Control and operation of tokamaks
Exercise 2 - Plasma current control and plasma position determination Solutions

Lecturer: F. Felici
Instructors: D. Biek, R. Coosemans, S. Marchoni, P. Molina, F. Pastore
EPFL - SPC

February 2023

Ex 2.1: Plasma current control

Assume a simple model describing the Ohmic coils, plasma current, and current in the first (slowest) eigenmode of the vacuum vessel current distribution

$$L_{OH}\dot{I}_{OH} + M_{p,OH}\dot{I}_p + M_{p,OH}\dot{I}_e + R_{OH}I_{OH} = V_{OH} \tag{1}$$

$$M_{e,OH}\dot{I}_{OH} + M_{e,p}\dot{I}_p + L_e\dot{I}_e + R_eI_e = 0 \tag{2}$$

$$M_{p,OH}\dot{I}_{OH} + M_{p,p}\dot{I}_p + M_{p,e}\dot{I}_e + R_pI_p = 0 \tag{3}$$

a)

Describe the spatial distribution of current in the slowest vacuum vessel eigenmode.

To answer this question it is necessary to do an eigenmode decomposition of the vacuum vessel. The procedure to do this is briefly described in the slides and results for some eigenmodes (including the slowest eigenmode, λ_1) are presented. In Figure 1 we show the plot of the current distribution associated to the eigenvalue λ_1, the slowest one. As can be seen, the current distribution associated to this eigenmode is approximately constant over the vacuum vessel.

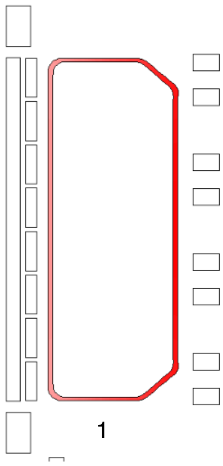

Figure 1: Vessel: first eigenmode .

b)

Calculate the loop voltage required to drive a plasma current of 300kA assuming $R = 3 \times 10^{-6}\Omega$.

To answer this question we directly used the definition of loop voltage, V_L (see the slides):

$$V_L := R_p I_p \,.$$

Substituting the known values $R_p = 3 \times 10^{-8}\,\Omega$ and $I_p = 3 \times 10^5\,A$ we obtain:

$$V_L = \left(3 \times 10^{-6}\right) \times \left(3 \times 10^5\right) \mathrm{V} = 9 \times 10^{-1}\,\mathrm{V}\,.$$

c)

Calculate the current ramp rate in the OH coils required to sustain this plasma current if $M_{p,OH} = 0.8748 \times 10^{-4}$.

To compute the current ramp rate in the OH coils we use the plasma circuit equation (see the slides):

$$0 = \boldsymbol{M}_{pa}\dot{\boldsymbol{I}}_a + \boldsymbol{M}_{pv}\dot{\boldsymbol{I}}_v + L_p\dot{I}_p + R_p I_p \,, \tag{4}$$

considering a steady state situation for the plasma and the vessel ($\dot{\boldsymbol{I}}_p = \dot{\boldsymbol{I}}_v = 0$). Under these conditions, (8) becomes:

$$0 = \boldsymbol{M}_{pa}\dot{\boldsymbol{I}}_a + R_p I_p \,.$$

If we consider only the Ohmic coils we have that $\boldsymbol{M}_{pa} := M_{p,OH}$, $\dot{\boldsymbol{I}}_a := I_{OH}$ and we can write:

$$0 = M_{p,OH}\dot{I}_{OH} + V_L \,,$$

where we used $V_L := R_p I_p$. This expression can easily be solved for $\dot{I}_{OH}$:

$$\dot{I}_{OH} = -\frac{V_L}{M_{p,OH}} = \frac{9 \times 10^{-1}}{0.8748 \times 10^{-4}} = 1.029 \times 10^4 \text{ A.s}^{-1} \,. \tag{5}$$

d)

Assume the OH coils have a current range of ±20kA. Compute the maximum flux swing in Wb if $M_{p,OH} = 0.8748 \times 10^{-4}$.

The maximum flux swing is defined as (see the slides):

$$\Delta\Psi_{OH} = \pm M_{p,OH}(I_{OH,\max} - I_{OH,\min}) \,. \tag{6}$$

For OH coils that have a current range of ±20kA, then the maximum flux swing is achieved for a current varying in this range:

$$\Delta\Psi_{OH} = M_{p,OH}(2\times10^4 - (-2\times10^4)) = 4\times10^4 \times M_{p,OH} = \left(4 \times 10^4\right)\times\left(0.8748 \times 10^{-4}\right) \text{ (Wb)} \,. \tag{7}$$

Which gives:

$$\Delta\Psi_{OH} = 3.5 \text{ Wb} \,.$$

e)

For how long can this plasma current be sustained at most?

The expression used to compute for how long can this plasma can be sustained at most is obtained from the plasma circuit equation (see the slides):

$$0 = \boldsymbol{M}_{pa}\dot{\boldsymbol{I}}_a + \boldsymbol{M}_{pv}\dot{\boldsymbol{I}}_v + L_p\dot{I}_p + R_p I_p \,, \tag{8}$$

considering a steady state situation for the plasma and the vessel ($\dot{\boldsymbol{I}}_p = \dot{\boldsymbol{I}}_v = 0$). Under these conditions, (8) becomes:

$$0 = \boldsymbol{M}_{pa}\dot{\boldsymbol{I}}_a + R_p I_p \,.$$

If we consider only the Ohmic coils we have that $\boldsymbol{M}_{pa} := M_{p,OH}$, $\dot{\boldsymbol{I}}_a := I_{OH}$ and we can write:

$$0 = M_{p,OH}\dot{I}_{OH} + V_L \,,$$

where we used $V_L := R_p I_p$. This expression can easily be solved for $\dot{I}_{OH}$:

$$\dot{I}_{OH} = -\frac{V_L}{M_{p,OH}} \, . \tag{9}$$

Considering that the Ohmic current, $\dot{I}_{OH}$, changes linearly with time from $I_{OH,\min}$ to $I_{OH,\max}$ in the time interval $[0, \Delta t]$, we can write:

$$I_{OH}(t) = I_{OH,\min} + \frac{t}{\Delta t}(I_{OH,\max} - I_{OH,\min}) \, .$$

This directly leads to:

$$\dot{I}_{OH} = \frac{I_{OH,\max} - I_{OH,\min}}{\Delta t} \, . \tag{10}$$

Substituting this into (9) we obtain the maximum time interval during which the plasma current can be sustained:

$$\dot{I}_{OH} \overset{(10)}{=} \frac{I_{OH,\max} - I_{OH,\min}}{\Delta t} \overset{(9)}{=} -\frac{V_L}{M_{p,OH}} \quad \Rightarrow \quad \Delta t = \frac{M_{p,OH}(I_{OH,\max} - I_{OH,\min})}{V_L} \overset{(6)}{=} \frac{\Delta\Psi_{OH}}{V_L} \, . \tag{11}$$

Substituting gives:

$$\Delta t = \frac{\Delta\Psi_{OH}}{V_L} = \frac{3.5}{9 \times 10^{-1}} \text{ s} = 3.89 \text{ s} \, .$$

Note that in reality a higher voltage is required for initial breakdown of the plasma and to ramp-up the current, which will ‘consume’ some flux before the actual flat-top phase.

Ex 2.2: Plasma position determination

Contents

- Run first the given file `Plasma_position_determination.mlx`
- a) least squares problem for current distribution at full x-grid
- b) least squares problem for current distribution at reduced h-grid
- c) solution of the least squares problem in (b) in the form $I_h = Hb$
- d) Write a linear estimator for the plasma current from the measurements such that $I_p = H_{I_p} \cdot b$
- e) Use the expressions for r and z to compute an estimate of r and z based on the measurements of magnetic probes and flux loops
- f) Write a estimator for the plasma rI_p and zI_p i.e. the product of radial/vertical position and plasma current
- g) Plot the outputs of the linear estimators and compare to the known values

Outputs of the first run are shown in below including the matlab code.

Load the data and get plasma grid

```
clear;
load ex2_data.mat;
```

Define plasma distribution on the x grid

```
Ix = zeros(L.nzx,L.nrx); Ix(2:end-1,2:end-1) = LY.Iy;
% true current centroid
r0 = sum(L.rrx(:).*Ix(:))/sum(Ix(:));
z0 = sum(L.zzx(:).*Ix(:))/sum(Ix(:));
Ip0=sum(Ix(:)); %calculating the total current
% plot distribution and measurement locations

figure(1); clf;
imagesc(L.rx,L.zx,Ix); axis tight; axis xy equal; %plots current distribution
hold on;
plot(L.G.rl,L.G.zl);
plot(L.G.rv,L.G.zv);
```

```
plot(r0,z0,'wo');% current centroid
colorbar;
legend('limiter','vessel','');
title('Example of a current distribution');
```

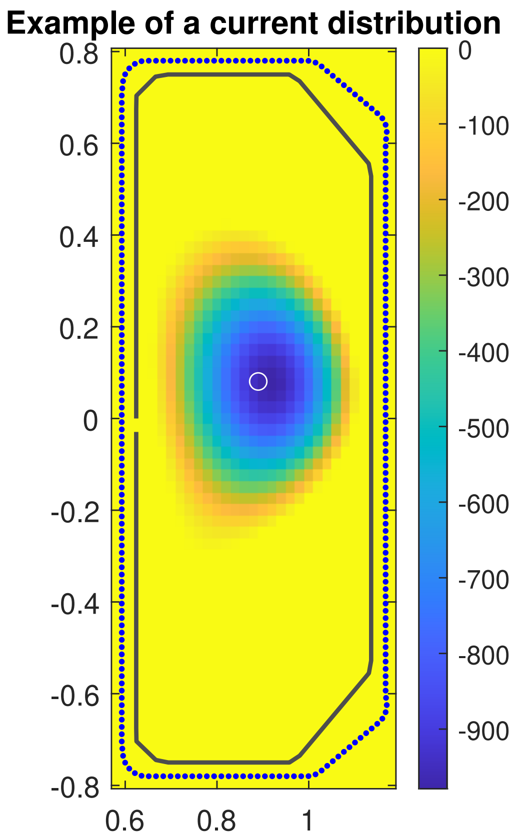

Figure 2: Current distribution of plasma current in the x-grid

Compute expected measurements for given plasma distribution

```
Ia = LY.Ia; % circuit currents

Bm  = L.G.Bmx*Ix(:) + L.G.Bma*Ia; % magnetic probes
Ff  = L.G.Mfx*Ix(:) + L.G.Mfa*Ia; % flux loops
```

Add noise and turn into time-dependent signal

In reality, it is a hard to clear measurements during operation. Usually, the measurements also contain noise. Hence, we add time-dependent noise to the magnetic fields and flux measurements.

```
eBm = 10e-3; % 10mT noise
eFf = 10e-3; % 10mWb noise
eIa = 0;     % assume no measurement error on currents

% time grid
time = 0:1e-3:0.1;
nt = numel(time);
% time evolution
Bm_meas = Bm + eBm*randn(numel(Bm),nt);
Ff_meas = Ff + eFf*randn(numel(Ff),nt);
Ia_meas = Ia + eIa*randn(numel(Ia),nt);
% to perform mathematical operation on inputs
% which do not have the same dimension
```

Plots

```
figure(2); clf;
% magnetic probes time evolution
subplot(221)
plot(time,Bm_meas);
title('B probe [T]'); xlabel('time [s]')
% flux loops time evolution
subplot(223)
plot(time,Ff_meas);
title('\psi_{fluxloop} [Wb]'); xlabel('time [s]')

% measurements per magnetic probe
subplot(222)
plot(1:numel(Bm),Bm_meas,'.',1:numel(Bm),Bm,'or')
title('B probes [T]')
xlabel('probe number');

% measurements per flux loop
subplot(224)
plot(1:numel(Ff),Ff_meas,'.',1:numel(Ff),Ff,'or');
title('flux loops [Wb]')
xlabel('flux loop');
```

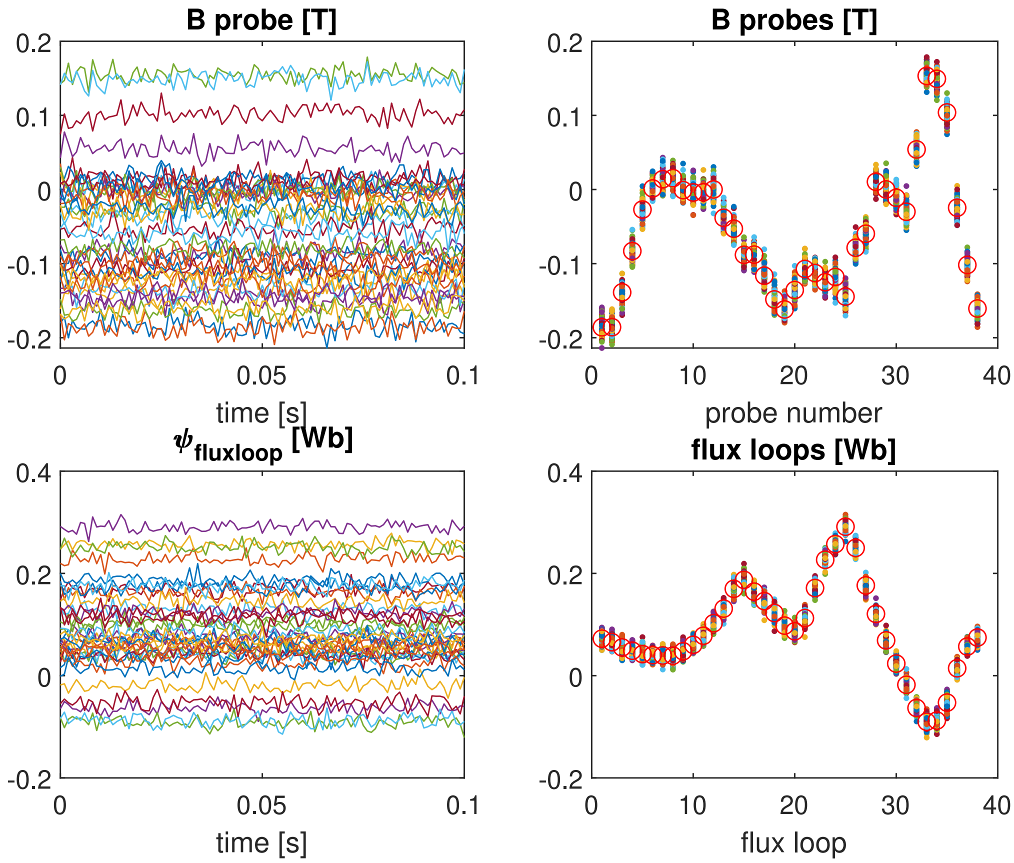

Figure 3: Measurements of magnetic probes and flux loops including the time-dependent noise

```
%Positons of the magentic probes and flux loops
figure(3); clf;
subplot(121)
imagesc(L.rx,L.zx,Ix);  axis tight; axis xy equal; colorbar ('westoutside');
hold on;
plot(L.G.rl,L.G.zl);
plot(L.G.rv,L.G.zv);
plot(L.G.rf,L.G.zf,'kx');
text(L.G.rf+0.02,L.G.zf,L.G.dimf);
legend('limiter','vessel','Flux loop');
title('flux loop in TCV');

subplot(122)
imagesc(L.rx,L.zx,Ix); axis tight; axis xy equal; colorbar('westoutside');
hold on;
plot(L.G.rl,L.G.zl);
```

```
plot(L.G.rv,L.G.zv);
plot(L.G.rm,L.G.zm,'kx')
text(L.G.rm+0.02,L.G.zm,L.G.dimm);
legend('limiter','vessel','B probe');
title('magnetic probes in TCV');
```

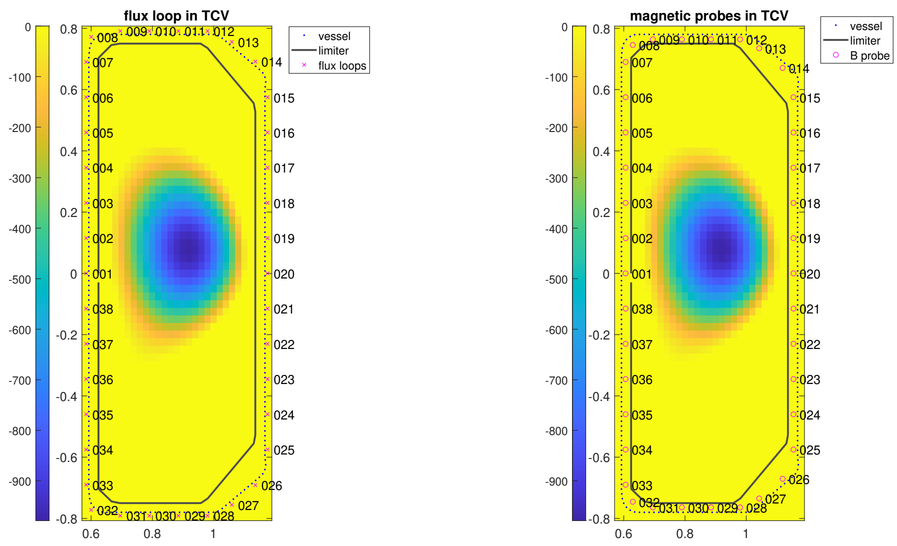

Figure 4: Current distributions in TCV (Right: Locations of magnetic probes and left: locations of flux loops

Define new grid boundaries with fewer elements (2x3)

In the following, we demonstrate how to perform the transformation into a smaller grid. As a first step the new grid is defined.

```
nr = 2; nz = 3; nh = nr*nz;
%number of filaments in radial and vertical direction;
%total number of filaments
rhgrid = linspace(min(L.G.rx),max(L.G.rx),nr+1);
zhgrid = linspace(min(L.G.zx),max(L.G.zx),nz+1);
```

Build transformation matrix T_{xh} from full 'x' grid to smaller grid 'h'

In this section, we compute a matrix T_{xh}, such that $I_h = T_{xh} \cdot I_x$ and $I_x = {T_{xh}}^\top \cdot I_h$. This matrix allows to convert current distributions on one grid to another. Additionally, we plot the orginal grid and the new defined gird and an example of current distribution on the h-grid (Note: the example is physically unlikely and is for illustration purpose).

```
Txh = zeros(L.nx,nh);
[rrx,zzx] = meshgrid(L.G.rx,L.G.zx);
for ir = 1:nr
    for iz = 1:nz
        ii = (ir-1)*(nz) + iz; % element index
        txy = (rhgrid(ir) <= rrx) & (rrx < rhgrid(ir+1))...
            & (zhgrid(iz) <= zzx) & (zzx < zhgrid(iz+1)) ;
        Txh(:,ii) = txy(:);
    end
end
% calculate mean r,z of each new grid point
rh = sum(Txh.*rrx(:),1)./sum(Txh,1);
zh = sum(Txh.*zzx(:),1)./sum(Txh,1);

% plot
figure(4); clf;
subplot(121)
p=plot(rrx,zzx,'xb',rh,zh,'or'); axis equal tight
[rhh,zhh] = meshgrid(rhgrid,zhgrid);
hold on; p1=plot(rhh,zhh,'+k');
legend([p(1) p(29) p1(1)],'x grid (filaments)',
'h grid midpoints','h grid boundary','location','south')
title('x and h grid definitions')
% example of a distribution at the smaller h-grid
Ih = zeros(nh,0); Ih([1,3,6],1) = 1e3;
Ix = Txh*Ih;
subplot(122)
pcolor(L.G.rx,L.G.zx,reshape(Ix,L.nzx,L.nrx));colorbar;
title('Example of an (unlikely) current distribution on the h grid')
```

a)

Write a least-squares problem to determine the current distribution $\boldsymbol{Ix}$ *on the full* $\boldsymbol{x}$ *grid from the measurements of magnetic probes and flux loops, in the form* $\min ||AI_x - b||_2$*. Can this least-squares problem be solved? If not, why not?*

The unknown parameters are the current distribution that we would like to estimate, where these are 1820 unknowns. The available measurements are 38 magnetic probe

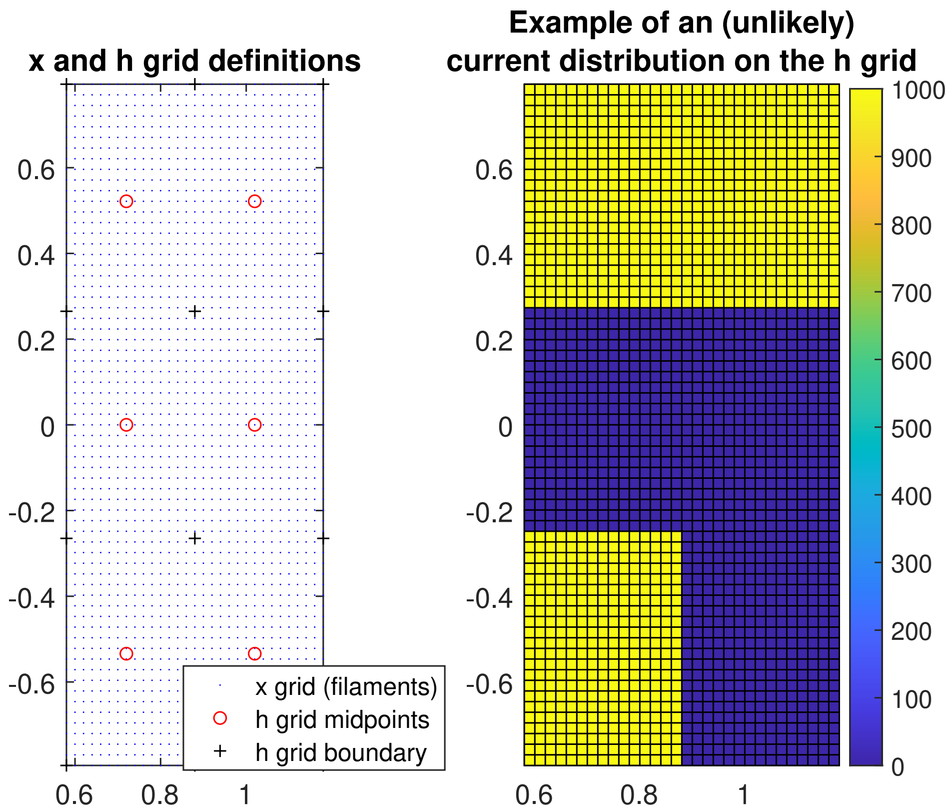

Figure 5: Right: Unlikely scenario of the current distribution on the h-grid; Left: The x-grid and the h-grid

measurements and 61 flux loops. From these 99 measurements we cannot uniquely derive the 1820 unknowns and we cannot formulate a least squares problem that can be solved.

b)

Write a least-squares problem to determine the current distribution `Ih` *on the reduced grid from the measurements. What is A and b? Can this problem be solved? If not, why not?*

On the reduced grid, we can solve the leat squares, since we do not have too many unknows due to the transformation. We will formulate the least-squares problem as min $||AI_h - b||_2$ with I_h being the current per filament. Also, we need to define the weighting matrix which includes the effects of the measurement errors. b and A are defined as on slide 26.

```
% Compute first the measurement matrices for the new grid
Mfh = L.G.Mfx*Txh; % flux loops
```

```
Bmh = L.G.Bmx*Txh;
% Formulate now the least-squares problem

W = diag([ones(L.G.nm,1)*1/eBm;ones(L.G.nf,1)*1/eFf]); % weight matrix
b = W*([Bm_meas;Ff_meas] - [L.G.Bma;L.G.Mfa]*Ia_meas); % measurements minus coil c
% (time evolution or alternatively data at single time step)
A = W*[Bmh;Mfh]; % matrices
% weighting matrix
```

c)

Write the solution of least-squares problem in (b) in the form $I_h = Hb$. How can be H determined from A?

We use linear estimators H such that quantity $= H$*measurements. Using the result from b), we can define linear estimators $H = (W \cdot A)^+$. Also, we need to weigh the measurements. Note: We use the pseudoinverse, as we are dealing with non-square matrices.

```
Q = pinv(A)*W;
Ih = Q*b; % solution for Ih
Ixhat = Txh*Ih; % current in x grid filaments for this solution

figure(5); clf;
imagesc(L.rx,L.zx,reshape(Ixhat(:,1),L.nzx,L.nrx)); axis xy equal; axis tight;
hold on;
plot(L.G.rl,L.G.zl);
plot(L.G.rv,L.G.zv);
colorbar;
```

d)

Recall that the plasma current can be determined from the current distribution as $I_p = \sum_{j=1}^{n_x} I_{x,j}$. Write a linear estimator for the plasma current from the measurements such that $I_p = H_{I_p} \cdot b$

The plasma current is the sum of the currents in all x grid filaments.

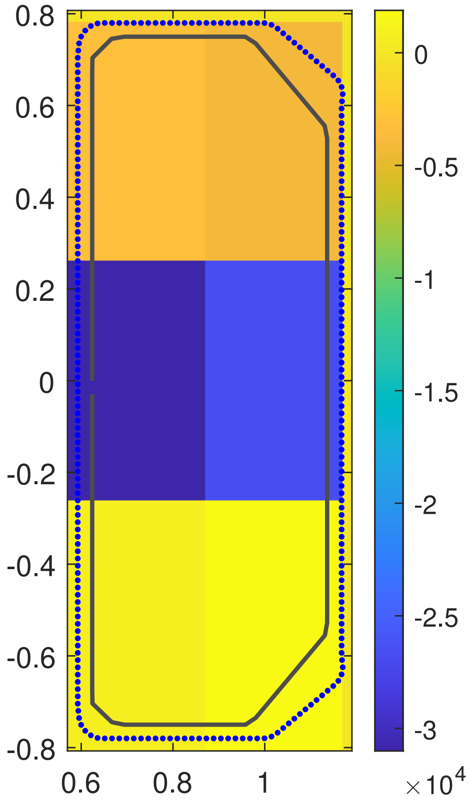

Figure 6: Current distribution for the h-grid

```
y = [Bm_meas;Ff_meas;Ia_meas]; % measurements
Ahy = [Q,-Q*[L.G.Bma;L.G.Mfa]]; % linear estimator fir Ih
HIp = ones(1,size(Txh,1)) * Txh * Ahy;
Ip = HIp*y;
```

e)

We can define an estimator for the plasma radial and vertical position, by assuming the position corresponds to the centroid of the current distribution:

$$\hat{r} = \frac{\sum_j^{n_x} r_{x,j} I_{x,j}}{\sum_i^{n_x} I_{x,i}} \tag{12}$$

$$\hat{z} = \frac{\sum_j^{n_x} z_{x,j} I_{x,j}}{\sum_i^{n_x} I_{x,i}} \tag{13}$$

Use these expressions to compute an estimate of r and z based on the measurements of magnetic probes and flux loops. Is this a linear estimator?

Equations 4 and 5 from the exercise sheet can be read as $\hat{r} = \frac{rI_p}{I_p}$ and $\hat{z} = \frac{zI_p}{I_p}$. We can compute this as follows:

```
% for rIp=HrIp*meas
HrIp = ones(1,size(Txh,1)) * (rrx(:).*Txh) * Ahy;
% for zIp=HzIp*meas
HzIp = ones(1,size(Txh,1)) * (zzx(:).*Txh) * Ahy;
rhat=(HrIp*y)./(HIp*y);
zhat=(HzIp*y)./(HIp*y);
```

Looking at $\hat{r}$ and $\hat{z}$, we can clearly observe that it is not a linear estimator.

f)

Write a estimator for the plasma rI_p and zI_p i.e. the product of radial/vertical position and plasma current. Is this linear?

```
% for rIp=HrIp*b
HrIp = ones(1,size(Txh,1)) * (rrx(:).*Txh) * Ahy;
% for zIp=HzIp*b
HzIp = ones(1,size(Txh,1)) * (zzx(:).*Txh) * Ahy;
```

This is a linear estimator.

g)

*P*lot the outputs of the linear estimators and compare to the known values of $r_0 I_{p0}$, $z_0 I_{p0}$, I_{p0}. Attempt to improve the estimate of $z_0 I_{p0}$ and I_{p0} by changing the number of degrees of freedom in `Ih` and explain the results.

First we compute the estimator for all three quantities and then compute these quantities.

```
Ahy = [HIp;HrIp;HzIp];
q = Ahy*y;
Ip = q(1,:);
```

```
rIp = q(2,:);
zIp = q(3,:);
```

We plot now the results (see Figure 7):

```
figure(6); clf;
subplot(311);
plot(time,rIp,'b',time,r0*Ip0*ones(numel(time),1),'r')
title('rIp estimate'); xlabel('time [s]');
subplot(312);
plot(time,zIp,'b',time,z0*Ip0*ones(numel(time),1),'r')
title('zIp estimate'); xlabel('time [s]');
subplot(313);
plot(time,Ip,'b',time,Ip0*ones(numel(time),1),'r')
title('Ip estimate'); xlabel('time [s]');
```

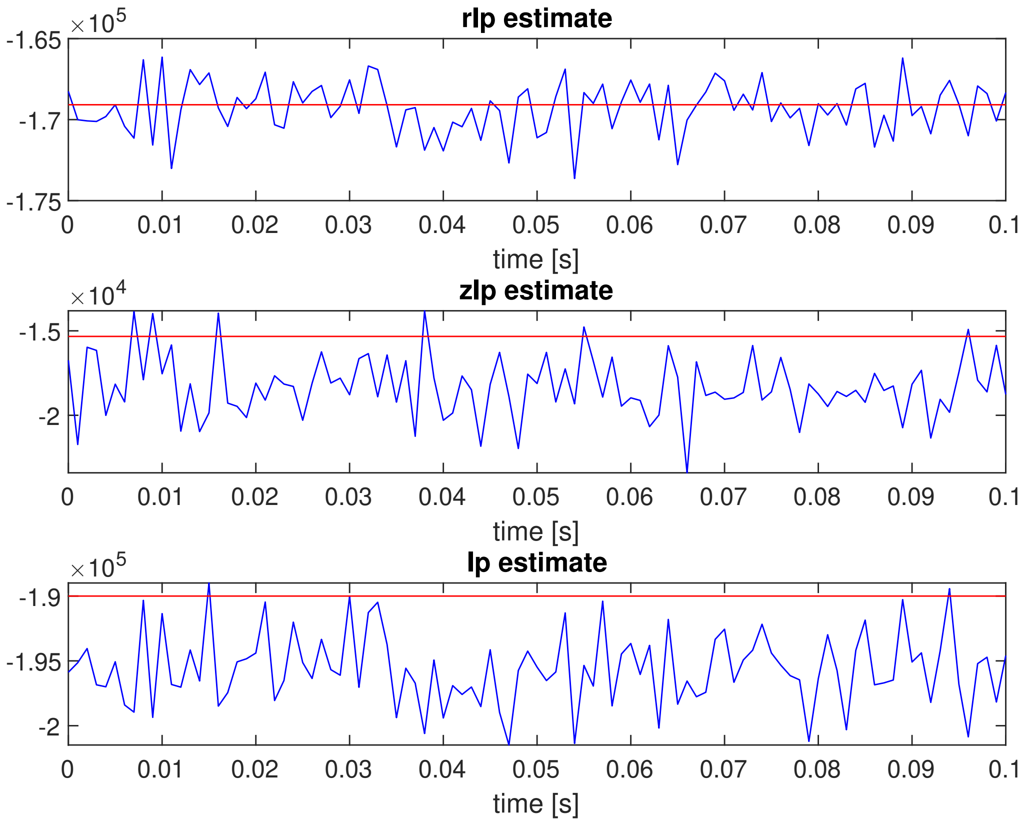

Figure 7: Estimate of rI_p, zI_p, I_p compared to the actual value

These results show that we can estimate the product of radial position and plasma current quite accurately, whereas we have an small offset in the vertical position and plasma current estimate. This can be improved by using more filaments in vertical and radial direction. We will use here 5 instead of 3 filaments for the vertical position and 3 filaments instead of two filaments for the radial position. Moreover, we use an auxiliary function to define $\mathbf{T_{xh}}$ for arbitrary nr, nz. This is used also for the h).

```
 % define new grid boundaries with more elements (3x5)
nr = 3; nz = 5; nh = nr*nz;
rhgrid = linspace(min(L.G.rx),max(L.G.rx),nr+1);
zhgrid = linspace(min(L.G.zx),max(L.G.zx),nz+1);
% rebuild transformation matrix from full 'x' grid to smaller grid 'h'
% use auxiliary functions to compute updated Txh ajd estimators
nr = 3; nz = 5;
Txh = define_Txh(L,nr,nz);
[Ahy,HIp,HrIp,HzIp] = define_estimators(L,W,Txh);

% calc new results
Ip = HIp*y;
rIp = HrIp*y;
zIp = HzIp*y;
% plot now the results:
figure(7); clf;
subplot(311);
plot(time,rIp,'b',time,r0*Ip0*ones(numel(time),1),'r')
title('rIp estimate'); xlabel('time [s]');
subplot(312);
plot(time,zIp,'b',time,z0*Ip0*ones(numel(time),1),'r')
title('zIp estimate'); xlabel('time [s]');
subplot(313);
plot(time,Ip,'b',time,Ip0*ones(numel(time),1),'r')
title('Ip estimate'); xlabel('time [s]');
```

Auxiliary function to define $\mathbf{T_{xh}}$ for arbitrary nr, nz

```
function Txh = define_Txh(L,nr,nz)
nh = nr*nz; % define new grid boundaries with more elements (2x5)
rhgrid = linspace(min(L.G.rx),max(L.G.rx),nr+1);
zhgrid = linspace(min(L.G.zx),max(L.G.zx),nz+1);
% rebuild transformation matrix from full 'x' grid to smaller grid 'h'
Txh = zeros(L.nx,nh);
[rrx,zzx] = meshgrid(L.G.rx,L.G.zx); % not affected by the h-grid
for ir = 1:nr
    for iz = 1:nz
        ii = (ir-1)*(nz) + iz; % element index
```

```
        txy = (rhgrid(ir) <= rrx) & (rrx < rhgrid(ir+1))...
            & (zhgrid(iz) <= zzx) & (zzx < zhgrid(iz+1)) ;
        Txh(:,ii) = txy(:);
    end
end
end
```

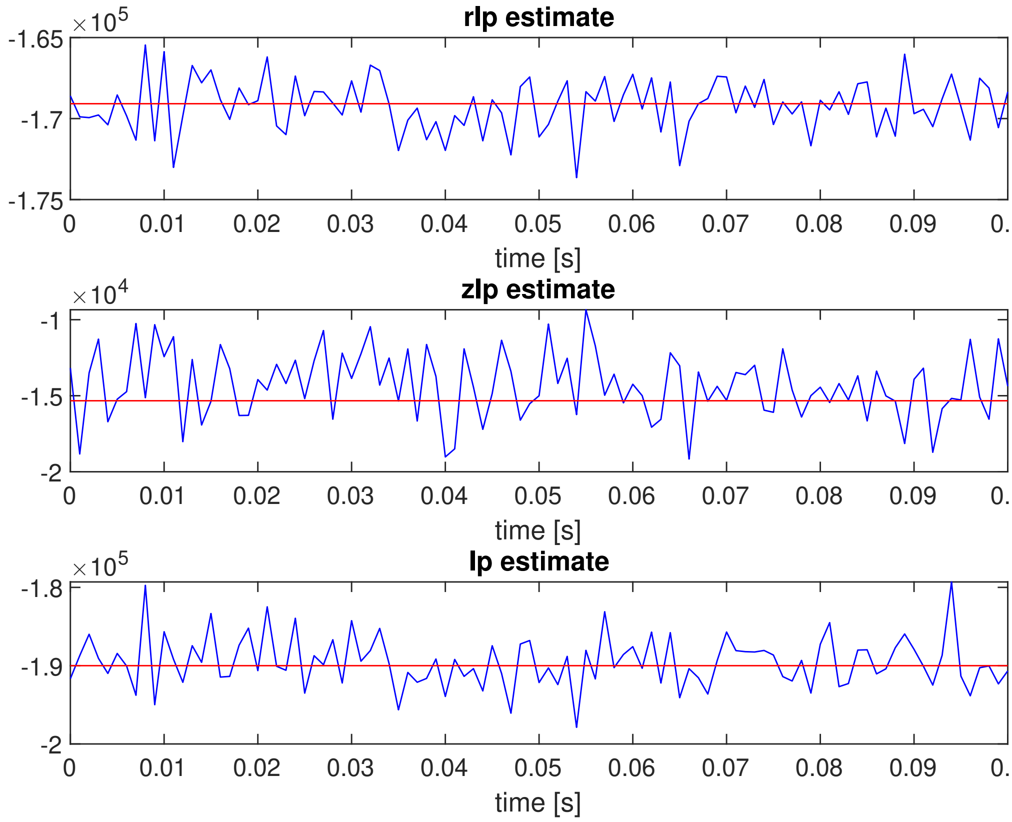

Figure 8: Estimate of rI_p, zI_p, I_p with 5 filaments in z-direction and 3 filaments in r-direction compared to the actual value

We now have also a close estimate of the plasma current and the product of vertical position and plasma current.

h)

Plot the mean and standard deviation of I_p, zI_p, rI_p estimates for various values of nr, nz (and $nh = nz * nr$). Does the estimator quality increase, decrease or stay the same when increasing nh? Explain why.

As a next step we investigate how does the estimator quality behaves for increasing nh and caclucate for each case the standard deviation.

```
 %% Scan of nr,nz and computing mean and standard deviation
%% Scan of nr,nz and computing mean and standard deviation
nrgrid = [1 2 3 4 10 20];
nzgrid = [2 4 6 8 20 40];
clear result
for ii=1:numel(nrgrid)
  nr= nrgrid(ii);
  nz= nzgrid(ii);
  Txh = define_Txh(L,nr,nz);
  [Ahy,HIp,HrIp,HzIp] = define_estimators(L,W,Txh);

  rIp = HrIp * y;
  zIp = HzIp * y;
  Ip  = HIp * y;
  Ix  = Txh*Ahy*y;
  % store result
  result(ii).rIp_mean = mean(rIp);
  result(ii).rIp_std  = std(rIp);
  result(ii).zIp_mean = mean(zIp);
  result(ii).zIp_std  = std(zIp);
  result(ii).Ip_mean  = mean(Ip);
  result(ii).Ip_std   = std(Ip);
end
% plot
figure(8); clf;
nres = numel(result);
subplot(311)
errorbar(1:numel(result),[result.rIp_mean],[result.rIp_std],'x'); hold on;
plot(1:numel(result),r0*Ip0*ones(nres,1),'r--');
set(gca,'XTick',[]);
ylabel('rIp');

subplot(312)
errorbar(1:numel(result),[result.zIp_mean],[result.zIp_std],'x'); hold on;
plot(1:numel(result),z0*Ip0*ones(nres,1),'r--');
set(gca,'XTick',[]);
ylabel('zIp');

subplot(313);
errorbar(1:numel(result),[result.Ip_mean],[result.Ip_std],'x'); hold on;
plot(1:numel(result),Ip0*ones(nres,1),'r--');
```

```
set(gca,'XTick',(1:nres));
set(gca,'XTickLabel',strcat('nr = ',num2str(nrgrid'),'  nz = ',num2str(nzgrid')))
set(gca,'XTickLabelRotation',45)
ylabel('Ip');
```

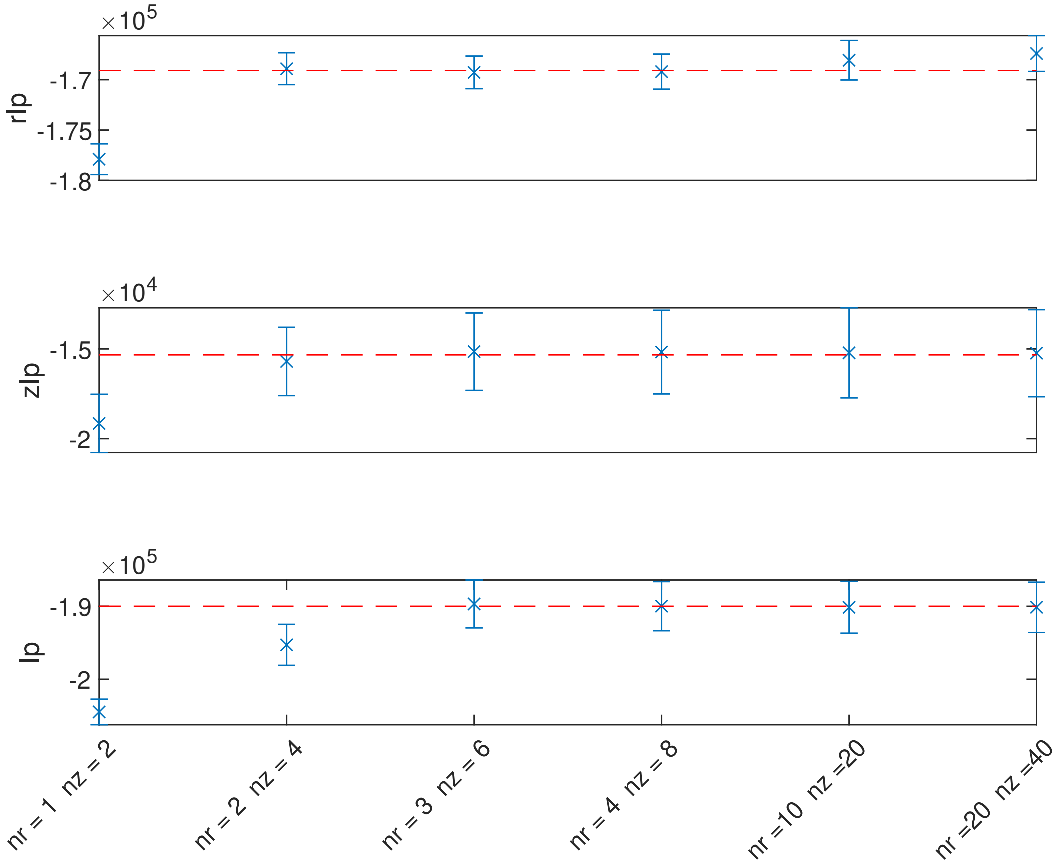

Figure 9: Standard deviation for the estimate of rI_p, zI_p, I_p for various nh

The quality initially increases (e.g. for nh=3 increasing to nh=18 because it initially needs enough degrees of freedom to represent the current distribution. When increasing nr,nz further, the mean and standard deviations stay roughly constant. This is because, while the problem of determining Ix is ill-conditioned, its average sum(rrx.*Ix) and sum(zzx.*(Ix)) stays the correct because it is well constrained by the measurements.

i)

Now assume that the circuit current measurements `Ia_meas` is also affected by measurement noise of standard deviation `eIa=100[A]`. Check the quality of your estimator and compare it to the non-noisy case. Reformulate the least squares problem to jointly estimate *the plasma current distribution and circuit currents from the measurements and compare the estimation results.*

```
    eIa = 100;    % 100A measurement error on currents

% new measurements and weights
Ia_meas = Ia + eIa*randn(numel(Ia),nt);
y2 = [Bm_meas;Ff_meas;Ia_meas];
W2 = diag([ones(L.G.nm,1)*1/eBm;ones(L.G.nf,1)*1/eFf;ones(L.G.na,1)*1/eIa]);

% take a reasonable nr,nz and recompute estimators
nr = 3; nz = 8; nh = nr*nz;
Txh = define_Txh(L,nr,nz);
% regular estimators
[Ahy,HIp,HrIp,HzIp] = define_estimators(L,W,Txh);
rIp = HrIp * y2;
zIp = HzIp * y2;
Ip  = HIp  * y2;

% new estimators - new least squares problem with [Ih;Ia] as unknown
Mfh = L.G.Mfx*Txh;
Bmh = L.G.Bmx*Txh;
% assemble A matrix
Ah = [Bmh;Mfh;zeros(L.G.na,nh)]; % Ih part
Aa = [L.G.Bma;L.G.Mfa;eye(L.G.na)]; % Ia part
A = W2*[Ah,Aa];
Q = pinv(A)*W2;
Ahy = Q(1:nh,:);     % estimator for Ih
Aay = Q(nh+1:end,:); % estimator for Ia

HIp = ones(1,size(Txh,1)) * Txh * Ahy;
HrIp = ones(1,size(Txh,1)) * (rrx(:).*Txh) * Ahy; % for rIp=HrIp*b
HzIp = ones(1,size(Txh,1)) * (zzx(:).*Txh) * Ahy; % for zIp=HzIp*b
rIp2     = HrIp * y2;
zIp2     = HzIp * y2;
Ip2      = HIp * y2;
Ia_est2  = Aay * y2; % estimate Ia

figure(9); clf;
subplot(411)
plot(time,rIp,'display',sprintf('old estimator[kA]:
%2.2f+/-%2.2f', mean(rIp/1e3),std(rIp/1e3))); hold on;
```

```
plot(time,rIp2,'k--','display',sprintf('new estimator[kA]:
%2.2f+/-%2.2f', mean(rIp2/1e3),std(rIp2/1e3)))
plot(time,r0*Ip0*ones(nt,1),'r--','display',sprintf('true value[kAm]:
%2.2f',r0*Ip0/1e3));
legend('show')

subplot(412)
plot(time,zIp,'display',sprintf('old estimator[kA]:
%2.2f+/-%2.2f', mean(zIp/1e3),std(zIp/1e3))); hold on;
plot(time,zIp2,'k--','display',sprintf('new estimator[kA]:
%2.2f+/-%2.2f', mean(zIp2/1e3),std(zIp2/1e3)))
plot(time,z0*Ip0*ones(nt,1),'r--','display',sprintf('true value[kA]:
%2.2f',z0*Ip0/1e3));
legend('show')

subplot(413)
plot(time,Ip,'display',sprintf('old estimator[kA]:
%2.2f+/-%2.2f', mean(Ip/1e3),std(Ip/1e3))); hold on;
plot(time,Ip2,'k--','display',sprintf('new estimator[kA]:
%2.2f+/-%2.2f', mean(Ip2/1e3),std(Ip2/1e3)))
plot(time,Ip0*ones(nt,1),'r--','display',sprintf('true value[kA]:
%2.2f',Ip0/1e3));
legend('show')

subplot(414)
ii=1; % plot one coil only
plot(time,Ia_meas(ii,:),'display',sprintf('measurement[kA]:
%2.2f+/-%2.2f', mean(Ip/1e3),std(Ip/1e3))); hold on;
plot(time,Ia_est2(ii,:),'k--','display',sprintf('new estimator[kA]:
%2.2f+/-%2.2f', mean(Ip2/1e3),std(Ip2/1e3)))
plot(time,Ia(ii,:)*ones(nt,1),'r--','display',sprintf('true value[kA]:
%2.2f',Ip0/1e3));
title(sprintf('coil %s current',L.G.dima{ii}))
legend('show')
```

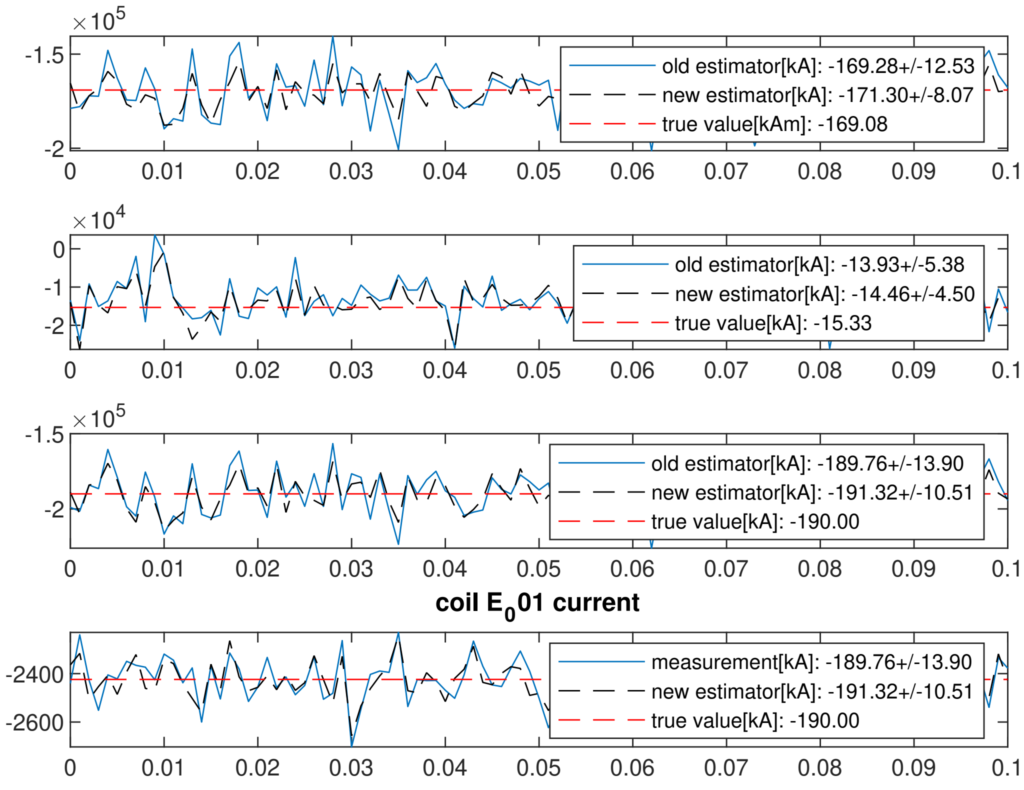

Figure 10: Old and new estimates of I_p, zI_p, rI_p and the estimate for the coil current after the measurement noise. All grpahs are plotted on a time axis