



NOVA SCHOOL OF  
SCIENCE & TECHNOLOGY

---

# EDA/CAD para Nanoelectrónica

2º Semestre 2020 / 2021

## Relatório

*Lab#3: Estudo e Aplicação do Modelo- $\pi$*

18/06/2021

Número de Aluno	Nome	Turno Prático
50726	Francisco Mendes Micaelo André	P1
51005	José Maria Corrêa Arouca Cortes Tamagnini	P1
43853	Rafael Gonçalves Feio de Oliveira	P1

**Docente:** Maria Helena Fino

# Índice

<b>Introdução.....</b>	<b>4</b>
<b>I-Objetivo.....</b>	<b>5</b>
<b>II - Apresentação do <i>Modelo-<math>\pi</math></i> .....</b>	<b>6</b>
2.1. Descrição do <i>Modelo-<math>\pi</math></i> .....	6
<b>III - Metodologia .....</b>	<b>9</b>
3.1. Determinação e caracterização dos parâmetros do <i>Modelo-<math>\pi</math></i> .....	9
3.2. Simulação LT Spice .....	11
3.3. Otimização .....	13
3.4. Cadence – <i>Verilog</i> e simulações .....	13
3.5. Gráficos e Erros.....	14
<b>IV - Resultados .....</b>	<b>16</b>
4.1. Parâmetros Obtidos .....	16
4.2. Análise de Resultados.....	17
<b>V - Conclusões .....</b>	<b>23</b>
<b>Bibliografia .....</b>	<b>24</b>
<b>Anexos .....</b>	<b>25</b>
Anexo A: Código do <i>Modelo-<math>\pi</math> Python</i> .....	25
Anexo B: Código do <i>Modelo-<math>\pi</math> Verilog</i> .....	33

## Índice de Figuras

Figura 1 – Esquema ilustrativo do indutor .....	5
Figura 2 – Esquema do <i>Modelo-<math>\pi</math></i> .....	8
Figura 3 - Esquema em LTSpice do <i>Modelo-<math>\pi</math></i> simulado.....	11
Figura 4 – Dataframe do tipo <i>Lt_Shape</i> .....	12
Figura 5 - Circuito para a simulação do <i>Modelo-<math>\pi</math></i> em <i>Cadence</i> .....	14
Figura 6 – Fator de Qualidade em função da frequência a) Quadrangular; b) Hexagonal; c) Octagonal.....	18
Figura 7 - Indutância em função da frequência a) Quadrangular; b) Hexagonal; c) Octagonal	19
Figura 8 – Erro relativo em função da frequência a) Quadrangular; b) Hexagonal; c) Octagonal .....	20
Figura 9 - Fator de Qualidade em função da frequência, simulação ótima vs não ótima .....	20
Figura 10 - Indutância em função da frequência, simulação ótima vs não ótima .....	21
Figura 11 – Simulação ótima geometria quadrangular, <i>Cadence (Verilog)</i> .....	22

## Índice de Tabelas

Tabela 1 – Fatores de forma do indutor .....	5
Tabela 2 - Ficheiro técnico do <i>Modelo-<math>\pi</math></i> .....	6
Tabela 3 – Valores iniciais para o cálculo dos restantes parâmetros do <i>Modelo-<math>\pi</math></i> .....	9
Tabela 4 – Limites superior e inferior e valores iniciais das variáveis para otimização do circuito .....	13
Tabela 5 - Parâmetros do <i>Modelo-<math>\pi</math></i> .....	16
Tabela 6 – Parâmetros de desempenho do <i>Modelo-<math>\pi</math></i> .....	16
Tabela 7 – Soluções de otimização .....	16
Tabela 8 - Parâmetros otimizados <i>Modelo-<math>\pi</math></i> .....	17
Tabela 9 - Parâmetros de desempenho otimizados do <i>Modelo-<math>\pi</math></i> .....	17

## Introdução

No âmbito da cadeira de EDA/CAD para Nanoelectrónica, é apresentado o primeiro trabalho laboratorial onde é realizado o estudo e determinação do Modelo- $\pi$ , utilizado para caracterizar bobines integradas presentes na camada superior de metal da tecnologia CMOS, [1].

Este trabalho está dividido em várias etapas de desenvolvimento, de modo a determinar de forma automática os parâmetros do Modelo- $\pi$ , a partir da análise do circuito equivalente da bobine integrada. O estudo do circuito é replicado para as várias geometrias de bobines, assumindo as formas quadrangular, hexagonal e octogonal, estabelecendo um termo de comparação entre os parâmetros obtidos para as diferentes estruturas.

Na primeira parte do projeto é realizado o estudo do Modelo- $\pi$ , determinando os parâmetros tecnológicos do circuito, calculando o fator de qualidade para uma frequência de 1GHz. Após o cálculo dos parâmetros de tecnológicos do circuito, é realizado o estudo da evolução do valor da indutância e do fator de qualidade em função da frequência, determinando a frequência de ressonância do indutor. Esta abordagem é repetida para as várias formas geométricas de bobines (quadrangular, hexagonal e octogonal), consideradas para a operação do circuito.

Seguidamente, é realizada a validação dos dados teóricos calculados, recorrendo ao simulador elétrico software LTspice reproduzindo as curvas características determinadas na análise do circuito, para as respetivas geometrias, [2].

Por último, é feita a otimização do Modelo- $\pi$ , tendo em conta a função de objetivo que melhor determina o comportamento do circuito, maximizando o desempenho do modelo para a geometria quadrangular, [3]. De modo, a confirmar o modelo implementado em *Python* é feita a descrição do circuito em Verilog recorrendo ao software Cadence, simulando o ponto ótimo de funcionamento para a respetiva geometria, [4], [5]. Atendendo aos resultados obtidos, é feita uma análise de sensibilidade do funcionamento ótimo do circuito, referindo quais as geométricas que permitem desempenhar um melhor ponto de operação,

## I-Objetivo

Este projeto tem como objetivo a determinação dos parâmetros do *Modelo- $\pi$*  para um circuito com bobines integradas com diferentes estruturas geométricas, verificando quais as vantagens e desvantagens do *Modelo- $\pi$* , tendo em conta a sua geometria. A Tabela 1 apresenta as diferentes geometrias utilizadas para o dimensionamento do circuito com bobines integradas.

Tabela 1 – Fatores de forma do indutor  
devia vir depois de introduzir as equações para  $L_s$

Estrutura Geométrica	$K1$	$K2$
<b>Quadrangular</b>	2.34	2.75
<b>Hexagonal</b>	2.33	3.82
<b>Octogonal</b>	2.25	3.55

Associadas às contantes de forma do indutor, são considerados os parâmetros que descrevem os componentes tecnológicos, representados na Figura 1, onde a variável  $D_{in}$  representa o diâmetro interno,  $D_{out}$  o diâmetro externo,  $W$  a espessura,  $N$  o número de enrolamentos,  $S$  o espaçamento entre voltas e  $N_{Sides}$  o número de lados da forma geométrica.

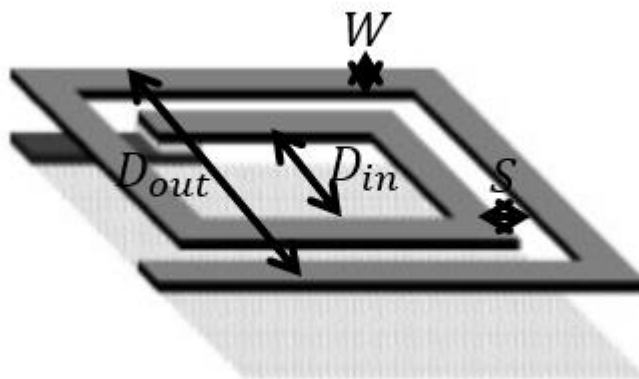


Figura 1 – Esquema ilustrativo do indutor

Posto isto, realizada a descrição do modelo, para ambas as geometrias definidas, estabelecendo uma comparação entre os resultados teóricos calculados e simulados obtidos em *LTspice* e *Cadence*.

## II - Apresentação do *Modelo- $\pi$*

### 2.1. Descrição do *Modelo- $\pi$*

Na primeira parte do trabalho é descrito o *Modelo- $\pi$* , de modo a obter os parâmetros de funcionamento do circuito, tendo em conta o ficheiro tecnológico UMC130mn apresentado em Tabela 2.

Tabela 2 - Ficheiro técnico do *Modelo- $\pi$*

Parâmetros	Valores
$\mu_{iR}$	1
$\mu_i$	$4 \cdot \pi \cdot 1e-7 \cdot \mu_{iR}$
$E_0$ (F/m)	$8.854187817e-12$
$c$ (m/s)	299792458
$t$ (m)	$2.8e-6$
$R_{sheet}$ (Ohm/square)	$10 \cdot 1e-3$
$R_o$ (Ohm.m)	$R_{sheet} \cdot t$
$t_{oxide}$	$5.42e-6$
$T_{M\_underpass}$	$0.4e-6$
$t_{oxide\_Munderpass}$	$t_{oxide} - t_{M\_underpass} - 4.76e-6$
$E_{ox}$	4
$E_{ox}$ (F/m)	$E_0 \cdot E_{ox}$
$E_{sub}$	11.9
$E_{sub}$ (F/m)	$E_0 \cdot E_{sub}$
$T_{sub}$	$700e-6$
$Sub\_resistiv$ (Ohm-m)	2800

Deste modo, o *Modelo- $\pi$*  é caracterizado pelos parâmetros  $L_s$ ,  $R_s$ ,  $C_{ox}$ ,  $C_p$ ,  $R_{sub}$  e  $C_{sub}$ , possibilitando a simulação do ponto de funcionamento do circuito com bobinas integradas. Assim, o sistema equações (2.1), representa o cálculo dos parâmetros para a obtenção da indutância série do circuito, onde esta depende das dimensões e forma geométrica do componente.

$$\left\{ \begin{array}{l} d_{in} = d_{out} - 2nw - 2(n-1)s \\ d_{avg} = \frac{d_{out} + d_{in}}{2} \\ \rho_x = \frac{d_{out} - d_{in}}{d_{out} + d_{in}} \\ L_s = \frac{n^2 * d_{avg} * k_1 * \mu_0}{(1 + k_s * \rho_x)} \end{array} \right. \quad (2.1)$$

O sistema de equações (2.2), ilustra do cálculo da determinação do valor da resistência série do circuito, sabendo que esta depende da frequência de entrada do sinal, assim como das características do ficheiro tecnológico.

$$\left\{ \begin{array}{l} l = N_{side} + d_{avg} * n * \tan\left(\frac{\pi}{N_{side}}\right) \\ t_{eff} = 1 - e^{\frac{-t}{\delta}} \\ \delta = \sqrt{\frac{\rho}{\pi * f * \mu}} \\ R_s = \frac{l * \rho}{w * \delta * t_{eff}} \end{array} \right. \quad (2.2)$$

significado das variáveis ??

Para o cálculo da capacidade em paralelo, é tido em conta o sistema de equações (2.3), que relaciona a espessura do óxido entre os dois níveis de metal  $t_{oxM1-M2}$ , número sobreposições metálicas  $n_c$  e número de voltas, dados presentes no ficheiro tecnológico.

$$\left\{ \begin{array}{l} C_p = n_c * w^2 * \frac{\epsilon_{ox}}{t_{oxM1-M2}} \\ n_c = n - 1 \end{array} \right. \quad (2.3)$$

Por último, para a determinação dos parâmetros  $C_{ox}$ ,  $R_{sub}$  e  $C_{sub}$  são definidas equações (2.4) a (2.6) que descrevem o valor dos respetivos parâmetros, onde as variáveis dependentes de cada equação está inicializada no ficheiro tecnológico utilizado.

$$C_{ox} = \frac{l * w * \epsilon_{ox}}{2 * t_{ox}} \quad (2.4)$$

$$C_{sub} = \frac{l * w * \epsilon_{sub}}{2 * t_{sub}} \quad (2.5)$$

$$R_{sub} = \frac{Sub_{resistiv}}{t * w} \quad (2.6)$$

Após a obtenção dos parâmetros de entrada do *Modelo- $\pi$* , é realizada a metodologia de estudo do respetivo circuito, de modo a verificar o seu comportamento para as diferentes geométricas utilizadas, sendo representado na Figura 2 o esquemático do *Modelo- $\pi$* .

o modelo devia ser apresentado antes das expressões para cálculo dos componentes

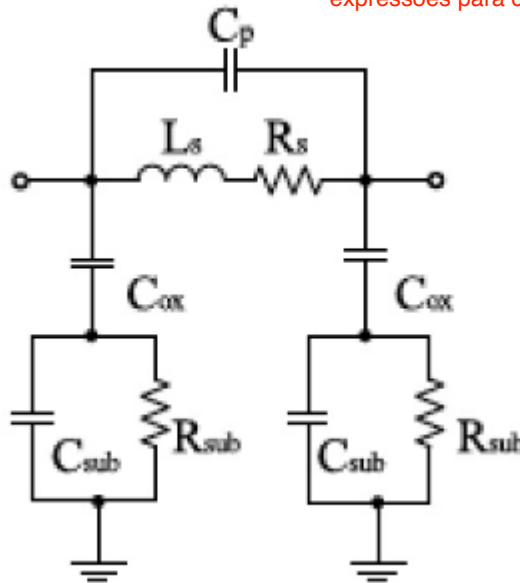


Figura 2 – Esquema do *Modelo- $\pi$*



### III - Metodologia

#### 3.1. Determinação e caracterização dos parâmetros do *Modelo- $\pi$*

Nesta secção é feita a descrição da metodologia utilizada para a implementação do *Modelo- $\pi$* , recorrendo ao software *Spyder-Python* para a simulação do ponto de funcionamento do circuito. Deste modo, a partir do ficheiro tecnológico fornecido é realizada a análise do circuito, considerando um dos terminais ligado a ground, assumindo como valores fixos os parâmetros característicos do indutor, dos quais o diâmetro externo, espessura, espaçamento entre voltas, número de voltas e frequência. Estes valores estão representados na Tabela 3.

Tabela 3 – Valores iniciais para o cálculo dos restantes parâmetros do *Modelo- $\pi$*

$D_{out}$ (nm)	$W$ (nm)	$S$ (nm)	$n$ (voltas)	$f$ (GHz)
340	100	5	3	1

Tendo em consideração a descrição do *Modelo- $\pi$* , são determinados os valores para o diâmetro interior e médio,  $D_{in}$  e  $D_{avg}$ , respetivamente, da indutância  $L_s$ , das resistências  $R_s$  e  $R_{sub}$ , das capacidades  $C_p$ ,  $C_{ox}$  e  $C_{sub}$ , da impedância  $\bar{Z}$ , da indutância equivalente  $L$ , do factor de qualidade  $Q_{factor}$  e da frequência de ressonância  $F_{res}$  obtidos através da função *parametros(freq, i, n\_turns, space, width, dout)*. Esta função realiza as operações a seguir descritas.

Inicialmente calcula o valor do diâmetro interior  $D_{in}$ , através da função *calc\_d\_in(dout, n\_turns, S, width)*. Caso o valor de  $D_{in}$  seja inferior a  $30\ \mu m$ , o diâmetro interior é forçado a  $30\ \mu m$ . De seguida é calculado o diâmetro médio através da função *calc\_d\_avg(Din, dout)*, sendo necessário determinar o valor de  $\rho_x$ , operação realizada pela função *calc\_rox(Din, dout)*. Calculados estes valores obtém-se o valor de  $L_s$  a partir da função *calc\_Ls(n\_turns, Davg, tab1['K1'][i], tab1['K2'][i], Rox, miu)*. Os cálculos realizados por estas funções estão descritos no sistema de equações (2.1), referentes ao cálculo dos parâmetros da indutância série apresentado no Capítulo II.

O cálculo de  $R_s$ , é estabelecido pela função *calc\_Rs(l, width, Delta, T\_eff)*, onde é necessário determinar os parâmetros  $\delta$ ,  $t_{eff}$  e  $l$ , recorrendo às funções *calc\_delta(freq, miu)*, *calc\_t\_eff(Delta)* e *calc\_l(tab1['N Sides'][i], Davg, n\_turns)*, respetivamente. Estes cálculos são realizados através do sistema de equações (2.2), descrito no Capítulo II.

As capacidades parasitas e do óxido,  $(C_p, C_{ox})$ , cujo cálculo pode ser realizado através do sistema de equações (3.1), foram obtidas através das funções  $calc\_Cp(n\_turns, width)$  e  $calc\_Cox(l, width)$ .

$$\begin{cases} C_p = (n - 1) * w^2 * \frac{\epsilon_{ox}}{t_{oxM1-M2}} \\ C_{ox} = \frac{l * w * \epsilon_{ox}}{2 * t_{ox}} \end{cases} \quad (3.1)$$

Os valores da capacidade e resistência do substrato,  $(C_{sub}, R_{sub})$ , são calculadas a partir das equações (2.5) e (2.6) referente ao Capítulo II, formulando a função  $calc\_Csub(l, width)$  e  $calc\_Rsub(l, width)$ .

Finalmente, os parâmetros referentes aos valores da impedância  $\bar{Z}$ , da indutância equivalente  $L$ , do factor de qualidade  $Q_{factor}$  e da frequência de ressonância  $F_{res}$  são obtidos através das funções  $calc\_Z(freq, Ls, Rs, Cp, Cox, Csub, Rsub)$ ,  $calc\_L(Z, freq)$ ,  $calc\_QF(Z)$  e  $calc\_Fres(Ls, Cp)$ , através do sistema de equações (3.2).

$$\left\{ \begin{array}{l} \bar{Z} = \frac{\bar{Z}_1 \cdot \bar{Z}_2}{\bar{Z}_1 + \bar{Z}_2} \\ \bar{Z}_1 = \bar{Z}_{par} + \bar{Z}_{Cox} \\ \bar{Z}_{par} = \frac{\bar{Z}_{Csub} \cdot R_{sub}}{\bar{Z}_{Csub} + R_{sub}} \\ \bar{Z}_{Csub} = \frac{1}{j\omega C_{sub}} \\ \bar{Z}_{Cox} = \frac{1}{j\omega C_{ox}} \\ \bar{Z}_2 = \frac{\bar{Z}_{Cp} \cdot \bar{Z}_{serie}}{\bar{Z}_{Cp} + \bar{Z}_{serie}} \\ \bar{Z}_{Cp} = \frac{1}{j\omega C_p} \\ \bar{Z}_{serie} = R_{sub} + j\omega L_s \\ L = \frac{Im\{\bar{Z}\}}{2\pi f} \\ Q_{factor} = \frac{Im\{\bar{Z}\}}{Re\{\bar{Z}\}} \\ F_{res} = \frac{1}{2\pi\sqrt{L_s C_p}} \end{array} \right. \quad (3.2)$$

Os valores resultantes destas operações foram armazenados numa dataframe denominada *params*. Uma vez que os valores de  $R_s$  variam com a frequência, são calculados os valores de  $R_s$ ,  $\bar{Z}$ ,  $L$  e  $Q_{factor}$  para o intervalo de frequências de 1Mhz a 100Ghz. Os vetores  $R_{vec}$ ,  $L_{vec}$ ,  $Q_{vec}$  e  $F_{vec}$  armazenam os valores para os respetivos parâmetros, tendo em consideração a variação da frequência de funcionamento do circuito. Em anexo encontra-se disponível o código referente à implementação do *Modelo- $\pi$* .

### 3.2. Simulação LT Spice

Com base nos parâmetros obtidos para o funcionamento do circuito a uma frequência de 1Ghz, é simulado o circuito da Figura 3 no *LT Spice* considerando o intervalo de frequências de 1Mhz a 100GHz. Após a simulação em *LT Spice*, são extraído os valores da indutância, com o objetivo de realizar uma comparação com os valores de indutância obtidos através do *Spyder*.

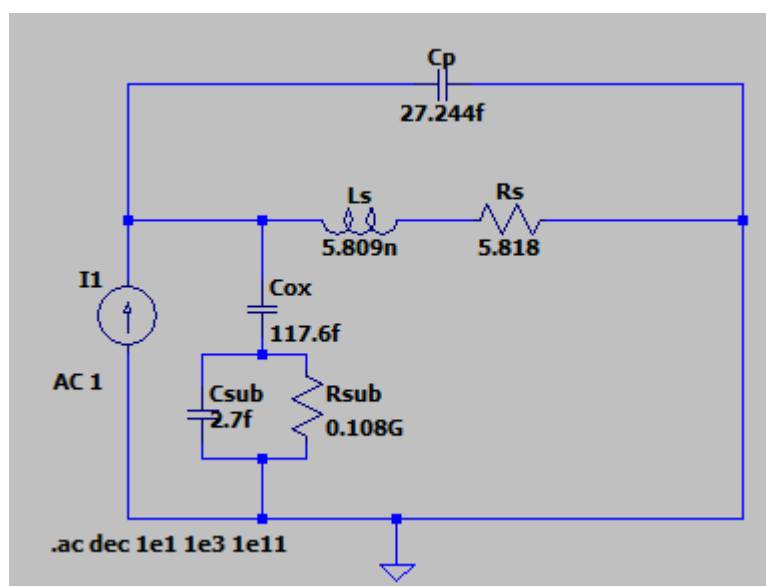


Figura 3 - Esquema em LTSpice do *Modelo- $\pi$*  simulado

Os valores resultantes foram armazenados em dataframes do tipo `Lt_Shape`, onde `Shape` se refere à forma da bobina. Na Figura 4 é representada a estrutura definida para o armazenamento de dados.

Index	F	L
0	1000	5.80412122356140e-009
1	1258.93	5.80413617659779e-009
2	1584.89	5.80415954217598e-009
3	1995.26	5.80419606574080e-009
4	2511.89	5.80425253976546e-009
5	3162.28	5.80433864051418e-009
6	3981.07	5.80446720867505e-009
7	5011.87	5.80465359518927e-009
8	6309.57	5.80491150282167e-009
9	7943.28	5.80524757082988e-009

Figura 4 – Dataframe do tipo `Lt_Shape`

### 3.3. Otimização

Nesta secção é realizada a descrição da metodologia utilizada para otimização do *Modelo- $\pi$* . Assim, recorrendo à função de otimização *minimize* (), são determinados os valores ótimos de  $D_{out}$ ,  $n$  e  $W$  para uma bobina quadrada de 6,5 nH, a uma frequência de 2,4 GHz. Os valores iniciais e os limites destes parâmetros podem ser vistos na Tabela 4.

Tabela 4 – Limites superior e inferior e valores iniciais das variáveis para otimização do circuito

Vetores definidos	$D_{out}$ ( $\mu\text{m}$ )	$n$ (voltas)	$W$ ( $\mu\text{m}$ )
Limite inferior	200	<u>1.5</u>	10
Ponto inicial x0	400	2	20
Limite superior	1200	8	60

De forma a tornar os valores otimizados o mais coerentes possível, são definidas ainda restrições para os valores de  $D_{in}$  e  $F_{res}$ , sendo que no mínimo terão que ser 20  $\mu\text{m}$  e 24 GHz, respetivamente.

Deste modo, os variáveis de decisão definidas são determinadas através da função `minimize(get_L, x0, method='SLSQP', bounds=boundx, constraints=({'type': 'ineq', 'fun': get_din_min}, {'type': 'ineq', 'fun': get_fres}))`.

Assim sendo, uma vez obtidos os novos valores de  $D_{out}$ ,  $n$  e  $W$ , são calculados os restantes parâmetros  $D_{in}$ ,  $D_{avg}$ ,  $L_s$ ,  $R_s$ ,  $C_p$ ,  $C_{ox}$ ,  $C_{sub}$ ,  $R_{sub}$ ,  $\bar{Z}$ ,  $L$ , e  $Q_{factor}$  e  $F_{res}$  através da função `parametros(freq, i, n_turns, space, width, dout)`. Estes valores são armazenados na dataframe *params2*.

De notar que, os valores de  $R_s$  variam com a frequência, sendo necessário recorrer aos vetores *Rvec*, *Lvec*, *Qvec* e *Fvec* para o registo dos novos parâmetros, com o objetivo de realizar uma comparação com os valores não otimizados.

### 3.4. Cadence – Verilog e simulações

Inicialmente, através do software *Cadence*, realizou-se a descrição do *Modelo- $\pi$*  para a forma quadrada, utilizando os parâmetros obtidos pela otimização descrita na secção anterior e calculando os valores dos restantes parâmetros, recorrendo à linguagem *Verilog*. Uma vez obtidos todos os valores que caracterizam o modelo, é feita a simulação AC representada na Figura 5. A fonte de corrente utilizada tem uma corrente de 1A para que se estabeleça a relação

$V = I\bar{Z} = \bar{Z}$ . Da simulação realizada são extraídos os gráficos da indutância e do factor de qualidade. O código desenvolvido em *Verilog* pode ser consultado em Anexos.

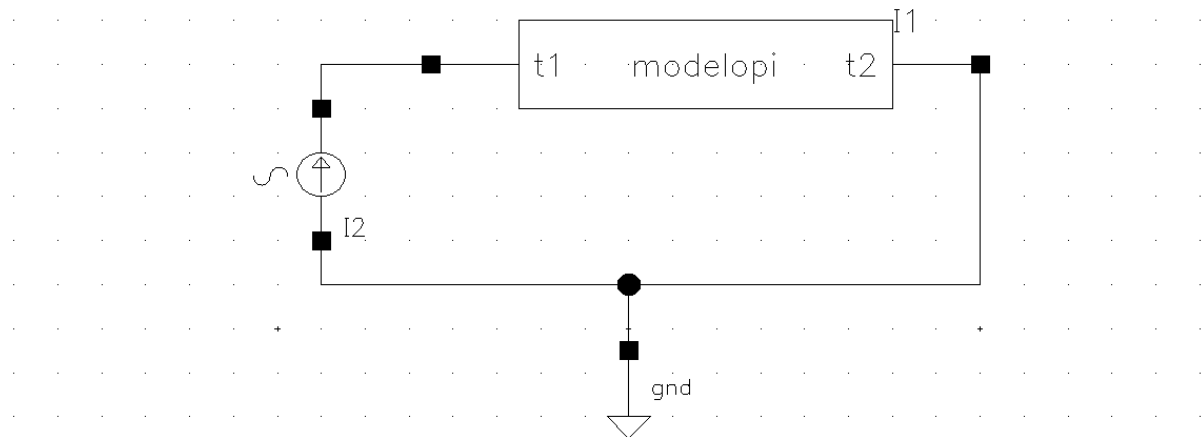


Figura 5 - Circuito para a simulação do *Modelo-π* em *Cadence*

### 3.5. Gráficos e Erros

Por último, são realizados os seguintes gráficos, de modo a reproduzir uma análise de sensibilidade do *Modelo-π* para os diferentes cenários de estudo considerados:

- Valores da indutância  $L$  em função da frequência obtidos em Python vs simulados em LT Spice.
- Erro relativo dos valores da indutância  $L$  obtidos em Python em relação aos obtidos em LT Spice
- Valores do fator de qualidade  $Q_{factor}$  em função da frequência obtidos em Python.
- Valores da indutância  $L$  em função da frequência não otimizados vs otimizados.
- Valores do fator de qualidade  $Q_{factor}$  em função da frequência não otimizados vs otimizados.

Para os gráficos em que se comparam os valores medidos com os valores modelados, são calculados os valores dos erros relativos, através da equação (3.6). Todos os gráficos referidos estão representados no Capítulo IV.

$$Erro (\%) = \frac{Valor\ modelado - Valor\ medido}{Valor\ medido} * 100 \quad (3.3)$$

Para os gráficos do fator de qualidade não se utilizaram os valores obtidos em *LT Spice*, uma vez que estas simulações fixam o valor de  $R_S$ , resultando em termos de comparação diferentes.

Relativamente à metodologia apresentada, foi realizado o estudo do modelo para as diferentes formas geométricas da bobine, Quadrada, Hexagonal e Octagonal, com exceção da metodologia de otimização, onde apenas se realizou o procedimento para a forma Quadrada. O código desenvolvido para estudo do *Modelo- $\pi$*  apresentado neste Capítulo, está disponível em Anexos.

## IV - Resultados

### 4.1. Parâmetros Obtidos

Aplicando a metodologia apresentada no Capítulo III, foram obtidos os seguintes parâmetros de entrada para o funcionamento do circuito, tendo em consideração a região de funcionamento definida para a simulação do *Modelo- $\pi$* . As Tabelas 5 e 6 apresentam os resultados simulados do modelo em estudo.

Tabela 5 - Parâmetros do *Modelo- $\pi$*

Forma Geométrica	$D_{in}$ ( $\mu\text{m}$ )	$D_{avg}$ ( $\mu\text{m}$ )	$L_s$ (nH)	$R_s$ ( $\Omega$ )	$C_p$ (fF)	$C_{ox}$ (fF)	$C_{sub}$ (fF)	$R_{sub}$ (M $\Omega$ )
Quadrada	260	300	5.81	5.81	27.2	118	2.71	109
Hexagonal	260	300	5.24	5.03	27.2	102	2.35	126
Octagonal	260	300	5.18	4.81	27.2	97.4	2.24	131

Tabela 6 – Parâmetros de desempenho do *Modelo- $\pi$*

Forma Geométrica	$\bar{Z}$	$L$ (nH)	$Q_{factor}$	$F_{res}$ (GHz)
Quadrada	(5.89+36.74j)	5.85	6.22	12.7
Hexagonal	(5.10+33.10j)	5.27	6.49	13.3
Octagonal	(4.87+32.74j)	5.21	6.71	13.4

Atendendo à metodologia de otimização desenvolvida e descrita no Capítulo III, foram registados os seguintes parâmetros de entrada referentes ao *Modelo- $\pi$* . As Tabelas 7, 8 e 9 ilustram os valores obtidos através da simulação do processo de otimização.

Tabela 7 – Soluções de otimização

$D_{out}$ ( $\mu\text{m}$ )	$n$ (voltas)	$W$ ( $\mu\text{m}$ )
1023	1.5	10



Tabela 8 - Parâmetros otimizados *Modelo- $\pi$*

Forma Geométrica	$D_{in}$ ( $\mu\text{m}$ )	$D_{avg}$ ( $\mu\text{m}$ )	$L_s$ (nH)	$R_s$ ( $\Omega$ )	$C_p$ (fF)	$C_{ox}$ (fF)	$C_{sub}$ (fF)	$R_{sub}$ (M $\Omega$ )
Quadrada	988	1006	6.35	12.17	6.88	1.98	4.56	64.6

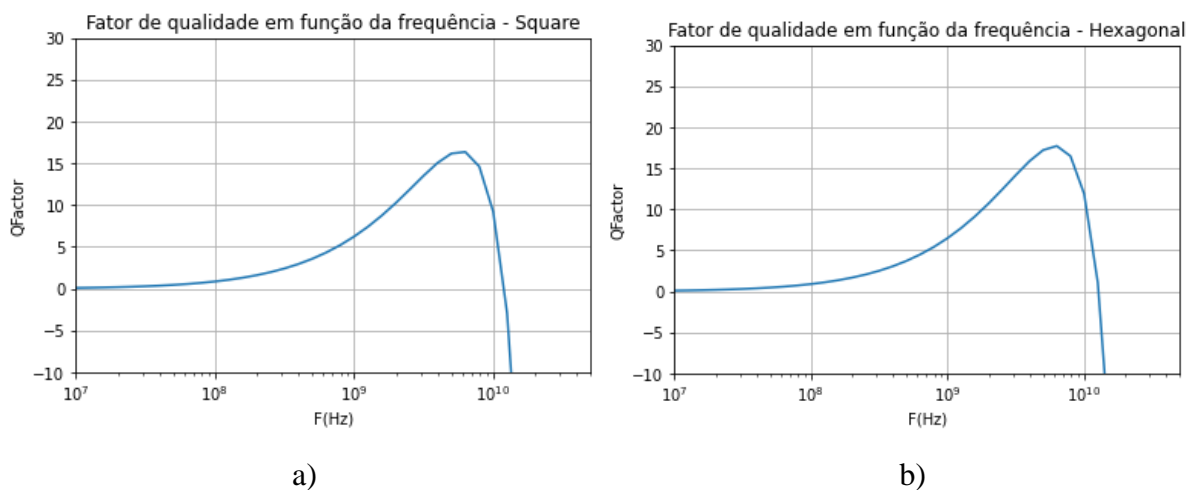
Tabela 9 - Parâmetros de desempenho otimizados do *Modelo- $\pi$*

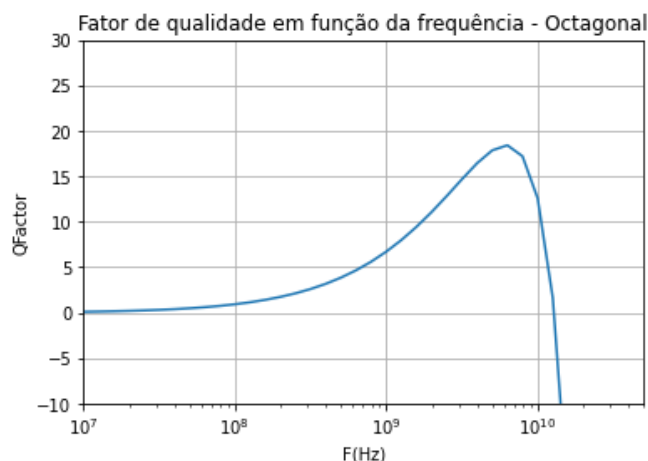
Forma Geométrica	$\bar{Z}$	$L$ (nH)	$Q_{factor}$	$F_{res}$ (GHz)
Quadrada	(12.58 + 97.31j)	6.45	7.74	24.1

## 4.2. Análise de Resultados

Neste Capítulo são apresentados e analisados os resultados obtidas das diferentes simulações realizadas para o estudo do *Modelo- $\pi$* , com recurso ao software *Spyder*.

Primeiramente é executada a simulação referente ao estudo do *Modelo- $\pi$*  para um intervalo de frequência de 1Mhz a 100Ghz, obtendo as curvas características do fator e qualidade e indutância em função da frequência de amostragem, aplicada às diferentes formas geométricas do indutor. Deste forma, de modo averiguar o comportamento do modelo para as diferentes estruturas geométricas, é estabelecido um termo de comparação entre as diferentes curvas obtidas, referentes ao mesmo parâmetro em análise. Assim sendo, na Figura 6 são representados os gráficos referentes ao fator de qualidade em função da frequência, para as respetivas formas geométricas.





c)

Figura 6 – Fator de Qualidade em função da frequência a) Quadrangular; b) Hexagonal; c) Octagonal

Analisando os gráficos do fator de qualidade em função da frequência para as diferentes geometrias, é visível que o seu comportamento é semelhante para as várias estruturas, dado que a fator de qualidade aumenta não linearmente com a frequência, porém para valores muito elevados de frequência o fator de qualidade diminui abruptamente. Também é possível constatar que para estruturas com maior número de lados o fator de qualidade atinge um ponto extremo da função maior (máximo). Este facto deve-se à diminuição da componente imaginária e real da impedância do circuito (Secção 4.1), com o aumento do número lados da estrutura considerada, destacando a coerência dos resultados com a descrição do modelo.

Tendo em conta os resultados obtidos do fator de qualidade em função da frequência são apresentados os gráficos com as curvas características que expressam a relação entre a indutância e a frequência, para as formas geométricas consideradas. Assim, uma vez executada a simulação do mesmo circuito em *LT Spice*, são representados os gráficos da indutância em função da frequência para as duas simulações realizadas, através da Figura 7. Deste modo, podemos estabelecer um termo de comparação entre os resultados obtidos do *Modelo- $\pi$*  formulado em *Python* com os dados simulados pelo software *LT Spice*.

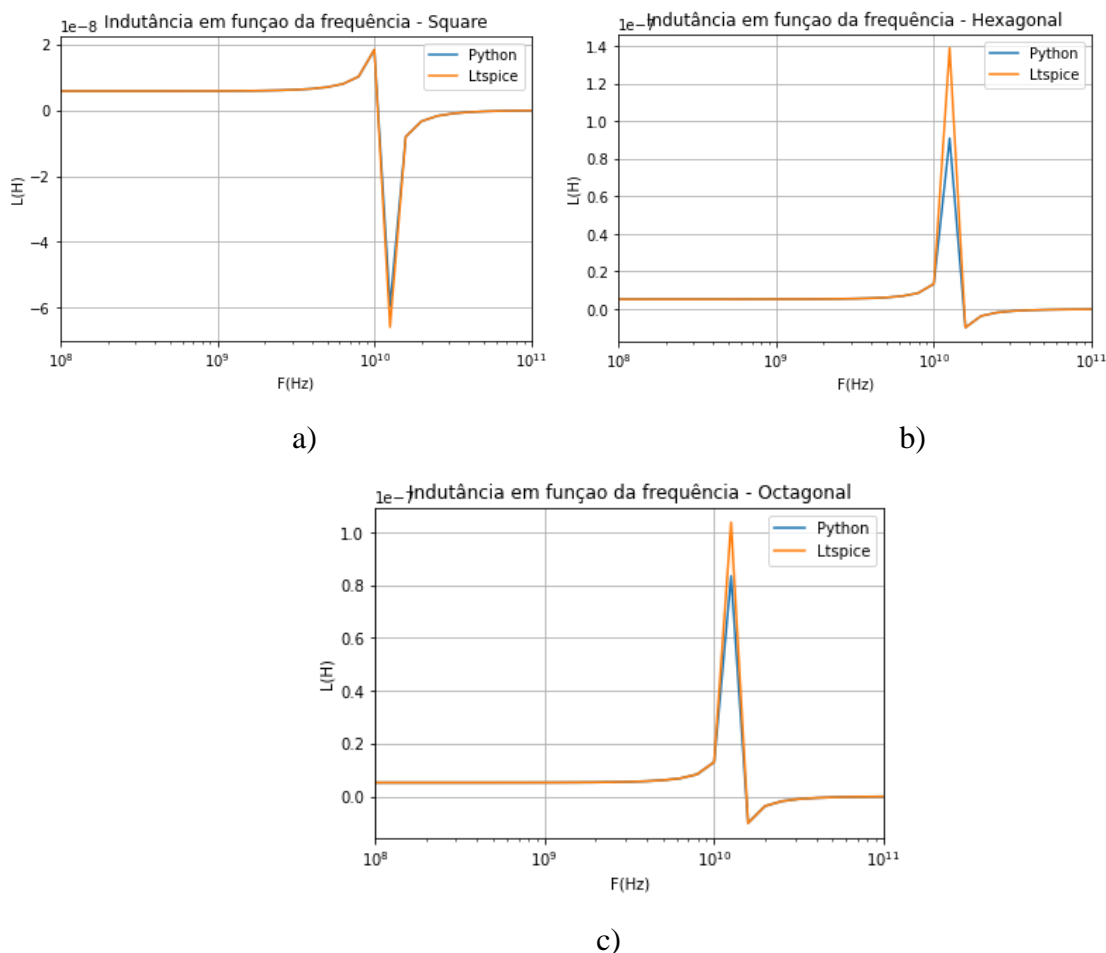
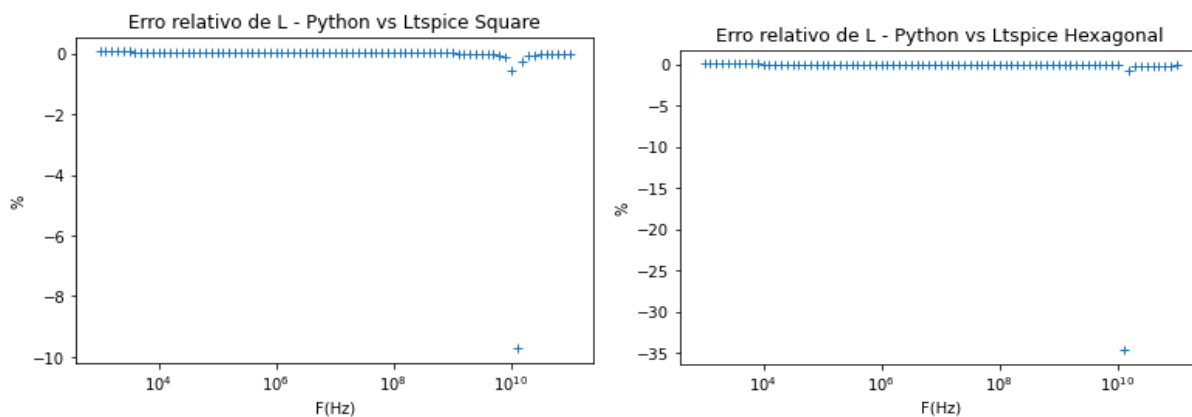


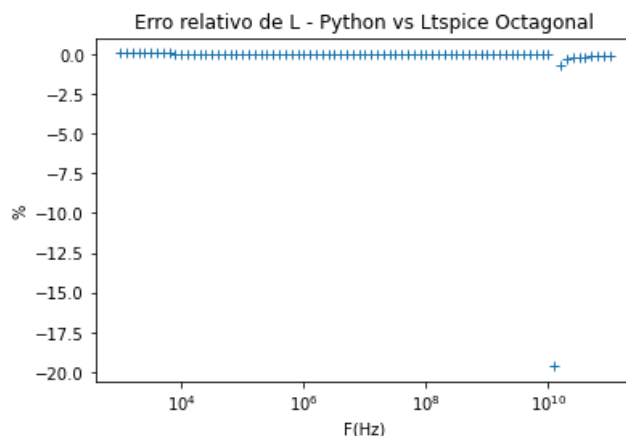
Figura 7 - Indutância em função da frequência a) Quadrangular; b) Hexagonal; c) Octagonal

Através dos gráficos da indutância em função da frequência, podemos averiguar que ambas as simulações coincidem com um elevado grau de precisão, para as três formas geométricas. Deste modo, é validado o *Modelo- $\pi$*  descrito anteriormente. Esta afirmação pode ser suportada através dos gráficos do erro relativo entre os dois métodos de simulação. A Figura 8, apresenta os gráficos do erro relativo para as diferentes geometrias.



a)

b)



c)

Figura 8 – Erro relativo em função da frequência a) Quadrangular; b) Hexagonal; c) Octagonal

Recorrendo aos gráficos da Figura 8, é visível que o erro é aproximadamente nulo até atingirmos a frequência de ressonância, onde o erro sofre uma oscilação significativa, convergindo novamente para zero.

Após os estudos e análise do circuito para os parâmetros de funcionamento definidos, é realizado a otimização do *Modelo- $\pi$* , com o objetivo de atingir uma melhor performance de funcionamento do circuito. Desta forma, aplicando metodologia descrita no Capítulo III referente à otimização, são obtidos os gráficos do fator de qualidade e indutância em função da frequência. Na Figura 9 é ilustrado o gráfico do fator de qualidade em função da frequência para a forma geométrica quadrangular, demonstrando o ponto de funcionamento do circuito otimizado e não otimizado.

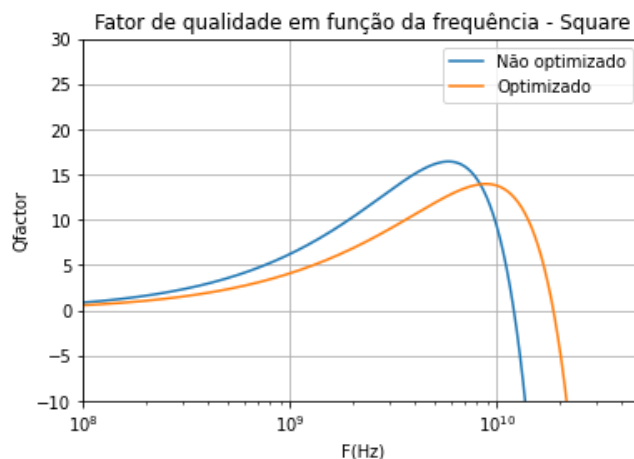


Figura 9 - Fator de Qualidade em função da frequência, simulação ótima vs não ótima

Analisando o da Figura 9, é notório que a simulação do circuito para parâmetros ótimos determinados é bem realizada, uma vez que o fator de qualidade otimizado permite garantir uma maior região de qualidade de funcionamento do circuito para uma maior gama de frequência. De notar que, o aumento do fator de qualidade ocorre dado que o valor da impedância do circuito altera significativamente em comparação com os dados normais simulados, onde a parte real da impedância diminui em relação à componente imaginária, (Secção 4.1). Assim sendo, reproduzindo a mesma comparação para os valores da indutância em função da frequência, é notório que a indutância diminui ligeiramente, tendo em conta os parâmetros ótimos determinados pelo algoritmo, apresentados na Secção 4.1. A Figura 10 representa as curvas da indutância em função da frequência para a estrutura quadrangular, considerando os dados obtidos das duas metodologias realizadas.

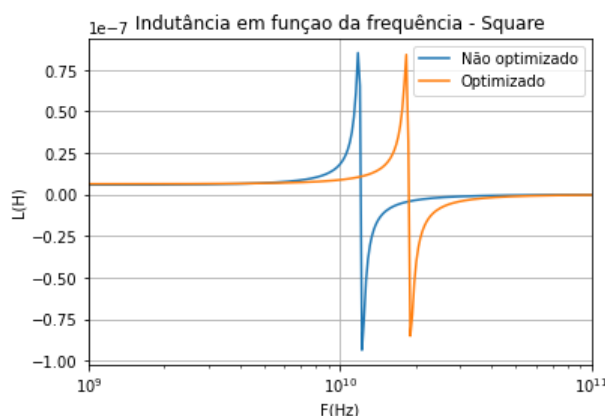


Figura 10 - Indutância em função da frequência, simulação ótima vs não ótima

Observando o gráfico da Figura 10, é ainda possível destacar que a indutância se mantém para a solução ótima determinada, e a frequência de ressonância aumenta comparativamente às duas simulações executadas. De verificar também, que o valor da indutância 6.5nH é assegurada com alguma precisão face à frequência de ressonância determinada pelo algoritmo de otimização. Analisando a solução ótima podemos constatar que o valor do diâmetro exterior do indutor tende a atingir o máximo valor da região de busca definida, enquanto o número de voltas e a largura assumem o mínimo da região de soluções admissíveis. <sup>porquê??</sup> Esta discrepância entre os valores da solução ótima obtidas é o que possibilita a otimização do *Modelo- $\pi$* . Determinada a solução de ótima do sistema, é possível verificar que as restrições impostas ao problema de otimização são garantidas. Deste modo, é possível assegurar a coerência da solução para a implementação do modelo, uma vez que a frequência de ressonância e o diâmetro interior são superiores aos limites estabelecidos de 20 nm e 24 GHz respetivamente, dados apresentados na Secção 4.1.

Por último, são efetuadas as simulações referentes ao software *Cadence*, implementado a metodologia apresentada na Secção 3.4. Após a simulação deste método com os parâmetros obtidos da otimização do circuito, é notório que o circuito assegura uma impedância próxima do valor pretendido (6.5nH) atingindo um valor de 6.45nH, para uma frequência de 2.4GHz. Também podemos constatar que o valor da indutância é garantido para um grande intervalo de frequência, evitando a operação o circuito para a região onde o fator de qualidade assume uma maior variância de valores. Contudo, a frequência de ressonância obtida é de aproximadamente 20GHz, afastando-se ligeiramente do valor calculado pela simulação de otimização. Deste modo, é possível constatar que a incerteza associada a esta simulação quando comparada com os dados obtidos do *Python*, deve-se a erros de aproximação numérica, inicializados na codificação em *Verilog*. Na Figura 11 está representado os resultados obtidos da simulação ótima do circuito recorrendo ao software *Cadence*.

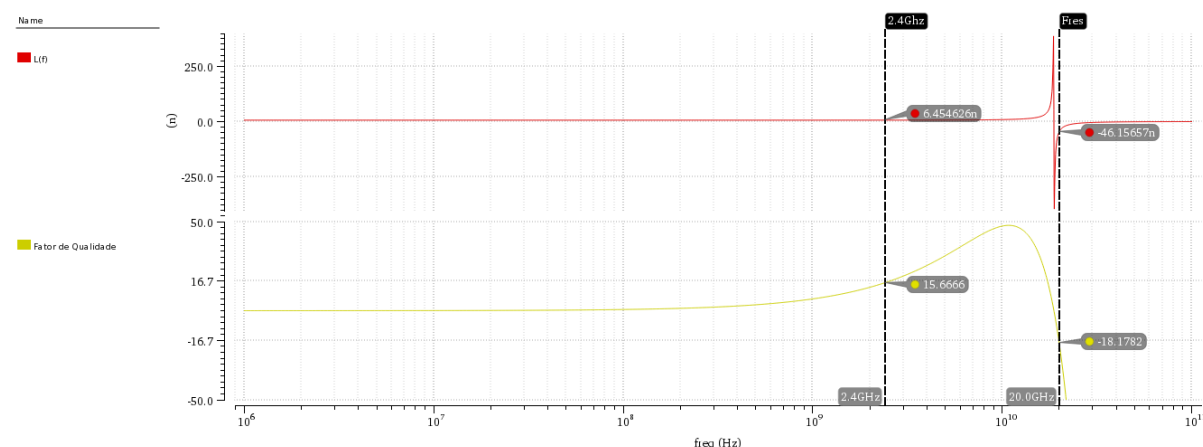


Figura 11 – Simulação ótima geometria quadrangular, *Cadence* (*Verilog*)

## V - Conclusões

Com a realização deste trabalho laboratorial foram abordados vários conceitos que nos permitem retirar algumas conclusões em cada uma das partes de desenvolvimento do projeto, descrevendo e analisando o funcionamento do Modelo- $\pi$  para uma bobine integrada.

Assim sendo, inicialmente realizou-se a descrição do modelo teórico do circuito no *Python*, onde foram determinados os valores de  $L_s$ ,  $R_s$ ,  $C_p$ ,  $C_{ox}$ ,  $C_{sub}$ , e  $R_{sub}$ . De seguida foi calculada a indutância, o fator qualidade e a frequência de ressonância do circuito dependendo do número de lados da geometria do indutor, aí concluiu-se que, com o aumento do número de lados diminui a indutância do circuito, mas aumenta o Q-Factor e a frequência de ressonância.

Em terceiro lugar foi simulado o circuito no *LT Spice* considerando os parâmetros obtidos inicialmente no *Python* e foram conseguidos os valores da indutância e do fator de qualidade em função da frequência. De realçar que apenas se pode comparar os gráficos simulados da indutância do circuito uma vez que a simulação *LT Spice* não contempla a variação da resistência  $R_s$  com a frequência. Deste modo comparando os gráficos da indutância em função da frequência verifica-se que o erro relativo é praticamente nulo, validando o modelo implementado para a descrição do circuito.

Por último, utilizando funções de otimização no *Python* foram determinados os parâmetros ótimos para um indutor quadrangular de 6.5nH de indutância a operar a uma frequência de 2.4GHz. Foram calculados os valores do fator de qualidade e da frequência de ressonância do circuito anterior. Conclui-se que o processo de otimização tende a selecionar um baixo número de voltas, maximizando o diâmetro exterior e minimizando a largura do indutor para que sejam garantidas as condições de operação definidas, respeitando as restrições impostas. Estes dados foram confirmados segundo o software *Cadence* com recurso do modelo descrito em *Verilog*, validando o processo de otimização, assim como a simulação do circuito realizado em *Python*.

## Bibliografia

- [1] E. D. A. Cad, P. Nanoelectrónica, M. H. Fino, E. D. A. Cad, and P. Nanoelectrónica, “Eda/cad para nanoelectrónica 2020/2021,” 2021.
- [2] E. D. A. Cad, P. Nanoelectrónica, M. H. Fino, E. D. A. Cad, and P. Nanoelectrónica, “How to measure inductance in LTSpice.”
- [3] U. Method, “A brief introduction to Python Optimization” pp. 1–5.
- [4] “VerilogA1.” .
- [5] K. S. Kundert, *THE DESIGNER ’ S GUIDE TO VERILOG-AMS THE DESIGNER ’ S GUIDE BOOK SERIES Consulting Editor*. 2004.



## Anexos

### Anexo A: Código do Modelo- $\pi$ Python

```
# -*- coding: utf-8 -*-
"""
Spyder Editor

This is a temporary script file.
"""

import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import minimize
# from scipy.optimize import curve_fit
os.chdir('G:\\O meu disco\\MIEEC\\5º Ano\\2º Semestre\\EDA\\Trabalho3\\LT')

#####
##
##### Tech file UMC130nm #####

miuR=1;
miu=4*np.pi*1e-7*miuR;
E0=8.854187817e-12;          #F/m
c=299792458;                #m/s

# Layer Oxide (Metal)

t=2.8e-6;                    #Metal thickness (m)
Rsheet=10*1e-3;              #Sheet Resistance [Ohm/square] (= Ro/t)
Ro=Rsheet*t;                 #Metal Resistance [Ohm.m]
toxide=5.42e-6;              #Dielectric (Oxide) Thickness (from metal to substrate)
t_M_underpass=0.4e-6;
toxide_Munderpass=toxide-t_M_underpass-4.76e-6;
Erox=4;                      #Oxide Er
Eox=E0*Erox;                 #Oxide permitivity %Substrate
Ersub=11.9;                  #substrate Er
Esub=E0*Ersub;               #substrate permitivity
Tsub=700e-6;                 #substrate thickness
Sub_resistiv=2800;           #28 ohm-cm ou 2800 Ohm-m

#####
##
#####
##
```

```
#####  
##  
##### Carregamento de Dados #####  
Lt_Quad=pd.read_csv('Quad.txt',sep='\t')  
Lt_Quad.columns=['F','L']  
Lt_Quad[['L','lixo']] = Lt_Quad['L'].str.split(',',expand=True)  
Lt_Quad.drop('lixo', axis='columns', inplace=True)  
Lt_Square=Lt_Quad['L'].astype(float)  
  
Lt_Hexa=pd.read_csv('Hexa.txt',sep='\t')  
Lt_Hexa.columns=['F','L']  
Lt_Hexa[['L','lixo']] = Lt_Hexa['L'].str.split(',',expand=True)  
Lt_Hexa.drop('lixo', axis='columns', inplace=True)  
Lt_Hexagonal=Lt_Hexa['L'].astype(float)  
  
Lt_Octa=pd.read_csv('Octa.txt',sep='\t')  
Lt_Octa.columns=['F','L']  
Lt_Octa[['L','lixo']] = Lt_Octa['L'].str.split(',',expand=True)  
Lt_Octa.drop('lixo', axis='columns', inplace=True)  
Lt_Octogonal=Lt_Octa['L'].astype(float)  
  
tab1={'Shape':['Square', 'Hexagonal', 'Octagonal'],  
      'N Sides':[4,6,8],  
      'K1':[2.34, 2.33, 2.25],  
      'K2':[2.75, 3.82, 3.55]}  
  
#####  
##  
#####  
##  
  
#####  
##  
##### Definição de funções #####  
  
def calc_Ls(n, d_avg, k1, k2, rox, miu0):  
    return (n**2)*d_avg*k1*miu0 / (1 + k2*rox)  
  
def calc_d_out(d_in, n, s, w):  
    return d_in + 2*n*w + 2*(n-1)*s  
  
def calc_d_in(d_out, n, s, w):  
    return d_out - 2*n*w - 2*(n-1)*s  
  
def calc_d_avg(d_in, d_out):  
    return 0.5*(d_in + d_out)  
  
def calc_rox(d_in, d_out):  
    return (d_out - d_in)/(d_out + d_in)
```

```
#RS VARIA COM F
def calc_Rs(l, w, delta, t_eff):
    return l*Ro/(w*delta*t_eff)

def calc_delta(f, miux):
    return np.sqrt(Ro / (np.pi*f*miux))

def calc_t_eff(delta):
    return 1 - np.exp(-t/delta)

def calc_l(inside, davg, n):
    return inside * davg * n * np.tan(np.pi/inside)

def calc_Cp(n, w):
    return (n-1) * w**2 * Eox/toxide_Munderpass

def calc_Cox(l, w):
    return 0.5 * l * w * Eox/toxide

def calc_Csub(l, w):
    return 0.5 * l * w * Esub/Tsub

def calc_Rsub(l, w):
    return 2 * Tsub * Sub_resistiv / (l * w)

def calc_Z(f, Ls, Rs, Cp, Cox, Csub, Rsub):

    ZCsub = -1j/(2*np.pi*f*Csub)

    Zpar = ZCsub * Rsub / (ZCsub + Rsub )

    ZCox = -1j/(2*np.pi*f*Cox)

    ZCp = -1j/(2*np.pi*f*Cp)

    ZSerie = Rs + 1j*2*np.pi*f*Ls

    Z1 = Zpar + ZCox

    Z2 = ZCp * ZSerie / (ZCp + ZSerie)

    Z = Z1 * Z2 / (Z1 + Z2)

    return Z

def calc_L(Z, f):
    return Z.imag/(2*np.pi*f)

def calc_QF(Z):
    return Z.imag/Z.real
```

```
def calc_Fres(L,C):
    return 1/(2*np.pi*np.sqrt(L*C))

def parametros(freq, i, n_turns , space, width, dout):
    Din = calc_d_in(dout, n_turns, S, width)
    if Din < 30e-6:
        Din=30e-6
    Davg = calc_d_avg(Din, dout)
    Rox = calc_rox(Din, dout)
    Ls = calc_Ls(n_turns, Davg, tab1['K1'][i], tab1['K2'][i], Rox, miu)

    Delta = calc_delta(freq, miu)
    T_eff = calc_t_eff(Delta)
    l = calc_l(tab1['N Sides'][i], Davg, n_turns)
    Rs = calc_Rs(l, width, Delta, T_eff)

    Cp = calc_Cp(n_turns, width)
    Cox = calc_Cox(l, width)
    Csub = calc_Csub(l, width)
    Rsub = calc_Rsub(l, width)

    Z = calc_Z(freq, Ls, Rs, Cp, Cox, Csub, Rsub)
    L = calc_L(Z, freq)
    Qfactor= calc_QF(Z)
    Fres= calc_Fres(Ls, Cp)

    return [Din, Davg, Ls, Rs, Cp, Cox, Csub, Rsub, Z, L, Qfactor, Fres]

def plotter_error(a, b, xaxis, xlabel, title):
    E_rel=((a-b)/b)*100
    plt.figure()
    plt.ylabel('%')
    plt.xlabel(xlabel)
    plt.title(title)
    plt.xscale("log")
    plt.plot(xaxis, E_rel, '+')

#####
##
#####
##

#####
##
##### Partes 1, 2 e 3 #####

W = 10e-6 # • Largura da fita de metal, w
Dout = 340e-6 # • Diâmetro externo, Dout
S = 5e-6 # • Espaço entre fitas, s
```

```
n = 3 # • Número de voltas, n
f = 1e9
```

```
U=[]
U.append(parametros(f, 0, n, S, W, Dout))
U.append(parametros(f, 1, n, S, W, Dout))
U.append(parametros(f, 2, n, S, W, Dout))
params = pd.DataFrame(U)
params.columns = ['Din','Davg', 'Ls', 'Rs', 'Cp', 'Cox',
                  'Csub', 'Rsub', 'Z', 'L', 'Qfactor', 'Fres']
params['Shape'] = tab1['Shape']
```

```
for i in range(0,3,1):
```

```
    Rvec1=[]
    Lvec1=[]
    Qvec1=[]
    Fvec1=[]
```

```
    for f in np.logspace(3, 11, num=81):
```

```
        Delta = calc_delta(f, miu)
        T_eff = calc_t_eff(Delta)
        l = calc_l(tab1['N Sides'][i], params['Davg'][i], n)
        Rs = calc_Rs(l, W, Delta, T_eff)

        Z = calc_Z(f, params['Ls'][i], Rs, params['Cp'][i],
                  params['Cox'][i], params['Csub'][i], params['Rsub'][i])
        L = calc_L(Z, f)
        Qfactor= calc_QF(Z)
        Rvec1.append(Z.real)
        Fvec1.append(f)
        Lvec1.append(L)
        Qvec1.append(Qfactor)
```

```
    if(i==0):
        X=Lt_Square
    elif(i==1):
        X=Lt_Hexagonal
    else:
        X=Lt_Octogonal
```

```
plt.figure()
plt.ylabel("L(H)")
plt.xlabel("F(Hz)")
plt.title("L(f) %s" %tab1['Shape'][i])
plt.xscale("log")
plt.plot(Fvec1, Lvec1, label='Python')
plt.plot(Fvec1, X, label='Ltspace')
plt.legend(bbox_to_anchor=(1, 1))
```

```
plotter_error(Lvec1, X, Fvec1, "F(Hz)", "Erro relativo de L - Python  
vs Ltspice %s" %tab1['Shape'][i])
```

```
plt.figure()  
plt.ylabel("Qfactor")  
plt.xlabel("F(Hz)")  
plt.title("Qfactor(f) %s" %tab1['Shape'][i])  
plt.xscale("log")  
plt.xlim(1e8, 5e10)  
plt.ylim(-10,40)  
plt.plot(Fvec1, Qvec1)
```

```
#####  
##  
#####  
##
```

```
#####  
##  
##### Parte 4 #####
```

```
W = 10e-6 # • Largura da fita de metal, w  
Dout = 340e-6 # • Diâmetro externo, Dout  
S = 5e-6 # • Espaço entre fitas, s  
n = 3 # • Número de voltas, n
```

```
#Queremos obter novos W, Dout e N  
L1 = 6.5e-9  
f=2.4e9  
x0 = (400e-6,2,20e-6)
```

```
boundd = (200e-6,1200e-6) #D_out  
boundn = (1.5,8) #N  
boundw = (10e-6,60e-6) #W  
boundx = (boundd, boundn, boundw)
```

```
def get_L(x):  
    Douto, No, Wo = x  
    [Din, Davg, Ls, Rs, Cp, Cox, Csub, Rsub, Z, L, Qfactor, Fres] =  
    parametros(f, 0, No, S, Wo, Douto)  
    print("L=", L)  
    return abs((L-L1))/L1
```

```
dmin=20e-6  
def get_din_min(x):  
    Douto, No, Wo = x  
    return Douto - 2*No*Wo - 2*(No-1)*S - dmin
```

## #€ 2.4

fresmin = 2.4e10

def get\_fres(x):

    Douto, No, Wo = x

    [Din, Davg, Ls, Rs, Cp, Cox, Csub, Rsub, Z, L, Qfactor, Fres] =  
    parametros(f, 0, No, S, Wo, Douto)

    return Fres-fresmin

res=minimize(get\_L, x0, method='SLSQP', bounds=boundx,  
              constraints=({'type': 'ineq', 'fun':get\_din\_min},  
                          {'type': 'ineq', 'fun':get\_fres}))

print("Res=", res)

results=res.x

p2=[]

p2.append(parametros(f, 0, results[1], S, results[2], results[0]))

params2 = pd.DataFrame(p2)

params2.columns = ['Din', 'Davg', 'Ls', 'Rs', 'Cp', 'Cox',  
                    'Csub', 'Rsub', 'Z', 'L', 'Qfactor', 'Fres']

Rvec1=[]

Lvec1=[]

Qvec1=[]

Fvec1=[]

Rvec2=[]

Lvec2=[]

Qvec2=[]

Fvec2=[]

for f in np.logspace(3, 11, num=1000):

    Delta1 = calc\_delta(f, miu)

    T\_eff1 = calc\_t\_eff(Delta1)

    l1 = calc\_l(tab1['N Sides'][0], params['Davg'][0], n)

    Rs1 = calc\_Rs(l1, W, Delta1, T\_eff1)

    Z1 = calc\_Z(f, params['Ls'][0], Rs1, params['Cp'][0],  
                params['Cox'][0], params['Csub'][0], params['Rsub'][0])

    L1 = calc\_L(Z1, f)

    Qfactor1 = calc\_QF(Z1)

    Rvec1.append(Z1.real)

    Fvec1.append(f)

    Lvec1.append(L1)

    Qvec1.append(Qfactor1)

    Delta = calc\_delta(f, miu)

    T\_eff = calc\_t\_eff(Delta)

    l = calc\_l(tab1['N Sides'][0], params2['Davg'][0], results[1])

```
Rs = calc_Rs(l, results[2], Delta, T_eff)
```

```
Z = calc_Z(f, params2['Ls'][0], Rs, params2['Cp'][0],  
           params2['Cox'][0], params2['Csub'][0], params2['Rsub'][0])
```

```
L = calc_L(Z, f)
```

```
Qfactor= calc_QF(Z)
```

```
Rvec2.append(Z.real)
```

```
Fvec2.append(f)
```

```
Lvec2.append(L)
```

```
Qvec2.append(Qfactor)
```

```
plt.figure()
```

```
plt.ylabel("L(H)")
```

```
plt.xlabel("F(Hz)")
```

```
plt.title("L(f) %s" %tab1['Shape'][0])
```

```
plt.xscale("log")
```

```
plt.plot(Fvec1, Lvec1, label='Não otimizado')
```

```
plt.plot(Fvec2, Lvec2, label='Otimizado')
```

```
plt.legend(bbox_to_anchor=(1, 1))
```

```
plt.xlim(1e9, 1e11)
```

```
# plotter_error(Lvec2, Lvec1, Fvec1, "F(Hz)",
```

```
# "Erro relativo de L - Python vs Ltspice %s" %tab1['Shape'][i])
```

```
plt.figure()
```

```
plt.ylabel("Qfactor")
```

```
plt.xlabel("F(Hz)")
```

```
plt.title("Qfactor(f) %s" %tab1['Shape'][0])
```

```
plt.xscale("log")
```

```
plt.xlim(1e8, 5e10)
```

```
plt.ylim(-10, 40)
```

```
plt.plot(Fvec1, Qvec1, label='Não otimizado')
```

```
plt.plot(Fvec2, Qvec2, label='Otimizado')
```

```
plt.legend(bbox_to_anchor=(1, 1))
```



## Anexo B: Código do Modelo- $\pi$ Verilog

```
// VerilogA for inductors, modelopi, veriloga
`include "constants.vams"
`include "disciplines.vams"

module modelop(t1,t2);
  inout t1,t2;
  electrical t1,t2, p1, p2;
  ground gnd;

  // TECH FILE UMC130nm
  real miur = 1;
  real miu = 4*`M_PI *1e-7*miur;
  real E0 = 8.854187817e-12;
  real c = 299792458;

  // Layer Oxide (Metal)
  real t = 2.8e-6;
  real Rsheet = 10*1e-3;
  real Ro= Rsheet * t;
  real toxide = 5.42e-6;
  real t_M_underpass = 0.4e-6;
  real toxide_Munderpass = toxide - t_M_underpass - 4.76e-6;
  real Erox = 4;
  real Eox =E0 * Erox;
  real Ersub = 11.9;
  real Esub = E0 * Ersub;
  real Tsub = 700e-6;
  real Sub_resistiv = 2800;

  //Square Shape
  parameter real nsides = 4;
  parameter real k1 = 2.34;
  parameter real k2 = 2.75;
  parameter real f = 2.4e9;
  parameter real s = 5e-6;
  parameter real dout = 1.02336158e-3;
  parameter real nturns = 1.5;
  parameter real w = 1.005e-5;

  //Calcs
  real din = dout - 2*nturns*w - 2*(nturns-1)*s;
  real davg = 0.5 * (din + dout);
  real rox = (dout - din) / (dout + din);
  real Ls = (nturns**2) * davg * k1 * miu / (1 + k2 * rox);

  real delta = sqrt(rox/(`M_PI*f*miu));
  real teff = 1 - exp(-t/delta);
  real l = nsides * davg * nturns * tan(`M_PI/nsides);
  real Rs = l * Ro / (w * delta * teff);
```

```
real cp = (nturns-1) * (w**2) * Eox / toxide_Munderpass;
real cox = 0.5 * l * w * Eox / toxide;
real csub = 0.5 * l * w * Esub / Tsub;
real rsub = 2 * Tsub * Sub_resistiv / (l * w);
```

```
//Connections
```

```
branch (t1, t2) bCp, bLR;
```

```
analog begin
```

```
    V(bLR) <+ I(bLR)*Rs;
    V(bLR) <+ Ls*ddt(I(bLR));
    I(bCp) <+ cp*ddt(V(bCp));
```

```
    I(t1, p1) <+ cox*ddt(V(t1, p1));
    I(p1, gnd) <+ V(p1, gnd)/rsub;
    I(p1, gnd) <+ csub*ddt(V(p1, gnd));
```

```
    I(t2, p2) <+ cox*ddt(V(t2, p2));
    I(p2, gnd) <+ V(p2, gnd)/rsub;
    I(p2, gnd) <+ csub*ddt(V(p2, gnd));
```

```
end
endmodule
```