ECE-597: Optimal Control Homework #6

Due: Wednesday May 3, 2006

1) Assume we have the optimization problem

$$\begin{aligned} \text{minimize } J &= x(N)^2 + \sum_{k=0}^{k=N-1} x(k)u(k) \\ \text{subject to} \quad & x(k+1) = x(k)u(k) + u(k)^2 \\ & x(k) \in \{-1, 0, 1, 2\}, \quad u(k) \in \{-1, 1\}, \quad N = 2 \end{aligned}$$

Use the dynamic programming method to determine the optimal cost to go and the optimal controls starting at any $x(0)$.
Note: For what its worth, I get $J_0^*(-1) = 0, J_0^*(0) = -1, J_0^*(1) = 0, J_0^*(2) = -3$

2) Assume we have the optimization problem

$$\begin{aligned} \text{minimize } J &= (x(N) - 1)^2 + \sum_{k=0}^{k=N-1} \{x(k)^2 + \frac{1}{2}u(k)^2\} \\ \text{subject to} \quad & x(k+1) = x(k) + u(k) \\ & x(k) \in \{-2, -1, 0, 1, 2\}, \quad u(k) \in \{-1, 0, 1\}, \quad N = 2 \end{aligned}$$

Use the dynamic programming method to determine the optimal cost to go and the optimal controls starting at any $x(0)$.
Note: For what its worth, I get $J_0^*(-2) = 7, J_0^*(-1) = 2, J_0^*(0) = 0.5, J_0^*(1) = 2, J_0^*(2) = 5.5$

3) The program **dynamic_tracker.m** implements a dynamic programming approach to solving the problem of finding the optimal control signal $u(t)$ to minimize the function

$$J = (y(N) - r(N))^T Q_f (y(N) - r(N)) + \sum_{k=0}^{k=N-1} \left[(y(k) - r(k))^T Q (y(k) - r(k)) + u(k)^T R u(k)\right]$$

where $r(k)$ is a known signal we want to track, and we have the continuous-time state equations

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned}$$

which, assuming a zero order hold, correspond to the discrete-time equations

$$\begin{aligned} x(k+1) &= Gx(k) + Hu(k) \approx (I + A\Delta T)x(k) + (B\Delta T)u(k) \\ y(k) &= Cx(k) \end{aligned}$$

Note that since with dynamic programming it is possible to change the time interval (ΔT) we need to be able to make our model reasonably accurate for any time step size. This is clearly an optimization problem with *soft* terminal constraints. For this problem assume $A = 1$, $B = 1$, $C = 1$, and $0 \leq t \leq 2$.

We want to look at trying to track the inputs

$$\begin{aligned} r(t) &= u(t) \\ r(t) &= tu(t) \\ r(t) &= cos(2\pi t) \end{aligned}$$

Your are to modify and run the code **dynamic_tracker.m** to do the following:

- Have the dynamic system track the input signal as well as possible throughout the interval $0 \leq t \leq 2$
- Have the dynamic system have the same value at the end of the interval (at $t = 2$) as $r(t)$ while minimizing the value of the control signal $u(t)$. Note that since we only have soft terminal constraints we cannot really force this to happen, we can only suggest it by changing the weights.

You need to do each of the above for the initial conditions $x(0) = 0$ and $x(0) = 0.5$

Since this dynamic tracking problem can be solved (assuming R is invertible), the program will also plot the results using the solution to the problems above. For each case, the optimal cost will be displayed at the top of the graph. If your meshes are fine enough the values should be nearly equal, though the optimal cost for the dynamic programming solution will often be larger.

As you go through these problems, you will need to change the range of possible x values and the range of possible u values. You will have to iterate a bit to determine a good range for these values. (If the program does not plot the results for $0 \leq t \leq 2$ you will need to change these values.) Your final results should look pretty much like those of the optimal tracking algorithm. You should have 12 graphs to turn in with this part.