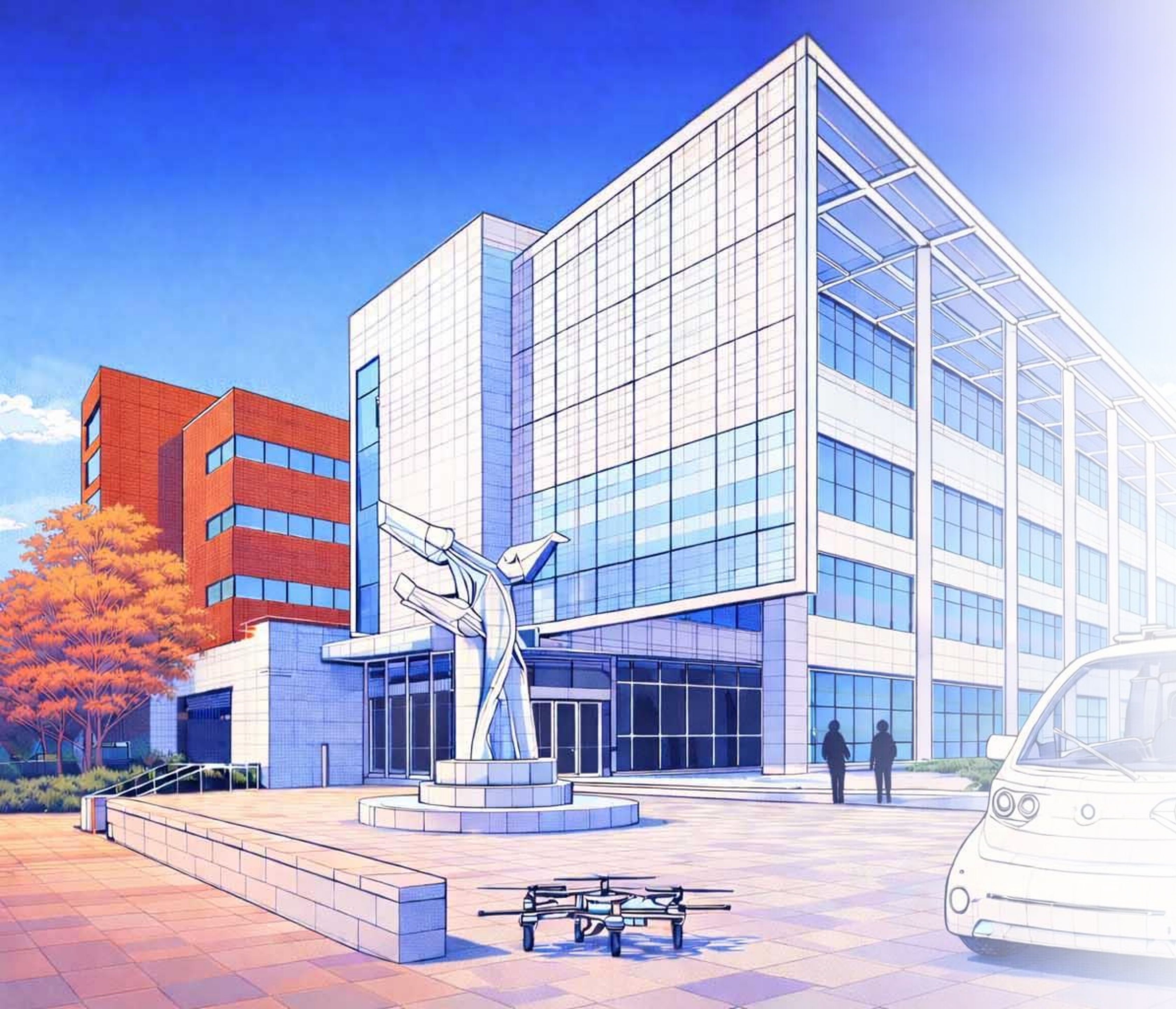


Principles of Safe Autonomy

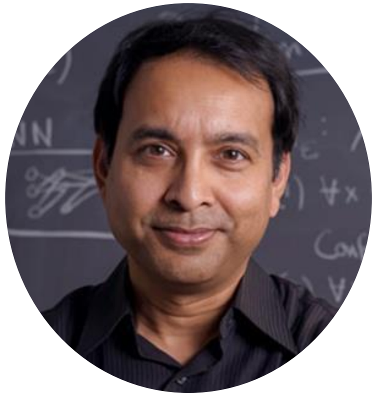
ECE 484 Spring 2026

Professor Sayan Mitra (mitras)

Jan 20, 2026

https://safeautonomy-illinois.github.io/ece484-site/

Warm welcome from ECE484 Spring 2026 team!



Prof. Sayan Mitra (mitras)

CSL 266

Graduate TAs

Hanna Chen

Abhishek Pai

James Menezes

Fatemeh Cheraghi

Alex Yuan

Laboratory support staff

Suraj Nair

John Hart

Undergraduate CAs

Tanvi Kulkarni (tanvik4)

Bach Hoang (bachh2)

Yanhao Yang (yanhaoy2)

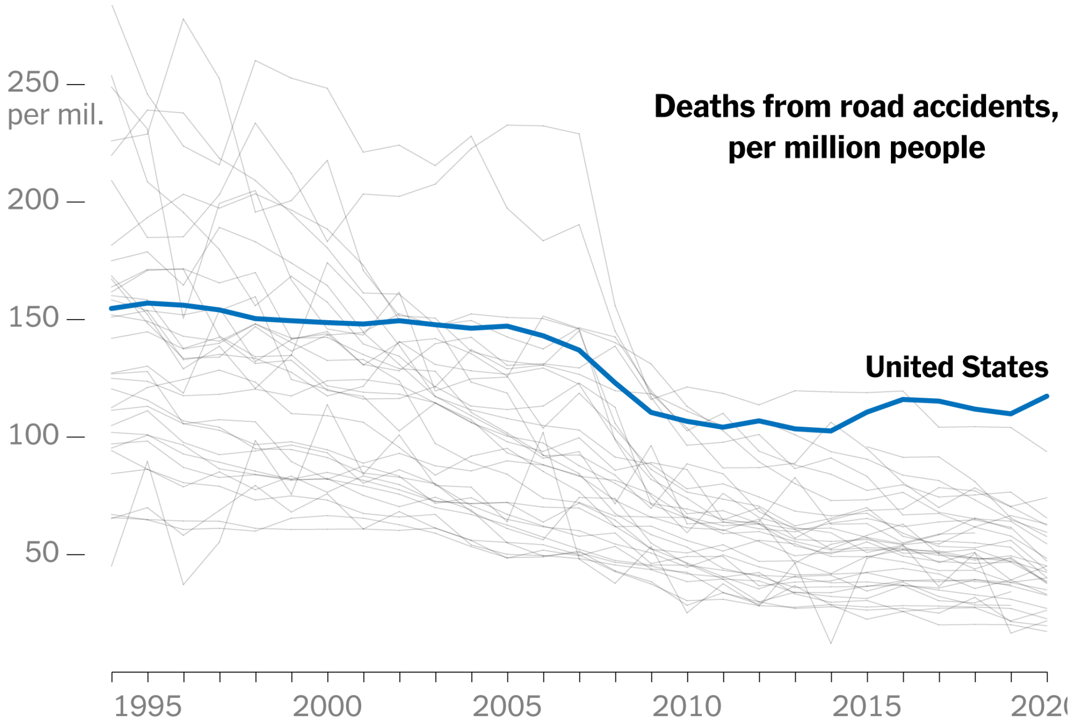
Plan for today

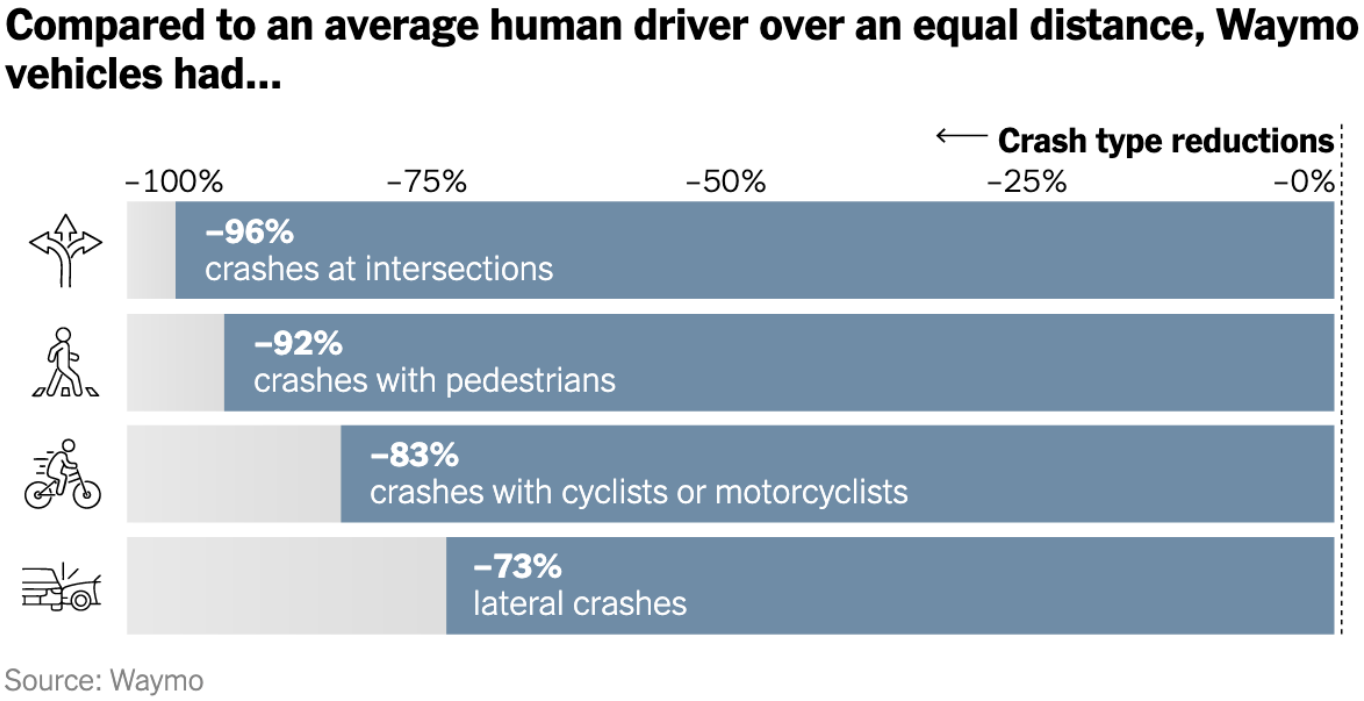
- Motivation
- Administrivia
- Introduction to Safety

Motivation

Autonomous systems can substantially benefit society, provided safety risks are mitigated

- Driverless cars will improve productivity
  - Americans drives 13,474 miles (~300 hours) per year
- Make cities greener
  - 40% of city surface is parking
- Make roads safer
  - Still 32K+ fatalities and 3M+ injuries every year in the USA





[The Data on Self-Driving Cars Is Clear. We Have to Change Course].

12/2/25 NYTimes, Jonathan Slotkin

Recent accomplishments in robotics & autonomy

- NASA’s Perseverance rover performed autonomous science operations on Mars, the **Ingenuity helicopter** performed the first powered, controlled flights on another planet (2021-22).

- Zipline, Wing, Amazon Prime Air have launched commercial deliveries. Air taxis are on the horizon.

O'HARE

Air Taxi To O'Hare Will Allow Chicago Travelers To Skip Traffic On The Kennedy

The city's first air taxi plans to launch in 2025. Company officials say the cost will be competitive with a rideshare between Downtown and the airport.

Ariel Parrella-Aureli 7:22 AM CDT on Mar 27, 2023

Credit: Archer Aviation

Autonomy Is a Frontier of Engineering

Autonomy is no longer about isolated demos. It is a frontier where **perception, learning, control, and verification meet reality**.

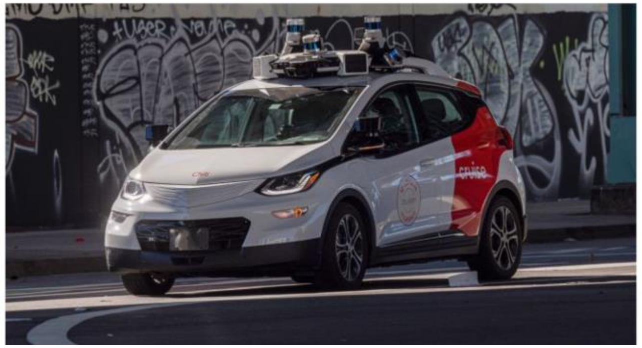
Despite striking advances, autonomous systems remain constrained by:

- **Cost** — sensing, compute, testing, and deployment at scale
- **Reliability** — rare failures, edge-cases, and long-tail uncertainty
- **Energy** — perception and learning on real, resource-limited machines

They are **fundamental engineering challenges**.

San Francisco Wants Halt to Cruise, Waymo Expansion Ruling

City Says Expanded Service for Robotaxis Can Cause 'Serious Harm'

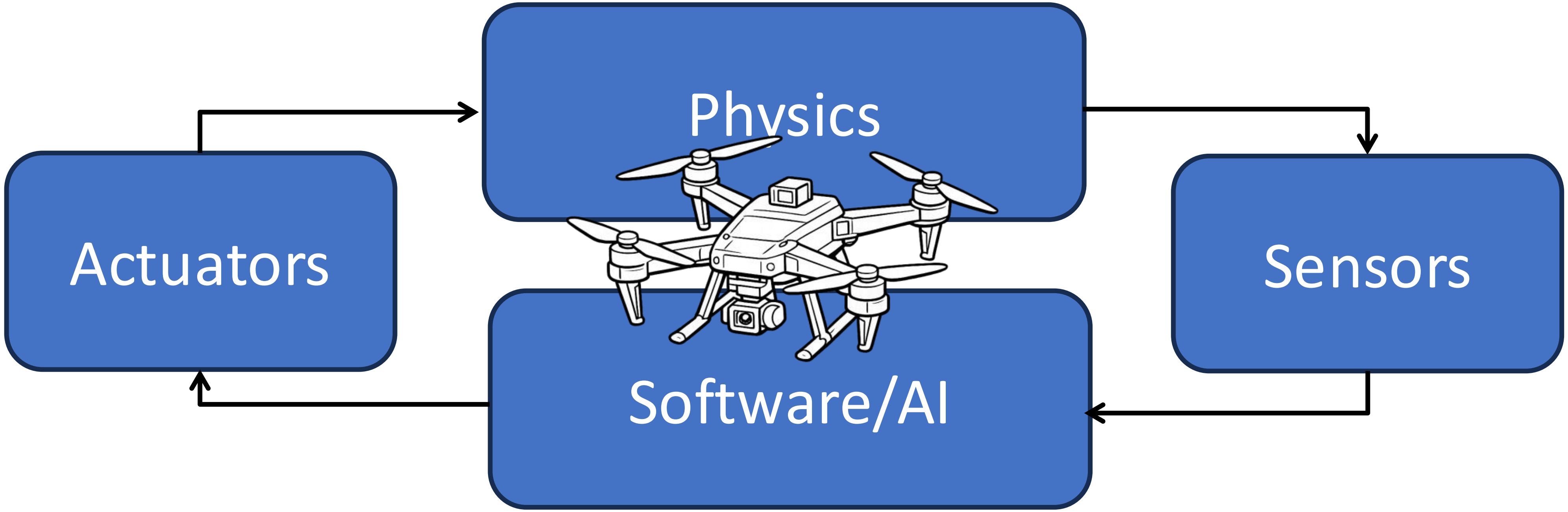


Aug. 19, 2023, at 12:32 p.m. General Motors' Cruise autonomous vehicle unit has agreed to cut its fleet of San Francisco robotaxis in half as authorities investigate two recent crashes in the city.

Principles of Safe Autonomy ECE 484

Autonomy is a frontier of engineering where perception, learning, control & verification meet the real world and ECE484 is where you begin to shape it.

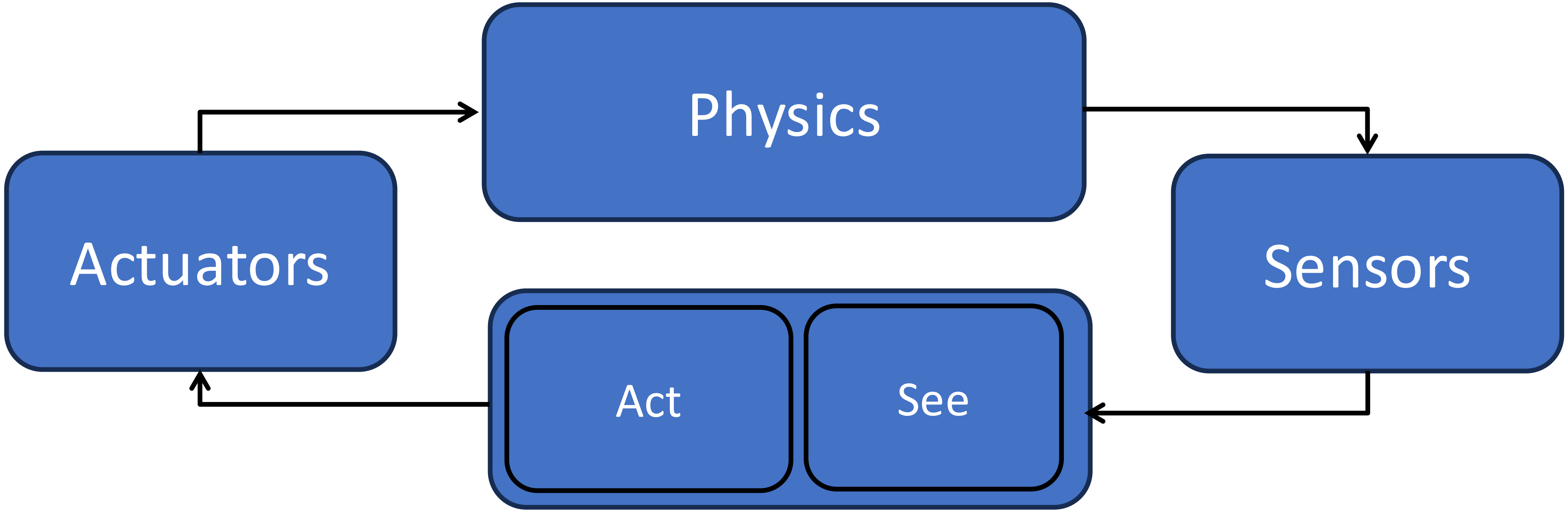
ECE484 develops the foundational principles behind autonomy



Principles of Safe Autonomy ECE 484

Autonomy is a frontier where perception, learning, control, and verification meet the real world and ECE484 is where you begin to shape it.

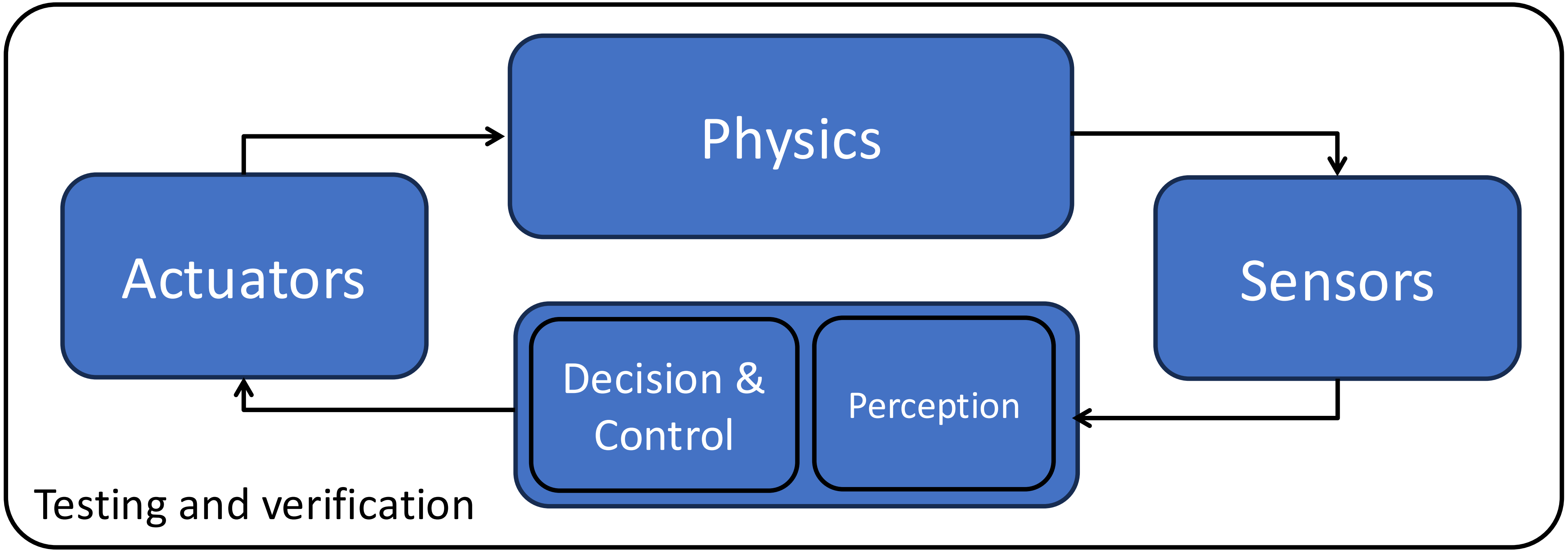
ECE484 develops the foundational principles behind autonomy



Principles of Safe Autonomy ECE 484

Autonomy is a frontier where perception, learning, control, and verification meet the real world and ECE484 is where you begin to shape it.

ECE484 develops the foundational principles behind autonomy



Principles of Safe Autonomy ECE 484

ECE484 develops the foundational principles behind autonomy, organized around three pillars:

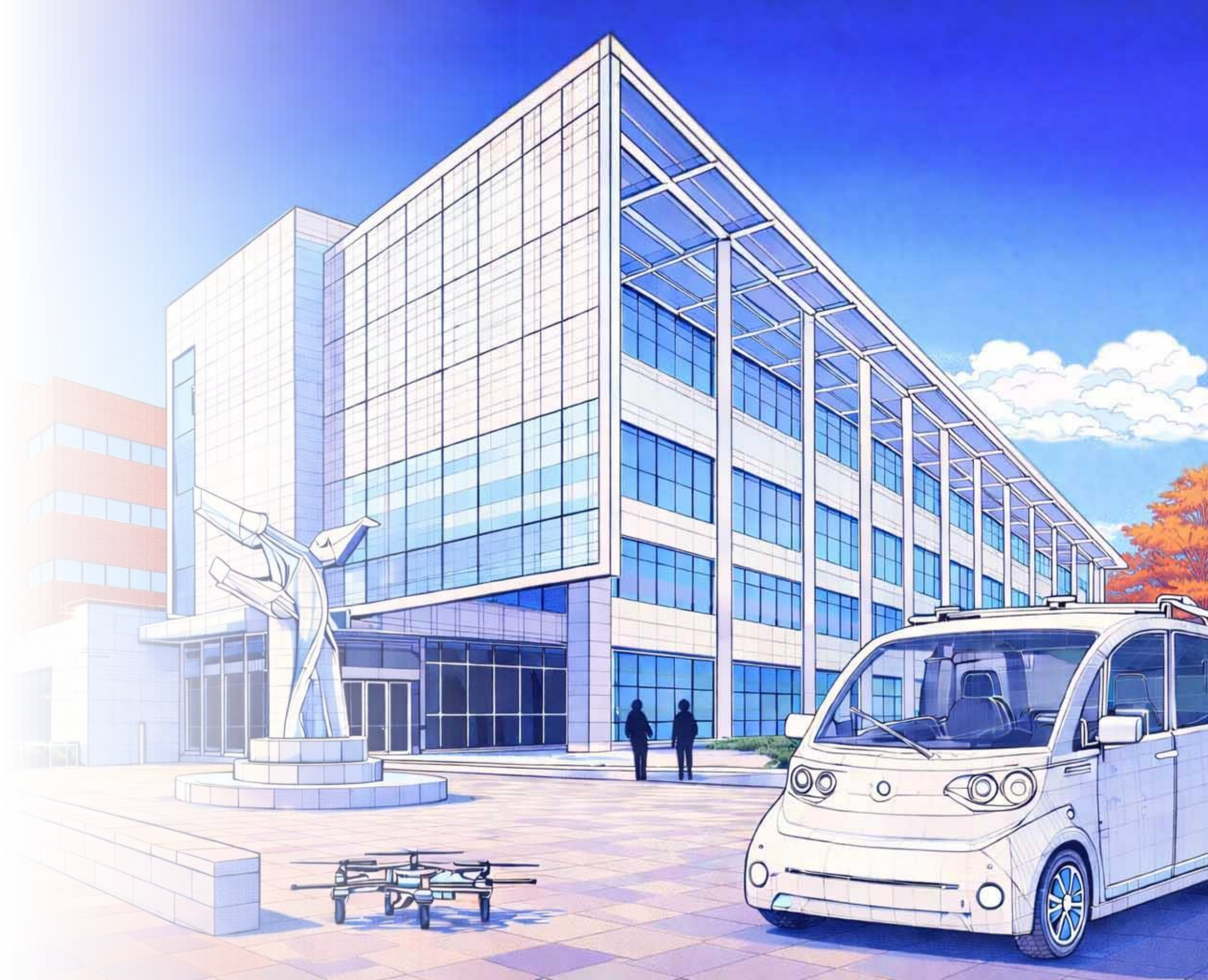
- Sensing, Perception & State Estimation
  - Turning raw sensor bits into state information
- Planning, Decision-Making & Control
  - Computing actions that are safe, robust, and effective based on state estimates
- Testing, Evaluation & Formal Verification
  - Reasoning about correctness, uncertainty, and failure

Machine learning is a cross-cutting enabling tool

Through hands-on implementation and system-level projects, you will engineer a complete autonomy stack for a sensor-rich platform, with emphasis on rigorous evaluation and safety-aware design.

Outline

- Motivation
- Administrivia
- Introduction to Safety



Administrivia

Website: https://safeautonomy-illinois.github.io/ece484-site/

- Policies, schedule, lab, resources, homework, code, project,

Campuswire for announcements, but no SLA, best effort response delay ~2 days.

- Discussions, forming teams, occasional polls, feedback

Canvass: https://canvas.illinois.edu/courses/67113

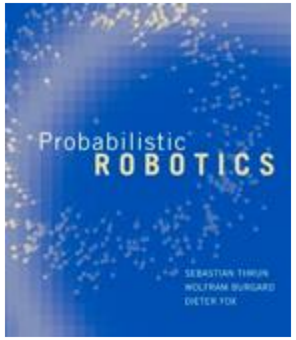
- Grade release and assignment submission

Week	Date	Day	Session	Topic	Notes	Private notes
1	20-Jan	Tue	Lecture 1	Overview / Intro to Safety / Linear Algebra		
1	22-Jan	Thu	Lecture 2	Safety 2: verification concepts, automata, requirements, counter-examples (slides)		
1	23-Jan	Fri	Lab	Intro, MP0 walkthrough slides	MP0 released	
2	27-Jan	Tue	Lecture 3	Safety 3: reachability, inductive invariants (slides)		
2	29-Jan	Thu	Lecture 4	Perception 1: neural networks, gradient descent (slides)		
2	30-Jan	Fri	Lab	Team formation, MP1 walkthrough	MP1 released	
3	3-Feb	Tue	Lecture 5	Perception 2: intrinsic, extrinsic matrices, homogeneous coordinates (slides)		
3	5-Feb	Thu	Lecture 6	Perception 3: calibration, perspective, projection, eigenvalue problem (slides)		
3	6-Feb	Fri	Lab	MP0 Demo (MP0 Due)	-	
4	10-Feb	Tue	Lecture 7	Perception 4: depth estimation, visual odometry, fundamental matrix, epipolar geometry (slides)		
4	12-Feb	Thu	Lecture 8	Control 1: ODEs, lipschitz contuinity, bang-bang control (slides)		
4	13-Feb	Fri	Lab	**MP1 Demo (MP1 due) (from week 5)**	-	
5	17-Feb	Tue	Lecture 9	GEM Field Trip (meet at the highbay facility)		
5	19-Feb	Thu	Lecture 10	Project Workthrough F1 Tenth, GRAIC, and Drone projects (meet at ECEB 1015; we will walk to CSL studio)		
5	20-Feb	Fri	Lab	**MP2 walkthrough slides (from week 4)**	**MP2 released**	
6	24-Feb	Tue	MT	Review Session (in class)		
6	26-Feb	Thu	MT	Midterm 1		Midterm 1
6	27-Feb	Fri	Lab	Open Lab		
7	3-Mar	Tue	Lecture 11	Control 2: PID, linear systems (slides)		
7	5-Mar	Thu	Lecture 12	Control 3: linear systems, stability, Lyapunov, Hurwitz criteria (slides)		
7	6-Mar	Fri	Lab	**MP2 Demo (MP2 due) (from week 6)**	-	
8	10-Mar	Tue		Project review 1		
8	12-Mar	Thu				
8	13-Mar	Fri	Lab	**MP3 Walkthrough slides (from week 7)**	**MP3 released**	
9	24-Mar	Tue	Lecture 13	Filtering 1: Markov chains, conditional probability, motion models (slides)		
9	26-Mar	Thu	Lecture 14	Filtering 2: localization bayes filter, histogram filter, beliefs (slides)		
9	27-Mar	Fri	Lab	Open Lab	-	
10	31-Mar	Tue	MT	Review Session (in class)		Review Session (in class)
10	2-Apr	Thu	MT	Miderm 2		
10	3-Apr	Fri	Lab			
11	7-Apr	Tue	Lecture 15	Filtering 3: Kalman filter, localization particle filter, importance sampling (slides)		
11	9-Apr	Thu	Lecture 16	Filtering 4: review; SLAM (slides)		
11	10-Apr	Fri	Lab	**MP3 Demo (MP3 due) (from week 10)**	-	
12	14-Apr	Tue	Lecture 17	Planning 1: graph search, uniform cost search (slides)		
12	16-Apr	Thu	Lecture 18	Planning 2: A*, optimal search, cost-to-go heuristics (slides)		
12	17-Apr	Fri	Lab	(See CampusWire for details of the project checkpoint)	-	
13	21-Apr	Tue	Lecture 19	Planning 3: hybrid A*, PRM, probabilistic completeness (slides)		
13	23-Apr	Thu	Lecture 20	Planning 4: RRT, RRG, asymptotic optimality (slides)		
13	24-Apr	Fri	Lab	-	-	
15	28-Apr	Tue	Lecture 21	Guest Lecture		
15	30-Apr	Thu	MT	Review Session (in class)		
15	1-May	Fri	Lab	-	-	
15	5-May	Tue	MT	Midterm 3	Last class	Sayan traveling 5-12; remote review of project possibly
15	6-May	Wed			Last day of classes	
	7-May				Reading day	
	8-May				Final exam week	
	14-May			Final presentation during final exam; project website presubmitted.	Final exam week	

Course materials

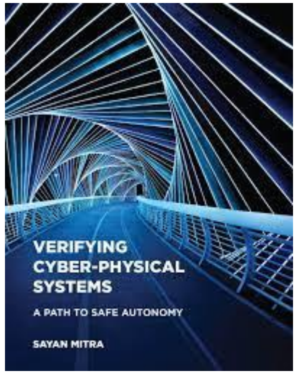
Primary sources

- Lectures and slides (from course webpage)
- Course reader (~100 pages with exercises): https://github.com/safeautonomy-illinois/ece484-site/blob/main/docs/assets/pdfs/coursereader.pdf
- Do the exercises and HW problems without using AI for midterms



References:

- *Probabilistic robotics, By Sebastian Thrun, Wolfram Burgard, and Dieter Fox, 2005*
- *Verifying Cyber-physical Systems, Sayan Mitra, MIT Press 2021*

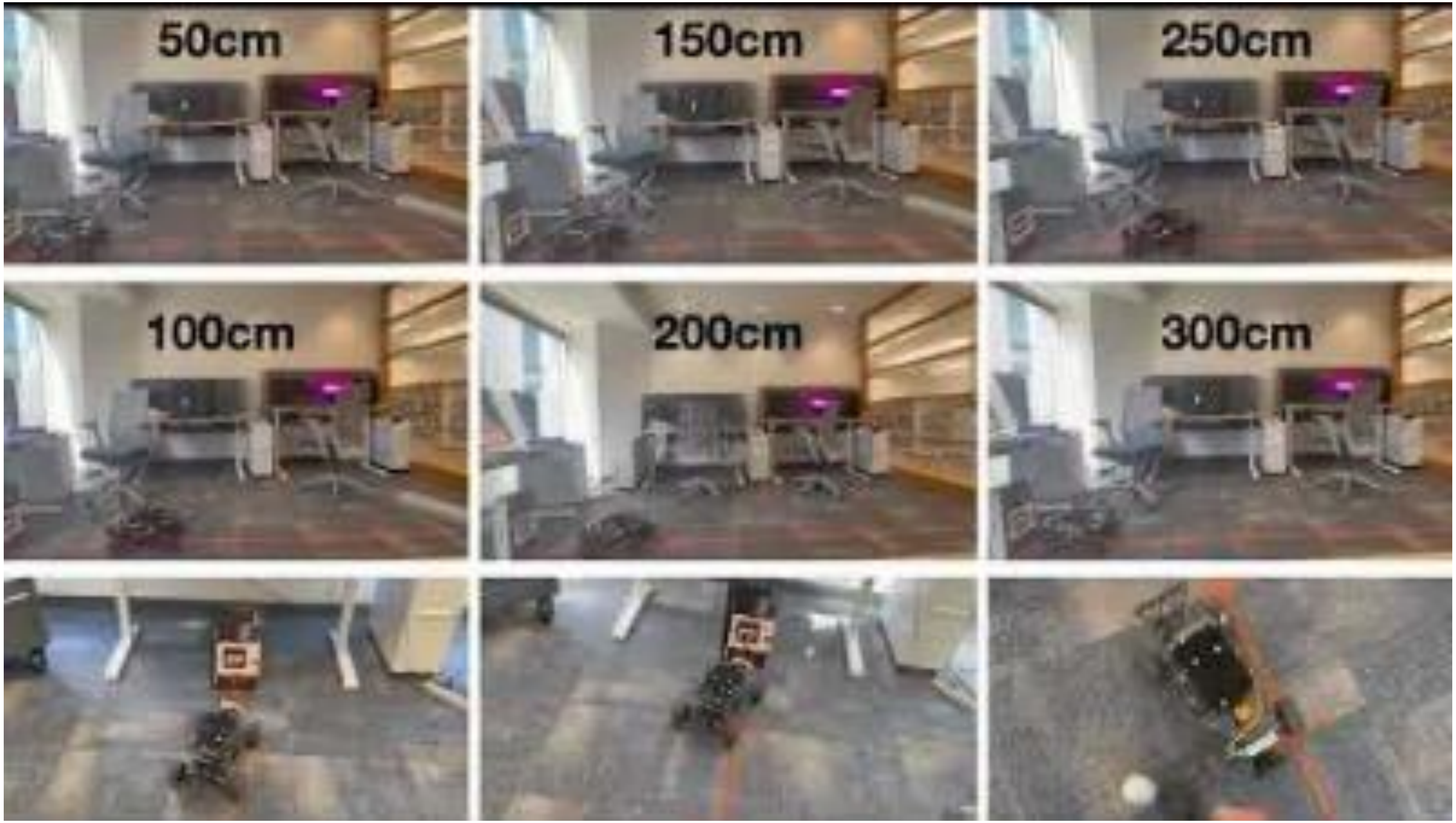


Course: components and (tentative) weights

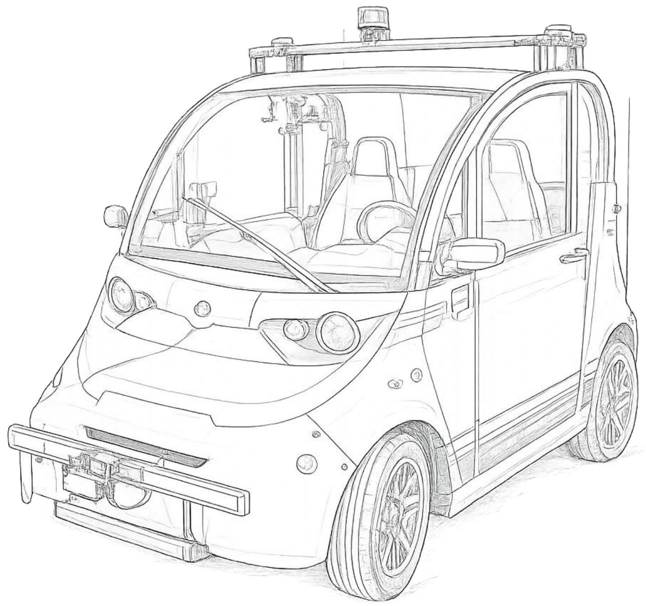
- 3-4 programming assignments or MPs **35%**
  - ROS + Python, Ubuntu, VM BYOD or use lab workstations
  - **Labs** (Friday 9am-8 pm starting 1/24 ECEB5072) walkthroughs and demos
  - Office hours TBD
  - MP0 will be individual **(starting 1/24),** MP1-3 in teams
- Homework assignments **10%** (individual)
  - math, analysis, critical reasoning; preparation for midterms
- Midterms x3 **30%** (individual); no final exam
  - In class, pencil-paper, based on HWs and CR exercises
- Project **25%** (group): more on this later, 4 tracks
  - Milestone and metrics driven development of autonomy stack
  - In teams
  - Presentation/demo/race during finals week

Tentative grade boundaries	
A	>90
B	>80
C	>65
D	>55

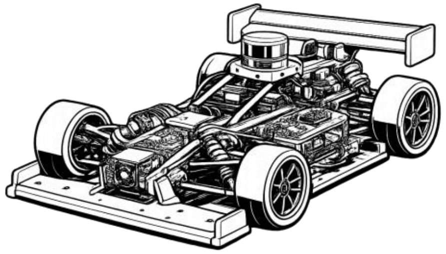
Example of what you will build in 484

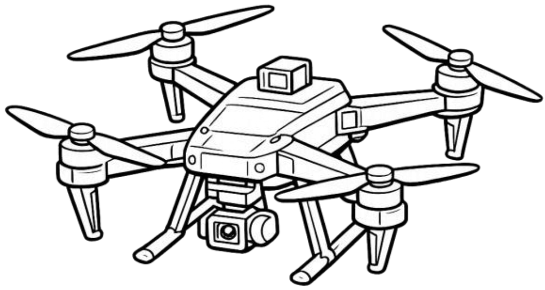


Projects: explore, inspire, and impress



- Build cool system and evaluate rigorously
- We provide platforms, drivers, simulators, and coaching,
- Your team of 3 develop software to meet milestones and optimize performance metrics
  - Polaris GEM Full-sized vehicle with LiDAR, multiple cameras
  - F1tenth Scaled RC car racing with camera and lidar
  - Quadrotors racing through gates using camera
  - GRAIC Simulation-based race in tacks with obstacles
- Timeline:
  - Form a project team, see past projects, decide track (now)
  - High-bay virtual site visit and training (in next 2 weeks)
  - Intermediate check-ins
  - Public presentation, demo, awards (End of Semester)
- Next: Jumpstart research, land autonomy jobs, found startups





Spring 2025 projects

Spring 2022 projects

Spring 2020 projects

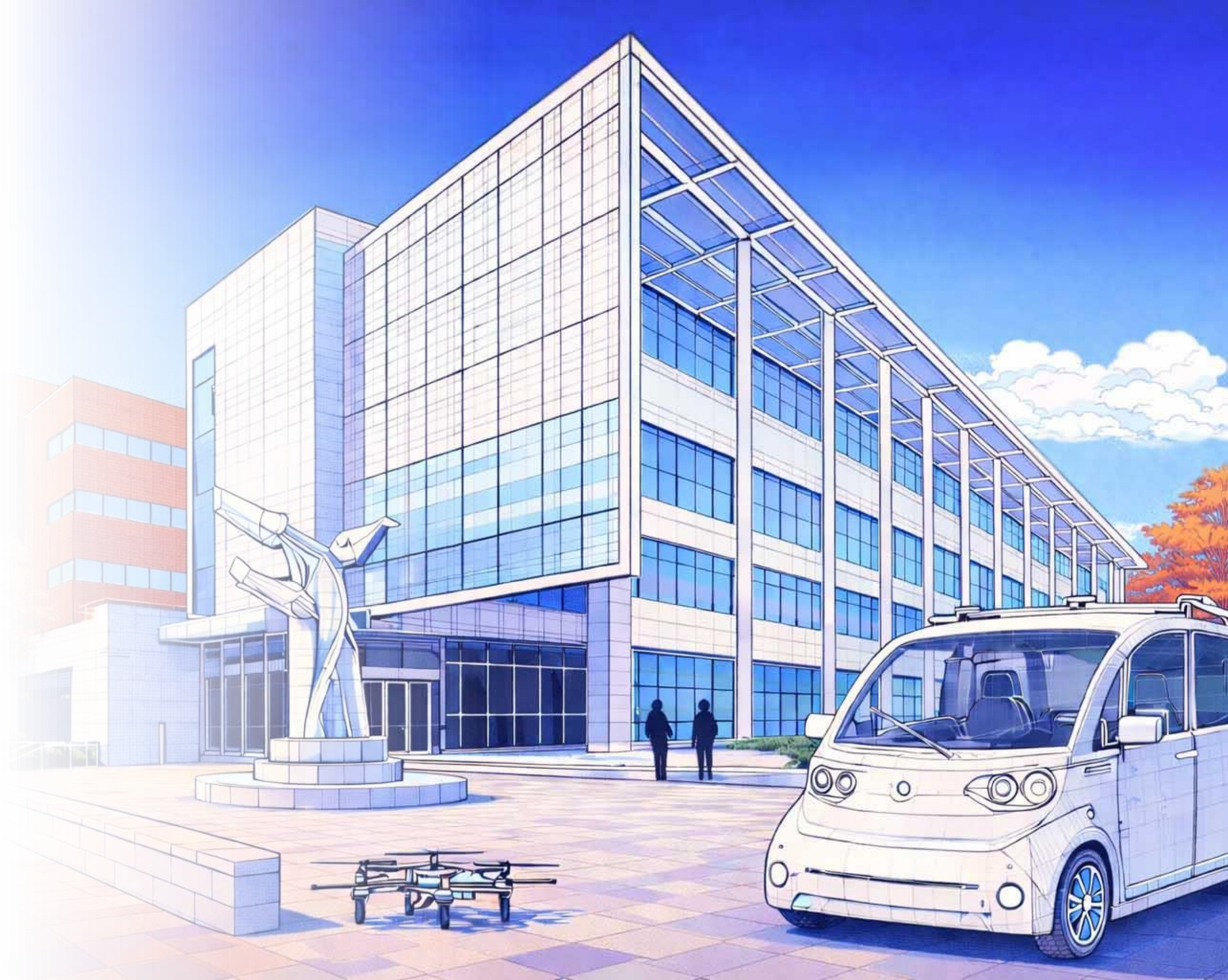
Fall 2020 projects

Outline

Motivation

Administrivia

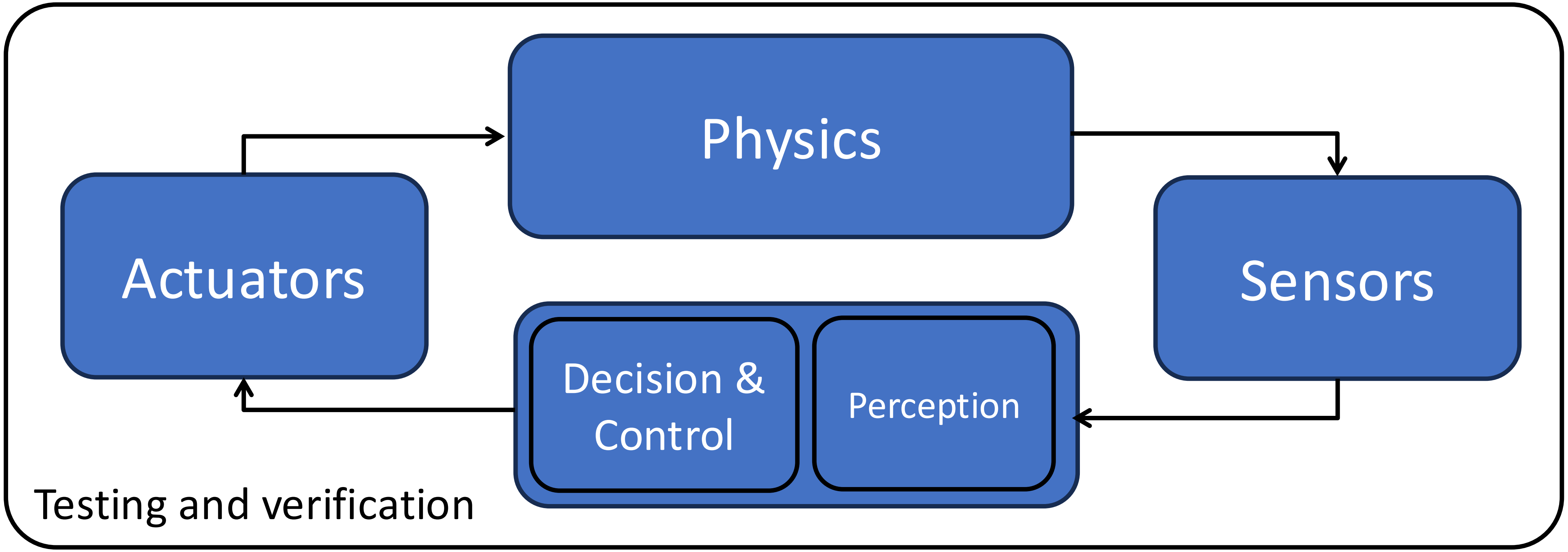
Introduction to Safety



Principles of Safe Autonomy ECE 484

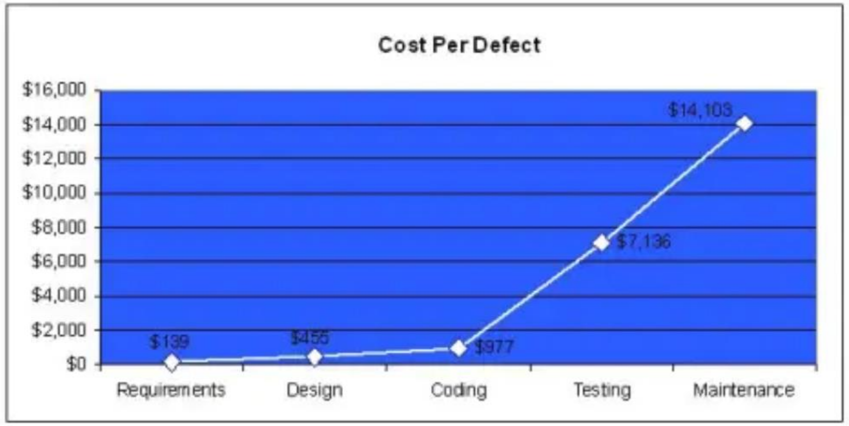
Autonomy is a frontier where perception, learning, control, and verification meet the real world and ECE484 is where you begin to shape it.

ECE484 develops the foundational principles behind autonomy



Cost of unreliability in autonomous systems

- Therac-25 radiation therapy machine delivered overdoses because of software bug which resulted in 6 fatalities.
- Elaine Herzberg was killed by self-driving Uber prototype in Tempe, Arizona in March 2018.
- Data conversion error caused the **$500M Ariane 5 rocket** to veer off course and explode shortly after launch.
- GM’s Cruise autonomous vehicle unit shut down its San Francisco robotaxi fleet after crashes in 2023.
- Cost of defects grow exponentially with time of discovery



Capers Jones, Software Assessments, Benchmarks, and Best Practices, Addison-Wesley, 2000

Limitations of testing for reliability and safety

Testing is an important and essential method for checking correctness of systems, but

"Testing can be used to show the presence of bugs, but never to show their absence!''

--- Edsger W. Dijkstra

Amount of testing required for autonomous systems can be prohibitive

- Probability of a fatality caused by an **accident per one hour of human driving** is ~ 10^{-6}
- Assume that for AVs this has to be 10^{-9}
- Data required to guarantee a probability of 10^{-9} fatality per hour of driving is proportional to its inverse, **10^9 hours, 30 billion miles**
- Multi-agent, open system, with human interactions => cannot be simulated offline to generate data
- Any change is software means tests must be rerun

On a Formal Model of Safe and Scalable Self-driving Cars by Shai Shalev-Shwartz, Shaked Shammah, Amnon Shashua, 2017 (Responsibility Sensitive Safety)

Mathematically Checking truthfulness of statements

The ultimate standard for truth: A theorem with a proof

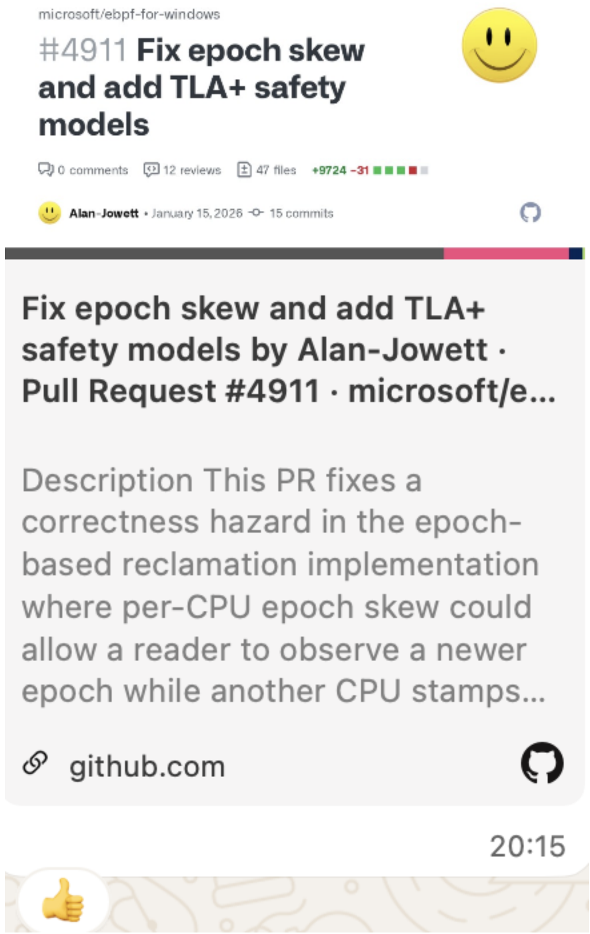
Formal verification: *The science of proving or disproving truth of statements asserting correctness of systems*

Proofs are used at scale in Amazon, Meta, Microsoft, NASA, …

"In 2017 alone the security team used deductive theorem provers or model checking tools to reason about cryptographic protocols, hypervisors, boot-loaders, firmware, garbage collectors, and network designs." Byron Cook, Amazon

In MP0 you will use a tool called Verse to verify collision avoidance of UAVs

To prove theorems about system safety, first we need precise assumptions about the its behavior and environment --- this is a model



Byron Cook at FLoC 2018
https://www.youtube.com/watch?v=JfjLKBO27nw

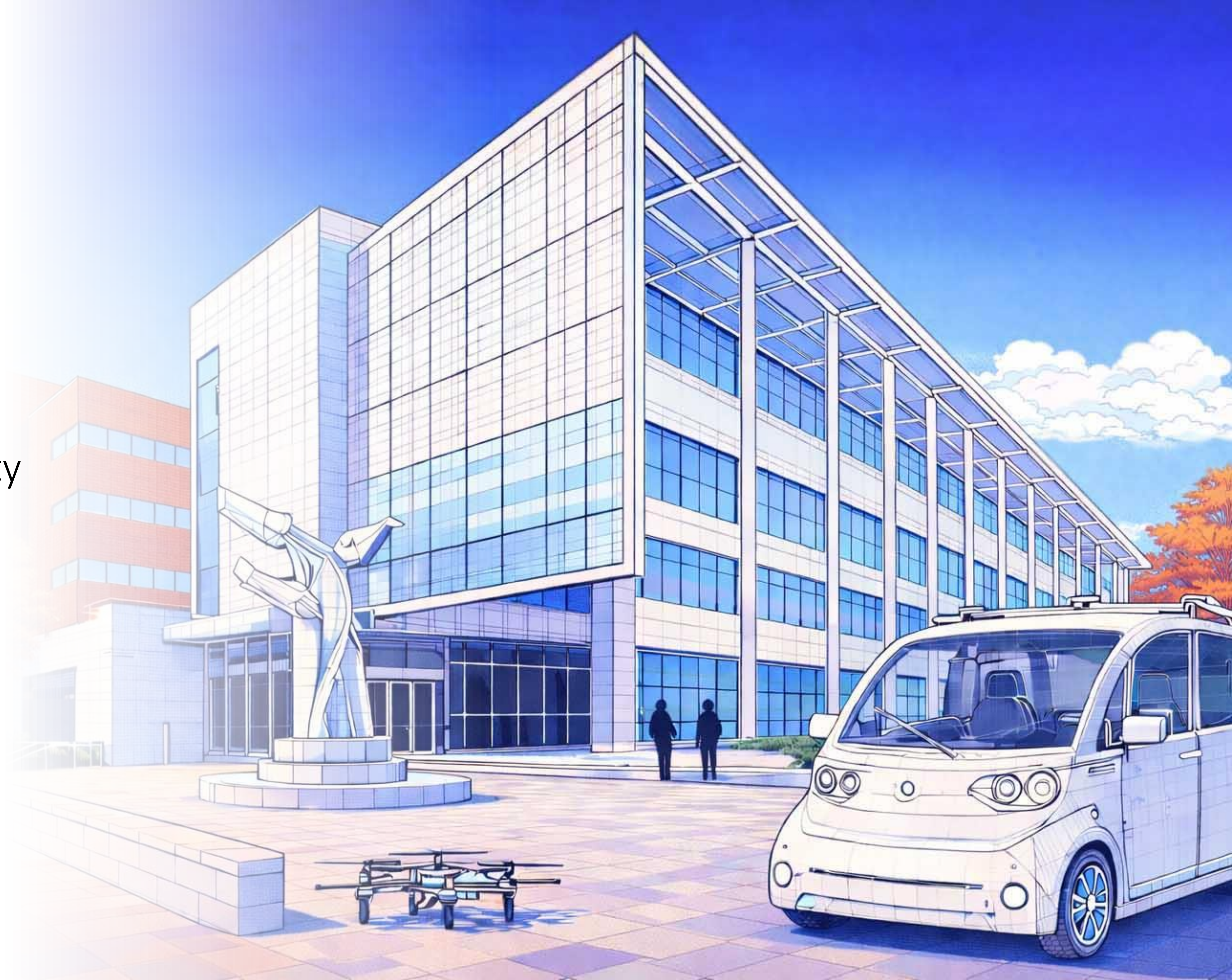
Outline

Motivation

Administrivia

Introduction to Safety

- Models
- Requirements
- Proofs



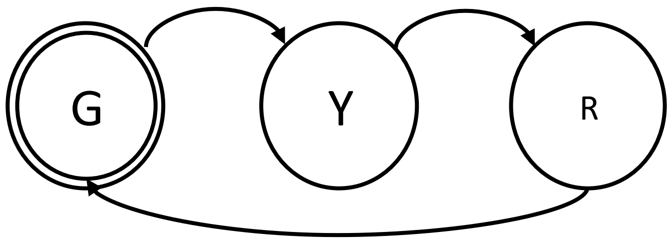
Automata or state machine models

An automaton A is defined by a triple $\langle Q, Q_0, D\rangle$, where

- Q is a set of states
- $Q_0 \subseteq Q$ is a set of initial states
- $D \subseteq Q \times Q$ is a set of transitions

An execution of A is a finite or infinite sequence $q_0, q_1, \ldots$ such that $q_0 \in Q_0$ and $(q_i, q_{i+1}) \in D$

Example: Traffic light automaton



- $Q = \{G, Y, R\}$ $Q_0 = \{G\}$
- $D = \{(G, Y), (Y, R), (R, G)\}$

Execution of traffic light G, Y, R, G, Y, R … infinite even though finite state

Infinite State Automata and Testing

Example 2

- $Q = \mathbb{N}\ Q_0 = \{n_0\}$
- $\left(n, \frac{n}{2}\right) \in D$ if n is even $(n, 3n+1) \in D$ if n is odd

Deterministic automaton because each state $q \in Q$ has a unique next state

Deterministic automata have a single execution

Execution from $n_0 = 6$ is 6, 3, 10, 5, 16, 8, 4, 2, 1, 4, 2, 1 …

Execution from $n_0 =$ 7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1, . . .

Testing question. For any n_0 does the execution end in the 4-2-1 loop?

This is a well-known open problem in mathematics called the Collatz conjecture

Testing and verification can be hard for simple deterministic automata

Requirements and Counter-examples

Requirements define what the system must and must not do

Example: “Car stays within speed limit”

Autonomous car: “Ego should not collide with lead car”

Collatz: “Every number eventually ends in the 4-2-1 cycle”

A requirement defines a set ***R*** of allowed executions

An execution α that is not in the set ***R*** is a *counter-example*

$R_{\text{eventually-1}} = \{\alpha \mid \exists k \; \alpha_k = 1\}$

An automaton A satisfies a requirement R if ***all*** executions of A satisfies R

Whether the Collatz automaton satisfies the requirement $R_{\text{eventually-1}}$ for all initial conditions remains an open problem, although no counter-example has been found up to 2^{70}

This is an example of a verification problem

Verification problem

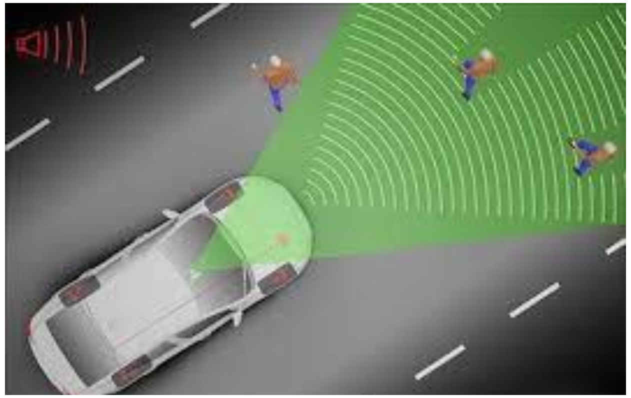
Verification problem: Given an automaton A and a requirement R, check whether all executions of A satisfy R or find a counter-example

Testing or checking individual executions can help find counter-examples but cannot show that there is no counter-example

Verification can be hard because

- |Q| is finite but large and testing may require visiting all the states (e.g., Collatz)
- |Q| is small but the number of executions is very large
- |Q| may be infinite and D may be nondeterministic --- typical in autonomy

Example: Automatic Emergency Braking (AEB)



Car must brake to maintain safe gap with lead vehicle/pedestrian

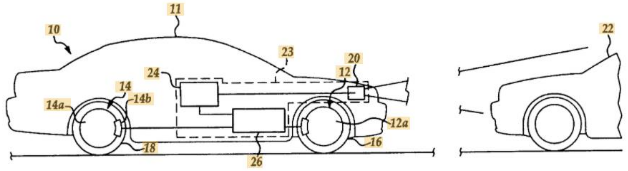


Figure 1

There is no standard for checking correctness of AEB

www.google.com › patents
US20110168504A1 - Emergency braking system - Google ...
Jump to **Patent** citations (18) - US4053026A * 1975-12-09 1977-10-11 Nissan Motor Co., Ltd. Logic circuit for an **automatic braking** system for a motor ...

www.google.com › patents
US5170858A - Automatic braking apparatus with ultrasonic ...
An automatic braking apparatus includes: an ultrasonic wave emitter provided in a ... Info: **Patent** citations (13); Cited by (7); Legal events; Similar documents; Priority and ... US6523912B1 2003-02-25 **Autonomous emergency braking** system.

Future: Every code commit in github from an AEB engineer, **proves a theorem** establishing A satisfies R_{gap}

www.google.com › patents
DE102004030994A1 - Brake assistant for motor vehicles ...
B60T7/22 **Brake**-action initiating means for **automatic** initiation; for initiation not ... Info: **Patent** citations (3); Cited by (9); Legal events; Similar documents ... data from the environment sensor and then **automatically** initiates **emergency braking**.

www.google.com.pg › patents
Braking control system for vehicle - Google Patents
An **automatic emergency braking** system for a vehicle includes a forward viewing camera and a control. At least in part responsive to processing of captured ...

www.automotiveworld.com › news-releases › toyota-ip...
Toyota IP Solutions and IUPUI issue first commercial license ...
Jul 22, 2020 - ... and validation of automotive **automatic emergency braking** (AEB) ... and Director of **Patent** Licensing for Toyota Motor North America. "We are ...

insurancenewsnet.com › oarticle › patent-application-tit...
Patent Application Titled "Multiple-Stage Collision Avoidance ...
Apr 3, 2019 - No assignee for this **patent** application has been made. ... **Automatic emergency braking** systems will similarly, also, soon be required for tractor ...

Automaton model of AEB

Automaton $A = \langle Q, Q_0, D \rangle$

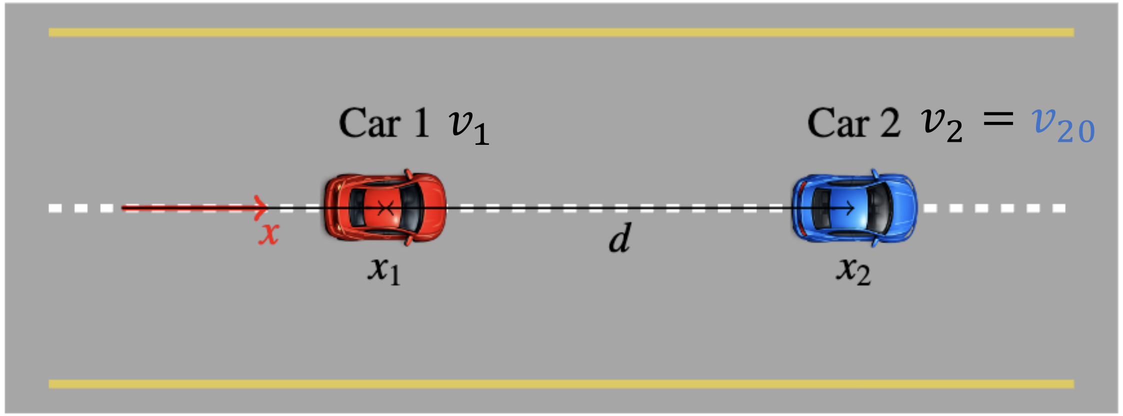
- $Q: [x_1, x_2, v_1] \in \mathbb{R}^3$
- $Q_0 = \{[x_1 = x_{10}, x_2 = x_{20}, v_1 = v_{10}]\}$
- $D \subseteq Q \times Q$ written as a program

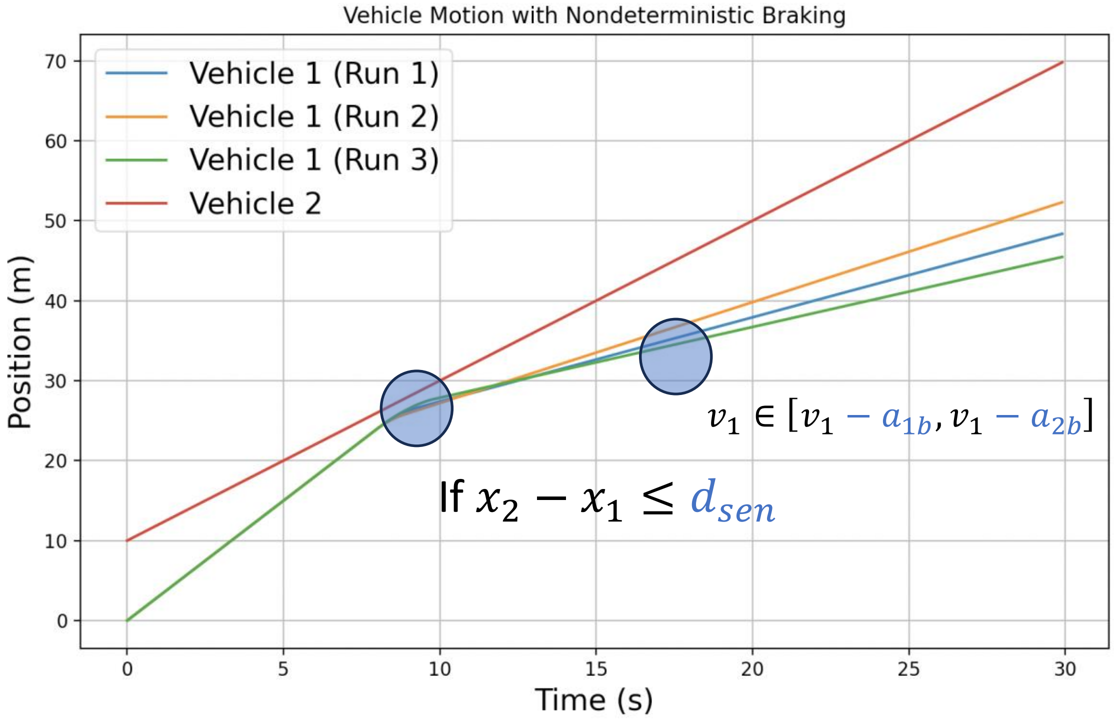
  If $x_2 - x_1 \le d_{sen}$

  $v_1 \in [v_1 - a_{1b}, v_1 - a_{2b}]$

  $x_2 = x_2 + v_2$

  $x_1 = x_1 + v_1$





Automaton model of AEB

Automaton $A = \langle Q, Q_0, D \rangle$

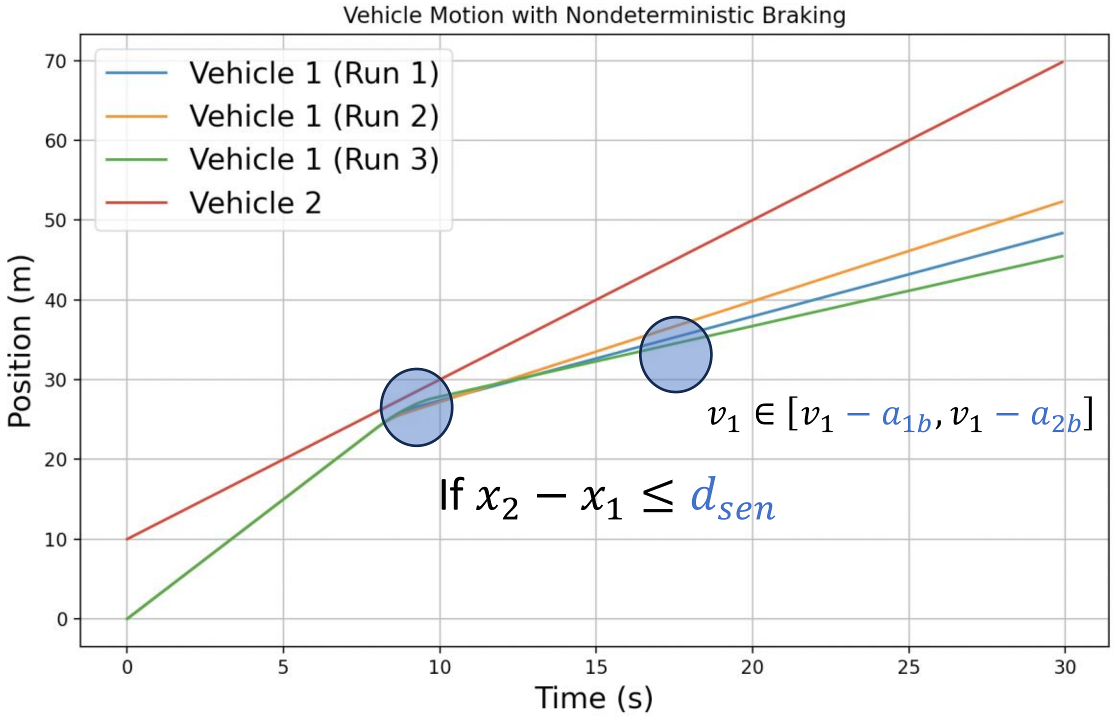
- $Q : \mathbb{R}^3$; $\boldsymbol{q} \in Q$ $\boldsymbol{q}.x_1, \boldsymbol{q}.x_2 \in \mathbb{R}$
- $Q_0 = \{\boldsymbol{q} \mid [\boldsymbol{q}.x_1 = x_{10}, \boldsymbol{q}.x_2 = x_{20}, \boldsymbol{q}.v_1 = v_{10}]\}$
- $(\boldsymbol{q}, \boldsymbol{q}') \in D$ iff

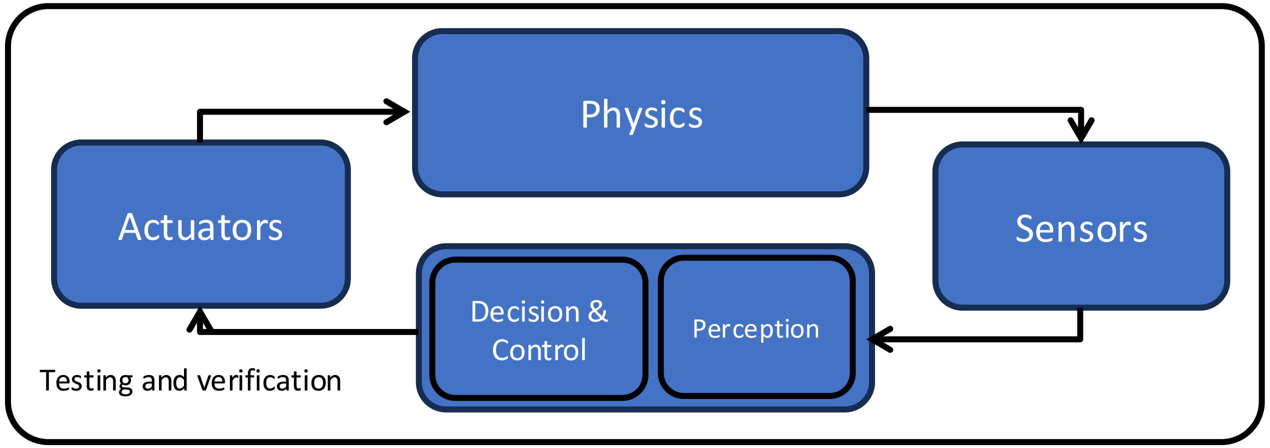
  If $\boldsymbol{q}.x_2 - \boldsymbol{q}.x_1 \leq d_{sen}$

  $\boldsymbol{q}'.v_1 \in [\boldsymbol{q}.v_1 - a_{1b}, \boldsymbol{q}.v_1 - a_{2b}]$

  $\boldsymbol{q}'.x_2 = \boldsymbol{q}.x_2 + \boldsymbol{q}.v_2$

  $\boldsymbol{q}'.x_1 = \boldsymbol{q}.x_1 + \boldsymbol{q}.v_1$





What did we miss in the AEB model?

If $x_2 - x_1 \leq 2.0$

  $v_1 \in [v_1 - a_{1b}, v_1 - a_{2b}]$

else $v_1 = v_1$

$x_2 = x_2 + v_2$

$x_1 = x_1 + v_1$

- Acceleration, friction in dynamics
- Uncertainty in sensing
- Uncertainty in lead vehicle behavior
- Rear vehicle

“All models are wrong, some are useful.”

Safety and liveness requirements

$R_{gap} = \{\alpha \mid \forall i\ \alpha_i.x_2 > \alpha_i.x_1\}$ non-zero gap $U_{gap} = \{\boldsymbol{q} | \boldsymbol{q}.x_2 - \boldsymbol{q}.x_1 \leq 0\}$

$R_{sp-lim} = \{\alpha \mid \forall i\ \alpha_i.v_1 \leq 70\}$ speed limit $U_{sp-lim} = \{\boldsymbol{q} | \boldsymbol{q}.x_1 \geq 70\}$

$R_{catch-up} = \{\alpha \mid \exists i\ 2 > \alpha_i.x_2 - \alpha_i.x_1 > 1\}$ catch eventually

A safety requirement is a requitement that every states along all executions should stay in certain good states

Equivalently, a safety requirement says that no execution of A ever reaches a bad or unsafe states

R_{gap} and R_{sp-lim} are examples of safety requirements with U_{gap} and U_{sp-lim} as the corresponding unsafe sets

$R_{catch-up}$ is not a safety requirement; it is an example of a liveness / progress requirement

A liveness requirement says that along every execution eventually some good state is reached

Summary

- Testing alone is inadequate---in theory and practice
- Automaton executions define system behaviors
- Requirements define desired or correct behaviors
- Verification proves that automaton satisfies requirements or finds counter-examples
- Safety requirements say that *no execution ever* hits unsafe states
- Next: Verification of safety requirements