HW3: 2.2-2, 4, 7

2.2-2 Find all possible solutions to (2.2-26):

(2.2-26) $\qquad S = A^T S A + Q$

Where $\qquad A = \begin{bmatrix} \frac{1}{2} & 1 \\ 0 & -\frac{1}{2} \end{bmatrix} \qquad C = \begin{bmatrix} 2 & 0 \end{bmatrix} \qquad Q = C^T C$

Let $\quad P = \begin{bmatrix} P_1 & P_2 \\ P_3 & P_4 \end{bmatrix}$, thus (2.2-26) becomes:

$$\begin{bmatrix} P_1 & P_2 \\ P_3 & P_4 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 \\ 1 & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} P_1 & P_2 \\ P_3 & P_4 \end{bmatrix} \begin{bmatrix} \frac{1}{2} & 1 \\ 0 & -\frac{1}{2} \end{bmatrix} + \begin{bmatrix} 2 \\ 0 \end{bmatrix} \begin{bmatrix} 2 & 0 \end{bmatrix}$$

$$\begin{bmatrix} P_1 & P_2 \\ P_3 & P_4 \end{bmatrix} = \begin{bmatrix} \frac{P_1}{2} & \frac{P_2}{2} \\ P_1 - \frac{P_3}{2} & P_2 - \frac{P_4}{2} \end{bmatrix} \begin{bmatrix} \frac{1}{2} & 1 \\ 0 & -\frac{1}{2} \end{bmatrix} + \begin{bmatrix} 4 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} P_1 & P_2 \\ P_3 & P_4 \end{bmatrix} = \begin{bmatrix} \frac{P_1}{4} + 4 & \frac{P_1}{2} - \frac{P_2}{4} \\ \frac{P_1}{2} - \frac{P_3}{4} & \left(P_1 - \frac{P_3}{2}\right) - \left(\frac{P_2}{2} - \frac{P_4}{4}\right) \end{bmatrix}$$

Solving simultaneously:

$P_1 = \frac{P_1}{4} + 4$	$P_2 = \frac{P_1}{2} - \frac{P_2}{4} = \frac{8}{3} - \frac{P_2}{4}$	$P_3 = \frac{P_1}{2} - \frac{P_3}{4} = \frac{8}{3} - \frac{P_3}{4}$
$\frac{3P_1}{4} = 4$	$\frac{5P_2}{4} = \frac{8}{3}$	$\boxed{P_3 = P_2 = \frac{32}{15}}$ $\rightsquigarrow$ symmetric
$\boxed{P_1 = \frac{16}{3}}$	$\boxed{P_2 = \frac{32}{15}}$	

$P_4 = P_1 - \frac{P_3}{2} - \overset{= P_2}{\frac{P_2}{2}} + \frac{P_4}{4} = \frac{16}{3} - \frac{32}{15} + \frac{P_4}{4} = \frac{80 - 32}{15} + \frac{P_4}{4}$

$\frac{3P_4}{4} = \frac{48}{15} = \frac{16}{5}$

$P_4 = \frac{64}{15}$

a. Thus $P = \begin{bmatrix} \frac{16}{3} & \frac{32}{15} \\ \frac{32}{15} & \frac{64}{15} \end{bmatrix}$

b. Since $|eig(A)| < 1$, A is stable, and since the observability matrix is,

$$\begin{bmatrix} C \\ CA \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 1 & 2 \end{bmatrix},$$

which has full rank, the solution to the Riccati equation has the unique solution found in part a.

2.2-4 Let $x_{k+1} = 2x_k + u_k$

a. find the homogeneous solution x_k for $k = 0, 5$ if $x_0 = 3$

homogeneous solution $\Rightarrow u_k = 0 \ \forall k$

$\Rightarrow x_{k+1} = 2x_k$

K	x_k	x_{k+1}
0	3	6
1	6	12
2	12	24
3	24	48
4	48	96
5	96	192

b. Minimum-energy control to drive state from $x_0 = 3$ to $x_5 = 0$

cost function: $J_0 = \frac{1}{2} \sum_{k=0}^{5} r u_k^2$

Thus the Hamiltonian is:

$$H^k = \frac{1}{2} r u_k^2 + \lambda_{k+1} (2x_k + u_k)$$

From this we get the state & costate equations:

state: $x_{k+1} = \dfrac{\partial H^k}{\partial \lambda_{k+1}} = 2x_k + u_k$

costate: $\lambda_k = \dfrac{\partial H^k}{\partial x_k} = 2\lambda_{k+1}$

Also have the stationarity condition:

$$0 = \frac{\partial H^k}{\partial u_k} = r u_k + \lambda_{k+1} \quad \Rightarrow \quad u_k = -\frac{\lambda_{k+1}}{r}$$

Substituting this u_k into the state equation:

$$x_{k+1} = 2x_k - \frac{\lambda_{k+1}}{r}$$

b)

Next we write λ_k in terms of λ_N:

$$\lambda_{N-1} = 2 \cdot \lambda_N$$

$$\lambda_{N-2} = 2^2 \lambda_N$$

$$\vdots$$

$$\lambda_k = 2^{N-k} \lambda_N = 2^{5-k} \lambda_5$$

Thus $\quad \lambda_{k+1} = 2^{N-(k+1)} \lambda_N = 2^{N-k-1} \lambda_N = 2^{4-k} \lambda_5$

Substituting into state equation:

$$X_{k+1} = 2X_k - \frac{2^{4-k}\lambda_5}{r} \quad (1^{st} \text{ order difference equation})$$

$$\Rightarrow \quad X_k = 2^k X_0 - \sum_{i=0}^{k-1} 2^{k-i-1} \cdot \frac{1}{r} \cdot 2^{N-i-1} \cdot \lambda_N$$

$$X_k = 2^k X_0 - \frac{\lambda_N}{r} \sum_{i=0}^{k-1} 2^{N+k-2i-2}$$

Setting $k = N = 5$, we can solve for $\lambda_N = \lambda_5$

$$X_5 = 2^5 (3) - \frac{\lambda_5}{r} \sum_{i=0}^{4} 2^{5+5-2i-2} = 2^5 \cdot 3 - \frac{\lambda_5}{r} \sum_{i=0}^{4} 2^{8-2i} = 96 - \frac{\lambda_5}{r} \cdot 341 = 0$$

$$\Rightarrow \lambda_5 = \frac{96}{341} \cdot r$$

Thus, $\quad \lambda_{k+1} = 2^{4-k} \cdot \frac{96}{341} r$

$$\Rightarrow \quad U_k = -\frac{\lambda_{k+1}}{r} = -\frac{2^{4-k} \cdot \frac{96}{341} \cdot r}{r} = -2^{4-k} \frac{96}{341}$$

K	U_k	X_k
0	$-2^4 \cdot \frac{96}{341} = 4.5044$	3
1	$-2^3 \frac{96}{341} = 2.522$	1.4956
2	1.1261	0.7390
3	0.5630	0.3519
4	0.2815	0.1408
5	—	$-9.38 \times 10^{-15} \sim 0$ → rounding error from MATLAB

c. optimal K_k to minimize $J_0 = 5x_5^2 + \frac{1}{2}\sum_{k=0}^{4}(x_k^2 + u_k^2)$

The key is finding the relation:

$$\frac{1}{2}S_N x_N^2$$

$$\lambda_N = S_N x_N , \quad S_N = 10$$

To do this we let $\lambda_k = S_k x_k \quad \forall k = 0, \dots, N$

To find the sequence of S_k, we solve the Riccati equation backwards:

$$S_k = \frac{a^2 r S_{k+1}}{b^2 S_{k+1} + r} + q$$

This can then be used to find the optimal gains K_k:

$$K_k = \frac{ab S_{k+1}}{b^2 S_{k+1} + r}$$

For our system, $a = 2$
$r = 1 \implies$
$q = 1$
$b = 1$

K	S_k	K_k
5	10	—
4	$\frac{4 \cdot 1 \cdot 10}{1 \cdot 10 + 1} + 1 = 4.6364$	$\frac{2 \cdot 10}{10 + 1} = 1.818$
3	4.2903	1.6452
2	4.2439	1.6220
1	4.2372	1.6186
0	4.2362	1.6181

With these gains, we can get the trajectory and cost:

$$u_k^* = -K_k x_k$$

$$x_{k+1} = 2x_k - K_k x_k = (2 - k_k)x_k \quad , \quad J_k^* = \frac{1}{2}S_k x_k^2$$

K	x_k	J_k^*
0	3	19.0631
1	0.5455	4.74
2	0.1935	1.16
3	0.0732	0.27
4	0.0279	0.046
5	0.0107	$4.4 \times 10^{-28} \sim 0$

2.2-7   Let $x_{k+1} = \overbrace{a x_k + b u_k}^{f^k}$ (scalar system)

$$J = \frac{1}{3} \underbrace{S_N x_N^3}_{\emptyset} + \frac{1}{3} \sum_{k=0}^{N-1} \underbrace{(q x_k^3 + r u_k^3)}_{L^k}$$

a.) Defining the Hamiltonian:

$$H^k = L^k + \lambda^{k+1}_{k+1} f^k = \frac{1}{3}(q_k x_k^3 + r u_k^3) + \lambda_{k+1}(a x_k + b u_k)$$

<div style="border:1px solid">

State equation : $x_{k+1} = \dfrac{\partial H^k}{\partial \lambda_{k+1}} = a x_k + b u_k$

Costate equation: $\lambda_k = \dfrac{\partial H^k}{\partial x_k} = x_k^2 \cdot q + a \lambda_{k+1}$

Stationarity condition: $0 = \dfrac{\partial H^k}{\partial u_k} = r u_k^2 + b \lambda_{k+1}$

</div>

b.   Solving the stationarity condition for u_k:

$$r u_k^2 = -b \lambda_{k+1}$$

$$u_k = \sqrt{-\frac{b}{r} \lambda_{k+1}}$$

$\Rightarrow$ Since we assume $r > 0$, $\lambda_k > 0$, b must be negative to solve for u_k

Substituting u_k into state (assuming $b < 0$)

$$\boxed{x_{k+1} = a x_k + b \sqrt{-\frac{b}{r} \lambda_{k+1}}}$$

C. Solve open-loop problem $(dx_N = 0, \; S_N = 0, \; q_p = 0)$

The performance index becomes:

$$J = \tfrac{1}{3} \sum_{k=0}^{N-1} r u_k^3$$

and the Hamiltonian:

$$H^k = \tfrac{1}{3} r u_k^3 + \lambda_{k+1}(a x_k + b u_k)$$

$\Rightarrow \qquad x_{k+1} = a x_k + b u_k \quad \text{—state}$

$$\lambda_k = \frac{\partial H^k}{\partial x_k} = a \lambda_{k+1} \quad \text{— costate}$$

$$0 = \frac{\partial H^k}{\partial u_k} = r u_k^2 + b \lambda_{k+1} \quad \text{— stationarity}$$

From part b. : $\quad x_{k+1} = a x_k + b\sqrt{-\tfrac{b}{r}\lambda_{k+1}}$

Since the costate is a difference equation, we can write

$$\lambda_k = a^{N-k} \lambda_N$$

$\Rightarrow \qquad x_{k+1} = a x_k + b\sqrt{\left(\tfrac{-b}{r}\right) \underbrace{a^{N-k-1}\lambda_N}_{f}}$

This new state equation is also a difference equation, with forcing function f.

$\Rightarrow \quad x_k = a^k x_0 + \sum_{i=0}^{k-1} a^{k-i-1} \left(b\sqrt{\left(\tfrac{-b}{r}\right)\left(a^{N-k-i}\cdot\lambda_N\right)} \right)$

$$= a^k x_0 + b\cdot\sqrt{\tfrac{-b}{r}}\cdot\sqrt{\lambda_N} \sum_{i=0}^{k-1} \underbrace{a^{k-i-1}\cdot a^{(N-k-i)/2}}_{}$$

$$a^{\frac{N+k-3i-2}{2}} = a^{\frac{N+k-2}{2}}\cdot a^{\frac{-3i}{2}}$$

$\Rightarrow \quad x_k = a^k x_0 + b\sqrt{-\tfrac{b}{r}\cdot\lambda_N\cdot a^{N+k-2}}\cdot \left. \sum_{i=0}^{k-1}\left(\tfrac{1}{a^{3/2}}\right)^i \right\} \text{geometric series}$

$$= a^k x_0 + b\sqrt{\tfrac{-b}{r}\lambda_N a^{N+k-2}}\cdot \frac{1 - a^{-\frac{3}{2}k}}{1 - a^{-3/2}} \quad \longrightarrow \text{only unknown is } \lambda_N$$

2.2-7 cont'd Let $k = N$

$$X_N = a^N x_0 + b\sqrt{\frac{-b}{r}\lambda_N a^{N+N-2}} \cdot \frac{1 - a^{-\frac{3}{2}N}}{1 - a^{-3/2}}$$

$$\frac{X_N - a^N x_0}{b} = \left(-\frac{b}{r}\lambda_N\right)^{1/2}\left(a^{2N-2}\right)^{1/2} \frac{1 - a^{-\frac{3N}{2}}}{1 - a^{-\frac{3}{2}}}$$

$$\underbrace{\frac{a^{2N/2}}{a^{2/2}} \cdot \frac{1 - a^{-3N/2}}{1 - a^{-3/2}}} = \frac{a^N - a^{-N/2}}{a - a^{-1/2}}.$$

$$\Rightarrow \left(-\frac{b}{r}\lambda_N\right)^{1/2} = \left[\frac{X_N - a^N x_0}{b}\right]\left[\frac{a - a^{-1/2}}{a^N - a^{-N/2}}\right]$$

$$\lambda_N = -\frac{r}{b}\left[\frac{X_N - a^N x_0}{b}\right]^2 \left[\frac{a - a^{1/2}}{a^N - a^{-N/2}}\right]^2$$

Thus the costate for arbitrary $k \leq N$

$$\lambda_k = -a^{N-k} \cdot \frac{r}{b} \cdot \left[\frac{X_N - a^N x_0}{b}\right]^2 \cdot \left[\frac{a - a^{1/2}}{a^N - a^{-N/2}}\right]^2$$

Then the optimal control sequence is:

$$u_k^* = \sqrt{-\frac{b}{r}\lambda_{k+1}} = \left\{ +\frac{b}{r}\cdot\frac{r}{b}\, a^{N-k-1} \cdot \left[\frac{X_N - a^N x_0}{b}\right]^2 \left[\frac{a - a^{1/2}}{a^N - a^{-N/2}}\right]^2 \right\}^{\frac{1}{2}}$$

$$\boxed{u_k^* = a^{\frac{N-k-1}{2}} \cdot \frac{X_N - a^N x_0}{b} \cdot \frac{a - a^{1/2}}{a^N - a^{-N/2}}}$$

Piero Risi Mortola
Optimal Control Theory
Dr. Zheng Chen

<p style="text-align:center">Homework 3 - LQR controllers</p>

The circuit from example 2.3-1 from the textbook was modeled in MATLAB using two LQR controller: a fixed final state controller and a free final state controller. The state trajectories and the optimal control sequence are shown below.

The MATLAB code is pasted at the end.

Optimal Control Sequence (fixed final state)

State Trajectory (fixed final state)

Optimal Control Sequence (free final state)

State Trajectory (free final state)

Final state = 11.59

Step (k)
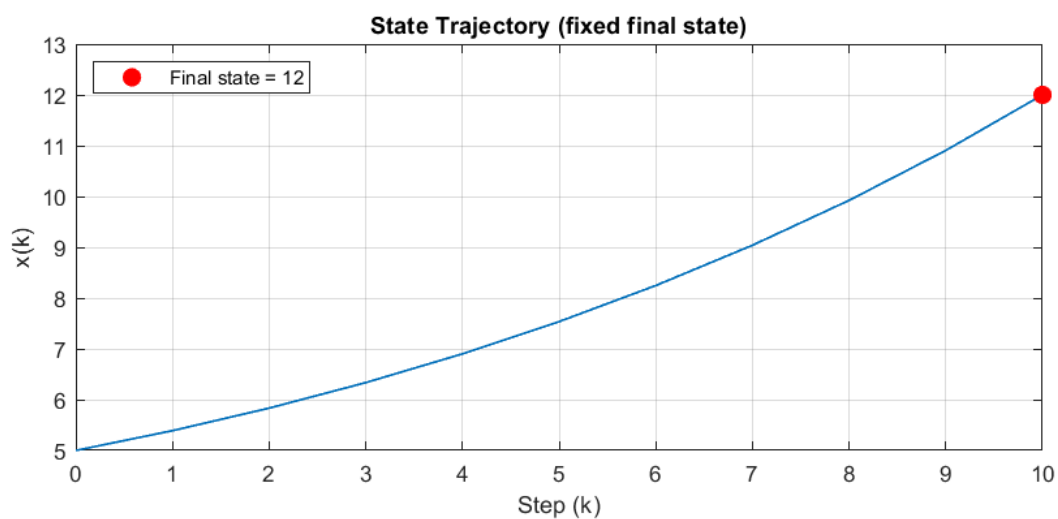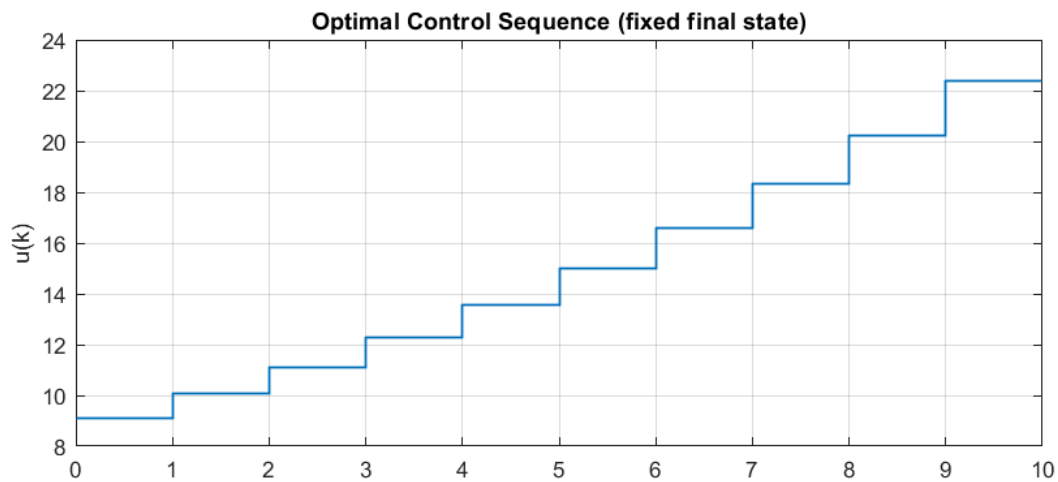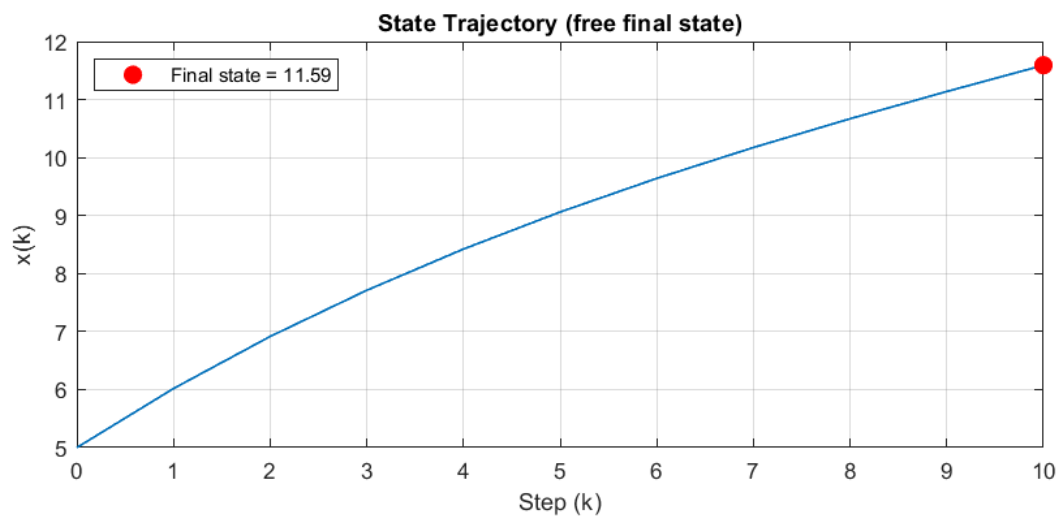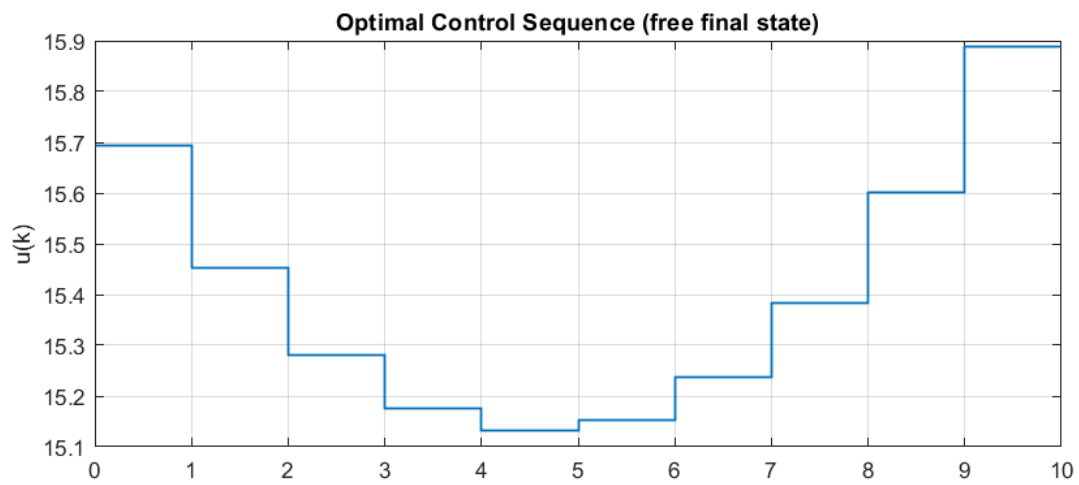
```matlab
clear all
clc
close all
% Parameters
a = 0.9048;
b = 0.0952;
r = 1;
q = 1;
sN = 100;
N = 10; % number of steps
S(1) = sN;
rN = 12; % target
% Free final state: adjust system to drive towards rN=12 not zero
x(1) = 5;
x_ref = rN; % xN = a*xN+b*u_ref
u_ref = (1-a)*x_ref/b;
% solving Riccati equation backwards in time
for k = 1:N
    S(k+1) = (a^2)*(r*S(k))/((b^2)*S(k)+r)+q;
    K(k) = a*b*S(k)/((b^2)*S(k)+r);
end
% applying the control to the system
for k=1:N
    u(k) = -K(end-k+1)*(x(k)-x_ref)+u_ref;
    x(k+1) = a*x(k)+b*u(k);
end
%% Fixed final state - becomes open loop control
xf(1) = 5;
for k = 0:N-1
    uf(k+1) = ((1 - a^2)/(b*(1 - a^(2*N))))*(rN - a^N * xf(1))*a^(N-1-k);
    xf(k+2) = a*xf(k+1)+b*uf(k+1);
end
%% Plotting the trajectory and control signal
% Free final state
figure
subplot(2,1,1)
stairs(0:N,[u u(end)],'LineWidth',1)
ylabel("u(k)")
title("Optimal Control Sequence (free final state)")
grid on
subplot(2,1,2)
plot(0:N,x,'LineWidth',1)
hold on
plot(N, x(end), 'ro', 'MarkerSize', 8, 'MarkerFaceColor','r');
```

```matlab
xlabel("Step (k)")
ylabel("x(k)")
title("State Trajectory (free final state)")
grid on
legend('',sprintf('Final state = %.2f', x(end)),'Location','northwest')
% Fixed final state
figure
subplot(2,1,1)
stairs(0:N,[uf uf(end)],'LineWidth',1)
ylabel("u(k)")
title("Optimal Control Sequence (fixed final state)")
grid on
subplot(2,1,2)
plot(0:N,xf,'LineWidth',1)
hold on
plot(N, xf(end), 'ro', 'MarkerSize', 8, 'MarkerFaceColor','r');
xlabel("Step (k)")
ylabel("x(k)")
title("State Trajectory (fixed final state)")
grid on
legend('',sprintf('Final state = %.0f', xf(end)),'Location','northwest')
```