Lecture 8
Receding Horizon Temporal Logic Planning & Compositional Protocol Synthesis

Ufuk Topcu

Nok Wongpiromsarn   Richard M. Murray

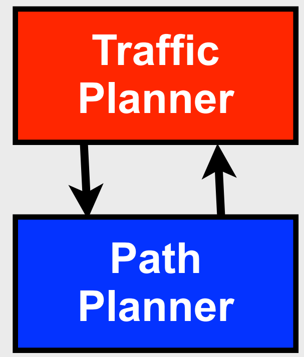
EECI, 18 May 2012

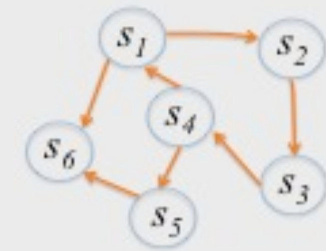
Outline:

- Receding horizon temporal logic planning (RHTLP)
  - Basic idea & main result
  - Discussion of the key details of implementation
  - Hierarchical control architecture
  - Autonomous driving examples
- Compositional control protocol synthesis and its application to smart camera networks and resource allocation

Problem: Design control protocols, that…

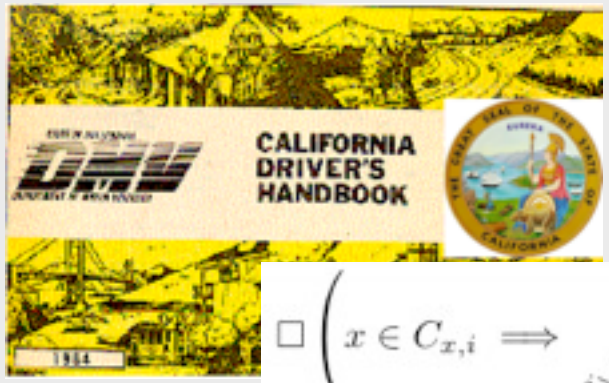
Handle mixture of discrete and continuous dynamics





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

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



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

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

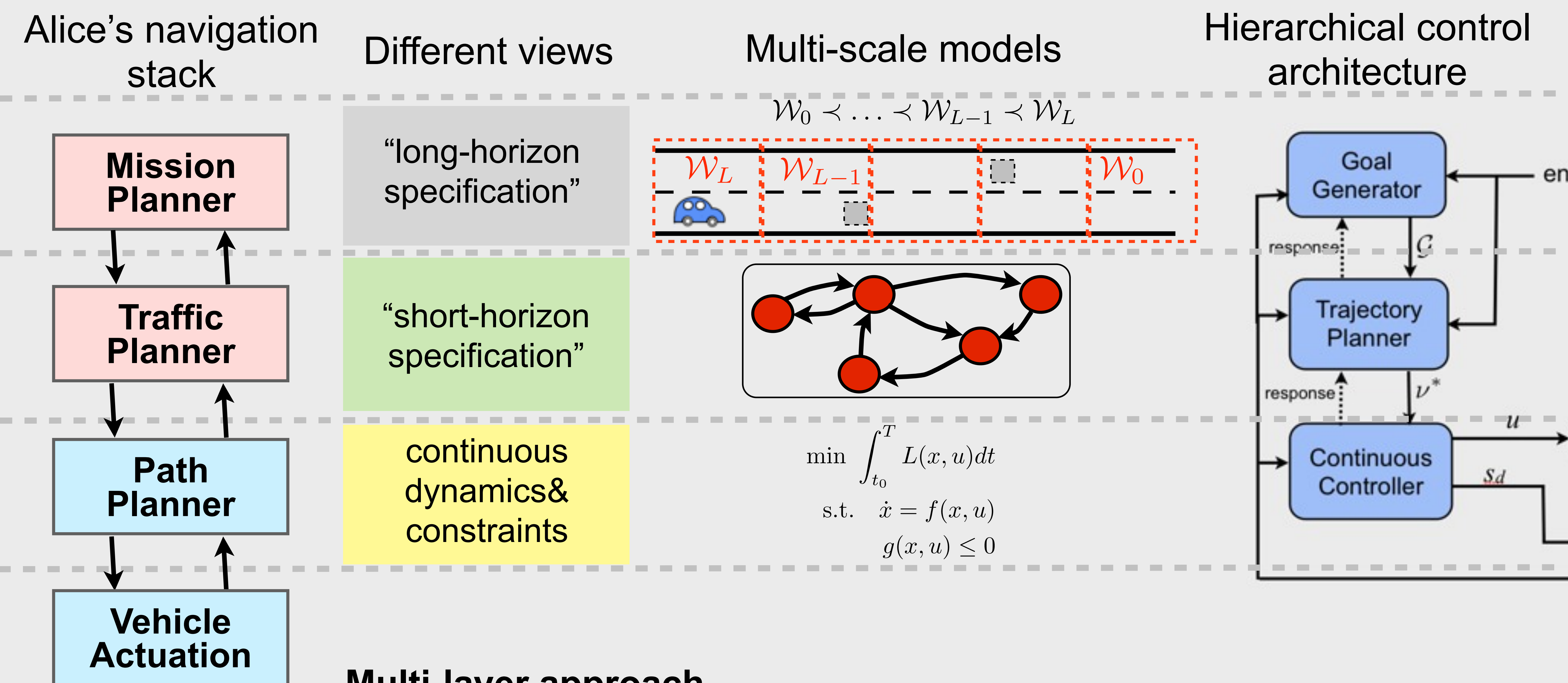
Reactively respond to changes in environment,

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

… with "correctness certificates."

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

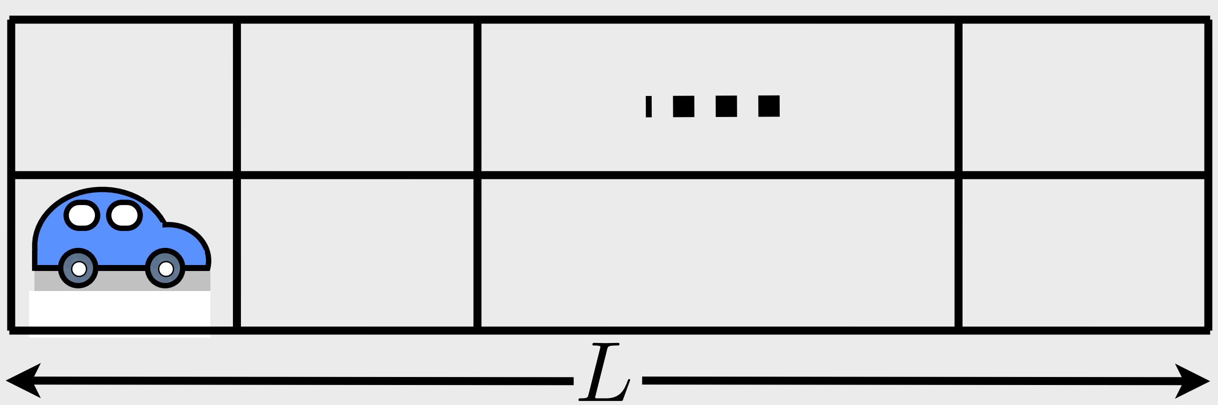
Preview



Multi-layer approach

- Use optimal trajectory generation to create a discrete abstraction that captures the dynamics at a simplified level
- Reactive planner based on GR(1) synthesis (possibly RHC)
- High level planner sends specifications to reactive planer
- Online versus offline decisions at each level

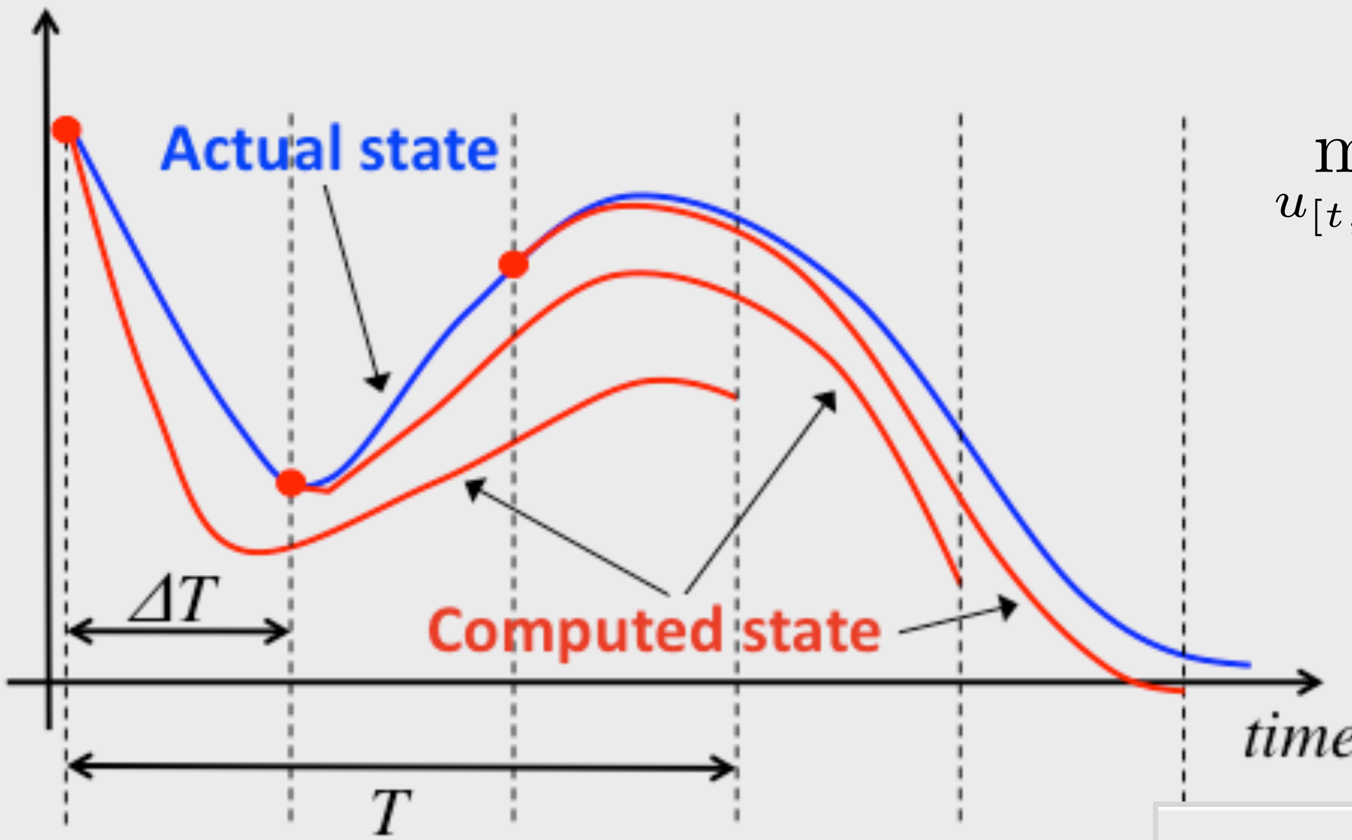
Computational Complexity



- 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

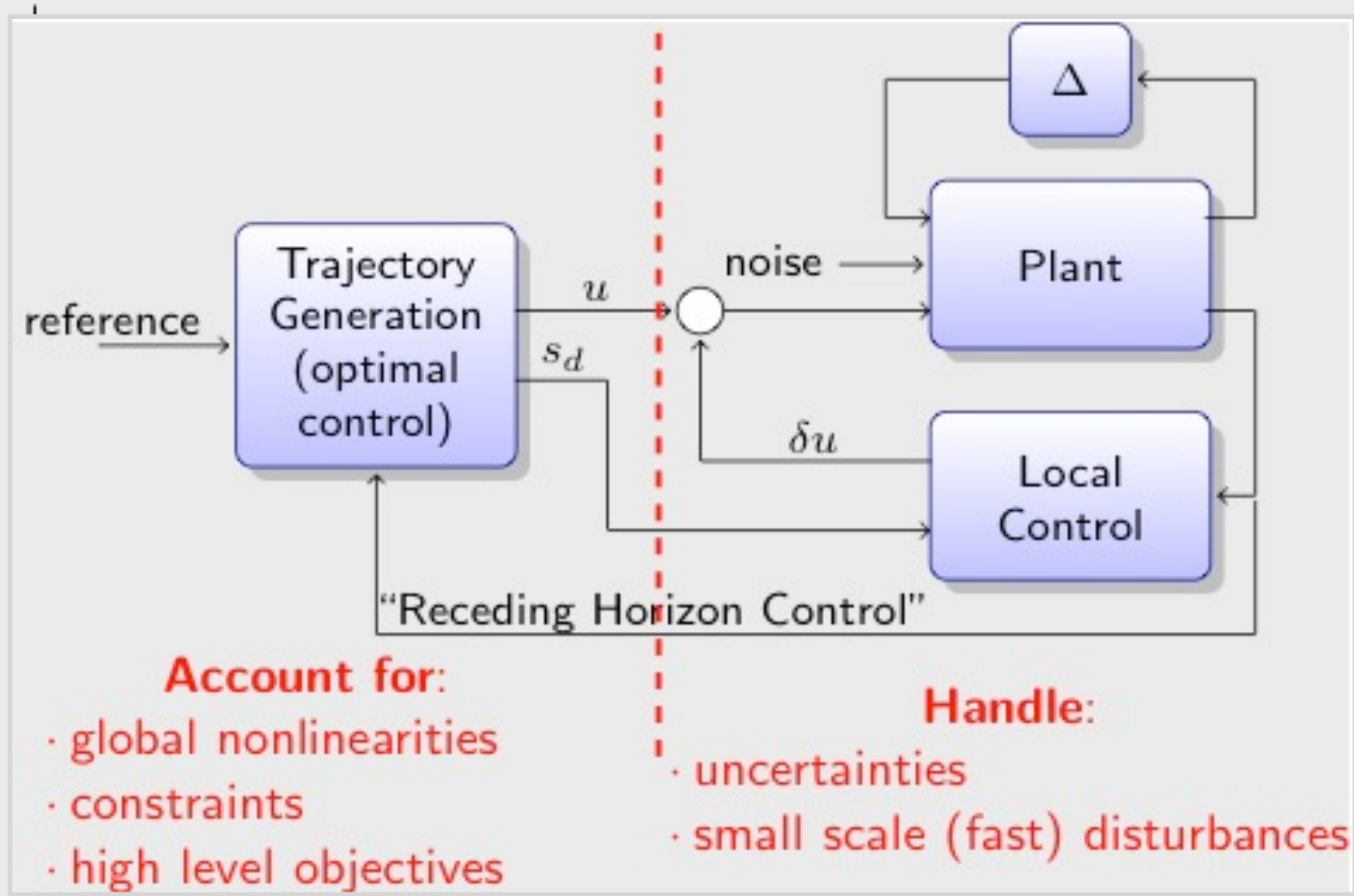


$$\min_{u_{[t,t+T]}} \int_t^{t+T} C(x(\tau), u)\tau))d\tau + V(x(t+T))$$

subject to:

$$\dot{x} = f(x,u), \quad x(t) \text{ given}$$
$$g(x,u) \le 0$$

- 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:

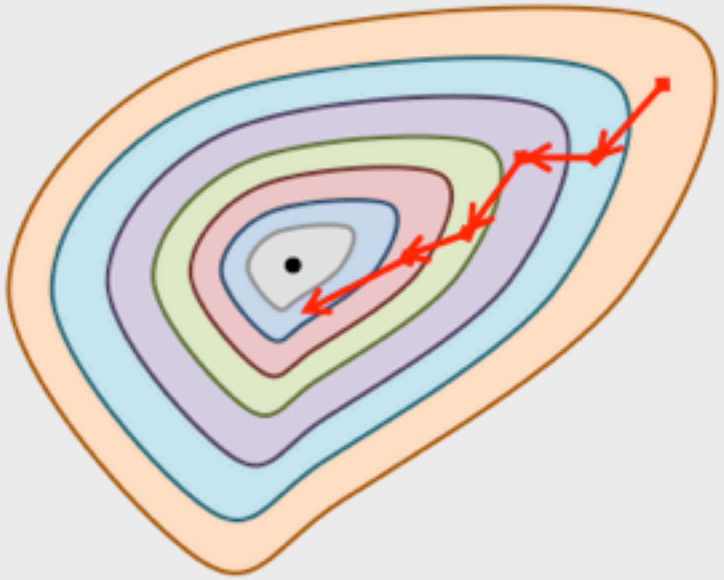
$$\dot{x} = f(x,u), \quad x(t) \text{ given}$$

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

- 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

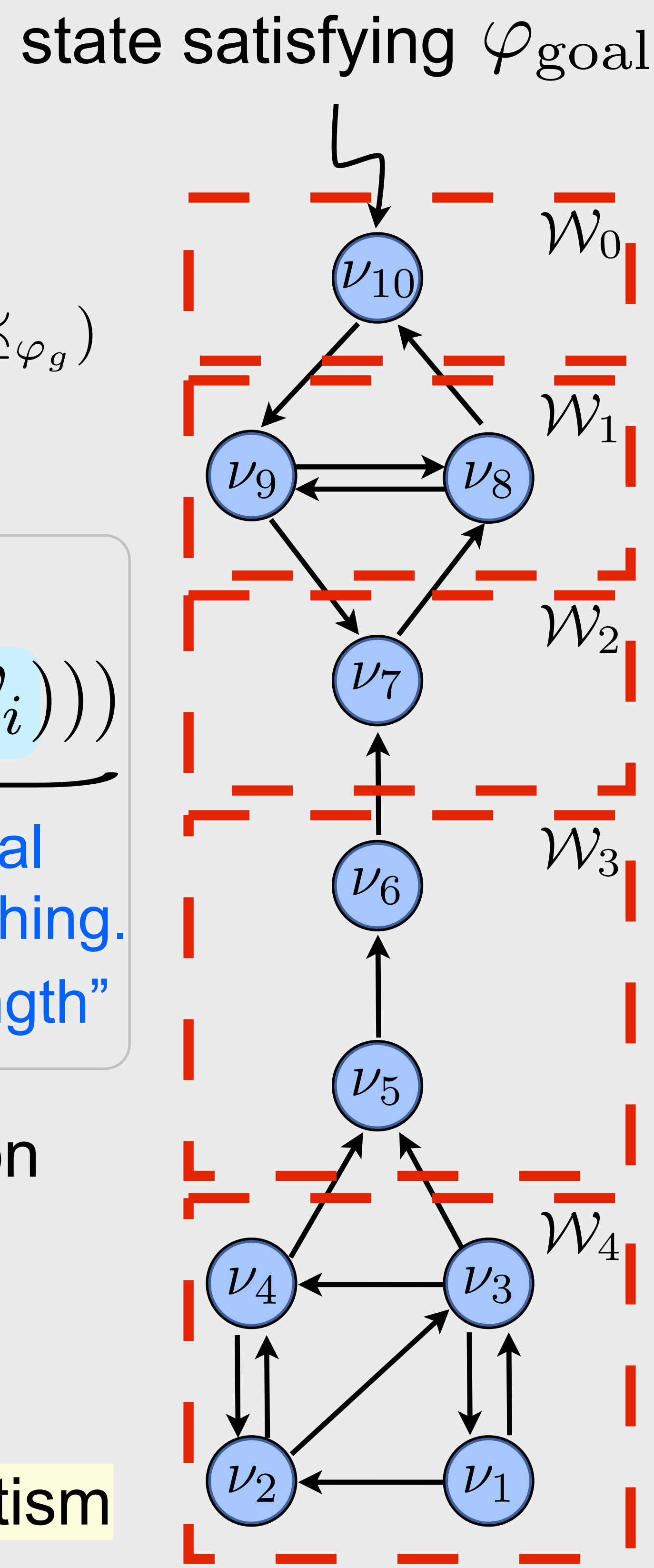
Short-horizon specification: For each i,

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

Plan from the current cell on

Receding horizon invariant: rules out "corner" cases

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



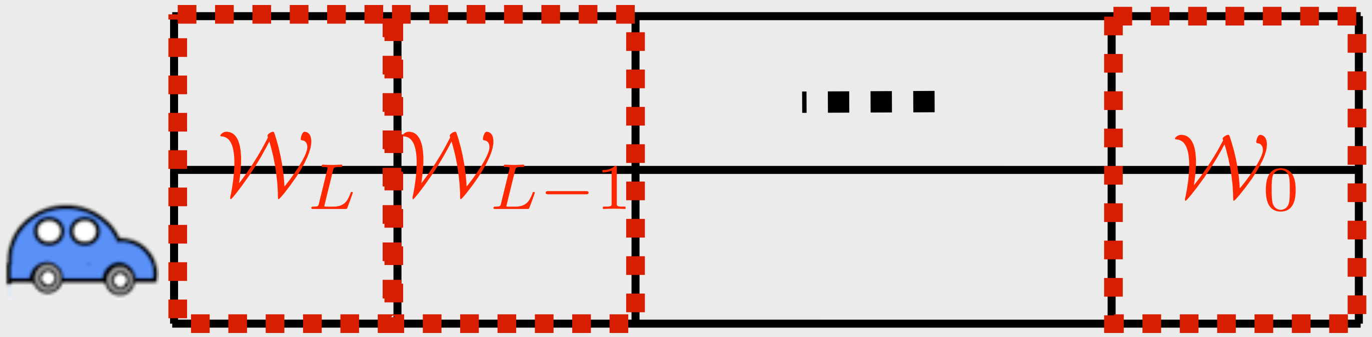
Theorem: Receding horizon implementation of the short-horizon strategies ensures the correctness of the global specification.

Trade-offs:

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

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

- 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 Φ (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:

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

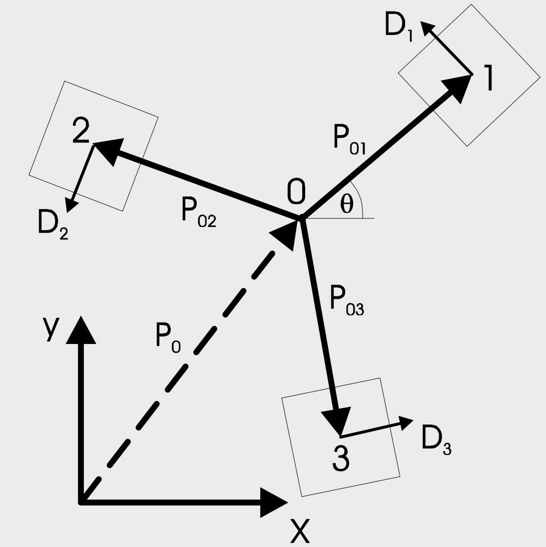
$$\ddot{y} + \dot{y} = q_y(t)$$

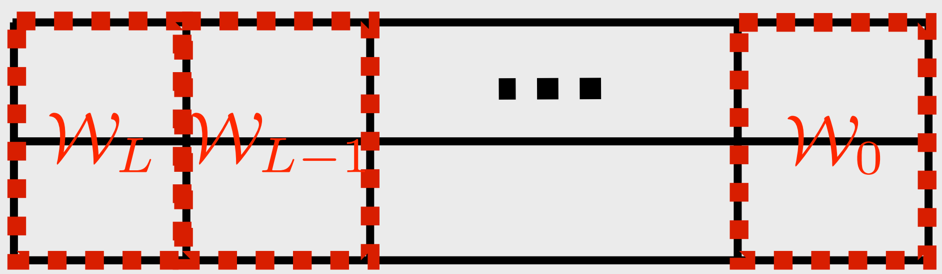
$$\ddot{\theta} + \frac{2mL^2}{J}\dot{\theta} = q_\theta$$

conservative bounds on control authority to decouple the dynamics:

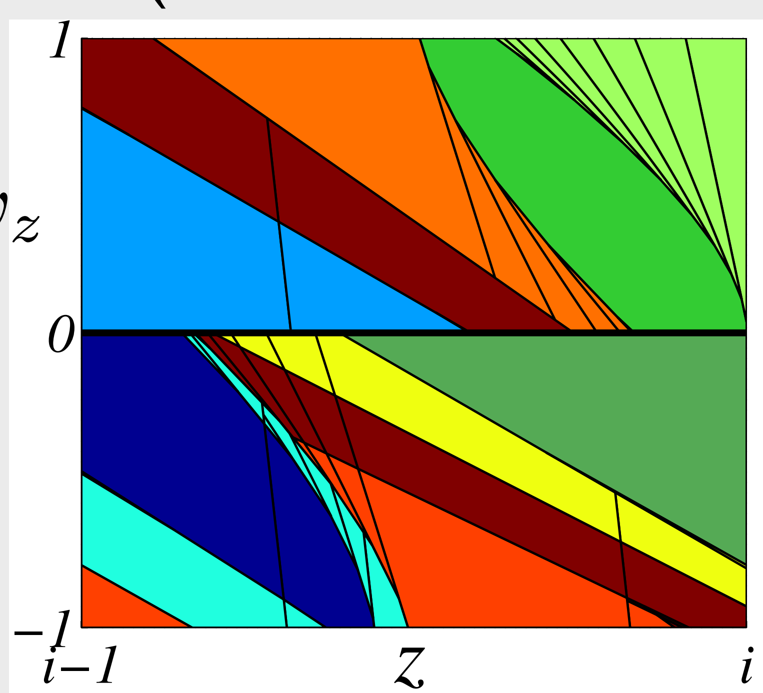
$$|q_x(t)|, |q_y(t)| \leq \sqrt{0.5}$$

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



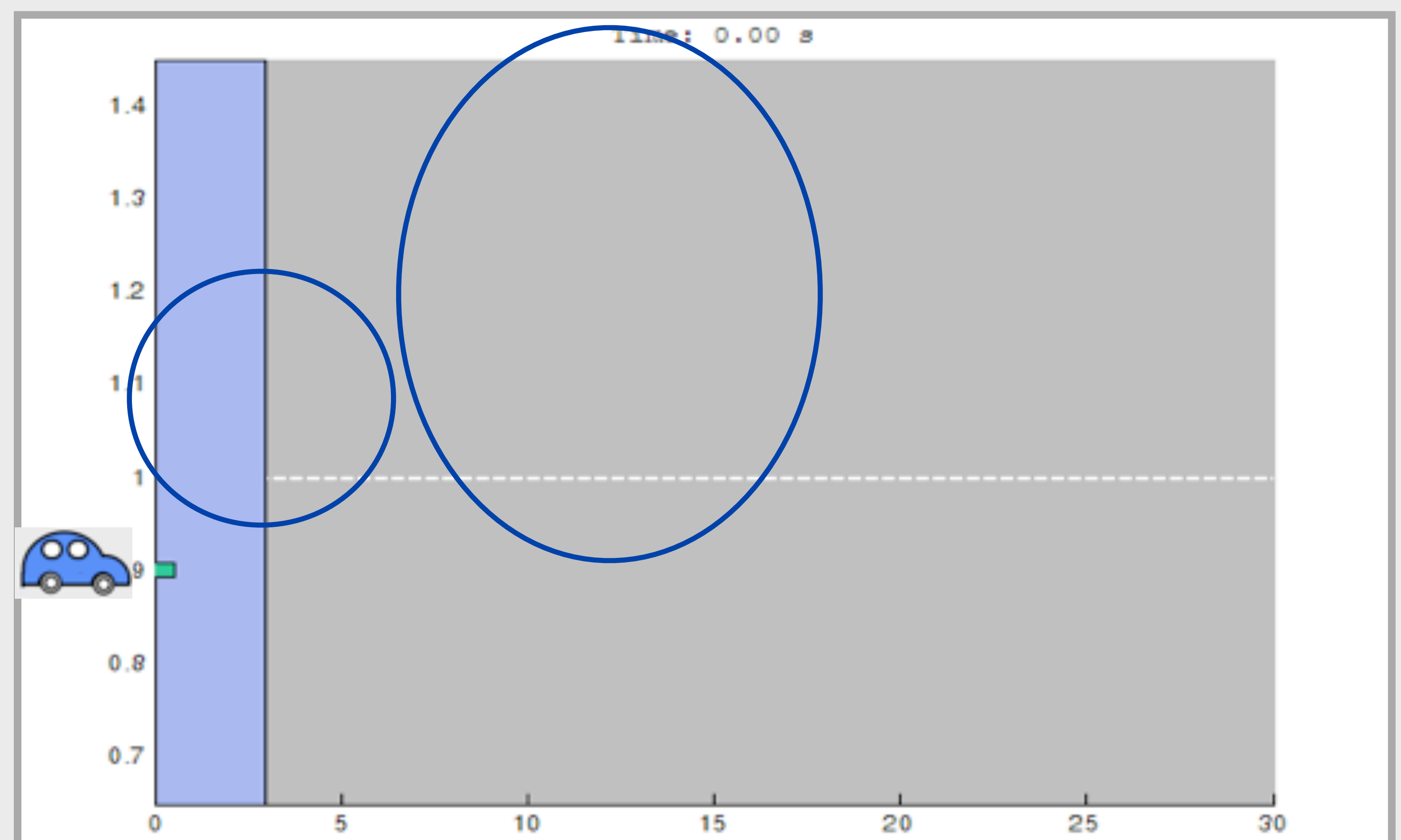


Partition (in two consecutive cells):



Reasons for the non-intuitive trajectories:

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

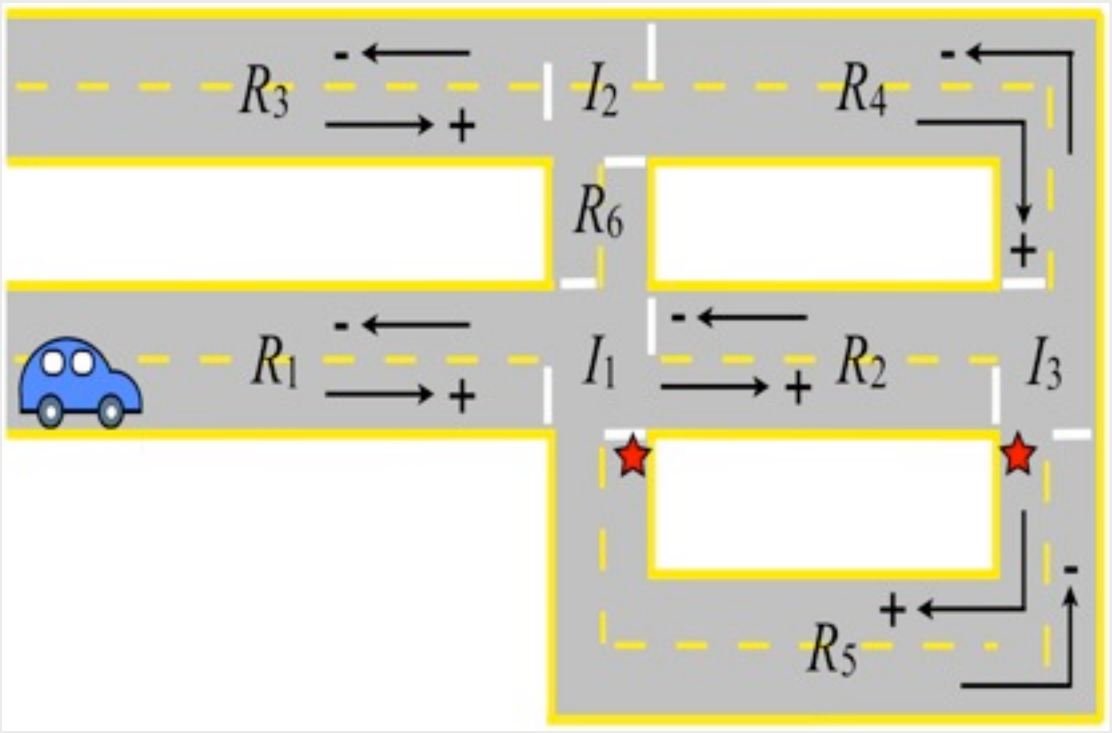


Example: Navigation In Urban-Like Environment

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

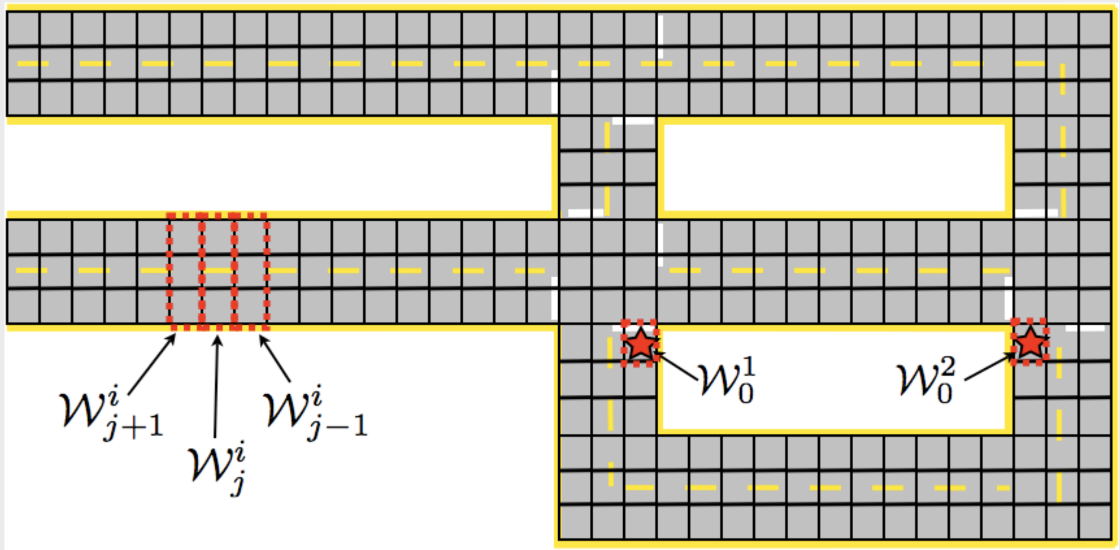


Traffic rules:

- 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



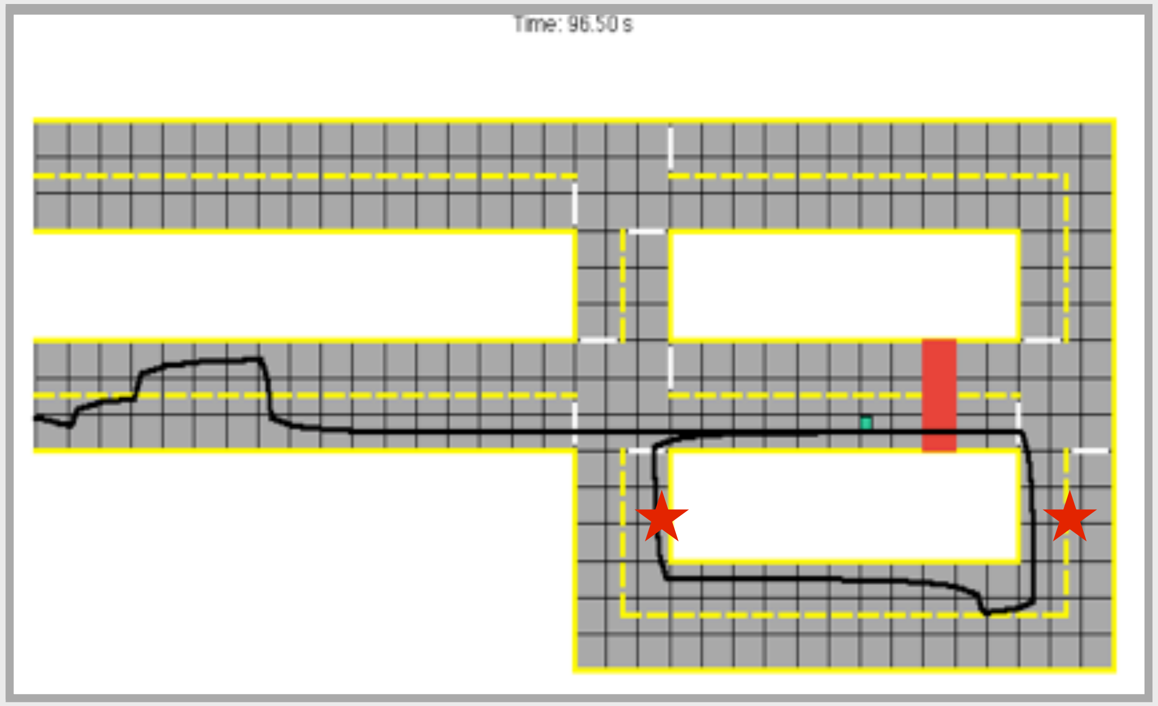
Goals: Visit the cells with *'s infinitely often.

Navigation In Urban-Like Environment

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

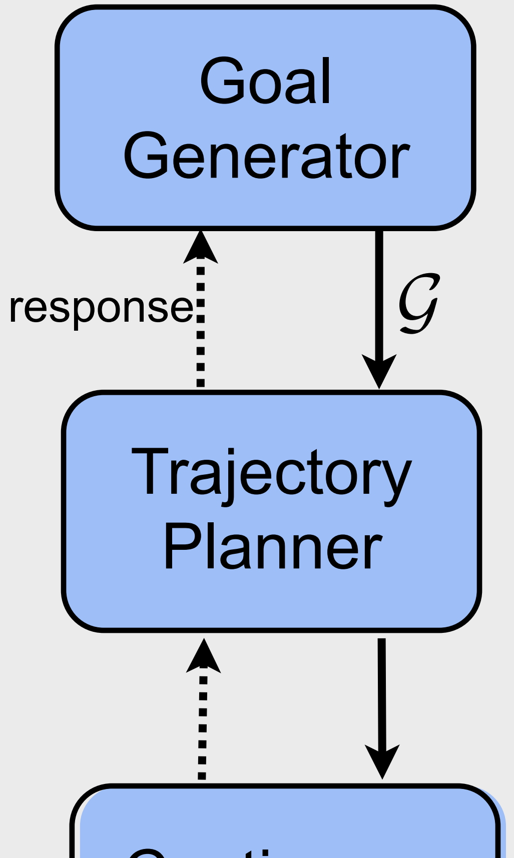
Setup:

- 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 Φ?

- 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 Φ.
- 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.)

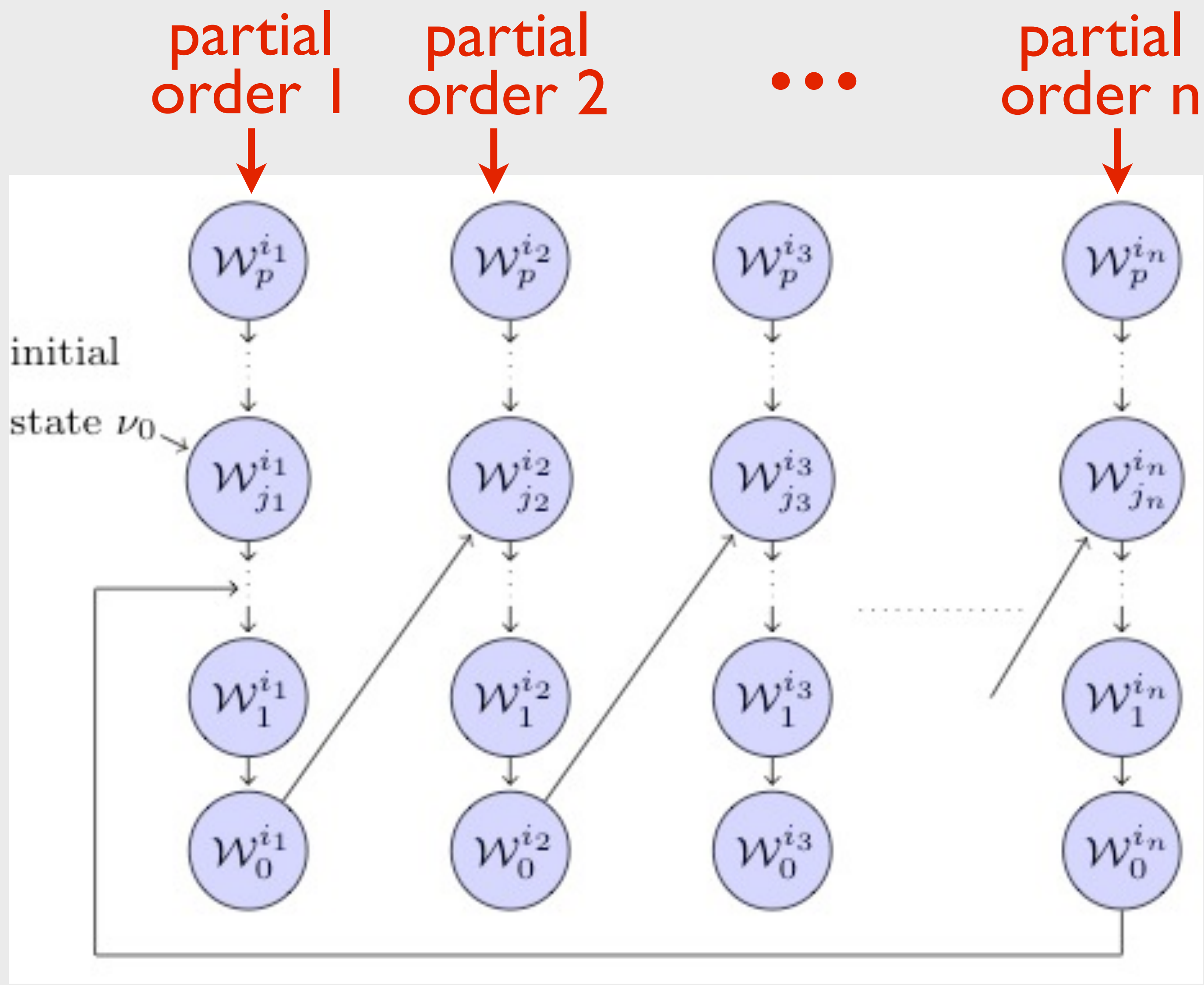
Generalization to multiple "goals"

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 ν 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.

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

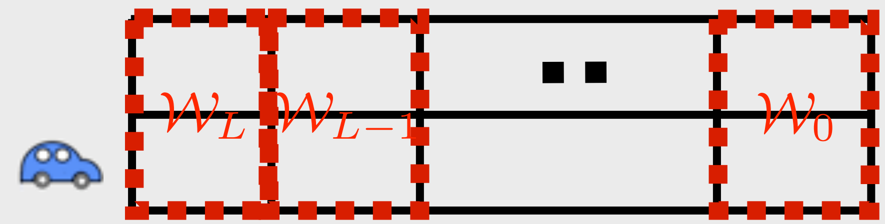
Receding horizon implementation...

- 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 Φ 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.



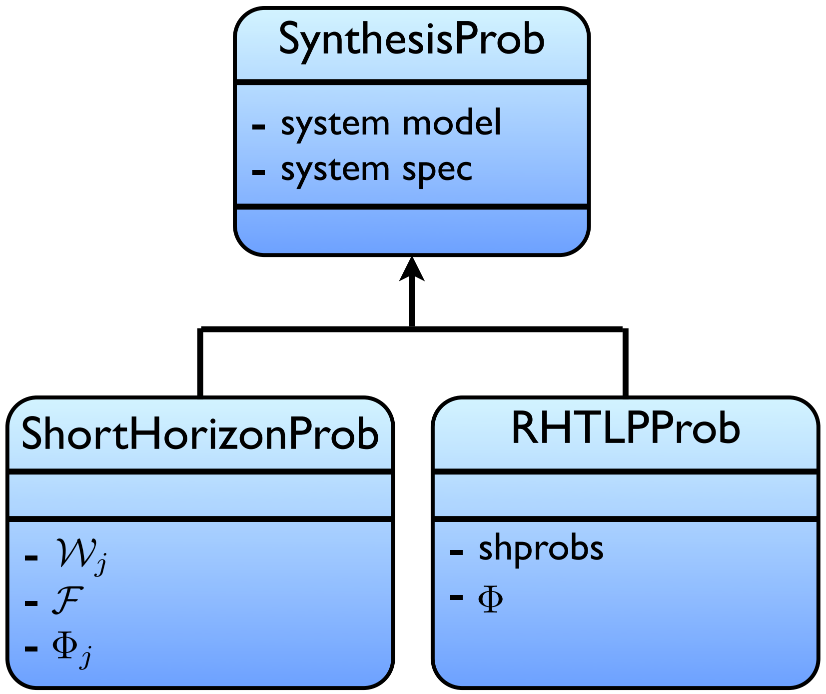
<u>Global synthesis problem</u>

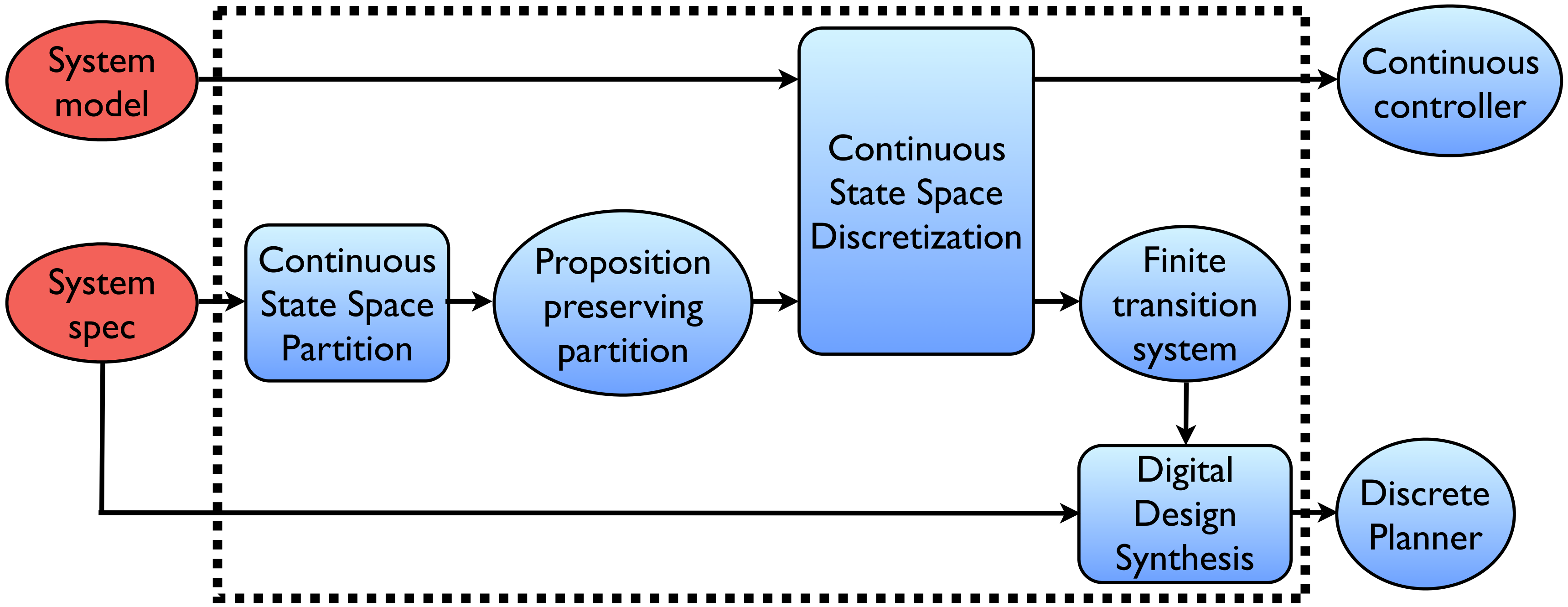
$$(\varphi_{init} \wedge \varphi_{env}) \rightarrow (\varphi_{safety} \wedge \varphi_{goal})$$

<u>Subproblems in RHTLP</u>

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

RHTLP in TuLiP





***ShortHorizonProb*: a** class for defining a short horizon problem

- **computeLocalPhi**(): compute ϕ that makes this short horizon problem realizable.

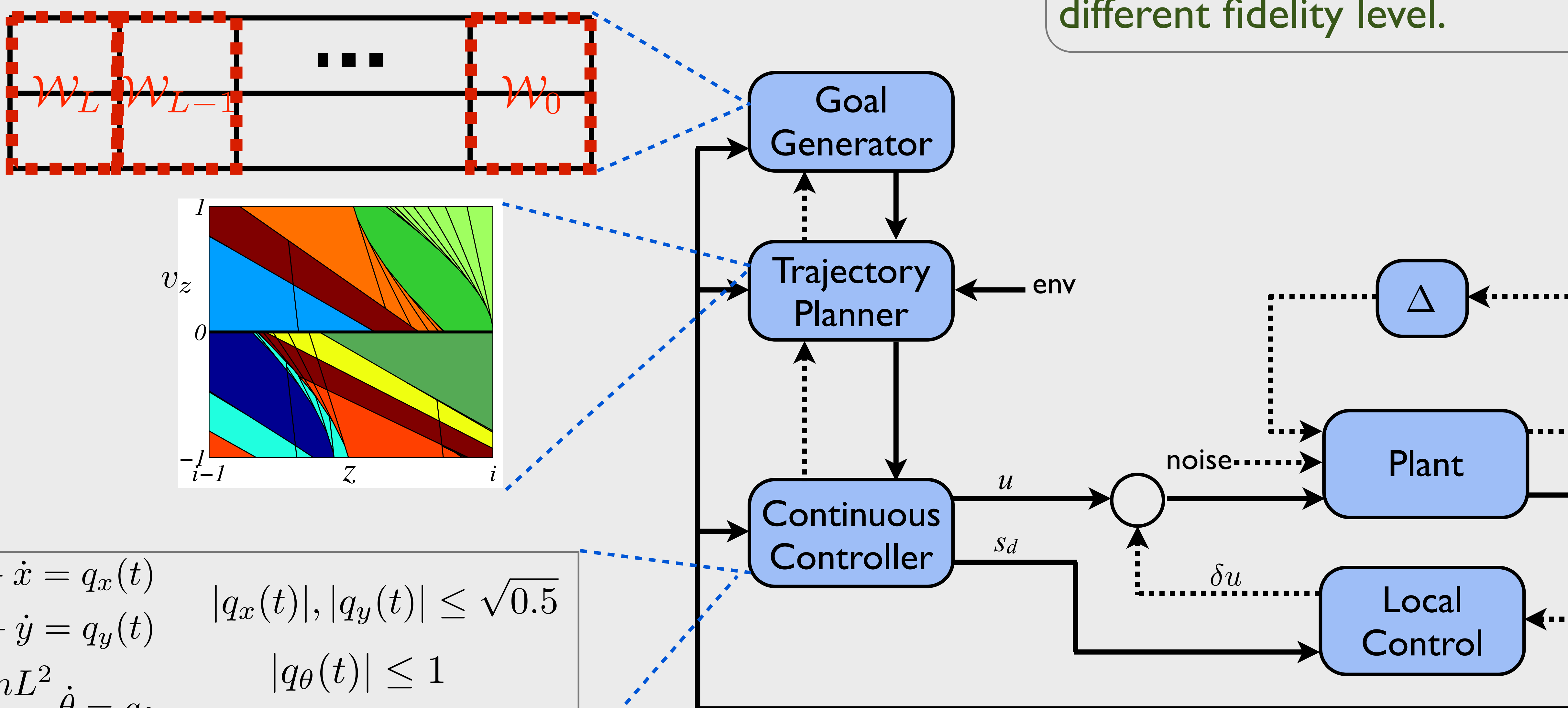
RHTLPProb: a class for defining a receding horizon temporal logic planning problem

- Contains a collection of short-horizon problems
- Useful methods
  - **computePhi**(): compute ϕ for this RHTLP problem if one exists.
  - **validate**(): validate that the sufficient conditions for applying RHTLP hold

Hierarchical control structure

Abstraction procedure and bisimulations relate models of different fidelity level.

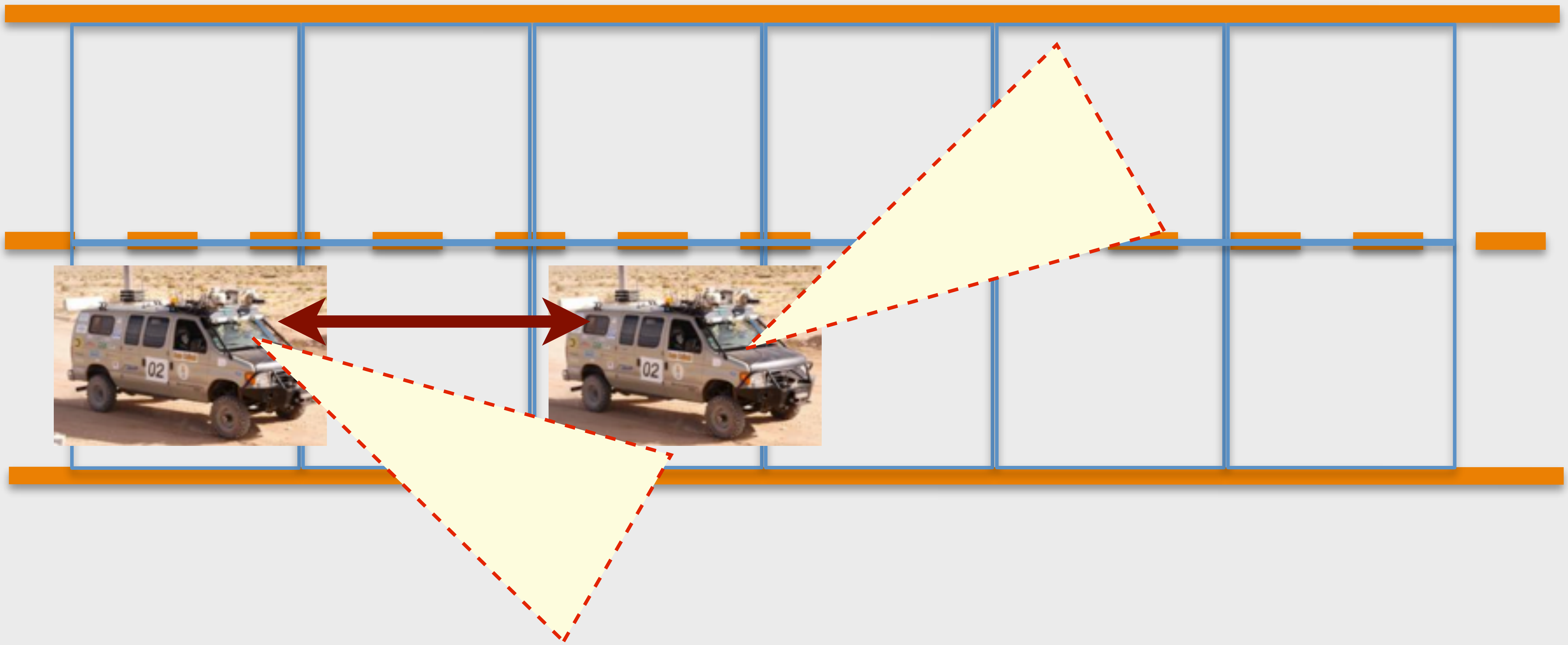
models of varying fidelity



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

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

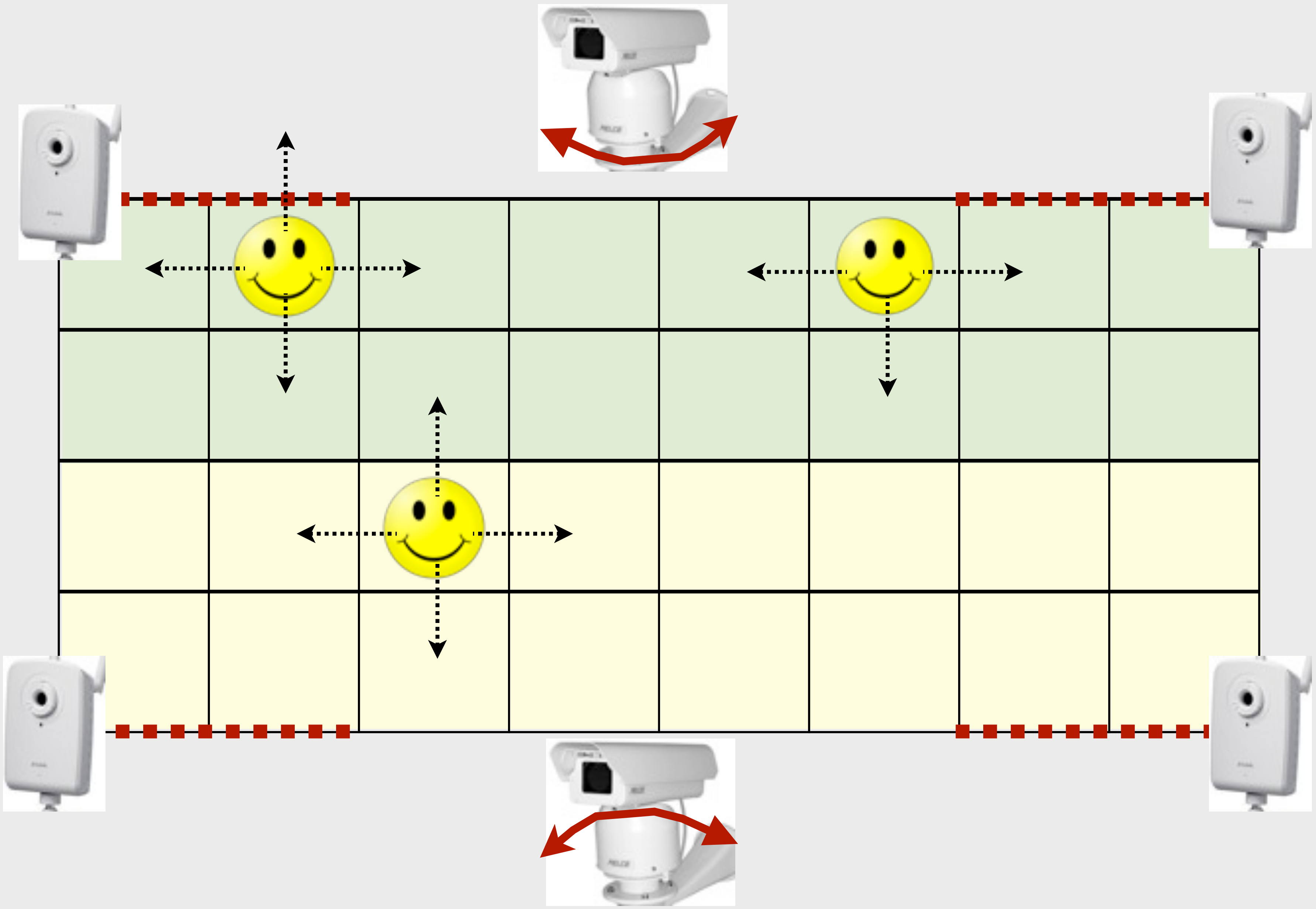
Decompositions in the state space



Decompositions induced by ...

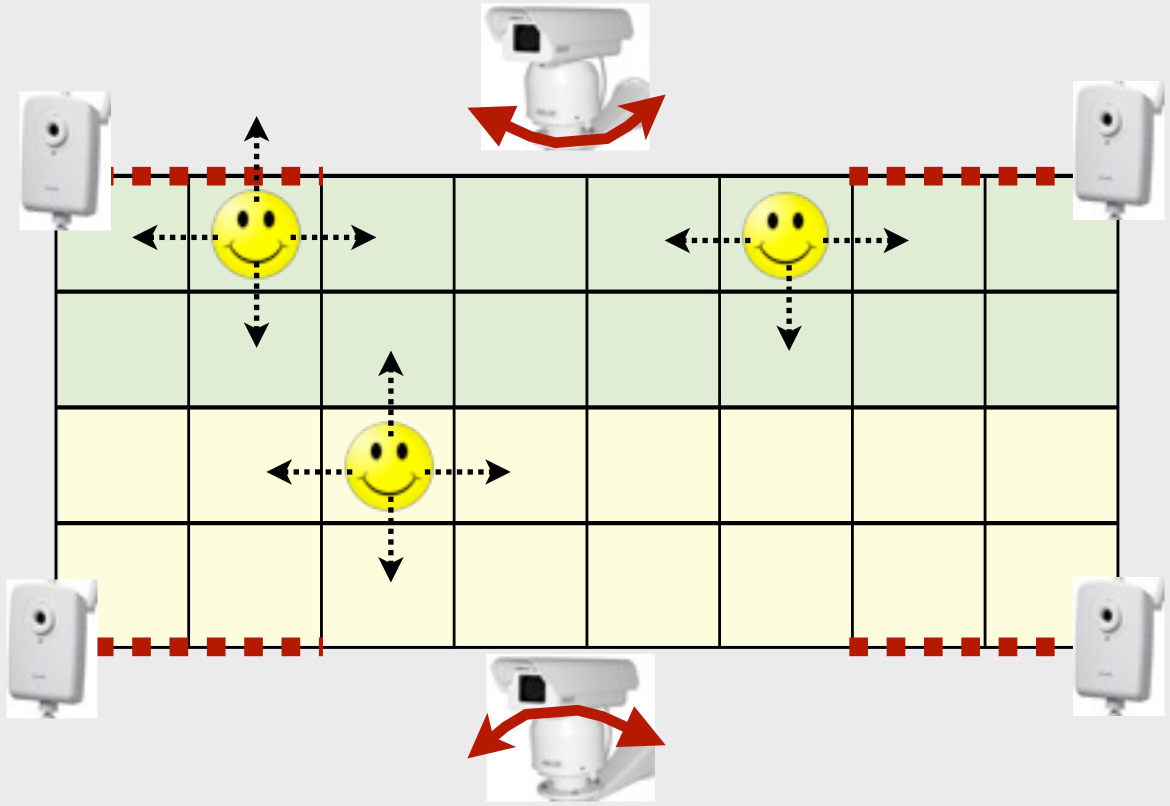
receding horizon	goal
distributed synthesis	underlying network

Smart camera networks
- static cameras for tracking targets
- pan-tilt-zoom (PTZ) for active recognition



Goal: synthesize control protocols for PTZ to ensure that one high resolution image of each target is captured at least once

Synthesis of protocols for active surveillance



System:

- region of view of PTZs
- governed by finite state automata

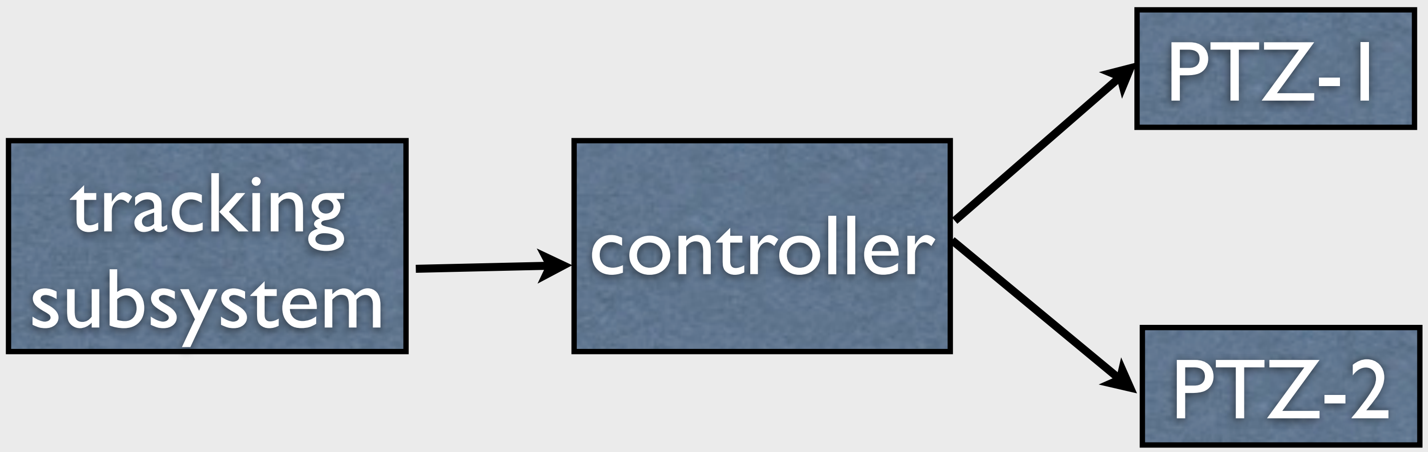
Additional requirement:

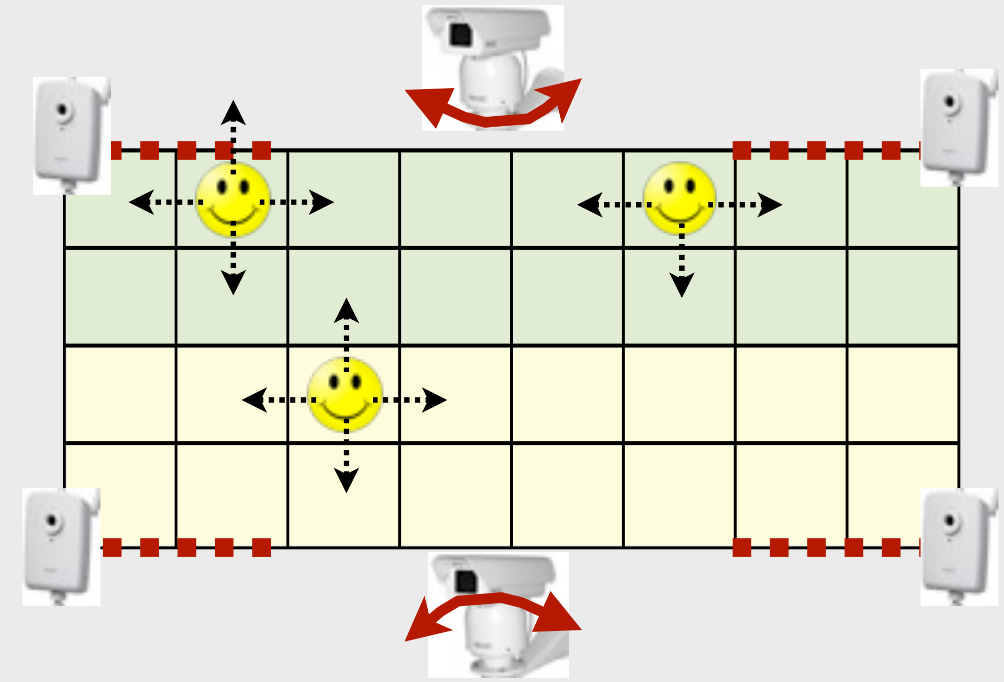
- Zoom-in the corner cells infinitely often.

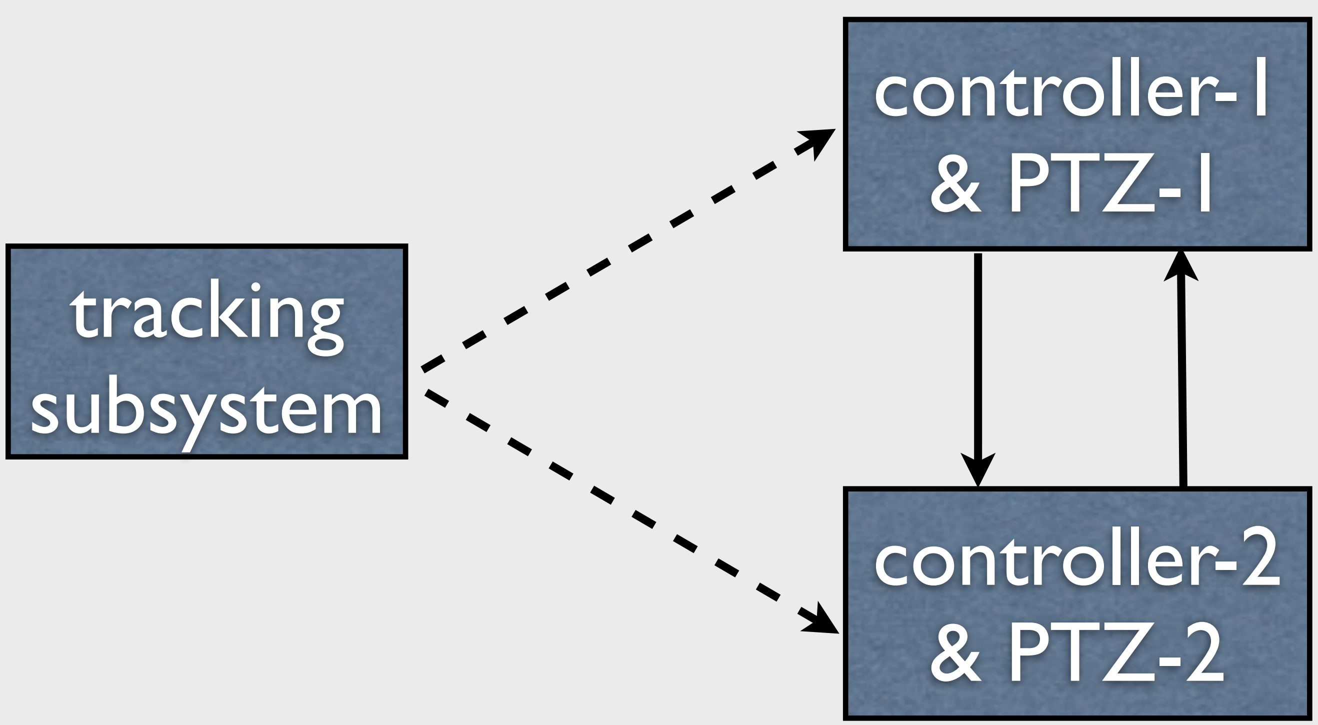
Environment specifications:

- At most N targets at a time.
- Every target remains at least T time steps and eventually leaves.
- Can only enter/exit through doors.
- Can only move to neighbors.

Centralized vs. decentralized control architecture







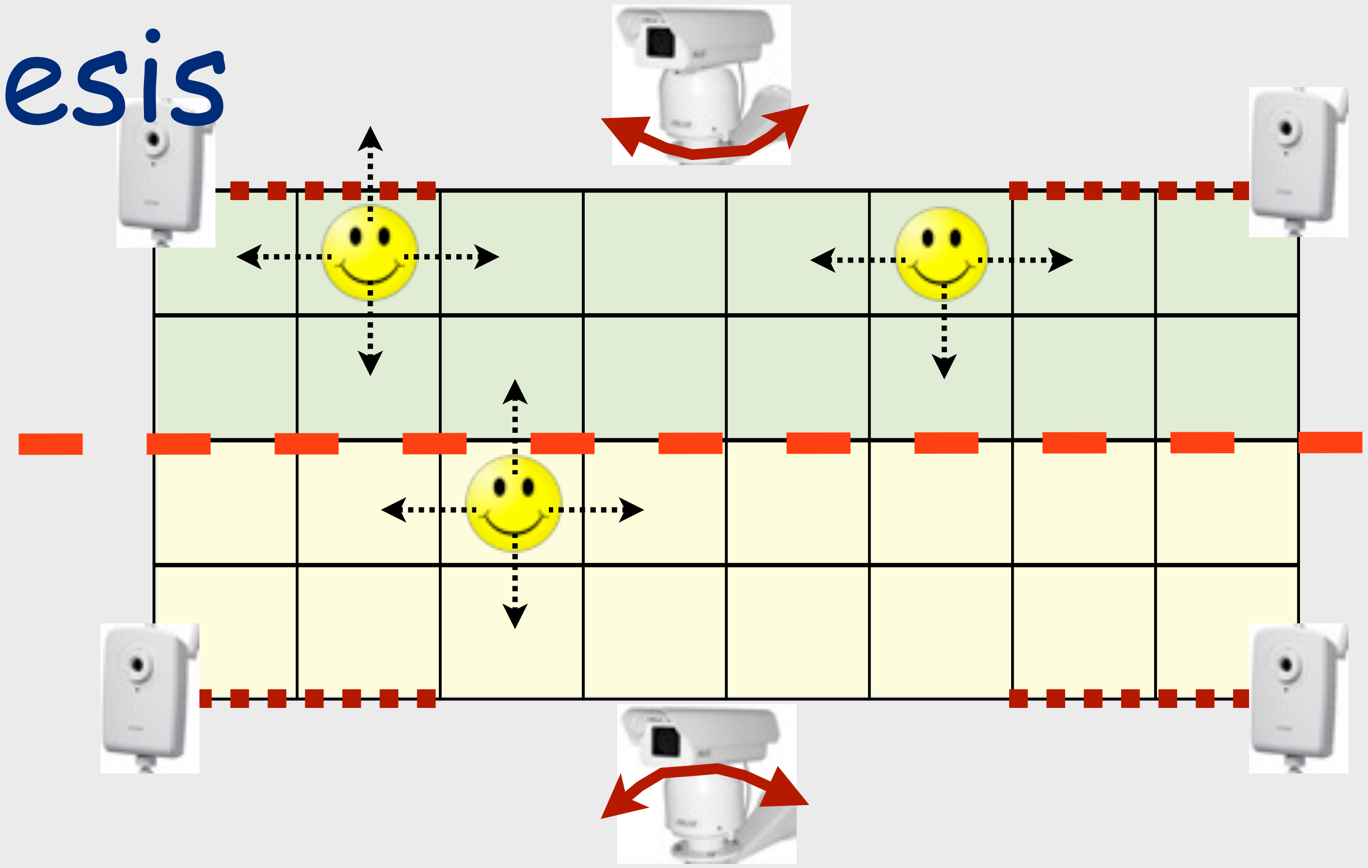
How to design control protocols that can be
- synthesized
- implemented

in a decentralized way?

What information exchange & interface models are needed?

Compositional Synthesis

<u>Goal</u>: Find control protocols for PTZ-1 & PTZ-2 so that $\varphi_e \to \varphi_s$ holds.



Simple & not very useful composition:

Any execution of the env't, satisfying φ_e, also satisfies $\varphi_{e_1} \wedge \varphi_{e_2}$

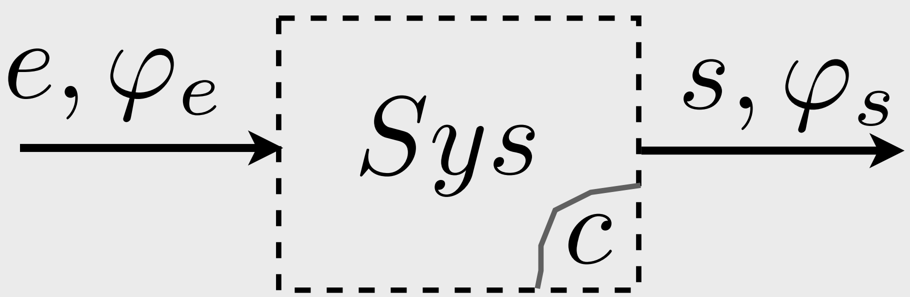
Any execution of the system, satisfying $\varphi_{s_1} \wedge \varphi_{s_2}$, also satisfies φ_s

No common controlled variables in φ_{s_1} and φ_{s_2}

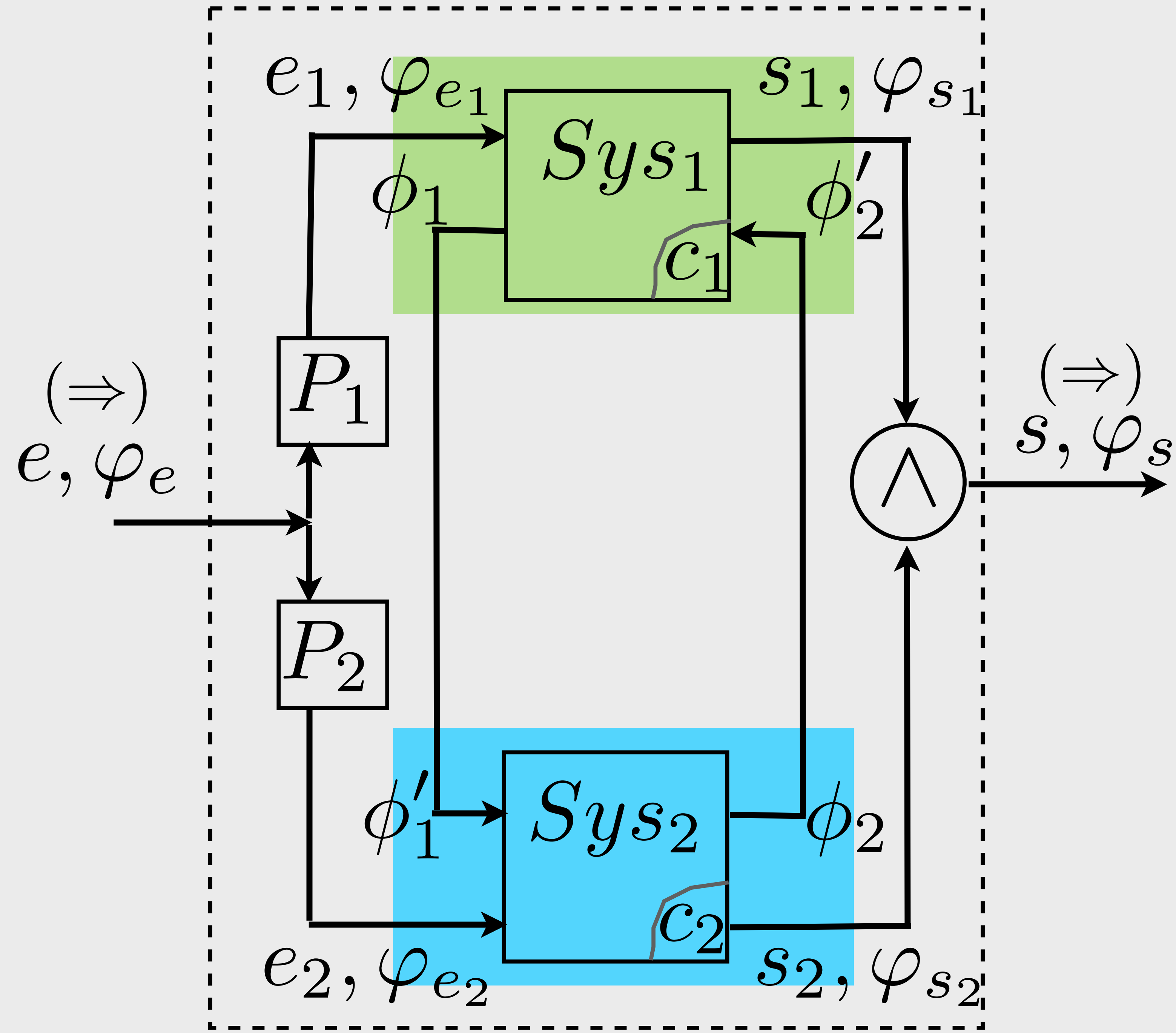
There exist control protocols that realize $\varphi_{e_1} \to \varphi_{s_1}$ & $\varphi_{e_2} \to \varphi_{s_2}$

$\Rightarrow$ $\varphi_e \to \varphi_s$ is realized.

Central



Compositional



(Refined) Compositional Synthesis

As before:

Any execution of the env't, satisfying φ_e, also satisfies $\varphi_{e_1} \wedge \varphi_{e_2}$

Any execution of the system, satisfying $\varphi_{s_1} \wedge \varphi_{s_2}$, also satisfies φ_s

No common controlled variables in φ_{s_1} and φ_{s_2}

Refined interfaces:

There exist control protocols that realize

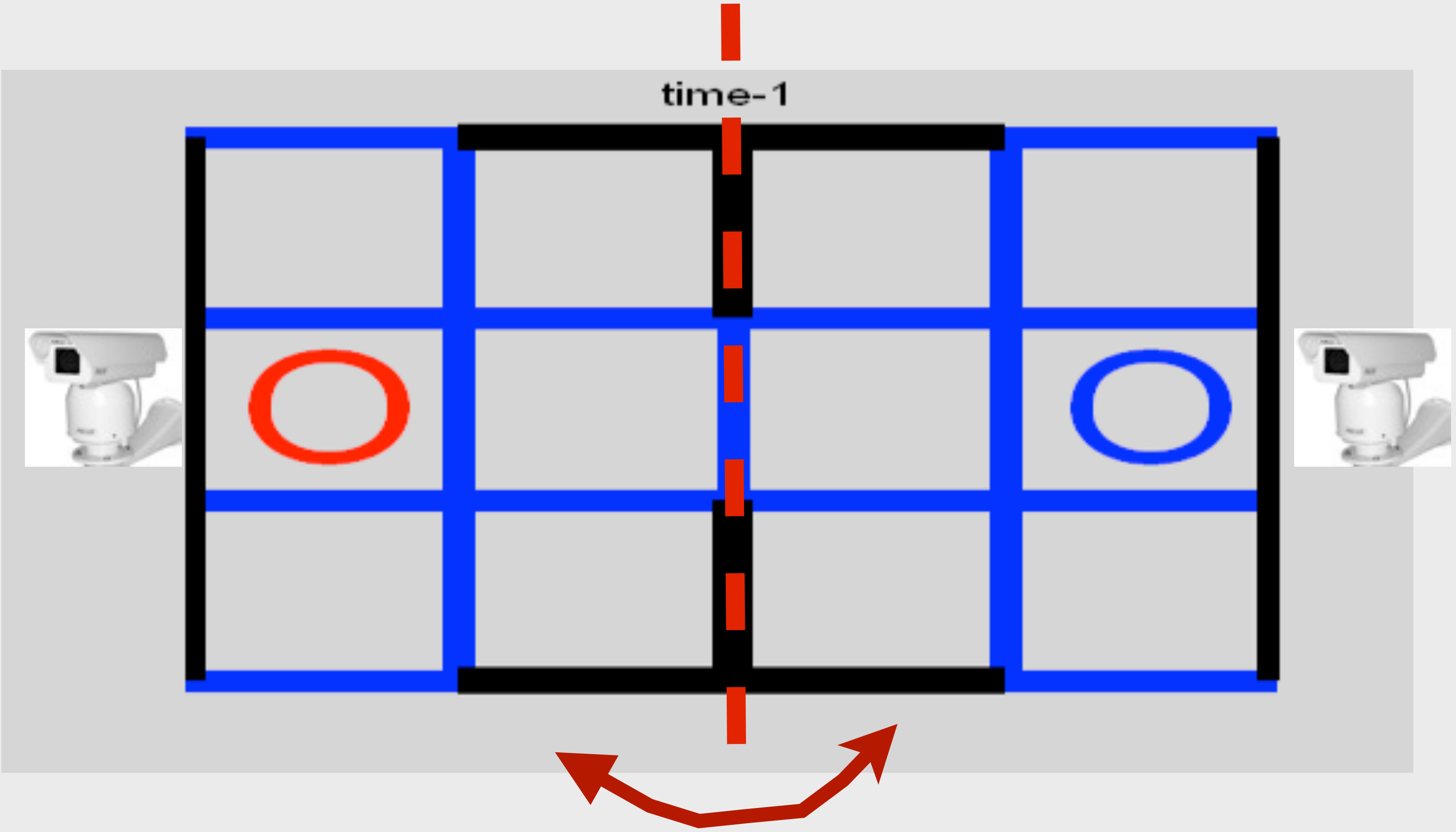
$(\phi_2' \wedge \varphi_{e_1}) \rightarrow (\varphi_{s_1} \wedge \phi_1)$ & $(\phi_1' \wedge \varphi_{e_2}) \rightarrow (\varphi_{s_2} \wedge \phi_2)$

For soundness and to avoid circularity:

$$\Box (\phi_i \rightarrow \circ \phi_i') \quad \text{for } i = 1, 2$$

➡ $\varphi_e \rightarrow \varphi_s$ is realized.

Application to a (very simple) smart camera network



IsZoomed & StepsInZone

ϕ_1 and ϕ_1'
limit the number of unzoomed targets entering zone 2 from zone 1

Case Study: Synthesis of Protocols for Electric Power Management

Multiple criticality levels:
- flight controllers
- active de-icing
- environmental control

↑ increasing criticality

Environment variables:
- wind gust (w)
- outside temperature (T)

Controlled variables:
- altitude
- power supply to different components

Source: http://www.e-envi2009.org/presentations/S3/Derouineau.pdf

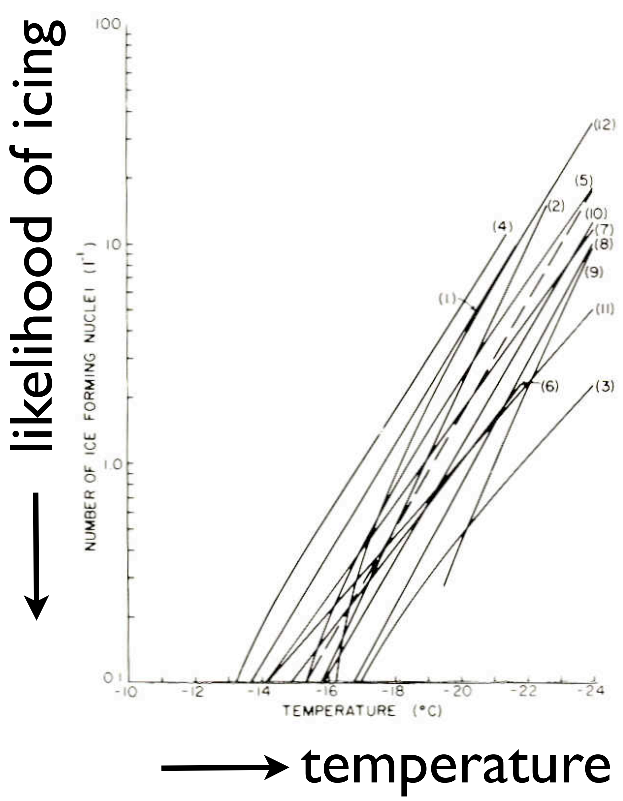
For environment & control variables, use crude discretization over their respective ranges. For example,
$T \in \{\text{low}, \text{low-medium}, \text{medium-high}, \text{high}\}$
representing the range of $[-22^o F, 32^o F]$

Dependent (state) variables:
- level of ice accumulation
- state-of-charge of the batteries
- cabin pressure level

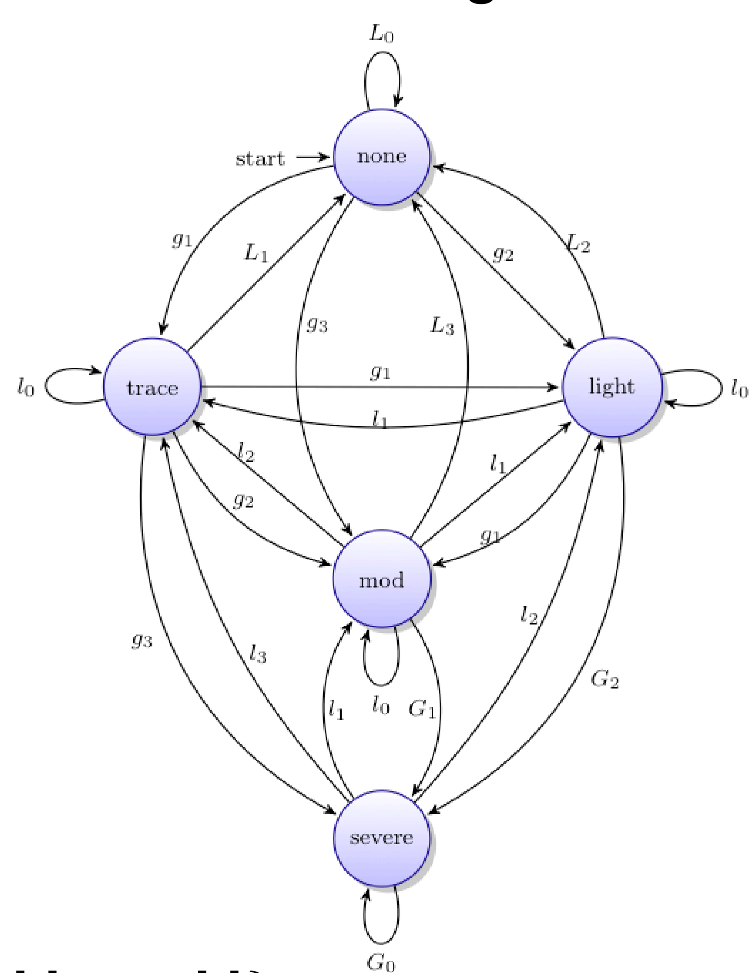
Modeling & The Dependent Variables

Use models based on finite transitions systems from a combination of empirical data and first principles.

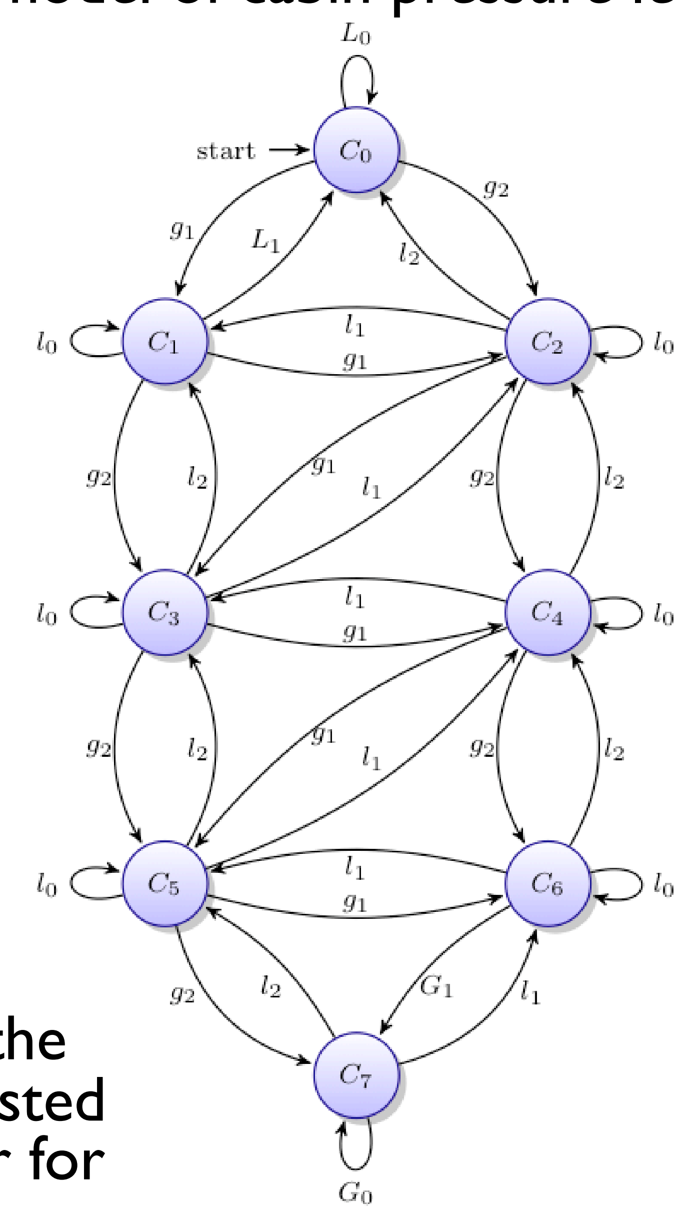
icing level	airspeed reduction	power increase to regain airspeed	climb-rate reduction	reduction in control authority
trace	< 10 knots	$< 10\%$	$< 10\%$	no effect
light	$10 - 19$ knots	$10 - 19\%$	$10 - 19\%$	no effect
moderate	$20 - 39$ knots	$20 - 39\%$	$\geq 20\%$	slow or overly sensitive response
severe	≥ 40 knots	unable	unable	limited or no response



model of icing level



model of cabin pressure level



State-of-charge evolves with:

$$b[t+1] = \min\{B, b[t] + \bar{P} - p_f[t] - p_d[t] - p_e[t]\}$$

storage capacity — generation capacity — power supply to each functionality

Transitions model the gap between requested and supplied power for each functionality.

Sample Specifications

power requests from flight controller (f), deicing (d), and pressure control (e):

$$r_f \equiv r_f(h, a, w)$$
$$r_d \equiv r_d(T, h)$$
$$r_e \equiv r_e(T, h)$$

Resource constraint: $\square(p_f + p_d + p_e \leq \bar{P} + b)$

Prioritization: $\square(p_f \geq r_f)$

$\square(p_f = \text{high} \wedge p_d = \text{high} \Rightarrow p_e = \text{low})$

Safety: Altitude cannot change too much between to consecutive instants, e.g.,

$\square\,(h = \text{low} \Rightarrow (\circ h \neq \text{medium-high} \wedge \circ h \neq \text{high}))$

Ice accumulation limits allowable altitude change, e.g.,

$\square(a = \text{severe} \Rightarrow \circ h = h)$

Ice accumulation cannot be severe: $\square(a \neq \text{severe})$

Performance: Cabin pressure does not exceed the level at 8000 ft.

Always go back to the desirable altitude: $\square \diamond (h = \text{high})$

Assumptions: Wind gusts cannot be severe too many consecutive steps.

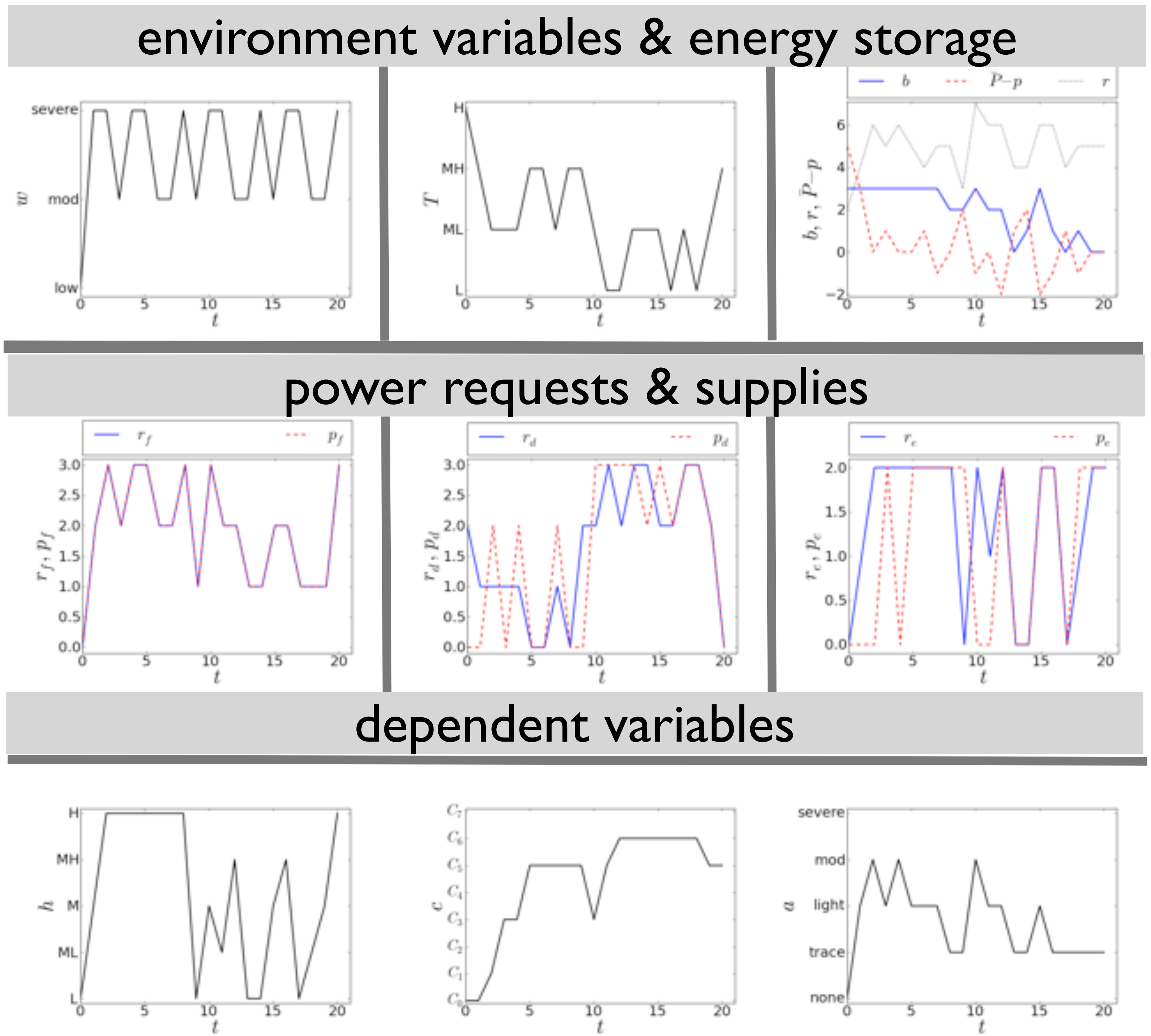
$\square(n_w \geq N_w \Rightarrow \circ(w \neq \text{severe})$

No abrupt change in outside temperature, e.g.,

$\square(T = \text{medium-low} \Rightarrow \circ T \neq \text{high})$

Notation may not be fully explained. Ask, if confused!!!

Dynamic power allocation allows reductions in peak power (i.e., generator weight) requirements.



Formulate as a temporal logic, reactive planning problem

$$(\varphi_{environment} \wedge \varphi_{initial} \wedge \varphi_{criticality}) \rightarrow (\varphi_{performance} \wedge \varphi_{safety})$$

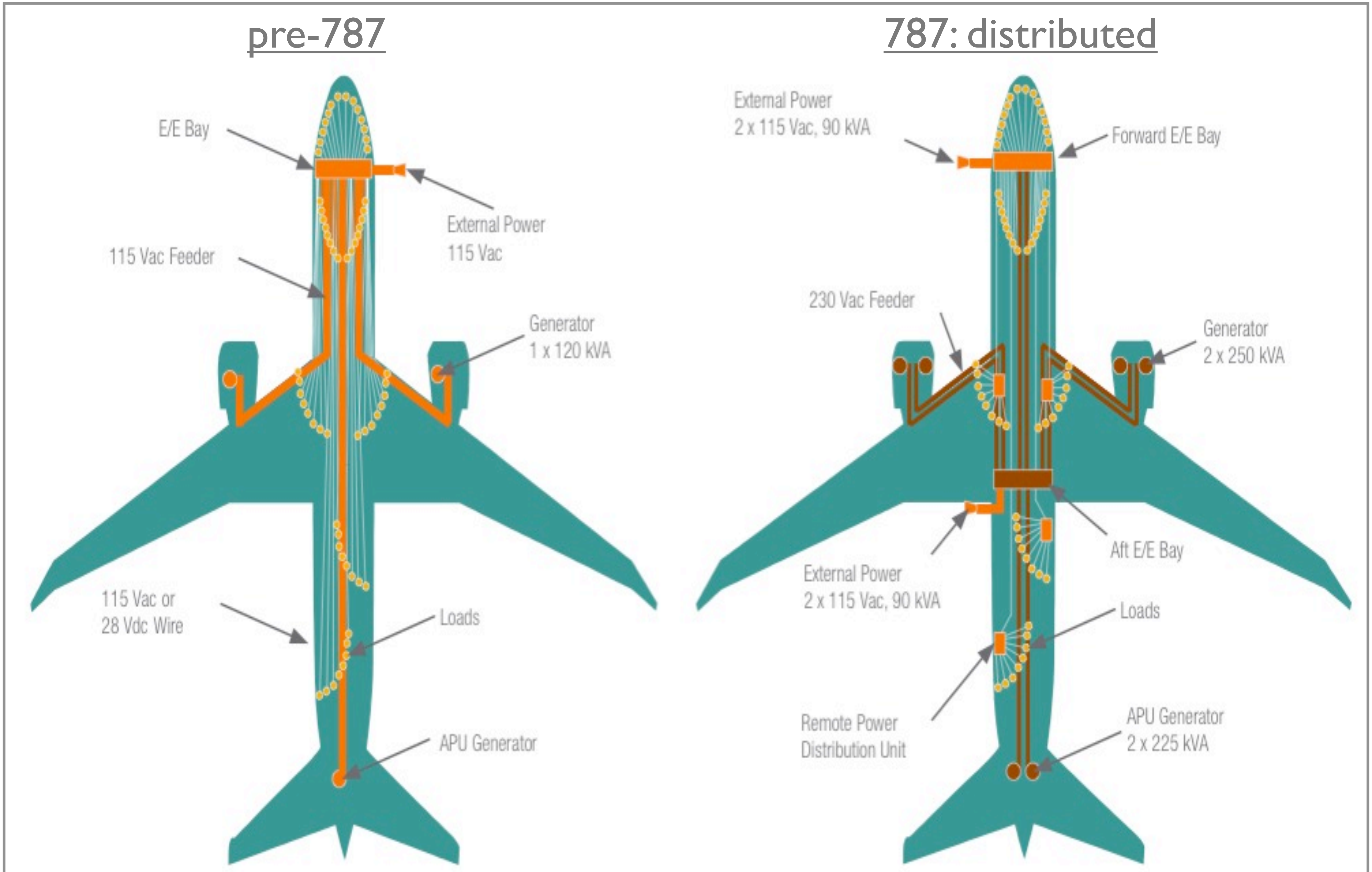
$N_w = 2, B = 3$

$\bar{P} = 5$

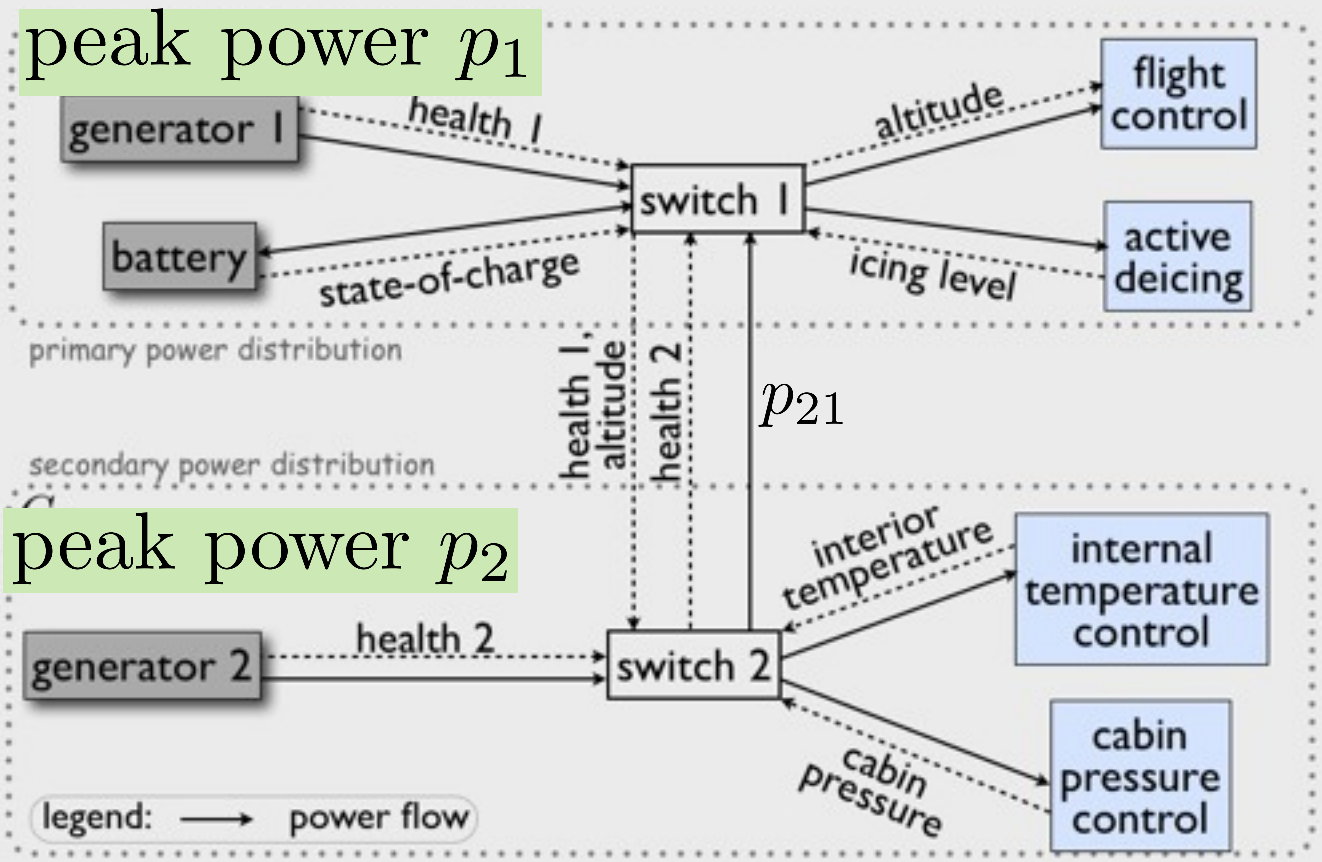
$r_f, r_d \in \{0, 1, 2, 3\}$

$r_e \in \{0, 1, 2\}$

Conventional vs. Boeing 787 Electric Power Network Structure



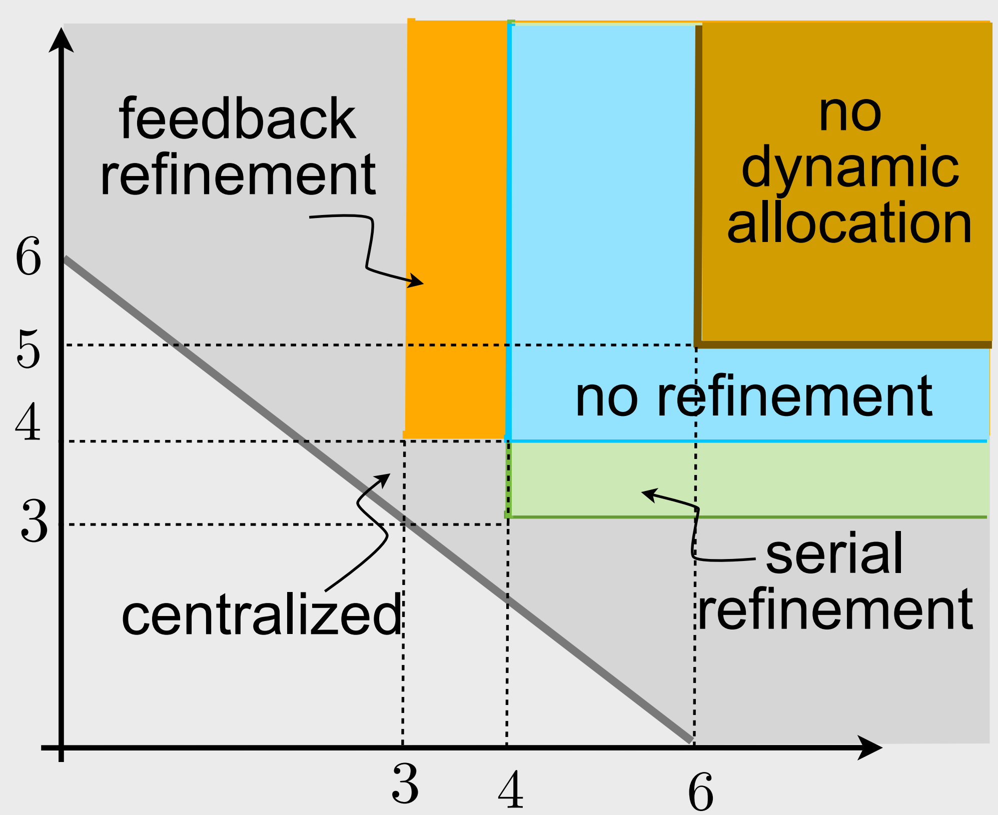
Distributed resource allocation



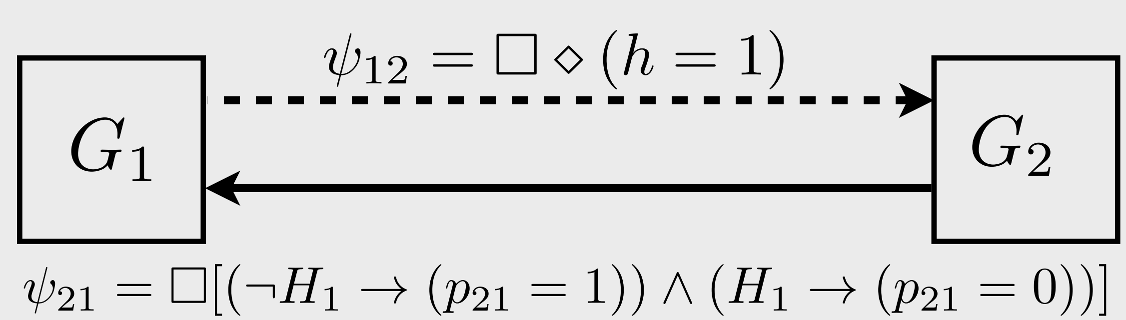
Controlled variables:
- Power supplies to each function
- Altitude

Environment variables:
- Wind gusts
- Outside temperature
- Generator health status

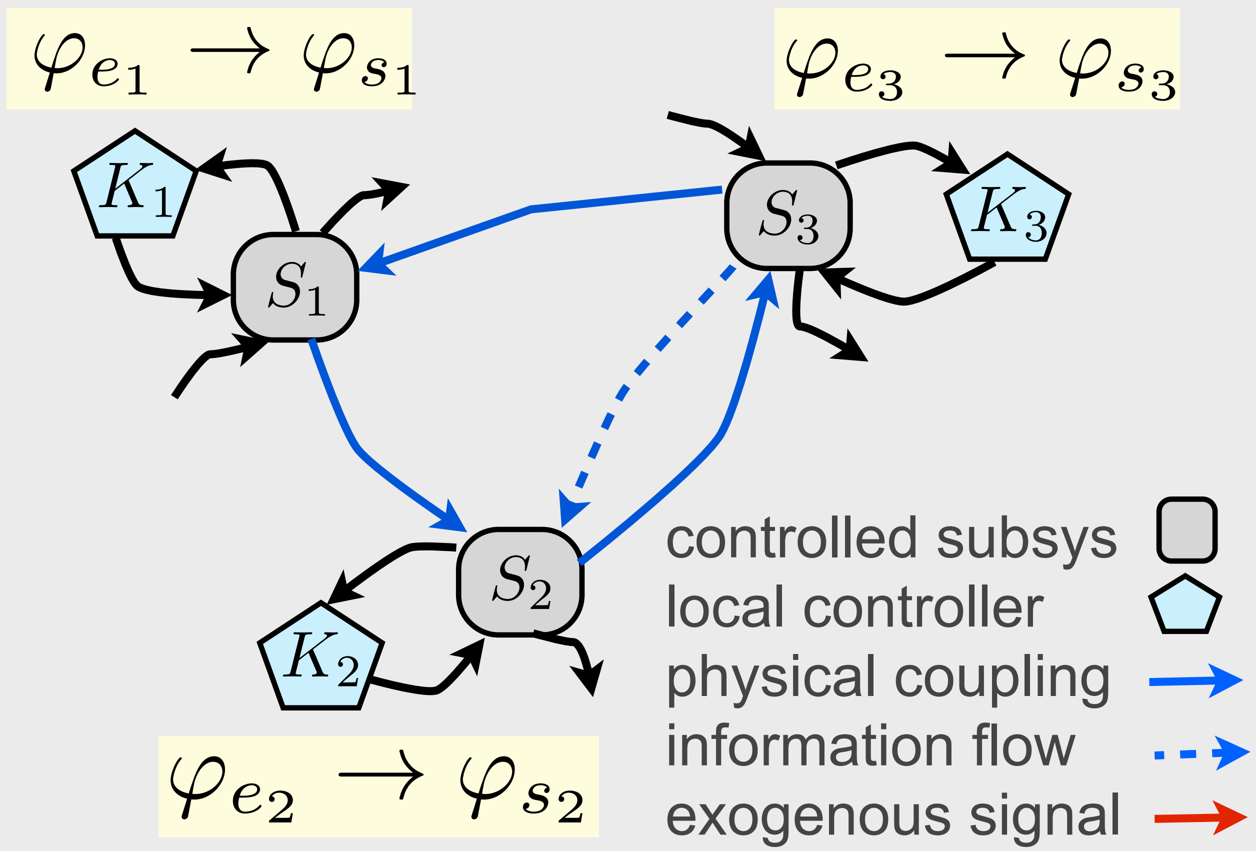
Dependent variables:
- Level of ice accumulation
- State-of-charge of the battery
- Cabin pressure & temperature



Interface refinements



Compositional Synthesis of Distributed Protocols



$$\underbrace{\bigwedge_i \varphi_{e_i} \rightarrow \varphi_e}_{\text{"weaker" environment assumptions}} \rightarrow \underbrace{\varphi_s \rightarrow \bigwedge_i \varphi_{s_i}}_{\text{"stronger" system requirements}}$$

<u>Extra (mild) technical conditions</u>: No common controlled variables & loops are well-posed.

Theorem: $\varphi_e \rightarrow \varphi_s$ is realizable if every $\varphi_{e_i} \rightarrow \varphi_{s_i}$ is realizable.

Contracts formalize the coupling and information exchange between subsystems.

Trade-offs:

conservatism vs. expressiveness of contracts vs. need for coordination & computational cost