Contents

1 System Identification **3**

1.1 Input and output data generation for a given plant 3

1.2 Estimate plant transfer function $G(s)$ using "System Identification App" 4

1.3 Estimated Model vs Actual Model 7

2 PID Controller Design **8**

List of Figures

1 Step response of system $G(s)$ with noise 4

2 System Identification Configuration 4

3 Confirm that generated data has been properly imported using time plot 5

4 Estimate Transfer Function Configuration 5

5 Estimated TF in work space 6

6 Completion of Identification 6

7 Estimated TF plot 6

8 Estimated Model Info 7

9 $G(s)$ and $\widehat{G}(s)$ in Time Domain 7

10 $G(s)$ and $\widehat{G}(s)$ Pole-Zero Plot 7

11 $G(s)$ and $\widehat{G}(s)$ Comparison 7

12 $G(s)$ and $\widehat{G}(s)$ DC Gain 8

13 PID Tuner GUI 8

14 Closed-loop step response with PID controller 9

15 Open-Loop vs Closed-Loop Step responses 9

16 PID Controller Parameters 10

17 Close-Loop Transfer Function for Controlled System 10

1 System Identification

1.1 Input and output data generation for a given plant

Following transfer function is used for this task:

$$G(s) = \frac{s+4}{s^3 + 4s^2 + 5s + 4}$$

Following MATLAB code is used for generate data,

```
clc; close all; clear;
s = tf('s');

G = (s + 4)/(s^3 + 4*s^2 + 5*s + 4);
dt =  0.0001;

t = 0:dt:8;
u = ones(length(t),1);
u(1:1/dt)=0;

% this sets the first 10 samples to zero
y = lsim(G,u,t);  % this generates the plant response for input u
y = y + rand(length(t),1)*0.02; % add random noise to the response

plot(t,[u,y]); axis([0 8 0 1.2]);
```

From this code we get y as output which is time domain step response of system $G(s)$ + added noise.

The generated step response,

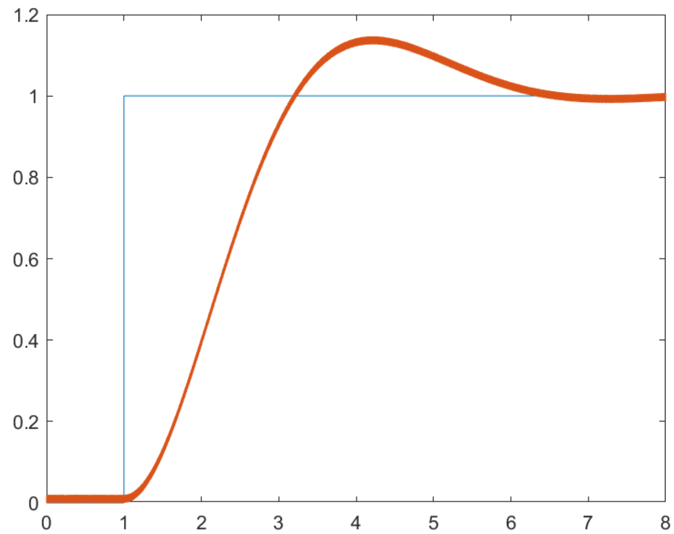

Figure 1: Step response of system $G(s)$ with noise

The simulation shows that the system settles around 1 with a nearly 0.2 overshoot.

1.2 Estimate plant transfer function $G(s)$ using "System Identification App"

Here we import input-output(u,y) data to System Identification App in MATLAB and Estimate transfer function for plant.

1. Opened the System Identification App:
   Command - **systemIdentification**
2. Imported the input ('u') and output ('y'),

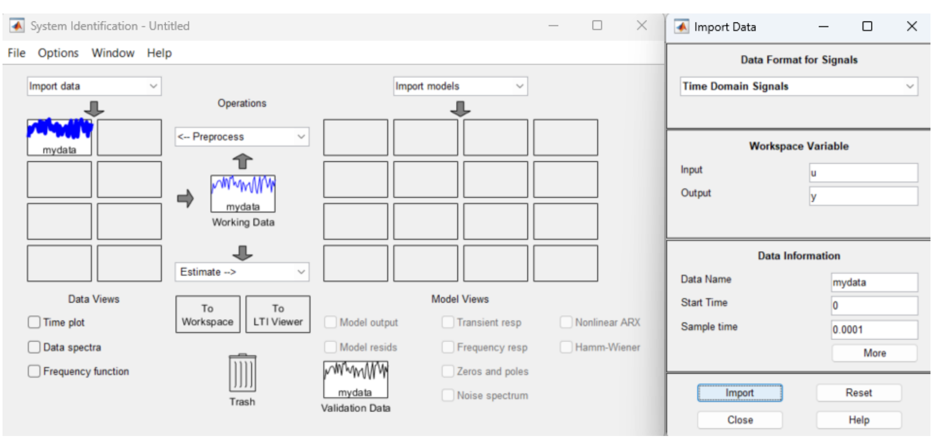

Figure 2: System Identification Configuration

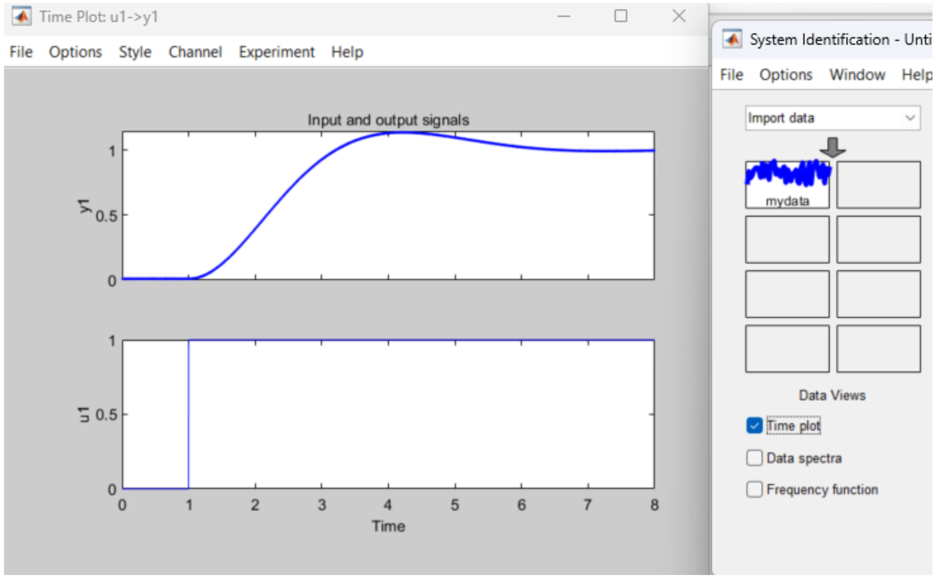

Figure 3: Confirm that generated data has been properly imported using time plot

3. Set the number of poles and the number of zeros.

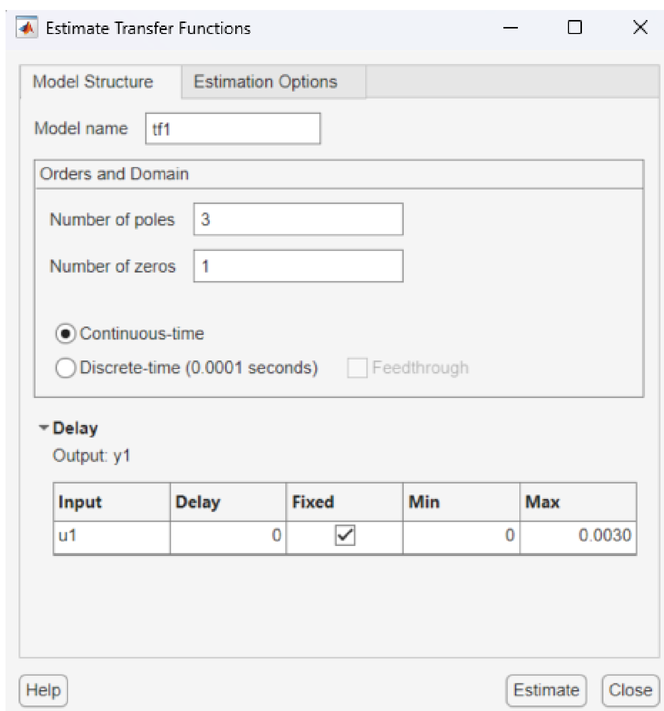

Figure 4: Estimate Transfer Function Configuration

4. Export estimated data to work space

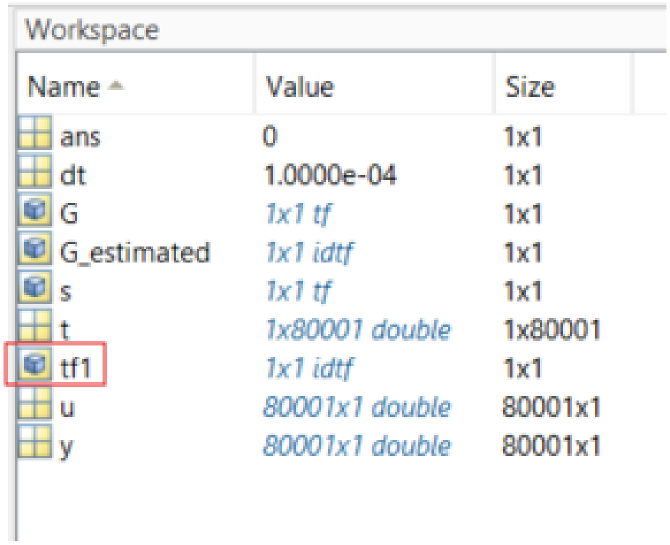

Figure 5: Estimated TF in work space

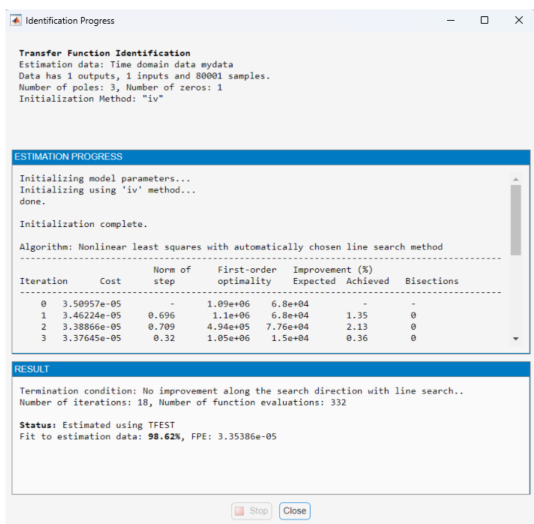

Figure 6: Completion of Identification

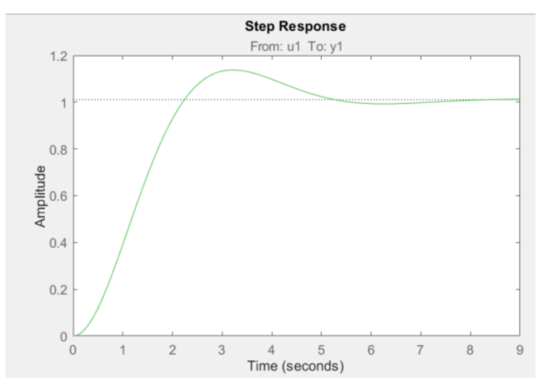

Figure 7: Estimated TF plot

1.3 Estimated Model vs Actual Model

Estimated model we got,

$$\widehat{G}(s) = \frac{1.071s + 3.428}{s^3 + 3.575s^2 + 5s + 3.393}$$

This model fit to estimation data **98.62%**

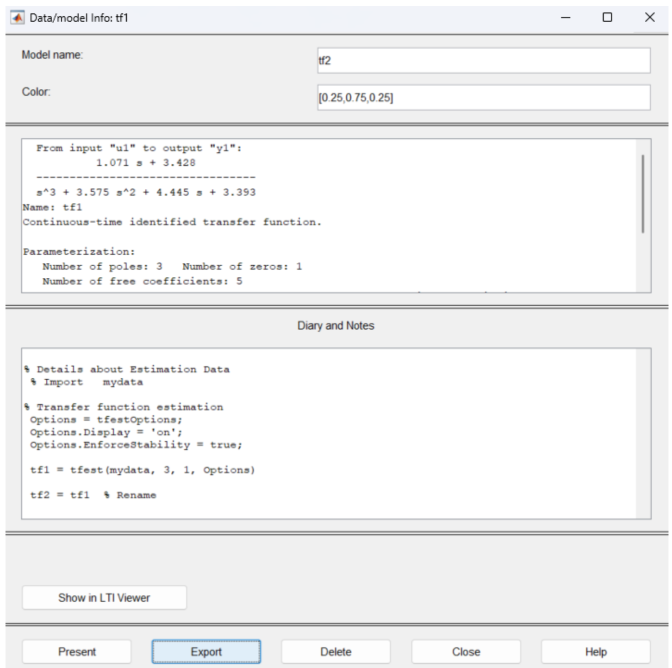

Figure 8: Estimated Model Info

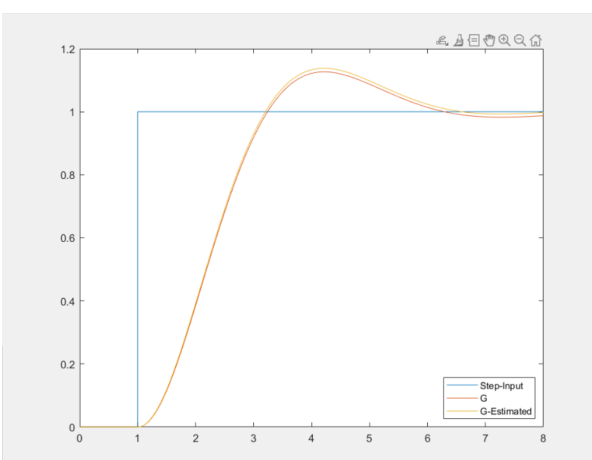

Figure 9: $G(s)$ and $\widehat{G}(s)$ in Time Domain

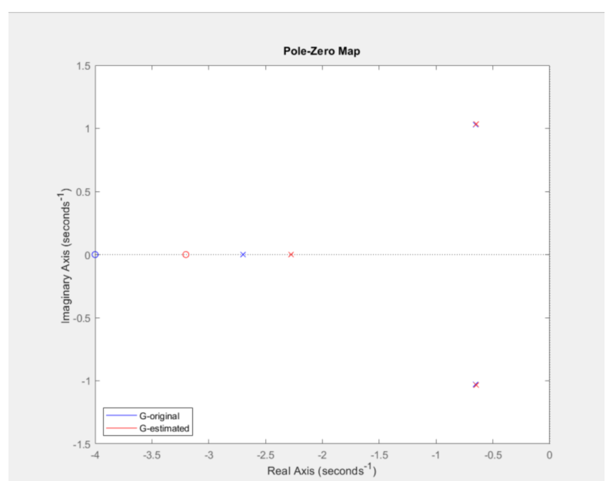

Figure 10: $G(s)$ and $\widehat{G}(s)$ Pole-Zero Plot

Figure 11: $G(s)$ and $\widehat{G}(s)$ Comparison

```
>> dcgain(G)

ans =

     1

>> dcgain(G_estimated)

ans =

    1.0104
```

Figure 12: $G(s)$ and $\widehat{G}(s)$ DC Gain

The estimated gain, poles and zeros closely match the actual values.From above details we can see that from Matlab we got accurate transfer function estimation.

2 PID Controller Design

To improve the system's performance a PID controller was designed using MATLAB's PID Tuner tool.The tuned controller is defined by the following parameters:

$$C(s) = K_p + \frac{K_i}{s} + K_d s$$

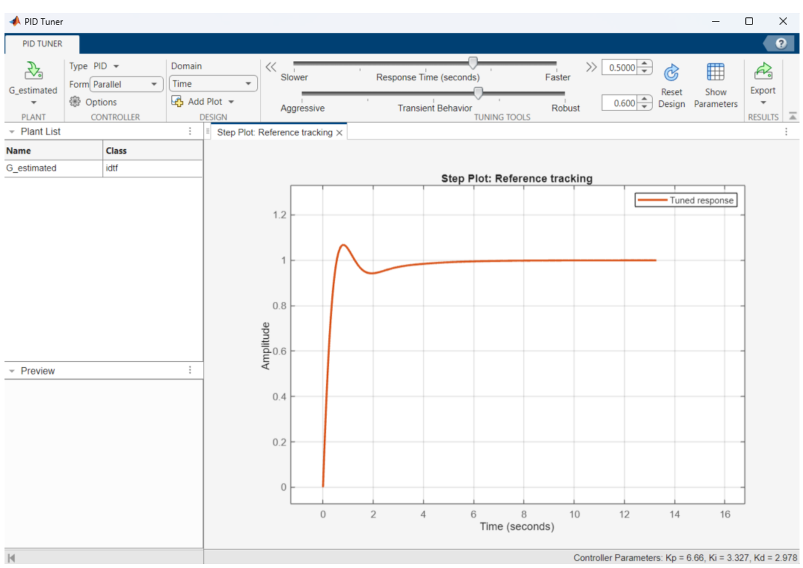

Figure 13: PID Tuner GUI

We can use GUI and set Performance Specifications as we want and get tuned PID values. (e.g., "settling time < 0.5 seconds, overshoot $< 10\%$"), input these in Tuning tab the PID Tuner.

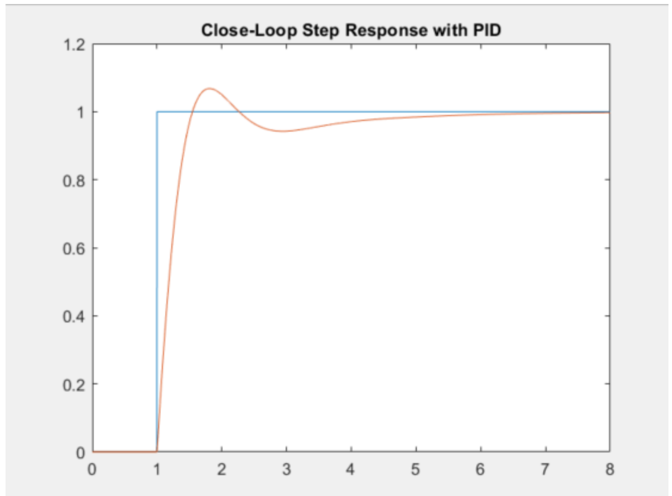

Figure 14: Closed-loop step response with PID controller

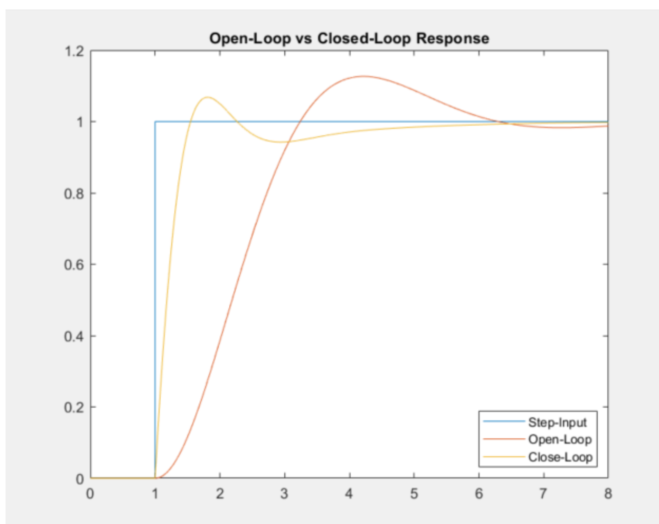

Figure 15: Open-Loop vs Closed-Loop Step responses

Controller Parameters

	Tuned
Kp	6.6596
Ki	3.3272
Kd	2.9783
Tf	n/a

Performance and Robustness

	Tuned
Rise time	0.394 seconds
Settling time	3.58 seconds
Overshoot	6.78 %
Peak	1.07
Gain margin	Inf dB @ NaN rad/s
Phase margin	69.7 deg @ 4 rad/s
Closed-loop stability	Stable

Close

Figure 16: PID Controller Parameters

```
T =

  From input "u1" to output:
     3.19 s^3 + 17.34 s^2 + 26.39 s + 11.41
  ---------------------------------------------
  s^4 + 6.765 s^3 + 21.79 s^2 + 29.79 s + 11.41

Continuous-time transfer function.
```

Figure 17: Close-Loop Transfer Function for Controlled System

The closed-loop response demonstrates significantly faster rise time and improved stability compared to the open-loop system. PID controller did those improvement to system. It also lower overshoot,reduced steady-state error