

Este código realiza a simulação numérica de um conversor Boost (elevador de tensão) que converte 48V para 200V com potência de 800W, usando o método de Euler Explícito para resolver as equações diferenciais do circuito.

1º Passo importa as bibliotecas

- **numpy:** Para cálculos numéricos e criação de vetores
- **matplotlib:** Para visualização gráfica dos resultados

```
1 ## Importando bibliotecas
2 ✓ import numpy as np
3 import matplotlib.pyplot as plt
4
```

2º definições do parâmetros

```
## Definição dos parâmetros alterados -
Vin = 48.0           # Tensão de entrada (V) - bateria 48V
Vout = 200.0          # Tensão de saída (V) - para sistema de 200V
Pout = 800.0          # Potência máxima na saída (W)
Fs = 100e3            # Frequência de chaveamento (Hz) - 100 kHz
Ts = 1 / Fs           # Período de chaveamento (s)
VIL = 0.25             # Ondulação da corrente no indutor (%) 
VVL = 0.015            # Ondulação da tensão no capacitor (%)
```

3º Cálculo dos Parâmetros do Circuito

```
## Cálculo dos valores nominais e dimensionamento de componentes
D = 1 - (Vin / Vout)      # Razão Cíclica para conversor Boost
Iout = Pout / Vout        # Corrente de saída
Iin = Pout / Vin          # Corrente de entrada
R = (Vout**2) / Pout       # Resistência da carga
L = (Vin * D) / (Fs * VIL * Iin)  # Indutância de entrada
C = (Iout * D) / (Fs * VVL * Vout) # Capacitância de saída
```

4º Configuração da Simulação Temporal

```
# Tempo de simulação: 15 ms para ver o amortecimento
t_end = 15e-3            # 15 ms
dt = Ts / 200              # subdividindo cada período em 200 passos
t = np.arange(0, t_end, dt)
```

5º Inicialização das Variáveis

```

# Vetores de simulação
iL = np.zeros_like(t)      # Corrente do indutor
v0 = np.zeros_like(t)      # Tensão do capacitor/saída

# Condições iniciais
iL[0] = 0.0
v0[0] = 0.0

```

6º : Loop Principal de Simulação

```

# Loop de simulação usando Euler Explícito
for k in range(len(t) - 1):
    t_cycle = t[k] % Ts

    # Determinar estado da chave (ON ou OFF)
    if t_cycle < D * Ts:
        # Estado ON: Chave fechada
        vL = Vin           # Tensão no indutor = Vin
        iC = -v0[k] / R   # Corrente no capacitor = - corrente na carga
    else:
        # Estado OFF: Chave aberta
        vL = Vin - v0[k]  # Tensão no indutor = Vin - Vout
        iC = iL[k] - v0[k] / R # Corrente no capacitor = iL - iR

    # Método de Euler Explícito para o indutor
    iL[k + 1] = iL[k] + (vL / L) * dt

    # Método de Euler Explícito para o capacitor
    v0[k + 1] = v0[k] + (iC / C) * dt

```

7º Visualização dos Resultados

```

# Plot apenas da tensão de saída
plt.figure(figsize=(12, 6))
plt.plot(t * 1e3, v0, 'b-', linewidth=1.5, label='Tensão de Saída (vout)')
plt.axhline(Vout, color='red', linestyle='--', label=f'Tensão Desejada = {Vout}V')
plt.title(f'Resposta do Conversor Boost ({Vin}V → {Vout}V, {Pout}W)')
plt.xlabel('Tempo (ms)')
plt.ylabel('Tensão (V)')
plt.grid(True, alpha=0.3)
plt.legend()
plt.tight_layout()
plt.show()

```