Оглавление

[Appendix E. MATLAB Code: Sizing Method 2 1](#_Toc318122073)

[% General variables 1](#_Toc318122074)

[% Rotor variables 1](#_Toc318122075)

[% Stator variables 1](#_Toc318122076)

[% Densities 2](#_Toc318122077)

[% Constants to be used 3](#_Toc318122078)

[% Calculate electrical frequency & rotor radius 3](#_Toc318122079)

[% Calculate magnet dimensions, tooth width, & air gap flux density 3](#_Toc318122080)

[% Set final values 4](#_Toc318122081)

[% Calculate magnetic gap factor 4](#_Toc318122082)

[% Calculate magnetic flux and internal voltage 6](#_Toc318122083)

["% Calculation of inductances/reactances 6](#_Toc318122084)

[% Output Section: 8](#_Toc318122085)

|  |  |  |  |
| --- | --- | --- | --- |
| Процедура | Название в delphi | Входные параметры | Выходные параметры |
| Calculate electrical frequency and rotor radius | ElectricalFrequencyRotorRadius | p – extended;  q – extended;  m | Ks  R  kp  kb  kw |
| Calculate magnet dimensions, tooth width and air gap flux density | MagnetDimensionsToothWidthAirGapFluxDensity | PC  R  Ws  Bg  Bsat | ge  eratio |
| Set final values | Final Values | thme  p  ge  Cphi  PC  R  G  wt  ws  hd  Ns | tfrac  wst  wsb  dc  Rci  Rco  lel |
| Calculate magnetic gap factor | MagneticGapFactor | R  hm  g  RSA  A  p  rhos  Rc  Ns  wt  hs  hd  wd  Bg  tfrac  dc  epsf  epsb  FO  f  BO  lams  m  Ja  As  q  Ds  pi | Lst  Ra  Pa  Lac  Pgap  A  PcperL  Bb  Bt  Mcperl |
| Calculate magnetic flux and internal voltage | MagneticFluxInternalVoltage | thm  pi  Bg  kg  p  Rs  Lst  Na  kw  ks  omega | Ea |
| Calculation of inductances/reactances | InductancesReactances | pi  Ia  Na  kw  Rs  g  hm  hs  wst  hd  Lst  perm  Nc  Ns  ws  wt  rhoc  McperL  thmrad  omega  R  g  nuair  Ea  muO | Pin  Ja  Ia  Ptemp  Perr(k)  Pwind  Rey  Mser  Mm  Mc  Dmach  Lmach  Mac  Lac  Lslot |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## 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

% 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;

# % 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);

% 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

% 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));

% 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);

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);