Hybrid RRT: Motion Planning for Hybrid Dynamical Systems

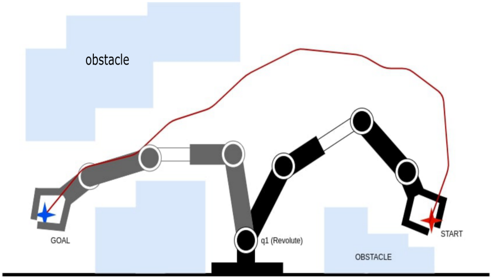
Nan Wang and Ricardo Sanfelice

Department of Computer Engineering
Hybrid System Laboratory
University of California, Santa Cruz, USA

June 3, 2022

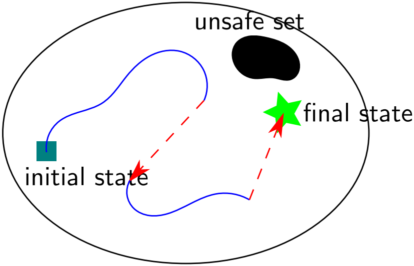
Hybrid Systems Laboratory

1. Motivation: robotics motion planning



1. Motivation: robots with hybrid dynamics

The motion planning problem for hybrid systems is to find a trajectory of states and inputs that starts from the initial state (set), ends within the final state (set), and satisfies both continuous and discrete dynamics and **safety** criterion.



Outline

1. **Motivation**
2. **Preliminaries on Hybrid System Modeling**
3. **Problem Formulation**
4. **HyRRT Algorithm**
   - Algorithm description
   - Probabilistic completeness
   - Simulation results

2. Preliminaries on Hybrid System Modeling

A hybrid system $\mathcal{H}$ with inputs is modeled by a system of differential and difference equations as

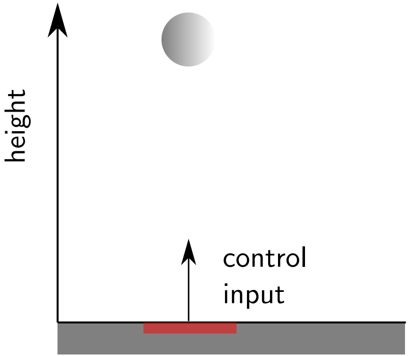
$$\mathcal{H} : \begin{cases} \dot{x} = f(x,u) & (x,u) \in C \\ x^+ = g(x,u) & (x,u) \in D \end{cases} \quad (1)$$

where $x \in \mathbb{R}^n$ is state, $u \in \mathbb{R}^m$ is input,

- $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is the flow map;
- $C \subset \mathbb{R}^n \times \mathbb{R}^m$ is the flow set;
- $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is the jump map;
- $D \subset \mathbb{R}^n \times \mathbb{R}^m$ is the jump set.

2. Preliminaries on Hybrid System Modeling

A motivating example: A classic example of hybrid system is bouncing ball system. Consider a ball bouncing on a fixed horizontal surface. The surface is capable of affecting the velocity of the ball after the impact through control actions.



2. Preliminaries on Hybrid System Modeling

A motivating example: A classic example of hybrid system is bouncing ball system. Consider a ball bouncing on a fixed horizontal surface. The surface is capable of affecting the velocity of the ball after the impact through control actions.

- x_1 = height, x_2 = velocity
- γ = the gravity constant, λ = coefficient of restitution

$$\mathcal{H} : \begin{cases} \dot{x} = \begin{bmatrix} x_2 \\ -\gamma \end{bmatrix} =: f(x, u) & \forall (x, u) \in C \\ x^+ = \begin{bmatrix} x_1 \\ -\lambda x_2 + u \end{bmatrix} =: g(x, u) & \forall (x, u) \in D \end{cases} \tag{2}$$

where

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

2. Preliminaries on Hybrid System Modeling

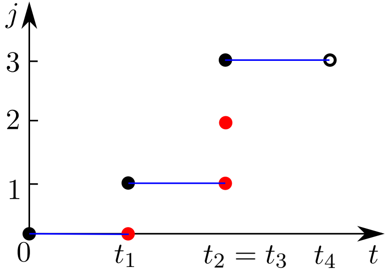
Hybrid time domain: The solutions and inputs to a hybrid system $\mathcal{H}$ are parameterized by (t, j) where

- $t \in \mathbb{R}_{\geq 0}$ denotes the normal time variable
- $j \in \mathbb{N}$ denotes the number of jumps.

Hybrid time domains:

$$\operatorname{dom}\psi := ([0, t_1]) \times \{0\} \cup ([t_1, t_2]) \times \{1\} \cup \cdots \cup ([t_j, t_{j+1}]) \times \{j\} \cup \cdots$$

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



2. Preliminaries on Hybrid System Modeling

Definition 2.1 (Solution pair to a hybrid system (**altin2019hybrid**))

Given a pair of functions $\phi : \operatorname{dom}\phi \to \mathbb{R}^n$ *and* $u : \operatorname{dom} u \to \mathbb{R}^m$ *defined on hybrid time domains,* (ϕ, u) *is a solution pair to hybrid system* $\mathcal{H} = (C, f, D, g)$ *if:*

1. *During flows,*

$$(\phi(t,j), u(t,j)) \in C \quad \dot{\phi}(t,j) = f(\phi(t,j), u(t,j))$$

2. Preliminaries on Hybrid System Modeling

Definition 2.1 (Solution pair to a hybrid system (**altin2019hybrid**))

Given a pair of functions $\phi : \operatorname{dom}\phi \to \mathbb{R}^n$ *and* $u : \operatorname{dom} u \to \mathbb{R}^m$ *defined on hybrid time domains,* (ϕ, u) *is a solution pair to hybrid system* $\mathcal{H} = (C, f, D, g)$ *if:*

1. *During flows,*

$$(\phi(t,j), u(t,j)) \in C \quad \dot{\phi}(t,j) = f(\phi(t,j), u(t,j))$$

2. *At jumps,*

$$(\phi(t,j), u(t,j)) \in D \quad \phi(t,j+1) = g(\phi(t,j), u(t,j)).$$

3. Problem Formulation

Problem 1 (Motion planning problem for hybrid systems)

Given

1. *a hybrid system* $\mathcal{H} = (C, f, D, g)$ *with input* $u \in \mathbb{R}^m$, *state* $x \in \mathbb{R}^n$;

find a pair $(\phi, u) : \mathrm{dom}(\phi, u) \to \mathbb{R}^n \times \mathbb{R}^m$, *namely, a motion plan, such that:*

1. (ϕ, u) *is a solution pair to* $\mathcal{H}$;

3. Problem Formulation

Problem 1 (Motion planning problem for hybrid systems)

Given

1. *a hybrid system* $\mathcal{H} = (C, f, D, g)$ *with input* $u \in \mathbb{R}^m$, *state* $x \in \mathbb{R}^n$;
2. *initial state set* $X_0 \subset \mathbb{R}^n$;

find a pair $(\phi, u) : \mathrm{dom}(\phi, u) \to \mathbb{R}^n \times \mathbb{R}^m$ *, namely, a motion plan, such that:*

1. (ϕ, u) *is a solution pair to* $\mathcal{H}$;
2. $\phi(0, 0) \in X_0$;

3. Problem Formulation

Problem 1 (Motion planning problem for hybrid systems)

Given

1. *a hybrid system* $\mathcal{H} = (C, f, D, g)$ *with input* $u \in \mathbb{R}^m$, *state* $x \in \mathbb{R}^n$;
2. *initial state set* $X_0 \subset \mathbb{R}^n$;
3. *final state set* $X_f \subset \mathbb{R}^n$;

find a pair $(\phi, u) : \mathrm{dom}(\phi, u) \to \mathbb{R}^n \times \mathbb{R}^m$ *, namely, a motion plan, such that:*

1. (ϕ, u) *is a solution pair to* $\mathcal{H}$;
2. $\phi(0, 0) \in X_0$;
3. *there exists* $(T, J) \in \mathrm{dom}(\phi, u)$ *such that* $\phi(T, J) \in X_f$;

3. Problem Formulation

Problem 1 (Motion planning problem for hybrid systems)

Given

1. *a hybrid system* $\mathcal{H} = (C, f, D, g)$ *with input* $u \in \mathbb{R}^m$, *state* $x \in \mathbb{R}^n$;
2. *initial state set* $X_0 \subset \mathbb{R}^n$;
3. *final state set* $X_f \subset \mathbb{R}^n$;
4. *unsafe set* $\mathbf{X_u} \subset \mathbb{R}^n \times \mathbb{R}^m$;

find a pair $(\phi, u) : \mathrm{dom}(\phi, u) \to \mathbb{R}^n \times \mathbb{R}^m$, *namely, a motion plan, such that:*

1. (ϕ, u) *is a solution pair to* $\mathcal{H}$;
2. $\phi(0, 0) \in X_0$;
3. *there exists* $(T, J) \in \mathrm{dom}(\phi, u)$ *such that* $\phi(T, J) \in X_f$;
4. $(\phi(t, j), u(t, j)) \notin \mathbf{X_u}$ *for each* $(t, j) \in \mathrm{dom}(\phi, u)$, $t + j \leq T + J$.

4. HyRRT Algorithm

4.1 Algorithm description

Algorithm 1 HyRRT algorithm

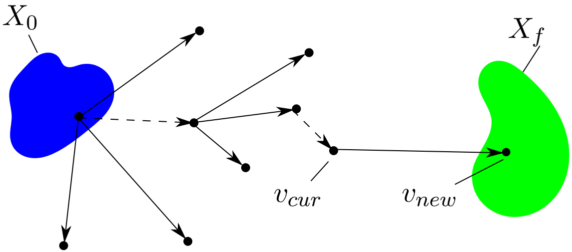
Input: $X_0, X_f, X_u, \mathcal{H} = (C, f, D, g), \mathcal{U} = (\mathcal{U}_C, \mathcal{U}_D), p_n \in (0, 1)$

$\mathcal{T}.init(X_0)$
for $i = 1$ to k **do**
    randomly select a real number n from $[0, 1]$
    if $n \leq p_n$ **then**
        $x_{rand} \leftarrow random\_state(\overline{C'})$
        $extended \leftarrow extend(\mathcal{T}, x_{rand}, \mathcal{U}, \mathcal{H}, X_u, flow)$
    else
        $x_{rand} \leftarrow random\_state(D')$
        $extended \leftarrow extend(\mathcal{T}, x_{rand}, \mathcal{U}, \mathcal{H}, X_u, jump)$
    end if
    if $extended == 1 \& check\_solution(\mathcal{T}, X_0, X_f, C, \psi_{sol}) == 1$ **then**
        return ψ_{sol}
    end if
end for

4. HyRRT Algorithm

4.1 Algorithm description

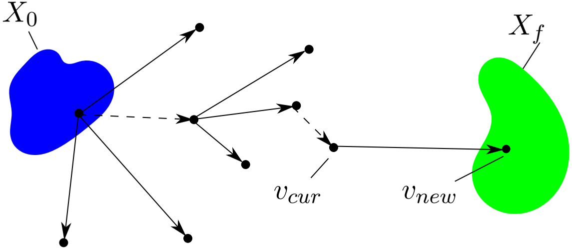
The propagation results:

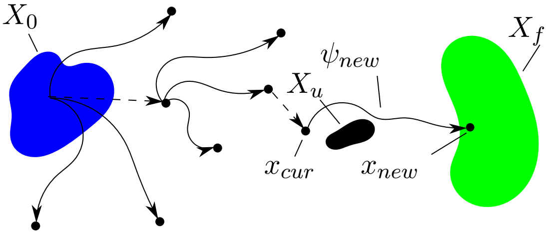


4. HyRRT Algorithm

4.1 Algorithm description

The propagation results:





4. HyRRT Algorithm

4.2 Probabilistic completeness

Definition 4.1

*(Probabilistic completeness (***kleinbort2018probabilistic***)) A sampling-based algorithm is said to be probabilistically complete if the probability of failing to find a solution is converging to* 0*, as the number of samples approaches to infinity.*

4. HyRRT Algorithm

4.2 Probabilistic completeness

Definition 4.1

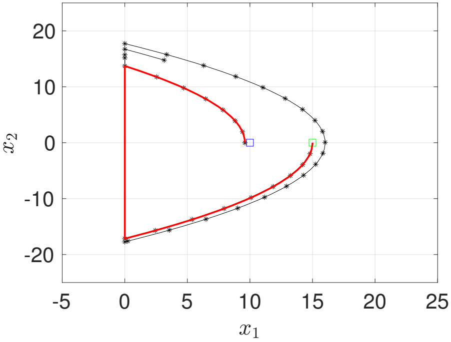
*(Probabilistic completeness (**kleinbort2018probabilistic**)) A sampling-based algorithm is said to be probabilistically complete if the probability of failing to find a solution is converging to 0, as the number of samples approaches to infinity.*

Theorem 1

The proposed HyRRT is probabilistically complete for any given motion planning problem for hybrid systems formulated as in Problem 1.

4. HyRRT Algorithm

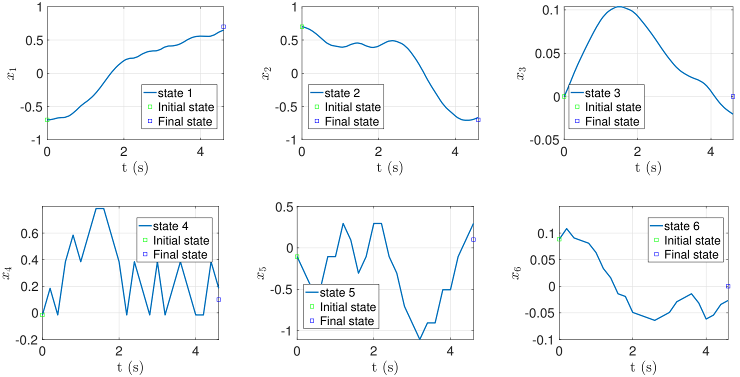
4.3 Simulation results: Algorithm 1 leads to a HyRRT software tool[1] to solve the motion planning problems for hybrid systems. The simulation is implemented in MATLAB software and processed by a 2.2 GHz Intel Core i7 processor. The simulation takes 0.34 seconds to finish.



[1] Code at https://github.com/HybridSystemsLab/hybridRRT.

4. HyRRT Algorithm

4.3 Simulation results: The simulation is implemented in MATLAB software and processed by a 3.5 GHz Intel Core i5 processor. The simulation takes 57.6 seconds to finish.



4. HyRRT Algorithm

4.3 Simulation results: The simulation is implemented in MATLAB software and processed by a 3.5 GHz Intel Core i5 processor. The simulation takes 57.6 seconds to finish.

Table: Computation Costs of HyRRT and FBP in the Biped Example .

	Time Consumption (seconds)	Vertices
HyRRT	57.6	2357
FBP	1608.2	3796

Acknowledgement

This research has been partially supported by the National Science Foundation under Grant no. ECS-1710621, Grant no. CNS-1544396, and Grant no. CNS-2039054, by the Air Force Office of Scientific Research under Grant no. FA9550-19-1-0053, Grant no. FA9550-19-1-0169, and Grant no. FA9550-20-1-0238, and by the Army Research Office under Grant no. W911NF-20-1-0253.

Thank you for your attention. Any questions?