Assignment 4

- These are reference solutions from MATLAB Grader (code in gray boxes is fixed on Grader)

Problem 1a: Comparison of State-Space and Step-Response Models **(2 + 2 points)**

```
G=tf(5,[4.2 1],'inputdelay',0.15);
h=0.5;
% State space model parameters
Gd=c2d(ss(G),h);
A=Gd.A; B=Gd.B; C=Gd.C;
% Step response model parameters
n=40;
S=step(G,[0:h:n*h]); S=S(2:end);
S=round(S,1);    % A added to ensure Y_fsr and Y_ss are different

% Input moves
UALL=[0, 1, 1, 1, 2, 0.5, 0.5, 0.5, 0.5, 0, 0, 0,0,0,0,0,0,0,0,0,0];
% NOTE
% Y_fsr is 1*20 array of results from step-response model
% Y_ss  is 1*20 array of results from state-space model
% -------------------------------------------------------
% Please start typing from lines below
```

```
N=20;
% STEP RESPONSE MODEL SIMULATIONS
YHAT=zeros(n,1);
dU=UALL-[0, UALL(1:end-1)];
Y_fsr=zeros(1,N);
for i=1:N
    YHAT=[YHAT(2:end);YHAT(end)]+S*dU(i);
    Y_fsr(i)=YHAT(1);
end

% STATE SPACE MODEL SIMULATIONS
xk=zeros(2,1);
Y_ss=zeros(1,N);
for i=1:N
    xk=A*xk+B*UALL(i);
    Y_ss(i)=C*xk;
end
```

$$\tilde{Y} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \Big\updownarrow n$$

$$\Delta U = \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix} - \begin{bmatrix} 0 \\ u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}$$

$$\tilde{Y}_{k+1} = \underbrace{M\tilde{Y}_k}_{\text{upward shift.}} + S\Delta u_k$$

$$x_{k+1} = Ax_k + Bu_k$$

Problem 1b: Repeat with arbitrary input moves **(2 + 2 points)**

The only change in the code is the first line below the gray box (yellow-highlight) is changed to:

```
N=20; UALL=U;
```

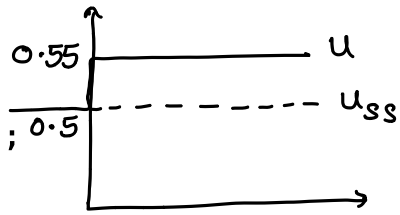
My code gives the appropriate response for arbitrary U with this minimal change! What about yours?

Problem 2: Nonlinear and Linearized Model Simulations **(4 + 2 points)**

Based on my past experience, I thought the nonlinear model would be easy for you since solving using `ode15s` is done in undergraduate and was shared in Week-1 videos. Please let us know if you found this assignment a bit tough and how we can address this.

```
% Compute the steady state values
uss=0.5;
xss=fsolve(@(x) rxtrFun(x,uss), [1;1]);
% You may use the above as initial condition or use your own
% Please report your solution in 1*25 array C_nonlin
% Start typing from line below
```

```
h=0.2; tEnd=5;     % <-- Sampling and Final times
u0=0.55;
[tSol,XSol]=ode15s(@(t,x) rxtrFun(x,u0), [0:h:tEnd], xss);
CSol=XSol(2:end,2);
C_nonlin=CSol';
```



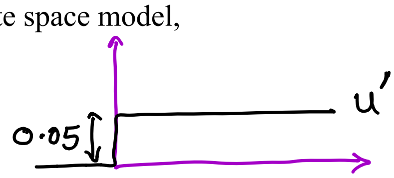

```
% ==================================================================
%% Function for use with ode15s (optional)
function dx=rxtrFun(x,u)
Area=0.2; kappa=0.5; k=1.5;
Cin=4;
h=x(1);
C=x(2);

dx(1,1)=u/Area-kappa/Area*sqrt(h);
dx(2,1)=u/(Area*h)*(Cin-C)-k*C^2;
end
```

IN DEVIATION VAR.
$u' = u - u_{ss}$

The linear case is exactly similar to Problem 1b, but with (i) A different state space model, and (ii) with `UALL=[0.05, 0.05, 0.05, ..., 0.05];`

Note that the resulting `Cdash` vector is deviation from steady state values.



Problem 3: Simulation of MIMO system to series of step inputs **(6 points)**

This problem is similar to the step-response part of Problem 1a. Note that the model has 2 outputs, instead of one. Hence, only the matrix sizes change, but the code remains same otherwise. The lines of code that have changed compared to Problem 1a are highlighted in yellow. You will notice that with `ny=1`, the code of this problem is exactly same as that of Problem 1a.

```
load SVal_hw4.mat         % Load S matrix
nu=1; ny=2;               % Number of inputs and outputs
n=25;                     % Number of time steps in S matrix
U=(0.5-rand(1,41))/10;    % Input moves
% The first line loads your S matrix
% ***** Start typing from the line below *****
```

```
Yhat=zeros(ny*n,1);
dU=U-[0, U(1:end-1)];
C_result=zeros(1,40);
for k=1:40
    Yhat=[Yhat(ny+1:end); Yhat(end-ny+1:end)] + S*dU(k);
    C_result(k)=Yhat(2);
end
```

$$\tilde{Y}_0 = \begin{bmatrix} \begin{bmatrix}0\\0\end{bmatrix} \\ \begin{bmatrix}0\\0\end{bmatrix} \\ \vdots \\ \begin{bmatrix}0\\0\end{bmatrix} \end{bmatrix}$$

n-times repeat

$(n_y \cdot n) \times 1$ VECTOR