ECE 3413 Lab 04
Analysis of Linear Time Invariant (LTI) Systems

Leomar Durán

28^{th} February 2023

1 Introduction

The purpose of this lab is to introduce the trace tools in Simulink that can be used in combination with the scope. It also introduces the concept of file input and output of time series in Matlab and Simulink. Generally, this experiment makes much more use of Simulink than we have so far.

Here we learn to expand on automating the process of finding transfer function characteristics from the step response that we have been working with in class.

2 Procedure

2.1 Part 01 – Standard signal responses

2.1.1 Step 01.01 – The simulation

The Parameters In part 1 we are using standard input functions and applying the transfer function

$$G = \frac{2}{s^2 + 5s + 9}$$

to these.

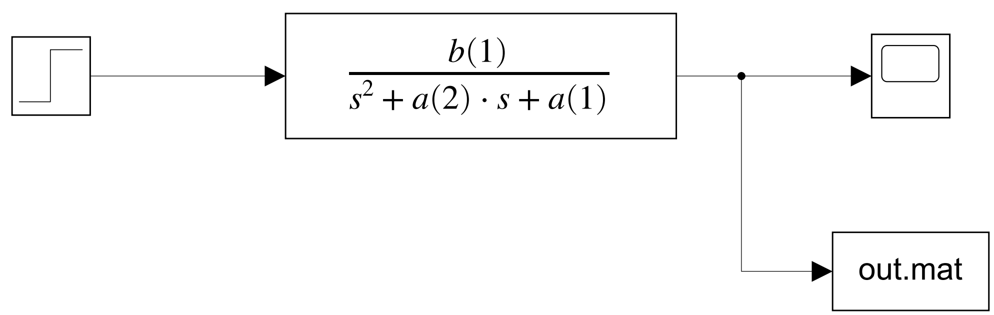
I have written a Matlab script to parameterize the model so that it is more flexible and reusable. This script is available in Appendix subsection A.1.

It sets up the simulation time `TSTOP = 10.0` in s, the parameters for the input function and the coefficients and constants for the transfer function. In Part 01a, this input function is a step function which becomes `stepFinal = 1` in V at time `Tstep = 0`. In Part 01b, this input function is a ramp function $r(t)$ whose slope becomes 1 V at time `Tstep = 0` until $r(t) =$ `stepFinal = 1`.

This script is also reused in Part 02 since it already contains much of the input needed for part 02 (namely, the simulation time and transfer function.

01(a) The step response model

part01a_step_response_slx



C:\simulink\lab04\part01a_step_response_slx.slx

printed 01-Mar-2023 21:17

In Part 01(a), we are modeling a step response. Thus, our input signal is a Heaviside step function as built in the model on page 3. The only unique block in this model, the Step block is configured as in Fig. 1.

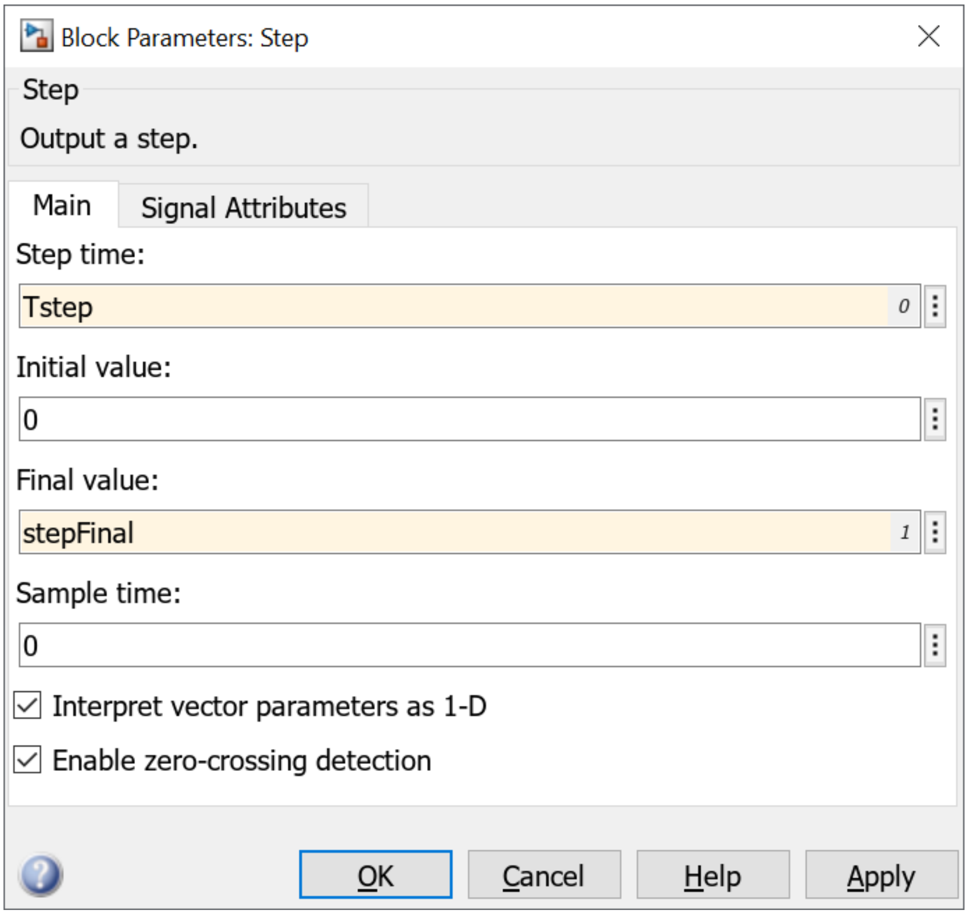


Figure 1: The configuration for step.

Common features The following features are common to all models simulated in this lab. All parameters are defined in Appendix subsection A.1.

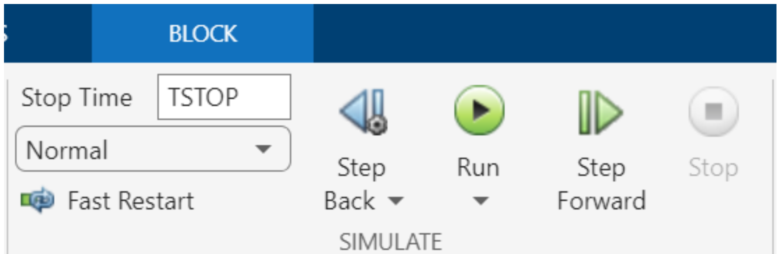


Figure 2: The simulation time used in this equation is `TSTOP`.

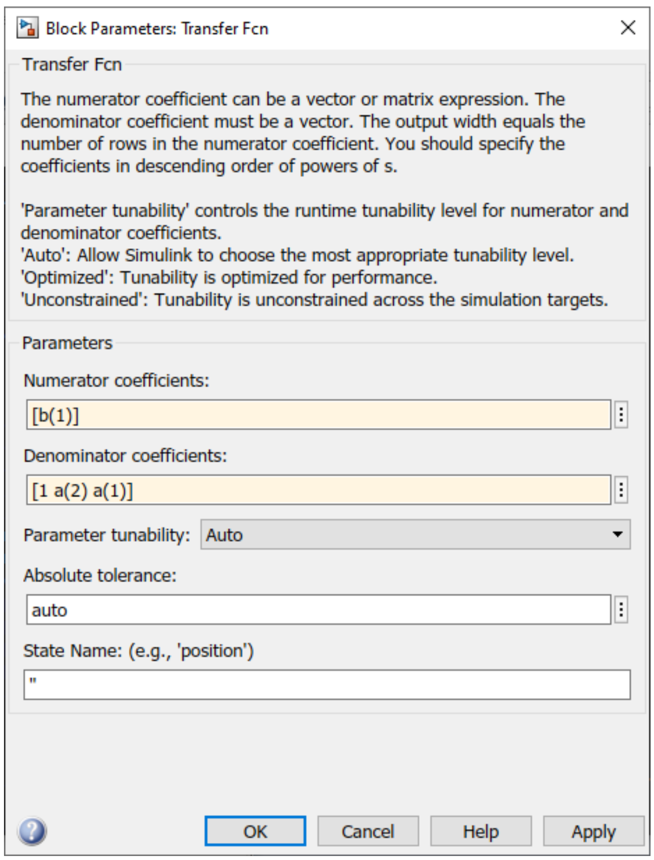


Figure 3: The transfer function is $\frac{b_1}{s^2+a_2 s+a_1}$. (Note that Matlab uses 1-indexed arrays.)

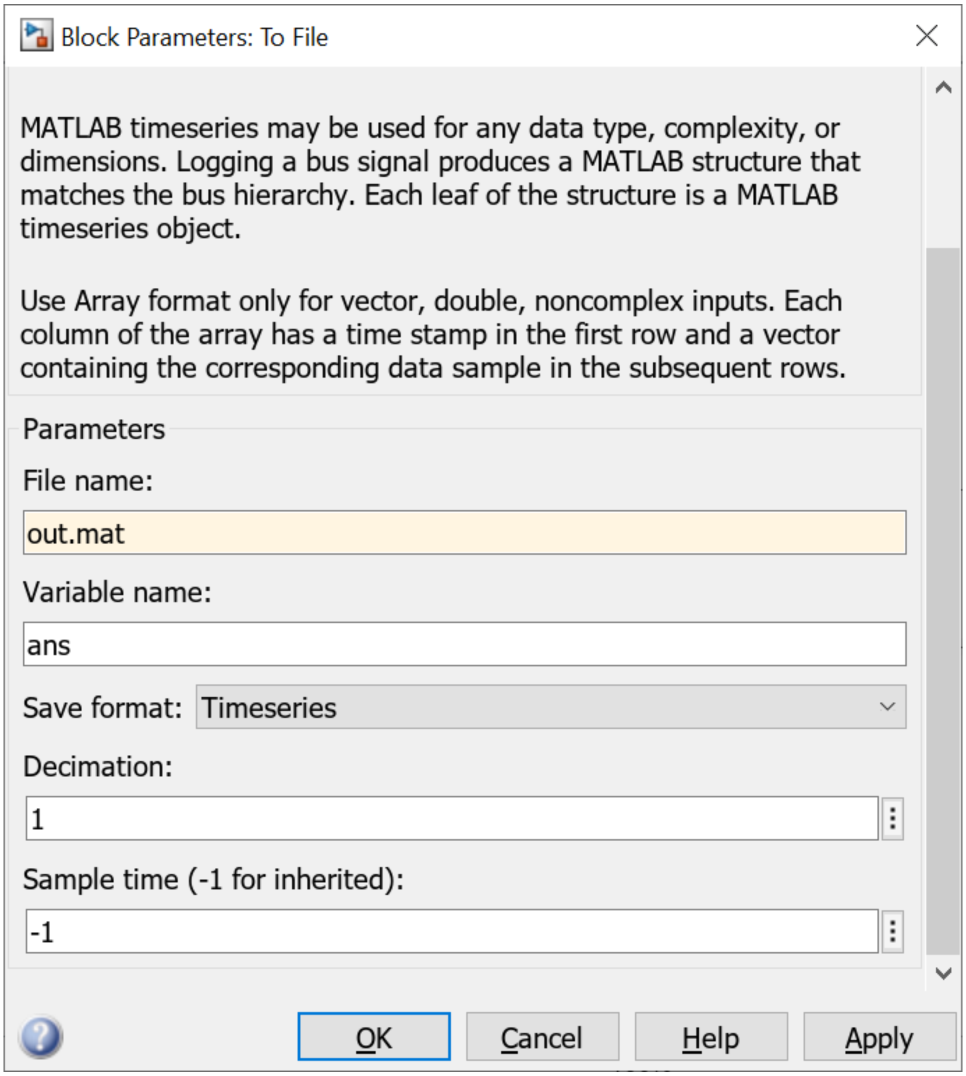
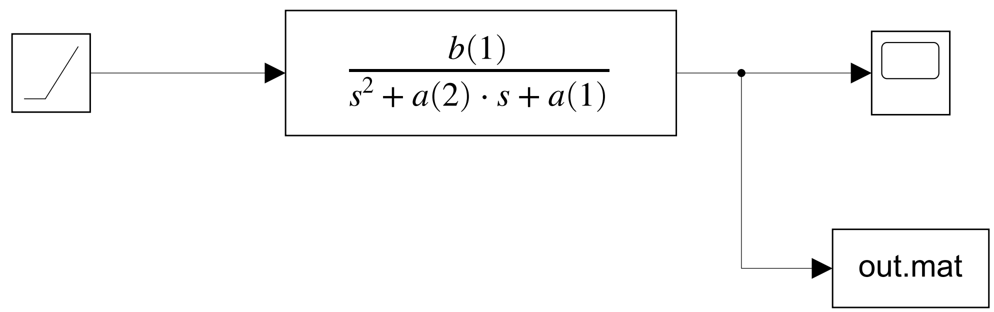


Figure 4: All models will output the time series data of the response to the file `out.mat` to the default variable `ans`.

01(b) The ramp response model We use the same features described in paragraph 2.1.1. However, the difference is that we are modelling the ramp response seen in the model on page 7. with the ramp parameters in Fig. 5.

part01b_ramp_response_slx



C:\simulink\lab04\part01b_ramp_response_slx.slx

printed 01-Mar-2023 21:21

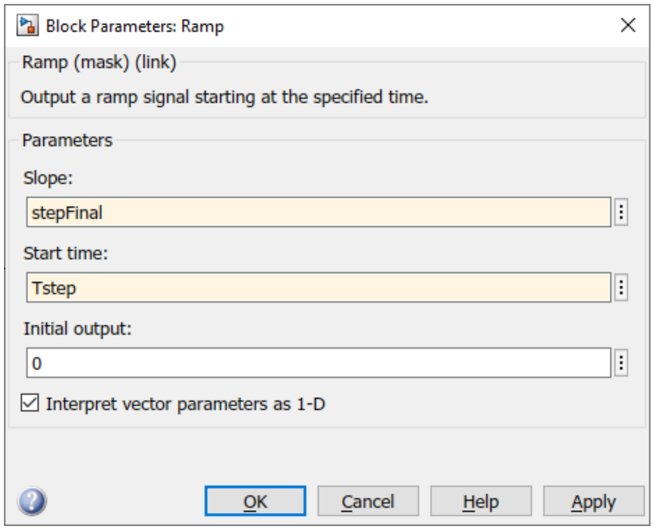


Figure 5: The configuration for ramp.

2.1.2 Step 01.02 – Measuring the traces on step response

As before, we find the maximum

The peak is found by finding the maximum output value of the plot. We find the peak $c(T_p) = 2.242 \times 10^{-1}$ in Fig. 6.

The peak time is the corresponding time, which we can find in the same plot. The peak time $T_p \in [1.879, 1.904]$s. Let's take $T_p = \frac{1}{2}(1.879 + 1.904)\text{s} = 1.8915\,\text{s}$.

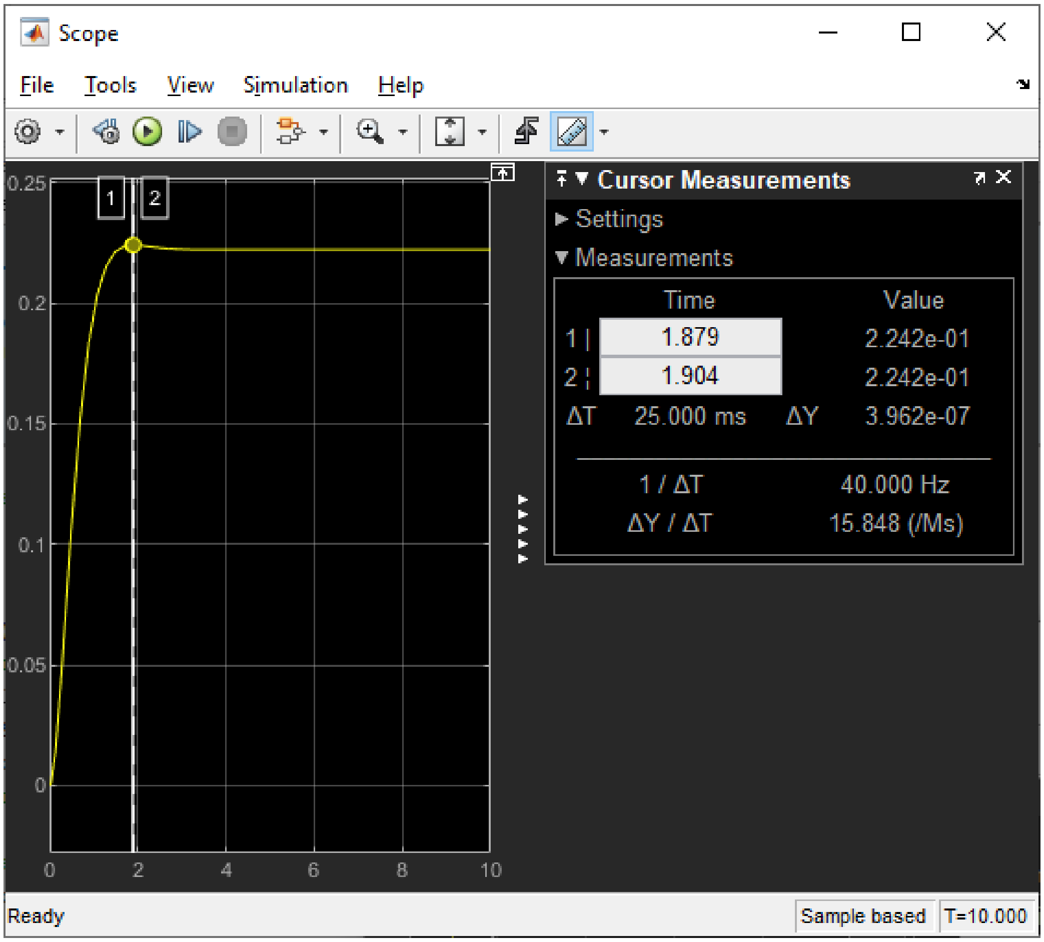


Figure 6: Measuring the peak and peak time.

The percent overshoot can be measured using Fig. 7. This is because cursor 1 is at the peak time and cursor 2 is on the final time. So the percent overshoot is the value $\frac{\Delta Y}{c(T_p)} = \frac{1.929 \times 10^{-3}}{2.242 \times 10^{-1}} \times 100\,\% = 0.8604\,\%$.

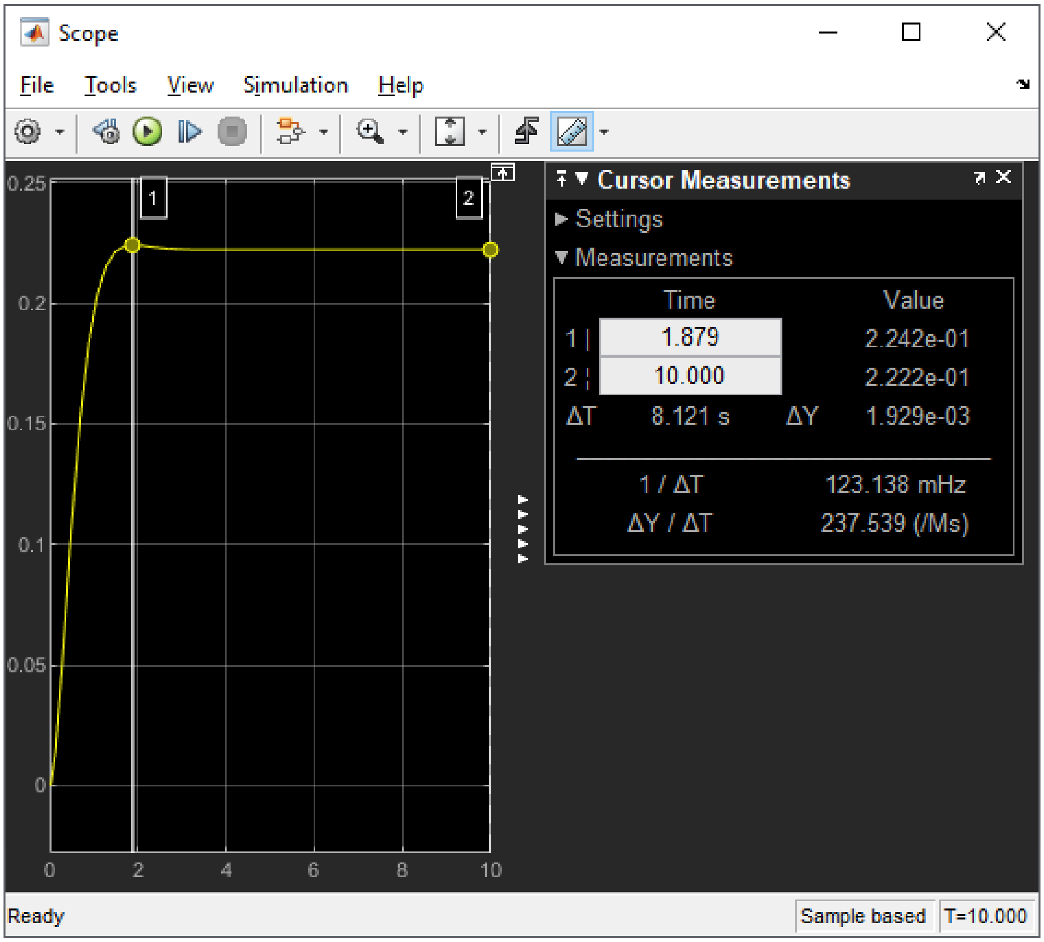


Figure 7: Measuring the percent overshoot, final value and steady state error.

The stead state error We can also see the final value $c_f = 2.222 \times 10^{-1}$ in this figure, and from that, we can calculate the steady state error. We expect a final value equal to `stepFinal = 1`. So $E_{ss} = 1 - 2.222 \times 10^{-1} = 0.7778$.

The rise time To measure the rise time, we must first find each of the first times when the output value is 10 % and 90 % of the final value.

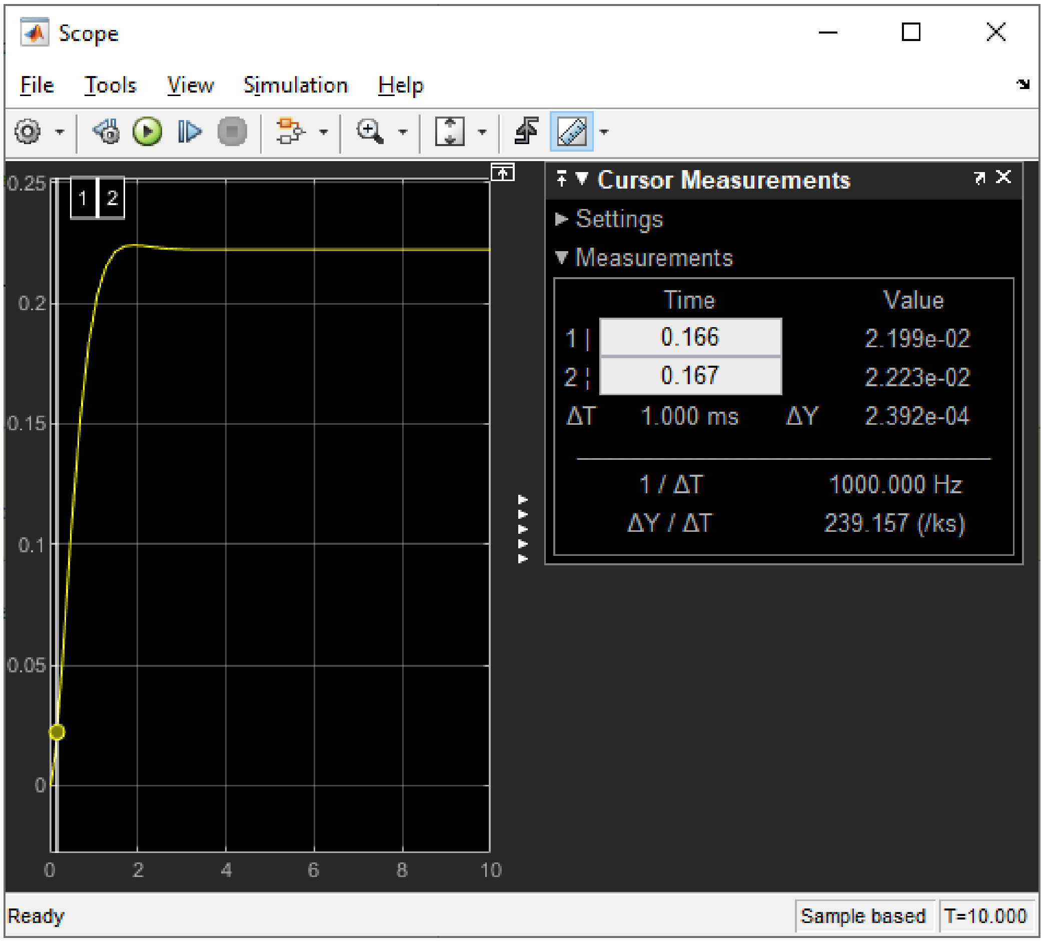


Figure 8: Estimating 10 %of final output.

Well, 10 % of the final value $10\,\% c_f = 10\,\%(2.222 \times 10^{-1}) = 2.222 \times 10^{-2}$. In Fig. 8, we estimate the time when $c = 2.222 \times 10^{-2}$ to be

$$t_{.10} = \left(0.166 + \frac{1 \times 10^{-3}}{2.392 \times 10^{-2}}\left(2.222 \times 10^{-2} - 2.199 \times 10^{-2}\right)\right)\mathrm{s} = 0.166\,01\,\mathrm{s}.$$

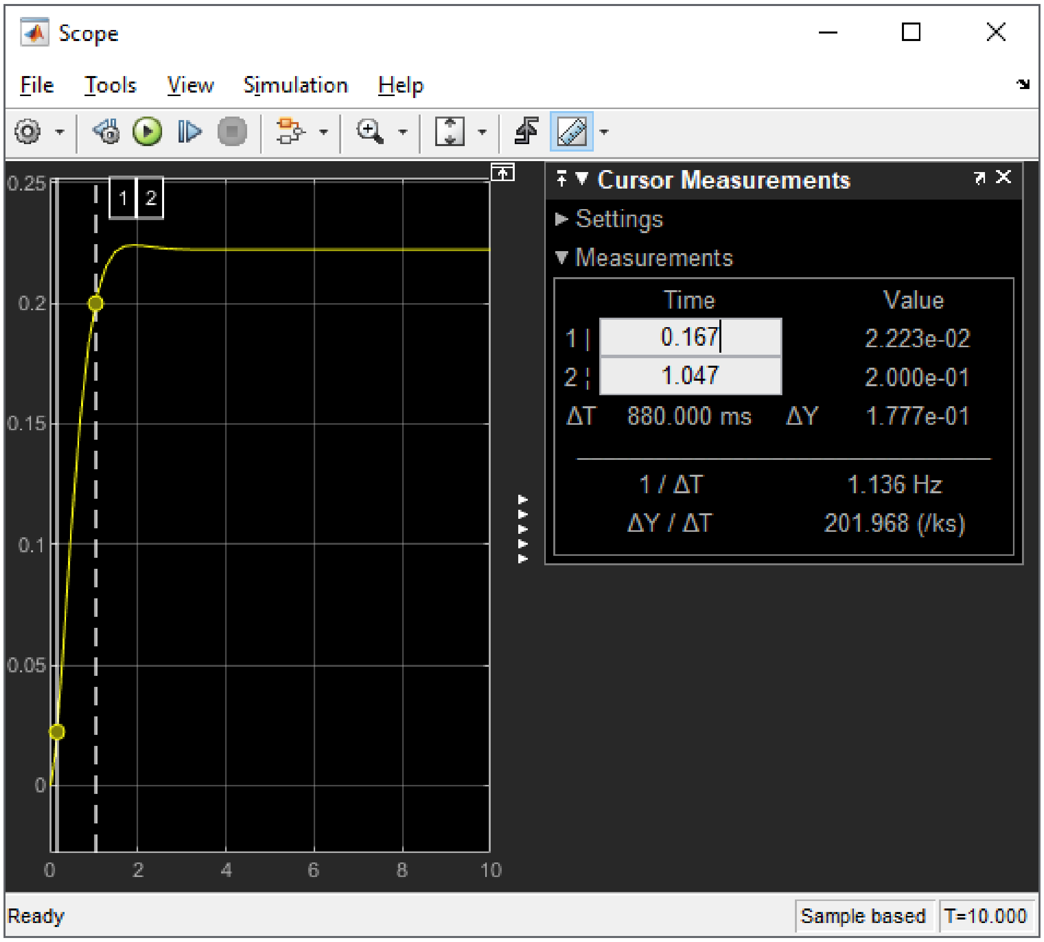


Figure 9: Measuring the rise time from 90 % and an estimate of 10 % of final output.

Now 90 % of the final value $90\,\% c_f = 0.0200$. In Fig. 9, we already have an exact measure of the time when $c = 0.0200$. So let's just use that time $t_{.90} = 1.047$. The time at cursor 1 represents the closest time to 3 decimal places.

The rise time $T_r = t_{.90} - t_{.10} = 1.047 - 0.16601\text{s} = 0.88099\,\text{s}$.

The settling time is when the output reaches a value within 5 % of its final value. Now, the signal can never supersede the maximum value of the signal, the peak $c(T_p) = 2.242 \times 10^{-1}$.

However, $c_f + 5\,\% = 2.222 \times 10^{-2}(1 + .05) + 2.333 \times 10^{-2}$. So we look for the last value where $c = c_f - 5\,\% = 2.111 \times 10^{-2}$. From Fig. 10, we see that this occurs at $T_s = 1.208\,\mathrm{s}$.

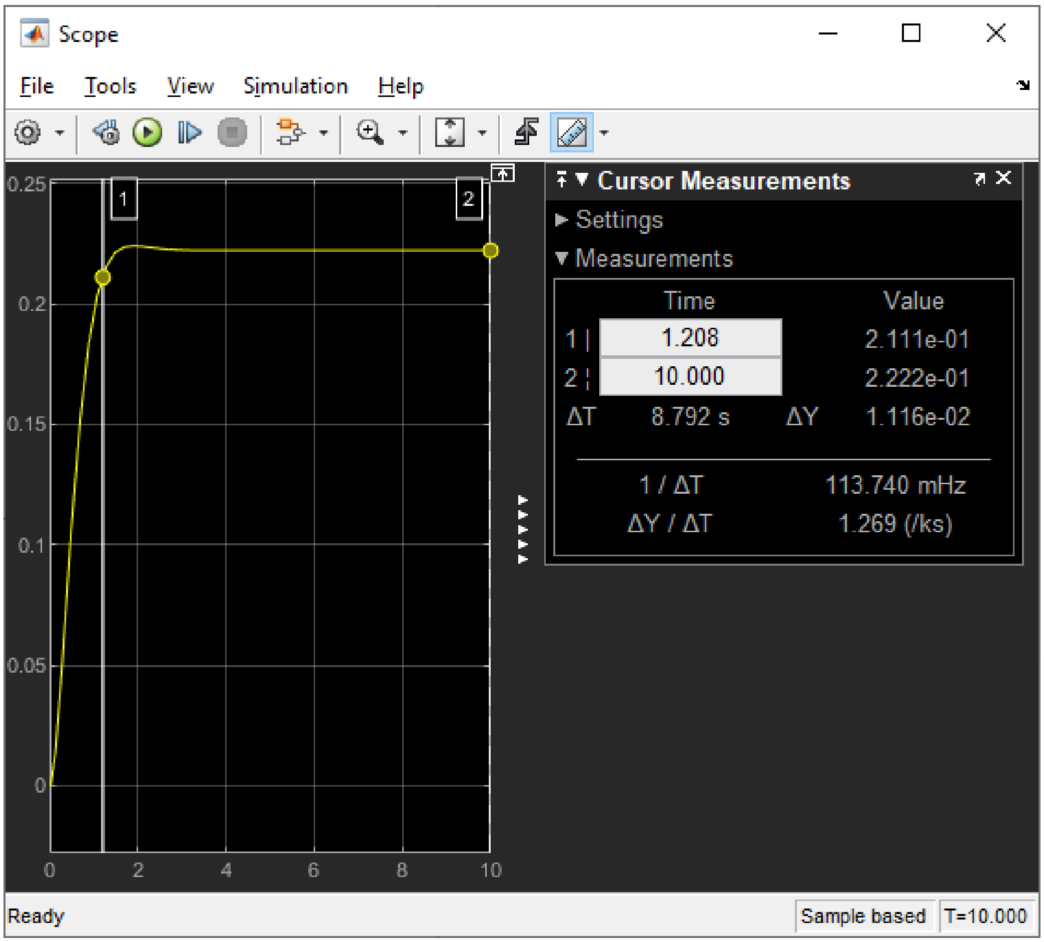


Figure 10: Measuring the settling time.

2.1.3 Step 02 - Measuring the traces on ramp response

The peak As before, we find the maximum output value of the plot. We see in Fig. 11, that the final value $c_f = 2.099$.

The peak time The peak occurs at time $T_p = 10.000\,\mathrm{s}$.

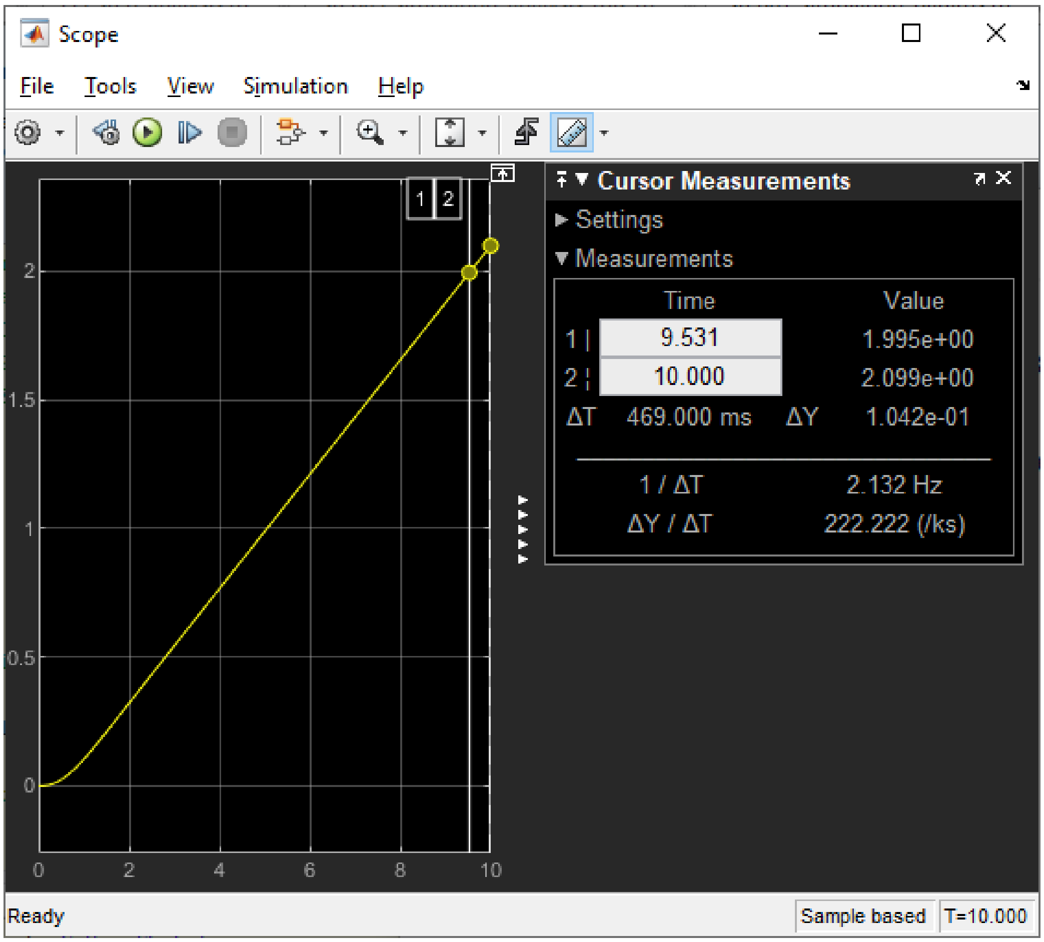


Figure 11: Measuring the peak, peak time, the percent overshoot, final value, steady state error, and the settling time.

The percent overshoot In the case of the ramp response to the G, the maximum value is the final value. Thus, the percent overshoot is $\frac{0}{2.099e+0} \times$

$100\,\% = 0.000\,\%$.

The stead state error is $1 - c_f = 1 - 2.099 = -1.0990$.

The rise time is the time it takes for the signal to rise from $10\,\%$, $t_{.10} = 1.507\,\mathrm{s}$ to $90\,\%$, $t_{.90} = 9.0585\,\mathrm{s}$, that is $T_r = 9.0585\,\mathrm{s} - 1.507\,\mathrm{s} = 7.5515\,\mathrm{s}$.

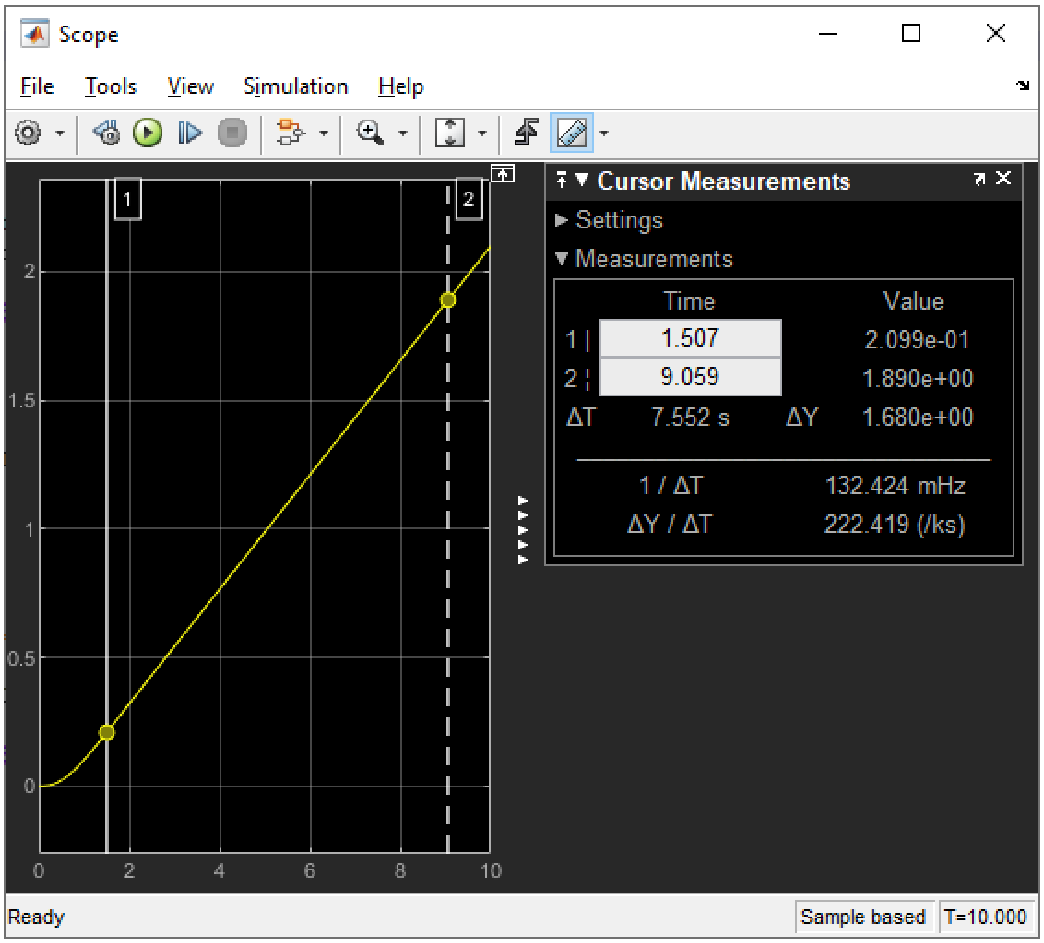


Figure 12: Measuring the rise time from $90\,\%$ and an estimate of $10\,\%$ of final output.

The settling time can be see back in Fig. 11. The ramp response is within 5 % of the final value at time $T_s = 9.5310\,\mathrm{s}$.

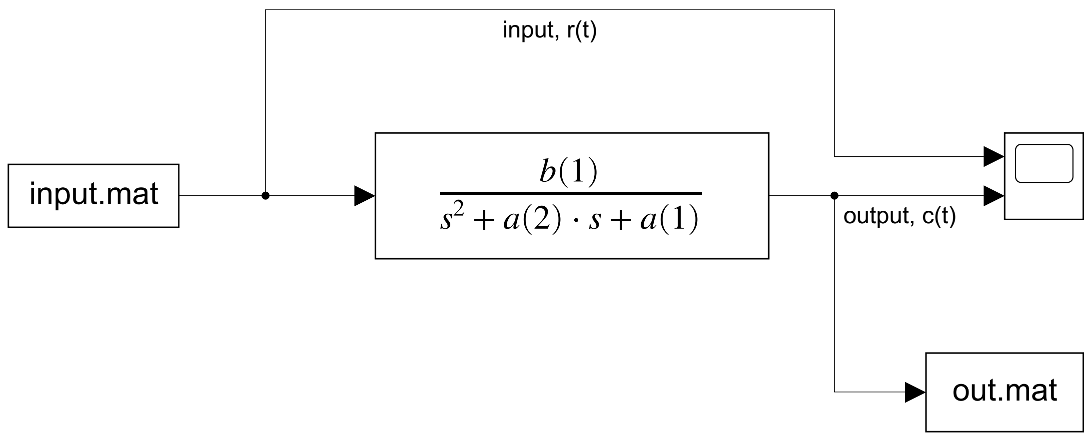
2.1.4 Step 01.03 – Simulation analysis using Matlab

To perform the analysis of the step and ramp response, I have written the Matlab Live Script available in Appendix subsection A.2.

2.2 Part 02 – Writing in Matlab/Reading in Simulink

We can write a signal in Matlab using a script such as that available in Appendix subsection A.3 To analyze the effect of the transfer function on the generated sinusoidal function, we use the model on page 17.

17



3 Results

3.1 Part 01 – Standard signal responses

Following the procedure, we have found the values in Table 1.

Table 1: The characteristics of the transfer function's responses calculated using traces.

Response	T_p[s]	$c(T_p)$	$\%OS$	E_{ss}	T_r[s]	T_s[s]
Step	1.8915	0.2242	0.8604 %	0.7778	0.880 99	1.208
Ramp	10.000	2.099	0.000 %	−1.0990	7.5515	9.5310

3.1.1 Step 03 – Analyses by Matlab

Here we see the results of the analyses by Matlab using the Matlab live script. We compare to the values that we found by using the traces using the percent difference formula

$$\% \text{ difference} = \frac{\Delta x}{\bar{x}} \times 100\,\% \tag{1}$$

Simulation analysis – Step response

Read the data file and load its data

```
TSTOP = 10

Tstep = 0

stepFinal = 1
```

```
data =
    ans: [1x1 timeseries]
```

Part 01-03 perform step analysis

We calculate the analysis using these intermediate values.

```
pc10Idx = 8

pc90Idx = 12

TsIdx = 12
```

The results are

```
stepCharacteristics =
    peak: 0.2242
    pcOS: 0.8771
      Tr: 0.7997
      Tp: 1.8865
      Ts: 1.0865
     Ess: 0.7778
```

Plot the steady state error, followed by the plot with characteristics traced.

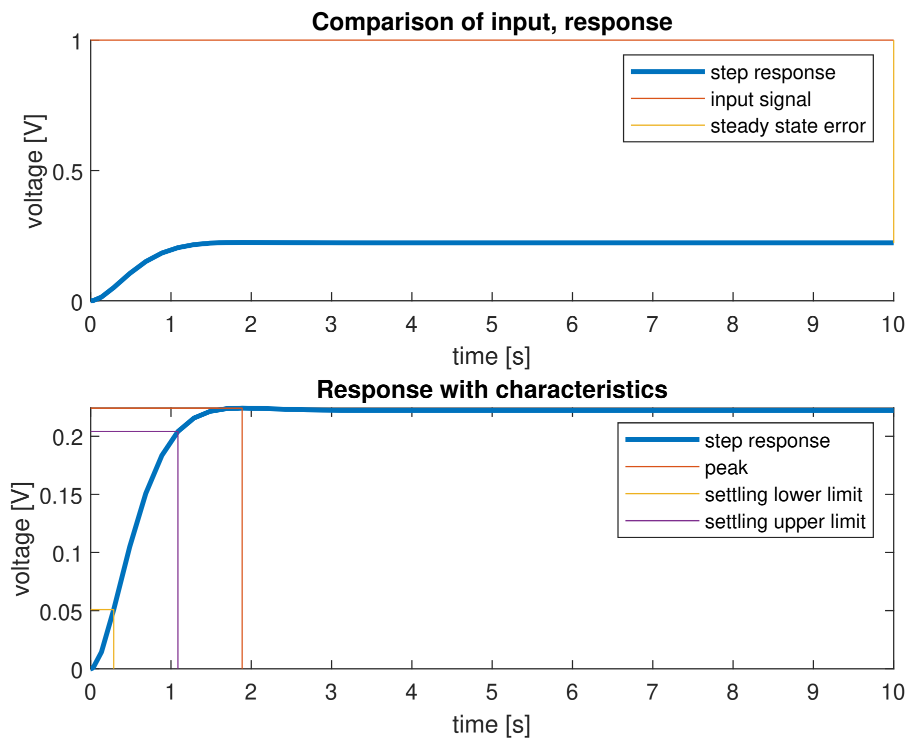


Table 2: Comparison of step responses: by traces and by Matlab.

Analysis	T_p[s]	$c(T_p)$	$\%OS$	E_{ss}	T_r[s]	T_s[s]
Trace	1.8915	0.2242	0.8604 %	0.7778	0.880 99	1.208
Matlab	1.8865	0.2242	0.8771 %	0.7778	0.7997	1.0865
% difference	0.2647 %	0.0000 %	1.9223 %	0.0000 %	9.6734 %	10.5905 %

Simulation analysis – Ramp response

Read the data file and load its data

```
TSTOP = 10

Tstep = 0

stepFinal = 1

data =
    ans: [1x1 timeseries]
```

Part 01-03 perform step analysis

We calculate the analysis using these intermediate values.

```
pc10Idx = 11

pc90Idx = 49

TsIdx = 50
```

The results are

```
stepCharacteristics =
    peak: 2.0988
    pcOS: 0
      Tr: 7.6000
      Tp: 10
      Ts: 9.3571
     Ess: -1.0988
```

Plot the steady state error, followed by the plot with characteristics traced.

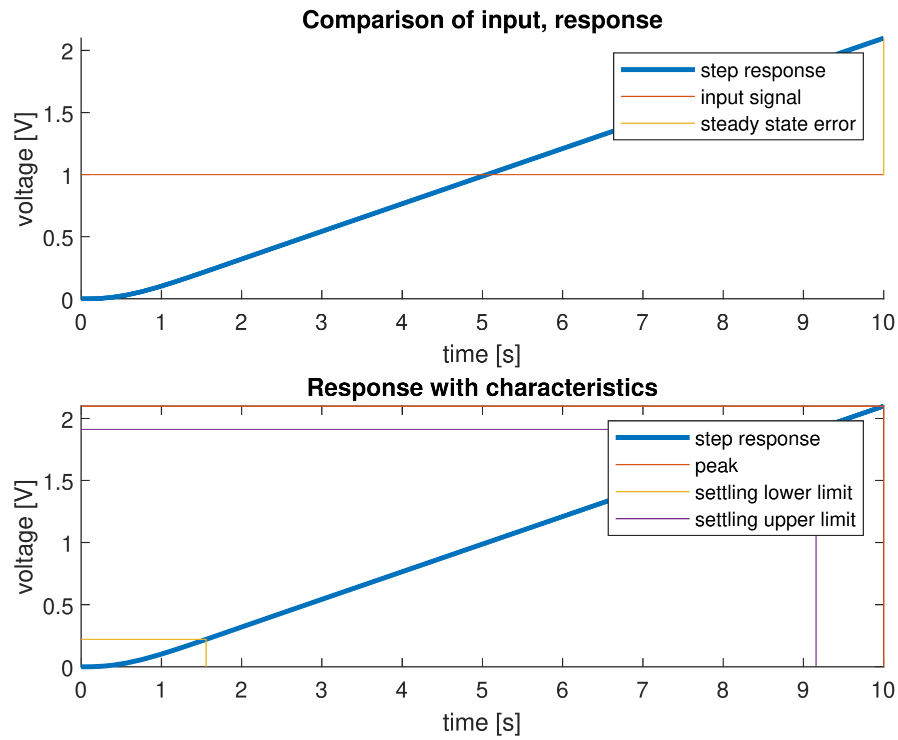


Table 3: Comparison of ramp responses: by traces and by Matlab.

Analysis	T_p[s]	$c(T_p)$	$\%OS$	E_{ss}	T_r[s]	T_s[s]
Trace	10.000	2.099	0.000 %	−1.0990	7.5515	9.5310
Matlab	10	2.0988	0 %	−1.0988	7.6000	9.3571
% difference	0 %	9.5288×10^{-3} %	0 %	1.8200×10^{-2} %	0.6402 %	1.8414 %

Finally, let's compare the two methods by inspection.

Table 4: Comparison of percent differences between step response and ramp response.

Response	T_p[s]	$c(T_p)$	$\%OS$	E_{ss}	T_r[s]	T_s[s]
Step	0.2647 %	0.0000 %	1.9223 %	0.0000 %	9.6734 %	10.5905 %
Ramp	0 %	9.5288×10^{-3} %	0 %	1.8200×10^{-2} %	0.6402 %	1.8414 %

Looking at Table 4, although the percentage differences are not bad because we are estimating using a cursor, which leaves room for human error, I am surprised to see that the ramp analysis is much closer and acceptable. The settling time seems to be the hardest characteristic to approximate by using traces giving 1.8414 % even for the ramp response.

3.2 Part 02 – Writing in Matlab/Reading in Simulink

For this part of the experiment, we wrote a sinusoidal wave to a file using Matlab and read it as an input signal for a Simulink model. My expectation is that since the sinusoidal wave is an oscillating signal with an indeterminate limit as time approaches infinity, that we will get an unstable output that oscillates and has an indeterminate limit as time approaches infinity.

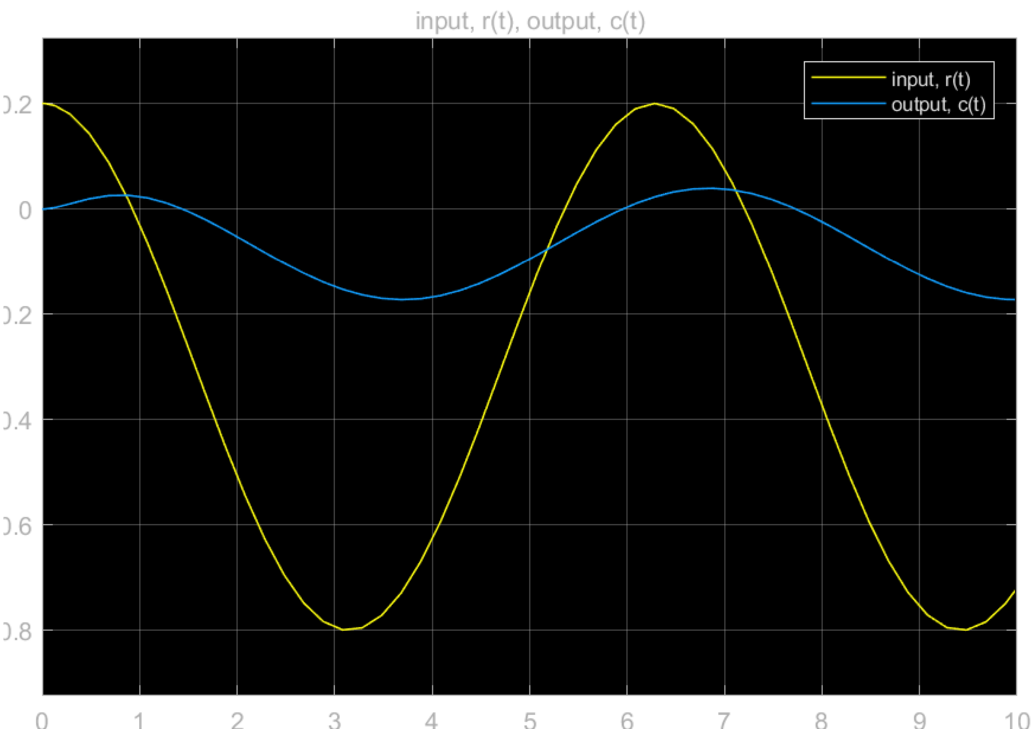


Figure 13: Comparison of the input signal and the output signal from performing the transfer function thereon.

From Fig. 13, we see that we do indeed get an oscillating output signal, which is very similar to a sine wave.

Simulation analysis – Response to Sinusoidal from file

Read the data file and load its data

```
TSTOP = 10

Tstep = 0

stepFinal = 1
```

```
data =
    ans: [1x1 timeseries]
```

Part 01-03 perform step analysis

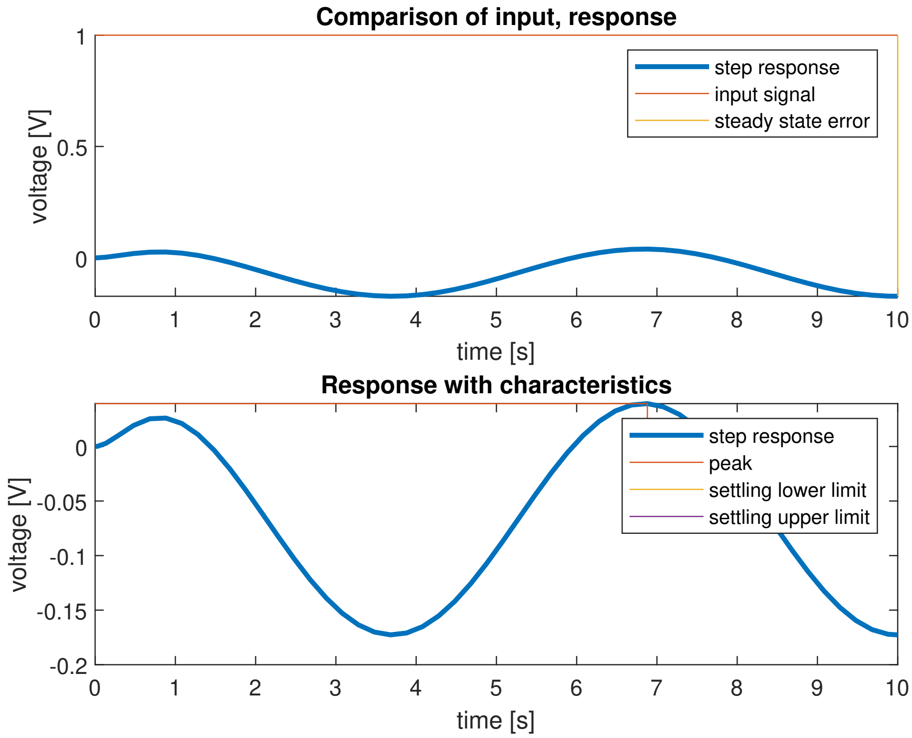
We calculate the analysis using these intermediate values.

```
pc10Idx = 1

pc90Idx = 1

TsIdx = 55
```

The results are

```
stepCharacteristics =
    peak: 0.0393
    pcOS: -122.7348
      Tr: 0
      Tp: 6.8808
      Ts: 10
     Ess: 1.1727
```

Plot the steady state error, followed by the plot with characteristics traced.



In the preceding Matlab live script, we saw that the Simulation analysis script (Appendix subsection A.2) attempted to find all of the characteristics of the sinusoidal response.

In this case, the peak was exactly as expected (the maximum value). Peak time refered to the corresponding first time that the sinusoidal response reached its maximum during the time of the simulation.

However, the values of the percent overshoot, the rising time and steady state error made no sense because there is no final value to the signal. This is because the limit of a sinusoidal function as time approaches infinite is indeterminate.

We can see that the settling time is near or at the entire time of the simulation because the sinusoidal response will never settle.

4 Discussion

This experiment showed that the process of finding a signal's characteristics through using the trace function can be found satisfactorily. It would also lead to less human error.

This experiment shows that the settling time may not be a good characteristic to use to identify a transfer function.

The characteristics that we found using the ramp response in Table 1 were very different from those we found using step response. I am not completely sure how the ramp response would be used. Overall, the step response seems to be the best one to use to identify a transfer function as compared to both the ramp response and the sinusoidal as seen in subsection 3.2.

Finally, we can predict the oscillatory nature of an output signal by knowing whether any of its components also oscillate.

A Appendix

A.1 Step 01 (in parts 01 and 02) – Simulation parameters, Matlab Script

```
%% step01_simulation_params.m
% Sets the parameters for simulating a second order transfer function.
% By      : Leomar Duran
% When    : 2023-02-28t18:53
% For     : ECE 3413 Classical Control Systems
%

clear

% simulation parameters
TSTOP = 10.0    % [s]

% step function parameters
Tstep = 0       % [s]
stepFinal = 1   % [V]

% transfer function parameters
B = 2;
A = [1 5 9];
G = tf(B, A)
% indices expected in reverse order and normalized to A(1)
b = B(end:-1:1)/A(1)
% A(1) is thus unneeded
a = A(end:-1:2)/A(1)
```

A.2 Part 01 Step 03 – Simulation analysis, Matlab Live Script

```
%% Simulation analysis

% step03_simulation_analysis_mlx.m
```

```matlab
% Uses step analysis to characterize the simulation of the system.
% By       : Leomar Duran
% When     : 2023-02-28t18:49
% For      : ECE 3413 Classical Control Systems
%

%% Read the data file and load its data

% constants for the script
SINK_FILE = 'out.mat';

% simulation parameters
TSTOP = 10.0    % [s]

% step function parameters
Tstep = 0       % [s]
stepFinal = 1   % [V]

% read the data from the sink file
data = load(SINK_FILE)
% load the step response data and time
c = data.ans.Data;
t = data.ans.Time;

%% Part 01-03 perform step analysis
% We calculate the analysis using these intermediate values.

[peak, peakIdx] = max(c);
% percent overshoot = (peak value - final value)/(final value) * 100%
pcOS = (peak - c(end))/c(end)* 100;
% rise time = time for output to go from 10% to 90% of the final value
pc10Idx = find(c >= .10*c(end), 1)
pc90Idx = find(c >= .90*c(end), 1)
Tr = t(pc90Idx) - t(pc10Idx);
% peak time = time for the output to reach its maximum value
Tp = t(peakIdx);
% settling time = time for output to be bound within 5% of its final
%                 value
```

```matlab
TsIdx = find(abs(c - c(end)) >= 0.05*c(end), 1, 'last')
Ts = t(TsIdx);
% steady state error
Ess = (stepFinal - c(end));

%%
% The results are
stepCharacteristics = struct('peak', peak, 'pcOS', pcOS, ...
    'Tr', Tr, 'Tp', Tp, 'Ts', Ts, 'Ess', Ess)

%%
% Plot the steady state error, followed by the plot with
% characteristics traced.
subplot(2,1,1)
hold on
plot(data.ans, 'LineWidth', 2)
plot(t([1 end]), [1 1]*stepFinal)
plot([1 1]*t(end), c(end) + [0 Ess])
hold off
title('Comparison of input, response')
xlabel('time [s]')
ylabel('voltage [V]')
legend('step response', 'input signal', ...
    'steady state error')
%

subplot(2,1,2)
plot(data.ans, 'LineWidth', 2)
hold on
plot([1 1 0]*Tp, [0 1 1]*peak)
plot([1 1 0]*t(pc10Idx), [0 1 1]*c(pc10Idx))
plot([1 1 0]*t(pc90Idx), [0 1 1]*c(pc90Idx))
hold off
title('Response with characteristics')
xlabel('time [s]')
ylabel('voltage [V]')
legend('step response', 'peak', ...
    'settling lower limit', 'settling upper limit')
```

```
%
```

A.3 Part 02 – Saving a sinusoidal function, Matlab function

```
function part02_save_sinusoid()
%% part02_save_sinusoid.m
% Creates and saves a sinusoidal function for simulation
% By      : Leomar Duran
% When    : 2023-02-01t21:38
% For     : ECE 3413 Classical Control Systems
%
    t = linspace(0,10,1000);
    y = 0.5*cos(t) - 0.3;
    y_ts = timeseries(y, t)

    save input.mat -v7.3 y_ts
end % function part02_save_sinusoid()
```