Contents

Offline magnetic measurements analysis

Christian Grech 18/10/2017

```
clc;
clear all;
close all;
```

Input file

```
filename = 'PBMD2_calib.mat';
n =500;                                   % downsampling graphs
rate = 1/0.5e6;                  % time resolution
freq = 0.5e6;
load(filename);                  % load file
```

Separate data

Degaussing cycle is removed for faster loading

```
time =  var(1.2e6:end,1);
current = var(1.2e6:end, 2);
coil5 = -var(1.2e6:end, 3);
NMR1_hi = var(1.2e6:end,4);
NMR2_hi = var(1.2e6:end,5);
```

Filter noisy current signal

```
 windowWidth = 100;
% % Moving average filter in both directions applied to Im
 kernel = ones(windowWidth,1) / windowWidth;
 DCCT = filtfilt(kernel, 1, current)*100;
```

Define flat top and flat bottom points

```
 integral = cumtrapz(time, coil5);
```

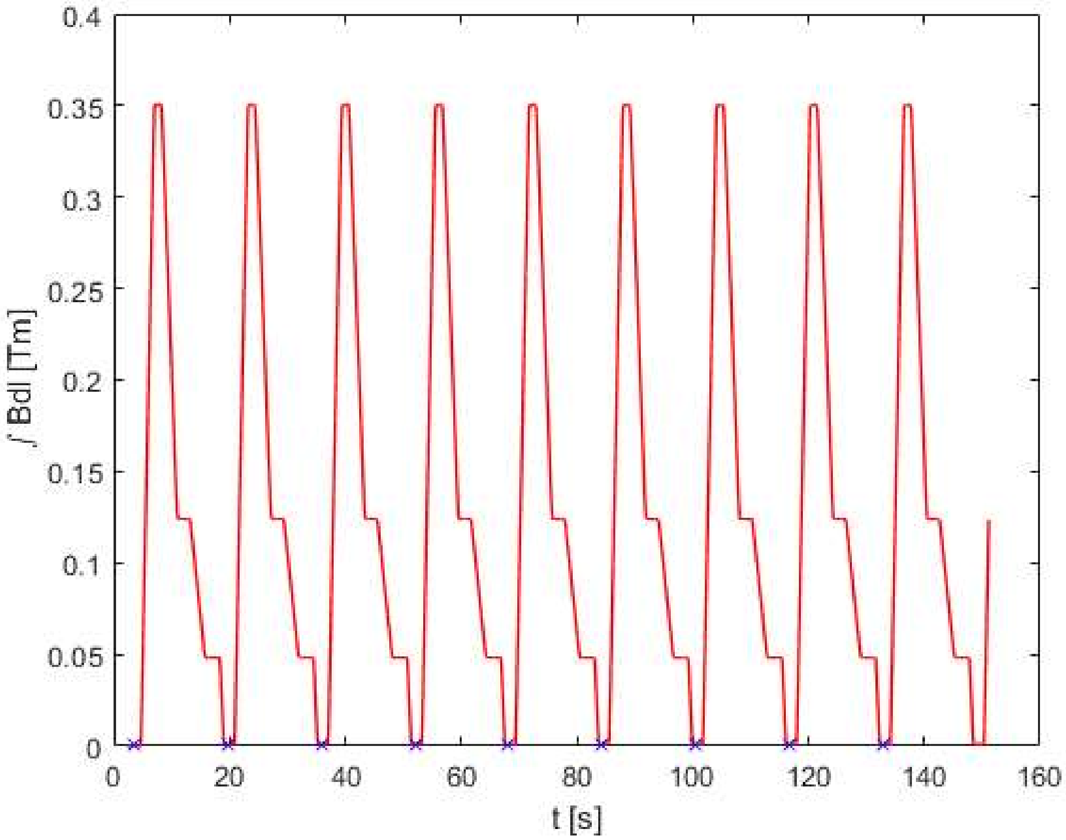
```
    % raw flux
% Flat bottom indexes
dt = 8.1e6;
% Flat bottom indexes
fb1 = 1*freq; fb2 = fb1+dt; fb3 = fb2+dt; fb4 = fb3+dt; fb5 = fb4+dt; fb6 =fb5+dt; fb7=f
b6+dt; fb8 =  fb7+dt; fb9 =  fb8+dt; fb10 =  fb9+dt;
fb = [fb1; fb2; fb3; fb4; fb5; fb6; fb7; fb8; fb9; fb10];   % Concatenate all flat top i
ndexes

% Index parameters for drift correction
indexMarker = [4.4e6 4.6e6; 4.4e6 4.6e6;  4.4e6 4.6e6; 1 dt; 1 dt; 1 dt; 1 dt; 1 dt; 1 d
t; 1 dt]';
coil_div = [coil5(fb1:fb2) coil5(fb2:fb3) coil5(fb3:fb4) coil5(fb4:fb5)  coil5(fb5:fb6)
coil5(fb6:fb7) coil5(fb7:fb8) coil5(fb8:fb9) coil5(fb9:fb10) ];
time_div = [time(fb1:fb2) time(fb2:fb3) time(fb3:fb4) time(fb4:fb5)  time(fb5:fb6) time(
fb6:fb7) time(fb7:fb8) time(fb8:fb9) time(fb9:fb10) ];
indicator = zeros(1, length(fb));
% plot(integral(1:1:end)); hold on; plot(fb, indicator, 'rx');
```

Drift correction

```
        for i = 1:9
                        rawFlux(:,i) = cumtrapz(time_div(:,i), coil_div(:,i));   % raw f
lux calculated for all cycles
        end
        timeCoil = time_div;

         for i=1:9
                    drift(i)=(mean(rawFlux(indexMarker(2,i)-500:indexMarker(2,i),i))-mea
n(rawFlux(indexMarker(1,i):indexMarker(1,i)+500,i)))...
                        ./(timeCoil(indexMarker(2,i),i)-timeCoil(indexMarker(1,i),i));
            % drift calculated for all cycles
         end

        drift(1:3) = mean(drift(4:end));
        coil5_drift_corrected(1:fb(1)-1) =  coil5(1:fb(1)-1)-mean(coil5(1:0.5e6));
         coil5_drift_corrected(fb(end)+1:length(time)) =  coil5(fb(end)+1:length(time));

        for i = 1:9
        coil5_drift_corrected(fb(i):(fb(i)+dt-1)) = coil5(fb(i):(fb(i)+dt-1))-drift(i);
        end;
```

BdL Calculation

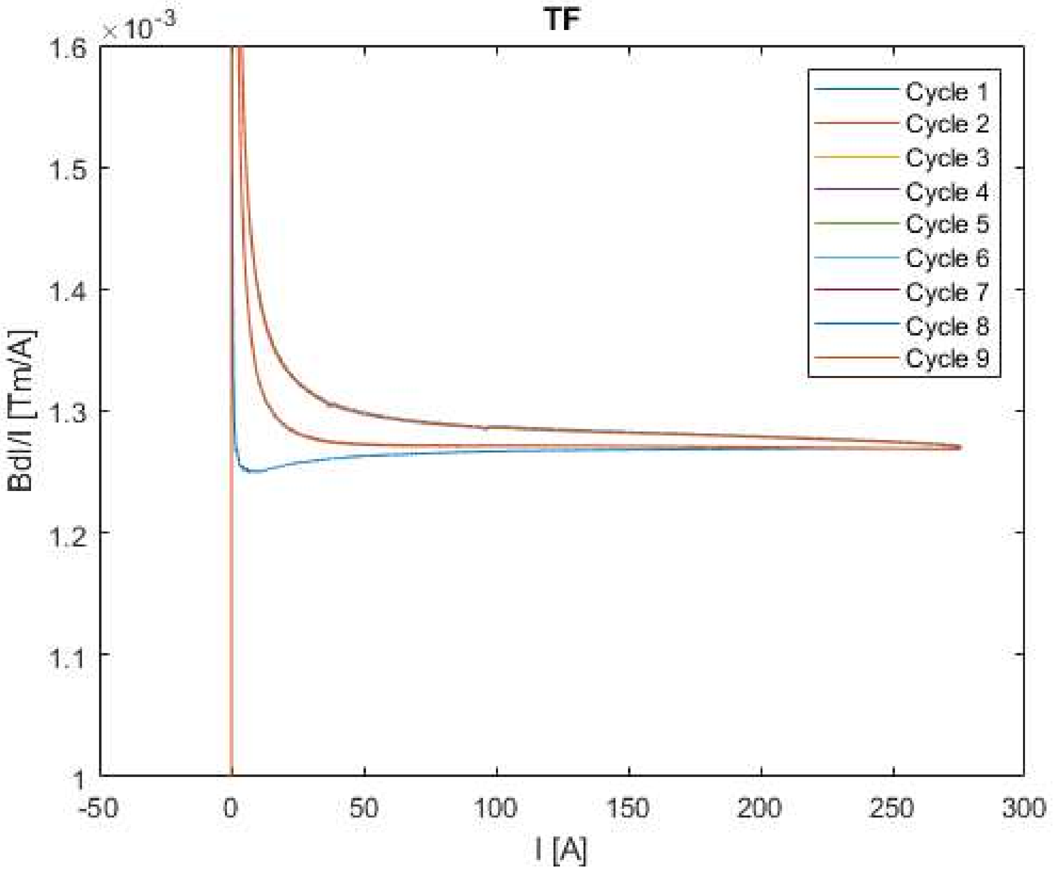
```
          phi_int = cumtrapz(time,coil5_drift_corrected);         % Integrated flux
         Bdl_int = phi_int/2.8415;                                            % Bdl calc
ulation
         Bdl_initial = mean(Bdl_int(4e5:6e5));                           % Initial Bdl
         Bdl = (Bdl_int-Bdl_initial+0.0303e-3);                  % Degaussing does not r
educe Bdl to zero but to 0.03mT


          figure; plot(time(1:n:end), Bdl(1:n:end),'r', 'LineWidth',1);   ylabel('\int B
dl [Tm]'); xlabel('t [s]');
         hold on;
         for i =1:9
         plot(time(fb(i)), Bdl(fb(i)), 'bx');
         end;
```



TF plot

```
n=1000;
  list=["Cycle 1", "Cycle 2", "Cycle 3", "Cycle 4", "Cycle 5", "Cycle 6", "Cycle 7", "Cy
cle 8", "Cycle 9", "Cycle 10", "Cycle 11", "Cycle 12", "Cycle 13"];
  figure;
 for i =1:9
 plot(DCCT(fb(i):n:fb(i+1))', Bdl(fb(i):n:fb(i+1))./DCCT(fb(i):n:fb(i+1))');   ylabel('B
dl/I [Tm/A]'); xlabel('I [A]'); title('TF'); ylim([1e-3 1.6e-3]);
 hold on;
 end;
 legend(list)
```

```
Warning: Ignoring extra legend entries.
```



Integration constant calculation

Find the BdI at the NMR trigger point

```
for i = 1:9
        [i_NMR_OP_high(i), C3(i)] = find(NMR1_hi(fb(i):(fb(i)+30e5))<1e-3,1);
        [i_NMR_SP_high(i), C4(i)] = find(NMR2_hi(fb(i):(fb(i)+30e5))<1e-3, 1);
         Bdl_OP_high(i) = Bdl(i_NMR_OP_high(i)+fb(i));
        Bdl_SP_high(i) = Bdl(i_NMR_SP_high(i)+fb(i));
end

% Calculate mean integration constant (ignoring first three values)
mean_OP_high = mean(Bdl_OP_high(4:end));
mean_SP_high = mean(Bdl_SP_high(4:end));
std_OP_high = std(Bdl_OP_high(4:end));
std_SP_high = std(Bdl_SP_high(4:end));
```

Published with MATLAB® R2018b