Table of Contents

Transformer design m-file

```
% Joseph Arsenault
% Date  12/13/2017
% design two transformers:
% Don't change anything in this code cell

n = 12;             % required turns ratio
Vrms = 120;         % rms input voltage
Irms = 0.1;         % rated rms input current
Isb = 5e-3;         % rms standby current

Muo = pi*4e-7;      % permeability of free space
GammaC = [7650 4982];  % mass density of core materials in Kg/m^3
% GammaC(1) = steel mass density, Gammac(2) = ferrite mass density
GammaW = 8.95e3;    % mass density of wire material in Kg/m^3
RhoW = 1.7241e-8; % resistivity of wire at 20degC in Ohm*m

Jmax = 5;           % maximum wire currend density in Arms/mm^2

Kfill = 0.5;        % maximum normalized amount of winding space
 occupied by covered wire
```

Functions given by Duane Hanselman

```
% Functions you can call in your design

% Leakage flux fraction for computation of leakage inductances
% Bm = flux density in the core when the input current is at its peak
 RMS value
alpha = @(Bm) 0.01 + Bm/20;
% Example: alpha(1) = 0.06 = 6% Bm is in tesla
% As the steel saturates and becomes less permeable, the more flux
 leaks out

% COPPER WIRE Functions
 ---------------------------------------------------
% bare wire diameter in mm from Gauge G
DiamWb = @(G) 8.251463*(0.8905257)^G;
% Example: DiamW(20) returns diameter of 20 gage wire in mm

% covered wire diameter in mm from Gauge G
DiamWc = @(G) 1.15*DiamWb(G);

% bare wire cross-sectional area in m^2 from Gauge G
```

```
AreaWb = @(G) pi*(DiamWb(G)/2000)^2;

% covered wire cross-sectional area in m^2 from Gauge G
AreaWc = @(G) pi*(DiamWc(G)/2000)^2;

% wire Gauge G from bare wire diameter in mm
GaugeWd = @(d) log(d/8.251463)./log(0.8905257);

% wire Gauge G from bare wire cross-sectional area in (mm)^2
GaugeWA = @(A) log((2*sqrt(A/pi))/8.251463)./log(0.8905257);

% Wire Resistance for length z in meters and Wire Gauge G
Rwire = @(z,G) RhoW*z./(pi*(DiamWb(G)/2000)^2);


% Core Parameter Data---------------------------------------------
cp1 = [3.8 26];
cp2 = [2.17 12.2];
cp3 = [396.2 206.2];
cp4 = [9.55 1.06e-4];

% Magnetic CORE Functions
 -------------------------------------------------

% Two materials are available, as selected by variable k
% k = 1 is steel
% k = 2 is ferrite

% BH curve of core material, H as a function of B
Hcore = @(B,k) B.*(cp1(k)*exp(cp2(k)*B.^2) + cp3(k));
% Example: Hcore(0.2) returns H for B=0.2 Tesla

% BH curve of core material, B as a function of H
Bcore = @(H,k) fzero(@(B) Hcore(B,k)-H,0.15);
% Example: Bcore(450) returns B for H=450 A/m

% dB(H)/dH of core material as a function of B
dBdHb = @(B,k) 1./(cp1(k)*exp(cp2(k)*B.^2) .*(2*cp2(k)*B.^2 + 1) +
 cp3(k));

% Permeability of core material as a function of B
Muc = @(B,k) (B./Hcore(B,k));
% if you already have B and H, this is simply Muc = B/H

% Relative Permeability of the core material as a function of B
Mucr = @(B,k) Muc(B,k)/Muo;

% Core loss density in Watts/m^3, f in Hz, B in Tesla
Pcore = @(f,B,k) cp4(k)*(f.*B)^2;
```

Transformer Design

```
%Find area of wire necessary for given current density, A/(A/mm^2)
```

```
Primary_area = Irms/Jmax;
asd = GaugeWA (Primary_area);

%Floors value as decimals are nonsensical values
Primary_gauge =  floor(asd);
Area_secondary = (n*Irms)/Jmax;
Gauge_secondary = floor(GaugeWA(Area_secondary));

%Prompts user for frequency,Steel core for 60hz, Ferrite core for 60
 kHz,
%changes width accordingly, checks to make sure answer is valid.
checkprompt = 1;
while checkprompt == 1
prompt = 'What is the frequency? Choose 60 or 60,000. ';
if ~exist('f', 'var')
f = input(prompt);
end
Omega = 2*pi*f;
if f == 60
     k_picked = 1;
     width =  29e-3;
     checkprompt = 0;
elseif f == 6e4
    k_picked = 2;
    width = 6.7e-3;
    checkprompt = 0;
else
    checkprompt = 1;
end
end
y = 1.5 * width;
x = .5 * width;
height = y + width;
length = 2*x + 2* width;

%Area of inner window
Inner_area = x * y;

%Volume of inner window
Inner_volume = Inner_area * width;
i=0;

%%Magnetic model
Horizontal_core_length = length/2 - 2*(width/4);
Vertical_core_length = height - 2*(width/4);
Length_core_parallel = 2*Horizontal_core_length +
 Vertical_core_length;
Length_core_series = height - 2*(width/4);
Full_core_length = 2*Length_core_parallel + Length_core_series;

%Initiliaze wire area, loops to find optimal wire area
Wire_area = 0;
while Wire_area < (Kfill*Inner_area)
    i = i+1;
```

```
    Wire_area = 12*AreaWc(Primary_gauge)*i +
 AreaWc(Gauge_secondary)*i;
    Primary_turns = 12*i;
    secondary_turns = i;

%Turn length parameter, also known as bobbin parameter
Bobbin = 4*width;

%Primary Wire Length
Primary_wire_length = Bobbin * Primary_turns;
Wire_volume_primary = Primary_wire_length * AreaWc(Primary_gauge);
Primary_mass = GammaW * Wire_volume_primary;

%Secondary Wire Length
Wire_length_secondary = Bobbin * secondary_turns;
Wire_volume_secondary = Wire_length_secondary *
 AreaWc(Gauge_secondary);
Mass_secondary = GammaW * Wire_volume_secondary;

%Magneto Motor Force (MMF),  F = Ni
MMF = Primary_turns*Irms + secondary_turns*n*Irms;

%Finding H, the ratio of F to core length
H = MMF/Full_core_length;

%B from H and core material
B = Bcore(H,k_picked);

%Mu core from B
Mu_core = Muc(B,k_picked);

%Permeance of middle branch
Permeance_series = (Mu_core*width^2)/Length_core_series;

%Permeance of parallel branch
Permeance_branch = (Mu_core*width*(width/2))/Length_core_parallel;

%Reluctance of core is 1/Permeance
Total_reluctance = (1/Permeance_series) +(1/(2*Permeance_branch));
Total_permeance = 1/Total_reluctance;

%%Electric model
% Lm = N^2 P
Lm = Primary_turns^2 * Total_permeance;

%Alpha is linkage coeffecient, let alpha = 1%
alpha = .01;
%Linkage inductances of primary and secondary
Llink_1 = Primary_turns^2 * Total_permeance*alpha;
Llink_2 = secondary_turns^2 * Total_permeance*alpha;

%resistivity
R_wire_primary = Rwire(Primary_wire_length, Primary_gauge);
```

```
R_wire_secondary = Rwire(Wire_length_secondary, Gauge_secondary);

end

%Magnetizing current
Im = Vrms/(R_wire_primary+1i*Omega*(Llink_1+Lm));

%Magnetizing current magnitude
Im_final = abs(Im);

%Magnetizing flux with weighted areas, since area of core is not
 uniformed,
%center has larger area than branches
Bm = (Lm*Im_final)/
(Primary_turns*((1/7)*(width*width)+(6/7)*(width*(width/2))));

%Core loss density for material
Core_loss_density = Pcore(f,Bm,k_picked);

%Volume of core is volume of full core minus window volumes
Volume = length*height*width - 2*Inner_volume;
Mass_core = GammaC(k_picked) * Volume;

%Wire volumes
Volume_wire = Primary_wire_length * AreaWc(Primary_gauge) +
 Wire_length_secondary * AreaWc(Gauge_secondary);
Mass_wire = GammaW * Volume_wire;

%Total mass
Total_mass = Mass_core + Mass_wire;

%Core losses
Core_losses = Core_loss_density * Volume;

%Resistivity of the core
Rce = (Vrms^2)/Core_losses;
```

Published with MATLAB® R2017b