



SIMULADOR DE CIRCUITOS ELÉTRICOS DE PEQUENO PORTE UTILIZANDO MODELAGEM ORIENTADA A OBJETOS

Daniel Moura de Souza

Projeto de Graduação apresentado ao Curso de Engenharia Elétrica da Escola Politécnica, Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Engenheiro.

Orientador: Antonio Carlos Siqueira de Lima

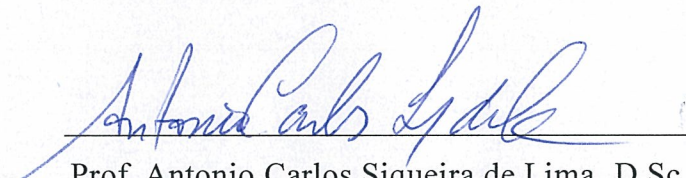
Rio de Janeiro
Setembro de 2011

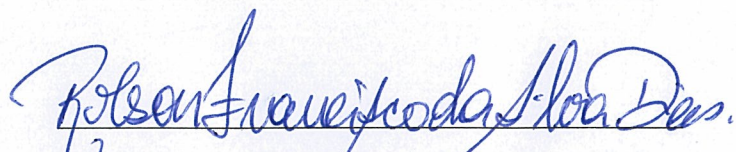
SIMULADOR DE CIRCUITOS ELÉTRICOS DE PEQUENO PORTE
UTILIZANDO MODELAGEM ORIENTADA A OBJETOS

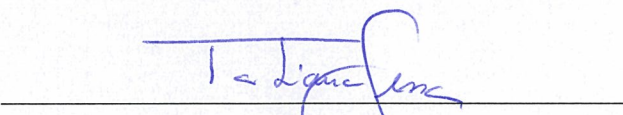
Daniel Moura de Souza

PROJETO DE GRADUAÇÃO SUBMETIDA AO CORPO DOCENTE DO CURSO DE ENGENHARIA ELÉTRICA DA ESCOLA POLITÉCNICA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE ENGENHEIRO ELETRICISTA.

Aprovada por:


Prof. Antonio Carlos Siqueira de Lima, D.Sc.
(Orientador)


Prof. Robson Francisco da Silva Dias, D.Sc.


Prof.ª Tatiana Mariano Lessa de Assis, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
SETEMBRO DE 2011

Souza, Daniel Moura de.

Simulador de Circuitos Elétricos de Pequeno Porte Utilizando Modelagem Orientada a Objetos/ Daniel Moura de Souza. – Rio de Janeiro: UFRJ/ Escola Politécnica, 2011.

X, 67 p.: il.; 29,7 cm.

Orientador: Antonio Carlos Siqueira de Lima

Projeto de Graduação – UFRJ/ Escola Politécnica/ Curso de Engenharia Elétrica, 2011.

Referências Bibliográficas: p. 58.

1. Power Systems Elettromagnetic Transients Simulation.
2. Creating an Elettromagnetic Transients Program in MATLAB: MatEMTP.
3. Análise de Sistemas de Potência
4. Modelagem e Projetos Baseados em Objetos.
5. Modelagem de Objetos através da UML – Análise e Desenho Orientados a Objeto.
6. Desenvolvimento de um Sistema Computacional Orientado a Objetos para Sistemas Elétricos de Potência.
7. Fundamentos de Circuitos Elétricos. I. de Lima, Antonio Carlos Siqueira. II. Universidade Federal do Rio de Janeiro, Escola Politécnica, Curso de Engenharia Elétrica. III. Título.

Seja bendito o nome de Deus para todo sempre,

porque dele é a sabedoria e a força.

Daniel Cap. 2, Vers. 20

A minha esposa Tais e

ao meu filho Pedro.

Agradecimentos

Em primeiro lugar, agradeço ao Deus da minha salvação, por Seu amor e misericórdia para com a minha vida, me concedendo a oportunidade de estudar na UFRJ. Agradeço também a minha família pelo apoio dado em todos os momentos, e em especial a minha irmã que mesmo diante das tempestades da vida, nunca me negou apoio, e quando muitos diziam que não ia dar certo, acreditou em meu potencial.

Agradeço a minha esposa e meu filho pelo amor, carinho e paciência demonstrados nas muitas horas dedicadas à realização desse trabalho.

Agradeço também ao professor Antonio Carlos Siqueira de Lima, pela atenção dispensada durante a confecção desse trabalho.

Agradeço a todo corpo docente do Departamento de Engenharia Elétrica.

Agradeço aos amigos que fiz durante este período da graduação. Citando algum nome, acabaria fazendo injustiça com tantos outros que também compartilharam bons momentos ao longo desses anos.

A todos que contribuíram direta ou indiretamente para a realização deste trabalho. Obrigado!

Resumo do Projeto de Graduação apresentado à Escola Politécnica/ UFRJ como parte dos requisitos necessários para a obtenção do grau de Engenheiro Eletricista.

Simulador de Circuitos Elétricos de Pequeno Porte
Utilizando Modelagem Orientada a Objetos

Daniel Moura de Souza

Setembro de 2011

Orientador: Antonio Carlos Siqueira de Lima

Curso: Engenharia Elétrica

Este trabalho tem por objetivo desenvolver um simulador de circuitos elétricos de pequeno porte, com a aplicação da Modelagem Orientada a Objetos aos elementos de circuito que compõem esses circuitos. A partir do *software* Matlab, os códigos foram desenvolvidos baseados na modelagem dos elementos de circuito descritos no EMTP Theory Book, e implementados através de classes de objetos para cada categoria: elementos passivos, fontes de tensão e de corrente, chaves e linhas de transmissão. Este trabalho também apresenta os resultados obtidos para diversas simulações em determinados circuitos diferentes, mostrando as formas de onda dos sinais de tensão e corrente nas barras destes circuitos, bem como nos casos em que existem chaves no mesmo. Adicionalmente, essa ferramenta visa oferecer aos estudantes de engenharia elétrica, uma oportunidade de ter contato com um ambiente de simulação produzido dentro de um *software* bastante difundido no meio acadêmico, dando espaço para novas atualizações/modificações e despertando-lhes o interesse por diversos assuntos da área como, transitórios eletromagnéticos, método nodal de solução de circuitos e programação orientada a objetos.

Palavras-chave: Modelagem Orientada a Objetos, Matlab, EMTP, Simulação.

Abstract of Undergraduate Project presented to POLI/UFRJ as a partial fulfillment of the requirements for the degree of Engineer.

Electrical Circuits Simulator Small
Using Object-Oriented Modeling

Daniel Moura de Souza

September/2011

Advisor: Antonio Carlos Siqueira de Lima

Course: Electrical Engineering

This work aims to develop a simulator of electrical circuits of small size, with the application of object-oriented modeling to the circuit elements that make up these circuits. From the Matlab software, the codes were developed based on modeling the circuit elements described in EMTP Theory Book, and implemented through object classes for each category: passive elements, voltage and current sources, switches and transmission lines. This paper also presents the results for several different circuits in certain simulation, showing the waveforms of voltage and current signals in the buses of these circuits, and where there are keys in it. Additionally, this tool aims to offer electrical engineering students, an opportunity to have contact with a simulation environment produced within software widespread in academia, giving space for new updates/changes and arousing their interest in various issues of the area as transient electromagnetic, method nodal solution of circuits, and object-oriented programming.

Keywords: Object-Oriented Modeling, Matlab, EMTP, Simulation.

Índice

Capítulo 1. Introdução	1
1.1. Comentários Iniciais	1
1.2. Objetivo	2
1.3. Estrutura do Trabalho	2
Capítulo 2. Modelagem dos Elementos de Circuito	4
2.1. Elementos Passivos	6
2.1.1. Resistor	6
2.1.2. Indutor	7
2.1.3. RL Série	8
2.1.4. Capacitor	10
2.1.5. RLC Série	11
2.2. Chaves	13
2.3. Fontes de Tensão ou de Corrente	13
2.4. Linhas de Transmissão	14
Capítulo 3. O Método de Solução Nodal Modificado	18
3.1. Origem – Método Nodal	18
3.1.1. Montagem da Matriz Nodal	18
3.2. Nodal Modificado	20
3.2.1. Montagem da Matriz Nodal Modificada	20
Capítulo 4. A Modelagem Orientada a Objetos Aplicada aos Elementos de Circuito	25
4.1. Introdução	25
4.2. O Conceito de Objeto	25
4.3. Herança, Polimorfismo e Encapsulamento	26
4.4. Representação das Classes de Objetos	27
4.5. Classes de Elementos de Circuito	28
4.5.1. Classe <i>Source</i>	28
4.5.2. Classe <i>RampSource</i>	29
4.5.3. Classe <i>PassiveElement</i>	30
4.5.4. Classe <i>Switch</i>	31
4.5.5. Classe <i>TransLine</i>	32
4.5.6. Classe <i>PowerSystem</i>	33

4.6.	Diagrama de Classes	35
4.7.	Fluxograma	36
Capítulo 5.	Resultados e Conclusões	37
5.1.	Circuito RL Série	37
5.2.	Circuito RLC Série	40
5.3.	Circuito com 3 Barras	44
5.4.	Circuito com 6 Barras e 2 Chaves.....	47
5.5.	Circuito Trifásico com 3 Linhas de Transmissão	53
5.6.	Conclusões e Sugestões para Trabalhos Futuros	57
	Referências Bibliográficas.....	58
	Anexo I.....	59
	Anexo II.....	61
	Anexo III	63
	Anexo IV	65
	Anexo V.....	67

Capítulo 1. Introdução

1.1. Comentários Iniciais

Com a crescente expansão do sistema elétrico, se faz cada vez mais necessário o uso de simuladores computacionais capazes de auxiliar nas atividades de operação e planejamento dos sistemas de energia. Neste sentido, são desenvolvidos diversos programas em várias linguagens de programação de alto nível, tais como Fortran (maioria), C e Pascal. No entanto a maioria destas linguagens apresentam algumas desvantagens como:

- Manutenção e atualização com alto custo;
- Fraca estrutura de dados quanto à manutenção dos códigos;
- Os códigos já escritos possuem pouca, ou nenhuma possibilidade de reutilização dos mesmos;
- Dificil integração com aplicativos diferentes.

Entretanto, devido à velocidade da expansão supracitada, torna-se imprescindível a atualização e manutenção de tais programas, o que leva a uma crescente demanda por soluções cada vez mais rápidas e confiáveis. Daí surge a necessidade da utilização da Modelagem Orientada a Objetos (MOO) que visa produção de *softwares* mais consistentes, facilitando a reutilização dos códigos, atualização e manutenção. Esta modelagem também apresenta outras vantagens em relação ao enfoque tradicional de modelagem para construção de sistemas computacionais, tais como:

- Modelagem do sistema mais próximo à visão conceitual do mundo real;
- Maior facilidade na passagem da fase de modelagem para a fase de implementação, devido à transparência das informações utilizadas (objetos modelados).

Esta modelagem também requer do programador alguns cuidados necessários como a necessidade de conhecimento mais profundo dos objetos a serem modelados, ou seja, a definição da fronteira entre o que o objeto possui (atributos) e o que o objeto faz (métodos). O capítulo 4 trás maiores detalhes sobre a MOO.

1.2. Objetivo

A partir da modelagem elétrica de cada elemento de circuito, a saber: resistor, indutor e capacitor, fontes de tensão e de corrente, chaves e linhas de transmissão, a MOO será implementada para cada objeto em questão, a fim de desenvolver um ambiente de simulação de circuitos elétricos de pequeno porte utilizando o *software* MATLAB.

Os resultados das simulações fornecerão as formas de onda das tensões e correntes nas barras de um determinado circuito em estudo.

1.3. Estrutura do Trabalho

O presente texto está subdividido em cinco capítulos, pela seguinte ordem:

O Capítulo 1 trás uma breve introdução, apresentado o objetivo principal do trabalho.

O Capítulo 2 apresenta a Modelagem dos Elementos de Circuito, em que são descritas as equações algébricas e modelos aplicados a cada elemento de circuito, no domínio do tempo.

O Capítulo 3 faz alusão ao método Nodal Modificado utilizado na análise de sistemas elétricos. Onde será descrito todo o processo de montagem da matriz de admitância nodal modificada necessária para a obtenção da resposta final da simulação.

O Capítulo 4 apresenta a Modelagem Orientada a Objetos Aplicada aos Elementos de Circuito. Todas as classes de objetos, associados a cada elemento de

circuitos utilizados nos programas são apresentados nesse capítulo, bem como suas relações de interdependência.

Finalmente, o Capítulo 5 apresenta os resultados obtidos das simulações e as conclusões, bem como sugestões para trabalhos futuros.

Os arquivos de entrada com os dados de cada sistema simulado, bem como o código utilizado no simulador, encontram-se nos anexos deste trabalho.

Capítulo 2. Modelagem dos Elementos de Circuito

A seguir serão apresentados os modelos dos elementos de rede [1] utilizados no desenvolvimento do programa. Considere o trecho de um circuito conforme figura abaixo:

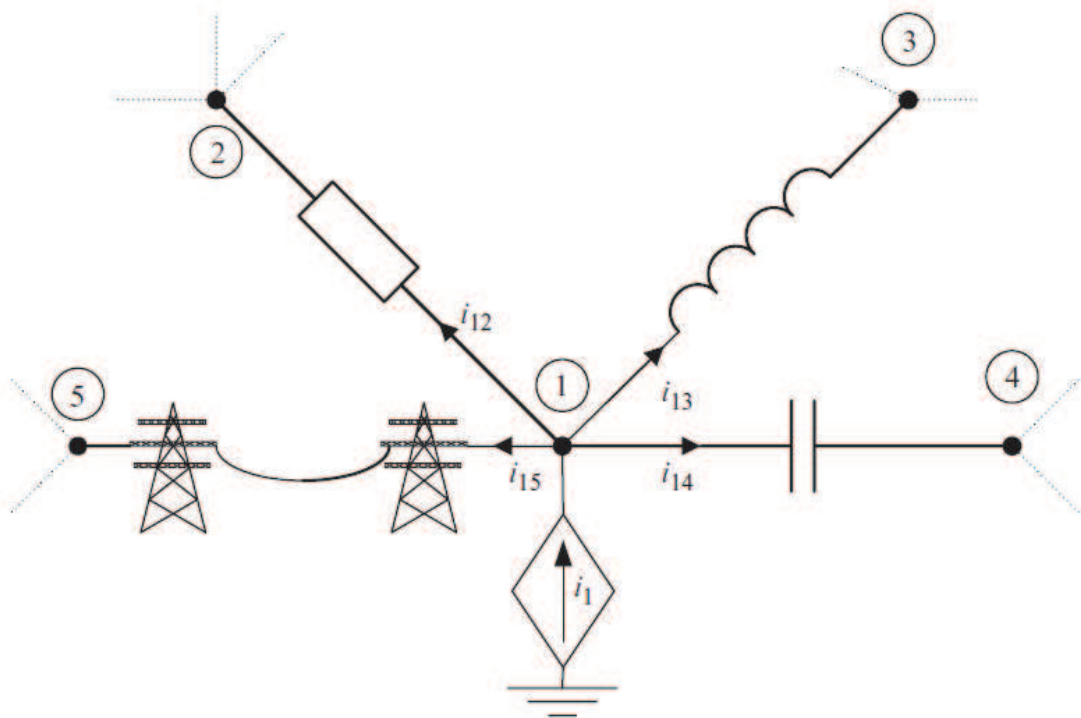


Figura 2.1 – Sistema elétrico genérico

Para a resistência entre os nós 1 e 2, pode-se escrever

$$i_{12}(t) = \frac{1}{R} \{v_1(t) - v_2(t)\} \quad (2.1)$$

No caso da indutância, é sabido que:

$$v = L \frac{di}{dt} \quad (2.2)$$

Que pode ser trocada pela equação a diferenças centrais aproximada [1]:

$$\frac{v(t)+v(t+\Delta t)}{2} = L \frac{i(t)-i(t-\Delta t)}{\Delta t} \quad (2.3)$$

Equação esta, que também pode ser obtida pela regra de integração trapezoidal [1]:

$$i(t) = i(t-\Delta t) + \frac{1}{L} \int_{t-\Delta t}^t \{v_k(u) - v_m(u)\} du \quad (2.4)$$

Assim, a partir da equação (2.3) o seguinte resultado é encontrado:

$$i_{13}(t) = i_{13}(t-\Delta t) + \frac{\Delta t}{2L} \{v_1(t) - v_3(t) + v_1(t-\Delta t) - v_3(t-\Delta t)\} \quad (2.5)$$

Que pode ser reescrito da seguinte forma:

$$i_{13} = \frac{\Delta t}{2L} \{v_1(t) - v_3(t)\} + hist_{13}(t-\Delta t) \quad (2.6)$$

onde o termo $hist_{13}(t-\Delta t)$, corrente histórica, é igual à:

$$hist_{13}(t-\Delta t) = i_{13}(t-\Delta t) + \frac{\Delta t}{2L} \{v_1(t-\Delta t) - v_3(t-\Delta t)\} \quad (2.7)$$

De forma análoga, pode-se escrever para o capacitor:

$$i_{14} = \frac{2C}{\Delta t} \{v_1(t) - v_4(t)\} + hist_{14}(t-\Delta t) \quad (2.8)$$

Da mesma forma, a corrente histórica será:

$$hist_{14}(t-\Delta t) = -i_{14}(t-\Delta t) - \frac{2C}{\Delta t} \{v_1(t-\Delta t) - v_4(t-\Delta t)\} \quad (2.9)$$

Para o caso da linha de transmissão entre os nós 1 e 5, considerando o modelo com parâmetros distribuídos sem perdas, temos a impedância característica Z_c constante definida por:

$$Z_c = \sqrt{\frac{L}{C}} \quad (2.10)$$

Assim, a equação da corrente no ramo em função do mesmo será:

$$i_{15}(t) = \frac{1}{Z_c} v_1(t) + hist_{15}(t - \tau) \quad (2.11)$$

onde τ representa o tempo de propagação da onda, que pode ser calculado pela relação entre o comprimento da linha l , e a velocidade de propagação da onda c , ou seja:

$$\tau = \frac{l}{c} \quad (2.12)$$

onde c é definido por:

$$c = \frac{l}{\sqrt{LC}} \quad (2.12a)$$

Com base no exposto acima, cada elemento de circuito pode ser reescrito de uma forma mais geral, conforme a seguir.

2.1. Elementos Passivos

2.1.1. Resistor

A figura 2.2 mostra a simbologia equivalente do resistor

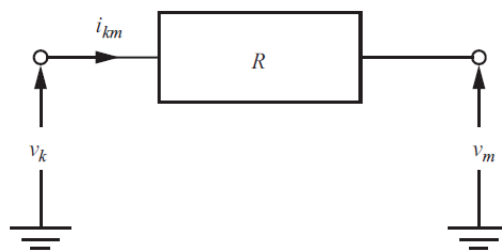


Figura 2.2 – Resistor

De onde podemos retirar a seguinte expressão:

$$i_{km}(t) = g_R \{v_k(t) - v_m(t)\} \quad (2.14)$$

onde:

$$g_R = \frac{1}{R} \quad (2.14a)$$

2.1.2. Indutor

Como pode ser visto na figura 2.3, para o indutor foi demonstrado que:

$$i_{km}(t) = g_L \{v_k(t) - v_m(t)\} + I_{L\ History}(t - \Delta t) \quad (2.15)$$

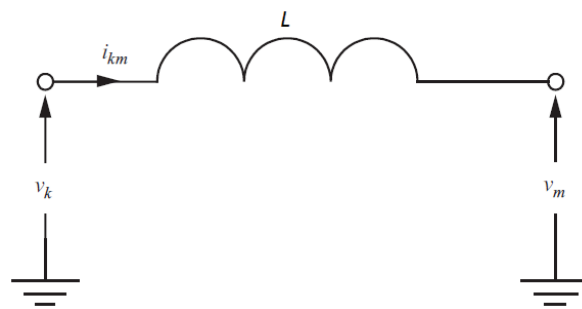


Figura 2.3 – Indutor

onde o termo $I_{L\ History}(t - \Delta t)$, corrente histórica, é igual à:

$$I_{L\ History}(t - \Delta t) = i_{km}(t - \Delta t) + g_L \{v_k(t - \Delta t) - v_m(t - \Delta t)\} \quad (2.16)$$

Agora, o indutor pode ser representado por uma resistência $R_{eff} = \frac{1}{g_L} = \frac{2L}{\Delta t}$, em paralelo com a fonte de corrente histórica, conforme figura 2.4 a seguir:

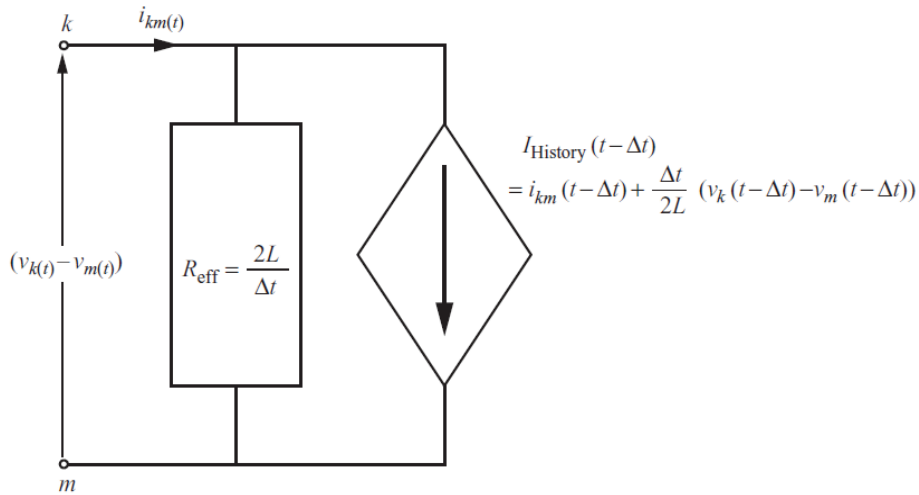


Figura 2.4 – Indutor equivalente

2.1.3. RL Série

A associação em série da resistência com a indutância, figura 2.5, pode ser equacionada da forma como se segue:

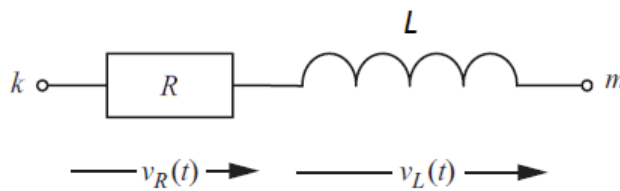


Figura 2.5 – Ramo RL série

Sabe-se que a tensão v_{km} obedece a seguinte expressão:

$$v_k - v_m = v_R + v_L \quad (2.17)$$

Substituindo as equações (2.15) e (2.16) na equação (2.17), temos:

$$v_k(t) - v_m(t) = \left(R + \frac{2L}{\Delta t} \right) i_{km}(t) - \frac{2L}{\Delta t} I_{L\text{History}}(t - \Delta t) \quad (2.18)$$

Desta forma, a topologia do ramo km equivalente fica como na figura 2.6:

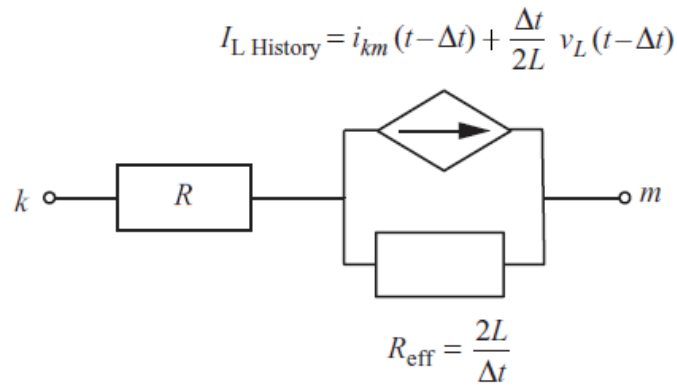


Figura 2.6 – Ramo RL série

Rearranjando os termos da equação (2.18), vem:

$$i_{km}(t) = g_{RL} \{v_k(t) - v_m(t)\} + I_{RL \text{ History}}(t - \Delta t) \quad (2.19)$$

E a corrente histórica é:

$$I_{RL \text{ History}}(t - \Delta t) = g_{RL} \left\{ \left(\frac{2L}{\Delta t} - R \right) i_{km}(t - \Delta t) + v_k(t - \Delta t) - v_m(t - \Delta t) \right\} \quad (2.19b)$$

Agora, a resistência efetiva do ramo km obedece a seguinte expressão:

$$R_{eff}' = \frac{1}{g_{RL}} = R + \frac{2L}{\Delta t} \quad (2.20)$$

Finalmente, tem-se a representação final para o caso RL série conforme figura 2.7:

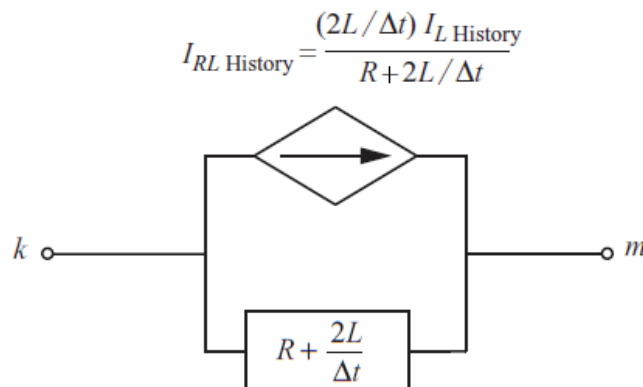


Figura 2.7 – Ramo RL série equivalente

2.1.4. Capacitor

A figura 2.8 representa um capacitor conectado entre as barras k e m :

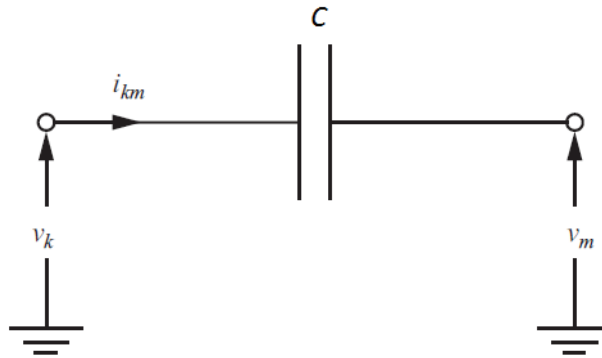


Figura 2.8 – Capacitor

De forma análoga ao indutor, para o capacitor podemos escrever:

$$i_{km}(t) = g_C \{v_k(t) - v_m(t)\} + I_{C \text{ History}}(t - \Delta t) \quad (2.21)$$

Da mesma forma, a corrente histórica $I_{C \text{ History}}(t - \Delta t)$ será:

$$I_{C \text{ History}}(t - \Delta t) = -i_{km}(t - \Delta t) - g_C \{v_k(t - \Delta t) - v_m(t - \Delta t)\} \quad (2.21a)$$

Logo, o capacitor também pode ser representado por uma resistência

$R_{eff} = \frac{1}{g_C} = \frac{\Delta t}{2C}$, em paralelo com a fonte de corrente histórica, como ilustra a

figura 2.9:

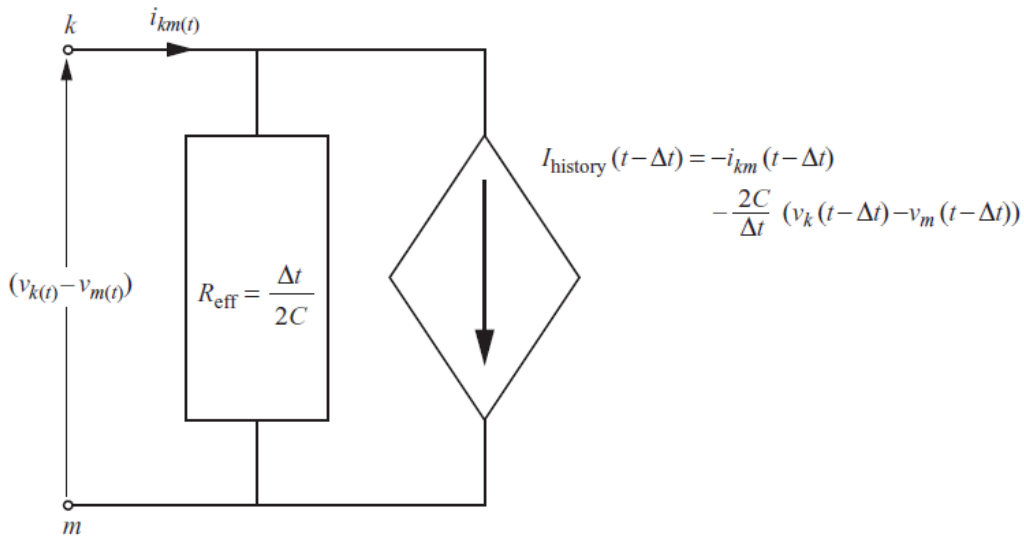


Figura 2.9 – Capacitor equivalente

2.1.5. RLC Série

Todos os ramos de circuito que contêm um ou mais elementos de rede como resistores, indutores e capacitores, podem ser considerados como uma combinação em série dos mesmos. A figura 2.10 mostra a associação em série destes três elementos:

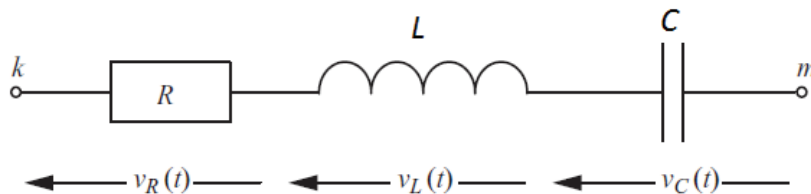


Figura 2.10 – Ramo RLC série

Sabe-se que a tensão v_{km} obedece a seguinte expressão:

$$v_k - v_m = v_R + v_L + v_C \quad (2.22)$$

Substituindo as equações (2.1), (2.6) e (2.8), na equação acima, e organizando os termos, vem:

$$v_k(t) - v_m(t) = \left(R + \frac{2L}{\Delta t} + \frac{\Delta t}{2C} \right) i_{km}(t) - \frac{2L}{\Delta t} I_{L\text{History}}(t - \Delta t) - \frac{\Delta t}{2C} I_{C\text{History}}(t - \Delta t) \quad (2.23)$$

onde as correntes históricas $I_{L\ History}(t-\Delta t)$ e $I_{C\ History}(t-\Delta t)$ obedecem as equações (2.16) e (2.21a), respectivamente.

Desta forma, a topologia do ramo km equivalente ficará como na figura 2.11:

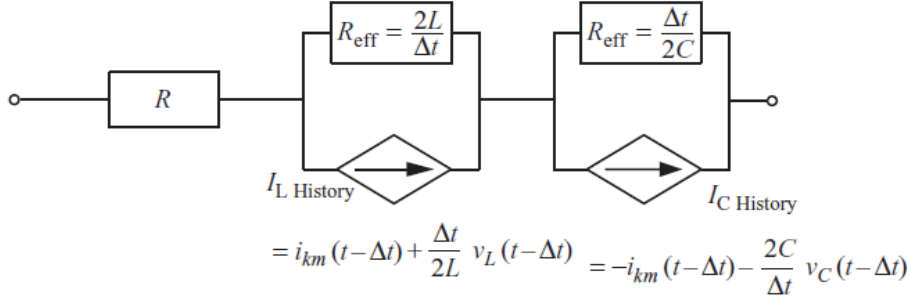


Figura 2.11 – Ramo RLC série

Rearranjando os termos, vem:

$$i_{km}(t) = G_{series} \{v_k(t) - v_m(t)\} + I_{RLC\ History}(t-\Delta t) \quad (2.24)$$

e a corrente histórica:

$$I_{RLC\ History}(t-\Delta t) = G_{series} \left\{ \left(\frac{2L}{\Delta t} - R - \frac{\Delta t}{2C} \right) i_{km}(t-\Delta t) + v_k(t-\Delta t) - v_m(t-\Delta t) - 2v_c(t-\Delta t) \right\} \quad (2.24a)$$

Com

$$R_{eff} = \frac{1}{G_{series}} = R + \frac{2L}{\Delta t} + \frac{\Delta t}{2C} \quad (2.24b)$$

Para atualizar o termo histórico da equação (2.24a), que também é dependente da tensão no capacitor, primeiro é calculada a corrente da equação (2.24), depois é obtida uma nova tensão no capacitor pela equação (2.25) a seguir:

$$v_c(t) = v_c(t-\Delta t) + \frac{\Delta t}{2C} \{i_{km}(t) + i_{km}(t-\Delta t)\} \quad (2.25)$$

A topologia final do ramo RLC série fica como na figura 2.12:

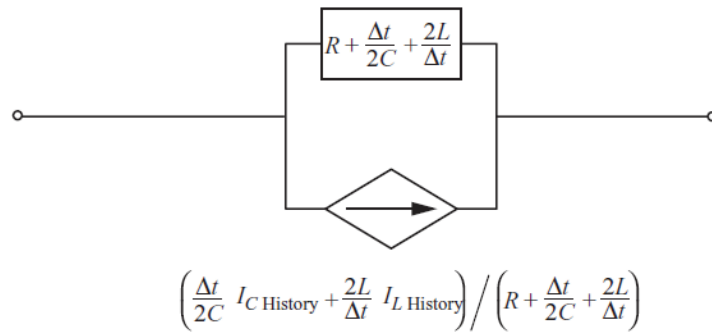


Figura 2.12 – Ramo RLC série equivalente

2.2. Chaves

Uma chave ideal possui dois estados: aberto ou fechado, conforme a figura 2.13 abaixo:

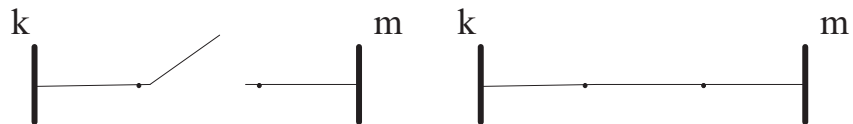


Figura 2.13 – chave aberta (a) e chave fechada (b)

Equacionando [2] para a chave fechada

$$v_k(t) - v_m(t) = 0 \text{ e } i_{km}(t) \neq 0 \quad (2.26)$$

E para a chave estiver aberta, segue:

$$v_k(t), v_m(t) \neq 0 \text{ e } i_{km}(t) = 0 \quad (2.27)$$

2.3. Fontes de Tensão ou de Corrente

Elementos ativos podem ser fontes de tensão ou corrente, tanto dependentes, com independentes. As fontes de tensão/corrente são modeladas de maneira convencional, conforme as figuras abaixo:

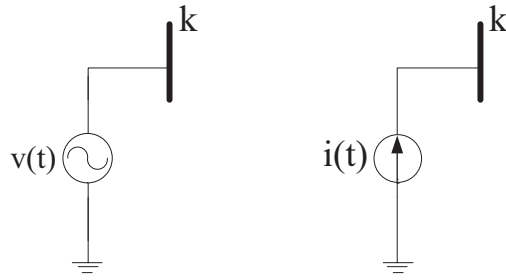


Figura 2.14 – fonte de tensão (a) e fonte de corrente (b)

Onde $v(t)$ e $i(t)$ podem ser funções senoidais, que possuem amplitude $G\sqrt{2}$, frequência f e fase ϕ :

$$g(t) = G\sqrt{2}\text{sen}(2\pi ft + \phi) \quad (2.28)$$

No caso trifásico, as tensões e/ou correntes das três fases estarão defasadas de 120° entre si, ou seja

$$\begin{aligned} g_a(t) &= G\sqrt{2}\text{sen}(2\pi ft) \\ g_b(t) &= G\sqrt{2}\text{sen}(2\pi ft - 120^\circ) \\ g_c(t) &= G\sqrt{2}\text{sen}(2\pi ft + 120^\circ) \end{aligned} \quad (2.29)$$

Dentro do escopo do deste trabalho, haverá casos em que as fontes de tensão e/ou corrente fornecem sinais em rampa, degraus, etc. Isso será mostrado mais adiante no capítulo 4.

2.4. Linhas de Transmissão

Considerando o modelo para linhas de transmissão como o de parâmetros distribuídos segundo a figura 2.14:

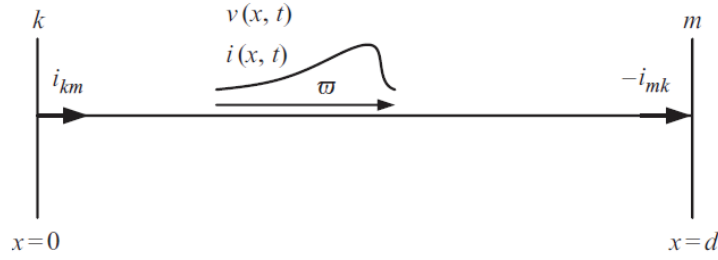


Figura 2.14 – Linha de transmissão com parâmetros distribuídos

Onde:

- d é o comprimento da linha;
- w é a velocidade de propagação da linha.

E, assumindo L' e C' como a indutância e a capacitância por unidade de comprimento respectivamente, as equações de propagação da onda [1] para essa linha podem ser escritas da seguinte forma:

$$-\frac{\partial v(x,t)}{\partial x} = L' \frac{\partial i(x,t)}{\partial t} \quad (2.30)$$

$$-\frac{\partial i(x,t)}{\partial x} = C' \frac{\partial v(x,t)}{\partial t} \quad (2.31)$$

Que têm como solução [1]:

$$i(x,t) = f_1(x - wt) + f_2(x + wt) \quad (2.32)$$

$$v(x,t) = Z f_1(x - wt) + Z f_2(x + wt) \quad (2.33)$$

Com $f_1(x - wt)$ e $f_2(x + wt)$ sendo funções arbitrárias de $(x - wt)$ e $(x + wt)$ respectivamente. A função $f_1(x - wt)$ representa a onda se propagando na direção km , enquanto que $f_2(x + wt)$, na direção mk , ambas com velocidade w . Conforme visto anteriormente, na equação (2.10), pode-se definir a impedância característica como sendo:

$$Z_c = \sqrt{\frac{L'}{C'}} \quad (2.34)$$

E a velocidade de propagação fica assim:

$$\varpi = \frac{1}{\sqrt{L'C}} \quad (2.35)$$

Multiplicando a equação (2.32) por Z_c , somando à equação (2.33), e também subtraindo à mesma, tem-se:

$$v(x,t) + Z_c i(x,t) = 2Z_c f_1(x - \varpi t) \quad (2.36)$$

$$v(x,t) - Z_c i(x,t) = -2Z_c f_1(x + \varpi t) \quad (2.37)$$

Quando $(x - \varpi t)$ é constante, $v(x,t) + Z_c i(x,t)$ também é constante, e como d é o comprimento da linha, o tempo de propagação também será constante e igual à:

$$\tau = \frac{d}{\varpi} = d\sqrt{L'C} \quad (2.38)$$

Por isso,

$$v_k(t - \tau) + Z_c i_{km}(t - \tau) = v_m(t) + Z_c (-i_{mk}(t)) \quad (2.39)$$

Organizando os termos da equação acima, vem:

$$i_{mk}(t) = \frac{1}{Z_c} v_m(t) + I_m(t - \tau) \quad (2.40)$$

Onde a corrente histórica é dada por:

$$I_m(t - \tau) = -\frac{1}{Z_c} v_k(t - \tau) - i_{km}(t - \tau) \quad (2.41)$$

De forma análoga:

$$i_{km}(t) = \frac{1}{Z_c} v_k(t) + I_k(t - \tau) \quad (2.42)$$

Onde a corrente histórica é dada por:

$$I_k(t - \tau) = -\frac{1}{Z_c} v_m(t - \tau) - i_{mk}(t - \tau) \quad (2.43)$$

Assim sendo, pode-se definir uma representação para a linha de transmissão sem perdas e com parâmetros distribuídos, conforme a figura 2.15 abaixo:

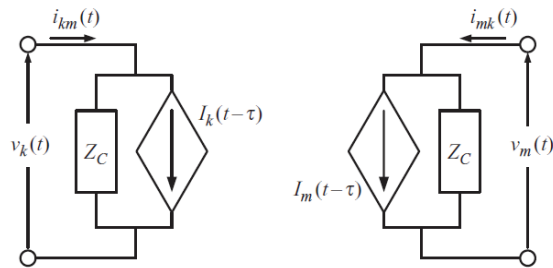


Figura 2.15 – Circuito equivalente de 2 portas para a linha sem perdas

Analisando a figura, observam-se que não há conexão direta entre os dois terminais, e as condições em uma extremidade são vistas de forma indireta e com atraso de tempo (tempo de propagação) pela outra extremidade através das fontes históricas.

Capítulo 3. O Método de Solução Nodal Modificado

3.1. Origem – Método Nodal

O método nodal [3] consiste em uma formulação matricial capaz de representar uma rede elétrica em estudo. Por exemplo, um sistema elétrico com n barras pode ser representado por um conjunto de equações do tipo $A\underline{x} = \underline{b}$, onde a matriz A é a matriz de impedância nodal, e os vetores \underline{x} e \underline{b} são os vetores de injeções de corrente e de tensões nodais respectivamente, ou seja, $Z_N \underline{I} = \underline{V}$. Invertendo-se a matriz Z_N é obtida outra matriz Y_N , chamada de matriz de admitância nodal, assim esse conjunto de equações ficará $Y_N \underline{V} = \underline{I}$.

3.1.1. Montagem da Matriz Nodal

Geralmente utiliza-se o método de solução nodal através da matriz Y_N , principalmente pela facilidade de entendimento e montagem da mesma.

Considere o sistema elétrico de 3 (três) barras da figura abaixo:

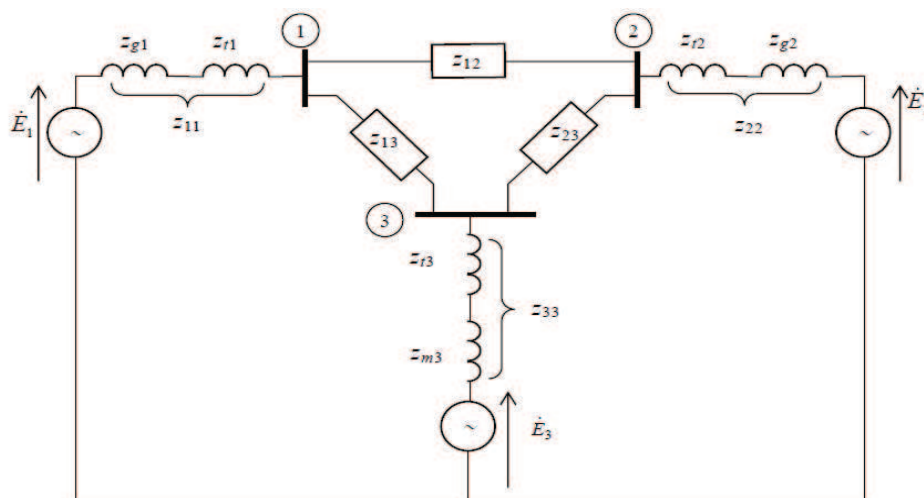


Figura 3.1 – Sistema elétrico com 3 barras

Transformando as fontes de tensão em fontes de corrente e as impedâncias em admitâncias, pode-se representar o sistema como mostra a figura 3.2:

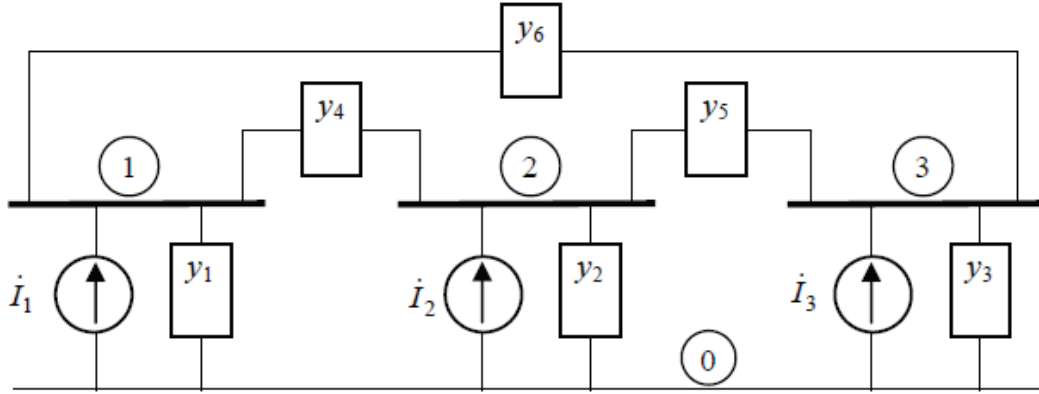


Figura 3.2 – Sistema elétrico com fontes de corrente e admitâncias

Agora, pode-se escrever um conjunto de equações nodais para cada barra do sistema, ou seja:

$$\text{Barra 1: } \dot{I}_1 = \dot{V}_1(y_1 + y_4 + y_6) - \dot{V}_2 y_4 - \dot{V}_3 y_6$$

$$\text{Barra 2: } \dot{I}_2 = -\dot{V}_1 y_4 + \dot{V}_2(y_2 + y_4 + y_5) - \dot{V}_3 y_5$$

$$\text{Barra 3: } \dot{I}_3 = -\dot{V}_1 y_6 - \dot{V}_2 y_5 + \dot{V}_3(y_3 + y_5 + y_6) \quad (3.1)$$

Que na forma matricial fica assim:

$$\begin{pmatrix} \dot{I}_1 \\ \dot{I}_2 \\ \dot{I}_3 \end{pmatrix} = \begin{pmatrix} y_1 + y_4 + y_6 & -y_4 & -y_6 \\ -y_4 & y_2 + y_4 + y_5 & -y_5 \\ -y_6 & -y_5 & y_3 + y_5 + y_6 \end{pmatrix} \begin{pmatrix} \dot{V}_1 \\ \dot{V}_2 \\ \dot{V}_3 \end{pmatrix} \quad (3.2)$$

Nota-se que a matriz Y_N é quadrada e de ordem igual ao número de barras do sistema. Pela simples inspeção da rede em estudo, é possível montar a matriz Y_N da seguinte forma:

- Os elementos que estão na diagonal principal, Y_{kk} , são a soma das admitâncias ligadas diretamente na barra k;

- Os elementos que estão fora da diagonal principal, Y_{km} , são o negativo da soma das admitâncias que conectam a barra k à barra m.

Como Z_N geralmente é uma matriz cheia, se torna mais fácil resolver o problema utilizando a matriz Y_N , que é uma matriz esparsa, o que demandará um esforço computacional menor.

3.2. Nodal Modificado

O método nodal modificado [2] é simplesmente uma expansão do método nodal citado acima. Agora, são acrescentadas as linhas da matriz correspondentes às barras que possuem fontes de tensão e chaves. Por exemplo, na figura 3.3 [2] a fonte de tensão está conectada na barra 6, as chaves estão entre as barras 5 e 3, e também 4 e 3:

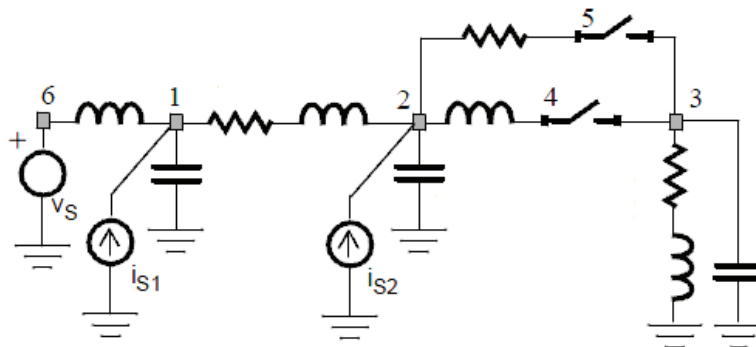


Figura 3.3 – Sistema elétrico com 6 barras [2]

3.2.1. Montagem da Matriz Nodal Modificada

Conforme mostrado anteriormente, a montagem da matriz de Y_N procede-se da mesma forma. Como o sistema possui 6 barras, a matriz Y_N será 6×6 , logo

$$Y_N \underline{V} = \underline{I} \quad (3.3)$$

onde:

$$Y_N = \begin{pmatrix} y_1 + y_{16} + y_{12} & -y_{12} & 0 & 0 & 0 & -y_{16} \\ -y_{12} & y_2 + y_{12} + y_{24} + y_{25} & 0 & -y_{24} & -y_{25} & 0 \\ 0 & 0 & y_3 & 0 & 0 & 0 \\ 0 & -y_{24} & 0 & y_{24} & 0 & 0 \\ 0 & -y_{25} & 0 & 0 & y_{25} & 0 \\ -y_{16} & 0 & 0 & 0 & 0 & y_{16} \end{pmatrix} \quad (3.3a)$$

Como o sistema possui 1 fonte de tensão e 2 chaves, a formulação matricial possuirá 3 (1+2) equações acrescidas à formulação original $Y_N \underline{V} = \underline{I}$. Assim, o novo sistema de equações ficará como na equação (3.4):

$$\begin{pmatrix} Y_N & V_a^t & S_a^t \\ V_a & 0_{V_S} & 0_{V_S S} \\ S_a & 0_{V_S S}^t & S_0 \end{pmatrix} \begin{pmatrix} \underline{V}_N \\ \underline{I}_{V_S} \\ \underline{I}_S \end{pmatrix} = \begin{pmatrix} \underline{I}_N \\ \underline{V}_S \\ \underline{0} \end{pmatrix} \quad (3.4)$$

Onde:

V_a = Matriz de incidência nodal de fontes de tensão, sua ordem é $nV_S \times n$;

S_a = Matriz de incidência nodal de chaves fechadas, sua ordem é $n_S \times n$;

0_{V_S} = Matriz nula $nV_S \times nV_S$;

$0_{V_S S}$ = Matriz nula $nV_S \times n_S$;

S_0 = Matriz esparsa usada para anular as correntes nas chaves abertas, sua ordem é $n_S \times n_S$;

$\underline{\dot{V}}_N$ = Vetor das tensões nodais desconhecidas;

$\underline{\dot{I}}_{V_S}$ = Vetor das correntes desconhecidas das fontes de tensão;

$\underline{\dot{I}}_S$ = Vetor das correntes desconhecidas das chaves;

$\underline{\dot{I}}_N$ = Vetor das injeções das fontes de corrente conhecidas; e

\underline{V}_S = Vetor das fontes de tensão conhecidas.

Com

nV_S = Número de fontes de tensão; e

n_S = Número de chaves.

A matriz Y_{NM} pode ser chamada de matriz de admitância nodal modificada e será quadrada de ordem $n + nV_S + n_S = 9$. Como só há uma fonte de tensão, $nV_S = 1$, então a sétima linha da matriz Y_{NM} , correspondente à fonte de tensão (matriz V_a), receberá valor 1 somente na 6ª coluna, porque a mesma está localizada na barra 6, assim a matriz V_a ficará como está representado na equação (3.5). Caso houvesse outra fonte de tensão conectada a uma barra k , outra linha seria acrescida à matriz V_a , e o mesmo procedimento seria feito, só que o valor 1 apareceria somente na coluna correspondente à barra k .

$$V_a = (0 \ 0 \ 0 \ 0 \ 0 \ 1) \quad (3.5)$$

Já para as chaves, como só há duas, $n_S = 2$. Então as variáveis correspondentes obedecerão a seguintes equações para as chaves fechadas:

$$\text{Tensões: } V_3 - V_4 = 0 \text{ e } V_3 - V_5 = 0;$$

$$\text{Correntes: } I_{34}, I_{35} \neq 0.$$

Logo, S_a e S_0 ficarão da seguinte forma:

$$S_a = \begin{pmatrix} 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 \end{pmatrix} \quad (3.6)$$

$$S_0 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \quad (3.7)$$

E quando as chaves estiverem abertas, segue:

Tensões: $V_3, V_4 \neq 0$;

Correntes: $I_{34} = 0$ e $I_{35} = 0$.

Portanto

$$S_a = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (3.8)$$

$$S_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad (3.9)$$

Finalmente, pode-se escrever a forma final da matriz de admitância nodal modificada. Quando as estiverem chaves fechadas, a mesma ficará assim:

$$Y_{NM} = \begin{pmatrix} y_1 + y_{16} + y_{12} & -y_{12} & 0 & 0 & 0 & -y_{16} & 0 & 0 & 0 \\ -y_{12} & y_2 + y_{12} + y_{24} + y_{25} & 0 & -y_{24} & -y_{25} & 0 & 0 & 0 & 0 \\ 0 & 0 & y_3 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & -y_{24} & 0 & y_{24} & 0 & 0 & 0 & -1 & 0 \\ 0 & -y_{25} & 0 & 0 & y_{25} & 0 & 0 & 0 & -1 \\ -y_{16} & 0 & 0 & 0 & 0 & y_{16} & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

E sabendo-se que

$$\begin{pmatrix} \underline{V_N} \\ \underline{I_{V_S}} \\ \underline{I_S} \end{pmatrix} = \begin{pmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \\ I_6 \\ I_{34} \\ I_{35} \end{pmatrix} \text{ e } \begin{pmatrix} \underline{I_N} \\ \underline{V_S} \\ \underline{0} \end{pmatrix} = \begin{pmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \\ I_6 \\ V_6 \\ 0 \\ 0 \end{pmatrix}$$

Com base da modelagem elétrica dos elementos de circuito apresentada no capítulo anterior, os elementos que compõem a matriz Y_{NM} serão o inverso das resistências efetivas encontradas para cada uma das configurações descritas. No caso da linha de transmissão, basta colocar o inverso da impedância característica da barra correspondente.

Uma das formas utilizadas para resolver este problema é invertendo-se a matriz Y_{NM} e multiplicando pelo vetor de variáveis conhecidas, para se obter o vetor de variáveis desconhecidas. Também podem ser usados os métodos de fatoração matricial encontrados na literatura técnica apropriada.

Capítulo 4. A Modelagem Orientada a Objetos Aplicada aos Elementos de Circuito

4.1. Introdução

O objetivo deste capítulo é ilustrar a metodologia da programação utilizada no Matlab. Para isso, será introduzido o conceito de objeto e da modelagem orientada a objetos. Também será apresentada a composição das classes de elementos de sistema associada à modelagem dos elementos de circuito, bem como os respectivos diagramas de classes.

4.2. O Conceito de Objeto

O conceito de objeto [4][5] está relacionado a um conjunto de dados (atributos) que o descrevem, e a um conjunto de funções (métodos) capazes de executar as tarefas que lhe são pertinentes. Cada objeto possui identidade única. Considere uma aplicação em geometria plana como na figura 4.1 [6]:

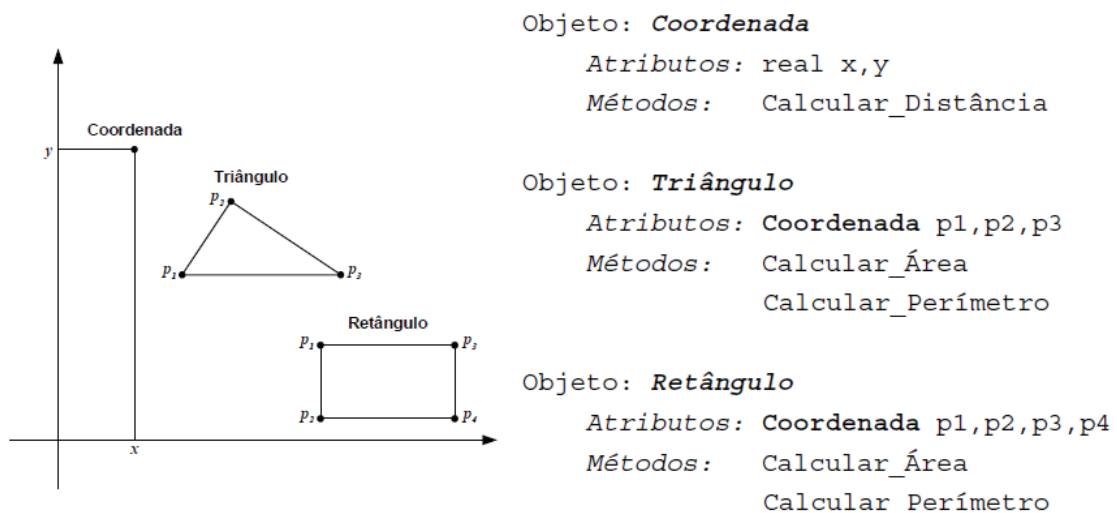


Figura 4.1 – Objetos e seus atributos e métodos [6]

Percebe-se que os atributos e métodos são particulares a cada objeto. Por exemplo, para a obtenção de uma coordenada, são necessários dois pontos, x e y , e com duas coordenadas é possível calcular a distância entre as mesmas. O mesmo raciocínio pode ser estendido para os casos do triângulo e retângulo, que precisam de três e quatro coordenadas respectivamente, para serem delimitados.

Quando se tem um tipo específico de objeto, a declaração da estrutura computacional que o descreve chama-se *classe*, ou seja, objetos que possuem a mesma estrutura de dados e mesmo comportamento pertencem à mesma classe. Então cada classe possui uma diversa gama de objetos, que são instâncias desta mesma classe.

4.3. Herança, Polimorfismo e Encapsulamento

Toda classe possui três propriedades chamadas de *herança*, *polimorfismo* e *encapsulamento* [5]. A seguir, são definidas cada uma dessas propriedades.

- Herança: É uma propriedade que possibilita ao objeto adquirir as propriedades de outro objeto. Observa-se que os objetos triângulo e retângulo (filhos) herdam todos os atributos do objeto coordenada (pai). Por exemplo, para formar uma coordenada, são necessários dois atributos reais x e y , que já foram declarados no objeto pai.
- Polimorfismo: Da mesma forma, se pode destacar o fato de que os dois métodos (*Calcular_Área* e *Calcular_Perímetro*) pertencentes a estes dois objetos, triângulo e retângulo, recebem o mesmo nome, porém a execução se dá de forma diferente. Isto facilita a legibilidade do código, caracterizando-se em outra propriedade conhecida como *polimorfismo*.
- Encapsulamento: Apesar de não estar explícita no exemplo em questão, essa propriedade pode ser definida como um mecanismo que une código e dados com a finalidade de construir um objeto, e manter seus atributos e métodos a salvo de interferências externas. Existem 3 (três) formas de encapsular um objeto, e para isso basta declarar no código se um determinado atributo ou

método é *público*, *protegido* ou *privado*. O primeiro determina que qualquer código poderá acessar ou modificar as informações ali contidas. O segundo permite o acesso e alteração das informações do objeto, somente à classe e seus filhos. E por fim, o último só garante acesso ao objeto pai.

4.4. Representação das Classes de Objetos

Uma das formas de se representar uma classe de objetos é utilizando a notação gráfica definida pela UML [5] (Unified Modeling Language), que estabelece uma representação padronizada para as classes, e esta consiste de um retângulo subdividido em três partes: nome da classe, lista dos atributos e lista dos métodos. A figura 4.2 [6] mostra a representação de uma classe através da UML:

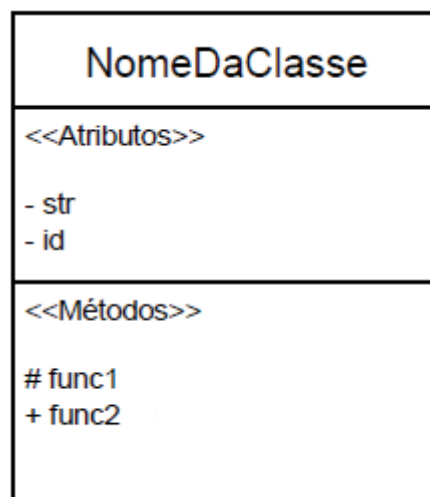


Figura 4.2 – Representação de uma classe através da UML [6]

É dentro do espaço reservado aos métodos, que são definidos os construtores da classe, a saber, as funções de inicialização da mesma. Os construtores são invocados no momento em que os objetos são criados, e possuem o mesmo nome da classe.

Agora, podem-se representar as classes de elementos de circuito utilizadas no desenvolvimento do simulador de acordo com a UML. Isto será detalhado na próxima seção.

4.5. Classes de Elementos de Circuito

Cada um dos elementos de circuito mostrados no capítulo 2 precisa ser modelado à luz do enfoque da MOO, de acordo com seus atributos e métodos. A presente seção trata da formulação das classes de elementos de circuito utilizadas no desenvolvimento dos códigos do simulador. Os referidos elementos (objetos) foram subdivididos em quatro classes principais:

- Classe *Source* – Fontes de tensão, fontes de corrente e fontes de corrente histórica;
- Classe *PassiveElement* – Elementos passivos: resistores, capacitores e indutores;
- Classe *Switch* – Chaves;
- Classe *TransLine* – Linhas de transmissão.

A seguir, será apresentada uma descrição das classes supracitadas:

4.5.1. Classe *Source*

A classe *Source* engloba todas as fontes do sistema, incluindo as fontes de corrente históricas dos elementos passivos. Esta classe também possui seus atributos e métodos, conforme a figura 4.3.

Observa-se que a classe *Source* obedece à modelagem elétrica das fontes de tensão ou corrente do capítulo 2. O construtor recebe as informações da fonte (atributos) e devolve uma função dependente do tempo (atributo *injection_function*) conforme as equações (2.28) e (2.29).

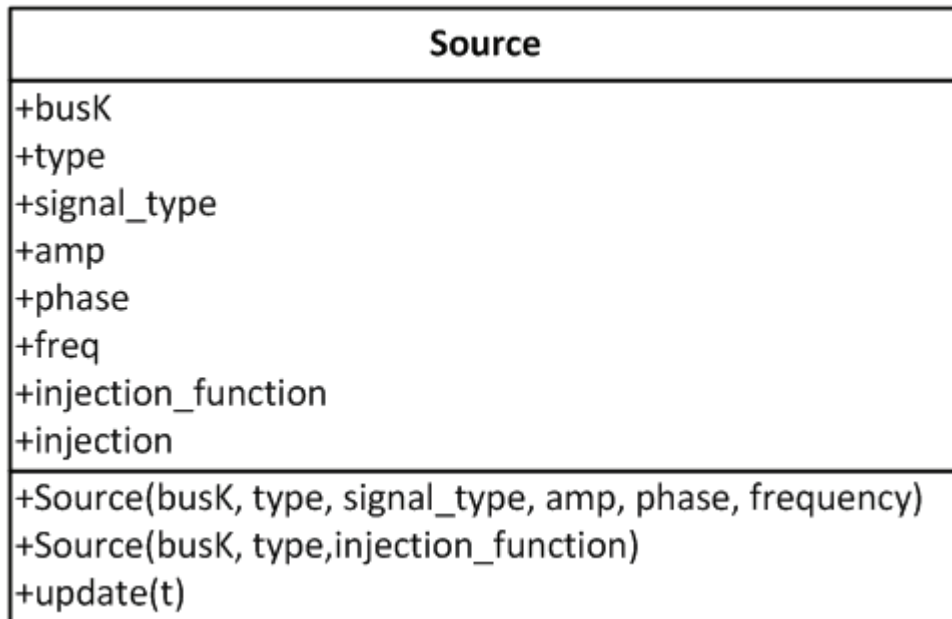


Figura 4.3 – Classe *Source*

O atributo *signal_type* é responsável pela informação do tipo de sinal que está sendo gerado pela fonte: senoidal, degrau ou rampa (classe *RampSource*), enquanto que o atributo *type* é utilizado para informar se é fonte de corrente ou de tensão. O método *update(t)* recebe o instante de tempo *t* atual do sistema, e devolve o valor do sinal da fonte no instante de tempo considerado (atributo *injection*).

4.5.2. Classe *RampSource*

A classe *RampSource* é semelhante a classe anterior, e somente é utilizada nos primeiros instantes de tempo da simulação para garantir a convergência. Esta classe é filha da classe *Source*, e herda todos os atributos da mesma, com exceção do *riseTime* (tempo que a rampa demora para atingir sua amplitude máxima) e *riseFactor* (razão entre a máxima amplitude (*maxAmp*) e o *riseTime*). A figura 4.4 revela mais detalhes sobre esta classe.

O método *update(t)* possui a mesma funcionalidade do método de mesmo nome da classe *Source*.

RampSource
+riseTime +riseFactor
+RampSource(busK, type, signal_type, maxAmp, phase, frequency, riseTime) +update(t)

Figura 4.4 – Classe *RampSource*

4.5.3. Classe *PassiveElement*

Os objetos pertencentes à classe *PassiveElement* são os elementos passivos como: resistor, indutor e capacitor. Essa classe também modela os objetos RL série e RLC série, sendo o resistor e o indutor, um caso particular do RL série, enquanto que o capacitor é um caso particular do RLC série. A figura 4.5 ilustra a referida classe.

PassiveElement
+busM +R +L +C +Gseries +ikm +Vc +prev_ikm
+PassiveElement(busK, busM, R, L, C) +update_injection(prev_vbusK, prev_vbusM) +update_ikm(vbusK, vbusM)

Figura 4.5 – Classe *PassiveElement*

Como se pode ver na figura, o atributo *busK* não está listado. Isso se deve ao fato de *Source* ser a classe pai de *PassiveElement*, que herdou esse atributo de *Source*. O atributo *Gseries* é a condutância calculada para cada objeto (RL série, RLC série, etc) conforme no capítulo 2, e é processado quando da construção do objeto através do construtor, que recebe as informações *R*, *L* e *C*, calcula *Gseries* e o associa às barras *k* e *m* do sistema em questão. O construtor também cria uma

função dependente do tempo, que corresponde à corrente histórica dos ramos preenchidos pelos elementos passivos. Essa função também é herdada da classe *Source* através do atributo *injection_function*. Os atributos *Vc* e *prev_ikm* são respectivamente a tensão no capacitor e a corrente no ramo km em que o elemento passivo está conectado.

Os demais métodos, *Atualiza_hist* e *Atualiza_ikm*, servem para atualizar a corrente histórica e a corrente que entra no ramo *km*, respectivamente. Para maiores detalhes, ver as equações do capítulo 2.

4.5.4. Classe *Switch*

Nesta classe são instanciados os objetos que representam as chaves. Seus atributos e métodos estão explicitados na figura abaixo.

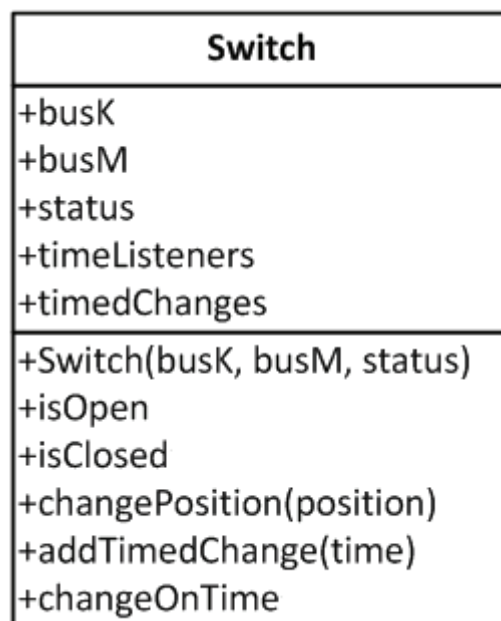


Figura 4.6 – Classe *Switch*

O método *timeListeners* armazena os eventos de troca de posição das chaves do sistema, determinados pelo atributo *timedChanges*.

O construtor dessa classe necessita das informações dos atributos *busK*, *busM* e *Status*, sendo este último, responsável pelo estado atual da chave: aberta ou fechada. Os métodos *isOpen* e *isClosed* servem para testar se o estado atual da chave é aberto ou fechado respectivamente, enquanto que *changePosition* recebe o

argumento *position* para trocar o estado da chave para o estado definido por *position*. O método *addTimedChange* adiciona um instante de tempo em que uma determinada chave trocará o seu estado, ao atributo *timedChanges* e também acrescenta um evento ao *timeListener*. E por fim, o método *changeOnTime* verifica se existem mais manobras a serem executadas por alguma chave e atualiza o atributo *timedChanges*, eliminando os instantes de tempo referentes a trocas de posição passadas.

4.5.5. Classe *TransLine*

A classe *TransLine* é utilizada para modelagem dos objetos referentes as linhas de transmissão, e basicamente, possui a mesma estrutura da classe *PassiveElement*, a menos do construtor e do *ringbuffer* [1]. O primeiro recebe as informações das barras que conectam a linha de transmissão em questão, bem como a impedância característica (atributo *Zc*) e a constante de propagação (atributo *tau*). Já o segundo, *ringbuffer*, é usado para armazenar as informações das injeções de correntes históricas (*prev_Ik* e *prev_Im*), e injeções de corrente (*prev_ikm* e *prev_imk*) e de tensão (*prev_vk* e *prev_vm*) nas barras *k* e *m*, no último instante em que a onda eletromagnética propagou de uma barra a para a outra.

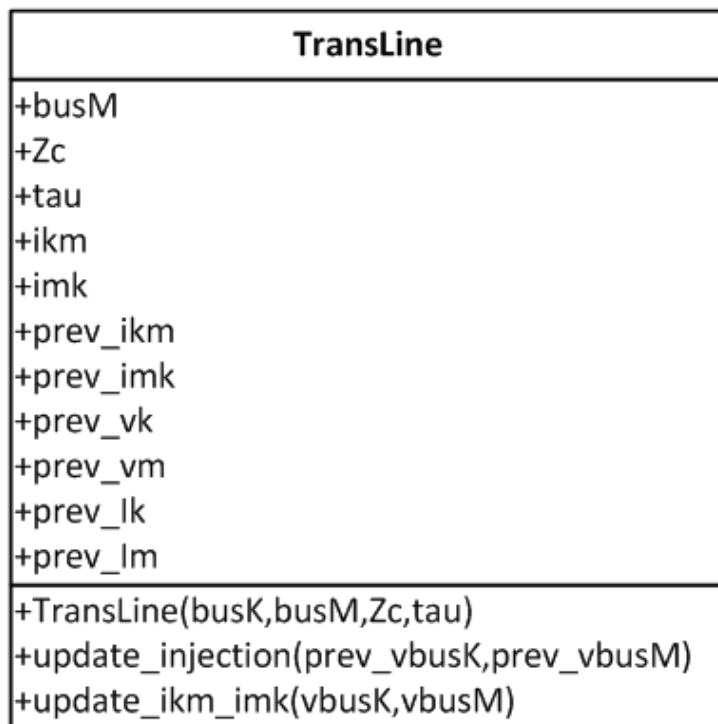


Figura 4.7 – Classe *TransLine*

A figura 4.7 ilustra a representação da classe *TranLine* de acordo com a UML. Os métodos que esta classe executa também são semelhantes aos métodos da classe *PassiveElement*: *update_injection* é utilizado para atualizar as injeções das fontes de corrente históricas da linha de transmissão, e o método *update_imk_ikm* serve para atualizar as injeções de corrente e de tensão nas barras que conectam a mesma.

4.5.6. Classe *PowerSystem*

A classe *PowerSystem* é responsável por tratar e armazenar os dados de entrada nos atributos identificados na figura abaixo.

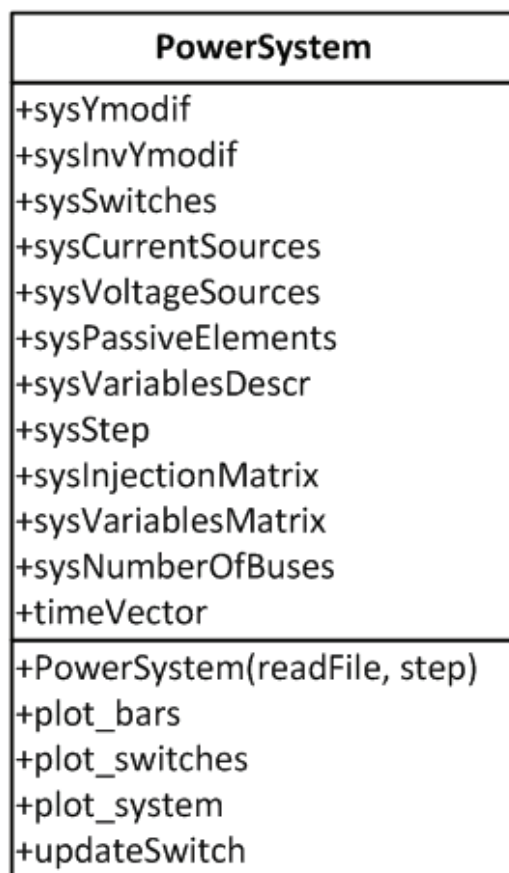


Figura 4.8 – Classe *PowerSystem*

A seguir, uma breve descrição dos atributos da classe *PowerSystem*:

- *sysYmodif*: Matrix de admitância nodal modificada;
- *sysInvYmodif*: Inversa da matriz de admitância nodal modificada;

- *sysSwitches*: Vetor contendo as chaves do sistema;
- *sysCurrentSources*: Vetor contendo as fontes de corrente do sistema;
- *sysVoltageSources*: Vetor contendo as fontes de tensão do sistema;
- *sysPassiveElements*: Vetor contendo os elementos passivos do sistema;
- *sysVariablesDescr*: Vetor contendo a descrição das variáveis de saída;
- *sysStep*: Passo de integração, $\text{step} (\Delta t)$;
- *sysInjectionMatrix*: Vetor contendo todas as injeções de tensão e corrente conhecidas;
- *sysVariablesMatrix*: Vetor contendo todas as variáveis das injeções de tensão e corrente nas barras;
- *sysNumberOfBuses*: Número total de barras do sistema;
- *timeVector*: Vetor contendo todos os instantes de tempo separados por Δt ;

Abaixo, segue a descrição dos métodos:

- *PowerSystem(readFile,step)*: Função que recebe como argumentos, o arquivo de entrada contendo os dados do sistema e o passo de tempo de simulação;
- *plot_bars*: Função que desenha os gráficos que contém somente as informações de tensão e corrente das barras sem chave do sistema;
- *plot_switches*: Função que desenha os gráficos que contém somente as informações de tensão e corrente das barras que possuem chave no sistema
- *plot_system*: Desenha os gráficos de todas as barras do sistema, incluindo as barras conectadas por chaves;
- *updateSwitch*: Função que atualiza o sistema quando ocorre uma mudança de estado em alguma chave.

Os dados de entrada dos sistemas a serem simulados estão contidos nos arquivos em *txt* (*readFile*) que se encontram nos anexos deste texto. Esses arquivos são lidos pela função *readPowerSystem*, então o construtor da classe *PowerSystem* prepara o sistema para ser revolvido pela função *run*. Terminada esta etapa, os resultados estarão prontos para serem mostrados em forma de gráfico pela função *plot_system*.

4.6. Diagrama de Classes

Um diagrama de classes é uma forma de representar as relações entre as classes de um determinado código. O caso em estudo possui basicamente duas formas de relacionamento entre classes: agregação (representado pelo losango em escuro) e herança (representado pelo triângulo). A primeira estabelece que um objeto de uma determinada classe, seja formado de vários outros objetos componentes de outras classes. Já a herança, foi definida anteriormente na secção 4.3. A figura abaixo ilustra o diagrama de classes utilizado na formulação dos códigos deste simulador.

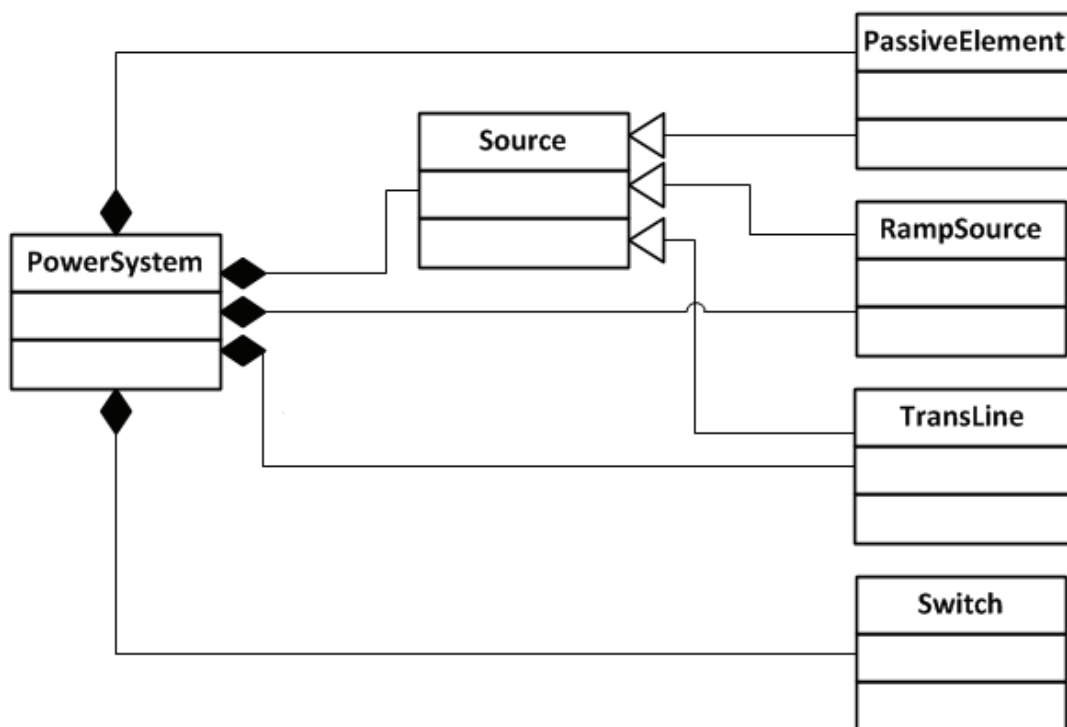


Figura 4.9 – Diagrama de Classes Simplificado

4.7. Fluxograma

Para a obtenção dos resultados presentes no capítulo seguinte, o código obedece a uma série de decisões que devem ser tomadas. A figura 4.10 mostra o fluxograma principal utilizado no simulador:

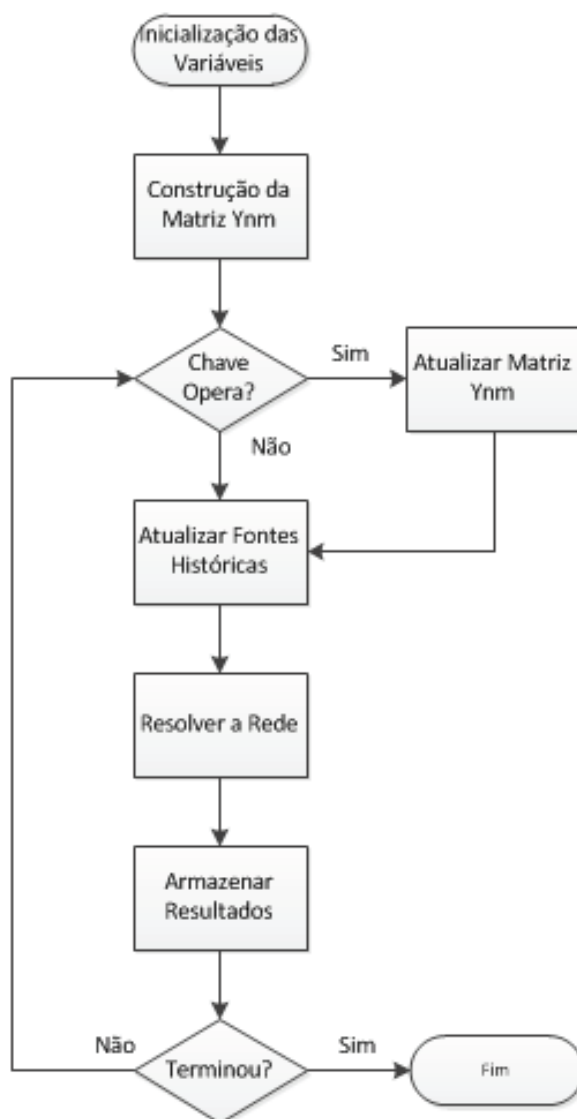


Figura 4.10 – Fluxograma

No próximo capítulo serão mostrados os resultados das simulações realizadas para vários casos diferentes.

Capítulo 5. Resultados e Conclusões

Nas próximas seções, serão apresentados os resultados das simulações realizadas para alguns casos. Adicionalmente, a título de comparação, os mesmos casos também foram simulados no Simulink do Matlab, e são estes:

- Circuito RL série;
- Circuito RLC série;
- Circuito com 3 barras;
- Circuito com 6 barras e 2 chaves;
- Circuito com 3 linhas de transmissão.

Para todos os casos, os scripts e arquivos de entrada necessários para as simulações no simulador desenvolvido no presente trabalho, estão nos anexos do mesmo.

5.1. Circuito RL Série

A figura 5.1 mostra uma associação série entre o resistor e o indutor.

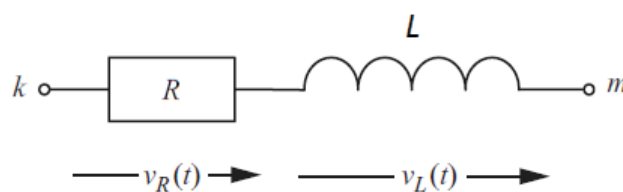


Figura 5.1 – Ramo RL série

Sabe-se da teoria de circuitos [7] que:

$$\tau = \frac{L}{R} \quad (5.1)$$

onde τ é a constante de tempo do circuito RL.

A corrente que percorre o ramo km é dada por:

$$i_{km}(t) = I_0 e^{-t/\tau} \quad (5.2)$$

cuja solução no tempo possui a seguinte forma:

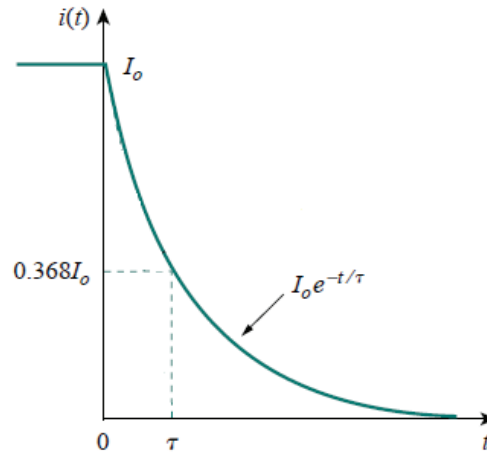


Figura 5.2 – Solução no tempo para a descarga do indutor em um circuito RL série [7]

Observa-se na figura 5.2, que quando o indutor está descarregando a energia armazenada, no caso do circuito RL, a forma de onda da corrente decai exponencialmente até o valor nulo. No instante de tempo equivalente a constante de tempo τ , o valor da corrente remanescente no indutor é cerca de 37 % da corrente total previamente armazenada no mesmo.

Um resultado semelhante é obtido quando da simulação do mesmo circuito no Simulink do Matlab. As figuras 5.3 e 5.4 mostram o diagrama de simulação e o gráfico resultante da mesma respectivamente, conforme a seguir:

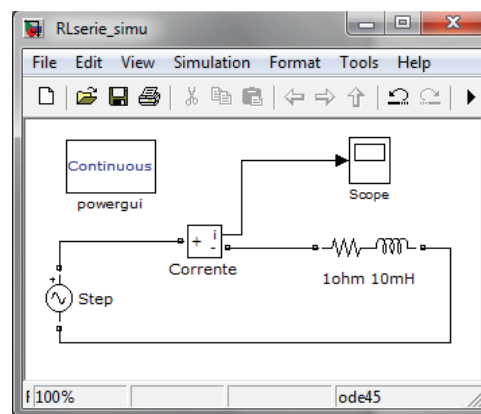


Figura 5.3 – Circuito RL simulado no Simulink do Matlab

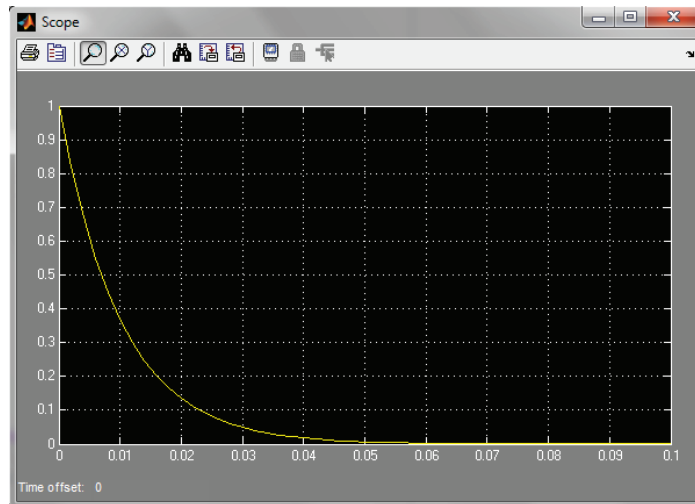


Figura 5.4 – Resultado da Simulação no Simulink

A tabela 5.1 mostra os dados utilizados no Simulink e no simulador desenvolvido neste trabalho. Os resultados obtidos neste último estão nas figuras 5.5 e 5.6 a seguir:

BarraK	BarraM	R (Ω)	L (mH)	τ (ms)	step (ms)
k	m	1	10	10	0.1

Tabela 5.1 – Dados do circuito RL série

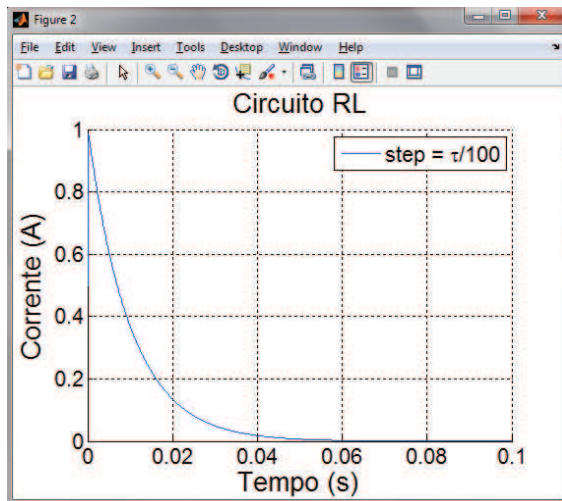


Figura 5.5 – Resposta do circuito RL série no simulador

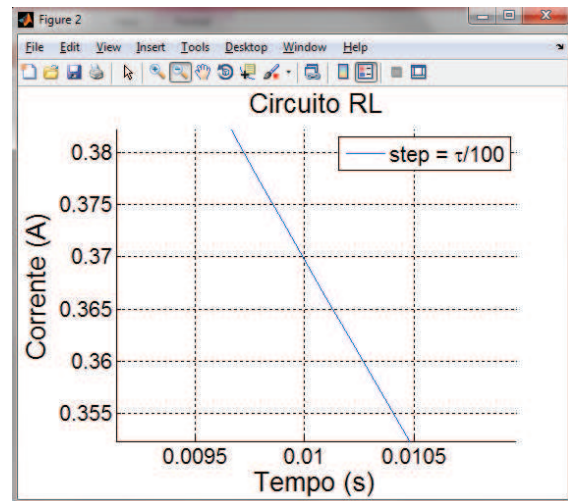


Figura 5.6 – Detalhe $t = \tau = 0.01\text{s}$ e $I \approx 0,368\text{A}$

Pela análise das figuras 5.2, 5.4, 5.5 e 5.6, constata-se que o simulador apresentou uma resposta para o circuito RL série semelhante à descrita na literatura técnica. E o mesmo resultado, quando comparado ao Simulink do Matlab, também se mostrou razoável e satisfatório.

5.2. Circuito RLC Série

A figura 5.7 mostra uma associação série entre o resistor, indutor e capacitor.

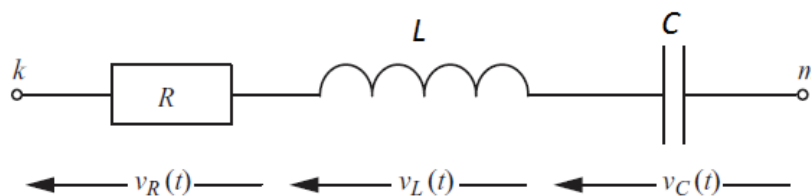


Figura 5.7 – Ramo RLC série

A equação diferencial que é obtida para associação RLC série é a seguinte [7]:

$$\frac{d^2i}{dt^2} + \frac{R}{L} \frac{di}{dt} + \frac{i}{LC} = 0 \quad (5.3)$$

na qual apresenta as seguintes soluções:

$$s_1 = -\alpha + \sqrt{\alpha^2 - \omega_0^2}, \quad s_2 = -\alpha - \sqrt{\alpha^2 - \omega_0^2} \quad (5.4)$$

onde:

- s_1 e s_2 → Frequências naturais de oscilação (Np/s – Nepers por segundo);
- $\alpha = \frac{R}{2L}$ → Coeficiente de amortecimento (Np/s);
- $\omega_0 = \frac{1}{\sqrt{LC}}$ → Frequência de ressonância (rad/s).

Um circuito RLC série possui três tipos de resposta diferentes, e são estes:

- Caso 1 → Superamortecido ($\alpha > \omega_0$) → Quando o circuito em questão está no modo superamortecido, a resposta no tempo da equação (5.3) é a equação (5.5):

$$i_{\text{km}}(t) = A_1 e^{s_1 t} + A_2 e^{s_2 t} \quad (5.5)$$

E a forma de onda que obedece a esta expressão no tempo é a da figura 5.8:

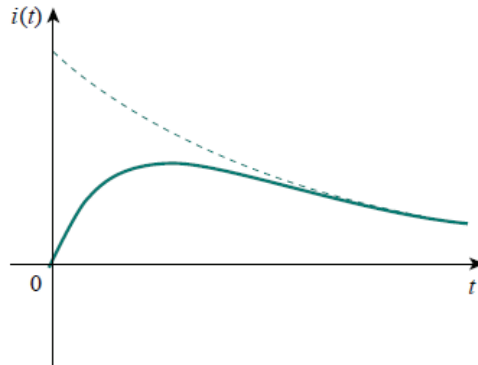


Figura 5.8 – Solução no tempo para o modo superamortecido em um circuito RLC série [7]

- Caso 2 → Criticamente amortecido ($\alpha = \omega_0$) → Quando o circuito em questão está no modo criticamente amortecido, $s_1 = s_2 = -\alpha = -\frac{R}{2L}$, e a resposta no tempo da equação (5.3) é a equação (5.6) abaixo:

$$i_{\text{km}}(t) = A_1 e^{-\alpha t} + A_2 t e^{-\alpha t} = A_3 t e^{-\alpha t} \quad (5.6)$$

E a forma de onda que obedece a esta última equação é a da figura 5.9:

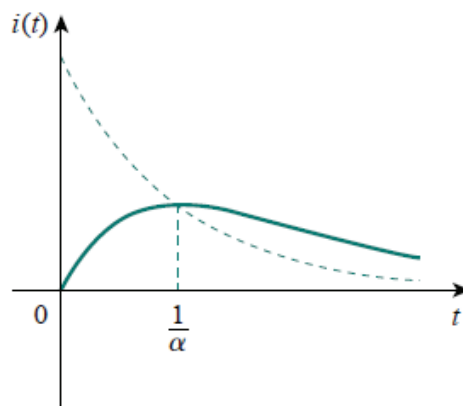


Figura 5.9 – Solução no tempo para o modo criticamente amortecido em um circuito RLC série [7]

- Caso 3 → Subamortecido ($\alpha < \omega_0$) → Quando o circuito em questão está no modo subamortecido, temos

$$s_1 = -\alpha + j\omega_d \text{ e } s_2 = -\alpha - j\omega_d \quad (5.7)$$

onde $j = \sqrt{-1}$, $\omega_d = \sqrt{\omega_0^2 - \alpha^2}$ e a resposta no tempo da equação (5.3) é a equação (5.8):

$$i_{\text{km}}(t) = e^{-\alpha t} [B_1 \cos(\omega_d t) + B_2 \sin(\omega_d t)] \quad (5.8)$$

E a forma de onda que obedece a expressão supracitada é a da figura 5.10:

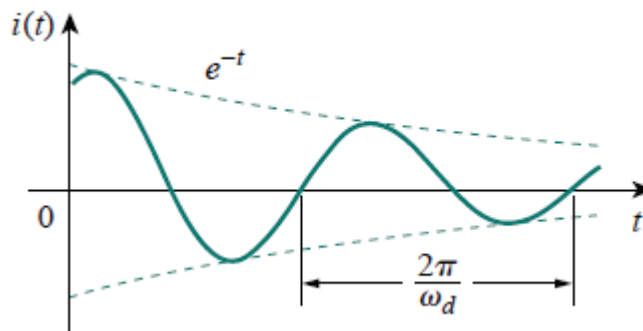


Figura 5.10 – Solução no tempo para o modo subamortecido em um circuito RLC série [7]

A seguir serão mostrados os resultados das simulações realizadas no Simulink do Matlab e no simulador produzido neste trabalho, para os três casos apresentados acima. Os dados utilizados em ambas as simulações estão na tabela 5.2. abaixo:

Resposta	R (Ω)	L (mH)	C (μF)	α (Np/s)	ω_0 (rad/s)	ω_d (Np/s)	step (μs)
Superamortecido	40	10	50	2000	1414	—	10
Criticamente Amortecido	28.28	10	50	1414	1414	—	10
Subamortecido	10	10	50	500	1414	1323	10

Tabela 5.2 – Dados do circuito RLC série

A figura 5.11 mostra a tela do Simulink utilizada para a obtenção das formas de onda encontradas na figura 5.12:

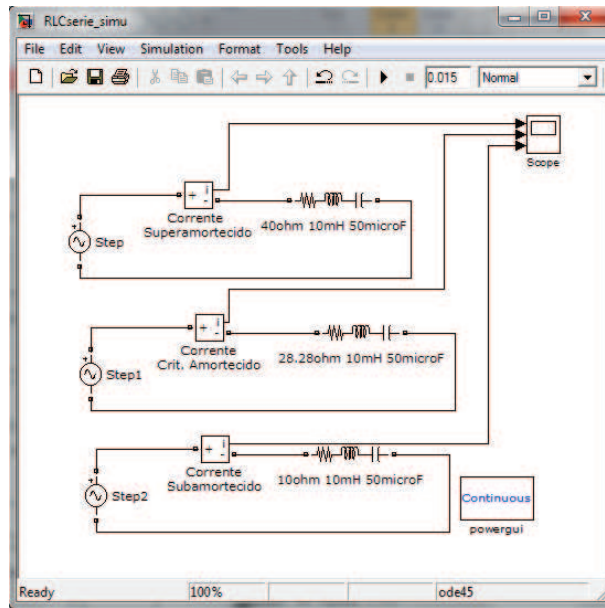


Figura 5.11 – Circuito RLC simulado no Simulink do Matlab

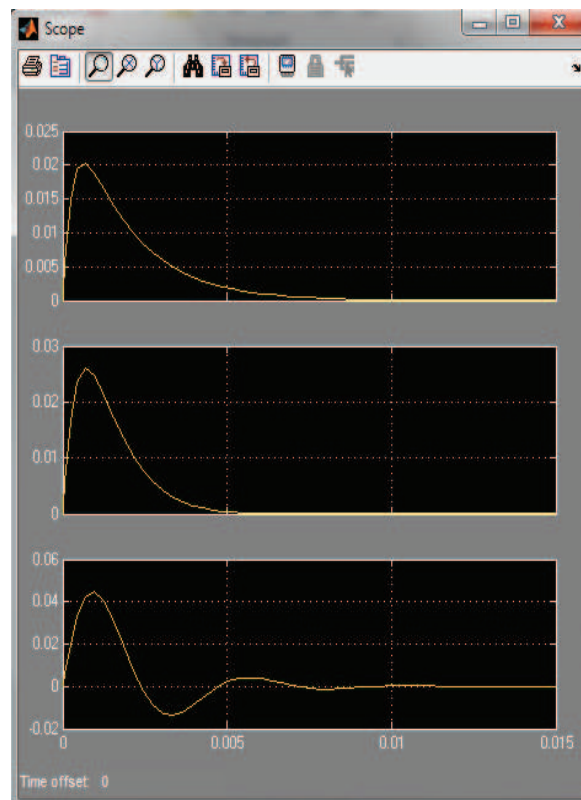


Figura 5.12 – Resultado da Simulação no Simulink (de cima para baixo: modo Superamortecido, modo criticamente amortecido e modo subamortecido)

As figuras 5.13 e 5.14 apresentam o resultado da simulação obtido através do programa desenvolvido:

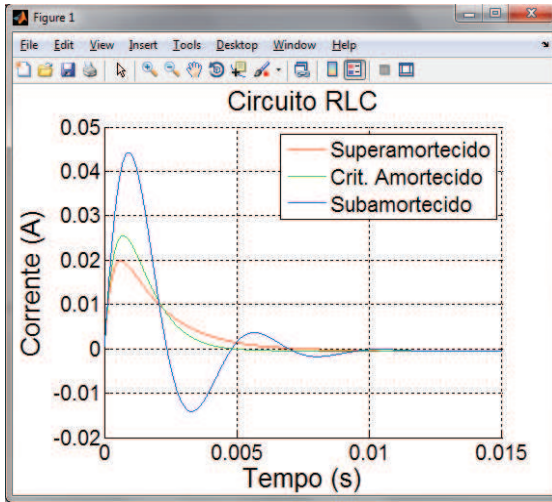


Figura 5.13 – Resposta do circuito RLC série no simulador

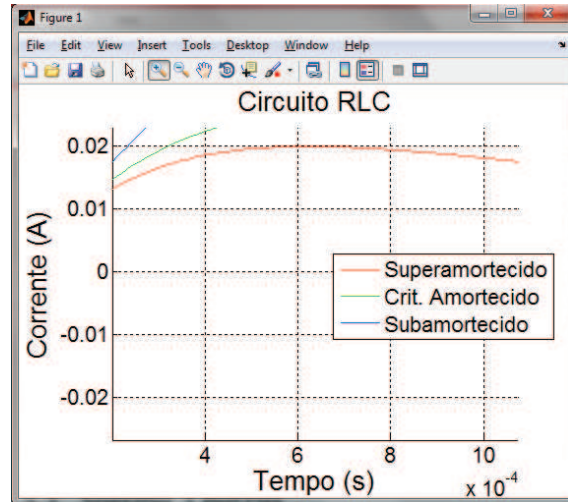


Figura 5.14 – Detalhe $1/\alpha = 0.0005$

Analisando-se os resultados obtidos, percebe-se que estão de acordo com a teoria apresentada, tanto no caso simulado pelo Simulink, quanto no simulado pelo simulador. Isto pode ser corroborado através das figura 5.13 e 5.14. Na primeira, para o modo subamortecido, a senóide amortecida pela exponencial apresenta período de oscilação igual à $2\pi/\omega_d \approx 0.047s$. Já na segunda, no modo criticamente amortecido, o instante de tempo para o qual a amplitude da resposta do circuito RLC é máxima, equivale ao inverso do coeficiente de amortecimento.

5.3. Circuito com 3 Barras

A figura 5.14 mostra o circuito com 3 barras simulado na presente seção:

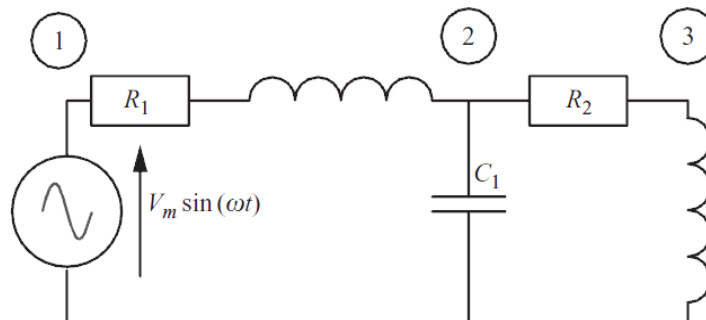


Figura 5.14 – Circuito com 3 barras

A tabela 5.3 apresenta os dados do circuito mostrado acima:

BarraK	BarraM	R (Ω)	XL (Ω)	XC (Ω)
1	2	0.05	0.75398	
2				3.31573
2	3	0.5		
3			2.26195	

Tabela 5.3 – Dados do sistema 3 barras

Utilizando estes dados e considerando um passo de integração (step) igual a $100\mu\text{s}$, foram realizadas simulações no Simulink do Matlab e no simulador. A amplitude da fonte de tensão senoidal é de 10 V como mostra a figura 5.15, que apresenta a tela do Simulink configurada na ocasião da simulação:

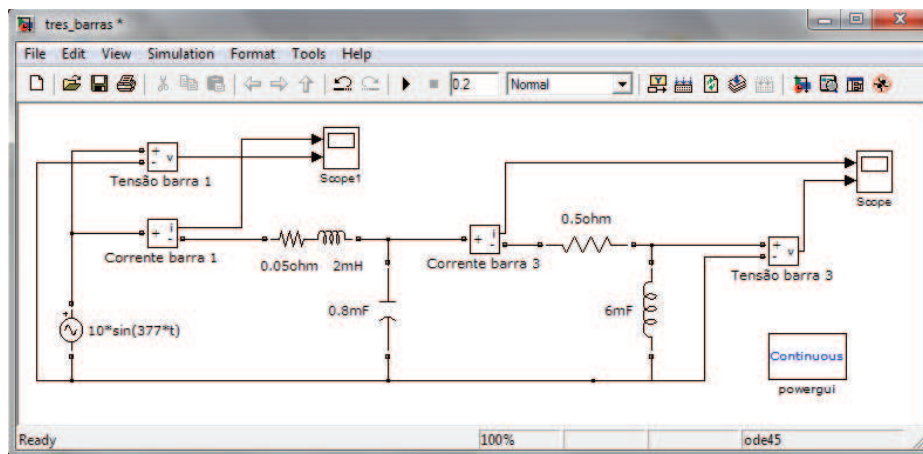


Figura 5.15 – Circuito com 3 barras simulado no Simulink do Matlab

As formas de onda das injeções de corrente e de tensão das barras 1 e 3 estão nas figuras 5.16 e 5.17, respectivamente:

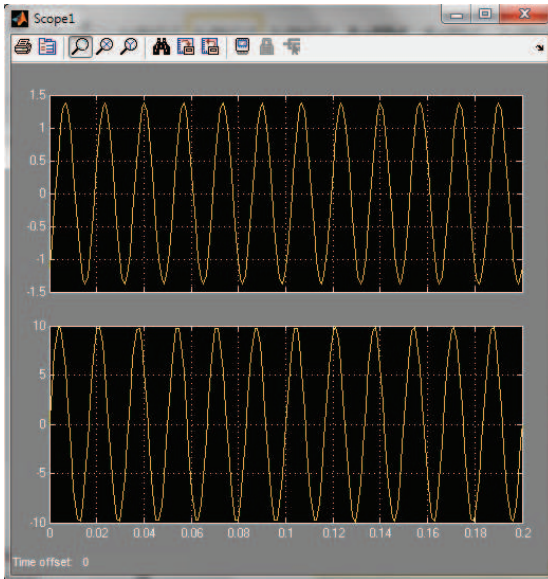


Figura 5.16 – De cima para baixo: formas de onda das injeções de corrente e de tensão na barra 1 no Simulink do Matlab

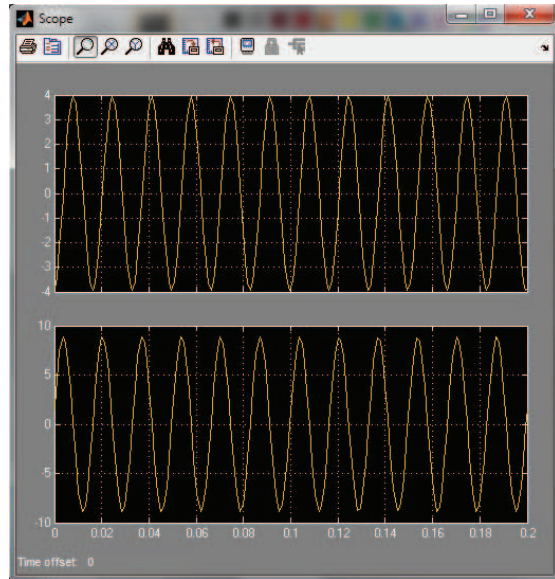


Figura 5.17 – De cima para baixo: formas de onda das injeções de corrente e de tensão na barra 3 no Simulink do Matlab

A seguir, serão mostrados os resultados obtidos no simulador para as barras 1 e 3 no circuito em estudo:

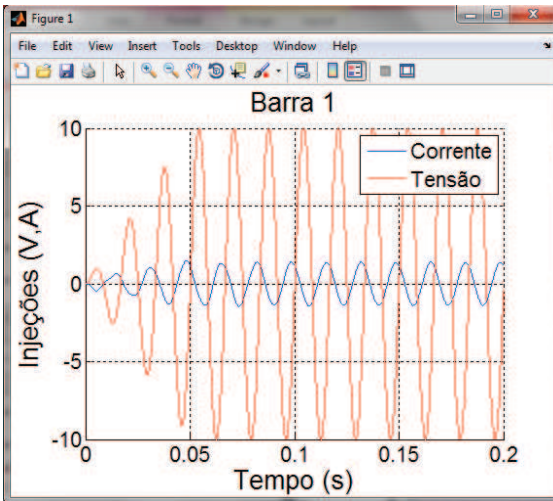


Figura 5.18 – Formas de onda das injeções de corrente e de tensão na barra 1 no simulador

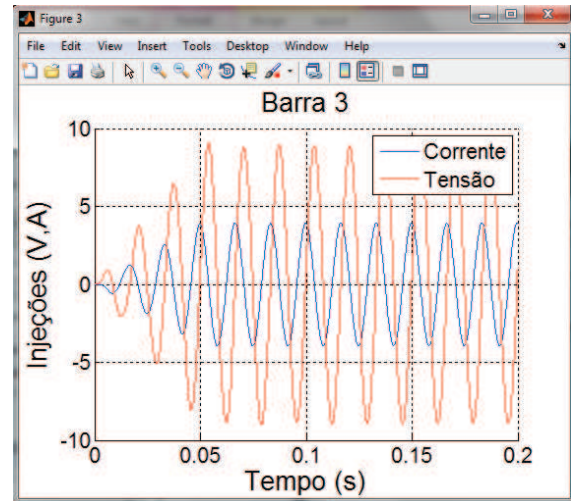


Figura 5.19 – Formas de onda das injeções de corrente e de tensão na barra 3 no simulador

Pela simples comparação entre as figuras 5.16 e 5.18, e 5.17 e 5.19, percebe-se a qualidade do desempenho do simulador, que se mostrou satisfatória para a simulação utilizando os modelos desenvolvidos para os elementos passivos.

As formas de onda das injeções de corrente e de tensão para a barra 2 do circuito em questão não foram mostradas, mas, no caso da corrente que desce pelo capacitor da referida barra, pode ser obtida facilmente pela diferença entre as correntes da barra 3 e da barra 1. Já a tensão na barra 2 é praticamente igual a da barra 3, pois o resistor que as conectam pode ser considerado desprezível.

5.4. Circuito com 6 Barras e 2 Chaves

A figura 5.20 apresenta um circuito com 6 barras e 2 chaves:

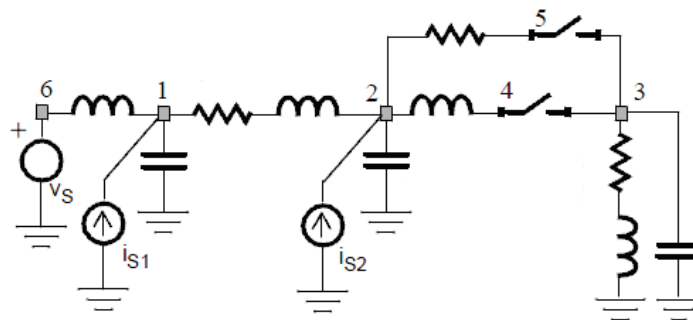


Figura 5.20 – Circuito com 6 barras e 2 chaves

Os dados utilizados nas simulações referentes aos elementos passivos e as fontes de tensão e de corrente estão nas tabelas 5.4 e 5.5 respectivamente:

BarraK	BarraM	R (Ω)	XL (Ω)	XC (Ω)
1	6		2.2619	
1	2	0.05	0.754	
1				3.3157
2	4		2.2619	
2	5	0.5		
2				3315.7
3		22.61	7.4343	
3				2652.6

Tabela 5.4 – Dados dos elementos passivos do circuito com 6 barras e 2 chaves

BarraK	Amp. (V)	Fase (deg)	Freq. (Hz)	Tipo
6	1.05	120	60	Tensão
1	1	0	60	Corrente
2	1	240	60	Corrente

Tabela 5.5 – Dados das fontes de tensão e de corrente do circuito com 6 barras e 2 chaves

Com o intuito de testar o funcionamento das chaves, foram simuladas duas situações diferentes tanto no Simulink como no simulador. Na primeira, as duas chaves trocam de estado alternadamente num mesmo instante de tempo. Já na segunda, as duas chaves mudam da posição aberta para fechada simultaneamente. A figura 5.21 apresenta a tela utilizada na simulação realizada no Simulink:

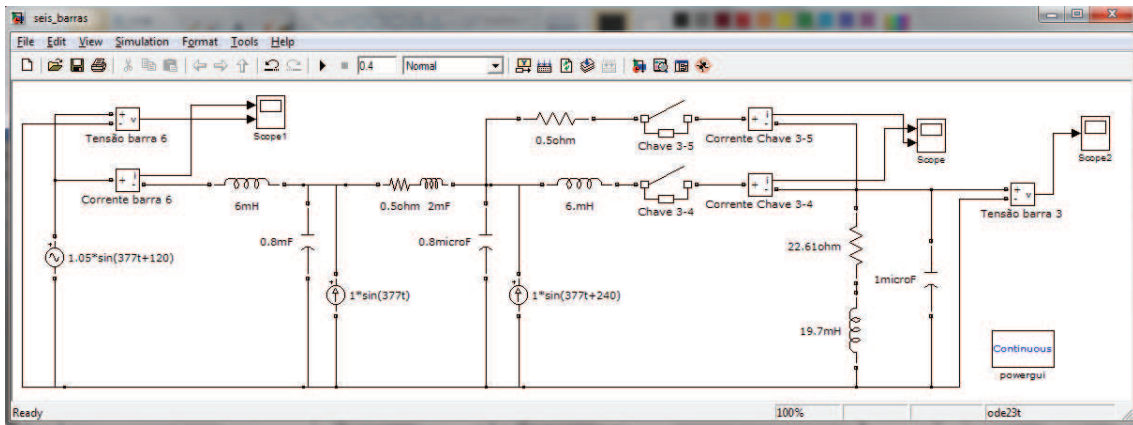


Figura 5.21 – Circuito com 6 Barras e 2 chaves simulado no Simulink do Matlab

Em todos os casos, são monitoradas as injeções de tensão e de corrente na barra 6, a injeção de tensão na barra 3 e as injeções de corrente nas chaves do circuito. O passo de integração é sempre igual a $100\mu\text{s}$ e o tempo total de simulação é de 0.4s. As duas situações são mostradas a seguir:

- 1º Caso → A tabela 5.6 resume a descrição dos eventos das chaves:

Chave	Posição Inicial	Posição Final	Instante de Troca
3-4	Fechada	Aberta	0.2s
3-5	Aberta	Fechada	0.2s

Tabela 5.6 – Descrição dos eventos das chaves para o 1º caso

As figuras 5.22 a 5.24 mostram os resultados das simulações realizadas no Simulink:

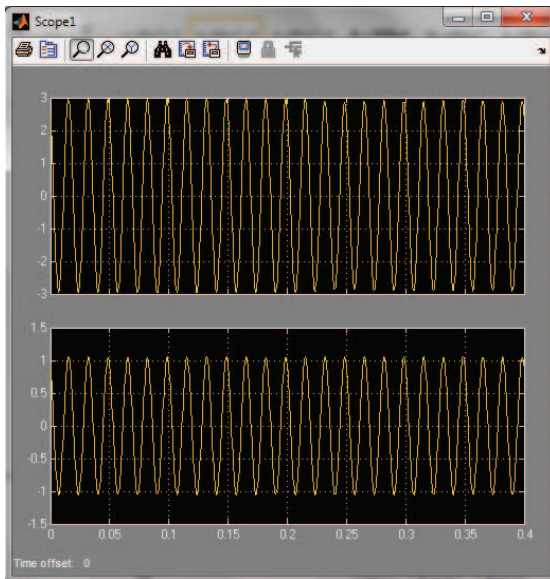


Figura 5.22 – De cima para baixo: formas de onda das injeções de corrente e de tensão na barra 6 no Simulink do Matlab

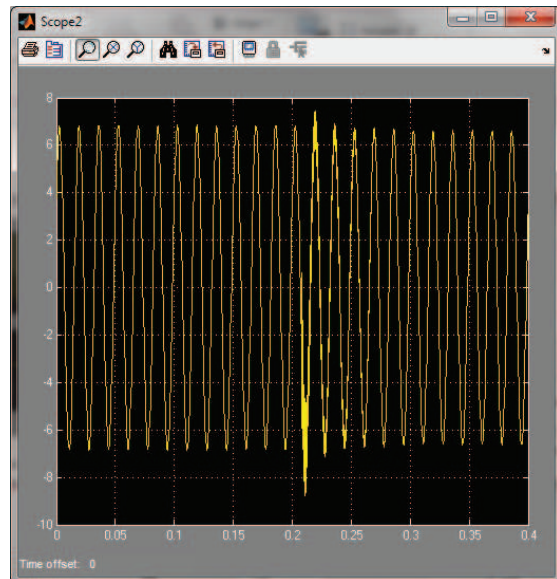


Figura 5.23 – Forma de onda da injeção de tensão na barra 3 no Simulink do Matlab

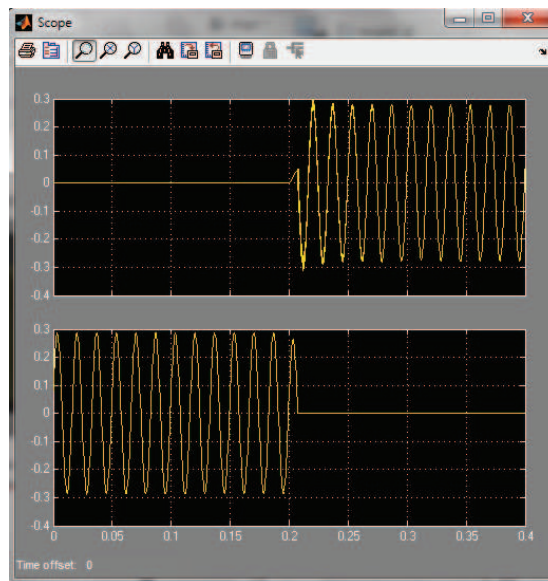


Figura 5.24 – De cima para baixo: formas de onda das injeções de corrente nas chaves 3-4 e 3-5 no Simulink do Matlab

Agora, serão apresentados os resultados obtidos pelo simulador. As figuras 5.25 a 5.28 ilustram o citado:

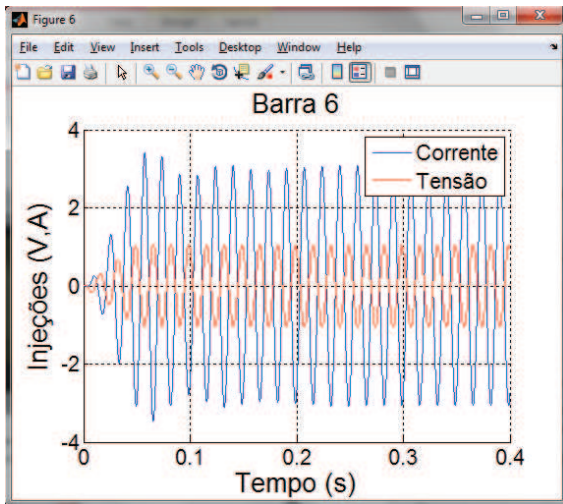


Figura 5.25 – Formas de onda das injeções de corrente e de tensão na barra 6 no simulador

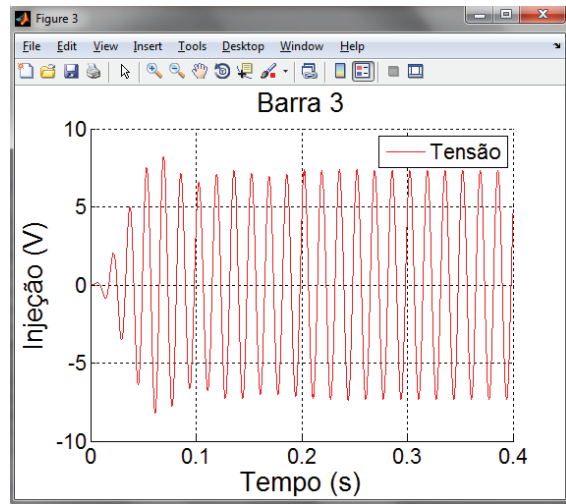


Figura 5.26 – Forma de onda da injeção de tensão na barra 3 no simulador

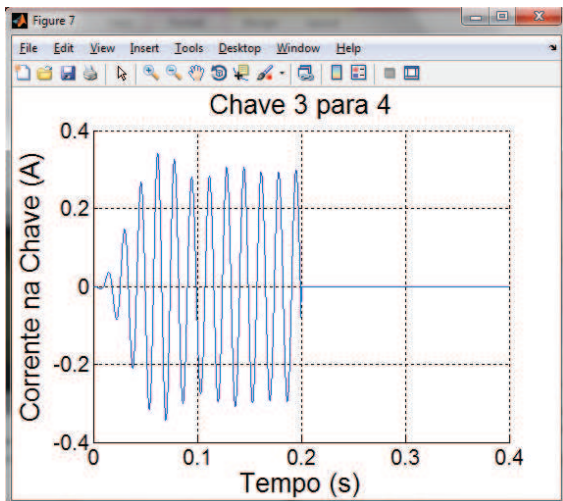


Figura 5.27 – Forma de onda da injeção de corrente na chave 3-4 no simulador

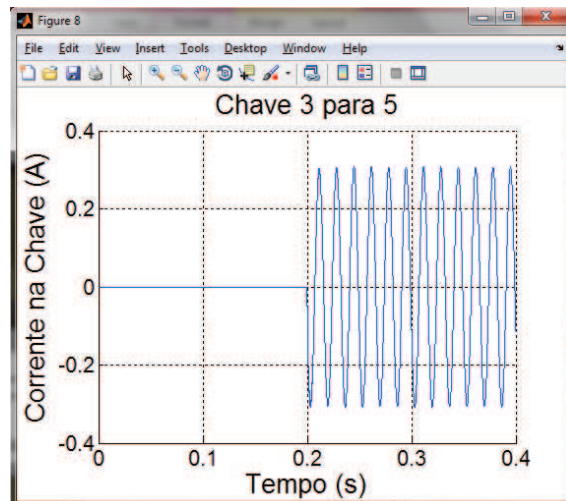


Figura 5.28 – Forma de onda da injeção de corrente na chave 3-5 no simulador

- 2º Caso → A tabela 5.7 descreve os eventos a serem realizados pelas chaves no segundo caso:

Chave	Posição Inicial	Posição Final	Instante de Troca
3-4	Aberta	Fechada	0.2s
3-5	Aberta	Fechada	0.2s

Tabela 5.7 – Descrição dos eventos das chaves para o 2º caso

As figuras 5.29 a 5.31 mostram os resultados das simulações realizadas no Simulink para o segundo caso:

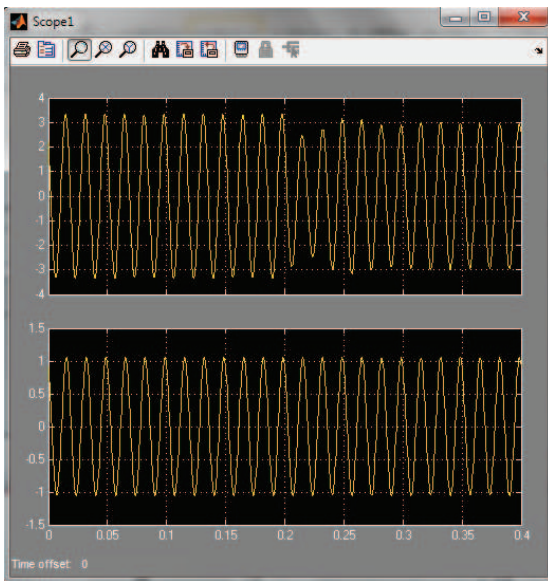


Figura 5.29 – De cima para baixo: formas de onda das injeções de corrente e de tensão na barra 6 no Simulink do Matlab

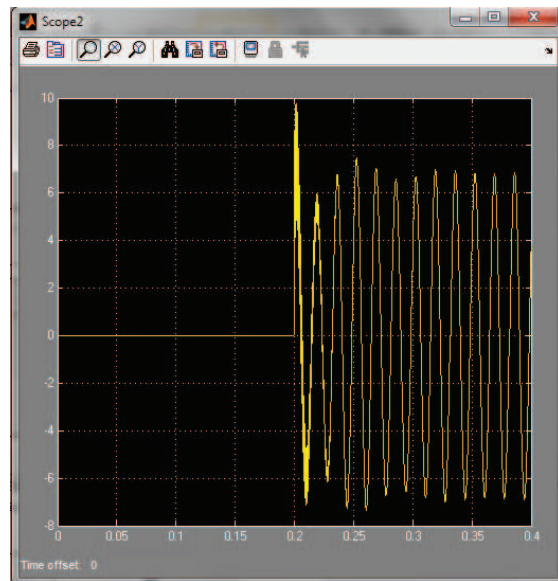


Figura 5.30 – Forma de onda da injeção de tensão na barra 3 no Simulink do Matlab

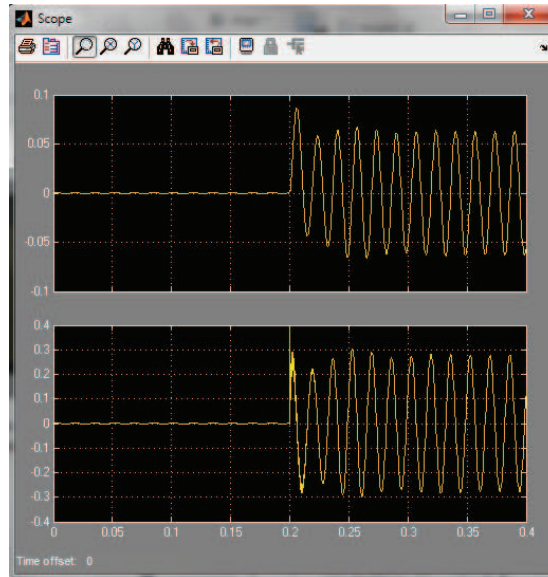


Figura 5.31 – De cima para baixo: formas de onda das injeções de corrente nas chaves 3-4 e 3-5 no Simulink do Matlab

As figuras 5.32 a 5.35 ilustram os resultados obtidos no simulador para este caso:

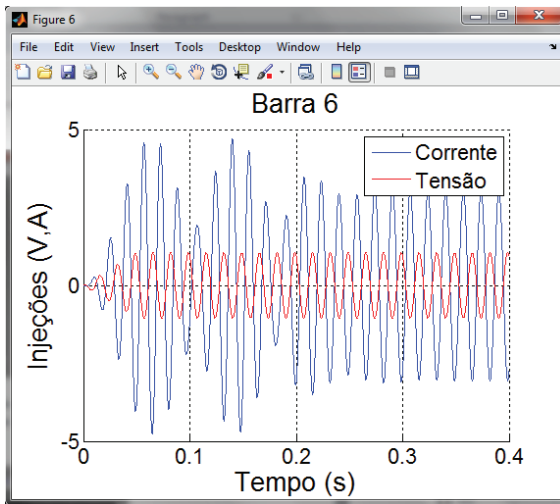


Figura 5.32 – Formas de onda das injeções de corrente e de tensão na barra 6 no simulador

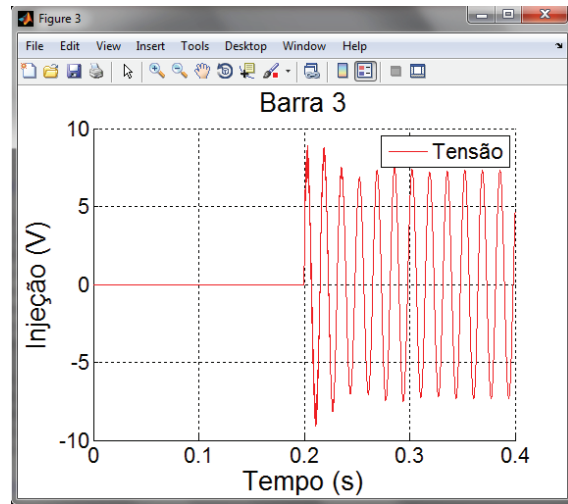


Figura 5.33 – Forma de onda da injeção de tensão na barra 3 no simulador

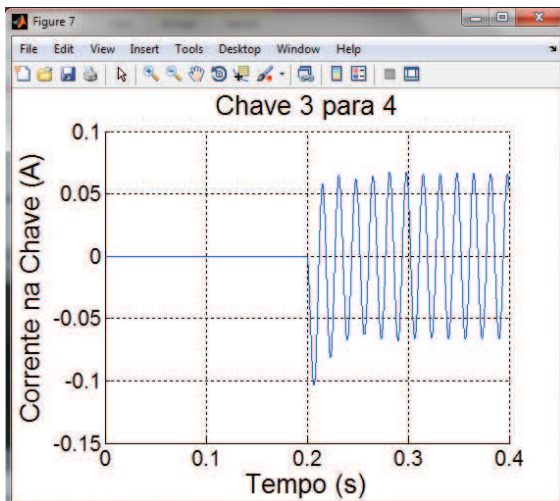


Figura 5.34 – Forma de onda da injeção de corrente na chave 3-4 no simulador

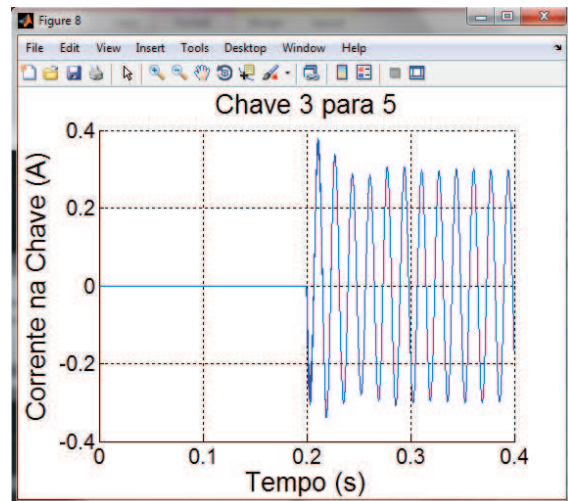


Figura 5.35 – Forma de onda da injeção de corrente na chave 3-5 no simulador

Confrontando os resultados obtidos para os dois casos, observa-se que a chave modelada no simulador funcionou razoavelmente bem, quando comparada à chave encontrada na biblioteca do Simulink. Cabe aqui destacar, o fato de nos primeiros instantes de simulação, a amplitude das injeções de corrente e/ou de tensão nas barras que iniciaram as simulações energizadas, nos dois casos, oscilou apenas nos gráficos correspondentes ao simulador, diferentemente dos gráficos obtidos no Simulink. Isto é devido à utilização do sinal em rampa apenas no simulador (classe *RampSource*) nos primeiros 0.05s de simulação.

5.5. Circuito Trifásico com 3 Linhas de Transmissão

A figura 5.36 a seguir, mostra um circuito trifásico com 3 linhas de transmissão:

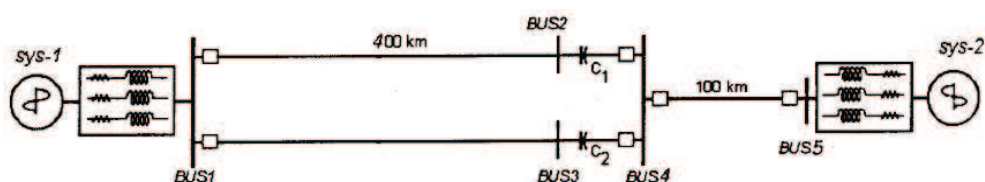


Figura 5.36 – Circuito trifásico com 3 linhas de transmissão

As tabelas 5.8 e 5.9 apresentam os dados utilizados nas simulações:

BarraK	BarraM	R (Ω)	XL (Ω)	XC (Ω)
sys-1	1		11.7282	
1				263.1579
1	2	5.0920	140.7986	
1	3	5.0920	140.7986	
2				526.3158
2	4			0.1005
3		0.5		526.3158
3	4			0.1005
4				2105.3
4	5	1.2730	35.1996	
5				2105.3
5	sys-2	6.5	130.0619	

Tabela 5.8 – Dados do circuito com 3 linhas de transmissão

BarraK	Amp. (V)	Fase (deg)	Freq. (Hz)	Tipo
sys-1	100	0	60	Tensão
sys-2	120	0	60	Tensão

Tabela 5.5 – Dados das fontes de tensão do circuito com 3 linhas de transmissão

O circuito em estudo foi configurado nos simuladores de acordo com os parâmetros de sequência positiva exposto nas tabelas anteriores. Porém, no caso das linhas de transmissão, os parâmetros de sequência zero não foram levados em

conta na preparação dos dados no simulador, apenas o Simulink foi carregado com esses parâmetros. Os referidos parâmetros encontram-se no anexo 5 deste trabalho.

A figura 5.37 mostra a tela do Simulink com o circuito preparado para dois modelos diferentes de linha de transmissão: um com o modelo de parâmetros distribuídos e o outro com modelo pi:

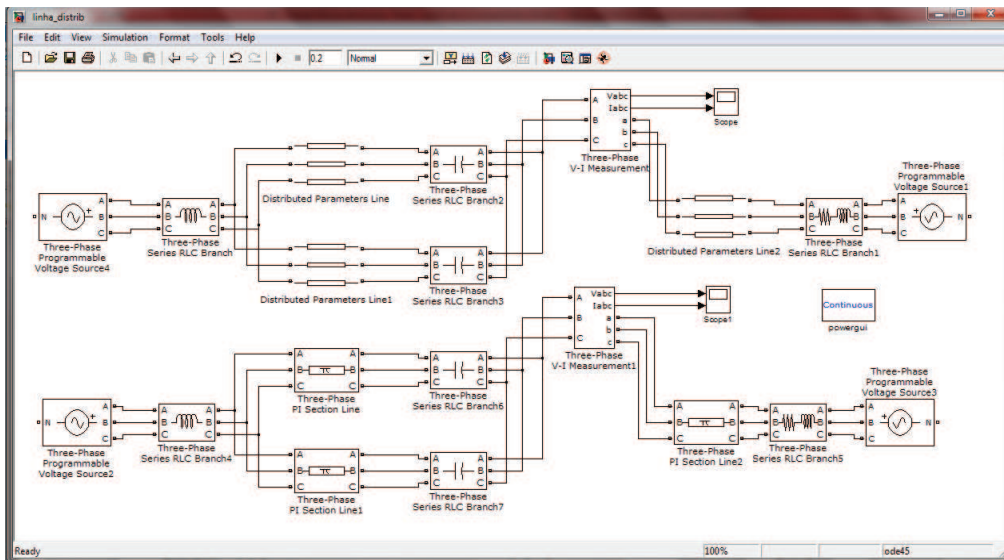


Figura 5.37 – Circuito com 3 linhas de transmissão simulado no Simulink do Matlab. De cima para baixo: linha no modelo com parâmetros distribuídos e linha no modelo pi

As figuras 5.38 e 5.39 mostram os sinais de tensão e de corrente para a barra 4, obtidos no Simulink.

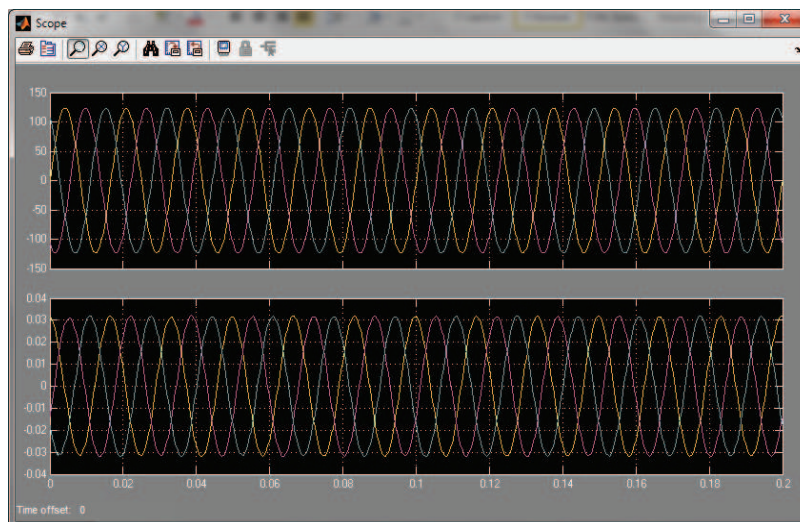


Figura 5.38 – Sinais de tensão e de corrente na barra 4 para o circuito com 3 linhas de transmissão simulado no Simulink com linha no modelo com parâmetros distribuídos

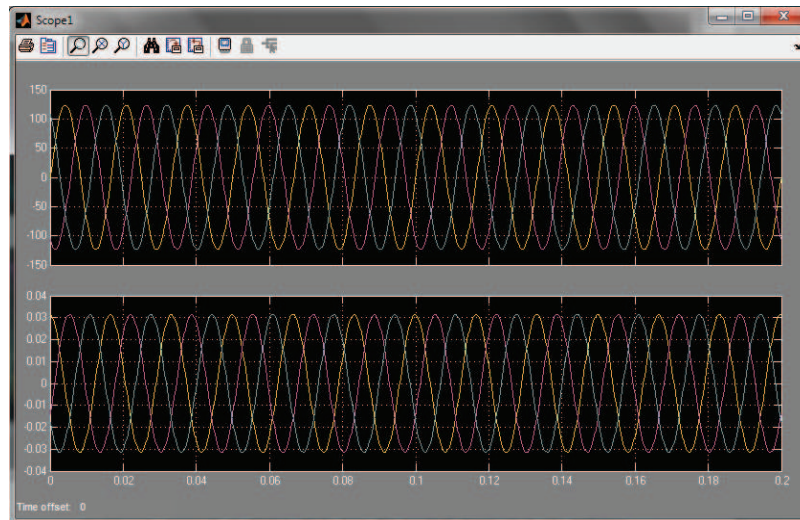


Figura 5.39 – Sinais de tensão e de corrente na barra 4 para o circuito com 3 linhas de transmissão simulado no Simulink com linha no modelo pi

Percebe-se, pela análise das figuras acima que os resultados das simulações para linhas de transmissão com modelos diferentes são semelhantes em regime permanente, não apresentando diferenças visíveis nas formas de onda de tensão e corrente.

O mesmo circuito foi testado no simulador, que nesse caso, seu desempenho não se mostrou satisfatório, quando se trata de simulações com linhas de transmissão. Isto significa que a modelagem utilizada para a referida linha não apresentou resultados coerentes com os encontrados no Simulink. As figuras 5.40 e 5.41 ilustram os resultados obtidos no simulador para a barra 4.

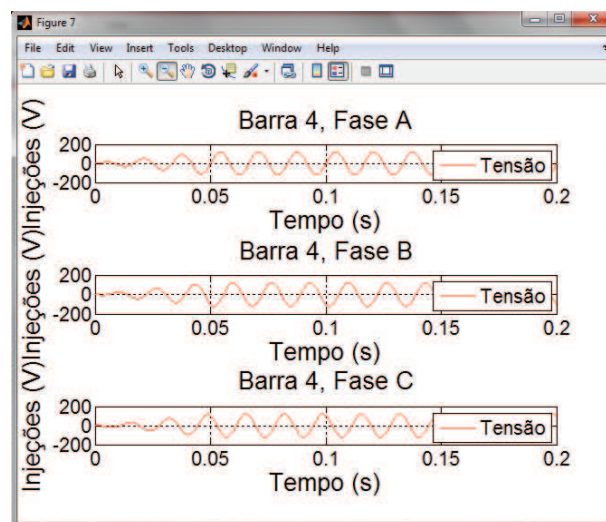


Figura 5.40 – Sinais de tensão na barra 4 para o circuito com 3 linhas de transmissão no simulador

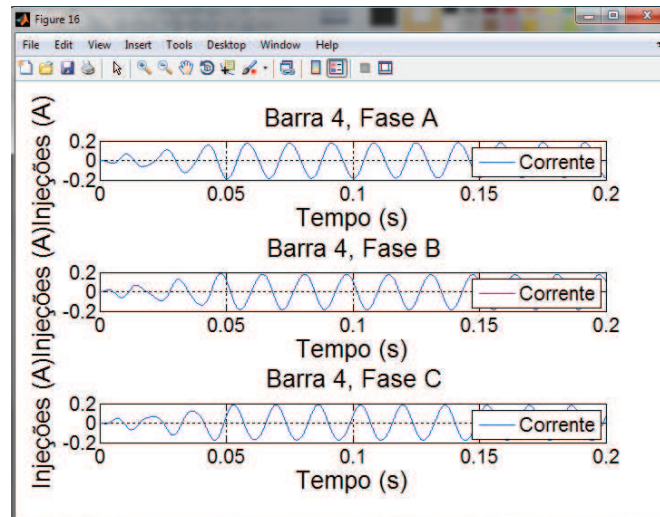


Figura 5.41 – Sinais de corrente na barra 4 para o circuito com 3 linhas de transmissão no simulador

Observa-se, pela análise das figuras 5.38 a 5.41, que as amplitudes das tensões e correntes na barra 4 encontradas no simulador são diferentes das obtidas no Simulink. Portanto, o simulador não é capaz de realizar simulações coerentes em circuitos que contenham linhas de transmissão inseridas em sua topologia.

O fato de a modelagem da linha de transmissão no simulador não ter levado em conta os parâmetros de sequência zero, impactou diretamente nos resultados encontrados pelo mesmo. Assim sendo, torna-se fundamental a inclusão deste na modelagem para a obtenção de resultados melhores.

5.6. Conclusões e Sugestões para Trabalhos Futuros

O presente trabalho buscou a modelagem dos elementos de circuito através da modelagem orientada a objetos. Com isso, foi possível desenvolver um ambiente capaz de simular circuitos elétricos de pequeno porte, bem como transitórios eletromagnéticos através de manobras por chaveamento em instantes de tempo previamente definidos.

Os resultados das simulações se mostraram satisfatórios em todos os circuitos simulados, exceto quando se tratam de circuitos linhas de transmissão. Os comportamentos dos circuitos RL e RLC série foram coerentes com a teoria apresentada durante o período da graduação. Já para os dois casos seguintes, o programa conseguiu realizar seu propósito com sucesso, tanto no circuito sem manobras, quanto no caso em que existem chaves. E no último caso, o simulador não alcançou o resultado esperado.

A ferramenta resultante pode vir a ser utilizada como fonte de pesquisa para iniciantes, e objeto de atualizações no sentido de ampliar o escopo de atuação do programa, incluindo a modelagem correta das linhas de transmissão, a inclusão da modelagem de transformadores, máquinas síncronas, etc. Também fica como sugestão para trabalhos futuros, a implementação deste programa em código aberto, como por exemplo, em C++.

Por fim, este trabalho utiliza-se de uma ferramenta que vem crescendo bastante nos últimos anos na área da programação, a saber, a modelagem orientada a objetos, e o mesmo pode ser útil na difusão deste conhecimento de forma mais aprofundada no nível da graduação.

Referências Bibliográficas

- [1] – J. ARRILIAGA, N. WATSON, “Power Systems Eletromagnetic Transients Simulation”, IEEE power and energy series; no. 39, 2007.
- [2] – J. MAHSEREDJIAN, F. ALVARADO, “Creating an Eletromagnetic Transients Program in MATLAB: MatEMTP”, 1997.
- [3] – C. L. T. BORGES, “Análise de Sistemas de Potência”, Março de 2005.
- [4] – J. RUMBAUGH, M. BLAHA, W. PREMERLANI, F. EDDY, W. LORENSEM, “Modelagem e Projetos Baseados em Objetos”, Ed. Campus, 1994.f
- [5] – J. D. FURLAN, “Modelagem de Objetos através da UML – Análise e Desenho Orientados a Objeto”, Makron Books, 1998.
- [6] – A. MANZONI, “Desenvolvimento de um Sistema Computacional Orientado a Objetos para Sistemas Elétricos de Potência: Aplicação e Simulação Rápida e Análise da Estabilidade de Tensão”, Tese – COPPE-UFRJ, 2005.
- [7] – C. K. ALEXANDER, M. N. O. SADIKU, “Fundamentos de Circuitos Elétricos”, Bookman, 2003.

Anexo I

Script para o Circuito RL Série

```

clear
clc

R = 1;
L = 10e-3;
tau = L/R;

I = zeros(1,10000);
step = tau/100;

RL = PassiveElement(step,0,0,R,L,0);
RL.update_injection(100,0);
RL.update_ikm(100,0);
I(1,1) = RL.injection;

for ii = 2:10000
    RL.update_injection(0,0);
    RL.update_ikm(0,0);
    I(1,ii) = RL.injection;
end

t = (1:1000)*step;
figure
hold on
plot(t,I(1,1:1000),'b-');
legend('step = \tau/100');
title('Circuito RL', 'FontSize',20)
xlabel('Tempo (s)', 'FontSize',20);
ylabel('Corrente (A)', 'FontSize',20);
set(gca, 'FontSize',16);
set(gcf, 'Color', 'w');
grid

```

Anexo II

Script para o Circuito RLC Série

```

clear
clc

R = [40 28.28 10];
L = 10e-3;
C = 50e-6;

I = zeros(3,1500);
step = 10e-6;

for ii = 1:3
    RLC = PassiveElement(step,0,0,R(ii),L,C);
    for jj = 1:1500
        RLC.update_injection(1,0);
        RLC.update_ikm(1,0);
        I(ii,jj) = RLC.injection;
    end
end

t = (1:1500)*step;
figure
hold on
plot(t,I(1,:), 'r-');
plot(t,I(2,:), 'g-');
plot(t,I(3,:), 'b-');
legend('Superamortecido', 'Crit. Amortecido', 'Subamortecido');
title('Circuito RLC', 'FontSize',20)
xlabel('Tempo (s)', 'FontSize',20);
ylabel('Corrente (A)', 'FontSize',20);
set(gca, 'FontSize',16);
set(gcf, 'Color', 'w');
grid

```


Anexo III
Arquivo de Entrada para o Circuito
com 3 Barras

```
% %  
% Dados de entrada para o circuito com 3 barras.  
% %  
% As Conexões de barra seguem a seguinte ordem:  
% BarraK BarraM R XL XC  
% %  
BUS1 BUS2 0.05 0.7539822368615  
BUS2 GROUND 0 0 3.315727981081153  
BUS2 BUS3 0.5  
BUS3 GROUND 0 2.261946710584651  
% %  
% Conexões de Fontes de Corrente e de Tensão  
% BarraK TipodaFonte Amplitude TipodoSinal FaseInicial  
% %  
BUS1 VOLTAGE 10 SINOIDAL 0 PHASE  
% %  
% Fim do arquivo
```

Anexo IV

Arquivo de Entrada para o Circuito com 6 Barras e 2 Chaves

```

% %
% Dados de entrada para o circuito com 6 barras e 2 chaves.
% %
% As Conexões de barra seguem a seguinte ordem:
% BarraK BarraM R XL XC
% %
BUS1 BUS6 0 2.2619
BUS1 BUS2 0.05 0.754
BUS1 GROUND 0 0 3.3157
BUS2 BUS4 0 2.2619
BUS2 BUS5 0.5
BUS2 GROUND 0 0 3315.7
BUS3 GROUND 22.61 7.4343
BUS3 GROUND 0 0 2652.5
% %
% Conexões de Fontes de Corrente e de Tensão
% BarraK TipodaFonte Amplitude TipodoSinal FaseInicial
% %
BUS6 VOLTAGE 1.05 SINOIDAL 120 PHASE
BUS2 CURRENT 1 SINOIDAL 0 PHASE
BUS1 CURRENT 1 SINOIDAL 240 PHASE
% %
% Conexões das Chaves
% BarraK BarraM Posição Inicial da Chave
% %
BUS3 BUS5 SWITCH OPEN
BUS3 BUS4 SWITCH CLOSED
% %
% Programa de manobras das chaves
% Instante de Mudança de Posição Localização da Chave
% %
TIMED_EVENT 0.2 SWITCH BUS3 BUS4
TIMED_EVENT 0.2 SWITCH BUS3 BUS5
% %
% Fim do arquivo

```

Anexo V
Arquivo de Entrada para Circuito com
3 Linhas de Transmissão

```

% %
% Dados de entrada para o circuito com 3 linhas de transmissao
% %
% As Conexões de barra seguem a seguinte ordem:
% BarraK BarraM R XL XC
% %
BUS1 BUS4 0 11.7282
BUS2 BUS5 0 11.7282
BUS3 BUS6 0 11.7282
BUS4 GROUND 0 0 263.1579
BUS5 GROUND 0 0 263.1579
BUS6 GROUND 0 0 263.1579
BUS4 BUS7 5.0920 140.7986
BUS5 BUS8 5.0920 140.7986
BUS6 BUS9 5.0920 140.7986
BUS4 BUS10 5.0920 140.7986
BUS5 BUS11 5.0920 140.7986
BUS6 BUS12 5.0920 140.7986
BUS7 GROUND 0 0 526.3158
BUS8 GROUND 0 0 526.3158
BUS9 GROUND 0 0 526.3158
BUS10 GROUND 0 0 526.3158
BUS11 GROUND 0 0 526.3158
BUS12 GROUND 0 0 526.3158
BUS7 BUS13 0 0 0.1005
BUS8 BUS14 0 0 0.1005
BUS9 BUS15 0 0 0.1005
BUS10 BUS13 0 0 0.1005
BUS11 BUS14 0 0 0.1005
BUS12 BUS15 0 0 0.1005
BUS13 GROUND 0 0 2.1053e+003
BUS14 GROUND 0 0 2.1053e+003
BUS15 GROUND 0 0 2.1053e+003
BUS13 BUS16 1.2730 35.1996
BUS14 BUS17 1.2730 35.1996
BUS15 BUS18 1.2730 35.1996
BUS16 GROUND 0 0 2.1053e+003
BUS17 GROUND 0 0 2.1053e+003
BUS18 GROUND 0 0 2.1053e+003
BUS16 BUS19 6.5 130.0619
BUS17 BUS20 6.5 130.0619
BUS18 BUS21 6.5 130.0619
% %
% Conexões de Fontes de Corrente e de Tensão
% BarraK TipodaFonte Amplitude TipodoSinal FaseInicial
% %
BUS1 VOLTAGE 100 SINOIDAL 0 PHASE
BUS2 VOLTAGE 100 SINOIDAL -120 PHASE
BUS3 VOLTAGE 100 SINOIDAL 120 PHASE
BUS19 VOLTAGE 120 SINOIDAL 0 PHASE

```

BUS20 VOLTAGE 120 SINOIDAL -120 PHASE
BUS21 VOLTAGE 120 SINOIDAL 120 PHASE
% %
% Fim do arquivo

**Parâmetros de Sequência Positiva e Zero para o Circuito com 3 Linhas de
Transmissão**

R1 (Ω/km)	R1 (Ω/km)	L1 (mH/km)	L0 (mH/km)	C1 (nF/km)	C1 (nF/km)
0.01273	0.3864	0.9337	4.1264	12.74	7.751