Modeling & Control of Hybrid Systems

Chapter 1 — Introduction [1]

Manuel Mazo Jr.

Utrecht, June 26th, 2020

[1] Based on the original slides from Bart De Schutter

Outline

Outline

General Info

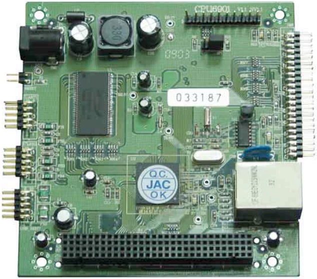
- Lecturers: Manuel Mazo Jr and Romain Postoyan
- Web site: https://mmazojr.3me.tudelft.nl/teaching/disc_hs/
- Lecture notes: on DISC course folder, linked on course website
- Slides: see website
- Homework: see website (also for deadlines)
- Final grade: average of 3 homework assignments
  + bonus points (by reporting errors)
  results will be communicated by end of October 2020
- Email addresses:
  - m.mazo@tudelft.nl
  - romain.postoyan@univ-lorraine.fr

Contents

Outline

Control systems meets computing

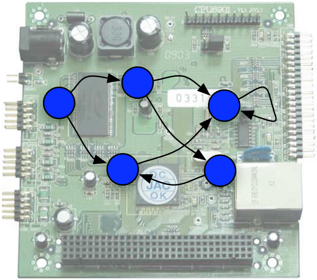
Cyber



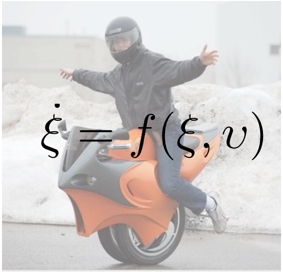
Physical

Control systems meets computing

Cyber



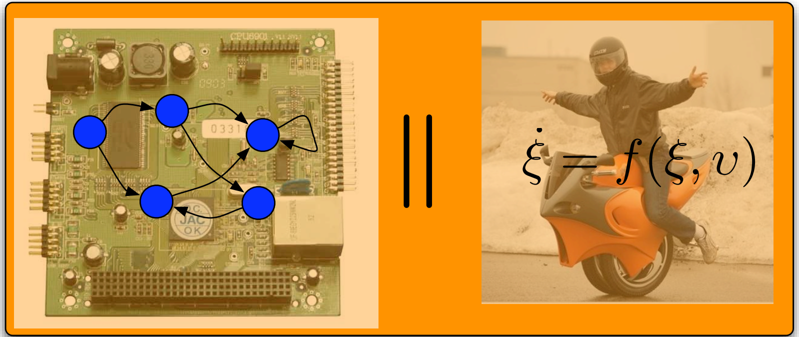
Physical



$$\dot{\xi} = f(\xi, \upsilon)$$

Control systems meets computing

Cyber-Physical



Control systems meets computing

Cyber-Physical

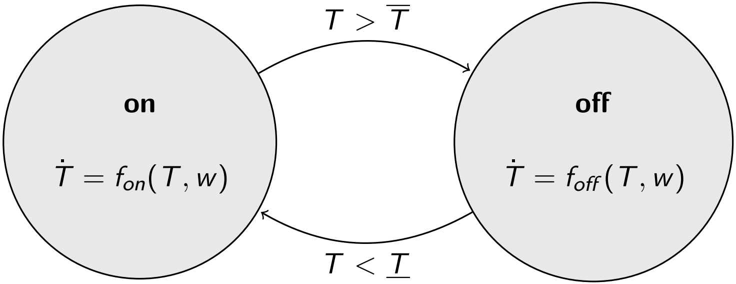
Switching dynamical regimes

- Evolution of rigid bodies, impact dynamics (contact/no contact)
- (Active) Electrical networks (switching, diodes)
- Fermentation process (lag, growth, stationary, inactivation)
- Saturation, hysteresis
- Actuator and sensor failures
- Human intervention in smooth systems

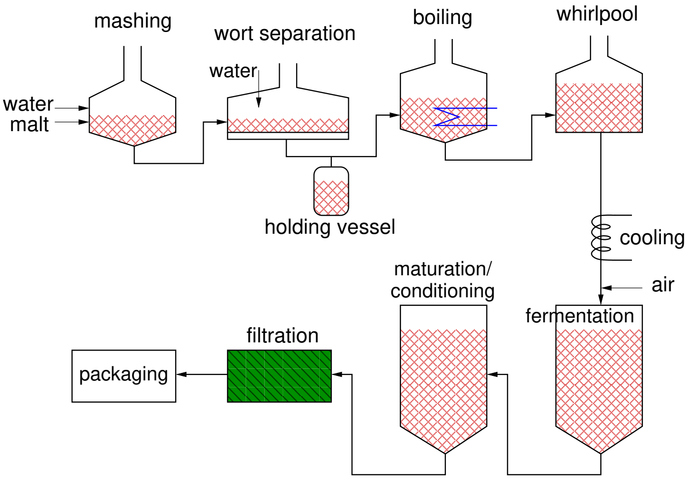
Switching between dynamical regimes $\rightarrow$ hybrid

A classical example

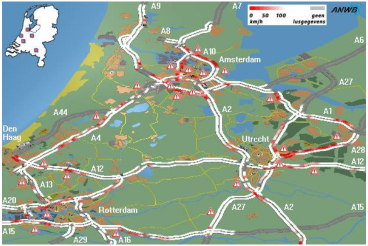
- Hybrid: combination of continuous and discrete dynamics
- Temperature control system:

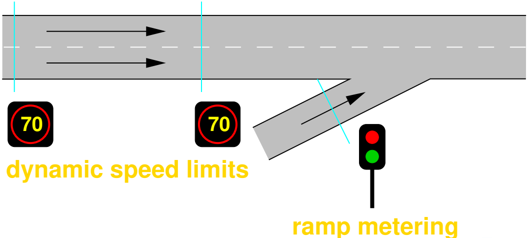


Beer brewing



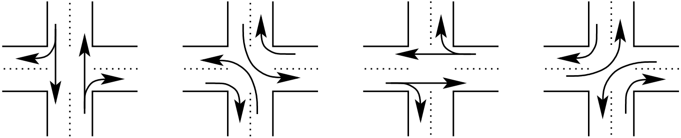
Traffic control systems





Traffic control systems

- Intersection with traffic signals

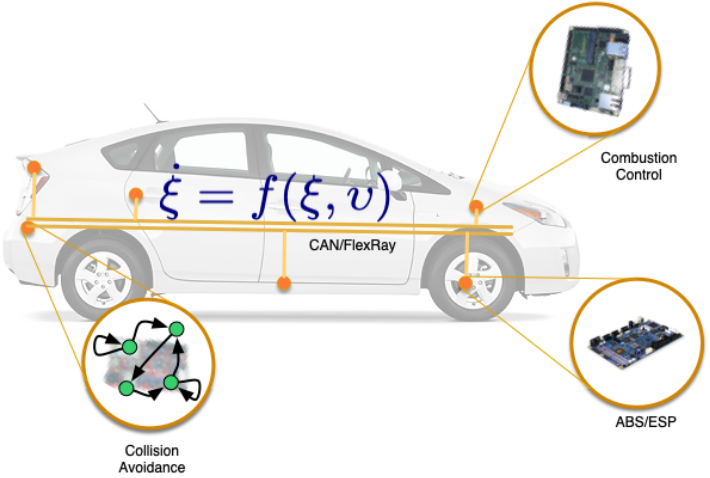


  4 modes, states: queue lengths

- Automatic platooning

  merging & splitting

Networked Control Systems



Challenges

- Analysis — Verification of properties/specifications
- Control — Synthesis for prescribed properties
- Traditional approaches:
  - often heuristic & ad-hoc
  - focus exclusively on either continuous or discrete dynamics

  $\rightarrow$ structured approach necessary
- Consider hybrid nature of systems (holistic view)
- Combination of systems & control, computer science, optimization, communications, mathematics, simulation...

Outline

Systems

Definition (System or Machine, Sontag)

A *system* or *machine* $\Sigma = (\mathcal{T}, \mathcal{X}, \mathcal{U}, \phi)$ consists of:

- A time set $\mathcal{T}$;
- A nonempty set $\mathcal{X}$ called the state space of Σ;
- A nonempty set $\mathcal{U}$ called the control-value or input-value space of Σ;
- A map $\phi : \mathcal{D}_\phi \to \mathcal{X}$ called the transition map of Σ, which is defined on a subset $\mathcal{D}_\phi$ of $\{(\tau, \sigma, x, \omega) \mid \tau, \sigma \in \mathcal{T}, \tau \leq \sigma, x \in \mathcal{X}, \omega : [\tau, \sigma) \to \mathcal{U}\}$ such that the non-triviality, restriction, semi-group and identity properties (see [Son98] for exact descriptions) hold.

- Example: $\dot{x}(t) = f(x(t), u(t))$, t: time, x: state, u: input

Son98 E.D. Sontag. Mathematical Control Theory: Deterministic Finite Dimensional Systems. Springer, New York, 1998. Texts in applied Mathematics, vol. 6

Generalized Transition Systems

Definition (Generalized Transition System)

A system is a sextuple $(X, X_0, U, \longrightarrow, Y, H)$ consisting of:

- a set of states X;
- a set of initial states $X_0 \subseteq X$;
- a set of inputs U;
- a transition relation $\longrightarrow \subseteq X \times U \times X$;
- a set of outputs Y;
- an output map $H : X \to Y$.

Tab09 P. Tabuada. Verification and control of hybrid systems: a symbolic approach. Springer Science & Business Media, 2009.

Classification of systems

- Continuous-state / discrete-state / finite-state ($\mathcal{X}$ or X)
- Continuous-time / discrete-time ($\mathcal{T}$)
- Time-driven / event-driven
  - time-driven $\rightarrow$ state changes as time progresses, i.e., continuously (for continuous-time), or at every tick of a clock (for discrete-time)
  - event-driven $\rightarrow$ state changes due to occurrence of event:
    - start or end of an activity
    - aperiodic (occurrence times not necessarily equidistant)

Combinations $\Rightarrow$ "hybrid"

Models for time-driven systems

- Continuous-time time-driven systems:

$$\dot{x}(t) = f(x(t), u(t))$$
$$y(t) = g(x(t), u(t))$$

- Discrete-time time-driven systems:

$$x(k+1) = f(x(k), u(k))$$
$$y(k) = g(x(k), u(k))$$

Models for event-driven systems

Definition (Automaton)

An *Automaton* is defined by the tuple $\Sigma = (\mathcal{Q}, \mathcal{Q}_0, \mathcal{U}, \mathcal{F}\phi)$ with

- $\mathcal{Q}$: finite or countable set of discrete states
- $\mathcal{Q}_0 \subseteq \mathcal{Q}$: subset of initial states
- $\mathcal{U}$: finite or countable set of discrete inputs ("input alphabet")
- $\mathcal{F} \subseteq \mathcal{Q}$: subset of final (or accepting) states
- $\phi : \mathcal{Q} \times \mathcal{U} \to P(\mathcal{Q})$: partial *transition function*.

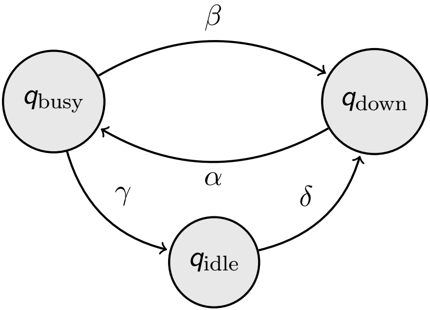
where $P(\mathcal{Q})$ is power set of $\mathcal{Q}$ (set of all subsets)

Finite automaton: $\mathcal{Q}$ and $\mathcal{U}$ finite.
Alternatively one can denote $\phi \subseteq \mathcal{Q} \times \mathcal{U} \times \mathcal{Q}$.
Depending on context often $\mathcal{Q}_0$ and $\mathcal{F}$ are dropped.
$P(X)$, often also denoted 2^X is the power set of X, i.e. the set of all subsets of X.

Evolution of automaton

- Given state $q \in \mathcal{Q}$ and discrete input symbol $u \in \mathcal{U}$, transition function ϕ defines collection of next possible states: $\phi(q, u) \subseteq \mathcal{Q}$
- Accepting states are used on automata to model computation, e.g. language acceptance.
  Acceptance depends on the type of automaton, e.g. finite, Büchi, Rabin,...
- If each set of next states has 0 or 1 element:
  $\rightarrow$ "deterministic" automaton
- If some set of next states has more than 1 element:
  $\rightarrow$ "non-deterministic" automaton

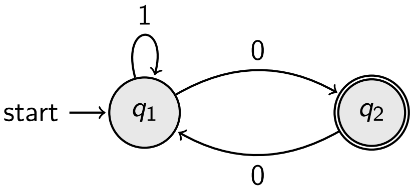
Deterministic automaton



$$\phi(q_{\text{busy}}, \beta) = \{q_{\text{down}}\} \qquad \phi(q_{\text{down}}, \alpha) = \{q_{\text{busy}}\}$$

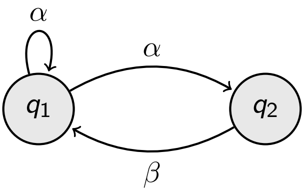
$$\phi(q_{\text{busy}}, \gamma) = \{q_{\text{idle}}\} \qquad \phi(q_{\text{idle}}, \delta) = \{q_{\text{down}}\}$$

Accepting automaton



- Acceps strings of the form: 10, 110110, 1000
- Does not accept: 100 or 1111
- The accepted strings define a language, in this case: $\{((1^*(00)^*)^*0\}$

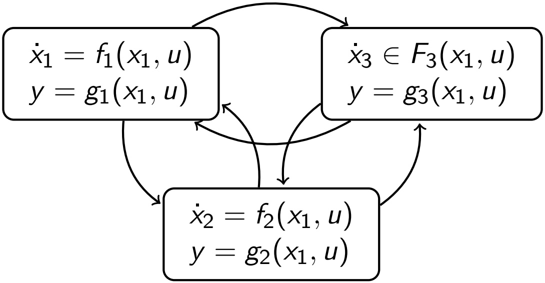
Non-deterministic automaton



$$\phi(q_1, \alpha) = \{q_1, q_2\} \qquad \phi(q_2, \beta) = \{q_1\}$$

$\rightarrow$ unmodeled dynamics, e.g. environment (player)

Hybrid system



- System can be in one of several modes
- In each mode: behavior described by system of difference or differential equations
- Mode switches due to occurrence of "events"

Hybrid system

- At switching time instant:
  $\rightarrow$ possible state reset or state dimension change
- Mode transitions may be caused by
  - external control signal
  - internal control signal
  - dynamics of system itself (crossing of boundary in state space)

Models for hybrid systems

- timed or hybrid Petri nets
- differential automata
- hybrid automata
- Brockett's model
- mixed logical dynamic models
- real-time temporal logics
- timed communicating sequential processes
- switched bond graphs
- predicate calculus
- piecewise-affine models
- ...

Analysis techniques

- formal verification
- computer simulation
- analytic techniques (for special subclasses)
- ...

$\Rightarrow$ no general modeling & analysis framework

modeling power $\leftrightarrow$ decision power

$+$ computational complexity (NP-hard, undecidable)

$\Rightarrow$ special subclasses (Chapter 2)
hierarchical / modular approach

Decidability and complexity

- **Undecidable problems**
  $\rightarrow$ no algorithm can solve the problem in general,
  i.e., finite termination cannot be guaranteed

- **NP-complete and NP-hard problems**
  - *decision problem*: solution is either "yes" or "no"
    e.g., traveling salesman decision problem:
    Given a network of cities, intercity distances, and a number B, does there exist a tour with length $\leqslant B$?
  - *search problem*
    e.g., traveling salesman problem:
    Given a network of cities, intercity distances, what is the shortest tour?

P and NP-complete decision problems

- Time complexity function $T(n)$: largest amount of time needed to solve problem instance of size n (*worst case!*)
- Polynomial time algorithm:

$$T(n) \leqslant |p(n)| \qquad \text{for some polynomial } p$$

$\rightarrow$ class P: solvable in polynomial time on a deterministic computer

- Nondeterministic computer:
  - guessing stage (tour)
  - checking stage (compute length of tour + compare it with B)

$\rightarrow$ class NP: "*nondeterministically polynomial*"

i.e., time complexity of checking stage is polynomial

N.B.: Computer here is used in the sense of a "Turing Machine"

P and NP-complete decision problems

- Every problem in NP can be solved in exponential time: $T(n) \leqslant 2^{n^k}$

Definition (NP-complete)

An NP problem $\mathcal{X}$ is NP-complete iff every NP problem $\mathcal{Y}$ can be reduced to $\mathcal{X}$ in polynomial time.

- NP-complete problems: "hardest" class in NP:
- any NP-complete problem solvable in polynomial time
  $\Rightarrow$ every problem in NP solvable in polynomial time
- any problem in NP intractable
  $\Rightarrow$ NP-complete problems also intractable

NP-hard problems

Definition (NP-hard)

A problem $\mathcal{X}$ is NP-hard, if there exist an NP-complete problem $\mathcal{Y}$, such that $\mathcal{Y}$ is reducible to $\mathcal{X}$ in polynomial time.

Remark: In this case $\mathcal{X}$ is *not* necessarily an NP problem.

- Decision problem is NP-complete $\Rightarrow$ search problem is NP-hard
- NP-hard problems: at least as hard as NP-complete problems
  - NP-complete (decision problem)

    $\rightarrow$ solvable in polynomial time *if and only if* P $=$ NP
  - NP-hard (search problem)

    $\rightarrow$ cannot be solved in polynomial time *unless* P $=$ NP

Complexity map

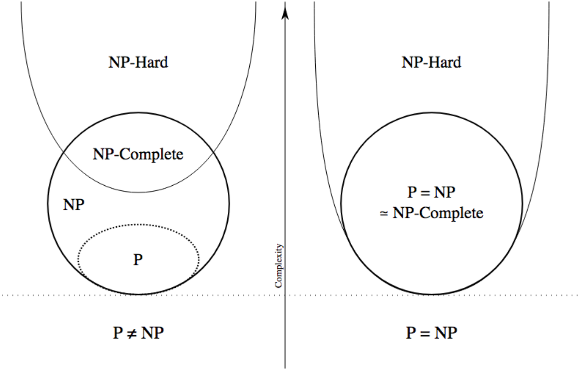


Figure: https://commons.wikimedia.org/wiki/File:P_np_np-complete_np-hard.svg

Examples of NP-hard and undecidable problems

- Consider simple hybrid system:

$$x(k+1) = \begin{cases} A_1 x(k) & \text{if } c^{\mathrm{T}} x(k) \geqslant 0 \\ A_2 x(k) & \text{if } c^{\mathrm{T}} x(k) < 0 \end{cases}$$

$\rightarrow$ deciding whether system is stable or not is NP-hard

- Given two Petri nets, do they have the same reachability set?
  $\rightarrow$ undecidable

Hybrid automaton

Definition (Hybrid automaton)

A Hybrid automaton H is collection $H = (Q, X, f, \mathrm{Init}, \mathrm{Inv}, E, G, R)$ where

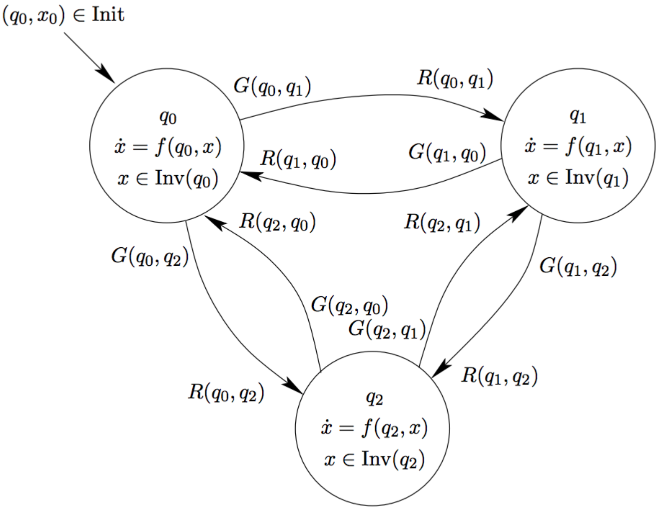
- $Q = \{q_1, \ldots, q_N\}$ is finite set of discrete states or *modes*
- $X = \mathbb{R}^n$ is set of continuous states
- $f : Q \times X \to X$ is a (collection of) vector field(s)
- $\mathrm{Init} \subseteq Q \times X$ is set of initial states
- $\mathrm{Inv} : Q \to P(X)$ describes *invariants*
- $E \subseteq Q \times Q$ is a set of edges or (discrete) *transitions*
- $G : E \to P(X)$ are *guard conditions*
- $R : E \to P(X \times X)$ is *reset map*

Hybrid automaton

Hybrid automaton $H = (Q, X, f, \mathrm{Init}, \mathrm{Inv}, E, G, R)$

- Hybrid state: (q, x)
- Evolution of continuous state in mode q: $\dot{x} = f(q, x)$
- Invariant Inv: describes conditions that continuous state has to satisfy in given mode
- Guard G: specifies subset of state space where certain transition is enabled
- Reset map R: specifies how new continuous states are related to previous continuous states

Hybrid automaton



Evolution of hybrid automaton

- Initial hybrid state $(q_0, x_0) \in \mathrm{Init}$
- Continuous state x evolves according to

$$\dot{x} = f(q_0, x) \quad \text{with } x(0) = x_0$$

  discrete state q remains constant: $q(t) = q_0$
- Continuous evolution can go on as long as $x \in \mathrm{Inv}(q_0)$
- If at some point state x reaches guard $G(q_0, q_1)$, then
  - transition $q_0 \to q_1$ is enabled
  - discrete state *may* change to q_1, continuous state then jumps from current value x^- to new value x^+ with $(x^-, x^+) \in R(q_0, q_1)$
- Next, continuous evolution resumes and whole process is repeated

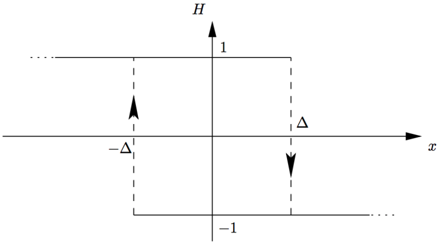
Outline

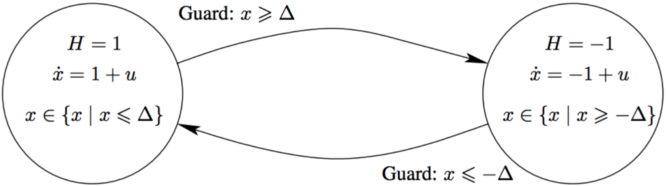
Examples of hybrid systems

1. Hysteresis
2. Manual transmission
3. Water-level monitor
4. Supervisor
5. Two-carts system
6. Boost converter

Control system with Hysteresis

$$\dot{x} = H(x) + u$$





Manual transmission

Simple model of manual transmission

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{-a\,x_2 + u}{1+v}\end{aligned}$$

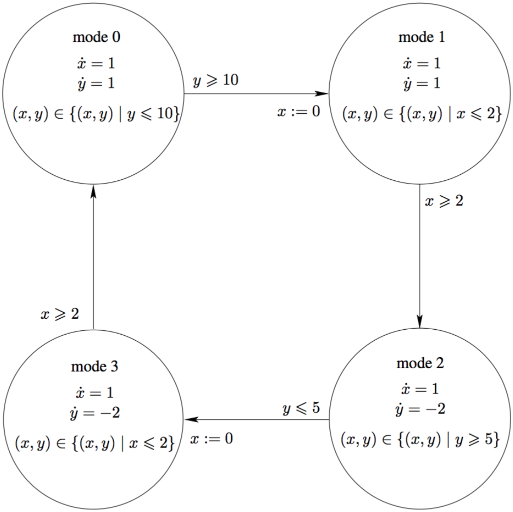
with v: gear shift position $v \in \{1, 2, 3, 4\}$
u: acceleration
a: parameter

$\rightarrow$ hybrid system with four modes, 2-dimensional continuous state, controlled transitions (switchings), and no resets

Water-level monitor

- Variables:
  - $y(t)$: water level, continuous
  - $x(t)$: time elapsed since last signal was sent by monitor, continuous
  - $P(t)$: status of pump, $\in \{\texttt{on}, \texttt{off}\}$
  - $S(t)$: nature of signal last sent by monitor, $\in \{\texttt{on}, \texttt{off}\}$
- Dynamics of system:
  - water level rises 1 unit per second when pump is on and falls 2 units per second when pump is off
  - when water level rises to 10 units, monitor sends switch-off signal; after delay of 2 seconds pump turns off
  - when water level falls to 5 units, monitor sends switch-on signal; after delay of 2 seconds pump switches on

Water-level monitor

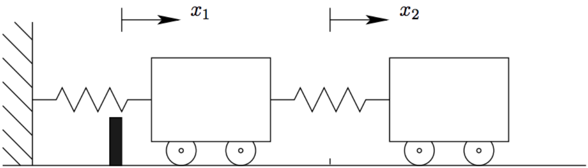


y: water level

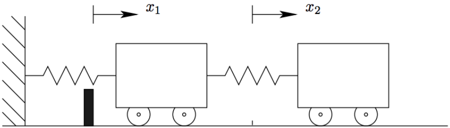
x: time since last signal

Two-carts system

- Two carts connected by spring
- Left cart attached to wall by spring;
  motion constrained by completely inelastic stop
  Stop is placed at equilibrium position of left cart
- Masses of carts and spring constants = 1

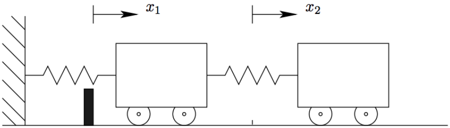


Two-carts system



- x_1, x_2: deviations of left and right cart from equilibrium position
- x_3, x_4: velocities of left and right cart
- z: reaction force exerted by stop
- Evolution:
$$\begin{aligned}\dot{x}_1(t) &= x_3(t)\\ \dot{x}_2(t) &= x_4(t)\\ \dot{x}_3(t) &= -2x_1(t) + x_2(t) + z(t)\\ \dot{x}_4(t) &= x_1(t) - x_2(t)\end{aligned}$$

Two-carts system



To model stop:

- Define $w(t) = x_1(t)$
- $w(t) \geq 0$ (since w is position of left cart w.r.t. stop)
- Force exerted by stop can act only in positive direction $\rightarrow z(t) \geq 0$
- If left cart not at stop ($w(t) > 0$), reaction force vanishes: $z(t) = 0$
- If $z(t) > 0$ then cart must necessarily be at the stop: $w(t) = 0$

$$0 \leq w(t) \perp z(t) \geq 0$$

Two-carts system

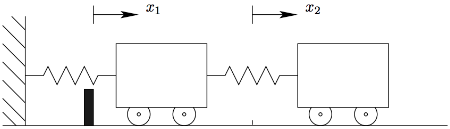
System can be represented by two modes (stop active or not)

$\boxed{z=0}$ unconstrained	constrained $\boxed{w=0}$
$\dot{x}_1(t) = x_3(t)$	$\dot{x}_1(t) = x_3(t)$
$\dot{x}_2(t) = x_4(t)$	$\dot{x}_2(t) = x_4(t)$
$\dot{x}_3(t) = -2x_1(t) + x_2(t)$	$\dot{x}_3(t) = -2x_1(t) + x_2(t) + z(t)$
$\dot{x}_4(t) = x_1(t) - x_2(t)$	$\dot{x}_4(t) = x_1(t) - x_2(t)$
$z(t) = 0$	$w(t) = x_1(t) = 0$
ODE (in state)	DAE (as z is not explicit)

System stays in mode as long as

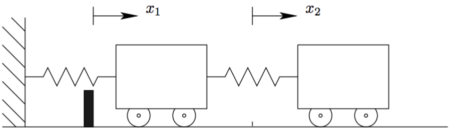
unconstrained	constrained
$z(t) = 0,\ w(t) > 0$	$w(t) = 0,\ z(t) > 0$

Mode transitions for two-carts system



- **Unconstrained → constrained**
  Suppose $x(\tau) = (0^+, -1, -1, 0)^T \to w(t) > 0$ tends to be violated
  Left cart hits stop and stays there. Velocity of left cart is reduced to zero instantaneously (purely inelastic collision)
- **Constrained → unconstrained**
  Suppose $x(\tau) = (0, 0, 0, 1)^T \to z(t) > 0$ tends to be violated
  Right cart is moving to right of its equilibrium position, so spring between carts pulls left cart away from stop

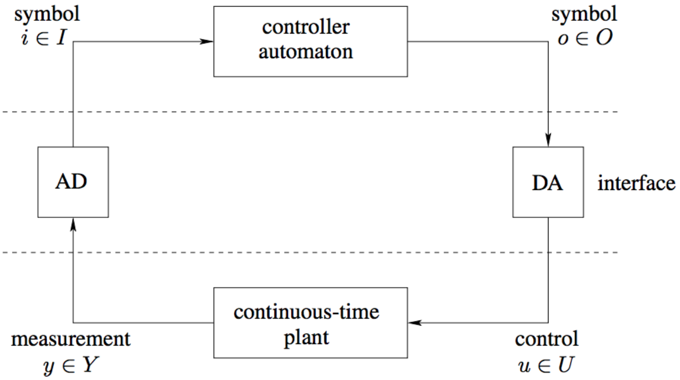
Mode transitions for two-carts system



- **Unconstrained → unconstrained with re-initialization according to constrained mode**
  Consider $x(\tau) = (0^+, 1, -1, 0)^{\mathrm{T}} \rightarrow w(t) > 0$ tends to be violated
  At impact, velocity of left cart is reduced to 0, i.e., state reset to $(0, 1, 0, 0)^{\mathrm{T}}$
  Right cart is at right of its equilibrium position, pulls left cart away from stop → smooth continuation in unconstrained mode

  So: After the reset, no smooth continuation is possible in constrained mode → second mode change, back to unconstrained mode

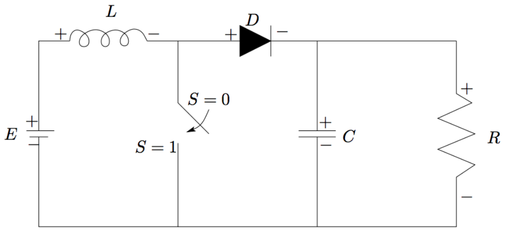
Supervisor model

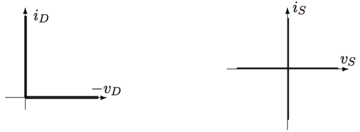


Controller is input-output automaton:

$$\begin{aligned} q &= \nu(q, i) \\ o &= \eta(q, i) \end{aligned}$$

Boost converter



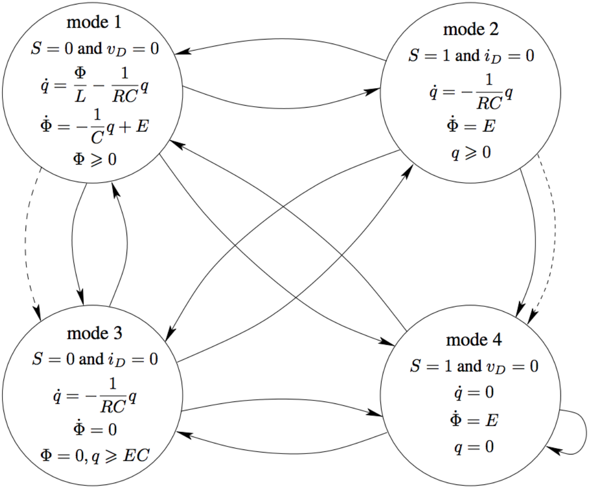


- Presence of switch and diode introduces hybrid dynamics
- 4 modes:
  $(v_S = 0, v_D = 0)$, $(v_S = 0, i_D = 0)$, $(i_S = 0, v_D = 0)$, $(i_S = 0, i_D = 0)$

Boost converter: transitions

transition	guard	reset
mode 1→mode 2	$S = 1$ and $q \geq 0$	
mode 1→mode 3	$\phi = 0$ and $q > CE$	
mode 1⇢mode 3	$\phi < 0$	$\phi^+ = 0$
mode 1→mode 4	$S = 1$ and $q \leq 0$	$q^+ = 0$
mode 2→mode 1	$S = 0$ and $\phi \geq 0$	
mode 2→mode 3	$S = 0$ and $\phi \leq 0$	$\phi^+ = 0$
mode 2→mode 4	$q = 0$	
mode 2⇢mode 4	$q < 0$	$q^+ = 0$
mode 3→mode 1	$q = CE$	
mode 3→mode 2	$S = 1$ and $q \geq 0$	
mode 3→mode 4	$S = 1$ and $q \leq 0$	$q^+ = 0$
mode 4→mode 1	$S = 0$ and $\phi \geq 0$	
mode 4→mode 3	$S = 0$ and $\phi \leq 0$	$\phi^+ = 0$
mode 4→mode 4	$q < 0$	$q^+ = 0$

Boost converter: Hybrid automaton



Boost converter: Linear Complementarity model

Hybrid automaton model is very involved

Alternatively, one may use the more compact model

$$\begin{aligned} \dot{q} &= -\frac{1}{RC}q + i_D \\ \dot{\phi} &= v_S + E \\ -v_D &= \frac{1}{C}q + v_S \\ i_S &= \tfrac{1}{L}\phi - i_D \\ 0 \leq i_D &\perp -v_D \geq 0 \\ v_S &\perp i_S \end{aligned}$$

$\rightarrow$ also complementarity relation (as in two-carts system)

Outline

Zeno behavior

- *Zeno behavior*: infinitely many mode switches in finite time interval
- Examples
  1. bouncing ball
  2. reversed Filippov's system
  3. two-tank system
  4. three-balls example

Bouncing ball

- Dynamics: $\ddot{x} = -g$ subject to $x \geq 0$ ($x(t)$: height)
- Newton's restitution rule ($0 < e < 1$):

$$\dot{x}(\tau+) = -e\dot{x}(\tau-) \quad \text{when } x(\tau-) = 0,\ \dot{x}(\tau-) < 0$$

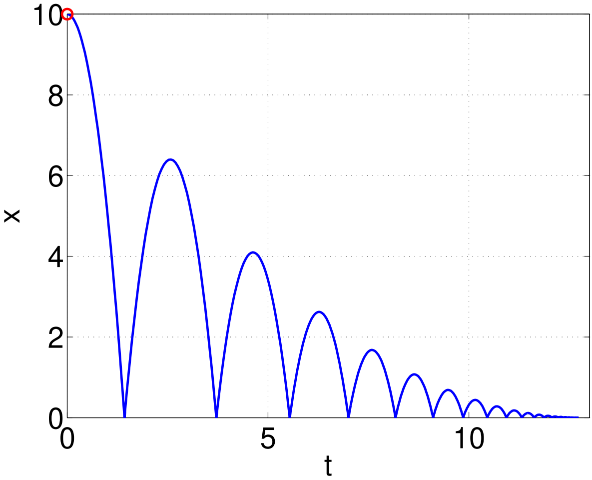
- Assuming $x(0) = 0$, $\dot{x}(0) > 0$, event times are related through

$$\tau_{i+1} = \tau_i + \frac{2e^i\dot{x}(0)}{g}$$

- Sequence has finite limit $\tau^* = \frac{2\dot{x}(0)}{g-ge} < \infty$ (geometric series)
- Physical interpretation: ball is at rest within finite time span, but after infinitely many bounces $\rightarrow$ Zeno behavior

  In this case: infinite number of state re-initializations, set of event times contains *right-accumulation point*

Bouncing ball



Reversed Filippov's example

- Dynamics:

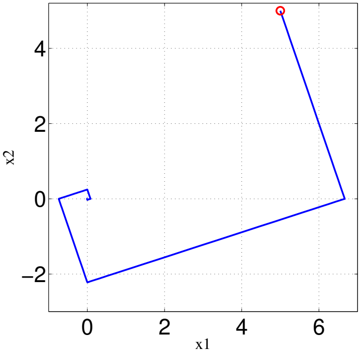
$$\begin{aligned}\dot{x}_1 &= -\text{sgn}(x_1) + 2\text{sgn}(x_2)\\ \dot{x}_2 &= -2\text{sgn}(x_1) - \text{sgn}(x_2),\end{aligned}$$

with

$$\begin{cases}\text{sgn}(x) = 1 & \text{if } x > 0\\ \text{sgn}(x) = -1 & \text{if } x < 0\\ \text{sgn}(x) \in [-1, 1] & \text{when } x = 0\end{cases}$$

- Solutions system are spiraling towards origin, which is an equilibrium

Reversed Filippov's example: Finite time convergence



- Since $\frac{\mathrm{d}}{\mathrm{d}t}(|x_1(t)| + |x_2(t)|) = -2$, solutions reach origin in finite time
- Solutions go through infinite number of mode transitions (relay switches) $\rightarrow$ Zeno behavior

Reversed Filippov's example: Finite time convergence

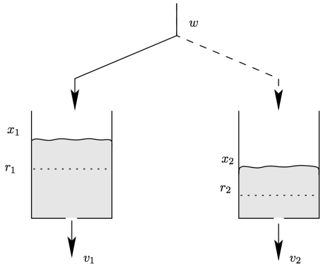
- Dynamics:

$$\begin{aligned}\dot{x}_1 &= -\mathrm{sgn}(x_1) + 2\mathrm{sgn}(x_2)\\ \dot{x}_2 &= -2\mathrm{sgn}(x_1) - \mathrm{sgn}(x_2),\end{aligned}$$

- "Derivative" of absolute value function: $\frac{\mathrm{d}}{\mathrm{d}x}|x| = \mathrm{sgn}(x)$
- So

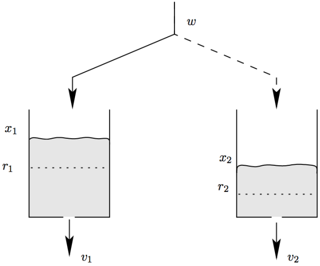
$$\begin{aligned}&\frac{\mathrm{d}}{\mathrm{d}t}(|x_1(t)| + |x_2(t)|)\\ &\quad= \dot{x}_1\mathrm{sgn}(x_1) + \dot{x}_2\mathrm{sgn}(x_2)\\ &\quad= -\mathrm{sgn}^2(x_1) + 2\mathrm{sgn}(x_2)\mathrm{sgn}(x_1) - 2\mathrm{sgn}(x_1)\mathrm{sgn}(x_2) - \mathrm{sgn}^2(x_2)\\ &\quad= -1 - 1\\ &\quad= -2\end{aligned}$$

Two-tank system



- Two tanks (x_i: volume of water in tank)
- Tanks are leaking at constant rate $v_i > 0$
- Water is added at constant rate w through hose, which at any point in time is dedicated to either one tank or the other
- Objective: keep water volumes above r_1 and r_2
- Controller that switches inflow to tank 1 whenever $x_1 \leq r_1$ and to tank 2 whenever $x_2 \leq r_2$

Description of two-tank system as hybrid automaton

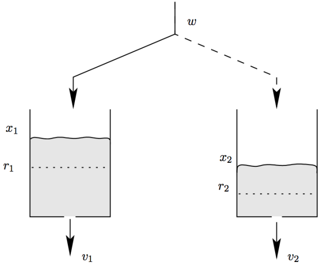


- Two modes: filling tank 1 (mode q_1) or tank 2 (mode q_2)
- Evolution of continuous state:

$$\begin{cases} \dot{x}_1 = w - v_1 \\ \dot{x}_2 = -v_2 \end{cases} \text{in mode } q_1 \qquad \begin{cases} \dot{x}_1 = -v_1 \\ \dot{x}_2 = w - v_2 \end{cases} \text{in mode } q_2$$

- $\mathrm{Init} = \{q_1, q_2\} \times \{(x_1, x_2) \mid x_1 \geq r_1 \text{ and } x_2 \geq r_2\}$

Description of two-tank system as hybrid automaton (cont.)



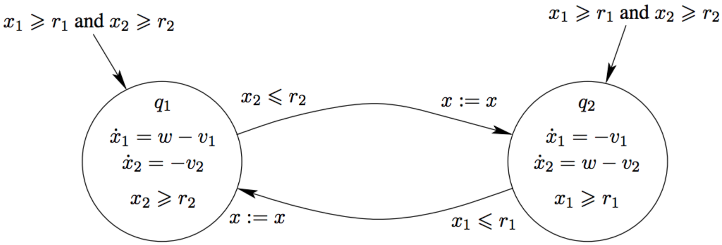
- Invariants: $\mathrm{Inv}(q_1) = \{x \in \mathbb{R}^2 \mid x_2 \geq r_2\}$

  $\mathrm{Inv}(q_2) = \{x \in \mathbb{R}^2 \mid x_1 \geq r_1\}$
- Guards: $G(q_1, q2) = \{x \in \mathbb{R}^2 \mid x_2 \leq r_2\}$

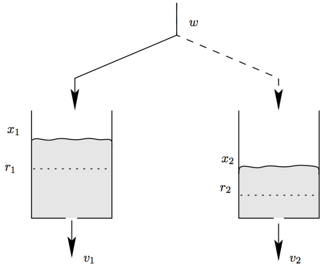
  $G(q_2, q1) = \{x \in \mathbb{R}^2 \mid x_1 \leq r_1\}$
- No resets:

$$R(q_1, q_2) = R(q_2, q_1) = \{(x^-, x^+) \mid x^-, x^+ \in \mathbb{R}^2 \text{ and } x^- = x^+\}$$

Description of two-tank system as hybrid automaton (cont.)

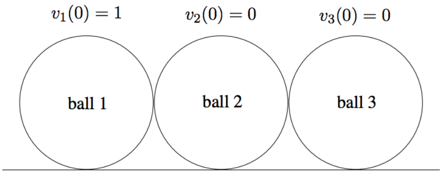


Two-tank system and Zeno behavior



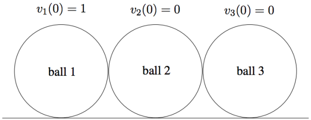
- Assume total outflow $v_1 + v_2 > w$
- Control objective cannot be met and tanks will empty in finite time
- Infinitely many switchings in finite time $\rightarrow$ Zeno behavior

Three-balls example: model



- System consisting of three balls
- Inelastic impacts modeled by successions of simple impacts
- Suppose unit masses, touching at time 0, and $v_1(0) = 1,\ v_2(0) = v_3(0) = 0$
- We model all impacts separately $\rightarrow$
  - first, inelastic collision between balls 1 and 2, resulting in $v_1(0+) = v_2(0+) = 0.5,\ v_3(0+) = 0$

Three balls example: Zeno



- next, ball 2 hits ball 3, resulting in $v_1(0{+}{+}) = \frac{1}{2}$, $v_2(0{+}{+}) = v_3(0{+}{+}) = \frac{1}{4}$
- next, ball 1 hits ball 2 again, etc.

  $\rightarrow$ sequence of resets:
  $$\begin{array}{llllllll} v_1: & 1 & \frac{1}{2} & \frac{1}{2} & \frac{3}{8} & \frac{3}{8} & \frac{11}{32} & \ldots \\ v_2: & 0 & \frac{1}{2} & \frac{1}{4} & \frac{3}{8} & \frac{5}{16} & \frac{11}{32} & \ldots \\ v_3: & 0 & 0 & \frac{1}{4} & \frac{1}{4} & \frac{5}{16} & \frac{5}{16} & \ldots \end{array}$$

  converges to $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})^{\mathrm{T}}$

- Afterwards, smooth continuation is possible with constant and equal velocity for all balls
- Infinite number of events (resets) at one time instant, sometimes called *live-lock* $\rightarrow$ another special case of Zeno behavior

Outline

Summary

- Definition and examples of hybrid systems
- Hybrid automaton
- Complexity issues: modeling power vs decision power
- Zeno behavior