This problem set is due *by 11:59pm* on **Tuesday, March 31**.

Problem 1 (Optimal Swing-up for the Simple Pendulum) *Consider the dynamics of the simple pendulum given by:* $ml^2\ddot{q} + b\dot{q} + mgl\sin(q) = u$*, where* $m = 1$*,* $l = 1$*,* $b = 0.1$*,* $g = 9.8$*. Your goal for this problem will be to minimize the cost function*

$$J = 50(\pi - q(t_f))^2 + 50\dot{q}^2(t_f) + \int_0^{t_f} \frac{1}{100} u^2(t)dt \tag{1}$$

using the SNOPT sequential quadratic programming (SQP) package that was discussed in class, while computing the required gradients via back-propagation through time (BPTT). When computing the gradients for this problem, take note that the cost in (1) includes a terminal cost that penalizes the final state $[q(t_f), \dot{q}(t_f)]^T$ *at* $t_f = 2.5$ *seconds. We will parameterize the policy as an open-loop tape of actions using a tape with a control time-step of .01 seconds and 251 entries (from* $t = 0$ *to* $t = 2.5$*).*

a) **Back-Propagation Through Time (BPTT)** *The SQP methods discussed in class require the gradients of the plant, cost, and policy in order to perform the update to the parameters. Determine the (explicit) expressions for* $\mathbf{F_x}$*,* $\mathbf{G_x}$*,* $\mathbf{F}_\alpha$*, and* $\mathbf{G}_\alpha$*, including policy gradients. (See the course notes for the definition of these expressions).*

b) The Lagrange multiplier derivation of BPTT (section 12.3.3 in the notes) sets $\mathbf{y}(T) = \mathbf{0}$ *(i.e. zeroes out the lagrange multiplier at the final time step). However, for the cost function given in eq.(1) this does not hold. Determine* $\mathbf{y}(T)$ *for this case.*

c) Download the files `pend_snopt.m` *and* `pendfun.m` *from the course website, as well as the SNOPT student version*[1]*. These files contains a basic matlab skeleton implementation calling SNOPT and using the BPTT algorithm to calculate the gradients for the swing-up task. You will need to complete the implementation by computing the appropriate gradients, integrating the adjoint equations, filling in the cost functions, etc. We have commented the code where necessary. Submit your code and a trajectory plot for your final solution. You should be able to converge to a solution that gets the pendulum sufficiently close to the top (to within* ~ 5 *degrees).*

[1] This can be found at `http://www.ccom.ucsd.edu/~peg/Software.html`. Simply download, unzip, and include the snopt package in your MATLAB path.

Problem 2 (Single-Pump Swing-up for the Cart-Pole) *For this problem, you will solve for a policy that will achieve the cart-pole swing-up task by minimizing the cost function*

$$J = 50(\pi - \theta(t_f))^2 + 50\dot{\theta}^2(t_f) + 50x(t_f)^2 + 50\dot{x}^2(t_f) + \int_0^{1.5} \frac{1}{100}u^2(t)dt \quad (2)$$

where θ, x denote the pole angle and cart position, respectively. Parameterize your policy strictly by time, i.e.

$$u[i] = \alpha_i \quad (3)$$

where i is the i^{th} discrete time step of your simulation. You should use your implemen tation for Problem 1 as a starting point, replacing the necessary plant/policy gradients (again use SNOPT to perform the optimization, and BPTT to compute the gradients). Submit your code and plots of the final trajectory. You should be able to converge to a solution that gets the pendulum sufficiently close to the top (to within ~ 5 degrees), e.g. within the basin of attraction of LQR (which you don't need to implement for this exercise). The equations of motion for this problem are:

$$\ddot{x} = \frac{1}{1+\sin^2\theta}\left[u + \sin\theta\left(\frac{\dot{\theta}^2}{2} + g\cos\theta\right)\right] \quad (4)$$

$$\ddot{\theta} = \frac{2}{(1+\sin^2\theta)}\left[-u\cos\theta - \frac{\dot{\theta}^2}{2}\cos\theta\sin\theta - 2g\sin\theta\right] \quad (5)$$

where we have set $m_c = 1, m_p = 1, l = 0.5$.

Problem 3 (The Rimless Wheel) *For this problem, you should be able to compute all of the quantities in closed form. A simulation of the rimless wheel is available on the class website, in case you find it useful for confirming your answers or your intuition. Also, see the course notes for the definition of variables.*

a) **Passive Dynamics.** *If we assume that the wheel is started in a configuration directly after a transfer of support, then forward walking occurs when the system has an initial velocity $\dot{\theta}(0^+) > \omega_1$. When the next foot touches down the conversion of potential energy into kinetic energy yields the velocity before touchdown $\dot{\theta}(t^-)$. Derive the expressions for ω_1 and $\dot{\theta}(t^-)$. Show your work. Describe in words why $\omega_1 = 0$ when $\gamma = \alpha$, and why ω_1 is nonexistent when $\gamma > \alpha$.*

b) **Gait Design.** *Consider a rimless wheel w/ $m = 1$ kg, $g = 9.8$ m/s^2, $\alpha = \frac{\pi}{8}$ rad, and $\gamma = 0.08$ rad. For what leg length, l, is the steady-state rolling speed of the system (measured when the pendulum is vertical over the stance feet: $\theta = 0$) equal to 4.0 rad/sec?*

c) **Terrain "Stability".** *Consider a rimless wheel with $m = 1$ kg, $l = 1$ m, $g = 9.8$ m/s^2, and $\alpha = \frac{\pi}{8}$ rad rolling down a slope, $\gamma = 0.16$ rad. Some time after reaching the rolling steady state, the terrain shallows to $\gamma = 0.02$ rad for a distance d, then returns to (and remains at) $\gamma = 0.08$ rad. You may assume that the shallow slope begins and ends precisely at a position of touchdown (not somewhere between steps). What is the largest value of d, subject to these constraints, for which the system will be at a rolling fixed point when $\gamma = 0.08$ as $t \rightarrow \infty$?*

6.832 Underactuated Robotics
Spring 2009