

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

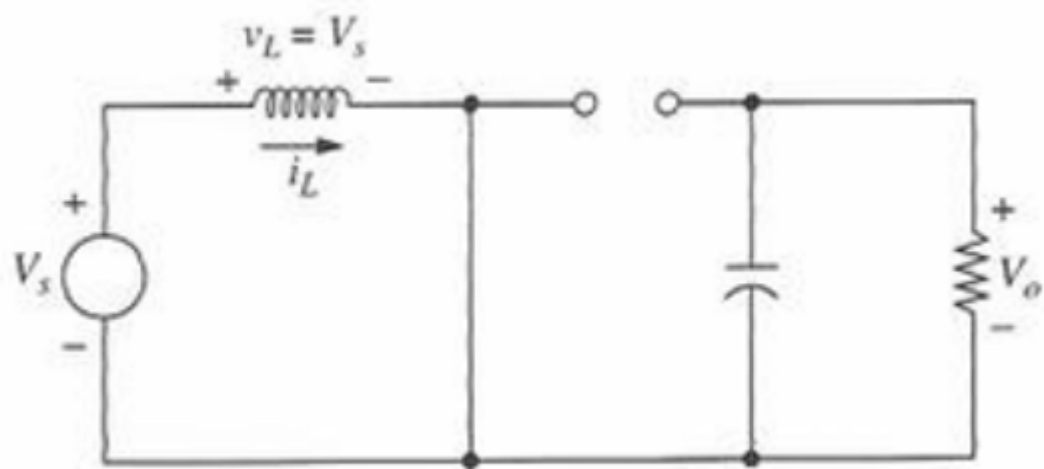
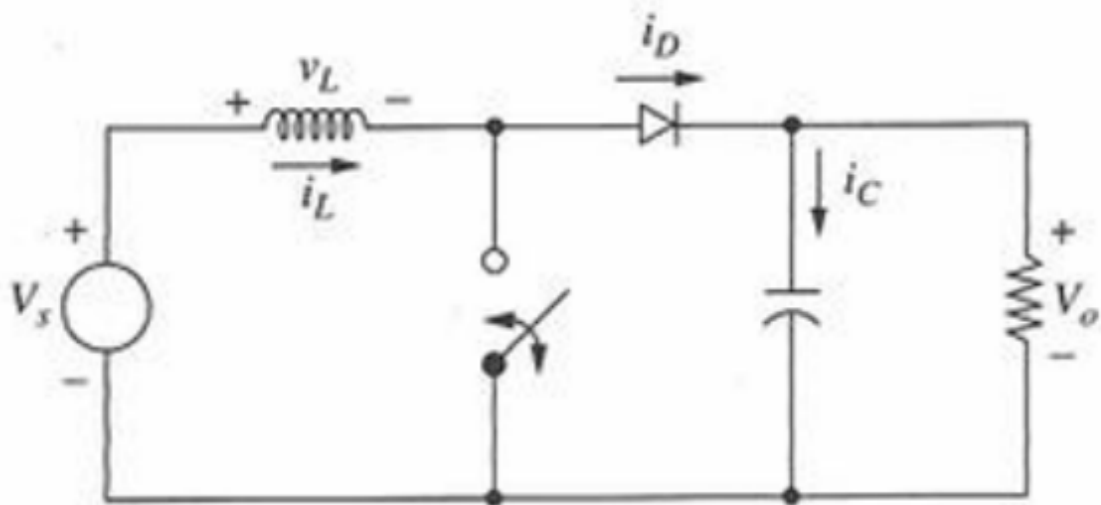
AMANDA JACOBS TAVARES

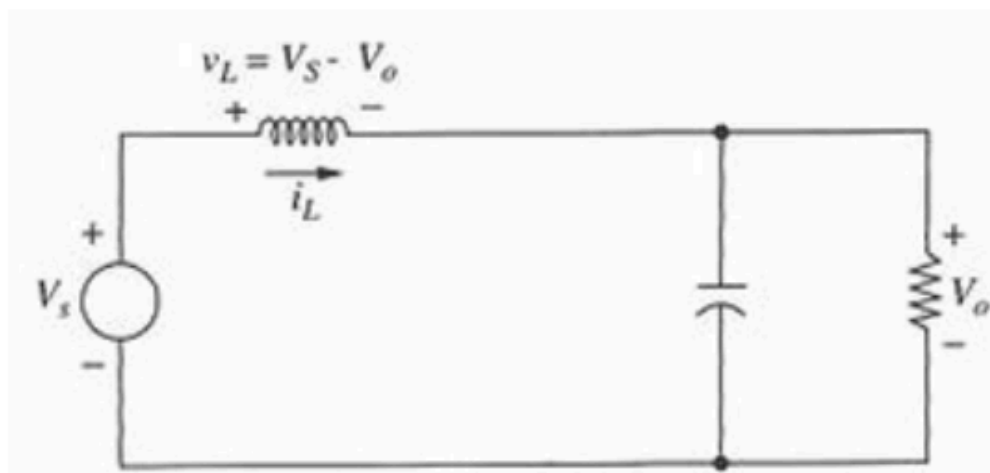
ELETRÔNICA DE POTÊNCIA - CONVERSOR BOOST

UNEMAT – Campus de Sinop

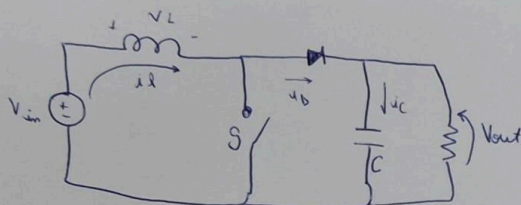
2025/2

1 INTRODUÇÃO



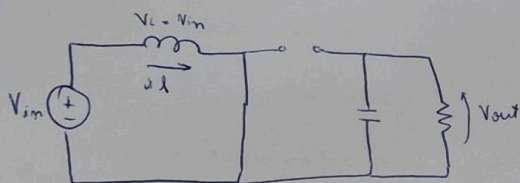


Converter Boost



Primeiro estágio:

Chave fechada

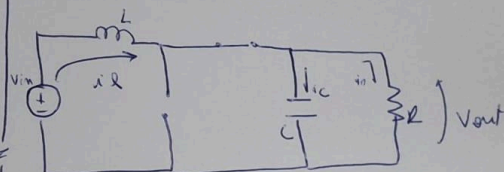


• Capacitor descarrega fornecendo energia para a carga

• Indutor carrega com a tensão de entrada

Segundo estágio

Chave aberta



• Indutor fornece energia para a saída, carregando o capacitor e fornecendo energia para a carga

• A injeção de energia no capacitor, fornecida pelo indutor, que aumenta a tensão de saída

2 METODOLOGIA DA SIMULAÇÃO

A simulação foi implementada em Python utilizando três módulos principais:

- **numpy** – para manipulação numérica e criação dos vetores de tempo,
- **matplotlib.pyplot** – para construção dos gráficos das grandezas,

```
#IMPORTANDO AS BIBLIOTECAS
import numpy as np
import matplotlib.pyplot as plt
```

PARÂMETROS UTILIZADOS

```
#Parâmetros

Vin = 180          #tensão de entrada (V)
Vout = 380         #tensão de saída (V)
Pout = 1300        #potência máxima na saída (W)
Fs = 40e3          #frequência de chaveamento (Hz)
Ts = 1 / Fs        #período de chaveamento (s)

VI1 = 0.2          # Ondulação da corrente no indutor (% da corrente média)
VW1 = 0.01         # Ondulação da tensão no capacitor (% da tensão de saída)
```

Onde definimos os parâmetros iniciais.

Fs = 40e3 define a frequência de chaveamento em 40 kHz. Produz um período

$T_s = 1 / F_s = 250 \text{ us}$.

② Valores nominais:

$$D = \frac{V_o - V_{in}}{V_o} = 0,5263$$

$$I_{out} = \frac{P_{out}}{V_{out}} = 3,42 \text{ A}$$

$$I_{in} = \frac{P_{out}}{V_{in}} = 7,222 \text{ A}$$

$$R_{out} = \frac{V_{out}^2}{P_{out}} = 111,08 \Omega$$

③ Projeto indutor

$$L = \frac{V_{in} \cdot D}{f_s \cdot \Delta I_L \% \cdot I_n} = 1,6396 \text{ mH}$$

④ Projeto capacitor

$$C = \frac{I_o \cdot D}{f_s \cdot \Delta V_o \% \cdot V_o} = 11,84 \text{ nF}$$

Calculamos L e C

```
#Tempo de simulação
t_end = 20e-3
dt = Ts / 200
t = np.arange(0, t_end, dt)
```

t_end = 20e-3: tempo total da simulação: 0,02 s = 20 ms.

dt = Ts / 200: passo de tempo da simulação. Você subdivide cada período de chaveamento em 200 pontos → integração bem fina dentro de um ciclo.

t = np.arange(0, t_end, dt):

Cria um vetor de tempo que vai de 0 até (quase) t_end, com incrementos dt. Cada elemento de t é um instante em que o sistema será calculado.

```
# Vetores de estado
iL = np.zeros_like(t) # Corrente no indutor
vO = np.zeros_like(t) # Tensão de saída

# Condições iniciais
iL[0] = 0.0
vO[0] = 0.0
```

iL = np.zeros_like(t):

Cria um array de mesmo tamanho que t, inicializado com zeros. Vai armazenar o valor da corrente no indutor em cada instante de tempo.

vO = np.zeros_like(t):

O mesmo para a tensão de saída (no capacitor/carga).

iL[0] = 0.0: corrente inicial no indutor é zero (conversor desligado no início).

vO[0] = 0.0: tensão inicial na saída também é zero (capacitor descarregado). Esses são os estados iniciais que o método numérico vai usar.

Funcionamento do Loop Principal

```
#LOOP DE SIMULAÇÃO
for k in range(len(t) - 1):
    t_cycle = t[k] % Ts
```

for k in range(len(t) - 1):

Loop de $k = 0$ até $k = \text{len}(t) - 2$.

Em cada iteração, você calcula o estado no instante $k+1$ a partir do estado no instante k .

t_cycle = t[k] % Ts:

Calcula o tempo “dentro” do período de chaveamento.

O operador % (módulo) pega o resto da divisão de $t[k]$ por T_s . Resultado: um valor entre 0 e T_s .

Isso diz em que ponto do PWM você está (em qual parte do ciclo ON/OFF).

```
# Chave ON ou OFF
if t_cycle < D * Ts:
    vL = Vin          # ON
else:
    vL = Vin - vO[k]  # OFF
```

Chave ON ou OFF: indica o que será decidido abaixo.

if t_cycle < D * Ts::

Se o tempo dentro do ciclo é menor que $D * T_s$, significa que ainda estamos no intervalo ON do PWM (MOSFET conduzindo).

Dentro do if (ON):

$v_L = V_{in}$: tensão no indutor é igual à tensão de entrada (condição ideal de boost em ON).

No else (OFF):

$v_L = V_{in} - v_O[k]$:

Quando a chave desliga, o indutor “se conecta” à saída via diodo, então a tensão sobre o indutor é $V_{in} - V_{out}$.

$v_O[k]$ é a tensão de saída no instante atual.

```

# Atualiza iL
iL[k+1] = iL[k] + (vL / L) * dt

# Atualiza vO
i_load = vO[k] / R
vO[k+1] = vO[k] + (iL[k] - i_load) / C * dt

```

indutor:

$$iL[k+1] = iL[k] + (V_L / L) \cdot dt$$

$$V_L = L \cdot \frac{di}{dt}$$

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

$$iL(t+dt) = iL(t) + \frac{V_L}{L} \cdot dt$$

$$iL[k] = \text{valor atual}$$

$$(V_L / L) \cdot dt = \text{incremento}$$

$$iL[k+1] = \text{próximo valor}$$

capacitor:

$$i_{\text{load}} = vO[k] / R$$

$$i_{\text{cap}} = iL - i_{\text{load}}$$

$$i_{\text{cap}} = C \cdot \frac{dv_O}{dt}$$

$$\frac{dv_O}{dt} = \frac{iL - i_{\text{load}}}{C}$$

$$v_O(t+dt) = v_O(t) + \frac{iL - i_{\text{load}}}{C} \cdot dt$$

3 RESULTADOS

```
#GRÁFICO
plt.figure(figsize=(9,4))
plt.plot(t * 1e3, vO, label='vO(t)')
plt.axhline(Vout_ref, linestyle='--', label=f'Referência = {Vout_ref:.1f} V')
plt.xlabel('Tempo (ms)')
plt.ylabel('Tensão (V)')
plt.title('Tensão de saída - Conversor Boost (180 V → 380 V)')
plt.grid(True)
plt.legend()
plt.tight_layout()
plt.show()
```

plt.figure(figsize=(9,4)):

Cria uma nova figura (janela de gráfico) de 9 por 4 polegadas.

plt.plot(t * 1e3, vO, label='vO(t')):

Plota a tensão de saída vO em função do tempo.

t * 1e3 converte o eixo x de segundos para milissegundos.

label='vO(t)' será usado na legenda.

plt.axhline(Vout_ref, linestyle='--', label=f'Referência = {Vout_ref:.1f} V'):

Desenha uma linha horizontal na altura da tensão de referência (380 V).

linestyle='--': linha tracejada.

O f'...' formata o texto com o valor numérico de Vout_ref com 1 casa decimal.

plt.xlabel('Tempo (ms)): rótulo do eixo x.

plt.ylabel('Tensão (V)): rótulo do eixo y.

plt.title('Tensão de saída - Conversor Boost (180 V → 380 V)):

Define o título do gráfico (tensões de entrada e saída).

plt.grid(True): Ativa a grade auxiliar.

plt.legend(): Exibe a legenda com os labels definidos no plot e no axhline.

plt.tight_layout(): Ajusta margens automaticamente para nada ficar cortado.

plt.show(): Exibe o gráfico na tela.

4 ANÁLISE

O gráfico apresenta a evolução temporal da tensão de saída do conversor Boost após o acionamento. Observa-se que, no instante inicial, o capacitor encontra-se descarregado, fazendo com que a aplicação imediata do duty cycle nominal provoque uma elevação abrupta da tensão, resultando em um pico significativo de sobretensão. Em seguida, surgem oscilações amortecidas decorrentes da troca de energia entre o indutor e o capacitor, característica típica de um sistema de segunda ordem com baixo amortecimento. Com o passar do tempo, parte dessa energia é dissipada pela carga, reduzindo gradualmente a amplitude das oscilações até que o sistema converge para o regime permanente. Ao final do intervalo analisado, a tensão estabiliza-se próxima ao valor desejado (380 V), confirmando que o duty cycle escolhido é adequado para a operação nominal, embora o comportamento transitório evidencie a ausência de controle de partida suave ou realimentação regulatória.

