EE363 homework 2 solutions

1. *Derivative of matrix inverse.* Suppose that $X : \mathbf{R} \to \mathbf{R}^{n\times n}$, and that $X(t)$ is invertible. Show that

$$\frac{d}{dt}X(t)^{-1} = -X(t)^{-1}\left(\frac{d}{dt}X(t)\right)X(t)^{-1}.$$

Hint: differentiate $X(t)X(t)^{-1} = I$ with respect to t.

Solution: Differentiating $X(t)X(t)^{-1}$ with respect to t, we get

$$\begin{aligned} 0 &= \frac{d}{dt}I \\ &= \frac{d}{dt}X(t)X(t)^{-1} \\ &= X(t)\left(\frac{d}{dt}X(t)^{-1}\right) + \left(\frac{d}{dt}X(t)\right)X(t)^{-1}. \end{aligned}$$

We conclude that

$$X(t)\left(\frac{d}{dt}X(t)^{-1}\right) = -\left(\frac{d}{dt}X(t)\right)X(t)^{-1},$$

and so

$$\frac{d}{dt}X(t)^{-1} = -X(t)^{-1}\left(\frac{d}{dt}X(t)\right)X(t)^{-1}.$$

2. *Infinite horizon LQR for a periodic system.* Consider the system $x_{t+1} = A_t x_t + B_t u_t$, where

$$A_t = \begin{cases} A^{\mathrm{e}} & t \text{ even} \\ A^{\mathrm{o}} & t \text{ odd} \end{cases} \qquad B_t = \begin{cases} B^{\mathrm{e}} & t \text{ even} \\ B^{\mathrm{o}} & t \text{ odd} \end{cases}$$

In other words, A and B are periodic with period 2. We consider the infinite horizon LQR problem for this time-varying system, with cost

$$J = \sum_{\tau=0}^{\infty}\left(x_\tau^T Q x_\tau + u_\tau^T R u_\tau\right).$$

In this problem you will use dynamic programming to find the optimal control for this system. You can assume that the value function is finite.

   (a) Conjecture a reasonable form for the value function. You do not have to show that your form is correct.

   (b) Derive the Hamilton-Jacobi equation, using your assumed form. *Hint:* you should get a pair of coupled nonlinear matrix equations.

(c) Suggest a simple iterative method for solving the Hamilton-Jacobi equation. You do not have to prove that the iterative method converges, but do check your method on a few numerical examples.

(d) Show that the Hamilton-Jacobi equation can be solved by solving a single (bigger) algebraic Riccati equation. How is the optimal u related to the solution of this equation?

Remark: The results of this problem generalize to general periodic systems.

Solution:

(a) In the time-invariant infinite horizon LQR problem, the value function $V(z)$ does not depend on time. This is because

$$\begin{aligned} V_{t_0}(z) &= \min_{u(t_0),\ldots} \sum_{\tau=t_0}^{\infty} (x(\tau)^T Q x(\tau) + u(\tau)^T R x(\tau)) = \\ V_{t_1}(z) &= \min_{u(t_1),\ldots} \sum_{\tau=t_1}^{\infty} (x(\tau)^T Q x(\tau) + u(\tau)^T R u(\tau)) \end{aligned}$$

as long as $z = x_{t_0} = x_{t_1}$. A similar statement can be made for the periodic case. That is, $V_0(z) = V_{2k}(z)$ for $k = 1, 2, \ldots$ as long as $z = x_0 = x_{2k}$. Similarly $V_1(z) = V_{2k+1}(z)$ as long as $z = x_1 = x_{2k+1}$. Therefore, we can define our value function to be $V_o(z)$ when t is odd and $V_e(z)$ when t is even.

In the general time-varying *finite* horizon case it can be shown that the value function is quadratic. Since this holds for any horizon N, it is reasonable to suggest (and indeed is true) that the value function remains quadratic as $N \to \infty$. A reasonable form for the value function is therefore $V_t(z) = V_o(z) = z^T P_o z$ when t is odd and $V_t(z) = V_e(z) = z^T P_e z$ when t is even.

(b) For t even we get

$$\begin{aligned} V_e(z) &= z^T Q z + \min_w (w^T R w + V_o(A_e z + B_e w)) \\ &= z^T Q z + \min_w (w^T R w + (A_e z + B_t w)^T P_o (A_e z + B_t w)) \\ &= z^T (Q + A_e^T P_o A_e - A_e^T P_o B_e (R + B_e^T P_o B_e)^{-1} B_e^T P_o A_e) z \\ &= z^T P_o z, \end{aligned}$$

and we get a similar expression for t odd. Putting these together we have

$$\begin{aligned} P_o &= Q + A_o^T P_e A_o - A_o^T P_e B_o (R + B_o^T P_e B_o)^{-1} B_o^T P_e A_o, \\ P_e &= Q + A_e^T P_o A_e - A_e^T P_o B_e (R + B_e^T P_o B_e)^{-1} B_e^T P_o A_e. \end{aligned}$$

(c) A simple iterative scheme for finding P_o and P_e is given by repeatedly setting

$$\begin{aligned} P_o &:= Q + A_o^T P_e A_o - A_o^T P_e B_o (R + B_o^T P_e B_o)^{-1} B_o^T P_e A_o \\ P_e &:= Q + A_e^T P_o A_e - A_e^T P_o B_e (R + B_e^T P_o B_e)^{-1} B_e^T P_o A_e, \end{aligned}$$

starting from arbitrary positive semidefinite matrices P_o and P_e. If these converge to a fixed point, we have our value functions.

(d) The pair of equations discussed in (b) and (c) can be rewritten as a single matrix equation

$$P = \hat{Q} + A^T PA - A^T PB(\hat{R} + B^T PB)^{-1} B^T PA,$$

where

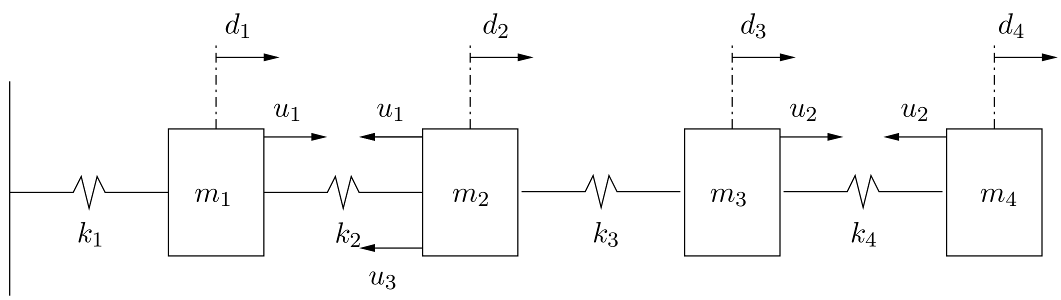
$$A = \begin{bmatrix} 0 & A_e \\ A_o & 0 \end{bmatrix}, \quad B = \begin{bmatrix} B_e & 0 \\ 0 & B_o \end{bmatrix}, \quad \hat{Q} = \begin{bmatrix} Q & 0 \\ 0 & Q \end{bmatrix},$$

$$\hat{R} = \begin{bmatrix} R & 0 \\ 0 & R \end{bmatrix}, \quad P = \begin{bmatrix} P_o & 0 \\ 0 & P_e \end{bmatrix}.$$

Note that this is simply an algebraic Riccati equation. Using the fact that $u_t = -(R + B_t^T P_{t+1} B_t)^{-1} B_t^T P_{t+1} A_t x_t$, the optimal control can be obtained from P by taking

$$u_t = \begin{cases} -(R + B_e^T P_o B_e)^{-1} B_e^T P_o A_e x_t = K_e x_t, & t \text{ even}, \\ -(R + B_o^T P_e B_o)^{-1} B_o^T P_e A_o x_t = K_o x_t, & t \text{ odd}. \end{cases}$$

3. *LQR for a simple mechanical system.* Consider the mechanical system shown below:



Here $d_1, \ldots, d_4$ are displacements from an equilibrium position, and $u_1, \ldots, u_3$ are forces acting on the masses. Note that u_1 is a tension between the first and second masses, u_2 is a tension between the third and fourth masses, and u_3 is a force between the wall (at left) and the second mass. You can take the mass and stiffness constants to all be one: $m_1 = \cdots = m_4 = 1$, $k_1 = \cdots = k_4 = 1$.

(a) Describe the system as a linear dynamical system with state $(d, \dot{d}) \in \mathbf{R}^8$.

(b) Using the cost function

$$J = \int_0^\infty \left(\|d(t)\|^2 + \|u(t)\|^2 \right) \, dt,$$

find the optimal state feedback gain matrix K. You may find the Matlab function `lqr()` useful, but check sign conventions: it's not unusual for the optimal feedback gain to be defined as $u = -Kx$ instead of $u = Kx$ (which is what we use).

(c) Plot $d(t)$ versus t for the open loop ($u(t) = 0$) and closed loop ($u(t) = Kx(t)$) cases using an arbitrary initial condition (but not, of course, zero).

(d) Solve the ARE for this problem using the method based on the Hamiltonian, described in lecture 4. Verify that you get the same result as you did using the Matlab function `lqr()`.

Solution:

(a) The equations of motion for this system can be written as

$$\begin{aligned}
m_1\ddot{d}_1 &= -k_1 d_1 + k_2(d_2 - d_1) + u_1 = -2d_1 + d2 + u_1 \\
m_2\ddot{d}_2 &= -k_2(d_2 - d_1) + k_3(d_3 - d_2) - u_1 - u_3 = d_1 - 2d_2 + d_3 - u_1 - u_3 \\
m_3\ddot{d}_3 &= -k_3(d_3 - d_2) + k_4(d_4 - d_3) + u_2 = d_2 - 2d_3 + d_4 + u_2 \\
m_4\ddot{d}_4 &= -k_4(d_4 - d_3) - u_2 = d_3 - d_4 - u_2
\end{aligned}$$

Defining $d_5 = \dot{d}_1, \ldots, d_8 = \dot{d}_4$ and $x = (d_1, \ldots, d_8)$, this system can be described by $\dot{x} = Ax + Bu$, where

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \end{bmatrix}.$$

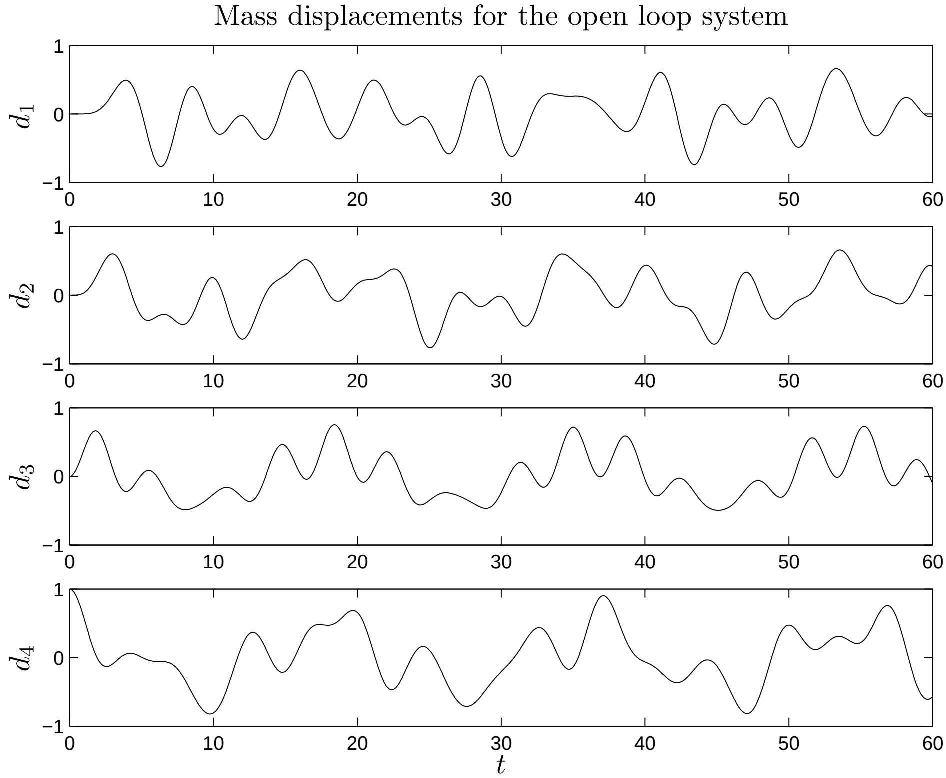
(b) Setting Q and R in the LQR cost function as

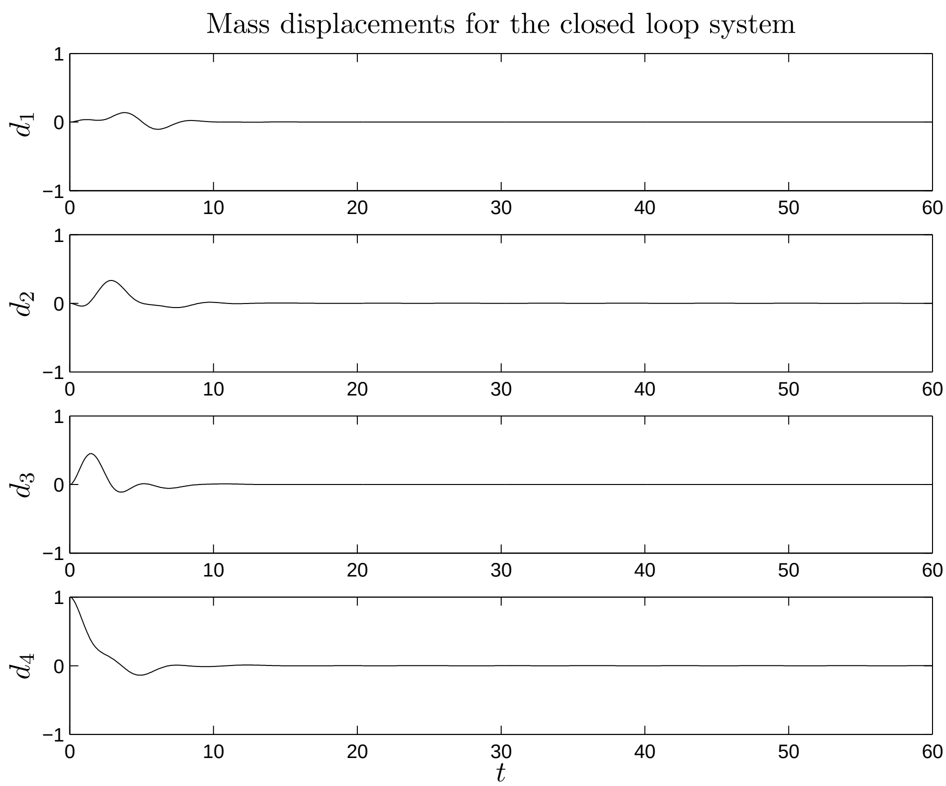
$$Q = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \qquad R = I$$

and using the Matlab function `-lqr()` (the Matlab function returns the negative of optimal gain matrix in our notation) we obtain

$$K = \begin{bmatrix} -0.1579 & 0.3393 & 0.0447 & 0.1948 & -0.3975 & 0.4758 & 0.5069 & 0.5787 \\ 0.1754 & 0.1755 & -0.1997 & 0.3923 & -0.0514 & 0.0203 & 0.0234 & 1.1090 \\ 0.1458 & 0.7680 & 0.2497 & 0.1537 & 0.6683 & 1.1442 & 0.8593 & 0.8796 \end{bmatrix}.$$

(c) Using an initial condition of $x(0) = (0, 0, 0, 1, 0, 0, 0, 0)$, the displacements $d_1, \ldots, d_4$ for the open loop and closed loop systems are shown below.





(d) K can be found from the Hamiltonian matrix by the following method:

- Form the 16×16 Hamiltonian matrix H.
- Find the eigenvalues of H.
- Find the eigenvectors corresponding to the 8 stable eigenvalues of H.
- Form the 16×8 matrix $M = [v_1 \cdots v_8]$ from the stable eigenvectors of H.
- Partition M into two 8×8 matrices $M^T = [X^T \; Y^T]$.
- $K = -R^{-1}B^TYX^{-1}$.

4. *Hamiltonian matrices.* A matrix

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$$

with $M_{ij} \in \mathbf{R}^{n \times n}$ is *Hamiltonian* if JM is symmetric, or equivalently

$$JMJ = M^T,$$

where

$$J = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}.$$

(a) Show that M is Hamiltonian if and only if $M_{22} = -M_{11}^T$ and M_{21} and M_{12} are each symmetric.

(b) Show that if v is an eigenvector of M, then Jv is an eigenvector of M^T.

(c) Show that if λ is an eigenvalue of M, then so is $-\lambda$.

(d) Show that $\det(sI - M) = \det(-sI - M)$, so that $\det(sI - M)$ is a polynomial in s^2.

Solution:

(a)

$$JMJ = M^T \Rightarrow \begin{bmatrix} M_{21} & M_{22} \\ -M_{11} & -M_{12} \end{bmatrix} \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix} = \begin{bmatrix} M_{11}^T & M_{21}^T \\ M_{12}^T & M_{22}^T \end{bmatrix}$$

or

$$\begin{bmatrix} -M_{22} & M_{21} \\ M_{12} & -M_{11} \end{bmatrix} = \begin{bmatrix} M_{11}^T & M_{21}^T \\ M_{12}^T & M_{22}^T \end{bmatrix}$$

and the claim follows.

(b)

$$J^{-1} = -J \implies Mv = \lambda v \implies JM^TJv = \lambda v \implies M^T(Jv) = -\lambda(Jv)$$

So λ is an eigenvalue and Jv is an eigenvector of M^T.

(c) It follows from (b), since the eigenvalues of J and J^T are identical.

(d) A little algebra:

$$\begin{aligned}\det(sI - M) &= \det(sI - M^T) = \det(sI - JMJ) = \det(sI + J^{-1}MJ)\\ &= \det(sI + M) = (-1)^{2n}\det(-sI - M) = \det(-sI - M)\end{aligned}$$

5. *Value function for infinite-horizon LQR problem.* In this problem you will show that the minimum cost-to-go starting in state z is a quadratic form in z. Let $Q = Q^T \geq 0$, $R = R^T > 0$ and define

$$J(u, z) = \sum_{t=0}^{\infty} \left(x_t^T Q x_t + u_t^T R u_t \right)$$

where $x_{t+1} = Ax_t + Bu_t$, $x_0 = z$. Note that u is a sequence in $\mathbf{R}^m$, and $z \in \mathbf{R}^n$. Of course, for some u's and z's, $J(u, z) = \infty$. Define

$$V(z) = \min_u J(u, z)$$

Note that this is a minimum over all possible input sequences. This is just like the previous problem, except that here u has infinite dimension. Assume (A, B) is controllable, so $V(z) < \infty$ for all z.

(a) Show that for all $\lambda \in \mathbf{R}$, $J(\lambda u, \lambda z) = \lambda^2 J(u, z)$, and conclude that

$$V(\lambda z) = \lambda^2 V(z). \tag{1}$$

(b) Let u and $\tilde{u}$ be two input sequences, and let z and $\tilde{z}$ be two initial states. Show that

$$J(u + \tilde{u}, z + \tilde{z}) + J(u - \tilde{u}, z - \tilde{z}) = 2J(u, z) + 2J(\tilde{u}, \tilde{z})$$

Minimize the RHS with respect to u and $\tilde{u}$, and conclude

$$V(z + \tilde{z}) + V(z - \tilde{z}) \leq 2V(z) + 2V(\tilde{z})$$

(c) Apply the above inequality with $\frac{1}{2}(z+\tilde{z})$ substituted for z and $\frac{1}{2}(z-\tilde{z})$ substituted for $\tilde{z}$ to get:

$$V(z + \tilde{z}) + V(z - \tilde{z}) = 2V(z) + 2V(\tilde{z}) \tag{2}$$

(d) The two properties (1) and (2) of V are enough to guarantee that V is a quadratic form. Here is one way to see it (you supply all details): take gradients in (2) with respect to z and $\tilde{z}$ and add to get

$$\nabla V(z + \tilde{z}) = \nabla V(z) + \nabla V(\tilde{z}) \tag{3}$$

From (1) show that

$$\nabla V(\lambda z) = \lambda \nabla V(z) \tag{4}$$

(3) and (4) mean that $\nabla V(z)$, which is a vector, is linear in z, and hence has a matrix representation:

$$\nabla V(z) = Mz \tag{5}$$

where $M \in \mathbf{R}^{n \times n}$.

(e) Show that $V(z) = z^T P z$, where $P = \frac{1}{4}(M + M^T)$. Show that $P = P^T \geq 0$. Thus we are done. *Hint*

$$V(z) = V(0) + \int_0^1 \nabla V(\theta z)^T z d\theta.$$

Solution:

(a) Let x_t be the state that corresponds to an input u_t and an initial condition z. Then

$$x_t = A^t z + \sum_{i=1}^{t} A^{i-1} B u_{t-i}.$$

Now denote by $x_{new,t}$ the state that corresponds to an input λu_t and an initial condition λz. Obviously,

$$x_{new,t} = A^t(\lambda z) + \sum_{i=1}^{t} A^{i-1} B(\lambda u_{t-i}) = \lambda x_t.$$

Thus,

$$J(\lambda u, \lambda z) = \sum_{t=0}^{\infty} (\lambda^2 x_t^T Q x_t + \lambda^2 u_t^T R u_t) = \lambda^2 J(u, z).$$

(b) Suppose

$$\begin{aligned} x_{t+1} &= A x_t + B u_t, \quad x_0 = z \\ \tilde{x}_{t+1} &= A \tilde{x}_t + B \tilde{u}_t, \quad \tilde{x}_0 = \tilde{z}. \end{aligned}$$

Adding or substracting the above equations yields

$$x_{t+1} \pm \tilde{x}_{t+1} = A(x_t \pm \tilde{x}_t) + B(u_t \pm \tilde{u}_t), \quad x_0 \pm \tilde{x} = z + \tilde{z}.$$

Thus, $x_t \pm \tilde{x}_t$ is the state that corresponds to an input $u_t \pm \tilde{u}_t$ and initial condition $z \pm \tilde{z}$, respectively. Now

$$\begin{aligned} & J(u + \tilde{u}, z + \tilde{z}) + J(u - \tilde{u}, z - \tilde{z}) \\ & = \sum_{t=0}^{\infty} (x_t + \tilde{x}_t)^T Q (x_t + \tilde{x}_t) + (x_t - \tilde{x}_t)^T Q (x_t - \tilde{x}_t) + \\ & (u_t + \tilde{u}_t)^T R (u_t + \tilde{u}_t) + (u_t - \tilde{u}_t)^T R (u_t - \tilde{u}_t) \\ & = \sum_{t=0}^{\infty} 2 x_t^T Q x_t + 2 \tilde{x}_t^T Q \tilde{x}_t + 2 u_t^T R u_t + 2 \tilde{u}_t^T R \tilde{u}_t = 2J(u, z) + 2J(\tilde{u}, \tilde{z}). \end{aligned}$$

So we have

$$J(u + \tilde{u}, z + \tilde{z}) + J(u - \tilde{u}, z - \tilde{z}) = 2J(u, z) + 2J(\tilde{u}, \tilde{z}).$$

Minimizing both sides with respect to u and $\tilde{u}$ obtains

$$\min_{u,\tilde{u}}\{J(u+\tilde{u}, z+\tilde{z}) + J(u-\tilde{u}, z-\tilde{z})\} = \min_{u} 2J(u,z) + \min_{\tilde{u}} 2J(\tilde{u},\tilde{z}).$$

The right-hand side of this equation is obviously $2V(z) + 2V(\tilde{z})$. Examining the left-hand side,

$$\begin{aligned}
&\min_{u,\tilde{u}}\{J(u+\tilde{u}, z+\tilde{z}) + J(u-\tilde{u}, z-\tilde{z})\} \\
&\quad \geq \min_{u,\tilde{u}} J(u+\tilde{u}, z+\tilde{z}) + \min_{u,\tilde{u}} J(u-\tilde{u}, z-\tilde{z}) \\
&\quad = V(z+\tilde{z}) + V(z-\tilde{z}).
\end{aligned}$$

Therefore,

$$V(z+\tilde{z}) + V(z-\tilde{z}) \leq 2V(z) + 2V(\tilde{z}).$$

(c) Performing this substitution yields

$$V(z) + V(\tilde{z}) \leq 2V(\frac{z+\tilde{z}}{2}) + 2V(\frac{z-\tilde{z}}{2}).$$

Using part (a), we have

$$V(z) + V(\tilde{z}) \leq \frac{1}{2}V(z+\tilde{z}) + \frac{1}{2}V(z-\tilde{z})$$

Combining this result with the one from part (b) yields

$$V(z+\tilde{z}) + V(z-\tilde{z}) = 2V(z) + 2V(\tilde{z}).$$

(d) Taking gradients of both sides of equation (1) in part (a), we obtain

$$\lambda \nabla V(\lambda z) = \lambda^2 \nabla V(z) \implies \nabla V(\lambda z) = \lambda \nabla V(z).$$

Thus $\nabla V(z)$ is linear in z, and hence $\exists M \in \mathbf{R}^{n\times n}$ such that $\nabla V(z) = Mz$.

(e)

$$V(z) = V(0) + \int_0^1 (\nabla V(\theta z))^T \, d(\theta z) = V(0) + \int (\nabla V(z))^T z\theta \, d\theta = V(0) + \frac{z^T M^T z}{2}.$$

Substituting $\lambda = 0$ in equation (1) in part (a) we obtain

$$V(0) = V(0\cdot z) = 0^2 \cdot V(z) = 0.$$

Thus,

$$V(z) = \frac{z^T M^T z}{2} = \frac{z^T M z}{2} = \frac{1}{2}\left(\frac{z^T M^T z}{2} + \frac{z^T M z}{2}\right) = \frac{z^T (M + M^T) z}{4}.$$

P is obviously symmetric; it is also positive semidefinite:

$$\min_u J(u,z) \geq 0 \;\; \forall z \implies V(z) = z^T P z \geq 0 \;\; \forall z \implies P \geq 0.$$

6. *Closed-loop stability for receding horizon LQR.* We consider the system $x_{t+1} = Ax_t + Bu_t$, $y_t = Cx_t$, with

$$A = \begin{bmatrix} 1 & 0.4 & 0 & 0 \\ -0.6 & 1 & 0.4 & 0 \\ 0 & 0.4 & 1 & -0.6 \\ 0 & 0 & 0.4 & 1 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \qquad C = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}.$$

Consider the receding horizon LQR control problem with horizon T, using state cost matrix $Q = C^T C$ and control cost $R = 1$. For each T, the receding horizon control has linear state feedback form $u_t = K_T x_t$. The associated closed-loop system has the form $x_{t+1} = (A + BK_T)x_t$. What is the smallest horizon T for which the closed-loop system is stable?

Interpretation. As you increase T, receding horizon control becomes less myopic and greedy; it takes into account the effects of current actions on the long-term system behavior. If the horizon is too short, the actions taken can result in an unstable closed-loop system.

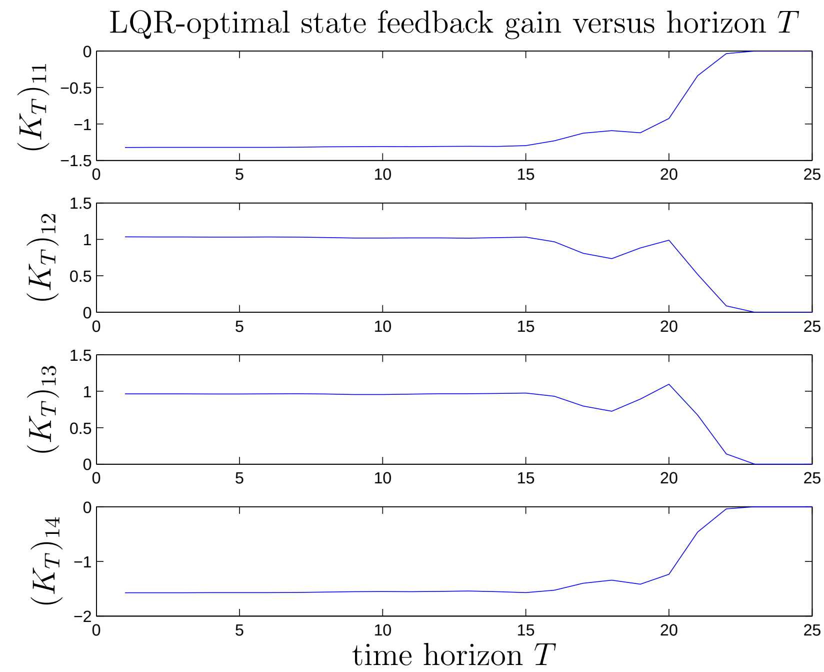
Solution: The closed-loop system $x_{t+1} = (A + BK_T)x_t$ is stable if the spectral radius of $A + BK_T$ is strictly less than 1. (The spectral radius of a square matrix P is $\max_{i=1,\ldots,n} |\lambda_i|$, where $\lambda_1 \ldots \lambda_n$ are its eigenvalues.)

In order to determine the smallest horizon T that gives a stable closed-loop system, we will find K_T for $T = 1, 2 \ldots$ and check whether the spectral radius of $A + BK_T$ is less than 1. We have

$$K_T = -(R + B^T P_T B)^{-1} B^T P_T A,$$

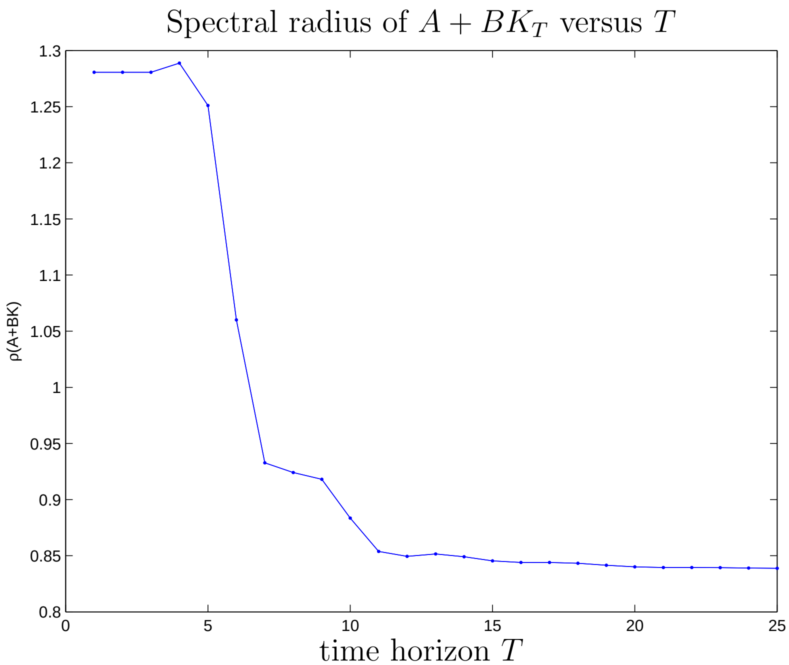
where $P_1 = Q$, $P_{i+1} = Q + A^T P_i - A^T P_i B(R + B^T PB)^{-1} B^T P_i A$.

A simple Matlab script which performs the method described is shown below. The first plot shows the optimal state feedback gains versus the receding time horizon T. The second plot shows the spectral radius of $A + BK_T$ as a function of T. We can see from the second plot that for $T \geq 7$, the resulting closed-loop system is stable.




```
% closed-loop stability for receding horizon LQR
% data
A = [  1 .4  0   0; -.6  1 .4   0;0 .4  1 -.6;0  0 .4   1];
B = [1 0 0 0]';
C = [0 0 0 1];
Q = C'*C;R = 1;
m=1;n=4;T=25;

% recursion
P=zeros(n,n,T+1);
K=zeros(m,n,T);
P(:,:,T+1)=Q;
spec_rad = zeros(1,T);

for i = T:-1:1
    % LQR-optimal state feedback
    K(:,:,i) = -inv(R + B'*P(:,:,i+1)*B)*B'*P(:,:,i+1)*A;
    % spectral radius
    spec_rad(T-i+1) = max(abs(eig(A+B*K(:,:,i))));
    P(:,:,i)=Q + A'*P(:,:,i+1)*A + A'*P(:,:,i+1)*B*K(:,:,i);
end

% plots
figure(1);
t = 0:T-1; K = shiftdim(K);
```

```
subplot(4,1,1); plot(t,K(1,:)); ylabel('K1(t)');
subplot(4,1,2); plot(t,K(2,:)); ylabel('K2(t)');
subplot(4,1,3); plot(t,K(3,:)); ylabel('K3(t)');
subplot(4,1,4); plot(t,K(4,:)); ylabel('K4(t)'); xlabel('t');
% print -depsc receding_horizon_gains.eps

figure(2)
plot(1:T,spec_rad); hold on; plot(1:T,spec_rad,'.')
xlabel('T'); ylabel('\rho(A+BK)')
title('title')
% print -depsc receding_horizon_specrad.eps
```



7. *LQR with exponential weighting.* A common variation on the LQR problem includes explicit time-varying weighting factors on the state and input costs,

$$J = \sum_{\tau=0}^{N-1} \gamma^\tau \left(x_\tau^T Q x_\tau + u_\tau^T R u_\tau \right) + \gamma^N x_N^T Q_f x_N,$$

where $x_{t+1} = Ax_t + Bu_t$, x_0 is given, and, as usual, we assume $Q = Q^T \geq 0$, $Q_f = Q_f^T \geq 0$ and $R = R^T > 0$ are constant. The parameter γ, called the exponential weighting factor, is positive. For $\gamma = 1$, this reduces to the standard LQR cost function. For $\gamma < 1$, the penalty for future state and input deviations is smaller than in the present; in this case we call γ the *discount factor* or *forgetting factor*. When $\gamma > 1$, future costs are accentuated compared to present costs. This gives added incentive for the input to steer the state towards zero quickly.

(a) Note that we can find the input sequence $u_0^*, \ldots, u_{N-1}^*$ that minimizes J using standard LQR methods, by considering the state and input costs as time varying, with $Q_t = \gamma^t Q$, $R_t = \gamma^t R$, and final cost given by $\gamma^N Q_f$. Thus, we know at least one way to solve the exponentially weighted LQR problem. Use this method to find the recursive equations that give u^*.

(b) Exponential weights can also be incorporated directly into a dynamic programming formulation. We define

$$W_t(z) = \min \sum_{\tau=t}^{N-1} \gamma^{\tau-t} \left(x_\tau^T Q x_\tau + u_\tau^T R u_\tau \right) + \gamma^{N-t} x_N^T Q_f x_N,$$

where $x_t = z$, $x_{\tau+1} = Ax_\tau + Bu_\tau$, and the minimum is over $u_t, \ldots, u_{N-1}$. This is the minimum cost-to-go, if we started in state z at time t, with the time weighting also starting at t. Argue that we have

$$W_N(z) = x_N^T Q_f x_N,$$

$$W_t(z) = \min_w \left(z^T Q z + w^T R w + \gamma W_{t+1}(Az + Bw) \right),$$

and that the minimizing w is in fact u_t^*. In other words, work out a backwards recursion for W_t, and give an expression for u_t^* in terms of W_t. Show that this method yields the same u^* as the first method.

(c) Yet another method can be used to find u^*. Define a new system as

$$y_{t+1} = \gamma^{1/2} A y_t + \gamma^{1/2} B z_t, \qquad y_0 = x_0.$$

Argue that we have $y_t = \gamma^{t/2} x_t$, provided $z_t = \gamma^{t/2} u_t$, for $t = 0, \ldots, N-1$. With this choice of z, the exponentially weighted LQR cost J for the original system is given by

$$J = \sum_{\tau=0}^{N-1} \left(y_\tau^T Q y_\tau + z_\tau^T R z_\tau \right) + y_N^T Q_f y_N,$$

i.e., the unweighted LQR cost for the modified system. We can use the standard formulas to obtain the optimal input for the modified system z^*, and from this, we can get u^*. Do this, and verify that once again, you get the same u^*.

Solution:

(a) The value function is

$$V_t(z) = \min \left(\sum_{\tau=t}^{N-1} \gamma^\tau \left(x_\tau^T Q x_\tau + u_\tau^T R u_\tau \right) + x_N^T \gamma^N Q_f x_N \right),$$

where the minimum is over all input sequences $u(0), \ldots, u(N-1)$, and $x_t = z$, $x_{\tau+1} = Ax_\tau + Bu_\tau$ for $\tau = t, \ldots, N-1$. We can express this as

$$V_t(z) = \min \sum_{\tau=t}^{N} \left(x_\tau^T Q_\tau x_\tau + u_\tau^T R_\tau u_\tau \right) + x_N^T \gamma^N Q_f x_N,$$

where $Q_\tau = \gamma^\tau Q$ and $R_\tau = \gamma^\tau R$. This is just a standard LQR problem with time-varying Q and R. Thus from the lecture notes we have

$$V_t(z) = z^T P_t z, \qquad u_t^* = K_t x_t,$$

where P_t and K_t are given by the following recursion

$$\begin{aligned}
P_N &= \gamma^N Q_f, \\
P_{t-1} &= Q_{t-1} + A^T P_t A - A^T P_t B \left(R_{t-1} + B^T P_t B\right)^{-1} B^T P_t A \\
&= \gamma^{t-1} Q + A^T P_t A - A^T P_t B \left(\gamma^{t-1} R + B^T P_t B\right)^{-1} B^T P_t A, \\
K_t &= -\left(R_t + B^T P_{t+1} B\right)^{-1} B^T P_{t+1} A \\
&= -\left(\gamma^t R + B^T P_{t+1} B\right)^{-1} B^T P_{t+1} A.
\end{aligned}$$

(b) Clearly we have $W_N(z) = z^T Q_f z$, since at the last step the cost is not a function of the input.

Now let $u_t = w$. We can apply the dynamic programming principle to the equation for $W_t(z)$ by first minimizing with respect to w and then with respect to the other inputs. If we do that, it is easy to see that

$$W_t(z) = \min_w \left(z^T Q z + w^T R w + \gamma W_{t+1}(Az + Bw)\right).$$

Note that the minimizing w is equal to the optimal input u_t^*. This is because $W_0(z) = V_0(z)$ for all z and therefore a set of minimizing inputs for the above HJ equation will also minimize J.

We will now show by induction that $W_t(z) = z^T S_t z$, *i.e.* that $W_t(z)$ is a quadratic form in z. Let us assume that this is true for $t = \tau + 1$. Then using the above HJ equation we have

$$W_\tau(z) = \min_w \left(z^T Q z + w^T R w + \gamma (Az + Bw)^T S_{\tau+1} (Az + Bw)\right).$$

To find w^*, we take the derivatives with respect to w of the argument within the minimum above, and set it equal to zero. We get

$$w^* = -\gamma (R + \gamma B^T S_{\tau+1} B)^{-1} B^T S_{\tau+1} A z.$$

By substituting the optimal w in the expression for $W_t(z)$ we obtain

$$W_t(z) = z^T \left(Q + \gamma A^T S_{t+1} A - \gamma^2 A^T S_{t+1} B (R + \gamma B^T S_{t+1} B)^{-1} B^T S_{t+1} A\right) z,$$

which is still a quadratic form. This combined with the fact that $W_N(z) = z^T Q_f z$ is sufficient to see that $W_t(z)$ is indeed a quadratic form. Also, the optimum input u_t^* is given by

$$u_t^* = -\gamma (R + \gamma B^T S_{\tau+1} B)^{-1} B^T S_{\tau+1} A x_t.$$

Actually S_t and P_t are closely related, since we have

$$\begin{aligned} W_t(z) &= \min\left(\sum_{\tau=t}^{N-1} \gamma^{\tau-t}\left(x_\tau^T Q x_\tau + u_\tau^T R u_\tau\right) + x_N^T Q_f x_N\right) \\ &= \gamma^{-t}\min\left(\sum_{\tau=t}^{N-1} \gamma^{\tau}\left(x_\tau^T Q x_\tau + u_\tau^T R u_\tau\right) + x_N^T Q_f x_N\right) \\ &= \gamma^{-t} V_t(z) \end{aligned}$$

Thus

$$z^T S_t z = \gamma^{-t} z^T P_t z, \quad \forall z,$$

or in other words $S_t = \gamma^{-t} P_t$. Substituting this into the formula for u_t^* we get

$$\begin{aligned} u_t^* &= -\left(R + B^T(\gamma^{-t}P_{t+1})B\right)^{-1} B^T A(\gamma^{-t}P_{t+1})x_t \\ &= -\left(\gamma^t R + B^T P_{t+1} B\right)^{-1} B^T A P_{t+1} x_t. \end{aligned}$$

Therefore this method gives the same formula for u_t^* as the one of part (a).

(c) We want to show that $y_t = \gamma^{t/2} x_t$, provided that $z_t = \gamma^{t/2} u_t$. This can be shown by induction. The argument holds for $t = 0$, since $y_0 = x_0$. Now suppose that $y_t = \gamma^{t/2} x_t$. We have

$$\begin{aligned} y_{t+1} &= \gamma^{1/2} A y_t + \gamma^{1/2} B z_t \\ &= \gamma^{(t+1)/2} A x_t + \gamma^{(t+1)/2} B u_t \\ &= \gamma^{(t+1)/2}\left(A x_t + B u_t\right) \\ &= \gamma^{(t+1)/2} x_{t+1}. \end{aligned}$$

Thus the argument also holds for y_{t+1}. Now, let's express the cost function in terms of y_t and z_t. We have

$$\begin{aligned} J &= \sum_{\tau=0}^{N-1} \gamma^\tau \left(x_\tau^T Q x_\tau + u_\tau^T R u_\tau\right) + \gamma^N x_N^T Q_f x_N \\ &= \sum_{\tau=0}^{N-1} \left((\gamma^{\tau/2} x_\tau^T) Q (\gamma^{\tau/2} x_\tau) + (\gamma^{\tau/2} u_\tau^T) R (\gamma^{\tau/2} u_\tau)\right) + (\gamma^{N/2} x_N^T) Q_f (\gamma^{N/2} x_N) \\ &= \sum_{\tau=0}^{N} \left(y_\tau^T Q y_\tau) + z_\tau^T R z_\tau\right) + y_N^T Q_f y_N. \end{aligned}$$

Using the formulas from the lecture notes we get that the minimizing z_t for the above expression is given by

$$z_t^* = -\gamma(R + \gamma B^T \tilde{P}_{t+1} B)^{-1} B^T \tilde{P}_{t+1} A y_t,$$

where $\tilde{P}_t$ is found by the following recursion

$$\begin{aligned}
\tilde{P}_N &= Q_f, \\
\tilde{P}_{t-1} &= Q + \gamma A^T \tilde{P}_t A - \gamma^2 A^T \tilde{P}_t B (R + \gamma B^T \tilde{P}_t B)^{-1} B^T \tilde{P}_t A.
\end{aligned}$$

Note that since the original problem and this modified problem have the same cost function, the optimum z_t^* will give us the optimum u_t^* of the original problem, using the relation $z_t^* = \gamma^{t/2} u_t^*$. We can also show by an inductive argument that $P_t = \gamma^t \tilde{P}_t$. Combining these two relations in the formula for z_t^*, we get

$$\begin{aligned}
\gamma^{t/2} u_t^* &= -\gamma^{-t}(R + \gamma^{-t} B^T P_{t+1} B)^{-1} B^T P_{t+1} A y_t \\
&= -\gamma^{t/2}(\gamma^t R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A x_t
\end{aligned}$$

We therefore get

$$u_t^* = -\left(\gamma^t R + B^T P_{t+1} B\right)^{-1} B^T A P_{t+1} x_t,$$

which is the same expression as the one obtained in parts (a) and (b).