Lecture 6
Abstractions for the Analysis and Synthesis of Control Protocols for Hybrid Systems

Ufuk Topcu

Nok Wongpiromsarn Richard M. Murray

EECI, 20 March 2013

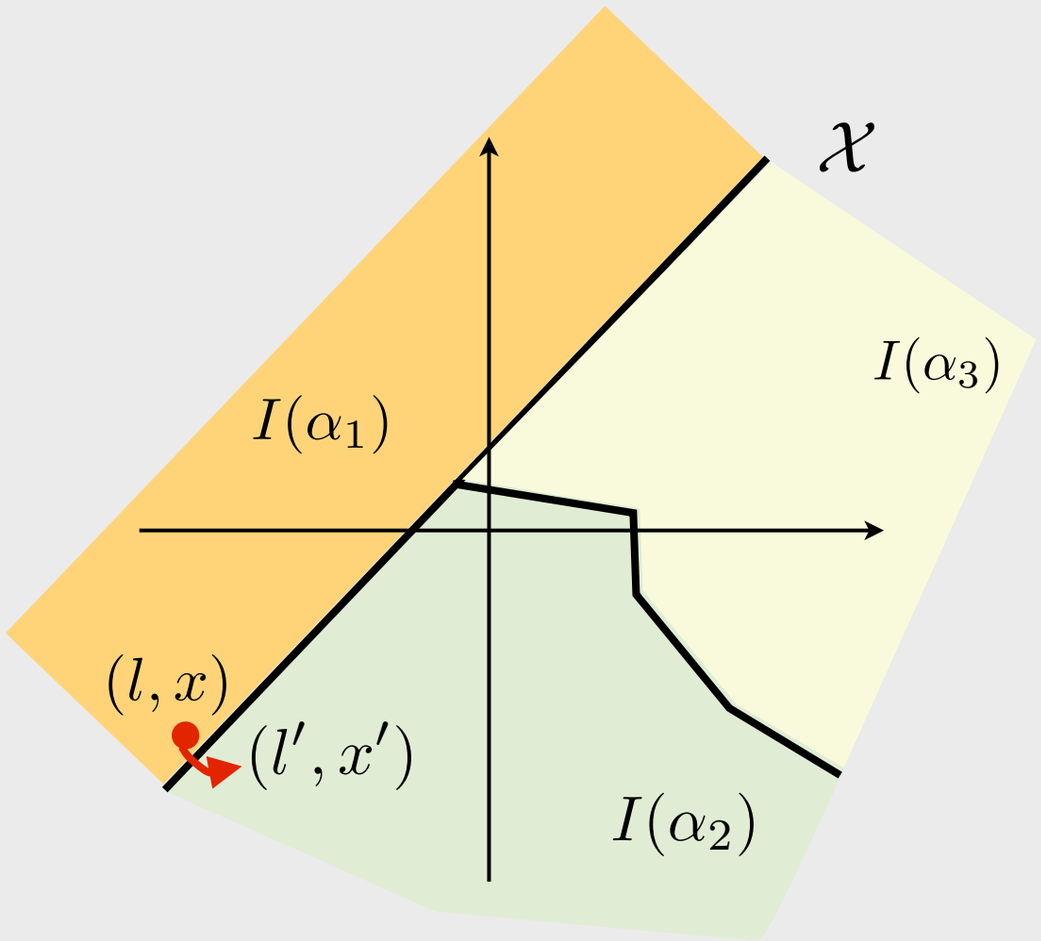
Outline:

- Finite-state approximations of hybrid systems
- Use of model checking for the verification of hybrid systems
- Construction of finite-state abstractions for synthesis
- “Approximate” abstractions

A (simple) hybrid system model

Hybrid system: $H = (\mathcal{X}, L, X_0, I, F, T)$ with

- $\mathcal{X}$, continuous state space;
- L, finite set of locations (modes);
- Overall state space $X = \mathcal{X} \times L$;
- $X_0 \subseteq X$, set of initial states;
- $I : L \to 2^{\mathcal{X}}$, *invariant* that maps $l \in L$ to the set of possible continuous states while in location l;
- $F : X \to 2^{\mathbb{R}^n}$, set of vector fields, i.e., $\dot{x} \in F(l, x)$;
- $T \subseteq X \times X$, relation capturing discrete transitions between locations.



$L = \{\alpha_1, \alpha_2, \alpha_3\}$

Specifications

Given: $H = (\mathcal{X}, L, X_0, I, F, T)$

Solution at time *t* with the initial condition $x_0 \in \mathcal{X}_0$: $\phi(t; x_0)$

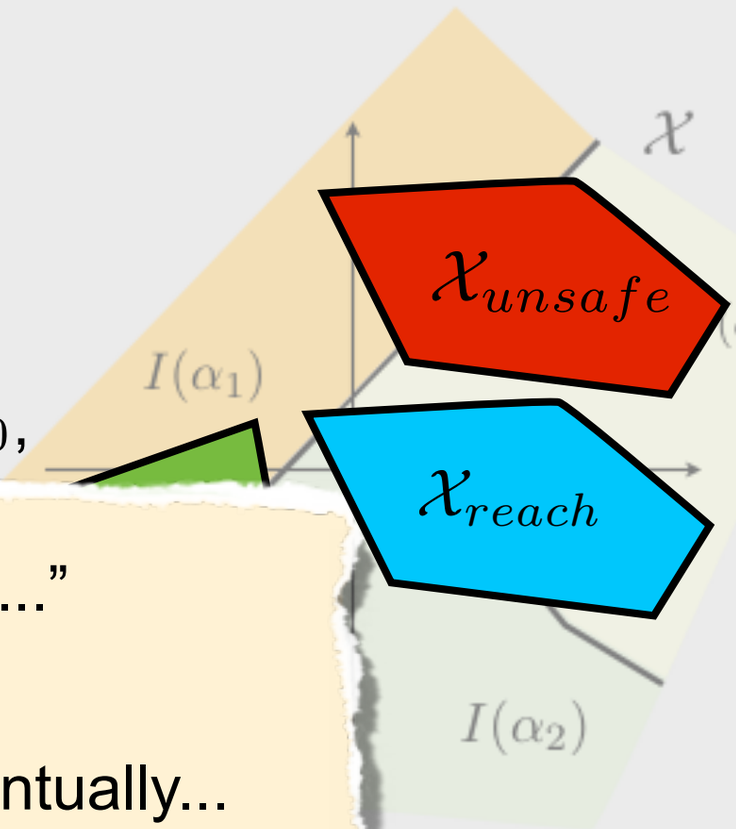
- With the simple model *H*, specifying the initial state also specifies the initial mode.

Sample temporal properties:

- *Stability*: Given equilibrium $x_e \in \mathcal{X}$, for all $x_0 \in \mathcal{X}_0 \subseteq \mathcal{X}$,

$$\phi(t; x_0) \in \mathcal{X}, \ \forall t \ \text{ and } \ \phi(t; x_0) \to x_e, \ t \to \infty$$

- *Safety*: Given $\mathcal{X}_{unsafe} \subseteq \mathcal{X}$, safety property holds if there exists no t_{unsafe} and trajectory with initial condition $x_0 \in \mathcal{X}_0$,



stability: ∀ initial conditions, "eventually always..."

safety: ∀ initial conditions, always...

reachability: ∃ an initial condition such that eventually...

eventuality: ∀ initial conditions, eventually...

- *Eventuality*: reachable from every initial condition

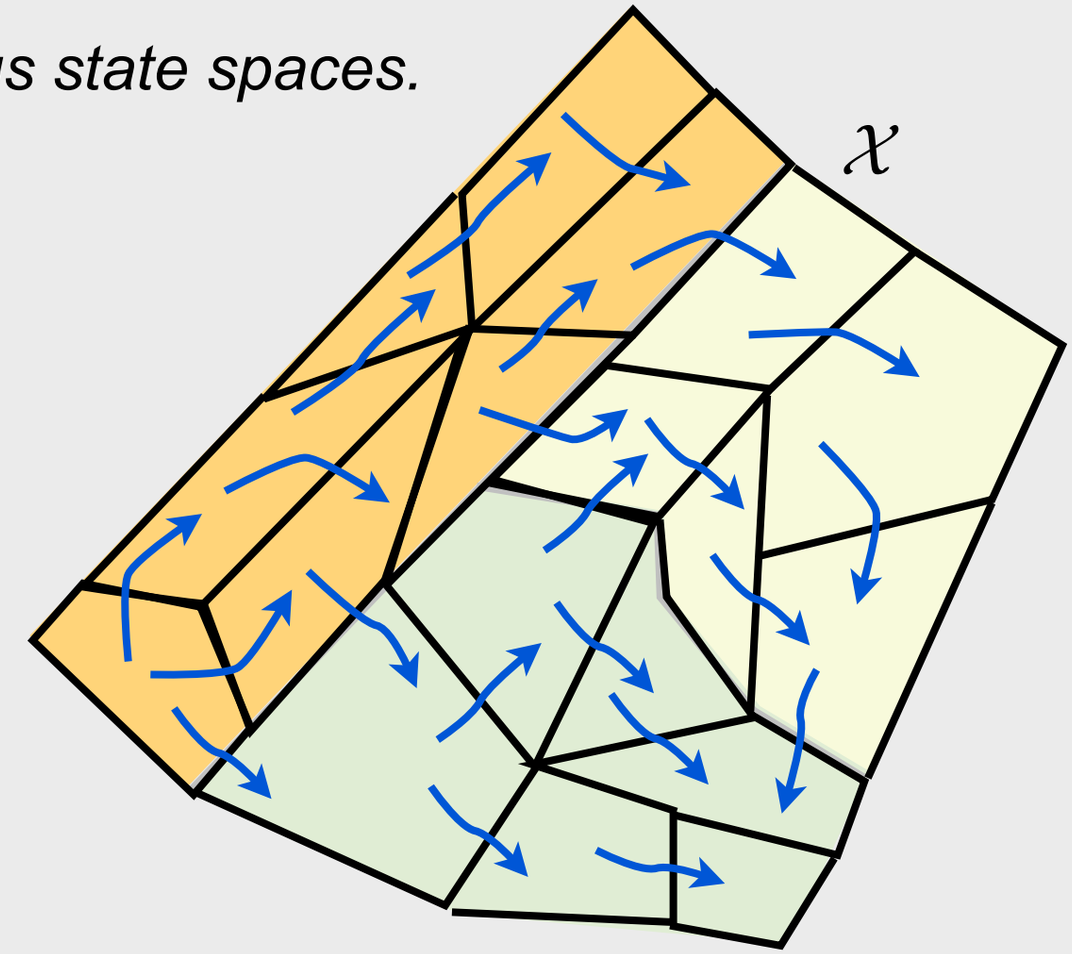
- Combinations of the above, e.g., starting in X_A, reach both X_B and X_C, but X_B will not be reached before X_C is reached while staying safe.

Analysis of hybrid systems

Why not directly use model checking?

- Model checking applied to finite transitions systems
- Exhaustively search for counterexamples....
  - if found, property does not hold.
  - if there is no counterexample in all possible executions, the property is verified.

Exhaustive search is not possible over continuous state spaces.



Approaches for hybrid system verification:

1. Construct finite-state approximations and apply model checking
   - Preserve the meaning of the properties, i.e., proposition preserving partitions
   - Use “over”- or “under”-approximations

2. Deductive verification
   - Construct Lyapunov-type certificates
   - Account for the discrete jumps in the construction of the certificate

3. Explicitly construct the set of reachable states
   - Limited classes of temporal properties (e.g., reachability and safety)
   - Not covered in this course

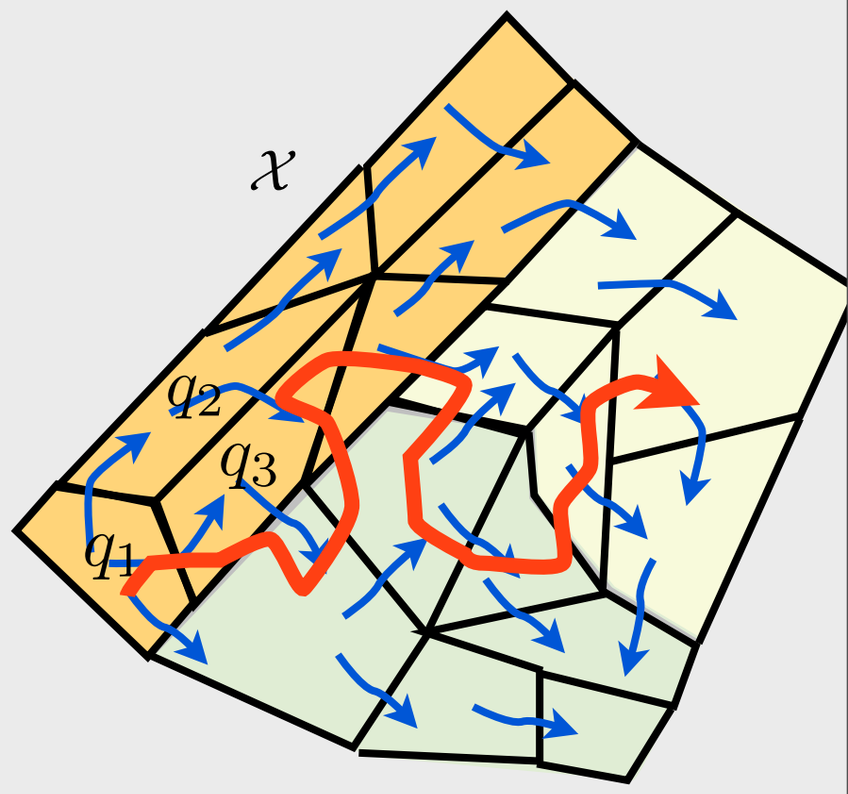
Finite-state, under- and over-approximations

Hybrid system: $H = (\mathcal{X}, L, X_0, I, F, \rightarrow_H)$

Finite-transition system: $TS = (Q, \rightarrow, Q_0)$

Define the map $T : Q \rightarrow 2^{\mathcal{X}}$

For discrete state q, $T^{-1}(q)$ is the corresponding cell in $\mathcal{X}$.



Under-approximation: *TS* is an under-approximation of *H* if the following two statements hold.

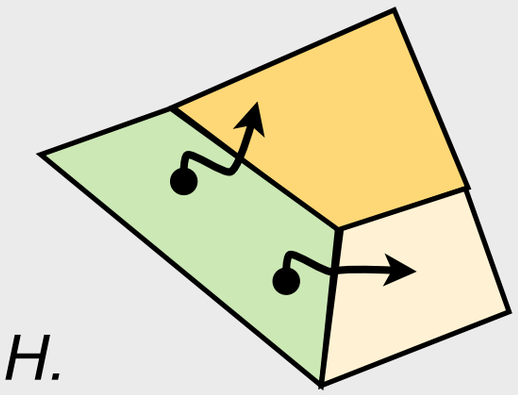
- Given $q, q' \in Q$ with $q \neq q'$, if $q \rightarrow q'$, then for all $x_0 \in T^{-1}(q)$, there exists finite $\tau > 0$ such that

$$\phi(\tau; x_0) \in T^{-1}(q'), \quad \phi(t; x_0) \in T^{-1}(q) \cup T^{-1}(q'), \quad \forall t \in [0, \tau]$$

- If $q \rightarrow q$, then $T^{-1}(q)$ is positively-invariant.

In other words:

- Every discrete trajectory in an under-approximation *TS* can be implemented by *H*.
- *H* "simulates" *TS*.



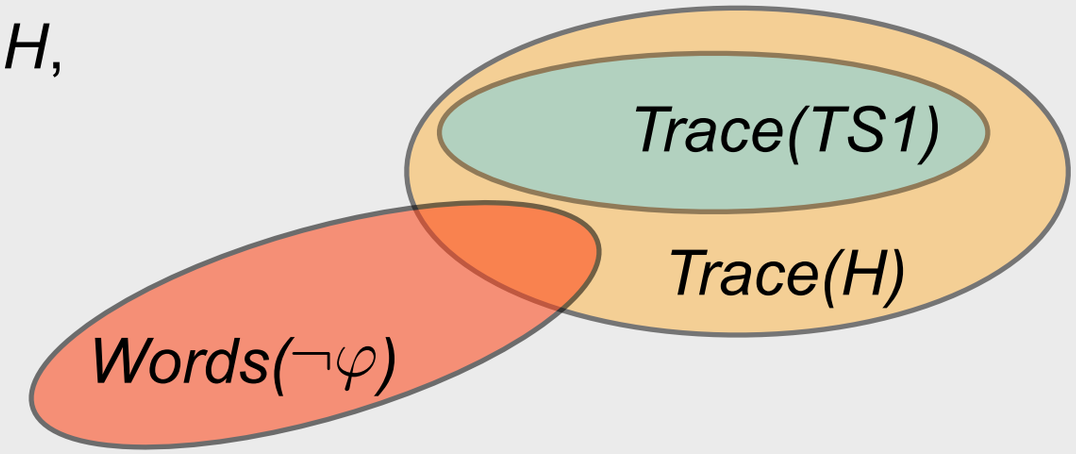
Over-approximation: *TS* is an over-approximation of *H,* if for each discrete transition in TS, there is a "possibility" to be implemented by *H.*

- Possibility induced by the coarseness of the partition.

Use of under-approximations

Let the following be given.

- A hybrid system *H*,
- a finite-state, under-approximation *TS1* for *H*,



Verification

- Let an LTL specification φ be given.
- Question: $H \models \varphi$?
- Model check "$TS1 \models \varphi$?"

$\mathrm{Words}(\neg\varphi) \cap \mathrm{Trace}(TS1)$ is nonempty

$\Downarrow$

$\mathrm{Words}(\neg\varphi) \cap \mathrm{Trace}(H)$ is nonempty

H cannot satisfy the specification.

$TS1 \not\models \varphi$

$\Downarrow$

$H \not\models \varphi$

$\mathrm{Words}(\neg\varphi) \cap \mathrm{Trace}(TS1)$ is empty

Inconclusive

Logic synthesis:

- If $\mathrm{Words}(\varphi) \cap \mathrm{Trace}(TS1)$ is nonempty, there exists a trajectory of *TS1* which satisfies φ and can be implemented by *H*.
- Otherwise, inconclusive.

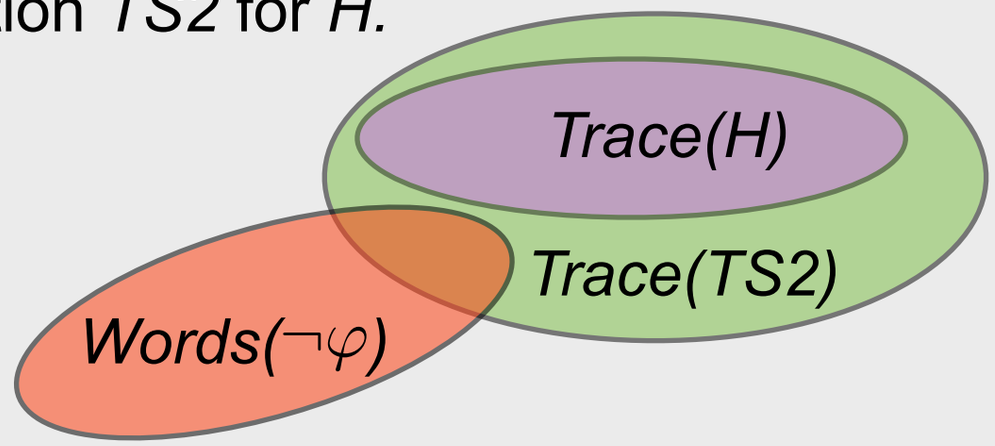
Use of over-approximations

Hybrid system *H* and a finite-state, over-approximation *TS2* for *H.*

Verification

$\text{Words}(\varphi) \cap \text{Trace}(TS2)$ is nonempty

Inconclusive



$\text{Words}(\neg\varphi) \cap \text{Trace}(TS2)$ is empty

$\Downarrow$

$\text{Words}(\neg\varphi) \cap \text{Trace}(H)$ is empty

H satisfies the specification.

$TS2 \models \varphi$

$\Downarrow$

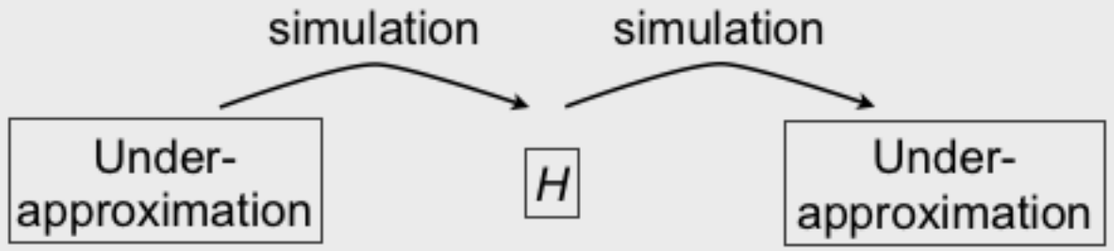
$H \models \varphi$

Logic synthesis:

- If $\text{Words}(\varphi) \cap \text{Trace}(TS2)$ is empty, no valid trajectories for *TS2* or *H*.
- Otherwise, inconclusive.

Remarks:

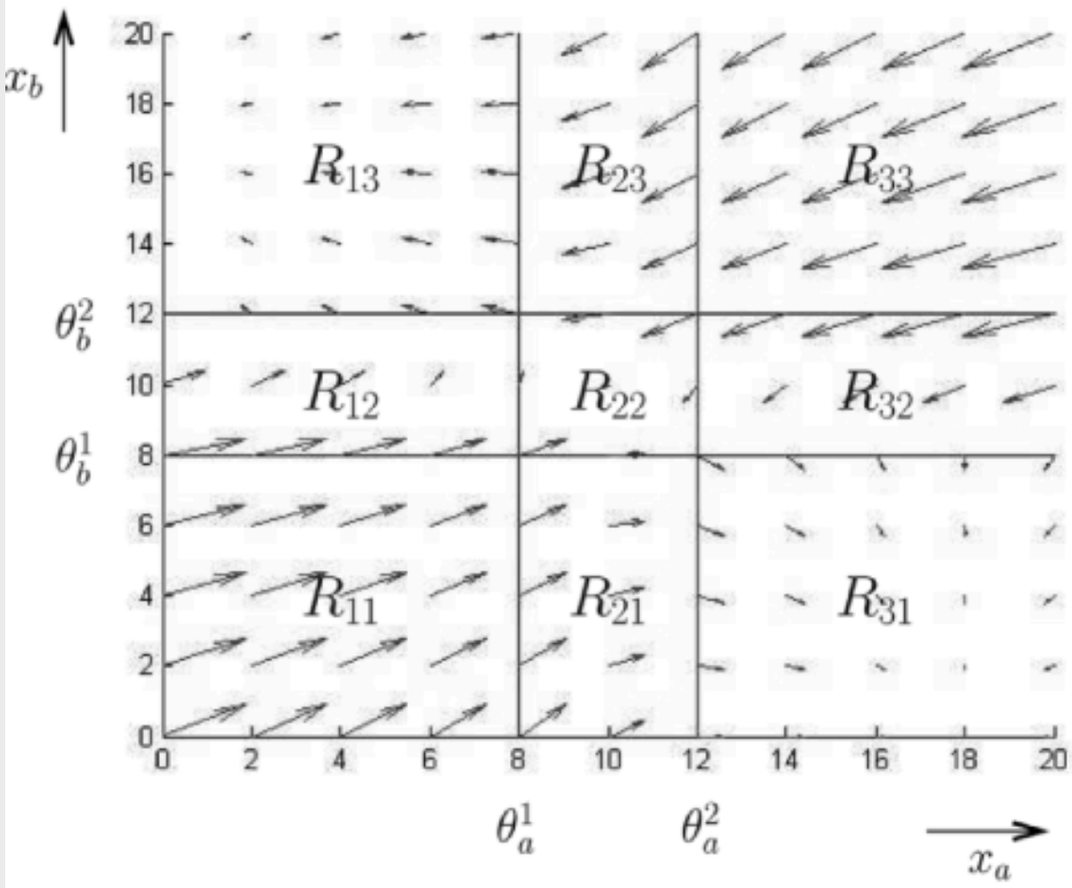
- Under- and over-approximations give partial results.
- Potential remedies:
  - Finer approximations
  - Bisimulations



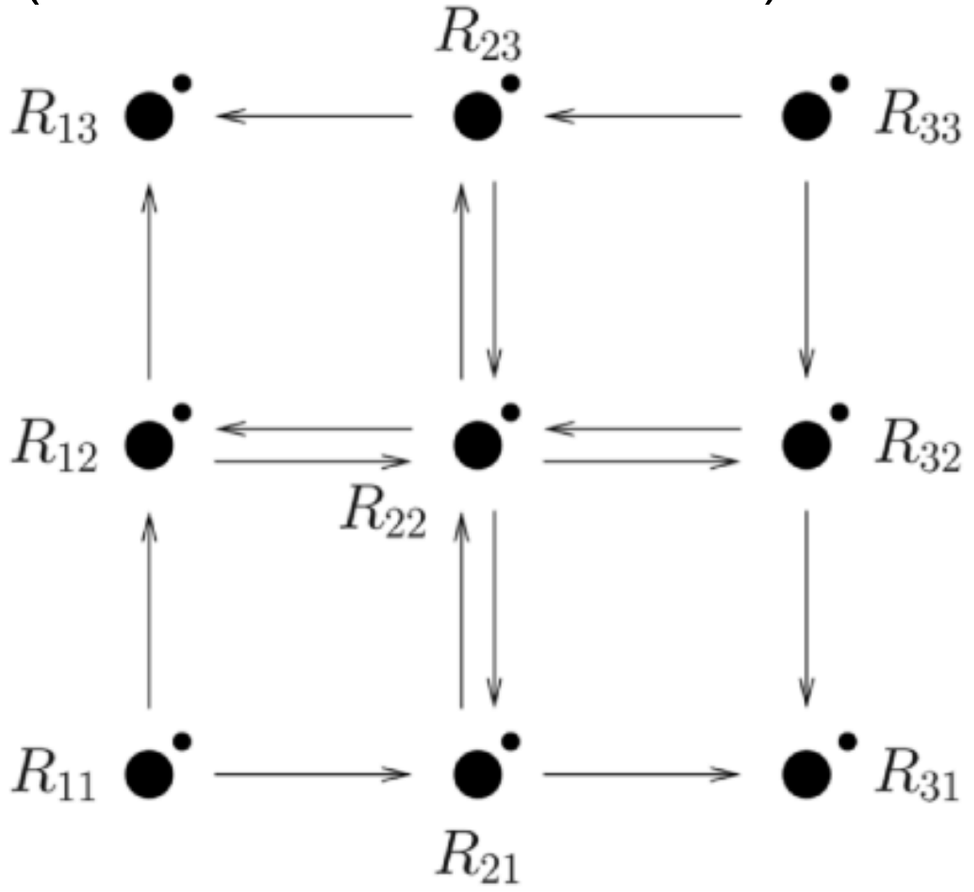
Example: verification

System models:

Continuous vector field:



Discrete over-approximation:
(small dots: self transitions)



Specifications:

$$\left(x_a < \theta_a^1 \wedge x_b > \theta_b^2 \rightarrow \Box\left(x_a < \theta_a^1 \wedge x_b > \theta_b^2\right)\right)$$
$$\wedge \left(x_b < \theta_b^1 \wedge x_a > \theta_a^2 \rightarrow \Box\left(x_b < \theta_b^1 \wedge x_a > \theta_a^2\right)\right)$$

Holds for the over-approximation; hence, also for the system itself.

$$\Diamond\left(x_a < \theta_a^2 \vee x_b < \theta_b^2\right)$$

Does not hold for the over-approximation.

Example from “Temporal logic analysis of gene networks under parametric uncertainty,” Batt, Belta, Weiss, Joint special issue of IEEE TAC & Trans on Circuits, 2008.

How to construct a finite-state abstraction?

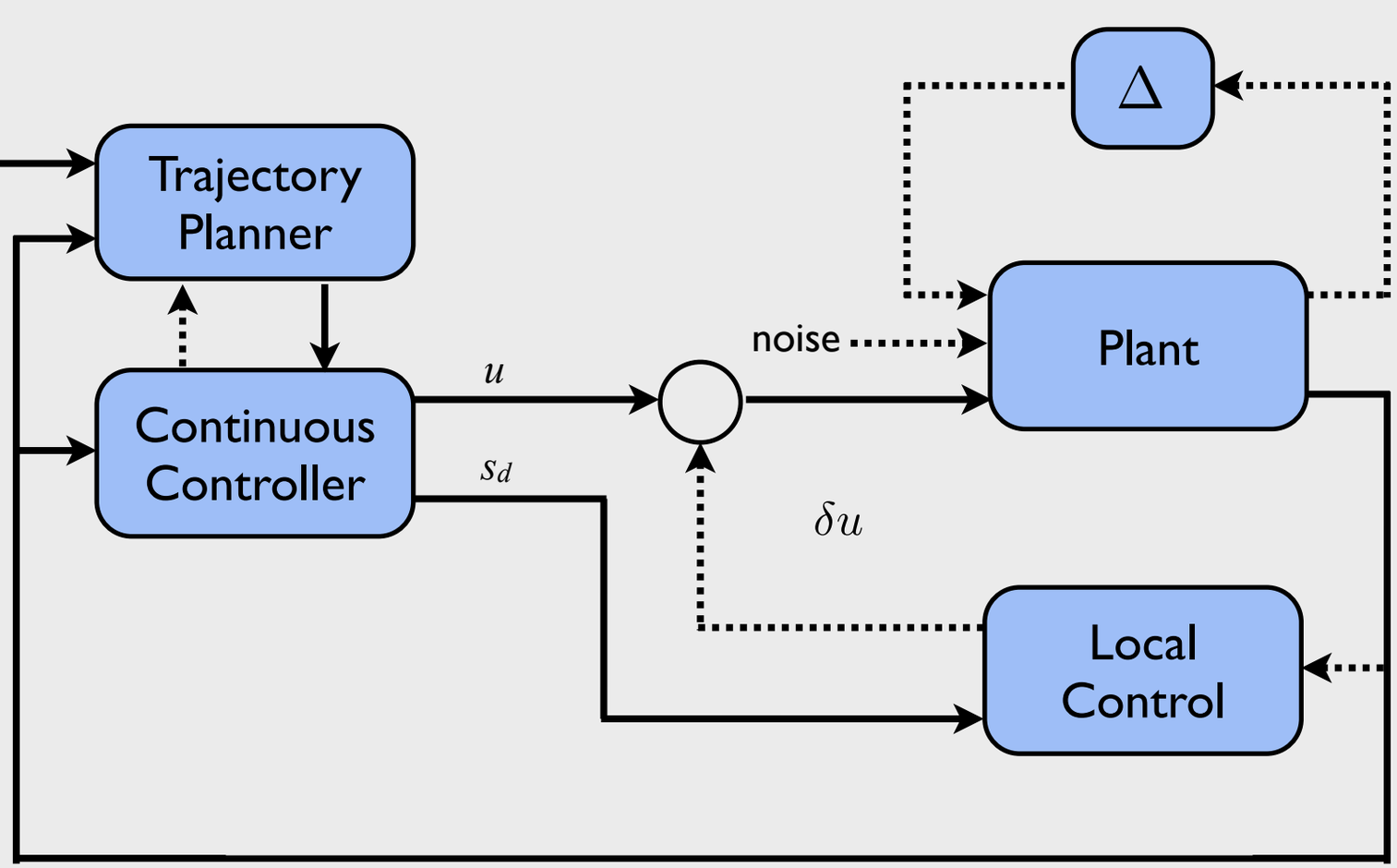
Focus on synthesis: Construct a finite-state under-approximation (of the underlying continuous/hybrid dynamics) such that

- the finite-state model is used in discrete planning, and
- all provably correct discrete plans can be implemented at the continuous level.

Discrete planner ensures that the spec is satisfied

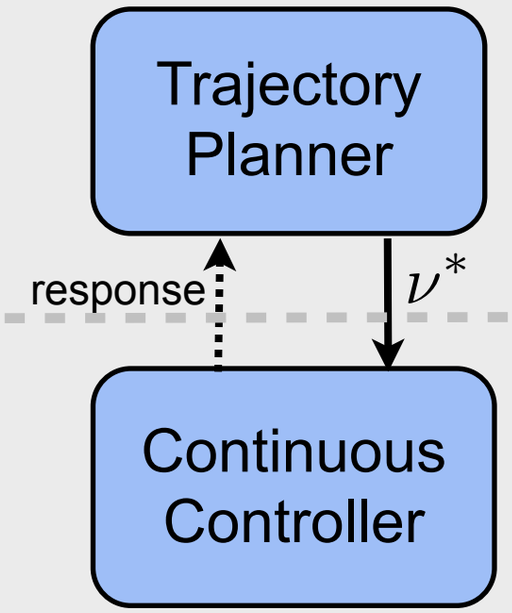
+

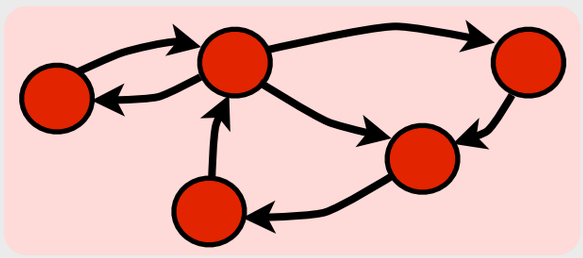
Continuous controller *implements* the discrete plan (handles low-level dynamics & constraints)

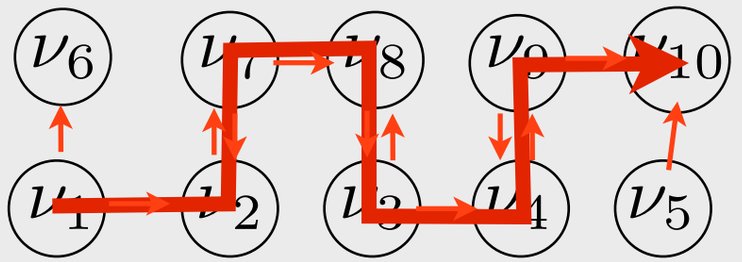


Incorporating continuous dynamics -- overview

Main idea:

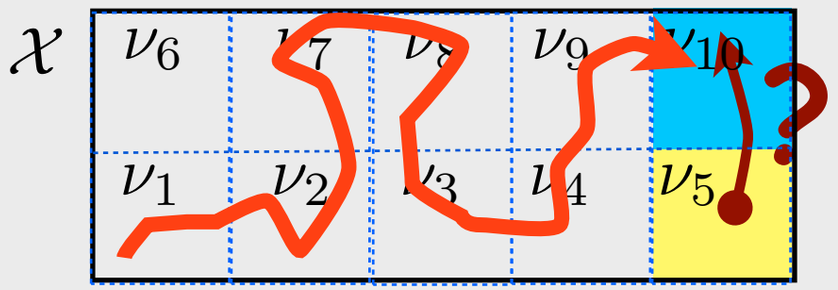






$$\dot{\xi} = f(\xi, w, u)$$

$$\xi \in \mathcal{X},\ u \in \mathcal{U}, w \in \mathcal{W}$$



Theorem: For any discrete run satisfying the specification, there exists an admissible control signal leading to a continuous trajectory satisfying the specification.

Proof: Constructive ⟶ Finite-state model + Continuous control signals.

Abstraction refinement for reducing potential conservatism.

Finite state abstraction

Given:

•A system with controlled variables $s \in S$ in domain $dom(S)$ and environment variables $e \in E$ in domain $dom(E)$.

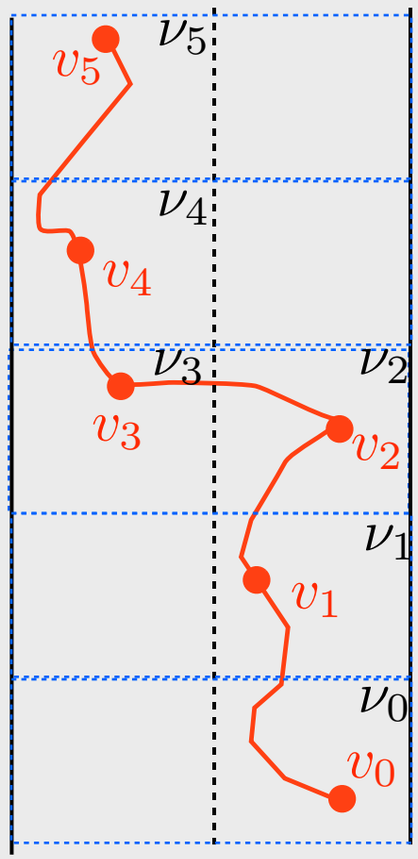
•Define $v = (s, e)$, $V = S \cup E$ and $dom(V) = dom(S) \times dom(E)$.

•Controlled variables evolve with (for *t = 0,1,2,...*):

$$s[t+1] = As[t] + B_u u[t] + B_d d[t] \longleftarrow \text{state evolution}$$
$$u[t] \in U \longleftarrow \text{admissible control inputs}$$
$$d[t] \in D \longleftarrow \text{exogenous disturbances}$$
$$\left.\begin{array}{r} s[0] \in dom(S) \\ s[t+1] \in dom(S) \end{array}\right\} \longleftarrow \text{set that states take values in}$$

•System specification φ

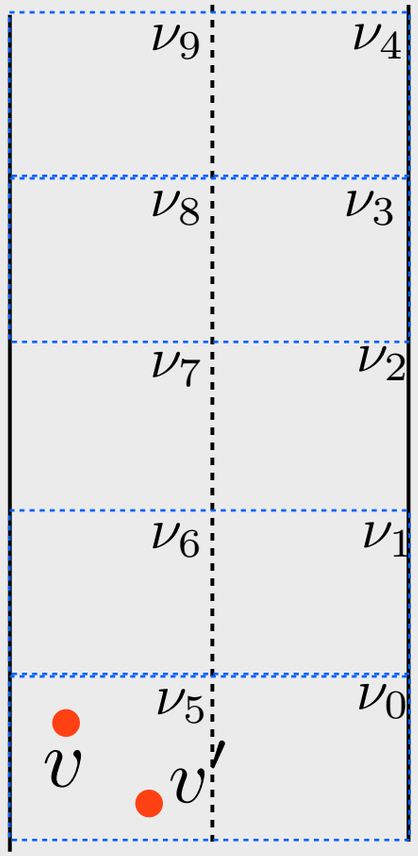
Find: A finite transition system with discrete states ν such that for any sequence $\nu_0\nu_1\ldots$ satisfying φ, (very roughly speaking) there exists a sequence of admissible control signals leading to an infinite sequence $v_0v_1v_2\ldots$ that satisfies φ.



Proposition preserving partition

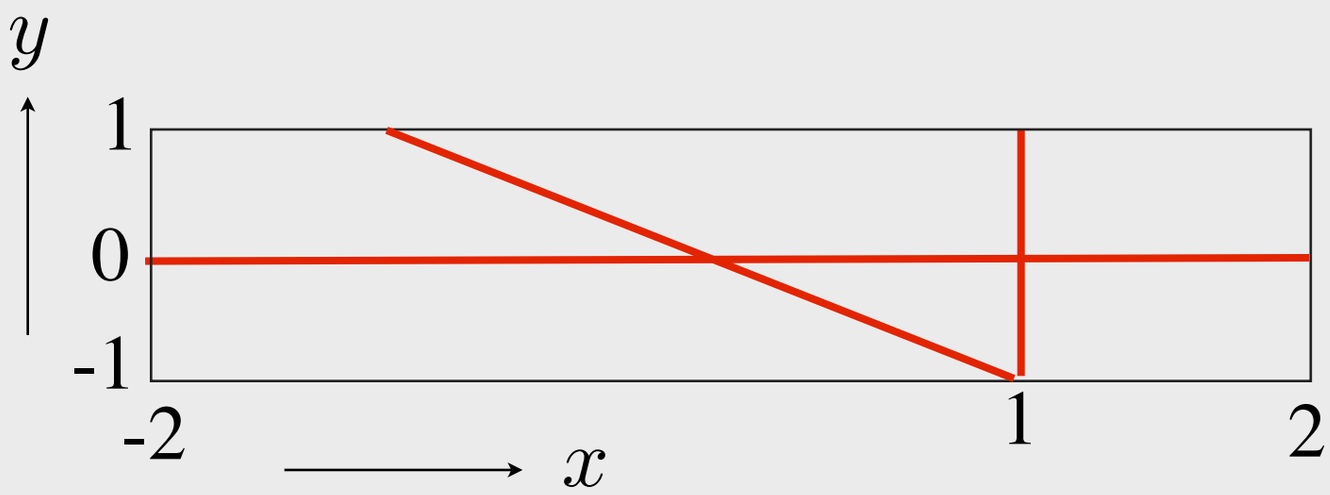
Given $dom(V)$ and atomic propositions in Π.

A partition of $dom(V)$ is said to be proposition preserving if, for any atomic proposition $\pi \in \Pi$ and any states v and v' that belong to the same cell of the partition, v satisfies π if and only if v' satisfies π.



A discrete state ν is said to satisfy π if and only if there exists a continuous state v, in the cell labeled, that satisfies π.

Example: $\Pi = \{x \leq 1, y \geq 0, x + y \geq 0, \ldots\}$



$$\nu_5 \Vdash_d \pi \Leftrightarrow \exists v \in \nu_5 \text{ s.t. } v \Vdash \pi$$

+

proposition preserving:

$$v \Vdash \pi \Leftrightarrow v' \Vdash \pi$$

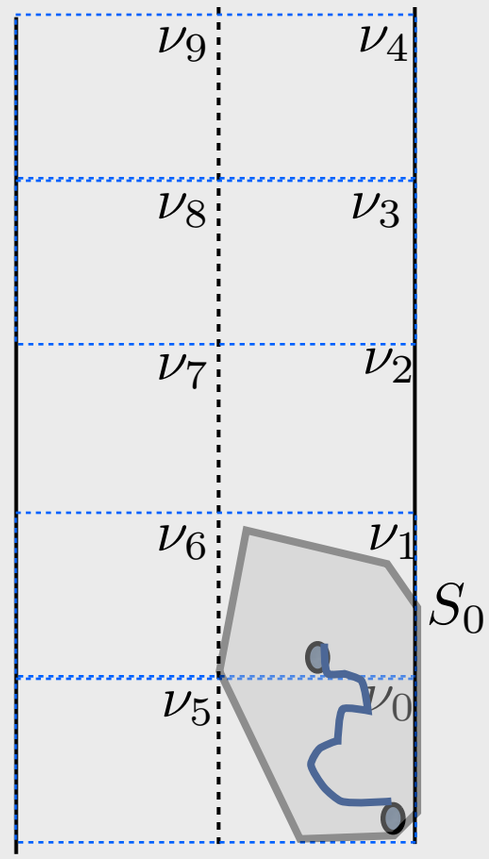
$$\Downarrow$$

$$\nu_5 \Vdash_d \pi \Leftrightarrow \forall v \in \nu_5 \text{ s.t. } v \Vdash \pi$$

Finite-time reachability

A discrete state ν_j is finite-time reachable from a discrete state ν_i, only if starting from <u>any</u> $s[0] \in T_s^{-1}(\nu_i)$, there exists
- a finite horizon length $N \in \{0, 1, \ldots\}$
- for <u>any</u> allowable disturbance, there exists $u[0], u[1], \ldots, u[N-1] \in U$ such that

$$s[N] \in T_s^{-1}(\nu_j)$$
$$s[t] \in T_s^{-1}(\nu_i) \cup T_s^{-1}(\nu_j),\ \forall t \in \{0, \ldots, N\}$$



Verifying the reachability relation:

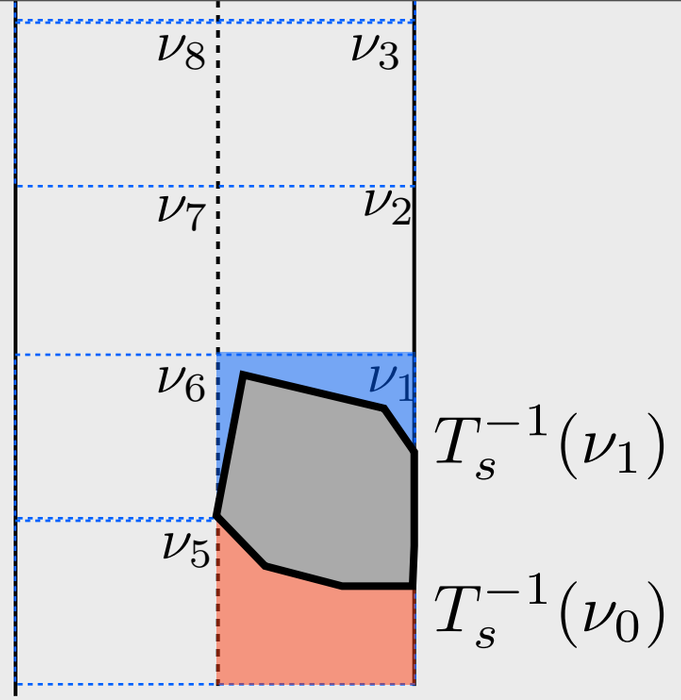
- Compute the set S_0 of $s[0]$ from which $T_s(\nu_j)$ can be reached under the system dynamics in a pre-specified time N.
- Check whether $T_s^{-1}(\nu_i) \subseteq S_0$.

system dynamics

$$\begin{cases} s[t+1] = As[t] + B_u u[t] + B_d d[t] \\ u[t] \in U \\ d[t] \in D \\ s[0] \in dom(S) \\ s[t+1] \in dom(S) \end{cases}$$

Computing S_0

Given N and polyhedral sets
$T_s^{-1}(\nu_i) = \{s \in \mathbb{R}^n \ : L_1 s \leq M_1\}$
$U = \{u \in \mathbb{R}^m \ : L_2 u \leq M_2\}$
$T_s^{-1}(\nu_j) = \{s \in \mathbb{R}^n \ : L_3 s \leq M_3\}$.



S_0 is computed as the set of s_0 such that there exist $u[0], \ldots, u[N-1]$ satisfying $L_2 u[t] \leq M_2$, for $t \in \{0, \ldots, N-1\}$, leading to

$L_1 s[t] \leq M_1$ for $t = 0, \ldots, N-1$

$L_3 s[N] \leq M_3$, → affine in s0 and u

where

$$s[t] = A^t s_0 + \sum_{k=0}^{t-1} \left(A^k B_u u[t-1-k] + A^k B_d d[t-1-k] \right),$$

for all $d[0], \ldots, d[N-1] \in D$ (D polyhedral).

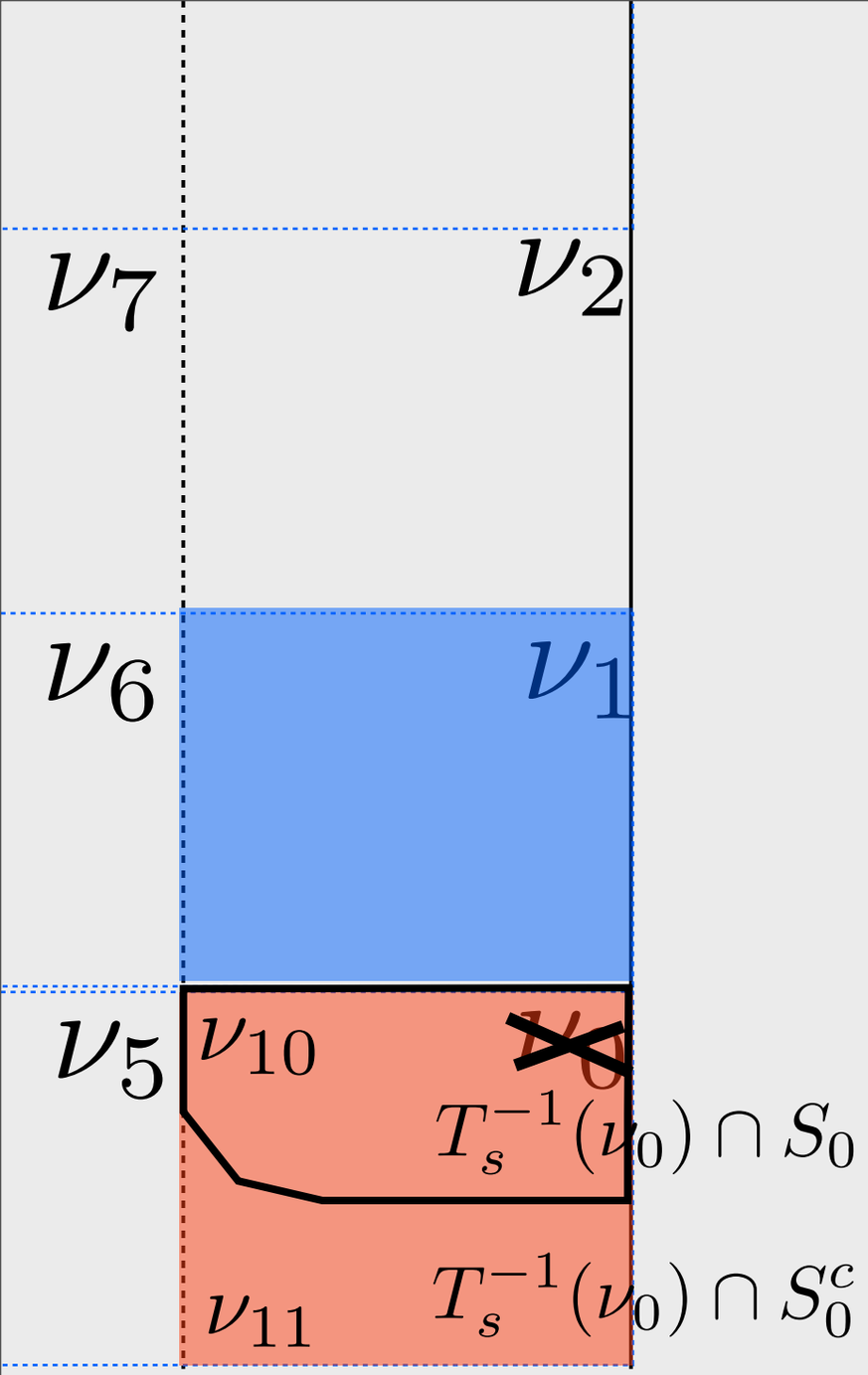
Put together: S_0 is computed as a polytope projection:

$$S_0 = \left\{ s_0 \in \mathbb{R}^n \ : \exists \hat{u} \in \mathbb{R}^{mN} \text{ s.t. } L \begin{bmatrix} s_0 \\ \hat{u} \end{bmatrix} \leq M - G\hat{d}, \ \forall \hat{d} \in \bar{D}^N \right\}$$

stacking of u and d

set of vertices of $D^N = D \times \cdots \times D$

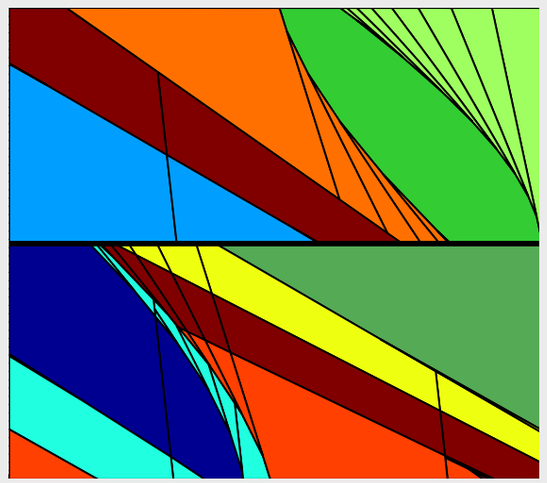
Refining the partition



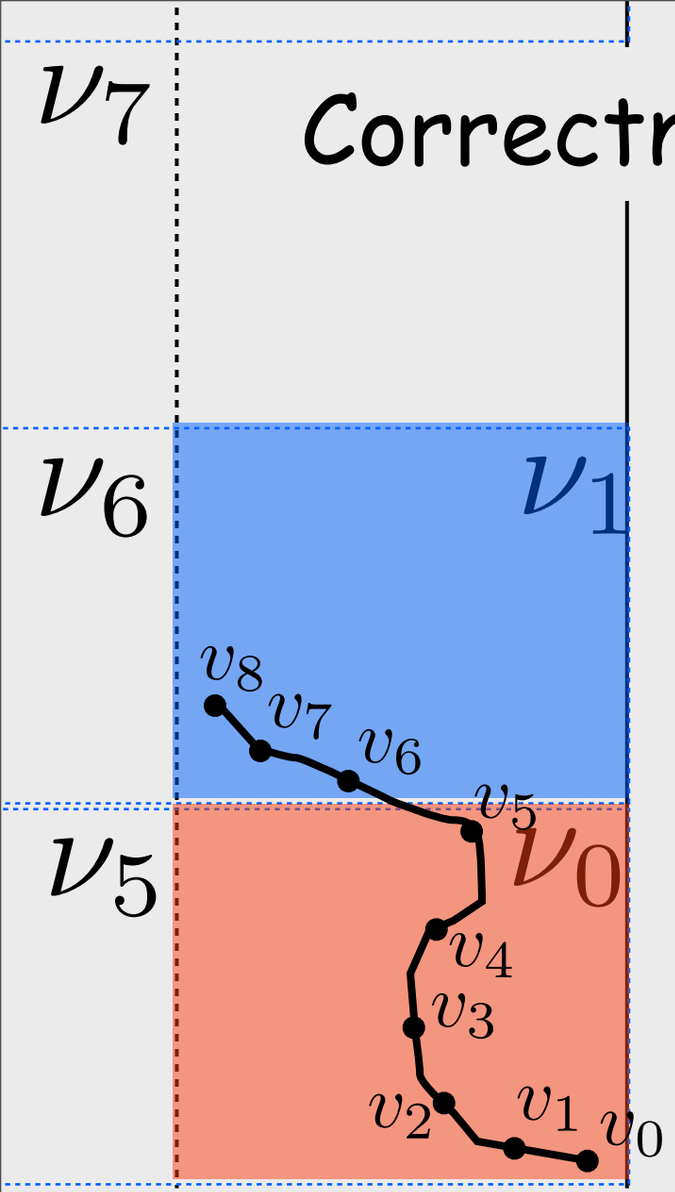
While checking the reachability from $T_s^{-1}(\nu_i)$ to $T_s^{-1}(\nu_j)$, if $T_s^{-1}(\nu_i) \not\subseteq S_0$, then
- Split $T_s^{-1}(\nu_i) \cap S_0$ and $T_s^{-1}(\nu_i) \cap S_0^c$
- Remove ν_i from the set of discrete states
- Add two new discrete states corresponding to $T_s^{-1}(\nu_i) \cap S_0$ and $T_s^{-1}(\nu_i) \cap S_0^c$

- Repeat until no cell can be sub-partitioned s.t. the volumes of the two resulting new cells both greater than Vol_{min} .
- Smaller Vol_{min} leads to more cells in the partition and more allowable transitions.
- If the initial partition is proposition preserving, so is the resulting.

Define the finite transition system $\mathbb{D}$, an abstraction of $\mathbb{S}$ as:
- $\mathcal{V} := \mathcal{S} \times \mathcal{E}$, set of discrete states (both controller and environment)
- $\nu_i = (\varsigma_i, \epsilon_i) \rightarrow v_j = (\varsigma_j, \epsilon_j)$ only if ς_j is reachable from ς_i.



Correctness of the hierarchical implementation



Using

- Proposition preserving property of the partition
- $\mathbb{D}$ only includes the transitions that are implemented by the control signal u within some finite time (by construction through the reachability formulation)
- Stutter invariance of the specification φ, ...

> Two words σ_1 and σ_2 over 2^{AP} are stutter equivalent, if there exists an infinite sequence $A_0 A_1 A_2 \ldots$ of sets of atomic propositions and natural numbers $n_0, n_1, n_2, \ldots$ and $m_0, m_1, m_2, \ldots$ such that σ_1 and σ_2 are of the form
>
> $$\sigma_1 = A_0^{n_0} A_1^{n_1} A_2^{n_2} \ldots \qquad \sigma_2 = A_0^{m_0} A_1^{m_1} A_2^{m_2} \ldots$$
>
> An LT property P is stutter-invariant if for any word $\sigma \in P$ all stutter-equivalent words are also contained in P.

Example: $v_0 v_1 \ldots v_8 \ldots$ **and** $\nu_0 \nu_1 \ldots$ **are stutter-equivalent.**

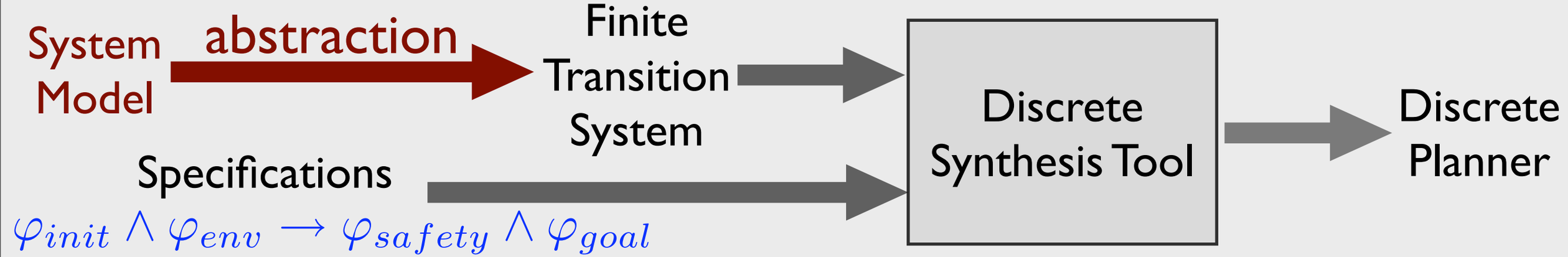
...we can prove:

Let $\sigma_d = \nu_0 \nu_1 \ldots$ be a sequence in $\mathbb{D}$ with $\nu_k \rightarrow \nu_{k+1}$, $\nu_k = (\varsigma_k, \epsilon_k)$, $\varsigma_k \in \mathcal{S}$ and $\epsilon_k \in \mathcal{E}$. If $\sigma_d \models_d \varphi$, then by applying a sequence of control signals from the Reachability Problem with initial set $T_s^{-1}(\varsigma_k)$ and final set $T_s^{-1}(\varsigma_{k+1})$, the sequence of continuous states $\sigma = v_0 v_1 v_2 \ldots$ satisfies φ.

How to use abstractions for synthesis?

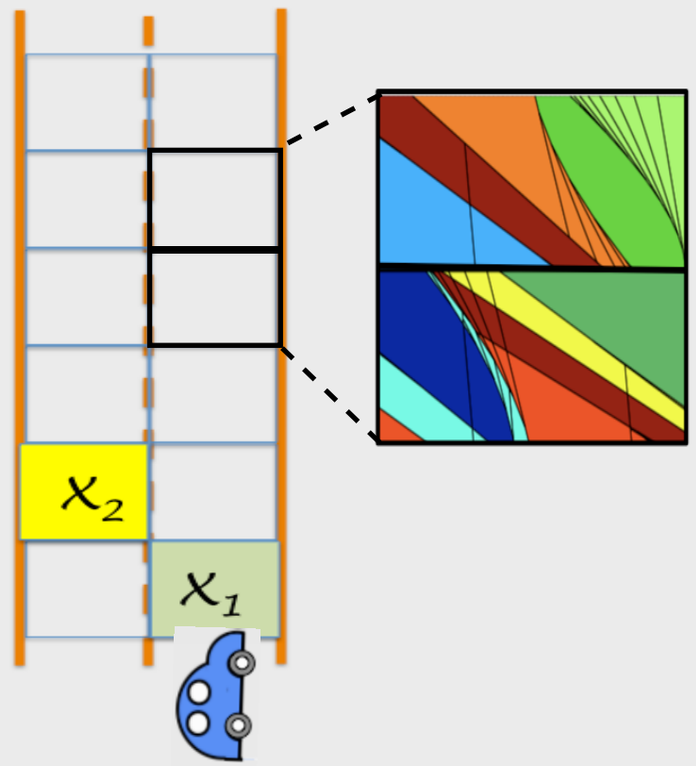
Existence of a continuous control action that implements the discrete transition (projection)

Control action itself (finite-time optimal control problem)



Starting with a proposition preserving partition:

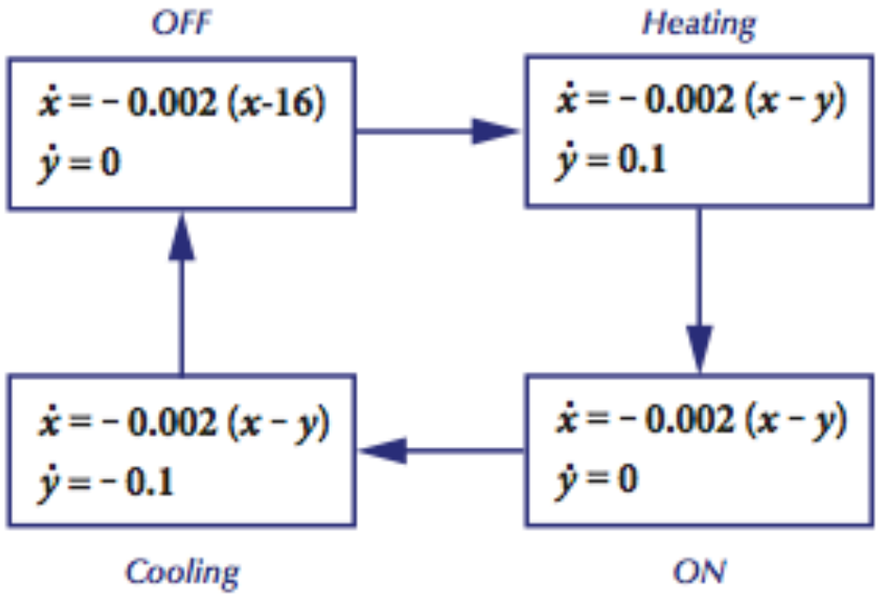
- Finite-time reachability to determine discrete transitions
- Refine the partition to increase the number of valid discrete transitions

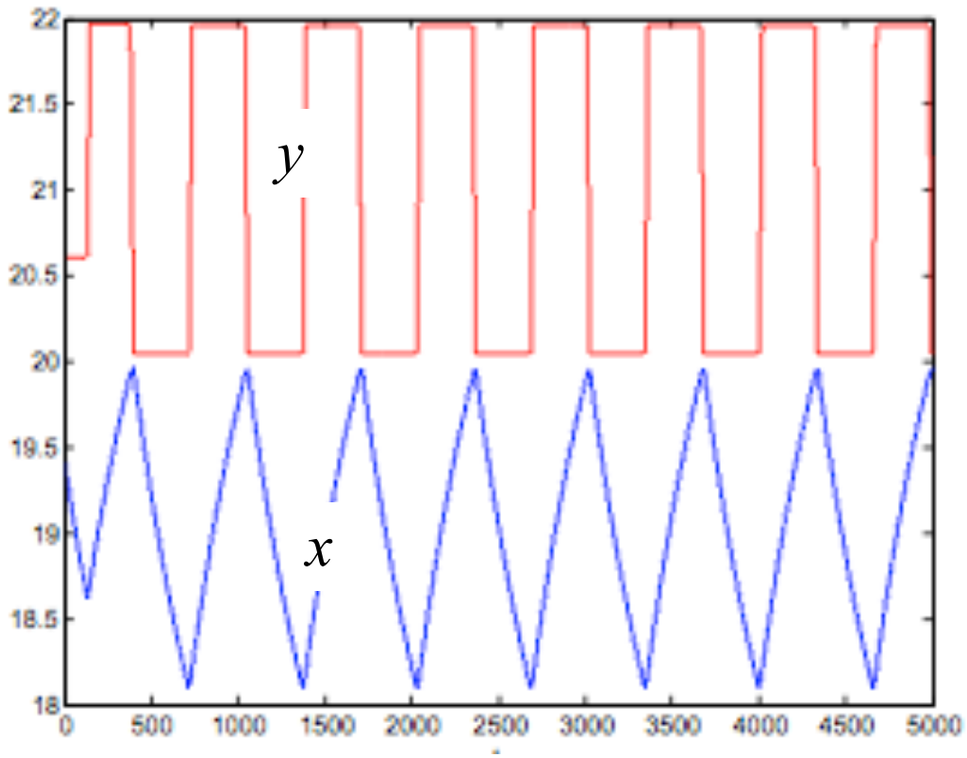


Example: synthesis

A four-mode thermostat:

x: room temperature, y: heater temp

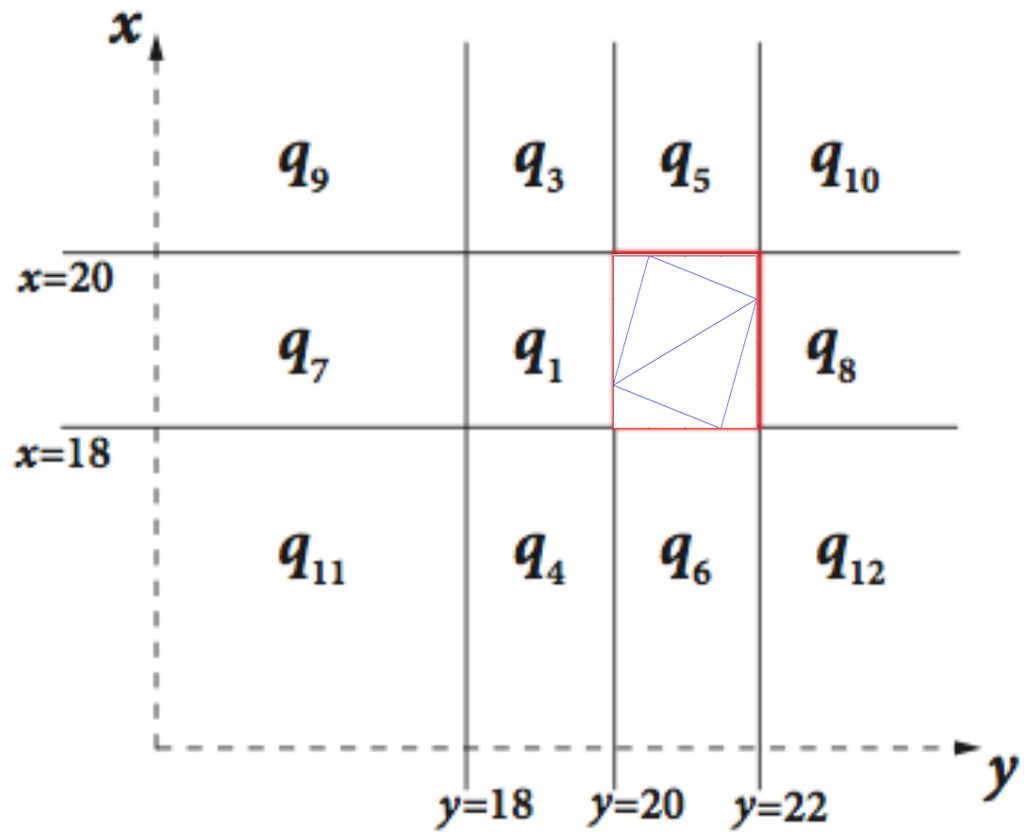




Find a switching sequence such that:

$$(18 \le x \le 20 \wedge 20 \le y \le 22) \rightarrow \Box(18 \le x \le 20 \wedge 20 \le y \le 22))$$

Construct an over-approximation using the partition in the figure below.



States in the finite-state abstraction:

$(q_i, mode)$

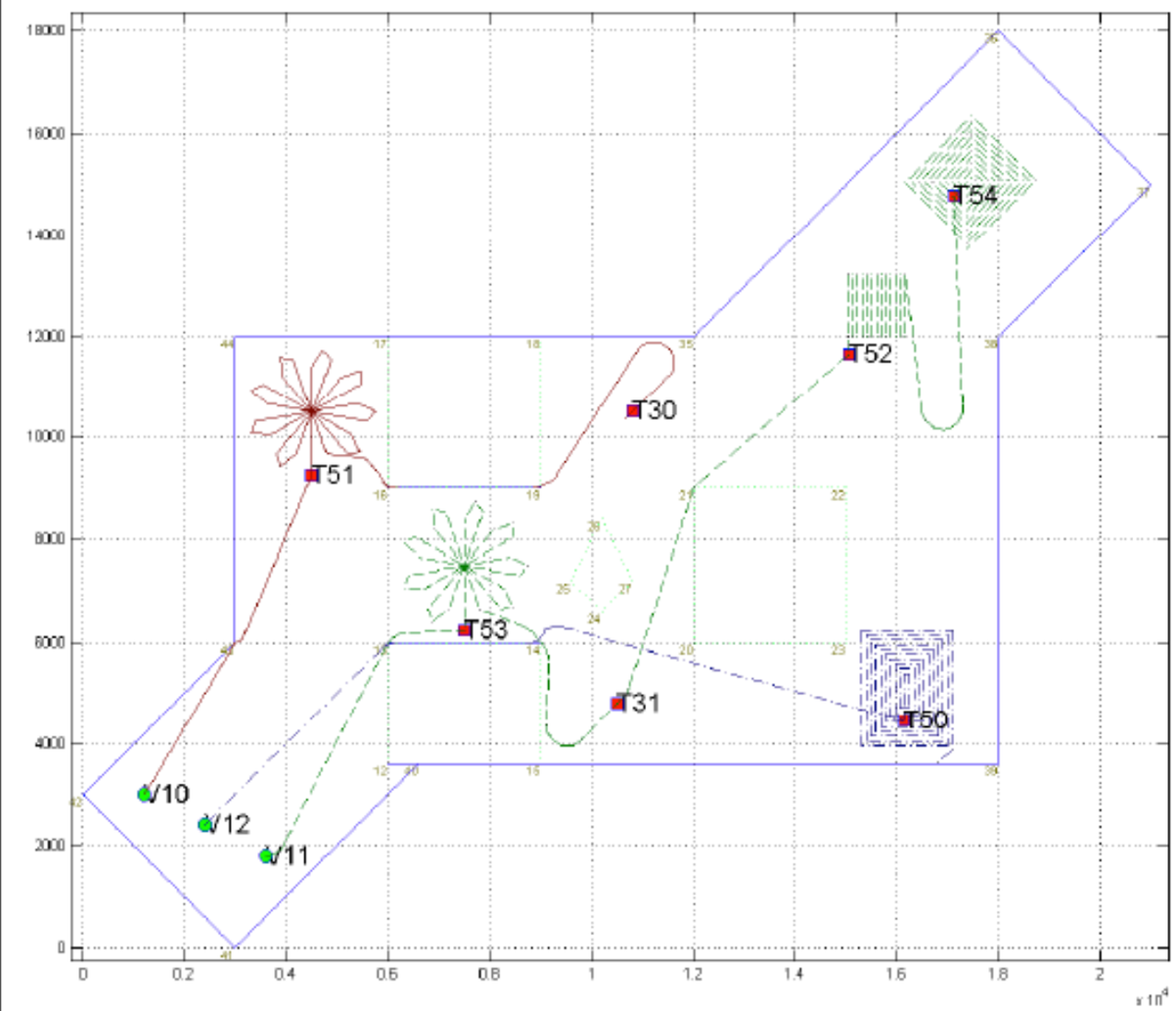
$mode \in \{off, heating, on, cooling\}$

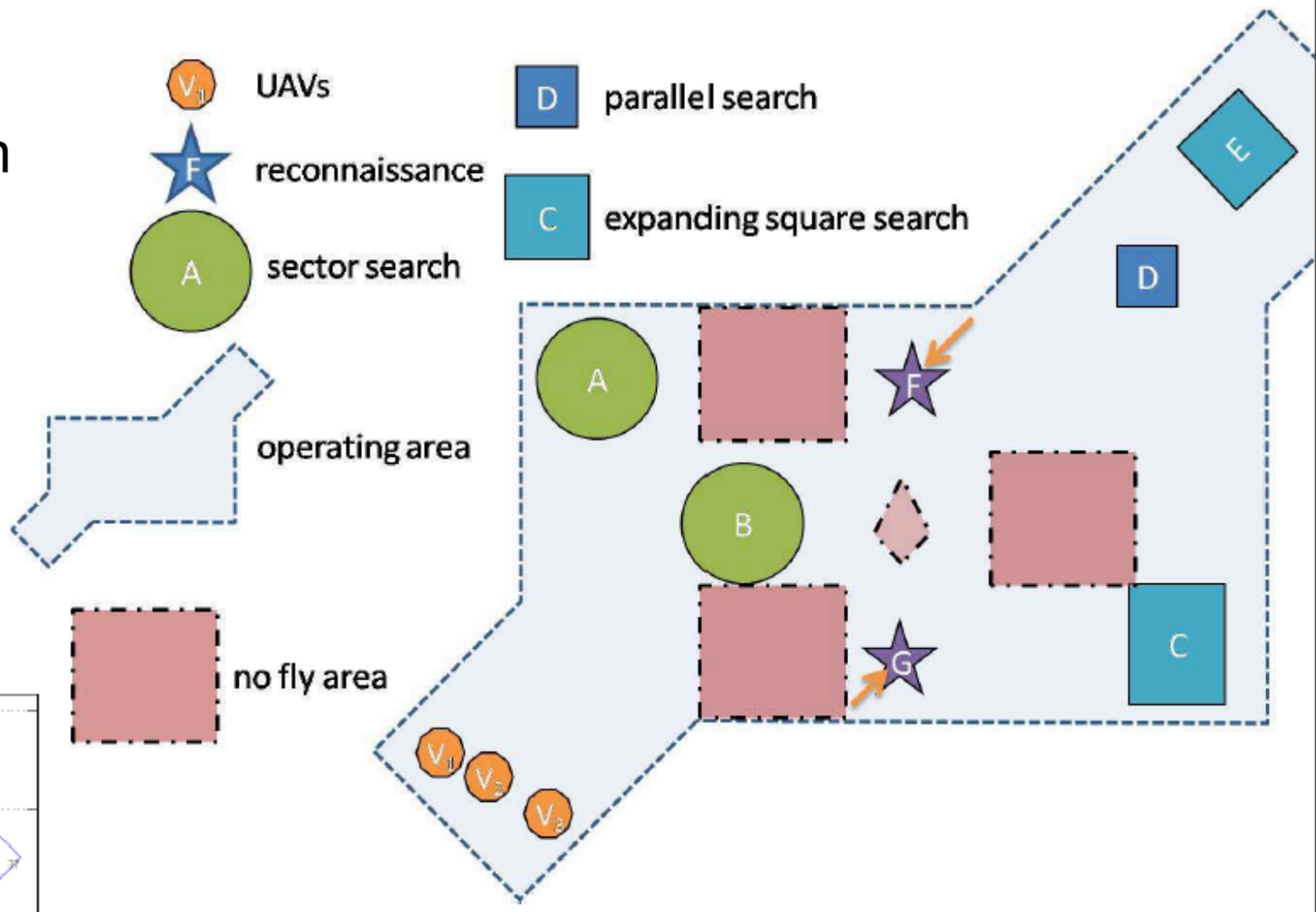
Abstractions using primitives

A task-level abstraction of the system using a library of primitives (low-level controllers) for

- executing tasks and
- transitioning between tasks

Figures from “Assignment of Heterogeneous Tasks to a Set of Heterogeneous Unmanned Aerial Vehicles,” Rasmussen & Kingston (AFRL).

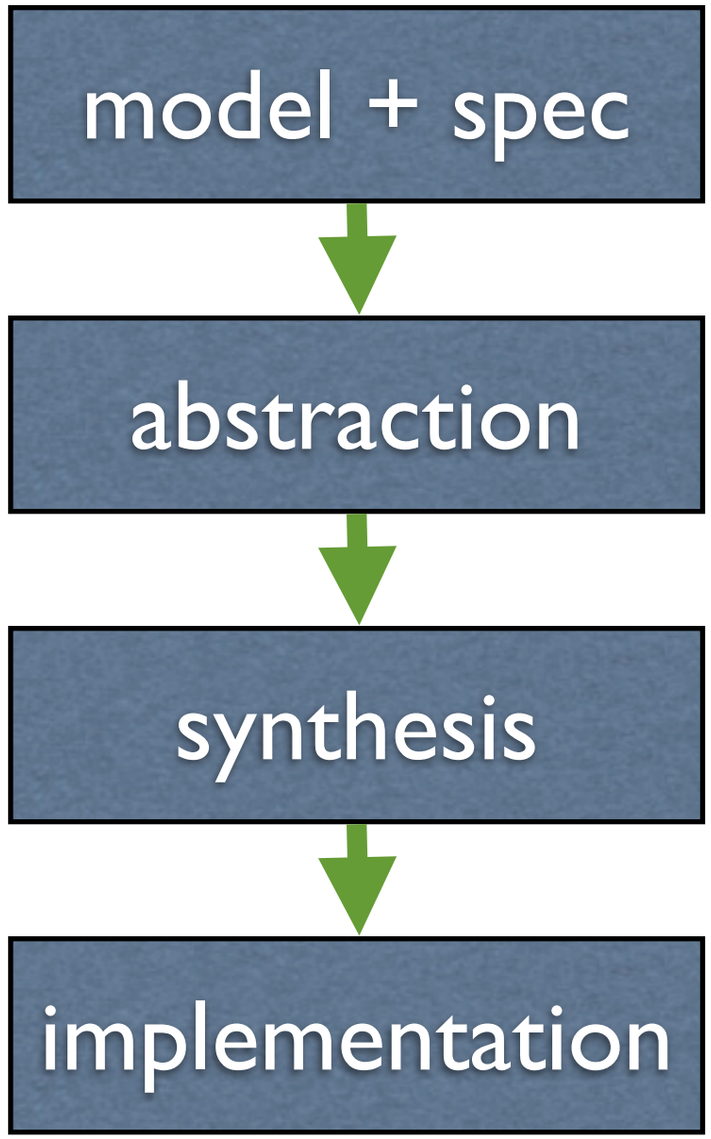




- The level of abstraction dictates the level at which it can be specified.
- Task-level abstraction allows task-level specifications, e.g.
  - never enter no-fly-area
  - every reconnaissance is eventually followed by a search
  - pop-up tasks have priority

Abstraction-based hierarchical control design

- Given $\dot{x} = f(x, u)$ and LTL formula φ.
- Compute finite-state, proposition preserving approximations.
- Solve a discrete synthesis problem and obtain a discrete control strategy.
- Implement the discrete control strategy to ensure that all trajectories of $\dot{x} = f(x, u)$ satisfies φ.



Issues:

- What approximations are appropriate and how to compute them?
- What discrete synthesis problems to solve and how to solve them?

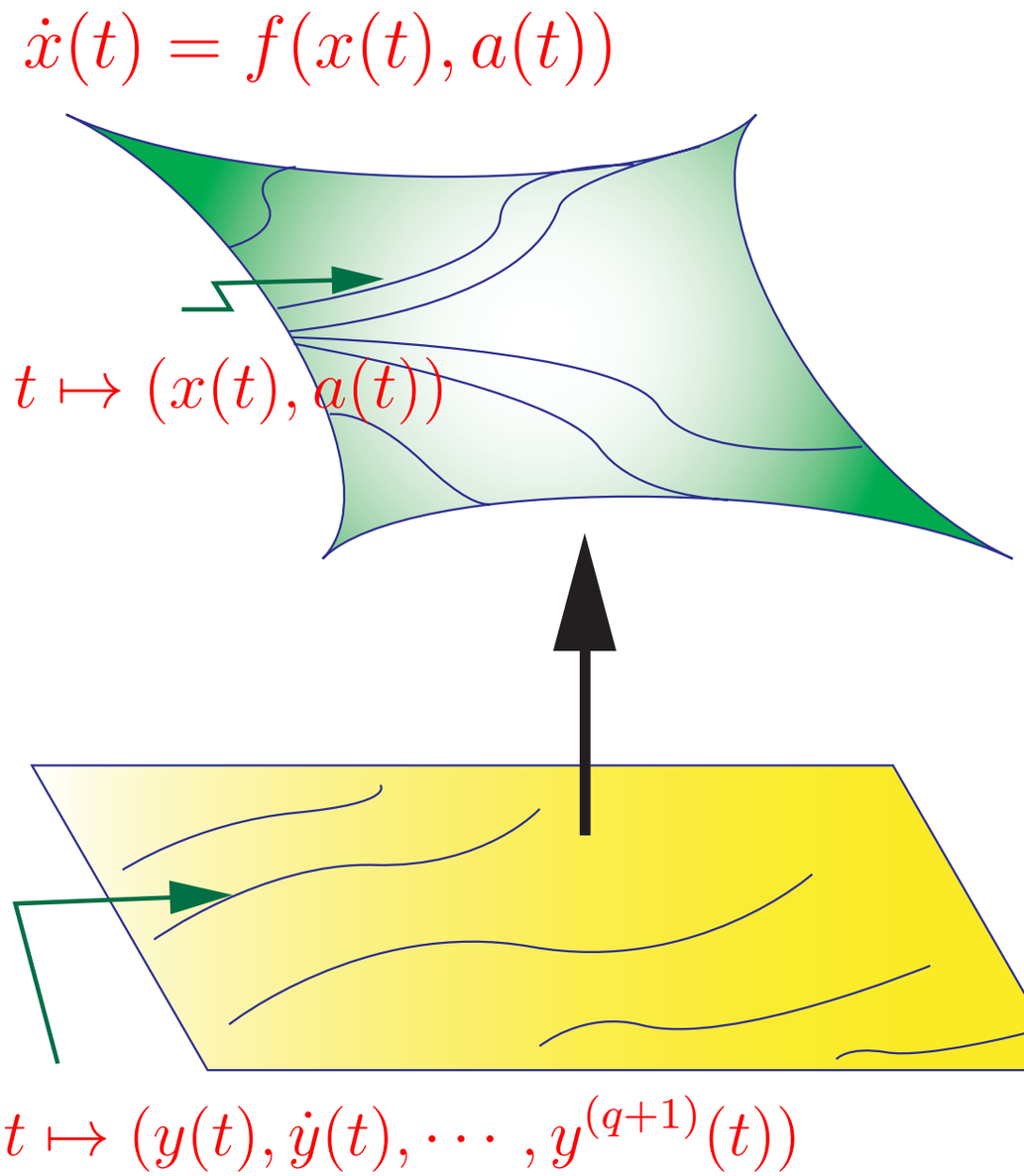
Motion planning for flat systems

- Flat systems:

$$\dot{x} = f(x, a), \quad y = h(x),$$

$$x = \Gamma(y, \dot{y}, \cdots, y^{(q)}),$$

$$a = \Theta(y, \dot{y}, \cdots, y^{(q+1)}).$$



- To every curve t↦y(t) that is sufficiently smooth, there corresponds a trajectory

$$t \mapsto \begin{bmatrix} x(t) \\ a(t) \end{bmatrix} = \begin{bmatrix} \Gamma(y(t), \dot{y}(t), \cdots, y^{(q)}(t)) \\ \Theta(y(t), \dot{y}(t), \cdots, y^{(q+1)}(t)) \end{bmatrix}$$

that identically satisfies the system equation.

Slide courtesy of Jun Liu

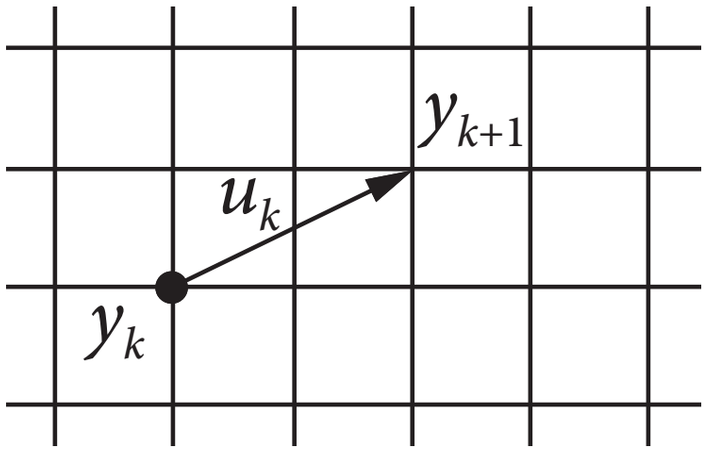
Construction of finite abstraction

- Let $\mathbb{Z}^m$ denote the integral lattice in $\mathbb{R}^m$ and define

$$U_\mu := \mu\mathbb{Z}^m \cap U, \quad Y_\eta := \eta\mathbb{Z}^m \cap Y, \quad \mu\mathbb{Z}^m := \{\mu z : z \in \mathbb{Z}^m\}$$

  where U and Y are compacts in $\mathbb{R}^m$ and $\mu > 0,\ \eta > 0.$

- Define a *labelled transition system* $TS = (Y_\eta, U_\mu, \rightarrow)$ by



$$\left(\begin{array}{ccc} k & k & k+1 \end{array} \right)$$

if and only of

$$u_k = (y_{k+1} - y_k)/\tau$$

  where $\tau = \eta/\mu.$

- The abstraction is trivial in the sense that it connects all grid points as long as they can be reached at a "feasible rate".

Slide courtesy of Jun Liu

Continuous-time implementation

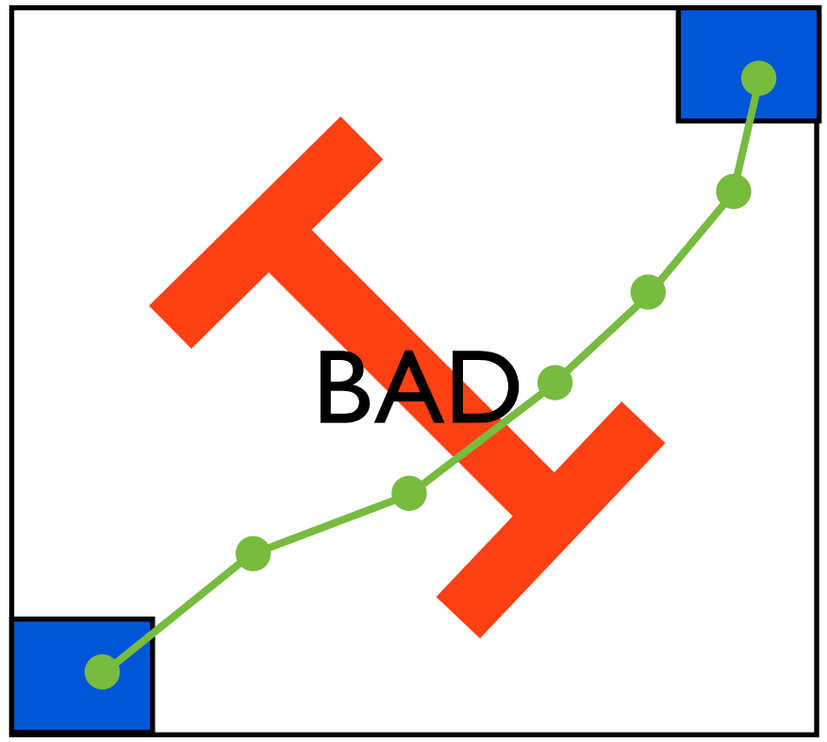
- Consider a sequence $y_0, y_1, y_2, \cdots$ such that $y_0 \models \varphi$.
- Suppose *y*(*t*) is a continuous-time output trajectory such that

$$y(0) = y_0,\ y(\tau) = y_1,\ \cdots,\ y(k\tau) = y_k\ , \cdots$$

- Does this output trajectory satisfy $y(0) \Vdash \varphi$?

Correctness of y(t) depends on

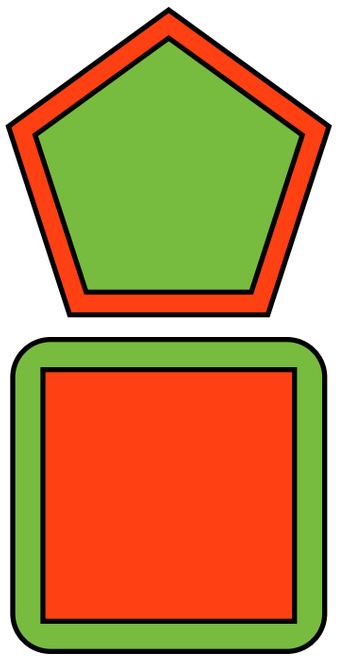
- Size of sampling time.
- Rate of dynamics.
- How "robust" does $y_0 \models \varphi$?



Robust interpretation of LTL formulas

- We use a notion of robustness for an LTL formula to be satisfied (Fainekos et al. (2009)).

ε-contraction and ε-inflation of atomic propositions (sets)



$$[\![\pi_\varepsilon]\!] := \{z \in [\![\pi]\!] : \; z + \varepsilon\mathbb{B} \in [\![\pi]\!]\}$$

$$[\![\pi^\varepsilon]\!] \; := \; [\![\pi]\!] + \varepsilon\mathbb{B}$$

ε-contraction φ^ε (ε-inflation φ_ε) of an LTL formula φ

- Write any given LTL formula φ in Negation Normal Form (NNF).
- Treat negations of atomic propositions as new atomic propositions.
- Replace all atomic propositions by their ε-contraction.

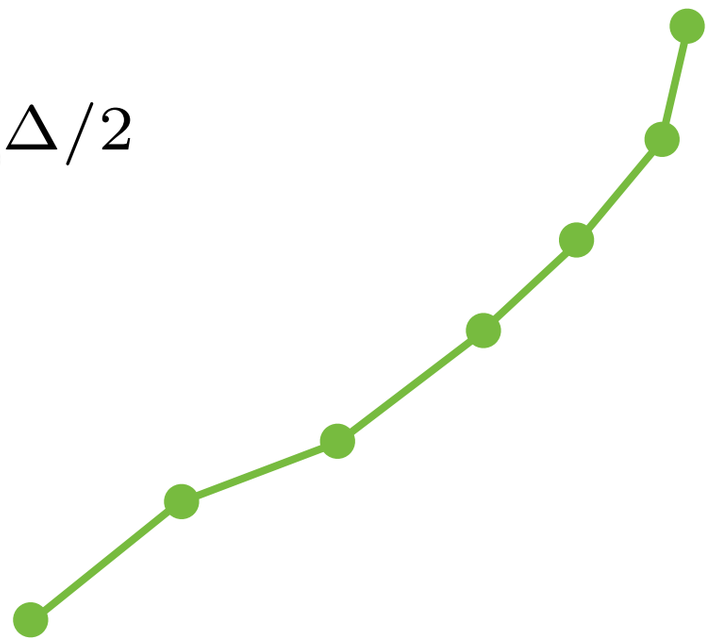
"φ^ε *is satisfied" implies "*φ *is satisfied with a robustness margin* ε".

Reasoning of correctness between sequences and trajectories

Sequence to trajectory:

y(t): linear interpolation at y_k's

$$\left.\begin{array}{l}\sup\limits_{k\geq 0}|y_{k+1}-y_k|\leq\Delta\\ y_0\vDash\varphi\end{array}\right\}\Longrightarrow\ y(0)\Vdash\varphi^{\Delta/2}$$



Trajectory to sequence:

y(t) is Lipschitz with constant γ

$$\left.\begin{array}{l}y_k=y(k\tau)\\ y(0)\Vdash\varphi\end{array}\right\}\Longrightarrow y_0\vDash\varphi^{\gamma\tau/2}$$

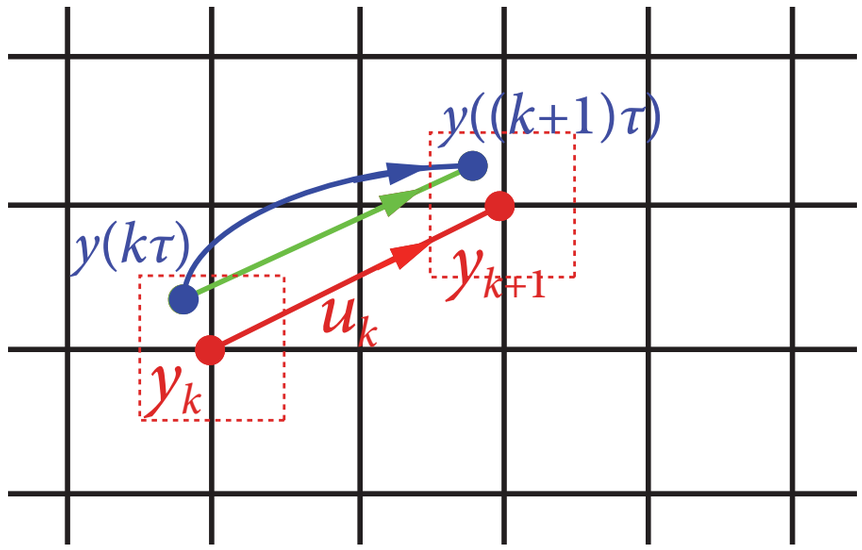
Generating correct trajectories for flat systems

Consider the following trajectories:

$\tilde{y}(t)$: linear interpolation at y_k's

$\hat{y}(t)$: linear interpolation at $y(k\tau)$'s

$y(t)$: the output to be generated



Polynomial basis:

$$y(t) = \sum_{i=0}^{2q+1} c_i (t - k\tau)^i / \tau^i, \quad t \in [k\tau, (k+1)\tau]$$

Error estimate:

$$|y(t) - \hat{y}(t)| \leq C\tau, \quad t \in [k\tau, (k+1)\tau]$$

Correctness implication:

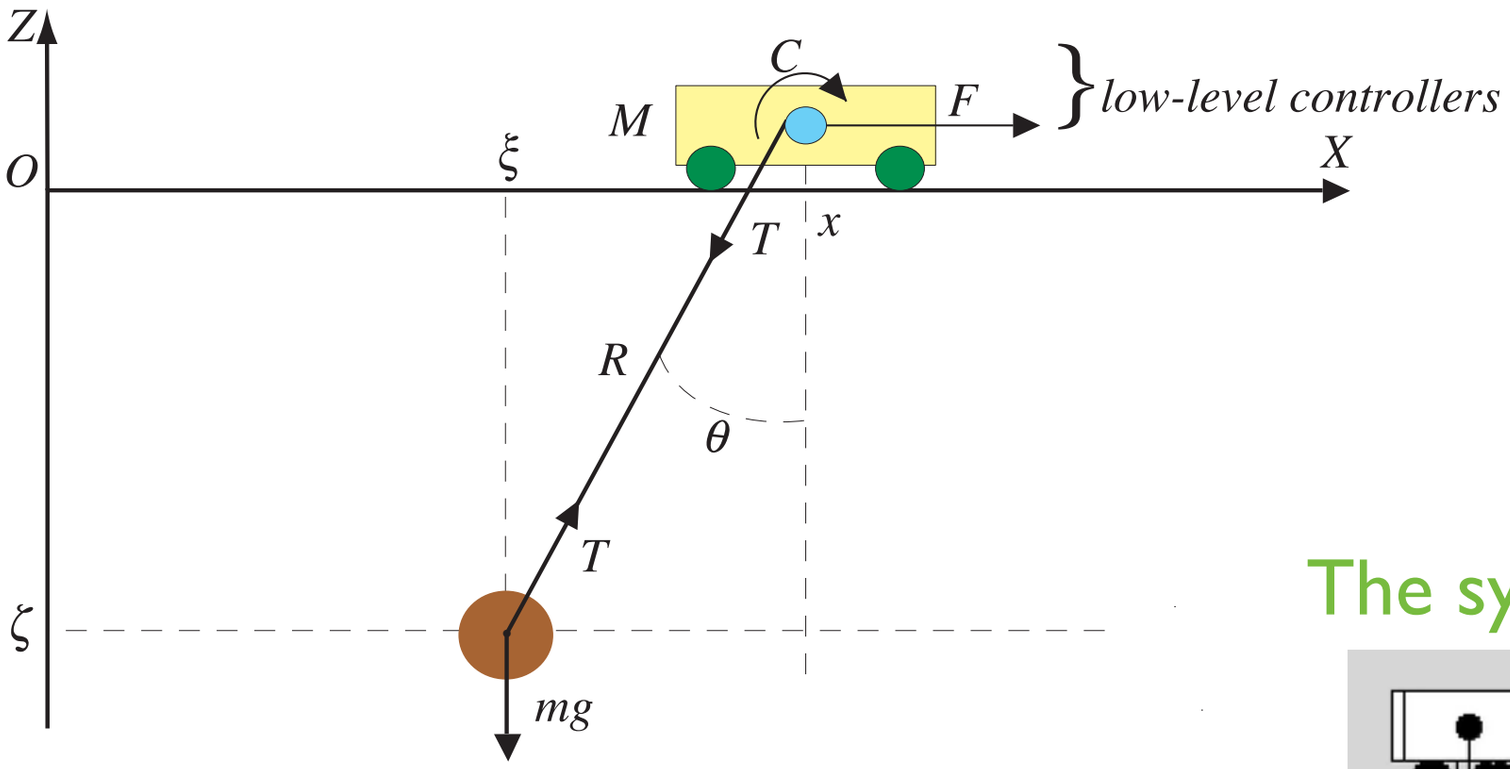
$$y_0 \models \varphi_\delta \Longrightarrow y(0) \Vdash \varphi_\varepsilon \Longrightarrow y(0) \Vdash \varphi$$

provided that

$$0 < \varepsilon \leq \delta - u_{\max}\tau/2 - \eta/2 - C\tau, \quad |u| \leq u_{\max} \text{ for all } u \in U_\mu$$

Control of 2-D overhead crane

Overhead crane (Levine, 2009)



(ξ, ζ)	position of load
x	position of trolley
R	length of rope
T	tension of rope
θ	angle of rope
F	force
C	torque

Equation of motion

$$m\ddot{\xi} = -T\sin\theta$$
$$m\ddot{\zeta} = T\cos\theta - mg$$
$$M\ddot{x} = -\gamma_1(\dot{x}) + F + T\sin\theta$$
$$\frac{J}{\rho}\ddot{R} = -\gamma_2(\dot{R}) - C + T\rho$$
$$\xi = x + R\sin\theta$$
$$\zeta = -R\cos\theta.$$

The system has flat outputs (ξ, ζ).