ECE 147C
Solutions, Problem 9.2

Contents

Consider an inverted pendulum operating near the upright equilibrium position, with linearized model given by

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{g}{l} & -\frac{b}{ml^2} \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{ml^2} \end{bmatrix} u$$

where u denotes an applied torque, the output $y = \theta$ is the pendulum's angle with a vertical axis pointing up, and $l = 1\,m$, $m = 1\,Kg$, $b = .1N/m/s$, $g = 9.8m/s^2$.

```
l = 1; m = 1; b = 0.1; g = 9.8;
```

Design a PD and PID controller using LQR. For both controllers provide the Bode plot of the open-loop gain and the closed-loop step response.

Hint: Consider an "augmented" process model with state θ, $\dot{\theta}$, $z(t) := \int_0^t \theta(s)ds$.

```
close all;
clearvars -EXCEPT l m b g
clc;
```

Part 1: PD design

Design a PD controller using LQR.

The given system is of the form

$$\dot{x} = Ax + Bu$$

where

$x = [\theta; \dot{\theta}], \qquad u = T$

and

$$A := \begin{bmatrix} 0 & 1 \\ \frac{g}{l} & -\frac{b}{ml^2} \end{bmatrix}$$

```
A = [0 1; g/l -b/(m*l^2)];
```

$$B := \begin{bmatrix} 0 \\ \frac{1}{ml^2} \end{bmatrix}$$

```
B = [0; 1/(m*l^2)];
```

In LQR, one assumes that all states are measured, i.e, $y = x$, i.e, $C = I$. Hence we have

```
C = eye(2); D = [0; 0];
```

Decide G and H. Let's pick the controlled output $z(t)$ to be

$z(t) := \theta(t),$

so that we have

$$G := \begin{bmatrix} 1 & 0 \end{bmatrix}, \qquad H = 0$$

```
G = [1 0]; H = 0;
```

Trial-and-error to figure out the parameter p. This is the trade-off between actuator useage and controller performance.

```
p = 10;
```

The formulae for matrices QQ, RR, and NN come from the book.

```
QQ = G'*G; RR = H'*H + p*1; NN = G'*H;
```

State-Feedback LQR controller (Optimal):

```
[K,S,E]=lqr(A,B,QQ,RR,NN);
K
```

```
K =

   19.6051    6.1626
```

Since the control law is $u = -Kx = -K_1\theta - K_2\dot{\theta}$, the PD controller's proportional gain is

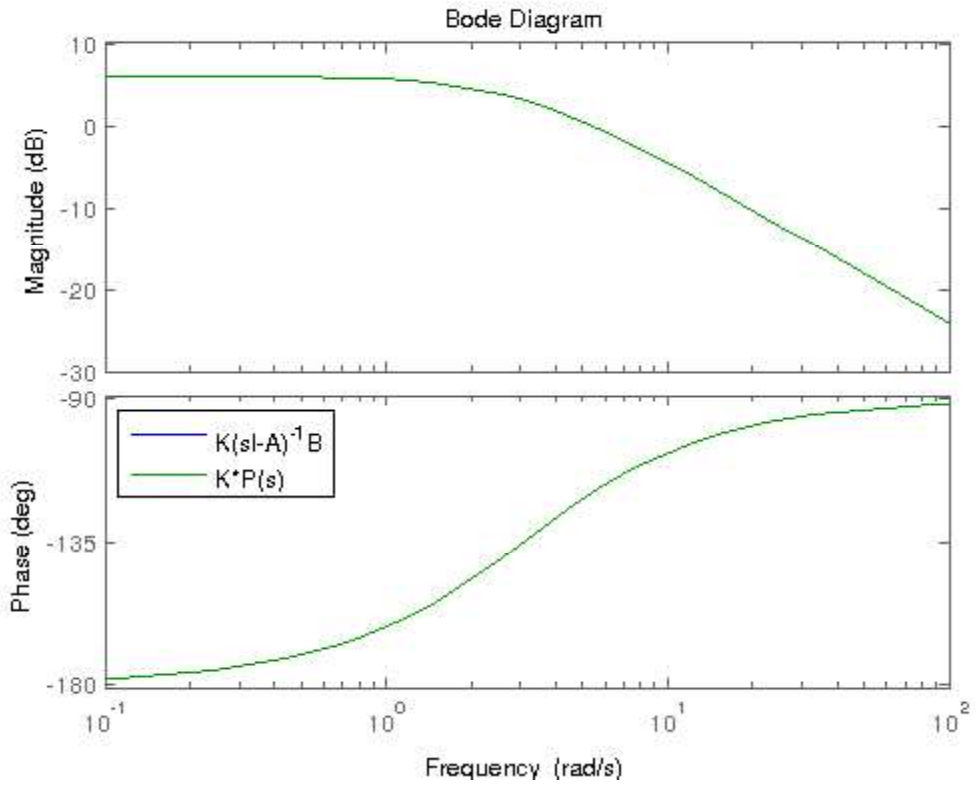
```
-K(1)
ans =

   -19.6051
```

and the derivative gain is

```
-K(2)
ans =

    -6.1626
```

__Open-loop gain bode plot.

```
s = tf('s');
Ls= K*(s*eye(size(A,1))-A)^-1*B;    % the open-loop gain LQR control
Ps=tf(ss(A,B,eye(size(A,1)),zeros(size(A,1),size(B,2))));  % plant
CPs = K*Ps;  % open-loop gain of the LQR controller (u to ubar)

figure(1); clf;
bode(Ls,CPs);  legend('K(sI-A)^{-
1}B','K*P(s)','interpreter','latex','location','northwest')
Warning: Ignoring extra legend
entries.
```



They're the same, just computed in different ways: TF vs SS.

__Closed-loop step response.

Suppose we had these initial conditions:

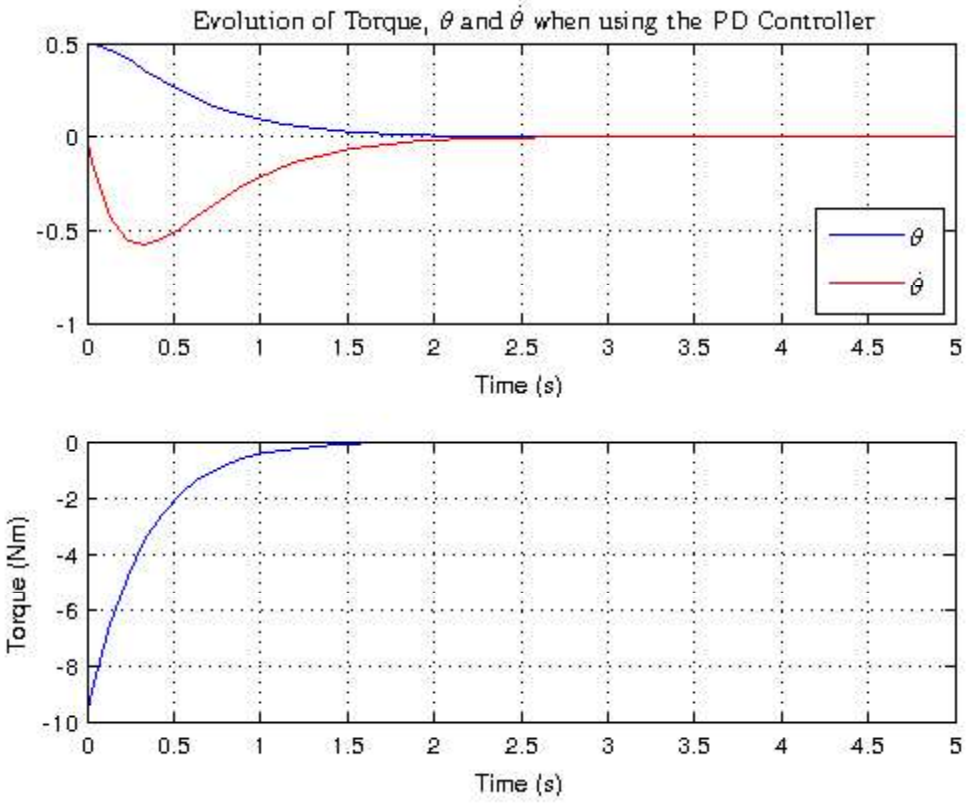
```
IC = [0.5 0];
```

Simulate.

```
warning off;
sim('P9_2_model');
theta = y.signals.values(:,1);
theta_dot = y.signals.values(:,2);
T = u.signals.values;
```

Plot controller performance.

```
figure(1); subplot(2,1,1);
plot(tout,theta,'-b',tout,theta_dot,'-r'); xlabel('Time (s)'); grid on;
h=legend('$\theta$','$\dot\theta$','Location','SouthEast');
set(h,'Interpreter','latex')
title('Evolution of Torque, $\theta$ and $\dot\theta$ when using the PD
Controller','Interpreter','latex');
subplot(2,1,2); plot(tout,T); xlabel('Time (s)'); ylabel('Torque (Nm)'); grid
on;
```



This figure shows how this controller stabilizes the pendulum from an initial condition.

Part 2: PID Design

Design a PID controller using LQR

```
clearvars -EXCEPT g l b m
```

When designing a PID controller, we augment the state to be

$$x = [\int \theta; \theta; \dot{\theta}] \qquad u = T$$

With this state augmentation, the augmented state transition matrix A is

```
A = [0 1 0; 0 0 1; 0 g/l -b/(m*l^2)];
```

and the augmented B matrix is

```
B = [0; 0; 1/(m*l^2)];
```

Since we're still doing LQR, we still assume $y = x$, so $C = I$ is 3-by-3.

```
C = eye(3); D = [0; 0; 0];
```

Decide the controlled variable z to be the vector $[\int \theta, \theta]$, so that G and H are

```
G = [1 0 0; 0 1 0]; H = [0; 0];
```

The exchange rate between nonzero state and controller actuation is (after some experimentation)

```
p = 10;
```

The QQ, RR, NN matrices (given by the book):

```
QQ = G'*G; RR = H'*H + p*1; NN = G'*H;
```

State-Feedback LQR controller (Optimal):

```
[K,S,E]=lqr(A,B,QQ,RR,NN);
K
```

```
K =

    0.3162   19.8061    6.1946
```

Since the control law is $u = -Kx = -K_1 \int \theta - K_2\theta - K_3\dot{\theta}$, the PID controller's integral gain is

```
-K(1)
```

```
ans =

   -0.3162
```

and its proportional gain is

```
-K(2)
```

```
ans =

  -19.8061
```

and its derivative gain is

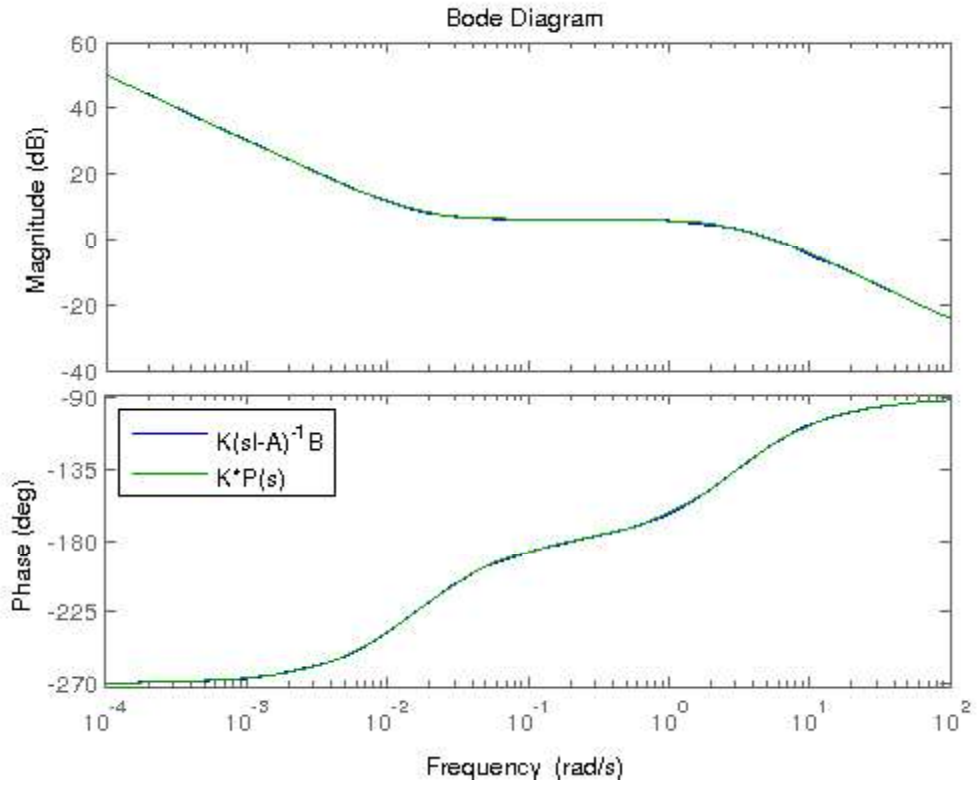
```
-K(3)
```

```
ans =

   -6.1946
```

Note that K_p and K_d are essentially unchanged from the PD controller.

__Open-loop gain bode plot.

```
s = tf('s');
Ls= K*(s*eye(size(A,1))-A)^-1*B;    % the open-loop gain LQR control
Ps=tf(ss(A,B,eye(size(A,1)),zeros(size(A,1),size(B,2))));   % plant
CPs = K*Ps;  % open-loop gain of the LQR controller (u to ubar)

figure(1); clf;
bode(Ls,CPs);  legend('K(sI-A)^{-
1}B','K*P(s)','interpreter','latex','location','northwest')
```



__Closed-loop step response.

Initial condition.

```
IC = [0 0.5 0];
```

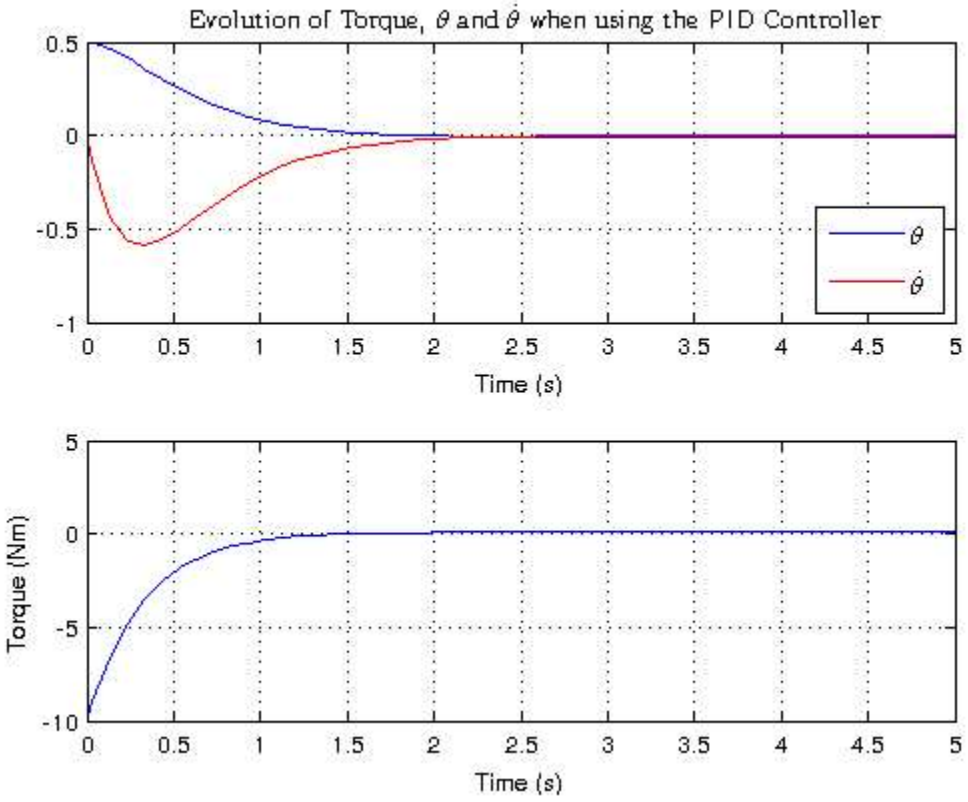
Simulate.

```
sim('P9_2_model');
theta = y.signals.values(:,2);
```

```
theta_dot = y.signals.values(:,3);
T = u.signals.values;
```

Plot.

```
figure(2); subplot(2,1,1);
plot(tout,theta,'-b',tout,theta_dot,'-r'); xlabel('Time (s)'); grid on;
h=legend('$\theta$','$\dot\theta$','Location','SouthEast');
set(h,'Interpreter','latex')
title('Evolution of Torque, $\theta$ and $\dot\theta$ when using the PID
Controller','Interpreter','latex');
subplot(2,1,2); plot(tout,T); xlabel('Time (s)'); ylabel('Torque (Nm)'); grid
on;
```



This figure shows how this controller stabilizes the pendulum from an initial condition. Slightly faster performance than the PD controller.