Department of Electronic and Telecommunication Engineering
University of Moratuwa

EN2143 Electronic Control Systems

Assignment 1

System Identification and PID Control

Prepared by:

MADUWANTHA L.H.H – 200370K

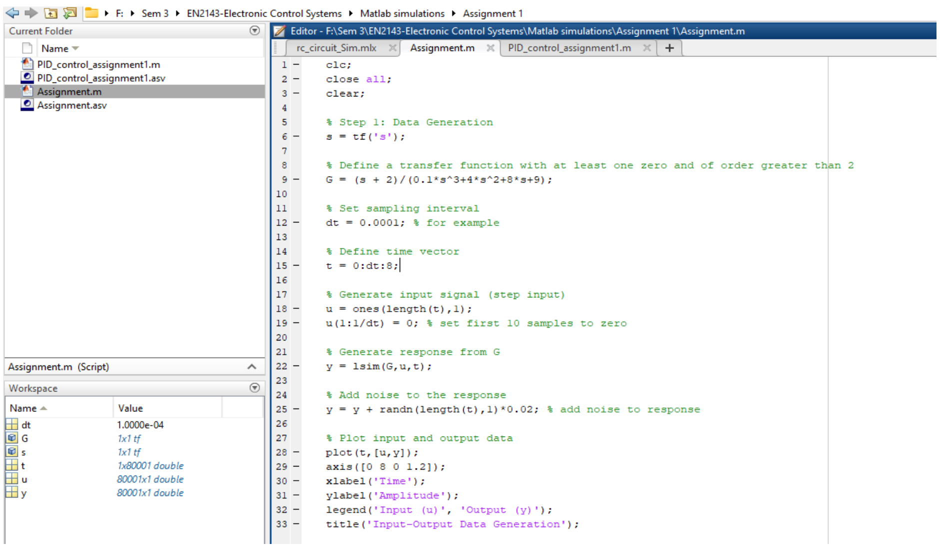
Date:

5th March 2024

Code for data generation.

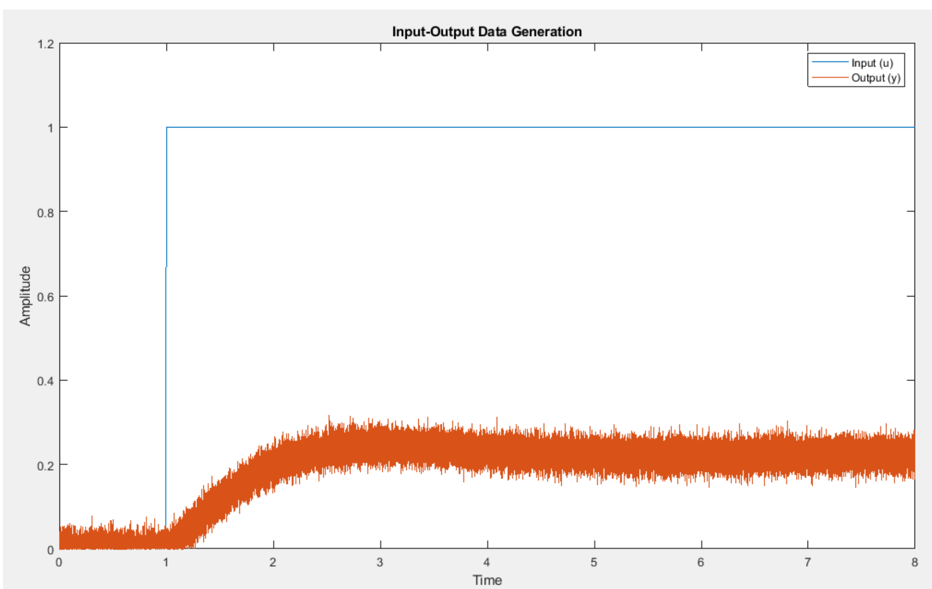
Transfer function $= \frac{s+2}{0.1s^3+4s^2+8s+9}$

Sample time = 0.0001s

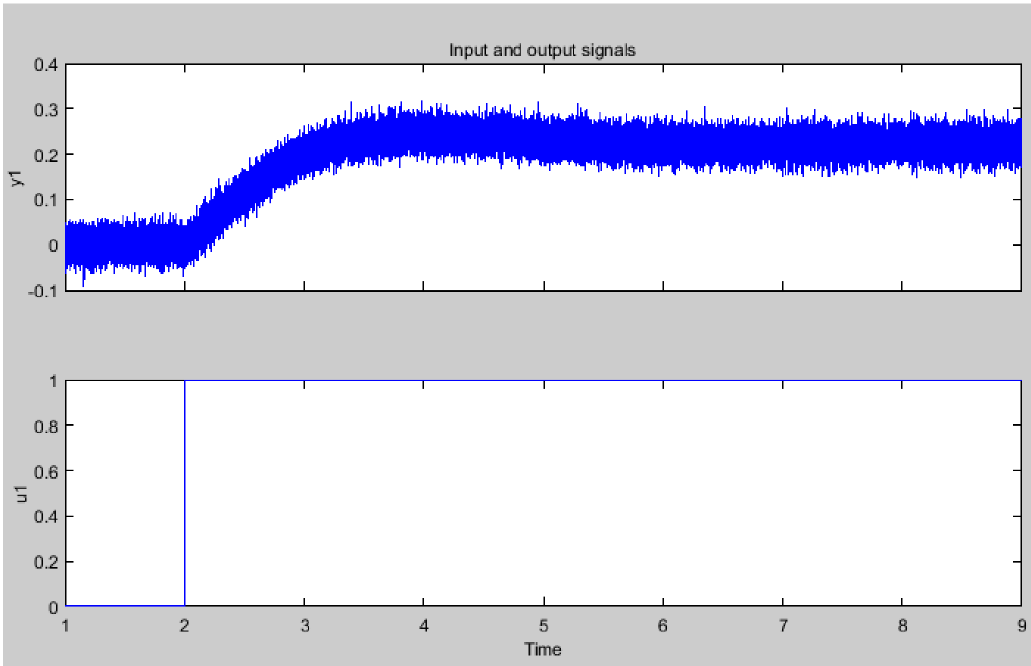


```
clc;
close all;
clear;

% Step 1: Data Generation
s = tf('s');

% Define a transfer function with at least one zero and of order greater than 2
G = (s + 2)/(0.1*s^3+4*s^2+8*s+9);

% Set sampling interval
dt = 0.0001; % for example

% Define time vector
t = 0:dt:8;

% Generate input signal (step input)
u = ones(length(t),1);
u(1:1/dt) = 0; % set first 10 samples to zero

% Generate response from G
y = lsim(G,u,t);

% Add noise to the response
y = y + randn(length(t),1)*0.02; % add noise to response

% Plot input and output data
plot(t,[u,y]);
axis([0 8 0 1.2]);
xlabel('Time');
ylabel('Amplitude');
legend('Input (u)', 'Output (y)');
title('Input-Output Data Generation');
```

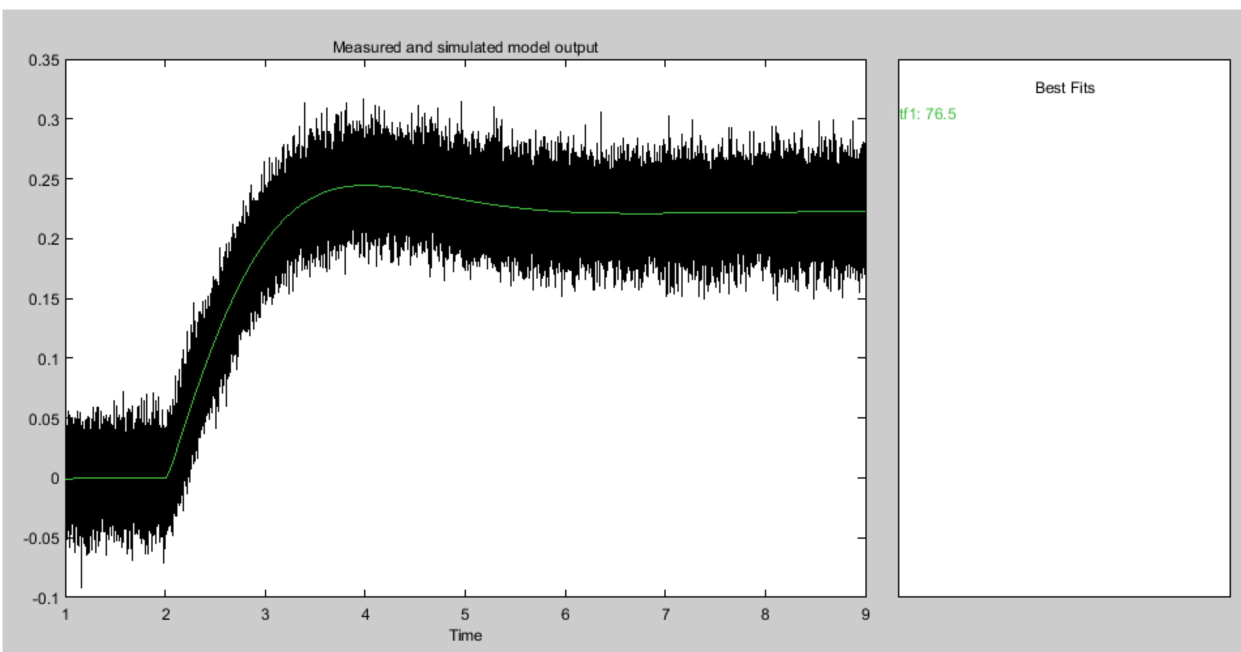
Input output data graph.



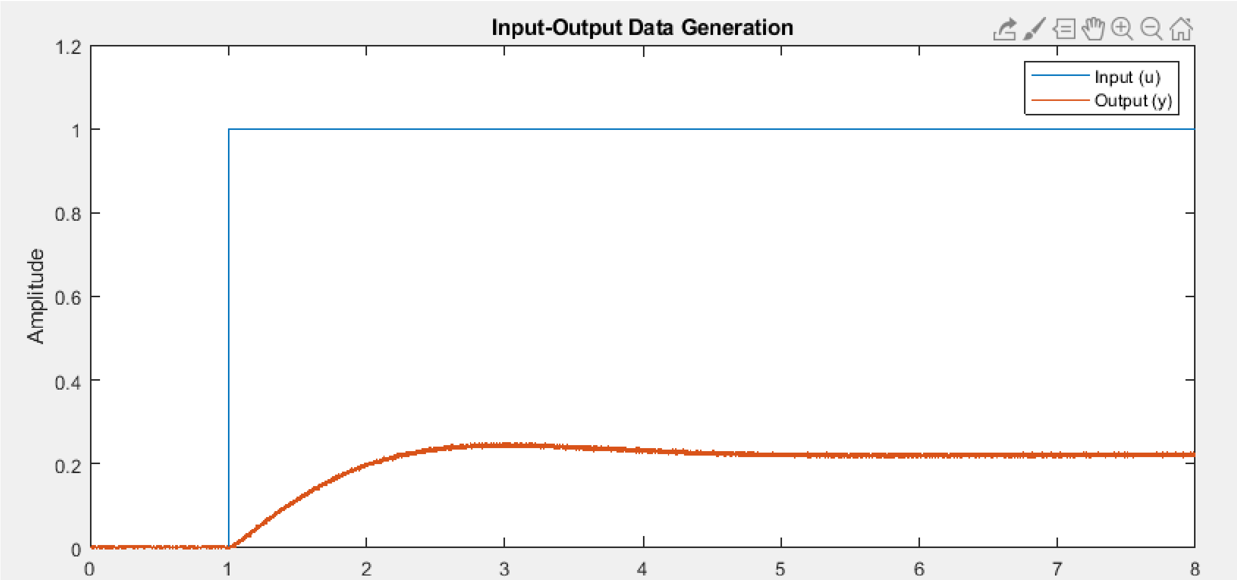
It is same as the time plots of imported data.



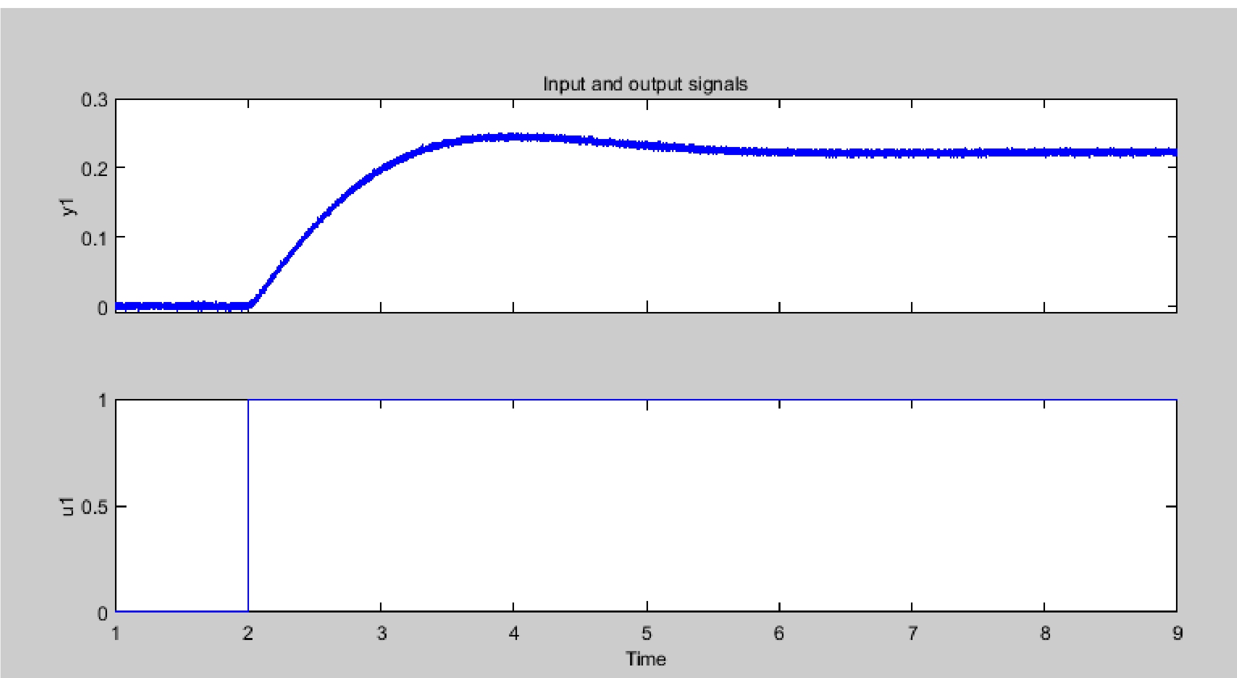
Model output and data fitting percentage are 76.5%. I believe this value is low due to the higher noise response.



In the above graphs, since the noise level is excessively high, I have reduced the noise component by 10%. The graphs now appear as follows:

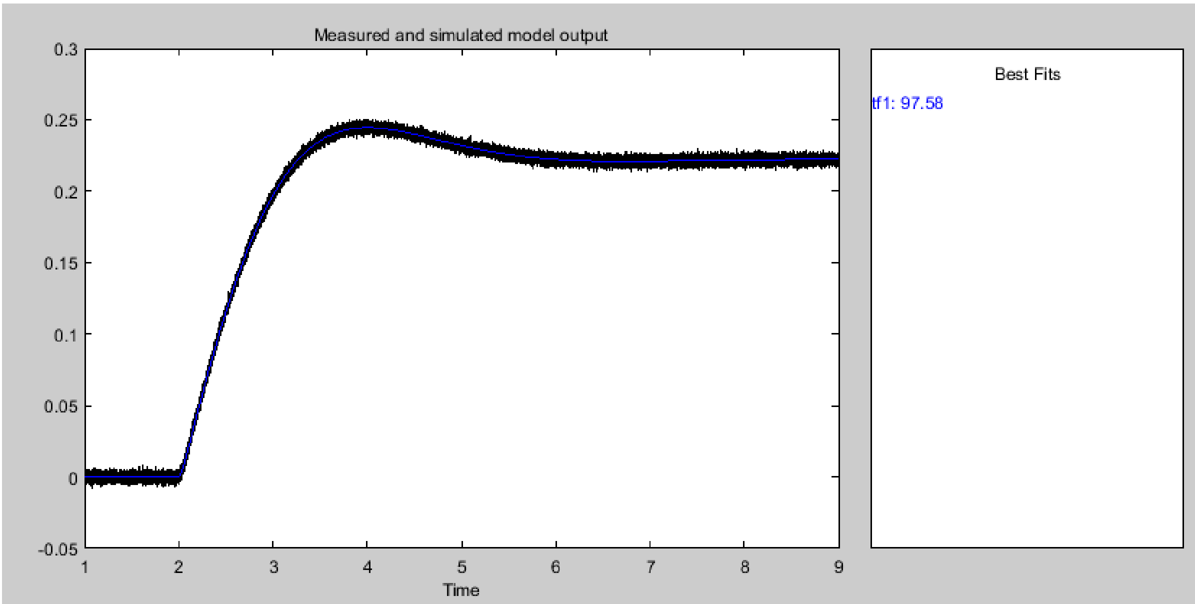


It is same as the time plots of imported data. The generated data has been properly imported.



After reducing the noise component, we can clearly observe that the model output and data fitting percentage have increased to 97.56%.

Reducing the noise component enhances the signal-to-noise ratio, allowing for more accurate data fitting. With reduced noise interference, the model output aligns closely with the actual data, resulting in a higher fitting percentage. This demonstrates the improved accuracy and reliability of the model under reduced noise conditions.



Code for manually calibrating PID parameters.

ic Control Systems ▸ Matlab simulations ▸ Assignment 1

Editor - F:\Sem 3\EN2143-Electronic Control Systems\Matlab simulations\Assignment 1\PID_control_assignment1.m

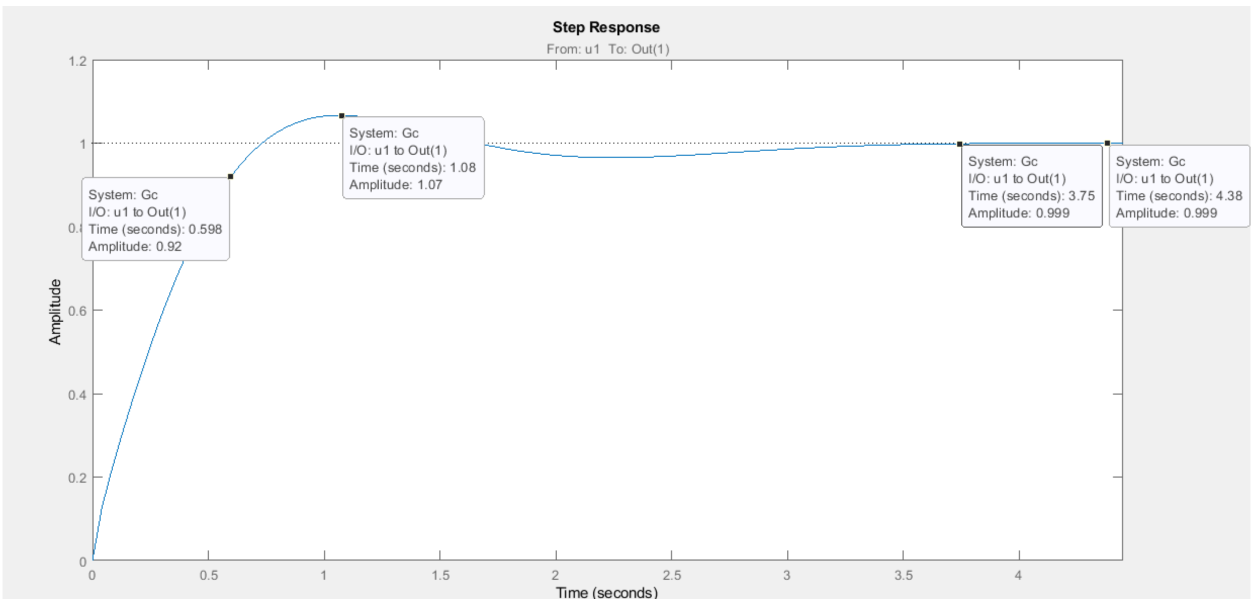
rc_circuit_Sim.mlx | Assignment.m | PID_control_assignment1.m | PIDtune.m | +

```
% Step 2: PID Controller Design
Gs = tf1;

% PID Controller Design
Kp = 1.2;
Ki = 0.6;
Kd = 0.1;
controller = pid(Kp, Ki, Kd);

% Closed-loop system
sys_cl = feedback(Gs*controller, 1);

% Step 3: Visualize Step Response
figure;
step(sys_cl);
title('Step Response of the Controlled Plant');
xlabel('Time');
ylabel('Amplitude');
```

PID tune

```
% Define the transfer function Gs
s = tf('s');
Gs = tf1; % Your transfer function

% Design PID controller using pidtune
Cs = pidtune(Gs, 'PID');

% Create the closed-loop system
Gc = feedback(Cs*Gs, 1);

% Plot step response of the closed-loop system
step(Gc);
```

PID tuned Step response in time domain



To compare the actual plant model and MATLAB-based input-output data, we need to consider both the provided characteristics of the actual plant and the response obtained from MATLAB's PID tuning.

Actual Plant Model:

- Transfer function $= \frac{s+2}{0.1s^3+4s^2+8s+9}$
- Sample time = 0.0001s

Estimated transfer function:

- Transfer function $= \frac{10.03s+20.06}{s^3+40.12s^2+80.24s+90.27}$

Data Obtained from MATLAB PID Tuning:

- Rise time: 0.598 seconds
- Overshoot: 0.07 (7%)

- Settling time: 3.75 seconds
- Steady-state error: 0.001 (0.1%)

1. Rise Time:

- Rise time is the time taken for the system's response to rise from a specified lower value to a specified higher value for the first time.
- In this case, the rise time obtained from MATLAB tuning is 0.598 seconds. This means it takes approximately 0.598 seconds for the system's response to reach 90% of its final value for the first time after a step input.
- A smaller rise time indicates a faster response of the system.

2. Overshoot:

- Overshoot refers to the maximum percentage by which the response exceeds its final steady-state value during transient behavior.
- In this case, the overshoot obtained from MATLAB tuning is 0.07 (or 7%).
- A lower overshoot indicates less oscillation or ringing in the system's response. While a 1-% overshoot is typical for many systems, 7% might be considered slightly higher.

3. Settling Time:

- Settling time is the time required for the system's response to reach and stay within a specified error band around its final steady-state value.
- In this case, the settling time obtained from MATLAB tuning is 3.75 seconds.
- The settling time is considered good because it falls within the range of 1-5 seconds, indicating that the system stabilizes relatively quickly.

4. Steady-state Error:

- Steady-state error refers to the difference between the desired value and the actual value of the output as time approaches infinity.
- In this case, the steady-state error obtained from MATLAB tuning is 0.001 (or 0.1%).
- A smaller steady-state error indicates better accuracy of the system in tracking the desired reference value.

The settling time is within the desired range, the higher overshoot might indicate some overshooting or oscillation in the system's response. The rise time is relatively fast, indicating a quick initial response of the system.