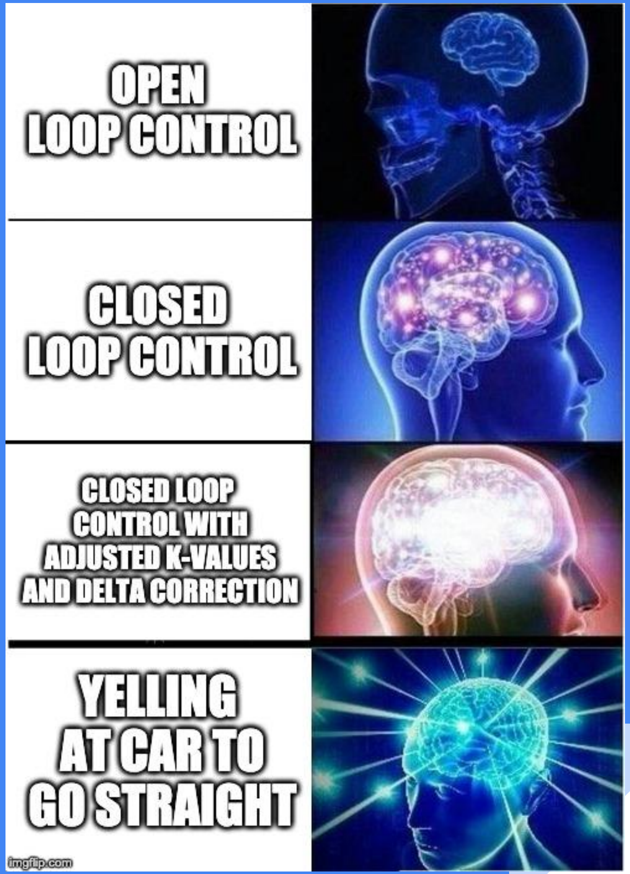
# Lab 7: Controls Part 1

EECS 16B Fall 2022

Slides: http://links.eecs16b.org/lab7-slides



# Lab 7 Overview

- Make your car drive straight!
- Open-Loop Control
    - Open loop simulation with/without model mismatch
    - Jolt value calculations
- Closed-Loop Control
    - Simulation + feedback gain f-value tuning
    - Steady-state error correction
- Next week: Turning!

# System ID → Open Loop Control

$$v_L[i] = d_L[i+1] - d_L[i] = \theta_L u_L[i] - \beta_L$$
$$v_R[i] = d_R[i+1] - d_R[i] = \theta_R u_R[i] - \beta_R$$

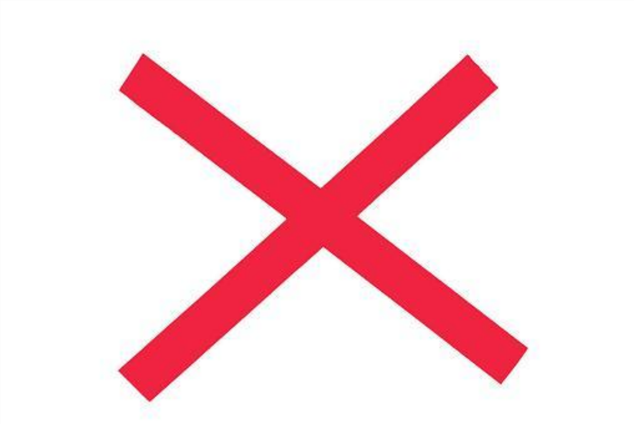
- Last week, we:
  - measured v, we knew u
  - We found $\theta_{L,R}$ and $\beta_{L,R}$ from least squares
  - Determined operating velocity point v*
- Given some target velocity v, what input u do we need?
  - Open Loop Control: solve the above equations for u

# Problems with Open Loop

Does open loop work well for systems with disturbances? Why or why not?

# Problems with Open Loop

- Will not correct for disturbance/noise (marginally stable)
- Assumes θ, β are the actual θ, β of the wheels
  - Any error will build up, preventing the car from going straight



# Closed Loop Intuition

- Introduce an error term that indicates the car's trajectory
  - Negative feedback allows us to correct for disturbance
- Assuming perfect model, will cause delta to converge to 0
- With model mismatch, delta converges to non-zero value
  - Steady-state error

# Closed Loop Equations

- Introduce an error term:

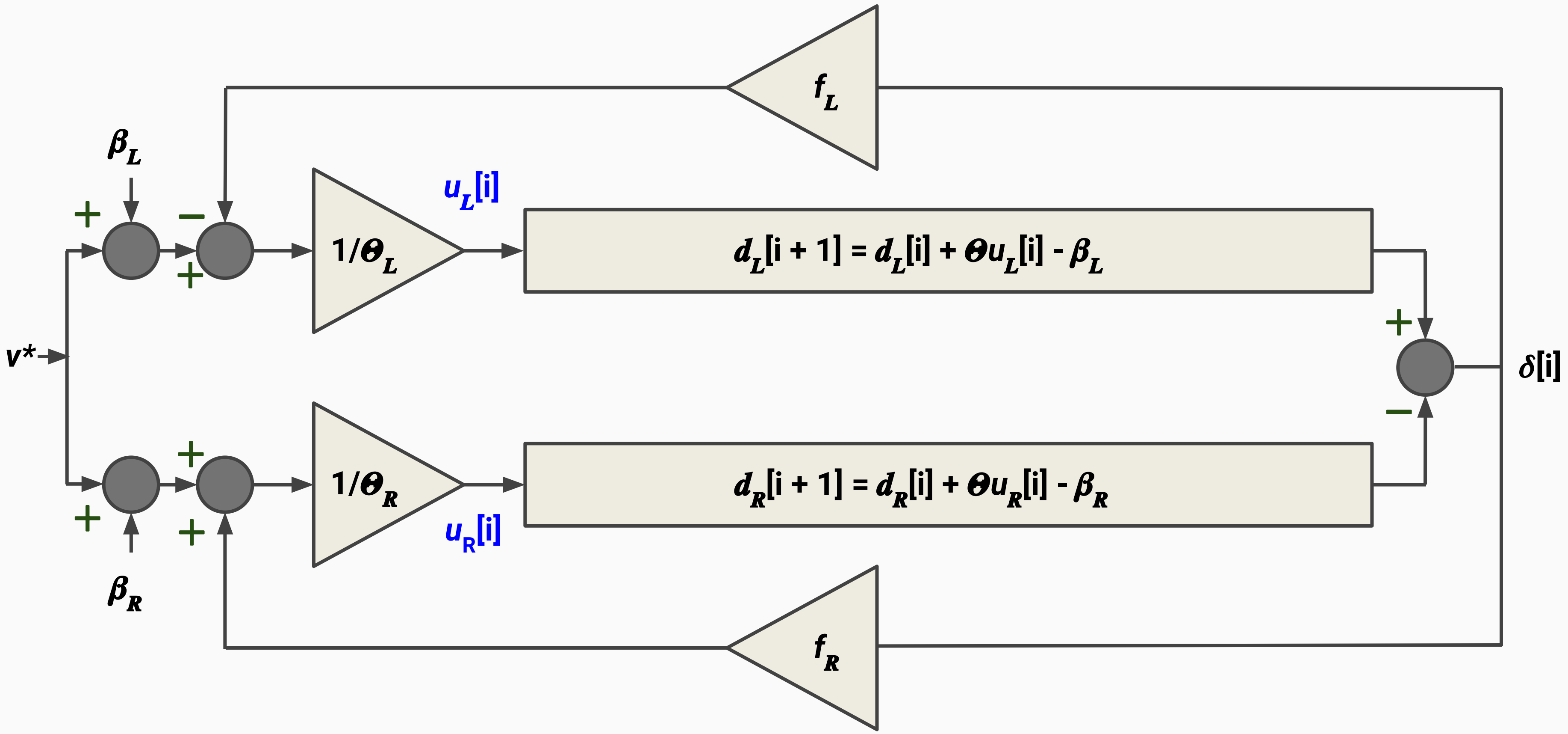
$$\delta[i] = d_L[i] - d_R[i]$$

- The wheel/motor models become

$$d_L[i+1] = d_L[i] + \theta_L u_L[i] - \beta_L - f_L \delta[i]$$
$$d_R[i+1] = d_R[i] + \theta_R u_R[i] - \beta_R + f_R \delta[i]$$

- Convention: $f > 0$

# Closed Loop Visualization for finding u



# Closed Loop Analysis

- What's the error after one step?
  - $\delta[i+1] = d_L[i+1] - d_R[i+1]$
  - Plug in for d[i+1] (remember v[i] = d[i + 1] - d[i])
  - $\delta[i+1] = v^* - f_L\delta[i] + d_L[i] - (v^* + f_R\delta[i] + d_R[i])$
  - Simplify
  - $\delta[i+1] = \delta[i]\,(1 - f_L - f_R)$
- Stability Analysis:
  - For discrete-time systems, we want $\lambda \in (-1, 1)$.
  - If $\lambda < 0$, if f-values are too large, we will see oscillatory behavior (over-correction)

# Tips and Tricks

- Don’t guess f-values, this will take you forever!
  - Make educated decisions on how to change your f values from iterations of testing.
  - If you car is turning left, how should you change $f_L$ and $f_R$ to fix it?
- Data is stored in RAM, just like last lab, so make sure you keep the 9V plugged in when you plug the USB into your computer

# Common Bugs

- Double check all closed-loop equations
  - Minus sign before $f_L$
  - Plus sign before $f_R$
- If a wheel jolts and stops moving:
  1. Double check that all pins (motor and encoder) you are using are correctly defined in the Arduino code
  2. Rerun encoder tests from System ID to make sure encoders are still working
- If motors are no longer running, rerun the encoder test from last lab

# Lab 7 Checkoff

- Our definition of “straight” is based on the floor tiling in Cory:
  - Inside Cory 125 (1x4 tiles)
  - Outside Cory 125 (3 x 11 on black)
  - Side entrance hallway, from the pink line to the red line (2 x 7 tiles)

# Important Forms/Links

- Help request form: https://eecs16b.org/lab-help
- Checkoff request form: https://eecs16b.org/lab-checkoff
- Extension Requests: https://eecs16b.org/extensions
- Makeup Lab: https://makeup.eecs16b.org
- Slides: links.eecs16b.org/lab7-slides
- Anon Feedback: https://eecs16b.org/lab-anon-feedback
- Lab Grades error: https://links.eecs16b.org/lab-checkoff-error