

Implementação e Análise dos Fatores de Distribuição de Corrente (CTDF) para Métodos Lineares de Fluxo de Potência

Gabriel H. Limp, Giovani Junqueira

Abstract—Este artigo apresenta a fundamentação matemática e a implementação computacional dos Fatores de Distribuição de Corrente (CTDF) para métodos lineares de fluxo de potência, conforme proposto por Sauer (1981). São detalhadas as formulações dos CTDFs ground, slack-referenced e z-tie, bem como sua implementação em Python. Resultados experimentais são apresentados para sistemas IEEE 14 barras, Sauer 6 e Sauer 11, comparando os métodos AC, DC e CTDF.

Index Terms—CTDF, Fluxo de Potência, IEEE 14 barras, Python, Sistemas Elétricos, Análise de Redes, Sauer.

I. INTRODUÇÃO

A análise de fluxo de potência é fundamental para o planejamento e operação de sistemas elétricos. Métodos lineares, como o DC Load Flow, são amplamente utilizados devido à sua simplicidade e eficiência. Os Fatores de Distribuição de Corrente (CTDFs) permitem calcular rapidamente o impacto de perturbações de carga ou geração sobre os fluxos de linha, sem a necessidade de resolver novamente todo o sistema. Este trabalho apresenta a fundamentação matemática dos CTDFs, conforme Sauer [1], e sua implementação computacional em Python.

II. FUNDAMENTAÇÃO MATEMÁTICA DOS CTDFs

A. Modelo Linearizado de Fluxo de Potência (DC Load Flow)

Considerando um sistema de potência com n barras e m linhas, o fluxo de potência ativa entre as barras i e j é dado por:

$$P_{ij} = \frac{1}{x_{ij}}(\theta_i - \theta_j) \quad (1)$$

onde x_{ij} é a reatância da linha (i, j) e θ_i é o ângulo da barra i .

A equação nodal para todas as barras (exceto a slack) pode ser escrita como:

$$\mathbf{P} = \mathbf{B}\boldsymbol{\theta} \quad (2)$$

onde \mathbf{P} é o vetor de injeções de potência, \mathbf{B} é a matriz de susceptância reduzida, e $\boldsymbol{\theta}$ é o vetor de ângulos.

B. CTDFs: Definição Geral

O CTDF para uma linha k e barra m é definido como:

$$CTDF_{k,m} = \frac{\partial f_k}{\partial P_m} \quad (3)$$

onde f_k é o fluxo na linha k e P_m é a potência injetada na barra m .

C. CTDF Ground-Referenced

No caso **ground-referenced**, considera-se que todas as barras estão referenciadas a um "terra" comum (sem slack explícito). O fluxo na linha k entre barras i e j é:

$$f_k = \frac{1}{x_k}(\theta_i - \theta_j) \quad (4)$$

Com $\boldsymbol{\theta} = \mathbf{B}^{-1}\mathbf{P}$, temos:

$$CTDF_{k,m}^{(g)} = \frac{1}{x_k} (e_i^T \mathbf{B}^{-1} e_m - e_j^T \mathbf{B}^{-1} e_m) \quad (5)$$

onde e_i é o vetor canônico com 1 na posição i .

D. CTDF Slack-Referenced

No caso **slack-referenced**, uma barra de referência (slack) é explicitamente definida. A matriz \mathbf{B} é reduzida, removendo a linha e coluna da slack. O CTDF é dado por:

$$CTDF_{k,m}^{(s)} = \frac{1}{x_k} (e_i^T \mathbf{B}_{red}^{-1} e_m - e_j^T \mathbf{B}_{red}^{-1} e_m) \quad (6)$$

onde m e i, j não incluem a slack.

E. CTDF Z-Tie Referenced

No caso **z-tie referenced**, considera-se uma ligação fictícia de impedância z_{tie} entre a slack e o "terra". A matriz admitância é modificada para incluir z_{tie} , e o CTDF é:

$$CTDF_{k,m}^{(z)} = \frac{1}{x_k} (e_i^T \mathbf{B}_{tie}^{-1} e_m - e_j^T \mathbf{B}_{tie}^{-1} e_m) \quad (7)$$

onde \mathbf{B}_{tie} é a matriz \mathbf{B} modificada.

F. Resumo Matricial

De forma compacta, para todas as linhas e barras:

$$\mathbf{CTDF} = \mathbf{A}\mathbf{B}^{-1} \quad (8)$$

onde \mathbf{A} é a matriz de incidência das linhas.

III. IMPLEMENTAÇÃO EM PYTHON

A implementação foi realizada de forma modular, utilizando as bibliotecas `numpy`, `matplotlib` e `pandas`. O código está dividido em:

- **Modelos Elétricos:** Definição de classes para barras, linhas, cargas e geradores.
- **Sistemas:** Instanciação dos sistemas IEEE 14, Sauer 6 e Sauer 11.
- **Fluxo de Potência:** Métodos AC e DC.
- **CTDFs:** Funções para cálculo dos três tipos de CTDF.
- **Notebook principal:** Execução dos experimentos e geração dos gráficos.

A. Exemplo de Cálculo do CTDF Ground em Python

```
B = network.get_B()
A = network.incidence_matrix()
CTDF_ground = np.dot(A, np.linalg.inv(B))
```

B. Execução dos Testes

Os experimentos foram realizados no notebook `main.ipynb`, com os seguintes passos:

- 1) Comparação dos ângulos de barra (AC vs DC).
- 2) Comparação dos fluxos de linha (DC vs CTDFs).
- 3) Análise de perturbação de carga (+10%) e impacto nos fluxos.

IV. COMO EXECUTAR O PROJETO

- 1) Clone o repositório:

```
git clone <URL_DO_REPOSITORIO>
cd CTDF-for-linear-load-methods
```

- 2) (Opcional) Crie um ambiente virtual:

```
python -m venv venv
venv\Scripts\activate # Windows
source venv/bin/activate # Linux/Mac
```

- 3) Instale as dependências:

```
pip install -r requirements.txt
```

- 4) Execute o notebook:

```
jupyter notebook
```

- 5) Abra e execute as células do arquivo `main.ipynb`.

V. RESULTADOS DOS EXPERIMENTOS

Os principais resultados são apresentados em gráficos gerados pelo notebook, incluindo:

- Diferença dos ângulos de barra entre AC e DC.
- Comparação dos fluxos de linha via DC e CTDF (ground, slack, z-tie).
- Variação dos fluxos diante de aumento de carga.

Os resultados mostram que os CTDFs fornecem estimativas rápidas e precisas para pequenas perturbações, sendo especialmente úteis para estudos de sensibilidade.

VI. CONCLUSÃO

A implementação dos CTDFs em Python mostrou-se eficiente para análise de redes elétricas, permitindo estudos rápidos de sensibilidade e contingência. O framework desenvolvido é flexível e pode ser expandido para outros sistemas e métodos.

AGRADECIMENTOS

Agradecemos ao corpo docente do programa de pós-graduação e aos colegas pelo apoio e discussões construtivas.

REFERENCES

- [1] P. W. Sauer, "On The Formulation of Power Distribution Factors for Linear Load Flow Methods," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-100, no. 2, pp. 764-770, Feb. 1981.