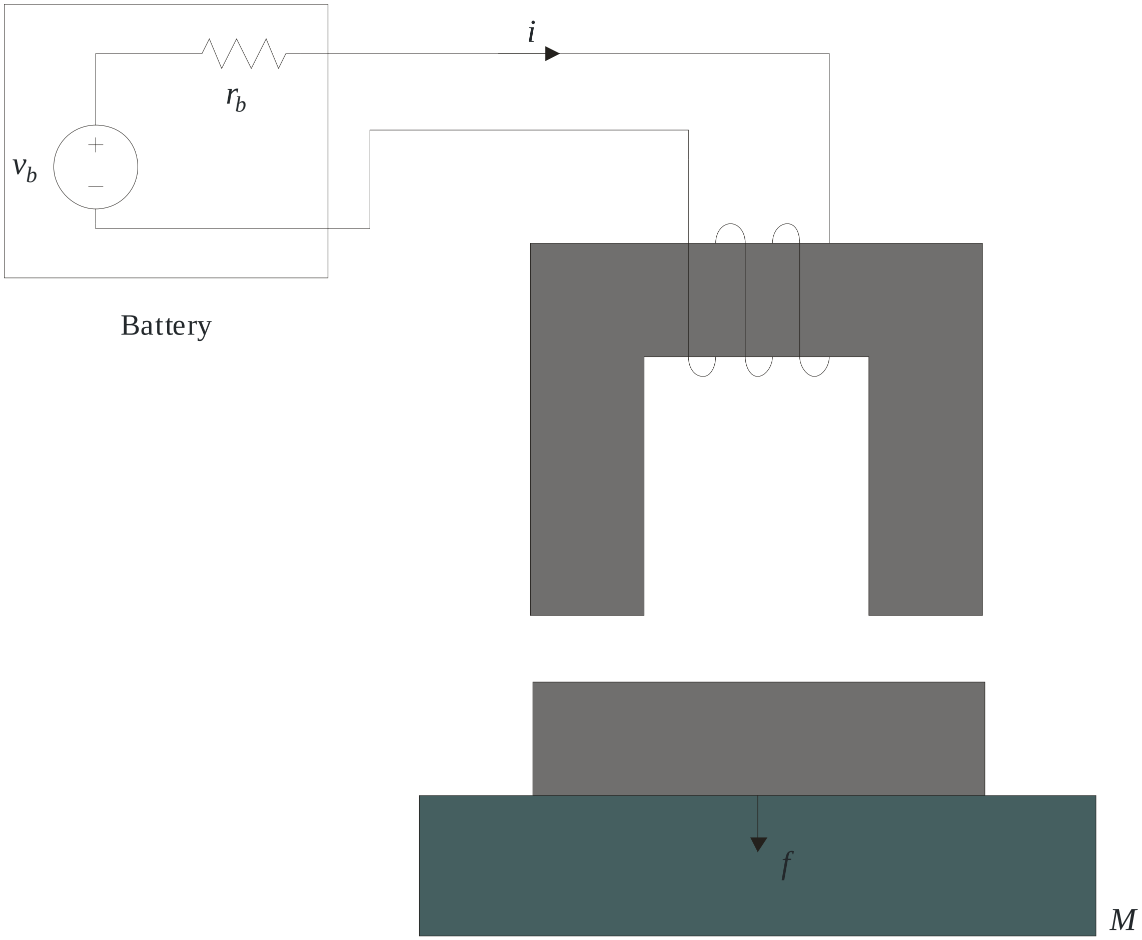
# Lecture 8:
# A Design Example - An Electromagnet

# Objective

- Use a genetic algorithm to solve an engineering design example

# The Problem

- Design an electromagnet to lift a mass.



# Specifications

- Mass
  - 10 kG
  - 10 cm by 10 cm area
  - Magnetic material should not overlap mass
  - Must lift from distance of 5 mm
  - Gravity is on surface of Earth ($G=9.8\ m/s^2$)
- Battery
  - $v_b = 12$ V
  - $r_b = 0.5\ \Omega$
  - Current should not exceed 6 A

# Specifications (Continued)

- Winding
  - Should not exceed packing factor of 0.7
  - May use Aluminum (Al) or Copper (Cu)
  - Aluminum
    - Density ($\rho_w$) = 2701 Kg/m$^3$
    - Conductivity ($\sigma_w$) = 35.4 1 1/(μmΩ)
    - Current Density ($j_{max}$) = 6.14 MA/m$^2$
  - Copper
    - Density ($\rho_w$) = 8900 Kg/m$^3$
    - Conductivity ($\sigma_w$) = 58.0 1/(μmΩ)
    - Current Density ($j_{max}$) = 7.62 MA/m$^2$

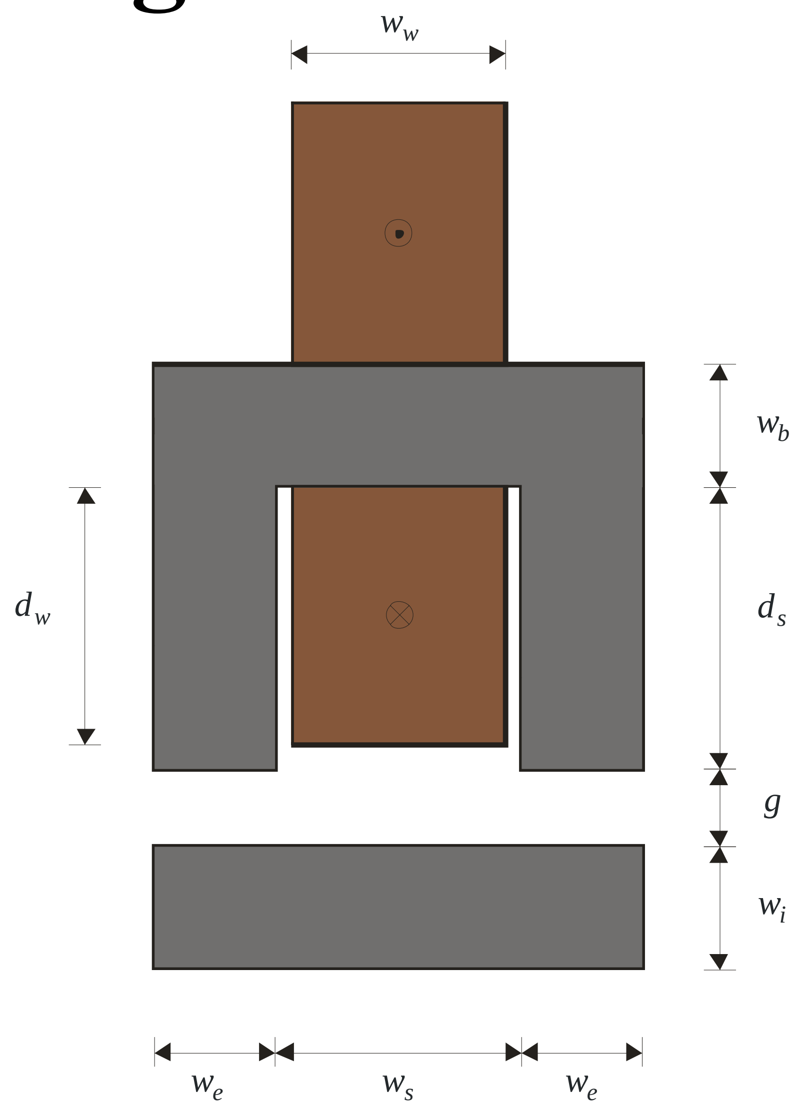
# Specifications (Continued)

- Magnetic Material
  - Microsil
    - $B_{sat}$ = 1.4 T, $\mu$ = 15000 $\mu_0$, $\rho_m$ = 7064 $Kg/m^3$
  - Superperm 80
    - $B_{sat}$ = 0.7 T, $\mu$ = 6000 $\mu_0$, $\rho_m$ = 8069 $Kg/m^3$
  - Superperm 49
    - $B_{sat}$ = 1.2 T, $\mu$ = 3500 $\mu_0$, $\rho_m$ = 7892 $Kg/m^3$

# Specifications (Continued Again)

- Size (Mass)
  - Total mass of electromagnet should be as small as possible

# Electromagnet Dimensions




Depth into
page = d

# Design Variables

- Wire
  - Material Type $m_w$
    - 1=Cu, 2=A)
  - Conductor Cross Section $a_c$
  - Number of Turns $N$
- Magnetic Material
  - Material Type $m_m$
    - 1=Microsil, 2=Superperm 80, 3=Superperm 49)

# Design Variables (Continued)

- Core
  - Slot width $w_s$
  - Wire window width $w_w$
  - Slot depth $d_s$
  - Wire window depth $d_w$
  - End width $w_e$
  - I-core width $w_i$
  - Back width $w_b$
  - Depth $d$

# Parameter Vector

$$\theta = \begin{bmatrix} m_w \\ a_w \\ N \\ m_m \\ w_s \\ r_{ww} \\ d_s \\ r_{dw} \\ w_e \\ r_{wi} \\ r_{wb} \\ d \end{bmatrix}$$

$$w_w = r_{ww} w_s$$

$$d_w = r_{dw} d_s$$

$$w_i = r_{wi} w_e$$

$$w_b = r_{wb} w_e$$

# Data Vector

$$\Psi = \begin{bmatrix} M \\ G \\ g \\ \rho_w \\ \rho_m \\ \sigma_w \\ B_{sat} \\ i_{max} \\ J_{max} \\ l_{max} \\ p_{f,max} \end{bmatrix}$$

# Electrical Analysis

- Goal

$$\begin{bmatrix} p_f \\ j \\ i \end{bmatrix} = F_{elec}(\boldsymbol{\theta}, \boldsymbol{\Psi})$$

- Matlab m-file: Electrical_Analysis.m

# Electrical Analysis

- Packing factor

$$p_f = \frac{a_c N}{w_w d_w}$$

- Winding volume (conductor)

$$v_w = p_f \left( w_w d_w \left( 2d + 2w_b \right) + \pi d_w^2 w_w \right)$$

- Winding length

$$l_w = \frac{v_w}{a_c}$$

# Electrical Analysis (Cont.)

- Winding resistance

$$r_w = \frac{l_w}{a_c \sigma_w}$$

- Current

$$i = \frac{v_b}{r_b + r_w}$$

- Current Density

$$j = \frac{i}{a_c}$$

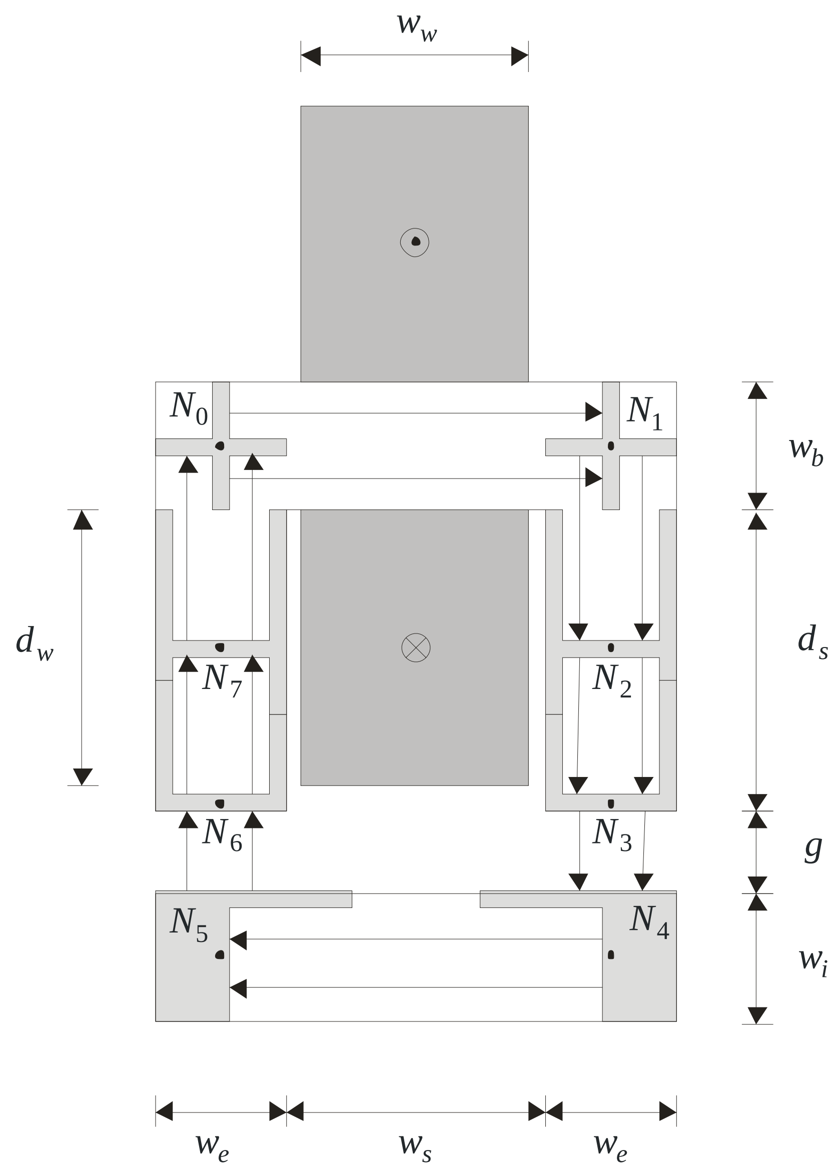
# Magnetic Analysis

- Goal

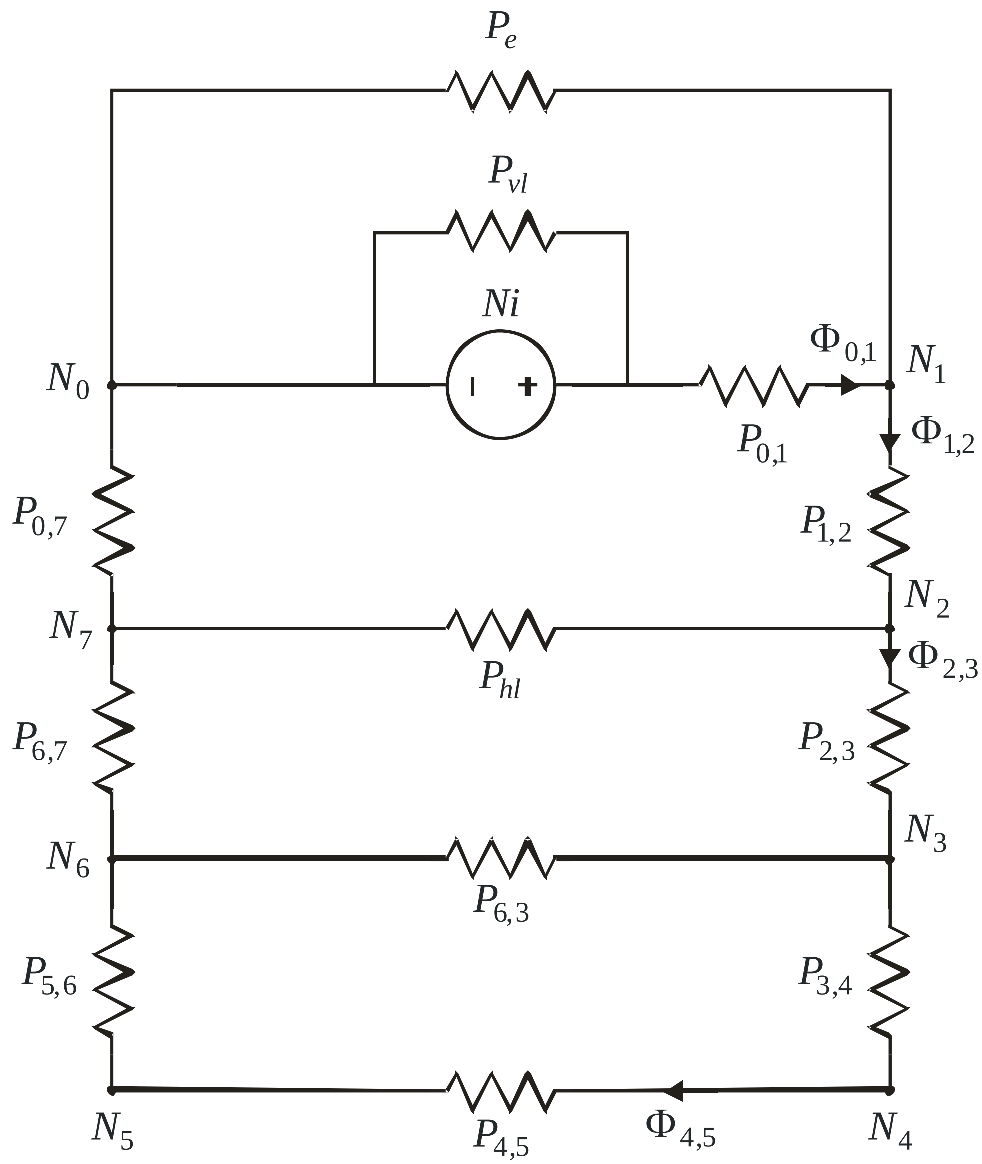
$$\begin{bmatrix} B_{0,1} \\ B_{1,2} \\ B_{2,3} \\ B_{4,5} \\ f_e \end{bmatrix} = F_{mag}\left(\boldsymbol{\theta}, i, \boldsymbol{\Psi}\right)$$

- Matlab m-file: Magnetic_Analysis.m

# Magnetic Equivalent Circuit



# Magnetic Equivalent Circuit (Cont)



# Review: Magnetic Equivalent Circuits

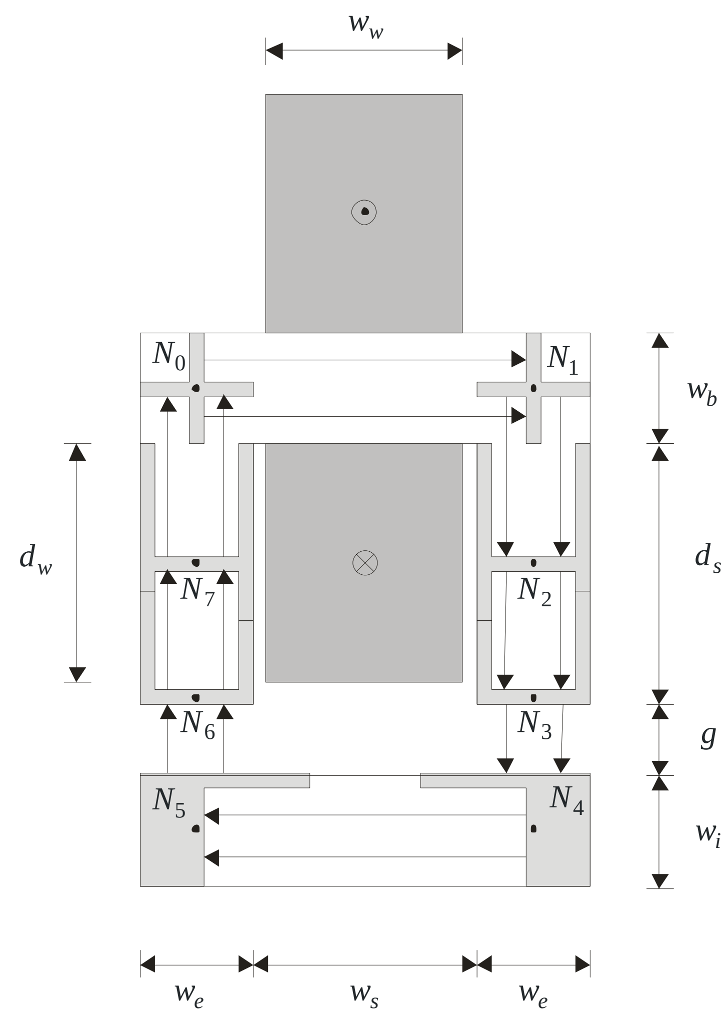
# Magnetic Material Permeances

$$P_{0,1} = \frac{w_b d\mu}{w_s + w_e}$$

$$P_{1,2} = P_{0,7} = \frac{2w_e d\mu}{w_b + d_w}$$

$$P_{2,3} = P_{6,7} = \frac{w_e d\mu}{d_s - d_w / 2}$$

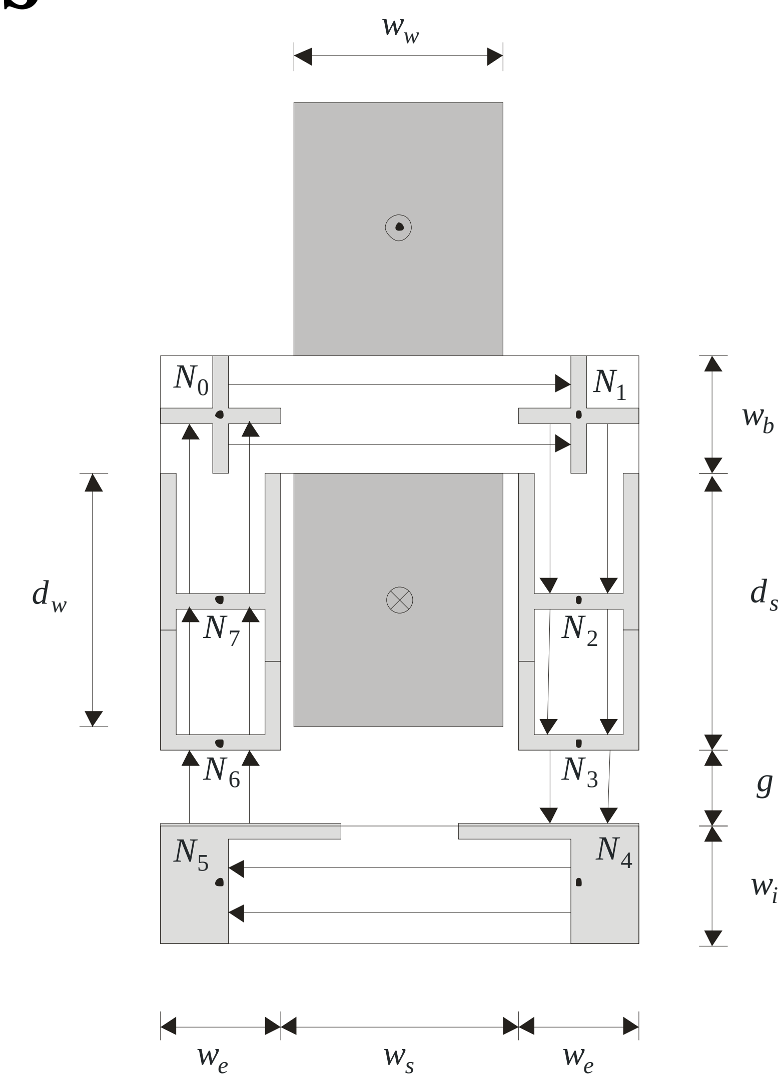
$$P_{4,5} = \frac{w_i d\mu}{w_s + w_e}$$



# Air Gap Permeances

$$P_{6,3} = \frac{(d_s - d_w) d \mu_0}{w_s} + \frac{2\mu_0 (d_s - d_w)}{\pi} \ln\left(1 + \frac{\pi w_e}{w_s}\right)$$

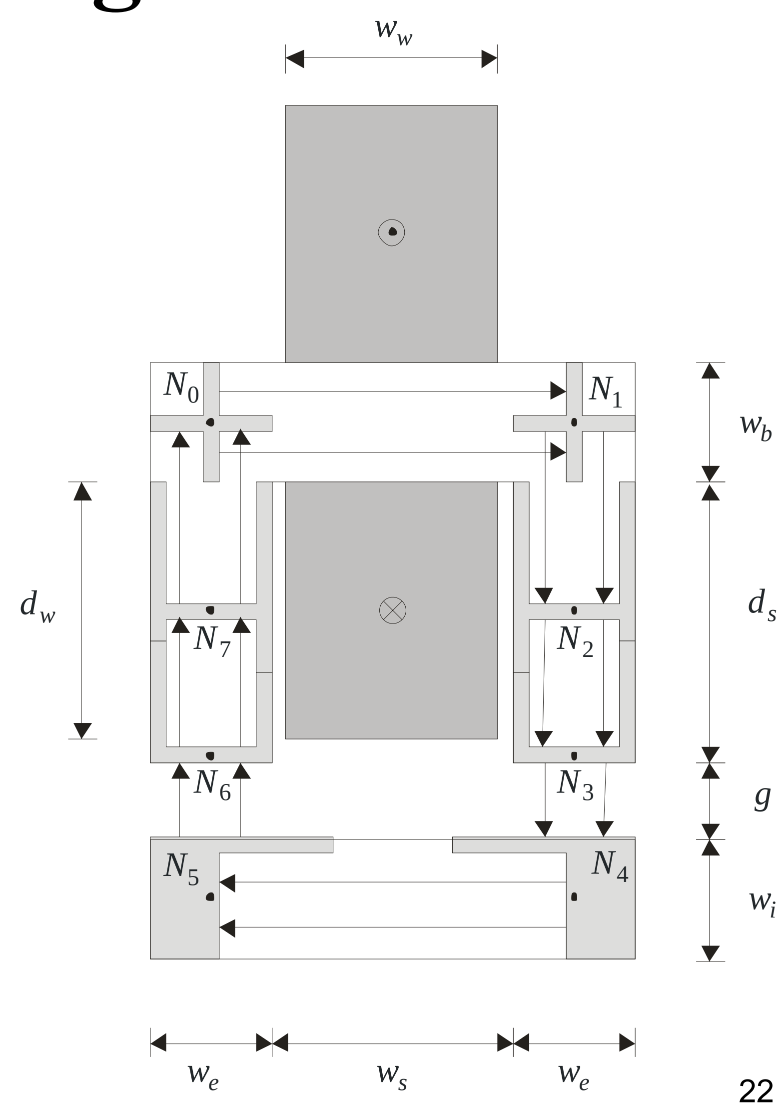
$$P_{3,4} = P_{5,6} = \underbrace{\frac{w_e d \mu_0}{g}}_{main\ face} + \underbrace{\frac{\mu_0 d}{\pi} \ln\left(1 + \frac{\pi w_i}{g}\right)}_{outer\ face} + \underbrace{\frac{2\mu_0 d}{\pi} \ln\left(1 + \frac{\pi \min(w_s, d_s)}{4g}\right)}_{inner\ face} + \underbrace{\frac{2\mu_0 w_e}{\pi} \ln\left(1 + \frac{\pi w_i}{g}\right)}_{front/back\ faces}$$



# Hor. And Vert. Leakage Permeances

$$P_{hl} = \frac{1}{3}\frac{\mu_0 d_w d}{w_s} + \frac{2\mu_0 d_w}{3\pi}\ln\left(1 + \frac{\pi w_e}{w_s}\right)$$

$$P_{vl} = \frac{1}{6}\frac{\mu_0 d w_w}{d_s + g}$$



# End Leakage Permeance

$$w_t = d_s - d_w$$

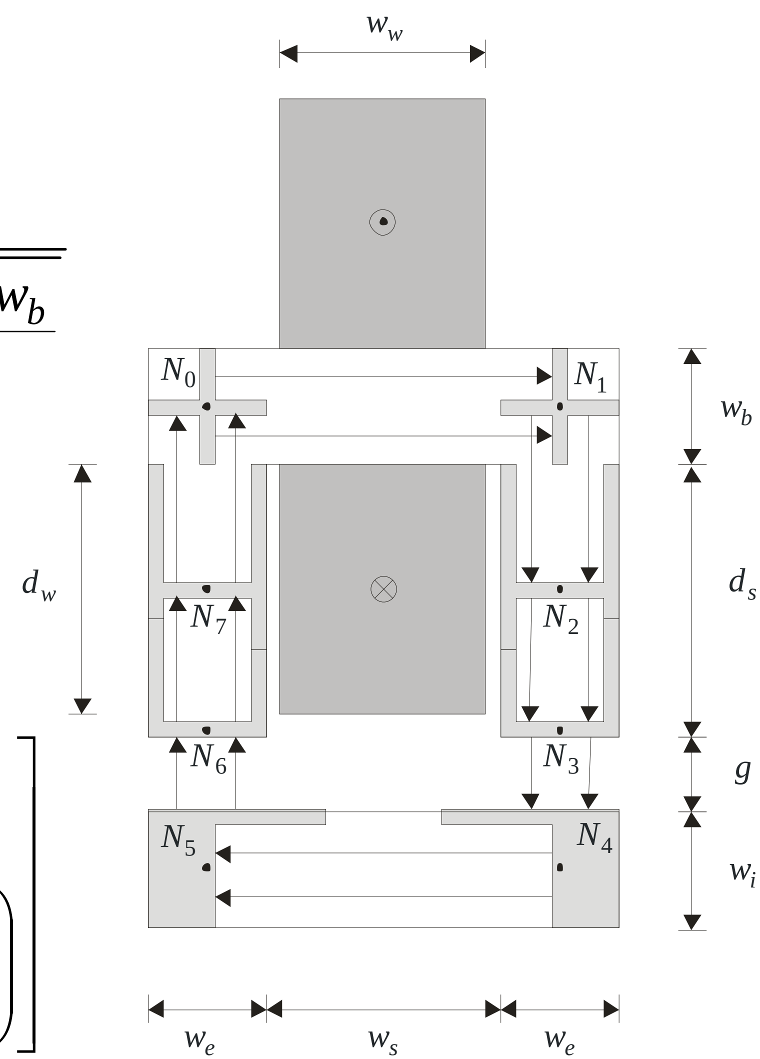
$$P_{e1} = \frac{\mu_0 w_t w_b (d + 2w_e)}{w_t + w_b} \frac{1}{w_w + \sqrt{\frac{(2d_w + w_t + w_b) w_t w_b}{w_t + w_b}}}$$

$$k_1 = \left| d_w - 2w_w \right|$$

$$k_2 = \begin{cases} \sqrt{2} w_w & d_w > 2w_w \\ \frac{d_w}{\sqrt{2}} & d_w \le 2w_w \end{cases}$$

$$P_{e2} = \frac{\mu_0 (d + 2w_e)}{8 w_w^2 d_w^2} \left[ \begin{array}{l} k_2^4 + \sqrt{2} k_1 k_2^3 + \frac{1}{4} k_1^2 k_2^2 \\ - \frac{\sqrt{2}}{8} k_1^3 k_2 + \frac{k_1^4}{16} \ln\left(1 + \frac{2\sqrt{2} k_2}{k_1}\right) \end{array} \right]$$

$$P_e = P_{e1} + P_{e2}$$



# Nodal Equations

$$\left(F_1 - Ni\right)P_{0,1} + F_1 P_e + \left(F_1 - F_2\right)P_{1,2} = 0$$

$$\left(F_2 - F_1\right)P_{1,2} + \left(F_2 - F_7\right)P_{hl} + \left(F_2 - F_3\right)P_{2,3} = 0$$

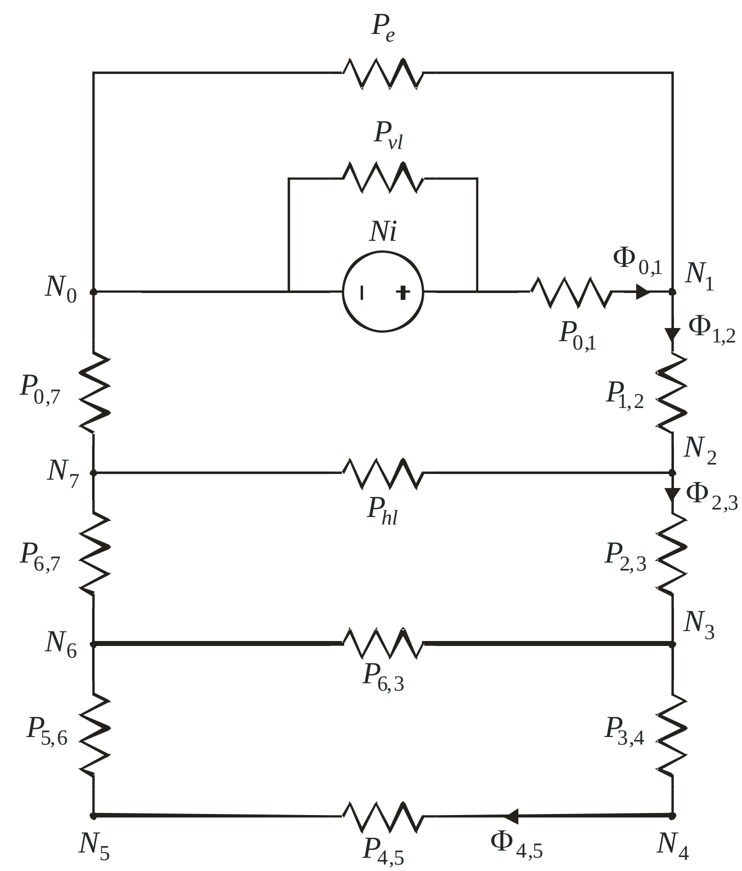
$$\left(F_3 - F_6\right)P_{6,3} + \left(F_3 - F_4\right)P_{3,4} + \left(F_3 - F_2\right)P_{2,3} = 0$$

$$\left(F_4 - F_3\right)P_{3,4} + \left(F_4 - F_5\right)P_{4,5} = 0$$

$$\left(F_5 - F_4\right)P_{4,5} + \left(F_5 - F_6\right)P_{5,6} = 0$$

$$\left(F_6 - F_7\right)P_{6,7} + \left(F_6 - F_5\right)P_{5,6} + \left(F_6 - F_3\right)P_{6,3} = 0$$

$$F_7 P_{0,7} + \left(F_7 - F_2\right)P_{hl} + \left(F_7 - F_6\right)P_{6,7} = 0$$



# Nodal Equations (Cont)

$$P = \begin{bmatrix} P_{0,1}+P_e+P_{1,2} & -P_{1,2} & 0 & 0 & 0 & 0 & 0 \\ -P_{1,2} & P_{1,2}+P_{hl}+P_{2,3} & -P_{2,3} & 0 & 0 & 0 & -P_{hl} \\ 0 & -P_{2,3} & P_{6,3}+P_{3,4}+P_{2,3} & -P_{3,4} & 0 & -P_{6,3} & 0 \\ 0 & 0 & -P_{3,4} & P_{3,4}+P_{4,5} & -P_{4,5} & 0 & 0 \\ 0 & 0 & 0 & -P_{4,5} & P_{4,5}+P_{5,6} & -P_{5,6} & 0 \\ 0 & 0 & -P_{6,3} & 0 & -P_{5,6} & P_{6,7}+P_{5,6}+P_{6,3} & -P_{6,7} \\ 0 & -P_{hl} & 0 & 0 & 0 & -P_{6,7} & P_{0,7}+P_{hl}+P_{6,7} \end{bmatrix}$$

$$F = NiP_{0,1}P^{-1}\Big|_{1st \quad column}$$

$$\begin{bmatrix} F_1 & F_2 & F_3 & F_4 & F_5 & F_6 & F_7 \end{bmatrix} = F^T$$

# Fluxes

$$\Phi_{0,1} = (Ni - F_1)P_{0,1}$$

$$\Phi_{1,2} = (F_1 - F_2)P_{1,2}$$

$$\Phi_{2,3} = (F_2 - F_3)P_{2,3}$$

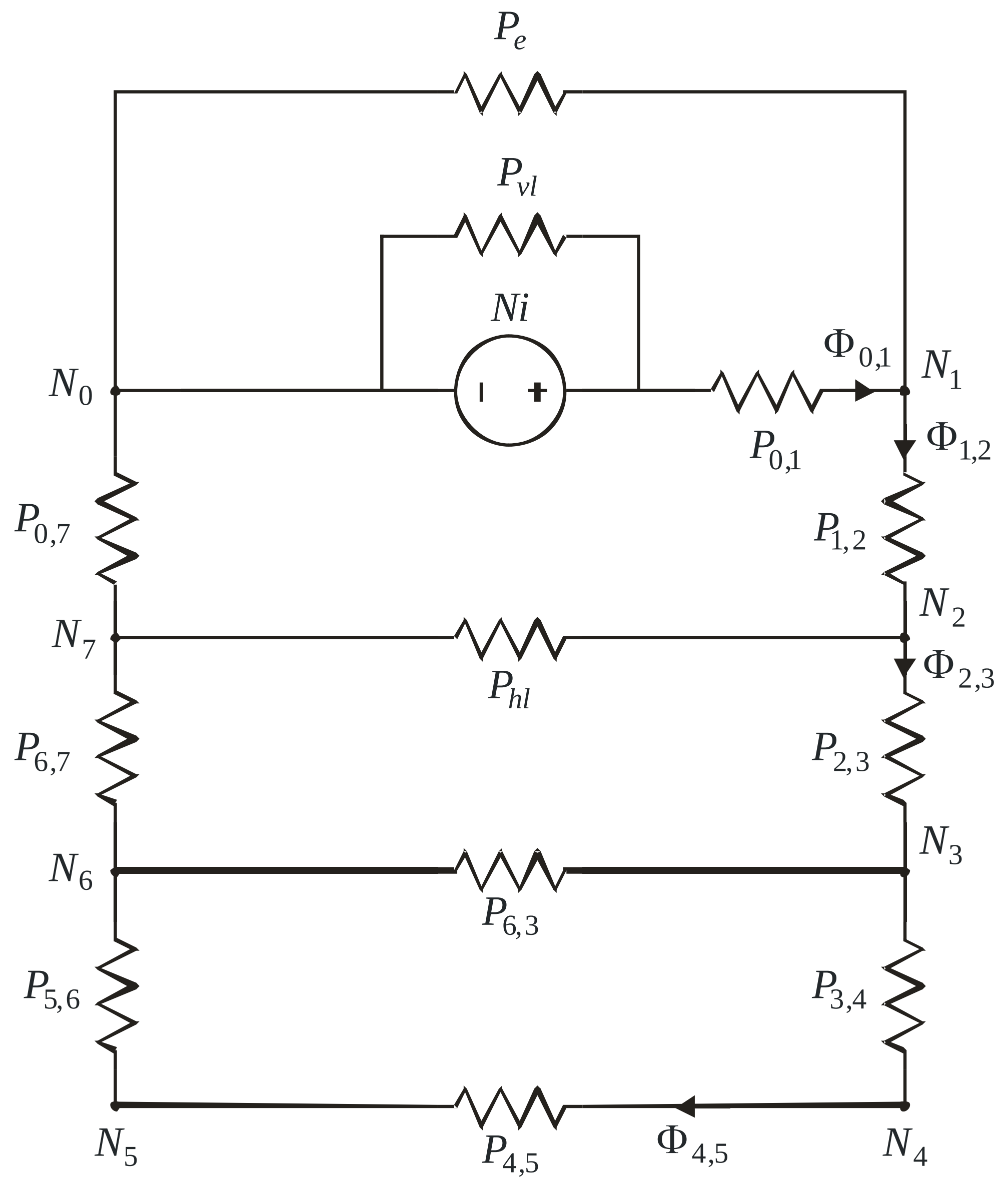
$$\Phi_{3,4} = (F_3 - F_4)P_{3,4}$$

$$\Phi_{4,5} = \left(F_4 - F_5\right)P_{4,5}$$

$$\Phi_{5,6} = (F_5 - F_6)P_{5,6}$$

$$\Phi_{6,7} = (F_6 - F_7)P_{6,7}$$

$$\Phi_{0,7} = F_7 P_{0,7}$$



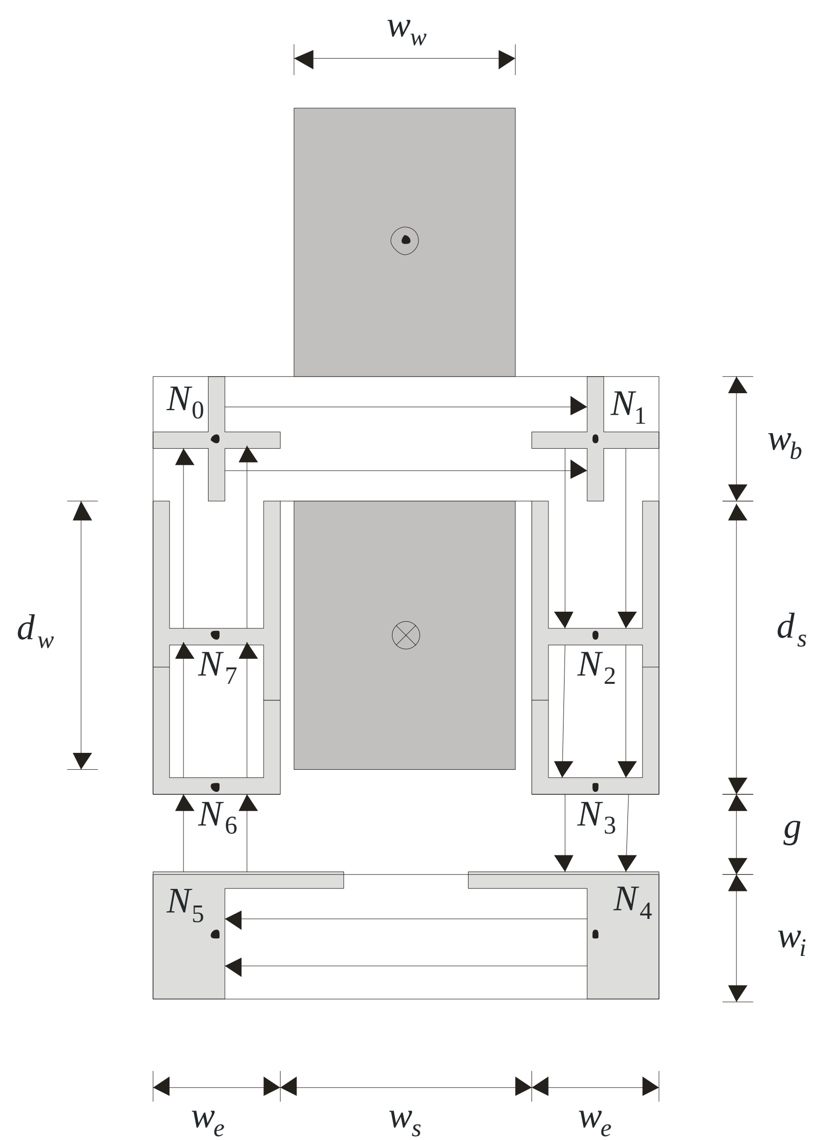
# Flux Densities

$$B_{0,1} = \frac{\Phi_{0,1}}{w_b d}$$

$$B_{1,2} = \frac{\Phi_{1,2}}{w_e d}$$

$$B_{2,3} = \frac{\Phi_{2,3}}{w_e d}$$

$$B_{4,5} = \frac{\Phi_{4,5}}{w_i d}$$



# Force

- Using co-energy techniques and assuming magnetic linearity it can be shown

$$\frac{\partial P_{3,4}}{\partial g} = \frac{\partial P_{5,6}}{\partial g} = -\frac{w_e d\mu_0}{g^2} - \frac{\mu_0 d w_i}{g(g+\pi w_i)} - \frac{2\mu_0 d \min(w_s, d_s)}{4g(4g+\pi \min(w_s, d_s))} - \frac{2\mu_0 d w_e w_i}{g(g+\pi w_i)}$$

$$\frac{\partial P_{vl}}{\partial g} = -\frac{1}{6}\frac{\mu_0 d w_w}{(d_s+g)^2}$$

# Force (Cont.)

$$\frac{\partial P}{\partial g} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\partial P_{3,4}}{\partial g} & -\frac{\partial P_{3,4}}{\partial g} & 0 & 0 & 0 \\ 0 & 0 & -\frac{\partial P_{3,4}}{\partial g} & \frac{\partial P_{3,4}}{\partial g} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\partial P_{5,6}}{\partial g} & -\frac{\partial P_{5,6}}{\partial g} & 0 \\ 0 & 0 & 0 & 0 & -\frac{\partial P_{5,6}}{\partial g} & \frac{\partial P_{5,6}}{\partial g} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

# Force (Continued)

$$\frac{\partial L}{\partial g} = N^2 P_{01}^2 \left[P^{-1}\right]\Big|_{\text{1st row}} \frac{\partial P}{\partial g} \left[P^{-1}\right]\Big|_{\text{1st column}} + N^2 \frac{\partial P_{vl}}{\partial g}$$

$$f_e = -\frac{1}{2}\frac{\partial L}{\partial g} i^2$$

# Setting Up Constraints

- Less than function

$$ltne(x, x_{max}, \Delta x) = \begin{cases} 1 & x \le x_{max} \\ \dfrac{1}{1 + \left| \dfrac{x_{max} - x}{\Delta x} \right|} & x > x_{max} \end{cases}$$

# Setting Up Constraints

- Greater than function

$$gtne(x, x_{min}, \Delta x) = \begin{cases} 1 & x \geq x_{min} \\ \dfrac{1}{1 + \left| \dfrac{x_{min} - x}{\Delta x} \right|} & x < x_{min} \end{cases}$$

# Geometrical Constraints

- Recall, magnetic material should not overlap mass

$$c_1 = ltne(w_s + 2w_e, l_{max}, 0.1 l_{\max})$$

$$c_2 = ltne(d, l_{max}, 0.1 l_{max})$$

# Electrical Constraints

- Packing Factor $c_3 = ltne(p_f, p_{f,max}, 0.1p_{f,max})$
- Current Density $c_4 = ltne(j, j_{max}, 0.1J_{max})$
- Current $c_5 = ltne(i, i_{max}, 0.1i_{max})$

# Flux Density Constraints

- Back of U-Core $c_6 = ltne(B_{0,1}, B_{sat}, 0.01B_{sat})$
- Back of Legs $c_7 = ltne(B_{1,2}, B_{sat}, 0.01B_{sat})$
- Front of Legs $c_8 = ltne(B_{2,3}, B_{sat}, 0.01B_{sat})$
- I-Core $c_9 = ltne(B_{4,5}, B_{sat}, 0.01B_{sat})$

# Force Constraints

- We must lift mass (and I-core)

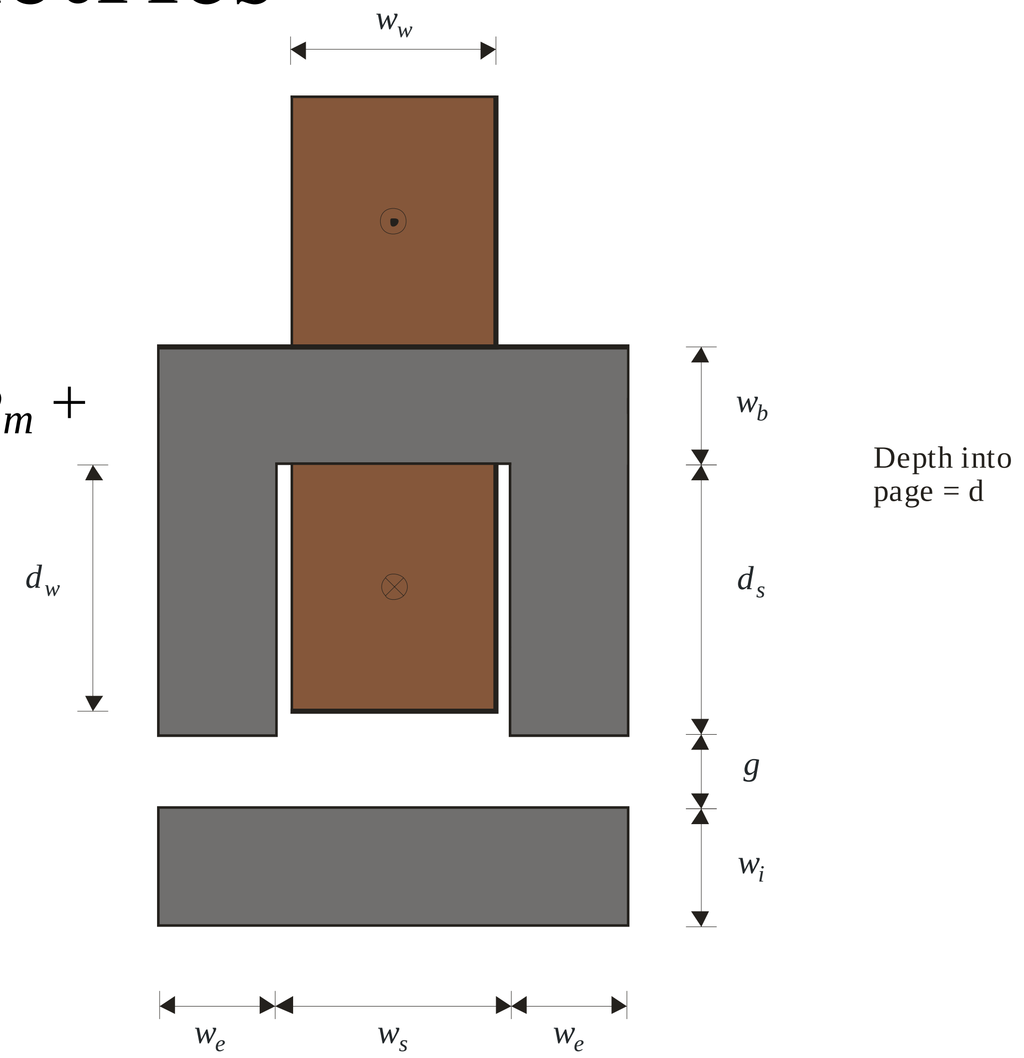
$$f_{req} = (M + w_i d(w_s + 2w_e)\rho_m)G$$

$$c_{10} = gtne(f_e, f_{req}, 0.01 f_{req})$$

# Performance Metrics

- Total mass

$$m_1 = \left(2d_s w_e d + (w_s + 2w_e)(w_b + w_i)d\right)\rho_m + p_f\left(w_w d_w(2d + 2w_b) + \pi d_w^2 w_w\right)\rho_w$$



# Fitness Function

- Combined constraint

$$c = \min(c_1, c_2, \cdots c_{10})$$

- Fitness Function

$$\mathbf{f} = \begin{cases} \dfrac{1}{m_1 + \varepsilon} & c = 1 \\ \sum_{c=1}^{10} c_i - 10 & c < 1 \end{cases}$$

# Code Organization

- Design Codes
  - Electromagnet_Design.m
    - GOSET
      - Electromagnet_Fitness.m
        - Electrical_Analysis.m
        - Magnetic_Analysis.m

# Code Organization (Cont)

- Supporting Routines
  - Electromagnet_Drawing1
  - Electromagnet_Drawing2
  - Electromagnet_Design_Review
  - Electromagnet_Design_Multirun
  - Electromagnet_Design_Multrun_Review

# Electromagnet_Design.m (1/3)

```
% Electromagnet Design
%
% Written by:
% Ricky Chan for S.D. Sudhoff
% School of Electrical and Computer Engineering
% 1285 Electrical Engineering Building
% West Lafayette, IN 47907-1285
% E-mail: sudhoff@ecn.purdue.edu

% Set Up Population
GAP         = gapdefault;
GAP.fp_ngen = 2500;
GAP.fp_ipop = 500;
GAP.fp_npop = 500;
GAP.mc_alg  = 4.0;
GAP.dt_alg  = 3;

% Set Up Migrations
GAP.mg_nreg=5;       % number of regions
GAP.mg_tmig=100;     % mean time between migrations
GAP.mg_pmig=0.05;    % probability of a individual migrating

% Units
mm=1.0e-3;
cm=1.0e-2;
```

# Electromagnet_Design.m (2/3)

```
% Problem Requirement Data
Psi.M      = 10;                                                  % Mass of object (Kg)
Psi.G      = 9.8;                                                 % Gravity (m/s2)
Psi.g      = 0.5*cm;                                              % Gap Width (m)
Psi.rhow   = [    8900;       2701];                              % Wire - Kg/m3
Psi.sigmaw = [    58.0;       35.4]*1e6;                          % Wire - A/Ohm
Psi.jmaxw  = [    7.62;       6.14]*1e6;                          % Wire - A/m2
Psi.descw  = ['Copper  '; 'Aluminum'];                            % Wire - Description
Psi.bmaxm  = [   1.4;              0.7;            1.2];          % Steel - Bsat, T
Psi.rhom   = [7064.1;           8069.4;         7892.1];          % Steel - Density Kg/m3
Psi.myum   = [ 15000;             6000;           3500];          % Steel - Perm., (relative)
Psi.descm  = ['Microsil    '; ...
              'Superperm 80'; ...
              'Superperm 49'];                                    % Steel - Description
Psi.imax   = 6;                                                   % Maximum current, A
Psi.lmax   = 10e-2;                                               % Maximum length
Psi.pfmax  = 0.7;                                                 % Maximum packing factor
Psi.vb     = 12;                                                  % Battery Voltage
Psi.rb     = 0.5;                                                 % Battery Resistance
```

# Electromagnet_Design.m (3/3)

```
% Genetic Mapping
%                mw   ac    N    mm    ws     rww    ds    rdw    we     rwi   rwb   d
GAP.gd_min  = [  1  1e-8   10    1    1*cm   0.1   1*mm   0.1   1*mm    0.2   0.2   1*mm];
GAP.gd_max  = [  2  1e-4  1e3    3   10*cm   1.0  20*cm   1.0   5*cm    2.0   2.0  10*cm];
GAP.gd_type = [  1    3     3    1     3      3     3      3      3      3     3     3  ];
GAP.gd_cid  = [  1    1     1    1     1      1     1      1      1      1     1     1  ];

% Solve Problem
[fP,GAS]= gaoptimize(@Electromagnet_Fitness,GAP,Psi,[],[],[]);

% Get Final Answer and Plot
final_parameters = GAS.bestgenes(:,end);
f = Electromagnet_Fitness(final_parameters,Psi,23);

% Save run
s=input('Type 1 to Save Run ');
if (s==1)
   save samplerun
end
```

# Electromagnet_Fitness.m (1/7)

```
function fitness = Electromagnet_Fitness(param,Psi,figNum)

% ELECTROMAGNET_FITNESS
%
% fitness = Electromagnet_Fitness(param,Psi,x)
%
% Inputs:
% param      = design parameters
% Psi        = data vectors
% x          = optional input to print out extra information
%
% Outputs:
% fitness    = fitness value
%
% Written by:
% Ricky Chan for S.D. Sudhoff
% School of Electrical and Computer Engineering
% 1285 Electrical Engineering Building
% West Lafayette, IN 47907-1285
% E-mail: sudhoff@ecn.purdue.edu
```

# Electromagnet_Fitness.m (2/7)

```
% Map to nicer variable names
mw      = param(1);
ac      = param(2);
N       = round(param(3));
mm      = param(4);
ws      = param(5);
rww     = param(6);
ds      = param(7);
rdw     = param(8);
we      = param(9);
rwi     = param(10);
rwb     = param(11);
d       = param(12);
```

# Electromagnet_Fitness.m (3/7)

```
% Computing the width of winding (ww), depth of winding (dw), width of the
% I core (wi), and the width of the back of the U core (wb)
ww = rww*ws;
dw = rdw*ds;
wi = rwi*we;
wb = rwb*we;

% Electrical Circuit Analysis
[pf, j, i] = Electrical_Analysis(param,Psi);

% Magnetic Analysis
[B01, B12, B23, B45, fe] = Magnetic_Analysis(param,Psi,i);

% Constraints
lmax = Psi.lmax;
pfmax = Psi.pfmax;
jmaxw = Psi.jmaxw(mw);
imax = Psi.imax;
bmax = Psi.bmaxm(mm);
rhom = Psi.rhom(mm);
rhow = Psi.rhow(mw);
```

# Electromagnet_Fitness.m (4/7)

```
c1  = ltne(ws+2*we,lmax,0.1*lmax);
c2  = ltne(d,lmax,0.1*lmax);
c3  = ltne(pf,pfmax,0.1*pfmax);
c4  = ltne(j,jmaxw,0.1*jmaxw);
c5  = ltne(i,imax,0.1*imax);
c6  = ltne(B01,bmax,0.01*bmax);
c7  = ltne(B12,bmax,0.01*bmax);
c8  = ltne(B23,bmax,0.01*bmax);
c9  = ltne(B45,bmax,0.01*bmax);
f_req = (Psi.M + wi*d*(ws+2*we)*rhom)*Psi.G;
c10 = gtne(fe,f_req,0.01*f_req);

% find the aggregrate constraint
c = [c1 c2 c3 c4 c5 c6 c7 c8 c9 c10];
cmin = min(c);

% design objective
m1 = (2*ds*we*d + (ws+2*we)*(wb+wi)*d)*rhom + ...
     pf*(ww*dw*(2*d+2*wb)+pi*dw^2*ww)*rhow;

% compute the fitness
if (cmin == 1)
    fitness = 1/(m1+1e-6);
else
    fitness = sum(c)-length(c);
end
```

# Electromagnet_Fitness.m (5/7)

```
% if more than 3 arguments, plot the result
% and write a report
if nargin >= 3

   format short g

   disp('Parameters');
   disp(['Parameter 1: Wire type ',Psi.descw(mw,:)]);
   disp(['Parameter 2: Wire cross section is ', num2str(ac), ' m^2']);
   disp(['Parameter 3: Number of turns is ', num2str(N)]);
   disp(['Parameter 4: Magnetic material is ', Psi.descm(mm,:)]);
   disp(['Parameter 5: Slot width is ',num2str(ws*100),' cm']);
   disp(['Parameter 6: Winding width is ',num2str(rww*100),'% slot width']);
   disp(['Parameter 7: Slot depth is ',num2str(ds*100),' cm']);
   disp(['Parameter 8: Winding depth is ',num2str(rdw*100),'% slot depth']);
   disp(['Parameter 9: End width is ',num2str(we*100),' cm']);
   disp(['Parameter 10: I width is ',num2str(rwi*100),'% end width']);
   disp(['Parameter 11: Base width is ',num2str(rwb*100),'% end width']);
   disp(['Parameter 12: Depth is ',num2str(d*100),' cm']);
   disp(' ');
```

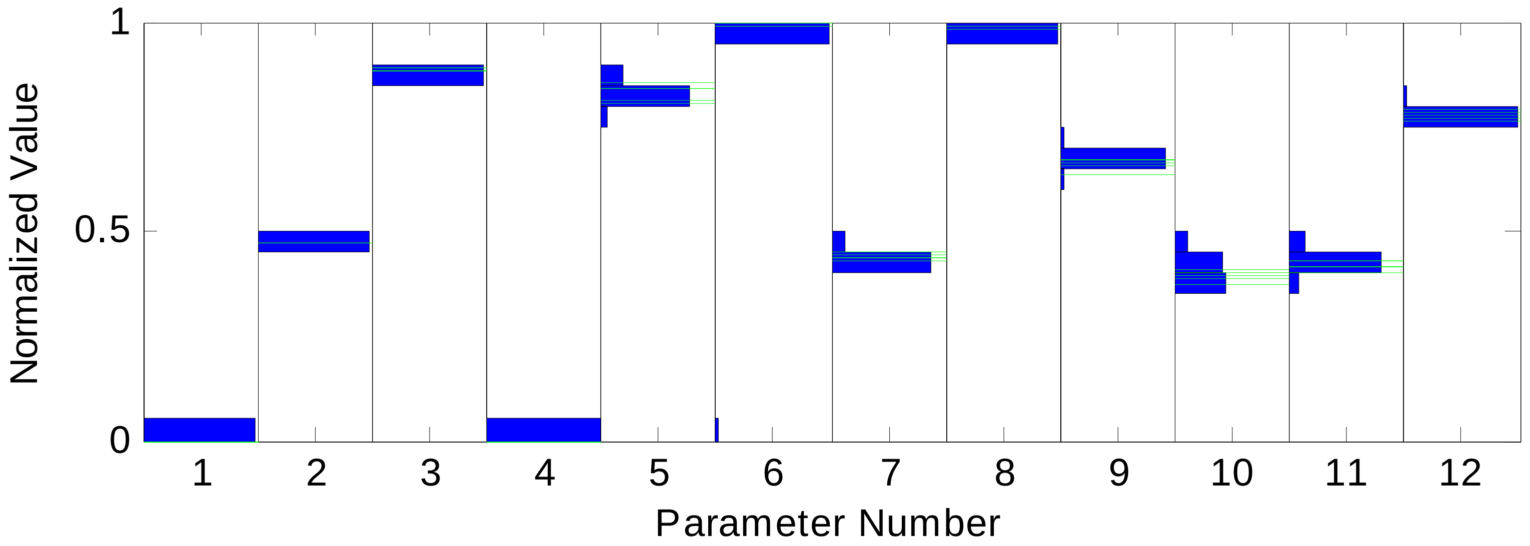
# Electromagnet_Fitness.m (6/7)

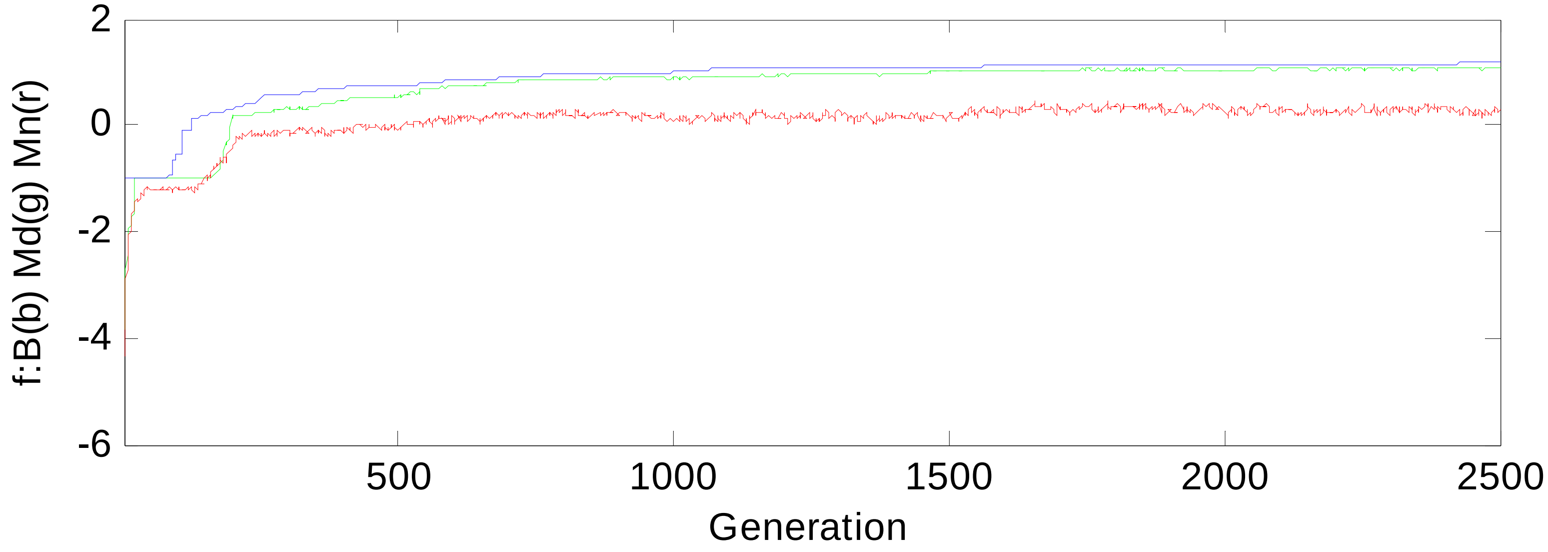
```
    disp('Constraints');
    disp(['Overall width is ',num2str(100*(ws+2*we)/lmax),'% allowed']);
    disp(['Overall depth is ',num2str(100*d/lmax),'% allowed']);
    disp(['Packing factors is ',num2str(100*pf/pfmax),'% allowed']);
    disp(['Current density is ',num2str(100*j/jmaxw),'% allowed']);
    disp(['Current is ',num2str(100*i/imax),'% allowed']);
    disp(['Back core flux density is ',num2str(100*B01/bmax),'% allowed']);
    disp(['Back end core flux density is ',num2str(100*B12/bmax),'% allowed']);
    disp(['Front end core flux density is ',num2str(100*B23/bmax),'% allowed']);
    disp(['I core flux density is ',num2str(100*B45/bmax),'% allowed']);
    disp(['Force required is ',num2str(f_req),' N']);
    disp(['Force produced is ',num2str(100*fe/f_req), '% required']);
    disp(' ');
    disp('Metrics');
    disp(['Mass is ',num2str(m1),' Kg']);

    figure(figNum);
    clf;
    Electromagnet_Drawing1(param,Psi);
    figure(figNum+1)
    Electromagnet_Drawing2(param,Psi);

end
```

# Electromagnet_Fitness.m (7/7)

```
function p = ltne(x,xmax,delta)

if x <= xmax
    p = 1;
else
    p = 1./(1+abs((xmax-x)./delta));
end

function p = gtne(x,xmin,delta)

if x >= xmin
    p = 1;
else
    p = 1./(1+abs((xmin-x)./delta));
end
```

# Sample Evolution



# Evolution – Best Fit Individual

- (Matlab Demo)

# Parameters

- Parameter 1: Wire type Copper
- Parameter 2: Wire cross section is 7.8519e-007 m^2
- Parameter 3: Number of turns is 585
- Parameter 4: Magnetic material is Microsil
- Parameter 5: Slot width is 7.1906 cm
- Parameter 6: Winding width is 97.7656% slot width
- Parameter 7: Slot depth is 0.96031 cm
- Parameter 8: Winding depth is 97.8117% slot depth
- Parameter 9: End width is 1.2104 cm
- Parameter 10: I width is 49.2545% end width
- Parameter 11: Base width is 52.2008% end width
- Parameter 12: Depth is 3.7947 cm

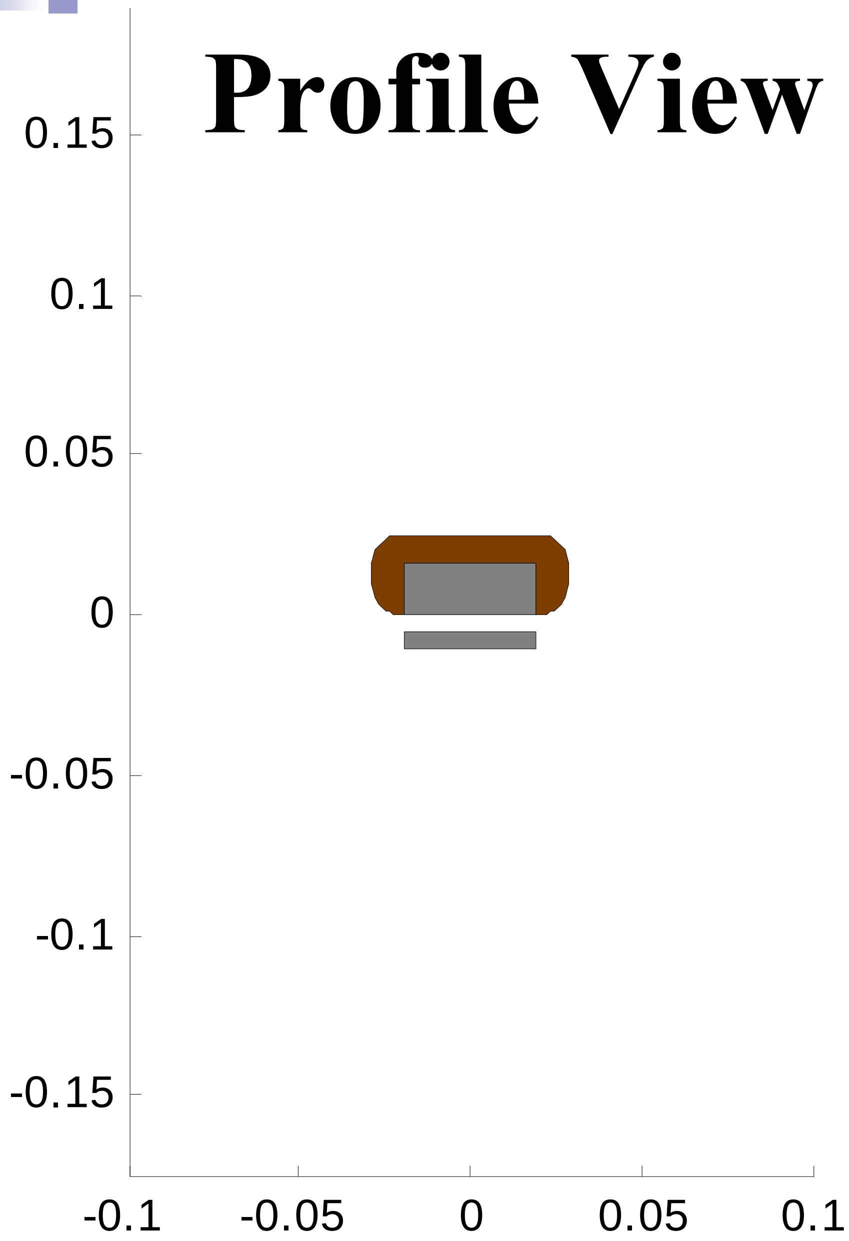
# Constraints

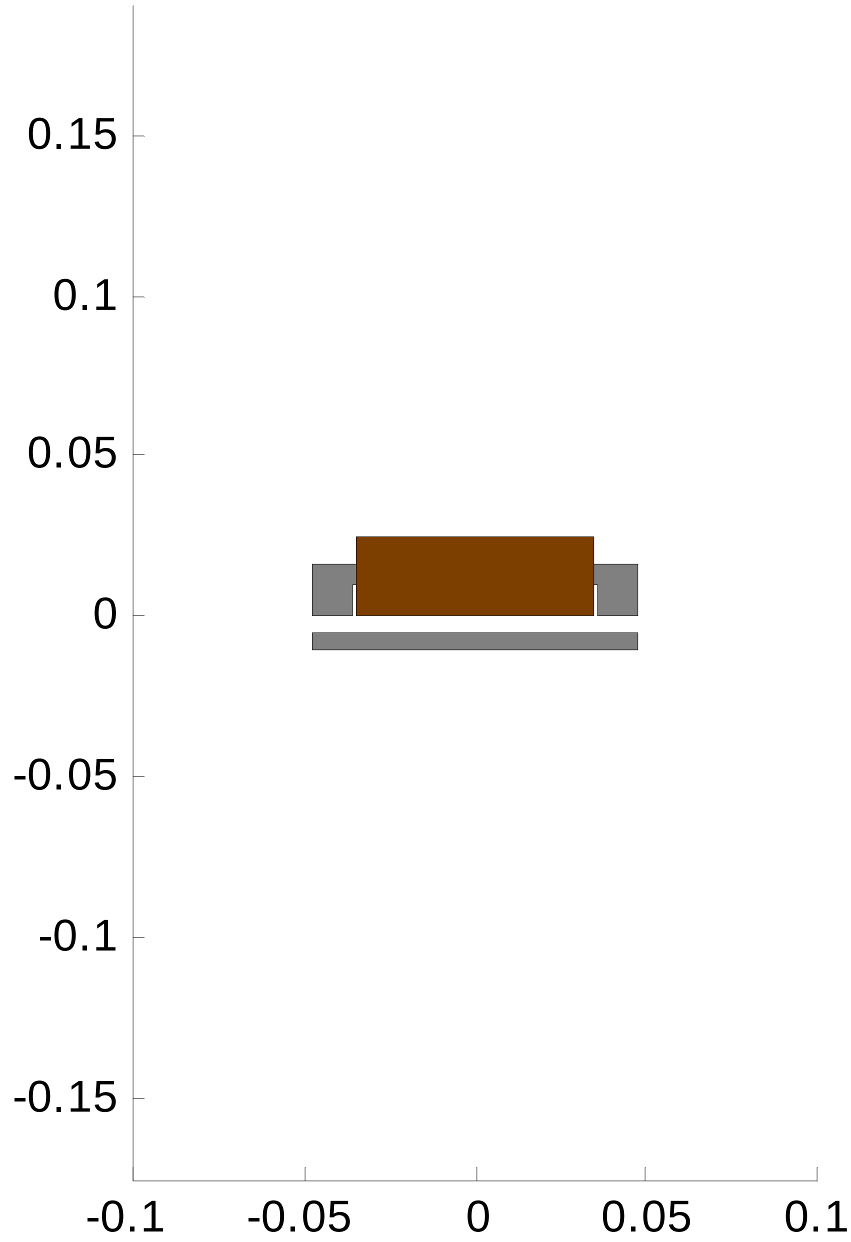
- Overall width is 96.1141% allowed
- Overall depth is 37.9474% allowed
- Packing factors is 99.3753% allowed
- Current density is 99.4713% allowed
- Current is 99.1917% allowed
- Back core flux density is 99.442% allowed
- Back end core flux density is 50.9116% allowed
- Front end core flux density is 49.215% allowed
- I core flux density is 99.6887% allowed
- Force required is 99.5053 N
- Force produced is 100.3524% required

# Metrics

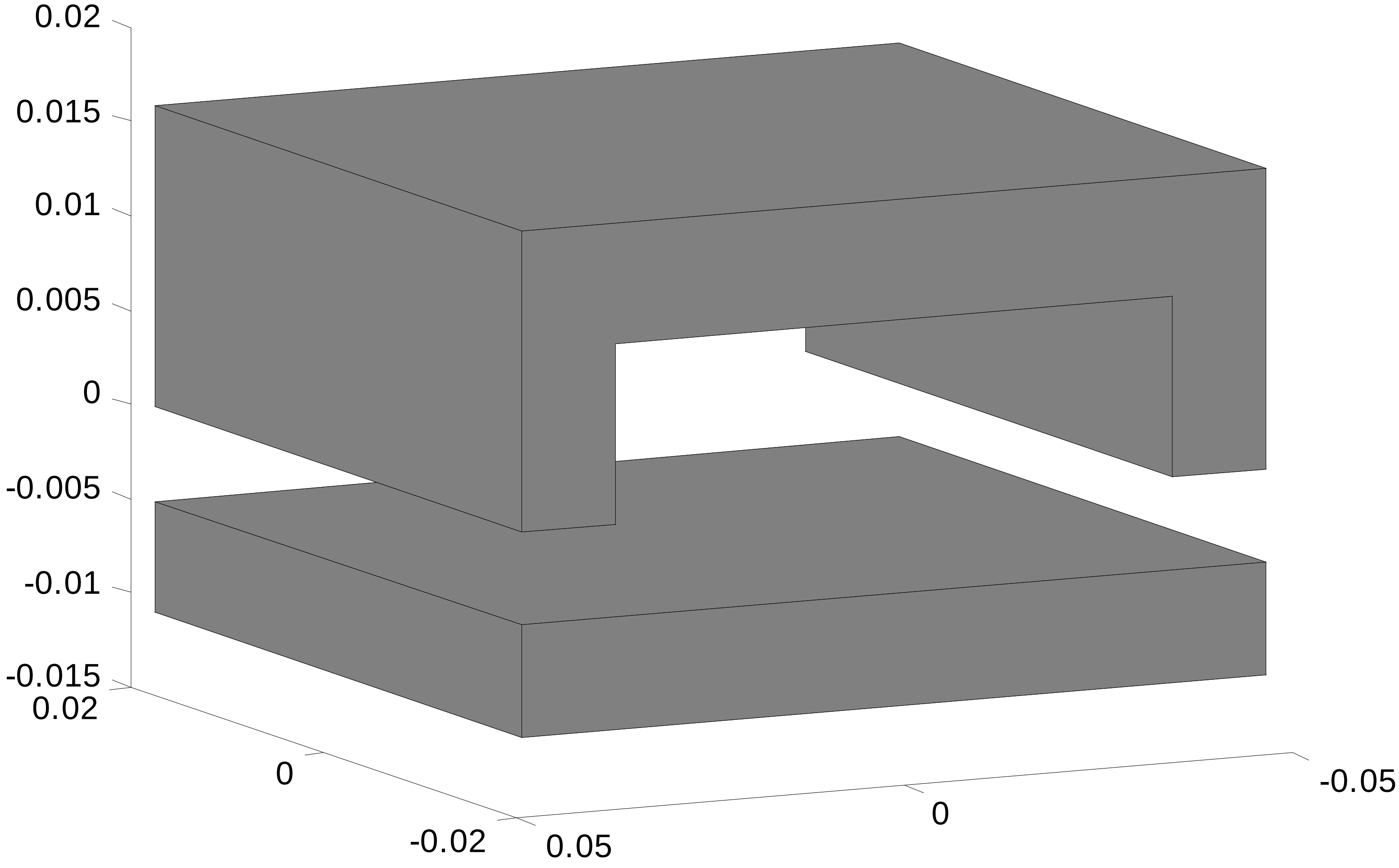
- Mass is 0.86127 Kg

# Profile View



# 3-D View



# Design Repeatability

- Repeat design 25 times:

| Parameter | Min/Mn | Max/Mn | Std/Mn |
|---|---|---|---|
| $m_w$ | 100.00 | 100.00 | 0 |
| $a_w$ | 95.26 | 102.05 | 1.46 |
| $N$ | 93.95 | 106.16 | 3.48 |
| $m_n$ | 100.00 | 100.00 | 0 |

# Design Repeatability

| Parameter | Min/Mn | Max/Mn | Std/Mn |
|---|---|---|---|
| $w_s$ | 88.44 | 110.01 | 5.06 |
| $r_{ww}$ | 97.93 | 100.72 | 0.57 |
| $d_s$ | 92.18 | 105.86 | 4.07 |
| $r_{dw}$ | 96.46 | 101.94 | 1.46 |

# Design Repeatability

| Parameter | Min/Mn | Max/Mn | Std/Mn |
|---|---|---|---|
| $w_e$ | 88.55 | 109.82 | 5.93 |
| $r_{wi}$ | 90.00 | 109.26 | 5.06 |
| $r_{wb}$ | 89.55 | 108.78 | 5.17 |
| $d$ | 83.18 | 118.91 | 9.02 |
| Fitness | 97.55 | 102.00 | 1.09 |