

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO ACADÊMICO DE ELETROTÉCNICA  
CURSO DE ENGENHARIA ELÉTRICA

ALICE NOGUEIRA FEY  
BRUNO RODRIGUES SOARES  
OLÍVIA NICHELE PERDONCINI

**ESTUDO DO FLUXO DE CARGA CONSIDERANDO OS LIMITES DE  
REATIVOS DAS MÁQUINAS SÍNCRONAS**

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA

2019

ALICE NOGUEIRA FEY  
BRUNO RODRIGUES SOARES  
OLÍVIA NICHELE PERDONCINI

**ESTUDO DO FLUXO DE CARGA CONSIDERANDO OS LIMITES DE  
REATIVOS DAS MÁQUINAS SÍNCRONAS**

Trabalho de conclusão do curso de Graduação em Engenharia Elétrica apresentado à disciplina de Trabalho de conclusão de curso 2, do Departamento Acadêmico de Eletrotécnica (DAELT) da Universidade Tecnológica Federal do Paraná (UTFPR), como requisito parcial para obtenção do título de Engenheiro Eletricista.

Orientadora: Prof.<sup>a</sup> Dr.<sup>a</sup>. Andréa Lucia Costa

CURITIBA

2019

ALICE NOGUEIRA FEY  
BRUNO RODRIGUES SOARES  
OLÍVIA NICHELE PERDONCINI

## ESTUDO DO FLUXO DE CARGA CONSIDERANDO OS LIMITES DE REATIVOS DAS MÁQUINAS SÍNCRONAS

Este Trabalho de Conclusão de Curso de Graduação foi julgado e aprovado como requisito parcial para a obtenção do Título de Engenheiro Eletricista, do curso de Engenharia Elétrica do Departamento Acadêmico de Eletrotécnica (DAELT) da Universidade Tecnológica Federal do Paraná (UTFPR).

Curitiba, 12 de junho de 2019.

---

Prof. Antonio Carlos Pinho, Dr.  
Coordenador de Curso  
Engenharia Elétrica

---

Profa. Annemarien Gehrke Castagna, Mestre  
Responsável pelos Trabalhos de Conclusão de Curso  
de Engenharia Elétrica do DAELT

### ORIENTAÇÃO

---

Andréa Lucia Costa, Dra.  
Universidade Tecnológica Federal do Paraná  
Orientadora

### BANCA EXAMINADORA

---

Nastasha Salame da Silva, Dra.  
Universidade Tecnológica Federal do Paraná

---

Raphael Augusto de Souza Benedito, Dr.  
Universidade Tecnológica Federal do Paraná

---

Andréa Lucia Costa, Dra.  
Universidade Tecnológica Federal do Paraná  
Orientadora

## AGRADECIMENTOS

Agradecemos, primeiramente, às nossas famílias que nos incentivaram durante todo período na universidade, proporcionando apoio e amor.

Aos nossos amigos que demonstraram interesse e foram compreensivos nos momentos de concretização do estudo e ainda nos fornecendo momentos de alegrias e descontração.

A Deus pela saúde, resiliência e energia para concluir este trabalho.

Ao Professor Dr. Raphael Augusto de Souza Benedito pelo amparo no desenvolvimento do algoritmo do FLOBA, que foi fundamental para clareza na evolução deste trabalho.

À Professora Dra. Andrea Lucia Costa pela orientação deste trabalho, disponibilidade e contribuições cedidas para o nosso progresso acadêmico e profissional.

Sobretudo, agradecemos à toda a comunidade da Universidade Tecnológica Federal do Paraná (UTFPR) pela assistência plena, estrutura e ensinamentos ao longo da nossa graduação.

FÜR TÄNZER

Glattes Eis

Ein Paradeis

Für Den, der gut zu tanzen weiss.

(NIETZSCHE, Friedrich W.)

## RESUMO

FEY, Alice, N.; SOARES, Bruno, R.; PERDONCINI, Olívia, N. **Estudo do Fluxo de Carga Considerando os Limites de Reativos das Máquinas Síncronas**. 2019. 100 f. Trabalho de Conclusão de Curso (Graduação – Curso de Engenharia Elétrica). Universidade Tecnológica Federal do Paraná, Curitiba, 2019.

Este trabalho aborda o estudo do Fluxo de Carga não linear para sistemas elétricos de potência, utilizando o Método de Newton-Raphson. É mostrada a estruturação de um algoritmo em MATLAB nomeado FLOBA. O algoritmo calcula o Fluxo de Potência de um sistema de “n” barras, empregando o método escolhido, e considera os limites de potência reativa das máquinas síncronas do sistema. As redes de “n” barras são representadas detalhadamente com os parâmetros de resistência elétrica, reatância indutiva, susceptância capacitiva, geradores como injeções de potência positiva e cargas como injeções de potência negativa, não sendo considerados os limites de tensão nas barras e restrições de capacidade das linhas de transmissão. São considerados como dados de entrada os limites de potência reativa provenientes da curva de capacidade das máquinas síncronas. O usuário do FLOBA, ao executar o algoritmo, toma conhecimento do estado da rede e, caso seja a situação, quais geradores do sistema ultrapassaram as restrições operacionais. É possível a interação do usuário com o algoritmo para que seja eliminada a violação dos limites de reativos para as barras que excederam tal limite, caso seja viável pela configuração do sistema. São apresentados cálculos e simulações para Sistemas Teste, sendo validados satisfatoriamente os resultados obtidos com os cálculos implementados pelo algoritmo através da utilização dos programas ANAREDE e SIMPOT.

**Palavras chave:** Fluxo de Carga. Sistema Elétrico de Potência. Método de Newton-Raphson. Geração de reativos.

## ABSTRACT

FEY, Alice, N.; SOARES, Bruno, R.; PERDONCINI, Olívia, N. **Load Flow Study Considering the Limits of Reactive Power of Synchronous Machines**. 2019. 100 f. Trabalho de Conclusão de Curso (Graduação – Curso de Engenharia Elétrica). Universidade Tecnológica Federal do Paraná, Curitiba, 2019.

This work presents the study of nonlinear Load Flow for electric power systems using the Newton-Raphson Method. It's shown the structuring of a MATLAB algorithm, entitled FLOBA. The algorithm calculates the Power Flow of a system of "n" bus-bars, using the method chosen, and considers the reactive power limits of the synchronous machines of the system. The "n" bars system is represented in detail with the parameters of electrical resistance, inductive reactance, capacitive susceptance, generators as positive power injections and loads as negative power injections, the limits of voltage in the bars and capacity constraints of the transmission lines are not considered. The reactive power limits from the synchronous machine capability curve are considered as input data. The FLOBA user, when executing the algorithm, becomes aware of the state of the network and, if it is the case, which generators of the system have exceeded the operational constraints. It is possible for the user to interact with the algorithm in order to eliminate the reactive limits violation for the bars that exceeded the limit, if that is a feasible option by the system configuration. Test Systems calculations and simulations are presented, a satisfactory validation of the results were obtained with the implemented algorithm's calculations through the use of ANAREDE and SIMPOT programs.

**Keywords:** Load Flow. Electrical Power System. Newton-Raphson Method. Reactive Power Generation.

## LISTA DE FIGURAS

Figura 1 – Representação do sentido do Fluxo de Carga conforme Leis de Kirchhoff .....	21
Figura 2 – Simbologia padronizada dos elementos de um SEP para o estudo do Fluxo de Potência .....	22
Figura 3 – Diagrama unifilar de um sistema elétrico .....	22
Figura 4 – Modelo equivalente $\pi$ de uma linha de transmissão .....	23
Figura 5 – Exemplo de sistema elétrico de potência de três barras.....	24
Figura 6 – Modelo $\pi$ -equivalente de um sistema elétrico de potência de três barras.....	24
Figura 7 – Circuito representativo do gerador síncrono.....	32
Figura 8 – Diagramas fasoriais	
(a) para o gerador síncrono sobreexcitado .....	33
(b) para o gerador síncrono subexcitado .....	33
Figura 9 – Exemplo de curva de capacidade de um gerador .....	34
Figura 10 – Fluxograma resumido para a resolução do Fluxo de Carga pelo método de Newton-Raphson.....	41
Figura 11 – Sistema de duas barras .....	43
Figura 12 – Matriz de dados de barra do sistema.....	51
Figura 13 – Matriz de dados de linha do sistema.....	52
Figura 14 – Considerações iniciais do algoritmo “FLOBA.m” .....	53
Figura 15 – Montagem da Matriz Y .....	54
Figura 16 – Cálculo dos valores iniciais de injeção de potência.....	54
Figura 17 – Criação das matrizes G e B.....	54
Figura 18 – Cálculo do número de barras $PQ$ e $PV$ .....	55
Figura 19 – Montagem dos vetores V e “theta” .....	55
Figura 20 – Apresentação da rotina de aviso.....	56
Figura 21 – Cálculos das injeções de potência ativa ( $P_i$ ) e reativa ( $Q_i$ ).....	56
Figura 22 – Teste dos resíduos de potência .....	57
Figura 23 – Teste de convergência do sistema .....	58
Figura 24 – Cálculo da submatriz H .....	59
Figura 25 – Inserindo valores de H na Matriz Jacobiana Reduzida.....	60
Figura 26 – Cálculo de $J_i$ e $\Delta X$ .....	60
Figura 27 – Vetor X e atualização do Vetor X.....	61
Figura 28 – Atualização dos vetores “theta” e V .....	61



Figura 29 – Rotina para apresentar o estado da rede.....	62
Figura 30 – Relatório FLOBA: Estado da rede e quantidade de iterações utilizando o Sistema Teste de 6 barras.....	63
Figura 31 – Rotina para verificação de geração de reativos nas barras .....	63
Figura 32 – Fluxograma detalhado do monitoramento da geração de reativos das máquinas síncronas do sistema.....	64
Figura 33 – Rotina para a verificação da violação do limite de geração de potência reativa ..	65
Figura 34 – Rotina para indicar a barra que obteve maior excedente de geração ou consumo de potência reativa .....	65
Figura 35 – Relatório FLOBA: Monitoramento da geração/consumo de reativos das máquinas síncronas do sistema.....	66
Figura 36 – Rotina para mostrar opções ao usuário .....	66
Figura 37 – Relatório FLOBA .....	67
Figura 38 - Rotina para alteração dos parâmetros das barras e recálculo do Fluxo de Potência .....	68
Figura 39 – Rotina para finalização da simulação .....	68
Figura 40 – Rotina para finalizar a simulação quando não há violação dos limites .....	69
Figura 41 – Mensagem impressa na janela de comando quando não há violação dos limites.	69
Figura 42 - Fluxograma de execução do algoritmo FLOBA .....	70
Figura 43 – Resultados obtidos para o estado da rede do sistema 6 barras com o algoritmo FLOBA.....	71
Figura 44 – Relatório (RBAR) do sistema teste de 6 barras com o programa ANAREDE.....	72
Figura 45 - Dados de entrada do sistema teste de 6 barras para execução no ANAREDE.....	72
Figura 46 - Resultados obtidos para o estado da rede do sistema teste de 3 barras com o algoritmo FLOBA .....	73
Figura 47 – Resultados obtidos na simulação do sistema 3 barras com o programa ANAREDE e dados de entrada .....	73
Figura 48 - Resultados de Qg obtidos na simulação do sistema 3 barras pelo FLOBA e pelo ANAREDE.....	74
Figura 49 – Sistema-teste de 6 barras utilizado nas simulações FLOBA .....	76
Figura 50 - Relatório de resultados FLOBA com interação do usuário (parte a).....	79
Figura 51 - Relatório de resultados FLOBA com interação do usuário (parte b) .....	80
Figura 52 - Matrizes DBARoriginal e DBAR.....	81
Figura 53- Envio do número da barra de referência.....	81

Figura 54 - Envio do número de alguma barra que não violou os limites de reativos.....	81
Figura 55 - Envio do número de alguma barra que não consta no sistema analisado.....	82
Figura 56 - Relatório de resultados FLOBA com interação do usuário - outra possibilidade (parte a) .....	83
Figura 57 - Relatório de resultados FLOBA com interação do usuário - outra possibilidade (parte b) .....	84

## LISTA DE SIGLAS

ANAREDE	Análise de Redes Elétricas
CEPEL	Centro de Pesquisas de Energia Elétrica
DBAR	Matriz de Dados de Barras
DLIN	Matriz de Dados de Linhas
FLOBA	Algoritmo Implementado para o Cálculo de Fluxo de Potência
LT	Linha de Transmissão
MATLAB <sup>®</sup>	<i>Matrix laboratory</i>
NUMBAR	Número de Barras
NUMLIN	Número de Linhas
PRODIST	Procedimentos de Distribuição de Energia Elétrica no Sistema Elétrico Nacional
PU	Por Unidade
SEP	Sistemas Elétricos de Potência
SIMPOT	Simulação de Fluxo de Potência pelo método Newton-Raphson
UTFPR	Universidade Tecnológica Federal do Paraná

## SUMÁRIO

<b>1. INTRODUÇÃO .....</b>	<b>13</b>
1.1. TEMA.....	13
<b>1.1.1. Delimitação do tema.....</b>	<b>14</b>
1.2. PREMISSAS E PROBLEMAS.....	15
1.3. OBJETIVOS .....	16
<b>1.3.1. Objetivo geral .....</b>	<b>16</b>
<b>1.3.2. Metas e Objetivos específicos .....</b>	<b>16</b>
1.4. JUSTIFICATIVA.....	17
1.5. METODOLOGIA .....	18
1.6. ESTRUTURA DO TRABALHO .....	18
<b>2. FLUXO DE CARGA.....</b>	<b>20</b>
2.1. INTRODUÇÃO .....	20
2.2. FORMULAÇÃO DO PROBLEMA DE FLUXO DE POTÊNCIA.....	21
2.3. MÉTODOS DE SOLUÇÃO do FLUXO DE POTÊNCIA.....	28
2.4. PRINCIPAIS LIMITES A SEREM CONSIDERADOS NO FLUXO DE CARGA.....	31
<b>2.4.1. Fornecimento de potência reativa pelo gerador .....</b>	<b>32</b>
2.5. CONCLUSÃO .....	34
<b>3. MÉTODO DE NEWTON-RAPHSON .....</b>	<b>35</b>
3.1. INTRODUÇÃO .....	35
3.2. FORMULAÇÃO DO MÉTODO DE NEWTON-RAPHSON APLICADO AO FLUXO DE CARGA .....	35
3.3. ALGORITMO DE NEWTON-RAPHSON PARA O FLUXO DE CARGA.....	38
3.4. RESOLUÇÃO DE EXERCÍCIO DE FLUXO DE CARGA E APLICAÇÃO DO MÉTODO DE NEWTON-RAPHSON .....	42
<b>3.4.1. Resolução de um exemplo numérico de Fluxo de Carga para um sistema de duas Barras .....</b>	<b>42</b>
3.5. CONCLUSÃO .....	49
<b>4. ALGORITMO FLOBA .....</b>	<b>50</b>
4.1. INTRODUÇÃO .....	50
4.2. DADOS DO SISTEMA .....	50
4.3. ESTRUTURA DO ALGORITMO .....	52

4.3.1. Rotina que calcula o Fluxo de Potência.....	52
4.3.2. Rotina que apresenta o estado da rede.....	62
4.3.3. Rotina para verificar a geração de reativos nas barras.....	63
4.3.4. Variáveis que podem ser visualizadas pelo usuário.....	70
4.4. VALIDAÇÃO DO ALGORITMO .....	71
4.5. CONCLUSÃO .....	74
<b>5. ESTUDOS DE CASO E ANÁLISES .....</b>	<b>76</b>
5.1. INTRODUÇÃO .....	76
5.2. SISTEMA DE 6 BARRAS.....	76
5.3. CONCLUSÃO .....	84
<b>6. CONCLUSÕES OBTIDAS E CONSIDERAÇÕES FINAIS .....</b>	<b>85</b>
<b>REFERÊNCIAS .....</b>	<b>87</b>
<b>ANEXOS .....</b>	<b>89</b>
ANEXO A – Relatório de Saída do Programa SIMPOT para o sistema de 2 barras ..	89
<b>APÊNDICES.....</b>	<b>91</b>
APÊNDICE A – ALGORITMO “dadosdosistemaseisbarras.m” .....	91
APÊNDICE B – ALGORITMO “FLOBA.m” .....	92

## 1. INTRODUÇÃO

### 1.1. TEMA

Desde sua descoberta, a energia elétrica mostra-se um elemento crucial no progresso econômico e na qualidade de vida da sociedade. Atualmente é indubitável a dependência do ser humano aos serviços relacionados à energia elétrica, não obstante a dependência às concessionárias de energia e de como projetam e operam os sistemas elétricos de potência.

Com o aumento do emprego da energia elétrica, tanto para fins industriais quanto residenciais, as unidades geradoras tornaram-se gradativamente mais distantes dos consumidores. Neste cenário, surgiram as subestações e as linhas de transmissão (LTs), elementos que permitiram a formação dos grandes Sistemas Elétricos de Potência (SEP). A estruturação de um SEP tem como objetivo gerar, transmitir e distribuir energia elétrica em conformidade com padrões de confiabilidade, disponibilidade, qualidade, segurança e custos (GOMEZ-EXPOSITO et al., 2011).

Para garantir a confiabilidade no fornecimento de energia elétrica deve-se assegurar o correto funcionamento dos sistemas elétricos de potência. Tendo isso em vista, o planejamento da operação do SEP deve respeitar alguns fatores limitantes, estes que são relacionados às características das linhas de transmissão, das máquinas e imposições normativas. Os limites mais comuns são as restrições de Fluxo de Potência ativa nas LTs, geração de reativos pelas unidades geradoras e as tensões nas barras do sistema, que devem estar dentro de intervalos de valores predeterminados.

Para assegurar que essas e outras restrições sejam atendidas durante a operação em tempo real, são feitas simulações computacionais do SEP, utilizando programas que calculam o Fluxo de Carga, também chamado Fluxo de Potência.

Há diversos métodos de solução do Fluxo de Potência, bem como várias ferramentas computacionais para simular a operação do sistema em regime permanente. Dentre estas, destaca-se no Brasil o programa ANAREDE, desenvolvido pelo Centro de Pesquisas de Energia Elétrica (CEPEL). Este programa é empregado nas concessionárias de energia elétrica, empresas do setor e em universidades, em suas versões acadêmicas (CEPEL, 2018).

De maneira geral, o cálculo do Fluxo de Potência consiste na modelagem da rede elétrica por meio de equações não-lineares, que requer o emprego de processos iterativos para

sua solução. Dentre os métodos iterativos mais utilizados encontram-se os métodos de Newton-Raphson, Gauss-Seidel, Métodos Desacoplados e Fluxo de Potência Linearizado (MONTICELLI, 1983; MONTICELLI e GARCIA, 2003; GOMEZ-EXPOSITO et al, 2011).

Os métodos podem ser complexos e mais detalhados com o acréscimo de variáveis de controle e detalhamento das máquinas, bem como podem ser simplificados, como os métodos linearizados. O método de Newton-Raphson é bastante eficiente, obtendo boa convergência mesmo para sistemas de grande porte (SALGADO, 2016).

Por sua ampla utilização, o método de Newton-Raphson foi estudado e utilizado para o desenvolvimento do algoritmo apresentado neste trabalho.

### **1.1.1. Delimitação do tema**

O objetivo deste trabalho de conclusão de curso foi estudar o problema de Fluxo de Carga não linear usando o método de Newton-Raphson, e estruturar um algoritmo de Fluxo de Carga de forma a considerar os limites de potência reativa dos geradores.

Foi elaborado um algoritmo computacional em MATLAB para o cálculo do Fluxo de Potência de um sistema de “n” barras. Além disso, foi utilizado o programa ANAREDE como ferramenta computacional para análise e comparação dos resultados obtidos com o algoritmo desenvolvido.

O detalhamento da rede limitou-se à representação dos parâmetros de resistência elétrica, reatância indutiva e susceptância capacitiva, não sendo representadas variáveis de controle como taps de transformadores. Os geradores foram representados como injeções de potência positiva e as cargas como injeções de potência negativa.

O trabalho limitou-se ao estudo e análise das restrições relativas à geração de potência reativa das máquinas síncronas. Restrições de capacidade das linhas de transmissão ou limites de tensão nas barras não foram considerados.

## 1.2. PREMISSAS E PROBLEMAS

Fluxo de Carga, ou Fluxo de Potência, é o estudo empregado para determinar valores de tensão, corrente e potências ativas e reativas nos diversos pontos de uma rede elétrica, ou seja, empregado para se determinar o estado da rede. A modelagem do Fluxo de Potência é estática e considera a operação em regime permanente, resultando em um sistema de equações no qual as variações no tempo podem ser desprezadas (MONTICELLI, 1983). O Fluxo de Potência é usado para simular uma condição de operação e confirmar se a geração, determinada pelo despacho econômico, será suficiente para atender a uma demanda prevista.

O despacho econômico consiste na distribuição de uma determinada demanda entre as unidades geradoras do sistema, de tal forma que os custos totais de produção possam ser minimizados (STEVENSON, 1978). As potências a serem geradas pelas unidades, resultantes do despacho econômico, podem ser utilizadas como variáveis de entrada de um programa de Fluxo de Carga, para que então seja verificado o estado da rede elétrica nessa condição de geração e carga.

Todavia, um problema que pode ocorrer é que os resultados do Fluxo de Carga podem não atender as restrições impostas pelas normas operativas ou devido às características das máquinas. Por exemplo, podem ocorrer valores de tensão fora de um intervalo preestabelecido porque não há geração de potência reativa suficiente. Ou seja, verifica-se neste caso que o despacho determinado a priori não é adequado para a operação da rede. Assim sendo, diz-se que a rede restringiu o despacho. Desta forma é preciso ajustar os despachos dos geradores, isto é, a potência a ser fornecida por estes, até que os resultados obtidos pelo Fluxo de Potência possam atender, além da carga demandada, os limites operacionais dos equipamentos e os limites de transmissão.

Com o emprego de ferramentas computacionais, as simulações minimizaram as inúmeras inconsistências de análise. No entanto, por envolver muitos cálculos, ainda se mostra trabalhosa a determinação das diversas variáveis envolvidas no sistema quando se almeja considerar as restrições do sistema e os limites físicos impostos (SCHIOCHET, 2006).

Este trabalho tem como objetivo abordar o método de Newton-Raphson para a solução de problemas de Fluxo de Carga, também explanar os conceitos necessários para a implementação de um algoritmo de Fluxo de Carga não linear que leve em conta os limites de potência reativa dos geradores em sua estrutura. Houve, como desafio, analisar as condições no algoritmo que refletissem a limitação dos geradores, visando representar cenários mais realistas.



Assim, os resultados provenientes a partir dos cálculos efetuados pelo algoritmo servem de base para uma análise mais adequada da operação do SEP, por descreverem o comportamento da rede elétrica considerando os limites de operação das máquinas.

### 1.3. OBJETIVOS

#### 1.3.1. Objetivo geral

Implementar um algoritmo de Fluxo de Carga que considere os limites de reativos das máquinas síncronas.

#### 1.3.2. Metas e Objetivos específicos

- Estudar métodos de cálculo de Fluxo de Potência;
- Estudar o Método de Newton-Raphson e realizar exercícios de cálculo de Fluxo de Potência;
- Identificar os possíveis fatores limitantes do Fluxo de Potência na transmissão de energia elétrica, bem como a maneira que os métodos de cálculo de Fluxo de Potência consideram os limites de reativos das máquinas síncronas;
- Utilizar a ferramenta computacional MATLAB para implementar um algoritmo de Fluxo de Potência pelo método de Newton-Raphson;
- Estudar o programa ANAREDE para Fluxo de Potência;
- Simular exemplos de Sistemas Teste utilizando os programas SIMPOT e ANAREDE;
- Comparar os resultados obtidos pelo algoritmo implementado no MATLAB, com os resultados obtidos por meio do programa ANAREDE, para validar as soluções encontradas;
- Inserir no algoritmo de cálculo de Fluxo de Potência implementado os limites de reativo das máquinas síncronas. Aperfeiçoar o algoritmo para que mostre as etapas de cálculo e resultados intermediários, visando aplicabilidade didática.

#### 1.4. JUSTIFICATIVA

O método de Newton-Raphson foi escolhido para os cálculos do Fluxo de Potência e elaboração de um algoritmo computacional, pois dentre os métodos existentes, esse apresenta uma melhor convergência dos valores das variáveis por meio de aproximações de primeira ordem das funções não lineares envolvidas no cálculo. Dessa forma, a convergência ocorre de maneira mais rápida, levando de três a cinco iterações, mesmo para sistemas com maior número de barras (MONTICELLI e GARCIA, 2003).

Nos cálculos do Fluxo de Potência, através do método de Newton-Raphson, não se tem conhecimento prévio das correntes nas linhas de transmissão e transformadores, as potências demandadas das barras de carga, e as potências ativas geradas. As potências reativas geradas e as tensões nas barras de carga são calculadas no processo iterativo. O método por si só não considera nenhum limite para os valores calculados em suas iterações, ou seja, não são considerados limites para os valores das tensões ou das potências reativas que os geradores devem fornecer, de modo a se obter a convergência no processo iterativo. Sendo assim, o desafio deste trabalho foi propor um algoritmo que, além de calcular o Fluxo de Carga, considere os limites de reativos da geração na solução final. Com isso, torna-se possível analisar o desempenho do sistema e verificar se o despacho é viável para essa configuração da rede e condição de carga, possibilitando a detecção de problemas que possam provocar possíveis contingências no sistema.

O uso de ferramentas computacionais, bem como o conhecimento de seu funcionamento, foi de extrema importância para o desenvolvimento deste trabalho, devido à complexidade da modelagem matemática de um sistema elétrico de potência. Torna-se inviável a obtenção de resultados confiáveis, considerando-se sistemas mais próximos aos reais, sem tomar como base as ferramentas computacionais de cálculo e simulação.

A execução de um algoritmo computacional que realize os cálculos do Fluxo de Potência e leve em consideração os limites de reativos da geração, visando aplicabilidade didática, mostra-se de grande valia para complementar os estudos e o entendimento do Fluxo de Carga em sistemas elétricos de potência.

## 1.5. METODOLOGIA

A execução deste trabalho teve como base a realização de pesquisas em artigos, teses, dissertações e livros relacionados ao tema. Dentre os diversos métodos empregados para o cálculo de Fluxo de Carga, foi escolhido o método de Newton-Raphson. Sua fundamentação teórica foi estudada detalhadamente, sendo também realizados cálculos de exemplos elementares disponíveis na literatura, a fim de se obter um bom entendimento sobre esse método.

Para efetuar simulações e validação dos cálculos de Fluxo de Carga, visando também a familiarização com ferramentas computacionais empregadas na modelagem dos SEP, foi utilizado o programa ANAREDE<sup>1</sup>, fornecido gratuitamente em versão acadêmica pelo CEPEL, bem como o programa SIMPOT, elaborado por alunos da UTFPR como trabalho de conclusão do curso de graduação de Engenharia Industrial Elétrica (BUENO; TAVARES; MAROCHI, 2012). Também se utilizou a ferramenta computacional MATLAB para executar e testar o algoritmo elaborado.

Para validar o algoritmo desenvolvido neste trabalho foram simulados sistemas teste conhecidos da literatura, utilizando o programa ANAREDE e o algoritmo proposto. Os resultados foram comparados e analisados.

## 1.6. ESTRUTURA DO TRABALHO

O trabalho proposto apresenta a seguinte estrutura:

- Capítulo 1: delimitação do tema; revisão bibliográfica; definição da metodologia de pesquisa e de atividades a serem realizadas;
- Capítulo 2: formulação do problema de Fluxo de Potência e descrição dos fatores que devem ser analisados, tais como limites de transmissão, tensões de operação das barras e geração de reativos, de modo a incluir as restrições de operação dos equipamentos, bem como garantir a qualidade de energia elétrica fornecida. Breve teoria sobre os limites de reativo das máquinas síncronas;

---

<sup>1</sup> CEPEL - CENTRO DE PESQUISAS DE ENERGIA ELETRICA - PROGRAMA DE ANALISE DE REDES - V10.01.03

- Capítulo 3: estudo do método de Newton-Raphson e resolução de exercícios de Fluxo de Potência;
- Capítulo 4: apresentação do algoritmo implementado. Validação do algoritmo comparando os resultados obtidos com simulações no ANAREDE;
- Capítulo 5: simulações computacionais com sistemas teste, análises e conclusões;
- Capítulo 6: conclusões e considerações finais.

## 2. FLUXO DE CARGA

### 2.1. INTRODUÇÃO

O Fluxo de Potência de um SEP, ou Fluxo de Carga, é o estudo para determinar-se o estado de uma rede elétrica, como suas tensões complexas e seus limites de transmissão (MONTICELLI, 1983).

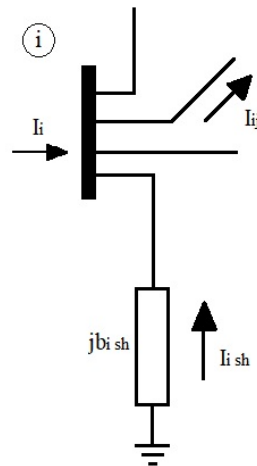
Equações algébricas não lineares são utilizadas no estudo do Fluxo de Potência, sendo representado o sistema em condição estática, ignorando efeitos transitórios. Matematicamente, é um problema de resolução de sistemas de equações não lineares de modo iterativo. Os geradores e as cargas são modelados como injeção de potência nas barras e os transformadores, reatores e linhas de transmissão por seus circuitos equivalentes (SALGADO, 2016).

Determina-se, então, o modo operativo do SEP, os riscos técnicos, desvios de tensão, perdas elétricas e instabilidade da rede, pelo estudo do Fluxo de Carga. Pode-se, ainda, observar se o sistema está operando de forma correta e segura, bem como o que pode ser feito para corrigir as situações inadequadas de operação, e o planejamento de expansão da rede para analisar o aumento da demanda. Assim, busca-se garantir que o perfil de tensão esteja dentro de limites pré-especificados e os equipamentos e as linhas de transmissão operem sem sobrecarga.

Os elementos do sistema de potência podem ser considerados internos ou externos (MONTICELLI, 1983). A barra de um sistema é um ponto no qual estes elementos se interconectam. Na parte interna do sistema, tem-se linhas de transmissão, transformadores e defasadores, sendo eles ligados entre dois nós da rede. Os equipamentos que estão conectados à uma barra (ou nó) qualquer e à barra (ou nó) terra são ditos externos ao sistema, sendo esses os geradores, cargas, reatores e capacitores.

Empregam-se as Leis de Kirchhoff no Fluxo de Carga, para análise nodal de forma generalizada, sabendo que a soma das potências dos componentes internos equivale à potência líquida injetada no sistema e que a injeção de potências ativa e reativa em cada barra da rede deve ser conservada (MONTICELLI, 1983). Considerando o enunciado pelas Leis de Kirchhoff, como ilustrado na Figura 1, encontra-se a expressão matemática dada pela Equação 2.1.

Figura 1 – Representação do sentido do Fluxo de Carga conforme Leis de Kirchhoff



Fonte: Autoria própria.

$$[\dot{I}] = [Y_{barra}] \cdot [\dot{V}] \quad (2.1)$$

Sendo:

$[\dot{I}]$ : vetor das injeções de correntes nas barras, provenientes dos componentes externos do sistema;

$[\dot{V}]$ : vetor das tensões complexas nas barras do sistema;

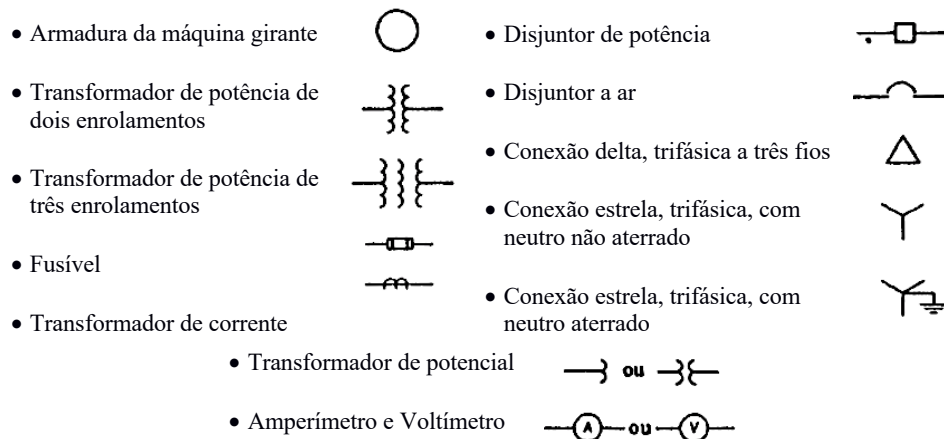
$[Y_{barra}]$ : matriz de admitâncias dos componentes internos do sistema.

## 2.2. FORMULAÇÃO DO PROBLEMA DE FLUXO DE POTÊNCIA

Os elementos mais relevantes de um sistema elétrico de potência são representados por meio de um diagrama unifilar, que possibilita a análise do sistema de maneira concisa e simplificada (STEVENSON, 1978).

Para facilitar o estudo e os cálculos, representa-se os sistemas trifásicos por seus equivalentes monofásicos. Inicialmente o sistema é representado com seus componentes por símbolos padronizados, conforme a Figura 2, e omite-se o neutro, por não haver circulação de corrente pelo mesmo e por estar no mesmo potencial que o terra, considerando uma condição de equilíbrio no sistema (STEVENSON, 1978). A partir do diagrama unifilar, como mostrado na Figura 3, é modelado o diagrama de impedâncias equivalente para o sistema a ser estudado.

Figura 2 – Simbologia padronizada dos elementos de um SEP para o estudo do Fluxo de Potência



Fonte: Adaptado de Stevenson (1978).

Figura 3 – Diagrama unifilar de um sistema elétrico



Fonte: Autoria própria.

Os equipamentos de um SEP são considerados, no problema do Fluxo de Carga, com seus valores de impedâncias representados em *pu* (por unidade), considerando uma base comum para todo sistema (STEVENSON, 1978).

Na formulação do Fluxo de Carga, são associadas quatro variáveis a cada barra do sistema elétrico, sendo duas dadas e duas incógnitas, dependendo do tipo de barra. As variáveis são as seguintes: a magnitude da tensão nodal ( $V$ ), o ângulo da tensão nodal ( $\theta$ ), a injeção de potência ativa líquida ( $P$ ) e a injeção de potência reativa ( $Q$ ). Definem-se três tipos de barras:  $PQ$ ,  $PV$  e  $V\theta$ , dependendo da função determinada para a barra do sistema e do modo de entrada de seus dados (MONTICELLI, 1983).

As barras do tipo  $PQ$  representam as barras de carga e não possuem controle de tensão. São dados para as barras  $PQ$  os valores de injeção de potência ativa ( $P$ ) e a injeção de potência reativa ( $Q$ ) e são calculados os valores de tensão nodal e ângulo de tensão nodal (SALGADO, 2016).

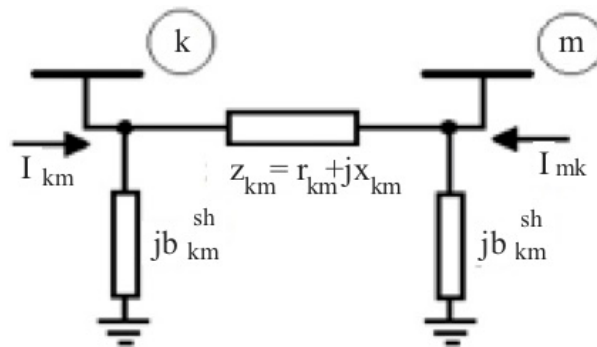
As barras do tipo  $PV$  representam as barras de geração, com controle do módulo da tensão. São fornecidos os valores de injeção de potência ativa ( $P$ ) e magnitude da tensão nodal

(V). São calculados os valores de injeção líquida de potência reativa e ângulo de tensão nodal, conforme mostrado por Monticelli (1983).

A barra do tipo  $V\theta$ , chamada também de referência angular do sistema ou barra de folga, ou de barra *swing*, representa um gerador tomado como barra de referência angular e utilizada para completar o balanço de potência do sistema, considerando as perdas de transmissão (SALGADO, 2016). Para esse tipo de barra são fornecidos valores de  $V$  e ângulo  $\theta$ , sendo calculadas a injeção de potência ativa e a injeção de potência reativa.

Para representar um sistema, modelam-se os elementos como linhas de transmissão e transformadores, utilizando-se do modelo  $\pi$ , conforme mostrado na Figura 4. Este modelo é empregado pois sua configuração facilita a elaboração dos cálculos do Fluxo de Carga.

Figura 4 – Modelo equivalente  $\pi$  de uma linha de transmissão

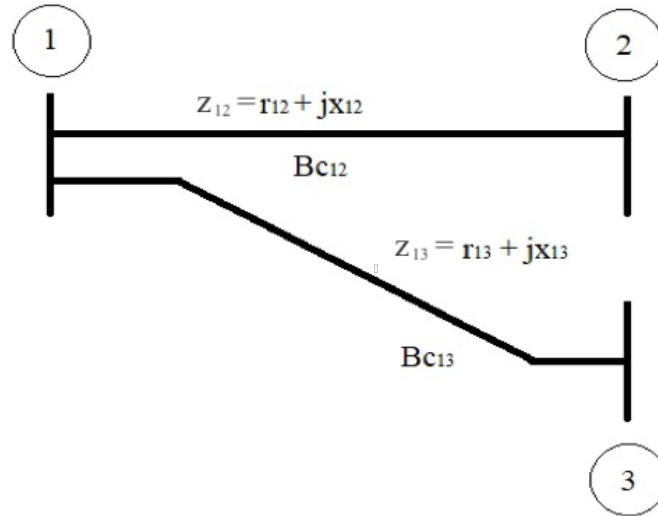


Fonte: Autoria própria.

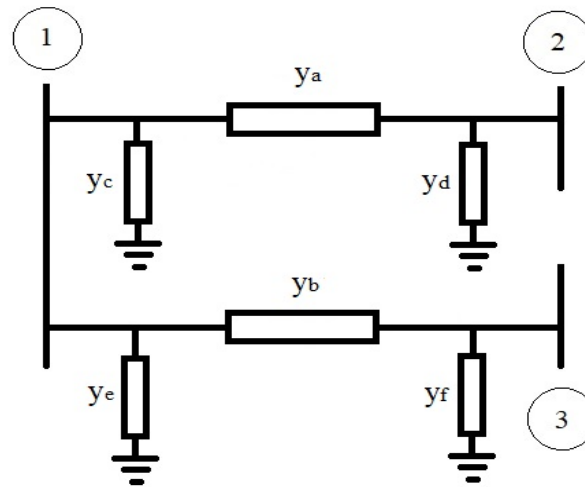
Um sistema de potência de três barras hipotético é mostrado na Figura 5, em que cada linha possui resistência série ( $r$ ), reatância indutiva série ( $x$ ) e elementos *shunt* de susceptâncias capacitivas ( $Bc$ ). A representação desse sistema com os parâmetros de impedâncias e susceptâncias totais das linhas é mostrada na Figura 6.



Figura 5 – Exemplo de sistema elétrico de potência de três barras



Fonte: Autoria própria.

Figura 6 – Modelo  $\pi$ -equivalente de um sistema elétrico de potência de três barras

Fonte: Autoria própria.

As admitâncias série  $y_a$  e  $y_b$  (vide Figura 6) são definidas pelas Equações 2.2 e 2.3, sendo  $r_{12}$  e  $r_{13}$  as resistências série entre as respectivas linhas e  $x_{12}$  e  $x_{13}$  as reatâncias indutivas série das respectivas linhas. Já as admitâncias *shunt* podem ser definidas pelas Equações 2.4 e 2.5, em que  $B_{c12}$  e  $B_{c13}$  são as susceptâncias capacitivas das respectivas linhas.

$$y_a = \frac{1}{r_{12} + j \cdot x_{12}} = \frac{1}{z_{12}} \quad (2.2)$$

$$y_b = \frac{1}{r_{13} + j \cdot x_{13}} = \frac{1}{z_{13}} \quad (2.3)$$

$$y_c = y_d = j \frac{B_{c12}}{2} \quad (2.4)$$

$$y_e = y_f = j \frac{B_{c13}}{2} \quad (2.5)$$

Após feita a modelagem do sistema elétrico utilizando os circuitos equivalentes de cada elemento, deve-se construir a matriz de admitâncias da rede elétrica em questão. A elaboração da matriz de admitâncias é essencial para o cálculo do Fluxo de Potência, pois relaciona as correntes elétricas injetadas no sistema com as tensões elétricas nodais, como já exposto pela Equação 2.1. Ao invés de considerar os valores de resistências, reatância capacitiva e indutiva, transforma-se tais valores em admitâncias e susceptâncias, como mostrado acima.

Após encontrados os valores de susceptâncias e admitâncias dos elementos do sistema, calcula-se a matriz de admitância ( $\mathbf{Y}_{barra}$ ), como mostrado na Equação 2.6.

$$\mathbf{Y}_{barra} = \begin{bmatrix} Y_{11} & Y_{12} & Y_{13} \\ Y_{21} & Y_{22} & Y_{23} \\ Y_{31} & Y_{32} & Y_{33} \end{bmatrix} = \begin{bmatrix} y_a + y_b + y_c + y_e & -y_a & -y_b \\ -y_a & y_a + y_d & 0 \\ -y_b & 0 & y_b + y_f \end{bmatrix} \quad (2.6)$$

Para se obter a matriz  $\mathbf{Y}_{barra}$ , considera-se o número de barras do sistema. Para um sistema de “n” barras, tem-se uma matriz  $n \times n$ . Na construção da diagonal principal da matriz  $\mathbf{Y}_{barra}$ , inserem-se os valores da soma das susceptâncias e admitâncias conectadas a cada barra, como mostrado na Equação 2.7. Por exemplo, no elemento  $\dot{Y}_{ii}$  (admitância própria do nó  $i$ ) inserem-se todos os valores de susceptância capacitiva, indutiva e admitância conectados à barra  $i$  (Equação 2.7). Os demais elementos da matriz representam a admitância entre duas barras. No elemento  $\dot{Y}_{ij}$  (admitância mútua entre os nós  $i$  e  $j$ ) coloca-se o valor da admitância série entre as barras  $i$  e  $j$ , mas com sinal negativo, conforme mostra a Equação 2.8.

$$\dot{Y}_{ii} = \sum_{j=1}^n \dot{y}_{ij} \quad (2.7)$$

$$\dot{Y}_{ij} = -\dot{y}_{ij} \quad (2.8)$$

Sabendo-se que a matriz de admitâncias  $\mathbf{Y}_{\text{barra}}$  é formada por números complexos, pode-se escrever os elementos da matriz  $\mathbf{Y}_{\text{barra}}$  em relação às suas partes reais e imaginárias (Equação 2.9).

$$\dot{Y}_{ij} = G_{ij} + jB_{ij} \quad (2.9)$$

Pode-se calcular a injeção de corrente em determinada barra do sistema pela Equação 2.10. No cálculo podem ser utilizados os valores de potência base do sistema ( $\dot{S}$ ) e a tensão de determinada barra ( $\dot{V}$ ), ou a somatória da multiplicação da tensão na barra pela admitância.

$$\dot{I}_i = \frac{\dot{S}_i^*}{\dot{V}_i^*} = \sum_{j=1}^n \dot{Y}_{ij} \dot{V}_j \quad (2.10)$$

Na obtenção da injeção líquida de potência complexa utiliza-se a Equação 2.11, na qual multiplica-se a tensão na barra pela somatória da admitância.

$$\dot{S}_i^* = |V_i| \sum_{j=1}^n \dot{Y}_{ij} (|V_j| \angle \theta_{ij}) \quad (2.11)$$

Sendo:

$|V_i|$  e  $|V_j|$ : módulo das tensões nas barras  $i$  e  $j$ ;

$\theta_{ij} = \theta_i - \theta_j$ : diferença entre os ângulos das tensões nas barras  $i$  e  $j$ .

No desenvolvimento das equações do Fluxo de Carga, separam-se as partes real e imaginária para o cálculo da injeção de potência aparente em cada barra. Dessa forma, as injeções de potência ativa e reativa (potência gerada na barra menos a potência consumida na barra), são calculadas usando as Equações 2.12 e 2.13, respectivamente.

$$P_i(V, \theta) = |V_i| \cdot \sum_{j=1}^n (G_{ij} \cdot \cos \theta_{ij} + B_{ij} \cdot \sin \theta_{ij}) \cdot |V_j| \quad (2.12)$$

$$Q_i(V, \theta) = |V_i| \cdot \sum_{j=1}^n (G_{ij} \cdot \text{sen } \theta_{ij} - B_{ij} \cos \theta_{ij}) \cdot |V_j| \quad (2.13)$$

Os métodos de solução do Fluxo de Carga baseiam-se em encontrar os valores de  $V_i$  e  $\theta_i$  de cada barra que tornem os desvios de potência  $\Delta P_i$  e  $\Delta Q_i$  próximos de zero, ou menores que uma tolerância pré especificada. Como os valores de potência ativa gerada ( $P_{gi}$ ), potência ativa consumida ( $P_{di}$ ) e potência reativa consumida ( $Q_{di}$ ) na barra  $i$  são conhecidos, utilizando as Equações 2.12 e 2.13, calculam-se os desvios de potência ativa e reativa,  $\Delta P_i$  e  $\Delta Q_i$ , através das Equações 2.14 e 2.15.

$$\Delta P_i = (P_{gi} - P_{di}) - V_i \sum_{j=1}^n (G_{ij} \cdot \cos \theta_{ij} + B_{ij} \text{sen } \theta_{ij}) \cdot V_j \quad (2.14)$$

$$\Delta Q_i = (Q_{gi} - Q_{di}) - V_i \sum_{j=1}^n (G_{ij} \cdot \text{sen } \theta_{ij} - B_{ij} \cos \theta_{ij}) \cdot V_j \quad (2.15)$$

Sendo:

$P_{gi}$ : potência ativa gerada na barra  $i$  (valor conhecido);

$Q_{gi}$ : potência reativa gerada na barra  $i$  (valor calculado anteriormente);

$P_{di}$ : potência ativa consumida na barra  $i$  (valor conhecido);

$Q_{di}$ : potência reativa consumida na barra  $i$  (valor conhecido).

Após realizados os cálculos de tensões complexas em todas as barras, que determinam o estado da rede, calculam-se os fluxos de potência ativa e reativa entre as barras (Equações 2.16, 2.17, 2.18 e 2.19). O Fluxo de Potência ativa da barra  $i$  para a barra  $j$  ( $P_{ij}$ ) e o Fluxo de Potência reativa da barra  $i$  para a barra  $j$  ( $Q_{ij}$ ) são calculados pelas Equações 2.16 e 2.17. Para valores de fluxos positivos, a potência flui de  $i$  para  $j$  e para negativos, de  $j$  para  $i$ .

$$P_{ij} = \frac{1}{r_{ij}^2 + x_{ij}^2} (r_{ij} |V_i|^2 - r_{ij} |V_i| \cdot |V_j| \cos \theta_{ij} + x_{ij} |V_i| \cdot |V_j| \text{sen } \theta_{ij}) \quad (2.16)$$

$$Q_{ij} = -\frac{|V_i|^2 B_c}{2} + \frac{1}{r_{ij}^2 + x_{ij}^2} (x_{ij}|V_i|^2 - r_{ij}|V_i| \cdot |V_j| \sin \theta_{ij} - x_{ij}|V_i| \cdot |V_j| \cos \theta_{ij}) \quad (2.17)$$

O Fluxo de Potência ativa da barra  $j$  para barra  $i$  é calculada pela Equação 2.18 e o Fluxo de Potência reativa da barra  $j$  para barra  $i$  é calculada pela Equação 2.19.

$$P_{ji} = \frac{1}{r_{ij}^2 + x_{ij}^2} (r_{ij}|V_j|^2 - r_{ij}|V_i| \cdot |V_j| \cos \theta_{ij} - x_{ij}|V_i| \cdot |V_j| \sin \theta_{ij}) \quad (2.18)$$

$$Q_{ji} = -\frac{|V_j|^2 B_c}{2} + \frac{1}{r_{ij}^2 + x_{ij}^2} (r_{ij}|V_j|^2 - r_{ij}|V_i| \cdot |V_j| \sin \theta_{ij} - x_{ij}|V_i| \cdot |V_j| \cos \theta_{ij}) \quad (2.19)$$

### 2.3. MÉTODOS DE SOLUÇÃO DO FLUXO DE POTÊNCIA

Existem vários métodos para solução do Fluxo de Carga. Os métodos matemáticos de Gauss-Seidel, Desacoplado, Fluxo de Potência Linearizado e Newton-Raphson são os mais conhecidos para determinação do estado de uma rede elétrica (MONTICELLI, 1983).

O Método de Gauss-Seidel é um método iterativo de solução das equações do Fluxo de Carga, e pode ser considerado como uma melhoria do Método de Gauss.

O Método de Gauss consiste na solução do conjunto de equações advindas da Equação 2.10, como mostrado a seguir, na Equação 2.20.

$$\frac{\dot{S}_i^*}{\dot{V}_i^*} = \sum_{j=1} \dot{Y}_{ij} \cdot \dot{V}_j = \sum_{j=1}^{i-1} (\dot{Y}_{ij} \dot{V}_i + \dot{Y}_{ii} \dot{V}_i) + \sum_{j=i+1}^n \dot{Y}_{ij} \dot{V}_j \quad (2.20)$$

Isolando o valor da tensão na barra  $i$ , obtêm-se a Equação 2.21, que é a solução pelo Método de Gauss para cada barra, a partir de um valor inicial para as tensões.

$$\dot{V}_i = \frac{1}{Y_{ii}} \left\{ \frac{\dot{S}_i^*}{\dot{V}_i^*} - \sum_{j=1}^{i-1} \dot{Y}_{ij} \dot{V}_j - \sum_{j=i+1}^n \dot{Y}_{ij} \dot{V}_j \right\} \quad (2.21)$$

No Método de Gauss-Seidel a substituição dos valores é atualizada a cada equação, acelerando-se o processo de convergência. A Equação 2.21 pode ser reescrita da forma mostrada na Equação 2.22 sendo  $k$  o número da iteração (GOMÉZ-EXPÓSITO et al, 2011).

$$\dot{V}_i^{k+1} = \frac{1}{Y_{ii}} \left\{ \frac{S_i^*}{\dot{V}_i^*} - \sum_{j=1}^{i-1} \dot{Y}_{ij} \dot{V}_j^{k+1} - \sum_{j=i+1}^n \dot{Y}_{ij} \dot{V}_j^k \right\} \quad (2.22)$$

Dessa forma, os valores a serem utilizados são substituídos à medida que são determinados, para executar com maior rapidez o processo iterativo (GOMÉZ-EXPÓSITO et al, 2011).

O excessivo número de iterações requerido no Método de Gauss-Seidel é uma das desvantagens do método. Porém, sua convergência lenta pode ser melhorada usando fatores de aceleração. Uma vantagem desse método é a pequena quantidade de cálculos necessários nas iterações.

Um método matemático de convergência rápida é o Método Desacoplado (BORGES, 2005). Elaborado em 1974, é uma especificação do Método de Newton-Raphson.

No Método Desacoplado considera-se relação entre a magnitude das tensões na barra e o Fluxo de Potência reativa, bem como a sensibilidade entre a potência ativa e o ângulo de tensão da barra (SALGADO, 2016). Uma variação na potência ativa resulta em variação proporcional sobre o ângulo de tensão, e pequena variação sob a magnitude da tensão. Ao passo que uma variação na potência reativa implica em variação proporcional na magnitude da tensão e pequena diferença no ângulo de tensão.

Para resolver o problema principal do sistema elétrico de potência através do Método Desacoplado, divide-o em subproblemas  $P\theta$  e  $QV$ , chamado de desacoplamento entre as partes ativa e reativa (SALGADO, 2016). Melhora-se a convergência do método com a omissão dos elementos capacitores *shunt* e transformadores com comutação sob carga, que afetam as variações em potência reativa. Também, despreza-se os elementos transformadores defasadores, que afetam as variações em potência ativa.

O Método Linearizado, também chamado de Fluxo de Carga CC, é considerado como um modelo de aproximações. É utilizado para níveis elevados de tensão, fornecendo uma visão geral do sistema e desconsiderando a queda de tensão entre as barras (MONTICELLI, 1983).

Considera-se a magnitude da tensão igual à 1,0 pu para a resolução do sistema através do Método Linearizado (SALGADO, 2016). Esse método é realizado com arredondamentos.

Para resolução, impõe-se que são pequenas as aberturas angulares das linhas de transmissão, desconsidera-se a malha  $QV$  e também a resistência série, logo, as perdas de potência ativa nas linhas de transmissão.

O Método de Newton-Raphson é um método iterativo considerado mais rápido comparado aos outros métodos iterativos, pois possui convergência quadrática (SALGADO, 2016). Uma desvantagem desse método é a necessidade de estimativa inicial dos valores de tensão e ângulo ( $V_k^i$  e  $\theta_k^i$ ). Caso sejam má escolhidos, a estimativa errônea pode fazer com que o processo de convergência se mostre oscilatório, ora convergindo, ora divergindo. Usualmente, os valores iniciais escolhidos para tensão e ângulo são 1 e 0, respectivamente

A convergência dos resultados a partir do Método de Newton-Raphson tem menor sensibilidade a modificações, como por exemplo, a escolha de barra de folga (SALGADO, 2016). Sendo esse um método de alta confiabilidade, foi o escolhido a ser estudado neste trabalho e será abordado com maior detalhamento no próximo capítulo.

## 2.4. PRINCIPAIS LIMITES A SEREM CONSIDERADOS NO FLUXO DE CARGA

Na geração de energia elétrica, existem valores limitantes à operação das unidades geradoras. Da mesma forma, também existem fatores de capacidade a serem considerados nos projetos das linhas de transmissão e equipamentos de subestação, cujos valores não devem ser excedidos em sua operação normal e são monitorados pelo operador quando em serviço (MILLER, 1984).

Dois exemplos de variáveis a serem monitoradas são a elevação de temperatura e o fator de potência do equipamento (MILLER, 1984). A temperatura é limitante de carregamento em todos os equipamentos elétricos. O fator de potência envolve a análise das potências ativa e reativa, para estudo da capacidade total das unidades geradoras.

Se uma unidade está suprindo potência reativa além do limite preestabelecido, sua capacidade em MVA poderá ser excedida, mesmo que a potência ativa da carga esteja abaixo do seu valor nominal. Quando o gerador está operando com o fator de potência adiantado, a quem do seu limite de operação, pode ocorrer o aquecimento das lâminas de sua armadura.

Segundo Miller (1984), a capacidade das linhas de transmissão é determinada pelo tipo e bitola do condutor, bem como pelo comprimento da linha. O regime térmico, o tipo e a bitola do condutor são importantes fatores limitantes das linhas de transmissão curtas, não sendo necessário, para essas, considerar o defasamento angular, visto que este não causa problemas de estabilidade. Já em linhas longas, o limite de estabilidade é um fator limitante, sendo atingido antes do limite máximo de corrente do condutor.

O limite de tensão em uma linha de transmissão é determinado segundo a distribuidora de energia elétrica, dependendo da tensão nominal de operação da linha. Segundo o Módulo 8 dos Procedimentos de Distribuição de Energia Elétrica no Sistema Elétrico Nacional (PRODIST):

“A tensão a ser contratada nos pontos de conexão pelos acessantes atendidos em tensão nominal de operação superior a 1 kV deve situar-se entre 95% (noventa e cinco por cento) e 105% (cento e cinco por cento) da tensão nominal de operação do sistema no ponto de conexão e, ainda, coincidir com a tensão nominal de um dos terminais de derivação previamente exigido ou recomendado para o transformador da unidade consumidora (ANEEL, 2017, pag. 7).”

As máquinas síncronas podem funcionar como geradores ao entregar potência ativa, como motores ao absorver potência ativa, ou como compensadores se sua potência ativa tender

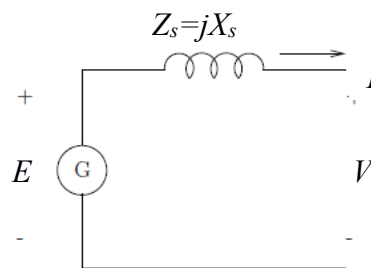


a zero (SALGADO, 2016). A operação das máquinas síncronas pode ser sobreexcitada se fornecer potência reativa ou subexcitada se absorver potência reativa. O suprimento de potência reativa da máquina síncrona é obtido através de sua relação de curto-circuito. A curva de capacidade da máquina é o que determina os limites de absorção e geração de potência ativa e reativa.

#### 2.4.1. Fornecimento de potência reativa pelo gerador

O modelo clássico do gerador síncrono é mostrado na Figura 7, para o qual se ignora a resistência de armadura.

Figura 7 – Circuito representativo do gerador síncrono



Fonte: Salgado (2016).

As equações de potência ativa e reativa do gerador síncrono são representadas pelas Equações 2.23 e 2.24, sendo que  $E$  representa a tensão de excitação do gerador,  $V$  representa a tensão de terminal e  $\delta$  é representado pelo ângulo interno de carga da máquina síncrona.

$$P_g = \frac{|\dot{V}| |\dot{E}|}{X_s} \sin \delta \quad (2.23)$$

$$Q_g = \frac{|\dot{V}|}{X_s} (|\dot{E}| \cos \delta - |\dot{V}|) \quad (2.24)$$

Conforme o gerador entrega potência reativa ( $Q > 0$ ), este se comportará como um capacitor, seu fator de potência ( $\cos \phi$ ) atuará em atraso, sua corrente de campo ( $I$ ) será proporcionalmente maior. O gerador trabalhará sobreexcitado e com isso sua tensão de

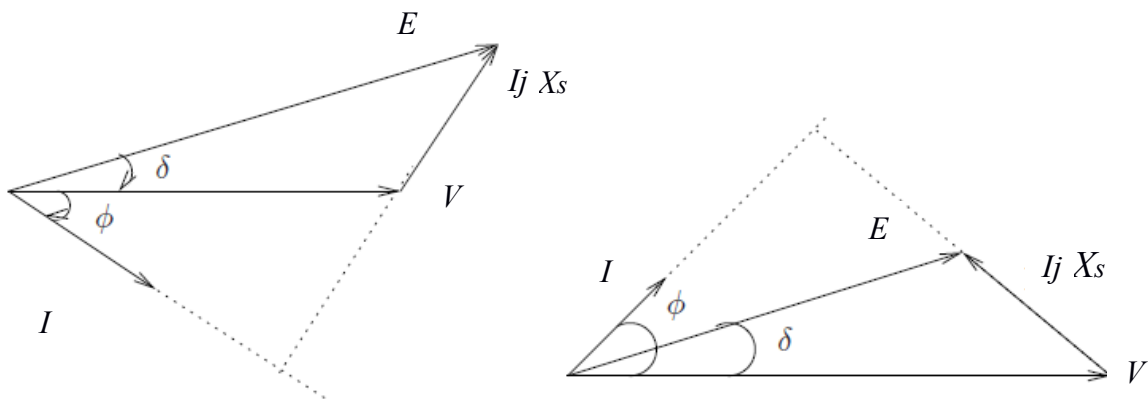
excitação ( $E$ ) será maior do que a tensão de terminal ( $V$ ). O diagrama fasorial para o gerador síncrono sobreexcitado está apresentado na Figura 8(a).

Por outro lado, quando o gerador absorve potência reativa ( $Q < 0$ ), se comportará como um indutor, seu fator de potência ( $\phi$ ) atuará em avanço e sua corrente de campo ( $I$ ) será proporcionalmente menor. O gerador trabalhará subexcitado e com isso sua tensão de excitação ( $E$ ) será menor do que a tensão de terminal ( $V$ ), como mostrado no diagrama fasorial apontado na Figura 8(b).

Figura 8 – Diagramas fasoriais

(a) para o gerador síncrono sobreexcitado

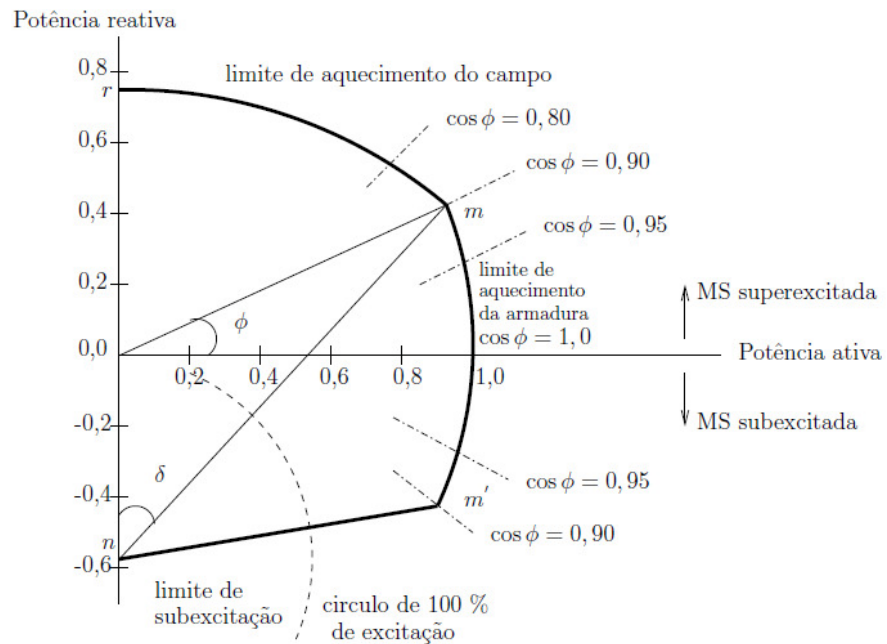
(b) para o gerador síncrono subexcitado



Fonte: Salgado (2016).

O limite de potência reativa a ser gerada, ou consumida, pelo gerador depende do limite de aquecimento do enrolamento de campo. No caso da operação subexcitada, se a corrente de campo for muito baixa, o gerador pode não ter fem ( $E$ ) suficiente para gerar tensão nominal em seus terminais. Portanto, existe um valor limite de potência reativa a ser gerada pela máquina, bem como um valor limite de potência que pode ser absorvida (definido como limite mínimo de reativo). Esses valores são definidos em uma curva de operação da máquina, denominada de curva de capacidade. Para exemplificar, é mostrada na Figura 9 a curva de capacidade de um gerador síncrono. Para que se opere de forma segura, os limites das máquinas síncronas devem ser respeitados.

Figura 9 – Exemplo de curva de capacidade de um gerador



Fonte: Salgado (2016).

## 2.5. CONCLUSÃO

Neste capítulo foi abordada a modelagem do sistema elétrico de potência e como se formula o problema do Fluxo de Carga. Do estudo das referências bibliográficas, foram observados diversos métodos de resolução do Fluxo de Potência, cada um com suas vantagens e desvantagens. Dependendo do sistema a ser analisado e da precisão requerida nos resultados a serem obtidos, escolhe-se o método de solução mais adequado. Estudou-se o método de Newton-Raphson, bem como notou-se as vantagens dos métodos de Gauss-Seidel, Desacoplado e Fluxo de Potência Linearizado. Após análise, considerou-se o Método de Newton-Raphson o mais adequado para abordar nesse trabalho. Esse método escolhido será elucidado na seção subsequente, sendo também expostos exemplos de cálculos considerando sistemas teste.

### 3. MÉTODO DE NEWTON-RAPHSON

#### 3.1. INTRODUÇÃO

A finalidade do método de Newton-Raphson é encontrar a solução de um conjunto de funções do tipo  $g(x) = 0$ . Inicialmente é realizada uma aproximação numérica e na sequência, por meio de derivadas, é calculada a equação de um sub-espço no ponto estimado com o intuito de encontrar a maior aproximação para as raízes das funções.

Para solucionar o problema do Fluxo de Carga de um SEP este trabalho tomará como base o método de Newton-Raphson, que se fundamenta na realização de sucessivas iterações sobre as equações envolvidas, a partir de um suposto valor inicial. No âmbito do trabalho aqui exposto, o valor inicial a se considerar são as tensões complexas nas barras dos sistemas. Em outras palavras, dada a função para qual se deseja encontrar uma raiz aproximada por um erro, escolhe-se um valor inicial de tensão complexa, aplicando o método de iterações repetidamente, obtendo assim uma sequência de pontos a qual convergirá a uma raiz da função.

#### 3.2. FORMULAÇÃO DO MÉTODO DE NEWTON-RAPHSON APLICADO AO FLUXO DE CARGA

Como descrito no Capítulo 2, para o problema de Fluxo de Carga são definidos três tipos de barras:

- $PQ$ , onde são fornecidos  $\mathbf{P}$  e  $\mathbf{Q}$ , e calculados os valores  $\mathbf{V}$  e  $\boldsymbol{\theta}$ ;
- $PV$ , onde são fornecidos  $\mathbf{P}$  e  $\mathbf{V}$ , e calculados os valores  $\mathbf{Q}$  e  $\boldsymbol{\theta}$ ;
- $V\boldsymbol{\theta}$ , onde são fornecidos  $\mathbf{V}$  e  $\boldsymbol{\theta}$ , e calculados os valores  $\mathbf{P}$  e  $\mathbf{Q}$ .

Ao final do cálculo do Fluxo de Carga, são encontrados os valores de duas incógnitas em cada barra, dependendo do tipo de barra:

- $\mathbf{V}$  e  $\boldsymbol{\theta}$  (valores encontrados para as barras de carga ( $PQ$ ) com a obtenção do estado da rede);

- $\theta$  é obtido também para barras do tipo *PV* (unidades geradoras);
- $Q$  é obtido para todas as barras de geração, incluindo a barra de referência (barra  $V\theta$ );
- $P$  é calculado para a barra  $V\theta$ .

O subconjunto das equações  $g(x)$  do problema de Fluxo de Carga representado de uma forma genérica por  $g(x)$ , onde  $x$  são as incógnitas e  $g(x)$  o vetor das equações não lineares (MONTICELLI e GARCIA, 2003):

$$x = \begin{cases} \theta_k & ; k \in [\text{barras PV}] \\ \theta_k & ; k \in [\text{barras PQ}] \\ V_k & ; k \in [\text{barras PQ}] \end{cases}$$

e

$$g(x) = \begin{cases} P_k^{esp} - P_k(x) & ; k \in [\text{barras PV}] \\ P_k^{esp} - P_k(x) & ; k \in [\text{barras PQ}] \\ Q_k^{esp} - Q_k(x) & ; k \in [\text{barras PQ}] \end{cases}$$

Onde *esp* representa *especificado*. Expandindo  $g(x)$  em série de Taylor, em torno de um ponto inicial  $x^i$  e desprezando termos de ordem superior, tem-se:

$$g(x) \approx g(x^i) + \left[ \frac{\partial g(x^i)}{\partial x} \right] \Delta x^i \quad (3.1)$$

Forçando  $g(x) = 0$ , obtém-se a correção  $\Delta x^i$ , resolvendo-se o sistema linear:

$$\left[ \frac{\partial g(x^i)}{\partial x} \right] \Delta x^i = -g(x^i) \quad (3.2)$$

Considerando o vetor de estados:

$$x = \begin{bmatrix} \theta \\ V \end{bmatrix} \quad \begin{array}{l} \rightarrow \text{ângulos} \\ \rightarrow \text{magnitude das tensões} \end{array}$$

Reescrevendo os *mismatches* (resíduos ou desvios de potência) como mostrado nas Equações 2.14 e 2.15.

$$\Delta Q(\theta^i, V^i) = P_k^{esp} - Q_k(\theta^i, V^i) \quad (3.4)$$

$$\Delta P(\theta^i, V^i) = P_k^{esp} - P_k(\theta^i, V^i) \quad (3.3)$$

A Equação 3.1 pode ser escrita como:

$$\begin{bmatrix} H & N \\ M & L \end{bmatrix} \begin{bmatrix} \Delta\theta^i \\ \Delta V^i \end{bmatrix} = \begin{bmatrix} \Delta P(\theta^i, V^i) \\ \Delta Q(\theta^i, V^i) \end{bmatrix} \quad (3.5)$$

Onde  $H = \frac{\partial P}{\partial \theta}$ ,  $L = \frac{\partial Q}{\partial V}$ ,  $M = \frac{\partial Q}{\partial \theta}$  e  $N = \frac{\partial P}{\partial V}$ . É válido salientar que as derivadas parciais dos valores especificados  $P^{esp}$  e  $Q^{esp}$  são nulas.

O sistema de equações 3.5 é esparso. As submatrizes  $H$ ,  $L$ ,  $M$  e  $N$  têm elementos não nulos apenas nas posições correspondentes às ligações entre barras, além das diagonais (as submatrizes têm estrutura semelhante à da matriz admitância nodal).

As componentes das submatrizes jacobianas  $H$ ,  $L$ ,  $M$  e  $N$  são dadas por:

$$H = \begin{cases} H_{km} = \frac{\partial P_k}{\partial \theta_m} = V_k V_m (G_{km} \text{sen } \theta_{km} - B_{km} \text{cos } \theta_{km}) \\ H_{kk} = \frac{\partial P_k}{\partial \theta_k} = -V_k^2 B_{kk} - V_k \sum_{m \in k} V_m (G_{km} \text{sen } \theta_{km} - B_{km} \text{cos } \theta_{km}) \end{cases} \quad (3.6)$$

$$N = \begin{cases} N_{km} = \frac{\partial P_k}{\partial V_m} = V_k (G_{km} \text{cos } \theta_{km} + B_{km} \text{sen } \theta_{km}) \\ N_{kk} = \frac{\partial P_k}{\partial V_k} = V_k G_{kk} + \sum_{m \in k} V_m (G_{km} \text{cos } \theta_{km} + B_{km} \text{sen } \theta_{km}) \end{cases} \quad (3.7)$$

$$M = \begin{cases} M_{km} = \frac{\partial Q_k}{\partial \theta_m} = -V_k V_m (G_{km} \text{cos } \theta_{km} + B_{km} \text{sen } \theta_{km}) \\ M_{kk} = \frac{\partial Q_k}{\partial \theta_k} = -V_k^2 G_{kk} + V_k \sum_{m \in k} V_m (G_{km} \text{cos } \theta_{km} + B_{km} \text{sen } \theta_{km}) \end{cases} \quad (3.8)$$

$$L = \begin{cases} H_{km} = \frac{\partial Q_k}{\partial V_m} = V_k (G_{km} \text{sen } \theta_{km} - B_{km} \text{cos } \theta_{km}) \\ H_{kk} = \frac{\partial Q_k}{\partial V_k} = -V_k B_{kk} + \sum_{m \in k} V_m (G_{km} \text{sen } \theta_{km} - B_{km} \text{cos } \theta_{km}) \end{cases} \quad (3.9)$$

Sendo  $k = \{k \in N : 1 \leq k \leq \text{Número de barras}\}$

A partir das expressões 3.6 a 3.13, pode-se concluir que, se  $Y_{km} = G_{km} + jB_{km}$  for nulo, então os elementos  $H_{km}$ ,  $N_{km}$ ,  $M_{km}$  e  $L_{km}$  também serão nulos. Isto implica que as matrizes H, N, M e L têm as mesmas características de esparsidade que a matriz Y.

### 3.3. ALGORITMO DE NEWTON-RAPHSON PARA O FLUXO DE CARGA

Os passos executados para a solução do método de Newton-Raphson são os apresentados a seguir (MONTICELLI e GARCIA, 2003; CASTRO, 2017):

**1º passo:** Dado o contador inicial  $i = 0$  de iterações, a princípio são escolhidos valores iniciais de  $V_k^i$  e  $\theta_k^i$ .

**2º passo:** Para as barras tipo PV e PQ, calcula-se as injeções de potências ativas  $P_k(V^i, \theta^i)$  e para as tipo PQ, calcula-se  $Q_k(V^i, \theta^i)$ . As injeções de potência se dão pela aplicação das Leis de Kirchhoff à todas as “n” barras da rede elétrica, de acordo com as Equações 3.14 e 3.15 (similares às Equações 2.12 e 2.13, no entanto foram utilizados outros índices):

$$P_k(V, \theta) = V_k \sum_{m=1}^n V_m [G_{km} \cos(\theta_{km}) + B_{km} \text{sen}(\theta_{km})] \quad (3.14)$$

$$Q_k(V, \theta) = V_k \sum_{m=1}^n V_m [G_{km} \text{sen}(\theta_{km}) - B_{km} \text{cos}(\theta_{km})] \quad (3.15)$$

Lembrando que, para calcular as injeções de potências, é preciso encontrar previamente a matriz de admitâncias Y (como descrito no Capítulo 2), que contém os valores de condutância ( $G_{km}$ ) e susceptância ( $B_{km}$ ) a serem aplicados nas Equações 3.14 e 3.15. Também, é preciso aplicar os valores de tensão complexa referente à atual iteração ( $i$ ) para o cálculo das injeções de potência nos barramentos. Deste modo, na primeira iteração, arbitra-se valores de tensão complexa nos barramentos. No entanto, para as iterações subsequentes, utilizam-se os valores de tensões complexas encontrados no método de Newton-Raphson, referente à iteração anterior.

**3º passo:** Testar a convergência. Com os valores de  $P_k(V, \theta)$  e  $Q_k(V, \theta)$ , determina-se o vetor dos resíduos de potência  $\Delta P_k^i$  e  $\Delta Q_k^i$ , (que representa  $g(x^i)$ ) como mostrado nas Equações 3.3 e 3.4.

$$g(x^i) = \begin{cases} \Delta P = P^{esp} - P_k(V, \theta) & (3.16) \\ \Delta Q = Q^{esp} - Q_k(V, \theta) & (3.17) \end{cases}$$

A resolução das Equações 3.16 e 3.17, forma o vetor dos resíduos  $g(x^i)$  que satisfaz a solução do Fluxo de Potência quando  $g(x^i)=0$ . Ou seja, quando os valores de injeções de potência ativa e reativa especificados ( $P^{esp}$  e  $Q^{esp}$ ) nas barras forem iguais aos valores das injeções de potência ativa e reativa calculados pelas Equações 3.14 e 3.15. Na prática, as possibilidades dessa igualdade ser igual a zero para a primeira iteração são mínimas. Tendo isso em vista, o operador do sistema deve arbitrar um nível de tolerância ( $\epsilon$ ) próximo de zero para o vetor dos resíduos, que não necessariamente precisa ser o mesmo para os resíduos de potência ativa e reativa ( $\Delta P_k$  e  $\Delta Q_k$ ). Sendo assim, conforme são efetuadas as iterações, o vetor dos resíduos  $g(x^i)$  deve tender ao nível de tolerância ( $\epsilon$ ), até que se torne um valor menor, acarretando a convergência e o encerramento do processo.

Em outras palavras, a obtenção da convergência se dá comparando o valor calculado dos resíduos com a tolerância especificada ( $\epsilon$ ), conforme mostrado nas Equações 3.18 e 3.19.

$$\max |\Delta P_k| \leq \epsilon_p \quad (3.18)$$

e

$$\max |\Delta Q_k| \leq \epsilon_q \quad (3.19)$$

Portanto, caso as comparações sejam verdadeiras, o processo iterativo convergiu para a solução de  $V_k^i$  e  $\theta_k^i$ . Caso contrário, é preciso continuar o processo iterativo, seguindo o próximo passo.

**4º passo:** Consiste em montar e solucionar a matriz Jacobiana, que é determinada pela derivada das equações de injeção de potência nas barras da rede elétrica em relação à tensão e ao ângulo da atual iteração.

$$J(x^i) = \begin{bmatrix} \frac{\partial P_k}{\partial \theta_k} & \frac{\partial P_k}{\partial V_k} \\ \frac{\partial Q_k}{\partial \theta_k} & \frac{\partial Q_k}{\partial V_k} \end{bmatrix} \quad (3.20)$$



Obtidos  $J(x^i)$  e o vetor de resíduo de potência  $g(x^i)$ , torna-se possível determinar o vetor correção de estado de tensão complexa ( $\Delta x^i$ ) pela Equação 3.5.

De forma geral, para as barras tipo  $PV$  e  $PQ$ , calcula-se  $\Delta\theta^i$  e para as barras tipo  $PQ$ , calcula-se  $\Delta V^i$ .

**5º passo:** Trata-se da última etapa, cujo objetivo é calcular os novos valores de  $V_k^{i+1}$  e  $\theta_k^{i+1}$  e partir para a nova iteração ( $i + 1$ ).

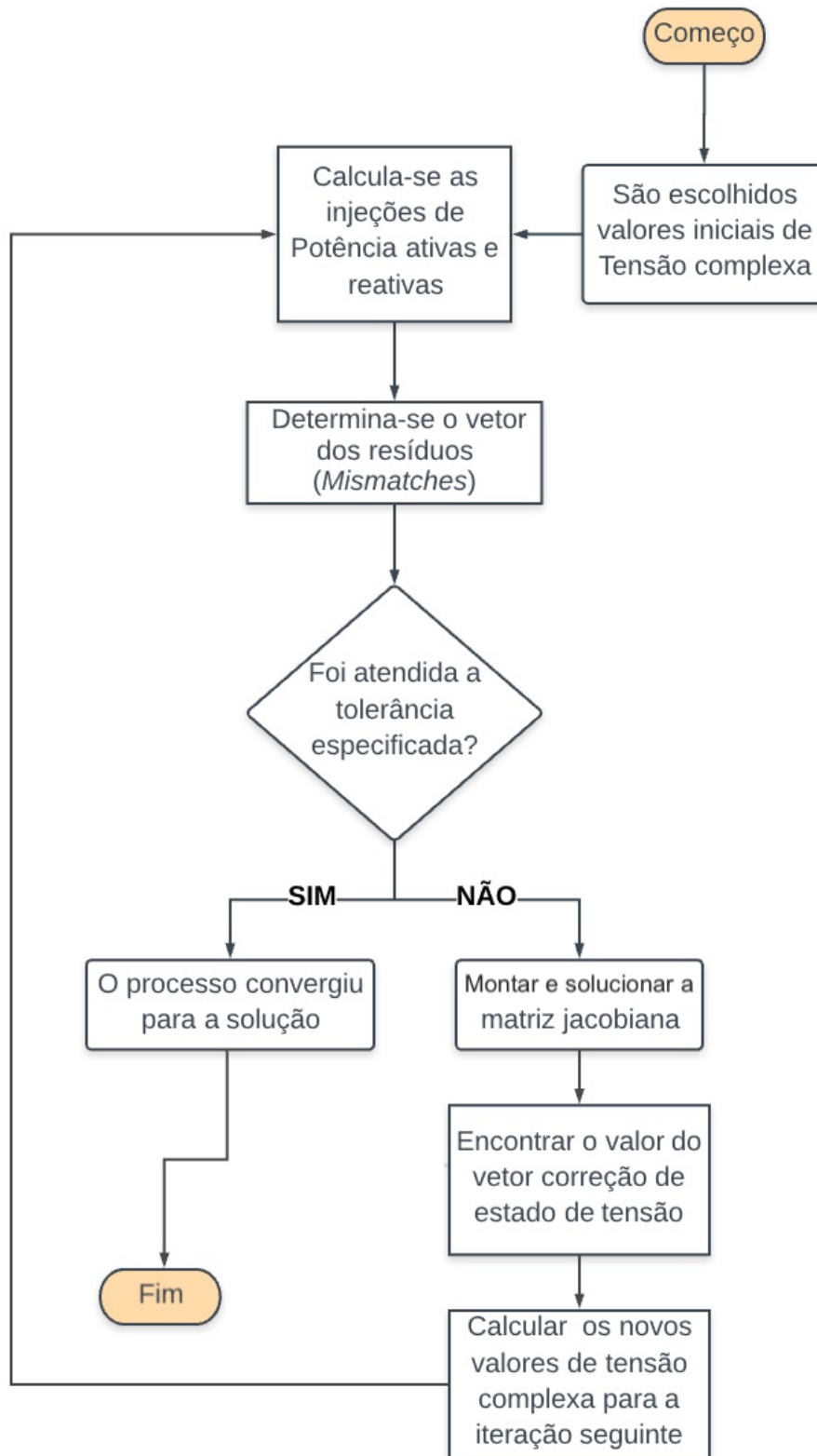
As Equações 3.21 e 3.22, apresentadas a seguir, representam a obtenção do estado atual da rede submetida ao cálculo do Fluxo de Carga:

$$\begin{cases} \theta^{i+1} = \theta^i + \Delta\theta^i & (3.21) \\ V^{i+1} = V^i + \Delta V^i & (3.22) \end{cases}$$

Dessa forma, com o valores de tensões complexas  $V_k^{i+1}$  e  $\theta_k^{i+1}$  da nova iteração ( $i + 1$ ), torna-se possível novamente o cálculo de injeção de potência e a obtenção do teste dos resíduos para a iteração seguinte ( $i + 1$ ). Neste estágio deve-se verificar se os resíduos de potência são menores do que a tolerância especificada pelo operador. Caso seja menor, o processo se encerra, sendo obtido o estado da rede. Caso contrário, continua-se o processo iterativo até que sejam atendidos os requisitos de tolerância (CASTRO, 2017).

O fluxograma apresentado na Figura 10 mostra os passos para a resolução do método de Newton-Raphson de forma simplificada.

Figura 10 – Fluxograma resumido para a resolução do Fluxo de Carga pelo método de Newton-Raphson



Fonte: Autoria própria.

### 3.4. RESOLUÇÃO DE EXERCÍCIO DE FLUXO DE CARGA E APLICAÇÃO DO MÉTODO DE NEWTON-RAPHSON

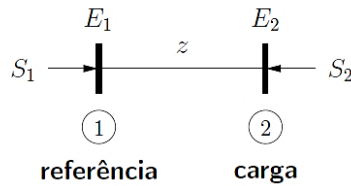
Visando o estudo do método de Newton-Raphson, foi resolvido o problema de Fluxo de Carga para um sistema-teste de 2 barras (CASTRO, 2017). Para esse sistema os cálculos foram efetuados sem auxílio de ferramentas computacionais, realizando-se manualmente cada passo do método de Newton-Raphson para encontrar os resultados. Para a validação dos resultados obtidos, também foi utilizado o programa computacional SIMPOT (BUENO; TAVARES; MAROCHI, 2012).

O programa SIMPOT foi desenvolvido por alunos da UTFPR como trabalho de conclusão de curso do curso de Engenharia Industrial Elétrica, ênfase Eletrotécnica (BUENO; TAVARES; MAROCHI, 2012). A proposta deste trabalho foi elaborar um software didático com uma interface gráfica amigável ao usuário, para ser utilizado por alunos que ainda estariam aprendendo os conceitos aplicados aos problemas de Fluxo de Carga e ao método iterativo de Newton-Raphson. O SIMPOT foi desenvolvido em uma plataforma de programação de licença pública geral, o JAVA, possibilitando a sua distribuição gratuita para os alunos, porém atualmente não é mais compatível com os sistemas computacionais utilizados, tais como WINDOWS 10.

#### **3.4.1. Resolução de um exemplo numérico de Fluxo de Carga para um sistema de duas Barras**

O sistema a resolvido utilizando o método de Newton-Raphson foi o sistema de 2 barras mostrado na Figura 11, apresentado por Castro (2017). O Fluxo de Potência para o sistema apresentado foi solucionado matematicamente como apresentado na sequência (considerando  $\varepsilon = 0,0001$ ). Embora seja um sistema de apenas duas barras, em que não havia barramento *PV*, apenas a barra de referência e uma barra de carga, a resolução do exercício envolve uma quantidade considerável de cálculos, justamente por se tratar de um processo iterativo. A resolução do Fluxo de Carga para esse exercício foi de grande importância para a fixação dos conceitos iniciais do método de Newton-Raphson.

Figura 11 – Sistema de duas barras



Fonte: Castro (2017, p. 22).

Antes de realizar os passos para o cálculo do Fluxo de Carga pelo método de Newton-Raphson, deve-se fazer os cálculos preliminares da matriz de admitâncias  $\mathbf{Y}_{barra}$ . Tem-se os dados do problema:

Dados

$E_1 = 1,0112 \angle 0^\circ \text{ pu}$ $z = 0,01 + j 0,05 \text{ pu}$ $S_2 = -1,0 \angle 0^\circ \text{ pu}$
--

Dessa forma, dadas as equações apresentadas no Capítulo 2, Seção 2.2, tem-se a matriz de admitâncias para o sistema de duas barras.

$$\mathbf{Y}_{barra} = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix}$$

$$\mathbf{Y}_{barra} = \begin{bmatrix} 3,8462 + j19,2308 & -3,8462 + j19,2308 \\ -3,8462 + j19,2308 & 3,8462 - j19,2308 \end{bmatrix}$$

A seguir apresentam-se os passos para a resolução pelo método de Newton-Raphson.

### 1ª iteração:

**1º passo:** deve-se escolher valores iniciais dos ângulos das tensões nas barras  $PV$  e  $PQ$  e somente os valores iniciais das tensões nas barras  $PQ$ . Nesse caso, analisando os dados do sistema, temos o valor de  $E_1$ , e então, os valores de  $V_1^{i=0}$  e  $\theta_1^{i=0}$ . Os valores de  $V_2^{i=0}$  e  $\theta_2^{i=0}$  foram arbitrados como módulo unitário e ângulo inicial nulo.

$$V_1^{i=0} = 1,0112 \text{ pu}$$

$$\theta_1^{i=0} = 0 \text{ rad}$$

$$V_2^{i=0} = 1,0 \text{ pu}$$

$$\theta_2^{i=0} = 0 \text{ rad}$$

**2º passo:** Para as barras tipo PV e PQ, calcula-se as injeções de potências ativa  $P_k(V^i, \theta^i)$  e para as tipo PQ, calcula-se  $Q_k(V^i, \theta^i)$ , de acordo com as Equações 3.14 e 3.15.

$$P_2(V, \theta) = V_2 * \{V_1[G_{21} \cos(\theta_{21}) + B_{21} \text{sen}(\theta_{21})] + V_2[G_{22} \cos(\theta_{22}) + B_{22} \text{sen}(\theta_{22})]\}$$

$$P_2(V, \theta) = 1 * \{1,0112 * [-3,8462 \cos(0) + 19,2308 \text{sen}(0)] + 1 * [3,8462 \cos(0) - 19,2308 \text{sen}(0)]\}$$

$$P_2(V, \theta)^{i=0} = 1,0112 * [-3,8462 * \cos(0)] + 3,8462$$

$$P_2(V, \theta)^{i=0} = -0,04307 \text{ pu}$$

$$Q_2(V, \theta) = V_2 * \{V_1[G_{21} \text{sen}(\theta_{21}) - B_{21} \cos(\theta_{21})] + V_2[G_{22} \text{sen}(\theta_{22}) - B_{22} \cos(\theta_{22})]\}$$

$$Q_2(V, \theta) = 1 * \{1,0112 * [-3,8462 \text{sen}(0) - 19,2308 \cos(0)] + 1 * [3,8462 \text{sen}(0) - (-19,2308) \cos(0)]\}$$

$$Q_2(V, \theta)^{i=0} = 1,0112 * [-19,2308 \cos(0)] + 19,2308$$

$$Q_2(V, \theta)^{i=0} = -0,21538 \text{ pu}$$

**3º passo:** Determina-se o vetor dos *mismatches*  $\Delta P_k^i$  e  $\Delta Q_k^i$ , que é representado por  $g(x^i)$ , de acordo com as Equações (3.16) e (3.17). Substituindo os valores do exemplo, tem-se:

$$g(x^{i=0}) = \begin{cases} \Delta P_2 = -1 + 0,04307 = -0,95693 \\ \Delta Q_2 = 0 + 0,21538 = 0,21538 \end{cases}$$

É possível verificar que os *mismatches* de potência  $g(x^i)$  são maiores do que tolerância  $\varepsilon = 0,0001$ . Então, o processo não convergiu para a solução, e deve-se continuar os cálculos no passo seguinte.

**4º passo:** Consiste em montar a matriz Jacobiana. Aplicando-se as equações (3.06) a (3.13), tem-se:

$$H = \frac{\partial P_2^{i=0}}{\partial \theta_2} = 19,4462 \text{ pu}$$

$$N = \frac{\partial P_2^{i=0}}{\partial V_2} = 3,8031 \text{ pu}$$

$$M = \frac{\partial Q_2^{i=0}}{\partial \theta_2} = -3,8892 \text{ pu}$$

$$L = \frac{\partial Q_2^{i=0}}{\partial V_2} = 19,0154 \text{ pu}$$

$$J(x^{i=0}) = \begin{bmatrix} 19,4462 & 3,8031 \\ -3,8892 & 19,0154 \end{bmatrix}$$

Calculando a matriz inversa de  $J(x^i)$ :

$$[J(x^i)]^{-1} = \begin{bmatrix} 0,0494 & -0,0099 \\ 0,0101 & 0,0506 \end{bmatrix}$$

Obtidos  $J(x^i)$  e o vetor de resíduo de potência  $g(x^i)$ , torna-se possível determinar o vetor correção de estado de tensão complexa ( $\Delta x^i$ ).

$$\Delta x^{i=0} = [J(x^{i=0})]^{-1} * g(x^{i=0})$$

$$\Delta x^{i=0} = \begin{bmatrix} 0,0494 & -0,0099 \\ 0,0101 & 0,0506 \end{bmatrix} * \begin{bmatrix} -0,95693 \\ 0,21538 \end{bmatrix}$$

$$\Delta x^i = \begin{cases} \Delta \theta_2^i = -0,0494 \\ \Delta V_2^i = 0,0012 \end{cases}$$

**5º passo:** Trata-se da última etapa, cujo objetivo é calcular os novos valores de  $V_k^{i+1}$  e  $\theta_k^{i+1}$ .

$$\begin{cases} \theta_2^{i+1} = \theta_2^i + \Delta \theta_2^i \\ V_2^{i+1} = V_2^i + \Delta V_2^i \end{cases}$$

Valores de tensão complexa que serão utilizados para a próxima iteração:

$$\begin{aligned} \theta_2^{i+1} &= \theta_2^{i=1} = -0,0494 \text{ rad} \\ V_2^{i+1} &= V_2^{i=1} = 1,0012 \text{ pu} \end{aligned}$$

**2ª iteração:**

Seguindo-se os mesmos passos da primeira iteração ( $i=0$ ), de forma resumida será realizada a segunda iteração ( $i=1$ ) com os valores de  $V_2^{i=1}$  e  $\theta_2^{i=1}$ . Sendo assim:

**1º passo:**

$$\begin{aligned}\theta_2^{i=1} &= -0,0494 \text{ rad} \\ V_2^{i=1} &= 1,0012 \text{ pu}\end{aligned}$$

**2º passo:**

$$\begin{aligned}P_2(V, \theta)^{i=1} &= -0,9960 \text{ pu} \\ Q_2(V, \theta)^{i=1} &= 0,0240 \text{ pu}\end{aligned}$$

**3º passo:**

$$g(x^i) = \begin{cases} \Delta P_2 = -1 - 0,9960 \\ \Delta Q_2 = 0 - 0,0240 \end{cases}$$

$$g(x^i) = \begin{cases} \Delta P_2 = -0,0040 \\ \Delta Q_2 = -0,0240 \end{cases}$$

Os *mismatches* ainda são maiores do que a tolerância especificada ( $\varepsilon=0,0001$ ), então, o processo não convergiu para a solução do estado da rede, e é necessário continuar o exemplo, seguindo para o próximo passo:

**4º passo:**

$$\frac{\partial P_2^{i=1}}{\partial \theta_2} = 19,2532 \text{ pu}$$

$$\frac{\partial P_2^{i=1}}{\partial V_2} = 2,85699 \text{ pu}$$

$$\frac{\partial Q_2^{i=1}}{\partial \theta_2} = -4,8503 \text{ pu}$$

$$\frac{\partial Q_2^{i=1}}{\partial V_2} = 19,2773 \text{ pu}$$

$$J(x^{i=1}) = \begin{bmatrix} 19,2532 & 2,85699 \\ -4,8504 & 19,2773 \end{bmatrix}$$

Calculando a matriz inversa de  $J(x^{i=1})$ :

$$[J(x^{i=1})]^{-1} = \begin{bmatrix} 0,0501 & -0,0074 \\ 0,0126 & 0,0500 \end{bmatrix}$$

Bem como na primeira iteração, obtidos  $J(x^i)$  e o vetor de resíduos de potência  $g(x^i)$  pelo 2º passo, torna-se possível determinar o vetor correção de estado de tensão complexa ( $\Delta x^i$ ):

$$\Delta x^{i=1} = [J(x^{i=1})]^{-1} * g(x^{i=1})$$

$$\Delta x^{i=1} = \begin{bmatrix} 0,0501 & -0,0074 \\ 0,0126 & 0,0500 \end{bmatrix} * \begin{bmatrix} -0,0040 \\ -0,0240 \end{bmatrix}$$

$$\Delta x^{i=1} = \begin{cases} \Delta \theta_2^i = -0,0001 \\ \Delta V_2^i = 0,0012 \end{cases}$$

**5º passo:**

$$\begin{cases} \theta_2^{i+1} = \theta_2^i + \Delta \theta_2^i \\ V_2^{i+1} = V_2^i + \Delta V_2^i \end{cases}$$

Da mesma forma, têm-se os valores de tensões complexas que serão utilizados para a próxima iteração:

$$\begin{aligned} \theta_2^{i+1} &= -0,0494 - 0,0001 \\ V_2^{i+1} &= 1,0012 - 0,0012 \text{ pu} \end{aligned}$$

$$\begin{aligned} \theta_2^{i+1} &= \theta_2^{i=2} = -0,0495 \text{ rad} \\ V_2^{i+1} &= V_2^{i=2} = 1 \text{ pu} \end{aligned}$$



**3ª iteração:**

Seguindo os mesmos critérios da segunda iteração ( $i=1$ ), de forma resumida será realizada a terceira iteração ( $i=2$ ) com os valores de  $V_2^{i=2}$  e  $\theta_2^{i=2}$ . Sendo assim:

**1º passo:**

$$\begin{aligned}\theta_2^{i=2} &= -0,0495 \text{ rad} \\ V_2^{i=2} &= 1 \text{ pu}\end{aligned}$$

**2º passo:**

$$\begin{aligned}P_2(V, \theta)^{i=2} &= -1 \text{ pu} \\ Q_2(V, \theta)^{i=1} &= 0 \text{ pu}\end{aligned}$$

**3º passo:**

$$g(x^i) = \begin{cases} \Delta P_2 = -1 - (-1) \\ \Delta Q_2 = 0 - 0 \end{cases}$$

$$g(x^i) = \begin{cases} \Delta P_2 = 0 \\ \Delta Q_2 = 0 \end{cases}$$

Os *mismatches* são menores do que a tolerância especificada ( $\epsilon=0,0001$ ), portanto o processo convergiu para a solução do estado da rede e tem-se o resultado das tensões complexas nos barramentos.

$$\begin{aligned}V_2^{i=2} &= 1 \text{ pu} \\ \theta_2^{i=2} &= -0,0495 \text{ rad} = -2,8361^\circ\end{aligned}$$

A seguir, é possível determinar os valores de  $P_1$  e  $Q_1$ , através do cálculo de  $S_1$ :

$$S_1 = E_1 I_{12}^* = E_1 \left[ \frac{1}{Z} (E_1 - E_2) \right]^* = 1,01 + j0,05 \text{ pu}$$

$$P_1 = 1,01 \text{ pu}$$

$$Q_1 = 0,05 \text{ pu}$$

No ANEXO A apresenta-se o relatório de saída do programa SIMPOT para a solução do Fluxo de Carga do sistema de 2 barras em questão. O uso dessa ferramenta computacional foi fundamental para comparação e validação dos resultados obtidos.

### 3.5. CONCLUSÃO

Ao realizar-se o cálculo do Fluxo de Carga pelo método de Newton-Raphson obtêm-se o estado da rede elétrica (módulo e ângulo das tensões). O estado da rede é utilizado pelo operador do sistema em etapas do planejamento, como o estudo de contingências, avaliação da estabilidade da rede, avaliação de sobrecarga de componentes e solução de problemas de otimização, por exemplo.

Durante o estudo do método de Newton-Raphson, utilizou-se o programa computacional SIMPOT (BUENO; TAVARES; MAROCHI, 2012) simulando sistemas de pequeno porte com esse programa. O SIMPOT fornece os resultados dos *mismatches* e jacobiano de cada iteração. Dessa forma, verificou-se os cálculos realizados a cada passo.

O estudo de Fluxo de Carga pelo método de Newton-Raphson convergiu com poucas iterações para tolerância ( $\epsilon$ ) de  $10^{-4}$ . Apresentou-se, então, como um método de convergência rápida e bastante eficiente.

## 4. ALGORITMO FLOBA

### 4.1. INTRODUÇÃO

Após estudo da formulação do Fluxo de Carga em sistemas elétricos de potência, bem como do Método de Newton-Raphson para solução do estado de uma rede elétrica, obteve-se conhecimento suficiente para elaborar um algoritmo que, além de efetuar os cálculos para um sistema com “n” barras, também considere um fator limitante no Fluxo de Potência de um sistema.

Neste capítulo apresenta-se o algoritmo do FLOBA implementado em MATLAB. Antes deste ser executado, deve-se executar o arquivo com os dados do sistema a ser analisado, no formato de “DadosDoSistema.m”. Neste arquivo os dados de entrada são inseridos pelo usuário, bem como os dados de barra e dados de linha do sistema a ser simulado, respeitando as grandezas indicadas no modelo. O arquivo “FLOBA.m” contém o código que realiza os cálculos do Fluxo de Carga pelo método de Newton-Raphson, e analisa se os limites de reativos das máquinas síncronas foram atingidos. Caso isso ocorra, o FLOBA apresenta opções de solução do Fluxo de Carga para o sistema analisado, considerando a violação do limite de geração ou consumo de reativos.

### 4.2. DADOS DO SISTEMA

Os dados do sistema a ser analisado devem ser inseridos no arquivo modelo “DadosDoSistema.m”. O valor de potência aparente base do sistema, o número de barras do sistema, o número de linhas e a tolerância ou erro do sistema devem ser digitados nesse arquivo, na seção “Entrada de Dados”.

Os dados de barra do sistema devem ser inseridos de acordo com a disposição e as unidades dos termos indicados no modelo, conforme mostrado na Figura 12, com os dados de um sistema de 6 barras (arquivo “DadosDoSistemaSeisBarras.m”, apresentado completamente no APÊNDICE A).

Figura 12 – Matriz de dados de barra do sistema

```

%***** Entrada de Dados *****%
clear all
Sbase = 100;
NUMBAR = 6;      % Número de barras
NUMLIN = 11;    % Número de linhas
e=0.001;        % Tolerância/erro

%***** Matriz de Dados de Barras *****%

% *** OBSERVAÇÕES:
%           - A barra de referência deve ser sempre a barra (Bar):
1
%           - Barra tipo (T) 0: PQ
%                               1: PV
%                               2: Vtheta

%           Bar T Tensão (p.u.) Âng(°) Pg (MW) Qg (Mvar) Pl (MW) Ql (Mvar) Shunt (%) Qgmin (Mvar) Qgmax (Mvar) Pgmax (MW)
DBAR=[ 1 2 1.050 0.0 0.0 0.0 0.0 0.0 0.0 -100 100 200 ;
       2 1 1.050 0.0 50 0.0 0.0 0.0 0.0 -100 70 150 ;
       3 1 1.070 0.0 60 0.0 0.0 0.0 0.0 -100 70 180 ;
       4 0 1.000 0.0 0.0 0.0 70 70 0.0 0.0 0.0 0.0 ;
       5 0 1.000 0.0 0.0 0.0 70 70 0.0 0.0 0.0 0.0 ;
       6 0 1.070 0.0 0.0 0.0 70 70 0.0 0.0 0.0 0.0 ];

```

Fonte: Autoria própria.

Deve-se inserir na matriz de dados de barras “DBAR” o número da barra, o tipo, o valor da tensão em pu, o ângulo em graus, a potência ativa de geração em MW, a potência reativa de geração em Mvar, a potência ativa de carga em MW, a potência reativa de carga em Mvar, o valor de shunt em %, o valor de potência reativa de geração mínima e máxima (limites de reativo) em Mvar e a potência ativa de geração máxima em MW. Deve-se considerar as observações comentadas no algoritmo, admitir a barra de referência como a barra número 1 do sistema obrigatoriamente para o correto funcionamento do FLOBA. Outra consideração é que deve-se indicar o tipo da barra pelo número 0 para barras do tipo *PQ*, número 1 para barra tipo *PV*, e número 2 para barra tipo *Vθ*, sendo os mesmos números para definição do tipo de barra do padrão ANAREDE.

Os dados de linha do sistema devem ser inseridos de acordo com a disposição e a unidade dos termos indicada no modelo (Figura 13). Indicar na matriz “DLIN” as barras que estão conectadas à linha a ser analisada, ou seja, inserir na matriz os números das barras de “origem” e “destino” da linha. Devem ser inseridos, também, o valor das resistências, reatâncias e susceptâncias em pu, nessa ordem.

Figura 13 – Matriz de dados de linha do sistema

```

%***** Matriz de dados de linha *****%
%
%          Origem  Destino  Res(p.u)  Reat(p.u.)  Suscept(p.u.)
DLIN =    [1      2      0.1      0.2      0.04;
           1      4      0.05     0.2      0.04;
           1      5      0.08     0.3      0.06;
           2      3      0.05     0.25     0.06;
           2      4      0.05     0.1      0.02;
           2      5      0.1      0.3      0.04;
           2      6      0.07     0.2      0.05;
           3      5      0.12     0.26     0.05;
           3      6      0.02     0.1      0.02;
           4      5      0.2      0.4      0.08;
           5      6      0.1      0.3      0.06];

```

Fonte: Autoria própria.

No arquivo modelo “DadosDoSistema.m” também constam três rotinas de cálculo. A primeira serve para dividir os valores de susceptância por 2, visto que nos cálculos considera-se o modelo  $\pi$  de linhas de transmissão. A segunda rotina de cálculo transforma os valores de potência e shunt para pu, utilizando o valor inserido como potência aparente base do sistema. A terceira rotina transforma os valores dos ângulos “theta” de graus para radianos, para que os cálculos sejam efetuados corretamente pelo MATLAB.

### 4.3. ESTRUTURA DO ALGORITMO

#### 4.3.1. Rotina que calcula o Fluxo de Potência

O cálculo do Fluxo de Potência pelo método de Newton-Raphson, considerando os limites de reativo das máquinas síncronas, é realizado no arquivo “FLOBA.m”. Esse arquivo não deve ser modificado pelo usuário, já que os dados do sistema são inseridos previamente através do arquivo “DadosDoSistema.m”, conforme indicado nas considerações iniciais do algoritmo (Figura 14).

Figura 14 – Considerações iniciais do algoritmo “FLOBA.m”

```

%----- FLOBA -----%
% Estudo do Fluxo de Carga, pelo método de Newton Raphson, considerando
% os limites de reativos das máquinas síncronas do sistema.
% Alunos: Alice Fey, Bruno Soares e Olivia Nichele.
% Prof.(a): Andréa Lucia Costa.
% Trabalho de conclusão do curso de Eng. Elétrica - UTFPR - 2019/1
%-----%

%----- OBSERVAÇÕES -----%
% Antes de ser executado o FLOBA, deve ser executado o arquivo .m com os
% dados de entrada do sistema, seguindo o modelo "DadosDoSistema.m".
% Os dados devem ser inseridos conforme as grandezas indicadas no modelo,
% respeitando também os tipos de barras.

%----- Variáveis utilizadas -----%
aviso=0;          % Variável que indica convergência (0 = não convergência)
                  %                               (1 = convergência)

tolP=e;
tolQ=e;
iteracoes=0;    % Variável que indica o número de iterações efetuadas.
BarraCritica=0; % Variável que indica qual barra do sistema mais excedeu
                  % o limite de geração de reativos da máquina síncrona.

%-----%

% Função para indicar qual o tipo da barra (PQ, PV ou Vtheta)
tip = DBAR(:,2);

```

Fonte: Autoria própria.

O algoritmo foi separado em seções através de comentários no código, que indicam o que cada rotina realiza, para facilitar o entendimento. A primeira seção após as considerações iniciais é “Variáveis Utilizadas”, que diz respeito às variáveis utilizadas para o cálculo do Fluxo de Carga pelo método de Newton-Raphson. A primeira variável é o “aviso”, que indica convergência quando o valor é igual a 1, e não convergência, quando igual a 0. Tem-se, também, as variáveis “iteracoes” (indica o número de iterações efetuadas), “BarraCritica” (indica a barra do sistema que mais excedeu o limite de reativos da máquina síncrona), “tolP” e “tolQ” (indicam a tolerância das potências ativa e reativa, respectivamente, e são especificadas iguais ao erro tolerado).

A montagem da matriz de admitâncias (Y) é realizada utilizando os valores inseridos no arquivo com os dados do sistema em “DBAR” e “DLIN”, de origem e destino da linha, bem como os valores de resistência e reatância. Realiza-se o cálculo dos elementos da matriz conforme mostrado na Figura 15.

Figura 15 – Montagem da Matriz Y

```

%----- MONTAGEM DA MATRIZ Y(admitâncias)-----%
Y = zeros(NUMBAR,NUMBAR);           % Criação da matriz Y preenchida com zeros

for t = 1:NUMLIN
    ORIGEM = DLIN(t,1);
    DESTINO = DLIN(t,2);
    RESISTENCIA = DLIN(t,3);
    REATANCIA = DLIN(t,4);
    %Cálculo das diagonais secundárias:
    Y(ORIGEM,DESTINO) = -1/(RESISTENCIA + j*REATANCIA);
    %Cálculo das diagonais secundárias:
    Y(DESTINO,ORIGEM) = Y(ORIGEM,DESTINO);
    %Cálculo da diagonal principal:
    Y(ORIGEM,ORIGEM) = (1/(RESISTENCIA + j*REATANCIA)) + j*DLIN(t,5) + Y(ORIGEM,ORIGEM);
    %Cálculo da diagonal principal:
    Y(DESTINO,DESTINO) = (1/(RESISTENCIA + j*REATANCIA)) + j*DLIN(t,5) + Y(DESTINO,DESTINO);
end

```

Fonte: Autoria própria.

A estrutura mostrada na Figura 16 demonstra como são calculados os valores iniciais de injeção de potência ativa e potência reativa, sendo a diferença entre a potência gerada na barra por uma unidade geradora e a potência consumida na barra.

Figura 16 – Cálculo dos valores iniciais de injeção de potência

```

% Cálculo dos valores iniciais de injeção de pot. ativa e de pot. reativa
for i = 1:NUMBAR
    Pesp(i) = DBAR(i,5) - DBAR(i,7);           % Pesp = P(geração) - P(carga)
end
for i = 1:NUMBAR
    Qesp(i) = DBAR(i,6) - DBAR(i,8);           % Qesp = Q(geração) - Q(carga)
end

```

Fonte: Autoria própria.

A Figura 17 mostra a criação das matrizes **G** e **B**.

Figura 17 – Criação das matrizes G e B

```

% Criação das matrizes G e B
G=zeros(NUMBAR,NUMBAR); B=zeros(NUMBAR,NUMBAR);
G(:, :) = real(Y(:, :)); B(:, :) = imag(Y(:, :));

```

Fonte: Autoria própria.

Determina-se o número de barras *PQ* e barras *PV* através da estrutura mostrada na Figura 18.

Figura 18 – Cálculo do número de barras *PQ* e *PV*

```
% Determinação do numero de barras PQ e barras PV
npq=0; npv=0;
for i=1:NUMBAR
    if tip(i,1)==0
        npq = npq+1;
    end
end

for i=1:NUMBAR
    if tip(i,1) == 1
        npv = npv+1;
    end
end
npqpv=npq+npv;
```

Fonte: Autoria própria.

São criados os vetores **V** (tensões nas barras) e “theta” (ângulos das tensões das barras) conforme a Figura 19. Estes vetores são utilizados nas próximas rotinas de cálculo e contém os dados do estado da rede, que é mostrado ao usuário no relatório final.

Figura 19 – Montagem dos vetores **V** e “theta”

```
% Montagem do vetor V(tensões das barras) e theta(ângulos das tensões das
% barras)
V = DBAR(:,3); theta = DBAR(:,4);
```

Fonte: Autoria própria.

A partir dos comandos mostrados na Figura 19, os comandos subsequentes correspondem a uma rotina (dentro de um comando *while*) a ser executada até que a solução do Fluxo de Carga apresente convergência (Figura 20). Ou seja, até que a variável “aviso” atinja valor 1, os cálculos apresentados a seguir se repetem no laço que envolve esta rotina.



Figura 20 – Apresentação da rotina de aviso

```

%-----%
% Rotina que será executada até que a solução do Fluxo de Carga apresente
% convergência
while aviso==0

```

Fonte: Autoria própria.

Os comandos dentro do laço do comando *while* iniciam-se com os cálculos das injeções de potência ativa ( $P_i$ ) e reativa ( $Q_i$ ), conforme mostrado na Figura 21.

Figura 21 – Cálculos das injeções de potência ativa ( $P_i$ ) e reativa ( $Q_i$ )

```

%----- Cálculo das Injeções de Potência -----%
% Cálculo da injeção de potência ativa (Pi)
P=zeros(NUMBAR,1);
for k=2:NUMBAR
    som = 0;
    for m=1:NUMBAR
som = som + V(m,1) * (G(k,m) * cos(theta(k,1)-theta(m,1)) + B(k,m) * sin(theta(k,1)-theta(m,1)));
    end
    P(k,1)=V(k,1) * som;
end

% Cálculo da injeção de potência reativa (Qi)
Q=zeros(NUMBAR,1);
for k=1:NUMBAR
    som = 0;
    for m=1:NUMBAR
som = som + V(m,1) * (G(k,m) * sin(theta(k,1)-theta(m,1)) - B(k,m) * cos(theta(k,1)-theta(m,1)));
    end
    Q(k,1)=V(k,1) * som;
end

```

Fonte: Autoria própria.

Efetua-se, então, o teste dos resíduos de potência, em que se compara o valor de  $g(x)$  com a tolerância “ $\epsilon$ ” (Figura 22). Se o valor do módulo de  $g(x)$  for menor ou igual à tolerância, a solução está dentro da faixa e tem-se o estado da rede. Caso contrário, o processo iterativo deve continuar.

Figura 22 – Teste dos resíduos de potência

```

%----- Teste dos resíduos/mismatches de potência [g(x)] -----%

% Comparar o valor de g(x) com a tolerância "e". Se |g(x)|<=e, a solução
% procurada está dentro da faixa de tolerância, portanto tem-se o estado
% da rede, caso contrário o processo iterativo deve continuar.

i=0;
for k=1:NUMBAR
    if tip(k,1) ~= 2
        i=i+1;
        dP(i,1) = Pesp(k,1) - P(k,1); % Resíduo de potência ativa
        deltaP(i,1) = abs(dP(i,1));
    end
end

i=0;
for k=1:NUMBAR
    if tip(k,1) == 0
        i=i+1;
        dQ(i,1) = Qesp(k,1) - Q(k,1); % Resíduo de potência reativa
        deltaQ(i,1) = abs(dQ(i,1));
    end
end

for i=1:npqpv
    residuo(i,1)=dP(i,1);
end
for n=1:npq
    i=npqpv+n;
    residuo(i,1)=dQ(n,1);
end

```

Fonte: Autoria própria.

Em seguida tem-se o cálculo do teste de convergência do sistema (Figura 23). Caso a convergência não seja alcançada, deve ser efetuada a rotina para calcular a matriz Jacobiana e encontrar os valores de correção para as tensões complexas nas barras.

Figura 23 – Teste de convergência do sistema

```

% Teste da convergência

for i=1:npqpv
    if deltaP(i,1)>tolP
        aviso=0;
    else aviso=1;
    end
end

if aviso==1
    for i=1:npq
        if deltaQ(i,1)>tolQ
            aviso=0;
        else aviso=1;
        end
    end
end

%-----%
% Caso a convergência não tenha sido alcançada, será efetuada a rotina para
% calcular a matriz Jacobiana e encontrar os valores de correção para as
% tensões complexas nas barras.

if aviso==0

%----- Cálculo da matriz Jacobiana -----%
% J = [ H  N
%       M  L ]

```

Fonte: Autoria própria.

Para obter-se a matriz Jacobiana (Equação 3.20), já em sua forma reduzida, são calculados os valores de cada um dos termos das submatrizes que compõem a Jacobiana (chamadas de **H**, **N**, **M** e **L**), sendo posteriormente montada a Jacobiana reduzida. Os termos das submatrizes **H**, **N**, **M** e **L** (Figura 24) são calculados separadamente, sendo empregadas rotinas distintas para os termos da diagonal principal e os termos fora da diagonal principal.

Figura 24 – Cálculo da submatriz H

```

% Montagem da matriz M
M=zeros(NUMBAR,NUMBAR);

% Cálculo de M(k,k)- diagonal principal
for k=1:NUMBAR
    som=0;
    for m=1:NUMBAR
som = som + V(m,1) * (G(k,m) *cos(theta(k,1)-theta(m,1)) +B(k,m) *sin(theta(k,1)-theta(m,1)));
    end
    M(k,k) = -G(k,k) * (V(k,1) ^2) +V(k,1) *som;
end

% Cálculo de M(k,m)
for k=1:NUMBAR
    for m=(k+1):NUMBAR
M(k,m) = -V(k,1) *V(m,1) * (G(k,m) *cos(theta(k,1)-theta(m,1)) +B(k,m) *sin(theta(k,1)-theta(m,1)));
    end
end

% Cálculo de M(m,k)
for k=1:NUMBAR
    for m=(k+1):NUMBAR
M(m,k) = -V(k,1) *V(m,1) * (G(k,m) *cos(theta(k,1)-theta(m,1)) -B(k,m) *sin(theta(k,1)-theta(m,1)));
    end
end

```

Fonte: Autoria própria.

Elabora-se, então, a matriz Jacobiana reduzida, formada pelos valores que provém das submatrizes **H**, **N**, **M** e **L**. A rotina para inserir os valores de **H** na matriz é mostrada na Figura 25. As rotinas para as demais submatrizes apresentam a mesma estrutura. A matriz Jacobiana reduzida deve ser quadrada para se possa calcular sua inversa nas próximas etapas do cálculo. A dimensão da matriz (número de linhas, bem como o número de colunas) deve ser a soma do número de barras *PQ* multiplicado por 2, adicionado ao número de barras *PV*.

Figura 25 – Inserindo valores de H na Matriz Jacobiana Reduzida

```

%----- Montando a matriz Jacobiana (J) reduzida -----
-%
% Inserindo os valores de H na matriz Jacobiana
linha=0;
coluna=0;
for i=2:NUMBAR
    linha=linha+1;
    coluna=0;
    for n=2:NUMBAR
        coluna=coluna+1;
        J(linha,coluna)=H(i,n);
    end
end
end

```

Fonte: Autoria própria.

Para encontrar novos valores das tensões complexas, a serem utilizados na próxima iteração, deve-se calcular a matriz Jacobiana inversa ( $\mathbf{J}_i$ ) e o vetor de correções ( $\mathbf{\Delta X}$ ), de acordo com a Figura 26.

Figura 26 – Cálculo de  $\mathbf{J}_i$  e  $\mathbf{\Delta X}$ 

```

%----- Correção dos valores de tensões complexas -----%
% Calculando a matriz J inversa e o vetor de correções (deltaX)

Ji=inv(J);
deltaX=Ji*residuo;

```

Fonte: Autoria própria.

Cria-se um vetor de estados ( $\mathbf{X}$ ) com os valores iniciais de tensões (somente pra barras  $PQ$ ) e ângulos (para barras  $PQ$  e  $PV$ ). Atualiza-se esse vetor, criando um vetor ( $\mathbf{X}_{\text{novo}}$ ) e somando o valor das tensões complexas da iteração anterior ( $\mathbf{X}$ ) com as correções nas tensões complexas ( $\mathbf{\Delta X}$ ), conforme a Figura 27.

Figura 27 – Vetor X e atualização do Vetor X

```

% Criando um vetor X com os valores iniciais de tensões e ângulos
% sendo X=[theta (barras PQ e PV)
%          V ] (somente barras PQ)

i=0;
for k=1:NUMBAR
    if tip(k,1) ~= 2
        i=i+1;
        X(i,1) = theta(k,1);
    end
end

i=npqpv;
for k=1:NUMBAR
    if tip(k,1) == 0
        i=i+1;
        X(i,1) = V(k,1);
    end
end

% Atualizando o vetor X com as correções das tensões complexas

Xnovo=deltaX+X;

```

Fonte: Autoria própria.

Inserire-se os valores corrigidos dos ângulos  $\theta$  no vetor “theta”, bem como os valores corrigidos de magnitude de tensão no vetor **V**, conforme mostra a Figura 28.

Figura 28 – Atualização dos vetores “theta” e V

```

% Inserindo os valores corrigidos dos ângulos theta no vetor "theta"
for i=1:npqpv
    theta(i+1,1)=Xnovo(i,1);
end

% Inserindo os valores corrigidos de magnitude de tensão no vetor "V"
n=npqpv;
for i=1:NUMBAR
    if tip(i,1)==0
        n=n+1;
        V(i,1)=Xnovo(n,1);
    end
end

```

Fonte: Autoria própria.

Atualiza-se o valor da variável “iteracoes”, que indica o número de iterações efetuadas, através do cálculo “iteracoes=iteracoes+1”.

Finaliza-se, portanto, o laço do comando *while* no algoritmo FLOBA. Caso a variável “aviso” atinja o valor 1, indica-se a convergência do sistema de acordo com a tolerância ( $\epsilon$ ) pré-estabelecida. Para verificar de forma mais detalhada todos os comandos presentes no laço *while*, consulte o APÊNDICE B.

#### 4.3.2. Rotina que apresenta o estado da rede

Dentre os resultados apresentados pelo algoritmo FLOBA, tem-se a quantidade de iterações que foram necessárias para a convergência do sistema e o valor das tensões complexas em cada barra. A Figura 29 ilustra os comandos elaborados para esta etapa.

Figura 29 – Rotina para apresentar o estado da rede

```

%----- Rotina para mostrar o estado da rede -----%

fprintf('\n-----\n');
fprintf('-----\n');
fprintf('RELATÓRIO FLOBA\n');
fprintf('-----\n');
fprintf('-----\n\nNúmero de iterações realizadas = %i',iteracoes);

Barra=zeros(k,1);          %Criando um vetor com o número de barras do sistema
for k=1:NUMBAR
    Barra(k,1)=k;
end

% Equação para passar os valores dos ângulos de radianos para graus
theta=theta*180/pi;

EstadoDaRede=table(Barra,V,theta) %Mostrando uma tabela com o estado da rede

```

Fonte: Autoria própria.

A Figura 30 apresenta parte do relatório de resultados do FLOBA, que consta o número de iterações realizadas e o estado da rede obtido (valores de tensão complexa de cada barra do sistema, sendo os valores de “theta” apresentados em graus). Estes são resultados para o Sistema Teste de 6 barras (arquivo “DadosDoSistemaSeisBarras.m”, descrito no APÊNDICE A).

Figura 30 – Relatório FLOBA: Estado da rede e quantidade de iterações utilizando o Sistema Teste de 6 barras

```

-----
                                RELATÓRIO FLOBA
-----

Número de iterações realizadas = 3
EstadoDaRede =

  Barra      V      theta
-----
  1          1.05      0
  2          1.05     -3.6699
  3          1.07     -4.2716
  4          0.98938   -4.195
  5          0.98546   -5.2753
  6          1.0045   -5.9454
-----

```

Fonte: Autoria própria.

#### 4.3.3. Rotina para verificar a geração de reativos nas barras

A proposta do algoritmo implementado é a de calcular o Fluxo de Carga considerando os limites de reativos das máquinas síncronas, portanto foi elaborada uma rotina para verificar a geração de reativos nas barras  $PV$  e  $V\theta$  do sistema analisado (Figura 31). Criou-se um vetor para armazenar os valores de potência reativa gerada em cada barra, um vetor para sinalizar qual(is) barra(s) atingiu o limite de reativos, e, um terceiro vetor para armazenar o desvio entre a potência máxima e o atual valor de potência reativa encontrado após a primeira solução do Fluxo de Carga pelo algoritmo FLOBA.

Figura 31 – Rotina para verificação de geração de reativos nas barras

```

%-----%
%----- Verificação da geração de reativos nas barras -----%

for k=1:NUMBAR
    Qg(k,1)=Q(k,1)+DBAR(k,8); % Vetor que guarda os valores de Q gerados
end

limitereativo=zeros(NUMBAR,1); % Vetor que sinaliza qual barra atingiu o
                                % limite de geração de reativos (+1 ou -1.
QgDesvio=zeros(NUMBAR,1);      % Variável que armazena o desvio entre o
                                % Qg máximo e o atual valor de Qg.

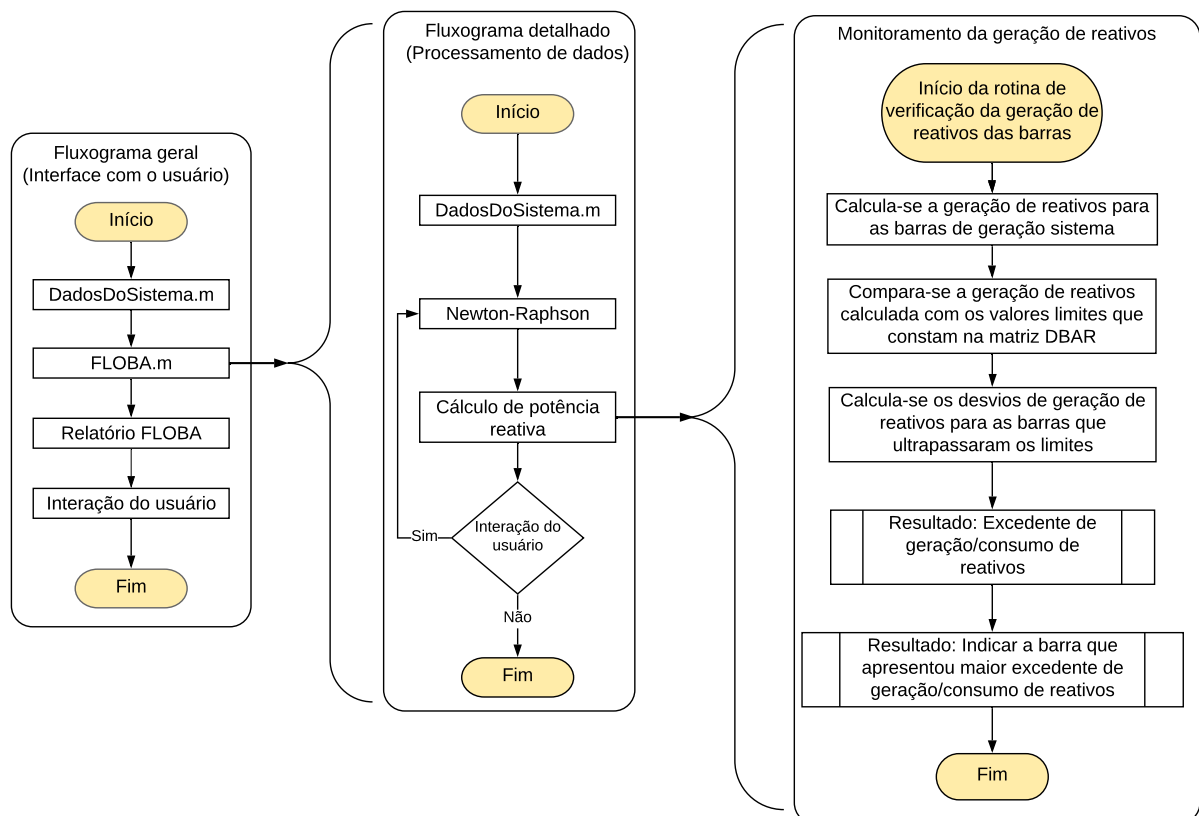
```

Fonte: Autoria própria.



Na Figura 32 é mostrado o fluxograma geral que descreve a interface do FLOBA com o usuário, juntamente ao fluxograma que sumariza o processamento de dados do algoritmo, e em um detalhamento sobre como ocorre o monitoramento da geração de reativos das máquinas síncronas do sistema. Após a convergência do sistema e a obtenção do estado da rede, calcula-se a geração de reativos para todas as barras de geração. Compara-se, então, estes valores com os valores limites que constam na matriz “DBAR”, calcula-se os desvios de geração de reativos para as barras que ultrapassaram os limites, indicando o excedente de geração/consumo de reativos, e quais a(s) barra(s) do sistema que ultrapassaram os limites máximos ou mínimos de potência reativa. A Figura 33 apresenta a rotina descrita.

Figura 32 – Fluxograma detalhado do monitoramento da geração de reativos das máquinas síncronas do sistema



Fonte: Autoria própria.

Figura 33 – Rotina para a verificação da violação do limite de geração de potência reativa

```

% Cálculo dos desvios de geração de reativos para as barras que ultrapassa-
% ram os limites (Qgmax ou Qgmin de DBAR)

fprintf('\n-----');
fprintf('-----\n\nMonitoramento da geração/consumo de');
fprintf(' reativos das máquinas síncronas do sistema:\n\n');

for k=1:NUMBAR
    if tip(k)~=0
        if Qg(k,1)>DBAR(k,11) % Verificação do excedente de Qgmax
            limitereativo(k,1)=1;
            QgDesvio(k,1)=abs(DBAR(k,11)-Qg(k,1));
            QgmaxDesvio=QgDesvio(k,1);
            fprintf(' - A barra %i excedeu em %fpu ',k,QgmaxDesvio);
            fprintf('o limite máximo de geração de reativos. \n');
        end
        if Qg(k,1)<DBAR(k,10) % Verificação do excedente de Qgmin
            limitereativo(k,1)=-1;
            QgDesvio(k,1)=abs(DBAR(k,10)-Qg(k,1));
            QgminDesvio=QgDesvio(k,1);
            fprintf(' - A barra %i excedeu em %fpu ',k,QgminDesvio);
            fprintf('o limite máximo de consumo de reativos. \n');
        end
    end
end
end

```

Fonte: Autoria própria.

A Figura 34 ilustra a rotina utilizada para indicar a barra que obteve maior excedente de geração ou de consumo de potência reativa, em relação ao limite pré-estabelecido na entrada de dados do sistema. Esta rotina salva o número da barra mais crítica do sistema, em relação ao excedente de geração de reativos, bem como retorna ao usuário tal informação.

Figura 34 – Rotina para indicar a barra que obteve maior excedente de geração ou consumo de potência reativa

```

QgDesvioMax=max(QgDesvio);
if QgDesvioMax~=0
    for k=1:NUMBAR
        if QgDesvio(k,1)==QgDesvioMax
            fprintf('\nBarra que apresentou o maior excedente de ');
            fprintf('geração/consumo de reativos = %i\n',k);
            BarraCritica=k;
        end
    end
end

```

Fonte: Autoria própria.

Portanto, ao executar o FLOBA, é possível visualizar também no relatório de resultados, as barras que excederam o limite de reativos, o valor excedido em relação ao limite

máximo de potência reativa, bem como a barra que apresentou maior excedente de geração ou consumo de reativos, conforme mostrado na Figura 35.

Figura 35 – Relatório FLOBA: Monitoramento da geração/consumo de reativos das máquinas síncronas do sistema

```

-----
Monitoramento da geração/consumo de reativos das máquinas síncronas do sistema:

- A barra 2 excedeu em 0.043320pu o limite máximo de geração de reativos.
- A barra 3 excedeu em 0.195954pu o limite máximo de geração de reativos.

Barra que apresentou o maior excedente de geração/consumo de reativos = 3
-----

```

Fonte: Autoria própria.

Havendo excesso de reativos nas máquinas síncronas do sistema, o usuário pode decidir entre duas opções. A primeira opção é escolher e digitar o número de uma das barras que excedeu o limite de reativos, podendo, ou não, ser a que apresentou o maior excedente, para que o FLOBA altere os parâmetros desta barra (passando a ser uma barra  $PQ$ ) e recalcule o fluxo de potência considerando o limite da máquina síncrona, fixando um valor de potência reativa na barra. A segunda escolha é optar por não recalculá-lo, enviando 0 (zero) na janela de comando do MATLAB, finalizando, assim, a simulação. A rotina para mostrar ao usuário essas opções é mostrada na Figura 36.

Figura 36 – Rotina para mostrar opções ao usuário

```

fprintf('\n-----\n');
fprintf('-----\n');
fprintf('\n## Digite o número de alguma das barras PV do sistema');
fprintf(' (exceto a barra 1) que o\n## FLOBA irá alterar os ');
fprintf('parâmetros, para que então seja recalculado o fluxo de');
fprintf('\n## potência considerando o limite da máquina síncrona.');
```

comando=input(' Ou aperte 0 para finali-\n## zar a execução:');

Fonte: Autoria própria.

Mostram-se as opções para o usuário na janela de comando do MATLAB no relatório do FLOBA, que se apresenta completo conforme a Figura 37.

Figura 37 – Relatório FLOBA

```

-----
                                RELATÓRIO FLOBA
-----

Número de iterações realizadas = 3
EstadoDaRede =

      Barra      V      theta
      -----      -      -
      1          1.05      0
      2          1.05     -3.6699
      3          1.07     -4.2716
      4          0.98938   -4.195
      5          0.98546   -5.2753
      6          1.0045   -5.9454

-----

Monitoramento da geração/consumo de reativos das máquinas síncronas do sistema:

- A barra 2 excedeu em 0.043320pu o limite máximo de geração de reativos.
- A barra 3 excedeu em 0.195954pu o limite máximo de geração de reativos.

Barra que apresentou o maior excedente de geração/consumo de reativos = 3

-----

## Digite o número de alguma das barras PV do sistema (exceto a barra 1) que o
## FLOBA irá alterar os parâmetros, para que então seja recalculado o fluxo de
## potência considerando o limite da máquina síncrona. Ou aperte 0 para finali-
## zar a execução:

```

Fonte: Autoria própria.

Caso o usuário opte por recalculer o Fluxo de Carga, considerando os limites da máquina síncrona de uma determinada barra, o algoritmo faz a alteração de alguns parâmetros desta barra e elimina a violação dos limites de reativos da máquina síncrona em questão. Esta ação não garante que outra barra deixe de apresentar excedentes na geração ou consumo de reativos. Os parâmetros da matriz “DBAR” a serem alterados, para a barra escolhida, são: o tipo da barra (de *PV* a barra passa a ser *PQ*), a tensão (1 pu), a potência ativa de carga ( $P_l$ ) assume o valor oposto de potência ativa de geração ( $P_g$ ), a potência reativa de carga ( $Q_l$ ) assume o valor oposto do limite máximo de potência reativa de geração ( $Q_{g\max}$ ), as potências ativa ( $P_g$ ) e reativa ( $Q_g$ ) de geração assumem o valor zero e os limites mínimos e máximos de potência reativa ( $Q_{g\min}$  e  $Q_{g\max}$ ) assumem o valor zero, pois a barra passa ser de tipo *PV* para tipo *PQ*. A rotina do algoritmo que realiza essas alterações é mostrada na Figura 38.

Figura 38 - Rotina para alteração dos parâmetros das barras e recálculo do Fluxo de Potência

```

if comando~=0
    if comando~=1
        if comando<=NUMBAR
            if limitereativo(comando,1)~=0
                BarraModificada=comando; % Atribui à variável a barra escolhida
                DBAR(BarraModificada,2)=0; % Muda o tipo da barra de PV para PQ
                DBAR(BarraModificada,3)=1; % Altera tensão para 1 pu
                DBAR(BarraModificada,4)=0; % Altera ângulo theta para 0 graus
                % Realoca o valor de Pg para Pl:
                DBAR(BarraModificada,7)=-DBAR(BarraModificada,5);
                if limitereativo(BarraModificada,1)==1
                    % Realoca o valor de Qgmax para Ql:
                    DBAR(BarraModificada,8)=-DBAR(BarraModificada,11);
                else
                    % Realoca o valor de Qgmin para Ql:
                    DBAR(BarraModificada,8)=-DBAR(BarraModificada,10);
                end
                % Modifica para zero os valores que não serão mais utilizados:
                DBAR(BarraModificada,5)=0;
                DBAR(BarraModificada,6)=0;
                DBAR(BarraModificada,10)=0;
                DBAR(BarraModificada,11)=0;
            end
            run FLOBA
        end
    end
end

```

Fonte: Autoria própria.

Se o usuário erroneamente optar por alterar os parâmetros de uma das barras que não apresentou violação na geração ou consumo de reativos, ou alterar a barra de referência, ou alterar uma barra que não existe, o algoritmo não executa nenhuma alteração. São mostrados avisos na janela de comando do MATLAB e finaliza-se a simulação, conforme o código mostrado na Figura 39.

Figura 39 – Rotina para finalização da simulação

```

else fprintf('\n!! Esta barra não apresentou violação na ');
    fprintf('geração/consumo de reativos, rode o\n');
    fprintf('!! FLOBA novamente. \n\n');
end % end para caso o comando tenha sido de uma barra que não
    % violou os limites de reativos
else fprintf(' \n!! Este número de barra não existe,');
    fprintf(' rode o FLOBA novamente. \n\n');
end % end para caso o comando tenha sido um valor de barra
    % inexistente
else fprintf('\n!! A barra de referência não pode');
    fprintf(' ser alterada, rode o FLOBA novamente. \n\n');
end % end para caso o comando tenha sido igual a 1
end % end para caso o comando tenha sido igual à zero

```

Fonte: Autoria própria.

Na opção pela não alteração dos parâmetros de alguma das barras que apresentou violação na geração/consumo de reativos, deve-se digitar zero, finalizando a execução da simulação, conforme mostrado no Relatório FLOBA na Figura 37.

Caso não haja barras com excesso de geração ou consumo de potência reativa, uma mensagem é mostrada ao usuário elucidando que não houve violação dos limites de reativos das máquinas síncronas do sistema, encerrando-se, assim, a simulação, de acordo com a rotina da Figura 40 e a mensagem impressa na janela de comando, Figura 41.

Figura 40 – Rotina para finalizar a simulação quando não há violação dos limites

```
else %else para caso não tenha ocorrido violação nos limites
    fprintf('## Não houve violação no limite de geração/consumo de ');
    fprintf('reativos das máquinas\n## síncronas do sistema! \n  :)\n');
end %end para o if QgDesvioMax~=0
```

Fonte: Autoria própria.

Figura 41 – Mensagem impressa na janela de comando quando não há violação dos limites

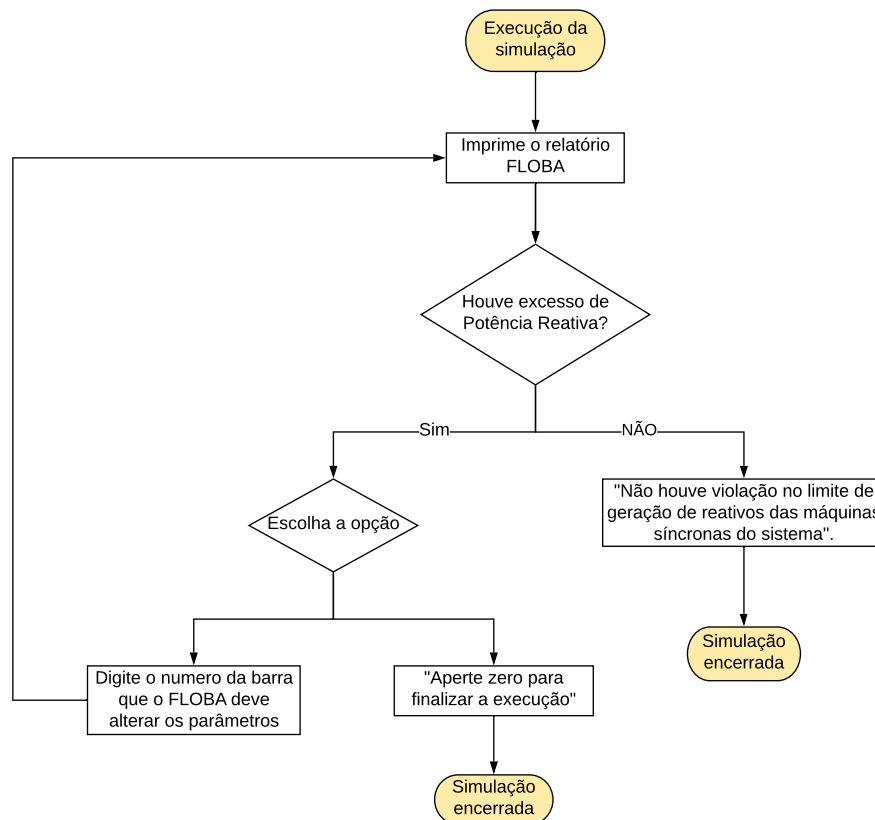
```
-----
Monitoramento da geração/consumo de reativos das máquinas síncronas do sistema:

## Não houve violação no limite de geração/consumo de reativos das máquinas
## síncronas do sistema!
  :)
..
```

Fonte: Autoria própria.

A Figura 42 apresenta o fluxograma de execução do algoritmo FLOBA, ao se considerar o uso da janela de comando do MATLAB.

Figura 42 - Fluxograma de execução do algoritmo FLOBA



Fonte: Autoria própria.

#### 4.3.4. Variáveis que podem ser visualizadas pelo usuário

Além dos resultados fornecidos pelo relatório FLOBA, após a finalizada a execução o usuário pode visualizar mais algumas variáveis importantes empregadas no algoritmo, através da janela de comando do MATLAB. As variáveis são: “DBAR” e “DBARoriginal”, para que o usuário verifique como eram os dados de barra do sistema original e como o FLOBA alterou-os após os comandos; “DLIN”; “V” e “theta”, que são as variáveis mostradas em forma de tabela no relatório FLOBA; “Qg”, que é o vetor com os valores de potência reativa gerada nas barras; “iteracoes”; “J”, que é a última matriz Jacobiana reduzida utilizada nos cálculos; “npv”, que é o número de barras PV; “npq”, que é o número de barras PQ; “limitereativo”, que é um vetor que indica, com zeros e uns, quais barras que excederam (1) ou não (0) o limite de reativos; “NUMBAR”; “NUMLIN”; “BarraCritica”, que é a barra que apresentou o maior excedente na geração ou consumo de reativos, caso alguma barra do sistema tenha violado os limites; e “BarraModificada”, que é a barra escolhida pelo usuário para ser modificada, possibilitando um novo cálculo do Fluxo de Carga considerando os limites de sua máquina síncrona.

#### 4.4. VALIDAÇÃO DO ALGORITMO

Utilizou-se o ANAREDE para validação do algoritmo FLOBA. Os resultados do sistema teste de 6 barras (arquivo “DadosDoSistemaSeisBarras.m”, apresentado no APÊNDICE A) obtidos pelo FLOBA (Figura 43) e pelo ANAREDE (Figura 44) foram comparados, verificando a correta implementação dos cálculos que constam no algoritmo. Obteve-se resultados muito próximos, porém com casas decimais distintas devido aos arredondamentos inerentes a cada uma das ferramentas computacionais. A Figura 44 apresenta a interface do ANAREDE e os resultados do Sistema Teste de 6 barras. Nas áreas destacadas pelas caixas mostram-se: o indicativo de convergência do sistema, a função “FP” utilizada para rodar o Fluxo de Potência pelo ANAREDE e o relatório de resultados das barras do sistema (RBAR). A Figura 45 mostra como os dados do sistema teste de 6 barras foram inseridos no ANAREDE.

Figura 43 – Resultados obtidos para o estado da rede do sistema 6 barras com o algoritmo FLOBA

```

Command Window
>> DadosDoSistemaSeisBarras
>> run FLOBA

-----
                                RELATÓRIO FLOBA
-----

Número de iterações realizadas = 3
EstadoDaRede =

  Barra      V      theta
  -----
  1          1.05      0
  2          1.05     -3.6699
  3          1.07     -4.2716
  4          0.98938   -4.195
  5          0.98546   -5.2753
  6          1.0045   -5.9454
-----

```

Fonte: Autoria própria.



Figura 44 – Relatório (RBAR) do sistema teste de 6 barras com o programa ANAREDE

Num...	Tensão (p.u.)	Ângulo (graus)	Carga Ativa (MW)	Carga Reativa (Mvar)	Geração Ativa (MW)	Geração Reativa (Mvar)	Tipo da Barra
1	1.05	-0.00	0.00	0.00	107.80	16.42	2
2	1.05	-3.65	0.00	0.00	50.00	74.87	1
3	1.07	-4.21	0.00	0.00	60.00	88.59	1
4	0.99	-4.19	70.00	70.00	0.00	0.00	0
5	0.99	-5.30	70.00	70.00	0.00	0.00	0
6	1.01	-5.85	70.00	70.00	0.00	0.00	0

Fonte: Autoria própria.

Figura 45 - Dados de entrada do sistema teste de 6 barras para execução no ANAREDE

```

01 FITU
02 Sistema 6 barras
03 DBAR
04 (Num) OETGb( nome ) G1 ( V ) ( A ) ( Pg ) ( Qg ) ( Qn ) ( Qm ) ( Bc ) ( Pl ) ( Q1 ) ( Sh ) Are ( Vf )
05 1 2 1Barra 1 1.05
06 2 1 1Barra 2 1.05 50.
07 3 1 1Barra 3 1.07 60.
08 4 0 1Barra 4 1.00 70. 70.
09 5 0 1Barra 5 1.00 70. 70.
10 6 0 1Barra 6 1.07 70. 70.
11 99999
12 DLIN
13 (De) d O d (Pa) NcEP ( R% ) ( X% ) (Mvar) (Tap) (Tmn) (Tmx) (Phs) (Bc) (Cn) (Ce) Ns
14 1 2 10. 20. 4.
15 1 4 5. 20. 4.
16 1 5 8. 30. 6.
17 2 3 5. 25. 6.
18 2 4 5. 10. 2.
19 2 5 10. 30. 4.
20 2 6 7. 20. 5.
21 3 5 12. 26. 5.
22 3 6 2. 10. 2.
23 4 5 2. 40. 8.
24 5 6 1. 30. 6.
25 99999
26 FIM

```

Fonte: Autoria própria.

Um Sistema Teste de 3 barras, que consta em (SANTOS; DAHER; NAZARI, 2013), também foi utilizado para validar o FLOBA, sendo os resultados obtidos pelo algoritmo presentes na Figura 46 e os resultados obtidos pelo ANAREDE, juntamente com os dados de entrada do sistema, na Figura 47. Verificou-se a correta implementação dos cálculos presentes no algoritmo FLOBA.

Figura 46 - Resultados obtidos para o estado da rede do sistema teste de 3 barras com o algoritmo FLOBA

```

Command Window
>> DadosDoSistemaCOELE
>> run FLOBA

-----
                          RELATÓRIO FLOBA
-----

Número de iterações realizadas = 3
EstadoDaRede =

  Barra      V      theta
  -----
  1          1.04      0
  2          0.9806   -1.1006
  3          0.98      0.30698
-----
  
```

Fonte: Autoria própria.

Figura 47 – Resultados obtidos na simulação do sistema 3 barras com o programa ANAREDE e dados de entrada

The screenshot shows the ANAREDE software interface. The top window, 'Sistema 3 barras - COELE', has a 'Convergente' status indicator highlighted in green. Below it, the 'RBAR' window displays the following table:

Num...	Tensão (p.u.)	Ângulo (graus)	Carga Ativa (MW)	Carga Reativa (Mvar)	Geração Ativa (MW)	Geração Reativa (Mvar)	Tipo da Barra
1	1.04	-0.00	65.00	30.00	155.02	174.80	2
2	0.98	-1.10	100.00	40.00	0.00	0.00	0
3	0.98	0.31	35.00	20.00	50.00	-89.73	1

The bottom window, 'EditCEPEL - ANAREDE', shows the input data for the system. The data is organized into sections: TITU (System Name), DBAR (Bus Data), DLIN (Line Data), and FIM (End). The bus data section (lines 04-07) lists the buses and their parameters:

```

04 (Num)OETGb( nome )G1( V ) ( A ) ( Pg ) ( Qg ) ( Qn ) ( Qm ) ( Bc ) ( P1 ) ( Q1 ) ( Sh )Are (VF)
05 1 2 1Barra 1 1.04
06 2 0 1Barra 2 1.0
07 3 1 1Barra 3 0.98 50.
  
```

The line data section (lines 10-13) lists the lines and their parameters:

```

10 (De )d O d(Pa )NcEP ( R% ) ( X% ) (Mvar) (Tap) (Tmn) (Tmx) (Phs) (Bc ) (Cn) (Ce)Ns
11 1 2 1 3. 7. 8.2
12 1 3 1 3. 6. 4.2
13 2 3 1 2. 4. 3.1
  
```

Fonte: Autoria própria.

Os resultados de potência reativa gerada (Qg) para as barras do sistema teste de 3 barras obtidos pelo FLOBA e pelo ANAREDE (Figura 48) foram também comparados. Constatou-se também a correta implementação dos cálculos de potência reativa presentes no algoritmo, mas deve-se salientar que o algoritmo FLOBA retorna os resultados em pu, e o programa ANAREDE retorna-os em Mvar. Nas áreas destacadas da Figura 47 mostram-se o resultado do vetor Qg no FLOBA e os resultados de Geração Reativa no ANAREDE.

Figura 48 - Resultados de Qg obtidos na simulação do sistema 3 barras pelo FLOBA e pelo ANAREDE

```

-----
Monitoramento da geração/consumo de reativos das máquinas síncronas do sistema:

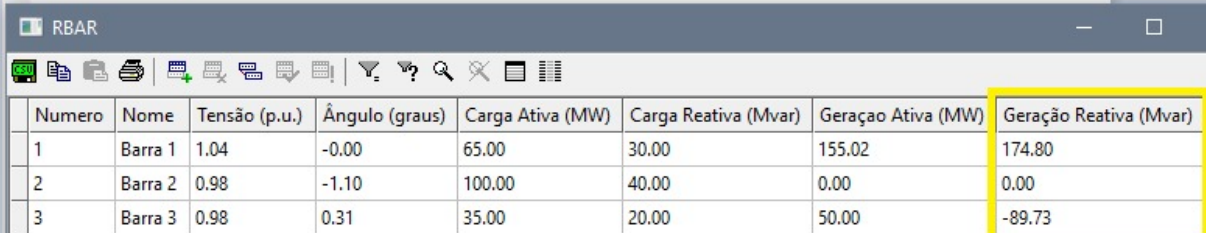
- A barra 1 excedeu em 0.748017pu o limite máximo de geração de reativos.
- A barra 3 excedeu em 0.197357pu o limite máximo de consumo de reativos.

Barra que apresentou o maior excedente de geração/consumo de reativos = 1

-----

## Digite o número de alguma das barras PV do sistema (exceto a barra 1) que o
## FLOBA irá alterar os parâmetros, para que então seja recalculado o fluxo de
## potência considerando o limite da máquina síncrona. Ou aperte 0 para finali-
## zar a execução:0
>> Qg

Qg =
    1.7480
    0.0000
   -0.8974
  
```



Numero	Nome	Tensão (p.u.)	Ângulo (graus)	Carga Ativa (MW)	Carga Reativa (Mvar)	Geração Ativa (MW)	Geração Reativa (Mvar)
1	Barra 1	1.04	-0.00	65.00	30.00	155.02	174.80
2	Barra 2	0.98	-1.10	100.00	40.00	0.00	0.00
3	Barra 3	0.98	0.31	35.00	20.00	50.00	-89.73

Fonte: Autoria própria.

#### 4.5. CONCLUSÃO

Neste capítulo foi apresentado o algoritmo FLOBA, conforme a proposta de implementação deste trabalho. As várias sub-rotinas do algoritmo foram mostradas e detalhadas, bem como apresentada a interface com o usuário e as funcionalidades do FLOBA.

Descreveu-se também como executar corretamente o algoritmo. Os resultados obtidos pelo FLOBA foram comparados aos resultados de Sistemas Teste obtidos com o programa ANAREDE, de uso consagrado por empresas do setor elétrico. Como os resultados foram muito próximos, considera-se que o método de Newton-Raphson foi implementado de forma adequada no algoritmo do FLOBA e que os resultados apresentados são válidos.

## 5. ESTUDOS DE CASO E ANÁLISES

### 5.1. INTRODUÇÃO

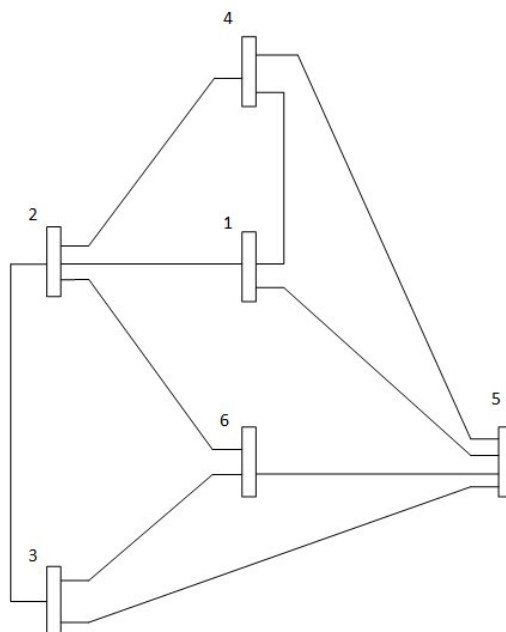
Neste capítulo são mostrados estudos de caso com o sistema de seis barras descrito no APÊNDICE A, visando a verificação de todas as funcionalidades do algoritmo FLOBA. O comportamento dos parâmetros variáveis do sistema em relação à geração de reativos e os limites das máquinas síncronas foi analisado.

Além disso, foram mostradas, de forma mais detalhada, todas as opções conferidas ao usuário do FLOBA, objetivando facilitar a compreensão do algoritmo e a análise dos possíveis resultados.

### 5.2. SISTEMA DE 6 BARRAS

O sistema-teste de 6 barras utilizado nas simulações com o FLOBA é ilustrado na Figura 49. Na Tabela 1, podem ser vistos os dados de barra (inseridos na matriz DBAR) e na Tabela 2, os dados de linha deste sistema (inseridos na matriz DLIN).

Figura 49 – Sistema-teste de 6 barras utilizado nas simulações FLOBA



Fonte: Autoria própria.

Tabela 1 – Dados de Barra do sistema-teste de 6 barras

Dados de Barra											
Barra	Tipo	Tensão (pu)	Ângulo (°)	Pg (MW)	Qg (MW)	Pl (MW)	Ql (Mvar)	Shunt (%)	Qgmin (Mvar)	Qgmáx (Mvar)	Pgmáx (MW)
1	<i>Vθ</i>	1,050	0,0	0,0	0,0	0,0	0,0	0,0	-100	100	200
2	<i>PV</i>	1,050	0,0	50,0	0,0	0,0	0,0	0,0	-100	70	150
3	<i>PV</i>	1,070	0,0	60,0	0,0	0,0	0,0	0,0	-100	70	180
4	<i>PQ</i>	1,000	0,0	0,0	0,0	70,0	70,0	0,0	0,0	0,0	0,0
5	<i>PQ</i>	1,000	0,0	0,0	0,0	70,0	70,0	0,0	0,0	0,0	0,0
6	<i>PQ</i>	1,070	0,0	0,0	0,0	70,0	70,0	0,0	0,0	0,0	0,0

Fonte: Autoria própria.

Tabela 2 - Dados de Linha do sistema-teste de 6 barras

Dados de Linha				
Origem	Destino	Res (pu)	Reat (pu)	Suscept (pu)
1	2	0,10	0,20	0,04
1	4	0,05	0,20	0,04
1	5	0,08	0,30	0,06
2	3	0,05	0,25	0,06
2	4	0,05	0,10	0,02
2	5	0,10	0,30	0,04
2	6	0,07	0,20	0,05
3	5	0,12	0,26	0,05
3	6	0,02	0,10	0,02
4	5	0,20	0,40	0,08
5	6	0,10	0,30	0,06

Fonte: Autoria própria.

Conforme já explanado na seção 4.2, para que os dados do sistema a ser analisado sejam lidos pelo FLOBA, deve-se executar primeiramente um arquivo do formato “DadosDoSistema.m”, como mostrado no APÊNDICE A. Para a realização das análises apresentadas a seguir, foram executados, respectivamente, o arquivo “DadosDoSistemaSeisBarras.m” e o “FLOBA.m”. As Figuras 50 e 51 mostram a janela de comando do MATLAB após o usuário executar e interagir com o algoritmo.

Nota-se que para o sistema-teste de 6 barras, ao executar o FLOBA, o usuário é notificado de que as barras 2 e 3 excederam o limite máximo (geração) de reativos de suas máquinas síncronas, sendo que a barra 3 a que apresentou o maior excedente.

Dentre as possibilidades de interação do usuário com o algoritmo tem-se: enviar comando 0 e finalizar a execução; enviar o número de uma das barras que excedeu os limites de reativos (conforme mostrado nas Figuras 50, 51, 56, 57); enviar o número 1, que é a barra de referência (mostrado na Figura 53); enviar o número de alguma barra que não excedeu os limites de reativos (mostrado na Figura 54); ou enviar o número de uma barra que não exista no sistema analisado (mostrado na Figura 56).

As Figuras 50 e 51 mostram o Relatório FLOBA para o caso em que o usuário escolhe eliminar primeiramente a violação do limite de reativos na barra 3. Na sequência, o algoritmo recalcula o Fluxo de Carga e notifica que a barra 2 ainda apresenta excedentes na geração de reativos. O usuário então escolhe eliminar também a violação para a barra 2 e, ao final da execução, tem-se o estado da rede para a situação de não violação dos limites de reativos para todas as barras do sistema teste.

Analisando os resultados do FLOBA sobre as variáveis das barras do sistema, conforme o usuário foi interagindo com o algoritmo da maneira descrita, percebeu-se que os valores de tensões complexas nas barras modificadas não se mantiveram nos níveis pré-estabelecidos. Esse comportamento é em decorrência da mudança do tipo das barras, que não foram mais consideradas barras *PV* e passaram a ser *PQ*. Portanto, uma das possibilidades para que o sistema de 6 barras não viole nenhum limite de reativos de suas máquinas síncronas, é que os níveis de tensão das barras 2 e 3 deixem de ser 1,05 pu e 1,07 pu, respectivamente, e sejam em torno de 1,03 pu, com uma menor abertura angular. Notou-se também, após a interação do usuário, que para se atingir a convergência do sistema de 6 barras, o número de iterações passou de 3 para 4.

Percebeu-se também que, após a primeira escolha do usuário (eliminar o excedente de reativos para a barra 3), para que o Fluxo de Carga continuasse convergente, o excedente de geração de reativos da barra 2 subiu de 0,0433 pu para 0,2124 pu. Esse comportamento é em decorrência de a máquina síncrona presente na barra 2 ter de assumir parte da geração de reativos que foi restringida na barra 3.

Para tomar melhor conhecimento das alterações que o FLOBA realizou na matriz DBAR, a Figura 52 mostra sua versão inicial (DBAR<sub>original</sub>), com todos os valores já em pu, e sua versão final (DBAR), após a interação com o usuário e a eliminação dos excedentes de reativos. No APÊNDICE A pode se verificar o que cada coluna das matrizes representa.

Figura 50 - Relatório de resultados FLOBA com interação do usuário (parte a)

```
>> DadosDoSistemaSeisBarras
>> run FLOBA
```

```
-----
RELATÓRIO FLOBA
-----
```

```
Número de iterações realizadas = 3
EstadoDaRede =
```

Barra	V	theta
1	1.05	0
2	1.05	-3.6699
3	1.07	-4.2716
4	0.98938	-4.195
5	0.98546	-5.2753
6	1.0045	-5.9454

```
-----
Monitoramento da geração/consumo de reativos das máquinas síncronas do sistema:
```

- A barra 2 excedeu em 0.043320pu o limite máximo de geração de reativos.
- A barra 3 excedeu em 0.195954pu o limite máximo de geração de reativos.

```
Barra que apresentou o maior excedente de geração/consumo de reativos = 3
```

```
-----
## Digite o número de alguma das barras PV do sistema (exceto a barra 1) que o
## FLOBA irá alterar os parâmetros, para que então seja recalculado o fluxo de
## potência considerando o limite da máquina síncrona. Ou aperte 0 para finali-
## zar a execução:3
```

```
-----
RELATÓRIO FLOBA
-----
```

```
Número de iterações realizadas = 4
EstadoDaRede =
```

Barra	V	theta
1	1.05	0
2	1.05	-3.6687
3	1.0523	-3.9703
4	0.98829	-4.1877
5	0.97828	-5.1949
6	0.99285	-5.7691



Figura 51 - Relatório de resultados FLOBA com interação do usuário (parte b)

Monitoramento da geração/consumo de reativos das máquinas síncronas do sistema:

- A barra 2 excedeu em 0.212444pu o limite máximo de geração de reativos.

Barra que apresentou o maior excedente de geração/consumo de reativos = 2

-----

## Digite o número de alguma das barras PV do sistema (exceto a barra 1) que o  
## FLOBA irá alterar os parâmetros, para que então seja recalculado o fluxo de  
## potência considerando o limite da máquina síncrona. Ou aperte 0 para finali-  
## zar a execução:2

-----

RELATÓRIO FLOBA

-----

Número de iterações realizadas = 4

EstadoDaRede =

Barra	V	theta
1	1.05	0
2	1.0297	-3.251
3	1.0342	-3.6143
4	0.97352	-3.9994
5	0.96263	-4.9909
6	0.97358	-5.4844

-----

Monitoramento da geração/consumo de reativos das máquinas síncronas do sistema:

## Não houve violação no limite de geração/consumo de reativos das máquinas  
## síncronas do sistema!

:)

>> |

Fonte: Autoria própria.

Figura 52 - Matrizes DBARoriginal e DBAR

```

DBARoriginal =
  1.0000  2.0000  1.0500  0  0  0  0  0  0 -1.0000  1.0000  2.0000
  2.0000  1.0000  1.0500  0  0.5000  0  0  0  0 -1.0000  0.7000  1.5000
  3.0000  1.0000  1.0700  0  0.6000  0  0  0  0 -1.0000  0.7000  1.8000
  4.0000  0  1.0000  0  0  0  0.7000  0.7000  0  0  0  0
  5.0000  0  1.0000  0  0  0  0.7000  0.7000  0  0  0  0
  6.0000  0  1.0700  0  0  0  0.7000  0.7000  0  0  0  0

>> DBAR

DBAR =
  1.0000  2.0000  1.0500  0  0  0  0  0  0 -1.0000  1.0000  2.0000
  2.0000  0  1.0000  0  0  0 -0.5000 -0.7000  0  0  0  1.5000
  3.0000  0  1.0000  0  0  0 -0.6000 -0.7000  0  0  0  1.8000
  4.0000  0  1.0000  0  0  0  0.7000  0.7000  0  0  0  0
  5.0000  0  1.0000  0  0  0  0.7000  0.7000  0  0  0  0
  6.0000  0  1.0700  0  0  0  0.7000  0.7000  0  0  0  0

```

Fonte: Autoria própria.

Figura 53- Envio do número da barra de referência

```

-----
## Digite o número de alguma das barras PV do sistema (exceto a barra 1) que o
## FLOBA irá alterar os parâmetros, para que então seja recalculado o fluxo de
## potência considerando o limite da máquina síncrona. Ou aperte 0 para finali-
## zar a execução:1

!! A barra de referência não pode ser alterada, rode o FLOBA novamente.

>> |

```

Fonte: Autoria própria.

Figura 54 - Envio do número de alguma barra que não violou os limites de reativos

```

-----
## Digite o número de alguma das barras PV do sistema (exceto a barra 1) que o
## FLOBA irá alterar os parâmetros, para que então seja recalculado o fluxo de
## potência considerando o limite da máquina síncrona. Ou aperte 0 para finali-
## zar a execução:4

!! Esta barra não apresentou violação na geração/consumo de reativos, rode o
!! FLOBA novamente.

>> |

```

Fonte: Autoria própria.

Figura 55 - Envio do número de alguma barra que não consta no sistema analisado

```
-----
## Digite o número de alguma das barras PV do sistema (exceto a barra 1) que o
## FLOBA irá alterar os parâmetros, para que então seja recalculado o fluxo de
## potência considerando o limite da máquina síncrona. Ou aperte 0 para finali-
## zar a execução:8

!! Este número de barra não existe, rode o FLOBA novamente.

>> run FLOBA
```

Fonte: Autoria própria.

Outra possibilidade de interação do usuário com o algoritmo, para que também sejam eliminadas as violações dos limites das máquinas síncronas para o sistema de 6 barras, é apresentada a seguir (Figuras 56 e 57). Desta maneira, o usuário escolheu modificar primeiramente a barra 2, em seguida a barra 3 (anteriormente foi mostrado o caso em que o usuário alterou primeiramente a barra 3, em seguida a barra 2).

Percebe-se que após a primeira escolha do usuário, de alteração dos parâmetros da barra 2, ao recalculer o Fluxo de Carga, para se atingir a convergência ainda são necessárias apenas 3 iterações. Percebeu-se então, conforme o esperado se considerando o que já foi exposto, que a barra 3 continuou excedendo a geração de reativos. Para as configurações originais das barras o excedente da barra 3 era de 0,1959 pu, passando, após modificações na barra 2, para 0,21717 pu.

Para ambas as possibilidades mostradas, de eliminação da violação dos limites das máquinas síncronas, o estado final de rede apresenta-se muito similar. Notou-se que, para o sistema estudado, não há maneira de se considerar os limites das máquinas sem que se alterem os níveis de tensão pré-estabelecidos para as barras *PV* (barras 2 e 3).

Figura 56 - Relatório de resultados FLOBA com interação do usuário - outra possibilidade (parte a)

-----  
 Monitoramento da geração/consumo de reativos das máquinas síncronas do sistema:

- A barra 2 excedeu em 0.043320pu o limite máximo de geração de reativos.
- A barra 3 excedeu em 0.195954pu o limite máximo de geração de reativos.

Barra que apresentou o maior excedente de geração/consumo de reativos = 3

-----  
 ## Digite o número de alguma das barras PV do sistema (exceto a barra 1) que o  
 ## FLOBA irá alterar os parâmetros, para que então seja recalculado o fluxo de  
 ## potência considerando o limite da máquina síncrona. Ou aperte 0 para finali-  
 ## zar a execução:2

-----  
 RELATÓRIO FLOBA  
 -----

Número de iterações realizadas = 3  
 EstadoDaRede =

Barra	V	theta
1	1.05	0
2	1.0478	-3.6209
3	1.07	-4.2612
4	0.9879	-4.1727
5	0.98457	-5.2583
6	1.0037	-5.9285

-----  
 Monitoramento da geração/consumo de reativos das máquinas síncronas do sistema:

- A barra 3 excedeu em 0.217177pu o limite máximo de geração de reativos.

Barra que apresentou o maior excedente de geração/consumo de reativos = 3

-----  
 ## Digite o número de alguma das barras PV do sistema (exceto a barra 1) que o  
 ## FLOBA irá alterar os parâmetros, para que então seja recalculado o fluxo de  
 ## potência considerando o limite da máquina síncrona. Ou aperte 0 para finali-  
 ## zar a execução:3

Fonte: Autoria própria.

Figura 57 - Relatório de resultados FLOBA com interação do usuário - outra possibilidade (parte b)

```

-----
                                RELATÓRIO FLOBA
-----

Número de iterações realizadas = 4
EstadoDaRede =

      Barra      V      theta
      -----      -      -----
      1           1.05           0
      2           1.0297        -3.251
      3           1.0342        -3.6143
      4           0.97352       -3.9994
      5           0.96263       -4.9909
      6           0.97358       -5.4844

-----

Monitoramento da geração/consumo de reativos das máquinas síncronas do sistema:

## Não houve violação no limite de geração/consumo de reativos das máquinas
## síncronas do sistema!
:)
>> |

```

Fonte: Autoria própria.

### 5.3. CONCLUSÃO

Este capítulo apresentou alguns casos de teste para o sistema de 6 barras com intuito de mostrar as diversas possibilidades de execução do algoritmo FLOBA para a análise de uma rede elétrica que, em seu estado original, apresenta violações dos limites de reativos das máquinas síncronas.

Percebeu-se a relação entre as tensões complexas das barras e a geração de reativos, uma vez que para a convergência de um sistema, caso se objetive a eliminação da violação dos limites de reativos, deve-se tomar conhecimento que os níveis de tensão serão afetados.

Conclui-se que o FLOBA permite ao usuário obter informações sobre o sistema analisado e suas diversas possibilidades de configuração, para que ele compreenda melhor a relação entre as variáveis envolvidas na operação de uma rede elétrica.

## 6. CONCLUSÕES OBTIDAS E CONSIDERAÇÕES FINAIS

Para a realização deste trabalho foi necessário o aprofundamento nos estudos sobre Fluxo de Carga em sistemas elétricos de potência, sendo efetuadas inúmeras pesquisas para a correta compreensão matemática e aplicação do Fluxo de Potência pelo método de Newton-Raphson.

Anteriormente à implementação do algoritmo apresentado, foram realizados exercícios e simulações computacionais para embasar o conhecimento necessário. Foram verificadas maneiras de se implementar rotinas de cálculos que encontrassem não somente o estado da rede para um sistema de “n” barras, mas que também considerasse algum dos limites impostos no Fluxo de Potência.

Foi efetuada a implementação computacional do algoritmo de Fluxo de Carga considerando os limites de reativos das máquinas síncronas, sendo intitulado de FLOBA, utilizando a plataforma MATLAB. Pode-se dizer que o algoritmo foi elaborado em duas etapas. A primeira etapa consistiu na realização de rotinas de cálculos do Fluxo de Carga pelo método iterativo escolhido, e posterior validação destes cálculos, consultando exercícios resolvidos. A segunda etapa somente foi possível após a correta execução da primeira, pois aquela resultou na elaboração de rotinas que calculam a geração de reativos nas barras do sistema e consideram os limites de reativos das máquinas síncronas.

Foram estudadas e verificadas as restrições operacionais impostas ao Fluxo de Carga, a existência de valores limitantes e fatores de capacidade, sendo o limite de potência reativa em máquinas síncronas o fator principal considerado neste trabalho. Analisou-se também a relação entre valores de tensão complexa nas barras e a geração de reativos, verificando de que maneira essas grandezas são interdependentes e se relacionam.

Os valores limites de potência reativa utilizados como dados de entrada provém da curva de capacidade das máquinas síncronas e possibilitaram uma análise mais realista do sistema e não a somente uma verificação do estado da rede. É conferido ao usuário a possibilidade de tomar conhecimento sobre quais geradores ultrapassaram os limites de potência reativa, observando as restrições operacionais das máquinas síncronas, possibilitando a eliminação da violação dos limites de reativos para as barras que os excederam, caso seja possível para a configuração do sistema analisado.

Foram elucidadas as condições necessárias para a correta operação do algoritmo FLOBA, sendo também validados os resultados obtidos com sistemas teste, fazendo uso do

programa ANAREDE. Comparou-se os resultados e constatou-se a correta elaboração do algoritmo apresentado nesse trabalho.

Análises dos possíveis resultados a serem obtidos para um determinado sistema também foram efetuadas, visando abranger os casos de teste previstos na elaboração do FLOBA, bem como tomar conhecimento da interdependência entre as inúmeras variáveis envolvidas no Fluxo de Carga.

Espera-se que o FLOBA seja utilizado por mais alunos que desejem se aprofundar nos estudos do Fluxo de Carga de sistemas elétricos e que auxilie na compreensão dos cálculos envolvidos, que por muitas vezes mostram-se “nebulosos”. O algoritmo funciona de maneira satisfatória na análise de sistemas simples, mesmo que com um elevado número de barras, tendo em vista que não considera elementos como tapes de transformadores e outros equipamentos de uma rede.

Sugere-se que, para trabalhos futuros, o algoritmo implementado seja aprimorado. Há a possibilidade de melhorar a maneira como se é executado o monitoramento de geração de reativos, podendo, também, adicionar ao estudo os tapes de transformadores, bem como os shunts das barras. Propõe-se, ainda, a análise e a inserção no algoritmo de elementos de geração distribuída, visto que é uma tecnologia de crescente utilização. Por fim, recomenda-se que o arquivo “FLOBA.m” seja transformado em um arquivo “FLOBA.exe” (executável), para que seja utilizado independentemente em computadores que não apresentem a plataforma MATLAB.

Percebeu-se que ao colocar em prática, de maneira mais estruturada, os conhecimentos até então apenas teóricos, a consolidação dos conceitos ocorre de maneira muito mais profunda.

## REFERÊNCIAS

AGÊNCIA NACIONAL DE ENERGIA ELÉTRICA (ANEEL). **PROCEDIMENTOS DE DISTRIBUIÇÃO DE ENERGIA ELÉTRICA NO SISTEMA ELÉTRICO NACIONAL – PRODIST: Qualidade da Energia Elétrica. Módulo 8**. Brasília, 2017. 88 p. Disponível em: <[http://www.aneel.gov.br/documents/656827/14866914/M%C3%B3dulo\\_8Revis%C3%A3o\\_10/2f7cb862-e9d7-3295-729a-b619ac6baab9](http://www.aneel.gov.br/documents/656827/14866914/M%C3%B3dulo_8Revis%C3%A3o_10/2f7cb862-e9d7-3295-729a-b619ac6baab9)>. Acesso em: 19 nov. 2018.

BORGES, Carmen Lucia Tancredo. **Análise de Sistemas de Potência**. Apostila. Departamento de Eletrotécnica - Universidade Federal do Rio de Janeiro, UFRJ, Rio de Janeiro, 2005.

BUENO, A. A. C., TAVARES, L. F., MAROCHI, M. V. Z. **Desenvolvimento de programa computacional com interface gráfica para cálculo de Fluxo de Potência**. 2012. 176p. Trabalho de Conclusão de Curso (Graduação em Engenharia Industrial Elétrica) - Departamento Acadêmico de Eletrotécnica, Universidade Tecnológica Federal do Paraná, Curitiba, 2012.

CASTRO, Carlos A. **Resolução do Problema de Fluxo de Carga Pelo Método de Newton**. Notas de aula. Escola de Engenharia Elétrica e de Computação. Universidade Estadual de Campinas. São Paulo, 2017. Disponível em: <http://www.dsee.fee.unicamp.br/~ccastro>. Acesso em 10 out. 2018.

CENTRO DE PESQUISAS EM ENERGIA ELÉTRICA (CEPEL). Eletrobras. **ANAREDE**. Disponível em: <<http://www.cepel.br/produtos/programas-computacionais/menu/anarede-analise-de-redes-eletricas.htm>>. Acesso em: 7 nov. 2018.

COURY, Denis Vinicius. **Introdução aos Sistemas Elétricos de Potência**. Universidade de São Paulo, Escola de Engenharia de São Carlos. São Carlos/SP, 2012.

GÓMEZ-EXPÓSITO, Antonio; CONEJO, Antonio J.; CANIZARES, Claudio. **Sistemas de Energia Elétrica – Análise e operação**. Rio de Janeiro: LTC – Livros Técnicos e Científicos Editora Ltda, 2011. 554 p.



LEÃO, Ruth. **GTD – Geração, Transmissão e Distribuição de Energia Elétrica**. Apostila. Centro de Tecnologia/Departamento de Engenharia Elétrica-Universidade Federal do Ceará–UFC, Fortaleza, 2009.

MILLER, Robert H. **Operação de Sistemas de Potência**. McGraw-Hill, 1984.

MONTICELLI, Alcir José. **Fluxo de Carga em redes de energia elétrica**. São Paulo: E. Blucher, 1983. 166 p.

MONTICELLI, Alcir J. GARCIA, Ariovaldo. **Introdução a sistemas de energia elétrica**. Campinas, SP: UNICAMP, 2003. 251 p.

SALGADO, R. S.; **Introdução aos Sistemas de Energia Elétrica**. Apostila. Curso de Graduação em Engenharia Elétrica. Universidade Federal de Santa Catarina. Florianópolis, 2016.

SANTOS, D. L. N.; DAHER, R.; NAZARI, V. **Análise estática de sistema elétrico de potência utilizando o programa computacional Anarede**. 2013. 99 f. Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica) – Departamento Acadêmico de Eletrotécnica, Universidade Tecnológica Federal do Paraná, Curitiba, 2013.

SCHIOCHET, André Fernando. **Fluxo de Potência Multi-Objetivo: Abordagem utilizando uma estratégia de parametrização**. 2006. 7f. Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal de Santa Catarina.

STEVENSON, William D. **Elementos de análise de sistemas de potência**. São Paulo: McGraw-Hill do Brasil, 1978. 347 p.

## ANEXOS

## ANEXO A – RELATÓRIO DE SAÍDA DO PROGRAMA SIMPOT PARA O SISTEMA DE 2 BARRAS

**Relatório do cálculo do Fluxo de Potência pelo método de Newton Rapshon**

Número máximo de iterações especificado: 100

Erro máximo especificado: 1.0E-4

Número de barras PQ: 1

Número de barras PV: 0

Número de barras Vteta: 1

Dados das Barras						
Barra	Tipo Barra	V (pu)	Teta (rad)	P (pu)	Q (pu)	b (pu)
1.0000	Vteta	1.0112	0.0000	0.0000	0.0000	0.0000
2.0000	PQ	0.0000	0.0000	-1.0000	-0.0000	0.0000

Dados das Linhas					
Barra DE	Barra PARA	R (pu)	X (pu)	Bsh (pu)	a
1.0000	2.0000	0.0100	0.0500	0.0000	1.0000

Admitâncias das Linhas					
Barra DE	Barra PARA	G (pu)	B (pu)	Bsh (pu)	a
1.0000	2.0000	3.8462	-19.2308	0.0000	1.0000

Ybarra					
G + jB (pu)					
3.8462		-19.2308		-3.8462	19.2308
-3.8462		19.2308		3.8462	-19.2308

Mismatches de Potências		
Barra	Delta P	Delta Q
1.0000	0.0000	0.0000
2.0000	-0.9569	0.2154

Jacobiana	
19.4462	3.8031
-3.8892	19.0154

Deltas de Módulo e Ângulo das Tensões		
Barra	Delta V	Delta Teta
1.0000	0.0000	0.0000
2.0000	-0.0494	0.0012

Mismatches de Potências		
Barra	Delta P	Delta Q

1.0000	0.0000	0.0000
2.0000	-0.0040	-0.0240

<b>Jacobiana</b>	
19.6384	2.8560
-2.9269	19.2781

<b>Deltas de Módulo e Ângulo das Tensões</b>		
Barra	Delta V	Delta Teta
1.0000	0.0000	0.0000
2.0000	-0.0000	-0.0012

<b>Mismatches de Potências</b>		
Barra	Delta P	Delta Q
1.0000	0.0000	0.0000
2.0000	-0.0000	-0.0001

<b>Módulo e Ângulo das Tensões nas Barras</b>						
Iteração atual: 1				Maior Erro: 7.175564155603605E-5		
Barra	Tipo Barra	V (pu)	Teta (graus)	P (pu)	Q (pu)	b (pu)
1.0000	3.0000	1.0112	0.0000	0.0000	0.0000	0.0000
2.0000	1.0000	1.0000	-2.8343	-1.0000	-0.0000	0.0000

<b>Potências nas Barras</b>		
Barra	P (pu)	Q (pu)
1.0000	1.0100	0.0499
2.0000	-1.0000	0.0001

## APÊNDICES

### APÊNDICE A – ALGORITMO “DADOSDOSISTEMASEISBARRAS.M”

```

%***** Entrada de Dados *****%
clear all
Sbase = 100;
NUMBER = 6;           % Número de barras
NUMLIN = 11;          % Número de linhas
e=0.001;              % Tolerância/erro

%***** Matriz de Dados de Barras *****%

% *** OBSERVAÇÕES:
% - A barra de referência deve ser sempre a barra (Bar): 1
% - Barra tipo (T) 0: PQ
%                   1: PV
%                   2: Vtheta

%   Bar T V(p.u.) Ang(°) Pg(MW) Qg(Mvar) Pl(MW) Ql(Mvar) Shunt(Mvar) Qgmin(Mvar) Qgmax(Mvar) Pgmax(MW)
DBAR=[ 1 2 1.050 0.0 0.0 0.0 0.0 0.0 0.0 0.0 -100 100 200 ;
       2 1 1.050 0.0 50 0.0 0.0 0.0 0.0 0.0 -100 70 150 ;
       3 1 1.070 0.0 60 0.0 0.0 0.0 0.0 0.0 -100 70 180 ;
       4 0 1.000 0.0 0.0 0.0 70 70 0.0 0.0 0.0 0.0 ;
       5 0 1.000 0.0 0.0 0.0 70 70 0.0 0.0 0.0 0.0 ;
       6 0 1.070 0.0 0.0 0.0 70 70 0.0 0.0 0.0 0.0 ];

%***** Matriz de dados de linha *****%

%   Origem Destino Res(p.u) Reat(p.u.) Suscept(p.u.)
DLIN = [1 2 0.1 0.2 0.04;
        1 4 0.05 0.2 0.04;
        1 5 0.08 0.3 0.06;
        2 3 0.05 0.25 0.06;
        2 4 0.05 0.1 0.02;
        2 5 0.1 0.3 0.04;
        2 6 0.07 0.2 0.05;
        3 5 0.12 0.26 0.05;
        3 6 0.02 0.1 0.02;
        4 5 0.2 0.4 0.08;
        5 6 0.1 0.3 0.06];

% Rotina para dividir os valores de susceptância por 2, conforme o modelo
% pi de linhas de transmissão
for i = 1:NUMLIN
    DLIN(i,5) = DLIN(i,5)/2;
end

% Rotina para colocar os valores de potência e shunt em pu
for i = 1:NUMBER
    for k = 5:12
        DBAR(i,k) = DBAR(i,k)/Sbase;
    end
end

% Rotina para colocar o ângulo em radianos
for i = 1:NUMBER
    DBAR(i,4) = DBAR(i,4)*pi/180;
end

DBARoriginal=DBAR; %Variável que guardará o DBAR inalterado

```

## APÊNDICE B – ALGORITMO “FLOBA.M”

```

%----- FLOBA -----%
% Estudo do Fluxo de Carga, pelo método de Newton Raphson, considerando
% os limites de reativos das máquinas síncronas do sistema.
% Alunos: Alice Fey, Bruno Soares e Olivia Nichele.
% Prof.(a): Andréa Lucia Costa.
% Trabalho de conclusão do curso de Eng. Elétrica - UTFPR - 2019/1
%-----%

%----- OBSERVAÇÕES -----%
% Antes de ser executado o FLOBA, deve ser executado o arquivo .m com os
% dados de entrada do sistema, seguindo o modelo "DadosDoSistema.m".
% Os dados devem ser inseridos conforme as grandezas indicadas no modelo,
% respeitando também os tipos de barras.

%----- Variáveis utilizadas -----%
aviso=0;          % Variável que indica convergência (0 = não convergência)
                  % (1 = convergência)

tolP=e;
tolQ=e;
iteracoes=0;    % Variável que indica o número de iterações efetuadas.
BarraCritica=0; % Variável que indica qual barra do sistema mais excedeu
                  % o limite de geração de reativos da máquina síncrona.

%-----%

% Função para indicar qual o tipo da barra (PQ, PV ou Vtheta)
tip = DBAR(:,2);

%----- MONTAGEM DA MATRIZ Y(admitâncias)-----%

Y = zeros(NUMBAR,NUMBAR);          % Criação da matriz Y preenchida com zeros

for t = 1:NUMLIN
    ORIGEM = DLIN(t,1);
    DESTINO = DLIN(t,2);
    RESISTENCIA = DLIN(t,3);
    REATANCIA = DLIN(t,4);
%Cálculo das diagonais secundárias:
    Y(ORIGEM,DESTINO) = -1/(RESISTENCIA + j*REATANCIA);
%Cálculo das diagonais secundárias:
    Y(DESTINO,ORIGEM) = Y(ORIGEM,DESTINO);
%Cálculo da diagonal principal:
    Y(ORIGEM,ORIGEM) = (1/(RESISTENCIA + j*REATANCIA)) + j*DLIN(t,5) + Y(ORIGEM,ORIGEM);
%Cálculo da diagonal principal:
    Y(DESTINO,DESTINO) = (1/(RESISTENCIA + j*REATANCIA)) + j*DLIN(t,5) + Y(DESTINO,DESTINO);
end

% Montagem da matriz S(potências complexas)
Pesp = zeros(NUMBAR,1);
Qesp = zeros(NUMBAR,1);

% Cálculo dos valores iniciais de injeção de pot. ativa e de pot. reativa
for i = 1:NUMBAR
    Pesp(i) = DBAR(i,5) - DBAR(i,7);          % Pesp = P(geração) - P(carga)
end
for i = 1:NUMBAR
    Qesp(i) = DBAR(i,6) - DBAR(i,8);          % Qesp = Q(geração) - Q(carga)
end

```

```

% Criação das matrizes G e B
G=zeros(NUMBAR,NUMBAR); B=zeros(NUMBAR,NUMBAR);
G(:, :) = real(Y(:, :)); B(:, :) = imag(Y(:, :));

% Determinação do numero de barras PQ e barras PV
npq=0; npv=0;
for i=1:NUMBAR
    if tip(i,1)==0
        npq = npq+1;
    end
end

for i=1:NUMBAR
    if tip(i,1) == 1
        npv = npv+1;
    end
end
npqp= npq+npv;

% Montagem do vetor V(tensões das barras) e theta(angulos das tensões das
% barras)
V = DBAR(:,3); theta = DBAR(:,4);

%-----%
% Rotina que será executada até que a solução do Fluxo de Carga apresente
% convergência

while aviso==0

%----- Cálculo das Injeções de Potência -----%

% Cálculo da injeção de potência ativa (Pi)
P=zeros(NUMBAR,1);
for k=2:NUMBAR
    som = 0;
    for m=1:NUMBAR
        som = som + V(m,1)*(G(k,m)*cos(theta(k,1)-theta(m,1))+B(k,m)*sin(theta(k,1)-theta(m,1)));
    end
    P(k,1)=V(k,1)*som;
end

% Cálculo da injeção de potência reativa (Qi)
Q=zeros(NUMBAR,1);
for k=1:NUMBAR
    som = 0;
    for m=1:NUMBAR
        som = som + V(m,1)*(G(k,m)*sin(theta(k,1)-theta(m,1))-B(k,m)*cos(theta(k,1)-theta(m,1)));
    end
    Q(k,1)=V(k,1)*som;
end

```

```

%----- Teste dos resíduos/mismatches de potência [g(x)] -----%

% Comparar o valor de g(x) com a tolerância "e". Se |g(x)|<=e, a solução
% procurada está dentro da faixa de tolerância, portanto tem-se o estado
% da rede, caso contrário o processo iterativo deve continuar.

i=0;
for k=1:NUMBAR
    if tip(k,1) ~= 2
        i=i+1;
        dP(i,1) = Pesp(k,1) - P(k,1); % Resíduo de potência ativa
        deltaP(i,1) = abs(dP(i,1));
    end
end

i=0;
for k=1:NUMBAR
    if tip(k,1) == 0
        i=i+1;
        dQ(i,1) = Qesp(k,1) - Q(k,1); % Resíduo de potência reativa
        deltaQ(i,1) = abs(dQ(i,1));
    end
end

for i=1:npqpv
    residuo(i,1)=dP(i,1);
end
for n=1:npq
    i=npqpv+n;
    residuo(i,1)=dQ(n,1);
end

% Teste da convergência

for i=1:npqpv
    if deltaP(i,1)>tolP
        aviso=0;
    else aviso=1;
    end
end

if aviso==1
    for i=1:npq
        if deltaQ(i,1)>tolQ
            aviso=0;
        else aviso=1;
        end
    end
end

%-----%
% Caso a convergência não tenha sido alcançada, será efetuada a rotina para
% calcular a matriz Jacobiana e encontrar os valores de correção para as
% tensões complexas nas barras.

if aviso==0

%----- Cálculo da matriz Jacobiana -----%
% J = [ H  N
%       M  L ]

```

```

% Montagem da matriz H
H=zeros (NUMBAR, NUMBAR);

% Cálculo de H(k,k) - valores da diagonal principal
for k=1:NUMBAR
    som=0;
    for m=1:NUMBAR
        som = som + V(m,1) * (G(k,m) * sin(theta(k,1) - theta(m,1)) -
B(k,m) * cos(theta(k,1) - theta(m,1)));
    end
    H(k,k) = -B(k,k) * (V(k,1)^2) - V(k,1) * som;
end

% Cálculo de H(k,m)
for k=1:NUMBAR
    for m=(k+1):NUMBAR
        H(k,m) = V(k,1) * V(m,1) * (G(k,m) * sin(theta(k,1) -
B(k,m) * cos(theta(k,1) - theta(m,1)));
    end
end

% Cálculo de H(m,k)
for k=1:NUMBAR
    for m=(k+1):NUMBAR
        H(m,k) = -V(k,1) * V(m,1) * (G(k,m) * sin(theta(k,1) -
theta(m,1)) + B(k,m) * cos(theta(k,1) - theta(m,1)));
    end
end

% Montagem da matriz N
N=zeros (NUMBAR, NUMBAR);

% Cálculo de N(k,k) - diagonal principal
for k=1:NUMBAR
    som=0;
    for m=1:NUMBAR
        som = som + V(m,1) * (G(k,m) * cos(theta(k,1) -
theta(m,1)) + B(k,m) * sin(theta(k,1) - theta(m,1)));
    end
    N(k,k) = G(k,k) * (V(k,1)) + som;
end

% Cálculo de N(k,m)
for k=1:NUMBAR
    for m=(k+1):NUMBAR
        N(k,m) = V(k,1) * (G(k,m) * cos(theta(k,1) -
theta(m,1)) + B(k,m) * sin(theta(k,1) - theta(m,1)));
    end
end

% Cálculo de N(m,k)
for k=1:NUMBAR
    for m=(k+1):NUMBAR
        N(m,k) = V(m,1) * (G(k,m) * cos(theta(k,1) - theta(m,1)) -
B(k,m) * sin(theta(k,1) - theta(m,1)));
    end
end

```



```

% Montagem da matriz M
M=zeros (NUMBAR, NUMBAR);

% Cálculo de M(k,k)- diagonal principal
for k=1:NUMBAR
    som=0;
    for m=1:NUMBAR
        som = som + V(m,1) * (G(k,m) * cos(theta(k,1)-theta(m,1)) + B(k,m) * sin(theta(k,1)-theta(m,1)));
    end
    M(k,k) = -G(k,k) * (V(k,1) ^2) + V(k,1) * som;
end

% Cálculo de M(k,m)
for k=1:NUMBAR
    for m= (k+1) : NUMBAR
        M(k,m) = -V(k,1) * V(m,1) * (G(k,m) * cos(theta(k,1)-theta(m,1)) + B(k,m) * sin(theta(k,1)-theta(m,1)));
    end
end

% Cálculo de M(m,k)
for k=1:NUMBAR
    for m= (k+1) : NUMBAR
        M(m,k) = -V(k,1) * V(m,1) * (G(k,m) * cos(theta(k,1)-theta(m,1)) - B(k,m) * sin(theta(k,1)-theta(m,1)));
    end
end

% Montagem da matriz L
L=zeros (NUMBAR, NUMBAR);

% Cálculo de L(k,k)- diagonal principal
for k=1:NUMBAR
    som=0;
    for m=1:NUMBAR
        som = som + V(m,1) * (G(k,m) * sin(theta(k,1)-theta(m,1)) - B(k,m) * cos(theta(k,1)-theta(m,1)));
    end
    L(k,k) = - (B(k,k) * V(k,1)) + som;
end

% Cálculo de L(k,m)
for k=1:NUMBAR
    for m= (k+1) : NUMBAR
        L(k,m) = V(k,1) * (G(k,m) * sin(theta(k,1)-theta(m,1)) - B(k,m) * cos(theta(k,1)-theta(m,1)));
    end
end

% Cálculo de L(m,k)

for k=1:NUMBAR
    for m= (k+1) : NUMBAR
        L(m,k) = -V(m,1) * (G(k,m) * sin(theta(k,1)-theta(m,1)) - B(k,m) * cos(theta(k,1)-theta(m,1)));
    end
end

```

```

%----- Montando a matriz Jacobiana (J) reduzida -----%

% Inserindo os valores de H na matriz Jacobiana
linha=0;
coluna=0;
for i=2:NUMBAR
    linha=linha+1;
    coluna=0;
    for n=2:NUMBAR
        coluna=coluna+1;
        J(linha,coluna)=H(i,n);
    end
end

% Inserindo os valores de N na matriz Jacobiana
linha1=0;
coluna1=0;
for i=2:NUMBAR
    linha1=linha1+1;
    coluna1=coluna;
    for n=2:NUMBAR
        if tip(n)==0
            coluna1=coluna1+1;
            J(linha1,coluna1)=N(i,n);
        end
    end
end

% Inserindo os valores de M na matriz Jacobiana
linha2=0;
coluna2=0;
for i=2:NUMBAR
    if tip(i)==0
        linha2=linha+1;
        for n=2:NUMBAR
            coluna2=coluna2+1;
            J(linha2,coluna2)=M(i,n);
        end
        linha=linha2;
    end
    coluna2=0;
end

% Inserindo os valores de L na matriz Jacobiana
for i=2:NUMBAR
    if tip(i)==0
        linha2=linha1+1;
        coluna1=coluna;
        for n=2:NUMBAR
            if tip(n)==0
                coluna1=coluna1+1;
                J(linha2,coluna1)=L(i,n);
            end
        end
        linha1=linha2;
    end
end

```

```

%----- Correção dos valores de tensões complexas -----%

% Calculando a matriz J inversa e o vetor de correções (deltaX)

Ji=inv(J);
deltaX=Ji*residuo;

% Criando um vetor X com os valores iniciais de tensões e ângulos
% sendo X=[theta          (barras PQ e PV)
%          V ]          (somente barras PQ)

i=0;
for k=1:NUMBAR
    if tip(k,1) ~= 2
        i=i+1;
        X(i,1) = theta(k,1);
    end
end

i=npqpv;
for k=1:NUMBAR
    if tip(k,1) == 0
        i=i+1;
        X(i,1) = V(k,1);
    end
end

% Atualizando o vetor X com as correções das tensões complexas

Xnovo=deltaX+X;

% Inserindo os valores corrigidos dos ângulos theta no vetor "theta"
for i=1:npqpv
    theta(i+1,1)=Xnovo(i,1);
end

% Inserindo os valores corrigidos de magnitude de tensão no vetor "V"
n=npqpv;
for i=1:NUMBAR
    if tip(i,1)==0
        n=n+1;
        V(i,1)=Xnovo(n,1);
    end
end

end

iteracoes=iteracoes+1;

%Rotina para limitar o número de iterações

if iteracoes>50
    aviso=2;
end

end

```

```

%----- Rotina para mostrar o estado da rede -----%

fprintf('\n-----');
fprintf('-----\n                                     ');
fprintf('RELATÓRIO FLOBA                                     \n-----');
fprintf('-----');
fprintf('-----\n\nNúmero de iterações realizadas = %i',iteracoes);

Barra=zeros(k,1);          %Criando um vetor com o número de barras do sistema
for k=1:NUMBAR
    Barra(k,1)=k;
end

% Equação para passar os valores dos ângulos de radianos para graus
theta=theta*180/pi;

EstadoDaRede=table(Barra,V,theta)%Mostrando uma tabela com o estado da rede

%-----
%----- Verificação da geração de reativos nas barras -----%

for k=1:NUMBAR
    Qg(k,1)=Q(k,1)+DBAR(k,8); % Vetor que guarda os valores de Q gerados
end

limitereativo=zeros(NUMBAR,1); % Vetor que sinaliza qual barra atingiu o
                                % limite de geração de reativos (+1 ou -1.
QgDesvio=zeros(NUMBAR,1);      % Variável que armazena o desvio entre o
                                % Qg máximo e o atual valor de Qg.

% Cálculo dos desvios de geração de reativos para as barras que ultrapassa-
% ram os limites (Qgmax ou Qgmin de DBAR)

fprintf('\n-----');
fprintf('-----\n\nMonitoramento da geração/consumo de');
fprintf(' reativos das máquinas síncronas do sistema:\n\n');

for k=1:NUMBAR
    if tip(k)~=0
        if Qg(k,1)>DBAR(k,11) % Verificação do excedente de Qgmax
            limitereativo(k,1)=1;
            QgDesvio(k,1)=abs(DBAR(k,11)-Qg(k,1));
            QgmaxDesvio=QgDesvio(k,1);
            fprintf(' - A barra %i excedeu em %fpu ',k,QgmaxDesvio);
            fprintf('o limite máximo de geração de reativos. \n');
        end
        if Qg(k,1)<DBAR(k,10) % Verificação do excedente de Qgmin
            limitereativo(k,1)=-1;
            QgDesvio(k,1)=abs(DBAR(k,10)-Qg(k,1));
            QgminDesvio=QgDesvio(k,1);
            fprintf(' - A barra %i excedeu em %fpu ',k,QgminDesvio);
            fprintf('o limite máximo de consumo de reativos. \n');
        end
    end
end
end
end

```

```

QgDesvioMax=max(QgDesvio);
if QgDesvioMax~=0
    for k=1:NUMBAR
        if QgDesvio(k,1)==QgDesvioMax
            fprintf('\nBarra que apresentou o maior excedente de ');
            fprintf('geração/consumo de reativos = %i\n',k);
            BarraCritica=k;
        end
    end
    fprintf('\n-----');
    fprintf('-----\n');
    fprintf('\n## Digite o número de alguma das barras PV do sistema');
    fprintf(' (exceto a barra 1) que o\n## FLOBA irá alterar os ');
    fprintf('parâmetros, para que então seja recalculado o fluxo de');
    fprintf('\n## potência considerando o limite da máquina síncrona.');
```

comando=input(' Ou aperte 0 para finali-\n## zar a execução:');

```

if comando~=0
    if comando~=1
        if comando<=NUMBAR
            if limitereativo(comando,1)~=0
                BarraModificada=comando; % Atribui à variável a barra escolhida
                DBAR(BarraModificada,2)=0; % Muda o tipo da barra de PV para PQ
                DBAR(BarraModificada,3)=1; % Altera tensão para 1 pu
                DBAR(BarraModificada,4)=0; % Altera ângulo theta para 0 graus
                % Realoca o valor de Pg para Pl:
                DBAR(BarraModificada,7)=-DBAR(BarraModificada,5);
                if limitereativo(BarraModificada,1)==1
                    % Realoca o valor de Qgmax para Ql:
                    DBAR(BarraModificada,8)=-DBAR(BarraModificada,11);
                else
                    % Realoca o valor de Qgmin para Ql:
                    DBAR(BarraModificada,8)=-DBAR(BarraModificada,10);
                end
                % Modifica para zero os valores que não serão mais utilizados:
                DBAR(BarraModificada,5)=0;
                DBAR(BarraModificada,6)=0;
                DBAR(BarraModificada,10)=0;
                DBAR(BarraModificada,11)=0;
                run FLOBA
            else fprintf('\n!! Esta barra não apresentou violação na ');
                fprintf('geração/consumo de reativos, rode o\n');
                fprintf('!!! FLOBA novamente. \n\n');
            end % end para caso o comando tenha sido de uma barra que não
                % violou os limites de reativos
            else fprintf(' \n!! Este número de barra não existe,');
                fprintf(' rode o FLOBA novamente. \n\n');
            end % end para caso o comando tenha sido um valor de barra
                % inexistente
            else fprintf('\n!! A barra de referência não pode');
                fprintf(' ser alterada, rode o FLOBA novamente. \n\n');
            end % end para caso o comando tenha sido igual a 1
        end % end para caso o comando tenha sido igual à zero
    else %else para caso não tenha ocorrido violação nos limites
        fprintf('## Não houve violação no limite de geração/consumo de ');
        fprintf('reativos das máquinas\n## síncronas do sistema! \n :)\n');
    end %end para o if QgDesvioMax~=0
end

```