1. (30 pts) Continuous vs Discrete Time Control (Handout and Matlab)
For each part, hand in relevant Matlab code as well as plots. Use hold on to superimpose plots.
Given the following continuous time (CT) system

\[
\dot{x} = Ax + Bu = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -40 & -44 & -14 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u(t), \quad y = [4 \ 1 \ 0] x
\]

[4pts] a) Find the corresponding discrete time (DT) system \( x[n+1] = Gx[n] + Hu[n] \), \( y[n] = Cx[n] \) which can be found using the Matlab function \( \text{c2d}(\text{ctsys},T,'zoh') \), with sampling period \( T_s = 0.1 \) sec. (Note \( \text{ctsys} \) can be found from \( \text{ss}(A,B,C,D) \)). Compare eigenvalues for CT \( A \) and DT \( G \); are both systems stable?

[8pts] b) The continuous time system uses output feedback such that \( u(t) = r(t) - kCx(t) \). Find \( k \) such that damping factor associated with dominant poles \( \zeta \approx 0.3 \) (using \( \text{rlocus()} \) is sufficient). Plot the closed-loop step response using Matlab.

[6pts] c) Consider the CT system having output feedback applied in discrete time using a D/A converter such that \( u[n] = u(nT) = r[n] - k_dCx[n] \). (Use \( \text{rlocus()} \) on the discrete time system to find \( k_d \) to give \( \zeta \approx 0.3 \)). Thus the closed loop DT system has state equation \( x[n+1] = (G - Hk_dC)x[n] + Hr[n] \), \( y[n] = Cx[n] \). Plot the step response for the closed loop step response on the same axes as the CT step response of part b).

[6pts] d) Consider the step response for the DT converted system (using \( \text{c2d}() \)) for \( \dot{x} = (A - BkC)x + Br \), and plot. Explain why the step response using output feedback for the DT system from part c) does not look like this DT version of the step response.

[6pts] e) Use Matlab (iteratively if necessary) to find a sampling period \( T_s \) which gives a closed-loop step response for DT that is “reasonably close” (within 5%) to the CT closed-loop step response (using \( k_d = k \)). Determine DT closed-loop pole locations, and plot the DT step response on same axes as part c. How does the \( T_s \) found compare to the fastest eigenvalue of the openloop CT system?

2. (10 pts) Laplace to Z conversion (Nise 13.3)
Given \( H(s) = \frac{1}{s + a} \) and sample rate \( T \), find \( H(z) \) using the definition of Z transform, i.e.

\( H(z) = \sum_{k=0}^{\infty} h(kT)z^{-k} \).

3. (10 pts) Z transform (Nise 13.3)
For each \( F(z) \), find \( f(kT) \) using partial fraction expansion.

a) \( F(z) = \frac{(z+4)(z+1)}{(z-0.3)(z-0.6)} \)

b) \( F(z) = \frac{(z+0.2)(z+1)}{z(z-0.1)(z-0.2)(z-0.3)} \)

4. (10 pts) Final value (Nise 13.7)
For each linear difference equation with input \( u(kT) \), find \( \lim_{k \to \infty} f(kT) \) for a unit step input.

a) \( f((k+1)T) = u(kT) + \frac{1}{2} f(kT) \)

b) \( f(kT) = u((k-1)T) - u((k-2)T) - f((k-1)T) - \frac{1}{4} f((k-2)T) \)
5. (20 pts) SS to TF (Nise 3.6, 13.3, 13.4, DT handout)
Given the following discrete time (DT) system, with sample period \( T = 1 \):

\[
x(k + 1) = Gx(k) + Hu(k) = \begin{bmatrix} 0 & 1 \\ \frac{1}{4} & \frac{5}{12} \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u
\]

(a) Find the transfer function \( \frac{X(z)}{U(z)} \).
(b) Is the system BIBO stable?
(c) Find \( x[k] \) for \( k \geq 0 \) for a unit step input and zero initial conditions using partial fraction expansion.

6. (20 pts) Transient performance using gain compensation (Nise 13.8, 13.9)
Given a CT plant \( G_1(s) = \frac{K}{s(s+1)(s+3)} \).

(a) For the block diagram below find \( \frac{C(z)}{R(z)} \).
(b) With sample period \( T_s = 0.1 \), find \( G_1(z) \), the Z transform of \( G_1(s) \) (ideal sampling, as in Table 13.1).
(c) Sketch the root locus for \( G_1(z) \) in unity gain feedback, and find the range of \( K \) for stability (Matlab ok for \( K \)).
(d) With unity gain feedback, find the value of \( K \) for damping factor \( \zeta = 0.5 \), and note this \( K \) value in root locus for CT and DT.
(e) Plot step response for the closed-loop CT and DT system in Matlab, and compare.