Winded Coils and Ferromagnetic Cores in Comsol Multiphysics (in progress)

Domenico Lahaye

July 14, 2009

Notes to Dalibor and Peers

We intend to further expand these notes for a number of different reasons among which

- we are motivated by creating a transparent inventory of what we know and what we do not yet know in the finite element modelling of inductive fault current limiters. In the *short term* this inventory will help us further developing three-dimensional FE simulations of the scale models of the current limiters. In the *long term* these notes will serve as aid in performing future work (FE simulations of full scale limiters, creating accurate simplified models, extending to complex networks, extend to size optimization, e.g.) and provide a basis for similar projects (computation of eddy current losses joint with H. Polinder e.g.)
- we would like to provide the Comsol Multiphysics support team and users community with feedback and examples on how we use the software as we expect that maintaining these contacts will be beneficial in future projects.
- we hope to take some of this material to our teaching, in particular to our first years calculus course (ordinary differential equations, vector fields, .e.g.) and our scientific computing course.

0.1 Notes to Dalibor

- size of the coils: in a FE model the size of the coils determines the size of the support of the source function. Therefore the size of the coils will influence the results
- distance between coils and core: I created a model that clearly shows the influence of the distance between the coils and the code
- can you create a model in Sabor of my single-period limiter?
- the three-leg configuration is thinner than the open-core configuration. For the former end-effects might become more importasnt.
- in the report comparing Ansys and Comsol in 2D, the parameter lz was set wrong. This was corrected in the new version of the report

Chapter 1

Introduction

In these notes we develop a sequence of numerical models of increasing accuracy and complexity for the current in coils winded around ferromagnetic cores. We start by detailing analytical and semi-analytical models, build two-dimensional finite element methods and extend these into three dimensions. Modelling results for the current wave forms are compared with laboratory measuments on scale models of the devices under study. Initially we were motivated by comparing different configurations of so-called inductive fault current limiters. The study of these devices was presented at the Comsol Multiphysics Users Conference 2008 in Hannover (+ reference). The configurations we study are however representative for a wide range of other applications in AC/DC modelling such as electrical machines, transformers and actuators. We therefore decided to document the solution to different difficulties we encountered in the modelling in Comsol Multiphysics, hoping that you (the reader) might learn from it.

1.1 Inductive Fault Current Limiters

Chapter 2

Quasi Stationary Magnetic Fields

- Maxwell equations
- double curl equations for the magnetic vector potential
- perpendicular current application mode
- stranded conductor model - winding function

Chapter 3

Inductance and Induced Voltage

In this section we review some preliminaries used in subsequent chapters.

Goals In this chapter we aim at

- computing the induced voltage, i.e., the time variation of the magnetic flux through a multi-turn coil wounded around the leg of a ferromagnetic core, excited by a sinusoidal current able to bring the core leg periodically in and out of saturation;
- relating this induced voltage with the time-variation of the impedance of the coil on one hand and the magnetic permeability of the core leg on the other;
- presenting a field-circuit coupled model allowing to compute the induced voltage for arbitrary coil-core configurations.

3.1 Inductance

Consider a multi-turn coil wounded around a leg of a non-linear ferromagnetic core as shown in Figure xxx. A sinusoidal current will induce a time-varying magnetic field and induced voltage in the core. If the amplitude of the current is sufficiently large, the working point of the core will move along the non-linear magnetic material characteristic and bring the core leg alternatively in and out of saturation. This implies that the magnetic permeability will periodically vary between high to low values, corresponding to desaturation and saturation state, respectively. As the coil impedance is proportional to the permeability, the former will vary accordingly. In this section we aim at quantifying this change in coil impedance using the magnetic energy and induced voltage.

3.1.1 Definition

- give definition of (self and mutual) inductance
- give units (Henry) and function of primary units

$$1\mathrm{H} = 1\mathrm{Wb/A} = 1\mathrm{T\,m^2/A} = 1\frac{\mathrm{V\,s}}{\mathrm{m^2}}\mathrm{m^2/A} = 1\frac{\mathrm{V}}{\mathrm{A}}\mathrm{s} = 1\Omega\,\mathrm{s} \tag{3.1}$$

- give range of value for applications (tens of miliHenry)

3.1.2 Model of a Solenoid

3.1.3 Magnetic Energy

Consider that the volume Ω encloses the core and ferromagnetic core and has a permeability $\mu = \mu(\mathbf{x}, t)$. The magnetic energy W_m in Ω due to the magnetic flux $\mathbf{B}$ and field $\mathbf{H}$ induced due a time-varying current is given by the volume integral

$$W_m(t) = \frac{1}{2}\int_\Omega \mathbf{B}\cdot\mathbf{H}\,d\Omega = \frac{1}{2}\int_\Omega \mu\,\mathbf{B}\cdot\mathbf{B}\,d\Omega\,. \tag{3.2}$$

In a 2D perpendicular current formulation on a computational domain with cross-section Ω_{xy} in the xy-plane and length ℓ_z in the z-direction (examples will be given in subsequent chapters), this formula simplifies to the surface integral

$$W_m(t) = \frac{\ell_z}{2} \int_{\Omega_{xy}} \big[B_x H_x + B_y H_y\big] \, d\Omega = \frac{\ell_z}{2} \int_{\Omega_{xy}} \mu \big[B_x B_x + B_y B_y\big] \, d\Omega \,. \tag{3.3}$$

In case that the magnetic field is generated by a total current $I_{tot}(t)$ flowing through a coil with N_t windings and current $I(t)$ per turn, i.e., $I_{tot}(t) = N_t \, I(t)$, the magnetic energy and the coil impedance $L(t)$ are related by

$$W_m(t) = \frac{1}{2} L(t) I^2(t) \Leftrightarrow L(t) = 2 \frac{W_m(t)}{I^2(t)} \,. \tag{3.4}$$

Not that the impedance scales

- *quadratically* with N_t;
- *linearly* with μ.

In case that μ is constant (no magnetic saturation), the impedance is current independent and can therefore be computed using any non-zero value for the current. In case that a sinusoidal current brings (a part of) the core alternatively in and out of saturation, the impedance can be computed for a particular working condition. In configurations in which different coils are present (AC and DC coil in the fault current limiter), the self-inductance of the coil can be computed by considering the current in the single coil only (setting the current in the other coils equal to zero).

3.1.4 Magnetic Flux

The magnetic flux $\psi(t)$ passing through an oriented surface S with outward normal $\mathbf{n}$ is given by the surface integral

$$\psi(t) = \int_S \mathbf{B} \cdot d\mathbf{S} = \int_S \mathbf{B} \cdot \mathbf{n} \, dS \,. \tag{3.5}$$

In applications of this expression we are typically interested in, the surface S typically denotes the cross-section of the ferromagnetic core perpendicular to the flux path. In a 2D perpendicular current formulation on the domain Ω_{xy}, the surface S is then a line piece perpendicular to the y-axis extending from $x = x_m$ to $x = x_M$, extruded by a length ℓ_z in the z-direction. Using the vector potential $\mathbf{A}$ ($\mathbf{B} = \nabla \times \mathbf{A}$) as unknown, the above expression reduces to

$$\begin{aligned}
\phi(t) &= \int_{S_{core}} \mathbf{B} \cdot \mathbf{n} \, dS && (3.6)\\
&= \int_{S_{core}} B_y \, dx \, dz && (3.7)\\
&= -\int_{x_m}^{x_M} dx \int_{z_m}^{z_M} dz \frac{\partial A_z}{\partial x} \qquad [B_y = -\frac{\partial A_z}{\partial x}] && (3.8)\\
&= -\ell_z \int_{x_m}^{x_M} dx \frac{\partial A_z}{\partial x} && (3.9)\\
&= -\ell_z \left[A_z(x = x_M, t) - A_z(x = x_m, t)\right] && (3.10)\\
&= \ell_z \left[A_z(x = x_m, t) - A_z(x = x_M, t)\right] \,. && (3.11)
\end{aligned}$$

Considering the coil to be an interconnection of N_t flux contributions, the magnetic flux and the coil impedance are related by

$$\psi(t) = N_t \, \phi(t) = L(t) \, I(t) \Leftrightarrow L(t) = \frac{N_t \, \psi(t)}{I(t)} = \frac{\phi(t)}{I(t)} \,. \tag{3.12}$$

Assuming that $I(t) \neq 0$, this expression allow to compute the impedance via the magnetic flux.

3.2 Impedance

The Ohmic resistance of the wire and the inductance of a coil can be combined to form the total impedance denoted by X and defined by

$$X = \sqrt{R^2 + \omega^2 L^2} \,. \tag{3.13}$$

3.3 Induced Voltage

The voltage induced the time-varrying magnetic flux is given by

$$V_{ind} = -N_t \frac{d\psi}{dt} \,, \tag{3.14}$$

where the minus sign is due to Lenz's Law. Using the flux-impedance relation (3.12) the above relation can be written as

$$V_{ind} = -\frac{d}{dt}(LI) \,. \tag{3.15}$$

In case that $\frac{dL}{dt} = 0$, this formula reduces to

$$V_{ind} = -L \frac{d}{dt}(I) \,. \tag{3.16}$$

meaning that the induced voltage is large (small) when the core leg is out (in) of saturation. In models where leakage or fringing flux appears, it is not a-priori clear how to choose the integration surface S leading to a correct expression for the induced voltage. This issue is resolved in the next section by integration the induced voltage density over the cross-section of the conductor instead.

3.4 Stranded Conductor

In this section we develop a model allowing to compute the induced voltage by integrating the density over the cross-section of the conductor. We need to describe the 3D model from which the 2D can be derived and observe that the coil fill factor drops out in the description of the induced voltage.

3.4.1 Series Connection of Two Stranded Conductors

The induced voltage in the AC coil (with cross-section $S_{ac,1}$ and $S_{ac,2}$ on both siddes of the core) has two contributions and can be computed as follows

$$\begin{aligned}
V_{ind} &= V_{ind,1} + V_{ind,2} && (3.17)\\
&= -\text{sign}(J_z) \frac{N_{t,ac}\,\ell_z}{S_{ac,1}} \int_{S_{ac,1}} E_z \, dS - \text{sign}(J_z) \frac{N_{t,ac}\,\ell_z}{S_{ac,2}} \int_{S_{ac,2}} E_z \, dS && (3.18)\\
&= -\text{sign}(J_z) \frac{N_{t,ac}\,\ell_z}{S_{ac,1}} \int_{S_{ac,1}} \frac{\partial A_z}{\partial t} \, dS - \text{sign}(J_z) \frac{N_{t,ac}\,\ell_z}{S_{ac,2}} \int_{S_{ac,2}} \frac{\partial A_z}{\partial t} \, dS && (3.19)
\end{aligned}$$

where the minus sign stems from the fact that the current flows in opposite directions in both sides of the coil. An alternative is to compute the induced voltage as

$$\begin{aligned}
V_{ind} &= -N_t \frac{d\,\phi^{tot}}{dt} && (3.20)\\
&= -N_t \frac{d\,(\phi^L + \phi^R)}{dt} \quad \text{if} \quad \frac{d\,\phi^{air}}{dt} \text{ is small.} && (3.21)
\end{aligned}$$

$$\boxed{\frac{d}{dt}(LI) = \frac{N_t\,\ell_z}{S_{coil}} \int_{S_{coil}} E_z \, dS} \tag{3.22}$$

3.5 Magnetic Field - Electrical Circuit Coupled Model

Able to compute the current limiting effect, we develop a field-circuit coupled model.

Chapter 4

Magnetic Saturation

In this section we describe the modeling of magnetic saturation in ferromagnetic materials, i.e., the modeling of the non-linear constitute relation between the magnetic flux $\mathbf{B}$ (units T) and the magnetic field $\mathbf{H}$ (units A/m). Using a vector potential formulation and denoting by $B = \|\mathbf{B}\|$, $H = \|\mathbf{H}\|$, the magnetic material law typically considered is

$$H(B) = \nu\, B = \nu_0\, \nu_r(B)\, B \Leftrightarrow \nu(B) = \frac{dH}{dB}(B)\,, \tag{4.1}$$

where ν_0 and ν (ν_r) denote the reluctivity of vacuum and the material (relative reluctivity), respectively [see paper Herbert on differential vs. chord reluctivity]. The inverse of the reluctivity is the permeability μ. Magnetic saturation is such that $\mu_r(B)$ is large and almost constant for small values of B and small almost constant for large values of B and has a non-linear transition between these two extreme values (see for instance Figure xxx).

In practise engineering practise, the function $\nu_r(B)$ is to be constructed from measured B-H samples. This process makes the convergence of an FEM computation prone stagnation. We therefore consider analytical expressions allowing to describe the function $\nu_r(B)$ analytically.

Goals In this chapter we aim at

- giving different analytical expressions for the non-linear B-H -curve modelling magnetic saturation
- give an example of a measured B-H
- illustrate a least square curve fitting technique allow to match the analytical expressions to the given measured data

To do

- add reference to rational BH-curve approximation
- add reference to Pechstein on approximating the BH-curve

4.1 Analytical Models

To do:

- make all plots of the BH-curves again
- compute the second derivative of the analytical model to see where the curvature changes from positive to negative.
- give a plot on double axis and deduce for which value of the current the core is in saturation.

4.1.1 Rational Function Approximation

In [1] the following rational expression modelling the relative reluctivity is given (denoting by $B = \|\mathbf{B}\|$)

$$\nu_r = a + \frac{(1-a)B^{2b}}{B^{2b}+c} \Leftrightarrow \mu_r = \frac{1}{a + \frac{(1-a)B^{2b}}{B^{2b}+c}} \tag{4.2}$$

where the values for the parameters a, b and c are given in Table 4.1. For this models holds that

$$\nu_r(B=0) = a \tag{4.3}$$

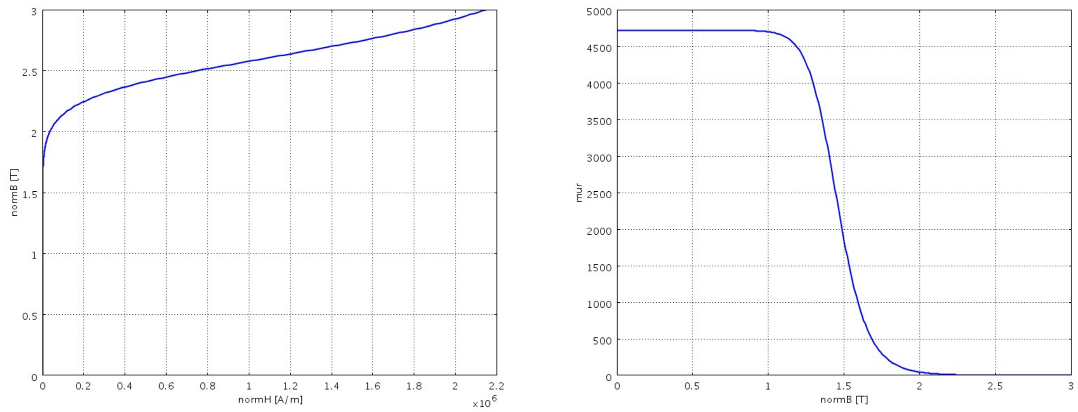
and thus the relative permeability at $B = 0$ is given by $1/a$ and that

$$\lim_{B\to\infty} \nu_r(B) = 1 \tag{4.4}$$

and thus the permeability never becomes smaller than μ_0 (which is physically correct).

a	2.12e-4
b	7.358
c	1.18e6

Table 4.1: Constants Used in the Rational Approximation



(a) B-H-curve

(b) μ_r-B curve

Figure 4.1: Rational Function Approximation of the B-H curve.

4.1.2 Hyperbolic Function Approximation

In [2] the following approximation is given:

$$B = C_1 \operatorname{arcsinh}(C_2 H) \Leftrightarrow H = \frac{1}{C_2}\sinh(\frac{B}{C_1}) \tag{4.5}$$

where the values for the parameters C_1 and C_2 are given in Table (4.2). From this we obtain the chord permeability

$$\mu = \frac{B}{H} = \frac{C_2 B}{\sinh(\frac{B}{C_1})} \tag{4.6}$$

$$\mu_r = \frac{\mu}{\mu_0} = \frac{C_2 B}{\mu_0 \sinh(\frac{B}{C_1})} \tag{4.7}$$

and the differential permeability

$$\nu = \frac{dH}{dB} = \frac{\cosh(C_2 B)}{C_1 C_2} \tag{4.8}$$

$$\mu = \frac{1}{\nu} = \frac{C_1 C_2}{\cosh(C_2 B)} \tag{4.9}$$

$$\mu_r = \frac{\mu}{\mu_0} = \frac{C_1 C_2}{\mu_0 \cosh(C_2 B)} \tag{4.10}$$

$$\tag{4.11}$$

As in this model

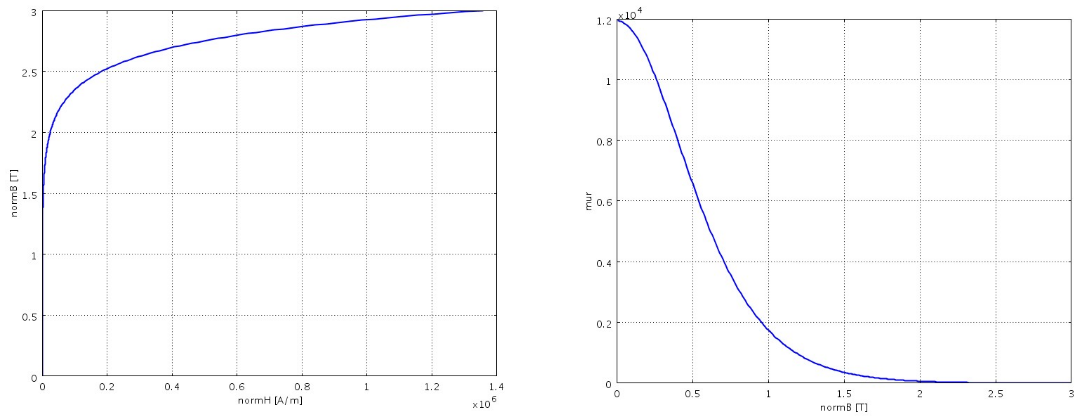
$$\lim_{B2 \to \infty} \mu_r(B2) = 0\,, \tag{4.12}$$

is has to be used with due care.

Remark This model requires a non-zero initial guess during intial guess to avoid the singularity at $B = 0$.

C_1	.25 [T]
C_2	.06 [m/A]

Table 4.2: Constants Used in the Sinh Approximation



(a) B-H-curve

(b) μ_r-B curve

Figure 4.2: Hyperbolic Function Approximation of a B-H curve.

4.2 Measured Data

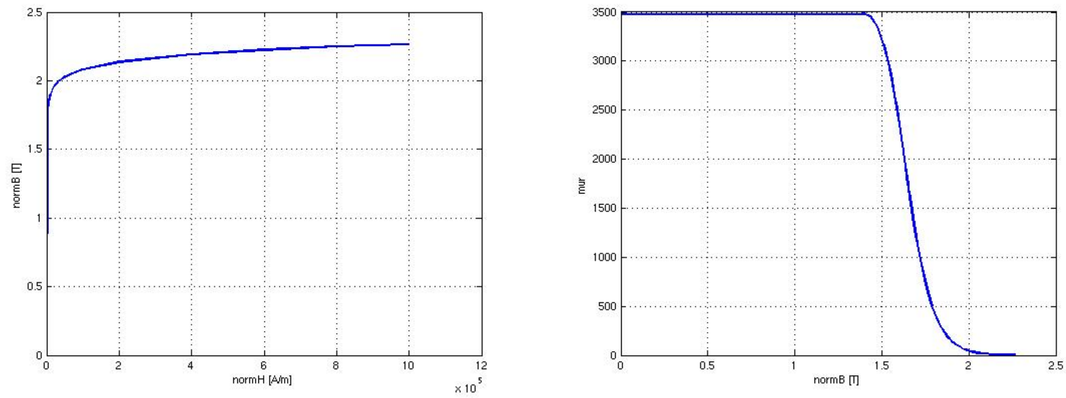
In this section we give the measured B-H-data we will use in our numerical examples in subsequent chapters.

4.2.1 Measured BH data

```
H = [0 310      315     320     330     350     380     410     430 ...
470     500     540     580     620     650     670     720     750 ...
770     820     900     1000    1100    1200    1400    1800    2300 ...
2800    3300    4300    5300    8300    10300   15300   20300   25300 ...
30300   40300   50300  70300  100300  200300  400300  600300  800300 ...
1000300];

B = [0 1.3449   1.3773  1.4003  1.4328  1.4736  1.5112  1.5367  1.5501 ...
1.5715  1.5845  1.5991  1.6114  1.6221  1.6293  1.6337  1.6439  1.6494 ...
1.6529  1.661          1.6724  1.6847  1.6954  1.7048  1.7209  1.7457 ...
```

```
1.7687  1.7866  1.8012  1.8242  1.842   1.8796  1.8975  1.9299  1.9529 ...
1.9708  1.9854  2.0084  2.0262  2.0532  2.0817  2.1371  2.1926  2.225  ...
       2.248   2.2659 ];
```



(a) B-H-curve (b) μ_r-B curve

Figure 4.3: B-H-curve used in tabular form.

4.3 Tuning the Analyical Models

In this section we performs a fitting of the parameters in the rational and hyperbolic approximation to the measured data by a non-linear optimisation procedure.

Chapter 5

Lumped Parameter Models of RL Circuits

Prior to detailing finite element models in subsequent chapters, we develop in this chapter simplified lumped parameter approximation that serve to get an intuitive feeling and point of comparison for the subsequent models. The point of departure for deriving lumped parameter models is the following first order ordinary differential equation relating the voltage excitation $V(t)$ with the current $I(t)$ in a circuit with an Ohmic resistance R and inductance L

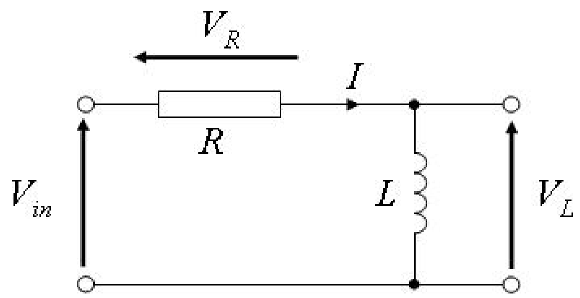


Figure 5.1: Series connection of a resistance and inductance.

$$V_R(t) + V_L(t) = V_{in}(t) \Leftrightarrow \frac{d}{dt}(L\,I) + R\,I = V_{in}(t)\,. \tag{5.1}$$

The terms in the left-hand side can be identified as the induced and resistive voltage, respectively, and the equation states that at all times the sum of the induced and resistive voltage is equal to the externally applied one. The equation needs to be supplied with an initial value for the current.

The Ohmic resistance R determined by the electrical conductivity of the medium. In the case that the coil is solenoid wounded N_t times around a ferromagnetic core with magnetic permeability μ, the impedance L can be expressed as

$$L = \mu\,N_t^2\,\frac{S}{l_{path}} = \mu_0\,\mu_r\,N_t^2\,\frac{S}{l_{path}}\,, \tag{5.2}$$

where S and l_{path} cross-section and length of the flux path in the ferromagnetic core respectively.

In this chapter we first derive an analytical expression for the current in an RL-circuit with *constant* impedance and two excitations: a constant and sinusoidally varying voltage source. These models illustrate how the presence of an impedance causes a phase shift in and amplitude reduction of the current. In more realistic models however the magnetic permeability μ of the core changes by moving the operation point on a non-linear B-H characteristic. In a second stage we therefore extend the model to include *changes* in the impedance induced by changes in the magnetic permeability. This model assumes the coil to be a solenoid for which expression (5.2) is correct. The generalisation of this model to more complex coil-core configurations is therefore not immediate.

Describe a mechanical equivalent: variable impedance and variable mass.

Goals In this chapter we aim at

- describing a simple model able to explain the inductive current limiting principle (including the concepts of induced voltage and phase difference between applied voltage and current). This simple model can possibly serve as coarse model inside an surrogate based optimisation algorithm.

- explaining why this simple model is not sufficient for the type of devices considered

To do:

1. describe drop in resistive voltage
2. describe presence of DC coil by additive constant in the flux

5.1 Constant Impedance Model

In this section we assume that $\frac{dL}{dt} = 0$. In this case, equation (5.1) reduces to

$$L\frac{dI}{dt} + R\,I = V(t)\,. \tag{5.3}$$

Given some initial condition, this ordinary differential equation can be solved numerically using a time-integrator taking a time-dependent resistance (simulation of fault) into account. In order to derive analytical expressions however, we assume from here on that $\frac{dR}{dt} = 0$. The ratio $\frac{R}{L}$ has the dimensions of an (angular) pulsation and will be denoted by ω_1 from here on. The solution of the homogeneous equation to (5.3) is

$$I_h(t) = C\exp(-R/L\,t) = C\exp(-\omega_1 t) \tag{5.4}$$

where C is chosen so to satisfy the initial conditions.

5.1.1 Constant Applied Voltage

In case that the applied voltage is constant and equal to V_0, the method of variation of constants yields that $I(t) = C(t)\,I_h(t)$, where

$$C'(t) = V_0/L\exp(\omega_1 t) \Rightarrow C(t) = V_0/L\,\omega_1\exp(\omega_1 t) + C_0 = V_0/R\exp(\omega_1 t) + C_0\,. \tag{5.5}$$

For the current we then have

$$\begin{aligned} I(t) &= \left[V_0/R\exp(\omega_1 t) + C_0\right]\,\exp(-\omega_1 t) && (5.6)\\ &= V_0/R + C_0\exp(-\omega_1 t)\,, && (5.7)\end{aligned}$$

where the integration constant C_0 is related to the initial condition $I(t=0) = I_0$ by

$$C_0 = I_0 - V_0/R\,. \tag{5.8}$$

The current is then given by

$$I(t) = \frac{V_0}{R} + \left[I_0 - V_0/R\right]\exp(-\omega_1 t)\,. \tag{5.9}$$

If the relaxation time $\tau = L/R$ is sufficiently small (i.e., if the resistance R is not too small and the inductance L is not too large), then after a few multiples of the relaxation time, the current is independent of the initial condition and equal to its stationary value

$$\boxed{I(t) = \frac{V_0}{R}\,.} \tag{5.10}$$

5.1.2 Sinusoidally Varrying Applied Voltage

In case that the applied is sinusoidally varying, i.e., $V(t) = V_0\sin(\omega_0 t)$, the method of variation of constants yields that $I(t) = C(t)\,I_h(t)$, where

$$C'(t) = V_0/L\sin(\omega_0 t)\exp(\omega_1 t) \Rightarrow C(t) = (V_0/L)\int^t \sin(\omega_0 s)\exp(\omega_1 s)\,ds + C_0\,. \tag{5.11}$$

for some integration constant C_0. Applying integration by parts twice yields

$$
\begin{aligned}
L/V_0\, C(t) &= \big[1/\omega_1 \sin(\omega_0 s) \exp(\omega_1 s)\big]^t - \omega_0/\omega_1 \int^t \cos(\omega_0 t) \exp(\omega_1 s)\, ds + C_0 && (5.12)\\
&= 1/\omega_1 \sin(\omega_0 t) \exp(\omega_1 t) - \omega_0/\omega_1^2 \big[\cos(\omega_0 s) \exp(\omega_1 s)\big]^t && (5.13)\\
&\quad -\omega_0^2/\omega_1^2 \int^t \sin(\omega_0 s) \exp(\omega_1 s)\, ds + C_0 && (5.14)\\
&= 1/\omega_1 \sin(\omega_0 t) \exp(\omega_1 t) - \omega_0/\omega_1^2 \cos(\omega_0 t) \exp(\omega_1 t) && (5.15)\\
&\quad -\omega_0^2/\omega_1^2 \int^t \sin(\omega_0 s) \exp(\omega_1 s)\, ds + C_0 && (5.16)
\end{aligned}
$$

Hence

$$
[1 + \omega_0^2/\omega_1^2]\, C(t) = 1/\omega_1 (V_0/L) \sin(\omega_0 t) \exp(\omega_1 t) - \omega_0/\omega_1^2 (V_0/L) \cos(\omega_0 t) \exp(\omega_1 t) + C_0 \qquad (5.17)
$$

or

$$
\begin{aligned}
C(t) &= (V_0/L)\big[\frac{\omega_1}{\omega_0^2+\omega_1^2} \sin(\omega_0 t) - \frac{\omega_0}{\omega_0^2+\omega_1^2} \cos(\omega_0 t)\big] \exp(\omega_1 t) + C_0 && (5.18)\\
&= \frac{V_0}{L(\omega_0^2+\omega_1^2)} \big[\omega_1 \sin(\omega_0 t) - \omega_0 \cos(\omega_0 t)\big] \exp(\omega_1 t) + C_0 && (5.19)\\
&= \frac{V_0\sqrt{\omega_0^2+\omega_1^2}}{L(\omega_0^2+\omega_1^2)} \big[\frac{\omega_1}{\sqrt{\omega_0^2+\omega_1^2}} \sin(\omega_0 t) - \frac{\omega_0}{\sqrt{\omega_0^2+\omega_1^2}} \cos(\omega_0 t)\big] \exp(\omega_1 t) + C_0 && (5.20)\\
&= \frac{V_0}{L\sqrt{\omega_0^2+\omega_1^2}} \big[\sin(\omega_0 t)\cos\Theta - \cos(\omega_0 t)\sin\Theta\big] \exp(\omega_1 t) + C_0 && (5.21)\\
&= \frac{V_0}{X} \sin(\omega_0 t - \Theta) \exp(\omega_1 t) + C_0 && (5.22)
\end{aligned}
$$

where we've introduced the phase shift

$$
\frac{\sin\Theta}{\cos\Theta} = \frac{\omega_0}{\omega_1} \Leftrightarrow \Theta = \arctan(\frac{\omega_0}{\omega_1}) = \arctan(\frac{2\pi f R}{L})\,, \qquad (5.23)
$$

and where the integration constant C_0 is related to the initial condition by

$$
C_0 = I_0 - \frac{V_0}{X} \sin(\Theta)\,. \qquad (5.24)
$$

The current is then given by

$$
I(t) = \frac{V_0}{X} \sin(\omega_0 t - \Theta) + \big[I_0 - \frac{V_0}{X} \sin\Theta\big] \exp(-\omega_1 t)\,. \qquad (5.25)
$$

If the relaxation time $\tau = 1/\omega_1 = L/R$ is suffuciently large, then after a few multiples of the relaxation time, the current is independent of the initial condition and equal to

$$
\boxed{I(t) = \frac{V_0}{X} \sin(\omega_0 t - \Theta)\,.} \qquad (5.26)
$$

Compared with a purely resistive network, the current has both a lower amplitude and a phase shift. A large inductance in particular will lead to a lower current value and a larger phase shift. This is illustrated in Figure 5.1.2.

5.2 Flux-Variable Impedance Model

The models developed in the previous section cease to be valid in situations in which the working point changes in time over a range in which the permeability and therefore the impedance can no longer assumed to be constant. The case that we will be interested in is the one in which the permeability varries along a non-linear B-H characteristic and which time-varrying voltage source bringing the ferromagnetic core in (low permeability and impedance) and out (high permeability and impedance) of saturation.

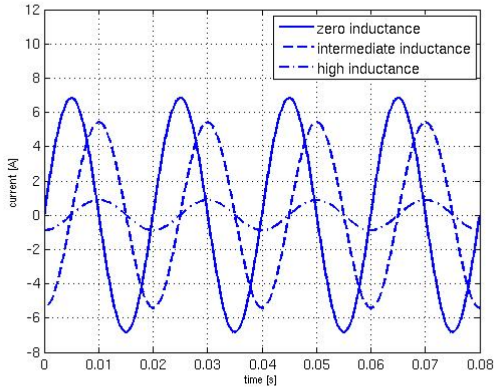


Figure 5.2: Current in RL-circuit for different values of the inductance.

To extend our models to variable impedance cases, it turns out to be convenient to replace the current by the flux as state variable

$$\psi = L\,I \Leftrightarrow I = \frac{\psi}{L(\psi)} \tag{5.27}$$

and to rewrite the equation (5.1) modelling an RL-circuit as

$$\frac{d}{dt}\psi + \frac{R}{L(\psi)}\psi = V(t)\,. \tag{5.28}$$

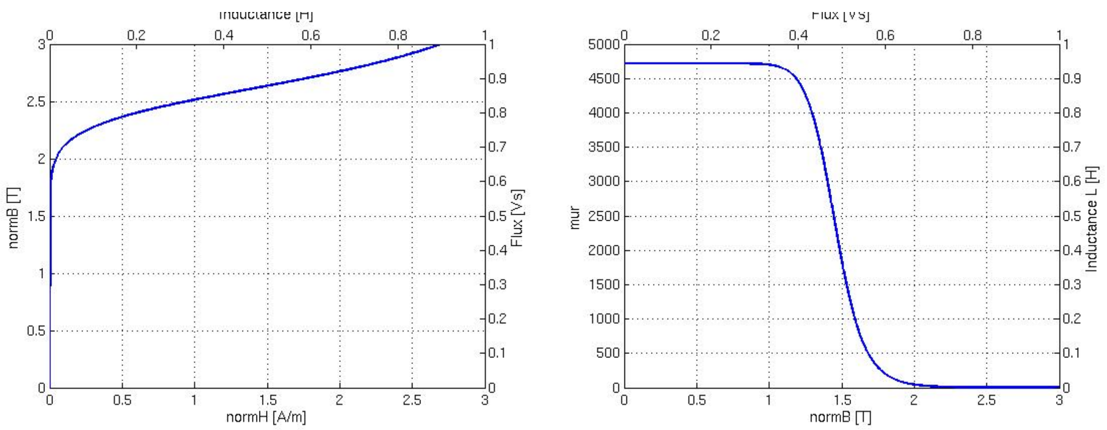
In this model the variable impedance can be computed using the non-linear characteristic data assuming the model (5.2) for a solenoid. In this case we have that

$$L(\psi) = \mu\big[B(\psi)\big]N_t^2\frac{S}{l_{path}} \tag{5.29}$$

$$= \mu\big[\frac{\psi}{S}\big]N_t^2\frac{S}{l_{path}}\,. \tag{5.30}$$

Given an inital condition $\psi(t=0) = L(t=0)I(t=0)$, the ODE (5.28) can be solved numerical for the flux ψ and thus also for current I using (5.27).

This model extends the models of the previous section to a variable impedance and thus allow to illustrate the inductive fault current limiting effect. This model still has limited applicability as it uses the expression for the solenoid.



(a) B-H-curve

(b) μ_r-B curve

Figure 5.3: Rational Function Approximation of the B-H curve.

5.3 Numerical Example

In this section we employ the model developed in the previous section to illustrate how a coil with a flux-variable impedance can work as a fault current limiter.

```
function [T,psi] = solve_rlcircuit(my_ctx)

  %..Compute initial condition taking phase shift into account..
  Rpre   = 5; Rpost = .1; R_init = Rpre+Rpost; Tfault = 1; Vmax = 28;
  L_init  = lookup_imped(0,my_ctx);
  om      = 2*pi*50;
  X_init  = sqrt(R_init^2 + om^2*L_init^2);
  phase_shift = atan(om*R_init/L_init);
  I_init   = - Vmax/X_init*sin(phase_shift);
  psi_init =  L_init*I_init;

  %..Add DC component..
  V_dc = -.4;
  psi_dc = L_init*V_dc/R_init;

  %..Solve ODE for the magnetic flux..
  Tend    = 0.32;
  options = odeset('RelTol',1e-12,'AbsTol',1e-12);
%  options = [];
  [T,psi] = ode45(@rlcircuit,[0 Tend],psi_init+psi_dc,options);

  function dpsidt = rlcircuit(t,psi)

    if (t<Tfault) Rline = Rpre+Rpost; else Rline = Rpost; end
    Vline  = V_dc + Vmax*sin(2*pi*50*t);
    Limpd  = lookup_imped(psi,my_ctx);
    dpsidt = Vline - Rline/Limpd*psi;

    end

  end
```

```
function Limpd = lookup_imped(psi, my_ctx);

  global Limpd

  btopsi = my_ctx.btopsi;
  murtoL = my_ctx.murtoL;
  psi_dc = my_ctx.psi_dc;

  %..Find b..
  b = psi/btopsi;

  %..Find relative permeability..
  mur = bhcurve(b);

  %..Find inductance..
  Limpd = murtoL*mur;

  %..Overwrite with an a-priori value
  if (0)
    Limpd = 1e-1;
  end
```

```
function mur = bhcurve(b)

 %..BH curve definition..
 bha = 2.12e-4;
 bhb = 7.358;
 bhc = 1.18e6;

 %..Define mur-b2 curve
 b2 = b.*b;
 nur = bha + (1-bha)*b2.^bhb./(b2.^bhb+bhc);
 mur = 1./nur;

 %..If desired, overwrite with linear material..
 if (0)
   mur = ones(size(b));
 end
```

Chapter 6

Two Dimensional FE Models of Inductive Fault Current Limiters

6.1 TO DO

1. in the first model
   - choice of surface S motivated by desire to minimize influence of fringing and leakage flux
   - investigate influence influence of space between coil and core
   - investige influence of gap in the core
   - describe three stage process in solving the model
   - include impedance computation after the second stage
   - describe the setting of the initial guess in assembling the first jacobian (different from jacobian specified in femsolver!)
2. additional model
   - open core model
   - three legs model

Goals

- simulate RL-circuit without having to resort analytical model for the impedance
- compute the impedance of a given configuration in three different ways, using the analytical formula, the magnetic energy and the magnetic flux
- compute the current waveform using only the AC coil, the DC coil in two different polarities and with linear and non-linear core
- check the magnetic flux density in the core legs and verify using the BH-curve to what extend the DC coils brings the legs in saturation
- investigate to what extend an ODE model allows to simulate this configuration, eventually by first computing the impedance
- investigate to what extend the geometry of the coils affects the current waveforms
- document issues on time integration

6.2 Half Period Limiter

6.2.1 Geometry

In defining the geometry the core acts master and the coil as slaves. This corresponds to the fact that the coils are wounded around the core.

1. **the core**:

   (a) core variables

   ```
   crwin = 12.5e-3; crwout =  37.5e-3; crwleg = crwout - crwin;
   crhin = 39e-3; crhout = 63e-3;
   rad = 3e-3;
   ```

   (b) core:

   ```
   core = fillet(rect2(-crwout,crwout,-crhout, crhout), 'radii', rad) ...
                        - rect2(crwin,crwin,-crhin,crhin);
   ```

2. **flux integration lines**:

   (a) **left core leg**: line from (-crwout,0) to (-crwin,0)

   ```
   fluxline1 = line1([-crwout,-crwin],[0,0]);
   ```

   (b) **right core leg**: line from (crwin,0) to (crwout,0)

   ```
   fluxline2 = line1([crwin,crwout],[0,0]);;
   ```

3. **the AC coil**:

   (a) AC coil variables

   ```
   accoilw = 10e-3; accoilh = 20e-3;
   xspacer =  2e-3
   accoilradin = crlegw/2+xspacer; accoilradout = accoilradin+accoilw;
   coilxc = crwin+crwleg/2;
   ```

   (b) AC coil:

   ```
   accoil_right = rect2(accoilradin,accoilradout,-accoilh/2,accoilh/2);
   accoil       = accoil_right+move(accoil_right,-2*accoilradin-accoilw,0);
   accoil       = move(accoil,accoilxc,0);
   ```

4. **the DC coil**:

   (a) DC coil variables

   ```
   dccoilh     = 10e-3; dccoilw = crwin/2;
   dccoilradin = (crhout-crhin)/2; dccoilradout = dccoilradin + dccoilh;
   dccoilyc    = crhin+(crhout-crhin)/2;
   ```

   (b) DC coil

   ```
   dccoil_top  = rect2(-dccoilw/2,dccoilw/2,dccoilradin,dccoilradout);
   dccoil      = dccoil_top + move(dccoil_top,0,-2*dccoilradin-dccoilh);
   dccoil      = move(dccoil,0,dccoilyc);
   ```

5. **the air**:

   (a) air variables

   ```
   airh = 400e-3; airw = 400e-3
   ```

   (b) `air = rect2(-airh, airh, -airw, airw);`

The 1D and 2D entities in the geometry are then combined using

```
clear c s
c.objs={fluxline1,fluxline2};
c.name={'fluxline1','fluxline2'};
c.tags={'g5','g6'};

s.objs={air,accoil,dccoil,core};
s.name={'air1','accoil','dccoil','core'};
s.tags={'g1','g2','g3','g4'};

fem.draw=struct('c',c,'s',s);
fem.geom=geomcsg(fem);
```

6.2.2 Constants, Functions and Subdomain and Global Expressions

Constants

Note that a fill-factor in the coils in **not** used as the expression for the induced voltage in scaling invariant for the induced voltage.

Electrical constants	
ω	$2\ \pi\ 50$
Vmax	28
Rpre	4.0
Rpost	0.1
Tfault	45e-3
Tsmooth	1e-3
Core	
lz (Length in z-direction)	25e-3
flcrwleg	crwleg
crcross (Core leg cross-section)	flcrwleg*lz
AC coil	
acNt (Number of turns)	200
accross (Cross-section)	accoilw*accoilh
Iac (Current value)	5
DC coil	
dcNt (Number of turns)	250
dccross (Cross-section)	dccoilw*dccoilh
Idc (Current value)	10
BH curve data	
linmurfe	1000
bha	2.12e-4
bhb	7.358
bhc	1.18e6
C_1	.25
C_2	.06

Table 6.1: Constants Used

Functions Functions are used to define the different BH-curves.

Global expressions The flux variables and their derivatives are included to check the model. They are not used in the computation as such. Flux variables are used to verify whether or not the core legs are in saturation. The flux derivative variables are used to check the induced voltage.

$$
\begin{aligned}
Vline &= Vmax \sin(\omega t) && (6.1)\\
Rline &= Rpre - (Rpre - Rpost) * flc2hs(t - Tfault, Tsmooth) && (6.2)\\
Vind1 &= -\text{sign}(Jz) * acNt * lz/accross * Eint1 && (6.3)\\
Vind2 &= -\text{sign}(Jz) * acNt * lz/accross * Eint2 && (6.4)\\
Vind &= Vind1 + Vind2 && (6.5)\\
Bavrg1 &= Bint1/crwleg && (6.6)\\
Bavrg2 &= Bint2/crwleg && (6.7)\\
fluxt1 &= -acNt * lz * Btint1 && (6.8)\\
fluxt2 &= -acNt * lz * Btint2 && (6.9)
\end{aligned}
$$

6.2.3 Integration Coupling Variables

We keep two variables for the induced voltage in order to be able to monitor them seperately. Subdomain integration coupling variables for the induced voltage

$$
\begin{aligned}
Eint1 &= \int_{accoil1} E_z \, d\Omega = \int_{accoil1} \frac{\partial A_z}{\partial t} \, d\Omega && (6.10)\\
Eint2 &= \int_{accoil2} E_z \, d\Omega = \int_{accoil2} \frac{\partial A_z}{\partial t} \, d\Omega && (6.11)
\end{aligned}
$$

and boundary integration coupling variables for the average flux and the time-derivative of the flux

$$
\begin{aligned}
Bint1 &= \int_{fluxline1} B_y \, dl = \int_{fluxline1} -\frac{\partial A_z}{\partial x} \, dl && (6.12)\\
Bint2 &= \int_{fluxline2} B_y \, dl = \int_{fluxline2} -\frac{\partial A_z}{\partial x} \, dl && (6.13)\\
Btint1 &= \frac{d}{dt} \int_{fluxline1} B_y \, dl = \int_{fluxline1} -\frac{\partial^2 A_z}{\partial x \, \partial t} \, dl && (6.14)\\
Btint2 &= \frac{d}{dt} \int_{fluxline2} B_y \, dl = \int_{fluxline2} -\frac{\partial^2 A_z}{\partial x \, \partial t} \, dl && (6.15)
\end{aligned}
$$

where the minus sign in the expressions with index two take the direction of the current into account.

6.2.4 Application mode, subdomain and boundary settings

The magnetic field is modeled by a partial differential equation for the z-component of the magnetic vector potential A_z (perpendicular current model) satisfying the following equation

$$
\sigma \frac{\partial A_z}{\partial t} + \frac{\partial}{\partial x}\big(\nu_0 \, \nu_r(B) \frac{\partial A_z}{\partial x}\big) + \frac{\partial}{\partial y}\big(\nu_0 \, \nu_r(B) \frac{\partial A_z}{\partial y}\big) = J_z(x, y, t) \qquad (6.16)
$$

where $\sigma = 0$ everywhere on Ω, supplied with boundary conditions. All time-dependency is thus in the current source! We do solve this equation with a time-stepping procedure as we need the derivative $\frac{\partial A_z}{\partial t}$ in defining the induced voltage.

- material characteristics:
  - ferromagnetic core: μ_r through BH-curve
  - coils and air: $\mu_r = 1$
- current exitation
  - DC coil: constant current density equal to $J_{z,dc} = \pm \frac{I_{dc,tot}}{S_{dc}} = \pm dcNt \frac{Idc}{dccross}$
  - AC coil: voltage driven by a sinusoidal voltage source $Vline$ through the circuit relation given below. The variable $Itot$ is to be computed such that $J_{z,ac} = \pm \frac{I_{ac,tot}}{accross} = \pm acNt \frac{Itot}{accross}$

Different Application Modes In different application modes we subsequently solve for

- the impedance using three different models
- the initial guess for the transient simulation
- the non-linear transient simulation including the fault

6.2.5 ODE Settings

The current in the AC coil is modeled by a crircuit relation (an ODE) for the variable I_{tot}

$$\begin{aligned} V_{tot} &= V_{res} + V_{ind} && (6.17) \\ &= RI_{tot} + Vind1 + Vind2 && (6.18) \end{aligned}$$

6.2.6 Numerical Results

Discuss that

- this configuration is not able to limit the during during both periods.

6.3 Open-Core Configuration

1. geometry of the AC coil changes

6.4 Three-Leg Configuration

1. new geometry
2. DC coils opposite polarity, AC coils different polarity
3. in the computation of the inductance only *only* AC coil needs to be taken into account
4. new integration coupling variables
5. new ODE setting

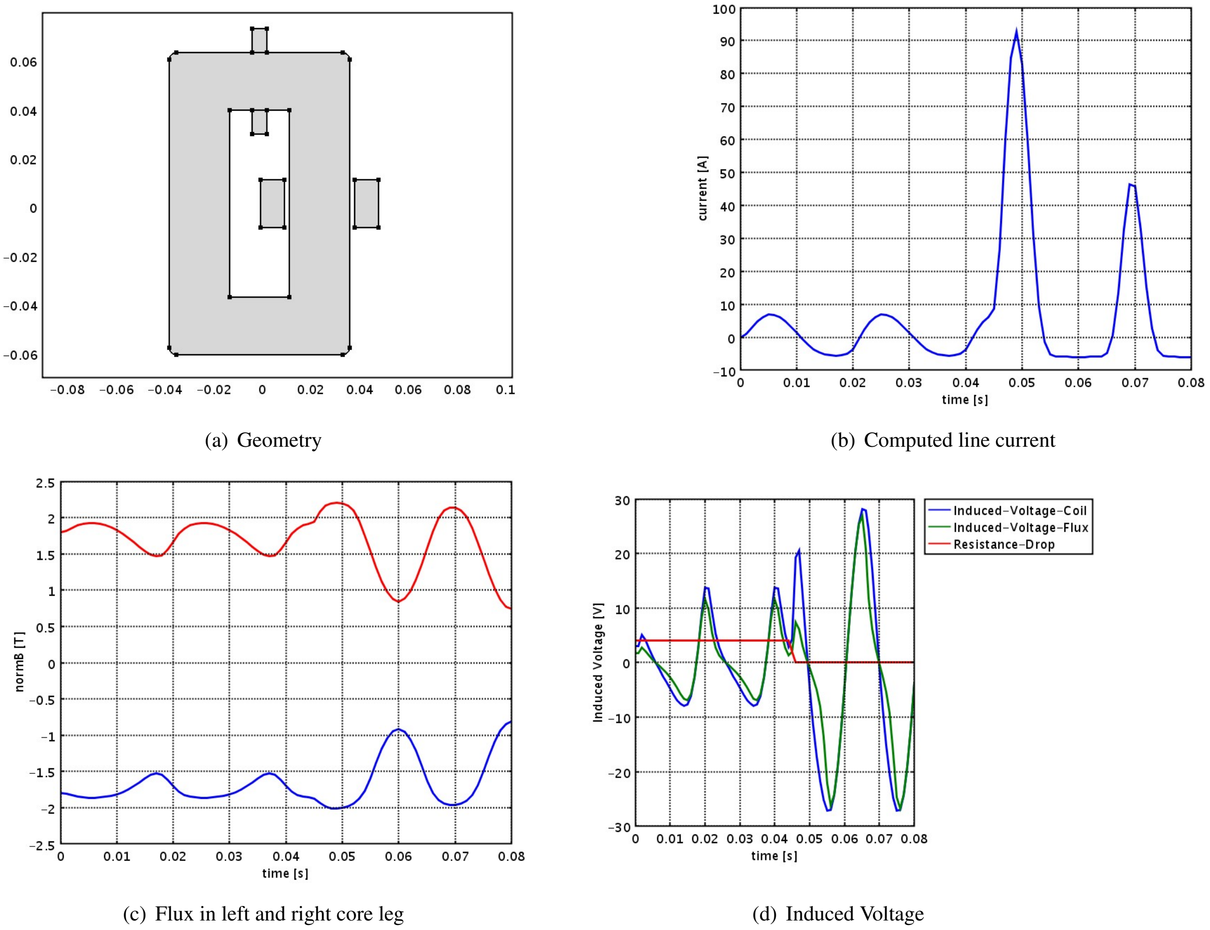
6.4.1 Source file of the open core model

```
% Model of two coils with pierced core
% The german BH curve requires a DC current away from zero in order
% to converge
  flclear fem fem0
  close all

 %..Set switches..
  bhswtch = 'hyperchord';

 %..Number of turns and current values
 %..The current in the AC coil is set to compute the inductance of the coil
  Idc  = 10;
  dcNt = 250;
  Iac  = 5;
  acNt = 200;

 %..Create geometery..
 %....Core....
  crwin  = 12.5e-3;
  crwout = 37.5e-3;
```



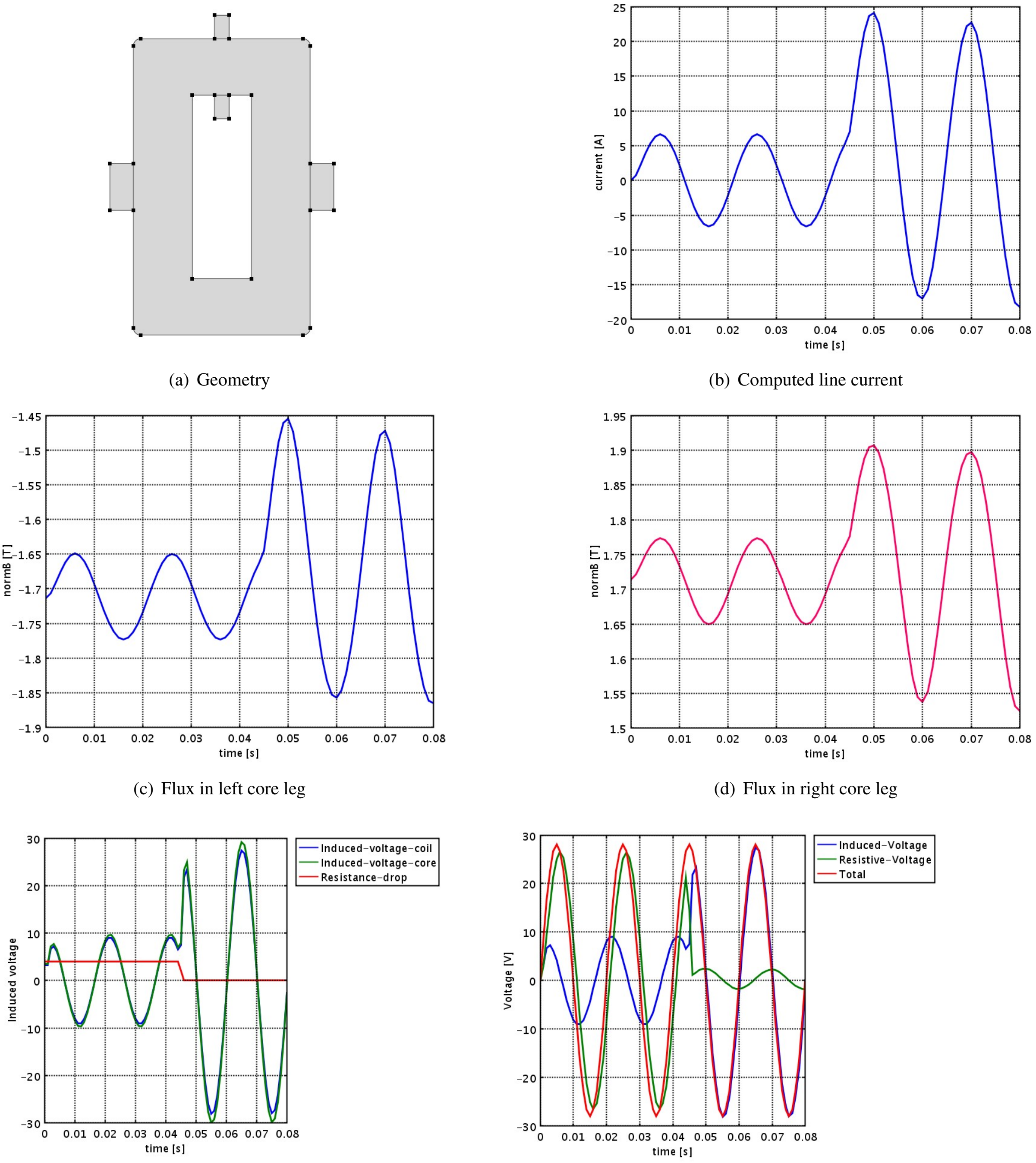
(a) Geometry

(b) Computed line current

(c) Flux in left and right core leg

(d) Induced Voltage

Figure 6.1: Numerical results for the O-shaped core configuration.



(a) Geometry

(b) Computed line current

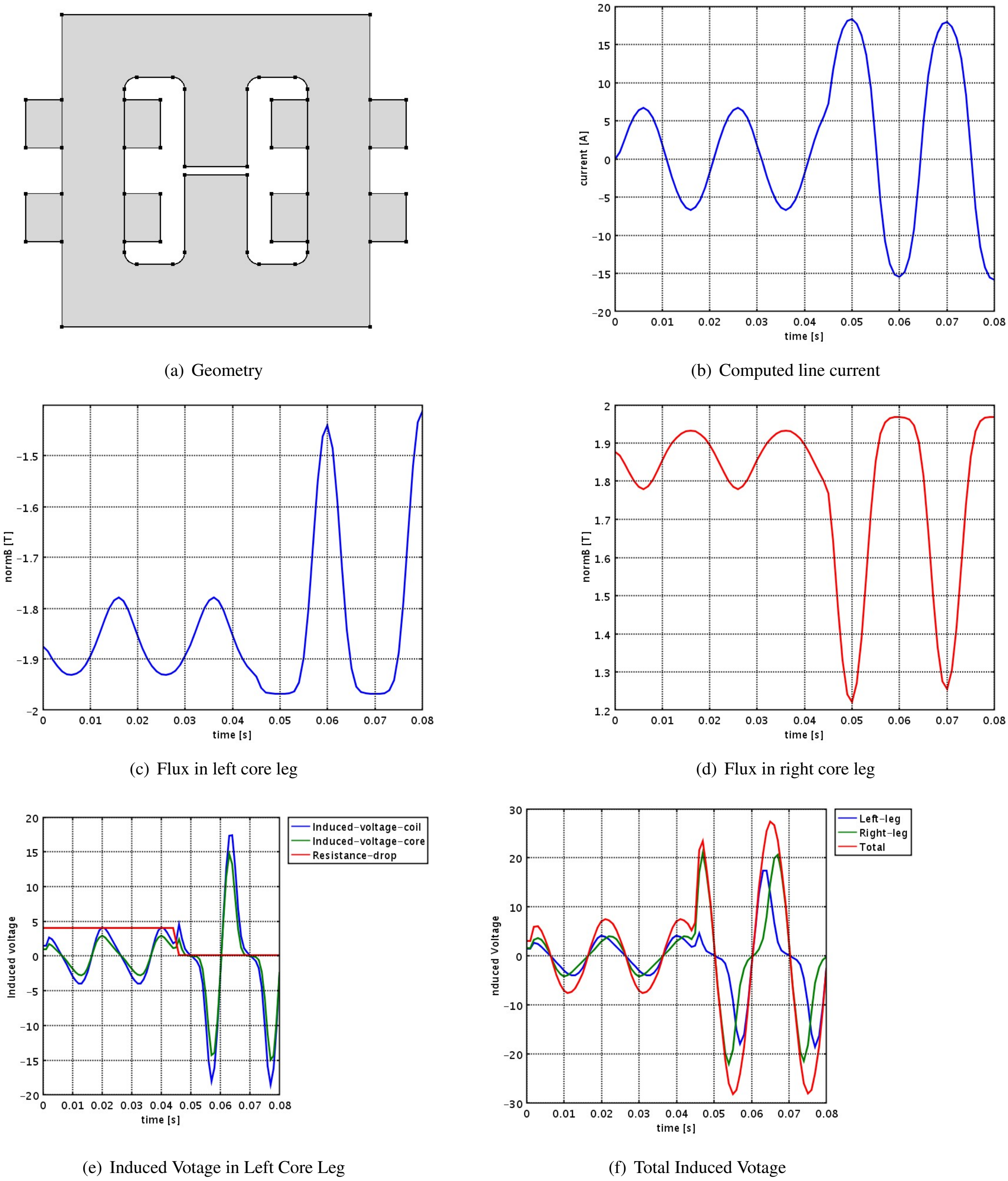
(c) Flux in left core leg

(d) Flux in right core leg

(e) Induced Votage

(f) Total Votage

Figure 6.2: Numerical results for the open-core configuration.



(a) Geometry

(b) Computed line current

(c) Flux in left core leg

(d) Flux in right core leg

(e) Induced Votage in Left Core Leg

(f) Total Induced Votage

Figure 6.3: Numerical results for the three-legs configuration.

```
crwleg = crwout - crwin;
crhin  = 39e-3;
crhout = 63e-3;
rad    = 3e-3;
core   = fillet(rect2(-crwout,crwout,-crhout,crhout),'radii',rad) - ...
                   rect2(-crwin,crwin,-crhin,crhin);
%....Flux integration lines....
fluxline1 = line1([-crwout,-crwin],[0,0]);
fluxline2 = line1([crwin,crwout],[0,0]);
%....AC coil....
accoilw      = 10e-3; accoilh = 20e-3;
xspacer      = 0e-3;
accoilradin  = crwin+crwleg+xspacer; accoilradout = accoilradin+accoilw;
accoil_right = rect2(accoilradin,accoilradout,-accoilh/2,accoilh/2);
accoil       = accoil_right+move(accoil_right,-2*accoilradin-accoilw,0);
%....DC coil....
dccoilh     = 10e-3; dccoilw = crwin/2;
dccoilradin = (crhout-crhin)/2; dccoilradout = dccoilradin + dccoilh;
dccoilyc    = crhin+(crhout-crhin)/2;
dccoil_top  = rect2(-dccoilw/2,dccoilw/2,dccoilradin,dccoilradout);
dccoil      = dccoil_top + move(dccoil_top,0,-2*dccoilradin-dccoilh);
dccoil      = move(dccoil,0,dccoilyc);
%....Air....
xmax = 400e-3; ymax = 400e-3;
air = rect2(-xmax, xmax, -ymax, ymax);

% Analyzed geometry
clear c s
c.objs={fluxline1,fluxline2};
c.name={'fluxline1','fluxline2'};
c.tags={'g5','g6'};

s.objs={air,accoil,dccoil,core};
s.name={'air1','accoil','dccoil','core'};
s.tags={'g1','g2','g3','g4'};

fem.draw=struct('c',c,'s',s);
fem.geom=geomcsg(fem);

%..Plot geometry..
if (0)
  geomplot(fem,'edgelabels','on')
  return
end

fem.const = {'om',           2*pi*50,...
             'Vmax',          28,...
             'Rpre',          4.0,...
             'Rpost',         0.1,...
             'Tfault',       45e-3,...
             'Tsmooth',      1e-3,...
             'lz',           50e-3,...
             'flcrwleg',     crwleg,...
             'crcross',      'lz*flcrwleg',...
             'flacNt',       acNt,...
             'accross',      accoilw*accoilh,...
             'flIac',        Iac,...
             'fldcNt',       dcNt,...
```

```
                'flIdc',            Idc,...
                'dccross',          dccoilw*dccoilh,...
                'linmurfe',         '1000',...
                'bha',              '2.12e-4',...
                'bhb',              '7.358',...
                'bhc',              '1.18e6',...
                'C1',               '.25',...
                'C2',               '.06'};

  % Functions
  clear fcns
  fcns{1}.type='inline';
  fcns{1}.name='rational(x,bha,bhb,bhc)';
  fcns{1}.expr='1/(bha + (1-bha)*x^bhb/(x^bhb+bhc))';
  fcns{1}.dexpr={'diff(1/(bha+(1-bha)*x^bhb/ (x^bhb+bhc)),x)',...
                 '0','0','0'};
  fcns{2}.type='inline';
  fcns{2}.name='hyperchord(x,C1,C2)';
  fcns{2}.expr='C2*x/(4*pi*1e-7*sinh(x/C1))';
  fcns{2}.dexpr={'diff(C2*x/(4*pi*1e-7*sinh(x/C1)),x)','0','0'};
  fcns{3}.type='interp';
  fcns{3}.name='bmurtabular';
  fcns{3}.method='linear';
  fcns{3}.extmethod='const';
  fcns{3}.filename='/mnt/dutita1/nw/domenico/software/bh_curve/mur_normb_data.txt';
  fem.functions = fcns;

  % Global expressions
  fem.globalexpr = {'Vline',      'Vmax*sin(om*t)',...
                    'Vind1',      'flacNt*lz/accross*Eint1',...
                    'Vind2',      'flacNt*lz/accross*Eint2',...
                    'Vind',       'Vind1+Vind2',...
                    'Bavrg1',     'Bint1/flcrwleg',...
                    'Bavrg2',     'Bint2/flcrwleg',...
                    'fluxt1',     'flacNt*lz*Btint1',...
                    'fluxt2',     'flacNt*lz*Btint2',...
                    'Rline',      'Rpre - (Rpre-Rpost)*flc2hs(t-Tfault,Tsmooth)'};

  % Initialize mesh
  fem.mesh=meshinit(fem);
%%  fem.mesh = meshrefine(fem);
%%  fem.mesh = meshrefine(fem);

   if (0)
     meshplot(fem)
     return
   end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  % Application mode 0: nominal current computation
  if (0)
    clear appl
    appl.mode.class = 'PerpendicularCurrents';
    appl.module = 'ACDC';
    clear bnd
    bnd.type = {'A0','cont'};
    bnd.ind = {[1,2,3,38],[4:37,39:41]};
```

```
    appl.bnd = bnd;
    clear equ
    equ.init = {'1e-6*sqrt(x^2+y^2)',0,0,0,0,0};
    if (strcmp(bhswtch,'linear')==1)
      equ.mur = {'linmurfe',1,1,1,1,1};
    elseif (strcmp(bhswtch,'rational')==1)
      equ.mur = {'rational(normB,bha,2*bhb,bhc)',1,1,1,1,1};
    elseif (strcmp(bhswtch,'hyperchord')==1)
      equ.mur = {'hyperchord(normB,C1,C2)',1,1,1,1,1};
    elseif (strcmp(bhswtch,'tabular')==1)
      equ.mur = {'bmurtabular(normB)',1,1,1,1,1};
    else
      error('  error:: undefined BH switch')
    end
        equ.Jez = {0,'-flacNt*flIac/accross','flacNt*flIac/accross',0,0,0};
    equ.ind = {[3,4],2,8,6,7,[1,5]};
    appl.equ = equ;
    fem.appl{1} = appl;
    fem.frame = {'ref'};
    fem.border = 1;
    clear units;
    units.basesystem = 'SI';
    fem.units = units;

    % ODE Settings
    clear ode
    ode.dim={'Itot'};
    ode.f={'0'};
    ode.init={'0'};
    ode.dinit={'0'};
    fem.ode=ode;

    % Multiphysics
    fem=multiphysics(fem);

    % Extend mesh
    fem.xmesh=meshextend(fem);

    % Solve problem
    fem.sol=femstatic(fem, ...
                      'solcomp',{'Az'}, ...
                      'outcomp',{'Az'});

    Rpre  = 4; Rpost = .1; om = 2*pi*50; Vmax = 28; lz = 25e-3;
    Wmtot = lz*postint(fem,'.5*Bx*Hx+.5*By*Hy','Edim',2,'Dl',[1:8]);
    L     = 2*Wmtot /Iac^2;
    Xpre  = sqrt(Rpre^2  + om^2*L^2); Ipre = Vmax/Xpre;
    Xpost = sqrt(Rpost^2 + om^2*L^2); Ipost = Vmax/Xpost;
    fprintf('  The inductance of the coil   = %f H. \n', L)
    fprintf('  Nominal current before fault = %f Amp. \n', Ipre)
    fprintf('  Nominal current after  fault = %f Amp. \n', Ipost)

    return

  end %.. Nominal current computation

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```
  % Application mode 1: generating initial guess for the DC coil flux
  if (1)
    clear appl
    appl.mode.class = 'PerpendicularCurrents';
    appl.module = 'ACDC';
    clear bnd
    bnd.type = {'A0','cont'};
    bnd.ind = {[1,2,3,38],[4:37,39:41]};
    appl.bnd = bnd;
    clear equ
    equ.mur = {'100',1,1,1,1,1};
    equ.Jez = {0,0,0,'fldcNt*flIdc/dccross',...
                  '-fldcNt*flIdc/dccross',0};
    equ.ind = {[3,4],2,8,6,7,[1,5]};
    appl.equ = equ;
    fem.appl{1} = appl;
    fem.frame = {'ref'};
    fem.border = 1;
    clear units;
    units.basesystem = 'SI';
    fem.units = units;

    % ODE Settings
    clear ode
    ode.dim={'Itot'};
    ode.f={'0'};
    ode.init={'0'};
    ode.dinit={'0'};
    fem.ode=ode;

    % Multiphysics
    fem=multiphysics(fem);

    % Extend mesh
    fem.xmesh=meshextend(fem);

    % Solve problem
    fprintf('  Generating initial guess for the DC coil flux ...\n')
    fem.sol=femstatic(fem, ...
                      'solcomp',{'Az'}, ...
                      'outcomp',{'Az','Itot'});
    fprintf('  ... done! \n')

    fem0 = fem;


  end %.. Parametric solver in DC winding

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  % Application mode 2: generating initial guess for the AC coil flux
  if (1)
    clear appl
    appl.mode.class = 'PerpendicularCurrents';
    appl.module = 'ACDC';
    clear bnd
    bnd.type = {'A0','cont'};
    bnd.ind = {[1,2,3,38],[4:37,39:41]};
```

```
      appl.bnd = bnd;
      clear equ
      if (strcmp(bhswtch,'linear')==1)
        equ.mur = {'linmurfe',1,1,1,1,1};
      elseif (strcmp(bhswtch,'rational')==1)
        equ.mur = {'rational(normB,bha,2*bhb,bhc)',1,1,1,1,1};
      elseif (strcmp(bhswtch,'hyperchord')==1)
        equ.mur = {'hyperchord(normB,C1,C2)',1,1,1,1,1};
      elseif (strcmp(bhswtch,'tabular')==1)
        equ.mur = {'bmurtabular(normB)',1,1,1,1,1};
      else
        error('  error:: undefined BH switch')
      end
      equ.Jez = {0,0,0,'fldcNt*flIdc/dccross',...
                 '-fldcNt*flIdc/dccross',0};
      equ.ind = {[3,4],2,8,6,7,[1,5]};
      appl.equ = equ;
      fem.appl{1} = appl;
      fem.frame = {'ref'};
      fem.border = 1;
      clear units;
      units.basesystem = 'SI';
      fem.units = units;

      % ODE Settings
      clear ode
      ode.dim={'Itot'};
      ode.f={'0'};
      ode.init={'0'};
      ode.dinit={'0'};
      fem.ode=ode;

      % Multiphysics
      fem=multiphysics(fem);

      % Extend mesh
      fem.xmesh=meshextend(fem);

      % Solve problem
      fprintf('  Generating initial guess for the DC coil flux ...\n')
      fem.sol=femstatic(fem, ...
                        'init', fem0.sol,...
                        'solcomp',{'Az'}, ...
                        'outcomp',{'Az','Itot'}, ...
                        'Pname','flIdc',...
                        'Plist',[Idc],...
                        'Maxiter',40,...
                        'Ntol',1e-8);
      fprintf('  ... done! \n')

      fem0 = fem;

    end %.. Generatring initial guess for the AC flux

  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    % Application mode 3: transient simulation
    % Saturation current in the DC winding
```

```
% ODE solution in the AC winding

clear appl
appl.mode.class = 'PerpendicularCurrents';
appl.module = 'ACDC';
clear prop
prop.analysis='transient';
appl.prop = prop;
clear bnd
bnd.type = {'A0','cont'};
bnd.ind = {[1,2,3,38],[4:37,39:41]};
appl.bnd = bnd;
clear equ
equ.init = {'1e-6*sqrt(x^2+y^2)',0,0,0,0,0};
if (strcmp(bhswtch,'linear')==1)
  equ.mur = {'linmurfe',1,1,1,1,1};
elseif (strcmp(bhswtch,'rational')==1)
  equ.mur = {'rational(normB,bha,2*bhb,bhc)',1,1,1,1,1};
elseif (strcmp(bhswtch,'hyperchord')==1)
  equ.mur = {'hyperchord(normB,C1,C2)',1,1,1,1,1};
elseif (strcmp(bhswtch,'tabular')==1)
  equ.mur = {'bmurtabular(normB)',1,1,1,1,1};
else
  error('␣␣error::undefined␣BH␣switch')
end
equ.Jez = {0,'flacNt*Itot/accross','-flacNt*Itot/accross',...
              'fldcNt*flIdc/dccross','-fldcNt*flIdc/dccross',0};
equ.ind = {[3,4],2,8,6,7,[1,5]};
appl.equ = equ;
fem.appl{1} = appl;
fem.frame = {'ref'};
fem.border = 1;
clear units;
units.basesystem = 'SI';
fem.units = units;

% Coupling variable elements
clear elemcpl
% Integration coupling variables
clear elem
elem.elem = 'elcplscalar';
elem.g = {'1'};
src = cell(1,1);
clear bnd
bnd.expr = {{{},{},'By',{},{},{}},{{},{},{},{},'By',{}},{{},{},'-Azxt',{}, ...
  {},{}},{{},{},{},{},'-Azxt',{}},{},{}};
bnd.ipoints = {{{},'4','4',{},{},{}},{{},{},{},'4','4',{}},{{},'4','4',{},{}, ...
  {}},{{},'4',{},'4','4',{}},{},{}};
bnd.frame = {{{},'ref','ref',{},{},'ref'},{{},{},{},'ref','ref','ref'},{{}, ...
  'ref','ref',{},{},'ref'},{{},'ref',{},'ref','ref','ref'},{},{}};
bnd.ind = {{'1','2','3','4','5','7','8','9','11','12','13','14','15', ...
  '16','17','18','19','20','21','22','23','24','25','26','27','28','31', ...
  '32','33','34','35','36','37','38','39','40'},{'6'},{'10'},{'29'},{'30'}, ...
  {'41','42'}};
clear equ
equ.expr = {{},{},{},{},{{},'-Ez',{},{}},{{},{},{},'Ez'}};
equ.ipoints = {{},{},{},{},{{},'4','4',{}},{{},{},{},'4'}};
equ.frame = {{},{},{},{},{{},'ref','ref','ref'},{{},{},'ref','ref'}};
```

```
equ.ind = {{'1','3','4','5','6'},{'2'},{'7'},{'8'}};
src{1} = {{},bnd,equ};
elem.src = src;
geomdim = cell(1,1);
geomdim{1} = {};
elem.geomdim = geomdim;
elem.var = {'Bint1','Bint2','Btint1','Btint2','Eint1','Eint2'};
elem.global = {'1','2','3','4','5','6'};
elem.maxvars = {};
elemcpl{1} = elem;
fem.elemcpl = elemcpl;

% ODE Settings
clear ode
ode.dim={'Itot'};
ode.f={'Rline*Itot+Vind-Vline'};
ode.init={'0'};
ode.dinit={'0'};
fem.ode=ode;

% Multiphysics
fem=multiphysics(fem);

% Extend mesh
fem.xmesh=meshextend(fem);

% Solve problem
fem.sol=femtime(fem, ...
                'init', fem0.sol,...
                'solcomp',{'Az','Itot'}, ...
                'outcomp',{'Az','Itot'}, ...
                'atol',1e-6,...
                'rtol',1e-6,...
                'tlist',[0:0.001:0.08], ...
                'tout','tlist',...
                'nlsolver','manual', ...
                'ntolfact',0.01, ...
                'maxiter',25, ...
                'dtech','const', ...
                'damp',1.0, ...
                'jtech','minimal',...
                'linsolver','pardiso', ...
                'errorchk','off');

% Plot current
postglobalplot(fem,'Itot')
postglobalplot(fem,{'Bavrg1','Bavrg2'})
postglobalplot(fem,{'Vind1+Vind2','fluxt1+fluxt2'})
postglobalplot(fem,'Itot')
```

Chapter 7

Three Dimensional FE Models of Inductive Fault Current Limiters

To do

- search for last implementation of open core and three legs model
- add references to De Gersem and Dular on the field circuit coupling

7.1 Open-Core Configuration

7.1.1 Geometry

In defining the geometry, the core acts as *master*, while the DC and AC coil act as slave. The geometry will consist of the following four parts:

1. **the core**: build by extruding a working plane in the yz-plane in the x-direction. There will be a relation between the depth and width of the core leg.

   (a) core variables

   ```
   crlegw = ;
   crwin = ; crwout = crwin + crlegw;
   crhout = ; crhin = ;
   crd = ;
   ```

   (b) core working plane:

   ```
   core_plane = fillet(rect2(-crwout,crwout,-crhout, crhout), 'rad', rad) ...
                             - fillet(rect2(crwin,crwin,-crhin,crhin), 'rad', rad);
   ```

   (c) core extrusion

   ```
   set working plane at x = 0
   core = extrude(core_plane, 'distance', 2*crd);
   ```

2. **flux integration surfaces**:

3. **the generic coil**: build by extruding a working plane in the xy-plane in the z-direction

   (a) generic coil variables

   ```
   coilw = ; coilh = ;
   yspacer = ;  zspacer = ;
   coilradin = crlegw/2+yspacer; coilradout = coilradin+coilw;
   coilyc = crwin+crwleg/2; coilzc= coilh/2 + zspacer;
   ```

   (b) coil working plane:

   ```
   coil_plane = circ2(coilradout) - circ2(coilradin);
   ```

(c) coil extrusion

```
set working plane at z = 0
coil = extrude(coil_plane, 'distance', coilh)
```

4. **the DC coil**: by moving the generic coil in positive y-direction and in positive z-direction

   (a) DC coil variable

   ```
   dccoilzc= coilh/2 + zspacer;
   ```

   (b) `dccoil = move(coil,0,coilyc,dccoilzc);`

5. **the AC coil**: by moving the generic coil in positive y-direction and in negative z-direction

   (a) AC coil variable

   ```
   accoilzc= -coilh/2 - zspacer;
   ```

   (b) `accoil = move(coil,0,coilyc,accoilzc);`

6. **the air**: build using the `block3` command

For the ease of modification of this geometry we will work with a full model.

7.1.2 Meshing

Currently the meshing happens fully automatically, excepts for the option `hauto` used in `meshinit`. Due to this automation, difficulties may occur in case the space between the coils and the core leg is to small. This issue will have to be dealt with in the future.

7.1.3 Constants, Functions and Subdomain and Global Expressions

Constants

Note that a fill-factor in the coils in **not** used

Electrical constants	
ω	$2\ \pi\ 50$
DC coil	
Number of turns Cross-section Current value	
AC coil	
Number of turns Cross-section	
BH curve data	
a b c C_1 C_2	2.12e-4 7.358 1.18e6 .25 .06

Table 7.1: Constants Used

Functions

Here we define the BH curves and the winding functions.

Subdomain expressions

Here we define the densities for the induced voltage

Global expressions

Here we define the electrical excitation and the induced voltage.

7.1.4 Application modes

We solve for both the vector and scalar potential. The scalar potential is used to avoid that during time integration components in the null space of the curl-curl operator are introduced.

7.1.5 Integration coupling variables

Here we integrate the induced voltage density to obtain the total induced voltage.

7.1.6 Definition of the ODE

The sum of resistive and induced voltage is at all times equal to the total applied voltage.

7.1.7 Solution process

7.1.8 Post processing

7.2 Different solution modes

7.2.1 Linear core

7.2.2 Non-linear core

Define the different stages.

Chapter 8

Size Optimization

1. zeroth order model by Dalibor: number of winding determines the Ohmic losses. These losses should not exceed an a-priori established limit. This is guaranteed by imposing the number of turns. From this number of turns and from the value of the applied voltage, normB and by using the analyical expressions for the impedance and the induced voltage, an estimate of the leg cross-section, and thus the leg width can be establised.

2. formulation of the sizing optimization (later extend to topology optimization) problem
(in the order in which we intend to solve the problem):

   (a) minimize mass subject to sufficient current limiting capabilities. This limiting capability can be set equal to the one of the initial configuration.

   (b) maximize current limiting capabilities subject to a constraint on the mass (same remark as in previous case)

   (c) multi-objective optimization problem with mass and limiting capability as conflicting objectives: compute the Pareto-front (can be done easily using an analytical model)

3. design variables and box constraints: we first consider the sizes of the core as design variables. In the design, the vertical and horizontal core leg width should remain equal. We therefore have four design variables: the half inner window width and height (cr_w_in and cr_h_in), the depth (cr_d) and core leg width (cr_leg_w). The lower bounds on the first two variables should be such to leave sufficient space for the dc coil. In a first analysis the size of the inner window and the core depth can be taken to constant. In a second analysis we can allow the height of the inner window to change. We observe that changing the width of the window implies more material for the AC coil. The same is not true for changes in the height.
Experiments by Dalibor indicate that for the current limiting capability is most sensitive to the cross-section of the core legs and the number of turns in the ac coil.
In a later stage we will add dimensions of the ac and dc coil as design variables and number of turns of the coil (mixed integer problem).

4. computation of the objectives: for the computation of the mass we use

$$\begin{aligned}\text{core-mass} &= \rho\, cr\_d\,[2(cr\_h\_in + cr\_leg\_w)\cdot 2(cr\_w\_in + cr\_leg\_w) - 2cr\_h\_in \cdot 2cr\_w\_in)] \quad (8.1)\\ &= \rho\, cr\_d\,[4cr\_leg\_w \cdot (cr\_h\_in + cr\_h\_in + cr\_leg\_w)]\,; \quad (8.2)\end{aligned}$$

   for the computation of the current limiting capability or induced voltage V_{ind} we will for time being consider a time-harmonic computation ($\frac{d}{dt} \rightarrow j\omega$) of the post-fault situation. Doing so we do **not** take the peak current into account. We have that

$$I = \text{Re}[\widehat{I}\exp(j\omega t)]\,, \quad (8.3)$$

   and solve for $\widehat{I}$. We consider the following three models of increasing complexity:

   (a) analytical model. One of the difficulties in the analytical model is the correct estimation of the flux path.

We have that

$$
\begin{aligned}
V_{ind} &= \frac{d}{dt}(L\,I) &(8.4)\\
&= j\omega L\,I &(8.5)\\
&= j\omega\mu_r\mu_0 N_{ac}^2\frac{A_c}{l}\,I &(8.6)\\
&= 2\,\pi\,j\,f\mu_r\mu_0 N_{ac}^2\frac{cr\_leg\_w\cdot cr\_d}{l}I\,, &(8.7)
\end{aligned}
$$

where l denotes the length of the flux path. This means that the induced voltage lags the current by $\pi/2$ and its amplitude is function of the design variables. In a classical ac core layout, one could put

$$l = 2(cr\_h\_in + cr\_w\_in + cr\_w\_leg) \tag{8.8}$$

For the open-core a more appropriate expression for the flux path could be

$$l = C\cdot cr\_h\_in \tag{8.9}$$

where $C > 1$ is a constant to be determined.

(b) 2D and 3D time-harmonic FEM model. In this model we have

$$V_{ind} = N_{ac}\frac{d}{dt}\int_{S_{cr}} B_y d\Omega = j\omega N_{ac}\int_{S_{cr}} B_y d\Omega = V_{ind,1} - V_{ind,2}\,, \tag{8.10}$$

where

$$V_{ind,i} = \frac{N_{ac}\,\ell_z}{S_{cl,i}}\int_{S_{cl,i}} E_z d\Omega = j\omega\frac{N_{ac}\,\ell_z}{S_{cl,i}}\int_{S_{cl,i}} A_z d\Omega\,. \tag{8.11}$$

5. initial configuration and objective values:

parameter	notation	value	units
mass density core	ρ	7850	kg/m^3
number of AC turns	N_{ac}	100	-

Table 8.1: parameter values

cr_w_in	xxx [mm]	core-mass	xxx [kg]
cr_h_in	xxx [mm]	V_{ind}	xxx [V]
cr_leg_w	xxx [mm]		
cr_d	xxx [mm]		

Table 8.2: Initial configuration and objective values.

6. coarse model (analytical) design problem and solution techniques: in case we set out to minimize the mass, we obtain:
find $x^* \in X$ such that:

$$x^* = \text{argmin}_{x\in X}\text{core-mass}(x) \quad \text{such that} \quad V_{ind}(x) \geq V_{ind,0} \tag{8.12}$$

By incorporating the constraint on the induced voltage in the definition of the design space space, we can formulate this as
find $x^* \in \overline{X}$ such that:

$$x^* = \text{argmin}_{x\in \overline{X}}\text{core-mass}(x) \tag{8.13}$$

We intend to solve this problem using the Nelder-Mead simplex method and a gradient based optimization algorithm with exact gradients with multiple starting points. Can the latter be done in Maple?

7. Pareto front: generate the coarse model Pareto front using a brute force approach (4-fold loop in the design space).
8. space mapping optimization: identity mapping on the mass, constraint mapping on the induced voltage.
9. work in stages: Pareto front for analytical model (Dalibor?), build 2D time-harmonic FEM (Domenico), Pareto front for 2D FEM model, build space-mapping function via a least-squares procedure, Pareto front for mapped coarse model and comparison with FEM model, extend to 3D FEM, extend from time-harmonic to transient.

Bibliography

[1] Donald E. Knuth. Safety Transformer Mofdel. Technical Report STAN-CS-83-980, Department of Computer Science, Stanford University, Stanford, CA 94305, September 1983.

[2] Donald E. Knuth. Safety Transformer Mofdel. Technical Report STAN-CS-83-980, Department of Computer Science, Stanford University, Stanford, CA 94305, September 1983.