Motion Planning with Dynamics, Physics-based Simulations, and Linear Temporal Objectives

Erion Plaku

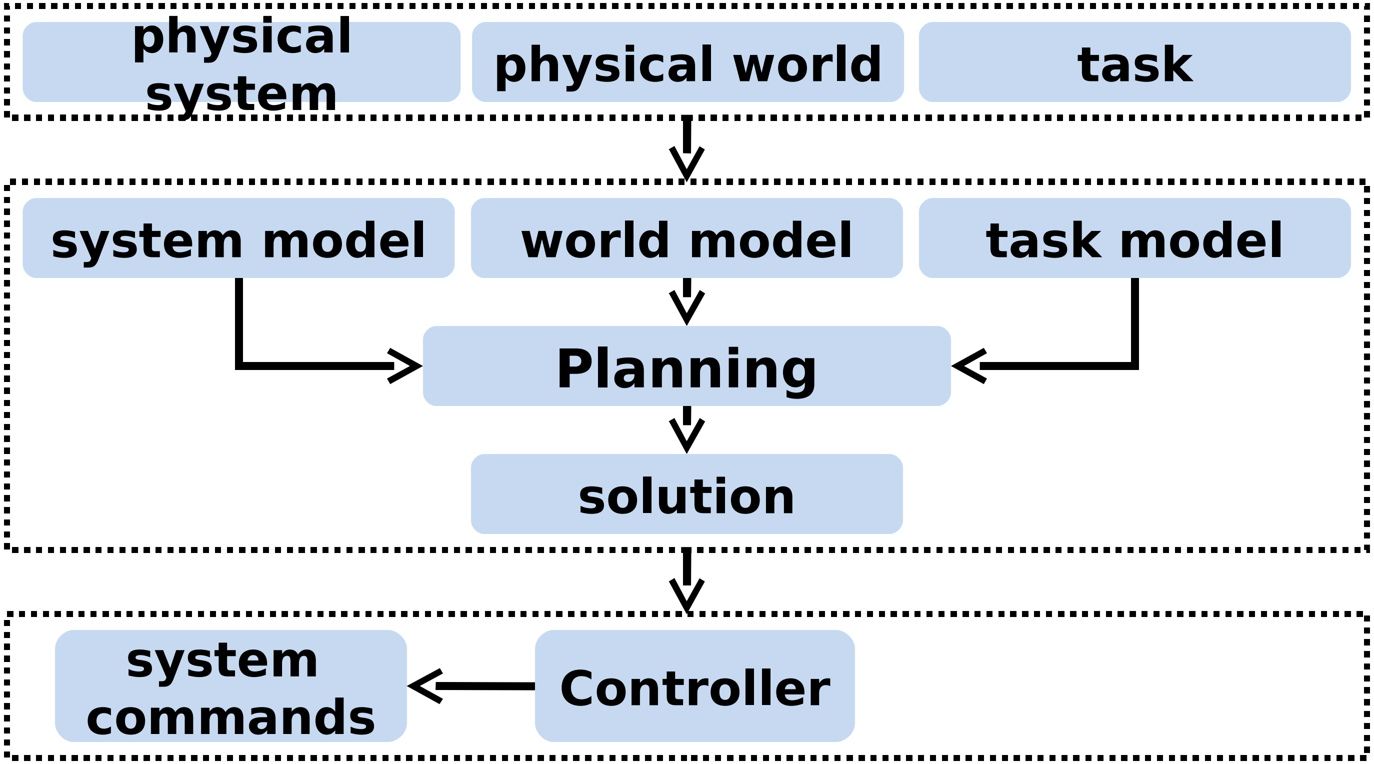
Laboratory for Computational Sensing and Robotics
Johns Hopkins University

Frontiers of Planning

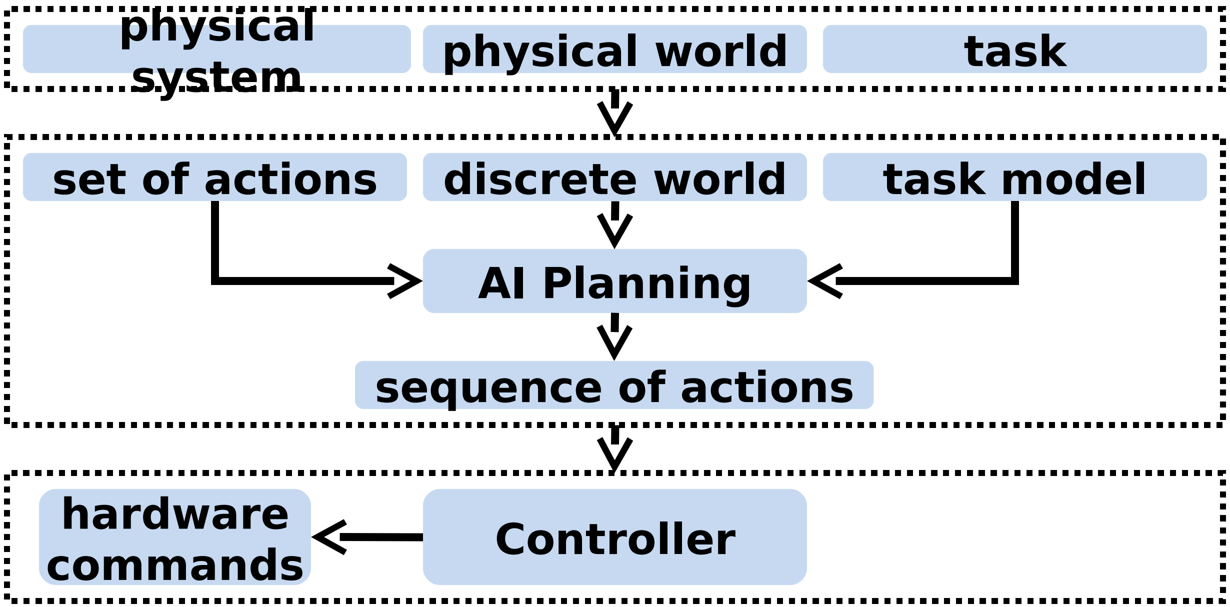
The goal is to be able to specify a **task** and have the planning system compute a **sequence of actions** to **accomplish** the task

(Simplified) Planning Schema

The goal is to be able to specify a **task** and have the planning system compute a **sequence of actions** to **accomplish** the task



Classic AI Planning

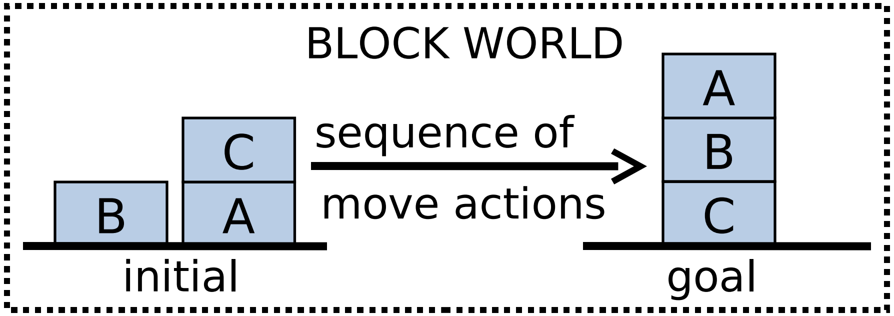


Applications
- Robotics
- Decision making
- Resource handling
- Game playing
- Model checking …

Planners
- STRIPS [Stanford]
- Graphplan [CMU]
- Blackbox [AT&T Labs] …

Advantages

Effectively handles
- Large number of states and actions
- Rich task models, e.g., **reachability** and **temporal objectives**

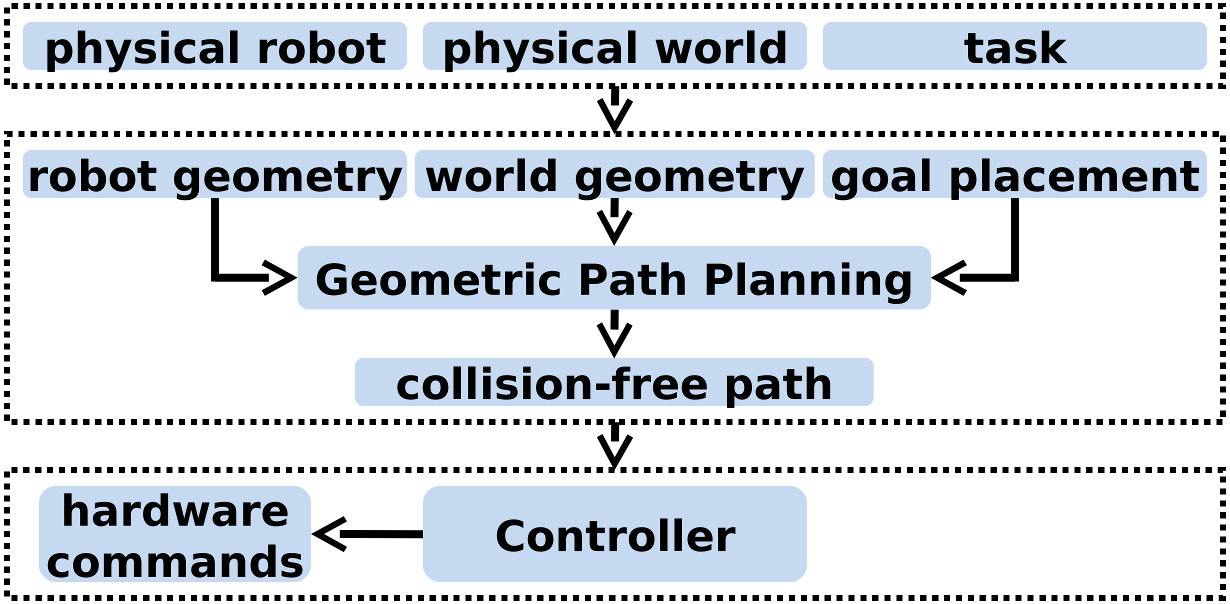


Limitations

- **Discrete world**
- **Finite set of discrete actions**
- Difficult to design general controllers that can follow sequence of actions

Planning in a continuous setting?

Geometric Path Planning



Applications

- Robotics
- Assembly
- Manipulation
- Character animation
- Computational biology …

Advantages

Effectively handles

- Collision avoidances
- High-dimensional continuous spaces

Limitations of Geometric Path Planning

1. Geometric path planning ignores

- robot dynamics
- robot interactions with the environment
- external forces, e.g., friction, gravity

⇒ Geometric paths are difficult to follow

2. Current methods in geometric path planning cannot handle

- **Temporal objectives**: reach desired states w.r.t. a linear ordering of time, i.e., “A or B” “A and B” “B after A” “B next to A”

  Example:

  “inspect all the contaminated areas, then visit one of the decontamination stations, and then return to the base”

Planning with rich models of the robot and physical world?

Significantly increases problem complexity
Renders current planners computationally impractical

Planning with temporal objectives?

Significantly increases problem complexity
Currently possible only in a discrete setting

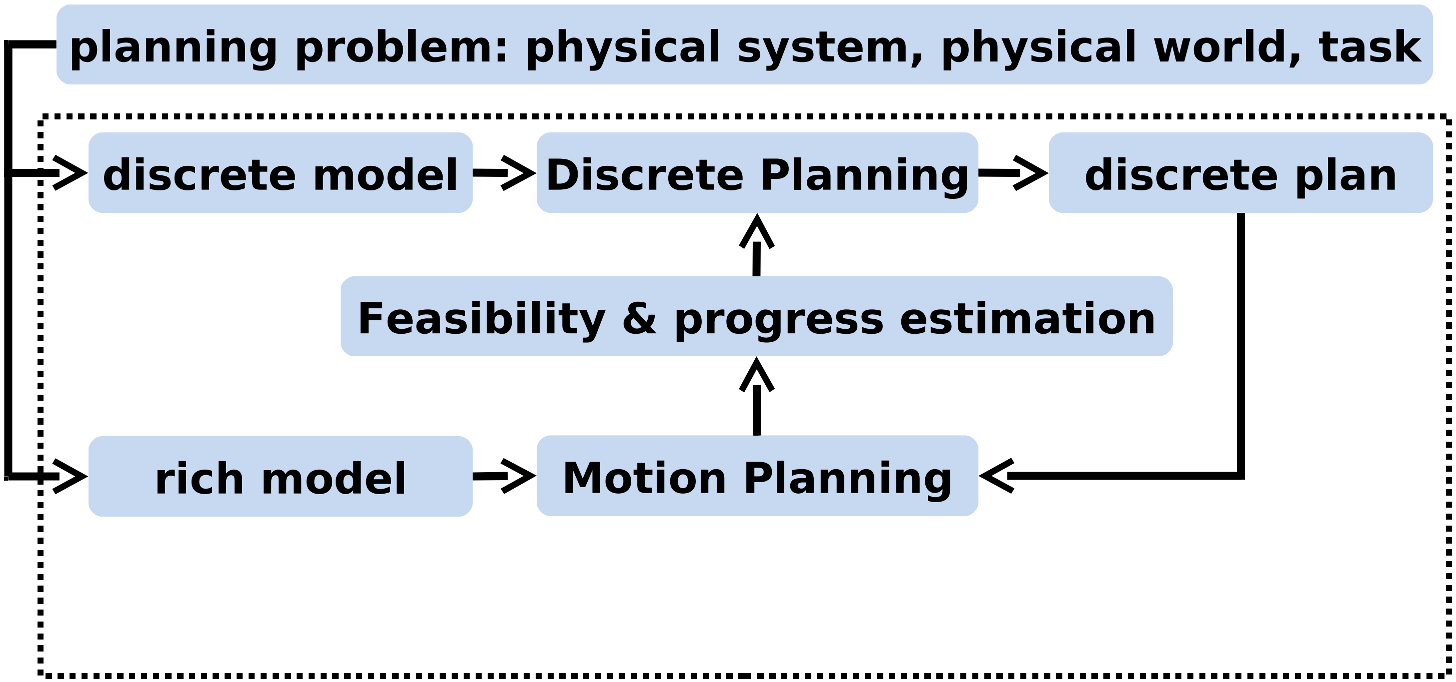
Approach

Discrete Planning

- Artificial Intelligence
- Computer Logic

Motion Planning

- Probabilistic Sampling
- Control Theory

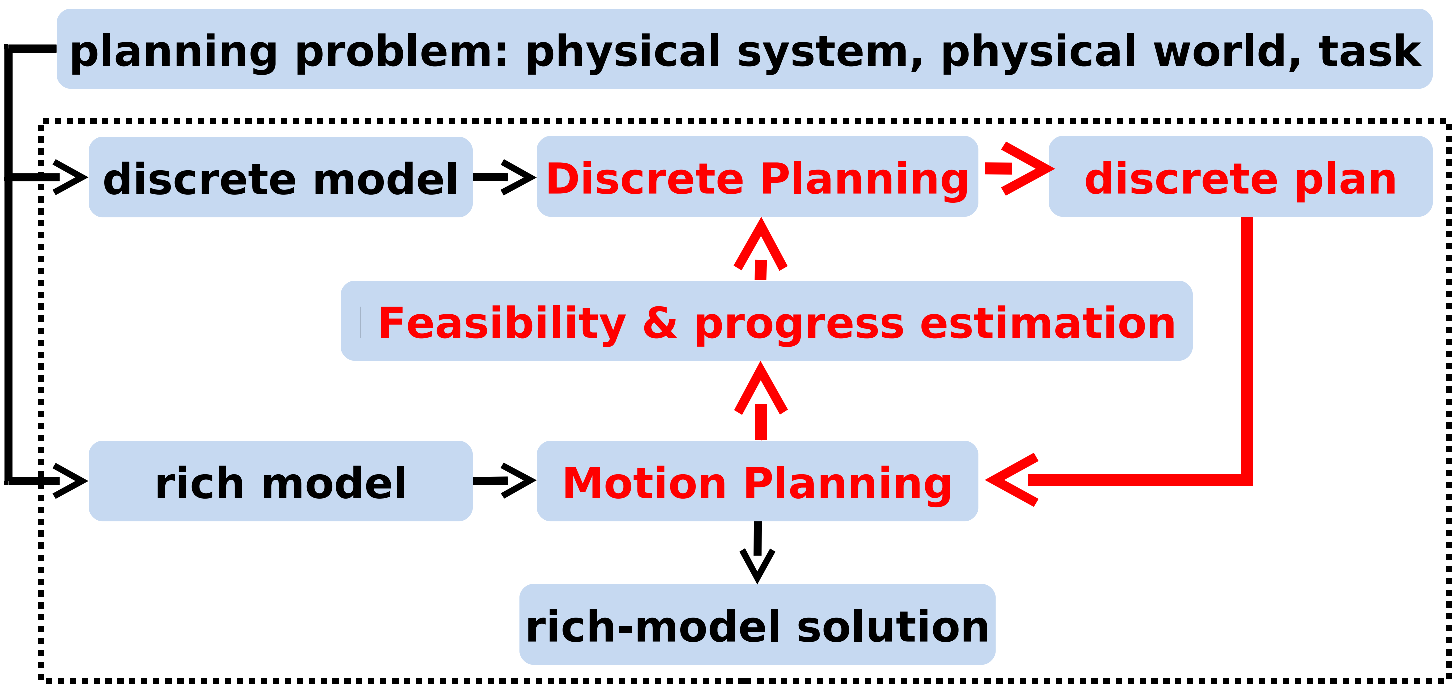


SyCLoP: Synergic Combination of Layers of Planning

Discrete Planning
- Artificial Intelligence
- Computer Logic

synergic combination ⟷

Motion Planning
- Probabilistic Sampling
- Control Theory



Tasks
- Reachability
- Temporal objectives

Plaku, Kavraki, Vardi:
TRO05, ICRA07, RSS07
CAV07, ICRA08,
FMSD08 , TACAS08

Rich Models
- Nonlinear Dynamics
- Physical Realism
- Hybrid Systems

Overview

- Motion Planning: Background & Related Work
- SyCLoP: Synergic Combination of Layers of Planning
- Applications of SyCLoP to Motion Planning with
  - Dynamics
  - Physics-based Simulations
  - Temporal Objectives
- Discussion

Motion-Planning Problem

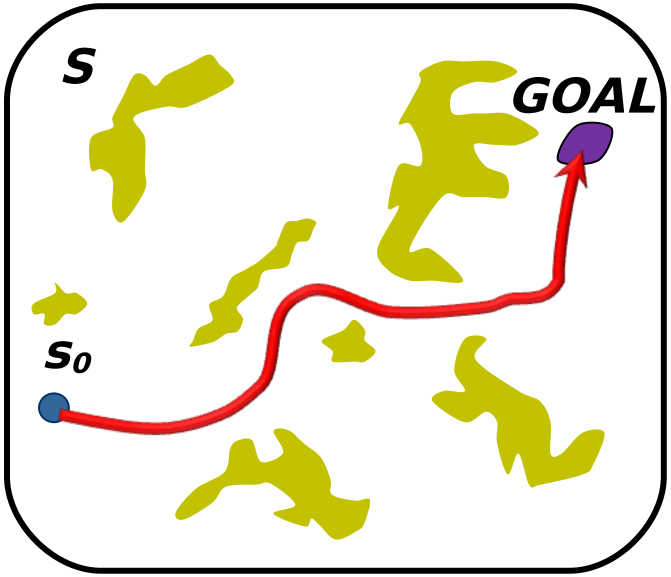
MPP = (S, INVALID, s_0, GOAL, U, f)

State Space — ***S*** — collection of variables that describe the system and world state

s ∈ ***S*** → ***INVALID*** → ***{true,false}***

s ∈ ***S*** → ***GOAL*** → ***{true, false}***

Control Space — ***U*** — controls/actions



Control Simulation

s, ***u***, ***t*** → ***f*** → ***s_{new}***

Compute a trajectory
ζ : [0, T] → S
such that

1. *ζ (0) = s_0*

2. *INVALID(ζ (t)) = false, ∀ t ∈ [0, T]*

3. *GOAL(ζ (T)) = true*

- Motion obeys physical constraints
- Accounts for system dynamics
- Accounts for interactions of the system with the world

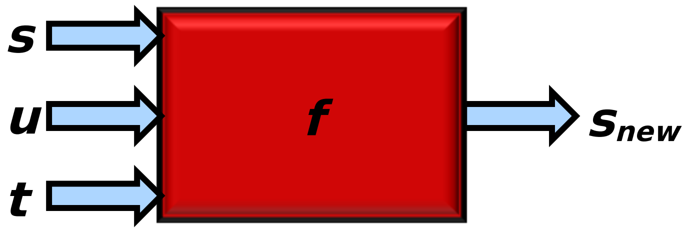
Tree-Search Framework in Motion Planning

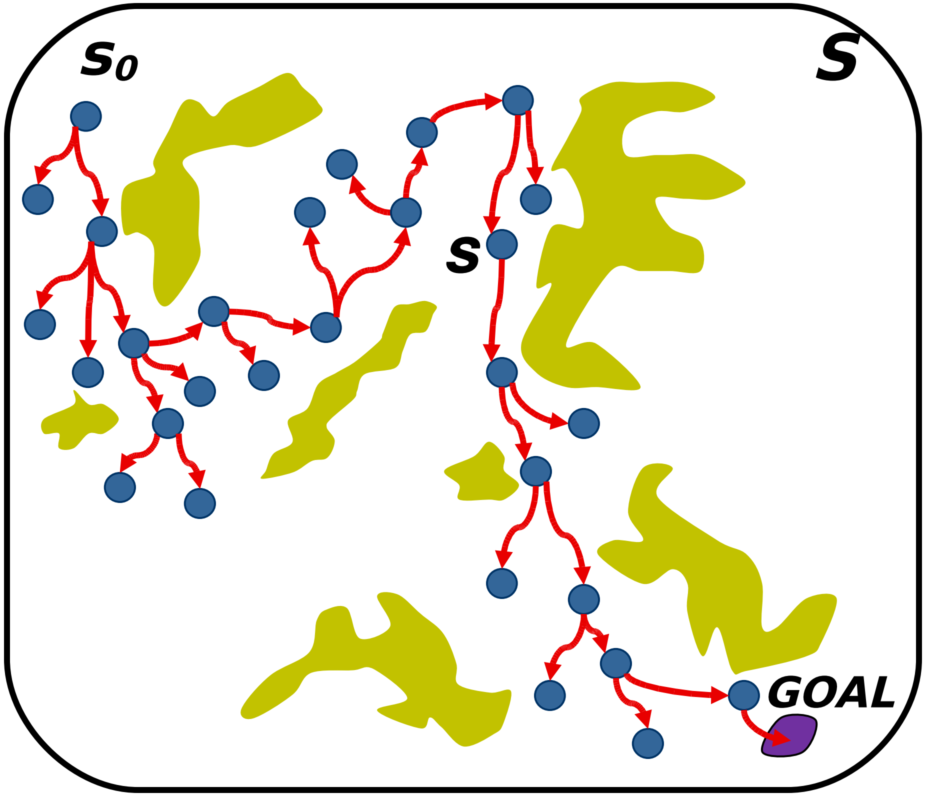
Search the state space ***S*** by growing a tree ***T*** rooted at the initial state s_0

REPEAT UNTIL GOAL IS REACHED

1. Select a state ***s*** from ***T***
2. Select a control ***u***
3. Select a time duration ***t***
4. Extend tree from ***s*** by applying the control ***u*** for ***t*** time units

Control Simulation





Related Work

- Probabilistic Roadmap Method PRM [Kavraki, Svestka, Latombe, Overmars '96]
- Obstacle based PRM [Amato, Bayazit, Dale '98]
- Expansive Space Tree (EST) [Hsu et al., '97, '00]
- Rapidly-exploring Random Tree (RRT) [Kuffner, LaValle '99, '01]
- Gaussian PRM [Boor, Overmars, van der Stappen '01]
- Single Query Bidirectional Lazy Tree (SBL) [Sanchez, Latombe '01]
- Extended Execution RRT (ERRT) [Bruce, Veloso '02]
- Guided Expansive Space Tree [Phillips et al. '03]
- Random Bridge Building Planner [Hsu, Jiang, Reif, Sun '03]
- Adaptive Dynamic Domain RRT (ADRRT) [Yershova et al., '04, '05]
- PDST [Ladd, Kavraki '04, '05]
- Utility-guided RRT [Burns, Brock '07]
- Particle RRT [Nik, Reid '07]
- GRIP [Bekris, Kavraki '07]
- Multipartite RRT [Zucker et al., '07]
- ...

Issues in Current Motion-Planning Approaches

On challenging motion-planning problems

- Exploration frequently gets stuck
- Progress slows down

Possible causes

(i) Exploration guided by limited information, such as distance metrics and nearest neighbors

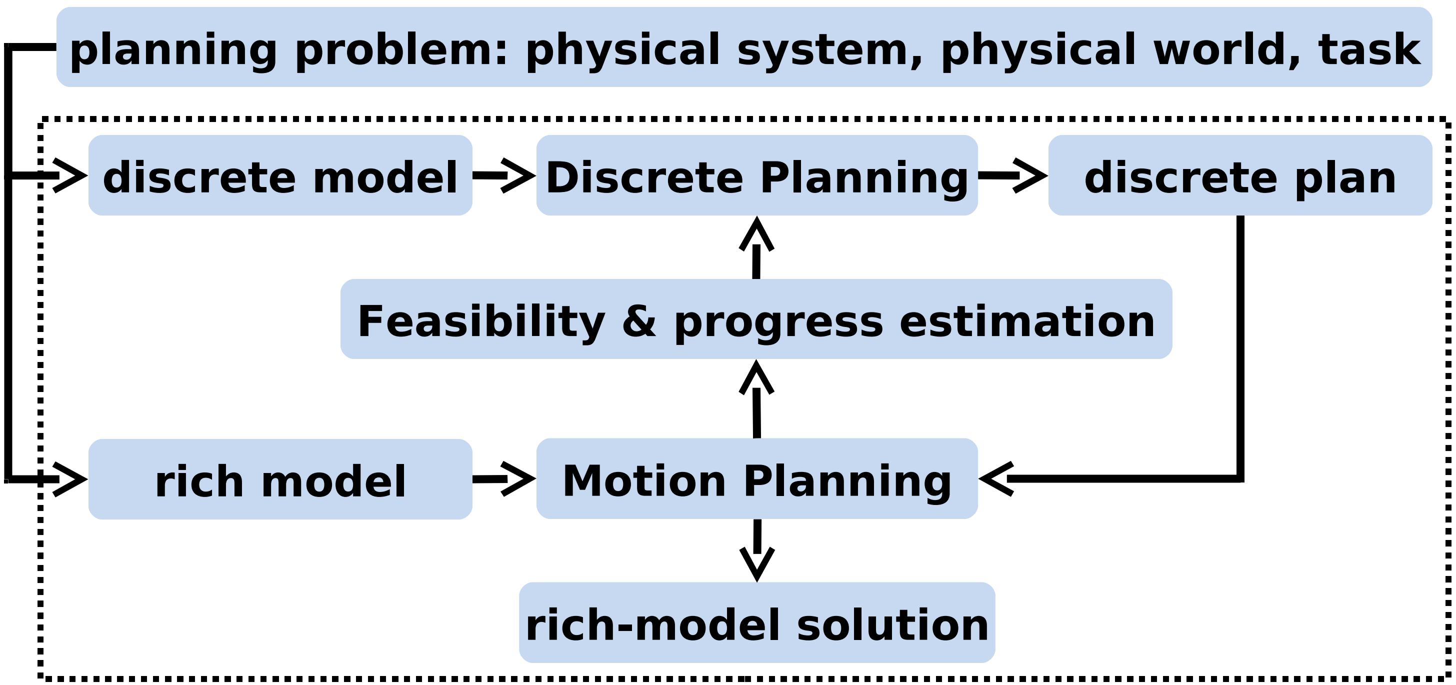
(ii) Lack of global sense of direction toward goal

(iii) Difficult to discover new promising directions toward goal

Overview

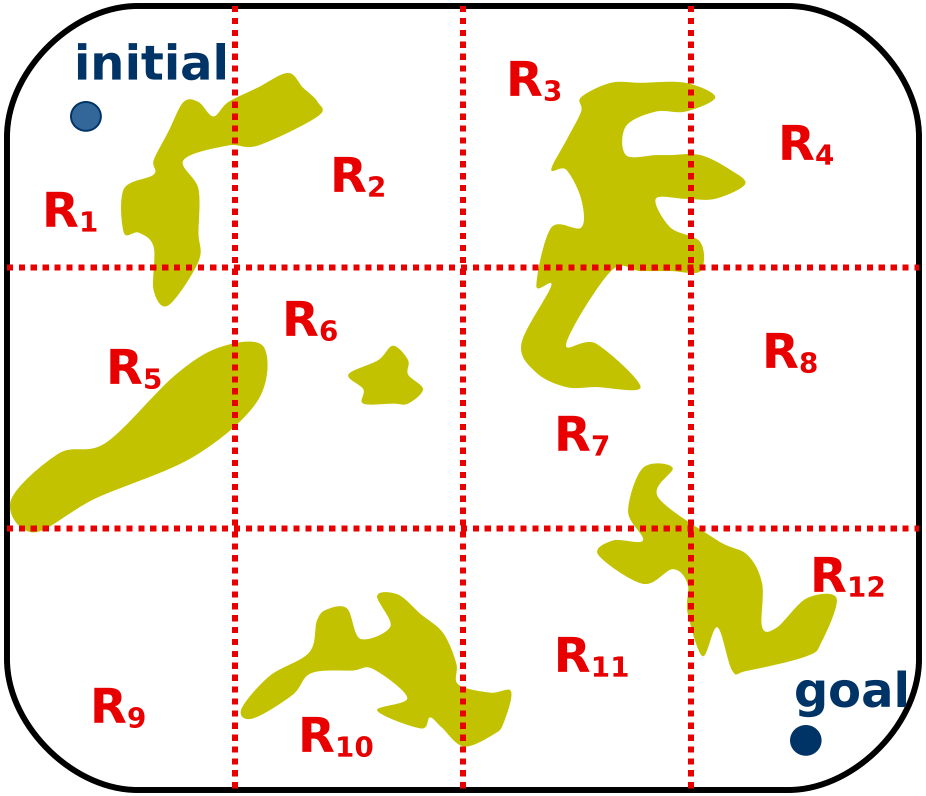
- Motion Planning: Background & Related Work
- **SyCLoP: Synergic Combination of Layers of Planning**
- Applications of SyCLoP to Motion Planning with
  - Dynamics
  - Physics-based Simulations
  - Temporal Objectives
- Discussion

SyCLoP: Synergic Combination of Layers of Planning

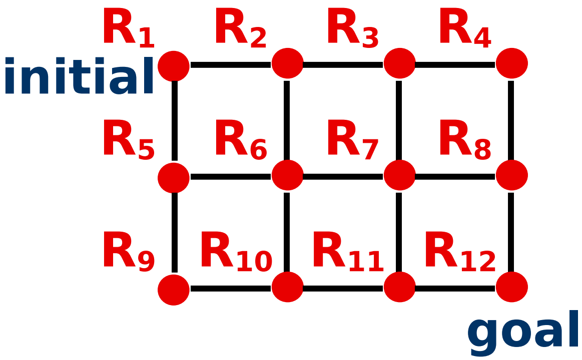


Discrete Model

- provides simplified high-level planning layer



- Decomposition of state space into regions
- Graph encodes adjacency of regions

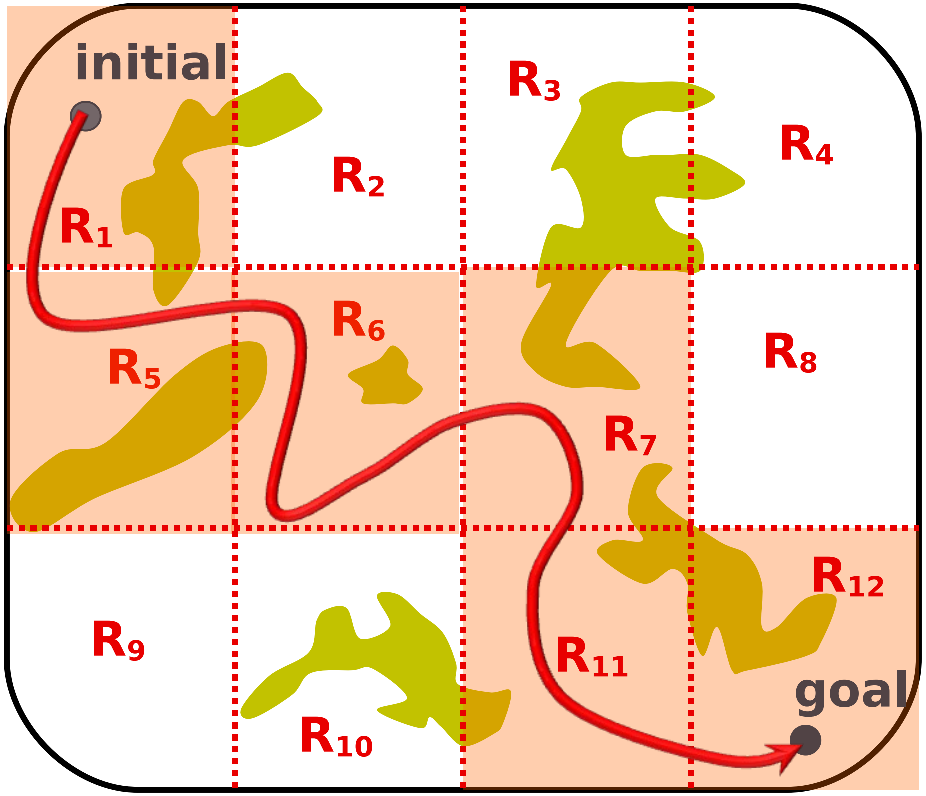


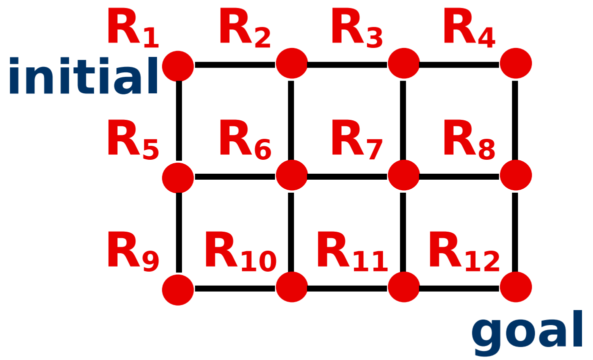
discrete plans:
sequences of regions connecting initial to goal

SyCLoP: Synergic Combination of Layers of Planning

Discrete Plan

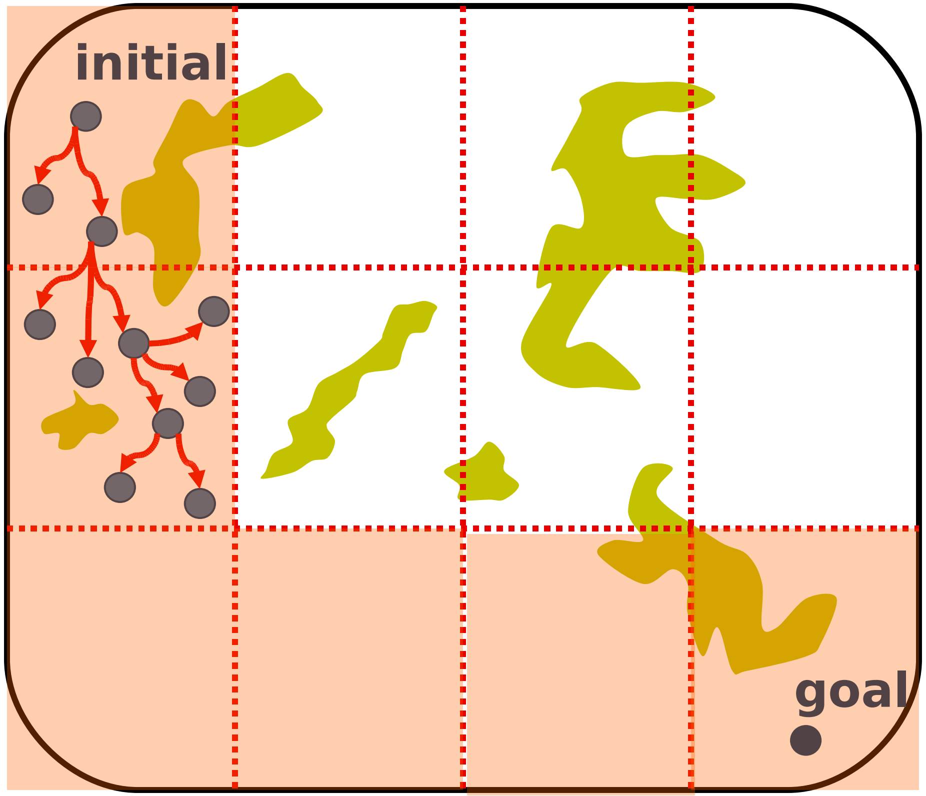
- sequence of regions connecting initial to goal

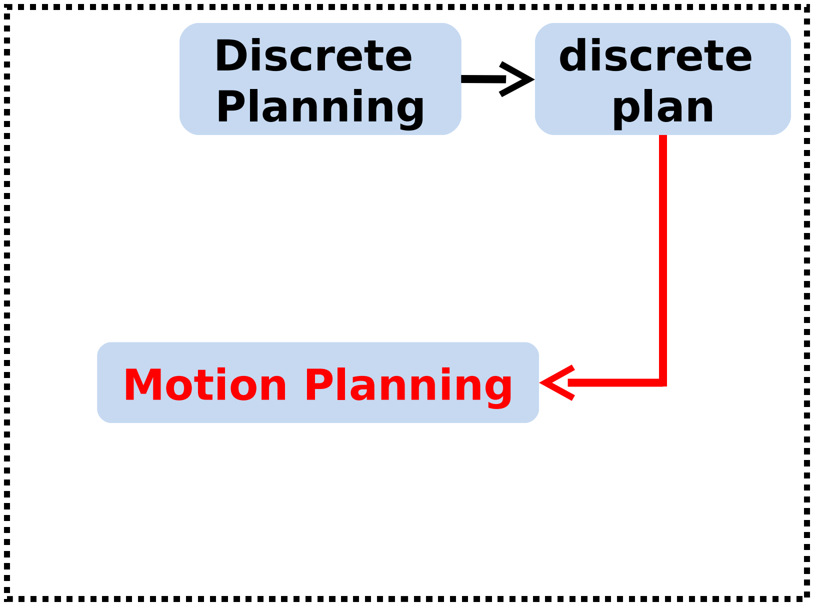




SyCLoP: Synergic Combination of Layers of Planning

Core Loop

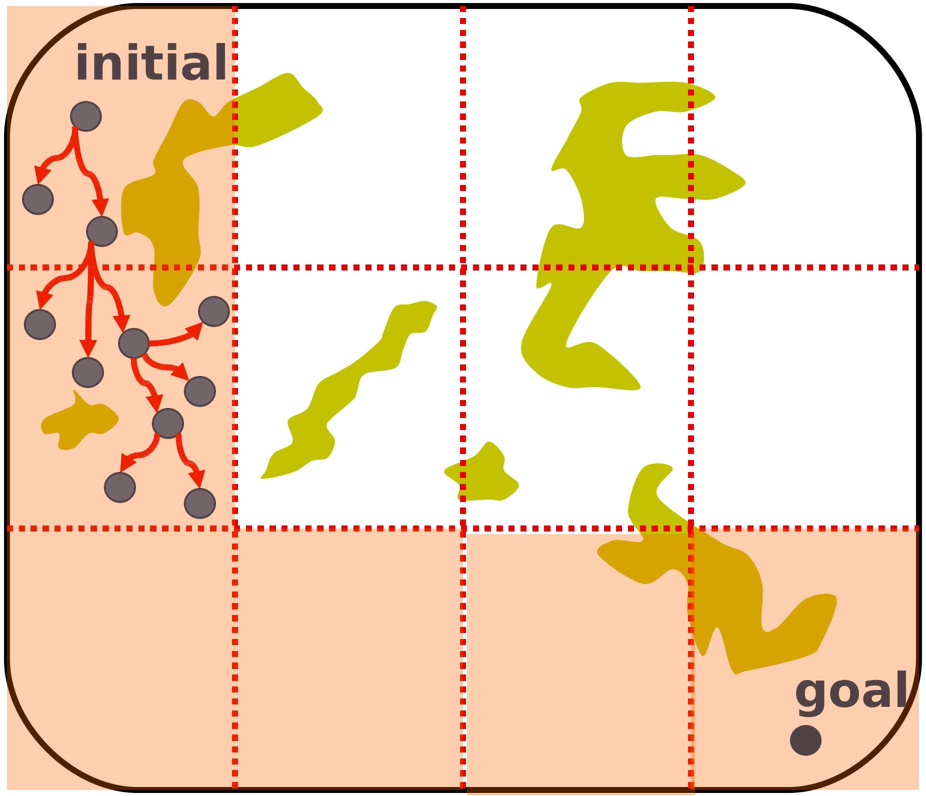


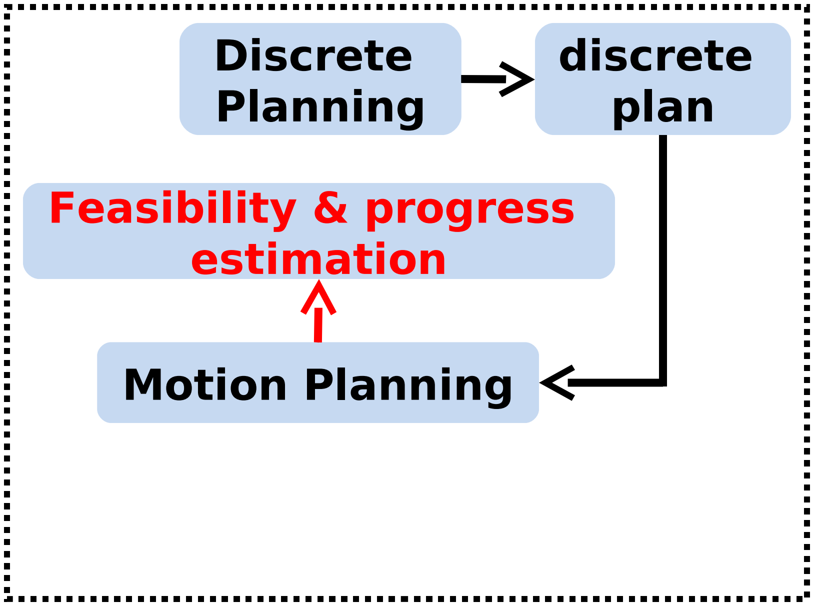


Extend tree branches along regions specified by current discrete plan

SyCLoP: Synergic Combination of Layers of Planning

Core Loop

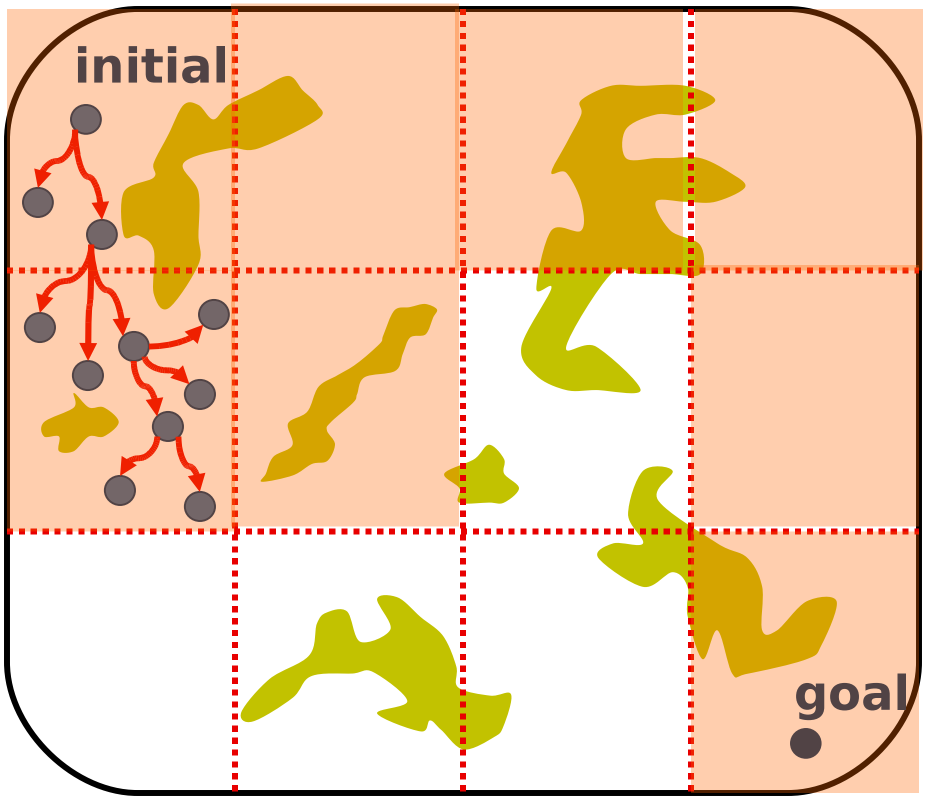


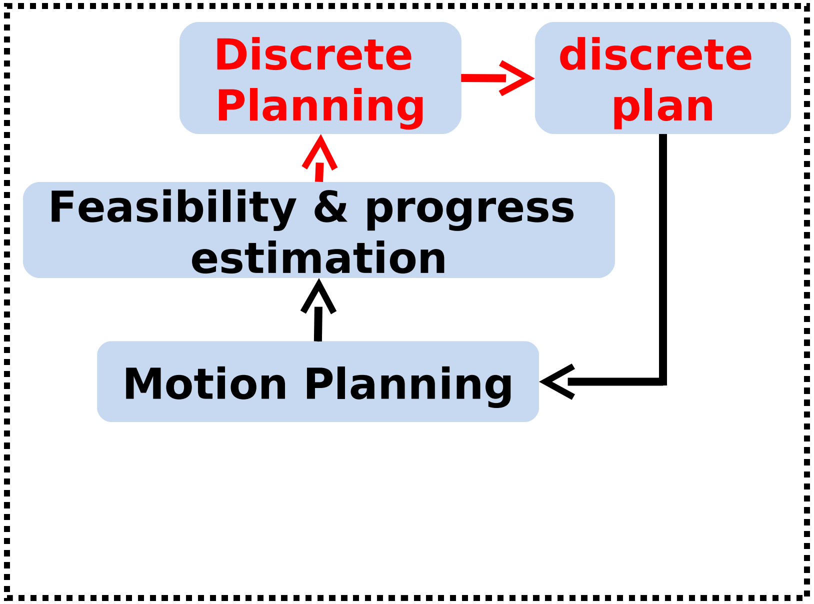


Update feasibility & progress estimation based on information gathered by motion planning

SyCLoP: Synergic Combination of Layers of Planning

Core Loop

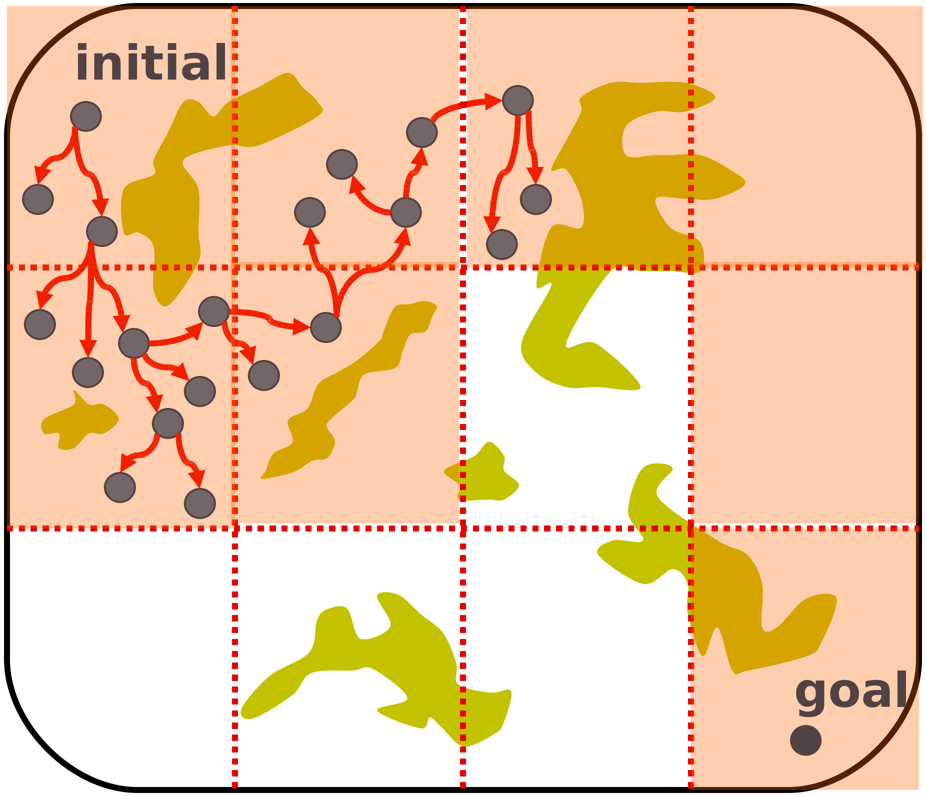


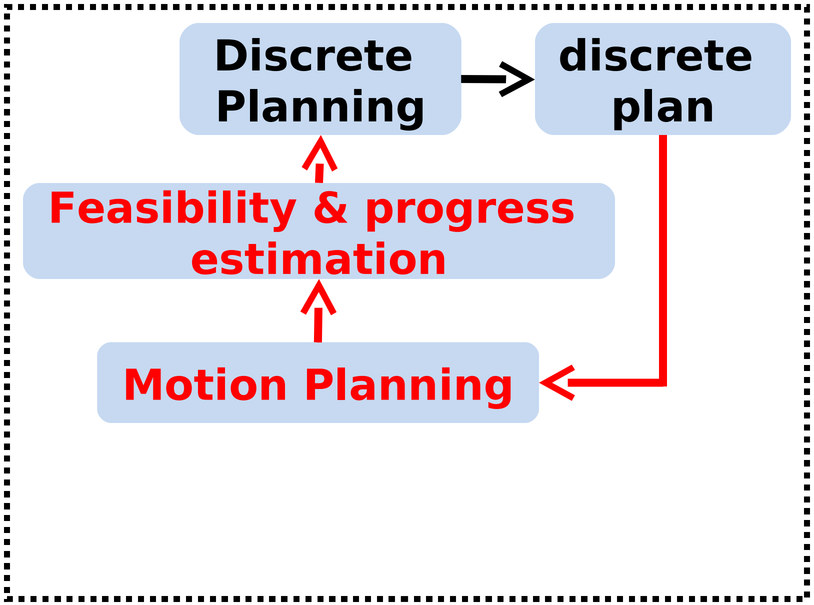


Compute new discrete plan based on updated feasibility/progress estimation

SyCLoP: Synergic Combination of Layers of Planning

Core Loop

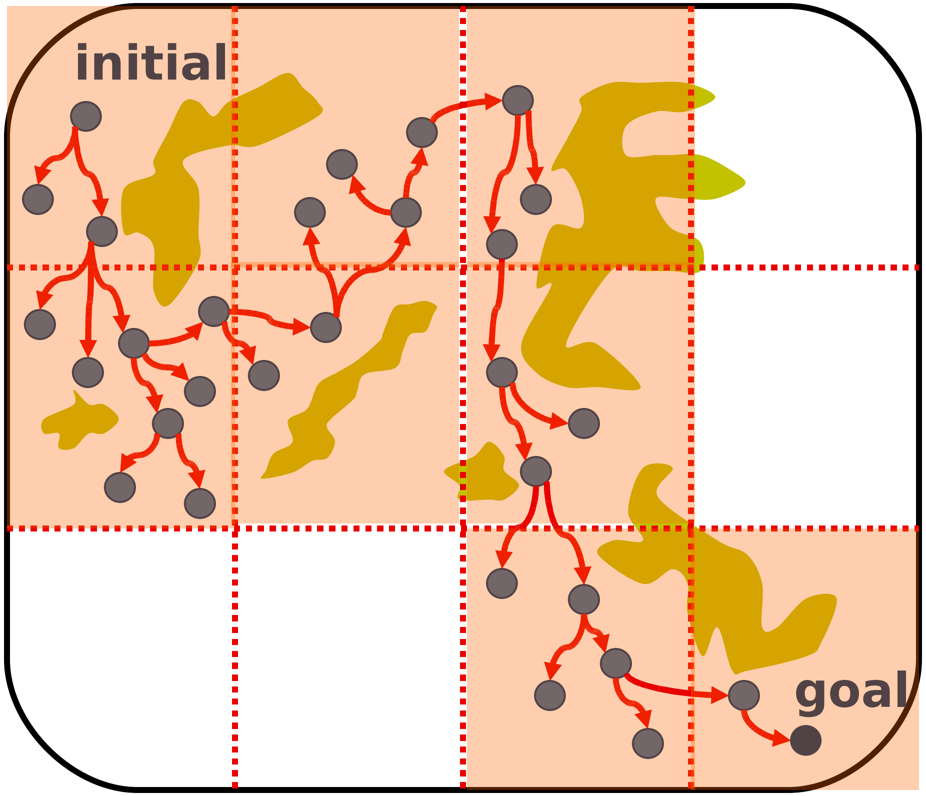


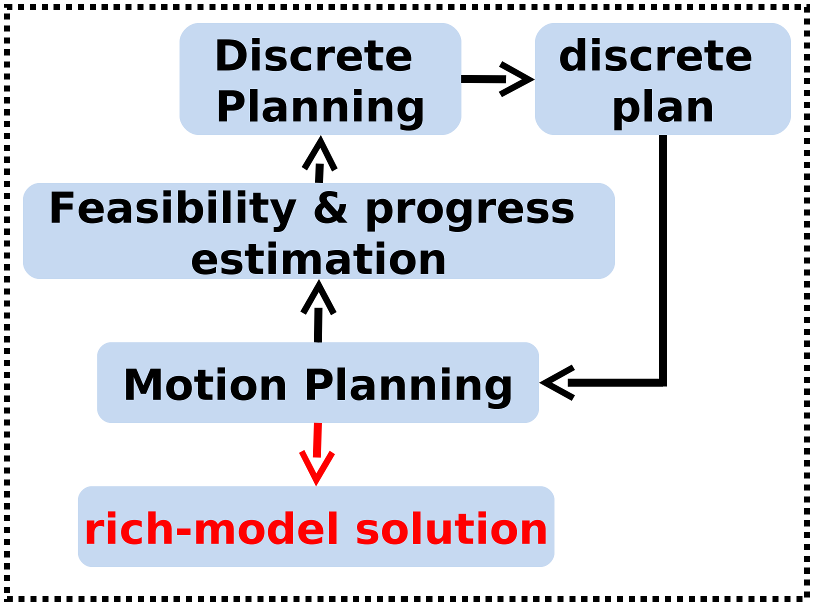


Extend branches along discrete plan & updated feasibility/progress estimation

SyCLoP: Synergic Combination of Layers of Planning

Core Loop



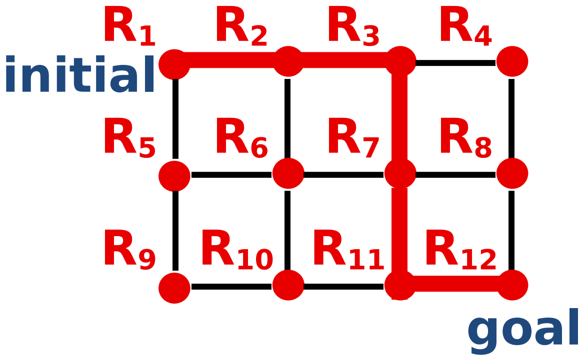


Repeat core loop until the search tree reaches a goal state

SyCLoP: Synergic Combination of Layers of Planning

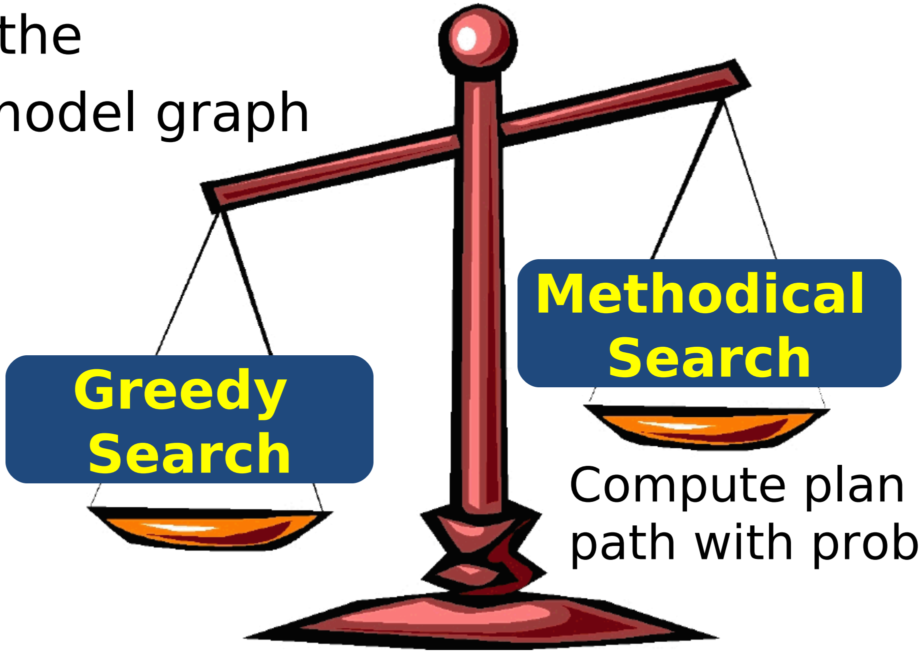
Discrete Planning

- Which discrete plan to select at each iteration?
- Combinatorially many possibilities



- Estimate feasibility of including region R in plan
- Search problem on the weighted discrete-model graph

Compute discrete plan as shortest path with high probability p



Compute plan as random path with probability (1 – p)

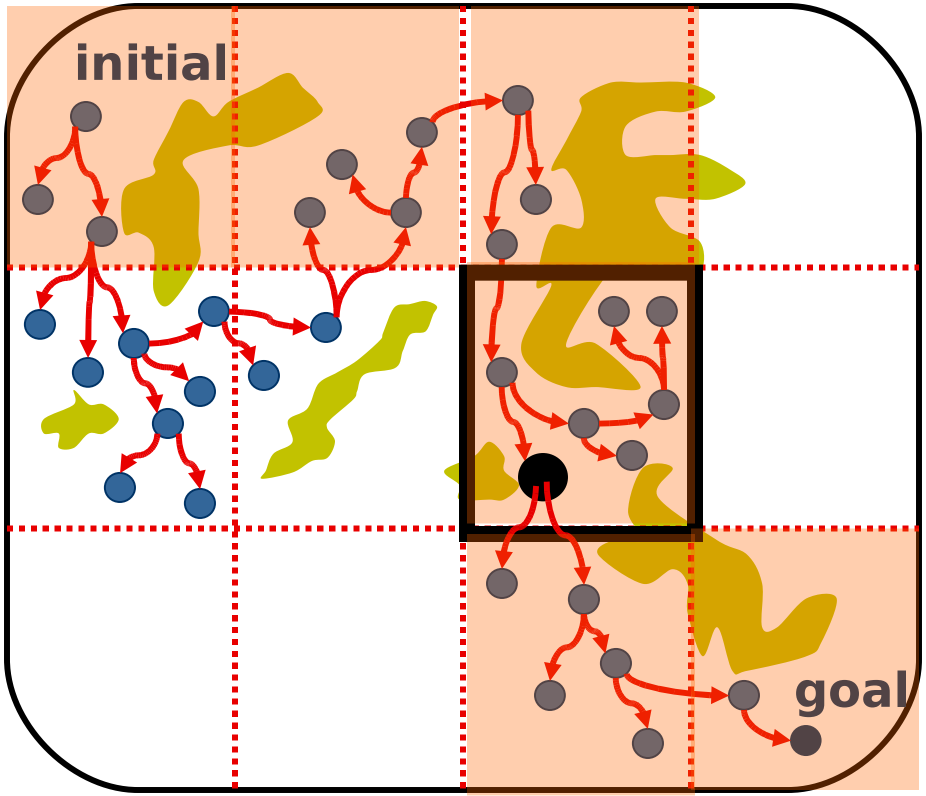
SyCLoP: Synergic Combination of Layers of Planning

Motion Planning

- Discrete plan: $\sigma = R_1, R_2, \ldots, R_n$
- Extend tree along discrete plan

REPEAT FOR A SHORT TIME

- Select region R_i from σ
- Select state s from R_i
- Extend branch from s

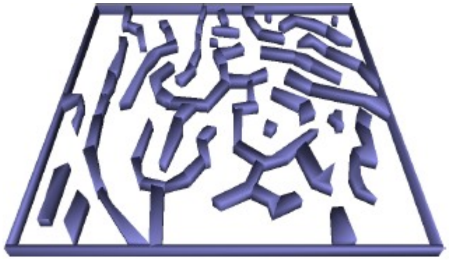


SyCLoP: Synergic Combination of Layers of Planning

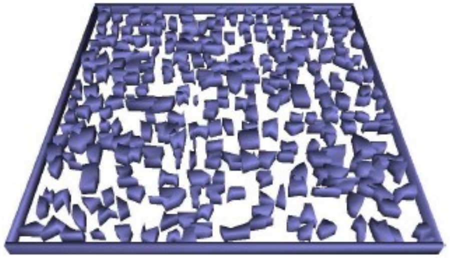
Application: Motion Planning with Dynamics

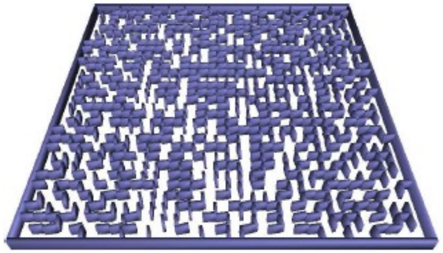
Various workspace environments
- Tens to hundreds of obstacles
- Long narrow corridors
- Random obstacles

Uniform grid-based decomposition


Misc


WindingTunnels


RandomObstacles
278 obstacles


RandomSlantedWalls
890 obstacles

Various robot models
- First-order car
- Second-order car
- Second-order unicycle
- Second-order differential drive

Compared to
- RRT [LaValle, Kuffner '01]
- ADDRRT [Yershova et al., '05]
- EST [Hsu et al., '01]

⇨ same math and utility functions
⇨ same tree data structure
⇨ same control parameters
⇨ same collision detector: PQP
⇨ same hardware

Application: Motion Planning with Dynamics

Second-order dynamics

Car [state = (x, y, θ, v, Φ)]

- u_0, u_1 – acceleration and steering velocity controls
- $x' = v \cos(\theta); y' = v \sin(\theta);$
  $\theta' = v \tan(\Phi) / L; \; v' = u_0; \Phi' = u_1$

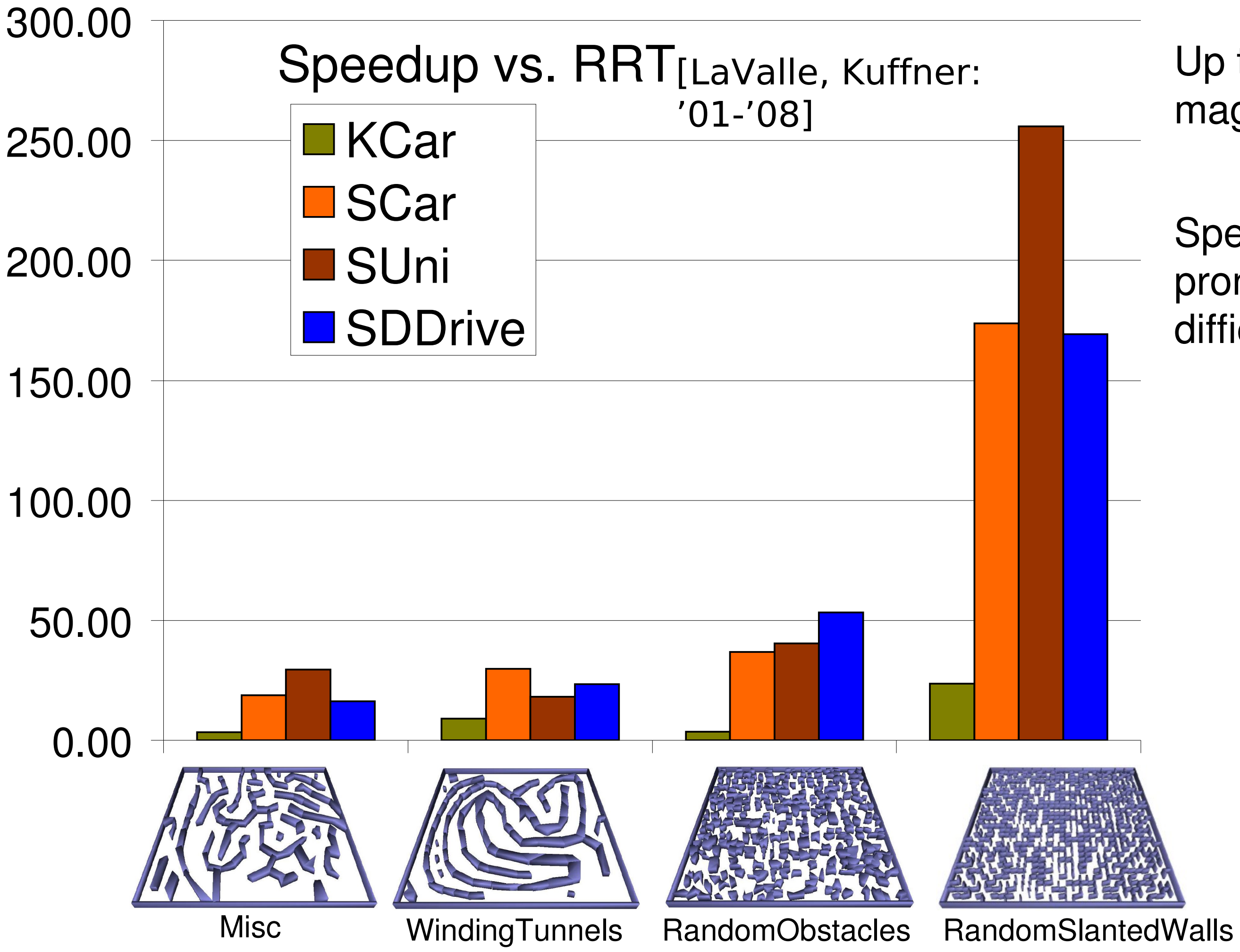
Differential drive [state = (x, y, θ, w_l, w_r)]

- u_0, u_1 – left and right wheel acceleration controls
- $x' = \cos(\theta) r(w_l+w_r)/2; y' = \sin(\theta) r(w_l+w_r)/2;$
  $\theta' = r(w_r-w_l)/L; \; w_l' = u_0; w_r' = u_1$

Unicycle [state = (x, y, θ, v, w)]

- u_0, u_1 – translational and rotational acceleration controls
- $x' = r \; v \cos(\theta); y' = r \, v \sin(\theta);$
  $\theta' = w; \; v' = u_0; w' = u_1$

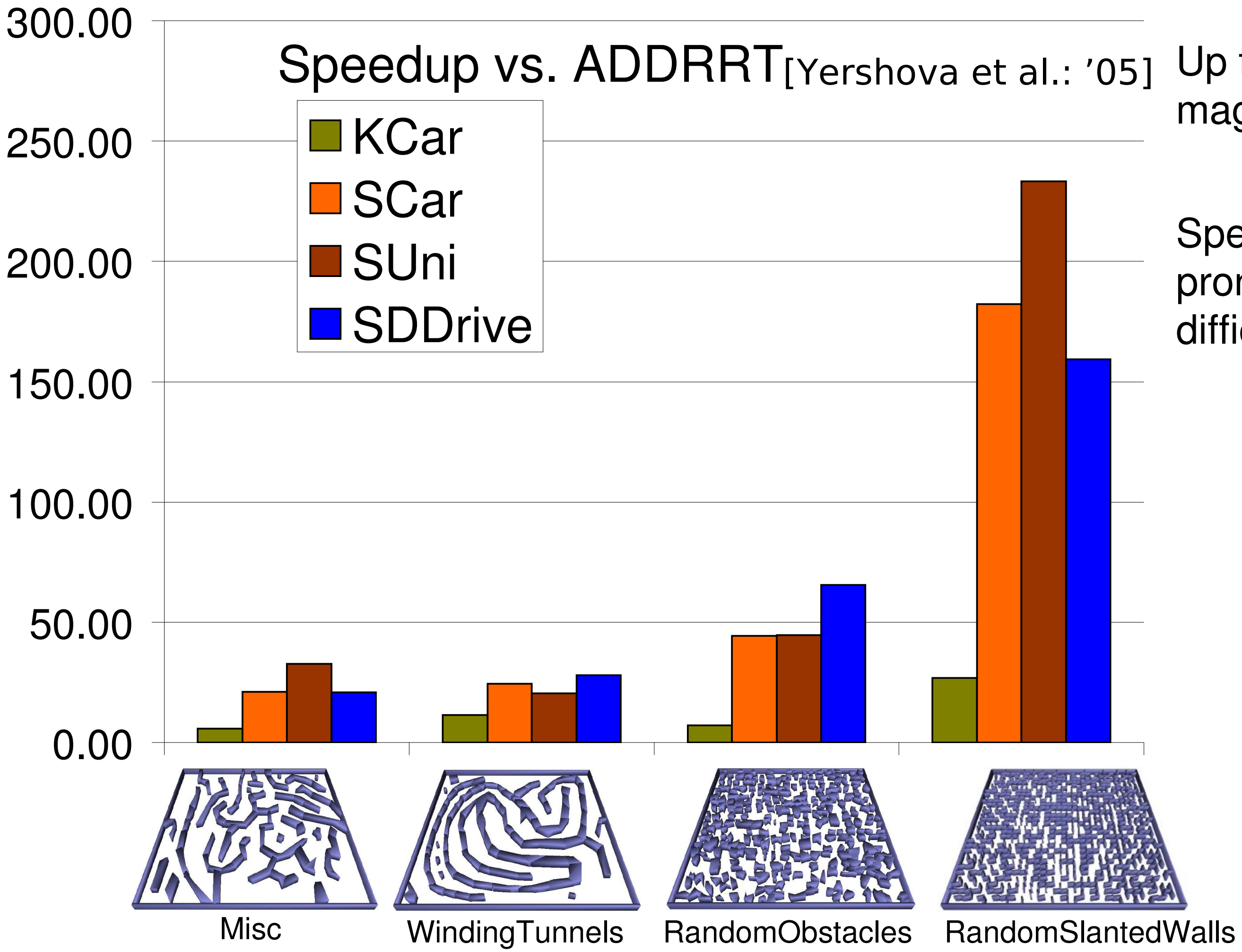
Application: Motion Planning with Dynamics



Up to two orders of magnitude speedup

Speedup becomes more pronounced as problem difficulty increases

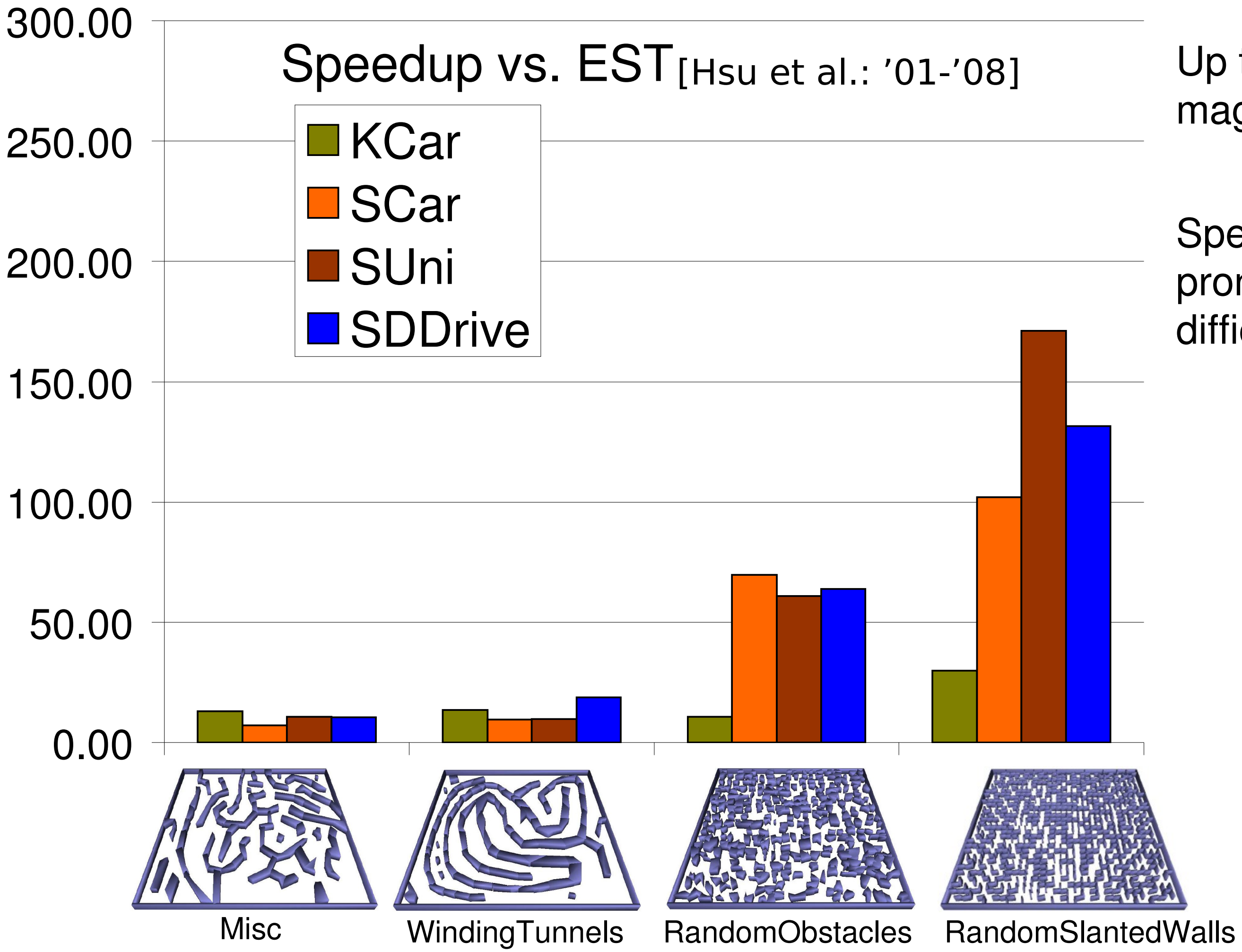
Application: Motion Planning with Dynamics



Up to two orders of magnitude speedup

Speedup becomes more pronounced as problem difficulty increases

Application: Motion Planning with Dynamics



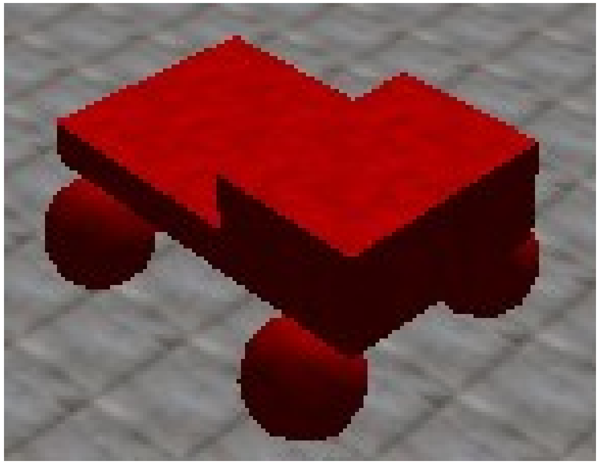
Up to two orders of magnitude speedup

Speedup becomes more pronounced as problem difficulty increases

SyCLoP: Synergic Combination of Layers of Planning

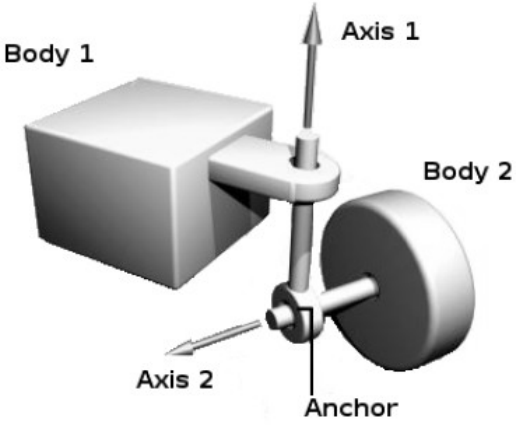
Application: Motion Planning with Physics-based Simulations

- 3D rigid body dynamics
- Wheels form friction contacts
- Torques are bounded



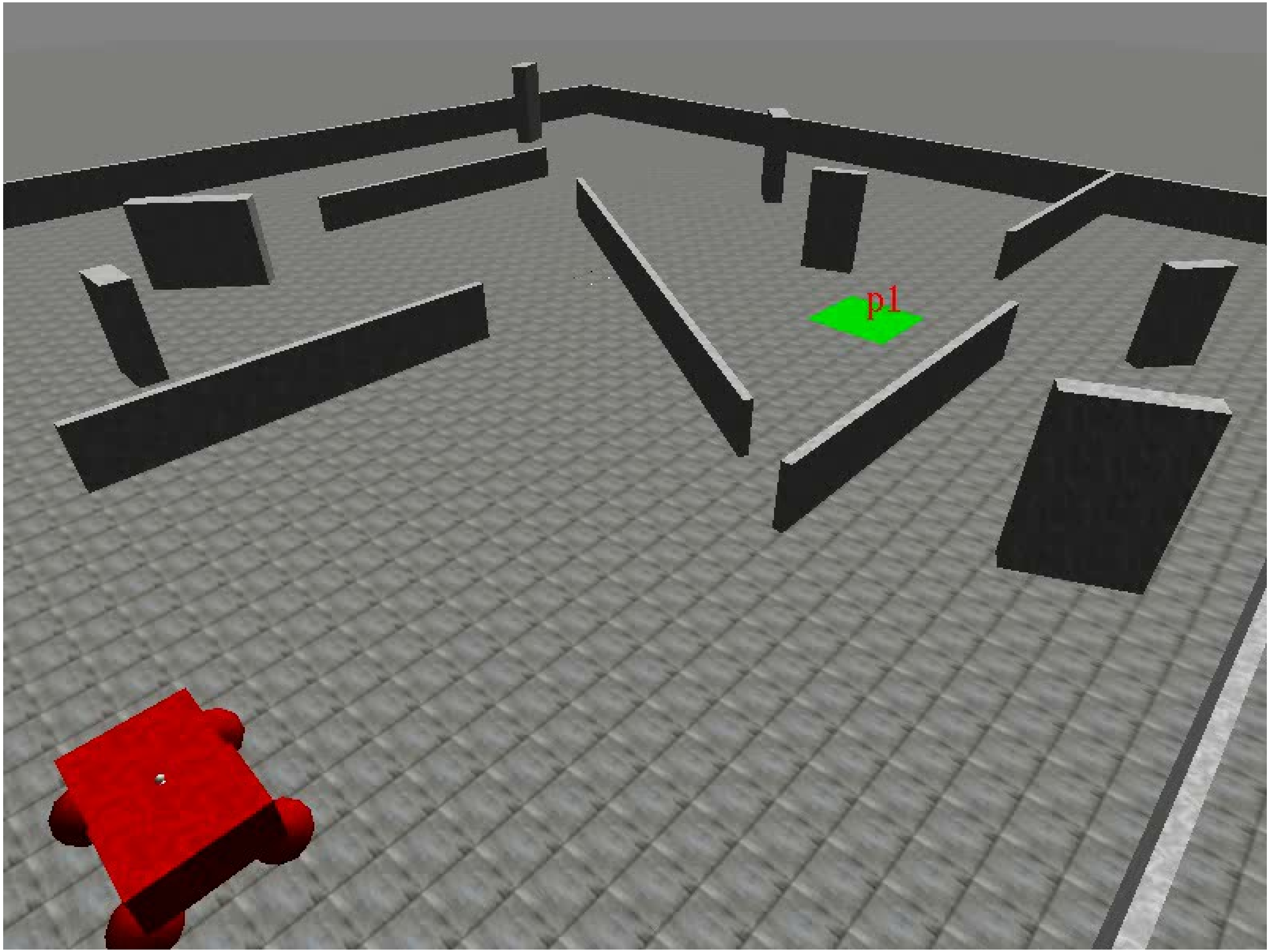
- Open-Dynamics Engine (ODE)
- Stewart-Trinkle model

- Accounts for system dynamics and interactions with the world



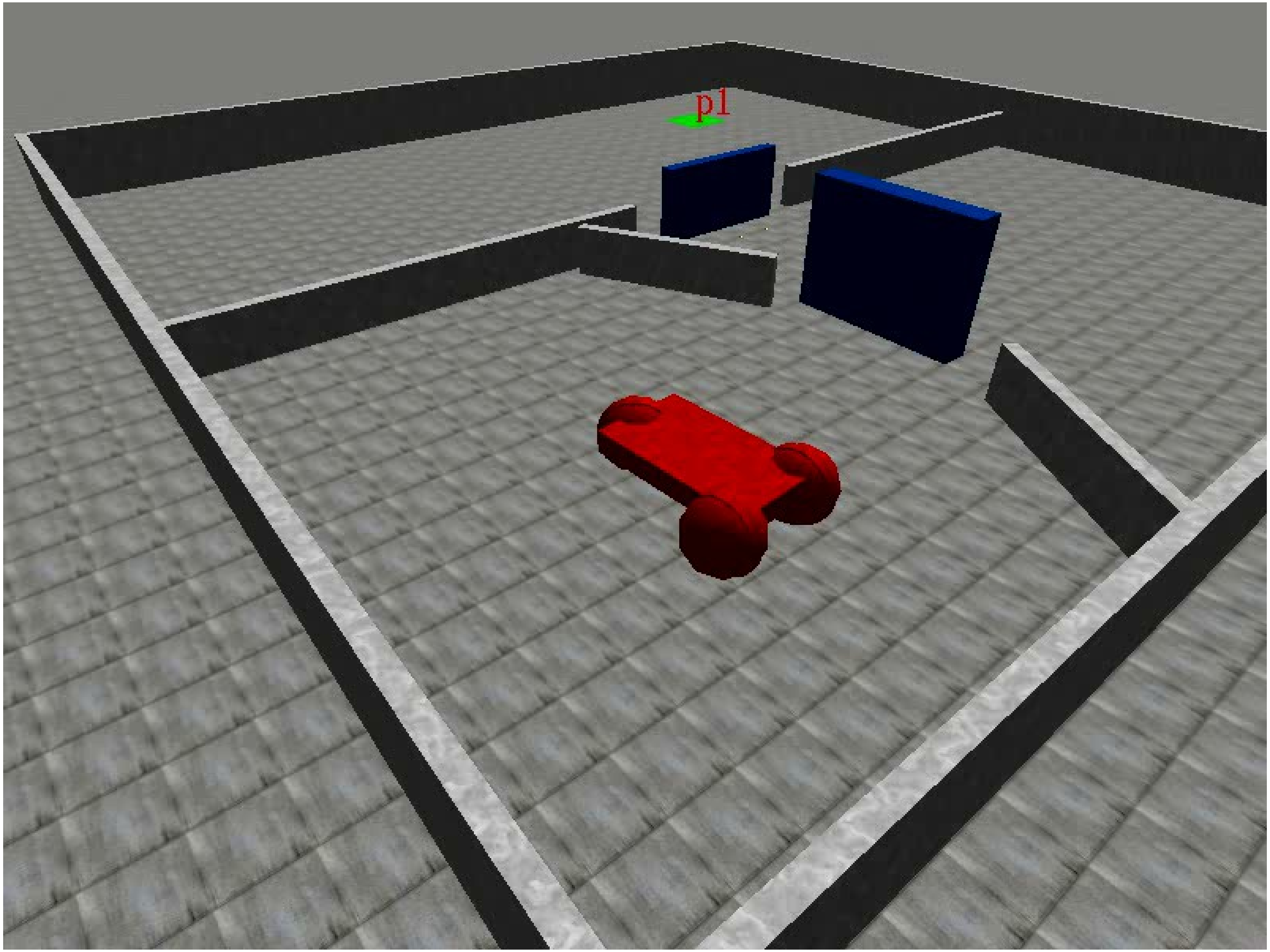
SyCLoP: Synergic Combination of Layers of Planning

Application: Motion Planning with Physics-based Simulations



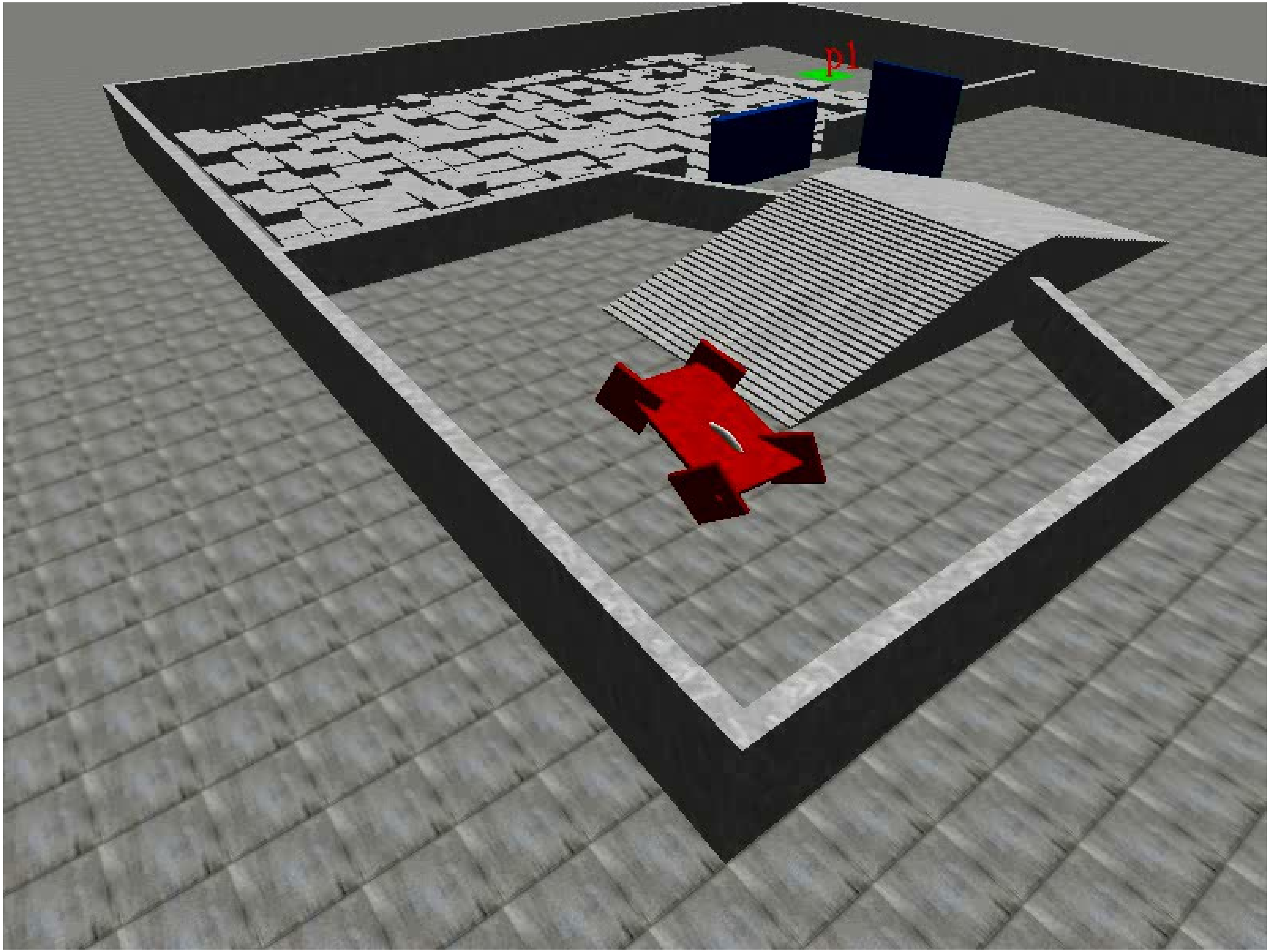
SyCLoP: Synergic Combination of Layers of Planning

Application: Motion Planning with Physics-based Simulations



SyCLoP: Synergic Combination of Layers of Planning

Application: Motion Planning with Physics-based Simulations

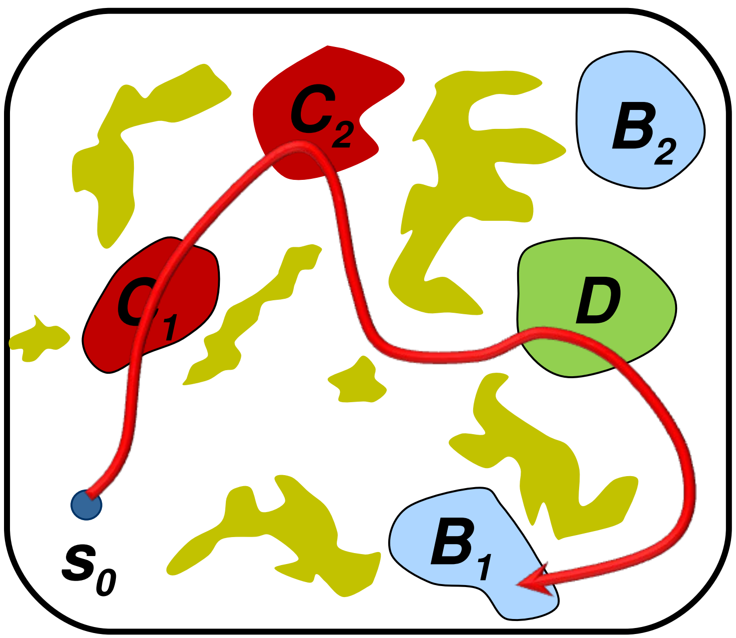


SyCLoP: Synergic Combination of Layers of Planning

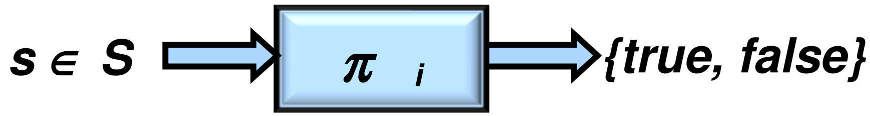
Application: Motion Planning with Linear Temporal Logic

- **Temporal objectives**: reach desired states w.r.t. a linear ordering of time, i.e., “A or B” “A and B” “B after A” “B next to A”

“After inspecting the contaminated areas C_1 and C_2, visit the decontamination station D, and then return to one of the base stations B_1 or B_2”



- Propositions: $\pi_1, \pi_2, \ldots, \pi_n$



- Boolean operators:
  & (and), | (or), ! (not)
- Temporal operators:
  U (until), **G** (always),
  F (eventually), **N** (next)

$\psi = O \; U \; (\psi_1 | \psi_2)$

- $O = !\,(B_1 | B_2 | C_1 | C_2 | D)$
- $\psi_1 = C_1 \,\&\, ((C_1 | O) \; U \; C_2 \,\&\, ((C_2 | O) \; U \; \psi_3))$
- $\psi_2 = C_2 \,\&\, ((C_1 | O) \; U \; C_1 \,\&\, ((C_1 | O) \; U \; \psi_3))$
- $\psi_3 = D \,\&\, ((D | O) \; U \; (B_1 | B_2))$

Summary

SyCLoP

Discrete Planning
- Artificial Intelligence
- Computer Logic

synergic ⟷ **combination**

Motion Planning
- Probabilistic Sampling
- Control Theory

Effective motion planning for:

Tasks
- Reachability
- Temporal objectives

Plaku, Kavraki, Vardi:
TRO05, ICRA07, RSS07
CAV07, ICRA08,
FMSD08 , TACAS09

Rich Models
- Nonlinear Dynamics
- Physical Realism
- Hybrid Systems

OOPSMP www.cs.jhu.edu/~erion/Software.html
- Extensive publicly-available motion-planning package for research or teaching robotics