Оглавление

[Appendix I. MATLAB Code: Rotor Losses from Winding Time and 3](#_Toc319334784)

[Space Harmonics 3](#_Toc319334785)

[% Calculate current THD 3](#_Toc319334786)

[% Calculate current densities 4](#_Toc319334787)

[% Harmonics to be evaluated 4](#_Toc319334788)

[% Surface coefficient at top of retaining sleeve 4](#_Toc319334789)

["%C alculate losses due to time harmonics 4](#_Toc319334790)

["%C alculate losses due to space harmonics 4](#_Toc319334791)

[Appendix J. MATLAB Code: Rotor Losses from Slot Effects 5](#_Toc319334792)

[% Calculate Bd as function of wst and wt (max 10% of Bg) 5](#_Toc319334793)

[% Calculate flux variation parameters 5](#_Toc319334794)

["% Calculate geometry and can loss factor for different rings 5](#_Toc319334795)

[% Calculate can losses 6](#_Toc319334796)

[% Output Results 6](#_Toc319334797)

[% Output Section: 6](#_Toc319334798)

|  |  |  |  |
| --- | --- | --- | --- |
| Процедура | Название в delphi | Входные параметры | Выходные параметры |
| Calculate current THD | Calculate\_current\_THD | Iah  For r  lab | THI~i |
| % Calculate current densities | Calculate\_current\_densities | Iz  Kz  Iz\_1 | KzI |
| % Harmonics to be evaluated | Harmonics\_to\_be\_evaluated | J  n  freq  lam | eta-m  eta-s  alpha-m  k |
| % Surface coefficient at top of retaining sleeve | Surface\_coefficient\_at\_top\_of\_retaining\_sleeve | topI  boti  top2  bot2 | J  alpha-s  Zs |
| "%Calculate losses due to time harmonics | Calculate\_losses\_due\_to\_time\_harmonics | kpn  kbn  kwn;  Kz-s | sys |
| "%C alculate losses due to space harmonics | Calculate\_losses\_due\_to\_space\_harmonics | Wst  Wd  bg | bd |
| % Calculate Bd as function of wst and wt (max 10% of Bg) | Calculate\_Bd\_as\_function | Wst  Pi  Rs  Ns;  lamB | beta |
| % Calculate flux variation parameters | Calculate\_flux\_variation\_parameters | Wst  Pi  Rs  Ns;  Bd  lamB | beta  B |
| "% Calculate geometry and can loss factor for different rings | Calculate\_geometry\_and\_can\_loss\_factor\_for different\_rings | fork  pi  R  Hm  Lst  k;  tanh  ti | A(k)  Ks(k)  t\_Stain |
| % Calculate can losses | Calculate\_can\_losses | piA2  B  Rpm  R  Hm  A2  T  Stain  Stain-res;  Ipm  Titan | w\_Stain  Titan-res;  wInconel  Pý\_Stain  PTitan(k)  P\_-CarFib(k)  PInconel(k  P\_-Magnet |
| % Output Results | Output\_Results | z(i);  P.Stain(k)  PjTitan(k)  PjInconel(k)  dc  Ls  Hmo  hm  g | dco  Lso  hmo  go |

## Appendix I. MATLAB Code: Rotor Losses from Winding Time and

## Space Harmonics

"% Jonathan Rucker, MIT Thesis

"% May 2005

"% Program: pmharmloss

"% Program performs rotor loss calculations caused by

"% winding time and space harmonics for permanent magnet

"% machines with surface magnets and slotted stators.

"% MUST RUN pmlinput, pmlcalc, pmloutput, and get harmonic

"% current data from PSIM prior to running pmharmloss

"% Constants to be used

mu0 = 4\*pi\*le-7; % Free space permeability

tol = le-2; % Tolerance factor

% Retaining sleeve/magnet material resistivity (ohm-m)

Stainres = 0.72e-6;

Titan-res = 0.78e-6;

CarFibres = 9.25e-6;

Inconelres = 0.98e-6;

Magnetcres = 1.43e-6;

% Retaining sleeve thickness set at 0.5mm less than air gap

h\_sl = g - 0.0005; % Sleeve thickness

g\_act = g - hsl; % Actual air gap

% Retaining sleeve conductivities (S/m)

conds = 1/Stain res;

condt = 1/Titan res;

condc = l/CarFib-res;

cond-i = 1/Inconelres;

% Magnet & actual sleeve cond (S/m)

condm = 1/Magnet-res;

condsl = cond-s;

% Input time harmonic peak currents from PSIM

**11 =** 2895;

**13 =** 0;

***15*** = 209;

17 = 89.2;

19 = 0;

Il1 = 39.2;

**113** = 27.6;

115 = 0;

117 = 17.0;

119 = 12.8;

121 **=** 0;

**123** = 7.3;

**125** = 6.3;

**127** = **0;**

167

129 =4.8;

**131 = 3.9;**

**%** Put currents in array

Iharm **= [11 13 15** 17 19 111113 **115** 117 119 121 123 ***125...***

127 129 **13 1];**

# % Calculate current THD

lab **= 0;**

for r **=** 2:length(Iharm)

Iah **=** Iah **+** Iharm(r)A2;

end

THI~i **=** 100\*sqrt(Iah/(Iharm(1)A2));

# % Calculate current densities

Iz **=** (1/sqrt(2)).\*Iharm;

Kz **=** ((q/2)\*(NaI(2\*pi\*Rs))).\*Iz;

Iz\_1 **=** (1/sqrt(2)).\*I1; **%** Fundamental RMS current

KzI **=** ((q/2)\*(Na/(2\*pi\*Rs))).\*Izj1; **%** Fundamental current density

# % Harmonics to be evaluated

n= **1:2:3 1;**

w =n .\* omega; **%** Harmonic angular frequencies

freq **=** w ***I1*** (2\*pi); **%** Harmonic frequencies

lam **=** (2\*(2\*pi/(2\*p)))./n;

**k =** (2\*pi)./lam; **%** Wavenumbers

**%** Eta values

eta-m **=** sqrt((j\*muO\*cond m).\*w **+ (k.A2));**

eta-s **=** sqrt((j\*muo\*cond~sl).\*w **+ (k.A2));**

**%** Surface coefficient at top of magnet layer

alpha-m **=j.** \*(k./eta-m). \*coth(eta-m. \*hm);

# % Surface coefficient at top of retaining sleeve

topI **=** (j.\*(k.fetas).\*sinh(etaS. \*h-sl)) **+** (alpha-m.\*cosh(eta-s.\*h-sl));

boti **=** (j \*(k./eta s).\*cosh(eta-s\*h sl)) **+** (alpha-m.\*sinh(eta-s.\*hLsl));

alpha-s **= j.** \*(k./eta s). \*(topl L/botl1);

**%** Surface coefficient at surface of stator

top2 **=** (j.\*sinh(k.\*g-act)) **+** (alpha-s.\*cosh(k.\*g-act));

bot2 **=** (j \*cosh(k.\*g-act)) **+** (alpha-s.\*sinh(k.\*g-act));

alpha-f **=** j.\*(top2./bot2);

**%** Surface impedance

Zs **=** (muO.\*wi/k).\*alpha-f;

# "%C alculate losses due to time harmonics

**"%**U se only fundamental space harmonic factors

Kz **-t** =kw.\*Kz;

Syt **= 0;**

for **i=** 1:length(n)

Syjt(i) **=** 0.5\*(abs(Kz -t(i))A2)\*real(Zs(1));

Syt **=** Syt **+** Syjt(i);

end

**168**

# "%C alculate losses due to space harmonics

**"%**U se only fundamental time harmonic current

kpn **=** sin(n .~pi/2) **.\*** sin(n **\*~** alfaI2);

kbn **=** sin(n .\*m\*gamal2) ***J1*** (m. \*sin(n A'gama/2));

kwn **=** kpn .\*kbn;

Kz-s **=** kwn .\*Kzl ***I.J*** n;

**Sys =0;**

for i =1:length(n)

Sy-s(i) **=** 0.5\*(abs(Kz-s(i))A2)\*real(Zs(i));

**Sys = Sys +** Sy-s(i);

end

fprintf(AnRotor Losses Caused **by** Harmonics:\n');

fprintfQ'Time Harmonic Losses **= %6.If** kW~n',Syt/lOOO);

fprintf('Space Harmonic Losses **= %6.lf** kW\n',Sys/1000);

fprintf('Total Losses **=%6.lf** kW~n,(Syt-iSys)/1000);

fprintf('Current THD **=%6.2f** %%\n',THDi);

## Appendix J. MATLAB Code: Rotor Losses from Slot Effects

"% Jonathan Rucker, MIT Thesis

"% May 2005

"% Program: pmcanloss

"% Program calculates and outputs rotor losses caused by

"% stator slot effects for permanent magnet machines

"% with surface magnets and slotted stators.

"% MUST RUN pmlinput, pmlcalc, pmwave, and pmcanstress

"% PRIOR TO RUNNING pmcanloss

% Calculate retaining sleeve losses

% Retaining sleeve/magnet material resistivity (ohm-m)

Stainres **=** 0.72e-6;

Titanres **=** 0.78e-6;

CarFibres **=** 9.25e-6;

Inconelres **=** 0.98e-6;

Magnet-res 1.43e-6;

# % Calculate Bd as function of wst and wt (max 10% of Bg)

Bd **=** (wst/wt)\*O.l\*Bg;

# % Calculate flux variation parameters

beta **=** (wst/(2\*pi\*Rs))\*2\*pi;

lamB **=** 2\*pi/Ns;

B **=** (Bd/sqrt(2))\*sqrt(beta/lamB);

# "% Calculate geometry and can loss factor for different rings

"% k is number of rings

fork= 1:10

A(k) **=** pi\*2\*(R+hm)\*Lst/k;

Ks(k) **=** 1 - ((tanh(p\*Lst/(k\*2\*(R+hm))))/(p\*Lst/(k\*2\*(R+hm))));

end

% Input Stainless Steel sleeve thickness based on stress results

for i **=** 1:stop

if SFHoop(i) **<=** Stain str

t\_Stain **=** t(i);

break

elseif t(stop) **>** Stain-str

fprintf('Hoop Stress too high for Stainless Steel.A');

else

dummy **=** t(i);

end

end

% Input Titanium sleeve thickness based on stress results

for i = 1:stop

if SFHoop(i) **<=** Titan str

t\_Titan **=** t(i);

break

elseif t(stop) **>** Titan-str

171

fprintfQ'Hoop Stress too high for Titanium.\n');

else

dummy =ti)

end

end

**%** Input Carbon Fiber sleeve thickness based on stress results

for **i=** 1:stop

if SFHoop(i) **<=** CarFib-str

tCarFib t= )

break

elseif t(stop) **>** CarFib\_str

fprintfQ'Hoop Stress too high for Carbon Fiber.\n');

else

dummy =ti)

end

end

**%** Input Inconel sleeve thickness based on stress results

for **i=** l:stop,

if SFHoop(i) **<=** Inconel-str

tInconel =ti)

break

elseif t(stop) **>** Inconel-str

fprintf('Hoop Stress too high for Inconel.\n');

else

dummy =ti)

end

end

# % Calculate can losses

w\_Stain **=** (piA2/3600)\*((B \*rpm\*(R+hm))A2\*t -Stain)/Stain-res;

w\_Titan **=** (piA2/3600)\*((B \*ipm\*(R+hM))A2\*tTitan)/Titan-res;

wCarFib =(piA2/3600)\*((B \*rpm\*(R+hM))A2\*t -CarFib)/CarFib~res;

wInconel =(piA2/3600)\*((B\*rpm\*(R+hm))A2\*tjlnconel)/Inconel-res;

fork= **1:10**

Pý\_Stain(k) **=** k\*wStain\*Ks(k)\*A(k)/1000;

PTitan(k) **=** k\*wTitan\*Ks(k)\*A(k)/1000;

P\_-CarFib(k) =k\*wCarFib\*Ks(k)\*A(k)/1000;

PInconel(k) =k\*w\_Inconel\*Ks(k)\*A(k)/1000;

End

**"%**C alculate magnet losses (only with carbon steel)

**"%**C alculate geometry and can loss factor

Am **=** pj\*2\*R\*Lst;

Ksm **=** 1 **-** ((tanh(p\*LstI(2\*R)))/(p\*LstI(2\*R)));

**"%**C alculate magnet losses

**"%**A ssumes only **10%** of magnet thickness sees effects

wMagnet =(piA2/3600)\*((B\*rpm\*R)A 2\*0. **1** \*hm)/Magnet-res;

P\_-Magnet =w-Magnet\*Ksm\*Am/lOOO;

# % Output Results

***z=[15 510];***

fprintfQ'Retaining Can Losses:\n');

for i=1:3

**k =** z(i);

fprintf'% 1 .Of Rings:\n',k)

fprintf('Material Thickness Can Loss\htO;

fprintfQ'Stainless Steel ***%5* .2f** mm **%6. if** kW~n',t Stain\* 1000,P.Stain(k));

fprintf('Titanium ***%5.2f*** mm **%6. if** kW~n',t Titan\* 1000,PjTitan(k));

fprintfQ'Carbon Fiber **%5.2f** mm **%6. if** kW\n',t-CarFib\*1000,P-CarFib(k));

fprintf(Q Associated Magnet Loss **%6. If** kAWn',Magnet);

fprintf('lnconel ***%5* .2f** mm **%6. if** kW~n\n',tjlnconel\* 1000,PjInconel(k));

end

dco = dc\* 1000;

Lso =Ls\* 1000;

hmo hm\* 1000;

go = g\*1000;

# % Output Section:

fprintf('\nPM Machine Design, Version 2: Surface Magnet, Slotted Stator\n');

fprintfQ'Machine Size:\n');

fprintfC'Machine Diameter = %8.3f m Machine Length = %8.3f m\n',Dmach,Lmach);

fprintf('Rotor radius = %8.3f m Active length = %8.3f mn~n',R,Lst);

fprintf(QSlot Avg Width =%8.3f mm Slot Height = %8.3f mm\n',wso,hso);

fprintf('Back Iron Thick =%8.3f mm. Tooth Width = %8.3f mm\n',dco,wto);

fprintf('Machine Ratings:\n');

fprintf('Power Rating = %8.if kW Speed = %8.Of RPM\n', Pout,rpm);

fprintf('Va (RMS) = %8.Of V Current = %8.If A\n', VaIa);

fprintf('Ea (RMS) = %8.Of V Arm Resistance = %8.5f ohm\n',Ea,Ra);

fprintf('Synch Reactance = %8.3f ohm Synch Induct =%8.3f mH\n',Xs,Lso);

fprintf('Stator Cur Den = %8.lf A/cm2 Tip Speed = %8.Of mls\n', Jao,vtip);

fprintf('Efficiency = %8.3f Power Factor =%8.3f\n', eff~pf);

fprintf('Phases = %8.Of Frequency = %8. if Hz\n',q,t);

fprintfQ'Stator Paramteters:\n');

fprintfQ'Number of Slots =%8.Of Num Arm Turns = %8.Of \n',Ns,Na);

fprintf('Breadth Factor = %8.3f Pitch Factor =%8.3f W, kb,kp);

fprintf('Tooth Flux Den =%8.2f T Back Iron =%8.2f T\n', Bt,Bb);

fprintf('Slots/pole/phase =%8.2t\n',m);

fprintf('Rotor Parameters:\n');

fprintf('Magnet Height = %8.2f mm Magnet Angle = %8.l1f degm\n',hmo,thm);

fprintf('Air gap = %8.2f mm Pole Pairs = %8.Of \n',go,p);

fprintf('Magnet Remanence = %8.2f T Aig Gap Bg = %8.2f T\n',Br,Bg);

fprintfC'Magnet Factor = %8.3f Skew Factor = %8.3f \n',kg,ks);

fprintf('Machine Losses:\n');

fprintfQ'Core Loss = %8.lf kW Armature Loss = %8.lIf kW\n', Pco,Pao);

fprintfQ'Windage Loss = %8.If kW Rotor Loss = TBD kWV\n', Pwindo);

fprintf('Machine Weights:\n');

fprintfC'Core = %8.2f kg Shaft = %8.2f kg\n',Mc,Ms);

fprintf('Magnet = %8.2f kg Armature = %8.2f kg\n',Mm,Mac);

fprintfQ'Services = %8.2f kg Total = %8.2f kg\n',Mser,Mtot);