# **Question 1**

**Part A**

To calculate the DC resistance of the object it can be split up into three distinct sections defined by the lengths , and respectively, the resistance for each section can then be calculated using the given formulas of;

Subbing into the equation the values given for and produces results for the first and third section of **0.412mΩ** and **24.9µΩ** respectively. Using the second formula above the resistance of the conical section is **0.312mΩ**. Therefore the total resistance of the object can be calculated by combining the resistance of each element to produce a total resistance of **0.748mΩ**.

**Part B**

At frequency the resistance of the object will change due to the skin effect. For the first and third section the skin effect will be constant across the length however, the radius of the conical section changes with length and therefore will be affected differently at higher frequencies. To calculate these changes the conical section is split into smaller chunks with increasing radius. The skin effect and the resistance can then be calculated for each chunk individually and summed together to produce an estimate for the resistance of the conical section at frequency.

The critical frequency is the point at which the skin effect is equal to the radius of the object, before this frequency is reached the resistance of the section will be at its normal DC value. The following MATLAB code calculates the critical frequency and DC resistance of each section of the object with the conical section being split into 1000 chunks.

alpha = atan((b-a)/Lc);

% Calculate critical frequencies and DC resistances

% First section

R1cf = critf(rho,a);

% Conical section

N = 1000;

Lc\_step = Lc/N;

for n = 1:1:N

h(n) = Lc\_step\*n.\*tan(alpha)+a;

RCcf(n) = critf(rho,h(n));

RCdc(n) = rwire(rho,Lc\_step,h(n));

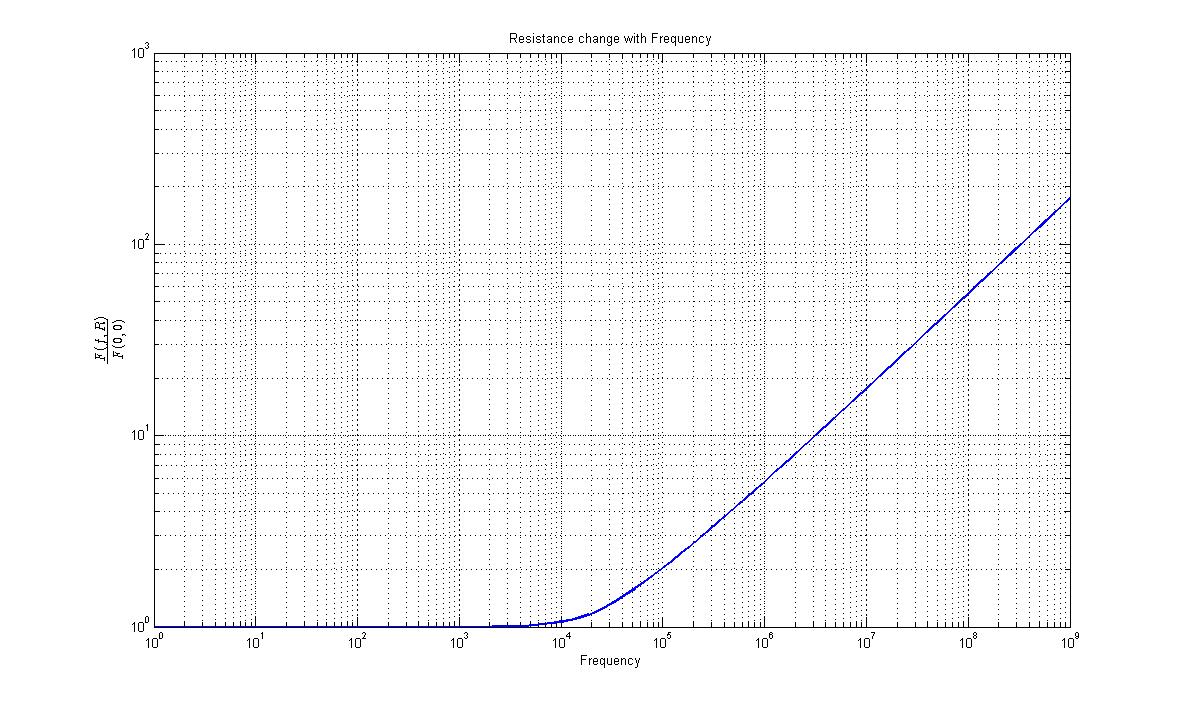
end

% End Section

R2cf = critf(rho,b);

The code for the rwire, rconical and critf functions are found in Appendix B: Function Code.

These values are then used in a for loop to calculate the resistance of the object, as a whole, over a range of 100 frequencies on a logarithmic scale up to 1GHz. This for loop compares the current frequency value to the critical frequency of each section, if the frequency is equal to or greater than the critical frequency then the skin depth and resistance are calculated from that frequency, else the DC value is passed through. The code for this loop is found in Appendix A: Q1.m Part B. The result of this code is shown in Figure 1.



**Figure 1: Ratio of Resistance at frequency to DC Resistance.**

# **Question 2**

**Part A**

To calculate the resistance of the helical coil at DC the given formula is used. This produces a result of **17.55Ω**. The MATLAB code that produces this answer is found in Appendix A: MATLAB Code, this uses a user-defined function called rhelical, found in Appendix B: Function Code. This function calculates the resistance of a helical coil with constant or varying pitch at DC, the function returns both the DC resistance and the length of the wire in the coil.

**Part B**

At a frequency of 1MHz the resistance of the helical coil will be affected by both the skin and proximity effect. The following MATLAB code calculates the resistance at 1MHz by calculating the skin and proximity effects separately and multiplying them by the DC value.

%% Part B

% Estimate the resistance at 1MHz

delta = dskin(rho,1\*10^6);

Sskin = 2.\*a.\*pi.\*delta.\*(1-(delta./(2.\*a)));

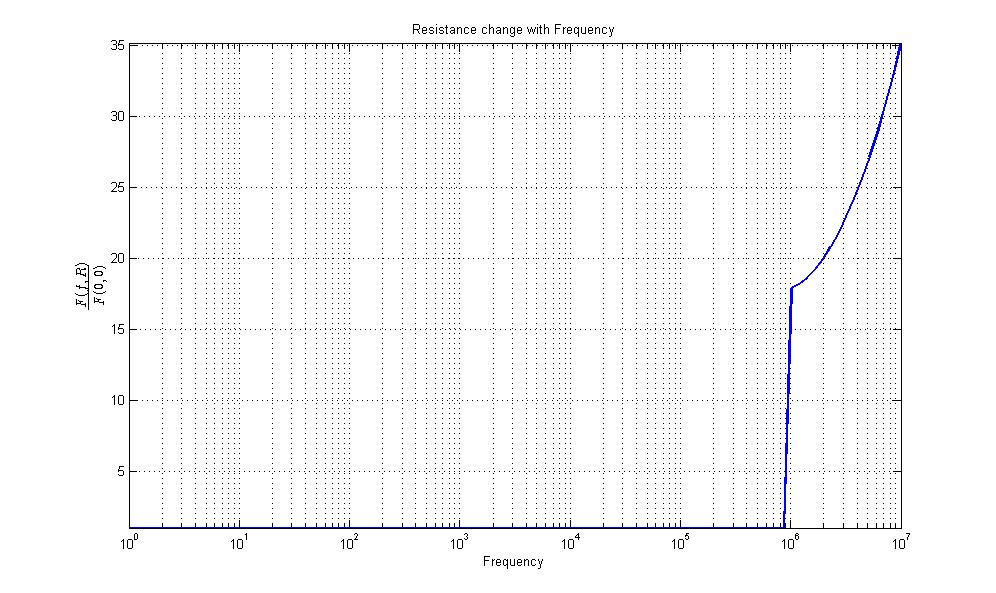
Rskin = (Coil\_length.\*rho)./Sskin;

RFreq = DCResis.\*Rskin.\*(1+((2.\*(a^2))./(p^2)))

This produces a result of approximately **314Ω (313.79)**.

**Part C**

To calculate the resistance change over frequency the same method used in question 1 part b is used. Firstly the critical frequency of the helical coil is calculated using the created critf function, this value is then used in a for loop to calculate the resistance of the coil, taking into account the skin and proximity effect above the critical frequency. The result is shown in figure 2.



**Figure 2: Helical Coil Resistance Ratio at Frequency.**

**The following MATLAB code produces the results for the graph above, the full code can be found in Appendix A: MATLAB Code Q2.m.**

Rcf = critf(rho,a);

% Loop to calculate resistance with frequency

count = 100;

RT = zeros(count);

Freq = logspace(0,7,count);

for ln = 1:1:count;

if Freq(ln) >= Rcf

delta = dskin(rho,Freq(ln));

Sskin = 2.\*a.\*pi.\*delta.\*(1-(delta./(2.\*a)));

Rskin = (Coil\_length.\*rho)./Sskin;

RT(ln) = DCResis.\*Rskin.\*(1+((2.\*(a^2))./(p^2)));

else

RT(ln) = DCResis;

end

end

**Part D**

**%%**

# **Question 3**

**Design**

# **Question 4**

**Part A**

# **Question 5**

# **Question 6**

# **Question 7**

# **Appendix A: MATLAB Code**

**Q1.m**

%% Fast Tansient Sensors - Q1 - Coursework 2

% B126949 - Tom Young

%% Pre - Cursor

clear all

clc

%% Constants

% Using 1&9 give constants as O-O

a = 3.5\*10^-3; %mm

b = 10.5\*10^-3; %mm

La = 22\*10^-3; %mm

Lb = 12\*10^-3; %mm

Lc = 50\*10^-3; %mm

rho = 7.2\*10^-7; %Ohm-Meter

%% Part A

% Calculate the DC Resistance of the conductor

% Resistence of small wire

R1 = rwire(rho,La,a);

% Resistence of large wire

R2 = rwire(rho,Lb,b);

% Resistance of conical section

RC\_DC = rconical(a,b,Lc,rho);

% Total resistance

DCResis = R1 + R2 + RC\_DC

%% Part B

% Draw a graphical representation of the conductor resistance variation

% with frequency up to 1GHz

alpha = atan((b-a)/Lc);

% Calculate critical frequencies and DC resistances

% First section

R1cf = critf(rho,a);

% Conical section

N = 1000;

Lc\_step = Lc/N;

for n = 1:1:N

h(n) = Lc\_step\*n.\*tan(alpha)+a;

RCcf(n) = critf(rho,h(n));

RCdc(n) = rwire(rho,Lc\_step,h(n));

end

% End Section

R2cf = critf(rho,b);

% Loop to calculate resistance with frequency

% Number of loops

count = 100;

RFreq=zeros(count);

Freq = logspace(0,9,count);

for ln = 1:1:count;

RC = 0;

% First Section

if Freq(ln) >= R1cf

delta = dskin(rho,Freq(ln));

Sskin = 2.\*a.\*pi.\*delta.\*(1-(delta./(2.\*a)));

RF = (La.\*rho)./Sskin;

%RF = R1.\*(a./(2.\*delta));

else

RF = R1;

end

% Conical Section

for n = 1:1:N

if Freq(ln) >= RCcf(n)

delta = dskin(rho,Freq(ln));

Sskin = 2.\*h(n).\*pi.\*delta.\*(1-(delta./(2.\*h(n))));

RC(n) = (Lc\_step.\*rho)./Sskin;

%RC(n) = RCdc(n).\*(h(n)./(2.\*delta));

else

RC(n) = RCdc(n);

end

end

% End Section

if Freq(ln) >= R2cf

delta = dskin(rho,Freq(ln));

Sskin = 2.\*b.\*pi.\*delta.\*(1-(delta./(2.\*b)));

RE = (Lb.\*rho)./Sskin;

%RE = R2.\*(b./(2.\*delta));

else

RE = R2;

end

% Total resistance

RFreq(ln) = (RE+RF+sum(RC));

end

% Plot data

loglog(Freq,RFreq./DCResis,'b','linewidth',2)

grid on

xlabel('Frequency')

ylabel('$${F(f,R)\over F(0,0) }$$','Interpreter','Latex')

xlim([0 1\*10^9])

title('Resistance change with Frequency')

**Q2.m**

%% Fast Tansient Sensors - Q2 - Coursework 2

% B126949 - Tom Young

%% Pre - Cursor

clear all

clc

%% Constants

% Using 1&9 give constants as O-O

d = 0.14\*10^-3; %mm

D = 4\*10^-2; %cm

L = 10\*10^-2; %cm

p = 0.8\*10^-3; %mm

rho = 1.72\*10^-8; %Ohm-Meter

a = d/2; %radius mm

%% Part A

% Caculate the DC resistance

[DCResis, Coil\_length] = rhelical(L,d,p,rho,D);

DCResis

%% Part B

% Estimate the resistance at 1MHz

delta = dskin(rho,1\*10^6);

Sskin = 2.\*a.\*pi.\*delta.\*(1-(delta./(2.\*a)));

Rskin = (Coil\_length.\*rho)./Sskin;

RFreq = DCResis.\*Rskin.\*(1+((2.\*(a^2))./(p^2)))

%% Part C

% Draw a graphical representation of the resistance variation with

% frequency up to a frequency of 1GHz.

Rcf = critf(rho,a);

% Loop to calculate resistance with frequency

count = 100;

RT = zeros(count);

Freq = logspace(0,7,count);

for ln = 1:1:count;

if Freq(ln) >= Rcf

delta = dskin(rho,Freq(ln));

Sskin = 2.\*a.\*pi.\*delta.\*(1-(delta./(2.\*a)));

Rskin = (Coil\_length.\*rho)./Sskin;

RT(ln) = DCResis.\*Rskin.\*(1+((2.\*(a^2))./(p^2)));

else

RT(ln) = DCResis;

end

end

% Plot data

semilogx(Freq,RT./DCResis,'b','linewidth',2)

%semilogx(Freq,RT,'b','linewidth',2)

grid on

xlabel('Frequency')

ylabel('$${F(f,R)\over F(0,0) }$$','Interpreter','Latex')

xlim([0 10\*10^6])

ylim([1 inf])

title('Resistance change with Frequency')

**Q3.m**

**Q4.m**

**Q5.m**

**Q6.m**

**Q7.m**

# **Appendix B: Function Code**

**rwire.m**

function [ rwire ] = rwire( resistivity, length, radius )

%RWIRE rwiretace of a wire of length l, with cross sectional area A, made

%of a material with risitivity ro.

% rwiretace of a wire derived from Ohm's law.

cross\_sectional\_area = pi.\*(radius.^2);

rwire = resistivity.\*(length./cross\_sectional\_area);

end

**rconical.m**

function [ rconical ] = rconical( starting\_radius, ending\_radius, length, resistivity )

%RCONICAL Resistace of a conical conductor of length l, with variable

% cross sectional area A, made of a material with risitivity ro.

% Resistance of a conical conductor of length l with varable cross

% section over length c.

rconical = resistivity.\*(length./(pi.\*starting\_radius.\*ending\_radius));

end

**critf.m**

function [ critf ] = critf( resitivity, radius )

%CRITF Calculates critical frequency

% Calculates ciritcal frequency when the skin effect is equal to the

% radius of the object.

%% Constants

u\_0 = 4\*pi\*10^-7;

%% Equations

critf = resitivity./((radius.^2).\*pi.\*u\_0);

end

**dskin.m**

function [ dskin ] = dskin( resitivity, frequency )

%DSKIN Calculates skin effect

% Calculates the skin effect depth at frequency f of a material with

% resistivity rho.

%% Constants

u\_0 = 4\*pi\*10^-7;

%% Equations

dskin = sqrt(resitivity./(frequency.\*pi.\*u\_0));

end

**rhelical.m**

function [ wire\_resis, wire\_length ] = rhelical( coil\_height, wire\_thickness, pitch, resistivity, varargin )

%rhelical Resistace of a helical coil made from a wire with constant pitch.

% Resistance of a helical coil of length l, made from a wire turned with

% a constant pitch on a mandral with constant radius r or varying radius

% r1, r2.

if length(varargin) == 1

length\_pre = coil\_height./pitch;

length\_coil = length\_pre.\*sqrt((2.\*pi.\*varargin{1}/2).^2 + pitch.^2);

else

r1 = varargin{0};

r2 = varargin{1};

alpha = (r2-r1)./coil\_height;

c\_pre = (pitch./(2\*pi)).^2;

c = c\_pre.\*(1+tan(alpha)^2);

a = sqrt((r2.^2)+(c.^2));

b = sqrt((r1.^2)+(c.^2));

length\_pre = pi./(pitch.\*tan(alpha));

length\_coil = length\_pre.\*((a.\*r2)- (b.\*r1)+c.\*log((a.\*r2)./(b.\*r1)));

end

wire\_resis = rwire(resistivity,length\_coil,wire\_thickness/2);

wire\_length = length\_coil;

end