Final Week Goals

* Merge complete 2R Robot to main GitHub repository
* Implement 2R robot using Crocoddyl library
* Create presentation for the work you have done

Crocoddyl

Pros:

Do not need to implement the mathematics behind the optimization solver

Lack of knowledge behind DDP and Dynamics

Have accessible notebooks showing how to run simulations

Good for sharing the results on an experiment

The one thing that is needed is a **model of the robot**

.urdf = unified robot description file = XML file containing description of robot

[Quickstart](http://wiki.ros.org/urdf/Tutorials)

For our 2R Robot

Three Links – 0, 1, 2

Inertial Matrix (mass) to simulate dynamics

Two Joints – 0, 1

Revolute Joints

[Double Pendulum](https://github.com/loco-3d/crocoddyl/blob/master/examples/double_pendulum.py) analogous model (Steps)

1. Load model
2. Add actuators (One actuator at link 0, one at link 1)
3. Add costs
4. Define forward kinematics using functions
5. Define a solver

Recommendation: Start by reading [introduction to Crocoddyl](https://github.com/loco-3d/crocoddyl/blob/master/examples/notebooks/introduction_to_crocoddyl.ipynb) notebook, and then examine double pendulum

Work on Google Colab as cannot install Crocoddyl library on windows:

Noe will do the integration given clean scripts

Step 0: Plan a meeting on Thursday to work through problems you are running into

Step 1: Get the double pendulum problem setup in colab

* Out of scope to install library locally
* Create a new branch for crocoddyl
  + Link that to Github

Kinematics = Study of motion without analysis of its cause

Dynamics = Study of forces acting on an object that cause its motion