AE353 Homework #5: Observability and Output Feedback

SOLUTIONS

(due at the beginning of class on Friday, March 13)

1. You have seen that the motion of a robot arm with one revolute joint can be described in state-space form as

$$\begin{aligned}\dot{x} &= \begin{bmatrix}0 & 1\\ 0 & -1/5\end{bmatrix} x + \begin{bmatrix}0\\ 1/5\end{bmatrix} u\\ y &= \begin{bmatrix}1 & 0\end{bmatrix} x\end{aligned}$$

where the state elements are the angle (x_1) and angular velocity (x_2) of the joint, the input u is a torque applied to the arm at the joint, and the output y is a measurement of the angle.

(a) (2 points) Show that this system is observable. Please do so by hand.

The system is observable iff its associated observability matrix has full *ROW* rank.

$$\begin{aligned}\mathcal{W}_o &= \begin{bmatrix}\mathbf{C}\\ \mathbf{CA}\end{bmatrix}\\ &= \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\end{aligned}$$

It is clear that $\mathcal{W}_o$ has full row rank $\implies$ the system is observable.

(b) (3 points) Design an observer

$$\dot{\widehat{x}} = A\widehat{x} + Bu - L(C\widehat{x} - y)$$

with both closed-loop eigenvalues at -10. Recall that by "closed-loop," we mean the system

$$\frac{d}{dt}(\widehat{x} - x) = (A - LC)(\widehat{x} - x)$$

where

$$L = \begin{bmatrix}\ell_1\\ \ell_2\end{bmatrix}$$

is a gain matrix and where

$$\widehat{x} - x$$

is the error in our estimate of the state. Please design this observer "by hand" (i.e., compute the characteristic equation of the closed-loop system in terms of ℓ_1 and ℓ_2, then choose gains that make this characteristic equation look like what you want).

First, we need to determine the closed-loop state matrix of the observer $\mathbf{A}_{cl-o}$:

$$\mathbf{A}_{cl-o} = \mathbf{A} - \mathbf{LC} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{1}{5} \end{bmatrix} - \begin{bmatrix} \ell_1 \\ \ell_2 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix}$$
$$= \begin{bmatrix} -\ell_1 & 1 \\ -\ell_2 & -\frac{1}{5} \end{bmatrix}$$

Now, we determine the closed-loop characteristic equation for the observer:

$$det\left(\mathbf{A}_{cl-o} - s\mathbb{I}\right) = 0$$
$$\begin{vmatrix} -\ell_1 - s & 1 \\ -\ell_2 & -\frac{1}{5} - s \end{vmatrix} = s^2 + \left(\ell_1 + \frac{1}{5}\right) s + \frac{\ell_1}{5} + \ell_2 = 0$$

If we want to place the two observer eigenvalues at $s = -\sigma_1 = -\sigma_2 = -10$, we know that the characteristic equation must be:

$$(s + \sigma_1)(s + \sigma_2) = s^2 + (\sigma_1 + \sigma_2)s + \sigma_1\sigma_2$$

We can now equate the coefficients of these two characteristic equations and solve for the observer gains:

$$\therefore \ell_1 + \frac{1}{5} = \sigma_1 + \sigma_2 \implies \ell_1 = \frac{99}{5} = 19.8$$
$$\therefore \frac{\ell_1}{5} + \ell_2 = \sigma_1\sigma_2 \implies \ell_2 = \frac{401}{5} = 96.04$$

(c) (1 point) Design a controller

$$u = -K\widehat{x} + k_{\text{reference}} r$$

with closed-loop eigenvalues at $-2 \pm j2$ and with zero steady-state error in reference tracking. Recall that by "closed-loop," we mean the system

$$\dot{x} = (A - BK)x + Bk_{\text{reference}} r,$$

which describes how the state would behave in closed-loop if our estimate of it were perfect, i.e., if $\widehat{x} = x$. Design this controller any way you like.

Using the MATLAB function `acker`, we can compute the controller gain matrix $\mathbf{K}$ that will place the closed-loop eigenvalues of the controller at $s = -2 \pm j2$. We also desire zero steady-state error in reference tracking, so we need to compute an appropriate value for the reference gain $\mathbf{k}_{reference}$:

```
A = [0 1;0 -1/5];
B = [0;1/5];
C = [1 0];
D = 0;

%controller design
K = acker(A,B,[-2+2*1i -2-2*1i]);
kref = -inv(C*(inv(A-B*K))*B);
```

This results in the following gain values:

$$\mathbf{K} = \begin{bmatrix} k_1 & k_2 \end{bmatrix} = \begin{bmatrix} 40.0 & 19.0 \end{bmatrix} \quad k_{reference} = 40$$

(d) (4 points) Express the entire closed-loop system—with both observer and controller—in the form

$$\frac{d}{dt}\begin{bmatrix} x \\ \hat{x} \end{bmatrix} = A_{\mathrm{cl}} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} + B_{\mathrm{cl}} r$$

for some matrices A_{cl} and B_{cl}. What are the eigenvalues of this system? Do you notice anything interesting?

We begin with the state space representation for the true states x and the estimated states $\hat{x}$:

$$\begin{aligned} \dot{x} &= \mathbf{A}x + \mathbf{B}u \\ \dot{\hat{x}} &= \mathbf{A}\hat{x} + \mathbf{B}u - \mathbf{L}\left(\mathbf{C}\hat{x} - y\right) \end{aligned}$$

We are implementing a controller of the form:

$$u = -\mathbf{K}\hat{x} + k_{reference} r$$

$$\begin{aligned} \therefore \dot{x} &= \mathbf{A}x + \mathbf{B}\left(-\mathbf{K}\hat{x} + k_{reference} r\right) \\ \dot{\hat{x}} &= \mathbf{A}\hat{x} + \mathbf{B}\left(-\mathbf{K}\hat{x} + k_{reference} r\right) - \mathbf{L}\left(\mathbf{C}\hat{x} - y\right) \end{aligned}$$

$$\begin{aligned} \therefore \dot{x} &= \mathbf{A}x - \mathbf{BK}\hat{x} + \mathbf{B}k_{reference} r \\ \dot{\hat{x}} &= \left(\mathbf{A} - \mathbf{LC} - \mathbf{BK}\right)\hat{x} + \mathbf{B}k_{reference} r + \mathbf{L}y \end{aligned}$$

We know that the output is equal to the product of $\mathbf{C}$ and the state vector x, so we have:

$$\begin{aligned} \therefore \dot{x} &= \mathbf{A}x - \mathbf{BK}\hat{x} + \mathbf{B}k_{reference} r \\ \dot{\hat{x}} &= \mathbf{LC}x + \left(\mathbf{A} - \mathbf{LC} - \mathbf{BK}\right)\hat{x} + \mathbf{B}k_{reference} r \end{aligned}$$

Or, as a matrix representation,

$$\begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & -\mathbf{BK} \\ \mathbf{LC} & \mathbf{A} - \mathbf{LC} - \mathbf{BK} \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{B} \end{bmatrix} k_{reference} r$$

The closed-loop state-space representation with an estimator for our system is:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{\hat{x}}_1 \\ \dot{\hat{x}}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{1}{5} & -8 & -\frac{19}{5} \\ \frac{99}{5} & 0 & -\frac{99}{5} & 1 \\ \frac{2401}{25} & 0 & -\frac{2601}{25} & -4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 8 \\ 0 \\ 8 \end{bmatrix} r$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{\hat{x}}_1 \\ \dot{\hat{x}}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -0.2 & -8 & -3.8 \\ 19.8 & 0 & -19.8 & 1 \\ 96.04 & 0 & -104.04 & -4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 8 \\ 0 \\ 8 \end{bmatrix} r$$

The eigenvalues of this system can be computed by determining the roots of the characteristic equation of the closed-loop state matrix:

$$det(\mathbf{A}_{cl} - s\mathbb{I}) = s^4 + 24s^3 + 188s^2 + 560s + 800 = 0$$

where,

$$\mathbf{A}_{cl} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{1}{5} & -8 & -\frac{19}{5} \\ \frac{99}{5} & 0 & -\frac{99}{5} & 1 \\ \frac{2401}{25} & 0 & -\frac{2601}{25} & -4 \end{bmatrix}$$

This is easily accomplished using the MATLAB function `roots`

```
roots([1 24 188 560 800])

ans =

-10.000000146463613 + 0.000000000000000i
 -9.999999853536389 + 0.000000000000000i
 -2.000000000000002 + 2.000000000000001i
 -2.000000000000002 - 2.000000000000001i
```

Or, we can ask MATLAB to compute the eigenvalues of $\mathbf{A}_{cl}$ directly using `eig`

```
[V,Lambda] = eig(Acl)
```

Note, that the eigenvalues of the combined closed-loop system are the union of the closed-loop controlled system and the closed-loop estimated system. This tells us that we can design our estimator and our controller separately, and then combine them to form the complete dynamical system.

(e) (3 points) Compute the response of the entire closed-loop system to a reference signal

$$r(t) = \pi/2 \qquad \text{for all } t \geq 0$$

for the initial condition

$$x(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

and for the initial state estimate

$$\widehat{x}(0) = \begin{bmatrix} -1 \\ 2 \end{bmatrix}$$

over a suitable time horizon. Do so however you like. Plot the following:

- x_1 and $\widehat{x}_1$ as functions of time on the same figure;
- x_2 and $\widehat{x}_2$ as functions of time on the same figure;
- u as a function of time.

There are many ways to compute the response. One way is to integrate the equations of motion using `ode45`

```
A = [0 1;0 -1/5];
B = [0;1/5];
C = [1 0];
D = 0;

% observer design
sig1 = 10;
sig2 = 10;

L1 = sig1 + sig2 - 1/5;
L2 = sig1*sig2 - L1/5;

L = [L1;L2];

% controller design
K = acker(A,B,[-2+2*1i -2-2*1i]);
kref = -inv(C*(inv(A-B*K))*B);

% full closed-loop system
Acl = [A -B*K;L*C A-L*C-B*K];

[V,Lambda] = eig(Acl);

% response
r = pi/2;
X0 = [0;0;-1;2];


tspan = linspace(0,10,1000);
```

```
[T,X] = ode45(@(t,X)stateSpaceEOM(t,X,K,kref,r,A,B,C,D,L),tspan,X0);

% recover the control history
uhistory = zeros(length(T),1);
for i = 1:length(T)
    uhistory(i) = -K*X(i,3:4)' + kref*r;
end

function [Xdot u] = stateSpaceEOM(t,X,K,kref,r,A,B,C,D,L)

    % let's separate the true states and the estimated states
    x    = X(1:length(X)/2);
    xhat = X(length(X)/2+1:end);

    y = C*x;

    % controller
    u = -K*xhat + kref*r;

    % state-space representation of the true and estimated dynamics
    xdot = A*x + B*u;
    xhatdot = A*xhat + B*u - L*(C*xhat - y);

    % package the state time derivative vector
    Xdot = [xdot;xhatdot];
```

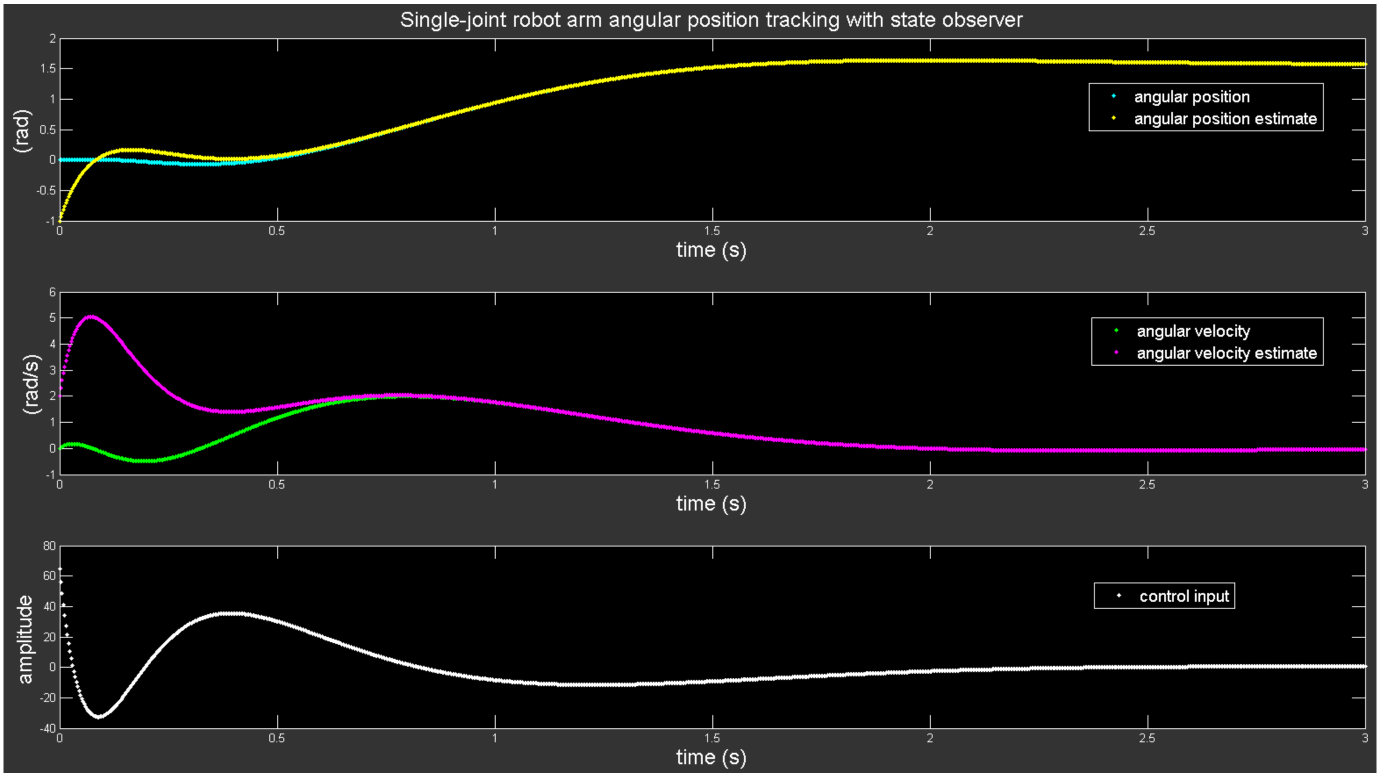


Figure 1: Robotic arm angular position tracking using a state estimator.

(f) (2 points) Implement your controller and observer by modifying the function

```
[u,userdata] = ControlLoop(y,r,userdata,params)
```

in the MATLAB script `hw5prob01.m`. As you know, this function should do two things:

- Choose the control input:

$$u(t) = -K\widehat{x}(t) + k_{\text{reference}}r(t)$$

- Update the state estimate:

$$\widehat{x}(t+\Delta t) = \widehat{x}(t) + \Delta t\left(A\widehat{x}(t) + Bu(t) - L(C\widehat{x}(t) - y(t))\right)$$

You will recognize the update equation as a first-order approximation to

$$\dot{\widehat{x}}(t) = A\widehat{x}(t) + Bu(t) - L(C\widehat{x}(t) - y(t)).$$

Submit only your `ControlLoop` function (this should be the only part of the code that you change) and a snapshot of the figure after the simulation has ended.

```
function hw5prob01
clear all
clc;

% %%%%%%%%%%%%%%%%%%%%
% PARAMETERS
%
% - state-space system
params.A = [0 1; 0 -1/5];
params.B = [0; 1/5];
params.C = [1 0];
params.K = acker(params.A,params.B,[-2+2*1i -2-2*1i]);
params.kref = -inv(params.C*(inv(params.A-params.B*params.K))*params.B);
params.xhat0 = [-1;2];

%observer design
sig1 = 10;
sig2 = 10;

L1 = sig1 + sig2 - 1/5;
L2 = sig1*sig2 - L1/5;
params.L = [L1;L2];

% - time step
params.dt = 1e-2;
%
% %%%%%%%%%%%%%%%%%%%%

RunSimulation(5,params);
```

```
function [u,userdata] = ControlLoop(y,r,userdata,params)

  persistent isFirstTime
if isempty(isFirstTime)
      isFirstTime = false;
      fprintf(1,'initialize control loop\n');
      userdata.xhat = params.xhat0;
end

u = -params.K*userdata.xhat + params.kref*r;
userdata.xhat = userdata.xhat + params.dt*(params.A*userdata.xhat + params.B*u ...
                - params.L*(params.C*userdata.xhat - y));
```

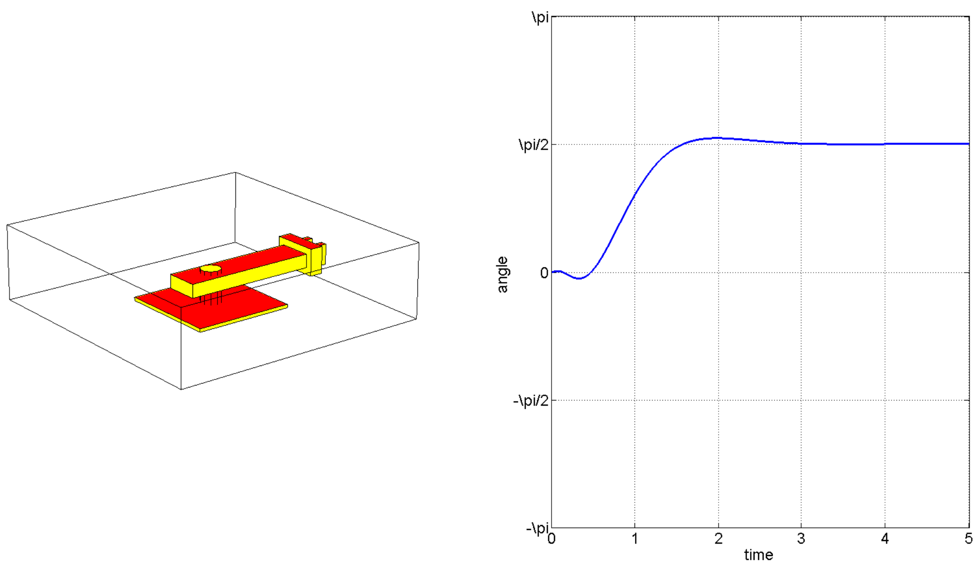


Figure 2: Robotic arm angular position tracking simulation using a state estimator.

2. You have seen that the rotational motion of an axisymmetric spacecraft about its yaw and roll axes can be described in state-space form as

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

where the state elements x_1 and x_2 are the angular velocities about yaw and roll axes, the input u is an applied torque, and the parameter $\lambda = 9$ is the relative spin rate.

(a) (2 points) Show that this system is observable. Please do so by hand.

The system is observable iff its associated observability matrix has full ROW rank.

$$\begin{aligned} \mathcal{W}_o &= \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \end{bmatrix} \\ &= \begin{bmatrix} 0 & 1 \\ -9 & 0 \end{bmatrix} \end{aligned}$$

$$RREF\left(\mathcal{W}_o{}^T\right) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$\therefore \mathcal{W}_o$ has full row rank $\implies$ the system is observable.

(b) (3 points) Design an observer

$$\dot{\widehat{x}} = A\widehat{x} + Bu - L(C\widehat{x} - y)$$

with both closed-loop eigenvalues at -15. Please design this observer "by hand."

First, we need to determine the closed-loop state matrix of the observer $\mathbf{A}_{cl-o}$:

$$\begin{aligned} \mathbf{A}_{cl-o} = \mathbf{A} - \mathbf{LC} &= \begin{bmatrix} 0 & \lambda \\ -\lambda & 0 \end{bmatrix} - \begin{bmatrix} \ell_1 \\ \ell_2 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 0 & \lambda - \ell_1 \\ -\lambda & -\ell_2 \end{bmatrix} \end{aligned}$$

Now, we determine the closed-loop characteristic equation for the observer:

$$det\left(\mathbf{A}_{cl-o} - s\mathbb{I}\right) = 0$$
$$\begin{vmatrix} -s & \lambda - \ell_1 \\ -\lambda & -\ell_2 - s \end{vmatrix} = s^2 + \ell_2 s + \lambda^2 - \lambda\ell_1 = 0$$

If we want to place the two observer eigenvalues at $s = -\sigma_1 = -\sigma_2 = -15$, we know that the characteristic equation must be:

$$(s+\sigma_1)(s+\sigma_2) = s^2 + (\sigma_1+\sigma_2)s + \sigma_1\sigma_2$$

We can now equate the coefficients of these two characteristic equations and solve for the observer gains:

$$\therefore \lambda^2 - \lambda\ell_1 = \sigma_1\sigma_2 \implies \ell_1 = -16$$
$$\therefore \ell_2 = \sigma_1 + \sigma_2 \implies \ell_2 = 30$$

(c) (1 point) Design a controller

$$u = -K\widehat{x}$$

with closed-loop eigenvalues at $-3 \pm j5$. Do so any way you like.

```
lmbda = 9;

A = [0 lmbda;-lmbda 0];
B = [1;0];
C = [0 1];
D = 0;

%controller design
K = acker(A,B,[-3+5*1i -3-5*1i]);
```

This results in the following gain values:

$$\mathbf{K} = \begin{bmatrix} k_1 & k_2 \end{bmatrix} = \begin{bmatrix} 6 & \frac{47}{9} \end{bmatrix} = \begin{bmatrix} 6.000000 & 5.222222 \end{bmatrix}$$

(d) (4 points) Express the entire closed-loop system—with both observer and controller—in the form

$$\frac{d}{dt}\begin{bmatrix} x \\ \widehat{x} \end{bmatrix} = A_{\text{cl}} \begin{bmatrix} x \\ \widehat{x} \end{bmatrix}$$

for some matrix A_{cl}. What are the eigenvalues of this system? Do you notice anything interesting?

$$\begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & -\mathbf{BK} \\ \mathbf{LC} & \mathbf{A}-\mathbf{LC}-\mathbf{BK} \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix}$$

The closed-loop state-space representation with an estimator for our system is:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{\hat{x}}_1 \\ \dot{\hat{x}}_2 \end{bmatrix} = \begin{bmatrix} 0 & 9 & -6 & -\frac{47}{9} \\ -9 & 0 & 0 & 0 \\ 0 & -16 & -6 & \frac{178}{9} \\ 0 & 30 & -9 & -30 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \hat{x}_1 \\ \hat{x}_2 \end{bmatrix}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{\hat{x}}_1 \\ \dot{\hat{x}}_2 \end{bmatrix} = \begin{bmatrix} 0 & 9.0 & -6.0 & -5.222222 \\ -9.0 & 0 & 0 & 0 \\ 0 & -16.0 & -6.0 & 19.777778 \\ 0 & 30.0 & -9.0 & -30.0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \hat{x}_1 \\ \hat{x}_2 \end{bmatrix}$$

The eigenvalues of this system can be computed by determining the roots of the characteristic equation of the closed-loop state matrix:

$$det(\mathbf{A}_{cl} - s\mathbb{I}) = s^4 + 36s^3 + 439s^2 + 2370s + 7650 = 0$$

where,

$$\mathbf{A}_{cl} = \begin{bmatrix} 0 & 9 & -6 & -\frac{47}{9} \\ -9 & 0 & 0 & 0 \\ 0 & -16 & -6 & \frac{178}{9} \\ 0 & 30 & -9 & -30 \end{bmatrix}$$

This is easily accomplished using the MATLAB function `roots`

```
roots([1 36 439 2370 7650])

ans =

-14.999999999999979 + 0.000000728122709i
-14.999999999999979 - 0.000000728122709i
 -3.000000000000003 + 4.999999999999995i
 -3.000000000000003 - 4.999999999999995i
```

Or, we can ask MATLAB to compute the eigenvalues of $\mathbf{A}_{cl}$ directly using `eig`

```
[V,Lambda] = eig(Acl)
```

Note, that the eigenvalues of the combined closed-loop system are the union of the closed-loop controlled system and the closed-loop estimated system. This tells us that we can design our estimator and our controller separately, and then combine them to form the complete dynamical system.

(e) (3 points) Compute the response of the entire closed-loop system for the initial condition

$$x(0) = \begin{bmatrix} 5 \\ -5 \end{bmatrix}$$

and for the initial state estimate

$$\widehat{x}(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

over a suitable time horizon. Doing so is easy using the matrix exponential, but you may use any other approach you like. Plot the following:

- x_1 and $\widehat{x}_1$ as functions of time on the same figure;
- x_2 and $\widehat{x}_2$ as functions of time on the same figure;
- u as a function of time.

Using `ode45` (see problem 1):

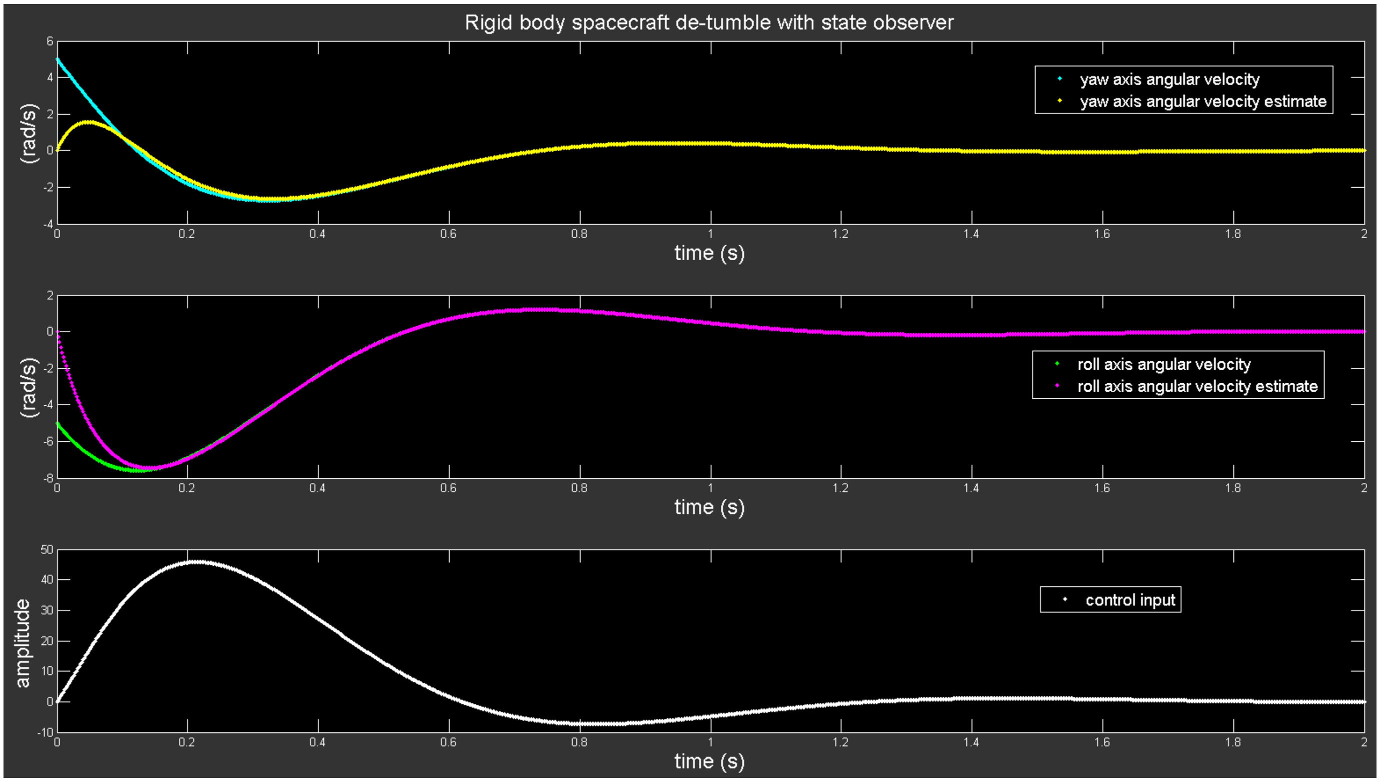


Figure 3: Rigid body spacecraft de-tumble using a state estimator.

(f) (2 points) Implement your controller and observer by modifying the function

`[u,userdata] = ControlLoop(y,userdata,params)`

in the MATLAB script `hw5prob02.m`. Submit only your `ControlLoop` function (this should be the only part of the code that you change) and a snapshot of the figure after the simulation has ended.

```
function hw5prob02
clear all
clc;

% %%%%%%%%%%%%%%%%%%
% PARAMETERS
%
% - spin rate
params.n = 10;
```

```
% - transverse and axial moments of inertia
params.Jt = 10000;
params.Ja = 1000;
% - moment of inertia matrix
params.J = diag([params.Jt,params.Jt,params.Ja]);
% - relative spin rate
params.lambda = ((params.Jt-params.Ja)/params.Jt)*params.n;
% - state-space system
params.A = [0 params.lambda; -params.lambda 0];
params.B = [1;0];
params.C = [0 1];

params.K = acker(params.A,params.B,[-3+5*1i -3-5*1i]);
params.xhat0 = [0;0];

%observer design
sig1 = 15;
sig2 = 15;

L1 = (params.lambda^2 - sig1*sig2)/params.lambda;
L2 = sig1 + sig2;

params.L = [L1;L2];

% - time step
params.dt = 2e-2;
%
% %%%%%%%%%%%%%%%%%%

RunSimulation(5,params);

function [u,userdata] = ControlLoop(y,userdata,params)

persistent isFirstTime
if isempty(isFirstTime)
    isFirstTime = false;
    fprintf(1,'initialize control loop\n');
    userdata.xhat = params.xhat0;
end

u = -params.K*userdata.xhat;
userdata.xhat = userdata.xhat + params.dt*(params.A*userdata.xhat ...
                + params.B*u - params.L*(params.C*userdata.xhat - y));
```

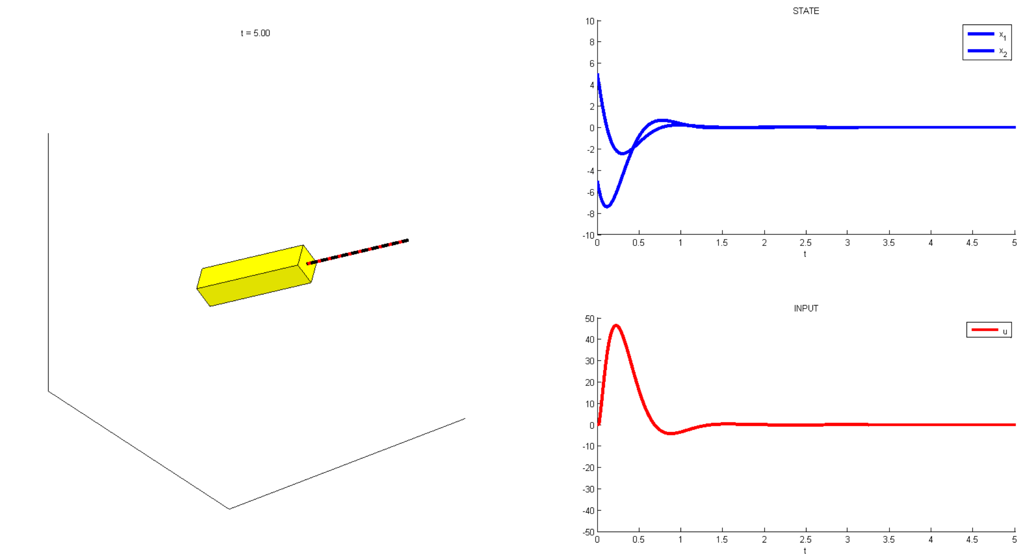


Figure 4: Rigid body spacecraft de-tumble simulation using a state estimator.

3. You have seen that the pitch motion of a spacecraft in LEO with a reaction wheel can be described in state-space form as

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 \\ \lambda & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1/J_{\text{pitch}} \\ -1/J_{\text{wheel}} \end{bmatrix} u,$$

where:

- x_1 is the pitch angle;
- x_2 is the pitch angular velocity;
- x_3 is the spin rate of the reaction wheel relative to the spacecraft;
- u is the applied torque (between the wheel and the spacecraft);
- $\lambda = 3n^2(J_{\text{yaw}} - J_{\text{roll}})/J_{\text{pitch}}$ is a constant;
- $n = 0.0011$ rad/sec is the orbital angular velocity;
- $J_{\text{pitch}} = 25 \text{ kg} \cdot \text{m}^2$, $J_{\text{roll}} = 15 \text{ kg} \cdot \text{m}^2$, $J_{\text{yaw}} = 5 \text{ kg} \cdot \text{m}^2$, and $J_{\text{wheel}} = 1 \text{ kg} \cdot \text{m}^2$ are moments of inertia.

In this problem, you will see how useful it is to be able to test that a system is observable before you actually try to implement an observer.

(a) (0.5 point) Suppose you can measure the pitch angle:

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

Show, any way you like, that the resulting system is not observable.

The system is observable iff its associated observability matrix has full *ROW* rank.

$$\begin{aligned} \mathcal{W}_o &= \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \lambda & 0 & 0 \end{bmatrix} \end{aligned}$$

$$RREF\left(\mathcal{W}_o{}^T\right) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$\therefore \mathcal{W}_o$ is rank deficient $\implies$ the system is NOT observable.

(b) (0.5 points) Suppose you can measure the pitch angular velocity:

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

Show, any way you like, that the resulting system is not observable.

$$
\begin{aligned}
\mathcal{W}_o &= \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \end{bmatrix} \\
&= \begin{bmatrix} 0 & 1 & 0 \\ \lambda & 0 & 0 \\ 0 & \lambda & 0 \end{bmatrix}
\end{aligned}
$$

$$
RREF\left(\mathcal{W}_o{}^T\right) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}
$$

$\therefore \mathcal{W}_o$ is rank deficient $\implies$ the system is NOT observable.

(c) (0.5 points) Suppose you can measure the spin rate of the reaction wheel relative to the spacecraft:

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

Show, any way you like, that the resulting system is not observable.

$$
\begin{aligned}
\mathcal{W}_o &= \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \end{bmatrix} \\
&= \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}
\end{aligned}
$$

$\therefore \mathcal{W}_o$ is rank deficient $\implies$ the system is NOT observable.

(d) (0.5 points) Suppose you can measure the *absolute* spin rate of the reaction wheel (i.e., the angular velocity of the wheel with respect to an inertial reference frame):

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

Show, any way you like, that the resulting system *is* observable. (Can you explain this?)

$$
\begin{aligned}
\mathcal{W}_o &= \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \end{bmatrix} \\
&= \begin{bmatrix} 0 & 1 & 1 \\ \lambda & 0 & 0 \\ 0 & \lambda & 0 \end{bmatrix}
\end{aligned}
$$

$$RREF\left(\mathcal{W}_o{}^T\right) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$\therefore \mathcal{W}_o$ has full row rank $\implies$ the system is observable.

It is fairly straightforward to see why having only spacecraft angular position, angular velocity or reaction wheel angular velocity alone will not allow you to observe all of the system's states. If we have only angular position data, we can use finite differencing to compute angular velocity. Similarly if we have spacecraft angular velocity data, we could integrate this data to obtain angular position. In neither of these situations do we have any way of accessing or computing the rotational speed of the wheel. Conversely, if we have only rotational data for the wheel, the spacecraft could have any position and velocity state with respect to the inertial frame and the wheel would be unaffected. On the other hand, if we have knowledge of the rotational velocity of the spacecraft AND the wheel, we can easily determine the spacecraft's angular position at any future time. Note that we could also achieve an observable system by sampling the spacecraft's angular position and the angular velocity of the wheel, however, it is much more realistic from a hardware/sensor point of view that we be able to measure the angular velocity of the spacecraft as we are doing in this problem.

(e) (1 point) For the output in part (d), design an observer

$$\dot{\widehat{x}} = A\widehat{x} + Bu - L(C\widehat{x} - y)$$

with closed-loop eigenvalues at -5, -0.05, and -0.0005. Do so any way you like.

First, we need to determine the closed-loop state matrix of the observer $\mathbf{A}_{cl-o}$:

$$\begin{aligned} \mathbf{A}_{cl-o} = \mathbf{A} - \mathbf{LC} &= \begin{bmatrix} 0 & 1 & 0 \\ \lambda & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} \ell_1 \\ \ell_2 \\ \ell_3 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 1-\ell_1 & -\ell_1 \\ \lambda & -\ell_2 & -\ell_2 \\ 0 & -\ell_3 & -\ell_3 \end{bmatrix} \end{aligned}$$

Now, we determine the closed-loop characteristic equation for the observer:

$$det\left(\mathbf{A}_{cl-o} - s\mathbb{I}\right) = 0$$

$$\begin{vmatrix} -s & 1-\ell_1 & -\ell_1 \\ \lambda & -\ell_2 - s & -\ell_2 \\ 0 & -\ell_3 & -\ell_3 - s \end{vmatrix} = s^3 + (\ell_2 + \ell_3)s^2 + (\lambda\ell_1 - \lambda)s - \lambda\ell_3 = 0$$

If we want to place the two observer eigenvalues at $s = \{-\sigma_1, -\sigma_2, -\sigma_3\} = \{-5, -0.05, -0.0005\}$, we know that the characteristic equation must be:

$$(s+\sigma_1)(s+\sigma_2)(s+\sigma_3) = s^3 + (\sigma_1+\sigma_2+\sigma_3)s^2 + (\sigma_1\sigma_2+\sigma_1\sigma_3+\sigma_2\sigma_3)s + \sigma_1\sigma_2\sigma_3$$

We can now equate the coefficients of these two characteristic equations and solve for the observer gains:

$$\begin{aligned}
\therefore \ell_1 &= 1 + \frac{\sigma_1\sigma_2+\sigma_1\sigma_3+\sigma_2\sigma_3}{\lambda} = 173914.289256 \\
\therefore \ell_2 &= \sigma_1+\sigma_2+\sigma_3 + \frac{\sigma_1\sigma_2\sigma_3}{\lambda} = 81.037654 \\
\therefore \ell_3 &= -\frac{\sigma_1\sigma_2\sigma_3}{\lambda} = 86.088154
\end{aligned}$$

(f) (1 point) Design a controller

$$u = -K\widehat{x}$$

with closed-loop eigenvalues at -1, -0.01, and -0.0001. Do so any way you like.

```
Jp = 25;
Jr = 15;
Jy = 5;
Jw = 1;
n = 0.0011;

lmbda = 3*n^2*(Jy - Jr)/Jp;

A = [0 1 0;lmbda 0 0;0 0 0];
B = [0;1/Jp;-1/Jw];
C = [0 1 1];
D = 0;

Wo = [C;C*A;C*A^2];

%observer design
sig1 = 5;
sig2 = 0.05;
sig3 = 0.0005;

L1 = -(Jp*(sig1*sig2 + sig1*sig3 + sig2*sig3 ...
     - (3*n^2*(Jr - Jy))/Jp))/(3*n^2*(Jr - Jy));
L2 = sig1 + sig2 + sig3 - (Jp*sig1*sig2*sig3)/(3*n^2*(Jr - Jy));
L3 = (Jp*sig1*sig2*sig3)/(3*n^2*(Jr - Jy));

L = [L1;L2;L3];

%controller design
K = acker(A,B,[-1 -0.01 -0.0001]);
```

This results in the following gain values:

$$\mathbf{K} = \begin{bmatrix} k_1 & k_2 & k_3 \end{bmatrix} = \begin{bmatrix} \frac{279}{1105} & \frac{6452}{803} & -\frac{250}{363} \end{bmatrix} = \begin{bmatrix} 0.252489 & 8.034869 & -0.688705 \end{bmatrix}$$

(g) (3 points) Express the entire closed-loop system—with both observer and controller—in the form

$$\frac{d}{dt}\begin{bmatrix} x \\ \widehat{x} \end{bmatrix} = A_{\text{cl}} \begin{bmatrix} x \\ \widehat{x} \end{bmatrix}$$

for some matrix A_{cl}. What are the eigenvalues of this system? (You should use MATLAB to verify your answer, but by this time I think you see the pattern.)

$$\begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & -\mathbf{BK} \\ \mathbf{LC} & \mathbf{A} - \mathbf{LC} - \mathbf{BK} \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix}$$

The closed-loop state-space representation with an estimator for our system is:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{\hat{x}}_1 \\ \dot{\hat{x}}_2 \\ \dot{\hat{x}}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -\frac{3n^2(J_{roll}-J_{yaw})}{J_{pitch}} & 0 & 0 & -\frac{k_1}{J_{pitch}} & -\frac{k_2}{J_{pitch}} & -\frac{k_3}{J_{pitch}} \\ 0 & 0 & 0 & \frac{k_1}{J_{wheel}} & \frac{k_2}{J_{wheel}} & \frac{k_3}{J_{wheel}} \\ 0 & \ell_1 & \ell_1 & 0 & 1-\ell_1 & -\ell_1 \\ 0 & \ell_2 & \ell_2 & -\frac{k_1}{J_{pitch}} - \frac{3n^2(J_{roll}-J_{yaw})}{J_{pitch}} & -\ell_2 - \frac{k_2}{J_{pitch}} & -\ell_2 - \frac{k_3}{J_{pitch}} \\ 0 & \ell_3 & \ell_3 & \frac{k_1}{J_{wheel}} & \frac{k_2}{J_{wheel}} - \ell_3 & \frac{k_3}{J_{wheel}} - \ell_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \hat{x}_1 \\ \hat{x}_2 \\ \hat{x}_3 \end{bmatrix}$$

We can ask MATLAB to compute the eigenvalues of $\mathbf{A}_{cl}$ directly using `eig`

NOTE: The following MATLAB output is using short format, which significantly truncates the numerical values for the eigenvalues and eigenvectors.

```
[V,Lambda] = eig(Acl)

V =

    0.0005    0.0282    0.1956    0.6854    0.8060    0.6641
   -0.0026   -0.0282   -0.0098   -0.0069   -0.0004   -0.0001
    0.0640    0.7060    0.2446    0.1738    0.0686    0.2427
   -0.9959    0.0282    0.9176    0.6854    0.5824    0.6641
   -0.0030   -0.0282   -0.0094   -0.0069   -0.0115   -0.0001
    0.0645    0.7060    0.2443    0.1738    0.0797    0.2427


Lambda =

   -5.0000         0         0         0         0         0
         0   -1.0000         0         0         0         0
         0         0   -0.0500         0         0         0
         0         0         0   -0.0100         0         0
         0         0         0         0   -0.0005         0
         0         0         0         0         0   -0.0001
```

Note, that the eigenvalues of the combined closed-loop system are the union of the closed-loop controlled system and the closed-loop estimated system. This tells us that we can design our estimator and our controller separately, and then combine them to form the complete dynamical system.

(h) (3 points) Compute the response of the entire closed-loop system for the initial condition

$$x(0) = \begin{bmatrix} 0.00 \\ 0.01 \\ 0.00 \end{bmatrix}$$

and for the initial state estimate

$$\widehat{x}(0) = \begin{bmatrix} 0.500 \\ 0.009 \\ 0.001 \end{bmatrix}$$

over a suitable time horizon. Doing so is easy using the matrix exponential, but you may use any other approach you like. Plot x and $\widehat{x}$ as functions of time on the same figure. It may be easier for you to make sense of these results if you plot them over two different time horizons, one short (e.g., 1 hour) and one long (e.g., 1 day).

Although certainly not required, you might be interested by what happens if the initial condition and initial state estimate are changed, even by a little tiny bit. Try it!

Using `ode45` (see problem 1):

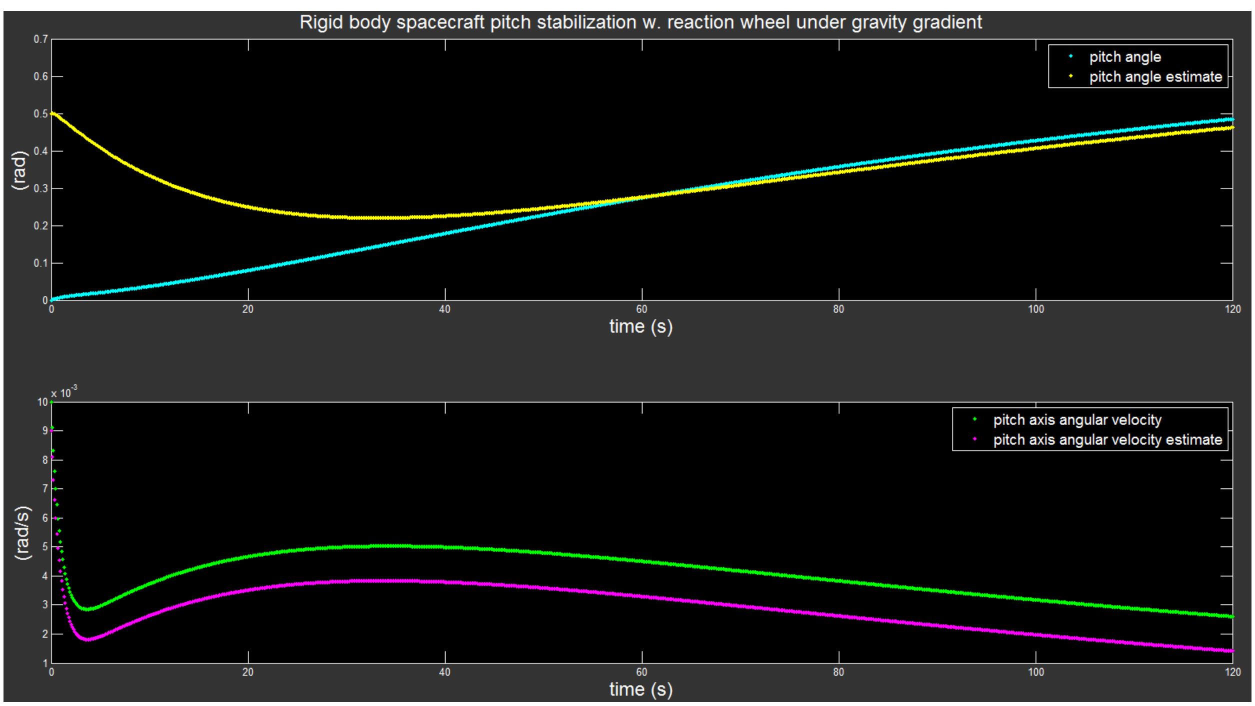


Figure 5: Rigid body spacecraft pitch stabilization using state estimation (2 minute horizon).

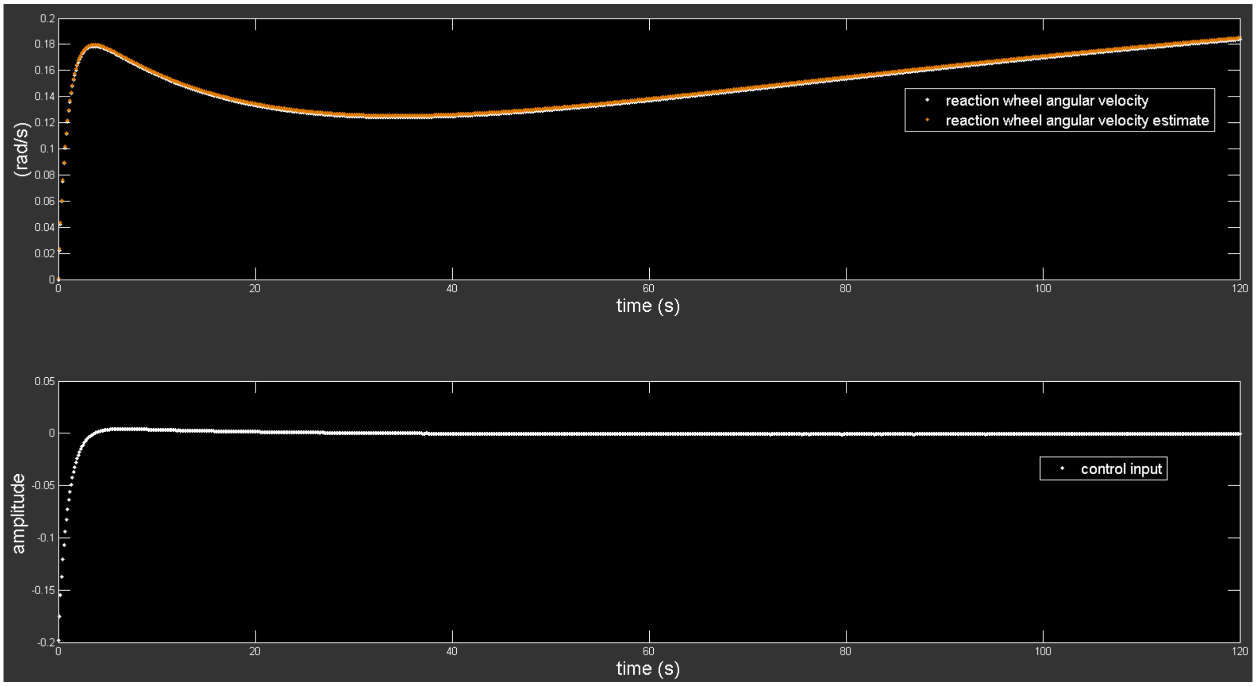


Figure 6: Rigid body spacecraft pitch stabilization using state estimation (2 minute horizon).

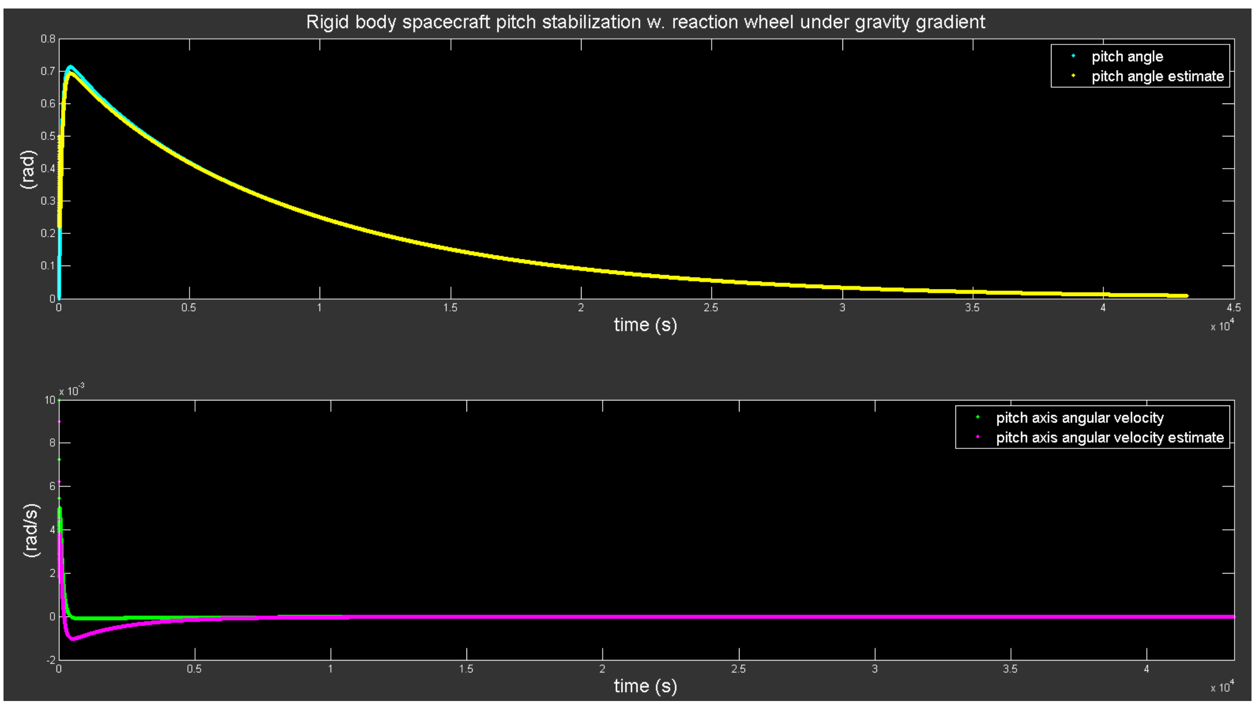


Figure 7: Rigid body spacecraft pitch stabilization using state estimation (half day horizon).

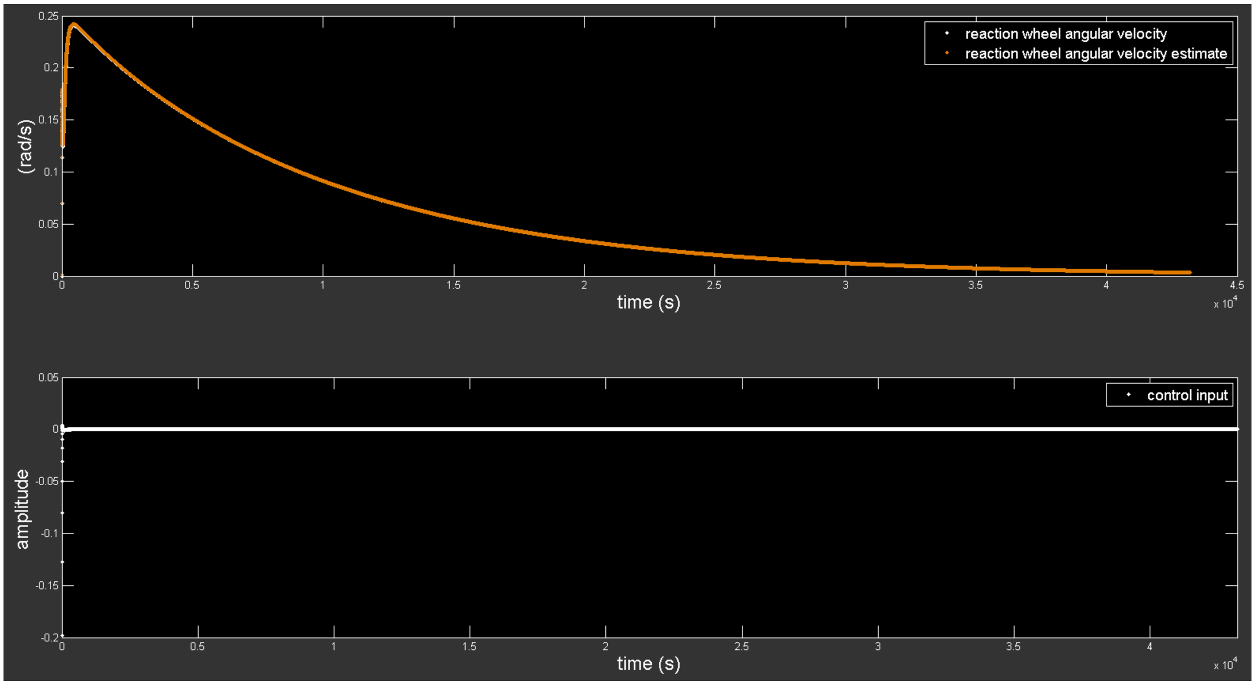
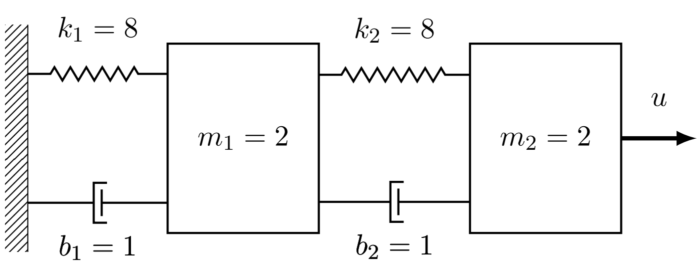


Figure 8: Rigid body spacecraft pitch stabilization using state estimation (half day horizon).



4. You have seen that the spring-mass-damper system shown above can be described in state-space form as

$$\dot{x} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -8 & 4 & -1 & 0.5 \\ 4 & -4 & 0.5 & -0.5 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.5 \end{bmatrix} u$$

where x_1 and x_2 are the absolute displacements of each mass from their equilibrium positions, x_3 and x_4 are the corresponding velocities of each mass, and u is the applied force.

(a) (1 point) Suppose you can measure the difference $x_2 - x_1$, i.e., the amount of stretch in the second spring. Express this measurement as

$$y = Cx$$

for an appropriate choice of C. Show that the resulting system is observable.

The system is observable iff its associated observability matrix has full ROW rank.

$$y = \mathbf{C}x = \begin{bmatrix} -1 & 1 & 0 & 0 \end{bmatrix}$$

$$\begin{aligned} \mathcal{W}_o &= \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \mathbf{CA}^3 \end{bmatrix} \\ &= \begin{bmatrix} -1.00 & 1.00 & 0 & 0 \\ 0 & 0 & -1.00 & 1.00 \\ 12.00 & -8.00 & 1.50 & -1.00 \\ -16.00 & 10.00 & 10.00 & -6.75 \end{bmatrix} \end{aligned}$$

$$RREF\left(\mathcal{W}_o{}^T\right) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$\therefore \mathcal{W}_o$ has full row rank $\implies$ the system is observable.

(b) (1 point) Design an observer

$$\dot{\widehat{x}} = A\widehat{x} + Bu - L(C\widehat{x} - y)$$

with all closed-loop eigenvalues at -25. Do so any way you like.

We could carry out the same coefficient matching procedure that we did for the first three problems, or we could let MATLAB do the heavy lifting and use `acker`. Using `acker` for observer eigenvalue placement requires a slightly different procedure though because we are placing the eigenvalues of:

$$\mathbf{A} - \mathbf{LC}$$

not,

$$\mathbf{A} - \mathbf{BK}$$

To do this, we note that the eigenvalues are invariant under the transpose operation, i.e.

$$eig(\mathbf{A} - \mathbf{LC}) = eig(\mathbf{A}^T - \mathbf{C}^T\mathbf{L}^T)$$

Therefore, we calculate the observer gains as follows:

```
A = [0 0 1 0;0 0 0 1;-8 4 -1 0.5;4 -4 0.5 -0.5];
B = [0;0;0;0.5];
C = [-1 1 0 0];
D = 0;
```

```
%observer design
sig1 = 25;
sig2 = 25;
sig3 = 25;
sig4 = 25;

L = acker(A',C',[-sig1 -sig2 -sig3 -sig4])';
```

Doing so yields the following observer gains:

$$\mathbf{L} = \begin{bmatrix} 3318.968750 \\ 3417.468750 \\ 94013.000000 \\ 97603.000000 \end{bmatrix}$$

Note, that there is nothing stopping us from performing the same coefficient matching that we have done in previous problems, however, the expressions for the observer gains for this four state system are fairly complicated:

$$\ell_1 = \frac{1}{4}\left(\sigma_1\sigma_2\sigma_3 + \sigma_1\sigma_2\sigma_4 + \sigma_1\sigma_3\sigma_4 + \sigma_2\sigma_3\sigma_4\right) - \left(\sigma_1 + \sigma_2 + \sigma_3 + \sigma_4\right) - \frac{\sigma_1\sigma_2\sigma_3\sigma_4}{32} + 1$$
$$\ell_2 = \frac{1}{4}\left(\sigma_1\sigma_2\sigma_3 + \sigma_1\sigma_2\sigma_4 + \sigma_1\sigma_3\sigma_4 + \sigma_2\sigma_3\sigma_4\right) - \frac{\sigma_1\sigma_2\sigma_3\sigma_4}{32} - \frac{1}{2}$$
$$\ell_3 = \sigma_1 + \sigma_2 + \sigma_3 + \sigma_4 - \left(\sigma_1\sigma_2 + \sigma_1\sigma_3 + \sigma_1\sigma_4 + \sigma_2\sigma_3 + \sigma_2\sigma_4 + \sigma_3\sigma_4\right) + \frac{\sigma_1\sigma_2\sigma_3\sigma_4}{4} + \frac{27}{4}$$
$$\ell_4 = \frac{\sigma_1\sigma_2\sigma_3\sigma_4}{4} - \frac{1}{2}\left(\sigma_1 + \sigma_2 + \sigma_3 + \sigma_4 +\right) - \frac{13}{4}$$

(c) (1 point) Design a controller

$$u = -K\widehat{x}$$

with closed-loop eigenvalues at -5. Do so any way you like.

Using `acker` again to place the controller's eigenvalues yields the following controller gain matrix:

$$\mathbf{K} = \begin{bmatrix} -36.562500 & 170.531250 & 135.937500 & 37.000000 \end{bmatrix}$$

$$
\begin{aligned}
k_1 &= \frac{1}{2}\left(\sigma_1\sigma_2\sigma_3 + \sigma_1\sigma_2\sigma_4 + \sigma_1\sigma_3\sigma_4 + \sigma_2\sigma_3\sigma_4\right) \\
&\quad - 4\left(\sigma_1\sigma_2 + \sigma_1\sigma_3 + \sigma_1\sigma_4 + \sigma_2\sigma_3 + \sigma_2\sigma_4 + \sigma_3\sigma_4\right) + \frac{7}{16}(\sigma_1\sigma_2\sigma_3\sigma_4) + 40 \\
k_2 &= 2(\sigma_1\sigma_2 + \sigma_1\sigma_3 + \sigma_1\sigma_4 + \sigma_2\sigma_3 + \sigma_2\sigma_4 + \sigma_3\sigma_4) \\
&\quad - \frac{1}{4}(\sigma_1\sigma_2\sigma_3 + \sigma_1\sigma_2\sigma_4 + \sigma_1\sigma_3\sigma_4 + \sigma_2\sigma_3\sigma_4) + \frac{1}{32}(\sigma_1\sigma_2\sigma_3\sigma_4) - 24 \\
k_3 &= \frac{1}{2}(\sigma_1\sigma_2\sigma_3 + \sigma_1\sigma_2\sigma_4 + \sigma_1\sigma_3\sigma_4 + \sigma_2\sigma_3\sigma_4) \\
&\quad - 4(\sigma_1 + \sigma_2 + \sigma_3 + \sigma_4) - \frac{1}{16}(\sigma_1\sigma_2\sigma_3\sigma_4) + 5 \\
k_4 &= 2(\sigma_1 + \sigma_2 + \sigma_3 + \sigma_4) - 3
\end{aligned}
$$

(d) (2 points) Express the entire closed-loop system—with both observer and controller—in the form

$$\frac{d}{dt}\begin{bmatrix} x \\ \widehat{x} \end{bmatrix} = A_{\mathrm{cl}} \begin{bmatrix} x \\ \widehat{x} \end{bmatrix}$$

for some matrix A_{cl}. What are the eigenvalues of this system? (You should use MATLAB to verify your answer, but by this time I think you see the pattern.)

$$\begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & -\mathbf{BK} \\ \mathbf{LC} & \mathbf{A} - \mathbf{LC} - \mathbf{BK} \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix}$$

The closed-loop state-space representation with an estimator for our system is:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{\hat{x}}_1 \\ \dot{\hat{x}}_2 \\ \dot{\hat{x}}_3 \\ \dot{\hat{x}}_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -8 & 4 & -1 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 4 & -4 & \frac{1}{2} & -\frac{1}{2} & \frac{585}{32} & -\frac{5457}{64} & -\frac{2175}{32} & -\frac{37}{2} \\ -\frac{106207}{32} & \frac{106207}{32} & 0 & 0 & \frac{106207}{32} & -\frac{106207}{32} & 1 & 0 \\ -\frac{58097}{17} & \frac{58097}{17} & 0 & 0 & \frac{58097}{17} & -\frac{58097}{17} & 0 & 1 \\ -94013 & 94013 & 0 & 0 & 94005 & -94009 & -1 & \frac{1}{2} \\ -97603 & 97603 & 0 & 0 & \frac{390501}{4} & -\frac{390769}{4} & -\frac{2159}{32} & -19 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \hat{x}_1 \\ \hat{x}_2 \\ \hat{x}_3 \\ \hat{x}_4 \end{bmatrix}$$

(e) (2 points) Compute the response of the entire closed-loop system for the initial condition and initial state estimate

$$x(0) = \begin{bmatrix} 1 \\ 3 \\ 0 \\ 0 \end{bmatrix} \qquad \widehat{x}(0) = \begin{bmatrix} 1.1 \\ 2.9 \\ -0.1 \\ 0.1 \end{bmatrix}$$

over a suitable time horizon. Doing so is easy using the matrix exponential, but you may use any other approach you like. Plot the following:

- y as a function of time;
- u as a function of time.
- x_1, $\widehat{x}_1$, x_2, and $\widehat{x}_2$ as functions of time on the same figure.

Using `ode45` (see problem 1):

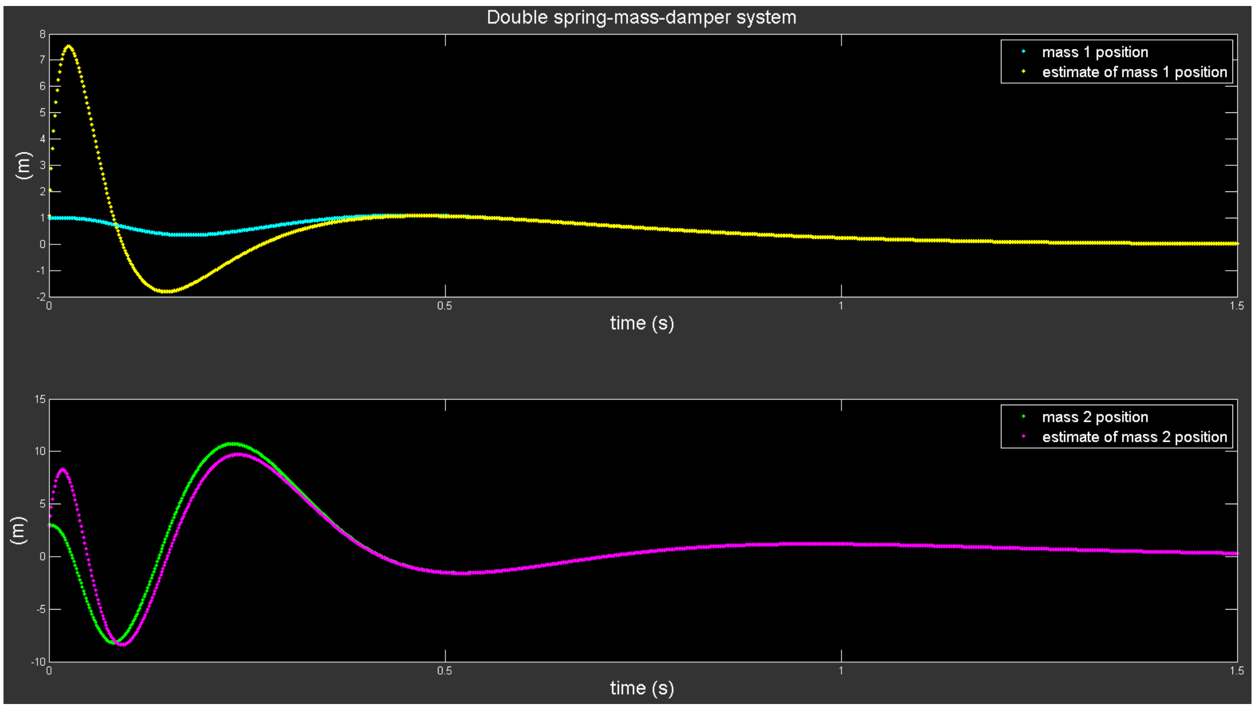


Figure 9: Double spring-mass-damper system mass positions (true and estimated).

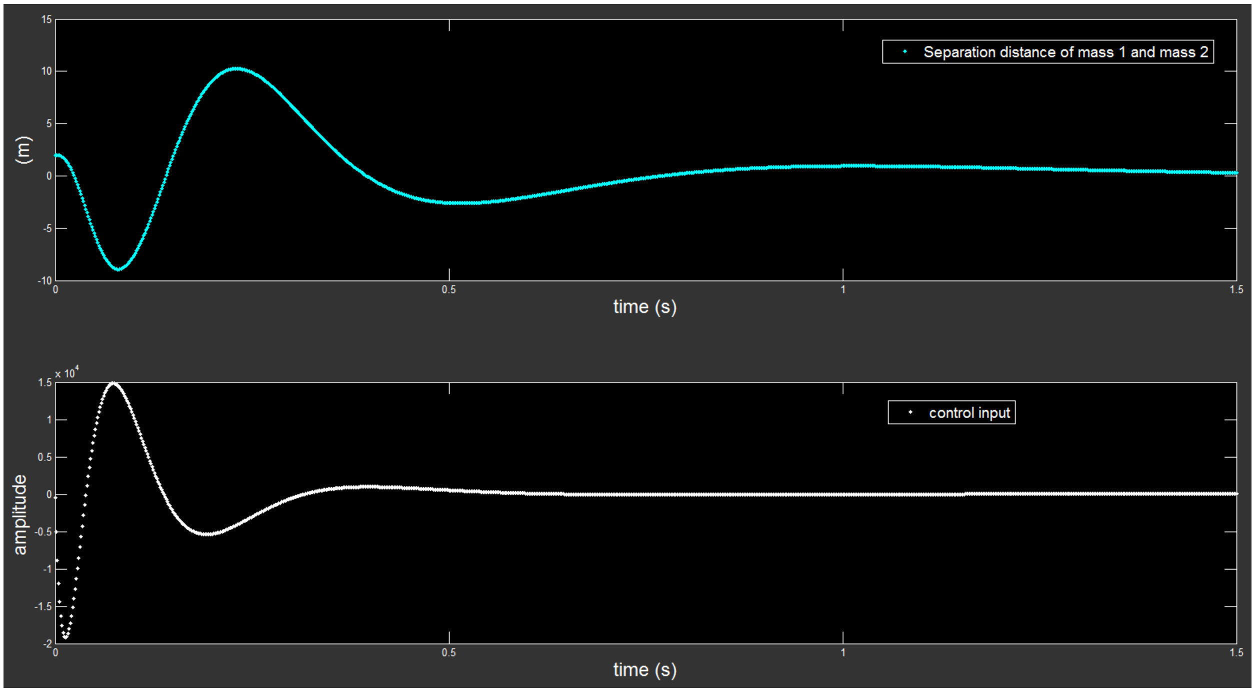


Figure 10: Double spring-mass-damper system output and control input.

(f) (2 points) Implement your controller and observer by modifying the function

```
[u,userdata] = ControlLoop(y,userdata,params)
```

in the MATLAB script `hw5prob04.m`. Submit only your `ControlLoop` function (this should be the only part of the code that you change) and a snapshot of the figure after the simulation has ended. Why are your results so terrible? Why are they so much different from what you obtained in part (e)? Briefly explain.

HINT: run `hw5prob04.m` again with zero input (i.e., with `u=0` in `ControlLoop`). What do you notice about the output response?

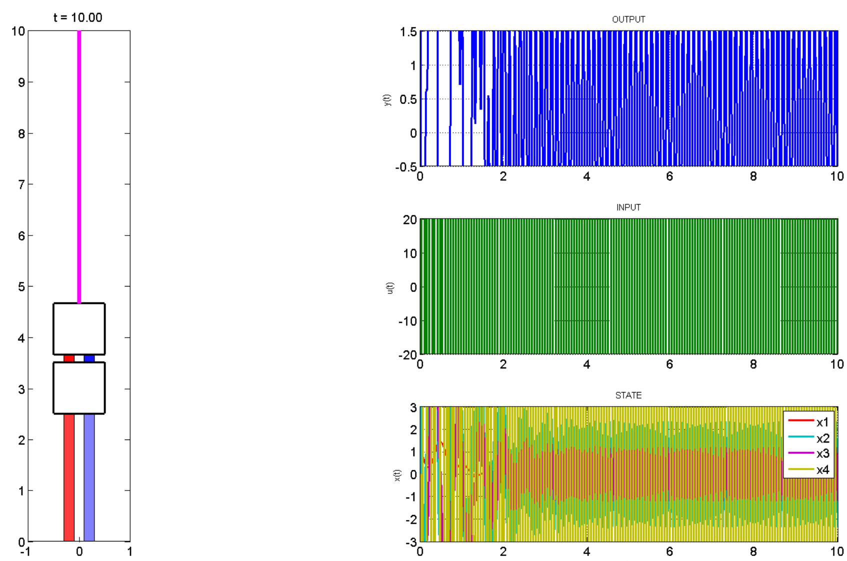


Figure 11: Double spring-mass-damper system simulation with observer eigenvalues at -25 and controller eigenvalues at -5.

If we examine the control output in Figure 10, we notice that we are requesting a VERY large control input early on in order to achieve steady-state as quickly as possible. This is in response to the very fast eigenvalues of the observer. When just considering the system on paper, as we did in the previous parts, we don't fully appreciate what we are doing to the system when we actually have to implement our controller/observer designs, i.e. we may not notice that the control input is enormous and the settling time is extremely fast, both of which would most likely be damaging to any mechanical system like the one we are considering in this problem. Notice that if we just let the dynamics play out naturally in the absense of control, the system is well-behaved. Since we are also using a discretized approximation to the control law, we might not expect the output to be exactly the same as when we are considering the continuous-time system.

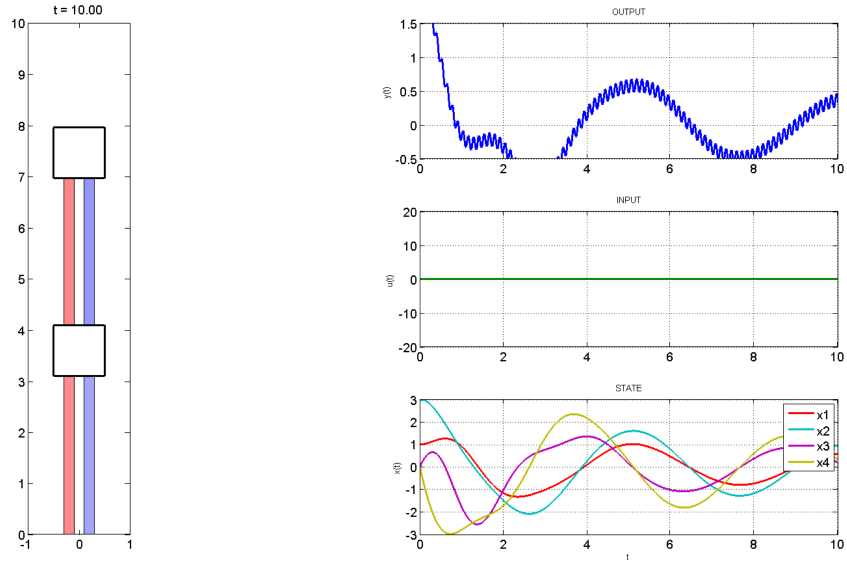


Figure 12: Double spring-mass-damper system simulation with no control input.

(g) (1 point) Redesign your observer to get better performance. In particular, repeat the following steps until you are satisfied:

- choose different locations for the observer eigenvalues;
- compute the gain matrix L that results in these eigenvalues;
- implement your observer and test using `hw5prob04.m`.

Briefly explain the design process that you used (e.g., how did you define "better" performance? how did you choose the eigenvalue locations?). You need only submit the `ControlLoop` function and the figure for your final design.

If we move the observer eigenvalues closer to the origin, we get a slower response from the system, but we are also not taxing the control input too much. The system takes a bit longer to reach steady-state, but it does so in a much smoother manner. If the observer eigenvalues are all placed at -5, then we get the following observer gain matrix:

$$\mathbf{L} = \begin{bmatrix} 86.468750 \\ 104.968750 \\ 33.000000 \\ 143.000000 \end{bmatrix}$$

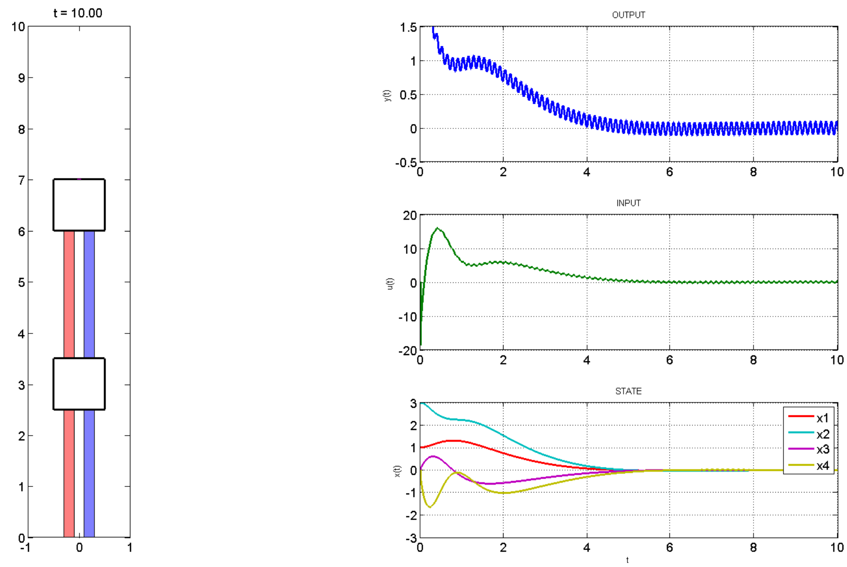


Figure 13: Double spring-mass-damper system simulation with observer eigenvalues at -5 and controller eigenvalues at -2.