ECE 4960

Prof. Kirstin Hagelskjær Petersen
kirstin@cornell.edu

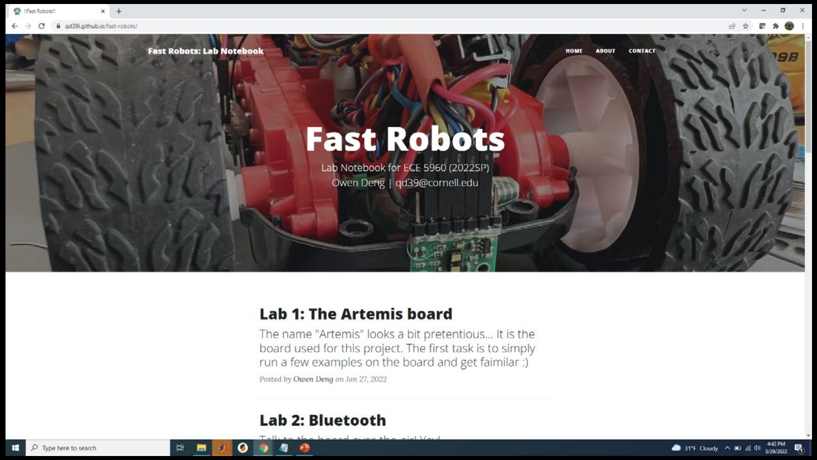
Fast Robots

Labs 1-8 recap and class feedback

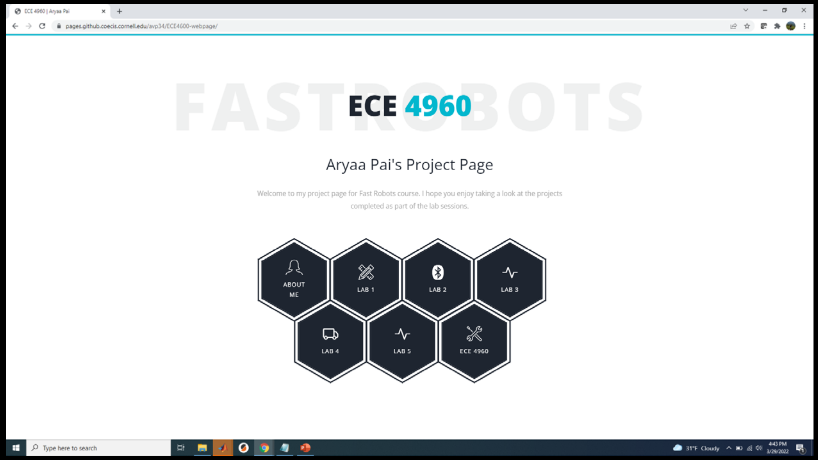
Lecture overview

- Recap from the dynamics portion of the class
  - Lectures
  - Labs
- Outlook on the rest of the semester
  - Loopholes
- Mid-semester feedback
  - Suggestions for improvements
- Please team up…

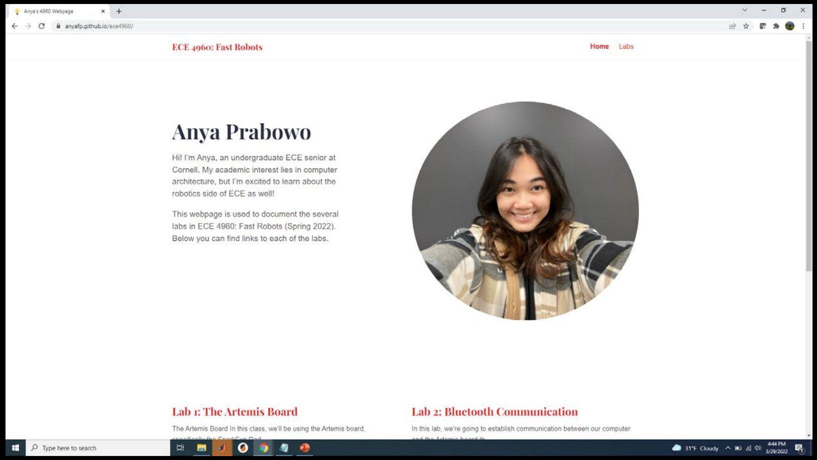
Lab 0 - Websites



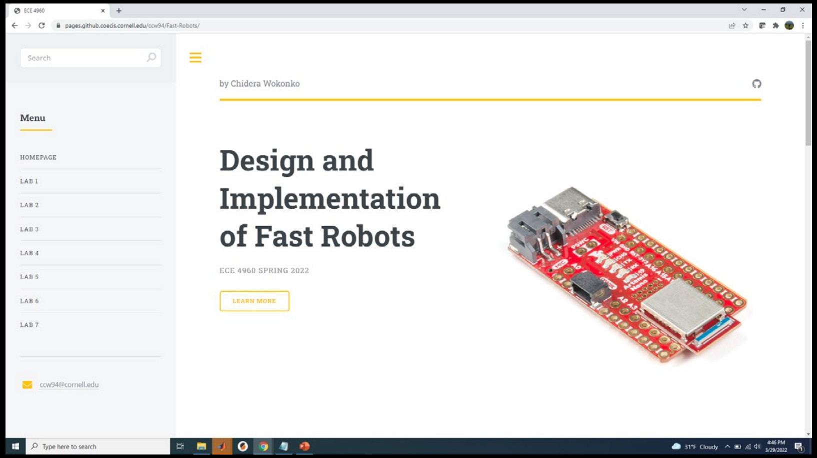
Owen Deng



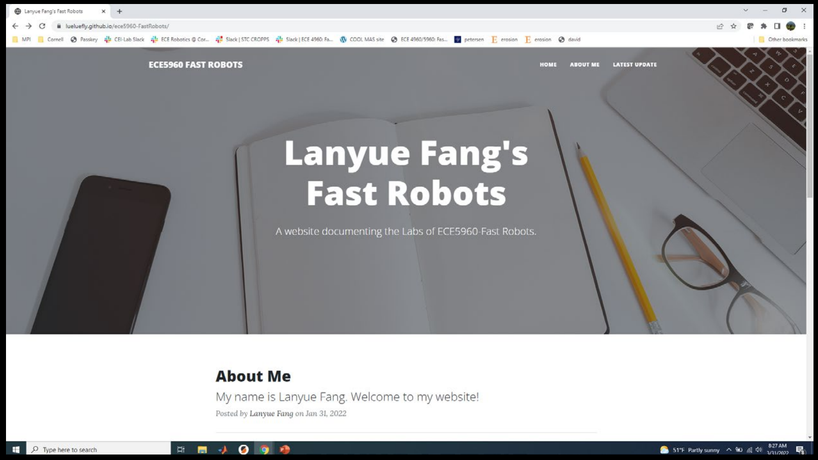
Arya Pai



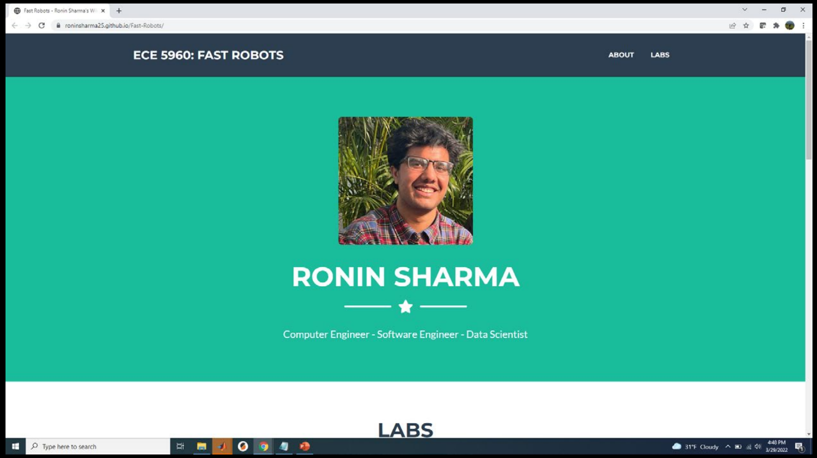
Anya Prabowo



Chidera Wokonko



Lanye Fang



Ronin Sharma

Lab 0 - Websites

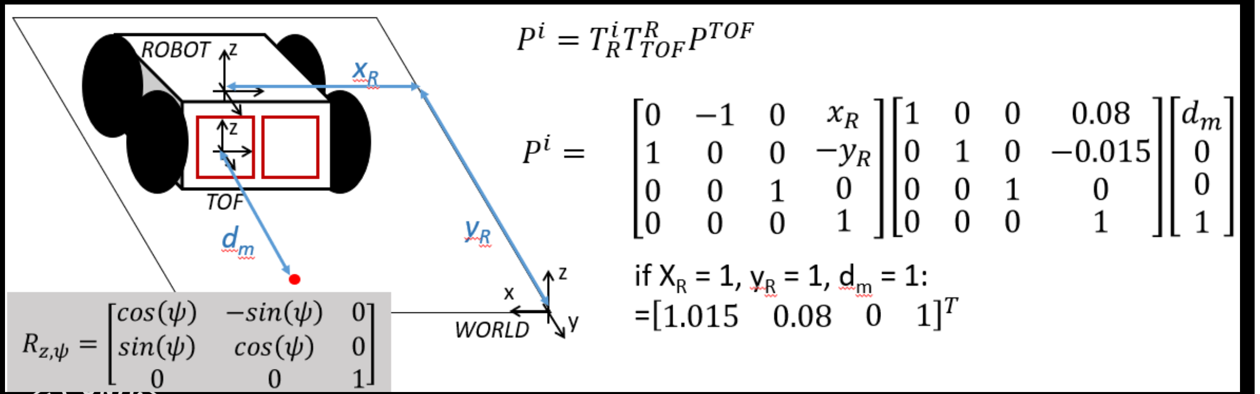
What do you like about these websites?

- Clear structure
- Easy navigation
- Bandwidth friendly
- Who are you?
- Why should the audience keep reading?
- Content freshness

…Consider updating your website

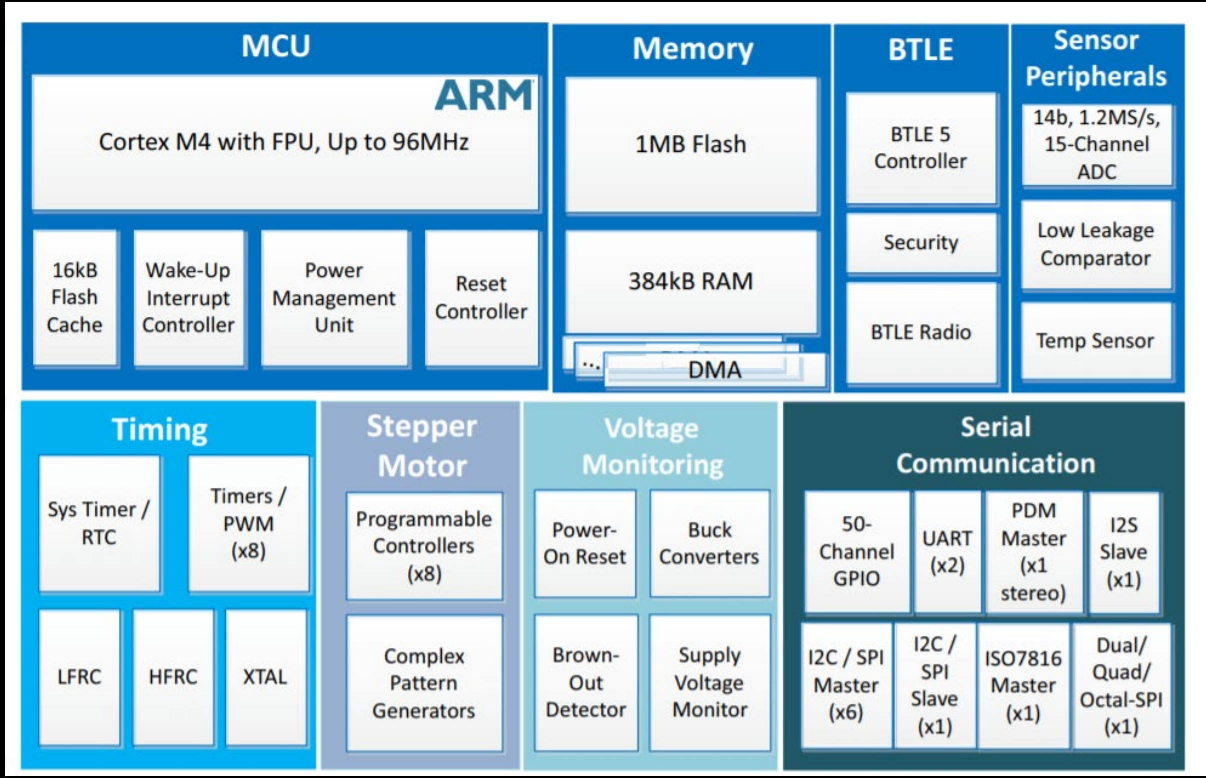
Module 1 - Intro

- Artemis
- Data types
- I2C
- Configuration space
- Transformation matrices
  - Rotation matrices
  - Translation matrices



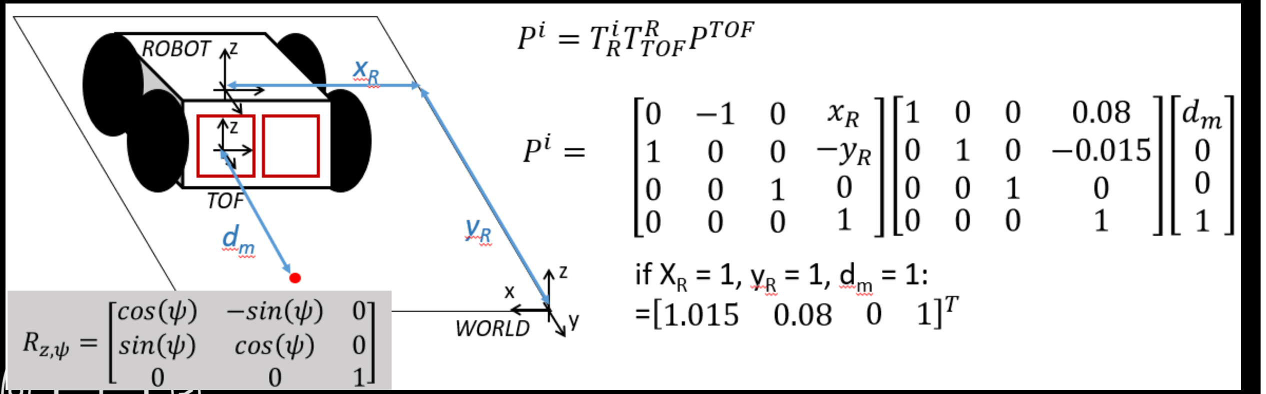
Lab 1 – Artemis

- Program board/Arduino IDE
- Debugging LED/serial messages
- Onboard temperature sensor, microphone FFT



Module 1 - Intro

- Artemis
- Data types
- I2C
- Configuration space
- Transformation matrices
    - Rotation matrices
    - Translation matrices

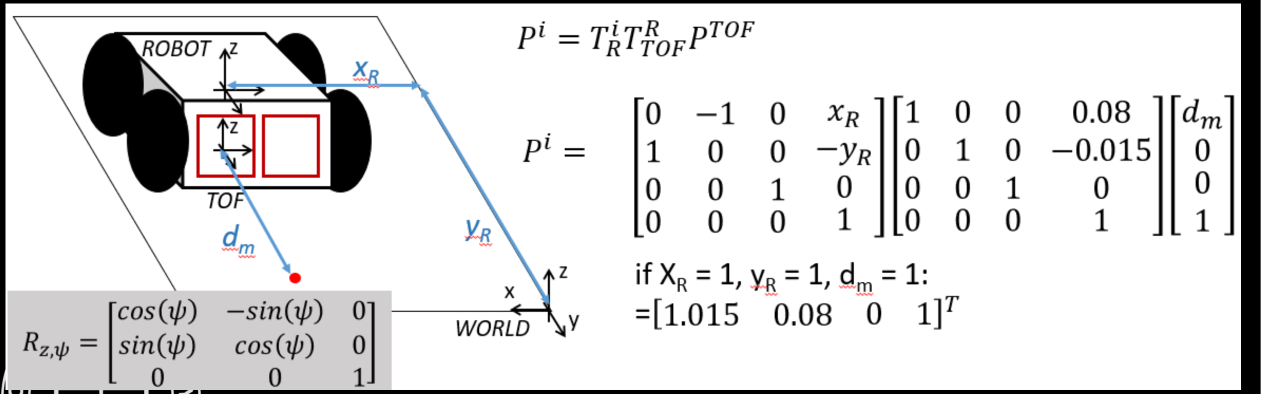


Lab 2 – Bluetooth

- Python virtual environment
- Jupyter lab
- Bluetooth, Low Energy (BLE)
- Sending commands to the robot
- Receiving floats and strings from the robot
- Effective data rate
- Reliability

Module 1 - Intro

- Artemis
- Data types
- I2C
- Configuration space
- Transformation matrices
  - Rotation matrices
  - Translation matrices



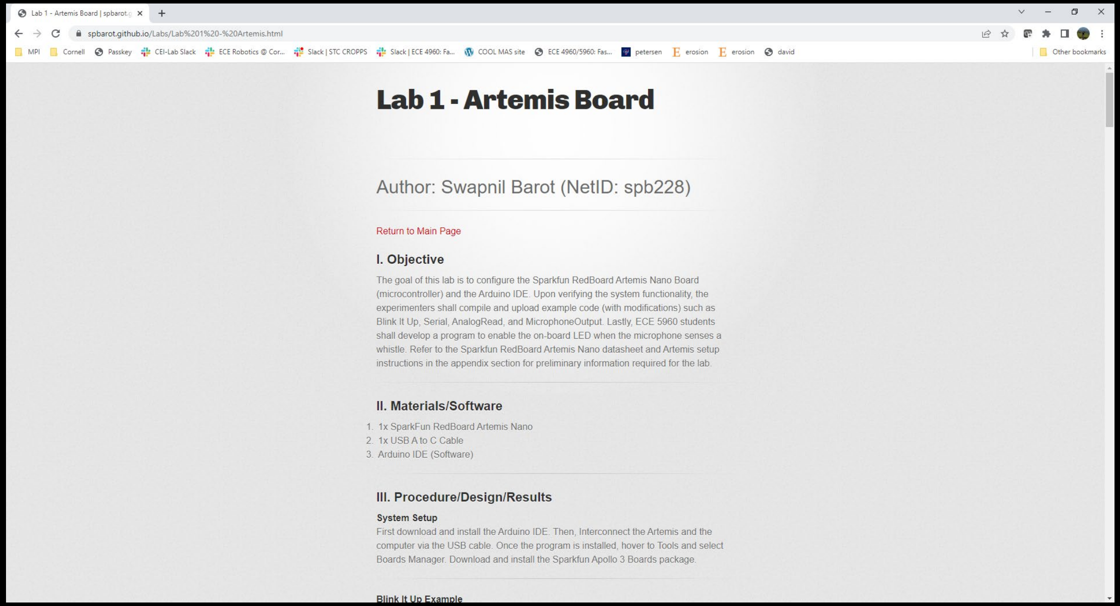
Lab 2 – Bluetooth

- Conflicting mac addresses
  - *Check EdDiscussion thread #140*

- Media Access Controll (MAC) address
  - Burned into the device
  - Centrally controlled
- Universally Unique Identifiers (UUIDs)
  - Mathematical algorithm
    - Current machine time, MAC address, local IP addresss, and a random number

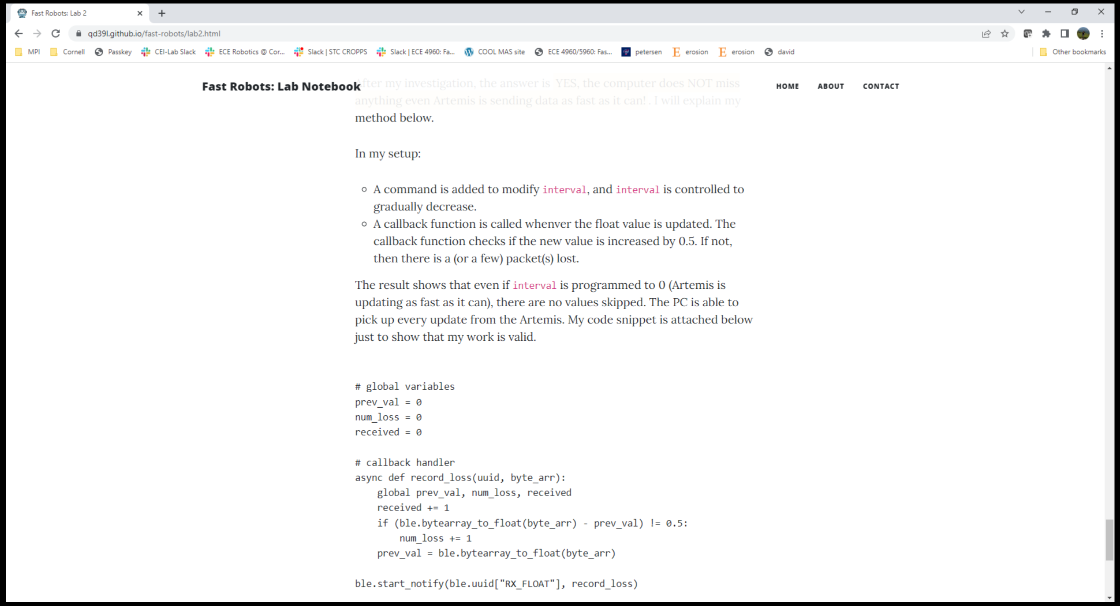
Lab 1 - Artemis

- Swapnil Barot
  - Objective, materials
  - Blink w. explanation for why this will come in handy!
  - Serial program
  - Internal die temperature w. LED output
  - Microphone, FFT w. LED and serial
  - Conclusion



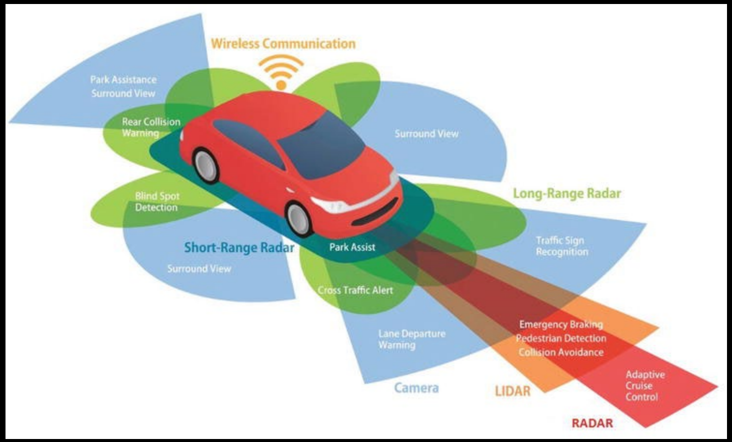
Lab 2 - Bluetooth

- Owen Deng
  - Positive
  - High level ideas
  - Echo with nice message
  - Send three floats (w. pseudocode)
  - Notification handler (w. pseudocode)
  - Float vs string
  - Effective data rate
  - Reliability



Module 2 – Sensors, Actuators, Wiring

- Sensor characteristics
  - Accuracy, precision, resolution, noise, sampling time
- Sensors
  - IR sensors (amplitude-, angle-, timing-based)
  - Ultrasound
  - IMUs
  - SW filters
  - Odometry



- Sensor fusion
- Actuators (AC, DC, BLDC, Steppers, Servos)
- EMI, wiring, soldering

Lab 3 – ToF and IMU

- Soldering
- ToF
  - Range modes, SNR, integration time, materials
- IMU
  - Acc – gyr - mag
  - Noise and drift, offsets and calibration

Lab 4 – Car characterization

- Learn what the car can do
- Increase comfortability with robot experiments
- When to use your sensors and when to use external sensors

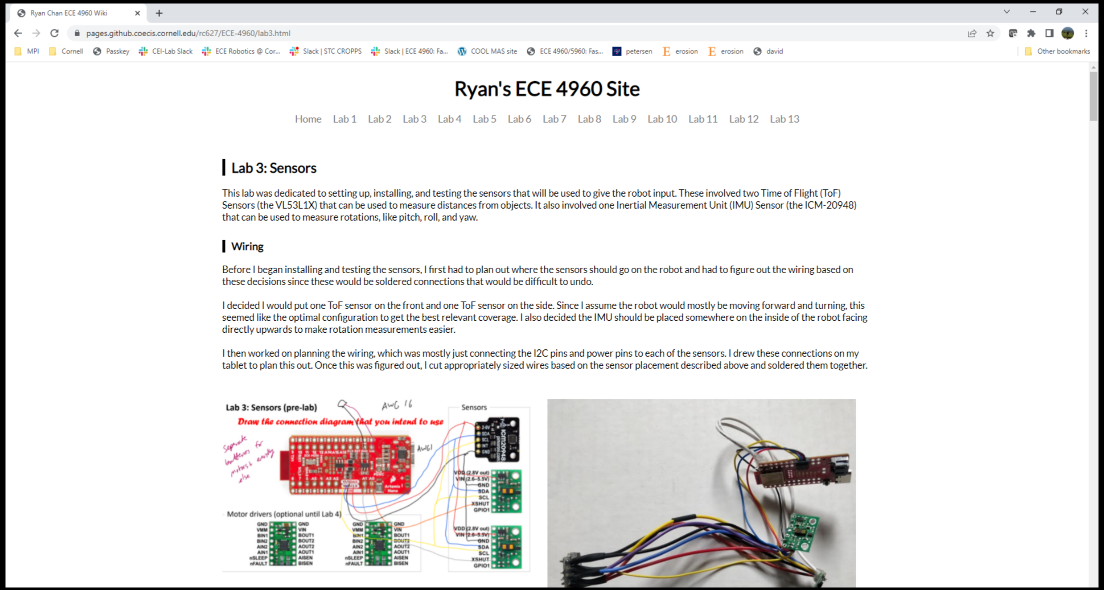
Lab 5 – Open loop control

- Integrate motor drivers
- Stand-alone car

Lab 3 - Sensors

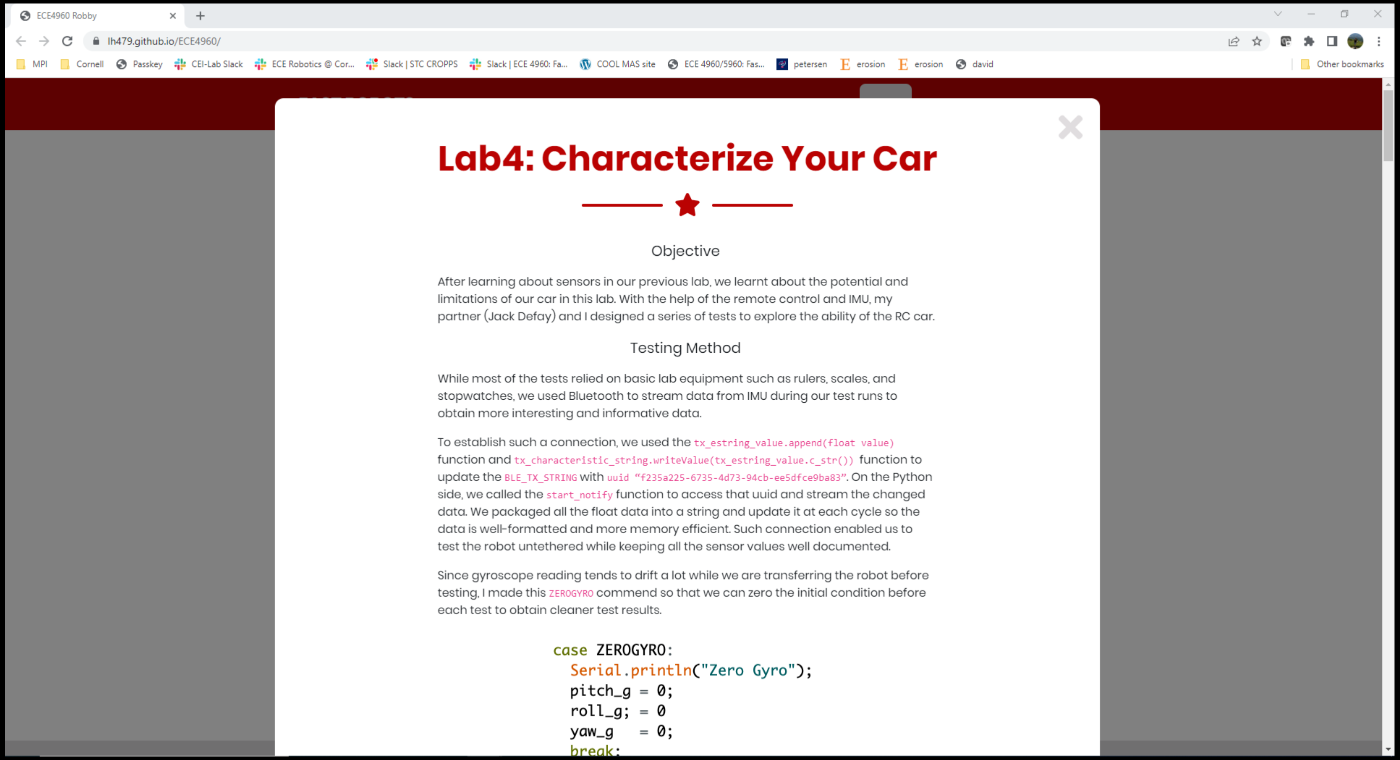
- Ryan Chan
  - Intro
  - Wiring, placement, schematic
  - ToF
    - Checked first ToF sensor
    - Pros and cons of short, medium, long distance modes
    - Average and std. dev. For two materials
    - Second Tof sensor
  - Discussion of other IR-based sensors
  - Timing budget (intermeasurementperiod)
  - Signal and sigma

- IMU
  - Address
  - Initial testing
  - Accelerometer (pitch and roll)
  - Taps, FFT, and low pass filter
    - Noisy
  - Gyroscope
    - Drift
  - Complimentary filter



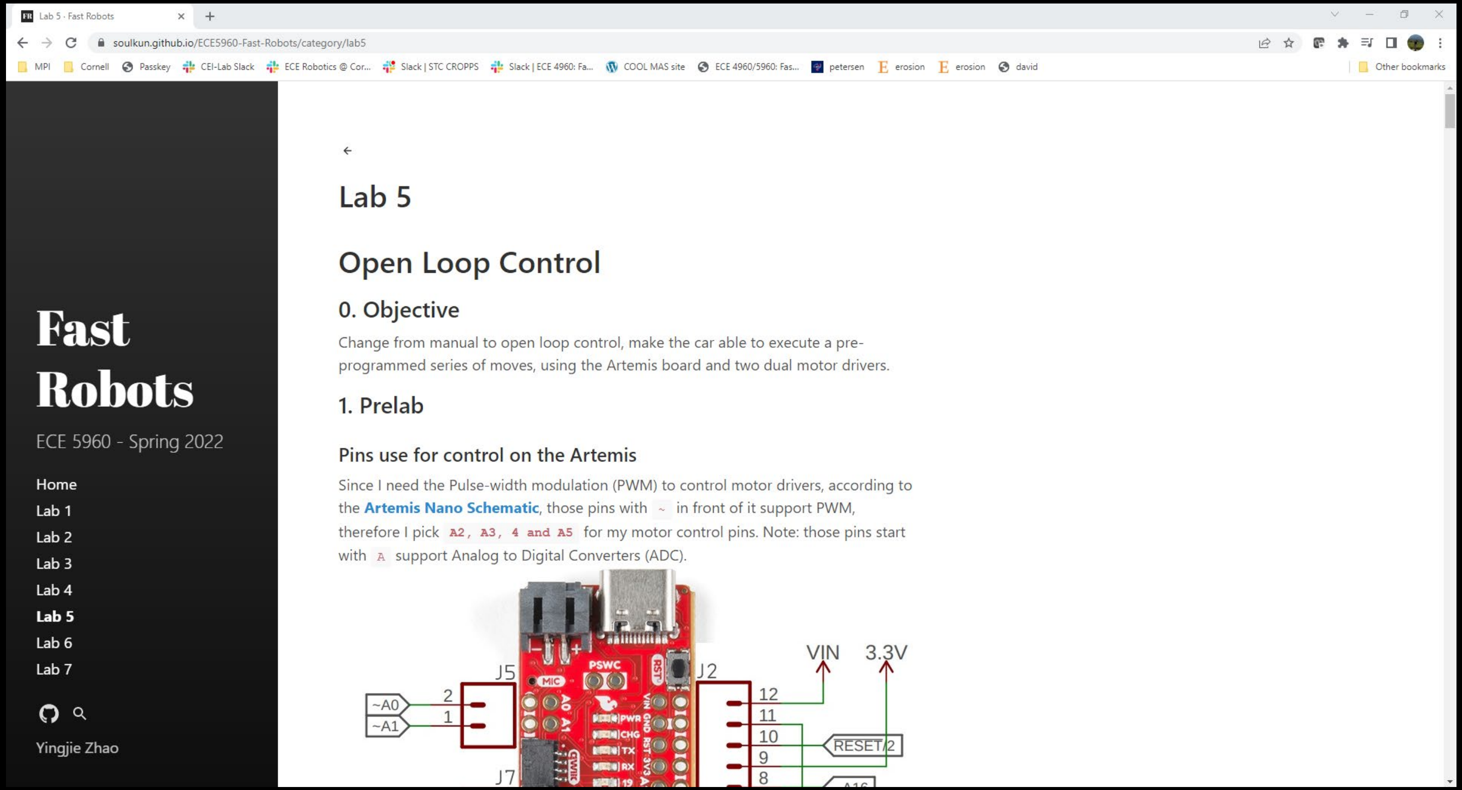
Lab 4 – Car characterization

- Robby Huang (w. Jack Defay)
  - Objective
  - Procedures
  - Dimension and weight
  - Battery discharge curve
  - Battery life time
  - Turn resolution w. IMU
  - Forward flips w. camera
    - Procedure
    - Min. distance to flip
  - Acc. distance for max. speed
  - Stunt
    - Bipedal motion w. gyroscope



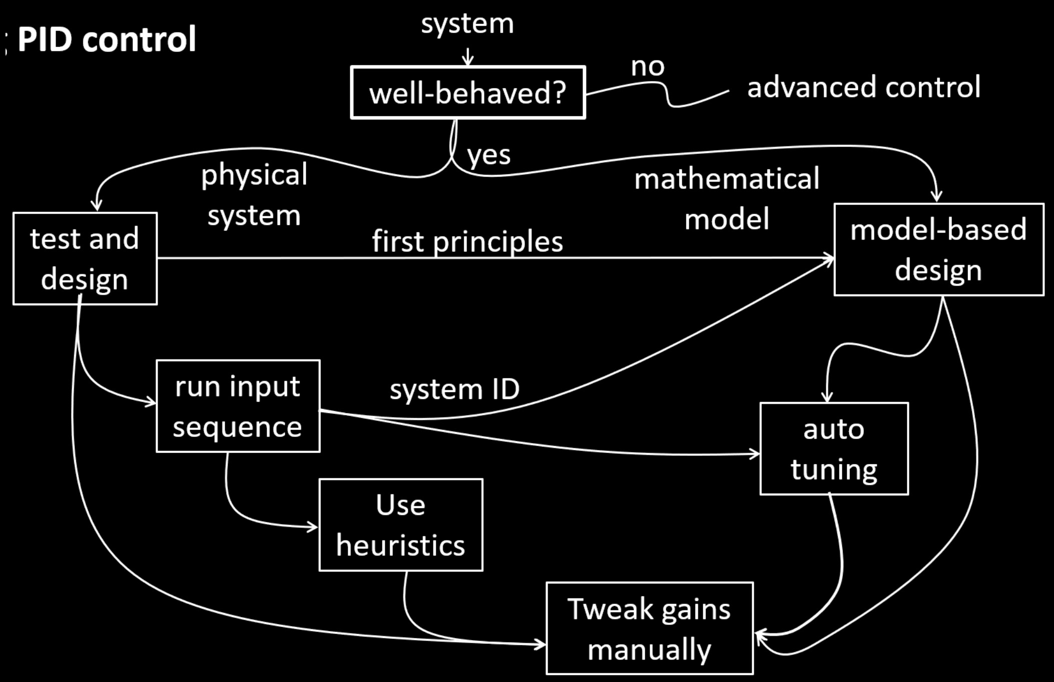
Lab 5 – Open Loop Control

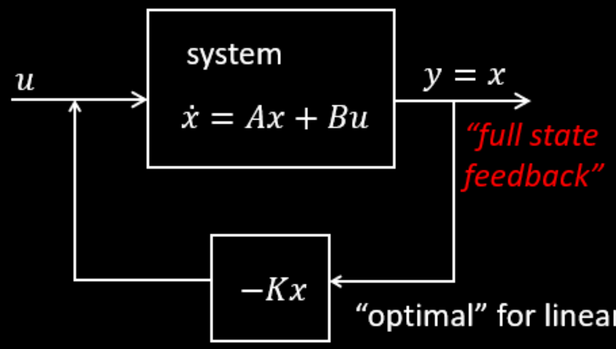
- Yingjie Zhao
  - Objective
  - Control pins
  - Separate batteries
  - Motor drivers
  - Wiring
  - Power motor driver from power supply
  - Two directions
  - Determine the deadband
  - Move in a straight line
  - Frequency of analog write
  - Ramp up/down speed



Module - Control

- PID control → lab 6
- Linear systems
- Linearizing non-linear systems
- Stability
- Controllability
- Reachability
- Optimal control (LQR)
- State space equation examples





"optimal" for linear systems

$$\dot{x} = Ax + Bu$$

$$\dot{x} = Ax - BKx$$

$$\dot{x} = \underbrace{(A - BK)}_{\text{New dynamics}}x$$

$x \in \mathbb{R}^n$

$A \in \mathbb{R}^{n\times m}$

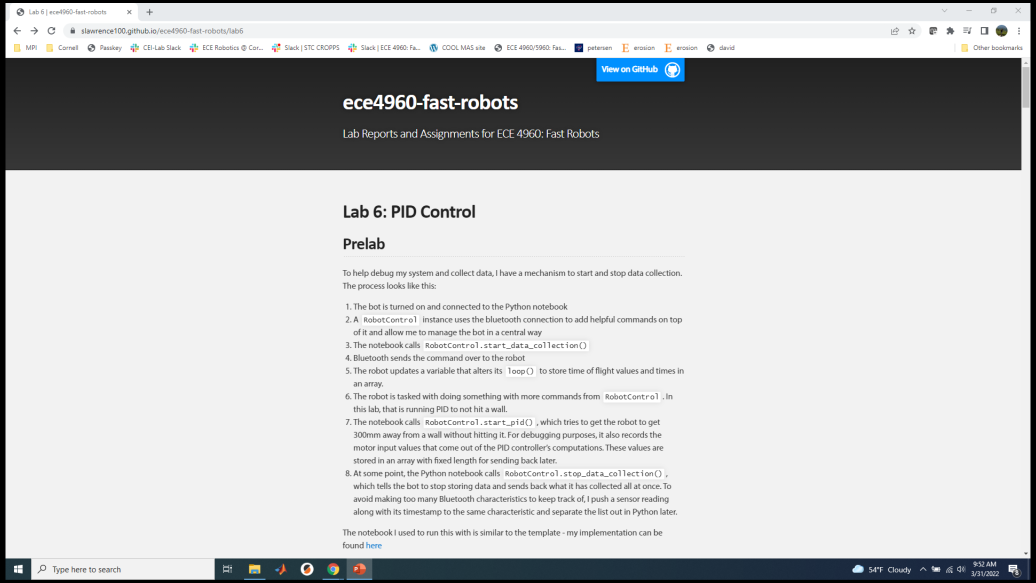
$u \in \mathbb{R}^q$

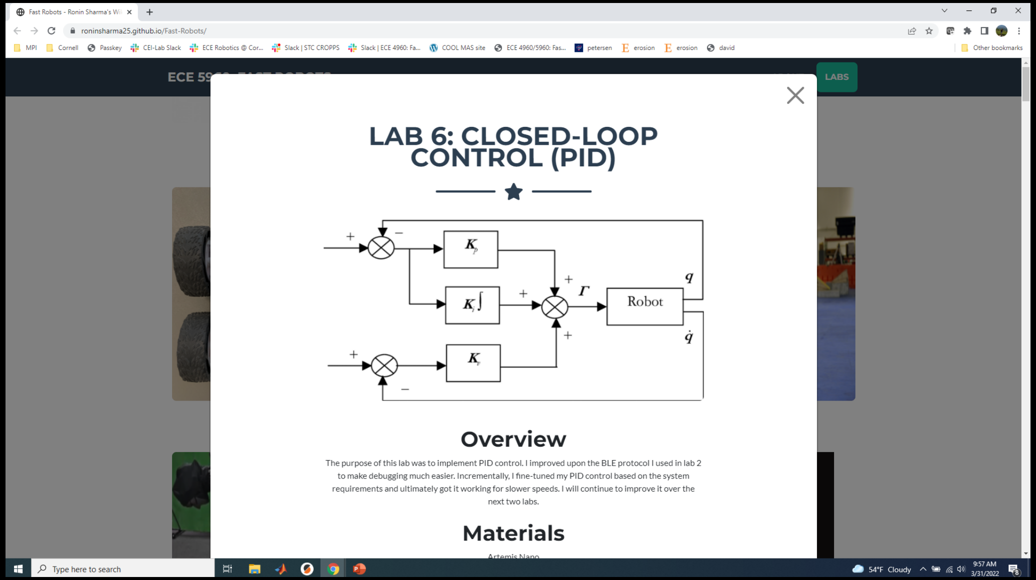
$B \in \mathbb{R}^{n\times q}$

A linear controller (K matrix) can be optimal for linear systems!

Lab 6 – Closed Loop Control (Don't hit the Wall!)

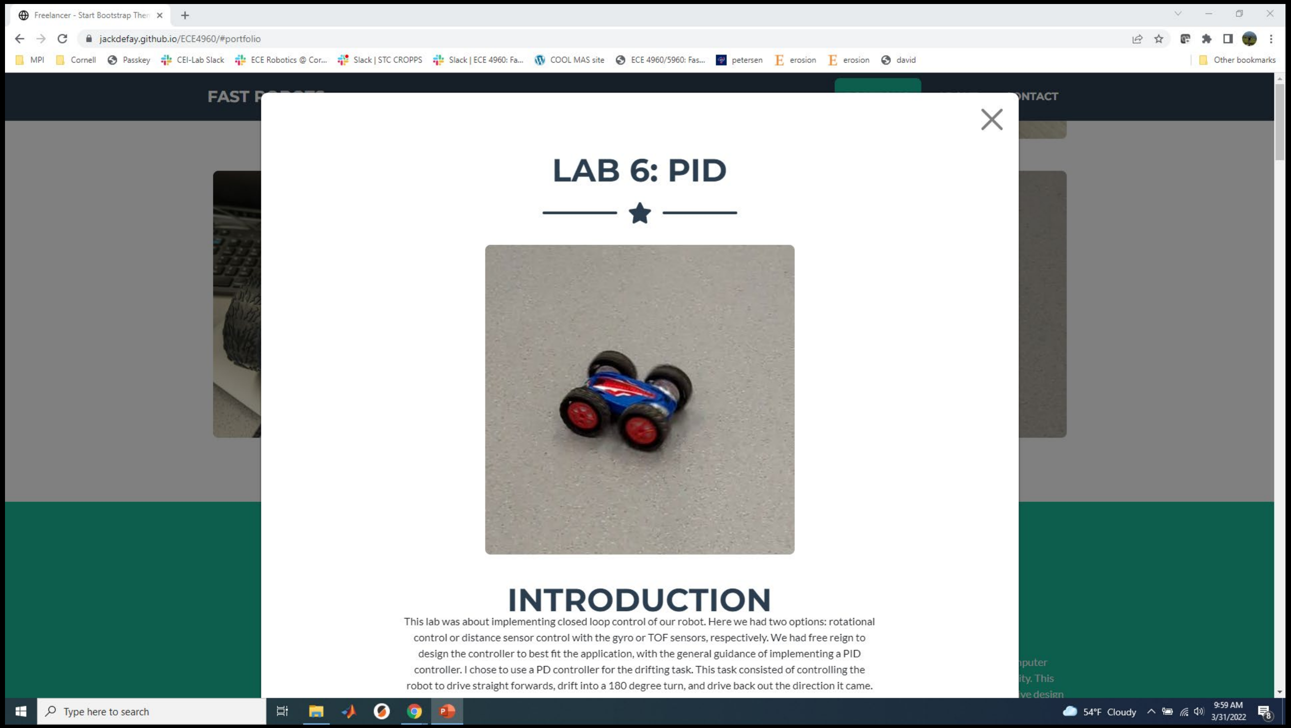
- Sydney Lawrence
  - Python debugging script
  - P-control
  - 4 successful runs
  - Sample time (20Hz, short range mode)
  - Deadband and coast-hack
  - P-controller value
- Ronin Sharma
  - Python debugging script
  - P, PI, PD, PID controller
  - Deadband
  - Sampling frequency
  - ToF offset - hack





Lab 6 – Closed Loop Control (Drift much?)

- Jack Defay
    - PID/PD control
    - Python debugging script
    - LPF on derivative branch
    - Derivative on measurement
    - Integrator wind-up protection
    - Threshold on setpoint
    - Calibration of motor values
    - Deadband
    - Tuning



Module - Observability

- Linear systems
- Observability
- Optimal observers (KF)
  - Probabilistic robots
  - Mean and uncertainty
- State space equation examples

Procedure for Lab 7

- Come up with the state space equations
  - Systems ID (step responses)
- Describe your initial state and uncertainty
- Describe your process noise
- Describe your measurement noise

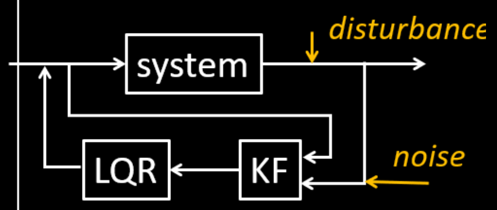
Kalman Filter Implementation

Kalman Filter (μ(t-1), Σ(t-1), u(t), z(t))

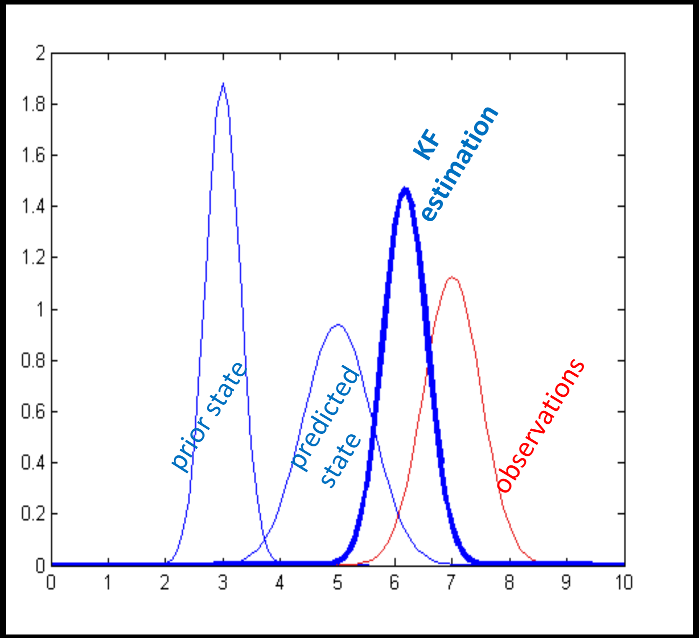
1. $\mu_p(t) = A\,\mu(t-1) + B\,u(t)$
2. $\Sigma_p(t) = A\,\Sigma(t-1)\,A^T + \Sigma_u$
3. $K_{KF} = \Sigma_p(t)\,C^T\,(\,C\,\Sigma_p(t)\,C^T + \Sigma_z)^{-1}$
4. $\mu(t) = \mu_p(t) + K_{KF}\,(\,z(t) - C\,\mu_p(t)\,)$
5. $\Sigma(t) = (\,\mathbf{I} - K_{KF}\,C)\,\Sigma_p(t)$
6. Return μ(t) and Σ(t)

(Steps 1–2: prediction; steps 3–5: update)

State estimate: μ(t)
State uncertainty: Σ(t)
Process noise: Σ_u
Kalman filter gain: K_{KF}
Measurement noise: Σ_z



$$\Sigma_u = \begin{bmatrix} \sigma_1{}^2 & 0 & 0 \\ 0 & \sigma_2{}^2 & 0 \\ 0 & 0 & \sigma_3{}^2 \end{bmatrix}, \Sigma_z = \begin{bmatrix} \sigma_4{}^2 & 0 \\ 0 & \sigma_5{}^2 \end{bmatrix}$$



Module - Observability

- Linear systems
- Observability
- Optimal observers (KF)
  - Probabilistic robots
  - Mean and uncertainty
- State space equation examples

Two common misunderstandings

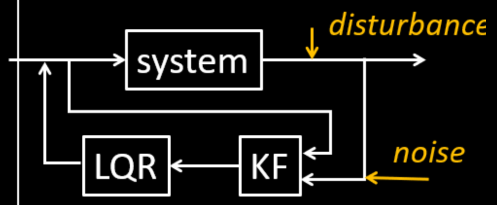
1. *Why is uncertainty necessary?*
2. *How to estimate noise?*

Kalman Filter Implementation

Kalman Filter ($\mu(t-1)$, $\Sigma(t-1)$, $u(t)$, $z(t)$)

1. $\mu_p(t) = A\,\mu(t-1) + B\,u(t)$ } prediction
2. $\Sigma_p(t) = A\,\Sigma(t-1)\,A^T + \Sigma_u$ } prediction
3. $K_{KF} = \Sigma_p(t)\,C^T\,(C\,\Sigma_p(t)\,C^T + \Sigma_z)^{-1}$ } update
4. $\mu(t) = \mu_p(t) + K_{KF}\,(z(t) - C\,\mu_p(t))$ } update
5. $\Sigma(t) = (\mathbf{I} - K_{KF}\,C)\,\Sigma_p(t)$ } update
6. Return $\mu(t)$ and $\Sigma(t)$

State estimate: $\mu(t)$
State uncertainty: $\Sigma(t)$
Process noise: Σ_u
Kalman filter gain: K_{KF}
Measurement noise: Σ_z



$$\Sigma_u = \begin{bmatrix} \sigma_1{}^2 & 0 & 0 \\ 0 & \sigma_2{}^2 & 0 \\ 0 & 0 & \sigma_3{}^2 \end{bmatrix}, \Sigma_z = \begin{bmatrix} \sigma_4{}^2 & 0 \\ 0 & \sigma_5{}^2 \end{bmatrix}$$

- Process noise:
  - Trust in modeled position:
    - $\sigma_1 = \sqrt{10^2 \cdot \frac{1}{0.13}} = 27.7mm$
  - Trust in modeled speed:
    - $\sigma_2 = \sqrt{10^2 \cdot \frac{1}{0.13}} = 27.7mm/s$
- Measurement noise
  - $\sigma_3^2 = (20mm)^2$

Module - Observability

- Linear systems
- Observability
- Optimal observers (KF)
  - Probabilistic robots
  - Mean and uncertainty
- State space equation examples

Lab 7 – Kalman Filter

- Understand the concept
- Try embedded implementation

Lab 8 – Stunt

- Use the KF to increase sampling rate
  - Open loop, repeatable stunts

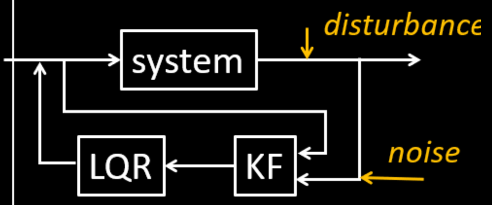
Kalman Filter Implementation

Kalman Filter (μ(t-1), Σ(t-1), u(t), z(t))

1. $\mu_p(t) = A\,\mu(t-1) + B\,u(t)$
2. $\Sigma_p(t) = A\,\Sigma(t-1)\,A^T + \Sigma_u$
3. $K_{KF} = \Sigma_p(t)\,C^T\,(C\,\Sigma_p(t)\,C^T + \Sigma_z)^{-1}$
4. $\mu(t) = \mu_p(t) + K_{KF}\,(z(t) - C\,\mu_p(t))$
5. $\Sigma(t) = (\mathbf{I} - K_{KF}\,C)\,\Sigma_p(t)$
6. Return $\mu(t)$ and $\Sigma(t)$

(Steps 1–2: prediction; steps 3–5: update)

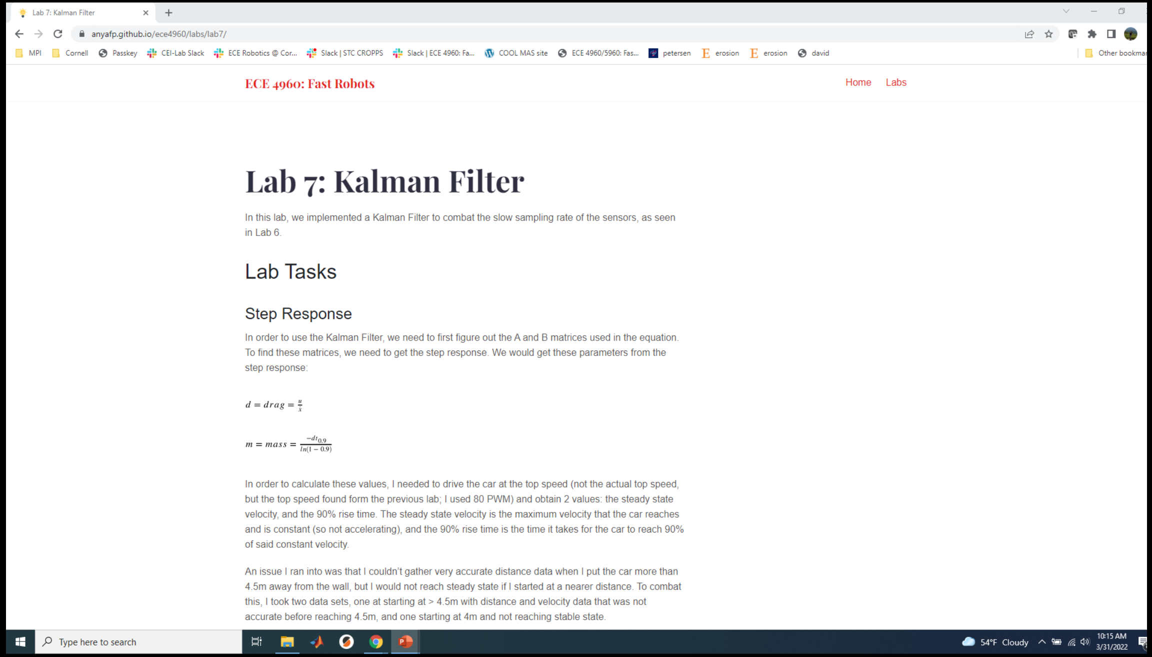
State estimate: $\mu(t)$
State uncertainty: $\Sigma(t)$
Process noise: Σ_u
Kalman filter gain: K_{KF}
Measurement noise: Σ_z



$$\Sigma_u = \begin{bmatrix} \sigma_1{}^2 & 0 & 0 \\ 0 & \sigma_2{}^2 & 0 \\ 0 & 0 & \sigma_3{}^2 \end{bmatrix}, \Sigma_z = \begin{bmatrix} \sigma_4{}^2 & 0 \\ 0 & \sigma_5{}^2 \end{bmatrix}$$

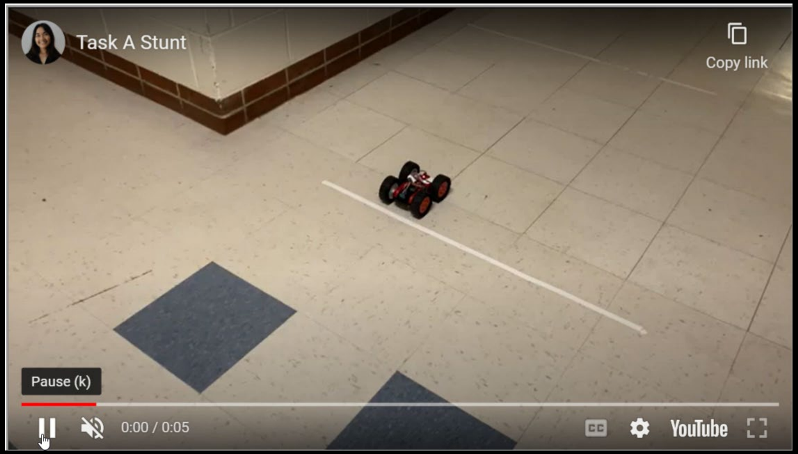
Lab 7 – Kalman Filters

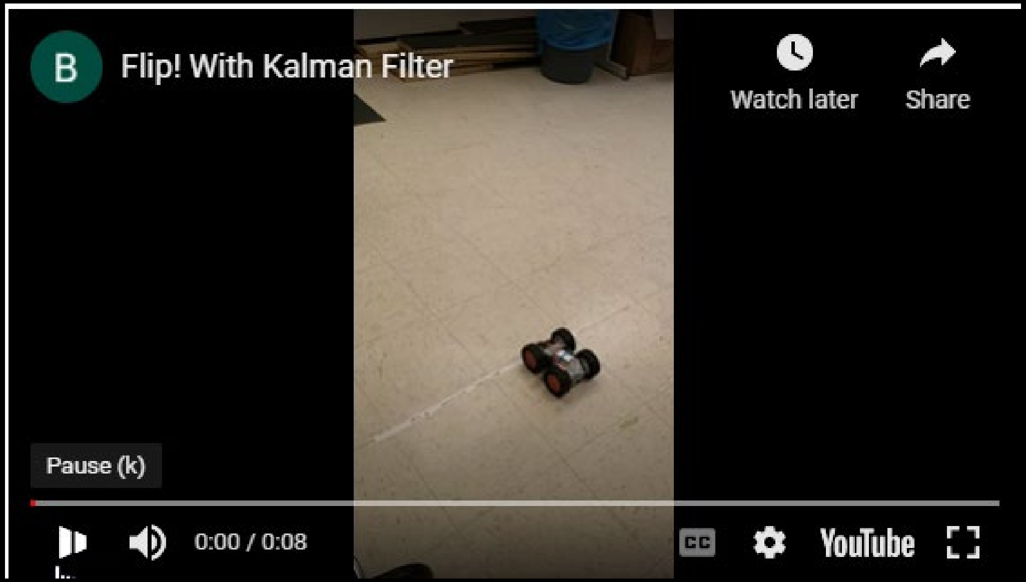
- Anya Prabowo
  - Step response to determine d and m
  - Determine A and B matrices
  - Estimate process and measurement noise
  - Determine C
  - Sanity check on previous data
    - More reliance on measurement vs dynamics
  - Implementation on real robot
    - Adjusted noise



Lab 8 - Stunts

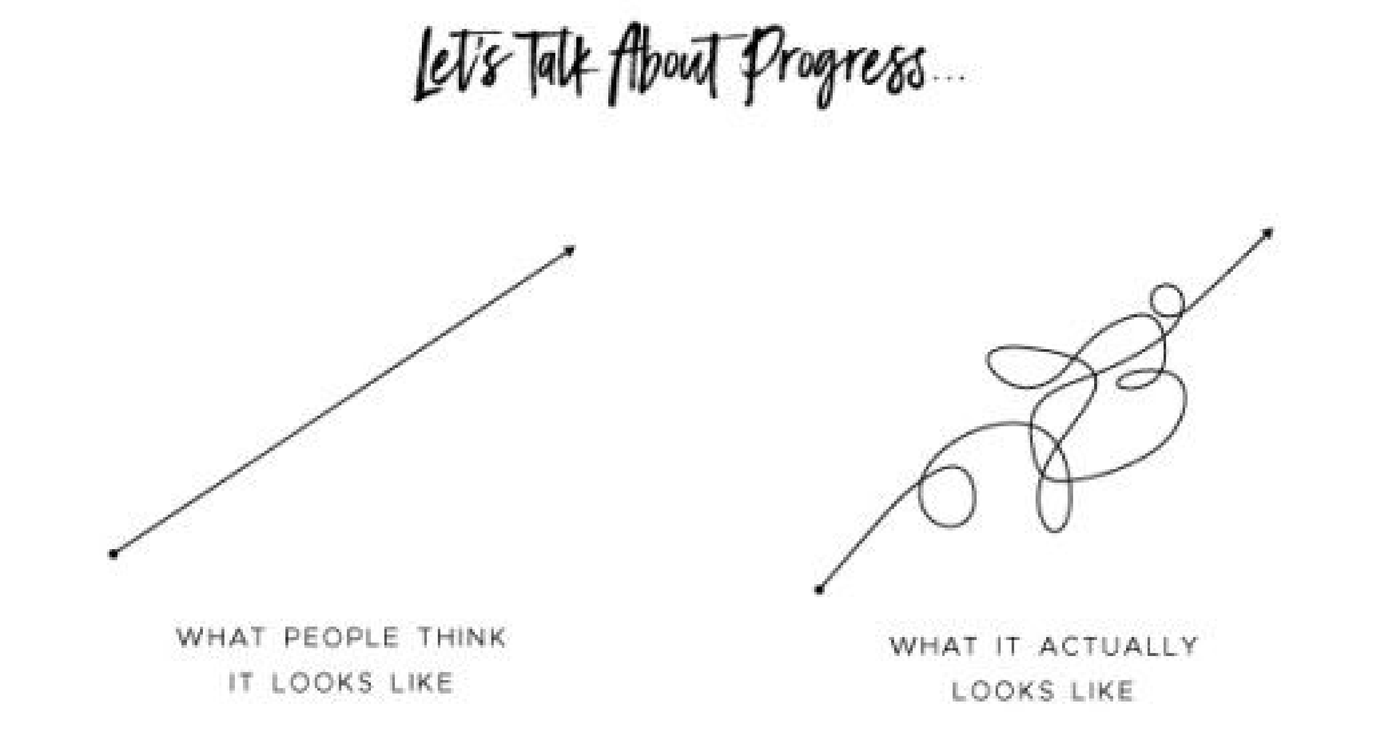
- Anya Prabowo
  - Battery charge
  - Dustiness
  - Motor calibration
  - “KF hack”
- Sydney Lawrence
  - “KF hack”
  - Motor calibration
- Ben Wagner
  - Modify A/B matrices
  - Tune run length, speed, reverse speed
  - ToF offset





Free lessons

- *Hardware is HARD*
  - …and progress is not linear
  - (we know!)
- Debugging robots
  - What was the last thing you changed?
  - Debugging scripts
  - Is the problem in Python, C, electronics, or mechanics?
  - Unit test
- You've done GOOD!
  - ASML presents…



Rest of Class

- Lab 9 – Mapping
- Lab 10 – Simulator (flipped classroom)
- Lab 11 – Localization (sim)
- Lab 12 – Localization (real)
- Lab 13 – Navigation (real)
  - Two+ weeks

- Loop-holes?
  - Common map
  - 1 ToF sensor
  - Lab 11 solution is given in the following week
  - Work together!!

Course Midway Feedback

- 16 students (8 ugrad/8 M.Eng.)
  - Pretty positive (thank you!!)
  - Clarify what to do with material
  - Expand on how the weekly lab is supposed to be done
  - More small group discussions
  - Provide more examples
  - Support Windows 11….
  - 6/16 felt lectures are progressing too fast
  - 10/16 preferred more open-ended labs
  - Workload less sporadic
  - **Shorter assignments**

Ideas for improvement?

- What would you cut/edit?...
- Our ideas…
  - Lab 1-2 Artemis and Bluetooth
    - Add arrays/debugging scripts
  - Lab 3-4 - Sensors and car characterization
    - 7 new soldering stations
    - Split lab 3 into two weeks, and skip lab 4
    - Just one sensor?
    - Avoid some soldering
  - Lab 5 – Open loop
  - Lab 6 - PID
  - Lab 7 – Kalman Filters
    - Could give you the code and just have you tune the KF
  - Lab 8 - Stunts

Also prelims…

- Please consider teaming up…
  - Strengths / weaknesses
  - Time commitment
  - Weekly availability