ECE-520: Linear Control Systems
Homework 4

Due: Friday January 6 at 5 PM

1) Consider the discrete-time state variable model

$$\underline{x}(k+1) = G(T)\underline{x}(k) + H(T)u(k)$$

where the explicit dependence of G and H on the sampling time T has been emphasized. Here

$$G(T) = e^{AT}$$
$$H(T) = \int_0^T e^{A\lambda} d\lambda B$$

a) Show that if A is invertible, we can write $H(T) = [e^{AT} - I]A^{-1}B$

b) Show that if A is invertible and T is small we can write the state model as

$$\underline{x}(k+1) = [I + AT]\underline{x}(k) + BTu(k)$$

c) Show that if we use the approximation

$$\dot{\underline{x}}(t) \approx \frac{\underline{x}((k+1)T) - \underline{x}(kT)}{T} = Ax(kT) + Bu(kT)$$

we get the same answer as in part **b,** but using this approximation we do not need to assume A is invertible.

d) Show that if we use two terms in the approximation for e^{AT} (and no assumptions about A being invertible), we can write the state equations as

$$\underline{x}(k+1) = \left[I + AT\right]\underline{x}(k) + [T + \frac{1}{2}AT^2]Bu(k)$$

2) For the state variable system

$$\dot{\underline{x}}(t) = \begin{bmatrix} 1 & -1 \\ 2 & 4 \end{bmatrix}\underline{x}(t) + \begin{bmatrix} 1 \\ 1 \end{bmatrix}u(t)$$

a) Show that

$$e^{AT} = \begin{bmatrix} 2e^{2T} - e^{3T} & e^{2T} - e^{3T} \\ 2e^{3T} - 2e^{2T} & 2e^{3T} - e^{2T} \end{bmatrix}$$

b) Derive the equivalent ZOH discrete-time system

$$\underline{x}(k+1) = G\underline{x}(k) + Hu(k)$$

for $T = 0.1$ (integrate each entry in the matrix $e^{A\lambda}$ separately.) Compare your answer with that given by Matlab's **c2d** command.

3) Consider the discrete-time model

$$\underline{x}(k+1) = G\underline{x}(k) + Hu(k)$$
$$y(k) = C\underline{x}(k-1)$$

where we will assume G is an invertible matrix.

a) Show that we can write

$$\underline{x}(k-1) = \begin{bmatrix} G^{-1} & -G^{-1}H \end{bmatrix} \begin{bmatrix} \underline{x}(k) \\ u(k-1) \end{bmatrix}$$

and if we want the new state as the final output we can write this system as

$$\begin{bmatrix} \underline{x}(k+1) \\ u(k) \end{bmatrix} = \begin{bmatrix} G & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \underline{x}(k) \\ u(k-1) \end{bmatrix} + \begin{bmatrix} H \\ I \end{bmatrix} u(k) = \tilde{G} \begin{bmatrix} \underline{x}(k) \\ u(k-1) \end{bmatrix} + \tilde{H}u(k)$$
$$y(k) = \begin{bmatrix} CG^{-1} & -CG^{-1}H \\ 0 & I \end{bmatrix} \begin{bmatrix} \underline{x}(k) \\ u(k-1) \end{bmatrix} = \tilde{C} \begin{bmatrix} \underline{x}(k) \\ u(k-1) \end{bmatrix}$$

b) Assume we want the states to be our outputs and that C (but not $\tilde{C}$)is an identity matrix. We will need to change the basis by letting $x(k) = P\hat{x}(k)$. In order for the output to equal the states, we will need $\tilde{C}P = I$, or in block matrix form

$$\begin{bmatrix} G^{-1} & -G^{-1}H \\ 0 & I \end{bmatrix} \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}$$

Determine the P_{ij} , and hence the matrix P we need to transform the system (clearly $P^{-1} = \tilde{C}$)

c) Change the basis (write a new system in terms of states $\hat{x}$) and show that we get our standard form for a system with a unit delay, shown below

$$\begin{bmatrix} \underline{x}(k+1) \\ u(k) \end{bmatrix} = \begin{bmatrix} G & H \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \underline{x}(k) \\ u(k-1) \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} u(k)$$
$$y(k) = [I] \begin{bmatrix} \underline{x}(k) \\ u(k-1) \end{bmatrix}$$

4) For the continuous time model

$$\underline{\dot{x}}(t) = \begin{bmatrix} 1 & -1 \\ 2 & 4 \end{bmatrix} \underline{x}(t) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(t-0.03)$$

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

derive the equivalent ZOH discrete-time system

$$\begin{bmatrix} \underline{x}([k+1]T) \\ u(kT) \end{bmatrix} = \begin{bmatrix} G & H_1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \underline{x}(kT) \\ u([k-1]T) \end{bmatrix} + \begin{bmatrix} H_0 \\ I \end{bmatrix} u(k)$$

$$y(kT) = C \begin{bmatrix} \underline{x}(kT) \\ u([k-1]T) \end{bmatrix}$$

for $T = 0.1$. Specifically, determine G, H_0, H_1, and C. You should do all of the calculations by hand (you've done most of the work in problem 2). You can check your answers in Matlab using the **c2d** command and the **expm** command.

5) Sometimes we would like to know what is happening to our continuous time system

$$\underline{x}(t) = A\underline{x}(t) + B\underline{u}(t)$$

$$\underline{y}(t) = C\underline{x}(t) + D\underline{u}(t)$$

between sample times, such as at time $t = kT + \Delta T$ where ΔT is less than the sampling interval T. From class, the solution to the continuous-time state equation system is given by

$$\underline{x}(t) = e^{A(t-t_o)}\underline{x}(t_o) + \int_{t_o}^{t} e^{A(t-\lambda)} B\underline{u}(\lambda) d\lambda$$

Assuming $t_o = kT$ and $t = kT + \Delta T$, derive an expression for the output at time $t = kT + \Delta T$, i.e., find $\underline{y}(kT + \Delta T)$. Do not assume D is zero.

6) Download the files **DT_openloop_ripple.mdl** and **DT_ripple_driver.m** from the class website. These two files together demonstrate a method of using Simulink and Matlab to determine what is happening between samples by running two simulations in parallel with a finer sampling in one of the simulations. Although at this point we could just as easily run two separate simulations, when we introduce feedback into the system we cannot just run two different systems, as you will see later. If you do not see the colors for the different times, select *Format* then *Sample time colors* after you've run the system once.

a) Simulate your *rectilinear* (model 210) 1 dof system for three different kinds of inputs (like a sine, step, ramp, pulse, etc.). Turn in your plots and a brief description of the inputs you used.

b) Simulate your *torsional* (model 205) 1 dof system for three different kinds of inputs (like a sine, step, ramp, pulse, etc.). Turn in your plots and a brief description of the inputs you used.

7) Consider the difference equation

$$x(k+2) - 4x(k+1) + 4x(k) = u(k+1)$$

where $u(k)$ is a unit step, $x(0) = 0$, and $x(1) = 1$. Sometimes we are interested in writing the solution in two parts, which we will do here.

a) Determine the *Zero Input Response* (ZIR), $x_{ZIR}(k)$. This is the part of $x(k)$ due to the initial conditions alone (Set all $u(k)$ to zero to do this.)

b) Determine the *Zero State Response* (ZSR), $x_{ZSR}(k)$. This is the part of $x(k)$ due to the input alone (Set all initial conditions on x to zero to do this.)

c) Find the total response, $x(k) = x_{ZIR}(k) + x_{ZSR}(k)$.

d)Find the transfer function, $H(z) = \frac{X(z)}{U(z)}$. Does this come from the ZIR or the ZSR?

e) Determine $x(0), x(1), x(2), x(3)$ and $x(4)$ from your answer to part **c.** Compare these with the know initial conditions, and use the difference equation to determine values for $x(3)$ and $x(4)$ to use in your comparison.