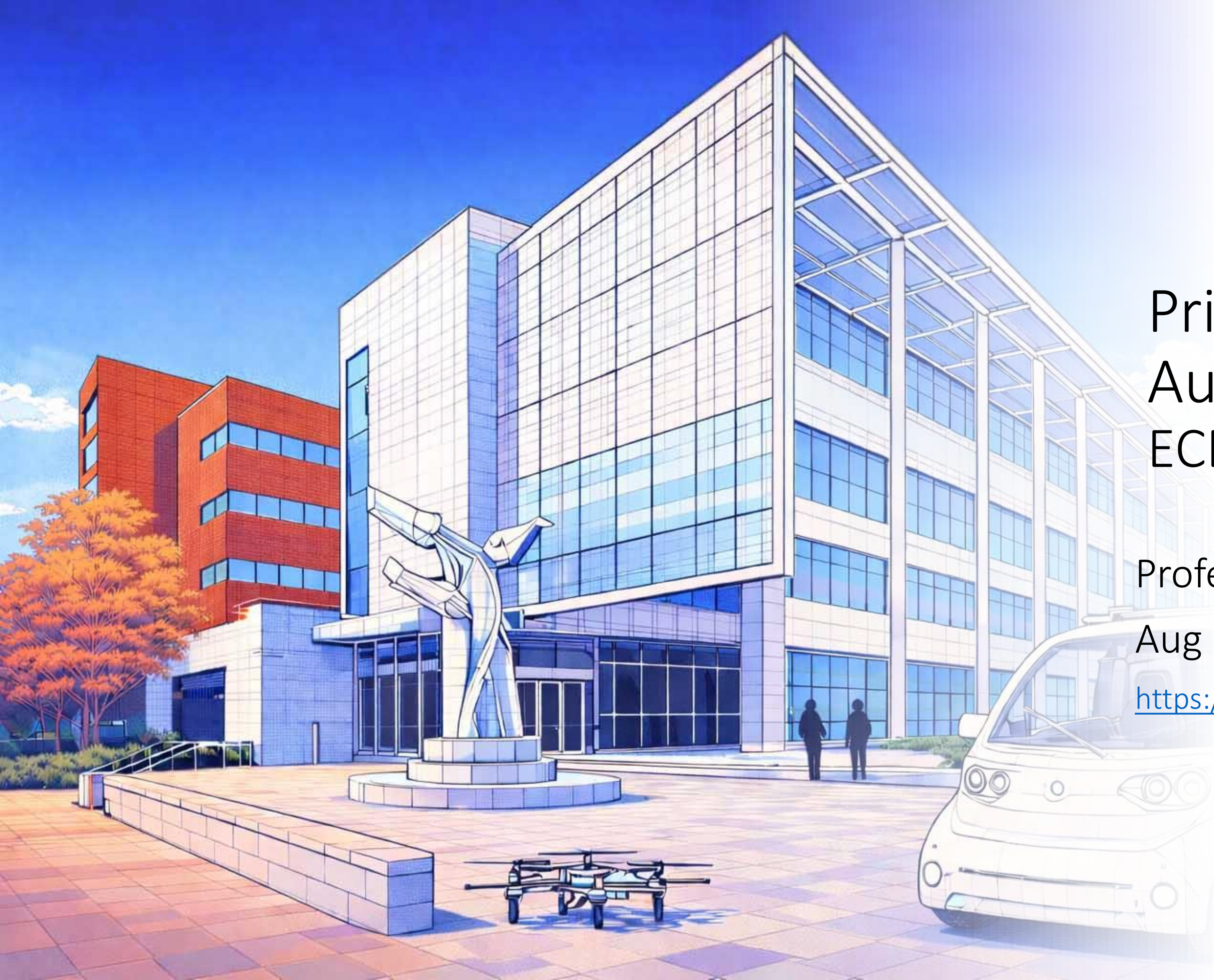


Principles of Safe Autonomy
ECE 484 Fall 2026

Professor Huan Zhang

Aug 25, 2026

https://publish.illinois.edu/safe-autonomy/

Plan for today

- Motivation
- Administrivia
- Introduction to Safety

Motivation

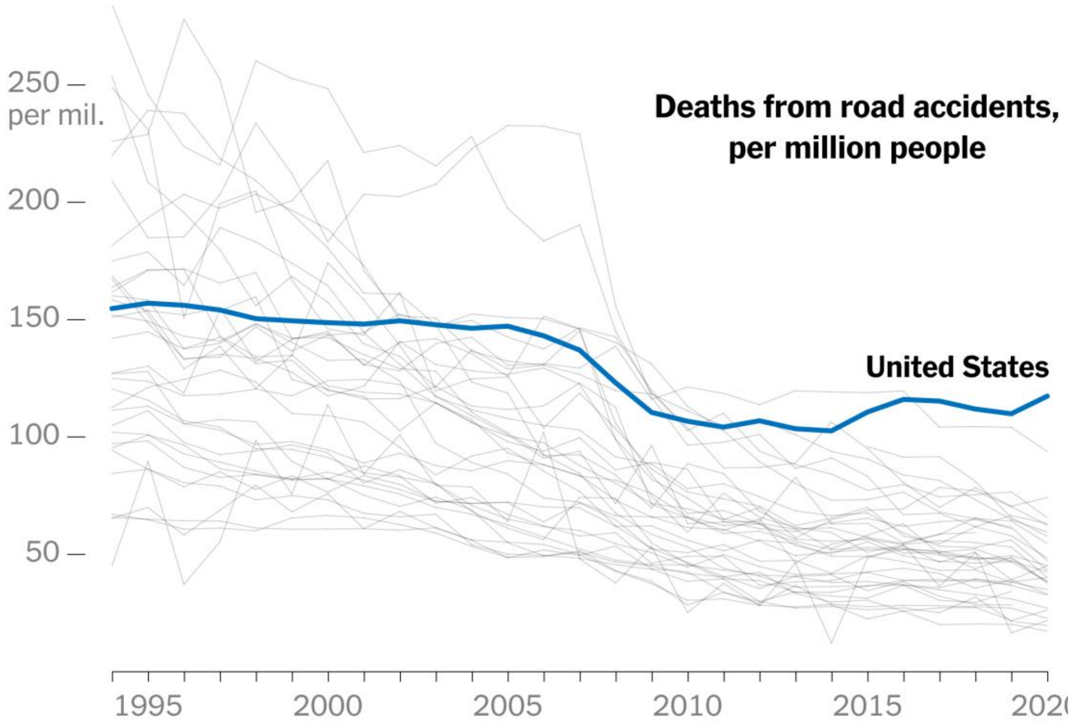
What are Autonomous systems?

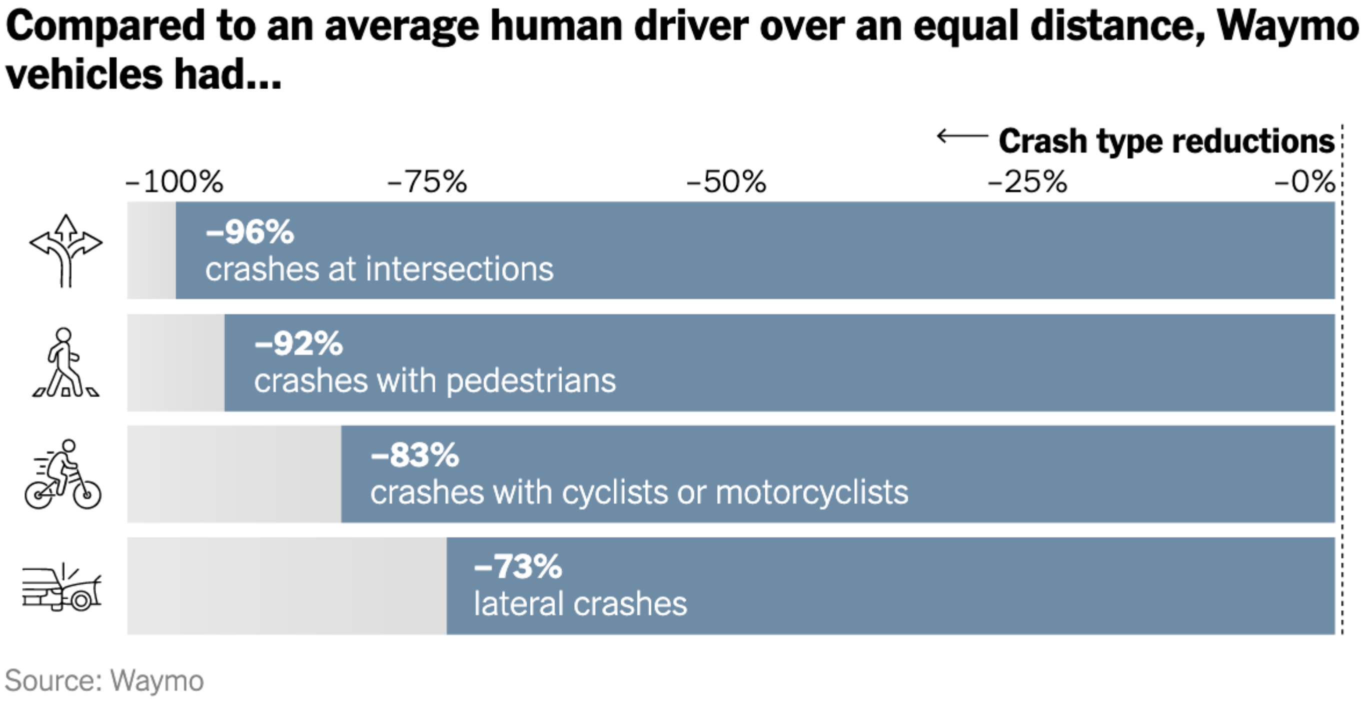
Why do we want to study them?

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

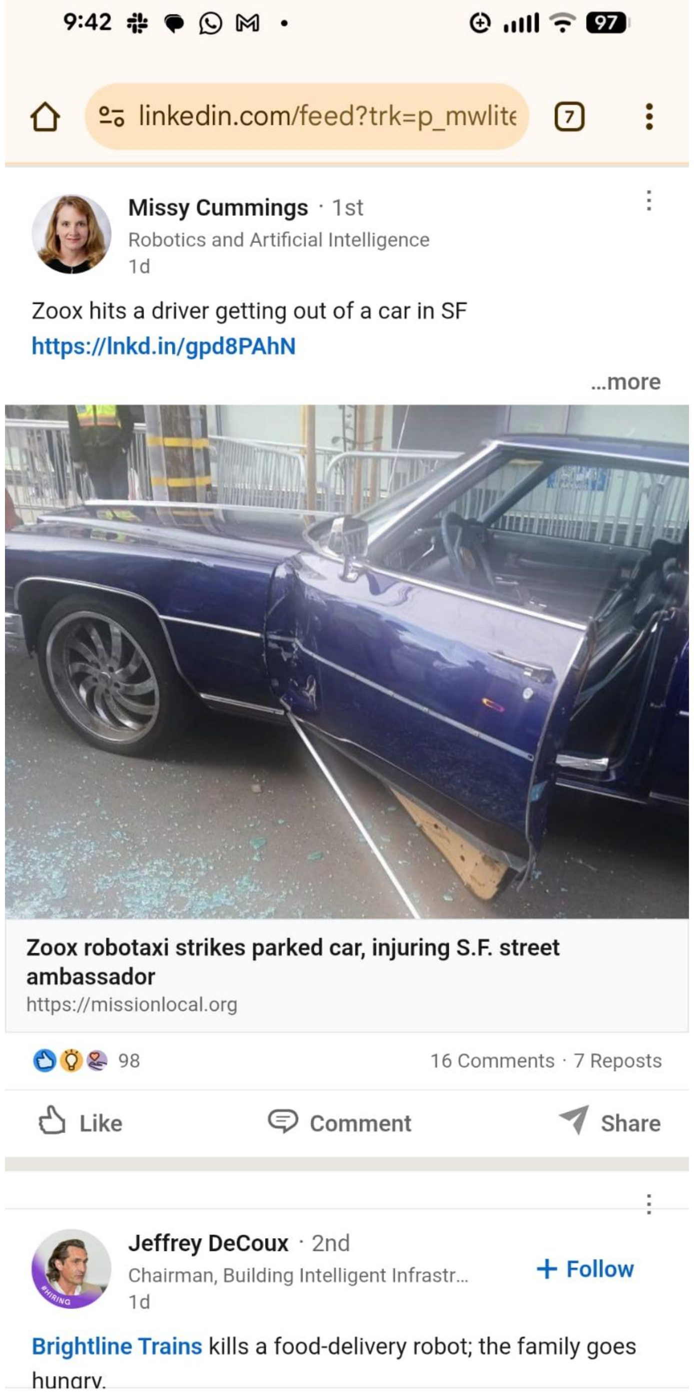
Autonomy Is a Frontier of Engineering

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:

- **Safety & Reliability** — 90% is a demo. 99.99% is a product.
  - Edge-cases, long-tail uncertainty, architecture & reasoning
- **Cost** — sensing, compute, testing, and deployment at scale
- **Energy** — perception & learning on resource-limited machines

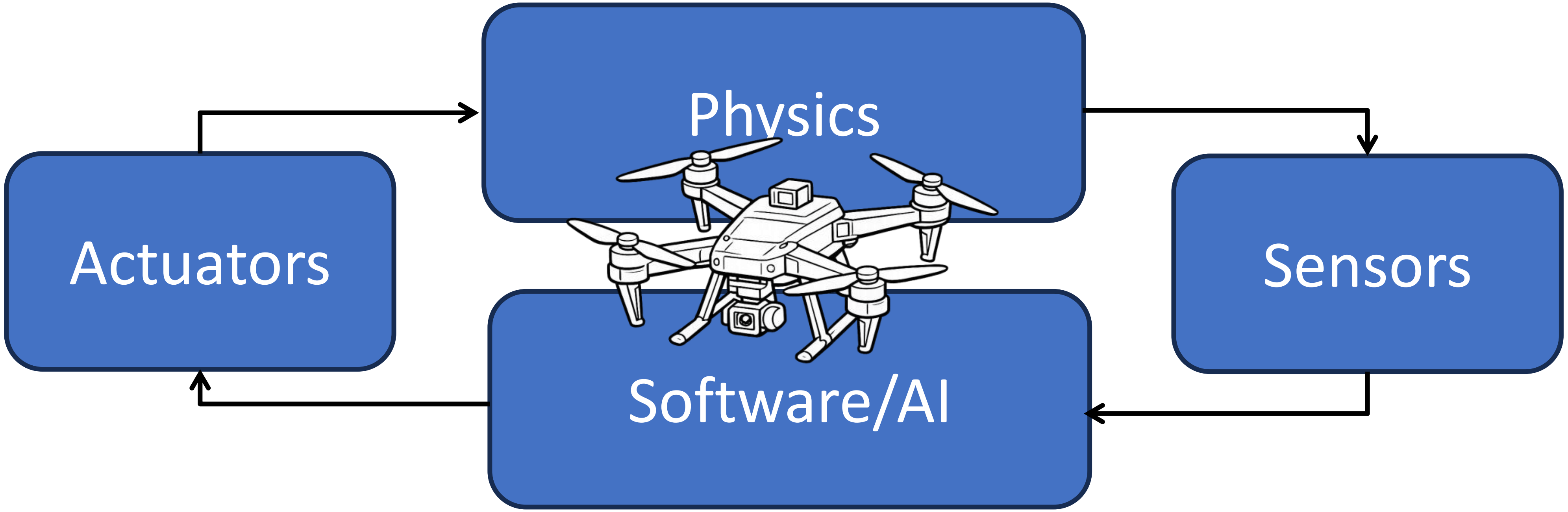
They are **fundamental engineering challenges**.



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.

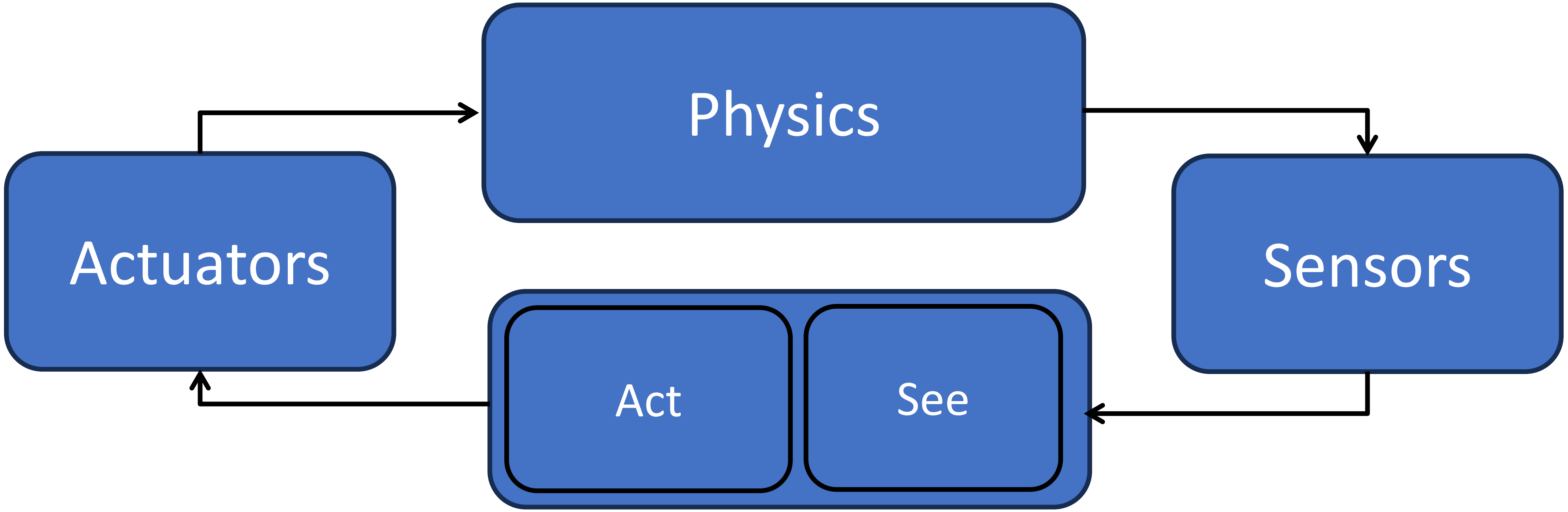
ECE484 develops the foundational principles behind autonomy



Principles of Safe Autonomy ECE 484

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

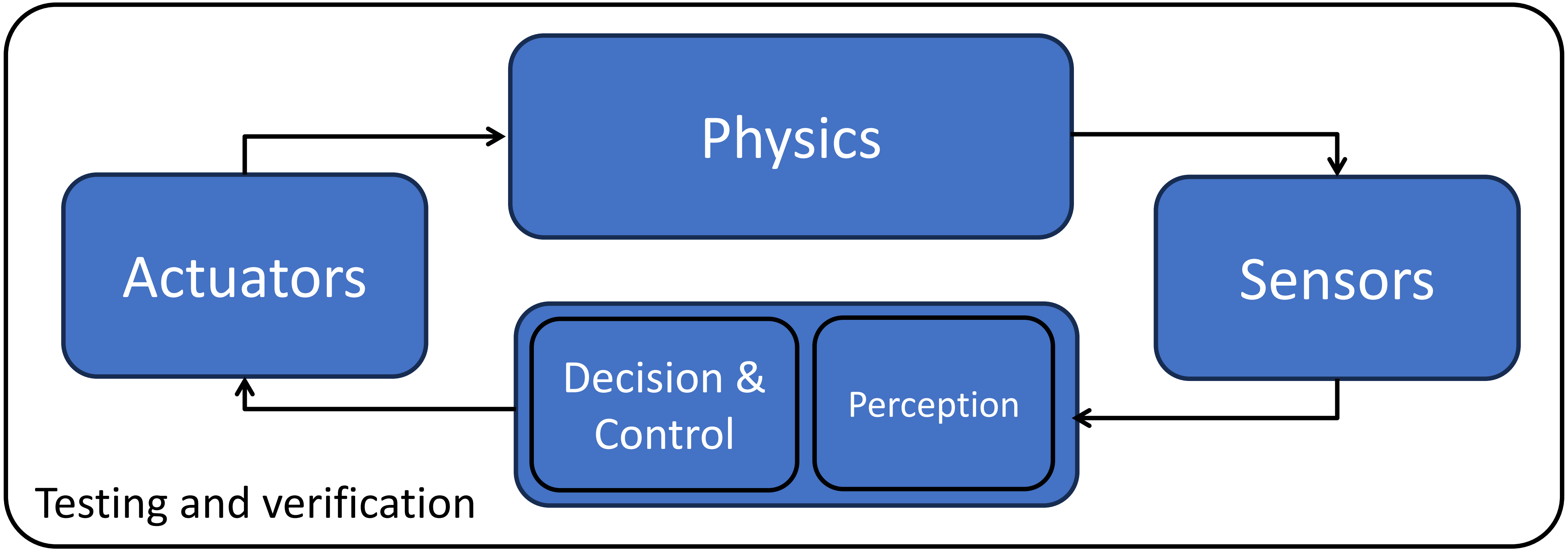
ECE484 develops the foundational principles behind autonomy



Principles of Safe Autonomy ECE 484

Autonomy is a frontier 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

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

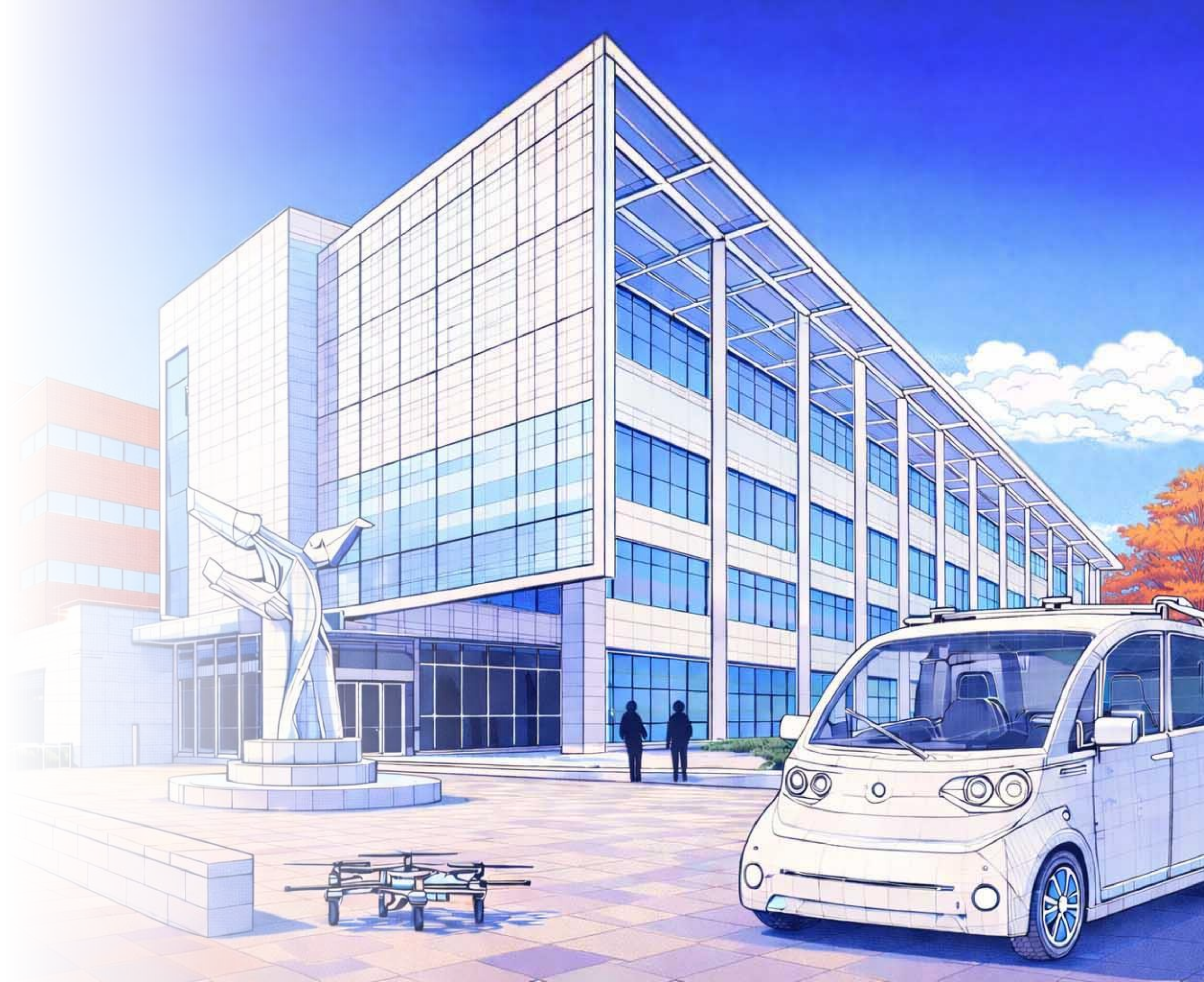
- **Sensing**, **Perception** & **State Estimation**
  - Turning raw sensor bits into state information
- **Planning**, **Decision-Making** & **Control**
  - Computing actions that are safe, robust, and effective based on state estimates
- **Testing**, **Evaluation** & **Formal Verification**
  - Reasoning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

Machine Learning / Neural Networks 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://publish.illinois.edu/safe-autonomy/

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

Campuswire for announcements; best effort response delay ~2 days.

- Discussions, forming teams, occasional polls, feedback
- https://campuswire.com/c/G70BCCBCC/

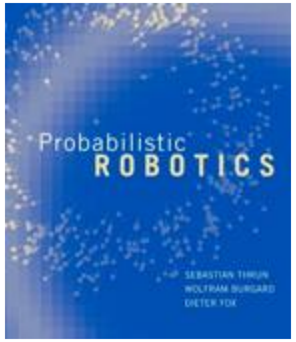
Canvas: https://canvas.illinois.edu/courses/74528

Grade release and assignment submission

Course materials

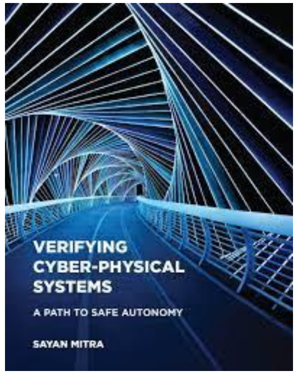
Primary sources

- 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 **8/28** ECEB5072) walkthroughs and demos
  - Office hours: see course website
  - MP0 will be individual **(starting 8/28),** 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

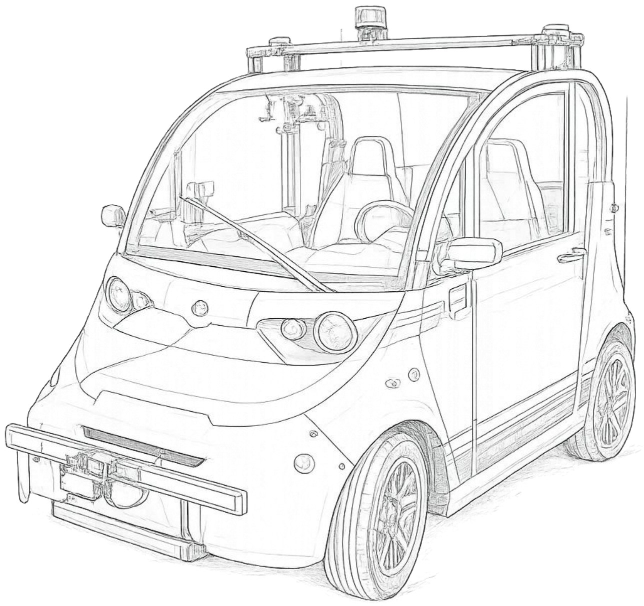
Tentative grade boundaries	
A	>90
B	>80
C	>65
D	>55

Examples of what you will build in 484

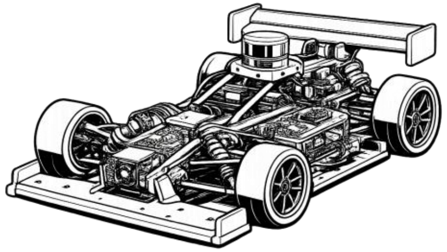
Quadrotor Projects from Fall 2025

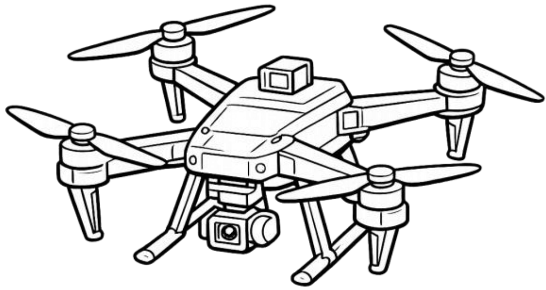
- https://www.youtube.com/watch?v=8l80orgLiXs
- https://www.youtube.com/watch?v=pHygPWMMgdw

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

Course schedule, slides

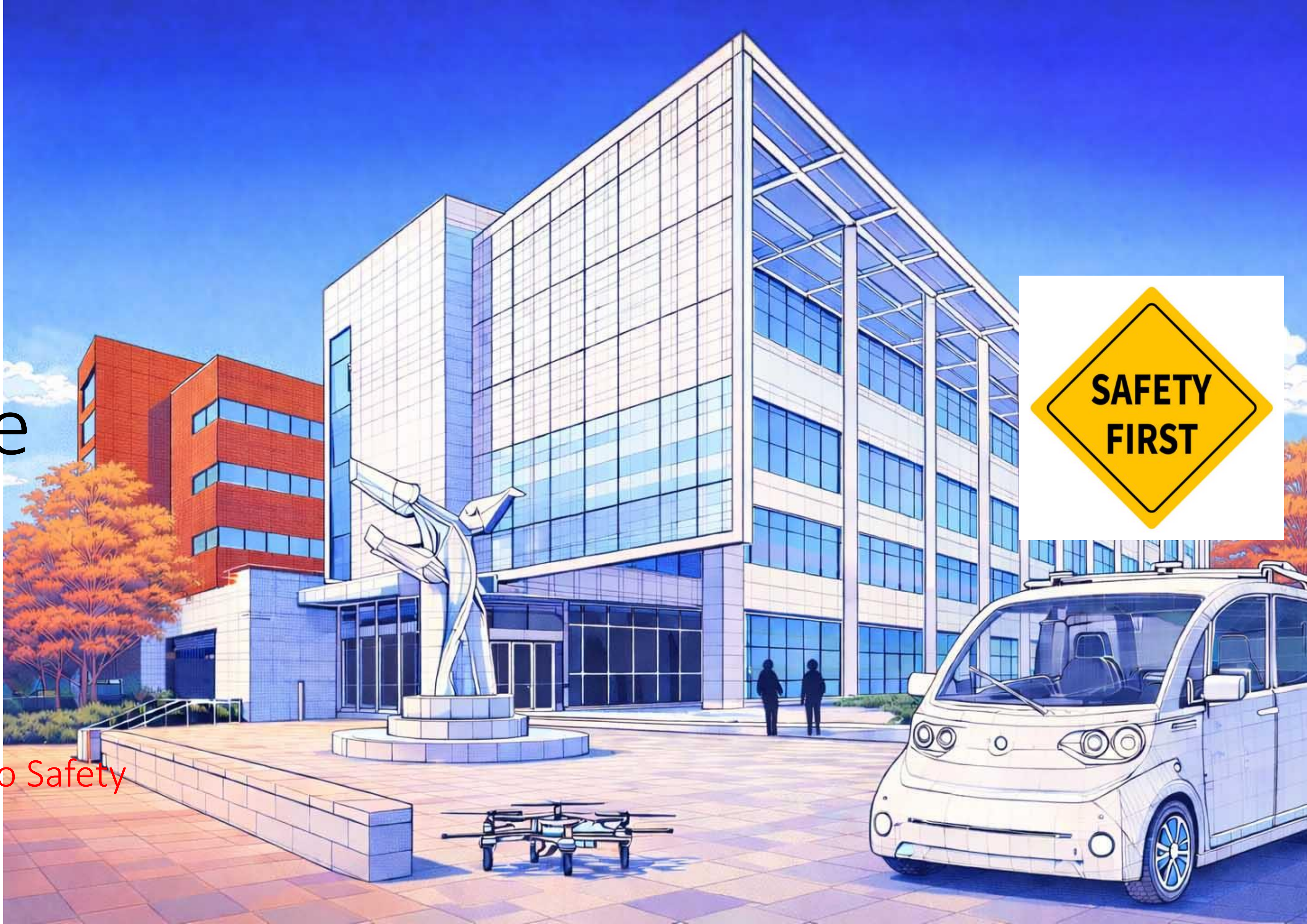
https://publish.illinois.edu/safe-autonomy/schedule-fall-2026/

Outline

Motivation

Administrivia

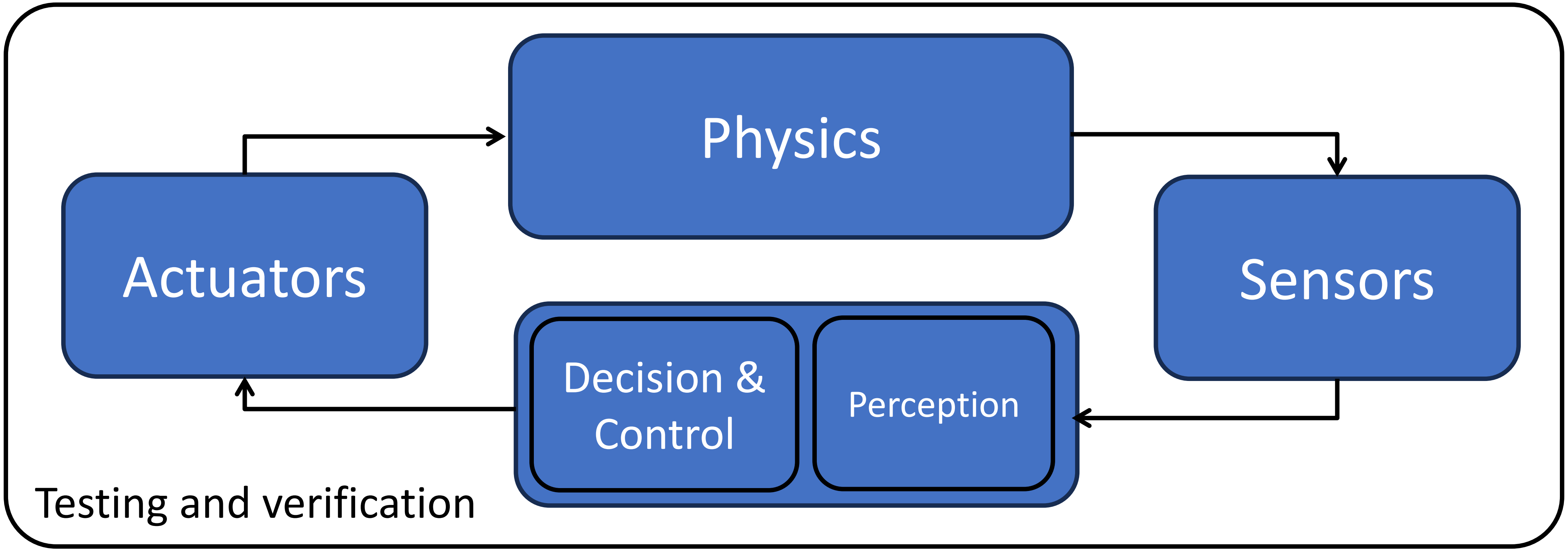
Introduction to Safety



Principles of Safe Autonomy ECE 484

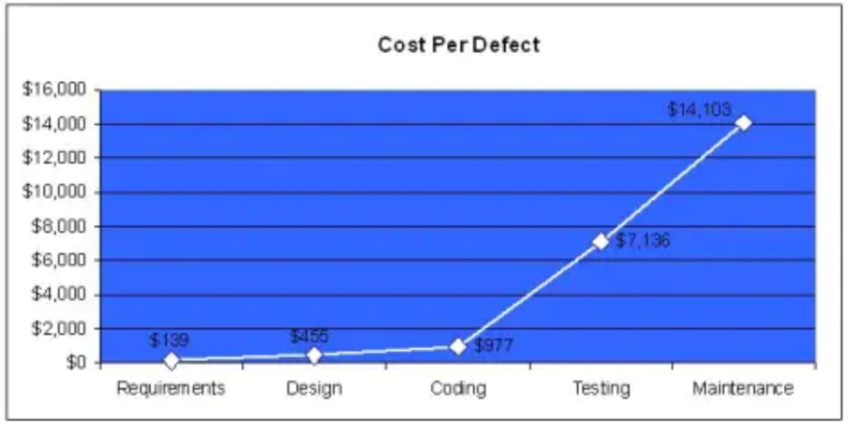
Autonomy is a frontier where perception, learning, control, and verification meet the real world and ECE484 is where you begin to shape it.

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 (1985 - 1987).
- Elaine Herzberg was killed by selfdriving Uber prototype in Tempe, Arizona in March 2018.
- A simple data conversion error (overflow) caused the **$500M Ariane 5 rocket** to veer off course and explode shortly after launch (June 4, 1996).
- GM’s Cruise autonomous vehicle unit shut down parts of its San Francisco robotaxi fleet after crashes in 2023.
- Cost of defects grow exponentially with the time of discovery



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

Checking truthfulness of statements about reliability and safety

- A popular method for checking truth: Statistical **testing**
- "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 known to be 10^{-6}
  - Assume that for AV 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 have to 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)

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 being 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/systems, hypervisors, boot-loaders/BIOS/firmware, garbage collectors, and network designs.” Byron Cook, Amazon

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

Formal Verification in Software: an example

Simple programming task: given a 32-bit unsigned integer, calculate how many bits are set to 1 (“population count”)

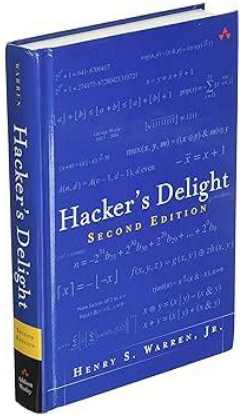
Naive implementation

```c
int popcount(uint32_t x) {
  int c = 0;
  for (int i = 0; i < 32; i++) {
    c += x & 1;
    x >>= 1;
  }
  return c;
}
```

Clever implementation

```c
int popcount (uint32_t x) {
    x = x - ((x >> 1) & 0x55555555);
    x = (x & 0x33333333) + ((x >> 2) & 0x33333333);
    x = ((x + (x >> 4) & 0xf0f0f0f) * 0x1010101) >> 24;
    return x;
}
```

Example source: Marijn J.H. Heule, “SAT and SMT Solvers in Practice”



Formal Verification in Software: an example

Can we trust this “clever implementation” of the same function?

What would you do to ensure this clever implementation is correct? Brute-force?

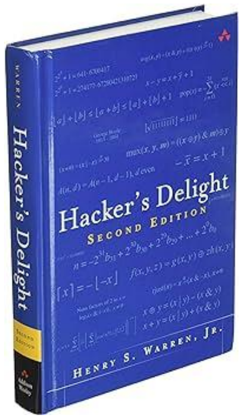
Naive implementation ?= Clever implementation

Naive implementation

```
int popcount(uint32_t x) {
  int c = 0;
  for (int i = 0; i < 32; i++) {
    c += x & 1;
    x >>= 1;
  }
  return c;
}
```

Clever implementation

```
int popcount (uint32_t x) {
    x = x - ((x >> 1) & 0x55555555);
    x = (x & 0x33333333) + ((x >> 2) & 0x33333333);
    x = ((x + (x >> 4) & 0xf0f0f0f) * 0x1010101) >> 24;
    return x;
}
```



Formal Verification in Software: Specification / Requirement

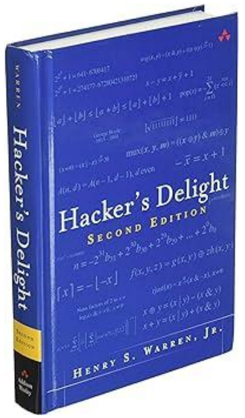
Formal verification aims to prove that **for all possible inputs,** the results of the two functions are the same (mathematically, same integer is returned)

Verification **specification** or **requirement**:

Naive implementation **==** Clever implementation

```
int popcount(uint32_t x) {
  int c = 0;
  for (int i = 0; i < 32; i++) {
    c += x & 1;
    x >>= 1;
  }
  return c;
}
```

```
int popcount (uint32_t x) {
    x = x - ((x >> 1) & 0x55555555);
    x = (x & 0x33333333) + ((x >> 2) & 0x33333333);
    x = ((x + (x >> 4) & 0xf0f0f0f) * 0x1010101) >> 24;
    return x;
}
```



Example: Automatic Emergency Braking (AEB)

A car moving down a straight road has to detect any pedestrian (or another car) in front and stop before it collides.

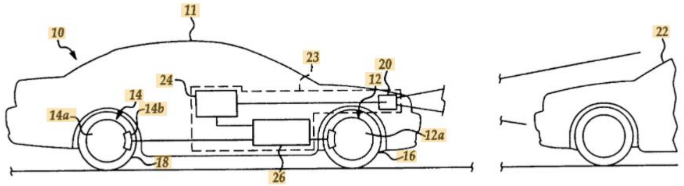


Figure 1

Today: There is no standard for checking correctness of AEB systems

Future: Every time an AEB engineer commits code in github, a theorem proves safety of the system under appropriate assumptions, or finds an unsafe scenario

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.

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 ...

Form your team now! Lab starts this Friday

- In groups: Form your group of 3-4 **now**! Create groups by this Friday (more details on Campuswire)
- Each MP will build a significant component of an autonomous system over ~**2 weeks**
- Use your computer with **Ubuntu 24.04** or ECEB5072 lab computers
- TAs and LAs will run live labs in ECEB 5072
- MP walkthough, setup, bridge the lecture and the assignments
- **MP0+HW0 will be released this Friday (8/28), labs starts this Friday**
  - MP0 and all HWs are individual; MP1 – 3 are teamwork
  - Your entire group has to attend one lab for the MP walkthrough
  - And 1 lab after the MP is due (to demo your work).