Control and operation of tokamaks
Exercise 3 - Vertical field design and position control
Solutions

Lecturer: F. Felici
Instructors: P. M. Cabrera, R. Coosemans, S. Dubbioso, A. Mele, P. Molina, F. Pastore, J. Svantner
EPFL - SPC

February 2025

Ex 3.1: Vertical field design

a)

Calculate the vertical field required to balance the radial forces on a TCV plasma of $I_p = 400kA$ *with* $l_i = 1$, $\beta_p = 0.5$, $\kappa = 1.6$, $a = 0.25m$, $R = 0.88m$.

To compute the vertical field required to balance the radial forces we start with the plasma force balance equation (see equation (5) slide 11 in Magnetic modeling and control of tokamaks, part III: Plasma position stability and control):

$$m_p \frac{\mathrm{d}^2 R}{\mathrm{d}t^2} = F_{R,\mathrm{loop}} + F_{R,\mathrm{tyre}} + F_{R,\mathrm{Lorentz}} \tag{1}$$

$$= \frac{\mu_0 I_p^2}{2}\left(\log\frac{8R}{a\sqrt{\kappa}} + \beta_p + \frac{l_i}{2} - \frac{3}{2}\right) + 2\pi R I_p B_z\,. \tag{2}$$

If the radial forces are balanced then $m_p \frac{\mathrm{d}^2 R}{\mathrm{d}t^2} = 0$ and we can solve (2) for B_z:

$$B_z = -\frac{\mu_0 I_p}{4\pi R}\left(\log\frac{8R}{a\sqrt{\kappa}} + \beta_p + \frac{l_i}{2} - \frac{3}{2}\right)\,.$$

If we now replace $I_p = 400$kA with $l_i = 1$, $\beta_p = 0.5$, $\kappa = 1.6$, $a = 0.25m$, $R = 0.88$m we get:

$$B_z = -\frac{4\pi \times 10^{-7} \times 4 \times 10^5}{4\pi .88}\left(\log\frac{8 \times 0.88}{0.25\sqrt{1.6}} + \beta_p + 0.5 - \frac{3}{2}\right) \approx -0.1183 \text{ T} . \quad (3)$$

b)

Find a combination of E and F coils in TCV that gives the required vertical field. Hint: Formulate the problem as a least squares problem trying to get the correct B_z over a large portion of the x grid.

$$\min_{\boldsymbol{I}_a} \|B_z(\boldsymbol{I}_a) - B_{z,required}\|^2 + \|B_r(\boldsymbol{I}_a)\|^2 ,$$

where B_z and B_r are the fields generated by $\boldsymbol{I}_a$ on the xgrid, given by `Bz=G.Bzxa*Ia` *and* `Br=G.Brxa*Ia`*.*

Each E/F coil produces a magnetic field due to the current, $I_{a,i}$, flowing through it. The larger this current the larger the magnetic field. If we set only the current of the coil i to 1, $I_{a,i} = 1A$, and all the other currents to 0, $I_{a,k} = 0$ for $k \neq i$, we can see the magnetic field produced by each coil given a unit current. In Figure 1, Figure 2 and Figure 3 we can see for each coil and a unit current the $B_{r,i}$, $B_{z,i}$ components and $\vec{B_i}$, respectively.

The total magnetic field, $\vec{B}$, is obtained by a linear combination of the magnetic field generated by each coil:

$$\vec{B} = \sum_i \vec{B_i} I_{a,i} \quad \Rightarrow \quad B_r = \sum_i B_{r,i} I_{a,i} \quad \text{and} \quad B_z = \sum_i B_{z,i} I_{a,i} . \quad (4)$$

Where $\vec{B_i} = B_{r,i}\vec{e}_r + B_{z,i}\vec{e}_z$ and $B_{r,i}$ and $B_{z,i}$ and the r and z components of the unit magnetic field generated by each coil. This can be written more compactly as matrix multiplications:

$$B_r(r_k, z_k) = (\mathsf{B}_r\, \boldsymbol{I}_a)_k \quad \text{and} \quad B_z(r_k, z_k) = (\mathsf{B}_z\, \boldsymbol{I}_a)_k ,$$

where $\mathsf{B}_r =$ `G.Brxa` and $\mathsf{B}_z =$ `G.Bzxa`.

The idea with the Least-Squares formulation is to find the combination of currents $I_{a,i}$ such that the total magnetic field produced has the minimum difference to a target magnetic field at a set of M points (r_k, z_k) with $k = 1, \ldots, M$. Therefore the Least-Squares problem is:

Find $\boldsymbol{I}_a = [I_{a,1}, \ldots I_{a,N}]$, such that:

$$\min_{I_a} \left[(\mathsf{B}_r\, \boldsymbol{I}_a - \boldsymbol{B}_{r,\text{required}})^2 + (\mathsf{B}_z\, \boldsymbol{I}_a - \boldsymbol{B}_{z,\text{required}})^2\right] .$$

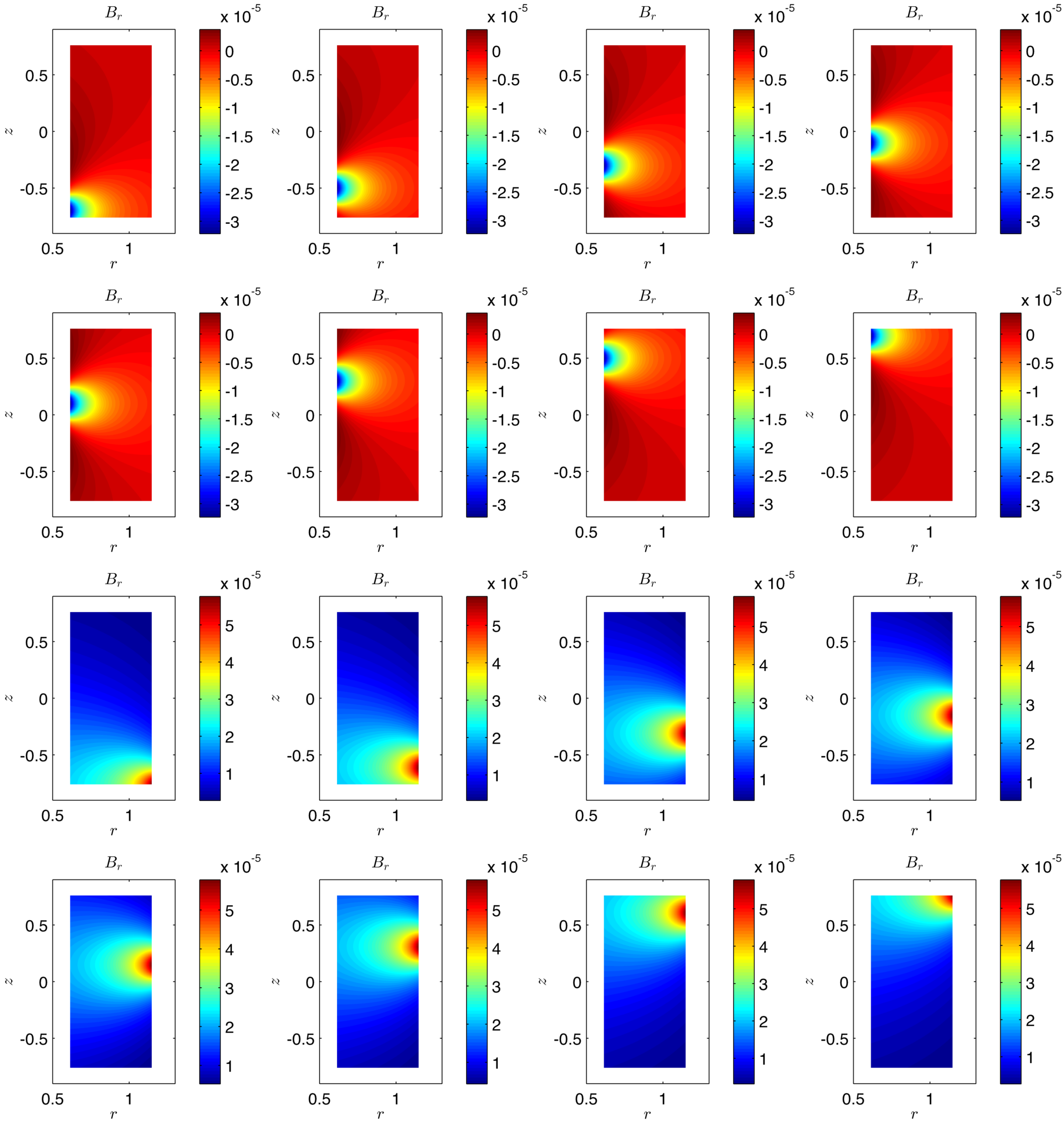


Figure 1: B_r component of the magnetic field produced by each E/F coil for a unit current.

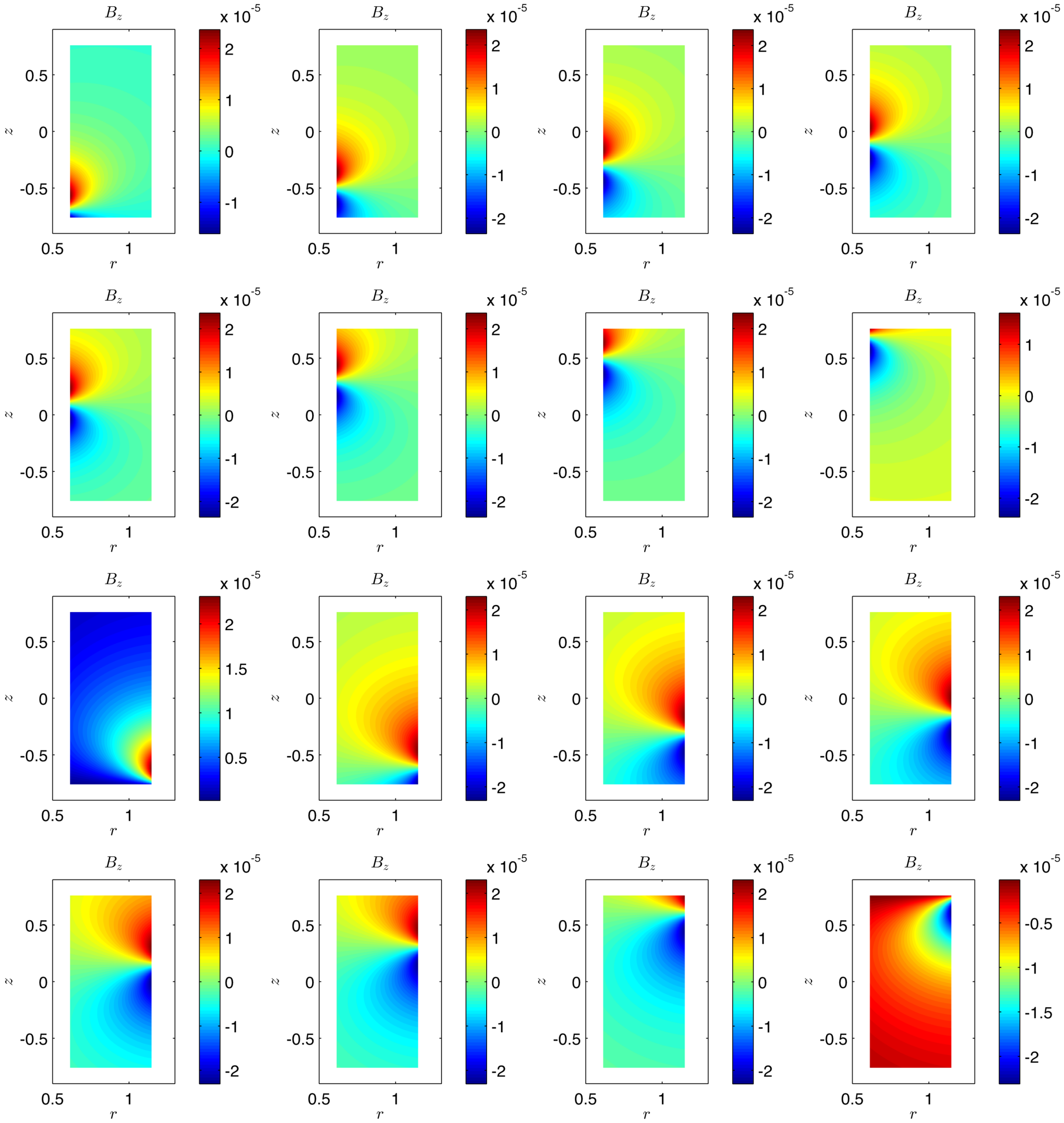


Figure 2: B_z component of the magnetic field produced by each E/F coil for a unit current.

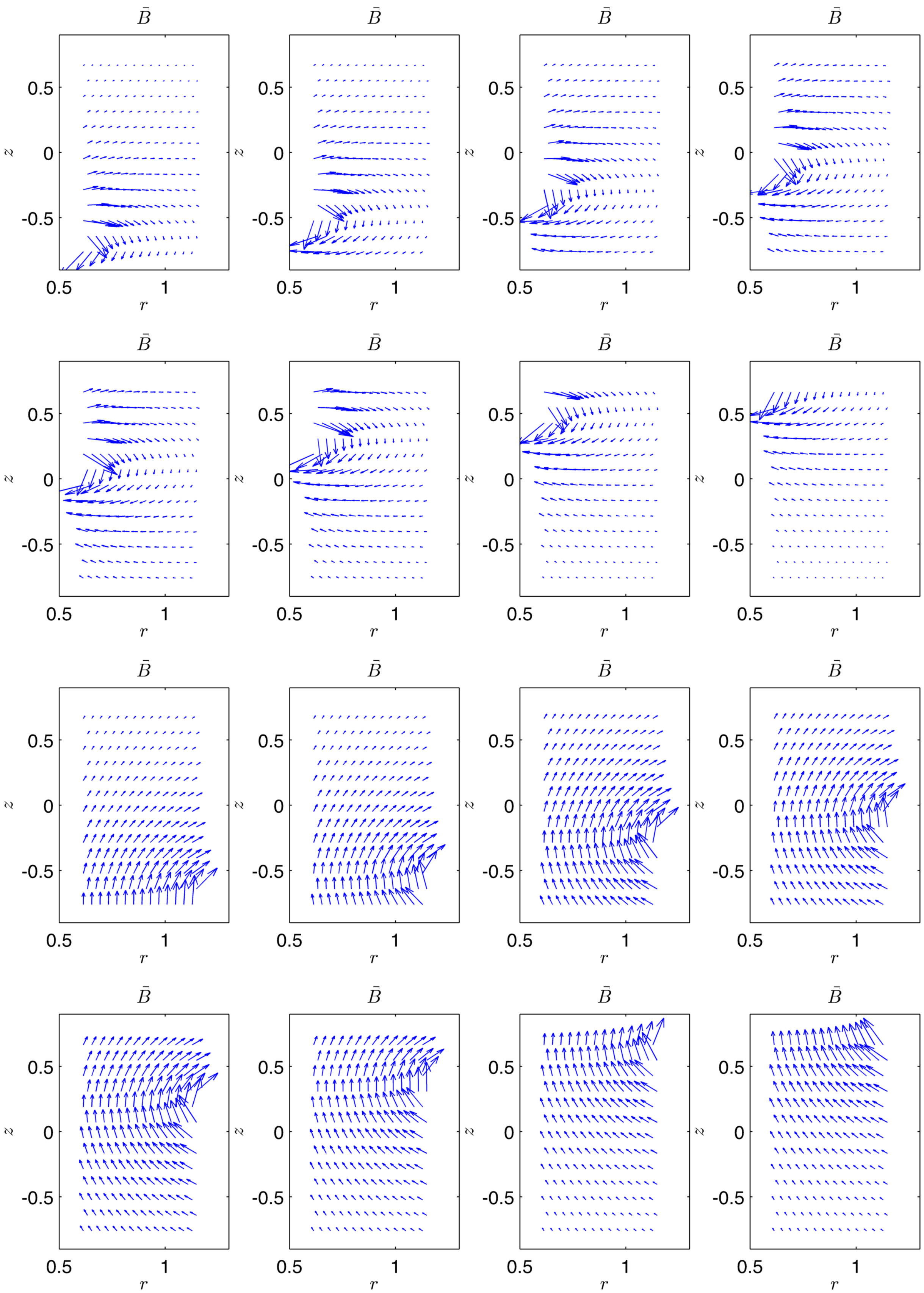


Figure 3: Magnetic field, $\vec{B}$, produced by each E/F coil for a unit current.

This can be made more compact:

Find $\boldsymbol{I}_a = [I_{a,1}, \ldots I_{a,N}]$*, such that:*

$$\min_{I_a} \left(\mathsf{B}\,\boldsymbol{I}_a - \boldsymbol{B}_{\text{required}}\right)^2 \ .$$

with:

$$\mathsf{B} = \begin{bmatrix} \mathsf{B}_r \\ \mathsf{B}_z \end{bmatrix} \quad \text{and} \quad \boldsymbol{B}_{\text{required}} = \begin{bmatrix} \boldsymbol{B}_{r,\text{required}} \\ \boldsymbol{B}_{z,\text{required}} \end{bmatrix}$$

The currents corresponding to the minimum are obtained by setting:

$$\frac{\partial \boldsymbol{J}}{\partial I_{a,i}} = 0, \quad i = 1, \ldots, N \ ,$$

with

$$\boldsymbol{J} = \left(\mathsf{B}\,\boldsymbol{I}_a - \boldsymbol{B}_{\text{required}}\right)^2 = \left(\mathsf{B}\,\boldsymbol{I}_a - \boldsymbol{B}_{\text{required}}\right)^t \left(\mathsf{B}\,\boldsymbol{I}_a - \boldsymbol{B}_{\text{required}}\right) \ .$$

This can be expanded, yielding:

$$\begin{aligned} \boldsymbol{J} &= \boldsymbol{I}_a^t \mathsf{B}^t \mathsf{B} \boldsymbol{I}_a - \boldsymbol{I}_a^t \mathsf{B}^t \boldsymbol{B}_{\text{required}} - \boldsymbol{B}_{\text{required}}^t \mathsf{B} \boldsymbol{I}_a + \boldsymbol{B}_{\text{required}}^t \boldsymbol{B}_{\text{required}} && (5) \\ &= \boldsymbol{I}_a^t \mathsf{B}^t \mathsf{B} \boldsymbol{I}_a - 2\boldsymbol{B}_{\text{required}}^t \mathsf{B} \boldsymbol{I}_a + \boldsymbol{B}_{\text{required}}^t \boldsymbol{B}_{\text{required}} \ . && (6) \end{aligned}$$

Now if we take the derivative of this with respect to $\boldsymbol{I}_a$ we obtain:

$$\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{I}_a} = 2\mathsf{B}^t \mathsf{B} \boldsymbol{I}_a - 2\mathsf{B}^t \boldsymbol{B}_{\text{required}} \ .$$

We now can build an algebraic system of equations:

$$\frac{\partial \boldsymbol{J}}{\partial I_{a,i}} = 0 \quad \Rightarrow \quad \mathsf{B}^t \mathsf{B} \boldsymbol{I}_a - 2\mathsf{B}^t \boldsymbol{B}_{\text{required}} = 0 \ .$$

Which can be easily inverted giving the following currents:

$$\boldsymbol{I}_a = \left(\mathsf{B}^t \mathsf{B}\right)^{-1} \mathsf{B}^t \boldsymbol{B}_{\text{required}} \ .$$

In our case we have:

$$\mathsf{B}_r = \texttt{G.Brxa} \quad \text{and} \quad \mathsf{B}_z = \texttt{G.Bzxa} \ ,$$

and

$$\boldsymbol{B}_{r,\text{required}} = 0 \ \text{T} \quad \text{and} \qquad \boldsymbol{B}_{z,\text{required}} = -0.1183 \ \text{T} \ .$$

This has been implemented in `Matlab` and the current distributions obtained for $\boldsymbol{I}_a$ are presented in Table 1. In Figure 4 the obtained magnetic field and the error with respect to the target magnetic field are presented.

Table 1: Current distribution obtained from Least-Squares formulation.

$I_{a,1}$	$I_{a,2}$	$I_{a,3}$	$I_{a,4}$	$I_{a,5}$	$I_{a,6}$	$I_{a,7}$	$I_{a,8}$
1446.6	592.8	883.6	781.1	781.1	883.6	592.8	1446.6

$I_{a,9}$	$I_{a,10}$	$I_{a,11}$	$I_{a,12}$	$I_{a,13}$	$I_{a,14}$	$I_{a,15}$	$I_{a,16}$
-1972.6	-318.4	-805.8	-660.1	-660.1	-805.8	-318.4	-1972.6

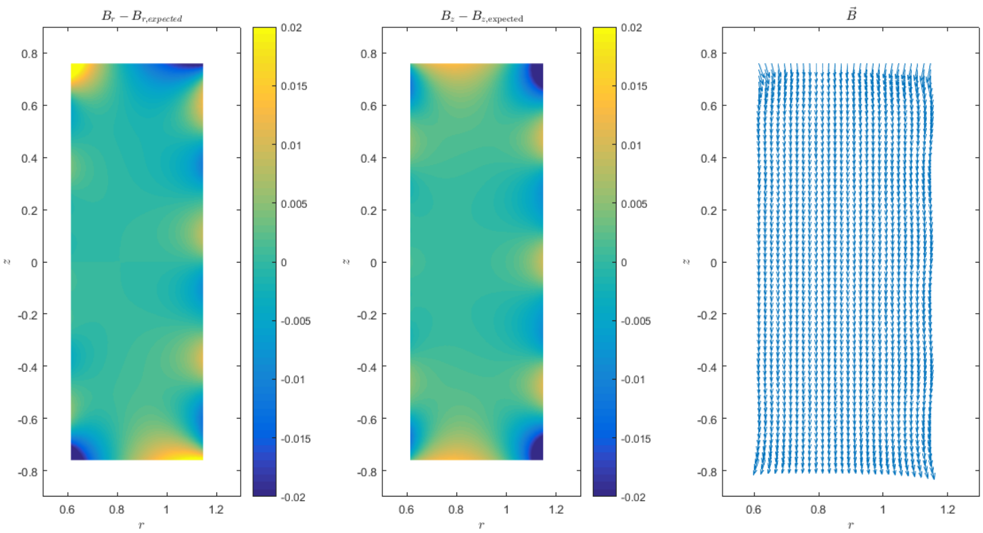


Figure 4: Least-Squares current distribution. Left: error between target B_r and obtained $B_{r,\text{LS}}$. Center: error between target B_z and obtained $B_{z,\text{LS}}$. Right: $\vec{B}_{\text{LS}}$ obtained from Least-Squares formulation.

c)

Calculate the vertical field required to balance the radial forces on an ITER plasma of $I_p = 15MA$ *with* $l_i = 1$, $\beta_p = 0.8$, $\kappa = 1.6$, $a = 2.0m$, $R = 6.2m$.

We follow the same procedure as in 3.2.a and use the final result from (3), now for $I_p = 15\text{MA}$ with $l_i = 1$, $\beta_p = 0.8$, $\kappa = 1.6$, $a = 2.0m$, $R = 6.2\text{m}$, to obtain:

$$B_z \approx -0.67157 \text{ T} .$$

d)

Suppose that the plasma beta suddenly decreases by 50% due to a sudden loss of confine-

ment. What consequences does this have for the radial position of the plasma? For a given vertical field, does it move inwards or outwards? What should be done to compensate this?

We start once more with the force balance equation (2):

$$m_p \frac{\mathrm{d}^2 R}{\mathrm{d}t^2} = \frac{\mu_0 I_p^2}{2} \left(\log \frac{8R}{a\sqrt{\kappa}} + \beta_p + \frac{l_i}{2} - \frac{3}{2} \right) + 2\pi R I_p B_z \,. \tag{7}$$

We now take:

$$F_p(\beta_p) = \frac{\mu_0 I_p^2}{2} \left(\log \frac{8R}{a\sqrt{\kappa}} + \beta_p + \frac{l_i}{2} - \frac{3}{2} \right) \,. \tag{8}$$

Substituting in (7) results in:

$$m_p \frac{\mathrm{d}^2 R}{\mathrm{d}t^2} = F_p(\beta_p) + 2\pi R I_p B_z \,.$$

Before the reduction of β_p to $\tilde{\beta}_p = 0.5\beta_p$ the plasma was in equilibrium, therefore:

$$m_p \frac{\mathrm{d}^2 R}{\mathrm{d}t^2} = F_p(\beta_p) + 2\pi R I_p B_z = 0 \,.$$

When β_p is reduced to $\tilde{\beta}_p = 0.5\beta_p$ we will have:

$$m_p \frac{\mathrm{d}^2 R}{\mathrm{d}t^2} = F_p(\tilde{\beta}_p) + 2\pi R I_p B_z < 0 \,.$$

This can be easily shown. First note that $F_p(\beta)$ depends linearly on β as can be easily seen in (8) and in Figure 5. Then note that $F_p(\beta) > 0$ and therefore $2\pi R I_p B_z < 0$. Therefore:

$$\tilde{\beta}_p = 0.5\beta_p \quad \Rightarrow \quad F_p(\beta_p) > F_p(\tilde{\beta}_p) \quad \Rightarrow \quad F_p(\tilde{\beta}_p) + 2\pi R I_p B_z < 0 \,.$$

If:

$$m_p \frac{\mathrm{d}^2 R}{\mathrm{d}t^2} < 0 \,,$$

This means that the plasma is experiencing a negative acceleration in the R-axis, therefore moving *inwards*.

To compensate for this we need to have again zero net force. This can be accomplished by reducing the absolute value of B_z.

Ex 3.2:Study of TCV plasma vertical position control

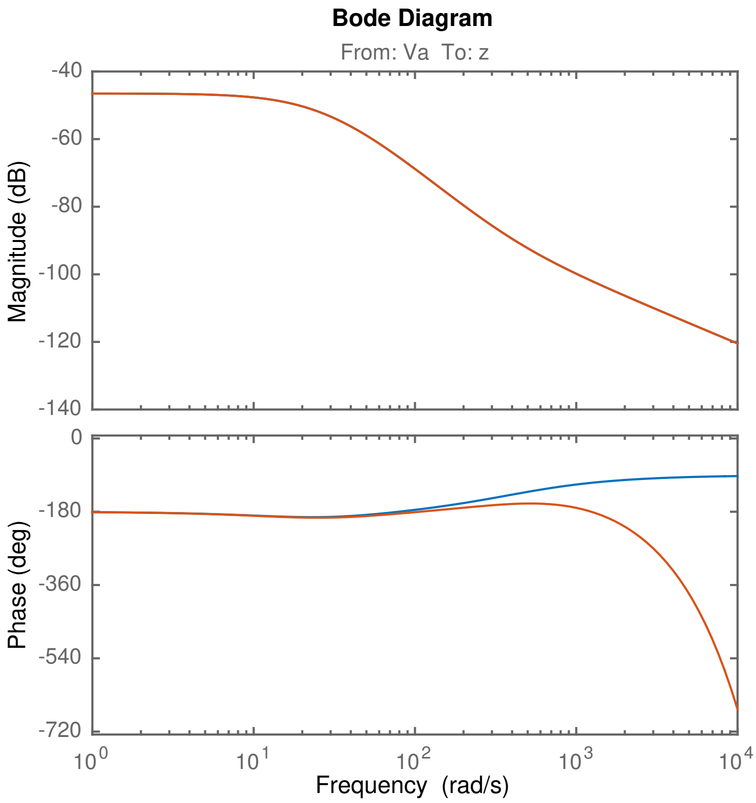
Contents


```
Vertical_control_model_TCV
```

a) Compare the Bode plots of the system with and without delay

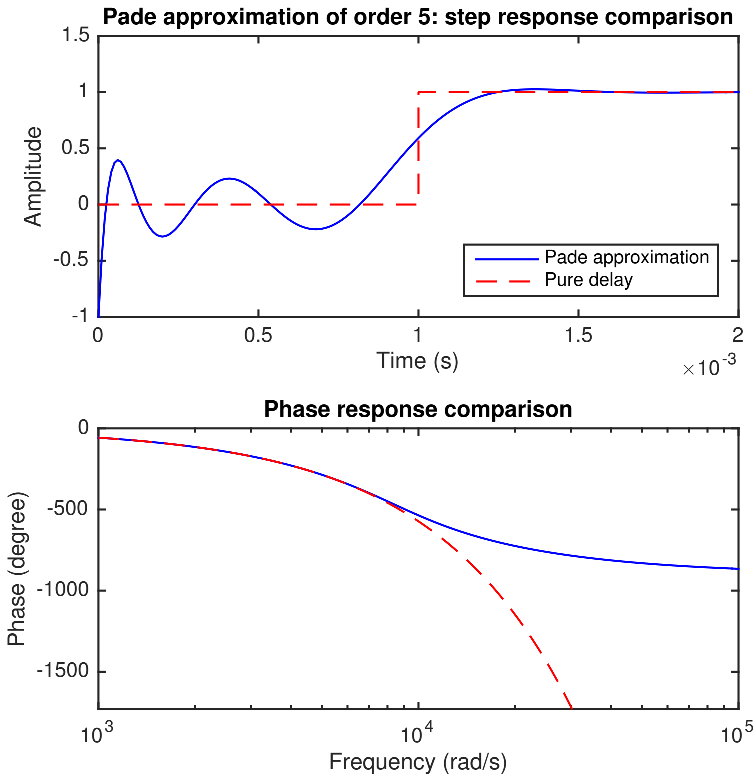
We see that the system with delay loses phase at high frequency.

```
figure(1); clf
set(gcf,'units','points','position',[0,0,400,400])
Wn = logspace(0,4,101);
bode(sysZ,sysD,Wn)
```



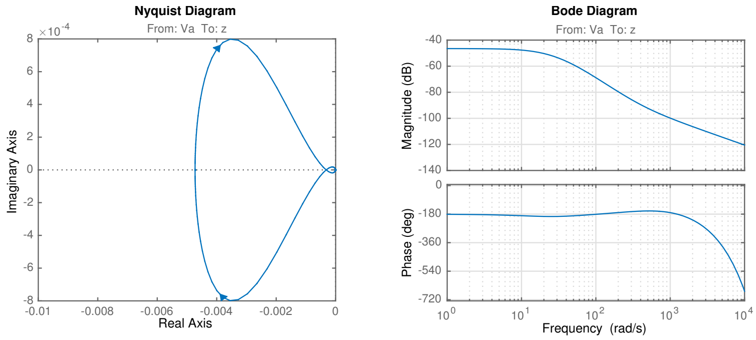
The Laplace transform of delay is exp(-tau*s). When included in series with the system, it generates a phase loss while the magnitude remains unchanged. The pade' approximant is a way of approximating the delay with a rational function of given order.

```
pade(1e-3,5);
```



b) Find the minimum and maximum P gain needed for stability

```
figure(2);clf; set(gcf,'units','points','position',[0,0,800,300])
subplot(121)
nyquist(sysD); set(gca,'xlim',[-0.01 0]);
subplot(122);
bode(sysD); grid on;
```



We see that there are three points in the bode diagram with phase = -180deg, hence three crossings of the negative real axis The first one is at w0 = 0 (log(w0) = -inf) , and the corresponding magnitude is the DC gain:

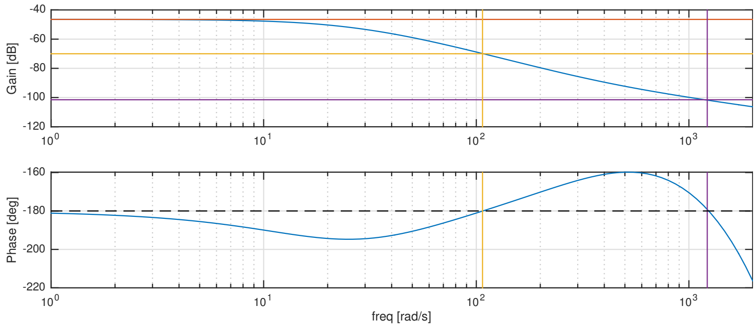
```
g0dB = 20*log10(abs(dcgain(sysD)));
disp(g0dB); %DC gain in dB
```

```
  -46.5060
```

the remaining two (w2 and w3) are found by inspection:

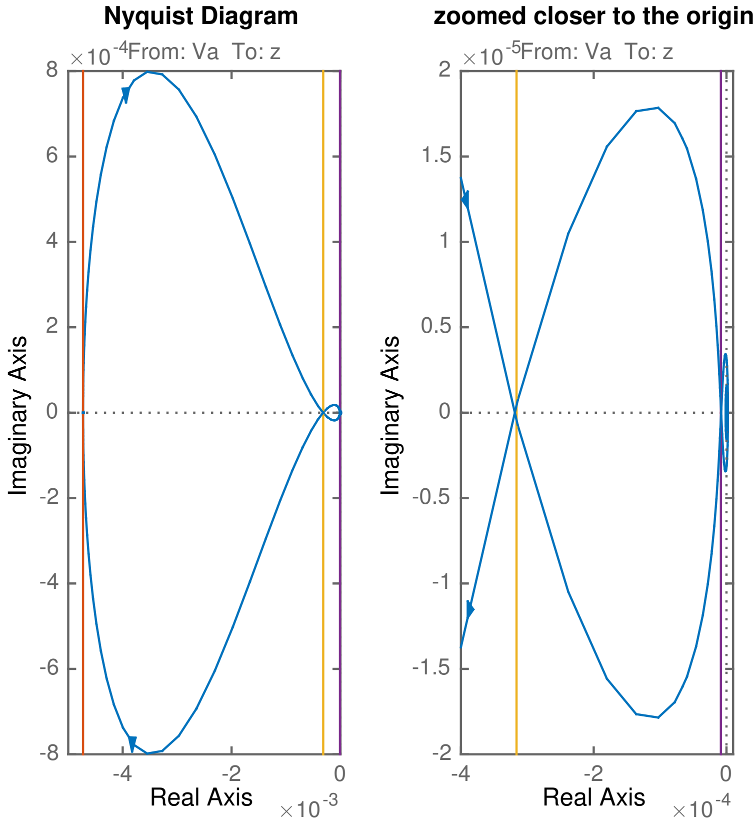
```
Wn = logspace(0,3.3,101);
[ga,ph] = bode(sysD,Wn);
ga=squeeze(ga); ph=squeeze(ph); % remove third dimension
w1 = 1.07e2; w2 = 1.22e3; % frequencies for crossings of -180deg
g1dB = -70; g2dB = -101.5; % gains at those frequencies
%
figure(2); clf;
subplot(211); semilogx(Wn,20*log10(ga)); hold on;
axis tight; grid on;

plot(Wn([1,end]),g0dB*[1,1]);
plot(Wn([1,end]),g1dB*[1,1]);
plot(Wn([1,end]),g2dB*[1,1]);
set(gca,'colorOrderIndex',3);
plot(w1*[1,1],[-120,-40])
plot(w2*[1,1],[-120,-40])
ylabel('Gain [dB]')

subplot(212); semilogx(Wn,ph); hold on;
axis tight; grid on;
set(gca,'colorOrderIndex',3);
plot(w1*[1,1],[-220,-160])
```

```
plot(w2*[1,1],[-220,-160])
plot(Wn([1,end]),-180*[1 1],'--k')
ylabel('Phase [deg]'); xlabel('freq [rad/s]');
```



We inspect the Nyquist plot and check which crossing corresponds to which gain

```
figure(1); clf;
subplot(121);
nyquist(sysD); set(gca,'xlim',[-0.005 0])
hold on;
set(gca,'colorOrderIndex',2);
g0 = 10^(g0dB/20);
g1 = 10^(g1dB/20);
g2 = 10^(g2dB/20);
plot(-g0*[1 1], [-1 1]);
plot(-g1*[1 1], [-1 1]);
plot(-g2*[1 1], [-1 1]);

subplot(122);
nyquist(sysD); set(gca,'xlim',[-0.0004 0.00001]); hold on;
set(gca,'colorOrderIndex',3);
plot(-g1*[1 1], [-1 1]);
plot(-g2*[1 1], [-1 1]);
title('zoomed closer to the origin')
```



To stabilise a system with 1 unstable pole, the Generalized Nyquist criterion tells us that we need 1 counter-clockwise encirclement of the -1 point.

For a counter-clockwise encirclement, we need to inflate (scale) the Nyquist diagram such that the -1 point falls in the second (smaller) encirclement

Thus the P gain needs to lie between 1/g1R and 1/g2R

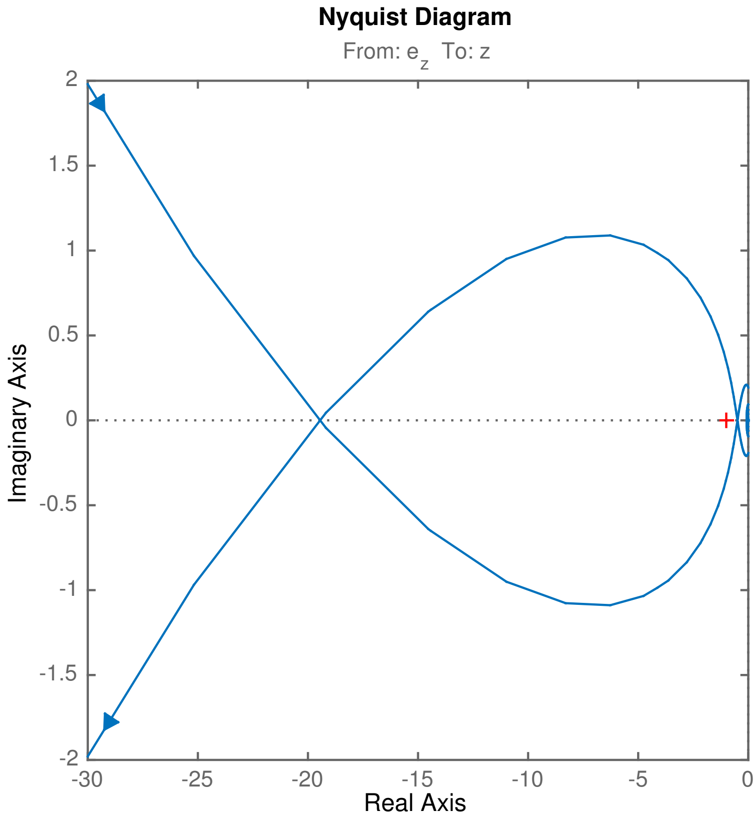
```
Kpmin = 1/g1;
Kpmax = 1/g2;
fprintf('Kpmin: %2.2f, Kpmax:%2.2f\n',[Kpmin,Kpmax]);
gmin = -g1dB;
gmax = -g2dB;
fprintf('Kpmin [dB]: %2.2f, Kpmax [dB]:%2.2f\n',[gmin,gmax]);
```

```
Kpmin: 3162.28, Kpmax:118850.22
Kpmin [dB]: 70.00, Kpmax [dB]:101.50
```

Note that the main consequence of the delay is that there is a maximum proportional gain for stability. Compared this to the case of the system without delay, where it is possible to increase the proportional gain without ever destabilizing the system (since the phase always stays above 90o) at high frequency. In theory you could get an infinite bandwidth this way! So it is important to consider the effect of the delay in this case.

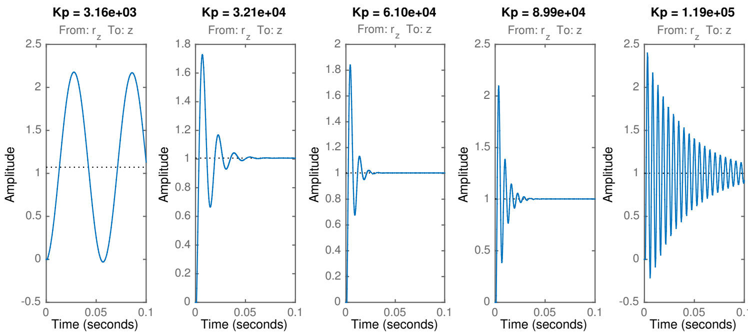
We check that indeed, if we are in between these gains, we encircle the -1 point once counter-clockwise,

```
Kp = (Kpmin+Kpmax)/2;
K = zpk(Kp); % make LTI object
K.InputName = 'e_z'; K.OutputName = 'V_a';

figure(1); clf;
nyquist(sysD*K); set(gca,'xlim',[-30 0]);
```



Let's plot the step responses for a few intermediate cases:

```
Kk = linspace(Kpmin,Kpmax,5);
figure(2); clf;
OL = sysD*Kk;
for ii=1:numel(Kk)
  sysCL = feedback(sysD*Kk(ii),1);
  sysCL.InputName = 'r_z';
  subplot(1,numel(Kk),ii)
  step(sysCL,0.1);
  title(sprintf('Kp = %2.2e',Kk(ii)));
end
```



We see that the response is indeed stable and has DC gain of approximately 1 already without adding an integral term. Still, the response is oscillatory in all cases.

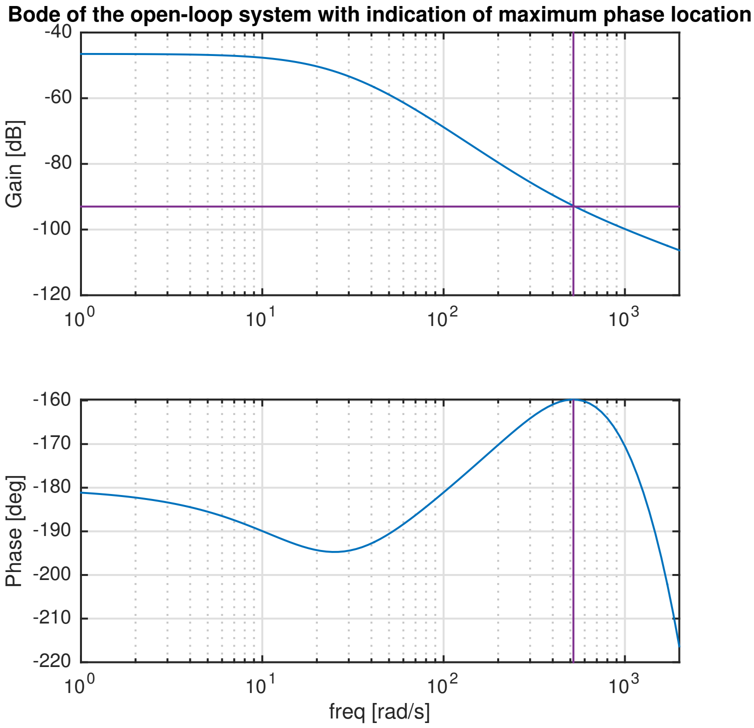
c) Maximize phase, gain and modulus margin separately with P control

Maximise Phase Margin

The phase margin is defined as the phase lead above -180deg at the frequency where the open loop magnitude crosses the 0dB line (cross-over frequency). For stabilization, the proportional gain has to be between Kpmin and Kpmax (and consequently the cross-over frequency lies between w1 and w2). From inspection of the Bode plot, it is therefore possible to observe that the maximum phase margin that can be obtained is 20deg (= -160 - (-180)) at omega=520rad/s:

```
wPM = 5.2e2; % frequency corresponding to phase at -160deg
gPM = -93;% gain at this frequency
figure(1); clf;
subplot(211);
semilogx(Wn,20*log10(squeeze(ga))); hold on;
axis tight; grid on;
set(gca,'colorOrderIndex',4);
```

```
plot(Wn([1,end]),gPM*[1,1]);
set(gca,'colorOrderIndex',4);
plot(wPM*[1,1],[-120,-40])
ylabel('Gain [dB]')
title('Bode of the open-loop system with indication of maximum phase location')

subplot(212); semilogx(Wn,squeeze(ph)); hold on;
axis tight; grid on;
set(gca,'colorOrderIndex',4);
plot(wPM*[1,1],[-220,-160])
ylabel('Phase [deg]'); xlabel('freq [rad/s]');
```



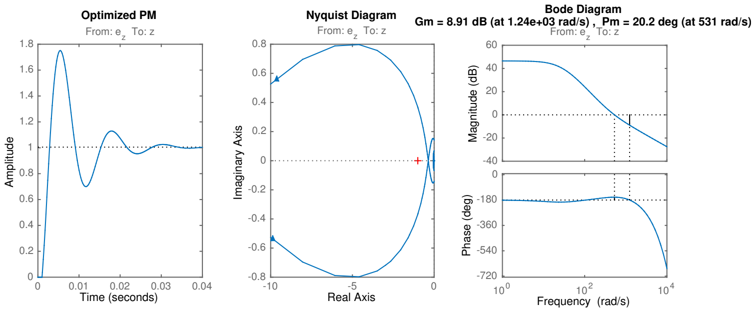
To maximize the phase margin, is therefore possible to set the controller gain to 1/(plant gain at the frequency where phase is maximized)

```
KpPM = 1/(10^(gPM/20)); %conversion in units from dB included here
disp(KpPM);
K.k = KpPM; % use LTI object
```

```
4.4668e+04
```

to get this closed loop response

```
sysCLPM = feedback(sysD*K,1);
figure(2); clf;
subplot(131); step(sysCLPM); title('Optimized PM')
subplot(132); nyquist(sysD*K); set(gca,'xlim',[-10 0]);
subplot(133); margin(sysD*K);
```



Maximise Gain Margin

The gain margin indicates by how much the open-loop gain can be increased or decreased before the 180o phase is crossed at 0dB (hence destabilizing the loop. We know that the maximum and minimum gains are, in dB

```
disp(20*log10([Kpmin,Kpmax]));
```

```
   70.0000   101.5000
```

To maximize the gain margin, we should choose a gain (in dB) exactly in between these two gains.

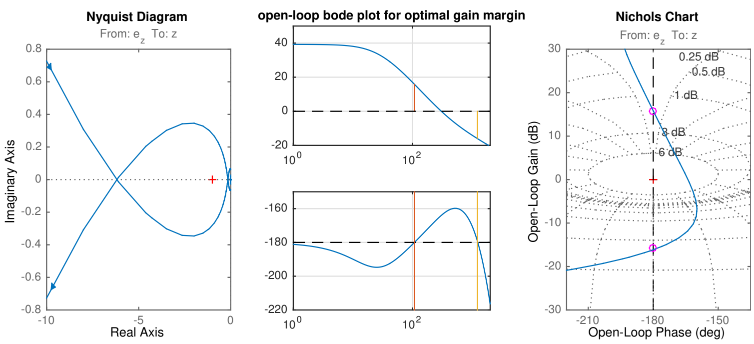
```
gGM = (gmin + gmax)/2;
KpGM = 10^(gGM/20);
K.k = KpGM; % use LTI object
disp(KpGM)
% The margin is then:
GM = gGM-gmin;
```

```
   1.9387e+04
```

In the following figures we show:

1. That the closed loop is indeed stable (Nyquist)
2. That the gain margin is maximized, since the amount (in dB) that we can decrease or increase the gain before destabilizing is equal
3. This can equivalently be seen in the so-called Nichols plot of Open-loop phase vs gain. Note that the open-loop crosses the -180deg line twice at equal distance from the -1 point.

```
figure(2); clf;
% nyquist
subplot(131); nyquist(sysD*K); set(gca,'xlim',[-10 0]);
% bode
subplot(232); semilogx(Wn,20*log10(ga*KpGM)); hold on;
plot(w1*[1,1],[0,GM]); plot(w2*[1,1],[0,-GM]);
plot(Wn([1,end]),0*[1 1],'--k'); grid on;
set(gca,'Xlim',Wn([1,end]),'Ylim',[-20,50]); grid on;
title('open-loop bode plot for optimal gain margin')
subplot(235); semilogx(Wn,ph); hold on;
plot(Wn([1,end]),-180*[1 1],'--k');
plot(w1*[1,1],[-220,-150]); plot(w2*[1,1],[-220,-150]);
set(gca,'Xlim',Wn([1,end]),'Ylim',[-220,-150]); grid on;
% nichols
subplot(133); nichols(sysD*K); grid on;
set(gca,'xlim',[-220,-135],'ylim',[-30 30]); hold on;
plot(-180*[1 1],[-30,30],'k--'); plot([-180,-180],GM*[-1 1],'om')
```



Maximise Modulus Margin

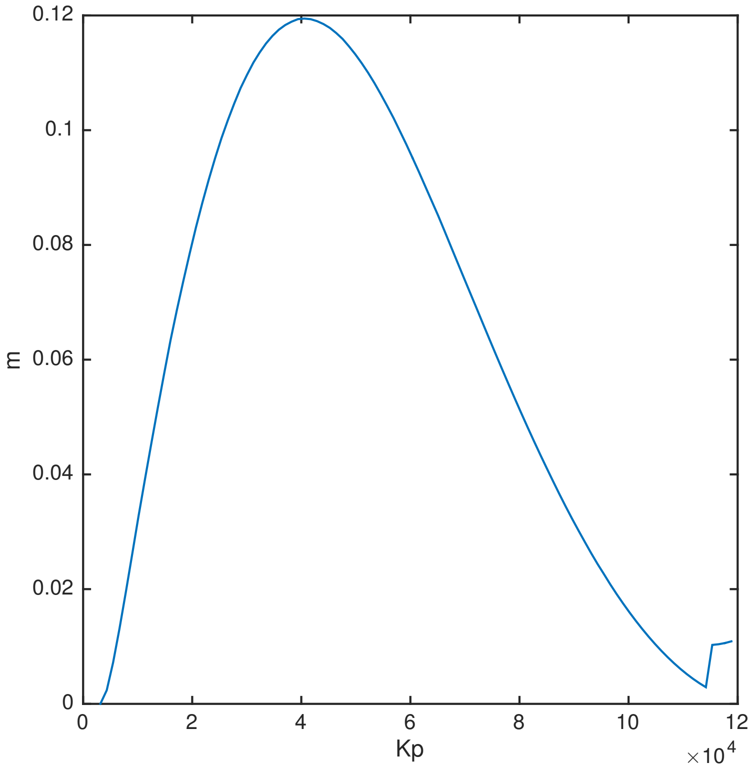
The modulus margin the minimum distance of the Nyquist contour from the -1 point on the complex plane. This can be defined as a minimization over the frequencies w of the function `KP - (-1)` (where P is the plant)

m = min_w (`KP - (-1)`)

Therefore the best controller k for this purpose is the one maximizing m

m_best = max_k (min_w (`1 + kG`))

```
kk=linspace(Kpmin,Kpmax,100); %look for the optimized KpMM in this set of values
m=zeros(1,length(kk));
for ii = 1:length(kk)
  [RR,II]=nyquist(1+kk(ii)*sysD); %take real and imaginary part of the transfer 1+KG
  R=zeros(1,length(RR)); I=zeros(1,length(II));
  R(1,:)=RR; I(1,:)=II;
  m(ii)=min(R.^2 + I.^2); % compute the min over all frequencies of |1+KG|
end
[MM,ind]=max(m); %find the maximum modulus margin
figure(1); clf
plot(kk,m);
xlabel('Kp'); ylabel('m');
KpMM=kk(ind);
```



The value of the modulus margin is:

```
disp(20*log10(1/MM))% in dB
disp(MM);
disp(KpMM);
```

```
18.4558
```

```
    0.1195

   4.0556e+04
```

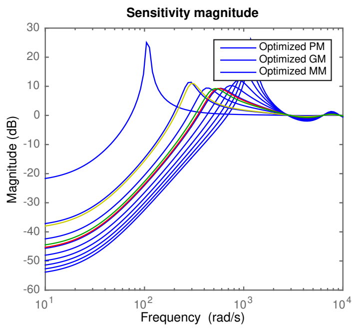
This is generally considered too large, usually we want the margin to be <6dB

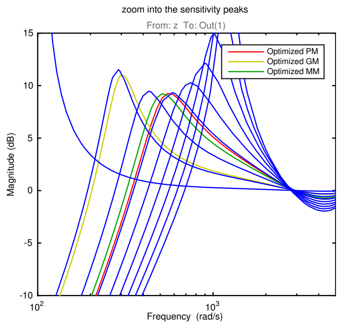
The modulus margin is equivalent to the inverse of the peak of the sensitivity function

S = 1 / (1 + K P)

So let's plot some sensitivity functions for various gains and check:

```
SensPM = 1/(1+sysD*KpPM);
SensGM = 1/(1+sysD*KpGM);
SensMM = 1/(1+sysD*KpMM);

figure(2); clf;
subplot(121); hold on
Wnplot = logspace(1,4,101);
KMM = linspace(Kpmin,Kpmax,9);
for ii =1:length(KMM)
  Sens(ii) = 1/(1+KMM(ii)*sysD);
  bodemag(Sens(ii),Wnplot,'b');
end
bodemag(SensPM,Wnplot,'r');
bodemag(SensGM,Wnplot,'y');
bodemag(SensMM,Wnplot,'g');

legend('Optimized PM','Optimized GM','Optimized MM')
title('Sensitivity magnitude')

subplot(122); hold on
opts = bodeoptions; opts.Xlim=[10^2,5*10^3]; opts.Ylim=[-10,15];

bodemag(SensPM,Wnplot,'r',opts);
bodemag(SensGM,Wnplot,'y',opts);
bodemag(SensMM,Wnplot,'g',opts);
for ii =1:length(KMM)
  bodemag(Sens(ii),'b',opts);
end
legend('Optimized PM','Optimized GM','Optimized MM')
title('zoom into the sensitivity peaks')
```





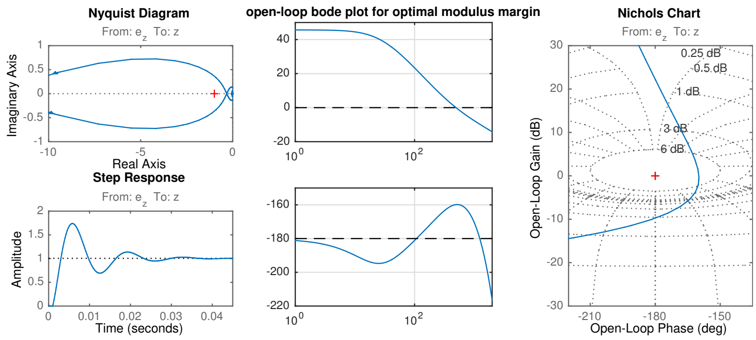
The optimized controller for the modulus margin has a gain which is close to the one provided by the optimized phase magin. In fact

```
disp([KpMM,KpPM,KpGM]);
```

```
  1.0e+04 *

   4.0556    4.4668    1.9387
```

The closed loop response is

```
K.k = KpMM; Lp0 = sysD*K;
figure(2);  clf;
% nyquist
subplot(231); nyquist(Lp0); set(gca,'xlim',[-10 0]);
subplot(234); step(feedback(Lp0,1));

% bode
subplot(232); semilogx(Wn,20*log10(ga*KpMM)); hold on;
plot(Wn([1,end]),0*[1 1],'--k'); grid on;
set(gca,'Xlim',Wn([1,end]),'Ylim',[-20,50]); grid on;
title('open-loop bode plot for optimal modulus margin')
subplot(235); semilogx(Wn,ph); hold on;
plot(Wn([1,end]),-180*[1 1],'--k');
set(gca,'Xlim',Wn([1,end]),'Ylim',[-220,-150]); grid on;
% nichols
subplot(133); nichols(Lp0); grid on;
set(gca,'xlim',[-220,-135],'ylim',[-30 30]); hold on;
```

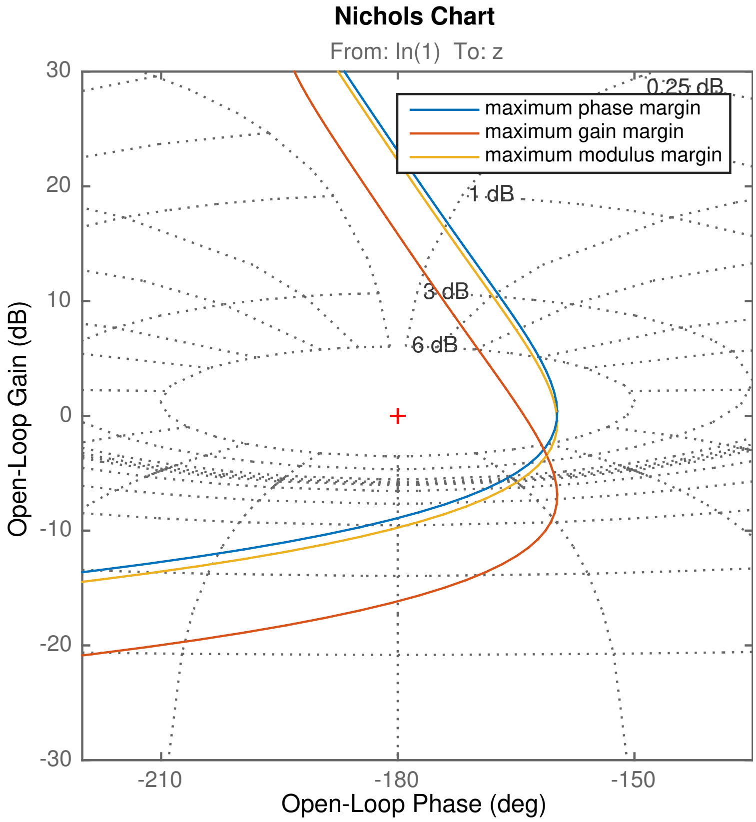


The three cases can conveniently be compared usign a Nichols plot. In this plot, we see how

1. The maximum gain margin means maximizing the distance from -1 along the phase = -180 axis;
2. The maximum phase margin means maximizing the distance from -1 along the gain = 0 axis;
3. The maximum modulus margin means maximizing the norm of the distance to -1 in the Re-Im plane (shown in the dotted grid lines)

Because of the characteristics of the plant, the gains for maximum phase margin and maximum modulus margin points are close to each other.

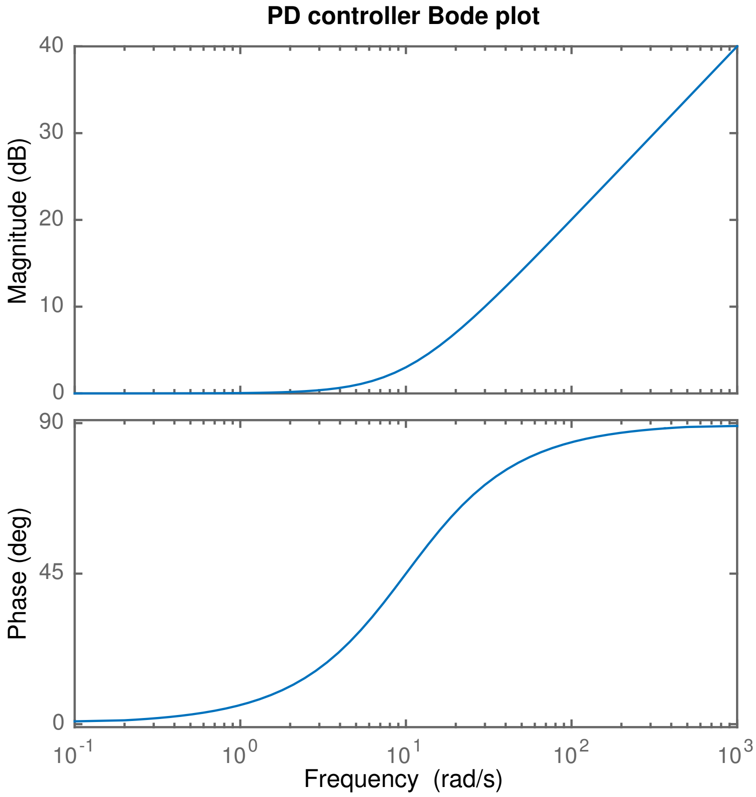
```
figure(1); clf;
nichols(sysD*KpPM,sysD*KpGM,sysD*KpMM,Wnplot); grid on;
set(gca,'xlim',[-220,-135],'ylim',[-30 30]); hold on;
legend('maximum phase margin','maximum gain margin','maximum modulus margin')
```



d) Optimal PD controller

A PD controller has one real zero, which increases the gain and phase at frequencies higher than the frequency of the zero.

```
Kp0 = 1; Td0 = 0.1;
s=tf('s');
Kpd = Kp0*(1 + s*Td0); %PD with zero placed at ps (zk = 1/Td = ps)
figure(1); clf;
bode(Kpd);
title('PD controller Bode plot');
```



We study the effect varying Kp and Td gains on the bandwidth and on the modulus margin m by computing bandwidth and modulus margin on a 2D grid of Kp vs Td.

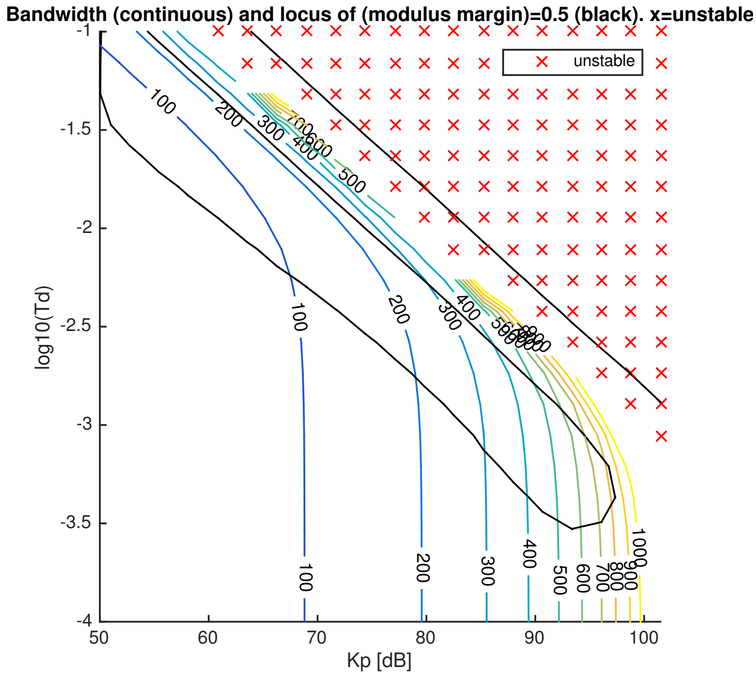
```
Kp = logspace(log10(Kpmin/10),log10(Kpmax),20);
Td = [0,logspace(-4,-1,20)];
b_surf = zeros(numel(Td),numel(Kp));
m_surf = zeros(numel(Td),numel(Kp));
for jj= 1:numel(Td)
  fprintf('.',jj,numel(Td))
  for ii = 1:numel(Kp)
    Kpd = Kp(ii)*(1+s*Td(jj)); L = sysD * Kpd; % Controller and open-loop
    [RR,II]=nyquist(L);
    mmarg = min(sqrt((RR+1).^2 + II.^2)); %modulus margin as min_w (|KG - 1)|)
    % bandwidth as point where mag(L) crosses 0dB
    mag = bode(L,Wn); magdB = 20*log10(mag); [iw] = find(magdB<0,1,'first');
    if isempty(iw)
      wb = NaN;
      % mag(L>1), always unstable due to many nyquist encirclements.
    elseif iw==1
```

```
      wb = NaN;
      % first point below 1, no encirclement of -1 point, so unstable!
    else
      % linear interpolation (in dB) to find bandwidth
      wb = Wn(iw)+(Wn(iw)-Wn(iw-1))/(magdB(iw-1)-magdB(iw)) * magdB(iw);
    end
    m_surf(jj,ii) = mmarg;
    b_surf(jj,ii) = wb;
  end
end
fprintf('\n');
```

· · · · · · · · · · · · · · · · · · · ·

Plot the results in a 2D plot of Kp vs Ts We show the loci where modulusmargin = 0.5 and contours of the bandwidth. Points marked with a cross yield an unstable closed-loop.

```
figure(1); clf
[XX,YY] = meshgrid(20*log10(Kp),log10(Td));
iunst = isnan(b_surf); oo = ones(size(b_surf));
hu=plot3(XX(iunst),YY(iunst),oo(iunst),'xr'); hold on; view(2); hold on;
%
[C,h]=contour(XX,YY,m_surf,0.5*[1 1],'k'); hold on %contour where modulus margin = 0.5

contour(XX,YY,b_surf,[100:100:1000],'ShowText','on'); %2D surface of the bandwidth

xlabel('Kp [dB]'); ylabel('log10(Td)')
title('Bandwidth (continuous) and locus of (modulus margin)=0.5 (black). x=unstable')
legend(hu,'unstable');
```



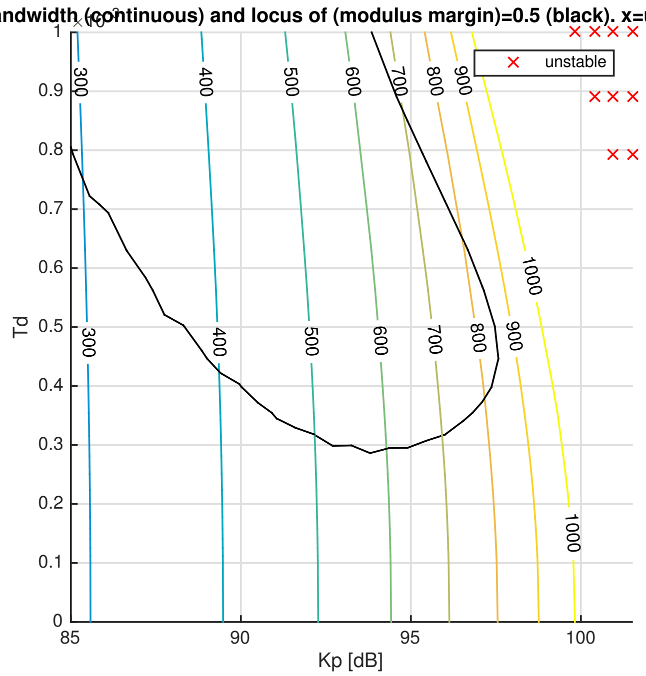
We see that the maximum bandwidth is obtained in the corner close to the maximum P gain; We perform a second scan in that area

```
Kp = logspace(85/20,log10(Kpmax),31);
Td = [0,logspace(-4,-3,21)];
b_surf = NaN(numel(Td),numel(Kp));
m_surf = zeros(numel(Td),numel(Kp));
for jj= 1:numel(Td)
  fprintf('.')
  for ii = 1:numel(Kp)
    Kpd = Kp(ii)*(1+s*Td(jj)); L = sysD * Kpd; % Controller and open-loop
    [RR,II]=nyquist(L);
    mmarg = min(sqrt((RR+1).^2 + II.^2)); %modulus margin as min_w (|KG - 1)|)
    % bandwidth as point where mag(L) crosses 0dB
    mag = bode(L,Wn); magdB = 20*log10(mag); [iw] = find(magdB<0,1,'first');
    if isempty(iw)
      wb = NaN;
      % mag(L>1), always unstable due to many nyquist encirclements.
    elseif iw==1
      wb = NaN;
      % first point below 1, no encirclement of -1 point, so unstable!
    else
      % linear interpolation (in dB) to find bandwidth
      wb = Wn(iw)+(Wn(iw)-Wn(iw-1))/(magdB(iw-1)-magdB(iw)) * magdB(iw);
```

```
    end
    m_surf(jj,ii) = mmarg;
    b_surf(jj,ii) = wb;
  end
end
fprintf('\n');
```

....

```
figure(1); clf
[XX,YY] = meshgrid(20*log10(Kp),Td);
iunst = isnan(b_surf); oo = ones(size(b_surf));
hu=plot3(XX(iunst),YY(iunst),oo(iunst),'xr'); hold on; view(2); hold on;
%
[C,h]=contour(XX,YY,m_surf,0.5*[1 1],'k'); hold on %contour where modulus margin = 0.5

contour(XX,YY,b_surf,[100:100:1000],'ShowText','on'); %2D surface of the bandwidth

xlabel('Kp [dB]'); ylabel('Td'); grid on;
title('Bandwidth (continuous) and locus of (modulus margin)=0.5 (black). x=unstable')
legend(hu,'unstable');
```



We find the maximum roughly at

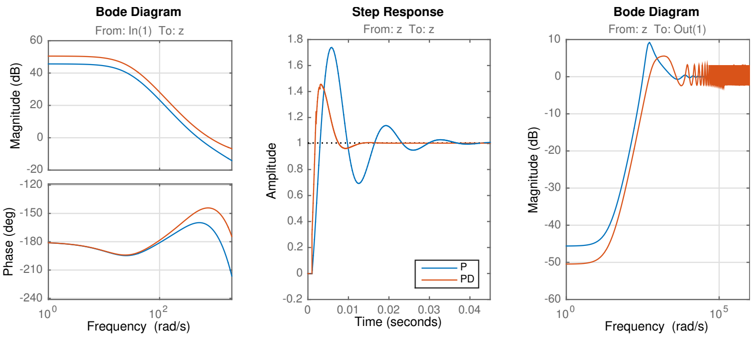
```
Td0 = 0.00045; Kp0 = 10^(97/20);
disp(Kp0); disp(Td0);
```

```
Kpd0 = Kp0*(1+s*Td0); L0 = Kpd0*sysD; % Controller and open-loop
Lpd0 = Kpd0*sysD;
```

```
7.0795e+04

4.5000e-04
```

Note the nice non-oscillatory step response and good gain, phase margins and sensitivity below 6dB

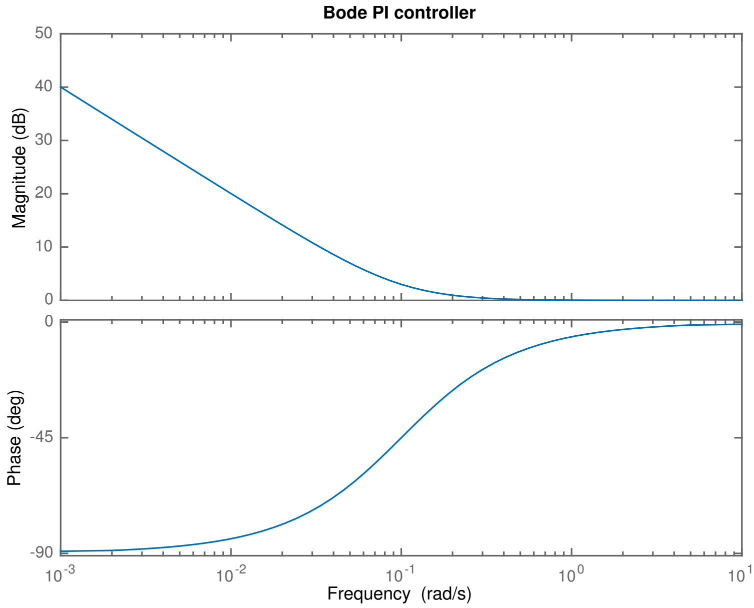
```
figure(2); clf;
subplot(131)
bode(Lp0,Lpd0,Wn); grid on;
subplot(132);
step(Lp0/(1+Lp0),Lpd0/(1+Lpd0)); legend('P','PD','location','southeast')
subplot(133)
bodemag(1/(1+Lp0),1/(1+Lpd0)); grid on;
```



e) Optimal PID controller

The PD controller is put in series with a PI controller, whose transfer function is

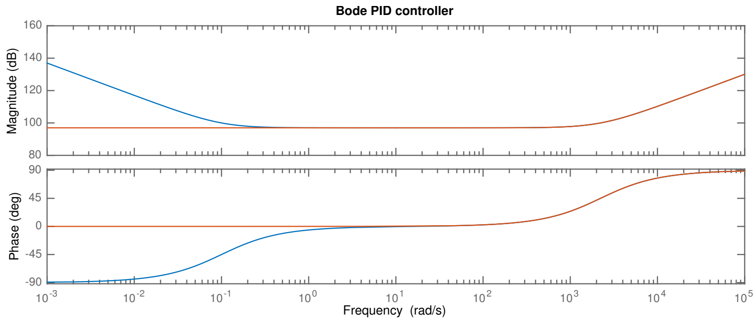
```
Ti = 10;
Kpi = (1+1/(Ti*s));
figure(3); clf
bode(Kpi);
title('Bode PI controller')
```



The presence of an integrator introduces a phase loss at low frequencies but allows perfect tracking of the reference for the position (i.e. the error e = z-z_ref goes to zero for t -> inf). In series with the previously defined PD controller, we have a PID controller, i.e. featuring a proportional, derivative and integral term

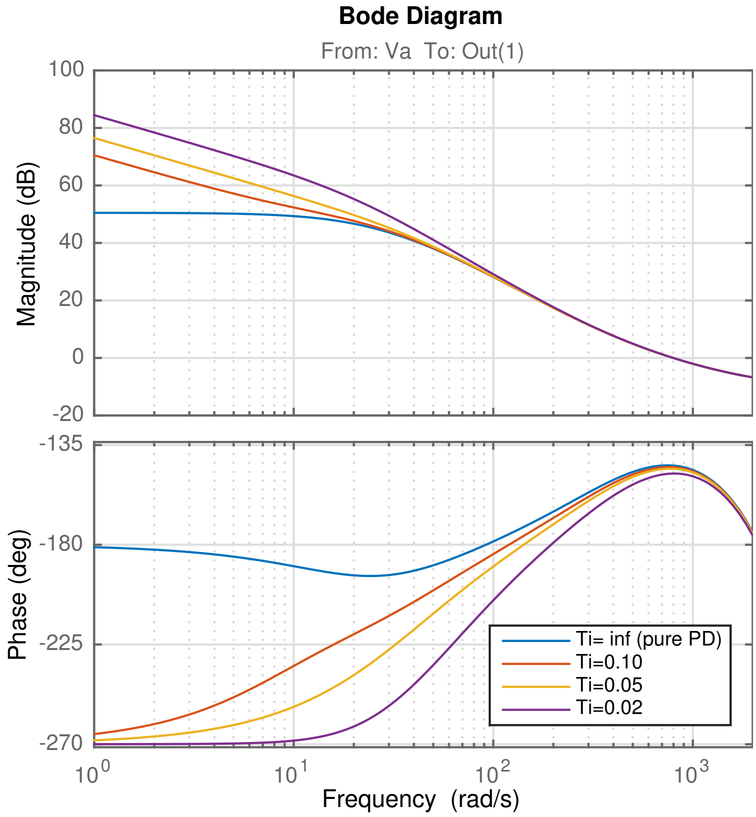
```
Kpid = Kpi * Kpd0

figure(2); clf;
bode(Kpid,Kpd0);
title('Bode PID controller')
```

```
Kpid =

  318.6 s^2 + 7.08e05 s + 7.079e04
  --------------------------------
                10 s

Continuous-time transfer function.
```



To avoid affecting the stability, we must avoid degrading the phase of the controller near the crossover frequency. We can evaluate this using bode plots:

```
figure(1); clf;
Ti = [0.1,0.05,0.02];

for ii=1:numel(Ti);
  Kpid(ii) = (1+1/(Ti(ii)*s))*Kpd0;
  legendstr{ii+1} = sprintf('Ti=%2.2f',Ti(ii));
end
bode(Kpd0*sysD,Kpid(1)*sysD,Kpid(2)*sysD,Kpid(3)*sysD,logspace(0,3.3,101)); grid on;
legendstr = {'Ti= inf (pure PD)',legendstr{2:end}};
legend(legendstr,'location','southeast')
```

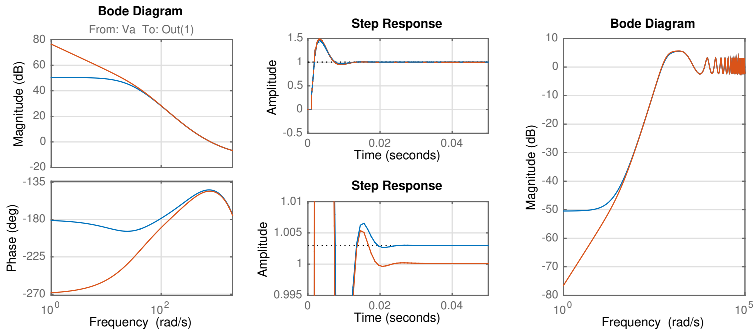


We see that already Ti=2e-2 starts to affect the phase around the crossover frequency, so we choose this as the limit.

```
Ti0 = 5e-2;
Kpid0 =  (1+1/(Ti0*s))*Kpd0;
Lpid0 = Kpid0*sysD;
```

We obtain perfect tracking at the expense of a slightly more oscillatory transient response;

```
figure(2); clf;
subplot(131)
bode(L0,Lpid0,Wn); grid on;
subplot(232);
step(L0/(1+L0),Lpid0/(1+Lpid0),0.05); grid on;
subplot(235);
step(L0/(1+L0),Lpid0/(1+Lpid0),0.05); grid on; set(gca,'Ylim',[0.995 1.01])
subplot(133)
bodemag(1/(1+L0),1/(1+Lpid0)); grid on;
```



f) Introduction of roll-off

The purpose of the roll off is to limit the high frequency noise amplification. This can be studied with the control sensitivity function which is the transfer function high frequency noise (or reference signals) and the control signal entering the plant.

CS = K / (1 + KG).

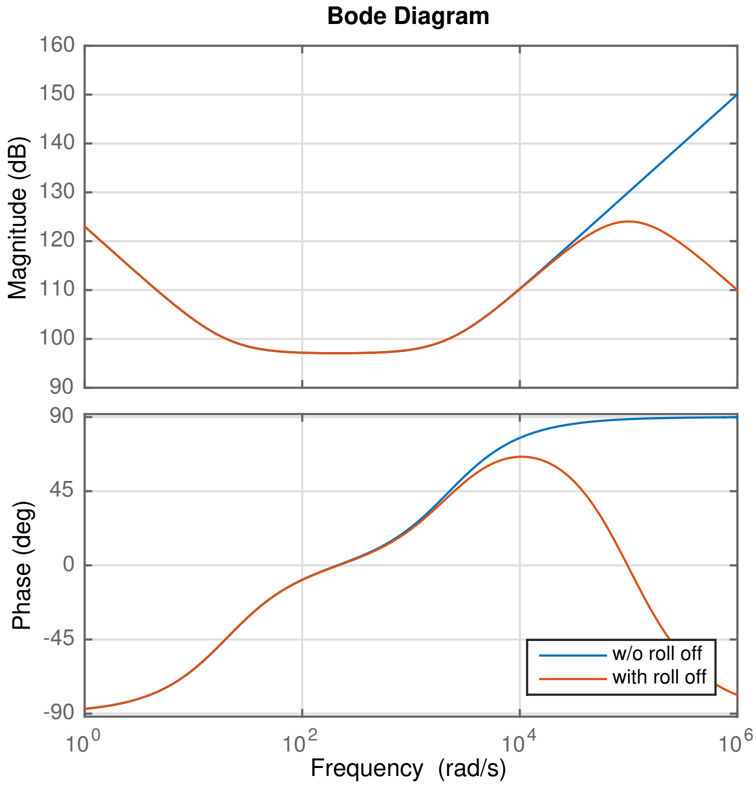
We see that to limit CS at high frequency, we need to limit the magnitude of K at high frequency.

Introducing a roll off in the controller of the form

K = Kpid * 1 / (1 + Tr * s^2)

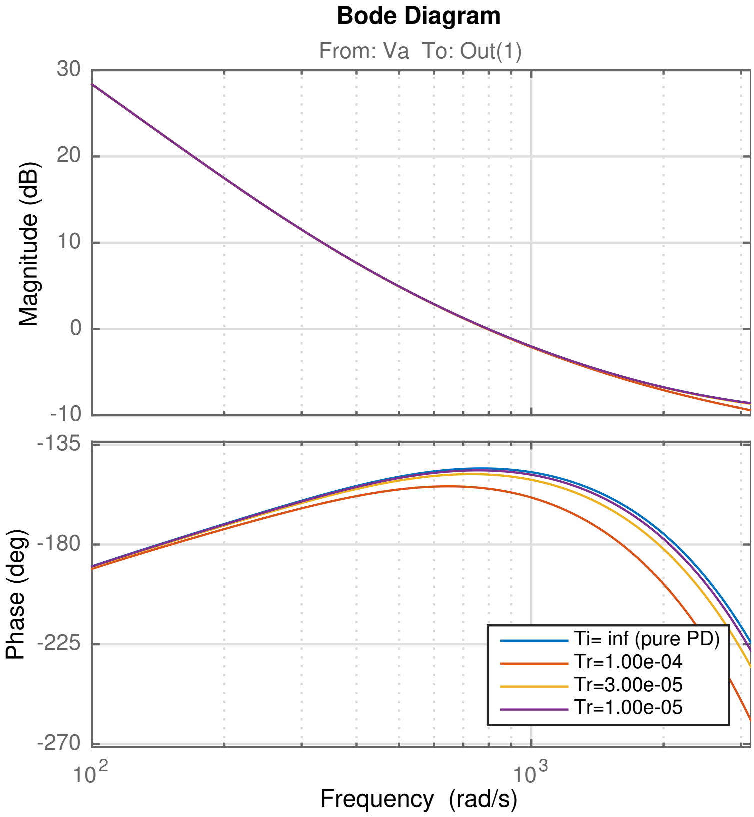
The controller becomes, for example:

```
Tr = 1e-5;
K = Kpid0 * 1/(1 + Tr * s)^2;
figure(1); clf;
bode(Kpid0,K,logspace(0,6,101)); grid on;

legend('w/o roll off','with roll off','location','southeast')
```



The presence of two extra poles in the controller leads not only to a reduction of the noise at high frequencies but also to a phase loss around w = 1/Tr. Therefore, the stabilizing effect of the controller can be lost if 1/Tr becomes too large. This can be seen in the open-loop bode plot.

```
figure(1); clf;
Tr = [1e-4,3e-5,1e-5];

for ii=1:numel(Tr);
  Kpidr(ii) = Kpid0/(1+Tr(ii)*s)^2;
  legendstr{ii+1} = sprintf('Tr=%2.2e',Tr(ii));
end
bode(Kpid0*sysD,Kpidr(1)*sysD,Kpidr(2)*sysD,Kpidr(3)*sysD,logspace(2,3.5,101)); grid on;
legendstr = {'Ti= inf (pure PD)',legendstr{2:end}};
legend(legendstr,'location','southeast')
```



The smallest value for Tr not affecting significantly the performance of the closed loop is therefore

```
Tr0=3e-5;
```

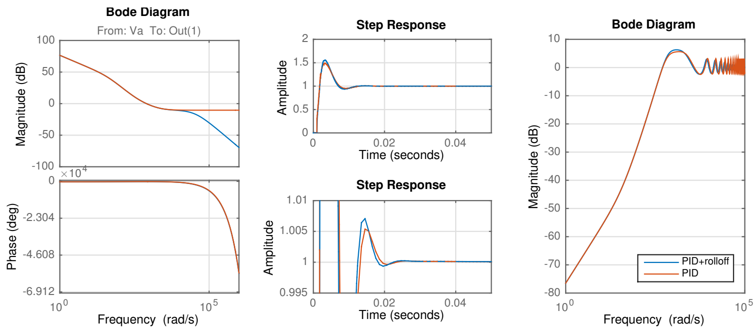
and the final controller is

```
Kpidr0 = Kpid0/((1 + Tr0*s)^2);
Lpidr0 = Kpidr0*sysD;
```

Check the closed-loop, note that it is almost the same except for the (very welcome) reduction of the high-frequency sensitivity.

```
figure(2); clf;
subplot(131)
bode(Lpidr0,Lpid0,logspace(0,6,101)); grid on;
subplot(232);
step(Lpidr0/(1+Lpidr0),Lpid0/(1+Lpid0),0.05); grid on;
subplot(235);
```

```
step(Lpidr0/(1+Lpidr0),Lpid0/(1+Lpid0),0.05); grid on; set(gca,'Ylim',[0.995 1.01])
subplot(133)
bodemag(1/(1+Lpidr0),1/(1+Lpid0)); grid on;
legend('PID+rolloff','PID','location','southeast')
```



g) Comparison of the all the controllers

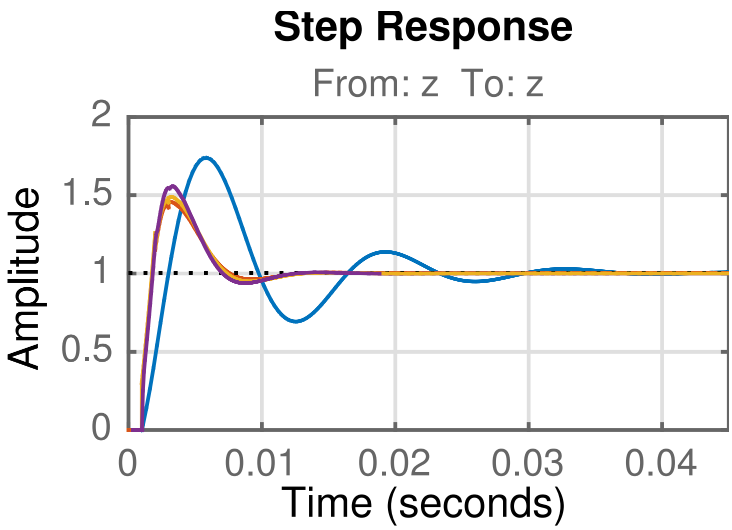
First we compute all the sensitivity functions

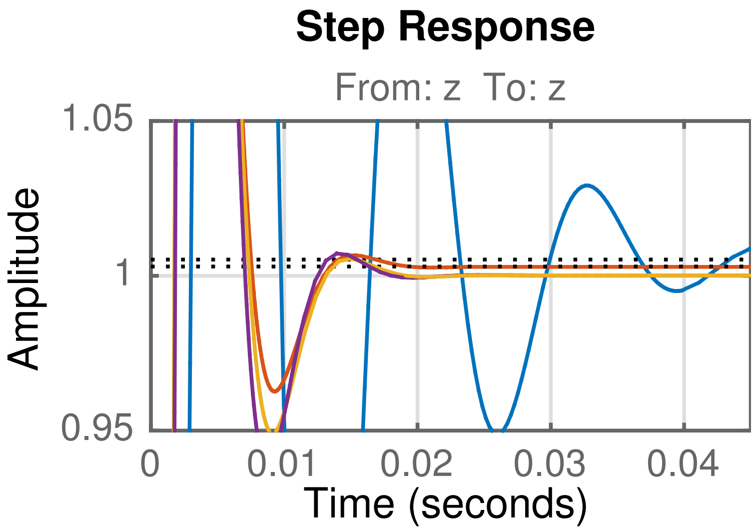
```
% Sensitivity functions
Sp0 = 1/(1+Lp0); % P
Spd0 = 1/(1+Lpd0); % PD
Spid0 = 1/(1+Lpid0); % PID
Spidr0 = 1/(1+Lpidr0); % with rolloff

% Control sensitivity
Cp0 = Kp0 * Sp0;
Cpd0 = Kpd0 * Spd0;
Cpid0 = Kpid0 * Spid0;
Cpidr0 = Kpidr0 * Spidr0;

% closed-loop transfer function
CLp0 = Lp0*Sp0;
CLpd0 = Lpd0*Spd0;
CLpid0 = Lpid0*Spid0;
CLpidr0 = Lpidr0*Spidr0;
```

Step responses: The step response with purely P gain is clearly the worst. The extra phase margin added by the PD and PID controllers gives a much better response. The controllers with integral gain have zero steady-state error (step response converges to 1) but the PD controller already had a quite small steady-state error due to its high gain at low frequency.

```
figure(2); clf;
```

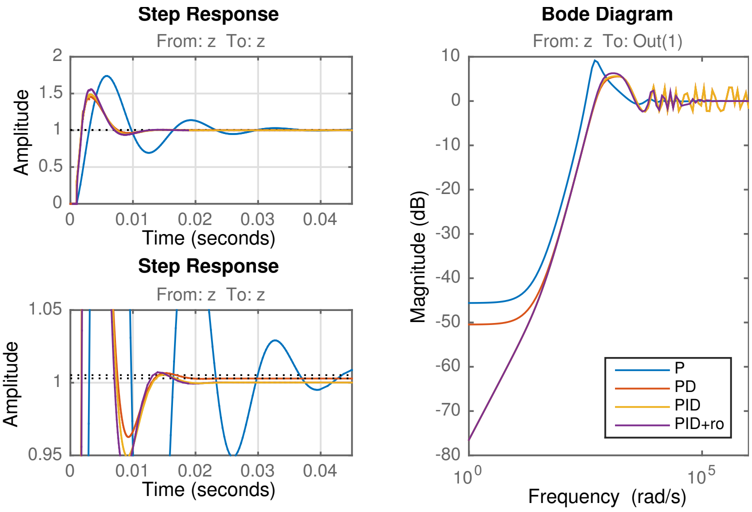
```
subplot(231);
step(CLp0,CLpd0,CLpid0,CLpidr0);grid on;
subplot(234);
step(CLp0,CLpd0,CLpid0,CLpidr0);grid on;
set(gca,'ylim',1+[-0.05,0.05]);
```





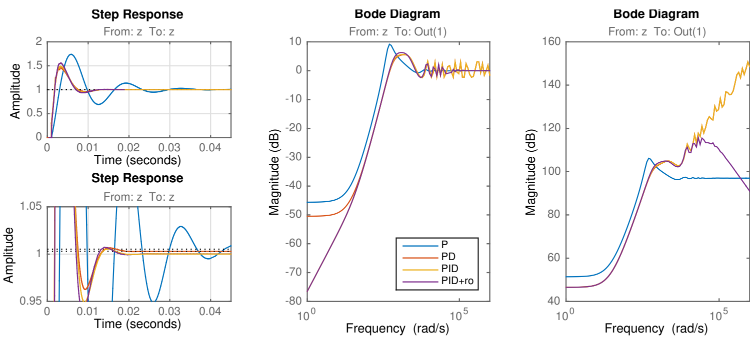
Sensitivities: These represent the response of the tracking error to reference inputs. They show that the error at low frequency is diminished for all controllers. The P controller, however, has a high sensitivity peak. The PD controller reduces this peak (again thanks to the better phase margin). The integral term of the PID controller forces the sensitivity to zero for low gain, assuring a zero error for step disturbances. Finally, adding the roll-off term allows to damp the peaks in the sensitivity at high frequencies by forcing the open-loop gain of the system to drop to zero at high frequencies. It is therefore always advised to use roll-off in a PID controller!

```
subplot(132);
bodemag(Sp0,Spd0,Spid0,Spidr0,logspace(0,6,101));
legend('P','PD','PID','PID+ro','location','southeast')
```



Control sensitivities These represent the control signal response to reference inputs We see that the P controller, having lower bandwidth, has a smaller response and peaks at lower frequency. The PD, PID and PID+rolloff controllers have higher bandwidth and higher peak. The rolloff ensures that the high-frequency controller response stays eventually decays, avoiding the injection of amplified high-frequency noise into the plant.

```
subplot(133);
bodemag(Cp0,Cpd0,Cpid0,Cpidr0,logspace(0,6,101));
```



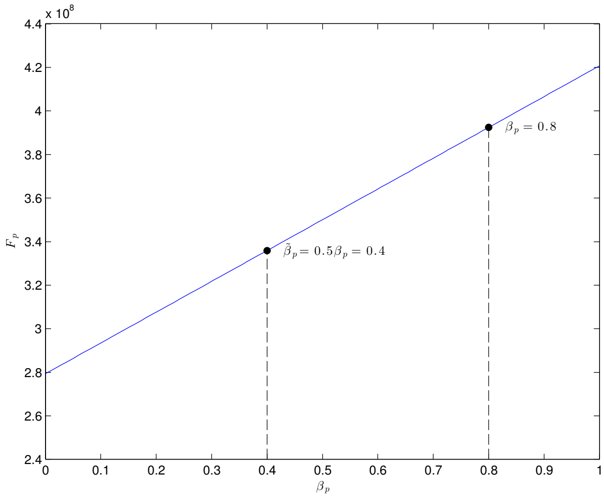


Figure 5: Dependency of F_p on β_p.