Nonsmooth Systems

Ricardo Sanfelice
Department Electrical and Computer Engineering
University of California

CoE Review @ Zoom - April 14, 2020

Motivation and Approach

Common features in AFOSR applications:

- Variables changing continuously (e.g., physical quantities) and discretely (e.g., logic variables, resetting timers).
- Abrupt changes in the dynamics (changes in the environment, control decisions, communication events, or failures).

Motivation and Approach

Common features in AFOSR applications:

- Variables changing continuously (e.g., physical quantities) and discretely (e.g., logic variables, resetting timers).
- Abrupt changes in the dynamics (changes in the environment, control decisions, communication events, or failures).

Driving Question:

How can we systematically design such systems featuring **switching** and **intermittency of information** with provable robustness to uncertainties arising in real-world environments?

Motivation and Approach

Common features in AFOSR applications:

- Variables changing continuously (e.g., physical quantities) and discretely (e.g., logic variables, resetting timers).
- Abrupt changes in the dynamics (changes in the environment, control decisions, communication events, or failures).

Driving Question:

How can we systematically design such systems featuring **switching** and **intermittency of information** with provable robustness to uncertainties arising in real-world environments?

Approach:

Motivation and Approach

Common features in AFOSR applications:

- Variables changing continuously (e.g., physical quantities) and discretely (e.g., logic variables, resetting timers).
- Abrupt changes in the dynamics (changes in the environment, control decisions, communication events, or failures).

Driving Question:

How can we systematically design such systems featuring **switching** and **intermittency of information** with provable robustness to uncertainties arising in real-world environments?

Approach:

- Model continuous and discrete behavior using dynamical models that are **hybrid**.

Motivation and Approach

Common features in AFOSR applications:

- Variables changing continuously (e.g., physical quantities) and discretely (e.g., logic variables, resetting timers).
- Abrupt changes in the dynamics (changes in the environment, control decisions, communication events, or failures).

Driving Question:

How can we systematically design such systems featuring **switching** and **intermittency of information** with provable robustness to uncertainties arising in real-world environments?

Approach:

- Model continuous and discrete behavior using dynamical models that are **hybrid**.
- Develop systematic control theoretical tools for **stability**, **invariance**, **safety**, and **temporal logic**, with **robustness**.

Hybrid Models Emerging in CoE Research

Switched Systems

$$\dot{x} = f_{\sigma(t)}(x)$$
$$\sigma(t) \in \{1, 2, \ldots\}$$

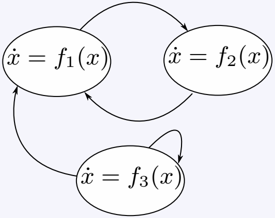
Impulsive Systems

$$\dot{x}(t) = f(x(t))$$
$$x(t^+) = g(x(t)) \quad t \in \{t_1, t_2, \ldots\}$$

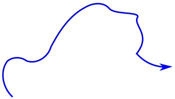
Differential-Algebraic Equations

$$\dot{x} = f(x, w)$$
$$0 = \eta(x, w)$$

Hybrid Automata

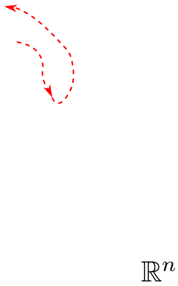


Modeling Hybrid Behavior

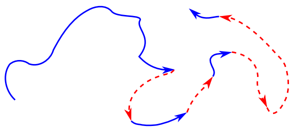


$\mathbb{R}^n$

Modeling Hybrid Behavior

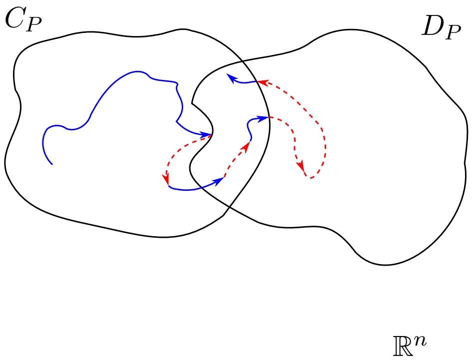


Modeling Hybrid Behavior

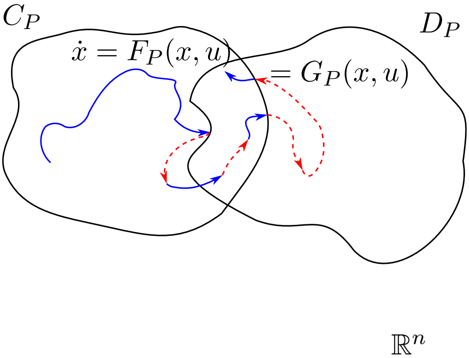


$\mathbb{R}^n$

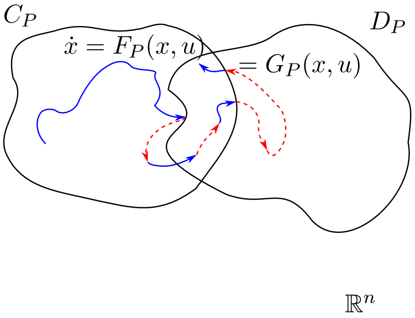
Modeling Hybrid Behavior



Modeling Hybrid Behavior



Hybrid Equations



Hybrid Equations

A hybrid system $\mathcal{H}$ with state $x \in \mathbb{R}^n$, input $u \in \mathbb{R}^m$:

$$\mathcal{H} \begin{cases} \dot{x} = F_P(x,u) & (x,u) \in C_P \\ x^+ = G_P(x,u) & (x,u) \in D_P \end{cases}$$

- C_P is the *flow set*
- F_P is the *flow map*
- D_P is the *jump set*
- G_P is the *jump map*

Solution pairs parametrized by (t,j):

- $t \in [0,\infty)$, time elapsed during flows
- $j \in \{0,1,\dots\}$, number of jumps that have occurred

Hybrid Equations

A hybrid system $\mathcal{H}$ with state $x \in \mathbb{R}^n$, input $u \in \mathbb{R}^m$:

$$\mathcal{H} \begin{cases} \dot{x} \in F_P(x,u) & (x,u) \in C_P \\ x^+ \in G_P(x,u) & (x,u) \in D_P \end{cases}$$

- C_P is the *flow set*
- F_P is the *flow map*
- D_P is the *jump set*
- G_P is the *jump map*

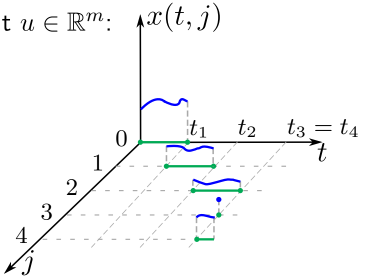
Solution pairs parametrized by (t, j):

- $t \in [0, \infty)$, time elapsed during flows
- $j \in \{0, 1, \dots\}$, number of jumps that have occurred

Hybrid Equations

A hybrid system $\mathcal{H}$ with state $x \in \mathbb{R}^n$, input $u \in \mathbb{R}^m$:

$$\mathcal{H} \begin{cases} \dot{x} \in F_P(x,u) \\ x^+ \in G_P(x,u) \end{cases}$$



- C_P is the *flow set*
- F_P is the *flow map*

Solution pairs parametrized by (t,j):

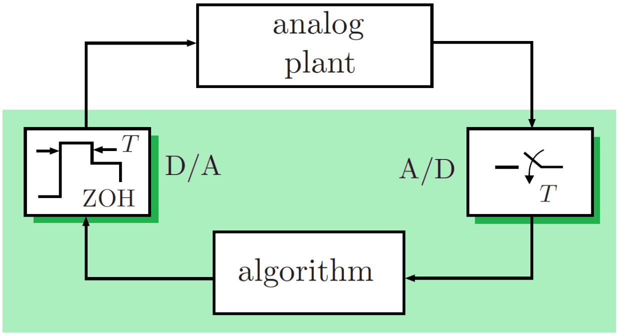
- $t \in [0,\infty)$, time elapsed during
- $j \in \{0,1,\dots\}$, number of jumps that have occurred

Domain of a solution pair $(t,j) \mapsto (x(t,j),u(t,j))$ of the form

$$([0,t_1] \times \{0\}) \cup ([t_1,t_2] \times \{1\}) \cup \dots,$$

where $t_1 \leq t_2 \leq \dots$ are the *jump times*.

Example: Sample-and-Hold Control



Example: Sample-and-Hold Control

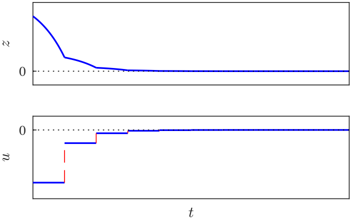
Hybrid dynamics:

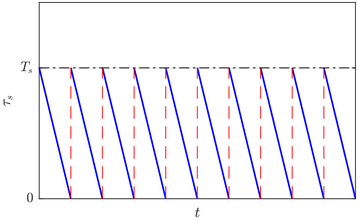
$$\begin{cases} \begin{bmatrix} \dot{z} \\ \dot{\tau}_s \\ \dot{u} \end{bmatrix} = \begin{bmatrix} Az + Bu \\ -1 \\ 0 \end{bmatrix} & \text{if } \tau_s \in [0, T_s] \\ \begin{bmatrix} z^+ \\ \tau_s^+ \\ u^+ \end{bmatrix} = \begin{bmatrix} z \\ T_s \\ Kz \end{bmatrix} & \text{if } \tau_s = 0 \end{cases}$$

Example: Sample-and-Hold Control

Hybrid dynamics:

$$\begin{cases} \begin{bmatrix} \dot{z} \\ \dot{\tau}_s \\ \dot{u} \end{bmatrix} = \begin{bmatrix} Az + Bu \\ -1 \\ 0 \end{bmatrix} & \text{if } \tau_s \in [0, T_s] \\ \begin{bmatrix} z^+ \\ \tau_s^+ \\ u^+ \end{bmatrix} = \begin{bmatrix} z \\ T_s \\ Kz \end{bmatrix} & \text{if } \tau_s = 0 \end{cases}$$





Synchronization Over a Network

Let the dynamics of each node of the network be

$$\dot{z}_i = Az_i + Bu_i$$

Synchronization Over a Network

Let the dynamics of each node of the network be

$$\dot{z}_i = Az_i + Bu_i$$

Goal

Design a **feedback controller** assigning u_i to drive the solutions to *synchronization* using information received only at communication event times.

Synchronization Over a Network

Let the dynamics of each node of the network be

$$\dot{z}_i = Az_i + Bu_i$$

Goal

Design a **feedback controller** assigning u_i to drive the solutions to *synchronization* using information received only at communication event times. Namely, ensure

- $\lim_{t \to \infty} |z_i(t) - z_k(t)| = 0$ for each $i, k \in \mathcal{V}$
- and Lyapunov stability of the set of points z such that

$$z_i = z_k \qquad \forall i, k \in \mathcal{V}$$

Synchronization Over a Network

Let the dynamics of each node of the network be

$$\dot{z}_i = A z_i + B u_i$$

Goal

Design a **feedback controller** assigning u_i to drive the solutions to *synchronization* using information received only at communication event times. Namely, ensure

- $\lim_{t \to \infty} |z_i(t) - z_k(t)| = 0$ for each $i, k \in \mathcal{V}$
- and Lyapunov stability of the set of points z such that

$$z_i = z_k \qquad \forall i, k \in \mathcal{V}$$

Proposed Controller: Controller with state η_i assigns $u_i = \eta_i$

$$\begin{cases} \dot{\eta}_i(t) &= 0 & \text{when } t \notin \{t_\ell\}_{\ell=1}^{\infty} \\ \eta_i^+ &= \dfrac{K_i}{d_i^{in}} \displaystyle\sum_{k \in \mathcal{J}_i} (z_i(t) - z_k(t)) & \text{when } t \in \{t_\ell\}_{\ell=1}^{\infty} \end{cases}$$

Synchronization Over a Network

Let the dynamics of each node of the network be

$$\dot{z}_i = Az_i + Bu_i$$

Goal

Design a **feedback controller** assigning u_i to drive the solutions to *synchronization* using information received only at communication event times. Namely, ensure

- $\lim_{t\to\infty} |z_i(t) - z_k(t)| = 0$ for each $i, k \in \mathcal{V}$
- and Lyapunov stability of the set of points z such that

$$z_i = z_k \qquad \forall i, k \in \mathcal{V}$$

Globally exponentially stabilize the set, denoted $\mathcal{A}_{sync}$, collecting all points such that

$$z_i = z_k$$

that is, render the "diagonal in z" set GES

$i \quad k \in \mathcal{J}_i$

Hybrid Control for Network Synchronization

Results with Sean Phillips

Now at AFRL/RV

Outline of Recent Results

1. **Optimization**
   - High Performance Optimization via Uniting Control
     ACC19, MTNS20 (submitted), + CoE collab. (M. Hale)
   - Model Predictive Control for Hybrid Systems
     ACC 19, CDC 19, CDC 19 Workshop, ACC 20, CDC 20 (submitted), IFAC WC 20 Workshop + Collab. AFRL/RV (Phillips & Petersen)

2. **Tools to Satisfy High-level Specifications**
   - Solution-independent Conditions for Invariance and Finite-time Attractivity *Automatica 19, TAC 19, NAHS 20, HSCC 20, CDC 20 (submitted) + Collab. NASA (Mavridou)*
   - (Necessary and Sufficient) Safety Certificates
     HSCC 19, ACC 19, ACC 20, HSCC 20 + CoE collab. (Dixon)

3. **Hybrid Control**
   - Global Robust Stabilization on Manifolds
     Automatica 19, TAC 19, ACC 19
   - Synchronization over Networks w/ Intermittent Information *Automatica 19, ACC 19, and ACC 20*

4. **Switched systems** (and applications to ES)

Results developed with my PhD students M. Guarro, S. Phillips, Y. Li at UCSC, with Postdocs F. Ferrante (now at Grenoble Alpes) and B. Altin, and R. Goebel (LUC)

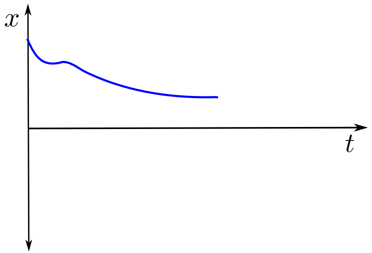
What is Model Predictive Control (MPC)?

Model Predictive Control (MPC)

Predict system behavior, *select* best decision:

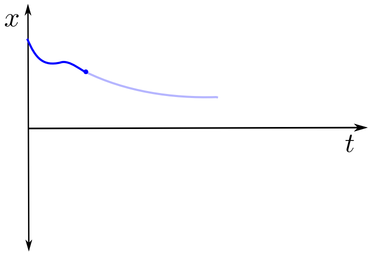
Model Predictive Control (MPC)

Predict system behavior, *select* best decision:



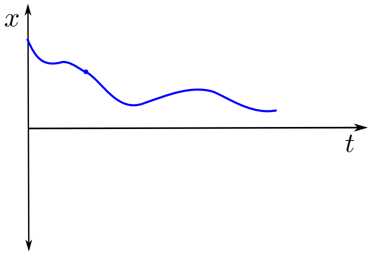
Model Predictive Control (MPC)

Predict system behavior, *select* best decision:



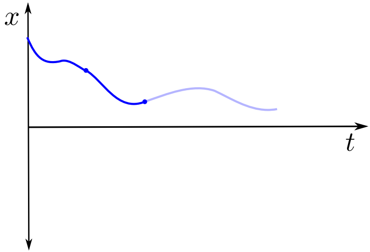
Model Predictive Control (MPC)

Predict system behavior, *select* best decision:



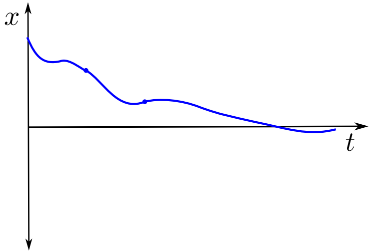
Model Predictive Control (MPC)

Predict system behavior, *select* best decision:



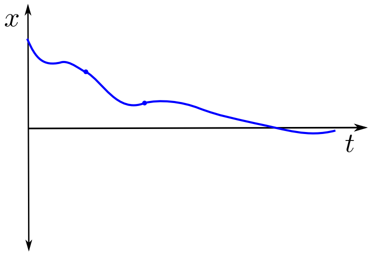
Model Predictive Control (MPC)

Predict system behavior, *select* best decision:



Model Predictive Control (MPC)

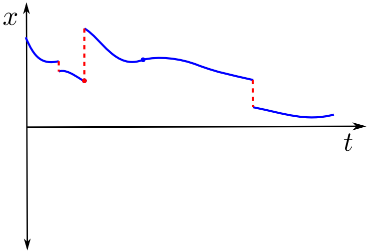
Predict system behavior, *select* best decision:



- Constraint satisfaction
- Optimal performance
- Maximal basin of attraction
- Does not require closed-form controller

Model Predictive Control (MPC)

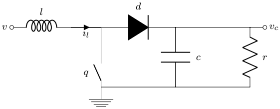
Predict system behavior, *select* best decision:



- Constraint satisfaction
- Optimal performance
- Maximal basin of attraction
- Does not require closed-form controller

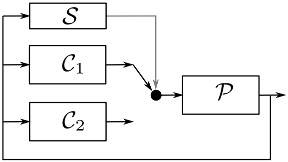
Background

"Hybrid", in the context of MPC:



- Mixed logical dynamical systems
- Continuous- and discrete-valued states
- Discontinuities in the plant dynamics
- Switching or sampled-data implementation

[El-Farra ea 05], [Mayne 14], [Camacho ea 10], [Borrelli ea 17], [Sanfelice 19]



- MPC for switching systems [Müller et al., 2012]
- MPC for linear impulsive systems [Sopasakis et al., 2015]
- Measure-driven for impulsive systems [Pereira et al., 2015]

Contributions

Sufficient conditions for

- recursive feasibility,
- asymptotic stability of closed sets,

Contributions

Sufficient conditions for

- recursive feasibility,
- asymptotic stability of closed sets,

that allow for consecutive *jumps* and *Zeno* solutions, without

Contributions

Sufficient conditions for

- recursive feasibility,
- asymptotic stability of closed sets,

that allow for consecutive *jumps* and *Zeno* solutions, without

- discretizing the continuous-time dynamics,
- partitioning the state into continuous/discrete-valued components.

Contributions

Sufficient conditions for

- recursive feasibility,
- asymptotic stability of closed sets,

that allow for consecutive *jumps* and *Zeno* solutions, without

- discretizing the continuous-time dynamics,
- partitioning the state into continuous/discrete-valued components.

An MPC scheme applicable to various hybrid modeling paradigms, theoretical foundations of a general hybrid MPC framework.

Outline

- **Modeling Hybrid Systems**
- **Outline of Model Predictive Control**
- **Hybrid MPC:** The General Case
  - Hybrid Optimal Control
  - The Prediction Horizon
  - Basic Assumptions and Results
- **Hybrid MPC:** Persistent Flows or Jumps
- **Numerical Solutions**
  - Hybrid System Simulators
  - Hybrid Optimal Control via Discretization
- **Conclusion**

The Cost Functional

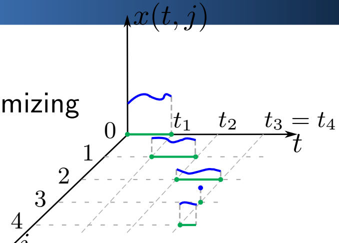
Given an initial condition x_0, find (x, u) minimizing

$$\mathcal{J}(x,u) := \overbrace{\sum_{j=0}^{J} \int_{t_j}^{t_{j+1}} L_c(x(t,j), u(t,j))\, dt}^{\text{Cost-to-flow}}$$

$$+ \underbrace{\sum_{j=0}^{J-1} L_d(x(t_{j+1}, j), u(t_{j+1}, j))}_{\text{Cost-to-jump}} + \underbrace{V(x(T,J))}_{\text{Terminal cost}}$$

The Cost Functional

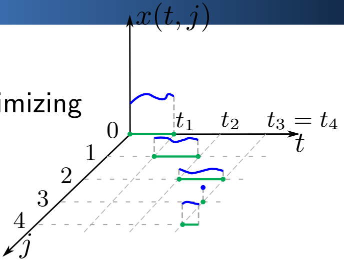
Given an initial condition x_0, find (x,u) minimizing



$$\mathcal{J}(x,u) := \overbrace{\sum_{j=0}^{J} \int_{t_j}^{t_{j+1}} L_c(x(t,j),u(t,j))\,dt}^{\text{Cost-to-flow}}$$

$$+ \underbrace{\sum_{j=0}^{J-1} L_d(x(t_{j+1},j),u(t_{j+1},j))}_{\text{Cost-to-jump}} + \underbrace{V(x(T,J))}_{\text{Terminal cost}}$$

The Cost Functional



Given an initial condition x_0, find (x, u) minimizing

$$\mathcal{J}(x,u) := \underbrace{\sum_{j=0}^{J} \int_{t_j}^{t_{j+1}} L_c(x(t,j), u(t,j))\, dt}_{\text{Cost-to-flow}} + \underbrace{\sum_{j=0}^{J-1} L_d(x(t_{j+1}, j), u(t_{j+1}, j))}_{\text{Cost-to-jump}} + \underbrace{V(x(T,J))}_{\text{Terminal cost}}$$

subject to

- $(T, J) \in \mathcal{T}$, given the *prediction horizon* $\mathcal{T}$,
- $x(T, J) \in X$, given the *terminal constraint set* X.

Hybrid Optimal Control Problem

Problem $(\star)$

Given an initial condition x_0,

$$\begin{aligned} \underset{(x,u)\in\mathcal{S}_{\mathcal{H}_P}(x_0)}{\text{minimize}} \quad & \mathcal{J}(x,u) \\ \text{subject to} \quad & (T,J)\in\mathcal{T} \\ & x(T,J)\in X, \end{aligned}$$

where (T,J) is the terminal time of (x,u).

Hybrid Optimal Control Problem

Problem $(\star)$

Given an initial condition x_0,

$$\begin{aligned} \underset{(x,u)\in\mathcal{S}_{\mathcal{H}_P}(x_0)}{\text{minimize}} \quad & \mathcal{J}(x,u) \\ \text{subject to} \quad & (T,J)\in\mathcal{T} \\ & x(T,J)\in X, \end{aligned}$$

where (T,J) is the terminal time of (x,u).

- The *feasible set* $\mathcal{X}$, set of all x_0 with *feasible* $(x,u)\in\mathcal{S}_{\mathcal{H}_P}(x_0)$.
- The *value function* $\mathcal{J}^*:\mathcal{X}\to\mathbb{R}_{\geq 0}$, defined as

$$\mathcal{J}^*(x_0) := \inf_{\substack{(x,u)\in\mathcal{S}_{\mathcal{H}_P}(x_0)\\ (T,J)\in\mathcal{T}\\ x(T,J)\in X}} \mathcal{J}(x,u) \quad \forall x_0\in\mathcal{X}.$$

Main Elements of Hybrid MPC

Given

1. hybrid system $\mathcal{H}_P$ with data (C_P, F_P, D_P, G_P)
2. *closed* set $\mathcal{A} \subset \mathbb{R}^n$ to stabilize

Main Elements of Hybrid MPC

Given

1. hybrid system $\mathcal{H}_P$ with data (C_P, F_P, D_P, G_P)
2. *closed* set $\mathcal{A} \subset \mathbb{R}^n$ to stabilize

find

1. flow cost L_c (**positive definite**)
2. jump cost L_d (**positive definite**)
3. terminal cost V (**CLF**)
4. terminal constraint set X (**forward invariant**)
5. prediction horizon $\mathcal{T}$ (**“reverse” hybrid time domain**)

Main Elements of Hybrid MPC

Given

1. hybrid system $\mathcal{H}_P$ with data (C_P, F_P, D_P, G_P)
2. *closed* set $\mathcal{A} \subset \mathbb{R}^n$ to stabilize

find

1. flow cost L_c (**positive definite**)
2. jump cost L_d (**positive definite**)
3. terminal cost V (**CLF**)
4. terminal constraint set X (**forward invariant**)

...and ensure $\exists$ a locally stabilizing feedback controllers κ_c and κ_d relating 1.-4.

Roadmap

Closed-Loop Asymptotic Stability

To certify asymptotic stability of $\mathcal{A}$, use the value function

$$\mathcal{J}^*(x_0) := \inf_{\substack{(x,u)\in\mathcal{S}_{\mathcal{H}_P}(x_0) \\ (T,J)\in\mathcal{T} \\ x(T,J)\in X}} \mathcal{J}(x,u) \quad \forall x_0 \in \mathcal{X},$$

as a Lyapunov function on the feasible set $\mathcal{X}$.

Roadmap

Closed-Loop Asymptotic Stability

To certify asymptotic stability of $\mathcal{A}$, use the value function

$$\mathcal{J}^*(x_0) := \inf_{\substack{(x,u)\in\mathcal{S}_{\mathcal{H}_P}(x_0) \\ (T,J)\in\mathcal{T} \\ x(T,J)\in X}} \mathcal{J}(x,u) \quad \forall x_0 \in \mathcal{X},$$

as a Lyapunov function on the feasible set $\mathcal{X}$.

Need to verify

- *recursive feasibility* (start in the feasible set $\mathcal{X}$, stay in $\mathcal{X}$)

Roadmap

Closed-Loop Asymptotic Stability

To certify asymptotic stability of $\mathcal{A}$, use the value function

$$\mathcal{J}^*(x_0) := \inf_{\substack{(x,u)\in\mathcal{S}_{\mathcal{H}_P}(x_0) \\ (T,J)\in\mathcal{T} \\ x(T,J)\in X}} \mathcal{J}(x,u) \quad \forall x_0 \in \mathcal{X},$$

as a Lyapunov function on the feasible set $\mathcal{X}$.

Need to verify

- *recursive feasibility* (start in the feasible set $\mathcal{X}$, stay in $\mathcal{X}$)

Closed-Loop Asymptotic Stability

To certify asymptotic stability of $\mathcal{A}$, use the value function

$$\mathcal{J}^*(x_0) := \inf_{\substack{(x,u)\in\mathcal{S}_{\mathcal{H}_P}(x_0) \\ (T,J)\in\mathcal{T} \\ x(T,J)\in X}} \mathcal{J}(x,u) \quad \forall x_0 \in \mathcal{X},$$

as a Lyapunov function on the feasible set $\mathcal{X}$.

Need to verify

- *recursive feasibility* (start in the feasible set $\mathcal{X}$, stay in $\mathcal{X}$)
- *positive definiteness* $(\alpha_1(|x_0|_{\mathcal{A}}) \leq \mathcal{J}^*(x_0) \leq \alpha_2(|x_0|_{\mathcal{A}}))$

Roadmap

Closed-Loop Asymptotic Stability

To certify asymptotic stability of $\mathcal{A}$, use the value function

$$\mathcal{J}^*(x_0) := \inf_{\substack{(x,u)\in\mathcal{S}_{\mathcal{H}_P}(x_0) \\ (T,J)\in\mathcal{T} \\ x(T,J)\in X}} \mathcal{J}(x,u) \quad \forall x_0 \in \mathcal{X},$$

as a Lyapunov function on the feasible set $\mathcal{X}$.

Need to verify

- *recursive feasibility* (start in the feasible set $\mathcal{X}$, stay in $\mathcal{X}$)
- *positive definiteness* ($\alpha_1(|x_0|_{\mathcal{A}}) \leq \mathcal{J}^*(x_0) \leq \alpha_2(|x_0|_{\mathcal{A}})$)
- *descent along optimal solution pairs* ($\mathcal{J}^*(x(s,i)) < \mathcal{J}^*(x(t,j))$ if $s > t$ or $i > j$)

Example: MPC for the Bouncing Ball

$$
\begin{cases}
(\dot{x}_1, \dot{x}_2) = (x_2, -\gamma) & x_1 \geq 0 \\
(x_1^+, x_2^+) = (0, -\lambda x_2 + u) & x_1 = 0 \text{ and } x_2 \leq 0, \quad u \geq 0
\end{cases}
$$

Control Objective

Reach height h after each impact, i.e., stabilize

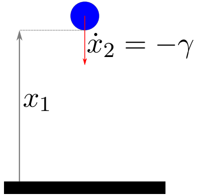
$$\mathcal{A} = \{x : x_1 \geq 0, W(x) = \gamma h\},$$

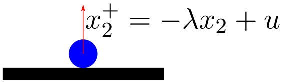
where $W(x) = \gamma x_1 + \frac{x_2^2}{2}$ is the total energy function.

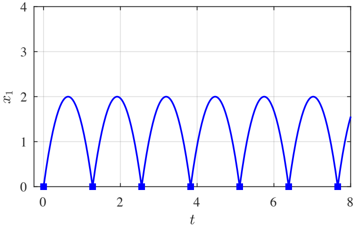
(Cost functions derived from the energy $W(x)$.)

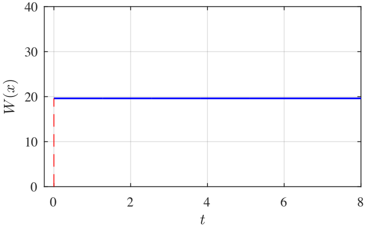
Example: MPC for the Bouncing Ball

$$\begin{cases} (\dot{x}_1, \dot{x}_2) = (x_2, -\gamma) & x_1 \geq 0 \\ (x_1^+, x_2^+) = (0, -\lambda x_2 + u) & x_1 = 0 \text{ and } x_2 \leq 0, \quad u \geq 0 \end{cases}$$



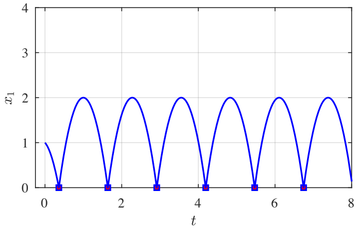


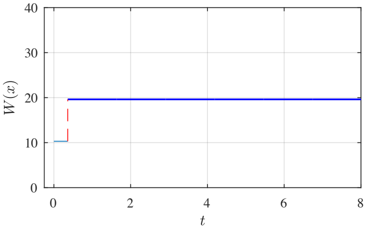




Example: MPC for the Bouncing Ball

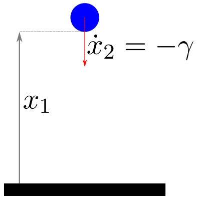
$$\begin{cases} (\dot{x}_1, \dot{x}_2) = (x_2, -\gamma) & x_1 \geq 0 \\ (x_1^+, x_2^+) = (0, -\lambda x_2 + u) & x_1 = 0 \text{ and } x_2 \leq 0, \quad u \geq 0 \end{cases}$$

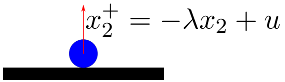


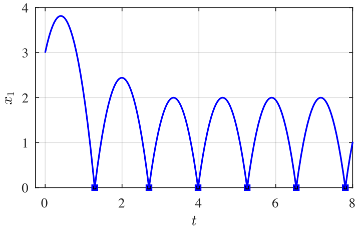


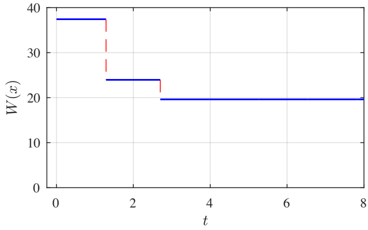
Example: MPC for the Bouncing Ball

$$\begin{cases} (\dot{x}_1, \dot{x}_2) = (x_2, -\gamma) & x_1 \geq 0 \\ (x_1^+, x_2^+) = (0, -\lambda x_2 + u) & x_1 = 0 \text{ and } x_2 \leq 0, \quad u \geq 0 \end{cases}$$









Selecting the Prediction Horizon $\mathcal{T}$

$$\begin{cases} (\dot{x}_1, \dot{x}_2) = (x_2, -\gamma) & x_1 \geq 0 \\ (x_1^+, x_2^+) = (0, -\lambda x_2 + u) & x_1 = 0 \text{ and } x_2 \leq 0, \quad u \geq 0 \end{cases}$$

Suppose $\mathcal{T} = \{(T, J)\}$, stabilize the ball to rest (the origin):

1. Flows not possible from the origin $\implies T = 0$.

Selecting the Prediction Horizon $\mathcal{T}$

$$\begin{cases} (\dot{x}_1, \dot{x}_2) = (x_2, -\gamma) & x_1 \geq 0 \\ (x_1^+, x_2^+) = (0, -\lambda x_2 + u) & x_1 = 0 \text{ and } x_2 \leq 0, \quad u \geq 0 \end{cases}$$

Suppose $\mathcal{T} = \{(T, J)\}$, stabilize the ball to rest (the origin):

1. Flows not possible from the origin $\implies T = 0$.
2. If $T = 0$ and $J > 0$, trajectories that only jump are predicted.

Selecting the Prediction Horizon $\mathcal{T}$

$$\begin{cases} (\dot{x}_1, \dot{x}_2) = (x_2, -\gamma) & x_1 \geq 0 \\ (x_1^+, x_2^+) = (0, -\lambda x_2 + u) & x_1 = 0 \text{ and } x_2 \leq 0, \quad u \geq 0 \end{cases}$$

Suppose $\mathcal{T} = \{(T, J)\}$, stabilize the ball to rest (the origin):

1. Flows not possible from the origin $\implies T = 0$.
2. If $T = 0$ and $J > 0$, trajectories that only jump are predicted.
3. From initial conditions $x_1 > 0$ or $x_2 > 0$, only flows are possible.

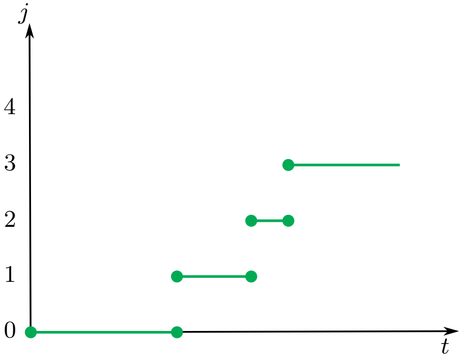
Selecting the Prediction Horizon $\mathcal{T}$

$$\begin{cases} (\dot{x}_1, \dot{x}_2) = (x_2, -\gamma) & x_1 \geq 0 \\ (x_1^+, x_2^+) = (0, -\lambda x_2 + u) & x_1 = 0 \text{ and } x_2 \leq 0, \quad u \geq 0 \end{cases}$$

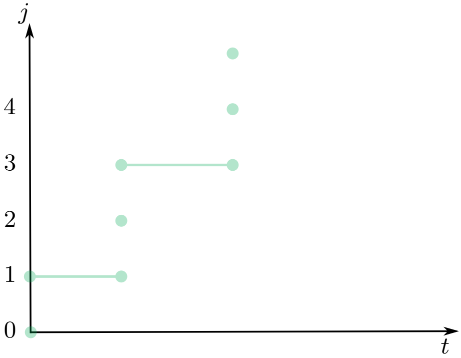
Suppose $\mathcal{T} = \{(T, J)\}$, stabilize the ball to rest (the origin):

1. Flows not possible from the origin $\implies T = 0$.
2. If $T = 0$ and $J > 0$, trajectories that only jump are predicted.
3. From initial conditions $x_1 > 0$ or $x_2 > 0$, only flows are possible.
4. This leads to $(T, J) = (0, 0)$, which means **no prediction**!

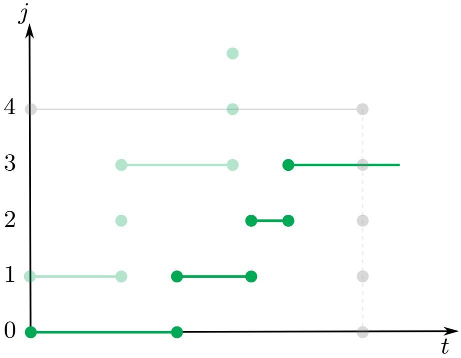
Free End-Time Optimal Control



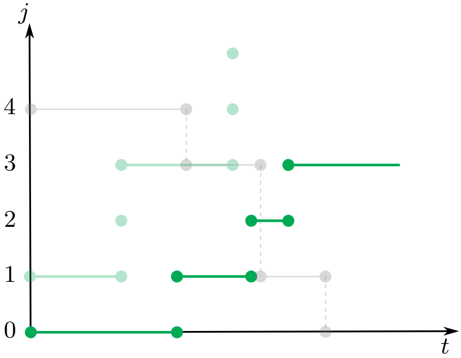
Free End-Time Optimal Control



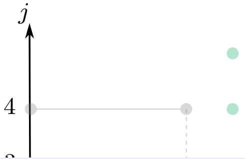
Free End-Time Optimal Control



Free End-Time Optimal Control



Free End-Time Optimal Control



Assumption 1 (prediction horizon)

There exist $t_0 \geq t_1 \geq \dots t_{J+1} = 0$ s.t.

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

where $t_1 > 0$ and $J \geq 1$.

Example Implementation

- "Predict for 4 seconds and/or 4 jumps."

$$\mathcal{T} = \{(T, J) : \max\{T, J\} = 4\}.$$

Example Implementation

- "Predict for 4 seconds and/or 4 jumps."

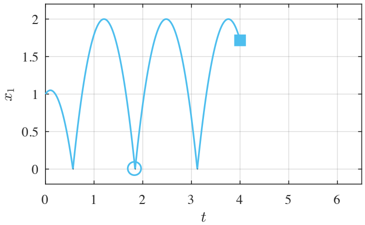
$$\mathcal{T} = \{(T, J) : \max\{T, J\} = 4\}.$$

- "Apply optimal control for 2 seconds and/or 2 jumps."

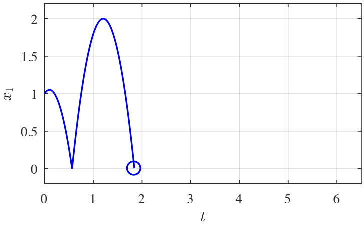
$$\widehat{\mathcal{T}} = \{(T, J) : \max\{T, J\} = 2\}.$$

Example Implementation

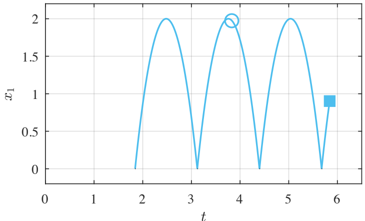
Predict for *4 seconds*



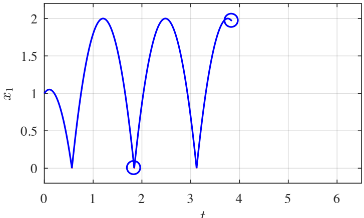
Apply for *2 jumps*



Predict for *4 seconds*



Apply for *2 seconds*



Sample-and-Hold MPC in the Hybrid Setting

Embed a ZOH state η, updated by the input u:

$$\begin{cases} \begin{bmatrix} \dot{z} \\ \dot{\tau}_s \\ \dot{\eta} \end{bmatrix} = \begin{bmatrix} Az + B\eta \\ -1 \\ 0 \end{bmatrix} & \text{if } \tau_s \in [0, T_s] \\ \begin{bmatrix} z^+ \\ \tau_s^+ \\ \eta^+ \end{bmatrix} = \begin{bmatrix} z \\ T_s \\ u \end{bmatrix} & \text{if } \tau_s = 0 \end{cases}$$

Sample-and-Hold MPC in the Hybrid Setting

Embed a ZOH state η, updated by the input u:

$$\begin{cases} \begin{bmatrix} \dot{z} \\ \dot{\tau}_s \\ \dot{\eta} \end{bmatrix} = \begin{bmatrix} Az + B\eta \\ -1 \\ 0 \end{bmatrix} & \text{if } \tau_s \in [0, T_s] \\ \begin{bmatrix} z^+ \\ \tau_s^+ \\ \eta^+ \end{bmatrix} = \begin{bmatrix} z \\ T_s \\ u \end{bmatrix} & \text{if } \tau_s = 0 \end{cases}$$

Minimize

$$\mathcal{J}(x,u) := \sum_{j=0}^{J} \int_{t_j}^{t_{j+1}} \left(x^\top(t,j) Q_c x(t,j) + u^\top(t,j) R_c u(t,j) \right) dt$$

$$+ \sum_{j=0}^{J-1} x^\top(t_{j+1},j) Q_d x(t_{j+1},j) + u^\top(t_{j+1},j) R_d u_{j+1}(t,j)$$

$Q_c = R_c = 0$ recovers discrete-time MPC!).

Basic Conditions for Hybrid MPC

Assumption 2 (basic conditions)

Given the closed set $\mathcal{A} \subset X$,

- there exist class-$\mathcal{K}_\infty$ functions α_c and α_d s.t.

$$L_c(x,u) \geq \alpha_c(|x|_{\mathcal{A}}) \quad \forall (x,u) \in C_P,$$

$$L_d(x,u) \geq \alpha_d(|x|_{\mathcal{A}}) \quad \forall (x,u) \in D_P,$$

- there exists class-$\mathcal{K}_\infty$ functions α_1, α_2 s.t.

$$\alpha_1(|x|_{\mathcal{A}}) \leq V(x) \leq \alpha_2(|x|_{\mathcal{A}}) \quad \forall x \in X,$$

Basic Conditions for Hybrid MPC

Assumption 2 (basic conditions)

Given the closed set $\mathcal{A} \subset X$,

- there exist class-$\mathcal{K}_\infty$ functions α_c and α_d s.t.

$$L_c(x,u) \geq \alpha_c(|x|_\mathcal{A}) \quad \forall (x,u) \in C_P,$$

$$L_d(x,u) \geq \alpha_d(|x|_\mathcal{A}) \quad \forall (x,u) \in D_P,$$

- there exists class-$\mathcal{K}_\infty$ functions α_1, α_2 s.t.

$$\alpha_1(|x|_\mathcal{A}) \leq V(x) \leq \alpha_2(|x|_\mathcal{A}) \quad \forall x \in X,$$

Positive definiteness properties mirroring those in continuous/discrete-time MPC.

Basic Conditions for Hybrid MPC

Assumption 2 (basic conditions)

Given the closed set $\mathcal{A} \subset X$,

- there exist class-$\mathcal{K}_\infty$ functions α_c and α_d s.t.

$$L_c(x,u) \geq \alpha_c(|x|_{\mathcal{A}}) \quad \forall (x,u) \in C_P,$$

$$L_d(x,u) \geq \alpha_d(|x|_{\mathcal{A}}) \quad \forall (x,u) \in D_P,$$

- there exists class-$\mathcal{K}_\infty$ functions α_1, α_2 s.t.

$$\alpha_1(|x|_{\mathcal{A}}) \leq V(x) \leq \alpha_2(|x|_{\mathcal{A}}) \quad \forall x \in X,$$

- there exists $\varepsilon > 0$ s.t. $(\mathcal{A} + \varepsilon\mathbb{B}) \cap \mathrm{Proj}_x(C_P \cup D_P) \subset X$,[1]

[1] Proj_x is the projection onto x component: $\mathrm{Proj}_x(x,u) = x$

Basic Conditions for Hybrid MPC

Assumption 2 (basic conditions)

Given the closed set $\mathcal{A} \subset X$,

- there exist class-$\mathcal{K}_\infty$ functions α_c and α_d s.t.

$$L_c(x,u) \geq \alpha_c(|x|_{\mathcal{A}}) \quad \forall (x,u) \in C_P,$$

$$L_d(x,u) \geq \alpha_d(|x|_{\mathcal{A}}) \quad \forall (x,u) \in D_P,$$

- there exists class-$\mathcal{K}_\infty$ functions α_1, α_2 s.t.

$$\alpha_1(|x|_{\mathcal{A}}) \leq V(x) \leq \alpha_2(|x|_{\mathcal{A}}) \quad \forall x \in X,$$

- there exists $\varepsilon > 0$ s.t. $(\mathcal{A} + \varepsilon\mathbb{B}) \cap \mathrm{Proj}_x(C_P \cup D_P) \subset X$,

$\mathcal{A}$ contained in the “relative” interior (w.r.t. to the state space) of the terminal constraint set (weaker version of the typical MPC assumption)

Basic Conditions for Hybrid MPC

Assumption 2 (basic conditions)

Given the closed set $\mathcal{A} \subset X$,

- there exist class-$\mathcal{K}_\infty$ functions α_c and α_d s.t.

$$L_c(x,u) \geq \alpha_c(|x|_{\mathcal{A}}) \quad \forall (x,u) \in C_P,$$

$$L_d(x,u) \geq \alpha_d(|x|_{\mathcal{A}}) \quad \forall (x,u) \in D_P,$$

- there exists class-$\mathcal{K}_\infty$ functions α_1, α_2 s.t.

$$\alpha_1(|x|_{\mathcal{A}}) \leq V(x) \leq \alpha_2(|x|_{\mathcal{A}}) \quad \forall x \in X,$$

- there exists $\varepsilon > 0$ s.t. $(\mathcal{A} + \varepsilon\mathbb{B}) \cap \mathrm{Proj}_x(C_P \cup D_P) \subset X$,

$\mathcal{A}$ contained in the "relative" interior (w.r.t. to the state space) of the terminal constraint set (weaker version of the typical MPC assumption)

Basic Conditions for Hybrid MPC

Assumption 2 (basic conditions)

- there exists $\delta > 0$ and a continuous σ s.t.

$$|F_P(x,u)| \leq \sigma\left(|x|_{\mathcal{A}}\right) \quad \forall (x,u) \in C_P : |x|_{\mathcal{A}} \leq \delta.$$

Holds with F_P continuous, $\mathcal{A}$ compact, and $C_P = C' \times \mathcal{U}_c$ for compact $\mathcal{U}_c$ (typical continuous-time MPC assumption).

Stabilizing Ingredients for Hybrid MPC

$$\mathcal{H}_P \begin{cases} \dot{x} = F_P(x,u) & (x,u) \in C_P \\ x^+ = G_P(x,u) & (x,u) \in D_P \end{cases}$$

Stabilizing Ingredients for Hybrid MPC

$$\mathcal{H} \begin{cases} \dot{x} = F_P(x, \kappa_c(x)) & (x, \kappa_c(x)) \in C_P \\ x^+ = G_P(x, \kappa_d(x)) & (x, \kappa_d(x)) \in D_P \end{cases}$$

Stabilizing Ingredients for Hybrid MPC

$$\mathcal{H}\begin{cases} \dot{x} = F_P(x, \kappa_c(x)) & (x, \kappa_c(x)) \in C_P \\ x^+ = G_P(x, \kappa_d(x)) & (x, \kappa_d(x)) \in D_P \end{cases}$$

Assumption 3 (CLF)

There exists a feedback $\kappa = (\kappa_c, \kappa_d)$ s.t.

$$\begin{aligned} \overbrace{\langle \nabla V(x), F_P(x, \kappa_c(x))\rangle}^{\dot{V}(x)} &\leq -L_c(x, \kappa_c(x)) \quad \forall x \in X \cap C, \\ \underbrace{V(G_P(x, \kappa_d(x))) - V(x)}_{\Delta V(x)} &\leq -L_d(x, \kappa_d(x)) \quad \forall x \in X \cap D, \end{aligned}$$

and the terminal constraint set X is *forward invariant* for $\mathcal{H}$.

(C, set of x s.t. $(x, \kappa_c(x)) \in C_P$; D...)

Stabilizing Ingredients for Hybrid MPC

$$\mathcal{H} \begin{cases} \dot{x} = F_P(x, \kappa_c(x)) & (x, \kappa_c(x)) \in C_P \\ x^+ = G_P(x, \kappa_d(x)) & (x, \kappa_d(x)) \in D_P \end{cases}$$

Assumption 3 (CLF)

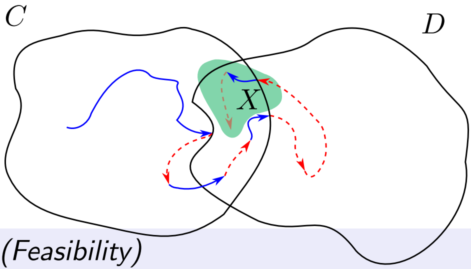
1. Control Lyapunov functions at 1:00 by R.G. Sanfelice; [Sanfelice, TAC 2013]
2. Notions of forward invariance at 1:30 by M. Maghenem; [Chai and Sanfelice, TAC 2019]
3. Barrier functions at 1:30 by M. Maghenem; [Maghenem and Sanfelice, ACC 2019]

(see also Goebel, Sanfelice, and Teel, PUP 2012)

(C, set of x s.t. $(x, \kappa_c(x)) \in C_P$; D...)

Stabilizing Ingredients for Hybrid MPC

$$\mathcal{H}\begin{cases} \dot{x} = F_P(x, \kappa_c(x)) & (x, \kappa_c(x)) \in C_P \\ x^+ = G_P(x, \kappa_d(x)) & (x, \kappa_d(x)) \in D_P \end{cases}$$



Proposition (Feasibility)

The feasible set $\mathcal{X}$ is forward invariant under hybrid MPC, provided the prediction horizon and CLF assumptions hold. Moreover, the terminal constraint set X is contained in $\mathcal{X}$; i.e. $X \subset \mathcal{X}$.

The Value Function $\mathcal{J}^*$

Proposition (Lyapunov Function Candidate)

Under the main assumptions, $\mathcal{J}^*$ satisfies the following:

- *(Continuity) There exists a class-$\mathcal{K}_\infty$ function α_2 s.t.*

$$\mathcal{J}^*(x) \leq V(x) \leq \alpha_2(|x|_{\mathcal{A}}) \quad \forall x \in X \subset \mathcal{X}.$$

- *(Positive Definiteness) There exists a continuous positive definite function γ s.t. $\liminf_{r\to\infty} \gamma(r) > 0$ and*

$$\mathcal{J}^*(x) \geq \gamma(|x|_{\mathcal{A}}) \quad \forall x \in \mathcal{X}.$$

The Value Function $\mathcal{J}^*$

Proposition (Lyapunov Function Candidate)

Under the main assumptions, $\mathcal{J}^*$ satisfies the following:

- *(Continuity) There exists a class-$\mathcal{K}_\infty$ function α_2 s.t.*

$$\mathcal{J}^*(x) \leq V(x) \leq \alpha_2(|x|_{\mathcal{A}}) \quad \forall x \in X \subset \mathcal{X}.$$

- *(Positive Definiteness) There exists a continuous positive definite function γ s.t. $\liminf_{r \to \infty} \gamma(r) > 0$ and*

$$\mathcal{J}^*(x) \geq \gamma(|x|_{\mathcal{A}}) \quad \forall x \in \mathcal{X}.$$

- $\mathcal{J}^*(x) \leq V(x)$ via the CLF inequalities.
- Positive definiteness from $|F_P(x,u)| \leq \sigma\left(|x|_{\mathcal{A}}\right)$.

$\mathcal{J}^*$ as a Stability Certificate

Proposition (Lyapunov Descent)

Under the main assumptions, for any optimal (x, u)*,*

$$\mathcal{J}^*(x(t,j)) \leq \mathcal{J}^*(x(0,0)) - \left(\sum_{i=0}^{j} \int_{s_i}^{s_{i+1}} \alpha_c(|x(s,i)|_{\mathcal{A}})\, ds + \sum_{i=0}^{j-1} \alpha_d(|x(s_{i+1},i)|_{\mathcal{A}}) \right),$$

where $s_1, \ldots, s_j$ *are the jump times and* $s_{j+1} = t$*.*

Asymptotic Stability

Asymptotic stability of $\mathcal{A}$ is certified by the value function $\mathcal{J}^*$, which is a Lyapunov function for the hybrid closed-loop system

Asymptotic Stability

Asymptotic stability of $\mathcal{A}$ is certified by the value function $\mathcal{J}^*$, which is a Lyapunov function for the hybrid closed-loop system

Theorem (Asymptotic Stability Induced by Hybrid MPC)

Under the main assumptions, the following hold:

- *(Existence) There exists $\mu > 0$ s.t. closed-loop solution pairs can be generated from every $x_0 \in \mathrm{Proj}_x(C_P \cup D_P)$ with $|x_0|_{\mathcal{A}} \leq \mu$.*
- *(Stability) For all $\varepsilon > 0$, there exists $\delta > 0$ s.t. for every closed-loop solution pair (x, u),*

$$|x(0,0)|_{\mathcal{A}} \leq \delta \implies |x(t,j)|_{\mathcal{A}} \leq \varepsilon \quad \forall (t,j) \in \mathrm{dom}\, x.$$

- *(Attractivity) Every maximal closed-loop solution pair (x, u) satisfies*

$$\lim_{t+j \to \infty} |x(t,j)|_{\mathcal{A}} = 0.$$

Summary:

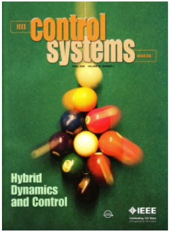
- Overview of Recent Results
- Introduction to hybrid MPC
- Optimal Control Problem
- Outline of Sufficient Conditions

Next steps:

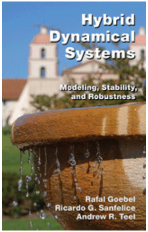
- Estimation
- Reachability
- Approximations
- Robustness

References at hybrid.soe.ucsc.edu

Acknowledgments: Partially supported by NSF, AFOSR, AFRL and by CITRIS and the Banatao Institute at the University of California



IEEE 2009



Princeton U. Press 2012

Princeton U. Press 2020