Table of Contents

Translation

```
% Run sections sequentially
```

System

```
% A Matrix

load Matrices/A_matrix.mat
A = A_matrix

% B Matrix: Stowed

load Matrices/B_stowed.mat
B = B_stowed

% Full-State Feedback

Cf = eye(12);

Df = [zeros(12, 6)];

sys_full = ss(A, B, Cf, Df);

tf_full = minreal(tf(sys_full));

tf_translation = minreal([tf_full(1:3, 1:3); tf_full(7:9, 1:3)]);

tf_full_sym = simplify(tf2sym(tf_full));
disp('tf_full_sym = ');
pretty(tf_full_sym);

tf_translation_sym = simplify(tf2sym(tf_translation));
disp('tf_translation_sym = ');
pretty(tf_translation_sym);
```

```
A =
     0     0     0     0     0     0     0     0     0     0     0
  0
     0     0     0     0     0     0     0     0     0     0     0
  0
```

```
     0     0     0     0     0     0     0     0     0     0     0
  0
     0     0     0     0     0     0     0     0     0     0     0
  0
     0     0     0     0     0     0     0     0     0     0     0
  0
     0     0     0     0     0     0     0     0     0     0     0
  0
     1     0     0     0     0     0     0     0     0     0     0
  0
     0     1     0     0     0     0     0     0     0     0     0
  0
     0     0     1     0     0     0     0     0     0     0     0
  0
     0     0     0     1     0     0     0     0     0     0     0
  0
     0     0     0     0     1     0     0     0     0     0     0
  0
     0     0     0     0     0     1     0     0     0     0     0
  0
B =
    0.0630         0         0         0         0         0
         0    0.0630         0         0         0         0
         0         0    0.0630         0         0         0
         0         0         0    5.4054         0         0
         0         0         0         0    4.9505         0
         0         0         0         0         0    5.3191
         0         0         0         0         0         0
         0         0         0         0         0         0
         0         0         0         0         0         0
         0         0         0         0         0         0
         0         0         0         0         0         0
         0         0         0         0         0         0
tf_full_sym =
/   500                                             \
| ------,    0,      0,     0,      0,     0   |
| 7939 s                                        |
|                                               |
|           500                                 |
|    0,   ------,    0,     0,      0,     0   |
|         7939 s                                |
|                                               |
|                   500                         |
|    0,      0,   ------,   0,      0,     0   |
|                 7939 s                        |
|                                               |
|                           200                 |
|    0,      0,      0,    ----,    0,     0   |
|                           37 s                |
|                                               |
|                                   500         |
|    0,      0,      0,     0,    -----,   0   |
|                                  101 s        |
|                                               |
```

```
|                                          250 |
|    0,      0,      0,     0,      0,    ---- |
|                                        47 s |
|                                              |
|   #1,      0,      0,     0,      0,     0   |
|                                              |
|    0,     #1,      0,     0,      0,     0   |
|                                              |
|    0,      0,     #1,     0,      0,     0   |
|                                              |
|                          200                 |
|    0,      0,      0,   -----,    0,     0   |
|                             2                |
|                         37 s                 |
|                                              |
|                                  500         |
|    0,      0,      0,     0,   ------,   0   |
|                                     2        |
|                                101 s         |
|                                              |
|                                         250  |
|    0,      0,      0,     0,      0,   ----- |
|                                            2 |
\                                        47 s  /

where

           500
   #1 == -------
               2
         7939 s


tf_translation_sym =
/    500                     \
|  ------,    0,       0     |
|  7939 s                    |
|                            |
|             500            |
|    0,      ------,   0     |
|            7939 s          |
|                            |
|                      500   |
|    0,       0,     ------  |
|                    7939 s  |
|                            |
|   500                      |
| -------,    0,       0     |
|       2                    |
| 7939 s                     |
|                            |
|            500             |
|    0,     -------,   0     |
|                2           |
```

```
|          7939 s           |
|                           |
|                    500    |
|    0,       0,    ------- |
|                        2  |
\                  7939 s  /
```

Smith-McMillan Form

```
% help smform

s=tf('s');

Gp = tf_translation;

Mp = minreal(smform(Gp));

syms s

Gp_sym = tf_translation_sym;

% UL_sym = [7939/(500*s), 0, 0,  0,  0,  0;
%            0, 7939/(500*s), 0,  0,  0,  0;
%            0, 0, 7939/(500*s),  0,  0,  0;
%            1/s, 0, 0, -1,  0,  0;
%            0, 1/s, 0,  0, -1,  0;
%            0, 0, 1/s,  0,  0, -1];
%
% UR_sym = eye(3);

UL_sym = [0, 0, 0,  1,  0,  0;
          0, 0, 0,  0,  1,  0;
          0, 0, 0,  0,  0,  1;
          -1, 0, 0,  s,  0,  0;
          0, -1, 0,  0,  s,  0;
          0, 0, -1,  0,  0,  s]

UR_sym = [7939/500, 0, 0;
          0, 7939/500, 0;
          0, 0, 7939/500]


disp('UL_sym = ');
pretty(UL_sym);

Mp_sym = tf2sym(Mp);
disp('Mp_sym = ');
pretty(Mp_sym);

disp('UR_sym = ');
UR_sym;
```

```
UL = sym2tf(UL_sym);
UR = UR_sym;

UL_sym =
[  0,  0,  0, 1, 0, 0]
[  0,  0,  0, 0, 1, 0]
[  0,  0,  0, 0, 0, 1]
[ -1,  0,  0, s, 0, 0]
[  0, -1,  0, 0, s, 0]
[  0,  0, -1, 0, 0, s]
UR_sym =
   15.8780         0         0
         0   15.8780         0
         0         0   15.8780
UL_sym =
/  0,  0,  0, 1, 0, 0 \
|                     |
|  0,  0,  0, 0, 1, 0 |
|                     |
|  0,  0,  0, 0, 0, 1 |
|                     |
| -1,  0,  0, s, 0, 0 |
|                     |
|  0, -1,  0, 0, s, 0 |
|                     |
\  0,  0, -1, 0, 0, s /

Mp_sym =
/  1         \
| --,  0,  0 |
|  2         |
| s          |
|            |
|      1     |
|  0, --,  0 |
|      2     |
|     s      |
|            |
|          1 |
|  0,  0, -- |
|          2 |
|         s  |
|            |
|  0,  0,  0 |
|            |
|  0,  0,  0 |
|            |
\  0,  0,  0 /

UR_sym =
```

Interpolation Conditions

```
% Run this section first to calculate 'tz' to ensure that the second
 interpolation condition is satisfied

% d^k(T)/ds^k|(s=0) = 0, where k = 1 (since there is a double unstable
 pole
% (multiplicity ap = 2) in the plant at s = 0; k = ap - 1) -> 2nd
% interpolation condition

% Constants & Design Parameters

C = 1; % Constant
Wn = 1; % Natural Frequency of the Control System
K = Wn^2/C; % Controller Gain
Z = 2^-0.5; % Damping Ratio
tp = 1/(10*Wn); % Time constant (of the first included pole)
tpx = 0.5; % Time constant (of the pole included to drop Youla at high
 frequencies)
% tzx = 999;

syms s tz

T_eqn = ((K*C)*(tz*s + 1)/((s^2 + 2*Z*Wn*s + Wn^2)*(tp*s + 1)*(tpx*s +
 1)^2));
dT_eqn = diff(T_eqn,s);
eqn = subs(dT_eqn,s,0) == 0;
tz = double(solve(eqn,tz))
```

```
tz =
    2.5142
```

Control Design

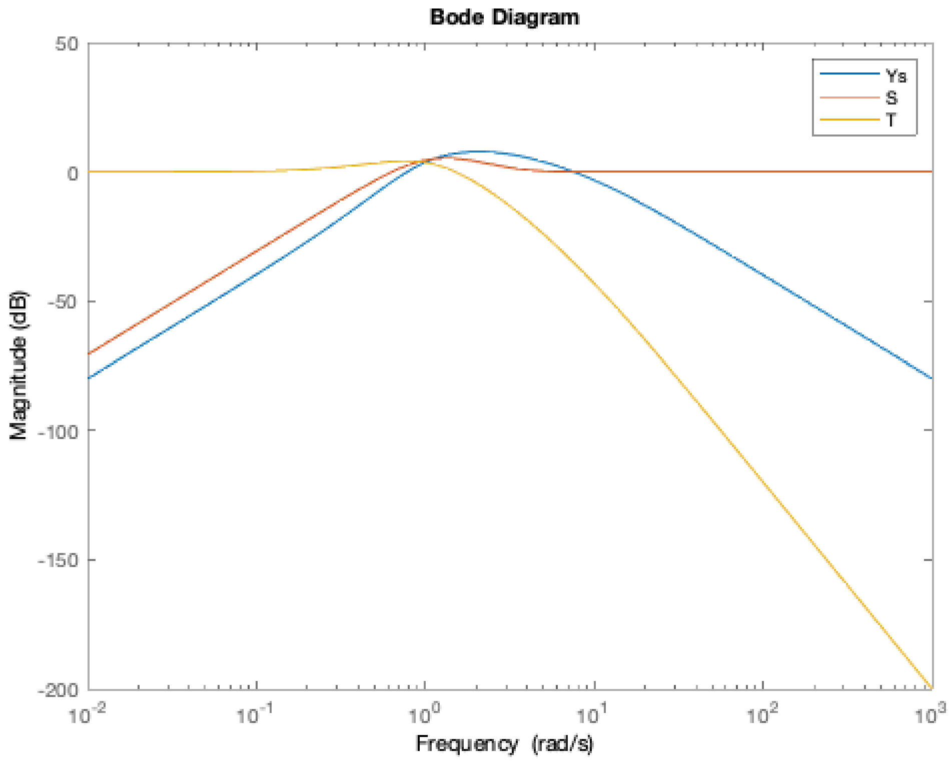
```
s = tf('s');

% Plant TF, 'Gp'
G = minreal(C/s^2, 1e-04) % Nonzero terms of Mp

% Chosen Youla Parameter, 'Y' -> Y(0) = 0
Ys = minreal(((K*s^2)*(tz*s + 1)/((s^2 + 2*Z*Wn*s + Wn^2)*(tp*s +
 1)*(tpx*s + 1)^2)),1e-04)

% Complementary Sensitivity TF, 'T' -> T(0) = 1 (1st interpolation
% condition)
T = minreal((Ys*G),1e-04)

% Sensitivity TF, 'S'
S = minreal((1-T),1e-04)

% Controller TF, 'Gc'
Gc_term = minreal((Ys/S),1e-04)
```

```
% Return Ratio, 'L'
L = minreal((Gc_term*G),1e-04)

GS = minreal((G*S),1e-04)

% Internal stability check
Y_stability = isstable(Ys)
T_stability = isstable(T)
S_stability = isstable(S)
GS_stability = isstable(GS)

M2 = 1/getPeakGain(S) % M2-margin
BW = bandwidth(T) % Bandwidth of the closed-loop
AE = getPeakGain(Ys) % Maximum actuator effort

figure(1)
bodemag(Ys, S, T);
legend('Ys','S','T');

% Gc = minreal([tf(Gc_term) 0 0 0 0 0; 0 tf(Gc_term) 0 0 0 0; 0 0
 tf(Gc_term) 0 0 0]);
%
% Gc_sym = expand(tf2sym(Gc));
% disp('Gc_sym = ');
% pretty(Gc_sym);

My = minreal([[Ys 0 0; 0 Ys 0; 0 0 Ys] zeros(3, 3)], 1e-04);
Mt = minreal(Mp * My, 1e-04);
% Mt = minreal(T * eye(6), 1e-04);
Y = minreal(UR * My * UL, 1e-04);
```

```
G =

   1
  ---
  s^2

Continuous-time transfer function.


Ys =

                    100.6 s^3 + 40 s^2
  -----------------------------------------------------
  s^5 + 15.41 s^4 + 64.8 s^3 + 116.2 s^2 + 100.6 s + 40

Continuous-time transfer function.


T =

                       100.6 s + 40
```

```
  -----------------------------------------------------
  s^5 + 15.41 s^4 + 64.8 s^3 + 116.2 s^2 + 100.6 s + 40

Continuous-time transfer function.


S =

  s^5 + 15.41 s^4 + 64.8 s^3 + 116.2 s^2 - 2.274e-13 s - 1.776e-13
  ----------------------------------------------------------------
       s^5 + 15.41 s^4 + 64.8 s^3 + 116.2 s^2 + 100.6 s + 40

Continuous-time transfer function.


Gc_term =

            100.6 s + 40
  --------------------------------
  s^3 + 15.41 s^2 + 64.8 s + 116.2

Continuous-time transfer function.


L =

               100.6 s + 40
  --------------------------------------
  s^5 + 15.41 s^4 + 64.8 s^3 + 116.2 s^2

Continuous-time transfer function.


GS =

            s^3 + 15.41 s^2 + 64.8 s + 116.2
  -----------------------------------------------------
  s^5 + 15.41 s^4 + 64.8 s^3 + 116.2 s^2 + 100.6 s + 40

Continuous-time transfer function.

Y_stability =
  logical
   1
T_stability =
  logical
   1
S_stability =
  logical
   1
GS_stability =
  logical
   1
M2 =
```

```
    0.5469
BW =
    1.8487
AE =
    2.4399
```



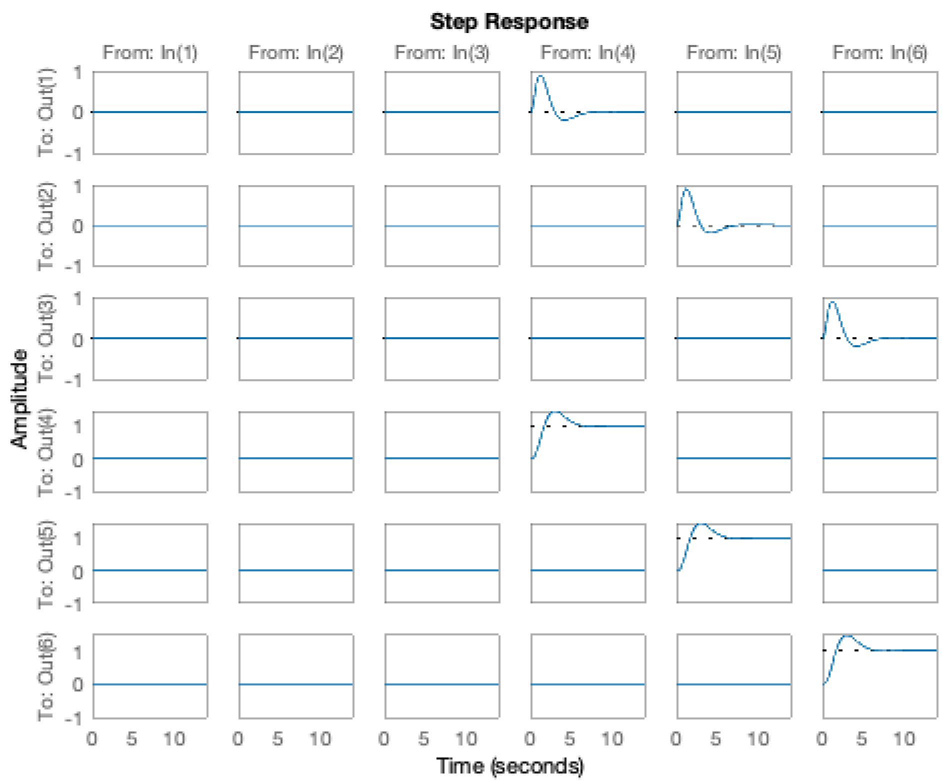
Simulation

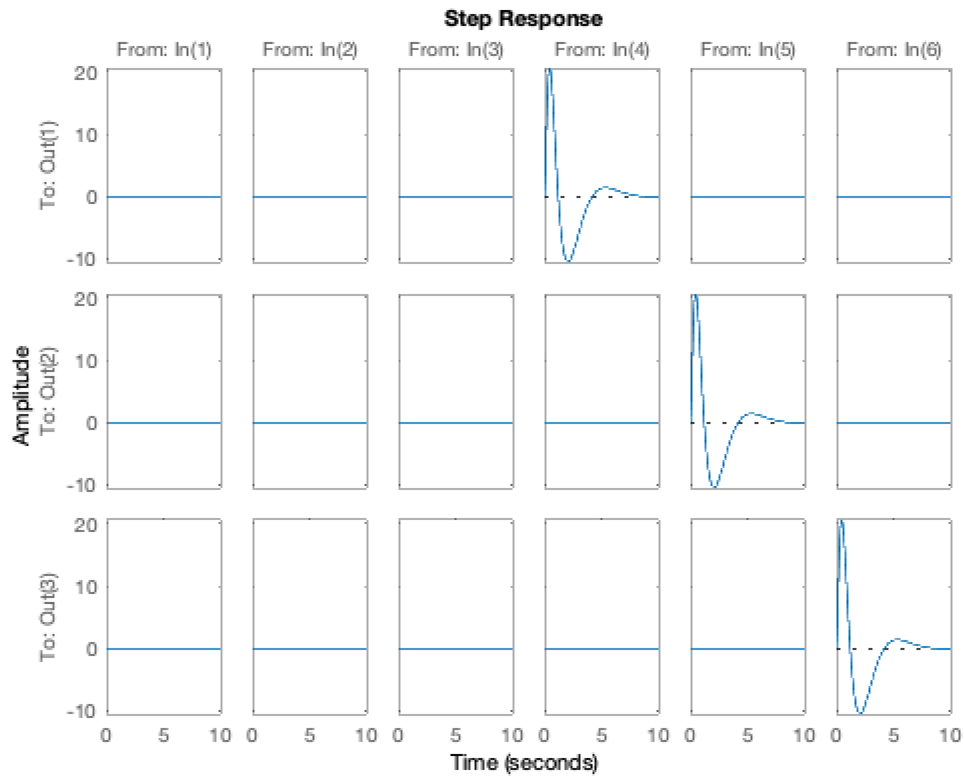
```
% Y = minreal(inv(eye(3) + Lu) * Gc);
% Ty = minreal(inv(eye(6) + Ly) * Ly);
% Sy = minreal(inv(eye(6) + Ly), 1e-04);

Ty = minreal(inv(UL) * Mp * My * UL, 1e-04);

Sy = minreal(eye(size(Ty)) - Ty, 1e-04);

Gc = minreal(UR * inv(eye(size(My * Mp)) - (My * Mp)) * My * UL,
 1e-04);

SyGp = minreal(inv(UL) * (eye(size(Mp * My)) - (Mp * My)) * Mp *
 inv(UR), 1e-04);

% MIMO Internal Stability Check:
Ty_stability = isstable(Ty)
Sy_stability = isstable(Sy)
Gc_stability = isstable(Gc)
```

```
SyGp_stability = isstable(SyGp)

SV_Gp = sigma(Gp);
k_Gp = max(max(SV_Gp))/min(min(SV_Gp)) % condition-number check for Gp

SV_Gc = sigma(Gc);
k_Gc = max(max(SV_Gc))/min(min(SV_Gc)) % condition-number check for Gc

Lu = minreal(Gc * Gp, 1e-04);
Ly = minreal(Gp * Gc, 1e-04);
Su = minreal(inv(eye(3) + Lu), 1e-04);

figure
step(Ty);

figure
step(Y);

figure
sigma(Y, Ty, Sy, Su)
[l, hObj] = legend('$Y
$', '$T_{y}$', '$S_{y}$', '$S_{u}$','Interpreter','latex','FontSize',
 12);
set(l,'string',{'$Y$', '$T_{y}$', '$S_{y}$', '$S_{u}$'});
hL = findobj(hObj,'type','line');
set(hL,'linewidth', 2);

figure
sigma(Gc, Gp, Ly, Y)
[l, hObj] = legend('$G_{c}$', '$G_{p}$', '$L_{y}$', '$Y
$','Interpreter','latex','FontSize', 12);
set(l,'string',{'$G_{c}$', '$G_{p}$', '$L_{y}$', '$Y$'});
hL = findobj(hObj,'type','line');
set(hL,'linewidth', 2);

figure
sigma(Gc, Gp, Y)
[l, hObj] = legend('$G_{c}$', '$G_{p}$', '$Y
$','Interpreter','latex','FontSize', 12);
set(l,'string',{'$G_{c}$', '$G_{p}$', '$Y$'});
hL = findobj(hObj,'type','line');
set(hL,'linewidth', 2);

figure
sigma(Ly, Sy, Ty)
[l, hObj] =
 legend('$L_{y}$', '$S_{y}$', '$T_{y}$','Interpreter','latex','FontSize',
 12);
set(l,'string',{'$L_{y}$', '$S_{y}$', '$T_{y}$'});
hL = findobj(hObj,'type','line');
set(hL,'linewidth', 2);

figure
sigma(Sy, Su)
```

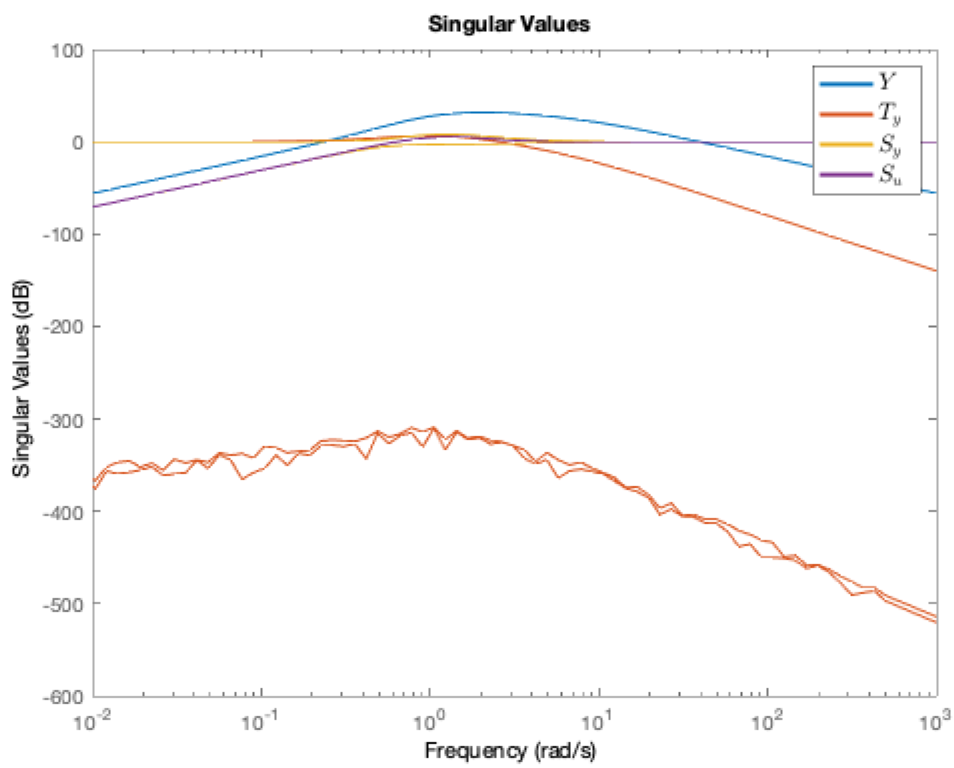
```
[l, hObj] =
 legend('$S_{y}$', '$S_{u}$','Interpreter','latex','FontSize', 12);
set(l,'string',{'$S_{y}$', '$S_{u}$'});
hL = findobj(hObj,'type','line');
set(hL,'linewidth', 2);

T_term2 = (100.6*s^2 + 40*s)/(s^5 + 15.41*s^4 + 64.8*s^3 + 116.2*s^2 +
 100.6*s + 40);
```

```
Ty_stability =
  logical
   1
Sy_stability =
  logical
   1
Gc_stability =
  logical
   1
SyGp_stability =
  logical
   1
k_Gp =
   1.0000e+03
k_Gc =
   1.7968e+04
```

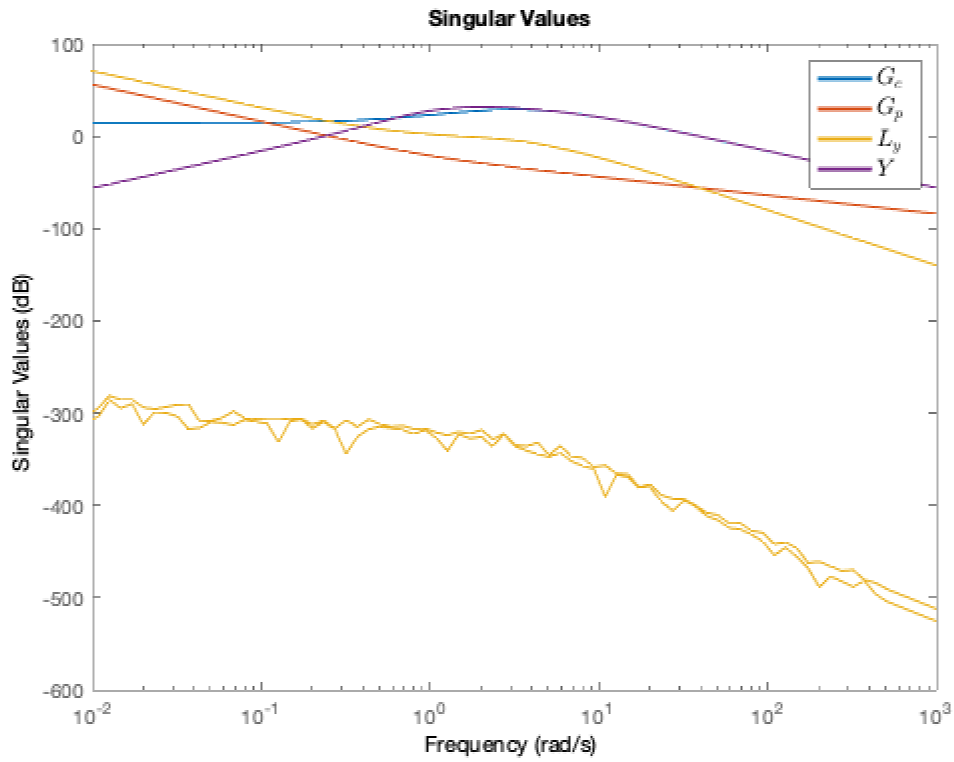


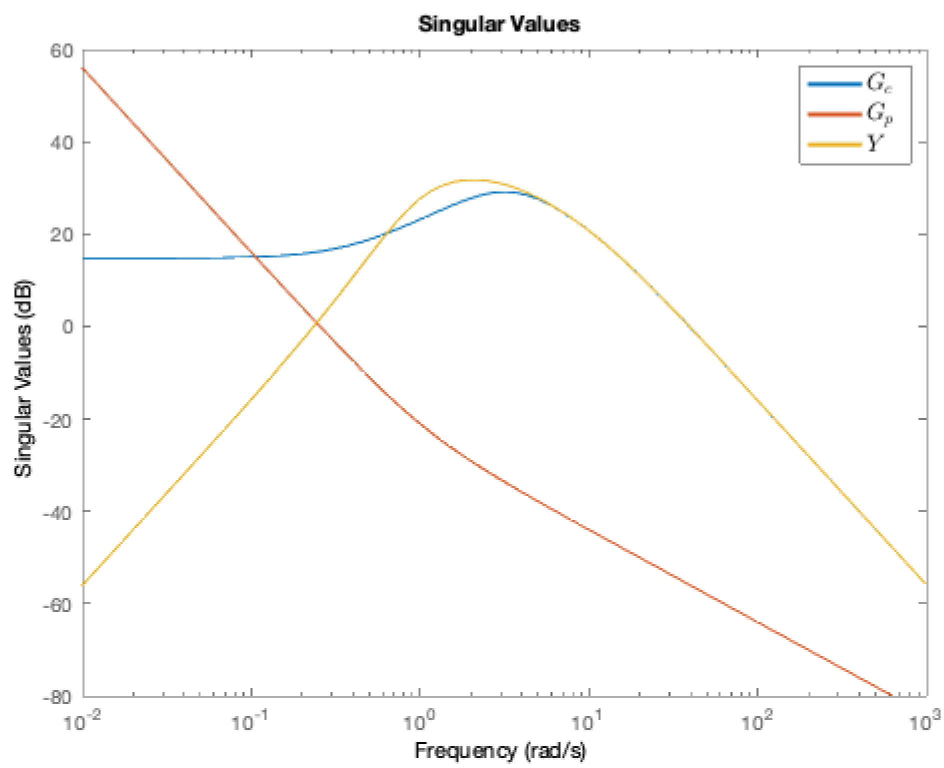


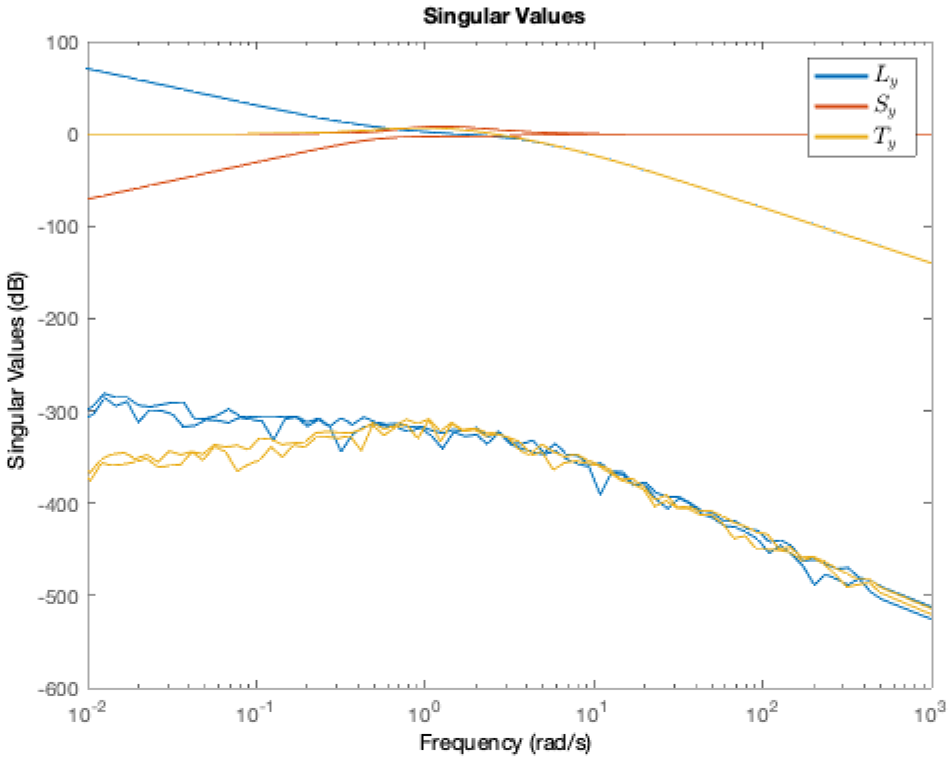
Step Response
From: In(1)
From: In(2)
From: In(3)
From: In(4)
From: In(5)
From: In(6)
Amplitude
To: Out(1)
To: Out(2)
To: Out(3)
20
10
0
-10
0
5
10
Time (seconds)

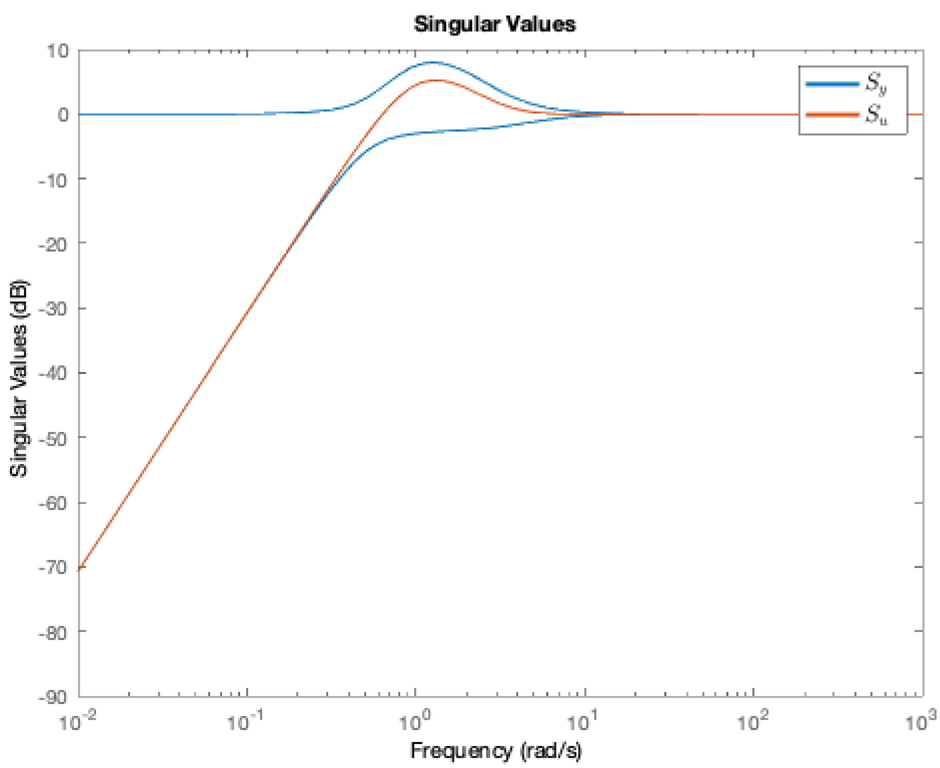


Singular Values
Singular Values (dB)
100
0
-100
-200
-300
-400
-500
-600
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
Frequency (rad/s)
Y
T_y
S_y
S_u


Singular Values
100
0
-100
-200
-300
-400
-500
-600
Singular Values (dB)
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
Frequency (rad/s)
G_c
G_p
L_y
Y


Singular Values
60
40
20
0
-20
-40
-60
-80
Singular Values (dB)
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
Frequency (rad/s)
G_c
G_p
Y


Singular Values
L_y
S_y
T_y
Singular Values (dB)
Frequency (rad/s)


Singular Values
S_y
S_u
Singular Values (dB)
Frequency (rad/s)

Published with MATLAB® R2019b