Robotics 16-748: Problem Set #4

Due **November 10th, 11:59 PM**

Problem #1: In this problem you will implement the iLQR algorithm. You will again use the simple pendulum *with damping* as your model system. Set each of the system parameters, expect the damping coefficient, equal to one; set $b = 0.3$.

a) **(5 points)** Generate desired state and control trajectories for the augmented pendulum

$$\ddot{\theta} = u - (\theta^2 - 1)\dot{\theta} - \sin(\theta)$$

using the direct collocation trajectory optimization algorithm you developed for Problem Set #3. Allow the state values to vary between $[-\infty, \infty]$ and constrain the controls to be $|u| \leq 1.5$. Use 100 grid points and set $dt = 0.1$. Constrain the initial state to be zero and terminal state to be $x(T) = [-\pi, 0]$. Use a quadratic cost in both the state and control and set the corresponding weight matrices to be the identity.

b) **(5 points)** Use direct collocation again, but this time to derive an initial nominal trajectory for the simple pendulum using the same parameters and constraints as above but with the terminal constraint $x(T) = [\pi, 0]$. Plot both the nominal as well as desired trajectories.

c) **(25 points)** Use the nominal trajectories to linearize the simple pendulum dynamics and then design a controller that attempts to track the desired state and control trajectories generated in a). Execute five iterations of this procedure. You will need to update your nominal state and control trajectories in each iteration. Use $Q = 100 * I$ and $R = 1$ to derive the controllers.

d) **(10 points)** Redefine your desired control signal to be $u_d = 0$. Use the same desired state trajectories from above. Repeat the iterative tracking procedure for the same number of steps you used in c).

Problem #2: In this question you will analyze the stability of the rimless wheel "walking" down a uniform slope. Start by downloading Russ Tedrake's rimless wheel simulation (a link to the rimlessWheel.m file is posted in the course material on the class website). Use $\gamma = 0.08$.

a) **(15 points)** Modify the simulation code such that it outputs the numerical value of the one-step return map corresponding to the system's velocity right after an impact event occurs (after the transfer of support), i.e., the Poincare map defined right after an impact occurs. Plot the Poincare map over the interval $-3 : 0.05 : 3$. Also plot the $x[n] = x[n+1]$ line. Use Matlab's plotting tools to identify the system's fixed points.

b) **(40 points)** Numerically find the fixed-point of the function

$$x^* - P(x^*) = 0$$

that we discussed in class using Newton's method for the initial condition $x = (-0.3127, 4.5)$. Once you have converged to the fixed point, evaluate its stability by analyzing the eigenvalues of the Jacobian of the Poincare map.