An Introduction to Hybrid Systems

Aaron D. Ames

Chess Seminar, November 1st 2004

Outline

Outline

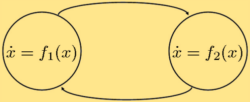
What is a hybrid system?

A hybrid system consists of the following components:

- Continuous component: A collection of dynamical systems.

- Discrete component: A discrete event system (an oriented graph).

- Hybrid system: discrete component interacting with continuous component.



Outline

Dynamical and Control systems

- An Ordinary Differential Equation (ODE) on $\mathbb{R}^n$ given by

$$\dot{x} = f(x)$$

- The *solution*, or *flow*, of an ODE is denoted by $\phi_t(x_0)$, and satisfies
  - $\phi_0(x_0) = x_0$ (identity)
  - $\phi_{t+s}(x_0) = \phi_t \circ \phi_s(x_0)$ (semigroup)
  - $\dot{\phi}_t(x_0) = f(\phi_t(x_0))$ (ODE)
- A control system is a set of equations

$$\begin{aligned} \dot{x} &= f(x,u) \\ y &= g(x,u) \end{aligned}$$

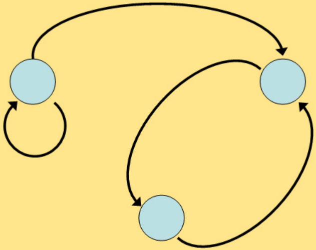
where $y \in \mathbb{R}^n$ is a state of outputs, $u \in \mathcal{U} \subseteq \mathbb{R}^m$ is a state of inputs.

Outline

Oriented Graph

The discrete component of a hybrid system is given by an oriented graph $\Gamma = (Q, E)$ where

- Q is a set of vertices,
- E is a set of oriented edges between these vertices; each edge $e \in E$ has a source $\mathfrak{s}(e) \in Q$ and a target $\mathfrak{t}(e) \in Q$.



Outline

Definition

A hybrid system is a tuple

$$\mathbf{H} = (Q, E, D, G, R, F)$$

where

- $Q = \{1, \ldots, m\} \subset \mathbb{Z}$ is a set of *discrete states* which is a finite subset of the integers.

Definition

A hybrid system is a tuple

$$\mathbf{H} = (Q, E, D, G, R, F)$$

where

- $E \subset Q \times Q$ is a set of *edges* which define relations between the domains. For $e = (i, j) \in E$ denote the source of e by $\mathfrak{s}(e) = i$ and the target of e by $\mathfrak{t}(e) = j$.

Definition

A hybrid system is a tuple

$$\mathbf{H} = (Q, E, D, G, R, F)$$

where

- $D = \{D_i\}_{i \in Q}$ is a set of *domains* where D_i is a subset of $\mathbb{R}^n$.

Definition

A hybrid system is a tuple

$$\mathbf{H} = (Q, E, D, G, R, F)$$

where

- $G = \{G_e\}_{e \in E}$ is a set of *guards*, where $G_e \subseteq D_{\mathfrak{s}(e)}$.

Definition

A hybrid system is a tuple

$$\mathbf{H} = (Q, E, D, G, R, F)$$

where

- $R = \{R_e\}_{e \in E}$ is a set of *reset maps* or *transition maps*; these are continuous maps from $G_e \subseteq D_{\mathfrak{s}(e)}$ to $R_e(G_e) \subseteq D_{t(e)}$.

Definition

A hybrid system is a tuple

$$\mathbf{H} = (Q, E, D, G, R, F)$$

where

- $F = \{f_i\}_{i \in Q}$ is a set of *vector fields* or *ordinary differential equations* (ODEs), such that f_i is Lipschitz on $\mathbb{R}^n$. The solution to the ODE f_i with initial condition $x_0 \in D_i$ is denoted by $\phi_i(t, x_0)$, we assume this solution is defined for all time.

Main Components of the Definition

A hybrid system is a tuple

$$\mathbf{H} = (Q, E, D, G, R, F)$$

where

- Continuous Component: A collection (D, F) of dynamical systems; for each $i \in Q$, (D_i, f_i) is a dynamical system.
- Discrete Component: An oriented graph $\Gamma = (Q, E)$.
- The interaction between the discrete and continuous components is given by the pair (G, R); the guards and resets.

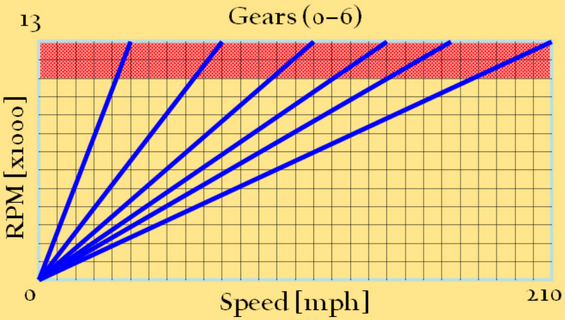
Outline

Problem Statement

Design an automatic transmission for a car, e.g.,

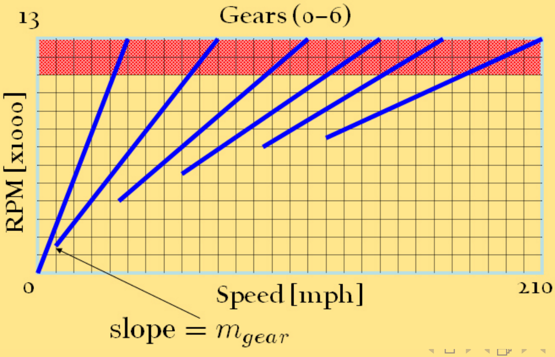
Basic Properties of the Gearbox

- Shifting gears allows higher speeds before damaging engine (a.k.a. redlining)
- However, not all gears function well at low RPM, requiring a certain speed before their use



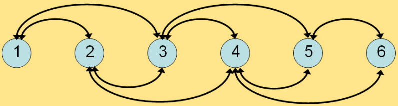
Constrained Gearbox

- Safe zones for each gear
- Limited shifting, due to safe zones
- Requires a smart controller for automatic transmissions: because of switching (gear changing) a hybrid system model is needed.



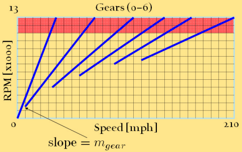
Hybrid System Model of Gearbox

Discrete Component: Given by the oriented graph $\Gamma = (Q, E)$ defined by the diagram



where

- $Q = \{1, 2, 3, 4, 5, 6\}$ is the set of gears
- E are edges defining the transitions that can occur between gears.

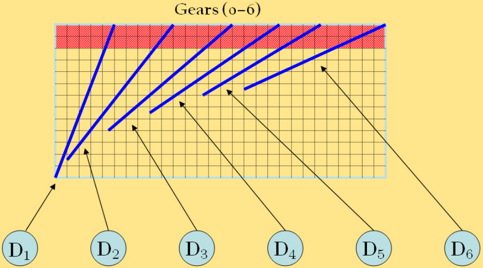


Hybrid System Model of Gearbox

Continuous Component: Given by pairs (D_g, f_g) for $g \in Q$, where

$$\dot{x} = f_g(x, u)$$

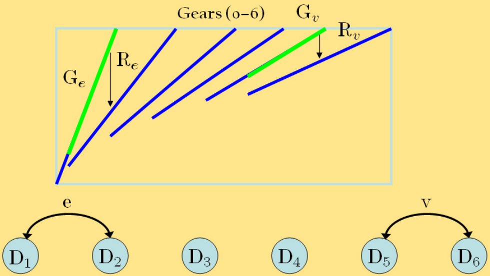
on D_g. Here $x = (\text{RPM}, \text{mph})$ and $u \in [u_{\min}, u_{\max}]$ is an external control (given by the driver). The domains D_g are given by the safety characteristics of the gear box:



Hybrid System Model of Gearbox

Discrete and Continuous Interaction: Given by pairs (G_e, R_e) for $e \in E$, where

- $G_e =$ overlap of $D_{\mathfrak{s}(e)}$ and $D_{\mathfrak{t}(e)}$ in the mph coordinate.
- $R_e =$ translation from $D_{\mathfrak{s}(e)}$ to $D_{\mathfrak{t}(e)}$ in the RPM coordinate.

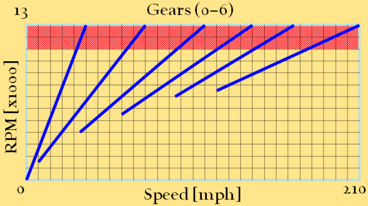


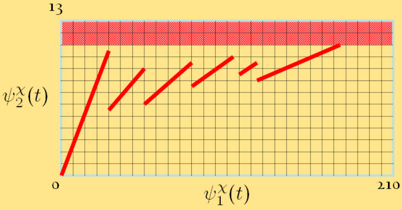
Outline

Outline

Example: Gearbox Execution

Executions describe the behavior of a hybrid system, e.g., for the Gearbox example an execution describes a way of getting to 200mph.





Executions

An execution is a tuple

$$\chi = (\Lambda, \tau, \xi, \eta)$$

where

- $\Lambda = \{0, 1, \ldots\} \subset \mathbb{N}$ is an indexing set. Let $\Lambda^+ = \Lambda \backslash \{0\}$.
- $\tau = \{\tau_i\}_{i \in \Lambda}$ with $\tau_0 = 0 \leq \tau_1 \leq \cdots \leq \tau_j \leq \cdots$ is a *hybrid time sequence* or a sequence of *switching times*.
- $\xi = \{\xi_i\}_{i \in \Lambda}$ with $\xi_i \in \bigcup_{i \in Q} D_i$ is a *sequence of initial conditions*.
- $\eta = \{\eta_i\}_{i \in \Lambda^+}$ with $\eta_i \in E$ is a *hybrid edge sequence*.

Executions

An execution is a tuple

$$\chi = (\Lambda, \tau, \xi, \eta)$$

where

- $\Lambda = \{0, 1, \ldots\} \subset \mathbb{N}$ is an indexing set. Let $\Lambda^+ = \Lambda \backslash \{0\}$.
- $\tau = \{\tau_i\}_{i \in \Lambda}$ with $\tau_0 = 0 \leq \tau_1 \leq \cdots \leq \tau_j \leq \cdots$ is a *hybrid time sequence* or a sequence of *switching times*.
- $\xi = \{\xi_i\}_{i \in \Lambda}$ with $\xi_i \in \bigcup_{i \in Q} D_i$ is a *sequence of initial conditions*.
- $\eta = \{\eta_i\}_{i \in \Lambda^+}$ with $\eta_i \in E$ is a *hybrid edge sequence*.

Let $\mathscr{E}(\mathbf{H})$ be the set of executions of $\mathbf{H}$ and $\mathscr{E}(\mathbf{H}, x_0) \subset \mathscr{E}(\mathbf{H})$ such that for every $\chi \in \mathscr{E}(\mathbf{H}, x_0)$, $\xi_0 = x_0$.

Conditions on Executions

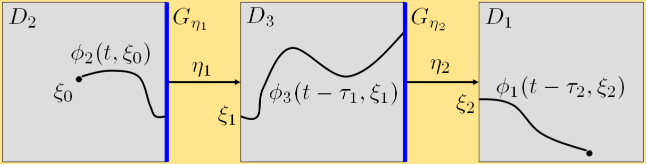
An execution

$$\chi = (\Lambda, \tau, \xi, \eta)$$

must satisfy the conditions:

- If $0 \leq i < |\Lambda| - 1$ and $1 \leq j < |\Lambda^+|$,

$$\begin{aligned} \tau_{i+1} &= \min\{t \geq \tau_i : \phi_{\mathfrak{s}(\eta_{i+1})}(t - \tau_i, \xi_i) \in G_{\eta_{i+1}}\} \\ \mathfrak{s}(\eta_{j+1}) &= \mathfrak{t}(\eta_j) \\ \xi_{i+1} &= R_{\eta_{i+1}}(\phi_{\mathfrak{s}(\eta_{i+1})}(\tau_{i+1} - \tau_i, \xi_i)) \end{aligned}$$



Conditions on Executions

An execution

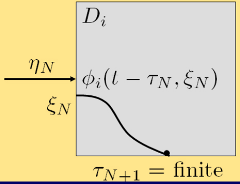
$$\chi = (\Lambda, \tau, \xi, \eta)$$

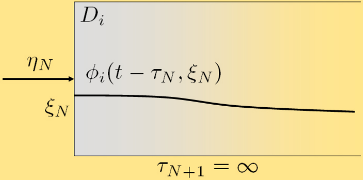
must satisfy the conditions:

- If $|\Lambda| = N+1$ for finite N, then define an additional element τ_{N+1} as follows: if $\phi_{\mathfrak{t}(\eta_N)}(t - \tau_N, \xi_N) \in \partial D_{\mathfrak{t}(\eta_N)}$ for some finite $t \geq \tau_N$ define

$$\tau_{N+1} = \min\{t \geq \tau_N : \phi_{\mathfrak{t}(\eta_N)}(t - \tau_N, \xi_N) \in \partial D_{\mathfrak{t}(\eta_N)}\}$$

  otherwise set $\tau_{N+1} = \infty$.





Notation

Define the following *open-closed* interval for an execution χ as follows

$$[\tau_i, \tau_{i+1}\rangle = \begin{cases} [\tau_i, \tau_{i+1}] & \text{if} \quad 0 \leq i < |\Lambda| \\ [\tau_N, \tau_{N+1}] & \text{if} \quad |\Lambda| = N+1 \text{ for finite N and } \tau_{N+1} \\ [\tau_N, \tau_{N+1}) & \text{if} \quad \tau_{N+1} = \infty \text{ and } |\Lambda| = N+1 \text{ for finite N} \end{cases}$$

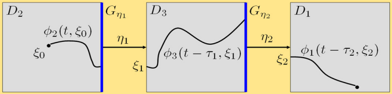
The time domain of an execution χ can be defined as

$$\mathcal{T}^{\chi} = \bigcup_{i \in \Lambda} [\tau_i, \tau_{i+1}\rangle.$$

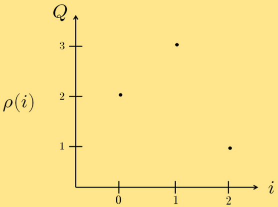
Hybrid Flows

The *discrete evolution* of an execution $\chi \in \mathscr{E}(\mathbf{H})$ is given by

$$\begin{aligned} \rho : \Lambda^{\chi} &\to Q \\ i &\mapsto \begin{cases} \mathfrak{s}(\eta_1) & \text{if} \quad i = 0 \\ \mathfrak{t}(\eta_i) & \text{if} \quad i > 0 \end{cases} \end{aligned}$$



$\Lambda^{\chi} = \{0, 1, 2\}$



Hybrid Flows

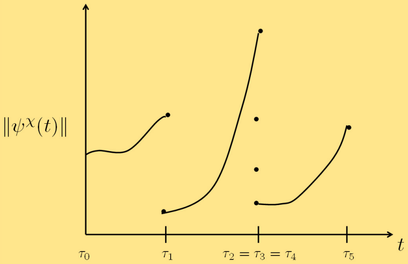
The *continuous evolution* or *hybrid flow* is given by

$$\psi^\chi : \mathcal{T}^\chi \to \mathscr{P}(\bigcup_{i\in Q} D_i), \qquad t \mapsto \psi^\chi(t)$$

where

$$\psi^\chi(t) = \{\phi_{\rho(i)}(t-\tau_i, \xi_i) : t \in [\tau_i, \tau_{i+1}]\}$$

If $\chi \in \mathscr{E}(\mathbf{H}, x_0)$ can define $\psi^\chi(t, x_0)$ analogously.



Outline

Historic Motivation: The Dichotomy

Zeno's Paradox (the Dichotomy)

There is no motion because that which is moved must arrive at the middle of its course before it arrives at the end, and so on ad infinitum.

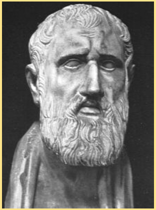
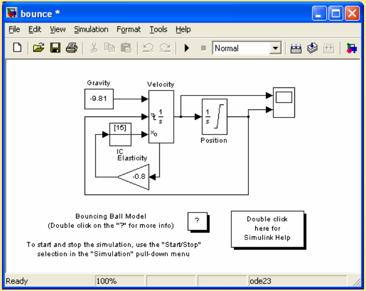


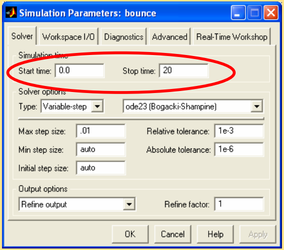
Figure: $Z\eta\nu\omega\nu\ \ o\ \ H\lambda\varepsilon\iota o\varsigma$

An Real "Example" of Zeno's Paradox



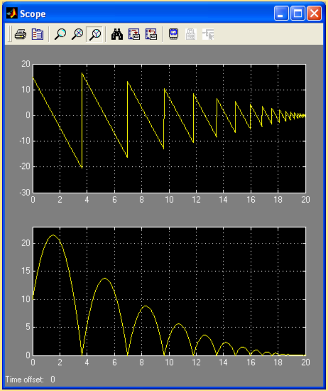
Bouncing Ball simulation in Matlab

An Real ”Example” of Zeno’s Paradox

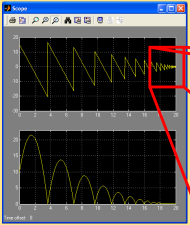


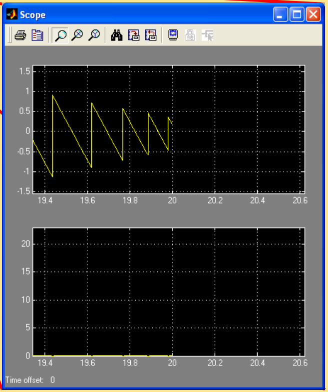
Running the simulation with

Stop time = 20



An Real ”Example” of Zeno’s Paradox

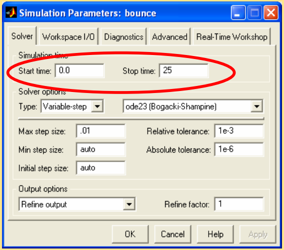




Zooming in:

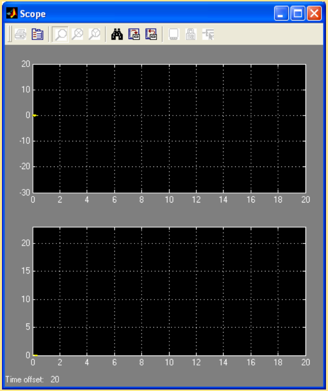
The bouncing ball has not stopped bouncing.

An Real ”Example” of Zeno’s Paradox

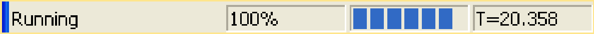


Running the simulation with

Stop time = 25



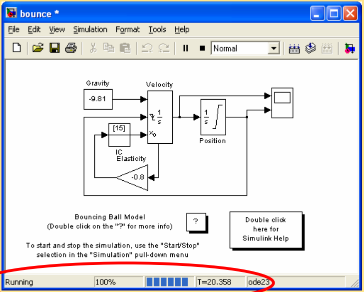
An Real "Example" of Zeno's Paradox



The program will

NEVER

stop running!



Zeno

Definition

H is *Zeno* if there exists an execution χ such that $|\Lambda^\chi| = \infty$ and there exists a finite constant τ_∞ such that

$$\lim_{i \to \infty} \tau_i = \sum_{i \in \Lambda} (\tau_{i+1} - \tau_i) = \tau_\infty$$

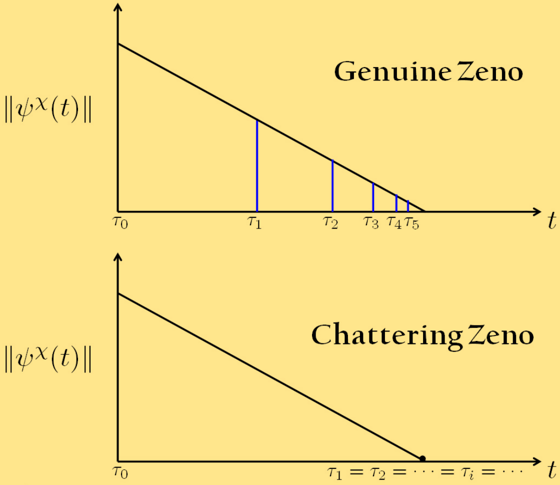
The execution χ is called a *Zeno execution*.

Types of Zeno Behavior

There are two qualitatively different types of Zeno behavior. If χ is Zeno, then χ is

- **Chattering Zeno:** If there exists a finite C such that $\tau_{i+1} - \tau_i = 0$ for all $i \geq C$.
  - Chattering Zeno executions result from the existence of a switching surface in which the vector fields "oppose" each other; for this reason they are easy to detect.
- **Genuine Zeno:** If $\tau_{i+1} - \tau_i > 0$ for all $i \in \mathbb{N}$.
  - There currently is no way to detect the existence of genuinely Zeno executions, and very little has been done in the area of eliminating these executions.

Example


Genuine Zeno
$\|\psi^\chi(t)\|$
τ_0
τ_1
τ_2
τ_3
τ_4
τ_5
t
Chattering Zeno
$\|\psi^\chi(t)\|$
τ_0
$\tau_1 = \tau_2 = \cdots = \tau_i = \cdots$
t

Zeno Detection

Let $\Gamma = (Q, E)$ be an oriented graph. Let $E = \{e_1, \ldots, e_{|E|}\}$. The incidence matrix of Γ, K_Γ, is a $|Q| \times |E|$ matrix given by

$$K_\Gamma = \begin{pmatrix} \lambda_{\mathfrak{t}(e_1)} - \lambda_{\mathfrak{s}(e_1)} & \cdots & \lambda_{\mathfrak{t}(e_{|E|})} - \lambda_{\mathfrak{s}(e_{|E|})} \end{pmatrix}$$

where λ_i is the i^{th} standard basis vector for $\mathbb{R}^{|Q|}$.

Proposition

Let Γ be the underlying graph of the hybrid system $\mathbf{H}$. Then

$$\dim(\mathcal{N}(K_\Gamma)) = 0 \qquad \Rightarrow \qquad \mathbf{H} \text{ is not Zeno}$$

Outline

Invariant Sets

- For a dynamical system (X, f), for $A \subseteq X$,

$$\mathrm{Inv}(A) = \{x \in A : \phi_t(x) \in A \text{ for all } t \in \mathbb{R}_{\geq 0}\}$$

  A set S is (positive) invariant if $\mathrm{Inv}(S) = S$.
- For a hybrid system $\mathbf{H}$ and a set $A \subseteq \bigcup_{i \in Q} D_i$

$$\mathrm{Inv}(A) = \{x \in A : \psi^{\chi}(t, x) \subseteq A \text{ for all } t \in \mathcal{T}^{\chi} \text{ and } \chi \in \mathscr{E}(\mathbf{H}, x)\}$$

  A set S is a *hybrid invariant set* or just an *invariant set* if $\mathrm{Inv}(S) = S$.

Basic Invariant Sets for Dynamical Systems

For a dynamical system (X, f) there are two basic types of invariant sets:

- **EP** = *Equilibrium point*: A point $x^* \in X$ such that $f(x^*) = 0$.
- **PO** = *Periodic orbit*: A set γ (not equal to a point) such that for all $x_0 \in \gamma$

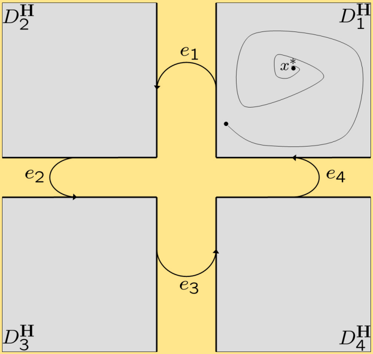
$$\phi_{t+T}(x_0) = \phi_t(x_0)$$

  for some constant $T > 0$.

Basic Invariant Sets for Hybrid Systems

For a hybrid system **H** there are four basic types of invariant sets:

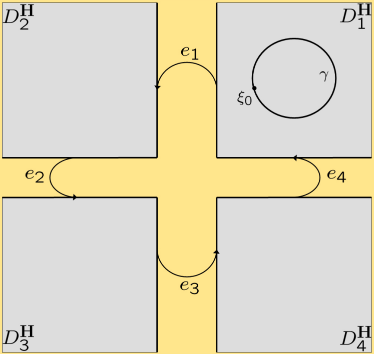
- **CSEP** = *Continuous State Equilibrium point*: An equilibrium point of (D_i, f_i) for some $i \in Q$.



Basic Invariant Sets for Hybrid Systems

For a hybrid system **H** there are four basic types of invariant sets:

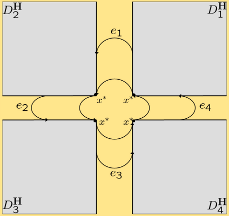
- **CSPO** = *Continuous State Periodic Orbit*: A periodic orbit of (D_i, f_i) for some $i \in Q$.



Basic Invariant Sets for Hybrid Systems

For a hybrid system **H** there are four basic types of invariant sets:

- **DSPO** = *Discrete State Periodic Orbit*: A set of points $X^* = \{x_1^*, \ldots, x_k^*\}$ such that $\psi^{\chi}(t,x) \equiv X^*$ for $x \in X^*$, $\chi \in \mathscr{E}(\mathbf{H}, x)$ and
$$\rho(i) = \rho(i + pK^{\chi})$$
for some integer $K^{\chi} > 0$ (dependent on χ) and all $p \in \mathbb{Z}^*$.



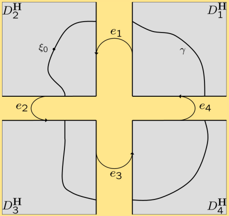
Basic Invariant Sets for Hybrid Systems

For a hybrid system **H** there are four basic types of invariant sets:

- **MSPO** = *Mixed State Periodic Orbit*: A set γ (not equal to a point) such that if $\chi \in \mathscr{E}(\mathbf{H}, x)$ for $x \in \gamma$, then

$$\psi^{\chi}(t,x) = \psi^{\chi}(t + T^{\chi}, x), \qquad \rho(i) = \rho(i + pK^{\chi})$$

  for some integer $K^{\chi} > 0$ and constant T^{χ} (dependent on χ) and all $p \in \mathbb{Z}^*$.



Equilibrium Points and Closed Orbits

Definition

Let

$$\mathscr{O}^{\mathbf{H}} = \left\{ \begin{array}{c} \textbf{CSEP}\text{'s}\ ,\ \textbf{DSPO}\text{'s}\ ,\ \textbf{CSPO}\text{'s} \\ \text{and } \textbf{MSPO}\text{'s of } \mathbf{H} \end{array} \right\} .$$

Outline

Different Forms of Stability: Lyapunov

Let $B_\delta(\mu)$ be a neighborhood of $\mu \in \mathscr{O}^{\mathbf{H}}$ given by

$$B_\delta(\mu) = \{x \in \bigcup_{i \in Q} D_i : \ d(x,\mu) = \min_{y \in \mu} \|x - y\| < \delta\}.$$

and let $d(\psi_t(\chi), \mu)$ be the distance between sets, i.e.,

$$d(\psi_t(\chi), \mu) = \min_{x \in \psi_t(\chi)} \min_{y \in \mu} \|x - y\|.$$

For $\xi_0 \in B_\delta(\mu)$, $\mu \in \mathscr{O}^{\mathbf{H}}$ is

(LYP = Stable in the sense of Lyapunov)

If for all $\chi \in \mathscr{E}(\mathbf{H}, \xi_0)$ *there exists an* $\epsilon > 0$ *such that for all* $t \in \mathcal{T}^\chi$

$$d(\psi^\chi(t, \xi_0), \mu) \leq \epsilon.$$

Different Forms of Stability: Asymptotic

Let $B_\delta(\mu)$ be a neighborhood of $\mu \in \mathscr{O}^{\mathbf{H}}$ given by

$$B_\delta(\mu) = \{x \in \bigcup_{i \in Q} D_i : \ d(x, \mu) = \min_{y \in \mu} \|x - y\| < \delta\}.$$

and let $d(\psi_t(\chi), \mu)$ be the distance between sets, i.e.,

$$d(\psi_t(\chi), \mu) = \min_{x \in \psi_t(\chi)} \min_{y \in \mu} \|x - y\|.$$

For $\xi_0 \in B_\delta(\mu)$, $\mu \in \mathscr{O}^{\mathbf{H}}$ is

(ASY = Asymptotically stable)

If for all $\chi \in \mathscr{E}(\mathbf{H}, \xi_0)$, μ is **LYP** *and*

$$\lim_{t \to \sup \mathcal{T}^\chi} d(\psi^\chi(t, \xi_0), \mu) \to 0$$

Different Forms of Stability: Exponential

Let $B_\delta(\mu)$ be a neighborhood of $\mu \in \mathscr{O}^{\mathbf{H}}$ given by

$$B_\delta(\mu) = \{x \in \bigcup_{i \in Q} D_i : \ d(x,\mu) = \min_{y \in \mu} \|x - y\| < \delta\}.$$

and let $d(\psi_t(\chi), \mu)$ be the distance between sets, i.e.,

$$d(\psi_t(\chi), \mu) = \min_{x \in \psi_t(\chi)} \min_{y \in \mu} \|x - y\|.$$

For $\xi_0 \in B_\delta(\mu)$, $\mu \in \mathscr{O}^{\mathbf{H}}$ is

(EXP = Exponentially stable)

If for all $\chi \in \mathscr{E}(\mathbf{H}, \xi_0)$ there exists and $\alpha, M > 0$ such that

$$d(\psi^\chi(t,\xi_0), \mu) \leq M e^{-\alpha t} d(\xi_0, \mu).$$

$\mathscr{P}$-Stability Equivalence

A stability property is denoted by $\mathscr{P}$ = **LYP**, **ASY**, or **EXP**.

Definition

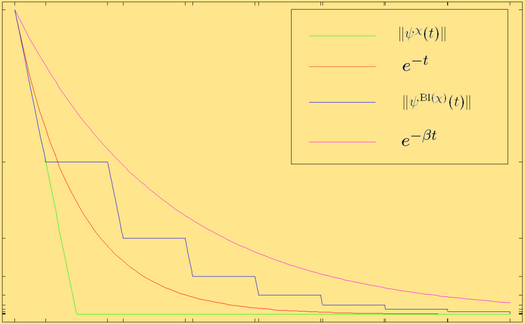
Two hybrid systems **H** and **G** are $\mathscr{P}$*-stability equivalent* if there exists a bijection $\Upsilon : \mathscr{O}^{\mathbf{H}} \to \mathscr{O}^{\mathbf{G}}$ such that

$$\mu \text{ is } \mathscr{P}\text{-stable} \quad \Leftrightarrow \quad \Upsilon(\mu) \text{ is } \mathscr{P}\text{-stable}$$

Blowing Up Hybrid Systems

Theorem

Let **H** *be an affine hybrid system with identity reset maps. There exists a hybrid system,* BL(**H**)*, such that* **H** *and* BL(**H**) *are* $\mathscr{P}$*-stability equivalent and* BL(**H**) *is not Zeno.*



Outline

Difficulties in Simulating Hybrid Systems

- Event Detection: The problem of finding the time when the next discrete event occurs without knowledge of the actual solutions of the ODE's.

Difficulties in Simulating Hybrid Systems

- Event Detection: The problem of finding the time when the next discrete event occurs without knowledge of the actual solutions of the ODE's.
- Ordering Events: The problem of determining which event occurs first, and whether events occur simultaneously (nondeterminism)–again, without knowledge of the solutions of the ODE's.

Difficulties in Simulating Hybrid Systems

- Event Detection: The problem of finding the time when the next discrete event occurs without knowledge of the actual solutions of the ODE's.
- Ordering Events: The problem of determining which event occurs first, and whether events occur simultaneously (nondeterminism)–again, without knowledge of the solutions of the ODE's.

Both of these problems are unsolved; it currently cannot be gauged how well-posed or ill-posed either of these problems are given a specific hybrid system.

Outline

The Initial Value Problem (IVP)

Initial Value Problem (IVP)

The problem of finding the solution $\phi(t - t_0, x_0)$ *to the ODE*

$$\dot{x} = f(x)$$

on some interval $[t_0, t_f]$ *subject to an initial condition* $\phi(0, x_0) = x_0$; *we denote such an IVP by*

$$\mathcal{I} = (f, [t_0, t_F], x_0).$$

Approximate solutions

It is possible to solve the IVP by using numerical integration techniques that guarantee accuracy.

- A numerical integration technique is an integration technique that associates to the IVP, $\mathcal{I} = (f, [t_0, t_F], x_0)$, an approximate solution $\hat{\phi}(t - t_0, x_0)$ on $[t_0, t_f]$ such that $\hat{\phi}(0, x_0) = x_0$.
- In some cases the numerical integration method produces a solution that is accurate of order $\mathcal{M}(h, t)$, where $\mathcal{M}(t, h)$ is a function such that $\mathcal{M}(0, h) = 0$ and $\mathcal{M}(t, h) \rightarrow 0$ as $h \rightarrow 0$ (here h is related to the integration step size).
- In other words, for the IVP $\mathcal{I}$ there exists a constant $C_{\mathcal{I}}$ such that

$$\|\phi(t - t_0, x_0) - \hat{\phi}(t - t_0, x_0)\| \leq C_{\mathcal{I}} \mathcal{M}(t - t_0, h).$$

Outline

The Event Detection Problem (EDP)

Let $G = \{x : g(x) = 0\}$ be a switching surface (a guard) determined by some smooth *event function* $g : \mathbb{R}^n \to \mathbb{R}$.

Event Detection Problem (EDP)

The problem of finding, for an IVP $\mathcal{I} = (f, [t_0, t_F], x_0)$, the first time t^* such that

$$g(\phi(t^* - t_0, x_0)) = 0.$$

The EDP and Approximate Solutions

- One can attempt to solve the EDP for the IVP $\mathcal{I}$ by using an approximate solution, i.e., solving for the first t^{**} such that

$$g(\hat{\phi}(t^{**} - t_0, x_0)) = 0$$

and hope that $|t^* - t^{**}| \approx 0$.

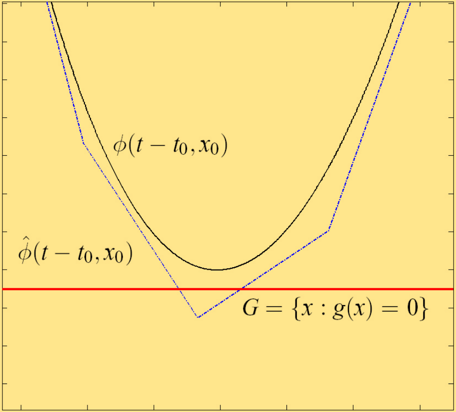
- This is not guaranteed. There is currently no way to verify if $|t^* - t^{**}| \approx 0$.
- What is needed is a condition number, i.e., a number dependent on the IVP $\mathcal{I}$, cond$\#(\mathcal{I})$, such that

$$|t^* - t^{**}| \leq \text{cond}\#(\mathcal{I})$$

This should be as tight a bound as possible.

Ill-posed Event Detection

It may happen that $|t^* - t^{**}|$ does not exist. There may exist a t^{**} but no t^*, or vice versa.



Matlab’s Advice

”If you suspect this is happening, tighten the error tolerances to ensure that the solver takes small enough steps.”

Outline

The Event Ordering Problem (EOP)

Consider a collection $\{g_i\}_{i\in I}$ of event functions $g_i(x)$ such that $g_i(x_0) > 0$ for the IVP $\mathcal{I} = (f, [t_0, t_F], x_0)$.

Event Ordering Problem (EOP)

The problem of finding a function $\mathrm{order} : I \to \mathbb{N}$ *such that if* t_i^* *is the minimum time such that*

$$g_i(\phi(t_i^* - t_0, x_0)) = 0$$

then

$$\begin{aligned} \mathrm{order}(i) < \mathrm{order}(j) \quad &\Leftrightarrow \quad t_i^* < t_j^* \\ \mathrm{order}(i) = \mathrm{order}(j) \quad &\Leftrightarrow \quad t_i^* = t_j^* \end{aligned}$$

The EOP and Approximate Solutions

- One can attempt to solve the EOP for the IVP $\mathcal{I}$ and the collection $\{g_i\}_{i\in I}$ of event functions by using an approximate solution, i.e., solving for the first t_i^{**} such that

$$g_i(\hat{\phi}(t_i^{**} - t_0, x_0)) = 0$$

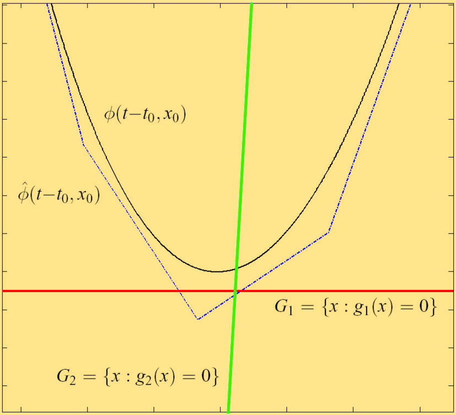
and then define $\widehat{\text{order}}$ by requiring that

$$\begin{aligned} \widehat{\text{order}}(i) < \widehat{\text{order}}(j) \quad &\Leftrightarrow \quad t_i^{**} < t_j^{**} \\ \widehat{\text{order}}(i) = \widehat{\text{order}}(j) \quad &\Leftrightarrow \quad t_i^{**} = t_j^{**} \end{aligned}$$

- There is currently no way to verify if $\widehat{\text{order}} = \text{order}$. This will only hold if the event functions are sufficiently separated.

Ill-posed Event Ordering

It may happen that $\widehat{\text{order}}$ and order give the opposite ordering of I, i.e., $\widehat{\text{order}}(1) < \widehat{\text{order}}(2)$ while $\text{order}(1) > \text{order}(2)$.



Outline

Simulating Hybrid Systems

- To simulate the executions of a hybrid system, approximate executions must be solved for by using the approximate solutions to the ODE's on each domain.
- For an approximate execution $\hat{\chi} = (\hat{\Lambda}, \hat{\tau}, \hat{\xi}, \hat{\eta})$ of an execution $\chi = (\Lambda, \tau, \xi, \eta)$ the EDP and the EOP must be solved: Need

$$\begin{aligned} |\hat{\tau}_i - \tau_i| &\approx 0 \\ \widehat{\text{order}}(\hat{\tau}_i) &= \text{order}(\tau_i) \end{aligned}$$

- Currently, no solution to the EDP and the EOP, so it is not possible to verify if a simulated execution is a valid approximation of an actual execution.

Challenges

Open Problem

Determine conditions on a hybrid system so that it can be guaranteed that a simulation of that hybrid system is correct.

Open Problem

Find bounds on the error between an actual execution and its approximation.

Conclusion

- Hybrid Systems model the interaction of discrete and continuous systems.
- This interaction results in the ability to model very complicated behavior. The cost is a complicated (or at least notation heavy) theory.
- Simulating hybrid systems is both challenging and subtle.
- There is currently no way to verify whether a simulation is correct–this is an IMPORTANT and OVERLOOKED area of hybrid systems.