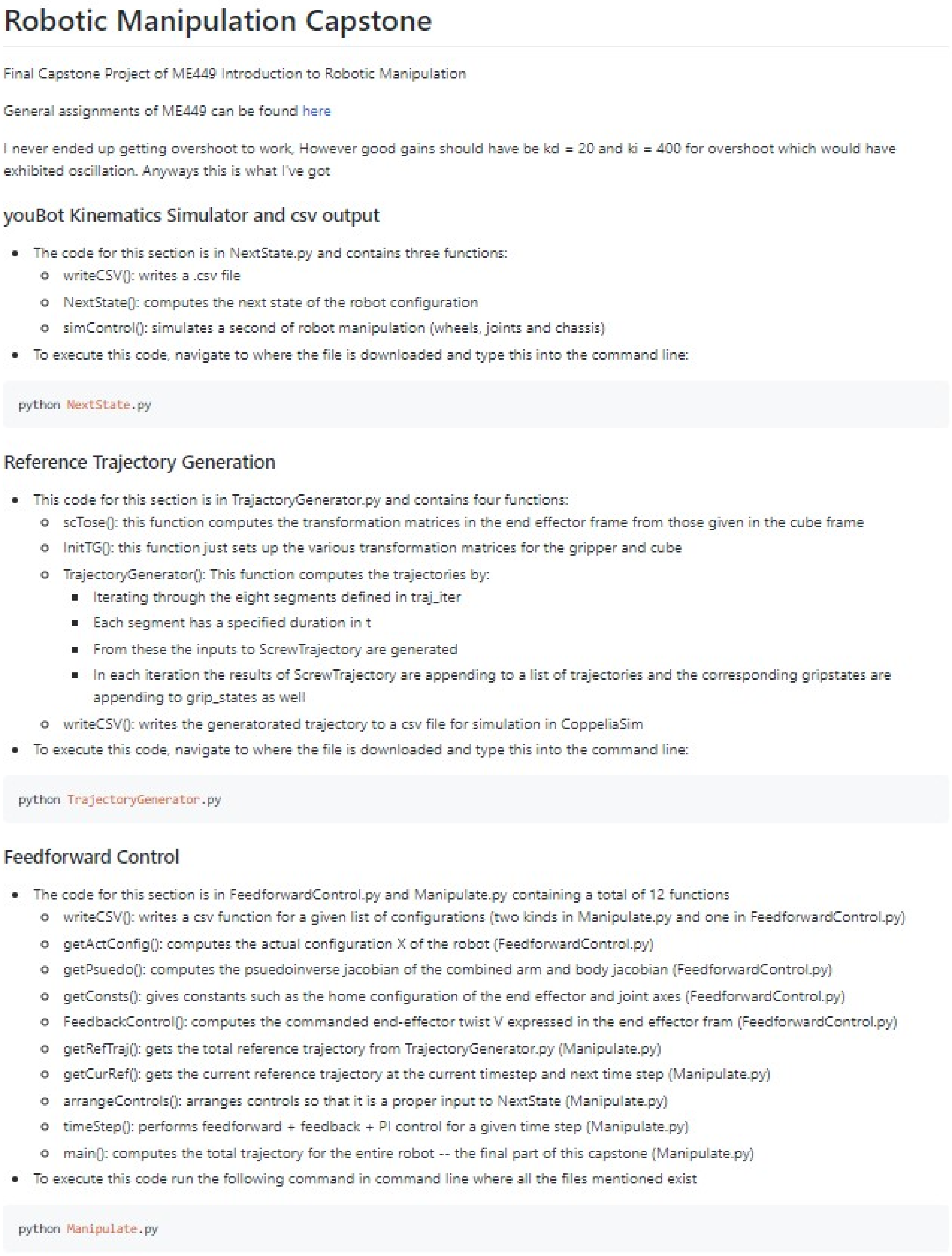

Robotic Manipulation Capstone

Final Capstone Project of ME449 Introduction to Robotic Manipulation

General assignments of ME449 can be found here

I never ended up getting overshoot to work, However good gains should have be kd = 20 and ki = 400 for overshoot which would have exhibited oscillation. Anyways this is what I've got

youBot Kinematics Simulator and csv output

- The code for this section is in NextState.py and contains three functions:
  - writeCSV(): writes a .csv file
  - NextState(): computes the next state of the robot configuration
  - simControl(): simulates a second of robot manipulation (wheels, joints and chassis)
- To execute this code, navigate to where the file is downloaded and type this into the command line:

```
python NextState.py
```

Reference Trajectory Generation

- This code for this section is in TrajactoryGenerator.py and contains four functions:
  - scTose(): this function computes the transformation matrices in the end effector frame from those given in the cube frame
  - InitTG(): this function just sets up the various transformation matrices for the gripper and cube
  - TrajectoryGenerator(): This function computes the trajectories by:
    - Iterating through the eight segments defined in traj_iter
    - Each segment has a specified duration in t
    - From these the inputs to ScrewTrajectory are generated
    - In each iteration the results of ScrewTrajectory are appending to a list of trajectories and the corresponding gripstates are appending to grip_states as well
  - writeCSV(): writes the generatorated trajectory to a csv file for simulation in CoppeliaSim
- To execute this code, navigate to where the file is downloaded and type this into the command line:

```
python TrajectoryGenerator.py
```

Feedforward Control

- The code for this section is in FeedforwardControl.py and Manipulate.py containing a total of 12 functions
  - writeCSV(): writes a csv function for a given list of configurations (two kinds in Manipulate.py and one in FeedforwardControl.py)
  - getActConfig(): computes the actual configuration X of the robot (FeedforwardControl.py)
  - getPsuedo(): computes the psuedoinverse jacobian of the combined arm and body jacobian (FeedforwardControl.py)
  - getConsts(): gives constants such as the home configuration of the end effector and joint axes (FeedforwardControl.py)
  - FeedbackControl(): computes the commanded end-effector twist V expressed in the end effector fram (FeedforwardControl.py)
  - getRefTraj(): gets the total reference trajectory from TrajectoryGenerator.py (Manipulate.py)
  - getCurRef(): gets the current reference trajectory at the current timestep and next time step (Manipulate.py)
  - arrangeControls(): arranges controls so that it is a proper input to NextState (Manipulate.py)
  - timeStep(): performs feedforward + feedback + PI control for a given time step (Manipulate.py)
  - main(): computes the total trajectory for the entire robot -- the final part of this capstone (Manipulate.py)
- To execute this code run the following command in command line where all the files mentioned exist

```
python Manipulate.py
```