Module 1: The Planning Problem

Lesson 1: Driving Missions, Scenarios, and Behavior

内容
- Understand what the autonomous driving motion planning mission entails.
- Important autonomous driving scenarios based on the road structure and obstacles present.
- Useful driving behaviors to handle autonomous driving scenarios.

Autonomous Driving Mission
- Mission is to navigate from point A to point B on the map.
- Mission planning is higher-level planning.
- Low-level details are abstracted away.
  - such as road structures, obstacles, and other agents on the road.
  - These lower level variables defined different driving scenarios.
- Goal is to find most efficient path (in terms of time or distance).

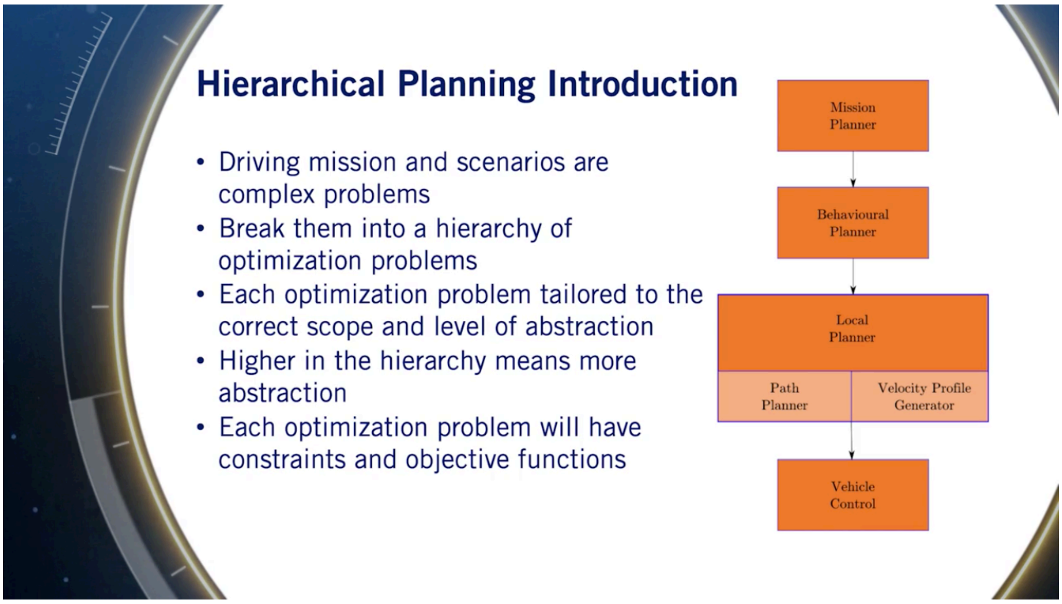
Road Structures Scenarios
- Road structure influences driving scenario through lane boundaries and regulatory elements.
- Simplest case is driving straight, following the center of the lane.
  - Lane Maintenance.
- Minimize deviation from centerline.
- Attain reference speed for efficiency.
  - Reference speed is often the speed limit.
- Lane changes are more complex.
- Different shapes for different situations.
- Shape depends on vehicle speed, acceleration limitations.
- Time horizon of execution affects the aggressiveness of the lane change.
- Left and right turn scenarios are common in intersections and drivelanes（?）.
- Shape of turn varies, similar to lane changes.
- State of surrounding environment impacts the ability of the vehicle to make turns.
- U-turns are useful for efficient direction changes.
- Shape of U-turn will depend on car’s speed and acceleration limits.
- Not always possible at all intersections.

Obstacle Scenarios
- Static and dynamic obstacles also impact the driving scenario.
- Static obstacles restrict which locations our path can occupy.
- Most important dynamic obstacle is often the leading vehicle in front of the ego vehicle.
  - Need to maintain time gap for safety.
  - Time gap is defined as the amount of time until we reach the lead vehicle’s current position while maintaining our current speed.
- Dynamic obstacles impact turns/lane changes as well.
- Depending on locations and speed, different time windows of execution are available for the autonomous vehicle.
- Need to use estimation and prediction to calculate these windows of opportunity.
  - 例えば交差点右折と合流路だ。
- Different dynamic obstacles in the scenario have different characteristics and behaviors.

Behaviors (Not an exhaustive list)
- Speed Tracking.
- Decelerate to Stop.
- Stay Stopped.
- Yield.
  - yield sign. 止まれ!
- Emergency Stop.

Challenges

- Only covered a small subset of scenarios.
  - Focused on common cases that follow the rules of the road.
- Edge cases make the driving task complex.
  - e.g. lane splitting（バイク）, jaywalking.



Hierarchical Planning Introduction
- Mission Planner.
  - Navigate to the destination at the map level.
- Behavioral Planner.
  - Depending on its current driving scenario.
  - つまり行動計画！
  - Useful driving behaviors include speed tracking, deceleration, and yielding.
- Local Planner.
  - Path Planner.
  - Velocity Profile Generator.

Lesson 2: Motion Planning Constraints

These constraints are often crucial for maintaining stability and comfort within the vehicle, as well as maintaining the safety of all agents in a given driving scenario.

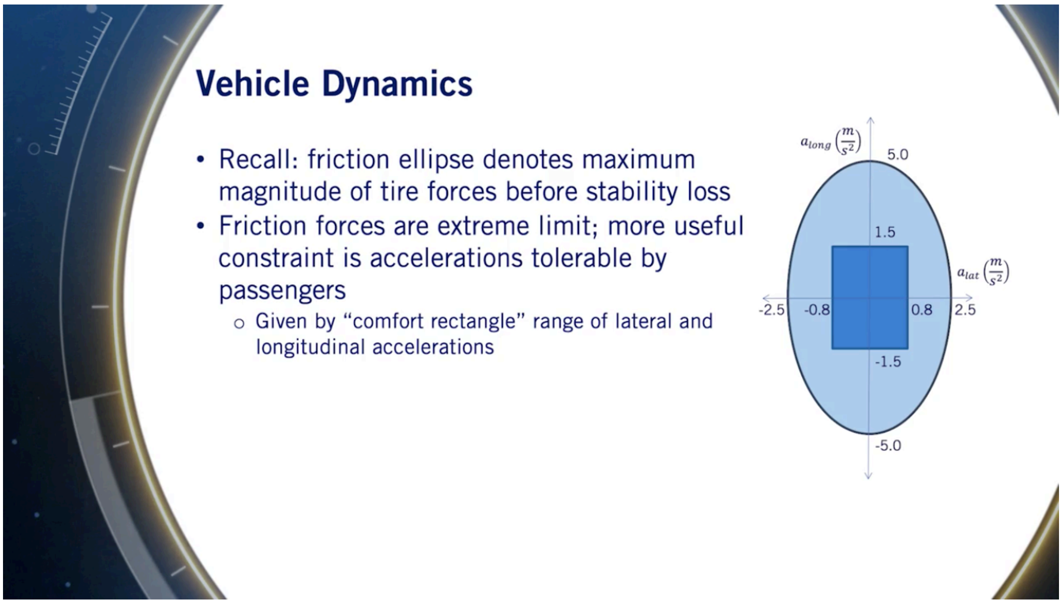
こういう意味のConstraintsだ！

内容
- How the vehicle's kinematics and dynamics constrain the motion planning problem.
- How static and dynamic obstacles affect motion planning.
- The impact of regulatory elements on motion planning.

Bicycle Model
- Kinematics simplified to bicycle model.
  - One reason why this model is chosen is that bicycles have a range of acceptable steering angle values similar to a car.
- Bicycle model imposes curvature constraint on path planning process.
  - $\dot{\theta} = \dfrac{Vtan(\delta)}{L}$.
  - $|\kappa| \leq \kappa_{max}$: curvatureの制限。

- ロジック：steering angle（δ）の制限があるから、$\dot{\theta}$（角速度）の制限もある。角速度の制限があるから、curvature（$R = \dfrac{v}{\omega}$）の制限もある。
- The curvature of any path we planned needs to be mindful of this maximum curvature.
- Curvature constraint is non-holonomic.
  - Intuitively, this means that the constraint doesn't depend only on the state of the robot, but also on how the robot got to its current state.
    - つまり前周期も使う。前周期と現在周期の変化量を制限。
  - Non-holonomic constraints reduce the directions a mobile robot can travel at any point.
  - Makes motion planning challenging.

Curvature
- curvatureの計算は$\kappa = \dfrac{x'y'' - y'x''}{(x'^2 + y'^2)^{\frac{3}{2}}}$。半径のinverse。
  - x and y components of a given path defined w.r.t. arc length. arc lengthって、s coordinateみたい。



Vehicle Dynamics（2種類目の制限源）
- Friction ellipse denotes maximum magnitude of tire forces before stability loss.
- The turning functionality of a vehicle relies on the tires gripping the road surface.
- Friction forces are extreme limit; more useful constraint is accelerations tolerable by passengers.
  - Given by "comfort rectangle" range of lateral and longitudinal accelerations.
- In non-emergency situations, we often focus on the comfort rectangle of accelerations.

Dynamics and Curvature
- From the lateral acceleration constraints, as well as the curvature of the path, we have now indirectly constrained the velocity of the car: $a_{lat} = \dfrac{v^2}{r}$, $a_{lat} \leq a_{lat_{max}}$.
  - Lateral acceleration is a function of instantaneous turning radius of path and velocity.
- Instantaneous curvature is inverse of instantaneous turning radius: $\kappa = \dfrac{1}{r}$.

- Substituting, velocity is constrained by path curvature and lateral acceleration: $v^2 \leq \frac{a_{lat_{max}}}{\kappa}$.
  - κはchanges at each point along the path.
  - Because of this, it's clear that when we are generating a velocity profile for our autonomous vehicle, we have to take the curvature of the path as well as the vehicle's maximum lateral acceleration into consideration.

Static Obstacles
- Static obstacles block portions of workspace.
  - Occupancy grid encoding stores obstacle locations.
- Static obstacle constraints satisfied by performing collision checking.
  - Can check for collisions using the swath of the vehicle's path.
  - Can also check for closest obstacle along ego vehicle's path.

Dynamic Obstacles
- Constraining our motion plan based on the behavior of these other agents will often involve prediction, which is subject to uncertainty.
- However, if we take the conservative approach and constrain ourselves to all possible behaviors of all agents, our motion planning problem quickly becomes over-constrained and impossible to solve meaningfully.
- The degree to which we balance between the safety of conservatism and the aggressiveness required for forward progress, is an active area of autonomous driving research.
- Dynamic obstacles will constrain both our behavior planning process, where we make maneuver decisions, as well as our local planning process, where it will affect our velocity profile planning.

Rules of the Road and Regulatory Elements
- Lane constraints restrict path locations.
- While the rules of the road provides some constraints to the planning problem, they also help us make informed decisions about other agents' behaviors in the environment.
  - For example, oncoming traffic is highly likely to stay in its lane, and not try to collide with our ego vehicle head on.
  - This can help reduce the search space when trying to predict what other agents will do.

Lesson 3: Objective Functions for Autonomous Driving（大事）

In this context, the objective function of our motion planning problem gives us a way of scoring our current motion plan, and allows us to optimize the motion plans such that it has desirable characteristics.

内容
- Some useful objective functions for performing motion planning.
- The benefits and the behaviors that each objective function tries to encourage.

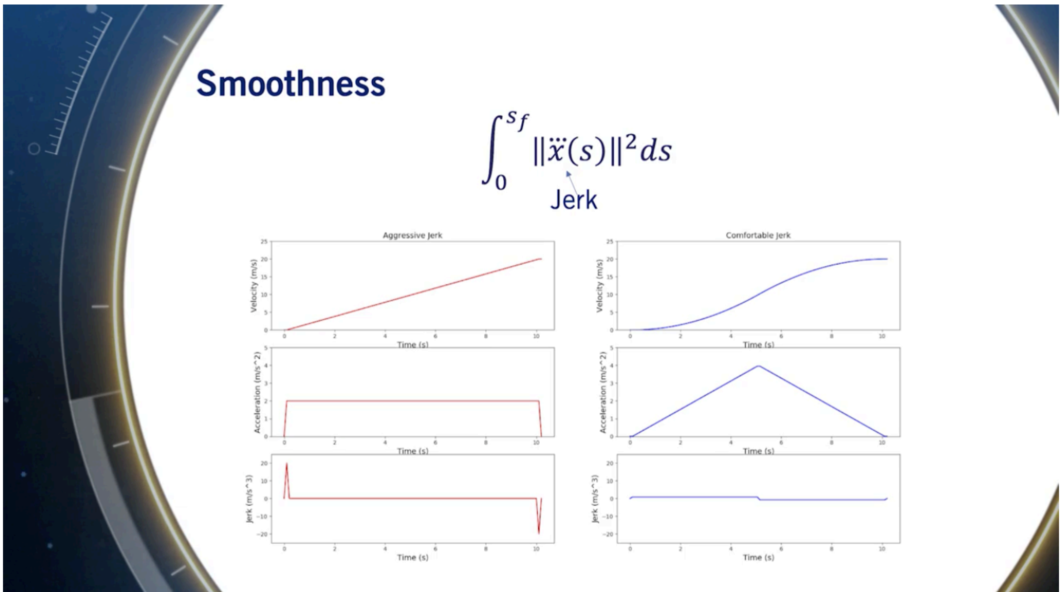
Efficiency
- Path Length:
  - Minimize the arc length of a path to generate the shortest path to the goal:

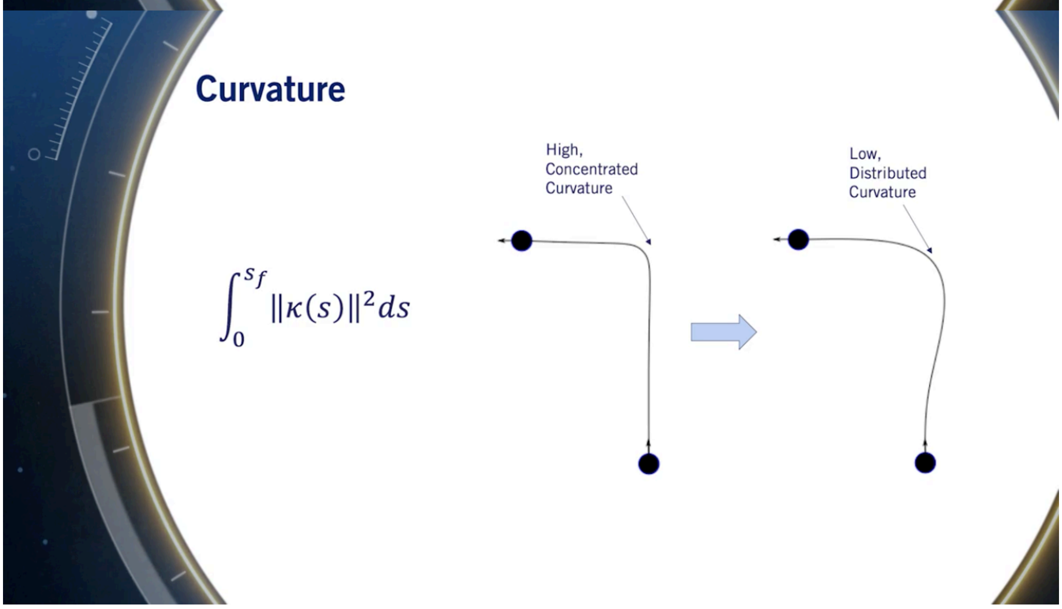
$$s_f = \int_{x_i}^{x_f} \sqrt{1 + \left(\frac{dy}{dx}\right)^2} dx.$$

    - x_i: the starting x coordinate of the path.
    - x_f: the end x coordinate of the path.
- Travel Time:
  - Minimize time to destination while following the planned path: $T_f = \int_0^{s_f} \frac{1}{v(s)} ds$.

Reference Tracking

- In many cases, this will be given to us by the mapping module as the center line of a lane, or the required path for a turn in an intersection.
- Penalize deviation from the reference path or speed profile:
  - $\int_0^{s_f} \|x(s) - x_{ref}(s)\| ds.$
  - $\int_0^{s_f} \|v(s) - v_{ref}(s)\| ds.$
  - Integral of Difference (IOD) term.
  - コース１のFinal Projectの目標じゃないか！
- For Velocity:
  - It is often the case that we want to penalize exceeding speed limits more harshly than staying below the speed limit.
  - Hinge loss to penalize speed limit violations severely: $\int_0^{s_f} \left(v(s) - v_{ref}(s)\right)_+ ds.$
    - Essentially, this term is only active when the velocity profile exceeds the speed limit, that is when the difference between our speed and the speed limit is positive, otherwise this term is zero.





Smoothness：ジャーク
- When optimizing comfort in the objective function of our velocity profile, minimizing the jerk along the trajectory: $\int_0^{s_f} \|\dddot{x}(s)\|^2 ds$.
- The jerk along the car's trajectory greatly impacts the user's comfort while in the car.
- So when planning our velocity profile, we would like to keep the accumulated absolute value of jerk as small as possible.
- 上記の図の例、左：linear ramp profile、右：biquadratic（of or relating to the fourth power）profile。

Curvature
- Bending energy of the path: $\int_0^{s_f} \|\kappa(s)\|^2 ds$. 強いcurvatureを避ける。またcurvatureが強いと、減速もしないといけない。
  - L2 Normなので、1つ強いcurvatureを複数緩いcurvaturesへのdistributionを促す。
- This objective distributes the curvature more evenly along the path, preventing any one along the path from reaching too high a total curvature value.

Lesson 4: Hierarchical Motion Planning

内容
- The scope of the mission, behavior, and local planner sub-problems.
- Methods for solving each sub-problem.
- The reasons for the use of a hierarchical structure for motion planning.

Mission Planner
- Can be solved with graph-based methods (Dijkstra's, A*).

Behavioral Planner
- Behavioral planner decides when it is safe to proceed.

Behavioral Planner - Finite State Machines（Matlabの簡易モデルはこういうタイプでしょう）
- 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 color.
- FSM is memoryless. 状態遷移だから。
  - Transitions only depend on input and current state, and not on past state sequence.

Behavioral Planner - Rule-based System
- Rule-based systems use a hierarchy of rules to determine output behavior.
- Rules are evaluated based on logical predicates.
  - Higher priority rules have precedence.
- Predicates -> Rule Evaluation -> Rule Resolution -> Maneuver Decision.
- Example scenario with two rules.
  - green light $\wedge$ intersection -> drive straight.
  - pedestrian $\wedge$ drive straight -> emergency stop.
- It is both important and challenging to make sure rule-based systems are logically consistent.

Behavioral Planner - Reinforcement Learning.（大事、勉強しよう）
- The reward function values the quality of a given chain of actions for all time steps discounting future states more heavily than the present. $R = \sum_{t=0}^{\infty} \gamma^t R_{a_t}(s_t, s_{t+1})$.

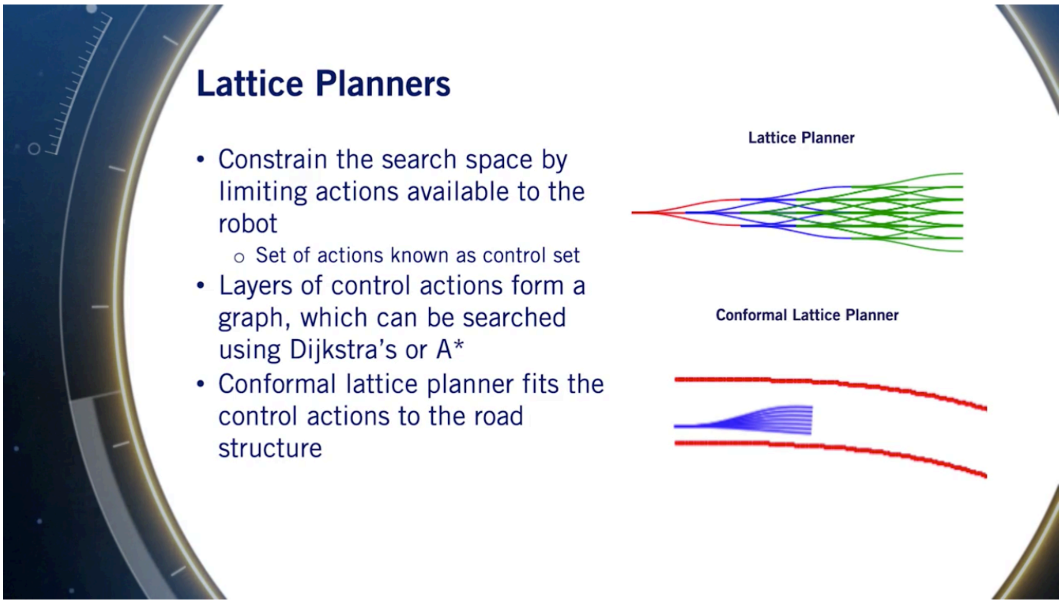
- $\gamma \in [0,1)$: discount factor. discountは重要度を下げる意味でしょう。
- R_{a_t}: stage reward given action, a_t.

- The process of reinforcement learning requires the agent to perform actions in an environment often given by simulation.
- This agent is then rewarded according to its interaction with the environment.
  - Which then allows it to converge to an optimal policy through successive interactions.

Local Planner
- Local planning generates feasible, collision-free paths and comfortable velocity profiles.
- The key ingenuity behind developing a good path planning algorithm is reducing the search space for optimization.
- Three main categories of path planners: sampling-based planners, variational planners and lattice planners.

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.
  - When run in a short number of cycles.
- Rapidly Exploring Random Tree, or RRT and its variance.
  - Construct the branches of the tree of paths by generating points in randomly sampled locations （多分sampled control inputsを使って？） and planning a path to the point, from the nearest point in the tree（直線でしょう？）.
  - If the path is free from collisions with any static obstacles, that path is added to the tree.
  - This tree quickly explores the workspace with many potential paths and when a goal region is reached, the path that terminates in that region is returned.

Variational Planners
- Optimize trajectory according to cost functional.
  - Contains penalties for collision avoidance and robot dynamics: $\min_{\delta x} J(x + \delta x)$.
- Variational planners are usually trajectory planners, which means they combine both path planning and velocity planning into a single-step.
- Can be slower, and less likely to converge to a feasible solution.



Lattice Planners（大事）

- 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*.
- Obstacles can set edges that cross them to infinite cost, so the graph search allows us to perform collision checking as well.
- While the lattice planner is often quite fast, the quality of paths are sensitive to the selected control set.
- Conformal lattice planner fits the control actions to the road structure.
  - Goal points are selected some distance ahead of the car, laterally offset from one another w.r.t. the direction of the road and a path is optimized to each of these points.
  - The path that best satisfies some objective while also remaining collision free is then chosen as the path to execute.
    - objectiveは例えばarc length, bending energy。

Velocity Profile Generation

- Usually set up as a constrained optimization problem.
- objectivesは例えば：
  - Smoothness: $\int_{o}^{s_f} \|\dddot{x}(s)\| ds.$
  - Deviation from reference: $\int_{0}^{s_f} \|v(s) - v_{ref}(s)\| ds.$
  - Lateral acceleration limit: $v^2 \leq \frac{a_{lat_{max}}}{\kappa}.$
- Optimize the velocity profile for a given plan path and reference speed.

- Lattice Planners論文：[2009] Differentially Constrained Mobile Robot Motion Planning in State Lattices.

https://ri.cmu.edu/pub_files/2009/3/ross.pdf