Indian Institute of Technology, Bombay — Chemical Engineering
CL686: Advanced Process Control — **Assignment 1, 2025**

Date Due: 19 Jan 2025, 11.59 PM (Note that time is not 12 PM midnight since that is confusing to set.)

Mode of Submission: Moodle submission

Late Submission Policy: Submissions late by 0-24 hrs will incur a 25% penalty, and submissions late by 24-48 hrs will incur a 50% penalty. Submissions later than 48 hrs will not be considered.

Topic: Dynamic Simulations using Nonlinear and Linear Dynamic Models

AIM: Simulate open loop Nonlinear and Linear Dynamic Models and compare the outputs

Task at Hand

1. Obtain discrete-time linear state space model from given continuous time state space model.
2. Simulate Nonlinear continuous time model and linear discrete-time model with given Step input for the specified number of samples.
3. Simulate a Nonlinear continuous time model and linear discrete-time model with given Pseudo random Binary Signals (PRBS) input for the specified number of samples.

System Description

Consider a nonlinear system governed by the following set of coupled nonlinear ODEs,

$$\frac{dh_1}{dt} = -\frac{a_1}{A_1}\sqrt{2gh_1} + \frac{a_3}{A_1}\sqrt{2gh_3} + \frac{\gamma_1 k_1}{A_1}v_1 \tag{1}$$

$$\frac{dh_2}{dt} = -\frac{a_2}{A_2}\sqrt{2gh_2} + \frac{a_4}{A_2}\sqrt{2gh_4} + \frac{\gamma_2 k_2}{A_2}v_2 \tag{2}$$

$$\frac{dh_3}{dt} = -\frac{a_3}{A_3}\sqrt{2gh_3} + \frac{(1-\gamma_2)k_2}{A_3}v_2 + \frac{\gamma_3}{A_3}F_d \tag{3}$$

$$\frac{dh_4}{dt} = -\frac{a_4}{A_4}\sqrt{2gh_4} + \frac{(1-\gamma_1)k_1}{A_4}v_1 + \frac{(1-\gamma_3)}{A_4}F_d \tag{4}$$

The initial steady-state operating conditions of the system are listed in the table below:

Variable	Operating point
A_1, A_3 $[cm^2]$	28
A_2, A_4 $[cm^2]$	32
a_1, a_3 $[cm^2]$	0.071
a_2, a_4 $[cm^2]$	0.057
g $[cm/s^2]$	981
$k1, k2$ $[cm^3/Vs]$	3.14, 3.29
$\gamma_1, \gamma_2, \gamma_3$	0.43, 0.34 , 0.4

- **System State:** h_1, h_2, h_3, h_4 are the liquid level in the quadruple tank system.
- **Process inputs:** v_1 and v_2 are the voltage applied to the pump.
- **Disturbance inputs:** F_d act as the disturbance input to the process.

Condition for Dynamic Simulation:

- **Steady state inputs:**
  $\mathbf{U}_s = [3.15 \quad 3.15]^T$; $\mathbf{D}_s = 2$
- **Equilibrium/Steady State Operating Point:**
  $\mathbf{X}_s = [14.30 \quad 16.84 \quad 5.90 \quad 7.33]^T$
- **Sampling interval**
  $T = 4$ sec
- **Manipulated Input profile:**

$$\mathbf{U}(k) = \mathbf{U}_s + \mathbf{u}(k)$$

1. **For step input:**
   where $\mathbf{u}(k)$ is generated as follows
   $$\mathbf{u}(k) = \begin{cases} [0 \quad 0]^T & \text{for k<50} \\ [0.5 \quad 0.5]^T & \text{for } 50 \leq \text{k<300} \end{cases}$$
   Simulate for $N_s = 300$ number of samples
2. **For PRBS input**
   where $\mathbf{u}(k) = [u_1(k) \;\; u_2(k)]^T$ is vector of PRBS input with amplitude range of $[-0.25 \;\; 0.25]$ and frequency band (in rad/sec) of $[0 \quad 0.05]$. MATLAB inbuilt function *idinput* can be used to generate PRBS input.
   $\mathbf{u} = idinput(N_s, Type, Band, Range)$ where, Type is '*prbs*'. Band and Range as specified above. Simulate for $N_s = 2000$ number of samples

- **Disturbance input D:**

$$\mathbf{D}(k) = \mathbf{D}_s + \mathbf{d}(k)$$

where $\mathbf{d}(k) \sim \mathcal{N}(0, 1)$ i.e. $\mathbf{d}(k)$ is a Gaussian random variable with mean 0 and variance 1.

Note: This random number will keep changing at each time instant. For generating random numbers use the MATLAB command 'randn'.

- **Continuous time model:**
  Continuous time linear perturbation model matrices (**A**, **B**, **H**, **C**) are provided in the .mat file which can be loaded in your code. The model is of the form:

$$\frac{d\mathbf{x}}{dt} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{H}\mathbf{d}$$
$$\mathbf{y} = \mathbf{C}\mathbf{x}$$

Open Loop Plant Dynamic Simulation Algorithm

Simulate open loop plant behaviour for N_s number of samples

- **Initialization**

  Load system parameters and $(\mathbf{X}_s, \mathbf{U}_s, D_s)$ from data file

  Set $\mathbf{X}(0) = \mathbf{X}_s, \quad \mathbf{U}(0) = \mathbf{U}_s, \quad \mathbf{D}(0) = \mathbf{D}_s; \quad \mathbf{Y}(0) = \mathbf{Y}_s$

  Load continuous time linear perturbation model matrices $(\mathbf{A}, \mathbf{B}, \mathbf{H}, \mathbf{C})$

  Generate discrete-time linear perturbation model matrices $(\mathbf{\Phi}, \mathbf{\Gamma}_u, \mathbf{\Gamma}_d)$ for the specified T

  Create matrices for storing state and input data

- **Dynamic Simulation**

  FOR k = 1 to $N_s - 1$

  - Generate $\mathbf{u}(k)$ and $d(k)$
    $\mathbf{U}(k) = \mathbf{U}_s + \mathbf{u}(k)$
    $\mathbf{D}(k) = \mathbf{D}_s + \mathbf{d}(k)$
    Solve for $\mathbf{x}(k+1) = \mathbf{\Phi x}(k) + \mathbf{\Gamma u}(k) + \mathbf{\Gamma}_d \mathbf{d}(k)$ *% Linear plant simulation*
    Find $\mathbf{X}_L(k+1) = \mathbf{X}_s + \mathbf{x}(k+1)$
    Solve for $\mathbf{X}(k+1) = F\left[\mathbf{X}(k), \mathbf{U}(k), D(k)\right]$ *% Nonlinear Plant Simulation*

  END FOR

  Note: For the nonlinear plant simulation, use an ode-integrator such as explicit Euler or Runge-Kutta method. In Matlab, you can use inbuilt integrators such as ode45.

- Display Simulation Results for step input as well as for PRBS input

  - Plot $\mathbf{X}_i(k)$ v/s Time and $\mathbf{X}_{L,i}(k)$ v/s Time in same figure for $i = 1, 2, 3, 4$
    Plot $\mathbf{X}_{L,i}(k) - \mathbf{X}_i(k)$ v/s Time for $i = 1, 2, 3, 4$
    Plot $\mathbf{U}_i(k)$ v/s Time for $i = 1, 2$
    Plot $\mathbf{D}(k)$ v/s Time

MATLAB Code To Submit

- Submit three files in a zipped folder: (i) The main MATLAB file, which upon execution, will generate the various plots. No other input/action should be required to generate these. Keep the file name as *Assn_01_ROLLNO.m* where ROLLNO is your roll number. (ii) The function file where the ODEs are written and it will be called by the main script file. This function file should be named as *System_Dynamics_ROLLNO.m*. (iii) The .mat file provided for the continuous-time linear perturbation model should be present in the same folder. [**Get in touch with TA for any issues.**]

- Make sure that the plots are labelled properly with axes' names, titles, and legends.

Learning is fun. Best of Luck!