Este relatório apresenta a análise da resposta dinâmica de um conversor Buck operando com tensão de entrada de 50 V e tensão de saída teórica de 20 V. A simulação foi realizada utilizando o método numérico de Euler Explícito, permitindo observar o comportamento transitório durante os primeiros 5 milissegundos de operação. Os resultados demonstram a estabilização progressiva da tensão de saída, convergindo para o valor esperado de 20 V com características de amortecimento típicas de um sistema de segunda ordem.

1ºImportação de Bibliotecas

- numpy (np): Biblioteca para cálculos numéricos e manipulação de arrays
- matplotlib.pyplot (plt): Biblioteca para criação de gráficos

```
import numpy as np
import matplotlib.pyplot as plt
```

2ºDefinição dos Parâmetros do Conversor

- Vin = 50.0: A fonte de alimentação fornece 50 volts
- L = 1.20e-3: O indutor tem 1.2 milihenries (0.0012 H)
- R = 4: A carga consome corrente através de um resistor de 4 ohms
- C = 15.60e-6: O capacitor de filtro tem 15.6 microfarads
- Fs = 20e3: A chave liga/desliga 20.000 vezes por segundo
- D = 0.4: A chave fica fechada 40% do tempo em cada ciclo
- Ts = 1/20000 = 0.00005 s: Cada ciclo dura 50 microssegundos

```
# Parâmetros conforme valores calculados fornecidos pelo usuário
Vin = 50.0           # Tensão de entrada (V)
L = 1.20e-3          # Indutância do indutor (H) = 1.20 mH
R = 4                # Resistência de carga (Ω) = 4 Ω
C = 15.60e-6         # Capacitância do capacitor de saída (F) = 15.60 µF
Fs = 20e3            # Frequência de chaveamento (Hz) = 20 kHz
D = 0.4              # Duty cycle (40%)
Ts = 1 / Fs          # Período de chaveamento (s)
```

3ºCálculo da Tensão de Saída Esperada

```
# Valor médio de saída (esperado)
Voutmed = D * Vin     # 20 V
```

4º Configuração do Tempo de Simulação

- t_end = 0.005 s: Simulação dura 5 milissegundos (100 ciclos completos)

- dt = Ts/200 = 0.25 μs: Cada ciclo tem 200 pontos de simulação
- np.arange(): Cria um vetor de tempo de 0 a 5 ms com passo de 0.25 μs
- Resultado: 20.000 pontos no total (5ms ÷ 0.25μs)

```
# Tempo de simulação: 5 ms para ver o amortecimento
t_end = 5e-3       # 5 ms
dt = Ts / 200      # subdividindo cada período em 200 passos
t = np.arange(0, t_end, dt)
```

5ºInicialização dos Vetores de Estado

- np.zeros_like(t): Cria arrays do mesmo tamanho que t, preenchidos com zeros
- iL: Armazenará a corrente no indutor em cada instante
- vout: Armazenará a tensão de saída em cada instante
- Condições iniciais: No tempo zero, corrente e tensão são zero

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

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

6º Loop Principal de Simulação (Método de Euler)

```
# Loop de integração (Euler explícito)
for k in range(len(t) - 1):
    t_cycle = t[k] % Ts
```

- for k in range(len(t) - 1): Percorre todos os pontos de tempo
- t_cycle = t[k] % Ts: Calcula a posição dentro do ciclo atual
    - Exemplo: Se Ts = 50μs e t[k] = 63μs, então t_cycle = 13μs

7º estado da chave

```python
# Determinar aberto ou fechado
if t_cycle < D * Ts:
    # Estado ON: Chave fechada
    vL = Vin - vout[k]    # Tensão no indutor = Vin - Vout
    iC = -vout[k] / R     # Corrente no capacitor = - corrente na car
else:
    # Estado OFF: Chave aberta
    vL = -vout[k]         # Tensão no indutor = -Vout
    iC = iL[k] - vout[k] / R  # Corrente no capacitor = iL - iR

# MÉTODO EULER EXPLÍCITO - Atualizar corrente do indutor
diL_dt = vL / L  # Derivada: diL/dt = vL/L
iL[k + 1] = iL[k] + diL_dt * dt

# MÉTODO EULER EXPLÍCITO - Atualizar tensão do capacitor
dvC_dt = iC / C  # Derivada: dvC/dt = iC/C
vout[k + 1] = vout[k] + dvC_dt * dt
```

Estado ON (Chave Fechada):

- vL = Vin - vout: A tensão no indutor é a diferença entre entrada e saída
- iC = -vout/R: O capacitor fornece toda a corrente para a carga

Estado OFF (Chave Aberta):

- vL = -vout: O indutor agora tem polaridade invertida
- iC = iL - vout/R: O capacitor recebe corrente do indutor menos a da carga

Equações Diferenciais do Sistema:

1. Indutor: diL/dt = vL/L
2. Capacitor: dvC/dt = iC/C

Método de Euler Explícito:

- Aproxima a derivada por diferença finita
- x[k+1] = x[k] + (dx/dt) × Δt
- Simples mas requer passo pequeno para estabilidade

8º **Visualização dos Resultados**

```python
# Plot apenas da tensão de saída
plt.figure(figsize=(12, 6))
plt.plot(t * 1e3, vout, 'b-', linewidth=1.5, label='Tensão de Saída (vout)')
plt.axhline(Voutmed, color='red', linestyle='--', label=f'Média Teórica = {Voutmed:.1f} V')
plt.title('Resposta Transitória e Estabilização da Tensão de Saída (50 V → 20 V) - Euler Explícito')
plt.xlabel('Tempo (ms)')
plt.ylabel('Tensão (V)')
plt.grid(True, alpha=0.3)
plt.legend()
plt.tight_layout()
plt.show()
```