VICTORIA UNIVERSITY OF WELLINGTON
TE HERENGA WAKA

XMUT315 Control Systems Engineering

Demo 4: Time Domain Analysis

A. Transient Response Analysis

In this section, we will cover several example-exercises of transient response analysis.

Exercise 1 (Transient Response Analysis - Unit Step Response)

A higher-order system is defined by:

$$\frac{C(s)}{R(s)} = \frac{7s^2 + 16s + 10}{s^4 + 5s^3 + 11s^2 + 16s + 10}$$

a. Plot the unit-step response curve of the system using MATLAB.

b. Obtain the rise time, peak time, maximum overshoot, and settling time using MATLAB.

Solution

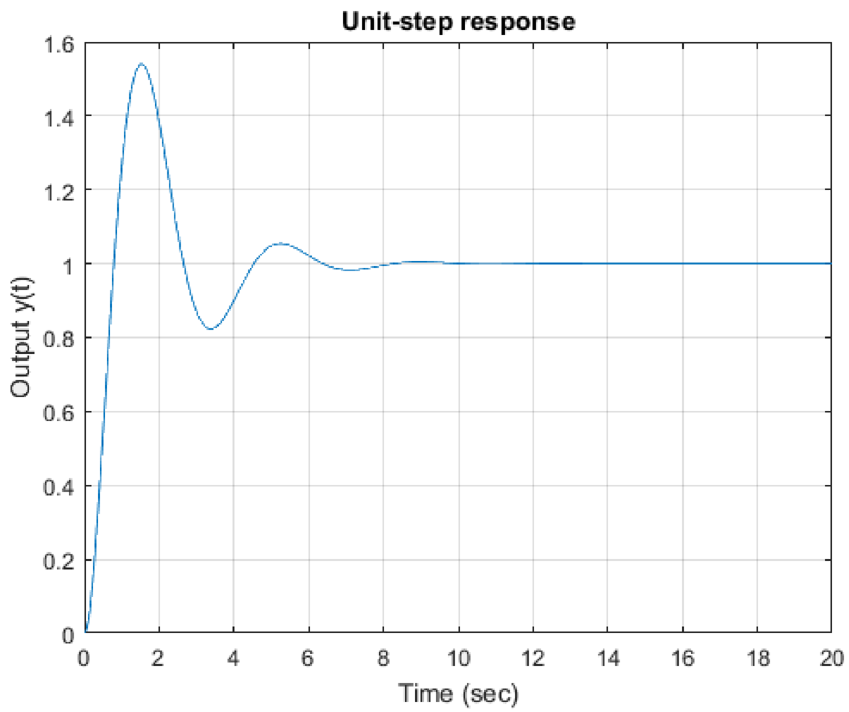
a. The MATLAB code for simulating the unit step response curve of the system is given below.

MATLAB code:

```
% demo41a.m

% Unit-step response curve

num=[0 0 7 16 10];

den=[1 5 11 16 10];

t=0:0.02:20;
```

```
[y,x,t]=step(num,den,t);
plot(t,y)
grid
title('Unit-step response')
xlabel('Time (sec)')
ylabel('Output y(t)')
```

The result of the MATLAB simulation is given in the diagram below.



b. Obtain the rise time, peak time, maximum overshoot, and settling time using MATLAB.

The rise time, peak time, maximum overshoot, and settling time of the system is simulated using the following MATLAB code.

MATLAB code:

```
% demo41b.m
% Response to rise from 10% to 90% of its final value
r1=1; while y(r1)<0.1, r1=r1+1; end
r2=1; while y(r2)<0.9, r2=r2+1; end
```

```
rise_time=(r2-r1)*0.02

[ymax,tp]=max(y);

peak_time=(tp-1)*0.02

max_overshoot=ymax-1

s=1001; while y(s)>0.98 & y(s)<1.02; s=s-1; end

settling_time=(s-1)*0.02
```

Type the following variables in the command prompt: rise_time, peak_time, max_overshoot, and settling_time. The outcomes of the MATLAB simulation are as shown below.

```
rise_time =

    0.5400


peak_time =

    1.5200


max_overshoot =

    0.5397


settling_time =

    6.0200
```

Exercise 2 (Transient Response Analysis – Second Order Systems)

For each of the second order systems below, find ζ, ω_n, T_s, T_p, T_r, $\%OS$, and plot the step response using MATLAB.

a. System 1:

$$T_1(s) = \frac{130}{s^2 + 15s + 130}$$

b. System 2:

$$T_2(s) = \frac{0.045}{s^2 + 0.025s + 0.045}$$

c. System 3:

$$T_3(s) = \frac{10^8}{s^2 + 1.325 \times 10^3 s + 10^8}$$

Comment on the results of the step response simulation.

Solution

a. The first system's ζ, ω_n, T_s, T_p, T_r, $\%OS$, and plot the step response.

The MATLAB code for simulating the first system's time response parameters and plotting the step response.

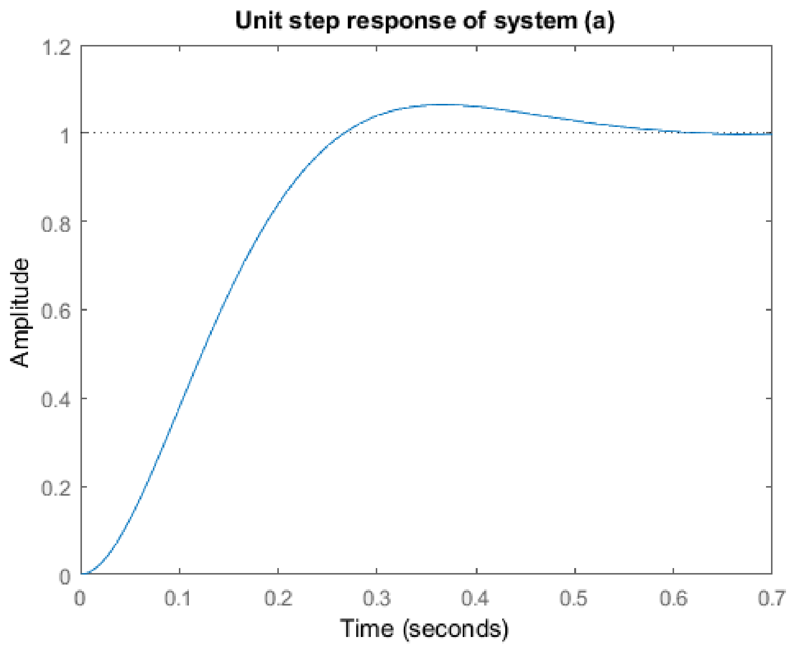
MATLAB code:

```
% demo42a.m

% Time domain responses of the first system

clf

numa=130;

dena=[1 15 130];

Ta=tf(numa,dena)

omegana=sqrt(dena(3))

zetaa=dena(2)/(2*omegana)

Tsa=4/(zetaa*omegana)

Tpa=pi/(omegana*sqrt(1-zetaa^2))

Tra=(1.76*zetaa^3-0.417*zetaa^2+1.039*zetaa+1)/omegana

percenta=exp(-zetaa*pi/sqrt(1-zetaa^2))*100

step(Ta)

title('Time response of system (a)')
```

The outcomes of the MATLAB simulation and plot are as shown below:

```
Ta =

         130
  ----------------
  s^2 + 15 s + 130

Continuous-time transfer function.


omegana =

   11.4018


zetaa =

    0.6578



Tsa =

    0.5333


Tpa =

    0.3658


Tra =

    0.1758


percenta =

    6.4335
```

The time response of the first system is as shown in the diagram given below.



b. The second system's ζ, ω_n, T_s, T_p, T_r, $\%OS$, and plot the step response.

The MATLAB code for simulating the second system's time response parameters and plotting the step response.

MATLAB code:

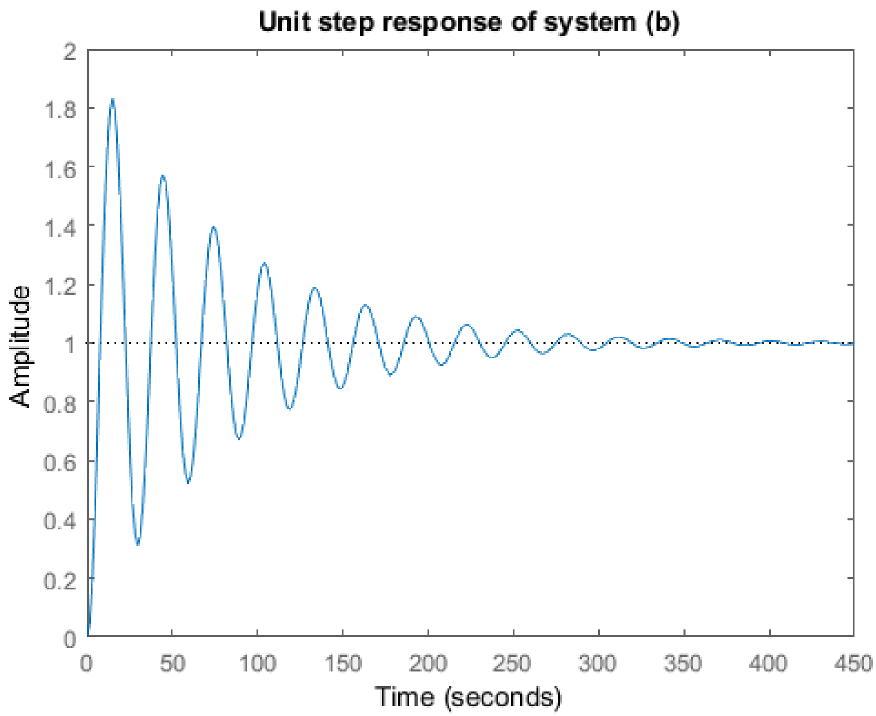
```
% demo42b.m
% Time domain responses of the second system
clf
numb=0.045;
denb=[1 0.025 0.045];
Tb=tf(numb,denb)
omeganb=sqrt(denb(3))
zetab=denb(2)/(2*omeganb)
Tsb=4/(zetab*omeganb)
Tpb=pi/(omeganb*sqrt(1-zetab^2))
Trb=(1.76*zetab^3-0.417*zetab^2+1.039*zetab+1)/omeganb
percentb=exp(-zetab*pi/sqrt(1-zetab^2))*100
step(Tb)
```

```
title('Unit step response of system (b)')
```

The outcomes of the MATLAB simulation and plot are as shown below:

```
Tb =

          0.045
  ---------------------
  s^2 + 0.025 s + 0.045

Continuous-time transfer function.


omeganb =

    0.2121


zetab =

    0.0589


Tsb =

   320


Tpb =

   14.8354


Trb =

    4.9975


percentb =

   83.0737
```

The time response of the second system is as shown in the diagram given below.



c. The third system's ζ, ω_n, T_s, T_p, T_r, $\%OS$, and plot the step response.

The MATLAB code for simulating the third system's time response parameters and plotting the step response.

MATLAB code:

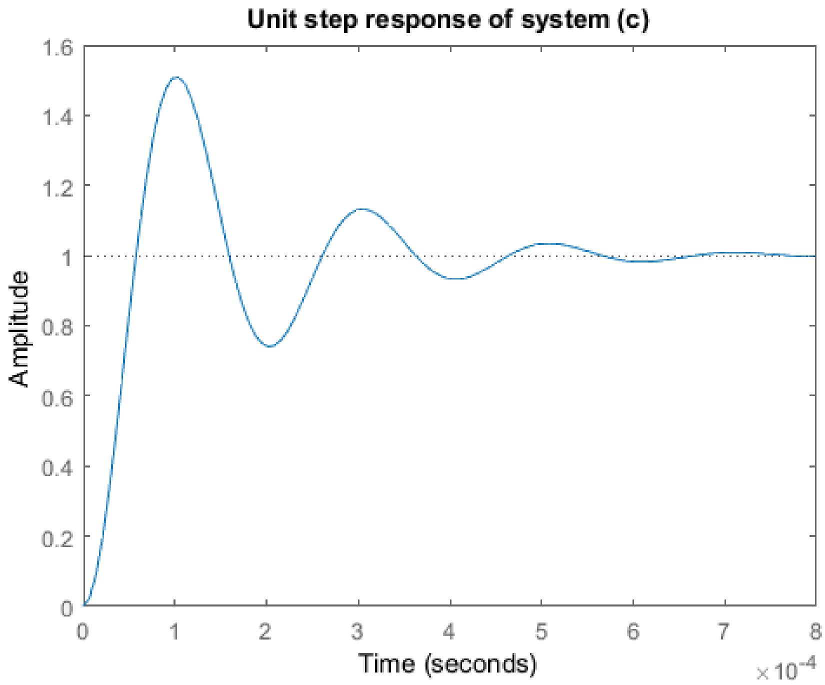
```
% demo42c.m

% Time domain responses of the third system

clf

numc=1e8;

denc=[1 1.325e3 1e8];

Tc=tf(numc,denc)

omeganc=sqrt(denc(3))

zetac=denc(2)/(2*omeganc)

Tsc=4/(zetac*omeganc)

Tpc=pi/(omeganc*sqrt(1-zetac^2))

Trc=(1.76*zetac^3-0.417*zetac^2+1.039*zetac+1)/omeganc
```

```
percentc=exp(-zetac*pi/sqrt(1-zetac^2))*100

step(Tc)

title('Unit step response of system (c)')
```

The outcomes of the MATLAB simulation and plot are as shown below:

```
Tc =

          1e09
  --------------------
  s^2 + 13250 s + 1e09

Continuous-time transfer function.


omeganc =

   3.1623e+04


zetac =

    0.2095


Tsc =

   6.0377e-04


Tpc =

   1.0160e-04


Trc =

   3.8439e-05


percentc =

   51.0123
```

The time response of the third system is as shown in the diagram given below.



The results of step responses simulations show that the first, second and third systems are underdamped responses. The first system settles down quicker than other systems which is followed by second and third systems.

Exercise 3 (Transient Response Analysis - System with Delay)

For a unit feedback system with the forward-path transfer function:

$$G(s) = \frac{K}{s(s+5)(s+12)}$$

This system has a delay of 0.5 second, estimate the percent overshoot for $K = 40$ using a second-order approximation. Model the delay using MATLAB function `pade(T,n)`. Determine the unit step response and check the second-order approximation assumption made.

Solution

The MATLAB code that simulates a unit step response of the given feedback system is as follows.

MATLAB code:

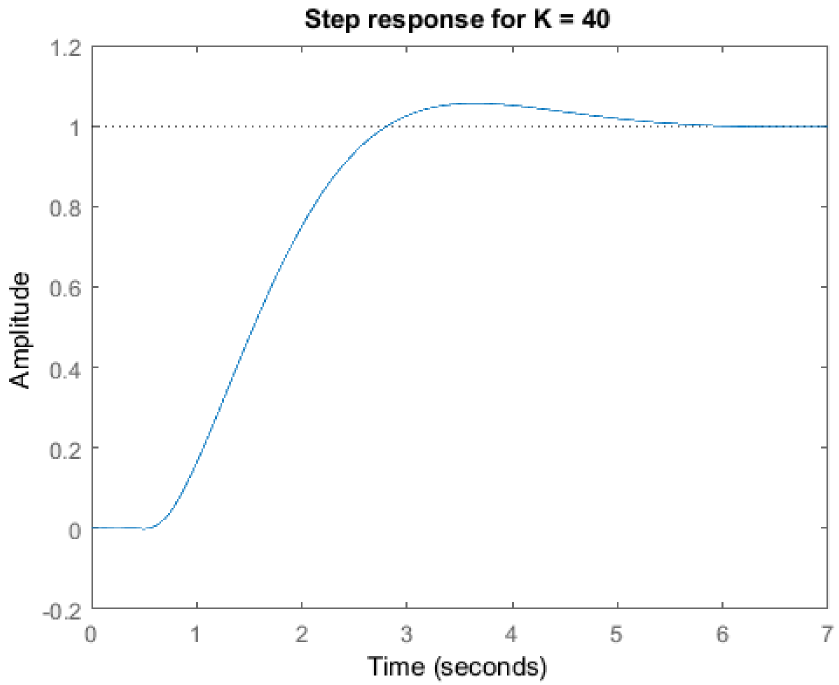
```
% demo43.m
```

```
% Unit step response of a system with delay with pade(T,n)

% Enter G(s)

numg1=1;

deng1=poly([0 -5 -12]);

'G1(s)'

G1=tf(numg1,deng1)

[numg2,deng2]=pade(0.5,5);

'G2(s)'

G2=tf(numg2,deng2)

'G(s)=G1(s)G2(s)'

G=G1*G2

% Enter K

K=input('Type gain, K: ');

T=feedback(K*G,1);

step(T)

title(['Step response for K = ', num2str(K)])
```

Output of this program is as follows:

```
    'G1(s)'


G1 =

           1
  -------------------
  s^3 + 17 s^2 + 60 s

Continuous-time transfer function.


ans =

    'G2(s)'


G2 =

  -s^5 + 60 s^4 - 1680 s^3 + 2.688e04 s^2 - 2.419e05 s + 9.677e05
  ----------------------------------------------------------------
  s^5 + 60 s^4 + 1680 s^3 + 2.688e04 s^2 + 2.419e05 s + 9.677e05

Continuous-time transfer function.

ans =

    'G(s)=G1(s)G2(s)'


G =

                  -s^5 + 60 s^4 - 1680 s^3 + 2.688e04 s^2 - 2.419e05 s + 9.677e05
  ----------------------------------------------------------------------------------------------
  s^8 + 77 s^7 + 2760 s^6 + 5.904e04 s^5 + 7.997e05 s^4 + 6.693e06 s^3 + 3.097e07 s^2 + 5.806e07 s

Continuous-time transfer function.
```

When the simulation is running, type Gain, $K = 40$, as a result the following figure is obtained.



B. Steady-State Analysis

In this section, we will demonstrate some example-exercises of steady-state analysis.

Exercise 4 (Steady-State Analysis - Ramp Input)

Determine the unit-ramp response of the following system using MATLAB and `lsim()` function.

$$\frac{C(s)}{R(s)} = \frac{1}{3s^2 + 2s + 1}$$

Answer

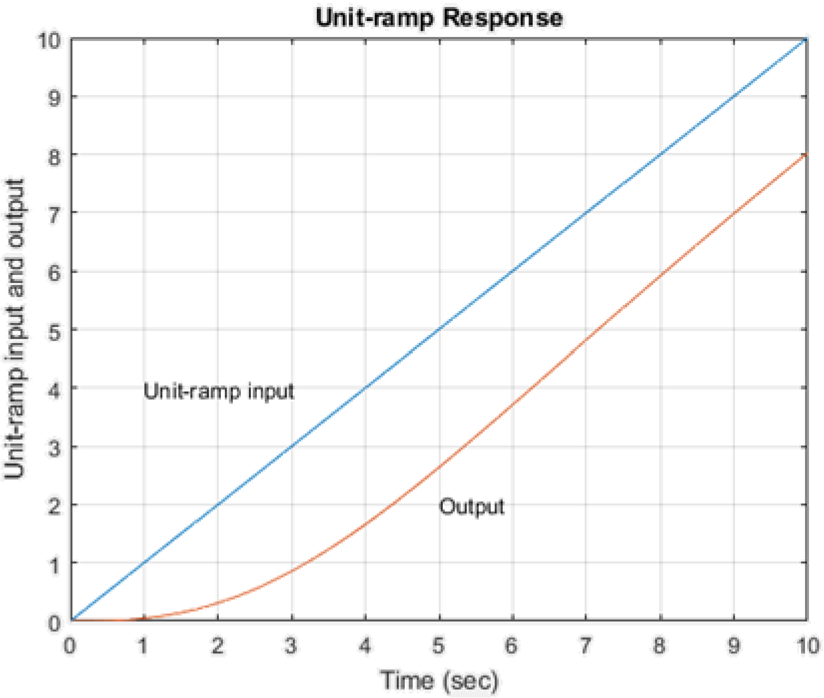
The MATLAB code for simulating a unit ramp response of the system given system is given as follows.

MATLAB code:

```
% demo34.m

% Unit-ramp response of a system with lsim() function

num=[0 0 1];

den=[3 2 1];
```

```
t=0:0.1:10;
r=t;
y=lsim(num,den,r,t);
plot(t,r,t,y)
grid
title('Unit-ramp Response')
xlabel('Time (sec)')
ylabel('Unit-ramp input and output')
text(1.0,4.0,'Unit-ramp input')
text(5.0,2.0,'Output')
```

The result of the MATLAB simulation is given in the diagram below.



Exercise 5 (Steady-State Analysis – Ramp and Exponential Inputs)

For the following closed-loop control system whose closed-loop transfer function is given by:

$$\frac{C(s)}{R(s)} = \frac{s+12}{s^3+5s^2+8s+12}$$

a. Obtain the unit-ramp response of the system using `lsim()` function in MATLAB.

b. Determine also the response of the system when the input is given by:

$$r = e^{-0.7t}$$

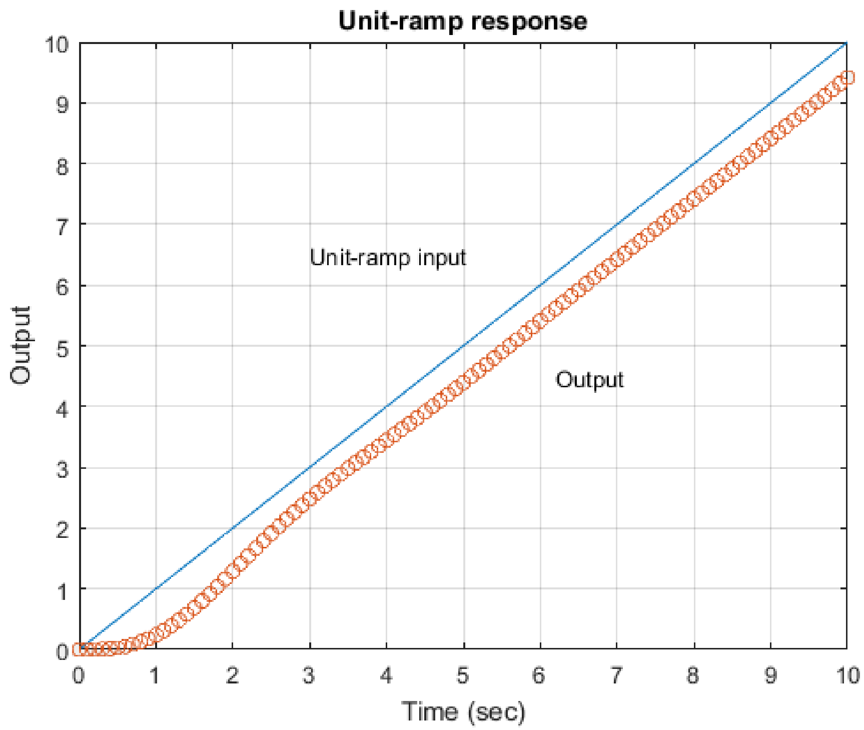
Answer

a. The MATLAB code for simulating a unit-ramp response of the given closed-loop control system is as follows:

MATLAB code:

```
% demo45a.m

% Unit-ramp response of a system with lsim() function

num=[0 0 1 12];
den=[1 5 8 12];

t=0:0.1:10;

r=t;

y=lsim(num,den,r,t);

plot(t,r,'-',t,y,'o')

grid

title('Unit-ramp response')
xlabel('Time (sec)')
ylabel('Output')
text(3.0,6.5,'Unit-ramp input')
text(6.2,4.5,'Output')
```

The unit-ramp response output of the MATLAB simulation is given in the following diagram below.



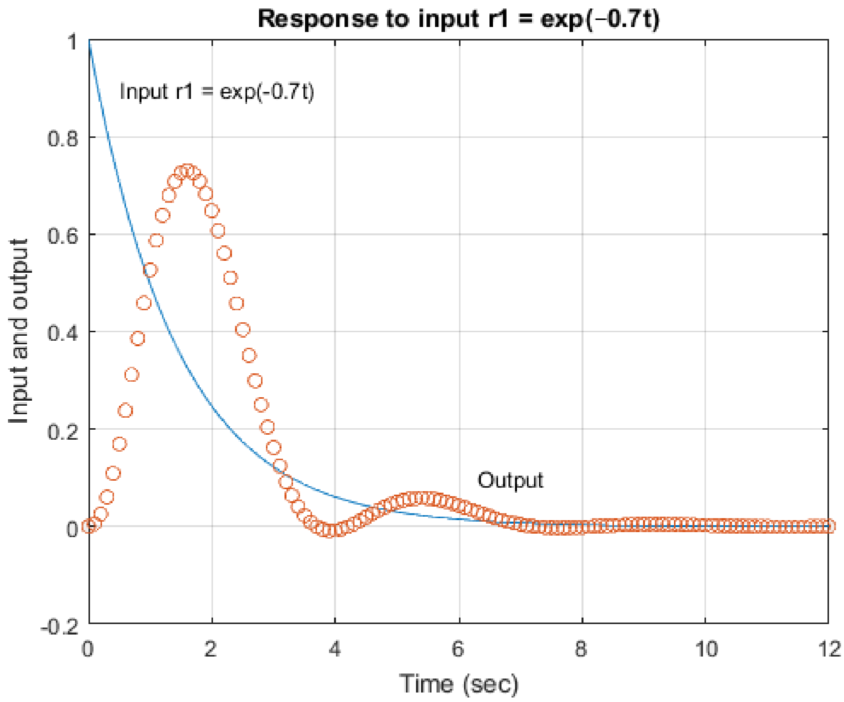
b. The following MATLAB code is used for simulating the response of the system whenever it is subjected to input of `r=exp(-0.7t)`.

MATLAB Code:

```
% demo45b.m

% Response of a system with exponential input

% Input r1=exp(-0.7t)

num=[0 0 1 12];
den=[1 5 8 12];

t=0:0.1:12;

r1=exp(-0.7*t);

y1=lsim(num,den,r1,t);

plot(t,r1,'-',t,y1,'o')

grid

title('Response to input r1 = exp(-0.7t)')
xlabel('Time (sec)')
ylabel('Input and output')
```

```
text(0.5,0.9,'Input r1 = exp(-0.7t)')
text(6.3,0.1,'Output')
```

The time response to the given exponential input outcome of the MATLAB simulation is given in the following diagram below.



Exercise 6 (Steady-State Analysis – Step and Ramp Inputs)

The closed-loop system is defined by:

$$\frac{C(s)}{R(s)} = \frac{7}{s^2 + 2 + 7}$$

Obtain the response of the closed-loop system using MATLAB when it is subjected to input `r(t)` is a step input of magnitude 3 plus unit-ramp input, `r(t)=3+t`.

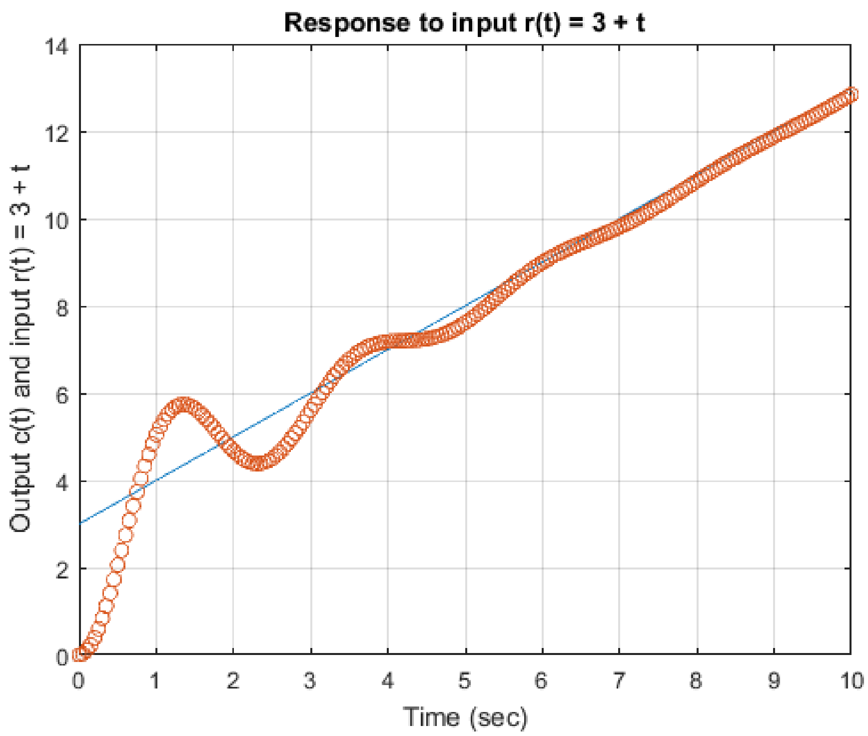
Answer

The following MATLAB code is for obtaining the response of the system whenever it is given a step input of magnitude 3 plus a unit ramp input (e.g. `r(t)=3+t`).

MATLAB code:

```
% demo46.m

% Unit step and ramp response of a system

num=[0 0 7];
den=[1 1 7];

t=0:0.05:10;

r=3+t;

c=lsim(num,den,r,t);

plot(t,r,'-',t,c,'o')

grid

title('Response to input r(t) = 3 + t')
xlabel('Time (sec)')
ylabel('Output c(t) and input r(t) = 3 + t')
```

The time response to the given ramp input result of the MATLAB simulation is given in the diagram below.



Appendix

`stepinfo()` function

Rise time, settling time, and other step-response characteristics.

Syntax

S = stepinfo(sys)

S = stepinfo(y,t)

S = stepinfo(y,t,yfinal)

S = stepinfo(y,t,yfinal,yinit)

S = stepinfo(___,'SettlingTimeThreshold',ST)

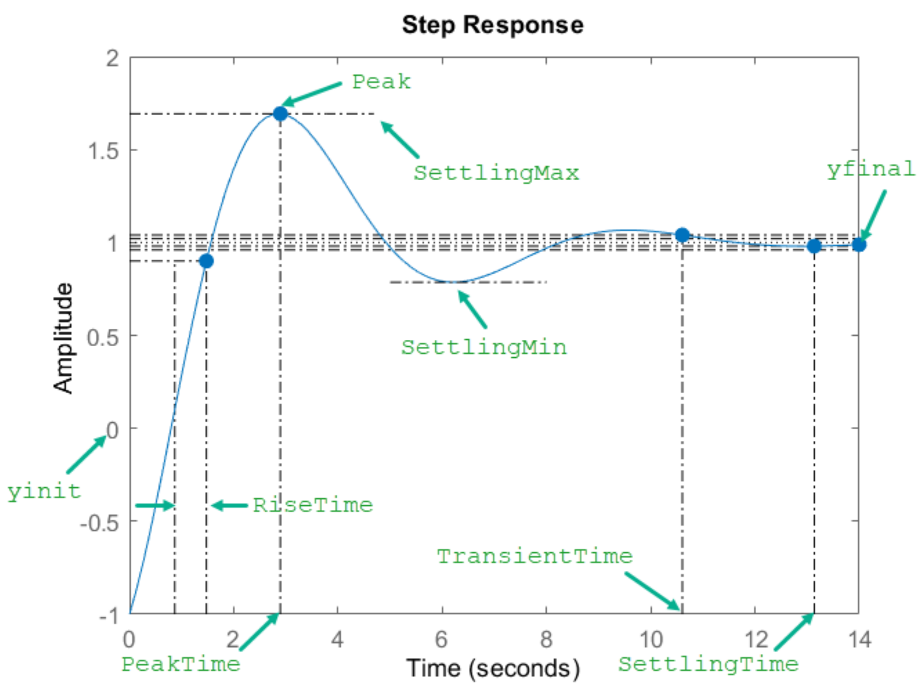
S = stepinfo(___,'RiseTimeLimits',RT)

Description

`stepinfo` lets you compute step-response characteristics for a dynamic system model or for an array of step-response data. For a step response $y(t)$, `stepinfo` computes characteristics relative to y_{init} and y_{final}, where y_{init} is the initial offset, that is, the value before the step is applied, and y_{final} is the steady-state value of the response. These values depend on the syntax you use.

- For a dynamic system model sys, `stepinfo` uses y_{init} = 0 and y_{final} = steady-state value.
- For an array of step-response data `[y,t]`, stepinfo uses y_{init} = 0 and y_{final} = last sample value of y, unless you explicitly specify these values.

The following figure illustrates some of the characteristics `stepinfo` computes for a step response. For this response, assume that $y(t)$ = 0 for t < 0, so y_{init} = 0.



Examples

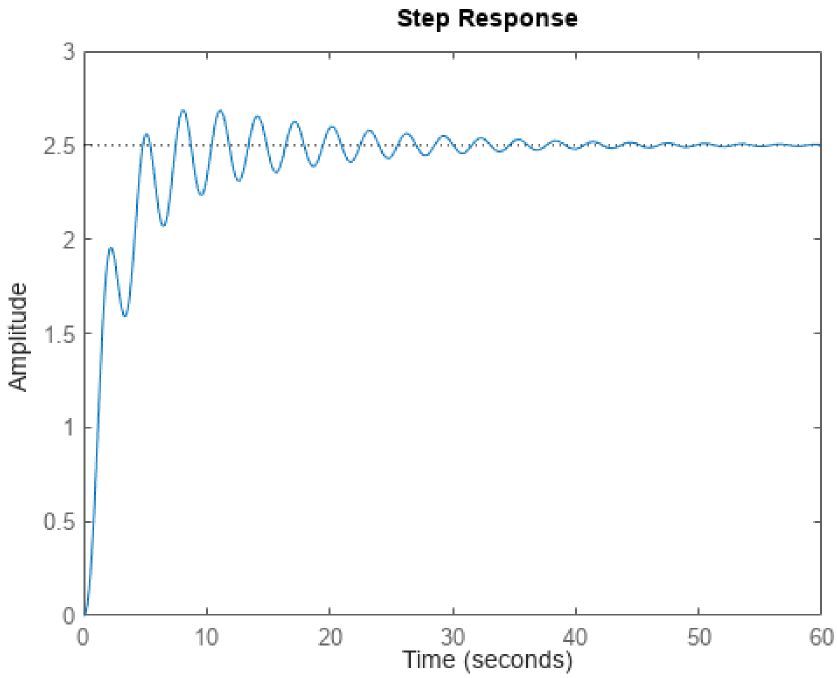
Step-Response Characteristics of Dynamic System

Compute step-response characteristics, such as rise time, settling time, and overshoot, for a dynamic system model. For this example, use a continuous-time transfer function:

$$sys = \frac{s^2 + 5s + 5}{s^4 + 1.65s^3 + 5s^2 + 6.5s + 2}$$

Create the transfer function and examine its step response.

```
sys = tf([1 5 5],[1 1.65 5 6.5 2]);
step(sys)
```



The plot shows that the response rises in a few seconds, and then rings down to a steady-state value of about 2.5. Compute the characteristics of this response using `stepinfo`.

```
S = stepinfo(sys)
S = struct with fields:
RiseTime: 3.8456
TransientTime: 27.9762
SettlingTime: 27.9762
SettlingMin: 2.0689
SettlingMax: 2.6873
Overshoot: 7.4915
Undershoot: 0
```

```
Peak: 2.6873

PeakTime: 8.0530
```

Here, the function uses y_{init} = 0 to compute characteristics for the dynamic system model `sys`.

By default, the settling time is the time it takes for the error to stay below 2% of $|y_{init} - y_{final}|$. The result `S.SettlingTime` shows that for sys, this condition occurs after about 28 seconds. The default definition of rise time is the time it takes for the response to go from 10% to 90% of the way from y_{init} = 0 to y_{final}. `S.RiseTime` shows that for sys, this rise occurs in less than 4 seconds. The maximum overshoot is returned in `S.Overshoot`. For this system, the peak value `S.Peak`, which occurs at the time `S.PeakTime`, overshoots by about 7.5% of the steady-state value.