**E6225**

All the assignment must submitted by

Due: Saturday, week 7, 25 Sep 2021

Format: The solution must be submitted in word document by e-mail

**Continious Assignment 1**

* 1. A process transfer function is described as



Write a Simulink/Mathlab software program to simulate the process (note that this is the process you are not to make any change in the following works) and to identify the parameters of the transfer function

1. Frist order plus time delay model using classic technics: two points method, log method and area method
2. Frist order plus time delay model using least squares mentod under open loop step test in time domain
3. Frist order plus time delay under open loop step test using least squares mentod in frequency domain
4. Using relay feedback to generate sustained oscillation and using the available informatuion to calculate the parameters of frist order plus time delay model.
5. For rach method, plot Nyquist chart for both the original transfer function and the identified transfer function to compare the identification results

Note: You need to show the program and step by step results.

Solution:

**Question 1:**

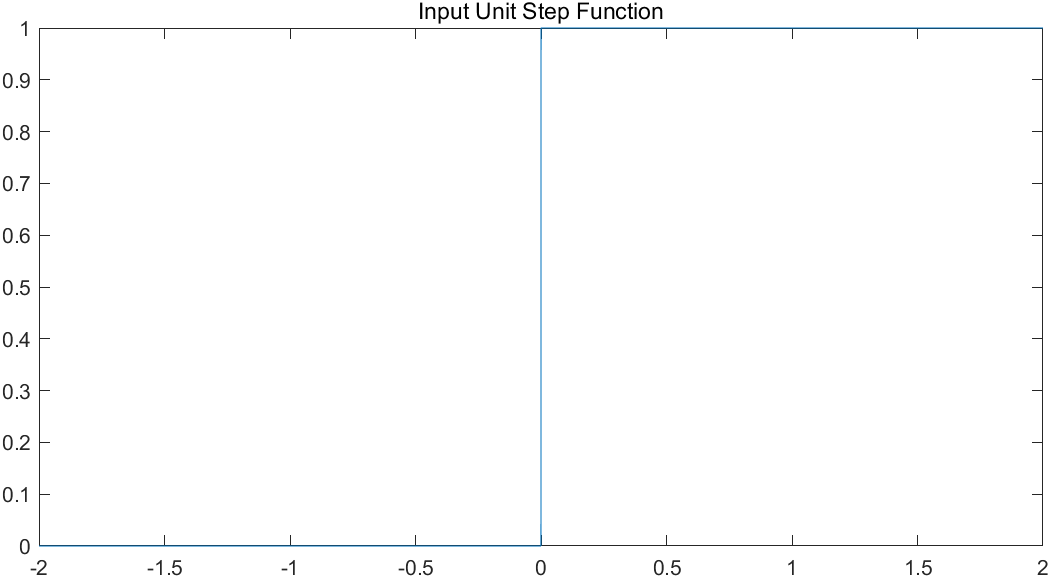
Using Unit Step Function as input.

syms x;

subplot(2,2,1);

fplot(heaviside(x), [-2, 2]);

title('Input Unit Step Function')



Then, plot the step response of given transfer function.

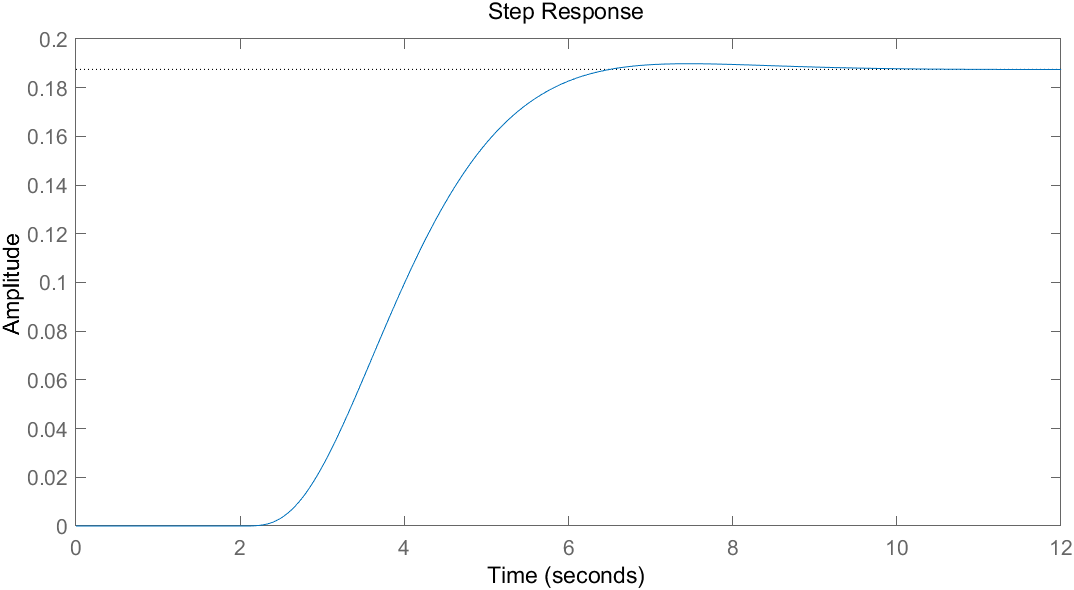
% Plot the step response of original transfer function

s = tf('s');

G0 = 3 \* exp(-2 \* s) / ((s + 4) \* (s^3 + 5\*s^2 + 7\*s + 4));

subplot(2,2,2);

step(G0);



① **Two Points Method**

As shown in the plot above, the steady state value is 0.1875. After applying two points method, t1 is about 3.406s and t2 is about 4.269s. Finally, T and L can be determined by t1 and t2.

% Two Points Method

steadyStateY = 0.1875;

y1 = steadyStateY \* 0.284;

y2 = steadyStateY \* 0.632;

% Find t1 & t2

t = 0:0.001:20;

Y0 = step(G0, t);

subplot(2,2,3);

plot(t, Y0);

hold on;

% Paramaters

t1 = 3.406;

t2 = 4.269;

A = 1;

K = steadyStateY / 1;

T = 1.5 \* (t2 - t1);

L = 0.5 \* (3 \* t1 - t2);

% Plot the step response of transfer function of FOPTD model

Gp = K \* exp(- L \* s) / (T \* s + 1);

subplot(2,2,4);

step(Gp);

% Compare two transfer functions

Yp = step(Gp, t);

subplot(2,2,3);

plot(t, Yp);

xlabel('Time (seconds)');

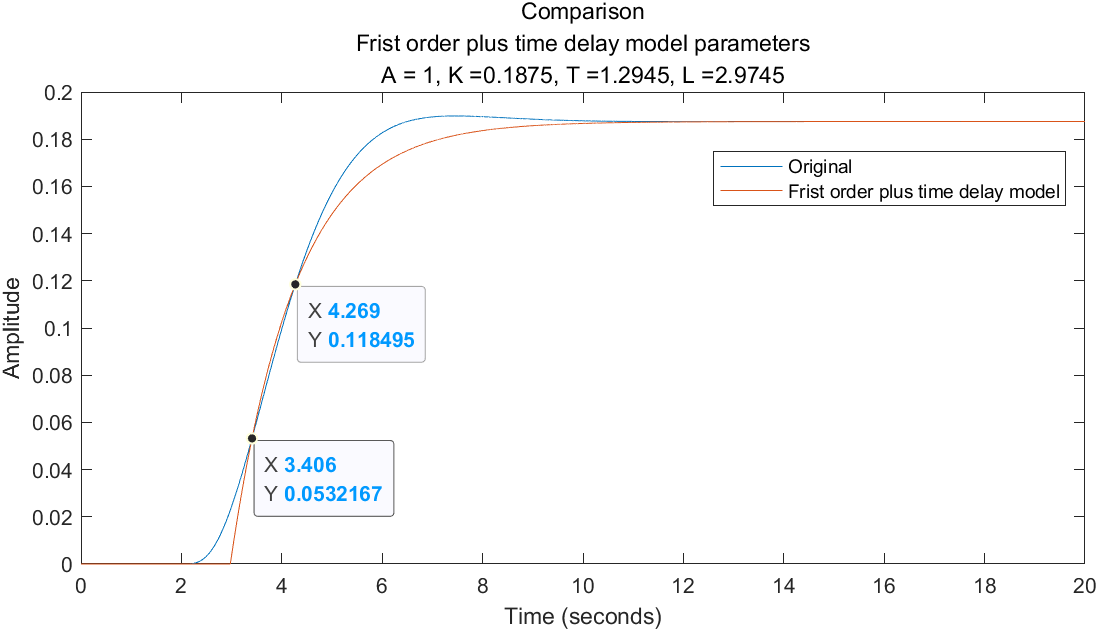
ylabel('Amplitude');

legend('Original','Frist order plus time delay model');

title('Comparison');

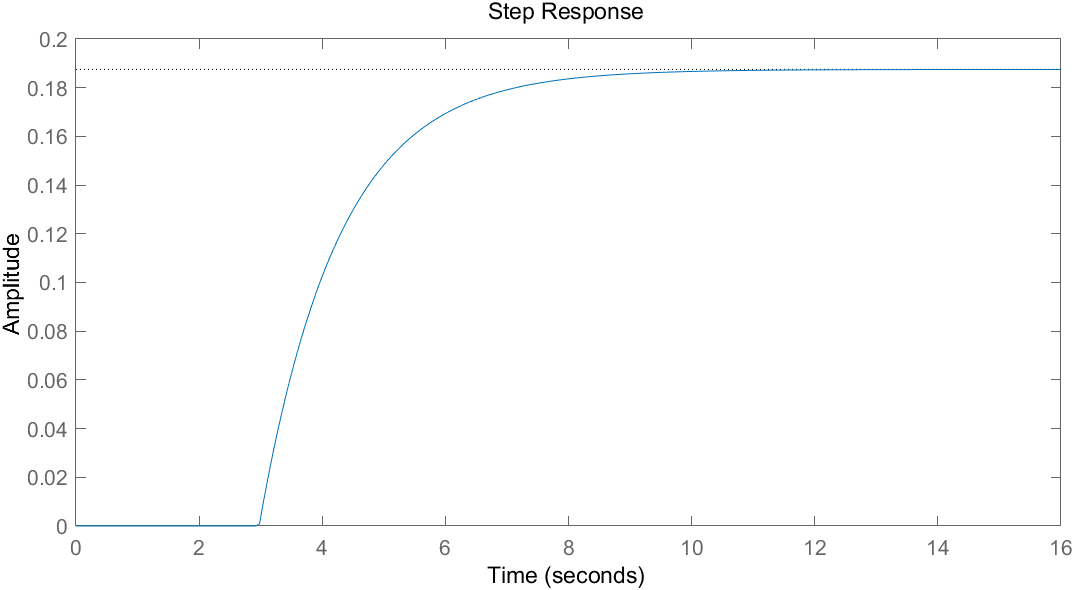
subtitle({'Frist order plus time delay model parameters', ['A = ', num2str(A), ', K =' num2str(K), ', T =', num2str(T), ', L =', num2str(L)]});

sgtitle('Two Points Method');



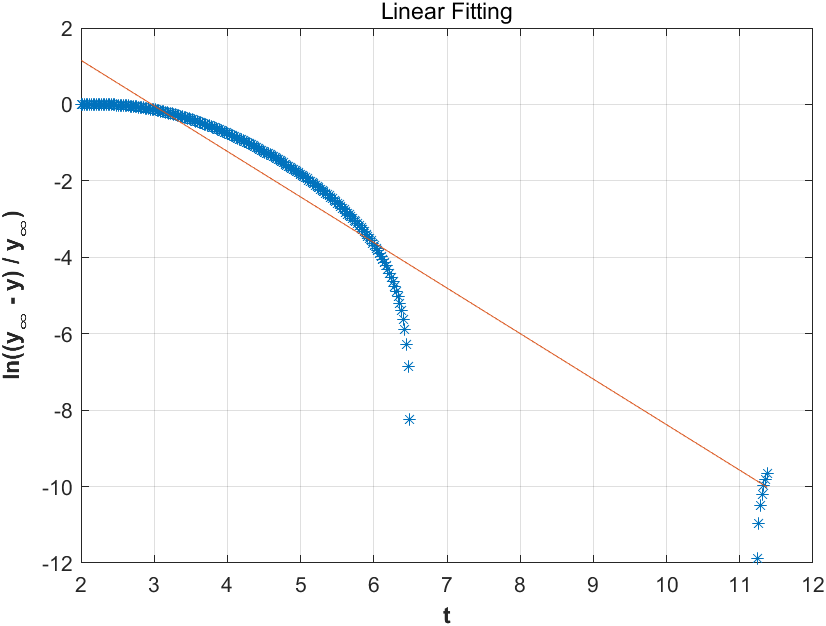
**K = 0.1875, T = 1.2945, L = 2.9745**

The step response of First order plus time delay model using two points method is shown in the below image.



② **Log Method**

By sampling on the original step response function, the transformation of the step-response data against time t can be computed. As the plot of the data against time t is supposed to be a straight line, the line shown in the below image can be determined by using linear fitting.



Here is the corresponding matlab code.

% Get the array of t and new y

s = tf('s');

G0 = 3 \* exp(-2 \* s) / ((s + 4) \* (s^3 + 5\*s^2 + 7\*s + 4));

[y, t] = step(G0);

k = size(y);

yln = [];

tln = [];

for n = 1 : 1 : k

if y(n, 1) > 0 && 0.1875 - y(n, 1) >= 0

yln = [yln, log((0.1875 - y(n, 1)) / 0.1875)];

tln = [tln, t(n, 1)];

end

end

% Linear fitting

h = polyfit(tln', yln', 1);

display(h);

subplot(1, 2, 1);

plot(tln, yln, '\*', tln, polyval(h, tln));

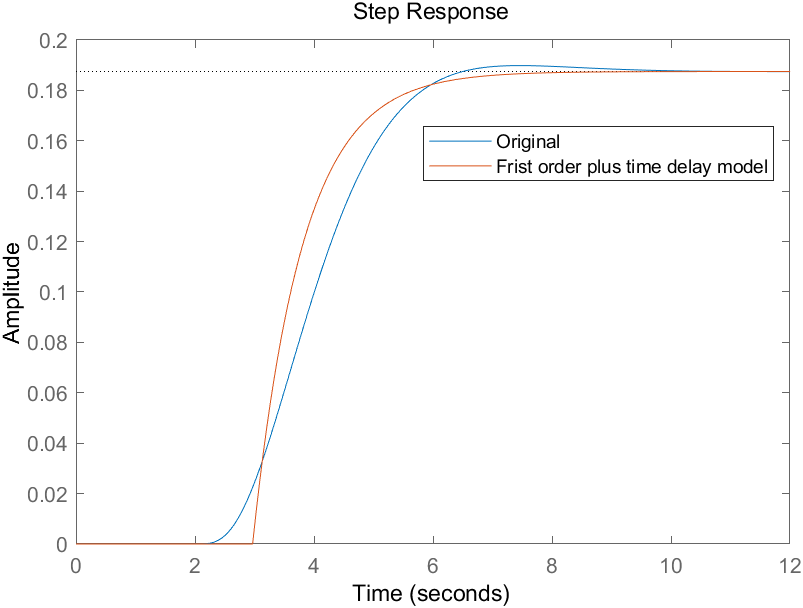
grid on;

title('Linear Fitting');

xlabel('t', 'FontWeight', 'bold');

ylabel('ln((y\_{¡Þ} - y) / y\_{¡Þ})', 'FontWeight', 'bold');

Then, parameter T, L can be easily determined by the scope of the straight line and the point where the line meet the t-axis. Finally, the First Order Plus Time Delay model using Log Method is shown in the below image.



**K = 0.1875, T = 0.8390, L = 2.9683**

% Parameters

steadyStateY = 0.1875;

A = 1;

K = steadyStateY / 1;

T = 1 / -h(1, 1);

L = h(1, 2) \* T;

subplot(1, 2, 2);

Gp = K \* exp(- L \* s) / (T \* s + 1);

step(G0, Gp);

legend('Original','Frist order plus time delay model');

③ **Arae Method**

According to the area method, the area of A0 and A1 can be computed by integration.

% Compute A0 & A1

s = tf('s');

G0 = 3 \* exp(-2 \* s) / ((s + 4) \* (s^3 + 5\*s^2 + 7\*s + 4));

t = 0 : 0.01 : 4;

y = step(G0, t);

num = size(y);

y0 = zeros(num(1, 1), 1);

steadyStateY = 0.1875;

A = 1;

K = steadyStateY / 1;

for n = 1 : 1 : num(1, 1)

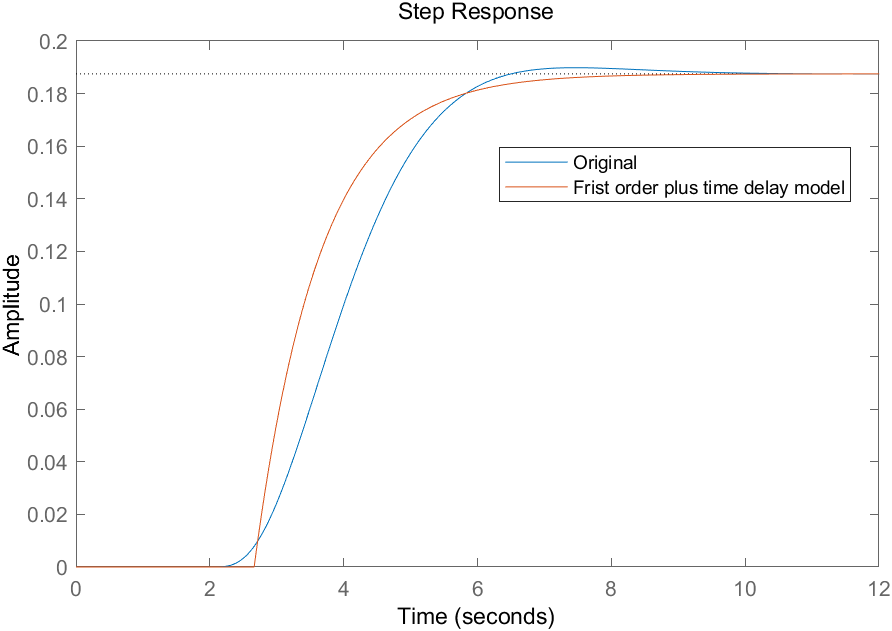
y0(n, 1) = steadyStateY - y(n, 1);

end

A0 = trapz(t, y0);

A1 = trapz(t, y);

Thus, T and L can be determined by A0 and A1.



**K = 0.1875, T = 0.9783, L = 2.6618**

% Parameters

T = (exp(1) \* A1) / K;

L = (A0 / K) - ((exp(1) \* A1) / K);

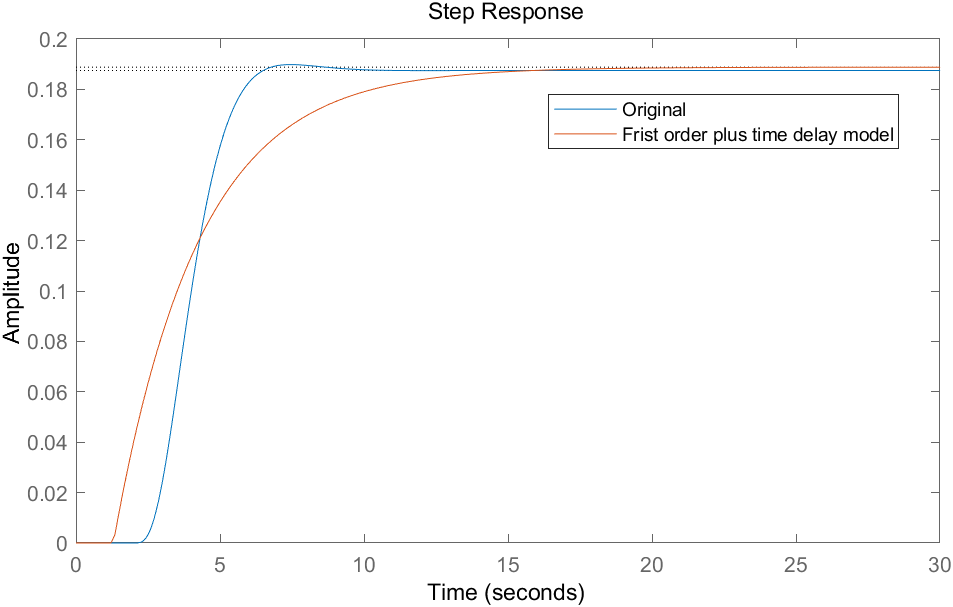
Gp = K \* exp(- L \* s) / (T \* s + 1);

step(G0, Gp);

legend('Original','Frist order plus time delay model');

**Question 2:**

Using least square method in time domain, the theta matrix can be determined and parameters a1, b1 and L can be computed using thetas in the matrix.



**a1 = 0.3406, b1 = 0.0643, L = 1.2992**

Here is the Matlab code.

s = tf('s');

G0 = 3 \* exp(-2 \* s) / ((s + 4) \* (s^3 + 5\*s^2 + 7\*s + 4));

t = 0 : 0.001 : 50;

y = step(G0, t);

h = 1;

sizeY = size(y);

psi = zeros(sizeY(1, 1), 3);

for k = 1 : 1 : sizeY(1, 1)

psi(k, 1) = -getIntegral(t(1, k), y);

psi(k, 2) = -h;

psi(k, 3) = t(1, k);

end

theta = inv(psi.' \* psi) \* psi.' \* y;

a1 = theta(1, 1);

b1 = theta(3, 1);

L = theta(2, 1) / theta(3, 1);

Gp = b1 \* exp(- L \* s) / (s + a1);

step(G0, Gp);

legend('Original','Frist order plus time delay model');

function integral = getIntegral(Tn, Y)

if Tn == 0

integral = 0;

return;

end

t1 = 0 : 0.001 : Tn;

sizet1 = size(t1);

yt = Y(1 : sizet1(1, 2), 1);

integral = trapz(t1, yt.');

end

**Question 3:**