DOI: 10.48011/asba.v2i1.1449

Uma Proposta para Automação de Bases de Dados Simuladas de Faltas de Alta Impedância

Érica Mangueira Lima * **Mateus Costa Lucena** **
Núbia Silva Dantas Brito * **Benemar Alencar de Souza** *

* *Universidade Federal de Campina Grande - UFCG, PB*
(e-mail: erica.lima@ee.ufcg.edu.br, {nubia, benemar}@dee.ufcg.edu.br).
** *SIDIA Instituto de Ciência e Tecnologia - Manaus, AM*
(e-mail: mateus.lucena@sidia.com)

Abstract: The steps for simulated database automatic construction for high impedance faults are presented in this paper, aiming to supply the scarcity of real oscillographic records for this type of fault. Matlab® and ATP softwares are used for structuring and simulation of the scenarios. The concept of robotic process automation is adopted as a solution for software integration and system automation, using Python language. To support the decision making, graph theory is applied to the selection of simulation variables. Presented results show productivity gains and suggest that the adopted methodology can be applied in simulated database construction for other disturbances in electric power systems.

Resumo: As etapas para a construção automática de uma base de dados simulada para faltas de alta impedância são apresentadas neste trabalho, visando suprir a escassez de registros oscilográficos reais para este tipo de falta. Os *softwares* Matlab® e ATP são utilizados para estruturação e simulação dos cenários. O conceito de automação robótica de processos é adotado enquanto solução para integração dos *softwares* e automação do sistema, utilizando a linguagem Python. Em auxílio à tomada de decisões, a teoria dos grafos é aplicada à seleção das variáveis de simulação. Os resultados apresentados evidenciam ganhos produtivos e sugerem que a metodologia adotada pode ser aplicada na construção de bases de dados simulados de outros distúrbios dos sistemas elétricos de potência.

Keywords: Database construction; High impedance faults; Robotic process automation; Graph theory; Electrical distribution systems.

Palavras-chaves: Construção de bases de dados; Faltas de alta impedância; Automação robótica de processos; Teoria dos grafos; Sistemas de distribuição de energia elétrica.

1. INTRODUÇÃO

Faltas de alta impedância (FAI) são distúrbios que ocorrem quando um condutor energizado em média tensão estabelece um contato indesejado com uma superfície de alto valor resistivo. As FAI incidem predominantemente em sistemas de distribuição de energia elétrica (SDEE) e tem como característica principal a baixa amplitude da corrente, a qual é insuficiente para sensibilizar o sistema de proteção convencional, composto principalmente por dispositivos de sobrecorrente (Costa, 2011). Outrossim, essa característica da corrente compromete o seu registro pelos dispositivos usualmente utilizados para o monitoramento de sinais nos SDEE. Por consequência, os registros oscilográficos reais desse distúrbio são escassos.

Diante desse cenário, os pesquisadores recorrem a registros simulados, o que requer um conhecimento amplo do sistema e do fenômeno. Os resultados das análises realizadas por meio de uma base de dados simulada podem ser favorecidos, mesmo que de maneira involuntária, pelas escolhas de seu desenvolvedor. Desse modo, é essencial garantir a isonomia na seleção e dimensionamento das variáveis do sistema e da falta. Para tanto, as simulações devem contemplar diversas condições de FAI e do SDEE. Além disso, os cenários devem abranger outros distúrbios e eventos comuns aos SDEE, os quais, devido às suas características, podem ser confundidos com uma FAI, provocando atuações indevidas da proteção (Santos et al., 2013).

No que se refere ao estado da arte, destacam-se aqui os trabalhos de Mora et al. (2006) e Santos et al. (2010), que propuseram metodologias para construção de bases de dados para faltas de baixa e alta impedância, respectivamente. Ambos utilizaram os *softwares Alternative Transients Program* - ATP e Matlab® integrados, adotando metodologias semelhantes para automatizar o processo de construção da base de dados. No entanto, Santos et al. (2010) simula um sistema para aquisição de dados, escrevendo-os em um arquivo a ser convertido para ao formato `COMTRADE` [1] . Mora et al. (2006), por sua vez, adota a conversão do arquivo de

[1] Formato usual para a representação de registros oscilográficos em dispositivos de proteção e registradores digitais nos SDEE.

saída do próprio ATP para o Matlab®, porém a estratégia para conversão entre os formatos não foi descrita. Além disso, ambos os trabalhos adotaram escolhas empíricas para as variáveis de simulação que determinam os cenários analisados na base de dados.

Este trabalho apresenta uma metodologia para uma construção automática de uma base de dados para FAI. Para tanto, explora-se o conceito de automação robótica de processos, do inglês *robotic process automation* (RPA) (Van der Aalst et al., 2018), programada na linguagem Python, para integração dos referidos *softwares* e automação dos processos. Na estratégia aplicada, dispensa-se a programação do sistema de aquisição de dados e a geração de um arquivos de dados, reduzindo assim, o tempo de execução de cada cenário e possibilitando a conversão do arquivo de saída do ATP para diferentes formatos, inclusive àqueles adotados nos trabalhos supracitados. Adicionalmente, a fim de reduzir a interferência das escolhas de desenvolvedor na confiabilidade dos métodos de diagnóstico, propõe-se a modelagem do sistema por meio da teoria dos grafos, e o uso de distribuições de probabilidade para escolha aleatória dos locais de falta, assegurando desse modo, a isonomia na distribuição dos pontos nos quais a FAI será aplicada.

Este trabalho está organizado da seguinte forma: na Seção 2 apresenta-se a metodologia para construção da base de dados, na Seção 3 apresenta-se o processo de automação da base de dados, na Seção 4 avalia-se a metodologia proposta e na Seção 5 apresentam-se as conclusões.

2. CONSTRUÇÃO DE UMA BASE DE DADOS

Neste trabalho, a base de dados consiste em um conjunto de cenários simulados correspondentes às diversas situações as quais um sistema de potência é submetido. Essas situações são modeladas por meio de variáveis de simulação, e.g., o carregamento e topologia do sistema ou parâmetros referentes à falta. Um processo típico para construção dos cenários é apresentado na Figura 1, no qual cada cenário refere-se a uma combinação das variáveis de simulação, que determina os parâmetros adotados tanto na simulação do sistema-teste quanto do distúrbio. A base de dados contempla todas as combinações possíveis dentro do escopo proporcionado pela escolha das variáveis.

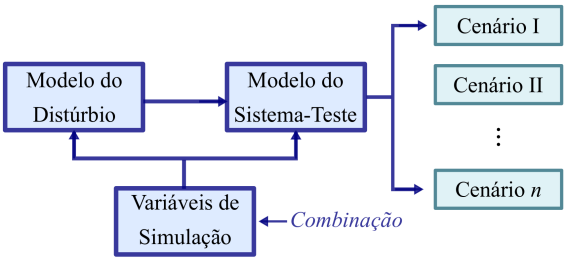

Figura 1. Processo de construção dos cenários da base de dados.

Os métodos propostos por Santos et al. (2010) e Souza et al. (2005) adotam etapas para sistematizar a construção da base de dados, iniciando com a seleção dos *softwares* utilizados, seguido da modelagem do sistema-teste e dos distúrbios, a escolha das variáveis de simulação e por fim, a geração dos cenários.

Para construção da base de dados proposta neste trabalho utilizaram-se os *softwares* Matlab© e ATP (*Alternative Transients Program*) conjuntamente, sendo o primeiro responsável pelos aspectos organizacionais e a estruturação da base e o segundo utilizado na simulação de cada um dos cenários. Para automação do processo, adotou-se a linguagem Python.

2.1 Seleção dos Modelos

O sistema-teste utilizado corresponde a um alimentador real da concessionária local, cujo diagrama unifilar está apresentado na Figura 2. O sistema opera em 13,8 kV e todos os dados das características elétricas e espaciais e das as cargas distribuídas ao longo dos transformadores, foram cedidos pela própria concessionária (Santos et al., 2010).

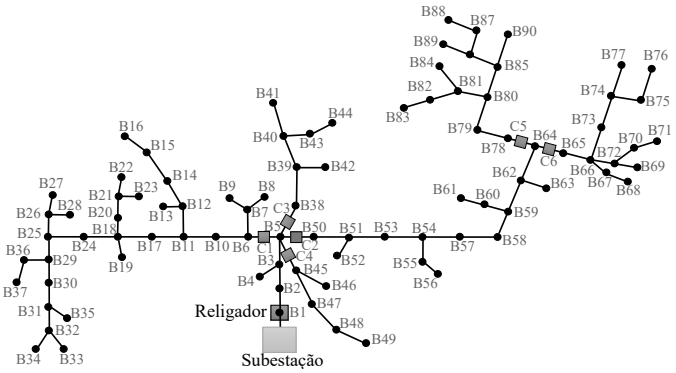

Figura 2. Sistema-teste. Fonte: (Santos et al., 2010)

Na modelagem da FAI adotou-se o modelo proposto por Santos (2016), apresentado na Figura 3. O modelo é composto por: i) duas chaves – a chave 1 que tem como função dar início a falta e simular intermitências na corrente e a chave 2 que simula o rompimento do condutor; ii) duas resistências variantes no tempo $R_1(t)$ e $R_2(t)$, que tem valores calculados com base em registros oscilográficos experimentais e simulam as principais características do fenômeno: não-linearidade, assimetria entre os semiciclos da corrente e o *build-up* e *shoulder*, que são ciclos de o crescimento da envoltória da corrente seguidos por alguns ciclos nos quais a amplitude se mantém constante.

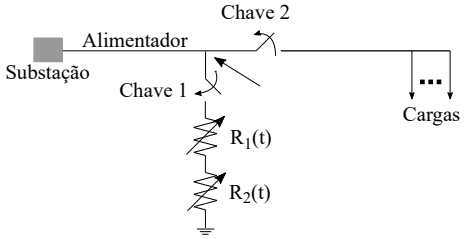

Figura 3. Modelo de FAI. Fonte: (Santos, 2016)

A base de dados contém ainda outros distúrbios, cujas características podem ser confundidas com as de uma FAI. Esses distúrbios são importantes para avaliar, por exemplo, questões como a seletividade em métodos de detecção. São eles:

- Energização de linhas (ENE) – simulado por meio de uma chave entre as barras correspondentes à localização desejada no sistema;

- Chaveamento de bancos de capacitores (BC) – representados por um banco de 1,8 Mvar, valor dado por ser usual na concessionária local;
- Faltas de baixa impedância (FBI) – considerando apenas faltas monofásicas do tipo AT (fase A - terra) e adotando uma resistência de 10 Ω.
- Energização de transformadores (ET) – simulado com auxílio da rotina `BCTRAN` do ATP, considerando transformadores de 300 kVA em locais específicos do sistema.

2.2 Seleção das variáveis de simulação

Uma etapa essencial na construção de uma base de dados é a escolha das variáveis de simulação, visto que elas determinam os cenários que serão considerados e, consequentemente, estabelecem a abrangência da base de dados. As variáveis de simulação devem estar relacionadas com parâmetros do sistema, como carregamento e topologia, e com parâmetros da própria falta, como: instante de ocorrência, determinado pelo ângulo de incidência; resistência de falta, que é função do tipo de solo; e localização da falta, estimada usualmente pela barra mais próxima ao ponto de rompimento do condutor.

No que se refere aos parâmetros do sistema, consideraram-se os carregamentos de 25%, 50% e 70%. Essa variação pode difererir conforme a aplicação. Quanto à topologia, definiram-se seis cenários adicionais correspondentes à abertura das chaves seccionadoras presentes no sistema. Cada cenário é definido pela quantidade de barras resultantes, que neste estudo foram: 78, 64, 46, 42 e 20 barras.

Em relação aos parâmetros de falta, a resistência é determinada pelo modelo de FAI adotado, o qual possibilita o estudo de seis tipos de solo: areia, asfalto, brita, calçamento, grama e terra local. De acordo com dados experimentais (Santos et al., 2010), as resistências correspondentes a esses solos podem variar entre aproximadamente 300 Ω para areia e terra local até cerca de 2500 Ω para solos compactados como asfalto e calçamento. Os valores para os ângulos de incidência foram escolhidos visando evitar cenários semelhantes, portanto, considerando a função seno, adotaram-se os seguintes valores: 0°, 90° e 180°.

A escolha dos locais de falta, no entanto, é uma decisão mais subjetiva. Idealmente, todos os pontos dos sistemas deveriam ser considerados, porém isso aumentaria o tempo de processamento e a necessidade de armazenamento, tornando a construção mais lenta e dispendiosa. Sendo assim, propõe-se a escolha de pontos estratégicos, sem intervenção do usuário, e em quantidade suficiente para caracterizar o distúrbio nos mais diversos pontos do sistema. Neste trabalho, considerou-se apenas 20% das barras presentes no sistema, ou seja, 18 barras. Porém, esses pontos devem estar distribuídos uniformemente ao longo do alimentador. Para isto, o sistema foi remodelado com auxílio da teoria dos grafos.

Teoria dos Grafos e Locais de Falta Sob o ponto de vista topológico, um SDEE pode ser modelado de forma simplicada por meio de grafos, trazendo para o estudo conceitos, técnicas e teoremas definidos com rigor matemático e já estabelecidos pelo uso em outras áreas de conhecimento (Braz, 2010). Na área de sistemas elétricos, a teoria dos grafos vem sendo largamente utilizada em estudos sobre configuração do sistema (Braz, 2010), processamento topológico (Bernardes et al., 2015), fluxo de potência (Marini et al., 2019; Borges et al., 2012) e ilhamento controlado (Souza, 2014).

Um grafo $G(\mathcal{N}, \mathcal{A})$ é composto pelos conjuntos $\mathcal{N}$ e $\mathcal{A}$, no qual um elemento n do conjunto $\mathcal{N}$ chama-se nó, e um elemento $a_{ij} = (n_i, n_j)$ do conjunto $\mathcal{A}$ chama-se aresta (Souza, 2014). No contexto dos SDEE, cada nó representa uma barra do sistema e cada aresta uma linha do alimentador. Portanto, para o sistema-teste analisado, $n(\mathcal{N}) = 91$, sendo um nó correspondente a subestação e os demais às 90 barras, e $n(\mathcal{A}) = 90$.

O principal objetivo da representação do sistema por meio de grafos, nesta aplicação, é proporcionar a escolha de locais de falta distribuídos uniformemente com a distância até o ponto de monitoramento, que, neste trabalho, é a subestação. Nesse sentido, utiliza-se um grafo ponderado no qual cada aresta possui valores associados, chamado de peso de a_{ij} e denotado por $p(a_{ij})$ (Costa, 2011). Neste caso, a ponderação das arestas é dada pela distância entre os nós adjacentes.

A distância entre cada barra e a subestação é dada pelo menor caminho entre o nó i e o nó correspondente à subestação. Em um grafo com arestas de peso não-negativo, este parâmetro pode ser calculado por meio do algoritmo de Dijkstra (1959). Este algoritmo traça o menor caminho buscando-o sucessivamente, e armazenando o nó mais próximo ao nó de origem em um conjunto, considerando os nós adjacentes e descartando os nós já armazenados. Esse procedimento é sintetizado no Algoritmo 1.

Algoritmo 1 Algoritmo de Dijkstra (Grafo G, Nó s)

```
1:  for n ∈ G(N) do
2:      dist(n) ← ∞;                    ▷ Distância inicial infinita
3:      visit(n) ← 0                    ▷ Nenhum nó foi visitado
4:  end for
5:  dist(s) ← 0    ▷ Distância do nó s de si mesmo é nula
6:  F ← N                               ▷ Fila de nós prioritários
7:  while F ≠ 0 do
8:      u ← min(U);
9:      for v proximo de u do       ▷ v: adjacente, u: atual
10:         if dist(v) > dist(u) + peso(u, v) then
11:             dist(v) = dist(u) + peso(u, v)
12:             visit(v) = u
13:             F ← v              ▷ Decresce a prioridade de v
14:         end if
15:     end for
16: end while
17: Saída: dist                         ▷ Distância calculada.
```

Com a distância, D_s, entre cada uma das barras a subestação calculadas, conforme apresentado na Tabela 1, é possível escolher aleatoriamente um conjunto de barras uniformemente distribuídos pelo alimentador. Uma distribuição uniforme, ou retangular, é aquela cuja probabilidade de se gerar qualquer ponto em um intervalo é a mesma, proporcional ao tamanho do intervalo. Neste trabalho, os extremos dos intervalos são a menor e a maior distância dentre as calculadas: 430 m e 32160 m, respectivamente.

Tabela 1. Distância (D_s) entre cada barra e a subestação.

Barra	D_s (m)	Barra	D_s (m)	Barra	D_s (m)
B01	430	**B13**	10260	**B65**	17860
B02	860	**B56**	10460	**B78**	17860
B03	1760	**B57**	10460	**B35**	18560
B05	3260	**B14**	10560	**B32**	18660
B04	3660	**B18**	10660	**B33**	19660
B06	4260	**B49**	11660	**B34**	19660
B50	4560	**B24**	11960	**B66**	19860
B38	5260	**B20**	12160	**B79**	19860
B45	5260	**B15**	12360	**B72**	21060
B10	5760	**B58**	12460	**B67**	21260
B46	5860	**B19**	12660	**B80**	22160
B51	6060	**B25**	13160	**B69**	22260
B07	6260	**B21**	13660	**B73**	22360
B39	6760	**B59**	13660	**B68**	22660
B08	7260	**B26**	14160	**B70**	22860
B09	7260	**B16**	14360	**B81**	23660
B11	7260	**B29**	14460	**B85**	24560
B47	7260	**B22**	14660	**B71**	24660
B52	7260	**B27**	14660	**B74**	24860
B53	7260	**B28**	14760	**B84**	25160
B40	7760	**B62**	14760	**B82**	25660
B42	7960	**B23**	14860	**B90**	26560
B54	8460	**B60**	15660	**B75**	26860
B12	8760	**B64**	15860	**B77**	26960
B43	8860	**B30**	15960	**B86**	27560
B17	8960	**B37**	15960	**B83**	27660
B48	9260	**B63**	17260	**B89**	28760
B41	9360	**B31**	17460	**B87**	29160
B55	9460	**B36**	17460	**B76**	29360
B44	9860	**B61**	17660	**B88**	32160

Dessa forma, 18 barras são sorteadas com o auxílio da função `unidrnd` do Matlab®, que gera números aleatórios considerando a distribuição uniforme discreta, a saber: B04, B08, B10, B15, B27, B34, B40, B47, B54, B63, B66, B71, B75, B79, B83, B84, B87 e B90. Essas barras estão destacadas no grafo da Figura 4, no qual o tamanho das arestas é proporcional a distância das linhas. O histograma de valores que relaciona a quantidade de barras com a distância até a subestação é apresentado na Figura 5, destacando a distribuição uniforme. Observa-se que as 18 barras escolhidas estão distribuídas em três grupos com 6 barras cada, sendo a distância média do primeiro 5000 m, do segundo 15000 m e do terceiro 25000 m. Destaca-se que a quantidade de grupos pode diferir conforme a necessidade da aplicação.

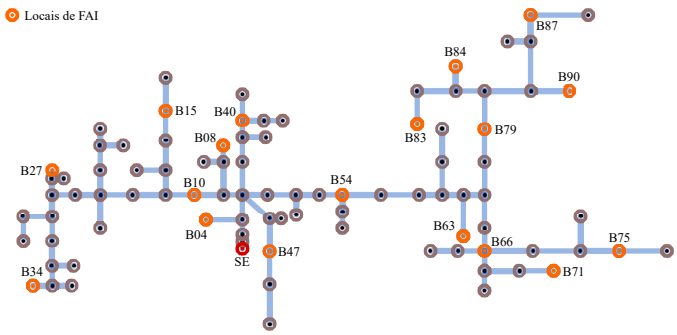

Figura 4. Modelagem do sistema-teste por meio de um grafo ponderado e locais escolhidos para aplicação da FAI.

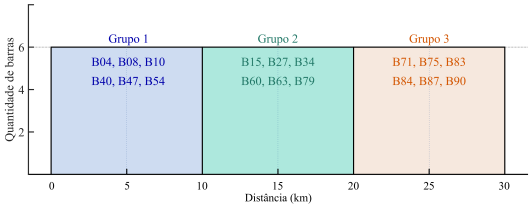

Figura 5. Histograma de valores relativo às distâncias das barras selecionadas até a subestação.

O procedimento proposto, muito embora aplicado para escolha dos locais de falta, pode ser utilizado para as demais variáveis de simulação, modificando a respresentação dada à ponderação das arestas. Se o peso de cada aresta for dado, por exemplo, pela carga em cada barra, é possível obter uma distribuição uniforme para o carregamento do sistema. Isso possibilita a realização de análises considerando variações mais realistas no carregamento do sistema.

3. AUTOMAÇÃO DA BASE DE DADOS

Os procedimentos associados à construção automática da base de dados proposta neste trabalho são apresentados na Figura 6 e descritos em três etapas denominadas pré-processamento, processamento e pós-processamento.

Pré-processamento: Consiste na construção dos cenários de simulação. Nesta etapa, realizada no Matlab®, a teoria dos grafos é utilizada para seleção dos locais de falta, os quais, associados às demais variáveis de simulação, determinam a abrangência da base de dados e os cenários de simulação.

Na rotina de montagem dos arquivos, para cada combinação das variáveis de simulação, um arquivo modelo no formato `.atp` é transferido para um novo arquivo. Esse arquivo modelo contém a modelagem do sistema-teste e dos distúrbios em linhas de código no ATP. As linhas correspondentes a cada uma das variáveis são reescritas para gerar um cenário de simulação específico. No arquivo modelo, aplica-se a função `$PARAMETER`[2], sendo possível alterar uma variável de simulação reescrevendo apenas uma linha de código.

Cada arquivo novo escrito é salvo com a extensão `.atp` e com uma nomenclatura específica, a qual deve conter todas as informações necessárias para identificação do cenário correspondente, conforme sugerido em Santos et al. (2010). Para uma FAI ocorrida na fase A (fAT), na barra 10 (BAR10), medida no registrador digital de perturbação (RDP) presente na subestação (RDP(S)), no tipo de solo areia (ARE), com carregamento de 50% (C50) e ângulo de incidência de 45° (A45), a nomenclatura é dada por:

RDP(S)_fAT_BAR10_ARE_C50_A45.

Ao final, a etapa de pré-processamento tem como saída um conjunto de arquivos `.atp` correspondentes aos cenários presentes na base de dados.

[2] Função do ATP que permite que uma variável declarada no início do código seja substituída por um dado pré-determinado ao longo da simulação.

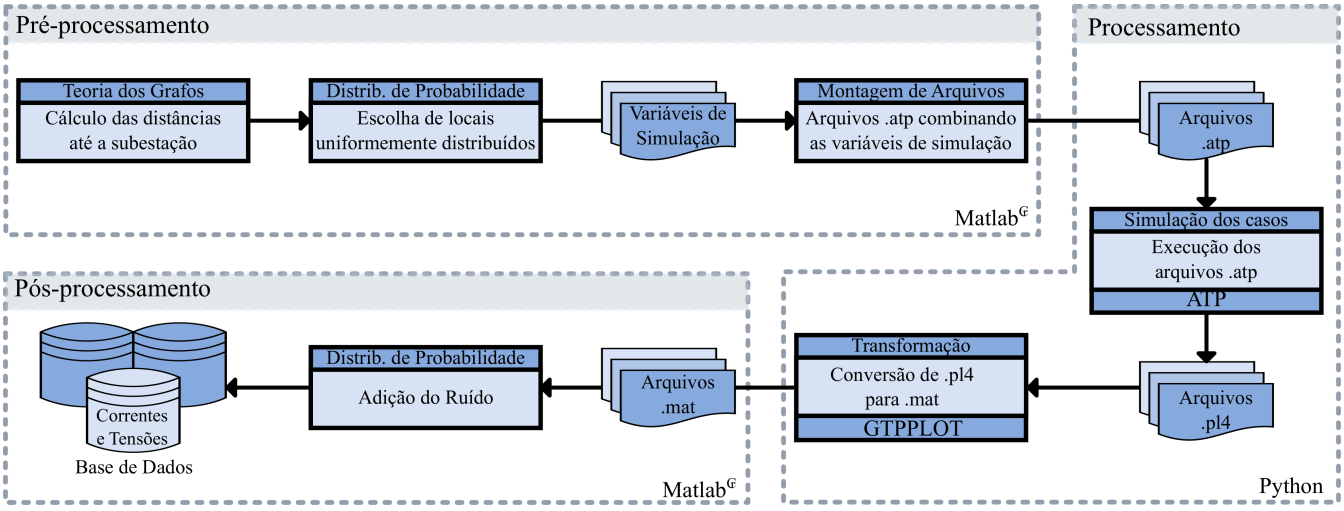

Figura 6. Etapas para construção automática da base de dados.

Processamento: Corresponde à simulação sequencial de cada um dos cenários construídos na etapa de pré-processamento. Esta etapa é dispendiosa e repetitiva, por isso, propõe-se a programação de rotinas de automação baseadas no conceito de RPA, o qual pode ser definido como uma solução baseada em *software*, que consiste em automatizar uma sequência de ações simulando o comportamento humano em alto nível (nível do usuário), com o objetivo de reduzir o esforço e o tempo necessário para realizar tarefas tipicamente simples cujas saídas são determinísticas (Aguirre and Rodriguez, 2017).

Visto que o RPA deve emular a ação humana, é fundamental compreender os procedimentos envolvidos para execução das atividades presentes na etapa de processamento. Nesse sentido, analisa-se as ações para simulação e conversão em um único cenário. Inicialmente, o cenário deve ser simulado no ATP, tendo como saída um arquivo `.pl4`, próprio do ATP, no qual estão armazenados os resultados obtidos na simulação em forma de vetores colunas. Em seguida, o programa `GTPPLOT`, extensão do ATP, converte os arquivos `.pl4` em arquivos `.mat`, por meio do comando `matlab all`, armazenando as variáveis em matrizes que podem ser lidas diretamente no Matlab® por meio da função `load`. É essencial que essa etapa seja realizada de forma sequencial e não simultânea, a fim de evitar erros na conversão. Salienta-se que o `GTPPLOT` também proporciona a conversão entre o `.pl4` e outros formatos, como COMTRADE (`.cfg` e `.dat`), MATHCAD (`.csv`), PostScript, PDF, dentre outros. Portanto, a automação não é restrita à integração com o *software* Matlab®.

A automação dos processos e integração dos *softwares* foi realizada na linguagem Python, conforme o pseudocódigo apresentado no Algoritmo 2. Inicialmente a rotina armazena, nas linhas de 1 a 7, as *strings* correspondentes aos nomes dos arquivos `.atp` em uma pilha. Em seguida, para cada arquivo da pilha, ou seja, para cada cenário, executam-se (linhas de 9 a 24) os passos explicitados no fluxograma da Figura 7. Essa etapa foi programada com auxílio das bibliotecas: i) `subprocess` e `os` – que permitem a abertura do ATP, ii)`keyboard` e `pyautogui` – que proporcionam a manipulação do teclado e mouse, e iii) `time` – que permite o atraso por um tempo determinado. A manipulação do teclado para execução de comandos é realizada por meio de atalhos: 'O' – para abertura dos arquivos, 'R' – para o comando `RunATP`, 'G' – para abrir o `GTPPLOT` e 'ALT+F4' – para fechar as janelas selecionadas. O processo é repetido até que todos os cenários sejam simulados. Salienta-se que o tempo de espera é um parâmetro subjetivo, dependente da configuração do computador utilizado para simulação. Neste trabalho, os tempos de espera destacados na Figura 7 foram calculados empiricamente com base em um caso típico, sendo $t_1 = 20$ minutos, $t_2 = 20$ segundos e $t_3 = 2,5$ minutos, considerando um computador com processador Intel Core i7-7500U e 16 GB de memória RAM.

Os arquivos `.mat`, obtidos ao fim do processo, são armazenados utilizando a mesma nomenclatura adotada para os `.atp`.

Pós-processamento: Esta etapa foi realizada no *software* Matlab® após a base de dados estar completamente simulada. Neste trabalho, optou-se por adicionar aos sinais de corrente e tensão um ruído com distribuição normal e parâmetros calculados por meio de registros reais em Lima et al. (2018).

3.1 Análise dos Resultados

Os cenários obtidos após a execução de todas as etapas estão descritos na Tabela 2. Ao final, foram construídos 1038 cenários, sendo 978 deles de FAI e os demais relacionados com os outros distúrbios e eventos comuns aos SDEE. A título de exemplo, apresenta-se na Figura 3.1 os sinais de tensão e corrente na subestação, e o sinal de corrente na chave que conecta a FAI ao sistema, obtidos para um único cenário de FAI. Esse é um cenário típico para esse distúrbio, no qual observa-se que a corrente de falta tem baixa amplitude, o que dificulta a distinção entre os regimes pré-falta e pós-falta, quando a corrente e a tensão de fase são observadas.

Em relação ao ganho produtivo, apresenta-se na Tabela 3 o tempo necessário para execução e conversão dos cenários,

Tabela 2. Descrição dos cenários da base de dados.

Distúrbio	Variáveis de Simulação		Quantidade de Cenários
FAI	Superfície de Contato	ARE, ASF, BRI, CAL, GRA, TER	978
	Ângulo de Incidência	0, 45, 90	
	Carregamento do Sistema	25%, 50%, 100%	
	Localização	B04, B08, B10, B15, B27, B34, B40, B47, B54, B63, B66, B71, B75, B79, B83, B84, B87, B90	
	Diferentes Topologias (Quantidade de Barras)	78, 64, 57, 46, 42, 20 Barras	
BC	Localização	B04, B08, B10, B15, B27, B34, B40, B47, B54, B63, B66, B71, B75, B79, B83, B84, B87, B90	54
ENE			
FBI			
ET		B04, B08, B34, B63, B71, B90	6
TOTAL			1038

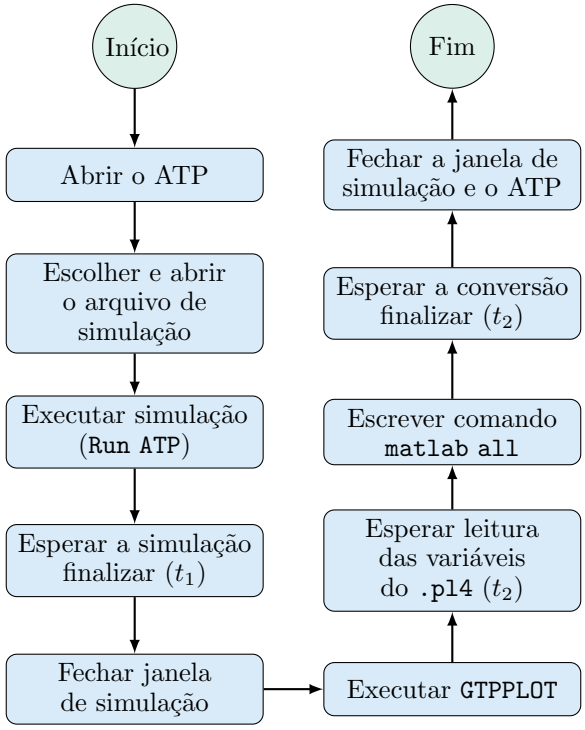

Figura 7. Fluxograma com as etapas necessárias para execução de um cenário.

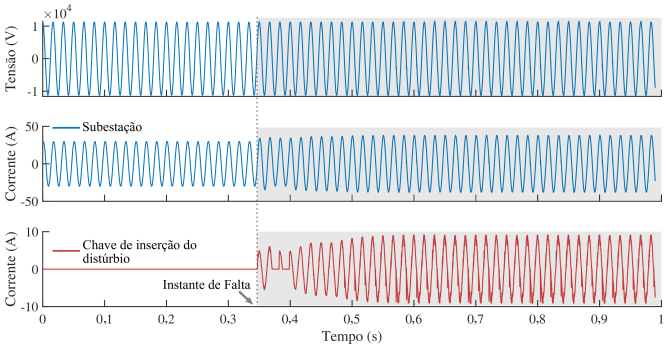

Figura 8. Sinais de corrente e tensão para uma FAI, presentes na base de dados construída.

bem como o tempo total caso a base fosse construída manualmente em comparação com a construção automática proposta neste trabalho. Os tempos de execução

Algoritmo 2 Algoritmo do RPA na execução dos cenários

```
diretorio ← Diretório dos arquivos
arquivos ← Nomes dos arquivos no diretório
for nome ∈ arquivos do
    if nome.endswith(.atp) then
        lista_arquivos ← nome
    end if
end for
for arquivo ∈ list_arquivos do
    Abre ATP
    Pressiona 'O'
    Escreve arquivo
    Pressiona 'Enter'
    Pressiona 'R'
    Espera
    Pressiona 'Alt+F4'
    Pressiona 'G'
    Espera
    Escreve matlab all
    Pressiona 'Enter'
    Espera
    Pressiona 'Alt+F4'
    Move mouse coordenadas(x, y)
    Pressiona 'Alt+F4'
    Print 'Processo Concluído (arquivo)'
end for
```

consideram a simulação de um cenário típico para cada distúrbio e a comparação supõe que, para uma execução manual, o desenvolvedor poderia dispor no máximo de 8 horas ininterruptas por dia, dedicadas exclusivamente à simulação dos cenários. Observa-se que há um ganho significativo quando se emprega a automação nesta atividade, reduzindo em mais de um mês (33 dias) o tempo necessário para a construção da base de dados completa com os 1038 cenários propostos.

Tabela 3. Tempo necessário para construção da base de dados.

Tempo	Execução do .atp		Conversão para .mat	
	FAI	OUTROS	FAI	OUTROS
Cada Cenário	20 min	15 min	3 min	3 min
Total de Cenários	326 horas	15 horas	49 horas	3 horas
Tempo Total	393 horas			
	Manual (8 h/dia)	49 dias		
	Automático (24 h/dia)	16 dias		

Por fim, salienta-se que a simulação de cada cenário da forma proposta dispensa a programação de um sistema de aquisição de dados, utilizando como saída arquivos próprios do ATP, reduzindo assim o tempo necessário para sua execução. Outro ponto importante diz respeito ao fato de que, na execução manual, apenas o tempo de execução de cada uma das atividades foi avaliado, isto é, o tempo necessário para que o usuário inicie cada atividade e alterne entre elas não foi mensurado nem contabilizado. Portanto, na prática, a execução manual deve ocupar ainda mais tempo do que os 49 dias calculados. Adicionalmente, aplicando a automação, o usuário é dispensado e pode se dedicar à outras atividades relacionadas com o estudo.

4. CONCLUSÃO

Este trabalho apresentou uma metodologia para construção e automação de bases de dados de FAI. Uma alternativa sistematizada para obtenção de registros simulados foi proposta e analisada.

Um grafo ponderado pela distância entre as barras foi utilizado para modelar o sistema e guiar a seleção dos locais de falta, adotando uma distribuição de probabilidade uniforme. Essa modelagem pode ser aplicada às demais variáveis do sistema e permite analisar a atuação de métodos de diagnóstico, considerando localizações de falta que abrangem todo o sistema.

Uma solução baseada no conceito de automação robótica de processos, do inglês, *robotic process automation* - RPA, programada na linguagem Python, foi apresentada para automação dos processos e integração dos softwares ATP e Matlab®. A metodologia proposta dispensou a programação de um sistema de aquisição de dados e permitiu a conversão dos arquivos de saída do ATP para diferentes formatos. Os respectivos fluxogramas e pseudocódigos foram apresentados em contribuição à literatura.

A análise comparativa entre a metodologia proposta e a execução manual evidenciou um ganho produtivo de aproximadamente 33 dias no tempo necessário para construção da base de dados. Por fim, destaca-se que a técnica apresentada, muito embora tenha sido aplicada exclusivamente para FAI, pode ser facilmente adaptada para diferentes estudos, por exemplo, análise de faltas de baixa impedância e demais distúrbios de qualidade da energia elétrica.

AGRADECIMENTOS

Os autores agradecem ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e a Coordenação de Pós Graduação em Engenharia Elétrica da UFCG (COPELE).

REFERÊNCIAS

Aguirre, S. and Rodriguez, A. (2017). Automation of a business process using robotic process automation (RPA): A case study. In *Workshop on Engineering Applications*, 65–71. Springer.

Bernardes, W.M.S., Asada, E.N., and Vieira, J.C.M. (2015). Topological processing of mutually coupled circuits for directional overcurrent protection. In *2015 IEEE Power Energy Society General Meeting*, 1–5.

Borges, T.T., Garcia, P.A.N., Carneiro Jr, S., and Pereira, J.L.R. (2012). Restabelecimento de sistemas de distribuição utilizando fluxo de potência ótimo. *Sba: Controle & Automação Sociedade Brasileira de Automatica*, 23(6), 737–748.

Braz, H.D.M. (2010). *Configuração de sistemas de distribuição usando um algoritmo genético sequencial.* Ph.D. thesis, Universidade Federal de Campina Grande.

Costa, P.P. (2011). Teoria dos grafos e suas aplicações.

Dijkstra, E.W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1), 269–271.

Lima, E.M., Junqueira, C.M.S., Coelho, R.A., Brito, N.S.D., and Souza, B.A. (2018). Modelagem probabilística do ruído em sistemas de distribuicão de energia elétrica. *Congresso Brasileiro de Automática - CBA 2018*.

Marini, A., Mortazavi, S., Piegari, L., and Ghazizadeh, M.S. (2019). An efficient graph-based power flow algorithm for electrical distribution systems with a comprehensive modeling of distributed generations. *Electric Power Systems Research*, 170, 229–243.

Mora, J.J., Bedoya, J.C., and Melendez, J. (2006). Extensive events database development using ATP and Matlab to fault location in power distribution systems. In *2006 IEEE/PES Transmission & Distribution Conference and Exposition: Latin America*, 1–6. IEEE.

Santos, W.C., Costa, F.B., Silva, J.A.C.B., Lira, G.R.S., Souza, B.A., Brito, N.S.D., and Paes, M.R.C. (2010). Automatic building of a simulated high impedance fault database. In *IEEE/PES Transmission and Distribution Conference and Exposition: Latin America (T&D-LA)*, 550–554. IEEE.

Santos, W.C., Souza, B.A., Brito, N.S.D., Costa, F.B., and Paes, M.R.C. (2013). High impedance faults: From field tests to modeling. *Journal of Control, Automation and Electrical Systems*, 24(6), 885–896.

Santos, W.C. (2016). *Identificação de Faltas de Alta Impedância em Sistemas de Distribuição.* Ph.D. thesis.

Souza, B.A., Brito, N.S.D., Neves, W.L.A., Dantas, K.M.C., Fontes, A.V., Silva, S.S.B., and Fernandes, A.B. (2005). Construção automática de bases de dados–uma experiência de P&D entre a Chesf e a UFCG. *Proc. 2005 Seminário Nacional de Produção e Transmissão de Energia Elétrica (XWIII SNPTEE), Curitiba.*

Souza, P.V.L. (2014). *Um método grafo-algébrico para projeto de ilhamento controlado em Sistemas Elétricos de Potência.* Ph.D. thesis, Universidade de São Paulo.

Van der Aalst, W.M.P., Bichler, M., and Heinzl, A. (2018). Robotic process automation.