**Introduction**

The objective of this lab is to familiarize ourselves with the concept of regression and interpolation with MATLAB while also understanding the principal of Bézier splines.

**Part 1 Case A**

clear

clc

close all

% Set up the initial conditions

f=10e9:1e6:11e9; % Investigate frequencies between 10 GHz and 11 GHz

fb=10.5e9; % Set the Brillouin frequency at 10.5 GHz

delf=35e6; % The gain profile has a 35 MHz width

amp=1; % Set the initial amplitude to 1

y=zeros(3,3);

% Add more noise (0.1)

disp('Noise @ 0.1')

for i=1:1:3

bg=amp\*(1+4\*((f-fb)./delf).^2).^-1; % Calculate the Brillouin gain over the frequency range

%Calculate the normally distributed noise

lnoise=randn(size(f)); % Add noise of standard deviation 0.1 to the gain profile

measured=0.1\*lnoise+bg; subplot(6,1,i)

% Plot the clean and the noisy data on the same graph

plot(f,measured,'r',f,bg,'g');

brillouin2=@(x) sum((measured-x(1)\*(1+4\*((f-x(2)\*1e10)./(x(3)\*1e7)).^2).^-1).^2);

[x,fval,exitflag,output]=fminsearch(brillouin2,[1,1.05,3.5]);

y(i,:)= x(1,:);

end

disp(y);

% Add more noise (0.2)

disp('Noise @ 0.2')

for i=4:1:6

bg=amp\*(1+4\*((f-fb)./delf).^2).^-1; % Calculate the Brillouin gain over the frequency range

lnoise=randn(size(f)); %Calculate the normally distributed noise

measured=(0.2\*lnoise)+bg; % Add noise of standard deviation 0.1 to the gain profile

subplot(6,1,i)

plot(f,measured,'r',f,bg,'g'); % Plot the clean and the noisy data on the same graph

brillouin2=@(x) sum((measured-x(1)\*(1+4\*((f-x(2)\*1e10)./(x(3)\*1e7)).^2).^-1).^2);

[x,fval,exitflag,output]=fminsearch(brillouin2,[1,1.05,3.5]);

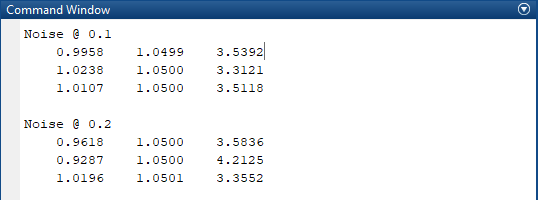
y2((i-3),:)= x(1,:);

end

disp(y2);



The top three are relevant to noise @ 0.1 while the bottom three display the noise @ 0.2 as indicated in the command window.



**Trial 1 : Plotting Std Dev vs Frequency**

clear

clc

close all

% Set up the initial conditions

f=10e9:1e6:11e9; % Investigate frequencies between 10 GHz and 11 GHz

fb=10.5e9; % Set the Brillouin frequency at 10.5 GHz

delf=35e6; % The gain profile has a 35 MHz width

amp=1; % Set the initial amplitude to 1

bg=amp\*(1+4\*((f-fb)./delf).^2).^-1; % Calculate the Brillouin gain over the frequency range

guide=repelem(2,100);

for i = 1:100

stdev(i)=0.01\*i;

for j=1:20

lnoise=randn(size(f));

measured=stdev(i)\*lnoise+bg;

brillouin2=@(x) sum((measured-x(1)\*(1+4\*((f- x(2)\*1e10)./(x(3)\*1e7)).^2).^-1).^2);

[x,error,exitflag,output]=fminsearch(brillouin2,[1,1.05,3.5]);

y(j)=x(2); % Pass Only Temp Data

err(j) = (abs(1.05-y(j))).\*10000;

end

freq(i) = sum(err)/20;

end

plot(stdev,freq,stdev,guide, 'k--')

xlabel('Standard Deviation')

ylabel('Temperature Error')

ylim([0 5]);

title('Standard Deviation vs. Temperature Error')



**Trial 2 : Plotting Brillouin Frequency vs Frequency**

clear

clc

clear all

f=10e9:1e6:11e9;

fb = 10.5e9;

delf=35e6;

amp=1;

bg=amp\*(1+4\*((f-fb)./delf).^2).^-1;

st(1) = 0; %Array of Standard Deviation Values

er(1) = 0; %Array of Error Values

j = 2; %Index of the Array

%Creates errors in Brillouin values with varying standard devations

for i = 0.1:0.01:1

lnoise=randn(size(f));

measured= i\*lnoise + bg;

brillouin2=@(x) sum((measured-x(1)\*(1+4\*((f-x(2)\*1e10)./(x(3)\*1e7)).^2).^-1).^2);

[x,fval,exitflag,output]=fminsearch(brillouin2,[1,1.05,3.5])

st(j) = i;

er(j) = abs(1.05 - x(2));

j = j + 1;

end

%Plots Error v.s. Standard Deviation

%plot(st,er,'o',st,guide)

plot(st,er,'o')

title(' Error in Brillouin Frequency v.s. Std Dev of Noise Level')

xlabel('Std Dev of Noise')

ylabel('Error in Brillouin Frequency')



**Part 2 : Bézier Splines**

clear

clc

close all

%s array

s=[

237 620 237 620 237 120 237 120

237 120 237 35 226 24 143 19

143 19 143 19 143 0 143 0

143 0 143 0 435 0 435 0

435 0 435 0 435 19 435 19

435 19 353 23 339 36 339 109

339 109 339 108 330 620 339 620

339 620 339 620 393 620 393 620

393 620 507 620 529 602 552 492

552 492 552 492 576 492 576 492

576 492 576 492 570 662 570 662

570 662 570 662 6 662 6 662

6 662 6 662 0 492 0 492

0 492 0 492 24 492 24 492

24 492 48 602 71 620 183 620

183 620 183 620 237 620 237 620];

s=s+100;

t=0:0.01:1;

for n=1:16

x1=s(n,1);y1=s(n,2);x2=s(n,3);y2=s(n,4);x3=s(n,5);y3=s(n,6);x4=s(n,7);y4=s(n,8);

bx=3\*(x2-x1);

cx=3\*(x3-x2)-bx;

dx=x4-x1-bx-cx;

by=3\*(y2-y1);

cy=3\*(y3-y2)-by;

dy=y4-y1-by-cy;

x=x1+bx\*t+cx\*t.^2+dx\*t.^3;

y=y1+by\*t+cy\*t.^2+dy\*t.^3;

hold on

plot(x,y)

end

%Array of Bézier splines

%Array of Bézier splines

s=[

200 50 200 50 200 650 200 750

200 750 200 750 350 750 350 750

350 750 350 750 550 600 350 450

350 450 350 450 275 450 275 450

275 450 275 450 275 50 275 50

200 50 200 50 275 50 275 50

350 700 450 600 350 500 350 500

275 500 275 500 353 500 353 500

275 700 275 700 353 700 353 700

275 500 275 500 275 700 275 700

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

800 800 800 800 800 800 800 800];

%Application of appropriate formulae and graphing

t=0:0.01:1;

for n=1:16

x1=s(n,1);y1=s(n,2);x2=s(n,3);y2=s(n,4);x3=s(n,5);y3=s(n,6);x4=s(n,7);y4=s(n,8);

bx=3\*(x2-x1);

cx=3\*(x3-x2)-bx;

dx=x4-x1-bx-cx;

by=3\*(y2-y1);

cy=3\*(y3-y2)-by;

dy=y4-y1-by-cy;

x=x1+bx\*t+cx\*t.^2+dx\*t.^3;

y=y1+by\*t+cy\*t.^2+dy\*t.^3;

hold on

plot(x,y)

end

title('Character P & T: Represented with Bézier Splines')

xlabel('X Value')

ylabel('Y Value')

%Application of appropriate formulae and graphing

t=0:0.01:1;

for n=1:16

x1=s(n,1);y1=s(n,2);x2=s(n,3);y2=s(n,4);x3=s(n,5);y3=s(n,6);x4=s(n,7);y4=s(n,8);

bx=3\*(x2-x1);

cx=3\*(x3-x2)-bx;

dx=x4-x1-bx-cx;

by=3\*(y2-y1);

cy=3\*(y3-y2)-by;

dy=y4-y1-by-cy;

x=x1+bx\*t+cx\*t.^2+dx\*t.^3;

y=y1+by\*t+cy\*t.^2+dy\*t.^3;

hold on

plot(x,y)

end

title('Character P & T: Represented with Bézier Splines')

xlabel('X Value')

ylabel('Y Value')



**Conclusion**

Overall we’ve successfully used the concept of regression and that of the standard deviation to analyze the overall effect on data, i.e. the distribution of sound (noise). After which we’ve used Bézier splines to create characters P & T via interpolation.