Last Time:

- Walking
- MPC control for quadrupeds

Today:

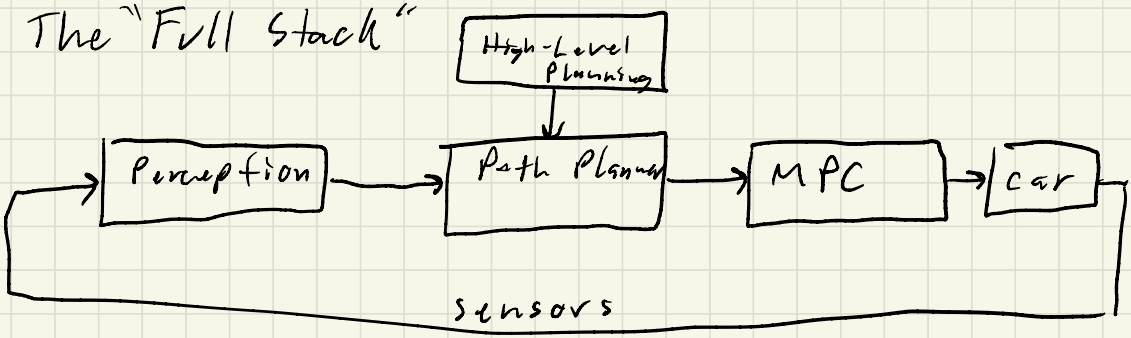
- How to drive a car
- Game-theoretic MPC

---

* History

- Primitive demos going back to 1920s-30s
- First serious modern work in 1980s at CMU and Mercedes-Benz
- Lots of demos in 1990s with 95%+ autonomy on long highway trips
- DARPA Grand Challenge in early 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 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

$$\dot{x} = v\cos(\theta) \qquad x = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \qquad u = \begin{bmatrix} v \\ \dot{\alpha} \end{bmatrix}$$
$$\dot{y} = v\sin(\theta)$$
$$\dot{\theta} = \frac{v\tan(\alpha)}{l}$$
$$\dot{\alpha} = u_2$$

- Works well for "normal" driving
- Breaks down in more aggressive settings (high acceleration).
- More Complex Models:
  - "Dynamic" Bicycle Model: Model car as a rigid body. $F=ma$, $\tau = J\dot{\omega}$, Reason engine torque and tire forces explicitly. Can handle more dynamic behaviors.
  - "Double-Track" Model: Model 4 tires, full 3D rigid body dynamics, suspension, etc. Reason about body roll + weight transfer in agressive cornering. Needed for e.g. off-road, racing, drifting.
  - Tire models can go from simple (no-slip) to medium complexity (Coulomb), to very complicated (contact patch, deformation, nonlinear stiffness, etc.)
- ★ State-of-the-Art Nonlinear MPC
  - Dynamic bicycle model with good tire models gets you really far.
  - Online MPC with IPOPT at ~50Hz

* The Frozen Robot Problem:

- We want the 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.

- Works well for highway driving

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

- 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"

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

← Stacked inputs for all cars stacked

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

$$\text{s.t.} \quad \left.\begin{aligned} D(\bar{x}, \bar{u}) &= 0 \\ C(\bar{x}, \bar{u}) &\leq 0 \end{aligned}\right\} \text{dynamics + collision constraints for all cars}$$

- 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 aggressiveness etc.

* Solution strategy (ALGAMES)

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

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

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

- 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 in Newton solver, can solve at 30~50 Hz for 4~5 cars