**Appendix II. Matlab script for minifilament analysis**

ROIS=66 % Note, the first ROI correspond to a line ROI (freehand, polygonal or straight) of the entire junction. The rest of the ROIS (from 2 to N) correspond to bipolar segments.

Rotate45=0; %If junction is too vertical, change this to 1, so then the %image is rotated and analysis can be performed successfully.

%% Import data from text file.

for i=1:ROIS

filename = strcat(sprintf('%1d',i),'.txt')

delimiter = '\t';

%% Format string for each line of text:

% column1: double (%f)

% column2: double (%f)

% For more information, see the TEXTSCAN documentation.

formatSpec = '%f%f%[^\n\r]';

%% Open the text file.

fileID = fopen(filename,'r');

%% Read columns of data according to format string.

dataArray = textscan(fileID, formatSpec, 'Delimiter', delimiter, 'ReturnOnError', false);

%% Close the text file.

fclose(fileID);

%% Allocate imported array to column variable names

VarName(:,1) = dataArray{:, 1};

VarName(:,2) = dataArray{:, 2};

if Rotate45==1;

 xtest(:,1)=VarName(:,1);

 ytest(:,1)=VarName(:,2);

 VarNametest(:,1)=xtest(:,1)-ytest(:,1)\*sin(pi/4);

 VarNametest(:,2)=xtest(:,1)\*sin(pi/4)+ytest(:,1);

 VarName(:,:)=VarNametest(:,:);

 clearvars xtest ytest VarNametest

end

cfs=sprintf('ROI%1d',i);

cfv=genvarname(cfs);

eval([cfv '=VarName;']);

clearvars VarName

end

%% Derivative function calculation and Clear temporary variables

JunctionLengthPx=size(ROI1,1);

if ROI1(JunctionLengthPx,1)-ROI1(1,1)<0

 ROI1corr=ROI1;

 for j=0:JunctionLengthPx-1

 ROI1corr(JunctionLengthPx-j,:)=ROI1(j+1,:);

 end

 ROI1=ROI1corr;

end

F=fit(ROI1(:,1),ROI1(:,2),'smoothingspline','SmoothingParam',0.9);

dFdisc = differentiate(F,ROI1(:,1));

dF=fit(ROI1(:,1),dFdisc(:),'smoothingspline','SmoothingParam',0.9);

%figure, plot(ROI1(:,1), feval(F,ROI1(:,1)),'r-')

%hold on

%plot(ROI1(:,1), ROI1(:,2),'b-')

%hold off

clearvars filename delimiter formatSpec dataArray fileID ans cfs cfv dFdisc i;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Measurements

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i=2:ROIS

cfs=sprintf('ROI%1d',i);

ROI=eval(cfs);

if ROI(2,1)-ROI(1,1)<0

 ROIcorr=ROI;

 for j=0:1

 ROIcorr(2-j,:)=ROI(j+1,:);

 end

 ROI=ROIcorr;

end

bipole\_length(i-1,1)=sqrt(((ROI(2,1)-ROI(1,1))^2)+((ROI(2,2)-ROI(1,2))^2));

x0(i-1,1)=ROI(1,1);

y0(i-1,1)=ROI(1,2);

x1=ROI(2,1);

y1=ROI(2,2);

xm=(x0(i-1,1)+x1)/2;

ym=(y0(i-1,1)+y1)/2;

xmin=min(ROI1(:,1));

xmax=max(ROI1(:,1));

step=0;

for k=xmin:0.1:xmax

step=step+1;

R(step,1)=k;

R(step,2)=((k-xm)+(feval(F,k)-ym)\*feval(dF,k))/sqrt((feval(F,k)-ym)^2+(k-xm)^2);

end

RF=fit(R(:,1),R(:,2),'smoothingspline','SmoothingParam',0.9);

xclosest(i-1,1) = fzero(RF,xm);

bipole\_distance(i-1,1)=sqrt((xm-xclosest(i-1,1))^2+(ym-feval(F,xclosest(i-1,1)))^2);

tangent\_slope=feval(dF,xm);

u0(i-1,1)=x1-x0(i-1,1);

v0(i-1,1)=y1-y0(i-1,1);

tetha(i-1,1)=atand(tangent\_slope); %tetha is the angle of the curve

gamma(i-1,1)=asind((v0(i-1,1))/sqrt((v0(i-1,1))^2+(u0(i-1,1))^2)); %gamma is the angle of the bipole

Angle(i-1,1)=gamma(i-1,1)-tetha(i-1,1);

u1(i-1,1)=bipole\_length(i-1,1)\*cosd(Angle(i-1,1));

v1(i-1,1)=bipole\_length(i-1,1)\*sind(Angle(i-1,1));

AAResults(i-1,1:5)=[bipole\_length(i-1,1), bipole\_distance(i-1,1),Angle(i-1,1),u1(i-1,1),v1(i-1,1)];

clearvars i y1 yf ym xmin xmax xm xcenter x1 tangent\_slope ordinate k ;

end

%figure,plot(x0(:),xclosest(:),'m.');

figure,compass(u1,v1)

figure,rose((Angle/90)\*(pi/2),40)

figure, plot(ROI1(:,1), feval(F,ROI1(:,1)),'r-')

hold on

quiver(x0,y0,u0,v0)

hold off

clearvars F R RF ROI ROI1corr Rotate45 j step xclosest