A Methodology for the Simulation of MEMS Spiral Inductances used as Magnetic Sensors

Sylvain Druart, Denis Flandre, and Laurent A. Francis
Université catholique de Louvain – ICTEAM Institute
B-1348 Louvain-la-Neuve, Belgium
{Sylvain.Druart, Denis.Flandre, Laurent.Francis}@uclouvain.be

Abstract: In this paper, a methodology to simulate the electric behavior of spiral inductances is presented and discussed. All the methodology is built with the Comsol software used with the Matlab scripting interface and then allows performing fully parameterized simulations. The program architecture is explained and is used to simulate two applications. The first calculates the voltage induced by an external AC magnetic field. The second is to detect the presence of moving metallic particles of micrometric dimensions; the program thus extracts the varying inductances values. The final goal of this approach is to manufacture sensors in MEMS technologies and co-integrate them with CMOS circuits.

Keywords: Inductance, FEM simulation, magnetic sensor, spiral-shaped.

1. Introduction

Magnetic sensors are needed in a lot of applications but because of the increasing complexity and consumption of systems, they have to be miniaturized to reach confined areas and work at very low power. The methodology presented here is used to simulate the electric response of a square-shaped spiral MEMS inductor in the case of two applications, magnetic field measurement and metallic particle detection. The future purpose of this work is to manufacture inductance in MEMS technologies and to integrate them with CMOS circuit to limit the impact of wire connections; it thus becomes easier to extract the information provided by the sensor. Several works have been achieved in the MEMS inductor fabrication like [3], [4] and [5].

The methodology starts from the existing library Comsol model of the square inductor [1]. We largely extend this approach and explain it with a more electronical than mathematical point of view on the parameter values.

This paper is divided in three sections. The first explains which equation model is used to solve the problem and presents the system geometry. The second part develops the programming structure and describes the strategy used to build a robust finite element structure. The last section shows the simulation results by extraction of the electric response. The inductance and resistance values are calculated before showing the magnetic field and particle detection results.

2. Model description

2.1 Equation model

The electromagnetic module is used for all the simulations; it consists in the general expression of the electric and magnetic potentials (expression 1 and 2). The system is defined by four variables to solve, one for the electric potential and three for the magnetic potential. All symbols are described in Table 1.

$$\boldsymbol{\nabla}\cdot[(j\omega\sigma-\omega^2\varepsilon_0\varepsilon_r)\boldsymbol{A}+(\sigma+j\varepsilon_0\varepsilon_r)\boldsymbol{\nabla}\mathrm{V}]=0 \quad (1)$$

$$(j\omega\sigma-\omega^2\varepsilon_0\varepsilon_r)\boldsymbol{A}+\boldsymbol{\nabla}\times\left(\frac{\mathbf{1}}{\boldsymbol{\mu_0\mu_r}}\boldsymbol{\nabla}\times\mathbf{A}\right)+(\sigma+j\varepsilon_0\varepsilon_r)\boldsymbol{\nabla}\mathrm{V}=0 \quad (2)$$

Table 1: equation parameter description

Parameter description	Symbol	Units	Type
Electric potential	V	[V]	/
Magnetic potential	**A**	[T.m]	/
Angular frequency	ω	[rad.s^{-1}]	variable
Electric conductivity	σ	S.m^{-1}	variable
Relative permittivity	ε_0	[-]	constant
Absolute permittivity	ε_r	[F.m^{-1}]	constant
Relative permeability	μ_0	[-]	constant
Absolute permeability	μ_r	[H.m^{-1}]	constant

2.2 Inductance geometry

The geometry is built with the Matlab scripting interface and is generated automatically according to the input parameters. Fig. 1 shows an example of the geometry built with the input parameters described in Table 2. The outer domain is divided into two parts, the substrate (lower box) and the air (upper box) which are large enough to avoid boundary effects. All the simulations are based on the resulting magnetic field generated in the system.

Table 2: geometric and electrical parameter description

Parameter description	Symbol	value	units
Inner square length	L_c	500	µm
Number of turns	N_{sp}	6	-
Turn spacing	W_{sp}	20	µm
Wire width	W_w	20	µm
Wire height	H_w	20	µm
Box relative permittivity	$\varepsilon_{r,Box}$	3.9	-
Air relative permittivity	$\varepsilon_{r,Air}$	1	-
Wire conductivity	σ_{Wire}	56.9	$MS.m^{-1}$

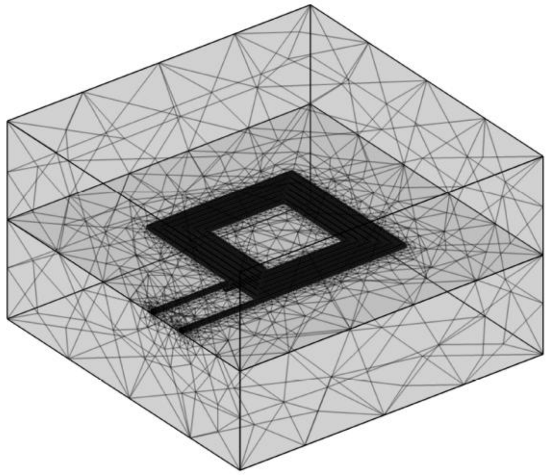


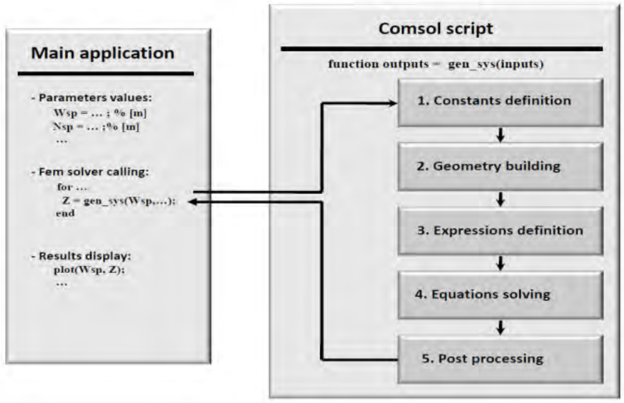
Fig. 1: inductance geometry

3. Comsol scripting techniques

In this section, an explanation of how Comsol is called is provided. The script method shows very powerful flexibility to compute a lot of various simulations.

3.1 Program hierarchy

As shown in Fig. 2, the main application class defines the simulation that will be performed. The class contains input parameters definitions, the calls for Comsol solving method and the results display functions. The right part of the class diagram contains the Comsol solving script. It includes five steps to provide the output results.



```
Wsp = ... ; % [m]
Nsp = ... ;% [m]
...
```

```
for ...
  Z = gen_sys(Wsp,...);
end
```

```
plot(Wsp, Z);
...
```

```
function outputs = gen_sys(inputs)
```

Fig. 2: Scripting diagram

3.2 Scripting steps explanation

1. Constants definition: This first step defines all the constants needed by the finite element structure:

```
...
% Input current [A]
fem.const.Ipol = 1e-5;
% External magnetic field [T]
fem.const.Bzext = 1e-4;
...
```

2. Geometry building: The finite element geometry is built with design methods containing the Comsol drawing functions using the input parameters:

```
...
Bulk = createBlock('xyz params');
Air = createBlock('xyz params');
Domain = geomcoerce('solid',{Bulk,Air});
...
```

The inductance turns are drawn with a "for" loop which assembles the geometry progressively. The final object is then meshed with specific refinement in areas which are critical for the solver.

3. Expressions definition: This step is not mandatory for all kind of simulations. Here are described variables and unknowns needed to provide the outputs parameters. For example, an external magnetic source is needed to generate an induced voltage on the inductor. The source is implemented using the magnetic potential *(Aextx; Aexty; Aextz)*:

```
...
fem.expr.Aextx = '-y*Bzext*0.5';
fem.expr.Aexty = 'x*Bzext*0.5';
fem.expr.Aextz = '0';
...
```

4. Equations solving: The fourth step includes the solving method core. The harmonic electromagnetic module is called and all the subdomains and boundaries are specified following their own index. The index allocation becomes a very complex problem when the number of geometry shapes increases. For example, modifying the number of turns causes all the index allocation to change. A way to perform this important problem is described below in the last part of the scripting techniques section. The second part of the equation step is the call of the Comsol solving algorithm. It uses the "multigrid" technique described in [2].

5. Post processing: The integration functions are used to provide the output parameters that have to be analyzed by the main file application. The post processing step extracts all the expected parameters like impedance or induced voltage.

3.3 Indexing problem

The numbering of subdomains and boundaries causes troubles if the geometry is modified. The way to avoid those problems is to calculate all mass centers of subdomains and boundaries and to memorize this information after the meshing step. Here are the code lines for the mass center calculation of the nth subdomain:

```
...
Vol = postint(fem,’1’,...,’edim’,3);
XM = postint(fem,’x’,...,’edim’,3);
YM = postint(fem,’y’,...,’edim’,3);
ZM = postint(fem,’z’,...,’edim’,3);
CrdsMVol(n,1) = XM/Vol;
CrdsMVol(n,2) = YM/Vol;
CrdsMVol(n,3) = ZM/Vol;
...
```

When the program assigns the subdomain and boundary conditions, only the space coordinates of where the condition has to be set are needed. Then a *getIndex* method provides the domain index which has the closest mass center to the coordinates chosen for the condition assignments. This method thus allows for translating all the condition assignment in coordinates instead of index numbers. For example, a subdomain condition for the air area is set:

```
...
indAir = getIndex(xin, yin, zin);
equ.ind(indAir) = 2;
...
```

4. Simulation results

4.1 Inductance and resistance extraction

As the inductance and resistance calculation requires isnulating the whole system electrically and magnetically, only the ports of the inductance have specific conditions. One port is connected to the ground to fix a potential reference and the other receives a fixed amount of inward current. The Comsol library model [1] fixes an input voltage instead of an inward current; it does not affect the solution but fixing a voltage across a metallic device is electrically not a good way to do for two reasons. The first lies in the very huge and unrealistic output current values obtained and the second lies in a bad estimation of the device power consumption.

The inductance value can be extracted by two ways, the first consists in the integration of the magnetic energy density *Wmav* provided by the module (eq. 3) and the second consists in the calculation of the equivalent impedance seen at the two ports (eq. 4). Fig. 3 shows the results of the inductance and resistance values. The input varying parameters are the inner square length L_c and the number of turns N_{bs}.

$$L = \frac{2\iiint_{\Omega} W_{mav} dV}{I_{eff}^2} \qquad (3)$$

$$L = \frac{Im(Z_{sys})}{\omega} \qquad (4)$$

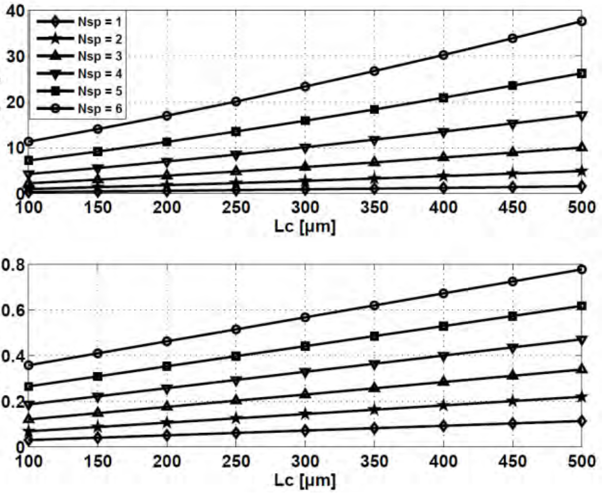


Fig. 3: Inductance and resistance results

4.2 Induced voltage calculation

The first concrete application is magnetic sensing. An external varying high frequency magnetic field has to be measured with the best accuracy and sensitivity. The simulation method applied here slightly changes the system geometry. The inductor bounds are connected to a high resistivity material in order to generate an induced voltage due to the external magnetic field (Fig. 4).

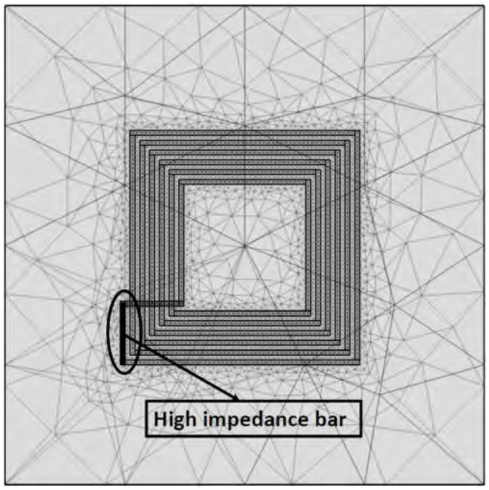


Fig. 4: Field detection modified geometry

The external magnetic field B is generated using the magnetic potential A which formula is given by expression 5.

$$\boldsymbol{B} = \nabla \times \boldsymbol{A} \qquad (5)$$

The following equations yield a solution for the magnetic potential expression according to a uniform magnetic field Bz_{ext}, perpendicular to the inductor plane. These expressions constitute the boundary conditions on the upper and lower faces of the whole domain.

$$\begin{cases} Ax = \dfrac{-Bz_{ext}}{2} y \\ Ay = \dfrac{Bz_{ext}}{2} x \\ Az = 0 \end{cases}$$

The simulation results show the sensitivity S of the sensor given by the induced voltage when a field magnitude Bz_{ext} of 1T is applied externally. The solution has been computed at 1MHz. A high frequency will produce a very good sensitivity but the higher the system frequency, the greater the difficulty and power consumption when interfaced with electronic circuits.

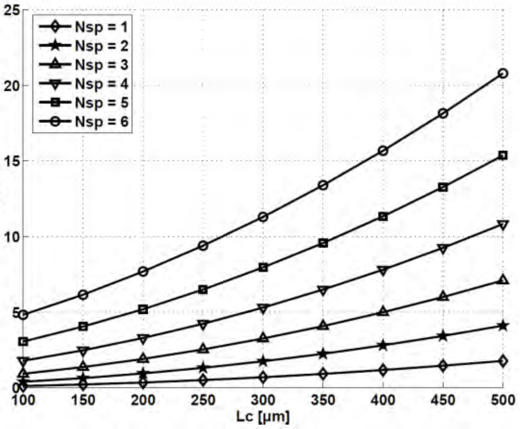


Fig. 5: Sensitivity to field detection

4.3 Metallic particles detection

Inductors are often used to make metallic mines detection at macrometric scales. The present application shows it is possible to observe the same phenomenon by simulation at micrometric scale. The analysis explained here shows that a single metallic particle moving over the sensitive area causes the inductance to change significantly. The travel of the particle is illustrated in Fig. 6 following its mass center coordinate X_c whose origin is fixed at the center of the square inner space. Its dimensions are close to the inductor inner square length. The particle height is smaller than the other dimensions in order to avoid a too heavy spatial complexity of the problem. The results plotted in Fig. 7 were obtained for two kinds of metal 200x200x20 µm^3 particles, zinc and silver. Silver is more conductive and then makes the inductance to decrease more. A 20 µm vertical distance between the particle and the coil is considered.

5. Conclusions

The simulation methodology presented in this paper shows results for the parameterized oriented simulations. The magnetic sensing effects have been studied with an original approach and led to accurate behavior prediction in terms of impedance and induced voltage calculation. The metallic particle detection has also been successfully simulated. A complete simulation tool has been developed and proved its efficiency in electromagnetic domain and thus can be extended to other modules and science domains.

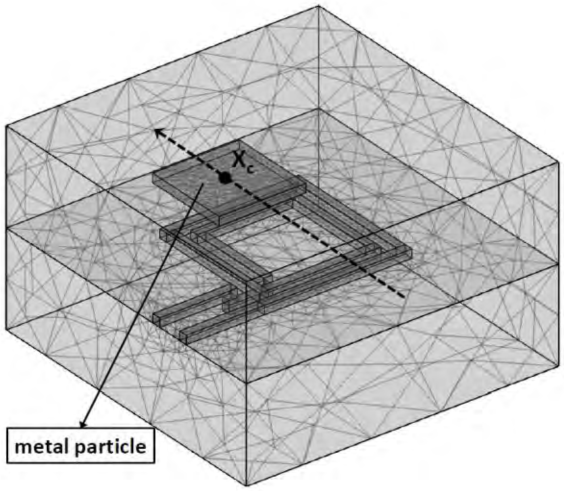


Fig. 6: Inductor with metal particle

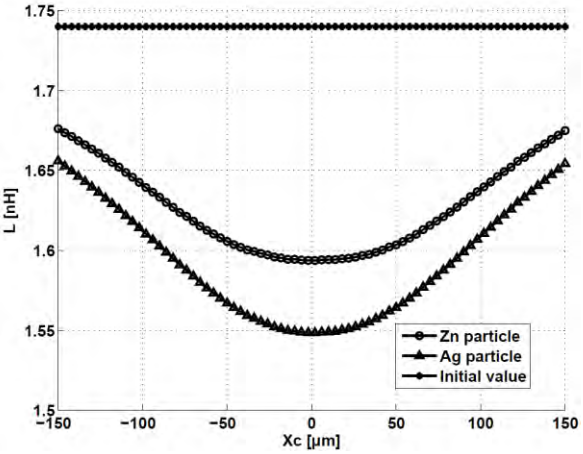


Fig. 7: Inductance value with moving particles

6. Acknowledgment

This work has been funded by the Wallonia Plan Marshall under the HM+ project for the airplanes health monitoring.

7. References

[1] *Comsol Multiphysics - AC/DC Module - Model Library*, version 3.5a - pp. 148-158

[2] *Comsol Multiphysics - AC/DC Module - User's Guide*, version 3.5a - pp. 114-118

[3] *High-Q solenoid inductors with a CMOS-compatible concave-suspending MEMS process*, L. Gu and X. Li - Journal of Microelectromechanical Systems, vol. 16, no. 5 - pp. 1162-1172, 2007

[4] *CMOS-compatible surface-micromachined suspended-spiral inductors for multi-GHz silicon RF ICs*, J. Yoon, Y. Choi, B. Kim, Y. Eo, and E. Yoon - IEEE Electron Device Letters, vol. 23, no. 10 - pp. 591-593, 2002

[5] *Realization of high-Q inductors using wirebonding technology*, S. Kim, Y. Lee, S. Yun, and H. Lee - AP-ASIC'99 - pp. 13-16, 1999