# Lecture 8
# Receding Horizon Temporal Logic Planning & Finite-State Abstraction

Ufuk Topcu

Nok Wongpiromsarn Richard M. Murray

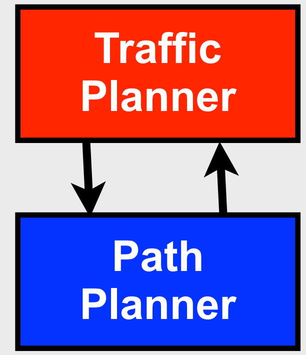
AFRL, 26 April 2012

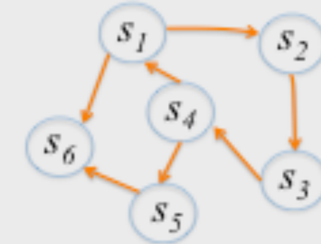
Contents of the lecture:

- Intro: Incorporating continuous dynamics & sources of computational complexity
- Recall: Receding horizon control
- Receding horizon temporal logic planning (RHTLP)
  - Basic idea & main result
  - Discussion of the key details of implementation
  - Autonomous driving examples
- Finite-state abstraction & hierarchical control architecture

# Problem: Design control protocols, that...

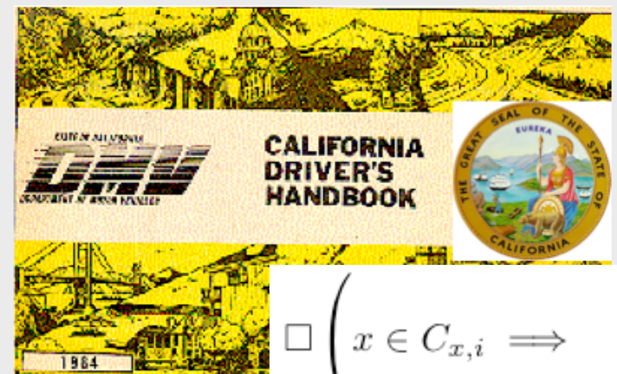
Handle mixture of discrete and continuous dynamics




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

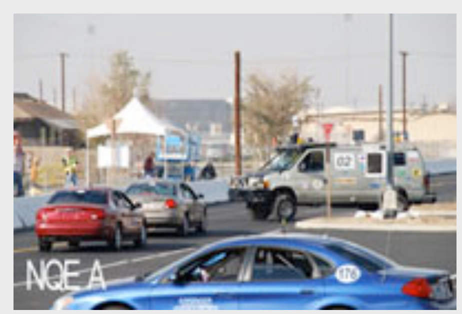
Account for both high-level specs and low-level constraints




$$g(x, u) \geq 0$$

$$\square\left(x \in C_{x,i} \implies \bigwedge_{j > i + d_{sr}} (\neg O_{j,1} \wedge \neg O_{j,2})\right)$$

Reactively respond to changes in environment,

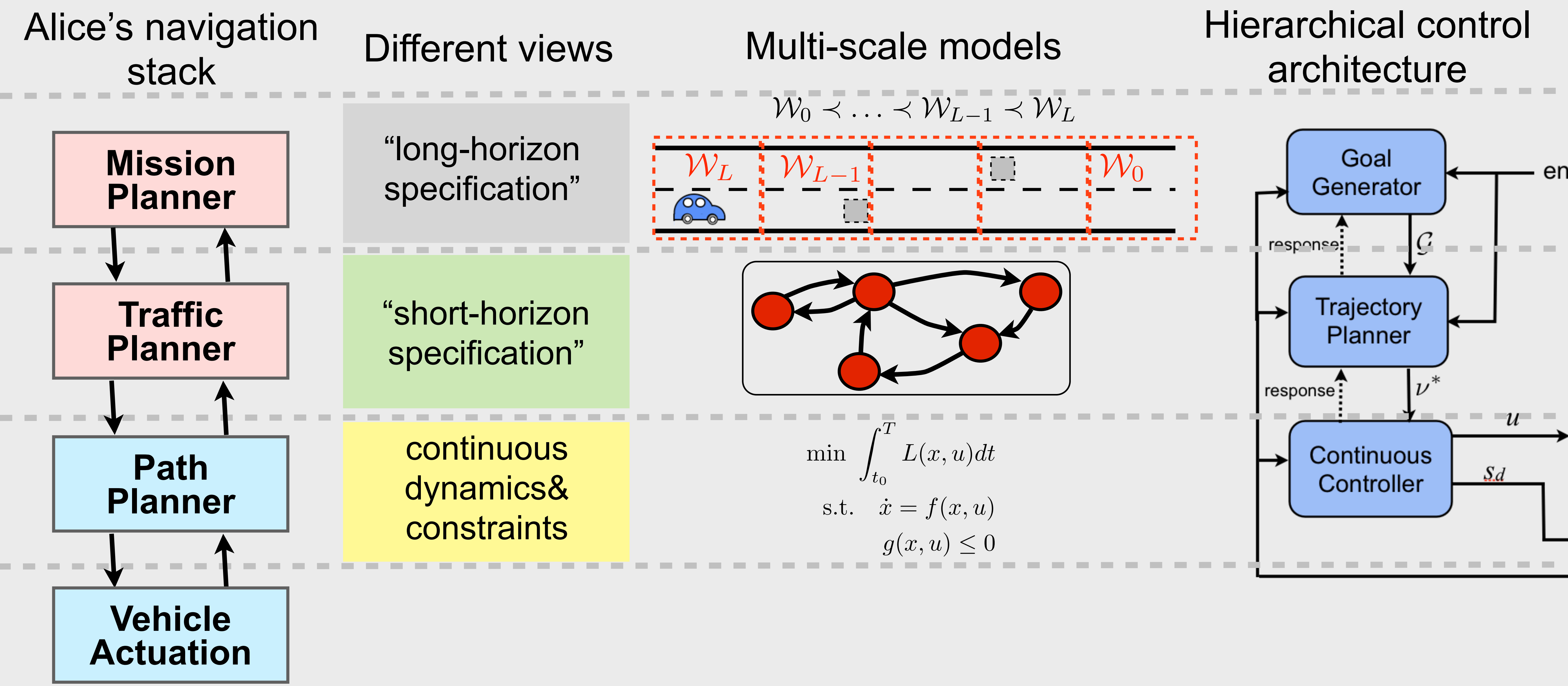


$$\square\left(\left(x \in \bigcup_{j=i-1}^{i+1} C_{x,j} \wedge O_{i,1}\right) \implies \square(O_{i,1})\right)$$

... with "correctness certificates."

$$\left[(\varphi_{init} \wedge \varphi_{env}) \rightarrow (\varphi_{safety} \wedge \varphi_{goal})\right]$$

# Preview



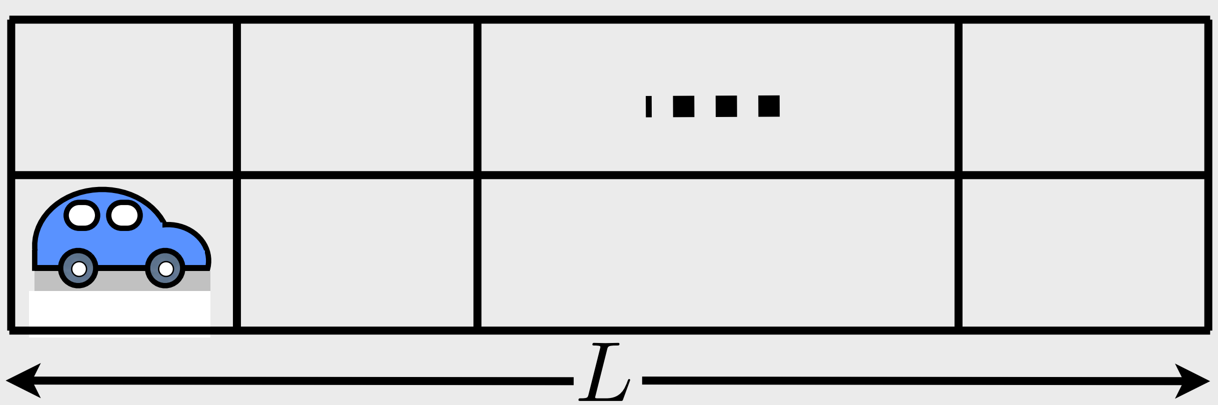

TuLiP: Temporal logic planning toolbox
(Open source at http://tulip-control.sf.net)

[Coming up in the next lecture]

# This lecture focuses on two of the remaining issues:

- Incorporating continuous dynamics
- Computational complexity

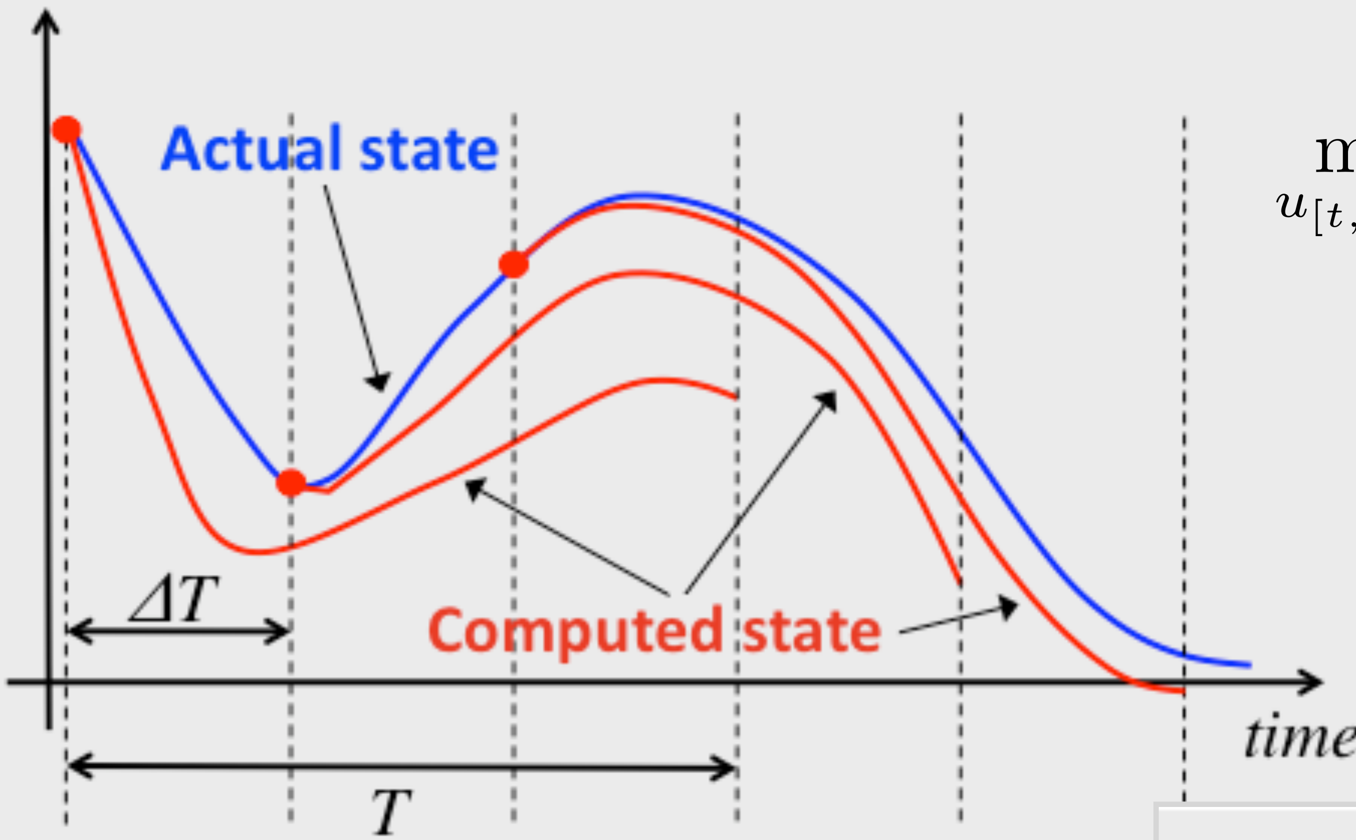
# Computational Complexity


- Each of these cells may be occupied by an obstacle.
- The vehicle can be in any of these cells.

$(2L)(2^{2L})$ possible states!

# Receding Horizon Control

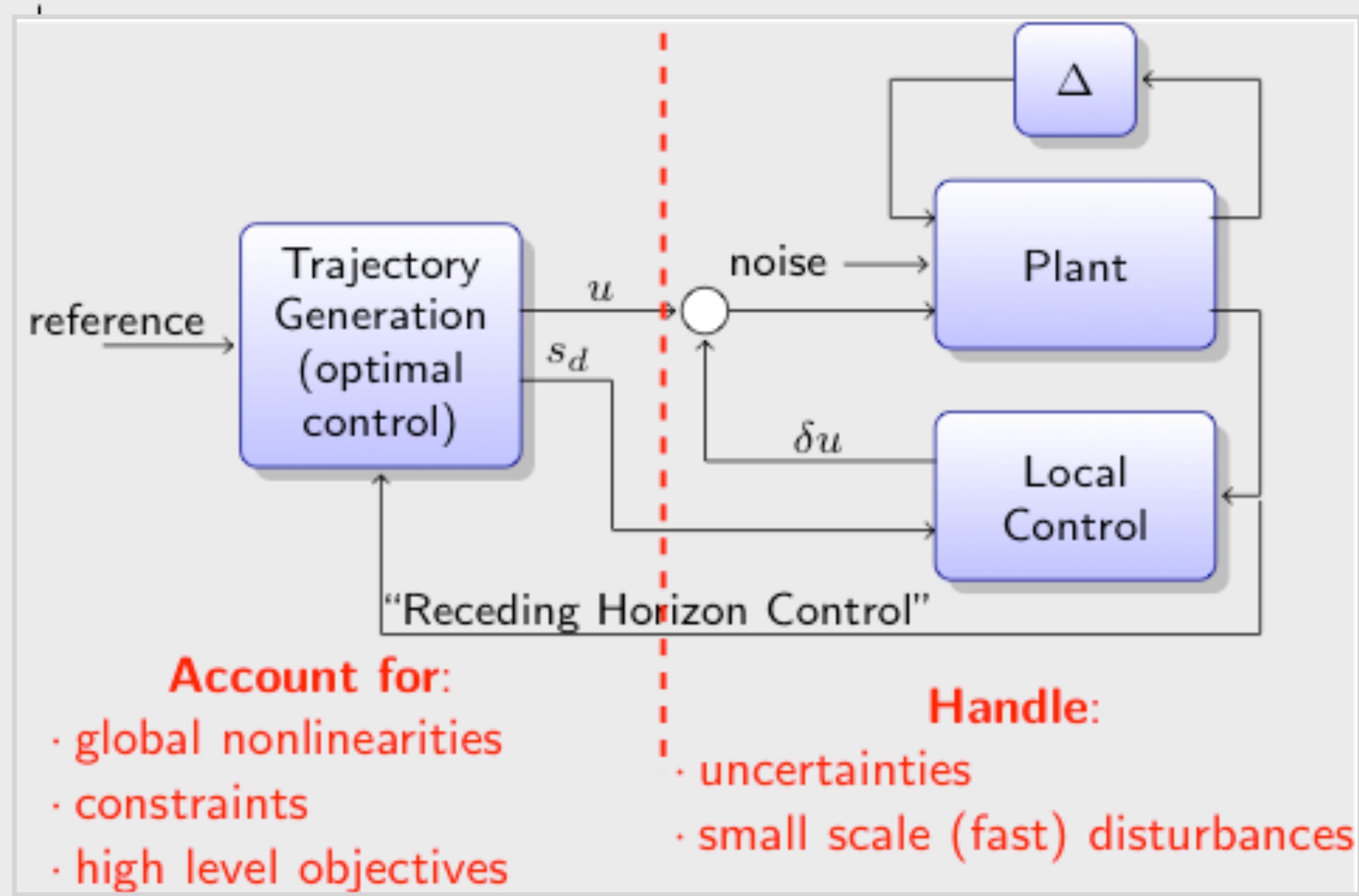



$$\min_{u_{[t,t+T]}} \int_t^{t+T} C(x(\tau), u)\tau))d\tau + V(x(t+T))$$

subject to:

$$\dot{x} = f(x,u), \quad x(t) \text{ given}$$
$$x(t+T) = x_f, \quad g(x,u) \leq 0$$

- Reduces the computational cost by solving smaller problems.
- Real-time (re)computation improves robustness.



# Receding Horizon Control

- If not implemented properly, global properties, e.g., stability, are not guaranteed.
- Increasing $T$ helps for stability at the expense of increased computational cost.

finite-horizon optimization

terminal cost

$$\min_{u_{[t,t+T]}} \int_t^{t+T} C(x(\tau), u)\tau))d\tau + V(x(t+T))$$

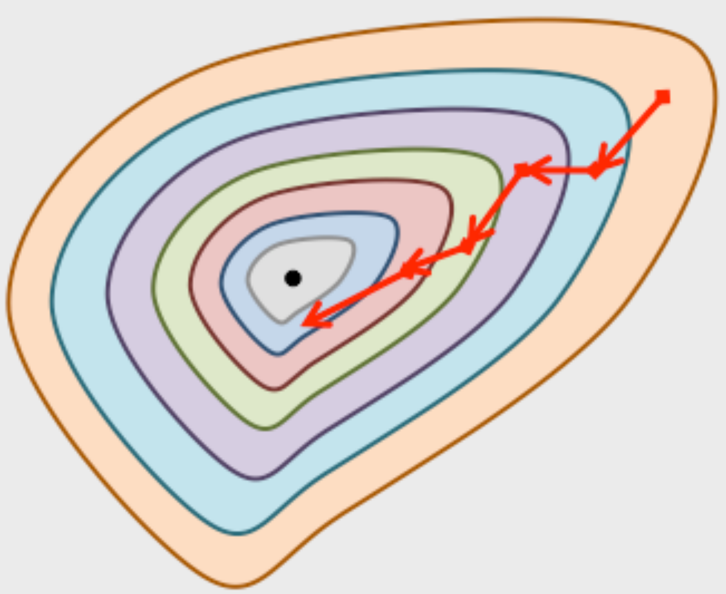
subject to:

$$\dot{x} = f(x,u), \quad x(t) \text{ given}$$
$$x(t+T) = x_f, \quad g(x,u) \leq 0$$

- If the terminal cost is chosen as a control Lyapunov function, i.e., V is (locally) positive definite and satisfy (for some r>0)

$$\min_u (\dot{V} + C)(x,u) < 0, \ \forall x \in \{x : V(x) \leq r^2\}$$

then stability is guaranteed.



- Alternative (related) approach, imposed contractiveness constraints in short-horizon problems.

# Receding Horizon for LTL Synthesis

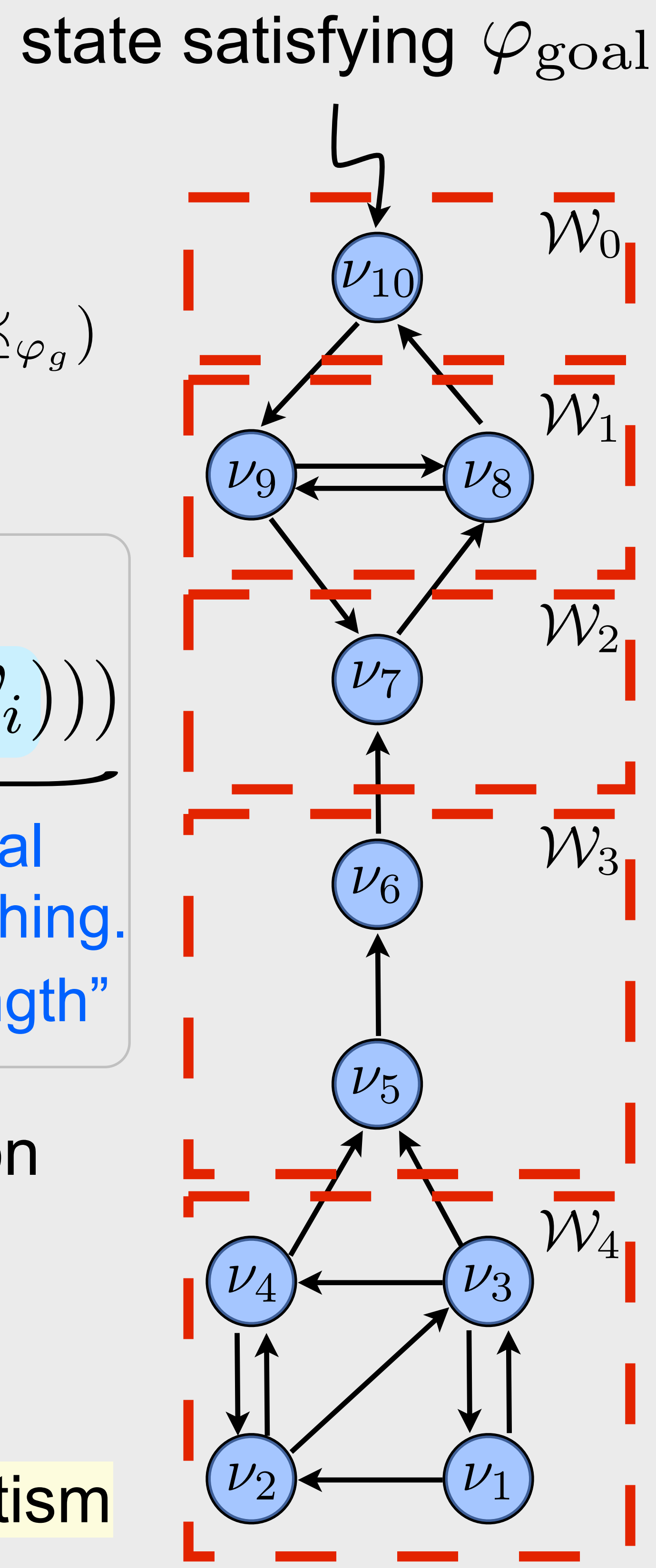
[TAC'11(submitted), HSCC'10]

**Global (long-horizon) specification:**

$$(\varphi_{\text{init}} \wedge \varphi_{\text{env}}) \rightarrow (\varphi_{\text{safety}} \wedge \varphi_{\text{goal}})$$

**Basic idea:**

- Partition the state space into a partially ordered set $(\{\mathcal{W}_j\}, \preceq_{\varphi_g})$
- Goal-induced partial order




**Short-horizon specification:** For each $i$,

$$((\nu \in \mathcal{W}_i) \wedge \Phi \wedge \varphi_{\text{env}}) \rightarrow (\Box\Phi \wedge \varphi_{\text{safety}} \wedge \Diamond(\nu \in \mathcal{F}_i(\mathcal{W}_i)))$$

Plan from the current cell on

Receding horizon invariant: rules out "corner" cases

Get closer to goal rather than reaching. $\mathcal{F}$: "horizon" length"

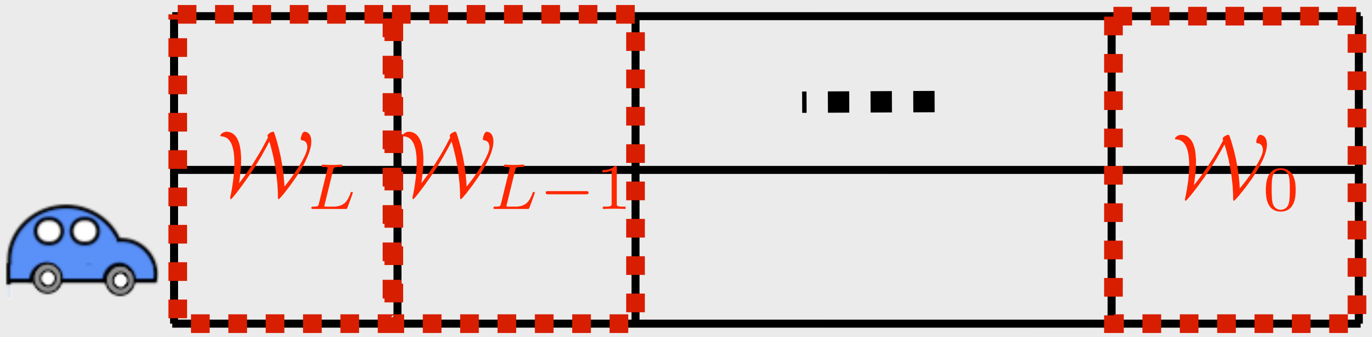
**Theorem:** Receding horizon implementation of the short-horizon strategies ensures the correctness of the global specification.

**Trade-offs:**

computational cost vs. horizon length vs. strength of invariant vs. conservatism

# How to come up with a partial order, $\mathcal{F}$ and $\Phi$?

- In general, problem-dependent and requires user guidance.
- Partial automation is possible (discussed later).
- Partial order: "measure of closeness" to the goal, i.e, to the states satisfying.
- The map $\mathcal{F}$ determines the "horizon length.




$$\mathcal{W}_0 \prec \ldots \prec \mathcal{W}_{L-1} \prec \mathcal{W}_L$$

$$\mathcal{F}(\mathcal{W}_j) = \mathcal{W}_{j-2}, \quad j \geq 2$$
$$\mathcal{F}(\mathcal{W}_j) = \mathcal{W}_0, \quad j < 2$$

- The invariant $\Phi$ (in this example) rules out the states that render the short horizon problems unrealizable.
- In the example above, it is the conjunction of the following propositional formulas on the initial states for each subproblem:
  - no collision in the initial state
  - vehicle cannot be in the left lane unless there is an obstacle in the right lane in the initial state
  - vehicle is able to progress from the initial state

# Navigation of point-mass omnidirectional vehicle

nondimensionalized dynamics:

$$\ddot{x} + \dot{x} = q_x(t)$$

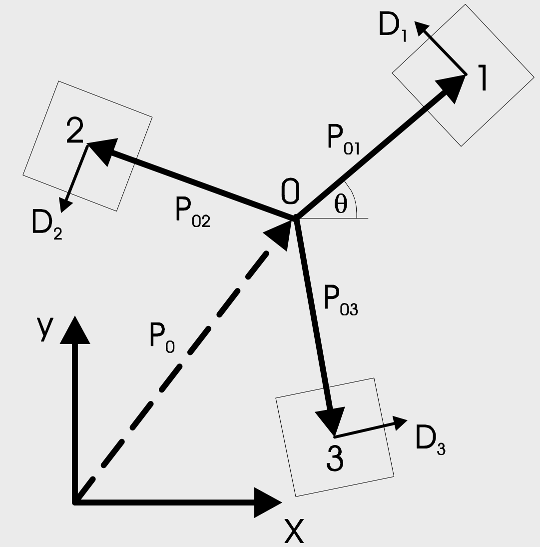
$$\ddot{y} + \dot{y} = q_y(t)$$

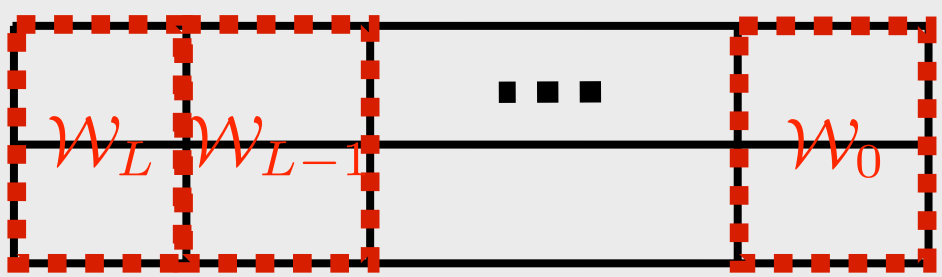
$$\ddot{\theta} + \frac{2mL^2}{J}\dot{\theta} = q_\theta$$

conservative bounds on control authority to decouple the dynamics:

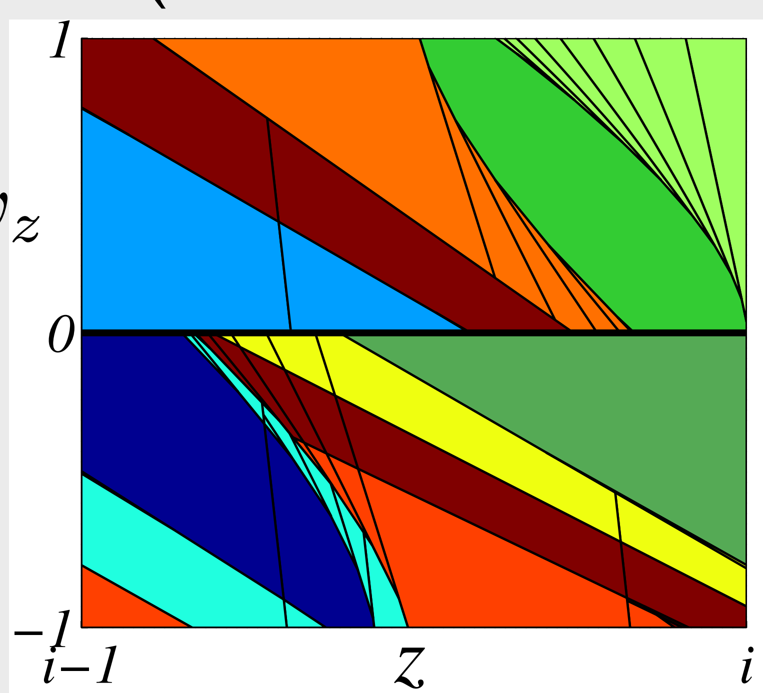
$$|q_x(t)|, |q_y(t)| \leq \sqrt{0.5}$$

$$|q_\theta(t)| \leq 1$$



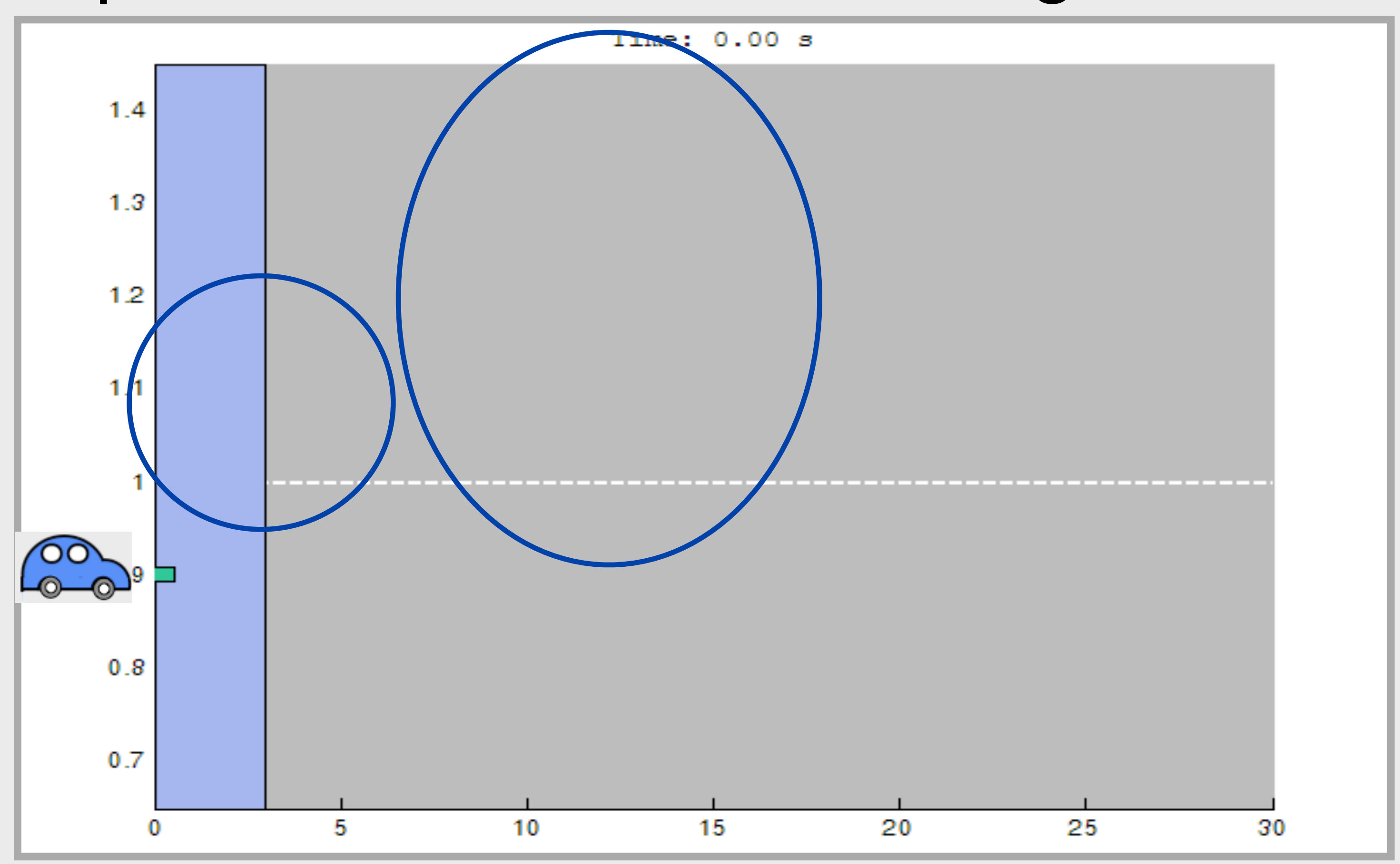




Partition (in two consecutive cells):




Reasons for the non-intuitive trajectories:

- Synthesis: feasibility rather than "optimality."
- Specifications are not rich enough.

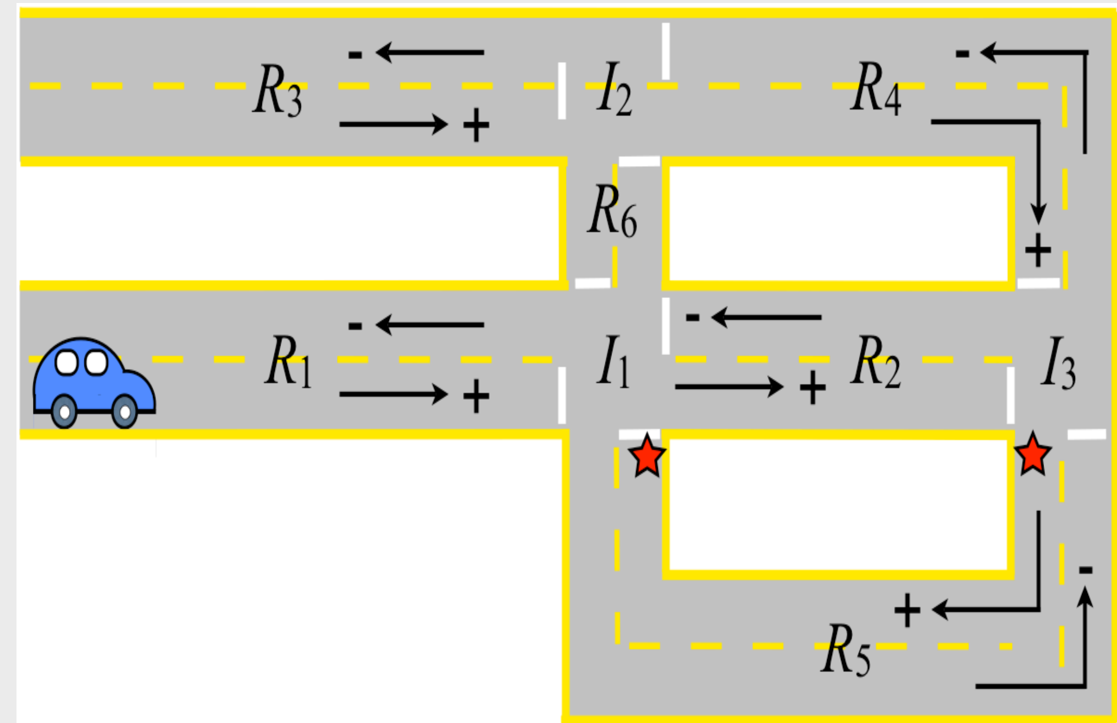


# Example: Navigation In Urban-Like Environment

Dynamics: $\dot{x}(t) = u_x(t) + d_x(t), \;\; \dot{y}(t) = u_y(t) + d_y(t)$

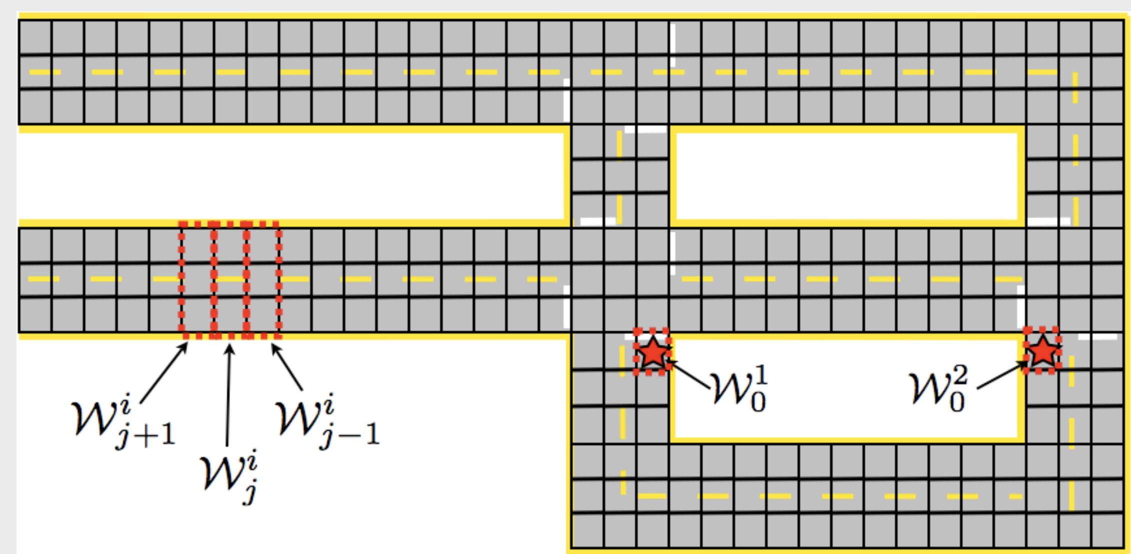
Actuation limits: $u_x(t), u_y(t) \in [-1, 1], \;\; \forall t \geq 0$

Disturbances: $d_x(t), d_y(t) \in [-.1, .1], \;\; \forall t \geq 0$




Traffic rules:

- No collision
- Stay in right lane unless blocked by obstacle
- Proceed through intersection only when clear




Environment assumptions:

- Obstacle may not block a road
- Obstacle is detected before it gets too close
- Limited sensing range (2 cells ahead)
- Obstacle does not disappear when the vehicle is in its vicinity
- Obstacles don't span more than certain # of consecutive cells in the middle of the road
- Each intersection is clear infinitely often
- Cells marked by star and adjacent cells are not occupied by obstacle infinitely often

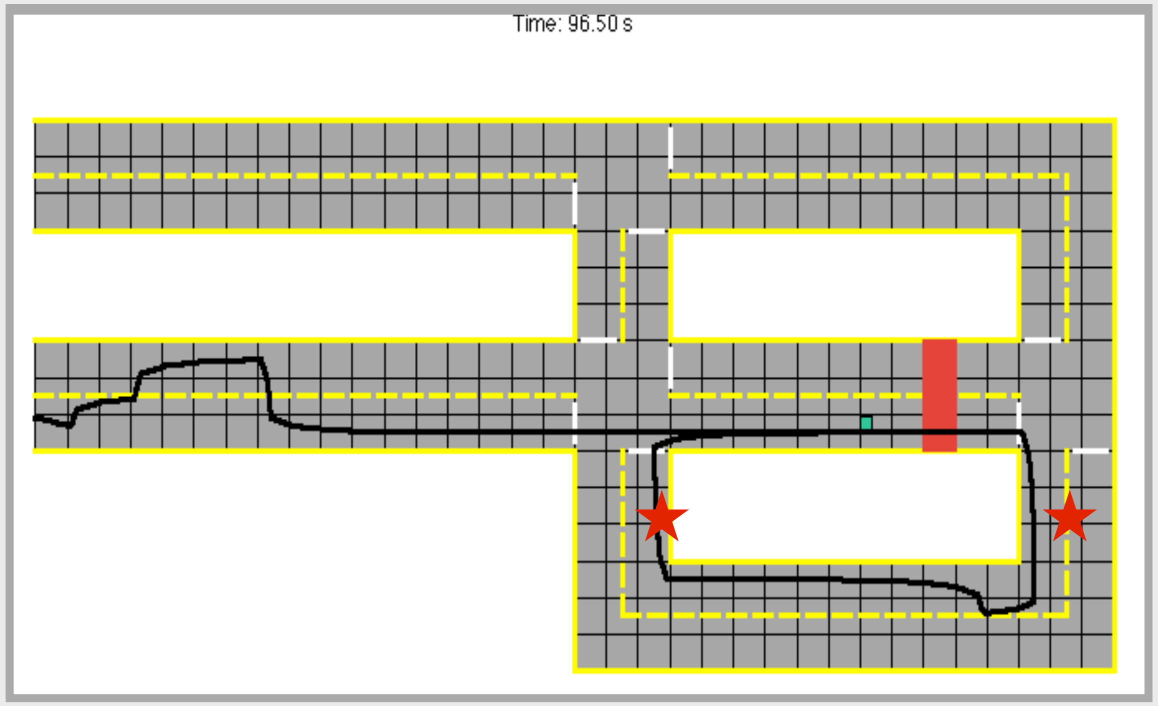
Goals: Visit the cells with *'s infinitely often.

# Navigation In Urban-Like Environment

[TAC'11(submit), HSCC'10]

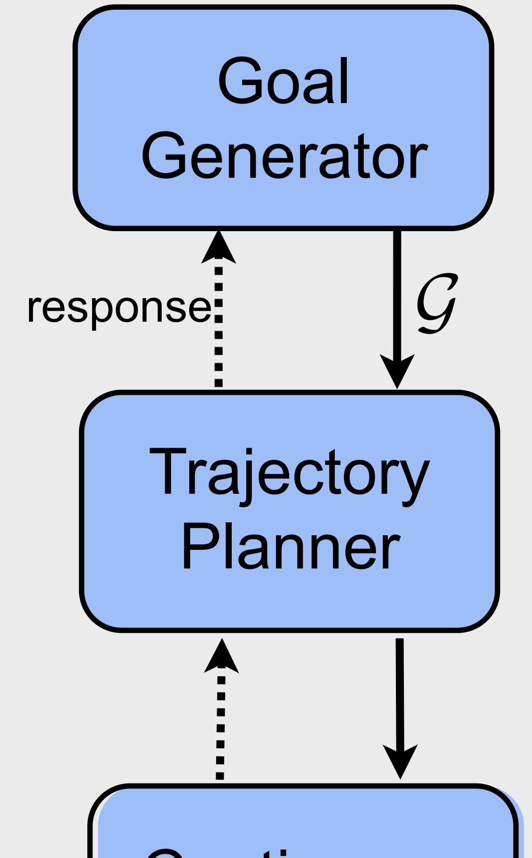
**Setup:**

- Dynamics: Fully actuated with actuation limits and bounded disturbances
- Specifications:
  - Traffic rules
  - Assumptions on obstacles, sensing range, intersections,...
- Goals: Visit the two stars infinitely often




**Results:**

- Without receding horizon: 1e87 states (hence, not solvable)
- Receding horizon:
  - Partial order: From the top layer of the control hierarchy
  - Horizon length = 2 $(\mathcal{F}(\mathcal{W}_j^i) = \mathcal{W}_{j-2}^i.)$
  - Invariant: Not surrounded by obstacles. If started in left lane, obstacle in right lane.
  - 1e4 states in the automaton.
  - ~1.5 sec for each short-horizon problem
  - Milliseconds for partial order generation



# What is $\Phi$?

- A propositional formula (that we call receding horizon invariant).
- Used to exclude the initial states that render synthesis infeasible, e.g., states from which collision is unavoidable

Short-horizon specification:

$$((\nu \in \mathcal{W}_i) \wedge \Phi \wedge \varphi_{\text{env}}) \rightarrow (\Box \Phi \wedge \varphi_{\text{safety}} \wedge \Diamond(\nu \in \mathcal{F}_i(\mathcal{W}_i)))$$

Given partial order and $\mathcal{F}$, computation of the invariant can be automated:

- Check realizability
- If realizable, done.
- If not, collect violating initiation conditions. Negate them and put in $\Phi$.
- Repeat until all subproblems or all possible states are excluded (in the latter case, either the global problem is infeasible or RHTLP with given partial order and $\mathcal{F}$ is inconclusive.)

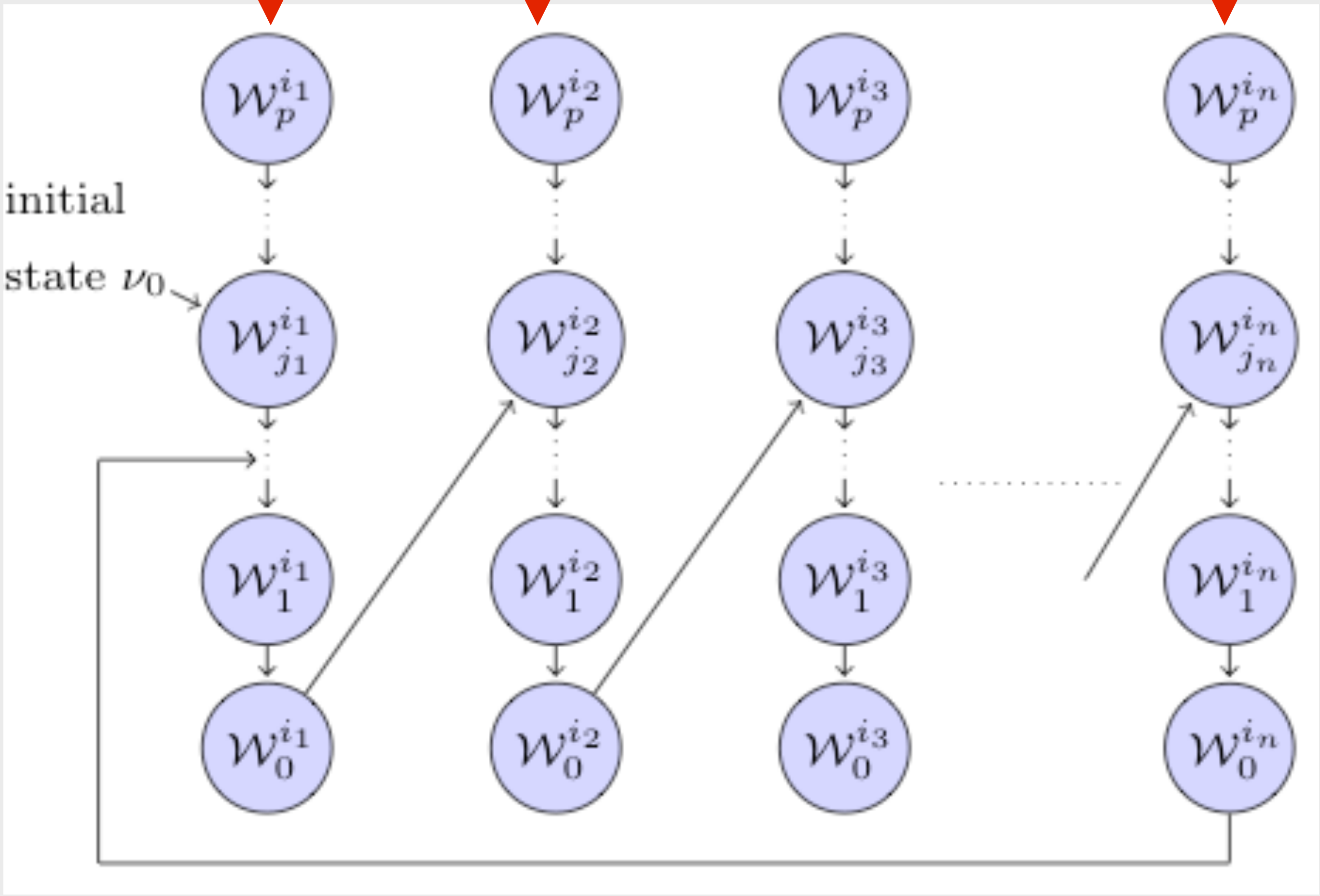
# Generalization to multiple "goals"

General form of LTL specifications considered in reactive control protocol synthesis:

multiple "goals"

$$\left(\psi_{init} \wedge \Box\psi_e \wedge \left(\bigwedge_{i\in I_f} \Box\Diamond\psi_{f,i}\right)\right) \rightarrow \left(\left(\bigwedge_{i\in I_s} \Box\psi_{s,i}\right) \wedge \left(\bigwedge_{i\in I_g} \Box\Diamond\psi_{g,i}\right)\right)$$

Each partial order covers the discrete (system) state space. For each $\nu \in \mathcal{W}_0^{i_j}$, one can find a cell in the "proceeding" partial order that $\nu$ belongs to.




<u>Strategy</u>: While in $\mathcal{W}_j^i$ implement (in a receding horizon fashion) the controller that realizes

$$\left((\nu \in \mathcal{W}_j^i) \wedge \Phi \wedge \Box\psi_e^e \wedge \bigwedge_{k\in I_f} \Box\Diamond\psi_{f,k}^e\right)$$
$$\Longrightarrow \left(\bigwedge_{k\in I_s} \Box\psi_{s,k} \wedge \Box\Diamond\left(\nu \in \mathcal{F}^i(\mathcal{W}_j^i)\right) \wedge \Box\Phi\right)$$

# Computational complexity & completeness

For Generalized Reactivity [1] formulas, the computation time of synthesis is $O(mn|\Sigma|^3)$, where $|\Sigma|$ is the number of discrete states.

$$\bigwedge_{i=1}^{m} \Box \Diamond p_i^e \rightarrow \bigwedge_{j=1}^{n} \Box \Diamond q_j^s$$

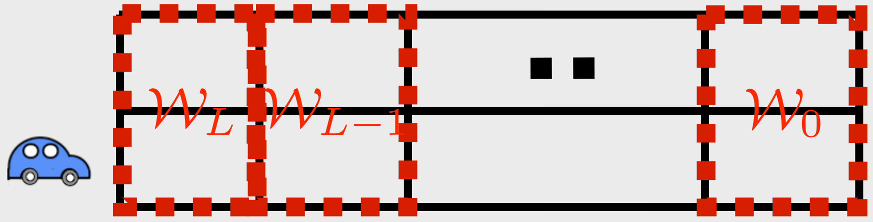
Receding horizon implementation...

- reduces the computational complexity by restricting the state space considered in each subproblem; and
- is not complete, i.e., the global problem may be solvable but the choice of $\{\mathcal{W}_j\}$, the partial order, the maps $\mathcal{F}_i$, and $\Phi$ may not lead to a solution.

Choose $\mathcal{F}_i$ to give "longer horizon":

- Subproblems in RHTLP are more likely to be realizable.
- Computational cost is higher.

E.g., for urban-like driving example is infeasible with horizon length of one.




Global synthesis problem

$$(\varphi_{init} \wedge \varphi_{env}) \rightarrow (\varphi_{safety} \wedge \varphi_{goal})$$

Subproblems in RHTLP

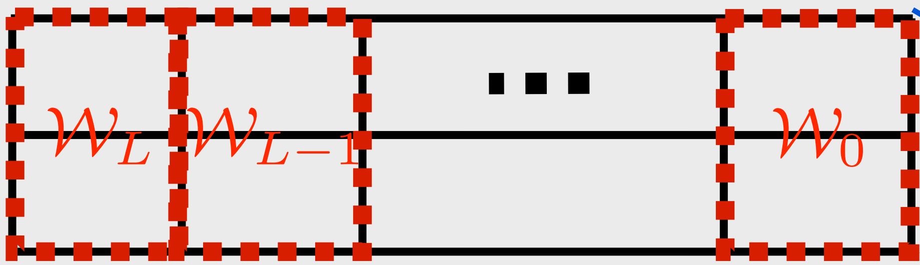
$$((v \in \mathcal{W}_i) \wedge \Phi \wedge \varphi_{end}) \rightarrow (\varphi_{safety} \wedge \Diamond(v \in \mathcal{F}_i(W_i) \wedge \Box\Phi)$$

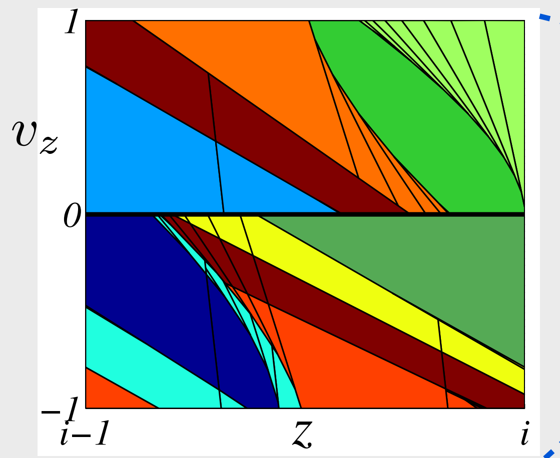
# Hierarchical control structure

Abstraction procedure and bisimulations relate models of different fidelity level.

(How: see the coming slides.)

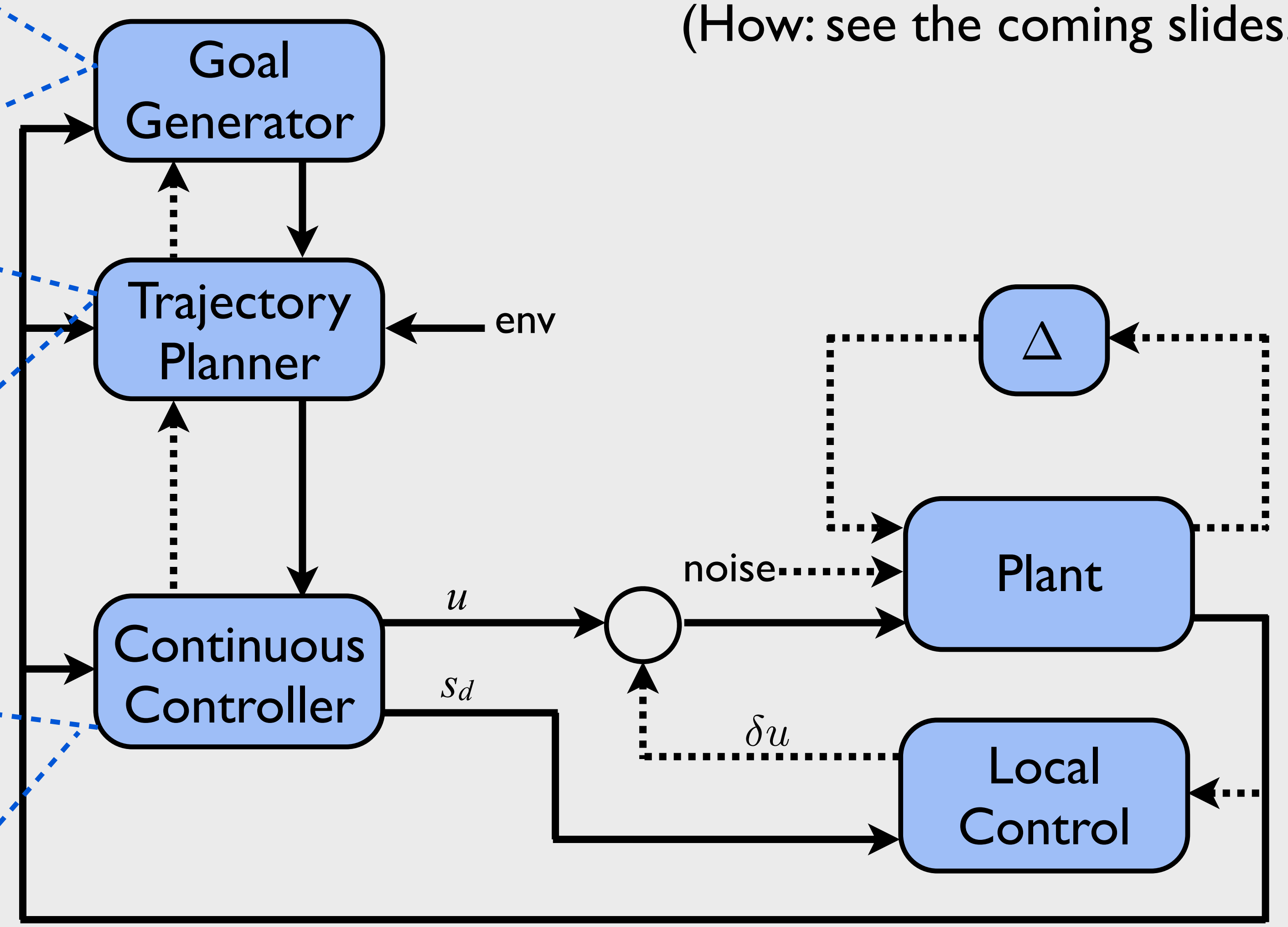
<u>models of varying fidelity</u>



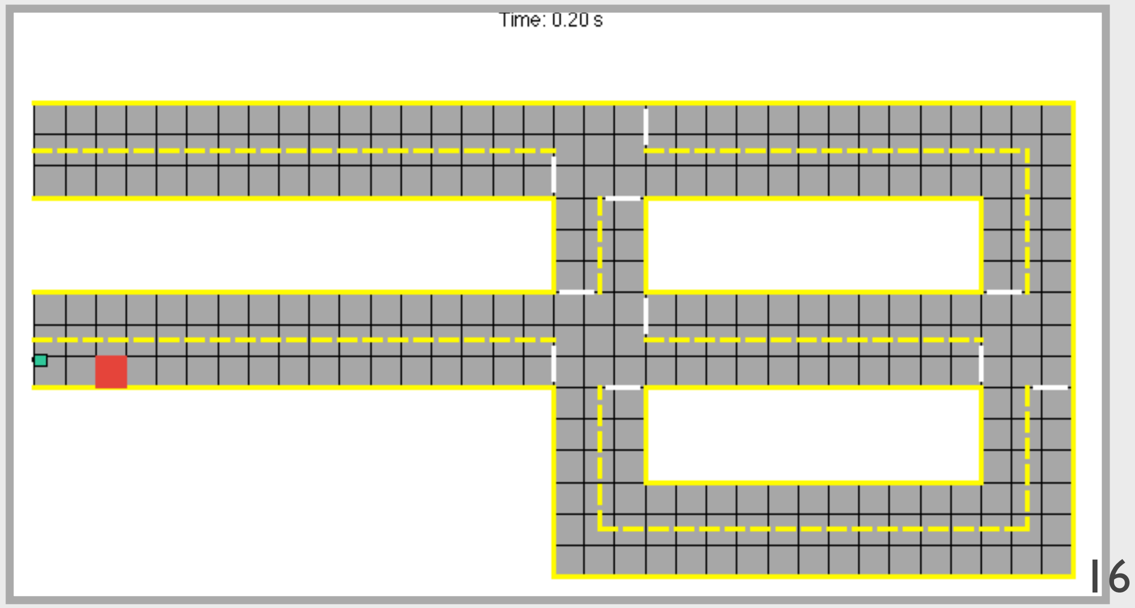




$$\ddot{x} + \dot{x} = q_x(t)$$
$$\ddot{y} + \dot{y} = q_y(t)$$
$$\ddot{\theta} + \frac{2mL^2}{J}\dot{\theta} = q_\theta$$

$$|q_x(t)|, |q_y(t)| \leq \sqrt{0.5}$$
$$|q_\theta(t)| \leq 1$$

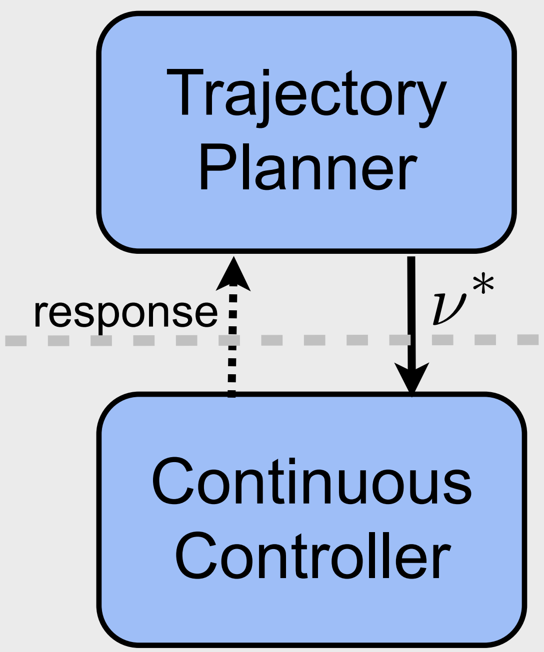


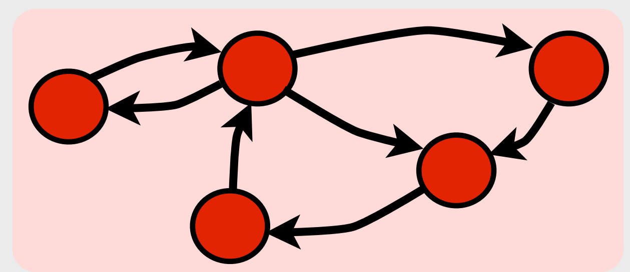

Response mechanism is introduced to compensate for mismatch between the system and its model and between the actual behavior of the environment and its assumptions.

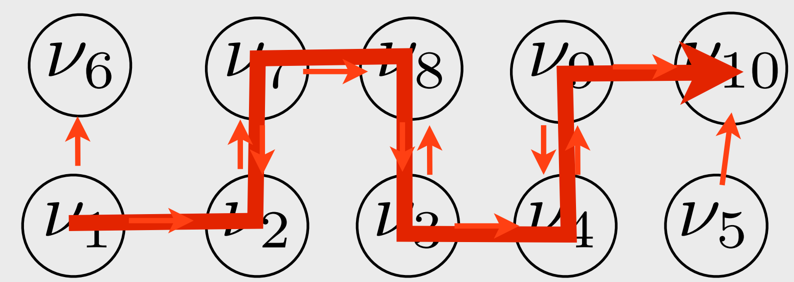


# Incorporating continuous dynamics: main idea

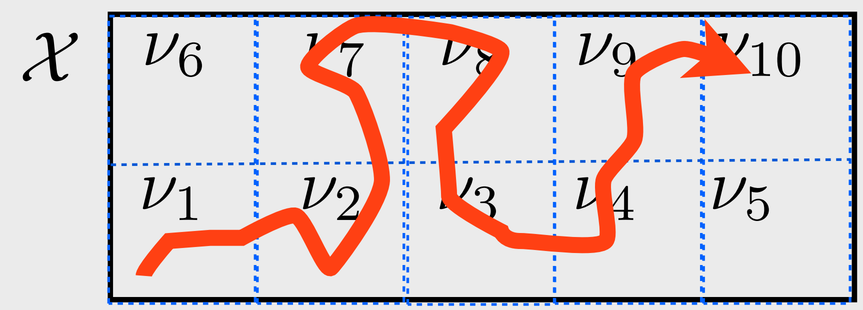
**Main idea:**






$$\dot{\xi} = f(\xi, w, u)$$
$$\xi \in \mathcal{X},\ u \in \mathcal{U}, w \in \mathcal{W}$$




**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 $dom(V) = dom(S) \times dom(E)$.

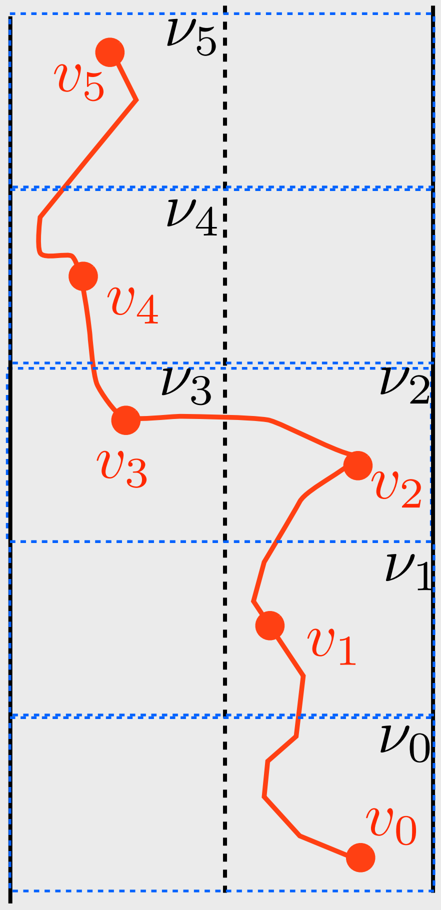
•Controlled variables evolve with (for $t = 0,1,2,...$):

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

•System specification $\varphi$

**Find:** A finite transition system with discrete states $\nu$ such that for any sequence $\nu_0 \nu_1 \ldots$ satisfying $\varphi$, (very roughly speaking) there exists a sequence of admissible control signals leading to an infinite sequence $v_0 v_1 v_2 \ldots$ that satisfies $\varphi$.

(stated more precisely later...)

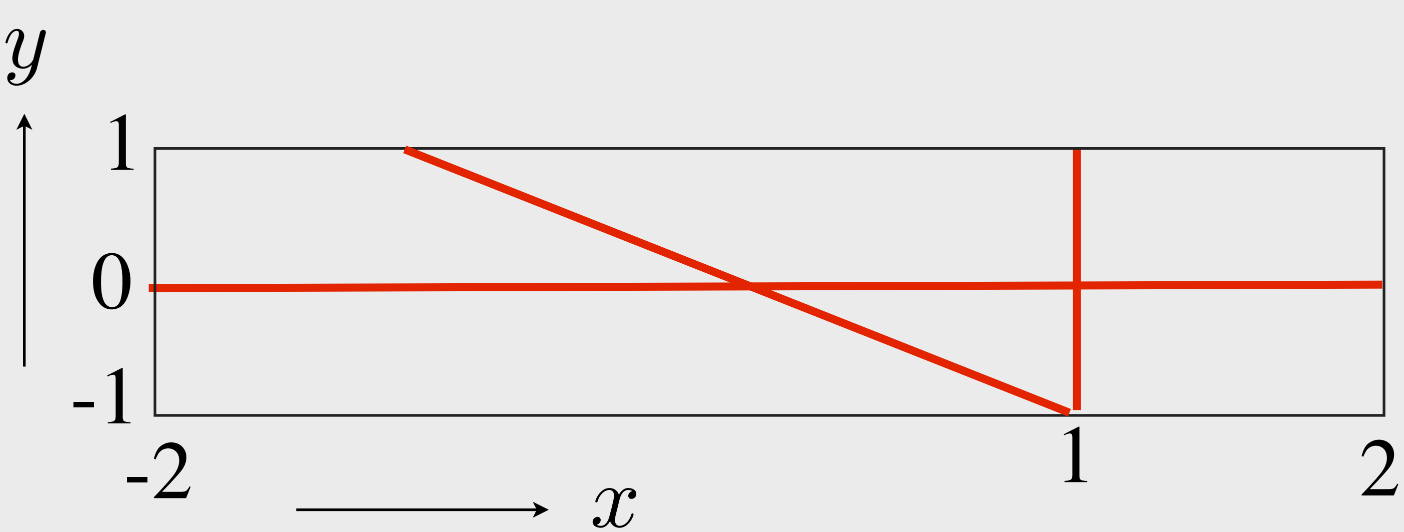


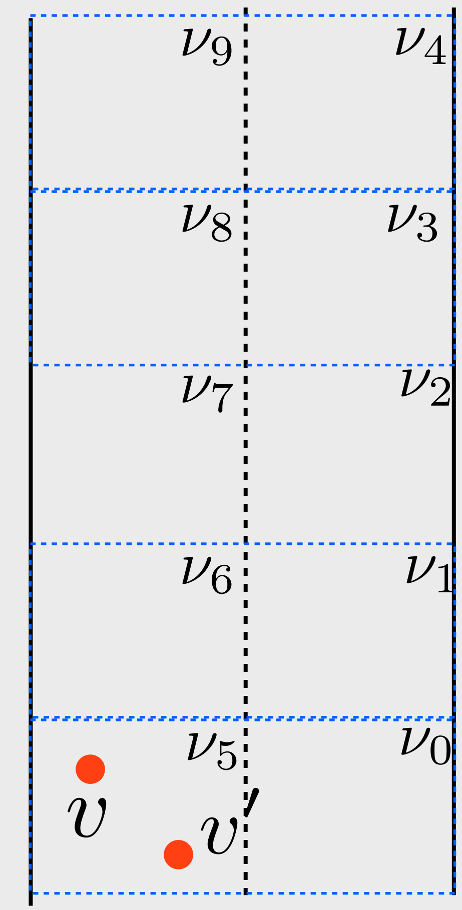
# Proposition preserving partition

Given $dom(V)$ and atomic propositions in $\Pi$.

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 $\pi$ if and only if $v'$ satisfies $\pi$.

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:

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

$$\Downarrow$$

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

A discrete state $\nu$ is said to satisfy $\pi$ if and only if there exists a continuous state $v$, in the cell labeled, that satisfies $\pi$.

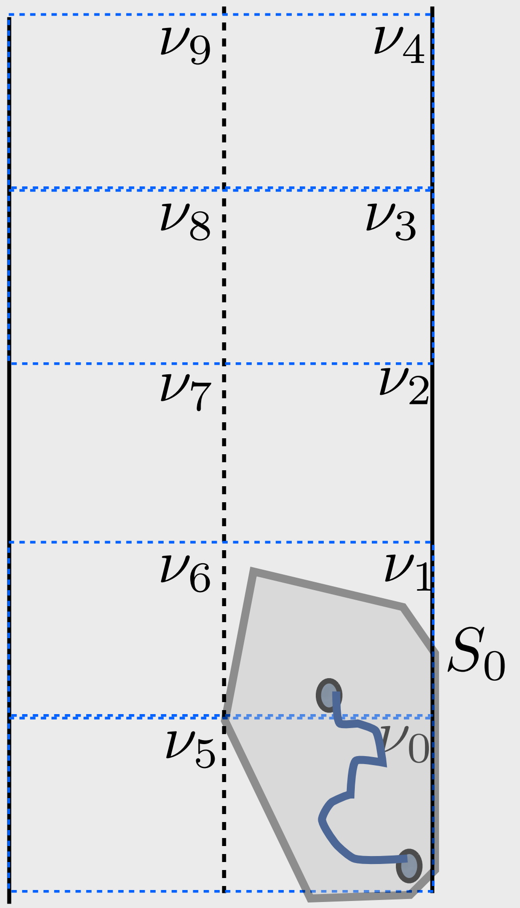
# Finite-time reachability

A discrete state $\nu_j$ is finite-time reachable from a discrete state $\nu_i$, only if starting from <u>any</u> $s[0] \in T_s^{-1}(\nu_i)$, there exists

- a finite horizon length $N \in \{0, 1, \dots\}$
- for <u>any</u> allowable disturbance, there exists

$u[0], u[1], \dots, u[N-1] \in U$ such that

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



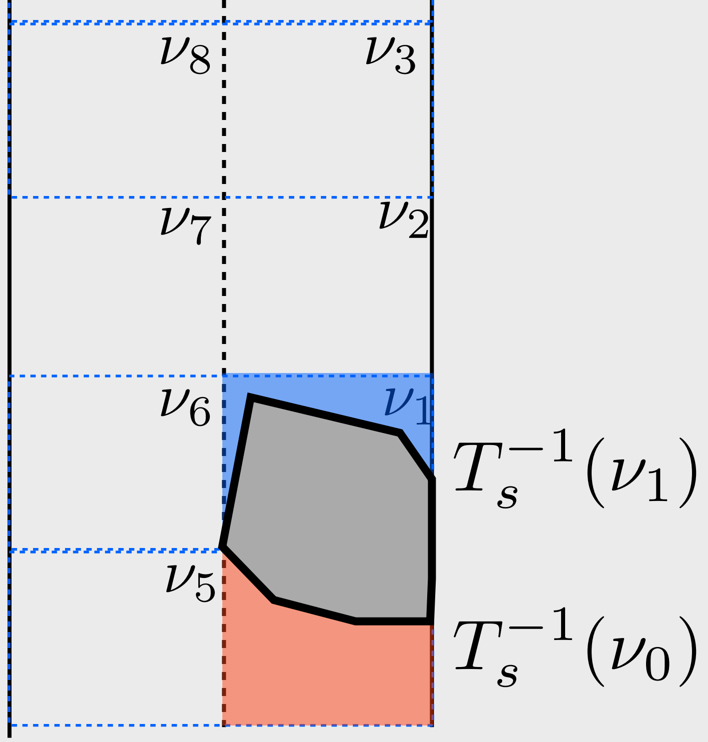

Verifying the reachability relation:

- 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

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

$$L_3 s[N] \leq M_3,$$

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).

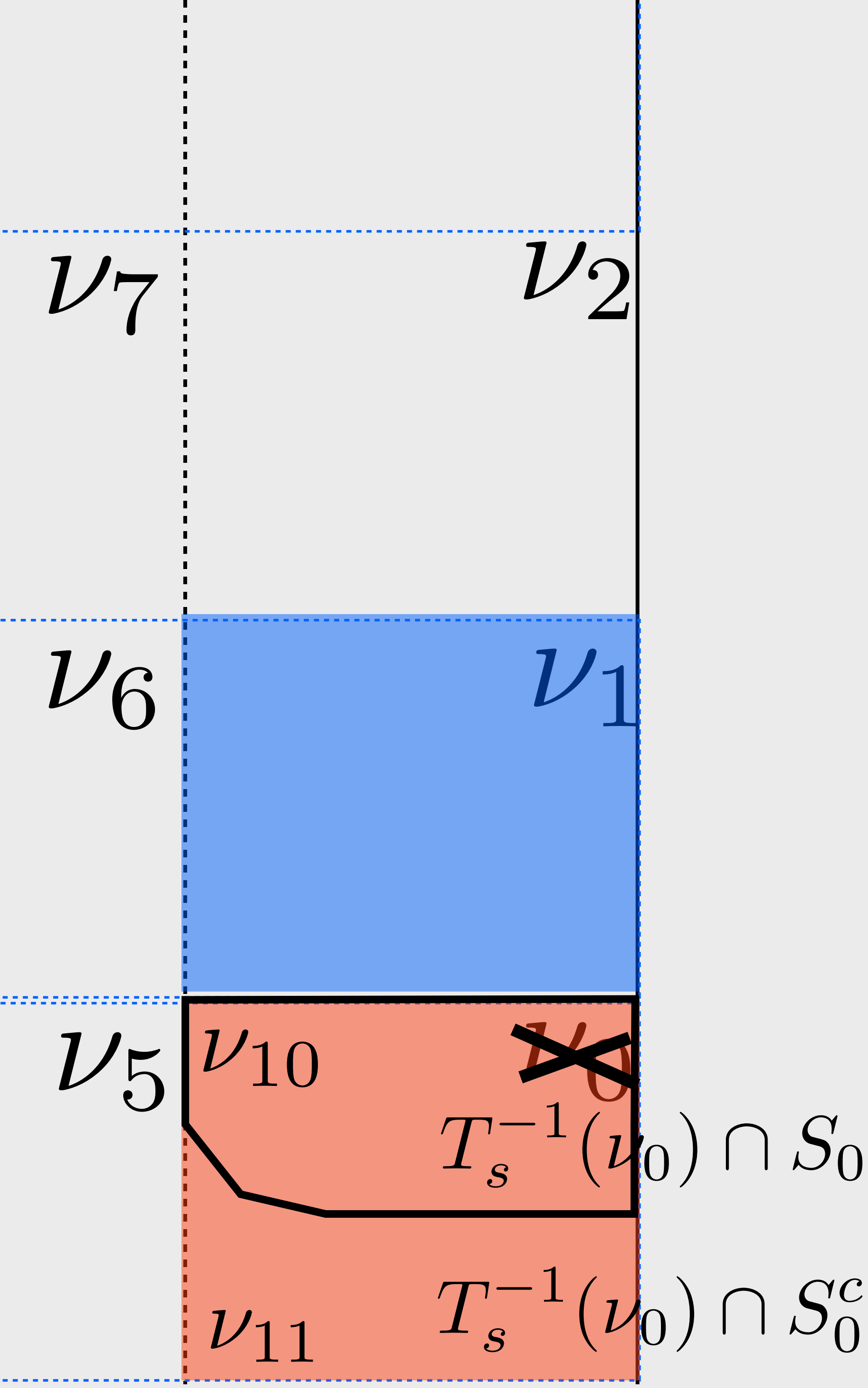
**Put together**: $S_0$ is computed as a polytope projection:

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

stacking of $u$ and $d$

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

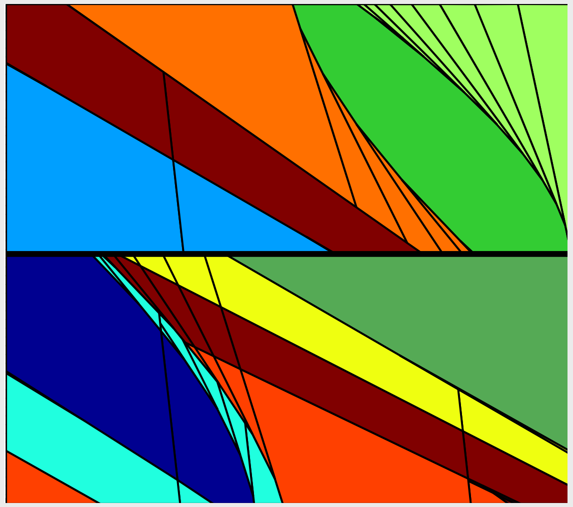
# Refining the partition



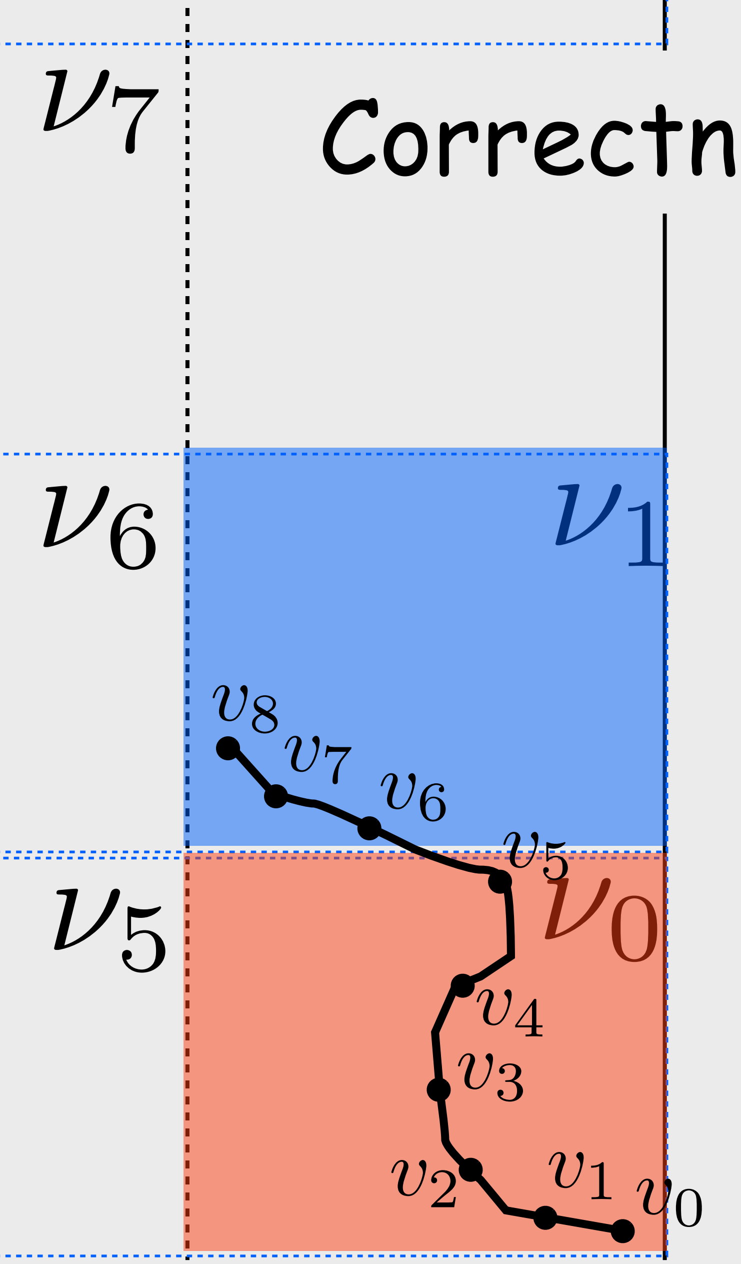

While checking the reachability from $T_s^{-1}(\nu_i)$ to $T_s^{-1}(\nu_j)$, if $T_s^{-1}(\nu_i) \nsubseteq S_0$, then
- Split $T_s^{-1}(\nu_i) \cap S_0$ and $T_s^{-1}(\nu_i) \cap S_0^c$
- Remove $\nu_i$ from the set of discrete states
- Add two new discrete states corresponding to $T_s^{-1}(\nu_i) \cap S_0$ 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 $\varsigma_j$ is reachable from $\varsigma_i$.



# 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 $\varphi$, ...

Two words $\sigma_1$ and $\sigma_2$ 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 $\sigma_1$ and $\sigma_2$ 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 $P$.

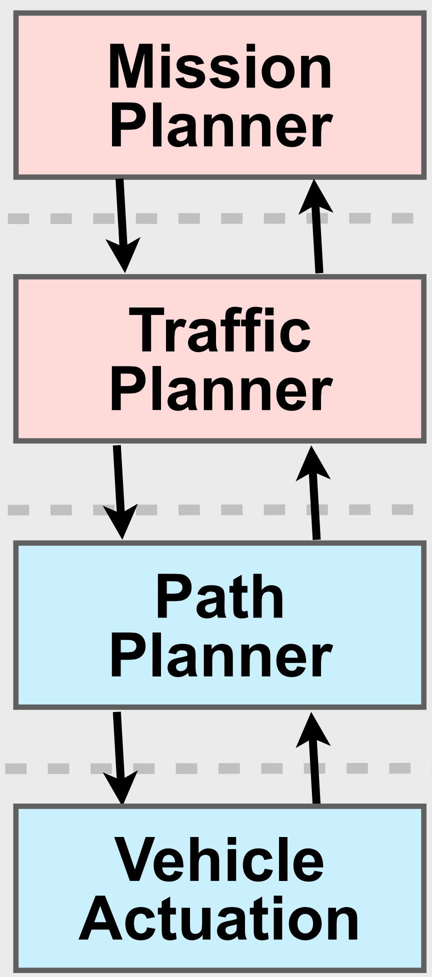
Example: $v_0 v_1 \ldots v_8 \ldots$ and $\nu_0 \nu_1 \ldots$ are stutter-equivalent.

**...we can prove:**

Let $\sigma_d = \nu_0 \nu_1 \ldots$ be a sequence in $\mathbb{D}$ with $\nu_k \to \nu_{k+1}$, $\nu_k = (\varsigma_k, \epsilon_k)$, $\varsigma_k \in \mathcal{S}$ 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 $\varphi$.

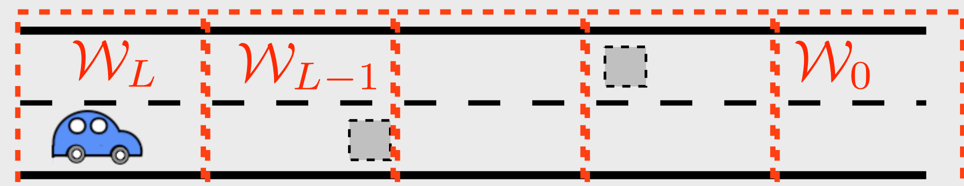
# Summary

| Alice's navigation stack | Different views | Multi-scale models | Hierarchical control architecture |
|---|---|---|---|

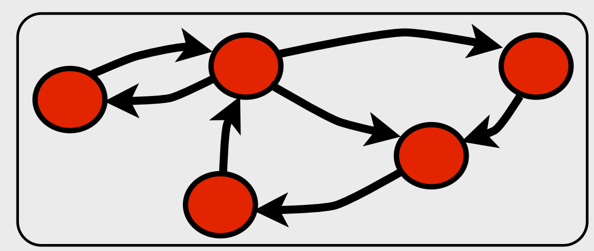



"long-horizon specification"

$\mathcal{W}_0 \prec \ldots \prec \mathcal{W}_{L-1} \prec \mathcal{W}_L$

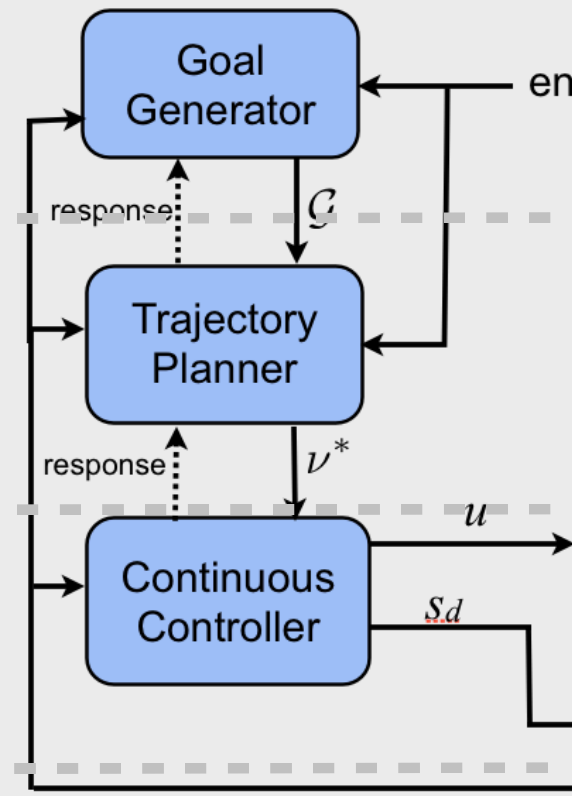



"short-horizon specification"



continuous dynamics& constraints

$$\min \int_{t_0}^{T} L(x,u)dt$$
$$\text{s.t.} \quad \dot{x} = f(x,u)$$
$$g(x,u) \leq 0$$




TuLiP: Temporal logic planning toolbox
(Open source at http://tulip-control.sf.net)

[Coming up in the next lecture]