CONTROL SYSTEMS THEORY
University of Florida
Mechanical and Aerospace Engineering

HW 7

You are encouraged to use MATLAB© to verify your answers whenever you can. However, unless specified, do not use MATLAB© to solve a problem.

Problem 1. Consider an LTI system described by the following matrices:

$$A = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & -3 \\ 0 & 1 & 0 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}, C = \begin{bmatrix} 0 & 1 & 1 \\ 0 & -1 & -1 \end{bmatrix} \tag{1}$$

1. Determine controllability and observability of the system. (Meaning, determine if they are controllable and observable. If not, determine the dimension of the controllable subspace and unobservable subspace.)
2. Comment on the relation, if any, between the number of inputs and controllability, and the number of outputs and observability.

Hint: If you use the rank test, unless the rank is obvious from inspection, reduce the controllability/observability matrices to row echelon form to determine their rank. Sometimes it is useful to use the fact that rank of a matrix is the same as the rank of its transpose, to first transpose the matrix and then do row reduction. Verify your answer in MATLAB© using `ctrb`*,* `obsv` *and* `rank` *commands.*

Problem 2. Consider the following system

$$\dot{x} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} x, \qquad y = \begin{bmatrix} c_1 & c_2 & c_3 \end{bmatrix} x,$$

where c_1, c_2, c_3 are unknown scalars.

1. Provide an example of values of c_1, c_2, c_3 for which the system is not observable.
2. Provide an example of values of c_1, c_2, c_3 for which the system is observable.
3. Provide a necessary and sufficient condition on the c_i's so that the system is observable.

Problem 3. This is a MATLAB©-only problem. A Simulink model `observer.slx` is provided with this homework, with an accompanying MATLAB©script `RunObserver_INCOMPLETE.m`. The plant being simulated is stable, with 2 inputs and 1 output. Your task is to complete the MATLAB© script so that all the parameters needed by the observer in the Simulink model are defined, and the model executes correctly when called, either from the script directly or by clicking the 'run' button on the Simulink model.

1. Show that the observer dynamics $\dot{\hat{x}} = A\hat{x} + Bu + L(y - (C\hat{x} + Du))$ can be rewritten as

$$\dot{\hat{x}} = (A - LC)\hat{x} + \begin{bmatrix} B - LD & L \end{bmatrix} \begin{bmatrix} u \\ y \end{bmatrix}$$

or as

$$\dot{\hat{x}} = (A - LC)\hat{x} + \begin{bmatrix} L & B - LD \end{bmatrix} \begin{bmatrix} y \\ u \end{bmatrix}$$

Beware: in analyzing an observer we frequenctly replace y with $Cx + Du$, but we cannot do it in implementation, since the state x is unknown. When implementing an observer, whether in a simulink simulation or a real system, you have to think of the observer as a dynamic system with state $\hat{x}$, output $\hat{x}$ (same as the state), and input $\begin{bmatrix} u \\ y \end{bmatrix}$ or $\begin{bmatrix} y \\ u \end{bmatrix}$. Either way of defining the input vector is fine, but you must be consistent. Check the simulink block to see which one I am using; your matlab script will have to be consistent with how I have defined the observer input in the Simulink model.

2. No need to resubmit the whole script, simply complete the following section of the script (the provided .m file) that needs completion:

```
obsv_poles = ; %choose desired eigenvalues of A-LC
....
L =....
```

Also submit the plots of the states and state estimates between $t = 0$ to $t = 15$. (Plot x_i and $\hat{x}_i$ in the same figure so the accuracy of the estimate can be visually checked easily.)

3. Describe your rationale for your choice of the eigenvalues of $A - LC$. What effect does varying these eigenvalues have on the state estimation accuracy?

You have to use duality to design the observer gain L by using the `place` command, since that command is meant for designing K in state feedback control. A frequent mistake here is to forget to use tranposes. Review duality carefully.

Problem 4. (In this problem you will design a control system for a SISO plant $P(s)$ to meet transient response performance requirements by using state space techniques. You are allowed - and encouraged - to use MATLAB© for any calculation needed in this problem.) Consider the following plant $P(s)$ whose input is $u + v$, where u is the control command and v is a disturbance, and output y:

$$P(s) = \frac{100(s+5)}{(s-0.5)(s+1)(s+10)(s^2+4s+13)} \tag{2}$$

A minimal realization of it the plant is provided below:

$$\dot{x} = \begin{bmatrix} -14.5 & -59.5 & -149.5 & -38.5 & 65 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} (u + v) \tag{3}$$

$$y = \begin{bmatrix} 0 & 0 & 0 & 100 & 500 \end{bmatrix} x \tag{4}$$

Your job is to design and output feedback controller for this plant by using a state observer and combining it with a state feedback controller (that is, $u(t) = -K\hat{x}(t)$, and $\hat{x}$ is an estimate of the state x), and simulate the closed loop. Here are the tasks:

1. Draw a sketch of the closed loop control system.

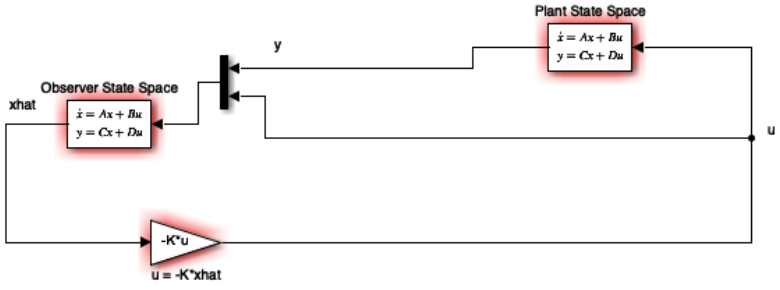
2. Show that with the output feedback controller, the closed loop system's state space representation is

$$\begin{aligned} \begin{bmatrix} \dot{x} \\ \dot{e} \end{bmatrix} &= \begin{bmatrix} A - BK & BK \\ 0 & A - LC \end{bmatrix} \begin{bmatrix} x \\ e \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} v \\ y &= \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x \\ e \end{bmatrix} \end{aligned} \tag{5}$$

   where $e = x - \hat{x}$, and A, B, C are the system matrices of the plant $P(s)$ shown in eq. (3)-(4).

3. Design the control system so that the closed loop transfer function H_{yv} is BIBO stable and has a 2% settling time less than 1. *hint: You have to choose controller poles and observer poles in this step, and then get K, L by using the place command. The choice of these eigenvalues are dictated by the settling time requirement (recall the material on dominant poles and relation of poles of second order underdamped systems and transient response). After that, define the closed loop system by using* `ss` *and equation* (5)*, and determine rise time by using* `stepinfo`*. Verify the results of* `stepinfo` *by plotting the step response and checking visually.*

4. Construct a SIMULINK model of the closed loop system. Write a MATLAB© script to initialize and define all parameters needed by the Simulink model, and also to run the model using the `sim` command. Set $v(t)$ to 0 (for all t). Choose a sufficiently long time to see transients die out, use the following initial conditions: $x(0) = [1, 1, 1, 1, 1]^T$, $\hat{x}(0) = 0$. Submit: (i) plots of output, (ii) a picture of your simulink block, (iii) a brief description of your design process (not more than 5 sentences). *Your simulink model should have the plant, observer, and the state feedback controller, all interconnected. The easiest path to constructing it is to copy the observer model I am providing in the previous problem, remove the external input, add the state feedback controller, and make other necessary changes. My simulink block looks like the picture shown below.*



Problem 5. This is a MATLAB© only problem. Your task in this problem is to write a MATLAB© script to compute the solution of an ODE by "numerical integration." The ODE is

$$\ddot{y} + 2\dot{y} + 5y = f(t), y(0) = -4, \dot{y}(0) = 5 \tag{6}$$

where the input is $f(t) = 2\sin(2t) + 5$.

1. First, use matlab's built in function `lsim` to compute the solution by doing the following. Express the ODE is state space form, and then use `lsim` to determine the solution between $t = 0$ and $t = 5$, using a time resolution of 0.25 seconds in specifying the input vector (Beware: do not forget to provide the initial condition to `lsim`.)

2. Now write a MATLAB©script to compute the solution approximately by using either (i) a first-order Euler forward-difference method[1], or (ii) discretizing the ODE using the continuous-to-discrete conversion covered in class. Either method is fine; they will produce similar results. Plot the solution in the same figure as the solution you get in step 1. Repeat this for three different sampling periods, $T_s = 0.5$, $T_s = 0.25$ and $T_s = 0.05$. Do all of these "solutions" look right? (Ans: The solution plots you should get are shown in Figure 1.)

3. An inexperienced engineer comes to you seeking your guidance in performing simulations of dynamic systems using an Euler approximation. What guidelines would you provide about the choice of sampling period for first order Euler forward-difference approximation?

Problem 6. Consider an LTI system given by

$$A = \begin{bmatrix} 0 & 1 \\ 3 & 2 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \qquad C = \begin{bmatrix} -3 & 1 \end{bmatrix}, \qquad D = 0$$

1. Compute the transfer function from i^{th} input to the output of the system.

2. Are the eigenvalues of A the same as the poles of the transfer function?

3. Is the system BIBO stable? Is the system stable in the sense of Lyapunov? Is the system asymptotically stable in the sense of Lyapunov?

Hint/answer key: this is an example of a pole-zero cancellation; the poles will be a strict subset of the eigenvalues.

Problem 7. (This is a MATLAB© problem) An incomplete C code `control.c` written for a particular hardware platform[2] to implement the following controller:

$$C(s) = 10[1 + \frac{25}{s} - \frac{19}{s+1}]$$

is given below. Use MATLAB© to determine the numerical values that should be in the incomplete statements in the code (i.e., use MATLAB©'s `c2d` function to compute the discrete-time A and B matrices of the controller).

[1] You can read about it in the texbook Barooah

[2] The platform happens to be dSpace™, but that's immaterial

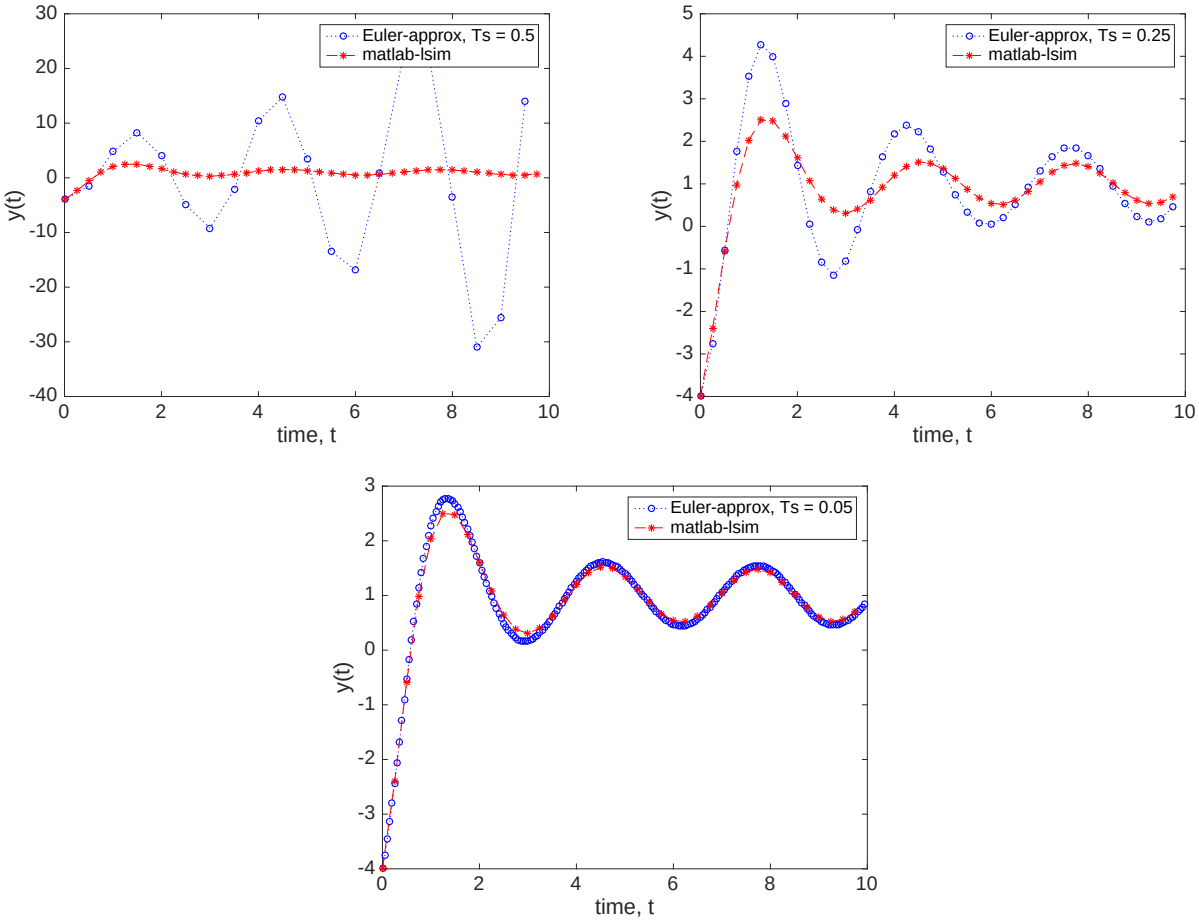


Figure 1: Comparison of numerical ODE solution from MATLAB©'s `lsim` and first order Euler approximation with various sampling periods.

```
/********************* control.c ***********************************/
#include "brtenv.h"                    /* basic real-time environment */

#define TS          1.25e-3           /* Sample period, in sec */
#define TMR0        0                  /* Use timer0 for exec-time trace */

float r,y,e,u;                           /* input/output variables */
float a11=
float a12=
float a21=
float a22=
float b11=
float b21=
float c11=
float c12=
```

```
float d =

float x1 = 0;
float x2 = 0;
float u_max = 10;
float u_min = -5;

/* timer0 interrupt service routine */
isr_t0()
{
  ds1102_ad_start();            /* start DS1102 ADCs */
  r = ds1102_ad(1);             /* read input r(t) from ADC channel 1 */
  y = ds1102_ad(2);             /* read input y(t) from ADC channel 2 */
  e = r-y; /* compute tracking error */

  x1next = a11*x1 + a12*x2 + b11*e;
  x2next = a21*x1 + a22*x2 + b21*e;
  u = c11*x1 + c12*x2 + d*e;

 /* clip the control command if it exceeds actuator ability */
 if (u>u_max)
   u_final  = u_max;
 else
 {
 if (u<u_min)
  u_final = u_min;
 else
  u_final = u;
 }

  ds1102_da(1,u_final);              /* write output to DAC channel 1*/

  /* dont forget the next step*/
  x1=x1next;
  x2=x2next;
}


main()
{
  init();                        /* initialize hardware */
  start_isr_t0(TS);              /* initialize sampling clock timer */
}
/********************* end of control.c *********************************/
```

Problems 8,9,10 are on one theme.

Problem 8 (unstable pole-zero cancellation)**.** A risky - but at first glance attractive - way of stabilizing an unstable plant is to cancel the unstable poles of the plant with right-half-plane zeros of the controller. This is a bad idea, for at least two reasons. The first reason is modeling uncertainty: since we do not know the precise locations of the plant poles, we will never be able to cancel unstable poles exactly. The second reason is that even if we knew the plant pole locations exactly, pole-zero cancellations only hide the instability but do not remove it. This problem illustrates the second reason. Consider a standard negative feedback loop with three external inputs: reference $r(t)$, disturbance $w(t)$ and noise $n(t)$, with plant and controller given by

$$P(s) = \frac{0.5}{(s+0.5)(s-1)}, \quad C(s) = \frac{s-1}{s+2}$$

As we usually do in such circumstances, the output of the plant is denoted by $z(t)$ and the noisy output that is used for feedback by $y(t)$.

1. Determine the BIBO stability property of each of the following closed loop transfer functions:

   (a) $H_{ZR}(s)$
   (b) $H_{YR}(s)$
   (c) $H_{ZW}(s)$
   (d) $H_{UR}(s)$

   Answer key: the first two are BIBO stable, third one is not.

Problem 9. Consider a standard feedback loop with a SISO plant and a SISO controller. Suppose a minimal realization of the plant is A_p, B_p, C_p, D_p with state x_p, input u and output y. And, a minimal realization of the controller is A_c, B_c, C_c, D_c, with state x_c, input e and output u. Recall that $e = r - y$. The only external input to the closed loop system is r, and the output of the closed loop system is the plant output y. Determine a realization of the closed loop system with state $z = [x_p^T, x_c^T]^T$ by combining the state equations of the plant and the controller and eliminating the internal variables e and u. (Hint: If you have to assume a particular matrix to be invertible, go ahead and assume that.) That is, determine what the matrices $A_{cl}, B_{cl}, C_{cl}, D_{cl}$ should be in

$$\frac{d}{dt}\begin{bmatrix} x_p(t) \\ x_c(t) \end{bmatrix} = A_{cl}\begin{bmatrix} x_p(t) \\ x_c(t) \end{bmatrix} + B_{cl}r(t) \qquad y(t) = C_{cl}\begin{bmatrix} x_p(t) \\ x_c(t) \end{bmatrix} + D_{cl}r(t)$$

Note:

1. *A frequent mistake is to compute the closed loop transfer function and then compute its realization! Write down a realization of plant and controller separately and then combine them. Also, pay attention that the matrices $A_{cl}, B_{cl}, C_{cl}, D_{cl}$ can only be a function of the matrices of the plant and controller's state space description, they cannot have any time varying functions.*

2. *Do this one carefully since you are going to use the answer in the next problem, so if you get this wrong you will get the next one wrong too.*

Problem 10. Now apply the results of the previous problems to determine the state space realization of the closed loop system with plant and controller.

$$P(s) = \frac{0.5}{(s+0.5)(s-1)}, \quad C(s) = \frac{s-1}{s+2}$$

Assume there are no noise and disturbance; the only external input is r.

1. Is the closed loop system stable in the sense of Lyapunov?

2. Discuss any connection you see between the previous answer and the answer to Problem 8.

A frequent error is to compute $CP/1 + CP$ and then determine its minimal realization. That's WRONG! .

Problem 11. [Ungraded problem] Based on the problems in this homework discuss two (or more) advantges of state-space descriptions of linear dynamic systems over transfer function descriptions. For each advantage, mention which problems in the homework illustrate those advantages.