ECE-520: Discrete-Time Control Systems
Homework 4

Due: Tuesday January 9 in class

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

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

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

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

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) For the matrix

$$A = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

a) Find the eigenvalues and characteristic equation for A.

b) Determine an expression for A^2 in terms of A and I and then show explicitly that the matrix A satisfies its own characteristic equation by using the A matrix and evaluating both sides of the equation.

c) Using the Cayley-Hamilton method (matching on eigenvalues), show that

$$e^{At} = \begin{bmatrix} e^t & 0 \\ te^t & e^t \end{bmatrix}$$

d) Compute e^{At} using the Laplace transform method.

4) Consider the discrete-time model

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

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

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

$$y(k) = [I]\begin{bmatrix} \underline{\hat{x}}(k) \\ u(k-1) \end{bmatrix}$$

5) 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 (zero order hold, this is our standard method of sampling) discrete-time system

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

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

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. Assume we want the system output to remain the same.

6) 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.

7) In this problem we will examine the effects of a zero order hold, and see how we do various things in Matlab. Most of these have been done for you, but you need to look through the files and try and understand as much as you can.

a) From the class website, download the Matlab file **DT_driver.m** and the Simulink file **DT_openloop.mdl.** From your modeling, load the Matlab files containing your models for your 1 degree of freedom systems. (Be sure they are all in the same directory)

b) Modify **DT_driver.m** to load your rectilinear (210) model file. Simulate the continuous time and discrete time system for an input of a step with amplitude 0.1 cm for 2 seconds using a sampling time of 0.05 seconds. This is the default, so you should not have to change anything except loading your file. Look at how the continuous and discrete time signals look, and the effects of the zero order hold. Turn in your plot.

c) The *Nyquist sampling rate* is given as $T_s = \frac{1}{2 f_m}$ where T_s is the interval between samples and f_m is the highest frequency content of the signal we are sampling. Based on this criteria, with a sampling rate of $T_s = 0.05$ seconds, what is the highest allowed frequency in our input signal, f_m? (f_m is in Hz)

d) **Modify DT_driver.m** and **DT_openloop.mdl** so the input is a sinusoid with a frequency of 1 Hz (you need to convert this to radians) and an amplitude of 0.1 cm. Simulate the system for 4 or 5 seconds and look at the output. In order to see the effects of the zero order hold, you may have to change the plotting commands to plot (t+Ts/2,…) due to the way Simulink does the sine wave.(It won't be exact this way either, but it's a better approximation.) Turn in your plots.

e) Modify **DT_driver.m** and simulate your system for 4 or 5 seconds for input sine waves with amplitude 0.1 cm and various frequencies. What happens when the input frequency is that given by the Nyquist criteria? What is the highest frequency sine wave you think you can use and still get a *reasonably good representation* of the input? Plot your results with the highest acceptable frequency (there is no correct answer, only look in intervals of 0.5 radians/sec, i.e., 1.5 Hz, 2.0 Hz, 2.5 Hz, …)

8) Repeat problem 7 for your torsional system, except the input should be 1 degree (convert to radians) and change the output labels on the graphs. Your output should also be in degrees (or degrees/sec). Since the plant is in radians your will need to convert your output. Be sure to read in the correct model file for the torsional system.