

**UNIVERSIDADE DO ESTADO DE MATO GROSSO
FACULDADE DE CIÊNCIAS EXATAS E TECNOLÓGICAS
CURSO DE BACHARELADO EM ENGENHARIA ELÉTRICA**

ARTHUR LAND OLIVEIRA

**RESPOSTA TEMPORAL DE CIRCUITO RLC COM DIODO:
IMPLEMENTAÇÃO E ESTUDO EM PYTHON**

UNEMAT – Campus de Sinop

2025/2

**UNIVERSIDADE DO ESTADO DE MATO GROSSO
FACULDADE DE CIÊNCIAS EXATAS E TECNOLÓGICAS
CURSO DE BACHARELADO EM ENGENHARIA ELÉTRICA**

ARTHUR LAND OLIVEIRA

**RESPOSTA TEMPORAL DE CIRCUITO RLC COM DIODO:
IMPLEMENTAÇÃO E ESTUDO EM PYTHON**

Atividade avaliativa apresentado na disciplina de Eletrônica de Potência do Curso de Bacharelado em Engenharia Elétrica – UNEMAT, Campus Universitário de Sinop – MT, como pré-requisito para obtenção de nota na segunda avaliação (P2).

Prof. Me.: Marcus Felipy Glatz Rodrigues

UNEMAT – Campus de Sinop

2025/2

LISTA DE FIGURAS

Figura 1 – Circuito associado a um estimulador magnético.....	5
Figura 2 – Resultado temporal de corrente e tensão.....	13

LISTA DE ABREVIATURAS

R – Resistência elétrica

L – Indutância

C – Capacitância

t – tempo

$i(t)$ – Corrente no indutor

$v_C(t)$ – Tensão no capacitor

V_0 – Tensão inicial do capacitor

V_x – Tensão do capacitor pré-carregado

EDO – Equação Diferencial Ordinária

RLC – Circuito formado por Resistor, Indutor e Capacitor

Ω – Ohm

H – Henry

F – Farad

A – Ampère

V – Volt

μs – Microsssegundo

Python – Linguagem utilizada nas simulações

NumPy – Biblioteca de computação numérica em Python

SciPy – Biblioteca científica de Python

Matplotlib – Biblioteca para gráficos em Python

solve_ivp – Método numérico de integração de EDOs da SciPy

SUMÁRIO

1 INTRODUÇÃO	4
2 DESCRIÇÃO DO PROBLEMA	5
3 MODELAGEM MATEMÁTICA DO CIRCUITO	6
4 IMPLEMENTAÇÃO EM PYTHON	7
4.1 DEFINIÇÃO DOS PARÂMETROS DO CIRCUITO	7
4.2 FUNÇÃO DO SISTEMA DE EQUAÇÕES DIFERENCIAIS	7
4.3 CONDIÇÕES INICIAIS E MALHA DE TEMPO	7
4.4 RESOLUÇÃO NUMÉRICA COM SOLVE_IVP	8
4.5 GERAÇÃO DOS GRÁFICOS	8
4.6 CÁLCULO DE RESULTADOS NUMÉRICOS	8
5 OBJETIVOS.....	9
5.1 OBJETIVOS ESPECÍFICOS	9
6 FUNDAMENTAÇÃO TEÓRICA.....	10
6.1 CIRCUITOS RLC	10
6.2 DIODOS E COMUTAÇÃO	10
6.3 SISTEMAS DINÂMICOS E EQUAÇÕES DIFERENCIAIS	10
6.4 ESTIMULAÇÃO MAGNÉTICA	11
7 METODOLOGIA	12
7.1 DEFINIÇÃO DOS PARÂMETROS FÍSICOS.....	12
7.2 FORMULAÇÃO DAS EQUAÇÕES DIFERENCIAIS	12
7.3 IMPLEMENTAÇÃO NO AMBIENTE PYTHON.....	12
7.4 ANÁLISE DOS RESULTADOS	12
8 RESULTADOS E DISCUSSÃO	13
9 CONCLUSÃO	15
REFÊRENCIAS.....	16

1 INTRODUÇÃO

Este relatório apresenta o desenvolvimento, em linguagem Python, de um modelo numérico para simular a resposta temporal de um circuito RLC com diodo, utilizado como estimulador magnético em aplicações biomédicas.

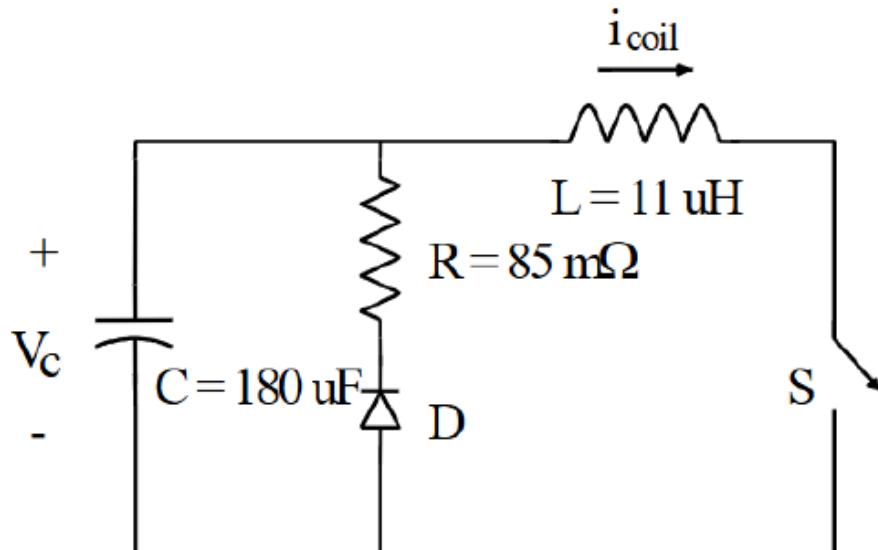
O circuito em questão representa um equipamento em que um capacitor é pré-carregado com uma tensão elevada e, ao fechar uma chave em $t = 0$, a energia armazenada é transferida para uma bobina, gerando um pulso de corrente que produz o campo magnético necessário ao tratamento.

O objetivo principal é obter, por meio de simulação numérica, a corrente na bobina e a tensão no capacitor ao longo do tempo, considerando o comportamento de comutação do diodo presente no circuito. A implementação foi realizada em Python utilizando as bibliotecas numpy, matplotlib e scipy.integrate.

2 DESCRIÇÃO DO PROBLEMA

O problema proposto envolve um circuito RLC associado a um estimulador magnético. O capacitor é pré-carregado com uma tensão V_x (entre 0 e 1000 V) e, em $t = 0$, o interruptor é fechado, liberando a energia para a bobina. O diodo é ideal e passa a conduzir a partir de determinado instante, modificando a topologia do circuito.

Figura 1: Circuito associado a um estimulador magnético.



Fonte: Exercício proposto pelo professor.

No código desenvolvido, foi simulada a situação em que a tensão inicial do capacitor é $V_0 = 900 \text{ V}$. Os demais parâmetros utilizados foram:

- Indutância: $L=11 \mu\text{H}$
- Resistência: $R=0,085 \Omega$
- Capacitância: $C=180 \mu\text{F}$

A partir desses dados, o problema se resume a resolver numericamente as equações diferenciais que descrevem a corrente no indutor e a tensão no capacitor, levando em conta o regime antes e depois da condução do diodo.

3 MODELAGEM MATEMÁTICA DO CIRCUITO

Para modelar o circuito, foram escolhidas duas variáveis de estado:

- $i(t)$: corrente no indutor
- $vC(t)$: tensão no capacitor

Essas variáveis são representadas no vetor de estados $x = [i, vC]$.

O circuito possui dois modos de operação, definidos pela tensão no capacitor:

Modo 1 – Diodo desligado ($vC > 0$)

Quando a tensão no capacitor é positiva, considera-se que o diodo está desligado. O capacitor e o indutor formam um circuito em série. As equações diferenciais utilizadas foram:

$$\begin{aligned}\frac{di}{dt} &= \frac{vC}{L} \\ \frac{dvC}{dt} &= -\frac{i}{C}\end{aligned}$$

A primeira equação decorre da relação $vL = Ldi/dt$. A segunda representa o capacitor descarregando por meio da corrente i .

Modo 2 – Diodo conduzindo ($vC \leq 0$)

Quando a tensão no capacitor se torna nula ou negativa, o diodo passa a conduzir e o circuito passa a ser equivalente a L , C e R em paralelo. Nesse caso, as equações adotadas foram:

$$\begin{aligned}\frac{di}{dt} &= \frac{vC - Ri}{L} \\ \frac{dvC}{dt} &= -\frac{i + \frac{vC}{R}}{C}\end{aligned}$$

Aqui, considera-se a queda de tensão no resistor R e a corrente total que atravessa o capacitor, somando a corrente do indutor e a corrente pelo resistor.

A lógica de comutação foi implementada diretamente na função que define o sistema de EDOs, utilizando uma estrutura condicional if/else baseada no sinal de vC .

4 IMPLEMENTAÇÃO EM PYTHON

O código foi estruturado em etapas numeradas e comentadas, facilitando a compreensão do fluxo lógico.

4.1 DEFINIÇÃO DOS PARÂMETROS DO CIRCUITO

Na primeira parte do script, foram importadas as bibliotecas necessárias (numpy, matplotlib.pyplot e solve_ivp) e definidos os parâmetros elétricos do circuito:

- $V_0 = 900$ – tensão inicial do capacitor
- $L = 11\text{e-}6$ – indutância em henry
- $R = 85\text{e-}3$ – resistência em ohms
- $C = 180\text{e-}6$ – capacitância em farad

Esses valores reproduzem o cenário físico descrito no enunciado, com um capacitor pré-carregado e uma bobina de baixa indutância para gerar um pulso rápido de corrente.

4.2 FUNÇÃO DO SISTEMA DE EQUAÇÕES DIFERENCIAIS

Na etapa seguinte, foi criada a função circuito(t, x), que recebe o tempo t e o vetor de estados x e devolve as derivadas [di_dt, dv_dt]:

- Dentro da função, as variáveis de estado são descompactadas ($i, vC = x$).
- Em seguida, aplica-se a lógica de comutação do diodo:
 - Se $vC > 0$, utiliza-se o modelo do circuito com capacitor e indutor em série.
 - Caso contrário, utiliza-se o modelo com L, C e R em paralelo.

Essa função traduz diretamente as equações diferenciais da seção anterior para código Python.

4.3 CONDIÇÕES INICIAIS E MALHA DE TEMPO

As condições iniciais adotadas foram:

- Corrente inicial no indutor: $i_0 = 0,0 \text{ A}$
- Tensão inicial no capacitor: $v_0 = V_0$

O vetor de estados inicial foi definido como $x_0 = [i_0, v_0]$.

Para o intervalo de simulação, foi escolhido:

- $t_0 = 0.0 \text{ s}$ (início)

- $tf = 400\text{e-}6$ s (fim da simulação, 400 μs)
- $t_eval = \text{np.linspace}(t0, tf, 4000)$ para gerar uma malha de 4000 pontos igualmente espaçados, garantindo boa resolução temporal para os gráficos.

4.4 RESOLUÇÃO NUMÉRICA COM SOLVE_IVP

A solução do sistema de EDOs foi realizada com a função `solve_ivp`, da biblioteca `scipy.integrate`:

```
sol = solve_ivp(circuito, [t0, tf], x0, t_eval = t_eval)
```

A partir do objeto `sol`, foram extraídos:

- `tempo = sol.t` – vetor de instantes de tempo
- `corrente = sol.y[0]` – solução para $i(t)$
- `tensao = sol.y[1]` – solução para $vC(t)$

4.5 GERAÇÃO DOS GRÁFICOS

Para visualizar os resultados, foi criada uma figura com dois subgráficos (`subplot`):

1. **Primeiro gráfico:** corrente no indutor em função do tempo
 - Eixo x: `tempo * 1e6` (tempo em microsegundos)
 - Eixo y: corrente (em ampères)
 - Foi definido o título geral “Resposta Temporal do Circuito RLC com Diodo”.
2. **Segundo gráfico:** tensão no capacitor em função do tempo
 - Eixo x: `tempo * 1e6`
 - Eixo y: `tensao` (em volts)

Ambos os gráficos utilizam grade (`plt.grid(True)`) para facilitar a leitura. O comando `plt.tight_layout()` ajusta automaticamente os espaços entre os subgráficos.

4.6 CÁLCULO DE RESULTADOS NUMÉRICOS

Por fim, o código calcula e exibe alguns valores numéricos de interesse:

- Corrente máxima no indutor: `np.max (corrente)`
- Tensão mínima no capacitor: `np.min (tensao)`

Esses resultados são impressos no console com formatação de duas casas decimais.

5 OBJETIVOS

Simular numericamente, por meio de Python, o comportamento transitório de um circuito RLC com diodo, analisando a evolução temporal da corrente no indutor e da tensão no capacitor.

5.1 OBJETIVOS ESPECÍFICOS

- Elaborar a modelagem matemática do circuito, incluindo seus modos de operação.
- Implementar as equações diferenciais no ambiente Python utilizando `solve_ivp`.
- Gerar gráficos representativos da resposta do sistema.
- Calcular valores característicos, tais como corrente máxima no indutor e tensão mínima no capacitor.
- Interpretar fisicamente os resultados obtidos, relacionando-os ao funcionamento de um estimulador magnético.

6 FUNDAMENTAÇÃO TEÓRICA

6.1 CIRCUITOS RLC

Um circuito RLC é composto por um resistor R, um indutor L e um capacitor C. A dinâmica desse tipo de circuito é descrita por equações diferenciais que representam a troca de energia entre elementos armazenadores:

O indutor armazena energia no campo magnético, determinado por

$$vL = L \frac{di}{dt}$$

O capacitor armazena energia no campo elétrico, determinado por

$$iC = C \frac{dvC}{dt}$$

A combinação desses componentes pode resultar em respostas oscilatórias, amortecidas ou superamortecidas, dependendo da resistência total do circuito.

6.2 DIODOS E COMUTAÇÃO

O diodo é um componente semicondutor que permite a passagem de corrente em apenas uma direção. No modelo ideal, considera-se que ele:

- Não conduz enquanto a tensão no sentido direto for menor ou igual a zero.
- Conduz instantaneamente quando a tensão direta torna-se positiva.

Essa característica cria **topologias distintas** ao longo do tempo:

- **Modo 1 (diodo desligado):** O capacitor descarrega-se apenas pela trajetória indutor-capacitor.
- **Modo 2 (diodo ligado):** O diodo cria uma malha alternativa, envolvendo o resistor, modificando o regime transitório.

Esse tipo de comutação é comum em circuitos de potência e conversores, assim como em sistemas de estimulação magnética.

6.3 SISTEMAS DINÂMICOS E EQUAÇÕES DIFERENCIAIS

A simulação numérica de circuitos elétricos exige a formulação do sistema como um conjunto de equações diferenciais ordinárias (EDOs). Ao definir um vetor de estados $x = [i, vC]$, pode-se representar o circuito como:

$$\frac{dx}{dt} = f(x, t)$$

O método empregado (`solve_ivp`) utiliza algoritmos robustos, como Runge–Kutta de quarta ou quinta ordem, adequados para problemas de comutação.

6.4 ESTIMULAÇÃO MAGNÉTICA

Em aplicações biomédicas, a descarga de um capacitor sobre uma bobina produz pulsos intensos de corrente, responsáveis por gerar campos magnéticos necessários à estimulação neural ou muscular. O modelo computacional desenvolvido representa a etapa de descarga de energia, permitindo prever:

- intensidade da corrente;
- duração do pulso;
- comportamento do capacitor após o disparo.

7 METODOLOGIA

A metodologia aplicada neste trabalho foi estruturada em quatro etapas principais:

7.1 DEFINIÇÃO DOS PARÂMETROS FÍSICOS

Com base no enunciado do exercício e no funcionamento do circuito, definiram-se valores realistas para resistência, indutância, capacidade e tensão inicial.

7.2 FORMULAÇÃO DAS EQUAÇÕES DIFERENCIAIS

Foram desenvolvidas duas versões das equações diferenciais, correspondentes aos modos de operação do diodo. Essa lógica foi incorporada à função Python responsável pela avaliação do sistema dinâmico.

7.3 IMPLEMENTAÇÃO NO AMBIENTE PYTHON

O código foi organizado em blocos:

- importação de bibliotecas;
- definição de parâmetros;
- função de equações diferenciais;
- configuração do intervalo de tempo;
- chamada de solve_ivp;
- geração de gráficos.

7.4 ANÁLISE DOS RESULTADOS

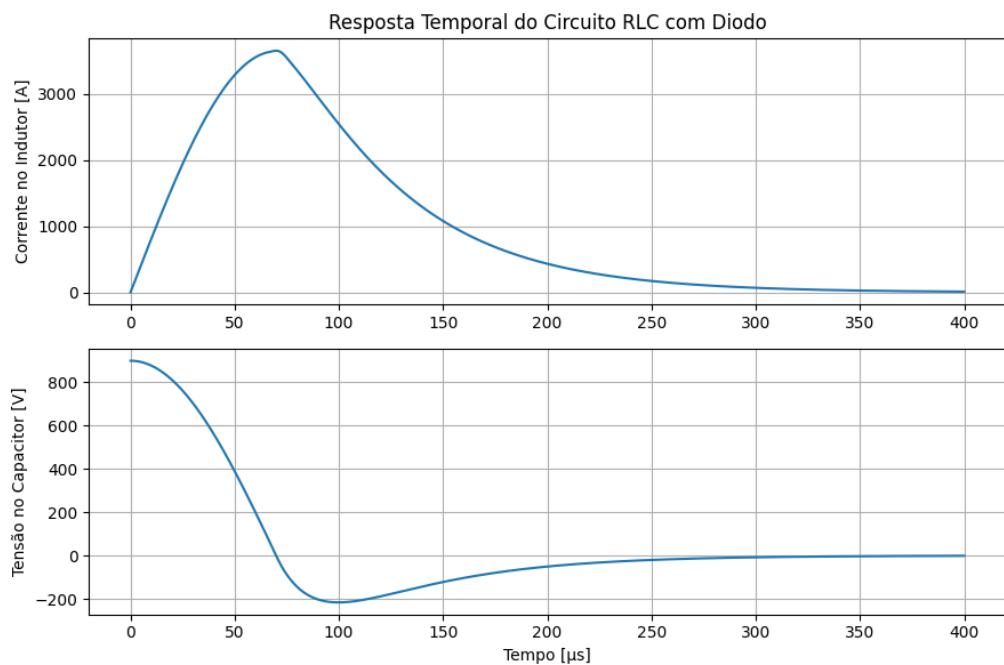
Após a simulação, realizou-se:

- verificação visual da evolução temporal das grandezas;
- identificação de máximos e mínimos de grandezas elétricas;
- discussão física sobre o comportamento geral do circuito.

8 RESULTADOS E DISCUSSÃO

A simulação numérica mostra que, logo após o fechamento da chave em $t = 0$, a energia armazenada no capacitor é transferida rapidamente para a bobina, gerando um pulso de corrente de alta intensidade. Para o caso $v_0 = 900V$, o valor de pico da corrente no indutor é da ordem de alguns quiloampéres, enquanto a tensão no capacitor passa de um valor inicial positivo para valores próximos ou abaixo de zero, momento em que o diodo passa a conduzir e altera o regime do circuito.

Figura 2: Resultado temporal de corrente e tensão.



Fonte: Acervo do autor.

===== *Resultados da Simulação* =====

Corrente máxima no indutor: 3648.93 A

Tensão mínima no capacitor: - 216.42 V

=====

Os gráficos permitem observar:

- Um aumento rápido da corrente na bobina seguido de oscilações amortecidas, associadas à dinâmica RLC.

- A queda da tensão no capacitor a partir de 900 V, cruzando o zero e assumindo valores negativos, até que a energia seja gradualmente dissipada no resistor.

O cálculo da corrente de pico e da tensão mínima fornece informações importantes para a avaliação do projeto do estimulador magnético, como a especificação adequada dos componentes (corrente suportada pela bobina, tensão suportada pelo capacitor e dissipação de potência no resistor).

9 CONCLUSÃO

Neste trabalho foi apresentado o desenvolvimento de um código em Python para simular a resposta temporal de um circuito RLC com diodo, usado como modelo simplificado de um estimulador magnético biomédico. A partir da modelagem em termos de equações diferenciais e da utilização do método numérico solve_ivp, foi possível obter a corrente na bobina e a tensão no capacitor ao longo do tempo, bem como valores característicos como corrente de pico e tensão mínima.

O uso de Python mostrou-se adequado por permitir:

- Implementação clara das equações do circuito;
- Flexibilidade na escolha do intervalo de simulação e dos parâmetros do sistema;
- Visualização gráfica imediata dos resultados;
- Extração simples de grandezas de interesse (máximos, mínimos etc.).

Esse tipo de abordagem numérica é especialmente útil em contextos de ensino e projeto, pois possibilita a análise de diferentes cenários (por exemplo, variar V₀, R, L ou C) sem necessidade de montagem física do circuito, contribuindo para a compreensão da dinâmica de sistemas RLC com comutação por diodos.

REFÊRENCIAS

- ALEXANDER, Charles K.; SADIKU, Matthew N. O.** *Fundamentos de Circuitos Elétricos*. 5. ed. Porto Alegre: AMGH, 2013.
- HAYT, William H.; KEMMERLY, Jack E.; DURBIN, Steven M.** *Análise de Circuitos em Engenharia*. 8. ed. Porto Alegre: AMGH, 2012.
- OGATA, Katsuhiko.** *Engenharia de Controle Moderno*. 5. ed. São Paulo: Pearson Prentice Hall, 2010.
- SCIPY.** *SciPy v1.10.1 Reference Guide*. Disponível em: <https://docs.scipy.org/>. Acesso em: 19 nov. 2025.
- MATPLOTLIB.** *Matplotlib Documentation*. Disponível em: <https://matplotlib.org/>. Acesso em: 19 nov. 2025.
- PYTHON SOFTWARE FOUNDATION.** *Python 3.11 Documentation*. Disponível em: <https://docs.python.org/3/>. Acesso em: 19 nov. 2025.