

OBJETIVOS

Este exercício programa tem por objetivo apresentar um algoritmo para localização de faltas em redes primárias de distribuição e permitir que o aluno possa desenvolvê-lo e implementá-lo em uma rotina computacional. Este algoritmo se baseia no artigo Manassero, G., Di Santo, S.G. and Souto, L., 2017. Heuristic method for fault location in distribution feeders with the presence of distributed generation publicado no IEEE Transactions on Smart Grid, 8(6), pp.2849-2858. <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7536107>.

INTRODUÇÃO

Os alimentadores primários estão sujeitos a faltas provocadas por uma variedade de situações, tais como, condições atmosféricas adversas, falhas em equipamentos, acidentes, etc. Os defeitos nestes alimentadores resultam em um grande número de problemas relacionados à confiabilidade do serviço e à qualidade da energia entregue aos consumidores.

Atualmente, as concessionárias efetuam a localização das faltas através de reclamações dos consumidores afetados. Quando há um defeito permanente, os técnicos do centro de operação da distribuição (COD) identificam o alimentador faltoso e enviam uma equipe de manutenção que o percorre, executando manobras nos dispositivos de proteção na tentativa de isolar e identificar o ponto de ocorrência do defeito.

Quando a falta é transitória esta localização se torna mais difícil, visto que esse tipo de defeito normalmente não resulta em algum fusível queimado, e não existem evidências que possam determinar sua causa e localização.

Esse panorama torna interessante para as concessionárias dispor de um sistema automatizado que possibilite localizar com rapidez qualquer falta nos seus alimentadores primários, minimizando desta forma, o tempo de deslocamento da equipe de manutenção e o restabelecimento do fornecimento de energia elétrica. Sendo assim, a imediata localização do ponto de ocorrência do defeito proporciona uma série de benefícios para os consumidores e para as concessionárias de distribuição de energia elétrica.

1 Algoritmo para localização de faltas em redes de distribuição

O algoritmo de localização de faltas a ser implementado é baseado na análise nodal de redes elétricas de distribuição e na investigação de todos os trechos de rede para determinação do ponto de ocorrência da falta e da sua resistência elétrica. Para tanto, considere o alimentador de distribuição ilustrado na Figura 1.

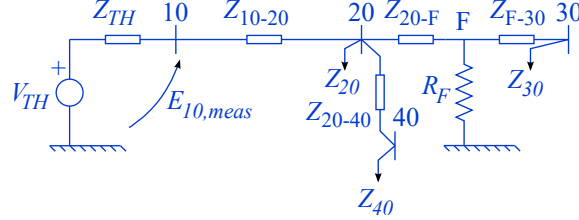


Figura 1: Alimentador de distribuição

Nessa figura está ilustrado um alimentador monofásico que possui quatro nós e três trechos. Esse alimentador é conectado ao sistema elétrico por meio do nó da subestação (nó 10) e o sistema elétrico pode ser representado por meio de um equivalente de Thévenin, onde $\dot{V}_{TH} = \frac{13800}{\sqrt{3}} \angle 0^\circ$ [V] e $\bar{Z}_{TH} = 0,033070 + j \cdot 0,187547$ [Ω] (nesse nó é possível medir os fasores de tensão e corrente $\dot{E}_{10,meas}$ e $\dot{I}_{10-20,meas}$, respectivamente).

O alimentador ilustrado pode ser descrito textualmente por meio de uma matriz de topologia que apresenta a conectividade entre os trechos, suas impedâncias, em ohms por unidade de comprimento, e as distâncias entre eles, em unidades de comprimento; e uma matriz de cargas que descreve cada uma das cargas, em termos de potência ativa e fator de potência, e o nó onde essas estão conectadas. As tabelas 1 e 2 descrevem esse conceito.

Tabela 1: Topologia - descrição pai-filho

Pai (barra inicial)	Filho (barra final)	Impedância do trecho [$\frac{\Omega}{m}$]	Distância [m]
10	20	$\bar{z}_{10-20} = \left(\frac{0,1830856}{1000} + j \cdot \frac{0,8662534}{1000} \right)$	$\ell_{10-20} = 50$
20	30	$\bar{z}_{20-30} = \left(\frac{0,2241236}{1000} + j \cdot \frac{0,9144412}{1000} \right)$	$\ell_{20-30} = 150$
20	40	$\bar{z}_{20-40} = \left(\frac{0,2154789}{1000} + j \cdot \frac{0,7567231}{1000} \right)$	$\ell_{20-40} = 80$

Tabela 2: Cargas

Barra	Potência ativa da carga [kW]	Fator de potência da carga
20	220	0,87 (indutivo)
30	130	0,95 (indutivo)
40	170	0,98 (indutivo)

Conforme mencionado, o algoritmo de localização investiga todos os trechos do alimentador defeituoso, estimando a distância da falta aos nós que delimitam o trecho investigado e também a resistência de falta. Essa investigação consiste em encontrar o par distância da falta e resistência da falta (x, R_F) que, quando utilizado na solução por análise nodal, torna a tensão calculada no nó da subestação igual à tensão medida. Nesse contexto, considerando que o trecho 20-30 esteja sob investigação, pode-se escrever uma matriz de admitâncias nodais, que é simétrica, conforme (1)

$$\underbrace{\begin{bmatrix} 10 & 20 & 30 & 40 & F \\ 10 & \bar{Y}_{10-10} & \bar{Y}_{10-20} & 0 & 0 & 0 \\ 20 & \bar{Y}_{10-20} & \bar{Y}_{20-20} & 0 & \bar{Y}_{20-40} & \bar{Y}_{20-F} \\ 30 & 0 & 0 & \bar{Y}_{30-30} & 0 & \bar{Y}_{30-F} \\ 40 & 0 & 0 & \bar{Y}_{40-40} & 0 & \bar{Y}_{40-F} \\ F & 0 & 0 & 0 & \bar{Y}_{F-40} & \bar{Y}_{F-F} \end{bmatrix}}_{[Y_{bus}]} \times \underbrace{\begin{bmatrix} \dot{E}_{10,calc} \\ \dot{E}_{20,calc} \\ \dot{E}_{30,calc} \\ \dot{E}_{40,calc} \\ \dot{E}_{F,calc} \end{bmatrix}}_{[E_{bus}]} = \underbrace{\begin{bmatrix} \dot{I}_{10} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{[I_{bus}]} \quad (1)$$

Onde:

$$\begin{aligned}\bar{Y}_{10-10} &= \frac{1}{Z_{TH}} + \frac{1}{\ell_{10-20} \cdot \bar{z}_{10-20}} + \frac{1}{Z_{10}}; \\ \bar{Y}_{10-20} &= -\frac{1}{\ell_{10-20} \cdot \bar{z}_{10-20}}; \\ \bar{Y}_{20-20} &= \frac{1}{\ell_{10-20} \cdot \bar{z}_{10-20}} + \frac{1}{Z_{20}} + \frac{1}{x \cdot \bar{z}_{20-30}} + \frac{1}{\ell_{20-40} \cdot \bar{z}_{20-40}}; \\ \bar{Y}_{20-40} &= -\frac{1}{\ell_{20-40} \cdot \bar{z}_{20-40}}; \\ \bar{Y}_{20-F} &= -\frac{1}{x \cdot \bar{z}_{20-30}}; \\ \bar{Y}_{30-30} &= \frac{1}{(\ell_{20-30}-x) \cdot \bar{z}_{20-30}} + \frac{1}{Z_{30}}; \\ \bar{Y}_{30-F} &= -\frac{1}{(\ell_{20-30}-x) \cdot \bar{z}_{20-30}}; \\ \bar{Y}_{40-40} &= \frac{1}{\ell_{20-40} \cdot \bar{z}_{20-40}} + \frac{1}{Z_{40}}; \text{ e} \\ \bar{Y}_{F-F} &= \frac{1}{x \cdot \bar{z}_{20-30}} + \frac{1}{(\ell_{20-30}-x) \cdot \bar{z}_{20-30}} + \frac{1}{R_F}.\end{aligned}$$

A partir dessa equação pode-se calcular as tensões nas barras, como em:

$$\begin{bmatrix} \dot{E}_{10,calc} \\ \dot{E}_{20,calc} \\ \dot{E}_{30,calc} \\ \dot{E}_{40,calc} \\ \dot{E}_{F,calc} \end{bmatrix} = [Y_{bus}]^{-1} \times \begin{bmatrix} \dot{I}_{10} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (2)$$

A investigação do trecho 20-30 consiste em se determinar o par distância da falta e resistência da falta (x, R_F) de modo encontrar o mínimo da função descrita por (3).

$$f(x, R_F) = \frac{|\dot{E}_{10,calc} - \dot{E}_{10,meas}|}{|\dot{E}_{10,meas}|} \quad (3)$$

Considerando que os limites de busca se encontram no intervalo:

$$\begin{cases} 0 \leq x \leq \ell_{20-30} \\ 0 \leq R_F \leq \max\{R_F\} \end{cases} \quad (4)$$

Onde:

$\dot{E}_{10,calc}$ é a tensão no nó 10, calculada por meio de (2);

$\dot{E}_{10,meas}$ é a tensão no nó 10, medida na subestação;

$\max\{R_F\}$ é o máximo valor que a resistência de falta pode assumir e, normalmente, é desconhecido.

Normalmente emprega-se algum método de otimização de funções multivariáveis para a determinação de ambos os valores. No entanto, neste exercício programa solicita-se que o aluno faça a busca por força bruta, visto que o problema é de pequenas dimensões e essa abordagem não traz prejuízo à precisão do método e ao tempo de convergência para encontrar a solução.

2 Montagem da matriz de admitâncias nodais

Em teoria de circuitos elétricos, a matriz de admitâncias nodais é obtida quando se aplica, na solução de um circuito elétrico, o método dos nós. O método dos nós vem da aplicação da primeira Lei de Kirchhoff (somatório das correntes que entram e saem de um nó deve ser igual a zero) aos nós do circuito.

Da aplicação do método dos nós à um circuito, é possível relacionar as correntes que entram (ou saem) dos nós com a tensão destes em relação à um nó escolhido como referência. Assim, (5) é obtida.

$$[I_{\text{nós}}] = [Y_{\text{nós}}] \times [V_{\text{nós}}] \quad (5)$$

De (5) é possível determinar as tensões, em relação ao nó eleito como referência, em cada nó do circuito, bastando inverter a matriz de admitâncias de nós e multiplicar pelo vetor de correntes que entram (ou saem) deles.

Existem várias formas de se montar a matriz de admitâncias nodais, e uma delas é utilizando a teoria de grafos orientados.

A topologia de uma rede pode ser representada por um grafo orientado, onde os ramos da rede são representados por um segmento de reta com uma seta, que indica a direção da corrente no ramo (deliberadamente escolhida), e são interconectados por nós. A Figura 2 ilustra um grafo orientado do circuito simples mostrado na Figura 3.

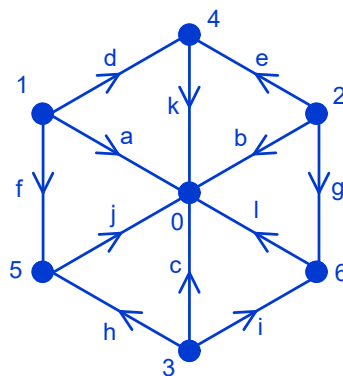


Figura 2: Grafo orientado

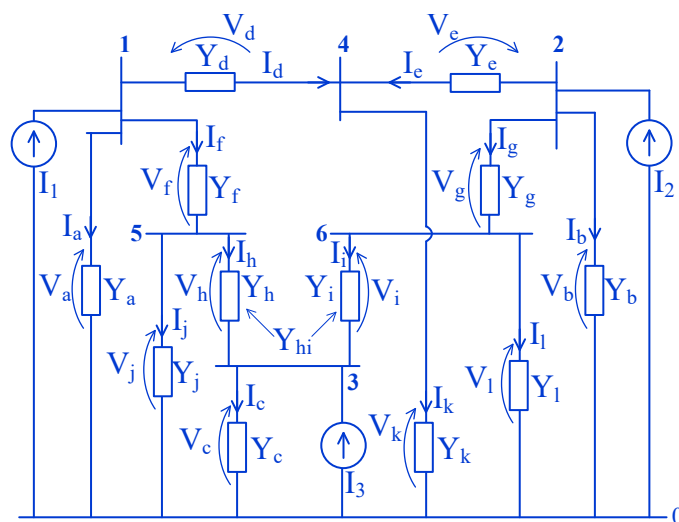


Figura 3: Circuito de exemplo

A fim de representar a topologia da rede matematicamente, uma matriz pode ser construída a partir do grafo orientado seguindo a regra descrita por (6) para definir os coeficientes da matriz.

$$q_{ij} = \begin{cases} 0 & \text{se o ramo } i \text{ não está conectado ao nó } j \\ 1 & \text{se a corrente através do ramo } i \text{ sai do nó } j \\ -1 & \text{se a corrente através do ramo } i \text{ chega do nó } j \end{cases} \quad (6)$$

Essa matriz é conhecida como matriz de incidências, e esse nome vem do fato de que quando um ramo se conecta a um nó eles são ditos incidentes.

Em (7) é mostrada a matriz de incidências construída para o circuito ilustrado na Figura 3. As linhas da matriz de incidências correspondem aos ramos e as colunas os nós. Na matriz de incidências o efeito das mútuas não é contemplado, visto que tal matriz representa as correntes nos ramos.

$$Q = \begin{array}{c|cccccc} \text{trechos} \backslash \text{nós} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline a & 1 & 0 & 0 & 0 & 0 & 0 \\ b & 0 & 1 & 0 & 0 & 0 & 0 \\ c & 0 & 0 & 1 & 0 & 0 & 0 \\ d & 1 & 0 & 0 & -1 & 0 & 0 \\ e & 0 & 1 & 0 & -1 & 0 & 0 \\ f & 1 & 0 & 0 & 0 & -1 & 0 \\ g & 0 & 1 & 0 & 0 & 0 & -1 \\ h & 0 & 0 & -1 & 0 & 1 & 0 \\ i & 0 & 0 & -1 & 0 & 0 & 1 \\ j & 0 & 0 & 0 & 0 & 1 & 0 \\ k & 0 & 0 & 0 & 1 & 0 & 0 \\ l & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \quad (7)$$

A tensão sobre o ramo, bem como a corrente através deste, definem sua admitância primitiva. No caso dos ramos mutuamente acoplados, que é o caso das linhas de transmissão, suas admitâncias primitivas são compostas por admitâncias próprias e mútuas.

Portanto, a matriz de admitâncias primitivas de toda a rede é construída inserindo-se em sua diagonal principal a admitância primitiva de cada ramo, como mostrado em (8). Observe a representação da admitância mútua entre os ramos h e i .

$$Y_{pr} = \begin{array}{c|cccccccccccc} \text{trechos} \backslash \text{trechos} & a & b & c & d & e & f & g & h & i & j & k & l \\ \hline a & Y_a & & & & & & & & & & & \\ b & & Y_b & & & & & & & & & & \\ c & & & Y_c & & & & & & & & & \\ d & & & & Y_d & & & & & & & & \\ e & & & & & Y_e & & & & & & & \\ f & & & & & & Y_f & & & & & & \\ g & & & & & & & Y_g & & & & & \\ h & & & & & & & & Y_h & -Y_{hi} & & & \\ i & & & & & & & & -Y_{hi} & Y_i & & & \\ j & & & & & & & & & & Y_j & & \\ k & & & & & & & & & & & Y_k & \\ l & & & & & & & & & & & & Y_l \end{array} \quad (8)$$

Assim, a matriz de admitâncias nodais é obtida por meio de (9), onde t representa a transposta da matriz de incidências.

$$[Y_{\text{nós}}] = [Q]^t \times [Y_{pr}] \times [Q] \quad (9)$$

No caso de a rede ser trifásica, os elementos da matriz de admitâncias primitivas serão matrizes e, neste caso, na matriz de incidências o número “1” será representado pela matriz identidade e o “0” por uma matriz de zeros. Todas as matrizes serão de ordem três.

3 Exemplo de código-fonte para o alimentador apresentado na figura 1

Nesta seção são apresentados os dois arquivos que contém a base de dados do problema (arquivo <TOP000.m> e arquivo <CAR000.m>, com os dados do alimentador e das cargas conectadas a ele). Esses arquivos são entrada para o arquivo <main_program.m>, que também está apresentado nesta seção, assim como o arquivo com os dados de dez eventos de curto-circuito (tensões medidas nas subestações para cada um dos eventos), que é denominado <VOL000.m>.

A rotina de localização de faltas está contida no arquivo <main_program.m> que fornece, como saída um arquivo <.CSV> que contém os valores de x e R_F para cada trecho investigado, em cada caso de simulação. Esse arquivo é denominado <OUT000.csv> e está apresentado ao final desta seção.

3.1 Matriz de topologia - arquivo <TOP000.m>

A matriz de topologia possui os seguintes dados:

- Coluna 1: nó pai do trecho do alimentador (atenção: o nó da subestação não é “filho” de ninguém);
- Coluna 2: nó filho do trecho do alimentador;
- Coluna 3: distância, em [m], entre nó pai e nó filho do trecho do alimentador;
- Coluna 4: resistência própria da fase A (raa) do trecho do alimentador, em [ohms/m];
- Coluna 5: reatância própria da fase A (xaa) do trecho do alimentador, em [ohms/m];
- Coluna 6: resistência mútua entre a fase A e a fase B (rab) do trecho do alimentador, em [ohms/m];
- Coluna 7: reatância mútua entre a fase A e a fase B (xab) do trecho do alimentador, em [ohms/m];
- Coluna 8: resistência mútua entre a fase A e a fase C (rac) do trecho do alimentador, em [ohms/m];
- Coluna 9: reatância mútua entre a fase A e a fase C (xac) do trecho do alimentador, em [ohms/m];
- Coluna 10: resistência mútua entre a fase B e a fase A (rba) do trecho do alimentador, em [ohms/m];
- Coluna 11: reatância mútua entre a fase B e a fase A (xba) do trecho do alimentador, em [ohms/m];
- Coluna 12: resistência própria da fase B (rbb) do trecho do alimentador, em [ohms/m];
- Coluna 13: reatância própria da fase B (xbb) do trecho do alimentador, em [ohms/m];
- Coluna 14: resistência mútua entre a fase B e a fase C (rbc) do trecho do alimentador, em [ohms/m];
- Coluna 15: reatância mútua entre a fase B e a fase C (xbc) do trecho do alimentador, em [ohms/m];
- Coluna 16: resistência mútua entre a fase C e a fase A (rca) do trecho do alimentador, em [ohms/m];
- Coluna 17: reatância mútua entre a fase C e a fase A (xca) do trecho do alimentador, em [ohms/m];
- Coluna 18: resistência mútua entre a fase C e a fase B (rcb) do trecho do alimentador, em [ohms/m];
- Coluna 19: reatância mútua entre a fase C e a fase B (xcb) do trecho do alimentador, em [ohms/m];
- Coluna 20: resistência própria da fase C (rcc) do trecho do alimentador, em [ohms/m]; e
- Coluna 21: reatância própria da fase C (xcc) do trecho do alimentador, em [ohms/m].

Essa matriz está descrita pelo *script* do Matlab apresentado a seguir.

```
1 topologia = [...
10 20 50 0.1830856/1000 0.8662534/1000 0 0 0 0 0 0.1830856/1000 0.8662534/1000 0 0 0 0 0.1830856/1000 0.8662534/1000; ...
3 20 30 150 0.2241236/1000 0.9144412/1000 0 0 0 0 0 0.2241236/1000 0.9144412/1000 0 0 0 0 0.2241236/1000 0.9144412/1000; ...
5 20 40 80 0.2154789/1000 0.7567231/1000 0 0 0 0 0 0.2154789/1000 0.7567231/1000 0 0 0 0 0.2154789/1000 0.7567231/1000; ...
];
```

Vale ressaltar que essa topologia descreve um sistema trifásico, simétrico e equilibrado e, portanto, é possível tratá-lo como um sistema monofásico, por meio do seu equivalente na ligação estrela.

A descrição do algoritmo na seção 1 considerou essa equivalência, tratando o sistema elétrico como monofásico (assim como o algoritmo de localização de faltas apresentado nesta seção).

3.2 Matriz de cargas - arquivo <CAR000.m>

A matriz de cargas possui os seguintes dados:

- Coluna 1: nó onde a carga está conectada (atenção: não há carga conectada no nó da subestação e, além disso, todas as cargas trifásicas são equilibradas, e conectadas em estrela aterrada;
- Coluna 2: potência ativa trifásica da carga, em [kW]; e
- Coluna 3: fator de potência da carga (neste exercício-programa considerou-se que as cargas têm natureza indutiva).

Essa matriz está descrita pelo *script* do Matlab apresentado a seguir.

```
1 cargas = [...
2 20 220 0.87; ...
3 30 130 0.95; ...
4 40 170 0.98; ...
5 ];
```

Para a solução do equivalente monofásico, considerou-se que a carga descrita por essa matriz é trifásica, ou seja, a carga monofásica consome $\frac{1}{3}$ dessa potência.

3.3 Tensões na subestação - arquivo <VOL000.m>

A matriz de tensões medidas na subestação possui os seguintes dados:

- Coluna 1: caso de simulação (neste exemplo há dez simulações distintas);
- Coluna 2: parte real da tensão na fase A, em [V];
- Coluna 3: parte imaginária da tensão na fase A, em [V];
- Coluna 4: parte real da tensão na fase B, em [V];
- Coluna 5: parte imaginária da tensão na fase B, em [V];
- Coluna 6: parte real da tensão na fase C, em [V]; e
- Coluna 7: parte imaginária da tensão na fase C, em [V].

Essa matriz está descrita pelo *script* do Matlab apresentado a seguir.

```
1 Emedido = [ ...
2 01 7925.18 -195.43 -4131.84 -6765.70 -3793.35 6961.13
3 02 7920.17 -209.38 -4141.42 -6754.38 -3778.76 6963.76
4 03 7905.45 -269.62 -4186.22 -6711.51 -3719.23 6981.13
5 04 7892.91 -301.71 -4207.75 -6684.61 -3685.17 6986.32
6 05 7891.81 -299.81 -4205.54 -6684.60 -3686.26 6984.41
7 06 7910.11 -223.70 -4148.79 -6738.50 -3761.32 6962.21
8 07 7469.35 -1228.21 -4798.33 -5854.54 -2671.02 7082.75
9 08 7854.38 -410.48 -4282.68 -6596.85 -3571.70 7007.33
10 09 7862.55 -378.62 -4259.17 -6619.85 -3603.38 6998.47
11 10 7875.38 -341.87 -4233.76 -6649.35 -3641.63 6991.21
12 11 7647.87 -971.76 -4665.50 -6137.37 -2982.37 7109.13
13 ];
```

3.4 Localização de faltas - arquivo <main_program.m>

Nota-se que a resposta da busca por força bruta retorna um valor de x e R_F para cada trecho investigado e, portanto, para cada caso de simulação há três repostas.

O algoritmo de localização de faltas está apresentado a seguir.

```
1 Vth = 13800/sqrt(3); % Fonte de tensão do equivalente de Thevenin do nó da subestação
2 Zth = (Vth^2)/(((1/3)*10E8)*(cos(80*pi/180) - 1i*sin(80*pi/180))); % Impedância equivalente do Thevenin do nó da subestação
3 Ith = Vth/Zth; % Fonte de corrente do equivalente de Norton do nó da subestação
4 Yth = 1/Zth; % Admitância do equivalente de Norton do nó da subestação
5 Rmax = 10; % Valor máximo da resistência de falta

6 % Dados de topologia
7 % <nó pai><nó filho><comprimento [m]><resistência do trecho [ohms/m]><reatância do trecho [ohms/m]>
8 TDP000;

9 % Dados das cargas
10 % <nó onde a carga está conectada><potência [kW] da carga><fator de potência (indutivo)>
11 CAR000;

12 % Dados das tensões medidas no nó da subestação para curtos-circuitos em pontos desconhecidos
13 % <número do evento de curto-circuito><parte real da tensão [V]><parte imaginária da tensão [V]>
14 VOL000;
15 for contcasos=1:10;
16 E10meas = Emedido(contcasos,2) + 1i*Emedido(contcasos,3); % Leitura da fase A (equivalente monofásico) do primeiro caso de simulação
```



```

21 % Cálculo das impedâncias das cargas
for aux=1:size(cargas,1)
23 Zcarga(aux) = (13800^2)/(1000*(cargas(aux,2) - 1i*cargas(aux,2).*tan(acos(cargas(aux,3)))));
end

25
27 % Teste do trecho 10-20
% 1) Inicialização da função a ser otimizada
funcao = Inf;
29 rf=0.1:0.1:Rmax % Laço for que testa todas as distâncias com passo de 1[m]
for x=1:1:topologia(1,3)-1 % Laço for que testa resistências de 0.1 a Rmax [ohms] com passo de 0.1 [ohm]
31 % 2) Montagem da matriz Ybus
% Posição 1 (linha e/ou coluna da matriz) -> equivale ao nó 10
33 % Posição 2 (linha e/ou coluna da matriz) -> equivale ao nó 20
% Posição 3 (linha e/ou coluna da matriz) -> equivale ao nó 30
35 % Posição 4 (linha e/ou coluna da matriz) -> equivale ao nó 40
% Posição 5 (linha e/ou coluna da matriz) -> equivale ao nó de falta
37 Ybus = zeros(5,5);
Ybus(1,1) = Yth + inv(x*(topologia(1,4) + 1i*topologia(1,5)));
39 Ybus(1,2) = 0;
Ybus(1,3) = 0;
41 Ybus(1,4) = 0;
Ybus(1,5) = -inv(x*(topologia(1,4) + 1i*topologia(1,5)));
43 Ybus(2,1) = Ybus(1,2);
Ybus(2,2) = inv((topologia(1,3)-x)*(topologia(1,4) + 1i*topologia(1,5))) + inv(Zcarga(1)) + inv(topologia(2,3)*(topologia(2,4) + 1i*topologia(2,5))) +
inv(topologia(3,3)*(topologia(3,4) + 1i*topologia(3,5)));
45 Ybus(2,3) = -inv(topologia(2,3)*(topologia(2,4) + 1i*topologia(2,5)));
Ybus(2,4) = -inv(topologia(3,3)*(topologia(3,4) + 1i*topologia(3,5)));
47 Ybus(2,5) = -inv((topologia(1,3)-x)*(topologia(1,4) + 1i*topologia(1,5)));
Ybus(3,1) = Ybus(1,3);
49 Ybus(3,2) = Ybus(2,3);
Ybus(3,3) = inv(Zcarga(2)) + inv(topologia(2,3)*(topologia(2,4) + 1i*topologia(2,5)));
51 Ybus(3,4) = 0;
Ybus(3,5) = 0;
53 Ybus(4,1) = Ybus(1,4);
Ybus(4,2) = Ybus(2,4);
55 Ybus(4,3) = Ybus(3,4);
Ybus(4,4) = inv(Zcarga(3)) + inv(topologia(3,3)*(topologia(3,4) + 1i*topologia(3,5)));
57 Ybus(4,5) = 0;
Ybus(5,1) = Ybus(1,5);
59 Ybus(5,2) = Ybus(2,5);
Ybus(5,3) = Ybus(3,5);
61 Ybus(5,4) = Ybus(4,5);
Ybus(5,5) = inv(x*(topologia(1,4) + 1i*topologia(1,5))) + inv(rf) + inv((topologia(1,3)-x)*(topologia(1,4) + 1i*topologia(1,5)));
63 % 3) Cálculo das tensões nodais
E = Ybus\Ith; 0; 0; 0; 0;
65 E10calc = E(1);
% 4) Cálculo da função a ser otimizada
67 % 5) Armazenamento da distância e da resistência de falta para o menor valor encontrado para a função a ser otimizada
funcao_old = abs(E10meas - E10calc)/abs(E10meas);
69 nheca(aux) = funcao_old;
if funcao_old<funcao
71 xcalc = x;
rcalc = rf;
73 funcao = funcao_old;
end
75 end
end
77 %fprintf('Trecho 10-20 - f: %2.4f\t - rf: %2.4f\t - x: %d\n',funcao,rcalc,xcalc);
fprintf('%02.f,%03.f,%03.f,%03.f,%2.1f,%2.3f\n',contcasos,10,20,xcalc,rcalc,funcao);
79
% Teste do trecho 20-30
% 1) Inicialização da função a ser otimizada
funcao = Inf;
83 for x=1:1:topologia(2,3)-1 % Laço for que testa todas as distâncias com passo de 1[m]
for rf=0.1:0.1:Rmax % Laço for que testa resistências de 0.1 a Rmax [ohms] com passo de 0.1 [ohm]
85 % 2) Montagem da matriz Ybus
% Posição 1 (linha e/ou coluna da matriz) -> equivale ao nó 10
87 % Posição 2 (linha e/ou coluna da matriz) -> equivale ao nó 20
% Posição 3 (linha e/ou coluna da matriz) -> equivale ao nó 30
89 % Posição 4 (linha e/ou coluna da matriz) -> equivale ao nó 40
% Posição 5 (linha e/ou coluna da matriz) -> equivale ao nó de falta
91 Ybus = zeros(5,5);
Ybus(1,1) = Yth + inv((topologia(1,3)*(topologia(1,4) + 1i*topologia(1,5))));
93 Ybus(1,2) = -inv((topologia(1,3)*(topologia(1,4) + 1i*topologia(1,5))));
Ybus(1,3) = 0;
95 Ybus(1,4) = 0;
Ybus(1,5) = 0;
97 Ybus(2,1) = Ybus(1,2);
Ybus(2,2) = inv(topologia(1,3)*(topologia(1,4) + 1i*topologia(1,5))) + inv(Zcarga(1)) + inv(x*(topologia(2,4) + 1i*topologia(2,5))) + inv(topologia(3,3)
*(topologia(3,4) + 1i*topologia(3,5)));
99 Ybus(2,3) = 0;
Ybus(2,4) = -inv(topologia(3,3)*(topologia(3,4) + 1i*topologia(3,5)));
101 Ybus(2,5) = -inv(x*(topologia(2,4) + 1i*topologia(2,5)));
Ybus(3,1) = Ybus(1,3);
103 Ybus(3,2) = Ybus(2,3);
Ybus(3,3) = inv(Zcarga(2)) + inv((topologia(2,3)-x)*(topologia(2,4) + 1i*topologia(2,5)));
105 Ybus(3,4) = 0;
Ybus(3,5) = -inv((topologia(2,3)-x)*(topologia(2,4) + 1i*topologia(2,5)));
107 Ybus(4,1) = Ybus(1,4);
Ybus(4,2) = Ybus(2,4);
109 Ybus(4,3) = Ybus(3,4);
Ybus(4,4) = inv(Zcarga(3)) + inv(topologia(3,3)*(topologia(3,4) + 1i*topologia(3,5)));
111 Ybus(4,5) = 0;
Ybus(5,1) = Ybus(1,5);
113 Ybus(5,2) = Ybus(2,5);
Ybus(5,3) = Ybus(3,5);
115 Ybus(5,4) = Ybus(4,5);
Ybus(5,5) = inv(x*(topologia(2,4) + 1i*topologia(2,5))) + inv(rf) + inv((topologia(2,3)-x)*(topologia(2,4) + 1i*topologia(2,5)));
117 % 3) Cálculo das tensões nodais
E = Ybus\Ith; 0; 0; 0; 0;
119 E10calc = E(1);
% 4) Cálculo da função a ser otimizada
121 % 5) Armazenamento da distância e da resistência de falta para o menor valor encontrado para a função a ser otimizada
funcao_old = abs(E10meas - E10calc)/abs(E10meas);
123 if funcao_old<funcao
xcalc = x;
125 rcalc = rf;
funcao = funcao_old;
127 end
end
129 end
%fprintf('Trecho 20-30 - f: %2.4f\t - rf: %2.4f\t - x: %d\n',funcao,rcalc,xcalc);
131 fprintf('%02.f,%03.f,%03.f,%03.f,%2.1f,%2.3f\n',contcasos,20,30,xcalc,rcalc,funcao);

% Teste do trecho 20-40
% 1) Inicialização da função a ser otimizada
135 funcao = Inf;
for x=1:1:topologia(3,3)-1 % Laço for que testa todas as distâncias com passo de 1[m]
137 for rf=0.1:0.1:Rmax % Laço for que testa resistências de 0.1 a Rmax [ohms] com passo de 0.1 [ohm]
% 2) Montagem da matriz Ybus

```

```

139 % Posição 1 (linha e/ou coluna da matriz) -> equivale ao nó 10
141 % Posição 2 (linha e/ou coluna da matriz) -> equivale ao nó 20
143 % Posição 3 (linha e/ou coluna da matriz) -> equivale ao nó 30
145 % Posição 4 (linha e/ou coluna da matriz) -> equivale ao nó 40
147 % Posição 5 (linha e/ou coluna da matriz) -> equivale ao nó de falta
149 Ybus = zeros(5,5);
151 Ybus(1,1) = Yth + inv((topologia(1,3)*(topologia(1,4) + 1i*topologia(1,5))));
153 Ybus(1,2) = -inv((topologia(1,3)*(topologia(1,4) + 1i*topologia(1,5))));
155 Ybus(1,3) = 0;
157 Ybus(1,4) = 0;
159 Ybus(1,5) = 0;
161 Ybus(2,1) = Ybus(1,2);
163 Ybus(2,2) = inv(topologia(1,3)*(topologia(1,4) + 1i*topologia(1,5)) + inv(Zcarga(1)) + inv(topologia(2,3)*(topologia(2,4) + 1i*topologia(2,5)) + inv(x
    *(topologia(3,4) + 1i*topologia(3,5))));
165 Ybus(2,3) = -inv(topologia(2,3)*(topologia(2,4) + 1i*topologia(2,5)));
167 Ybus(2,4) = 0;
169 Ybus(2,5) = -inv(x*(topologia(3,4) + 1i*topologia(3,5)));
171 Ybus(3,1) = Ybus(1,3);
173 Ybus(3,2) = Ybus(2,3);
175 Ybus(3,3) = inv(Zcarga(2)) + inv(topologia(2,3)*(topologia(2,4) + 1i*topologia(2,5)));
177 Ybus(3,4) = 0;
179 Ybus(3,5) = 0;
181 Ybus(4,1) = Ybus(1,4);
183 Ybus(4,2) = Ybus(2,4);
185 Ybus(4,3) = Ybus(3,4);
187 Ybus(4,4) = inv(Zcarga(3)) + inv((topologia(3,3)-x)*(topologia(3,4) + 1i*topologia(3,5)));
189 Ybus(4,5) = -inv((topologia(3,3)-x)*(topologia(3,4) + 1i*topologia(3,5)));
191 Ybus(5,1) = Ybus(1,5);
193 Ybus(5,2) = Ybus(2,5);
195 Ybus(5,3) = Ybus(3,5);
197 Ybus(5,4) = Ybus(4,5);
199 Ybus(5,5) = inv(x*(topologia(3,4) + 1i*topologia(3,5)) + inv(rf) + inv((topologia(3,3)-x)*(topologia(3,4) + 1i*topologia(3,5)));
201 % 3) Cálculo das tensões nodais
203 E = Ybus\Ith; 0; 0; 0; 0;
205 E10calc = E(1);
207 % 4) Cálculo da função a ser otimizada
209 % 5) Armazenamento da distância e da resistência de falta para o menor valor encontrado para a função a ser otimizada
211 funcao_old = abs(E10meas - E10calc)/abs(E10meas);
213 if funcao_old < funcao
215     xcalc = x;
217     rcalc = rf;
219     funcao = funcao_old;
221 end
223 end
225 %fprintf('Trecho 20-40 - f: %2.4f\t - rf: %2.4f\t - x: %d\n',funcao,rcalc,xcalc);
227 fprintf('%02.f,%03.f,%03.f,%03.f,%2.1f,%2.3f\n',contcasos,20,40,xcalc,rcalc,funcao);
229 end

```

3.5 Saída do processo de localização - arquivo <OUT000.csv>

Nota-se que a resposta da busca por força bruta retorna um valor de x e R_F para cada trecho investigado e, portanto, para cada caso de simulação há três repostas. Sendo assim, os dez casos de simulação resultam em um arquivo de trinta linhas. As colunas desse arquivo são:

- Número do caso de simulação;
- Nó pai;
- Nó filho;
- Distância do nó filho, em [m];
- Resistência de falta, em [ohms]; e
- Valor da função de otimização.

```

1 01,010,020,049,7.7,0.000
01,020,030,001,7.7,0.000
3 01,020,040,001,7.7,0.000
02,010,020,049,7.2,0.000
5 02,020,030,070,7.1,0.000
02,020,040,079,7.1,0.000
7 03,010,020,049,5.5,0.000
03,020,030,025,5.5,0.000
9 03,020,040,031,5.5,0.000
04,010,020,049,4.9,0.001
11 04,020,030,077,4.9,0.000
04,020,040,079,4.9,0.000
13 05,010,020,049,4.9,0.001
05,020,030,110,4.9,0.000
15 05,020,040,079,4.9,0.000
06,010,020,049,6.6,0.001
17 06,020,030,149,6.6,0.000
06,020,040,079,6.6,0.001
19 07,010,020,049,1.1,0.003
07,020,030,003,1.1,0.003
21 07,020,040,002,1.1,0.003
08,010,020,049,3.5,0.002
23 08,020,030,106,3.5,0.000
08,020,040,079,3.5,0.001
25 09,010,020,049,3.8,0.002
09,020,030,140,3.8,0.000
27 09,020,040,079,3.8,0.001
10,010,020,049,4.3,0.001
29 10,020,030,149,4.2,0.000
10,020,040,079,4.3,0.001

```

3.6 Saída do processo de localização - arquivo <REL000.csv>

A resposta da busca por força bruta retorna um valor de x e R_F para cada trecho investigado e, portanto, para se determinar o trecho “mais provável” de ocorrência da falta, deve-se selecionar aquele que apresentou o menor valor para a função de otimização. Essa informação é armazenada nesse arquivo, que possui os seguintes dados:

01	10	20	49	7.70	0.000
02	10	20	49	7.30	0.000
03	10	20	01	5.70	0.000
04	10	20	49	5.10	0.000
05	10	20	49	5.20	0.000
06	10	20	49	7.30	0.000
06	10	20	49	1.10	0.000
07	10	20	49	3.80	0.000
08	10	20	49	4.20	0.000
09	10	20	49	4.80	0.000
10	10	20	01	1.50	0.000

- Número do caso de simulação;
- Nó pai;
- Nó filho;
- Distância do nó filho, em [m];
- Resistência de falta, em [ohms]; e
- Valor da função de otimização.

4 Enunciado do exercício programa

O exercício programa consiste na elaboração de um algoritmo para localização de faltas em redes de distribuição, com base nas informações apresentadas nas seções anteriores, que possa ser empregado para quaisquer redes de distribuição trifásicas, equilibradas ou não, e monofásicas.

O exercício programa deve ser conduzido por grupos de alunos. Portanto, os alunos devem se organizar em grupos de **três alunos** e registrar sua escolha no Moodle. Um número identificador é atribuído ao grupo quando essa escolha é feita, de modo que cada grupo deve possuir apenas um número. **A data limite para escolha e registro dos grupos é 30/04/2019.**

Esse exercício programa deve ser implementado e desenvolvido utilizando-se o Matlab ou o Octave, por meio de arquivos do tipo *script* e/ou *function* salvos no formato **<m-file>**.

Os arquivos de entrada respeitam a descrição apresentada na seção 3 e há um conjunto desses arquivos para cada grupo. Todos os conjuntos estão armazenados em apenas um arquivo compactado, denominado **<EntradaDados.zip>**, e possuem os mesmos números identificadores dos grupos. Cada grupo deve trabalhar com **apenas um** conjunto de arquivos (p. ex. o grupo 034 deve trabalhar com os arquivos TOP034.m, CAR034.m e VOL034.m).

Vale ressaltar que o arquivo de tensões (p. ex. o grupo 034 deve utilizar o arquivo <VOL034.m>) possui dez casos de curto-circuito em diferentes pontos do circuito de distribuição, com resistências de falta distintas. Os primeiros cinco casos foram elaborados considerando equilibrada a rede de distribuição descrita pelo arquivo de topologia (p. ex. para o grupo 034 é o arquivo <TOP034.m>). No entanto, os dados dessa rede, apresentados no arquivo de topologia, não estão equilibrados. Isto significa que o grupo de alunos deve primeiro 'equilibrar' a rede e utilizar os dados da rede equilibrada para efetuar a localização de faltas nesses cinco primeiros casos (dica: a impedância própria do trecho deve ser a média das próprias e a impedância mútua do trecho deve ser a média das mútuas). Os demais casos foram elaborados considerando desequilibrada a rede de distribuição descrita nesse arquivo de topologia.

É **mandatório** que os arquivos de saída possuam nome e formato conforme descrito nas seções 3.5 e 3.6 (p. ex. o grupo 034 deve produzir dois arquivos de saída cujos nomes devem ser OUT034.csv e REL034.csv, e cujos dados deve estar no formato descrito nessas respectivas seções).

Após a elaboração, implementação e testes do algoritmo de localização de faltas desenvolvido pelo grupo de alunos, esse grupo deve submeter o código-fonte e os arquivos de saída por meio de um arquivo compactado no formato **<.ZIP>** (p. ex. o grupo 034 deve submeter um arquivo cujo nome deve ser <034.ZIP> e que deve conter todos os arquivos de código-fonte e os dois arquivos de saída <OUT034.csv> e <REL034.CSV>). É importante ressaltar que os códigos-fonte serão testados e devem rodar em modo *standalone* sem a necessidade de nenhum tipo de configuração de *path* de execução ou de entrada de dados e, obviamente, devem produzir os mesmos arquivos de saída entregues pelo grupo.

A data limite para entrega do exercício programa é 09/06/2019.