EECS 16B Designing Information Devices and Systems II
Spring 2017 Murat Arcak and Michel Maharbiz **Midterm 2**
Practice Questions

1. **Textbook Sols**

   (a) Exercise 3.6

Exercise 2.6 (Normalized oscillator dynamics) Consider a damped spring–mass system with dynamics

$$m\ddot{q} + c\dot{q} + kq = F.$$

Let $\omega_0 = \sqrt{k/m}$ be the natural frequency and $\zeta = c/(2\sqrt{km})$ be the damping ratio.

(a) Show that by rescaling the equations, we can write the dynamics in the form

$$\ddot{q} + 2\zeta\omega_0\dot{q} + \omega_0^2 q = \omega_0^2 u, \tag{S2.1}$$

where $u = F/k$. This form of the dynamics is that of a linear oscillator with natural frequency ω_0 and damping ratio ζ.

(b) Show that the system can be further normalized and written in the form

$$\frac{dz_1}{d\tau} = z_2, \qquad \frac{dz_2}{d\tau} = -z_1 - 2\zeta z_2 + v. \tag{S2.2}$$

The essential dynamics of the system are governed by a single damping parameter ζ. The *Q-value* defined as $Q = 1/2\zeta$ is sometimes used instead of ζ.

(c) Show that the solution for the unforced system ($v = 0$) with no damping ($\zeta = 0$) is given by

$$z_1(\tau) = z_1(0)\cos\tau + z_2(0)\sin\tau, \qquad z_2(\tau) = -z_1(0)\sin\tau + z_2(0)\cos\tau.$$

Invert the scaling relations to find the form of the solution $q(t)$ in terms of $q(0)$, $\dot{q}(0)$ and ω_0.

(d) Consider the case where $\zeta = 0$ and $u(t) = \sin\omega t$, $\omega > \omega_0$. Solve for $z_1(\tau)$, the normalized output of the oscillator, with initial conditions $z_1(0) = z_2(0) = 0$ and use this result to find the solution for $q(t)$.

Solution.[S. Fuller, 2007; modified by S. Han, Oct 2008]

(a) Dividing (2.6) by m and introducing $u = F/k$ gives

$$\ddot{q} + \frac{c}{m}\dot{q} + \frac{k}{m}q = \frac{k}{m}u$$

introducing $\omega_0 = \sqrt{k/m}$ and $\zeta = \dfrac{c}{2\sqrt{km}}$ gives (S2.1).

(b) Let $\tau = \omega_0 t$ be scaled time and let $z_1 = \omega_0 q$, $z_2 = \dot{q}$, and $v = \omega_0 u$. Then

$$\frac{dz_1}{d\tau} = \frac{d(\omega_0 q)}{d(\omega_0 t)} = \frac{dq}{dt} = \dot{q} = z_2$$

and from equation (S2.1)

$$\frac{dz_2}{d\tau} = \frac{\ddot{q}}{\omega_0} = -2\zeta\dot{q} - \omega_0 q + \omega_0 u = -2\zeta z_2 - z_1 + v,$$

which gives equation (S2.2).

(c) If the damping is zero, the normalized solution is a simple oscillator and the form of the solution can be verified by substition. Inverting the scaling relations yields

$$q(t) = z_1/\omega_0 = q(0)\cos\omega_0 t + (\dot{q}(0)/\omega_0)\sin\omega_0 t$$
$$\dot{q}(t) = z_2 = -q(0)\omega_0\sin\omega_0 t + \dot{q}(0)\cos\omega_0 t.$$

This solution can also be verified by direct substitution into the original dynamics.

(d) Let $\eta = \omega/\omega_0$ Written in second order form, the normalized equations have the form

$$\ddot{z}_1 + z_1 = v = \omega_0 \sin\eta\tau.$$

We look for a particular solution of the form $z_1 = a\sin\eta\tau + b\cos\eta\tau$. Substitution into the differential equation and equating the sine and cosine terms yields $a = \omega_0/(1-\eta^2)$ and $b = 0$, so the particular solution is given by

$$z_1 = \frac{\omega_0}{1-\eta^2}\sin\eta\tau, \qquad z_2 = \frac{\omega_0\eta}{1-\eta^2}\cos\eta\tau.$$

Since the particular solution is nonzero $\tau = 0$, we must add a homogeneous solution of the form given in part (c). The resulting solution is given by

$$z_1 = \frac{\omega_0}{1-\eta^2}(\sin\eta\tau - \eta\sin\tau), \qquad z_2 = \frac{\omega_0\eta}{1-\eta^2}(\cos\eta\tau - \cos\tau).$$

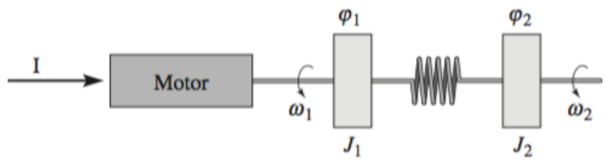
The solution can be written in the original coordinates by rescaling as in part (c). The resulting solution is

$$q = \frac{1}{\omega_0^2-\omega^2}(\omega_0^2\sin\omega t - \omega_0\omega\sin\omega_0 t), \qquad \dot{q} = \frac{\omega\omega_0^2}{\omega_0^2-\omega^2}(\cos\omega t - \cos\omega_0 t).$$

Note that as the forcing frequency ω approaches the natural frequency ω_0, the amplitude of the response grows.

(b) Exercise 3.10

Exercise 2.10 (Motor drive) Consider a system consisting of a motor driving two masses that are connected by a torsional spring, as shown in the diagram below.



This system can represent a motor with a flexible shaft that drives a load. Assuming that the motor delivers a torque that is proportional to the current, the dynamics of the system can be described by the equations

$$\begin{aligned} J_1\frac{d^2\varphi_1}{dt^2} + c\left(\frac{d\varphi_1}{dt} - \frac{d\varphi_2}{dt}\right) + k(\varphi_1 - \varphi_2) &= k_I I, \\ J_2\frac{d^2\varphi_2}{dt^2} + c\left(\frac{d\varphi_2}{dt} - \frac{d\varphi_1}{dt}\right) + k(\varphi_2 - \varphi_1) &= T_d. \end{aligned} \tag{S2.8}$$

Similar equations are obtained for a robot with flexible arms and for the arms of DVD and optical disk drives.

Derive a state space model for the system by introducing the (normalized) state variables $x_1 = \varphi_1, x_2 = \varphi_2$, $x_3 = \omega_1/\omega_0$, and $x_4 = \omega_2/\omega_0$, where $\omega_0 = \sqrt{k(J_1 + J_2)/(J_1 J_2)}$ is the undamped natural frequency of the system when the control signal is zero.

Solution.[S. Han, Apr 08] Introducing the state variables $x_1 = \varphi_1, x_2 = \varphi_2, x_3 = \omega_1/\omega_0$, and $x_4 = \omega_2/\omega_0$ and substituting them into equation (S2.8) give

$$\begin{aligned} J_1\ddot{x}_1 + c(\dot{x}_1 - \dot{x}_2) + k(x_1 - x_2) &= k_I I, \\ J_2\ddot{x}_2 + c(\dot{x}_2 - \dot{x}_1) + k(x_2 - x_1) &= T_d. \end{aligned}$$

Therefore

$$\begin{aligned} \frac{dx_1}{dt} &= \dot{\varphi}_1 = \omega_1 = \omega_0 x_3, \quad \frac{dx_2}{dt} = \omega_0 x_4, \\ \frac{dx_3}{dt} &= \frac{\ddot{x}_1}{\omega_0} = \frac{1}{\omega_0 J_1}(-c\dot{x}_1 + c\dot{x}_2 - kx_1 + kx_2 + k_I I) \\ &= -\frac{k}{\omega_0 J_1}x_1 + \frac{k}{\omega_0 J_1}x_2 - \frac{c}{J_1}x_3 + \frac{c}{J_1}x_4 + \frac{k_I}{\omega_0 J_1}I, \\ \frac{dx_4}{dt} &= \frac{\ddot{x}_2}{\omega_0} = \frac{1}{\omega_0 J_2}(-c\dot{x}_2 + c\dot{x}_1 - kx_2 + kx_1 + T_d) \\ &= \frac{k}{\omega_0 J_2}x_1 - \frac{k}{\omega_0 J_2}x_2 + \frac{c}{J_2}x_3 - \frac{c}{J_2}x_4 + \frac{1}{\omega_0 J_2}T_d. \end{aligned}$$

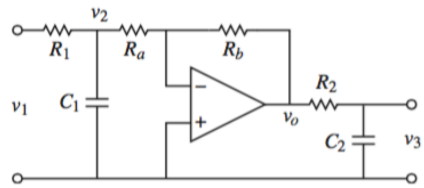
Rewrite in state-space form:

$$\dot{x} = \begin{bmatrix} 0 & 0 & \omega_0 & 0 \\ 0 & 0 & 0 & \omega_0 \\ -\frac{k}{\omega_0 J_1} & \frac{k}{\omega_0 J_1} & -\frac{c}{J_1} & \frac{c}{J_1} \\ \frac{k}{\omega_0 J_2} & -\frac{k}{\omega_0 J_2} & \frac{c}{J_2} & -\frac{c}{J_2} \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ \frac{k_I}{\omega_0 J_1} \\ 0 \end{bmatrix} I + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{\omega_0 J_2} \end{bmatrix} T_d$$

Supplemental Exercises.

(c) Exercise 4.4

Exercise 3.4 (Operational amplifier circuit) Consider the op amp circuit shown below.



Show that the dynamics can be written in state space form as

$$\frac{dx}{dt} = \begin{bmatrix} -\frac{1}{R_1C_1} - \frac{1}{R_aC_1} & 0 \\ \frac{R_b}{R_a}\frac{1}{R_2C_2} & -\frac{1}{R_2C_2} \end{bmatrix} x + \begin{bmatrix} \frac{1}{R_1C_1} \\ 0 \end{bmatrix} u, \quad y = \begin{bmatrix} 0 & 1 \end{bmatrix} x,$$

where $u = v_1$ and $y = v_3$. (Hint: Use v_2 and v_3 as your state variables.)

Solution.[S. Han, Apr 08] Letting v_2 and v_3 represent the states, we can write down the "nodal equations", which state that the total current at any node in the circuit must be zero. This yields three equations:

$$C_1\dot{v}_2 = \frac{v_1 - v_2}{R_1} - \frac{v_2}{R_a}, \qquad \frac{v_2}{R_a} = \frac{v_o}{R_b}, \qquad C_2\dot{v}_3 = \frac{v_o - v_3}{R_2} = \frac{-v_3}{R_2} + \frac{R_b v_2}{R_a R_2},$$

where the last equality follows by solving the second equation for v_o and substituting it into the third equation. Let $x_1 = v_2, x_2 = v_3, u = v_1$ and $y = v_3$:

$$\dot{x}_1 = \dot{v}_2 = \Big(-\frac{1}{R_1C_1} - \frac{1}{R_aC_1}\Big)v_2 + \frac{1}{R_1C_1}v_1 = \Big(-\frac{1}{R_1C_1} - \frac{1}{R_aC_1}\Big)x_1 + \frac{1}{R_1C_1}u,$$
$$\dot{x}_2 = \dot{v}_3 = \frac{R_b}{R_a}\frac{1}{R_2C_2}v_2 - \frac{1}{R_2C_2}v_3 = \frac{R_b}{R_a}\frac{1}{R_2C_2}x_1 - \frac{1}{R_2C_2}x_2.$$

Written in state space form,

$$\frac{dx}{dt} = \begin{bmatrix} -\frac{1}{R_1C_1} - \frac{1}{R_aC_1} & 0 \\ \frac{R_b}{R_a}\frac{1}{R_2C_2} & -\frac{1}{R_2C_2} \end{bmatrix} x + \begin{bmatrix} \frac{1}{R_1C_1} \\ 0 \end{bmatrix} u, \quad y = \begin{bmatrix} 0 & 1 \end{bmatrix} x.$$

(d) Exercise 6.9

Exercise 5.9 (Keynesian economics) Consider the following simple Keynesian macroeconomic model in the form of a linear discrete-time system discussed in Exercise 5.8:

$$\begin{bmatrix} C[t+1] \\ I[t+1] \end{bmatrix} = \begin{bmatrix} a & a \\ ab-b & ab \end{bmatrix} \begin{bmatrix} C[t] \\ I[t] \end{bmatrix} + \begin{bmatrix} a \\ ab \end{bmatrix} G[t],$$
$$Y[t] = C[t] + I[t] + G[t].$$

Determine the eigenvalues of the dynamics matrix. When are the magnitudes of the eigenvalues less than 1? Assume that the system is in equilibrium with constant values capital spending C, investment I and government expenditure G. Explore what happens when government expenditure increases by 10%. Use the values $a = 0.25$ and $b = 0.5$.

Solution. The eigenvalues of the dynamics matrix can be found by solving for the roots of

$$\det(\lambda I - A) = \begin{vmatrix} \lambda - a & -a \\ a - ab & \lambda - ab \end{vmatrix} = \lambda^2 - a(b+1)\lambda + a^2,$$

which gives

$$\lambda_{1,2} = \frac{a(b+1) \pm \sqrt{a^2((b+1)^2-4)}}{2}.$$

To guarantee $|\lambda_{1,2}| < 1$, we need to have

(a) If $(b+1)^2 \leq 4$, the eigenvalues are complex and $|\lambda_1| = |\lambda_2|$, so $|\lambda_{1,2}| < 1 \iff a^2 = |\lambda_1\lambda_2| = |\lambda_1^2| < 1 \iff |a| < 1$.

(b) If $(b+1)^2 > 4$, the eigenvalues are real. In this case we need

$$\left|\frac{a(b+1) + \sqrt{a^2((b+1)^2-4)}}{2}\right| < 1 \quad \text{and} \quad \left|\frac{a(b+1) - \sqrt{a^2((b+1)^2-4)}}{2}\right| < 1$$

Assuming $a, b > 0$, the first eigenvalue is larger and the condition becomes

$$a\left(\frac{(b+1) + \sqrt{(b+1)^2-4}}{2}\right) < 1.$$

When the system is at equilibrium, C, I, and G are all constants, so

$$C = aC + aI + aG, \qquad I = a(b-1)C + abI + abG.$$

We can express C and I using G:

$$C = \frac{a}{a^2 - ab - a + 1}G, \qquad I = \frac{ab - a^2}{a^2 - ab - a + 1}G.$$

When there is a 10% increase in G, using $a = 0.25$ and $b = 0.5$, we can calculate that C will increase by 3.65% while I will only increase by 0.91%.

(e) Exercise 6.10

Exercise 5.10 Consider a scalar system

$$\frac{dx}{dt} = 1 - x^3 + u.$$

Compute the equilibrium points for the unforced system ($u = 0$) and use a Taylor series expansion around the equilibrium point to compute the linearization. Verify that this agrees with the linearization in equation (5.33).

Solution. The point $(x_e, u_e) = (1, 0)$ is an equilibrium point for this system and we can thus set

$$z = x - 1 \qquad v = u.$$

We can now compute the equations in these new coordinates as

$$\begin{aligned}
\dot{z} &= \frac{d}{dt}(x - 1) = \dot{x} \\
&= 1 - x^3 + u = 1 - (z + 1)^3 + v \\
&= 1 - z^3 - 3z^2 - 3z - 1 + v = -3z - 3z^2 - z^3 + v.
\end{aligned}$$

If we now assume that x stays very close to the equilibrium point, then $z = x - x_e$ is small and $z \ll z^2 \ll z^3$. We can thus *approximate* our system by a *new* system

$$\dot{z} = -3z + v.$$

This set of equations should give behavior that is close to that of the original system as long as z remains small.

(f) Exercise 7.12

Exercise 6.11 (Motor drive) Consider the normalized model of the motor drive in Exercise 2.10. Using the following normalized parameters,

$$J_1 = 10/9, \qquad J_2 = 10, \qquad c = 0.1, \qquad k = 1, \qquad k_I = 1,$$

verify that the eigenvalues of the open loop system are $0, 0, -0.05 \pm i$. Design a state feedback that gives a closed loop system with eigenvalues $-2, -1$ and $-1 \pm i$. This choice implies that the oscillatory eigenvalues will be well damped and that the eigenvalues at the origin are replaced by eigenvalues on the negative real axis. Simulate the responses of the closed loop system to step changes in the command signal for θ_2 and a step change in a disturbance torque on the second rotor.

Solution. It is useful to write a script that sets all parameters. This is accomplished by the script below:

```
%robotarm_data.m
%Data for the normalized motor drive
%   kja 070820
%Parameters
J1=10/9;J2=10;k=1;kd=0.1;ki=5;
w0=sqrt(k*(J1+J2)/J1/J2);a1=J2/(J1+J2);a2=J1/(J1+J2);
b1=kd/(w0^2*J1);b2=kd/(w0^2*J2);
g1=ki/(w0^2*J1);g2=1/(w0^2*J2);
%System matrices
A=[0 0 1 0;0 0 0 1;-a1 a1 -b1 b1;a2 -a2 b2 -b2]*w0;
B1=[0;0;g1;0];B2=[0;0;0;g2];B=[B1 B2];
C1=[1 0 0 0];C2=[0 1 0 0];D1=0;C=[C1;C2];D=zeros(2,2);
%SISO system where output is angle of first rotor
sys1=ss(A,B1,C1,D1);
%SISO system where output is angle of second rotor
sys2=ss(A,B2,C2,D1);
%Complete system
sys=ss(A,B,C,D);
```

The state feedback is given by Theorem 6.3, in component form we have

$$u = -k_1x_1 - k_2x_2 - k_3x_3 - k_4x_4 + k_rx_r$$

The numerical values are obtained using the following script:

```
%robotarm_sfb.m - Design of State feedback for robotarm
%   kja 070820
%Get process model
motordrivedata;B=B1;C=C2;
%Desired closed loop poles
P=[-1 -2 -1+i -1-i];
K=acker(A,B,P);Acl=A-B*K;
Kr=-1/(C*inv(Acl)*B);
disp('Feedback gain L=');disp(K)
disp('Refererence gain Lr=');disp(Kr)
%Check the results
n=size(A);disp('Specified closed loop eigenvalues')
disp(P);
disp('Closed loop eigenvalues obtained')
disp(eig(Acl)');
```

which gives the values $k_1 = 1.78$ $k_2 = 7.10$ $k_3 = 1.09$ and $k_4 = 20.24$ and $k_r = 8.89$. Notice that we have added a line where we compute the closed loop eigenvalues with the computed controller gains. It is a good practice to add checks of this type in the design programs.

To explore the properties of the system we will simulate it. A SIMULINK block diagram of the system is shown in Figure S6.3. Notice that the block is generic, the specifics are given by the matrices defining the process, the controller and the inputs. In the simulation we will first apply a unit step in the reference signal

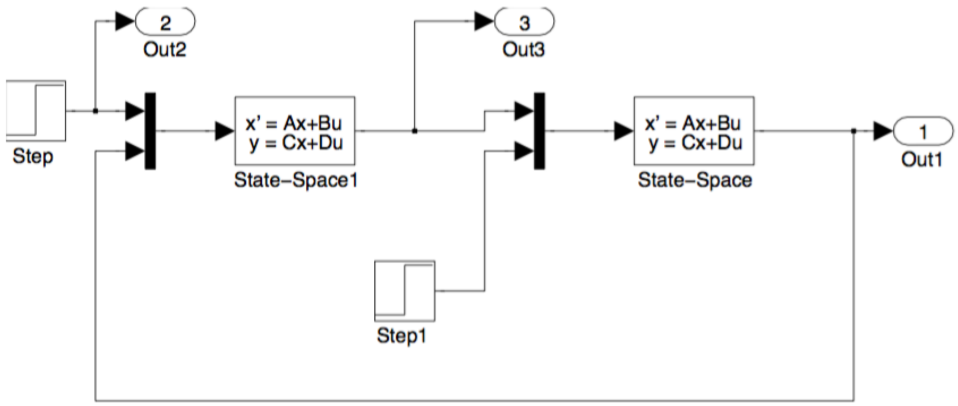


Figure S6.3: SIMULINK block diagram of the system with a controller.

and we will then apply a disturbance torque of 2 units at the second rotor. The simulation is executed using the script below:

```
% motordrive_sfbsim.m
%Simlation of motor drive with state feedback
%   kja 070820
%Get procees parameters and design state feedback
aminit;
motordrive_sfb;
Ap=A;Bp=[B B2];Cp=eye(n);Dp=zeros(4,2);
%Set controller parameters
Ac=0;Bc=zeros(1,5);Cc=0;Dc=[Kr -K];
%Execute simulation
r=1;d=2;ts=20;[t,x,y]=sim('motordrive_blk',[0 ts]);
subplot(311);
plot(t,y(:,1),'r--',t,y(:,2),'b',t,ones(size(t)),'r:','linew',2);
axis([0 20 -2 4]);grid on;
ylabel('y');
subplot(312);plot(t,y(:,6),t,zeros(size(t)),'r--','linew',2);
ylabel('u');xlabel('t');grid on;
axis([0 20 -2 10]);
amprint('-deps', 'dcmotor-statefbk-sfbsim.eps');
```

The results of the simulation are shown in Figure S6.4. The simulation shows that the second rotor responds very nicely to the reference signal. To achieve the fast response the first rotor has a large overshoot. This is necessary because the only way to exert a force on the second rotor is to turn the first rotor so that a large force is obtained. To execute effective control it is necessary to coordinate the motion of both rotors accurately. This is done very efficiently by the controller based on state feedback. To obtain the response to a unit step in the reference shown in the figure the initial value of the control signal is $8.89k_r$. The value is proportional to the magnitude of the step. To judge if the value is reasonable it it necessary to know the numerical values of typical steps and the signal level when the control signal saturates. If the a larger value of the control signal can be permitted the specified closed loop poles can be chose to be faster. If the value is too large the desired closed loop poles should have smaller magnitudes.

The response to a torque disturbance at the second rotor shows that there is a steady-state error. The reason for this is that the controller does not have integral action. Also notice that the angle φ_1 of the first rotor is negative. This is necessary because the only way to exert a torque on the second rotor is by turning the first rotor.

(g) Exercise 7.13

Exercise 6.12 (Whipple bicycle model) Consider the Whipple bicycle model given by equation (3.7) in Section 3.2. Using the parameters from the companion web site, the model is unstable at the velocity $v = 5$ m/s and the open loop eigenvalues are -1.84, -14.29 and $1.30 \pm 4.60i$. Find the gains of a controller that stabilizes the bicycle and gives closed loop eigenvalues at -2, -10 and $-1 \pm i$. Simulate the response of the system to a step change in the steering reference of 0.002 rad.

Find the controller gains corresponding to choosing the final pair of complex poles at $-1 \pm i$ as stated in the text, and also with these poles at $-2 \pm 2i$ and $-5 \pm 5i$. For each case, simulate the response to a step change in the steering reference of 0.002 rad and plot both the steering angle and the torque command.

Solution. The model is given by equation (3.7)

$$M \begin{bmatrix} \ddot{\varphi} \\ \ddot{\delta} \end{bmatrix} + Cv_0 \begin{bmatrix} \dot{\varphi} \\ \dot{\delta} \end{bmatrix} + (K_0 + K_2 v_0^2) \begin{bmatrix} \varphi \\ \delta \end{bmatrix} = \begin{bmatrix} 0 \\ T \end{bmatrix},$$

where φ is the tilt angle, δ the steering angle and T the steering torque. Introducing the state variables $x_1 = \varphi$, $x_2 = \delta$, $x_3 = \dot{\varphi}$ and $x_4 = \dot{\delta}$ the equations can be written in the state space form as

$$\frac{dx}{dt} = \Big[O, I; M^{-1}Cv_0 M^{-1}(K_0 + K_2 v_0^2) \Big] \, x + \Big[O; M^{-1}[0; T] \Big] \, u$$

where O is a 2×2 matrix of zeros I a 2×2 identity matrix. The system is now in state space format and it is straight forward to place the closed loop eigenvalues using Theorem 6.3.

The choice of closed loop eigenvalues given in the problem statement was based on the following argument. To obtain controller gains that are not too large we did not want to change the closed loop poles too much. Since two eigenvalues were in the right half-plane they have to be moved into the left half-plane placing those eigenvalues at $-1 \pm i$ give well damped eigenvalues that are not too fast. The remaining two eigenvalues were chosen close to the open loop eigenvalues.

The parameter values are generated by the script `bike_linmod` and the controller design is obtained by the script `bike_sfb`

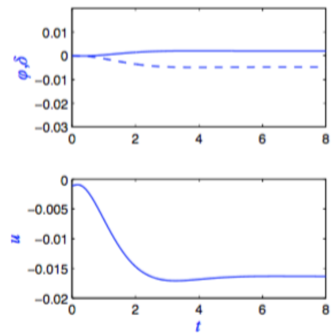
```
%bike_sfb.m
%Design of State feedback for Whipple bicycle
%   kja 070820
%   Get process model
bike_linmod;
%   Desired closed loop eigenvalues
P=[1*[-1+i -1-i] -2 -10];
K=acker(A,B,P);Acl=A-B*K;
C=[0 1 0 0];
Kr=-1/(C*inv(Acl)*B);
disp('Feedback gain K=');disp(K)
disp('Refererence gain Kr=');disp(Kr)
%   Check the results
n=size(A);disp('Specified closed loop eigenvalues')
disp(P);
disp('Closed loop eigenvalues obtained')
disp(eig(Acl)');
%   Collect matrices of closed loop system
bike_cls=ss(Acl,B*Kr,C,0);
```

The system is simulated using the script `bike_clstep`

```
% bike_sim.m
% Step response of closed loop system with command in steering angle
%   kja 070821
%   Get linear model and design state feedback
aminit;
bike_sfb;
%   Closed loop system response to reference signal (steering angle)
```

```
t=0:0.01:8;
[y,t,x]=step(bike_cls,t);
u=Kr-K*x';
subplot(321);
pl=plot(t,x(:,1)/500,'b--',t,x(:,2)/500,'b-');
set(pl,'Linewidth',1.5);grid on;
axis([0 8 -0.03 0.02]);
ylabel('x');
subplot(323);
pl=plot(t,u/500,'b-');
set(pl,'Linewidth',1.5);grid on;
amprint('-deps', 'bike-clstep.eps');
xlabel('t');ylabel('u');
```

The result of the simulation is shown in the figure below.



A step is of 0.002 rad/s is given in the steering angle reference. The steering angle δ which is shown by solid curves follows the command with a nice response without overshoot. When the signal settles the bicycle moves in a circle to the right with radius $v_0^2\delta/b$ where b is the wheel base. To remain in steady state the bicycle has to tilt to the left ($\varphi = -v_0^2\delta/(bg)$ to maintain steady state. When the bicycle tilts the interaction between the road and the front wheel creates a torque that tends to turn the front wheel to the left. This tendency has to be compensated by applying a negative steering torque on the front wheel. The fact that the initial steering torque is also to the left is a consequence of the effect called counter steering, see [**?**]. It is interesting to see that the eigenvalue assignment generates this automatically.

For the other eigenvalues, the line below is changed in bike_sfb.m:

```
P=[1*[-1+i -1-i] -2 -10];
```

The first 1 is changed to a 2 and then 5 and the simulation is re-run.

(h) Exercise 8.4

Exercise 7.4 (Bicycle dynamics) The linearized model for a bicycle is given in equation (3.5), which has the form

$$J\frac{d^2\varphi}{dt^2} - \frac{Dv_0}{b}\frac{d\delta}{dt} = mgh\varphi + \frac{mv_0^2 h}{b}\delta,$$

where φ is the tilt of the bicycle and δ is the steering angle. Give conditions under which the system is observable and explain any special situations where it loses observability.

Solution. Taking the torque on the handle bars as an input and the lateral deviation as the output, we can write the dynamics in state space form as (Exercise 3.3)

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & mgh/J \\ 1 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u,$$

$$y = \begin{bmatrix} \dfrac{Dv_0}{bJ} & \dfrac{mv_0^2 h}{bJ} \end{bmatrix} x.$$

The observability of this system determines whether it is possible to determine the entire system state (tilt angle and tilt rate) from observations of the input (steering angle) and output (tilt angle).

The observability matrix is

$$W_0 = \begin{bmatrix} \dfrac{Dv_0}{bJ} & \dfrac{mv_0^2 h}{bJ} \\ \dfrac{mv_0^2 h}{bJ} & \dfrac{mgh}{J}\cdot\dfrac{Dv_0}{bJ} \end{bmatrix}$$

and its determinant is

$$\det W_o = \left(\frac{Dv_0}{bJ}\right)^2 \frac{mgh}{J} - \left(\frac{mv_0^2 h}{bJ}\right)^2.$$

Under most choices of parameters, the determinant will be nonzero and hence the system is observable. However, if the parameters of the system are chosen such that

$$\frac{mv_0 h}{D} = \sqrt{\frac{mgh}{J}}$$

then we see that W_o becomes singular and the system is not observable.

(i) Exercise 8.10

Exercise 7.10 (Observer design for motor drive) Consider the normalized model of the motor drive in Exercise 2.10 where the open loop system has the eigenvalues $0, 0, -0.05 \pm i$. A state feedback that gave a closed loop system with eigenvalues in -2, -1 and $-1 \pm i$ was designed in Exercise 6.11. Design an observer for the system that has eigenvalues -4, -2 and $-2 \pm 2i$. Combine the observer with the state feedback from Exercise 6.11 to obtain an output feedback and simulate the complete system.

Solution. The model given in Exercise 6.11 is

$$\frac{1}{\omega_0}\frac{dx}{dt} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\alpha_1 & \alpha_1 & -\beta_1 & \beta_1 \\ \alpha_2 & -\alpha_2 & \beta_2 & -\beta_2 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ \gamma_1 \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \gamma_2 \end{bmatrix} d$$

where the states are $x_1 = \varphi_1$, $x_2 = \varphi_2$, $x_3 = \dot{\varphi}_1/\omega_0$ and $x_4 = \dot{\varphi}_2/\omega_0$. The estimator gain is given by Theorem 7.3. It can be calculated using the following script.

```
% motordrive_obs.m
%   Design of observers for the motor drive
%   kja 070822
%
%   Part 1 regular observer
%   Get process model
motordrive_data;
%   Compute observer gain
P=[-4 -2 2*(-1+i) 2*(-1-i)];
L=place(A',C2',P)';Acl=A-L*C2;
disp('Observer gain L=')
disp(L')
%Check the results
disp('Specified closed loop eigenvalues')
disp(P);
disp('Closed loop eigenvalues obtained')
disp(eig(Acl)')
```

The observer gains are $L = \begin{bmatrix} 646.9 & 9.9 & 297.9 & 38.0 \end{bmatrix}^T$. The observer is

$$\frac{d\hat{x}}{dt} = A\hat{x} + Bu + L(y - C\hat{x})$$

and the control signal is given by $u = -K\hat{x} + k_r r$. Inserting this expression in the estimator we find

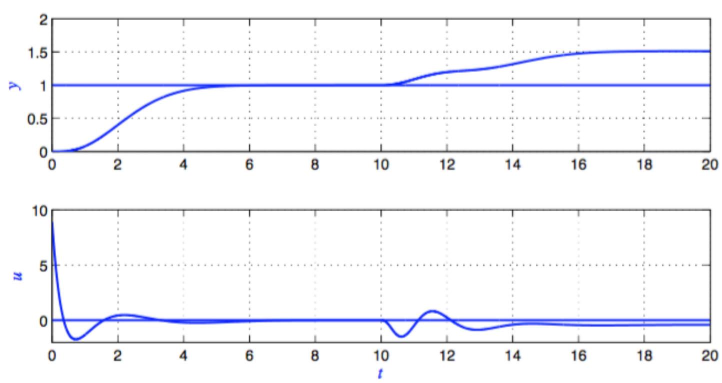
$$\frac{d\hat{x}}{dt} = A\hat{x} + Bu + L(y - C\hat{x}) = (A - KB - LC)\hat{x} + Ky$$

The closed loop simulation is obtained using the script

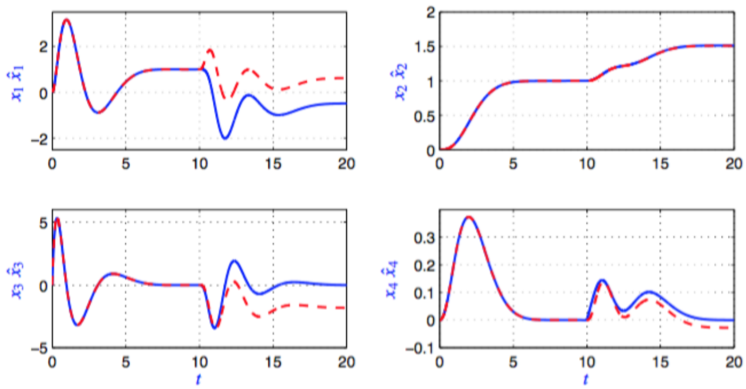
```
% motordrive_ofbsim.m
% Simlation of motor drive with state feedback
%   kja 070822
%   Get procees parameters
aminit;
%   Design state feedback and observer
motordrive_sfb;
motordrive_obs;
%   Create process model block
Ap=A;Bp=[B1 B2];Cp=C2;Dp=zeros(1,2);
%   Create controller block
Ac=A-L*C2-B1*K;Bc=[Kr*B1 L];Cc=-K;Dc=[Kr 0];
%   Execute simulation
r=1;d=2;ts=20;[t,x,y]=sim('motordrive_blk',[0 ts]);
```

```
%   Plot output and control variable
figure(1);subplot(311);
pl=plot(t,y(:,1),'b-',t,y(:,2),'b');
set(pl,'Linewidth',1.5);
axis([0 20 0 2]);grid on;
ylabel('y');
subplot(312);
pl=plot(t,y(:,3),'b-',t,zeros(size(t)),'b-');
set(pl,'Linewidth',1.5);
ylabel('u');xlabel('t');grid on;
axis([0 20 -2 10]);
amprint('motordrive-ofbsim.eps');
%   Plot states and estimates
figure(2);subplot(321);
pl=plot(t,x(:,1),'b-',t,x(:,5),'b--');
set(pl,'Linewidth',1.5);grid on;
axis([0 ts -2.5 3.5]);
ylabel('x1');
subplot(322);
pl=plot(t,x(:,2),'b-',t,x(:,6),'b--');
set(pl,'Linewidth',1.5);grid on;
ylabel('x2');
subplot(323);
pl=plot(t,x(:,3),'b-',t,x(:,7),'b--');
set(pl,'Linewidth',1.5);grid on;
axis([0 ts -5 6]);
xlabel('t');ylabel('x3');
subplot(324);
pl=plot(t,x(:,4),'b-',t,x(:,8),'b--');
set(pl,'Linewidth',1.5);grid on;
axis([0 ts -0.1 0.4]);
xlabel('t');ylabel('x4');
amprint('motordrive-ofbsim-est.eps');
```

The figure below shows the response of the system to a unit step in the reference at time $t = 10$ there is a step change in the torque on the second wheel.



The load disturbance gives a steady-state error because the controller has no integral action. Further insight into this is obtained from the figure below which shows the states using solid lines and their estimates using dashed lines.



Notice that the estimates are exact in the first part of the simulation because there were no model errors and the estimator and the system were initialized with all states equal to zero. There are however significant deviations when the load disturbance is introduced. The estimate of state x_2 is still very good because this state is measured. It is natural that there will be errors because the disturbance is not accounted for in the estimator. The difficulty can be eliminated by introducing integral action.