CS284 Problem Set 2

Charles Liu
cliu02@g.harvard.edu
February 18, 2016

1: Linear Optimal Control

$$\begin{aligned} \text{Dynamics: } \dot{x} &= -2x + 5u \\ \text{Infinite Cost Function: } J &= \int_0^\infty (10x^2 + u^2)dt \end{aligned}$$

(a) Given optimal cost-to-go function $J^* = px^2$, we can plug into the HJB equation

$$\begin{aligned} 0 &= min_u[g(x,u) + \frac{\delta J}{\delta x} f(x,u)] \\ &= min_u[10x^2 + u^2 + 2px(-2x + 5u)] \\ &= min_u[(10 - 4p)x^2 + u^2 + 10pxu] \end{aligned}$$

Finding the minimum, we take $\frac{\delta}{\delta u}$, and find

$$\begin{aligned} 2u + 10px &= 0 \\ u &= -5px \end{aligned}$$

Substituting back into the HJB equation we get

$$\begin{aligned} 0 &= (10 - 4p)x^2 + (-5px)^2 + 10px(-5px) \\ &= (10 - 4p)x^2 + 25p^2x^2 - 50p^2x^2 \\ &= (10 - 4p - 25p^2)x^2 \end{aligned}$$

Since the cost must be positive, p must be the positive root of the quadratic: $\frac{-2+\sqrt{254}}{25}$

(b) From part a, $u = -5px = -kx$, meaning $k = 5p = \frac{-2+\sqrt{254}}{5}$

(c) Changing $g(x,u)$, the equation becomes

$$\begin{aligned} 0 &= min_u[(20 - 4p)x^2 + 2u^2 + 10pxu] \\ \frac{\delta}{\delta u} &= 4u + 10px = 0 \\ u^* &= \frac{-5px}{2} \end{aligned}$$

Substituting back into the HJB equation

$$0 = min_u[(20 - 4p)x^2 + 2(\frac{-5px}{2})^2 + 10px(\frac{-5px}{2})] \tag{1}$$

$$= min_u[(20 - 4p - \frac{25}{2}p^2)x^2] \tag{2}$$

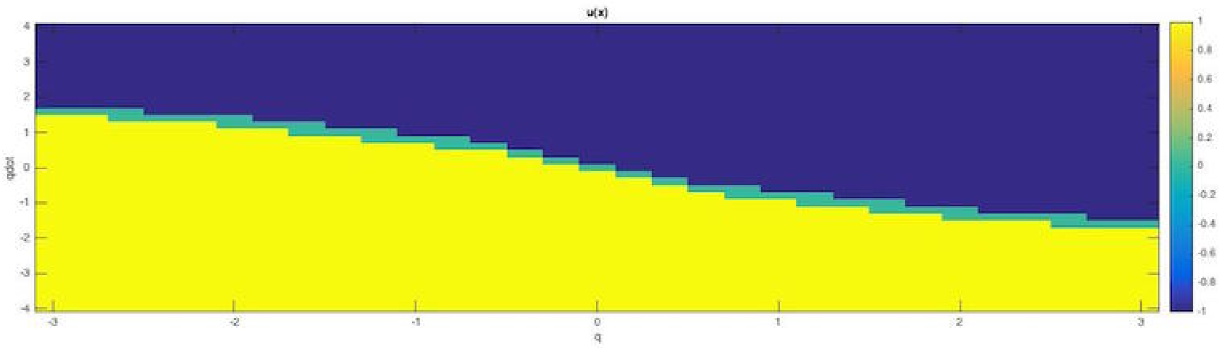
The positive root $p = \frac{-4+2\sqrt{254}}{25}$, and $u = -\frac{5px}{2} = -kx$ means $k = \frac{5p}{2} = \frac{-2+\sqrt{254}}{5}$, so (iii) p was doubled

2: Value Iteration

(a) Recall the analytical solution for the double integrator:

$$u = \begin{cases} 1 & \text{if } (\dot{q} < 0 \text{ and } q \leq \frac{1}{2}\dot{q}^2) \text{ or } (\dot{q} \geq 0 \text{ and } q < -\frac{1}{2}\dot{q}^2) \\ 0 & q = 0 \text{ and } \dot{q} = 0 \\ -1 & \text{otherwise} \end{cases}$$

From these equations the boundaries where you can brake to 0 perfectly have deterministic policies, whereas from running the value iteration the policy hovered somewhere in the middle as shown in the graph.



In particular, in the case where $q(0) = 1$, the boundary condition for braking straight to the origin with $u = 1$ is when $1 = \frac{1}{2}\dot{q}^2(0)$, meaning $\dot{q}^2(0) = -\sqrt{2}$. However, after returning the MarkovDecisionProcessPolicy object from the DoubleIntegrator code, we see the following:

```
>> P_mat = reshape(ubins(PI.PI),length(xbins{1}),length(xbins{2}));
>> P_mat(find(xbins{1}==0),find(xbins{2}==0))

ans =

     0

>> P_mat(find(xbins{1}==1),find(xbins{2}==-1))

ans =

     1
```

The first line formats the policy vector to a matrix, then we see that the origin point returns 0 as expected. The point (1,-1) returns $u = 1$ meaning brake, when we know that it should be $u = -1$ (accelerate) until we hit the boundary and then brake.

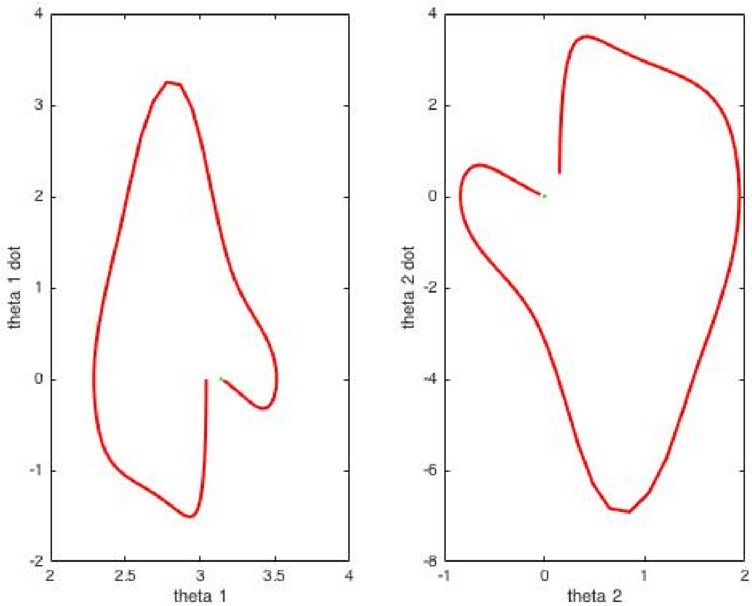
(b) The cost function is 0 if x=[0,0] and 1 otherwise. If xbins doesn't contain the point [0,0] it will never converge. For example, the following initialization of xbins goes from -.1 to .1, skipping 0:

$$xbins = [-3.1 : .2 : 3.1], [-4.1 : .2 : 4.1];$$

3: Acrobot LQR

(a) $u = \pi(A, B, Q, R, K, x, x_0) = -K(x - x_0)$ where x_0 is the desired fixed point, in this case the upright position $[[\pi, 0], [0, 0]]$

(b)



4: Partial Feedback Linearization

Equations of motion for Acrobot:

$$\begin{aligned}
&(I_1 + I_2 + m_2 l_1^2 + 2 m_2 l_1 l_{c2} c_2)\ddot{q}_1 \\
&+(I_2 + m_2 l_1 l_{c2} c_2)\ddot{q}_2 - 2 m_2 l_1 l_{c2} s_2 \dot{q}_1 \dot{q}_2 - m_2 l_1 l_{c2} s_2 \dot{q}_2^2 \\
&+m_1 g l_{c1} s_1 + m_2 g(l_1 s_1 + l_{c2} s_{1+2}) \quad = \quad 0 \\
&(I_2 + m_2 l_1 l_{c2} c_2)\ddot{q}_1 + I_2 \ddot{q}_2 + m_2 l_1 l_{c2} s_2 \dot{q}_1^2 + m_2 g l_{c2} s_{1+2} \quad = \quad \tau
\end{aligned}$$

Using collocated partial feedback linearization, for some desired u' we set $\ddot{q}_2 = u'$. We then have

$$\begin{aligned}
\ddot{q}_2 &= u' \\
-(I_2 + m_2 l_1 l_{c2} c_2)u' + 2 m_2 l_1 l_{c2} s_2 \dot{q}_1 \dot{q}_2 + m_2 l_1 l_{c2} s_2 \dot{q}_2^2 & \\
-m_1 g l_{c1} s_1 - m_2 g(l_1 s_1 + l_{c2} s_{1+2}) & \\
-------------------------- & \\
(I_1 + I_2 + m_2 l_1^2 + 2 m_2 l_1 l_{c2} c_2) &= \ddot{q}_1
\end{aligned}$$

To simplify, if

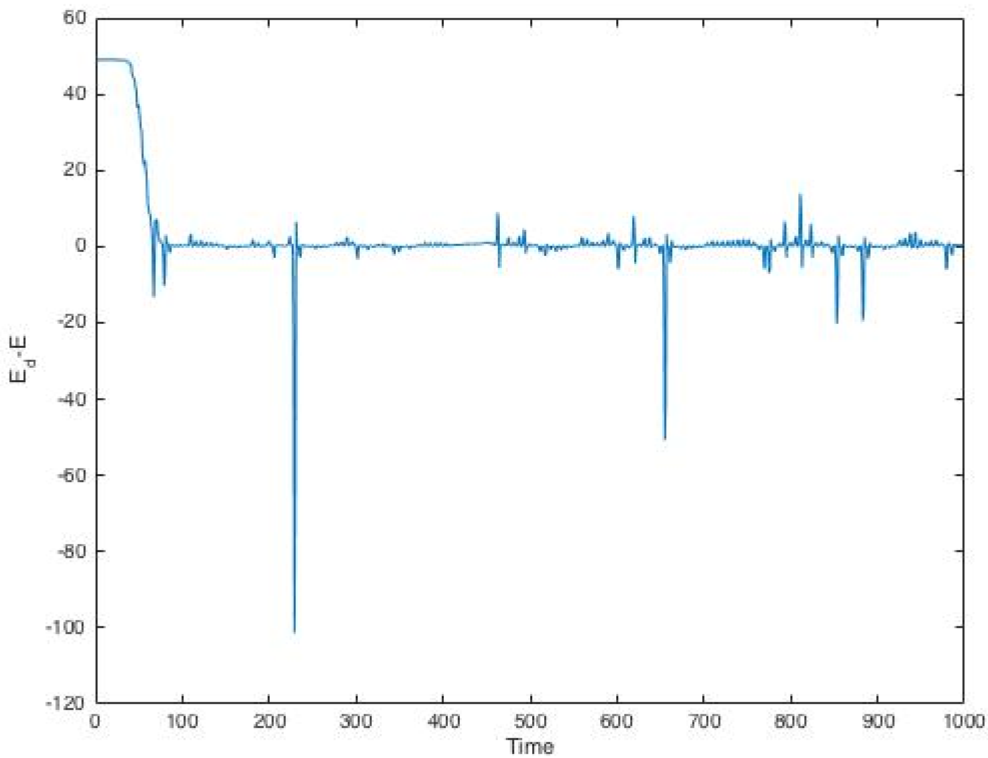
$$\ddot{q} = \begin{bmatrix} h_1 & h_2 \\ h_2 & h_3 \end{bmatrix}^{-1} \left(\begin{bmatrix} 0 \\ 1 \end{bmatrix} u - \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \right)$$

Then for some $\ddot{q}_2 = u'$

$$\begin{aligned} \tau &= h_2\ddot{q}_1 + h_3 u' + c_2 \\ \ddot{q}_1 &= \frac{-(h_2 u' + c_1)}{h_1} \end{aligned}$$

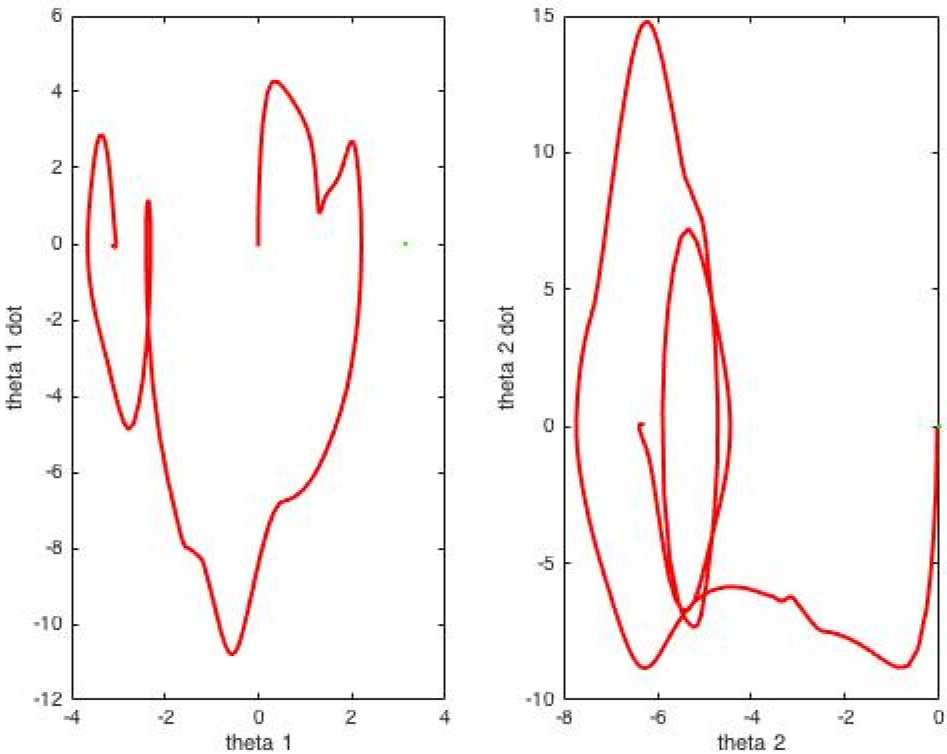
5: Energy Shaping

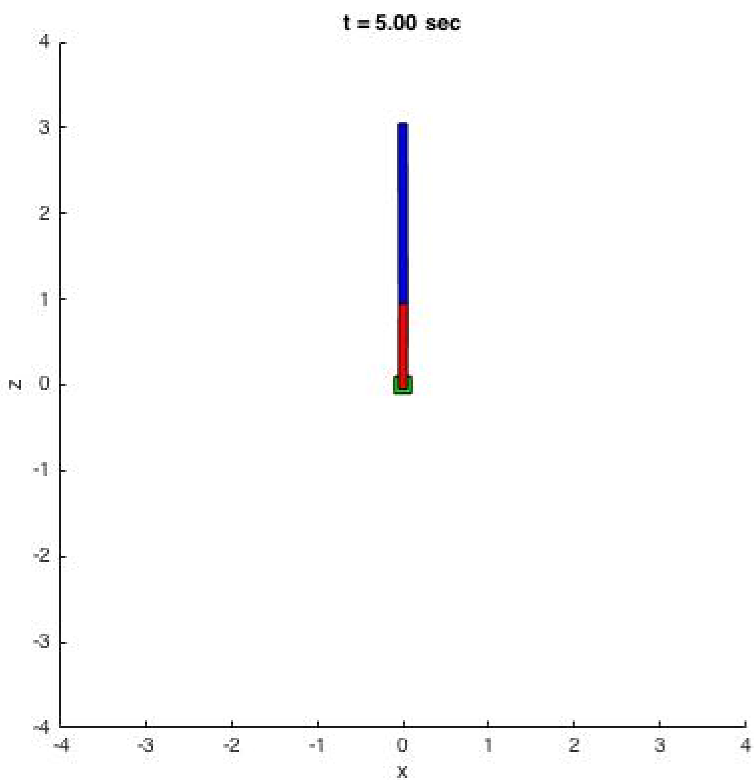
The system's energy eventually oscillates around the desired energy, but the system never reaches the upright configuration as the upper link is in constant motion. $E_d - E$ over time:



6: Acrobot Swing-up and Balance

Looks like it has q_1 going to $-\pi$ instead of π for some reason





7: Lecture Survey

(c) Pacing is about right, generally need to review the material after but am understanding concepts.

8: Pset Survey

10 hours