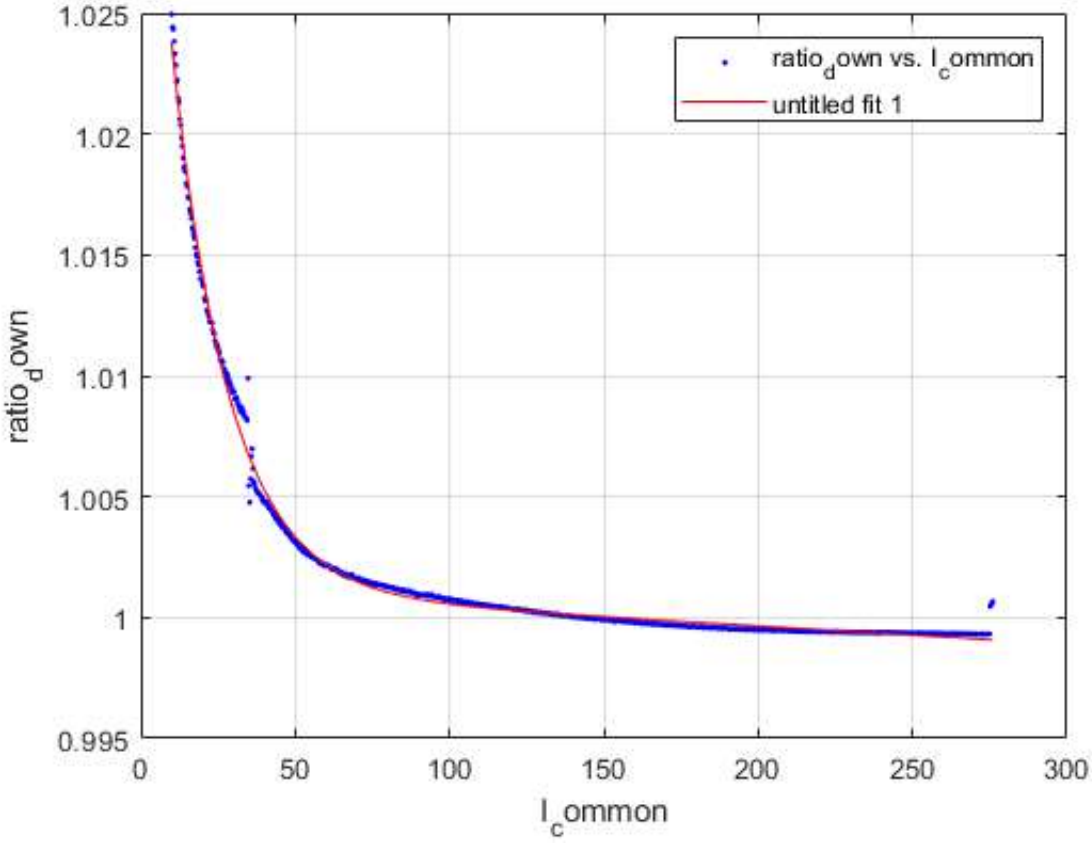
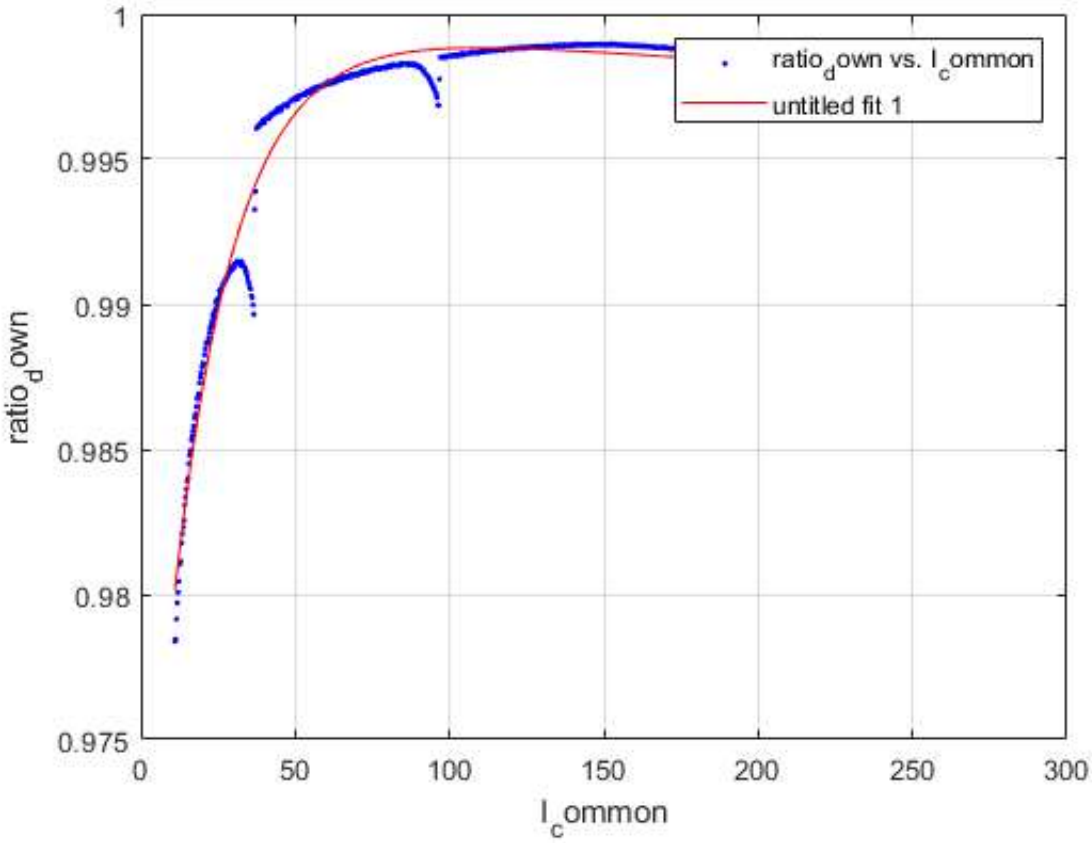
```
clear all;
close all;

testfolder = 'C:\Users\Dell\cernbox2\Commissioning\Measurement model\Scaling factor\scripts+mmm\ELENA\';
%   testfolder = 'C:\Users\Dell\cernbox\ELENA tests\MATLAB\Commissioning\matlab\scripts+mmm\ELENA\';
filelist = findtaggedfile(testfolder,'xmeter3_coil5_pos5','txt','multi',RECURSE.ON); nf = length(filelist);
load('200data');
load('PBMD2data')
load('PBMD1data')
% remanent field
% --------- analysis parameters --------------
np     =  300000;
kDCCT  =  1;
kH     = -1;
wc     =  2.8579; % coil effective width
refine =  1.0; % interpolation refinement ratio
l0     =  0.9708; % nominal (hard-edge) magnetic length

% --------- preallocate signal arrays ---------
t       = zeros(np,nf);   iav = 9;
I       = zeros(np,nf);
I200      = zeros(np,nf);
IPBMD2       = zeros(np,nf);
Vc      = zeros(np,nf);
BdL     = zeros(np,nf);
Bavg    = zeros(np,nf);
Lm      = zeros(np,nf);
TF      = zeros(np,nf);
V0      = zeros(1 ,nf);
delay   = zeros(1 ,nf);   % delay wrt mid=3 (first magnet in the ring)
t2 = linspace(0, length(BdL200)/100e3, length(BdL200)-1) ;
t3 = linspace(0, length(BdLPB)/100e3, length(BdLPB)-1) ;
mname   =  cell(1 ,nf+1);   mname{iav}='Ring Average'; mname{10}='Cycle Editor';


% % ----- machine mean (requires interpolation and delay) -----------------
dT2 = 1/100e3;
T2    = dT2*(1:2.1e6);
T3    = dT2*(1:5.8e6);
BdL_200 = zeros(refine*np,1);
BdL_PBMD2 = zeros(refine*np,1);
I_200 = zeros(refine*np,1);
I_PBMD2 = zeros(refine*np,1);

I_common = linspace(10, 276, 1000);

Bdl_200_up  =   interp1(curr200(60e3:200e3),BdL200(60e3:200e3), I_common);
Bdl_PBMD2_up  =  interp1(currPB(4e3:237e3),BdLPB(4e3:237e3), I_common);
Bdl_200_down  =   interp1(curr200(400e3:550e3),BdL200(400e3:550e3), I_common);
Bdl_PBMD1_down  =  interp1(currPB1(8.463e5:9.278e5),BdlPB1(8.463e5:9.278e5), I_common);

px1=currPB(350e3:1710e3);
py1=BdLPB(350e3:1710e3);

[Isorted, SortIndex] = sort(px1);                % Sort the current values in an ascending order
```

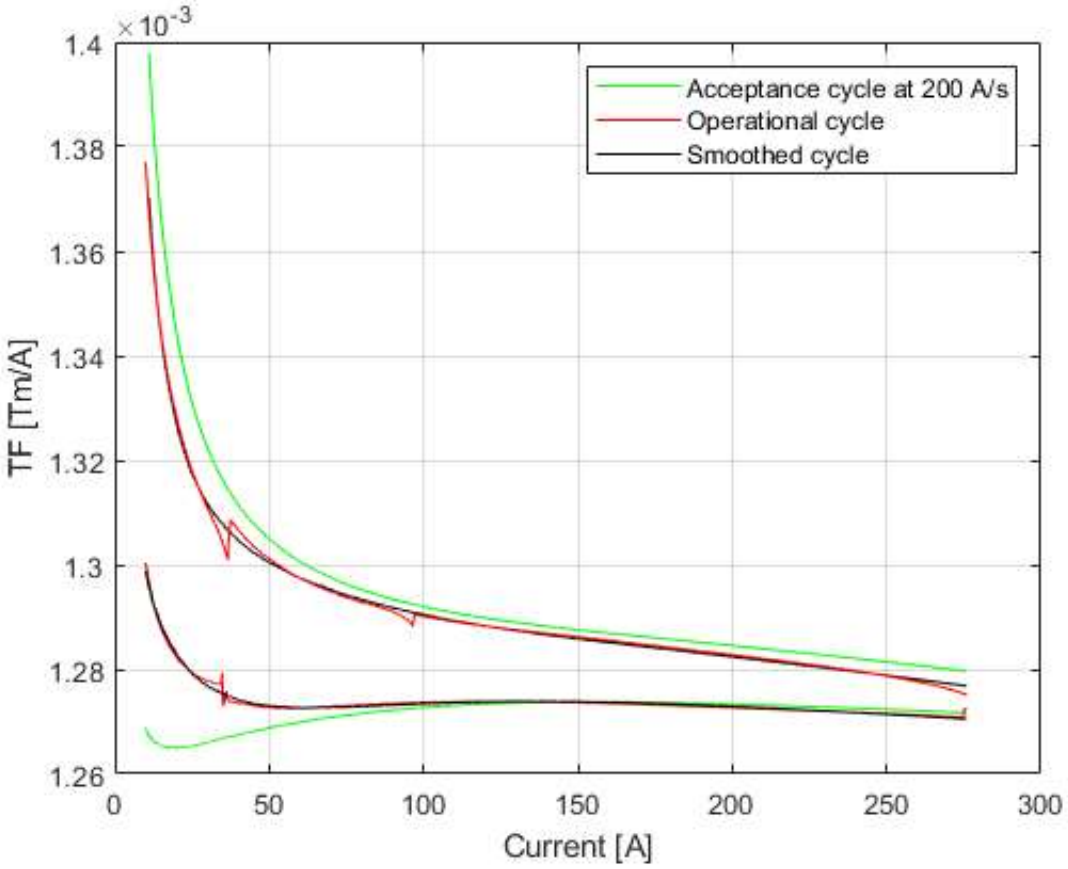
```
Bsorted = py1(SortIndex);                          % Sort the corresponding B-values to m
atch the sorted current values
[Iunique,ia,idx] = unique(Isorted,'stable');       % find repeated values of I and keep
 only one value
Bunique = accumarray(idx,Bsorted,[], @mean);       % find the mean of the corresponding
 B values. This could be changed to min/max for example
F = griddedInterpolant(Iunique,Bunique);           % Create an interpolant
Bdl_PBMD2_down = F(I_common);

px2=currPB1(5e5:8.376e5);
py2=BdlPB1(5e5:8.376e5);

[Isorted, SortIndex] = sort(px2);                  % Sort the current values in an ascend
ing order
Bsorted = py2(SortIndex);                          % Sort the corresponding B-values to m
atch the sorted current values
[Iunique,ia,idx] = unique(Isorted,'stable');       % find repeated values of I and keep
 only one value
Bunique = accumarray(idx,Bsorted,[], @mean);       % find the mean of the corresponding
 B values. This could be changed to min/max for example
F = griddedInterpolant(Iunique,Bunique);           % Create an interpolant
Bdl_PBMD1_up = F(I_common);

TF200_up = Bdl_200_up./I_common;
TFPB_up= Bdl_PBMD2_up./I_common;
TFPB1_up= Bdl_PBMD1_up./I_common;
TF200_down = Bdl_200_down./I_common;
TFPB_down= Bdl_PBMD2_down./I_common;
TFPB1_down= Bdl_PBMD1_down./I_common;

  ratio_up = TFPB_up./TF200_up;
ratio_down = TFPB_down./TF200_down;

  ratio_upPB1 = TFPB1_up./TF200_up;
ratio_downPB1 = TFPB1_down./TF200_down;

[fitresult, gof] = createFit(I_common, ratio_upPB1)
[fitresult1, gof1] = createFit(I_common, ratio_down)
% ------------------ plots ----------------------------------------------------------
----------------------------------
TF_pb1_up_pred = fitresult(I_common).*TF200_up';
TF_pb2_down_pred = fitresult1(I_common).*TF200_down';
 mid=[1:nf];

figure;
plot(I_common, TF200_up, 'g', I_common, TFPB1_up,'r',   I_common, TF_pb2_down_pred ,'k-'
, I_common, TF_pb1_up_pred ,'k-', I_common, TF200_down, 'g', I_common, TFPB_down,'r');
grid on;
legend('Acceptance cycle at 200 A/s',  'Operational cycle', 'Smoothed cycle');
ylabel('TF [Tm/A]');
xlabel('Current [A]');

figure;
plot( I_common, TF200_down, 'g', I_common, TFPB_down,'r', I_common, TF_pb2_down_pred ,'k
--', 'LineWidth',1.5);
legend('Acceptance cycle at 200 A/s',  'Operational cycle', 'Smoothed cycle (extrapolate
d)');
ylabel('$\mathcal{T}$ [Tm/A]','interpreter','latex');
xlabel('$I$ [A]','interpreter','latex');  ylim([1.27e-3 1.38e-3])
```

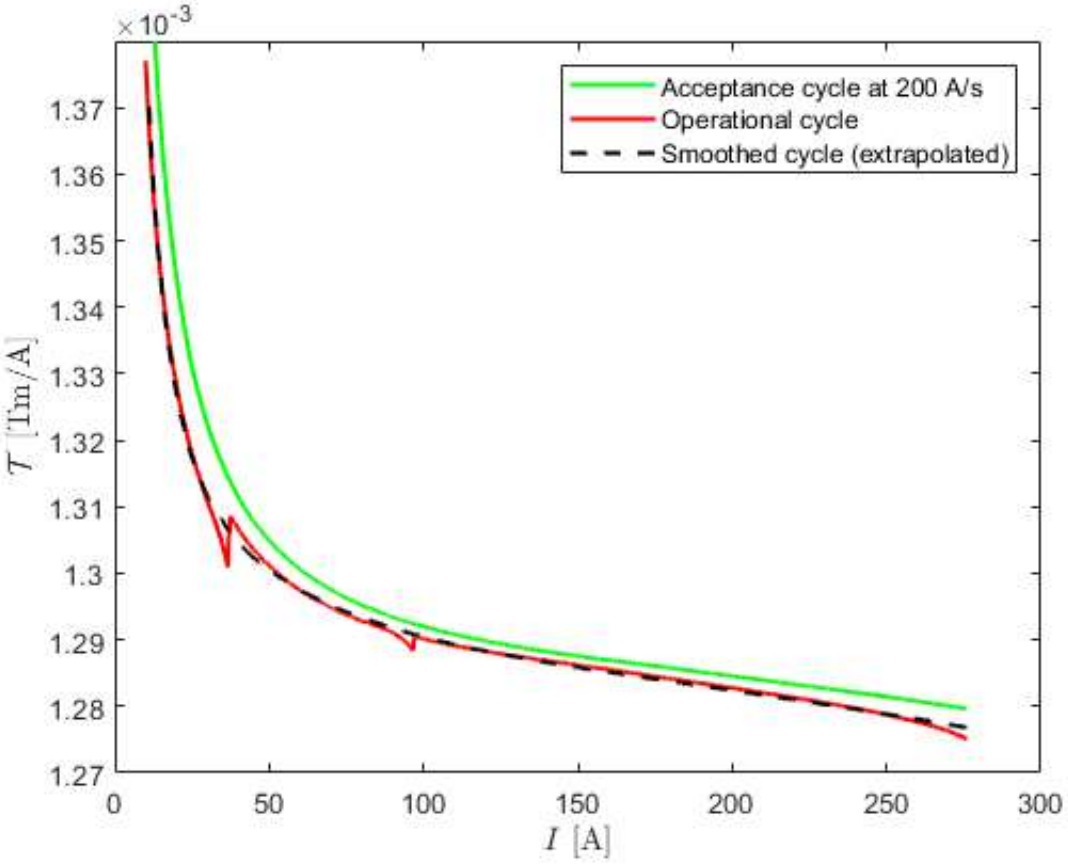
```
figure;
plot(curr200, BdL200'./curr200, 'k', currPB1, BdlPB1'./currPB1, 'b' , currPB, BdLPB'./cu
rrPB, 'r' );
legend('Test cycle','Accelerating cycle',  'Decelerating cycle');
ylabel('TF [Tm/A]');
xlabel('Current [A]');

figure;
plot(curr200(1:end/3), BdL200(1:end/3)'./curr200(1:end/3), 'k', currPB(1:end/3), BdLPB(1
:end/3)'./currPB(1:end/3), 'r' );  grid on;
legend('Acceptance cycle at 200 A/s', 'Decelerating cycle at 115 A/s');
ylabel('TF [Tm/A]'); ylim([1.26e-3 1.32e-3]);
xlabel('Current [A]');
```

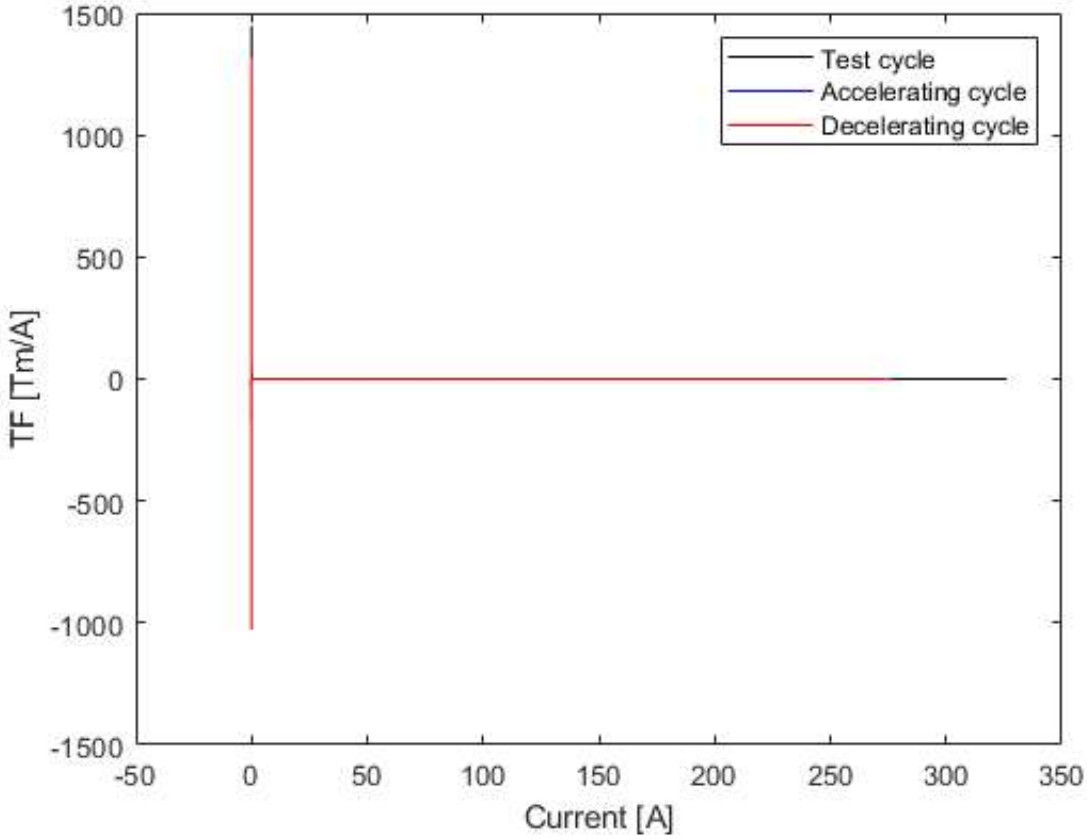
```
fitresult =
     General model Exp2:
     fitresult(x) = a*exp(b*x) + c*exp(d*x)
     Coefficients (with 95% confidence bounds):
       a =     0.03911  (0.0387, 0.03953)
       b =     -0.0544  (-0.05507, -0.05373)
       c =       1.001  (1.001, 1.001)
       d =  -7.491e-06  (-7.878e-06, -7.104e-06)
gof =
  struct with fields:

           sse: 8.7985e-05
       rsquare: 0.9953
           dfe: 996
    adjrsquare: 0.9953
          rmse: 2.9722e-04
Warning: Removing NaN and Inf from data
fitresult1 =
     General model Exp2:
     fitresult1(x) = a*exp(b*x) + c*exp(d*x)
     Coefficients (with 95% confidence bounds):
       a =    -0.03385  (-0.03465, -0.03305)
       b =    -0.04951  (-0.05091, -0.04812)
       c =      0.9998  (0.9996, 0.9999)
       d =  -7.204e-06  (-8.025e-06, -6.383e-06)
gof1 =
  struct with fields:

           sse: 3.4710e-04
       rsquare: 0.9676
           dfe: 992
    adjrsquare: 0.9675
          rmse: 5.9152e-04
Warning: Integer operands are required for colon operator when used as index.
Warning: Integer operands are required for colon operator when used as index.
Warning: Integer operands are required for colon operator when used as index.
Warning: Integer operands are required for colon operator when used as index.
Warning: Integer operands are required for colon operator when used as index.
Warning: Integer operands are required for colon operator when used as index.
```

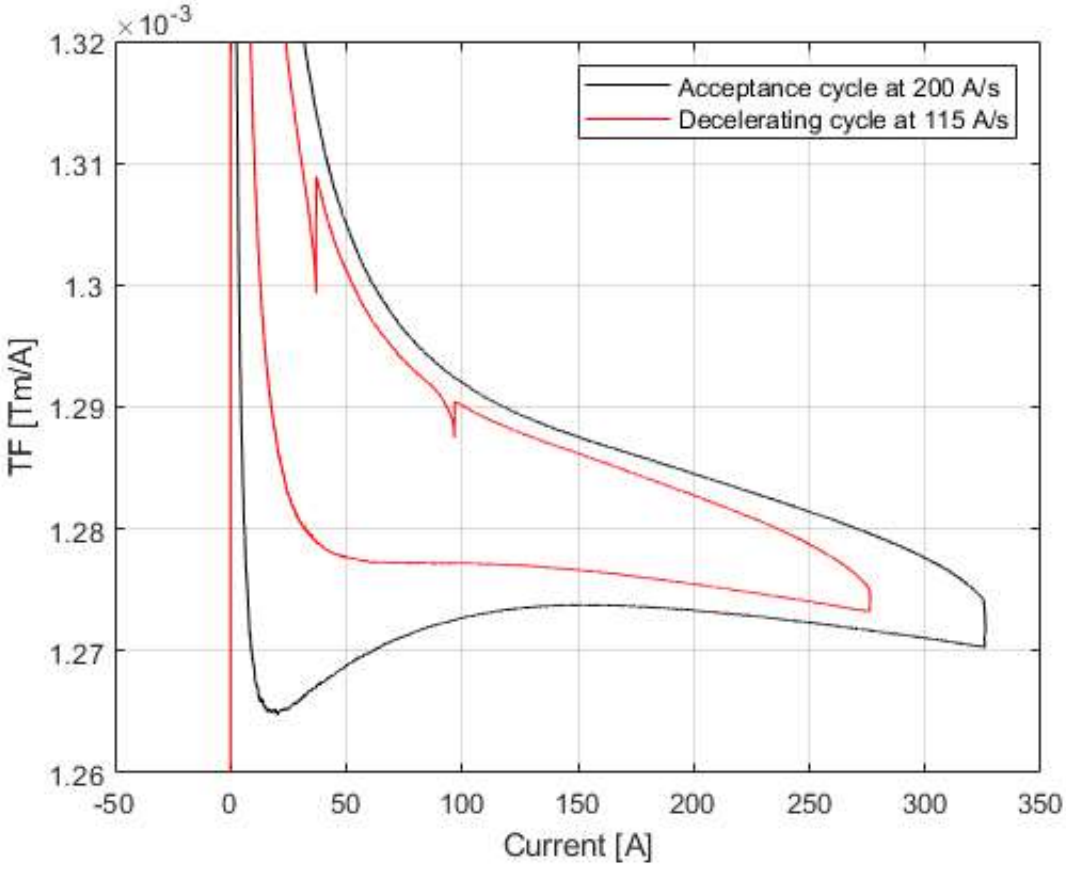

ratio_down vs. I_common
untitled fit 1
1.025
1.02
1.015
1.01
1.005
1
0.995
ratio_down
0
50
100
150
200
250
300
I_common


ratio_down vs. I_common
untitled fit 1
1
0.995
0.99
0.985
0.98
0.975
ratio_down
0
50
100
150
200
250
300
I_common


×10^-3
1.4
1.38
1.36
1.34
1.32
1.3
1.28
1.26
TF [Tm/A]
0
50
100
150
200
250
300
Current [A]
Acceptance cycle at 200 A/s
Operational cycle
Smoothed cycle


×10^-3
1.37
1.36
1.35
1.34
1.33
1.32
1.31
1.3
1.29
1.28
1.27
T [Tm/A]
0
50
100
150
200
250
300
I [A]
Acceptance cycle at 200 A/s
Operational cycle
Smoothed cycle (extrapolated)





Published with MATLAB® R2018b