# 7 Module 7: State Feedback

## 7.1 Overview

State space design methods are introduced in this module. Chapter 6 of the text covers all of the state space design approaches. We will start with full state feedback. This assumes that all state variables can be measured. A satellite tracking dish is used as an example. A more detailed study of this problem is given in Appendix A.2 of the text. Figure 4 illustrates the plant.

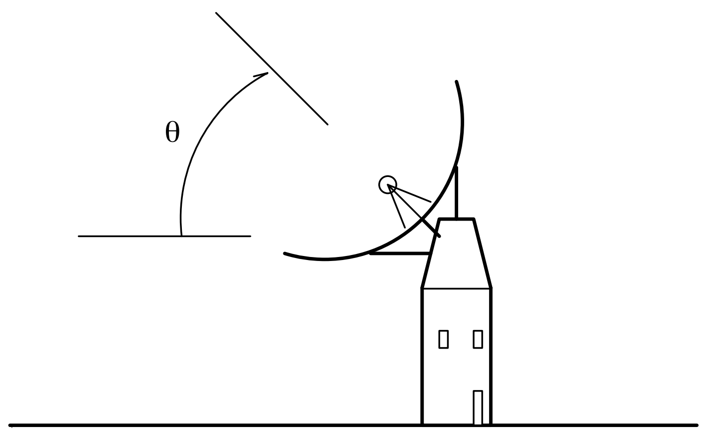



Figure 4: Satellite antenna dish

## 7.2 Notes

This module covers a relatively involved state-feedback problem. The accompanying block diagrams are discussed in the notes section. It is often easier to have the equivalent picture in your mind when you are looking at the mathematics.

The open loop system is formed in the continuous domain. This is illustrated schematically in Figure 5. A wind disturbance, at input $Td(t)$, is modelled as low passed random noise signal. For the open loop simulation $Tc(t) = 0$. This simulation forms the base line for comparison of the closed loop schemes.

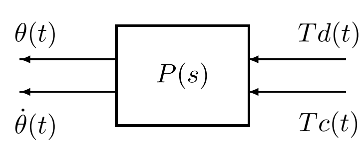



Figure 5: Open loop plant: Satellite antenna dish

Now a zero order hold equivalent is obtained and used for state feedback design.

The closed loop, discrete time, state feedback system is illustrated if Figure 6. The discrete time system, $Pd(z)$, is a zero order hold equivalent to $P(s)$. For the design purposes, we are only interested in the control input, $Tc$.

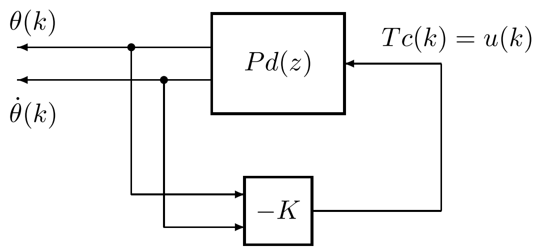



Figure 6: Design configuration for discrete time state feedback

The objective of the state feedback design is to drive the state to zero. Often, the non zero state results from an initial condition. In this case the wind disturbance will drive the states away from zero.

The simulation is now much more complex than any we have previously performed. The controller, $-K$, is discrete time. I know it doesn't have any dynamics, however, we designed it for a discrete time plant, to put the poles at good locations in the unit disk. The wind disturbance signal, $Td(t)$, is continuous and drives the states of the continuous plant $P(s)$. To simulate this sampled-data (i.e. mixed continuous and discrete) system we use SIMULINK.

The full system is illustrated in Figure 7.

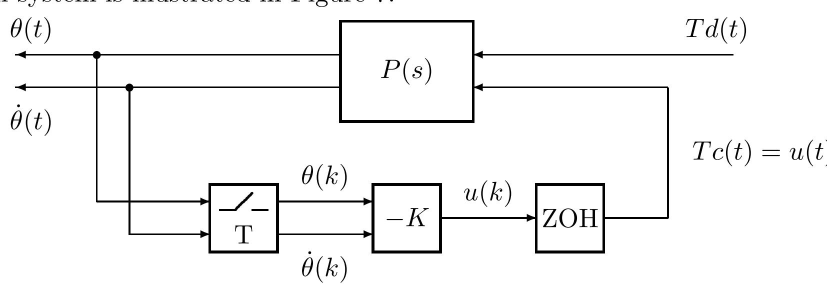


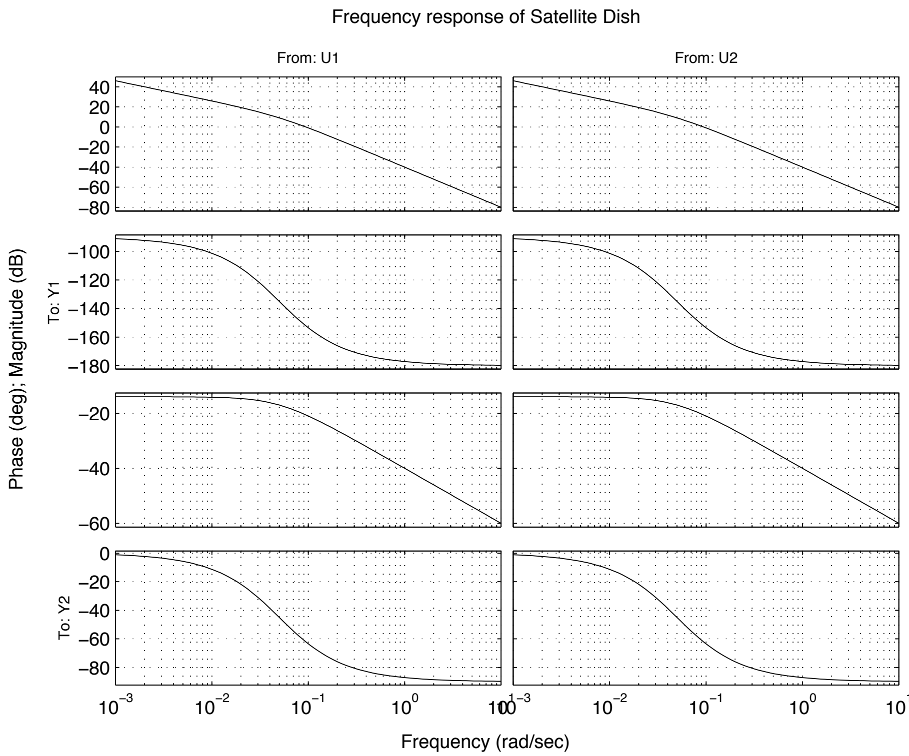

Figure 7: Sampled data feedback system used for simulation

This state space representation of this interconnected system is constructed in `m7sd.mdl`.

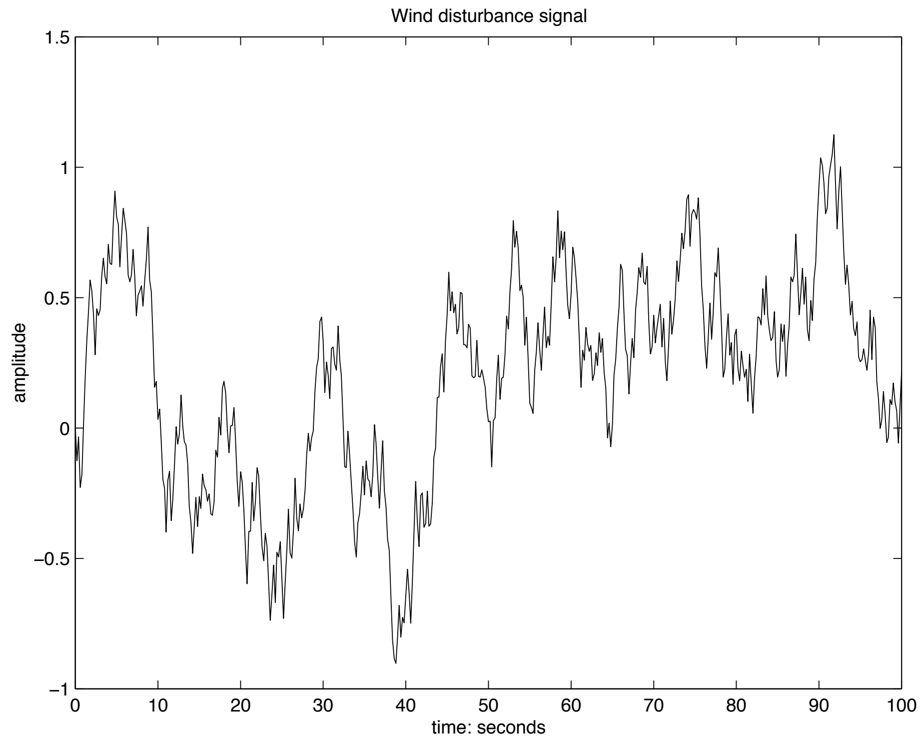
## 7.3 Module 7: Listing and Graphics

```
clf;axis([1,2,3,4]);axis;
echo on
%--------------------------------------------------------
%
%          Dept. of Electrical & Computer Eng.
%       University of California, Santa Barbara.
%
%               ECE 147 B:  Digital Control Systems
%
%               module 7:    State feedback
%
%       -------------------------------------------------

% State feedback design is studied in this module.
% This material is covered in Chapter 8 of the text.

% Adjust the Command and Graphics windows to your preferred
% sizes.

pause   % press any key to continue.

% A simple second order system is studied as an example.  The
% system is initially specified as a continuous system and
% then transformed, via zero-order hold equivalence, to
% a discrete time system.

% Consider the problem of pointing a satellite receiving
% dish.  The angle of the dish must be maintained constant
% in the presence of disturbances.  Define theta as this angle.
% The equation of motion is,
%
%    J*theta" + B*theta' = Tc + Td,
%
% where theta' is the derivative of the angle, theta" is the second
% derivative, J is the moment of inertia, B is the damping,
% Tc is the motor drive torque and Td is the torque exerted
% by the wind disturbance.
%
% In Laplace form, this equation is,
%
%       (J*s^2 + B*s)theta = Td + Tc.
%

pause   % press any key to continue
```

```
% Tc is the control input and Td is the disturbance to be
% rejected.  Assume that the system is in equilibrium, pointed
% in the correct direction.  We can measure theta and theta' via
% sensors mounted on the dish.
%
% Define the state as x = [theta ; theta'].  The state equations are,
%
%       theta'  =  theta'  = [0 1] x,
% and
%       theta"  =  -(B/J)*theta' + (1/J)*Td + (1/J)*Tc.
%
% In the more usual form, these are,
%
%              | 0     1 |        |  0  |         |  0  |
%       x'  =  |         | x   +  |     | Td  +   |     | Tc
%              | 0  -B/J |        | 1/J |         | 1/J |
%


pause   % press any key to continue.

% OK, this gives us the A and B matrices of a state-space
% description.  Choose some values and form these matrices.

J = 100;
B = 5;

A = [0, 1; 0, -B/J];
B = [0, 0; 1/J, 1/J];

% Note that we defined B with two inputs: Td and Tc.
% To form a complete state-space system, we need to
% create C and D matrices.  Because we measure the
% state directly C is the identity and D is zero.

C = eye(2,2);
D = zeros(2,2);
sat_dish = ss(A,B,C,D);
set(sat_dish,'Notes','Satellite dish');

pause   % press any key to continue.

% We can look at a frequency response of this system.

omega = logspace(-3,1,100);
bode(sat_dish,omega);
title('Frequency response of Satellite Dish')
```

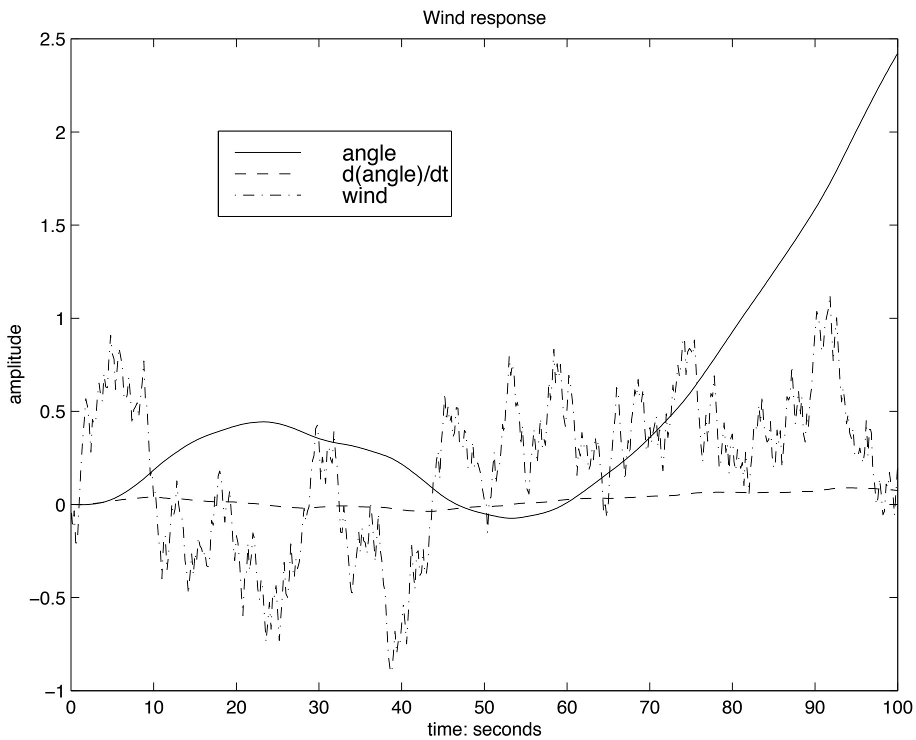



```
% The plot with the 1/s characteristic at low frequencies
% is that from Tc (and Td - they are identical) to theta.

pause   % press any key to continue.

% Let's also do an open-loop time response to see the
% effect of wind on the satellite dish.  Wind will be
% modeled as a random noise sequence going through a
% low pass filter.

t = [0:0.2:100]';
w = rand(length(t),1) - 0.5;
lpfilter = tf(20,[10,1]);
wind = lsim(lpfilter,w,t);
clear w lpfilter

% Let's look at the wind disturbance signal.

plot(t,wind,'-')
title('Wind disturbance signal')
xlabel('time: seconds')
ylabel('amplitude')
```



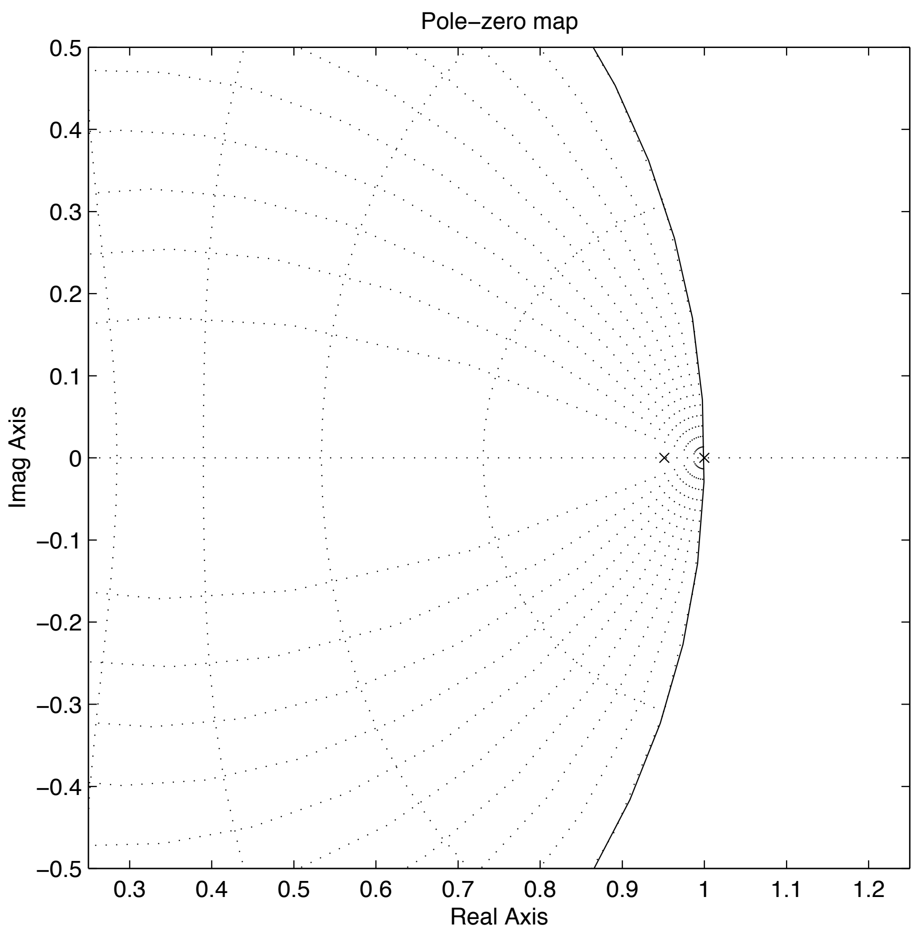

```
% It looks gusty to me.  It goes in both directions which may
% not be too realistic.  If it was predominantly in one direction,
% the integrator in the plant would integrate up.  I.e. it would
% push the dish around.

pause   % press any key to continue.

% Now lets look at the effect of this on the open-loop system.
% Because there is no control, that input must be set to
% zero.

dist = [wind,0*wind];
wind_resp = lsim(sat_dish,dist,t);

% The first column of wind_resp is the angle and the second
% column is the derivative of the angle.  We can select each
% of these out for plotting.

theta = wind_resp(:,1);
thetadot = wind_resp(:,2);

plot(t,theta,'-',t,thetadot,'--',t,wind,'-.')
title('Wind response')
```

```
legend('angle','d(angle)/dt','wind')
xlabel('time: seconds')
ylabel('amplitude')
```




```
% The angle drifts around quite badly and the objective
% will to be improve this response.

pause   % press any key to continue.

% The actual system will be preceeded by a zero-order hold
% and both outputs will be sampled with period T.  We can
% calculate the discrete equivalent of this with sys2dsys.

T = 1;                  % 1 second sample period.

sat_dish_Z = c2d(sat_dish,T,'zoh');
set(sat_dish_Z,'Notes','ZOH equivalent of the satellite dish');

% Examine the pole locations:

clf
zgrid
axis('square')
```

```
axis([0.25,1.25,-0.5,0.5])
pzmap(sat_dish_Z)
```

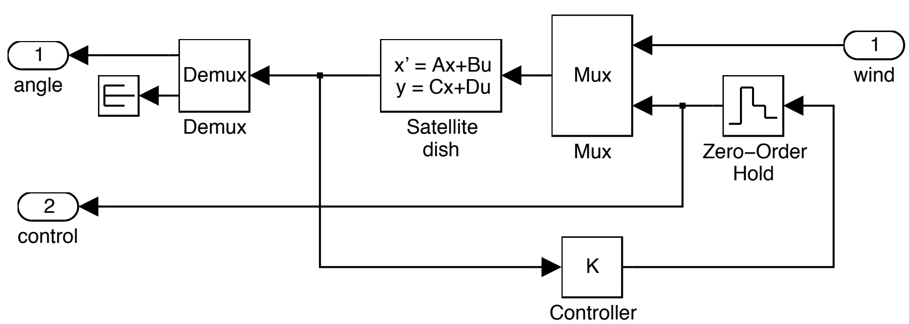



```
pole(sat_dish_Z)

ans =

    1.0000
    0.9512


% We can see the integrator in the discrete system.

pause   % press any key to continue.

% To get the discrete state-space matrices, the ssdata
% command is used.
```

```
[Ad,Bd,Cd,Dd] = ssdata(sat_dish_Z)

Ad =

    1.0000    0.9754
         0    0.9512


Bd =

    0.0049    0.0049
    0.0098    0.0098


Cd =

     1     0
     0     1


Dd =

     0     0
     0     0


% The C and D matrices are unchanged as expected.

pause   % press any key to continue.

% Now let's check whether the system is controllable.
% The controllability matrix is formed with a bit of
% manipulation.  The first thing to observe is that we
% wish to control the system using the second input (i.e. Tc).
% So the B matrix in question is just the first column of Bd.

BdTc = Bd(:,2);

Cmat = [BdTc, Ad*BdTc];

size(Cmat)

ans =

     2     2
```

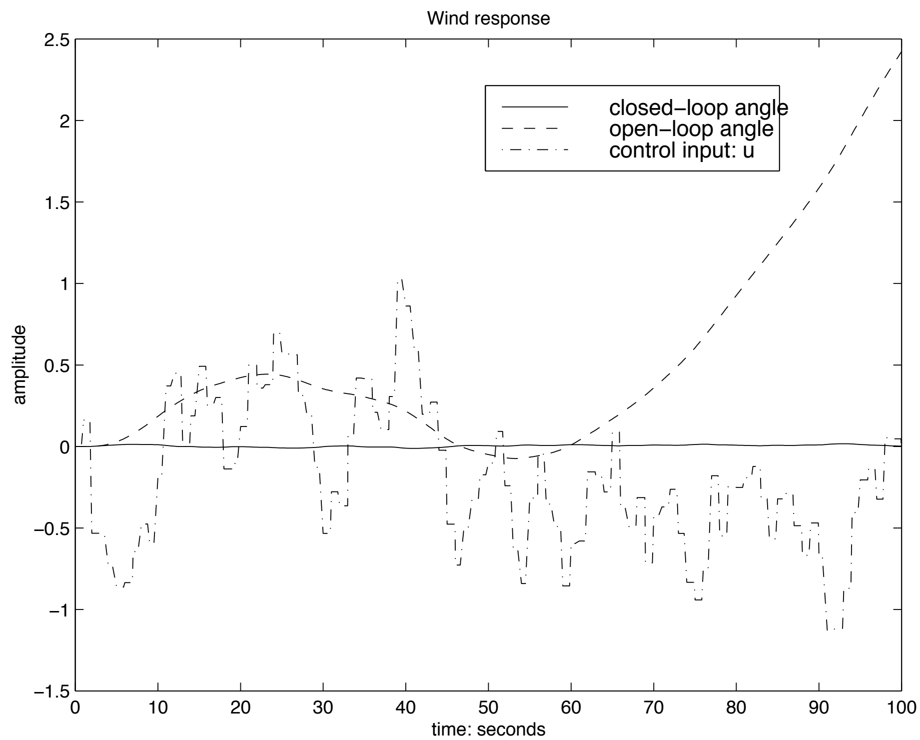
```
rank(Cmat)

ans =

     2


% The controllability matrix is full rank.  Good;  we can place
% both poles where we would like.

pause   % press any key to continue.

% Now to choose some pole locations.  Lets try z = 0 and
% z = 0.5.  That should give a fast response.

% Ackermann's formula can be used to calculate the state
% feedback gain.  Firstly we need to figure out what is
% the polynomial that has roots at these locations.
%
% I.e.  z(z - 0.5) = 0
%
%       z^2 - 0.5z = 0.
%
% So, using the notation in the book,
%
%        alpha_c(Ad) = Ad^2 - 0.5*Ad

alphac = Ad^2 - 0.5*Ad;

% and K can now be calculated:

K = [0, 1]*inv(Cmat)*alphac

K =

   51.2604  122.9375


% There is a control toolbox command called "place" which also
% does this calculation.  Run help on that if you can't understand
% what I have done here.

pause   % press any key to continue

% The numbers look quite large.  We will do a simulation
% to see what happens.  To do this we must form the closed
% loop system.
```

```
% First, we will check where the poles of the system are.
% One way is to look at the eigenvalues of the closed loop
% system.   These are,

eig(Ad - BdTc*K)

ans =

    0.5000
    0.0000


% Good.  Exactly where we want them to be.

pause   % press any key to continue.


% This is quite a tricky simulation to do.  The input signal
% is in continuous time and the controller is in discrete time.
% OK, now do the simulation.  First we have to multiply K by
% -1 as negative feedback is assumed in the problem.

negK = -1*K;
[A,B,C,D] = ssdata(sat_dish);    % get the matrix data.
m7sd;
```



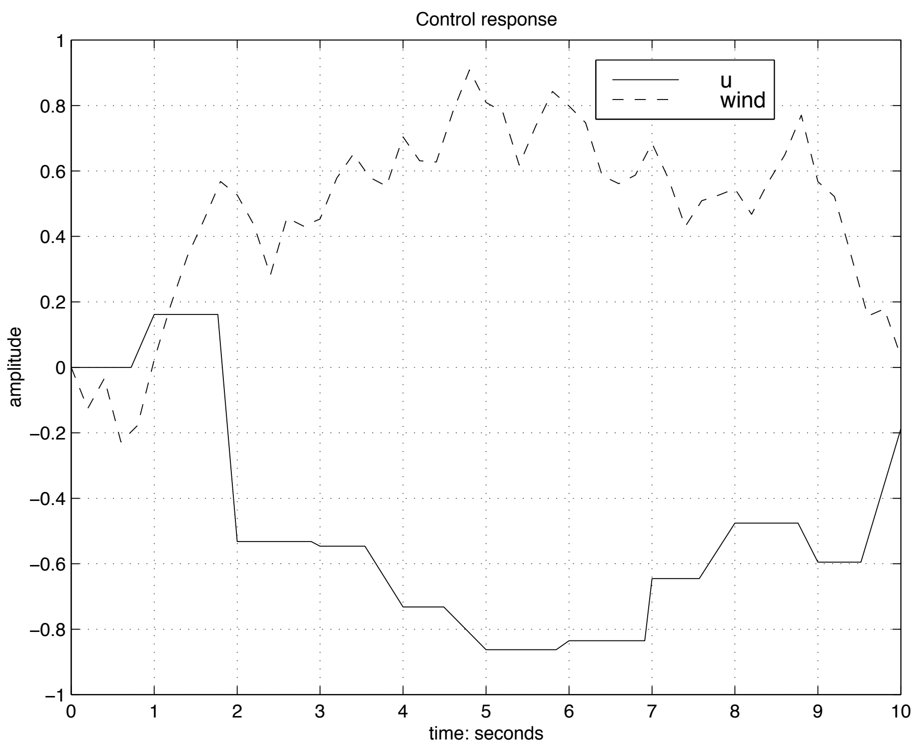

```
% We now do a sampled data simulation using simulink.  Note
% that we are interested in both the angle and the control
% actuator signal.

options = simset('InitialState',[0,0]);
options = simset(options,'OutputPoints','specified');
[tout,x,y] = sim('m7sd',[0,max(t)],options,[t,wind]);
thetaclp = y(:,1);
u = y(:,2);
```

```
clf
plot(tout,thetaclp,'-',t,theta,'--',tout,u,'-.')
title('Wind response')
legend('closed-loop angle','open-loop angle','control input: u')
xlabel('time: seconds')
ylabel('amplitude')
```




```
% Excellent closed-loop response!

% The input signal, u, actually looks a lot like the negative
% of the wind.  The controller doesn't get to measure the
% wind directly - only its affect on the state.  However, the
% best control will more or less cancel out the effect of the
% wind.

pause   % press any key to continue.

% We can look at the control and wind signals in greater detail.
% Select only the first 10 seconds of each signal

t10idx = find(t <= 10);         % indices for t <= 10
to10idx = find(tout <= 10);     % indices for tout <= 10
```

```
clf
plot(tout(to10idx),u(to10idx),'+',t(t10idx),wind(t10idx),'-')
title('Control response')
legend('u','wind')
xlabel('time: seconds')
ylabel('amplitude')
grid
```




```
% Note that the control signal, u, seems to be a slightly
% delayed, and inverted version of wind.  The delay is about
% how long it takes for the effect of the wind to be seen at
% the output.  The sample period can also be seen in u.

%-----------------------------------------------------------

echo off
```

## 7.4 Problems

1. Study the tradeoff between requested pole positions and actuator effort. Select and simulate two other pole locations, one more aggressive and one less aggressive than shown here. Use the same wind input for each so that a direct comparison is possible.

2. Now we will use a state feedback approach to design a controller to balance an inverted pendulum. Naturally, this may help you in your lab work. Think about how you might reuse the m-file written here for your lab later on.

   I won't give a complete description of the experiment here. For details refer to experiment number 3 in the ECE 147b laboratory manual.

   The state is defined as,

   $$x = [\,\theta\ \dot{\theta}\ p\ \dot{p}\,]',$$

   where $\theta$ is the pendulum angle and $p$ is the cart position. This gives a continuous state-space representation of

   $$\dot{x}(t) \quad = \quad Ax(t) + Bu(t) \tag{1}$$

   where

   $$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \alpha_2 & -\alpha_1 & 0 & \beta_1/\tau \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1/\tau \end{bmatrix},$$

   and

   $$B = \begin{bmatrix} 0 \\ -\beta_1\beta/\tau \\ 0 \\ \beta/\tau \end{bmatrix}.$$

   In this problem, we will assume that all state variables are available for measurement. Use the following values in the simulation.

   $$\tau = 0.02, \quad \beta = 0.05, \quad \beta_1 = 3.3, \quad \alpha_1 = 0.36, \quad \alpha_2 = 32.$$

   The sample time is $T = 0.03$ seconds. Do a design to place the closed loop poles at: $[0.75, 0.84, 0.99 + 0.01i, 0.99 - 0.01i]$.

   Simulate this design with an initial condition of $\theta(0) = 1.0$ degrees and $p(0) = 0.01$ meters. The actuator is limited to $\pm 1.0$ Volts (this doesn't exactly correspond to the lab but we have left out some of the scaling factors here). Evaluate your design in light of this requirement.

   Find another initial condition, of about the same size, which is more difficult to control. Attempt to generate a design which performs better than your first one on this more difficult initial condition.