Lecture 8: Control I

Professor Katie Driggs-Campbell

February 8, 2024

ECE484: Principles of Safe Autonomy

Administrivia

- Field trips next week!
  - 2/13 Field trip 1 to high bay to see the GEM
  - 2/15 Field trip 2 to see F1tenth cars
  - 2/20 Simulation project walkthrough
- No office hours next week (Tues. 2/13)
- MP1 released Friday 2/9
- Upcoming due dates:
  - HW0 and MP0 due Friday 2/9
    - "Demos" in lab sections
  - HW1 and MP1 due Friday 2/23

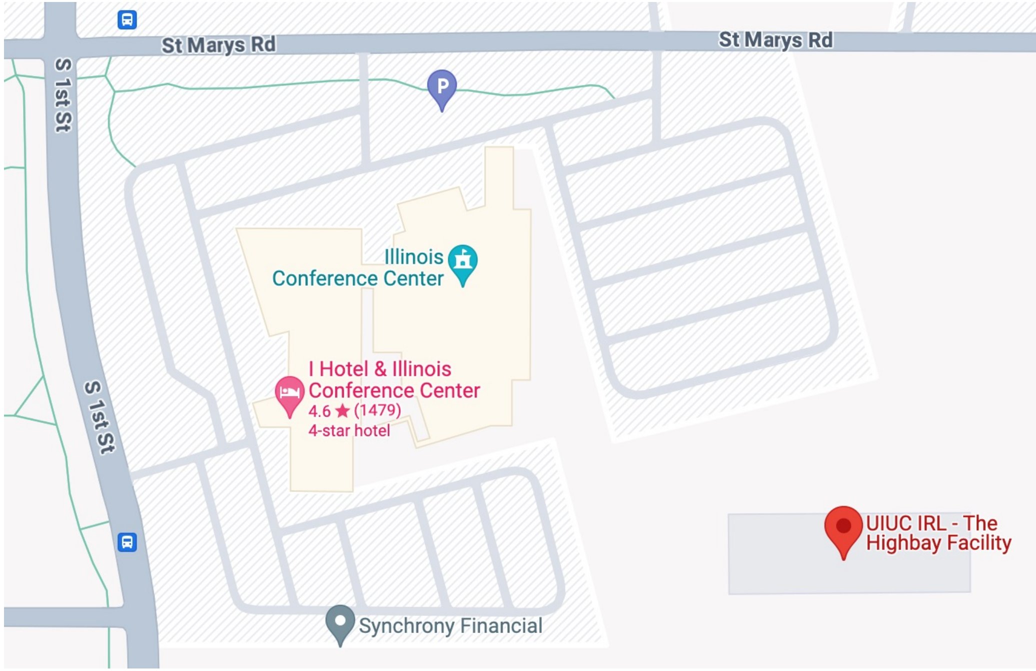
Field Trip to High Bay

Address: 201 St Mary's Rd (near I-Hotel)

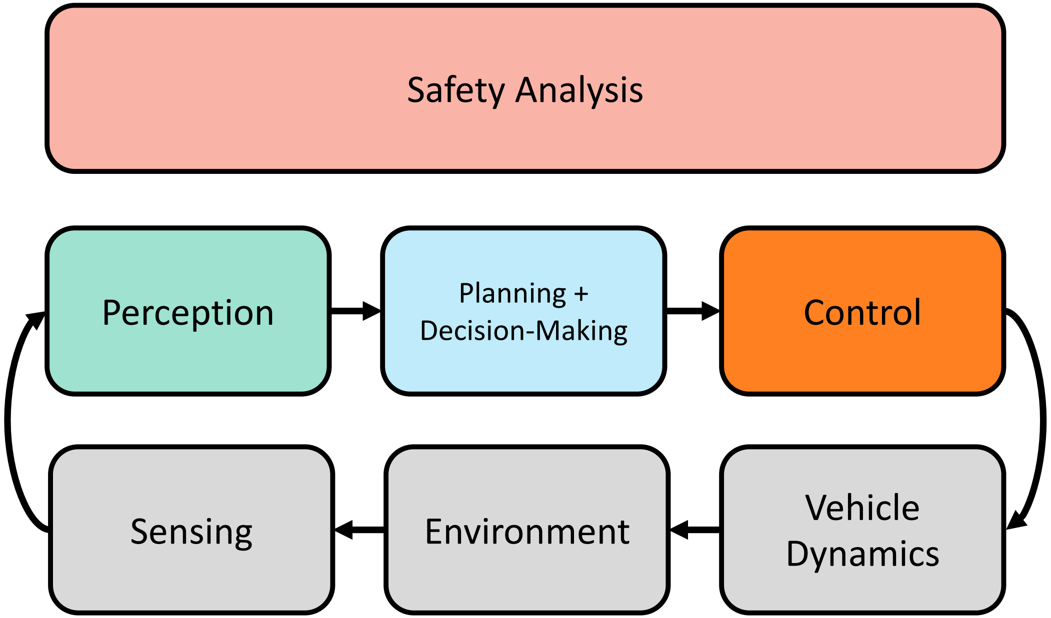
Two sessions:

- 9:30am lab sections AB1 & AB2
- 10:10am lab sections AB4 & AB5
- AB3 can attend either!

Indoor and outdoor tour – please dress appropriately!



Autonomous GEM Vehicle

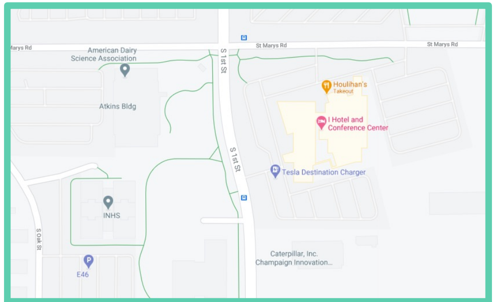


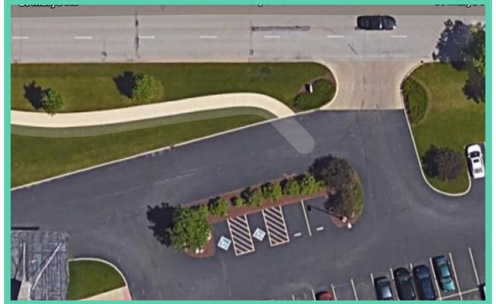
Today's Plan

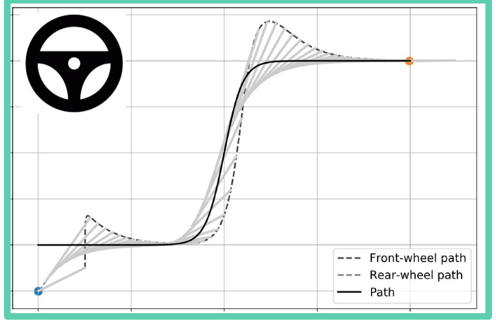
- What's a model?
- Planning and Control Motivation
  - Open-loop control
- Vehicle Models
  - How to design your model
  - Dubin's Car
  - Advanced Models: bicycle, tire dynamics

Typical planning and control modules

- Global navigation and planner
  - Find paths from source to destination with static obstacles
  - Algorithms: Graph search, Dijkstra, Sampling-based planning
  - Time scale: Minutes
  - Look ahead: Destination
  - Output: reference center line, semantic commands
- Local planner
  - Dynamically feasible trajectory generation
  - Dynamic planning w.r.t. obstacles
  - Time scales: 10 Hz
  - Look ahead: Seconds
  - Output: Waypoints, high-level actions, directions / velocities
- Controller
  - Waypoint follower using steering, throttle
  - Algorithms: PID control, MPC, Lyapunov-based controller
  - Lateral/longitudinal control
  - Time scale: 100 Hz
  - Look ahead: current state
  - Output: low-level control actions







What is control?

- A means of regulating or limiting something
- Algorithms (or process) for manipulating a system to achieve to desired value

Image Credit: Justas Galaburda and Vincent Mokuenko

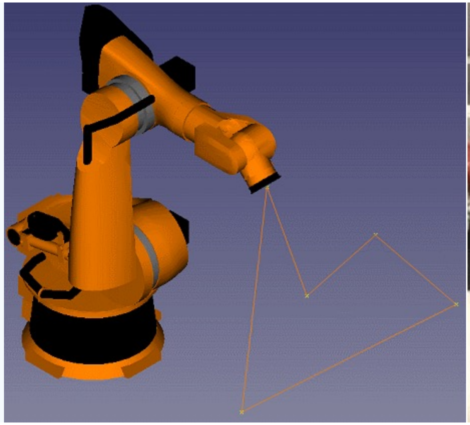
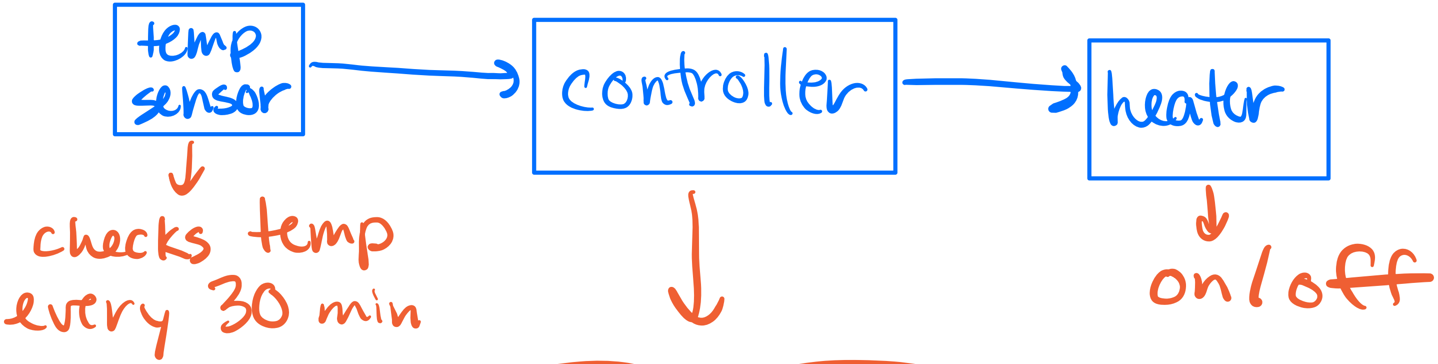


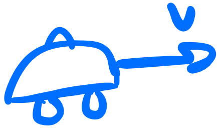
Image Credit: FreeCAD

Open-Loop Example



if temp ≤ 70 then
heat on for T min
if temp > 75 then
heat off (for T min)

Closed-Loop Example



goal: maintain v_d

Basic intuition:

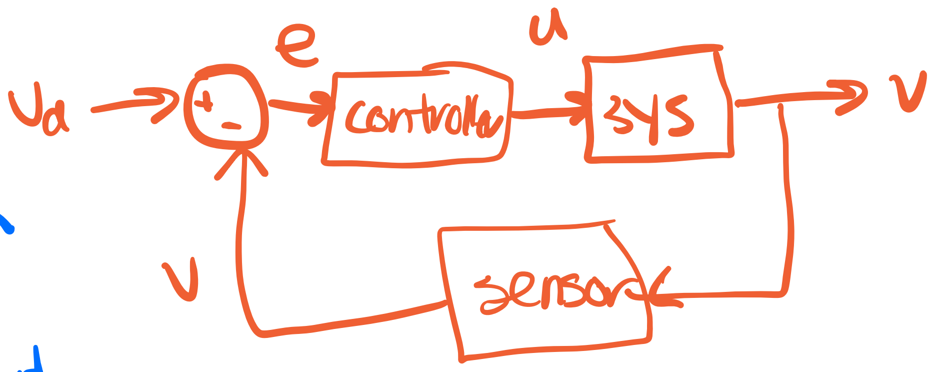
if $v > v_d$, want input ↓

if $v < v_d$, want input ↑

if $v \approx v_d$, want small input

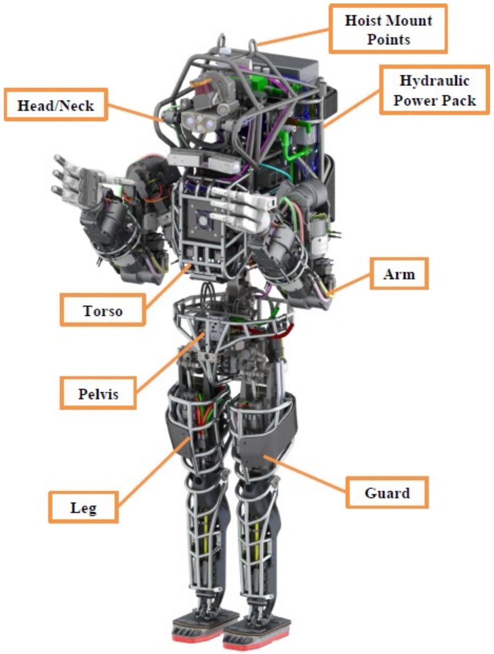
→ look at errors!

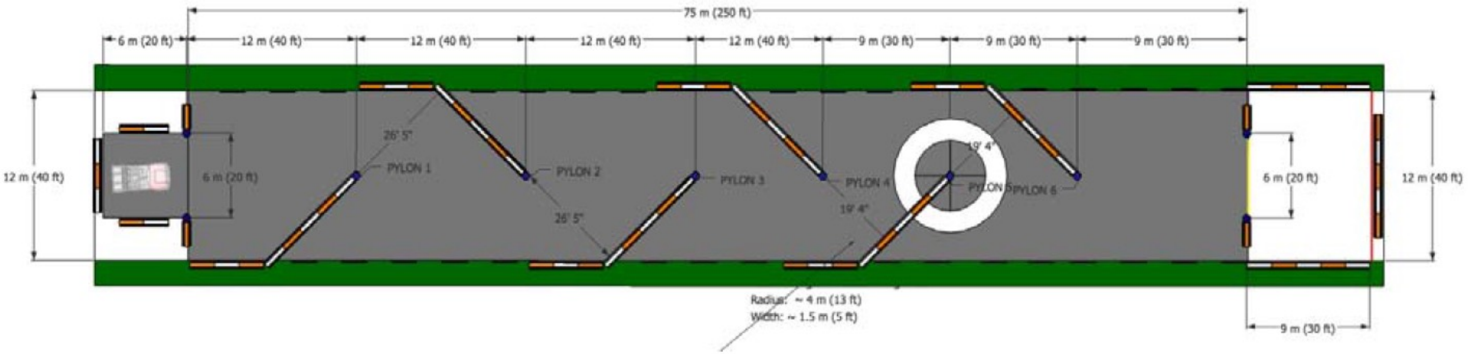
$e = v_d - v$



Complex control tasks: DARPA Robotics Challenge

- Robot drives the vehicle through the course
- Robot gets out of the vehicle and travels dismounted out of the end zone






POLARIS
2

First: Come up with a model!

For some common AV tasks, what are the desired behaviors, requirements of the system, actions/ inputs?

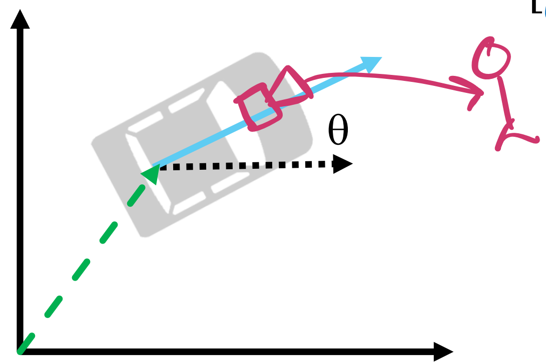
MP0 has a very simple model. How can it be improved?

task: staying in lane
actions: steering angle
assumption: const vel

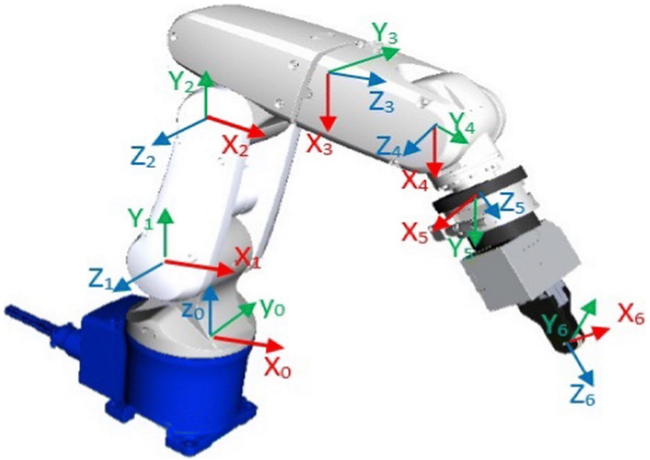
task: parking
actions: steering, acc

Coordinate Systems and Configurations

Car pose = $\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$



End effector pose = $\begin{bmatrix} x \\ y \\ z \\ \theta_x \\ \theta_y \\ \theta_z \end{bmatrix}$



Dynamical Systems Model

Describe behavior in terms of instantaneous laws:

$$\frac{dx(t)}{dt} = \dot{x}(t) = f\big(x(t), u(t)\big) \quad \Big| \quad x[t+1] = f(x[t], u[t])$$

x'

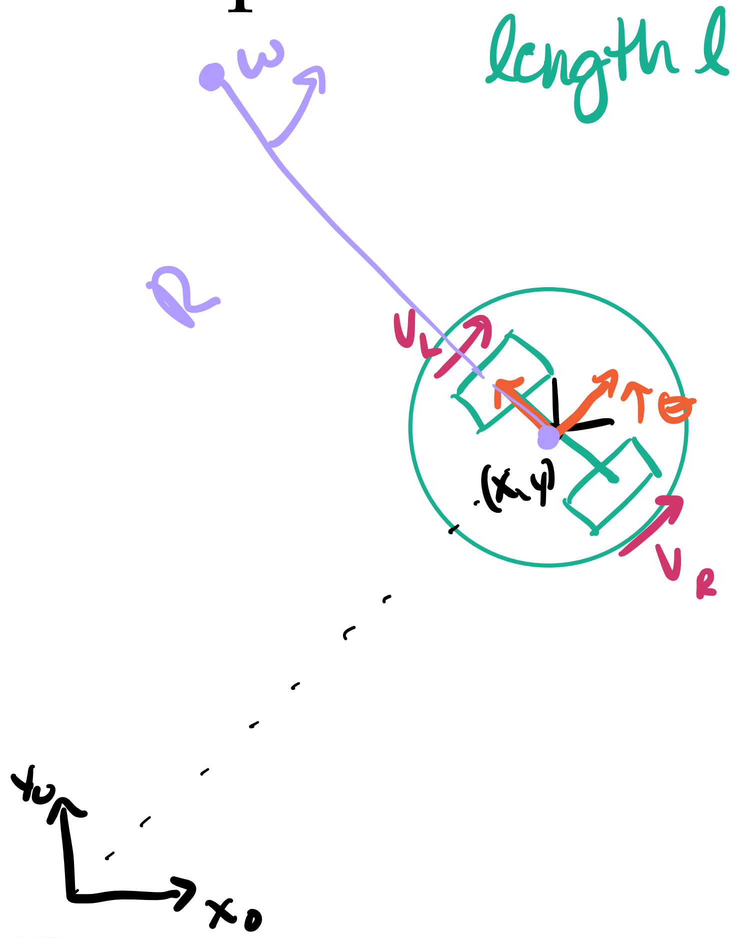
x_{t+1}

where $t \in \mathbb{R}, x(t) \in \mathbb{R}^n, u(t) \in \mathbb{R}^m$, and $f: \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ gives the dynamics / transition function

$$x_1 = \theta, \quad x_2 = \dot{\theta} = \dot{x}_1$$

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ \_\_\_\_ \end{bmatrix}$$

Example: Differential Drive Robot



Instaneous Center of Curvature

$$\begin{bmatrix} ICC_x \\ ICC_y \end{bmatrix} = \begin{bmatrix} x - R\sin\theta \\ y - R\cos\theta \end{bmatrix}$$

$$V_R = \omega (R + l/2)$$

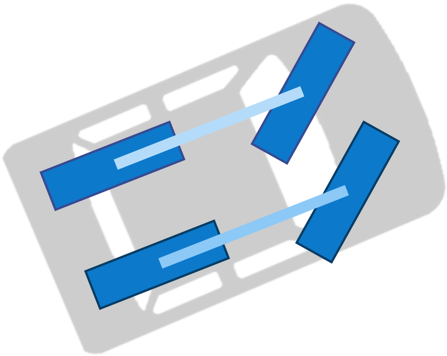
$$V_L = \omega (R - l/2)$$

$$R = l/2 \cdot \frac{(V_R + V_L)}{(V_R - V_L)}$$

$$\omega = \frac{V_R - V_L}{l}$$

Simple vehicle model: Dubin's car

- Key assumptions
  - Front and rear wheel in the plane in a stationary coordinate system
  - Steering input, front wheel steering angle δ
  - No slip: wheels move only in the direction of the plane they reside in
- Zeroing out the accelerations perpendicular to the plane in which the wheels reside, we can derive simple equations



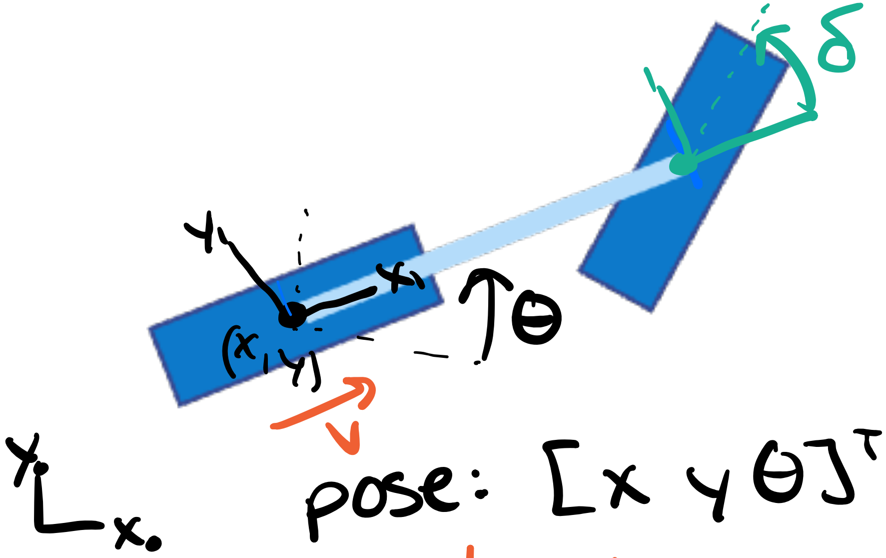
Reference: Paden, Brian, Michal Cap, Sze Zheng Yong, Dmitry S. Yershov, and Emilio Frazzoli. 2016. A survey of motion planning and control techniques for self-driving urban vehicles. IEEE Transactions on Intelligent Vehicles 1 (1): 33–55.

Dubin's Car

$$\dot{x} = v \cdot \cos\theta$$

$$\dot{y} = v \cdot \sin\theta$$

$$\dot{\theta} = v/l \tan\delta$$



pose: $[x\ y\ \theta]^T$

speed: v

steering: δ

length: l

Many more advanced models…

[Kinematic] Bicycle Model

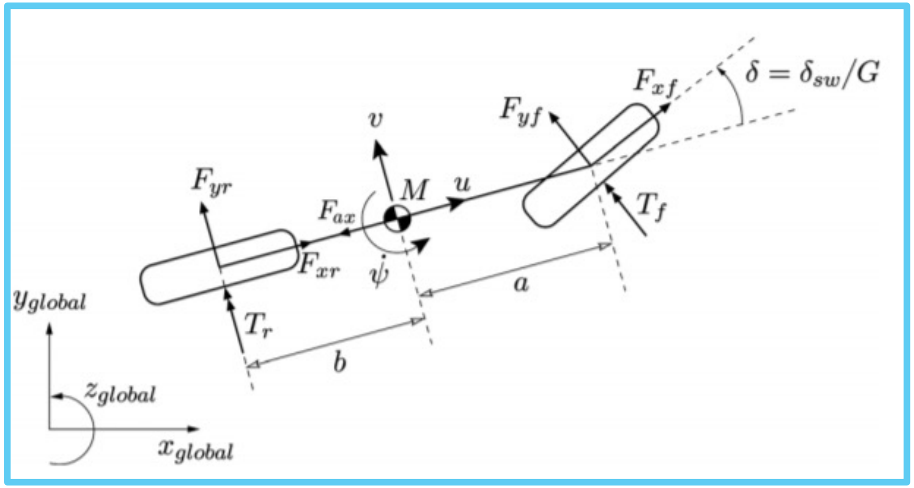
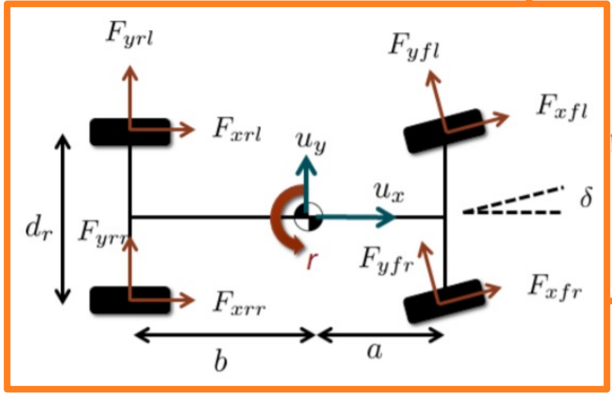


Image Credit and Reference: J.P. Timings and D.J. Cole. "Minimum maneuver time calculation using convex optimization." Journal of Dynamic Systems, Measurement, and Control 135.3 (2013).

[Dynamic] Tire Models



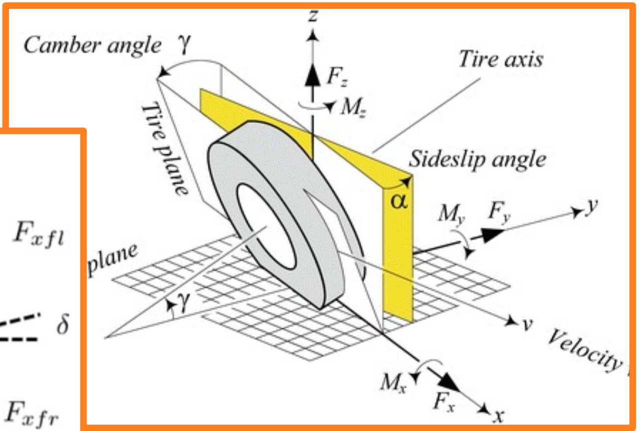
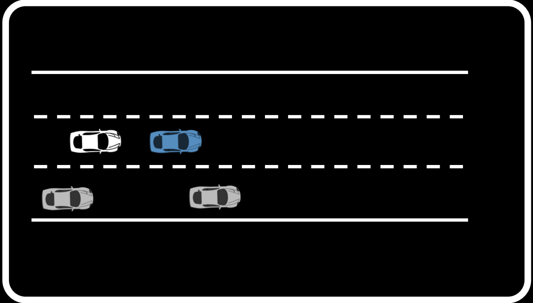


Image Credit: Springer

Image Credit and Reference: J.K. Subosits and J.C. Gerdes. "Impacts of Model Fidelity on Trajectory Optimization for Autonomous Vehicles in Extreme Maneuvers." IEEE Transactions on Intelligent Vehicles, 2021.

Dynamical system models



Nonlinear dynamics

Generally, nonlinear ODEs do not have closed form solutions!

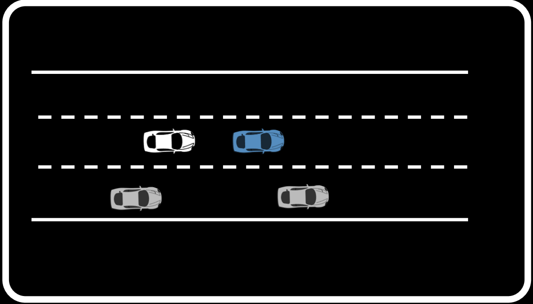
Dubin's car model

$\dot{v} = a$	Speed
$\frac{ds_x}{dt} = v\cos(\psi)$	Horizontal position
$\frac{ds_y}{dt} = v\sin(\psi)$	Vertical position
$\frac{d\delta}{dt} = v_\delta$	Steering angle
$\frac{d\psi}{dt} = \frac{v}{l}\tan(\delta)$	Heading angle

Physical plant

$\frac{dx}{dt} = f(x, u)$	System dynamics
$x[t+1] = f(x[t], u[t])$	
$x = [v, s_x, s_y, \delta, \psi]$	State variables
$u = [a, v_\delta]$	Control inputs

Nonlinear *hybrid* dynamics



Physical plant

$$\frac{dx}{dt} = f(x, u)$$

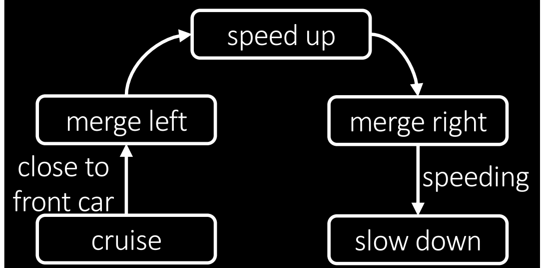
$$x[t+1] = f(x[t], u[t])$$

System dynamics

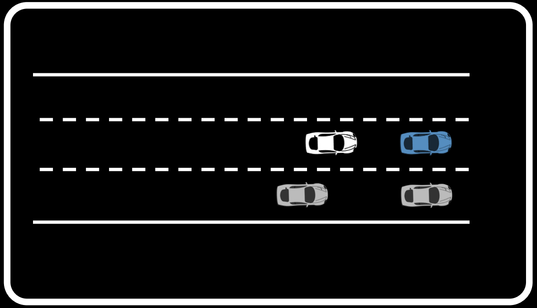
$x = [v, s_x, s_y, \delta, \psi]$ State variables

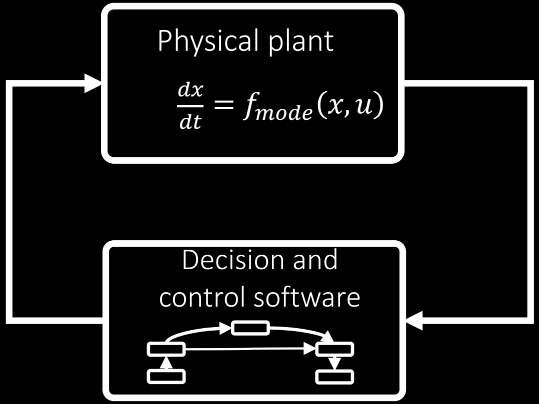
$u = [a, v_\delta]$ Control inputs

Decision and control software



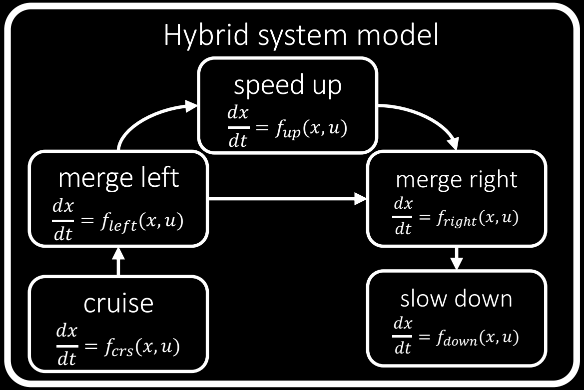
Typical system models





Nonlinear *hybrid* dynamics

Interaction between computation and physics can lead to unexpected behaviors



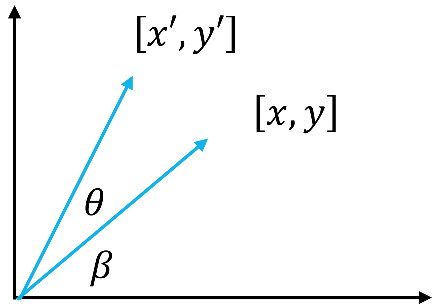
Summary

- Dynamical systems models allow us to reason about low-level behaviors of systems and determine what is and is not feasible
  - Typically required to design controllers!
- Discussed a few types of models from simple to complex
- *Next time:* Look at simple PID control design for trajectory following

Extra Slides

An aside: Coordinate transformations
Rotation matrix

The following matrix rotates a vector $[x, y]$ counter-clockwise by an angle of θ



$$R(\theta) = \begin{bmatrix} \cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}$$

That is:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Derivation

$x' = rcos(\beta + \theta) = r(cos\beta cos\theta - sin\theta \sin\beta)$

$= rcos\beta cos\theta - r\mathrm{sin}\theta\sin\beta$

$= xcos\theta - y\mathrm{sin}\theta$

Path following control

- The path followed by a robot can be represented by a *trajectory or path* parameterized by time
  - → from a higher-level planner
- Defines the desired instantaneous pose $p(t)$