Homework Notebook

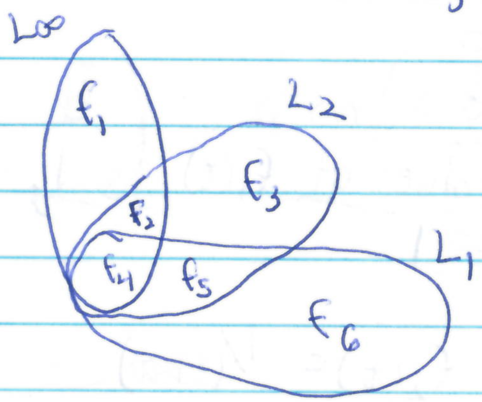
By Jason Nicholson

Table of Contents

Problem from Kelkar Lecture Norms lecture

Problem:

Prove that the following functions (at least two functions except $f_4(t)$) are located as shown in the figure



$f_k(t)=0$ f $t<0$ $(i=1,2,3,\dots,6)$

$f_1(t)=1$ $\quad f_2(t)=\dfrac{1}{1+t}$, $\quad f_3(t)=\dfrac{1}{1+t}\,\dfrac{1+t^{1/4}}{t^{1/4}}$

$f_4(t)=\exp(-t)$ $\quad f_5(t)=\dfrac{1}{1+t^2}\,\dfrac{1+t^{1/4}}{t^{1/4}}$

$f_6(t)=\dfrac{1}{1+t^2}\,\dfrac{1+t^{1/2}}{t^{1/2}}$

Solution:

Formulas to use

$L_\infty \Rightarrow \|f_k(t)\| \sup(|f_k(t)|)$

$L_2 \Rightarrow \|f_k(t)\|_2 = \left(\int_0^\infty (f_k(t))^2\,dt\right)^{1/2}$

$L_1 \Rightarrow \|f_k(t)\|_1 = \int_0^\infty |f_k(t)|\,dt$

solve for $f_1(t)=1$

$f_1(t)$ does not go to zero as t goes to infinity which implies the L_1 and L_2 norms are infinite.

$$\|f_1(t)\|_1 = \infty$$

$$\|f_1(t)\|_2 = \infty$$

The max value of $f_1(t)$ is 1, Therefore:

$$\|f_1(t)\|_\infty = 1$$

solve for $f_2(t) = 1/(1+t)$

$$\|f_2(t)\|_1 = \int_0^\infty \left|\frac{1}{(1+t)}\right| dt = \int_0^\infty \frac{1}{1+t}\, dt$$

$$= \lim_{\tau\to\infty} \ln(1+t)\Big|_0^\tau = \infty - 0 = \infty$$

$$\|f_2(t)\|_2 = \left(\int_0^\infty \frac{1}{(1+t)^2}\, dt\right)^{1/2} = \left(\lim_{\tau\to\infty} \frac{-1}{1+t}\Big|_0^\tau\right)^{1/2}$$

$$= (0+1)^{1/2} = 1$$

$\|f_2(t)\|_\infty = 1$ which occurs at $t=0$

Solve for $f_3(t)$

$f_3(t)$ is positive for $t\geq 0$

$$\|f_3(t)\|_1 = \int_0^\infty \left|\frac{1}{1+t}\,\frac{1+t^{1/4}}{t^{1/4}}\right| dt = \int_0^\infty \frac{1}{1+t}\,\frac{1+t^{1/4}}{t^{1/4}}\, dt$$

Used wxMaxima to integrate

$$\int_0^\infty \frac{1}{1+t}\,\frac{1+t^{1/4}}{t^{3/4}}\,dt =$$

$$4\left(\frac{(\sqrt{2}-1)\ln(\sqrt{t}+\sqrt{2}\,t^{1/4}+1)}{2^{5/2}}+\frac{(\sqrt{2}+1)\ln(\sqrt{t}-\sqrt{2}\,t^{1/4}+1)}{2^{5/2}}\right.$$

$$\left.+\frac{\operatorname{atan}\left(\frac{2t^{1/4}+\sqrt{2}}{\sqrt{2}}\right)}{2^{3/2}}+\frac{\operatorname{atan}\left(\frac{2t^{1/4}-\sqrt{2}}{\sqrt{2}}\right)}{2^{3/2}}\right)\Bigg|_0^\infty$$

$$= \infty - 0 = \infty$$

$$\|f_3(t)\|_2 = \left(\int_0^\infty \left(\frac{1}{1+t}\,\frac{1+t^{1/4}}{t^{1/4}}\right)^2 dt\right)^{1/2} = \left(\int_0^\infty \frac{(t^{1/4}+1)^2}{\sqrt{t}\,(t+1)^2}\,dt\right)^{1/2}$$

$$= \frac{\pi}{\sqrt{2}} - \frac{\pi+2}{2}$$

$$\|f_3(t)\|_\infty = \infty \quad \text{at } t=0$$

Solve case $f_4(t) = e^{-t}$

$$\|f_4(t)\|_1 = \int_0^\infty \exp(-t)\,dt = -\exp(t)\Big|_0^\infty = 0+1 = 1$$

$$\|f_4(t)\|_2 = \left(\int_0^\infty \exp(-2t)\,dt\right)^{1/2} = -\frac{1}{2}\exp(-2t)\Big|_0^\infty = \left(0+\frac{1}{2}\right)^{1/2} = \frac{1}{4}$$

$$\|f_4(t)\|_\infty = 1 \quad \text{at } t=0$$

Solve case $f_5(t) = \frac{1}{1+t^2} \frac{1+t^{1/4}}{t^{1/4}}$

$$\|f_5(t)\|_1 = \int_0^\infty \left| \frac{1}{1+t^2} \frac{1+t^{1/4}}{t^{1/4}} \right| dt = \int_0^\infty \frac{1}{1+t^2} \frac{1+t^{1/4}}{t^{1/4}} dt$$

~~Have to use Mupad in MATLAB~~

Used Mathmatica to integrate
www.integrals.wolfran.com

The integral is long so I will just present the numbers.

$$\boxed{\frac{1218665277476389}{2251799813685248}\pi + \frac{\pi}{2} \approx 3.271}$$

L_1 for f_5 exists

$$\|f_5(t)\|_2 = \left(\int_0^\infty \left(\frac{1}{(1+t^2)} \frac{(1+t^{1/4})}{t^{1/4}}\right)^2 dt\right)^{1/2}$$

Again used mathmatica to compute the integral

Only showing numbers for brevity

$$\cancel{\frac{6093263873819445}{900719925474092}\pi + 2.4156479}$$

$$(2.1252721154634 2 + 2.45147926520684)^{1/2} \approx 2.139$$

$$\|f_5(t)\|_\infty = \infty$$

because unbounded at $t=0$

Solve case $f_6(t) = \frac{1}{1+t^2}\,\frac{1+t^{1/2}}{t^{1/2}}$

$$\|f_6(t)\|_1 = \int_0^\infty \left|\frac{1}{1+t^2}\,\frac{1+t^{1/2}}{t^{1/2}}\right| dt = \int_0^\infty \frac{1}{1+t^2}\,\frac{1+t^{1/2}}{t^{1/2}}\,dt$$

Use mathmatica to integrate

$$= \frac{1}{4}\Big(\sqrt{2}\big(\ln(t+\sqrt{2}\sqrt{t}+1) - \ln(x-\sqrt{2}\sqrt{x}+1)\big) - 2(2+\sqrt{2})\arctan(1-\sqrt{2x}) + 2(\sqrt{2}-2)\arctan(\sqrt{2x}+1)\Big)\Big|_0^\infty$$

$$\boxed{= 2.22144146907918 + \frac{\pi}{2} \approx 3.79}$$

$$\|f_6(t)\|_2 = \left(\int_0^\infty \left(\frac{1}{1+t^2}\,\frac{(1+t^{1/2})}{t^{1/2}}\right)^2 dt\right)^{1/2} = (3.3321 + \infty)^{1/2} = \infty$$

L_2 does not exist

$$\|f_6(t)\|_\infty = \infty$$

because $f_6(t)$ is unbounded at $t=0$

From Linear Systems Theory and Design by Chen

2.5 Consider a system with input u and output y. Three experiments are performed on the system using the inputs $u_1(t)$, $u_2(t)$, and $u_3(t)$ for $t \geq 0$. In each case, the initial state $\mathbf{x}(0)$ at time $t = 0$ is the same. The corresponding outputs are denoted by y_1, y_2, and y_3. Which of the following statements are correct if $\mathbf{x}(0) \neq 0$?

1. If $u_3 = u_1 + u_2$, then $y_3 = y_1 + y_2$.
2. If $u_3 = 0.5(u_1 + u_2)$, then $y_3 = 0.5(y_1 + y_2)$.
3. If $u_3 = u_1 - u_2$, then $y_3 = y_1 - y_2$.

Which are correct if $\mathbf{x}(0) = \mathbf{0}$?

$$\vec{y}_1(t) = [C]\exp([A]t)\vec{x}(0) + \int_0^t [C]\exp([A](t-\tau))[B]\vec{u}_1(\tau)\,d\tau + D\vec{u}_1(t)$$

$$y_2(t) = [C]\exp([A]t)\vec{x}(0) + \int_0^t [C]\exp([A](t-\tau))[B]\vec{u}_2(\tau)\,d\tau + D\vec{u}_2(t)$$

$$y_1(t) + y_2(t) = 2[C]\exp([A]t)\vec{x}(0) + \int_0^t [C]\exp([A](t-\tau))[B](\vec{u}_1(\tau) + \vec{u}_2(\tau)) + [D](\vec{u}_1(t) + \vec{u}_2(t))$$

Statement one is true iff $\vec{x}(0) = 0$

Statement two is true iff $\vec{x}(0) = 0$

$$y_3(t) = [C]\exp([A]t)\vec{x}(0) + \int_0^t [C]\exp([A](t-\tau))[B](\vec{u}_1(\tau) + \vec{u}_2(\tau)) + [D](\vec{u}_1(t) + u_2(t))$$

$$y_1(t) - y_2(t) = \vec{0} + \int_0^t [C]\exp([A](t-\tau))[B](\vec{u}_1(t) - \vec{u}_2(t)) + [D](\vec{u}_1(t) - \vec{u}_2(t))$$

Statement three is true without $\vec{x}(0) = 0$ or with $\vec{x}(0) \neq 0$. It is always true

~~Proof with numbers for 2.5~~

$$A = \begin{bmatrix} -2 & -0.3 \\ 1 & -8 \end{bmatrix} \qquad B = \begin{bmatrix} -2 \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} -2 & 3 \end{bmatrix} \qquad D = 5$$

$$\vec{y}_1(t) = [C]\, e^{[A]t}\, \vec{x}(0) + \int_0^t [C]\, e^{[A](t-\tau)}\, [B]\, \vec{u}_1(\tau)\, d\tau + [D]\, \vec{u}_1(t)$$

$$\vec{y}_2(t) = [C]\, e^{[A]t}\, \vec{x}(0) + \int_0^t [C]\, e^{[A](t-\tau)}\, [B]\, \vec{u}_2(\tau)\, d\tau + [D]\, \vec{u}_2(\tau)$$

$$\vec{y}_1(t) + \vec{y}_2(t) = 2[C]\, e^{[A]t}\, \vec{x}(0) + \int_0^t [C]\, e^{[A](t-\tau)}\, [B]\left(\vec{u}_1(\tau) + \vec{u}_2(\tau)\right) d\tau$$
$$+ [D]\left(\vec{u}_1(t) + \vec{u}_2(t)\right)$$

$$0.5\left(\vec{y}_1(t) + \vec{y}_2(t)\right) = [C]\, e^{[A]t}\, \vec{x}(0) + 0.5\int_0^t [C]\, e^{[A](t-\tau)}\, [B]\left(\vec{u}_1(\tau) + \vec{u}_2(\tau)\right) d\tau$$
$$+ 0.5\,[D]\left(\vec{u}_1(t) + \vec{u}_2(t)\right)$$

$$\vec{y}_1(t) - \vec{y}_2(t) = \vec{0} + \int_0^t [C]\, e^{[A](t-\tau)}\, [B]\left(\vec{u}_1(\tau) - \vec{u}_2(\tau)\right) d\tau$$
$$+ [D]\left(\vec{u}_1(\tau) - \vec{u}_2(\tau)\right)$$

Case 1 $u_3 = u_1 + u_2$. Does $y_3 = y_1 + y_2$?
for $\vec{x}(0) \neq 0$

$$\vec{y}_3(t) = [C] e^{[A]t} \vec{x}(0) + \int_0^t [C] e^{[A](t-\tau)} [B] (\vec{u}_1(\tau) + \vec{u}_2(\tau)) d\tau$$
$$+ [D] (\vec{u}_1(t) + \vec{u}_2(t))$$

The initial condition term of $\vec{y}_3$ is different than $\vec{y}_1 + \vec{y}_2$, thus $\vec{y}_3 \neq \vec{y}_1 + \vec{y}_2$

Case 2 $u_3 = 0.5(u_1 + u_2)$. Does $\vec{y}_3 = 0.5(\vec{y}_1 + \vec{y}_2)$?

$$\vec{y}_3 = [C] e^{[A]t} \vec{x}(0) + 0.5 \int_0^t [C] e^{[A](t-\tau)} [B] (\vec{u}_1(t) + \vec{u}_2(t))$$

$$+ 0.5 [D] (\vec{u}_1(t) + \vec{u}_2(t))$$

All terms are same for $\vec{y}_3$ and $0.5(\vec{y}_1 + \vec{y}_2)$, thus $\vec{y}_3 = 0.5(\vec{y}_1 + \vec{y}_2)$ is true.

Case 3 $u_3 = u_1 - u_2$ Does $y_3 = y_1 - y_2$?

$$\vec{y}_3 = [C] e^{[A]t} \vec{x}(0) + \int_0^t [C] e^{[A](t-\tau)} [B] (\vec{u}_1(\tau) - \vec{u}_2(\tau)) d\tau$$
$$+ [D] (\vec{u}_1(t) - \vec{u}_2(t))$$

The initial condition term of $\vec{y}_3$ is different than $\vec{y}_1 - \vec{y}_2$, thus $\vec{y}_3 \neq y_1 + y_2$

Case 4 $\vec{u}_3 = \vec{u}_1 + \vec{u}_2$ and $\vec{x}(0) = 0$. Does $\vec{y}_3 = \vec{y}_1 + \vec{y}_2$?

Yes, $\vec{y}_3 = \vec{y}_1 + \vec{y}_2$ because in case 1 the only difference was the initial conditions term. Zeroing the initial conditions term makes everything equal.

Case 5 $\vec{u}_3 = \vec{u}_1 + \vec{u}_2$ and $\vec{x}(0) = 0$. Does $\vec{y}_3 = \vec{y}_1 + \vec{y}_2$?

Yes, $\vec{y}_3 = \vec{y}_1 + \vec{y}_2$. Same as case 2.

Case 6 $\vec{u}_3 = \vec{u}_1 - \vec{u}_2$ and $\vec{x}(0) = 0$ Does $\vec{y}_3 = \vec{y}_1 + \vec{y}_2$?

Yes, case 3 was only different by initial conditions term. Zero initial conditions term allows $\vec{y}_3 = \vec{y}_1 - \vec{y}_2$.

I also proved all this with numbers in MATLAB on the following pages.

Table of Contents


```
clc;
close all;
clear;
```

Create State Space System

```
A = [-2 -0.3; 1 -8];
B = [-2;0];
C=[-2 3];
D = 5;
sys = ss(A,B,C,D);
```

Create Various Inputs

```
t = 0:0.001:6; % 0 to 6 sec

% inputs to system
u1 = ones(size(t));
u2 = 4.25*ones(size(t));

% initial conditions
x0 = [5; 8];
```

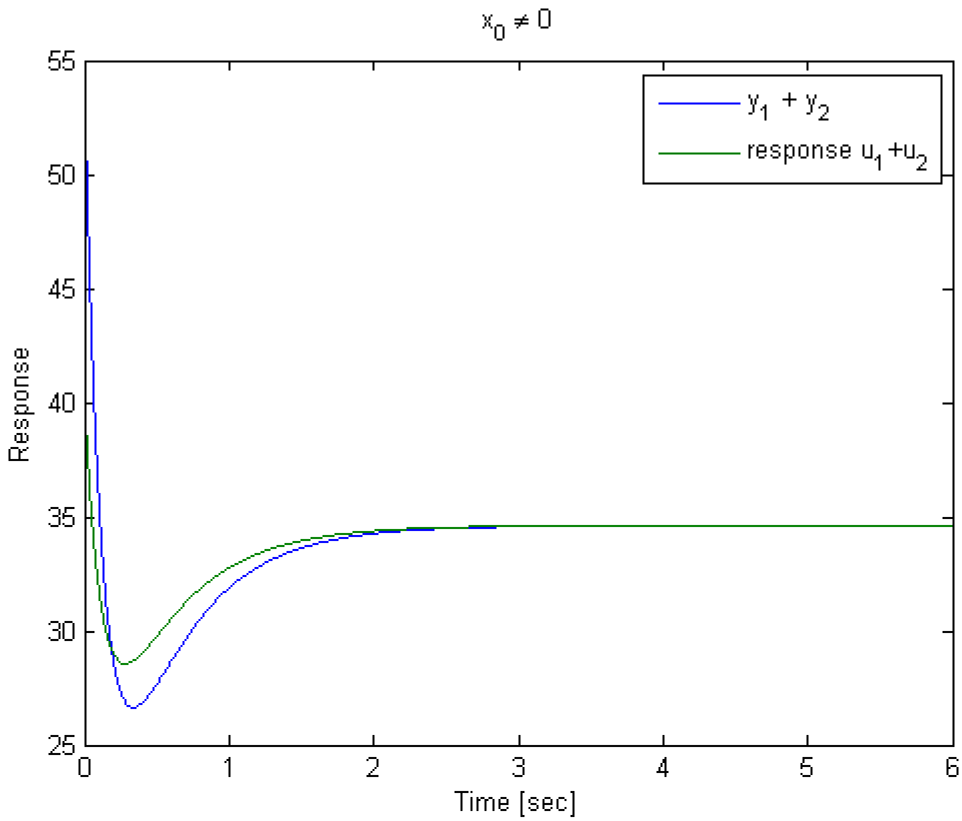
Response of System with Initial Conditions

```
[y1,t1,x1] = lsim(sys,u1,t,x0);
[y2,t2,x2] = lsim(sys,u2,t,x0);
```

Case 1 $x_0 \neq 0$

For initial conditions $x_0 \neq 0$, $u_3 = u_1 + u_2$ does not imply $y_3 = y_1 + y_2$.

```
u3 = u1 + u2;
[y_u3,t_u3,x_u3] = lsim(sys,u3,t,x0);
y3 = y1 + y2;
figure;
plot(t,y3,'DisplayName','y_1 + y_2');
hold('all');
plot(t,y_u3,'DisplayName','response u_1+u_2')
xlabel('Time [sec]');
ylabel('Response');
legend('show');
title('x_0 \neq 0');
```

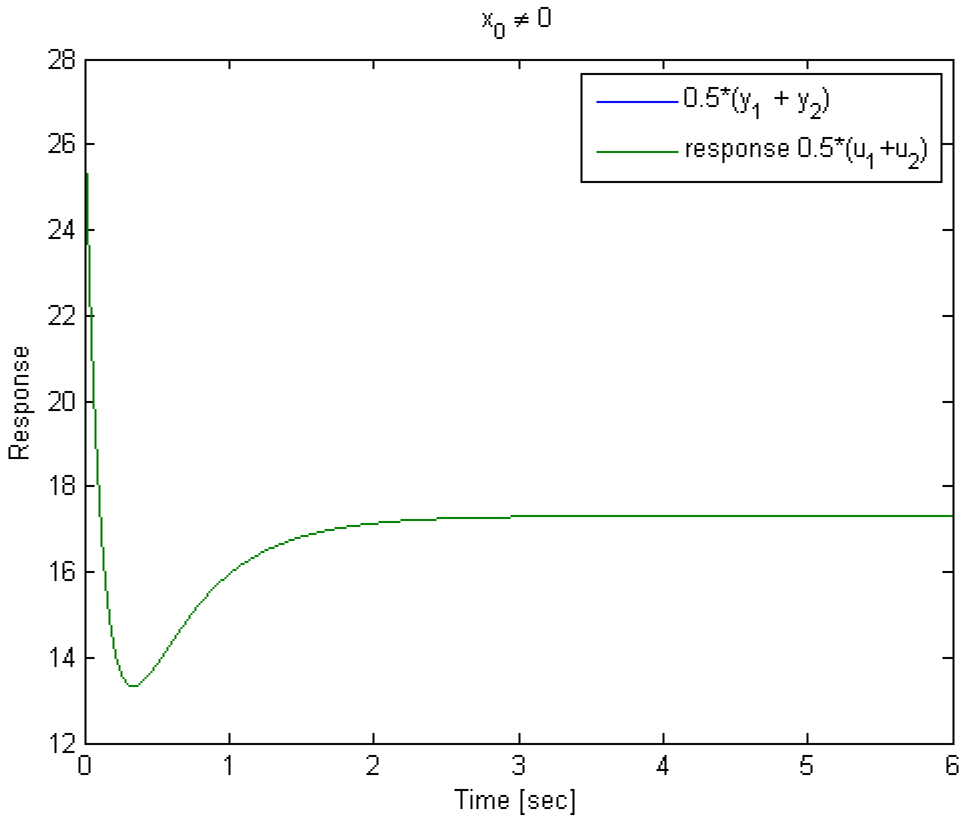


Case 2 $x_0 \neq 0$

For initial conditions $x_0 \neq 0$, $u_3 = 0.5 * (u_1 + u_2)$ implies $y_3 = 0.5 * (y_1 + y_2)$.

```
u3 = 0.5*(u1 + u2);
[y_u3,t_u3,x_u3] = lsim(sys,u3,t,x0);
y3 = 0.5*(y1 + y2);
figure;
plot(t,y3,'DisplayName','0.5*(y_1 + y_2)');
hold('all');
plot(t,y_u3,'DisplayName','response 0.5*(u_1+u_2)')
xlabel('Time [sec]');
ylabel('Response');
```

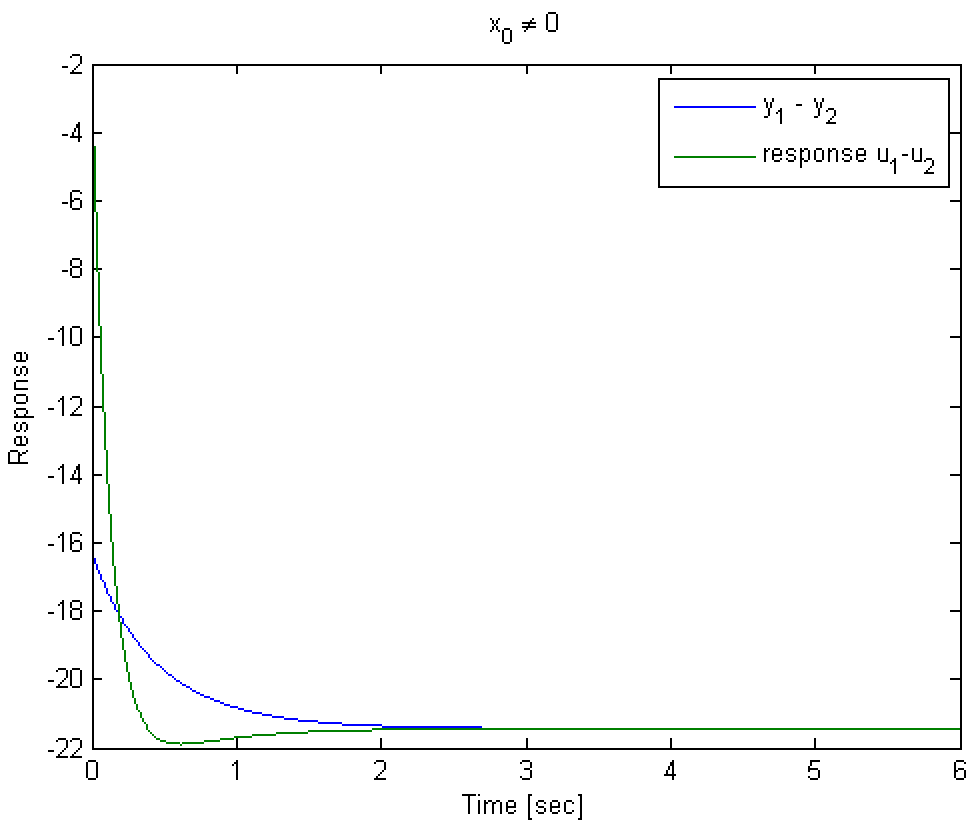
```
legend('show');
title('x_0 \neq 0');
```



Case 3 $x_0 \neq 0$

For initial conditions $x_0 \neq 0$, $u_3 = u_1 - u_2$ does not imply $y_3 = y_1 - y_2$.

```
u3 = u1 - u2;
[y_u3,t_u3,x_u3] = lsim(sys,u3,t,x0);
y3 = y1 - y2;
figure;
plot(t,y3,'DisplayName','y_1 - y_2');
hold('all');
plot(t,y_u3,'DisplayName','response u_1-u_2')
xlabel('Time [sec]');
ylabel('Response');
legend('show');
title('x_0 \neq 0');
```



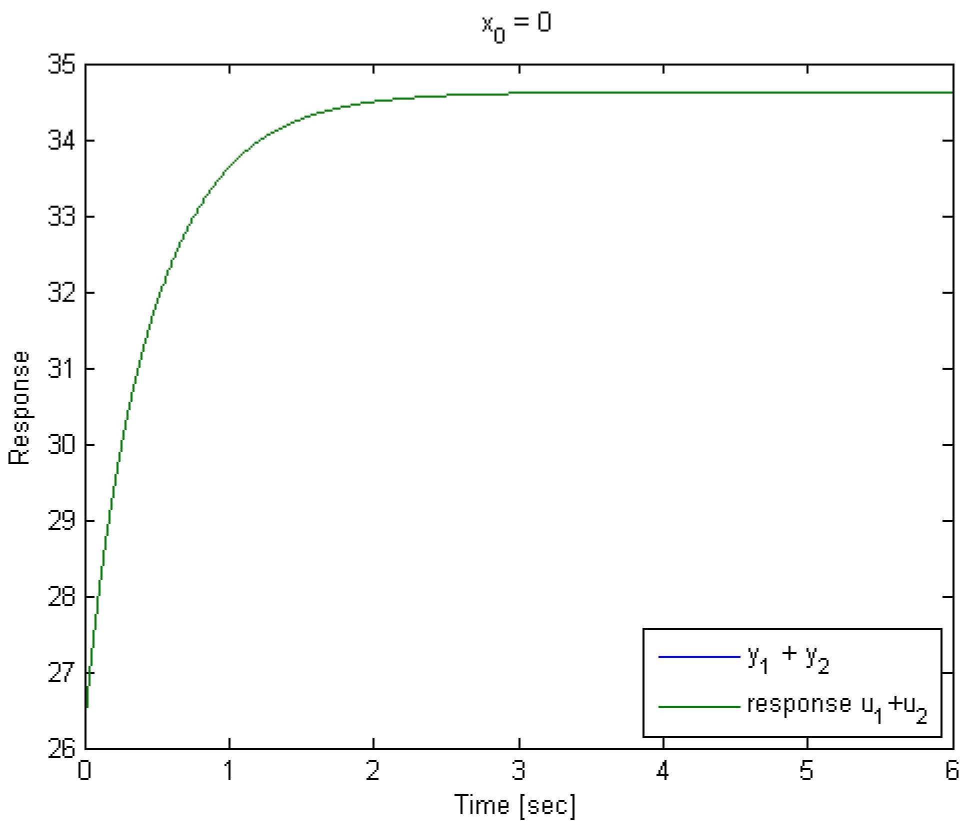
Set Initial Conditions $x_0 = 0$

```
x0 = [0;0];

[y1,t1,x1] = lsim(sys,u1,t,x0);
[y2,t2,x2] = lsim(sys,u2,t,x0);
```

Case 4 $x_0 = 0$

For initial conditions $x_0 = 0$, $u_3 = u_1 + u_2$ implies $y_3 = y_1 + y_2$.

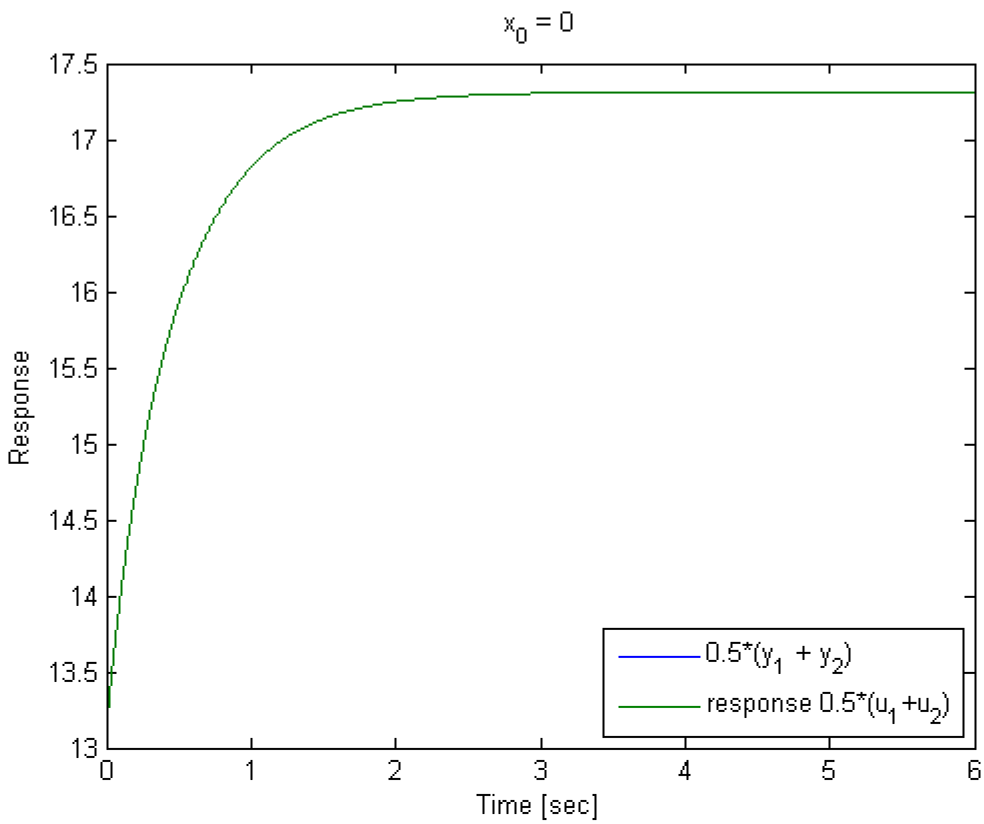
```
u3 = u1 + u2;
[y_u3,t_u3,x_u3] = lsim(sys,u3,t,x0);
y3 = y1 + y2;
figure;
plot(t,y3,'DisplayName','y_1 + y_2');
hold('all');
plot(t,y_u3,'DisplayName','response u_1+u_2')
xlabel('Time [sec]');
ylabel('Response');
legend('show','Location','SouthEast');
title('x_0 = 0');
```



Case 5 $x_0 = 0$

For initial conditions $x_0 = 0$, $u_3 = 0.5 * (u_1 + u_2)$ implies $y_3 = 0.5 * (y_1 + y_2)$.

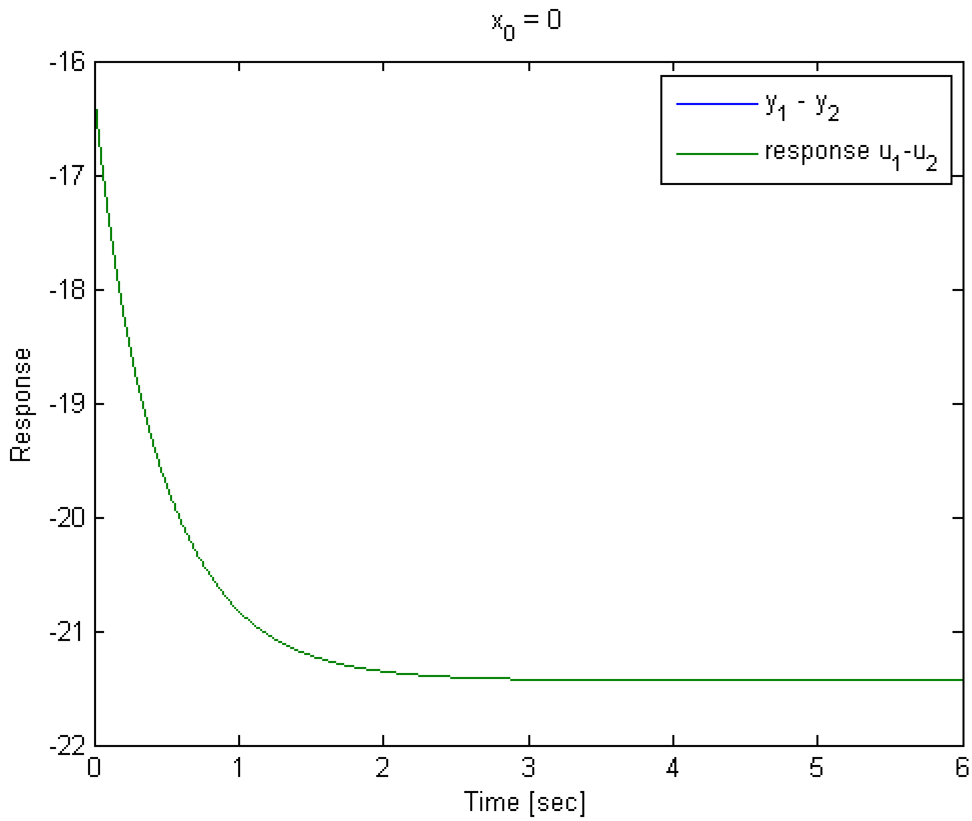
```
u3 = 0.5*(u1 + u2);
[y_u3,t_u3,x_u3] = lsim(sys,u3,t,x0);
y3 = 0.5*(y1 + y2);
figure;
plot(t,y3,'DisplayName','0.5*(y_1 + y_2)');
hold('all');
plot(t,y_u3,'DisplayName','response 0.5*(u_1+u_2)')
xlabel('Time [sec]');
ylabel('Response');
legend('show','Location','SouthEast');
title('x_0 = 0');
```



Case 6 $x_0 = 0$

For initial conditions $x_0 = 0$, $u_3 = u_1 - u_2$ implies $y_3 = y_1 - y_2$.

```
u3 = u1 - u2;
[y_u3,t_u3,x_u3] = lsim(sys,u3,t,x0);
y3 = y1 - y2;
figure;
plot(t,y3,'DisplayName','y_1 - y_2');
hold('all');
plot(t,y_u3,'DisplayName','response u_1-u_2')
xlabel('Time [sec]');
ylabel('Response');
legend('show');
title('x_0 = 0');
```



Published with MATLAB® R2013b

2.10 Chen

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 3 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} -1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

State Space Representation

$$\boxed{\frac{Y(s)}{U(s)} = \frac{(s-1)}{(s^2+2s-3)} = \frac{\cancel{(s-1)}}{(s+3)\cancel{(s-1)}} = \frac{1}{(s+3)}}$$

$$r_{1,2} = \frac{-2 \pm \sqrt{4-4\cdot 1\cdot -3}}{2} = \frac{-2 \pm 4}{2} = 1, -3$$

for impulse $u(s)=1$ & $y(t_0)=0$, $t_0=0$

$$\boxed{y(t) = e^{-3t}}$$

3.2 $\|\vec{x}\|_1 = \sum_{i=1}^{n} |x_i|$ n = length of $\vec{x}$

$\|\vec{x}\|_2 = \left(\sum_{i=1}^{n} (x_i)^2\right)^{1/2}$ " "

$\|\vec{x}\|_\infty = \max_{i=1..n} |x_i|$ " "

$\vec{x}_1 = \begin{bmatrix} 2 \\ -3 \\ 1 \end{bmatrix}$ $\vec{x}_2 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$

$\|x_2\|_1 = 3$ $\|x_2\| = \sqrt{3}$

$\|\vec{x}_1\|_1 = 6$ $\|\vec{x}_1\|_2 = \sqrt{14}$ $\|\vec{x}_1\|_\infty = 3$ $\|\vec{x}_2\|_\infty = 1$

3.5 Chen

Find rand & nullity of

$A_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ $\quad A_2 = \begin{bmatrix} 4 & 1 & -1 \\ 3 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix}$

$A_3 = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 0 & -1 & -2 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

~~A_1 is full rank, nullity is zero.~~

A_1 is rank 2. nullity is 1.

A_2 is full rank

$6+2$	$\sqrt{5}\ \sqrt{13}$
$8-2$	$\sqrt{5}\ \sqrt{21}$
$12-4$	$\sqrt{13}\ \sqrt{21}$

A_3 is rank 3 but close to rank TO because of column 2 & 3

3.7 $x = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ n~~ot~~ unique. ~~infinite solutions~~

for $y = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$, no solution

3.8

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 0 & -1 & -2 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} -1 \\ 0 \\ 0 \\ 1 \end{bmatrix}}_{= \vec{x}} = \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix}$$

3.12

$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ for } A_3$$

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

3.16

$$\begin{bmatrix} -\alpha_1 & -\alpha_2 & -\alpha_3 & -\alpha_4 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1/\alpha_4 & \frac{-\alpha_1}{\alpha_4} & \frac{-\alpha_2}{\alpha_4} & \frac{-\alpha_3}{\alpha_4} \end{bmatrix} =$$

$$\begin{bmatrix} -\alpha_4 \cdot \frac{-1}{\alpha_4} & -\alpha_1 + -\alpha_4 \frac{-\alpha_1}{\alpha_4} & -\alpha_2 + -\alpha_4 \cdot \frac{\alpha_2}{\alpha_4} & -\alpha_3 + \frac{-\alpha_4 \alpha_3}{\alpha_4} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

problem A-10-11 Ogata

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

u is a unit step

$$\vec{x}(0) = \vec{0}$$

$$\vec{x}(t) = \cancel{e^{[A]t}\,\vec{x}(0)}^{\,0} + \int_0^t e^{[A](t-\tau)}\,[B]\,\cancel{[K]}^{\,1}\,d\tau$$

$$\vec{x}(t) = e^{[A]t}\left(\int_0^t e^{-[A]\tau}\,d\tau\right)[B]$$

$$\vec{x}(t) = \underset{2\times2}{e^{[A]t}}\left(\underset{2\times2}{e^{-[A]t}} - \underset{2\times2}{[I]}\right) - \underset{2\times2}{[A]}^{-1}\,\underset{2\times1}{[B]}$$

$$\vec{x}(t) = \left([I] - e^{[A]t}\right) - [A]^{-1}\,[B]$$

$$\vec{x}(t) = \left(e^{[A]t} - [I]\right)[A]^{-1}\,[B]$$

$$-\lambda(-2-\lambda) - 0 = 0$$

$$\lambda = 0, -2$$

$$\cancel{e^{[A]t} = \begin{bmatrix} 1 & 0 \\ 0 & e^{-2t} \end{bmatrix}}$$

$$\begin{bmatrix} -\lambda & 1 \\ 0 & -2-\lambda \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = 0$$

$$\begin{bmatrix} 0 & 1 \\ 0 & -2 \end{bmatrix}$$

$v_1 =$ anything
$v_2 = 0$

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = 0$$

$2v_1 + v_2 = 0$
$v_2 = -2v_1$

$$\begin{bmatrix} 1 \\ -2 \end{bmatrix}$$

problem B-10-14 Ogata

Obtain $e^{[A]x}$

$$[A] = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & -3 & 3 \end{bmatrix}$$

$$\det(A) = \lambda^3 - 3\lambda^2 + 3\lambda - 1 = 0$$

$$\lambda = 1$$

$$\begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \\ 1 & -3 & 2 \end{bmatrix} \qquad J = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$V_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\cancel{[A]\vec{V}_1 = \vec{V}_1 + 1\vec{V}_2}$$

$$\begin{bmatrix} 1 & -2 & 1 \\ 1 & -2 & 1 \\ 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \vec{0}$$

$$M = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

$$\cancel{e^{At} = \begin{bmatrix} 2e^t & e^t+1 & -2e^t \\ e^t+1 & e^t+1 & -e^t-1 \\ \exp(t) & 1 & -1 \end{bmatrix}}$$

$$\cancel{A = \begin{bmatrix} 6 & 5 \\ 1 & 2 \end{bmatrix}}$$

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} e^t & te^t & \frac{t^2}{2}e^t \\ 0 & e^t & te^t \\ 0 & 0 & e^t \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & -1 \\ 1 & -1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} e^t + t^2 e^t \frac{1}{2} + t e^t & -t^2 e^t \frac{1}{2} & -t e^t \\ t^2 e^t \frac{1}{2} + t e^t & e^t - t^2 e^t \frac{1}{2} & -t e^t \\ \frac{1}{2} t^2 e^t & t e^t - t^2 e^t \frac{1}{2} & e^t - t e^t \end{bmatrix}$$

Problem B-10-13 Ogata

Obtain $e^{[A]t}$ $\quad [A] = \begin{bmatrix} -2 & -1 \\ 2 & -5 \end{bmatrix}$

$$\lambda_{1,2} = -3, -4$$

$$V_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$V_2 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$e^{At} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} e^{-3t} & 0 \\ 0 & e^{-4t} \end{bmatrix} \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}$$

$$e^{[A]t} = \begin{bmatrix} 2e^{-3t} - e^{-4t} & e^{-4t} - e^{-3t} \\ 2e^{-3t} - 2e^{-4t} & 2e^{-4t} - e^{-3t} \end{bmatrix}$$

Problem B-10-12 Ogata

$$G(s) = \underset{1\times 3}{\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}} \underset{3\times 3}{\begin{bmatrix} s & -1 & 0 \\ 0 & s & -1 \\ 600 & -100 & s+10 \end{bmatrix}^{-1}} \underset{3\times 1}{\begin{bmatrix} 0 \\ 10 \\ 0 \end{bmatrix}}$$

$$\cancel{G(s) = \frac{-10(s-10)}{s^3-10s^2+100s-600}}$$

$$G(s) = \frac{10(s+10)}{s^3+10s^2+100s+600}$$

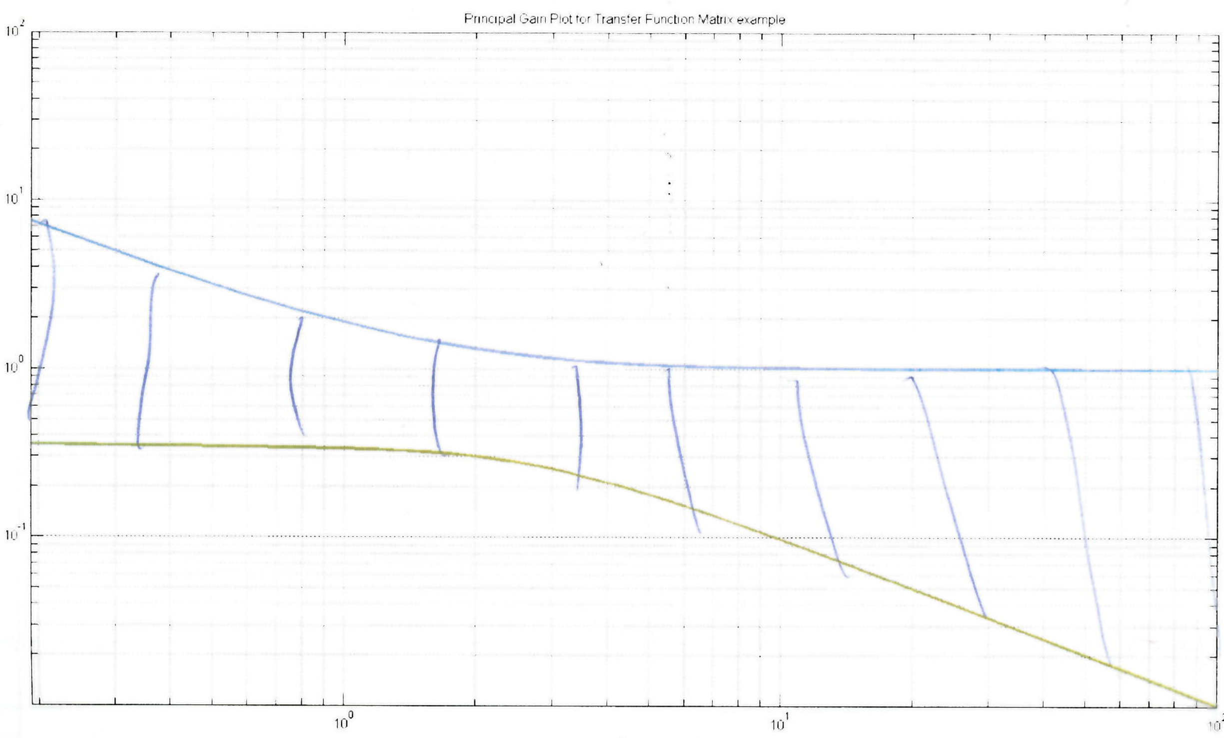
$$\begin{bmatrix} \dot{P}_c \\ \dot{Q}_I \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ -4 & -4 \end{bmatrix} \begin{bmatrix} P_c \\ Q_I \end{bmatrix} + \begin{bmatrix} 0 \\ 4 \end{bmatrix} P_{in}$$

$$Q_0 = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} P_c \\ Q_I \end{bmatrix}$$

$$G = \begin{bmatrix} \frac{1}{s+1} & \frac{1}{s+2} \\ \frac{1}{s} & \frac{1}{s+1} \end{bmatrix}$$

find principal gains

1. Sub $j\omega$ for s. Simplify
2. Calculate G^*
3. Find eigenvalues and eigen vectors of GG^*.
4. Plot $\sqrt{\lambda_1}$ or $\sqrt{\lambda_2}$ to make principal gain plot



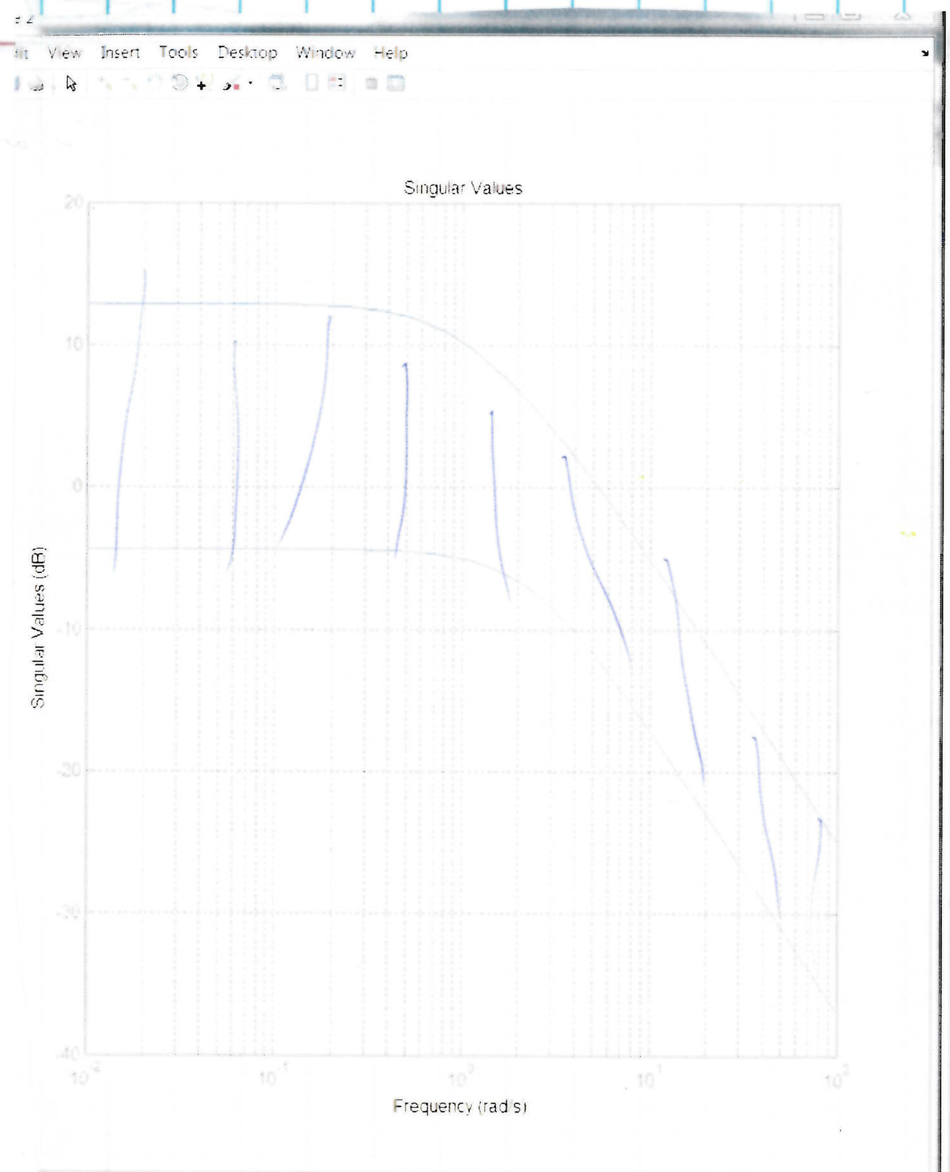
Exomple 2,3 of Burl

Find principal gains of the following system:

$$\dot{\vec{x}} = \begin{bmatrix} -1 & 2 \\ 0 & -3 \end{bmatrix} \vec{x} + \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix} \vec{u}$$

$$y = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \vec{x}$$

1. Find transfer function matrix, G
2. Sub $j\omega$ simplify
3. Find eigen values & vectors of GG^*.
4. Plot min & max $\sqrt{\text{eigen values}}$.



View Insert Tools Desktop Window Help
Singular Values
Singular Values (dB)
Frequency (rad/s)

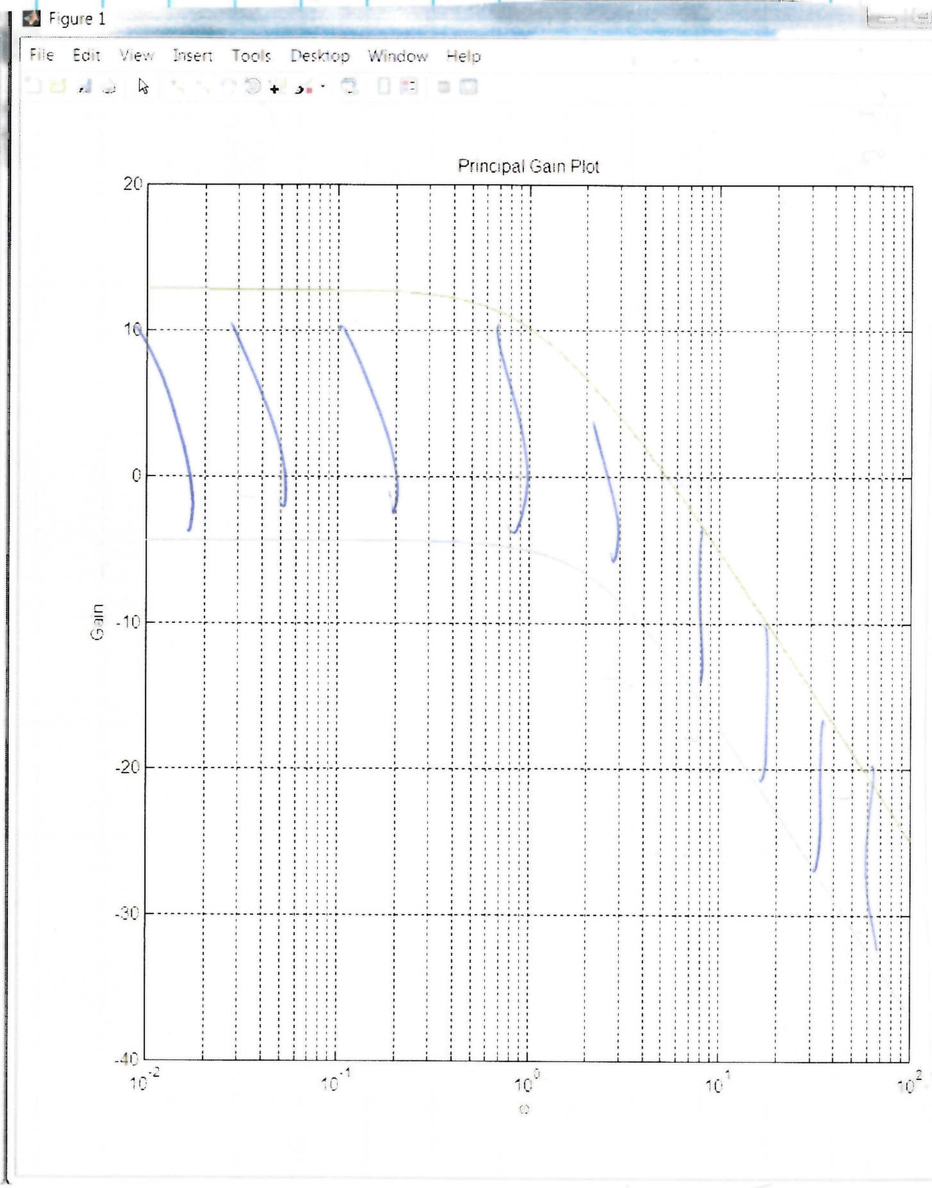


Figure 1
File Edit View Insert Tools Desktop Window Help
Principal Gain Plot
Gain
20
10
0
-10
-20
-30
-40
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
ω

2.5 Burl

Given:

$$\dot{\vec{x}} = \begin{bmatrix} 0 & 1 \\ -4 & -5 \end{bmatrix} \vec{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \vec{u}$$

$$\vec{y} = \begin{bmatrix} 1 & 0 \\ 0 & 10 \end{bmatrix} \vec{x}$$

Problem:

a. compute the tf matrix.

b. compute the poles, zeros, and modes?

c. compute the impulse response matrix.

d. How does this relate to the poles and modes?

Solution:

a.

$$sX = AX + Bu$$

$$(sI - A)X = Bu$$

$$X = (sI - A)^{-1} Bu$$

$$y = C(sI - A)^{-1} Bu + Du$$

$$y = (C(sI - A)^{-1} B + D)u$$

$$\boxed{G(s) = \begin{bmatrix} \dfrac{1}{s^2 + 5s + 4} \\ \dfrac{10s}{s^2 + 5s + 4} \end{bmatrix} = \begin{bmatrix} \dfrac{1}{(s+1)(s+4)} \\ \dfrac{10s}{(s+1)(s+4)} \end{bmatrix}}$$

b) poles $= -4, -1$ they are simple

zeros

$$sX = AX + BU$$
$$y = Cx + Du$$

$$\begin{array}{cc} \text{one} & \text{two} \\ \begin{bmatrix} sI - A & -B \\ C & D \end{bmatrix} & \begin{bmatrix} A - sI & B \\ C & D \end{bmatrix} \end{array}$$

No zeros? Yes no zeros

modes are e^{-t} and e^{-4t}

(i) $x_1 = x$
$x_2 = y$

53/60

+10 (a)(ii) $\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -2 & -5 \\ -6 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 2 \\ -4 \end{bmatrix} u$

$z = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

+10 b) $G(s) = C(sI - A)^{-1}B + D$

$\dfrac{10}{s^2+4s+3}$

2) (a)

+15

$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -2 & -1 & -5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u$

$y = \begin{bmatrix} 2 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$

1 b)

$s^3 y(s) + 5s^2 y(s) + s\, y(s) + 2y(s) = s\, u(s) + 2u(s)$

$\dfrac{y(s)}{u(s)} = \dfrac{(s+2)}{s^3+5s^2+s+2}$

3) $\dfrac{\partial f(x)}{\partial x} = \dfrac{2x}{(x+1)} - \dfrac{(x^2+1)}{(x+1)^2}$

+10

$\dfrac{\partial f(0)}{\partial x} = 0 - \dfrac{1}{1} = -1$

$f(0) = \dfrac{1}{1} = 1$

$\hat{f}(0.2) = 0.8$

underestimate

$\hat{f}(x) - 1 = -1(x - 0)$

$\hat{f}(x) = -x + 1$

(4) $\begin{bmatrix} \Delta\dot{x}_1 \\ \Delta\dot{x}_2 \end{bmatrix} = \begin{bmatrix} \sin(x_{20}) & f + x_{10}\cos(x_{20}) \\ e^{-x_{20}} & -x_{10}e^{-x_{20}} \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} + \begin{bmatrix} x_{20} \\ 2f_{20} \end{bmatrix} \Delta f$

-7

+8

$y = \begin{bmatrix} 2x_{20} & 2x_{10} + 2x_{20} \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix}$

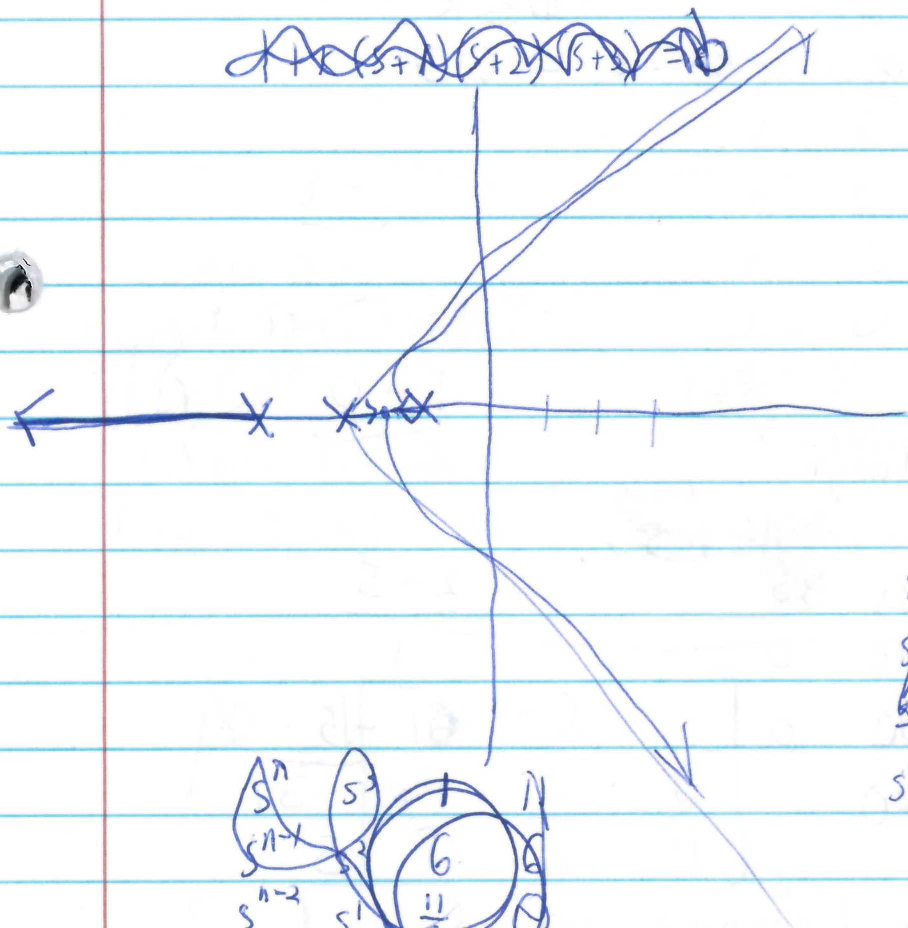
---

1 ~~$K \cdot K(s+1)(s+2)(s+3)$~~

$\frac{1}{1+GH}$

~~$1 + K(s+1)(s+2)(s+3) = 0$~~

$1 + K \frac{1}{(s+1)(s+2)(s+3)} = 0$



$(s+1)(s+2)(s+3) + 1 = 0$

$1 \quad 6 \quad 11 \quad (6+K)$

s^3	1	11
s^2	6	$(6+K)$
s^1	$10 + \frac{6}{K}$	0
s^0	$6+K$	

s^n	s^3	1
s^{n-1}	s^2	6
s^{n-2}	s^1	$\frac{11}{6}$
	s^0	6

$0 \le 6 + K$

$-6 \le K$

$0 < 10 + \frac{6}{K}$

~~$\left(10 + \frac{6}{K}\right)(6+K)$~~

$60 + 10K + \frac{36}{K} + 6$

$-10K \le 10K + 6$

CC $\quad -\frac{6}{10} \le K$

$$\begin{array}{lll} s^3 & 1 & 11 \\ s^2 & 6 & (6+K) \\ s^1 & \frac{66-(6+K)}{6} & 0 \\ s^0 & (6+K) & \end{array}$$

$$\frac{66-(6+K)}{6}$$

$11 -$

$$\frac{60-K}{6} \geq 0$$

$60-K \geq 0$

$60 \geq K$

$-6 < K$

$-6 \leq K \leq 60$

2) a)

$$\begin{array}{llll} s^4 & 1 & 3 & 5 \\ s^3 & 1 & 2 & 0 \\ s^2 & 1 & 5 & 0 \\ s^1 & -3 & 0 & \\ s^0 & 5 & & \end{array}$$

$$\frac{3-2}{1}$$

$$\frac{5-0}{1}$$

2 sign changes, means two unstable poles

$$\frac{2-5}{1}$$

$$\frac{-15-0}{-3}$$

5

b)

$$\begin{array}{llll} s^4 & 1 & 33 & 30 \\ s^3 & 10 & 46 & 0 \\ s^2 & 28.4 & 30 & 0 \\ s^1 & 35.4366 & 0 & \\ s^0 & 30 & & \end{array}$$

all stable poles

$$\frac{330-46}{10}$$

3 $G(s) = \dfrac{50\left(\frac{s}{10}+1\right)\left(\frac{s}{5}+1\right)}{20 \cdot 2 \cdot 20\left(\frac{s}{20}+1\right)(s+1)\left(\frac{s}{20}+1\right)}$

$G(j\omega) = \dfrac{50}{800} \dfrac{\left(\frac{j\omega}{10}+1\right)\left(\frac{j\omega}{5}+1\right)}{\left(\frac{j\omega}{20}+1\right)\left(j\omega+1\right)\left(\frac{j\omega}{20}+1\right)}$

$20\log_{10}|G(j\omega)| = 20\log_{10}\left(\frac{50}{800}\right) + 20\log_{10}\left(\frac{j\omega}{5}+1\right)$
$- 20\log_{10}\left(\frac{j\omega}{20}+1\right) - 20\log_{10}(j\omega+1)$
$- 20\log_{10}\left(\frac{j\omega}{20}+1\right)$

Quiz 3

1 ~~(i) (a) 0 (b) 1/2~~
~~(ii) (a)~~
(i) (a) 0 (b)
(ii) (a) 0 (b)

$\dfrac{1}{1+\frac{s(s+3)(s+2)}{s^2(s+3)(s+4)}} = \dfrac{s^2(s+3)(s+4)}{s^2(s+3)(s+4)+(s+3)(s+2)}$

$A = \begin{bmatrix} 1 & -2 & 3 \\ -4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$

Find 1, 2, ∞, spectral radius, Frobenius, and max norms

$\|A\|_1 = \max(\{12 \quad 15 \quad 18\}) = 18$
$\|A\|_2 = \max(\text{svd}(A)) = 16.8481$
$\|A\|_\infty = \max(\{6, 15, 24\}) = 24$

$\rho(A) = \max(\text{eig}(A)) = \cancel{6.\ldots}\ 16.1168$

$\|A\|_{max} = 9$

$\|A\|_F = \cancel{8.8750}\ 16.8819$

Example 1 on controllability

Given:

$$\dot{\vec{x}} = \begin{bmatrix} -0.5 & 0 \\ 0 & -1 \end{bmatrix} \vec{x} + \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} u$$

Find: Find $u(\tau)$ that can take system from $x(0) = [10 \quad -1]^T$ to $x(2) = [0 \quad 0]^T$

Solution 1st check is it controllable

$$\mathcal{C} = \begin{bmatrix} B & AB \end{bmatrix} = \begin{bmatrix} 0.5 & -0.25 \\ 1 & -1 \end{bmatrix}$$

$\det(\mathcal{C}) = -0.25$ which $\Rightarrow$ full rank

$$W_c = \int_0^2 e^{-A\tau} BB^T e^{-A^T\tau}\, d\tau$$

$$W_c = \frac{1}{4}\int_0^2 \begin{bmatrix} e^{\tau} & 2e^{\frac{\tau}{2}} e^{\tau} \\ 2e^{\tau/2} e^{\tau} & 4e^{2\tau} \end{bmatrix} d\tau$$

$$W_c = \begin{bmatrix} \frac{e^2}{4} - \frac{1}{4} & \frac{e^3}{3} - \frac{1}{3} \\ \frac{e^3}{3} - \frac{1}{3} & \frac{e^4}{2} - \frac{1}{2} \end{bmatrix}$$

$$W_c \approx \begin{bmatrix} 1.5973 & 6.3618 \\ 6.3618 & 26.799 \end{bmatrix}$$

sub into equation for u

$$\vec{u}(t) = B^T e^{-At} W_c^{-1} e^{-At} \left(\vec{x}(2) - e^{At} \vec{x}(0) \right)$$

$$\vec{u}(t) = \begin{bmatrix} 0.5 & 1 \end{bmatrix} \begin{bmatrix} e^{0.5t} & 0 \\ 0 & e^{t} \end{bmatrix} W_c^{-1} \begin{bmatrix} e^{0.5t} & 0 \\ 0 & e^{t} \end{bmatrix} \left(\begin{bmatrix} 0 \\ 0 \end{bmatrix} - \right.$$

$$\left. \begin{bmatrix} e^{-0.5t} & 0 \\ 0 & e^{-t} \end{bmatrix} \begin{bmatrix} 10 \\ -1 \end{bmatrix} \right)$$

$$\vec{u}(t) \approx 27.964\, e^{t} - 58.82 \cdot e^{0.5t}$$

Example 3 from Kelkar Lectures $\dot{x} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u$

$$C_o = [B \quad AB]$$

$$= \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \text{ rank} = 2 \text{ controllable}$$

(a) $x_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ $x_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

$$W_c(0,2) = \int_0^2 \begin{bmatrix} 1 & 0 \\ 0 & e^t \end{bmatrix} \overset{2\times1}{\begin{bmatrix} 1 \\ 1 \end{bmatrix}} \overset{1\times2}{\begin{bmatrix} 1 & 1 \end{bmatrix}} \begin{bmatrix} 1 & 0 \\ 0 & e^t \end{bmatrix} dt$$

$$\overset{2\times2}{\begin{bmatrix} 1 & 0 \\ 0 & e^t \end{bmatrix}} \overset{2\times2}{\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}}$$

$$\begin{bmatrix} 1 & 1 \\ e^t & e^t \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & e^t \end{bmatrix}$$

$$W_c(0,2) \int_0^2 \begin{bmatrix} 1 & e^t \\ e^t & e^{2t} \end{bmatrix} dt = \begin{bmatrix} t & e^t \\ e^t & \frac{e^{2t}}{2} \end{bmatrix} \Bigg|_0^2 = \begin{bmatrix} 2 & e^2-1 \\ e^2-1 & \frac{e^4-1}{2} \end{bmatrix}$$

$$W_c^{-1} = \frac{1}{2e^2-2} \begin{bmatrix} \frac{e^4-1}{2} & -e^2+1 \\ -e^2+1 & 2 \end{bmatrix}$$

$$u(t) = \overset{1\times2}{\begin{bmatrix} 1 & 1 \end{bmatrix}} \overset{2\times2}{\begin{bmatrix} 1 & 0 \\ 0 & e^{2-t} \end{bmatrix}} \frac{1}{2e^2-2} \begin{bmatrix} \frac{e^4-1}{2} & -e^2+1 \\ -e^2+1 & 2 \end{bmatrix} \left(\begin{bmatrix} 0 \\ 1 \end{bmatrix} - \right.$$

$$\begin{bmatrix} 1 & 0 \\ 0 & e^2 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \end{bmatrix}\Bigg)$$

$$u(t) = \begin{bmatrix} 1 & e^{2-t} \end{bmatrix} \frac{1}{2e^2-2} \begin{bmatrix} \frac{e^4-1}{2} & -e^2+1 \\ -e^2+1 & 2 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$u(t) = \frac{1}{2e^2-2} \begin{bmatrix} 1 & e^{2-t} \end{bmatrix} \begin{bmatrix} -e^2+1 \\ 2 \end{bmatrix}$$

$$u(t) = \frac{1}{2e^2-2}\left(-e^2+1 + 2e^{2-t}\right)$$

$$u(t) = \frac{1}{2(e^2-1)}\left(-(e^2-1) + 2e^{2-t}\right)$$

$$\boxed{u(t) = -\frac{1}{2} + \frac{e^{2-t}}{e^2-1}}$$

Compared to Matlab plot. It is correct.

(b) $x_0 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$ $\quad x_1 = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$

$t_0 = 0 \quad t_1 = 4$

$$W_c(4) = \begin{bmatrix} t & e^t \\ e^t & \frac{e^{2t}}{2} \end{bmatrix}\Bigg|_0^4 = \begin{bmatrix} 4 & e^4-1 \\ e^4-1 & \frac{e^8-1}{2} \end{bmatrix}$$

$$W_c^{-1} = \begin{bmatrix} \frac{e^8-1}{2} & -e^4+1 \\ -e^4+1 & 4 \end{bmatrix} \frac{1}{e^8+2e^4-3}$$

$$u(t) = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & e^{4-t} \end{bmatrix} \begin{bmatrix} \frac{e^8-1}{2} & -e^4+1 \\ -e^4+1 & 4 \end{bmatrix} \frac{1}{e^8+2e^4-3}$$

$$\left(\begin{bmatrix} 3 \\ 3 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & e^4 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} \right)$$

$$u(t) = \begin{bmatrix} \frac{e^8-1}{2} - e^{8-t} + e^{4-t} & -e^4+1+4e^{4-t} \end{bmatrix} \frac{1}{e^8+2e^4-3} \begin{bmatrix} 1 \\ 3-e^4 \end{bmatrix}$$

$$u(t) = \frac{\left(\frac{1}{2}e^8 - \frac{1}{2} - e^{8-t} + e^{4-t} + e^8 - e^4 - 4e^{8-t} - 3e^4 + 3 + 12e^{4-t}\right)}{e^8+2e^4-3}$$

$$\boxed{u(t) = \frac{\left(\left(-5e^8 + 13e^4\right)e^{-t} + \left(\frac{5}{2} + \frac{3}{2}e^8 - 4e^4\right)\right)}{e^8+2e^4-3}}$$

Compared to Matlab plot of correct answer. It is correct

Table of Contents


```
% clean up
clc; clear; close all;
```

Given

```
A = [ 0  0;
      0  1];
B = [1;
     1];
x_0 = [0;
       0];
x_f = [0;
       1];

t_f = 2;
```

Solution

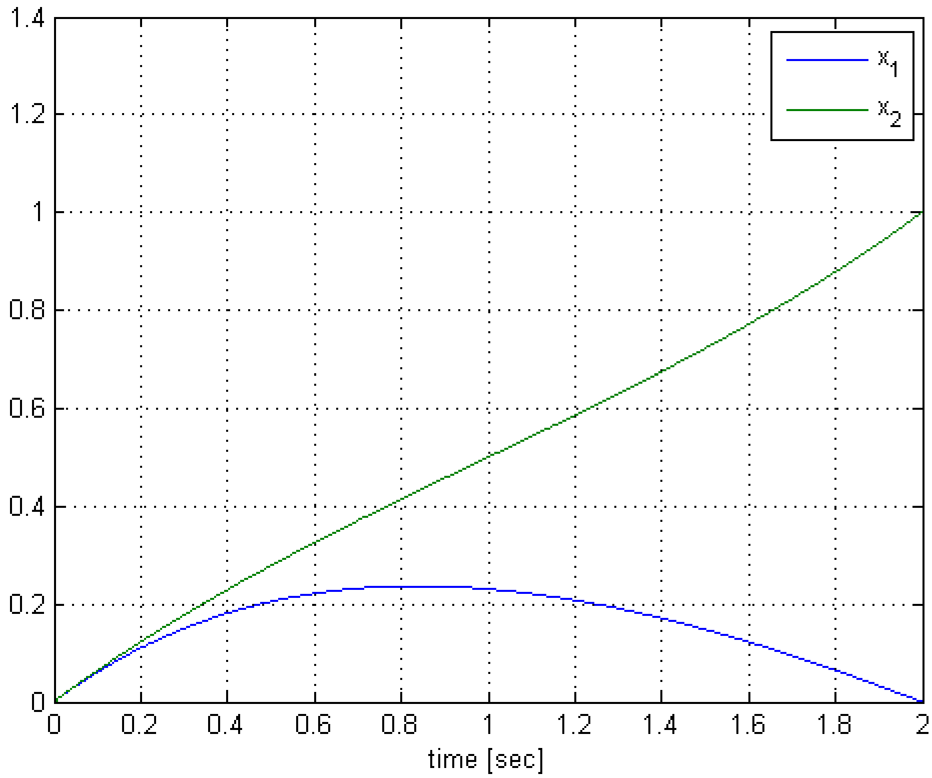
```
syms tau t;
W_c = int(expm(A*tau)*B*B'*expm(A'*tau),tau,0,t_f); % Using your definition
                                                     % of W_c
disp('W_c = ')
disp(vpa(W_c,5)) % 5 digits of accuracy

u = B'*expm(A'*(t_f-t))*W_c^(-1)*(x_f-expm(A*t_f)*x_0);
disp('u = ')
disp(vpa(u,5)) % 5 digits of accuracy
```

```
W_c =
[    2.0, 6.3891]
[ 6.3891, 26.799]

u =
0.15652*exp(2.0 - 1.0*t) - 0.5
```

Check By plotting

```
tt = 0:0.001:t_f;
% convert symbolic expression to string and substitute tt for t
uu = eval(strrep(char(u),'t','tt'));
```

```
C = [1 0;0 1]; % This means y = x
D = 0;
sys = ss(A,B,C,D);

xx = lsim(sys,uu,tt,x_0);
plot(tt,xx)
hold('all');
xlabel('time [sec]')
grid('on')
legend('x_1','x_2')
```



Published with MATLAB® R2014a

Table of Contents


```
% clean up
clc; clear; close all;
```

Given

```
A = [ 0  0;
      0  1];
B = [1;
     1];
x_0 = [2;
       1];
x_f = [3;
       3];

t_f = 4;
```

Solution

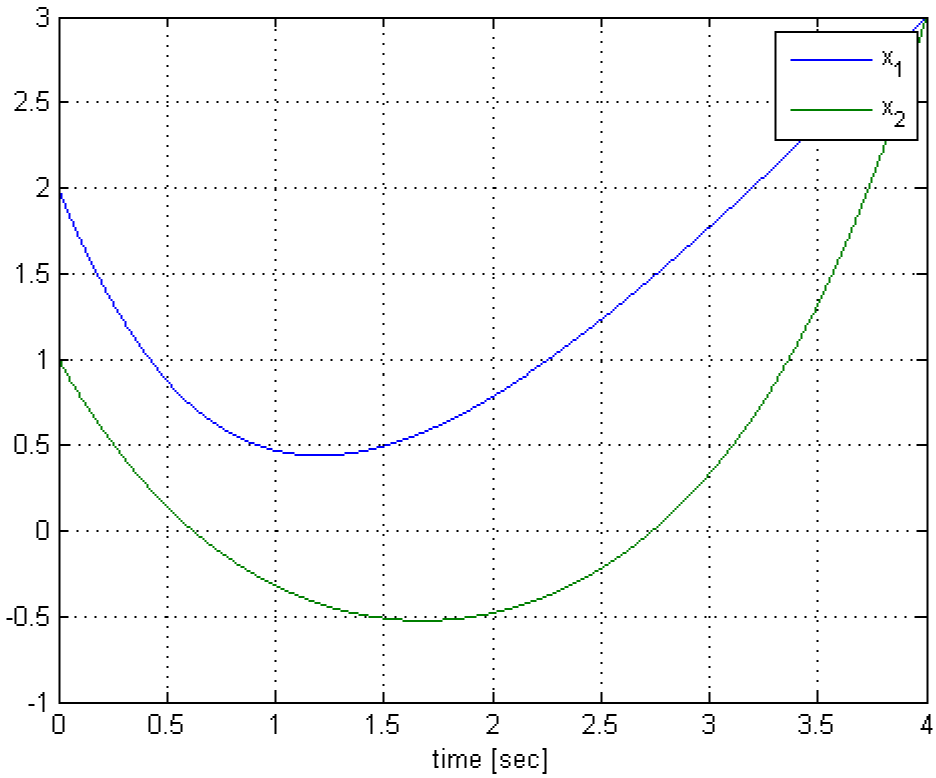
```
syms tau t;
W_c = int(expm(A*tau)*B*B'*expm(A'*tau),tau,0,t_f); % Using your
                                                    % definition of W_c
disp('W_c = ')
disp(vpa(W_c,5)) % 5 digits of accuracy

u = B'*expm(A'*(t_f-t))*W_c^(-1)*(x_f-expm(A*t_f)*x_0);
disp('u = ')
disp(vpa(u,5)) % 5 digits of accuracy
```

```
W_c =
[    4.0, 53.598]
[ 53.598, 1490.0]

u =
1.3785 - 0.084217*exp(4.0 - 1.0*t)
```

Check by Plot

```
tt = 0:0.001:t_f;
% convert symbolic expression to string and substitute tt for t
uu = eval(strrep(char(u),'t','tt'));
```

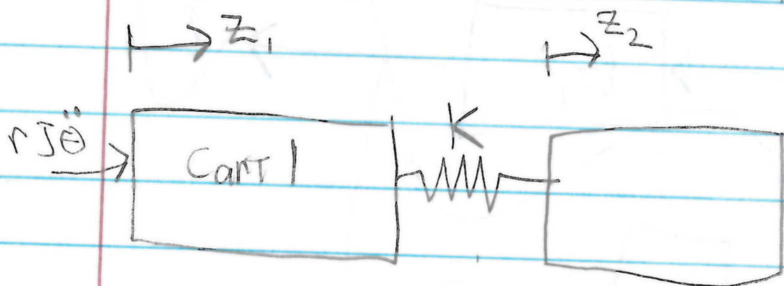
```
C = [1 0;0 1]; % This means y = x
D = 0;
sys = ss(A,B,C,D);

xx = lsim(sys,uu,tt,x_0);
plot(tt,xx)
hold('all');
xlabel('time [sec]')
grid('on')
legend('x_1','x_2')
```



Published with MATLAB® R2014a

From Kelkar Lectures

Example 4 Controllability and Observability



States to use to define system

$z_1, \dot{z}_1, z_2, \dot{z}_2, \dot{\theta}_1, \dot{\theta}_2$

$$\underbrace{\begin{bmatrix} \dot{z}_1 \\ \ddot{z}_1 \\ \dot{z}_2 \\ \ddot{z}_2 \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix}}_{6\times1} = \underbrace{\begin{array}{c} \begin{array}{cccccc} z_1 & \dot{z}_1 & z_2 & \dot{z}_2 & \dot{\theta}_1 & \dot{\theta}_2 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -\frac{K}{M} & 0 & \frac{K}{M} & 0 & -\frac{k^2 r}{RM} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{K}{M} & 0 & -\frac{K}{M} & 0 & 0 & -\frac{k^2 r}{RM} \\ 0 & 0 & 0 & 0 & -\frac{k^2}{RJ} & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{k^2}{RJ} \end{bmatrix} \end{array}}_{6\times6} \underbrace{\begin{bmatrix} z_1 \\ \dot{z}_1 \\ z_2 \\ \dot{z}_2 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}}_{6\times1} + \underbrace{\begin{bmatrix} 0 & 0 \\ \frac{rk}{RM} & 0 \\ 0 & 0 \\ 0 & \frac{kr}{RM} \\ \frac{k}{RJ} & 0 \\ 0 & \frac{k}{RJ} \end{bmatrix}}_{6\times2} \underbrace{\begin{bmatrix} e_1 \\ e_2 \end{bmatrix}}_{2\times1}$$

Controllability Matrix

$$Co = \begin{bmatrix} B & AB & A^2B & A^3B & A^4B & A^5B \end{bmatrix}$$

$\text{rank}(Co) = 5$ thus system is not controllable.

Contents


```
function controllability_example4
```

```
syms J k K M r R positive
```

Given

```
A = [    0       1    0       0    0                         0;
        -K/M     0    K/M     0    -k^2*r/(R*M)              0;
        0        0    0       1    0                         0;
        K/M      0    -K/M    0    0              -k^2*r/(R*M);
        0        0    0       0    -k^2/(R*J)                0;
        0        0    0       0    0                -k^2/(R*J)];

B = [0          0;
    r*k/(R*M)  0;
    0          0;
    0  k*r/(R*M);
    k/(R*J)    0;
    0    k/(R*J)];
```

Quick Checks

```
disp('rank(A)=')
disp(rank(A))
disp('rank(B)=')
disp(rank(B))
```

Part (a)

Is the system controllable using only one motor?

Answer: No it is not controllable because the rank is of the controllability matrix is less than 6.

```
co = controllabilityMatrix(A(1:5,1:5),B(1:5,1))
rank(co)
```

```
co =

[           0,        (k*r)/(M*R),                    -(k^3*r)/(J*M*R^2),       (k^5*r)/(J^2*M*R^3) - (K*k*r)/(M^2*R),                         (K*k^3*r)/(J*M^2*R^2) - (k^7*r)/(J^3*M*R^4),              (K^2*k*r)/(M^3*R) - (K*((k^5*r)/(J^2*M*R^3) - (K*k*r)/(M^2*R)))/M + (k^9*r)/(J^4*M*R^5)]
[ (k*r)/(M*R), -(k^3*r)/(J*M*R^2), (k^5*r)/(J^2*M*R^3) - (K*k*r)/(M^2*R), (K*k^3*r)/(J*M^2*R^2) - (k^7*r)/(J^3*M*R^4), (K^2*k*r)/(M^3*R) - (K*((k^5*r)/(J^2*M*R^3) - (K*k*r)/(M^2*R)))/M + (k^9*r)/(J^4*M*R^5), (K*((k^7*r)/(J^3*M*R^4) - (K*k^3*r)/(J*M^2*R^2)))/M - (k^11*r)/(J^5*M*R^6) - (K^2*k^3*r)/(J*M^3*R^2)]
[           0,                  0,                                     0,                       (K*k*r)/(M^2*R),                                      -(K*k^3*r)/(J*M^2*R^2),                                        (K*((k^5*r)/(J^2*M*R^3) - (K*k*r)/(M^2*R)))/M - (K^2*k*r)/(M^3*R)]
[           0,                  0,                       (K*k*r)/(M^2*R),                      -(K*k^3*r)/(J*M^2*R^2),                 (K*((k^5*r)/(J^2*M*R^3) - (K*k*r)/(M^2*R)))/M - (K^2*k*r)/(M^3*R),                        (K^2*k^3*r)/(J*M^3*R^2) - (K*((k^7*r)/(J^3*M*R^4) - (K*k^3*r)/(J*M^2*R^2)))/M]
[     k/(J*R),     -k^3/(J^2*R^2),                         k^5/(J^3*R^3),                              -k^7/(J^4*R^4),                                               k^9/(J^5*R^5),                                                                            -k^11/(J^6*R^6)]
[           0,                  0,                                     0,                                           0,                                                           0,                                                                                          0]


ans =

4
```

Part (b)

Is the system controllable using two motors?

Answer: No it is not controllable because the rank is of the controllability matrix is less than 6.

```
co2 = controllabilityMatrix(A,B)
rank(co2)
```

```
co2 =

[           0,           0,        (k*r)/(M*R),                  0,                  -(k^3*r)/(J*M*R^2),                                    0,       (k^5*r)/(J^2*M*R^3) - (K*k*r)/(M^2*R),                             (K*k*r)/(M^2*R),                                              (K*k^3*r)/(J*M^2*R^2) - (k^7*r)/(J^3*M*R^4),                                                                   -(K*k^3*r)/(J*M^2*R^2),              (K^2*k*r)/(M^3*R) - (K*((k^5*r)/(J^2*M*R^3) - (K*k*r)/(M^2*R)))/M + (k^
9*r)/(J^4*M*R^5),                 (K*((k^5*r)/(J^2*M*R^3) - (K*k*r)/(M^2*R)))/M - (K^2*k*r)/(M^3*R)]
[ (k*r)/(M*R),           0, -(k^3*r)/(J*M*R^2),                  0, (k^5*r)/(J^2*M*R^3) - (K*k*r)/(M^2*R),                      (K*k*r)/(M^2*R), (K*k^3*r)/(J*M^2*R^2) - (k^7*r)/(J^3*M*R^4),                      -(K*k^3*r)/(J*M^2*R^2), (K^2*k*r)/(M^3*R) - (K*((k^5*r)/(J^2*M*R^3) - (K*k*r)/(M^2*R)))/M + (k^9*r)/(J^4*M*R^5),                        (K*((k^5*r)/(J^2*M*R^3) - (K*k*r)/(M^2*R)))/M - (K^2*k*r)/(M^3*R), (K*((k^7*r)/(J^3*M*R^4) - (K*k^3*r)/(J*M^2*R^2)))/M - (k^11*r)/(J^5*M*R^6) - (K^2*k^
3*r)/(J*M^3*R^2),             (K^2*k^3*r)/(J*M^3*R^2) - (K*((k^7*r)/(J^3*M*R^4) - (K*k^3*r)/(J*M^2*R^2)))/M]
[           0,           0,                  0,        (k*r)/(M*R),                                   0,                   -(k^3*r)/(J*M*R^2),                             (K*k*r)/(M^2*R),       (k^5*r)/(J^2*M*R^3) - (K*k*r)/(M^2*R),                                                   -(K*k^3*r)/(J*M^2*R^2),                              (K*k^3*r)/(J*M^2*R^2) - (k^7*r)/(J^3*M*R^4),                                 (K*((k^5*r)/(J^2*M*R^3) - (K*k*r)/(M^2*R)))/M - (
K^2*k*r)/(M^3*R),           (K^2*k*r)/(M^3*R) - (K*((k^5*r)/(J^2*M*R^3) - (K*k*r)/(M^2*R)))/M + (k^9*r)/(J^4*M*R^5)]
[           0, (k*r)/(M*R),                  0, -(k^3*r)/(J*M*R^2),                     (K*k*r)/(M^2*R), (k^5*r)/(J^2*M*R^3) - (K*k*r)/(M^2*R),                      -(K*k^3*r)/(J*M^2*R^2), (K*k^3*r)/(J*M^2*R^2) - (k^7*r)/(J^3*M*R^4),                        (K*((k^5*r)/(J^2*M*R^3) - (K*k*r)/(M^2*R)))/M - (K^2*k*r)/(M^3*R), (K^2*k*r)/(M^3*R) - (K*((k^5*r)/(J^2*M*R^3) - (K*k*r)/(M^2*R)))/M + (k^9*r)/(J^4*M*R^5),                     (K^2*k^3*r)/(J*M^3*R^2) - (K*((k^7*r)/(J^3*M*R^4) - (K*k^3*r)
/(J*M^2*R^2)))/M, (K*((k^7*r)/(J^3*M*R^4) - (K*k^3*r)/(J*M^2*R^2)))/M - (k^11*r)/(J^5*M*R^6) - (K^2*k^3*r)/(J*M^3*R^2)]
[     k/(J*R),           0,     -k^3/(J^2*R^2),                  0,                       k^5/(J^3*R^3),                                    0,                              -k^7/(J^4*R^4),                                           0,                                                            k^9/(J^5*R^5),                                                                        0,
 -k^11/(J^6*R^6),                                                                                          0]
[           0,     k/(J*R),                  0,     -k^3/(J^2*R^2),                                   0,                        k^5/(J^3*R^3),                                           0,                              -k^7/(J^4*R^4),                                                                        0,                                                            k^9/(J^5*R^5),
               0,                                                                            -k^11/(J^6*R^6)]


ans =

5
```

Part (c)

Is the system observable if only z_1 is measured?

Answer It is not observable becuase the rank of the observability matrix is less thank 6.

```
C = sym([1 0 0 0 0 0]);

ob1 = observabilityMatrix(A,C)
rank(ob1)
```

```
ob1 =

[           1,           0,            0,            0,                                                        0,                     0]
[           0,           1,            0,            0,                                                        0,                     0]
[        -K/M,           0,          K/M,            0,                                       -(k^2*r)/(M*R),                     0]
[           0,        -K/M,            0,          K/M,                                     (k^4*r)/(J*M*R^2),                     0]
[ (2*K^2)/M^2,           0, -(2*K^2)/M^2,            0,                (K*k^2*r)/(M^2*R) - (k^6*r)/(J^2*M*R^3),    -(K*k^2*r)/(M^2*R)]
[           0, (2*K^2)/M^2,            0, -(2*K^2)/M^2, -(k^2*((K*k^2*r)/(M^2*R) - (k^6*r)/(J^2*M*R^3)))/(J*R), (K*k^4*r)/(J*M^2*R^2)]


ans =

5
```

Part (d)

Is the system observable if only $\dot{z}_1$ is measured?

Answer: It is not observable becuase the rank of the observability matrix is less thank 6.

```
C = sym([0 1 0 0 0 0]);
ob = observabilityMatrix(A,C)
```

```
rank(ob)
```

```
ob2 =

[            0,            1,            0,            0,                                                                               0,                                                  0]
[         -K/M,            0,          K/M,            0,                                                                  -(k^2*r)/(M*R),                                                  0]
[            0,         -K/M,            0,          K/M,                                                               (k^4*r)/(J*M*R^2),                                                  0]
[  (2*K^2)/M^2,            0, -(2*K^2)/M^2,            0,                                           (K*k^2*r)/(M^2*R) - (k^6*r)/(J^2*M*R^3),                              -(K*k^2*r)/(M^2*R)]
[            0,  (2*K^2)/M^2,            0, -(2*K^2)/M^2,                               -(k^2*((K*k^2*r)/(M^2*R) - (k^6*r)/(J^2*M*R^3)))/(J*R),                           (K*k^4*r)/(J*M^2*R^2)]
[ -(4*K^3)/M^3,            0,  (4*K^3)/M^3,            0, (k^4*((K*k^2*r)/(M^2*R) - (k^6*r)/(J^2*M*R^3)))/(J^2*R^2) - (2*K^2*k^2*r)/(M^3*R), (2*K^2*k^2*r)/(M^3*R) - (K*k^6*r)/(J^2*M^2*R^3)]


ans =

4
```

Part (e)

Is the system observable if only $\dot{z}_1$ and $\dot{z}_2$ are measured?

Answer it is not observable becuase the rank of the observability matrix is less thank 6.

```
C = sym([0 1 0 0 0 0;
    0 0 0 1 0 0]);
ob = observabilityMatrix(A,C)
rank(ob)
```

```
ob3 =

[            0,            1,            0,            0,                                                                               0,                                                                               0]
[            0,            0,            0,            1,                                                                               0,                                                                               0]
[         -K/M,            0,          K/M,            0,                                                                  -(k^2*r)/(M*R),                                                                               0]
[          K/M,            0,         -K/M,            0,                                                                               0,                                                                  -(k^2*r)/(M*R)]
[            0,         -K/M,            0,          K/M,                                                               (k^4*r)/(J*M*R^2),                                                                               0]
[            0,          K/M,            0,         -K/M,                                                                               0,                                                               (k^4*r)/(J*M*R^2)]
[  (2*K^2)/M^2,            0, -(2*K^2)/M^2,            0,                                           (K*k^2*r)/(M^2*R) - (k^6*r)/(J^2*M*R^3),                                                              -(K*k^2*r)/(M^2*R)]
[ -(2*K^2)/M^2,            0,  (2*K^2)/M^2,            0,                                                              -(K*k^2*r)/(M^2*R),                                           (K*k^2*r)/(M^2*R) - (k^6*r)/(J^2*M*R^3)]
[            0,  (2*K^2)/M^2,            0, -(2*K^2)/M^2,                               -(k^2*((K*k^2*r)/(M^2*R) - (k^6*r)/(J^2*M*R^3)))/(J*R),                                                           (K*k^4*r)/(J*M^2*R^2)]
[            0, -(2*K^2)/M^2,            0,  (2*K^2)/M^2,                                                           (K*k^4*r)/(J*M^2*R^2),                               -(k^2*((K*k^2*r)/(M^2*R) - (k^6*r)/(J^2*M*R^3)))/(J*R)]
[ -(4*K^3)/M^3,            0,  (4*K^3)/M^3,            0, (k^4*((K*k^2*r)/(M^2*R) - (k^6*r)/(J^2*M*R^3)))/(J^2*R^2) - (2*K^2*k^2*r)/(M^3*R),                                 (2*K^2*k^2*r)/(M^3*R) - (K*k^6*r)/(J^2*M^2*R^3)]
[  (4*K^3)/M^3,            0, -(4*K^3)/M^3,            0,                                 (2*K^2*k^2*r)/(M^3*R) - (K*k^6*r)/(J^2*M^2*R^3), (k^4*((K*k^2*r)/(M^2*R) - (k^6*r)/(J^2*M*R^3)))/(J^2*R^2) - (2*K^2*k^2*r)/(M^3*R)]


ans =

5
```

```
end % end function
```

Controllability Matrix Function

```
function co  = controllabilityMatrix(a,b)
n = size(a,1);
nu = size(b,2);
co = sym(zeros(n,n*nu));
co(:,1:nu) = b;
for k=1:n-1
    co(:,k*nu+1:(k+1)*nu) = a * co(:,(k-1)*nu+1:k*nu);
end


end % end funciton
```

Observability Matrix Function

```
function ob = observabilityMatrix(a,c)
n = size(a,1);
ny = size(c,1);
ob = sym(zeros(n*ny,n));
ob(1:ny,:) = c;
for k=1:n-1
    ob(k*ny+1:(k+1)*ny,:) = ob((k-1)*ny+1:k*ny,:) * a;
end
end % end function
```

Published with MATLAB® R2014a

2.6 from Barl

$$\dot{\vec{x}} = \begin{bmatrix} -2 & 1 \\ 1 & -2 \end{bmatrix} \vec{x} + \begin{bmatrix} 2 & 0 \\ 0 & 4 \end{bmatrix} \vec{u}$$

$$\vec{y} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \vec{x} + 0\vec{u}$$

(a) Compute Transfer function matrix.

$$G(s) = C(sI - A)^{-1}B + D$$

$$G(s) = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{pmatrix} s+2 & -1 \\ -1 & s+2 \end{pmatrix}^{-1} \begin{bmatrix} 2 & 0 \\ 0 & 4 \end{bmatrix}$$

$$G(s) = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} s+2 & 1 \\ 1 & s+2 \end{bmatrix} \frac{1}{(s+2)^2-1} \begin{bmatrix} 2 & 0 \\ 0 & 4 \end{bmatrix}$$

$$G(s) = \begin{bmatrix} s+3 & s+3 \\ s+1 & -s-1 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 4 \end{bmatrix} \frac{1}{(s+2)^2-1}$$

$$G(s) = \begin{bmatrix} 2/(s+1) & 4/(s+1) \\ 2/(s+3) & -4/(s+3) \end{bmatrix}$$

(b) Compute the poles, zeros, and modes.

$$(s+2)^2 - 1 = 0$$

$$(s+2)^2 = 1$$

$$s+2 = \pm 1$$

$s = -3, -1$ are the poles

zeros:

$$\det\left(\left[\begin{array}{c|c} sI - A & -B \\ \hline C & D \end{array}\right]\right) = 0$$

$$\det\left(\begin{bmatrix} s+2 & -1 & -2 & 0 \\ -1 & s+2 & 0 & -4 \\ 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix}\right) = 0$$

$$-16 \neq 0$$

no zeros

modes: e^{-3t}, e^{-t}

(c) Compute the impulse response matrix. How does this relate to the poles and modes?

$$g(t) = \mathcal{L}^{-1}\{G(s)\} = Ce^{At}B$$

$$g(t) = \begin{bmatrix} 2e^{-t} & 4e^{-t} \\ 2e^{-3t} & -4e^{-3t} \end{bmatrix}$$

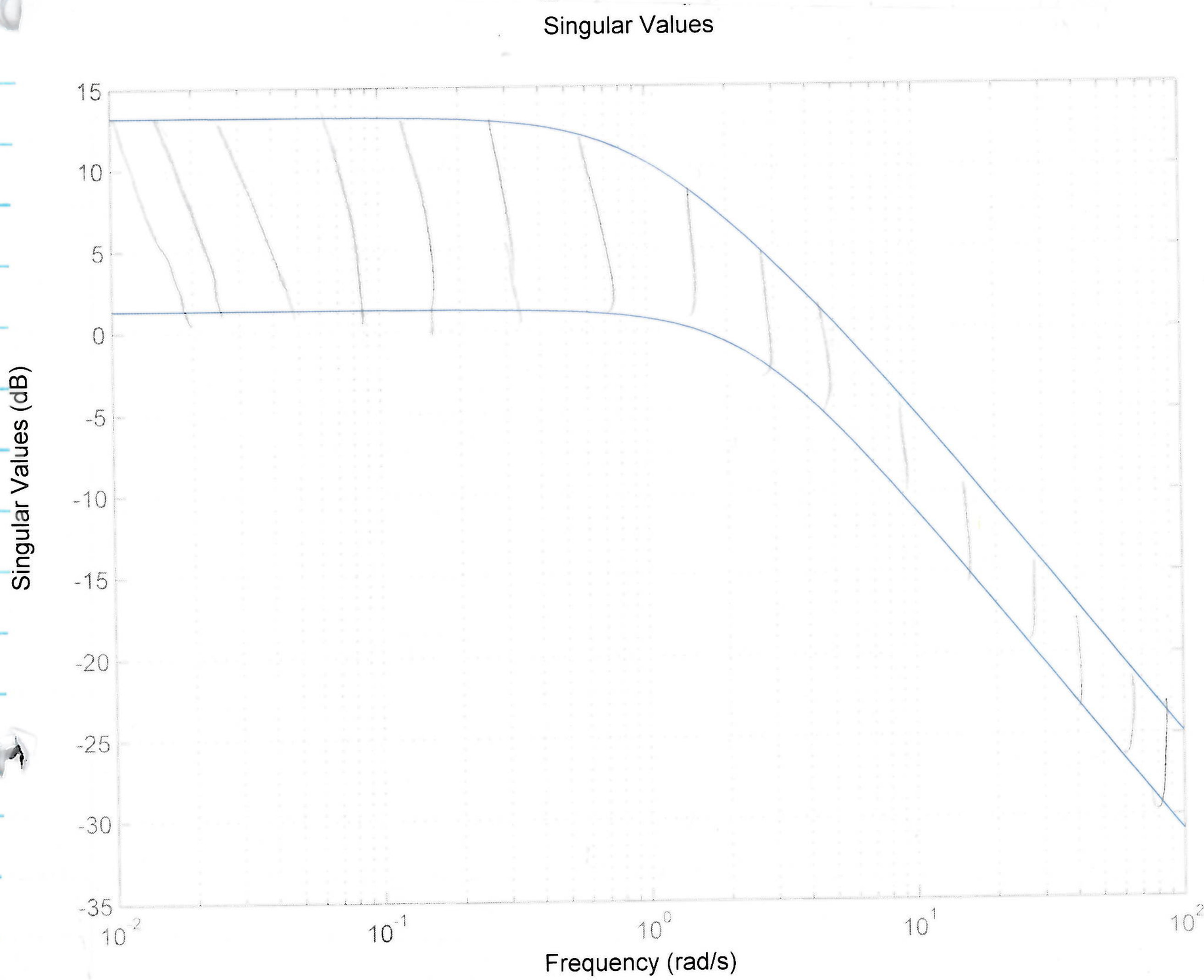
The modes are contained in the impulse response matrix. The poles define the modes

(d) Compute the frequency response matrix and the principal gains.

$$G(j\omega) = \begin{bmatrix} \frac{2}{(j\omega+1)} & \frac{4}{(j\omega+1)} \\ \frac{2}{(j\omega+3)} & \frac{-4}{(j\omega+3)} \end{bmatrix}$$

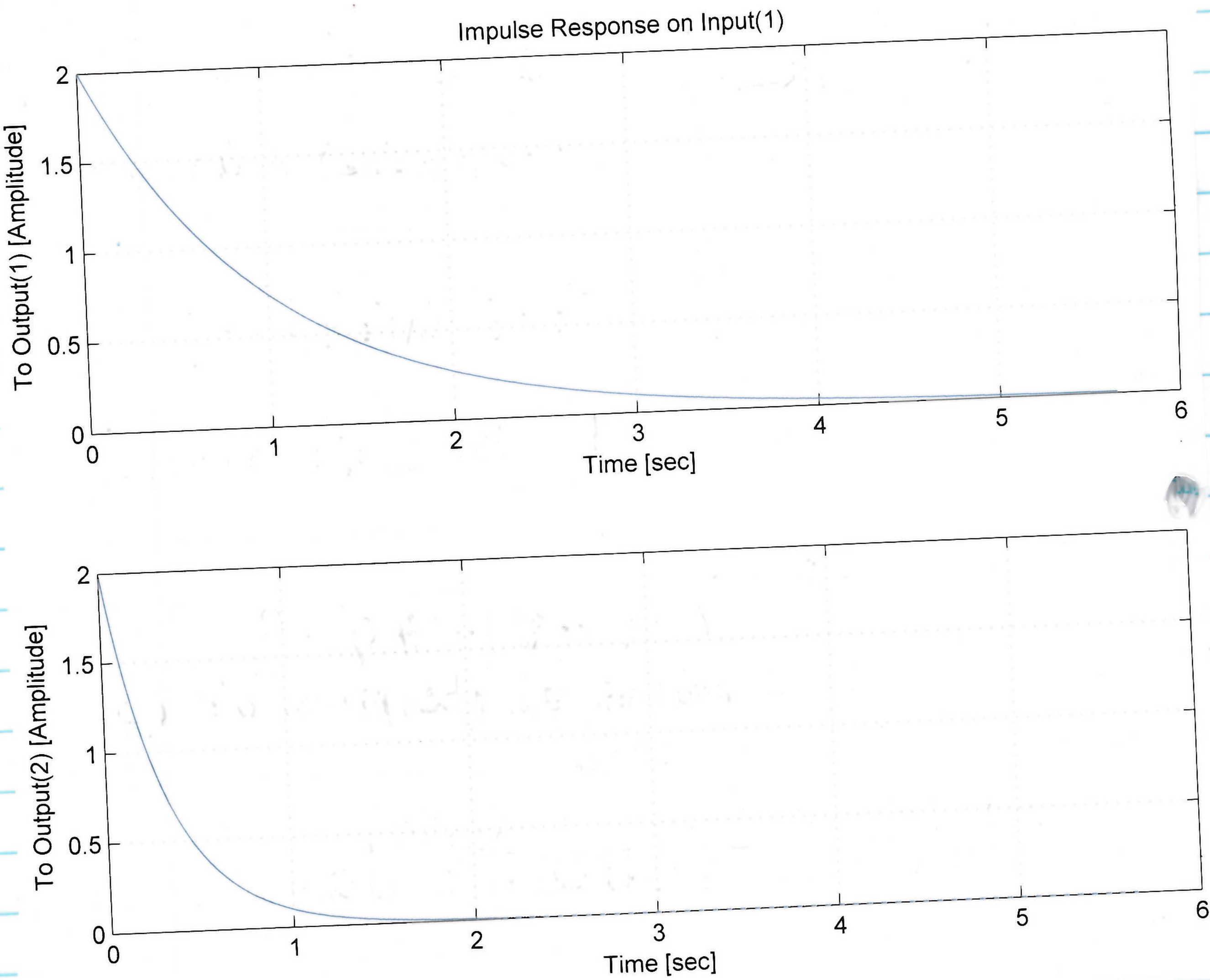
$$|G(j\omega)| = \begin{vmatrix} \frac{2}{\sqrt{\omega^2+1}} & \frac{4}{\sqrt{\omega^2+1}} \\ \frac{2}{\sqrt{\omega^2+9}} & \frac{4}{\sqrt{\omega^2+9}} \end{vmatrix}$$

$$\det\left(\lambda I - G(j\omega)\cdot G(j\omega)^*\right) = \text{principal gains}$$


Singular Values
Singular Values (dB)
Frequency (rad/s)
15
10
5
0
-5
-10
-15
-20
-25
-30
-35
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}

Compute the impulse response applied to u_1

(e) $y = \mathcal{L}^{-1}\{G(s)U(s)\} = \mathcal{L}^{-1}\left\{\begin{bmatrix}\frac{2}{s+1}\\ \frac{2}{s+3}\end{bmatrix}\right\} = \begin{bmatrix}2e^{-t}\\ 2e^{-3t}\end{bmatrix}$



(f) Compute the output and the gain when the input is $u(t) = [0 \ \cos(t)]^T$. How does this compare to the principal gains?

$$y = \mathcal{L}^{-1}\left\{ G(s) \cdot \begin{bmatrix} 0 \\ \frac{s}{s^2+\omega^2} \end{bmatrix} \right\} = \begin{bmatrix} 2\cos(t) + 2\sin(t) - 2e^{-t} \\ -\frac{6}{5}\cos(t) - \frac{2}{5}\sin(t) + \frac{6}{5}e^{-3t} \end{bmatrix}$$

$$y = \begin{bmatrix} 2\sqrt{2}\sin(t + \frac{\pi}{4}) - 2e^{-t} \\ -\frac{2\sqrt{10}}{5}\cos(t - \arctan(\frac{1}{3})) + \frac{6}{5}e^{-3t} \end{bmatrix}$$

The gains of y are within the bounds of the principal gains at $\omega = 1$ rad/sec

$(\det(\lambda I - G(i)G(i)^*))^{1/2}$ defines the principal gains.

$$\text{Principal min Gain} = \frac{\sqrt{15 - \sqrt{29}\sqrt{5}}\sqrt{5}\sqrt{2}}{5} \approx 1.088$$

$$\text{Principal max Gain} = \frac{\sqrt{\sqrt{29}\sqrt{5} + 15}\sqrt{5}\sqrt{2}}{5} \approx 3.289$$

Gain of $y_1 \approx 2.83$

Gain of $y_2 \approx 1.26$

$\sqrt{2.83^2 + 1.26^2} = 3.098$ is within the principal gains

(g) Compute the output $y_2(t)$ when the input is $u(t) = [\cos(t)\ 0]^T$. Compute the gain from this input to the output y_2 and compare to the principal gains. Does this make sense?

$$y_2 = \mathcal{L}^{-1}\left\{\frac{2}{(s+3)} \cdot \frac{s}{s^2+1}\right\} = \mathcal{L}^{-1}\left\{\frac{3}{5}\frac{s}{s^2+1} + \frac{1}{5}\frac{1}{s^2+1} - \frac{3}{5}\frac{1}{s+3}\right\}$$

$$y_2 = \frac{3}{5}\cos(t) + \frac{1}{5}\sin(t) - \frac{3}{5}e^{-3t}$$

$$y_2 = \frac{\sqrt{10}}{5}\cos\left(t - \arctan\left(\frac{1}{3}\right)\right) - \frac{3}{5}e^{-3t}$$

$$\text{gain of } y_2 = \frac{\sqrt{10}}{5} \approx 0.632$$

You should not compare it to the principal gains because the gain of y_1 is not included. Therefore no relevant comparison should be made

2.7 Burl

I am going to do each part with MATLAB but first I will describe the process the meaning.

Finding principal gains is the same as varying ω while calculating the minimum and maximum singular values of transfer function matrice.

Problem 2.7 from Burl

Table of Contents

Calculate the principal gains for the following systems. Also, for each of these systems, find the input that results in the maximum gain at a frequency of 2[rad/sec].

Part (a)

Note that the number of singular values is equal to the lesser of the number of columns or rows.

```
G = [tf([1 2],[1 5 4]);tf(2,[1 2 5])]
sigma(G)
grid('on')
```

```
G =

  From input to output...
          s + 2
   1:  -------------
       s^2 + 5 s + 4

            2
   2:  -------------
       s^2 + 2 s + 5

Continuous-time transfer function.
```

Problem 2.7 from Burl

Table of Contents

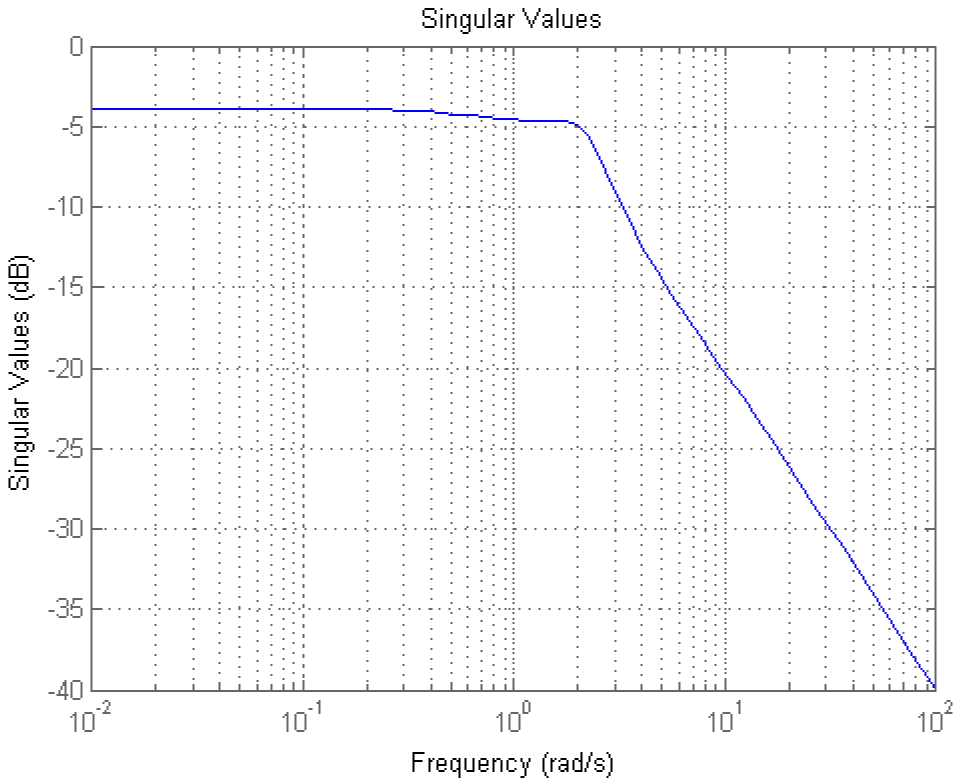
Calculate the principal gains for the following systems. Also, for each of these systems, find the input that results in the maximum gain at a frequency of 2[rad/sec].

Part (a)

Note that the number of singular values is equal to the lesser of the number of columns or rows.

```
G = [tf([1 2],[1 5 4]);tf(2,[1 2 5])]
sigma(G)
grid('on')
```

```
G =

  From input to output...
          s + 2
   1:  -------------
       s^2 + 5 s + 4

             2
   2:  -------------
       s^2 + 2 s + 5

Continuous-time transfer function.
```



The input that results in the maximum gain is

```
Gjw = evalfr(G, 2j);
[~,~,V] = svd(Gjw);
disp('Input vector magnitude = ')
disp(abs(V(:,1)))
disp('Input vector phase = ')
disp(phase(V(:,1)))
```

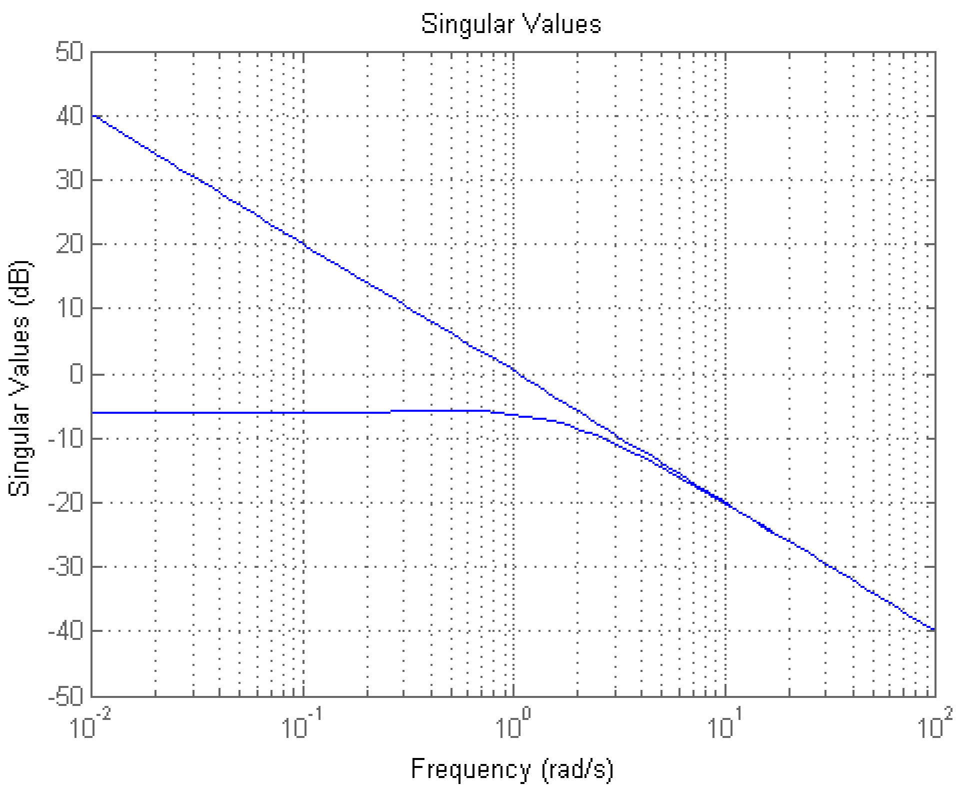
```
Input vector magnitude =
     1

Input vector phase =
     0
```

Part (b)

```
G = [tf(1,[1 1])  , tf(1,conv([1 1],[1 2]));
     tf(1,[1 1 0]), tf(1,[1 2])]
sigma(G)
grid('on')
```

```
G =
```

```
  From input 1 to output...
          1
   1:   -----
        s + 1

           1
   2:   -------
        s^2 + s

  From input 2 to output...
               1
   1:   -------------
        s^2 + 3 s + 2

          1
   2:   -----
        s + 2

Continuous-time transfer function.
```



The input that results in the maximum gain is

```
Gjw = evalfr(G, 2j);
[~,~,V] = svd(Gjw);
disp('Input vector magnitude = ')
disp(abs(V(:,1)))
```

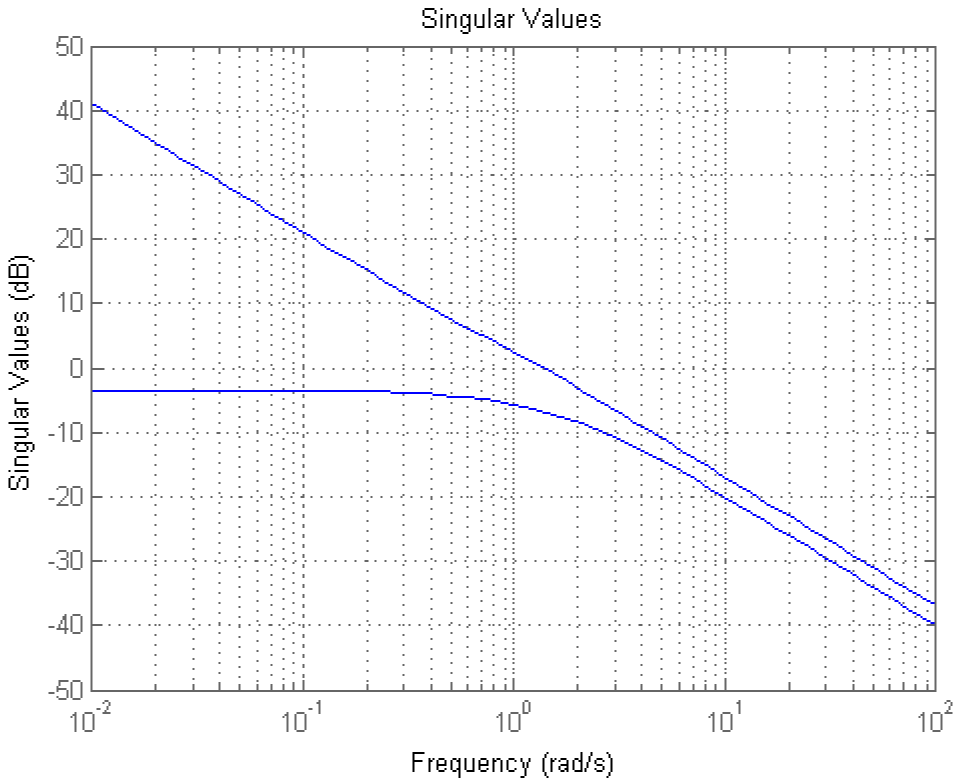
```
disp('Input vector phase = ')
disp(phase(V(:,1)))
```

```
Input vector magnitude =
    0.9530
    0.3029

Input vector phase =
   -3.1416
   -3.9270
```

Part (c)

```
G = [tf(1,[1 1]) , tf(1,conv([1 1],[1 2])), tf(1,[1 1]);
    tf(1,[1 1 0]), tf(1,[1 2])            , tf(1,[1 2 0])]
sigma(G)
grid('on')
```

```
G =

  From input 1 to output...
         1
   1:  -----
       s + 1

          1
   2:  -------
       s^2 + s

  From input 2 to output...
               1
   1:  -------------
       s^2 + 3 s + 2

         1
   2:  -----
       s + 2

  From input 3 to output...
         1
   1:  -----
       s + 1

           1
   2:  ---------
       s^2 + 2 s

Continuous-time transfer function.
```



The input that results in the maximum gain is

```
Gjw = evalfr(G, 2j);
[~,~,V] = svd(Gjw);
disp('Input vector magnitude = ')
disp(abs(V(:,1)))
disp('Input vector phase = ')
disp(phase(V(:,1)))
```

```
Input vector magnitude =
    0.7068
    0.1731
    0.6859

Input vector phase =
   -1.9622
   -2.4671
   -2.0251
```

Published with MATLAB® R2013b

Problem 2.10 Burl

Transform the following state Equations by T. Is the new system description simpler than the original?

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} -2 & 1 \\ 1 & -2 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 2 & 0 \\ 0 & -4 \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}$$

$$\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

$$T = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$$

$x = T^{-1} z$ sub

$$\begin{bmatrix} \dot{z}_1(t) \\ \dot{z}_2(t) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} -2 & 1 \\ 1 & -2 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix} + \begin{bmatrix} 2 & 0 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}$$

$$\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix}$$

$$\begin{bmatrix} \dot{z}_1(t) \\ \dot{z}_2(t) \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -3 \end{bmatrix} \begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix} + \begin{bmatrix} 2 & 0 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}$$

$$\begin{matrix} y_1(t) \\ y_2(t) \end{matrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix}$$

Yes, ~~the~~ the transformed system is now decoupled.

Table of Contents

Example 2.11 from Burl

```
A = [0 1;0 -1];
B = [0;1];
C = [1 0];
D = 0;
systemOpenLoop = ss(A,B,C,D);
desiredPoles = [-8 -8];
```

Observer Design

Note `place()` cannot be used because the rank of C is less than the number of repeated poles. Therefore `acker()` is used.

Observer gain matrix

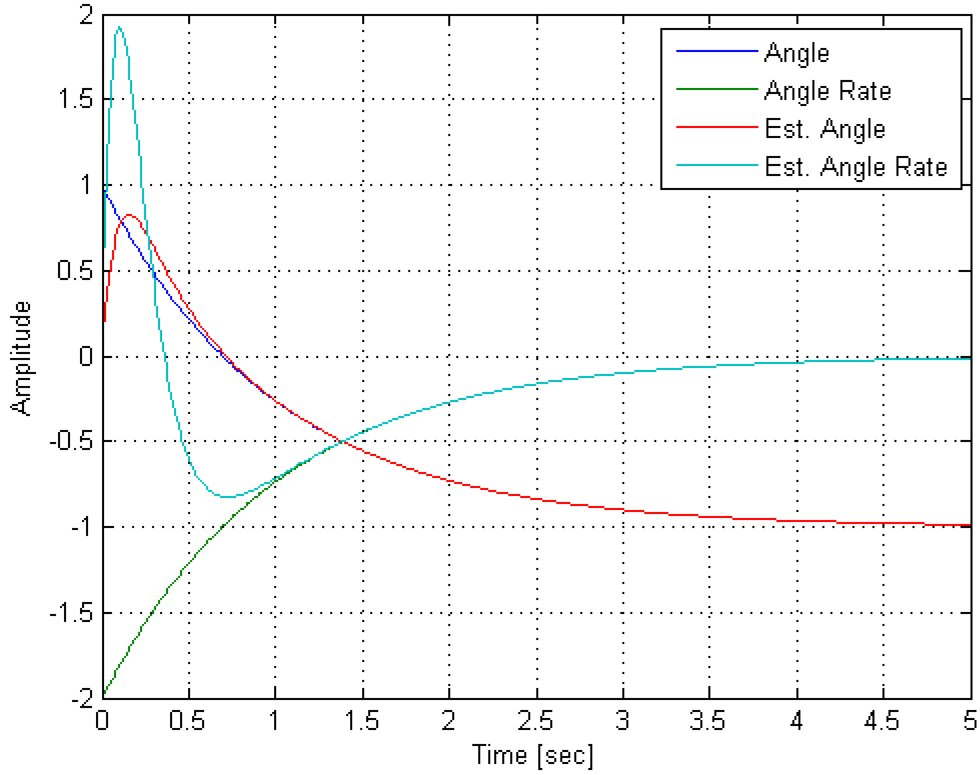
```
L = acker(A',C',desiredPoles)';
```

Construct the observer state equation

```
observer = ss([A, zeros(size(A));L*C, A-L*C],[B;B],[],[]);
```

Simulate the States and State Estimates

```
t = 0:0.01:5;
[~,~,x]=initial(observer,[1 -2 0 0]',t);
plot(t,x);
xlabel('Time [sec]');
ylabel('Amplitude');
legend('Angle', 'Angle Rate', 'Est. Angle', 'Est. Angle Rate');
grid('on');
```



Published with MATLAB® R2013b

2.13 from Burl

Given the following system

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} -2 & 0 \\ 0 & -2 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

determine

(a) is controllable?

(b) is observable?

Solution:

(a) Controllability Matrix $= \begin{bmatrix} 1 & -2 \\ 1 & -2 \end{bmatrix} \Rightarrow \text{rank} < 2$

$\Downarrow$

Not controllable

(b) Observability Matrix $= \begin{bmatrix} 1 & 0 \\ -2 & 0 \end{bmatrix} \Rightarrow \text{rank} < 2$

$\Downarrow$

Not observable

2.14 from Burl

Given the following model,

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -2 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} u(t)$$

$$\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix}$$

determine:

(a) is controllable?

(b) is observable?

(a) Controlability Matrix =

$$\begin{bmatrix} 0 & 0 & 1 & 0 & -2 & 1 \\ 1 & 0 & -2 & 1 & 4 & -1 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \Rightarrow \text{rank} = 3 \Downarrow \text{Controllable}$$

(b) Observability matrix

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -2 & 1 \\ 0 & 0 & 1 \\ 0 & 4 & -1 \end{bmatrix} \Rightarrow \text{rank} < 3 \Downarrow \text{not observable}$$

2.15 from Burl

Given:

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t)$$

$$m(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(t)$$

Find:

observer feedback controller to place poles at $p_1 = -2+2j$ and $p_2 = -2-2j$. The observer poles should be 4 times the dominant poles

Solution:

System is controllable.

$K = \begin{bmatrix} k_{11} & k_{12} \end{bmatrix}$ feedback gain matrix

Given the system above and the give poles, the following equations arise

$$\underbrace{\begin{matrix} k_{12} + 1 = 4 \\ k_{11} + 1 = 8 \end{matrix}}$$

$$\boxed{K = \begin{bmatrix} 7 & 3 \end{bmatrix}}$$

System is observable

Using 4 times given poles for the observer and the given state equations yields the following equations.

$$L = \begin{bmatrix} l_{11} \\ l_{21} \end{bmatrix}$$

$$l_{21} + 1 = 16$$
$$l_{11} + l_{21} + 1 = 128$$

$$\Downarrow$$

$$L = \begin{bmatrix} 15 \\ 112 \end{bmatrix}$$

$$\dot{\hat{x}} = \begin{bmatrix} -15 & 1 \\ -113 & -1 \end{bmatrix} \hat{x} + \begin{bmatrix} 15 \\ 112 \end{bmatrix} m - \begin{bmatrix} 0 & 0 \\ 7 & 3 \end{bmatrix} \hat{x}(1)$$

$$\Downarrow$$

$$\dot{\hat{x}} = \begin{bmatrix} -15 & 1 \\ -120 & -4 \end{bmatrix} \hat{x} + \begin{bmatrix} 15 \\ 112 \end{bmatrix} m$$

Computer Exercise 2.3 from Burl

Given:

$$\ddot{\theta} = -\frac{d}{M}\dot{\theta} - \frac{c}{M}\alpha + W$$

$$\dot{\alpha} = -0.1\alpha + 0.1\alpha_c$$

$M = 10^7\ kgm^2$

$d = 10^6\ \frac{N\text{-}m\text{-}sec}{rad}$

$c = 5000\ \frac{N\text{-}m}{rad}$

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{d}{M} & -\frac{c}{M} \\ 0 & 0 & -0.1 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ \alpha \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0.1 \end{bmatrix} \alpha_c + \begin{bmatrix} 0 \\ 1/M \\ 0 \end{bmatrix} W$$

Generate state feedback controllers to place the poles as follows:

a) $p_1 = p_2 = -0.02,\ p_3 = -0.1$

$$\cancel{\dot{\hat{x}} = (A - LC)\hat{x} + L\,m(t) + (B - LD)\,u(t)}$$

$$\cancel{\hat{x} = (A - LC}$$

See matlab published file for the rest

Computer Problem 2.3 from Burl

Clean up

```
clc;close('all');clear;
```

Given

```
M = 10^7; % kg-m^2
d = 10^6; % N-m-sec/rad
c = 5000; % N-m/rad
% System matrices
A = [0     1       0;
     0   -d/M    -c/M;
     0     0    -0.1];

B = [   0;
        0;
      0.1];

B_w = [0;
       1/M;
       0];

C = [1 0 0;
     0 0 1];

 % initial conditions
x0 = [0.1; 0; 0];
```

```
% Labels
stateNames = {'Heading Error', 'Heading Error derivative [rad/sec]', 'Rudder Angle'};
stateUnits = {'rad', 'rad/sec', 'rad'};
inputName = 'Rudder Angle Command';
inputUnit = 'rad';
disturbanceInputName = 'Disturbance Torque';
disturbanceInputUnit = 'N-m';
outputNames = {'Heading Error'; 'Rudder Angle'};
outputUnits = {'rad', 'rad'};
```

Form Open Loop System

```
openLoopSys = ss(A,[B B_w],C,0);
openLoopSys.Name = 'Open Loop System';
openLoopSys.InputName = {inputName, disturbanceInputName};
openLoopSys.InputUnit = {inputUnit, disturbanceInputUnit};
openLoopSys.StateName = stateNames;
openLoopSys.stateUnit = stateUnits;
openLoopSys.OutputName = outputNames;
openLoopSys.OutputUnit = outputUnits;
```

Form Empty Controller System

```
controllerSys = ss(zeros(size(A)), B_w, C, []);
controllerSys.InputName = disturbanceInputName;
controllerSys.InputUnit = disturbanceInputUnit;
controllerSys.StateName = stateNames;
controllerSys.StateUnit = stateUnits;
controllerSys.OutputName = outputNames;
controllerSys.OutputUnit = outputUnits;
```

Form Empty Observer System

```
observer = ss(zeros(size(A)*2),[B B_w;B B_w],[],[]);
observer.Name = 'Observer';
observer.StateName = [stateNames, cellfun(@(x) [x ' Estimate'],
stateNames,'UniformOutput',false)];
observer.StateUnit = [stateUnits, stateUnits];
observer.InputName = {inputName, disturbanceInputName};
observer.InputUnit = {inputUnit, disturbanceInputUnit};
```

Prepare Plots

```
initialConditionsHandle = figure;
set(gcf,'position', [1352,59,1031,1125]);
stepResposeHande = figure;
set(gcf,'position', [1352,59,1031,1125]);
```

Part (a) Design Controller

```
suffix = 'Part (a) Design Controller';
% poles
controllerPoles = [-0.02 -0.02 -0.1];
```

Gain matrix

```
disp(['Controller Gain Matrix for ' suffix])
K = acker(A,B,controllerPoles);
disp(K)
```

```
Controller Gain Matrix for Part (a) Design Controller
   -0.8000   -8.0000   -0.6000
```

Form the controller state equation

```
A_controller = A-B*K;
controllerSys.a = A_controller;
controllerSys.name = [ 'Controller System, ' suffix];
display(controllerSys);
```

```

controllerSys =

  a = 
                 Heading Erro  Heading Erro  Rudder Angle
   Heading Erro             0             1             0
   Heading Erro             0          -0.1       -0.0005
   Rudder Angle          0.08           0.8         -0.04

  b = 
                  Disturbance
   Heading Erro             0
   Heading Erro         1e-07
   Rudder Angle             0

  c = 
                 Heading Erro  Heading Erro  Rudder Angle
   Heading Erro             1             0             0
   Rudder Angle             0             0             1

  d = 
                  Disturbance
   Heading Erro             0
   Rudder Angle             0

Name: Controller System, Part (a) Design Controller
Continuous-time state-space model.
```

Simulate initial conditions

```
figure(initialConditionsHandle);
initial(controllerSys,x0);
grid('on');
hold('all');
```

Simulate step response in disturbance

```
stepOptions = stepDataOptions;
stepOptions.StepAmplitude = 1000;
figure(stepResposeHande);
step(controllerSys, stepOptions);
grid('on');
hold('all');
```

Part (b) Design Controller

```
suffix = 'Part (b) Design Controller';
% poles
controllerPoles = [-0.04 -0.04 -0.1];
```

Gain matrix

```
disp(['Controller Gain Matrix for ' 'Part (b) Design Controller'])
K = acker(A,B,controllerPoles);
disp(K)
```

```
Controller Gain Matrix for Part (b) Design Controller
   -3.2000  -32.0000   -0.2000
```

Form the controller state equation and simulate

```
A_controller = A-B*K;
controllerSys.a = A_controller;
controllerSys.name = [ 'Controller System, ' suffix];
display(controllerSys);
```

```

controllerSys =

  a =
                Heading Erro  Heading Erro  Rudder Angle
   Heading Erro             0             1             0
   Heading Erro             0          -0.1       -0.0005
   Rudder Angle          0.32           3.2         -0.08

  b =
                 Disturbance
   Heading Erro            0
   Heading Erro        1e-07
   Rudder Angle            0

  c =
                Heading Erro  Heading Erro  Rudder Angle
   Heading Erro             1             0             0
   Rudder Angle             0             0             1

  d =
                 Disturbance
   Heading Erro            0
   Rudder Angle            0

Name: Controller System, Part (b) Design Controller
Continuous-time state-space model.
```

simulate intial conditions

```
figure(initialConditionsHandle);
initial(controllerSys,x0);
```

Simulate step response in disturbance

```
figure(stepResposeHande);
step(controllerSys, stepOptions);
```

Part (c) Design Controller

```
suffix = 'Part (c) Design Controller';
% poles
controllerPoles = [-0.02+0.03j, -0.02-0.03j, -0.1];
```

Gain matrix

```
disp(['Controller Gain Matrix for ' 'Part (c) Design Controller'])
K = acker(A,B,controllerPoles);
disp(K)
```

```
Controller Gain Matrix for Part (c) Design Controller
Warning: Pole locations are more than 10% in error.
   -2.6000  -26.0000   -0.6000
```

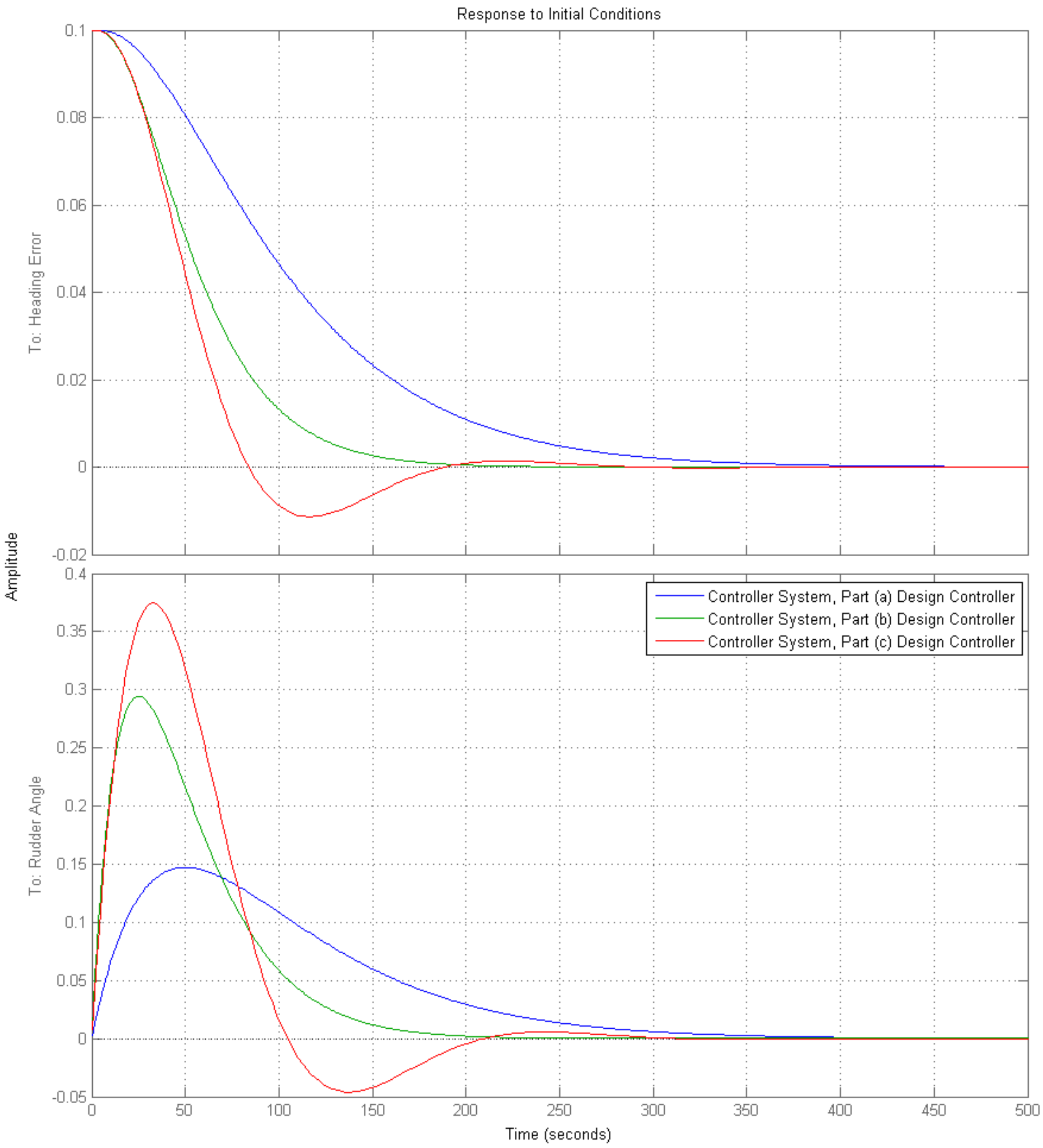
Form the controller state equation and simulate

```
A_controller = A-B*K;
controllerSys.a = A_controller;
controllerSys.name = [ 'Controller System, ' suffix];
display(controllerSys);
```

```
controllerSys =

  a =
                 Heading Erro  Heading Erro  Rudder Angle
   Heading Erro             0             1             0
   Heading Erro             0          -0.1       -0.0005
   Rudder Angle          0.26           2.6         -0.04

  b =
                  Disturbance
   Heading Erro             0
   Heading Erro         1e-07
   Rudder Angle             0

  c =
                 Heading Erro  Heading Erro  Rudder Angle
   Heading Erro             1             0             0
   Rudder Angle             0             0             1

  d =
                  Disturbance
   Heading Erro             0
   Rudder Angle             0

Name: Controller System, Part (c) Design Controller
Continuous-time state-space model.
```

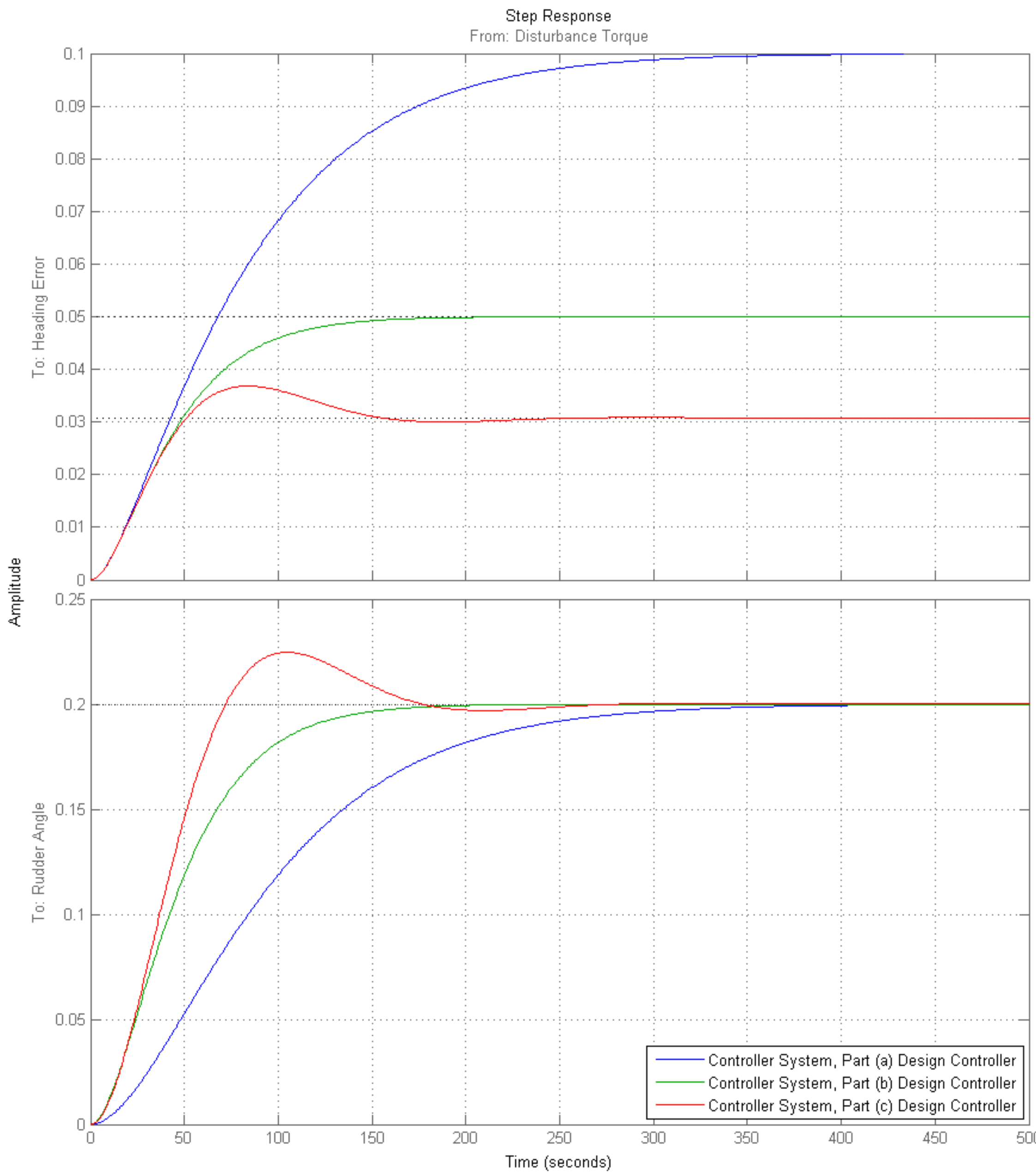
Simulate initial conditions

```
figure(initialConditionsHandle);
initial(controllerSys,x0);
legend('show');
```



Simulate step response in disturbance

```
figure(stepResposeHande);
step(controllerSys, stepOptions);
legendHandle = legend('show');
set(legendHandle,'Location','SouthEast');
```



Commentary on Controller Design

As the poles get further away from the imaginary axis, the faster the closed loop system responds and disturbance rejection is better. However, it is probably not practical in case c to have heading error overshoot. A ship should stay on course without overshoot to minimize fuel consumption and control energy.

Part (a) Design Observer

```
suffix = 'Part (a) Design Observer';
observerPoles = [-0.08 -0.08 -0.4];
disp(['Observer Gain Matrix for ' suffix])
L = place(A', C', observerPoles)';
disp(L);
```

```
Observer Gain Matrix for Part (a) Design Observer
    0.3800         0
   -0.0060   -0.0005
         0   -0.0200
```

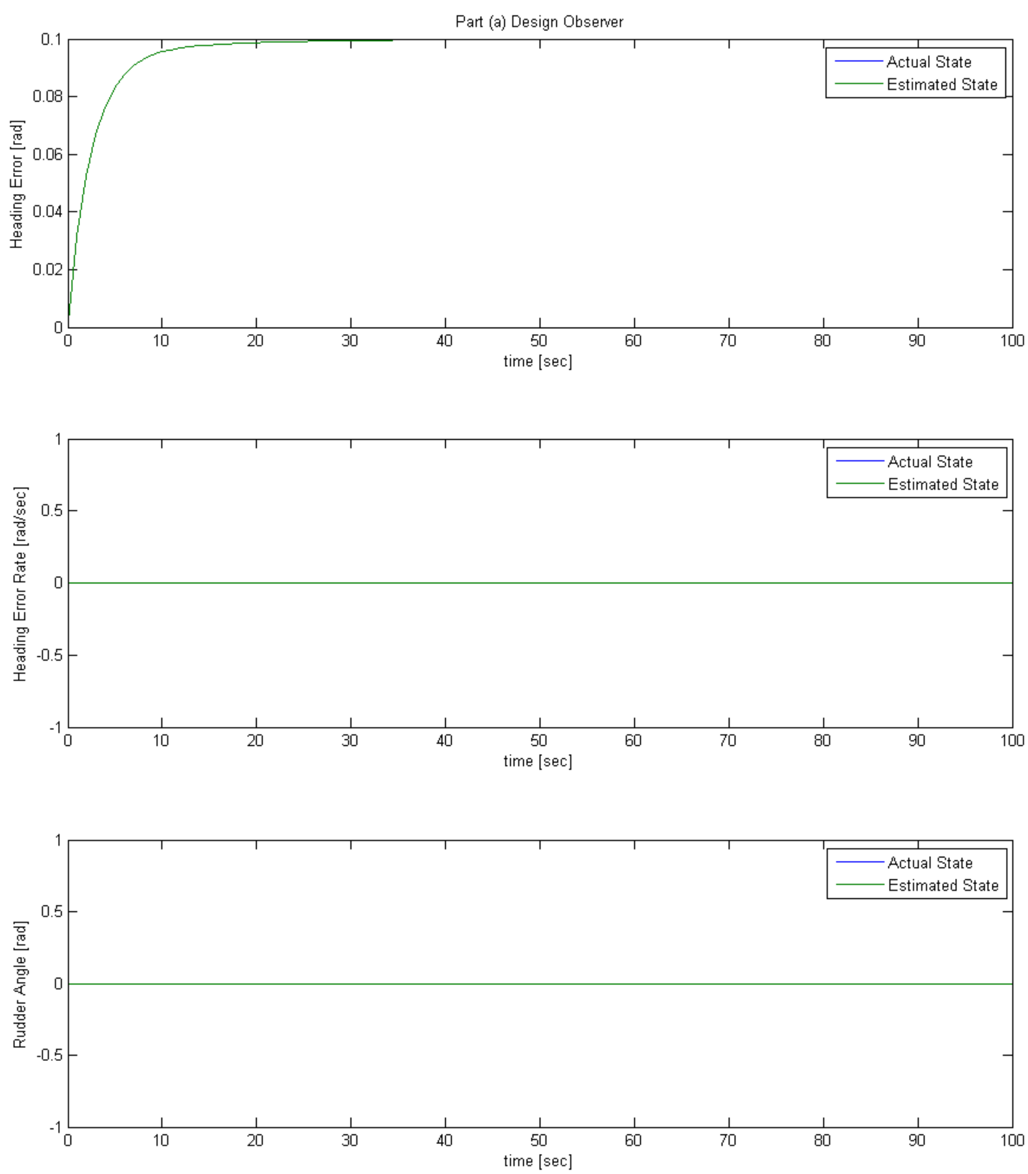
Create observer system

```
observer.a = [A, zeros(size(A));L*C, A-L*C];
display(observer)
```

```
observer =

  a =
              Heading Erro  Heading Erro  Rudder Angle  Heading Erro
   Heading Erro             0             1             0             0
   Heading Erro             0          -0.1       -0.0005             0
   Rudder Angle             0             0          -0.1             0
   Heading Erro          0.38             0             0         -0.38
   Heading Erro        -0.006             0       -0.0005         0.006
   Rudder Angle             0             0         -0.02             0

              Heading Erro  Rudder Angle
   Heading Erro             0             0
   Heading Erro             0             0
   Rudder Angle             0             0
   Heading Erro             1             0
   Heading Erro          -0.1             0
   Rudder Angle             0         -0.08

  b =
              Rudder Angle   Disturbance
   Heading Erro             0             0
   Heading Erro             0         1e-07
   Rudder Angle           0.1             0
   Heading Erro             0             0
   Heading Erro             0         1e-07
   Rudder Angle           0.1             0

  c =
     Empty matrix: 0-by-6

  d =
     Empty matrix: 0-by-2

Name: Observer
Continuous-time state-space model.
```

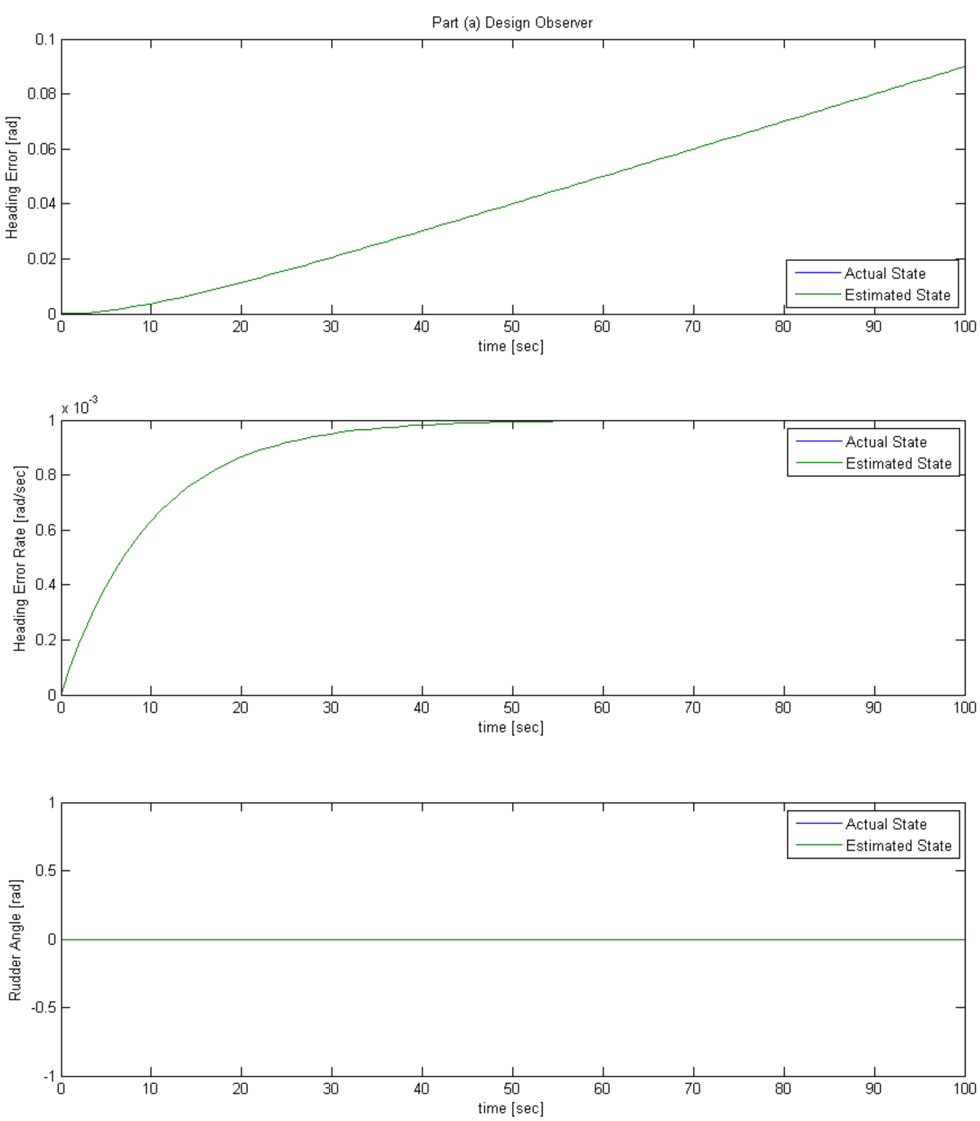
Simulate initial conditions and plot

```
[~,t,x] = initial(observer,[x0;zeros(size(x0))],100);
figure
set(gcf,'position', [1352,59,1031,1125]);
subplot(311)
plot(t,x(:,1),t,x(:,4));
xlabel('time [sec]');
ylabel('Heading Error [rad]');
legend('Actual State','Estimated State');
title(suffix)
subplot(312)
plot(t,x(:,2),t,x(:,2));
xlabel('time [sec]');
ylabel('Heading Error Rate [rad/sec]');
legend('Actual State','Estimated State');
subplot(313)
plot(t,x(:,3),t,x(:,6));
xlabel('time [sec]');
ylabel('Rudder Angle [rad]');
legend('Actual State','Estimated State');
```



Simulate disturbance and plot

```
[~,t,x] = lsim(observer,[t*0, 1000*t.^0],t);
figure
set(gcf,'position', [1352,59,1031,1125]);
subplot(311)
plot(t,x(:,1),t,x(:,4));
xlabel('time [sec]');
ylabel('Heading Error [rad]');
legendHandle = legend('Actual State','Estimated State');
set(legendHandle,'Location','SouthEast');
title(suffix)
subplot(312)
plot(t,x(:,2),t,x(:,2));
xlabel('time [sec]');
ylabel('Heading Error Rate [rad/sec]');
legend('Actual State','Estimated State');
subplot(313)
plot(t,x(:,3),t,x(:,6));
xlabel('time [sec]');
ylabel('Rudder Angle [rad]');
legend('Actual State','Estimated State');
```



Part (b) Design Observer

```
suffix = 'Part (b) Design Observer';
observerPoles = [-0.16 -0.16 -0.4];
disp(['Observer Gain Matrix for ' suffix])
L = place(A', C', observerPoles)';
disp(L);
```

```
Observer Gain Matrix for Part (b) Design Observer
    0.4600         0
    0.0180   -0.0005
         0    0.0600
```

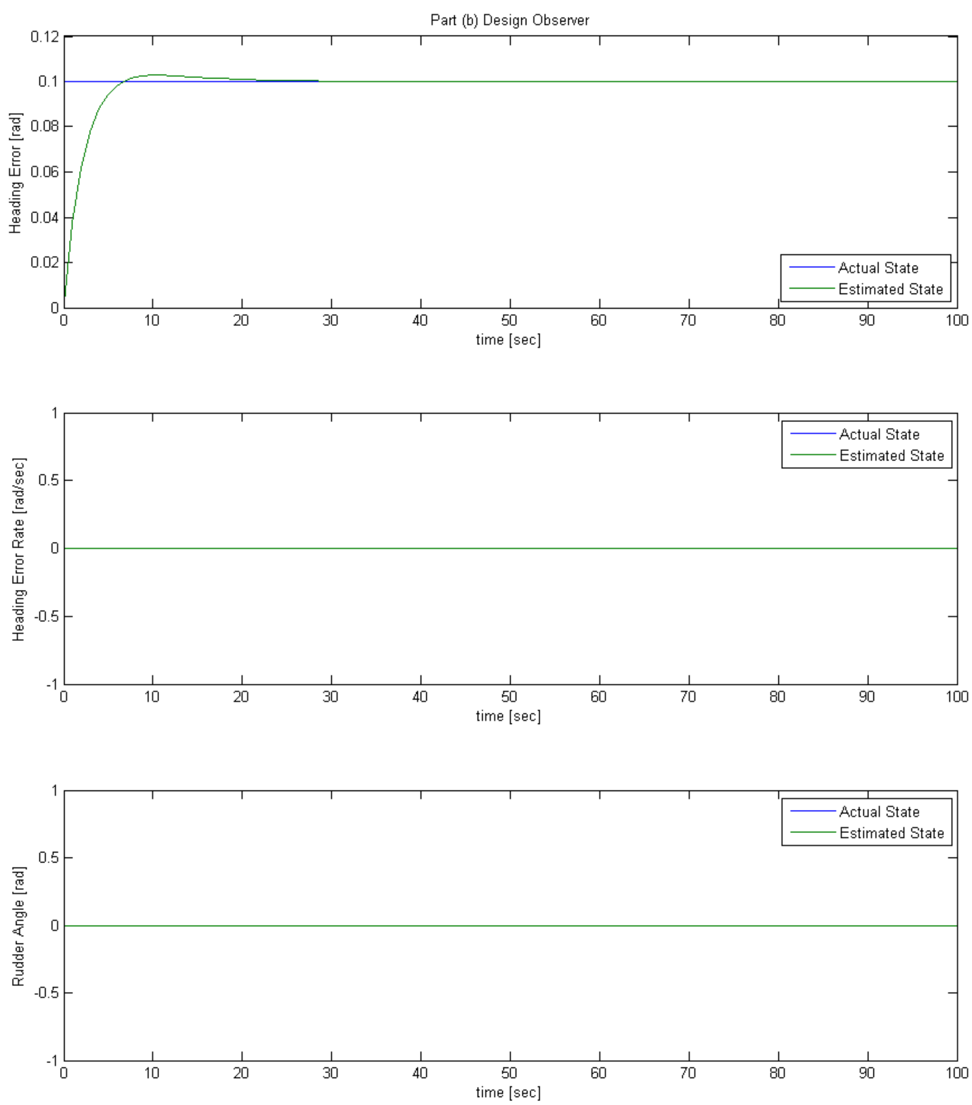
Create observer system

```
observer.a = [A, zeros(size(A));L*C, A-L*C];
display(observer)
```

```
observer =

  a =
               Heading Erro  Heading Erro  Rudder Angle  Heading Erro
   Heading Erro             0             1             0             0
   Heading Erro             0          -0.1       -0.0005             0
   Rudder Angle             0             0          -0.1             0
   Heading Erro          0.46             0             0         -0.46
   Heading Erro         0.018             0       -0.0005        -0.018
   Rudder Angle             0             0          0.06             0

               Heading Erro  Rudder Angle
   Heading Erro             0             0
   Heading Erro             0             0
   Rudder Angle             0             0
   Heading Erro             1             0
   Heading Erro          -0.1             0
   Rudder Angle             0         -0.16

  b =
               Rudder Angle   Disturbance
   Heading Erro             0             0
   Heading Erro             0         1e-07
   Rudder Angle           0.1             0
   Heading Erro             0             0
   Heading Erro             0         1e-07
   Rudder Angle           0.1             0

  c =
     Empty matrix: 0-by-6

  d =
     Empty matrix: 0-by-2

Name: Observer
Continuous-time state-space model.
```

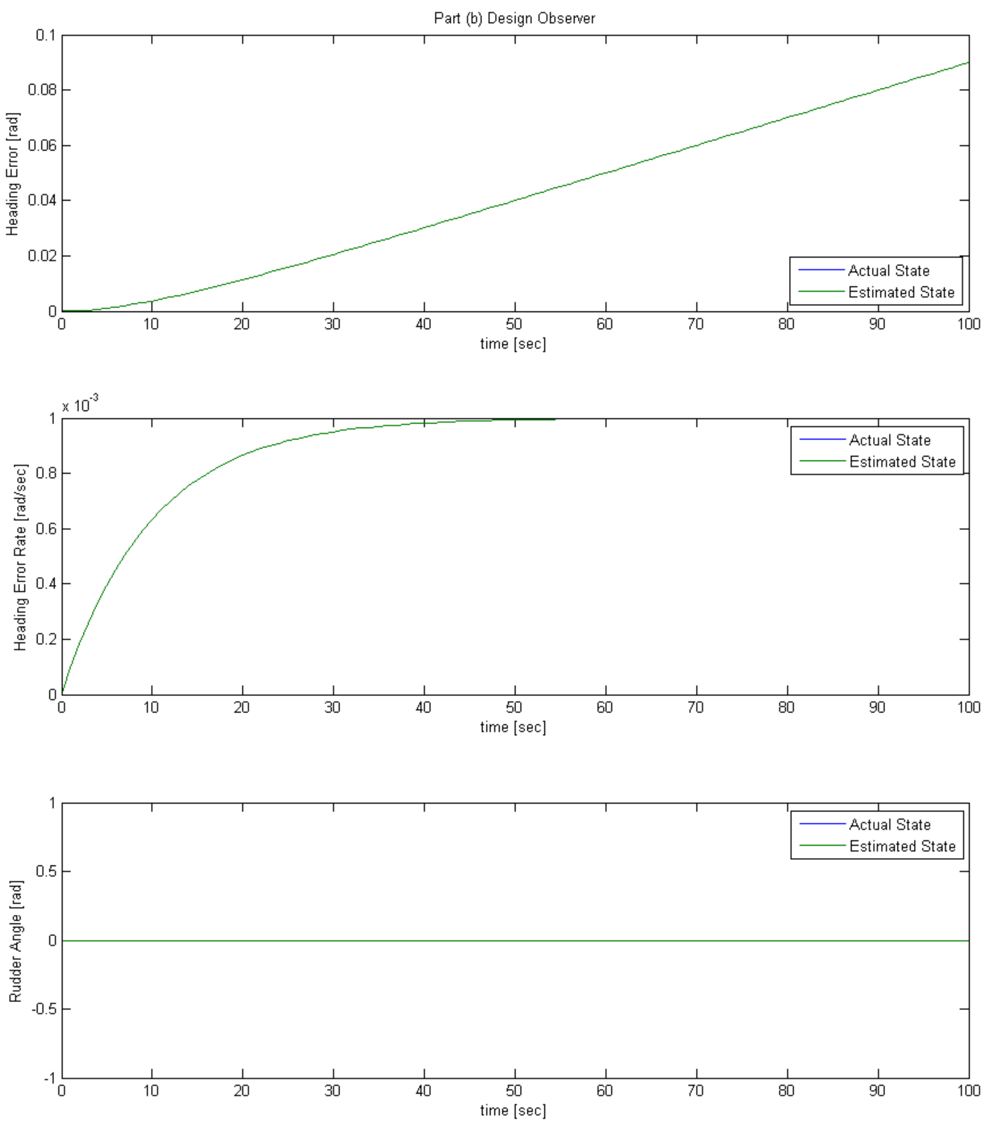
Simulate initial conditions and plot

```
[~,t,x] = initial(observer,[x0;zeros(size(x0))],100);
figure
set(gcf,'position', [1352,59,1031,1125]);
subplot(311)
plot(t,x(:,1),t,x(:,4));
xlabel('time [sec]');
ylabel('Heading Error [rad]');
legendHandle = legend('Actual State','Estimated State');
set(legendHandle,'Location','SouthEast');
title(suffix)
subplot(312)
plot(t,x(:,2),t,x(:,2));
xlabel('time [sec]');
ylabel('Heading Error Rate [rad/sec]');
legend('Actual State','Estimated State');
subplot(313)
plot(t,x(:,3),t,x(:,6));
xlabel('time [sec]');
ylabel('Rudder Angle [rad]');
legend('Actual State','Estimated State');
```



Simulate disturbance and plot

```
[~,t,x] = lsim(observer,[t*0, 1000*t.^0],t);
figure
set(gcf,'position', [1352,59,1031,1125]);
subplot(311)
plot(t,x(:,1),t,x(:,4));
xlabel('time [sec]');
ylabel('Heading Error [rad]');
legendHandle = legend('Actual State','Estimated State');
set(legendHandle,'Location','SouthEast');
title(suffix)
subplot(312)
plot(t,x(:,2),t,x(:,2));
xlabel('time [sec]');
ylabel('Heading Error Rate [rad/sec]');
legend('Actual State','Estimated State');
subplot(313)
plot(t,x(:,3),t,x(:,6));
xlabel('time [sec]');
ylabel('Rudder Angle [rad]');
legend('Actual State','Estimated State');
```



Commentary on Observer Design

The faster poles speed up convergence of the estimated states.

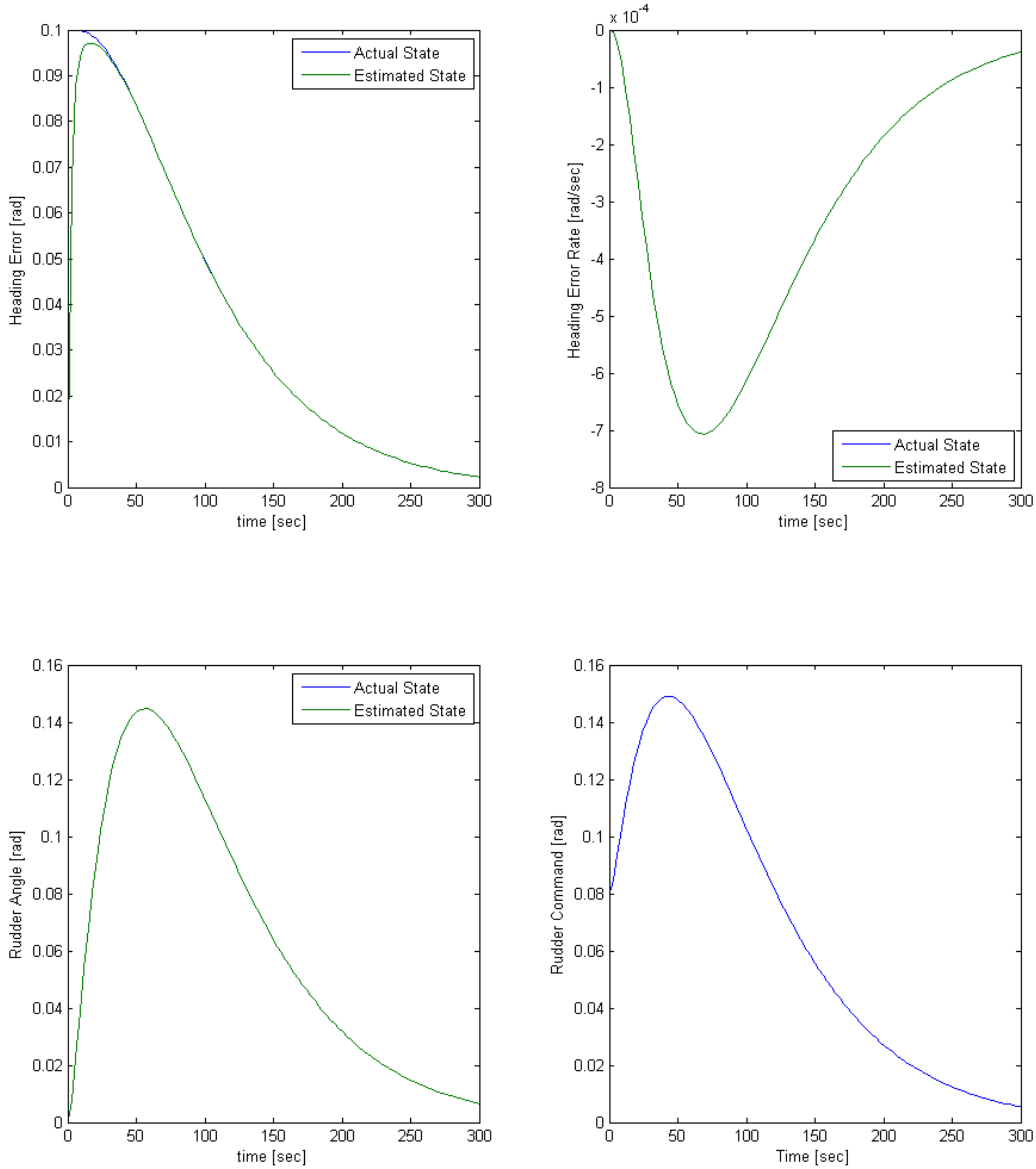
Combined Observer and Controller

```
controllerPoles = [-0.02 -0.02 -0.1];
observerPoles = [-0.08 -0.08 -0.4];
K = acker(A,B,controllerPoles);
L = place(A',C',observerPoles)';
regulator = ss([A, -B*K;L*C, A-L*C-B*K],[B_w;B_w],[],[]);
regulator.StateName = observer.StateName;
regulator.StateUnit = observer.StateUnit;
regulator.InputName = disturbanceInputName;
regulator.InputUnit = disturbanceInputUnit;
display(regulator)
```

```
regulator =

  a =
                Heading Erro  Heading Erro  Rudder Angle  Heading Erro
   Heading Erro             0             1             0             0
   Heading Erro             0          -0.1       -0.0005             0
   Rudder Angle             0             0          -0.1          0.08
   Heading Erro          0.38             0             0         -0.38
   Heading Erro        -0.006             0       -0.0005         0.006
   Rudder Angle             0             0         -0.02          0.08

                Heading Erro  Rudder Angle
   Heading Erro             0             0
   Heading Erro             0             0
   Rudder Angle           0.8          0.06
   Heading Erro             1             0
   Heading Erro          -0.1             0
   Rudder Angle           0.8         -0.02

  b =
                 Disturbance
   Heading Erro            0
   Heading Erro        1e-07
   Rudder Angle            0
   Heading Erro            0
   Heading Erro        1e-07
   Rudder Angle            0

  c =
     Empty matrix: 0-by-6

  d =
     Empty matrix: 0-by-1

Continuous-time state-space model.
```

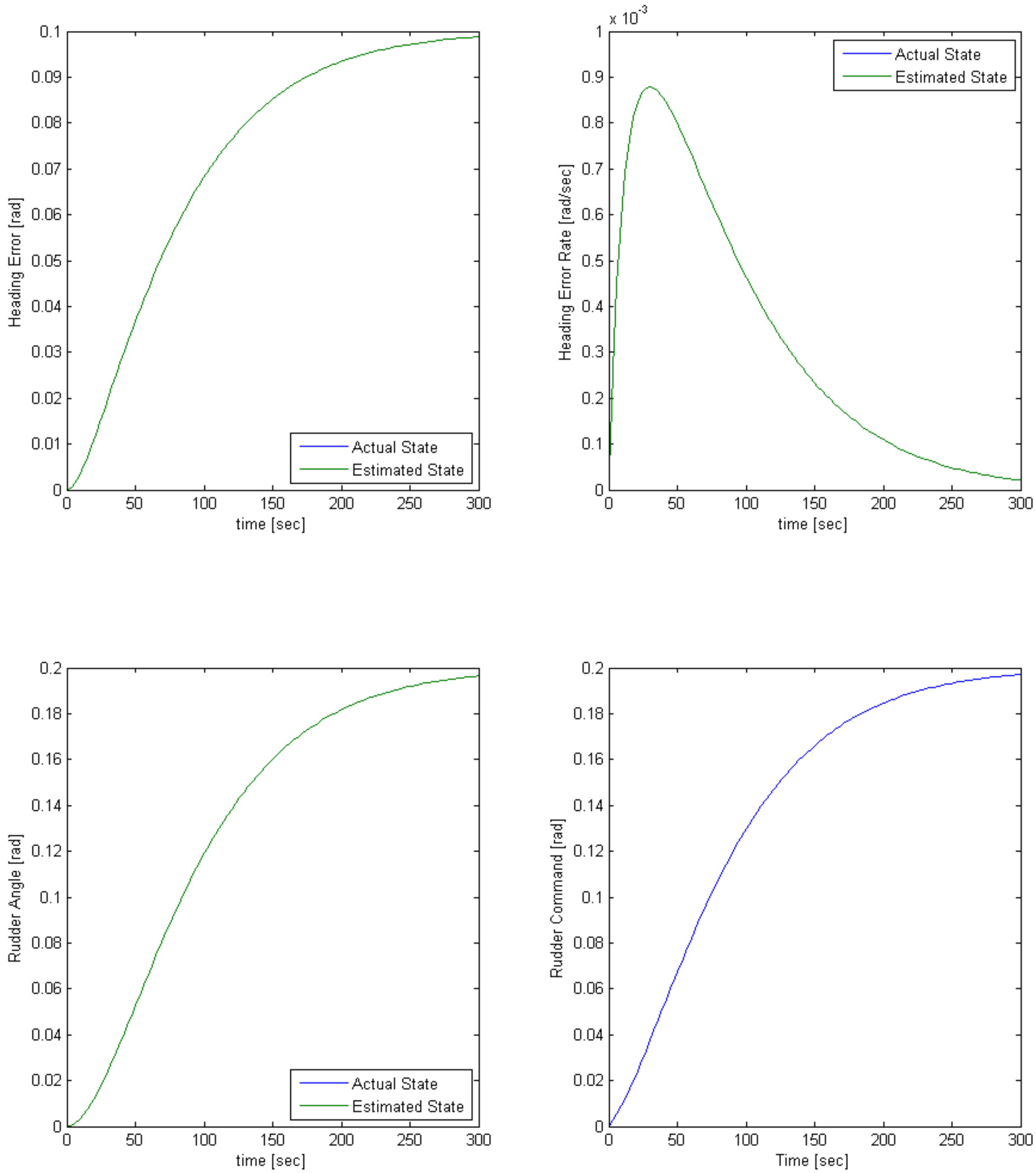
Simulate initial conditions and plot

```
x0 = [0.1; 0; 0; 0; 0; 0];

[~,t,x] = initial(regulator,x0,300);
rudderCommand = -K*x(:,1:3)';
figure('Name','Response to Initial Conditions')
set(gcf,'position', [1352,59,1031,1125]);
subplot(221)
plot(t,x(:,1),t,x(:,4));
xlabel('time [sec]');
ylabel('Heading Error [rad]');
legend('Actual State','Estimated State');
subplot(222)
plot(t,x(:,2),t,x(:,2));
xlabel('time [sec]');
ylabel('Heading Error Rate [rad/sec]');
legendHandle = legend('Actual State','Estimated State');
set(legendHandle,'Location','SouthEast');
subplot(223)
plot(t,x(:,3),t,x(:,6));
xlabel('time [sec]');
ylabel('Rudder Angle [rad]');
legend('Actual State','Estimated State');
subplot(224)
plot(t,rudderCommand)
ylabel('Rudder Command [rad]')
xlabel('Time [sec]')
```


Heading Error [rad]
time [sec]
Actual State
Estimated State
Heading Error Rate [rad/sec]
x 10^-4
time [sec]
Actual State
Estimated State
Rudder Angle [rad]
time [sec]
Actual State
Estimated State
Rudder Command [rad]
Time [sec]

Simulate disturbance and plot

```
[~,t,x] = lsim(regulator,1000*t.^0,t);
rudderCommand = -K*x(:,1:3)';
figure('Name','Response to Initial Conditions')
set(gcf,'position', [1352,59,1031,1125]);
subplot(221)
plot(t,x(:,1),t,x(:,4));
xlabel('time [sec]');
ylabel('Heading Error [rad]');
legendHandle = legend('Actual State','Estimated State');
set(legendHandle,'Location','SouthEast');
subplot(222)
plot(t,x(:,2),t,x(:,2));
xlabel('time [sec]');
ylabel('Heading Error Rate [rad/sec]');
legend('Actual State','Estimated State');
subplot(223)
plot(t,x(:,3),t,x(:,6));
xlabel('time [sec]');
ylabel('Rudder Angle [rad]');
legendHandle = legend('Actual State','Estimated State');
set(legendHandle,'Location','SouthEast');
subplot(224)
plot(t,rudderCommand)
ylabel('Rudder Command [rad]')
xlabel('Time [sec]')
```



Published with MATLAB® R2014a

6.1 from Burl

Given:

$$J(x) = x^T Q x + v^T x$$

Q is postive definite and v is fixed

$$\Delta J(x+\delta x) = (x+\delta x)^T Q (x+\delta x) + v^T(x+\delta x) - x^T Q x - v^T x$$

$$= x^T Q \delta x + \delta x^T Q x + \delta x^T Q \delta x + v^T \delta x$$

$$\delta J = (x^T Q + x^T Q + v^T)\delta x = 0$$

$$2x^T Q + v^T = 0$$

$$x = \left(\frac{-v^T Q^{-1}}{2}\right)^T$$

$$\boxed{x = \frac{-Q^{-1} v}{2}}$$

6.2 from Burl

Given:

$$J(x,y) = 5x^2 - 10x + 5 + y^2$$
$$y = x^2$$

Find: use Lagrange multiplier method to find x and y that minimize $J(x,y)$.

Solution:

Augment cost function so that the constraint becomes part of the cost function and the problem turns into unconstrained minimization.

$$J_a = 5x^2 - 10x + 5 + y^2 + \lambda(y - x^2)$$

Set partial derivatives of J_a to zero and solve to find critical point.

$$\frac{\partial J_a}{\partial x} = 10x - 10 - 2\lambda x = 0$$

$$\frac{\partial J_a}{\partial y} = 2y + \lambda = 0$$

$$\frac{\partial J_a}{\partial \lambda} = y - x^2 = 0$$

Solve for x, y, λ

$x = 0.79728$, $y = -0.63566$, $\lambda = -1.2713$

Burl

6.5

$\dot{x} = 2x + u$

$J(x,u) = 5x^2(10) + \int_0^{10} 2x^2 + u^2$

(a) Generate the Hamiltonian

$A = 2 \qquad Q = 4 \quad R = 2$

$B = 1 \qquad H = 10$

$$\begin{bmatrix} \dot{X} \\ \dot{P} \end{bmatrix} = \begin{bmatrix} 2 & -\frac{1}{2} \\ -4 & -2 \end{bmatrix} \begin{bmatrix} X \\ P \end{bmatrix}$$

(b) Generate ricatti equation

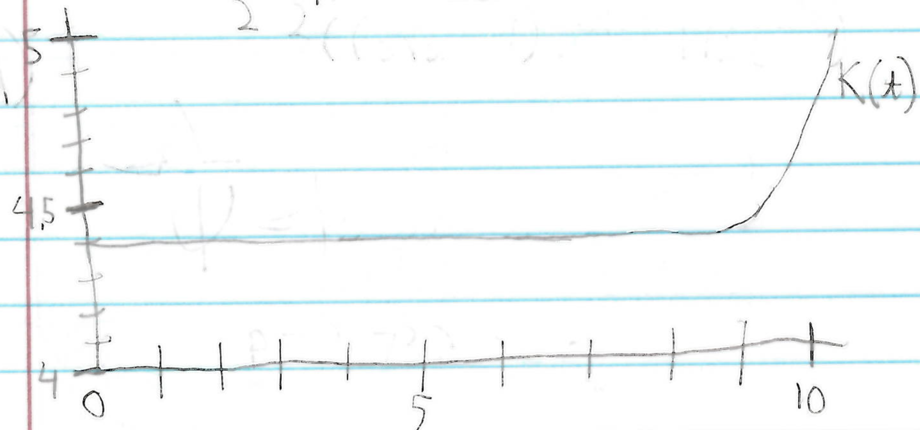
$\dot{P} = -P \cdot 2 - 2 \cdot P - 4 + P \cdot 1 \cdot \frac{1}{2} \cdot P$

$\dot{P} = \frac{1}{2} P^2 - 4P - 4$

(c) Solve for K(t) equation

$$P(t) = \frac{-2\left((\sqrt{6} - 2)\, e^{2\sqrt{6}t} + (9\sqrt{6} + 22)\, e^{20\sqrt{6}}\right)}{e^{2\sqrt{6}t} - (2\sqrt{6} + 5)\, e^{20\sqrt{6}}}$$

$K(t) = \frac{1}{2} \cdot 1 \cdot P(t)$



(d) $\Lambda = 2.4494$

$$W = \begin{bmatrix} 0.1117 & -0.7436 \\ 0.9937 & 0.6685 \end{bmatrix}$$

$$\bar{P} = 0.9937 \cdot \frac{1}{0.11167} = 8.899$$

$$\bar{K} = \frac{1}{2} \cdot 1 \cdot 8.899 = 4.449$$

(e) -2.4494

6.8 from Burl

Given: $J(x) = x^T Q x + V^T x$

Q is positive definite

V is given

Find: Show J has a unique minimum.

Solution:

Using the variation of J found in 6.1

$$\delta J(x) = (2x^T Q + V^T)\,\delta x = 0 \quad \forall\, \delta x$$

$$\Downarrow$$

$$2x^T Q + V^T = 0$$

solve for x

$$2x^T Q = -V^T$$

$$x^T = \frac{-V^T Q^{-1}}{2}$$

Using property of transpose $(AB)^T = B^T A^T$ and $(A^T)^T = A$ to get

$$x = \frac{-Q^{-1} V}{2}$$

The solution is unique if the Q^{-1} exist. Since Q is positive definite, Q^{-1} always exists for positive definite matrices.

(6.9) Given: $\dot{x}(t) = Ax(t) + Bu(t)$

$$J(x(t), u(t)) = \frac{1}{2} x^T(t_f) H x(t_f) + \frac{1}{2}\int_0^{t_f} \left([x(t) - r(t)]^T Q [x(t) - r(t)] + u^T(t) R u(t) \right) dt$$

Find: use variations to Lagrange multipliers to derive optimal control for this system.

Solution:

Change of variables $z(t) = x(t) - r(t)$

$\dot{z}(t) = \dot{x}(t) - \dot{r}(t)$

Sub into system equation

$$\dot{z}(t) + \dot{r}(t) = Az(t) + Ar(t) + Bu(t)$$

Form augmented cost function with new system equation.

$$J_a(z(t), u(t), p(t)) = \frac{1}{2} z^T(t_f) H z(t_f) + \frac{1}{2} r^T(t_f) H r(t_f)$$
$$+ \frac{1}{2}\int_0^{t_f} \left[z^T(t) Q z(t) + u^T(t) R u(t) \right] dt$$
$$+ \int_0^{t_f} p^T(t) \left(Az(t) + Ar(t) + Bu(t) - \dot{z}(t) - \dot{r}(t) \right) dt$$

Rearrange to show the augmented cost function can be split into two parts:

$$J_a(z(t), u(t)) = J_z(z(t), u(t), p(t)) + J_r(p(t))$$

$$J_z = \frac{1}{2} z^T(t_f) H z(t_f) + \frac{1}{2}\int_0^{t_f} \left(z^T(t) Q z(t) + u(t) R u(t)\right) dt$$

$$+ \int_0^{t_f} p^T(t) \left(A z(t) + B u(t) - \dot{z}(t)\right) dt$$

$$J_r = \frac{1}{2} r^T(t_f) H r(t_f) + \int_0^{t_f} p^T(t) \left(A r(t) - \dot{r}(t)\right) dt$$

Note the only unknown in J_r is $p(t)$. J_z is just LQR in $z(t)$. Now we form the variation with respect to the indepent variables.

$$\Delta J_r(p(t) + \delta p(t)) = \frac{1}{2} r^T(t_f) H r(t_f) + \int_0^{t_f} (p(t) + \delta p(t))^T (A r(t) - \dot{r}(t)) dt$$

$$- \frac{1}{2} r^T(t_f) H r(t_f) - \int_0^{t_f} p(t) (A r(t) - \dot{r}(t)) dt$$

$$\Delta J_r(p(t) + \delta p(t)) = \int_0^{t_f} \delta p^T(t) \left(A r(t) - \dot{r}(t)\right) dt$$

$$\Downarrow$$

$$\delta J_r(p(t), \delta p(t)) = \int_0^{t_f} \delta p^T(t) \left(A r(t) - \dot{r}(t)\right) dt$$

The variation for J_2 is the same as LQR. So to get variation in J_a, add δJ_2 to δJ_p.

$$\delta J_a = (z^T(t_f) H - p^T(t_f))\delta z(t_f)$$
$$+ \int_0^{t_f} \Big[(z^T(t) Q + p^T(t) A + \dot{p}^T(t))\delta z(t) + (u^T(t) R + p^T(t) B)\delta u(t)$$
$$+ \delta p^T(t) (A z(t) + B u(t) - \dot{z} + A r(t) - \dot{r}(t)) \Big] dt = 0$$

For the previous equation to be true all the terms multiplying the variations must be equal to zero.

① $z^T(t_f) H - p^T(t_f) = 0$
② $z^T(t) Q + p^T(t) A + \dot{p}^T(t) = 0$
③ $u^T(t) R + p^T(t) B = 0$
④ $\dot{z}(t) + \dot{r}(t) = A z(t) + A r(t) + B u(t)$

From equation ①, $p(t_f) = H z(t_f)$
From equation ③, $u(t) = -R^{-1} B^T p(t)$
From equation ② and ④ we get the system of equations

$$\dot{p}^T(t) = -z^T(t) Q - p^T(t) A$$
$$\dot{z}(t) + \dot{r}(t) = A z(t) + A r(t) + B u(t)$$

A little rearranging is in order to setup the costate equation

$$\dot{p}(t) = -Q\,z(t) - A^T p(t)$$

$$(5)\quad \dot{p}(t) = -Q\,x(t) + Q\,r(t) - A^T p(t)$$

$$(6)\quad \dot{x}(t) = A x(t) + B u(t)$$

Now use the costate relationship

$$p(t) = P(t) X(t) + s(t)$$

$$(7)\quad \dot{p}(t) = \dot{P}(t) x(t) + P(t) \dot{x}(t) + \dot{s}(t)$$

Substitute (3), (5), & (6) into (7)

$$-Q\,x(t) + Q\,r(t) - A^T P(t) x(t) - A^T s(t) =$$
$$\dot{P}(t)\,x(t) + P(t)\,A\,x(t) - P(t) B R^{-1} B^T P(t)\,x(t)$$
$$- P(t) B R^{-1} B^T s(t) + \dot{s}(t)$$

Factor with respect to $r(t)$, $s(t)$ and $x(t)$.

$$(8)\quad 0 = \left[\dot{P}(t) + P(t) A - P(t) B R^{-1} B^T P(t) + Q + A^T P(t)\right] X(t)$$

$$- P(t) B R^{-1} B^T s(t) + \dot{s}(t) + A^T s(t) - Q\,r(t)$$

For equation 8 to equal zero the factor multiplying $x(t)$ must equal zero, therefore $P(t)$ is the solution to the Riccati Equation. The 2nd part of equation must be equal to zero for all $r(t)$ therefore solve for $s(t)$

$$\dot{s}(t) = -(P(t)BR^{-1}B^T + A^T)s(t) + Qr(t)$$

The final conditions can be found by equation 1

$$p(t_f) = Hz(t_f) = Hx(t_f) - Hr(t_f) = P(t_f)x(t_f) + s(t_f) \quad (9)$$

Equation 9 implies the following

$$0 = (P(t_f) - H)x(t_f) + s(t_f) + Hr(t_f)$$

$$P(t_f) = H$$

6.1 Computer Problem from Burl

$$(U(s) - s\Theta(s))\, 0.02 \cdot \frac{1}{s} \cdot \frac{1}{s} = \Theta(s)$$

$$\frac{1}{0.02} s^2\Theta(s) + s\Theta(s) = U(s)$$

$$\ddot{\Theta} + 0.02\dot{\Theta} = 0.02\, U(s)$$

$$\begin{bmatrix} \dot{\Theta} \\ \ddot{\Theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -0.02 \end{bmatrix} \begin{bmatrix} \Theta \\ \dot{\Theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0.02 \end{bmatrix} u$$

$\Theta(t_f) = 0$

$\Theta(0) = 10\text{deg} = \frac{10}{180}\pi \text{ rad}$

$\dot{\Theta}(0) = 0 \frac{\text{rad}}{s}$

See following pages for MATLAB Output

Computer Problem 6.1 from Burl

Clean up

```
clc; close all; clear;
```

Cost Parameters

```
Q = [1 0;0 0];
R = 10^-8;
```

State space matrices

```
A = [0 1; 0 -0.2];
B = [0;0.02];


% Initial Conditions of states
xInitial = [10; 0];

% anonymous function to simulate time varying gains
x_dot = @(t,x,A,B,u) A*x+B*u;
```

Compare Time-Varying to Steady State for Time Ending 0.4 sec

```
timeFinal = 0.4; % sec
timeInitial = 0; % sec
Pfinal = zeros(2,2); % Final conditions of riccati differential equation
n = size(Pfinal,1); % The size of P
options = odeset('MaxStep', 1e-3);
[tP,P] = ode113(@(t,P) riccattiDifferentialEquation(t,P,n,A,B,Q,R), [timeFinal timeInitial],
Pfinal(:),options);

% reshape P to be a matrax that varies in time and thus is a 3d array
P = reshape(P', [n n size(P,1)]);
```

```
% time dependent gain matrix
K = zeros([size(B,2) size(B,1) size(P,3)]);
for i = 1:size(P,3)
    K(:,:,i) = R^(-1)*B'*P(:,:,i);
end

% integrate state equation
[t,x] = ode113(@(t,x) x_dot(t,x,A,B,-matrixInterp1(tP,K,t)*x),[timeInitial timeFinal],
xInitial,options);


% control voltage time varying
u = zeros(size(t));
for i=1:size(t)
    u(i) = - matrixInterp1(tP,K,t(i))*x(i,:)';
end

% Solve Steady State LQR
K2 = lqr(A,B,Q,R);
[~,t2,x2] = initial(ss(A-B*K2,[],[],[]),xInitial,timeFinal);
u2 = -K2*x2';
```

Cost Calculations

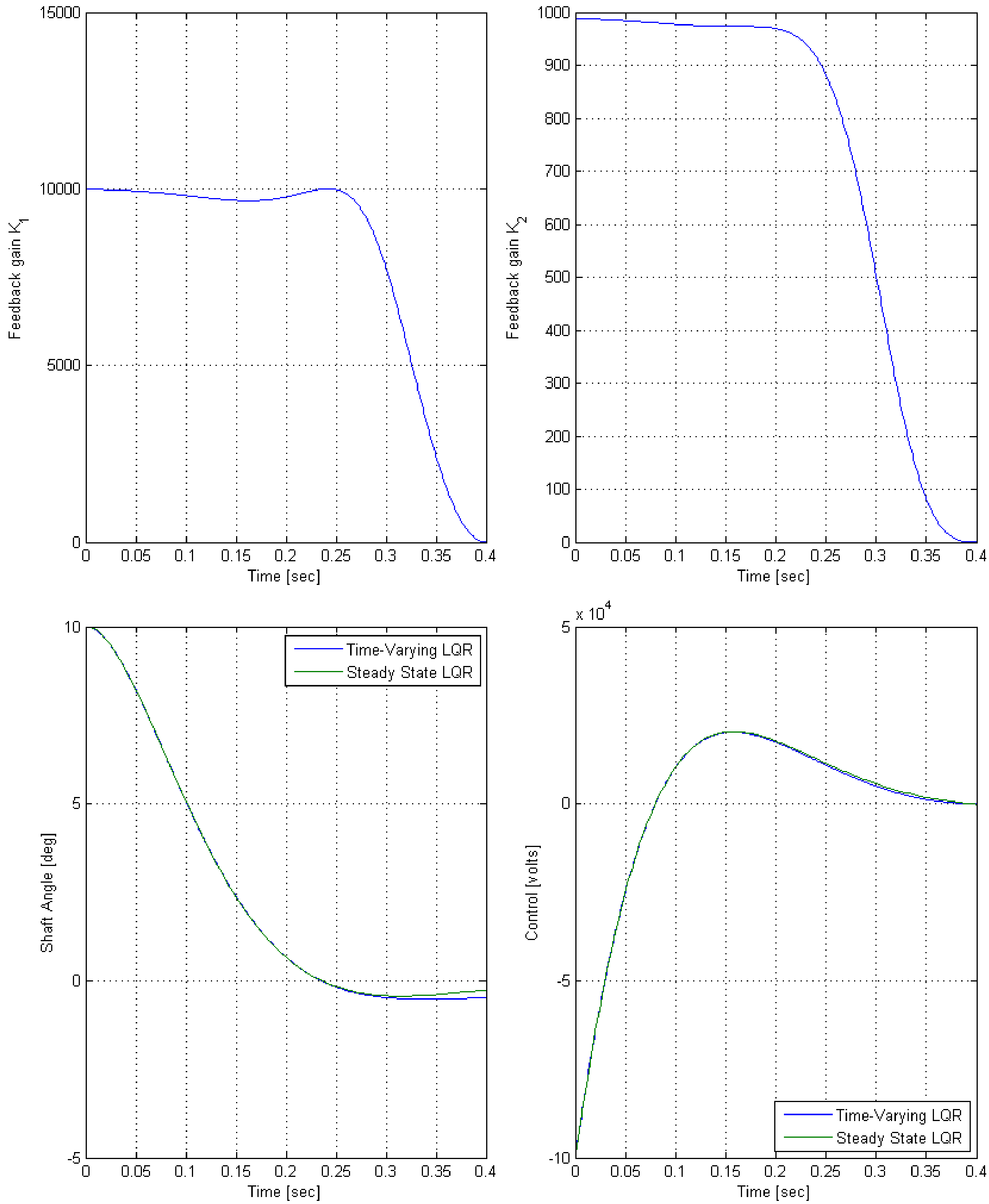
```
J = 1/2*xInitial'*P(:,:,end)*xInitial;
disp(['Time-Varying Cost for time ending ' num2str(timeFinal) ' is J = ' num2str(J)]);

% to calculate the cost for the steady state LQR but with a final time, the
% definition of the cost function must be used
dummy = zeros(size(t2));
for i=1:length(t2)
    dummy(i) = x2(i,:)*Q*x2(i,:)'+ R*u2(i)^2;
end
J2 = 1/2*integral(@(tt) interp1(t2,dummy,tt),timeInitial,timeFinal);
disp(['Steady State Cost for time ending ' num2str(timeFinal) ' is J = ' num2str(J2)]);
```

```
Time-Varying Cost for time ending 0.4 is J = 4.9974
Steady State Cost for time ending 0.4 is J = 5.0012
```

Create Plots

```
figure;
screenSize = get(0,'screensize');
set(gcf,'position', [ screenSize(3)/10 screenSize(4)/10 831 1012]);
subplot = @(m,n,p) subtightplot (m, n, p, [0.07 0.09], [0.04 0.01], [0.09 0.01]);
subplot(2,2,1)
plot(tP,squeeze(K(1,1,:)))
xlabel('Time [sec]')
ylabel('Feedback gain K_1');
```

```
ylim([0 15000]);
grid('on')
subplot(2,2,2)
plot(tP,squeeze(K(1,2,:)))
xlabel('Time [sec]')
ylabel('Feedback gain K_2');
grid('on')
subplot(2,2,3)
plot(t,x(:,1),t2,x2(:,1))
ylim([-5 10])
xlabel('Time [sec]')
ylabel('Shaft Angle [deg]')
grid('on')
legend('Time-Varying LQR', 'Steady State LQR')
subplot(2,2,4)
plot(t,u,t2,u2)
ylim([-10e4 5e4]);
xlabel('Time [sec]')
ylabel('Control [volts]')
grid('on')
legendHandle = legend('Time-Varying LQR', 'Steady State LQR');
set(legendHandle,'Location','SouthEast')
```



Feedback gain K1
Time [sec]
Feedback gain K2
Time [sec]
Shaft Angle [deg]
Time [sec]
Time-Varying LQR
Steady State LQR
x 10^4
Control [volts]
Time [sec]
Time-Varying LQR
Steady State LQR

Part (a) Compare Time-Varying to Steady State for Time Ending 0.2 sec

```
timeFinal = 0.2; % sec
timeInitial = 0; % sec
Pfinal = zeros(2,2); % Final conditions of riccati differential equation
n = size(Pfinal,1); % The size of P
options = odeset('MaxStep', 1e-3);
[tP,P] = ode113(@(t,P) riccatiDifferentialEquation(t,P,n,A,B,Q,R), [timeFinal timeInitial],
Pfinal(:),options);

% reshape P to be a matrax that varies in time and thus is a 3d array
P = reshape(P', [n n size(P,1)]);

% time dependent gain matrix
K = zeros([size(B,2) size(B,1) size(P,3)]);
for i = 1:size(P,3)
    K(:,:,i) = R^(-1)*B'*P(:,:,i);
end

% integrate state equation
[t,x] = ode113(@(t,x) x_dot(t,x,A,B,-matrixInterp1(tP,K,t)*x),[timeInitial timeFinal],
xInitial,options);


% control voltage time varying
u = zeros(size(t));
for i=1:size(t)
    u(i) = - matrixInterp1(tP,K,t(i))*x(i,:)';
end

% Solve Steady State LQR
K2 = lqr(A,B,Q,R);
[~,t2,x2] = initial(ss(A-B*K2,[],[],[]),xInitial,timeFinal);
u2 = -K2*x2';
```

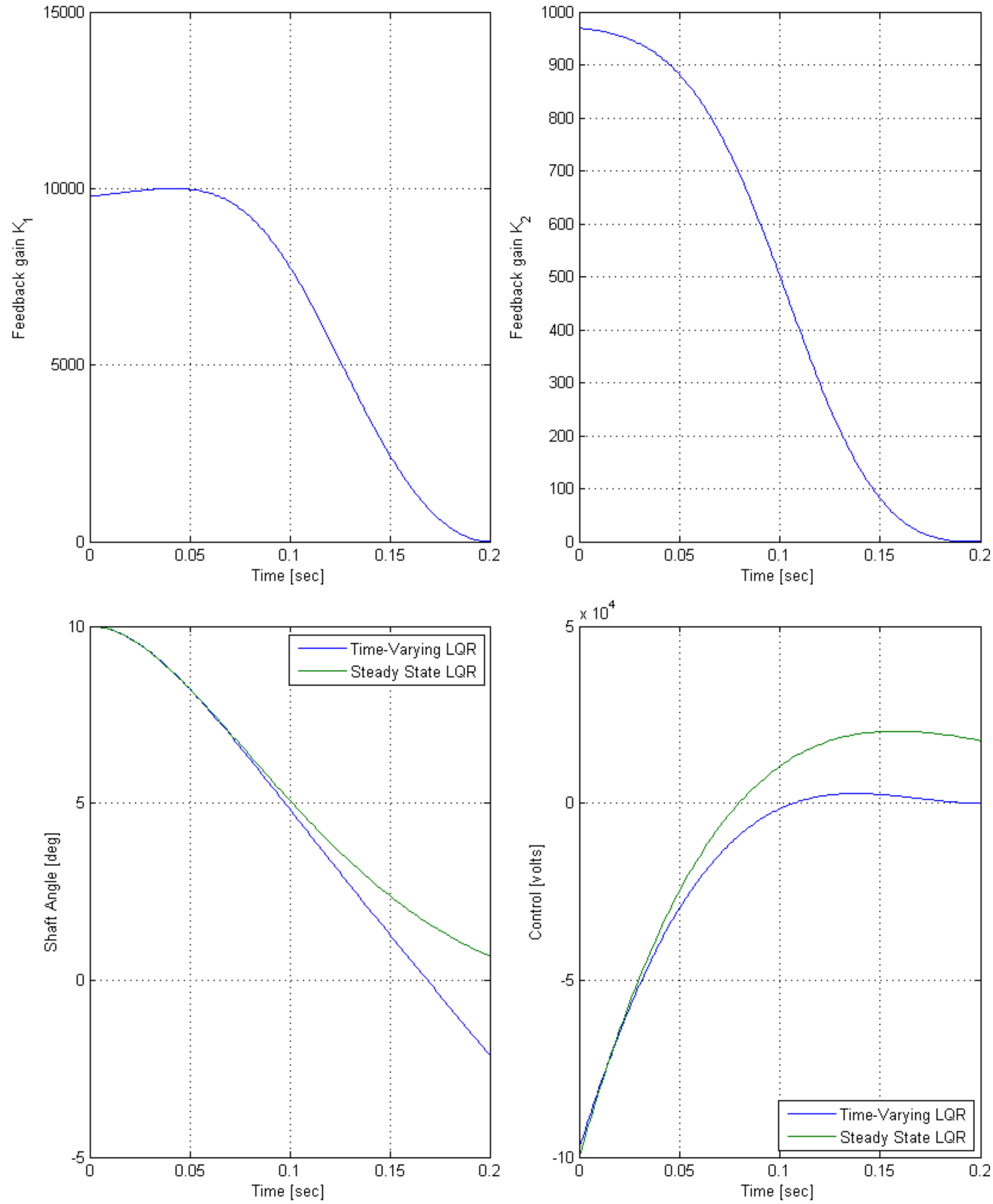
Cost Calculations

```
J = 1/2*xInitial'*P(:,:,end)*xInitial;
disp(['Time-Varying Cost for time ending ' num2str(timeFinal) ' is J = ' num2str(J)]);
% to calculate the cost for the steady state LQR but with a final time, the
% definition of the cost function must be used
dummy = zeros(size(t2));
for i=1:length(t2)
    dummy(i) = x2(i,:)*Q*x2(i,:)'+ R*u2(i)^2;
end
J2 = 1/2*integral(@(tt) interp1(t2,dummy,tt),timeInitial,timeFinal);
disp(['Steady State Cost for time ending ' num2str(timeFinal) ' is J = ' num2str(J2)]);
```

```
Time-Varying Cost for time ending 0.2 is J = 4.6418
Steady State Cost for time ending 0.2 is J = 4.9108
```

Create Plots

```
figure;
screenSize = get(0,'screensize');
set(gcf,'position', [ screenSize(3)/10 screenSize(4)/10 831 1012]);
subplot = @(m,n,p) subtightplot (m, n, p, [0.07 0.09], [0.04 0.01], [0.09 0.01]);
subplot(2,2,1)
plot(tP,squeeze(K(1,1,:)))
xlabel('Time [sec]')
ylabel('Feedback gain K_1');
ylim([0 15000]);
grid('on')
subplot(2,2,2)
plot(tP,squeeze(K(1,2,:)))
xlabel('Time [sec]')
ylabel('Feedback gain K_2');
grid('on')
subplot(2,2,3)
plot(t,x(:,1),t2,x2(:,1))
ylim([-5 10])
xlabel('Time [sec]')
ylabel('Shaft Angle [deg]')
grid('on')
legend('Time-Varying LQR', 'Steady State LQR')
subplot(2,2,4)
plot(t,u,t2,u2)
ylim([-10e4 5e4]);
xlabel('Time [sec]')
ylabel('Control [volts]')
grid('on')
legendHandle = legend('Time-Varying LQR', 'Steady State LQR');
set(legendHandle,'Location','SouthEast')
```


Feedback gain K_1
Time [sec]
Feedback gain K_2
Time [sec]
Shaft Angle [deg]
Time [sec]
Time-Varying LQR
Steady State LQR
x 10^4
Control [volts]
Time [sec]
Time-Varying LQR
Steady State LQR

Part (b) Compare Time-Varying to Steady State for Time Ending 0.1 sec

```
timeFinal = 0.1; % sec
timeInitial = 0; % sec
Pfinal = zeros(2,2); % Final conditions of riccati differential equation
n = size(Pfinal,1); % The size of P
options = odeset('MaxStep', 1e-3);
[tP,P] = ode113(@(t,P) riccatiDifferentialEquation(t,P,n,A,B,Q,R), [timeFinal timeInitial],
Pfinal(:),options);

% reshape P to be a matrax that varies in time and thus is a 3d array
P = reshape(P', [n n size(P,1)]);

% time dependent gain matrix
K = zeros([size(B,2) size(B,1) size(P,3)]);
for i = 1:size(P,3)
    K(:,:,i) = R^(-1)*B'*P(:,:,i);
end

% integrate state equation
[t,x] = ode113(@(t,x) x_dot(t,x,A,B,-matrixInterp1(tP,K,t)*x),[timeInitial timeFinal],
xInitial,options);


% control voltage time varying
u = zeros(size(t));
for i=1:size(t)
    u(i) = - matrixInterp1(tP,K,t(i))*x(i,:)';
end

% Solve Steady State LQR
K2 = lqr(A,B,Q,R);
[~,t2,x2] = initial(ss(A-B*K2,[],[],[]),xInitial,timeFinal);
u2 = -K2*x2';
```

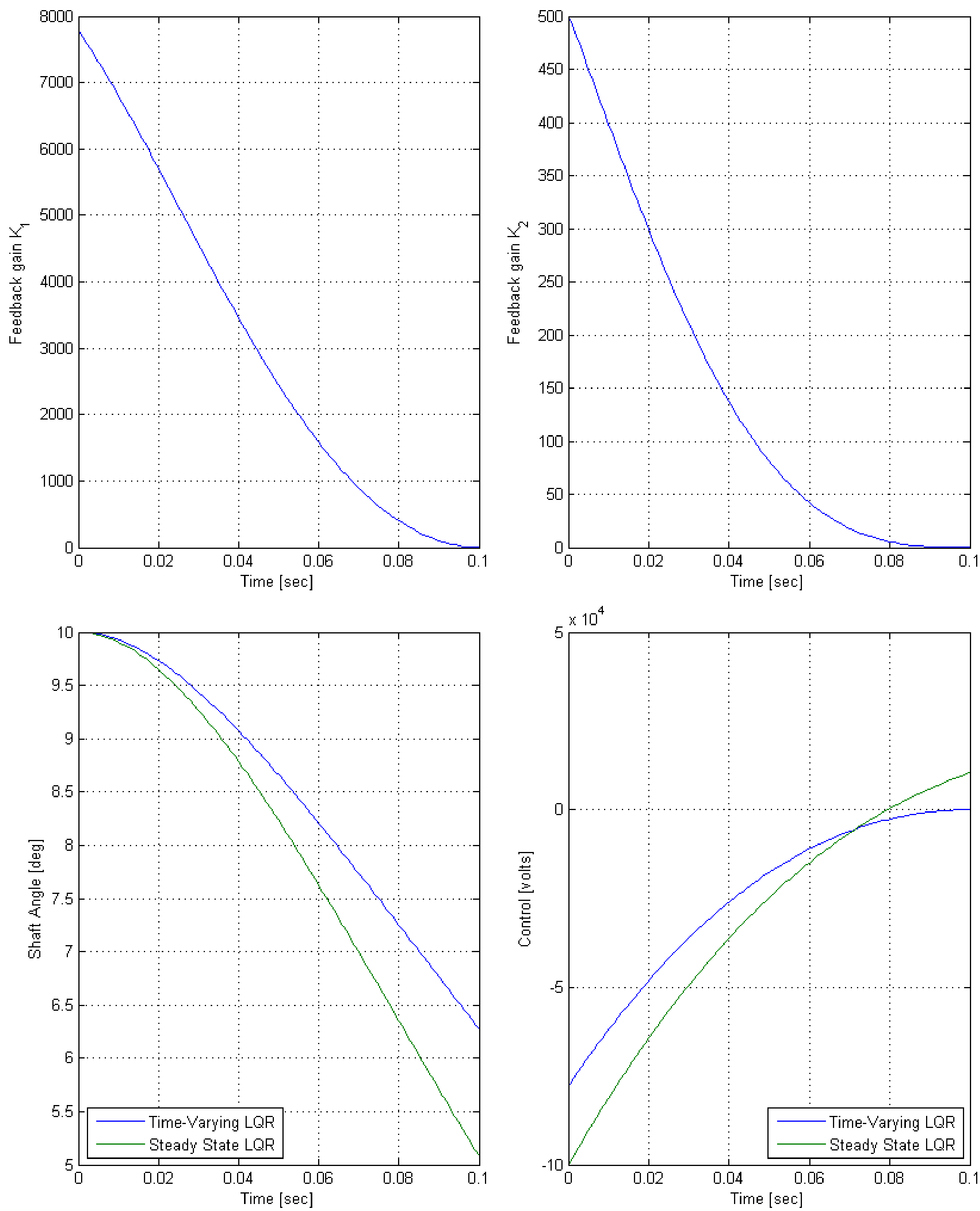
Cost Calculations

```
J = 1/2*xInitial'*P(:,:,end)*xInitial;
disp(['Time-Varying Cost for time ending ' num2str(timeFinal) ' is J = ' num2str(J)]);
% to calculate the cost for the steady state LQR but with a final time, the
% definition of the cost function must be used
dummy = zeros(size(t2));
for i=1:length(t2)
    dummy(i) = x2(i,:)*Q*x2(i,:)'+ R*u2(i)^2;
end
J2 = 1/2*integral(@(tt) interp1(t2,dummy,tt),timeInitial,timeFinal);
disp(['Steady State Cost for time ending ' num2str(timeFinal) ' is J = ' num2str(J2)]);
```

```
Time-Varying Cost for time ending 0.1 is J = 4.2487
Steady State Cost for time ending 0.1 is J = 4.3332
```

Create Plots

```
figure;
screenSize = get(0,'screensize');
set(gcf,'position', [ screenSize(3)/10 screenSize(4)/10 831 1012]);
subplot = @(m,n,p) subtightplot (m, n, p, [0.07 0.09], [0.04 0.01], [0.09 0.01]);
subplot(2,2,1)
plot(tP,squeeze(K(1,1,:)))
ylim([0 8000])
xlabel('Time [sec]')
ylabel('Feedback gain K_1');
grid('on')
subplot(2,2,2)
plot(tP,squeeze(K(1,2,:)))
ylim([0 500])
xlabel('Time [sec]')
ylabel('Feedback gain K_2');
grid('on')
subplot(2,2,3)
plot(t,x(:,1),t2,x2(:,1))
ylim([5 10])
xlabel('Time [sec]')
ylabel('Shaft Angle [deg]')
grid('on')
legendHandle = legend('Time-Varying LQR', 'Steady State LQR');
set(legendHandle,'Location','SouthWest');
subplot(2,2,4)
plot(t,u,t2,u2)
ylim([-10e4 5e4])
xlabel('Time [sec]')
ylabel('Control [volts]')
grid('on')
legendHandle = legend('Time-Varying LQR', 'Steady State LQR');
set(legendHandle,'Location','SouthEast')
```



Published with MATLAB® R2013b

Riccati Differential Equation Function

```
function P_dot = riccattiDifferentialEquation(t,P, n, A, B, Q, R) %#ok<INUSL>
% Used to with ode solver the time varying riccati equation
% t = time
% P = Current value of P
% n = size of P
% A = from state equation
% B = from state equation
% Q = from cost equation
% R = from cost equation

% P must be sent in as a vector so I have to turn back into a matrix before
% using it
P = reshape(P,[n n]);

% Calculate the current value of P_dot
P_dot = -P*A-A'*P-Q+P*B*R^(-1)*B'*P;

% Turn the P_dot back into a vector
P_dot = P_dot(:);
```

Published with MATLAB® R2013b

Matrix Interpolation Function

```
function yq = matrixInterp1(x,y,xq)
% interpolates a matrix yq given the 3d array y and vector x at points xq
%

% Setup output array
yq = zeros([size(y,1) size(y,2) size(xq)]);

% iterate over each column and row of y thinking of the 3rd dimension as a
% vector in time
for rowY = 1:size(y,1)
    for columnY = 1:size(y,2)
        currentYvector = squeeze(y(rowY,columnY,:));
        yq(rowY,columnY,:) = interp1(x,currentYvector,xq);
    end
end
```

Published with MATLAB® R2013b