Experiment -1-

Tutorial on Matlab commands for control systems

Objectives:

To learn various Matlab commands for control systems.

Introduction:

In this experiment, we introduce several Matlab commands that are considered essential for solving control systems such as the transfer function, feedback, step, and stepinfo.

Transfer function:

A transfer function (**tf**) is a rational function of a complex variable that represents a linear time-invariant dynamical system with zero initial conditions. It describes the relation between the input and the output of the system.

To create the transfer function in Matlab, one can use the following command:

```
>> sys= tf (num, den)
```

where it creates a continuous-time transfer function with numerator and denominator specified by num and den.

Another way of creating a transfer function is by assigning a variable, for example, 's', as a variable of the transfer function. Then, simply type the transfer function in the command window. This is a more convenient way, in case we handle subsystems. In such a case, one can use

```
>> s= tf ('s');
```

This command is used to assign variable s as a variable of the transfer function.

Feedback:

If either the output or some part of the output is returned to the input side and utilized as part of the system input, then it is known as **feedback**. Feedback plays an important role in improving the performance of the control systems.

There are two types of feedback:-

- Positive feedback
- Negative feedback

Positive feedback

The positive feedback adds the reference input **R(s)** and feedback output. The following figure shows the block diagram of the **positive feedback control system**:

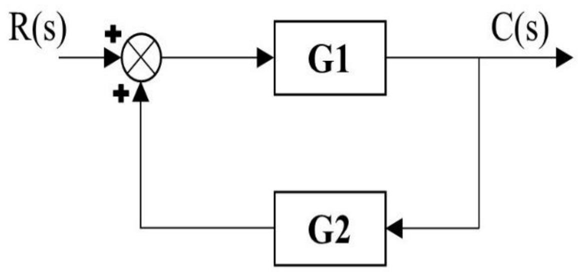


Fig. (1) Positive feedback representation.

where **G1** is the open loop gain, **G2** is the gain of the feedback path, and **C(s)** is the output of the given system. The transfer function of the **positive feedback control system** is given in the following function:

$$T = \frac{G1}{1 - G1G2} \dots\dots\dots (1)$$

The command to define a system with positive feedback in Matlab is

```
>> T=feedback (G1, G2, 1) % for positive feedback.
```

Negative feedback

Negative feedback reduces the error between the reference input **R(s)** and system output. The following figure shows the block diagram of the **negative feedback control system**.

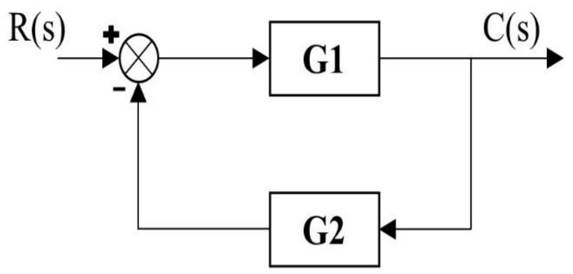


Fig. (2) Negative feedback representation.

The transfer function of the **negative feedback control system** is given in the following function:

$$T = \frac{G1}{1 + G1G2} \dots \dots \dots (2)$$

The command to define a system with negative feedback in Matlab is

```
>> T=feedback (G1, G2,-1) % for negative feedback.
```

step:

This command is used to plot the step response of a given system. We can use the following command in Matlab to represent the **step** function:

```
>> step(T);
```

stepinfo:

This command is used to compute the step response characteristics such as rising time, settling time, peak time, overshoot, etc. The following Matlab command is used to obtain the step response characteristics:

```
>> stepinfo(T)
```

Example 1: Consider the following closed-loop block diagram

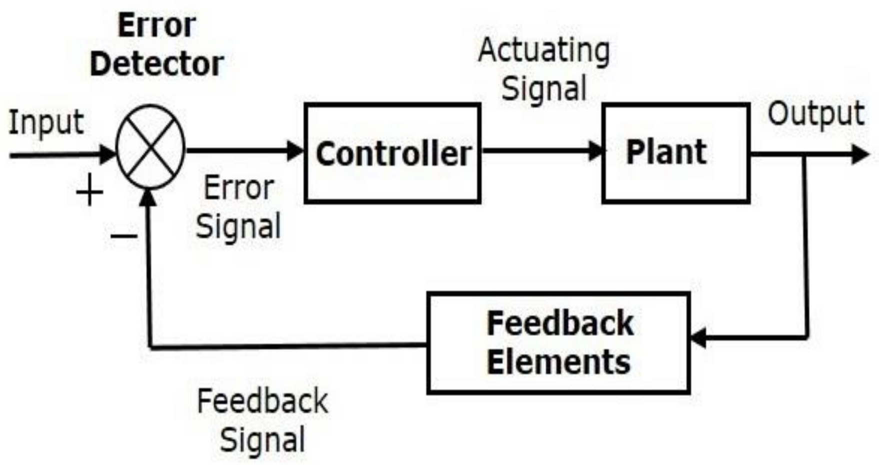


Fig. (3) Closed loop system.

A step input response with unity feedback has been applied along with unity controller gain and the following plant representation:

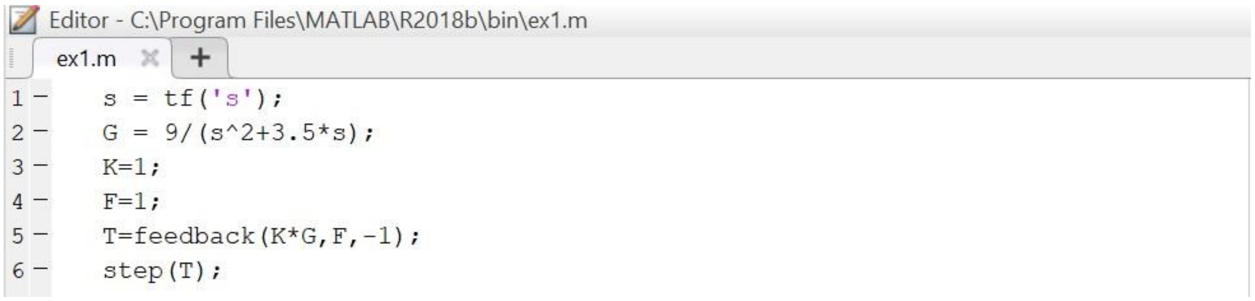
$$G(s) = \frac{\mathbf{9}}{\boldsymbol{s^2 + 3.5s}} \quad \dots\dots\dots (\mathbf{3})$$

1-Obtain the output response using (step, feedback, tf).

Sol:-

Firstly, let us define K as the controller transfer function, G as the plant transfer function, and F as the feedback transfer function.

Next, create a new m-file and save this file then write the following Matlab code, from which we can get the output response:



```
s = tf('s');
G = 9/(s^2+3.5*s);
K=1;
F=1;
T=feedback(K*G,F,-1);
step(T);
```

Now, after running the m-file, the following output response is obtained

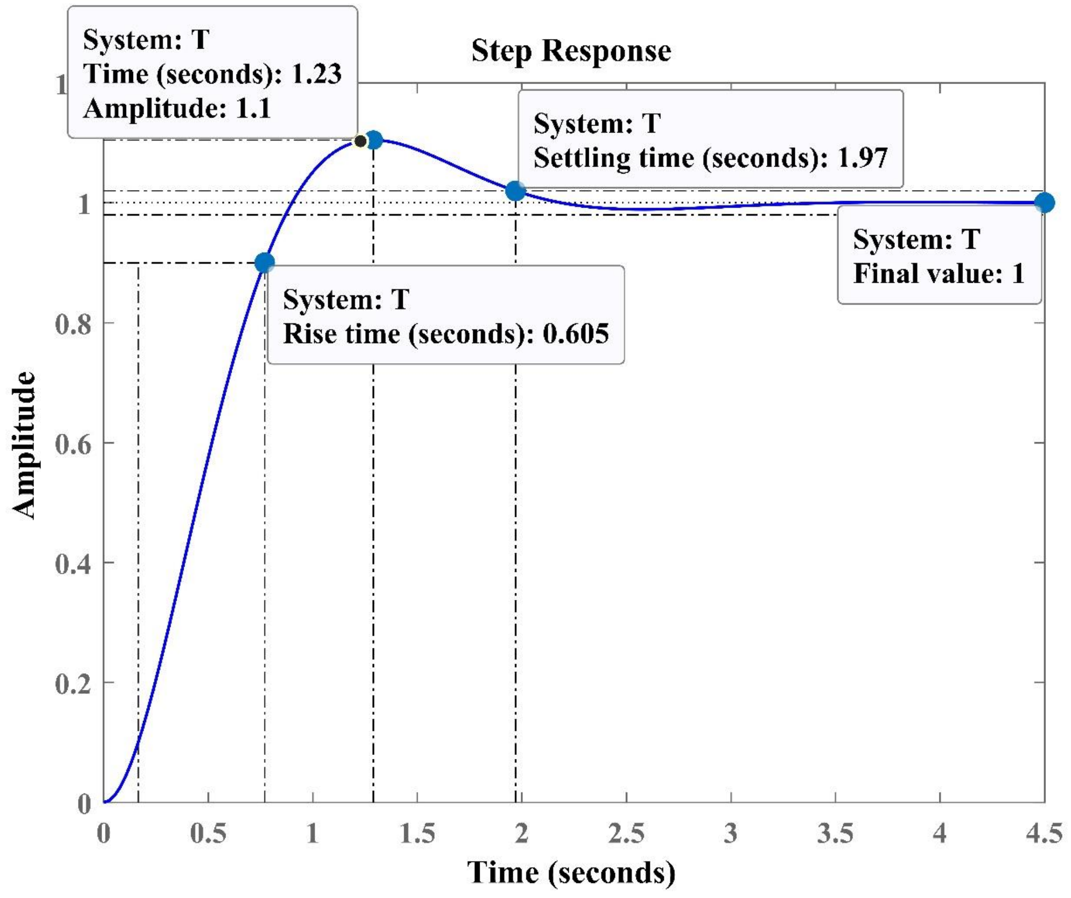


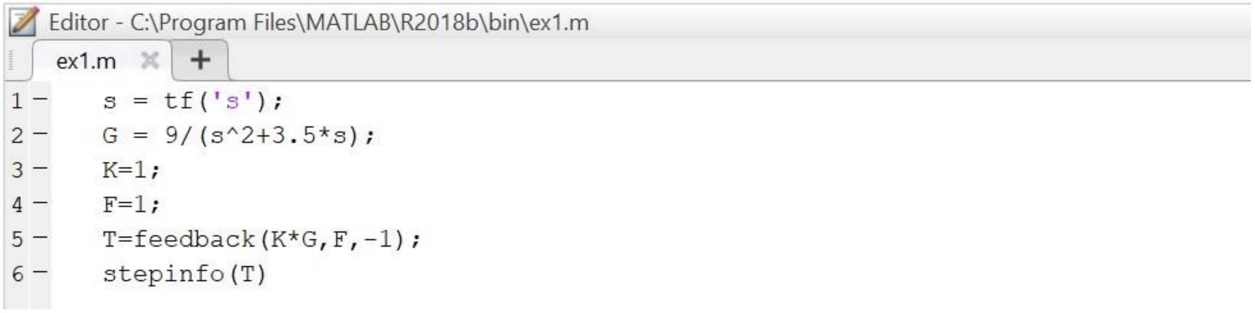
Fig. (4) The output step response.

The output step response characteristics can be easily obtained by moving the pointer inside the figure and press→right click→ characteristics→tick →(Peak response, Settling time, Rise time, steady state) as shown in Fig. (4).

2- Obtain the rising time, settling time, peak time, and overshoot using **stepinfo()**.

Sol:-

Write the following code into the m-file and press run:-



```
s = tf('s');
G = 9/(s^2+3.5*s);
K=1;
F=1;
T=feedback(K*G,F,-1);
stepinfo(T)
```

```
>> ex1
ans =

  struct with fields:

        RiseTime: 0.6053
    SettlingTime: 1.9699
     SettlingMin: 0.9175
     SettlingMax: 1.1047
       Overshoot: 10.4743
      Undershoot: 0
            Peak: 1.1047
        PeakTime: 1.2894
```

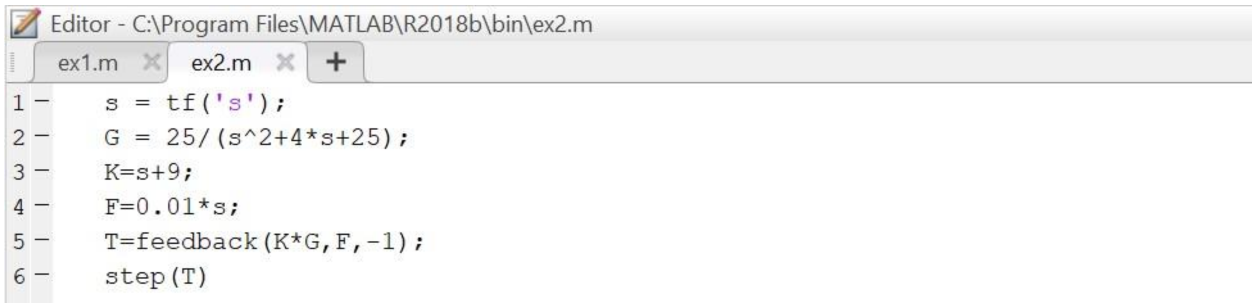
<u>Example 2</u>: Consider the system given in Fig. (3) that has the following:

$G(s) = \frac{25}{s^2+4s+25}$, $K = s + 9$, and $F = 0.01 * s$.

1-Obtain the output response using (step, feedback, tf).

Sol:-

Create a new m-file and save this file then write the following Matlab code:



```
s = tf('s');
G = 25/(s^2+4*s+25);
K=s+9;
F=0.01*s;
T=feedback(K*G,F,-1);
step(T)
```

The output response is obtained as shown in the following graph:

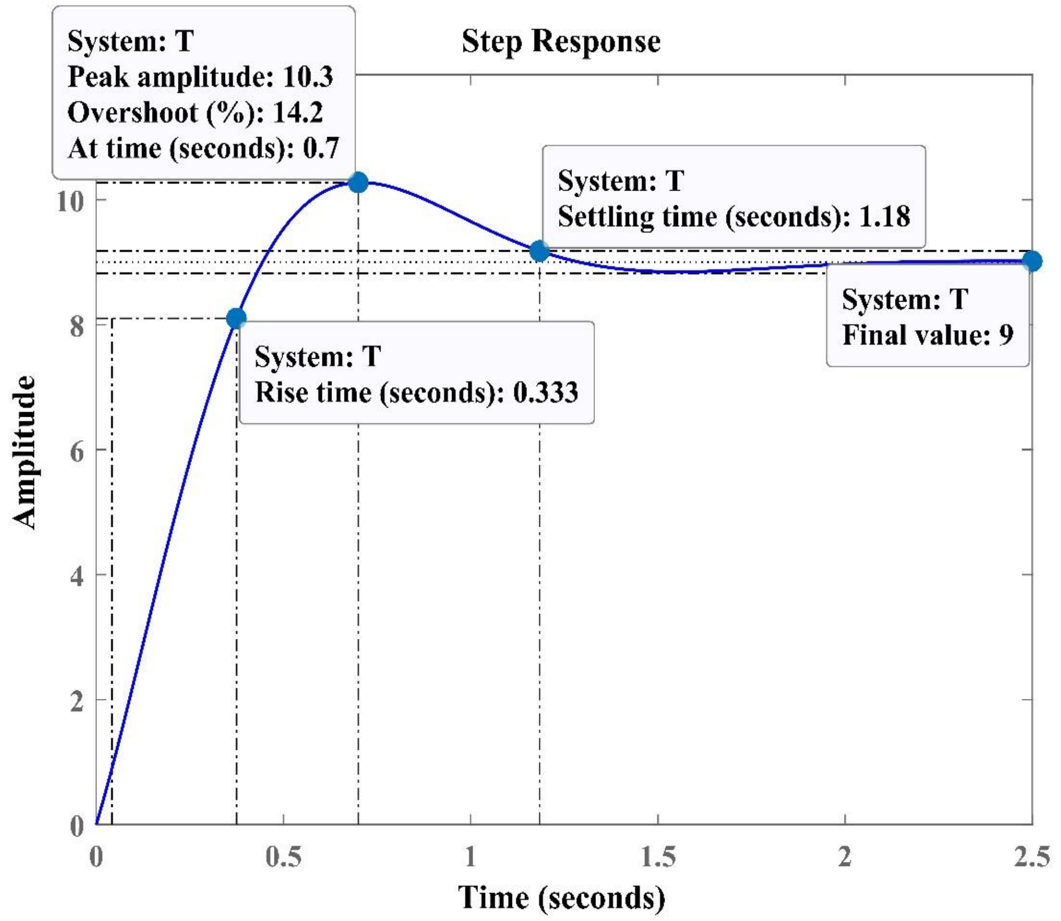
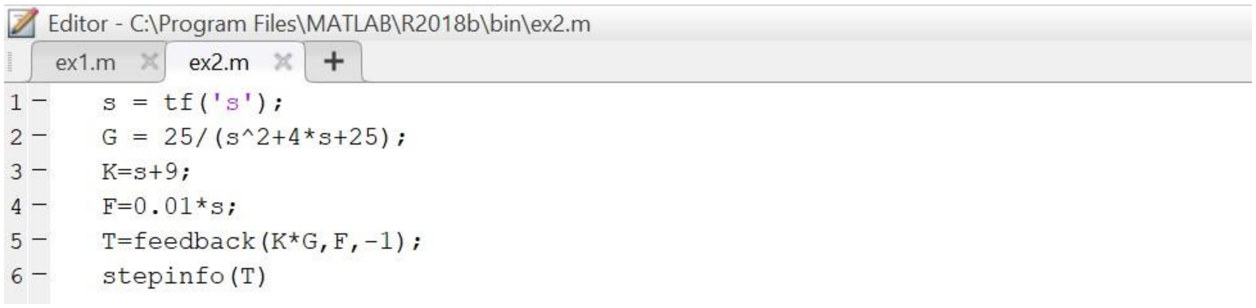


Fig. (5) The output step response.

2- Obtain the rising time, settling time, peak time, and overshoot using **stepinfo()**.

Sol:-

Write the following code into the m-file and press run:-



```
s = tf('s');
G = 25/(s^2+4*s+25);
K=s+9;
F=0.01*s;
T=feedback(K*G,F,-1);
stepinfo(T)
```

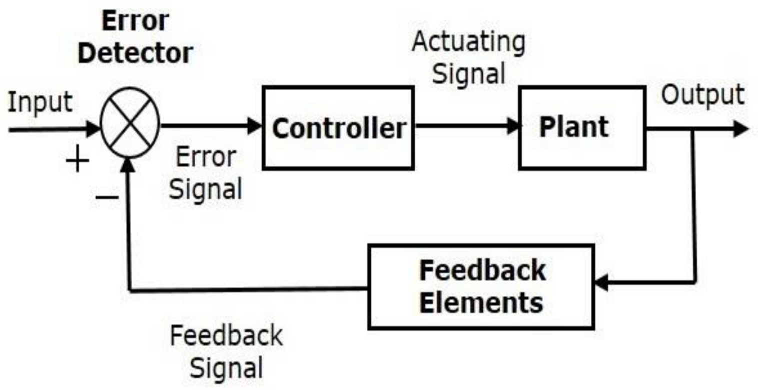
```
>> ex2
ans =

  struct with fields:

        RiseTime: 0.3327
    SettlingTime: 1.1845
     SettlingMin: 8.2809
     SettlingMax: 10.2758
       Overshoot: 14.1756
      Undershoot: 0
            Peak: 10.2758
        PeakTime: 0.7000
```

Exercise 1:

Consider the following closed-loop block diagram



1- obtain the step response using the (step, feedback, tf) functions. where

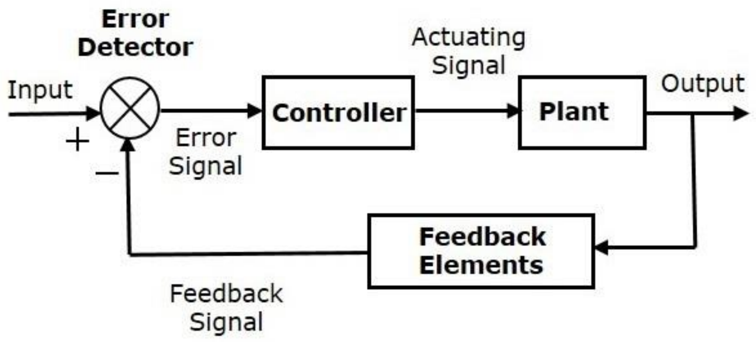
$Input = unit\ step$		$Plant = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s}$		
Case	**Controller**	ω_n	ξ	**Feedback**
Case-1	**1**	**4**	**0**	**0**
Case-2	**1**	**4**	**0.6**	**0**
Case-3	**s+4**	**4**	**0**	**1**
Case-4	**1**	**4**	**0**	**0.25*s**

For each case, write the Matlab code and plot the response on graph paper.

2- Use the "stepinfo()" instruction to obtain the rising time, settling time, peak time, and overshoot.

Exercise 2:

Consider the following closed-loop block diagram



1- obtain the step response using the (step, feedback, tf) functions. where

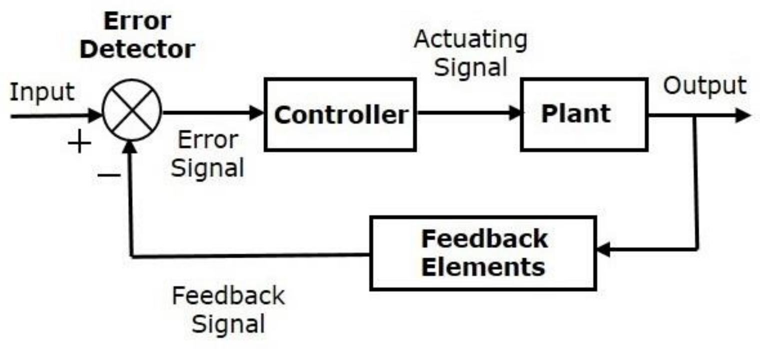
$\boldsymbol{Input = unit\ step}$			$\boldsymbol{Plant = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s}}$	
Case2	**Controller**	$\boldsymbol{\omega_n}$	$\boldsymbol{\xi}$	**Feedback**
Case-1	**1**	**2**	**0**	**0**
Case-2	**1**	**2**	**0.3**	**0**
Case-3	**10(s+6)**	**2**	**0.3**	**1**
Case-4	**1**	**2**	**0.3**	**0.6*s**

For each case, write the Matlab code and plot the response on graph paper.

2- Use the "stepinfo()" instruction to obtain the rising time, settling time, peak time, and overshoot.

<u>Exercise 3</u>:

Consider the following closed-loop block diagram



1- obtain the step response using the (step, feedback, tf) functions. where

$\boldsymbol{Input = unit\ step}$			$\boldsymbol{Plant = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s}}$	
Case	**Controller**	$\boldsymbol{\omega_n}$	$\boldsymbol{\xi}$	**Feedback**
Case-1	**1**	**1**	**0**	**0**
Case-2	**1**	**1**	**0.6**	**0**
Case-3	**10(s+4)**	**1**	**0.6**	**1**
Case-4	**1**	**1**	**0.6**	**0.8*s**

For each case, write the Matlab code and plot the response on graph paper.

2- Use the "stepinfo()" instruction to obtain the rising time, settling time, peak time, and overshoot.