

Hierarchical Motion Planning

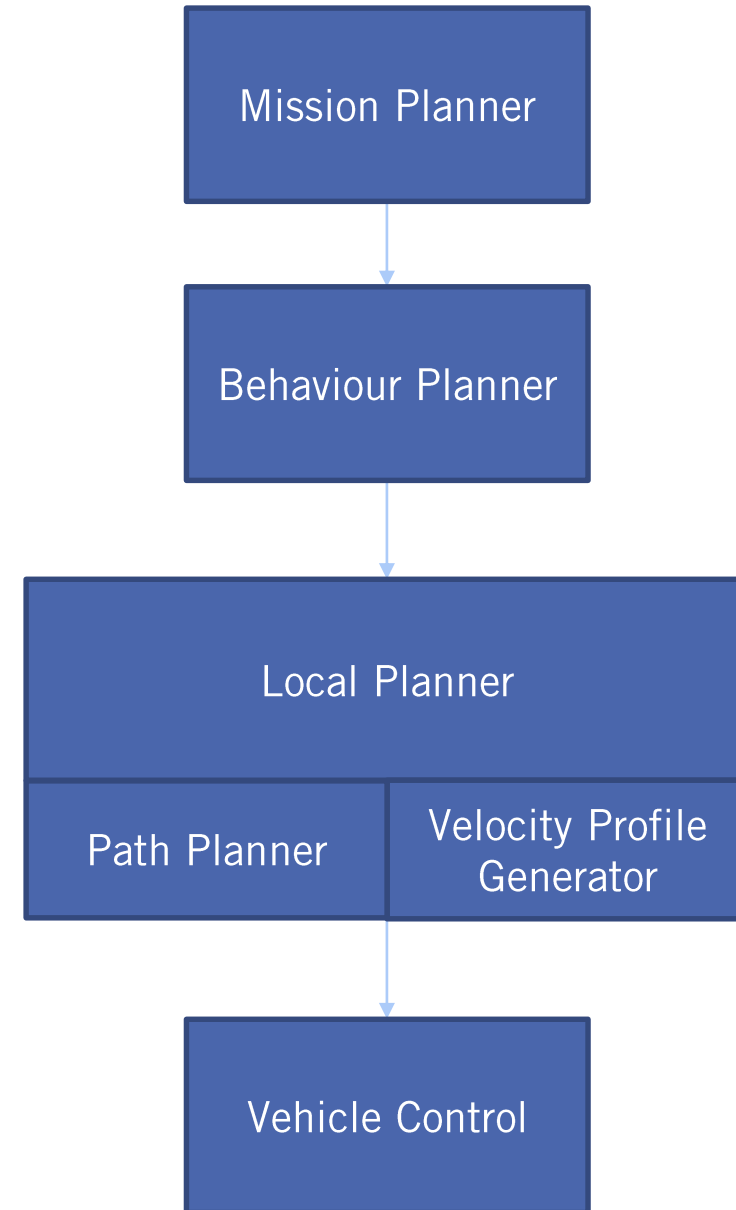
Course 4, Module 1, Lesson 4



UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE & ENGINEERING

Recall: Hierarchical Planner

- Motion planning broken into hierarchy of subproblems
- Mission planner is highest level, focuses on map-level navigation
- Behavioural planner focuses on other agents, rules of the road, driving behaviours
- Local planner focuses on generating feasible, collision-free paths



Mission Planner

map-level

- Highest level planner
- Focuses on autonomous driving mission
 - Navigate to destination at the map level
- Abstract away lower level details
- Can be solved with graph-based methods (Dijkstra's, A*)

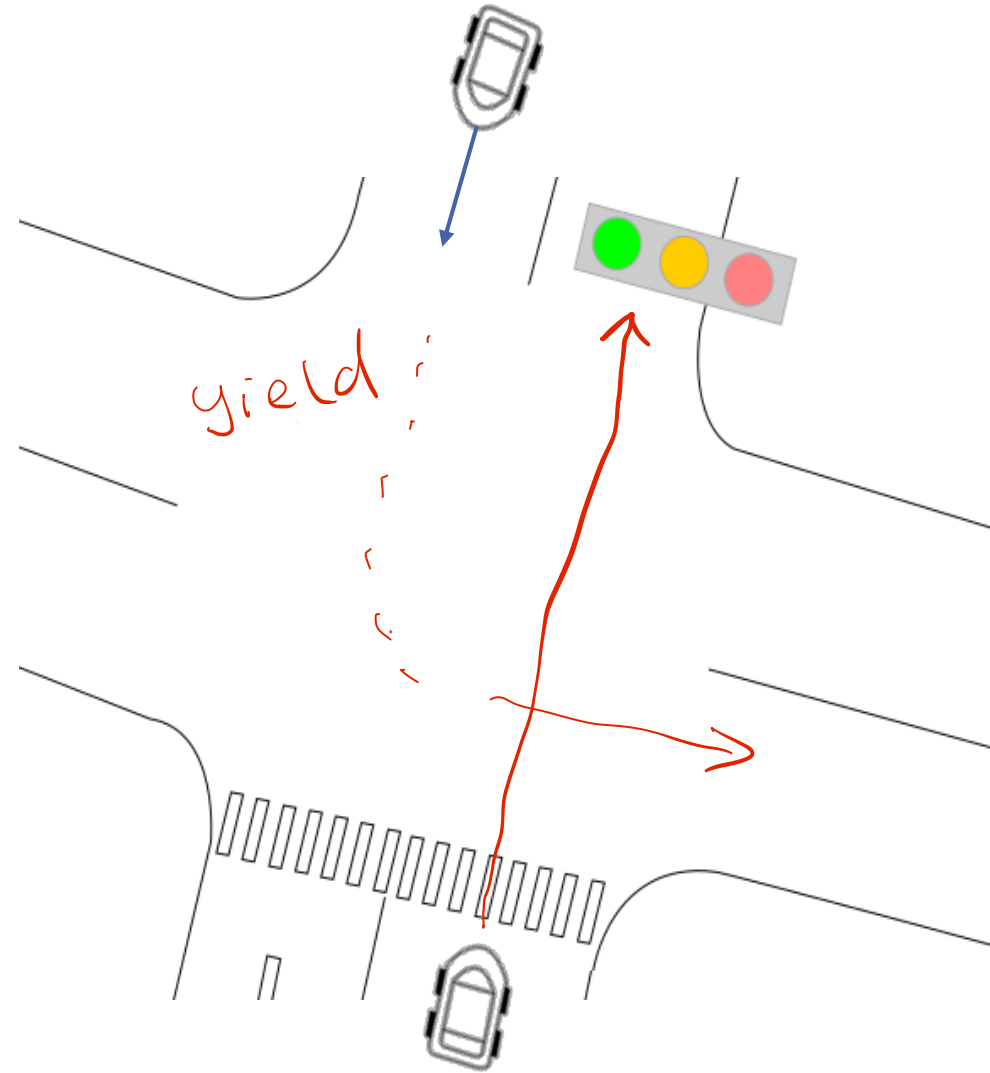
ex. obstacles & regulatory elements

*shortest path
or time (more complex)*



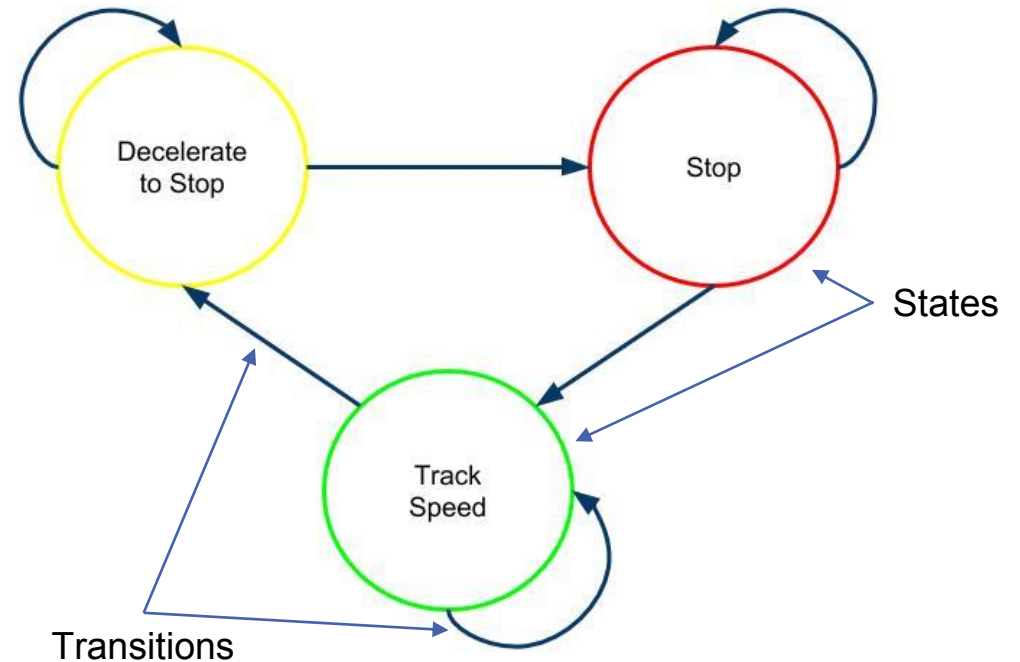
Behavioural Planner

- Behavioural planner decides when it is safe to proceed
- Takes pedestrians, vehicles, cyclists into consideration
- Also looks at regulatory elements, such as traffic lights and stop signs



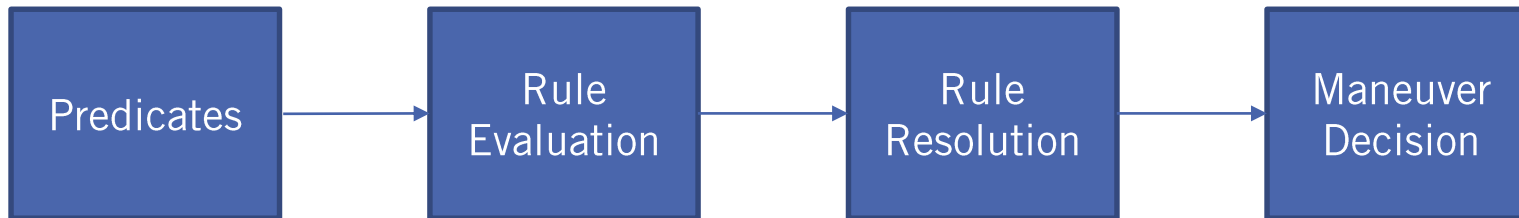
Behavioural Planner - Finite State Machines

- Composed of states and transitions
 - States are based on perception of surroundings
 - Transitions are based on inputs to the driving scenario
 - e.g. stop lights changing colour
- FSM is memoryless
 - Transitions only depend on input and current state, and not on past state sequence



Behavioural Planner - Rule-based System

- Rule-based systems use a hierarchy of rules to determine output behaviour
- Rules are evaluated based on logical predicates
 - Higher priority rules have precedence



- Example scenario with two rules
 - green light \wedge intersection \rightarrow drive straight
 - pedestrian \wedge driving straight \rightarrow emergency stop

higher priority than

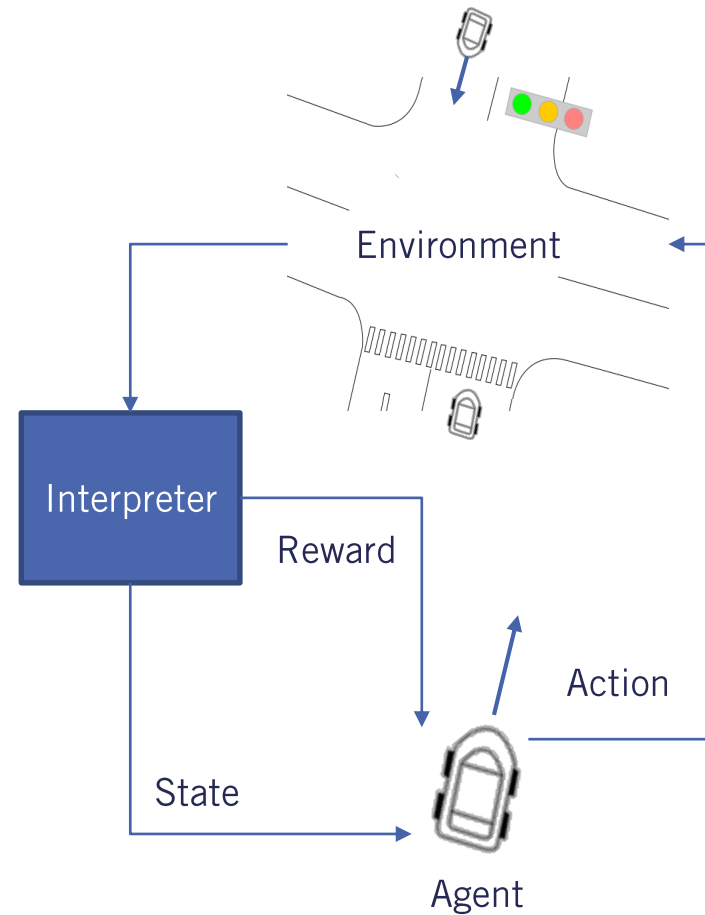
Behavioural Planning - Reinforcement Learning

$$R = \sum_{t=0}^{\infty} \gamma^t R_{a_t}(s_t, s_{t+1})$$

Total Reward

Discount factor $\gamma \in [0,1)$

Stage reward given action, a_t



Local Planner

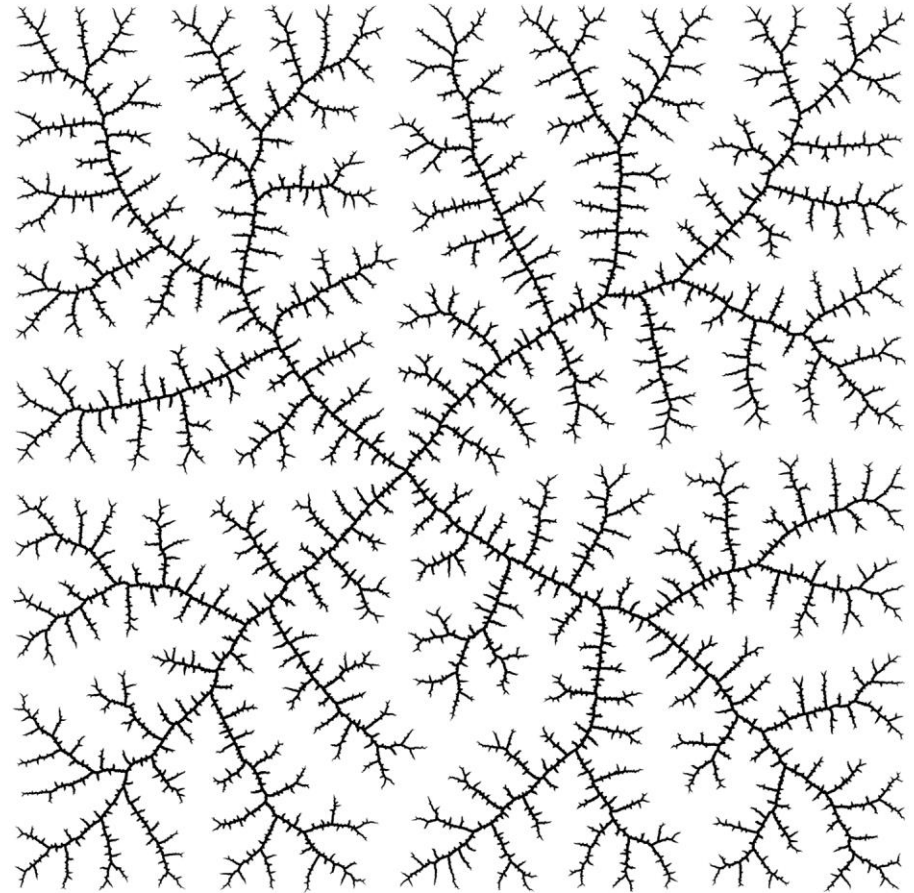
- Local planning generates feasible, collision-free paths and comfortable velocity profiles
- Decomposed into path planning and velocity profile generation



www.shutterstock.com • 648263401

Sampling-based Planners

- Randomly sample the control inputs to quickly explore the workspace
- Collision-checking is performed as new points are added to the explored space
- Often very fast, but can generate poor-quality paths



Rapidly Exploring Random Tree (RRT)

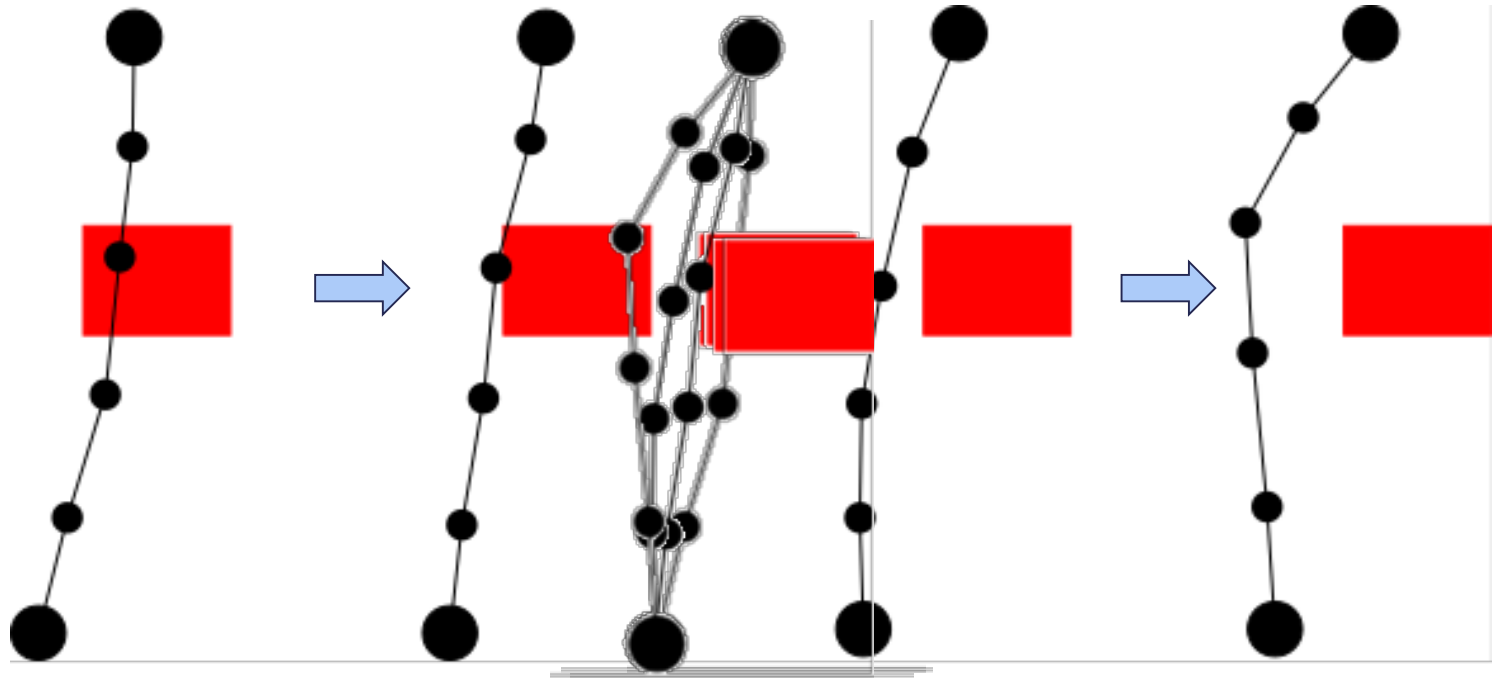
Variational Planners

- Optimize trajectory according to cost functional
 - Contains penalties for collision avoidance and robot dynamics
- Can be slower, and less likely to converge to a feasible solution

$$\min_{\delta x} J(x + \delta x)$$

trajectory planners:
Combine path planning &
velocity planning
into a single step

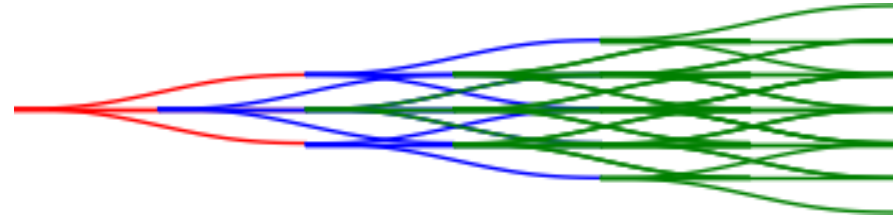
Ex: Chomp algorithms



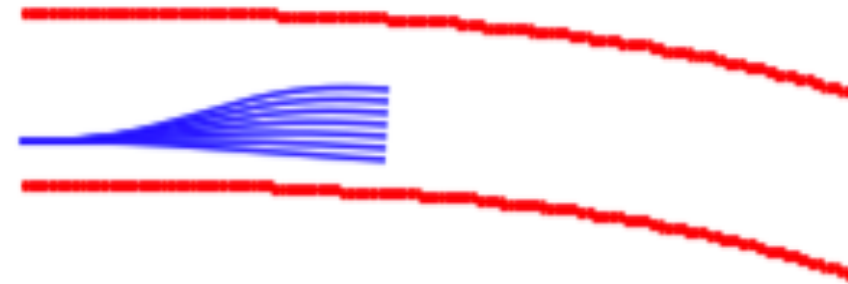
Lattice Planners

- Constrain the search space by limiting actions available to the robot
 - Set of actions known as control set
- Layers of control actions form a graph, which can be searched using Dijkstra's or A*
- Conformal lattice planner fits the control actions to the road structure

Lattice Planner



Conformal Lattice Planner



Velocity Profile Generation

a constrained optimization problem

- Smoothness

$$\int_0^{s_f} \|\ddot{x}(s)\| ds$$

- Deviation from reference

$$\int_0^{s_f} \|v(s) - v_{ref}(s)\| ds$$

- Lateral acceleration limit

$$v^2 \leq \frac{a_{lat_{max}}}{\kappa}$$

