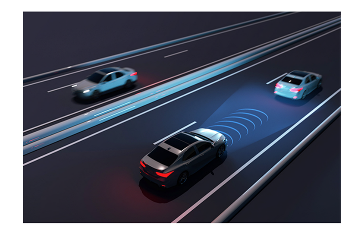
Motion Planning for Hybrid Dynamical Systems

Nan Wang and Ricardo G. Sanfelice

Abstract

Hybrid dynamical systems have state variables that evolve continuously and, at times, exhibit jumps. Standard motion planning algorithms are solely for purely continuous-time [1][2][3] or purely discrete-time [4] models and, hence, do not apply to hybrid systems. One of the main challenges is that the jump times are not known in advance and need to be determined by the planner. Considering the incompatibility between standard motion planning methods and hybrid systems, the goal of this research is to develop a general motion planning algorithm for hybrid systems. The objective is for the planner to produce a motion plan for states and inputs connecting initial and target state sets, while satisfying given static and dynamic constraints. This poster outlines results to date, including bouncing ball systems and actuated point mass systems. The effectiveness of the proposed algorithm is illustrated by two examples.

Walking robots (UMich)



Self-driving vehicles (Samsung)

Robotics manipulator (da Vinci)

I. Preliminaries on Hybrid Systems

Hybrid System Model

A hybrid system $\mathcal{H}$ can be written as

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

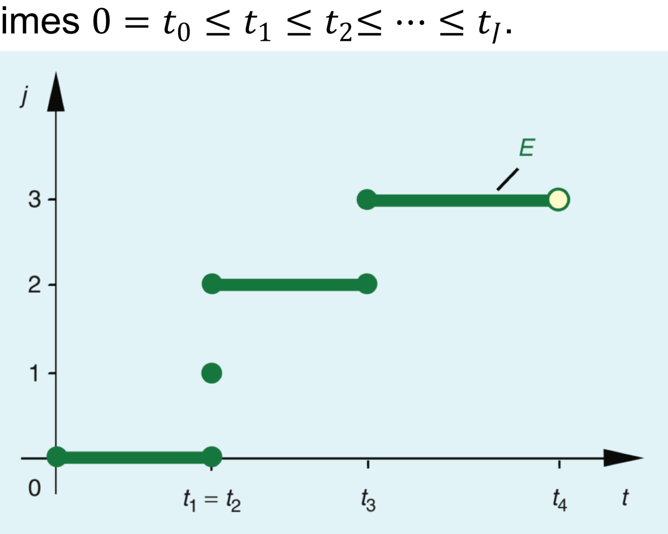
where C, f, D and g represent the flow set, the flow map, the jump set, and the jump map, respectively. The state and input of this system are denoted by x and u, respectively.

Solution to Hybrid System

A solution to the hybrid system $\mathcal{H}$ is given by a hybrid arc x satisfying the dynamics of $\mathcal{H}$. A hybrid arc x is a function on a hybrid time domain that, for each $j \in N$, $t \mapsto x(t,j)$ is absolutely continuous on the interval $I := \{t : (t,j) \in \text{dom } x)\}$.

Hybrid Time Domain

Following [5], besides the usual time variable $t \in R_{\geq 0}$, we consider the number of jumps, $j \in N := \{0, 1, 2, \ldots\}$, as an independent variable. Thus, hybrid time is parameterized by (t,j). The domain of a solution to $\mathcal{H}$ is given by a hybrid time domain. A hybrid time domain defined as a subset E of $R_{\geq 0} \times N$ that, for each $(T,J) \in E$, $E \cap ([0,T] \times \{0,1,\ldots,J\})$ can be written as $\cup_{j=0}^{J-1} ([t_j, t_{j+1}], j)$ for some finite sequence of times $0 = t_0 \leq t_1 \leq t_2 \leq \cdots \leq t_J$.

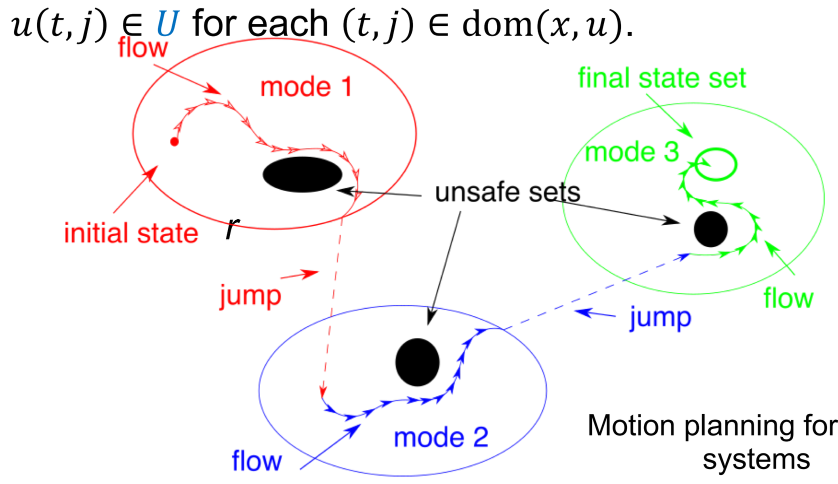


Hybrid time domain

II. Problem Statement

Given hybrid system $\mathcal{H}$ with input $u \in \mathbb{R}^m$, input set $U \subset \mathbb{R}^m$, state $x \in \mathbb{R}^n$, unsafe set $X_u \subseteq \mathbb{R}^n$, final state set $X_f \subseteq \mathbb{R}^n$ and initial state set $X_0 \subseteq \mathbb{R}^n$, for each $x_0 \in X_0$, find (x,u): dom $(x,u) \longmapsto \mathbb{R}^n \times \mathbb{R}^m$ such that the following hold:

- $x(0,0) = x_0$.
- (x,u) is a solution to $\mathcal{H}$.
- $\exists (T,J) \in \text{dom}(x,u) : x(T,J) \in X_f$.
- $x(t,j) \notin X_u$ for each $(t,j) \in \text{dom}(x,u)$
- $u(t,j) \in U$ for each $(t,j) \in \text{dom}(x,u)$.



Motion planning for hybrid systems

IV. A General Motion Planning Algorithm for Hybrid Systems

Algorithm 1 Motion Planning Algorithm for Hybrid Systems

Input: Initial state x_0, final state set X_f, unsafe set X_u, input set U, hybrid system $\mathcal{H}$ and its backward system $\mathcal{H}^{\text{bw}}$

Output: state and input (x, u)

1: Set i = 1, $X_0^i = \{x_0\}$.
2: Propagate backward in hybrid time from X_f by $u \in U$ using $\mathcal{H}^{\text{bw}}$ until the state reaches D^{bw} and compute the set T of all the potential states in D^{bw}.
3: **while** $X_0^i \cap T \neq \emptyset$ **do**
4: Propagate forward in hybrid time from X_0^i by $u \in U$ using $\mathcal{H}$ until the state reaches D and compute set V_i of all the potential states in D.
5: Propagate forward in hybrid time from V_i by $u \in U$ using $\mathcal{H}$ until the state reaches C and compute set Q_i of all the potential states in C.
6: $X_0^{i+1} = Q_i$, $i = i + 1$,
7: **end while**
8: Pick $x_p \in Q_i \cap T$, propagate forward from x_p to X_f and backward from x_0 to x_p. Concatenate the solutions and return.

Backward-in-time Hybrid System

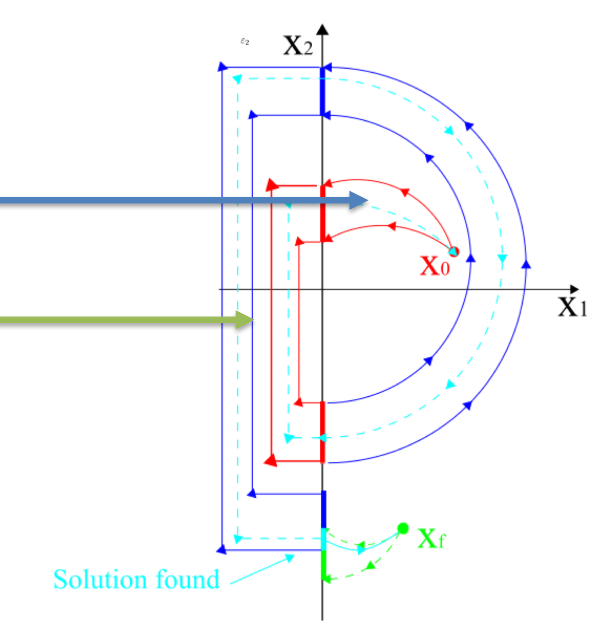
Given the forward-in-time hybrid system $\mathcal{H}$, the backward-in-time hybrid system $\mathcal{H}^{\text{bw}}$ is given by:

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

where C and f are the flow set and the flow map in the $\mathcal{H}$, and g^{bw} and D^{bw} are the backward versions of jump map g and jump set D.

Motion Planning Methodology

- Forward propagation of hybrid motion from initial set and backward propagation from target sets.
- Iteratively propagate the states forward and backward, during flow and jump, until overlap is found. If none is found, report infeasibility of planning.
- Compute motion plan by connecting intermediate forward and backward plans.



To implement the propagations: jump

- For each initial state $x_0 \in X_0 \subset C$ and all the possible input $u \in U$, compute t_D and $x(t_D, 0)$ such that $t \mapsto x(t,0)$ is a solution to $\mathcal{H}$. At $t = t_D$, $x(t_D, 0) \in D$. Compute the set A of all possible $x(t_D, 0)$, $A \subset D$.
- For each initial state $x_0 \in X_0 \subset D$ and all the possible input $u \in U$, compute j_C and $x(0, j_C)$ such that $j \mapsto x(0,j)$ is a solution to $\mathcal{H}$. At $j = j_C$, $x(0, j_C) \in C$. Compute the set B of all possible $x(0, j_C)$, $B \subset C$.

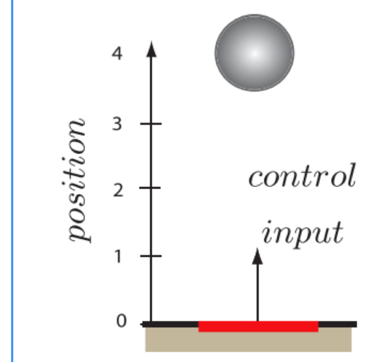
III. Examples

Bouncing Ball System [5]

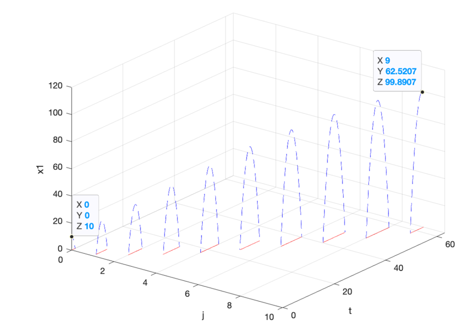
$$\mathcal{H}: \begin{cases} \dot{x} = \begin{bmatrix} x_2 \\ -\gamma \end{bmatrix} & x_1 \geq 0 \\ x^+ = \begin{bmatrix} x_1 \\ -ex_2 + u \end{bmatrix} & x_1 = 0, \; x_2 \leq 0 \end{cases}$$

Simulation setup:
$X_0 = \{(10, 0)\}$,
$X_f = \{(100, 0)\}$,
$U = [0, 10]$,
$\gamma = 9.81$, $e = 0.8$.

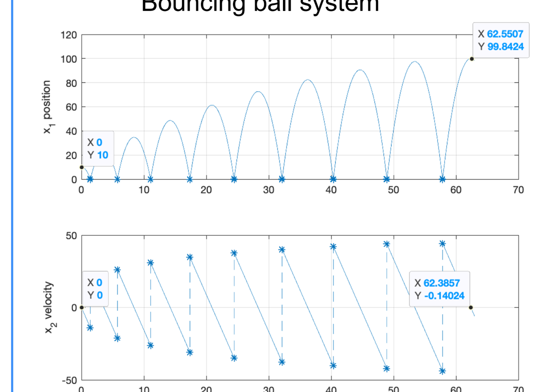
where $x := \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \mathbb{R}^2$, x_1 is the height, x_2 is the velocity of the ball, $u \in U$ is the input, γ is the gravity constant, and $e \in (0,1)$.



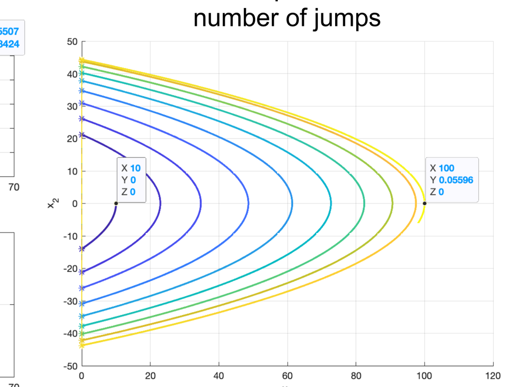
Bouncing ball system



State with respect to time and number of jumps



State with respect to time



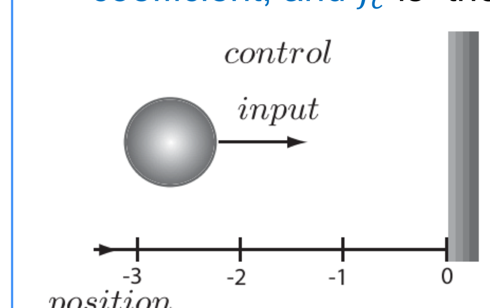
Trajectory of state under motion plans

Actuated Point Mass System [6]

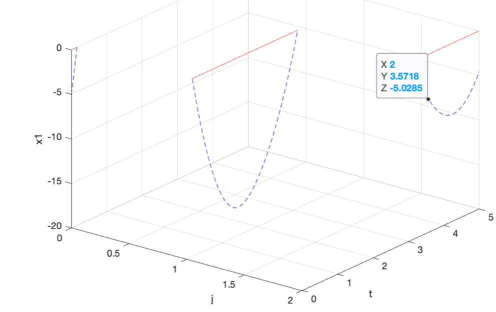
$$\mathcal{H}: \begin{cases} \dot{x} = \begin{bmatrix} x_2 \\ u - f_c(x) \end{bmatrix} & x_1 \leq 0 \text{ or } x_2 \leq \hat{x}_2 \\ x^+ = \begin{bmatrix} x_1 \\ -e_R x_2 \end{bmatrix} & x_1 \geq 0, \; x_2 \geq \hat{x}_2 \end{cases}$$

Simulation setup:
$X_0 = \{(-5, 30)\}$,
$X_f = \{(-5, 10)\}$,
$U = [10, 20]$,
$e_R = 0.8$, $k_c = 1$,
$b_c = 1$, $\hat{x}_2 = 2$

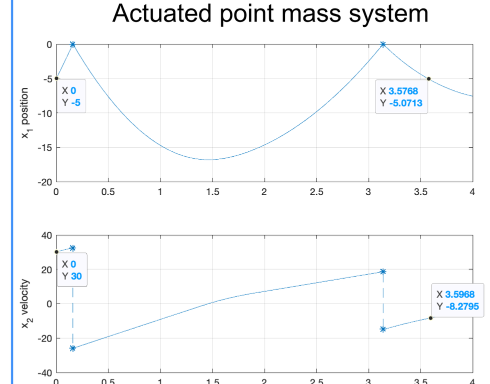
where $x := \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \mathbb{R}^2$, x_1 is the position, x_2 is the velocity of the point mass, $u_c \in U$ denotes the steering input, $e_R \in [0,1]$ represents the uncertain restitution coefficient, and f_c is the contact force [7].



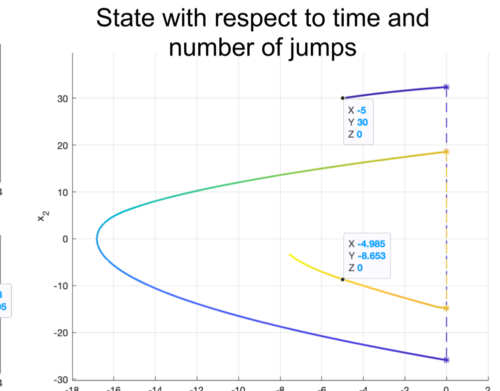
Actuated point mass system



State with respect to time and number of jumps



State with respect to time



Trajectory of state under motion plans

Funding Information

This research has been partially supported by the National Science Foundation under Grant no. ECS-1710621 and Grant no. CNS-1544396, by the Air Force Office of Scientific Research under Grant no. FA9550-16-1-0015, Grant no. FA9550-19-1-0053, and Grant no. FA9550-19-1-0169, and by CITRIS and the Banatao Institute at the University of California. Principal Investigator: Dr. Ricardo G. Sanfelice.

[1] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in Autonomous robot vehicles, 1986, pp. 396–404.
[2] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars, "Prob- abilistic roadmaps for path planning in high-dimensional configuration spaces," IEEE Transactions on Robotics and Automation, vol. 12, no. 4, pp. 566–580, 1996.
[3] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in Autonomous robot vehicles, 1986, pp. 396–404.
[4] M. Likhachev, D. I. Ferguson, G. J. Gordon, A. Stentz, and S. Thrun, "Anytime dynamic A*: An anytime, replanning algorithm," in The International Conference on Automated Planning and Scheduling, 2005, pp. 262–271.
[5] R. Goebel, R. G. Sanfelice, and A. R. Teel, "Hybrid dynamical systems," IEEE Control Systems Magazine, vol. 29, no. 2, pp. 28–93, 2009.
[6] R. Naldi and R. G. Sanfelice, "Passivity-based control for hybrid systems with applications to mechanical systems exhibiting impacts," *Automatica*, vol. 49, no. 5, pp. 1104–1116, 2013.
[7] W. J. Stronge, Impact mechanics. Cambridge university press, 2018.