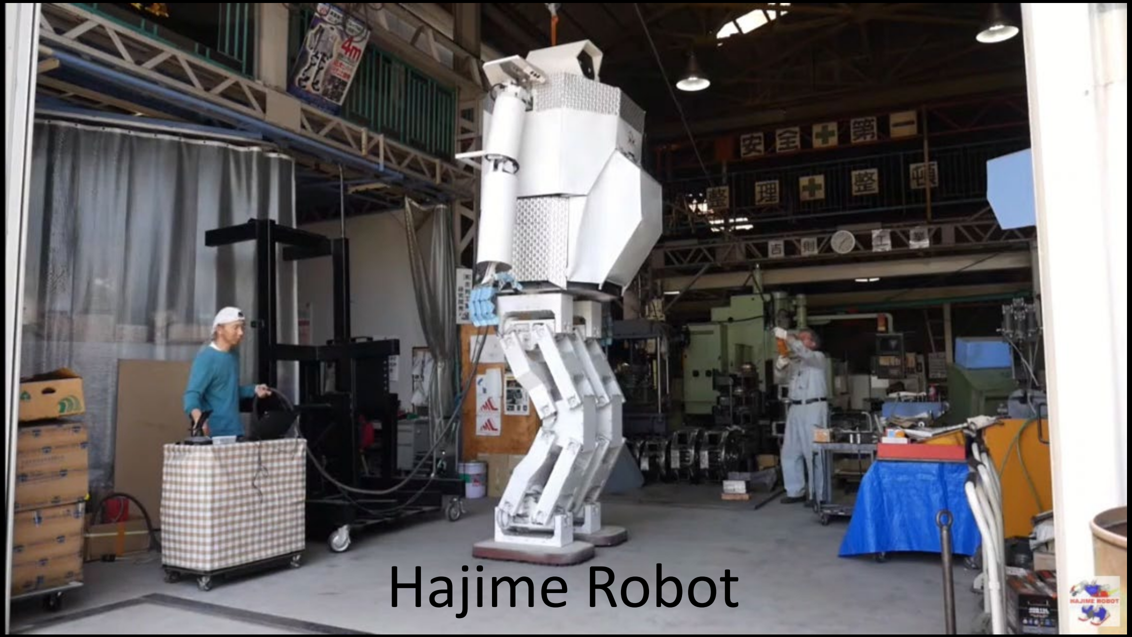
ECE 4960

Prof. Kirstin Hagelskjær Petersen
kirstin@cornell.edu

Fast Robots

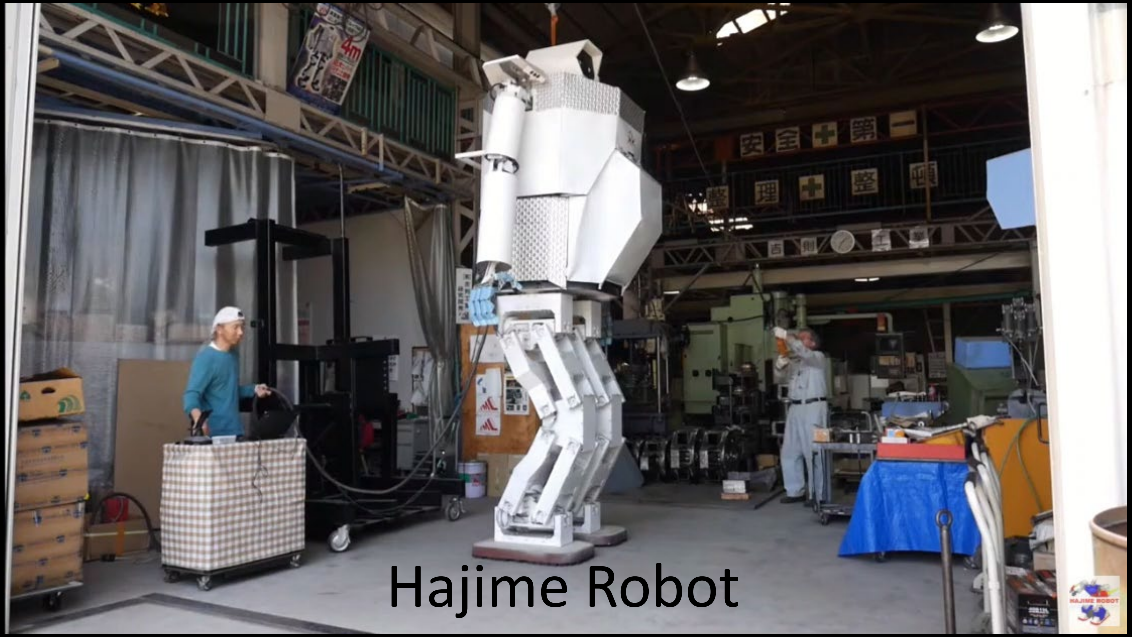
Why *this* class?

- Fast robots are fundamentally different from slow robots



Why *this* class?

- Fast robots are fundamentally different from slow robots
  - Kinematics – Dynamics



Hajime Robot

Why *this* class?

- Fast robots are fundamentally different from slow robots
  - Kinematics – Dynamics
  - Stable – Unstable


Cubli, ETH Zurich

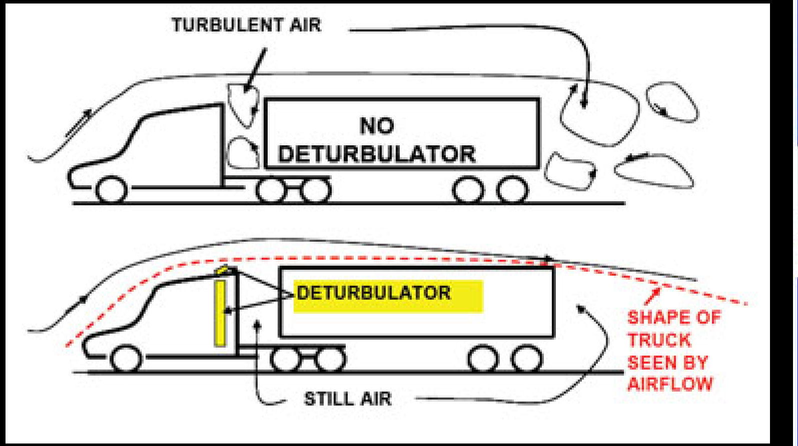
Why *this* class?

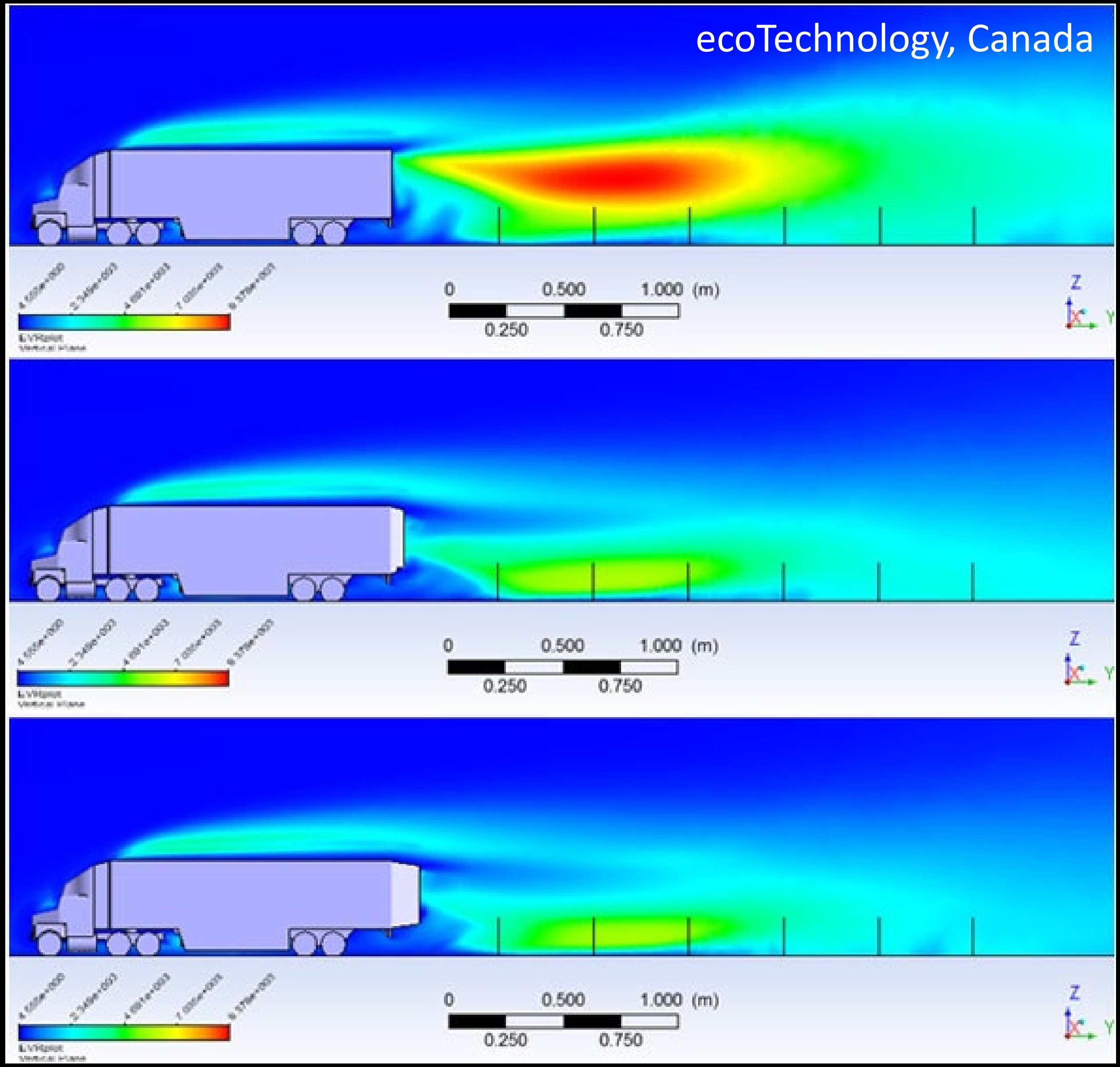
- Fast robots are fundamentally different from slow robots
  - Kinematics – Dynamics
  - Stable – Unstable
- Design for fast robots goes beyond just good control theory and dynamic models
  - Practical implementation, mechanics, sensors, processing, estimation, etc.

Control and its implications in fast robots

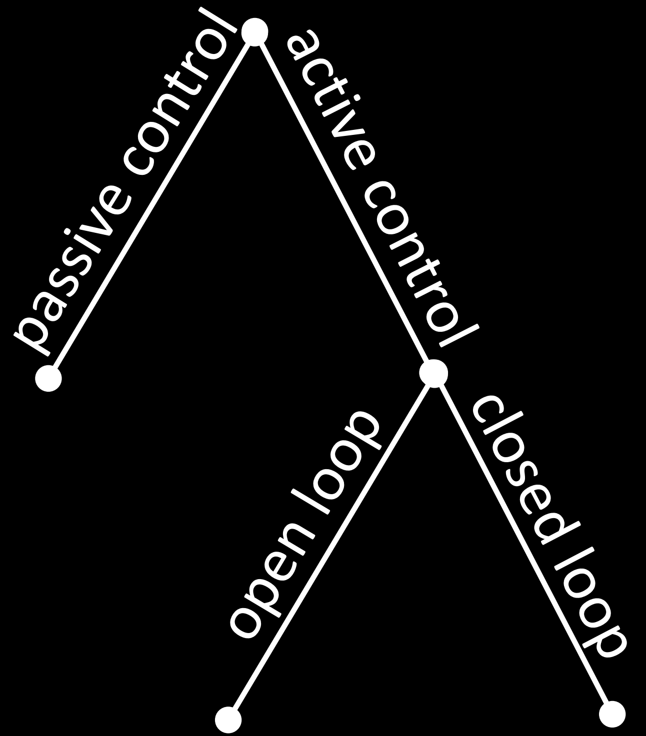
passive control

active control





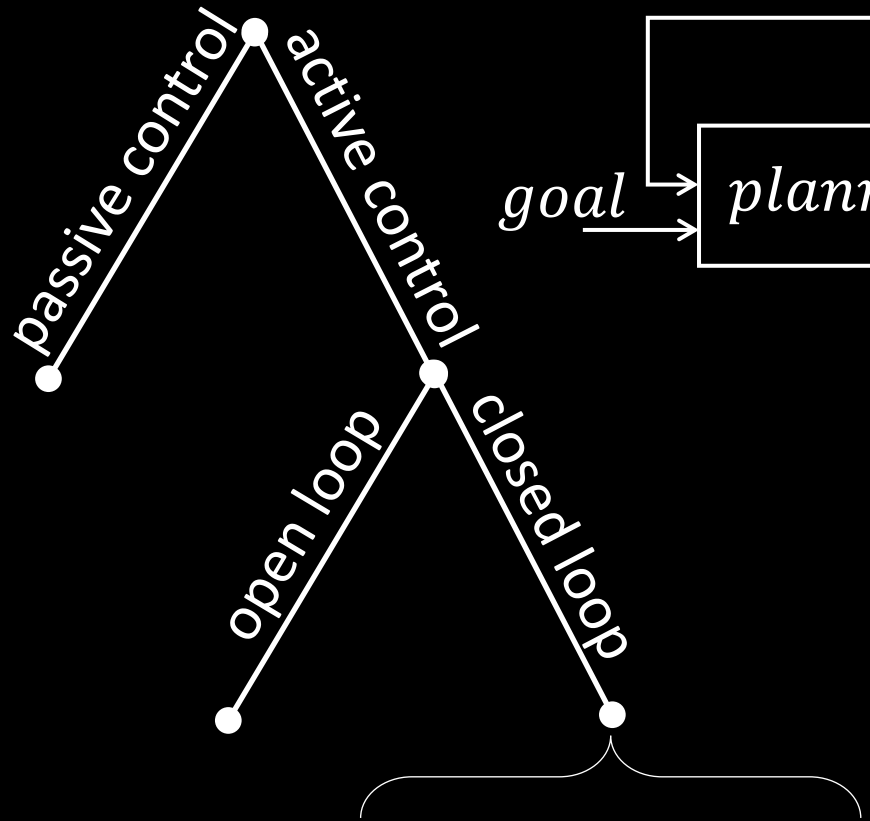
Control and its implications in fast robots



Why feedback?

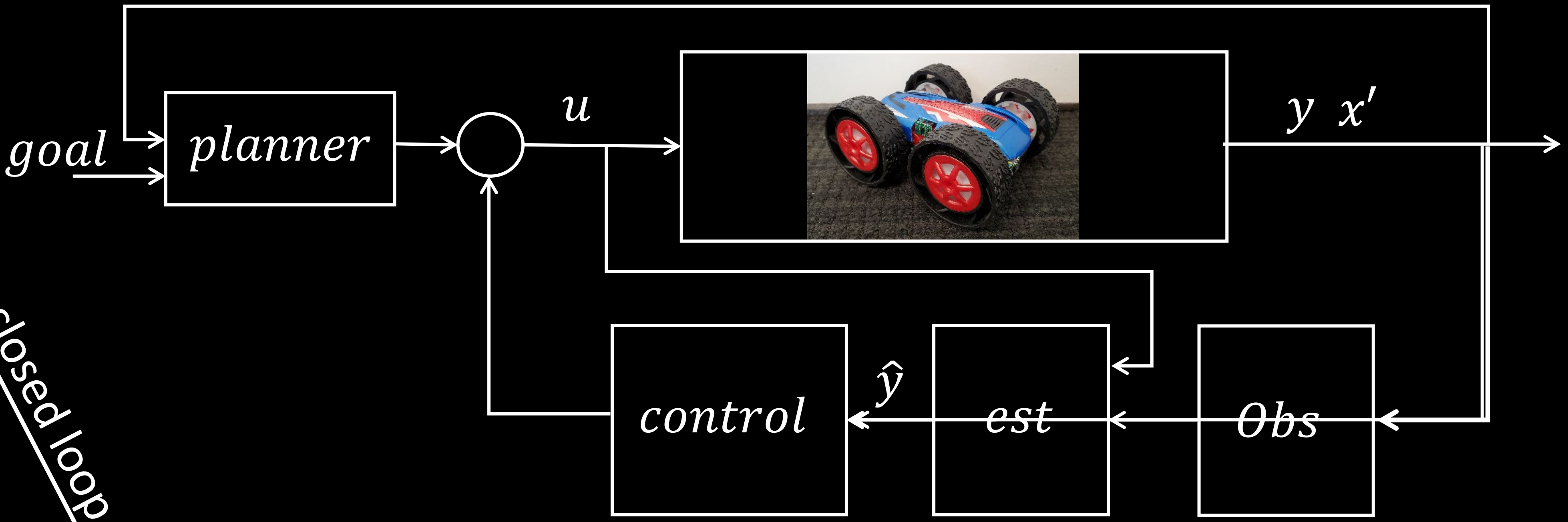
- System uncertainty
- Instability
- Disturbances
- Efficiency

Control and its implications in fast robots



- processor
- drivers
- limits

- sensors
- noise/bias



Why feedback?

- System uncertainty
- Instability
- Disturbances
- Efficiency

Fast Robots
Class Layout

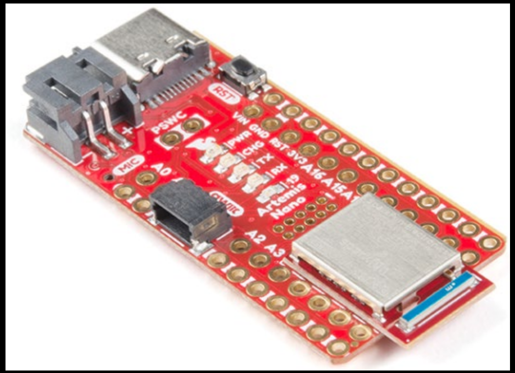
Part 1: Implement and demo a Stunt Robot!

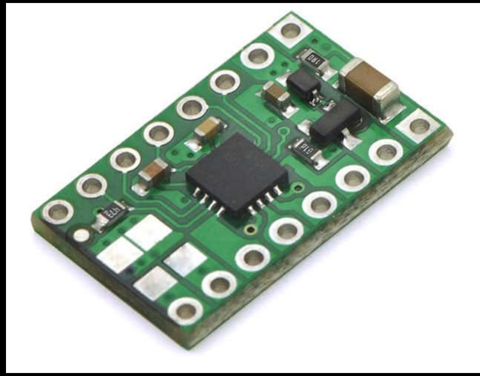
Part 1: Implement and demo a Stunt Robot!

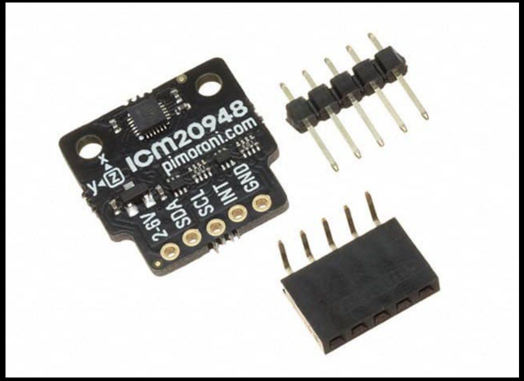
- Combine base with processor, drivers, and sensors
- Refresh on linear algebra and T-matrices
- Sensor modalities and types of sensors
- Actuators, drivers, circuits and routing, and EMI

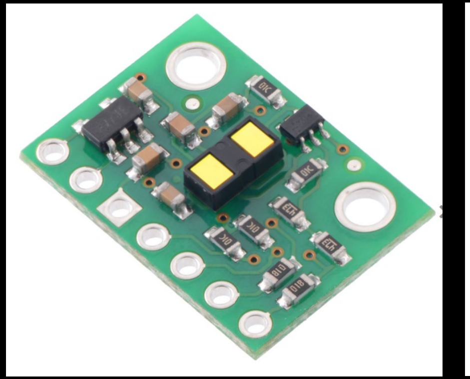
ASML

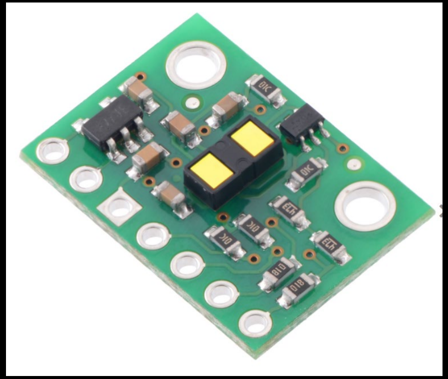
- $130 lab kit
- Sponsored entirely by *ASML!*





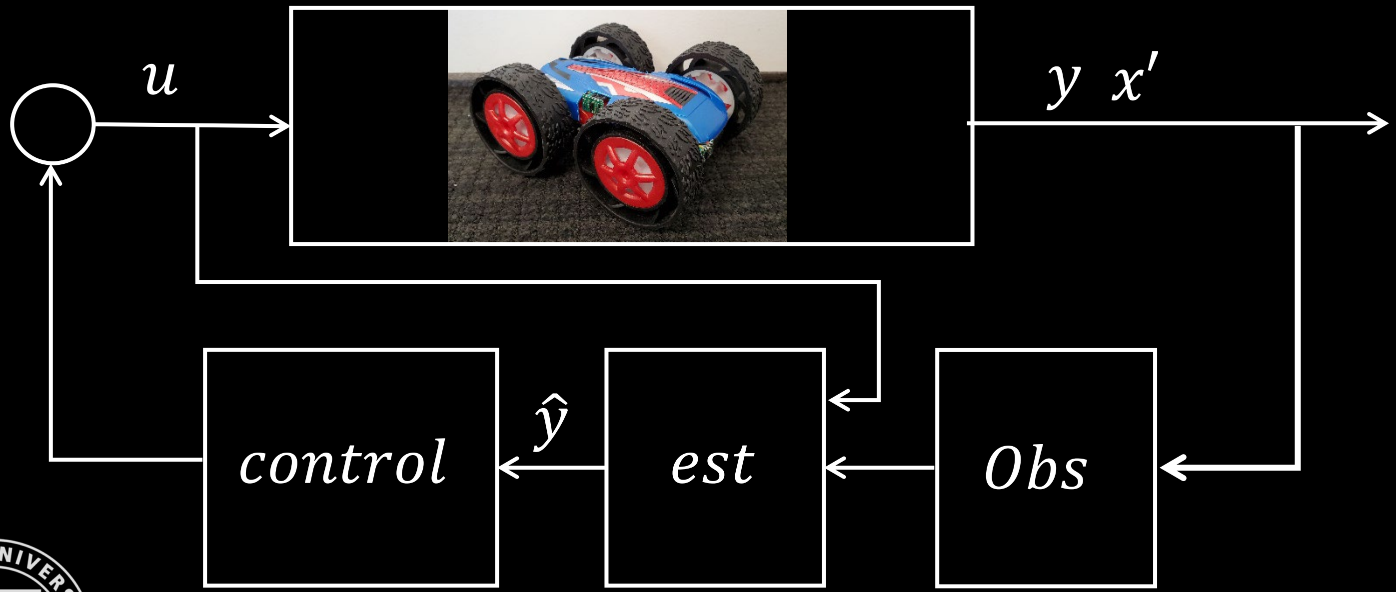






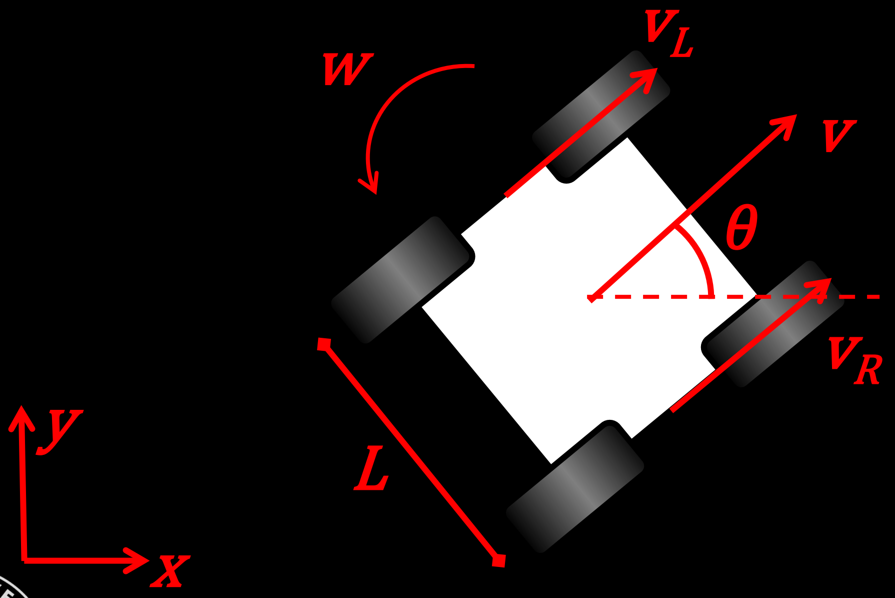
Part 1: Implement and demo a Stunt Robot!

- Combine base with processor, drivers, and sensors
- Refresh on linear algebra and T-matrices
- Sensor modalities and types of sensors
- Actuators, drivers, circuits and routing, and EMI
- Linear systems, model-free and model-based control



Part 1: Implement and demo a Stunt Robot!

- Combine base with processor, drivers, and sensors
- Refresh on linear algebra and T-matrices
- Sensor modalities and types of sensors
- Actuators, drivers, circuits and routing, and EMI
- Linear systems, model-free and model-based control
  - PID controllers, Control theory, LQG control, KF

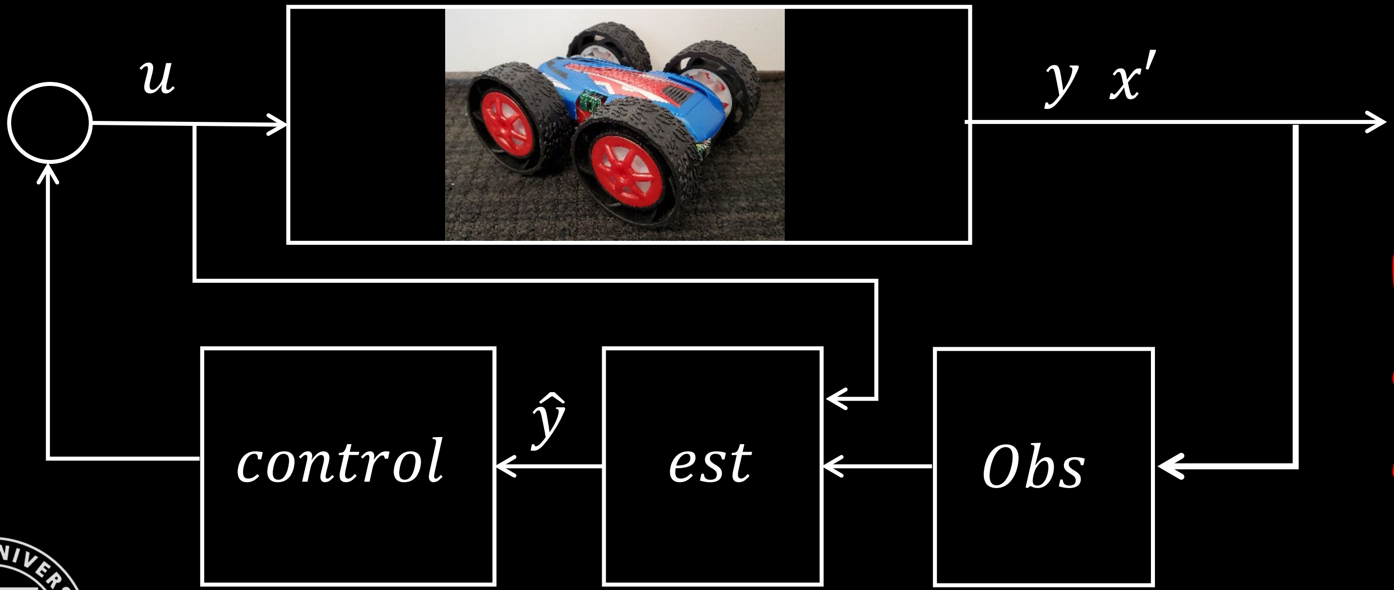


$$\dot{x} = cos(\theta)v$$
$$\dot{y} = sin(\theta)v$$
$$\dot{\theta} = w$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} cos(\theta)v & 0 \\ sin(\theta)v & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix}$$

Part 1: Implement and demo a Stunt Robot!

- Combine base with processor, drivers, and sensors
- Refresh on linear algebra and T-matrices
- Sensor modalities and types of sensors
- Actuators, drivers, circuits and routing, and EMI
- Linear systems, model-free and model-based control
  - PID controllers, Control theory, LQG control, KF

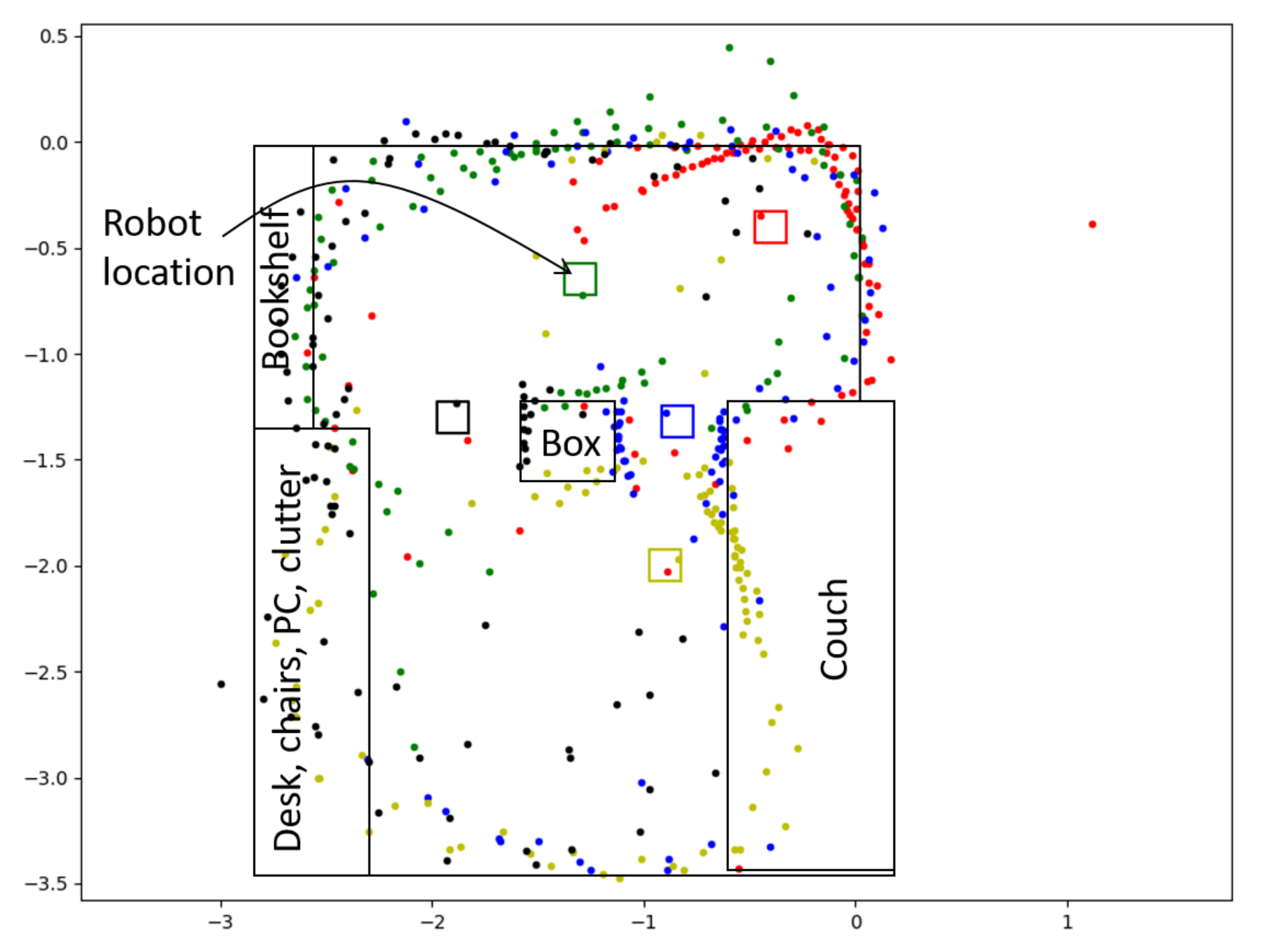


Why do you think feedback control and observers are necessary?

- Performance is battery dependent
- Our sensors are relatively slow

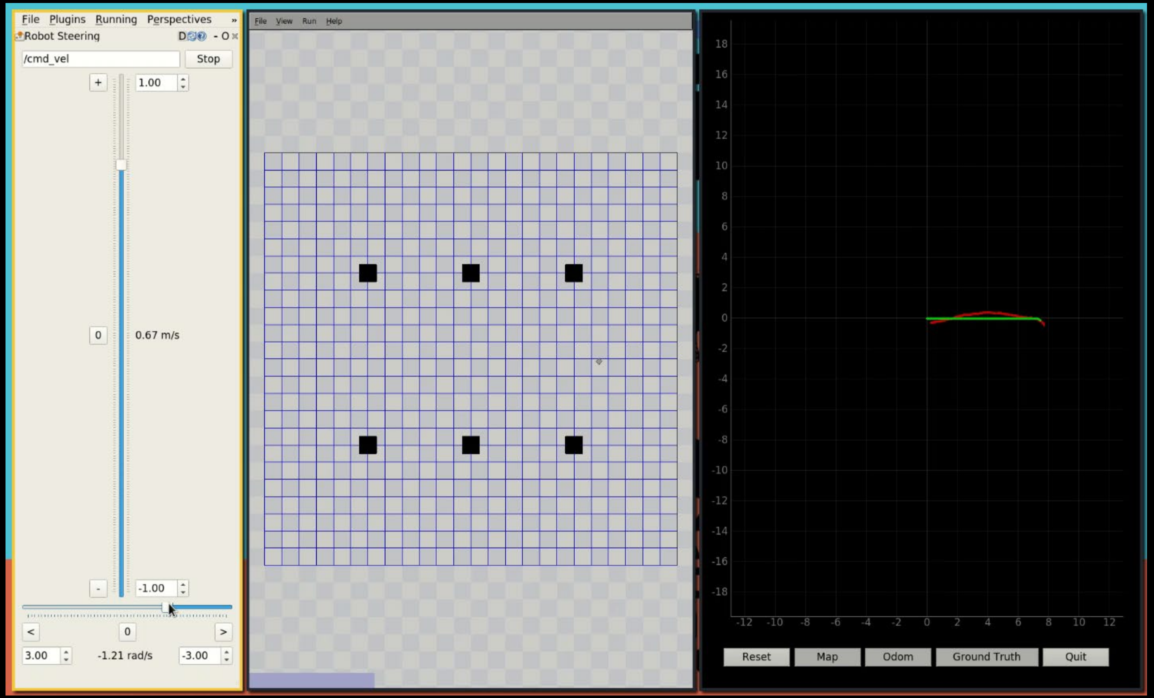
Part 2: Localization and Planning

- Map representations
- Search and planning



Part 2: Localization and Planning

- Map representations
- Search and planning
- Noise, discrete probability
- Motion and sensor models

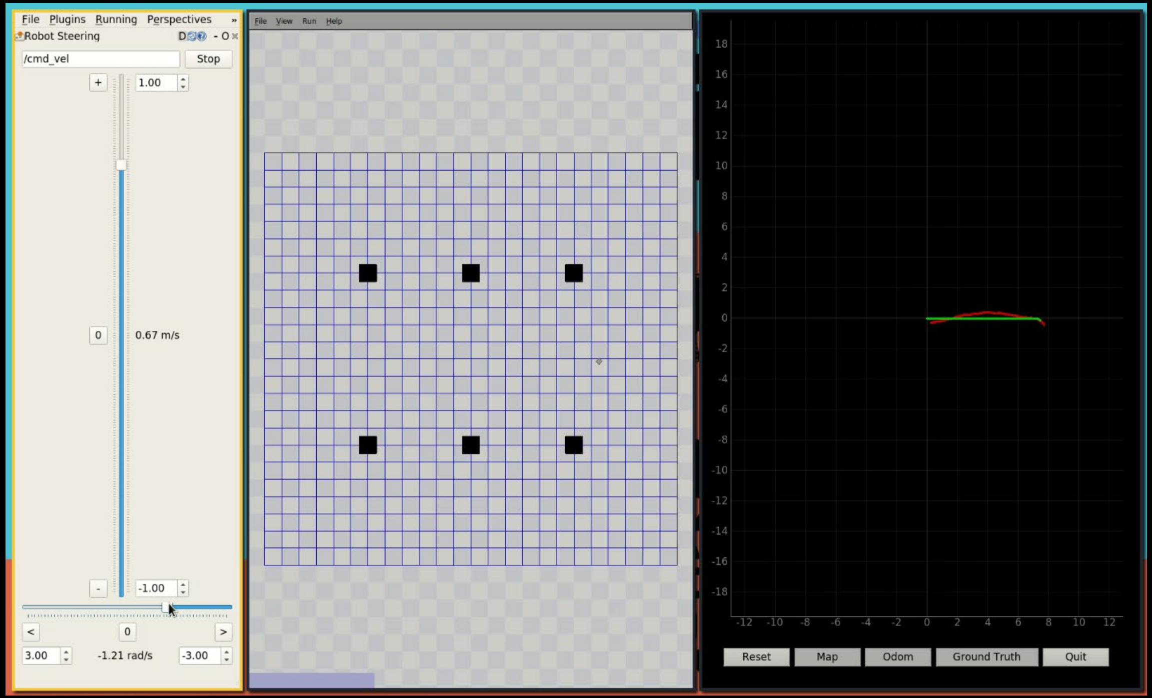


What are sources of error?

- Skid steering

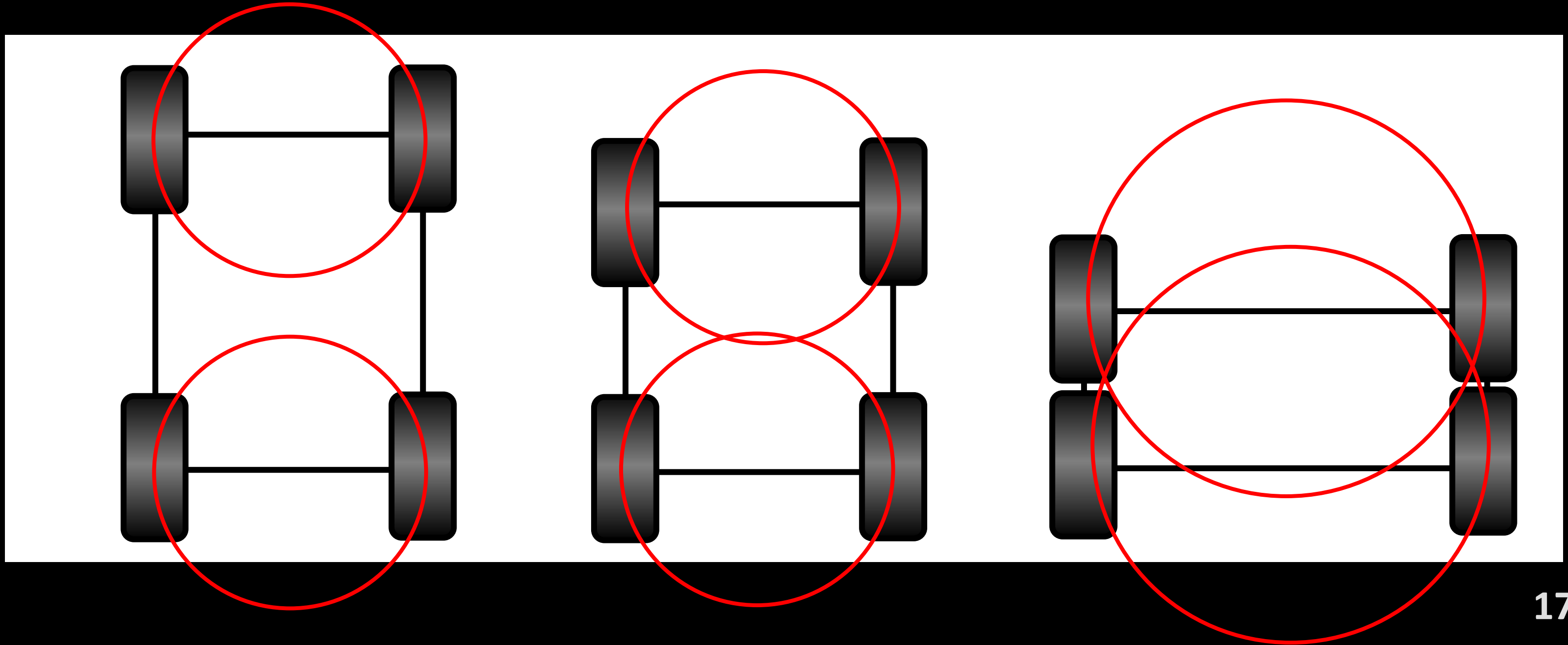
Part 2: Localization and Planning

- Map representations
- Search and planning
- Noise, discrete probability
- Motion and sensor models



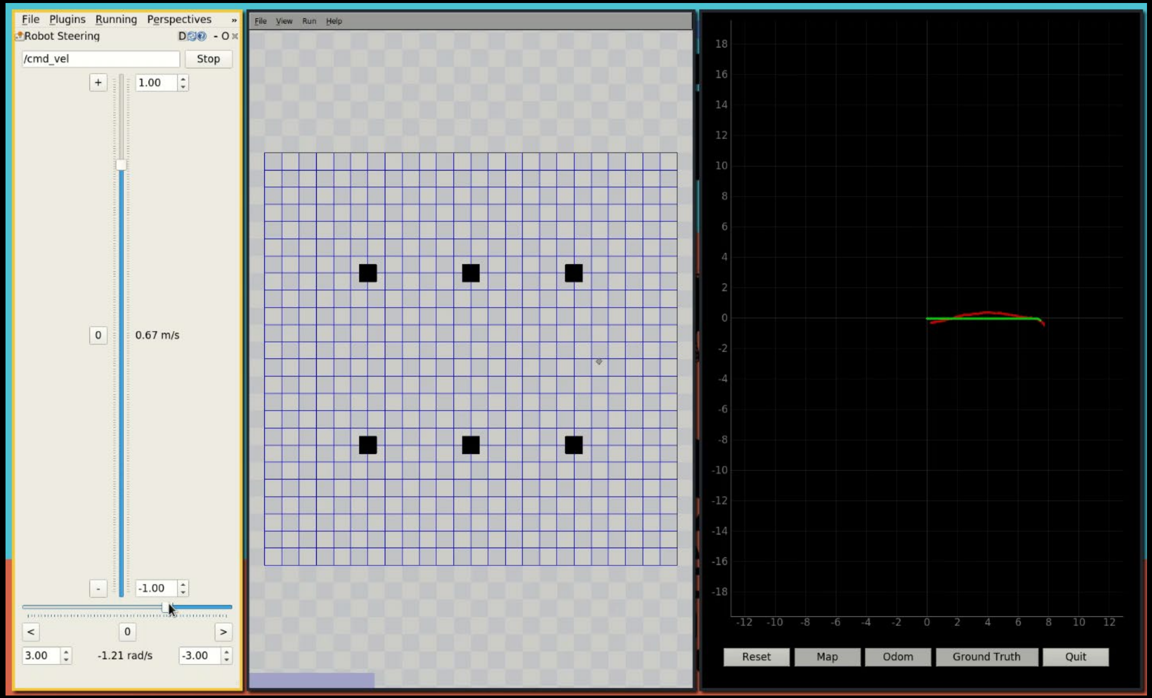
What are sources of error?

- Skid steering



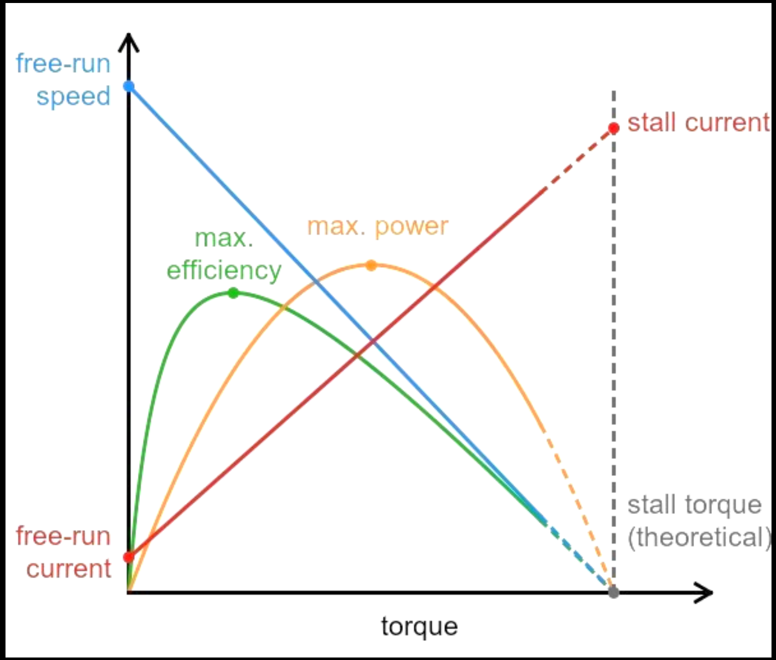
Part 2: Localization and Planning

- Map representations
- Search and planning
- Noise, discrete probability
- Motion and sensor models



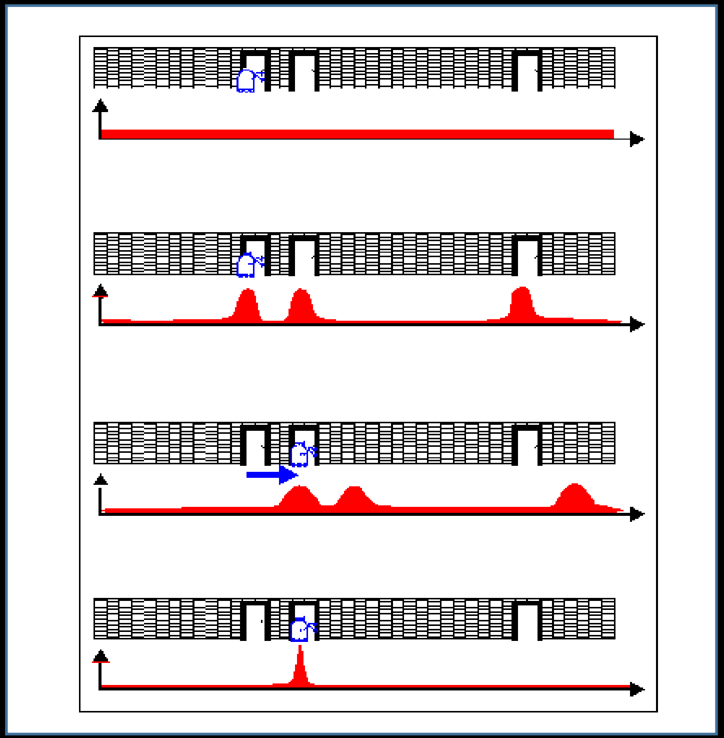
What are sources of error?

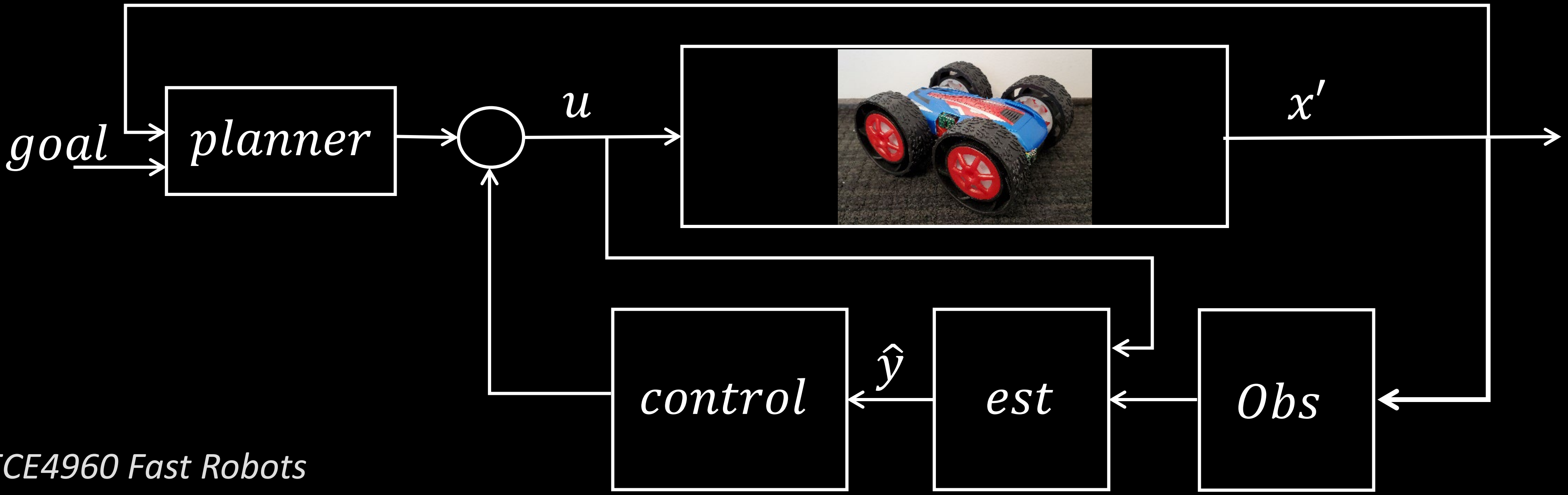
- Skid steering
- Momentum and slippage
- Weak motors
- Sensor noise, resolution



Part 2: Localization and Planning

- Map representations
- Search and planning
- Noise, discrete probability
- Motion and sensor models
- Bayes theorem/filters
- Localization





Tentative Schedule

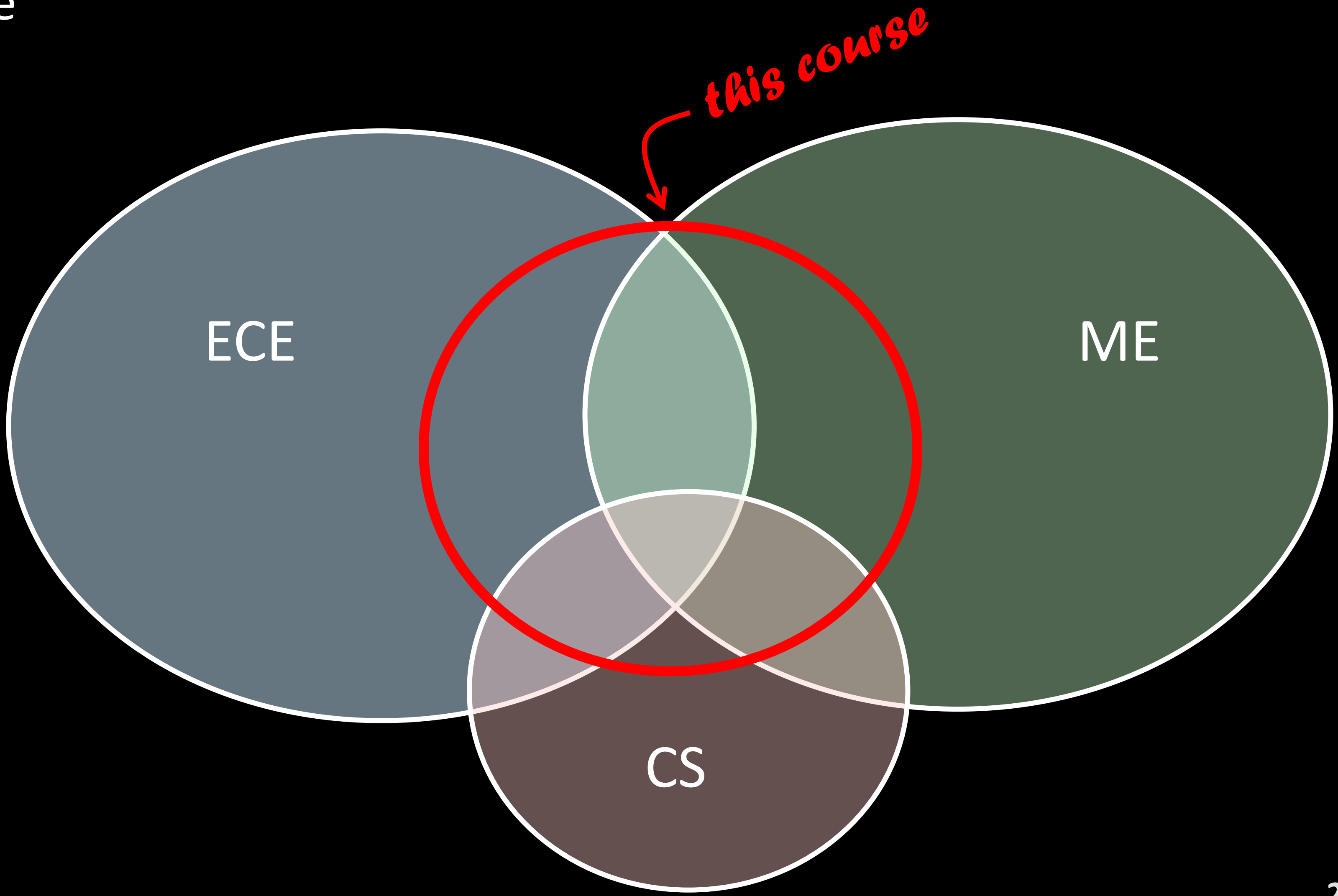
Week	Topic	Homework
1 (TR)	Class intro, Data, Comms, Linear Algebra refresh, T-matrices	Lab 1, Artemis, Make and share your Github page
2 (TR)	Sensors (TOF, proximity, encoders, IMU) and sensor fusion	Lab 2, Bluetooth communication
3 (TR)	Actuators, circuits and routing, EMI	Lab 3, TOF and IMU
4 (TR)	Linear systems, PID control	Lab 4, Characterize your car
5 (TR)	Linearizing, controllability	Lab 5, Motor driver and open loop control
6 (R)	Observability/LQG/KF	No lab
7 (TR)	Map representations, Graph search	Lab 6, PID speed control
8 (TR)	Path planning, PRM, RRT	Lab 7, Kalman Filters (sensor fusion)
9 (TR)	Noise, discrete probability, Motion models	Lab 8, Stunts
10 (TR)	In-class demonstrations, Sensor models	Lab 9, Mapping (real)
11 ( )	Spring break	No lab
12 (TR)	Bayes theorem, Bayes filter	Lab 10, Simulator
13 (TR)	Localization	Lab 11, Localization (sim)
14 (TR)	Ethics	Lab 12, Localization (real)
15 (TR)	Guest lectures, Adam Kane from AMSL (May 5th)	Lab 13, Planning and execution (real)
16 (T)	Trivia	Lab 13, Planning and execution (real)

- *Very dense schedule, we'll reconfigure as needed…*
- *Labs: ~8hrs/week on average*

Course Objective

- Somewhere between a Culminating Design Experience (learn through implementation)
- ...and a foundations course

- Overlap with Autonomous Mobile Robots, Foundations of Robotics, and Feedback Control Systems



Disclaimer!

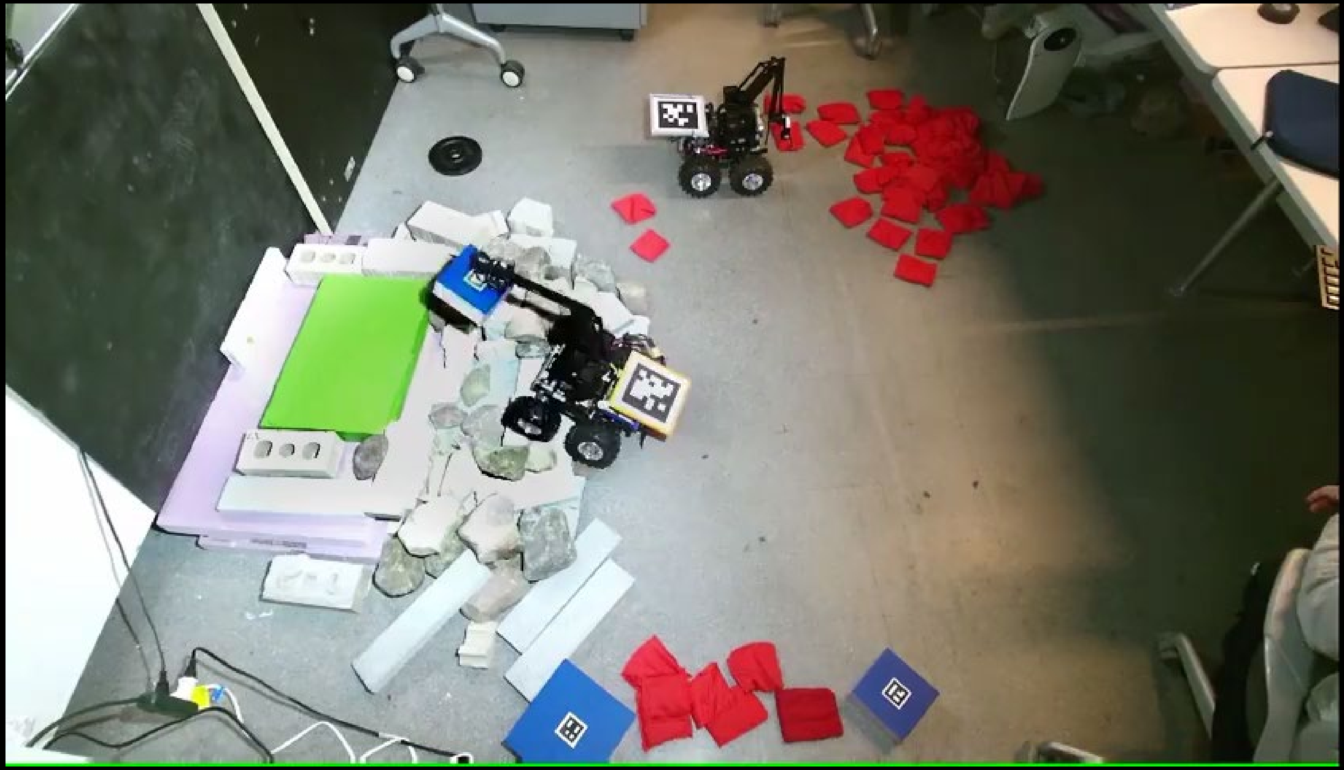
- Only the second offering
- The first time with more than 15 students
- And who knows where the pandemic takes us next?!

- Take this course if you want a highly interactive teaching team, fun and advanced challenges, experience with real robots, and an opportunity to build up an online portfolio
- Do *not* take this class, if you prefer a deep dive into fundamentals, mostly simulation, or a highly polished curriculum.

Fast Robots
Teaching Team

Your Teaching Team: Vivek ('we wake') Thangavelu

- Vivek is a graduate student in the Napp lab
- Research focus is on robot construction of support structures with found materials such as stones
- Vivek developed the simulator we will use for Part 2 and has taught courses on SLAM at UB



Your Teaching Team: Jonathan Jaramillo

- Graduate student in the CEI-lab
- Research focus is on low-cost systems to enable precision viticulture in small-scale vineyards
- Other projects include trackers for honey bees and robots for Human Robot Interaction

CornellEngineering

Collective Embodied Intelligence Lab
www.cei.ece.cornell.edu

Mobile and Inflatable Interface for Human Robot Interaction

Jonathan Jaramillo, Andrew Lin, Emma Sung, Isabel Jane Hunt Richter, and Kirstin Petersen

Ubiquitous Robots 2021

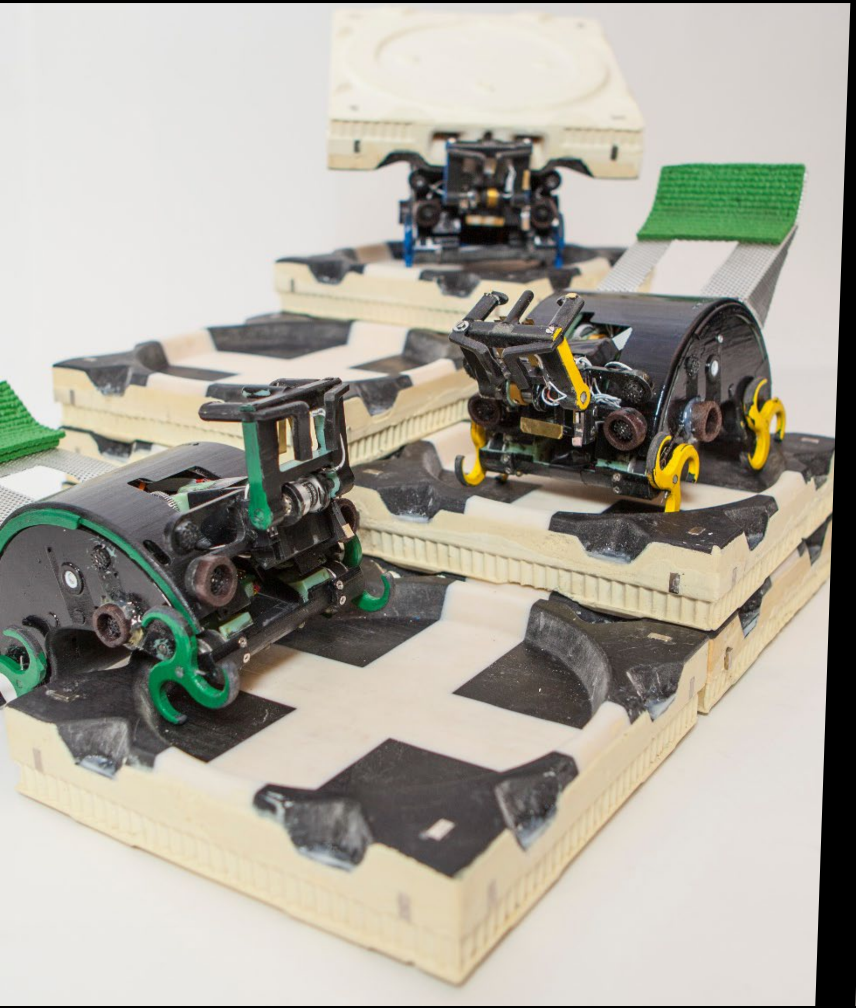
Your Teaching Team: Jade Pinkenburg

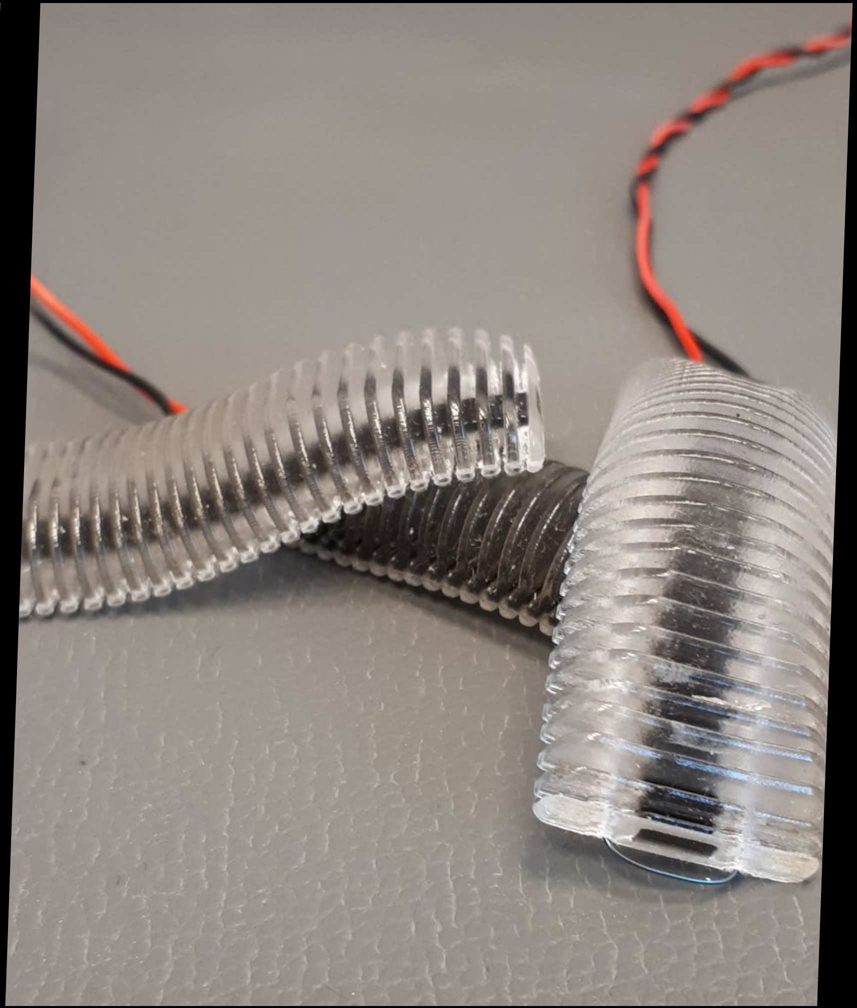
- ECE Senior, Undergrad Researcher in the Organic Robotics Lab
- EE Projects include: control systems for soft robots, biosensing chips, EOG controlled robot arm, and Jell-O transistors
- Thinks the word “solder” should not have an L in it
- Has broken the class robot many, many times

Your Teaching Team: Aratrika Ghatab

- Senior ECE starting her early M.Eng. this semester
- Conducts research in the Architectural Robotics Lab, working on simulating ocean waves on soft pneumatic surfaces using pressure regulators and Arduino.
- Interested in embedded systems and circuit design

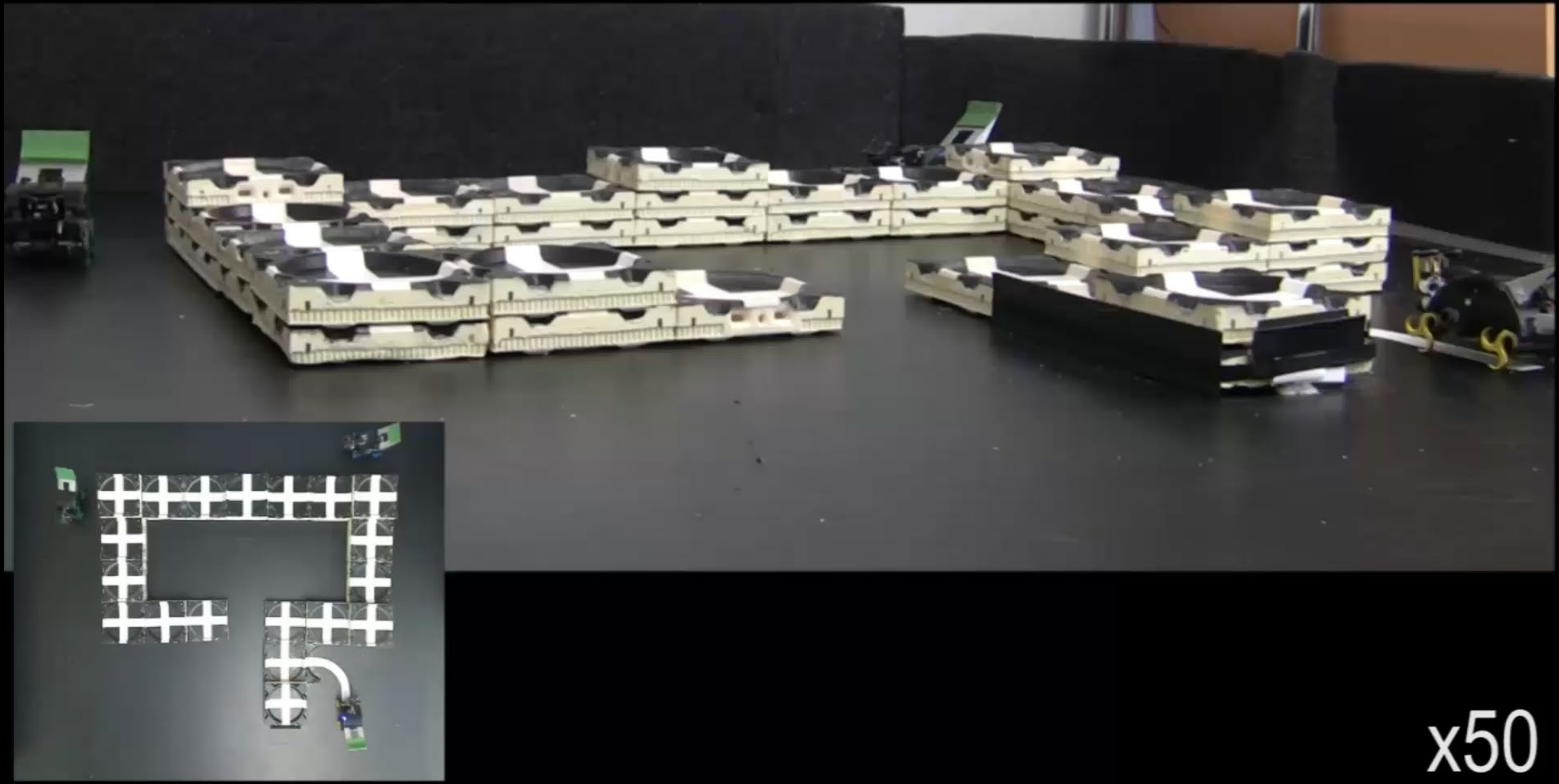
Your Teaching Team: Kirstin Petersen



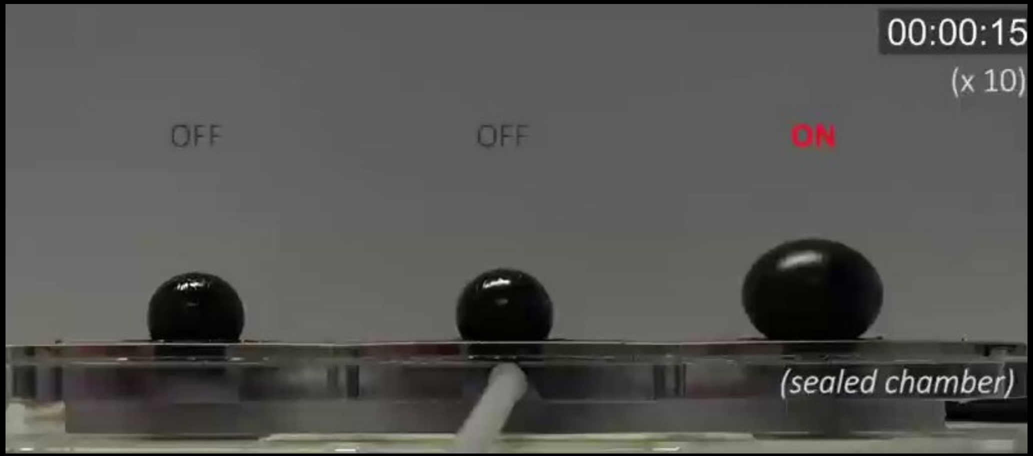


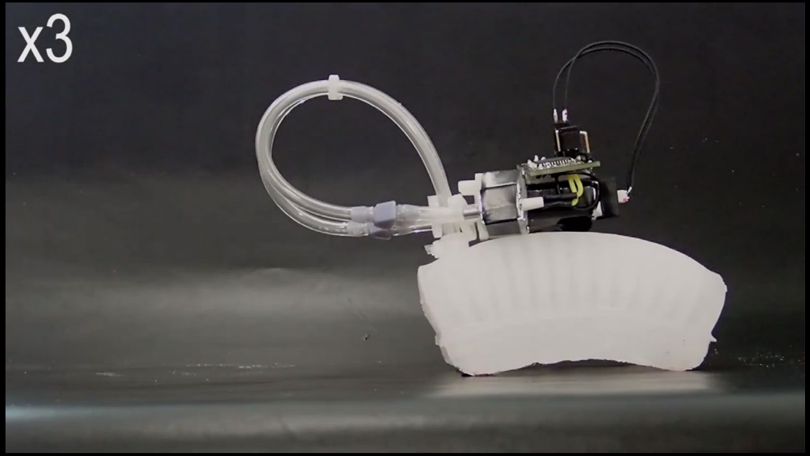
Collective Embodied Intelligence lab (www.cei.ece.cornell.edu)

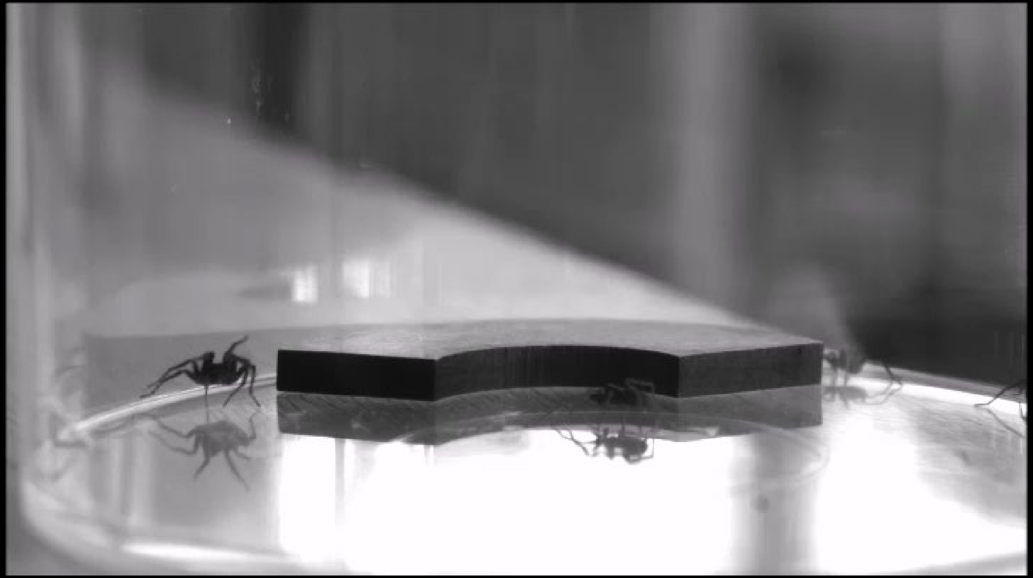
Your Teaching Team: Kirstin Petersen

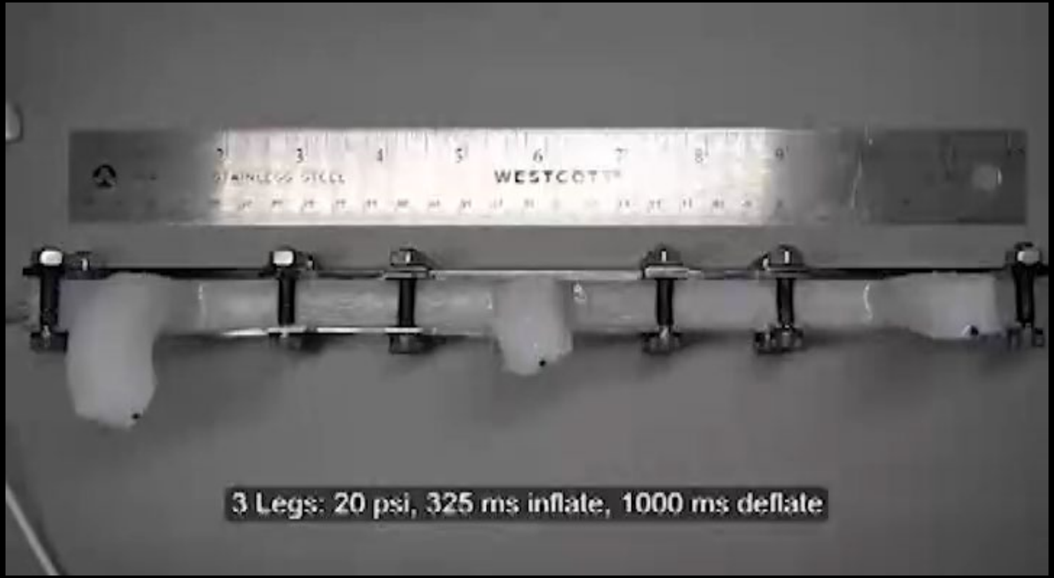


Your Teaching Team: Kirstin Petersen

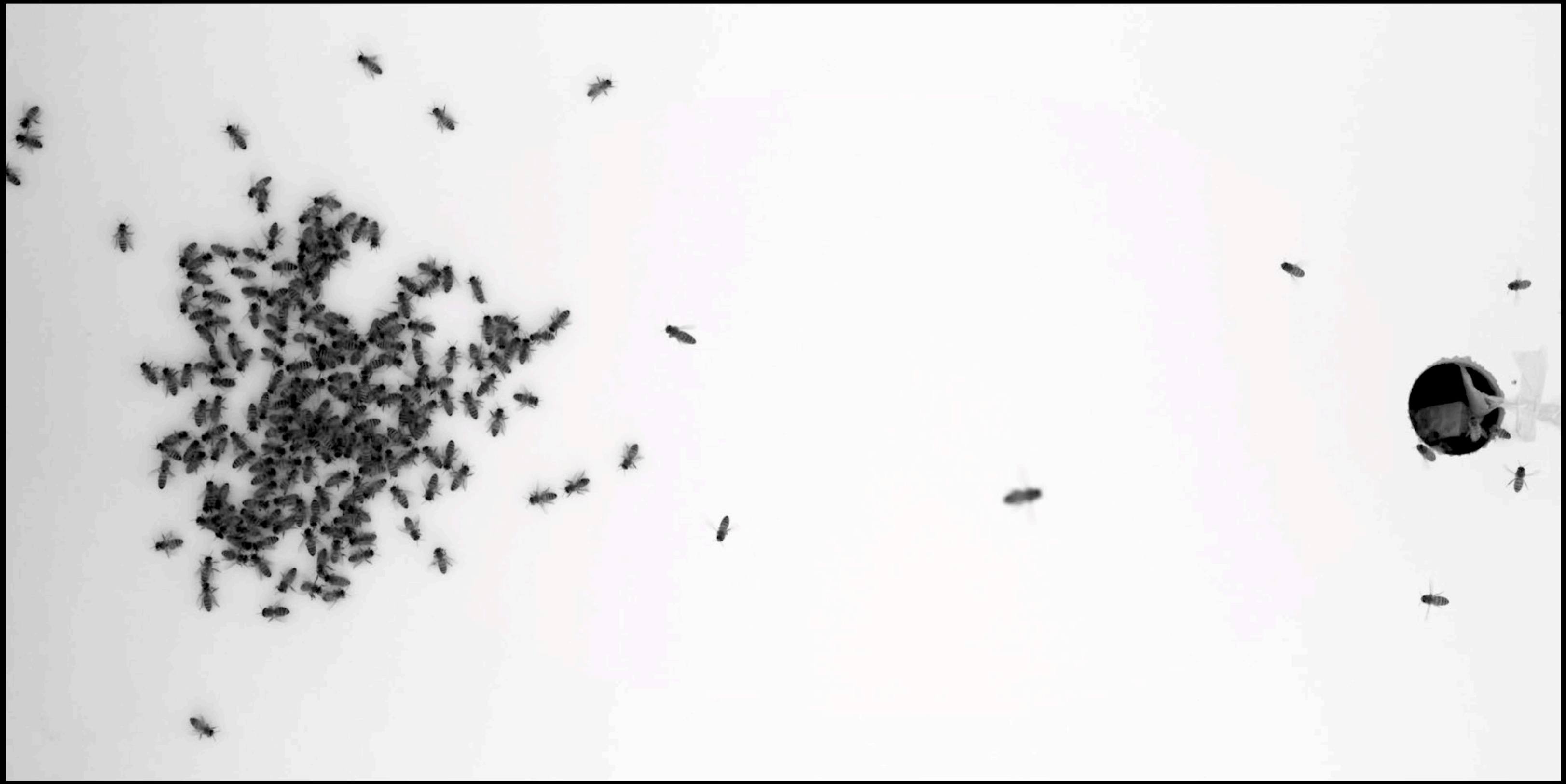








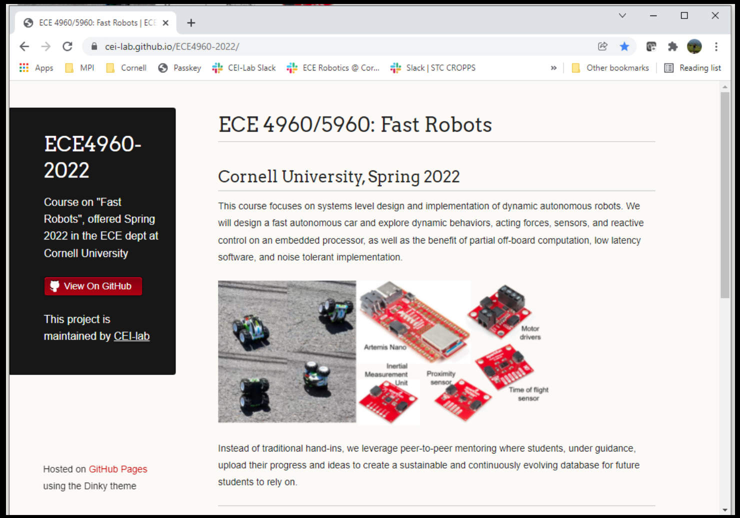
Your Teaching Team: Kirstin Petersen

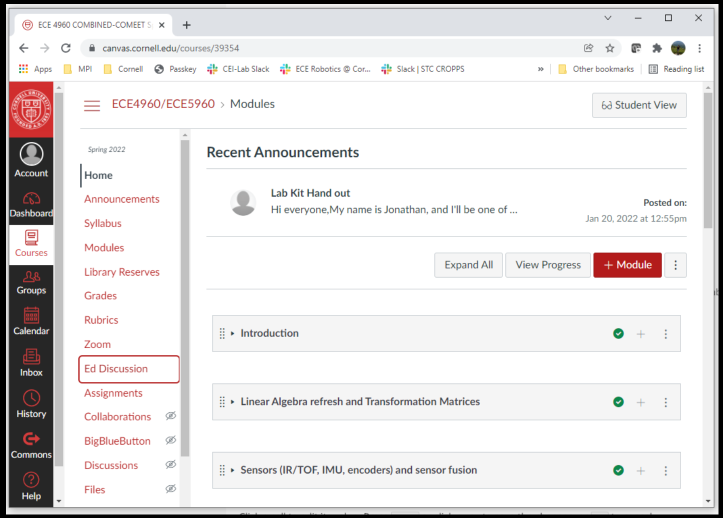


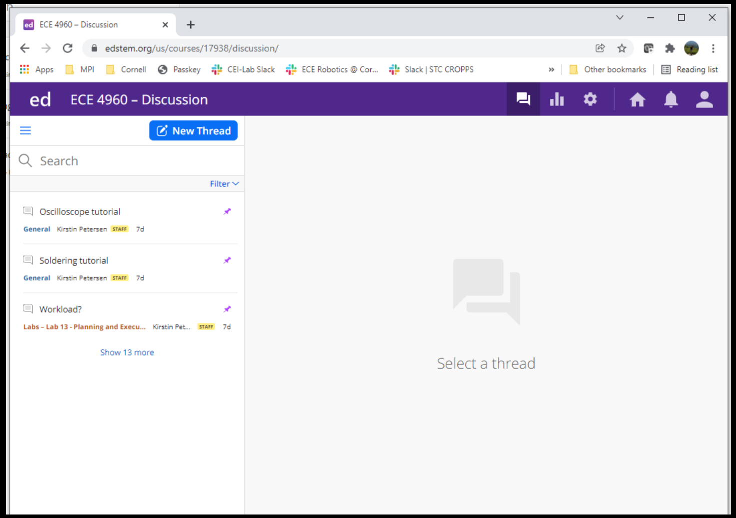
Fast Robots Logistics

Logistics I

- *Github* (https://cei-lab.github.io/ECE4960-2022/)
  - Schedule, lab schedule, lecture slides, lab documents, tutorials, code examples
- *Canvas*
  - Lecture slides, lab documents, zoom-links, grades
- *EdDiscussion*
  - If you didn't get an invite, please reach out asap!







Logistics II

- Lab kit
  - On campus
    - If you haven’t received the first half from Jonathan, please reach out asap!
  - Off campus
    - If you haven’t reached already, please do so!
  - ASML generously paid for every lab kit
  - ….Things will break, we have a small set of extra loaner components, but please be careful
  - ….Supply crisis!
- Lab software
  - Windows 10, MacOS 12 and Linux (bluez>5.48, kernel=4.15)
  - Processor: Core i3-8100 3.6 Ghz/AMD Ryzen 5 1400 or equivalent
  - Memory: 4 GB RAM, Free Space: 8 GB (Windows)/1GB (else)
- Lab computers…

Logistics III

- Labs
  - Tuesday – Wednesday – Thursdays in PH427, 2.40-5.10pm (max 15 students!)
  - Labs are meant to take an average of 8 hrs
  - Many labs (and most of every lab) can be done *remotely*!
  - Beyond your lab section, there are open lab hours 12-5pm Saturdays and 4-7 Sundays
    - Max 18 people in the lab (incl. TAs)
  - *¡The car has limited battery life, do the labs over multiple days!*

  - If you run low on time…
    - You can redo any* *two* labs for a complete re-grading, at any point up until May 10th, provided you inform Kirstin at least 7 days in advance of re-submission.
    - *With the exception of Lab 11
- Lab reports → Your own Github sites (check out examples from 2020 here)

Logistics IV - Grading

- 13 Labs (90 pts)
  - Points for solution (60)
  - Points for write-up (30)
- Quizzes/Polls (0 pts)
- Participation (10 pts)
- Course evaluations (2 bonus pts)

- Grading policy
  - Collaboration is welcome
  - Optional tutorials for fellow students
  - But implement your own code
  - Always credit collaborators/references

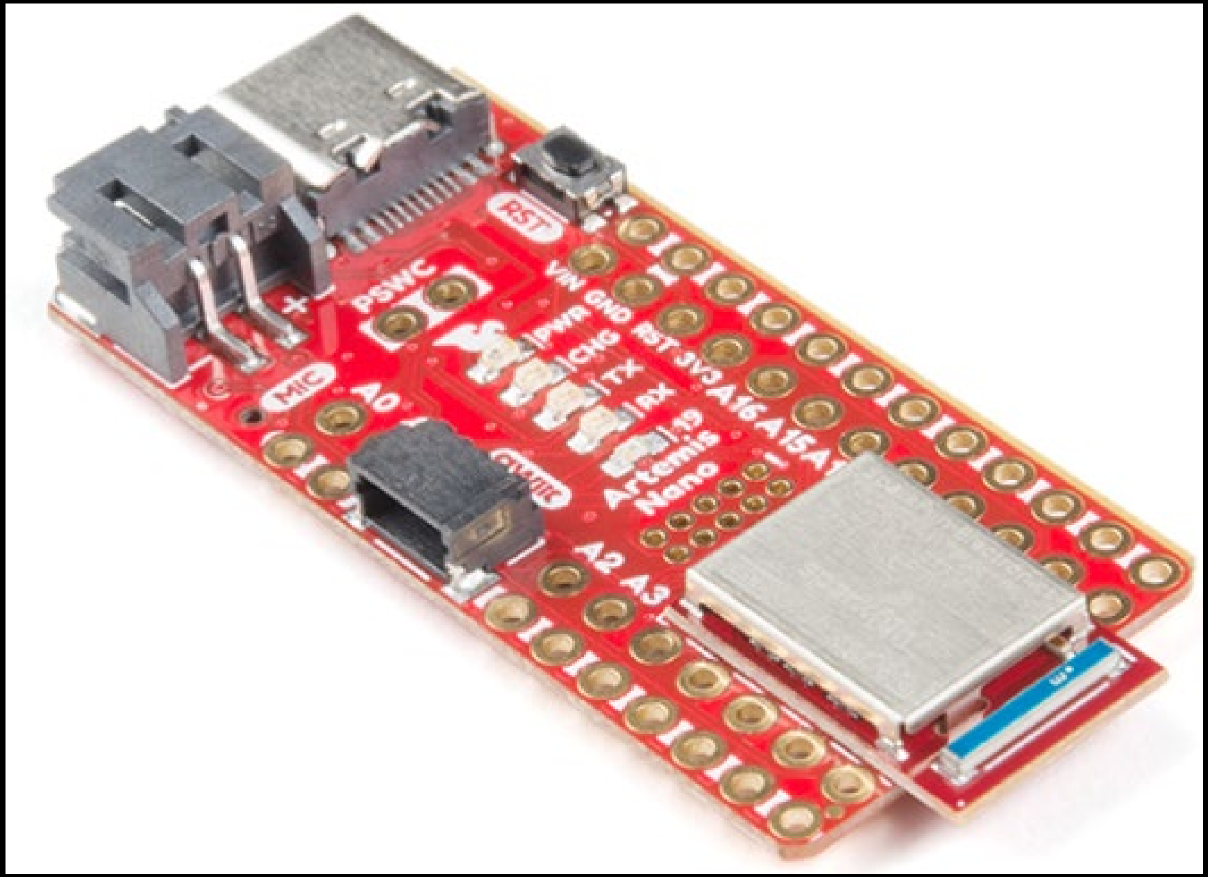
Task	pts
Lab 1 Artemis	5
Lab 2 Bluetooth	5
Lab 3 Sensors	7.5
Lab 4 Car characterization	5
Lab 5 Motor driver, open loop control	5
Lab 6 PID	5
Lab 7 KF	7.5
Lab 8 Stunts	10
Lab 9 Mapping	5
Lab 10 Simulation	5
Lab 11 Localization (sim)	10
Lab 12 Localization (real)	10
Lab 13 Planning and Execution	10
Participation	10
Bonus points for midway and final course evaluations!	2
Total:	**102**

Action items

- *If you decide not to take the course, let Kirstin know ASAP*
- Fill out the Google form
- Jan 28th, midnight:
  - Make a Github repository and build a Github page
    - Your name, a small introduction, the class number, and a photo
  - Share the page link with Kirstin
- Labs starts online *this* week
  - Lab 1 should be quick
  - TAs are available during your regular lab times on a zoom link (TBD)
  - Upload your write-up of Lab 1 by 8am the following week
    - (E.g. Tuesday lab write-ups are due the following Tuesday 8am)
- Please fill out the polls on EdDiscussion
  - Oscilloscopes
  - Soldering

Fast Robots
Artemis

Lab 1: The Artemis Board



- The Board:

https://www.sparkfun.com/products/15443

- Support forum:

https://forum.sparkfun.com/viewforum.php?f=167&sid=903070e43f577f5afd5010828e1bf716

- Bluetooth
- PDM
- LiPo Charger
- I2C Qwiic connectors
- *3V board*
- Apollo 3 MCU

Lab 1: The Artemis Board

https://cdn.sparkfun.com/assets/d/e/8/b/4/Apollo3_Blue_MCU_Data_Sheet_v0_12_1_rZ9Akgo.pdf

ambiq micro

Apollo3 Blue MCU Datasheet

Ultra-Low Power Apollo MCU Family

Features

Ultra-low supply current:
- 6 µA/MHz executing from FLASH or RAM at 3.3 V
- 1 µA deep sleep mode (BLE Off) with RTC at 3.3 V (BLE in SD)

High-performance ARM Cortex-M4 Processor
- 48 MHz nominal clock frequency, with 96 MHz high performance TurboSPOT™ Mode
- Floating point unit
- Memory protection unit
- Wake-up interrupt controller with 32 interrupts

Integrated Bluetooth[1] 5 low-energy module
- RF sensitivity: -93 dBm (typical)
- TX: 3 mA @ 0 dBm, RX: 3 mA
- Tx peak output power: 4.0 dBm (max)

Ultra-low power memory:
- Up to 1 MB of flash memory for code/data
- Up to 384 KB of low leakage RAM for code/data
- 16 kB 2-way Associative/Direct-Mapped Cache

Ultra-low power interface for on- and off-chip sensors:
- 14 bit ADC at up to 1.2 MS/s, 15 selectable input channels available

- 3.37 x 3.25 mm(<0.35mm thk pkg) 66-pin CSP with 37 GPIO
- 5 x 5 mm (<0.5mm thk pkg) 81-pin BGA with 50 GPIO

Applications

- Voice-on-SPOT™ compatible for always-listening keyword detect, audio command recognition and voice assistant integration in battery-powered devices including:
  - Bluetooth headsets, earbuds, and truly wireless earbuds
  - Remote and Gaming Controls
  - Smart home
- Wearables including smart watches and fitness/activity trackers
- Hearing aids, Digital Health Monitoring and Sensing Devices
- Smart Home Automation, Security and Lighting control applications

Description

The Apollo MCU Family is an ultra-low power, highly integrated microcontroller platform based on Ambiq Micro's patented Subthreshold Power Optimized Technology (SPOT™) and designed for battery-powered and portable, mobile devices. The Apollo3 Blue MCU sets a new standard in energy efficiency for battery-powered devices with an integrated ARM Cortex-M4 processor with Floating Point Unit and TurboSPOT™ increasing the compu-

*Single-Instruction Multiple-Data ops, floating point unit
-> Audio, Fast-control loop closure

Lab 1: The Artemis Board

https://cdn.sparkfun.com/assets/d/e/8/b/4/Apollo3_Blue_MCU_Data_Sheet_v0_12_1_rZ9Akgo.pdf

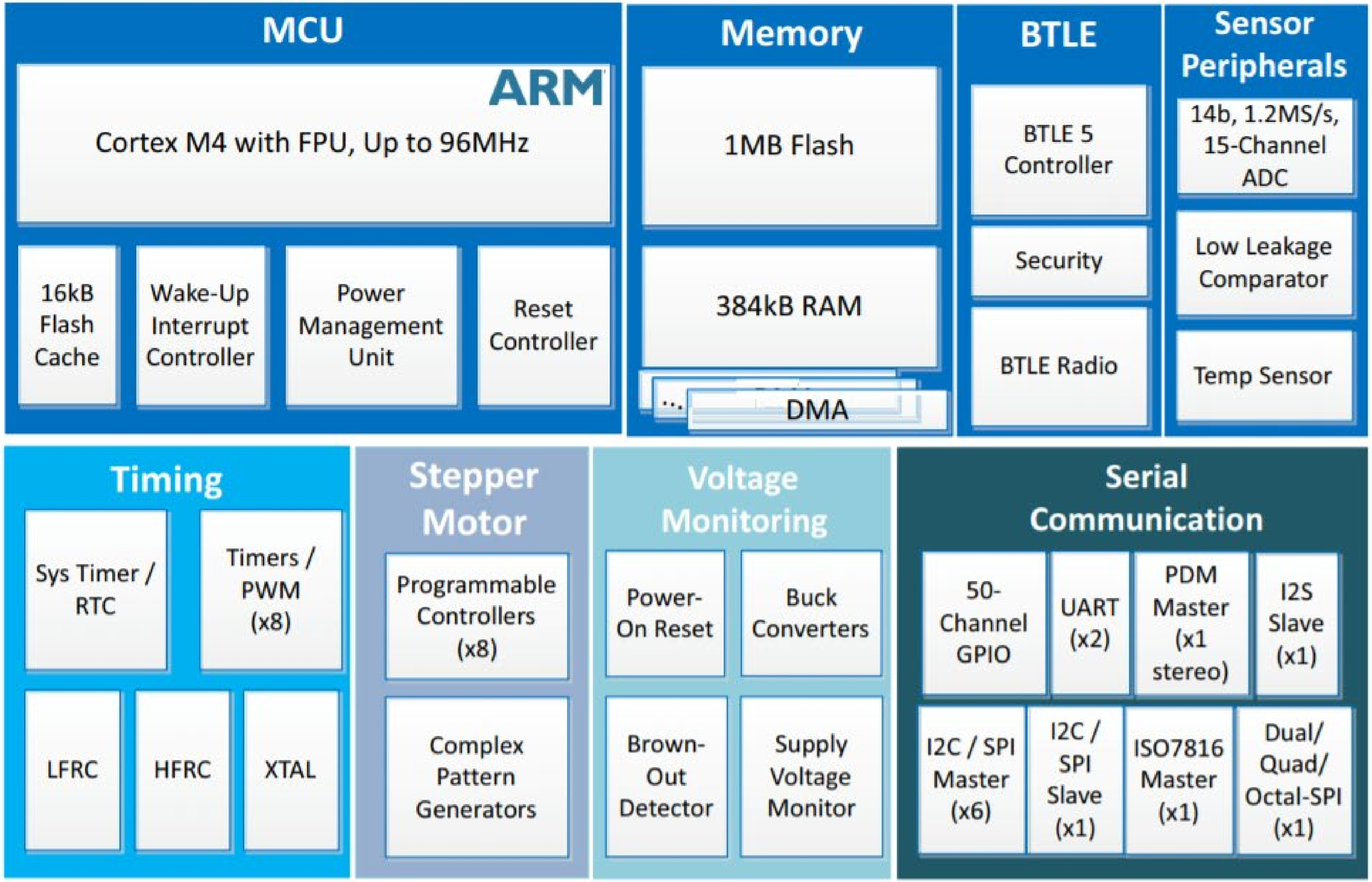


Figure 3. Block Diagram for the Ultra-Low Power Apollo3 Blue MCU

Prof. Kirstin Hagelskjær Petersen
kirstin@cornell.edu

DATA TYPES

Data types

- What data types will you have in your system?
  - Bluetooth: char
  - Time of flight: unsigned int
  - Serial.print: strings
  - IMU: float
  - PID: double
  - millis(): unsigned long
  - if-statements: bool
- *Pay attention!*
- https://www3.ntu.edu.sg/home/ehchua/programming/java/datarepresentation.html

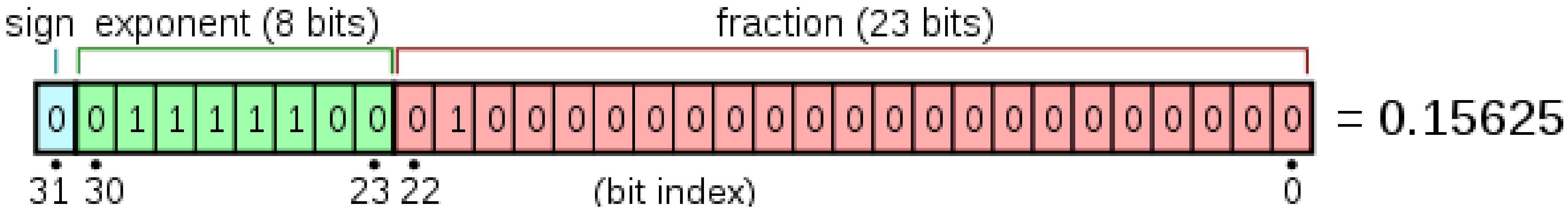
Data types

- Variable memory allocation depends on your processor *and* the compiler
  - Char
    - $Char_{8bit}$: 8 bits
    - $Char_{32bit}$: 8 bits
  - Int
    - Int_{8bit} : 16 bits
    - Int_{32bit} : 32 bits
  - Long
    - $Long_{32bit}$: 32bits
    - $Long_{64bit}$: 64 bits
- Two's complement
    - 0 b 0 0 0 0 0 1 0 1
    - 0 b 0 0 0 0 0 1 0 1 > invert > 0 b 1 1 1 1 1 0 1 0 > add 1 > 0 b 1 1 1 1 1 0 1 1
    - 0 b 1 1 1 1 1 1 1 1 ?
    - Range of a signed $char_{32bit}$: $[-2^7 ; 2^7 -1]$ = [-128 ; 127]
    - Range of a signed int_{32bit}: $[-2^{31} ; 2^{31}-1]$

You can specify the length:
- int16_t
- uint32_t

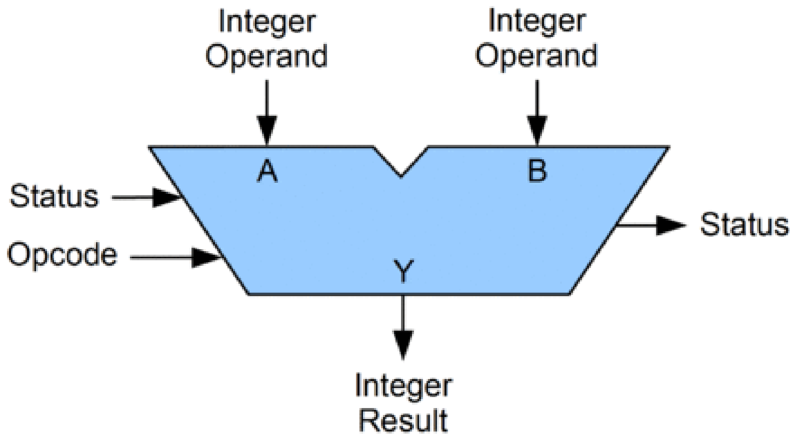
Data types

- Variable memory allocation depends on your processor *and* the compiler
    - Float
        - $Float_{8bit}$: 32 bits
        - $Float_{32bit}$: 32 bits
        - Single-precision floating point number:
            - Max value : ≈ 3.4028235×10^{38}

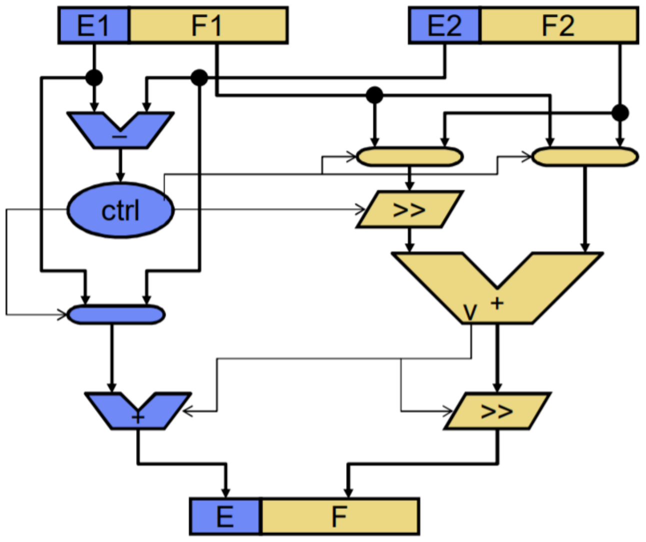


$$(-1)^{b_{31}} \times 2^{(b_{30}b_{29}\dots b_{23})_2-127} \times (1.b_{22}b_{21}\dots b_0)_2$$

Integer ALU



Floating point ALU



Data types

- Variable storage depends on your processor *and* the compiler
  - Float
    - Float_{8bit} : 32 bits
    - Float_{32bit} : 32 bits
    - Single-precision floating point number:
      - Max value : $\approx 3.4028235 \times 10^{38}$
  - Double
    - Double_{8bit} : 64 bits
    - Double_{32bit} : 64 bits
  - Long Double
    - 8, 12, 16 bytes

Communication with Sensors / Sparkfun “Qwiic” connectors

I2C