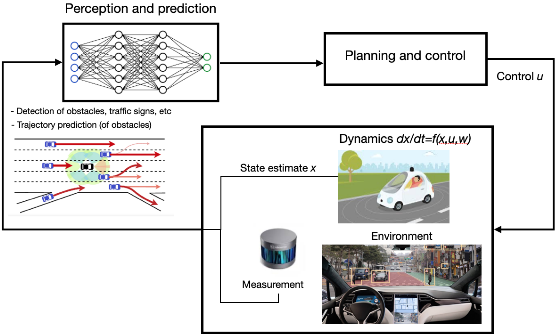
CTRLVERIF. Analysis of control systems

Lecture 3. Reachability and stability analysis of controlled and hybrid systems

Eric Goubault and Sylvie Putot

MPRI

Remember: the typical control loop

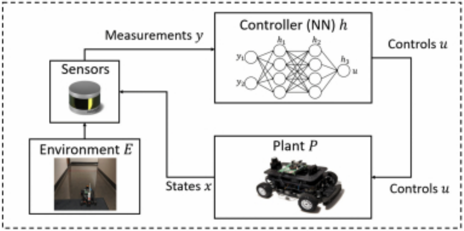

Perception and prediction
Planning and control
Control *u*
- Detection of obstacles, traffic signs, etc
- Trajectory prediction (of obstacles)
Dynamics *dx/dt=f(x,u,w)*
State estimate *x*
Environment
Measurement

- Up to now: abstraction-based verification of open loop (program, neural network)
- Today and later: abstraction-based verification of the closed loop
  - finite-horizon properties: reach-avoid properties, (robust) control problems
    - **Safety verification:** verify the system cannot enter into an unsafe region
    - **Validation of control strategies:** verify the system trajectories satisfy some properties such as staying within a maximal distance from a reference trajectory or reaching a setpoint/target
    - **Controller synthesis:** finding some parameter sets of controllers that satisfy safety or performance constraints
  - finite or infinite-horizon properties: incremental stability, invariant sets, etc

The closed-loop: a neural-network controlled system

Given

- plant dynamic f,
- state x, control u, disturbance $w \in \mathcal{W}$
- controller h
- control period Δt_u



Time-triggered (u computed every $\Delta_u t$) dynamical system with non-linear feedback:

$$
\begin{aligned}
\dot{x}(t) &= f(x(t), u(t), w(t)) \\
x(t_0) &= x_0 \in \mathcal{X}_0 \\
u(t) &= u_k = h(y(x(\tau_k))), \text{ for } t \in [\tau_k, \tau_{k+1}), \text{ with } \tau_k = t_0 + k\Delta t_u, \ \forall k \geq 0
\end{aligned}
$$

A particular case of a hybrid system

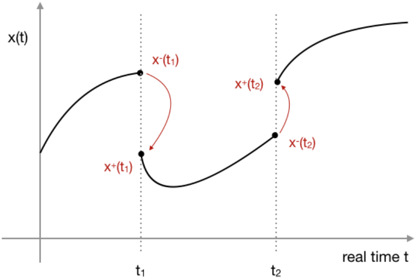
Outline

- A quick detour by hybrid systems
- Bounded time properties: (forward) reachability analyis
  - continuous dynamics with affine vector fields
  - non linear vector fields: Taylor methods
  - neural network controlled systems
- Alternatives to reachability analysis
  - Unbounded time properties: stability analysis and barrier functions
  - Online verification: monitoring

From continuous to hybrid systems, informally

Simple hybrid system:

- smooth dynamics almost all the time, except for state jumps $x^+ = g(x^-)$ at some discrete *t*.
- transitions can be time-dependent or state-dependent
- possibly several smooth dynamics



Examples of hybrid behavior

"Naturally" hybrid dynamics: as a simplification of a nonlinear model

- In mechanical systems, continuous motion may be interrupted by collisions
- Systems with different phases: walking robots, biological cell growth and division
- In electrical circuits, continuous phenomena such as capacitors charging, interrupted by switches opening and closing, or diodes going on or off.

Continuous systems controlled by discrete logics

- Logic-based switching: control modes in transportation systems (aircraft autopilot modes for example)
- Finite input and observation sets: in chemical process control the continuous evolution of chemical reactions is controlled by valves and pumps

Coordinating processes (e.g. automated highway)

For all these systems: convenient model as instantaneous discrete changes within continuous components

Models of hybrid systems

Control theory perspective

- Logic-based switching between dynamics, no state jumps
- Modeling result : switched control systems
- Analysis: focus on stability and invariance properties

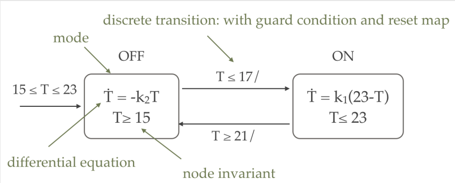
Computer science perspective

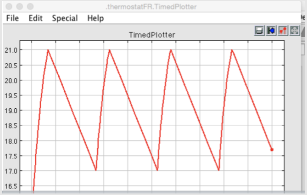
- More expressivity allowed in discrete transitions
- Modeling result : hybrid automata, an extension of timed automata
- Analysis: focus on reachability analysis

Hybrid Automata: the most classical model for hybrid systems

Example (Self-regulating switching thermostat with hysteresis)

- State machine with continuous state variable *T*
- Time progresses within modes (ON/OFF) and *T* changes continuously according to differential equations
- Transitions between modes are instantaneous and enabled by the satisfaction of guards on *T*; *T* can be discontinuously updated during mode-switches
- Invariants constrain how long the system can stay in a discrete mode

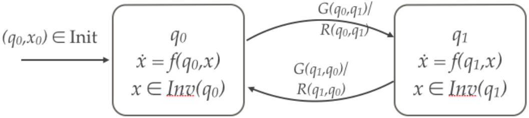




Hybrid Automaton: notations

A Hybrid Automaton is a collection $HA = (Q, X, f, Init, Inv, E, G, R)$, with

- $Q = \{q_1, \ldots, q_N\}$: finite set of discrete states or nodes
- $X = \mathbb{R}^n$: set of continuous states
- $f(.,.) : Q \times X \to X$: vector fields
- $Init \subseteq Q \times X$: set of initial states
- $Inv(.) : Q \to \mathcal{P}(X)$: invariants or domain ($\mathcal{P}(X)$ denoting the power set (set of all subsets) of X)
- $E \subseteq Q \times Q$: set of edges or transitions
- $G(.) : E \to \mathcal{P}(X)$: guard conditions
- $R(.) : E \to \mathcal{P}(X \times X)$: reset maps



Execution of a Hybrid Automaton

Hybrid time trajectory $\tau = \{I_i\}_i, i = 0, \ldots, N$

A finite or infinite sequence of intervals of reals:

- $I_i = [\tau_i, \tau_{i+1}]$ with $\tau_i \leq \tau_{i+1}$ (τ_i are the times of discrete transitions)
- if the sequence is finite, $N < \infty$, $I_N = [\tau_N, \tau_{N+1}]$ or $I_N = [\tau_N, \infty[$

An execution of a HA is defined by (τ, q, x)**:**

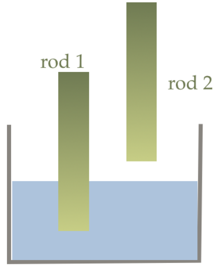
- τ a hybrid time trajectory
- a sequence of discrete-time states $q = \{q_i\}_i, i = 0, \ldots, N$ where $q_i : I_i \rightarrow Q$ is constant on I_i
- a sequence of continuous-time states $x = \{x_i\}_i$ with $xi : I_i \rightarrow X$

- Initial condition $(q(\tau_0), x(\tau_0)) \in Init$
- Continuous evolution for all i
  - x_i is solution to $\dot{x}(t) = f(q_i(t), x(t))$ on I_i with initial condition $x_i(\tau_i)$ at τ_i,
  - for all $t \in [\tau_i, \tau_{i+1}[$, the mode invariant must hold: $x_i(t) \in Inv(q_i(t))$
- Discrete transition for all i, $e = (q_i(\tau_i + 1), q_{i+1}(\tau_{i+1})) \in E$
  - enabled when the guard of the transition is satisfied $x_i(\tau_{i+1}) \in G(e)$
  - continuous state then jumps from x_i to $x_{i+1} : (x_i(\tau_{i+1}), x_{i+1}(\tau_{i+1})) \in R(e)$.

Example: nuclear reactor controller

Plant behavior

- Without rods: $\dot{T} = 0.1T - 50$ (temperature increases if greater than 500)
- With rod 1: $\dot{T} = 0.1T - 56$ (temperature decreases if lower than 560)
- With rod 2: $\dot{T} = 0.1T - 60$ (temperature decreases if lower than 600)
- Rods 1 and 2 cannot be used simultaneously
- Once a rod is removed, you cannot use it for 10 minutes



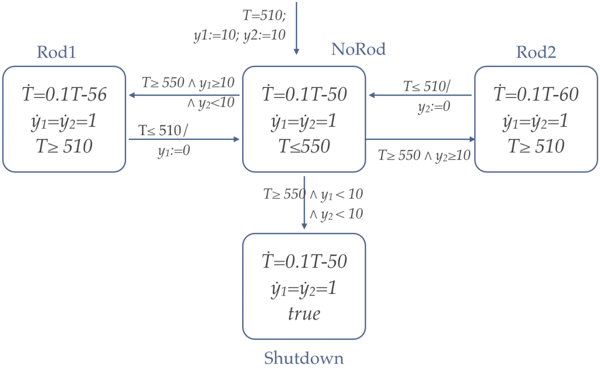
Controller specification

- Keep temperature between 510 and 550 degrees.
- If T=550 then either a rod is available or we shutdown the plant.

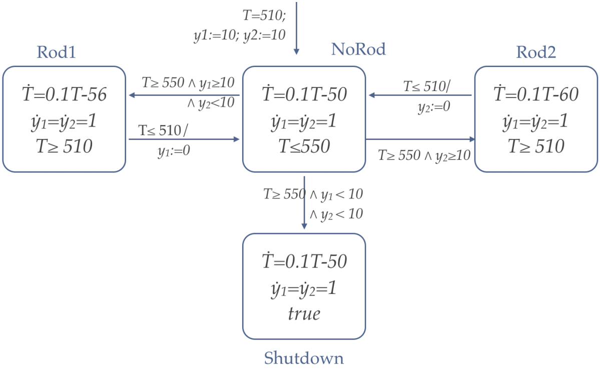
Exercise: write the corresponding hybrid automaton

Once the automaton is built, verification will aim at proving for instance that shutdown is not reachable.

Example: nuclear reactor controller


T=510;
$y1$:=10; $y2$:=10
Rod1
$\dot{T}$=0.1T-56
$\dot{y}_1$=$\dot{y}_2$=1
$T \geq 510$
NoRod
$\dot{T}$=0.1T-50
$\dot{y}_1$=$\dot{y}_2$=1
$T \leq 550$
Rod2
$\dot{T}$=0.1T-60
$\dot{y}_1$=$\dot{y}_2$=1
$T \geq 510$
$T \geq 550 \wedge y_1 \geq 10 \wedge y_2 < 10$
T$\leq$ 510/ y_1:=0
$T \leq 510$/ y_2:=0
$T \geq 550 \wedge y_2 \geq 10$
$T \geq 550 \wedge y_1 < 10 \wedge y_2 < 10$
$\dot{T}$=0.1T-50
$\dot{y}_1$=$\dot{y}_2$=1
true
Shutdown

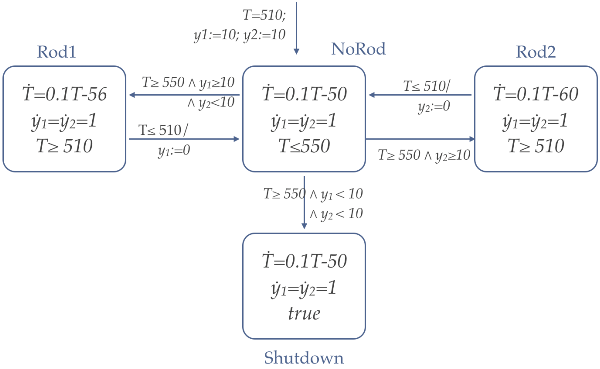
Example: nuclear reactor controller



Is shutdown reachable ?

Algorithmic verification: NO

Example: nuclear reactor controller



Is shutdown reachable ?

Algorithmic verification: NO

Parametric specification (synthesis)

For which temperature intervals and delay between rods is the system (un)safe ?

Continuous dynamics: well-posedness of ODE system

For $f : D \subset \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n$ where D is an open set of $\mathbb{R} \times \mathbb{R}^n$, existence and uniqueness of solutions to the Initial Value Problem ?

$$\begin{aligned} \dot{x}(t) &= f(t, x(t)) \\ x(t_0) &= x_0 \end{aligned}$$

Picard–Lindelöf / Cauchy-Lipschitz Theorem (existence and uniqueness)

Suppose f is uniformly Lipschitz continuous in x

$$\exists K \geq 0, \forall t, \|f(t, x(t)) - f(t, y(t))\| \leq K\|x - y\|$$

and continuous in t. Then, for some value $\epsilon > 0$, there exists a unique maximal solution $x(t)$ to the Initial Value Problem on $[t_0 - \epsilon, t_0 + \epsilon]$.

Peano existence Theorem

If f is only continuous, then existence is ensured, but not uniqueness.

Example. $\dot{x} = x^{1/3}, x(0) = 0$ has 2 solutions: $x_1(t) = 0$ and $x_2(t) = (2t/3)^{3/2}$.

Hybrid systems: generalization of the notion of solution

Given an input signal $u : [0, \infty[\to \mathbb{R}^m$, then $x : [0, \infty[\to \mathbb{R}^n$ piecewise differentiable and such that

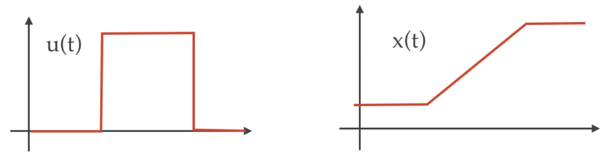
$$x(t) = x(0) + \int_0^t f(x(\tau), u(\tau)) d\tau, \forall t \geq 0$$

is a weak solution, in the sense of Carathéodory, to $\dot{x} = f(x, u)$. (existence not ensured)

If x is a solution then $\dot{x}(t) = f(x(t), u(t))$ holds at any time t for which the derivative exists.

Example

$\dot{x}(t) = u(t)$



What can go wrong?

Existence non ensured when f is not continuous

Example.

$$
\begin{aligned}
\dot{x} = f(x) &= -1 \text{ if } x \geq 0 \\
&= 1 \text{ if } x < 0
\end{aligned}
$$

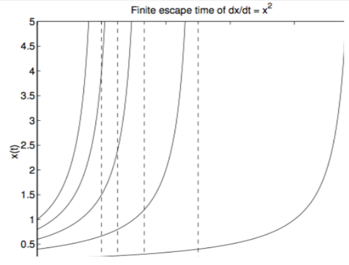
There is no solution to this differential equation that starts with $x(0) = 0$: on any time interval $[0, \epsilon[$, x cannot remain 0, nor become negative, nor become positive. (we will discuss later Filippov's solutions)

Possible finite escape time when f is not globally Lipschitz on all $\mathbb{R}$

Example. $\dot{x} = x^2, x(0) = x_0$ has the solution

$$
x(t) = \frac{x_0}{1 - x_0 t}, \quad 0 \leq t < t_f = \frac{1}{x_0}
$$

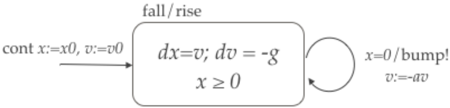
and finite escape time t_f.



A classical example: the bouncing ball

Hybrid systems are useful to model also purely physical phenomena such as collisions (and not only interaction between controller and physical world)

- Ball dropped from initial height x_0 with initial vertical velocity v_0
- Dynamics subject to $\dot{x}(t) = v, \dot{v}(t) = -g$
- When the ball hits the ground ($x = 0$), velocity changes discretely: $v := -a.v$, with $0 < a < 1$ dampening constant



fall/rise

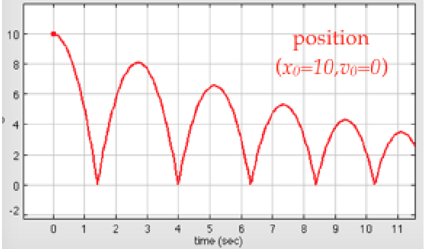
cont $x := x0,\ v := v0$

$dx = v;\ dv = -g$

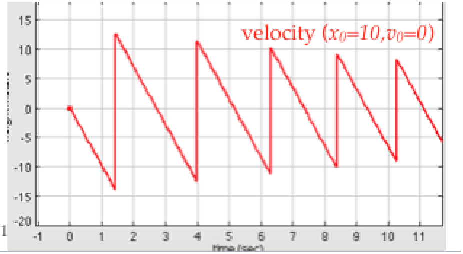
$x \geq 0$

$x = 0$/bump!

$v := -av$



position ($x_0 = 10, v_0 = 0$)



velocity ($x_0 = 10, v_0 = 0$)

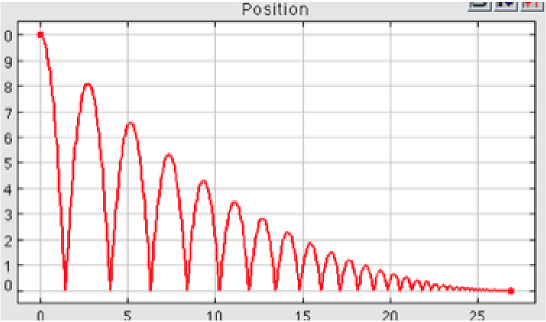
Analysis of the behavior

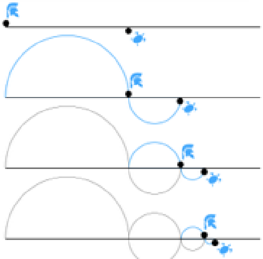
- state just before first bump: $(v(t) = -gt, x(t) = x_0 - gt^2/2 = 0)$
- 1st bump occurs at $t_1 = \sqrt{(2x_0/g)}$, with $v_1(t_1) = -\sqrt{(2x_0 g)}$
- just after 1st discrete bump: $v_2 = -a.v_1 = a.\sqrt{(2x_0 g)}$
- evolution after discrete bump (reinitializing time to 0): $v(t) = v_2 - gt, x(t) = v_2 t - gt^2/2$
- time between 1st and 2nd bump: $\Delta t_2 = 2v_2/g$, velocity just before 2nd jump (at time $t_1 + \Delta t_2$): $-v_2$
- …velocity after k bumps $v_k = a^k.v_1$, sum of durations $\sqrt{(2x_0/g)} + (2v_1/g)\sum_k a^k$: sequence has finite limit $\sqrt{(2x_0/g)} + 2v_1/(g(1-a))$

Bouncing ball: Zeno behavior

Zeno of Elea 490 B.C

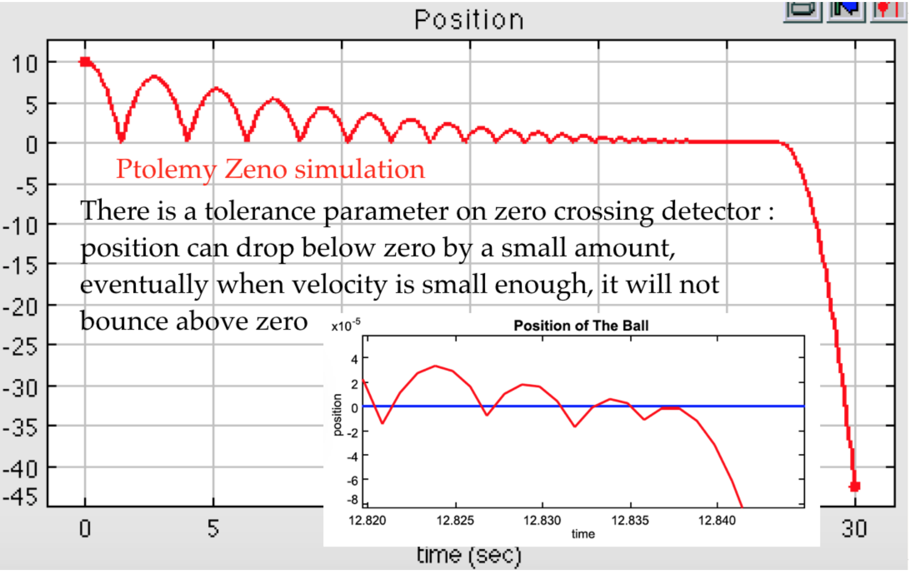
- Physical interpretation: ball is at rest within finite time interval, but after infinitely many bounces;
- Infinitely many discrete actions in finite time: Zeno behavior
- Set of event time contains a right-accumulation point: the execution does not describe what happens at time K and beyond (stationary ball on the ground).
- Non-blocking and deterministic HA, but no solutions defined on $t \in [0, \infty[$





paradox: Achille never catches up the tortoise?

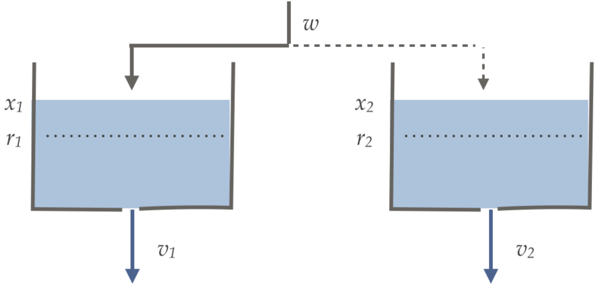
Zeno bouncing ball: simulation in practice



Ptolemy Zeno simulation

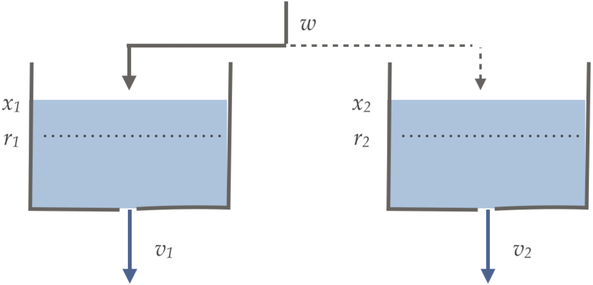
There is a tolerance parameter on zero crossing detector : position can drop below zero by a small amount, eventually when velocity is small enough, it will not bounce above zero

Another classical example: the two-tanks system



- Two tanks: x_i = volume of water in tank i
- Tanks are leaking at constant rate v_i
- Water is added at rate w, either in one or the other tank at a given time
- Objective = keep water levels above r_1 and r_2
- Controller that switches w to tank 1 when $x_1 \leq r_1$ and tank 2 when $x_2 \leq r_2$

The two-tanks system as a hybrid automaton



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

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

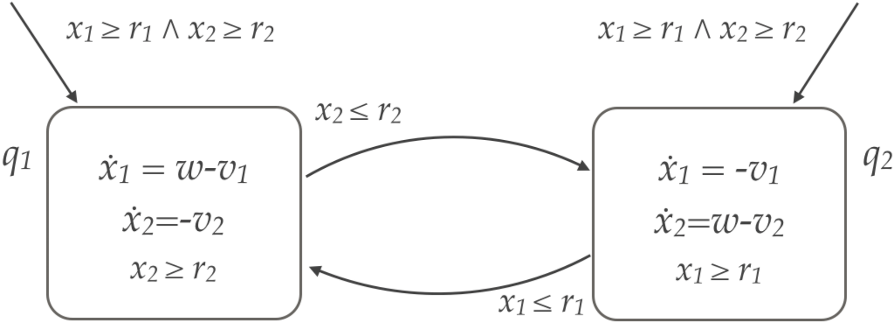
- $\mathit{Init} = \{q_1, q_2\} \times \{(x_1, x_2) | x_1 \geq r_1 \wedge x_2 \geq r_2\}$

The two-tanks system as a hybrid automaton

Write the corresponding HA. Assume $\max(v_1, v_2) < w < v_1 + v_2$ and verify that $\{(q, x) | (x_1 \geq r_1) \wedge (x_2 \geq r_2)\}$ is invariant of the HA.

The two-tanks system as a hybrid automaton

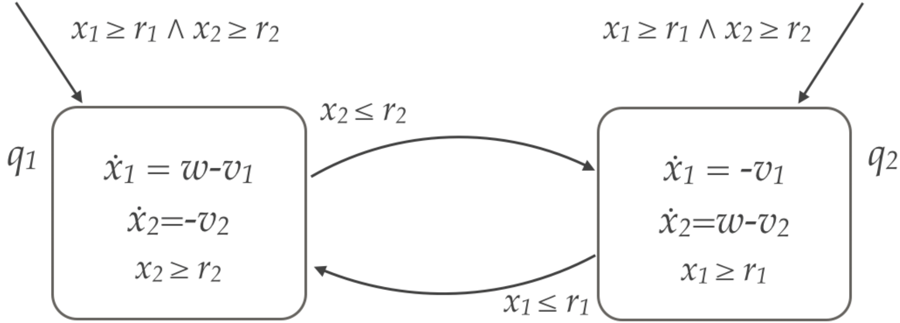
Write the corresponding HA. Assume $\max(v_1, v_2) < w < v_1 + v_2$ and verify that $\{(q, x) | (x_1 \geq r_1) \wedge (x_2 \geq r_2)\}$ is invariant of the HA.



$x_1 \geq r_1 \wedge x_2 \geq r_2$
$x_1 \geq r_1 \wedge x_2 \geq r_2$
$x_2 \leq r_2$
q_1
$\dot{x}_1 = w\text{-}v_1$
$\dot{x}_2 = \text{-}v_2$
$x_2 \geq r_2$
$\dot{x}_1 = \text{-}v_1$
$\dot{x}_2 = w\text{-}v_2$
$x_1 \geq r_1$
q_2
$x_1 \leq r_1$

The two-tanks system as a hybrid automaton

Write the corresponding HA. Assume $\max(v_1, v_2) < w < v_1 + v_2$ and verify that $\{(q, x) | (x_1 \geq r_1) \wedge (x_2 \geq r_2)\}$ is invariant of the HA.



$x_1 \geq r_1 \wedge x_2 \geq r_2$
$x_1 \geq r_1 \wedge x_2 \geq r_2$
$x_2 \leq r_2$
q_1
$\dot{x}_1 = w\text{-}v_1$
$\dot{x}_2 = \text{-}v_2$
$x_2 \geq r_2$
$\dot{x}_1 = \text{-}v_1$
$\dot{x}_2 = w\text{-}v_2$
$x_1 \geq r_1$
q_2
$x_1 \leq r_1$

Automaton is deterministic and non-blocking: there is a unique infinite execution for each initial state.

The two tanks system as a hybrid automaton

But

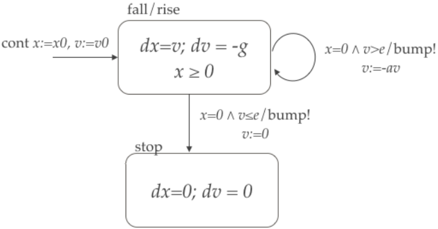
- Condition $\max(v_1, v_2) < w < v_1 + v_2$ implies that less water is added than removed to the system. At least one of the water tanks will eventually drain !
- So why could we prove the invariant ?
- All executions are Zeno: an infinite number of transitions by time $t_{max} = (x_1(0) + x_2(0) - r_1 - r_2)/(v_1 + v_2 - w)$.
- The previous analysis holds along all executions of the system. The problem is that there are no executions defined after the Zeno time, so that the analysis will fail to predict any properties after that time.

Zeno behavior and system modelling

Zeno behavior

- prevents solutions to be globally defined for $t \in [0, \infty[$: simulators get "stuck" and all components are blocked
- physical systems do not exhibit Zeno behaviour, but their abstraction can often lead to Zeno models!
- $\Rightarrow$ Zeno components should be avoided!

Making the bouncing ball non-Zeno



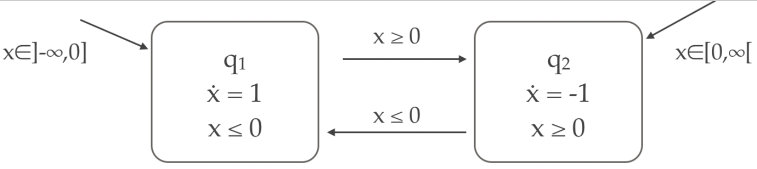
When velocity becomes too small, stop modeling the dynamics accurately

- bounded number of bumps;
- timed actions of arbitrary durations are now possible

Continuation of Zeno executions by extensions: here straightforward by adding regularization parameter *e*

Chattering Zeno (infinitely many switches at a single time instant)

Example



- Accepts a unique infinite execution for all initial states, but all are Zeno: an execution starting from x_0 at time τ_0 reaches 0 in finite time $\tau_0 + x_0$ and takes an infinite number of transitions from then on, without time progress: chattering Zeno
- Can be written compactly as a state-dependent switched system

$$\begin{cases} \dot{x} = & 1 & \text{if } x \leq 0 \\ \dot{x} = & -1 & \text{if } x \geq 0 \end{cases} \quad \text{or even } \dot{x} = -\mathrm{sgn}(x)$$

Switched systems

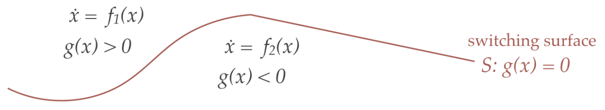
Switched system

$\dot{x} = f_\sigma(x)$, with $\{f_1(x), f_2(x), \ldots, f_p(x)\}$ a family of smooth vector fields from $\mathbb{R}^n$ to $\mathbb{R}^n$.

- Time-dependent switching: switching signal $\sigma : [0, \infty[\to \{1, 2, \ldots, p\}$ piecewise constant function of time
- State-dependent switching $\sigma(x(t))$

In particular, compared to general hybrid systems, no resets : state variable x evolves continuously, only its derivative may be discontinuous

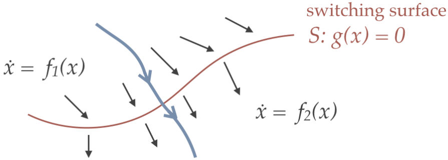
State-dependent switching



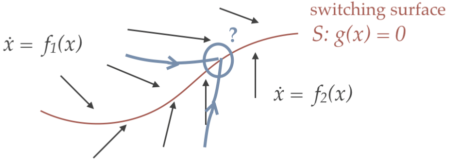
State-dependent switched systems

Two cases

- transversal crossing: when the trajectory hits S, it crosses the surface



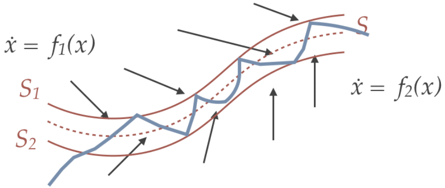
- (attractive) sliding mode configuration: both fields point "inwards"



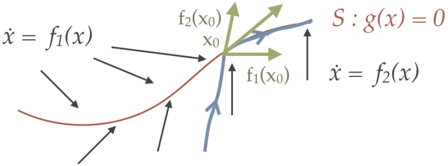
Hysteresis switching and sliding mode

Two solutions:

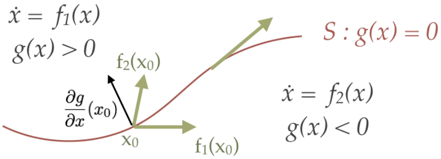
- relaxation by spatial or time hysteresis switching



- sliding mode control : Filippov's definition as convex combination $\dot{x} = \lambda f_1(x) + (1 - \lambda) f_2(x)$ with λ such that x "slides" along $g(x) = 0$: third mode.



Sliding mode



- Sliding occurs if

$$\frac{\partial g}{\partial x}.f_1 < 0 \text{ and } \frac{\partial g}{\partial x}.f_2 > 0$$

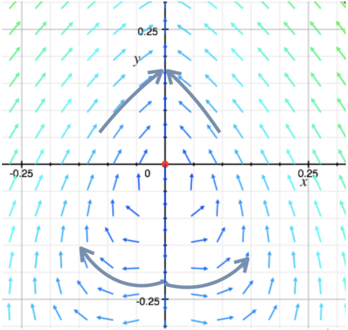
- Derivative of g in direction f is called Lie derivative and noted $L_f g = (\partial g/\partial x).f$: sliding occurs when $L_{f_1} g < 0$ and $L_{f_2} g > 0$
- Filippov solution of $\dot{x} = f = \lambda f_1 + (1-\lambda) f_2$ staying on $g(x) = 0$ means

$$\frac{dg(x)}{dt} = \frac{\partial g}{\partial x}\dot{x} = \frac{\partial g}{\partial x}(\lambda f_1 + (1-\lambda) f_2) = \lambda L_{f_1} g + (1-\lambda) L_{f_2} g = L_f g = 0$$

$$\Rightarrow \dot{x} = f = \frac{1}{L_{f_2} g - L_{f_1} g}(L_{f_2} g\, f_1 - L_{f_1} g\, f_2)$$

Example: piecewise linear system

$$\begin{cases} \dot{x}_1 = & -2x_1 - 2x_2\mathrm{sgn}(x_1) \\ \dot{x}_2 = & x_2 + 4x_1\mathrm{sgn}(x_1) \end{cases}$$



or equivalently

$$f_1(x) = \begin{cases} \dot{x}_1 = & -2x_1 + 2x_2 \\ \dot{x}_2 = & x_2 - 4x_1 \end{cases} \quad \text{for } x_1 < 0$$

$$f_2(x) = \begin{cases} \dot{x}_1 = & -2x_1 - 2x_2 \\ \dot{x}_2 = & x_2 + 4x_1 \end{cases} \quad \text{for } x_1 > 0$$

Exercice: can you find a sliding mode ? (define the corresponding sliding dynamics)

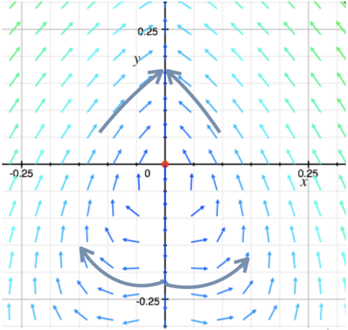
Example: piecewise linear system

$$\begin{cases} \dot{x}_1 = & -2x_1 - 2x_2\mathrm{sgn}(x_1) \\ \dot{x}_2 = & x_2 + 4x_1\mathrm{sgn}(x_1) \end{cases}$$

or equivalently

$$f_1(x) = \begin{cases} \dot{x}_1 = & -2x_1 + 2x_2 \\ \dot{x}_2 = & x_2 - 4x_1 \end{cases} \quad \text{for } x_1 < 0$$

$$f_2(x) = \begin{cases} \dot{x}_1 = & -2x_1 - 2x_2 \\ \dot{x}_2 = & x_2 + 4x_1 \end{cases} \quad \text{for } x_1 > 0$$



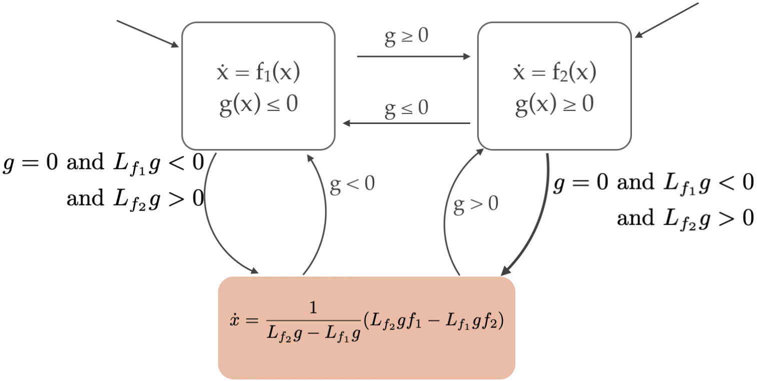
Chattering Zeno on $\{x_1 = 0, x_2 > 0\}$

- let $g(x) = x_1 = 0$, then $L_{f_2}g = -2x_2 < 0$ and $L_{f_1}g = 2x_2 > 0$

Filippov solutions on $\{x_1 = 0, x_2 \geq 0\}$ satisfy $\dot{x} \in \lambda f_1(x) + (1-\lambda)f_2(x)$ for some $\lambda \in [0, 1]$

- $x(t)$ staying on $S : \{x_1 = 0\}$ means $\dot{x}_1 = 0$, i.e. $\lambda(2x_2) + (1-\lambda)(-2x_2) = 2x_2(2\lambda - 1) = 0$
- the solution is given by $\lambda = 0.5$, resulting in the sliding dynamics $\dot{x}_1 = 0, \dot{x}_2 = x_2$

Hybrid automaton: regularization of chattering Zeno



$\dot{x} = f_1(x)$
$g(x) \leq 0$

$g \geq 0$

$g \leq 0$

$\dot{x} = f_2(x)$
$g(x) \geq 0$

$g = 0$ and $L_{f_1}g < 0$
and $L_{f_2}g > 0$

$g < 0$

$g > 0$

$g = 0$ and $L_{f_1}g < 0$
and $L_{f_2}g > 0$

$$\dot{x} = \frac{1}{L_{f_2}g - L_{f_1}g}(L_{f_2}gf_1 - L_{f_1}gf_2)$$

Outline

- A quick detour by hybrid systems
- Bounded time properties: (forward) reachability analyis
  - continuous dynamics with affine vector fields
  - non linear vector fields: Taylor methods
  - neural network controlled systems
- Alternatives to reachability analysis
  - Unbounded time properties: stability analysis and barrier functions
  - Online verification: monitoring

Forward reachability for hybrid systems

Safety verification based on forward reachability: given an unsafe set of states U, fixpoint computation

```
$R := Init$;
while (true do) {
  if ($R \cap U \neq \emptyset$) return UNSAFE;
  if ($Reach_D(R) \cup Reach_C(R) \subseteq R$) return SAFE; // R reachable states
  $R := R \cup Reach_D(R) \cup Reach_C(R)$;
}
```

In general termination not guaranteed (often bounded-time reachability)

- Continuous steps Given a dynamical system defined by $\dot{x} = f(x), x \in X$, and given a set $R \subseteq X$, over-approximate the set $Reach_C(R) \subseteq X$ of points reached by trajectories (solutions) starting in R.
- Discrete steps Given a discrete transition of the hybrid system and $R \subseteq X$, over-approximate the set $Reach_D(R) \subseteq X$ of points in reachable by taking the discrete transition starting in R.

We focus on continuous steps.

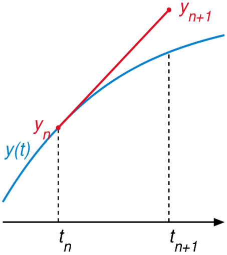
Simulation: relies on numerical integration

Simulation tools compute a numerical approximation to the solution of the Initial Value Problem $\dot{x}(t) = f(x,t),\ x(t_0) = x_0$:

- approximation of the discretized solution $x(k\Delta t)$ of $x(t) = x_0 + \int_0^t f(x,t)dt$
- the simplest (and least accurate) is the forward (or explicit) Euler method

$$\dot{x} \approx \frac{x[k+1] - x[k]}{\Delta t}$$

$$x((k+1)\Delta t) = x[k+1] = x[k] + \Delta t f(x[k], k\Delta t)$$



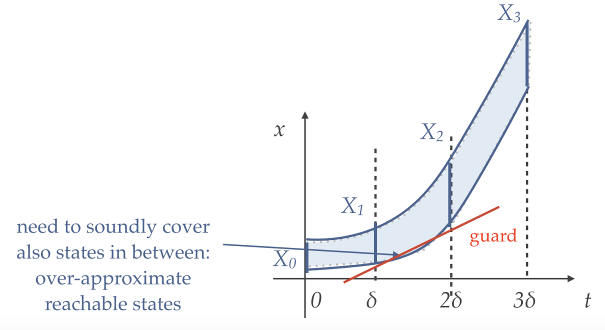
Among the most widely used integration schemes are Runge-Kutta methods

- Runge-Kutta methods estimate the Taylor series expansion of $x(t)$ by using quadrature formulas involving first-order derivatives within the intervals
- Higher order quadrature formulas are used to estimate the approximation error and adapt the integration step Δt to keep the error estimate below a tolerance

From numerical simulation to reachability analysis

Time discretization based numerical integration can be extended to initial condition x_0 given in a set X_0 and uncertain parameters, BUT:

- this is unsound if the approximation errors are not computed
- one can miss discrete transitions (the approximation may no longer stay in a neighborhood of the solution) or unsafe states



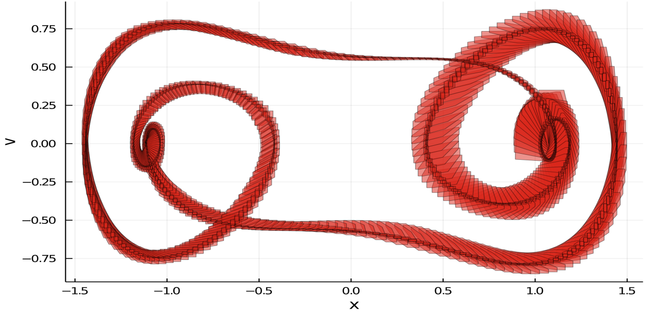
need to soundly cover also states in between: over-approximate reachable states

- if computed without care, often very conservative ("wrapping effect" of interval arithmetic)

Reachability analysis

Scalable "flowpipes" for discrete-time or continuous-time dynamical system:

- from a set of initial states I
- subject to a set of admissible inputs $u(t) \in U$ or disturbances $w(t) \in W$



Example: Duffing oscillator from the JuliaReach library (`https://github.com/JuliaReach/`)

Piecewise affine dynamics (no inputs)

Linear dynamical systems

$$\dot{x}(t) = Ax(t),\ x_0 \in X \subseteq \mathbb{R}^n$$

- Trajectories are exponential functions: closed form solution

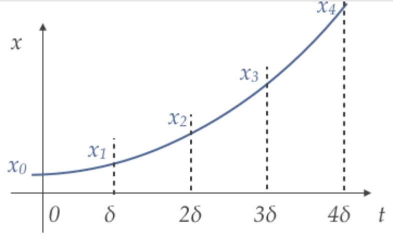
$$x(t) = e^{At}x(0),\ e^{At} = 1 + At + \frac{(At)^2}{2!} + \frac{(At)^3}{3!} + \dots$$

- satisfiability not known for first-order logic with exponentiation

Time discretization of exact computation $x(t_k)$ with $t_k = \delta k$

Solution $x(\delta(k+1)) = e^{A\delta}e^{A\delta k}x(0) = e^{A\delta}x(\delta k)$ computed by linear transforms:

- choose a time step δ
- compute : $x_0 = x(0),\ x_{k+1} = e^{A\delta}x_k$ (multiplication by a constant matrix)



From time discretization to reach state: flowpipe approximation

Uncertain linear system (bounded inputs u):

$$\dot{x}(t) = Ax(t) + u(t),\ \|u(t)\|_\infty = \max_{1\leq i\leq n} |u_i(t)| \leq \mu$$

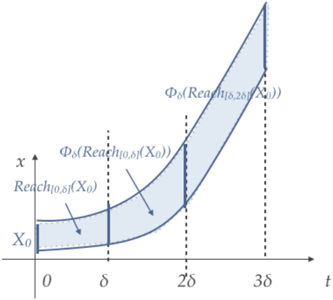
Given a set of initial values X_0, the reachable set at time t is

$$\Phi_t(X_0) = \{x_f \in \mathbb{R}^n |\ \exists x \text{ solution of } \dot{x} = Ax + u, x(0) \in X_0 \wedge x(t) = x_f\}$$

Given a time step δ and $T = N\delta$,

$$Reach_{[0,T]}(X_0) = \bigcup_{k=0}^{N-1} Reach_{[k\delta,(k+1)\delta]}(X_0)$$

$$Reach_{[k\delta,(k+1)\delta]}(X_0) = \Phi_\delta(Reach_{[(k-1)\delta,k\delta]}(X_0))$$


$\Phi_\delta(Reach_{[\delta,2\delta]}(X_0))$
$\Phi_\delta(Reach_{[0,\delta]}(X_0))$
x
$Reach_{[0,\delta]}(X_0)$
X_0
0
δ
2δ
3δ
t

Need conservative approximations of $Reach[0, \delta](X_0)$ and Φ_δ: for linear system $\dot{x}(t) = Ax(t) + u(t)$, $\Phi_\delta(X) = \Phi_\delta X + V$, with $\Phi_\delta = e^{\delta A}$

Affine dynamics with inputs: over-approx of $\mathit{Reach}[0,\delta](X_0)$ and Φ_δ

$$\dot{x}(t) = Ax(t) + u(t),\ \ \|u(t)\|_\infty = \max_{1\le i\le n} |u_i(t)| \le \mu$$

Solution is:

$$x(t) = e^{At}x(0) + \int_0^t e^{A(t-\tau)}u(\tau)d\tau$$

- Inputs influence can be over-approximated with a box of radius β_t :

$$\| \textstyle\int_0^t e^{A(t-\tau)}u(\tau)d\tau\|_\infty \le \mu \displaystyle\int_0^t \|e^{A(t-\tau)}\|_\infty d\tau = \mu \frac{e^{\|A\|t} - 1}{\|A\|} = \beta_t$$

$$\Phi_\delta(Z) \subseteq e^{\delta A}Z + \square(\beta_\delta)$$

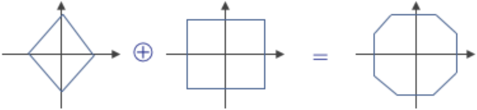
- Need a conservative approximation of $\mathit{Reach}_{[0,\delta]}(X_0)$:

$$\mathit{Reach}[0,\delta](X_0) \subseteq (\bigcup_{t\in[0,\delta]} e^{tA}X_0) + \square(\beta_\delta) \subseteq Q_0$$

Reachability algorithm:

```
Choose $\delta$, $T = N\delta$
$R_0 = Q_0 \supseteq \mathit{Reach}_{[0,\delta]}(X_0)$
$Q_{i+1} = e^{\delta A}Q_i \oplus \square_\delta$
$R_{i+1} = R_i \cup Q_{i+1}$
$\mathit{Reach}_{[0,T]}(X_0) \subseteq R_N$
```

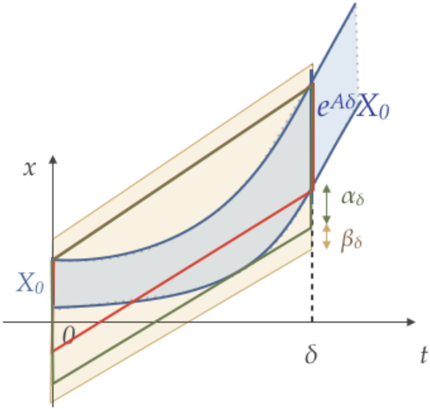
Minkowski sum $\oplus$ = point-wise sum of sets $A \oplus B = \{a + b |\ a \in A \wedge b \in B\}$



Abstraction with zonotopes again: initialization step

Conservative approximation of

$$Reach[0,\delta](X_0) \subseteq \Big(\bigcup_{t\in[0,\delta]} e^{tA} X_0 \Big) + \square(\beta_\delta)$$



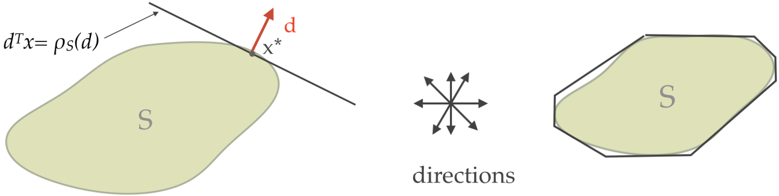
A simple solution :

1. compute a simple zonotope enclosing X_0 and $e^{A\delta}X_0$
2. bloat the zonotope by a box α_δ so that it encloses the reachable set
3. bloat the zonotope by β_δ to take inputs into account

Ref: Reachability of uncertain linear systems using zonotopes, A. Girard, 2005

Support functions / template polyhedra

- For an arbitrary chosen direction/vector d, compute $\rho_S(d) = \max_{x \in S} d^T.x$
- For a given number of directions d_i, set S is over-approximated by a template polyhedra represented as the intersection of half spaces $d^T.x \leq \rho_S(d)$
- lazy representation of any convex set: gives an outer polyhedral approximation that can be refined
- scalable - linear programming for most operations



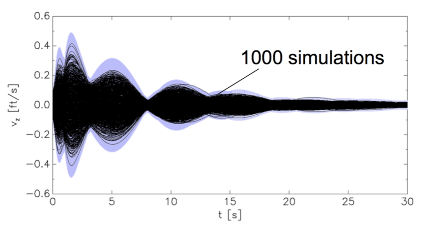
Ref: Reachability analysis of linear systems using support functions, Le Guernic and Girard, 2010

SpaceEx (`http://spaceex.imag.fr`): piecewise affine vector fields

- Different algorithms, main abstractions are zonotopes and support functions
- Writing the model of the system to verify:
  - a graphical model editor to write hybrid systems
  - SL2SX: a translation tool from (a small subset of) Simulink to SpaceEx
- State-of-the-art tool for piecewise affine systems

Example

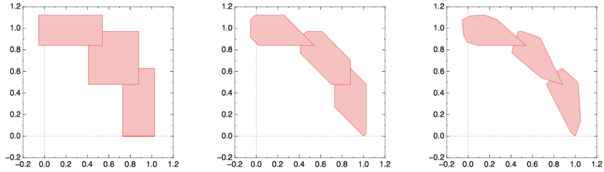
- On 28-dim model of a controlled Westland Lynx helicopter
- 5-10 seconds to compute the following guaranteed enveloppe for 30sec flight



Example with support functions (from SpaceEx documentation)

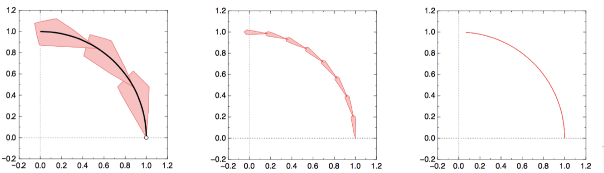
$$\begin{cases} \dot{x} = -y, \dot{y} = x \\ x(0) = 1, y(0) = 0 \end{cases}$$

For different choices of template directions:



(a) box (b) octagonal (c) 16 uniform

For different time steps:



(a) $\delta = 0.5$ (b) $\delta = 0.2$ (c) $\delta = 0.05$

Taylor models to abstract nonlinear dynamics

- Taylor model: a pair $T = (P, I)$ of a multivariate polynomial P of order n, and a remainder interval I, defined over an interval domain D.
  - Taylor model arithmetic: combine interval arithmetic and symbolic computations

    $$T_1 + T_2 = (P_1 + P_2, I_1 + 1_2)$$

    $$T_1.T_2 = (P_{1.2}, I_{1.2})$$

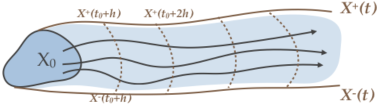
    where $P_{1.2}$ is the part of the polynomial $P_1.P_2$ up to order n and $I_{1.2} = B(P_e) + B(P_1).I_2 + B(P_2).I_1 + I_1.I_2$ where P_e is the part of the polynomial $P_1.P_2$ of orders $(n+1)$ to $2n$, and $B(P)$ denotes a bound of P on the domain D
- Initially introduced by Berz, Hoefkens and Makino 1997 then Nedialkov, Neher, Tucker, Wittig
  - for a problem of nonlinear dynamics in beam physics
  - other applications such as near-earth encounter of the Apophis asteroid
  - can be very accurate with high order models (and small uncertainties)
- In the context of reachability analysis for verification
  - guaranteed ODE integration with large uncertainties on initial values, inputs and params
  - moderate order (3-4) in time, low order in space
  - in what follows: Taylor model in time, intervals/affine forms/zonotopes in space

Ref: *From Taylor series to Taylor models, M Berz, 1997.*

On Taylor model based integration of ODEs, Neher, Jackson, and Nedialkov, 2007

Taylor expansions in time for nonlinear ODEs reachability I

Compute tubes of trajectories $[x](t)$, or flowpipes, guaranteed to enclose all trajectories of system $\dot{x}(t) = f(x,t), \; x(t_0) \in [x_0]$



For $f \in C^k$, over-approximate the solution of $\dot{x}(t) = f(x(t)), \; x(t_0) \in [x_0]$ on $[t_0, T]$:

- Time grid $t_0 < t_1 < \ldots < t_N = T$
- Taylor-Lagrange expansion in t of the solution on each time slice $[t_j, t_{j+1}]$

$$x(t) = x(t_j) + \sum_{i=1}^{k-1} \frac{(t-t_j)^i}{i!} \frac{d^i x}{dt^i}(t_j) + \frac{(t-t_j)^k}{k!} \frac{d^k x}{dt^k}(t_j + \theta h), \; 0 < \theta < 1$$

- initial value $x(t_j) \in [x_j]$ given by Taylor expansion on previous time slice $[t_{j-1}, t_j]$, evaluated at time t_j
- coefficients of the expansion computed using that x is solution of $\dot{x}(t) = f(x,t)$

Set-valued computations: evaluation with intervals, *affine forms / zonotopes*, etc.

Taylor expansion based method for nonlinear ODE reachability

- Taylor-Lagrange expansion of the exact solution, valid for $t \in [t_j, t_j + h]$:

$$x(t) = x(t_j) + \sum_{i=1}^{k-1} \frac{(t-t_j)^i}{i!} \frac{d^i x}{dt^i}(t_j) + \frac{(t-t_j)^k}{k!} \frac{d^k x}{dt^k}(t_j + \theta h),\ 0 < \theta < 1$$

  with initial value given by Taylor expansion on previous time slice $[t_{j-1}, t_j]$, evaluated at time t_j: $x(t_j) \in [x_j] = [x]_{j-1}([x_{j-1}], t_j)$
- Coefficients are computed using that x is solution of $\dot{x}(t) = f(x,t)$:

$$\frac{dx}{dt}([x_j], t_j) = L_f^1([x_j], t_j) = \{f(x_j, t_j),\ x_j \in [x_j]\}$$

$$\frac{d^2x}{dt^2}([x_j], t_j) = L_f^2([x_j], t_j) = \{\frac{d}{dt}(f(x_j, t_j)),\ x_j \in [x_j]\} = \{\langle \frac{\partial f}{\partial x}.f\rangle(x_j, t_j),\ x_j \in [x_j]\}$$

$$\frac{d^ix}{dt^i}([x_j], t_j) = L_f^i([x_j], t_j) = \{\langle \frac{\partial L_f^{i-1}}{\partial x}.f\rangle(x_j, t_j),\ x_j \in [x_j]\}$$

  Defined inductively, can be computed efficiently by automatic differentiation

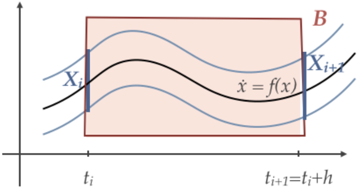
$$[\boldsymbol{x}]_j([\boldsymbol{x}_j], t) = [\boldsymbol{x}_j] + \sum_{i=1}^{k-1} \frac{(t-t_j)^i}{i!} f^{[i]}([\boldsymbol{x}_j], t_j) + \frac{(t-t_j)^k}{k!} f^{[k]}([\boldsymbol{r}_{j+1}], [t_j, t_{j+1}]),$$

- Last coefficient a priori unknown: we need an a priori enclosure $[\boldsymbol{r}_{j+1}]$ of $x(t), t \in]t_j, t_j + h[$ (Lohner's method, based on Picard iteration).
- Initialization of next iterate $[\boldsymbol{x}_{j+1}] = [\boldsymbol{x}]_j([\boldsymbol{x}_j], t_{j+1})$

Taylor-based methods are 2-steps methods on each time step

For each time step $[t_j, t_j + h]$ of the integration, 2 steps:

1. Existence proof and a priori enclosure: given $x(t_j) \in [x_j]$, find a step size h and a coarse box/interval enclosure B (noted $[r_{j+1}]$ on the previous slide) such that for all $t \in [t_j, t_j + h]$, the solution exists and satisfies $x(t) \in B$
2. Tightening: using B to compute the last term of the Taylor expansion, compute a tight enclosure $[x_{j+1}]$ for $x(t)$ at $t_{j+1} = t_j + h$



Step 1: bounding box

Theorem (Lohner 88)

Let $\dot{x} = f(x)$, $x(t_j) \in [x_j]$, and B^0 be an initial guess for the enclosure of x on $[t_j, t_j + h]$. If

$$B^1 := [x_j] + [0, h].f(B^0) \subseteq B^0$$

then the above initial value problem has exactly one solution over $[t_j, t_j + h]$, which lies entirely within the B^1 bounding box.

This comes from (an interval version of) the Picard-Lindelöf iteration: for f Lipschitz-continuous in x, the Banach fixed-point theorem applied to the integral equation (the integral operator is a contraction for f Lipschitz)

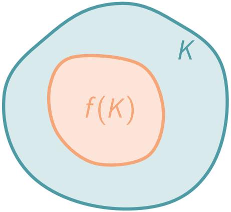
$$x(t) = x_0 + \int_0^t f(x(s))ds$$

proves the existence and uniqueness of solution of $\dot{x} = f(x), x(0) = x_0$, which can be obtained as limit of the sequence of iterates

$$x_0(t) = x_0,\ x_{i+1}(t) = x_0 + \int_0^t f(x_i(s))ds$$

Related to Brouwer-Schauder theorem

A continuous function f mapping a nonempty compact convex set K of a Euclidean space to itself has a fixed point on K.



Applies in particular to intervals.

One of the first emblematic results in algebraic topology (see 2nd part of the course for more refined fixpoint theorems)

Algorithm for Taylor model construction

Note that if $[x_j] \subseteq B^0$, we can always find h such that $[x_j] + [0,h].f(B^0) \subseteq B^0$ holds.

Step1

Compute at the same time a priori bounding box and valid step size (some version of the interval Picard iteration)

```
Input: $[x_j], h, a$ ;
Init: $B := [x_j] + [0,h].f([x_j])$;
while ($[x_j] + [0,h].f(B) \notin B$) {
 $B := [x_j] + [0,h].f(B)$;
   $B := B + [-a,a]B$;
   $h := h/2$;
}
Output: $h$ and $B$, $\{x(t), \forall t \in [t_j, t_j + h] \in B\}$
```

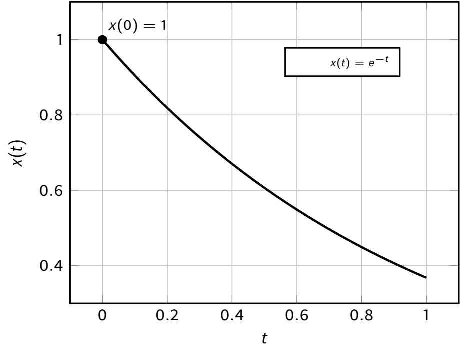
Step2

Using B, compute a tighter enclosure $B \supseteq [x_{j+1}] \ni x(t_{j+1})$ by

$$[x_{j+1}] = [x_j] + \sum_{i=1}^{k-1} \frac{h^i}{i!} L_f^i([x_j]) + \frac{h^k}{k!} L_f^k(B)$$

Example (exercise)

A priori enclosure B and step h for $\dot{x} = -x$, with $x(0) = 1$?



Example (exercise)

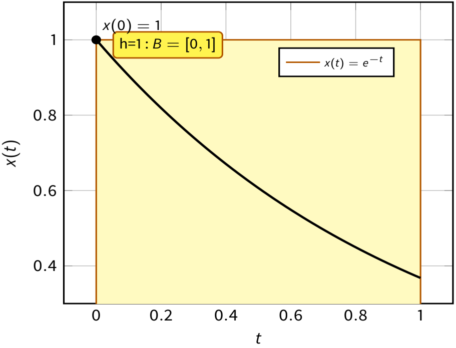
A priori enclosure B and step h for $\dot{x} = -x$, with $x(0) = 1$?

First iterate $B := 1 + [0, h].(-1) = [1 - h, 1]$

$$1 + [0, h].[-1, h - 1] = \begin{cases} 1 + [-h, 0] = [1 - h, 1] \subseteq B & \text{if } h \leq 1 \\ 1 + [-h, h(h - 1)] & \text{otherwise} \end{cases}$$

Thus $\{x(t), t \in [0, h]\} \subseteq [1 - h, 1], \ \forall h \leq 1.$

In particular:
- h=1: $\{x(t), t \in [0, 1]\} \subseteq [0, 1]$.



Example (exercise)

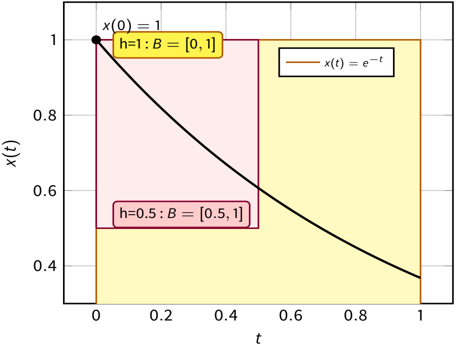
A priori enclosure B and step h for $\dot{x} = -x$, with $x(0) = 1$?

First iterate $B := 1 + [0, h].(-1) = [1 - h, 1]$

$$1 + [0, h].[-1, h - 1] = \begin{cases} 1 + [-h, 0] = [1 - h, 1] \subseteq B & \text{if } h \leq 1 \\ 1 + [-h, h(h - 1)] & \text{otherwise} \end{cases}$$

Thus $\{x(t), t \in [0, h]\} \subseteq [1 - h, 1], \ \forall h \leq 1$.

In particular:
- h=1: $\{x(t), t \in [0, 1]\} \subseteq [0, 1]$.
- h=0.5: $\{x(t), t \in [0, 0.5]\} \subseteq [0.5, 1]$



Example (exercise)

Compute Taylor Model of order 3 valid on t=[0,1] and deduce bounds for $x(1)$

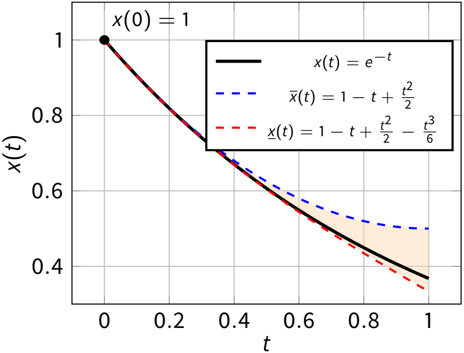
Example (exercise)

Compute Taylor Model of order 3 valid on t=[0,1] and deduce bounds for $x(1)$

$$L_f^n(x) = (-1)^n x$$

$$x(t) = 1 - t + \frac{t^2}{2} - \frac{t^3}{3!}[0,1], \ \forall t \in [0,1]$$

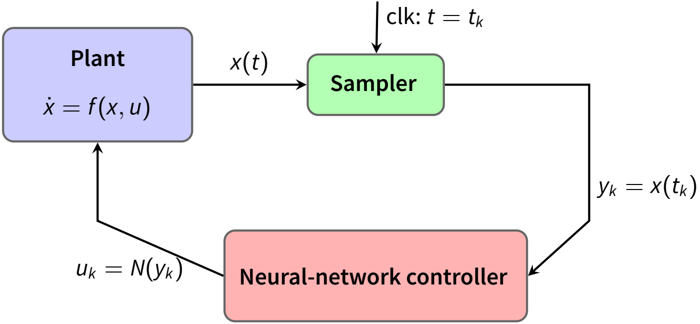
$$x(1) \in [\frac{1}{3}, \frac{1}{2}]$$



Neural network controlled systems

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t), w(t)) \\ x(t_0) &= x_0 \in \mathcal{X}_0 \\ u(t) &= u_k = h(y(x(\tau_k))), \text{ for } t \in [\tau_k, \tau_{k+1}), \text{ with } \tau_k = t_0 + k\Delta t_u, \ \forall k \geq 0 \end{aligned}$$

Taylor model (zonotope coeff)



clk: $t = t_k$

Plant

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

$x(t)$

Sampler

$y_k = x(t_k)$

Neural-network controller

$u_k = N(y_k)$

Zonotope

Outline

- A quick detour by hybrid systems
- Bounded time properties: (forward) reachability analyis
  - continuous dynamics with affine vector fields
  - non linear vector fields: Taylor methods
  - neural network controlled systems
- Alternatives to reachability analysis
  - Unbounded time properties: stability analysis and barrier functions
  - Online verification: monitoring

Stability of dynamical systems

Stability is a key correctness requirement

Small perturbations in the input values should not cause disproportionately large changes in the outputs

Example (cruise controller)

- Safety: Speed should always be within certain threshold values
- Liveness: Actual speed should eventually get close to desired speed
- Stability: If grade of the road changes, speed should change only slowly

Some mathematical formalizations of stability

- Lyapunov stability: with respect to an equilibrium
- Bounded-Input-Bounded-Output (BIBO) stability: with respect to a response
- Input-to-state stability (ISS): unifies Lyapunov stability and BIBO stability (implies BIBO + global asymptotic stability when no input/disturbance)
- Incremental stability: stability of trajectories with respect to one another, rather than with respect to some attractor (stronger property for nonlinear systems)

Stability of systems of differential equations

Consider system $\dot{x} = f(x),\ x(0) = x_0$ where $f : \mathbb{R}^n \to \mathbb{R}^n$ is globally Lipschitz continuous.

Definition

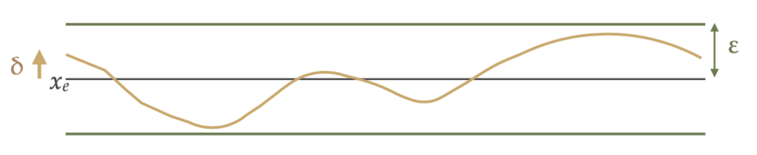
$x = x_e$ is an equilibrium point of the system if $f(x_e) = 0$.

Definition

The equilibrium point x_e is Lyapunov-stable if

$$\forall \epsilon > 0, \exists \delta > 0,\ \|x(t_0) - x_e\| \leq \delta \implies \|x(t) - x_e\| \leq \epsilon\ \forall t \geq t_0 \geq 0$$

- if the solution starts close to x_e, it will remain close forever
- ϵ can be made arbitrary small by choosing δ sufficiently small



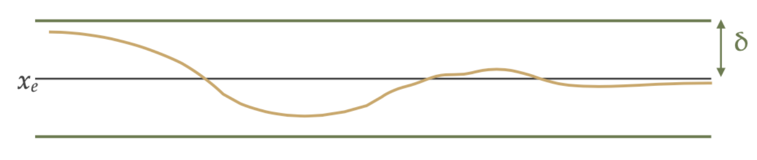
Stability of systems of differential equations

If, in addition, the signal converges to the equilibrium, then it is asymptotically stable:

Definition

The equilibrium point x_e is asymptotically stable if it is stable and there exists $\delta > 0$ such that

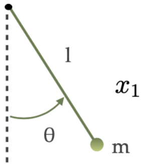
$$\|x(t_0) - x_e\| \leq \delta \implies lim_{t\to\infty}\|x(t) - x_e\| = 0$$



- The set of initial states from which the trajectories converge to the equilibrium is the region of attraction.
- The equilibrium is globally asymptotically stable (GAS) if the signal converges to the equilibrium for every initial state.

Example: pendulum

$$\begin{cases} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{g}{l}\sin x_1 - \frac{k}{m}x_2 \end{cases}$$

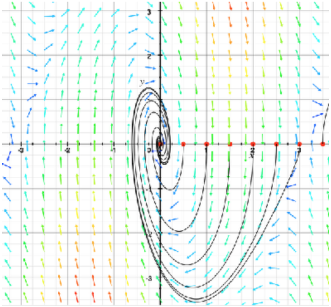


$x_1 = \theta,\ x_2 = \dot{\theta}$

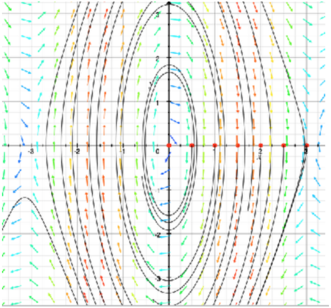
Two equilibrium points:

- $(\pi, 0)$ vertically upwards, $(0, 0)$ vertically downwards
- $(\pi, 0)$ is unstable, $(0, 0)$ is stable.

$k > 0$ (friction): $(0, 0)$ asymptotically stable



$k = 0$ (no friction): $(0, 0)$ not asym. stable



Lyapunov stability of differential systems

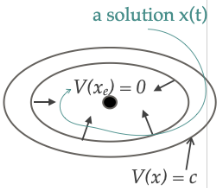
For linear systems: the origin is an asymptotically stable equilibrium point of $\dot{x} = Ax$ if and only if the eigenvalues of A have strictly negative real part

Theorem (General case: Lyapunov's stability theorem)

Let x_e be an equilibrium of $\dot{x} = f(x), x(0) = x_0$ with $x_e \in D \subset \mathbb{R}^n$.

If there exists a C^1 function $V : D \to \mathbb{R}$ such that

1. *V is positive definite: $V(x_e) = 0$ and $V(x) > 0, \forall x \neq x_e$*
2. $\frac{dV}{dt}(x - x_e) = \nabla V(x - x_e).f(x - x_e) = \mathcal{L}_f V(x - x_e) \leq 0, \forall x \in D$



then x_e is stable. If moreover $\mathcal{L}_f V(x - x_e) < 0,\ \forall x \in D - \{x_e\}$, then x_e is asymptotically stable.

Conversely, if x_e is a GAS equilibrium, there exists a Lyapunov function

We can think of V as a generalized energy function, and $-dV/dt$ as the associated dissipation function.

Extends to systems with control inputs: control-Lyapunov functions

Example: pendulum

$$\begin{cases} \dot{x}_1 & = x_2 \\ \dot{x}_2 & = -\frac{g}{l}\sin x_1 - \frac{k}{m}x_2 \end{cases}$$

$$V(x) = \frac{g}{l}(1 - \cos x_1) + \frac{x_2^2}{2}$$

Exercise: can we use V to prove $(0, 0)$ stable ? asymptotically stable?

Example: pendulum

$$\begin{cases} \dot{x}_1 & = x_2 \\ \dot{x}_2 & = -\frac{g}{l}\sin x_1 - \frac{k}{m}x_2 \end{cases} \qquad V(x) = \frac{g}{l}(1 - \cos x_1) + \frac{x_2^2}{2}$$

Exercise: can we use V to prove $(0,0)$ stable ? asymptotically stable?

- V is positive definite: $V(x) \geq 0$ and $V(x) = 0$ only on $x_e = (x_1 = 0, x_2 = 0)$ for $x_1 \in [0, 2\pi[$.
- for $x_e = (0,0)$:

$$\mathcal{L}_f V(x - x_e) = \frac{\partial V}{\partial x}(x).f(x) = (\frac{g}{l}\sin x_1, x_2).(x_2, -\frac{g}{l}\sin x_1 - \frac{k}{m}x_2) = -\frac{k}{m}x_2^2 \leq 0$$

  - $(0,0)$ is stable
  - asymptotically stable ?

Example: pendulum

$$\begin{cases} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{g}{l}\sin x_1 - \frac{k}{m}x_2 \end{cases} \qquad V(x) = \frac{g}{l}(1-\cos x_1) + \frac{x_2^2}{2}$$

Exercise: can we use V to prove $(0,0)$ stable ? asymptotically stable?

- V is positive definite: $V(x) \geq 0$ and $V(x) = 0$ only on $x_e = (x_1 = 0, x_2 = 0)$ for $x_1 \in [0, 2\pi[$.
- for $x_e = (0,0)$:

$$\mathcal{L}_f V(x - x_e) = \frac{\partial V}{\partial x}(x).f(x) = (\frac{g}{l}\sin x_1, x_2).(x_2, -\frac{g}{l}\sin x_1 - \frac{k}{m}x_2) = -\frac{k}{m}x_2^2 \leq 0$$

  - $(0,0)$ is stable
  - asymptotically stable ? cannot conclude, inequality not strict for $x_2 = 0,\ x_1 \neq 0$!

Example: pendulum

$$\begin{cases} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{g}{l}\sin x_1 - \frac{k}{m}x_2 \end{cases} \qquad V(x) = \frac{g}{l}(1-\cos x_1) + \frac{x_2^2}{2}$$

Exercise: can we use V to prove $(0,0)$ stable ? asymptotically stable?

- V is positive definite: $V(x) \geq 0$ and $V(x) = 0$ only on $x_e = (x_1 = 0, x_2 = 0)$ for $x_1 \in [0, 2\pi[$.
- for $x_e = (0,0)$:

$$\mathcal{L}_f V(x - x_e) = \frac{\partial V}{\partial x}(x).f(x) = (\frac{g}{l}\sin x_1, x_2).(x_2, -\frac{g}{l}\sin x_1 - \frac{k}{m}x_2) = -\frac{k}{m}x_2^2 \leq 0$$

  - $(0,0)$ is stable
  - asymptotically stable ? cannot conclude, inequality not strict for $x_2 = 0,\ x_1 \neq 0$!

LaSalle's Invariance principle:

Solution will converge to the largest invariant set contained in $\{x,\ \mathcal{L}_f V(x) = 0\}$, which is $(0,0)$, thus we can conclude to asymptotic stability

Algorithmically proving the stability

Difficulty = find these Lyapunov functions (verifying the properties is easy):

- define form of parametrized Lyapunov function candidate (e.g., quadratic)
- try to find values of parameters so that the required hypotheses hold

We search for candidate Lyapunov functions (multivariate) polynomial in x

- the conditions of the Lyapunov theorem become polynomial non- negativity conditions for $V(x)$ and $-dV/dt(x)$
- instead of trying to verify this (NP hard), relaxation to the sufficient condition that they admit decomposition as sums of squares (SOS), which can be done efficiently using semi-definite programming (SDP)
  - a multivariate polynomial $p(x)$ of degree $2d$ is a sum of squares (SOS) if there exist polynomials $f_i(x),\ i = 1 \dots M$ such that $p(x) = \sum_i f_i^2(x)$, or equivalently iff there exists a positive symmetric semi-definite matrix Q and a vector $Z(x)$ of monomials in x of degree $\leq d$, s.t. $p(x) = Z^T(x)QZ(x)$
  - the problem of finding Q is cast as a SDP (finding Q symmetric positive semidefinite that satisfies the above equality ie linear constraints on coeff)

Implemented in the Matlab toolbox SOSTOOLS

Ref: A Tutorial on Sum of Squares Techniques for Systems Analysis, by Papachristodoulou and Prajna

Casting the SOS problem to SDP: example

Suppose we want to know if $p(x_1, x_2) = 2x_1^4 + 2x_1^3x_2 - x_1^2x_2^2 + 5x_2^4$ is a SOS, we define $Z(x) = \begin{bmatrix} x_1^2 & x_2^2 & x_1x_2 \end{bmatrix}^T$ and look for Q symmetric positive semidefinite such that

$$\begin{array}{rcl} p(x_1, x_2) & = & 2x_1^4 + 2x_1^3x_2 - x_1^2x_2^2 + 5x_2^4 \\ & = & Z(x)^T QZ(x) \\ & = & q_{11}x_1^4 + q_{22}x_2^4 + (2q_{12} + q_{33})x_1^2x_2^2 + 2q_{13}x_1^3x_2 + 2q_{23}x_1x_2^3 \end{array}$$

from which we get $q_{11} = 2, q_{22} = 5, q_{13} = 1, q_{23} = 0$ and $2q_{12} + q_{33} = -1$.
For $q_{12} = -3$ and $q_{33} = 5$, Q is positive semidefinite and we have

$$Q = L^T L, \quad \text{where } L = \frac{1}{\sqrt{2}} \begin{bmatrix} 2 & -3 & 1 \\ 0 & 1 & 3 \end{bmatrix}$$

from which we can immediately deduce the SOS decomposition

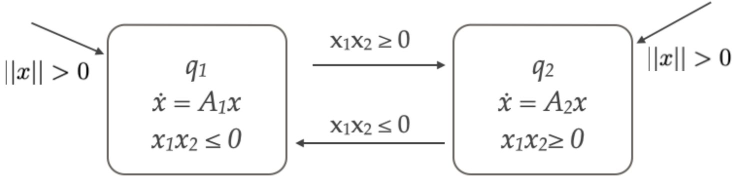
$$p(x) = \frac{1}{2}(2x_1^2 - 3x_2^2 + x_1x_2)^2 + \frac{1}{2}(x_2^2 + 3x_1x_2)^2$$

Hybrid stability: combining sub-systems

Stable subsystems do not guarantee a stable hybrid system!

- Combining stable modes, the hybrid system may be unstable!
- Combining unstable modes, the hybrid system may be stable!

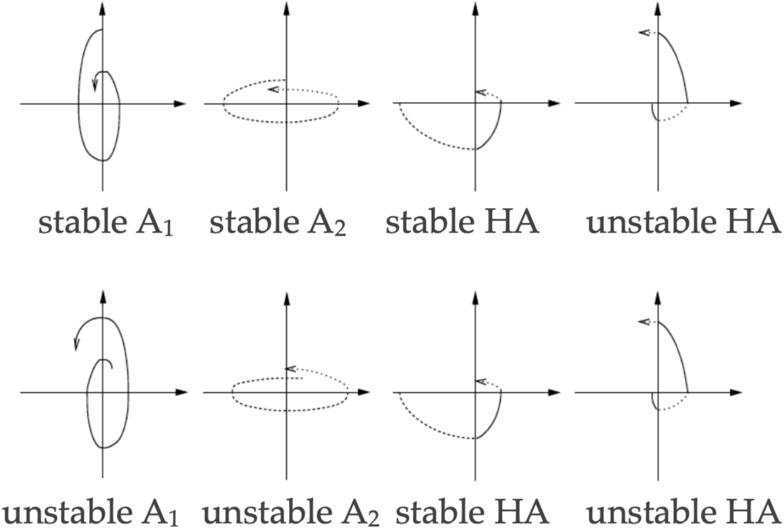
Example



$$A_1 = \begin{pmatrix} -1 & 10 \\ -100 & -1 \end{pmatrix} \qquad A_2 = \begin{pmatrix} -1 & 100 \\ -10 & -1 \end{pmatrix}$$

- both subsystems are asymptotically stable at $x_e = 0$: eigenvalues $-1 + i\sqrt{1000}$
- the HA is unstable
- if switching conditions flipped, $x_e = 0$ is stable

Combining subsystems



stable A_1 stable A_2 stable HA unstable HA

unstable A_1 unstable A_2 stable HA unstable HA

Stability of hybrid/switched systems

Given a switched system $\dot{x} = f_\sigma(x)$:

Problem 1. Universal asymptotic stability, or stability under arbitrary switching.

- not always true even if sub-systems are stable
- true if common Lyapunov function (sufficient condition)

Problem 2. Existential asymptotic stability. Is there a given switching strategy or class of switching strategies for which the system is GAS

- true is sub-systems are stable (take one mode and never switch)

If the switch signal is generated by an hybrid automaton: hybrid asymptotic stability

Common Lyapunov function (universal asymptotic stability)

Try to find a shared Lyapunov function that decreases along all sub-modes:

A positive definite C^1 function $V : \mathbb{R}^n \to \mathbb{R}$ is called a common Lyapunov function for $\dot{x} = f_\sigma(x)$ with $\sigma \in \{1, \dots, N\}$ if $\mathcal{L}_{f_i} V(x - x_e) < 0$ when $x \neq x_e$ and for all $i = 1, \dots, n$.

Theorem

If all the submodes f_i share a positive definite radially unbounded ($V(x) \to \infty$ when $\|x\| \to \infty$) common Lyapunov function, then the switched system is GUAS.

It is a necessary and sufficient condition:

Theorem (Converse theorem)

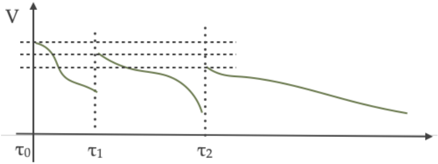
If the switched system is GUAS, then all f_i share a positive definite radially unbounded common Lyapunov function.

Relaxations to the common Lyapunov approach

In the case of switched (linear) systems with state- dependent switching, we can relax the conditions and prove stability more efficiently:

- Relaxation 1: a common Lyapunov function $V(x) = x^T Px$ decreasing for each mode f_i only in the corresponding active region
- Relaxation 2: multiple Lyapunov functions $V_i(x) = x^T P_i x$, « connected » in a suitable way: continuous piecewise quadratic (or polynomial) Lyapunov functions
- Relaxation 3: each Lyapunov function $V_i(x)$ actually needs to be positive definite only in its active region

Extension for hybrid systems with jumps and resets, with generalized Lyapunov functions decreasing on all subsystems, that can take jumps as long as the sequence of values after each jump is decreasing.



(Continuous) systems with inputs and disturbances

Comparison functions

- class $\mathcal{K}$ function: zero at zero, continuous, increasing
- class $\mathcal{K}_\infty$ is a class $\mathcal{K}$ function $\alpha(.)$ and $\alpha(r) \to \infty$ when $r \to \infty$
- class $\mathcal{KL}$ function: $\beta : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ such that
  - $\beta(., t)$ is a class $\mathcal{K}$ function (zero at zero, continuous and increasing)
  - $\beta(s, .)$ is continuous, decreasing and vanishing at infinity

Definition

The system with u bounded external input and f a Lipschitz continuous function w.r.t. x

$$\dot{x} = f(x, u),\ x(0) = x_0,\ x(t) \in \mathbb{R}^n \tag{1}$$

is called input-to-state stable (ISS) if there exist functions $\beta \in \mathcal{KL}$ and $\mu \in \mathcal{K}_\infty$ such that for all x_0 and u

$$|x(t, x_0, u)| \leq \beta(\|x_0\|, t) + \mu(\|u\|),\ \forall t \geq 0$$

Implies globally asymptotically stable in the absence of external inputs + its trajectories are bounded by a function of the size of the input

ISS and ISS-Lyapunov functions

Definition

A continuous function $V : \mathbb{R}^n \to \mathbb{R}_+$ is called an ISS-Lyapunov function in a dissipative form for the system (1), if $\exists \Psi_1, \Psi_2 \in \mathcal{K}_\infty$, $\alpha \in \mathcal{K}_\infty$, and $\xi \in \mathcal{K}$ such that

$$\Psi_1(|x|) \leq V(x) \leq \Psi_2(|x|), \;\; \forall x \in \mathbb{R}^n,$$

and $\forall x \in \mathbb{R}^n, \forall u \in \mathbb{R}^m$, the following holds:

$$\mathcal{L}_f V(x) \leq -\alpha(V(x)) + \xi(\|u\|).$$

Theorem (Sontag and Wang (1995))

Let f be Lipschitz continuous on bounded balls of $\mathbb{R}^n \times \mathbb{R}^m$ *and* $f(0,0) = (0,0)$. *Then (1) is ISS if and only if there is an infinitely differentiable ISS-Lyapunov function.*

Closed-loop robustness = incremental stability

Incremental stability = stability of trajectories with respect to one another instead of with respect to an equilibrirum or a reference trajectory:

Definition

The system (1) with fixed input u is incremental globally asymptotatically stable (δ-GAS) if $\exists \beta$ of class $\mathcal{KL}$ such that $\forall x_0, \tilde{x_0} \in \mathbb{R}^n, \forall t \geq 0$ the following holds:

$$|x(t, x_0, u) - x(t, \tilde{x_0}, u)| \leq \beta(|\tilde{x_0} - x_0|, t)$$

The system (1) is incremental input-to-state stable (δ-ISS) if $\exists \beta$ of class $\mathcal{KL}$ and $\mu \in \mathcal{K}_\infty$ such that $\forall x_0, \tilde{x_0} \in \mathbb{R}^n, \forall u, \tilde{u} \in \mathbb{R}^m, \forall t \geq 0$ the following holds:

$$|x(t, x_0, u) - x(t, \tilde{x_0}, u)| \leq \beta(|\tilde{x_0} - x_0|, t) + \mu(\|u - \tilde{u}\|)$$

Proof by:

- $\delta-$GAS and $\delta-$ISS Lyapunov functions
- Contraction theory and reachability analysis

Ref: A Lyapunov Approach to Incremental Stability Properties, D. Angeli, TAC 2002

A Lyapunov approach in incremental stability, Zamani and Majumdar, CDC 2011

Proving safety using a (barrier) certificate function

We focused on stability analysis, similar techniques for safety verification:

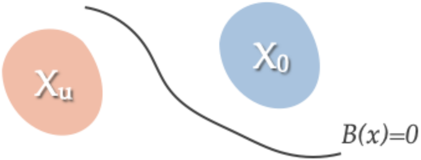
- safety verification problem formulated as an invariant property which should be satisfied by all reachable states
- we want to prove safety without computing flows explicitly
- barrier certificate = a function of the state, that separates safe states from unsafe ones: must be satisfied by initial states, and preserved by discrete and continuous transitions (conditions on function and its Lie derivative)
- similar to the Lyapunov stability approach

Barrier certificate (here for a continuous system)

Let $\dot{x} = f(x)$ defined on X, with initial states X_0 and an unsafe set X_u. Suppose there exists a function $B : X \to \mathbb{R}$ differentiable with respect to its argument and satisfying

1. $B(x) > 0, \ \forall x \in X_u$,
2. $B(x) \leq 0, \ \forall x \in X_0$,
3. $\mathcal{L}_f B(x) \leq 0, \ \forall x \in X$ such that $B(x) = 0$,

then the safety of the system is guaranteed. That is, there exists no trajectory contained in X that starts from an initial state in X_0 and reaches a state in X_u.



Can be extended to hybrid systems similarly to the Lyapunov functions.

Ref: Safety Verification of Hybrid Systems Using Barrier Certificates, Prajna and Jadbabaie, 2004.

Online verification and monitoring

Online/Offline verification

- Offline verification : verify a system prior to its use
- Online verification : embed a (finite-time horizon) verifier on the system
  - to dynamically monitor its behavior
  - go to a failsafe state/controller in case of problem
  - particularly useful in case offline verification is next to impossible

Often more refined properties than (un)safety regions:

- Expressed in temporal logics
- Possibly interpreted with a robust semantics, allowing optimization/falsification

Robust Satisfaction of Temporal Logic over Real-Valued Signals, Donzé and Maler, 2010

Robust Online Monitoring of Signal Temporal Logic, Deshmukh et al., 2015

Use in safe AI: safe training

"Safe training" of controllers:

- Enforcing Lyapunov like properties (barrier certificates) in learning
  A Review On Safe Reinforcement Learning Using Lyapunov and Barrier Functions, Kushwaha et al., 2025
- Using robust semantics of STL formulas for devising reward functions in RL based controller synthesis
  Reinforcement Learning With Temporal Logic Rewards, Li et al., 2017

Next

For next week: paper reading

- *Robust Satisfaction of Temporal Logic over Real-Valued Signals, A. Donzé and O. Maler, 2010*
- Who is presenting ?

Next time (on January 20): Inner and outer approximations of general quantified reachability problems and applications