ECE5410 / Fall 2017: Exam 1

1.

Solve the roots of the numerator and denominator of the transfer function in Mathematica. The code and results are shown as follow.

```
In[84]:= Solve[z^2 - 0.65 z - 1.57 == 0, z]
Out[84]= {{z → -0.969459}, {z → 1.61946}}

In[85]:= Solve[z^4 - 1.7 z^3 + 0.75 z^2 + 0.031 z - 0.039 == 0, z]
Out[85]= {{z → -0.2}, {z → 0.3}, {z → 0.8 - 0.1 i}, {z → 0.8 + 0.1 i}}

In[88]:= Abs[0.8 + 0.1 I]
Out[88]= 0.806226
```

Figure 1 Roots Results

(a) is true, because the absolute value of roots of the denominator are all less than 1, meaning that all poles are in unit circle. Therefore, it is stable. (In85,88)

(b) False. Because the numerator has a root 1.61946 greater than 1, meaning that the system has a zero outside the unit circle, so it is not minimum-phase.

(c) False. The system is 4^{th} order as the order of the denominator is 4.

(d) True. The DC gain is 20.33. See the writing script.

1. (d)

$$H(z) = \frac{Y(z)}{U(z)}.$$

$$\text{DC gain} = H(e^{j\omega})\Big|_{\omega=0} = H(1)$$

$$= \frac{0.7(1 - 0.65 - 1.57)}{1 - 1.7 + 0.75 + 0.031 - 0.039} = -20.33$$

Figure 2 Writing Script of 1(d)

(e) False. See the root of denominator above (Fig 1 In[85]), the poles on real axis are -0.2, 0.3, not close to 0.6.

(f) False. The relative degree is the difference of the order of denominator and numerator, which is 4 - 2 = 2.

(g) True. Use dstep in Matlab to plot the step response of the system. The result is shown as followed. From Fig 3 we can see the step response goes up to positive first and then drops to negative.

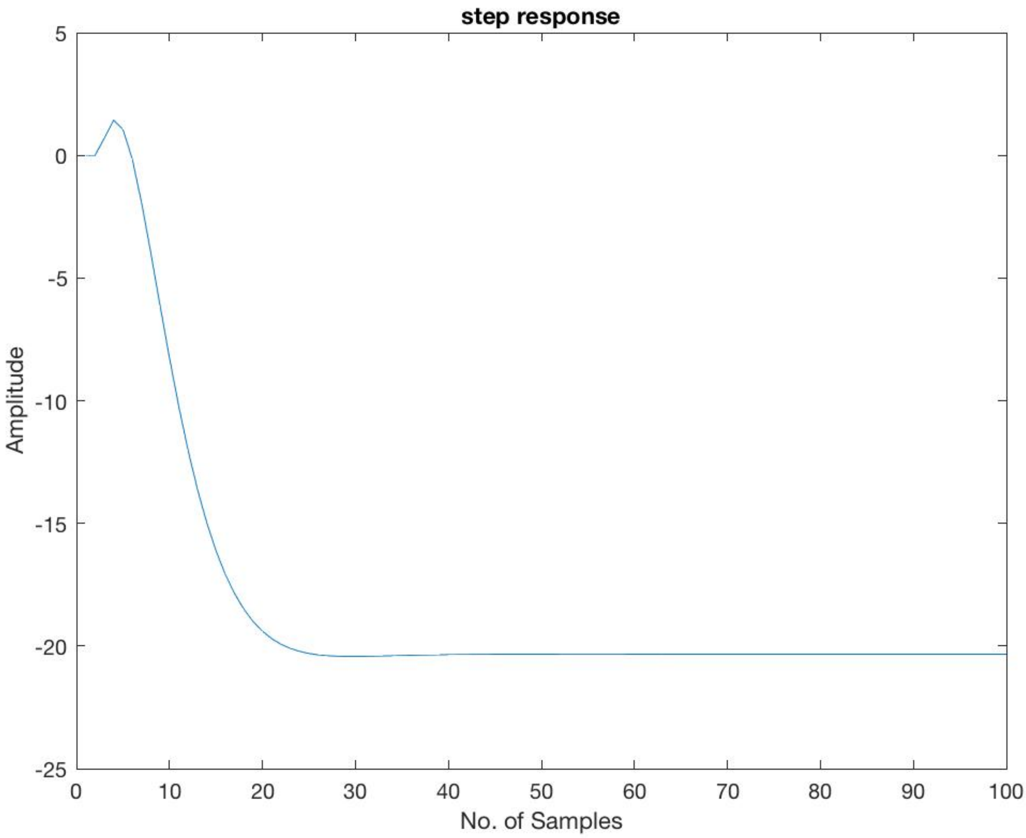


Figure 3 1(g)

(h) False. Use freqz in Matlab the plot the frequency response of the system. The response is plotted below. As is shown, the phase shift in the frequency of 0.75 radians per sample is about -121.218 degrees, not in the range of -30 to -50 degrees.

```
In[1974]:= H[z_] := 0.7 (z^2 - 0.65 z - 1.57) / (z^4 - 1.7 z^3 + 0.75 z^2 + 0.031 z - 0.039)

In[1979]:= Arg[H[Exp[I 0.75]]] / π * 180

Out[1979]= -121.218
```

(i) True. See the writing script to compute the feedback transfer function G(z).

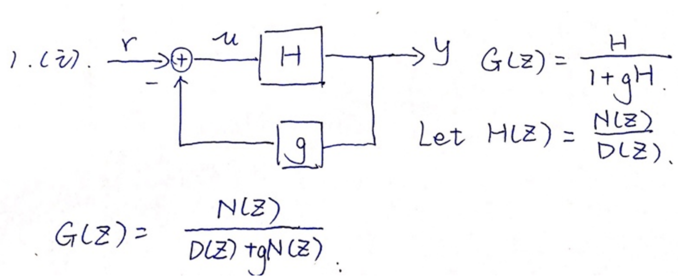


Figure 4 1(i) Computation

Then use root locus to find if there is a g that makes G(z) stable. See (exam1_g.m), the root locus is plotted and one set of poles is selected. Then the step response of the new G(z) is computed. As is shown in one result, if we select

```
selected_point =

   0.5050 + 0.1488i
```

we can get a g

```
gain =

    0.0181
```

And also the poles set

```
poles =

   0.9189 + 0.0000i
   0.5057 + 0.1489i
   0.5057 - 0.1489i
  -0.2304 + 0.0000i
```

This g makes the system stable, as is shown in the poles set (all in unit circle) and the step response below.

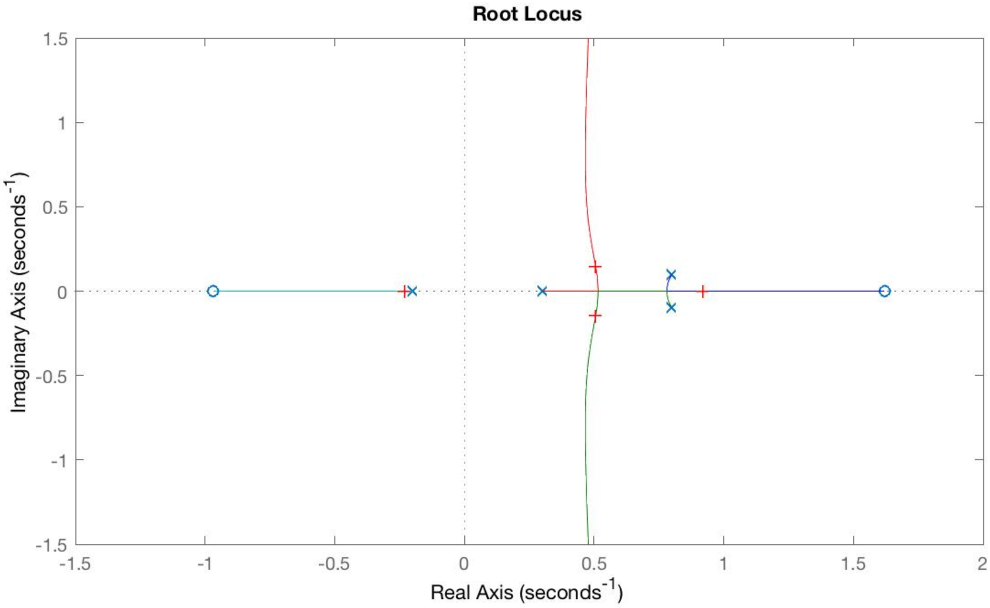


Figure 5 1(i) Selected Root Locus

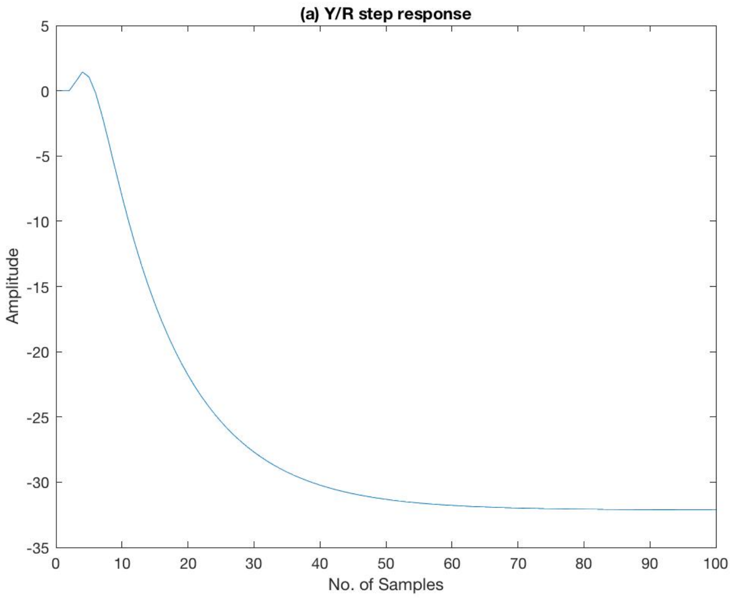


Figure 6 1(i) Stable Step response

(j) True. We do the same experiment in (i) again, except we choose a pole outside the unit circle. The root locus and step response is shown below. One selection is

```
selected_point =

   0.4719 + 1.2000i
```

The corresponding g is

```
gain =

    1.1771
```

The poles set is

```
poles =

   1.3515 + 0.0000i
   0.4709 + 1.2000i
   0.4709 - 1.2000i
  -0.5934 + 0.0000i
```

This selection makes the system unstable, which can be seen in the poles that are outside the unit circle, and the step response plot.

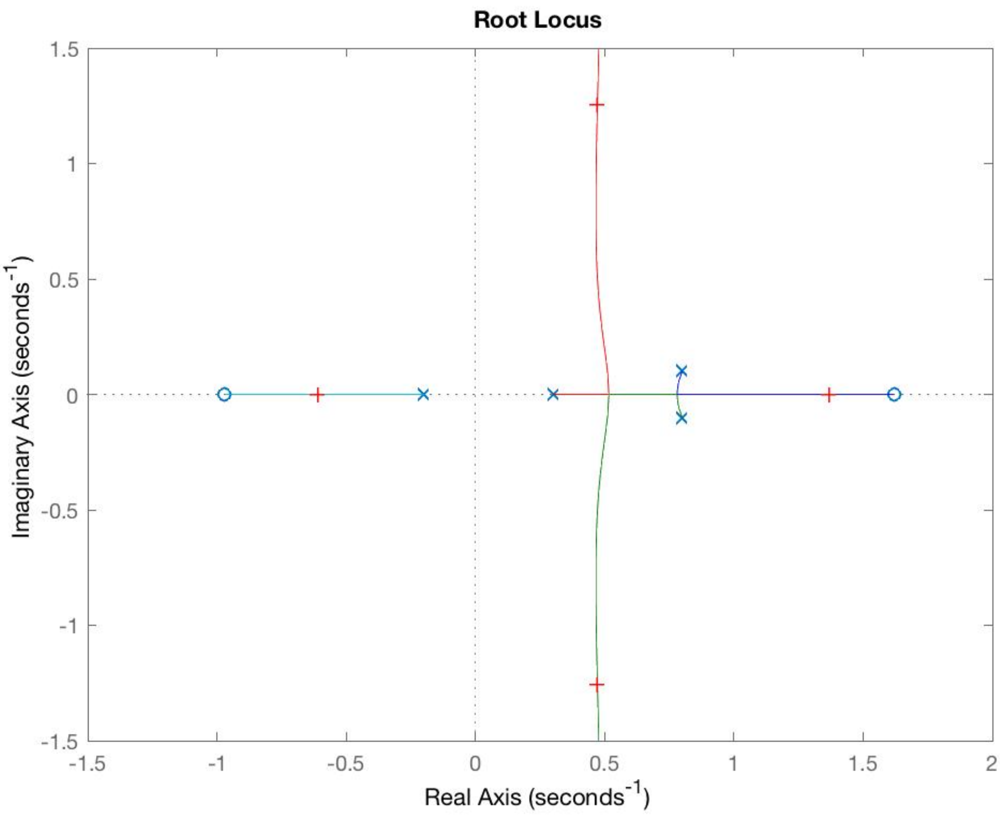


Figure 7 1(j) selected root locus

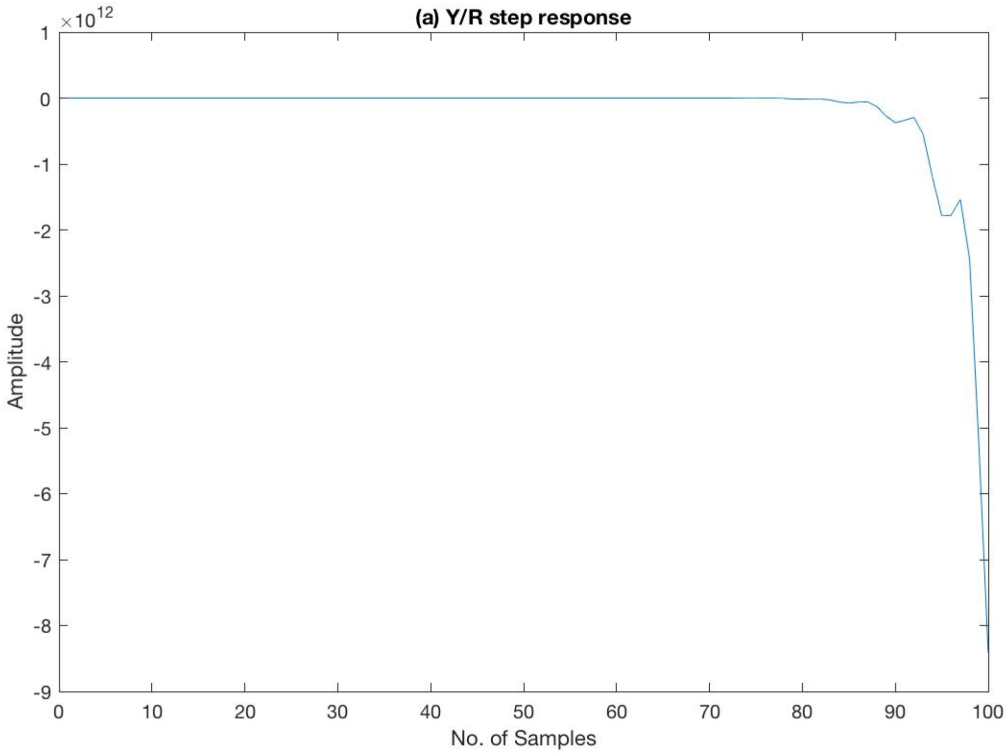


Figure 8 Unstable step response goes infinte.

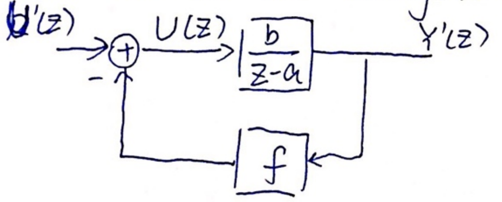
2.

(a)

See the writing script

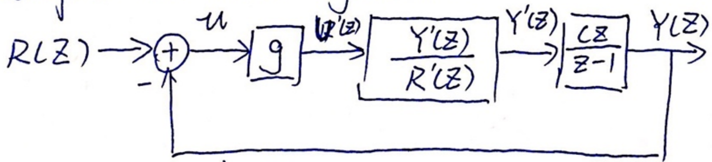
$$G(z) = \frac{N(z)}{D(z) + gN(z)}$$

2. (a) First look at the subsystem. We can compute the $\frac{Y'(z)}{U'(z)}$.



$$\frac{Y'(z)}{U'(z)} = \frac{\frac{b}{z-a}}{1 + f \cdot \frac{b}{z-a}} = \frac{b}{z-a+fb}.$$

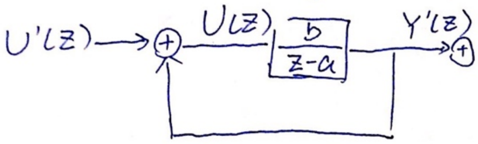
Therefore the whole system can be shown as



$$\therefore \frac{Y(z)}{R(z)} = \frac{\frac{gb \cdot cz}{(z-a+fb)(z-1)}}{1 + \frac{gbcz}{(z-a+fb)(z-1)}} = \frac{gbc \cdot z}{z^2 + (-1-a+fb+gbc)z + a - fb}$$

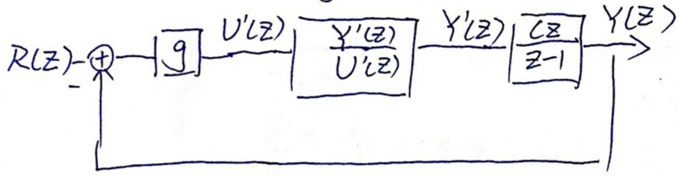
(b) Writing SCRIPT

2(b). From the subsystem we can compute.



$$\frac{U(z)}{U'(z)} = \frac{1}{1 + f\frac{b}{z-a}} = \frac{z-a}{z-a+fb}$$

From the whole system. we can compute



$$\frac{U'(z)}{R(z)} = \frac{g}{1 + \frac{gbcz}{(z-a+fb)(z-1)}}$$

$$\therefore \frac{U(z)}{R(z)} = \frac{U(z)}{U'(z)} \times \frac{U'(z)}{R(z)} = \frac{g[z^2 - (a+1)z + a]}{z^2 + (-1-a+fb+gbc)z + a - fb}.$$

(c) Yes. See the writing script.

2(c). $R \pm jI$ are the wanted poles.

Then the denominator is $(z-R-jI)(z-R+jI)$.

$$= z^2 - 2Rz + R^2 + I^2 = z^2 + (-1-a+fb+gbc)z + a-fb$$

To achieve the poles. we solve g, f, by

$$\begin{cases} -1-a+fb+gbc = -2R \\ a-fb = R^2+I^2 \end{cases}$$

As long as $bc \neq 0$. These equation of f. g ~~is~~ are solvable. meaning that with a set of poles, we can solve f.g to achieve it.

(d) See the writing script.

2(d) $z^2 + (-1-a+fb+gbc)z + a - fb$

$$= z^2 + (-2.2 + 0.4f + 0.28g)z + 1.2 - 0.4f$$

Choose $R = \frac{0.7}{\sqrt{2}}$

$I = \frac{0.7}{\sqrt{2}}$.

$$\begin{cases} 0.4f + 0.28g - 2.2 = -2 \cdot \frac{0.7}{\sqrt{2}} \\ -0.4f + 1.2 = \left(\frac{0.7}{\sqrt{2}}\right)^2 + \left(\frac{0.7}{\sqrt{2}}\right)^2 = 0.49 \end{cases}$$

We can select.
$$\begin{cases} f = \cancel{4.225}\ \cancel{1.775}\ 1.775 \\ g = \cancel{17.428391}\ \cancel{8.85696247}\ 1.785894 \end{cases}$$

The poles are $\left(-\frac{0.7}{\sqrt{2}}, \pm\frac{0.7}{\sqrt{2}}\right)$ on the r=0.7 circle.

Use Matlab to confirm the result. (exam2d.m). Choose the f,g value as above, then pzmap the system, we can see the poles are on the r = 0.7 circle.

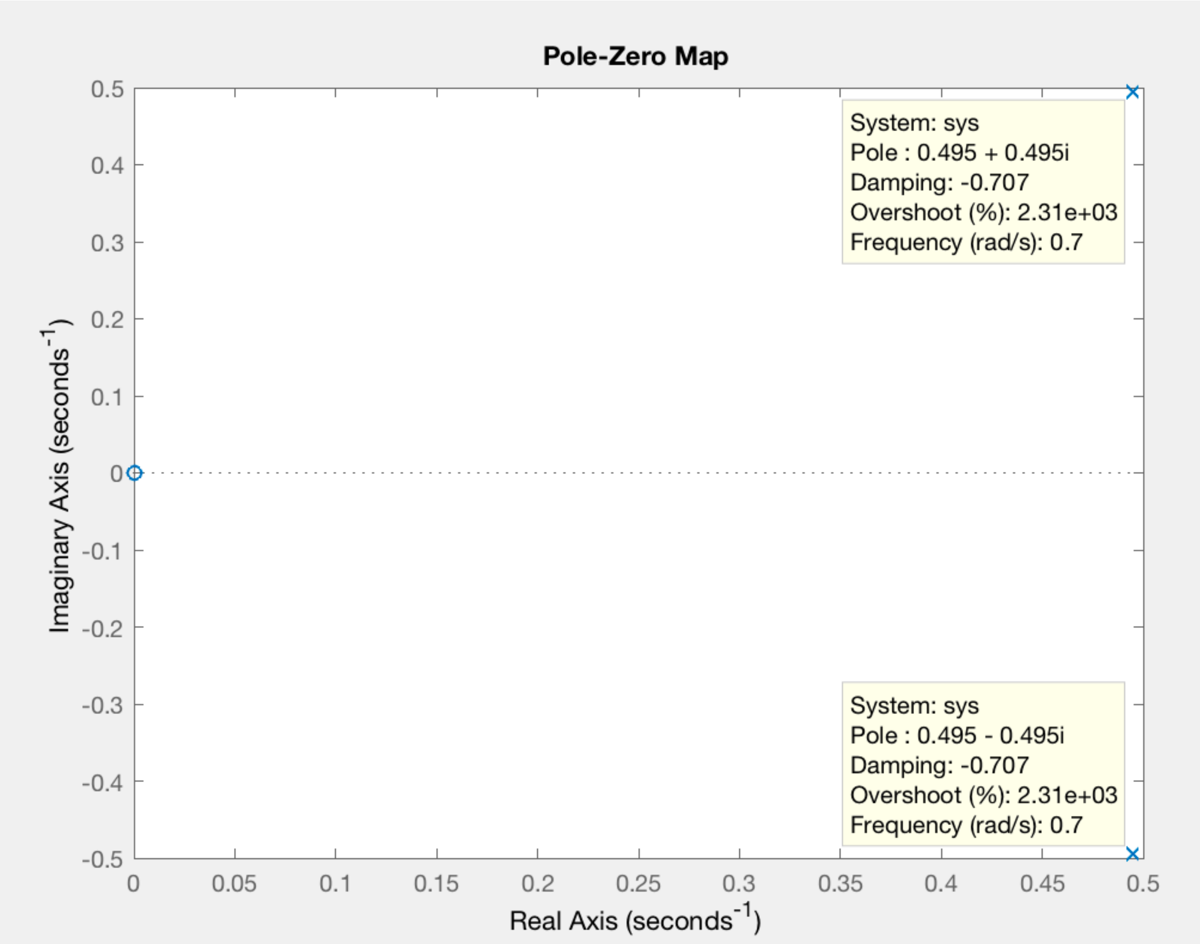


Figure 9 2(d) pzmap on r = 0.7

(e) In exam2d.m, the system is analyzed in step response, tracking error for unit step input and unit slope ramp input. The plots are shown as followed.

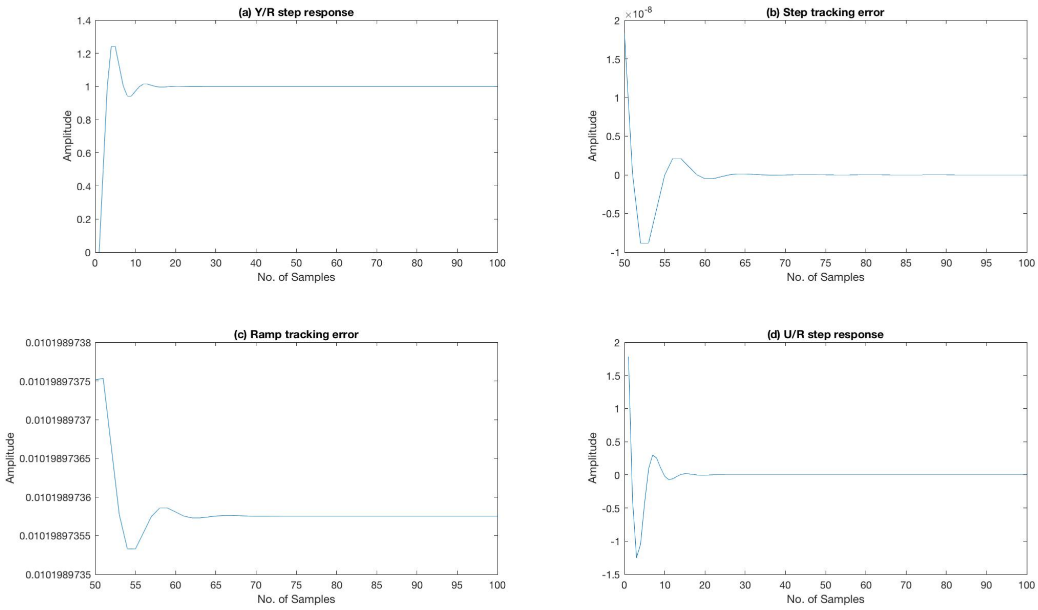


Figure 10 System Response and its Tracking Error

And the percent overshoot is computed as

```
maxy =

    1.2401

persentovershoot =

   24.0100
```

To compute the 2% settling time, we can look backwards in the step response results, to find the “first” one exceeds 0.02.

```
Settling time =

   11
```

We can also find in the matrix that the 10^{th} sample is the last one exceeds 2%. Therefore, the 2% settling time is 11 samples.

5	1.2401
6	1.1200
7	1.0012
8	0.9424
9	0.9424
10	0.9712
11	0.9997
12	1.0138
13	1.0138
14	1.0069
15	1.0001

10	0.9275
11	1.0043
12	1.0490
13	1.0517
14	1.0275
15	0.9990
16	0.9821
17	0.9807
18	0.9895
19	1.0001

Figure 11 Step Response Results (a) 2e; (b) 2f

(f) Use exam2f.m to compute the results, shown as followed.

```
maxy =

    1.3728

persentovershoot =

   37.2760

Settling time =

   15
```

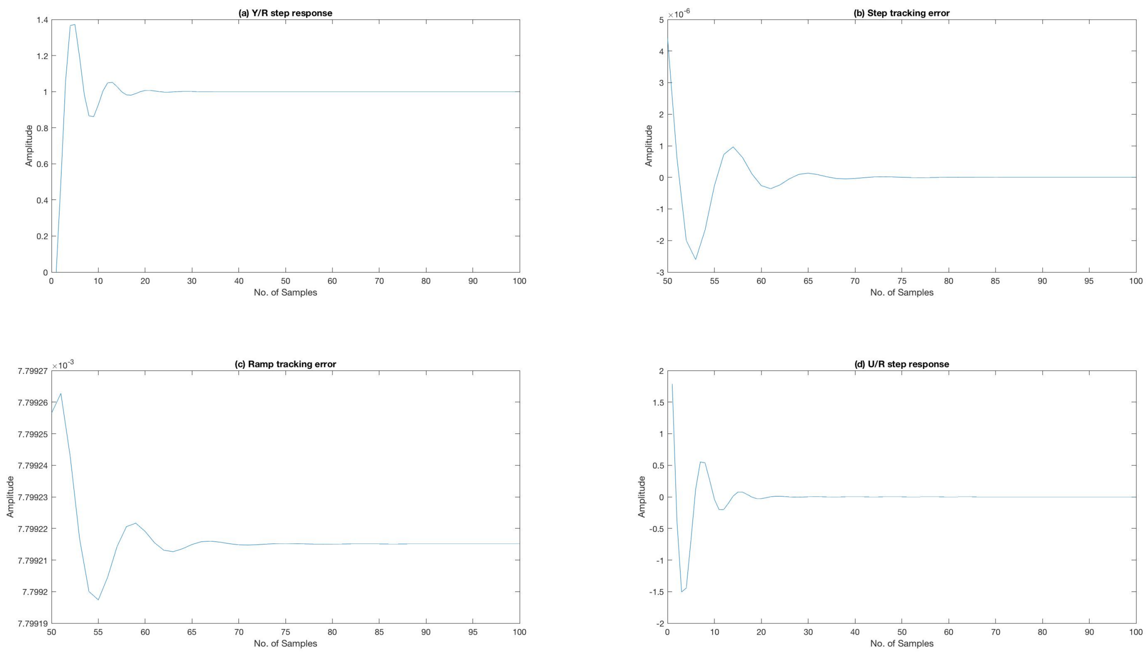


Figure 12 Misspecification System Response and its Tracking Error

Compared with 2e, the misspecificated system has larger overshoot and larger 2% settling time.

But it has smaller ramp tracking error.

3. (a)(b)(c)

3. (a) $y(n) = c^2 u^2(n) - 2ac\, u(n) + a^2 - b$

$$= \begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{pmatrix} \left(u^2(n) \quad u(n) \quad 1 \right)$$

$\therefore \theta^T = \left(c^2,\ -2ac,\ a^2-b \right)$ $\quad \theta_1 = c^2$, $\theta_2 = -2ac$, $\theta_3 = a^2 - b$

(b) $c = \sqrt{\theta_1}$ $\quad a = -\dfrac{\theta_2}{2c} = -\dfrac{\theta_2}{2\sqrt{\theta_1}}$

$$b = a^2 - \theta_3 = \frac{\theta_2^2}{4\theta_1} - \theta_3 .$$

$$\therefore \begin{cases} a = -\dfrac{\theta_2}{2\sqrt{\theta_1}} \\ b = \dfrac{\theta_2^2}{4\theta_1} - \theta_3 \\ c = \sqrt{\theta_1} . \end{cases}$$

(As a and c are ~~int~~ squared in computation of $y(n)$, it does not matter whether ~~a~~ c is positive or negative. Here assume $c > 0$)

(c). $u(n) = \pm 1$. $\therefore \Psi(n)^T = [1, u(n), 1]$.

$$\therefore M^T = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ u(1) & u(2) & \cdots & u(n) \\ 1 & 1 & \cdots & 1 \end{bmatrix}$$

$$M^T \cdot M = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ u(1) & u(2) & \cdots & u(n) \\ 1 & 1 & \cdots & 1 \end{pmatrix} \begin{pmatrix} 1 & u(1) & 1 \\ 1 & u(2) & 1 \\ & \vdots & \\ 1 & u(n) & 1 \end{pmatrix}$$

$$= \begin{pmatrix} \sum_{i=1}^n 1 & \sum_{i=1}^n u(i) & \sum_{i=1}^n 1 \\ \sum_{i=1}^n u(i) & \sum_{i=1}^n 1 & \sum_{i=1}^n u(i) \\ \sum_{i=1}^n 1 & \sum_{i=1}^n u(i) & \sum_{i=1}^n 1 \end{pmatrix} = \begin{pmatrix} n & \sum u(i) & n \\ \sum u(i) & n & \sum u(i) \\ n & \sum u(i) & n \end{pmatrix}$$

$\det(M^T M) = 0$. It is not invertible.

3 (d)

If p is even,

$$M^T M = \begin{pmatrix} 1 & 4 & 1 & 4 & \cdots & 1 & 4 \\ 1 & -2 & 1 & -2 & \cdots & 1 & -2 \\ 1 & 1 & 1 & 1 & \cdots & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 4 & -2 & 1 \\ 1 & 1 & 1 \\ 4 & -2 & 1 \\ & \vdots & \\ 1 & 1 & 1 \\ 4 & -2 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} \frac{p}{2}(1+4^2) & \frac{p}{2}(1-8) & \frac{p}{2}(1+4) \\ \frac{p}{2}(1-8) & \frac{p}{2}(1+4) & \frac{p}{2}(1-2) \\ \frac{p}{2}(1+4) & \frac{p}{2}(1-2) & \frac{p}{2}(1+1) \end{pmatrix}$$

$$= \begin{pmatrix} \frac{17}{2}p & -\frac{7}{2}p & \frac{5}{2}p \\ -\frac{7}{2}p & \frac{5}{2}p & -\frac{1}{2}p \\ \frac{5}{2}p & -\frac{1}{2}p & p \end{pmatrix}$$

$\cancel{M^T M_{1:} - M^T M_{3:} - (}$

$$\det(M^T M) = 2 \times \begin{vmatrix} \frac{17}{2}p & -\frac{7}{2}p & \frac{5}{2}p \\ -\frac{7}{2}p & \frac{5}{2}p & -\frac{1}{2}p \\ 5p & -p & 2p \end{vmatrix} = 2 \begin{vmatrix} 5p & -p & 2p \\ -\frac{7}{2}p & \frac{5}{2}p & -\frac{1}{2}p \\ 5p & -p & 2p \end{vmatrix} =$$

$$= 0$$

p is odd, p-1 is even

$$M^TM = \begin{pmatrix} \frac{17}{2}(p-1) & -\frac{7}{2}(p-1) & \frac{5}{2}(p-1) \\ -\frac{7}{2}(p-1) & \frac{5}{2}(p-1) & -\frac{1}{2}(p-1) \\ \frac{5}{2}(p-1) & -\frac{1}{2}(p-1) & (p-1) \end{pmatrix} + \begin{pmatrix} 16 & -8 & 4 \\ -8 & 4 & -2 \\ 4 & -2 & 1 \end{pmatrix} \text{ or } \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

As
$$\begin{vmatrix} \frac{17(p-1)}{2} & -\frac{7}{2}(p-1) & \frac{5}{2}(p-1) \\ -\frac{7}{2}(p-1) & \frac{5}{2}(p-1) & -\frac{1}{2}(p-1) \\ \frac{5}{2}(p-1) & -\frac{1}{2}(p-1) & (p-1) \end{vmatrix} = 0. \quad |M^TM| \approx 0.$$

makes it hardly invertiable

3(e)

3.(e). Make p large enough.

$$M^TM \left(\begin{matrix} \cdots 1 & 4 & 1 & 4 & 0 \cdots \\ \cdots 1 & -2 & 1 & -2 & 0 \cdots \\ \cdots 1 & 1 & 1 & 1 & 1 \cdots \end{matrix}\right) \begin{pmatrix} \vdots & \vdots & \vdots \\ 1 & 1 & 1 \\ 4 & -2 & 1 \\ 1 & 1 & 1 \\ 4 & -2 & 1 \\ 0 & 0 & 1 \\ \vdots & \vdots & \vdots \end{pmatrix}$$

$$\approx \begin{pmatrix} \frac{p}{5}(1+4^2+4^2+1) & \frac{p}{5}(1-8+1-8) & \frac{p}{5}(4+1+4+1) \\ \frac{p}{5}(1-8+1-8) & \frac{p}{5}(1+4+1+4) & \frac{p}{5}(1-2+1-2) \\ \frac{p}{5}(1+4+1+4) & \frac{p}{5}(1-2+1-2) & \frac{p}{5}(1+1+1+1+1) \end{pmatrix}.$$

$$= \begin{pmatrix} \frac{34p}{5} & -\frac{14}{5}p & 2p \\ -\frac{14}{5}p & 2p & -\frac{2}{5}p \\ 2p & -\frac{2}{5}p & p \end{pmatrix}.$$

$$= \frac{144p^3}{125} \neq 0. \qquad \therefore M^TM \text{ invertiable}.$$

3(f)

3.(f). Make p large enough.

$$M^TM = \begin{pmatrix} \cdots & 1 & 4 & 4 & 1 & \cdots \\ \cdots & 1 & 2 & -2 & -1 & \cdots \\ \cdots & 1 & 1 & 1 & 1 & \cdots \end{pmatrix} \begin{pmatrix} & \vdots & \\ 1 & 1 & 1 \\ 4 & 2 & 1 \\ 4 & -2 & 1 \\ 1 & -1 & 1 \\ & \vdots & \end{pmatrix}$$

$$\approx \begin{pmatrix} \frac{p}{4}(1+4^2+4^2+1) & \frac{p}{4}(1+8-8-1) & \frac{p}{4}(1+4+4+1) \\ \frac{p}{4}(1+8-8-1) & \frac{p}{4}(1+4+4+1) & \frac{p}{4}(1+2-2-1) \\ \frac{p}{4}(1+4+4+1) & \frac{p}{4}(1+2-2-1) & \frac{p}{4}(1+1+1+1) \end{pmatrix}$$

$$= \begin{pmatrix} \frac{17}{2}p & 0 & \frac{5}{2}p \\ 0 & \frac{5}{2}p & 0 \\ \frac{5}{2}p & 0 & 1 \end{pmatrix}$$

$$= \frac{17}{2}p \cdot \left(\frac{5}{2}p \times 1\right) + \frac{5}{2}p\left(-\frac{5}{2}p \times \frac{5}{2}p\right)$$

The Det does not equal to 0, it is invertible.

3(g)

Mathematica is used to compute the condition number of M in e and f.

```
In[2667]:= Clear[p];

M1 = {{34/5 p, -14/5 p, 2 p}, {-14/5 p, 2 p, -2/5 p}, {2 p, -2/5 p, p}};

M2 = {{17/2 p, 0, 5/2 p}, {0, 5/2 p, 0}, {5/2 p, 0, 1}};

In[2670]:= p = 51;
N[LinearAlgebra`MatrixConditionNumber[M1]]
N[LinearAlgebra`MatrixConditionNumber[M2]]

Out[2671]= 116.

Out[2672]= 19.8904
```

The results shows that M in e has a smaller condition number. Therefore, we use M in e to compute the estimate.

Mathematica is used to generate noisy y. Code and figure are shown as followed.

In[2151]:=
```
(* Exam 3g*)
Myrandom[a_] := Module[{x = a},
   x = Switch[RandomInteger[4], 0, 0, 1, 1, 2, -2, 3, 1, 4, -2];
   x];
a = 1.7; b = -1; c = 0.5;
y[x_] := (c x - a)^2 - b;
uin = Table[Myrandom[1], {n, 1, 51}];
yout = y[uin];
noisey = yout + Table[RandomReal[{-10, 10}], {n, 51}];
ListLinePlot[noisey]
fi[x_] := {x^2, x, 1};
M = Table[fi[uin[[n]]], {n, 1, 51}];
thetax = Inverse[M^T . M].M^T.noisey
```

$$\texttt{abc} = \left\{\frac{-\texttt{thetax[[2]]}}{2\sqrt{\texttt{thetax[[1]]}}}, \frac{(\texttt{thetax[[2]]})^2}{4\,\texttt{thetax[[1]]}} - \texttt{thetax[[3]]}, \sqrt{\texttt{thetax[[1]]}}\right\}$$

```
error = abc - {1.7, -1, 0.5}
```

Out[2157]=

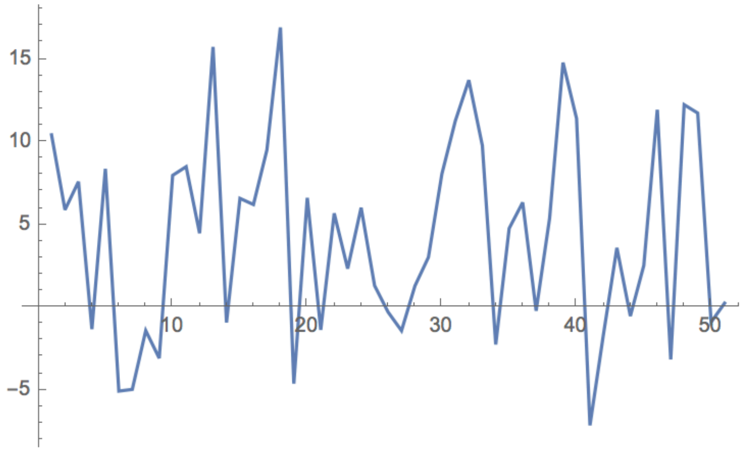
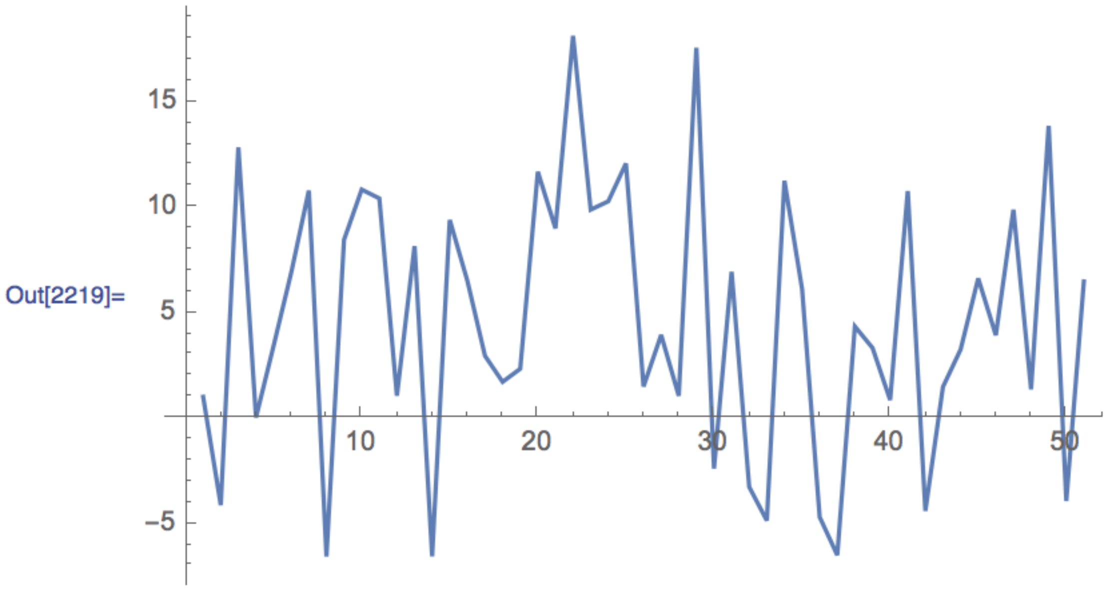


Figure 13 3(g) Code and Noisy y

Estimate abc and the error of the estimate abc is

```
Abc = {0.103805, -3.00127, 1.0564}
Error = {-1.59619, -2.00127, 0.556399}
```

Do another one, the results is



Out[2222]= {0.215833, -1.08685, 3.86549}

Out[2223]= {1.16972, -2.49725, 0.464578}

Out[2224]= {-0.530282, -1.49725, -0.0354219}

```
abc = {1.16972, -2.49725, 0.464578}
error = {-0.530282, -1.49725, -0.0354219}
```

Comparing several results, we can conclude that the estimate result is not accurate and it is almost random.

3(h)

Repeat the process in g and let n = 101,501,1001. The results are shown in Table I.

Table 1 3(h) Accuracy Results when n changes

n	Estimate abc and error 1	Estimate abc and error 2
101	{1.35559, -0.766205, 0.702181} {-0.344405, 0.233795, 0.202181}	{1.06653, -3.03909, 0.583006} {-0.63347, -2.03909, 0.0830058}
501	{1.01868, -2.82537, 0.710259} {-0.681318, -1.82537, 0.210259}	{1.45193, -1.37333, 0.6271} {-0.248073, -0.373334, 0.1271}
1001	{1.24835, -2.01966, 0.599164} {-0.451649, -1.01966, 0.0991641}	{1.50607, -1.53619, 0.563866} {-0.193926, -0.536193, 0.0638664}
10001	{2.12137, 0.496054, 0.419582} {0.421371, 1.49605, -0.0804183}	{1.34721, -1.85971, 0.60159} {-0.352792, -0.859708, 0.10159}

Based on the observation, the bigger n is ,the higher probability that the estimate is close to the condition. As is shown in the table, when n goes up, the average error of estimate abc tends to drop. When n goes up to large number like 10001, this method can give a good estimate.

4. In writing script, Inverse Z transform the transfer function, then we can get the differentiate equation as

$$\frac{Y(z)}{U(z)} = \frac{\beta_1 z + \beta_2}{z^2 + \alpha_1 z + \alpha_2}.$$

$$\Rightarrow \quad y(n) + \alpha_1 y(n-1) + \alpha_2 y(n-2) = \beta_1 u(n-1) + \beta_2 u(n-2).$$

$$-\alpha_1 y(n-1) - \alpha_2 y(n-2) + \beta_1 u(n-1) + \beta_2 u(n-2) = y(n).$$

Using this equation, M can be constructed.
M is constructed in Mathematica, shown as followed.

```
In[105]:= (* EECS5410 Exam 4 *)
Un = Table[-0.3, {n, 14}] ~Join~ Table[2.3, {n, 9}];
Yn = {3.96, 3.74, 3.52, 3.3, 3.08, 2.87, 2.65, 2.44, 2.22, 2.01, 1.79, 1.58,
   1.37, 1.16, 0.95, 0.84, 0.76, 0.71, 0.7, 0.71, 0.76, 0.83, 0.94};
M = Table[0, {n, 21}];
For[i = 1, i ≤ 21, i ++,
  M[[i]] = {-Yn[[i + 1]], -Yn[[i]], Un[[i + 1]], Un[[i]]}
 ];
MatrixForm[M]
```

Out[109]//MatrixForm=

$$
\begin{pmatrix}
-3.74 & -3.96 & -0.3 & -0.3 \\
-3.52 & -3.74 & -0.3 & -0.3 \\
-3.3 & -3.52 & -0.3 & -0.3 \\
-3.08 & -3.3 & -0.3 & -0.3 \\
-2.87 & -3.08 & -0.3 & -0.3 \\
-2.65 & -2.87 & -0.3 & -0.3 \\
-2.44 & -2.65 & -0.3 & -0.3 \\
-2.22 & -2.44 & -0.3 & -0.3 \\
-2.01 & -2.22 & -0.3 & -0.3 \\
-1.79 & -2.01 & -0.3 & -0.3 \\
-1.58 & -1.79 & -0.3 & -0.3 \\
-1.37 & -1.58 & -0.3 & -0.3 \\
-1.16 & -1.37 & -0.3 & -0.3 \\
-0.95 & -1.16 & 2.3 & -0.3 \\
-0.84 & -0.95 & 2.3 & 2.3 \\
-0.76 & -0.84 & 2.3 & 2.3 \\
-0.71 & -0.76 & 2.3 & 2.3 \\
-0.7 & -0.71 & 2.3 & 2.3 \\
-0.71 & -0.7 & 2.3 & 2.3 \\
-0.76 & -0.71 & 2.3 & 2.3 \\
-0.83 & -0.76 & 2.3 & 2.3
\end{pmatrix}
$$

(b) Use M to compute the theta using this equation

$$\hat{\theta}^* = (M^T M)^{-1} M^T Y$$

```
In[113]:= Y1 = Yn[[3 ;;]];
theta = Inverse[Mᵀ.M].Mᵀ.Y1

Out[114]= {-1.98145, 0.981178, 0.038392, -0.0249655}
```

Using theta, we can get the G(z) as

$$\frac{-0.0249655 + 0.038392\, z}{0.981178 - 1.98145\, z + z^2}$$

And then plot the system pzmap in Matlab.

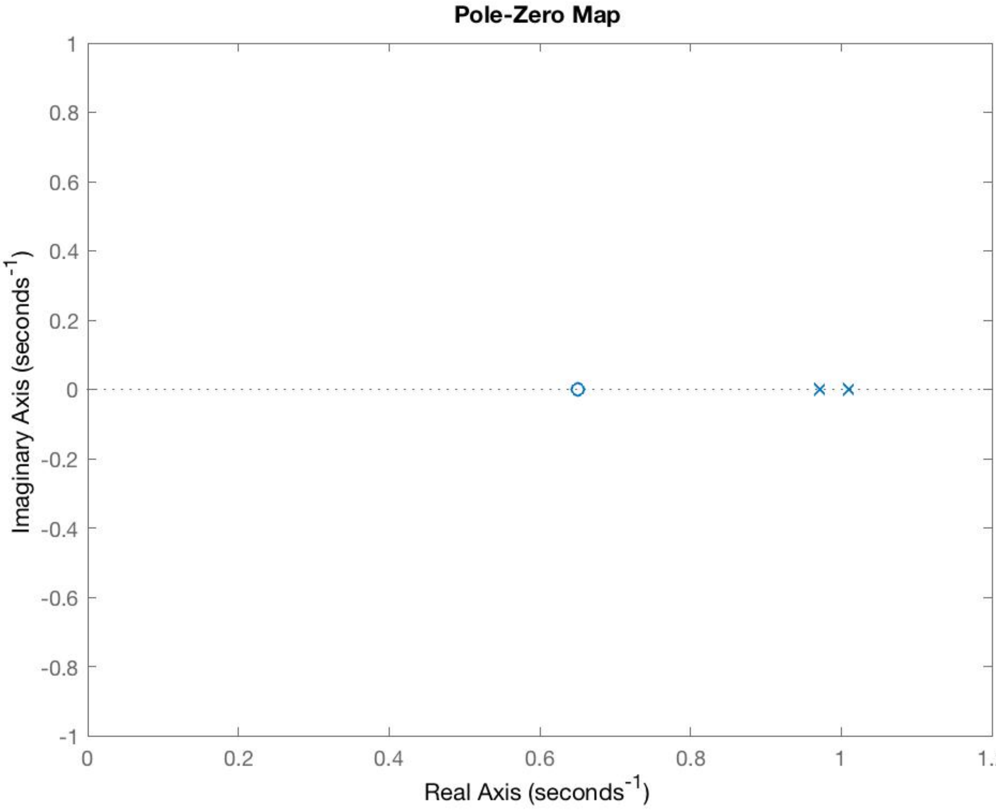


Figure 14 4b pzmap

5.

(a)

Choose f = c = -0.65 to cancel this pole, choose p = 0.1.

Then use root locus to determine the gain (g). Based on what are chosen, root locus is plotted as followed. One set of poles is selected by rlocfind, then we get the gain, and its step response, shown as followed.

```
selected_point =

   0.2050 + 0.4465i

gain =

    6.1073

poles =
```

```
   0.6500 + 0.0000i
   0.2057 + 0.4469i
   0.2057 - 0.4469i

maxy =

    1.1368
```

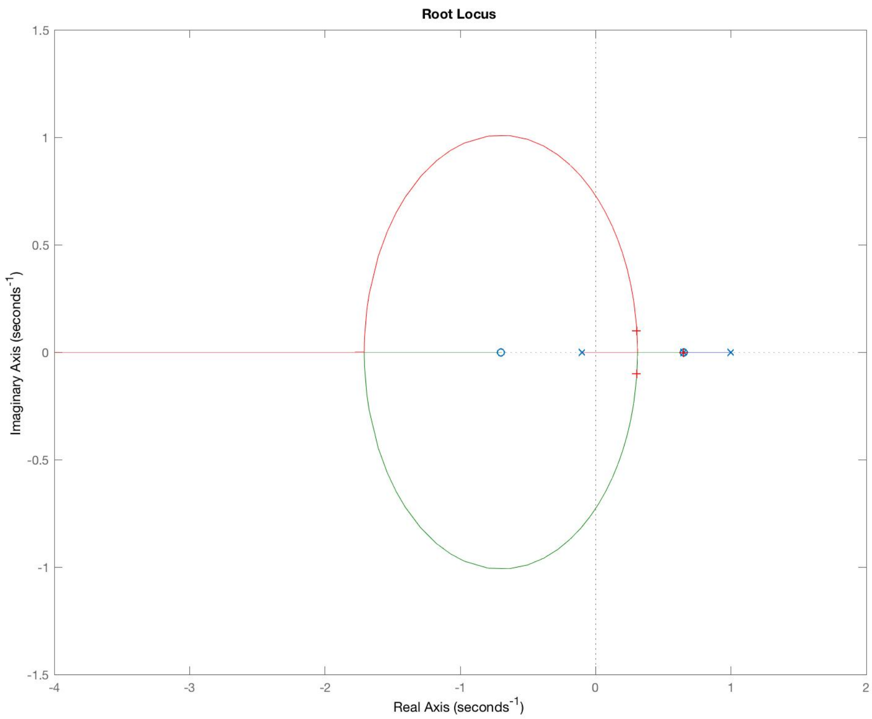
Then we have g = 6.1073. As its shown in the result and plots:

The maximum of step response is 1.1368, and the final DC gain is 1. The percent overshoot is 13.68%, within the range of 2% to 18%;

Examine the results of step response, we can find out the last one excesses 2% of the final value. Therefore, the 2% settling time is 6 samples.

	1
1	0
2	0.4886
3	1.0316
4	1.1368
5	1.0486
6	0.9869
7	0.9828
8	0.9961
9	1.0026
10	1.0020
11	1.0002
12	0.9996
13	0.9998
14	1.0000
15	1.0001

Figure 15 5a step response matrix



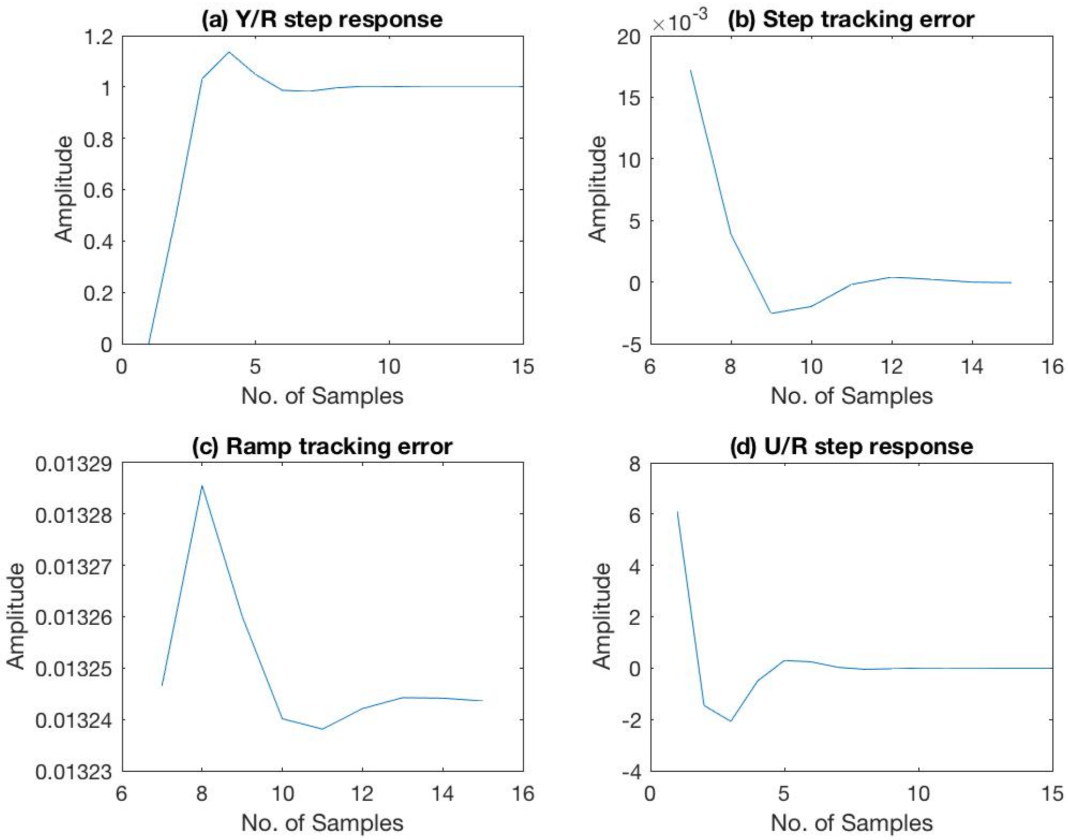


Figure 16 5a rootlocus and step response

(b)

Choose f = c = -0.5 to cancel this pole, choose p = 0.1.

Then use root locus to determine the gain (g). Based on what are chosen, root locus is plotted as followed. One set of poles is selected by rlocfind, then we get the gain, and its step response, shown as followed.

```
selected_point =

   0.2192 + 0.4186i

gain =

    5.7718

poles =

   0.5000 + 0.0000i
   0.2191 + 0.4186i
   0.2191 - 0.4186i

maxy =

1.1146
```

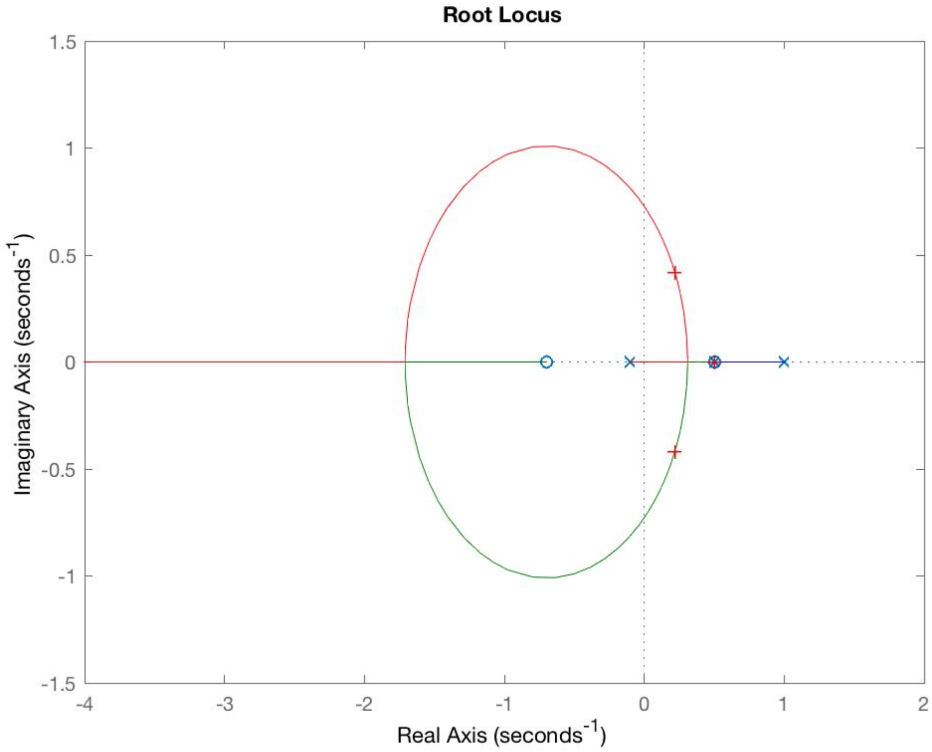
g = 5.77. As its shown in the result and plots:

The maximum of step response is 1.1146, and the final DC gain is 1. The percent overshoot is 11.46%, within the range of 2% to 18%;

Examine the results of step response, we can find out the last one excesses 2% of the final value. Therefore, the 2% settling time is 6 samples.

	1
1	0
2	0.4617
3	0.9873
4	1.1146
5	1.0531
6	0.9977
7	0.9871
8	0.9949
9	1.0006
10	1.0014
11	1.0005
12	0.9999
13	0.9998
14	1.0000
15	1.0000

Figure 17 5b step response matrix



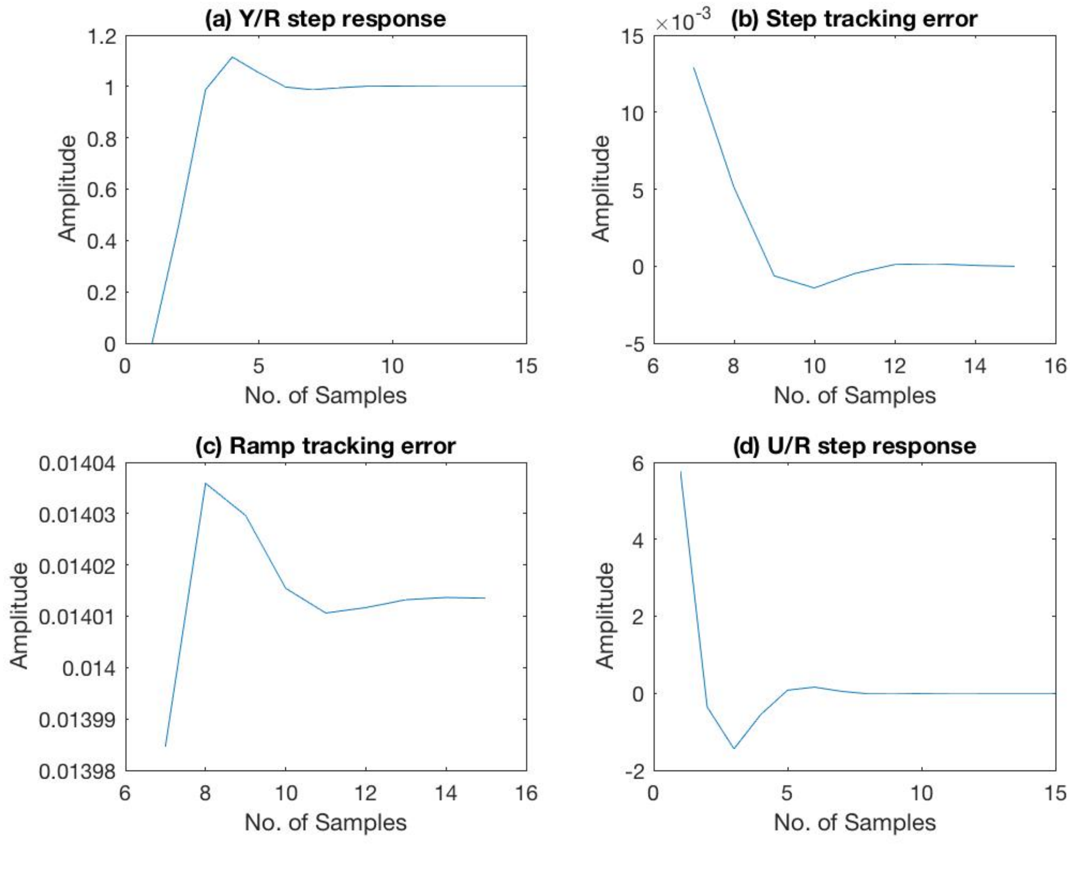


Figure 18 5b rootlocus and step response