

HyRRT-Connect: An Efficient Bidirectional Rapidly-Exploring Random Trees Motion Planning Algorithm for Hybrid Dynamical Systems^{*}

Nan Wang^{*} Ricardo G. Sanfelice^{*}

^{*} *Department of Electrical and Computer Engineering,
University of California, 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 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 that the motion plan 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, effectively eliminating the discontinuity. The proposed algorithm is applied to an actuated bouncing ball system to highlight its computational improvement.

Keywords: Hybrid systems, Motion planning, RRT, Robotics

1. INTRODUCTION

In the context of dynamical systems, 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 dynamical systems and purely discrete-time dynamical systems has been extensively explored in existing literature; see, e.g., LaValle (2006). Some examples include graph search algorithms, artificial potential field method and sampling-based algorithms. The sampling-based algorithms have drawn much attention because of their fast exploration speed for high-dimensional problems as well as their theoretical guarantees; especially, probabilistic completeness, which means that the probability of failing to find a motion plan converges to zero, as the number of iterations 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 most successful 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 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. For a special class of hybrid systems like continuous-time hybrid systems or hybrid automata, some RRT-type motion planning algorithms, such as hybrid RRT (Branicky et al., 2003) and R3T (Wu et al., 2020), respectively, have been developed. These methodologies have also found application in falsification of hybrid automata such as the Monte-Carlo sampling algorithm (Nghiem et al., 2010) and the Breach toolbox (Donzé, 2010). However, this class of hybrid systems lacks the generality needed to cope with hybrid system arising in robotics applications, especially those with state variables capable of both continuous evolution and discrete behaviors. In recent work, Wang and Sanfelice (2022) formulates a motion planning problem for hybrid systems using hybrid equations, as in Sanfelice (2021). This formulation presents a general framework that encompasses a broad class of hybrid systems. In the same paper, a probabilistically complete RRT algorithm, referred to as HyRRT, specifically designed to address the motion planning problem for hybrid systems, is introduced. Building on this work, Wang and Sanfelice (2023) introduces HySST, an asymptotically near-optimal motion planning algorithm for hybrid systems from the Stable Sparse RRT (SST) algorithm. HySST stands as the first optimal motion planning algorithm for hybrid systems and its computational efficiency is notable, achieved through the sparsification of vertices during the search process.

It is significantly challenging for the majority of motion planning algorithms to maintain efficient computation performance, especially in solving high-dimensional problems. To improve the computational performance, a modular mo-

^{*} This research is supported by NSF grants nos. CNS-2039054 and CNS-2111688, by AFOSR grants nos. FA9550-19-1-0169, FA9550-20-1-0238, FA9550-23-1-0145, and FA9550-23-1-0313, by AFRL grant nos. FA8651-22-1-0017 and FA8651-23-1-0004, by ARO grant no. W911NF-20-1-0253, and by DoD grant no. W911NF-23-1-0158.

tion planning system for purely continuous-time systems, named FaSTrack (Herbert et al., 2017), is designed to simultaneously plan and track a trajectory. FaSTrack accelerates the planning process by only considering a low-dimensional model of the system dynamics. Kuffner and LaValle (2000) introduces RRT-Connect, an algorithm that propagates both in forward direction and backward direction, leading to a notable improvement in computational performance. Inspired by this work, we design a bidirectional RRT-type algorithm for hybrid dynamical systems, called HyRRT-Connect. HyRRT-Connect incrementally constructs two search trees, one rooted in the initial state set and constructed forward in hybrid time, and the other rooted in the final state set and constructed backward in hybrid time. To the best of the authors' knowledge, this is the first bidirectional RRT-type algorithm for systems with hybrid dynamics. The proposed algorithm is illustrated in an actuated bouncing ball system, showing a significant improvement in computational performance, which highlights the efficiency of the proposed algorithm.

The remainder of the paper is structured as follows. Section 2 presents notation and preliminaries. Section 3 presents the problem statement and introduces an example. Section 4 presents HyRRT-Connect algorithm. Section 5 presents the reconstruction process. Section 6 illustrates HyRRT-Connect in an example. Proofs and more details will be published elsewhere.

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 $\text{int } I$ denotes the interior of the interval I . The notation \bar{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 $\{p + q : p \in P, q \in Q\}$.

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, $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ represents the flow map, $D \subset \mathbb{R}^n \times \mathbb{R}^m$ represents the jump set, and $g : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ 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 $\Pi_\star(K) := \{x : \exists u \in U_\star \text{ such that } (x, u) \in K\}$ as the projection of K onto \mathbb{R}^n , and define

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

The domain of a solution to \mathcal{H} , hybrid input/arc, solution pair to \mathcal{H} , and concatenation operation are defined as follows; see Sanfelice (2021) for more details.

Definition 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 $E \cap ([0, T] \times \{0, 1, \dots, J\})$ 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 $0 = t_0 \leq t_1 \leq t_2 \leq \dots \leq t_{J+1} = T$.

Definition 2. (Hybrid input). A function $v : \text{dom } v \rightarrow \mathbb{R}^m$ is a hybrid input if $\text{dom } v$ is a hybrid time domain and if, for each $j \in \mathbb{N}$, $t \mapsto v(t, j)$ is Lebesgue measurable and locally essentially bounded on the interval $I_v^j := \{t : (t, j) \in \text{dom } v\}$.

Definition 3. (Hybrid arc). A function $\phi : \text{dom } \phi \rightarrow \mathbb{R}^n$ is a hybrid arc if $\text{dom } \phi$ 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 $I_\phi^j := \{t : (t, j) \in \text{dom } \phi\}$.

Definition 4. (Solution pair to a hybrid system). A hybrid input v and a hybrid arc ϕ define a solution pair (ϕ, v) to the hybrid system \mathcal{H} as in (1) if

- 1) $(\phi(0, 0), v(0, 0)) \in \bar{C} \cup D$ and $\text{dom } \phi = \text{dom } v (= \text{dom } (\phi, v))$.
- 2) For each $j \in \mathbb{N}$ such that I_ϕ^j has nonempty interior $\text{int}(I_\phi^j)$, (ϕ, v) satisfies $(\phi(t, j), v(t, j)) \in C$ for all $t \in \text{int } I_\phi^j$, and $\frac{d}{dt} \phi(t, j) = f(\phi(t, j), v(t, j))$ for almost all $t \in I_\phi^j$.
- 3) For all $(t, j) \in \text{dom } (\phi, v)$ such that $(t, j + 1) \in \text{dom } (\phi, v)$, $(\phi(t, j), v(t, j)) \in D$, $\phi(t, j + 1) = g(\phi(t, j), v(t, j))$.

Definition 5. (Concatenation operation). Given two functions $\phi_1 : \text{dom } \phi_1 \rightarrow \mathbb{R}^n$ and $\phi_2 : \text{dom } \phi_2 \rightarrow \mathbb{R}^n$, where $\text{dom } \phi_1$ and $\text{dom } \phi_2$ are hybrid time domains, ϕ_2 can be concatenated to ϕ_1 if ϕ_1 is compact and $\phi : \text{dom } \phi \rightarrow \mathbb{R}^n$ is the concatenation of ϕ_2 to ϕ_1 , denoted $\phi = \phi_1 | \phi_2$, namely,

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

3. PROBLEM STATEMENT AND APPLICATIONS

The motion planning problem for hybrid systems studied in this paper is given 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 $x \in \mathbb{R}^n$, initial state set $X_0 \subset \mathbb{R}^n$, final state set $X_f \subset \mathbb{R}^n$, and unsafe set $X_u \subset \mathbb{R}^n \times \mathbb{R}^m$, find $(\phi, v) : \text{dom } (\phi, v) \rightarrow \mathbb{R}^n \times \mathbb{R}^m$, namely, a *motion plan*, such that, for some $(T, J) \in \text{dom } (\phi, v)$, 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, v) is a solution pair to \mathcal{H} as defined in Definition 4;
- 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) $(\phi(t, j), v(t, j)) \notin X_u$ for each $(t, j) \in \text{dom } (\phi, v)$ 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 $\mathcal{P} = (X_0, X_f, X_u, (C, f, D, g))$.

Consider the following motivation example.

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 $\dot{x} = \begin{bmatrix} x_2 \\ -\gamma \end{bmatrix} =: f(x, u)$ when $(x, u) \in C$, where $x := (x_1, x_2) \in \mathbb{R}^2$, the height of the ball is denoted by x_1 , and the velocity of the ball is denoted by x_2 . The gravity constant is denoted by γ . The ball is allowed to flow when the ball is on or 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 are given by $x^+ = \begin{bmatrix} x_1 \\ -\lambda x_2 + u \end{bmatrix} =: g(x, u)$ when $(x, u) \in D$, where $u \geq 0$ is the input applied only at jumps 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 eventually reaches a given target height with zero velocity. The input is constrained to be positive and upper bounded by a specific positive real number. 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.

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, as described 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 $p = (v_1, v_2, \dots, v_k)$ such that $(v_i, v_{i+1}) \in E$ for all $i = \{1, 2, \dots, k-1\}$.

The search tree constructed forward in hybrid time is denoted as $\mathcal{T}^{\text{fw}} = (V^{\text{fw}}, E^{\text{fw}})$ and the search tree con-

structed backward in hybrid time is denoted as $\mathcal{T}^{\text{bw}} = (V^{\text{bw}}, E^{\text{bw}})$. For consistency, we denote \mathcal{H} in (1) as $\mathcal{H}^{\text{fw}} = (C^{\text{fw}}, f^{\text{fw}}, D^{\text{fw}}, g^{\text{fw}})$. Each vertex v in V^{fw} (respectively, V^{bw}) is associated with a state of \mathcal{H}^{fw} (respectively, the hybrid system that represents inverse dynamics of \mathcal{H}^{fw} , denoted \mathcal{H}^{bw}), denoted \bar{x}_v . Each edge e in E^{fw} (respectively, E^{bw}) is associated with a solution pair to \mathcal{H}^{fw} (respectively, \mathcal{H}^{bw}), denoted $\bar{\psi}_e$, that connects the states associated with their endpoint vertices. The solution pair associated with the path $p = (v_1, v_2, \dots, v_k)$ is the concatenation of all the solutions associated with the edges therein, namely, $\tilde{\psi}_p := \bar{\psi}_{(v_1, v_2)} | \bar{\psi}_{(v_2, v_3)} | \dots | \bar{\psi}_{(v_{k-1}, v_k)}$, where $\tilde{\psi}_p$ denotes the solution pair associated with p .

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

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

- Step 1:** Sample a finite number of points from X_0 (respectively, X_f) and initialize a search tree $\mathcal{T}^{\text{fw}} = (V^{\text{fw}}, E^{\text{fw}})$ (respectively, $\mathcal{T}^{\text{bw}} = (V^{\text{bw}}, E^{\text{bw}})$) by adding vertices associated with each sample.
- Step 2:** Incrementally construct \mathcal{T}^{fw} forward in hybrid time and \mathcal{T}^{bw} backward in hybrid time, executing both procedures in an interleaved manner.
- Step 3:** If an appropriate overlap between \mathcal{T}^{fw} and \mathcal{T}^{bw} is found, reverse the solution pair in \mathcal{T}^{bw} , concatenate it to the solution pair in \mathcal{T}^{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 $\mathcal{H}^{\text{fw}} = (C^{\text{fw}}, f^{\text{fw}}, D^{\text{fw}}, g^{\text{fw}})$, denoted $\mathcal{H}^{\text{bw}} = (C^{\text{bw}}, f^{\text{bw}}, D^{\text{bw}}, g^{\text{bw}})$, is required when propagating trajectories from X_f . The construction of \mathcal{H}^{bw} is as follows.

Definition 6. (Backward-in-time hybrid system). Given a hybrid system $\mathcal{H}^{\text{fw}} = (C^{\text{fw}}, f^{\text{fw}}, D^{\text{fw}}, g^{\text{fw}})$, the backward-in-time hybrid system of \mathcal{H}^{fw} , denoted \mathcal{H}^{bw} , is defined as

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

where

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

While the jump map g^{fw} of the forward-in-time system \mathcal{H}^{fw} is single-valued, the corresponding map g^{bw} in \mathcal{H}^{bw} may not be, especially if g^{bw} is not invertible. Therefore, a difference inclusion in (3) governs the discrete dynamics.

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}^{bw} and concatenates it with a solution pair associated with a path detected in \mathcal{T}^{fw} , according to Definition 5. The following result shows that the resulting concatenation is a solution pair to \mathcal{H}^{fw} under mild conditions.

Proposition 7. Given two solution pairs $\psi_1 = (\phi_1, v_1)$ and $\psi_2 = (\phi_2, v_2)$ to a hybrid system \mathcal{H}^{fw} , their concatenation $\psi = (\phi, v) = (\phi_1 | \phi_2, v_1 | v_2)$, denoted $\psi = \psi_1 | \psi_2$, is a solution pair to \mathcal{H}^{fw} if the following hold:

- 1) $\psi_1 = (\phi_1, v_1)$ is compact;
- 2) $\phi_1(T, J) = \phi_2(0, 0)$, where $(T, J) = \max \text{dom } \psi_1$;
- 3) If both $I_{\psi_1}^J$ and $I_{\psi_2}^0$ have nonempty interior, where $I_{\psi}^j = \{t : (t, j) \in \text{dom } \psi\}$ and $(T, J) = \max \text{dom } \psi_1$, then $\psi_2(0, 0) \in C$.

The reversal operation is defined next.

Definition 8. (Reversal of a solution pair). Given a compact solution pair (ϕ, v) to $\mathcal{H}^{\text{fw}} = (C^{\text{fw}}, f^{\text{fw}}, D^{\text{fw}}, g^{\text{fw}})$, where $\phi : \text{dom } \phi \rightarrow \mathbb{R}^n$, $v : \text{dom } v \rightarrow \mathbb{R}^m$, and $(T, J) = \max \text{dom } (\phi, v)$, the pair (ϕ', v') is the reversal of (ϕ, v) , where $\phi' : \text{dom } \phi' \rightarrow \mathbb{R}^n$ with $\text{dom } \phi' \subset \mathbb{R}_{\geq 0} \times \mathbb{N}$ and $v' : \text{dom } v' \rightarrow \mathbb{R}^m$ with $\text{dom } v' = \text{dom } \phi'$, if:

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

Proposition 9 shows that the reversal of the solution pair is a solution pair to its backward-in-time hybrid system.

Proposition 9. Given a hybrid system \mathcal{H}^{fw} and its backward-in-time system \mathcal{H}^{bw} , if $\psi = (\phi, v)$ is a compact solution pair to \mathcal{H}^{fw} , then the reversal $\psi' = (\phi', v')$ of $\psi = (\phi, v)$ is a compact solution pair to \mathcal{H}^{bw} .

The following assumption on the solution pairs that are used to construct motion plans integrates the conditions in Proposition 7 and Proposition 9.

Assumption 10. Given a solution pair $\psi_1 = (\phi_1, v_1)$ to a hybrid system $\mathcal{H}^{\text{fw}} = (C^{\text{fw}}, f^{\text{fw}}, D^{\text{fw}}, g^{\text{fw}})$ and a solution pair $\psi_2 = (\phi_2, v_2)$ to the backward-in-time hybrid system \mathcal{H}^{bw} associated to \mathcal{H}^{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 $(T_1, J_1) = \max \text{dom } \psi_1$ and $(T_2, J_2) = \max \text{dom } \psi_2$;
- 3) If both $I_{\psi_1}^{J_1}$ and $I_{\psi_2}^{J_2}$ have nonempty interior, where $I_{\psi}^j = \{t : (t, j) \in \text{dom } \psi\}$, $(T_1, J_1) = \max \text{dom } \psi_1$, and $(T_2, J_2) = \max \text{dom } \psi_2$, then $\psi_2(T_2, J_2) \in C$.

The following result validates that the hybrid signal constructed using the solution pairs satisfying Assumption 10 is a solution pair to \mathcal{H}^{fw} .

Lemma 11. Given a hybrid system \mathcal{H}^{fw} and its backward-in-time hybrid system \mathcal{H}^{bw} , if ψ_1 is a solution pair to \mathcal{H}^{fw} and ψ_2 is a solution pair to \mathcal{H}^{bw} such that ψ_1 and ψ_2 satisfy Assumption 10, then the concatenation $\psi = \psi_1 | \psi_2'$ is a solution pair to \mathcal{H}^{fw} , where ψ_2' is the reversal of ψ_2 .

4.4 HyRRT-Connect Algorithm

The proposed algorithm is given in Algorithm 1. **Step 1** in Section 4.1 corresponds to the function calls $\mathcal{T}^{\text{fw}}.\text{init}$ and $\mathcal{T}^{\text{bw}}.\text{init}$ in line 1 of Algorithm 1. The construction of \mathcal{T}^{fw} in **Step 2** is implemented in lines 3 - 10. The construction of \mathcal{T}^{bw} in **Step 2** is implemented in lines 11 - 18. The solution checking in **Step 3** is executed depending on the return of the function call **extend** and will be further discussed in Section 5. For the function calls in Algorithm 1 and Algorithm 2, see Wang and Sanfelice (2022).

Algorithm 1 HyRRT-Connect algorithm

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

- 1: $\mathcal{T}^{\text{fw}}.\text{init}(X_0), \mathcal{T}^{\text{bw}}.\text{init}(X_f)$
- 2: **for** $k = 1$ to K **do**
- 3: randomly select a real number r^{fw} from $[0, 1]$.
- 4: **if** $r^{\text{fw}} \leq p_n^{\text{fw}}$ **then**
- 5: $x_{\text{rand}}^{\text{fw}} \leftarrow \text{random_state}(\overline{C^{\text{fw}'}})$.
- 6: **extend** $(\mathcal{T}^{\text{fw}}, x_{\text{rand}}^{\text{fw}}, (\mathcal{U}_C^{\text{fw}}, \mathcal{U}_D^{\text{fw}}), \mathcal{H}^{\text{fw}}, X_u, X_c^{\text{fw}})$.
- 7: **else**
- 8: $x_{\text{rand}}^{\text{fw}} \leftarrow \text{random_state}(D^{\text{fw}'})$.
- 9: **extend** $(\mathcal{T}^{\text{fw}}, x_{\text{rand}}^{\text{fw}}, (\mathcal{U}_C^{\text{fw}}, \mathcal{U}_D^{\text{fw}}), \mathcal{H}^{\text{fw}}, X_u, X_d^{\text{fw}})$.
- 10: **end if**
- 11: randomly select a real number r^{bw} from $[0, 1]$.
- 12: **if** $r^{\text{bw}} \leq p_n^{\text{bw}}$ **then**
- 13: $x_{\text{rand}}^{\text{bw}} \leftarrow \text{random_state}(\overline{C^{\text{bw}'}})$.
- 14: **extend** $(\mathcal{T}^{\text{bw}}, x_{\text{rand}}^{\text{bw}}, (\mathcal{U}_C^{\text{bw}}, \mathcal{U}_D^{\text{bw}}), \mathcal{H}^{\text{bw}}, X_u, X_c^{\text{bw}})$.
- 15: **else**
- 16: $x_{\text{rand}}^{\text{bw}} \leftarrow \text{random_state}(D^{\text{bw}'})$.
- 17: **extend** $(\mathcal{T}^{\text{bw}}, x_{\text{rand}}^{\text{bw}}, (\mathcal{U}_C^{\text{bw}}, \mathcal{U}_D^{\text{bw}}), \mathcal{H}^{\text{bw}}, X_u, X_d^{\text{bw}})$.
- 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: $v_{\text{cur}} \leftarrow \text{nearest_neighbor}(x, \mathcal{T}, \mathcal{H}, X_*)$;
- 3: **(is_a_new_vertex_generated, } $x_{\text{new}}, \psi_{\text{new}}) \leftarrow \text{new_state}(v_{\text{cur}}, (\mathcal{U}_C, \mathcal{U}_D), \mathcal{H}, X_u)$**
- 4: **if** **is_a_new_vertex_generated** = **true** **then**
- 5: $v_{\text{new}} \leftarrow \mathcal{T}.\text{add_vertex}(x_{\text{new}})$;
- 6: $\mathcal{T}.\text{add_edge}(v_{\text{cur}}, v_{\text{new}}, \psi_{\text{new}})$;
- 7: **return** **Advanced**;
- 8: **end if**
- 9: **return** **Trapped**;
- 10: **end function**

5. MOTION PLAN RECONSTRUCTION

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

- S1) A vertex in \mathcal{T}^{fw} is associated with the same state in the flow set as some vertex in \mathcal{T}^{bw} .
- S2) A vertex in \mathcal{T}^{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}^{bw} , or conversely, a vertex in \mathcal{T}^{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}^{fw} .

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}^{fw} and \mathcal{T}^{bw}
In S1, HyRRT-Connect identifies if there exists a path

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

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

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

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

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

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

- 1) By C1, it follows that $\psi^{\text{fw}}|\psi^{\text{bw}'}$ starts from X_0 . Namely, item 1 in Problem 1 is satisfied.
- 2) Due to C2 (respectively, C4), by iterative applying Proposition 7 to each pair of $\bar{\psi}_{e_i^{\text{fw}}}$ and $\bar{\psi}_{e_{i+1}^{\text{fw}}}$ (respec-

tively, $\bar{\psi}_{e_i^{\text{bw}}}$ and $\bar{\psi}_{e_{i+1}^{\text{bw}}}$) where $i \in \{0, 1, \dots, m-2\}$ (respectively, $i \in \{0, 1, \dots, n-2\}$), it follows that ψ^{fw} (respectively, ψ^{bw}) is a solution pair to \mathcal{H}^{fw} (respectively, \mathcal{H}^{bw}). Furthermore, given C5 and C6, Lemma 11 establishes that $\psi^{\text{fw}}|\psi^{\text{bw}'}$ is a solution pair to \mathcal{H}^{fw} .

- 3) C3 ensures that $\psi^{\text{fw}}|\psi^{\text{bw}'}$ ends within X_f . This confirms the satisfaction of item 3 in Problem 1.
- 4) For any edge $e \in p^{\text{fw}} \cup p^{\text{bw}}$, the trajectory $\bar{\psi}_e$ 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.

In practice, guaranteeing C5 above is not possible in most hybrid systems. Thus, given $\delta > 0$ representing the tolerance associated with C5, we implement C5 as

$$|\bar{x}_{v_m^{\text{fw}}} - \bar{x}_{v_n^{\text{bw}}}| \leq \delta, \quad (6)$$

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 v^{bw} of ψ^{bw} identified in S1, the reconstruction process involves simulating a hybrid arc, denoted ϕ^{r} , such that it starts from the final state of ϕ^{fw} , flows when $v^{\text{bw}'}$ flows, jumps when $v^{\text{bw}'}$ jumps, and applies the input $(t, j) \mapsto v^{\text{bw}'}(t, j)$, where $v^{\text{bw}'}$ denotes the reversal of v^{bw} . We generate ϕ^{r} via the following hybrid system, denoted $\mathcal{H}_{v^{\text{bw}'}}$, with state $x \in \mathbb{R}^n$ and dynamics:

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

where

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

We require that the reconstruction result, given by the hybrid arc ϕ^{r} , satisfies the following conditions:

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

Notice that the final state of the reconstructed motion plan ϕ^{r} converges to $\phi^{\text{bw}}(0, 0) \in X_f$ as the tolerance δ in (6) approaches zero. Furthermore, if $\phi^{\text{bw}}(0, 0)$ is not on the boundary of X_f , then there is a tolerance ensuring that ϕ^{r} ends in X_f .

5.3 Connecting Two Search Trees via a Jump

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:

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

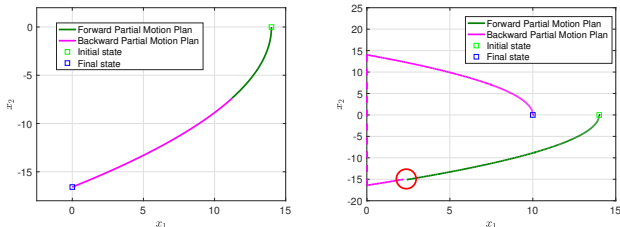
At times, this constrained equation can be solved analytically (e.g., for the system in Example 1) or numerically (e.g., using ideas in Boyd and Vandenberghe (2004)).

Hence, a motion plan is constructed by concatenating ψ^{fw} , a single jump from $\bar{x}_{v^{\text{fw}}}$ to $\bar{x}_{v^{\text{bw}}}$, and ψ^{bw} . This approach constructs a motion plan before detecting overlaps between \mathcal{T}^{fw} and \mathcal{T}^{bw} in S1, improving efficiency and preventing the discontinuity introduced by (6) through a jump.

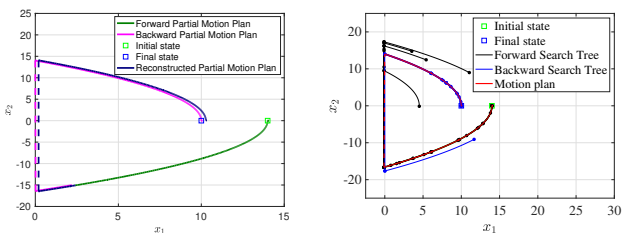
6. SOFTWARE TOOL AND SIMULATION RESULTS

We illustrate the HyRRT-Connect algorithm and this tool¹ in Example 1.

Example 2. (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. 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). However, for most scenarios, such as the settings in Example 1, if we require strict equality without allowing any tolerance as in (6), 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 (6). The simulation results, allowing a tolerance of $\delta = 0.2$, are shown in Figure 1(b). A discontinuity during the flow between the partial motion plans is observed, as depicted in the red circle in Figure 1(b). This discontinuity is addressed through the reconstruction process and a deviation between the endpoint of the reconstructed motion plan and the final state set is also observed in Figure 1(c).



(a) Precise connection during the flow is achieved. (b) A discontinuity during the flow is in red circle.



(c) The backward partial motion plan is reconstructed. (d) HyRRT-Connect

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

We proceed to perform simulation results of HyRRT-Connect showcasing its full functionalities, including the ability to connect partial motion plans via jumps, and results are shown in Figure 1(d). 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. 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.

7. CONCLUSION

We present HyRRT-Connect, a bidirectional algorithm to solve motion planning problems for hybrid systems. It includes a backward-in-time hybrid system formulation, validated by reversing and concatenating functions on the hybrid time domain. To tackle discontinuities during the flow, we introduce a reconstruction process. The computational improvement is exemplified in an actuated bouncing ball example.

REFERENCES

- Boyd, S.P. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.
- Branicky, M.S., Curtiss, M.M., Levine, J.A., and Morgan, S.B. (2003). Rrts for nonlinear, discrete, and hybrid planning and control. In *42nd IEEE International Conference on Decision and Control*, volume 1, 657–663.
- Donzé, A. (2010). Breach, a toolbox for verification and parameter synthesis of hybrid systems. In *Computer Aided Verification: 22nd International Conference*.
- 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 *56th IEEE Conference on Decision and Control*, 1517–1522.
- Kuffner, J.J. and LaValle, S.M. (2000). RRT-connect: An efficient approach to single-query path planning. In *International Conference on Robotics and Automation*.
- 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.
- Nghiem, T., Sankaranarayanan, S., Fainekos, G., Ivancić, F., Gupta, A., and Pappas, G.J. (2010). Monte-carlo techniques for falsification of temporal properties of nonlinear hybrid systems. In *Proceedings of the 13th HSCC*.
- Sanfelice, R.G. (2021). *Hybrid Feedback Control*. Princeton University Press.
- Wang, N. and Sanfelice, R.G. (2022). A rapidly-exploring random trees motion planning algorithm for hybrid dynamical systems. In *IEEE 61st Conference on Decision and Control*, 2626–2631.
- Wang, N. and Sanfelice, R.G. (2023). HySST: An asymptotically near-optimal motion planning algorithm for hybrid systems. In *62nd IEEE Conference on Decision and Control*, 2865–2870.
- Wu, A., Sadraddini, S., and Tedrake, R. (2020). R3T: Rapidly-exploring random reachable set tree for optimal kinodynamic planning of nonlinear hybrid systems. In *International Conference on Robotics and Automation*.

¹ Code at <https://github.com/HybridSystemsLab/HyRRTConnect.git>