Last Time:

- How to land a space ship

Today:

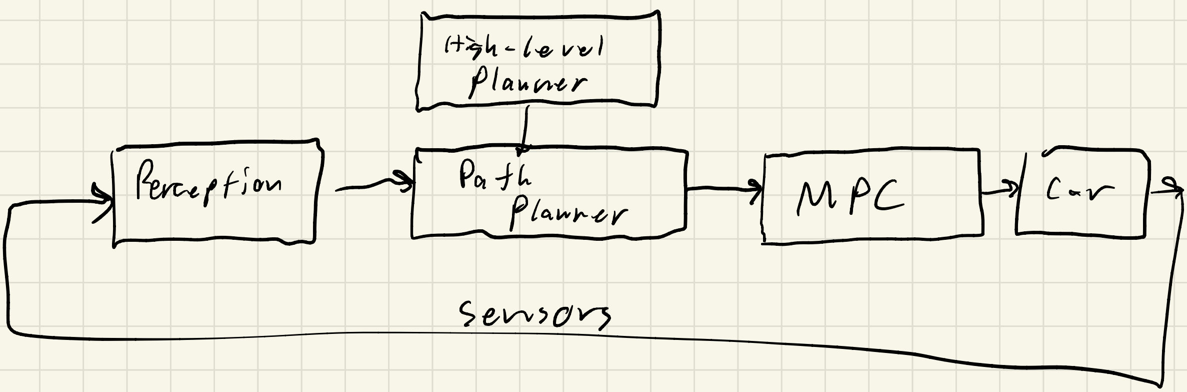
- How to drive a car

---

* History:

- Primitive demos going back to 1920s–30s
- First serious work in 1980s at CMU + Mercedes Benz
- Lots of demos in 1990s with 95%+ autonomy on long trips
- DARPA Grand Challenges in 2000s greatly accelerated progress
- Most current efforts trace back to ~2010

* The "Full Stack"



- Sensors: GPS, IMU, Cameras, LIDAR, RADAR

- Perception is the hard part

- High-level planner: route planning with graph search (generates waypoints)

- Path planner: generate smooth spline curve between waypoints that is collision free. (No dynamics).

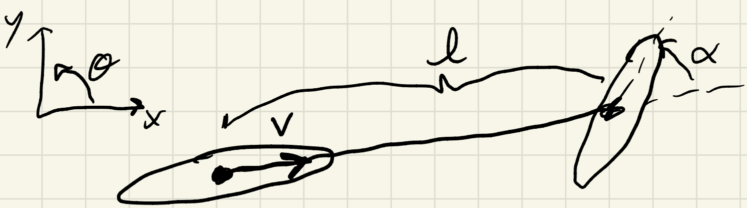
- MPC Controller: Tracks spline curve while reasoning about vehicle dynamics + constraints.

* Vehicle Dynamics:

- Lots of options with different levels of fidelity

- Most common is the "bicycle" or "single-track" model

- "Kinematic" Bicycle Model:



- Assume inputs v, α (or $\dot{\alpha}$)

- Assume tires don't slip

$$\dot{x} = v\cos(\theta)$$
$$\dot{y} = v\sin(\theta)$$
$$\dot{\theta} = \frac{v\tan(\alpha)}{l}$$
$$\dot{\alpha} = u_2$$

$$u = \begin{bmatrix} v \\ \dot{\alpha} \end{bmatrix}$$

- Works well for "normal" driving
- Breaks down in more agressive settings (high acceleration).

- More Complex Models:
  - "Dynamic" Bicycle Model: Model car as a rigid body, $F = ma$, $\tau = J\dot{\omega}$, Reason about engine torque and wheel forces explicitly. Can handle more dynamic behaviors.
  - "Double-Track Model": Model 4 tires, full rigid body dynamics, (possibly) suspension. Reasons about body roll + weight transfer in aggressive cornering. Needed for e.g. off-road, racing, drifting.
- Tire models can go from simple (no-slip, coulomb) to extremely complicated (contact patch, deformation, non-linear stiffness).

- State-of-the-Art Nonlinear MPC
  - Dynamic bycicle model with good tire models gets you really far.
  - Online MPC with IPOPT at ~50 Hz

* The Frozen Robot Problem:

- We want to MPC controller to reason about coupled behavior with other drivers: "If I do this, the other car will probably do that."

- Current systems assume other cars will continue at constant velocity over MPC horizon. MPC gives reactive behavior.

- Works well for highway driving.

- Breaks down in scenarios with tighter coupling between cars. e.g. ramp merging in traffic.

- If there's not a big enough gap between cars, MPC controller will just stop

* Game-Theoretic Trajectory Optimization

- High-level idea: Assume other cars are also solving an optimal control problem like me.

- Solve a joint trajectory optimization problem for all cars simultaneously.

- One version of this idea: "Nash equilibrium"

$$\bar{X} = \begin{bmatrix} X^1 \\ X^1 \\ \vdots \\ X^n \end{bmatrix}, \quad \bar{U} = \begin{bmatrix} U^1 \\ U^2 \\ \vdots \\ U^n \end{bmatrix}$$

← State + Input trajectories for all cars

$$\min_{\bar{X}, u^i} \quad J^i(\bar{X}, u^i) \quad \leftarrow \text{cost for car } i$$

$$\text{s.t.} \quad D(\bar{X}, \bar{U}) = 0$$
$$C(\bar{X}, \bar{U}) \leq 0$$

Dynamics + collision constraints for all cars

- We get n (number of cars) of these problems that we must solve simultaneously
- Interpretation: No player can unilaterally improve their cost.
- Good model of non-cooperative driving behavior
- Cost functions can capture driver agressiveness etc.

* Solution Strategy (ALGAMES)

- Form Augmented Lagrangian for each player and 1st-order necessary conditions:

$$L^i(\bar{X}, \bar{U}, \lambda^i, \mu^i) = J^i(\bar{X}, u^i) + \frac{\rho}{2}\|D(\bar{X}, \bar{U})\|_2^2 + \lambda^{i^T} D(\bar{X}, \bar{U})$$
$$+ \frac{\rho}{2}\|C(\bar{X}, \bar{U})^+\|_2^2 + \mu^{i^T} C(\bar{X}, \bar{U})$$

$$\Rightarrow \quad \nabla_{u^i} L^i = 0$$

- Stack FON conditions for all players:

$$\begin{bmatrix} \nabla_{u^1} L^1 \\ \nabla_{u^2} L^2 \\ \vdots \\ \nabla_{u^n} L^n \end{bmatrix} = 0$$

- Solve with Newton's method
- Apply standard AL update to estimate λ, μ
- Generates human-like interaction strategies
- By taking advantage of sparsity structure in Newton solver, can solve at 30-50Hz 4-5 cars.