ESE 547 Homework 4 / State Feedback H_∞ Optimal Control /

Seif Elkhashab / 02-23-2024 /

Hinf State Feedback Design

1. Using the pitch axis aircraft plant data (same as RSLQR design model in homework 3) design a Hinf state feedback controller to command Az. Include the 2nd order actuator in the design model as presented in class. Tune the Hinf controller to yield a similar rise time as your RSLQR controller.

List out all matrices used in the Hinf design, closed loop eigenvalues, and eigenvectors.

List the controller matrices $(A_c, B_{c1}, B_{c2}, C_c, D_{c1}, D_{c2})$ implementing the Hinf control

$$\dot{x}_c = A_c x_c + B_{c_1} y + B_{c_2} r$$
$$u = C_c x_c + D_{c_1} y + D_{c_2} r$$

where $\mathbf{y} = \mathbf{x_p}$.

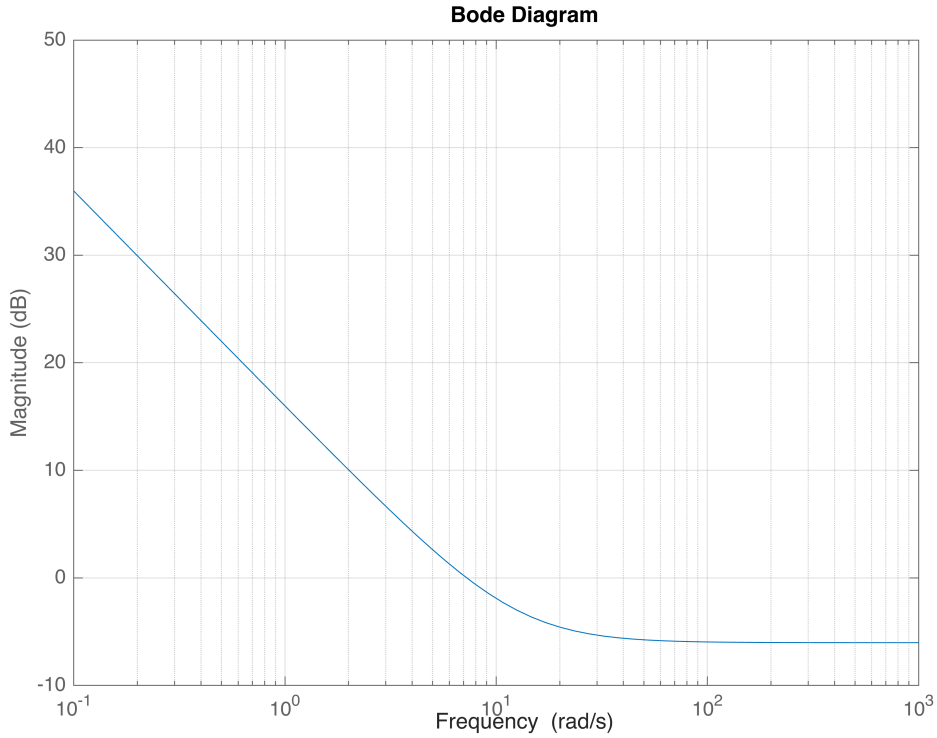
Plant

```
% constants
V = 886.78;
Md = -104.8346;
Ma = 47.7109;
Za = -1.3046*V;
Zd = -0.2142*V;

omega = 11*2*pi;
zeta = .707;

% Used for Plotting Later
dd=0.:.001:2*pi;
xx1=cos(dd)-1;yy1=sin(dd);


% Plant Matricies with the actuator
```

```
Ap = [Za/V 1    Zd/V        0;
       Ma  0     Md         0;
       0   0     0          1;
       0   0 -omega^2 -2*zeta*omega];
Bp = [0;
      0;
      0
      omega^2];
Cp = [Za  0 Zd  0];
Dp = 0.*Cp*Bp;
```

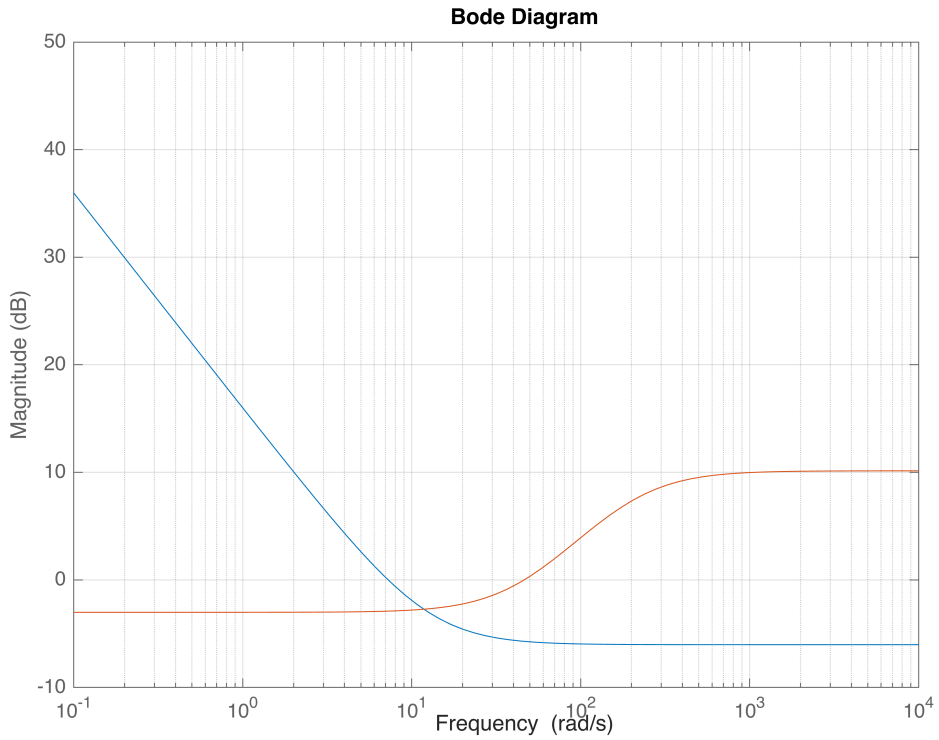
Sensitivity W_s

```
wlgcf = 2; %Hz

Ws.Tau = 1/(2*pi*wlgcf);
Ws.K = 0.5/Ws.Tau;
Ws.num = Ws.K * [Ws.Tau, 1];
Ws.den = [1, 0];
[Ws.A,Ws.B,Ws.C,Ws.D] = tf2ss(Ws.num,Ws.den);

Ws.sys = ss(Ws.A,Ws.B,Ws.C,Ws.D);
WS = [ Ws.A Ws.B; ...
       Ws.C Ws.D ];

figure;
bodemag(Ws.sys)
hold on;
grid on
```



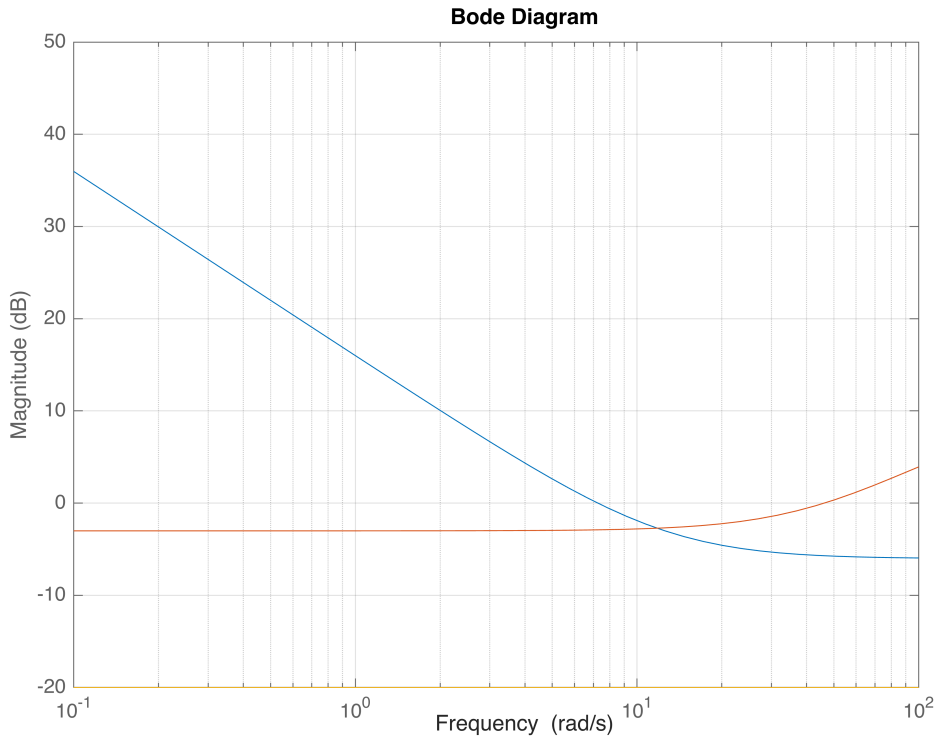
Complementary Sensitivity W_t

```
Wt.TauN = 1/(3.5*2*pi*wlgcf);
Wt.TauD = 0.005;
Wt.K = 0.707;
Wt.num = Wt.K * [Wt.TauN, 1];
Wt.den = [Wt.TauD, 1];
[Wt.A,Wt.B,Wt.C,Wt.D] = tf2ss(Wt.num,Wt.den);

Wt.sys = ss(Wt.A,Wt.B,Wt.C,Wt.D);

bodemag(Wt.sys)
grid on
```



Control Activity

```
Wc = 0.1;
Cc = [0, 0, 0, 1];
Wc_sys = ss(0,0,0,Wc);

bodemag(Wc_sys)
xlim([0.1,100])
hold off
```



Hinf State-Feedback Matrix Setup

```
HinfSF.A = [     Ap, zeros(4,1), zeros(4,1);...
           Ws.B*Cp,       Ws.A,          0;...
           Wt.B*Cp,          0,       Wt.A];

HinfSF.B = [   Bp;...
           Ws.B*Dp;...
           Wt.B*Dp];

HinfSF.E = [zeros(4,1);...
                -Ws.B;...
                    0];

HinfSF.C = [ Ws.D*Cp, Ws.C,    0;...
             Wt.D*Cp,    0, Wt.C;...
           Wc*Cc*Ap,    0,    0];

HinfSF.D1 = [ Ws.D*Dp;...
              Wt.D*Dp;...
             Wc*Cc*Bp];

HinfSF.D2 = [-Ws.D;...
                0;...
                0];
```

```
HinfSF.B_hat = [HinfSF.B, HinfSF.E];

HinfSF.D_hat = [HinfSF.D1, HinfSF.D2];

HinfSF.Q = HinfSF.C'*HinfSF.C;
HinfSF.S = [HinfSF.C'*HinfSF.D1 HinfSF.C'*HinfSF.D2];
```

Gamma Search Algorithm

```
gamma_max = 20;
gamma_min = 1;
gamma = gamma_max;
to_test = 1;

while(abs(gamma_max - gamma_min)>1e-10)
    if to_test == 1
        gamma_max = gamma;
    else
        gamma_min = gamma;
    end
    gamma = (gamma_max + gamma_min)/2;
    disp(['gamma_max = ',num2str(gamma_max,  '%12.10g'),'  gamma =
',num2str(gamma,  '%12.10g'),...
          '  gamma_min = ',num2str(gamma_min,  '%12.10g')])

    HinfSF.R = [HinfSF.D1'*HinfSF.D1 HinfSF.D1'*HinfSF.D2;...
                HinfSF.D2'*HinfSF.D1 HinfSF.D2'*HinfSF.D2 - gamma^2];

    [P,CL_eig,K] = icare(HinfSF.A,HinfSF.B_hat,HinfSF.Q,HinfSF.R,HinfSF.S);

    pos_test = 1;
    ev = eig(P);
    if (isempty(P))
        gamma = gamma + 0.000005;
        disp(['gamma_max = ',num2str(gamma_max,  '%12.10g'),'  gamma =
',num2str(gamma,  '%12.10g'),...
          '  gamma_min = ',num2str(gamma_min,  '%12.10g!!!')])

        HinfSF.R = [HinfSF.D1'*HinfSF.D1 HinfSF.D1'*HinfSF.D2;...
                HinfSF.D2'*HinfSF.D1 HinfSF.D2'*HinfSF.D2 - gamma^2];
        [P,CL_eig,K] =
icare(HinfSF.A,HinfSF.B_hat,HinfSF.Q,HinfSF.R,HinfSF.S);
        break
    end
    for i = 1:6
        if ev(i) < 0
            pos_test = 0;
        end
```

```
    end

    HinfSF.Kxopt = -[1 0]*HinfSF.R^-1*(HinfSF.B_hat'*P + HinfSF.S');
    HinfSF.Aref = HinfSF.A + HinfSF.B*HinfSF.Kxopt;

    eig_test = 1;
    ev_cl = eig(HinfSF.Aref);
    for i = 1:6
        if ev_cl(i) >= 0
            eig_test = 0;
        end
    end
    to_test = pos_test*eig_test;
end
```

```
gamma_max = 20  gamma = 10.5  gamma_min = 1
gamma_max = 10.5  gamma = 5.75  gamma_min = 1
gamma_max = 5.75  gamma = 3.375  gamma_min = 1
gamma_max = 3.375  gamma = 2.1875  gamma_min = 1
gamma_max = 2.1875  gamma = 1.59375  gamma_min = 1
gamma_max = 1.59375  gamma = 1.296875  gamma_min = 1
gamma_max = 1.296875  gamma = 1.1484375  gamma_min = 1
gamma_max = 1.1484375  gamma = 1.07421875  gamma_min = 1
gamma_max = 1.07421875  gamma = 1.037109375  gamma_min = 1
gamma_max = 1.07421875  gamma = 1.055664062  gamma_min = 1.037109375
gamma_max = 1.07421875  gamma = 1.064941406  gamma_min = 1.055664062
gamma_max = 1.07421875  gamma = 1.069580078  gamma_min = 1.064941406
gamma_max = 1.07421875  gamma = 1.071899414  gamma_min = 1.069580078
gamma_max = 1.07421875  gamma = 1.073059082  gamma_min = 1.071899414
gamma_max = 1.07421875  gamma = 1.073638916  gamma_min = 1.073059082
gamma_max = 1.073638916  gamma = 1.073348999  gamma_min = 1.073059082
gamma_max = 1.073348999  gamma = 1.073204041  gamma_min = 1.073059082
gamma_max = 1.073204041  gamma = 1.073131561  gamma_min = 1.073059082
gamma_max = 1.073131561  gamma = 1.073095322  gamma_min = 1.073059082
gamma_max = 1.073131561  gamma = 1.073113441  gamma_min = 1.073095322
gamma_max = 1.073131561  gamma = 1.073122501  gamma_min = 1.073113441
gamma_max = 1.073122501  gamma = 1.073117971  gamma_min = 1.073113441
gamma_max = 1.073122501  gamma = 1.073120236  gamma_min = 1.073117971
gamma_max = 1.073120236  gamma = 1.073119104  gamma_min = 1.073117971
gamma_max = 1.073120236  gamma = 1.073124104  gamma_min = 1.073117971!!!
```

```
HinfSF.Kxref = HinfSF.Kxopt;
disp(['Gamma optimal = ',num2str(gamma,  '%12.10g')])
```

```
Gamma optimal = 1.073124104
```

```
disp(['K optimal = [ ' num2str(HinfSF.Kxopt) ' ]'])
```

```
K optimal = [ 1293128.4925      67119.513158     -98620.326445     -880.20393905     -8357.6784804      11
```

```
Niter = 0;

EE=HinfSF.A'*P+P*HinfSF.A
- (P*HinfSF.B_hat+HinfSF.S)*inv(HinfSF.R)*(HinfSF.B_hat'*P+HinfSF.S')
+HinfSF.Q;
disp(['Norm EE opt = ',num2str(norm(EE))]);
```

```
Norm EE opt = 428.9333
```

```
tuned_gamma = gamma;
while(HinfSF.Kxref(2)>1 && Niter < 200)
    tuned_gamma = tuned_gamma + 0.0005;
    disp(['gamma = ',num2str(tuned_gamma),' Kq =
',num2str(HinfSF.Kxref(2)),...
    ' # of iterations = ', num2str(Niter)])
    HinfSF.R = [HinfSF.D1'*HinfSF.D1 HinfSF.D1'*HinfSF.D2;...
    HinfSF.D2'*HinfSF.D1 HinfSF.D2'*HinfSF.D2 - tuned_gamma^2];
    [P,CL_eig,K] = care(HinfSF.A,HinfSF.B_hat,HinfSF.Q,HinfSF.R,HinfSF.S);
    HinfSF.Kxref = -[1 0]*HinfSF.R^-1*(HinfSF.B_hat'*P + HinfSF.S');
    HinfSF.Aref = HinfSF.A + HinfSF.B*HinfSF.Kxref;
    Niter = Niter + 1;
end
```

```
gamma = 1.0736 Kq = 67119.5132 # of iterations = 0
gamma = 1.0741 Kq = 123.2469 # of iterations = 1
gamma = 1.0746 Kq = 62.0765 # of iterations = 2
gamma = 1.0751 Kq = 41.5565 # of iterations = 3
gamma = 1.0756 Kq = 31.272 # of iterations = 4
gamma = 1.0761 Kq = 25.0934 # of iterations = 5
gamma = 1.0766 Kq = 20.9711 # of iterations = 6
gamma = 1.0771 Kq = 18.025 # of iterations = 7
gamma = 1.0776 Kq = 15.8145 # of iterations = 8
gamma = 1.0781 Kq = 14.0947 # of iterations = 9
gamma = 1.0786 Kq = 12.7185 # of iterations = 10
gamma = 1.0791 Kq = 11.5924 # of iterations = 11
gamma = 1.0796 Kq = 10.6537 # of iterations = 12
gamma = 1.0801 Kq = 9.8594 # of iterations = 13
gamma = 1.0806 Kq = 9.1785 # of iterations = 14
gamma = 1.0811 Kq = 8.5883 # of iterations = 15
gamma = 1.0816 Kq = 8.0718 # of iterations = 16
gamma = 1.0821 Kq = 7.6161 # of iterations = 17
gamma = 1.0826 Kq = 7.211 # of iterations = 18
gamma = 1.0831 Kq = 6.8485 # of iterations = 19
gamma = 1.0836 Kq = 6.5222 # of iterations = 20
gamma = 1.0841 Kq = 6.227 # of iterations = 21
gamma = 1.0846 Kq = 5.9586 # of iterations = 22
gamma = 1.0851 Kq = 5.7135 # of iterations = 23
gamma = 1.0856 Kq = 5.4889 # of iterations = 24
gamma = 1.0861 Kq = 5.2822 # of iterations = 25
gamma = 1.0866 Kq = 5.0914 # of iterations = 26
gamma = 1.0871 Kq = 4.9148 # of iterations = 27
gamma = 1.0876 Kq = 4.7507 # of iterations = 28
gamma = 1.0881 Kq = 4.598 # of iterations = 29
gamma = 1.0886 Kq = 4.4554 # of iterations = 30
gamma = 1.0891 Kq = 4.3221 # of iterations = 31
gamma = 1.0896 Kq = 4.197 # of iterations = 32
gamma = 1.0901 Kq = 4.0796 # of iterations = 33
gamma = 1.0906 Kq = 3.969 # of iterations = 34
gamma = 1.0911 Kq = 3.8648 # of iterations = 35
gamma = 1.0916 Kq = 3.7664 # of iterations = 36
gamma = 1.0921 Kq = 3.6732 # of iterations = 37
gamma = 1.0926 Kq = 3.585 # of iterations = 38
gamma = 1.0931 Kq = 3.5013 # of iterations = 39
gamma = 1.0936 Kq = 3.4218 # of iterations = 40
gamma = 1.0941 Kq = 3.3462 # of iterations = 41
gamma = 1.0946 Kq = 3.2741 # of iterations = 42
```

```
gamma = 1.0951 Kq = 3.2054 # of iterations = 43
gamma = 1.0956 Kq = 3.1399 # of iterations = 44
gamma = 1.0961 Kq = 3.0772 # of iterations = 45
gamma = 1.0966 Kq = 3.0173 # of iterations = 46
gamma = 1.0971 Kq = 2.9599 # of iterations = 47
gamma = 1.0976 Kq = 2.9049 # of iterations = 48
gamma = 1.0981 Kq = 2.8522 # of iterations = 49
gamma = 1.0986 Kq = 2.8016 # of iterations = 50
gamma = 1.0991 Kq = 2.7529 # of iterations = 51
gamma = 1.0996 Kq = 2.7061 # of iterations = 52
gamma = 1.1001 Kq = 2.6611 # of iterations = 53
gamma = 1.1006 Kq = 2.6178 # of iterations = 54
gamma = 1.1011 Kq = 2.576 # of iterations = 55
gamma = 1.1016 Kq = 2.5357 # of iterations = 56
gamma = 1.1021 Kq = 2.4969 # of iterations = 57
gamma = 1.1026 Kq = 2.4593 # of iterations = 58
gamma = 1.1031 Kq = 2.4231 # of iterations = 59
gamma = 1.1036 Kq = 2.388 # of iterations = 60
gamma = 1.1041 Kq = 2.3541 # of iterations = 61
gamma = 1.1046 Kq = 2.3213 # of iterations = 62
gamma = 1.1051 Kq = 2.2896 # of iterations = 63
gamma = 1.1056 Kq = 2.2588 # of iterations = 64
gamma = 1.1061 Kq = 2.229 # of iterations = 65
gamma = 1.1066 Kq = 2.2001 # of iterations = 66
gamma = 1.1071 Kq = 2.172 # of iterations = 67
gamma = 1.1076 Kq = 2.1448 # of iterations = 68
gamma = 1.1081 Kq = 2.1183 # of iterations = 69
gamma = 1.1086 Kq = 2.0927 # of iterations = 70
gamma = 1.1091 Kq = 2.0677 # of iterations = 71
gamma = 1.1096 Kq = 2.0434 # of iterations = 72
gamma = 1.1101 Kq = 2.0198 # of iterations = 73
gamma = 1.1106 Kq = 1.9968 # of iterations = 74
gamma = 1.1111 Kq = 1.9745 # of iterations = 75
gamma = 1.1116 Kq = 1.9527 # of iterations = 76
gamma = 1.1121 Kq = 1.9315 # of iterations = 77
gamma = 1.1126 Kq = 1.9109 # of iterations = 78
gamma = 1.1131 Kq = 1.8907 # of iterations = 79
gamma = 1.1136 Kq = 1.8711 # of iterations = 80
gamma = 1.1141 Kq = 1.852 # of iterations = 81
gamma = 1.1146 Kq = 1.8333 # of iterations = 82
gamma = 1.1151 Kq = 1.8151 # of iterations = 83
gamma = 1.1156 Kq = 1.7973 # of iterations = 84
gamma = 1.1161 Kq = 1.7799 # of iterations = 85
gamma = 1.1166 Kq = 1.7629 # of iterations = 86
gamma = 1.1171 Kq = 1.7463 # of iterations = 87
gamma = 1.1176 Kq = 1.7301 # of iterations = 88
gamma = 1.1181 Kq = 1.7143 # of iterations = 89
gamma = 1.1186 Kq = 1.6988 # of iterations = 90
gamma = 1.1191 Kq = 1.6836 # of iterations = 91
gamma = 1.1196 Kq = 1.6688 # of iterations = 92
gamma = 1.1201 Kq = 1.6543 # of iterations = 93
gamma = 1.1206 Kq = 1.6401 # of iterations = 94
gamma = 1.1211 Kq = 1.6262 # of iterations = 95
gamma = 1.1216 Kq = 1.6126 # of iterations = 96
gamma = 1.1221 Kq = 1.5993 # of iterations = 97
gamma = 1.1226 Kq = 1.5863 # of iterations = 98
gamma = 1.1231 Kq = 1.5735 # of iterations = 99
gamma = 1.1236 Kq = 1.5609 # of iterations = 100
gamma = 1.1241 Kq = 1.5487 # of iterations = 101
gamma = 1.1246 Kq = 1.5366 # of iterations = 102
gamma = 1.1251 Kq = 1.5248 # of iterations = 103
gamma = 1.1256 Kq = 1.5132 # of iterations = 104
gamma = 1.1261 Kq = 1.5019 # of iterations = 105
gamma = 1.1266 Kq = 1.4907 # of iterations = 106
```

```
gamma = 1.1271 Kq = 1.4798 # of iterations = 107
gamma = 1.1276 Kq = 1.469 # of iterations = 108
gamma = 1.1281 Kq = 1.4585 # of iterations = 109
gamma = 1.1286 Kq = 1.4482 # of iterations = 110
gamma = 1.1291 Kq = 1.438 # of iterations = 111
gamma = 1.1296 Kq = 1.428 # of iterations = 112
gamma = 1.1301 Kq = 1.4182 # of iterations = 113
gamma = 1.1306 Kq = 1.4086 # of iterations = 114
gamma = 1.1311 Kq = 1.3991 # of iterations = 115
gamma = 1.1316 Kq = 1.3898 # of iterations = 116
gamma = 1.1321 Kq = 1.3807 # of iterations = 117
gamma = 1.1326 Kq = 1.3717 # of iterations = 118
gamma = 1.1331 Kq = 1.3629 # of iterations = 119
gamma = 1.1336 Kq = 1.3542 # of iterations = 120
gamma = 1.1341 Kq = 1.3456 # of iterations = 121
gamma = 1.1346 Kq = 1.3372 # of iterations = 122
gamma = 1.1351 Kq = 1.329 # of iterations = 123
gamma = 1.1356 Kq = 1.3208 # of iterations = 124
gamma = 1.1361 Kq = 1.3128 # of iterations = 125
gamma = 1.1366 Kq = 1.3049 # of iterations = 126
gamma = 1.1371 Kq = 1.2972 # of iterations = 127
gamma = 1.1376 Kq = 1.2896 # of iterations = 128
gamma = 1.1381 Kq = 1.282 # of iterations = 129
gamma = 1.1386 Kq = 1.2746 # of iterations = 130
gamma = 1.1391 Kq = 1.2674 # of iterations = 131
gamma = 1.1396 Kq = 1.2602 # of iterations = 132
gamma = 1.1401 Kq = 1.2531 # of iterations = 133
gamma = 1.1406 Kq = 1.2462 # of iterations = 134
gamma = 1.1411 Kq = 1.2393 # of iterations = 135
gamma = 1.1416 Kq = 1.2325 # of iterations = 136
gamma = 1.1421 Kq = 1.2259 # of iterations = 137
gamma = 1.1426 Kq = 1.2193 # of iterations = 138
gamma = 1.1431 Kq = 1.2129 # of iterations = 139
gamma = 1.1436 Kq = 1.2065 # of iterations = 140
gamma = 1.1441 Kq = 1.2002 # of iterations = 141
gamma = 1.1446 Kq = 1.194 # of iterations = 142
gamma = 1.1451 Kq = 1.1879 # of iterations = 143
gamma = 1.1456 Kq = 1.1819 # of iterations = 144
gamma = 1.1461 Kq = 1.1759 # of iterations = 145
gamma = 1.1466 Kq = 1.1701 # of iterations = 146
gamma = 1.1471 Kq = 1.1643 # of iterations = 147
gamma = 1.1476 Kq = 1.1586 # of iterations = 148
gamma = 1.1481 Kq = 1.153 # of iterations = 149
gamma = 1.1486 Kq = 1.1474 # of iterations = 150
gamma = 1.1491 Kq = 1.1419 # of iterations = 151
gamma = 1.1496 Kq = 1.1365 # of iterations = 152
gamma = 1.1501 Kq = 1.1312 # of iterations = 153
gamma = 1.1506 Kq = 1.1259 # of iterations = 154
gamma = 1.1511 Kq = 1.1207 # of iterations = 155
gamma = 1.1516 Kq = 1.1156 # of iterations = 156
gamma = 1.1521 Kq = 1.1105 # of iterations = 157
gamma = 1.1526 Kq = 1.1055 # of iterations = 158
gamma = 1.1531 Kq = 1.1006 # of iterations = 159
gamma = 1.1536 Kq = 1.0957 # of iterations = 160
gamma = 1.1541 Kq = 1.0909 # of iterations = 161
gamma = 1.1546 Kq = 1.0862 # of iterations = 162
gamma = 1.1551 Kq = 1.0815 # of iterations = 163
gamma = 1.1556 Kq = 1.0768 # of iterations = 164
gamma = 1.1561 Kq = 1.0722 # of iterations = 165
gamma = 1.1566 Kq = 1.0677 # of iterations = 166
gamma = 1.1571 Kq = 1.0632 # of iterations = 167
gamma = 1.1576 Kq = 1.0588 # of iterations = 168
gamma = 1.1581 Kq = 1.0544 # of iterations = 169
gamma = 1.1586 Kq = 1.0501 # of iterations = 170
```

```
gamma = 1.1591 Kq = 1.0459 # of iterations = 171
gamma = 1.1596 Kq = 1.0416 # of iterations = 172
gamma = 1.1601 Kq = 1.0375 # of iterations = 173
gamma = 1.1606 Kq = 1.0334 # of iterations = 174
gamma = 1.1611 Kq = 1.0293 # of iterations = 175
gamma = 1.1616 Kq = 1.0253 # of iterations = 176
gamma = 1.1621 Kq = 1.0213 # of iterations = 177
gamma = 1.1626 Kq = 1.0173 # of iterations = 178
gamma = 1.1631 Kq = 1.0134 # of iterations = 179
gamma = 1.1636 Kq = 1.0096 # of iterations = 180
gamma = 1.1641 Kq = 1.0058 # of iterations = 181
gamma = 1.1646 Kq = 1.002 # of iterations = 182
```

```
disp(['K ref = ',num2str(HinfSF.Kxref)])
```

```
K ref = 16.3895       0.9983     -1.30224    -0.011844   -0.0938166   -0.0434332
```

```
EE=HinfSF.A'*P+P*HinfSF.A
- (P*HinfSF.B_hat+HinfSF.S)*inv(HinfSF.R)*(HinfSF.B_hat'*P+HinfSF.S')
+HinfSF.Q;
disp(['Norm EE opt = ',num2str(norm(EE))]);
```

```
Norm EE opt = 2.0667e-06
```

Hinf Closed Loop

```
Cp = [  Za   0.  Zd  0. ;
          eye(4)];
Dp = 0*Cp*Bp;

% controller
Hinf.Ac = [ Ws.A       0.;
            0.    Wt.A];
Hinf.Bc1 = [ Ws.B*[1 0 0 0 0]
      Wt.B*[1 0 0 0 0]];
Hinf.Bc2 = [-Ws.B
             0.];
Hinf.Cc   = [ HinfSF.Kxref(:,5:6) ];
Hinf.Dc1 = [ 0. HinfSF.Kxref(:,1:4)];
Hinf.Dc2 = [ 0. ];

% closed loop
Zi = inv(eye(1) - Hinf.Dc1*Dp);
Hinf.Acl = [     Ap + Bp*Zi*Hinf.Dc1*Cp                Bp*Zi*Hinf.Cc;
         Hinf.Bc1*(eye(5) + Dp*Zi*Hinf.Dc1)*Cp    Hinf.Ac +
Hinf.Bc1*Dp*Zi*Hinf.Cc];
Hinf.Bcl = [    Bp*Zi*Hinf.Dc2;
       Hinf.Bc2 + Hinf.Bc1*Dp*Zi*Hinf.Dc2];
Hinf.Ccl = [(eye(5) + Dp*Zi*Hinf.Dc1)*Cp Dp*Zi*Hinf.Cc];
Hinf.Dcl = Dp*Zi*Hinf.Dc2;
sys_cl_hinf = ss(Hinf.Acl,Hinf.Bcl,Hinf.Ccl,Hinf.Dcl);
info_hinf = stepinfo(sys_cl_hinf,'RiseTimeThreshold', [0
0.63],'SettlingTimeThreshold', 0.05);
```

```
% Open loop at plant input
Hinf.Alu = [Ap     zeros(4,2);
       Hinf.Bc1*Cp     Hinf.Ac];
Hinf.Blu = [  Bp;
           Hinf.Bc1*Dp];
Hinf.Clu = -[Hinf.Dc1*Cp Hinf.Cc];
Hinf.Dlu = 0;
sys_u_hinf = ss(Hinf.Alu,Hinf.Blu,Hinf.Clu,Hinf.Dlu);

% Loop input break Info
Hinf.lu_margins = allmargin(sys_u_hinf);
Hinf.gm_lu = 20*log10(Hinf.lu_margins.GainMargin);
Hinf.pm_lu = Hinf.lu_margins.PhaseMargin;
Hinf.lgcf_lu = Hinf.lu_margins.PMFrequency;
Hinf.pc_lu = Hinf.lu_margins.GMFrequency;

% Open Loop at plant output
Hinf.Aly = [Ap           Bp*Hinf.Cc;
       zeros(2,4)     Hinf.Ac];
Hinf.Bly = [Bp*Hinf.Dc1;
        Hinf.Bc1];
Hinf.Cly = -[Cp Dp*Hinf.Cc];
Hinf.Dly = 0;
sys_y_hinf = ss(Hinf.Aly,Hinf.Bly,Hinf.Cly,Hinf.Dly);
```

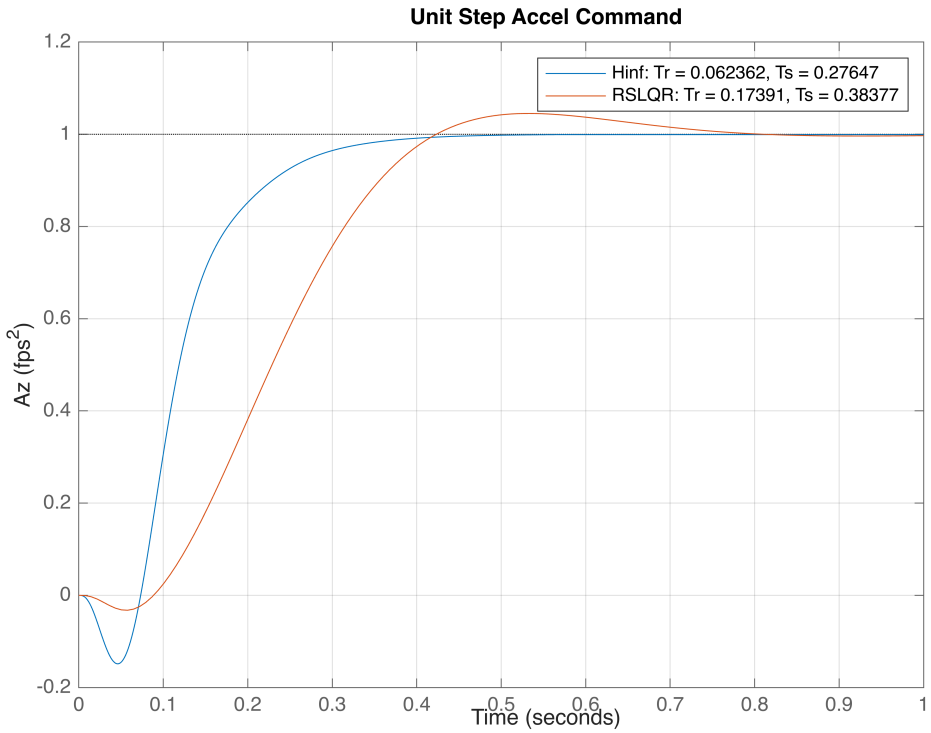
RSLQR Closed Loop (comparison)

```
% gains for target bandwidth (from HW 3)
rslqr.Kx = [0.00999999999999995    -3.12048241475416
-0.301153536676667    0.551627642200742    0.00502690076684165];

rslqr.Ac = 0.;
rslqr.Bc1 = [1. 0. 0. 0. 0.];
rslqr.Bc2 = -1;
rslqr.Cc = -rslqr.Kx(1);
rslqr.Dc1 = [0. -rslqr.Kx(2:4) 0];
rslqr.Dc2 = 0;

Zi = inv(eye(1) - rslqr.Dc1*Dp);
rslqr.Acl = [     Ap + Bp*Zi*rslqr.Dc1*Cp                Bp*Zi*rslqr.Cc;
         rslqr.Bc1*(eye(5) + Dp*Zi*rslqr.Dc1)*Cp    rslqr.Ac +
rslqr.Bc1*Dp*Zi*rslqr.Cc];
rslqr.Bcl = [    Bp*Zi*rslqr.Dc2;
       rslqr.Bc2 + rslqr.Bc1*Dp*Zi*rslqr.Dc2];
rslqr.Ccl = [(eye(5) + Dp*Zi*rslqr.Dc1)*Cp Dp*Zi*rslqr.Cc];
rslqr.Dcl = Dp*Zi*rslqr.Dc2;
sys_cl_rslqr = ss(rslqr.Acl,rslqr.Bcl,rslqr.Ccl,rslqr.Dcl);
info_rslqr = stepinfo(sys_cl_rslqr,'RiseTimeThreshold', [0
0.63],'SettlingTimeThreshold', 0.05);
```

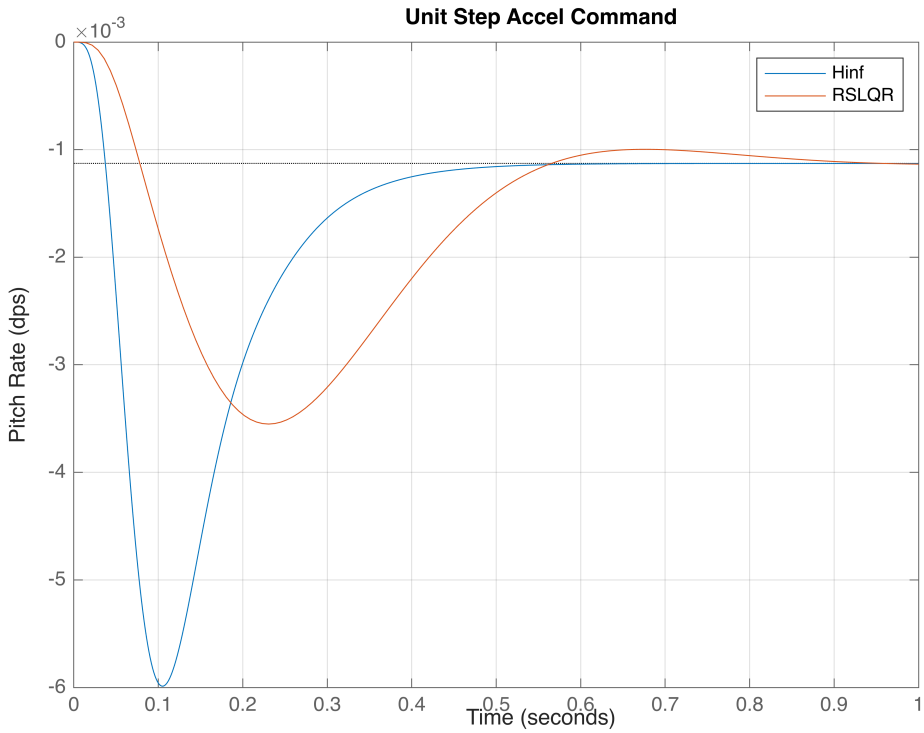
```
% Open loop at plant input
rslqr.Alu = [Ap     zeros(4,1);
       rslqr.Bc1*Cp     rslqr.Ac];
rslqr.Blu = [  Bp;
           rslqr.Bc1*Dp];
rslqr.Clu = -[rslqr.Dc1*Cp rslqr.Cc];
rslqr.Dlu = 0;
sys_u_rslqr = ss(rslqr.Alu,rslqr.Blu,rslqr.Clu,rslqr.Dlu);

% Loop input break Info
rslqr.lu_margins = allmargin(sys_u_rslqr);
rslqr.gm_lu = 20*log10(rslqr.lu_margins.GainMargin);
rslqr.pm_lu = rslqr.lu_margins.PhaseMargin;
rslqr.lgcf_lu = rslqr.lu_margins.PMFrequency;
rslqr.pc_lu = rslqr.lu_margins.GMFrequency;

% Open Loop at plant output
rslqr.Aly = [Ap           Bp*rslqr.Cc;
       zeros(1,4)     rslqr.Ac];
rslqr.Bly = [Bp*rslqr.Dc1;
        rslqr.Bc1];
rslqr.Cly = -[Cp Dp*rslqr.Cc];
rslqr.Dly = 0;
sys_y_rslqr = ss(rslqr.Aly,rslqr.Bly,rslqr.Cly,rslqr.Dly);
```

a) Simulate the closed loop system to a unit step Az command. Plot the RSLQR response with the Hinf response. Compute the rise time and settling time. Label on the plot.

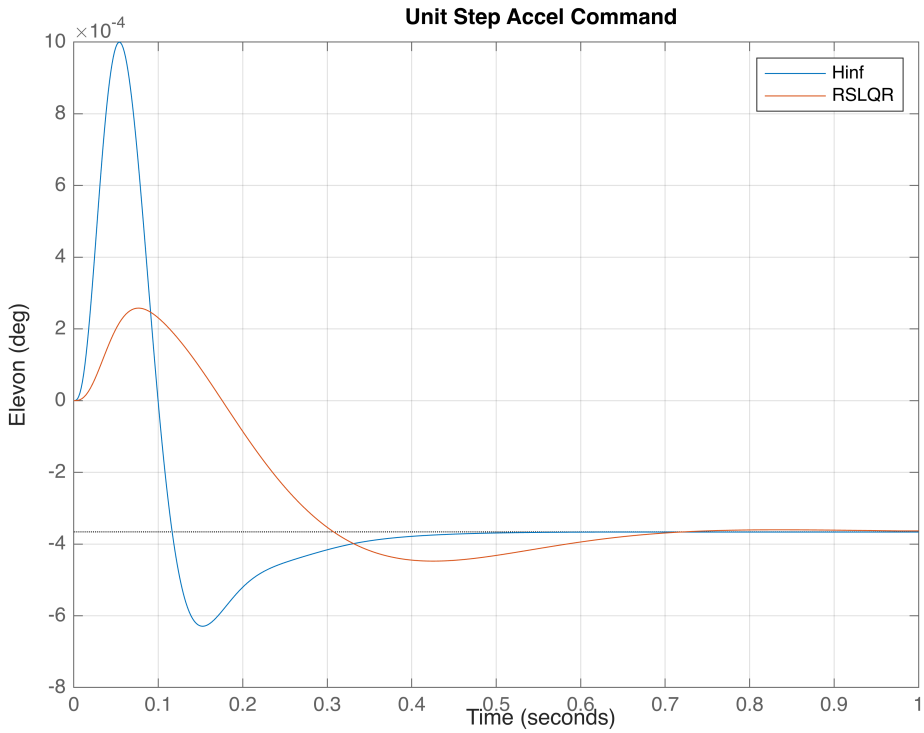
```
figure;
step(sys_cl_hinf(1,1));
hold on;
step(sys_cl_rslqr(1,1))
xlim([0 1]), grid on; legend(['Hinf:
Tr = ',num2str(info_hinf(1).RiseTime),', Ts
= ',num2str(info_hinf(1).SettlingTime)],['RSLQR: Tr
= ',num2str(info_rslqr(1).RiseTime),', Ts =
',num2str(info_rslqr(1).SettlingTime)])
ylabel('Az (fps^2)'), title('Unit Step Accel Command')
```



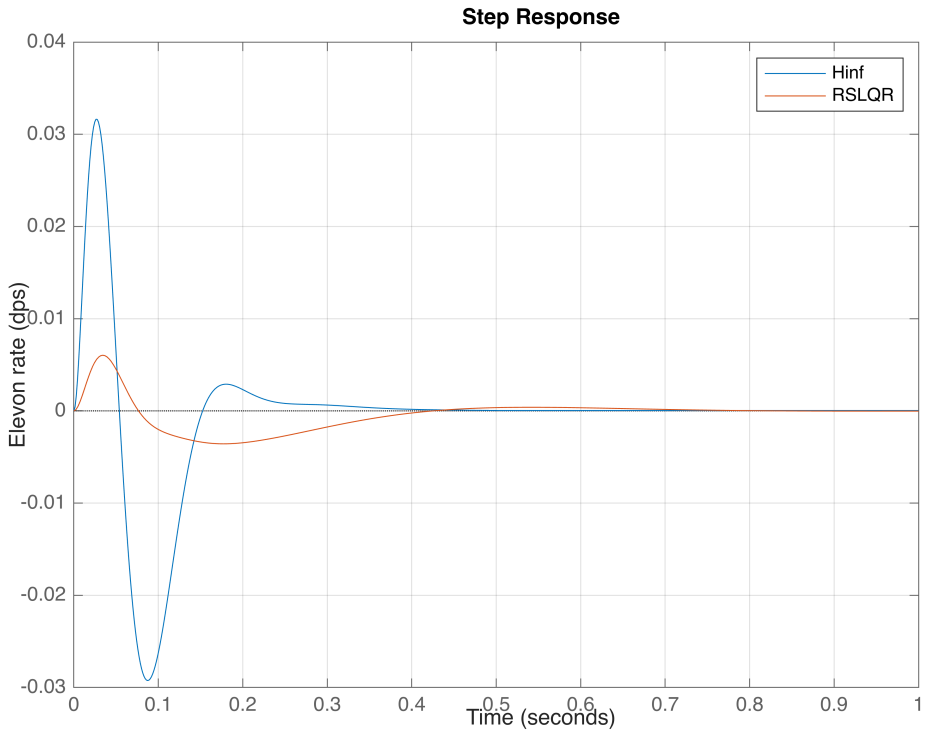

```
figure;
step(sys_cl_hinf(3,1));
hold on;
step(sys_cl_rslqr(3,1))
xlim([0 1]), grid on; legend('Hinf','RSLQR')
ylabel('Pitch Rate (dps)'), title('Unit Step Accel Command')
```




```
figure;
step(sys_cl_hinf(4,1));
hold on;
step(sys_cl_rslqr(4,1))
xlim([0 1]), grid on; legend('Hinf','RSLQR')
ylabel('Elevon (deg)'),title('Unit Step Accel Command')
```

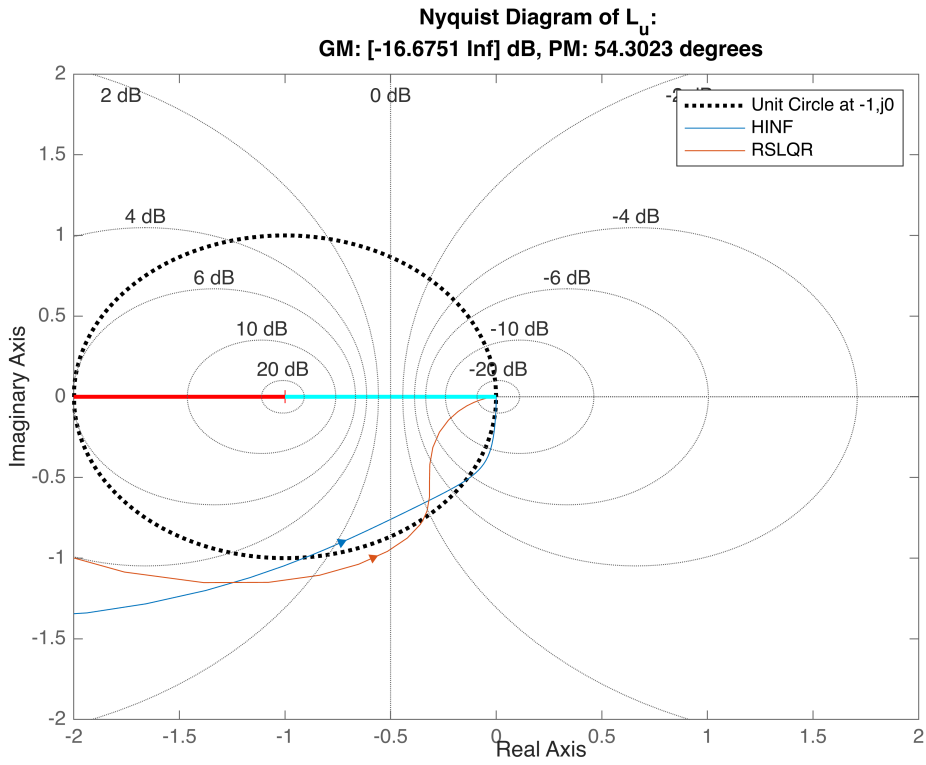



```
figure;
title('Unit Step Accel Command')
step(sys_cl_hinf(5,1));
hold on;
step(sys_cl_rslqr(5,1))
xlim([0 1]), grid on; legend('Hinf','RSLQR')
ylabel('Elevon rate (dps)')
```



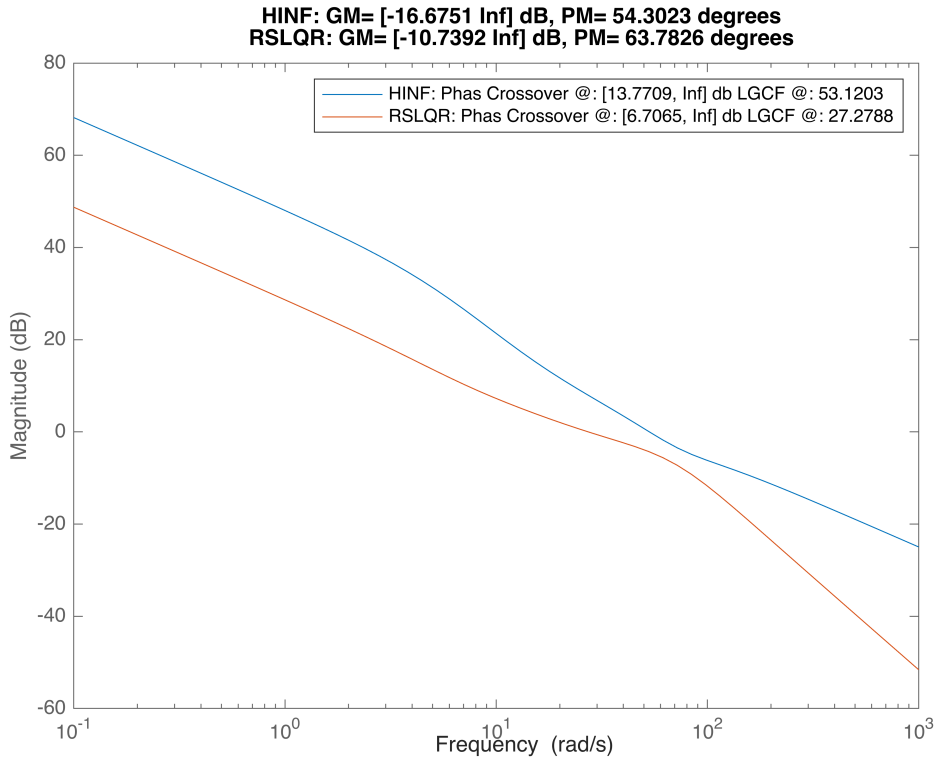
b) Plot a Nyquist plot and identify on the plot the gain and phase margins, loop gain and phase crossover frequencies. Plot the RSLQR Nyquist with the Hinf Nyquist.

```
figure;
plot(xx1,yy1,'k:','LineWidth',2)
hold on;
n = nyquistplot(sys_u_hinf);
n = nyquistplot(sys_u_rslqr);
plot([-1/Hinf.lu_margins.GainMargin(1) -1.],[0. 0.],'r',[-1 0],[0.
0.],'c','LineWidth',2);grid
xlim([-2,2]), ylim([-2,2])
setoptions(n,'ShowfullContour','off'), grid on
title(['Nyquist Diagram of L_u: ',...
    newline 'GM: [', num2str(Hinf.gm_lu(1)),' Inf] dB, ' 'PM: '
num2str(Hinf.pm_lu), ' degrees'])
legend('Unit Circle at -1,j0','HINF','RSLQR')
hold off;
```



c) Plot a Bode plot and identify on the plot the gain and phase margins, loop gain and phase crossover frequencies. Plot the RSLQR Bode with the Hinf Bode.

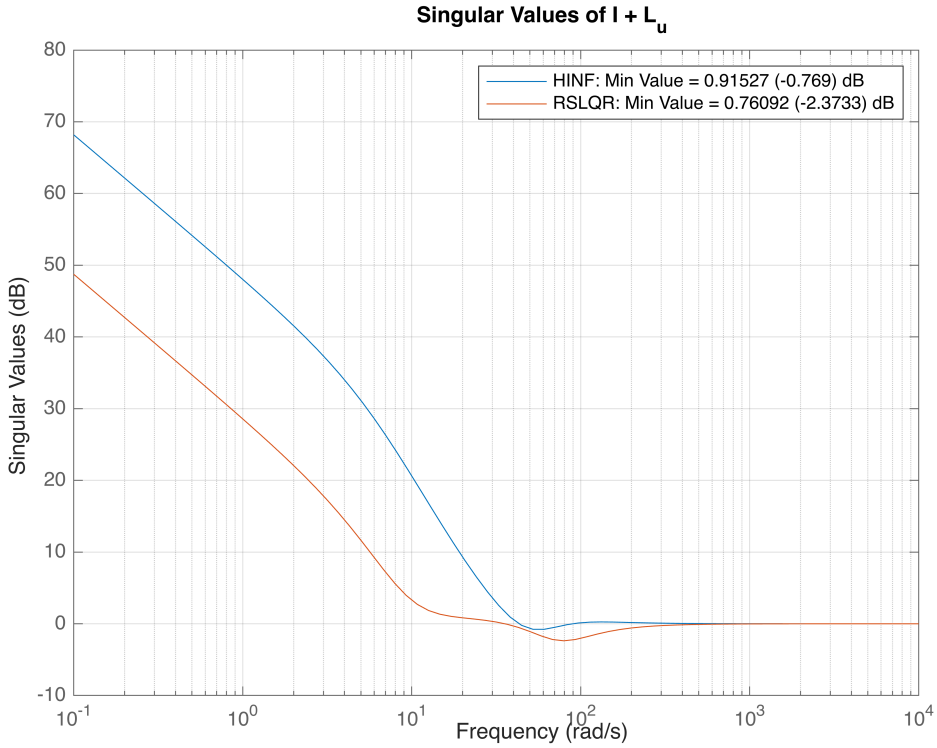
```
figure;
bodemag(sys_u_hinf);
hold on;
bodemag(sys_u_rslqr)
legend(['HINF: Phas Crossover @: [', num2str(Hinf.pc_lu(1)),', Inf] db
','LGCF @: ',num2str(Hinf.lgcf_lu(1))],...
    ['RSLQR: Phas Crossover @: [', num2str(rslqr.pc_lu(1)),', Inf] db
','LGCF @: ',num2str(rslqr.lgcf_lu(1))])
title(['HINF: GM= [', num2str(Hinf.gm_lu(1)), ' Inf] dB, ', 'PM=
',num2str(Hinf.pm_lu(1)), ' degrees ',...
    newline 'RSLQR: GM= [', num2str(rslqr.gm_lu(1)), ' Inf] dB, ', 'PM=
',num2str(rslqr.pm_lu(1)), ' degrees '])
hold off;
```



d) Plot the minimum singular value of the return difference matrix in dB vs frequency. Identify on the plot the minimum value of the return difference matrix (not in dB). Plot the RSLQR $\underline{\sigma}(I + L_u)$ including the actuator in the plant model with the Hinf $\underline{\sigma}(I + L_u)$ (both plant models should have the actuator in them).

```
[a,b] = ss2tf(Hinf.Alu,Hinf.Blu,Hinf.Clu,Hinf.Dlu);
c = b + a;
sys_IL_hinf = tf(c,b);
alpha_0_hinf = min(sigma(sys_IL_hinf));

sys_ILi_hinf = tf(c,a);
beta_0_hinf = (min(sigma(sys_ILi_hinf)));

[a,b] = ss2tf(rslqr.Alu,rslqr.Blu,rslqr.Clu,rslqr.Dlu);
c = b + a;
sys_IL_rslqr = tf([c 0],[b 0]);
alpha_0_rslqr = min(sigma(sys_IL_rslqr));

sys_ILi_rslqr = tf(c,a);
beta_0_rslqr = (min(sigma(sys_ILi_rslqr)));

figure;
sigma(sys_IL_hinf);
```

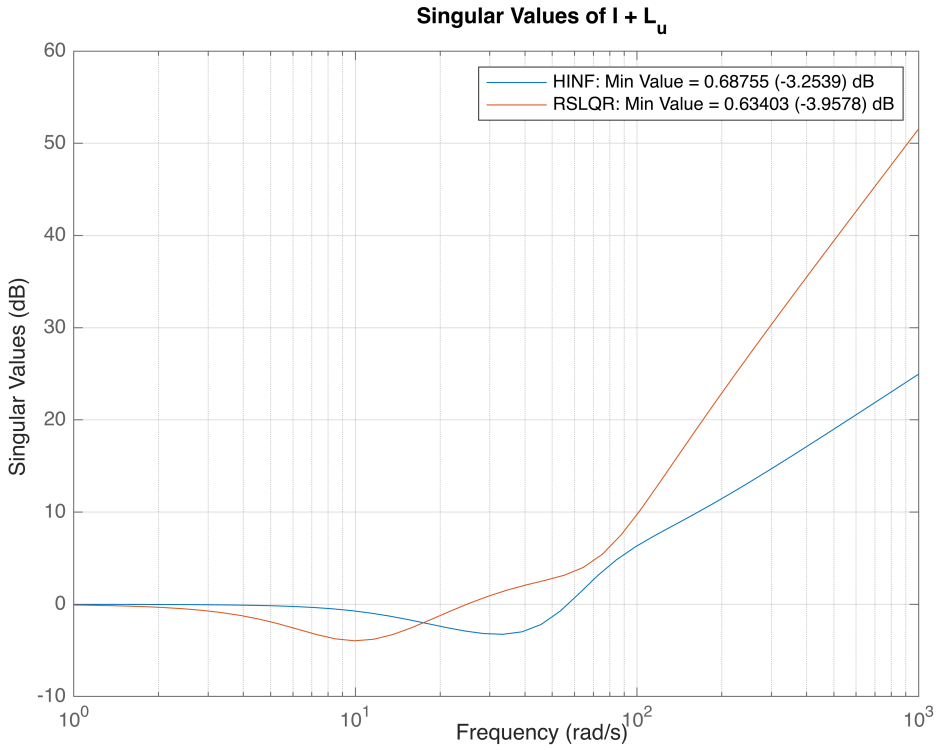
```
hold on;
sigma(sys_IL_rslqr);
legend(['HINF: Min Value = ',num2str(alpha_0_hinf), ' (',
num2str(20*log10(alpha_0_hinf)),') dB'],...
       ['RSLQR: Min Value = ',num2str(alpha_0_rslqr), ' (',
num2str(20*log10(alpha_0_rslqr)),') dB'])
grid on, title('Singular Values of I + L_u')
hold off;
```



e) Plot the minimum singular value of the stability robustness matrix in dB vs frequency. Identify on the plot the minimum value of the stability robustness matrix (not in dB). Plot the RSLQR $\underline{\sigma}\left(I + L_u^{-1}\right)$ including the actuator in the plant model with the Hinf $\underline{\sigma}\left(I + \bar{L}_u^{-1}\right)$ (both plant models should have the actuator in them).

```
figure;
sigma(sys_ILi_hinf);
hold on
sigma(sys_ILi_rslqr);
legend(['HINF: Min Value = ',num2str(beta_0_hinf), ' (',
num2str(20*log10(beta_0_hinf)),') dB'],...
       ['RSLQR: Min Value = ',num2str(beta_0_rslqr), ' (',
num2str(20*log10(beta_0_rslqr)),') dB'])
grid on, title('Singular Values of I + L_u')
```

```
hold off;
```



f) Compute the singular value gain and phase margins for the system (at the plant input) and compare them with the RSLQR margins (both plant models should have the actuator in them).

```
pm = 2*asin(alpha_0_hinf/2)*180/pi;
GM_l = 20*log10(1/(1 + alpha_0_hinf));
GM_u = 20*log10(1/(1 - alpha_0_hinf));
Gm_IL_hinf = [GM_l GM_u];
pm_IL_hinf = [-pm pm];

pm = 2*asin(alpha_0_rslqr/2)*180/pi;
GM_l = 20*log10(1/(1 + alpha_0_rslqr));
GM_u = 20*log10(1/(1 - alpha_0_rslqr));
Gm_IL_rslqr = [GM_l GM_u];
pm_IL_rslqr = [-pm pm];

text = ['HINF Gain Margin from (I+L_u):  [',num2str(Gm_IL_hinf(1)),',
',num2str(Gm_IL_hinf(2)),'] dB',...
        newline 'HINF Phase Margin from (I+L_u):
[',num2str(pm_IL_hinf(1)),', ',num2str(pm_IL_hinf(2)),'] degrees',...
        newline 'RSLQR Gain Margin from (I+L_u):
[',num2str(Gm_IL_rslqr(1)),', ',num2str(Gm_IL_rslqr(2)),'] dB',...
        newline 'RSLQR Phase Margin from (I+L_u):
[',num2str(pm_IL_rslqr(1)),', ',num2str(pm_IL_rslqr(2)),'] degrees'];
```

```
disp(text);
```

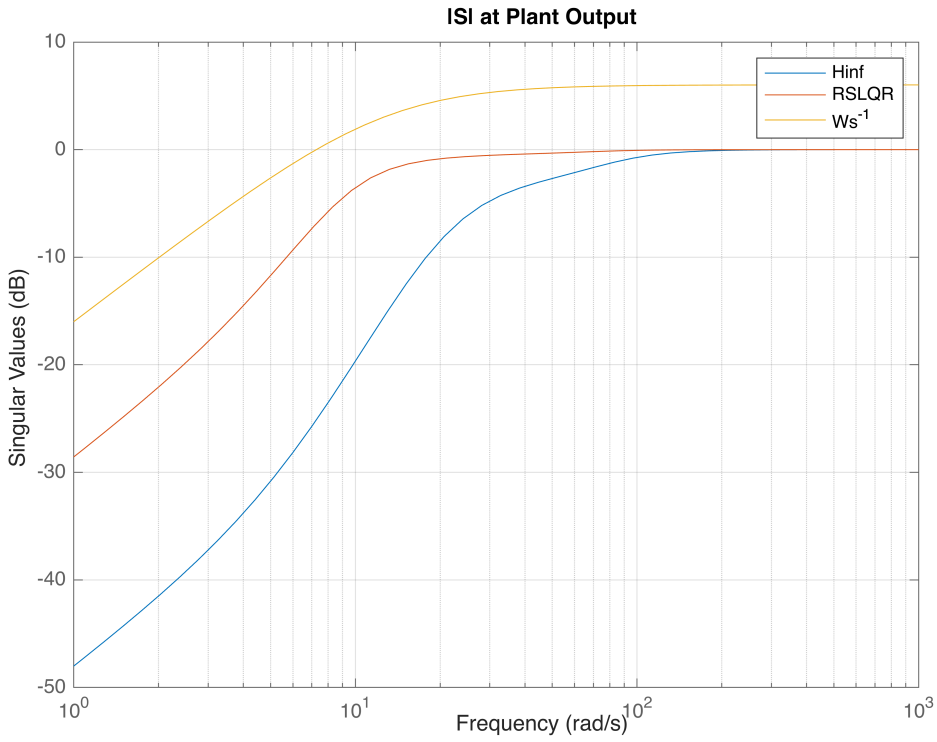
```
HINF Gain Margin from (I+L_u):  [-5.6446, 21.4394] dB
HINF Phase Margin from (I+L_u): [-54.4693, 54.4693] degrees
RSLQR Gain Margin from (I+L_u):  [-4.9148, 12.429] dB
RSLQR Phase Margin from (I+L_u): [-44.7242, 44.7242] degrees
```

```
GM_u = 20*log10(1 + beta_0_hinf);
GM_l = 20*log10(1 - beta_0_hinf);
pm = 2*asin(beta_0_hinf/2)*180/pi;
Gm_ILi_hinf = [GM_l GM_u];
pm_ILi_hinf = [-pm pm];

GM_u = 20*log10(1 + beta_0_rslqr);
GM_l = 20*log10(1 - beta_0_rslqr);
pm = 2*asin(beta_0_rslqr/2)*180/pi;
Gm_ILi_rslqr = [GM_l GM_u];
pm_ILi_rslqr = [-pm pm];

text = ['HINF Gain Margin from inv(I+L_u):  [',num2str(Gm_ILi_hinf(1)),',
',num2str(Gm_ILi_hinf(2)),'] dB',...
        newline 'HINF Phase Margin from inv(I+L_u):
[',num2str(pm_ILi_hinf(1)),', ',num2str(pm_ILi_hinf(2)),'] degrees',...
        newline 'RSLQR Gain Margin from inv(I+L_u):
[',num2str(Gm_ILi_rslqr(1)),', ',num2str(Gm_ILi_rslqr(2)),'] dB',...
        newline 'RSLQR Phase Margin from inv(I+L_u):
[',num2str(pm_ILi_rslqr(1)),', ',num2str(pm_ILi_rslqr(2)),'] degrees'];
disp(text);
```

```
HINF Gain Margin from inv(I+L_u):  [-10.1044, 4.5451] dB
HINF Phase Margin from inv(I+L_u): [-40.2141, 40.2141] degrees
RSLQR Gain Margin from inv(I+L_u):  [-8.731, 4.2652] dB
RSLQR Phase Margin from inv(I+L_u): [-36.9648, 36.9648] degrees
```

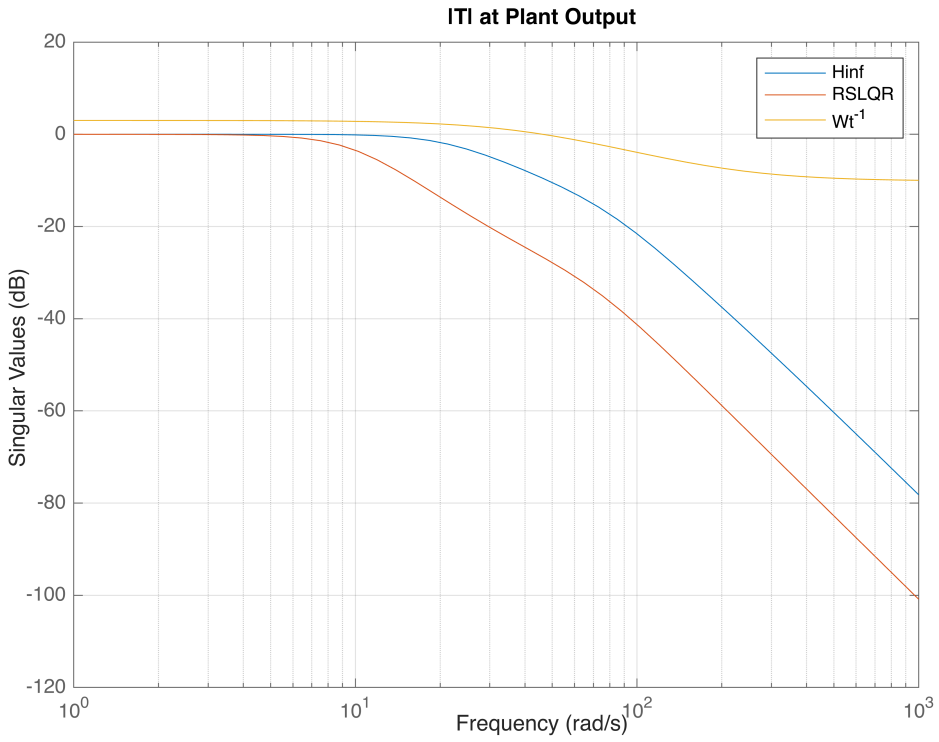
g) Compute the sensitivity function $e/r = S$ (at the output).). Plot the RSLQR $e/r = S$ including the actuator in the plant model with the Hinf $e/r = S$ (both plant models should have the actuator in them). Plot your Weighting filter WS with these two frequency responses.

```
S_y_hinf = 1/(1+sys_y_hinf(1,1));
S_y_rslqr = 1/(1+sys_y_rslqr(1,1));

sigma(S_y_hinf), hold on;
sigma(S_y_rslqr),
sigma(inv(Ws.sys))
grid on, title('|S| at Plant Output')
legend('Hinf', 'RSLQR', 'Ws^{-1}')
hold off
```



h) Compute the complementary sensitivity $y/r = T$ (at the output). Plot the RSLQR $A_z/A_{zc} = T$ including the actuator in the plant model with the Hinf $A_z/A_{zc} = T$ (both plant models should have the actuator in them). Plot your Weighting filter WT with these two frequency responses.

```
T_y_hinf = sys_y_hinf(1,1)/(1+sys_y_hinf(1,1));
T_y_rslqr = sys_y_rslqr(1,1)/(1+sys_y_rslqr(1,1));

sigma(T_y_hinf), hold on;
sigma(T_y_rslqr),
sigma(inv(Wt.sys))
legend('Hinf', 'RSLQR', 'Wt^{-1}')
grid on, title('|T| at Plant Output')
xlim([1 1000]), hold off
```


ITI at Plant Output
Hinf
RSLQR
Wt-1
Singular Values (dB)
Frequency (rad/s)
20
0
-20
-40
-60
-80
-100
-120
100
101
102
103