

Leonardo Silva Vieira

Aplicação do Método WLS na Estimação de Tensões Desconhecidas em Barras de Sistemas de Potência

Uberlândia, MG

2025

Leonardo Silva Vieira

Aplicação do Método WLS na Estimação de Tensões Desconhecidas em Barras de Sistemas de Potência

Trabalho de Conclusão de Curso da Engenharia Elétrica da Universidade Federal de Uberlândia - UFU - Campus Santa Mônica, como requisito para a obtenção do título de Bacharel em Engenharia Elétrica.

Universidade Federal de Uberlândia - UFU
Faculdade de Engenharia Elétrica - FEELT

Orientador Prof. Dr. Geraldo Caixeta Guimarães
Coorientador: Thales Lima de Oliveira

Uberlândia, MG

2025

Resumo

Este trabalho investiga a aplicação do método dos mínimos quadrados ponderados (WLS) na estimação de estados de sistemas elétricos de potência, utilizando o ambiente MATLAB.

A metodologia proposta consiste na estimação das tensões desconhecidas em barras específicas do sistema. A pesquisa abrange o estudo teórico do método WLS, sua implementação computacional e a análise dos resultados obtidos em condições simuladas. Espera-se, com isso, alcançar maior precisão na estimação dos estados do sistema.

O objetivo principal foi estimar as magnitudes e os ângulos de tensão fasorial em um sistema teste de 9 barras, processando um conjunto de 2000 medições simuladas de tensão e ângulo ao longo do tempo para todas as barras. A metodologia incluiu a construção programática da matriz de admitância Y_{bus} a partir dos parâmetros da rede elétrica, a formulação do problema WLS considerando as variâncias das medições e a implementação de um script iterativo otimizado para análise temporal. Os resultados obtidos demonstram a capacidade do estimador WLS em reconstruir, de forma consistente, os perfis de tensão e ângulo para todos os instantes de tempo analisados, refletindo condições operacionais, como quedas de tensão em barras de carga e diferenças angulares associadas ao fluxo de potência. A comparação entre os dados de medição brutos e os estados estimados validou visualmente o efeito de filtragem característico do WLS. Conclui-se, portanto, que a implementação realizada é eficaz para a estimação de estado sequencial e constitui uma ferramenta validada para a análise do comportamento de sistemas de potência com base em dados fasoriais.

Palavras-chaves: Estimação de Estado; Método dos Mínimos Quadrados Ponderados (WLS); PMU (Unidade de Medida Fasorial); Análise Fasorial;

Abstract

This work investigates the application of the Weighted Least Squares (WLS) method in the state estimation of electric power systems using the MATLAB environment.

The proposed methodology consists of estimating the unknown voltages at specific buses in the system. The research encompasses the theoretical study of the WLS method, its computational implementation, and the analysis of the results obtained under simulated conditions. This approach is expected to achieve higher accuracy in estimating the system states.

The main objective was to estimate the phasor voltage magnitudes and angles in a 9-bus test system by processing a set of 2000 simulated voltage and angle measurements over time for all buses. The methodology included the programmatic construction of the admittance matrix Y_{bus} from the electrical network parameters, the formulation of the WLS problem considering the measurement variances, and the implementation of an iterative script optimized for temporal analysis. The results demonstrate the WLS estimator's ability to consistently reconstruct voltage and angle profiles across all analyzed time steps, reflecting operational conditions such as voltage drops at load buses and angular differences associated with power flow. The comparison between raw measurement data and the estimated states visually validated the characteristic filtering effect of the WLS algorithm. Therefore, it is concluded that the implemented method is effective for sequential state estimation and serves as a validated tool for analyzing the behavior of power systems based on phasor data.

Key-words: State Estimation; Weighted Least Squares (WLS); PMU (Phasor Measurement Unit); Phasor Analysis.

Lista de ilustrações

Figura 1 – Esquemático funcional de medições fasorial	19
Figura 2 – SEP modelo	20
Figura 3 – Circuito equivalente π das linhas de transmissão e distribuição	22
Figura 4 – Tensões estimadas da barra 2, 5 e 7 ao longo do tempo	27
Figura 5 – Ângulos estimadas da barra 2, 5 e 7 ao longo do tempo	28
Figura 6 – Tensão estimada X Medido na barra 5	29
Figura 7 – Tensão estimada nas barra 2, 5 e 7 durante um curto	30
Figura 8 – Tensão estimada na barra 2, 5 e 7 durante um curto	31
Figura 9 – Tensão estimada sobre a barra 5 durante um curto	31
Figura 10 – Angulo estimado sobre a barra 5 durante um curto	32

Lista de tabelas

Tabela 1 – Cargas conectadas às barras do sistema	20
Tabela 2 – Geradores síncronos do sistema	21
Tabela 3 – Dados das linhas de transmissão do sistema	21
Tabela 4 – Dados dos transformadores do sistema	21

Lista de abreviaturas e siglas

PMU	Phasor Measurement Unit, em português, Unidade de Medição Fasorial
PSP-UFU	Plataforma de Sistemas de Potência da Universidade Federal de Uberlândia
p.u	Por Unidade
y_{bus}	matriz admitância de barras
WLS	Estimação de Estado por Mínimos Quadrados Ponderados
GPS	Sistema de Posicionamento Global

Sumário

1	INTRODUÇÃO	9
1.1	Justificativas	9
1.2	Objetivos	10
1.2.1	Objetivos Específicos	10
1.3	Metodologia	10
1.4	Organização do Trabalho	12
2	REFERENCIAIS TEÓRICOS	13
2.1	Introdução à Estimação de Estados	13
2.1.1	Método dos Mínimos Quadrados Ponderados (WLS)	13
2.2	Estimação de Estado por Mínimos Quadrados Ponderados (WLS)	13
2.2.1	Formulação Matemática do WLS	13
2.2.2	Solução Iterativa do WLS	15
2.3	Detalhes da Implementação Computacional	16
2.4	Unidades de Medição Fasorial (PMUs)	18
3	METODOLOGIA	20
3.1	Sistema de Potência Modelo	20
3.2	Matriz admitância de barras (y_{bus})	21
3.2.1	Linha de transmissão	22
3.2.2	Transformadores	22
3.2.3	Ambiente e Estrutura do Código	23
3.2.4	Dados e Parâmetros do Sistema	23
3.2.5	Construção da Matriz de Admitância (Y_{bus})	23
3.2.6	Implementação do Estimador WLS Sequencial	24
3.2.7	Solução Iterativa e Armazenamento de Resultados	25
3.2.8	Visualização dos Resultados	25
4	RESULTADOS E DISCUSSÕES	26
4.1	Considerações iniciais	26
4.2	Análise dos resultados	26
4.2.1	Caso 1: Regime permanente	26
4.2.2	Caso 2: Falta trifásica para a terra	29
5	CONCLUSÃO	33

REFERÊNCIAS BIBLIOGRÁFICAS	35
ANEXO A – SCRIPT MATLAB	36
ANEXO B – SCRIPT MATLAB FUNÇÃO	45

1 Introdução

No presente trabalho, o estudo e a análise têm como base o sistema elétrico de potência (SEP), o qual agrega elementos interconectados, tais como geradores, transformadores, linhas de transmissão, barras e cargas. O comportamento de elementos físicos é descrito por um modelo matemático que reflete suas características elétricas.

Muitas ideias matemáticas surgiram a partir de problemas práticos, assim como a Matemática já desenvolvida passou a ser usada em situações novas e diversas. A habilidade de empregar matemática em situações concretas e em outras áreas do conhecimento humano consiste em tomar um problema prático relativamente complexo, transformá-lo em um modelo matemático, ou seja, traduzir a questão na linguagem de números, gráficos, tabelas, equações etc., e procurar uma solução que possa ser reinterpretada em termos da situação concreta original. (BASSANEZI, 2002)

Com base nesses modelos, é possível estabelecer a matriz de admitância nodal (Y_{bus}), elemento fundamental para a execução de análises essenciais, como o cálculo de fluxo de carga e a estimação de estados do sistema. A representação algébrica subjacente a essas formulações é um fator determinante, pois possibilita a implementação eficaz de algoritmos computacionais capazes de processar dados em alta velocidade e gerenciar a complexidade inerente a sistemas de grande porte.

Na estimação de estados de sistemas de potência, o método dos mínimos quadrados ponderados (WLS) constitui uma abordagem amplamente adotada, notadamente por sua capacidade de fornecer estimativas robustas a partir de um conjunto de medições redundantes e suscetíveis a ruído.

Recentemente, o advento e a proliferação das Unidades de Medição Sincronizada (PMUs) têm revolucionado o monitoramento de sistemas de potência ao fornecer medições síncronas de alta precisão dos fasores de tensão e corrente.

No escopo deste trabalho, propõe-se investigar a aplicação do método WLS empregando dados coletados por duas Unidades de Medição Sincronizada (PMUs), com o objetivo primordial de estimar as tensões em barras específicas do sistema. Esta pesquisa busca, assim, contribuir para o avanço da confiabilidade e eficiência na operação de sistemas elétricos.

1.1 Justificativas

As PMUs oferecem medições sincronizadas em tempo real, representando uma revolução no monitoramento de sistemas de potência. A implementação do método WLS

com dados de PMUs possibilita uma maior precisão na estimação de tensões em barras com medições limitadas, contribuindo para a evolução dos sistemas modernos de gerenciamento de energia. Este trabalho é relevante tanto do ponto de vista acadêmico quanto prático, pois busca combinar avanços teóricos com uma implementação aplicável em sistemas reais. (EHRENSPERGER, 2004)

Os benefícios do uso da medição de fasores sincronizados para a monitoração, operação e controle do sistema de potência têm sido bastante reconhecidos pelo setor elétrico no mundo. Tais benefícios e outros aspectos relacionados à aplicação das PMU são discutidos em detalhes nesta dissertação, em capítulos posteriores. (VIDE, 2013)

1.2 Objetivos

O principal objetivo deste trabalho é desenvolver e implementar uma metodologia para a estimação de estados em sistemas elétricos de potência utilizando o método dos mínimos quadrados ponderados (*Weighted Least Squares*-WLS) com dados de PMUs (*Phasor Measurement Units*). A base de dados usada (NAGLIC, 2022) como modelo de caso onde será aplicada a metodologia de WLS, para se obter resultados.

1.2.1 Objetivos Específicos

Os objetivos específicos do trabalho são aqui sumarizados:

- Revisão de literatura sobre o método dos mínimos quadrados ponderados (WLS) e a uso de PMUs em sistemas de potência.
- Obtenção da base de dados a ser utilizada no estudo.
- Implementação computacional do método WLS em uma base de dados composta por medições do SEP.
- Avaliação da precisão do método na estimação de tensões desconhecidas em barras do sistema.
- Comparação dos resultados obtidos com métodos tradicionais de estimação de estados.

1.3 Metodologia

Para alcançar os objetivos propostos, a metodologia deste trabalho compreende a geração de dados, a implementação do estimador de estado e a validação dos resultados.

Iniciou-se com a obtenção da base de dados necessária para as análises. Dada a escassez de dados públicos detalhados de Unidades de Medição Sincronizada (PMUs), optou-se pela geração de dados sintéticos. Utilizou-se um modelo pré-existente do sistema teste de 9 barras, cujos parâmetros já vêm estabelecidos e foram adotados.

Uma simulação dinâmica deste sistema foi realizada no software Plataforma de Sistemas de Potência da Universidade Federal de Uberlândia (PSP-UFU) por um período de 20 segundos com um passo de integração de 1 milissegundo. Desta simulação, foram extraídos os valores fasoriais "verdadeiros" de tensão e ângulo (V_{true}, θ_{true}) em todas as barras para cada instante de tempo. Os dados resultantes, representando as medições disponíveis ao estimador, foram então exportados para um arquivo no formato CSV.

Esta base de dados serviu como entrada para o estimador WLS em regime permanente sequencial. Crucialmente, os "valores verdadeiros" (V_{true}, θ_{true}) obtidos da simulação original foram preservados e utilizados posteriormente como referência para a validação da acurácia do estimador, permitindo quantificar o erro da estimação. A avaliação da convergência do algoritmo WLS, por outro lado, foi realizada internamente ao processo iterativo, baseada na norma da atualização do vetor de estado. Os scripts MATLAB desenvolvidos para a implementação e análise encontram-se nos Anexos A e B.

Neste trabalho, a implementação do método dos Mínimos Quadrados Ponderados (WLS) foi realizada no ambiente MATLAB, com o objetivo de estimar os estados de tensão e ângulo em um sistema elétrico de potência a partir de medições simuladas. A estrutura do algoritmo foi concebida de forma a explorar a correlação temporal entre os estados do sistema, adotando como palpite inicial para cada novo instante de tempo o vetor de estados estimado no instante anterior. Tal estratégia visa não apenas acelerar a convergência do método, como também garantir maior robustez frente a possíveis flutuações nos dados. A formulação matemática envolveu a construção da matriz de admitância (Y_{bus}), o cálculo da Jacobiana associada ao modelo de medição, e a iteração do algoritmo Gauss-Newton até a convergência, considerando tolerâncias numéricas pré-definidas. A escolha do MATLAB como plataforma de desenvolvimento se justifica por sua eficiência na manipulação de estruturas matriciais, ampla gama de funções numéricas e consolidação no meio acadêmico e industrial, além de sua capacidade de lidar com simulações temporais em larga escala de forma estruturada. Essa abordagem computacional, ao empregar o WLS em um contexto sequencial, permitiu a análise detalhada do comportamento dinâmico do sistema modelado, refletindo variações operacionais e destacando o potencial do método na reconstrução precisa de estados elétricos a partir de dados fasoriais.

1.4 Organização do Trabalho

O Capítulo 2 dedica-se à fundamentação teórica, discorrendo sobre os conceitos e tecnologias essenciais envolvidas no estudo. Nele, são abordados os princípios subjacentes às medições fasoriais sincronizadas e o papel relevante das Unidades de Medição Fasorial (PMUs - Phasor Measurement Units) na monitoramento moderna de sistemas elétricos de potência. Adicionalmente, este capítulo revisa a teoria do método dos Mínimos Quadrados Ponderados (WLS) para estimação de estados.

Em seguida, o Capítulo 3 concentra-se nos aspectos práticos da pesquisa. Este capítulo detalha o processo de desenvolvimento e implementação da ferramenta computacional em MATLAB, destinada à estimação de estado via WLS. Descreve-se como a teoria do método foi aplicada na criação do script para processar os dados de entrada e obter as estimativas de estado (tensões e ângulos).

Por fim, o Capítulo 4 apresenta os resultados numéricos e gráficos gerados pela ferramenta, realizando uma análise comparativa entre os valores estimados e os valores de referência (obtidos da simulação) para validar a eficácia da implementação.

2 Referenciais Teóricos

O referencial teórico apresenta os conceitos fundamentais para o desenvolvimento deste trabalho, incluindo a estimação de estados, o método WLS e as unidades de medição fasorial (PMUs).

2.1 Introdução à Estimação de Estados

A estimação de estados é uma técnica essencial no monitoramento de sistemas elétricos de potência. Ela fornece uma visão detalhada do sistema a partir de medições disponíveis, permitindo a tomada de decisões operacionais.

2.1.1 Método dos Mínimos Quadrados Ponderados (WLS)

O método dos mínimos quadrados ponderados é amplamente utilizado na estimação de estados devido por sua capacidade de lidar com medições redundantes e afetadas por ruído. Sua formulação matemática pode ser descrita por:

$$J(x) = \sum_{i=1}^m w_i [z_i - h_i(x)]^2, \quad (2.1)$$

onde $J(x)$ é a função objetivo, w_i são os pesos das medições, z_i são as medições, e $h_i(x)$ é a função que relaciona os estados x com as medições. (MONTICELLI, 1999)

2.2 Estimação de Estado por Mínimos Quadrados Ponderados (WLS)

A estimação de estado é um processo fundamental na operação e monitoramento de sistemas elétricos de potência. Seu objetivo é obter a melhor estimativa possível do estado operacional do sistema (tipicamente, as magnitudes e ângulos das tensões em todas as barras) com base em um conjunto de medições redundantes e sujeitas a erros, provenientes de diferentes pontos do sistema (ABUR; EXPÓSITO, 2004). O método padrão da indústria para essa finalidade é o Método dos Mínimos Quadrados Ponderados (Weighted Least Squares - WLS) (ABUR; EXPÓSITO, 2004)

2.2.1 Formulação Matemática do WLS

O WLS baseia-se na premissa de que existe uma relação funcional entre o vetor de estado do sistema, $\mathbf{x}$, e o vetor de medições esperado (ideal, sem erros), $\mathbf{h}(\mathbf{x})$. As medições

reais, $\mathbf{z}$, diferem das ideais devido a erros inerentes aos equipamentos de medição, $\mathbf{e}$. O modelo matemático é expresso como (ABUR; EXPÓSITO, 2004)

$$\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{e} \quad (2.2)$$

Onde:

$\mathbf{z} \in \mathbb{R}^m$: Vetor das medições reais disponíveis (dimensão m). No presente trabalho, este vetor contém as medições de magnitude de tensão e ângulo de fase para todas as N barras do sistema, obtidas a partir dos dados de entrada para um determinado instante:

$$\mathbf{z} = [V_{1,med}, \dots, V_{N,med}, \theta_{1,med}, \dots, \theta_{N,med}]^T$$

Portanto, $m = 2N$.

$\mathbf{x} \in \mathbb{R}^n$: Vetor de estado a ser estimado (dimensão n). Para um sistema de N barras, usualmente escolhe-se uma barra como referência angular (slack bus), cujo ângulo é fixado em 0. O vetor de estado inclui, então, os ângulos das $N - 1$ barras restantes e as magnitudes de tensão de todas as N barras:

$$\mathbf{x} = [\theta_2, \dots, \theta_N, V_1, \dots, V_N]^T$$

Neste caso, a barra 1 foi definida como slack, e a dimensão do vetor de estado é $n = (N - 1) + N = 2N - 1$.

$\mathbf{h}(\mathbf{x}) \in \mathbb{R}^m$: Vetor da função de medição. Esta função não-linear relaciona o vetor de estado $\mathbf{x}$ com os valores que seriam medidos se não houvesse erros. Para o conjunto de medições considerado neste trabalho (tensões e ângulos de todas as barras), a função $h(x)$ mapeia diretamente os elementos do vetor de estado estimado $\mathbf{x}$ para a estrutura do vetor de medições $\mathbf{z}$, adicionando o ângulo zero da barra de referência:

$$\mathbf{h}(\mathbf{x}) = [V_1, \dots, V_N, 0, \theta_2, \dots, \theta_N]^T$$

Note que os V_k e θ_k (para $k = 2..N$) em $\mathbf{h}(\mathbf{x})$ são os valores presentes no vetor de estado $\mathbf{x}$. A forma específica de $\mathbf{x}$ depende do tipo e localização das medições disponíveis no sistema,

$\mathbf{e} \in \mathbb{R}^m$: Vetor de erros de medição. Assume-se que os erros são estatisticamente independentes, possuem média zero e suas variâncias são conhecidas. O objetivo do WLS é encontrar a estimativa do vetor de estado, $\hat{\mathbf{x}}$, que minimiza a soma ponderada dos quadrados dos resíduos (diferença entre medições reais e calculadas) $J(\mathbf{x})$ (ABUR; EXPÓSITO, 2004)

$$\min_{\mathbf{x}} J(\mathbf{x}) = [\mathbf{z} - \mathbf{h}(\mathbf{x})]^T \mathbf{W} [\mathbf{z} - \mathbf{h}(\mathbf{x})] \quad (2.3)$$

A matriz de pesos $\mathbf{W} \in \mathbb{R}^{m \times m}$ é tipicamente escolhida como o inverso da matriz de covariância dos erros de medição, $\mathbf{R}$:

$$\mathbf{W} = \mathbf{R}^{-1} = E[\mathbf{e}\mathbf{e}^T]^{-1} \quad (2.4)$$

Considerando a hipótese de erros não correlacionados, $\mathbf{R}$ torna-se uma matriz diagonal, e $\mathbf{W}$ também será diagonal. Os elementos da diagonal de $\mathbf{W}$ são o inverso das variâncias (σ_i^2) dos erros de medição correspondentes:

$$W_{ii} = \frac{1}{\sigma_i^2} \quad (2.5)$$

Onde σ_i é o desvio padrão da i -ésima medição. Isso confere maior peso às medições mais precisas (com menor variância/desvio padrão) na determinação do estado estimado. No código implementado, foram atribuídos desvios padrão $\sigma_V = 0.05$ pu para medições de tensão e $\sigma_A = 0.01$ rad para medições de ângulo, e a matriz $\mathbf{W}$ foi construída com base nesses valores.

2.2.2 Solução Iterativa do WLS

Como a função de medição $\mathbf{h}(\mathbf{x})$ é geralmente não-linear (embora seja linear neste caso específico), a minimização de $J(\mathbf{x})$ requer um método iterativo. A condição necessária para minimizar $J(\mathbf{x})$ é que o gradiente de $J(\mathbf{x})$ em relação a $\mathbf{x}$ seja zero:

$$\frac{\partial J(\mathbf{x})}{\partial \mathbf{x}} = -2 \left[\frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}} \right]^T \mathbf{W}[\mathbf{z} - \mathbf{h}(\mathbf{x})] = \mathbf{0} \quad (2.6)$$

Definindo a matriz Jacobiana $\mathbf{H}(\mathbf{x})$ como a matriz das derivadas parciais $\mathbf{H}(\mathbf{x})$ em relação a $\mathbf{x}$

$$\mathbf{H}(\mathbf{x}) = \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}} \in \mathbb{R}^{m \times n} \quad (2.7)$$

A condição de otimalidade (2.7) se torna:

$$\mathbf{H}^T(\mathbf{x})\mathbf{W}[\mathbf{z} - \mathbf{h}(\mathbf{x})] = \mathbf{0} \quad (2.8)$$

Para resolver iterativamente, linearizamos $\mathbf{h}(\mathbf{x})$ em torno da estimativa atual $\mathbf{x}_k$ usando a expansão de Taylor de primeira ordem:

$$\mathbf{h}(\mathbf{x}) \approx \mathbf{h}(\mathbf{x}_k) + \mathbf{H}(\mathbf{x}_k)(\mathbf{x} - \mathbf{x}_k) \quad (2.9)$$

Substituindo (2.9) na condição de otimalidade (2.7) e definindo $\mathbf{x} = \mathbf{x}_{k+1}$ (a próxima estimativa) e $\Delta\mathbf{x}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$, obtemos as equações normais do problema de mínimos quadrados (ABUR; EXPÓSITO, 2004)

$$\mathbf{H}^T(\mathbf{x}_k)\mathbf{W}[\mathbf{z} - (\mathbf{h}(\mathbf{x}_k) + \mathbf{H}(\mathbf{x}_k)\Delta\mathbf{x}_k)] = \mathbf{0} \quad (2.10)$$

Rearranjando os termos, obtemos as equações normais do problema de mínimos quadrados:

$$\mathbf{H}^T(\mathbf{x}_k)\mathbf{W}\mathbf{H}(\mathbf{x}_k)\Delta\mathbf{x}_k = \mathbf{H}^T(\mathbf{x}_k)\mathbf{W}\mathbf{z} - \mathbf{h}(\mathbf{x}_k) \quad (2.11)$$

Define-se a Matriz de Ganho $\mathbf{G}(\mathbf{x}_k)$:

$$\mathbf{G}(\mathbf{x}_k) = \mathbf{H}^T(\mathbf{x}_k)\mathbf{W}\mathbf{H}(\mathbf{x}_k) \in \mathbb{R}^{n \times n} \quad (2.12)$$

A atualização do estado $\Delta\mathbf{x}_k$ é então encontrada resolvendo o sistema linear:

$$\mathbf{G}(\mathbf{x}_k)\Delta\mathbf{x}_k = \mathbf{H}^T(\mathbf{x}_k)\mathbf{W}[\mathbf{z} - \mathbf{h}(\mathbf{x}_k)] \quad (2.13)$$

E a próxima estimativa do estado é:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta\mathbf{x}_k \quad (2.14)$$

Este processo é repetido a partir de uma estimativa inicial $\mathbf{x}_0$ até que a atualização $\Delta\mathbf{x}_k$ seja suficientemente pequena, indicando convergência.(ABUR; EXPÓSITO, 2004)

2.3 Detalhes da Implementação Computacional

Estimativa Inicial ($\mathbf{x}_0$): Para o primeiro instante de tempo analisado ($t = 1$), utilizou-se $\mathbf{x}_0$ que continha ângulos de 0 rad e tensões de 1.0 pu palpite inicial. Para os instantes subsequentes ($t > 1$), utilizando o vetor de estado final convergido do instante anterior ($\mathbf{x}_{final,t-1}$) como estimativa inicial para o instante t . Esta abordagem visa acelerar a convergência, aproveitando a correlação temporal dos estados do sistema.

Cálculo da Jacobiana ($\mathbf{H}$): Dada a natureza das medições (tensões e ângulos) e a definição do vetor de estado $\mathbf{x}$, a função $\mathbf{h}(\mathbf{x})$ resulta em uma relação linear entre as variáveis de estado e as medições calculadas. Consequentemente, a matriz Jacobiana

$$\mathbf{H} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \quad (2.15)$$

é constante e não precisa ser recalculada a cada iteração. Sua estrutura, conforme detalhado na construção do código, contém blocos de matrizes identidade e matrizes nulas, refletindo as derivadas

$$\frac{\partial V_i}{\partial \theta_j} = 0 \quad \text{e} \quad \frac{\partial \theta_i}{\partial V_j} = 0. \quad (2.16)$$

Loop Iterativo O *while* loop implementado no código executa as etapas do método de Gauss-Newton: cálculo de $\mathbf{h}(\mathbf{x}_k)$, cálculo do resíduo

$$\Delta \mathbf{z}_k = \mathbf{z} - \mathbf{h}(\mathbf{x}_k) \quad (2.17)$$

cálculo da matriz de ganho $\mathbf{G}$ (constante neste caso, pois $\mathbf{H}$ e $\mathbf{W}$ são constantes), cálculo do lado direito $\mathbf{H}^\top \mathbf{W} \Delta \mathbf{z}_k$, e solução do sistema linear $\mathbf{G} \Delta \mathbf{x}_k = \text{rhs}$ usando o operador “\” (mldivide) do MATLAB para obter $\Delta \mathbf{x}_k$. Este operador não realiza diretamente a inversão da matriz $\mathbf{A}$, mas sim emprega algoritmos numéricos otimizados que levam em consideração as propriedades da matriz (como triangularidade, esparsidade ou simetria positiva definida), escolhendo automaticamente o método mais adequado para a resolução. Essa abordagem é preferível à inversão explícita da matriz $\mathbf{A}^{-1}$, tanto por razões de eficiência computacional quanto por estabilidade numérica, uma vez que evita os erros de arredondamento e a amplificação de perturbações associadas à inversão direta. Portanto, a utilização do operador “\” contribui para uma implementação mais robusta e eficiente do método dos mínimos quadrados ponderados (WLS).

Critério de Convergência O loop itera até que a norma Euclidiana do vetor de atualização do estado, $\|\Delta \mathbf{x}_k\|_2$, seja menor que uma tolerância pré-definida ($\text{tol} = 1\text{e-}5$), ou até que um número máximo de iterações ($\text{max}_{iter} = 10$) seja atingido, tal limitação atende mais que suficiente para esse caso específico, mas também para evitar loops infinitos ou seja caso de não convergência.

Processamento Temporal: O processo do método WLS completo foi encapsulado em um loop “for” que itera sobre todos os “numInstantes” disponíveis no arquivo de dados de entrada como a quantidade de amostras. Os resultados estimados de tensão e ângulo para cada instante convergido foram armazenados para posterior análise e plotagem temporal. Casos de não convergência para um determinado instante foram registrados, e utilizou o último resultado convergido disponível como estimativa inicial para o instante seguinte evitando um crash.

Saída: Ao final do processo, o estado estimado final $\hat{\mathbf{x}}$ é obtido para cada instante de tempo. Além disso, o índice de performance $J(\hat{\mathbf{x}})$, que quantifica o ajuste final ponderado entre medições e estimativas, pode ser calculado como uma medida da qualidade da estimação.

A implementação do método dos Mínimos Quadrados Ponderados (WLS) em Matlab se justifica por diversas razões técnicas e práticas que o tornam ideal para aplicações em engenharia elétrica, especialmente na estimação de estados em sistemas de potência. Primeiramente, o MATLAB é amplamente reconhecido por sua capacidade de lidar com

operações matriciais e algébricas de forma eficiente e intuitiva, característica fundamental para a formulação do método WLS, que envolve manipulações matemáticas intensivas, como o cálculo de derivadas parciais (Jacobianas), a montagem da matriz de Ganho e a resolução eficiente de sistemas de equações lineares em cada iteração. Além disso, sua extensa biblioteca de funções numéricas, combinada com a clareza sintática de sua linguagem, permite desenvolver algoritmos complexos com menos linhas de código e maior legibilidade, facilitando a validação, manutenção e reprodutibilidade dos experimentos.

Outro fator relevante na escolha do MATLAB está relacionado ao seu uso consolidado no meio acadêmico e industrial, o que garante ampla documentação, suporte técnico e disponibilidade de exemplos aplicados a sistemas de potência. No contexto deste experimento, que envolve a estimação de tensões em barras a partir de dados simulados e a construção de Y_{bus} , o MATLAB oferece recursos apropriados tanto para a simulação quanto para a verificação da qualidade das estimativas. Dessa forma, a adoção desse ambiente computacional contribui para a confiabilidade, flexibilidade e escalabilidade da implementação do método WLS, tornando-o uma escolha justificada para o desenvolvimento da aplicação proposta.

Embora o MATLAB seja uma ferramenta proprietária, é relevante destacar que o GNU Octave representa uma alternativa viável e gratuita para implementações semelhantes, sobretudo em contextos onde o acesso a licenças comerciais representa uma dificuldade ou um custo proibitivo.

2.4 Unidades de Medição Fasorial (PMUs)

Os Sistemas de Medição Fasorial Sincronizada (SMFS) consistem em uma arquitetura de medição que permite a coleta simultânea de fasores de grandezas elétricas em diversos locais geográficos. Essa coleta é realizada por meio das Unidades de Medição Fasorial (PMUs), dispositivos fundamentais do SMFS, que, por sua vez, transmitem os dados para um Concentrador de Dados Fasoriais (PDC). O PDC tem a função de coletar, agregar e armazenar os dados recebidos das PMUs, conforme ilustrado na Figura 1.

A PMU é o componente central do SMFS. Sua operação envolve a aquisição de sinais de tensão e corrente das barras do sistema elétrico, o processamento digital desses sinais e a estimação precisa dos fasores de tensão e corrente, bem como da frequência local e sua taxa de variação. Após o processamento, essas informações sincronizadas são transmitidas para o Concentrador de Dados Fasoriais (PDC).

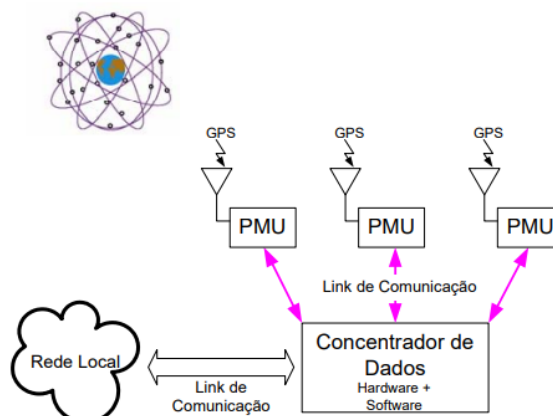
Uma característica distintiva das PMUs é sua sincronização temporal precisa, geralmente obtida via satélite (GPS). Essa sincronização confere às medições um carimbo de tempo uniforme em toda a área monitorada, permitindo uma comparação direta e significativa de grandezas elétricas medidas em pontos distintos da rede, mesmo a longas

distâncias ([ANDRADE SÔNIA, 2008](#)). Tal capacidade de adquirir dados de grandes áreas (wide area data) em tempo real revoluciona o monitoramento e o controle dinâmico de sistemas de potência ([ANDRADE SÔNIA, 2008](#)).

As PMUs possibilitam, assim, medições de alta precisão e sincronização em tempo real, características essenciais para aplicações avançadas em sistemas de potência modernos, onde a confiabilidade e a precisão das informações são primordiais.

Conforme as diretrizes de padrões como o IEEE Std C37.118 (que substituiu o IEEE 1344), os fasores calculados pelas PMUs são transmitidos através de canais de comunicação apropriados para o PDC. Ao agregar as medições sincronizadas de magnitude e ângulo dos fasores de tensão e corrente das barras equipadas com PMUs, o PDC fornece uma visão detalhada e quase instantânea do estado operacional do sistema, fundamental para aplicações como a estimação de estados e o monitoramento da segurança dinâmica.

Figura 1 – Esquemático funcional de medições fasorial



Adaptado de ([EHRENSPERGER, 2004](#)).

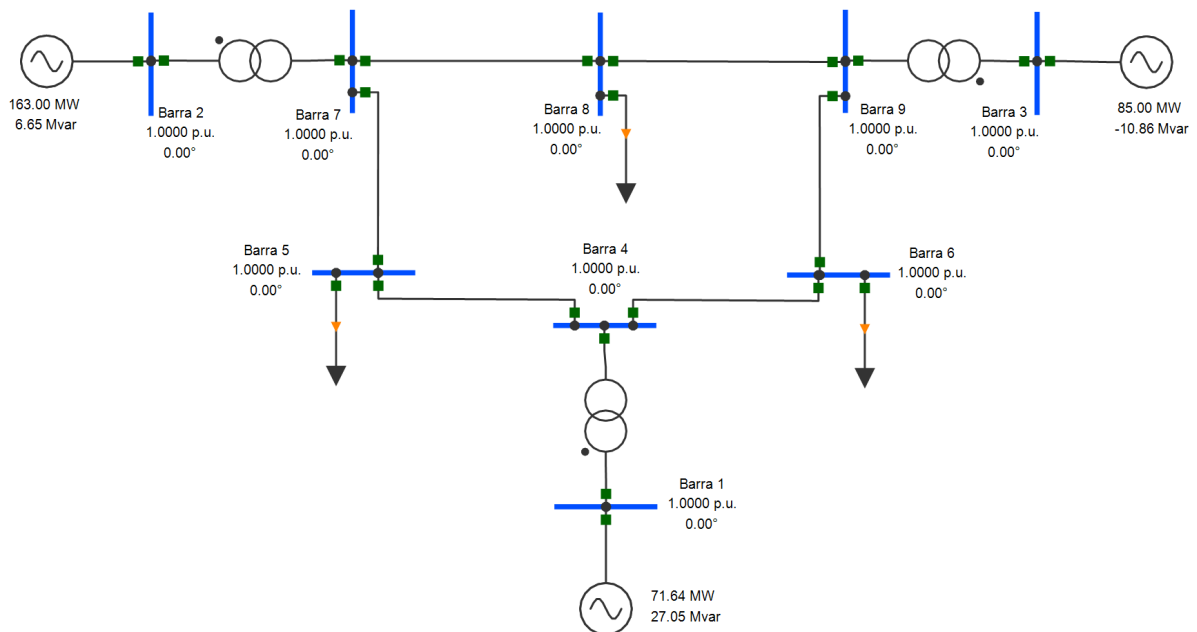
3 Metodologia

Os tópicos a seguir detalharão o SEP modelo utilizado e das etapas empregadas no desenvolvimento da ferramenta computacional para estimação de estado sequencial via Mínimos Quadrados Ponderados (WLS), utilizando o ambiente MATLAB (The MathWorks, Inc.). O código-fonte correspondente a esta implementação encontra-se referenciado nos Anexos A e B. A estrutura lógica do script principal segue uma sequência de etapas bem definidas, desde a entrada de dados até a visualização dos resultados.

3.1 Sistema de Potência Modelo

O sistema elétrico simulado, apresentado na Figura 2, foi utilizado como base para a obtenção dos dados. Os parâmetros utilizados sobre transformadores, linhas de transmissão e até mesmo de geradores e cargas do SEP se encontram especificados nas tabelas 1 até 4

Figura 2 – SEP modelo



Autoria própria

Tabela 1 – Cargas conectadas às barras do sistema

Barra	P_{load} (pu)	Q_{load} (pu)
5	1,25	0,50
6	0,90	0,30
8	1,00	0,35

Tabela 2 – Geradores síncronos do sistema

Barra	P_{gen} (pu)	Q_{gen} (pu)
1	0,716	0,27
2	1,630	0,06
3	0,850	-0,01

Tabela 3 – Dados das linhas de transmissão do sistema

De	Para	R (pu)	X (pu)	B (pu)	Tap
7	8	0,0085	0,0720	0,149	1,0
8	9	0,0119	0,1008	0,209	1,0
7	5	0,0320	0,1610	0,306	1,0
5	4	0,0100	0,0850	0,176	1,0
4	6	0,0170	0,0920	0,158	1,0
6	9	0,0390	0,1700	0,358	1,0

Tabela 4 – Dados dos transformadores do sistema

De	Para	R (pu)	X (pu)	B (pu)	Tap
2	7	0,0	0,0625	0,0	0,0783
1	4	0,0	0,0576	0,0	0,0717
3	9	0,0	0,0586	0,0	0,0600

3.2 Matriz admitância de barras (y_{bus})

A criação de matriz de admitância para das equações 2.1 presente no livro (ARRILLAGA; WATSON, 2001), assim também 2.2

$$Y_{ij} = -\bar{y}_{ij} \quad (2.1)$$

$$Y_{ii} = \bar{y}_{ii} + \sum_{\substack{j=1 \\ j \neq i}}^n \bar{y}_{ij} \quad (2.2)$$

Em que:

Y_{ii} Elementos da diagonal principal

Y_{ij} Elementos fora da diagonal principal

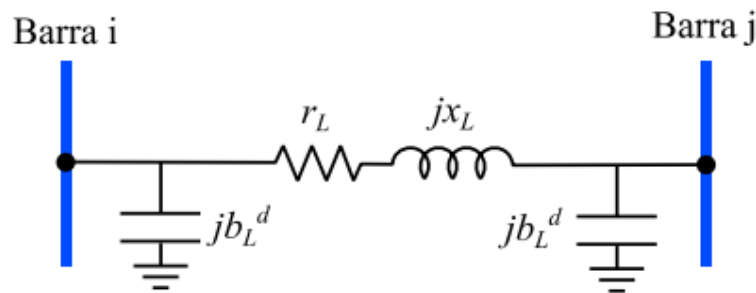
$\bar{y}_{ii}$ Admitância dos elementos shunt e dos parâmetros shunt de elementos de ramos

$\bar{y}_{ij}$ Admitância dos parâmetros série de elementos de ramos

3.2.1 Linha de transmissão

Um sistema de potência é composto por uma rede interligada de elementos passivos e ativos, majoritariamente constituído por linhas de transmissão e transformadores. As linhas de transmissão e distribuição são modeladas por seu circuito equivalente do tipo π (π), que representa uma impedância série ($r_L + jx_L$) e duas susceptâncias capacitivas shunt (b_L^t) localizadas nas extremidades da linha. Tanto nas linhas quanto nos transformadores, a admitância inserida nas equações (2.1) e (2.2) é obtida por meio do inverso da impedância. A Figura 3 ilustra o modelo equivalente adotado.

Figura 3 – Circuito equivalente π das linhas de transmissão e distribuição



Fonte: Adaptado de (OLIVEIRA, 2019).

3.2.2 Transformadores

Os transformadores são representados por seus circuitos equivalentes para determinar como eles contribuem para a matriz Y_{bus} . Os principais componentes a considerar são:

- Impedância Série ($R + jX$): Geralmente é fornecida em pu (por unidade) na base do próprio transformador ou na base do sistema. Visível na tabela 4, estas são as colunas 2 e 3
- Relação de Taps (*Taps Ratio*): São uteis para ajustar ligeiramente a relação de espira, a fim de controlar a tensão.
- Admitância Shunt (Ramo de Magnetização): Representa as perdas no núcleo. Em modelagem por ser pequena admitância série ocorre que não é considerada assim como nesse modelo.
- Defasamento Angular (*Phase Shift*): Pode introduzir uma diferença de ângulo entre os lados primário e secundário. o que não ocorreu nesse caso.

3.2.3 Ambiente e Estrutura do Código

A escolha do MATLAB como ambiente de desenvolvimento justifica-se por suas robustas capacidades de cálculo numérico, manipulação de matrizes e visualização gráfica, características essenciais para problemas de métodos matemáticos aplicados como a estimação de estado. O script principal foi estruturado nas seguintes etapas lógicas:

- Inicialização e limpeza do ambiente MATLAB.
- Carregamento e pré-processamento dos dados de medição.
- Definição dos parâmetros elétricos do sistema (linhas, transformadores).
- Construção programática da matriz de admitância (Y_{bus}).
- Configuração dos parâmetros e estruturas de dados para o WLS.
- Implementação do loop iterativo WLS para cada instante de tempo.
- Armazenamento dos resultados estimados sequencialmente.
- Pós-processamento e visualização gráfica dos resultados.

3.2.4 Dados e Parâmetros do Sistema

A primeira etapa operacional do script consiste no carregamento dos dados de medição simulados, armazenados em um arquivo no formato CSV `databus_psp.csv`. A função (`readmatrix`) do MATLAB foi utilizada para importar esses dados para uma matriz numérica (`dataMatrix`). A partir desta matriz, determinou-se o número de instantes de tempo (`numInstantes`) e o número de barras do sistema (`numBarras`). Em seguida, os dados foram pré-processados para extrair as séries temporais de magnitude de tensão V_{true} e ângulo de fase Θ_{true} para todas as barras. Os parâmetros fixos do sistema como os dados de linhas `dados_linhas` e transformadores `dados_transformadores`, incluindo impedâncias R e X, susceptância shunt (B) e relação de tap, foram definidos diretamente no código através de matrizes MATLAB.

3.2.5 Construção da Matriz de Admitância (Y_{bus})

Para relacionar as tensões e correntes nodais do sistema, a matriz de admitância Y_{bus} foi construída programaticamente. Inicializou-se uma matriz quadrada complexa Y_{bus} de dimensões $N \times N$ (onde N é "numBarras") com zeros. O script então itera sobre todos os ramos do sistema (linhas e transformadores, contidos em `dados_ramos`). Para cada ramo conectando as barras "de" e "para", calculou-se a admitância série $y_{serie} = 1 / (r + jx)$. A admitância shunt total do ramo `jb_sh` foi dividida por dois $y_{sh} = 1jb_sh/2$ para

representar o modelo π conforme ilustrado na figura 3, onde metade é alocada em cada barramento. Os elementos da Y_{bus} foram então preenchidos usando as formulações padrão, tratando adequadamente os taps de transformadores (tap):

```

1 Ybus(de,de) = Ybus(de,de) + y_serie/(tap^2) + y_sh;
2 Ybus(para,para) = Ybus(para,para) + y_serie + y_sh;
3 Ybus(de,para) = Ybus(de,para) - y_serie/tap;
4 Ybus(para,de) = Ybus(para,de) - y_serie/tap;

```

Deste modo, representa-se a correta topologia e dos parâmetros do SEP na matriz Y_{bus} .

3.2.6 Implementação do Estimador WLS Sequencial

O núcleo da ferramenta é a implementação do estimador WLS aplicado sequencialmente. Um loop "for" externo itera de "t = 1" até "numInstantes" (no caso 2001 iterações). Dentro deste loop:

- Vetor de Medição $\mathbf{z}_t$: As medições de tensão e ângulo para o instante específico "t" foram extraídas das matrizes $\mathbf{V_true}$ e $\mathbf{\Theta_true}$ (com ângulos convertidos para radianos) e concatenadas em um vetor coluna $\mathbf{z}_t$ de dimensão $m \times 1$ ($m = 2N$).
- Vetor de Estado ($\mathbf{x}$): Definiu-se a estrutura do vetor de estado "x" de dimensão $n \times 1$ ($n = 2N - 1$), contendo os ângulos das barras não-referência (θ_2 a θ_N) seguidos pelas magnitudes de tensão de todas as barras (V_1 a V_N). A barra 1 foi designada como referência angular $\text{slack_bus} = 1$.
- Estimativa Inicial ($\mathbf{x}_0$): Implementou-se a estratégia de "Flat Start" para "t=1" (tensões 1.0 pu, ângulos 0 rad) e "Warm Start" para "t>1", utilizando o resultado convergido do instante anterior $\mathbf{x_anterior_final}$ como estimativa inicial $\mathbf{x_inicial_t}$.
- Matriz de Pesos ($\mathbf{W}$): Uma matriz diagonal " $\mathbf{W}$ " ($m \times m$) foi construída com base nos desvios padrão assumidos para as medições (σ_V, σ_A), onde $W_{ii} = 1/\sigma_i^2$.
- Função de Medição ($\mathbf{h}(\mathbf{x})$): Foi implementada como uma função MATLAB `calculate_h_implemente` vide anexo B, que recebe o vetor de estado "x" e retorna o vetor de medições esperadas "h" com a mesma estrutura de "z", incluindo o ângulo zero da barra de referência.
- Matriz Jacobiana ($\mathbf{H}$): Calculou-se a matriz Jacobiana $\mathbf{H} = \partial \mathbf{h} / \partial \mathbf{x}$ ($m \times n$). Devido à relação linear entre "h(x)" e "x" neste caso específico (medições diretas de V e θ), a matriz "H" resultou constante e foi calculada uma única vez fora do loop temporal. Sua estrutura esparsa, contendo blocos de identidade e zeros, foi montada programaticamente.

3.2.7 Solução Iterativa e Armazenamento de Resultados

Dentro do loop temporal, para cada instante "t", um loop "while" interno implementou o processo iterativo para resolver o problema WLS:

- Cálculo do Resíduo: $\text{delta_z} = \text{z_t} - \text{calculate_h}(\text{x_atual}, \dots)$
- Cálculo da Matriz de Ganho: $G = H' W H$
- Cálculo do Lado Direito: $\text{rhs} = H' W \text{delta_z}$
- Solução do Sistema Linear: A atualização do estado delta_x foi obtida resolvendo $G \text{delta_x} = \text{rhs}$ através do operador "(mldivide)" do MATLAB: $\text{delta_x} = G \backslash \text{rhs}$. Esta abordagem é preferível à inversão explícita ($\text{inv}(G)$) por ser mais eficiente e numericamente estável.
- Atualização do Estado: $\text{x_novo} = \text{x_atual} + \text{delta_x}$
- Verificação de Convergência: O loop "while" terminava se a norma Euclidiana da atualização $\text{norm}(\text{delta_x})$ fosse menor que a tolerância "tol" ou se o número máximo de iterações max_iter fosse atingido. Após a convergência (ou falha) do loop interno, os resultados foram tratados: se convergisse, as tensões V_final e ângulos $A_final_completo_deg$ finais eram extraídos de x_final_t e armazenados nas matrizes de histórico pré-alocadas $V_estimado_historico$, $A_estimado_historico_deg$. O estado final também era salvo para uso no "Warm Start" do próximo instante $\text{x_anterior_final} = \text{x_final_t}$. Em caso de não convergência, registrava-se a falha e mantinham-se os valores "NaN" (valor nulo) no histórico para aquele instante.

3.2.8 Visualização dos Resultados

Como etapa final, após processar todos os instantes de tempo, o script utilizou as matrizes de histórico $V_estimado_historico$, $A_estimado_historico_deg$ e o vetor de tempo para gerar visualizações gráficas. Foram utilizadas funções padrão de plotagem do MATLAB (`figure`, `plot`, `title`, `xlabel`, `ylabel`, `legend`, `grid on`) para criar gráficos dos perfis estimados de tensão e ângulo ao longo do tempo para barras selecionadas, bem como gráficos comparativos entre as medições originais e os valores estimados para análise visual da qualidade da estimação uma vez que se sobrepõem, confirmando a precisão estabelecida de $1 \cdot 10^{-5}$.

4 Resultados e Discussões

4.1 Considerações iniciais

Nesta seção, apresentam-se os resultados obtidos a partir da aplicação do estimador de estados via Mínimos Quadrados Ponderados (WLS) sobre um conjunto de dados de medição sequenciais referentes ao sistema teste de 9 barras, abrangendo um total de 2001 amostras (instantes de tempo). O cenário avaliado simula o uso de Unidades de Medição Fasorial (PMUs), dispositivos que fornecem medições sincronizadas de magnitude de tensão e ângulo de fase nas barras do sistema.

Dessa forma, o algoritmo de estimação foi desenvolvido utilizando exclusivamente essas grandezas como entrada — valores de tensão e ângulos de fase, o que permitiu emular um ambiente operacional compatível com sistemas modernos de monitoramento. A análise dos resultados demonstrou que o método apresentou boa robustez e rápida convergência, sendo capaz de atingir a solução estimada, na maioria dos casos, em menos de duas iterações por amostra temporal, o que evidencia sua eficiência computacional e viabilidade prática.

4.2 Análise dos resultados

Para analisar os resultados do estimador de estados via Mínimos Quadrados Ponderados (WLS) são analisados dois estudos de caso: um primeiro sem perturbações significativas no sistema, ou seja, resultados em regime permanente e um segundo com a aplicação de uma falta trifásica em uma das barras.

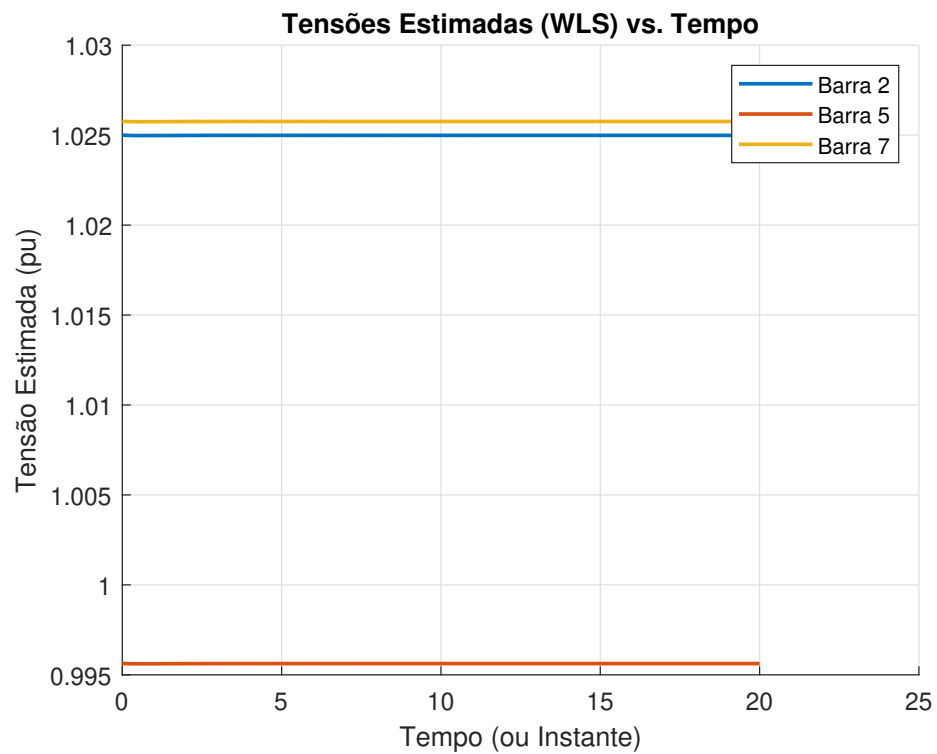
4.2.1 Caso 1: Regime permanente

Na primeira análise, a Figura 4 ilustra o comportamento das tensões nas barras 2, 5 e 7. A barra 2 está conectada a um gerador, enquanto a barra 5 representa uma carga significativa e, portanto, constitui o principal ponto de interesse. A barra 7, por sua vez, situa-se entre essas duas, funcionando como um elo intermediário. No comparativo entre a barra 2 (gerador, espera-se mais alta) e a barra 5 (carga, espera-se mais baixa), observa-se uma queda de tensão coerente com o esperado, resultado da característica da carga instalada nesse ponto do sistema.

A Figura 5 apresenta as diferenças angulares entre as barras, fator diretamente relacionado ao fluxo de potência ativa. Diferenças angulares maiores indicam um fluxo de potência mais intenso. No caso apresentado, como as cargas são estáticas, as diferenças

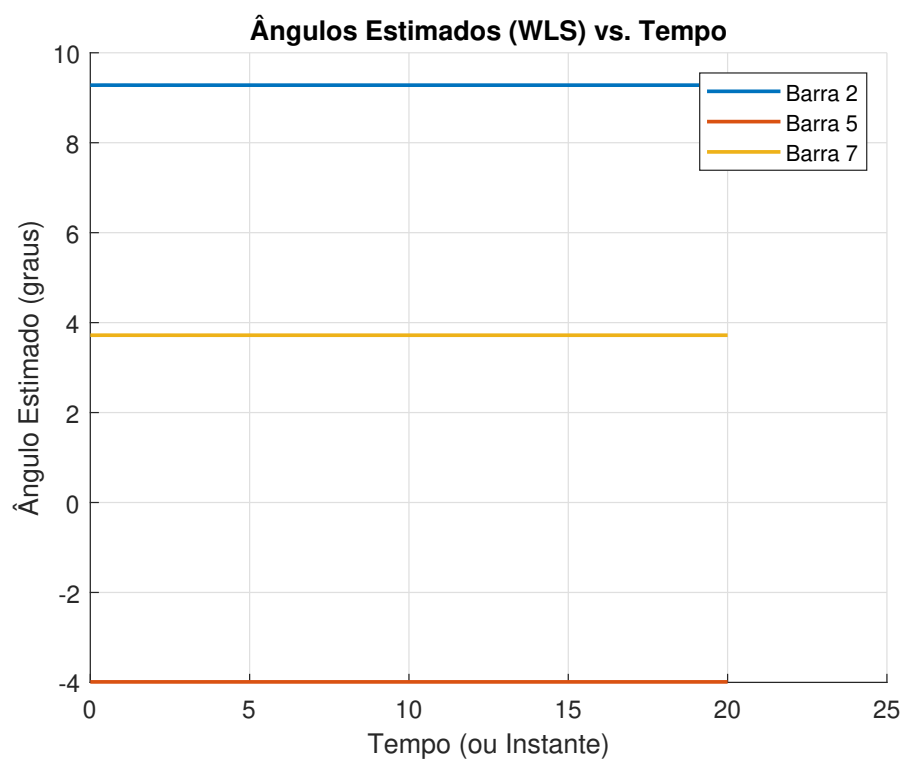
angulares permanecem praticamente constantes ao longo do tempo, não havendo variação significativa no perfil de fluxo de potência. Ressalta-se que a barra 1 não é incluída na análise de ângulo de fase, pois foi adotada como referência (ângulo nulo), o que é uma premissa comum na modelagem do sistema.

Figura 4 – Tensões estimadas da barra 2, 5 e 7 ao longo do tempo



Autoria própria

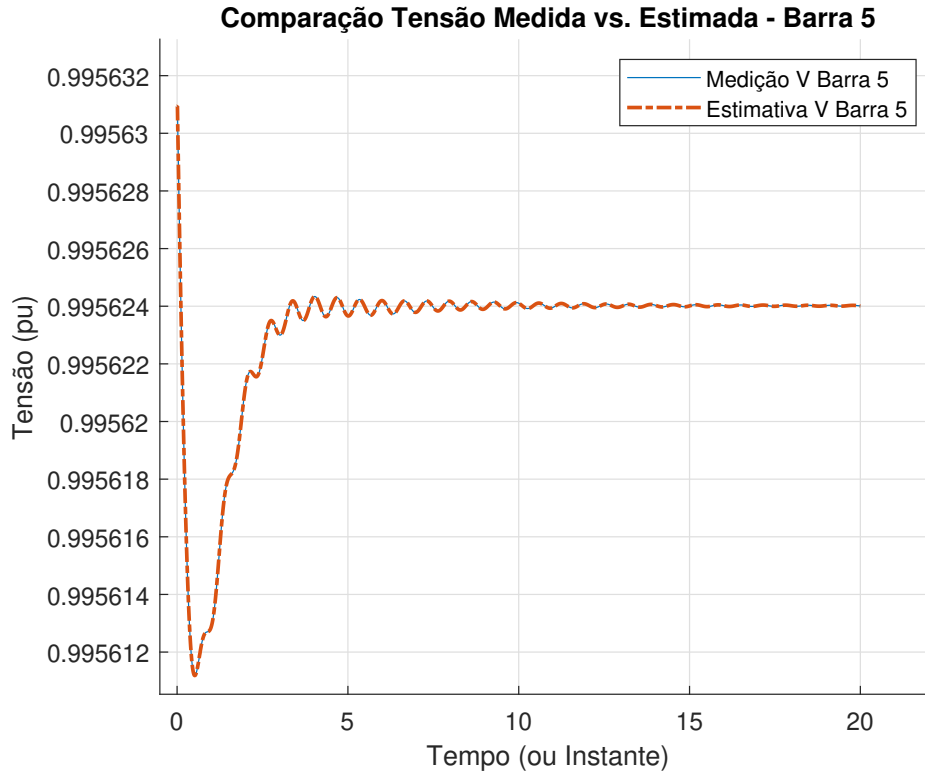
Figura 5 – Ângulos estimados da barra 2, 5 e 7 ao longo do tempo



Autoria própria

A relação entre medição bruta com o resultado estimado na Figura 6, destaca-se como a curva estimada (linha azul) se encontra em sobreposição aos dados medidos na prática (linha vermelha).

Figura 6 – Tensão estimada X Medido na barra 5



Autoria própria

4.2.2 Caso 2: Falta trifásica para a terra

No presente estudo, foi inserida uma perturbação no sistema elétrico simulado, com o intuito de avaliar a resposta dinâmica do estimador de estados diante de distúrbios de curta duração. A perturbação modelada corresponde a uma falta trifásica para a terra (tipo 3LG – Three-Line-to-Ground), aplicada diretamente na Barra 5.

A falha foi iniciada no instante $t = 2,0$ segundos e perdurou por 100 milissegundos, sendo eliminada no tempo $t = 2,1$ segundos. A impedância de falta considerada foi puramente resistiva $Z_f = 0,001 + j0,0$ p.u. A curta duração da falta caracteriza-a como um evento transitório, com impacto significativo sobre o perfil de tensões do sistema no intervalo em questão. A Figura 7 ilustra o comportamento do sistema no instante de 2,0 segundos, momento em que ocorre o curto-circuito trifásico para terra na barra 5. Observa-se uma perturbação significativa nas tensões dessas barras, evidenciando o impacto imediato da falta no sistema elétrico. Essa perturbação, embora de curta duração, provoca oscilações que se estendem por um intervalo de tempo após a remoção da falha.

A Figura 8 apresenta os ângulos de tensão correspondentes, nos quais a distorção é ainda mais evidente ao longo do tempo, indicando que o sistema requer um período maior para retornar ao seu estado de equilíbrio. Tal comportamento destaca a importância de

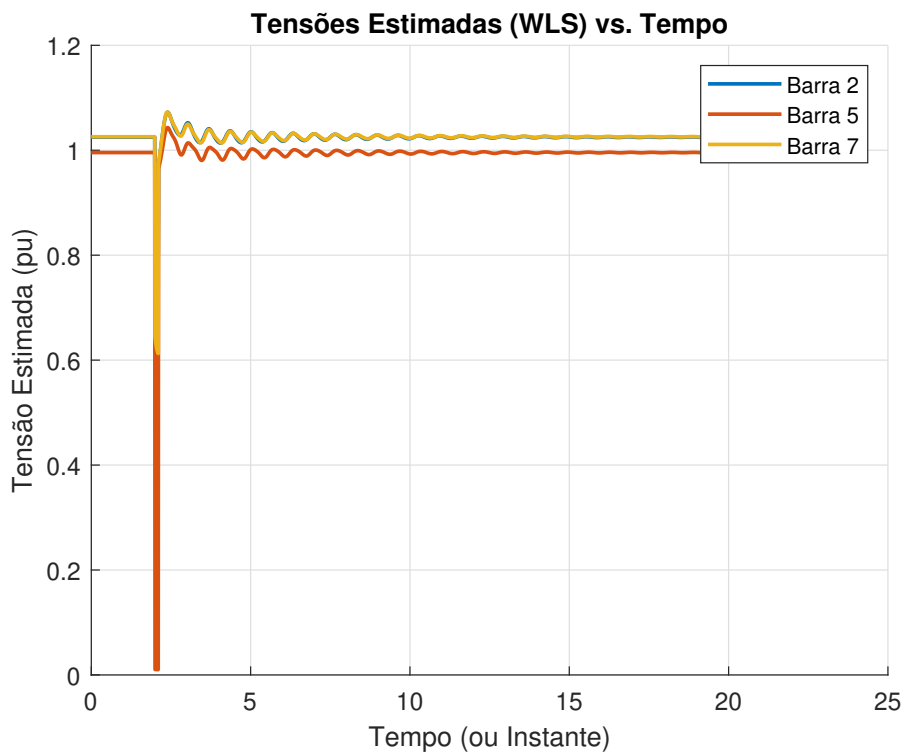
mecanismos de controle e monitoramento capazes de capturar e responder eficientemente a eventos transitórios dessa natureza.

Observa-se na barra 5, local onde ocorreu a falta, o comportamento pré-falta ($t < 2,0s$) não apresenta oscilações com tensões abaixo de 1,0 p.u., já durante a falta ($t = 2,0s$ até $t = 2,1s$), uma queda significativa da tensão, o efeito mais evidente e seguindo o esperado de ser próximo de zero, atingindo cerca de 0,01 p.u. por aplicação direta do curto-circuito de baixa impedância simulado no barramento. O comportamento angular apresenta o afundamento, alcançando valores de -78,5 graus.

Assim que a falta é eliminada, ou seja, em $t = 2,1$ segundos, a tensão teve o comportamento de grande elevação, iniciando em um valor próximo a zero para cerca de 0,8 p.u. Seu comportamento se caracteriza com um *overshoot* transitório antes de estabilizar em seu valor semelhante ao valor anterior à ocorrência de falta.

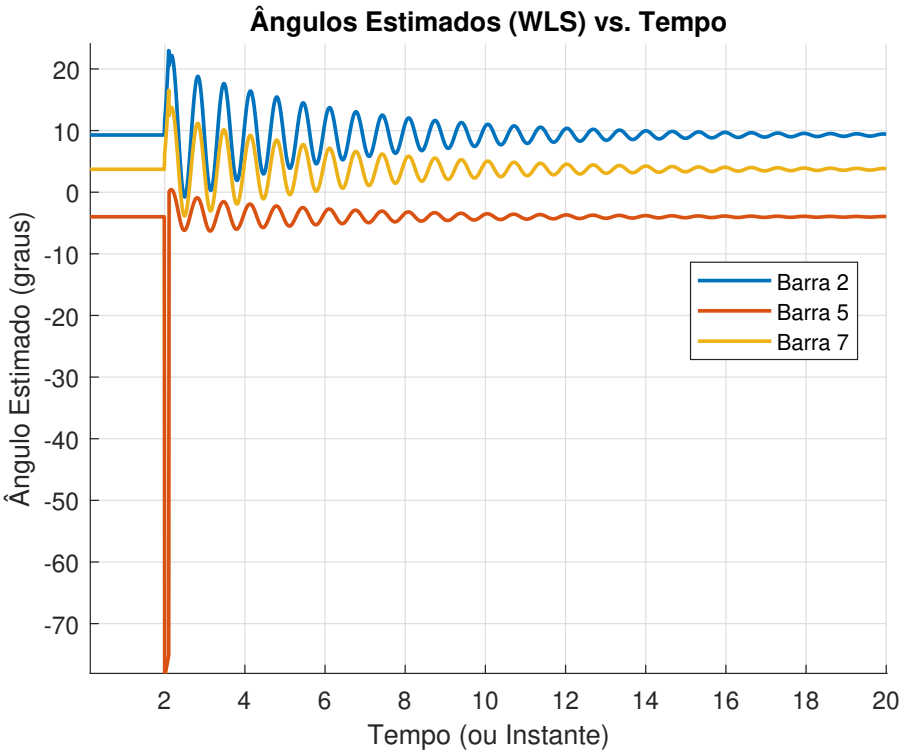
Similarmente ao ocorrido na tensão, se recupera e volta a um valor estável de comportamento análogo ao mencionado na tensão. O estimador WLS acompanhou adequadamente tanto no afundamento de tensão quanto na sua recuperação, fornecendo uma trajetória de estado filtrada mesmo sob condições de distúrbio severo, como evidenciado pela comparação com as medições nas figuras 9 e 10.

Figura 7 – Tensão estimada nas barra 2, 5 e 7 durante um curto



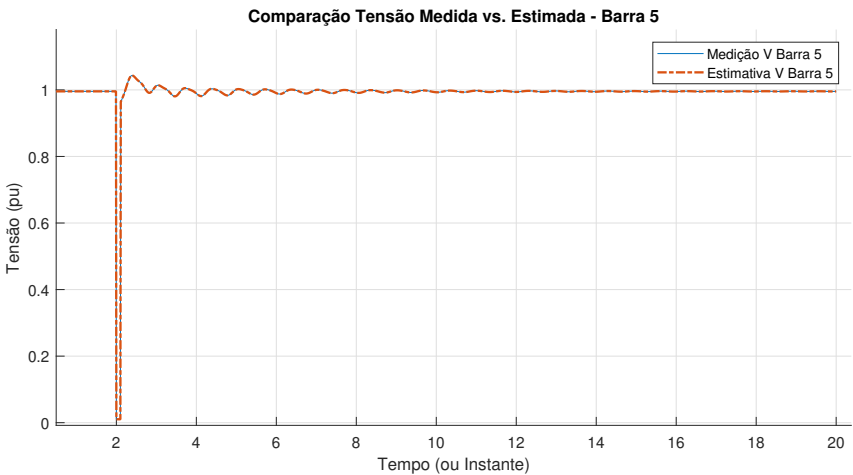
Autoria própria

Figura 8 – Tensão estimada na barra 2, 5 e 7 durante um curto



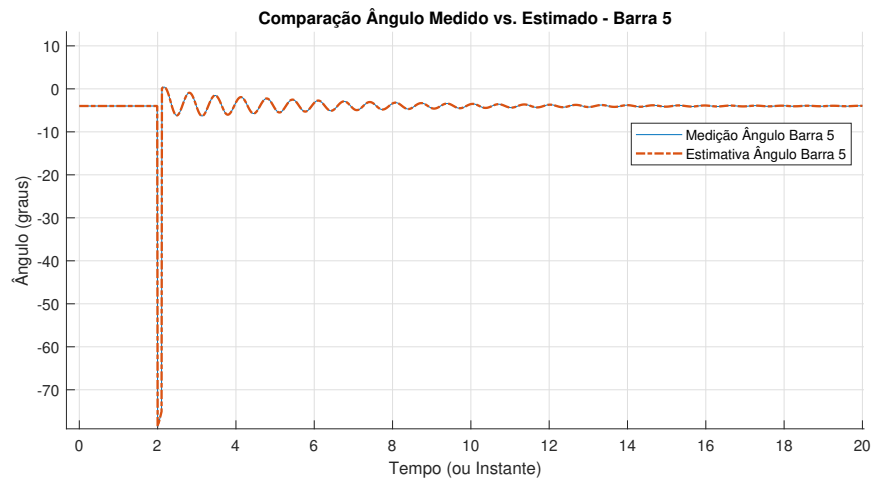
Autoria própria

Figura 9 – Tensão estimada sobre a barra 5 durante um curto



Autoria própria

Figura 10 – Ângulo estimado sobre a barra 5 durante um curto



Autoria própria

Esse tipo de falta (3LG) é uma das mais severas em termos de impacto elétrico, pois afeta simultaneamente as três fases e envolve ligação direta com o potencial de terra, podendo comprometer a estabilidade do sistema se não for rapidamente isolada. A modelagem desse cenário permitiu verificar o desempenho do algoritmo de estimação de estado baseado no método dos Mínimos Quadrados Ponderados (WLS), sobretudo sua capacidade de seguir a dinâmica das variáveis elétricas durante e após a ocorrência de um distúrbio severo.

5 Conclusão

Este Trabalho de Conclusão de Curso dedicou-se ao estudo e implementação do método dos Mínimos Quadrados Ponderados (WLS) para a estimação de estado em sistemas elétricos de potência, uma ferramenta essencial para a monitoramento e operação confiável da rede. O objetivo principal foi desenvolver e validar uma ferramenta computacional em MATLAB capaz de realizar a estimação de estado (magnitudes e ângulos de tensão) de um SEP, utilizando dados fasoriais variantes no tempo. Para tal, foi realizada a geração de dados sintéticos via simulação do sistema IEEE de 9 barras, seguida pela implementação do cálculo da matriz de admitância e do algoritmo iterativo WLS, culminando na análise e validação dos resultados obtidos.

Os resultados apresentados demonstraram o funcionamento eficaz do estimador de estado WLS implementado. A análise dos perfis de tensão e ângulo estimados ao longo de 2000 instantes de tempo revelou um comportamento consistente e fisicamente coerente com as características do sistema simulado, como a manutenção de níveis de tensão adequados nas barras de geração e quedas de tensão mais acentuadas em barras com cargas conectadas. A comparação entre as medições de entrada e os estados estimados validou visualmente a capacidade do WLS em produzir estimativas.

A principal contribuição deste trabalho reside na implementação didática e funcional de um estimador de estado WLS sequencial em MATLAB, utilizando medições fasoriais de tensão e ângulo. Esta implementação serve como uma ferramenta prática para o aprendizado e a demonstração dos conceitos de estimação de estado, permitindo a visualização do comportamento temporal do sistema e o efeito do algoritmo WLS. Adicionalmente, a análise realizada no sistema de 9 barras reforça a aplicabilidade e a robustez do método WLS, mesmo em sua formulação básica, para obter uma visão confiável do estado da rede a partir de medições imperfeitas. A simulação dessa contingência reforça a importância do monitoramento em tempo real via medições fasoriais (PMUs), pois apenas com alta resolução temporal e precisão na aquisição dos dados é possível capturar com fidelidade os efeitos transitórios e validar algoritmos que visam garantir a operação segura e confiável da rede elétrica.

Reconhece-se, contudo, algumas limitações neste estudo. A utilização de dados sintéticos, embora controlada e útil para validação, não captura toda a complexidade de erros presentes em medições reais de PMUs (como dados ausentes, erros não gaussianos ou falhas de sincronismo). A análise foi restrita a um sistema de pequeno porte (9 barras). Além disso, a implementação atual não incorpora técnicas avançadas para detecção e tratamento de dados espúrios (bad data detection/rejection), um componente importante

em estimadores industriais.

Como sugestões para trabalhos futuros, recomenda-se a aplicação e teste do estimador desenvolvido em sistemas de maior porte e complexidade (como os sistemas IEEE de 30, 57 ou 118 barras). Seria enriquecedor incorporar outros tipos de medição no algoritmo (fluxos de potência ativa/reativa, injeções de potência), o que demandaria o tratamento não-linear da função de medição e da matriz Jacobiana. A implementação de módulos para análise de resíduos e detecção/eliminação de dados espúrios aumentaria significativamente a robustez do estimador. Finalmente, a aplicação da ferramenta a dados reais provenientes de PMUs representaria um passo importante para validar seu desempenho em condições operacionais práticas.

Referências Bibliográficas

ABUR, A.; EXPÓSITO, A. G. *Power System State Estimation: Theory and Implementation*. New York: Marcel Dekker, 2004. ISBN 9780824755706. Disponível em: <<https://doi.org/10.1201/9780203913673>>. Citado 3 vezes nas páginas 13, 14 e 16.

ANDRADE SÔNIA, S. R. C. A. Sistemas de medição fasorial sincronizada: aplicações para melhoria da operação de sistemas elétricos de potência. *Repositório Institucional da Universidade Federal de Minas Gerais (UFMG)*, 2008. Notas adicionais (se houver). Disponível em: <<http://hdl.handle.net/1843/BUOS-8D3FHK>>. Citado na página 19.

ARRILLAGA, J.; WATSON, N. Load flow. In: _____. *Computer Modelling of Electrical Power Systems*. John Wiley & Sons, Ltd, 2001. cap. 4, p. 81–128. ISBN 9781118878286. Disponível em: <<https://doi.org/10.1002/9781118878286.ch4>>. Citado na página 21.

BASSANEZI, R. C. *Modelagem Matemática: Teoria e Prática*. São Paulo: Editora Contexto, 2002. ISBN 9788572441800. Citado na página 9.

EHRENSPERGER, J. Sistemas de medição fasorial sincronizada: Análise do estado da arte e aplicações no monitoramento de sistemas de energia elétrica. *Repositório Institucional da Universidade Federal de Santa Catarina (UFSC)*, 2004. Notas adicionais (se houver). Disponível em: <<http://repositorio.ufsc.br/xmlui/handle/123456789/87164>>. Citado 2 vezes nas páginas 10 e 19.

MONTICELLI, A. State estimation in electric power systems: A generalized approach. *Power Systems Research*, v. 48, 1999. Disponível em: <<https://doi.org/10.1007/978-1-4615-4999-4>>. Citado na página 13.

NAGLIC, M. *PMU Measurements for IEEE 39-Bus Power System Model*. 2022. Acesso em: 11 dez. 2024. Disponível em: <<https://ieee-dataport.org/open-access/pmu-measurements-ieee-39-bus-power-system-model>>. Citado na página 10.

OLIVEIRA, T. L. Desenvolvimento de um programa computacional livre, gráfico, e multiplataforma para analisar sistemas elétricos de potência em regime permanente e dinâmico. *Repositório Institucional da Universidade Federal de Uberlândia (UFU)*, 2019. Disponível em: <<http://dx.doi.org/10.14393/ufu.te.2019.2444>>. Citado na página 22.

VIDE, P. S. P. da Silva e C. *Estimação de Estado de um Sistema Elétrico de Energia, Usando Medidas Fasoriais Sincronizadas*. Tese (Tese de Doutorado) — Universidade de Engenharia do Porto, 2013. Acesso em: 11 dez. 2024. Disponível em: <https://sigarra.up.pt/reitoria/pt/pub_geral.show_file?pi_doc_id=10566>. Citado na página 10.

ANEXO A – Script Matlab

```

1  clc;
2  clear;
3  close all;
4
5  % --- extração dados ---
6  filename = 'databus_psp.csv';
7  dataMatrix = readmatrix(filename);
8
9  % --- Organizacao dados ---
10
11 [numInstantes, numColunas] = size(dataMatrix);
12 numBarras = (numColunas - 1) / 2; % coluna 1 refere ao tempo
13 fprintf('Número de Barras detectado: %d\n', numBarras);
14 fprintf('Número de Instantes de Medição detectado: %d\n', numInstantes);
15
16 V_medicoes = zeros(numInstantes, numBarras);
17 A_medicoes_deg = zeros(numInstantes, numBarras);
18 for j = 1:numBarras
19     V_medicoes(:, j) = dataMatrix(:, 2*j);
20     A_medicoes_deg(:, j) = dataMatrix(:, 2*j+1);
21 end
22
23 disp('Primeiras 5 linhas de Tensões Medidas (V_med):');
24 disp(V_medicoes(1:5,:));
25 disp('Primeiras 5 linhas de Ângulos Medidos em Graus (A_med_deg):');
26 disp(A_medicoes_deg(1:5,:));
27
28
29 S_base = 100; % 100MVA
30 % [Barra, P_load (pu), Q_load (pu)] - Cargas nas barras 5, 6, 8
31 dados_cargas = [ 5, 1.25, 0.5;
32                 8, 1, 0.35;
33                 6, 0.9, 0.3; ];
34
35 % [Barra, P_gen (pu), Q_gen (pu)] - Geração nas barras 1, 2, 3
36 dados_geradores = [ 1, 0.716, 0.27;
37                    2, 1.63, 0.06;

```

```

38         3, 0.85, -0.01; ];
39
40 % [De, Para, R (pu), X (pu), B (pu), Tap]
41 dados_linhas = [ 7, 8, 0.0085, 0.072, 0.149, 1;
42                 8, 9, 0.0119, 0.1008, 0.209, 1;
43                 7, 5, 0.032, 0.161, 0.306, 1;
44                 5, 4, 0.01, 0.085, 0.176, 1;
45                 4, 6, 0.017, 0.092, 0.158, 1;
46                 6, 9, 0.039, 0.17, 0.358, 1; ];
47
48 % [De, Para, R (pu), X (pu), B (pu), Tap]
49 dados_transformadores = [ 2, 7, 0.0, 0.0625, 0, 18/230;
50                           1, 4, 0.0, 0.0576, 0, 16.5/230;
51                           3, 9, 0.0, 0.0586, 0, 13.8/230; ];
52
53 dados_ramos = [dados_linhas; dados_transformadores];
54 nb = 9;
55
56 % --- Criar Ybus (matriz admittancias) ---
57 Ybus = zeros(nb, nb);
58
59 for k = 1:size(dados_ramos,1)
60     de = dados_ramos(k,1); para = dados_ramos(k,2); r = dados_ramos(k,3); x = dados_ramos(k,4);
61     z_serie = r + 1j*x; y_serie = 1/z_serie; y_sh = 1j*b_sh/2; %verrificar Susceptância (
62     Ybus(de,de) = Ybus(de,de) + y_serie/(tap^2) + y_sh;
63     Ybus(para,para) = Ybus(para,para) + y_serie + y_sh;
64     Ybus(de,para) = Ybus(de,para) - y_serie/tap;
65     Ybus(para,de) = Ybus(para,de) - y_serie/tap;
66 end
67
68 %G = real(Ybus); B = imag(Ybus);
69 disp('Matriz Ybus Calculada:'); disp(Ybus);
70
71 % --- Implementação WLS ---
72
73 instante_escolhido = 1;
74 V_med_instante = V_medicoes(instante_escolhido, :)' ;
75 A_med_rad_instante = deg2rad(A_medicoes_deg(instante_escolhido, :))';
76 z = [V_med_instante; A_med_rad_instante]; % Vetor z (Medidas)
77
78 disp('Vetor de Medidas (z):'); disp(z);
79

```

```

80 slack_bus = 1; % barra de referencia
81 angulos_idx_estado = setdiff(1:nb, slack_bus); % Índices dos ângulos em x sem slack
82 tensoes_idx_estado = 1:nb; % Índices das tensões em x inclui slack
83
84
85 theta_inicial = zeros(nb, 1);
86 V_inicial = ones(nb, 1);
87
88 % --- Montar vetor de estado inicial x0 ---
89 x0 = [theta_inicial(angulos_idx_estado); V_inicial(tensoes_idx_estado)];
90 disp('Vetor de Estado Inicial (x0:'); disp(x0);
91
92 % --- matriz Pesos (W) ---
93
94 sigma_V = 0.1; sigma_A = 0.01; % peso relativo
95 variancias = [repmat(sigma_V^2, nb, 1); repmat(sigma_A^2, nb, 1)];
96 W = diag(1 ./ variancias); % Matriz de Pesos W
97 disp('Dimensões da Matriz de Pesos (W:'); disp(size(W));
98
99 % --- Implementar a Função h(x) ---
100 % Esta função calcula o vetor de medições esperadas 'h' com base no vetor de estado atual
101
102 calculate_h = @(x, nb, slack_bus, angulos_idx_estado) ...
103     calculate_h_implementation(x, nb, slack_bus, angulos_idx_estado);
104
105 % Exemplo de chamada da função com o estado inicial (para teste)
106 h0 = calculate_h(x0, nb, slack_bus, angulos_idx_estado);
107 disp('Vetor h(x) calculado para o estado inicial x0 (h0:');
108 disp(h0);
109 disp('Dimensões de h0:');
110 disp(size(h0)); % supostamente mesmas dimensões que z
111
112
113
114 % --- Calcular a Matriz Jacobiana H = dh/dx ---
115
116 num_medidas = length(z); % 2*nb
117 num_estados = length(x0); % 2*nb - 1
118
119 % Inicializar a matriz H com zeros
120 H = zeros(num_medidas, num_estados);
121

```

```

122 % Índices correspondentes às variáveis no vetor de estado x
123 % x = [A2; ...; Anb; V1; ...; Vnb]
124 idx_angulos_em_x = 1:(nb-1);
125 idx_tensoes_em_x = (nb-1)+1 : num_estados; % Equivalente a nb : num_estados
126
127 % Índices correspondentes às medições no vetor h
128 % h = [V1; ...; Vnb; A1; ...; Anb]
129 idx_V_em_h = 1:nb;
130 idx_A_em_h = nb+1 : num_medidas; % Equivalente a nb+1 : 2*nb
131
132 % Bloco Superior Direito: dV / dV (Identidade nb x nb)
133 % Derivada das medições de tensão (linhas 1:nb de h) em relação
134 % aos estados de tensão (colunas nb:fim de x)
135 H(idx_V_em_h, idx_tensoes_em_x) = eye(nb);
136
137 % Bloco Inferior Esquerdo: dA / dA (Quase Identidade nb x (nb-1))
138 % Derivada das medições de ângulo (linhas nb+1:fim de h) em relação
139 % aos estados de ângulo (colunas 1:nb-1 de x)
140
141 % Para cada estado de ângulo Ak (k=2..nb), a derivada da medição Ak é 1.
142 % A linha da medição de A1 (slack) tem derivadas zero em relação a A2..Anb.
143 linhas_A_nao_slack_em_h = idx_A_em_h(angulos_idx_estado); % Índices das linhas de A2..Anb
144
145 % Construindo a submatriz identidade (nb-1)x(nb-1)
146 identidade_angulos = eye(nb-1);
147
148 % Colocando a identidade nas posições corretas de H
149 % Linhas: correspondentes às medições A2..Anb em h
150 % Colunas: correspondentes aos estados A2..Anb em x
151 H(linhas_A_nao_slack_em_h, idx_angulos_em_x) = identidade_angulos;
152
153 % Os outros blocos (dV/dA e dA/dV) já estão preenchidos com zeros
154 % devido à inicialização H = zeros(...);
155
156 disp('Matriz Jacobiana H Calculada:');
157 disp(H);
158 disp('Dimensões de H:');
159 disp(size(H)); % Deve ser [18, 17] para nb=9
160
161
162
163 % --- Loop WLS para Múltiplos Instantes de temps---

```

```

164
165 fprintf('\n--- Iniciando Estimação de Estado WLS para %d Instantes ---\n', numInstantes);
166
167 % Pré-alocar matrizes para armazenar os resultados históricos
168 % Usar NaN como placeholder inicial é útil para identificar instantes sem resultado
169 V_estimado_historico = NaN(numInstantes, nb);
170 A_estimado_historico_deg = NaN(numInstantes, nb);
171 convergencia_historico = false(numInstantes, 1); % Guarda se convergiu ou não
172
173 % Parâmetros de Convergência
174 tol = 1e-5; % Erro aceitável
175 max_iter = 10; % Evitar ciclo sem fim
176
177
178 % x_anterior_final começa para ser usado no primeiro instante
179 x_anterior_final = x0; % x0 é definido anteriormente
180
181 % Loop através de cada instante de tempo medido
182 for t = 1:numInstantes
183
184     fprintf('Processando Instante t = %d / %d...\n', t, numInstantes);
185
186     % 1. Montar o vetor de medições z para o instante ATUAL 't'
187     V_med_instante_t = V_medicoes(t, :)' ;
188     A_med_rad_instante_t = deg2rad(A_medicoes_deg(t, :))';
189     z_t = [V_med_instante_t; A_med_rad_instante_t];
190
191     % 2. Definir a estimativa inicial para esta iteração 't'
192     % Usar o resultado do instante anterior
193     x_inicial_t = x_anterior_final; % Usa o resultado de t-1 como chute para t
194
195     % --- Início do Loop WLS Interno (para o instante t) ---
196     x_atual = x_inicial_t;
197     iter = 0;
198     convergido_t = false;
199
200     while ~convergado_t && iter < max_iter
201         iter = iter + 1;
202
203         % Calcular h(x)
204         h_calculado = calculate_h(x_atual, nb, slack_bus, angulos_idx_estado);
205         % Calcular resíduo

```

```

206     delta_z = z_t - h_calculado;
207     % Jacobiano H é constante
208     % Calcular Ganho G
209     G = H' * W * H;
210     % Calcular RHS
211     rhs = H' * W * delta_z;
212     % Resolver para delta_x
213     delta_x = G \ rhs;
214     % Atualizar estado
215     x_novo = x_atual + delta_x;
216     % Verificar convergência
217     norma_dx = norm(delta_x);
218     if norma_dx < tol
219         convergido_t = true;
220     end
221     % Atualizar para próxima iteração interna
222     x_atual = x_novo;
223 end % Fim do loop WLS interno
224
225 % --- Fim do Loop WLS Interno ---
226
227 % 3. Armazenar Resultados (se convergiu)
228 if convergido_t
229     fprintf(' Instante t=%d convergiu em %d iterações.\n', t, iter);
230     convergencia_historico(t) = true;
231     x_final_t = x_atual;
232
233     % Extrair V e A finais
234     num_angulos_no_estado = nb - 1;
235     A_final_parcial_rad = x_final_t(1:num_angulos_no_estado);
236     V_final = x_final_t(num_angulos_no_estado + 1 : end);
237     A_final_completo_rad = zeros(nb, 1);
238     A_final_completo_rad(angulos_idx_estado) = A_final_parcial_rad;
239     A_final_completo_deg = rad2deg(A_final_completo_rad);
240
241     % Guardar nos históricos
242     V_estimado_historico(t, :) = V_final'; % Guarda como linha
243     A_estimado_historico_deg(t, :) = A_final_completo_deg'; % Guarda como linha
244
245     % Preparar para o próximo instante
246     x_anterior_final = x_final_t; % Usa este resultado como chute inicial para t+1
247

```

```
248     else
249         fprintf('  ATENÇÃO: Instante t=%d NÃO convergiu após %d iterações (Norma dx = %e)
250
251         % Poderia optar por resetar para ? x_anterior_final = x0;
252     end
253
254 end % Fim do for para cada instante de tempo
255
256 fprintf('--- Fim da Estimação de Estado para todos os instantes de tempos ---\n');
257
258
259 % --- Plotar Resultados ---
260
261 % Verificar se a primeira coluna de dataMatrix contém o tempo!!
262 % Assumindo que a coluna 1 é o tempo ou usar 1:numInstantes
263 if size(dataMatrix, 2) == (2*nb + 1)
264     tempo = dataMatrix(:, 1);
265     disp('Usando a primeira coluna do arquivo CSV como vetor de tempo.');
```

266 else

267 tempo = 1:numInstantes;

268 disp('Assumindo vetor de tempo como 1:numInstantes.');

269 end

270

271 % Escolher algumas barras para plotar (ex: barras 1, 5, 9)

272 barras_para_plotar = [1, 5, 9];

273

274 % Gráfico 1: Tensões Estimadas vs. Tempo

275 figure; % Cria uma nova janela de figura

276 hold on; % Permite plotar múltiplas linhas no mesmo gráfico

277 legend_entries_V = {}; % Inicializa legendas

278 for i = 1:length(barras_para_plotar)

279 barra_idx = barras_para_plotar(i);

280 if barra_idx >= 1 && barra_idx <= nb

281 plot(tempo, V_estimado_historico(:, barra_idx), 'LineWidth', 1.5);

282 legend_entries_V{end+1} = sprintf('Barra %d', barra_idx);

283 end

284 end

285 hold off;

286 title('Tensões Estimadas (WLS) vs. Tempo');

287 xlabel('Tempo (ou Instante)');

288 ylabel('Tensão Estimada (pu)');

289 legend(legend_entries_V);

```

290 grid on;
291
292 % Gráfico 2: Ângulos Estimados vs. Tempo
293 figure; % Cria uma nova janela de figura
294 hold on; % Permite plotar múltiplas linhas no mesmo gráfico
295 legend_entries_A = {}; % Inicializa legendas
296 for i = 1:length(barras_para_plotar)
297     barra_idx = barras_para_plotar(i);
298     if barra_idx >= 1 && barra_idx <= nb
299         plot(tempo, A_estimado_historico_deg(:, barra_idx), 'LineWidth', 1.5);
300         legend_entries_A{end+1} = sprintf('Barra %d', barra_idx);
301     end
302 end
303 hold off;
304 title('Ângulos Estimados (WLS) vs. Tempo');
305 xlabel('Tempo (ou Instante)');
306 ylabel('Ângulo Estimado (graus)');
307 legend(legend_entries_A);
308 grid on;
309
310 % jsut 4 fun: Comparar com medições originais em alguns gráficos
311 barra_comp = 5; % Exemplo: comparar barras cargas
312 if barra_comp >= 1 && barra_comp <= nb
313     figure;
314     hold on;
315     plot(tempo, V_medicoes(:, barra_comp), '--', 'DisplayName', sprintf('Medição V Barra %d', barra_comp));
316     plot(tempo, V_estimado_historico(:, barra_comp), '-', 'LineWidth', 1.5, 'DisplayName', sprintf('Estimado V Barra %d', barra_comp));
317     hold off;
318     title(sprintf('Comparação Tensão Medida vs. Estimada - Barra %d', barra_comp));
319     xlabel('Tempo (ou Instante)');
320     ylabel('Tensão (pu)');
321     legend;
322     grid on;
323
324     figure;
325     hold on;
326     % Linha da medição (tracejada azul com marcador leve)
327     plot(tempo, A_medicoes_deg(:, barra_comp), '--o', ...
328         'DisplayName', sprintf('Medição Ângulo Barra %d', barra_comp), ...
329         'LineWidth', 1.2, ...
330         'MarkerSize', 3, ...
331         'Color', [0 0.4470 0.7410]);

```

```
332
333 % Linha da estimativa (linha contínua laranja)
334 plot(tempo, A_estimado_historico_deg(:, barra_comp), '-', ...
335       'DisplayName', sprintf('Estimativa Ângulo Barra %d', barra_comp), ...
336       'LineWidth', 2, ...
337       'Color', [0.8500 0.3250 0.0980]);
338
339 title(sprintf('Comparação Ângulo Medido vs. Estimado - Barra %d', barra_comp));
340 xlabel('Tempo (ou Instante)');
341 ylabel('Ângulo (graus)');
342 legend;
343 grid on;
344 end
```

ANEXO B – Script Matlab função

```

1
2 % --- Definição da Implementação da Função h(x) ---
3
4 function h = calculate_h_implementation(x, nb, slack_bus, angulos_idx_estado)
5     % x: Vetor de estado atual [A2; ...; Anb; V1; ...; Vnb] (Ângulos em radianos)
6     % nb: Número de barras
7     % slack_bus: Índice da barra de referência (onde A=0)
8     % angulos_idx_estado: Índices das barras cujos ângulos estão em x (ex: [2, 3, ..., nb])
9
10    num_angulos_no_estado = nb - 1;
11    num_tensoes_no_estado = nb; % Todas as tensões estão no estado
12
13    % Validar tamanho de x (opcional, mas bom para debug)
14    tamanho_esperado_x = num_angulos_no_estado + num_tensoes_no_estado;
15    if length(x) ~= tamanho_esperado_x
16        error('Tamanho incorreto do vetor de estado x recebido pela função h(x). Esperado:
17    end
18
19    % 1. Extrair ângulos e tensões do vetor de estado x
20    A_estimado_parcial = x(1:num_angulos_no_estado); % Ângulos estimados (sem o slack)
21    V_estimado         = x(num_angulos_no_estado + 1 : end); % Tensões estimadas (todas)
22
23    % 2. Reconstruir o vetor completo de ângulos estimados (incluindo slack = 0)
24    A_estimado_completo = zeros(nb, 1);
25    A_estimado_completo(angulos_idx_estado) = A_estimado_parcial; % Preenche os ângulos d
26    % A_estimado_completo(slack_bus) já contém 0, que é o ângulo da referência
27
28    % 3. Montar o vetor h(x) na mesma ordem que o vetor de medições z
29    % Ordem de z: [V1_med; ...; Vnb_med; A1_med_rad; ...; Anb_med_rad]
30    h = [V_estimado; A_estimado_completo]; % Concatenar V e A estimados
31
32 end
33

```
