Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática

Simulação de Sistemas Elétricos por Quadripolos

Hércules Allan Barbosa Trindade
Natã Fragoso Trigueiro
Nyelton Gomes Faustino
Rodrigo Moreira Rios
Samuel Rufino da Silva Coutinho

Campina Grande, Paraíba
2025

Hércules Allan Barbosa Trindade
Natã Fragoso Trigueiro
Nyelton Gomes Faustino
Rodrigo Moreira Rios
Samuel Rufino da Silva Coutinho

Simulação de Sistemas Elétricos por Quadripolos

Relatório do Projeto de Quadripolos, destinado ao Prof. Dr. Luiz Augusto Medeiros referente à pontuação complementar da disciplina de Circuitos Elétricos II

Professor:
Prof. Dr. Luiz Augusto Medeiros

Campina Grande, Paraíba
2025

Sumário

1 Introdução

Esse trabalho consiste na documentação do projeto, destinado ao Prof. Dr. Luiz Augusto Medeiros, sobre a modelagem e simulação de sistemas elétricos por meio de quadripolos. A metodologia de abordagem utilizada foi a utilização da construção do sistema em Python e, em paralelo, foi modelado e simulado no LTspice. Essa abordagem concomitante por duas perspectivas diferente acerca do mesmo sistema permitiu a mútua validação e verificação.

Com isso, foi possível verificar os valores obtidos na simulação do LTspice com os valores da simulação do modelo em Python. Por conseguinte, essa metodologia garante mais confiabilidade dos resultados finais.

Além disso, utilizou-se da teoria de quadripolos para modelar cada componente em relação aos parâmetros de referência fornecidos pela proposta encaminhada pelo professor da disciplina. Após modelagem do sistema, calculou-se os valores de tensões, correntes e potências do sistema. Por fim, realizou-se o controle de perdas de potências, o ajuste do TAP dos transformadores e a inserção de banco de capacitores.

1.1 Objetivo Geral

O objetivo geral do projeto é a simulação de um sistema elétrico de alta tensão por meio da utilização da teoria de quadripolos.

1.2 Objetivos Específicos

Com o intuito da realização completa do objetivo geral do trabalho, é necessário estabeler objetivos específicos que possam guiar o percusso e a metrificação do progresso.

2 Fundamentação Teórica

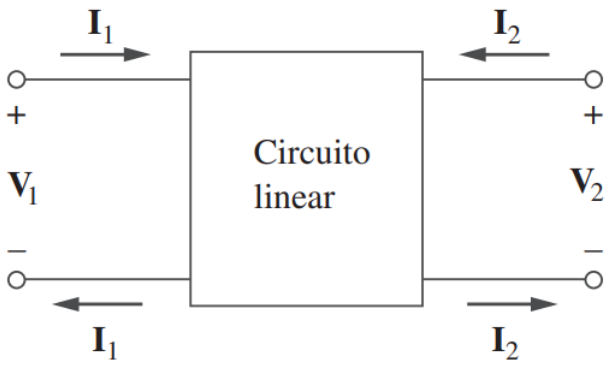
2.1 Quadripolos

Para analisar um sistema elétrico, é necessário entender a teoria empregada para análise do circuito. Esse arcabouço teórico baseia-sa em quadripolos ou circuitos de duas portas. Uma porta é definida por um par de terminais, o qual pode entrar ou sair uma corrente. Os quadripolos são circuitos de duas portas distintas para entrada e saída.

Esse tipo de representação é extremamente importante em circuitos de comunicação, sistemas de controle, sistemas de potência e eletrônica. Conhecer os parâmetros de um circuito permite a representação do circuito como uma caixa preta e facilita o seu estudo. Então, a análise na teoria de quadripolos busca entender o comportamento nos terminais do circuito.

Os quadripolos são representados por uma tensão de entrada V_1, uma tensão de saída V_2, uma corrente de entrada I_1 e uma corrente de saída I_2.

Figura 1: Circuito de duas portas.



Fonte: Sadiku, 2013.

No projeto, foi utilizado uma pequena parcela da teoria abragente de circuitos de duas portas, visto que foi utilizado os conceitos de parâmetros de transmissão, interconexão de quadripolos e modelos de alguns componentes.

2.2 Parâmetros de Transmissão

Existem vários parâmetros para modelagem dos quadripolos, tais como parâmetros de impedância, admitância, híbridos, entre outros. Contudo, a modelagem dos sistema elétrico baseia-se nos parâmetros de transmissão em que estabelece uma relação entre as variáveis de entrada e de saída de acordo com a Equação 1.

$$\begin{bmatrix} V_1 \\ I_1 \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_2 \\ -I_2 \end{bmatrix} \tag{1}$$

Nessa abordagem dos parâmetros, percebe-se que a corrente de saída possui um sinal contrário, uma vez que é considerado a corrente saindo do quadripolo para facilitar a compreensão da associação em cascata de quadripolos. Esse modelo é o mais utilizado na análise de linhas de transmissão e os parâmetros podem ser facilmente determinados.

$$A = \left.\frac{V_1}{V_2}\right|_{I_2=0} \quad B = -\left.\frac{V_1}{I_2}\right|_{V_2=0} \quad C = \left.\frac{1_1}{V_2}\right|_{I_2=0} \quad D = -\left.\frac{I_1}{I_2}\right|_{V_2=0} \tag{2}$$

O parâmetro A é a razão de tensão de circuito aberto, o B é a impedância de transferência de curto-circuito negativa, o C é a admitância de transferência de cirucito aberto, o D é a razão de corrente de curto-circuito negativa.

2.3 Interconexão de Quadripolos

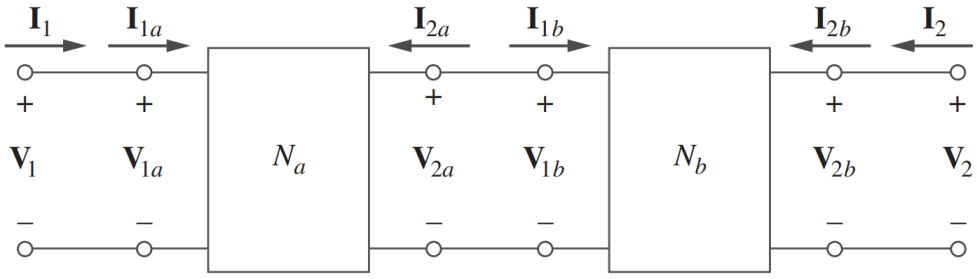
Um circuito elétrico pode ser grande e extenso tal como um sistema elétrico e, por isso, é necessário ser dividido em várias partes. Esses circuitos são divididos em vários quadripolos e interligados posteriormente. No trabalho, utilizou-se as associações em paralelo e em cascata, mas existe também a associação em série.

2.3.1 Cascata

Nesse tipo de associação de quadripolos, a saída de um quadripolo N_a é ligado na entrada de outro quadripolo N_b. Logo, os parâmetros de transmissão para o circuito global é o produto de cada parâmetro individual.

$$\begin{bmatrix} V_1 \\ I_1 \end{bmatrix} = \begin{bmatrix} A_a & B_a \\ C_a & D_a \end{bmatrix} \begin{bmatrix} A_b & B_b \\ C_b & D_b \end{bmatrix} \begin{bmatrix} V_2 \\ -I_2 \end{bmatrix} \tag{3}$$

Figura 2: Conexão em cascata de dois circuitos de duas portas.



Fonte: Sadiku, 2013.

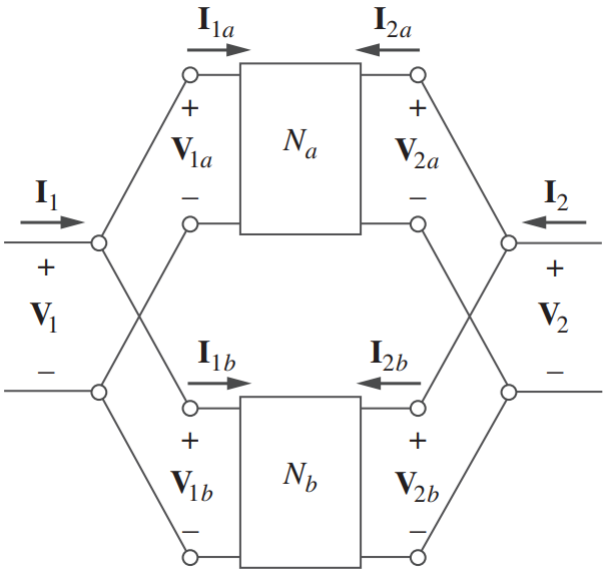
2.3.2 Paralelo

Na associação em paralelo, as tensões em suas portas são iguais e as correntes nas portas do circuito maior são a soma das correntes em cada porta. Além disso, o referencial

comum também estará interligado. Logo, matematicamente, os parâmetros individuais de y se somam para fornecer os parâmetros de y do circuito maior.

$$\begin{bmatrix} V_1 \\ I_1 \end{bmatrix} = \begin{bmatrix} y_{11a} + y_{11b} + & y_{12a} + y_{12b} \\ y_{21a} + y_{21b} & y_{22a} + y_{22b} \end{bmatrix} \begin{bmatrix} V_2 \\ -I_2 \end{bmatrix} \tag{4}$$

Figura 3: Conexão em paralelo de dois circuitos de duas portas.



Fonte: Sadiku, 2013.

2.4 Modelos de Componentes

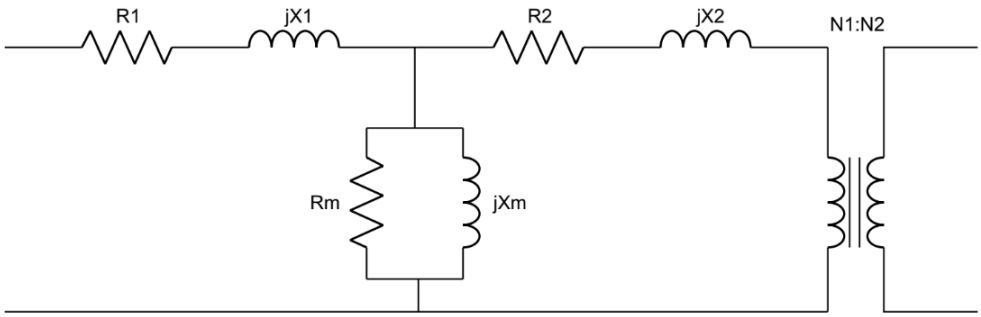
Com o conhecimento já alcançado da teoria de quadripolo, é possível representar componentes por meio dessa teoria, tais como transitores e equipamentos elétricos. Em razão disso, no densenvolvimento do trabalho, foi utilizado a modelagem de componentes de modo que facilitasse a compreensão do sistema elétrico em sua completude.

Logo, o primeiro dispositivo a ser modelado foi a linha de transmissão com o modelo π em que os parâmetros utilizados na modelagem dependem da resistência interna, da indutância e das capacitâncias parasitas. This model is descrita in Figrua 5 and in Equo 5.

$$\begin{bmatrix} V_1 \\ I_1 \end{bmatrix} = \begin{bmatrix} 1 + \frac{jwC}{2}(R + jwL) & R + jwL \\ jwC + \frac{(jwC)^2}{4}(R + jwL) & 1 + \frac{jwC}{2}(R + jwL) \end{bmatrix} \begin{bmatrix} V_2 \\ -I_2 \end{bmatrix} \tag{5}$$

Além disso, o transformador também foi modelado de acordo com os parâmetros de resistências internas, relação do número de espiras, indutância, entre outros. Essa modelagem pode ser visualizada pela Figura 4 e pelas equações subsequentes.

Figura 4: Modelo do transformador.



Fonte: Glover, 2012.

$$\begin{bmatrix} V_1 \\ I_1 \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \frac{N_1}{N_2} & 0 \\ 0 & \frac{N_2}{N_1} \end{bmatrix} \begin{bmatrix} V_2 \\ -I_2 \end{bmatrix} \tag{6}$$

$$A = 1 + \frac{R_m + jwL_m}{R_m + jwL_m}(R_1 + jwL_1) \tag{7}$$

$$B = R_1 + jwL_1 + R_2 + jwL_2 + \frac{R_m + jwL_m}{R_m - jwL_m}(R_1 + jwl_1) \tag{8}$$

$$C = \frac{R_m + jwL_m}{R_m - jwL_m} \tag{9}$$

$$D = 1 + \frac{R_m + jwL_m}{R_m - jwL_m}(R_2 + jwL_2) \tag{10}$$

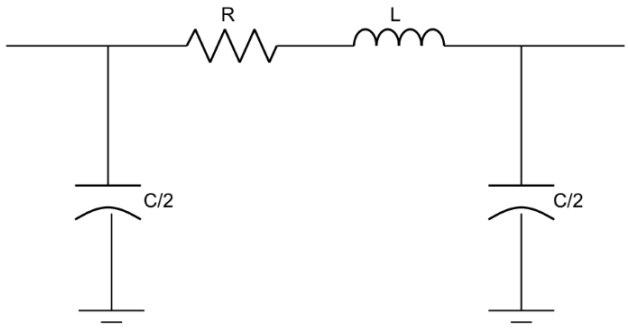
Por fim, foram utilizados os modelos de quadripolos para as cargas e para impedância série quando houve a necessidade. A modelagem da carga (Equação 11) e da impedãncia série (Equação 12) são feitas em função do valor da impedância Z, mas existe alteração na matriz da função de transferência.

$$\begin{bmatrix} V_1 \\ I_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{1}{z} & 1 \end{bmatrix} \begin{bmatrix} V_2 \\ -I_2 \end{bmatrix} \tag{11}$$

$$\begin{bmatrix} V_1 \\ I_1 \end{bmatrix} = \begin{bmatrix} 1 & Z \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V_2 \\ -I_2 \end{bmatrix} \tag{12}$$

Por meio da modelagem desses componentes, é possível realizar a associação deles

Figura 5: Modelo π da linha de transmissão.



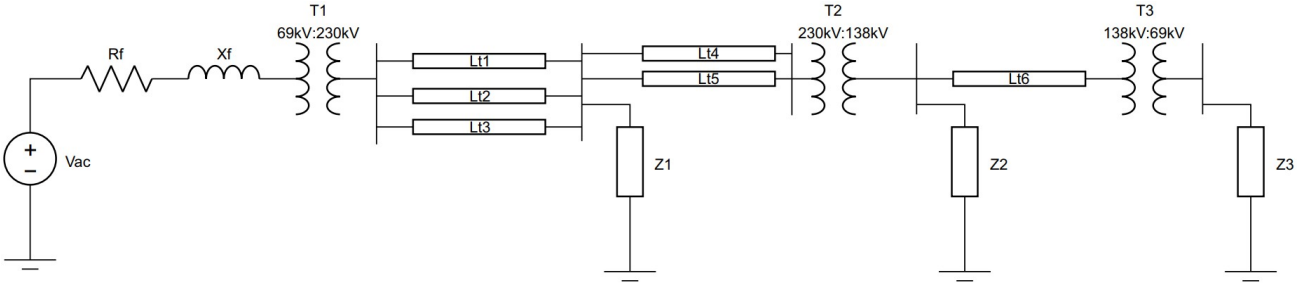
Fonte: Glover, 2012.

por meio de conexões em cascata ou em paralelo para conseguirmos simular e analisar os resultados apresentados.

3 Desenvolvimento

Possuindo os conhecimentos necessários sobre os quadripolos, as formas de interligação e a modelagem dos componentes elétricos, é possível realizar a modelagem do sistema elétrico da figura 6. O LTspice foi utilizado para simular o sistema elétrico e, em paralelo, foi realizado um script em python para validação mútua da simulação no Python e no LTSpice. Essa metodologia permitiu uma maior confiabilidade nos resultados.

Figura 6: Sistema Elétrico para a modelagem.



Fonte: Luiz Augusto, 2025.

3.1 Modelagem em Python

A modelagem em python foi realizada no CoCalc, um software que permite a programação com multiplos colaboradores ao mesmo tempo. Esse programa possui compatibilidade com o Jupyter Notebooks, o qual foi utilizado para segmentação e organização do código. Logo, o arquivo do código está dividido em células, o qual o usuário pode escolher se compila separadamente ou todas ao mesmo tempo.

3.1.1 Criação das Funções de Transferência

Primeiramente, iniciou-se importando as bibliotecas Numpy (Criação e manipulação de matrizes), Sympy (Solucionador de sistemas lineares), Matplotlib (Plotagem de gráficos), Cmath (Utilização e manipulação de números complexos), Scipy.optimize (Utilizado para determinar máximos e mínimos de funções).

Após isso, iniciou a modelagem dos componentes e associações utilizado o numpy para realizar as matrizes de transferência. Então, essas matrizes foram estabelecidas em funções que recebem os parâmetros do componente. Os seguintes trechos de código incluem o funcionamento geral de cada sessão, mas não incluem o trechos com as funções do print.

Criação das Funções de Transferência:

```
def carga(z): # Matriz de carga
  return np.array([[1, 0], [1/z, 1]])
def impedanciaSerie(z): # Matriz de impedância série
  return np.array([[1, z], [0, 1]])
def linhaTransmissao(r_km, l_km, c_km, km, w): # Matriz de linha de transmissão média (80 km - 200 km)
  r = r_km*km
  l = l_km*km
  c = c_km*km
  A = 1 + ((j*w*c)/2)*(r + j*w*l)
  B = r + j*w*l
  C = j*w*c + (((j*w*c)**2)/4)*(r + j*w*l)
  D = 1 + ((j*w*c)/2)*(r + j*w*l)
  return np.array([[A, B], [C, D]])
def transformador(r1, r2, rm, xl1, xl2, xlm, n): # Matriz de transformador de potência
  a = 1 + ((rm + j*xlm)/(rm*j*xlm))*(r1 + j*xl1)
  b = r1 + j*xl1 + r2 + j*xl2 + ((rm + j*xlm)/(rm*j*xlm))*(r1 + j*xl1)*(r2 + j*xl2)
  c = (rm + j*xlm)/(rm*j*xlm)
  d = 1 + ((rm + j*xlm)/(rm*j*xlm))*(r2 + j*xl2)
  m1 = np.array([[a, b], [c, d]])
  m2 = np.array([[1/n, 0], [0,n]])
  return np.dot(m1, m2)
def paralelo(m1,m2): # Interconexão de quadripolos em  paralela
    Aa = m1[0][0]
    Ba = m1[0][1]
    Ca = m1[1][0]
    Da = m1[1][1]
    Ab = m2[0][0]
    Bb = m2[0][1]
    Cb = m2[1][0]
    Db = m2[1][1]
    A = (Aa*Bb + Ab*Ba)/(Ba + Bb)
    B = (Ba*Bb)/(Ba + Bb)
    C = Ca + Cb + ((Aa - Ab)*(Db - Da))/(Ba + Bb)
    D = (Bb*Da + Ba*Db)/(Ba + Bb)
    return np.array([[A, B] , [C, D]])
def cascata(m1, m2): # Interconexão de quadripolos em cascata
  return np.dot(m1, m2)
```

3.1.2 Inicialização dos Parâmetros do Sistema

Após a definição das funções do circuito, é nessário definir os parâmetros do sistema que foram fornecidos pela proposta do Prof. Luiz. Logo, foram inicializados os valores de frequência de operação, impedância série, parâmetros da linha de transmissão, comprimentos da linha de transmissão, entre outros valores. Além disso, foi calculado a impedância das cargas pela tensão nominal de operação dos transformadores e pela potência complexa fornecida.

Inicialização dos Parâmetros do Sistema:

```
f = 60
w = 2*np.pi*f
Zf = 2 +(0.38*j)
R_lt = 0.182
L_lt = 2.28e-3
C_lt = 0.0140e-6
L123 = 100
L45 = 80
L6 = 120
R1_t = 7.6e-3
X1_t = 3.8e-3
R2_t = 33.9e-3
X2_t = 0.85e-3
Rm1_t = 4320
Xm1_t = 5050
Rm2_t = 432e3
Xm2_t = 505e3
Rm3_t = 402e3
Xm3_t = 607e3
n1 = 10/3
n2 = 0.6
n3  = 0.5
S1 = (11.9680e6) + (24.7076e6)*j
S2 = (30.8442e6) + (63.6053e6)*j
S3 = (17.0740e6) + (35.2865e6)*j
V_nominal = np.array([230e3,138e3,69e3])
z1 = (V_nominal[0]**2)/(S1.conjugate())
z2 = (V_nominal[1]**2)/(S2.conjugate())
z3 = (V_nominal[2]**2)/(S3.conjugate())
```

3.1.3 Modelagem do Sistema

A partir disso, foram criadas as matrizes de transmissão de cada elemento e, após isso, foram realizadas as associações entre os componentes com as associações em paralelo para as linhas de transmissão e o restante em cascata, incluindo as cargas.

Modelagem do Sistema:

```
# Matriz da Impedância Série de Thévenin
Zth = impedanciaSerie(Zf)
# Matrizes das Linhas de Transmissão
LT1 = LT2 = LT3 = linhaTransmissao(R_lt, L_lt, C_lt, L123, w)
LT4 = LT5 = linhaTransmissao(R_lt, L_lt, C_lt, L45, w)
LT6 = linhaTransmissao(R_lt, L_lt, C_lt, L6, w)
# Matrizes dos Transformadores
T1 = transformador(R1_t, R2_t, Rm1_t, X1_t, X2_t, Xm1_t, n1)
T2 = transformador(R1_t, R2_t, Rm2_t, X1_t, X2_t, Xm2_t, n2)
T3 = transformador(R1_t, R2_t, Rm3_t, X1_t, X2_t, Xm3_t, n3)
# Matrizes das Cargas
Z1 = carga(z1)
Z2 = carga(z2)
Z3 = carga(z3)
# Associação entre a fonte até a linha LT3
m1 = cascata(Zth, T1)
m2 = paralelo(LT1, LT2)
m3 = paralelo(m2, LT3)
m4 = cascata(m1, m3) # cascata até o transformador 1
# Associação entre Z1 e T2
m5 = cascata(m4, Z1)
m6 = paralelo(LT4, LT5)
m7 = cascata(m5, m6)
m8 = cascata(m7, T2) # cascata até o transformador 2
# Associação entre T2 e T3
m9 = cascata(m8, Z2)
m10 = cascata(m9, LT6)
m11 = cascata(m10,T3) # cascata até o transformador 3
# Associção entre T3 ao Gnd
T_equivalente = m11
```

3.1.4 Tensão Fasorial de Z_3

A partir disso, é possível iniciar os cálculos necessários, uma vez que já possui-se o sistema inteiramente modelado. A primeira etapa da análise do circuito, é calcular a tensão de saída da carga 3 e calcular a corrente em Z3. Para isso, utilizou-se a tensão de 69000V para o gerador e resolveu o sistema linear para a matriz de transformação equivalente.

Cálculo Tensão Fasorial de Z_3:

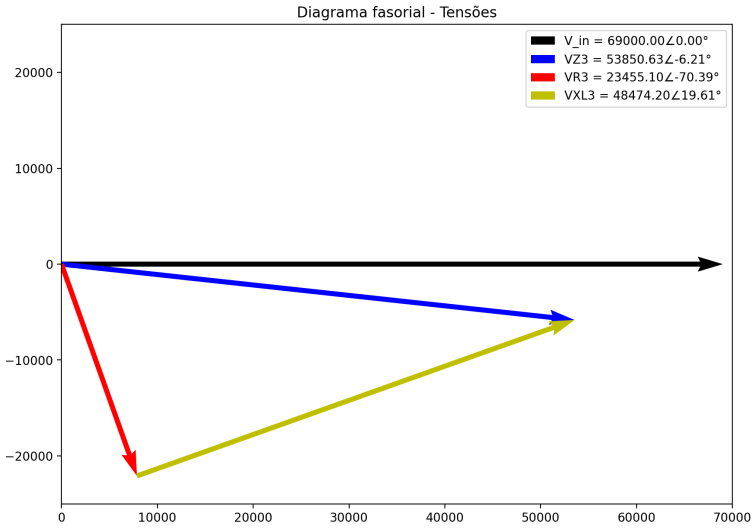
```
# Tensão  do Gerador
V_in = 69e3
# Variáveis do Sistema
I_inSym = sp.symbols("I_inSym")
V_Z3Sym = sp.symbols("V_Z3Sym")
I_Z3Sym = (V_Z3Sym/z3)
# Sistema de equações
T_eq = sp.Matrix(T_equivalente)
Out =  sp.Matrix([V_Z3Sym, I_Z3Sym])
In = sp.Matrix([V_in, I_inSym])
sistema = sp.Eq(T_eq*Out, In)
sol = sp.solve(sistema, (I_inSym,V_Z3Sym))
I_in = complex(sol[I_inSym])
V_Z3 = complex(sol[V_Z3Sym])
I_Z3 = complex(V_Z3/z3)
VR_Z3 = ((z3.real)*(I_Z3))
VXl_Z3 = ((z3.imag*j)*(I_Z3))
I_inFasor = fasor(I_in)
V_Z3Fasor = fasor(V_Z3)
I_Z3Fasor = fasor(I_Z3)
VR_Z3Fasor = fasor(VR_Z3)
VXl_Z3Fasor = fasor(VXl_Z3)
```

Resultados:

```
--------------------------------------------------------------------------------
Corrente de Entrada do Sistema
--------------------------------------------------------------------------------
I_in = 803.57 A 20.40°
--------------------------------------------------------------------------------
Tensão e Corrente na carga Z3
--------------------------------------------------------------------------------
V_Z3 = 53.85 kV -6.21°
I_Z3 = 443.39 A -70.39°
--------------------------------------------------------------------------------
```

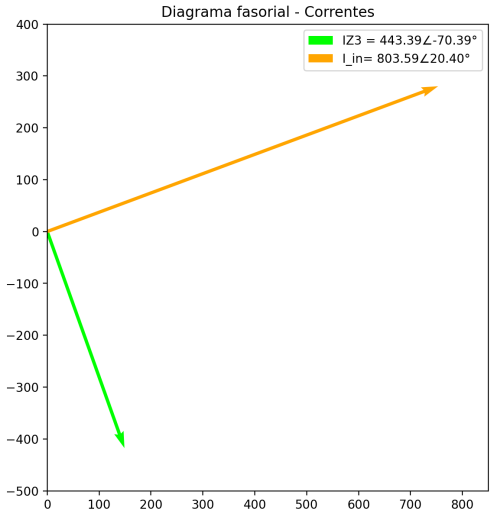
A partir da tensão fasorial calculada em V_3, é possível realizar a plotagem do diagrama fasorial na forma poligonal das tensões e das correntes. Para essa plotagem, deve ser levado em consideração a tensão de entrada, a queda de tensão no sistema elétrico e a tensão da carga. Para o diagrama das correntes, foi levado em consideração os valores da corrente de entrada e saída. Esses gráficos podem ser visualizados na Figura 7 e Figura 8.

Figura 7: Diagrama Fasorial das Tensões.



Fonte: Autoria própria, 2025.

Figura 8: Diagrama Fasorial das Correntes.



Fonte: Autoria Própria, 2025.

3.1.5 Cálculo da Tensão Fasorial de Z_1 e Z_2

Para realizar o cálculo da tensão fasorial nas demais cargas, utiliza-se o valor da corrente de entrada calculada na sessão anterior. Com a corrente e a tensão de entrada, é possível realizar o cálculo da tensão e corrente em ambas as cargas.

Cálculo da Tensão Fasorial de Z_1 e Z_2:

```
V_Z1Sym = sp.symbols("V_Z1Sym")
I_1Sym = sp.symbols("I_1Sym")
# Sistema de equações
T_eq = sp.Matrix(m4)
Out =  sp.Matrix([V_Z1Sym,I_1Sym])
In = sp.Matrix([V_in,I_in])
sistema = sp.Eq(T_eq*Out, In)
sol = sp.solve(sistema, [V_Z1Sym,I_1Sym])
V_Z1 = complex(sol[V_Z1Sym])
I_Z1 = complex(V_Z1/z1)
mag_I_in1, fase_I_in1 = cmath.polar(I_in)
fase_I_in1 = np.rad2deg(fase_I_in1)
mag_V_Z1, fase_V_Z1 = cmath.polar(V_Z1)
fase_V_Z1 = np.rad2deg(fase_V_Z1)
mag_I_Z1, fase_I_Z1 = cmath.polar(I_Z1)
fase_I_Z1 = np.rad2deg(fase_I_Z1)
# Corrente e tensão em Z2
# Variáveis do Sistema
V_Z2Sym = sp.symbols("V_Z2Sym")
I_2Sym = sp.symbols("I_2Sym")
# Sistema de equações
T_eq = sp.Matrix(m8)
Out =  sp.Matrix([V_Z2Sym,I_2Sym])
In = sp.Matrix([V_in,I_in])
sistema = sp.Eq(T_eq*Out, In)
sol = sp.solve(sistema, [V_Z2Sym, I_2Sym])
V_Z2 = complex(sol[V_Z2Sym])
I_Z2 = V_Z2/z2
mag_I_in, fase_I_in = cmath.polar(I_in)
fase_I_in = np.rad2deg(fase_I_in)
mag_V_Z2, fase_V_Z2 = cmath.polar(V_Z2)
fase_V_Z2 = np.rad2deg(fase_V_Z2)
I_Z2 = complex(V_Z2/z2)
mag_I_Z2, fase_I_Z2 = cmath.polar(I_Z2)
fase_I_Z2 = np.rad2deg(fase_I_Z2)
```

Resultados:

```
--------------------------------------------------------------------------------
Tensão e Corrente da carga Z1
--------------------------------------------------------------------------------
V_Z1 = 221.49 kV -2.23°
I_Z1 = 114.95 A -66.38°
--------------------------------------------------------------------------------
Tensão e Corrente da carga Z2
--------------------------------------------------------------------------------
V_Z2 = 127.06 kV -3.34°
I_Z1 = 471.65 A -67.47°
--------------------------------------------------------------------------------
```

3.1.6 Potência Fornecida pelo Gerador

Outro ponto extremamente importante, é calcular a potência complexa fornecida pela fonte geradora. Para isso, é nessário os valores da tensão de entrada em V_{RMS} e a corrente de entrada. Com isso, o usuário pode calcular esse valor em Python.

Potência Fornecida pelo Gerador:

```
#Cálculo da Potência Complexa
S_g = -V_in*(I_in.conjugate())
P_g = S_g.real
Q_g = S_g.imag
```

Resultados:

```
--------------------------------------------------------------------------------
Potência Ativa e Reativa fornecida pelo Gerador
--------------------------------------------------------------------------------
P: -51.969 MW
Q: 19.331 MVAR
--------------------------------------------------------------------------------
```

3.1.7 Potência Consumida pelas Cargas

Embora possui-se a potência complexa gerada pela fonte, é necessário calcular os valores da potência consumida pelas cargas. Com esse intuito, utiliza-se os valores da tensão na carga e a corrente na carga para calcular a potência ativa e reativa nas cargas.

Potência Consumida pelas Cargas:

```
#Cálculo da potência consumida pela carga 1
S_Z1 = complex(V_Z1*(I_Z1.conjugate()))
P_Z1 = S_Z1.real
Q_Z1 = S_Z1.imag
#Cálculo da potencia consumida pela carga 2
S_Z2 = complex(V_Z2*(I_Z2.conjugate()))
P_Z2 = S_Z2.real
Q_Z2 = S_Z2.imag
#Cálculo da potencia consumida pela carga 3
S_Z3 = complex(V_Z3*(I_Z3.conjugate()))
P_Z3 = S_Z3.real
Q_Z3 = S_Z3.imag
```

Resultados:

```
--------------------------------------------------------------------------------
Potência Ativa e Reativa Consumida pela carga Z1
--------------------------------------------------------------------------------
P_Z1: 11.099 MW
Q_Z1: 22.913 MVAR
--------------------------------------------------------------------------------
Potência Ativa e Reativa Consumida pela carga Z2
--------------------------------------------------------------------------------
P_Z2: 26.149 MW
Q_Z2: 53.924 MVAR
--------------------------------------------------------------------------------
Potência Ativa e Reativa Consumida pela carga Z3
--------------------------------------------------------------------------------
P_Z3: 10.400 MW
Q_Z3: 21.493 MVAR
--------------------------------------------------------------------------------
```

3.1.8 Potência Complexa Perdida

Em todo sistema elétrico existem percas de potência associada, uma vez que o sistema não é modelado idealmente. Então, é considerado as capacitâncias parasistas, o efeito ferrente, as resistências internas e outros parâmetros que tornam o sistema elétrico fidedigno com a realidade. Por conseguente, é interessante calcular essas perdas no sistema e isso é feito pela subtração da potência fornecida pelo gerador pela potência total das cargas.

Potência Complexa Perdida:

```
#Potência Complexa Perdida

S_TotalCargas = complex(S_Z1+S_Z2+S_Z3)

P_Perdida = abs(S_g.real)-abs(S_TotalCargas.real)
Q_Perdida = abs(abs(S_g.imag)-abs(S_TotalCargas.imag))

P_perG = abs(P_Perdida/S_g.real)*100
Q_perG = abs(Q_Perdida/S_g.imag)*100
```

Resultados:

```
--------------------------------------------------------------------------------
Potência Ativa e Reativa Perdida no Sistema
--------------------------------------------------------------------------------
P_Perdida: 4.321 MW (8.31%)
Q_Perdida: 78.999 MVAR (408.667%)
--------------------------------------------------------------------------------
```

3.1.9 Novos Valores de Impedância

Como os valores de perda do sistema foram relativamente altas, existe um erro de dimensionamento do sistema elétrico. Esse erro decorre da baixo valor das cargas em relação a tensão de operação do sistema, ou seja, é necessário alterar e aumentar os valores das cargas para que as perdas diminuam.

Em razão disso, foi definida uma nova função que simula todos os parâmetros do sistema novamente em função da nova impedância estipulada. Basicamente, essa função refaz o cálculo de todas as sessões anteriores com os novos valores de impedância. Além disso, foi criada também uma função resposável pelo calculo da perda de potência que tem que ser inferior a 10%.

Determinação de Impedâncias para Correção da Perda de Potência:

```
def calcularImpedanciaPot(Z):
    Z1_sim = Z[0]+(Z[1]*1j)
    Z2_sim = Z[2]+(Z[3]*1j)
    Z3_sim = Z[4]+(Z[5]*1j)
    Z_sim = np.array([Z1_sim,Z2_sim,Z3_sim])
    resultado = simularSistema(Z_sim)

    perda = np.array([resultado['P_Perdida'],resultado['Q_Perdida']])

    return max(perda)
lim = [(10,3e3),(10,3e3),(10,3e3),(10,3e3),(10,3e3),(10,3e3)]
resultado = opt.minimize(calcularImpedanciaPot,x0=Z_i, method = 'TNC', bounds=lim)
z1_new = complex(resultado.x[0],resultado.x[1])
z2_new = complex(resultado.x[2],resultado.x[3])
z3_new = complex(resultado.x[4],resultado.x[5])
z_new = np.array([z1_new,z2_new,z3_new])
Sim = simularSistema(z_new)
```

Resultados:

```
--------------------------------------------------------------------------------
Impedâncias Determinadas para Correção na Perda de Potência
--------------------------------------------------------------------------------
Z1 = 863.039+1751.831j Ω
Z2 = 294.195+294.897j Ω
Z3 = 137.242+64.969j Ω
--------------------------------------------------------------------------------
Potência Perdida
--------------------------------------------------------------------------------
P: 9.361%
Q: 9.361%
--------------------------------------------------------------------------------
```

3.1.10 Ajuste no TAP dos Transformadores

Sob outro prisma, existe a necessidade de fazer o ajuste do TAP para que as tensões aplicadas nas cargas sejam, respectivamente, de 230kV, 138kV e 69kV. Logo, assim como no cálculo das novas impedâncias, o sistema é simulado novamente com os novos ajustes

do TAP e recalculado os valores das tensões nas cargas.

Ajuste de Tap dos Transformadores:

```
new_Tap = np.array([3.5546, 0.64297, 0.6788])
Sim_newTap = simularSistema_newTap(new_Tap)
V_Z1_newTapCalc = abs(Sim_newTap['V_Z1'])
V_Z2_newTapCalc = abs(Sim_newTap['V_Z2'])
V_Z3_newTapCalc = abs(Sim_newTap['V_Z3'])
erros = np.array([
erro_rel(V_nominal[0],V_Z1_newTapCalc),
erro_rel(V_nominal[1],V_Z2_newTapCalc),
erro_rel(V_nominal[2],V_Z3_newTapCalc)
])
```

Resultados:

```
--------------------------------------------------------------------------------
Novas Relações de Transformação
--------------------------------------------------------------------------------
n1: 3.5546
n2: 0.64297
n3: 0.6788
--------------------------------------------------------------------------------
Tensões Simuladas
--------------------------------------------------------------------------------
V_Z1 = 229.751 kV (0.108%)
V_Z2 = 137.994 kV (0.004%)
V_Z1 = 68.882 kV (0.171%)
--------------------------------------------------------------------------------
```

3.1.11 Inserção do Banco de Capacitores em Derivação

Por fim, o último aspecto analisado foi a inserção do banco de capacitores em derivação para ajustar as tensões na carga, respectivamente, de 230kV, 138kV e 69kV com uma tolerância de 5% para mais ou para menos.

Banco de Capacitores:

```
C = np.array([6.11787478e-07, 5.24790523e-06, 1.91926849e-05])
z1_new = (z1*(1/(w*C[0]*j)))/((z1+(1/(w*C[0]*j))))
z2_new = (z2*(1/(w*C[1]*j)))/((z2+(1/(w*C[1]*j))))
z3_new = (z3*(1/(w*C[2]*j)))/((z3+(1/(w*C[2]*j))))
z_new = np.array([z1_new,z2_new,z3_new])
simulCalc = simularSistema(z_new)
erro_relativo = np.array([
erro_rel(V_nominal[0],abs(simulCalc['V_Z1'])),
erro_rel(V_nominal[1],abs(simulCalc['V_Z2'])),
erro_rel(V_nominal[2],abs(simulCalc['V_Z3']))
])
```

Resultados:

```
--------------------------------------------------------------------------------
Valores de Capacitores Determinados
--------------------------------------------------------------------------------
C1 = 0.612e-6 F
C2 = 5.248e-6 F
C3 = 19.193e-6 F
--------------------------------------------------------------------------------
Tensão Simulada com a adição dos Capacitores em Derivação
--------------------------------------------------------------------------------
V_Z1 = 230.398 kV
V_Z2 = 137.759 kV
V_Z3 = 69.113 kV
--------------------------------------------------------------------------------
Erro (V_Z1): 0.173%
Erro (V_Z2): 0.175%
Erro (V_Z3): 0.164%
--------------------------------------------------------------------------------
```

3.2 Modelagem no LTspice

Para simular o sistema elétrico do projeto ultilizando o LTspice foram usados os modelos da figura 4 e da figura 5. Com esses modelos podemos representar as linhas de transmissão e os transformadores de potência de forma compatível com o LTspice.

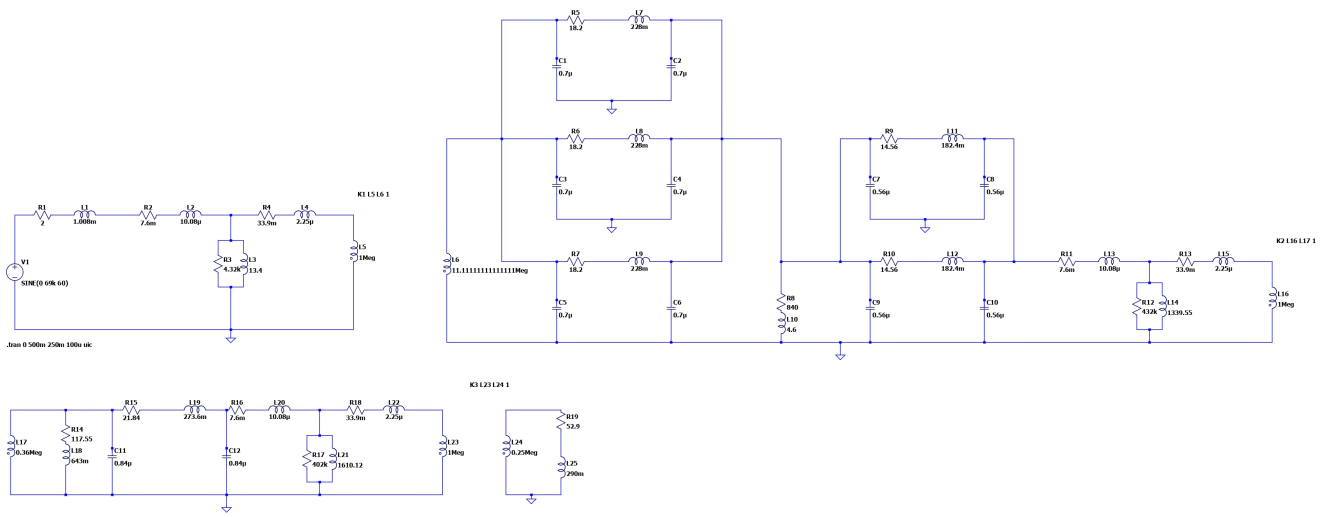
Para as linhas de transmissão os parâmetros foram os seguintes: $R = 0,182\Omega/km$, $L = 2,28mH/km$ e $C = 0,0140\mu F/km$. O comprimento das linhas 1 a 3 é de $100km$, as linhas 4 e 5 tem $80km$ e a linha 6 possui $120km$.

Os parâmetros dos transformadores foram: $R_1 = 7,6m\Omega$, $X_1 = 3,8m\Omega$, $R_2 = 33,9m\Omega$, $X_2 = 0,85m\Omega$, $R_{m1} = 4320\Omega$, $X_{m1} = 5050$, $R_{m2} = 432000\Omega$, $X_{m2} = 505000\Omega$, $R_{m3} = 402000$ e $X_{m3} = 607000\Omega$. Para o primeiro transformador temos $n = \frac{10}{3}$, para o segundo temos $n = 0,6$ e para o terceiro temos $n = 0,5$.

No inicio do circuito temos um gerador com $V = 69kVrms$ e uma impedância série $R_f = 2\Omega$ e $X_f = 0,38\Omega$. Já para as cargas, não foram informadas as impedâncias, mas usando a tensão nominal do secundário dos transformadores e as potências complexas de cada carga encontramos as impedâncias: $Z_1 = 840 + j4,6$, $Z_2 = 117,55 + j643m$ e $Z_3 = 52,9 + j290m$.

Com todas essas informações foi possivel construir todo o circuito da rede elétrica no LTspice. Na figura abaixo podemos observar todo o circuito

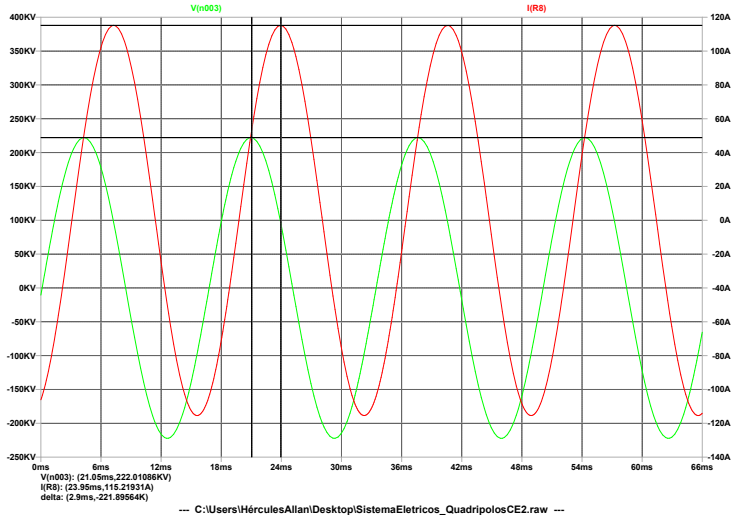
Figura 9: Sistema Elétrico no LTspice



Fonte: Autoria Própria, 2025

Afim de verificar os resultados obtidos no código em python das tensões e correntes das cargas Z_1, Z_2 e Z_3 foram feitas medições de tensão e corrente sobre R_8 e L_{10} para obter os valores de Z_1, R_{14} e L_{18} para obter os valores de Z_2 e R_{19} e L_{25} para obter os valores de Z_3.

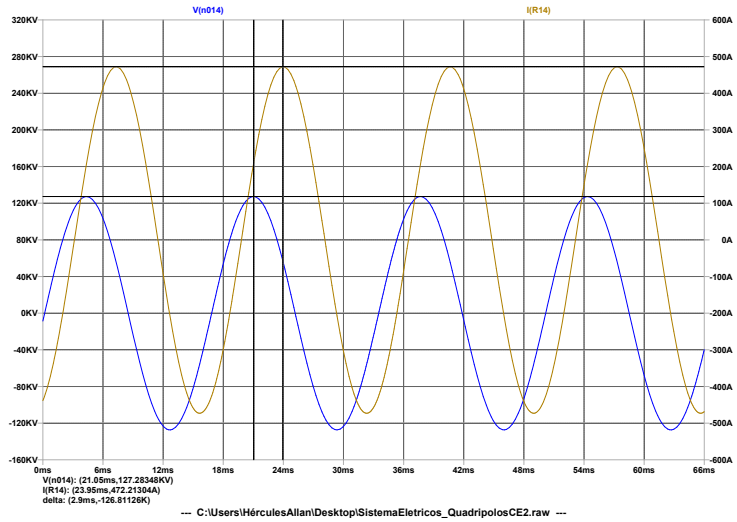
Figura 10: Osciloscópio do LTspice para tensão e corrente na carga Z1



Fonte: Autoria Própria, 2025

Assim, observando o gráfico das ondas de Z_1 temos uma tensão de aproximadamente $222kVrms$ passando pelo zero em aproximadamente $126,7\mu s$ e temos uma corrente de aproximadamente $115,2A$ passando pelo zero em aproximadamente $3,1ms$. Chegando por fim a uma tensão de $222kV\angle -2,72°$ e uma corrente de $115,2A\angle -66,99°$.

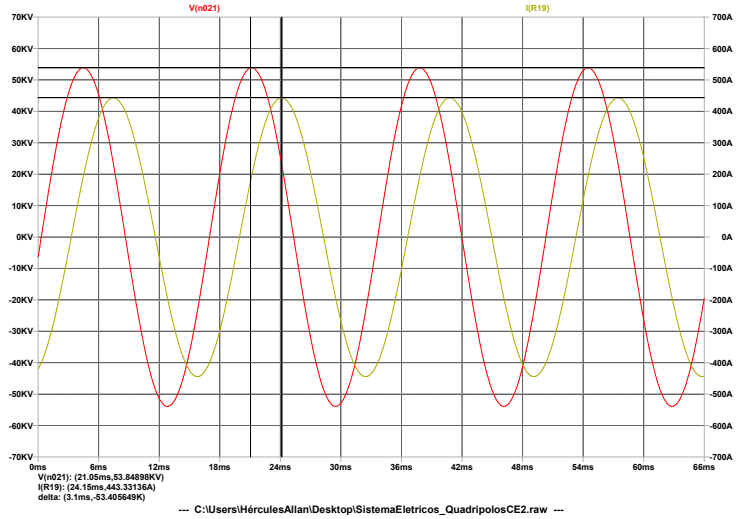
Figura 11: Osciloscópio do LTspice para tensão e corrente na carga Z2



Fonte: Autoria Própria, 2025

Observando o gráfico das ondas de Z_2 temos uma tensão de cerca de $127,28kV$ passando no zero em $176,97\mu s$e uma corrente de $472,2A$ passando no zero em $3,15ms$. Ou seja, temos uma tensão de $127,28kV\angle -3,83°$ e uma corrente de $472,2A\angle -68,07°$.

Figura 12: Osciloscópio do LTspice para tensão e corrente na carga Z3



Fonte: Autoria Própria, 2025

Agora por meio do gráfico das ondas de Z_3 encontramos uma tensão de $53,8kVrms$ passando no zero em $311,46\mu s$ e uma corrente de $443,3A$ passando no zero em $3,29ms$. Obtendo uma tensão de $53,8kV\angle -6,71°$ e uma corrente de $443,3A\angle -71,48°$.

4 Conclusão

Em síntese, a realização do projeto proporcionou a aplicação direta da teoria de quadripolos nos estudos de sistemas elétricos reais, evidenciando a eficiência da análise de sistemas a partir da matriz de parâmetros de transmissão.

Os resultados obtidos pelos algoritmos em Python revelam que a teoria apresentada se comporta de maneira semelhante aos modelos representados por meio da simulação no LTSpice, apesar das diferenças oriundas dos métodos adotados para modelagem e cálculo da simulação do sistema elétrico. Junto a isso, a plotagem do diagrama fasorial poligonal facilitou a observação dos parâmetros do circuito em relação às cargas, além disso, os experimentos de comissionamento de cargas e banco de capacitores para melhorar a eficiência do sistema se mostraram triviais em relação aos métodos algébricos comumente usados.

Por conseguinte, o desenvolvimento do projeto contribuiu para a prática e fixação da teoria apresentada em sala de aula, dando uma experiência semelhante ao comissionamento de sistemas elétricos reais aplicados pelos profissionais atuantes do mercado.

5 Referências Bibliográficas

Referências

[1] GLOVER, J. Duncan; SARMA, Mulukutla S.; OVERBYE, Thomas J. *Power system analysis and design*. 5ª edição. Stamford: Cengage Learning, 2012.

[2] SADIKU, M. N. O. *Fundamentos de Circuitos Elétricos*. 5ª edição – 2013. Editora Bookman.