<u>Robotics II</u>
Project
Due: 4/27/2018

The link .stl's, dimensions, and mass parameters of the Rethink Robotics Sawyer 7 degree-of-freedom robot are available on canvas. Assume that Sawyer's joints are actuated with 24 volt Pittman DC054B-1 motors with Pittman 19.7:1 G51A spur gearboxes. Also, assume that all friction in the joints arises from the motors. For this robot perform the following:

Part 1: Kinematics and Dynamics

1) Create a function that draws Sawyer in Matlab according to a given vector of configuration coordinates. Use this to verify that your forward kinematics are correct.

2) Use Lagrangian mechanics to find the dynamical model of the robot. Using this dynamical model, create a function in Matlab that computes the joint velocities and accelerations of the robot given an input of joint motor voltages.

3) Implement the recursive Newton-Euler dynamics method in a Matlab function. This should also compute the joint velocities and accelerations of the robot given an input of motor voltages. Verify that this function returns the same values as 2) for a given set of input voltages and system states.

4) Develop a simulation for Sawyer using the dynamics obtained with Lagrangian mechanics. Develop another simulation using the recursive dynamics. Using plots of the joint angles versus time, verify that 2) and 3) are equivalent. Demonstrate this to me.

Part 2: Joint Space Control
For the following, turn in a properly labeled figure of subplots of the joint angles versus time where the setpoints or trajectories (you choose these) are also plotted as dashed red lines.

Part A: Position Controllers
1) Design and simulate a proportional-derivative feedback position controller for the robot.

2) Design and simulate a proportional-derivative plus gravity compensation feedback position controller for the robot.

3) Design and simulate a computed-torque with a proportional derivative outer-loop position controller for the robot.

Part B: Trajectory Tracking Controllers
1) Design and simulate a proportional derivative feedback trajectory tracking controller for the robot.

2) Design and simulate a proportional-derivative plus gravity compensation feedback position controller for the robot.

3) Design and simulate a computed-torque trajectory tracking controller with a proportional derivative outer-loop for the robot.

Part 3: Task Space Control

Part A: Required Functions
1) Write a Matlab function that computes the position and orientation (forward kinematics) of the end-effector of Sawyer given an input of joint angles.
2) Find the geometric Jacobian for the end-effector of Sawyer and write a Matlab function that computes it given an input of joint angles.

For the following, turn in properly labeled subplots of the end-effector's position and orientation versus time where the task-space setpoints or trajectories are plotted as red dashed lines.
Part B: Position Controllers
1) Design and simulate an inverse Jacobian position controller using proportional-derivative feedback.

2) Design and simulate an inverse Jacobian position controller using proportional-derivative feedback and gravity compensation.

3) Design and simulate a transposed Jacobian position controller using proportional-derivative feedback.

4) Design and simulate a transposed Jacobian position controller using proportional-derivative feedback and gravity compensation.

5) Design and simulate a computed-torque inverse Jacobian controller using proportional-derivative feedback.

6) Design and simulate a computed-torque transposed Jacobian controller using proportional-derivative feedback.

Part C: Trajectory Tracking Controllers

1) Design and simulate an inverse Jacobian trajectory tracking controller using proportional-derivative feedback.

2) Design and simulate an inverse Jacobian trajectory tracking controller using proportional-derivative feedback and gravity compensation.

3) Design and simulate a transposed Jacobian trajectory tracking controller using proportional-derivative feedback.

4) Design and simulate a transposed Jacobian trajectory tracking controller using proportional-derivative feedback and gravity compensation.

5) Design and simulate a computed-torque inverse Jacobian trajectory tracking controller using proportional-derivative feedback and feed-forward acceleration.

6) Design and simulate a computed-torque transposed Jacobian trajectory tracking controller using proportional-derivative feedback and feed-forward acceleration.