EE564 First Project: Transformer Design a for X-Ray Device

Table of Contents

ID

NAME : Mehmet Kaan Mutlu

STUDENT NUMBER : 2121408

E-mail : kaan.mutlu@metu.edu.tr


```
function []=meka_mutlu_XRAY()
```

Specifications

- **Single Phase, High Frequency High Voltage Transformer**
- **Primary Winding Voltage:** ± 417 V (peak to peak 834 V for pulsing)
- **Secondary Winding Voltage:** ± 12.5 kV (peak to peak 25 kV for pulsing)
- **Rated Power:** 30 kW (for maximum 100 millisecond)
- **Switching Frequency:** Minimum 100 kHz
- **Ambient Temperature:** 0-40 °C

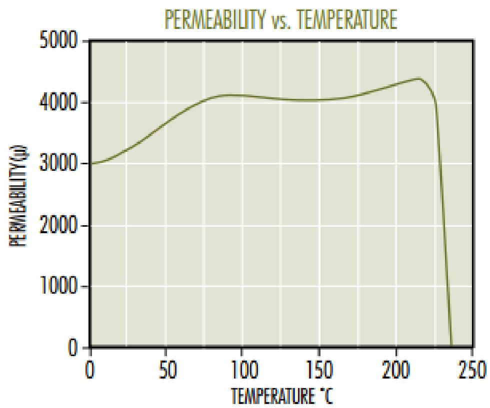
```
Prated       = 30e3; % Rated power [W]
fs           = 100e3; % switching frequency [Hz]

Vp_peak      = 417; % Primary side peak voltage [V]
Vp_fund_peak= Vp_peak*4/pi; % Peak of fundamental of primary voltage [V]
Vp_f_rms     = Vp_fund_peak/sqrt(2); % RMS value of fundamental [V]

Vs_peak      = 12.5e3; % Secondary side peak voltage [V]
Vs_fund_peak= Vs_peak*4/pi; % Peak of fundamental of secondary voltage [V]
Vs_f_rms     = Vs_fund_peak/sqrt(2); % RMS value of fundamental [V]
```

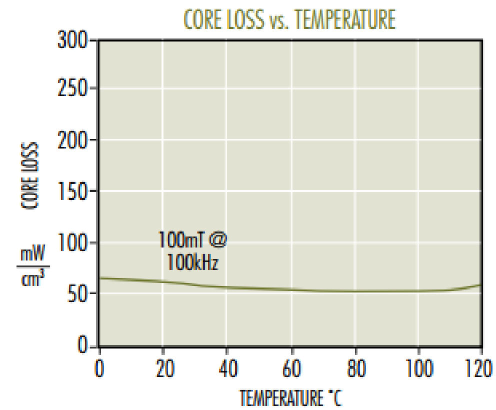
```
Ip_rms       = Prated/Vp_f_rms; % Primary side RMS current [A]
Is_rms       = Prated/Vs_f_rms;  % Secondary side RMS current [A]
```

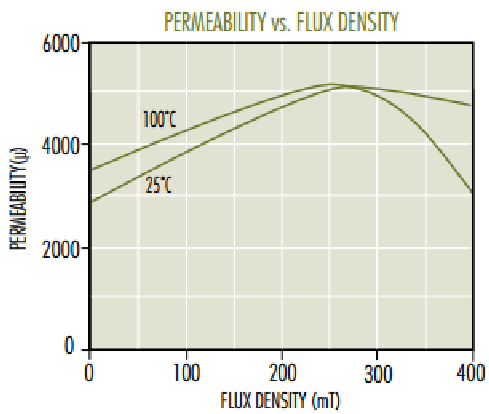
Choosing Initial Material

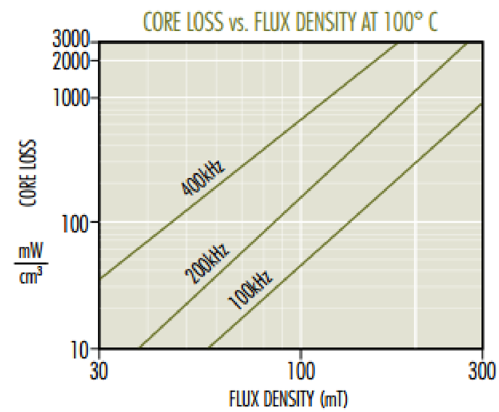
First step of transformer design is selecting an appropriate core material. After some researches on internet and company application guides, it is decided to use a ferrite material for XRAY transformer application at 100kHz switching frequency.

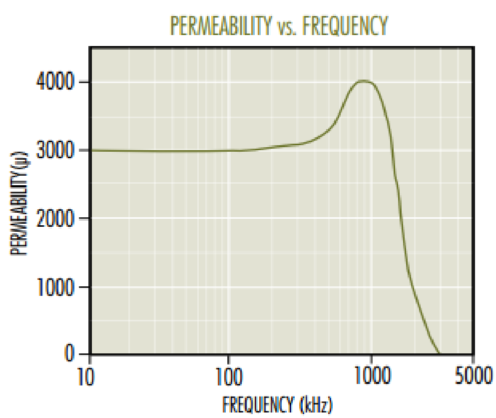
After this decision, Magnetics' ferrite catalog is read and different types of materials are compared. In that comprasion, power losses of materials at 25°C and 100kHz is used as basic elimination parameter and it is decided to use T material. It is possible to find its parametets below:











Choosing Operation Flux Density

For the second phase of design, it is going to be choosen operation flux density. T material's saturation flux density is 470mT for this project's defined temperature range. Our value should be smaller than saturation point. But how much? Let's consider over the formula below:

$$e = -\frac{2*\pi}{\sqrt{2}} N 2\pi f B_{peak} * A$$

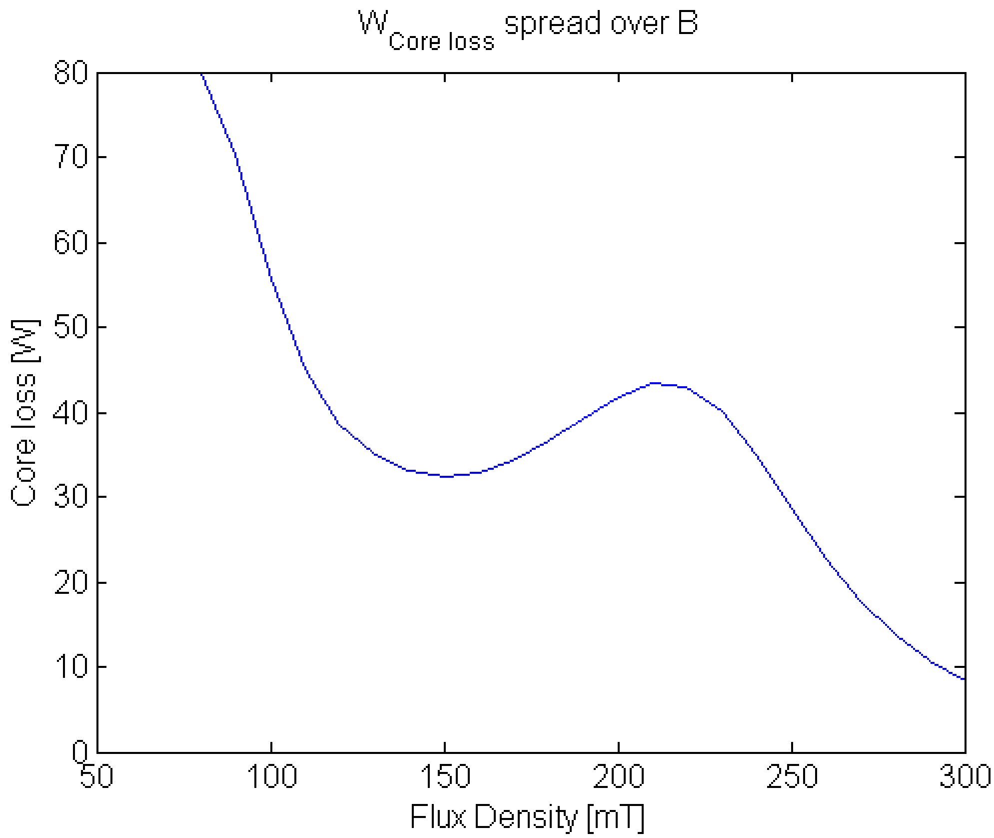
If only selectable parameters are considered, it is possible to see the trade-off between number of turns, flux density and area. Selecting high number of turns come with dificulties of cabling and copper losses. Cable size is decided over current values so it is constant in this discussion. Area is important for transformer's size and weight values. It also effects cable length and core loss (over weight). Flux density is directly related with core losses.

As B is increased, core loss is increased (nonlinearly). If we take BxA value constant, then increasing B will decrease A and therefore volume and weight will be less and it means less core loss. So here, by assuming area and weight are proportional, an optimization will be made over core loss.

```
B_opr = optimize_B();
```

Flux density vs core loss graphic has some missing points due to its nonlinearity. To be able to find required missing points Lagrange polynomial method is used (Function used in this project was written by me during my 3rd class undergraduate studies). After completion is done, assuming area effects weight proportionally, a basic multiplication is made. In coreloss plot unit and magnitude aren't considered but result shows how core loss changes as operation point of flux density is increased.

```
plot(B_req, Coreloss);
set(gca,'FontSize',12);
xlabel('Flux Density [mT]');
ylabel('Core loss [W]');
title('W_{Core loss} spread over B');
```



To be able to operate in a condition with less core loss 300mT is selected as operation flux density.

Determination of Core Dimensions & Number of turns

Comments

Determination of Core and Copper Losses

Comments

Determination of Operating Temperature

Comments

Determination of mass, cost etc.

Comments

```
function [output] = optimize_B()

    B_given = [80 90 100 200 300];
    Coreloss_coef = [25 32 45 120 900];
```

```
        B_req = 80:10:300;
        for i=1:length(B_req)
            Coreloss_coef2(i)=lagrange(B_given, Coreloss_coef, B_req(i));
            Coreloss(i) = B_req(i)*Coreloss_coef2(1)/Coreloss_coef2(i);

        end;
        output = B_req(find(Coreloss==min(Coreloss)));
    end

    function L=lagrange(x,y,k)
        n=length(x);
        l=1;
        L=0;
        for i=1:n
            for j=1:n
                if i~=j
                    l=l*(k-x(j))/(x(i)-x(j));
                end;
            end;
            L=L+l*y(i);
            l=1;
        end;
    end;

end
```

Published with MATLAB® R2013a