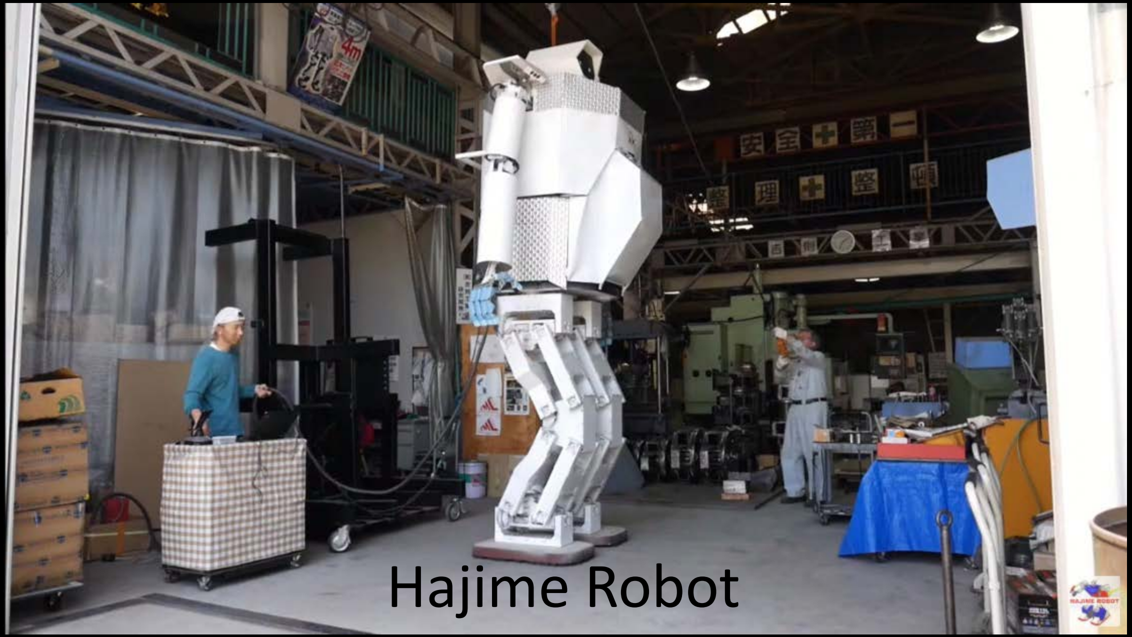
ECE 4960

Prof. Kirstin Hagelskjær Petersen
kirstin@cornell.edu

Fast Robots

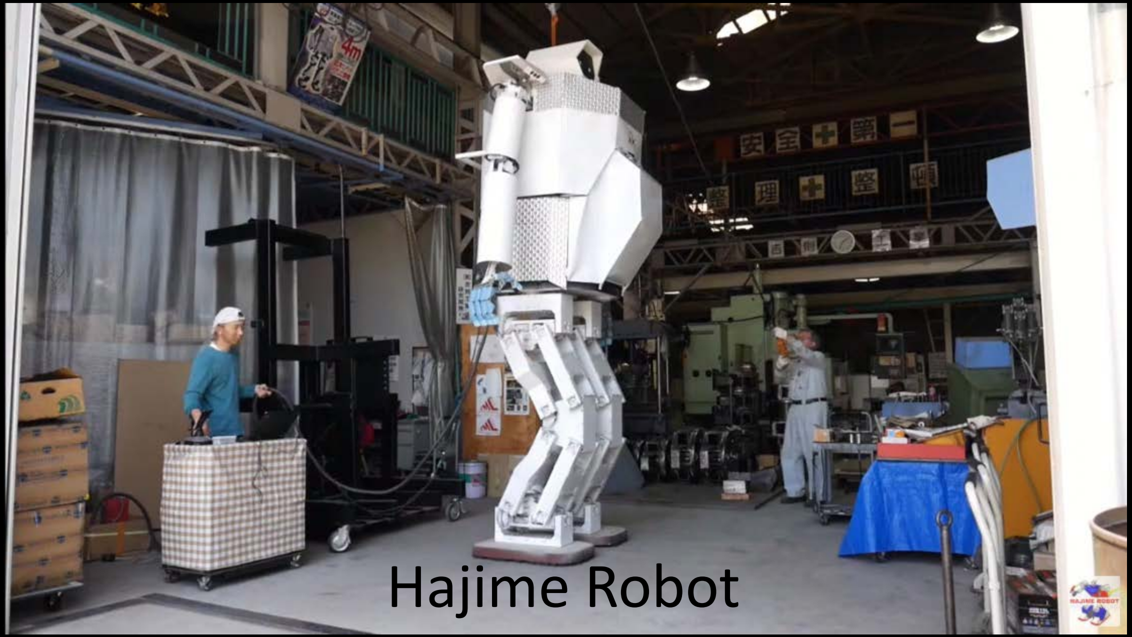
Why *this* class?

- Fast robots are fundamentally different from slow robots



Why *this* class?

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



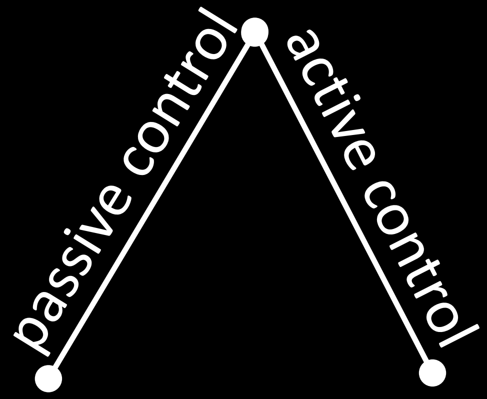
Why *this* class?

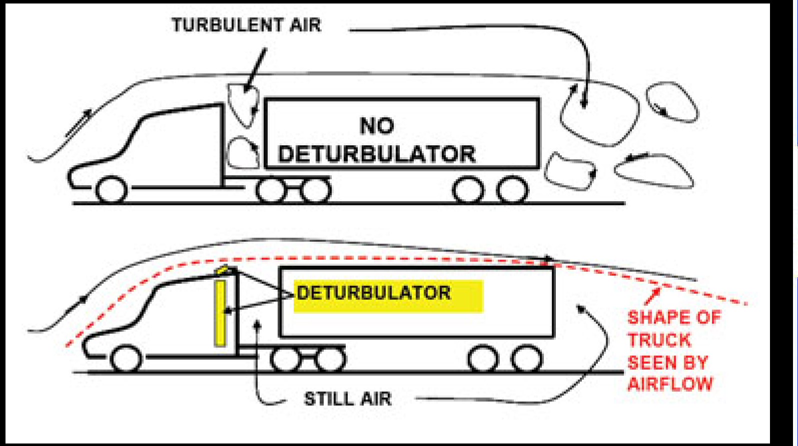
- Fast robots are fundamentally different from slow robots
  - Kinematics – Dynamics
  - Stable – Unstable

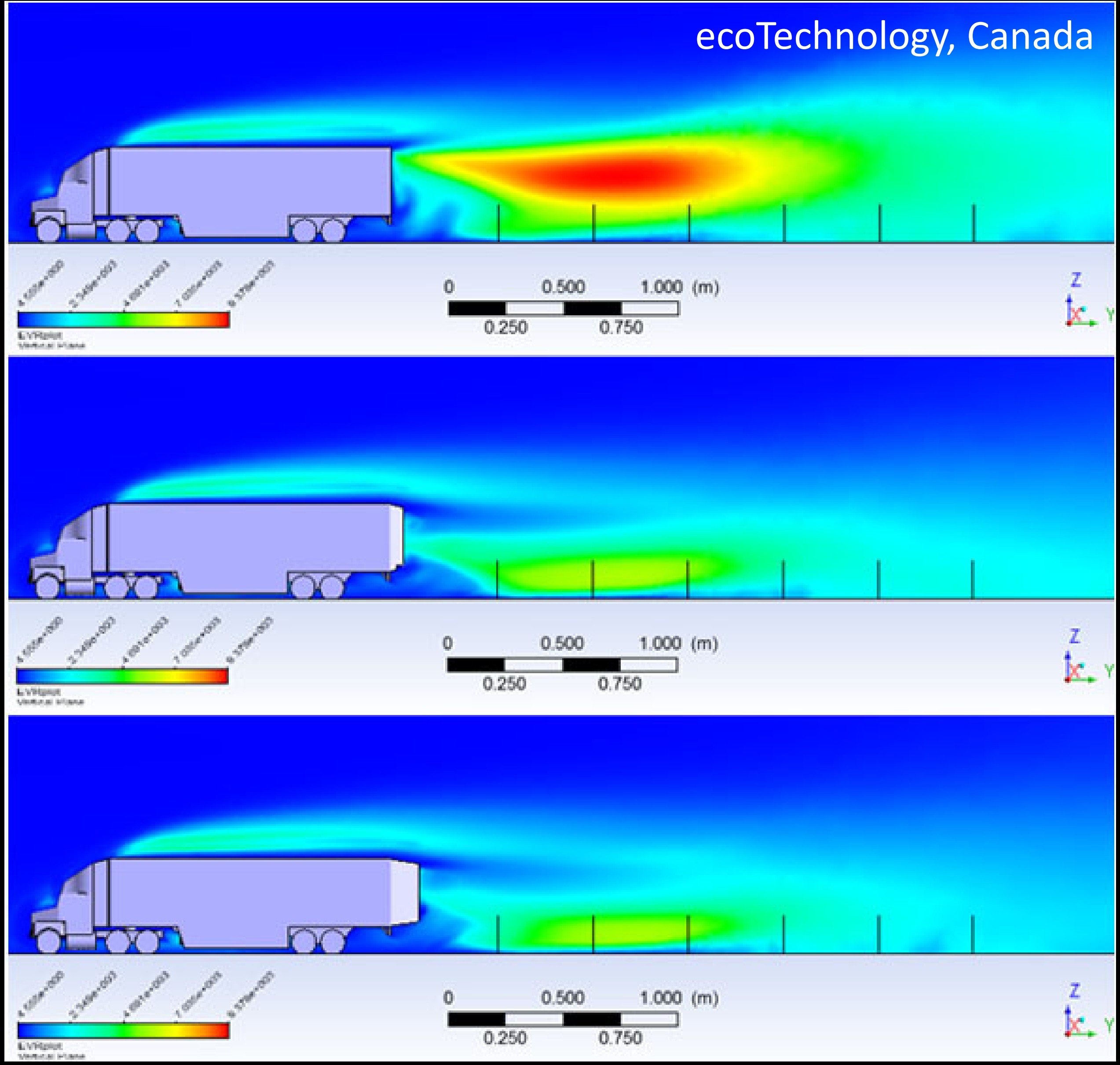
Why *this* class?

- 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.

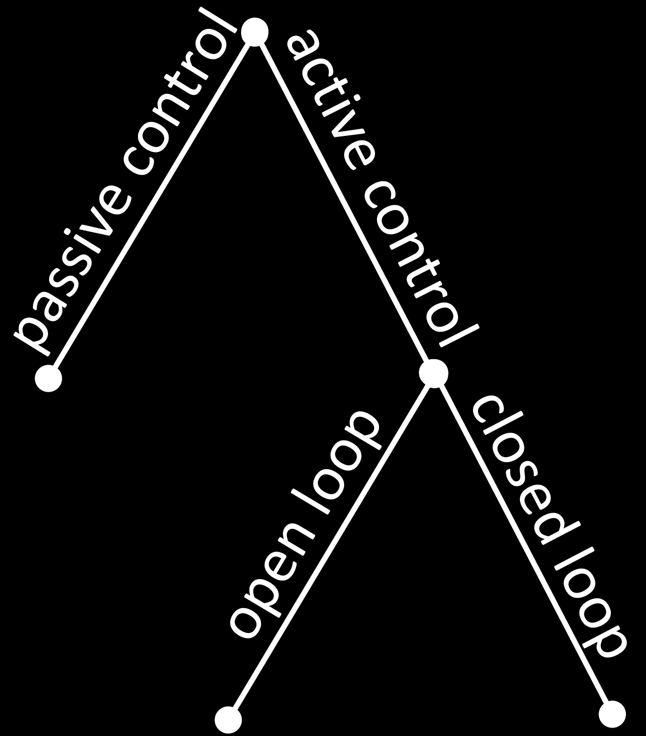
Control and its implications in fast robots







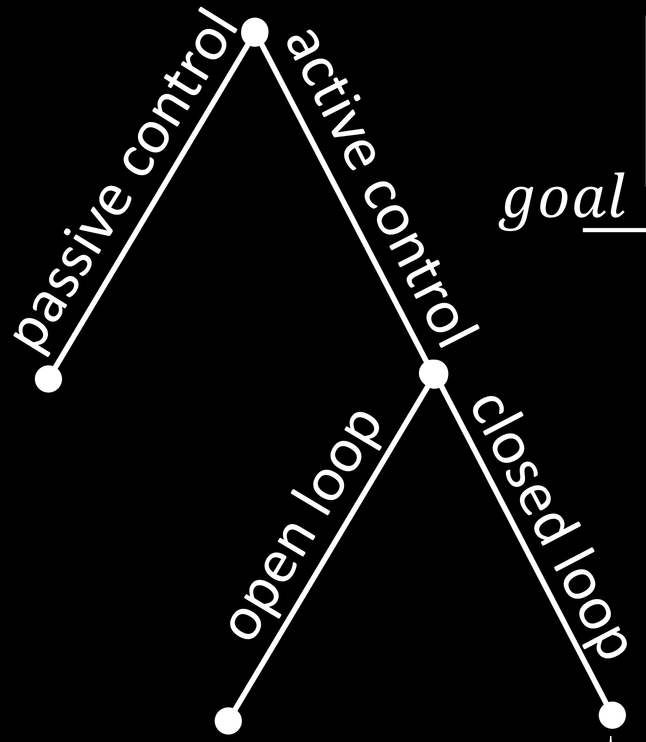
Control and its implications in fast robots

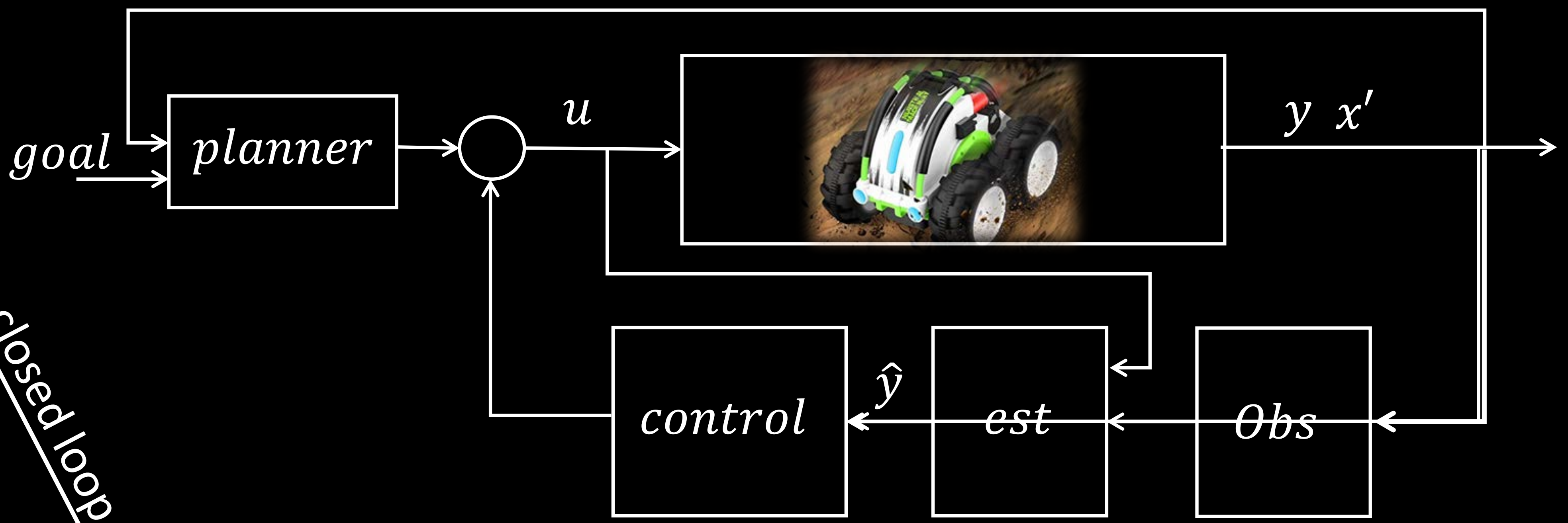


Why feedback?

- System uncertainty
- Instability
- Disturbances
- Efficiency

Control and its implications in fast robots





- processor
- drivers
- limits

- sensors
- noise/bias

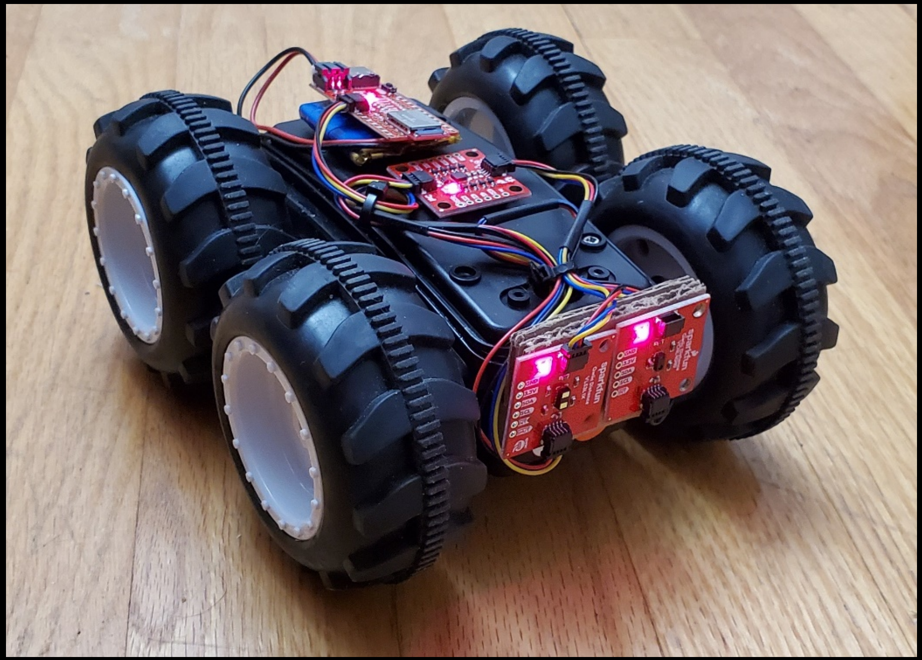
Why feedback?

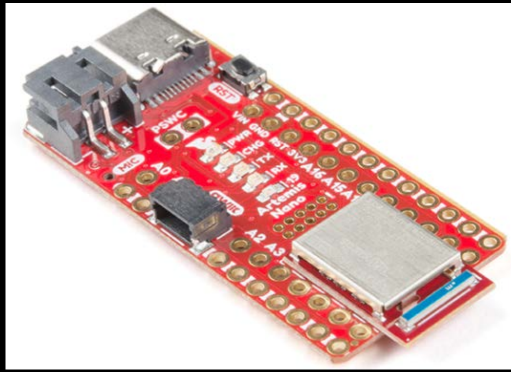
- System uncertainty
- Instability
- Disturbances
- Efficiency

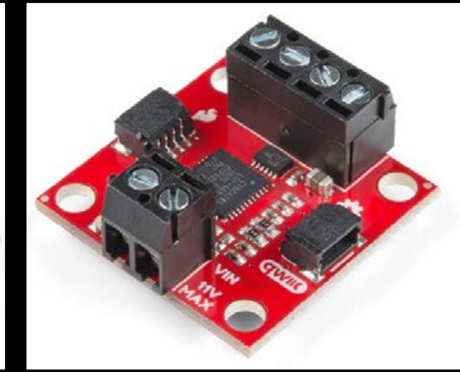
Fast Robots
Class Structure

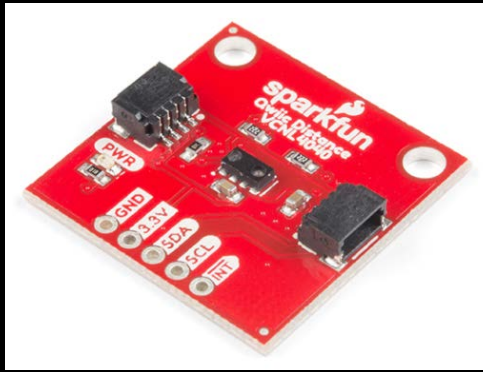
Part 1: Implement the Robot

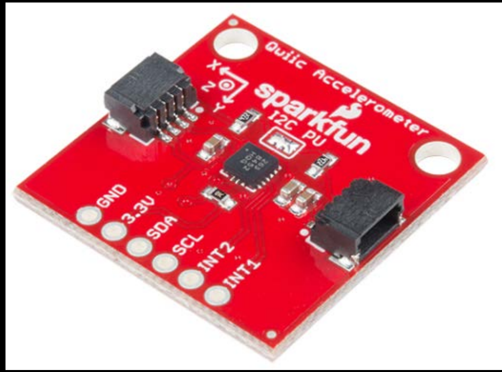
- Combine base with processor, drivers, and sensors
- Pros/cons of sensor modalities and types of sensors
  - Noise, bias, and sampling frequency
- Simulation platform
  - Sensor models









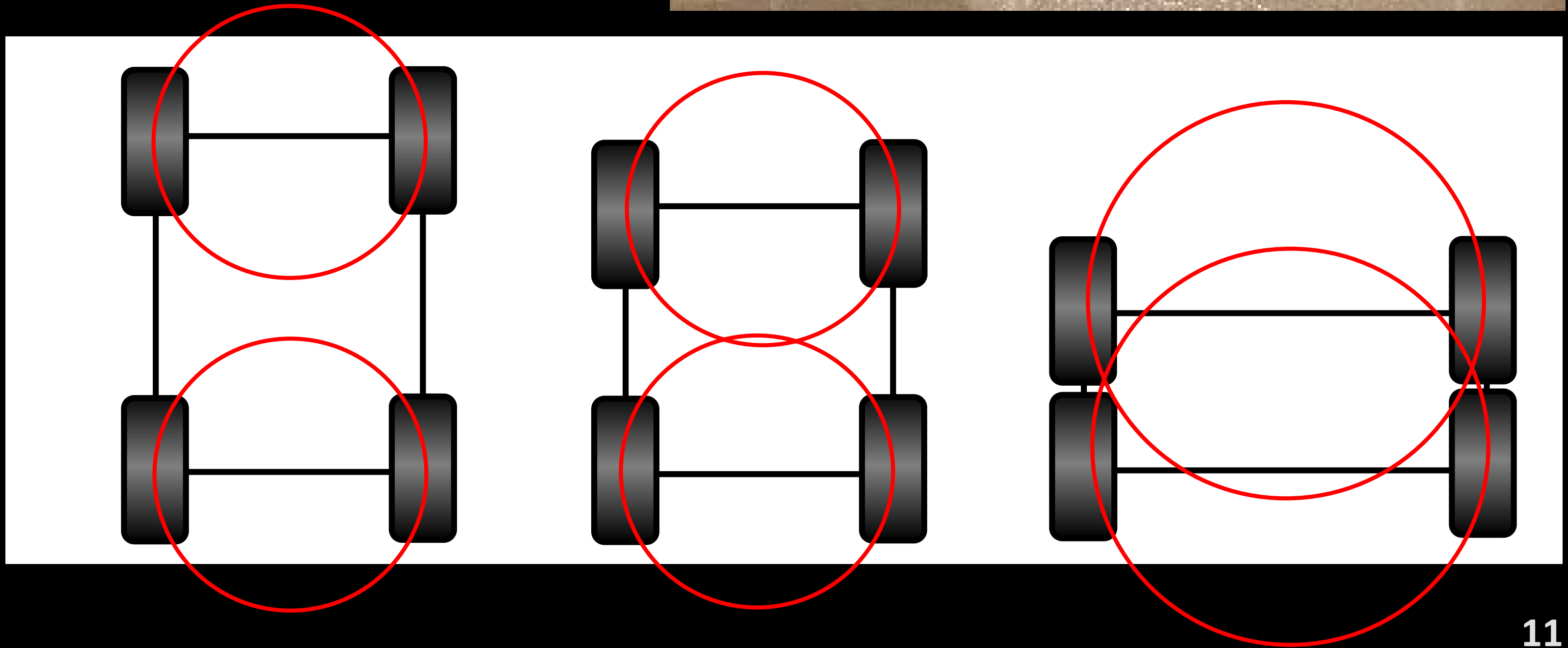


Part 1: Implement the Robot

- Combine base with processor, drivers, and sensors
- Pros/cons of sensor modalities and types of sensors
  - Noise, bias, and sampling frequency
- Simulation platform
  - Sensor models
  - Motion models

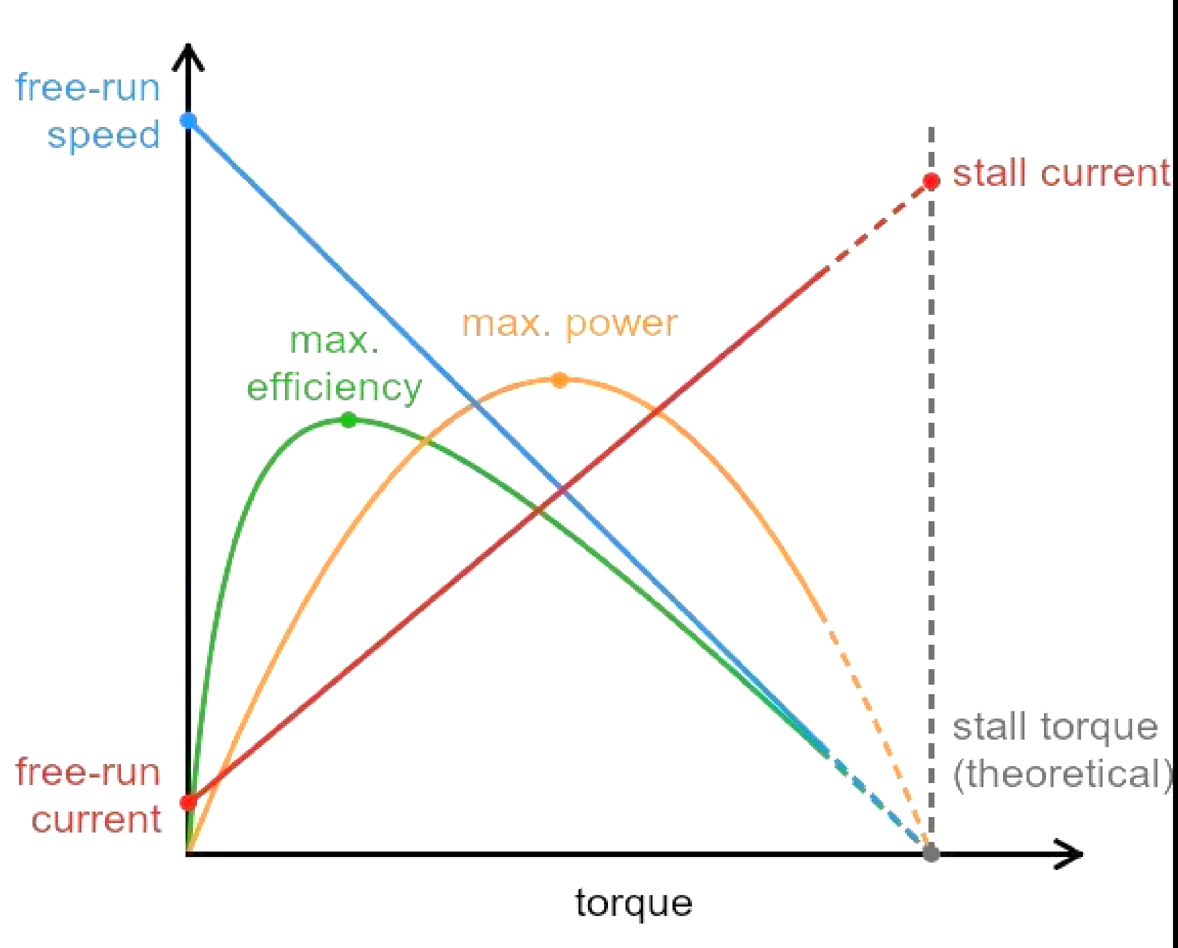
What are sources of error?

- Skid steering



Part 1: Implement the Robot

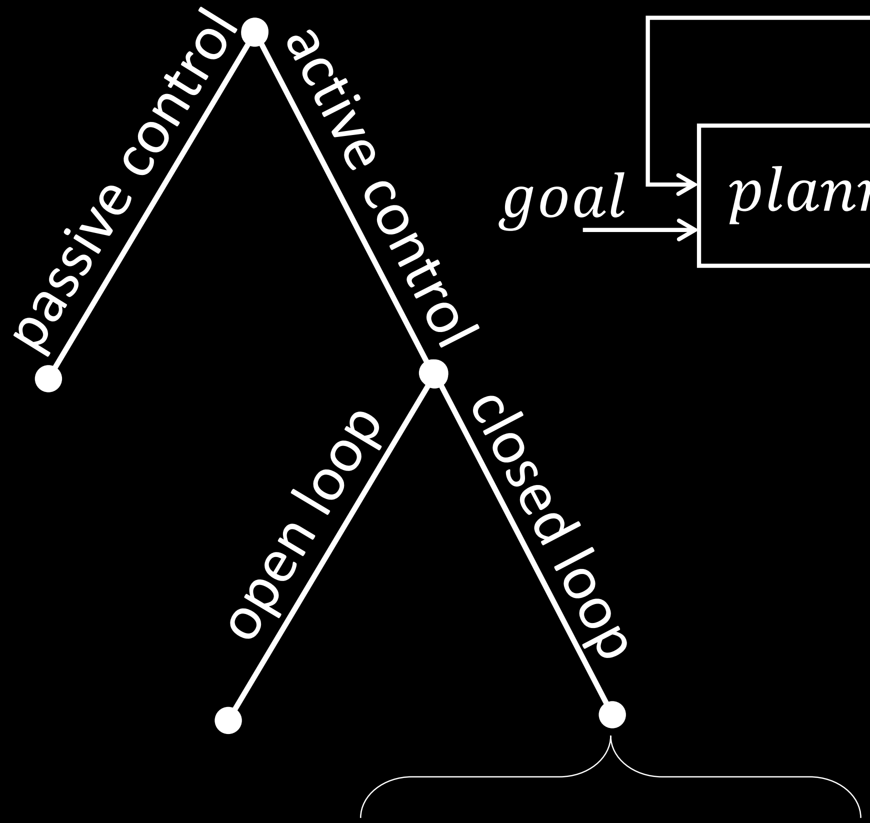
- Combine base with processor, drivers, and sensors
- Pros/cons of sensor modalities and types of sensors
  - Noise, bias, and sampling frequency
- Simulation platform
  - Sensor models
  - Motion models



What are sources of error?

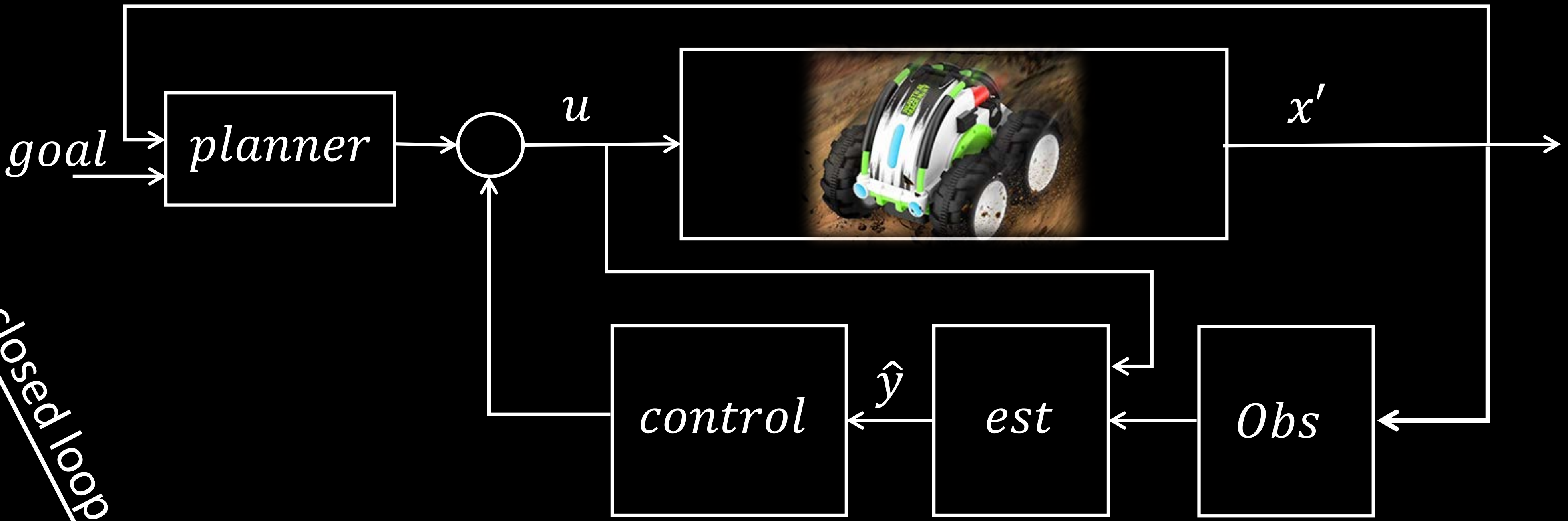
- Skid steering
- Momentum
- Weak motors

Part 1: Implement the Robot



- processor
- drivers
- limits

- sensors
- noise/bias

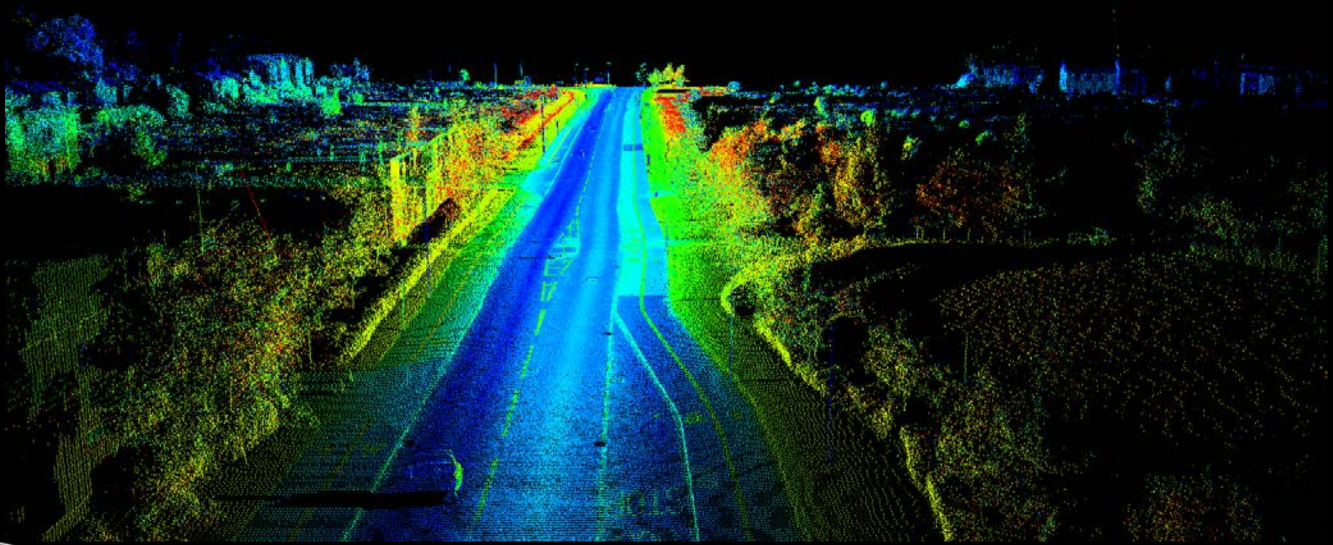


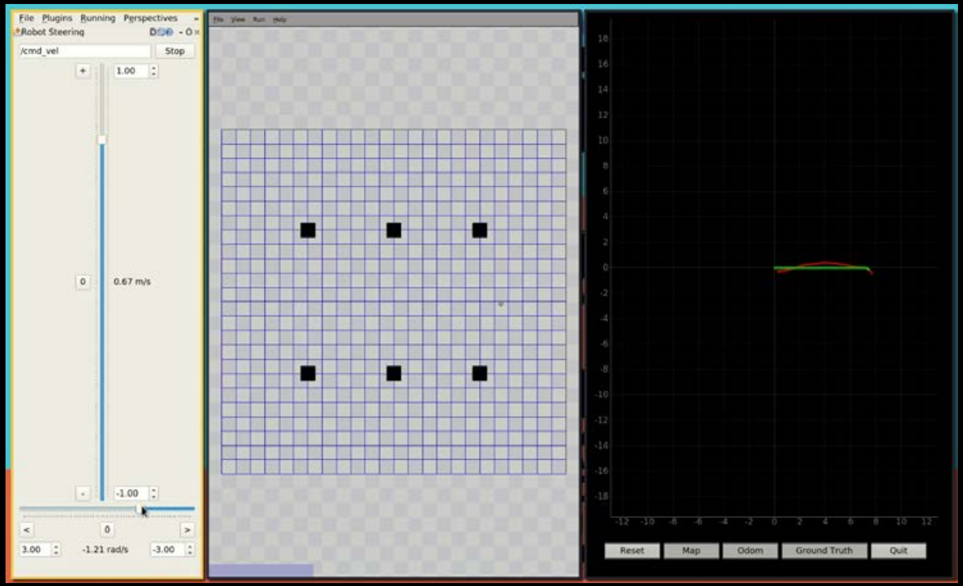
Why feedback?

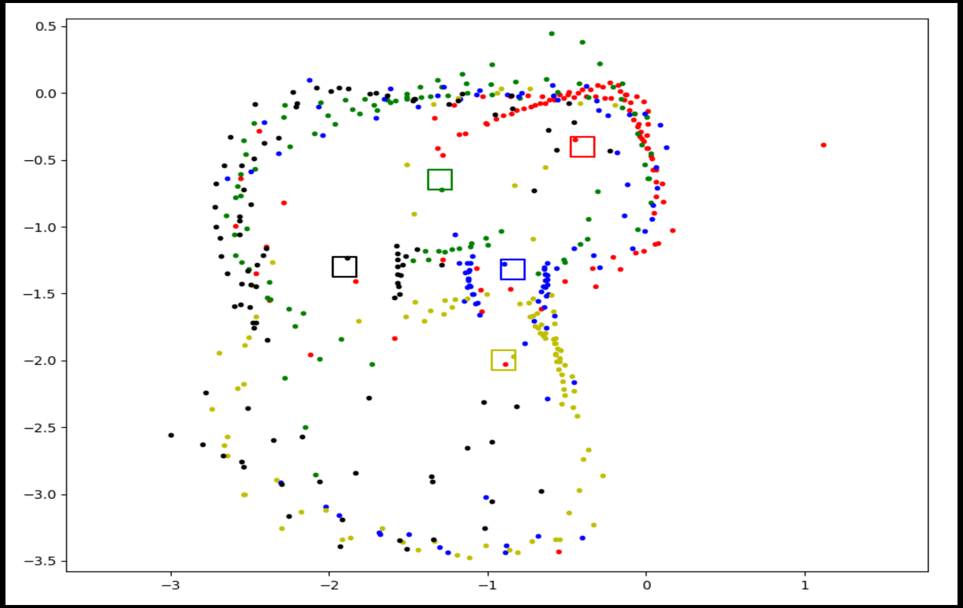
- System uncertainty
- Instability
- Disturbances
- Efficiency

Part 2: Enabling fast navigation (kinematic → dynamic)

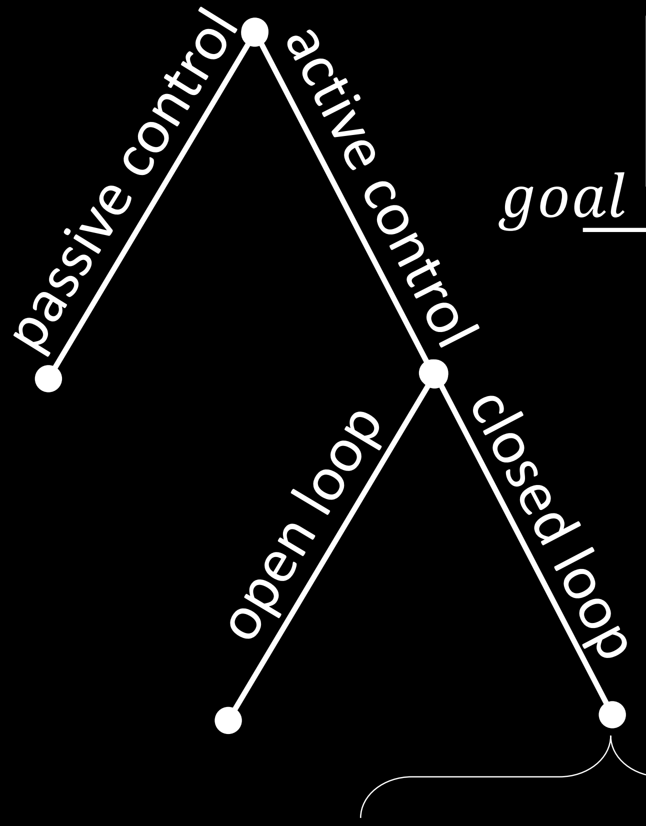
- Open loop navigation
- Obstacle avoidance
- PID control
- Map
- Localization
- Trajectory planning

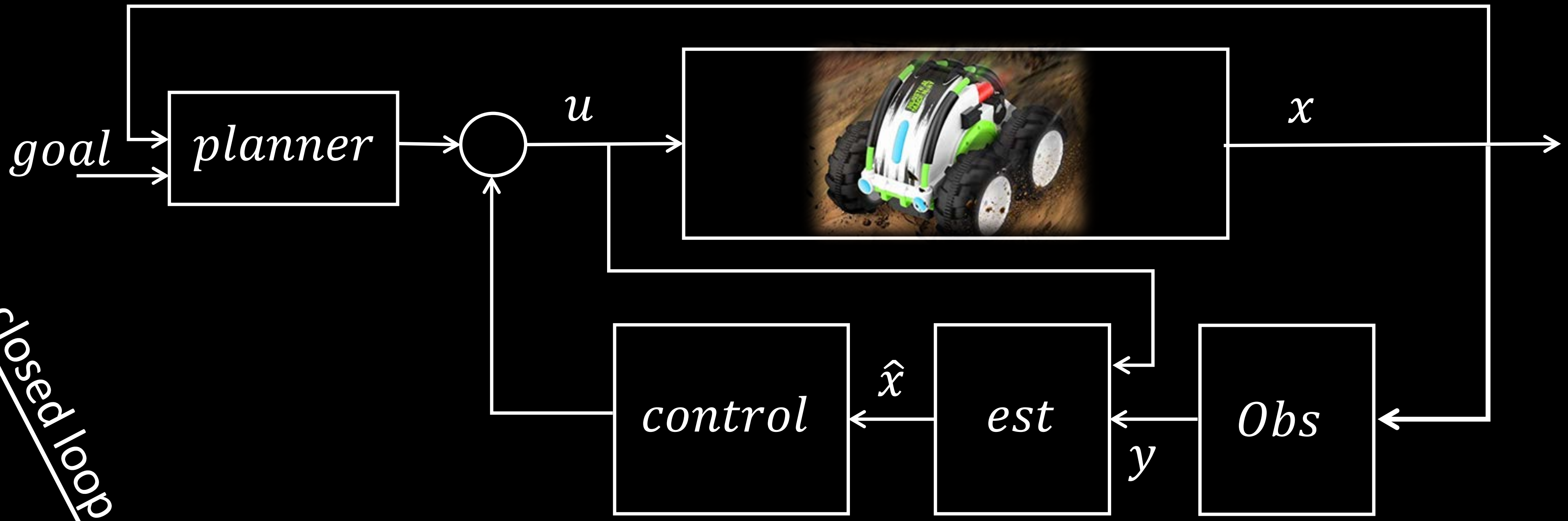






Part 2: Enabling fast navigation (kinematic → dynamic)





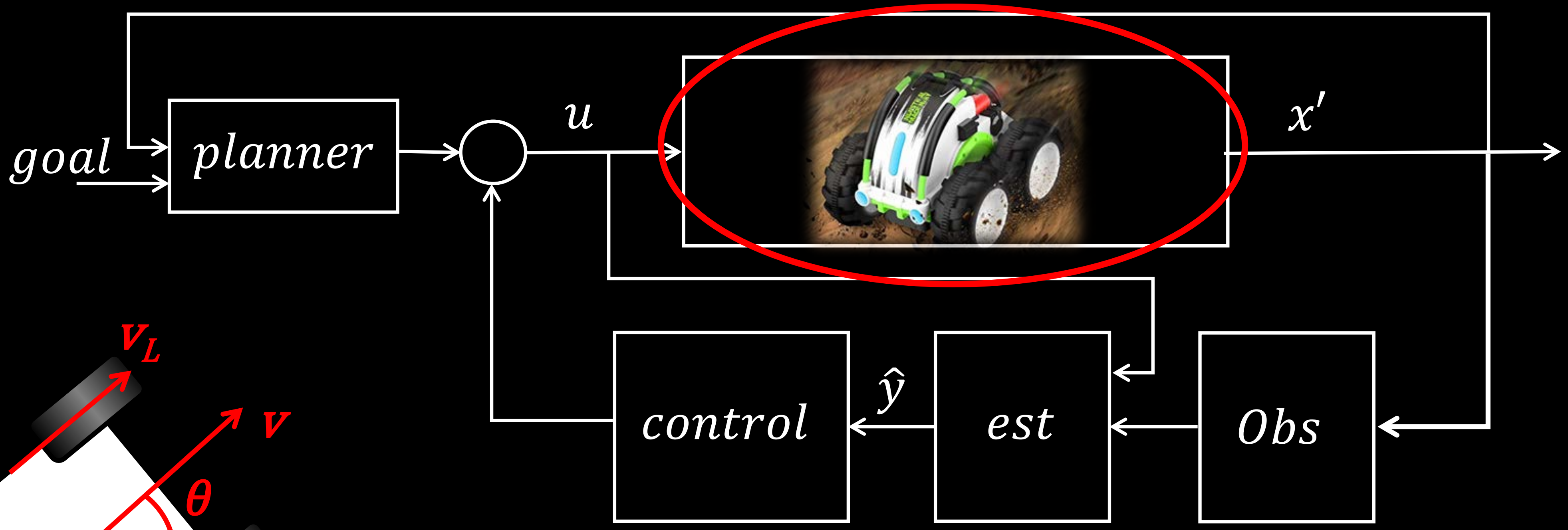
- processor
- drivers
- limits

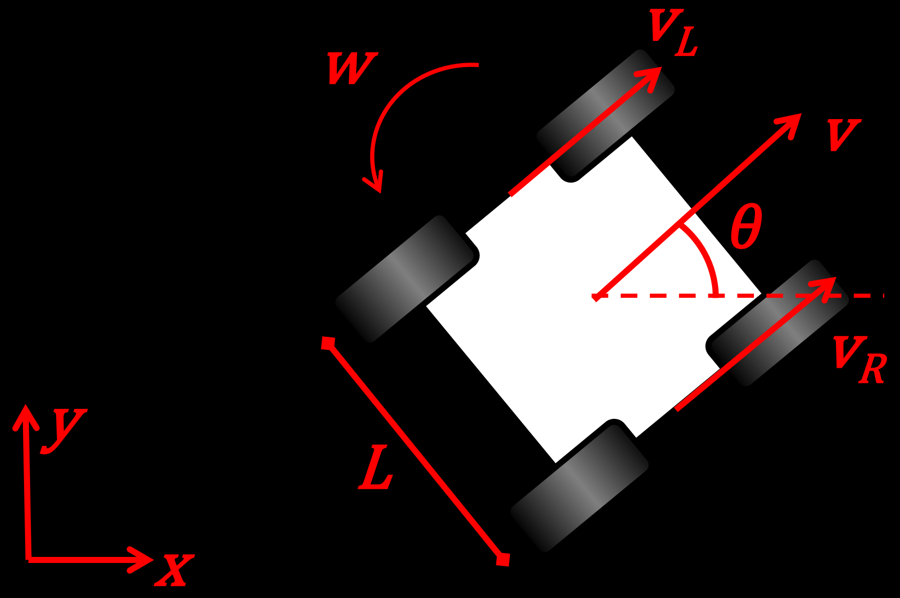
- sensors
- noise/bias

Why feedback?

- System uncertainty
- Instability
- Disturbances
- Efficiency

Part 2: Enabling fast navigation (kinematic → dynamic)



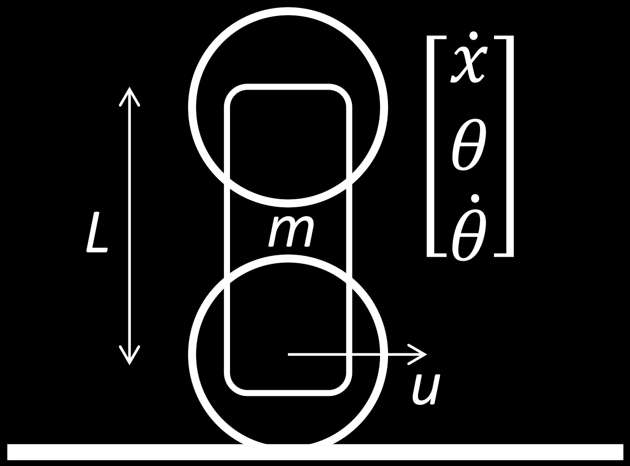


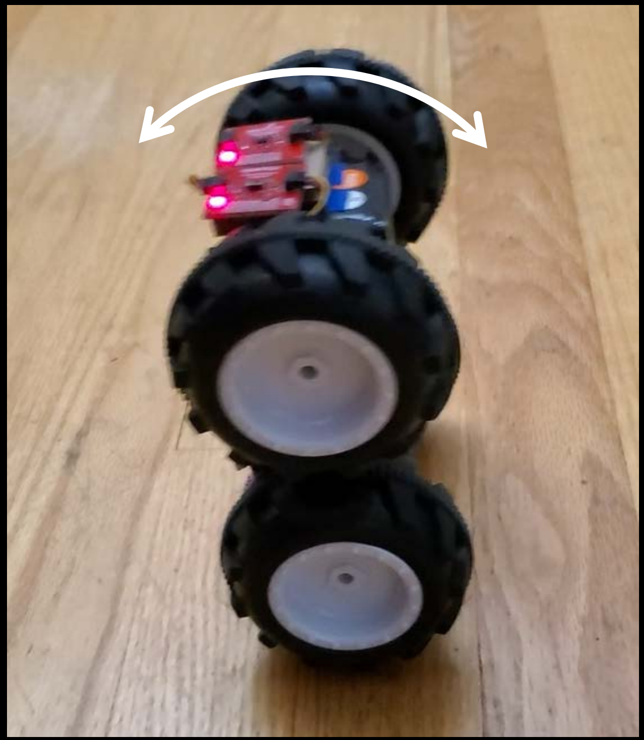
$$\dot{x} = cos(\theta)v$$
$$\dot{y} = sin(\theta)v$$
$$\dot{\theta} = w$$

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

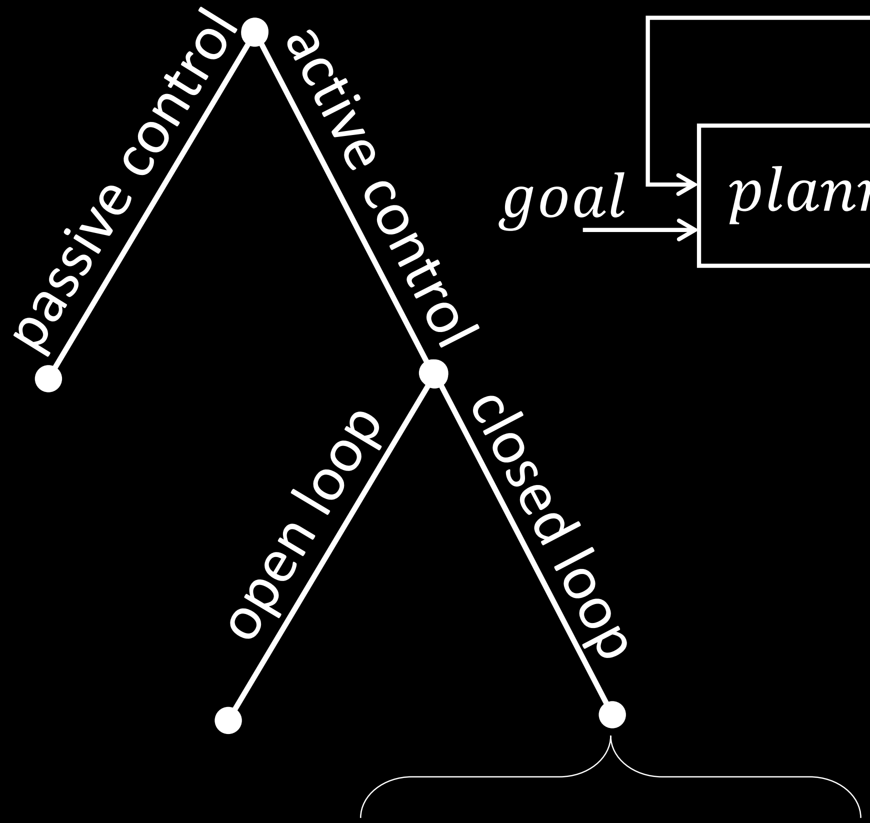
Part 3: Controlling an unstable system

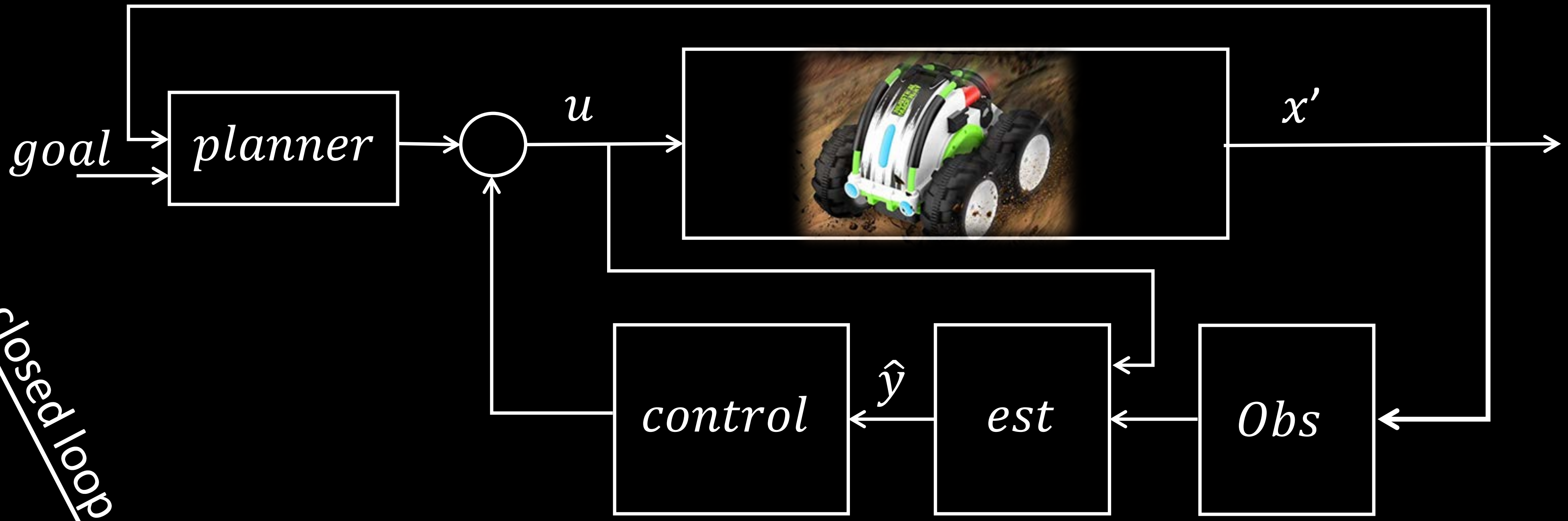
- Inverted pendulum
- Equations of motion
- Linear systems
- Controllability
- Observability/estimators
- LQR optimal control





Part 3: Controlling an unstable system





- processor
- drivers
- limits

- sensors
- noise/bias

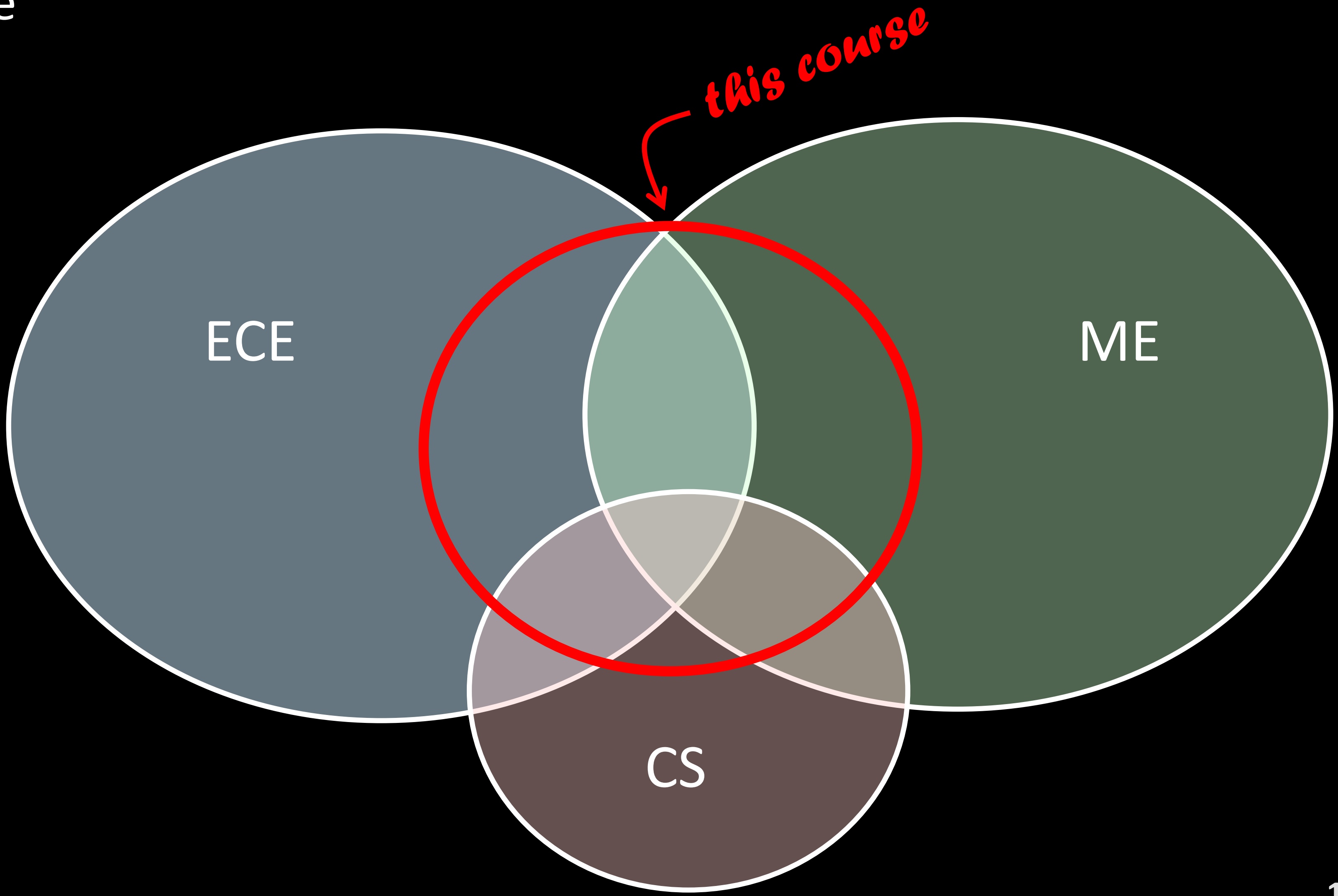
Why feedback?

- System uncertainty
- Instability
- Disturbances
- Efficiency

Course Objective

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

- Overlap with Autonomous Mobile Robots and Feedback Control Systems



Tentative **Schedule**

- Week 1: Intro ……………………………………………………………………………….
- Week 2: Rigid body transformations ………………………………………………….
- Week 3: Sensors - TOF, Proximity, encoders, IMU ………………………………
- Week 4: Noise, probability and estimation ………………………………………….
- Week 5: PID control, Mapping……………………………………………………….
- Week 6: Bayes filter, odometry/sensor models ………………………………….
- Week 7: Localization ……………………………………………………………………….
- Week 8: Planning, A* search, probablistic roadmap …………………………..
- Week 9: Linear systems and state space representation ……………………….
- Week 10: Inverted pendulum dynamics ……………………………………………...
- Week 11: Controllability (LQR) ……………………………………………………………
- Week 12: Observability, Kalman filter (LQE) ………………………………………….
- Week 13: Kalman filter (LQE) ……………………………………………………………….
- Week 14: LQG control on an inverted pendulum ……………………………….
- Week 15: Guest lectures ……………………………………………………………………
- Week 16: Recap ……………………………………………………………………………….

Sensors, data interpretation (Weeks 2–4)

Trajectory planning and execution (Weeks 5–8)

Control of an unstable system (Weeks 9–14)

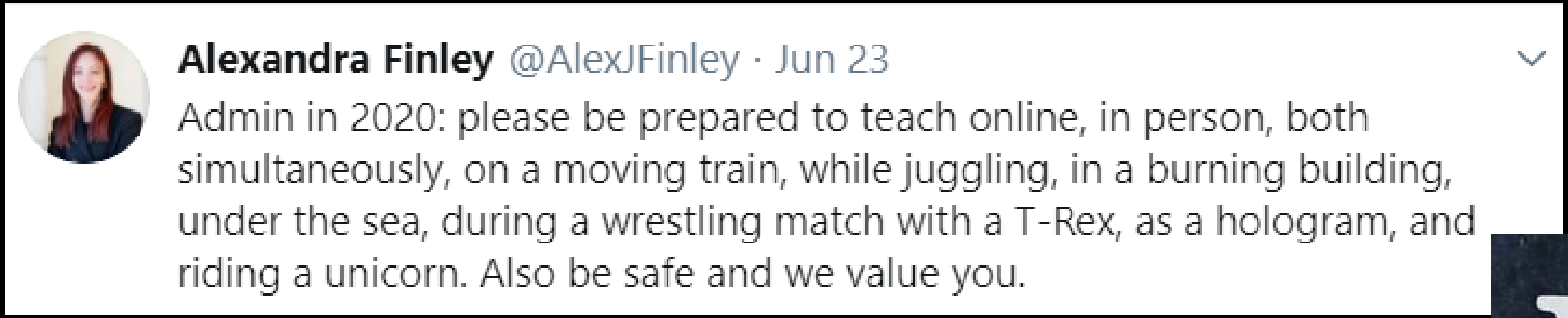
Real life examples (Weeks 15–16)

Tentative Schedule

- Week 1: Intro .... Github page
- Week 2: Rigid body transformations .... Artemis nano
- Week 3: Sensors - TOF, Proximity, encoders, IMU .... Bluetooth communication
- Week 4: Noise, probability and estimation .... Characterize your car
- Week 5: PID control, Mapping.. Open loop control
- Week 6: Bayes filter, odometry/sensor models .... Obstacle avoidance
- Week 7: Localization .... IMU
- Week 8: Planning, A* search, probablistic roadmap ....Odometry
- Week 9: Linear systems and state space representation ....PID control
- Week 10: Inverted pendulum dynamics ....Mapping
- Week 11: Controllability (LQR) ....Localization
- Week 12: Observability, Kalman filter (LQE) ....Planning
- Week 13: Kalman filter (LQE) .... -
- Week 14: LQG control on an inverted pendulum .... -
- Week 15: Guest lectures .... Inverted pendulum
- Week 16: Recap ....

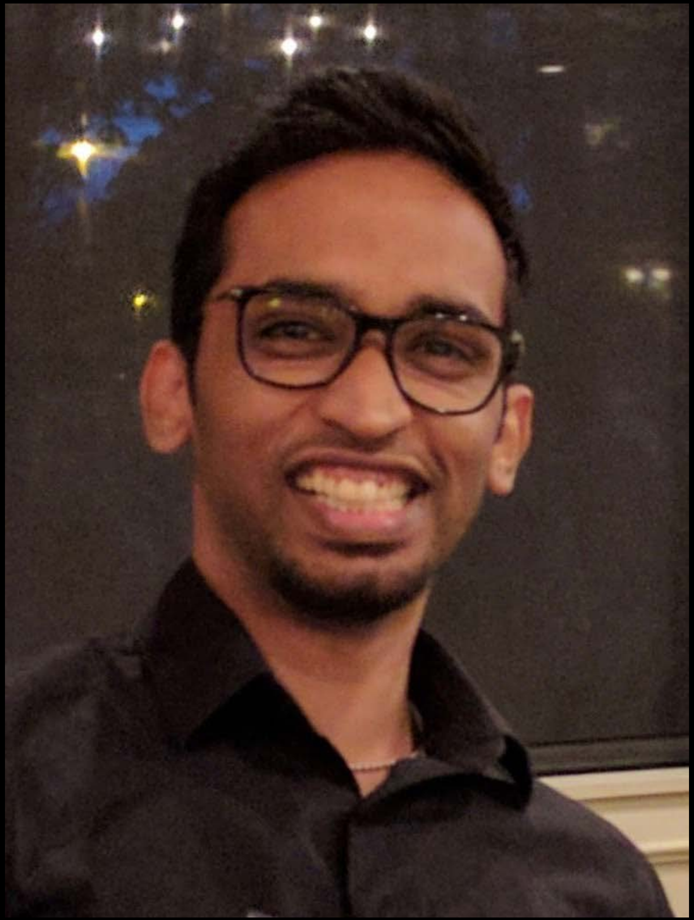
Disclaimer!

- *First offering, all online!*
  - 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
  - We are aiming high!

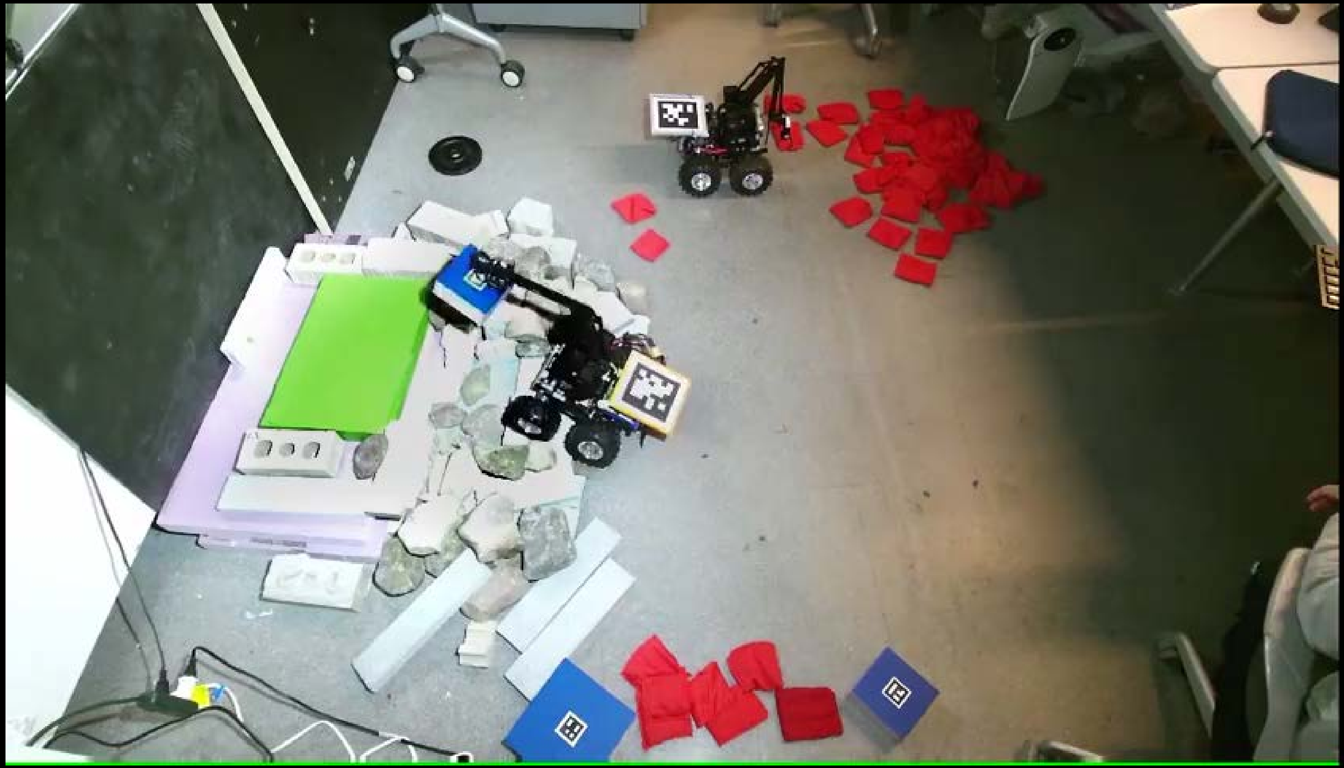

Alexandra Finley @AlexJFinley · Jun 23
Admin in 2020: please be prepared to teach online, in person, both simultaneously, on a moving train, while juggling, in a burning building, under the sea, during a wrestling match with a T-Rex, as a hologram, and riding a unicorn. Also be safe and we value you.

Fast Robots
Teaching Team

Your Teaching Team: Vivek (‘we wake’) Thangavelu



Hi, this is Vivek and he likes to talk in the third person.
He considers himself to be a field roboticist (a fancy term used to persuade oneself for lacking expertise in any one specific area) and wants to help build colonies that are not on earth (**insert futurama meme**\). He currently designs robots that can build structures with found stones, and hopefully someday teach them to love.



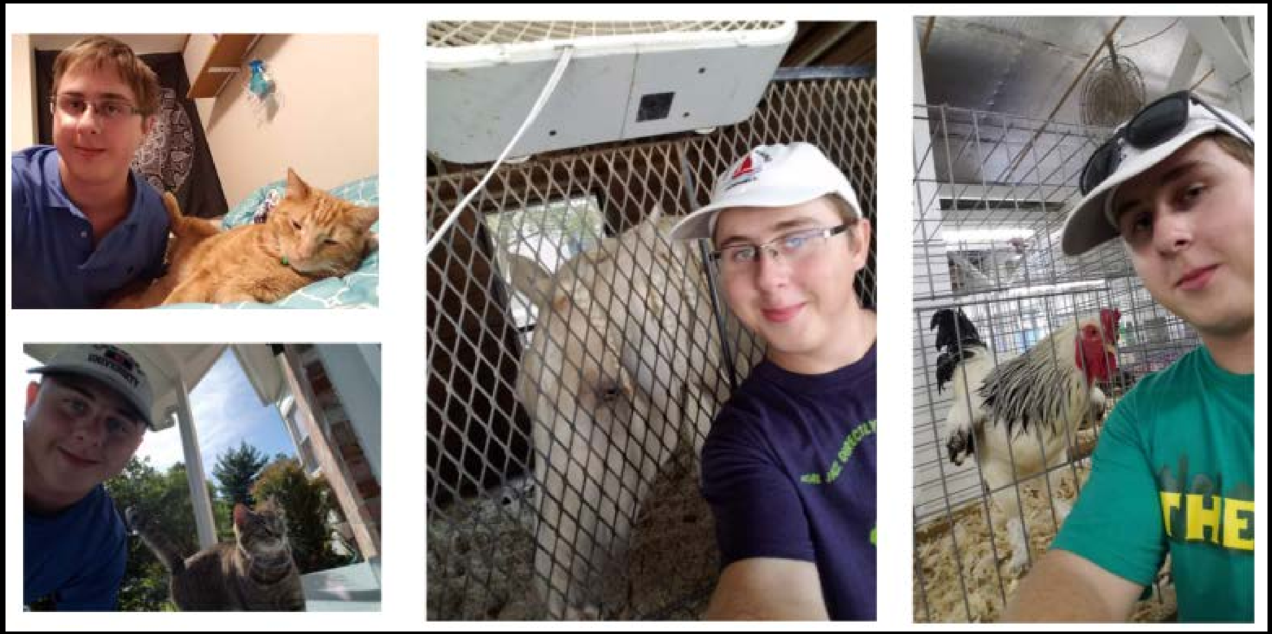
Your Teaching Team: Sadie Cutler

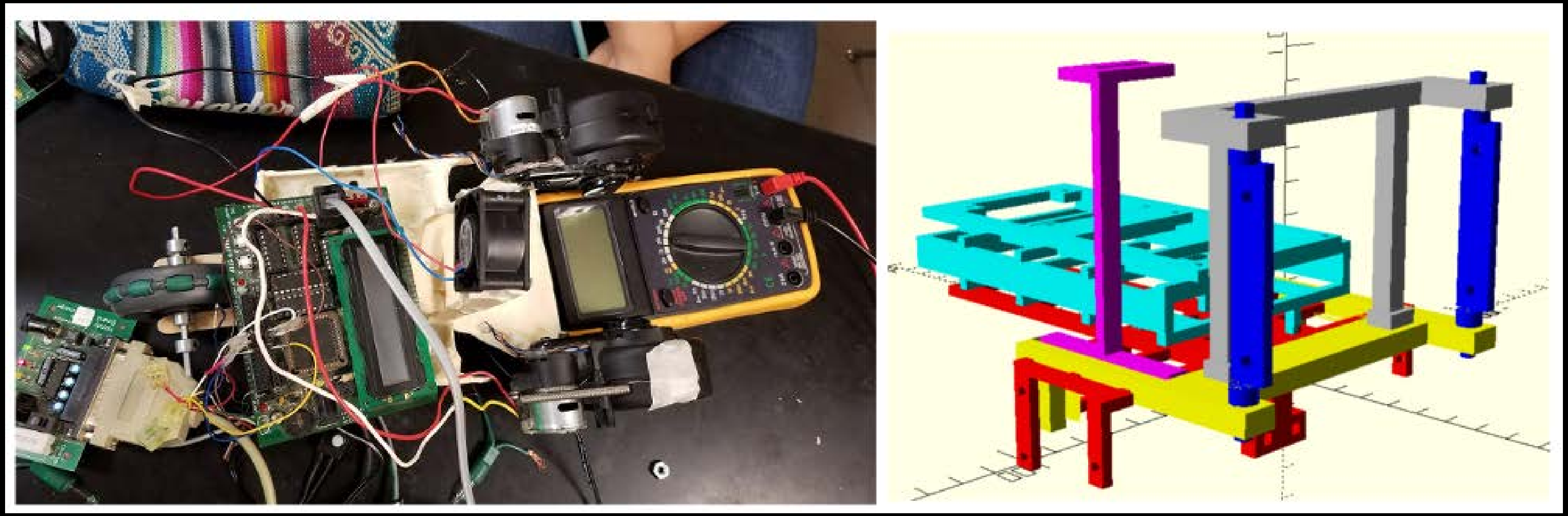
- Second semester graduate student in the CEI-lab
- Research focus is on robotic yield-improvement of pollen-limited crops
  - Fluent in German
  - Celebrate Harry Potter’s bday
  - Enjoy triathlons and Jiu Jitsu

A growing soft robot, Sadie worked with at Stanford

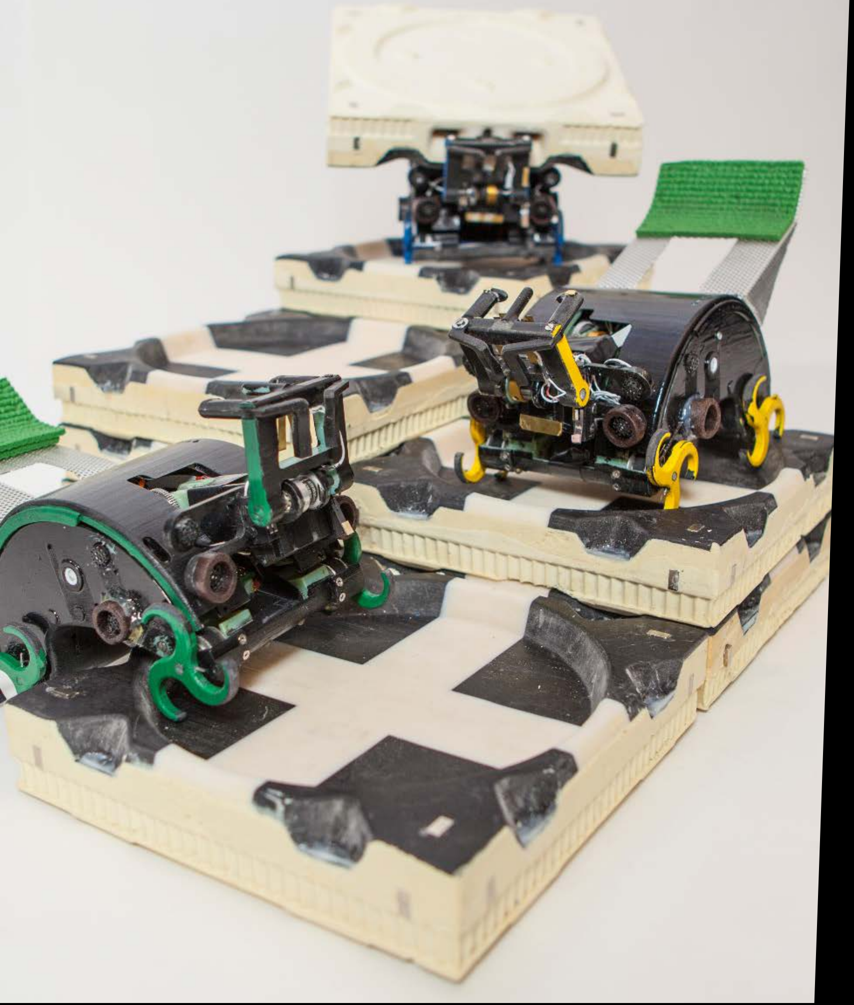
Your Teaching Team: Alex Coy

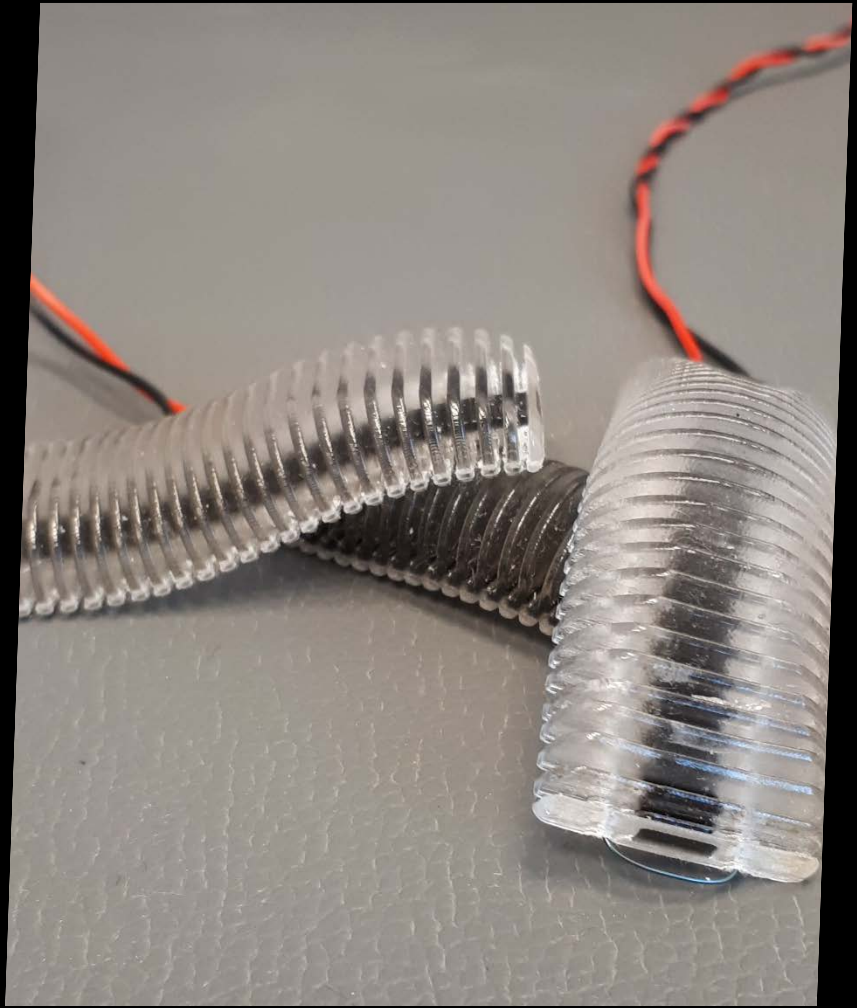
https://alexcoy.duckdns.org/





Your Teaching Team: Kirstin Petersen

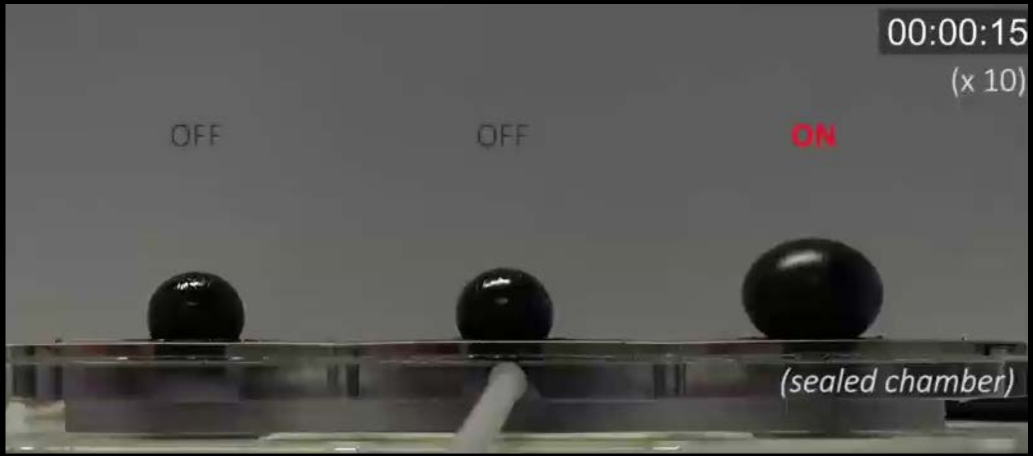


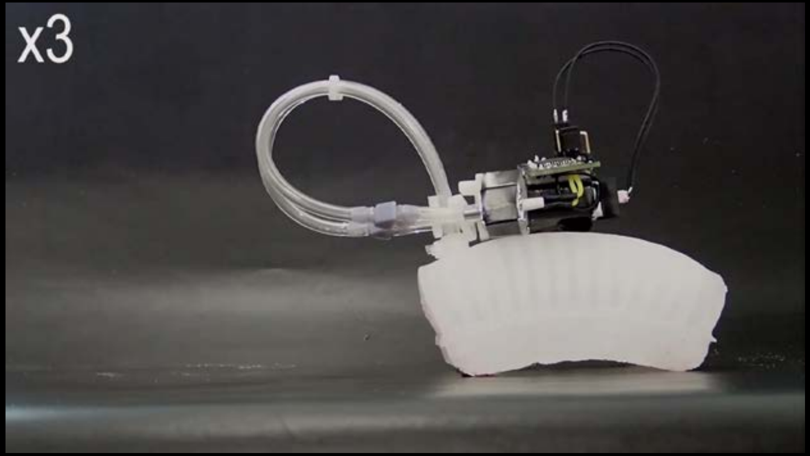


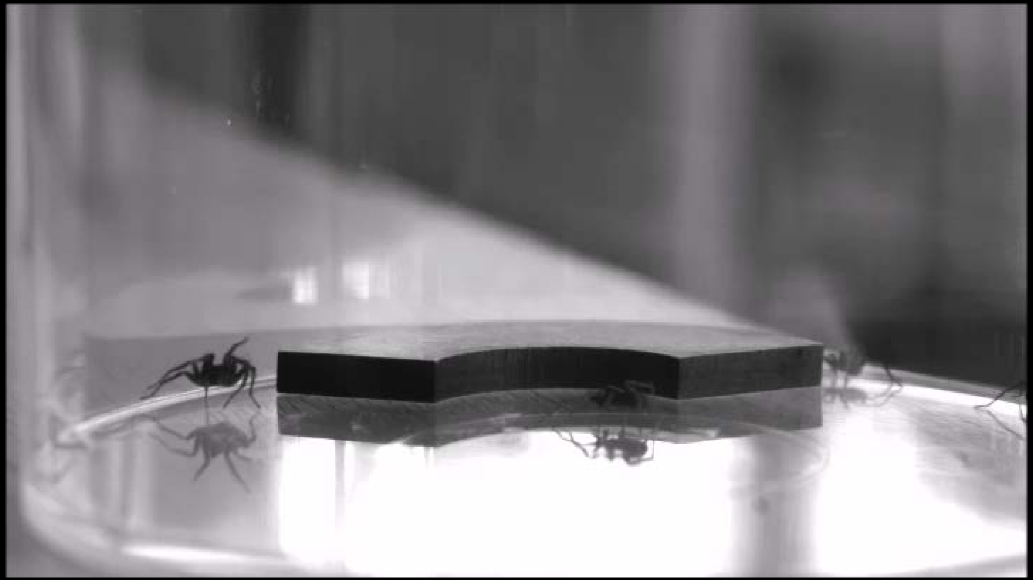
Collective Embodied Intelligence lab (www.cei.ece.cornell.edu)

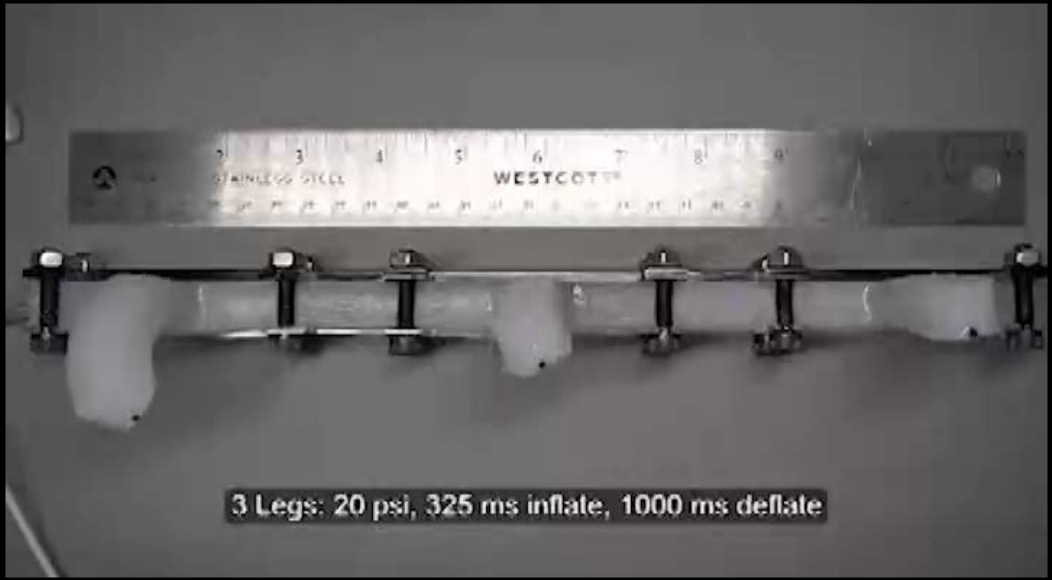
Your Teaching Team: Kirstin Petersen

Your Teaching Team: Kirstin Petersen









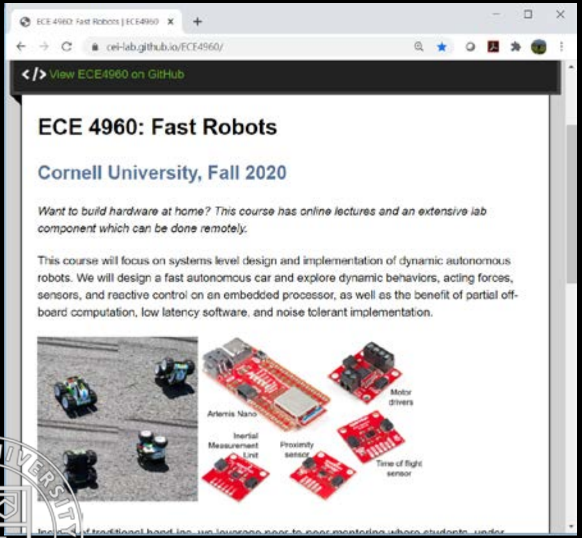
Your Teaching Team: Kirstin Petersen

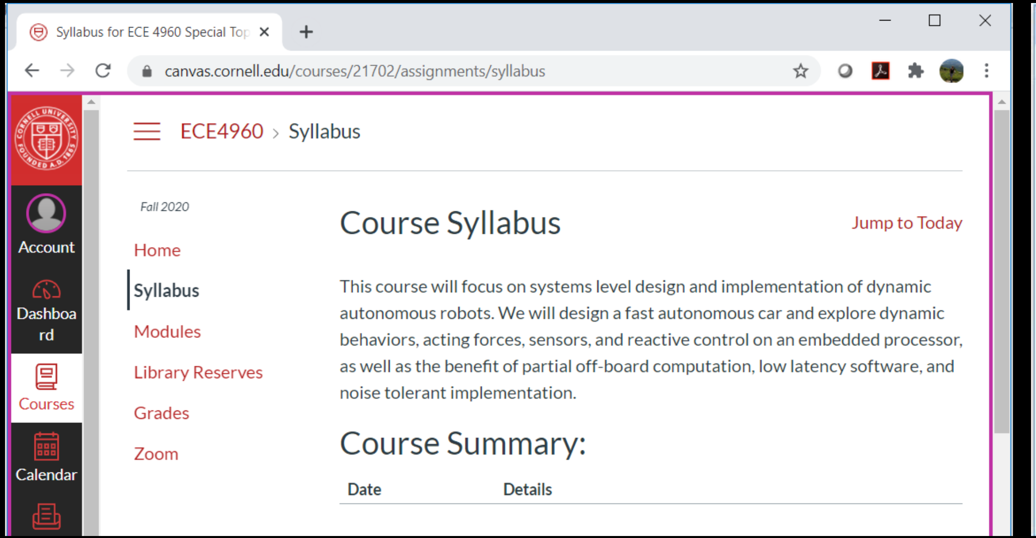
Your Teaching Team: Kirstin Petersen

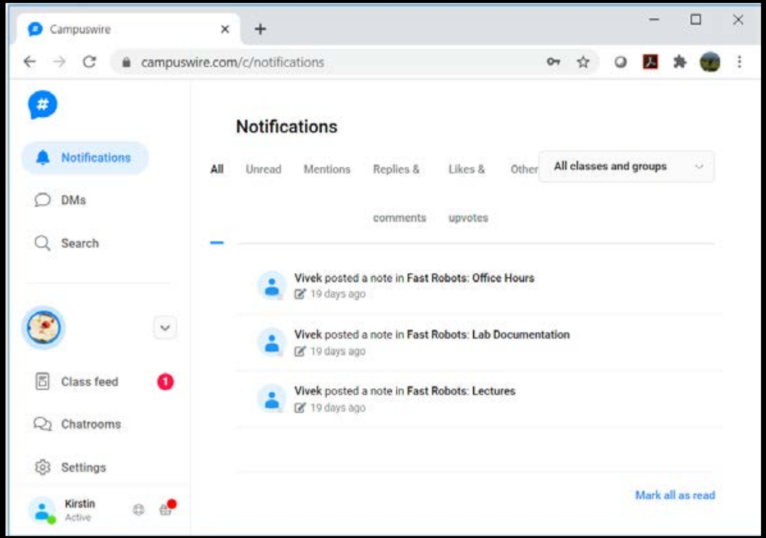
Fast Robots
Logistics

Logistics I

- Main page: *Github* (https://cei-lab.github.io/ECE4960/)
  - Schedule, lecture slides, lab documents, tutorials, code examples
- University regulations: *Canvas*
  - Lecture slides, lab documents, zoom-links, grades
- Quick questions/discussions: *Campuswire*
  - If you don’t get an invite, please sign up here: https://campuswire.com/p/GBD54AB15







Logistics II

- Lab kit
  - On campus
    - Pick up your kit outside of Rhodes Hall, Friday 4th (email: kirstin_at_cornell.edu) to arrange a time
  - Off campus
    - Send me your address ASAP for shipping
    - First three labs can be done with a partner who has a kit if need be
  - You'll need a no. 1-50mm flathead screwdriver, a wire cutter, a flat surface
- Lab software
  - Linux (kernel 3.0.34+), MacOS 10.10+, and Windows 10
  - Processor: Core i3-8100 3.6 Ghz/AMD Ryzen 5 1400 or equivalent, Memory: 4 GB RAM, Free Space: 10 GB

Logistics III

- All labs can be done *remotely*!
- Lab reports → Your own Github sites (check out examples from ECE3400 here)
- Labs
  - If you are on campus, you can get access to PH427 (one person at a time!)
  - If something breaks…
    - Contact us ASAP
    - You can try out solutions remotely on TA kits (limited time)
    - You can get partial credits for simulations
  - If you run low on time…
    - Everyone gets 7 days of extensions they can use as they wish
    - They MUST inform the teaching team before the deadline
  - *¡The car has limited battery life, do the labs over multiple days!*

Logistics IV - Grading

- Labs (85 pts)
  - Points for solution (65%)
  - Points for write-up (25%)
  - Points for speed (10%)
- Quizzes (10 pts)
- Campuswire participation (5 pts)
- Grading policy
  - Collaboration is welcome
  - But implement your own code
  - Always credit collaborators/references
- Extension policy
  - 7 days extension
  - No questions asked
  - Only if you notify us before the deadline

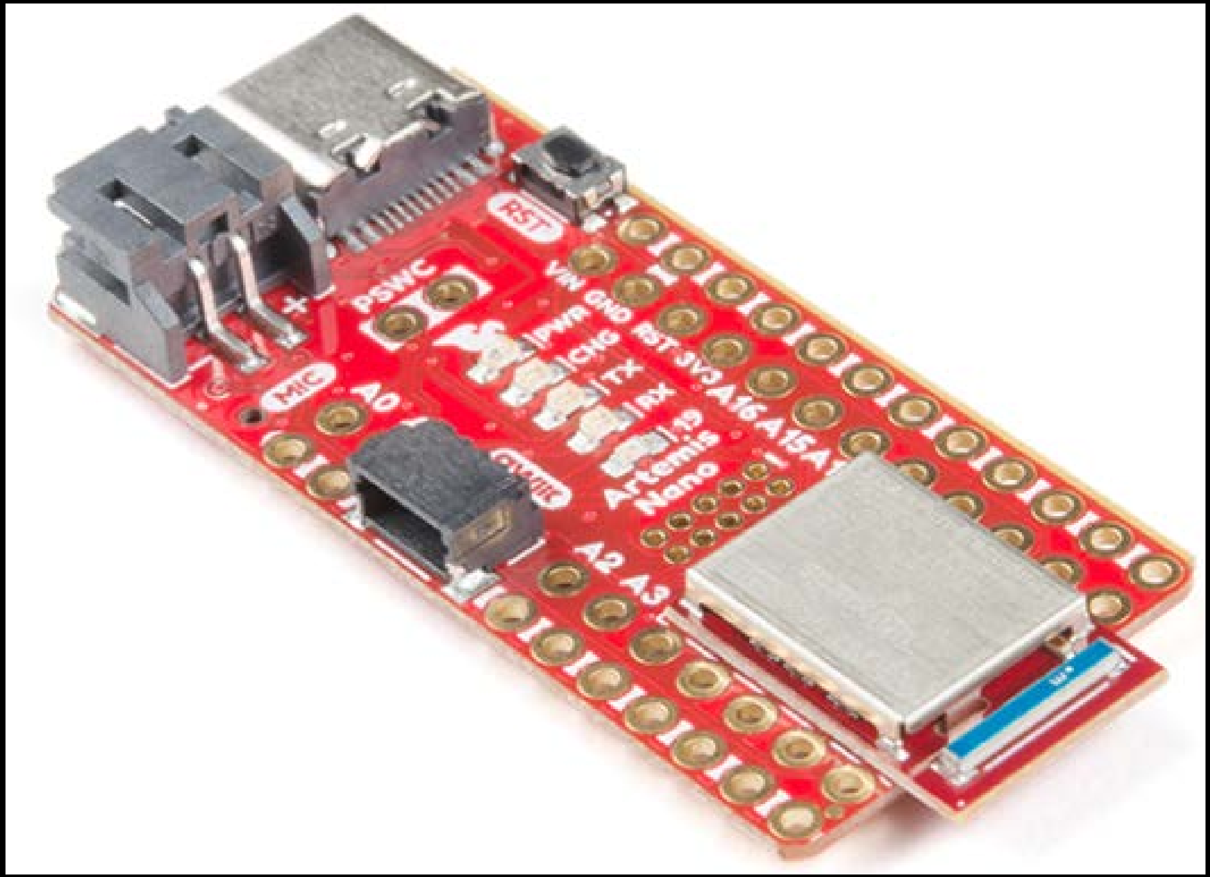
Task	pts
Lab 1 write-up (Artemis)	5
Lab 2 write-up (Bluetooth)	5
Lab 3 write-up (RC car test)	5
Lab 4 write-up (Open loop)	5
Lab 5 write-up (Obstacle avoidance)	10
Lab 6 write-up (IMU)	5
Lab 7 write-up (Odometry)	5
Lab 8 write-up (Mapping)	5
Lab 9 write-up (Localization)	10
Lab 10 write-up (Planning)	10
Lab 11 write-up (PID control)	10
Lab 12 write-up (Inverted pendulum, LQR control)	10
Quizzes	10
Campuswire participation	5
Bonus points for course evaluation!	2
Total:	**102**

Action items

- *If you decide not to take the course, let Kirstin know ASAP*
- Deadline 4pm Friday Sept 4th
  - Pick up your kit or send us your mailing address
- Deadline 8am Tuesday Sept 7th
  - 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
- Deadline 8am Monday Sept 14th
  - Upload your write-up of Lab 1
- *Thursday Sept 10th lecture:* installation support
  - Try VMware installation before this lecture
  - If you can't make this section and run into trouble, contact us over Campuswire

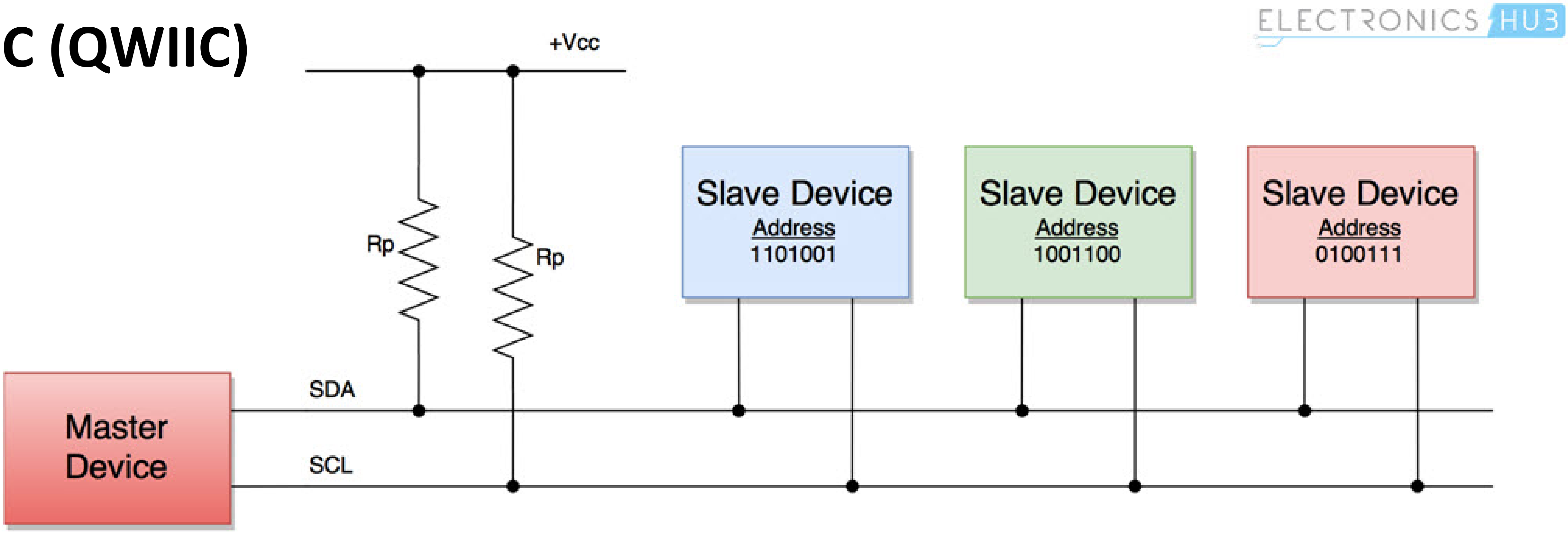
Fast Robots
Lab 1: Artemis

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

I2C (QWIIC)


ELECTRONICS HUB
+Vcc
Rp
Rp
Slave Device
Address
1101001
Slave Device
Address
1001100
Slave Device
Address
0100111
Master Device
SDA
SCL

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

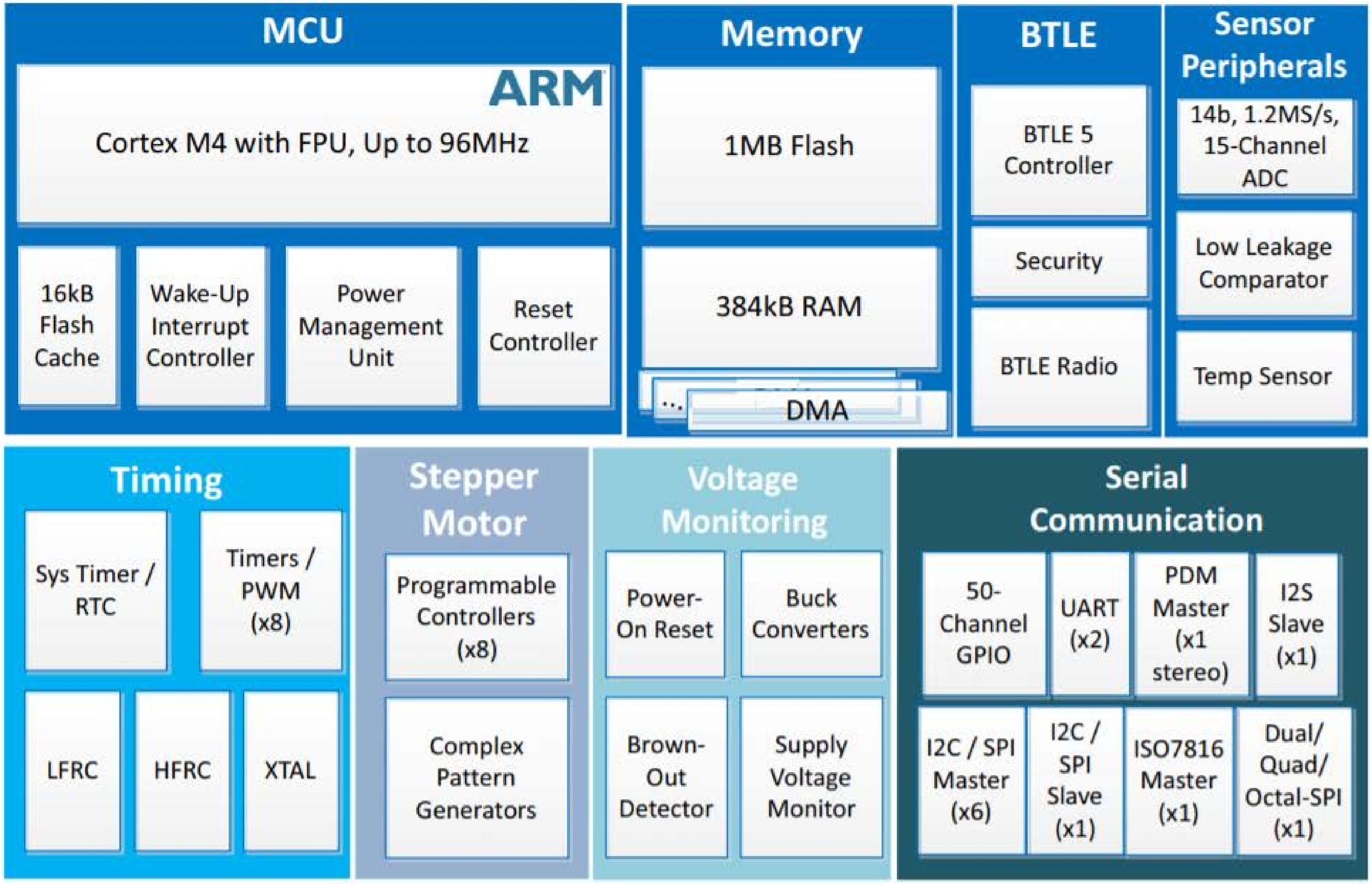


Figure 3. Block Diagram for the Ultra-Low Power Apollo3 Blue MCU