# *Note: The code used for each question is provided in Appendix A, all relevant user-defined functions are found in Appendix B*

# **Question 1**

**Load Voltage**

The time-dependence of the load voltage produced by the Marx generator is shown in figure 1. The load voltage has a peak voltage of **493kV,** a rise time of **98.4ns** and an estimated bandwidth of **3.56MHz**. To measure this impulse using a voltage sensor with an accuracy of 2% requires the voltage sensor to have a rise time which is five times faster than that of the load voltage. This means that the sensor is required to have a rise time of **19.7ns** and a bandwidth of **17.8MHz**.

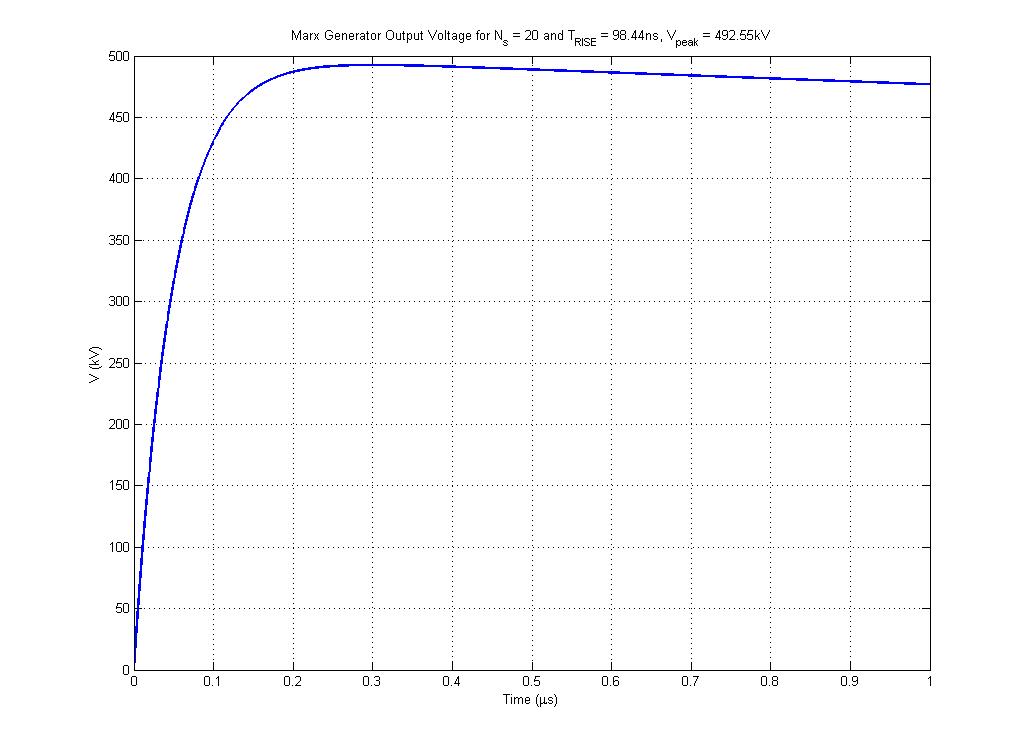


Figure 1: Marx Generator Discharge Load Voltage

**Voltage Divider**

The low voltage arm of the divider will carry the voltage to be measured by the oscilloscope; therefore to reduce the effect of mismatched loads distorting the signal a simple design for the low voltage arm has been chosen. The characteristics of this design are shown in Table 1; the value chosen for is **50Ω**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |

Table 1: Low Voltage Arm Characteristics

The oscilloscope that will measure the Marx generator output is only capable of measuring a maximum of 80V, therefore, using the attenuation factor equation and the value of the peak Marx generator output the high voltage arm resistance is required to be in the region of **308kΩ**. The high voltage arm is surrounded by polythene with a surface breakdown voltage of 5kV/cm requiring the high voltage column to be **98.5cm** tall to avoid the voltage breaking across the polythene.

Using the Ayrton-Perry method to reduce the inductance of the system requires the resistance of each wire to double, so that the parallel combination of the resistances is equal to the required value. For a mandrel radius of **5cm** and using wire with a diameter of **30µm** will require **630m** of wire at  **1004turns** over the height of the resistive column. The following code calculates the change in temperature of the chosen wire using the total energy in the Marx generator pulse. The time over which the energy is calculated was chosen as the time at which the load voltage reduced to zero. This results in a temperature change of approximately **32.6K**. Increasing the diameter of the wire would increase the amount of wire needed but would also decrease the temperature change.

%Energy

tmax = 130\*10^-6;

Q = ((tmax\*(max(VLoad)^2))/(2\*Zhv));

%Wire Mass

wire\_vol = (pi\*(wire\_radius^2))\*twirelength; %m^3

wire\_mass = wire\_vol\*evan\_dens; %kg

%Temperature

deltaT = Q./(wire\_mass.\*evan\_heat)

To minimise the stray capacitance the grading structure shown in figure 2 is used. This involves two circular rings, made of aluminium, on both the high voltage and low voltage ends of the voltage divider. This method creates an equipotential field along the height of the resistive column with a capacitance, calculated using Maxwell SV, of **16pF**.

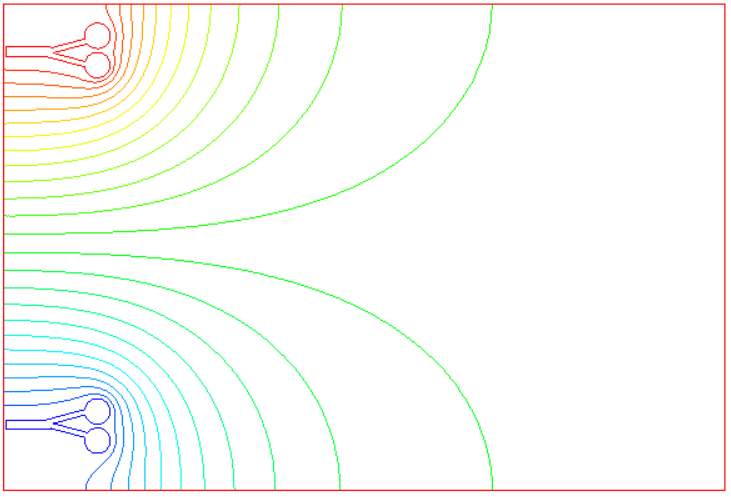


Figure 2: Field Controlling Structure for Voltage Divider.

Figure 3 shows that a metallic object can be placed up to **80cm** away from the field without causing much disturbance to the electric field. Therefore the high voltage connecting lead has a length of **80cm** and is made from wire with a diameter of **10mm**. To reduce the inductance of the connecting wire **4** are connected in parallel with a total inductance of **200nH**.The total inductance for the system is calculated using the equation for a rectangular arrangement and produced a value of **1.1µH**.

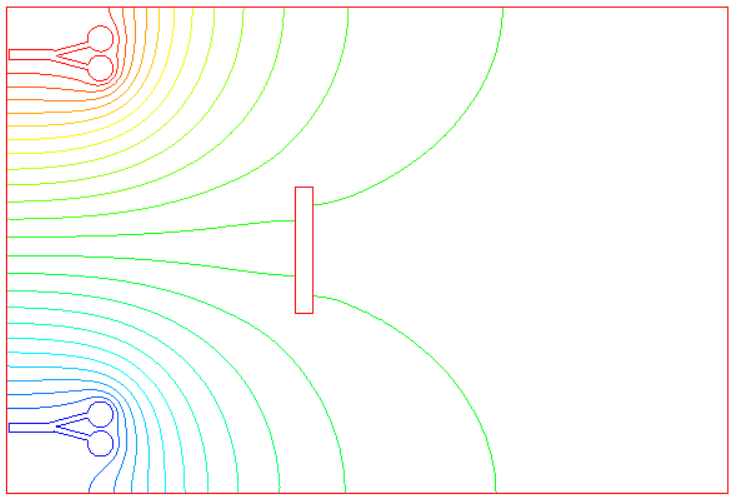


Figure 3: Field Controlling Structure with Foreign Metallic Object

Using the calculated inductance and capacitance of the system the transfer function of the system is a general second order system in the form;

Therefore the response time of the circuit can be expressed as below for an ideal damping ratio of 10% overshoot;

The following code calculates the rise time of the system as a whole using the ideal damping ratio of 10% to produce the total rise time of the system as **10.9ns**. Comparing this value to the required rise time of **19.7ns** shows that the system is capable of measuring the voltage from the Marx generator system.

rd\_sys = 1.2\*sqrt(indSys./cHV);

rd\_connector = 1.2\*sqrt(indConnector./cHV);

Ttotal = sqrt(((rd\_sys.\*cHV)+(rd\_connector.\*cHV))^2)

As mentioned above the system transfer function is that of a second order system, therefore the overshoot of the system can be expressed as below;

For an ideal overshoot of 10% the damping factor ζ can be expressed as;

An alternative method of designing the low voltage arm would be to choose the design shown in table 2. From this design the low voltage arm would have a negative rise time; this would reduce the overall rise time of the system.

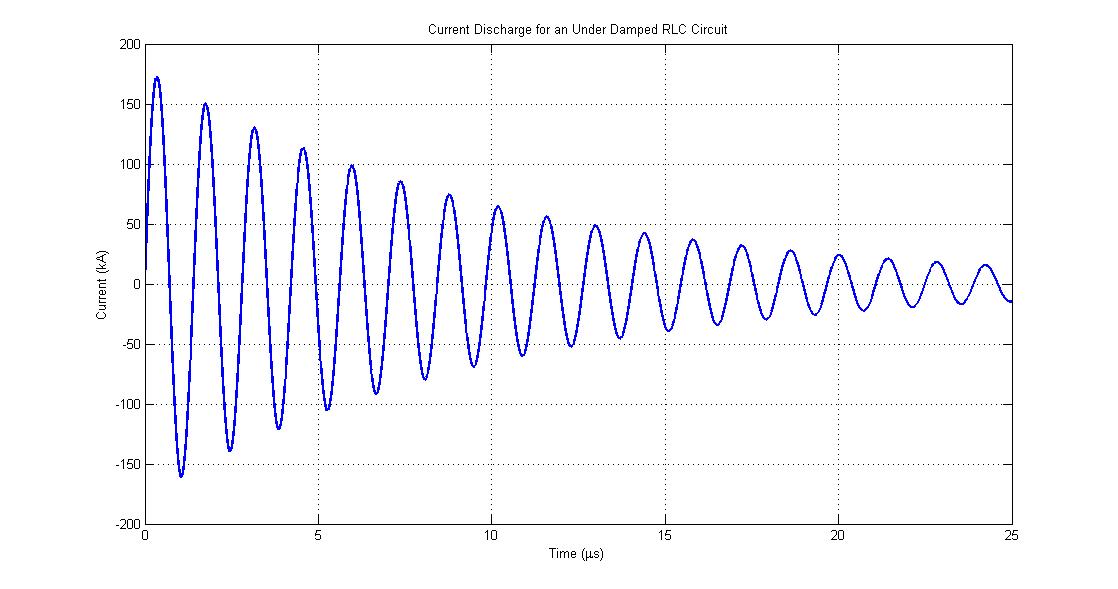
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |

Table 2: Alternative Low Voltage Arm Characteristics

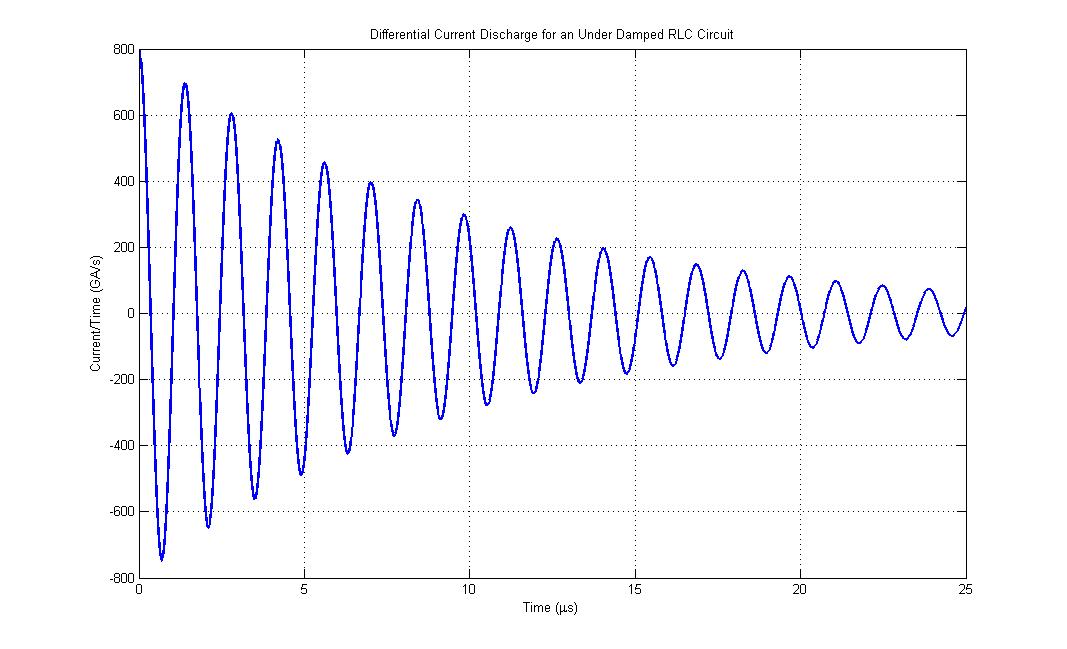
The full code for this question is found in Appendix A: Q1.m.

# **Question 2**

**Figure 4 and 5 show the current discharge in the RLC circuit ad it’s differential.**



**Figure 4: RLC Current Discharge**



**Figure 5: RLC Differential Current Discharge**

**Coil Design**

**The Rogowski coil was chosen to have characteristics of; major radius 2cm, minor radius 2mm, made with a wire of radius 1mm with a total of 10 turns. The cable impedance was chosen as 50Ω. The following code calculates the self-inductance and resistance for the described characteristics of the Rogowski coil and produces results of 38nH and 6.2mΩ. Figure 6 shows the time rate of change of the magnetic flux in the Rogowski coil.**

%Inductance

Induct\_pre = u0.\*minor\_radius.\*N;

Induct\_sum0 = 0.0007\*((log((2\*pi\*major\_radius)/p))^0);

Induct\_sum1 = 0.1773\*((log((2\*pi\*major\_radius)/p))^1);

Induct\_sum2 = -0.0322\*((log((2\*pi\*major\_radius)/p))^2);

Induct\_sum3 = 0.00197\*((log((2\*pi\*major\_radius)/p))^3);

Induct\_sum = Induct\_sum0 + Induct\_sum1 + Induct\_sum2 + Induct\_sum3;

Induct\_pre2 = (((pi\*minor\_radius)/p)+(log((2\*p)/(wire\_diameter)))-(5/4)-Induct\_sum);

RCoil\_inductance = Induct\_pre\*Induct\_pre2

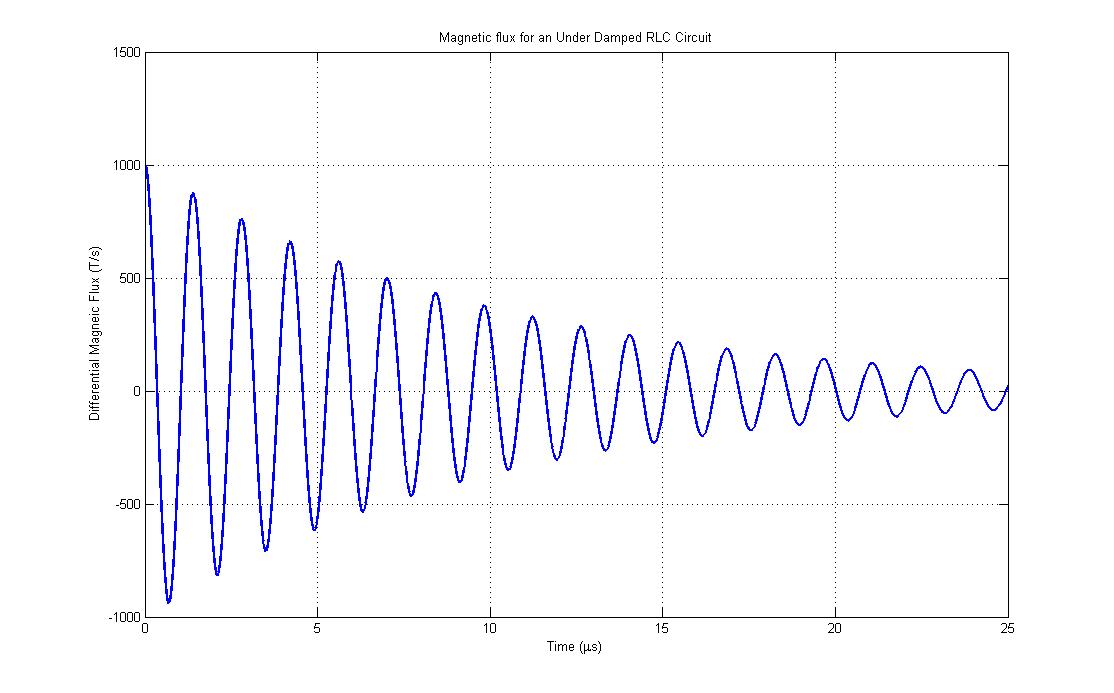
%Resistance

freq = 1/(2\*pi\*sqrt(Lb\*Cb));

RCoil1 = N/(pi\*wire\_diameter);

RCoil2 = sqrt((copper\_rho\*pi\*freq\*u0)\*((p^2)+((2\*pi\*minor\_radius)^2)));

RCoil\_resistance = RCoil1\*RCoil2



**Figure 6: Differential Magnetic Flux**

**The following coil calculates the current and differential current in the Rogowski coil. Figures 7 and 8 show the current and differential current in the coil.**

Rrt = RCoil\_resistance + Rcvr;

Lr = RCoil\_inductance;

omega = sqrt((1./(Lb.\*Cb))-((Rb./(2.\*Lb))^2));

rogowski\_current = @(x) exp((Rrt./Lr).\*x).\*(k.\*(Vo./(omega.\*Lb).\*((omega.\*exp(-1.\*(Rb./(2.\*Lb)).\*x).\*cos(omega.\*x))-((Rb./(2.\*Lb)).\*exp(-1.\*(Rb./(2.\*Lb)).\*x).\*sin(omega.\*x)))));

t2 = 0:1\*10^-10:10\*10^-9;

RIntegral = zeros(0,length(t2));

for n=1:length(t2)

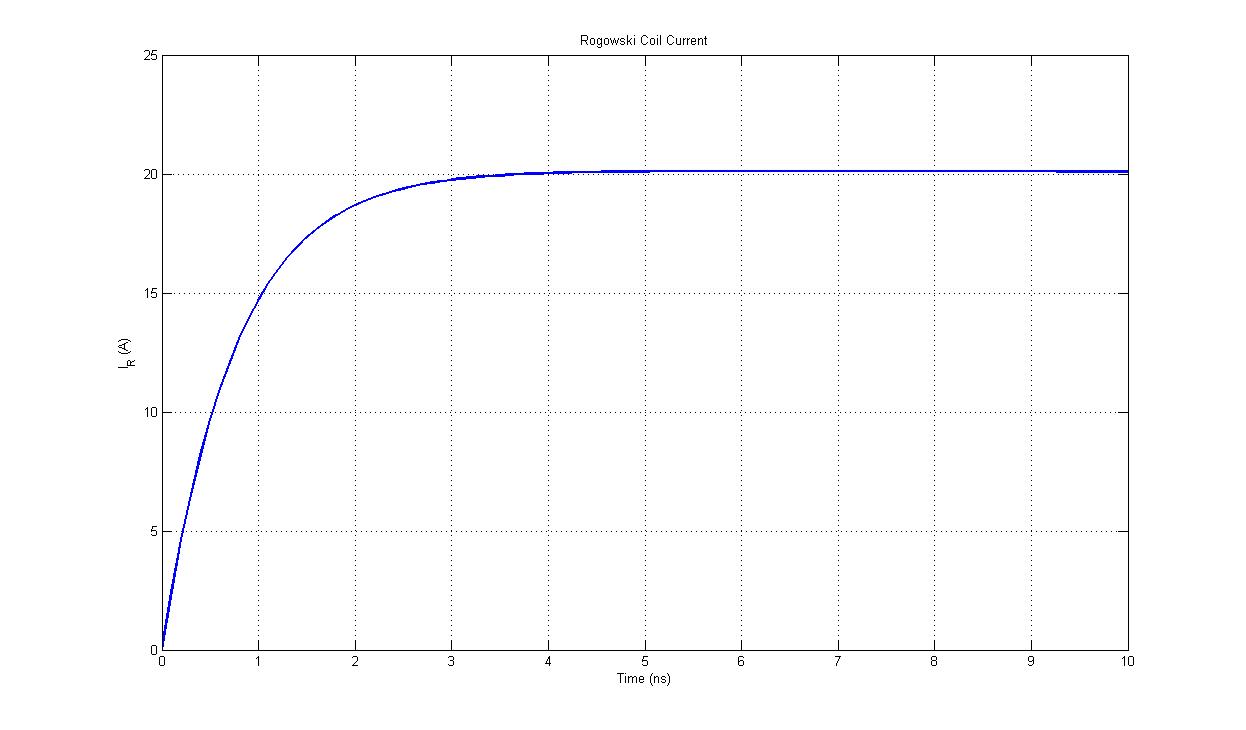
RIntegral(n) = integral(@(x)rogowski\_current(x),0,t2(n));

end

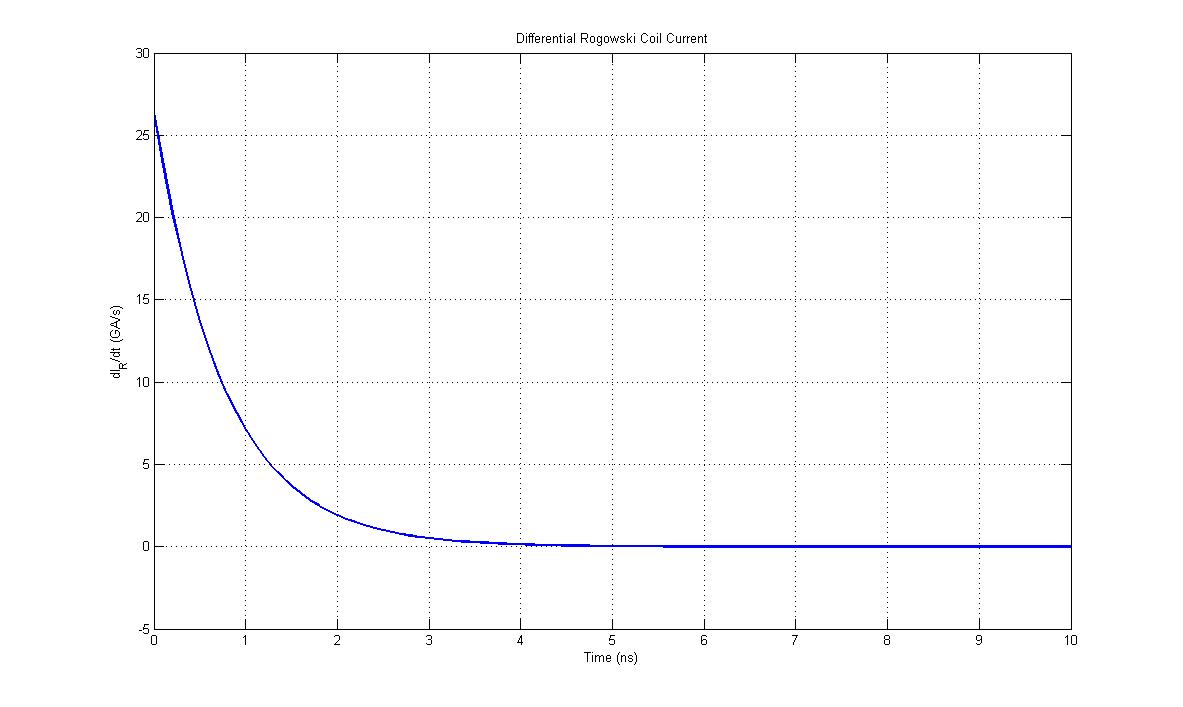
Rogowski\_Current = (exp(-(Rrt./Lr).\*t2)./Lr).\*RIntegral;

dRogowski\_Current\_prefix = (exp(-(Rrt./Lr).\*t2)./Lr).\*exp((Rrt./Lr).\*t2).\*(k.\*(Vo./(omega.\*Lb).\*((omega.\*exp(-1.\*(Rb./(2.\*Lb)).\*t2).\*cos(omega.\*t2))-((Rb./(2.\*Lb)).\*exp(-1.\*(Rb./(2.\*Lb)).\*t2).\*sin(omega.\*t2)))));

dRogowski\_Current = dRogowski\_Current\_prefix + (RIntegral.\*((-Rrt.\*exp((-Rrt./Lr).\*t2))./(Lr^2)));

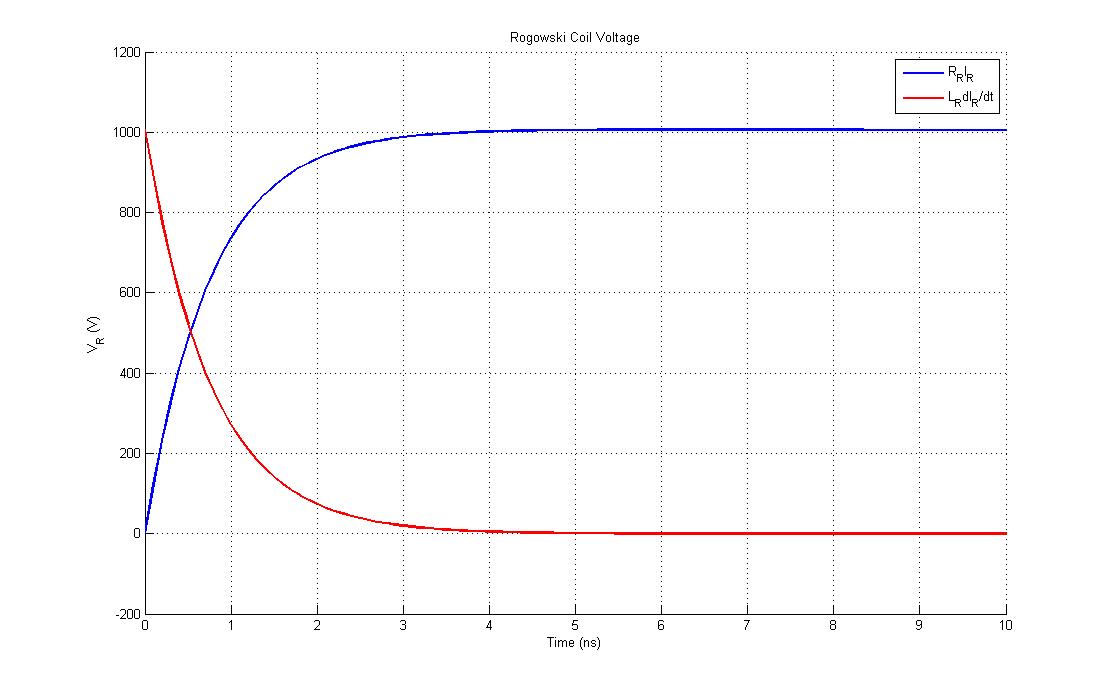


**Figure 7: Rogowski Current**

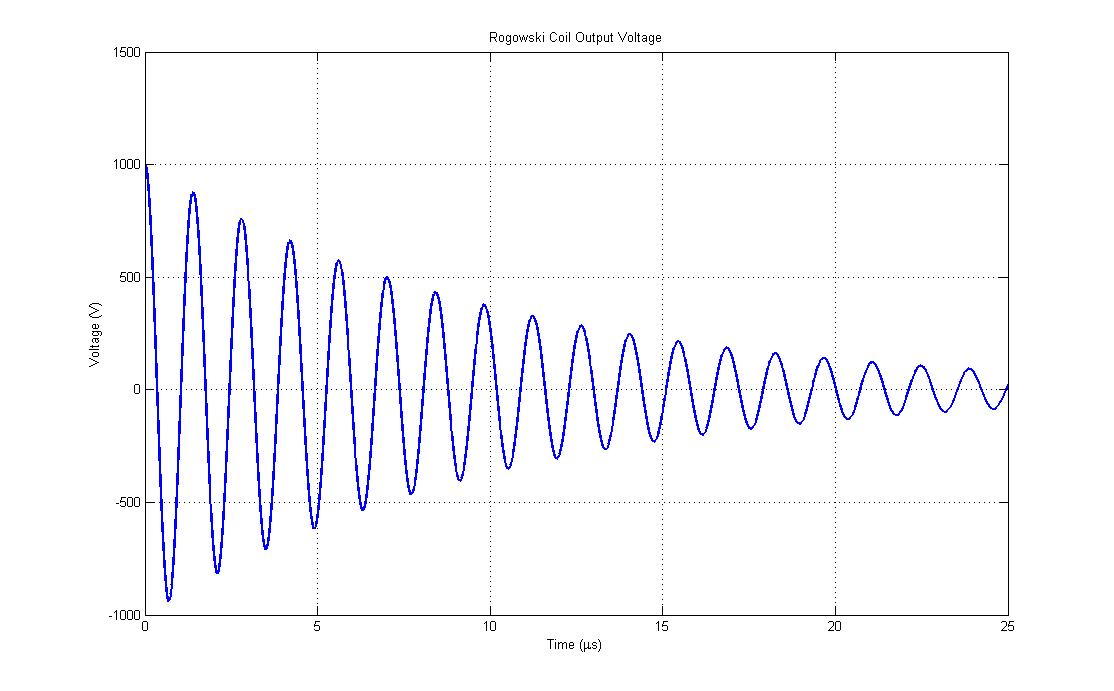


**Figure 8: Differential Rogowski Current**

**Using the same code as above and the total resistance and inductance of the coil, figure 9 demonstrates that the designed coil is acting as a differential coil for 0.5ns. Figure 10 shows the output voltage of the Rogowski coil.**

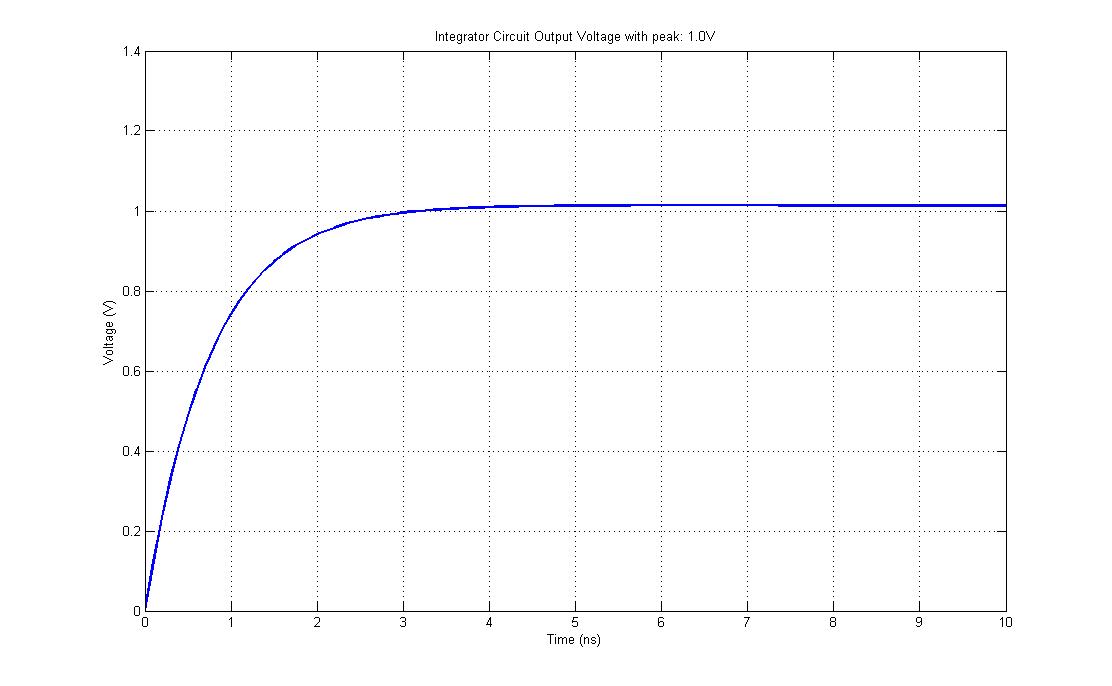


**Figure 9: Sensor Characteristics**



**Figure 10: Rogowski Coil Output Voltage**

**Figure 11 shows the output voltage after the integrator circuit with a resistance of 50mΩ and a capacitance of 500nF.**



**Figure 10: Integrator Output Voltage**

The full code for this question is found in Appendix A: Q2.m.

# **Question 3**

Figures 11 and 12 show the current discharge for the over-damped RLC circuit ad it’s differential.

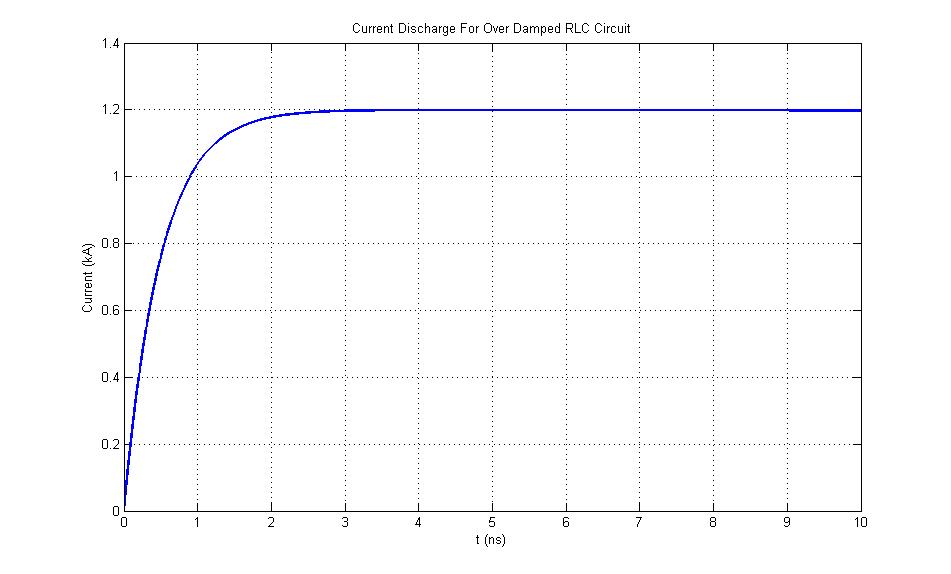


Figure 11: RLC Current Discharge

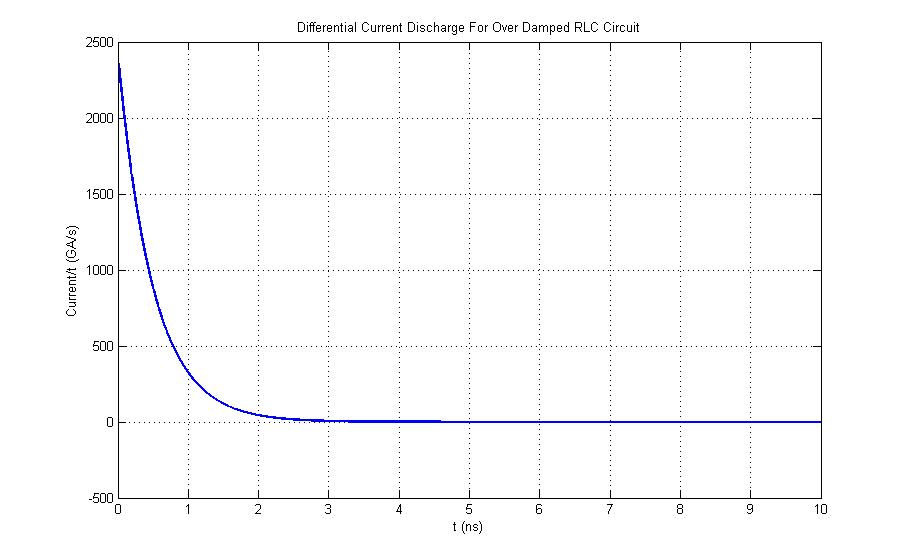


Figure 12: Differential RLC Current Discharge

The Rogowski coil was chosen to have characteristics of; major radius **10cm**, minor radius **2.5mm**, made with a wire of radius **1.5mm** with a total of **15 turns**. The cable impedance was chosen as **5Ω**. The following code calculates the self-inductance and resistance for the described characteristics of the Rogowski coil and produces results of **93.5nH** and **8.2mΩ**. Figure 13 shows the time rate of change of the magnetic flux in the Rogowski coil. Figures 14 and 15 show the current and differential current in the Rogowski coil. Figure 16 shows that the coil is self-integrating for a time of **1.5ns** and figure 17 shows the output voltage of the coil with a peak of **34V.**

%Inductance

Induct\_pre = u0.\*minor\_radius.\*N;

Induct\_sum0 = 0.0007\*((log((2\*pi\*major\_radius)/p))^0);

Induct\_sum1 = 0.1773\*((log((2\*pi\*major\_radius)/p))^1);

Induct\_sum2 = -0.0322\*((log((2\*pi\*major\_radius)/p))^2);

Induct\_sum3 = 0.00197\*((log((2\*pi\*major\_radius)/p))^3);

Induct\_sum = Induct\_sum0 + Induct\_sum1 + Induct\_sum2 + Induct\_sum3;

Induct\_pre2 = (((pi\*minor\_radius)/p)+(log((2\*p)/(wire\_diameter)))-(5/4)-Induct\_sum);

RCoil\_inductance = Induct\_pre\*Induct\_pre2;

%Resistance

freq = 1/(Rb.\*Cb);

% freq = 1./(Rb.\*Cb);

RCoil1 = N/(pi\*wire\_diameter);

RCoil2 = sqrt((copper\_rho\*pi\*freq\*u0)\*((p^2)+((2\*pi\*minor\_radius)^2)));

RCoil\_resistance = RCoil1\*RCoil2;

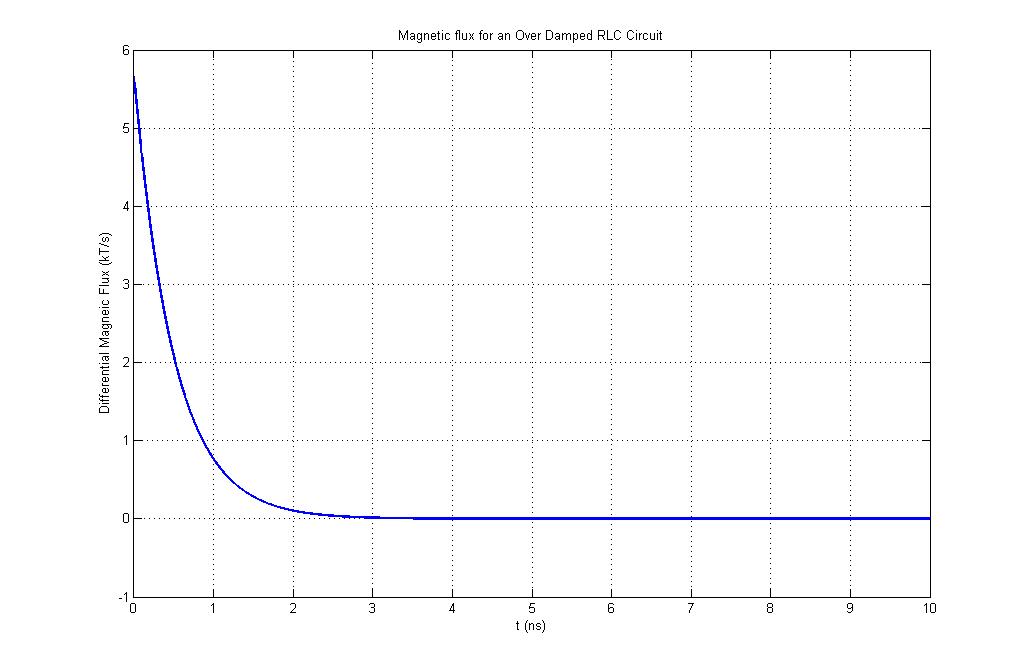


Figure 13: Differential Magnetic Flux

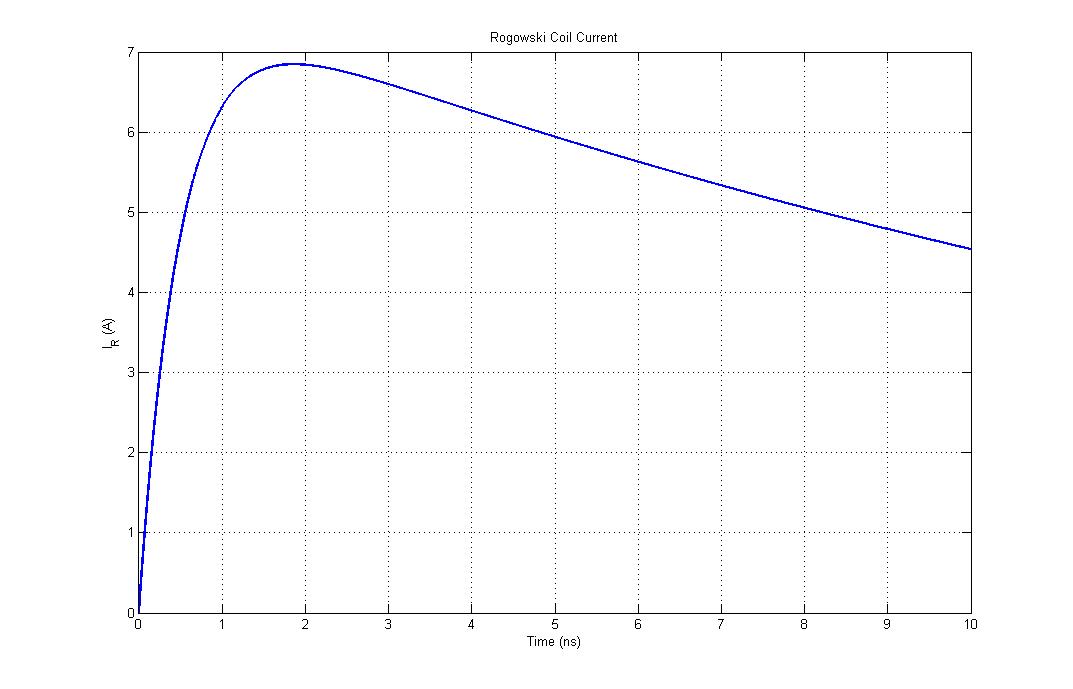


Figure 14: Rogowski Coil Current

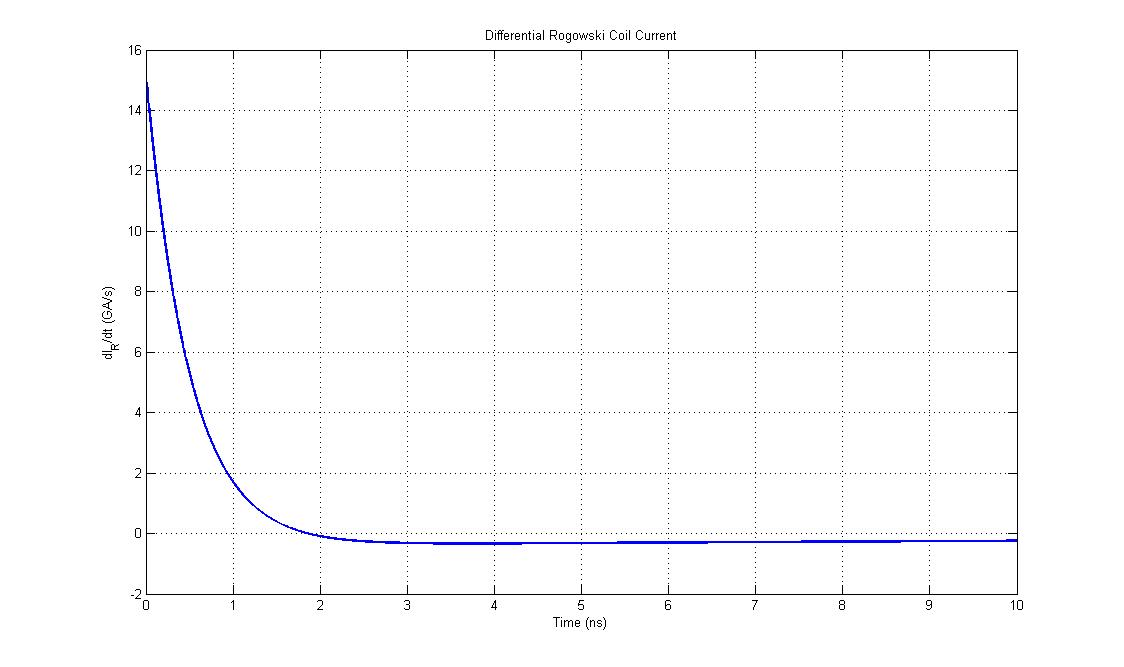


Figure 14: Differential Rogowski Coil Current

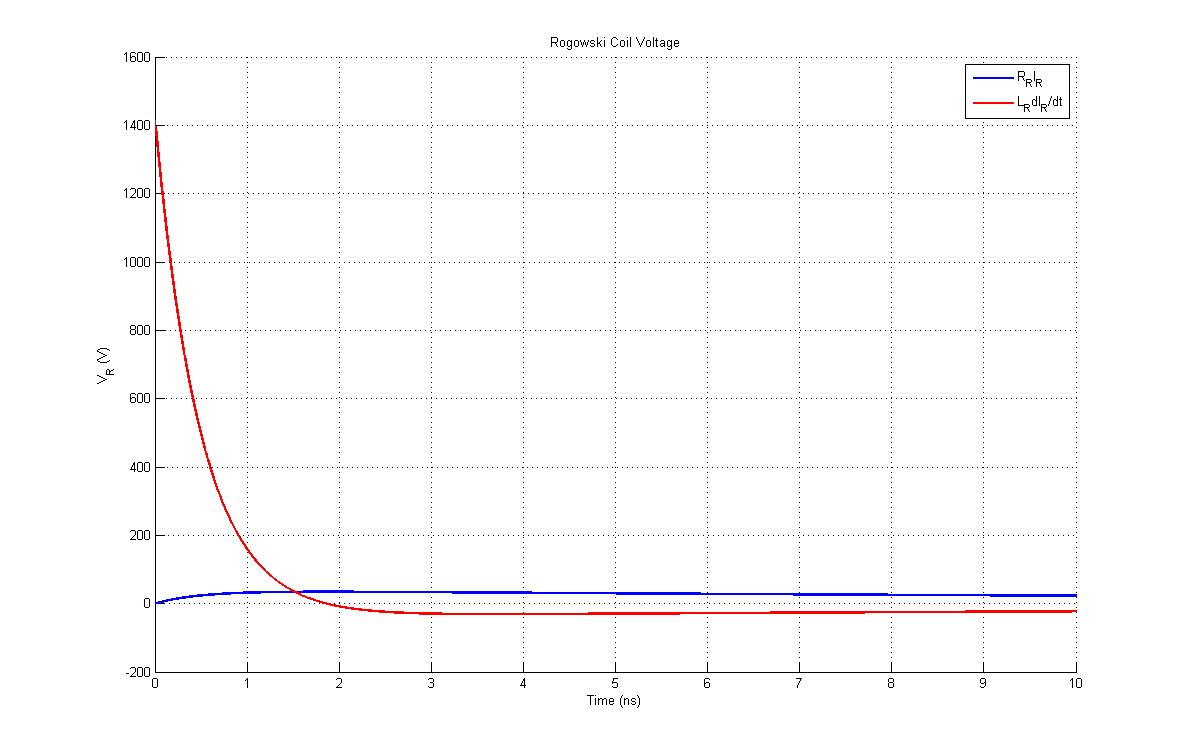


Figure 16: Rogowski Coil Characteristics

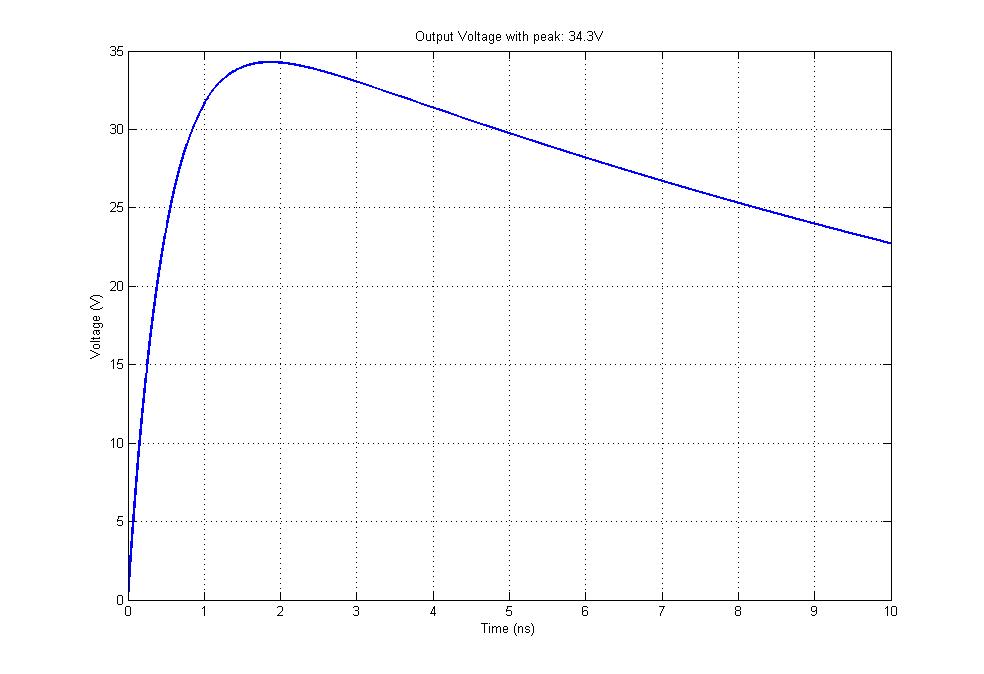


Figure 17: Output Voltage

The full code for this question is found in Appendix A: Q3.m.

# **Question 4**

Under initial conditions i.e. t=0, the charge on the capacitor Cb is assumed to be zero. The following code calculates the peak value of the rate of change of current and the peak value of the time rate-of-change of magnetic flux density produced during the capacitor bank discharge produces values of **600GA/s** and **1.88MT** respectively. The function ‘baxial’ is found in Appendix B: baxial.m.

%Bank Current Discharge

dI = (Vo-(Q./Cb)-(Rb.\*I))./Lb

%Peak Mag Flux Change

L = 40\*10^-2;

r = 0;

a = Pr\_tunnel./2;

z = 0;

Bz = baxial(a,r,z,L,I);

dB = Bz.\*dI

Along the width of the transmission line the magnetic flux along the centre axis of the tunnel is homogenous, therefore an arbitrary length of **30cm** was chosen for the magnetic pick up probe to maximise the area the probe covers in a homogenous field. A wire diameter of approximately **245µm** was chosen due to the frequency of the capacitor bank. To produce a maximum output of 200V requires the pick-up probe to have approximately **15 turns**. The following code calculates the self-inductance and resistance of the probe and produces values of **6.65nH** and **110mΩ** respectively.

%Inductance

indPr = indhelical(Nt,Pr\_diameter./2,Pr\_length)

%Resistance

freq = 1/(2\*pi\*sqrt(Lb\*Cb));

wire\_diameter = round(dskin(copper\_rho,freq).\*10^4).\*10^-4;

p = Pr\_length./Nt;

rPr = rhelical(Pr\_length,wire\_diameter,p,copper\_rho,Pr\_diameter/2)

The magnetic pick-up probe is acting as a differentiating probe if; . With a value of 50Ω chosen for means that this statement holds true and the probe is differentiating. The probe can be calibrated using a calibration shot; this involves measuring the current discharge to obtain the values for L and R of the system using the following equations;

The L and R values obtained should match that of the capacitor bank. See note(1) and (2) attached for the equation for the input voltage to the pick-up probe. Figure 18 shows the time variation of the magnetic pick-up probe for a step input of 1V.

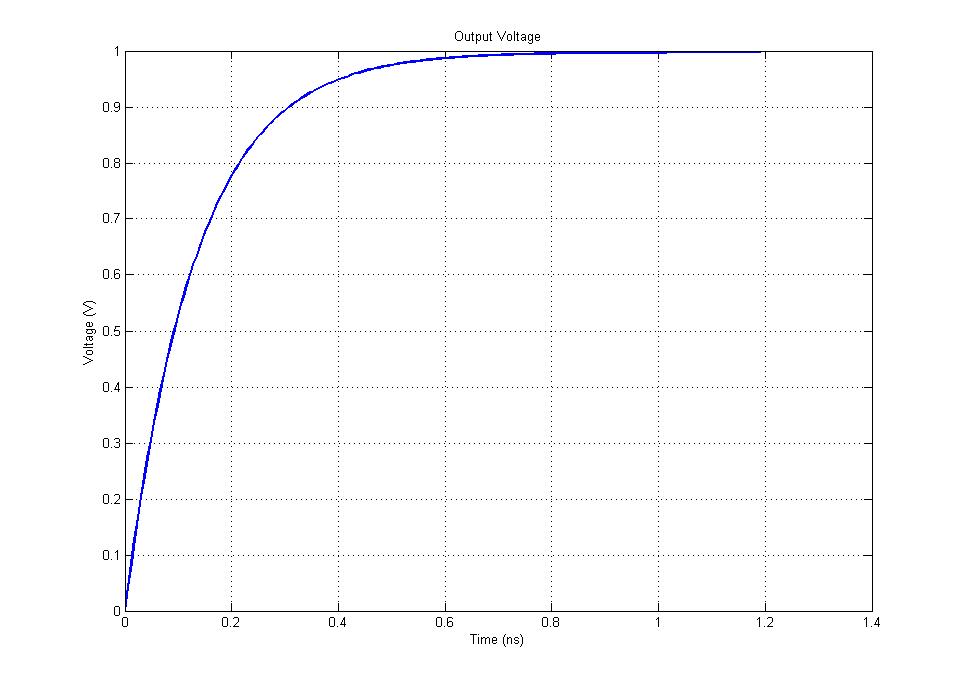


Figure 18: Voltage Response to 1V Step Input

The full code for this question is found in Appendix A: Q4.m.

# **Appendix A**

**Q1.m**

%% Fast Tansient Sensors - Q1 - Coursework 3

% B126949 - Tom Young

%% Pre - Cursor

clear all

clc

%% Constants

%Global constants

u0 = 4\*pi\*10^-7; % H/m

e0 = 8.85\*10^-12; % F/m

%Wire constants

evan\_rho = 1.38\*10^-6; %Ohm/m

evan\_heat = 435; %J/kg/K

evan\_dens = 8100; %kg/m^3;

%Generator Constants

Ns = 20;

CStage = 40\*10^-9;

V0 = 25\*10^3;

RStage = 10\*10^-3;

RLoad = 10\*10^3;

CLoad = 250\*10^-9;

%% Load Voltage

t=0:1\*10^-10:1\*10^-6;

VMax = V0.\*Ns;

RTotal = RStage.\*Ns;

CTotal = CStage./Ns;

Beta1 = 1./(RTotal.\*CLoad);

Beta2 = 1./(RLoad.\*CTotal);

VLoad=(VMax/((Beta1-Beta2)\*RTotal\*CLoad))\*(exp(-Beta2\*(t))-exp(-Beta1\*(t)));

rt = risetime(VLoad)./10^10;

VMaxPeak = max(VLoad)./10^3;

fband = (0.35)./rt;

%Plot VLoad

plot(t.\*10^6,VLoad./10^3,'Linewidth',2)

grid on

str = sprintf('Marx Generator Output Voltage for N\_{s} = %d and T\_{RISE} = %.2fns, V\_{peak} = %.2fkV',Ns,rt.\*10^9,VMaxPeak);

title(str);

ylabel('V (kV)');

xlabel('Time (\mus)');

%Voltage Sensor

rtvs = rt./5;

fbsens = (0.35)./rtvs;

%% Voltage Divider

%Low voltage arm

Zc = 50;

Z1 = Zc;

Z2 = Zc;

Z = Zc/2;

Tlv = 0;

%High voltage arm

Zhv = (((VMaxPeak\*10^3)/80)\*Zc)-Zc;

%Resistive Column

ColumnHeight = (VMaxPeak./5).\*10^-2;

wire\_radius = 12\*10^-6;

rmandrel = 0.05;

wire\_length = ((pi\*(wire\_radius)^2)\*Zhv)/evan\_rho;

wirel = wire\_length\*2;

Nt = wirel./(pi.\*rmandrel.\*2);

twirelength = wire\_length\*4;

%Energy

t=0:1\*10^-10:300\*10^-6;

VLoad=(VMax/((Beta1-Beta2)\*RTotal\*CLoad))\*(exp(-Beta2\*(t))-exp(-Beta1\*(t)));

figure

plot(t.\*10^6,VLoad./10^3,'Linewidth',2)

grid on

str = sprintf('Marx Generator Output Voltage for N\_{s} = %d and T\_{RISE} = %.2fns, V\_{peak} = %.2fkV',Ns,rt.\*10^9,VMaxPeak);

title(str);

ylabel('V (kV)');

xlabel('Time (\mus)');

% x0 = interp1(VLoad,t,0)

%Energy

tmax = 130\*10^-6;

Q = ((tmax\*(max(VLoad)^2))/(2\*Zhv));

%Wire Mass

wire\_vol = (pi\*(wire\_radius^2))\*twirelength; %m^3

wire\_mass = wire\_vol\*evan\_dens; %kg

%Temperature

deltaT = Q./(wire\_mass.\*evan\_heat)

% Capacitance was 16pF with graded column made of aluminium

cHV = 16\*10^-12;

%Connecting Cable

%Chosen wire length of 2m

width = 0.8;

connecting\_wire\_radius = 5\*10^-3;

indSys = indpoly('Rectangle',width,ColumnHeight,connecting\_wire\_radius)

indConnector = indpoly('RoundWire',width,connecting\_wire\_radius)/4

%Transfer functiona and response time from notes

rd\_sys = 1.2\*sqrt(indSys./cHV);

rd\_connector = 1.2\*sqrt(indConnector./cHV);

Ttotal = sqrt(((rd\_sys.\*cHV)+(rd\_connector.\*cHV))^2)

**Q2.m**

%% Fast Tansient Sensors - Q2 - Coursework 3

% B126949 - Tom Young

%% Pre - Cursor

clear all

clc

syms x

%% Constants

%Global constants

u0 = 4\*pi\*10^-7; % H/m

e0 = 8.85\*10^-12; % F/m

%Bank constants

Cb = 1\*10^-6;

Rb = 10\*10^-3;

Lb = 50\*10^-9;

Vo = 40\*10^3;

%Rod Diameter

Rd = 10\*10^-3;

%% Current Discharge

t = 0:1\*10^-10:25\*10^-6;

[Current\_Discharge, Diff\_Current\_Discharge, damping\_string] = dcdisch(Vo,Lb,Cb,Rb,t);

figure('name','RLC Current Discharge')

plot(t.\*10^6,Current\_Discharge.\*10^-3,'Linewidth',2)

grid on

xlabel('Time (\mus)')

ylabel('Current (kA)')

title(['Current Discharge for an ', damping\_string ,' RLC Circuit'])

figure('name','RLC Differential Current Discharge')

plot(t.\*10^6,Diff\_Current\_Discharge.\*10^-9,'Linewidth',2)

grid on

xlabel('Time (\mus)')

ylabel('Current/Time (GA/s)')

title(['Differential Current Discharge for an ', damping\_string ,' RLC Circuit'])

%% Probe Design

%Coil Parameters

major\_radius = 20\*10^-3

minor\_radius = 2\*10^-3

wire\_radius = 1\*10^-3

wire\_diameter = 2\*wire\_radius;

N = 10 %5-10 turns

copper\_rho = 1.7\*10^-8;

%Current Viewing Resistor

Rcvr = 50 %matches cable impedence

%Pitch

p = (2.\*pi.\*major\_radius)./N;

%Inductance

Induct\_pre = u0.\*minor\_radius.\*N;

Induct\_sum0 = 0.0007\*((log((2\*pi\*major\_radius)/p))^0);

Induct\_sum1 = 0.1773\*((log((2\*pi\*major\_radius)/p))^1);

Induct\_sum2 = -0.0322\*((log((2\*pi\*major\_radius)/p))^2);

Induct\_sum3 = 0.00197\*((log((2\*pi\*major\_radius)/p))^3);

Induct\_sum = Induct\_sum0 + Induct\_sum1 + Induct\_sum2 + Induct\_sum3;

Induct\_pre2 = (((pi\*minor\_radius)/p)+(log((2\*p)/(wire\_diameter)))-(5/4)-Induct\_sum);

RCoil\_inductance = Induct\_pre\*Induct\_pre2

%Resistance

freq = 1/(2\*pi\*sqrt(Lb\*Cb));

RCoil1 = N/(pi\*wire\_diameter);

RCoil2 = sqrt((copper\_rho\*pi\*freq\*u0)\*((p^2)+((2\*pi\*minor\_radius)^2)));

RCoil\_resistance = RCoil1\*RCoil2

%% Magnetic flux

%Magnetic flux time rate-of-change

k = u0.\*N.\*(major\_radius-sqrt((major\_radius^2)-(minor\_radius^2)));

flux = k.\*Current\_Discharge;

figure('name','Magnetic Flux')

plot(t.\*10^6,flux.\*10^6,'Linewidth',2)

grid on

xlabel('Time (\mus)')

ylabel('Magneic Flux (\muT)')

title(['Magnetic flux for an ', damping\_string ,' RLC Circuit'])

% Differential Magnetic Flux

dflux = k.\*Diff\_Current\_Discharge;

figure('name','Differntial Magnetic Flux')

plot(t.\*10^6,dflux,'Linewidth',2)

grid on

xlabel('Time (\mus)')

ylabel('Differential Magneic Flux (T/s)')

title(['Magnetic flux for an ', damping\_string ,' RLC Circuit'])

%% Rogowski Coil Current

Rrt = RCoil\_resistance + Rcvr;

Lr = RCoil\_inductance;

omega = sqrt((1./(Lb.\*Cb))-((Rb./(2.\*Lb))^2));

rogowski\_current = @(x) exp((Rrt./Lr).\*x).\*(k.\*(Vo./(omega.\*Lb).\*((omega.\*exp(-1.\*(Rb./(2.\*Lb)).\*x).\*cos(omega.\*x))-((Rb./(2.\*Lb)).\*exp(-1.\*(Rb./(2.\*Lb)).\*x).\*sin(omega.\*x)))));

t2 = 0:1\*10^-10:10\*10^-9;

RIntegral = zeros(0,length(t2));

for n=1:length(t2)

RIntegral(n) = integral(@(x)rogowski\_current(x),0,t2(n));

end

Rogowski\_Current = (exp(-(Rrt./Lr).\*t2)./Lr).\*RIntegral;

dRogowski\_Current\_prefix = (exp(-(Rrt./Lr).\*t2)./Lr).\*exp((Rrt./Lr).\*t2).\*(k.\*(Vo./(omega.\*Lb).\*((omega.\*exp(-1.\*(Rb./(2.\*Lb)).\*t2).\*cos(omega.\*t2))-((Rb./(2.\*Lb)).\*exp(-1.\*(Rb./(2.\*Lb)).\*t2).\*sin(omega.\*t2)))));

dRogowski\_Current = dRogowski\_Current\_prefix + (RIntegral.\*((-Rrt.\*exp((-Rrt./Lr).\*t2))./(Lr^2)));

figure('name','Rogowski Coil Voltage')

plot(t2.\*10^9,Rogowski\_Current,'b','Linewidth',2)

grid on

xlabel('Time (ns)')

ylabel('I\_{R} (A)')

title('Rogowski Coil Current')

figure('name','Differential Rogowski Coil Voltage')

plot(t2.\*10^9,dRogowski\_Current.\*10^-9,'b','Linewidth',2)

grid on

xlabel('Time (ns)')

ylabel('dI\_{R}/dt (GA/s)')

title('Differential Rogowski Coil Current')

figure('name','Rogowski Coil Characteristics')

hold on

plot(t2.\*10^9,Rogowski\_Current.\*Rrt,'b','Linewidth',2)

plot(t2.\*10^9,dRogowski\_Current.\*Lr,'r','Linewidth',2)

grid on

xlabel('Time (ns)')

ylabel('V\_{R} (V)')

title('Rogowski Coil Voltage')

legend('R\_{R}I\_{R}','L\_{R}dI\_{R}/dt')

hold off

%% Input Voltage

%Differential Magnetic Flux

Vin = k.\*Diff\_Current\_Discharge;

figure('name','Input Voltage')

plot(t.\*10^6,Vin,'Linewidth',2)

grid on

xlabel('Time (\mus)')

ylabel('Voltage (V)')

title('Rogowski Coil Output Voltage')

%% Oscilloscope Voltage

% Integrator Design

Ri = 50\*10^-3;

Ci = 500\*10^-9;

tau = Ri.\*Ci;

Vosc = (Rogowski\_Current.\*k)./tau;

figure('name','Rogowski Coil Voltage')

plot(t2.\*10^9,Vosc,'b','Linewidth',2)

Voutp = max(Vosc);

grid on

xlabel('Time (ns)')

ylabel('Voltage (V)')

str = sprintf('Integrator Circuit Output Voltage with peak: %.1fV',Voutp);

title(str)

**Q3.m**

%% Fast Tansient Sensors - Q2 - Coursework 3

% B126949 - Tom Young

%% Pre - Cursor

clear all

clc

syms x

%% Constants

%Global constants

u0 = 4\*pi\*10^-7; % H/m

e0 = 8.85\*10^-12; % F/m

%Bank constants

Cb = 100\*10^-9;

Rb = 50;

Lb = 25\*10^-9;

Vo = 60\*10^3;

%Rod Diameter

Rd = 10\*10^-3;

%% Current Discharge

t = 0:1\*10^-12:10\*10^-9;

[Current\_Discharge, Diff\_Current\_Discharge, damping\_string] = dcdisch(Vo,Lb,Cb,Rb,t);

figure('name','RLC Current Discharge')

plot(t.\*10^9,Current\_Discharge.\*10^-3,'Linewidth',2)

grid on

xlabel('t (ns)')

ylabel('Current (kA)')

title(['Current Discharge For ', damping\_string ,' RLC Circuit'])

figure('name','RLC Differential Current Discharge')

plot(t.\*10^9,Diff\_Current\_Discharge.\*10^-9,'Linewidth',2)

grid on

xlabel('t (ns)')

ylabel('Current/t (GA/s)')

title(['Differential Current Discharge For ', damping\_string ,' RLC Circuit'])

%% Probe Design

%Coil Parameters

major\_radius = 100\*10^-3;

minor\_radius = 2.5\*10^-3;

wire\_radius = 1.5\*10^-3;

wire\_diameter = 2\*wire\_radius;

N = 15; %5-10 turns

copper\_rho = 1.7\*10^-8;

%Current Viewing Resistor

Rcvr = 5; %Matches cable impedence

%Pitch

p = (2.\*pi.\*major\_radius)./N;

%Inductance

Induct\_pre = u0.\*minor\_radius.\*N;

Induct\_sum0 = 0.0007\*((log((2\*pi\*major\_radius)/p))^0);

Induct\_sum1 = 0.1773\*((log((2\*pi\*major\_radius)/p))^1);

Induct\_sum2 = -0.0322\*((log((2\*pi\*major\_radius)/p))^2);

Induct\_sum3 = 0.00197\*((log((2\*pi\*major\_radius)/p))^3);

Induct\_sum = Induct\_sum0 + Induct\_sum1 + Induct\_sum2 + Induct\_sum3;

Induct\_pre2 = (((pi\*minor\_radius)/p)+(log((2\*p)/(wire\_diameter)))-(5/4)-Induct\_sum);

RCoil\_inductance = Induct\_pre\*Induct\_pre2;

%Resistance

freq = 1/(Rb.\*Cb);

% freq = 1./(Rb.\*Cb);

RCoil1 = N/(pi\*wire\_diameter);

RCoil2 = sqrt((copper\_rho\*pi\*freq\*u0)\*((p^2)+((2\*pi\*minor\_radius)^2)));

RCoil\_resistance = RCoil1\*RCoil2;

%% Magnetic flux

%Magnetic flux t rate-of-change

k = u0.\*N.\*(major\_radius-sqrt((major\_radius^2)-(minor\_radius^2)));

flux = k.\*Current\_Discharge;

figure('name','Magnetic Flux')

plot(t.\*10^9,flux.\*10^6,'Linewidth',2)

grid on

xlabel('t (ns)')

ylabel('Magneic Flux (\muT)')

title(['Magnetic flux for an ', damping\_string ,' RLC Circuit'])

%Differential Magnetic Flux

dflux = k.\*Diff\_Current\_Discharge;

figure('name','Differntial Magnetic Flux')

plot(t.\*10^9,dflux.\*10^-3,'Linewidth',2)

grid on

xlabel('t (ns)')

ylabel('Differential Magneic Flux (kT/s)')

title(['Magnetic flux for an ', damping\_string ,' RLC Circuit'])

%% Rogowski Coil Current

Rrt = RCoil\_resistance + Rcvr;

Lr = RCoil\_inductance;

omegadot1 = (Rb./(2\*Lb))^2;

omegadot2 = 1./(Lb\*Cb);

omegadot = sqrt(omegadot1-omegadot2);

rogowski\_current = @(x) exp((Rrt./(Lr)).\*x).\*(k.\*(Vo./(2.\*omegadot.\*Lb).\*((exp(-1.\*(Rb./(2.\*Lb)).\*x).\*(omegadot.\*exp(omegadot.\*x)+omegadot.\*exp(-omegadot.\*x)))+(-1.\*(Rb./(2.\*Lb)).\*exp(-1.\*(Rb./(2.\*Lb)).\*x).\*(exp(omegadot.\*x)-exp(-omegadot.\*x))))));

t2 = 0:1\*10^-12:10\*10^-9;

RIntegral = zeros(0,length(t2));

for n = 1:length(t2)

RIntegral(n) = integral(@(x)rogowski\_current(x),0,t2(n));

end

Rogowski\_Current = (exp(-(Rrt./Lr).\*t2)./Lr).\*RIntegral;

dRogowski\_Current\_prefix = (exp(-(Rrt./Lr).\*t2)./Lr).\*exp((Rrt./Lr).\*t2).\*(k.\*(Vo./(2.\*omegadot.\*Lb).\*((exp(-1.\*(Rb./(2.\*Lb)).\*t2).\*(omegadot.\*exp(omegadot.\*t2)+omegadot.\*exp(-omegadot.\*t2)))+(-1.\*(Rb./(2.\*Lb)).\*exp(-1.\*(Rb./(2.\*Lb)).\*t2).\*(exp(omegadot.\*t2)-exp(-omegadot.\*t2))))));

dRogowski\_Current = dRogowski\_Current\_prefix + (RIntegral.\*((-Rrt./Lr).\*exp(-(Rrt./Lr).\*t2)./Lr));

figure('name','Rogowski Coil Voltage')

plot(t2.\*10^9,Rogowski\_Current,'b','Linewidth',2)

grid on

xlabel('Time (ns)')

ylabel('I\_{R} (A)')

title('Rogowski Coil Current')

figure('name','Differential Rogowski Coil Voltage')

plot(t2.\*10^9,dRogowski\_Current.\*10^-9,'b','Linewidth',2)

grid on

xlabel('Time (ns)')

ylabel('dI\_{R}/dt (GA/s)')

title('Differential Rogowski Coil Current')

figure('name','Rogowski Coil Characteristics')

hold on

plot(t2.\*10^9,Rogowski\_Current.\*Rrt,'b','Linewidth',2)

plot(t2.\*10^9,dRogowski\_Current.\*Lr,'r','Linewidth',2)

grid on

xlabel('Time (ns)')

ylabel('V\_{R} (V)')

title('Rogowski Coil Voltage')

legend('R\_{R}I\_{R}','L\_{R}dI\_{R}/dt')

hold off

%% Output Voltage

Vosc = (Rogowski\_Current.\*Rrt);

figure('name','Rogowski Coil Voltage')

plot(t2.\*10^9,Vosc,'b','Linewidth',2)

Voutp = max(Vosc);

grid on

xlabel('Time (ns)')

ylabel('Voltage (V)')

str = sprintf('Output Voltage with peak: %.1fV',Voutp);

title(str)

**Q4.m**

%% Fast Tansient Sensors - Q4 - Coursework 3

% B126949 - Tom Young

%% Pre - Cursor

clear all

clc

%% Constants

%Global constants

u0 = 4\*pi\*10^-7; % H/m

e0 = 8.85\*10^-12; % F/m

copper\_rho = 1.7\*10^-8;

%Bank Constants

Cb = 100\*10^-6;

Trans\_width = 40\*10^-2;

Vo = 30\*10^3;

Lb = 50\*10^-9;

Rb = 0;

Q = 0;

I = 1;

%Probe constants

Pr\_tunnel = 15\*10^-3;

Pr\_diameter = 3\*10^-3;

Pr\_MaxV = 200;

Pr\_area = pi.\*(Pr\_diameter./2)^2;

%% Current Discharge

%Bank Current Discharge

dI = (Vo-(Q./Cb)-(Rb.\*I))./Lb

%Peak Mag Flux Change

L = 40\*10^-2;

r = 0;

a = Pr\_tunnel./2;

z = 0;

Bz = baxial(a,r,z,L,I);

dB = Bz.\*dI

%% Probe Design

%Coil Length

z = 0:1\*10^-4:20\*10^-2;

BzInc = zeros(0,length(z));

for n = 1:length(z)

BzInc(n) = baxial(a,r,z(n),L,I);

end

figure

plot(z.\*10^2,BzInc.\*10^6,'Linewidth',2)

xlabel('Distance along transmission line (cm)')

ylabel('Flux (\muT)')

title('Flux Along Transmission Line Width')

grid on

Pr\_length = 30\*10^-2;

%Number of turns

Nt = Pr\_MaxV./(Pr\_area.\*dB)

%Inductance

indPr = indhelical(Nt,Pr\_diameter./2,Pr\_length)

%Resistance

freq = 1/(2\*pi\*sqrt(Lb\*Cb));

wire\_diameter = round(dskin(copper\_rho,freq).\*10^4).\*10^-4;

p = Pr\_length./Nt;

rPr = rhelical(Pr\_length,wire\_diameter,p,copper\_rho,Pr\_diameter/2)

%Capacitance

cPr = ccoil(Pr\_diameter/2,p,wire\_diameter/2)./(Nt-1)

%Demonstration

Rt = 50;

Rsys = 50+rPr;

Xl = (2.\*pi.\*freq).\*indPr;

%% Vout

Emf = 1;

Eqn = [((indPr\*cPr)) (((indPr/Rt)+(rPr\*cPr))) (1+(rPr/Rt))];

Vout = tf(Emf,Eqn);

[VOutput, t] = step(Vout);

figure

plot(t\*10^9,VOutput,'Linewidth',2)

grid on

title('Output Voltage')

xlabel('Time (ns)')

ylabel('Voltage (V)')

# **Appendix B**

**indpoly.m**

function [ indpoly ] = indpoly( polygon\_type, varargin )

%indpoly Inductance of a number of polygon shapes

%

% Inductance of a polygon shape s defined as 'polygon'. For corret

% calcualtion of sel inductance input are defined as;

%

% 'RoundWire', wire\_length, wire\_radius

% 'RoundWireHF', wire\_length, wire\_radius

% 'Rectangle', side\_length1, side\_length2, wire\_radius

% 'Planar', perimeter, area, wire\_radius

% 'EqTriangle', side\_length, wire\_radius

% 'Square', side\_length, wire\_radius

% 'Pentagon', side\_length, wire\_radius

%% Constants

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

%% Equations

switch polygon\_type

case 'RoundWire'

wire\_length = varargin{1};

wire\_radius = varargin{2};

indpoly\_pre = (u\_0\*wire\_length)/(2\*pi);

indpoly = indpoly\_pre\*(log((2\*wire\_length/wire\_radius))-(3/4));

case 'RoundWireHF'

wire\_length = varargin{1};

wire\_radius = varargin{2};

indpoly\_pre = (u\_0\*wire\_length)/(2\*pi);

indpoly = indpoly\_pre\*(log(2\*wire\_length/wire\_radius)-1);

case 'Rectangle'

a = varargin{1};

b = varargin{2};

r = varargin{3};

d = sqrt(a^2 + b^2);

l = 2\*(a + b);

indpoly\_pre = u\_0/pi;

indpoly = indpoly\_pre\*(a\*log((2\*a\*b)/((a+d)\*r))+b\*log((2\*a\*b)/((b+d)\*r))+2\*d - (7/8)\*l);

case 'Planar'

P = varargin{1};

A = varargin{2};

r = varargin{3};

indpoly\_pre = (u\_0\*P)/(2\*pi);

indpoly = indpoly\_pre\*(log((2\*P)/r)+(1/4)-log((P^2)/A));

case 'EqTriangle'

s = varargin{1};

r = varargin{2};

indpoly\_pre = (3\*u\_0\*s)/(2\*pi);

indpoly = indpoly\_pre\*(log(s/r)-1.15546);

case 'Square'

s = varargin{1};

r = varargin{2};

indpoly\_pre = (4\*u\_0\*s)/(2\*pi);

indpoly = indpoly\_pre\*(log(s/r)-0.52401);

case 'Pentagon'

s = varargin{1};

r = varargin{2};

indpoly\_pre = (5\*u\_0\*s)/(2\*pi);

indpoly = indpoly\_pre\*(log(s/r)-0.15914);

otherwise

disp(' Incorrect input statements, type help indpoly for supported types')

end

end

**dcdisch.m**

function [ cdisch, dcdisch, damping\_string] = dcdisch( initial\_voltage, bank\_inductance, bank\_capacitance, bank\_resistance, time )

%CDISCH Current discharge and its differential of a RLC circuit

% Detailed explanation goes here

%% Constants

damping = 2.\*sqrt(bank\_inductance./bank\_capacitance);

% gamma = 0.5.\*bank\_resistance.\*sqrt(bank\_capacitance./bank\_inductance);

omega = sqrt((1./(bank\_inductance.\*bank\_capacitance))-((bank\_resistance./(2.\*bank\_inductance))^2));

omegadot1 = (bank\_resistance./(2\*bank\_inductance))^2;

omegadot2 = 1./(bank\_inductance\*bank\_capacitance);

omegadot = sqrt(omegadot1-omegadot2);

% omega = (1./sqrt(bank\_inductance.\*bank\_capacitance)).\*sqrt(1-(gamma^2));

exp\_component = -1.\*(bank\_resistance./(2.\*bank\_inductance)).\*time;

%% Equations

if bank\_resistance < damping

cdisch\_prefix = initial\_voltage./(omega.\*bank\_inductance);

cdisch = cdisch\_prefix.\*exp(exp\_component).\*sin(omega.\*time);

dcdisch\_prefix = initial\_voltage./(omega.\*bank\_inductance);

dcdisch = dcdisch\_prefix.\*((omega.\*exp(exp\_component).\*cos(omega.\*time))+((-1.\*(bank\_resistance./(2.\*bank\_inductance))).\*exp(exp\_component).\*sin(omega.\*time)));

damping\_string = 'Under Damped';

end

if bank\_resistance == damping

cdisch\_prefix = (initial\_voltage.\*time)./bank\_inductance;

cdisch = cdisch\_prefix.\*exp(exp\_component);

dcdisch = ((initial\_voltage./bank\_inductance).\*exp(exp\_component))+(-bank\_resistance.\*((initial\_voltage.\*time)./(2.\*bank\_inductance.\*bank\_inductance)).\*exp(exp\_component));

damping\_string = 'Critcally Damped';

end

if bank\_resistance > damping

cdisch\_prefix = initial\_voltage./((2.\*omegadot).\*bank\_inductance);

cdisch = cdisch\_prefix.\*exp(exp\_component).\*(exp(omegadot.\*time)-exp(-omegadot.\*time));

dcdisch = cdisch\_prefix.\*((exp(exp\_component).\*(omegadot.\*exp(omegadot.\*time)+omegadot.\*exp(-omegadot.\*time)))+(-1.\*(bank\_resistance./(2.\*bank\_inductance)).\*exp(exp\_component).\*(exp(omegadot.\*time)-exp(-omegadot.\*time))));

damping\_string = 'Over Damped';

end

end

**baxial.m**

function [ baxial ] = baxial( a,r,z,l,I )

%UNTITLED3 Summary of this function goes here

% Detailed explanation goes here

%% Constants

u0 = 4\*pi\*10.^-7;

y = a./l;

d = r./a;

zd = z./(l./2);

%% Axial Component

Bz\_pre = (u0.\*I)./(4.\*pi.\*l);

axial\_flux = @(x) ((1-(d.\*cos(x)))./(1+(d.^2)-(2.\*d.\*cos(x)))).\*(((1-zd)./sqrt((y.^2).\*(1+(d.^2)-(2.\*d.\*cos(x)))+((1-zd).^2)./4))+((1+zd)./sqrt((y.^2).\*(1+(d.^2)-(2.\*d.\*cos(x)))+((1-zd).^2)./4)));

baxial = Bz\_pre.\*integral(@(x)axial\_flux(x),0,pi);

%% Radial Component

% Br\_pre = (u0.\*I.\*a)./(4.\*pi.\*(l^2));

% radial\_flux = @(x) (((cos(x))./sqrt((y.^2).\*(1+(d.^2)-(2.\*d.\*cos(x)))+((1-zd).^2)./4))-((cos(x))./sqrt((y.^2).\*(1+(d.^2)-(2.\*d.\*cos(x)))+((1+zd).^2)./4)));

% bradial = Br\_pre.\*integral(@(x)radial\_flux(x),0,pi);

end

**indhelical.m**

function [ indhelical ] = indhelical( coil\_turns, coil\_radius, coil\_length )

%indhelical.

%% Constants

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

%% Equations

if coil\_length >= coil\_radius

K = coil\_length/(coil\_length+(0.9\*coil\_radius));

else

K = coil\_length/((11/10)\*coil\_length+(0.8\*coil\_radius));

end

indhelical = ((u\_0\*pi\*(coil\_turns^2)\*(coil\_radius^2))/coil\_length)\*K;

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

**ccoil.m**

function [ ccoil ] = ccoil( coil\_radius, pitch, wire\_radius, varargin )

%CCOIL Capacitance between two single turn coils

% Capacitance between two single turn coils of equal radius made from a

% wire with a constant circular radius and possitiond along the same axis

% at a distance p either for non insulated or insulated wires.

%% Constants

e\_0 = 8.854\*10^-12;

%% Equations

if length(varargin) > 0

m\_perm = [varargin{1}];

t = [varargin{2}];

F1 = F(pitch,m\_perm,t,wire\_radius);

ccoil\_num = 2\*pi^2\*e\_0\*coil\_radius;

ccoil\_dem\_1 = (1+(t/wire\_radius))^(2/m\_perm);

ccoil\_denum = log(F1+sqrt((F1^2)-ccoil\_dem\_1));

ccoil = ccoil\_num/ccoil\_denum;

else

ccoil\_num = 2\*pi^2\*e\_0\*coil\_radius;

ccoil\_denum = log((pitch/(2\*wire\_radius))+sqrt(((pitch/(2\*wire\_radius))^2)-1));

ccoil = ccoil\_num/ccoil\_denum;

end

end

function [ F ] = F(p, e\_r, t ,r)

% F = p/(2\*r\*((1+(t/r)^((m-1)/m))));

F\_num = p;

F\_denum1 = (1+(t/r))^((e\_r-1)/e\_r);

F\_denum2 = 2\*r\*F\_denum1;

F = F\_num/F\_denum2;

end