Appendix B. MATLAB Code: Basic Sizing Method

Jonathan Rucker, MIT Thesis

"% May 2005

"% Program: pmlbasic

"% Program performs basic sizing and parameter calculations

"% for generators.

"% Definition of variables

"% Name Variable

"% General variables

"% Pwr Required power

"% rpm Speed (RPM)

"% psi Power factor angle

"% f Electrical frequency (Hz)

"% omega Electrical frequency (rad/sec)

"% vtip Tip speed (m/s)

"% LovD L/D ratio

"% stress Gap shear stress (psi)

"% Rotor variables

% R Rotor radius (m)

% D Rotor diameter (m)

% Lst Rotor stack length (m)

% p Number of pole pairs

"% Bg Expected air gap flux density (T)

"% Stator variables

"% Kz Surface current density (A/m)

"% Jz Current density (A/m2)

"% hs Slot height (m)

clear;

% Constants & conversion factors

hs = .015; % Assume slot depth of 15 mm

lams = *0.5;* % Assume slot fill fraction

convI = 9.81; % 9.81 W per Nm/s

conv2 = 703.0696; % 703.0696 N/m2 per psi

% INPUTS

Pwr = 16e6; % Required power

vtip = 200; % Max tip speed (m/s)

LovD = 2.85 1; % Wound rotor usually 0.5-1.0, PM 1.0-3.0

"% Shear stress usually 1-10 psi small machines, 10-20 large liquid

"% liquid cooled machines

stress = 15;

**p =** 3; % Pole pairs

Bg **=** 0.8; % Tesla

"% Calculations

"% Size

139

**%** Initially use Pwr **=** 2\*pi\*R\*Lst\*stress\*vtip

**%** Lst =2\*LovD\*R

hscm =hs\* 100;

R **=** sqrt(Pwr/(2\*pi\*(LovD\*2)\*vtip\*stress\*convl \*conv2));

**D =** 2\*R;

Lst **=** LovD\*D;

**%** Speed

omega **=** (p\*vtip)/R;

***f* =** omega/(2"'pi);

rpm **= (60\*f)/p;**

**%** Current densities

Kz **=** (stress\*conv2)/(Bg\* 100);

Ja **=** 1O\*KzJ(hscm\*lams);

**%** Output

fprintf('Basic Machine Design\n');

fprintf('lnput Parameters:\n');

fprintfQ'Power **= %1O.lf** kW Shear Stress **=%10.lf** psi\n',Pwr/1e3,stress);

fprintf('L/D Ratio **= %10.2f** Tip Speed **= %lO.lf** m/s\n',LovD,vtip);

fprintf('Pole Pairs **= %1O.lf** Air Gap Bg = %10.lfl~hi,p,Bg);

fprintf('Output:\n');

fprintfQ'Rotor Radius **= %10.3f** m Stack Length **= %10.3f** m\n',R,Lst);

fprintfC'Speed = %10.0fRPM Frequency = %10.lfHz\n',rpm,f);

fprintfQ'Ja **=%10.2f** A/cm2\n',Ja);

140

Appendix C. PM Machine Database

Parameter Specific Machines Motor or Generator -.specif.yw/M or **G)**

4 **5 6 7 8 9** 10 11 12

Type (M-Motor, G-Generator) G G M M M M M G G G G G

Power (kW) 36500 2030 20000 5.6 36500 5000 20000 2.562 1500 100 2.5 400

Power factor 0.854

Rotational Speed (rpm) 3600 19000 150 127 150 180 1800 20000 159.1 1500 36

Rotor radius (cm) 11 42.5 6.2 4.5 15 3.56 4.37 105

ir gap (mm) 1 2 1.6

Marnet thickness (mm) **3.3** 22 4.5 25.4

Active length (m) 0.124 0.225 0.1 1.05

Polepairs 2 6 2 2 4 **8** 2 83

Number of phases 3 15 3 3 3 **3** 3 3

Number of Slots 36 24 12 36 180

Slots per pole per phase 3 2 0.5 3

Slots short pitched 2 1 2

Slot depth (mmm) 22.5 40 21

Slot depression depth (mm) 0 0 0

Slot depression width (mm) 5.99 0 17

Peripheral Tooth Fraction 0.5 0.5 0.5

Back iron thickness (mm) 14.2 28

Turns per coil 30 2 2 6 9

Coils per phase 16 4

Slot fill fraction 0.4

Magnet remant flux density (T) 1.21 1.16 1.18 1.1 1.128 0.925

Magnet electrical angle (deg e\_) 143.1 80 144

Base frequency (Hz) 60 60 60 50 133.3 50 50

Mass (MT) 14 112.5 118.8 71 39 5 0.2205

Volume (cubic m) 4.75 2.08 48 155.05 81.65 21.93 0.02

idth (m) 1.2 1.4 3.25 0.229 5.5 5.4 2.8 0.44 0.463 **0,155**

Length (m) 2.64 1.35 4.87 0.102 5.3 2.8 2.9 **1 1 1** 0.222 **1 1**

Height (m) 1.5 1.1 3.35 0.229 5.3 5.4 2.8 0.44 0.463 0.155

Flux Layout Radial Radial Radial Radial Radial Radial Radial Radial Radial Radial Radial Radial

Voltagae 800 VDC 460 VAC 5000 **VAq** 42 **VDC** 800 **VDC. 110** VAC 2200 VAC

Parameter Specific Machines (Motor or Generator - specify w/M or G)

13 14 15 16 17 18 19 20 21 22 23 24

Type (M-Motor, G-Generator) M G **G** G G M M M G G G G

Power (kW) 10 200 200 250 250 3800 5000 4300 1400 25 110

Power factor

Rotational Speed (rpm) 1265 3600 3600 1050 10500 120 120 150 90000 18000 60000 70000

Rotor radius (cm) 5 2.16 3.3

Air gap (mm) 1.1 3 3.5 3 9 0.51

Magnet thickness (mm) 5 3.5 6 3.5 5.84

Active length (m) 0.12 0.125 0.097 0.16

Pole pairs 3 12 12 14 3 100 2 2 3 2

Number of phases 3 3 3 5 3 3 3 3 3

Number of Slots 36 40 36 192 12 36 9 24

Slots per pole per phase 2 0.286 2 2 3 4

Slots short pitched 1 **1** 0 0

Slot depth (mm)

Slot depression depth (mm)

Slot depression width (mm)

Peripheral Tooth Fraction

Back iron thickness (mm) 12.5

Turns per coil 2

Coils per phase

Slot fill fraction

Ma net remant flux density (T) 1.17 1.08

Magnet electrical angle (deg e) 84 180 120

Base frequency (Hz) 60 720 720 245 525 3000 600

Mass (MT) 0.2833 0.0871 52 67 65 **1**

Volume (cubic m)

idth (m) 2.325 2.67 3.7 0.107 0.135

Length (m) 3.47 3.41 3.8

Heliht (m) \_2.325 2.67 3.7 0.107

Flux Layout Radial Radial Radial Radial Radial Radial Radial Radial Radial Radial Radial Radial

Voltage 18 VAC **1** 730 VAC 600 **VDC** 600 **VDC** 705 **VDC 1** 500 **VDC**

141

Page Intentionally Left Blank

142

Appendix D. MATLAB Code: Sizing Method 1

"% Jonathan Rucker, MIT Thesis

"% May 2005

"% Program: pmlinput

"% Program used as input file for pmlcalc

"% All necessary input parameters entered here.

clear;

"% Definition & Entry of variables

"% General variables

Pwr = 16e6; % Required power (W)

rpm = 13000; % Speed (RPM)

psi = 0; % Power factor angle

% Rotor variables

R = 0.147; % Rotor radius (m)

hm = 0.025; % Magnet thickness (m)

Lst = 0.838; % Rotor stack length (m)

p = 3; % Number of pole pairs

Br = 1.2; % Magnet remnant flux density (T)

thm = *50;* % Magnet physical angle (deg)

thsk = 10; % Magnet skew angle (actual deg)

% Stator variables

q = 3; % Number of phases

Ns = 36; % Number of slots

Nsp = 1; % Number of slots short pitched

g = .004; % Air gap (m)

tfrac = 0.5; % Peripheral tooth fraction

hs = .025; % Slot depth (m)

hd = .0005; % Slot depression depth (m)

wd = le-6; % Slot depression width (m)

syrat = 0.7; % Stator back iron ratio (yoke thick/rotor radius)

Nc = 1; % Turns per coil

lams = 0.5; % Slot fill fraction

sigst = 6.0e+7; % Stator winding conductivity

% Densities

rhos = 7700; % Steel density (kg/m3)

rhom = 7400; % Magnet density (kg/m3)

rhoc = 8900; % Conductor density (kg/m3)

"% Jonathan Rucker, MIT Thesis

"% May 2005

"% Program: pmlcalc

"% Program performs sizing and parameter calculations

"% for permanent magnet machines with surface magnets and

"% slotted stators.

"% Program developed from J.L. Kirtley script with permission

"% MUST RUN pmlinput PRIOR TO RUNNING pmlcalc

143

% Definition of variables

% Name Variable

"% General variables

"% Pwr Required power (W)

"% rpm Speed (RPM)

"% psi Power factor angle

"% f Electrical frequency (Hz)

"% omega Electrical frequency (rad/sec)

"% vtip Tip speed (m/s)

"% lambda Flux linkage

% Ea RMS Internal voltage (V)

% Rotor variables

% R Rotor radius **(in)**

% hm Magnet thickness **(in)**

% Lst Rotor stack length **(in)**

% p Number of pole pairs

"% Br Magnet remnant flux density (T)

"% thin Magnet physical angle (deg)

"% thsk Magnet skew angle (actual deg)

"% Stator variables

% q Number of phases

% **m** Slots per pole per phase

"% Ns Number of slots

"% Nsp Number of slots short pitched

% g Air gap **(in)**

% ge Effective air gap **(in)**

"% tfrac Peripheral tooth fraction

"% hs Slot depth **(in)**

"% hd Slot depression depth **(in)**

"% wd Slot depression width **(in)**

"% syrat Stator back iron ratio (yoke thick/rotor radius)

% Nc Turns per coil

"% lams Slot fill fraction

"% sigst Stator conductivity

% Kc Carter coefficient

"% Loss Models

"% P0 Base power for core losses

"% FO Base frequency for core loss

"% BO Base flux density

"% epsb Flux density exponent

"% epsf Frequency exponent

"% rhos Steel density

"% rhom Magnet density

% rhoc Conductor density

% Constants to be used

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

tol = **I** e-2; % Tolerance factor

cpair = 1005.7; % Specific heat capacity of air (J/kg\*C)

rhoair = 1.205; % Density of air at 20 C (kg/m3)

nuair = 1.5e-5; % Kinematic viscosity of air at 20 C (m2/s)

P0 = 36.79; % Base Power Losss, W/lb

FO = 1000; % Base freuency, 60 Hz

B0 = 1.0; % Base flux density, 1.0 T

144

epsb = 2.12;

epsf= 1.68;

"% Generate geometry of machine

"% Number of slots/pole/phase

m = Ns/(2\*p\*q);

% Number of armature turns (each slot has 2 half coils)

Na = 2\*p\*m\*Nc;

% Tooth width

wt = 2\*pi\*(R+g+hm+hd)\*tfrac/Ns;

% Slot top width (at air gap)

wst = 2\*pi\*(R+g+hm+hd)\*(1-tfrac)/Ns;

% Slot bottom width

wsb **=** wst\*(R+g+hd+hs)/(R+g+hm+hd);

% Stator core back iron depth (as p increases, dc decreases)

dc = syrat\*R/p;

% Full-pitch coil throw

Nsfp = floor(Ns/(2\*p));

% Actual coil throw

Nsct **=** Nsfp - Nsp;

"% Estimate end turn length

"% End turn travel (one end)

laz = pi\*(R+g+hm+hd+0.5\*hs)\*Nsct/Ns;

% End length (half coil)

le2 = pi\*laz;

% End length (axial direction)

lel = 2\*le2/(2\*pi);

% Calculate electrical frequency & surface speed

**f** = p\*rpm/60;

omega = 2\*pi\*f;

vtip = R\*omega/p;

% Winding & skew factors

gama = 2\*pi\*p/Ns;

alfa = pi\*Nsct/Nsfp;

kp = sin(pi/2)\*sin(alfa/2);

kb = sin(m\*gama/2)/(m\*sin(gama/2));

kw **=** kp\*kb;

ths = ((p\*thsk)+le-6)\*(pi/180); % skew angle (elec rad)

ks = sin(ths/2)/(ths/2);

% Calculate magnetic gap factor

Rs = R+hm+g;

Ri **=** R;

R1 **=** R;

R2 **=** R+hm;

kg = ((RiA(p-1))/(Rs^(2\*p)-RiA(2\*p)))\*((p/(p+l))\*(R2A(p+l)-R1A(p+l))...

+(p\*RsA(2\*p)/(p-1))\*(R1 A(1-p)-R2A(1-p)));

"% Calculate air gap magnetic flux density

"% Account for slots, reluctance, and leakage

ws = (wst+wsb)/2; % Average slot width

taus = ws + wt; % Width of slot and tooth

145

Kc = 1/(1-(1/((taus/ws)\*((5\*g/ws)+l))));

ge = Kc\*g;

Cphi = (p\*thm)/180; % Flux concentration factor

KI = 0.95; % Leakage factor

Kr = 1.05; % Reluctance factor

murec = *1.05;* % Recoil permeability

PC = hm/(ge\*Cphi); % Permeance coefficient

Bg = ((Kl\*Cphi)/(1+(Kr\*murec/PC)))\*Br;

% Calculate magnetic flux and internal voltage

thmrad = thm\*(pi/180);

B **I** = (4/pi)\*Bg\*kg\*sin(p\*thmrad/2);

lambda = 2\*Rs\*Lst\*Na\*kw\*ks\*B l/p;

Ea = omega\*lambda/sqrt(2); % RMS back voltage

"% Calculation of inductances/reactances

"% Air-gap inductance

Lag = (q/2)\*(4/pi)\*(muo\*Na^2\*kwA2\*Lst\*Rs)/(pA2\*(g+hm));

% Slot leakage inductance

perm = mu0\*((1/3)\*(hs/wst) + hd/wst);

Las = 2\*p\*Lst\*perm\*(4\*NcA2\*(m-Nsp)+2\*Nsp\*NcA2);

Lam = 2\*p\*Lst\*Nsp\*NcA2\*perm;

if q == 3

Lslot = Las + 2\*Lam\*cos(2\*pi/q); % 3 phase equation

else

Lslot = Las - 2\*Lam\*cos(2\*pi/q); % multiple phases

end

% End-turn inductance (Hanselman)

As = ws\*hs; % Slot area

Le = ((Nc\*muO\*(taus)\*Na^2)/2)\*log(wt\*sqrt(pi)/sqrt(2\*As));

% Total inductance and reactance

Ls = Lag+Lslot+Le;

Xs = omega\*Ls;

% Lengths, Volumes, and Weights

% Armature conductor length

Lac = 2\*Na\*(Lst+2\*le2);

% Armature conductor area (assumes form wound)

Aac = As\*lams/(2\*Nc);

% Mass of armature conductor

Mac = q\*Lac\*Aac\*rhoc;

% Overall machine length

Lmach = Lst+2\*lel;

% Core inside radius

Rci = R+hm+g+hd+hs;

% Core outside radius

Rco = Rci+dc;

% Overall diameter

Dmach = 2\*Rco;

% Core mass

Mcb = rhos\*pi\*(RcoA2-RciA2)\*Lst; % Back iron

Mct = rhos\*Lst\*(Ns\*wt\*hs+2\*pi\*R\*hd-Ns\*hd\*wd); % Teeth

Mc = Mcb + Mct;

% Magnet mass

146

Mm **=** O.5\*(p\*thmriad)\*((R+hM)A2-RA2)\*Lst\*rhom;

**%** Shaft mass

Ms **=** pi\*R A2\*Lst\*rhos;

***%* 15%** service fraction

Mser **= *0.* 15** \*(Mc+Ms+Mm+Mac);

**%** Total mass

Mtot **=** Mser+Mc+Ms+Mm+Mac;

**%** Stator resistance

Ra **=** Lac/(sigst\*Aac);

**"%**C ore Loss Calculations

**"%**T ooth Flux Density

Bt **=** Bg/tfrac;

**%** Back iron flux density (Hanselman)

**Bb =** Bg\*RI(p\*dc);

**%** Core back iron loss

Pcb **=** Mcb\*PO\*abs(BbIBO)A epsb\*abs(fIFO)A epsf;

**%** Teeth Loss

Pct **=** Mct\*PO\*abs(BtIBO)A epsb\*abs(ffFO)A epsf;

**%** Total core loss

Pc **=** Pcb **+** Pct;

**%** Start loop to determine terminal voltage and current

notdone **= 1;**

1 **=0;**

la **=** Pwr/(q\*Ea);

while notdone **==I1**

i **= i+1;**

xa **=** Xs\*IaIEa;

**%** Conductor losses

Pa **=** q\*IaA2\*Ra;

**"%**G ap friction losses

**"%**R eynold's number in air gap

omegam **=** omega/p;

Rey **=** omegam\*R\*g/nuair;

**%** Friction coefficient

**Cf** = .0725/ReyA.2;

**%** Windage losses

Pwind **=** Cf\*pi\*rhoair\*omegamA3\*RA4\*Lst;

**%** Get terminal voltage

Va **=** sqrt(EaA2-((Xs+Ra)\*Ia\*COS(pSi))A2)-(Xs+Ra)\*Ia\*sin(psi);

Ptemp, **=** q\*Va\*Ia\*cos(psi)..Pwind;

error **=** PwrlPtemp;

err(i) **=** error;

if abs(eff or- **1) <** tol

notdone **= 0;**

else

Ia **=** Ia\*error;

end

end

147

**"%**R emaining performance parameters

**"%**C urrent density

Ja **=** IaIAac;

**%** Power and efficiency

Pin =Pwr+Pc+Pa+Pwind;

eff =Pwr/Pin;

**pf =** cos(psi);

fprintff'pmlcalc complete: Ready\n');

**"%**Jo nathan Rucker, MIT Thesis

**"%**M ay **2005**

**%** Program: pmloutput

**"%**Pr ogram outputs values from pmlcalc.

**"%**P rogram developed from **J.L.** Kirtley script with permission

**"%M UST RUN** pmlinput and pmlcalc PRIOR TO **RUNNING** pmloutput

**"%**V ariables for output display

Pout **=** Pwr/le3;

Jao =Ja/le4;

Pco =Pc/le3;

Pwindo **=** Pwind/1e3;

Pao **=** PaIl e3;

**wso = ws\* 1000;**

hso **=** hs\* 1000;

wto **=** wt\* **1000;**

dco = dc\* 1000;

Lso =Ls\* **1000;**

hmo =hm\* 1000;

**go = g\*1000;**

**%** Output Section:

fprintfC'\nPM Machine Design, Version **1:** Surface Magnet, Slotted Stator\n');

fprintf('Machine Size:\n');

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

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

fprintfQ'Slot Avg Width **=%8.3f** mm Slot Height **= %8.3f** mm\n',wso,hso);

fprintfQ'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);

fprintfQ'Va (RMS) **= %8.Of** V Current = %8.lfA\n', Va,Ia);

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

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

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

fprintfQ'Efficiency = **%8.3f** Power Factor =%8.3f\n', eff~pf;

fprintf('Phases **= %8.Of** Frequency = **%8.lf** Hz\n',q,f);

fprintfC'Stator Parameters:\n');

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

148

fprintfQ'Breadth Factor **= %8.3f** Pitch Factor **=%8.3f** nW, **kb,kp);**

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

fprintf('Slotsfpole/phase =%8.21\n',m);

fprintf(QRotor Parameters:\n');

fprintf(QMagnet Height **= %8.2f** mm Magnet Angle **= %8.lf** 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** TWn,Br,Bg);

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

fprintf('Machine Losses:\n');

fprintf('Core Loss **= %8.lf** kW Armature Loss **= %8.If** kW~n', Pco,Pao);

fprintfQ'Windage Loss = **%8.lf** kW Rotor Loss = TBD **kWfn',** Pwindo);

fprintf('Machine Weights:\n');

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

fprintfQ'Magnet = **%8.2f kg** Armature = **%8.2f** kg\n',Mm,Mac);

fprintf('Services = **%8.2f kg** Total = **%8.2f** kg\n',Mser,Mtot);

149

Page Intentionally Left Blank

150

Appendix E. MATLAB Code: Sizing Method 2

"% Jonathan Rucker, MIT Thesis

"% May 2005

"% Program: pm2input

"% Program used as input file for pm2calc

"% All necessary input parameters entered here.

clear;

% Definition & Entry of variables

% General variables

Pwr = 16e6; % Required power (W)

rpm = 13000; % Speed (RPM)

psi = 0; % Power factor angle

Bsat = 1.65; % Stator saturation flux density

% Rotor variables

vtip = 200; % Tip speed limit (m/s)

p = 3; % Number of pole pairs

Br = 1.2; % Magnet remnant flux density (T)

thsk = 10; % Magnet skew angle (elec deg)

PC = 5.74; % Permeance coefficient for magnets

% Stator variables

Ja = 2200; % Initial current density (A/cm2)

q = 3; % Number of phases

m = 2; % Slots/pole/phase

Nsp = 1; % Number of slots short pitched

g = .004; % Air gap **(in)**

hs = .025; % Slot depth **(in)**

hd = .0005; % Slot depression depth **(in)**

wd = le-6; % Slot depression width **(in)**

ws = .016; % Avg slot width **(in)**

Nc = 1; % Turns per coil

lams = *0.5;* % Slot fill fraction

sigst = 6.0e+7; % Stator winding conductivity

% Densities

rhos = 7700; % Steel density (kg/m3)

rhom = 7400; % Magnet density (kg/m3)

rhoc = 8900; % Conductor density (kg/m3)

"% Jonathan Rucker, MIT Thesis

"% May 2005

"% Program: pm2calc

"% Program performs sizing and parameter calculations

"% for permanent magnet machines with surface magnets and

"% slotted stators.

"% MUST RUN pm2input PRIOR TO RUNNING pm2calc

% Definition of variables

% Name Variable

151

% General variables

% Pwr Required power (W)

% rpm Speed (RPM)

% psi Power factor angle

% f Electrical frequency (Hz)

% omega Electrical frequency (rad/sec)

% vtip Tip speed (m/s)

% lambda Flux linkage

% Ea RMS Internal voltage (V)

% Rotor variables

% R Rotor radius (m)

% hm Magnet thickness (m)

% Lst Rotor stack length (m)

% p Number of pole pairs

% Br Magnet remnant flux density (T)

% thm Magnet physical angle (deg)

% thsk Magnet skew angle (actual deg)

% Stator variables

% q Number of phases

% m Slots per pole per phase

% Ns Number of slots

% Nsp Number of slots short pitched

% g Air gap (m)

% ge Effective air gap (m)

% tfrac Peripheral tooth fraction

% hs Slot depth (m)

% hd Slot depression depth (m)

% wd Slot depression width (m)

% syrat Stator back iron ratio (yoke thick/rotor radius)

"% Nc Turns per coil

"% lams Slot fill fraction

"% sigst Stator conductivity

% Kc Carter coefficient

"% Loss Models

"% P0 Base power for core losses

"% FO Base frequency for core loss

"% BO Base flux density

"% epsb Flux density exponent

"% epsf Frequency exponent

"% rhos Steel density

"% rhom Magnet density

"% rhoc Conductor density

% Constants to be used

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

tol = le-2; % Tolerance factor

cpair = 1005.7; % Specific heat capacity of air (J/kg\*C)

rhoair = 1.205; % Density of air at 20 C (kg/m3)

nuair = 1.5e-5; % Kinematic viscosity of air at 20 C (m2/s)

P0 = 36.79; % Base Power Losss, W/lb

FO = 1000; % Base freuency, 60 Hz

BO = 1.0; % Base flux density, 1.0 T

epsb = 2.12;

epsf= 1.68;

152

% Calculate electrical frequency & rotor radius

**f =** p\*rprn/60;

omega **=** 2\*pi\*f;

R **=** p\*vtip/omega;

% Winding & skew factors

Ns **=** floor(2\*q\*p\*m); % Number of slots

gama **=** 2\*pi\*p/Ns;

Nsfp **=** floor(Ns/(2\*p));

Nsct **=** Nsfp - Nsp;

alfa **=** pi\*Nsct/Nsfp;

kp **=** sin(pi/2)\*sin(alfa/2);

kb **=** sin(m\*gama/2)/(m\*sin(gama/2));

kw **=** kp\*kb;

ths **=** ((p\*thsk)+le-6)\*(pi/180); % skew angle (elec rad)

ks **=** sin(ths/2)/(ths/2);

% Calculate magnet dimensions, tooth width, & air gap flux density

thme **=** 1; % Initial Magnet angle (deg e)

notdone = 1;

ge **=** g; % Initial effective air gap

while notdone **==** 1

alpham **=** thme/1 80; % Pitch coverage coefficient

Cphi **=** (2\*alpham)/(l+alpham); % Flux concentration factor

hm **=** ge\*Cphi\*PC; % Magnet height

Ds **=** 2\*(R+hm+g); % Inner stator/air gap diameter

K1 **=** 0.95; % Leakage factor

Kr **=** *1.05;* % Reluctance factor

murec **=** 1.05; % Recoil permeability

Bg **=** ((Kl\*Cphi)/(1+(Kr\*murec/PC)))\*Br;

wt **=** ((pi\*Ds)/Ns)\*(Bg/Bsat); % Tooth width

taus **=** ws + wt; % Width of slot and tooth

Kc **=** 1/(1-(1/((taus/ws)\*((5\*g/ws)+1)))); % Carter's coefficient

ge **=** Kc\*g;

eratio **=** ws/wt;

if abs(eratio - 1) < tol

notdone = 0;

else

thme **=** thme + 1;

end

end

% Set final values

thm **=** thme/p; % Magnet physical angle

thmrad **=** thm\*(pi/1 80);

hm **=** ge\*Cphi\*PC; % Magnet height

Ds **=** 2\*(R+hm+g); % Inner stator/air gap diameter

"% Generate geometry of machine

"% Peripheral tooth fraction

tfrac **=** wt/(wt+ws);

% Slot top width (at air gap)

wst **=** 2\*pi\*(R+g+hm+hd)\*tfrac/Ns;

% Slot bottom width

wsb **=** wst\*(R+g+hm+hd+hs)/(R+g+hm+hd);

153

**%** Stator core back iron depth

dc **=** (pi\*Ds\*thmradl(4\*p))\*(BglB sat);

**%** Core inside radius

Rci **=** R+hm+g+hd+hs;

**%** Core outside radius

Rco **=** Rci+dc;

**%** Slot area

As **=** ws\*hs;

**"%**E stimate end turn length

**"%**E nd turn travel (one end)

laz **=** pi\*(R+g+hm+hd+0.5\*hs)\*NsctlNs;

**%** End length (half coil)

1e2 **=** pi\*laz;

**%** End length (axial direction)

lel **=** 2\*1e2/(2\*pi);

**%** Calculate magnetic gap factor

Rs **=** R+hm+g;

Ri **=** R

RI **=** R;

R2 =R+hm;

**kg =** ((RiA(p~1 ))/(RSA (2\*p)-RiA **(2\*p)))\*((p/(p+l1))\*(R2A(p+ 1)-Ri A(p+ 1))...**

***+(p\*RSA (2\*p)I(p-* 1))\*(R JA( I -p)-R2A( 1p)));**

**%** Core loss calculations (per length)

**%** Core mass per length

Mcbperl = rhos\*pi\*(RcOA 2-RciA 2); **%** Back iron

MctperL =rhos \*(Ns\*wt\*hs+2\*pi\*R\*hd-Ns\*hd\*wd); **%** Teeth

Mcperl = McbperL **+** MctperL;

**%** Tooth Flux Density

Bt **=** Bg/tfrac;

**%** Back iron flux density (Hanselman)

**Bb =** Bg\*R/(p\*dc);

**%** Core back iron loss per length

PcbperL **=** McbperL\*PO\*abs(Bb/BO0)A epsb\*abs(f/FO)A epsf,

**%** Teeth Loss per length

PctperL **=** MctperL\*PO\*abs(Bt!BO)A epsb\*abs(f/FO)A epsf;

**%** Total core loss per length

PcperL **=** PcbperL **+** PctperL;

**%** Current and surface current density

**%** Armature turns (each slot has 2 half coils)

Na **=** 2\*p\*m\*Nc;

**%** Arm cond area (assumes form wound)

Aac **=** (As\*lams)/(2\*Nc);

**%** Power **&** Current waveform factors (Lipo)

ke **0.52;**

ki =sqrt(2);

**%** Initial terminal current

la **=** Ns\*lams\*As\*Ja\* 1 e4/(2\*q\*Na);

notfin **= 1;**

Lst **=0. 1; %** Initial stack length

= 1

*154*

% Start loop to determine Lst, Ea, Va, and Ia

notdone = 1;

**k =** 0;

while notdone == 1

k=k+ **1;**

% Surface current density

A = 2\*q\*Na\*Ia/(pi\*Ds);

% Calculate stack length of machine

% Loop to get stack length

while notfin **==** 1

% Gap power

Pgap = 4\*pi\*ke\*ki\*kw\*ks\*kg\*sin(thmrad)\*(f/p)\*A\*Bg\*(Ds^2)\*Lst;

% Length of conductor

Lac = 2\*Na\*(Lst+2\*le2);

% Stator resistance

Ra = Lac/(sigst\*Aac);

% Copper Loss

Pa = q\*IaA2\*Ra;

% Core losses

Pc = PcperL\*Lst;

% Iterate to get length

Ptempl = Pgap-Pa-Pc;

error = Pwr/Ptemp 1;

err(i) = error;

if abs(error-1) < tol

notfin = 0;

else

Lst = Lst\*error;

i=i+ **1;**

end

end

% Calculate magnetic flux and internal voltage

thmrad = thm\*(pi/180);

B 1 = (4/pi)\*Bg\*kg\*sin(p\*thmrad/2);

lambda = 2\*Rs\*Lst\*Na\*kw\*ks\*B **I/p;**

Ea = omega\*lambda/sqrt(2); % RMS back voltage

"% Calculation of inductances/reactances

"% Air-gap inductance

Lag = (q/2)\*(4/pi)\*(muO\*NaA2\*kwA2\*Lst\*Rs)/(pA2\*(g+hm));

% Slot leakage inductance

perm = muO\*((1/3)\*(hs/wst) + hd/wst);

Las = 2\*p\*Lst\*perm\*(4\*NcA2\*(m-Nsp)+2\*Nsp\*NcA2);

Lam = 2\*p\*Lst\*Nsp\*NcA2\*perm;

if q == 3

Lslot = Las + 2\*Lam\*cos(2\*pi/q); % 3 phase equation

else

Lslot = Las - 2\*Lam\*cos(2\*pi/q); % multiple phases

end

% End-turn inductance (Hanselman)

taus = ws + wt; % Width of slot and tooth

Le = ((Nc\*muO\*(taus)\*NaA2)/2)\*log(wt\*sqrt(pi)/sqrt(2\*As));

*155*

% Total inductance and reactance

Ls = Lag+Lslot+Le;

Xs = omega\*Ls;

"% Lengths, Volumes, and Weights

"% Armature conductor length

Lac = 2\*Na\*(Lst+2\*le2);

% Mass of armature conductor

Mac = q\*Lac\*Aac\*rhoc;

% Overall machine length

Lmach = Lst+2\*lel;

% Overall diameter

Dmach = 2\*Rco;

% Core mass

Mc = McperL\*Lst;

% Magnet mass

Mm = 0.5\*(p\*thmrad)\*((R+hm)A2-RA2)\*Lst\*rhom;

% Shaft mass

Ms = pi\*RA2\*Lst\*rhos;

% 15% service fraction

Mser = 0.15\*(Mc+Ms+Mm+Mac);

% Total mass

Mtot = Mser+Mc+Ms+Mm+Mac;

"% Gap friction losses

"% Reynold's number in air gap

omegam = omega/p;

Rey = omegam\*R\*g/nuair;

% Friction coefficient

Cf = .0725/ReyA.2;

% Windage losses

Pwind = Cf\*pi\*rhoair\*omegamA3\*RA4\*Lst;

% Get terminal voltage

xa = Xs\*Ia/Ea;

Va = sqrt(EaA2-((Xs+Ra)\*Ia\*cos(psi))A2)-(Xs+Ra)\*Ia\*sin(psi);

Ptemp = q\*Va\*Ia\*cos(psi)-Pwind;

Perror = Pwr/Ptemp;

Perr(k) = Perror;

if abs(Perror-1) < tol

notdone = 0;

else

Ia = Ia\*Perror;

end

end

% Remaining performance parameters

% Current density

Ja = Ia/Aac;

% Power and efficiency

Pin = Pwr+Pc+Pa+Pwind;

eff = Pwr/Pin;

pf = cos(psi);

156

fprintf('pm2calc complete: Ready.\n');

**"%**Jo nathan Rucker, MIT Thesis

**"%**M ay **2005**

**"%**P rogram: pm2output

**"%**P rogram outputs values from pm2calc.

**"%**Pr ogram developed from **J.L.** Kirtley script with permission

**"%M UST RUN** pm2input and pm2calc PRIOR TO **RUNNING** pm2output

**"%**V ariables for output display

Pout =Pwr/1e3;

Jao =Ja/l e4;

Pco =Pc/1e3;

Pwindo **=** Pwind/1e3;

Pao =Pa/le3;

**wso = ws\*'1000;**

hso **=** hs\* 1000;

wto **=** wt\* 1000;

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);

**157**

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);

**158**

Appendix F. MATLAB Code: Bode Plot

"% Jonathan Rucker, MIT Thesis

"% May 2005

"% Program: Buckfilter

"% Program calculates transfer function and outputs

"% Bode plot for buck converter input filter

clear;

% Input parameters

R = 4.79e-3;

Cf= 2.84e-3;

**Cb** = 28.4e-3;

Lf= 1.415e-6;

% Set up transfer function

num = [R\*Cb 1];

den = [Lf\*R\*Cf\*Cb Lf\*(Cf+Cb) R\*Cb 1];

H = tf(num,den);

bode(H)

159

Page Intentionally Left Blank

Appendix **G.** MATLAB Code: PM Generator Waveforms

% Jonathan Rucker, MIT Thesis

% May 2005

"% Program: pmwave

"% Program calculates and outputs different waveforms,

"% calculates THD, and computes the harmonic content

"% for permanent magnet machines with surface magnets and

% slotted stators.

% MUST RUN pmlinput and pmlcalc PRIOR TO RUNNING pmwave

Definition of variables

"% Name Variable

"% General variables

"% Pwr Required power (W)

"% rpm Speed (RPM)

"% psi Power factor angle

"% f Electrical frequency (Hz)

"% omega Electrical frequency (rad/sec)

"% vtip Tip speed (m/s)

"% lambda Flux linkage

"% Ea RMS Internal voltage (V)

"% Rotor variables

% R Rotor radius **(in)**

"% hm Magnet thickness **(in)**

"% Lst Rotor stack length **(in)**

% p Number of pole pairs

"% Br Magnet remnant flux density (T)

"% thin Magnet physical angle (deg)

"% thsk Magnet skew angle (actual deg)

"% Stator variables

% q Number of phases

**%in** Slots per pole per phase

% Ns Number of slots

% Nsp Number of slots short pitched

%g Air gap (m)

"% ge Effective air gap **(in)**

"% tfrac Peripheral tooth fraction

"% hs Slot depth **(in)**

"% hd Slot depression depth **(in)**

"% wd Slot depression width **(in)**

"% syrat Stator back iron ratio (yoke thick/rotor radius)

"% Nc Turns per coil

"% lams Slot fill fraction

"% sigst Stator conductivity

"% Kc Carter coefficient

% Constants to be used

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

tol = le-2; % Tolerance factor

161

**%** Harmonics to be evaluated

n **= 1:2:35;**

np, **= p .\*** n; **%** Use in kgn equation

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

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

**%** Harmonic winding and skew factors

gama **=** 2\*pi\*p/Ns;

alfa **=** pi\*NsctlNsfp;

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

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

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

ths =((p\*thsk)+1e-6)\*(piI18O); **%** skew angle (elec rad)

ksn =sin(n A' ths/2) *./* (n A' ths/2);

**%** Calculate magnetic gap factor

Rs =R+hm+g;

Ri =R;

RI =R;

R2 =R+hm;

kgn **=** ((Ri.A(np 1 ))./(Rs.A (2. \*np)..Ri.A (2.\*np))).\*((np./(np+ **1))...**

.\*(R2 A(np+ 1)-Ri.A(np+ 1))+(np.\*Rs.A(2.\*np)./(np- **1))...**

**%** Calculate magnetic flux and internal voltage

thmrad **=** thm\*(pi/I 80);

thmerad **=** P\*thmjrad;

Bn **=** Bg.\*((4/pi)./n). \*kgn.\*sin(n. \*thmerad/2).\*sin(n. \*pi/2);

lambdan **=** ((2\*Rs\*Lst\*Na). \*kwn.\*kn.\*Bn)./p;

Ean **=** (omega. \*Iambdan); **%** Peak back voltage

**%** Normalized values for plotting

Eanorm **=** abs(Ean) ***J1*** Ean(1);

**%** Voltage THD

Eah=0;

for r =2:length(n)

Eah **=** Eah **+** Ean(r)A2;

end

THD **=** 100\*sqrt(Eah/(Ean(1)A2));

**"%**G enerate waveforms

**"%**R otor physical angle goes from **0** to 2\*pi **-** electrical to 2\*p\*pi

ang **=** *O:pi/lOO:2\*pi;*

angp, **=** p\*ang;

Bout **=** zeros(size(angp));

Eaout **=** zeros(size(angp));

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

Bout **=** Bout **+** Bn(i).\*sin(n(i).\*angp);

Eaout **=** Eaout **+** Ean(i).\*sin(n(i).\*angp);

end

**%** Plot waveforms

figure( 1)

plot(ang,Bout);

title('PM Generator: Flux Density');

**162**

ylabel(CB (Tesla)');

xlabelQ'Rotor Angle (rad)');

figure(2)

plot(ang,Eaout);

title(QPM Generator: Back EMF);

ylabel('Peak Voltage (V)');

xlabel(QRotor Angle (rad)');

figure(3)

hold on

title(['PM Generator: EMF Harmonics, THD = ,num2str(THD), %']);

ylabel('Normalized Back EMF');

xlabel('Harmonic Number');

text(20,O.7,'Dark: Above **10%** of Fundamental','FontSize', 10);

text(20,0.65,'Light: Below **10%** of Fundamental',FontSize', 10);

for z **=** 1 :Iength(Eanorm)

if Eanorm(z) **< 0.10**

bar(n(z),Eanorm(z),'c');

else

bar(n(z),Eanorm(z),'b');

end

end

hold off

**163**

Page Intentionally Left Blank

164

Appendix H. MATLAB Code: Retaining Sleeve Stress Calculations

**"%**Jo nathan Rucker, MIT Thesis

**"%**M ay ***2005***

**%** Program: pmcanstress

**"%**P rogram calculates and outputs retaining can stress

**"%**fo r permanent magnet machines with surface magnets and

**"%**sl otted stators.

**"%M UST** RUN pmiinput, pmicalc, and pmwave PRIOR TO **RUNNING** pmcanstress

**%** Calculate retaining sleeve stresses

**%** Conversion

Patopsi **=** 1.45038e-4; **%** psi per Pa

**%** Material yield stresses (ksi)

Stain-str **=90;**

Alum-str ***=75;***

Titan -str **=110;**

CarFib\_str **=100;**

Inconel-str **=132;**

**%** Safety factor

**SF =** 1.2;

**"%**F orce on magnets/sleeves is centrifugal force

**"%**M agnet tangential velocity

vmag **=** ((R+hm)\*omega)/p;

**%** Centrifugal force

Fm **=** (Mm\*vmagA2)/(R+hm);

**%** Outward pressure

Phoop **=** Fm/(2\*pi\*(R+hm)\*Lst);

**%** Hoop Stress (in general, str **=** P\*R/t)

stop **=** 22;

for **i=** I: stop

t(i) **=** i\*.0005; **%** sleeve thickness **t**

slev(i) **=** t(i)\*1000;

Sthoop(i) **=** (Phoop\*(R+hm)/t(i))\*Patopsi/l 000;

SFHoop(i) **=** Sthoop(i)\*SF;

end

**%** Output results

fprintf('Retaining Sleeve Stress:\n');

fprintf(QStress Limits:\n');

fprintf('Stainless Steel **= %6.lf** ksi Aluminum Alloy **=%6.1f** ksi~n',Stain-str,Alum-str);

fprintfQ'Titanium Alloy **= %6. if** ksi Carbon Fiber **= %6. if** ksi\n',Titan-StrCarFib str);.

fprintf('Inconel **=%6. If** ksi\n',Inconel-str);

fprintfQ'Actual Sleeve Stress:\n');

fprintf('Sleeve Thickness Actual Stress **SF** Stress\n');

for i **=** 1:stop

fprintf(Q ***%5.2f*** mm **%6. if** ksi **%6. If** ksi\n',...

slev(i),Sthoop(i),SFHoop(i));

end

Page Intentionally Left Blank

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

**172**

**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

Page Intentionally Left Blank