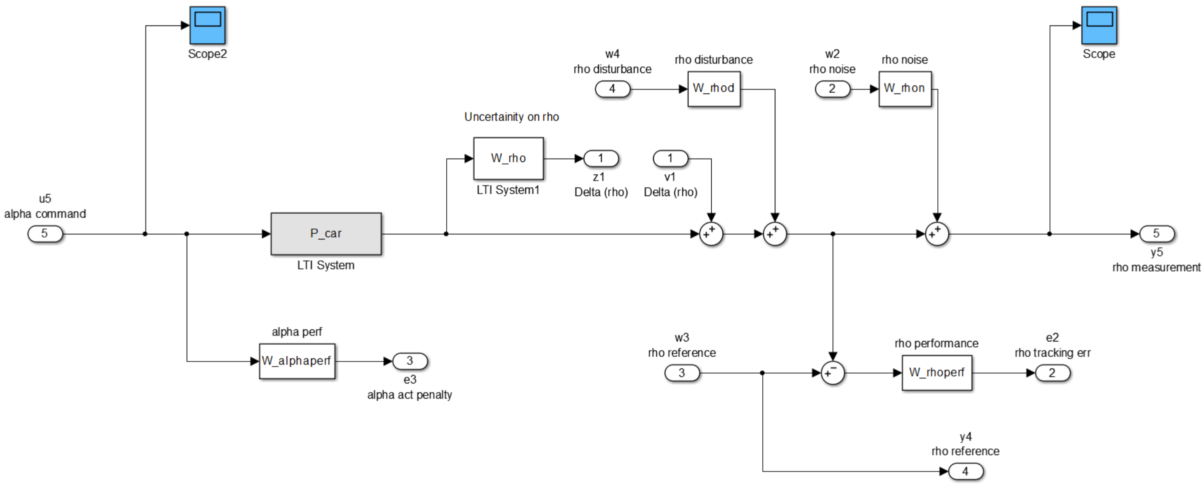
ATIC Exercise 11

Some more findings about the crayz arab drivers :)

a)

i) model_car:




```
[A_P,B_P,C_P,D_P] = linmod('model_car');
P = ss(A_P,B_P,C_P,D_P);
```

```
P =

  a =
              x1         x2         x3         x4         x5         x6         x7         x8
   x1          0          0          1          0          0          0          0          0
   x2          0          0          0          1          0          0          0          0
   x3      78.23     -59.38          0          0          0          0          0          0
   x4          0       -400          0        -20          0          0          0          0
   x5      40.74          0          0          0         -4          0          0          0
   x6     -2.546          0          0          0          0  -0.001667          0      -0.02
   x7          0          0          0          0          0          0      -1000          0
   x8          0          0          0          0          0          0          0     -0.002

  b =
            u1      u2      u3      u4      u5
   x1        0       0       0       0       0
   x2        0       0       0       0       0
   x3        0       0       0       0       0
   x4        0       0       0       0   69.81
   x5        0       0       0       0       0
   x6    -0.25       0    0.25       0       0
   x7        0       0       0       0     128
   x8        0       0       0  0.0625       0

  c =
            x1      x2      x3      x4      x5      x6      x7      x8
   y1    20.37       0       0       0      -2       0       0       0
   y2        0       0       0       0       0  0.1333       0       0
   y3        0       0       0       0       0       0    -125       0
   y4        0       0       0       0       0       0       0       0
   y5    10.19       0       0       0       0       0       0    0.08

  d =
         u1    u2    u3    u4    u5
   y1     0     0     0     0     0
   y2     0     0     0     0     0
   y3     0     0     0     0    16
   y4     0     0     1     0     0
   y5     1  0.01     0     0     0
```

ii)

```
%                             ________
%                             |       |
%      Delta inputs:  z  <---|       |<--- v:  Delta output
%      cost/errors:   e  <---|   P   |<--- w:  exogenous inputs
%      measurements:  y  <---|       |<--- u:  control actuation
%                             |_______|
%
%                       Outputs     Inputs
%
%    Delta1 (rho)         z1          v1
%
%    rho tracking err     e2          w2  rho noise
%    alpha act penalty    e3          w3  rho reference
%                                     w4  rho disturbance
%
%    rho reference        y4          u5  alpha command
%    rho measurement      y5

Iz = [1:1]';
Ie = [2:3]';
Iy = [4:5]';

Iv = [1:1]';
Iw = [2:4]';
Iu = [5:5]';

A   = A_P;
Bw  = B_P(:,Iw);
Bu  = B_P(:,Iu);
Ce  = C_P(Ie,:);
Cy  = C_P(Iy,:);
Dew = D_P(Ie,Iw);
Dyw = D_P(Iy,Iw);
Deu = D_P(Ie,Iu);
Dyu = D_P(Iy,Iu);

P_test  = ss(A,[Bw,Bu], [Ce;Cy],[Dew, Deu;Dyw,Dyu]);

P_test =

  a = 
            x1        x2        x3        x4        x5        x6        x7        x8
   x1        0         0         1         0         0         0         0         0
   x2        0         0         0         1         0         0         0         0
   x3    78.23    -59.38         0         0         0         0         0         0
   x4        0      -400         0       -20         0         0         0         0
   x5    40.74         0         0         0        -4         0         0         0
   x6   -2.546         0         0         0         0 -0.001667         0     -0.02
   x7        0         0         0         0         0         0     -1000         0
   x8        0         0         0         0         0         0         0    -0.002

  b = 
          u1      u2      u3      u4
   x1      0       0       0       0
   x2      0       0       0       0
   x3      0       0       0       0
   x4      0       0       0   69.81
   x5      0       0       0       0
   x6      0    0.25       0       0
   x7      0       0       0     128
   x8      0       0  0.0625       0

  c = 
          x1      x2      x3      x4      x5      x6      x7      x8
   y1      0       0       0       0       0  0.1333       0       0
   y2      0       0       0       0       0       0    -125       0
   y3      0       0       0       0       0       0       0       0
   y4  10.19       0       0       0       0       0       0    0.08

  d = 
        u1    u2    u3    u4
   y1    0     0     0     0
   y2    0     0     0    16
   y3    0     1     0     0
   y4 0.01     0     0     0
```

iii)

```
cvx_begin sdp
    cvx_solver sedumi

    variable X(n,n) symmetric;
    variable Y(n,n) symmetric;
    variable W(length(Ie),length(Ie)) symmetric;
    variable Ah(n,n);
    variable Bh(n,length(Iy));
    variable Ch(length(Iu),n);
    variable gamma_2lmi

    minimize gamma_2lmi;
    subject to
        trace(W) < gamma_2lmi;

        0.5*(...
        [W, Ce*X + Deu*Ch, Ce;
            X*Ce'+Ch'*Deu', X, eye(n,n);
            Ce', eye(n,n),Y]+...
        [W, Ce*X + Deu*Ch, Ce;
            X*Ce'+Ch'*Deu', X, eye(n,n);
            Ce', eye(n,n),Y]') > 0;

        0.5*(...
        [A*X+Bu*Ch+X*A'+Ch'*Bu', A+Ah', Bw;
            A'+Ah, Y*A'+A'*Y+Bh*Cy+Cy'*Bh', Y*Bw+Bh*Dyw;
            Bw',Bw'*Y+Dyw'*Bh', -eye(length(Iw),length(Iw))]+...
        [A*X+Bu*Ch+X*A'+Ch'*Bu', A+Ah', Bw;
            A'+Ah, Y*A'+A'*Y+Bh*Cy+Cy'*Bh', Y*Bw+Bh*Dyw;
            Bw',Bw'*Y+Dyw'*Bh', -eye(length(Iw),length(Iw))]') < 0;

cvx_end
```

iv)

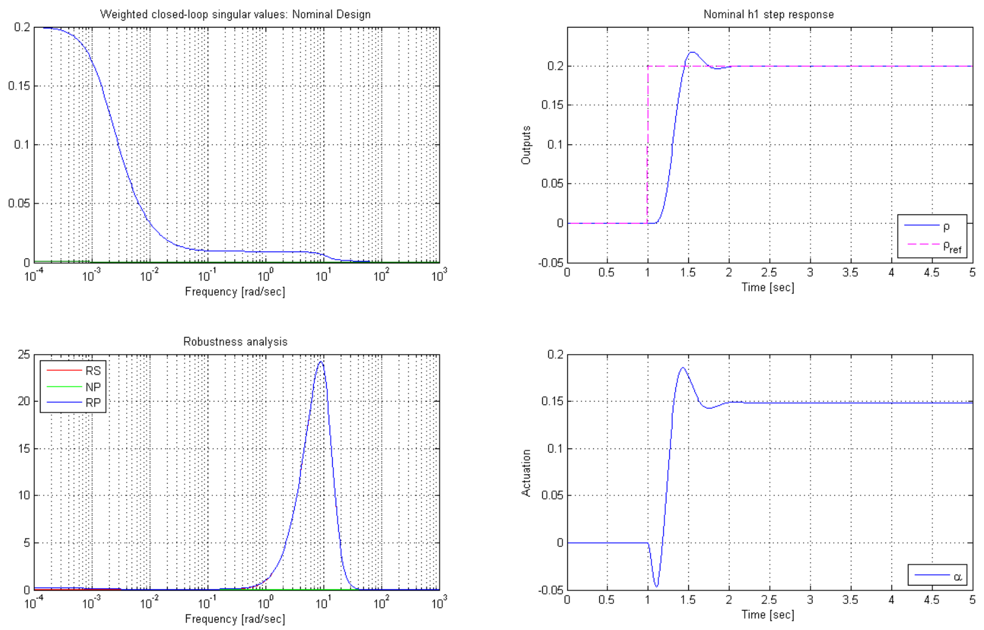
With the LMI, I don't get a stable closed loop. There is one non-negative pole in the CL system. See plots at the back of the document for further comparison. Therefore, the H2 norm of the system is inf.

LMI	**H2Syn**
open_loop_poles =	open_loop_poles =
1.0e+03 *	1.0e+03 *
-0.0000 + 0.0000i	-0.0000 + 0.0000i
-0.0100 + 0.0173i	-0.0100 + 0.0173i
-0.0100 - 0.0173i	-0.0100 - 0.0173i
0.0088 + 0.0000i	0.0088 + 0.0000i
-0.0088 + 0.0000i	-0.0088 + 0.0000i
-1.0000 + 0.0000i	-1.0000 + 0.0000i
-0.0000 + 0.0000i	-0.0000 + 0.0000i
closed_loop_poles =	closed_loop_poles =
1.0e+03 *	1.0e+03 *
-0.0000 + 0.0000i	-0.0000 + 0.0000i
-1.0000 + 0.0000i	-1.0000 + 0.0000i
-7.3444 + 0.0000i	-0.0057 + 0.0100i
-0.0042 + 0.0557i	-0.0057 - 0.0100i
-0.0042 - 0.0557i	-0.0005 + 0.0000i
-0.0101 + 0.0172i	-0.0125 + 0.0034i
-0.0101 - 0.0172i	-0.0125 - 0.0034i
-0.0176 + 0.0000i	-0.0088 + 0.0000i
-0.0014 + 0.0108i	-0.0088 + 0.0000i
-0.0014 - 0.0108i	-0.0100 + 0.0172i
-0.0103 + 0.0000i	-0.0100 - 0.0172i
-0.0044 + 0.0000i	-0.0100 + 0.0173i
-0.0040 + 0.0000i	-0.0100 - 0.0173i
0.0000 + 0.0000i	-0.0000 + 0.0000i
-0.0000 + 0.0000i	-0.0040 + 0.0000i
closed_loop_norm =	closed_loop_norm =
Inf	0.0188

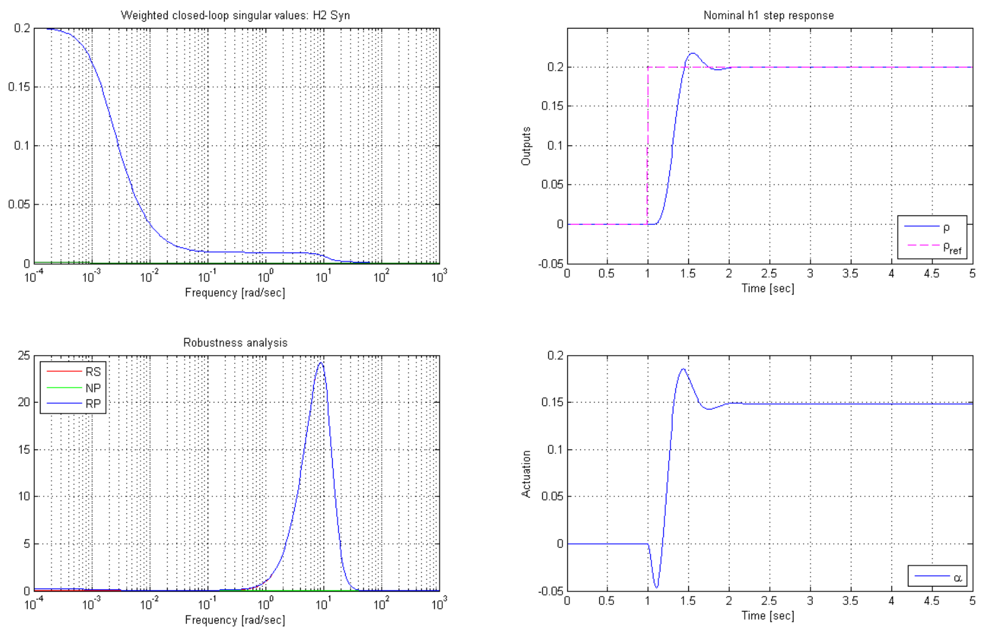
b)

Unfortunately, the LMI has a non-negative pole and is therefore not stable. The plots show step responses and corresponding steering angle alpha for different controllers. For the LMI controller, we see a lot of oscillations which expresses this instability.

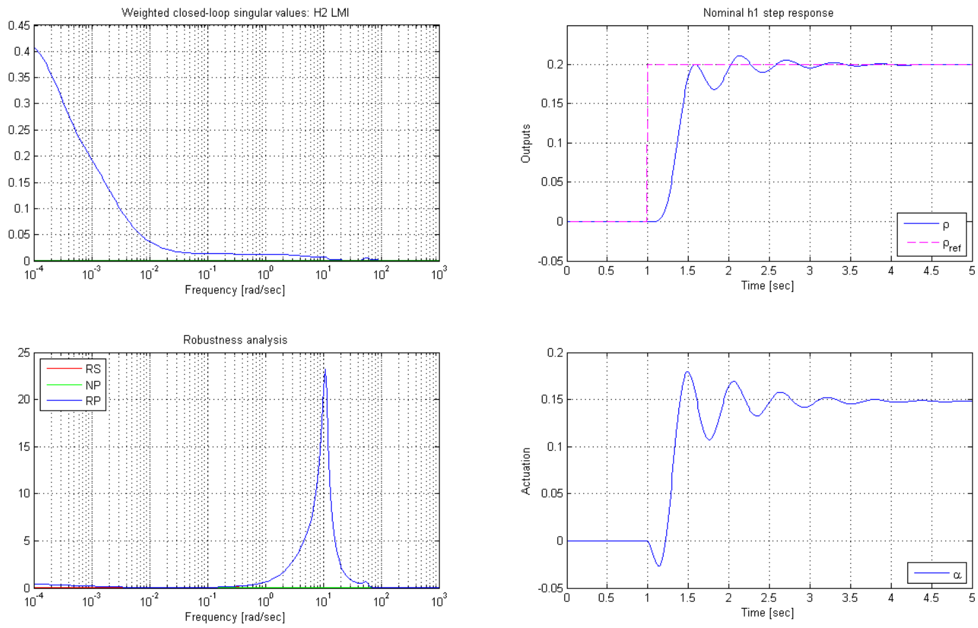
Nominal Design



H2syn



LMI



c)

The inf-norm of the nominal design is slightly smaller. This makes sense, since it optimizes for a minimum inf-gain. The same holds for the 2-norm, where the nominal design results in a bigger norm.

```
h2_nom - h2_syn = 9.6015e-07
h2_nom_inf - h2_syn_inf = -2.0374e-04
```

<table>
<tr>
<td><pre>
Nominal design

open_loop_poles =

   1.0e+03 *

  -0.0000 + 0.0000i
  -0.0100 + 0.0173i
  -0.0100 - 0.0173i
   0.0088 + 0.0000i
  -0.0088 + 0.0000i
  -1.0000 + 0.0000i
  -0.0000 + 0.0000i

closed_loop_poles =

   1.0e+03 *

  -0.0000 + 0.0000i
  -1.0000 + 0.0000i
  -0.0057 + 0.0100i
  -0.0057 - 0.0100i
  -0.0005 + 0.0000i
  -0.0125 + 0.0034i
  -0.0125 - 0.0034i
  -0.0088 + 0.0000i
  -0.0088 + 0.0000i
  -0.0100 + 0.0172i
  -0.0100 - 0.0172i
  -0.0100 + 0.0173i
  -0.0100 - 0.0173i
  -0.0000 + 0.0000i

closed_loop_norm =

    0.0188
</pre></td>
<td><pre>
H2Syn

open_loop_poles =

   1.0e+03 *

  -0.0000 + 0.0000i
  -0.0100 + 0.0173i
  -0.0100 - 0.0173i
   0.0088 + 0.0000i
  -0.0088 + 0.0000i
  -1.0000 + 0.0000i
  -0.0000 + 0.0000i

closed_loop_poles =

   1.0e+03 *

  -0.0000 + 0.0000i
  -1.0000 + 0.0000i
  -0.0057 + 0.0100i
  -0.0057 - 0.0100i
  -0.0005 + 0.0000i
  -0.0125 + 0.0034i
  -0.0125 - 0.0034i
  -0.0088 + 0.0000i
  -0.0088 + 0.0000i
  -0.0100 + 0.0172i
  -0.0100 - 0.0172i
  -0.0100 + 0.0173i
  -0.0100 - 0.0173i
  -0.0000 + 0.0000i
  -0.0040 + 0.0000i

closed_loop_norm =

    0.0188
</pre></td>
</tr>
</table>