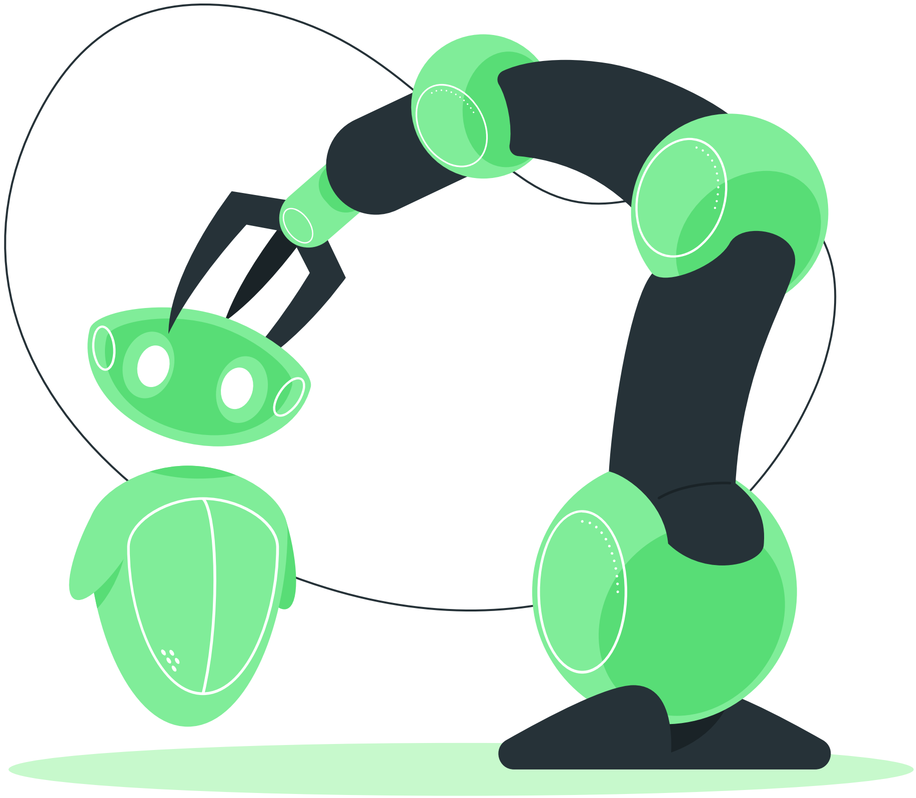
EECS C106B Week 3 Lab

Project 1B Intro and Lab 0



Agenda

1

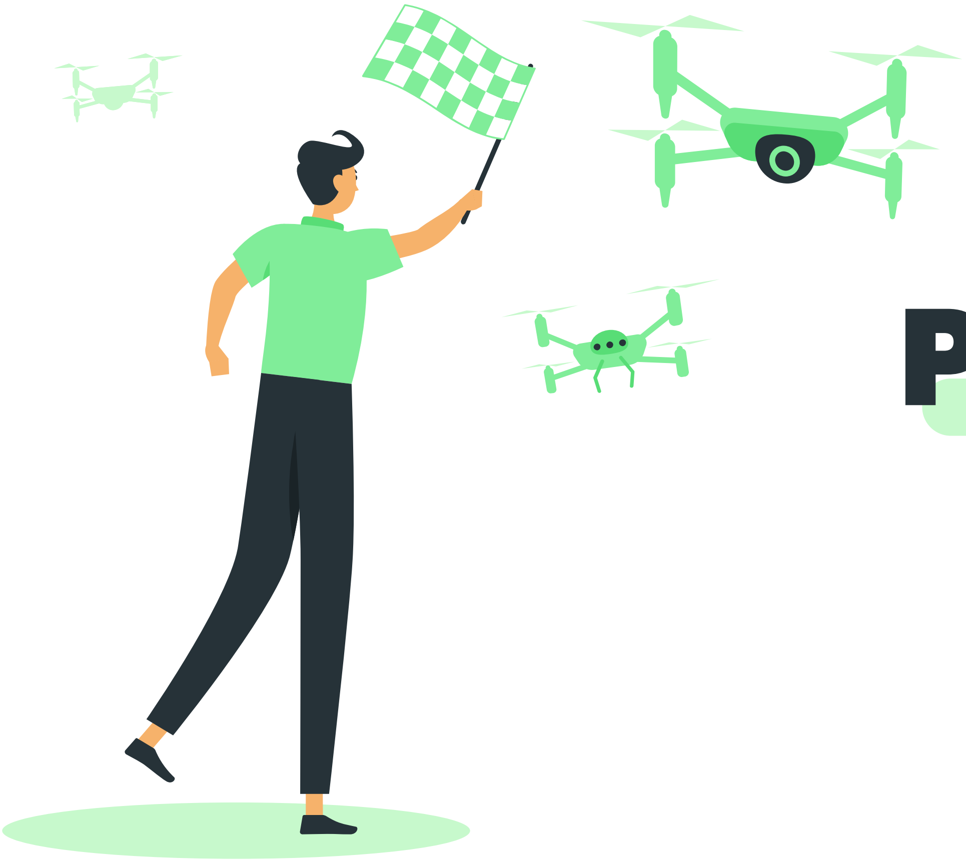
Project 1B Intro

Some relevant info for Project 1B

2

Project 0

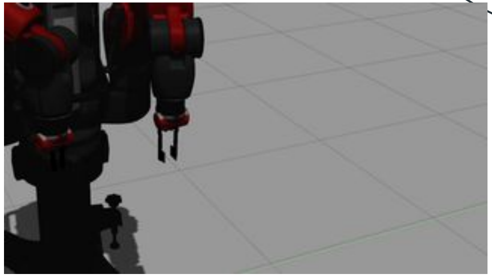
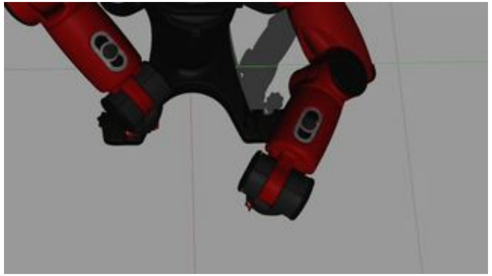
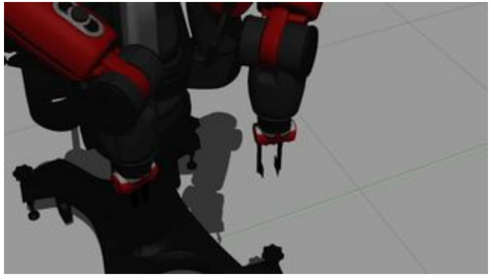
Get (re)acquainted with the robots!



Project 1B

Goals

- Should already have trajectories defined (line, circle, arbitrary) in terms of SE(3) pose and se(3) body velocity at every point in time
- Implement 3 closed-loop controllers
    - **Jointspace Velocity**
    - **Workspace Velocity**
    - **Jointspace Torque**
- Compare trajectory tracking performance amongst your own controllers and MoveIt!
- Present findings in conference paper format

Controllers

1

Jointspace Velocity

Given: desired joint positions, velocities, and accelerations
Produce: control input as joint velocities

2

Jointspace Torque

Given: desired joint positions, velocities, and accelerations
Produce: control input as joint torques

3

Workspace

Given: desired workspace positions, velocities, and accelerations
Produce: control input as joint velocities

Jointspace Velocity Controller

- **$u(t)$**: control input
- **$u_{ff}(t)$** and **$u_{fb}(t)$**: feedforward and feedback terms
- **$\theta(t)$** and **$\theta_d(t)$**: actual and desired joint positions
- **K_p** and **K_v**: proportional and derivative terms for PD controller
- **$e(t)$**: error in joint positions

$$u(t) = u_{ff}(t) + u_{fb}(t)$$

$$u_{ff}(t) = \dot{\theta}_d(t)$$

$$u_{fb}(t) = K_p e(t) + K_v \dot{e}(t)$$

$$e(t) = \theta_d(t) - \theta(t)$$

Jointspace Torque Controller

- **M(θ)**: inertia matrix
- **C(θ)**: coriolis matrix
- **G(θ)**: gravity vector

$$u(t) = u_{ff}(t) + u_{fb}(t)$$

$$e(t) = \theta_d(t) - \theta(t)$$

$$u_{ff}(t) = M(\theta)\ddot{\theta}_d(t) + C(\theta, \dot{\theta})\dot{\theta}_d(t) + G(\theta)$$

Augmented PD Control Law (MLS p. 194)

$$u_{fb}(t) = K_p e(t) + K_v \dot{e}(t)$$

Computed Torque Control Law (MLS p. 191)

$$u_{fb}(t) = M(\theta)(K_p e(t) + K_v \dot{e}(t))$$

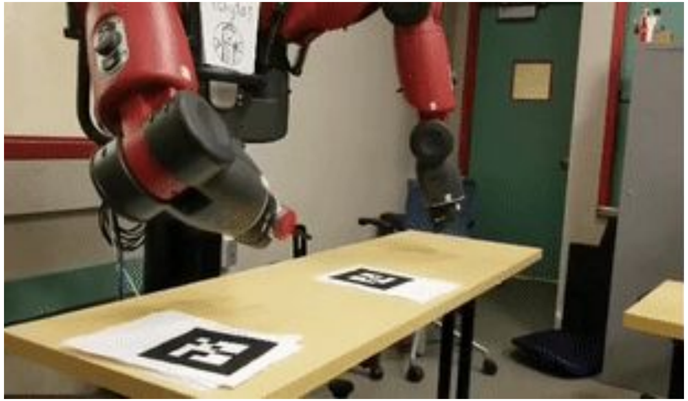
Workspace Controller

- $\mathbf{g_t}$ and $\mathbf{g_d}$: current and desired robot frames
- $\mathbf{K_p}$: 6x6 positive diagonal matrix of proportional controller gains
- $\boldsymbol{\xi_{td}}$: velocity which reduces error between frames
- $\mathbf{V_d^b}$: velocity in desired frame
- $\mathbf{Ad_{gtd}}$: Adjoint of g_{td} frame
- $\mathbf{(J^s_{st}(\theta))^\dagger}$: spatial Jacobian pseudo-inverse

USE PROJECT DOC FOR UPDATED FORMULAS

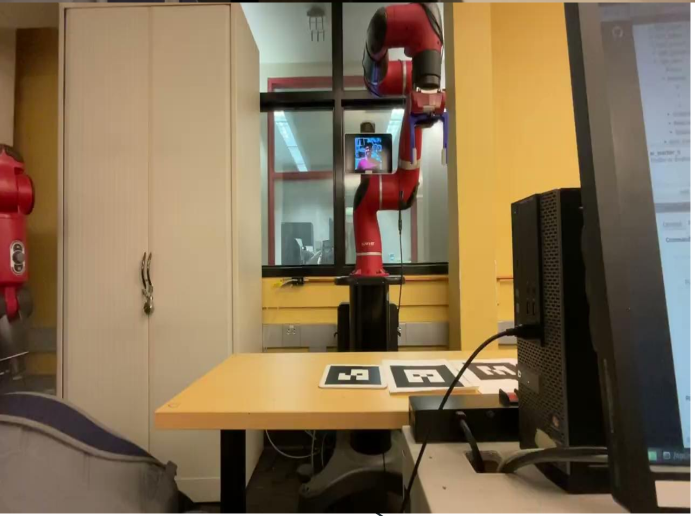
Advice

- **Don't expect things to work perfectly**
    - Explain why in your report
- Implementation difficulty (probably)
    - Jointspace Velocity < Jointspace Torque < Workspace Velocity
    - Tackle an easier controller first to understand the codebase and get something working
- Test open-loop controllers before adding feedback terms
- Remember to tune the proportional term first, then the derivative term



Line

Circle



Polygon

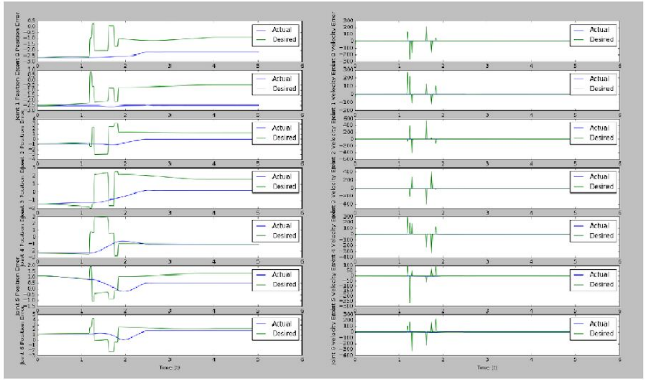
Visual Servo

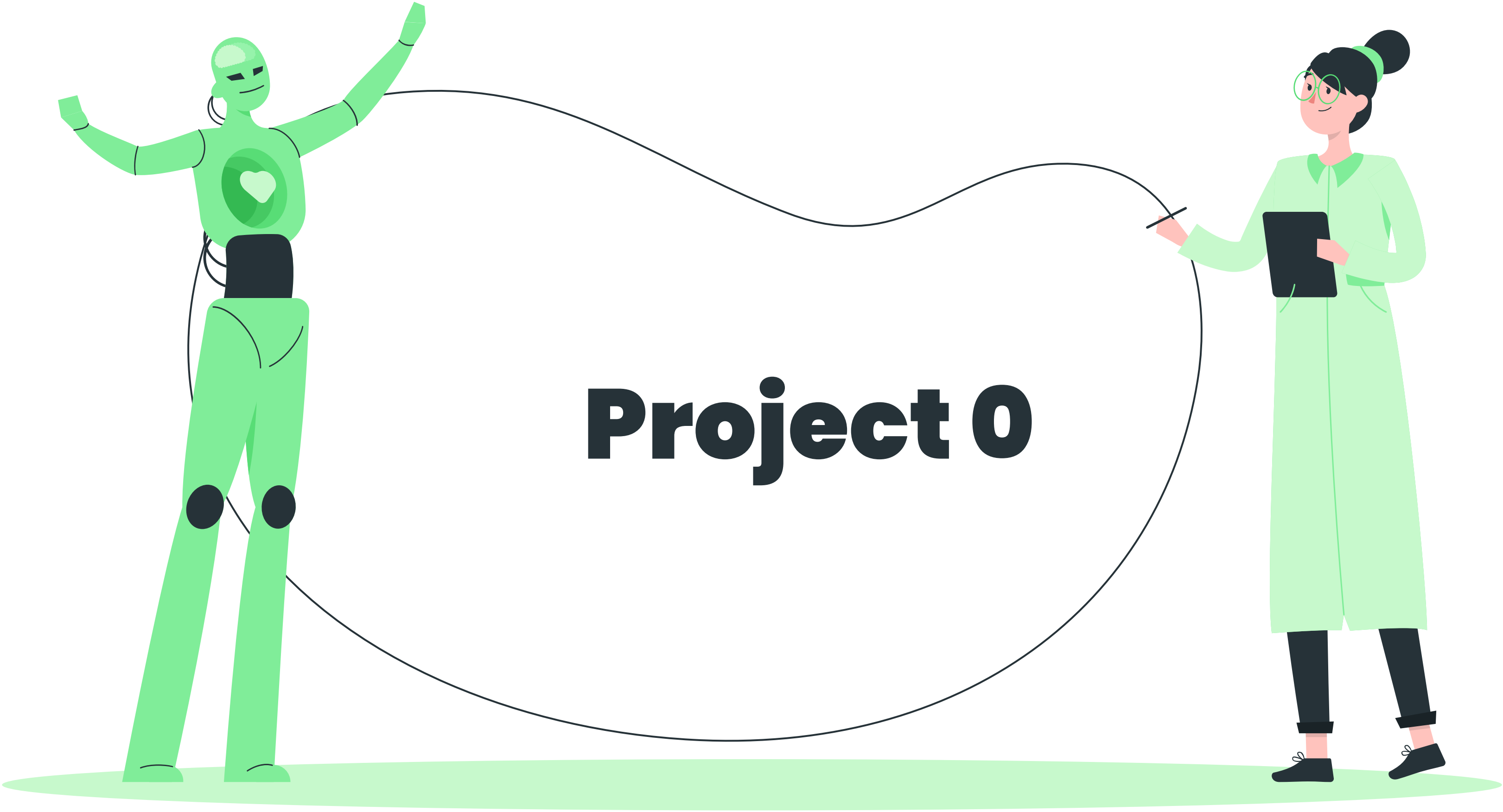
Levity

>source devel/setup.bash

rosrun main.py -task circle -controller_name jointspace -arm left -rate 200 -timeout None -num_way 4000 --moveit --log -literaly every -other command -line argument -in existance

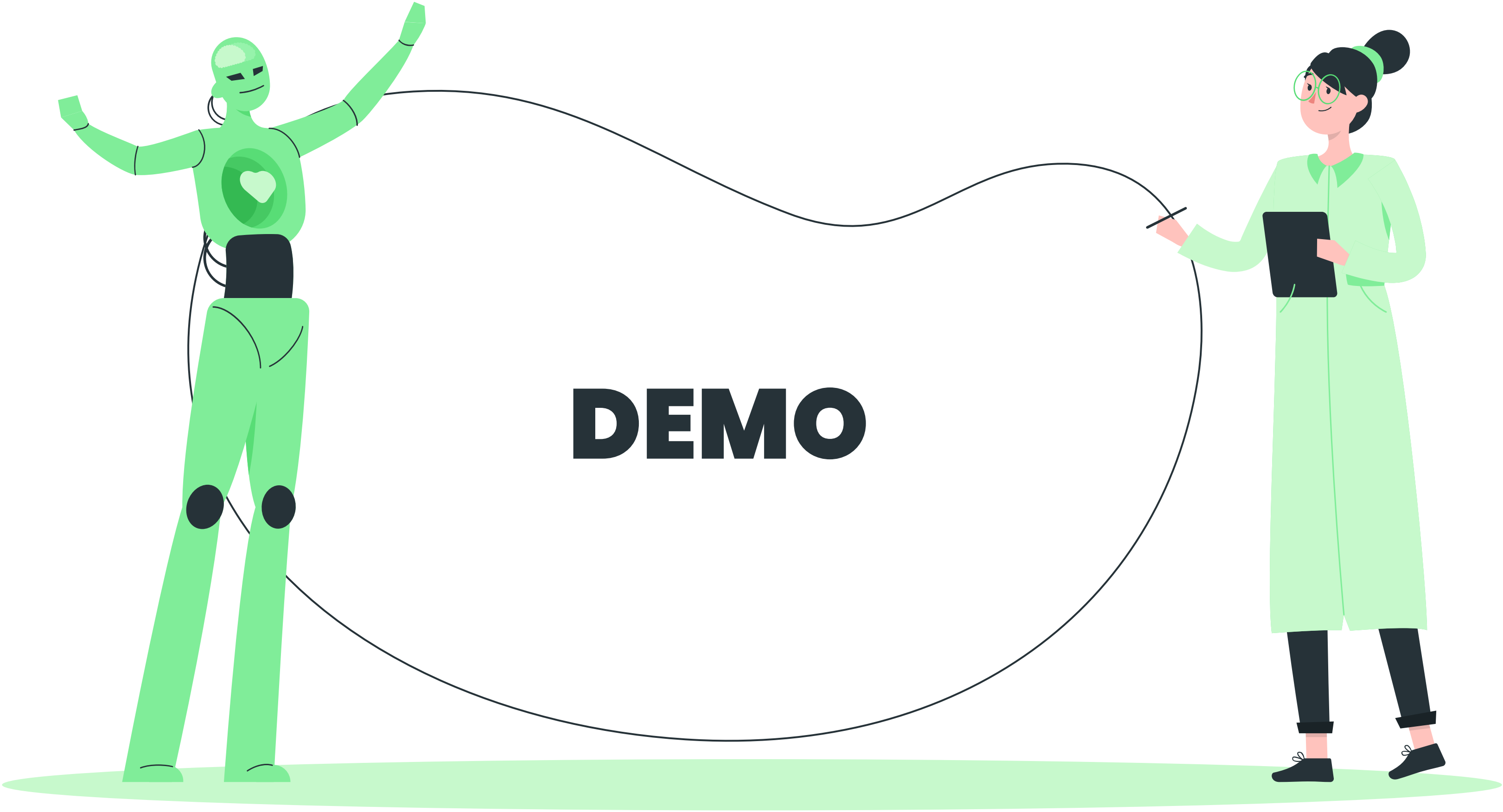
discontinuities in joint trajectory




Project 0

Project 0

- Learn how to use lab robots safely - Sawyer, Turtlebots
- Work in pairs to complete Project 0, does not have to be a project group member
- Make sure you know how to do concept checkpoints!


DEMO

Timeline of the Near Future

Sign Up for Papers

First paper presentation is next week. Skim the first paper before your next lab section

HW2

Due 2/7

Complete Robot Usage Quiz!

Project 1B

Due 2/14

Urgent To Dos!

Proj 1B Team Registration

https://docs.google.com/forms/d/e/1FAIpQLSflnxcwP4Bxc73h-jBlbQ1N-hy4L91Gf1KjwAhYKw1qs6_W5g/viewform

Allows us to assign robots for Project 1B

Optional Project 1A Peer Eval

https://forms.gle/zeSaR6VzwuunMmgy9

Robot Reservations!

Link