

3.32 In aircraft control systems, an ideal pitch response (q_o) versus a pitch command (q_c) is described by the transfer function

$$\frac{Q_o(s)}{Q_c(s)} = \frac{\tau\omega_n^2(s+1/\tau)}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

The actual aircraft response is more complicated than this ideal transfer function; nevertheless, the ideal model is used as a guide for autopilot design. Assume that t_r is the desired rise time, and that

$$\omega_n = \frac{1.789}{t_r}$$

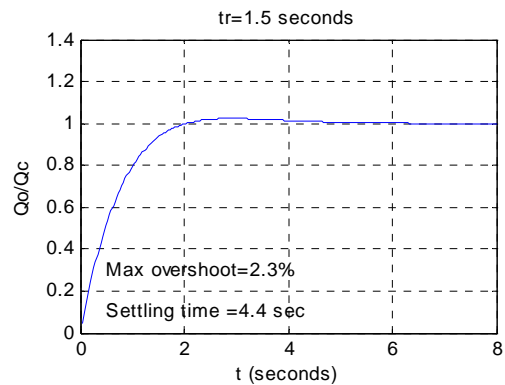
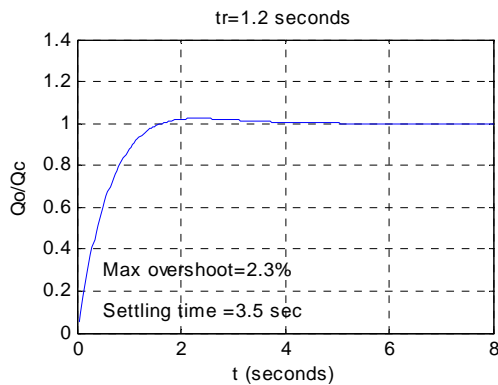
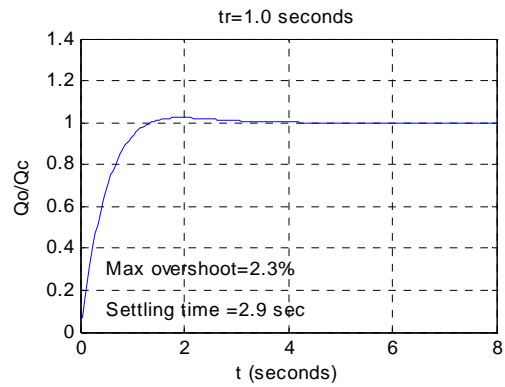
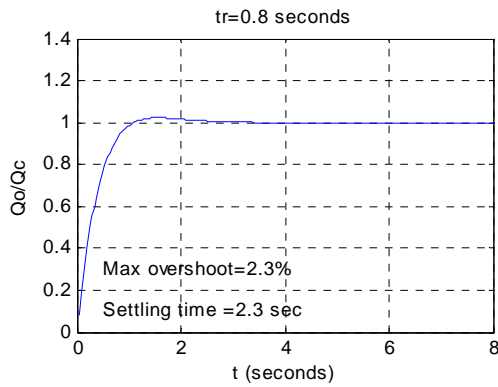
$$\frac{1}{\tau} = \frac{1.6}{t_r}$$

$$\zeta = 0.89$$

Show that this ideal response possesses a fast settling time and minimal overshoot by plotting the step response for $t_r = 0.8, 1.0, 1.2,$ and 1.5 sec.

Solution: The following program statements in MATLAB produce the following plots:

```
% Problem 3.32
tr = [0.8 1.0 1.2 1.5];
t=[1:240]/30;
tback=fliplr(t);
clf;
for I=1:4,
    wn=(1.789)/tr(I); % Rads/second
    tau=tr(I)/(1.6); % tau
    zeta=0.89; %
    b=tau*(wn^2)*[1 1/tau];
    a=[1 2*zeta*wn (wn^2)];
    y=step(b,a,t);
    subplot(2,2,I);
    plot(t,y);
    titletext=sprintf('tr=%3.1f seconds',tr(I));
    title(titletext);
    xlabel('t (seconds)');
    ylabel('Qo/Qc');
    ymax=(max(y)-1)*100;
    msg=sprintf('Max overshoot=%3.1f%%',ymax);
    text(.50,.30,msg);
    yback=flipud(y);
    yind=find(abs(yback-1)>0.01);
    ts=tback(min(yind));
    msg=sprintf('Settling time =%3.1f sec',ts);
    text(.50,.10,msg);
    grid;
end
```



3.35 Consider the two nonminimum phase systems,

$$G_1(s) = -\frac{2(s-1)}{(s+1)(s+2)}$$

$$G_2(s) = -\frac{3(s-1)(s-2)}{(s+1)(s+2)(s+3)}$$

- Sketch the unit step responses for $G_1(s)$ and $G_2(s)$, paying close attention to the transient part of the response.
- Explain the difference in the behavior of the two responses as it relates to the zero locations.
- Consider a stable, strictly proper system (that is, m zeros and n poles, where $m < n$).

Let $y(t)$ denote the step response of the system. The step response is said to have an undershoot if it initially starts off in the "wrong" direction. Prove that a stable, strictly proper system has an undershoot if and only if its transfer function has an odd number of real RHP zeros.

Solution: (a) For $G_1(s)$:

$$Y_1(s) = \frac{1}{s} G_1(s) = -\frac{2(s-1)}{s(s+1)(s+2)}$$

$$H(s) = k \frac{\prod_j (s - z_j)}{\prod_l (s - p_l)}$$

$$R_{p_i} = \lim_{s \rightarrow p_i} [(s - p_i)H(s)] = \lim_{s \rightarrow p_i} k \frac{\prod_j (s - z_j)}{\prod_{l \neq i} (s - p_l)} = k \frac{\prod_j (p_i - z_j)}{\prod_{l \neq i} (p_i - p_l)}$$

Each factor $(p_i - z_j)$ or $(p_i - p_l)$ can be thought of as a complex number (a magnitude and a phase) whose pictorial representation is a vector pointing to p_i and coming from z_j or p_l respectively.

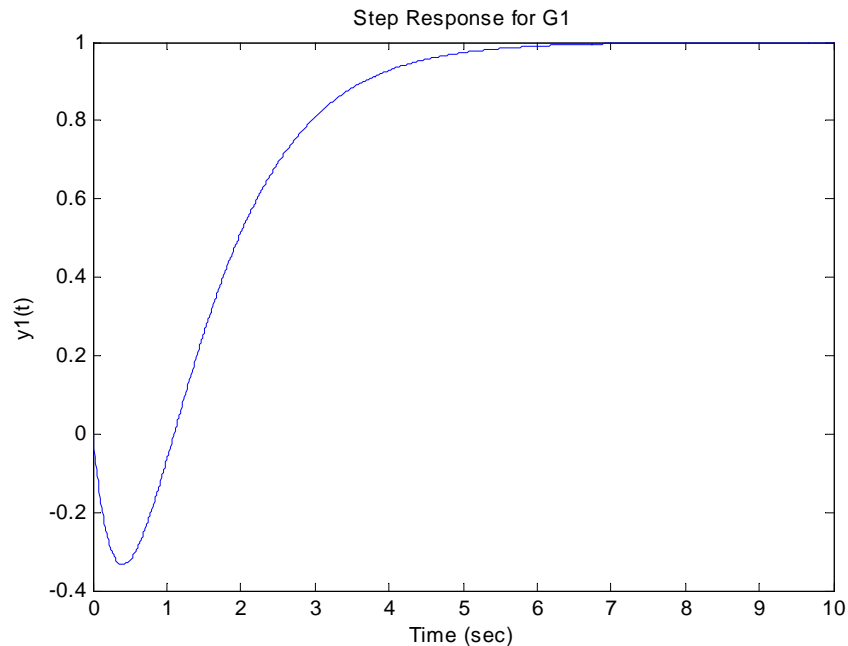
The method for calculating the residue at a pole p_i is:

- (1) Draw vectors from the rest of the poles and from all the zeros to the pole p_i .
- (2) Measure magnitude and phase of these vectors.
- (3) The residue will be equal to the gain, multiplied by the product of the vectors coming from the zeros and divided by the product of the vectors coming from the poles.

In our problem:

$$Y_1(s) = -\frac{2(s-1)}{s(s+1)(s+2)} = \frac{R_0}{s} + \frac{R_{-1}}{s+1} + \frac{R_{-2}}{s+2} = \frac{1}{s} - \frac{4}{s+1} + \frac{3}{s+2}$$

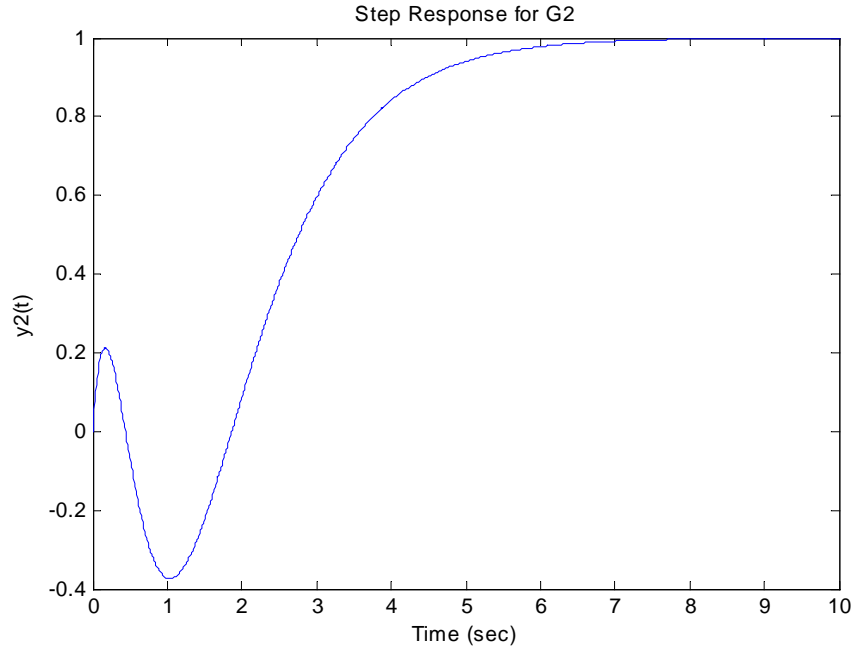
$$y_1(t) = 1 - 4e^{-t} + 3e^{-2t}$$



For $G_2(s)$:

$$Y_2(s) = \frac{3(s-1)(s-2)}{s(s+1)(s+2)(s+3)} = \frac{1}{s} + \frac{-9}{s+1} + \frac{18}{s+2} + \frac{-10}{s+3}$$

$$y_2(t) = 1 - 9e^{-t} + 18e^{-2t} - 10e^{-3t}$$



- (b) The first system presents an "undershoot". The second system, on the other hand, starts off in the right direction.

The reasons for this initial behavior of the step response will be analyzed in part c.

In $y_1(t)$: dominant at $t = 0$ the term $-4e^{-t}$

In $y_2(t)$: dominant at $t = 0$ the term $18e^{-2t}$

- (c) The following concise proof is from [1] (see also [2]-[3]).

Without loss of generality assume the system has unity DC gain ($G(0) = 1$). Since the system is stable, $y(\infty) = G(0) = 1$, and it is reasonable to assume $y(\infty) \neq 0$.

Let us denote the pole-zero excess as $r = n - m$. Then, $y(t)$ and its $r - 1$ derivatives are zero at $t = 0$, and $y^r(0)$ is the first non-zero derivative. The system has an undershoot if $y^r(0)y(\infty) < 0$. The transfer function may be re-written as

$$G(s) = \frac{\prod_{i=1}^m (1 - \frac{s}{z_i})}{\prod_{i=1}^{m+r} (1 - \frac{s}{p_i})}$$

The numerator terms can be classified into three types of terms:

- (1). The first group of terms are of the form $(1 - \alpha_i s)$ with $\alpha_i > 0$.
- (2). The second group of terms are of the form $(1 + \alpha_i s)$ with $\alpha_i > 0$.

(3). Finally, the third group of terms are of the form, $(1 + \beta_i s + \alpha_i s^2)$ with $\alpha_i > 0$, and β_i could be negative.

However, $\beta_i^2 < 4\alpha_i$, so that the corresponding zeros are complex.

All the denominator terms are of the form (2), (3), above. Since,

$$y^r(0) = \lim_{s \rightarrow \infty} s^r G(s)$$

it is seen that the sign of $y^r(0)$ is determined entirely by the number of terms of group 3 above. In particular, if the number is odd, then $y^r(0)$ is negative and if it is even, then $y^r(0)$ is positive. Since $y(\infty) = G(0) = 1$, then we have the desired result.

[1] Vidyasagar, M., "On Undershoot and Nonminimum Phase Zeros," **IEEE Trans. Automat. Contr.**, Vol. AC-31, p. 440, May 1986.

[2] Clark, R., N., **Introduction to Automatic Control Systems**, John Wiley, 1962.

[3] Mita, T. and H. Yoshida, "Undershooting phenomenon and its control in linear multivariable servomechanisms," **IEEE Trans. Automat. Contr.**, Vol. AC-26, pp. 402-407, 1981.

3.39 Suppose that unity feedback is to be applied around the following open-loop systems. Use Routh's stability criterion to determine whether the resulting closed-loop systems will be stable.

$$(a) KG(s) = \frac{4(s+2)}{s(s^3 + 2s^2 + 3s + 4)}$$

$$(b) KG(s) = \frac{2(s+4)}{s^2(s+1)}$$

$$(c) KG(s) = \frac{4(s^3 + 2s^2 + s + 1)}{s^2(s^3 + 2s^2 - s - 1)}$$

Solution:

$$(a) T(s) = \frac{KG(s)}{1 + KG(s)} = \frac{4(s+2)}{s^4 + 2s^3 + 3s^2 + 8s + 8}, \quad a(s) = s^4 + 2s^3 + 3s^2 + 8s + 8$$

$$s^4: \quad 1 \quad 3 \quad 8$$

$$s^3: \quad 2 \quad 8$$

$$s^2: \quad a \quad b$$

$$s^1: \quad c$$

$$s^0: \quad d$$

$$\text{where } a = -\frac{8 \times 1 - 2 \times 3}{2} = -1, \quad b = -\frac{1 \times 0 - 2 \times 8}{2} = 8,$$

$$c = -\frac{2b - 8a}{a} = 24, \quad d = b = 8$$

2 sign changes in first column \Rightarrow 2 roots not in LHP \Rightarrow unstable.

$$(b) T(s) = \frac{KG(s)}{1+KG(s)} = \frac{2(s+4)}{s^3 + s^2 + 2s + 8}, \quad a(s) = s^3 + s^2 + 2s + 8 = 0$$

The Routh's array is,

$$\begin{array}{l} s^3 : 1 \quad 2 \\ s^2 : 1 \quad 8 \\ s^1 : -6 \\ s^0 : 8 \end{array}$$

There are two sign changes in the first column of the Routh array. Therefore, there are two roots not in the LHP.

3.40 Use Routh's stability criterion to determine how many roots with positive real parts the following equations have.

$$(b) s^5 + 10s^4 + 30s^3 + 80s^2 + 344s + 480 = 0$$

$$(d) s^3 + s^2 + 20s + 78 = 0$$

Solution:

$$(b) s^5 + 10s^4 + 30s^3 + 80s^2 + 344s + 480 = 0$$

$$\begin{array}{l} s^5 : 1 \quad 30 \quad 344 \\ s^4 : 10 \quad 80 \quad 480 \\ s^3 : 11 \quad 148 \quad \left(\times \frac{1}{2}\right) \\ s^2 : -5 \quad 44 \quad \left(\times \frac{11}{120}\right) \\ s^1 : 1 \quad \left(\times \frac{5}{1224}\right) \\ s^0 : 1 \quad \left(\times \frac{1}{44}\right) \end{array}$$

\Rightarrow 2 roots not in LHP.

$$(d) s^3 + s^2 + 20s + 78 = 0$$

$$\begin{array}{l} s^3 : 1 \quad 20 \\ s^2 : 1 \quad 78 \\ s^1 : -1 \quad \left(\times \frac{1}{58}\right) \\ s^0 : 1 \quad \left(\times \frac{1}{78}\right) \end{array}$$

\Rightarrow 2 roots not in LHP.

3.41 Find the range of K for which all the roots of the following polynomial are in the LHP.

$$s^5 + 5s^4 + 10s^3 + 10s^2 + 5s + K = 0$$

Use MATLAB to verify your answer by plotting the roots of the polynomial in the s-plane for various values of K .

Solution: $s^5 + 5s^4 + 10s^3 + 10s^2 + 5s + K = 0$

$$s^5 : \quad 1 \quad 10 \quad 5$$

$$s^4 : \quad 5 \quad 10 \quad K$$

$$s^3 : \quad a_1 \quad a_2$$

$$s^2 : \quad b_1 \quad K$$

$$s^1 : \quad c_1$$

$$s^0 : \quad K$$

$$\text{Where } a_1 = -\frac{1(10) - 5(10)}{5} = 8, \quad a_2 = -\frac{1(K) - 5(5)}{5} = \frac{25 - K}{5}$$

$$b_1 = -\frac{5a_2 - 10a_1}{a_1} = \frac{55 + K}{8},$$

$$c_1 = -\frac{Ka_1 - a_2b_1}{b_1} = -\frac{K^2 + 350K - 1375}{5(K + 55)},$$

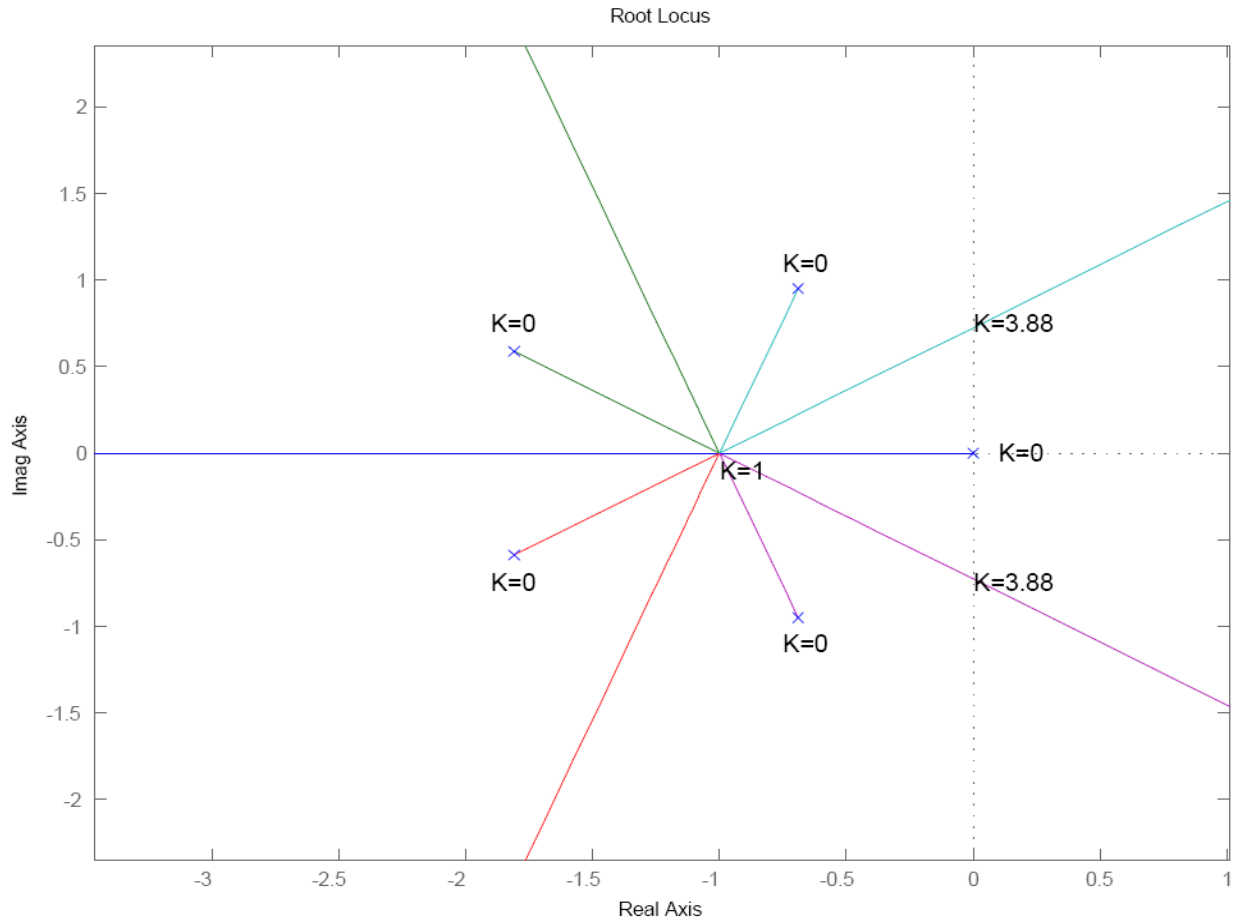
For stability: all terms in first column > 0

$$(1) b_1 = \frac{55 + K}{8} > 0 \Rightarrow K > -55$$

$$(2) c_1 = -\frac{K^2 + 350K - 1375}{5(K + 55)} > 0, \quad \frac{(K - 3.89)(K + 354)}{5(K + 55)} < 0 \Rightarrow -55 < K < 3.89$$

$$(3) K > 0$$

Combining (1), (2), and (3) $\Rightarrow 0 < K < 3.89$. If we plot the roots of the polynomial for various values of K we obtain the following root locus plot (see Chapter 5),



A: Use Matlab to compute the overshoot, rise time, settling time (with respect to step response) for system $H(s) = \frac{(s + 0.5\alpha)}{0.5\alpha(s^2 + s + 1)}$ when $\alpha = 10, 4, 2, 1$, respectively. Plot the time response for each case and compare the results. Choose the final time as 15 seconds. Be careful that when α is small, the step response may cross the line $y=0.9$ several times.

Solution:

```
% Problem 5.A
linespec=['r','g','b','k'];
alpha = [10 4 2 1];
mp=[]; ts=[]; tr=[];
t=[0:0.01:15];
clf;
for i=1:4
    b=[1 0.5*alpha(i)];
    a=0.5*alpha(i)*[1 1 1];
    y=step(b,a,t);
    plot(t,y,linespec(i));
    hold on;
% Overshoot
```



```

mp=[mp (max(y)-1)*100];
% Rising Time
yind=find(y>0.1);
t1=t(min(yind));
yind=find(y>0.9);
t2=t(min(yind));
tr=[tr t2-t1];
% Settling Time
yind=find(abs(y-1)>0.01);
ts=[ts t(max(yind))];
end
xlabel('t (seconds)');
ylabel('Step Response y(t)');
legend(sprintf('\alpha = %d, mp = %3.1f%%, tr = %3.1f, ts = %3.1f', alpha(1), mp(1), tr(1),
ts(1)),...
sprintf('\alpha = %d, mp = %3.1f%%, tr = %3.1f, ts = %3.1f', alpha(2), mp(2), tr(2), ts(2)),...
sprintf('\alpha = %d, mp = %3.1f%%, tr = %3.1f, ts = %3.1f', alpha(3), mp(3), tr(3), ts(3)),...
sprintf('\alpha = %d, mp = %3.1f%%, tr = %3.1f, ts = %3.1f', alpha(4), mp(4), tr(4), ts(4)));
grid;

```

