Contents

ELENA magnet tests - Calculating magnetic length value along s axis

Author: Christian Grech (TE-MSC-MM) Date: 29/07/2019 Version: 1.0

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

```
count = 0;
for y = 1:3
```

```
      if y == 1
       x = 0;      elseif y ==2
       x = 500;      elseif y ==3
       x = 800;      end;
%     figure;
    for s = 1:9
        if s == 1
            v = 0;      elseif s ==2
            v = 100;      elseif s ==3
            v = 200;      elseif s ==4
            v = 220;      elseif s ==5
            v = 270;      elseif s ==6
            v = 330;      elseif s ==7
            v = 360;      elseif s ==8
            v = 400;      elseif s ==9
            v = 470;     end;

        for k = 1:3
```

```
            filename1 = sprintf('Voff_%d_%d_%d.mat',x,v, k);
            load(filename1);
            n = 1;
            % Separate data
             time1 = var(1:n:end, 1);
            Vc1nf = -var(1:n:end,3);
            Probe_1 = var(1:n:end, 4)/5;
            Probe_2 = var(1:n:end, 5)/5;
            Probe_3 = var(1:n:end, 6)/5;
            current_1 = var(1:n:end, 2)*100;
               count = count+1;
            f = round(1/(time1(2)-time1(1)));
```

Data filtering

```
 windowWidth = 400;
 windowWidth2 = 400;
windowWidth3 = 400;
 kernel = ones(windowWidth,1) / windowWidth;
 kernel2 = ones(windowWidth2,1) / windowWidth2;
kernel3 = ones(windowWidth3,1) / windowWidth3;
 curr = filtfilt(kernel, 1, current_1);
 Probe1f= filtfilt(kernel2, 1, Probe_1);
 Probe2f= filtfilt(kernel2, 1, Probe_2);
 Probe3f= filtfilt(kernel2, 1, Probe_3);
Vc1= filtfilt(kernel3, 1, Vc1nf);
```

Drift correction mechnism

```
    for i = 1: length(Vc1(1,:))
        flux1(:,i) = cumtrapz(time1(:,i), Vc1(:,i));
    end

%        figure; plot(flux1);
  drift_index = 8e4:10e4;
  pol = polyfit(time1(drift_index), flux1(drift_index),1);
  V01 = pol(1);

l = 0.9714;
wc = 2.8579;
r1= 0.65e4:4.724e4;
r2 = 1.075e5:1.486e5;
```

Calculating BdL

```
    Flux0 = 0.0019126;
    Phi1 = Flux0+cumtrapz(time1, Vc1-V01);
    BdL_sep = Phi1/(wc);
%     off2 = 6.43e-4;   % in Tm
     off2 = 0;
    BdL_sep1 = BdL_sep+off2;
    I_int = 60:1:274;   d=2;
    %I_int = 60:1:274;
    I_lp = [ I_int; I_int(end:-1:1)  ];
    BdL_int_up(:, count) =  interp1(curr(r1), BdL_sep1(r1), I_int);
    BdL_int_down(:, count) =  interp1(curr(r2), BdL_sep1(r2), I_int);
    B_int_up_1(:, count) =  interp1(curr(r1), Probe1f(r1), I_int);
    B_int_up_2(:, count) =  interp1(curr(r1), Probe2f(r1), I_int);
    B_int_up_3(:, count) =  interp1(curr(r1), Probe3f(r1), I_int);
    B_int_down_1(:, count) =  interp1(curr(r2), Probe1f(r2), I_int);
    B_int_down_2(:, count) =  interp1(curr(r2), Probe2f(r2), I_int);
    B_int_down_3(:, count) =  interp1(curr(r2), Probe3f(r2), I_int);
    B_int_lp_1(:, count) = [B_int_up_1(:, count); B_int_down_1(end:-1:1, cou
nt)];
    B_int_lp_2(:, count) = [B_int_up_2(:, count); B_int_down_2(end:-1:1, cou
nt)];
    B_int_lp_3(:, count) = [B_int_up_3(:, count); B_int_down_3(end:-1:1, cou
nt)];
    BdL_lp(:, count) = [BdL_int_up(:, count); BdL_int_down(end:-1:1, count)]
;
    lm_up_5mm_all(:, count) = BdL_int_up(:, count)./B_int_up_3(:, count);
    lm_up_16mm_all(:, count) = BdL_int_up(:, count)./B_int_up_2(:, count);
```

```
                lm_up_25mm_all(:, count) = BdL_int_up(:, count)./B_int_up_1(:, count);
                lm_down_5mm_all(:, count) = BdL_int_down(:, count)./B_int_down_3(:, coun
t);
                lm_down_16mm_all(:, count) = BdL_int_down(:, count)./B_int_down_2(:, cou
nt);
                lm_down_25mm_all(:, count) = BdL_int_down(:, count)./B_int_down_1(:, cou
nt);
                lm_lp_5mm_all(:, count) = BdL_lp(:, count)./B_int_lp_3(:, count);
                lm_lp_16mm_all(:, count) = BdL_lp(:, count)./B_int_lp_2(:, count);
                lm_lp_25mm_all(:, count) = BdL_lp(:, count)./B_int_lp_1(:, count);

                if k == 3
                       downsamp = count/3;
                    lm_up_5mm(:, downsamp) = mean(lm_up_5mm_all(:,(count-2):count)');
                                                        sd_lm_up_5mm(:, downsamp) = std(lm_
up_5mm_all(:,(count-2):count)');
                    lm_up_16mm(:, downsamp) = mean(lm_up_16mm_all(:,(count-2):count)');
                                                      sd_lm_up_16mm(:, downsamp) = std(lm_u
p_16mm_all(:,(count-2):count)');
                    lm_up_25mm(:, downsamp) = mean(lm_up_25mm_all(:,(count-2):count)');
                                                      sd_lm_up_25mm(:, downsamp) = std(lm_u
p_25mm_all(:,(count-2):count)');
                    lm_down_5mm(:, downsamp) = mean(lm_down_5mm_all(:,(count-2):count)')
;                                                    sd_lm_down_5mm(:, downsamp) = std(lm_dow
n_5mm_all(:,(count-2):count)');
                    lm_down_16mm(:, downsamp) = mean(lm_down_16mm_all(:,(count-2):count)
');                                                  sd_lm_down_16mm(:, downsamp) = std(lm_dow
n_16mm_all(:,(count-2):count)');
                    lm_down_25mm(:, downsamp) = mean(lm_down_25mm_all(:,(count-2):count)
');                                                  sd_lm_down_25mm(:, downsamp) = std(lm_dow
n_25mm_all(:,(count-2):count)');
                    lm_lp_5mm(:, downsamp) = mean(lm_lp_5mm_all(:,(count-2):count)');
                                                            sd_lm_lp_5mm(:, downsamp) = st
d(lm_lp_5mm_all(:,(count-2):count)');
                    lm_lp_16mm(:, downsamp) = mean(lm_lp_16mm_all(:,(count-2):count)');
                                                           sd_lm_lp_16mm(:, downsamp) = std
(lm_lp_16mm_all(:,(count-2):count)');
                    lm_lp_25mm(:, downsamp) = mean(lm_lp_25mm_all(:,(count-2):count)');
                                                           sd_lm_lp_25mm(:, downsamp) = std
(lm_lp_25mm_all(:,(count-2):count)');
                     column_track(:, downsamp) = [x; v; k];
                end;
```

```
        end;
    end;
```

Interpolate wrt s

```
         s_int = 0:0.05:470;    m=9;
         s_actual = column_track(2, 1:m); % 9 unique distances on the s-axis

        if y==1
             lm_int_up_5mm_x0 = interp1(s_actual, lm_up_5mm(:,1:m)', s_int);
             lm_int_down_5mm_x0 = interp1(s_actual, lm_down_5mm(:,1:m)', s_int);
             lm_int_lp_5mm_x0 = interp1(s_actual, lm_lp_5mm(:,1:m)', s_int);
             lm_int_up_16mm_x0 = interp1(s_actual, lm_up_16mm(:,1:m)', s_int);
             lm_int_down_16mm_x0 = interp1(s_actual, lm_down_16mm(:,1:m)', s_int);
             lm_int_lp_16mm_x0 = interp1(s_actual, lm_lp_16mm(:,1:m)', s_int);
```

```
            lm_int_up_25mm_x0 = interp1(s_actual, lm_up_25mm(:,1:m)', s_int);
            lm_int_down_25mm_x0 = interp1(s_actual, lm_down_25mm(:,1:m)', s_int);
            lm_int_lp_25mm_x0 = interp1(s_actual, lm_lp_25mm(:,1:m)', s_int);
%                [s_star_up, lm_star_up, dlm_rel_star_up, dlm_star_up, dlm_rel_0_up, improvement_up, dlm_up_5mm_0] = findOptimalSingleSensorPosition_new( lm_int_up_5mm_x0,s_int, 'UP, 5mm, central', 'rx-');
%                fprintf('UP @ 5 mm; central:   s* = %f,\t Lm*=%f,\t DeltaLm*=%f,\t  DeltaLm_rel*=%f,\t DeltaLm_rel0=%f,\t improvement =%f \n',s_star_up, lm_star_up, dlm_star_up, dlm_rel_star_up, dlm_rel_0_up, improvement_up);
%                [s_star_down, lm_star_down, dlm_rel_star_down, dlm_star_down, dlm_rel_0_down, improvement_down, dlm_dw_5mm_0] = findOptimalSingleSensorPosition_new( lm_int_down_5mm_x0,s_int, 'DOWN, 5mm, central', 'rx-');
%                fprintf('DOWN @ 5 mm, central:   s* = %f,\t Lm*=%f,\t DeltaLm*=%f,\t  DeltaLm_rel*=%f,\t DeltaLm_rel0=%f,\t improvement =%f \n',s_star_down, lm_star_down, dlm_star_down, dlm_rel_star_down, dlm_rel_0_down, improvement_down);
            [s_star_lp, lm_star_lp, dlm_rel_star_lp, dlm_star_lp, dlm_rel_0_lp, improvement_lp, dlm_lp_5mm_0, h(1,:)] = findOptimalSingleSensorPosition_new( lm_int_lp_5mm_x0,s_int, 'LOOP, 5mm, central', 'rx-');
            fprintf('LOOP @ 5 mm; central:   s* = %f,\t Lm*=%f,\t DeltaLm*=%f,\t  DeltaLm_rel*=%f,\t DeltaLm_rel0=%f,\t improvement =%f \n',s_star_lp, lm_star_lp, dlm_star_lp, dlm_rel_star_lp, dlm_rel_0_lp, improvement_lp);
%                [s_star_up16, lm_star_up16, dlm_rel_star_up16, dlm_star_up16, dlm_rel_0_up16, improvement_up16, dlm_up_16mm_0] = findOptimalSingleSensorPosition_new( lm_int_up_16mm_x0,s_int, 'UP, 16mm, central', 'rx-');
%                fprintf('UP @ 16 mm, central:   s* = %f,\t Lm*=%f,\t DeltaLm*=%f,\t  DeltaLm_rel*=%f,\t DeltaLm_rel0=%f,\t improvement =%f \n',s_star_up16, lm_star_up16, dlm_star_up16, dlm_rel_star_up16, dlm_rel_0_up16, improvement_up16);
%                [s_star_down16, lm_star_down16, dlm_rel_star_down16, dlm_star_down16, dlm_rel_0_down16, improvement_down16, dlm_dw_16mm_0] = findOptimalSingleSensorPosition_new( lm_int_down_16mm_x0,s_int, 'DOWN, 16mm, central', 'rx-');
%                fprintf('DOWN @ 16 mm, central:   s* = %f,\t Lm*=%f,\t DeltaLm*=%f,\t  DeltaLm_rel*=%f,\t DeltaLm_rel0=%f,\t improvement =%f \n',s_star_down16, lm_star_down16, dlm_star_down16, dlm_rel_star_down16, dlm_rel_0_down16, improvement_down16);
              [s_star_lp16, lm_star_lp16, dlm_rel_star_lp16, dlm_star_lp16, dlm_rel_0_lp16, improvement_lp16, dlm_lp_16mm_0, h(2,:)] = findOptimalSingleSensorPosition_new( lm_int_lp_16mm_x0,s_int, 'LOOP, 16mm, central', 'gx-');
            fprintf('LOOP @ 16 mm, central:   s* = %f,\t Lm*=%f,\t DeltaLm*=%f,\t  DeltaLm_rel*=%f,\t DeltaLm_rel0=%f,\t improvement =%f \n',s_star_lp16, lm_star_lp16, dlm_star_lp16, dlm_rel_star_lp16, dlm_rel_0_lp16, improvement_lp16);
%                [s_star_up25, lm_star_up25, dlm_rel_star_up25, dlm_star_up25, dlm_rel_0_up25, improvement_up25, dlm_up_25mm_0] = findOptimalSingleSensorPosition_new( lm_int_up_25mm_x0,s_int, 'UP, 25mm, central', 'rx-');
%                fprintf('UP @ 25 mm, central:   s* = %f,\t Lm*=%f,\t DeltaLm*=%f,\t  DeltaLm_rel*=%f,\t DeltaLm_rel0=%f,\t improvement =%f \n',s_star_up25, lm_star_up25, dlm_star_up25, dlm_rel_star_up25, dlm_rel_0_up25, improvement_up25);
%                [s_star_down25, lm_star_down25, dlm_rel_star_down25, dlm_star_down25, dlm_rel_0_down25, improvement_down25, dlm_dw_25mm_0] = findOptimalSingleSensorPosition_new( lm_int_down_25mm_x0,s_int, 'DOWN, 25mm, central', 'rx-');
%                fprintf('DOWN @ 25 mm, central:   s* = %f,\t Lm*=%f,\t DeltaLm*=%f,\t  DeltaLm_rel*=%f,\t DeltaLm_rel0=%f,\t improvement =%f \n',s_star_down25, lm_star_down25, dlm_star_down25, dlm_rel_star_down25, dlm_rel_0_down25, improvement_down25);
            [s_star_lp25, lm_star_lp25, dlm_rel_star_lp25, dlm_star_lp25, dlm_rel_0_lp25, improvement_lp25, dlm_lp_25mm_0, h(3,:)] = findOptimalSingleSensorPosition_new( lm_int_lp_25mm_x0,s_int, 'LOOP, 25mm, central', 'bx-');
            fprintf('LOOP @ 25 mm, central:   s* = %f,\t Lm*=%f,\t DeltaLm*=%f,\t  DeltaLm_rel*=%f,\t DeltaLm_rel0=%f,\t improvement =%f \n',s_star_lp25, lm_star_lp25, dlm_star_lp25, dlm_rel_star_lp25, dlm_rel_0_lp25, improvement_lp25);
            legtext = {'y = 5 mm',...
          'y = 16 mm',...
          'y = 25 mm'};
           styles  = {'r-','g-','b-',};
```

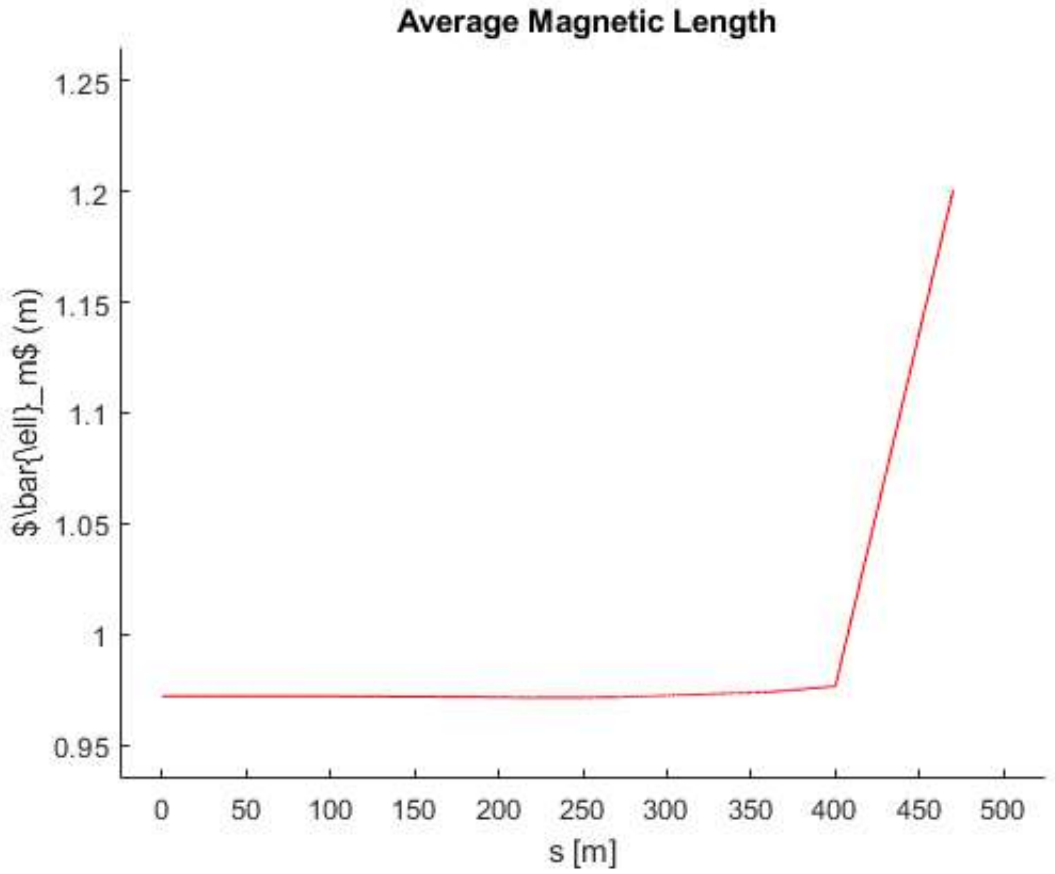
```
            % combinePlots([h(1,1),h(2,1),h(3,1),h(1,3),h(2,3),h(3,3)],'lin','log',legte
xt,styles);  hold on; xlabel('s (m)'); ylabel('(m) or (-)'); hold off;
            combinePlots([h(1,3),h(2,3),h(3,3)],'lin','log',legtext,styles);  hold on; x
label('s [mm]'); ylabel('$\Delta\ell_m/\bar{\ell}_m$ '); hold on;
            xl = xline(430, '.-k',{'Magnet pole'});
            xl.LabelVerticalAlignment = 'middle';
            xl.LabelHorizontalAlignment = 'center';hold off;
        elseif y==2
             lm_int_up_5mm_x500 = interp1(s_actual, lm_up_5mm(:,(m+1):(2*m))', s_int);
        lm_int_lp_5mm_x500 = interp1(s_actual, lm_lp_5mm(:,(m+1):(2*m))', s_int);
             lm_int_down_5mm_x500 = interp1(s_actual, lm_down_5mm(:,(m+1):(2*m))', s_int
);
             lm_int_up_16mm_x500 = interp1(s_actual, lm_up_16mm(:,(m+1):(2*m))', s_int);
      lm_int_lp_16mm_x500 = interp1(s_actual, lm_lp_16mm(:,(m+1):(2*m))', s_int);
             lm_int_down_16mm_x500 = interp1(s_actual, lm_down_16mm(:,(m+1):(2*m))', s_i
nt);
             lm_int_up_25mm_x500 = interp1(s_actual, lm_up_25mm(:,(m+1):(2*m))', s_int);
      lm_int_lp_25mm_x500 = interp1(s_actual, lm_lp_25mm(:,(m+1):(2*m))', s_int);
             lm_int_down_25mm_x500 = interp1(s_actual, lm_down_25mm(:,(m+1):(2*m))', s_i
nt);
%              [s_star_up, lm_star_up, dlm_rel_star_up, dlm_star_up, dlm_rel_0_up, impro
vement_up, dlm_up_5mm_5] = findOptimalSingleSensorPosition_new( lm_int_up_5mm_x500,s_int
, 'UP, 5mm,  x=500mm', 'rx-');
%               fprintf('UP @ 5 mm; x=500:   s* = %f,\t Lm*=%f,\t DeltaLm*=%f,\t  DeltaLm
_rel*=%f,\t DeltaLm_rel0=%f,\t improvement =%f \n',s_star_up, lm_star_up, dlm_star_up, d
lm_rel_star_up, dlm_rel_0_up, improvement_up);
%               [s_star_down, lm_star_down, dlm_rel_star_down, dlm_star_down, dlm_rel_0_
down, improvement_down, dlm_dw_5mm_5, dlm_dw_5mm_5] = findOptimalSingleSensorPosition_ne
w( lm_int_down_5mm_x500,s_int, 'DOWN, 5mm, x=500mm', 'rx-');
%               fprintf('DOWN @ 5 mm, x=500:   s* = %f,\t Lm*=%f,\t DeltaLm*=%f,\t  Delta
Lm_rel*=%f,\t DeltaLm_rel0=%f,\t improvement =%f \n',s_star_down, lm_star_down, dlm_star
_down, dlm_rel_star_down, dlm_rel_0_down, improvement_down);
             [s_star_lp, lm_star_lp, dlm_rel_star_lp, dlm_star_lp, dlm_rel_0_lp, improve
ment_lp, dlm_lp_5mm_5,  p(1,:)] = findOptimalSingleSensorPosition_new( lm_int_lp_5mm_x50
0,s_int, 'LOOP, 5mm, x=500mm', 'kx-');
             fprintf('LOOP @ 5 mm; x=500:   s* = %f,\t Lm*=%f,\t DeltaLm*=%f,\t  DeltaLm
_rel*=%f,\t DeltaLm_rel0=%f,\t improvement =%f \n',s_star_lp, lm_star_lp, dlm_star_lp, d
lm_rel_star_lp, dlm_rel_0_lp, improvement_lp);
%               [s_star_up16, lm_star_up16, dlm_rel_star_up16, dlm_star_up16, dlm_rel_0_
up16, improvement_up16, dlm_up_16mm_5] = findOptimalSingleSensorPosition_new( lm_int_up_
16mm_x500,s_int, 'UP, 16mm, x=500mm', 'rx-');
%              fprintf('UP @ 16 mm, x=500:   s* = %f,\t Lm*=%f,\t DeltaLm*=%f,\t  DeltaL
m_rel*=%f,\t DeltaLm_rel0=%f,\t improvement =%f \n',s_star_up16, lm_star_up16, dlm_star_
up16, dlm_rel_star_up16, dlm_rel_0_up16, improvement_up16);
%               [s_star_down16, lm_star_down16, dlm_rel_star_down16, dlm_star_down16, dl
m_rel_0_down16, improvement_down16, dlm_dw_16mm_5] = findOptimalSingleSensorPosition_new
( lm_int_down_16mm_x500,s_int, 'DOWN, 16mm, x=500mm', 'rx-');
%               fprintf('DOWN @ 16 mm, x=500:   s* = %f,\t Lm*=%f,\t DeltaLm*=%f,\t  Delt
aLm_rel*=%f,\t DeltaLm_rel0=%f,\t improvement =%f \n',s_star_down16, lm_star_down16, dlm
_star_down16, dlm_rel_star_down16, dlm_rel_0_down16, improvement_down16);
             [s_star_lp16, lm_star_lp16, dlm_rel_star_lp16, dlm_star_lp16, dlm_rel_0_lp1
6, improvement_lp16, dlm_lp_16mm_5,  p(2,:)] = findOptimalSingleSensorPosition_new( lm_i
nt_lp_16mm_x500,s_int, 'LOOP, 16mm, x=500', 'mx-');
             fprintf('LOOP @ 16 mm, x=500:   s* = %f,\t Lm*=%f,\t DeltaLm*=%f,\t  DeltaL
m_rel*=%f,\t DeltaLm_rel0=%f,\t improvement =%f \n',s_star_lp16, lm_star_lp16, dlm_star_
lp16, dlm_rel_star_lp16, dlm_rel_0_lp16, improvement_lp16);
%               [s_star_up25, lm_star_up25, dlm_rel_star_up25, dlm_star_up25, dlm_rel_0_u
p25, improvement_up25, dlm_up_25mm_5] = findOptimalSingleSensorPosition_new( lm_int_up_2
5mm_x500,s_int, 'UP, 25mm, x=500mm', 'rx-');
%              fprintf('UP @ 25 mm, x=500:   s* = %f,\t Lm*=%f,\t DeltaLm*=%f,\t  DeltaL
m_rel*=%f,\t DeltaLm_rel0=%f,\t improvement =%f \n',s_star_up25, lm_star_up25, dlm_star_
```

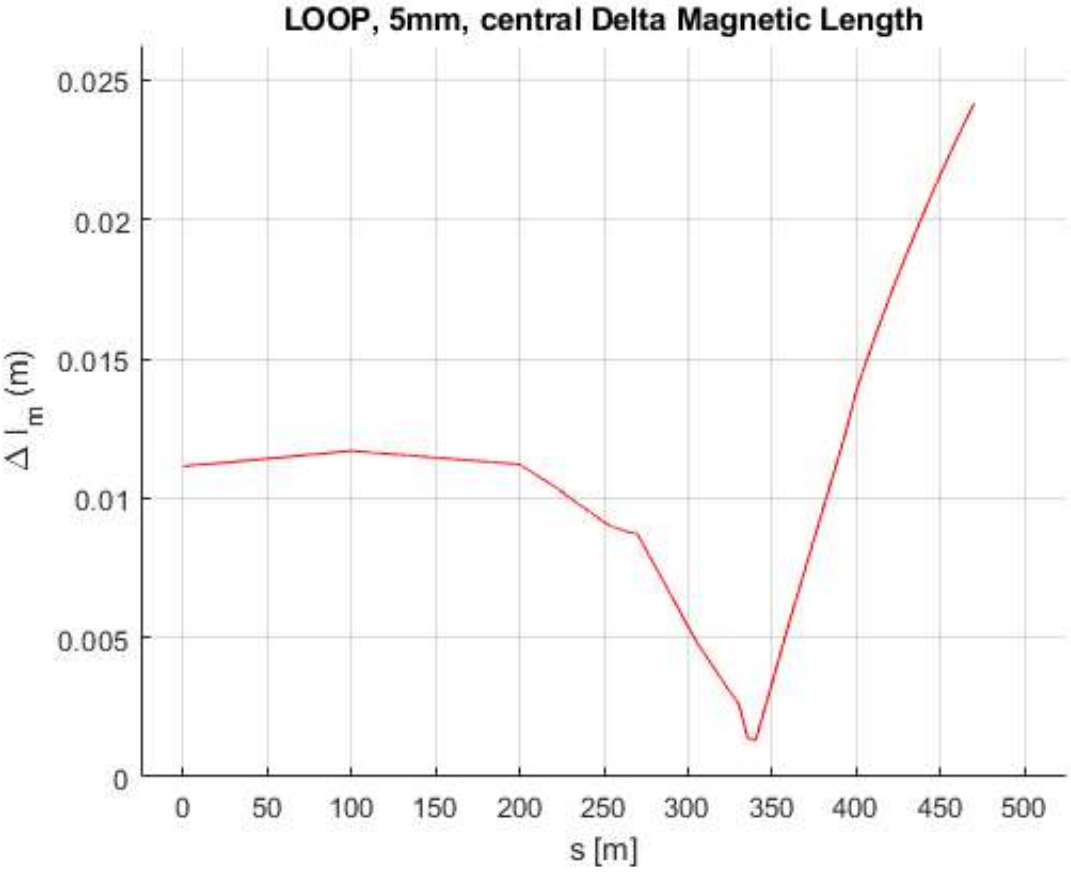
```
up25, dlm_rel_star_up25, dlm_rel_0_up25, improvement_up25);
%               [s_star_down25, lm_star_down25, dlm_rel_star_down25, dlm_star_down25, dl
m_rel_0_down25, improvement_down25, dlm_dw_25mm_5] = findOptimalSingleSensorPosition_new
( lm_int_down_25mm_x500,s_int, 'DOWN, 25mm, x=500mm', 'rx-');
%               fprintf('DOWN @ 25 mm, x=500:   s* = %f,\t Lm*=%f,\t DeltaLm*=%f,\t  Delt
aLm_rel*=%f,\t DeltaLm_rel0=%f,\t improvement =%f \n',s_star_down25, lm_star_down25, dlm
_star_down25, dlm_rel_star_down25, dlm_rel_0_down25, improvement_down25);
              [s_star_lp25, lm_star_lp25, dlm_rel_star_lp25, dlm_star_lp25, dlm_rel_0_lp2
5, improvement_lp25, dlm_lp_25mm_5,  p(3,:)] = findOptimalSingleSensorPosition_new( lm_i
nt_lp_25mm_x500,s_int, 'LOOP, 25mm, x=500', 'cx-');
              fprintf('LOOP @ 25 mm, x=500:   s* = %f,\t Lm*=%f,\t DeltaLm*=%f,\t  DeltaL
m_rel*=%f,\t DeltaLm_rel0=%f,\t improvement =%f \n',s_star_lp25, lm_star_lp25, dlm_star_
lp25, dlm_rel_star_lp25, dlm_rel_0_lp25, improvement_lp25);
              legtext = {'y = 5 mm',...
             'y = 16 mm',...
             'y = 25 mm'};
              styles  = {'r-','g-','b-',};
              % combinePlots([h(1,1),h(2,1),h(3,1),h(1,3),h(2,3),h(3,3)],'lin','log',legte
xt,styles);  hold on; xlabel('s (m)'); ylabel('(m) or (-)'); hold off;
              combinePlots([p(1,3),p(2,3),p(3,3)],'lin','log',legtext,styles);  hold on; x
label('s [mm]'); ylabel('$\Delta\ell_m/\bar{\ell}_m$'); hold on;
              xl = xline(430, '.-k',{'Magnet pole'});
              xl.LabelVerticalAlignment = 'middle';
              xl.LabelHorizontalAlignment = 'center';hold off;
          else
               lm_int_up_5mm_x800 = interp1(s_actual, lm_up_5mm(:,(2*m+1):(3*m))', s_int);
                                    lm_int_lp_5mm_x800 = interp1(s_actual, lm_lp_5mm(:,(2*m
+1):(3*m))', s_int);
               lm_int_down_5mm_x800 = interp1(s_actual, lm_down_5mm(:,(2*m+1):(3*m))', s_i
nt);
               lm_int_up_16mm_x800 = interp1(s_actual, lm_up_16mm(:,(2*m+1):(3*m))', s_int
);                                  lm_int_lp_16mm_x800 = interp1(s_actual, lm_lp_16mm(:,(2*
m+1):(3*m))', s_int);
               lm_int_down_16mm_x800 = interp1(s_actual, lm_down_16mm(:,(2*m+1):(3*m))', s
_int);
               lm_int_up_25mm_x800 = interp1(s_actual, lm_up_25mm(:,(2*m+1):(3*m))', s_int
);                                  lm_int_lp_25mm_x800 = interp1(s_actual, lm_lp_25mm(:,(2*
m+1):(3*m))', s_int);
               lm_int_down_25mm_x800 = interp1(s_actual, lm_down_25mm(:,(2*m+1):(3*m))', s
_int);
%               [s_star_up, lm_star_up, dlm_rel_star_up, dlm_star_up, dlm_rel_0_up, impro
vement_up, dlm_up_5mm_8] = findOptimalSingleSensorPosition_new( lm_int_up_5mm_x800,s_int
, 'UP, 5mm,  x=800mm', 'rx-');
%               fprintf('UP @ 5 mm; x=800:   s* = %f,\t Lm*=%f,\t DeltaLm*=%f,\t  DeltaLm
_rel*=%f,\t DeltaLm_rel0=%f,\t improvement =%f \n',s_star_up, lm_star_up, dlm_star_up, d
lm_rel_star_up, dlm_rel_0_up, improvement_up);
%               [s_star_down, lm_star_down, dlm_rel_star_down, dlm_star_down, dlm_rel_0_
down, improvement_down, dlm_dw_5mm_8] = findOptimalSingleSensorPosition_new( lm_int_down
_5mm_x800,s_int, 'DOWN, 5mm, x=800mm', 'rx-');
%               fprintf('DOWN @ 5 mm, x=800:   s* = %f,\t Lm*=%f,\t DeltaLm*=%f,\t  Delta
Lm_rel*=%f,\t DeltaLm_rel0=%f,\t improvement =%f \n',s_star_down, lm_star_down, dlm_star
_down, dlm_rel_star_down, dlm_rel_0_down, improvement_down);
               [s_star_lp, lm_star_lp, dlm_rel_star_lp, dlm_star_lp, dlm_rel_0_lp, impro
vement_lp, dlm_lp_5mm_8, q(1,:)] = findOptimalSingleSensorPosition_new( lm_int_lp_5mm_x8
00,s_int, 'LOOP, 5mm, x=800mm', 'rx--');
               fprintf('LOOP @ 5 mm; x=800:   s* = %f,\t Lm*=%f,\t DeltaLm*=%f,\t  DeltaLm
_rel*=%f,\t DeltaLm_rel0=%f,\t improvement =%f \n',s_star_lp, lm_star_lp, dlm_star_lp, d
lm_rel_star_lp, dlm_rel_0_lp, improvement_lp);
%               [s_star_up16, lm_star_up16, dlm_rel_star_up16, dlm_star_up16, dlm_rel_0_
up16, improvement_up16, dlm_up_16mm_8] = findOptimalSingleSensorPosition_new( lm_int_up_
16mm_x800,s_int, 'UP, 16mm, x=800mm', 'rx-');
```

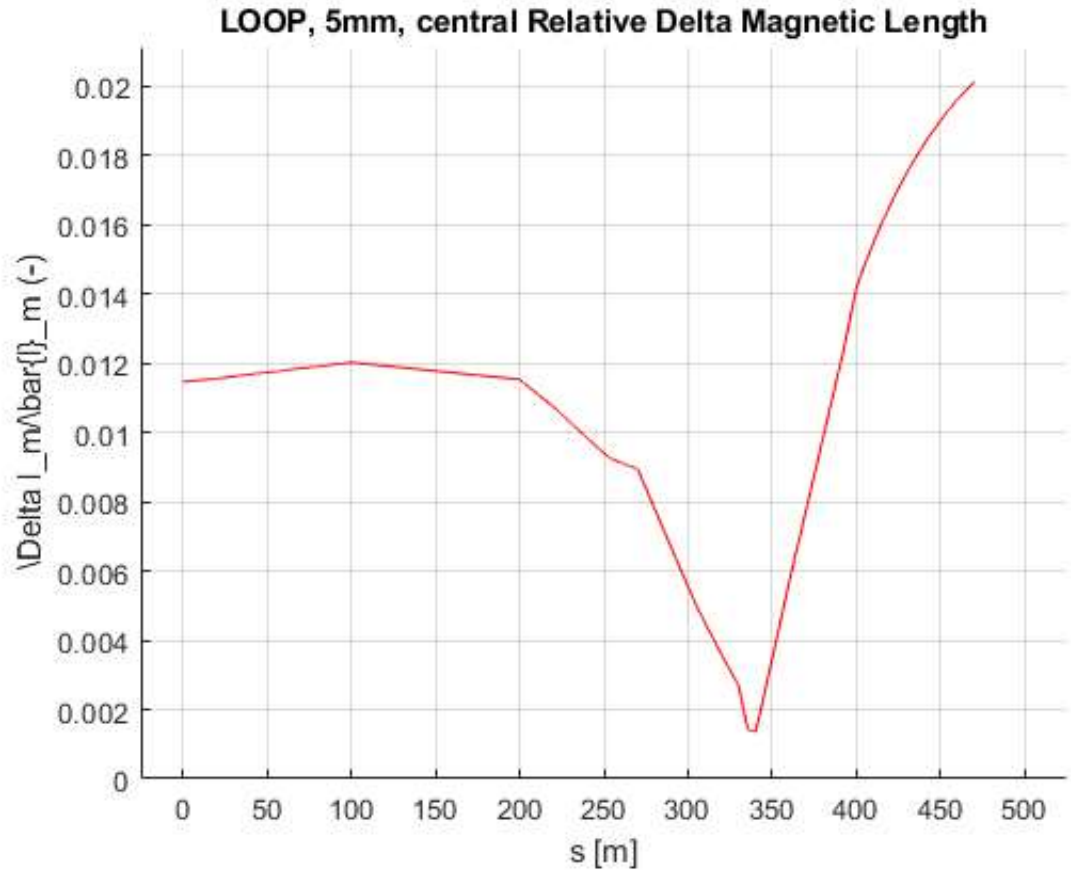
```
%              fprintf('UP @ 16 mm, x=800:   s* = %f,\t Lm*=%f,\t DeltaLm*=%f,\t  DeltaL
m_rel*=%f,\t DeltaLm_rel0=%f,\t improvement =%f \n',s_star_up16, lm_star_up16, dlm_star_
up16, dlm_rel_star_up16, dlm_rel_0_up16, improvement_up16);
%               [s_star_down16, lm_star_down16, dlm_rel_star_down16, dlm_star_down16, dl
m_rel_0_down16, improvement_down16, dlm_dw_16mm_8] = findOptimalSingleSensorPosition_new
( lm_int_down_16mm_x800,s_int, 'DOWN, 16mm, x=800mm', 'rx-');
%              fprintf('DOWN @ 16 mm, x=800:   s* = %f,\t Lm*=%f,\t DeltaLm*=%f,\t  Delt
aLm_rel*=%f,\t DeltaLm_rel0=%f,\t improvement =%f \n',s_star_down16, lm_star_down16, dlm
_star_down16, dlm_rel_star_down16, dlm_rel_0_down16, improvement_down16);
             [s_star_lp16, lm_star_lp16, dlm_rel_star_lp16, dlm_star_lp16, dlm_rel_0_lp1
6, improvement_lp16, dlm_lp_16mm_8, q(2,:)] = findOptimalSingleSensorPosition_new( lm_in
t_lp_16mm_x800,s_int, 'LOOP, 16mm, x=800mm', 'gx--');
             fprintf('LOOP @ 16 mm, x=800:   s* = %f,\t Lm*=%f,\t DeltaLm*=%f,\t  DeltaL
m_rel*=%f,\t DeltaLm_rel0=%f,\t improvement =%f \n',s_star_lp16, lm_star_lp16, dlm_star_
lp16, dlm_rel_star_lp16, dlm_rel_0_lp16, improvement_lp16);
%              [s_star_up25, lm_star_up25, dlm_rel_star_up25, dlm_star_up25, dlm_rel_0_u
p25, improvement_up25, dlm_up_25mm_8] = findOptimalSingleSensorPosition_new( lm_int_up_2
5mm_x800,s_int, 'UP, 25mm, x=800mm', 'rx-');
%              fprintf('UP @ 25 mm, x=800:   s* = %f,\t Lm*=%f,\t DeltaLm*=%f,\t  DeltaL
m_rel*=%f,\t DeltaLm_rel0=%f,\t improvement =%f \n',s_star_up25, lm_star_up25, dlm_star_
up25, dlm_rel_star_up25, dlm_rel_0_up25, improvement_up25);
%               [s_star_down25, lm_star_down25, dlm_rel_star_down25, dlm_star_down25, dl
m_rel_0_down25, improvement_down25, dlm_dw_25mm_8] = findOptimalSingleSensorPosition_new
( lm_int_down_25mm_x800,s_int, 'DOWN, 25mm, x=800mm', 'rx-');
%              fprintf('DOWN @ 25 mm, x=800:   s* = %f,\t Lm*=%f,\t DeltaLm*=%f,\t  Delt
aLm_rel*=%f,\t DeltaLm_rel0=%f,\t improvement =%f \n',s_star_down25, lm_star_down25, dlm
_star_down25, dlm_rel_star_down25, dlm_rel_0_down25, improvement_down25);
              [s_star_lp25, lm_star_lp25, dlm_rel_star_lp25, dlm_star_lp25, dlm_rel_0_lp
25, improvement_lp25, dlm_lp_25mm_8, q(3,:)] = findOptimalSingleSensorPosition_new( lm_i
nt_lp_25mm_x800,s_int, 'LOOP, 25mm, x=800mm', 'bx--');
             fprintf('LOOP @ 25 mm, x=800:   s* = %f,\t Lm*=%f,\t DeltaLm*=%f,\t  DeltaL
m_rel*=%f,\t DeltaLm_rel0=%f,\t improvement =%f \n',s_star_lp25, lm_star_lp25, dlm_star_
lp25, dlm_rel_star_lp25, dlm_rel_0_lp25, improvement_lp25);
%              legtext = {'y = 5 mm',...
%             'y = 16 mm',...
%             'y = 25 mm'};
%              styles  = {'r-','g-','b-',};
%              % combinePlots([h(1,1),h(2,1),h(3,1),h(1,3),h(2,3),h(3,3)],'lin','log',leg
text,styles);  hold on; xlabel('s (m)'); ylabel('(m) or (-)'); hold off;
%              combinePlots([q(1,3),q(2,3),q(3,3)],'lin','log',legtext,styles);  hold on;
 xlabel('s (m)'); ylabel('(m) or (-)'); title('x=80mm'); hold off;
             legtext = {'x = 0 mm',...
            'x = 50 mm',...
            'x = 80 mm'};
             styles  = {'r-','g-','b-',};
             combinePlots([h(1,3),p(1,3),q(1,3)],'lin','log',legtext,styles);  hold on; x
label('s [mm]'); ylabel('$\Delta\ell_m/\bar{\ell}_m$'); hold on;
             xl = xline(430, '.-k',{'Magnet pole'});
             xl.LabelVerticalAlignment = 'middle';
             xl.LabelHorizontalAlignment = 'center';hold off;
        end;
```

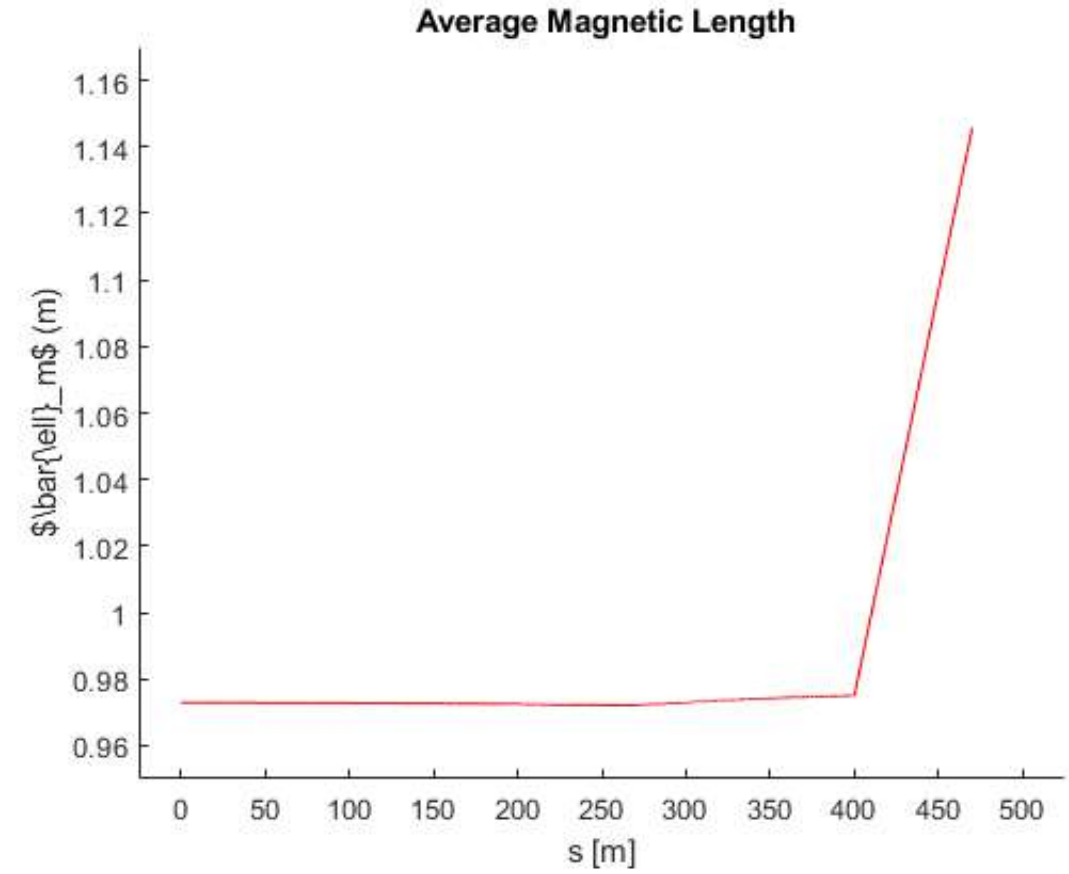
```
LOOP @ 5 mm; central:   s* = 340.150000,         Lm*=0.973611,   DeltaLm*=0.001281,
  DeltaLm_rel*=0.001316,         DeltaLm_rel0=0.011468,  improvement =8.716955
LOOP @ 16 mm, central:   s* = 342.400000,        Lm*=0.974067,   DeltaLm*=0.001621,
  DeltaLm_rel*=0.001664,         DeltaLm_rel0=0.011121,  improvement =6.681365
LOOP @ 25 mm, central:   s* = 347.900000,        Lm*=0.975223,   DeltaLm*=0.002956,
  DeltaLm_rel*=0.003031,         DeltaLm_rel0=0.010335,  improvement =3.409205
Warning: Error updating Text.
```

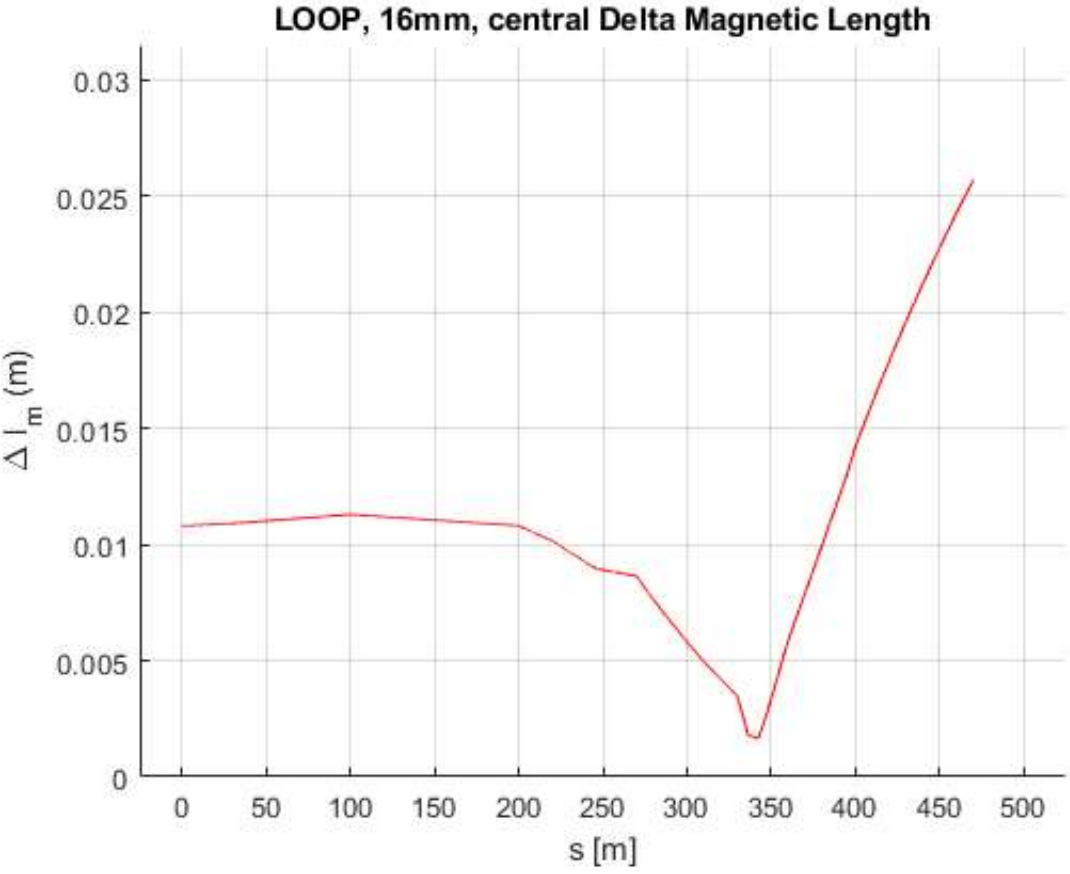
```
 String scalar or character vector must have valid interpreter syntax:
\Delta l_m/\bar{l}_m (-)

Warning: Error updating Text.

 String scalar or character vector must have valid interpreter syntax:
\Delta l_m/\bar{l}_m (-)

Warning: Error updating Text.

 String scalar or character vector must have valid interpreter syntax:
\Delta l_m/\bar{l}_m (-)

Warning: Error updating Text.

 String scalar or character vector must have valid interpreter syntax:
\Delta l_m/\bar{l}_m (-)

Warning: Marker input is ignored.
```

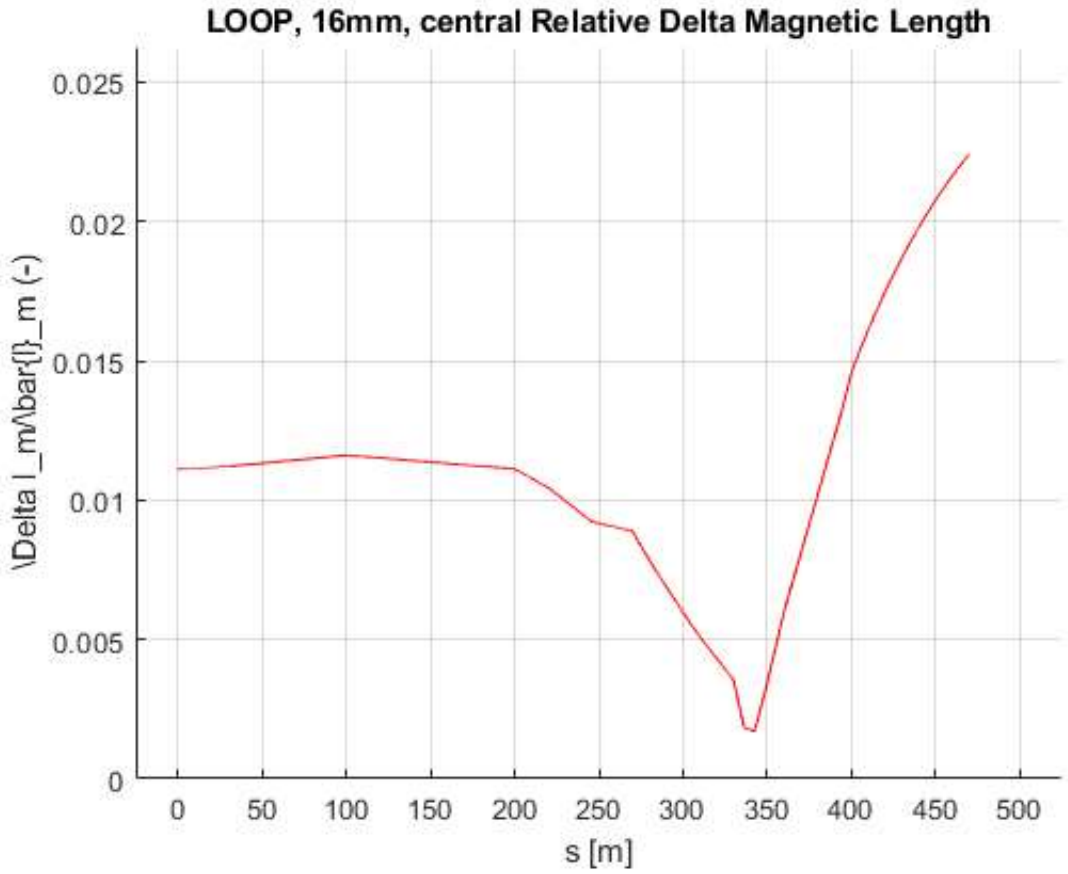


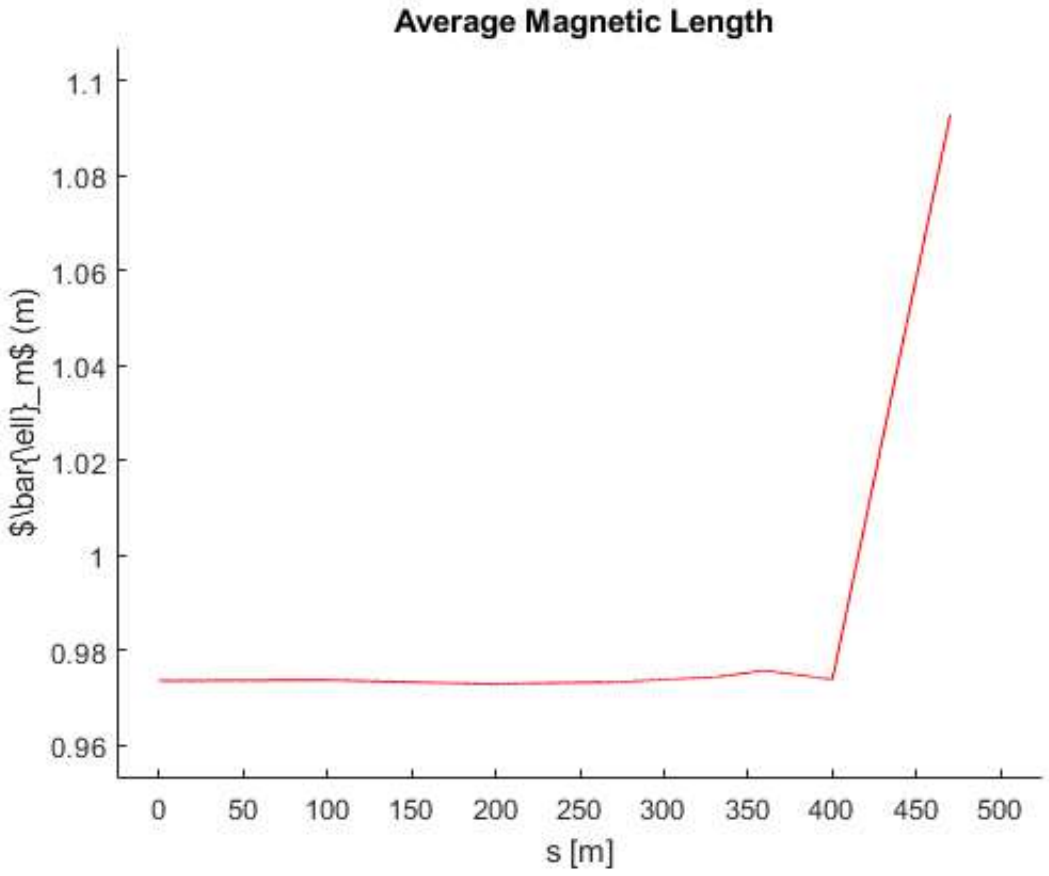

LOOP, 5mm, central Delta Magnetic Length
0.025
0.02
0.015
0.01
0.005
0
Δ l_m (m)
0
50
100
150
200
250
300
350
400
450
500
s [m]

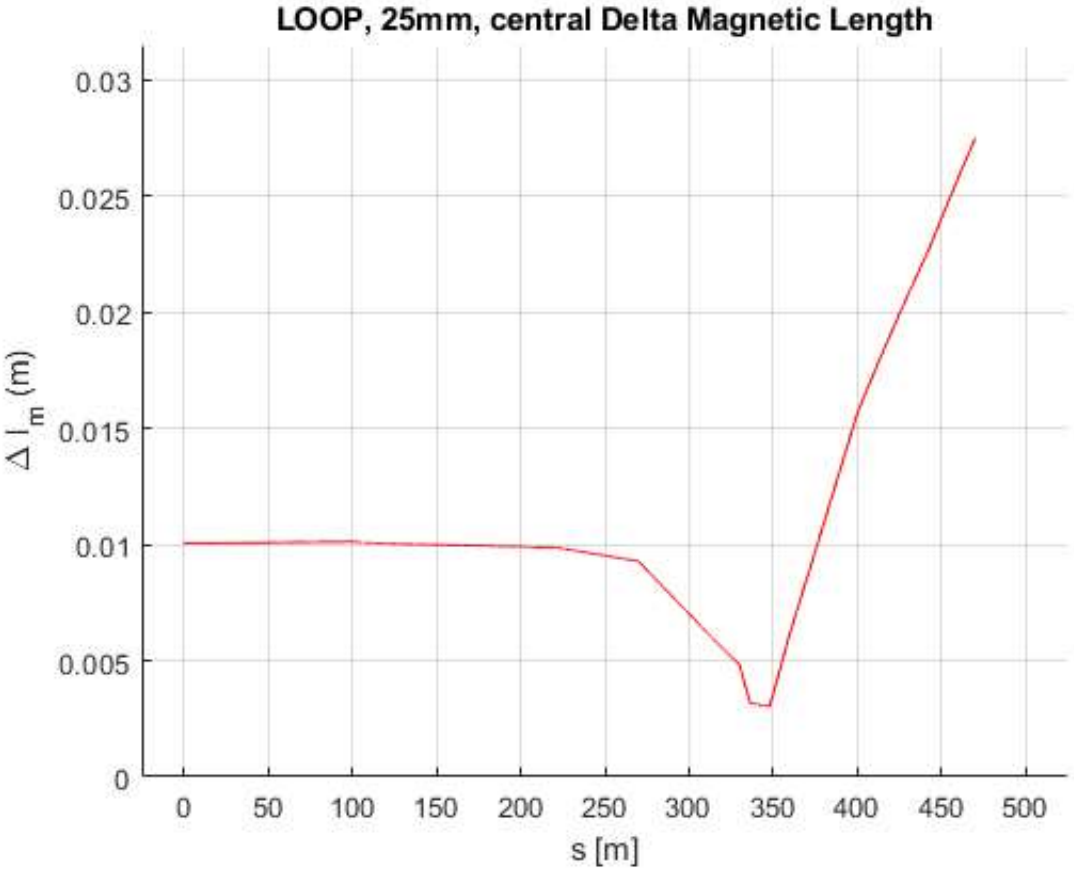

LOOP, 5mm, central Relative Delta Magnetic Length
0.02
0.018
0.016
0.014
0.012
0.01
0.008
0.006
0.004
0.002
0
\Delta l_m/\bar{l}_m (-)
0
50
100
150
200
250
300
350
400
450
500
s [m]

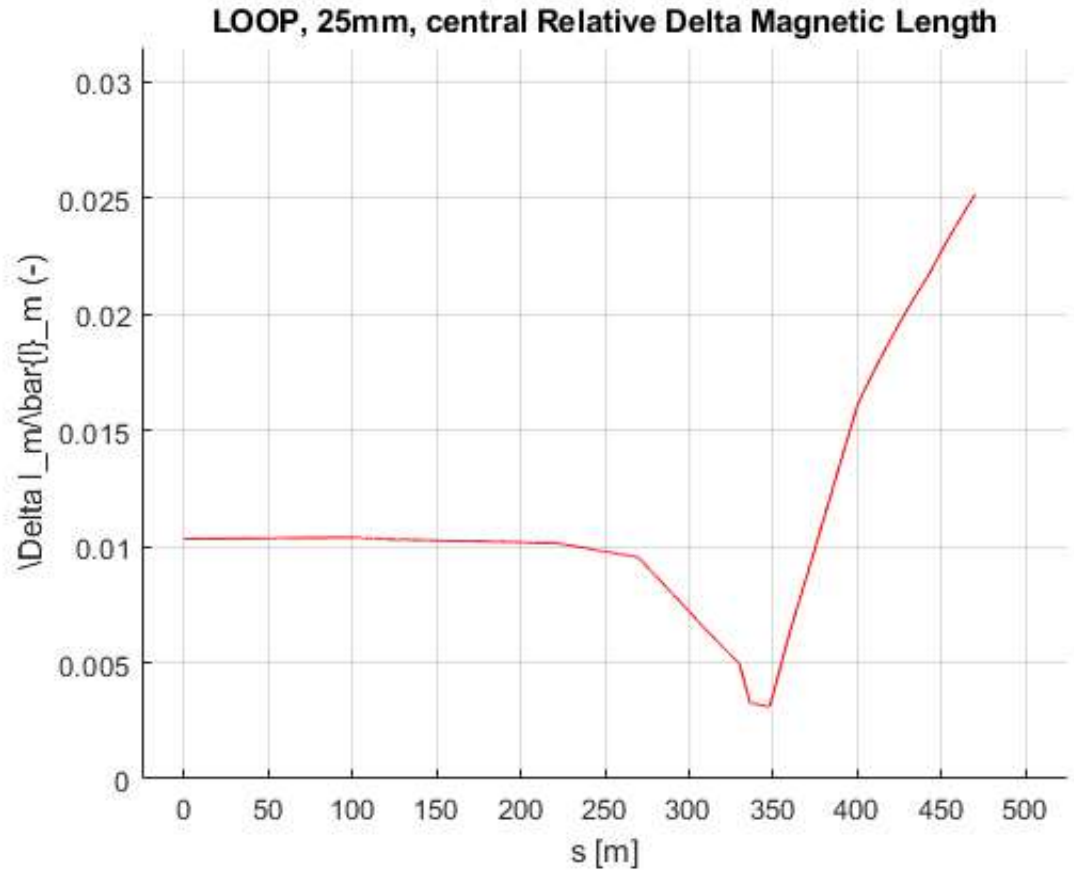

Average Magnetic Length
$\bar{\ell}_m$ (m)
1.16
1.14
1.12
1.1
1.08
1.06
1.04
1.02
1
0.98
0.96
0
50
100
150
200
250
300
350
400
450
500
s [m]

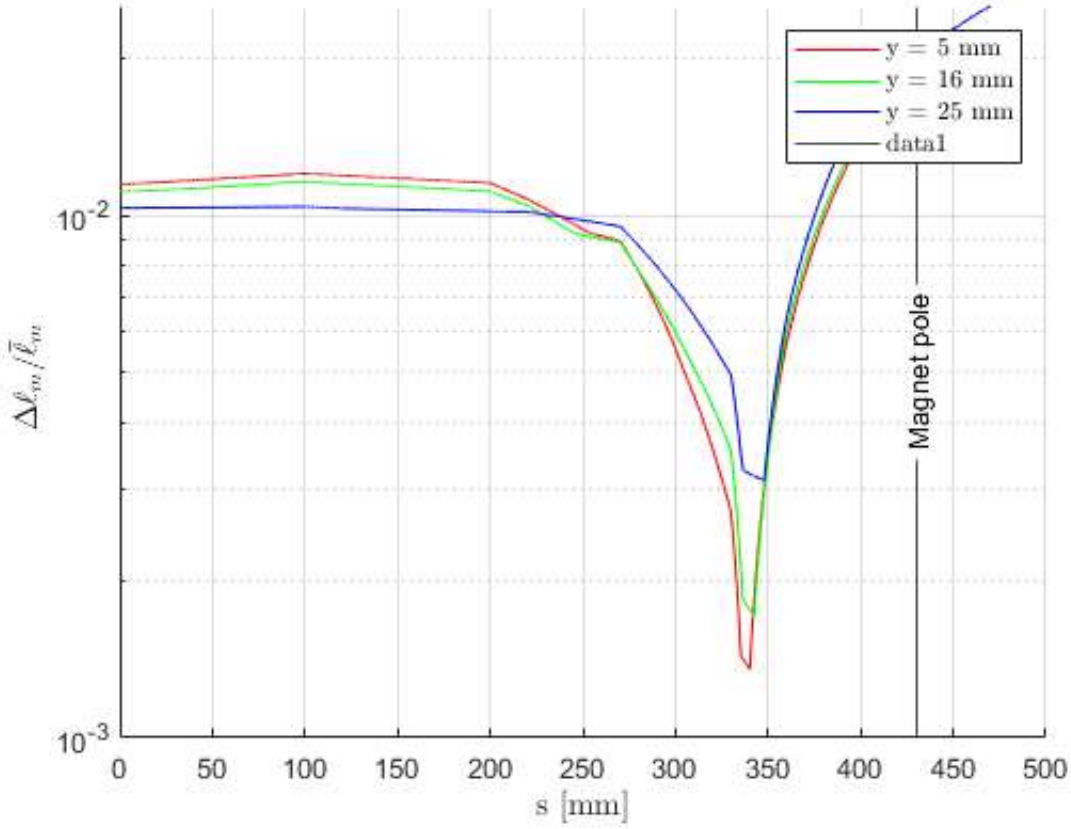

LOOP, 16mm, central Delta Magnetic Length
Δ l_m (m)
0.03
0.025
0.02
0.015
0.01
0.005
0
0
50
100
150
200
250
300
350
400
450
500
s [m]


LOOP, 16mm, central Relative Delta Magnetic Length
\Delta l_m/\bar{l}_m (-)
0.025
0.02
0.015
0.01
0.005
0
0
50
100
150
200
250
300
350
400
450
500
s [m]

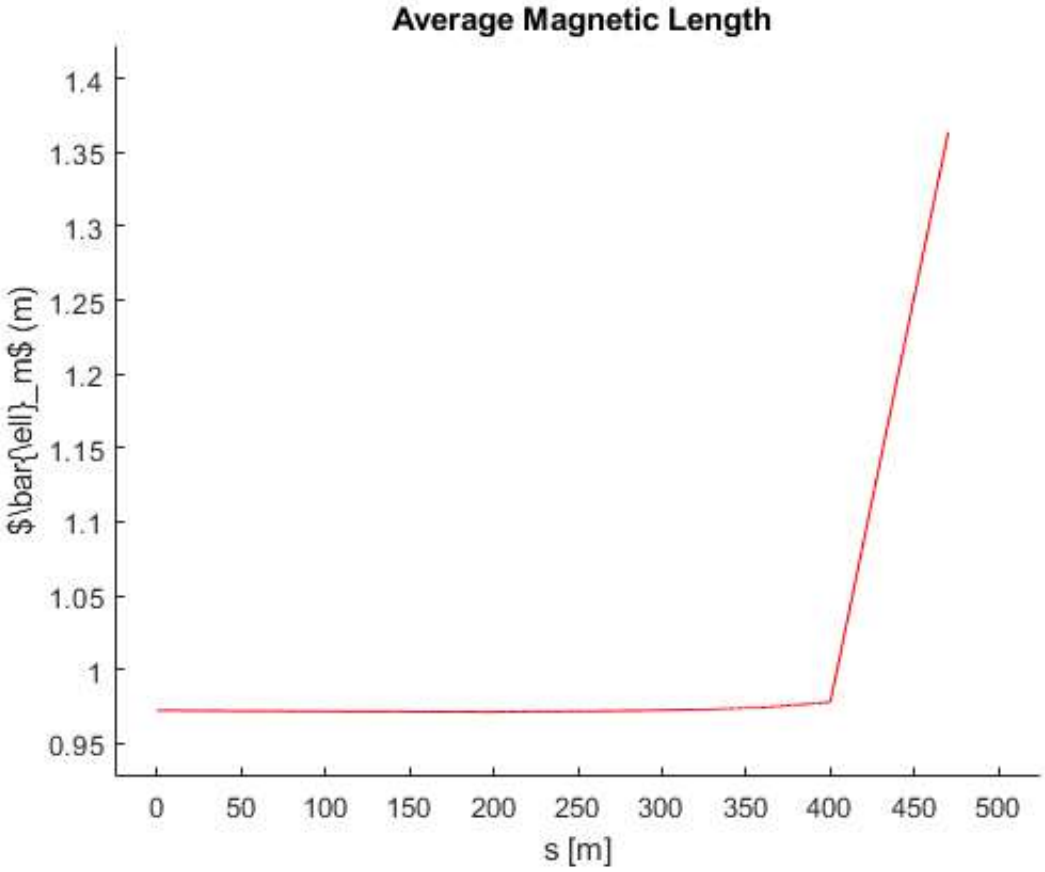

Average Magnetic Length
$\bar{\ell}_m$ (m)
1.1
1.08
1.06
1.04
1.02
1
0.98
0.96
0
50
100
150
200
250
300
350
400
450
500
s [m]

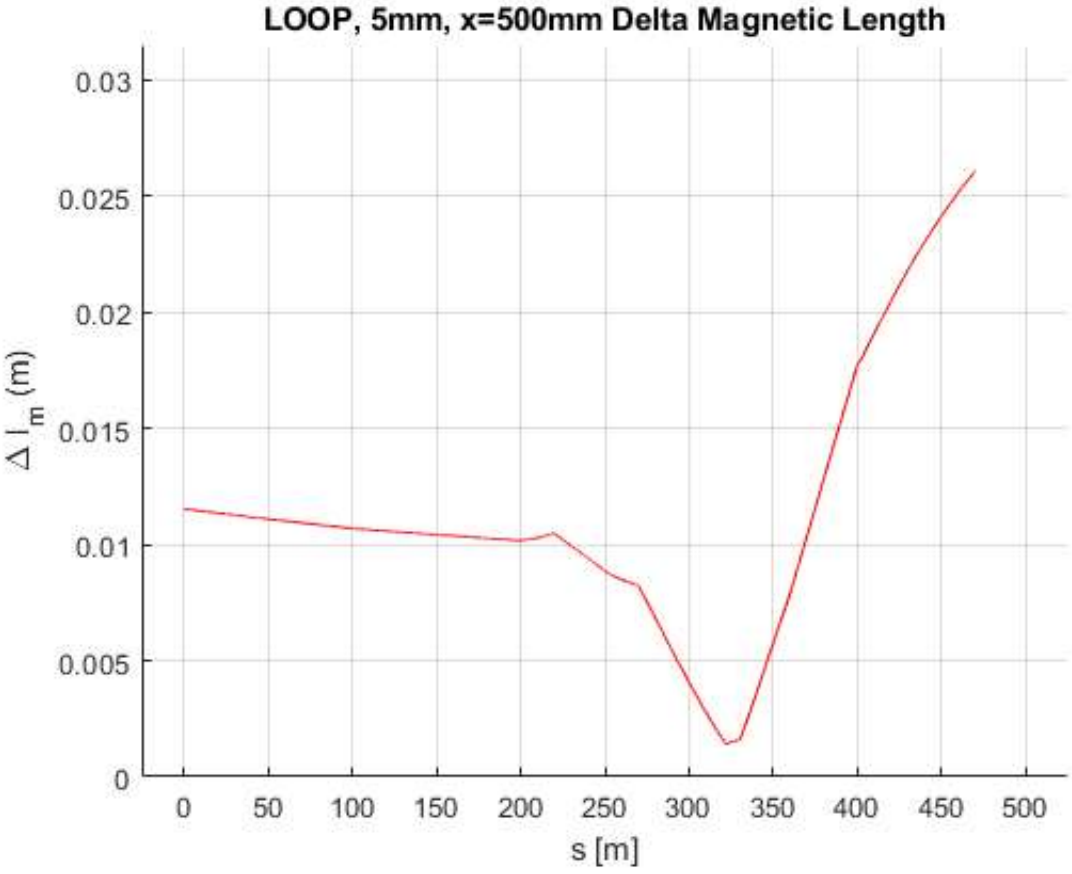

LOOP, 25mm, central Delta Magnetic Length
0.03
0.025
0.02
0.015
0.01
0.005
0
Δ l_m (m)
0
50
100
150
200
250
300
350
400
450
500
s [m]

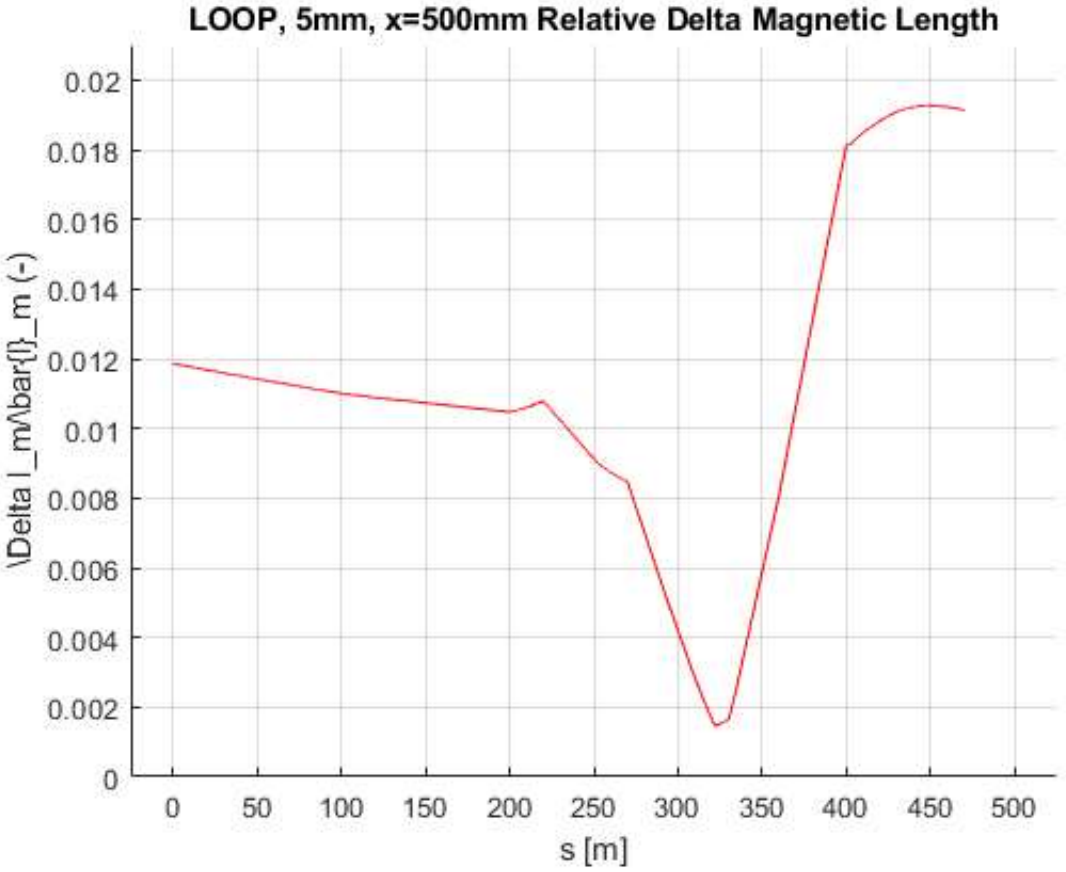

LOOP, 25mm, central Relative Delta Magnetic Length
0.03
0.025
0.02
0.015
0.01
0.005
0
\Delta l_m/\bar{l}_m (-)
0
50
100
150
200
250
300
350
400
450
500
s [m]

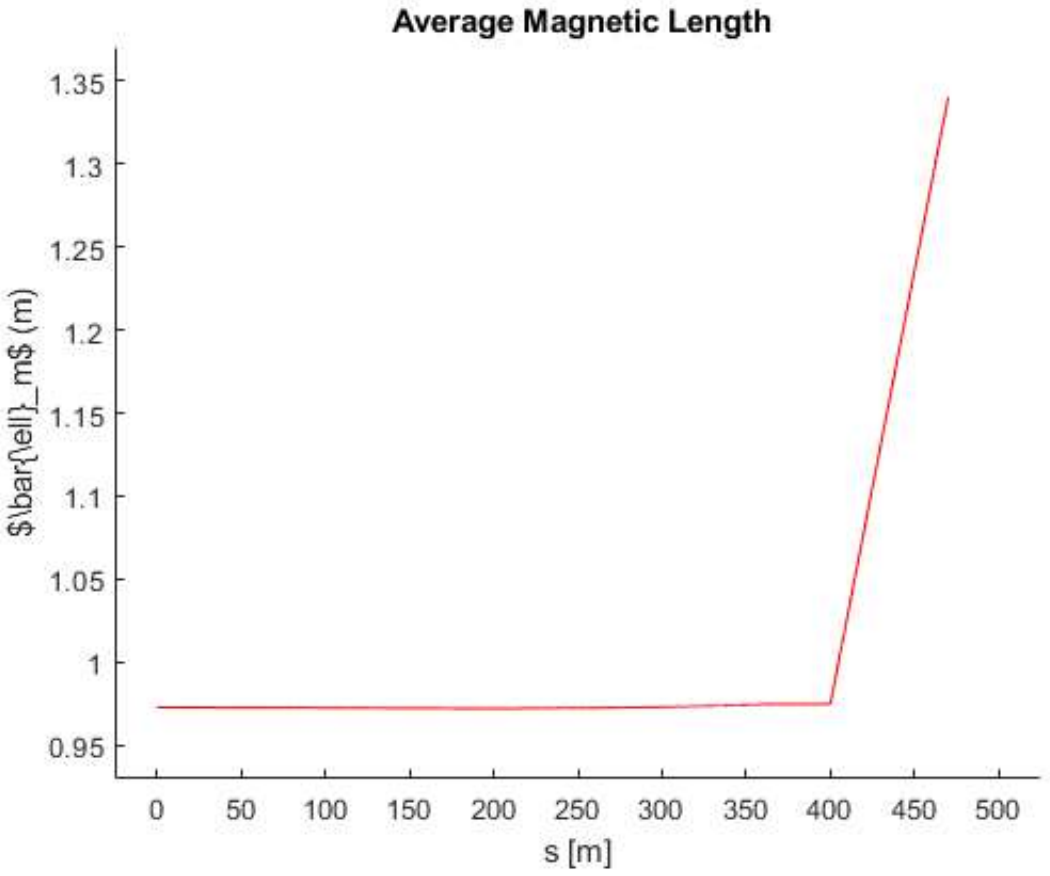


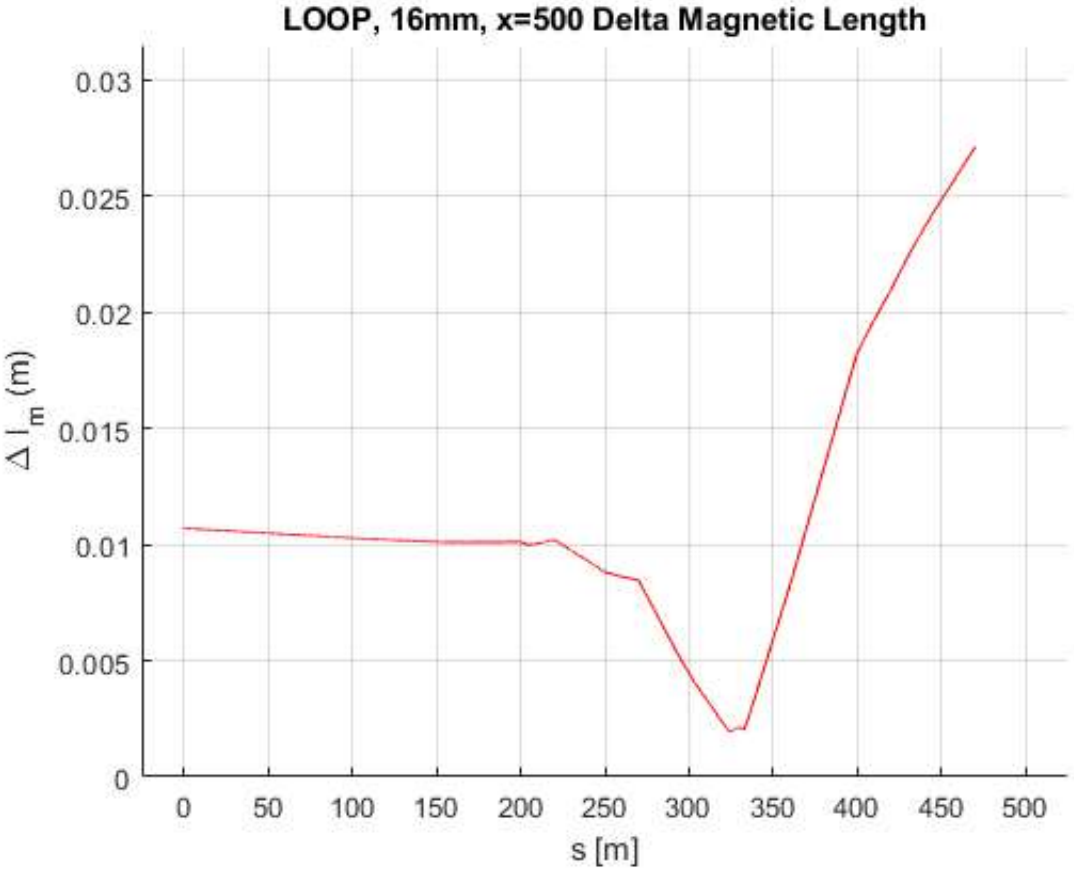

```
LOOP @ 5 mm; x=500:   s* = 321.850000,   Lm*=0.973022,   DeltaLm*=0.001381,       DeltaL
m_rel*=0.001420,         DeltaLm_rel0=0.011858,  improvement =8.353030
LOOP @ 16 mm, x=500:   s* = 324.150000,  Lm*=0.973692,   DeltaLm*=0.001898,       DeltaL
m_rel*=0.001949,         DeltaLm_rel0=0.010998,  improvement =5.643024
LOOP @ 25 mm, x=500:   s* = 337.800000,  Lm*=0.975966,   DeltaLm*=0.003108,       DeltaL
m_rel*=0.003185,         DeltaLm_rel0=0.010202,  improvement =3.203537
Warning: Error updating Text.

 String scalar or character vector must have valid interpreter syntax:
\Delta l_m/\bar{l}_m (-)

Warning: Error updating Text.

 String scalar or character vector must have valid interpreter syntax:
\Delta l_m/\bar{l}_m (-)

Warning: Error updating Text.

 String scalar or character vector must have valid interpreter syntax:
\Delta l_m/\bar{l}_m (-)

Warning: Error updating Text.

 String scalar or character vector must have valid interpreter syntax:
\Delta l_m/\bar{l}_m (-)

Warning: Marker input is ignored.
```

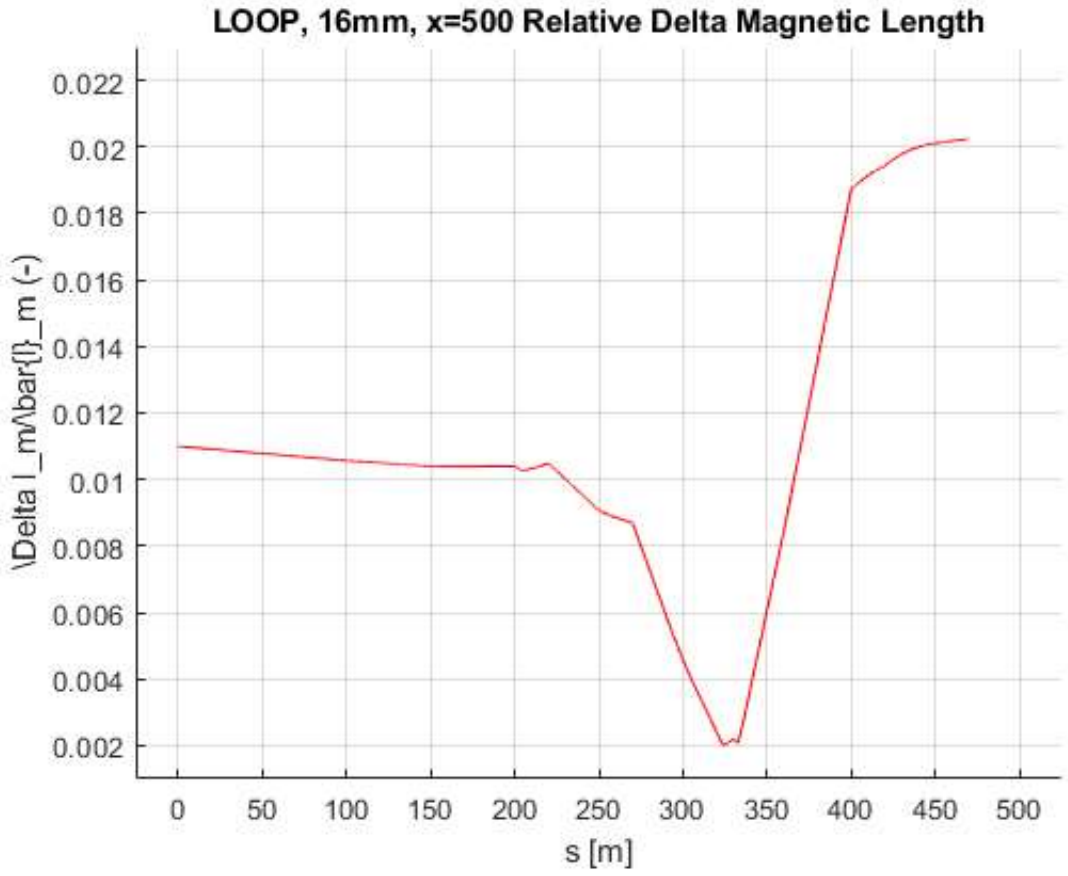

Average Magnetic Length
$\bar{\ell}_m$ (m)
s [m]
1.4
1.35
1.3
1.25
1.2
1.15
1.1
1.05
1
0.95
0
50
100
150
200
250
300
350
400
450
500

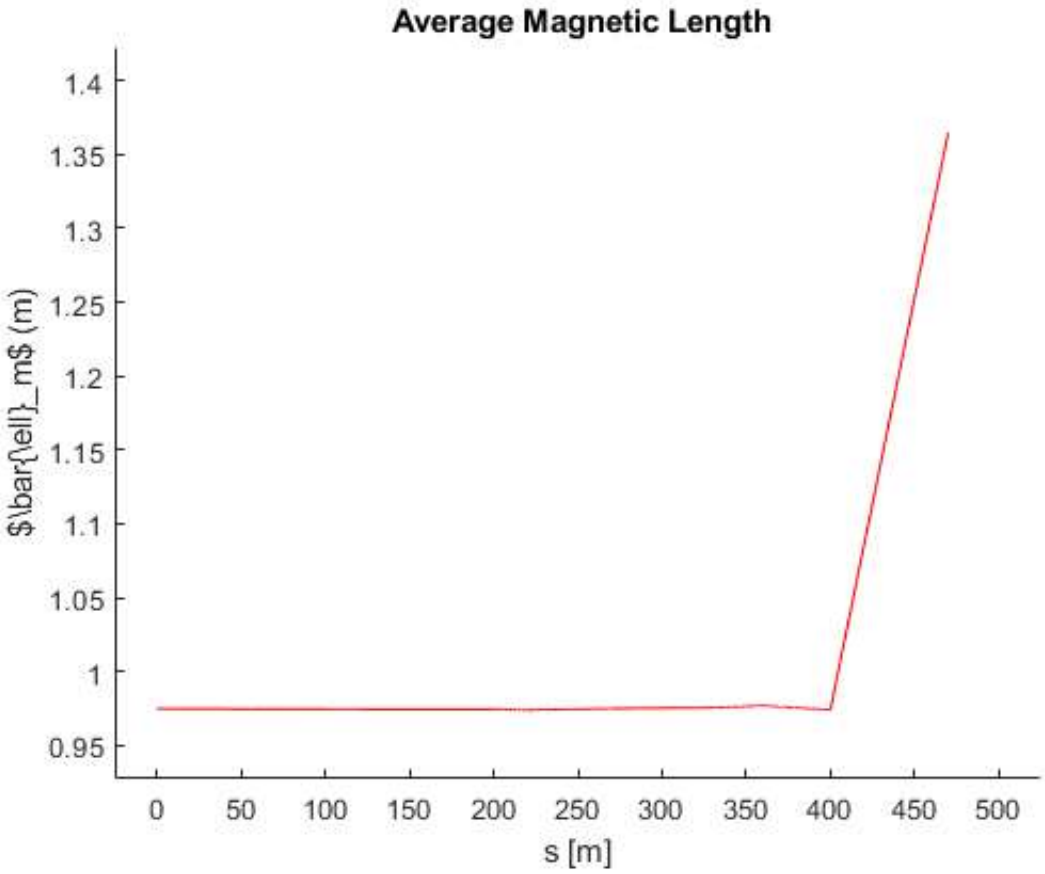

LOOP, 5mm, x=500mm Delta Magnetic Length
Δ l_m (m)
s [m]
0.03
0.025
0.02
0.015
0.01
0.005
0
0
50
100
150
200
250
300
350
400
450
500

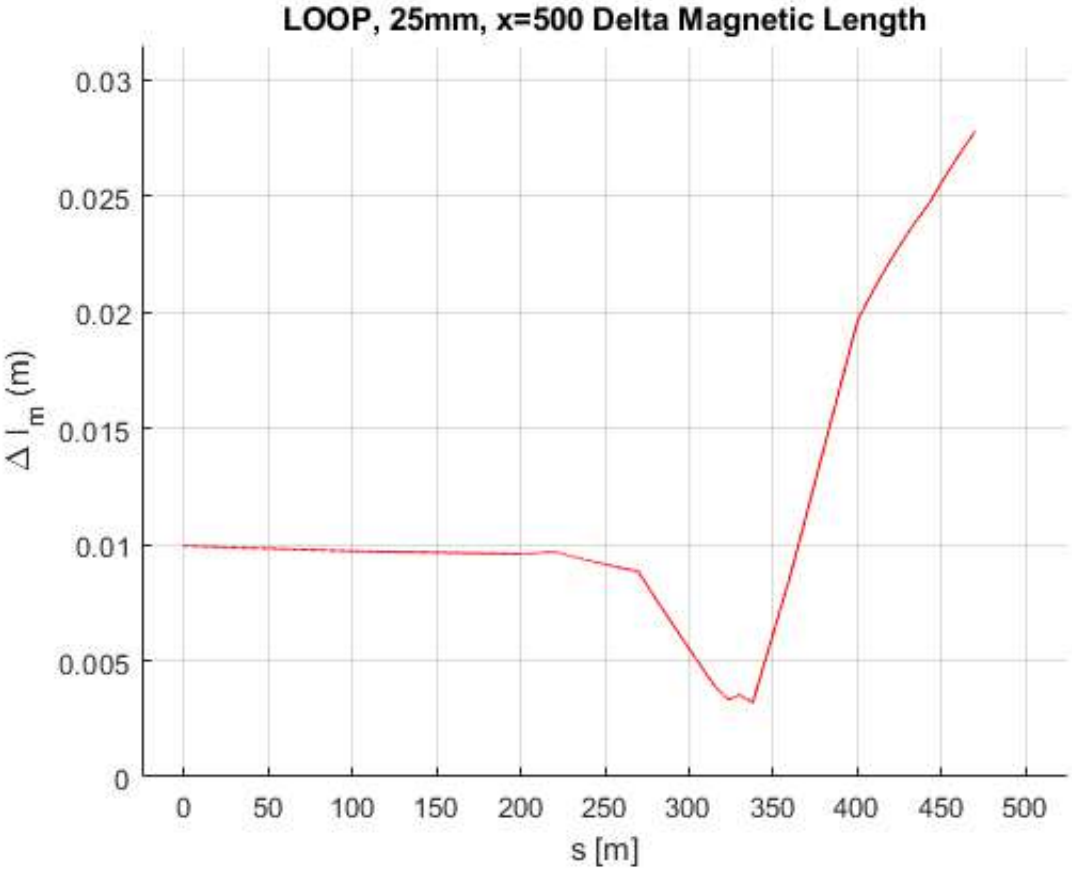

LOOP, 5mm, x=500mm Relative Delta Magnetic Length
\Delta l_m/\bar{l}_m (-)
0.02
0.018
0.016
0.014
0.012
0.01
0.008
0.006
0.004
0.002
0
0
50
100
150
200
250
300
350
400
450
500
s [m]

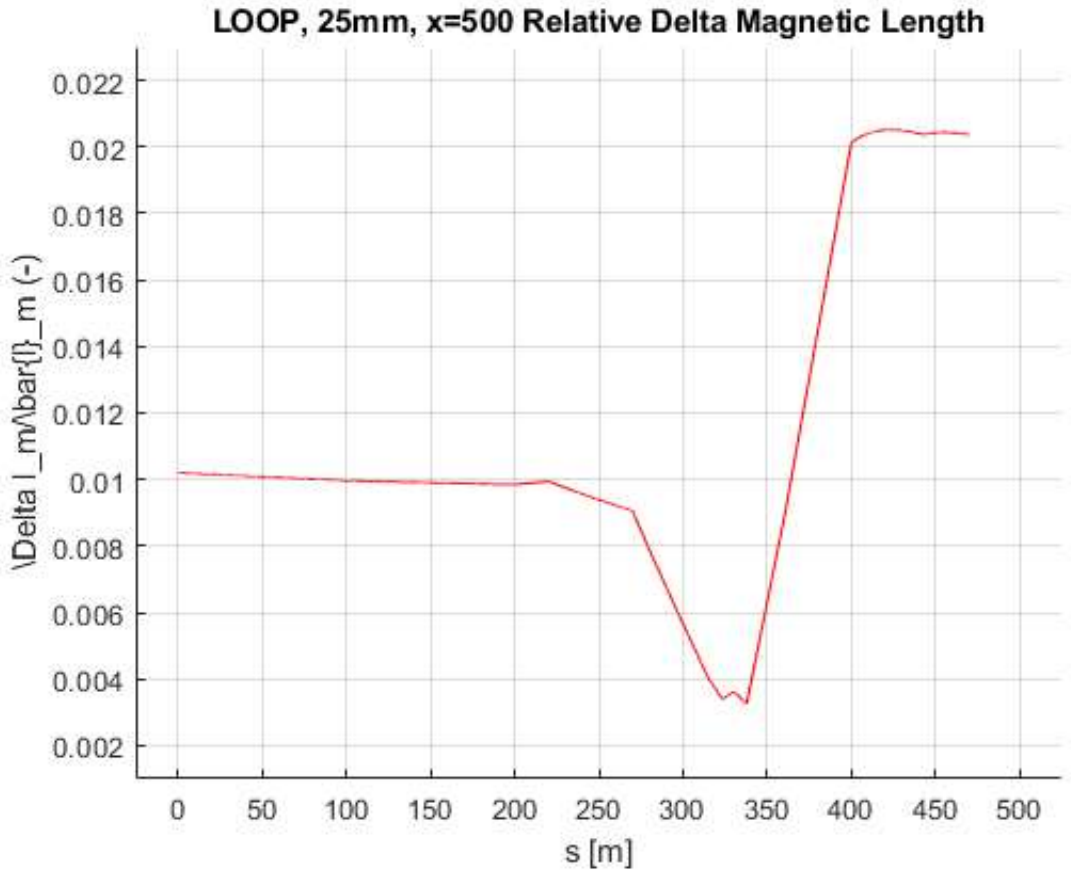

Average Magnetic Length
$\bar{\ell}_m$ (m)
1.35
1.3
1.25
1.2
1.15
1.1
1.05
1
0.95
0
50
100
150
200
250
300
350
400
450
500
s [m]

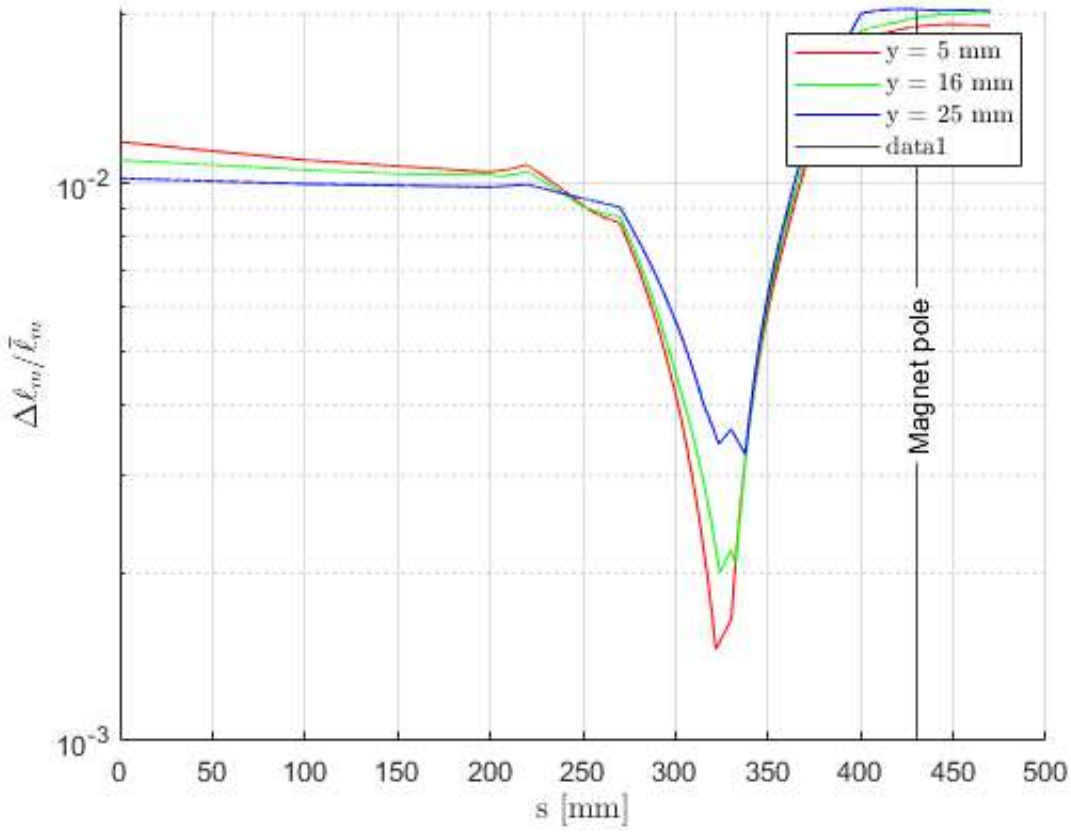

LOOP, 16mm, x=500 Delta Magnetic Length
0.03
0.025
0.02
0.015
0.01
0.005
0
Δ l_m (m)
0
50
100
150
200
250
300
350
400
450
500
s [m]


LOOP, 16mm, x=500 Relative Delta Magnetic Length
0.022
0.02
0.018
0.016
0.014
0.012
0.01
0.008
0.006
0.004
0.002
\Delta l_m/\bar{l}_m (-)
0
50
100
150
200
250
300
350
400
450
500
s [m]


Average Magnetic Length
$\bar{\ell}_m$ (m)
1.4
1.35
1.3
1.25
1.2
1.15
1.1
1.05
1
0.95
0
50
100
150
200
250
300
350
400
450
500
s [m]

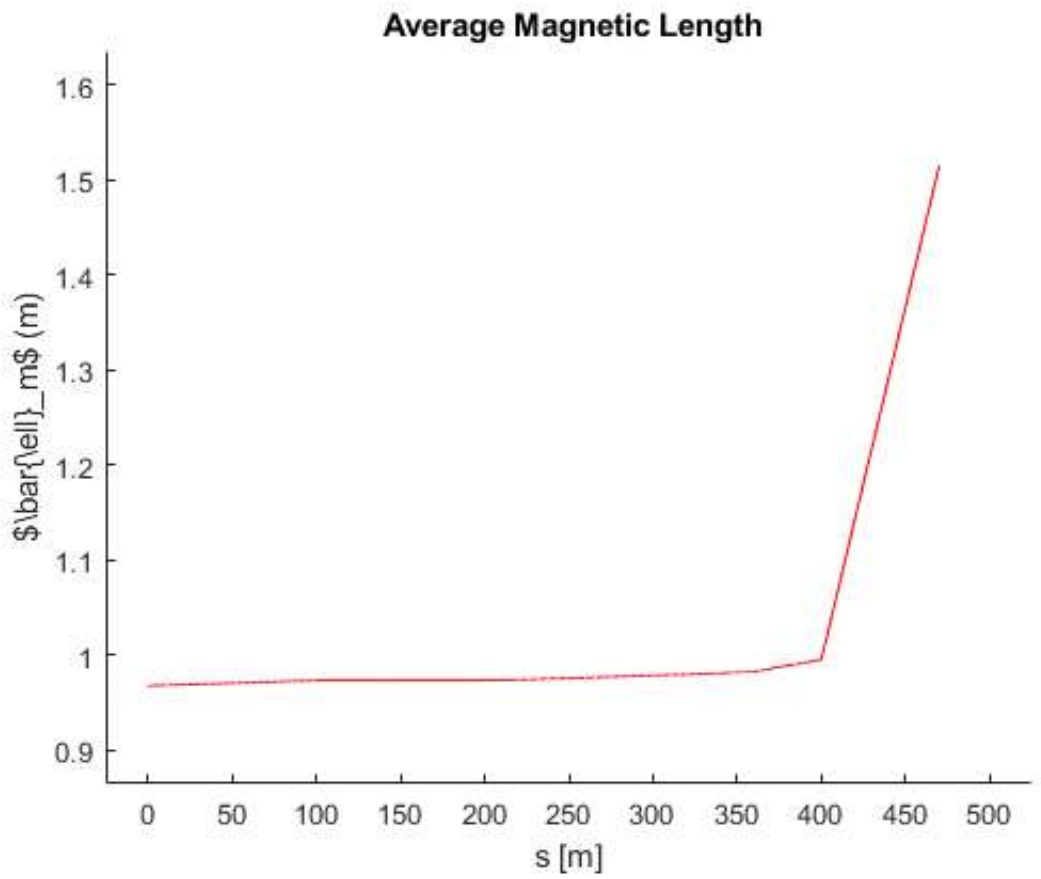

LOOP, 25mm, x=500 Delta Magnetic Length
Δ l_m (m)
0.03
0.025
0.02
0.015
0.01
0.005
0
0
50
100
150
200
250
300
350
400
450
500
s [m]

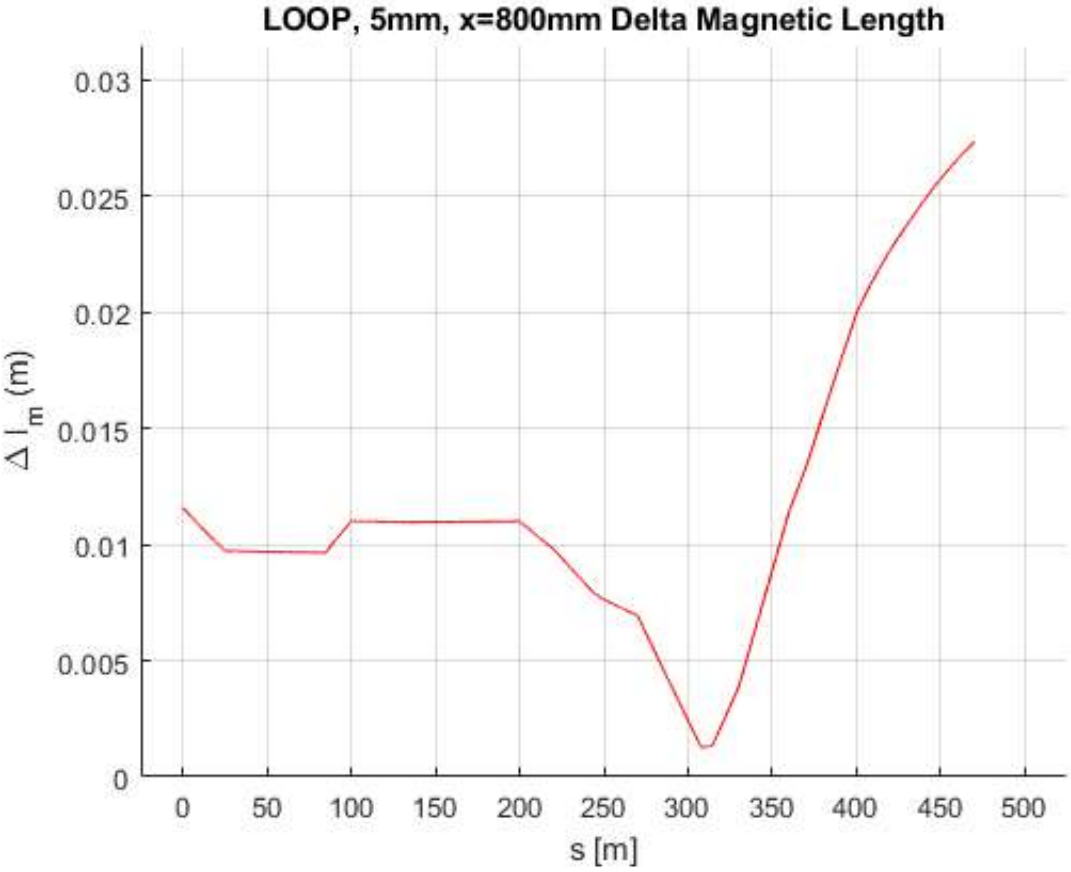


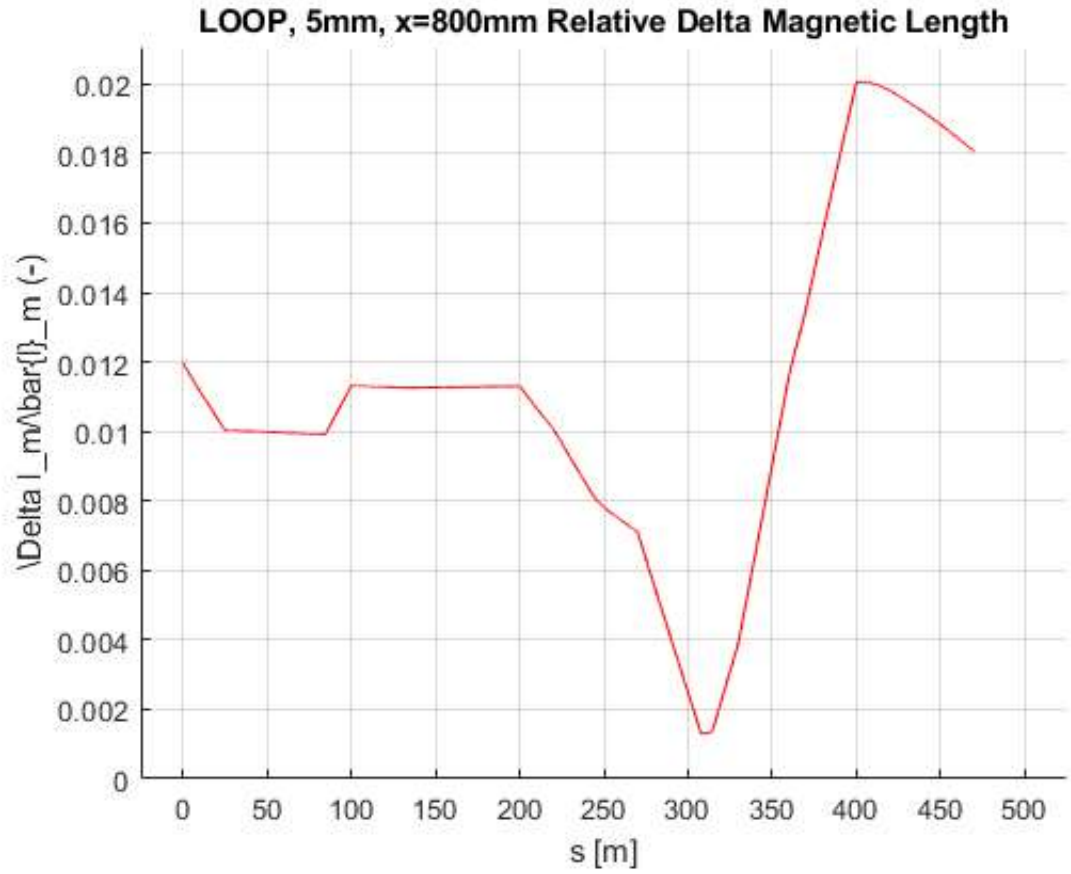


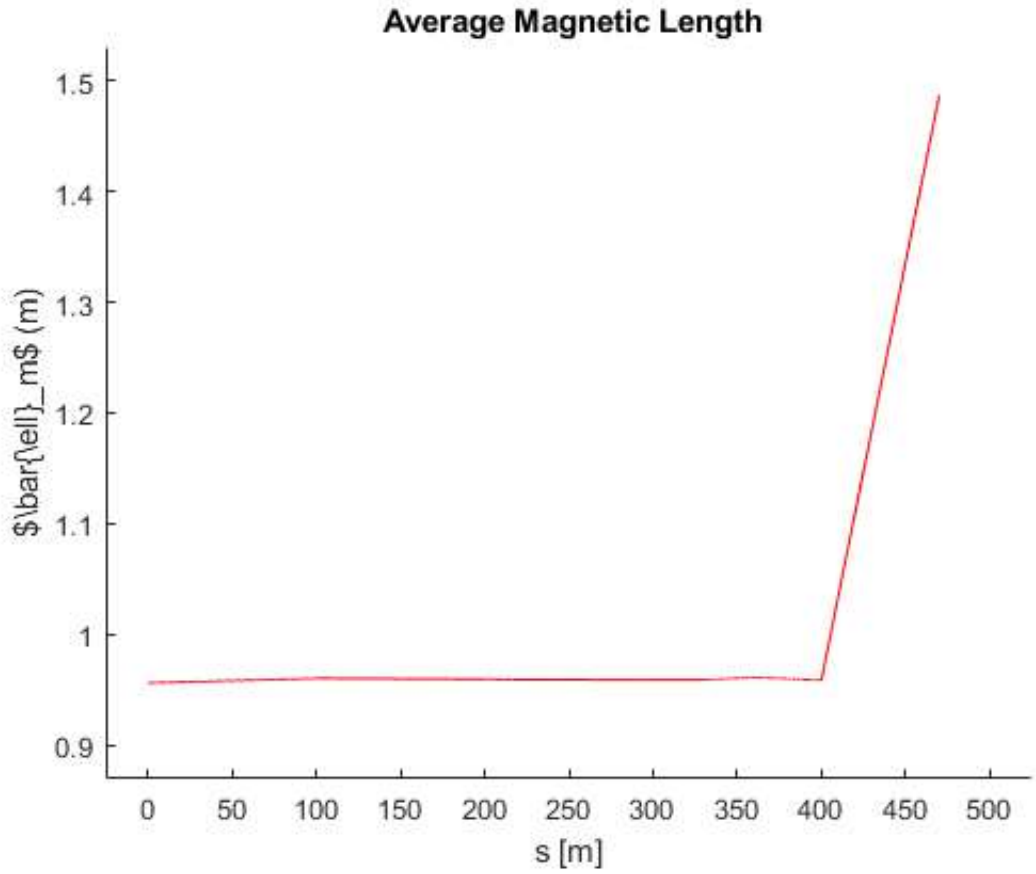

```
LOOP @ 5 mm; x=800:    s* = 307.800000,   Lm*=0.979326,   DeltaLm*=0.001244,       DeltaL
m_rel*=0.001270,          DeltaLm_rel0=0.011968,  improvement =9.421122
LOOP @ 16 mm, x=800:    s* = 307.700000,  Lm*=0.959302,   DeltaLm*=0.001597,       DeltaL
m_rel*=0.001665,          DeltaLm_rel0=0.010076,  improvement =6.052683
LOOP @ 25 mm, x=800:    s* = 306.050000,  Lm*=0.938340,   DeltaLm*=0.002546,       DeltaL
```

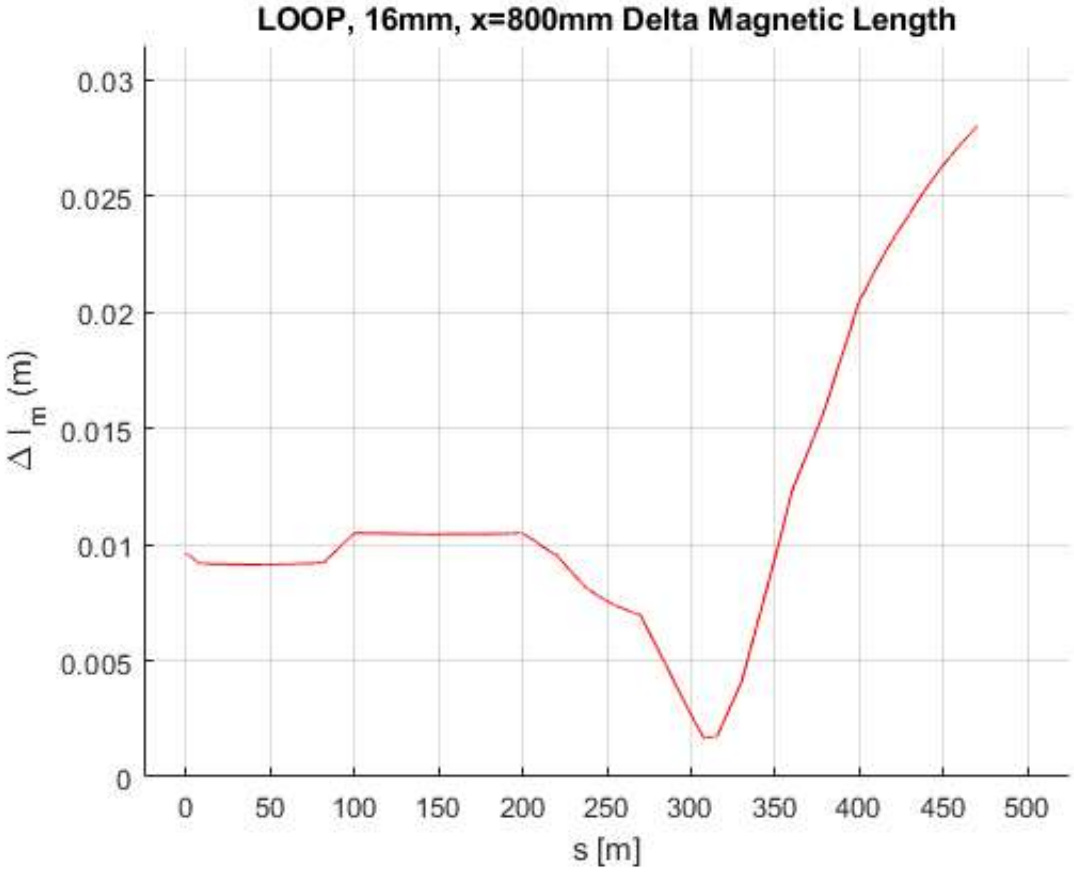
```
m_rel*=0.002713,          DeltaLm_rel0=0.010541,  improvement =3.884770
Warning: Error updating Text.

 String scalar or character vector must have valid interpreter syntax:
\Delta l_m/\bar{l}_m (-)

Warning: Error updating Text.

 String scalar or character vector must have valid interpreter syntax:
\Delta l_m/\bar{l}_m (-)

Warning: Marker input is ignored.
```

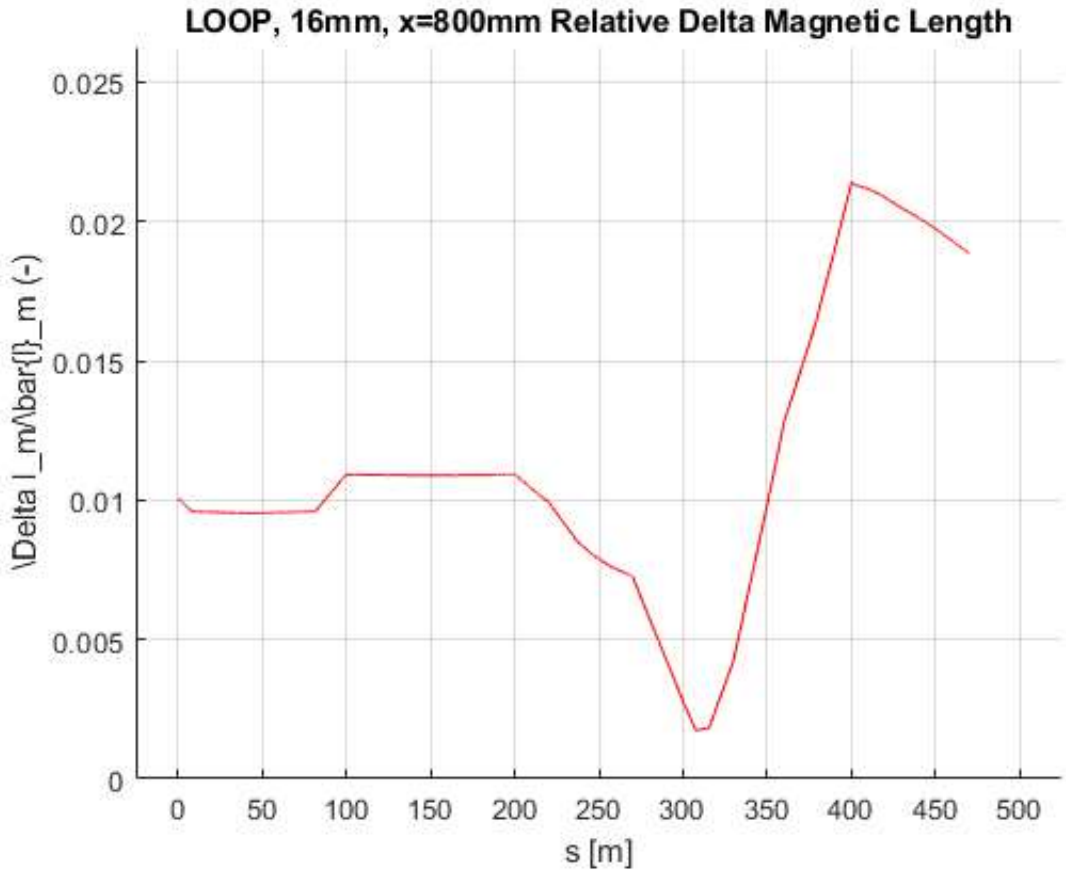


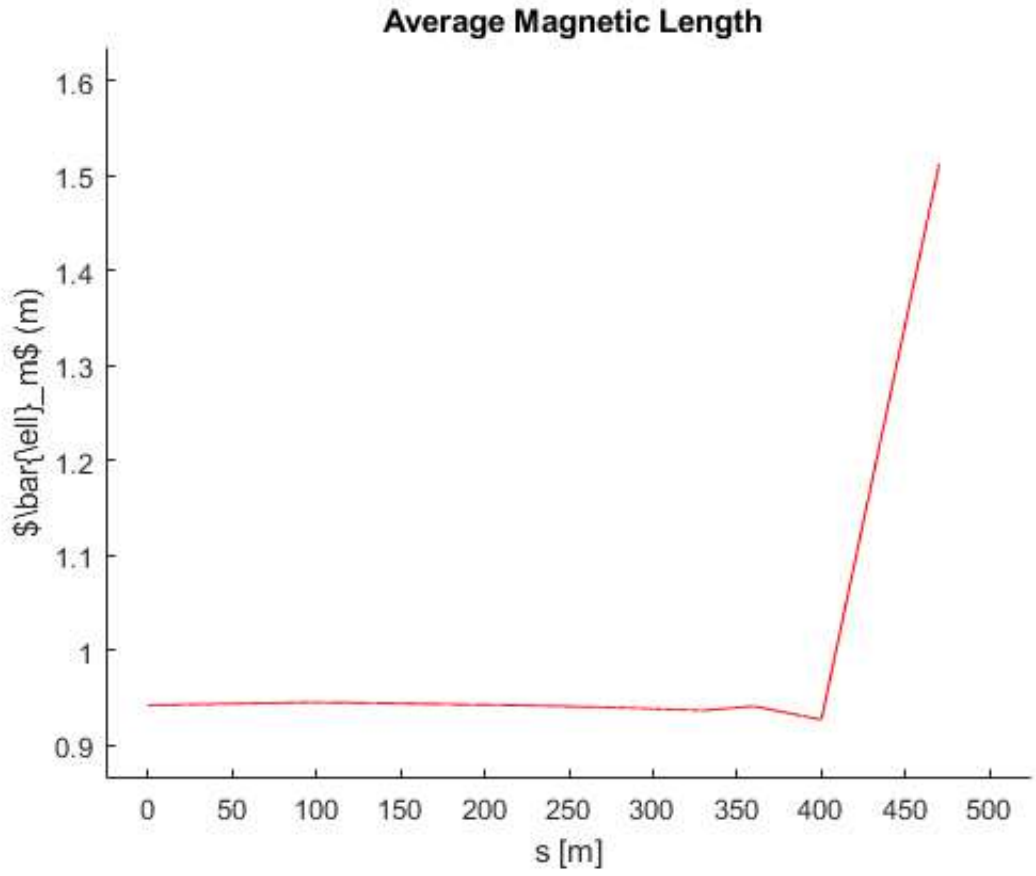

LOOP, 5mm, x=800mm Delta Magnetic Length
0.03
0.025
0.02
0.015
0.01
0.005
0
Δ l_m (m)
0
50
100
150
200
250
300
350
400
450
500
s [m]

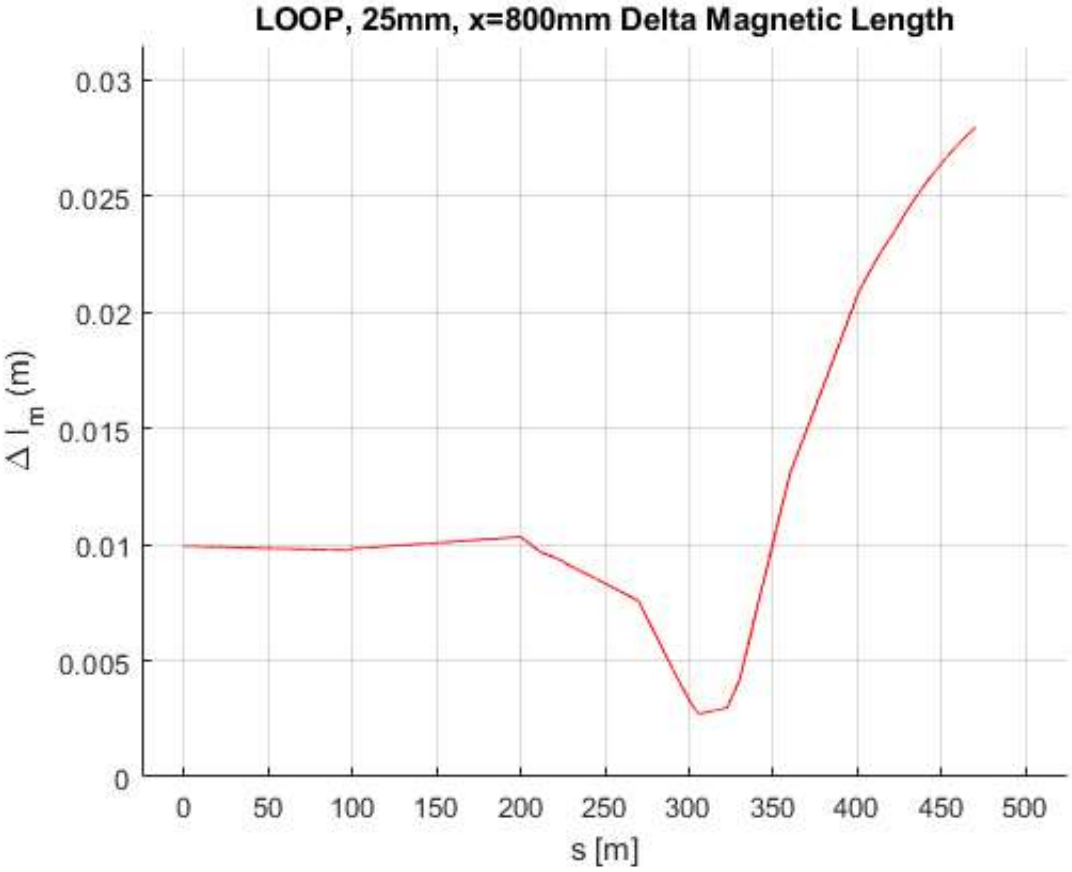

LOOP, 5mm, x=800mm Relative Delta Magnetic Length
0.02
0.018
0.016
0.014
0.012
0.01
0.008
0.006
0.004
0.002
0
\Delta l_m/\bar{l}_m (-)
0
50
100
150
200
250
300
350
400
450
500
s [m]

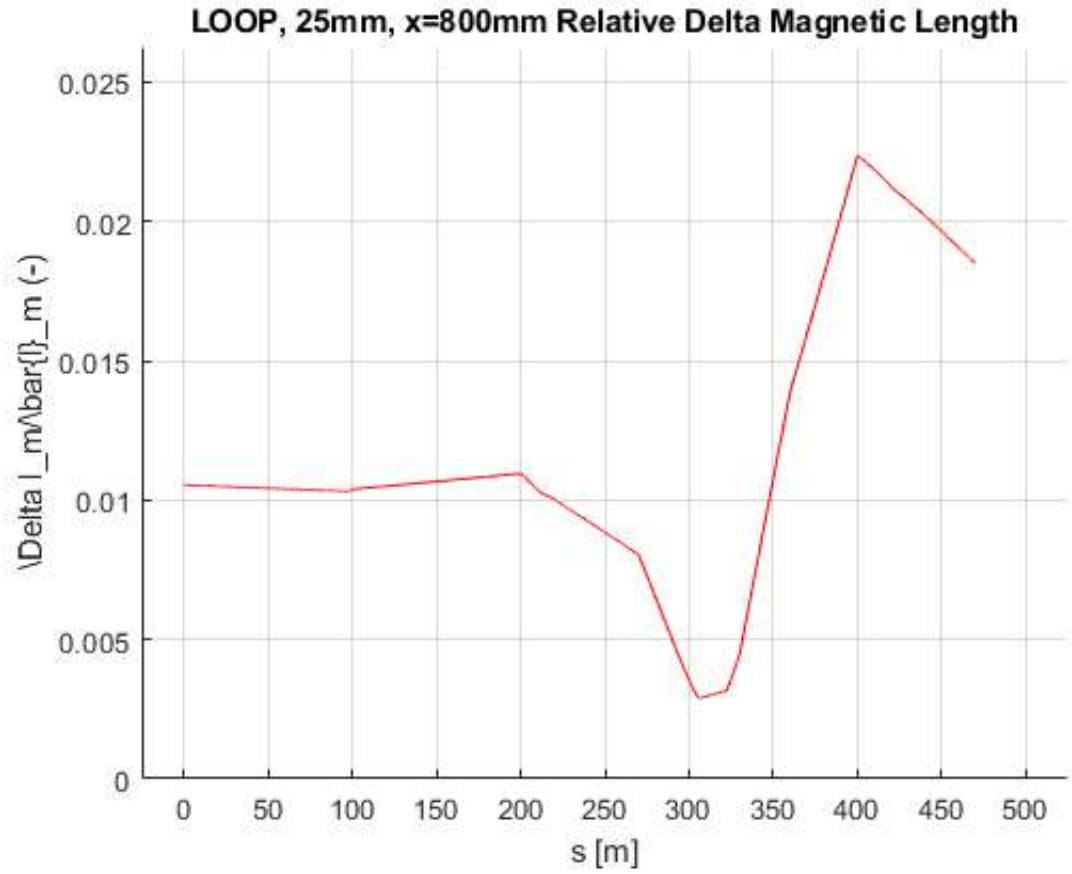

Average Magnetic Length
$\bar{\ell}_m$ (m)
s [m]
1.5
1.4
1.3
1.2
1.1
1
0.9
0
50
100
150
200
250
300
350
400
450
500

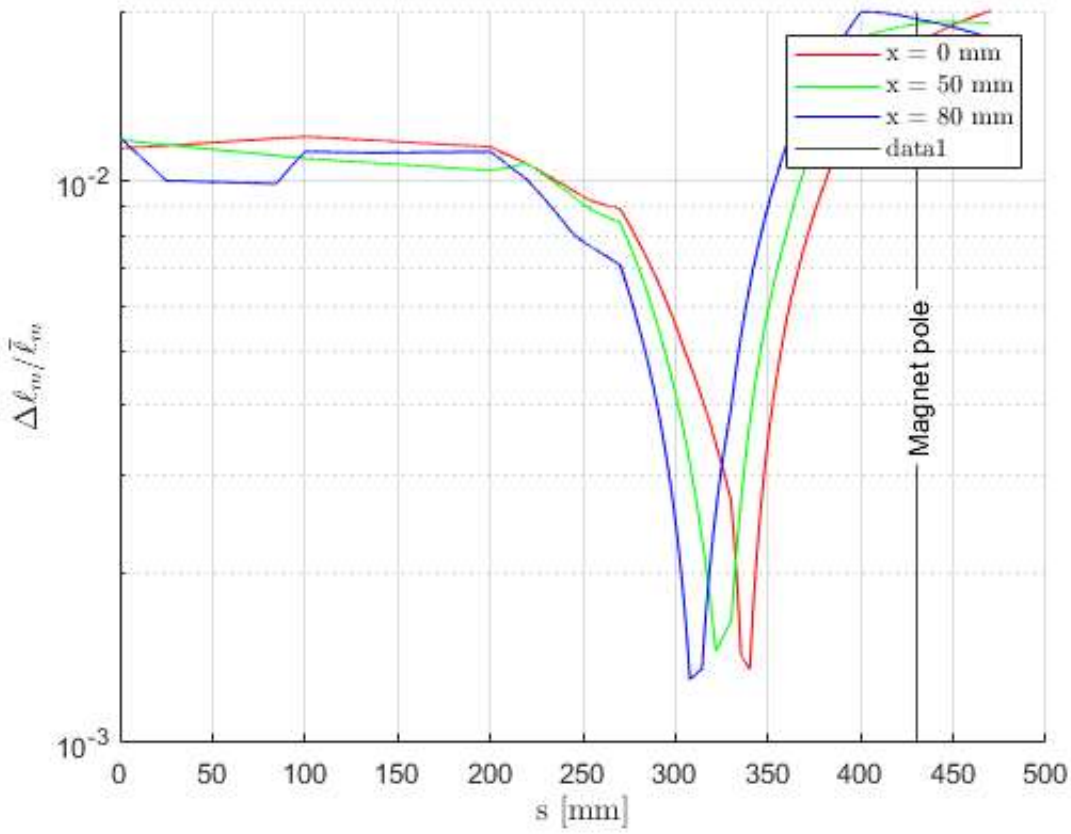

LOOP, 16mm, x=800mm Delta Magnetic Length
Δ l_m (m)
s [m]
0.03
0.025
0.02
0.015
0.01
0.005
0
0
50
100
150
200
250
300
350
400
450
500


LOOP, 16mm, x=800mm Relative Delta Magnetic Length
\Delta l_m/\bar{l}_m (-)
0.025
0.02
0.015
0.01
0.005
0
0
50
100
150
200
250
300
350
400
450
500
s [m]


Average Magnetic Length
$\bar{\ell}_m$ (m)
1.6
1.5
1.4
1.3
1.2
1.1
1
0.9
0
50
100
150
200
250
300
350
400
450
500
s [m]


LOOP, 25mm, x=800mm Delta Magnetic Length
0.03
0.025
0.02
0.015
0.01
0.005
0
Δ l_m (m)
0
50
100
150
200
250
300
350
400
450
500
s [m]


LOOP, 25mm, x=800mm Relative Delta Magnetic Length
0.025
0.02
0.015
0.01
0.005
0
\Delta l_m/\bar{l}_m (-)
0
50
100
150
200
250
300
350
400
450
500
s [m]




```
end;
```

```
Warning: Error updating Text.

 String scalar or character vector must have valid interpreter syntax:
$\bar{\ell}_m$ (m)

Warning: Error updating Text.

 String scalar or character vector must have valid interpreter syntax:
$\bar{\ell}_m$ (m)

Warning: Error updating Text.

 String scalar or character vector must have valid interpreter syntax:
$\bar{\ell}_m$ (m)

Warning: Error updating Text.

 String scalar or character vector must have valid interpreter syntax:
$\bar{\ell}_m$ (m)

Warning: Error updating Text.

 String scalar or character vector must have valid interpreter syntax:
$\bar{\ell}_m$ (m)

Warning: Error updating Text.

 String scalar or character vector must have valid interpreter syntax:
$\bar{\ell}_m$ (m)
```

```
Warning: Error updating Text.

 String scalar or character vector must have valid interpreter syntax:
$\bar{\ell}_m$ (m)

Warning: Error updating Text.

 String scalar or character vector must have valid interpreter syntax:
$\bar{\ell}_m$ (m)

Warning: Error updating Text.

 String scalar or character vector must have valid interpreter syntax:
\Delta l_m/\bar{l}_m (-)

Warning: Error updating Text.

 String scalar or character vector must have valid interpreter syntax:
$\bar{\ell}_m$ (m)

Warning: Error updating Text.

 String scalar or character vector must have valid interpreter syntax:
\Delta l_m/\bar{l}_m (-)
```

Published with MATLAB® R2018b