Table of Contents


```
% Hinf example problem

w=logspace(-3,3);

s=tf('s');

%
Gtf=Gp;

Gtf=minreal(Gtf);
Gtfss=ss(Gtf);
[Ag,Bg,Cg,Dg]=ssdata(Gtfss);
[Ag,Bg,Cg,Dg]=minreal(Ag,Bg,Cg,Dg);

% Hinf does not work well, as Hinf norm is not defined on the jw-axis,
 with poles and zeros of Gp on the jw-axis,
% So we use a simple shifting trick to make the Ag BIBO stable.

Ag1=Ag-0.05*eye(6);

%define Wd (Robust uncertainty filter)

Wdtf=[makeweight(0.01,5,1000) 0 0 0 0 0; 0 makeweight(0.01,5,1000) 0 0
 0 0; 0 0 makeweight(0.01,5,1000) 0 0 0; 0 0 0 makeweight(0.01,5,1000)
 0 0; 0 0 0 0 makeweight(0.01,5,1000) 0; 0 0 0 0 0
 makeweight(0.01,5,1000)];
```

3 states removed.

By students, define a new robust uncertanity filter

```
Wdss=ss(Wdtf);
[Ad,Bd,Cd,Dd]=ssdata(Wdss);
[Ad,Bd,Cd,Dd]=minreal(Ad,Bd,Cd,Dd);

%define Wu (Actuator filter)

Wutf=[0.01 0 0; 0 0.01 0; 0 0 0.01];
Wuss=ss(Wutf);
[Au,Bu,Cu,Du]=ssdata(Wuss);
```

```
[Au,Bu,Cu,Du]=minreal(Au,Bu,Cu,Du);

%define Wp (Performance filter)

Wptf=[makeweight(1000,0.5,0.1) 0 0 0 0 0; 0 makeweight(1000,0.5,0.1)
 0 0 0 0; 0 0 makeweight(1000,0.5,0.1) 0 0 0; 0 0 0
 makeweight(1000,0.5,0.1) 0 0; 0 0 0 0 makeweight(1000,0.5,0.1) 0; 0 0
 0 0 0 makeweight(1000,0.5,0.1)];
```

By students, defina a new performance filter

```
Wpss=ss(Wptf); [Ap,Bp,Cp,Dp]=ssdata(Wpss);
[Ap,Bp,Cp,Dp]=minreal(Ap,Bp,Cp,Dp);

%compute augmented plant
[A,B1,B2,C1,C2,D11,D12,D21,D22]=...
    augss(Ag1,Bg,Cg,Dg,...
          Ap,Bp,Cp,Dp,...
          Au,Bu,Cu,Du,...
          Ad,Bd,Cd,Dd);



%Compute Hinf controller
[Gamma,acp,bcp,ccp,dcp,acl,bcl,ccl,dcl]=...
    hinfopt(A,B1,B2,C1,C2,D11,D12,D21,D22);

% We have to shift back the A matrix of the controller to make this
% approach work properly.
acp1=acp+0.05*eye(size(acp));
```

```
                    << H-Infinity Optimal Control Synthesis >>

  No      Gamma     D11<=1    P-Exist    P>=0    S-Exist    S>=0    lam(PS)<1
    C.L.
 ---------------------------------------------------------------------------
   1  1.0000e+00    OK       FAIL       FAIL      OK        OK         OK
    STAB
   2  5.0000e-01    OK       FAIL       FAIL      OK        OK         OK
    STAB
   3  2.5000e-01    OK       FAIL       FAIL      OK        OK         OK
    UNST
   4  1.2500e-01    OK       FAIL       FAIL      OK        OK         OK
    STAB
   5  6.2500e-02    OK       FAIL        OK       OK        OK         OK
    STAB
   6  3.1250e-02    OK       FAIL        OK       OK        OK         OK
    STAB
   7  1.5625e-02    OK       FAIL        OK       OK        OK         OK
    STAB
   8  7.8125e-03    OK       FAIL        OK       OK        OK         OK
    STAB
```

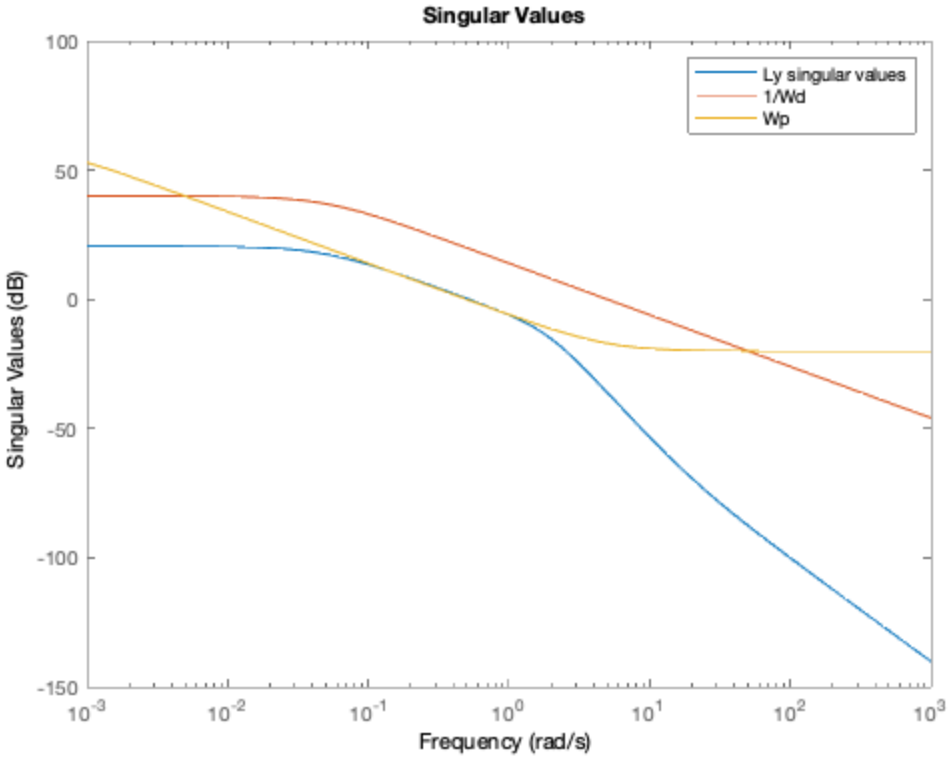
```
   9  3.9062e-03    OK       FAIL       OK       OK       OK       OK
     STAB
  10  1.9531e-03    OK       FAIL       OK       OK       OK       OK
     STAB
  11  9.7656e-04    OK        OK        OK       OK       OK       OK
     STAB
  12  1.4648e-03    OK       FAIL       OK       OK       OK       OK
     STAB
  13  1.2207e-03    OK       FAIL       OK       OK       OK       OK
     STAB
  14  1.0986e-03    OK       FAIL       OK       OK       OK       OK
     STAB
  15  1.0376e-03    OK       FAIL       OK       OK       OK       OK
     STAB
  16  1.0071e-03    OK       FAIL       OK       OK       OK       OK
     STAB
  17  9.9182e-04    OK        OK        OK       OK       OK       OK
     STAB
  18  9.9945e-04    OK        OK        OK       OK       OK       OK
     STAB

      Iteration no. 18 is your best answer under the tolerance:
 0.0100 .
```

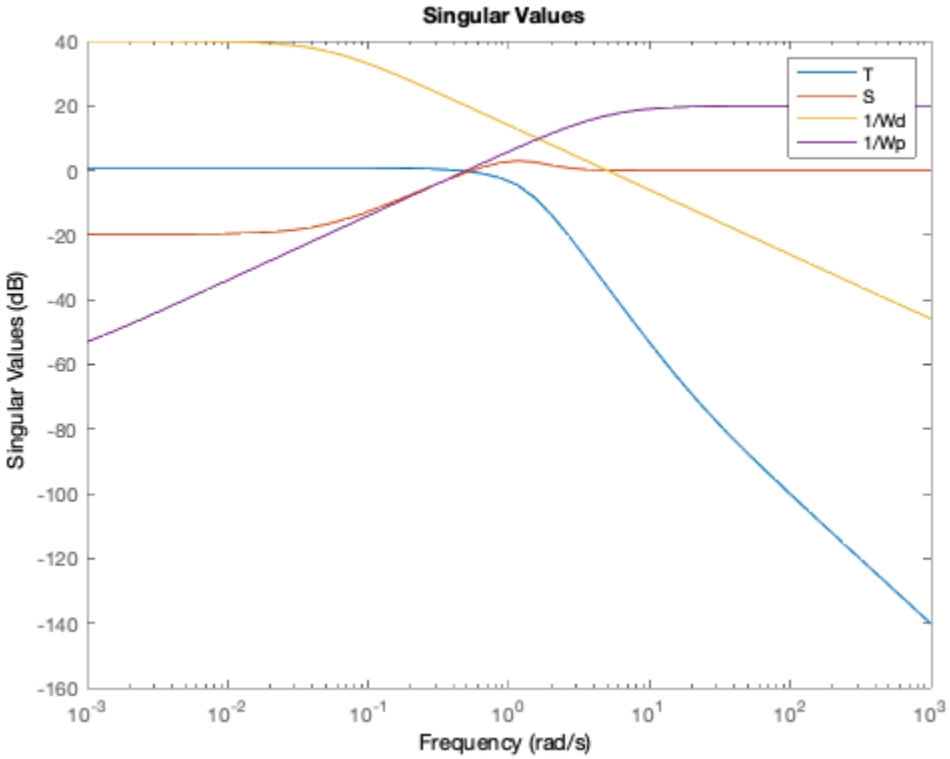
By students, compute Tzw using lft command

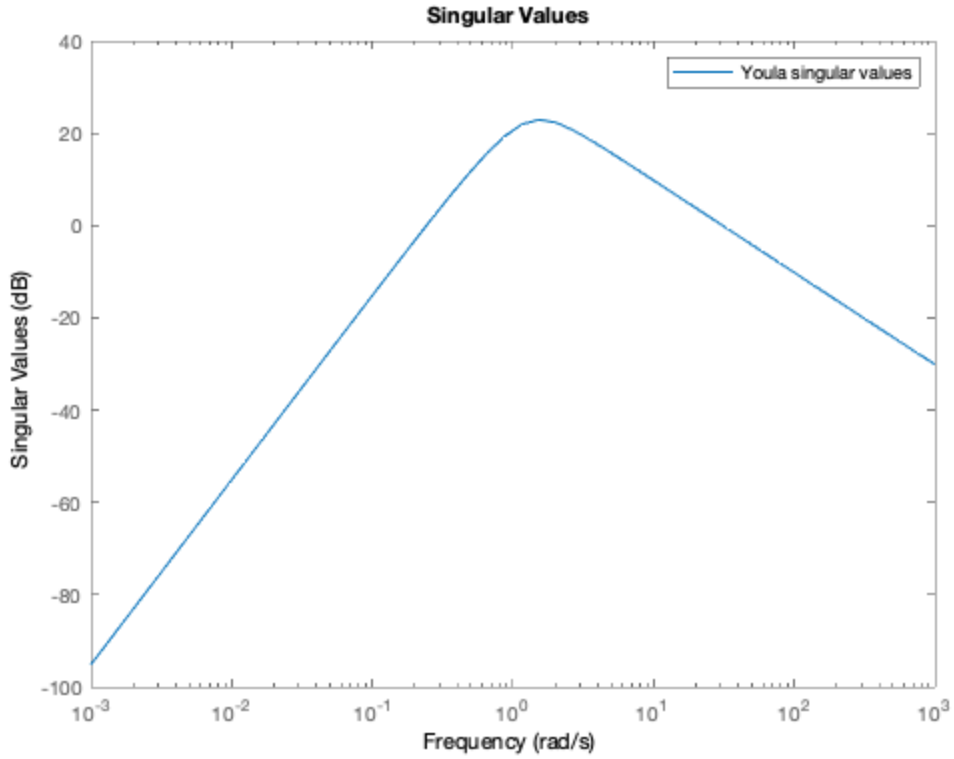
```
% open loop
[aly,bly,cly,dly]=series(Ag,Bg,Cg,Dg,acp1,bcp,ccp,dcp);
[aly,bly,cly,dly]=minreal(aly,bly,cly,dly);

%closed loop
[at,bt,ct,dt]=cloop(aly,bly,cly,dly,-1);
[at,bt,ct,dt]=minreal(at,bt,ct,dt);
as=at;
bs=bt;
cs=-ct;
ds=eye(3)-dt;

[ay,by,cy,dy]=series(acp1,bcp,ccp,dcp,as,bs,cs,ds);
[ay,by,cy,dy]=minreal(ay,by,cy,dy);

figure(1);
sigma(aly,bly,cly,dly)
hold on
sigma(inv(Wdtf(2,2)),w)
hold on
sigma(Wptf(1,1),w)
legend('Ly singular values','1/Wd','Wp')


figure(2)
sigma(at,bt,ct,dt,w)
hold on
```

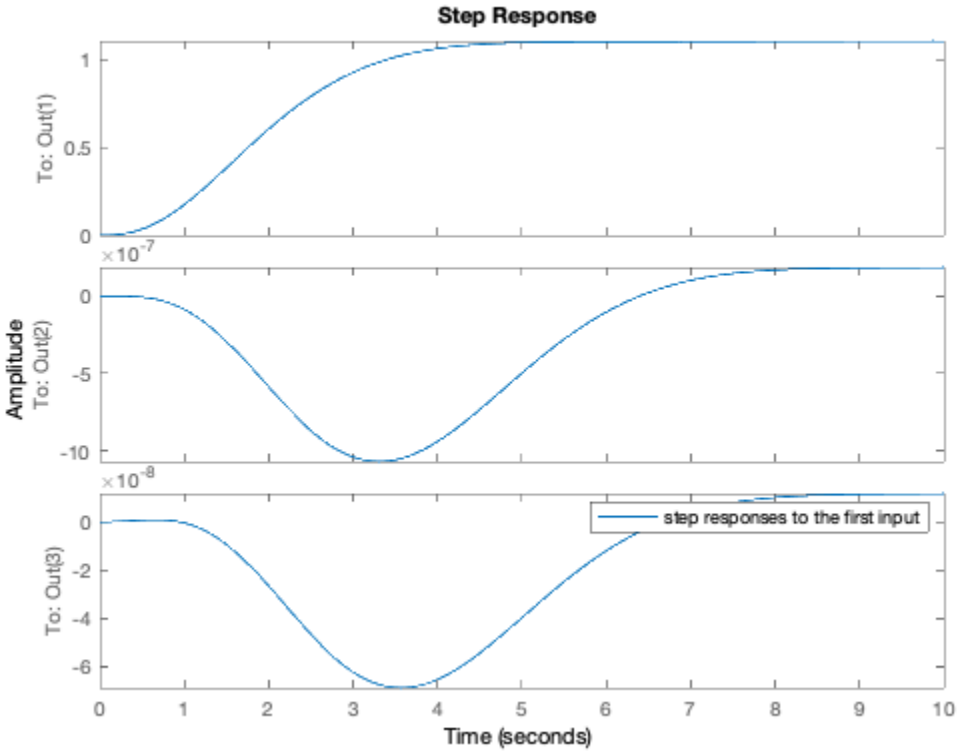
```
sigma(as,bs,cs,ds,w)
hold on
sigma(inv(Wdtf(2,2)),w)
hold on
sigma(inv(Wptf(1,1)),w)
legend('T','S','1/Wd','1/Wp')
hold off


figure(3)
sigma(ay,by,cy,dy,w)
legend('Youla singular values')

%closed loop time response
t=0:0.01:10;
figure(4)
step(at,bt,ct,dt,1,t)
legend('step responses to the first input')
figure(5)
step(at,bt,ct,dt,2,t)
legend('step responses to the second input')


figure(6)
sigma(Gtf)
legend('Plant singular values')

figure (7)
```

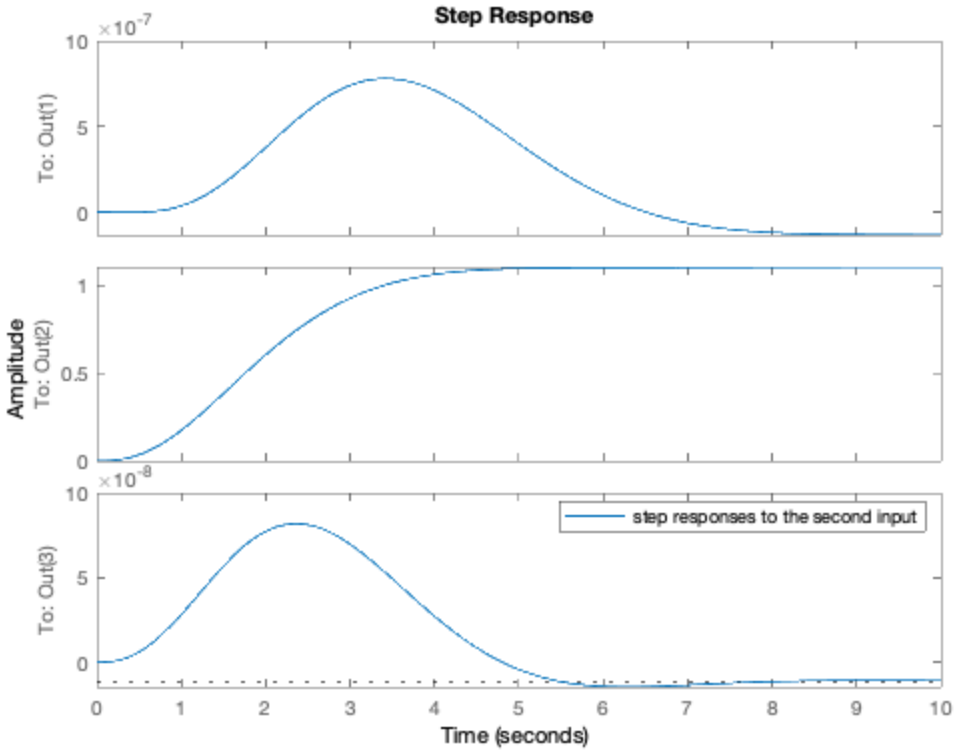
```
12 states removed.
12 states removed.
```

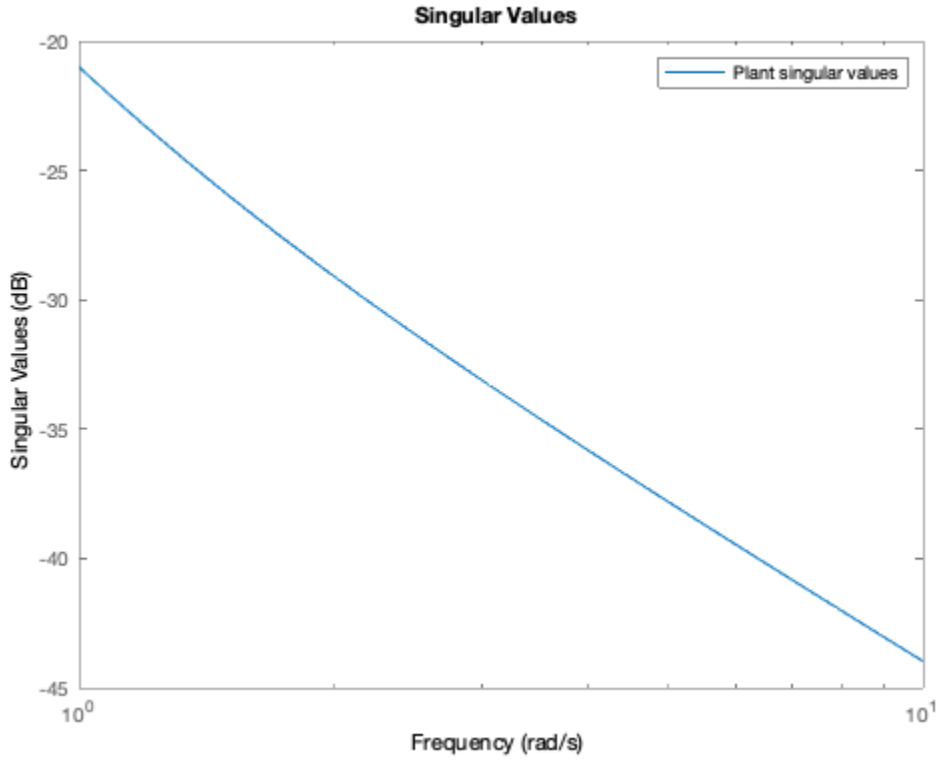

Singular Values
Ly singular values
1/Wd
Wp
Singular Values (dB)
Frequency (rad/s)


Singular Values
T
S
1/Wd
1/Wp
Singular Values (dB)
Frequency (rad/s)


Singular Values
Youla singular values
Singular Values (dB)
Frequency (rad/s)

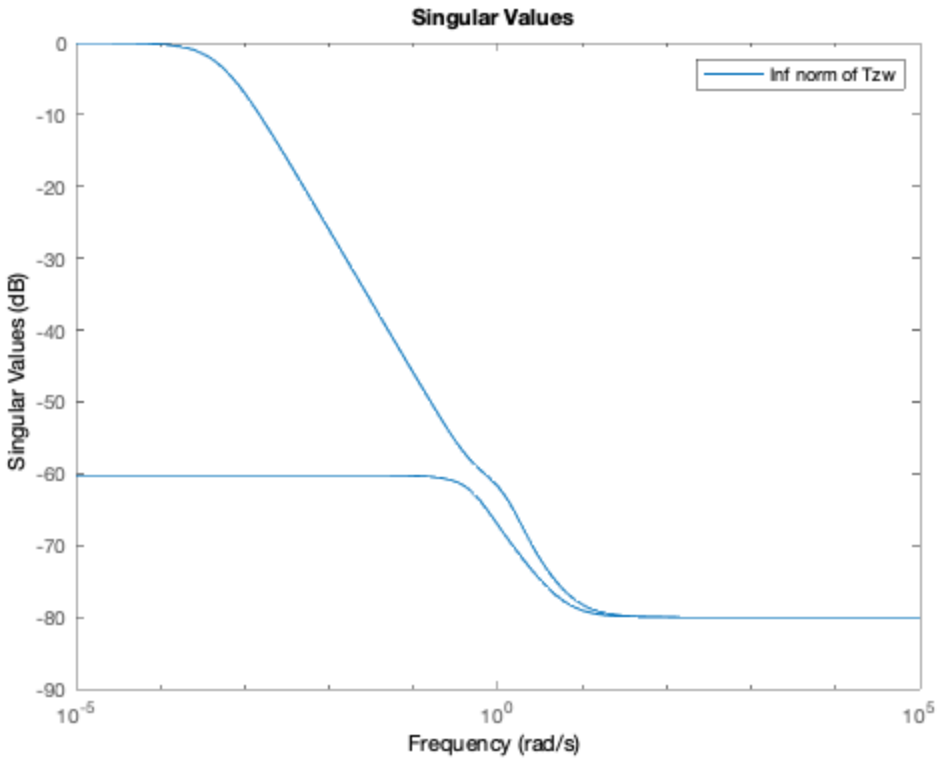

Step Response
To: Out(1)
To: Out(2)
To: Out(3)
Amplitude
step responses to the first input
Time (seconds)


Step Response
×10^-7
10
5
0
To: Out(1)
1
0.5
0
To: Out(2)
Amplitude
×10^-8
10
5
0
To: Out(3)
step responses to the second input
0
1
2
3
4
5
6
7
8
9
10
Time (seconds)


Singular Values
-20
-25
-30
-35
-40
-45
Singular Values (dB)
Plant singular values
10^0
10^1
Frequency (rad/s)

By students, compute the maximum sigular value of Tzw from lft computation

```
%sigma(Tzw)
hold on
sigma(acl,bcl,ccl,dcl)
legend('Inf norm of Tzw')

Gc_ss=ss(acp,bcp,ccp,dcp);
Gc_tf=tf(Gc_ss);
Gc_tf=minreal(Gc_tf);


% save('Gc_Hinf_ss.mat','Gc_ss')  % function form
%
% save('Gc_Hinf_tf.mat','Gc_tf')  % function form

% Y = minreal(inv(eye(3) + Lu) * Gc);
% Ty = minreal(inv(eye(6) + Ly) * Ly);
% Sy = minreal(inv(eye(6) + Ly), 1e-02);
```



Published with MATLAB® R2019b