fw}}}$;
      iii) For all $t \in \partial I^j$ such that $(t, j + 1) \notin \mathrm{dom}\, u'$ and $(t, j) \neq (0, 0)$, $u'(t, j) \in \mathbb{R}^m$.
   c) For all $(t, j) \in \mathrm{dom}\, u'$ such that $(t, j + 1) \in \mathrm{dom}\, u'$, $u'(t, j) = u(T - t, J - j - 1)$.

Proposition 4 shows that the reversal of the solution pair to a hybrid system is a solution pair to its backward-in-time hybrid system.

Proposition 4. Given a hybrid system $\mathcal{H}^{\mathrm{fw}}$ and its backward-in-time system $\mathcal{H}^{\mathrm{bw}}$, if $\psi = (\phi, u)$ is a compact solution pair to $\mathcal{H}^{\mathrm{fw}}$, the reversal $\psi' = (\phi', u')$ of $\psi = (\phi, u)$ is a compact solution pair to $\mathcal{H}^{\mathrm{bw}}$.

Proposition 2 and Proposition 4 validate the results of the concatenation and reversal operations, respectively. The following assumption integrates the conditions in Proposition 2 and Proposition 4 and is imposed on the solution pairs that are used to construct motion plans.

Assumption 5. Given a solution pair $\psi_1 = (\phi_1, u_1)$ to a hybrid system $\mathcal{H}^{\mathrm{fw}} = (C^{\mathrm{fw}}, f^{\mathrm{fw}}, D^{\mathrm{fw}}, g^{\mathrm{fw}})$ and a solution

pair $\psi_2 = (\phi_2, u_2)$ to the backward-in-time hybrid system $\mathcal{H}^{\mathrm{bw}}$ associated to $\mathcal{H}^{\mathrm{fw}}$, the following hold:

1) ψ_1 and ψ_2 are compact;
2) $\phi_1(T_1, J_1) = \phi_2(T_2, J_2)$, where $(T_1, J_1) = \max \operatorname{dom} \psi_1$ and $(T_2, J_2) = \max \operatorname{dom} \psi_2$;
3) If both $I_{\psi_1}^{J_1}$ and $I_{\psi_2}^{J_2}$ have nonempty interior, where $I_{\psi}^{j} = \{t : (t, j) \in \operatorname{dom} \psi\}$, $(T_1, J_1) = \max \operatorname{dom} \psi_1$, and $(T_2, J_2) = \max \operatorname{dom} \psi_2$, then $\psi_2(T_2, J_2) \in C$.

Remark 6. Given a hybrid system $\mathcal{H}^{\mathrm{fw}}$, its backward-in-time system $\mathcal{H}^{\mathrm{bw}}$, a solution pair ψ_1 to $\mathcal{H}^{\mathrm{fw}}$, and a solution pair ψ_2 to $\mathcal{H}^{\mathrm{bw}}$, Assumption 5 is imposed on ψ_1 and ψ_2 to guarantee that the concatenation of the reversal of ψ_2 to ψ_1 is a solution pair to $\mathcal{H}^{\mathrm{fw}}$. Assumption 5 guarantees that the conditions needed to apply Proposition 4 and Proposition 2 hold. Note that conditions that guarantee the existence of nontrivial solutions have been proposed in (Chai and Sanfelice, 2018, Proposition 3.4). If $\xi \in X_0$ is such that $\xi \in D'$, where D' is defined in (2), or there exist $\epsilon > 0$, an absolutely continuous function $z : [0, \epsilon] \to \mathbb{R}^n$ with $z(0) = \xi$, and a Lebesgue measurable and locally essentially bounded function $\tilde{u} : [0, \epsilon] \to U_C$ such that $(z(t), \tilde{u}(t)) \in C^{\mathrm{fw}}$ for all $t \in (0, \epsilon)$ and $\frac{\mathrm{d}}{\mathrm{d}t} z(t) = f^{\mathrm{fw}}(z(t), \tilde{u}(t))$ for almost all $t \in [0, \epsilon]$, where [2] $\tilde{u}(t) \in \Psi_c^u(z(t))$ for every $t \in [0, \epsilon]$, then the existence of nontrivial solution pairs is guaranteed from ξ. Items 2 and 3 in Assumption 5 relate the final states and their "last" interval of flow of the given solution pairs. ▲

The following result validates that the result constructed by the solution pairs satisfying Assumption 5 is a solution pair to $\mathcal{H}^{\mathrm{fw}}$.

Lemma 7. Given a hybrid system $\mathcal{H}^{\mathrm{fw}}$ and its backward-in-time hybrid system $\mathcal{H}^{\mathrm{bw}}$, if ψ_1 is a solution pair to $\mathcal{H}^{\mathrm{fw}}$ and ψ_2 is a solution pair to $\mathcal{H}^{\mathrm{bw}}$ such that ψ_1 and ψ_2 satisfy Assumption 5, then the concatenation $\psi = \psi_1 | \psi_2'$ is a solution pair to $\mathcal{H}^{\mathrm{fw}}$, where ψ_2' is the reversal of ψ_2.

Lemma 7 is exploited by our forthcoming HyRRT-Connect algorithm when detecting overlaps between $\mathcal{T}^{\mathrm{fw}}$ and $\mathcal{T}^{\mathrm{bw}}$.

4.4 HyRRT-Connect Algorithm

The proposed algorithm is given in Algorithm 1. The inputs of Algorithm 1 are the problem $\mathcal{P} = (X_0, X_f, X_u, (C^{\mathrm{fw}}, f^{\mathrm{fw}}, D^{\mathrm{fw}}, g^{\mathrm{fw}}))$, the backward-in-time hybrid system $\mathcal{H}^{\mathrm{bw}}$ obtained from (3), the input libraries $\mathcal{U}^{\mathrm{fw}}$ and $\mathcal{U}^{\mathrm{bw}}$, two parameters $p_n^{\mathrm{fw}} \in (0, 1)$ and $p_n^{\mathrm{bw}} \in (0, 1)$, which tune the probability of proceeding with the flow regime or the jump regime during the forward and backward construction, respectively, an upper bound $K \in \mathbb{N}_{>0}$ for the number of iterations to execute, and four tunable sets $X_c^{\mathrm{fw}} \supset \overline{C^{\mathrm{fw}'}}$, $X_d^{\mathrm{fw}} \supset \overline{D^{\mathrm{fw}'}}$, $X_c^{\mathrm{bw}} \supset \overline{C^{\mathrm{bw}'}}$ and $X_d^{\mathrm{bw}} \supset \overline{D^{\mathrm{bw}'}}$ where $C^{\mathrm{fw}'}$, $C^{\mathrm{bw}'}$, $D^{\mathrm{fw}'}$ and $D^{\mathrm{bw}'}$ are defined as in (2), which act as constraints in finding a closest vertex to x_{rand}. **Step** 1 in Section 4.1 corresponds to the function calls $\mathcal{T}^{\mathrm{fw}}.\mathtt{init}$ and $\mathcal{T}^{\mathrm{bw}}.\mathtt{init}$ in line 1 of Algorithm 1. The construction of $\mathcal{T}^{\mathrm{fw}}$ in **Step** 2 is implemented in lines 3 - 10. The construction of $\mathcal{T}^{\mathrm{bw}}$ in **Step** 2 is implemented in lines 11 - 18. The solution checking in **Step** 3 is executed depending

---

[2] Given a flow set $C^{\mathrm{fw}} \subset \mathbb{R}^n \times \mathbb{R}^m$, the set-valued maps $\Psi_c^u : \mathbb{R}^n \to U_C$ is defined for each $x \in \mathbb{R}^n$ as $\Psi_c^u(x) := \{u \in U_C : (x, u) \in C^{\mathrm{fw}}\}$.

---

on the return of the function call $\mathtt{extend}$ and will be further discussed in Section 5. Due to space limitations,

---

Algorithm 1 HyRRT-Connect algorithm

Input: $X_0, X_f, X_u, \mathcal{H}^{\mathrm{fw}} = (C^{\mathrm{fw}}, f^{\mathrm{fw}}, D^{\mathrm{fw}}, g^{\mathrm{fw}}), \mathcal{H}^{\mathrm{bw}}$ $= (C^{\mathrm{bw}}, f^{\mathrm{bw}}, D^{\mathrm{bw}}, g^{\mathrm{bw}}), (\mathcal{U}_C, \mathcal{U}_D), p_n^{\mathrm{fw}}, p_n^{\mathrm{bw}} \in (0, 1), K \in \mathbb{N}_{>0},$ $X_c^{\mathrm{fw}} \supset \overline{C^{\mathrm{fw}'}}, X_d^{\mathrm{fw}} \supset \overline{D^{\mathrm{fw}'}}, X_c^{\mathrm{bw}} \supset \overline{C^{\mathrm{bw}'}}$ and $X_d^{\mathrm{bw}} \supset \overline{D^{\mathrm{bw}'}}.$

1: $\mathcal{T}^{\mathrm{fw}}.\mathtt{init}(X_0)$, $\mathcal{T}^{\mathrm{bw}}.\mathtt{init}(X_f)$
2: **for** $k = 1$ to K **do**
3:     randomly select a real number r^{fw} from $[0, 1]$.
4:     **if** $r^{\mathrm{fw}} \leq p_n^{\mathrm{fw}}$ **then**
5:        $x_{rand}^{\mathrm{fw}} \leftarrow \mathtt{random\_state}(\overline{C^{\mathrm{fw}'}})$.
6:        $\mathtt{extend}(\mathcal{T}^{\mathrm{fw}}, x_{rand}^{\mathrm{fw}}, (\mathcal{U}_C^{\mathrm{fw}}, \mathcal{U}_D^{\mathrm{fw}}), \mathcal{H}^{\mathrm{fw}}, X_u, X_c^{\mathrm{fw}})$.
7:     **else**
8:        $x_{rand}^{\mathrm{fw}} \leftarrow \mathtt{random\_state}(D^{\mathrm{fw}'})$.
9:        $\mathtt{extend}(\mathcal{T}^{\mathrm{fw}}, x_{rand}^{\mathrm{fw}}, (\mathcal{U}_C^{\mathrm{fw}}, \mathcal{U}_D^{\mathrm{fw}}), \mathcal{H}^{\mathrm{fw}}, X_u, X_d^{\mathrm{fw}})$.
10:     **end if**
11:    randomly select a real number r^{bw} from $[0, 1]$.
12:    **if** $r^{\mathrm{bw}} \leq p_n^{\mathrm{bw}}$ **then**
13:       $x_{rand}^{\mathrm{bw}} \leftarrow \mathtt{random\_state}(\overline{C^{\mathrm{bw}'}})$.
14:       $\mathtt{extend}(\mathcal{T}^{\mathrm{bw}}, x_{rand}^{\mathrm{bw}}, (\mathcal{U}_C^{\mathrm{bw}}, \mathcal{U}_D^{\mathrm{bw}}), \mathcal{H}^{\mathrm{bw}}, X_u, X_c^{\mathrm{bw}})$.
15:    **else**
16:       $x_{rand}^{\mathrm{bw}} \leftarrow \mathtt{random\_state}(D^{\mathrm{bw}'})$.
17:       $\mathtt{extend}(\mathcal{T}^{\mathrm{bw}}, x_{rand}^{\mathrm{bw}}, (\mathcal{U}_C^{\mathrm{bw}}, \mathcal{U}_D^{\mathrm{bw}}), \mathcal{H}^{\mathrm{bw}}, X_u, X_d^{\mathrm{bw}})$.
18:    **end if**
19: **end for**

Algorithm 2 Extend function

1: **function** EXTEND$((\mathcal{T}, x, (\mathcal{U}_C, \mathcal{U}_D), \mathcal{H}, X_u, X_*))$
2:    $v_{cur} \leftarrow \mathtt{nearest\_neighbor}(x, \mathcal{T}, \mathcal{H}, X_*)$;
3:    $(\mathtt{is\_a\_new\_vertex\_generated}, x_{new}, \psi_{new}) \leftarrow \mathtt{new\_state}$ $(v_{cur}, (\mathcal{U}_C, \mathcal{U}_D), \mathcal{H}, X_u)$
4:    **if** $\mathtt{is\_a\_new\_vertex\_generated} = true$ **then**
5:      $v_{new} \leftarrow \mathcal{T}.\mathtt{add\_vertex}(x_{new})$;
6:      $\mathcal{T}.\mathtt{add\_edge}(v_{cur}, v_{new}, \psi_{new})$;
7:      **return** Advanced;
8:    **end if**
9:    **return** Trapped;
10: **end function**

---

for the definition of the function calls in Algorithm 1 and Algorithm 2, see Wang and Sanfelice (2024).

5. MOTION PLAN IDENTIFICATION AND RECONSTRUCTION

The following two scenarios are identified where a motion plan can be constructed by utilizing one path from $\mathcal{T}^{\mathrm{fw}}$ and another from $\mathcal{T}^{\mathrm{bw}}$:

S1) A vertex in $\mathcal{T}^{\mathrm{fw}}$ is associated with the same state in the flow set as some vertex in $\mathcal{T}^{\mathrm{bw}}$.
S2) A vertex in $\mathcal{T}^{\mathrm{fw}}$ is associated with a state such that a forward-in-hybrid time jump from such state results in the state associated with some vertex in $\mathcal{T}^{\mathrm{bw}}$, or conversely, a vertex in $\mathcal{T}^{\mathrm{bw}}$ is associated with a state such that a backward-in-hybrid time jump from such state results in the state associated with some vertex in $\mathcal{T}^{\mathrm{fw}}$.

In the HyRRT-Connect algorithm, each of these scenarios is evaluated whenever an Advanced signal is returned by the $\mathtt{extend}$ function. Neglecting approximation errors due to numerical computation, it is typically possible to solve for an exact input at a jump from one state to an other, as required in S2. However, due to the random selection of the

inputs and the family of signals used, satisfying S1 is not typically possible. This may lead to a discontinuity along the flow in the resulting motion plan. A reconstruction process is introduced below to address this issue.

5.1 Same State Associated with Vertices in $\mathcal{T}^{\mathrm{fw}}$ and $\mathcal{T}^{\mathrm{bw}}$

In S1, HyRRT-Connect identifies if there exists a path

$$
\begin{aligned}
p^{\mathrm{fw}} &:= ((v_0^{\mathrm{fw}}, v_1^{\mathrm{fw}}), (v_1^{\mathrm{fw}}, v_2^{\mathrm{fw}}), ..., (v_{m-1}^{\mathrm{fw}}, v_m^{\mathrm{fw}})) \\
&=: (e_0^{\mathrm{fw}}, e_1^{\mathrm{fw}}, ..., e_{m-1}^{\mathrm{fw}})
\end{aligned}
\tag{4}
$$

in $\mathcal{T}^{\mathrm{fw}}$, where $m \in \mathbb{N}$, and a path

$$
\begin{aligned}
p^{\mathrm{bw}} &:= ((v_0^{\mathrm{bw}}, v_1^{\mathrm{bw}}), (v_1^{\mathrm{bw}}, v_2^{\mathrm{bw}}), ..., (v_{n-1}^{\mathrm{bw}}, v_n^{\mathrm{bw}})) \\
&=: (e_0^{\mathrm{bw}}, e_1^{\mathrm{bw}}, ..., e_{n-1}^{\mathrm{bw}})
\end{aligned}
\tag{5}
$$

in $\mathcal{T}^{\mathrm{bw}}$, where $n \in \mathbb{N}$, satisfying the following conditions:

C1) $\overline{x}_{v_0^{\mathrm{fw}}} \in X_0$,
C2) for $i \in \{0, 1, ..., m-2\}$, if $\overline{\psi}_{e_i^{\mathrm{fw}}}$ and $\overline{\psi}_{e_{i+1}^{\mathrm{fw}}}$ are both purely continuous, then $\overline{\psi}_{e_{i+1}^{\mathrm{fw}}}(0,0) \in C^{\mathrm{fw}}$,
C3) $\overline{x}_{v_0^{\mathrm{bw}}} \in X_f$,
C4) for $i \in \{0, 1, ..., n-2\}$, if $\overline{\psi}_{e_i^{\mathrm{bw}}}$ and $\overline{\psi}_{e_{i+1}^{\mathrm{bw}}}$ are both purely continuous, then $\overline{\psi}_{e_{i+1}^{\mathrm{bw}}}(0,0) \in C^{\mathrm{bw}}$,
C5) $\overline{x}_{v_m^{\mathrm{fw}}} = \overline{x}_{v_n^{\mathrm{bw}}}$,
C6) if $\overline{\psi}_{e_{m-1}^{\mathrm{fw}}}$ and $\overline{\psi}_{e_{n-1}^{\mathrm{bw}}}$ are both purely continuous, then $\overline{\psi}_{e_{n-1}^{\mathrm{bw}}}(T^{\mathrm{bw}}, 0) \in C^{\mathrm{fw}}$ where $(T^{\mathrm{bw}}, 0) = \max \mathrm{dom}\, \overline{\psi}_{e_{n-1}^{\mathrm{bw}}}$.

If HyRRT-Connect is able to find a path p^{fw} in $\mathcal{T}^{\mathrm{fw}}$ and a path p^{bw} in $\mathcal{T}^{\mathrm{bw}}$ satisfying C1-C6, then a motion plan to $\mathcal{P}$ can be constructed by $\psi^{\mathrm{fw}}|\psi^{\mathrm{bw}'}$, where, for notation simplicity, $\psi^{\mathrm{fw}} = (\phi^{\mathrm{fw}}, \upsilon^{\mathrm{fw}}) := \tilde{\psi}_{p^{\mathrm{fw}}}$ denotes the solution pair associated with the path p^{fw} in (4) and is referred to as *forward partial motion plan*, $\psi^{\mathrm{bw}} = (\phi^{\mathrm{bw}}, \upsilon^{\mathrm{bw}}) := \tilde{\psi}_{p^{\mathrm{bw}}}$ denotes the solution pair associated with the path p^{bw} in (5) and is referred to as *backward partial motion plan*, and $\psi^{\mathrm{bw}'}$ denotes the reversal of ψ^{bw}. The result $\psi^{\mathrm{fw}}|\psi^{\mathrm{bw}'}$ is guaranteed to satisfy each item in Problem 1 as follows:

(1) By C1, it follows that $\psi^{\mathrm{fw}}|\psi^{\mathrm{bw}'}$ starts from X_0. Namely, item 1 in Problem 1 is satisfied.
(2) Due to C2 (respectively, C4), by iterative applying Proposition 2 to each pair of $\overline{\psi}_{e_i^{\mathrm{fw}}}$ and $\overline{\psi}_{e_{i+1}^{\mathrm{fw}}}$ (respectively, $\overline{\psi}_{e_i^{\mathrm{bw}}}$ and $\overline{\psi}_{e_{i+1}^{\mathrm{bw}}}$) where $i \in \{0, 1, ..., m-2\}$ (respectively, $i \in \{0, 1, ..., n-2\}$), it follows that ψ^{fw} (respectively, ψ^{bw}) is a solution pair to $\mathcal{H}^{\mathrm{fw}}$ (respectively, $\mathcal{H}^{\mathrm{bw}}$). Furthermore, given C5 and C6, Lemma 7 establishes that $\psi^{\mathrm{fw}}|\psi^{\mathrm{bw}'}$ is a solution pair to $\mathcal{H}^{\mathrm{fw}}$.
(3) C3 ensures that $\psi^{\mathrm{fw}}|\psi^{\mathrm{bw}'}$ ends within X_f. This confirms the satisfaction of item 3 in Problem 1.
(4) For any edge $e \in p^{\mathrm{fw}} \cup p^{\mathrm{bw}}$, the trajectory $\overline{\psi}_e$ avoids intersecting the unsafe set as a result of the exclusion of solution pairs that intersect the unsafe set in the function call `new_state`. Therefore, item 4 in Problem 1 is satisfied.

Since each requirement in Problem 1 is satisfied, it is established that $\psi^{\mathrm{fw}}|\psi^{\mathrm{bw}'}$ is a motion plan to $\mathcal{P}$.

In practice, as guaranteeing C5 above is not possible in most hybrid systems, given $\delta > 0$ representing the

tolerance associated with this condition, we implement C5 as

$$
|\overline{x}_{v_m^{\mathrm{fw}}} - \overline{x}_{v_n^{\mathrm{bw}}}| \leq \delta.
\tag{6}
$$

leading to a potential discontinuity during the flow.

5.2 Reconstruction Process

To smoothen and control the discontinuity associated with (6), we propose a reconstruction process. Given the hybrid input υ^{bw} of ψ^{bw} identified in S1, which is backward in hybrid time, the reconstruction process involves simulating a hybrid arc, denoted ϕ^{r}, such that it starts from the final state of ϕ^{fw}, flows when $\upsilon^{\mathrm{bw}'}$ flows, jumps when $\upsilon^{\mathrm{bw}'}$ jumps, and applies the input $(t, j) \mapsto \upsilon^{\mathrm{bw}'}(t, j)$ where $\upsilon^{\mathrm{bw}'}$ denotes the reversal of υ^{bw}; see item 2 in Definition 4 for the reversal of a hybrid input. We generate ϕ^{r} via the following hybrid system, denoted $\mathcal{H}_{\upsilon^{\mathrm{bw}'}}$, with state $x \in \mathbb{R}^n$ and dynamics:

$$
\mathcal{H}_{\upsilon^{\mathrm{bw}'}} : \begin{cases} \dot{x} = f_{\upsilon^{\mathrm{bw}'}}(x, \upsilon^{\mathrm{bw}'}(t, j)) & (t, j) \in C_{\upsilon^{\mathrm{bw}'}} \\ x^+ = g_{\upsilon^{\mathrm{bw}'}}(x, \upsilon^{\mathrm{bw}'}(t, j)) & (t, j) \in D_{\upsilon^{\mathrm{bw}'}} \end{cases}
\tag{7}
$$

where

(1) $D_{\upsilon^{\mathrm{bw}'}} := \{(t, j) \in \mathrm{dom}\, \upsilon^{\mathrm{bw}'} : (t, j+1) \in \mathrm{dom}\, \upsilon^{\mathrm{bw}'}\}$;
(2) $C_{\upsilon^{\mathrm{bw}'}} := \overline{\mathrm{dom}\, \upsilon^{\mathrm{bw}'} \backslash D_{\upsilon^{\mathrm{bw}'}}}$;
(3) $g_{\upsilon^{\mathrm{bw}'}}(x, u) := g(x, u)$ for all [3] $(x, u) \in \mathbb{R}^n \times \mathbb{R}^m$;
(4) $f_{\upsilon^{\mathrm{bw}'}}(x, u) := f(x, u)$ for all $(x, u) \in \mathbb{R}^n \times \mathbb{R}^m$.

In addition to satisfying the hybrid dynamics in (7), we also require that the reconstruction result ϕ^{r} satisfies the following conditions:

R1) $\phi^{\mathrm{r}}(0, 0) = \phi^{\mathrm{fw}}(T^{\mathrm{fw}}, J^{\mathrm{fw}})$, where ϕ^{fw} is the state trajectory of ψ^{fw} identified in S1 and $(T^{\mathrm{fw}}, J^{\mathrm{fw}}) = \max \mathrm{dom}\, \phi^{\mathrm{fw}}$;
R2) ϕ^{r} is a maximal solution to $\mathcal{H}_{\upsilon^{\mathrm{bw}'}}$ such that $\mathrm{dom}\, \phi^{\mathrm{r}} = \mathrm{dom}\, \upsilon^{\mathrm{bw}'}$.

Remark 8. The definitions of $C_{\upsilon^{\mathrm{bw}'}}$ and $D_{\upsilon^{\mathrm{bw}'}}$ indicate that ϕ^{r} follows the flow or jump of $\upsilon^{\mathrm{bw}'}$. R1 ensures that the reconstructed motion plan begins at the final state of the forward partial motion plan, effectively eliminating any discontinuity. Given that R2 ensures that ϕ^{r} is maximal, it follows that $\mathrm{dom}\, \phi^{\mathrm{r}} = \mathrm{dom}\, \upsilon^{\mathrm{bw}'} = \mathrm{dom}\, \phi^{\mathrm{bw}'}$.

Convergence of ϕ^{r} to X_f   We first show the dependency between the difference $|\phi^{\mathrm{r}}(T^{\mathrm{r}}, J^{\mathrm{r}}) - \phi^{\mathrm{bw}}(0, 0)|$ and the tolerance δ in (6) where $(T^{\mathrm{r}}, J^{\mathrm{r}}) = \max \mathrm{dom}\, \phi^{\mathrm{r}}$. The following assumption is imposed on the flow map f of the hybrid system $\mathcal{H}$ in (1).

Assumption 9. 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 $(x_0, u_0) \in C$, $(x_1, u_0) \in C$, and $(x_0, u_1) \in C$, $|f(x_0, u_0) - f(x_1, u_0)| \leq K_x^f|x_0 - x_1|, |f(x_0, u_0) - f(x_0, u_1)| \leq K_u^f|u_0 - u_1|$.

The following Lipschitz assumption is imposed on the jump map g of the hybrid system $\mathcal{H}$ in (1).

Assumption 10. 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 $(x_1, u_1) \in D$, $|g(x_0, u_0) - g(x_1, u_1)| \leq K_x^g|x_0 - x_1| + K_u^g|u_0 - u_1|$.

---

[3] The flow map f and the jump map g in (1) are defined on the domain $\mathbb{R}^n \times \mathbb{R}^m$.

Next, we show that the final state of the reconstructed motion plan ϕ^{r} converges to $\phi^{\mathrm{bw}}(0,0) \in X_f$ as the tolerance δ in (6) approaches zero. See Wang and Sanfelice (2024) for a detailed proof.

Theorem 11. Suppose Assumptions 9 and 10 are satisfied, and there exist a solution pair $\psi^{\mathrm{fw}} = (\phi^{\mathrm{fw}}, \upsilon^{\mathrm{fw}})$ to $\mathcal{H}^{\mathrm{fw}}$ and a solution pair $\psi^{\mathrm{bw}} = (\phi^{\mathrm{bw}}, \upsilon^{\mathrm{bw}})$ to $\mathcal{H}^{\mathrm{bw}}$ identified in S1. For each $\epsilon > 0$, there exists a tolerance $\delta > 0$ in (6) such that $|\phi^{\mathrm{fw}}(T^{\mathrm{fw}}, J^{\mathrm{fw}}) - \phi^{\mathrm{bw}}(T^{\mathrm{bw}}, J^{\mathrm{bw}})| \leq \delta$ leads to $|\phi^{\mathrm{r}}(T^{\mathrm{r}}, J^{\mathrm{r}}) - \phi^{\mathrm{bw}}(0,0)| \leq \epsilon$ where $(T^{\mathrm{fw}}, J^{\mathrm{fw}}) = \max \mathrm{dom}\, \phi^{\mathrm{fw}}$, $(T^{\mathrm{bw}}, J^{\mathrm{bw}}) = \max \mathrm{dom}\, \phi^{\mathrm{bw}}$, ϕ^{r} is a solution to $\mathcal{H}_{\upsilon^{\mathrm{bw}\prime}}$ following R1 and R2, and $(T^{\mathrm{r}}, J^{\mathrm{r}}) = \max \mathrm{dom}\, \phi^{\mathrm{r}}$.

Furthermore, if $\phi^{\mathrm{bw}}(0,0)$ is not on the boundary of X_f, the following result shows there is a tolerance ensuring that ϕ^{r} concludes within X_f.

Corollary 12. Suppose Assumptions 9 and 10 are satisfied, and there exist a solution pair $\psi^{\mathrm{fw}} = (\phi^{\mathrm{fw}}, \upsilon^{\mathrm{fw}})$ to $\mathcal{H}^{\mathrm{fw}}$, and a solution pair $\psi^{\mathrm{bw}} = (\phi^{\mathrm{bw}}, \upsilon^{\mathrm{bw}})$ to $\mathcal{H}^{\mathrm{bw}}$ identified in S1, and some $\epsilon' > 0$ such that $\phi^{\mathrm{bw}}(0,0) + \epsilon'\mathbb{B} \subset X_f$. Then, there exists a tolerance $\delta > 0$ in (6) such that $|\phi^{\mathrm{fw}}(T^{\mathrm{fw}}, J^{\mathrm{fw}}) - \phi^{\mathrm{bw}}(T^{\mathrm{bw}}, J^{\mathrm{bw}})| \leq \delta$ leads to $\phi^{\mathrm{r}}(T^{\mathrm{r}}, J^{\mathrm{r}}) \in X_f$ where $(T^{\mathrm{fw}}, J^{\mathrm{fw}}) = \max \mathrm{dom}\, \phi^{\mathrm{fw}}$, $(T^{\mathrm{bw}}, J^{\mathrm{bw}}) = \max \mathrm{dom}\, \phi^{\mathrm{bw}}$, ϕ^{r} is a solution to $\mathcal{H}_{\upsilon^{\mathrm{bw}\prime}}$ following R1 and R2, and $(T^{\mathrm{r}}, J^{\mathrm{r}}) = \max \mathrm{dom}\, \phi^{\mathrm{r}}$.

Proof. By selecting $\epsilon = \epsilon'$, Theorem 11 ensures the existence of some $\delta > 0$ such that $|\phi^{\mathrm{r}}(T^{\mathrm{r}}, J^{\mathrm{r}}) - \phi^{\mathrm{bw}}(0,0)| \leq \epsilon'$. By $\phi^{\mathrm{bw}}(0,0) + \epsilon'\mathbb{B} \subset X_f$, it is established that $\phi^{\mathrm{r}}(T^{\mathrm{r}}, J^{\mathrm{r}}) \in X_f$. $\qquad\square$

Then, by replacing ϕ^{bw} with ϕ^{r} and concatenating the reconstructed pair $\psi^{\mathrm{r}} := (\phi^{\mathrm{r}}, \upsilon^{\mathrm{bw}\prime})$ to $\psi^{\mathrm{fw}} = (\phi^{\mathrm{fw}}, \upsilon^{\mathrm{fw}})$, HyRRT-Connect generates the motion plan $\psi^{\mathrm{fw}}|\psi^{\mathrm{r}}$, where the discontinuity associated with (6) is removed. Note that the tolerance δ in (6) is adjustable. Setting δ to a smaller value brings the endpoint of ϕ^{r} closer to X_f, However, it also reduces the possibility of finding a motion plan, thereby increasing the time expected to find forward and backward partial motion plans.

5.3 Connecting Forward and Backward Search Trees via Jumps

In S2, HyRRT-Connect checks the existence of p^{fw} in (4) and p^{bw} in (5) which, in addition to meeting C1-C4 in Section 5.1, results in a solution to the following constrained equation, denoted u^*, provided one exists[4]:

$$\overline{x}_{v_n^{\mathrm{bw}}} = g(\overline{x}_{v_m^{\mathrm{fw}}}, u^*), \quad (\overline{x}_{v_m^{\mathrm{fw}}}, u^*) \in D^{\mathrm{fw}}. \qquad (8)$$

The constrained equation above can be solved analytically for certain hybrid systems such as the one in Example 1 and numerically Boyd and Vandenberghe (2004) in general. A solution to (8) implies that $\overline{x}_{v_m^{\mathrm{fw}}}$ and $\overline{x}_{v_n^{\mathrm{bw}}}$ can be connected by applying u^* at a jump from $\overline{x}_{v_m^{\mathrm{fw}}}$ to $\overline{x}_{v_n^{\mathrm{bw}}}$. Hence, a motion plan is constructed by concatenating ψ^{fw}, a single jump from $\overline{x}_{v_m^{\mathrm{fw}}}$ to $\overline{x}_{v_n^{\mathrm{bw}}}$, and $\psi^{\mathrm{bw}\prime}$. This approach constructs a motion plan before detecting overlaps between $\mathcal{T}^{\mathrm{fw}}$ and $\mathcal{T}^{\mathrm{bw}}$ in S1, improving efficiency and preventing the discontinuity introduced by (6) through a jump.

---

[4] It is indeed possible that all the motion plans are purely continuous. In this case, no solution to (8) would be found since no jumps exist in every motion plan.

6. SOFTWARE TOOL AND SIMULATION RESULTS

Algorithm 1 leads to a software tool[5] to solve Problem 1. Next, we illustrate the HyRRT-Connect algorithm and this tool in Example 1 and Example 2.

Example 3. (Actuated bouncing ball system in Example 1, revisited) We initially showcase the simulation results of the HyRRT-Connect algorithm without the functionality of connecting via jumps discussed in Section 5.3. We consider the case where HyRRT-Connect precisely connects the forward and backward partial motion plans. This is demonstrated by deliberately setting the initial state set as $X_0 = \{(14, 0)\}$ and the final state set as $X_f = \{(0, -16.58)\}$. In this case, no tolerance is applied, and thus, no reconstruction process is employed. The motion plan detected under these settings is depicted in Figure 1(a), where the forward and backward partial motion plans identified in S1 are depicted by the green and magenta lines, respectively. However, for most scenarios, such as $X_0 = \{(14, 0)\}$ and $X_f = \{(10, 0)\}$ in Example 1, if we require strict equality without allowing any tolerance, then HyRRT-Connect fails to return a motion plans in almost all the runs. This demonstrates the necessity of allowing a certain degree of tolerance in HyRRT-Connect. The simulation results, allowing a tolerance of $\delta = 0.2$, are shown in Figure 1(b). A discontinuity during the flow between the forward and backward partial motion plans is observed, as depicted in the red circle in Figure 1(b). This discontinuity is addressed through the reconstruction process, as is shown in Figure 1(c). A deviation between the endpoint of the reconstructed motion plan and the final state set is also observed in Figure 1(c), which, according to Theorem 11, is bounded.

Next, we proceed to perform simulation results of HyRRT-Connect showcasing its full functionalities, including the ability to connect partial motion plans via jumps. Figure 1(d) shows this situation. This feature enables HyRRT-Connect to avoid discontinuities during the flow, as it computes exact solutions at jumps to connect forward and backward partial motion plans. Furthermore, we compare the computational performance of the proposed HyRRT-Connect algorithm, its variant Bi-HyRRT (where the function to connect partial motion plans via jumps is disabled), and HyRRT given in Wang and Sanfelice (2022). Conducted on a 3.5GHz Intel Core i7 processor using MATLAB, each algorithm is run 20 times on the same problem. HyRRT-Connect on average creates 78.8 vertices in 0.27 seconds, Bi-HyRRT 186.5 vertices in 0.76 seconds, and HyRRT 457.4 vertices in 3.93 seconds. Compared to HyRRT, both HyRRT-Connect and Bi-HyRRT show considerable improvements in computational efficiency. Notably, HyRRT-Connect, with its jump-connecting capability, achieves a 64.5% reduction in computation time and 57.7% fewer vertices than Bi-HyRRT, demonstrating the benefits of jump connections.

Example 4. (Walking robot system in Example 2, revisited) The simulation results demonstrate that HyRRT-Connect successfully finds a motion plan for the high-dimensional walking robot system with a tolerance δ of 0.3. The forward search tree $\mathcal{T}^{\mathrm{fw}}$, with its partial motion plan shown in green, is displayed in Figure 2(a). Similarly, the backward search tree $\mathcal{T}^{\mathrm{bw}}$, with its partial motion plan

---

[5] Code at https://github.com/HybridSystemsLab/HyRRTConnect.git.
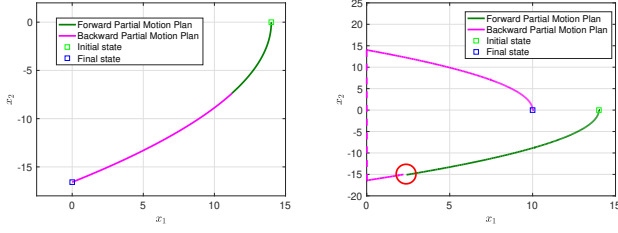
(a) Precise connection during the flow is achieved.

(b) A discontinuity during the flow in red circle.
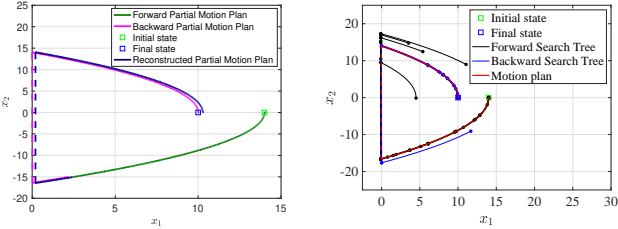
(c) The backward partial motion plan is reconstructed.

(d) HyRRT-Connect

Fig. 1. Motion plans for the actuated bouncing ball example.

(a) Forward search tree and partial motion plan.

(b) Backward search tree and partial motion plan.

Fig. 2. Selected states of the partial motion plans for the walking robot system.

set as discontinuity tolerance decreases. The effectiveness and computational improvement of HyRRT-Connect are exemplified through applications to an actuated bouncing ball and a walking robot.
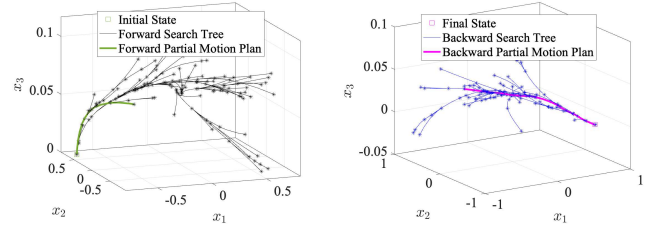
in magenta, is shown in Figure 2(b). Running HyRRT-Connect and HyRRT 20 times each for the same problem, HyRRT-Connect generates 470.2 vertices and takes 19.8 seconds, while HyRRT generates 2357.1 vertices and takes 71.5 seconds. This indicates a significant 72.3% improvement in computation time and 80.1% in vertex creation for HyRRT-Connect compared to HyRRT, highlighting the efficiency of bidirectional exploration.

7. DISCUSSION ON PARALLEL IMPLEMENTATION

After each new vertex is added to the search tree, the HyRRT-Connect algorithm frequently halts and restarts parallel computations to check for overlaps between the forward and backward search trees. This process can prevent the potential computational performance improvement from parallelization. Our simulations using MATLAB's `parpool` for parallel computation showed no improvement compared to an interleaved approach. In fact, the parallel implementation took significantly longer — about 2.47 seconds for the actuated bouncing ball system, compared to 0.27 second with interleaving, and 167.8 seconds for the walking robot system versus only 19.8 seconds. It is important to note that the time required for halting and restarting parallel computation is contingent upon factors like the specific parallel computation software toolbox used, the hardware platform, and other implementation details. Consequently, the conclusions regarding performance may differ with varying implementations.

8. CONCLUSION

In this paper, we present HyRRT-Connect, a bidirectional algorithm designed to solve the motion planning problems for hybrid systems. This algorithm includes a novel backward-in-time hybrid system formulation, validated by reversing and concatenating functions on the hybrid time domain. To tackle potential flow discontinuities between forward and backward motion plans, we introduce a reconstruction process. In addition to smoothening the discontinuity, this process ensures convergence to the final state

REFERENCES

Boyd, S.P. and Vandenberghe, L. (2004). *Convex optimization.* Cambridge university press.

Chai, J. and Sanfelice, R.G. (2018). Forward invariance of sets for hybrid dynamical systems (part i). *IEEE Transactions on Automatic Control*, 64(6), 2426–2441.

Herbert, S.L., Chen, M., Han, S., Bansal, S., Fisac, J.F., and Tomlin, C.J. (2017). Fastrack: A modular framework for fast and guaranteed safe motion planning. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 1517–1522. IEEE.

Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *The international journal of robotics research*, 5(1), 90–98.

Kuffner, J.J. and LaValle, S.M. (2000). Rrt-connect: An efficient approach to single-query path planning. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 2, 995–1001. IEEE.

LaValle, S.M. (2006). *Planning algorithms.* Cambridge university press.

LaValle, S.M. and Kuffner Jr, J.J. (2001). Randomized kinodynamic planning. *The international journal of robotics research*, 20(5), 378–400.

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

Sanfelice, R.G. (2021). *Hybrid feedback control.* Princeton University Press.

Song, M., Wang, N., Gordon, T., and Wang, J. (2019). Flow-field guided steering control for rigid autonomous ground vehicles in low-speed manoeuvring. *Vehicle System Dynamics*, 57(8), 1090–1107.

Wang, N. and Sanfelice, R.G. (2022). A rapidly-exploring random trees motion planning algorithm for hybrid dynamical systems. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, 2626–2631. IEEE.

Wang, N. and Sanfelice, R.G. (2023). Hysst: An asymptotically near-optimal motion planning algorithm for hybrid systems. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, 2865–2870. IEEE.

Wang, N. and Sanfelice, R.G. (2024). Hyrrt-connect: A bidirectional rapidly-exploring random

trees motion planning algorithm for hybrid dynamical systems. Technical report, University of California, Santa Cruz, Department of Electrical and Computer Engineering. URL: https://hybrid.soe.ucsc.edu/sites/default/files/preprints/TR-HSL-04-2023.pdf, password: `HyRRTConnect23`.

Wang, N., Song, M., Wang, J., and Gordon, T. (2017). A flow-field guided method of path planning for unmanned ground vehicles. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 2762–2767. IEEE.

Wilfong, G.T. (1988). Motion planning for an autonomous vehicle. In *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, 529–533. IEEE.