CALIFORNIA STATE UNIVERSITY
LONG BEACH

Control Systems Laboratory
EE 370L
Fall 2019
v1.21

Lab #3

Basics of Control Systems in MATLAB and Simulink

prepared by
Hossein Jula Ph.D.
Arya Daroui

1. Introduction

MATLAB and Simulink are critical pieces of software used for modeling in both academia and industry—especially in control systems. In this lab, we will explore important control system tools MATLAB and Simulink offer, and leverage them to gain a better understanding of phenomena in control systems.

2. Procedure

2.1 MATLAB practice

In terms of control systems, a polynomial function can be represented in two equivalent ways: its factored form and its expanded (polynomial) form. The factored form gives insight into the roots of the function, and the expanded form gives insight into the coefficients and degree of the polynomial. This is of interest because we deal with transfer functions in the form of

$$T(s) = k\frac{Z_{\circ}(s)}{P_{\times}(s)} \tag{1}$$

where $Z_{\circ}(s)$ and $P_{\times}(s)$ are polynomials with some gain k. The roots (zeros and poles), coefficients, and degree of these polynomials give important analytical information about their behavior and applicability. Accordingly, the MATLAB functions `roots()`, `poly()`, `tf()`, and `zpk()` help create and analyze transfer functions with these different polynomial forms. Use them in the following subsections where necessary.

2.1.1 Roots and polynomial operations

Use `roots()` to find the roots of the following expanded polynomials:

$$\begin{aligned}
E_1(s) &= 3s^6 + 7s^5 + 2s^4 + 9s^3 + 20s^2 + 6s + 10 \\
E_2(s) &= 3s^6 + 6s^5 + 1s^4 + 8s^3 + 19s^2 + 5s + 9 \\
E_3(s) &= E_1 + E_2 \\
E_4(s) &= E_1 - E_2 \\
E_5(s) &= E_1 E_2
\end{aligned}$$

2.1.2 Coefficients

The function below is factored out.

$$F_1(s) = (s+7)(s+8)(s+5)(s+4)(s+2)s$$

Use `poly()` to determine its polynomial coefficients.

2.1.3 Zpk to expanded form

The function

$$T_1(s) = \frac{20(s+2)(s+3)(s+6)(s+8)}{s(s+7)(s+9)(s+10)(s+15)}$$

is in zero-pole-gain form, also known as zpk (for a gain k). Use the **Symbolic Math Toolbox** to display the numerator and denominator in expanded polynomial form in the Command Window.

2.1.4 Expanded form to zpk

The function below is in expanded polynomial form.

$$T_2(s) = \frac{s^4 + 17s^3 + 99s^2 + 22s + 140}{s^5 + 32s^4 + 36s^3 + 12s^2 + 70s + 1}$$

Use the **Symbolic Math Toolbox** to display the function in zpk form.

2.1.5 Transfer function operations

Use the previously defined functions and the **Symbolic Math Toolbox** to find and display the functions below.

$$\begin{aligned}
T_3(s) &= T_1 + T_2 \\
T_4(s) &= T_1 - T_2 \\
T_5(s) &= T_1 T_2 \\
T_6(s) &= \frac{T_1}{1 + T_1 T_2}
\end{aligned}$$

2.1.6 Partial fraction expansion

Calculate the partial fraction expansions of the following:

$$\begin{aligned}
T_7(s) &= \frac{5(s+2)}{s(s^2 + 8s + 15)} \\
T_8(s) &= \frac{5(s+2)}{(s^2 + 6s + 9)} \\
T_9(s) &= \frac{5(s+2)}{(s^2 + 6s + 34)}
\end{aligned}$$

2.1.7 Inverse Laplace

Use the **Symbolic Math Toolbox** to obtain the inverse Laplace transforms of the functions T_7, T_8, and T_9.

2.2 Manual plotting

2.2.1 Plotting with MATLAB

Using MATLAB code, find the step response, Bode plot, and pole-zero map of the below functions.

$$\begin{aligned}
G_1 &= \frac{s+1}{s^2 + s + 1} \\
G_2 &= \frac{s+2}{(2s^2 + 8s + 4)} \\
G_3 &= \frac{3s^2 + 2s + 1}{(4s^3 + 3s^2 + 2s + 1)}
\end{aligned}$$

2.2.2 Plotting with Simulink.

Repeat 2.2.1 using only Simulink blocks. Do not use the Linear System Analyzer yet. For the Bode plot and pole-zero map, you will have to place input-output points in the model as described in 4.1., and then add the points into the `Bode Plot` and `Pole-zero Plot` blocks.

2.3 Linear System Analyzer

Use the Linear System Analyzer to do the following:

- Plot impulse response
- Plot Nyquist diagram
- Find rise time of step response
- Find peak time of step response
- Find settling time of step response
- Find % overshoot of step response

for the functions in the last section, G_1, G_2, and G_3. How to obtain the desired signal characteristics and what they look like are shown in Figure 1. To use the Linear System Analyzer, you will have to place input-output points as described in 4.1.

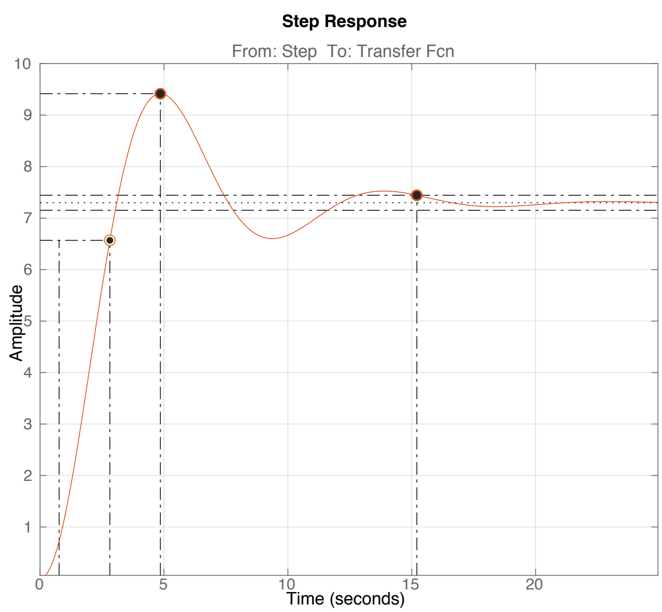

Figure 1. The signal characteristics of rise time, peak time, settling time, and % overshoot are shown as points without their exact values. To show the values in the plot, right-click on the graph and select the desired characteristics.

2.4 Harmonic oscillation

A classical damped harmonic oscillator will have a transfer function in the form of

$$F(s) = \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2} \tag{2}$$

with a natural fequency ω and a damping ratio ζ.

2.4.1 Changing the damping ratio

Find the natural frequency and damping ratio of the transfer function

$$G(s) = \frac{16}{s^2 + 8s + 16} \tag{3}$$

by using (2). Then, plot on the same graph the step responses for the values of

$$\zeta = 0.1,\ 0.2,\ 0.3,\ 0.5,\ 0.75.$$

On another graph, plot the step responses for the values of

$$\zeta = 1,\ 1.5,\ 2,\ 5,\ 10.$$

2.4.2 Changing the natural frequency

Plot a graph of (3) with its original natural frequency, double the original natural frequency, and quadruple the original natural frequency all on the same graph.

2.5 Block routing

There are three general forms of block diagram routing that are most common: cascade (series), parallel, and negative feedback. Their configurations and equivalent system functions are shown in Figures 2 through 4. So given the transfer functions

$$G_{12} = \frac{1}{2s+1},$$
$$G_{13} = \frac{2s+1}{3s^2+2s+1}, \text{ and}$$
$$G_{14} = \frac{3s^2+2s+1}{4s^3+3s^2+2s+1},$$

find the step responses of the system configurations in the subsubsections below by using Simulink and obtain the equivalent system functions using MATLAB code.

2.5.1 Cascade

Place the transfer functions G_{12}, G_{13}, and G_{14} in series, and obtain the step response of the system and its equivalent system function with `series()`.

2.5.2 Parallel

Place G_{12}, G_{13}, and G_{14} in parallel, and obtain the step response of the system and the equivalent system function with `parallel()`.

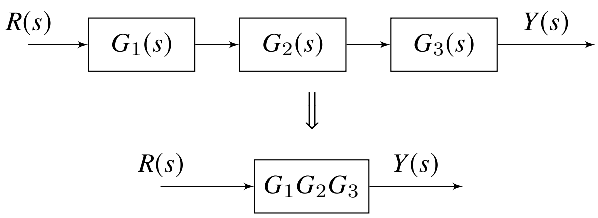

Figure 2. Cascading blocks is equivalent to multiplying system functions.

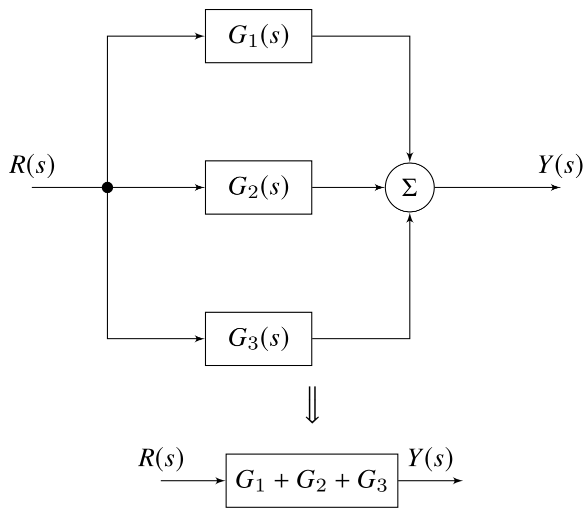

Figure 3. Parallelizing blocks is equivalent to summing system functions.

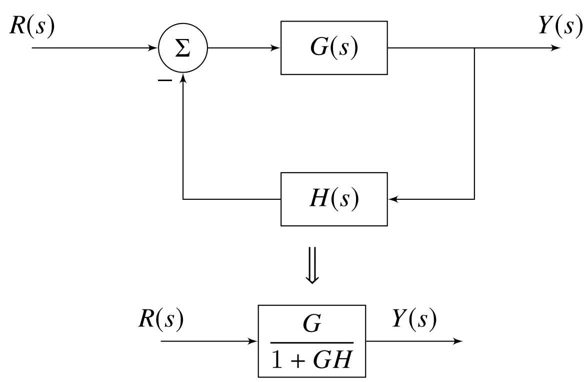

Figure 4. A block $G(s)$ with a block $H(s)$ in its negative feedback loop is equivalent to $\frac{G}{1+GH}$.

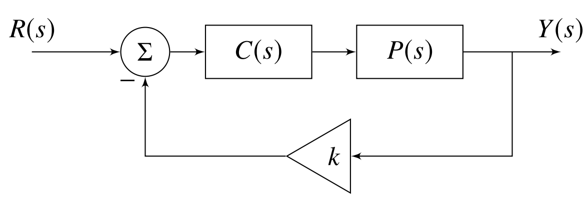

Figure 5. A common control systems scenario is being given a plant $P(s)$ with some behavior that we model our controller $C(s)$ around. We then adjust the feedback into the system with a manual gain (k) or sensor. When $k = 1$, this is known as a unity feedback system.

2.5.3 Negative feedback

Place G_{13} in the main path with G_{12} in its feedback loop, and obtain the step response of the system and the equivalent system function using `feedback()`.

2.5.4 Open-loop

Now just obtain the open-loop step responses of G_{12}, G_{13}, and G_{14} in one plot.

2.6 Negative feedback gain stability

2.6.1 Main path controller

Given the transfer function

$$C(s) = \frac{27s + 54}{5.5s^2 + 4s + 1}$$

Place this function in series with

$$P(s) = \frac{0.5}{s + 1}$$

With a negative feedback loop with adjustable gain, as shown in Figure 5. Adjust the gain in this system to values of

$$k = 0.01,\ 0.1,\ 0.5,\ 1,\ 2,\ 10,\ 100$$

and plot the step responses on the same graph. Consider what happens as k changes.

2.6.2 In-loop controller

Repeat the above experiment, but move $C(s)$ to the negative feedback path.

3. Report deliverables

For all parts include:

- Screenshots of Simulink block diagram models used. Make sure to resize the transfer function blocks to show the function
- Any MATLAB code used. Make sure to properly format it in monospace font. Do not screenshot the MATLAB IDE
- All mentioned plots
- Values that are explicitly asked for
- Functions that are asked for, either typeset or written `pretty` from the MATLAB command window

4. Appendix

4.1 Placing input-output points for linear analysis

To use the Linear System Analyzer and some other system plotting tools, a Simulink model must have input-output points. As shown in Figure 6, you can put the input point before a transfer function, and an output point after to analyze the lone function's response. After placing the input-output points, navigate the menu bar for

```
Analysis > Control Design > Linear Analysis...
```

This will open the Linear System Analyzer.

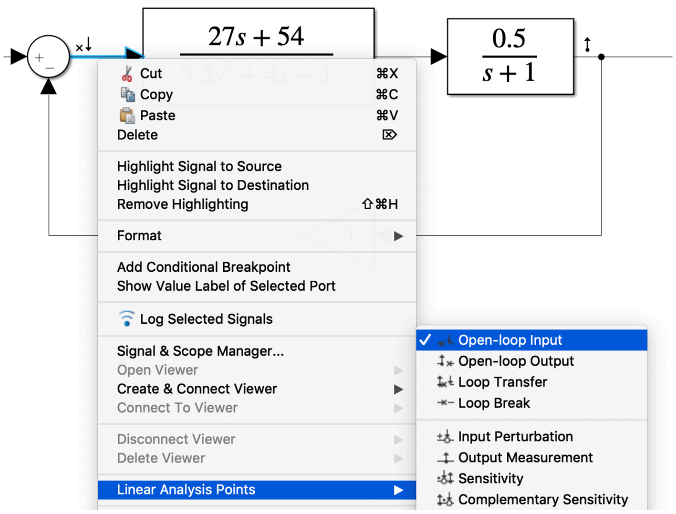

Figure 6. Right-click on the signal you want to add input or output points for, then go to `Linear Analysis Points`.

4.2 Poles and zeros

In a generic transfer function (1), zeros are the roots of $Z_{\circ}(s)$ (the numerator), and poles are the roots of $P_{\times}(s)$ (the denominator). The zeros and poles give insight into system behavior, e.g. zeros with a negative real part imply minimum phase and poles with a positive real part imply instability.