ECE-597: Optimal Control
Homework #3

Due: Thursday September 27, 2007

1. *Discrete-time Linear Quadratic Regulator via Dynamic Programming* In this problem we will derive the recursive equations the need to be solved in order to implement a discrete linear quadratic regulator (LQR).

Assume we have the following discrete-time state variable model

$$x(k+1) \;=\; Gx(k) + Hu(k)$$

We want to find the control sequence $u(k)$ to minimize the following performance index

$$J \;=\; \frac{1}{2}x(N)^T Q_f x(N) + \frac{1}{2}\sum_{k=1}^{N-1}\left\{x(k)^T Q x(k) + u(k)^T R u(k)\right\}$$

where $Q = Q^T$, $R = R^T$, $Q_f = Q_f^T > 0$

We will try and solve this recursively, using the ideas from Dynamic Programming. First we define

$$J_i \;=\; \frac{1}{2}x(N)^T Q_f x(N) + \frac{1}{2}\sum_{k=i}^{N-1}\left\{x(k)^T Q x(k) + u(k)^T R u(k)\right\}$$

So $J_0 = J$. For $i = N$ we have $J_N^* = \frac{1}{2}x(N)^T Q_f x(N)$. Next, we look at J_i for $i = N-1$,

$$J_{N-1} \;=\; \frac{1}{2}x(N)^T Q_f x(N) + \frac{1}{2}x(N-1)^T Q x(N-1) + \frac{1}{2}u(N-1) R u(N-1)$$

Now we (you) need to find the control $u(N-1)$ to minimize this expression.

a) Use the state variable model to eliminate $x(N)$.

b) Write out all of the terms in J_{N-1}. If you combine terms correctly, you should have five terms.

c) Find the optimal $u(N-1)$ by taking the derivative of J_{N-1} with respect to $u(N-1)$. You should get

$$\begin{aligned} u^*(N-1) &= -K(N-1)x(N-1) \\ K(N-1) &= \left[R + H^T S(N) H\right]^{-1} H^T S(N) G \end{aligned}$$

Where we have written $Q_f = S(N)$ for reasons that will become clear shortly. Note that we have state variable feedback!

d) Now put $u^*(N-1)$ into J_{N-1} to produce J^*_{N-1}. After some manipulation, you should get

$$
\begin{aligned}
J^*(N-1) &= \frac{1}{2}x(N-1)^T S(N-1)x(N-1) \\
S(N-1) &= \{G - HK(N-1)\}^T S(N) \{G - HK(N-1)\} + K(N-1)^T RK(N-1) + Q
\end{aligned}
$$

e) Now we need to look at J_i for $i = N-2$

$$
J_{N-2} = \frac{1}{2}x(N-1)^T S(N-1)x(N-1) + \frac{1}{2}x(N-2)^T Qx(N-2) + \frac{1}{2}u(N-2)Ru(N-2)
$$

We need to find the $u(N-2)$ to minimize this function. At this point, those of you with even the remotest clue should realize that *we have already solved this problem!* All we need to do is change indices $N \rightarrow N-1$.

Our solution is as follows:

$S(N) = Q_f$

for $i = N-1, ...0$

$$
\begin{aligned}
K(i) &= \left[R + H^T S(i+1)H\right]^{-1} H^T S(i+1)G \\
u(i) &= -K(i)x(i) \\
S(i) &= \{G - HK(i)\}^T S(i+1) \{G - HK(i)\} + K(i)^T RK(i) + Q
\end{aligned}
$$

2. 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

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

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 an optimization problem with *soft* terminal constraints, since we cannot force our system to exactly match $r(N)$ at the final time, we only penalize deviations from this. An optimization problem with *hard* terminal constraints would force $Y(N) = R(N)$.

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 $u(t)$.

You need to do each of the above for the initial condition $x(0) = 0.5$ For these programs you need to be sure $Q > 0$, $R > 0$, and $Q_f > 0$

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. Try not to be too depressed that we did not find the solution to this, we will next week.

As you go through these problems, you may 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 6 graphs to turn in with this part.
You should notice three things from these plots:

- The cost for the dynamic programming routine is always greater than for the optimal tracker
- The control signal is smoother for the optimal tracker than for the dynamic programming routine
- For tracking the sinusoid, the control signal $u(t)$ actually leads the signal $r(t)$ we are trying to track