

EME172 Discussion 3

Second Order System

(1) Relationship between transfer function and damping ratio/natural frequency

From the denominator of the transfer function for a second order system, we can find the system's damping ratio and natural frequency. From the obtained damping ratio, we can determine the nature of the system's response.

The general denominator of a second order transfer function is $s^2 + 2\zeta\omega_n s + \omega_n^2$

If $\zeta > 1$, the response is overdamped.

If $\zeta = 1$, the response is critically damped.

If $\zeta < 1$, the response is under damped.

If $\zeta = 0$, the response is undamped.

Example 1:

Find the damping ratio and natural frequency of the second order system shown below. Determine the nature of the system's response.

$$H(s) = \frac{9}{s^2 + 2s + 9}$$

Sol:

$$\omega_n^2 = 9, \quad \omega_n = 3$$

$$2\zeta\omega_n = 2, \quad \zeta = 0.33, \quad \text{The response is under damped.}$$

(2) Relationship between damping ratio/natural frequency and settling time/ peak time/ rise time/ percent overshoot

Listed below are equations that show the relationship between damping ratio/natural frequency and settling time/peak time/rise time/percent overshoot.

$$T_s = \frac{4}{\zeta\omega_n}$$

$$T_p = \frac{\pi}{\omega_n \sqrt{1 - \zeta^2}}$$

$$\omega_n T_r = 1.76\zeta^3 - 0.417\zeta^2 + 1.039\zeta + 1$$

$$\%OS = e^{-\zeta\pi / \sqrt{1 - \zeta^2}} \times 100$$

Example 2:

Find the settling time, peak time, rise time, percent overshoot of the second order system shown below.

$$H(s) = \frac{9}{s^2 + 2s + 9}$$

Sol:

From example 1, we know $\omega_n = 3$, $\zeta = 0.33$.

Use the above equations, we can obtain $T_s = 4$, $T_p = 1.11$, $T_r = 0.45$, $\%OS = 32.9$.

(3) Relationship between poles and percent overshoot/ settling time/ peak time

Listed below are equations that show the relationship between poles and percent overshoot/settling time/peak time.

$$\text{poles} = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2}$$

$$\zeta = \frac{-\ln\left(\frac{\%OS}{100}\right)}{\sqrt{\pi^2 + \ln^2\left(\frac{\%OS}{100}\right)}}$$

$$\omega_n = \frac{4}{\zeta T_s}$$

$$\omega_n = \frac{\pi}{T_p\sqrt{1-\zeta^2}}$$

Example 3:

Find the poles of a second order system whose $T_s = 4$ and $T_p = 1.11$.

Sol:

Use the above equations, we can obtain $\zeta\omega_n = 1$, $\omega_n\sqrt{1-\zeta^2} = 2.83$.

Therefore poles are $-1 + j2.83$, $-1 - j2.83$.

Ch Control Toolkit Examples

```
int step(class CPlot *plot, array double &yout,  
         array double &tout,  
         array double &xout,  
         /* double tf */);
```

- Calculate and plot step response of a system
- *plot* Pointer to an existing object of class CPlot
- *yout* Array of reference containing the output of the step response
- *tout* Array of reference containing the time vector for the simulation
- *xout* Array of reference containing the state trajectories
- *tf* Double value specifying the final time of the simulation

```
int conv(array double &den, array double &den1, array double &den2);
```

- Convolution and polynomial multiplication
- If *den1* and *den2* are arrays of reference containing polynomial coefficients, convolving them is equivalent to multiplying the two polynomials.
- *den* Array of reference containing the output of the convolution
- *den1* Array of reference containing the first input vector
- *den2* Array of reference containing the second input vector

Example 4 -- System with Denominator Consisting of Single Polynomial :

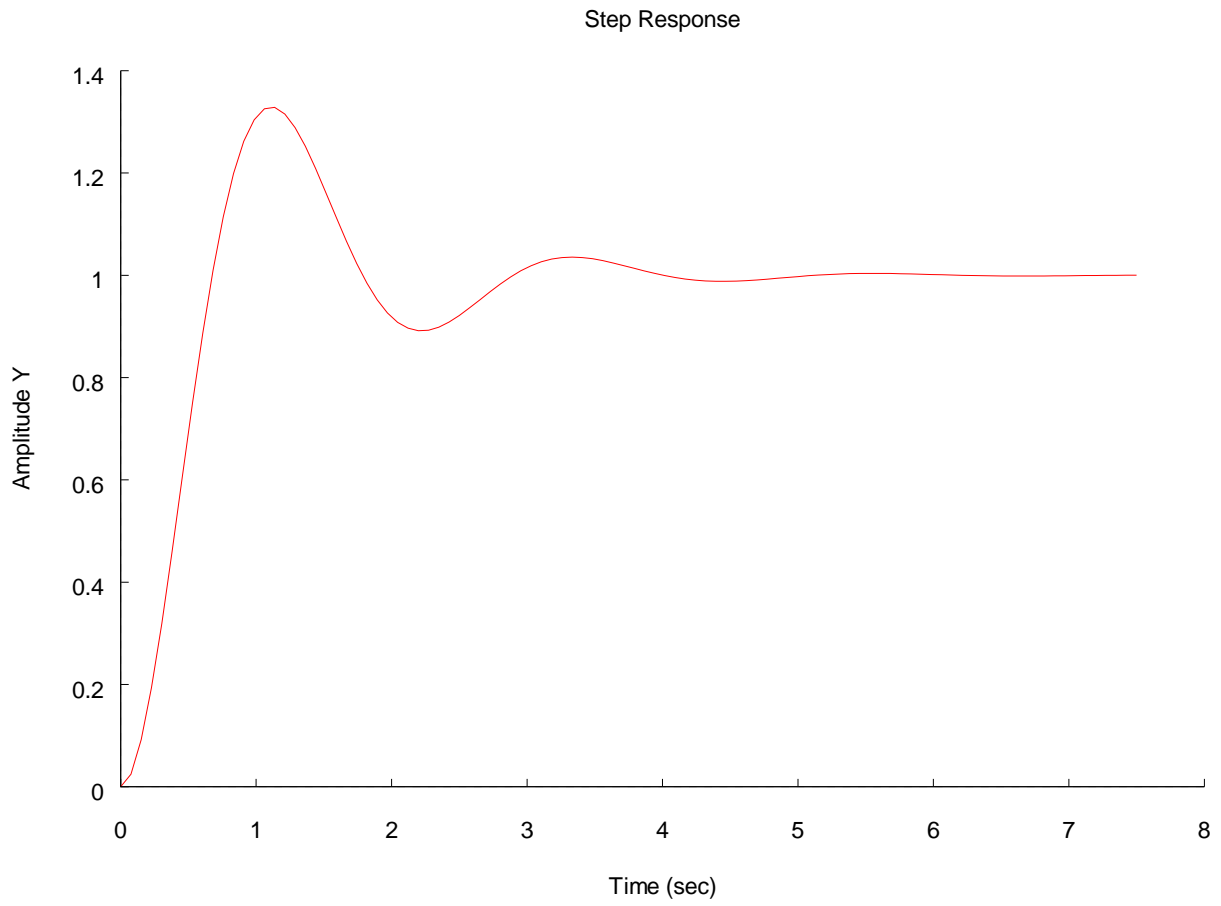
Plot the step response of the following system.

$$H(s) = \frac{9}{s^2 + 2s + 9}$$

Program:

```
/* example4.ch */  
#include <control.h>  
  
int main() {  
    array double num[1] = {9},  
                den[3] = {1, 2, 9};  
    CControl sys;  
    CPlot plot;  
  
    sys.model("tf", num, den);  
    sys.step(&plot, NULL, NULL, NULL);  
  
    return 0;  
}
```

Output :



Example 5 -- System with Denominator Consisting of Multiple Polynomials :

Plot the step response of the following system.

$$H(s) = \frac{9}{(s+3)(s^2+2s+9)}$$

Program:

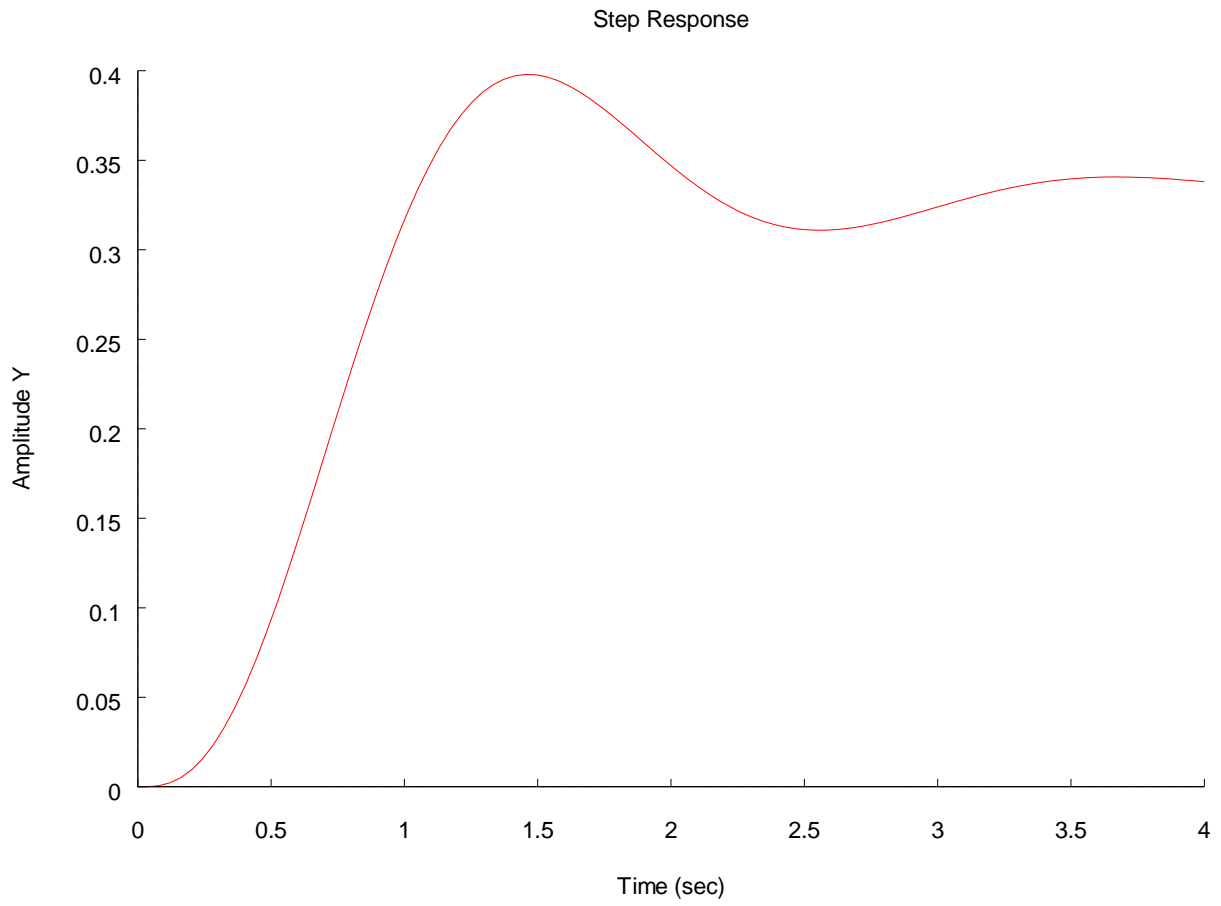
```
/* example5.ch */
#include <control.h>

int main() {
    array double num[1] = {9},
               den1[2] = {1, 3},
               den2[3] = {1, 2, 9},
               den[4] = {1, 2, 9};
    CControl sys;
    CPlot plot;

    conv(den, den1, den2);
    sys.model("tf", num, den);
    sys.step(&plot, NULL, NULL, NULL);

    return 0;
}
```

Output :



Example 6 -- Default Step Response :

Plot the step response of the following system.

$$H(s) = \frac{0.03}{s^2 + 0.03s + 0.06}$$

Program:

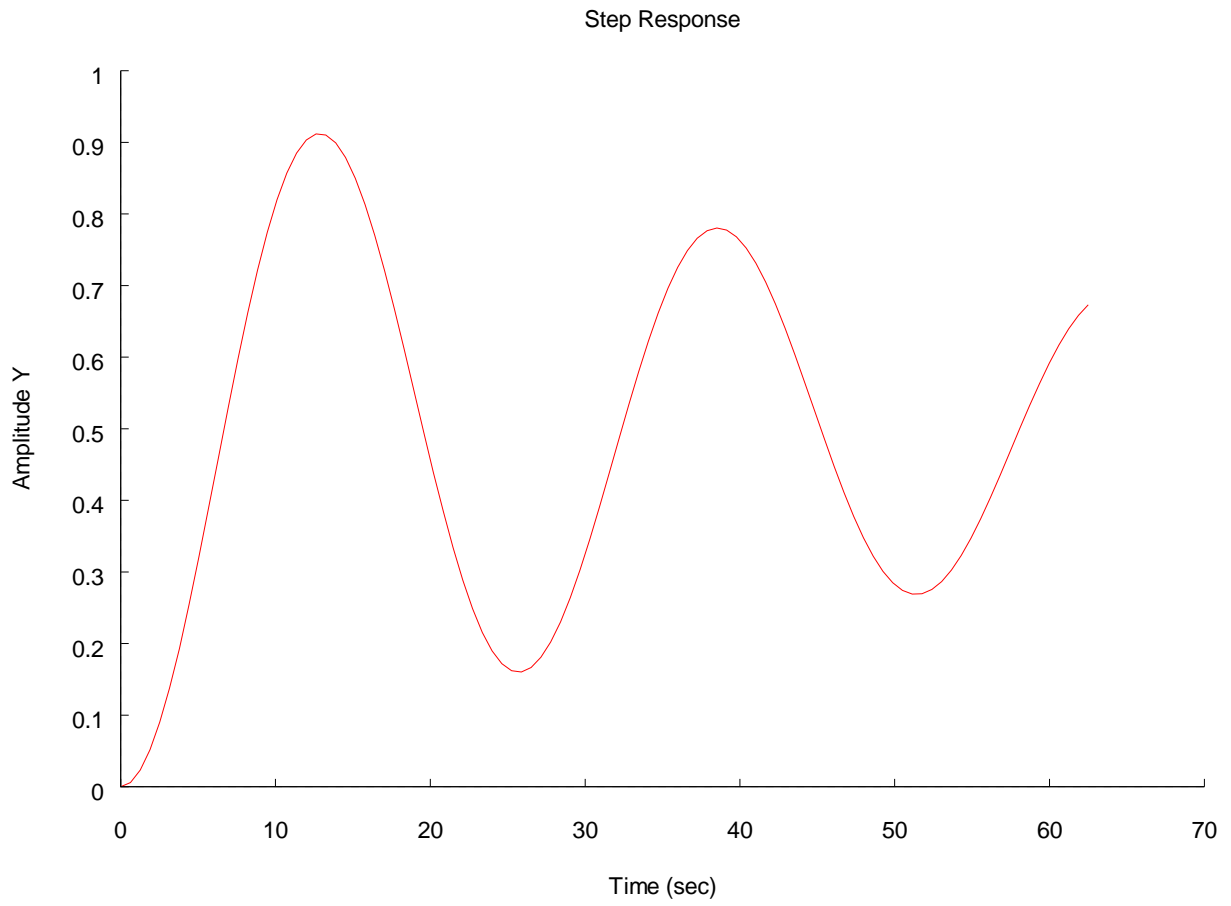
```
/* example6.ch */
#include <control.h>

int main() {
    array double num[1] = {0.03},
                den[3] = {1, 0.03, 0.06};
    CControl sys;
    CPlot plot;

    sys.model("tf", num, den);
    sys.step(&plot, NULL, NULL, NULL);

    return 0;
}
```

Output :



Example 7 -- Step Response with Simulation Time Specified :

Plot the step response of the following system.

$$H(s) = \frac{0.03}{s^2 + 0.03s + 0.06}$$

Program:

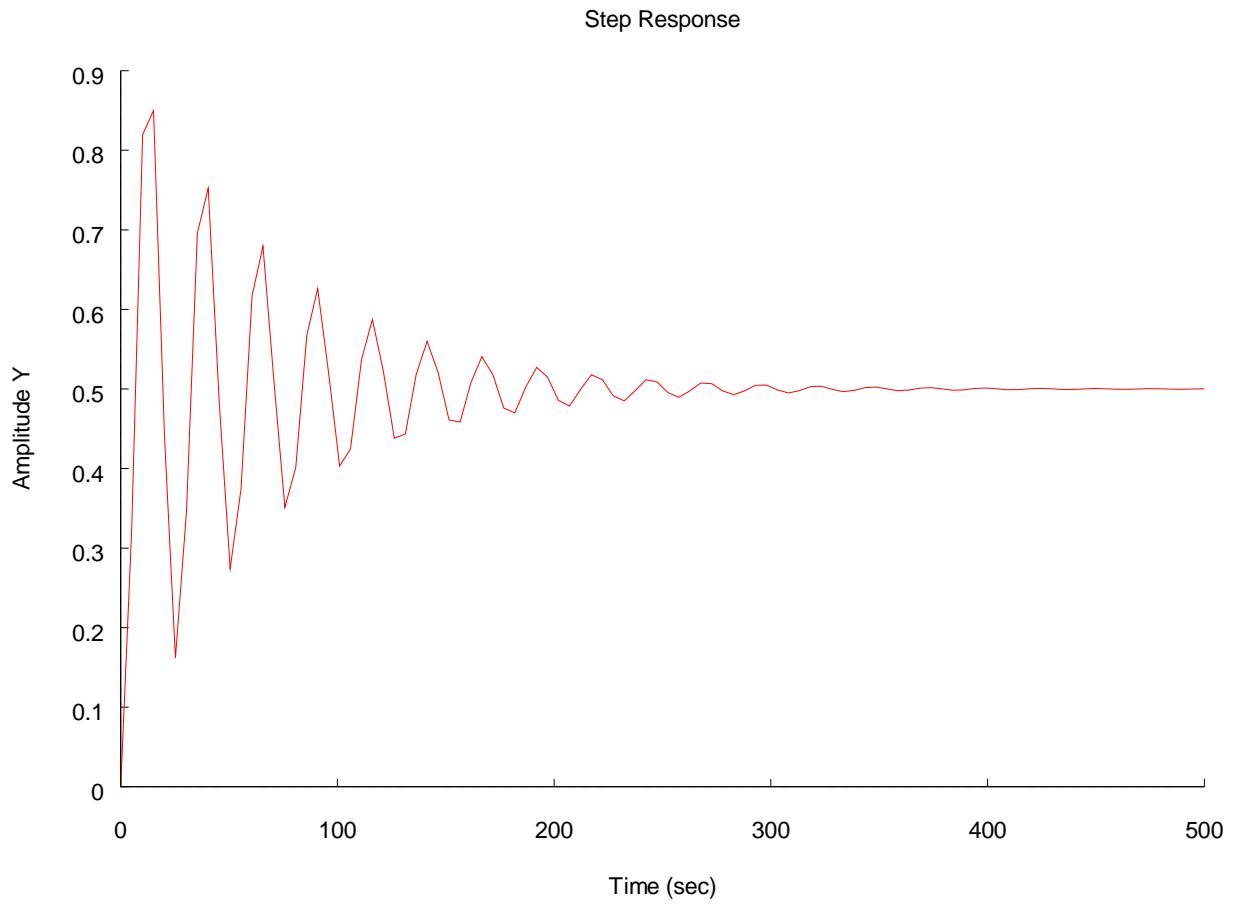
```
/* example7.ch */
#include <control.h>

int main() {
    array double num[1] = {0.03},
               den[3] = {1, 0.03, 0.06};
    double tf = 500;
    CControl sys;
    CPlot plot;

    sys.model("tf", num, den);
    sys.step(&plot, NULL, NULL, NULL, tf);

    return 0;
}
```

Output :



Example 8 -- Step Response with Simulation Time and Number of Points Specified :

Plot the step response of the following system.

$$H(s) = \frac{0.03}{s^2 + 0.03s + 0.06}$$

Program:

```
/* example8.ch */
#include <control.h>

int main() {
    array double num[1] = {0.03},
                den[3] = {1, 0.03, 0.06},
                yout[500], tout[500];
    double tf = 500;
    CControl sys;
    CPlot plot;

    sys.model("tf", num, den);
    sys.step(&plot, yout, tout, NULL, tf);

    return 0;
}
```

Output :

