Table of Contents


```
% Astrobee Model
```

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 = tf(sys_full);

syms s

tf_full_sym = simplify(Cf * inv(s * eye(12) - A) * B + Df);
pretty(tf_full_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
/   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
```

Section 1: Translation Controller Design -> Marginally Stable Pole at the Origin

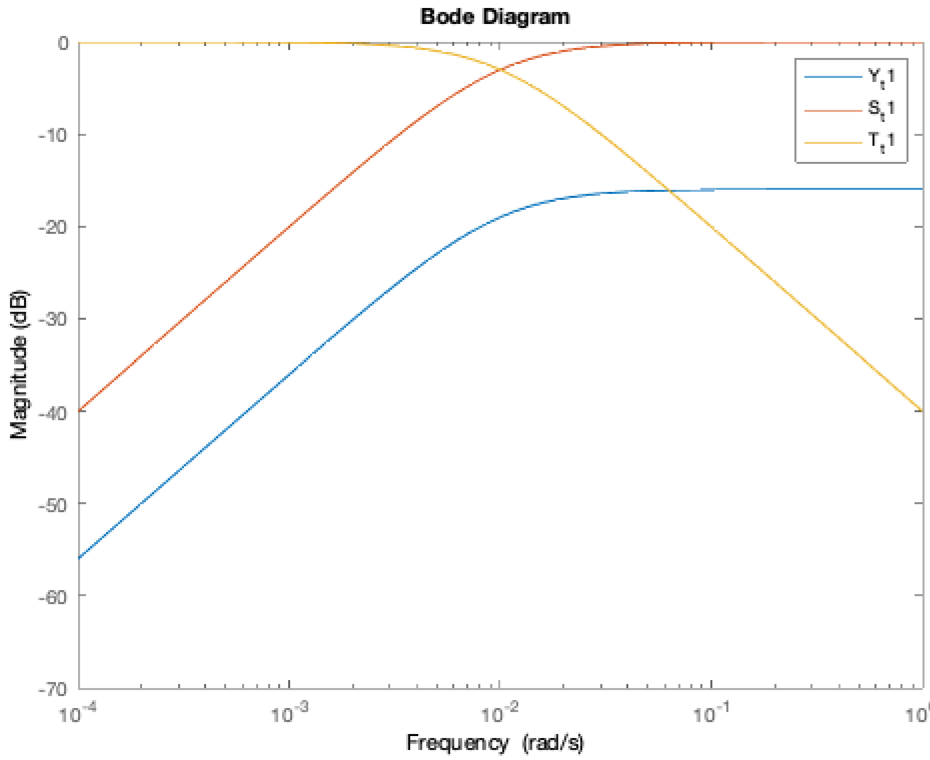
```
% Top-half matrix (t1):

% Satisfaction of the first interpolation condition

translation_full = [tf_full(1:3, 1:3); tf_full(7:9, 1:3)];
% pretty(translation_full);

G_t1 = translation_full(1:3, 1:3);

Gp_t1 = translation_full(1, 1);

K_t1 = 500/7939;

% P_t1 = Gp_t1 * s;
%
% [UL_t1, UR_t1, S_t1] = smithForm(P_t1, s)
%
% Mp_t1 = S_t1/s
%
% K_t1 = 1;
% tp_t1 = 100;
%
```

```
% Y1 = (K_t1 * s)/(tp_t1 * s + 1);
% Y2 = Y1;
% Y3 = Y1;
%
% My_t1 = diag([Y1 Y2 Y3])
%
% Mt = Mp_t1 * My_t1
%
% Gc_t1_sym = simplify((UR_t1 * inv(eye(size(My_t1 * Mp_t1)) - My_t1 *
 Mp_t1) * My_t1 * UL_t1))
%
% Gc_t1 = tf(double(Gc_t1_sym));

s = tf('s');

tp_t1 = 100;

% Chosen Youla Parameter, 'Y' -> Y(0) = 0
Y_t1 = s/(K_t1*(tp_t1 * s + 1));

% Complementary Sensitivity TF, 'T' -> T(0) = 1 (1st interpolation
% condition)
T_t1 = minreal((Y_t1*Gp_t1),1e-04)

% Sensitivity TF, 'S'
S_t1 = minreal((1-T_t1),1e-04)

% Controller TF, 'Gc'
Gc_t1 = minreal((Y_t1/S_t1),1e-04)

% Return Ratio, 'L'
L_t1 = minreal((Gc_t1*Gp_t1),1e-04)

GpS_t1 = minreal((Gp_t1*S_t1),1e-04)

% Internal stability check
Y_t1_stability = isstable(Y_t1)
T_t1_stability = isstable(T_t1)
S_t1_stability = isstable(S_t1)
GpS_t1_stability = isstable(GpS_t1)

M2_t2 = 1/getPeakGain(S_t1) % M2-margin
BW_t2 = bandwidth(T_t1) % Bandwidth of the closed-loop
AE_t2 = getPeakGain(Y_t1) % Maximum actuator effort

figure(1)
bodemag(Y_t1, S_t1, T_t1);
legend('Y_t1','S_t1','T_t1');

Gc_1 = Gc_t1 * eye(3);
```

T_t1 =

```
    0.01
  --------
  s + 0.01

Continuous-time transfer function.


S_t1 =

     s
  --------
  s + 0.01

Continuous-time transfer function.


Gc_t1 =

  0.1588

Static gain.


L_t1 =

  0.01
  ----
   s

Continuous-time transfer function.


GpS_t1 =

  0.06298
  --------
  s + 0.01

Continuous-time transfer function.

Y_t1_stability =
  logical
   1
T_t1_stability =
  logical
   1
S_t1_stability =
  logical
   1
GpS_t1_stability =
  logical
   1
M2_t2 =
     1
```

```
BW_t2 =
    0.0100
AE_t2 =
    0.1588
```



Section 2: Translation Controller Design -> Unstable Double-Pole at the Origin

```
% Bottom-half matrix (t2):

% 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

C_t2 = 500/7939; % Constant
Wn = 3.25; % Natural Frequency of the Control System
K_t2 = Wn^2/C_t2; % Controller Gain
Z = 2^-0.5; % Damping Ratio
tp_t2 = 1/(10*Wn); % Time constant (of the included pole)

syms s tz
```

```
TF = ((K_t2*C_t2)*(tz*s + 1))/((s^2 + 2*Z*Wn*s + Wn^2)*(tp_t2*s + 1))
dTF = diff(TF,s)
eqn = subs(dTF,s,0) == 0;
tz = double(solve(eqn,tz))

TF =
((169*s*tz)/16 + 169/16)/((((2*s)/65 + 1)*(s^2 + (13*2^(1/2)*s)/4 +
 169/16))
dTF =
(169*tz)/(16*((2*s)/65 + 1)*(s^2 + (13*2^(1/2)*s)/4 + 169/16))
 - (2*((169*s*tz)/16 + 169/16))/(65*((2*s)/65 + 1)^2*(s^2 +
 (13*2^(1/2)*s)/4 + 169/16)) - (((169*s*tz)/16 + 169/16)*(2*s +
 (13*2^(1/2))/4))/(((2*s)/65 + 1)*(s^2 + (13*2^(1/2)*s)/4 + 169/16)^2)
tz =
    0.4659
```

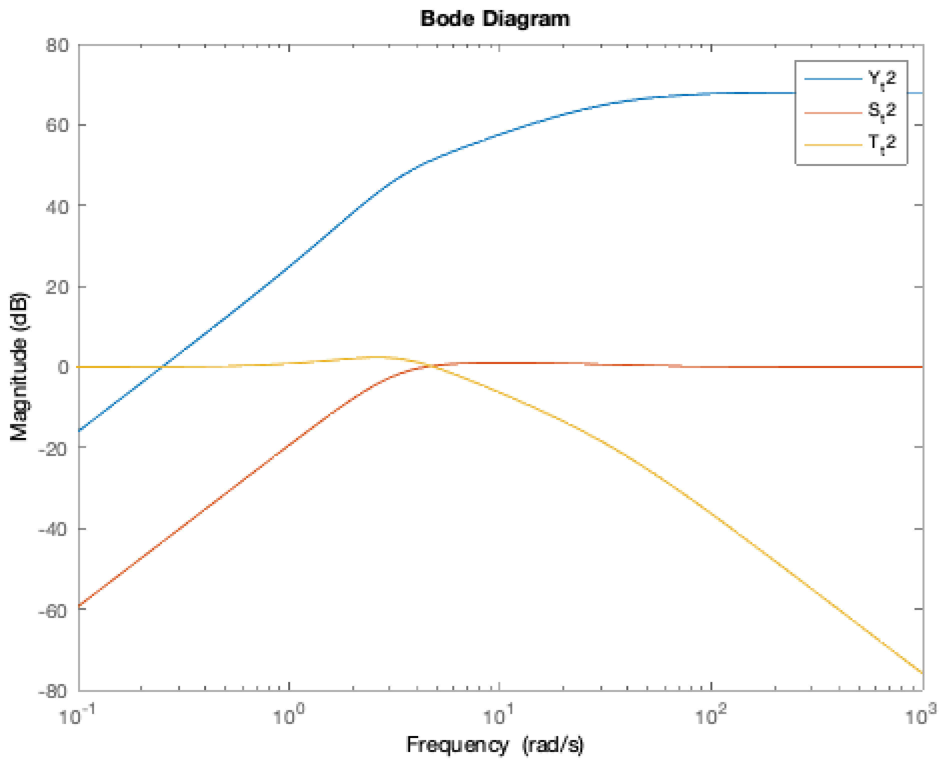
Section 3: Translation Controller Design -> Unstable Double-Pole at the Origin

```
% Youla Control Design

s = tf('s');

% % Constants & Design Parameters
% C_t2 = 500/7939; % Constant
% Wn = 3.25; % Natural Frequency of the Control System
% K = Wn^2/C_t2; % Controller Gain
% Z = 2^-0.5; % Damping Ratio
% tp = 1/(10*Wn); % Time Constant of the added pole
% tz = (4*2^(1/2))/13 + 2/65; % 100*2^(1/2) + 10;

% Plant TF, 'Gp'
Gp_t2 = zpk(minreal(C_t2/s^2))

% Chosen Youla Parameter, 'Y' -> Y(0) = 0
Y_t2 = zpk(minreal(((K_t2*s^2)*(tz*s + 1)/((s^2 + 2*Z*Wn*s +
 Wn^2)*(tp_t2*s + 1))),1e-04))

% Complementary Sensitivity TF, 'T' -> T(0) = 1 (1st interpolation
% condition)
T_t2 = zpk(minreal((Y_t2*Gp_t2),1e-04))

% Sensitivity TF, 'S'
S_t2 = zpk(minreal((1-T_t2),1e-04))

% Controller TF, 'Gc'
Gc_t2 = zpk(minreal((Y_t2/S_t2),1e-04))

% Return Ratio, 'L'
L_t2 = zpk(minreal((Gc_t2*Gp_t2),1e-04))
```

```
GpS_t2 = zpk(minreal((Gp_t2*S_t2),1e-04))

% Internal stability check
Y_t2_stability = isstable(Y_t2)
T_t2_stability = isstable(T_t2)
S_t2_stability = isstable(S_t2)
GpS_t2_stability = isstable(GpS_t2)

M2_t2 = 1/getPeakGain(S_t2) % M2-margin
BW_t2 = bandwidth(T_t2) % Bandwidth of the closed-loop
AE_t2 = getPeakGain(Y_t2) % Maximum actuator effort

figure(1)
bodemag(Y_t2, S_t2, T_t2);
legend('Y_t2','S_t2','T_t2');

Gc_2 = Gc_t2 * eye(3);
```

```
Gp_t2 =

  0.06298
  -------
    s^2

Continuous-time zero/pole/gain model.


Y_t2 =

        2539.5 s^2 (s+2.146)
  -------------------------------
  (s+32.5) (s^2 + 4.596s + 10.56)

Continuous-time zero/pole/gain model.


T_t2 =

          159.94 (s+2.146)
  -------------------------------
  (s+32.5) (s^2 + 4.596s + 10.56)

Continuous-time zero/pole/gain model.


S_t2 =

  (s+1.073e-07) (s-1.073e-07) (s+37.1)
  ------------------------------------
    (s+32.5) (s^2 + 4.596s + 10.56)

Continuous-time zero/pole/gain model.
```

```
Gc_t2 =

  2539.5 (s+2.146)
  ----------------
      (s+37.1)

Continuous-time zero/pole/gain model.


L_t2 =

  159.94 (s+2.146)
  ----------------
    s^2 (s+37.1)

Continuous-time zero/pole/gain model.


GpS_t2 =

         0.06298 (s+37.1)
  ------------------------------
  (s+32.5) (s^2 + 4.596s + 10.56)

Continuous-time zero/pole/gain model.

Y_t2_stability =
  logical
   1
T_t2_stability =
  logical
   1
S_t2_stability =
  logical
   1
GpS_t2_stability =
  logical
   1
M2_t2 =
    0.8909
BW_t2 =
    6.9502
AE_t2 =
   2.5395e+03
```



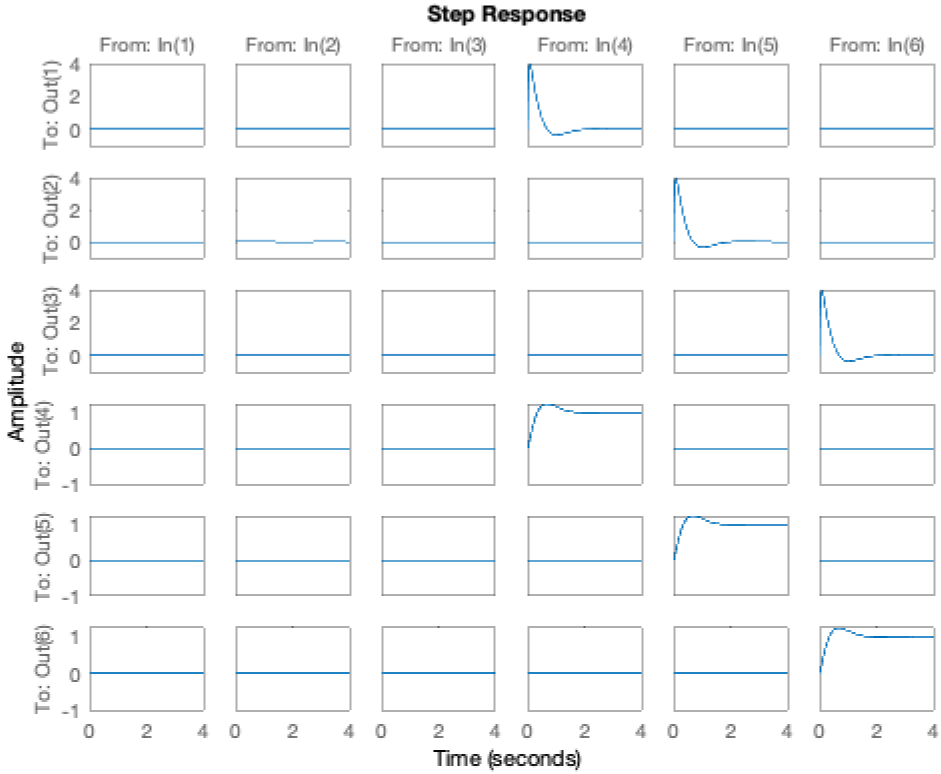
Simulation

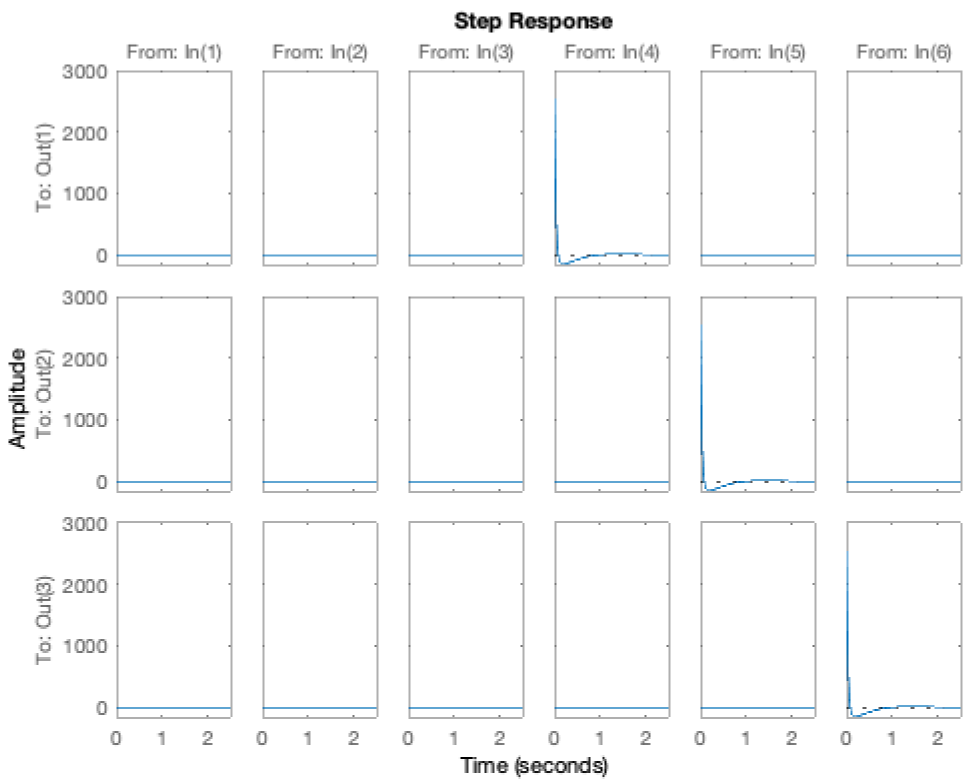
```
Gp = minreal([tf_full(1:3, 1:3); tf_full(7:9, 1:3)], 1e-04);
Gc = [Gc_1 Gc_2]
Lu = minreal(Gc * Gp, 1e-04);
Ly = minreal(Gp * Gc, 1e-04);
Y = minreal(inv(eye(3) + Lu) * Gc);
Ty = minreal(inv(eye(6) + Ly) * Ly);
Sy = minreal(inv(eye(6) + Ly), 1e-04);
Su = minreal(inv(eye(3) + Lu), 1e-04);

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);
```

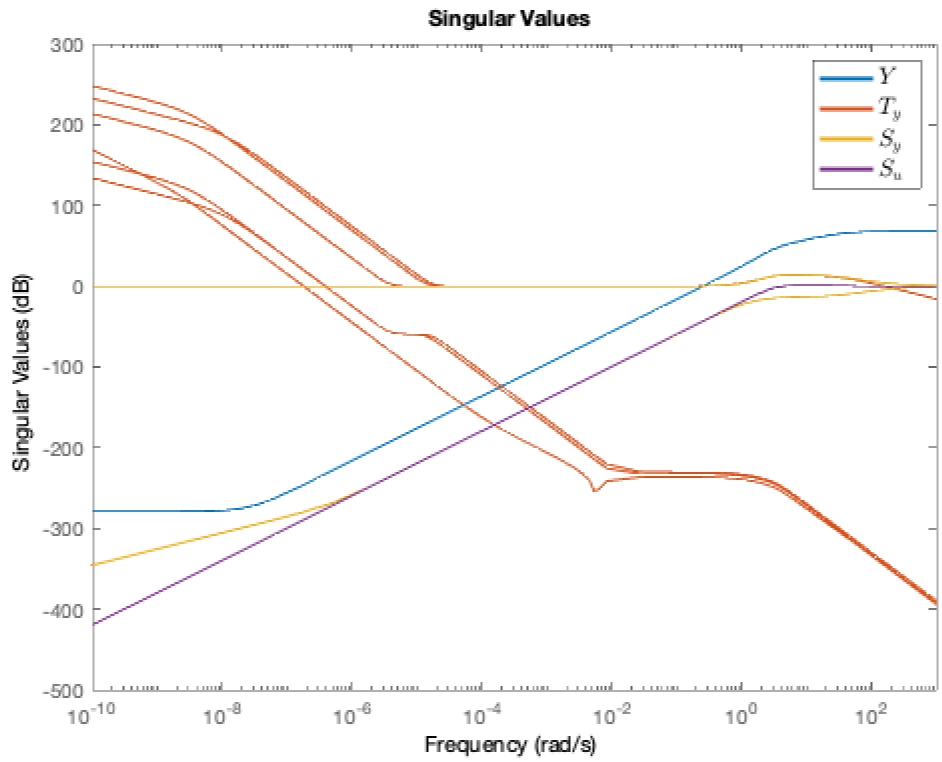
```
figure
sigma(Gc, Gp, Ly, Y)
[l, hObj] = legend('$G_{c}$', '$G_{p}$', '$L_{y}$', '$Y
$','Interpreter','latex','FontSize', 12);
set(l,'string',{'$G_{c}$', '$G_{p}$', '$L_{y}$', '$Y$'});
hL = findobj(hObj,'type','line');
set(hL,'linewidth', 2);

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);
```

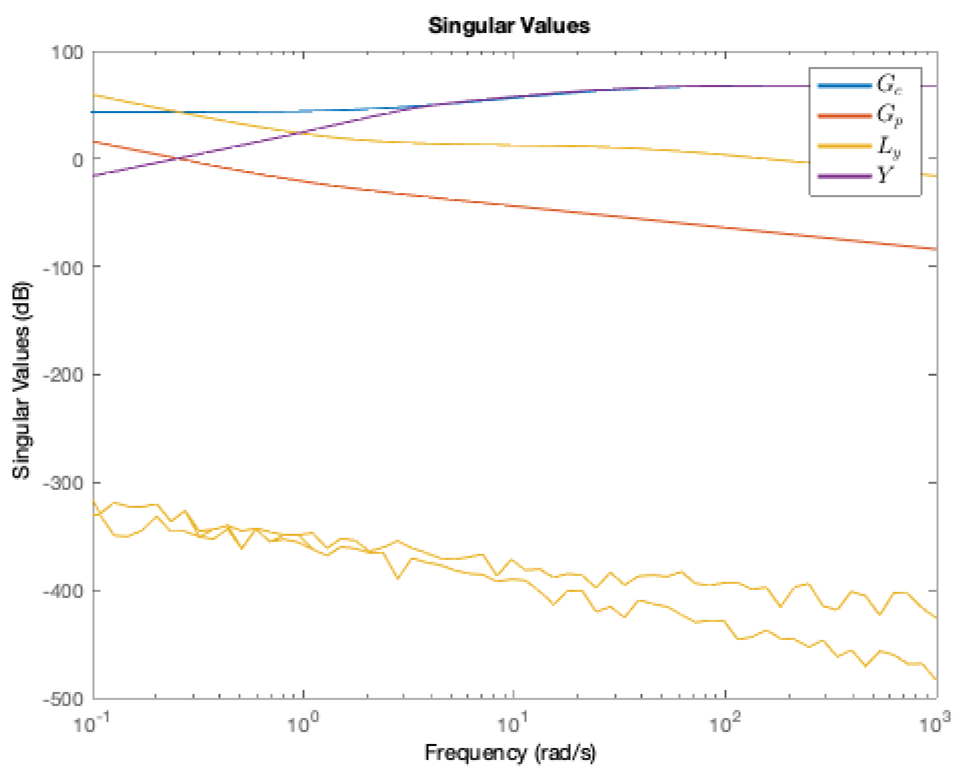
```
Gc =

  From input 1 to output...
   1:  0.15878

   2:  0

   3:  0

  From input 2 to output...
   1:  0

   2:  0.15878

   3:  0

  From input 3 to output...
   1:  0
```

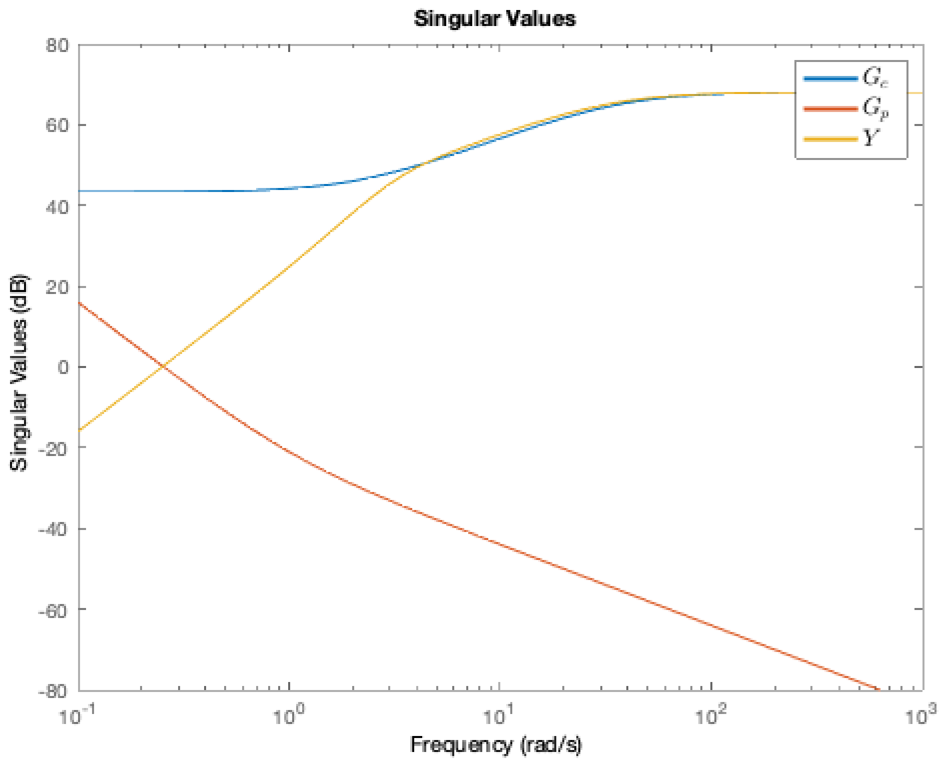
```
 2:  0

 3:  0.15878

From input 4 to output...
     2539.5 (s+2.146)
 1:  ----------------
         (s+37.1)

 2:  0

 3:  0

From input 5 to output...
 1:  0

     2539.5 (s+2.146)
 2:  ----------------
         (s+37.1)

 3:  0

From input 6 to output...
 1:  0

 2:  0

     2539.5 (s+2.146)
 3:  ----------------
         (s+37.1)

Continuous-time zero/pole/gain model.
```

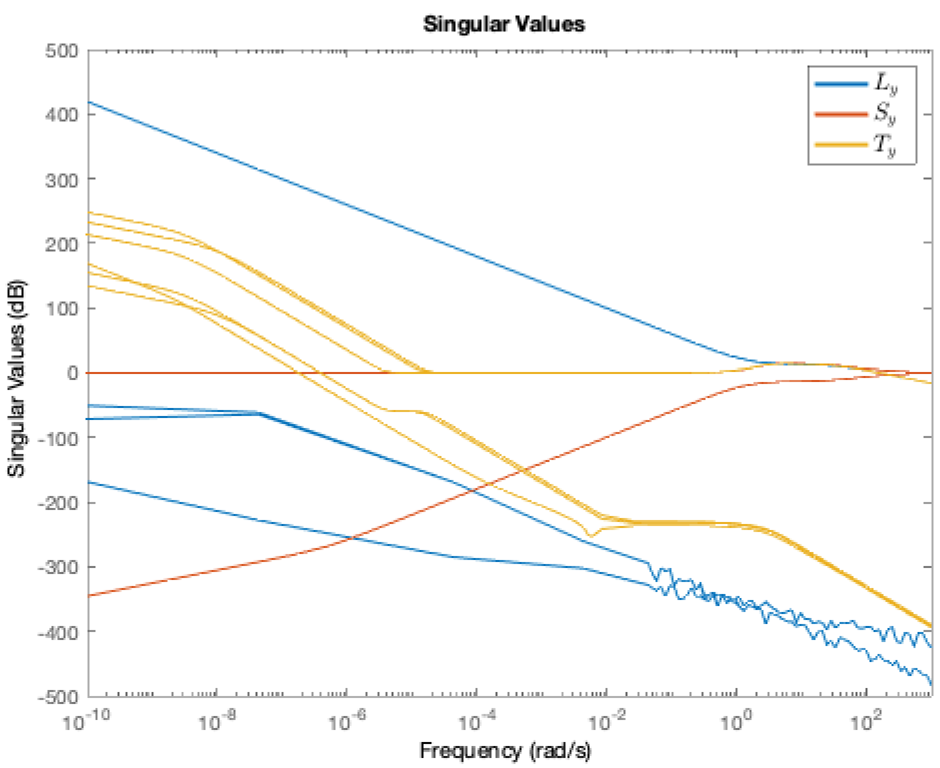

Step Response
From: In(1)
From: In(2)
From: In(3)
From: In(4)
From: In(5)
From: In(6)
Amplitude
To: Out(1)
To: Out(2)
To: Out(3)
To: Out(4)
To: Out(5)
To: Out(6)
Time (seconds)

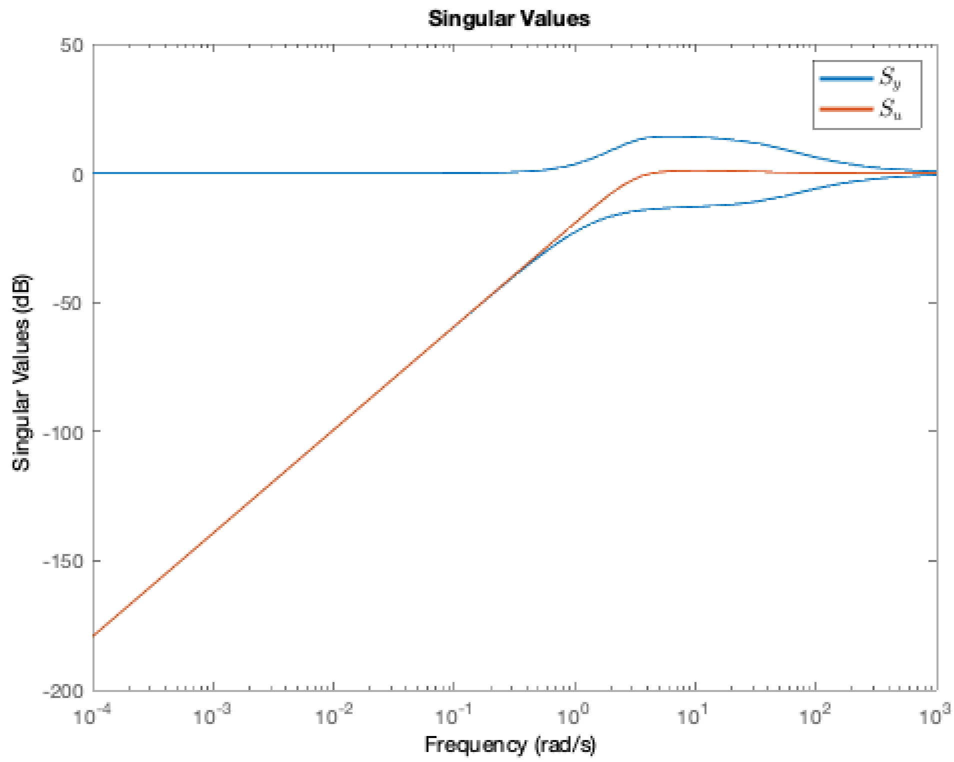

Step Response
From: In(1)
From: In(2)
From: In(3)
From: In(4)
From: In(5)
From: In(6)
Amplitude
To: Out(1)
To: Out(2)
To: Out(3)
Time (seconds)


Singular Values
Singular Values (dB)
Frequency (rad/s)
Y
T_y
S_y
S_u
300
200
100
0
-100
-200
-300
-400
-500
10^{-10}
10^{-8}
10^{-6}
10^{-4}
10^{-2}
10^{0}
10^{2}


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


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


Singular Values
L_y
S_y
T_y
Singular Values (dB)
Frequency (rad/s)
500
400
300
200
100
0
-100
-200
-300
-400
-500
10^{-10}
10^{-8}
10^{-6}
10^{-4}
10^{-2}
10^{0}
10^{2}



Coordinate Feedback

```
% Cc = [zeros(6, 12)];
% Cc(1:6, 1:6) = eye(6);
%
% Dc = [zeros(6, 6)];
%
% sys_coord = ss(A, B, Cc, Dc);
%
% tf_coord = tf(sys_coord);
%
% syms s
%
% tf_coord_sym = simplify(Cc * inv(s * eye(12) - A) * B + Dc);
% pretty(tf_coord_sym)
%
% translation_coord = [tf_coord_sym(1:3, 1:3); tf_coord_sym(7:9,
 1:3)];
% pretty(translation_coord)
%
% attitude_coord = [tf_coord_sym(4:6, 4:6); tf_coord_sym(10:12, 4:6)];
% pretty(attitude_coord)
```

Published with MATLAB® R2019b