UNIVERSIDADE FEDERAL DE JUIZ DE FORA
FACULDADE DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Ricardo Rodrigues Rapozo

Estimação de Fasores Harmônicos Baseada em Banco de Filtros Polifásicos: Implementação em FPGA com Processador Embarcado

Juiz de Fora

2026

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF
com os dados fornecidos pelo(a) autor(a)

Rapozo, Ricardo.
Estimação de Fasores Harmônicos Baseada em Banco de Filtros Polifásicos: Implementação em FPGA com Processador Embarcado / Ricardo Rodrigues Rapozo. – 2026.
95 p. : il.

Orientador: Leandro Rodrigues Manso Silva
Dissertação (Mestrado) – Universidade Federal de Juiz de Fora, Faculdade de Engenharia. Programa de Pós-Graduação em Engenharia Elétrica, 2026.

1. Fasores Harmônicos. 2. Banco de Filtros Polifásicos. 3. Processadores Embarcados. I. Silva, L. R, M., orient. II. Título.

Ricardo Rodrigues Rapozo

Estimação de Fasores Harmônicos Baseada em Banco de Filtros Polifásicos: Implementação em FPGA com Processador Embarcado

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: Sistemas Eletrônicos

Aprovada em 15 de setembro de 2026.

BANCA EXAMINADORA

Prof. Dr. Leandro Rodrigues Manso Silva - Orientador
Universidade Federal de Juiz de Fora

Prof. Dr. Henrique Luis Moreira Monteiro
Universidade Federal de Lavras

Prof. Dr. Luciano Manhães de Andrade Filho
Universidade Federal de Juiz de Fora

Juiz de Fora, 21/08/2026.

Documento assinado eletronicamente por **Leandro Rodrigues Manso Silva**, **Professor(a)**, em 15/09/2026, às 15:40, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Henrique Luis Moreira Monteiro**, **Usuário Externo**, em 15/09/2026, às 15:41, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Luciano Manhaes de Andrade Filho**, **Professor(a)**, em 15/09/2026, às 19:06, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no Portal do SEI-Ufjf (www2.ufjf.br/SEI) através do ícone Conferência de Documentos, informando o código verificador **3161986** e o código CRC **53B6DF7C**.

Dedico este trabalho ——-.

AGRADECIMENTOS

Agradeço primeiramente a Deus, por me conceder a oportunidade de viver exatamente o que sempre almejei para este período da minha vida. Este sonho foi cultivado por anos, e hoje, ao concluir este ciclo, sinto uma imensa gratidão por ter chegado até aqui.

À minha namorada, Angelina, meu muito obrigado. Seu apoio incondicional ao longo dos anos em que estamos juntos foi fundamental, especialmente nos momentos mais difíceis da caminhada. Sua presença fez toda a diferença.

À minha família e amigos, sou grato pela paciência e compreensão. Sei que minha ausência em diversos momentos foi necessária para que este sonho se realizasse, e agradeço por terem entendido e torcido por mim.

Ao meu orientador, Professor Leandro Manso, expresso minha profunda gratidão pela orientação impecável e, acima de tudo, pela compreensão das minhas limitações de tempo, impostas pela minha jornada de trabalho. Sua confiança e direcionamento foram essenciais.

Aos meus colegas Lucca e Rudson, agradeço por terem entendido minhas ideias e por me oferecerem o suporte necessário para que eu pudesse concluir o mestrado.

Ao Laboratório NIPS, meu agradecimento especial por todo o auxílio e pelo ambiente acolhedor. Em especial, não posso deixar de mencionar o Vinicius, cuja ajuda foi tão decisiva que, sem ele, esta trajetória teria sido muito mais árdua, senão impossível.

Por fim, mas não menos importante, agradeço à minha cachorrinha, Mel, que com suas brincadeiras e alegria me proporcionou momentos leves e felizes, essenciais para recarregar as energias.

RESUMO

A presença cada vez maior de cargas não lineares e de fontes conectadas por eletrônica de potência faz com que as tensões e correntes da rede elétrica se afastem da forma senoidal ideal, o que torna importante estimar não só a componente fundamental, mas também as componentes harmônicas do sinal. Essa estimação fica mais difícil quando a frequência fundamental se desvia do valor nominal, já que o erro de fase cresce com a ordem harmônica. Este trabalho apresenta e implementa uma metodologia de estimação de fasores harmônicos para sinais com frequência variável, formada por três etapas: estimação da frequência por cruzamento por zero, com suavização por filtro de Kalman; reamostragem dinâmica do sinal por interpolação B-Spline em estrutura de Farrow; e extração das harmônicas por um banco de filtros Flat-Top com decomposição polifásica, seguido de uma IDFT. Com a reamostragem, o banco de filtros trabalha sempre com o número nominal de pontos por ciclo, sem precisar ter seus coeficientes recalculados. A metodologia foi desenvolvida primeiro como um modelo de referência em Python e, depois, implementada em dois processadores *soft-core* da plataforma SAPHO, um dedicado ao interpolador e outro ao banco de filtros, que se comunicam por memórias FIFO e são coordenados por uma máquina de estados. Durante a implementação, foi preciso ajustar algumas etapas por conta da aritmética de 32 bits do processador, como o filtro de Kalman e o cálculo do período no cruzamento por zero. A implementação foi validada por simulação funcional e comparada com o modelo de referência em ensaios inspirados na norma IEC/IEEE 60255-118-1, com frequência fora da nominal, rampa de frequência e modulação. A frequência foi estimada com boa precisão em todos os ensaios. Na condição fora da nominal, o modelo de referência ficou abaixo do limite de 1% de TVE em toda a faixa avaliada, e a implementação embarcada só ultrapassou esse limite nas ordens harmônicas mais altas. Os resultados mostraram que essa diferença vem principalmente do desalinhamento temporal entre o sinal e a frequência usada na correção de fase, e não da precisão numérica do banco. Já no ensaio de rampa, o próprio modelo ultrapassa o limite nas ordens mais altas, o que indica uma limitação da metodologia sob variação contínua de frequência. De forma geral, os resultados mostram que a metodologia é viável em plataforma embarcada e apontam onde ela ainda pode ser melhorada.

Palavras-chave: Estimação fasorial harmônica. Interpolação B-Spline. TVE-Erro Vetorial Total. FFT-Fast Fourier Transform.

ABSTRACT

The growing number of nonlinear loads and power electronics based sources makes the voltages and currents in the power grid move away from the ideal sinusoidal waveform, so it becomes important to estimate not only the fundamental component but also the harmonic components of the signal. This task becomes harder when the fundamental frequency deviates from its nominal value, since the phase error grows with the harmonic order. This work presents and implements a harmonic phasor estimation method for signals with varying frequency, made up of three stages: frequency estimation by zero crossing, smoothed by a Kalman filter; dynamic resampling of the signal by B-Spline interpolation in a Farrow structure; and harmonic extraction by a Flat-Top filter bank with polyphase decomposition, followed by an IDFT. Thanks to the resampling, the filter bank always works with the nominal number of samples per cycle and its coefficients do not need to be recomputed. The method was first developed as a reference model in Python and then implemented on two soft-core processors of the SAPHO platform, one for the interpolator and one for the filter bank, which communicate through FIFO memories and are coordinated by a state machine. During the implementation, some stages had to be adjusted because of the 32-bit arithmetic of the processor, such as the Kalman filter and the period calculation in the zero crossing estimator. The implementation was validated by functional simulation and compared with the reference model in tests based on the IEC/IEEE 60255-118-1 standard, with off-nominal frequency, frequency ramp and modulation. The frequency was estimated accurately in all tests. In the off-nominal case, the reference model stayed below the 1% TVE limit over the whole range evaluated, and the embedded implementation only exceeded this limit at the highest harmonic orders. The results showed that this difference comes mainly from the time misalignment between the signal and the frequency used in the phase correction, and not from the numerical precision of the filter bank. In the ramp test, on the other hand, the reference model itself exceeds the limit at the highest orders, which points to a limitation of the method under continuous frequency variation. Overall, the results show that the method is feasible on an embedded platform and indicate where it can still be improved.

Keywords: Harmonic phasor estimation. B-Spline interpolation. Total Vector Error (TVE). Fast Fourier Transform (FFT).

LISTA DE FIGURAS

Figura 1 – Representação de uma senoide e de seu fasor associado no plano complexo, adaptada de (PHADKE; THORP, 2008). 20

Figura 2 – Representação de medições fasoriais sincronizadas em diferentes barras de um sistema elétrico: cada ponto de medição possui um fasor de tensão próprio, com magnitude e ângulo distintos, todos referidos a uma base temporal comum. 22

Figura 3 – Estrutura geral de um sistema de medição fasorial sincronizada, adaptada de (LUIZ, 2024). 23

Figura 4 – Resposta em magnitude do filtro protótipo Flat-Top: (a) resposta completa do protótipo modulado para a frequência fundamental; (b) desvio de magnitude em torno da frequência central, evidenciando a planicidade. 34

Figura 5 – Implementação polifásica de um banco de filtros uniforme: o ganho M na entrada precede a linha de atrasos, e cada ramo é decimado por M antes de alimentar o componente polifásico $E_\ell(z^M)$, cujas saídas são combinadas por uma IDFT de M pontos. 35

Figura 6 – Visão geral da cadeia de processamento completa do método proposto, do sinal de entrada às métricas de avaliação. 37

Figura 7 – Interpolação linear do instante de cruzamento por zero: a reta que liga as amostras adjacentes $x_f[n-1]$ e $x_f[n]$, nos instantes t_{n-1} e t_n, cruza o eixo no instante estimado $\hat{t}_{zc}$. 39

Figura 8 – Obtenção de uma amostra em instante fracionário pelo Bspline. 44

Figura 9 – Estrutura de Farrow para interpolação cúbica. 45

Figura 10 – Pipeline de implementação dividido em dois processadores *soft-core*, ligados por duas FIFOs — uma para o sinal reamostrado e outra para a frequência estimada — e um multiplexador na entrada do processador do banco. Como na Figura 6, o sinal e a frequência percorrem caminhos distintos a partir da entrada, cada um com a sua própria compensação de atraso: 255 amostras no sinal, antes do pré-filtro B-Spline, e 6 amostras na frequência, antes do interpolador Farrow (Tabela 2). No processador do banco os dois fluxos são decimados por M: a frequência encerra aí o seu percurso, emitida uma vez por quadro, enquanto o sinal segue para o banco polifásico e a IDFT, de onde saem os fasores $\hat{X}_h[k/M]$, um quadro a cada M amostras. Na seta entre o banco e a IDFT, $E_\ell(z)$ denota as M saídas das componentes polifásicas, que a transformada combina nos canais harmônicos. 51

Figura 11 – Resposta em magnitude do pré-filtro Bessel do Zero-Crossing, em cascata de três seções de segunda ordem, com a frequência de corte e a banda de interesse destacadas. 53
Figura 12 – Resposta de fase (acima) e atraso de grupo (abaixo) do pré-filtro Bessel do Zero-Crossing. Na banda de interesse a fase praticamente coincide com a reta de fase linear traçada com o atraso de grupo de 60 Hz, e o atraso permanece constante em torno de 126,5 amostras. 54
Figura 13 – Diagrama de polos e zeros do pré-filtro Bessel em SOS ($f_s = 15{,}36\,\text{kHz}$): (a) visão geral do plano z, com os seis zeros em $z = -1$; (b) detalhe dos três pares de polos, todos com $|p_i| < 1$. 55
Figura 14 – Origem das 128 amostras de latência do Zero-Crossing. Acima, a estimativa f_{zc} resulta de uma média sobre o meio ciclo entre dois cruzamentos, cuja referência temporal é o ponto médio do intervalo. Abaixo, esse valor é retido até o cruzamento seguinte, o que acrescenta uma idade média de mais um quarto de ciclo. 61
Figura 15 – Resposta de fase (acima) e atraso de grupo (abaixo) do filtro protótipo Flat-Top. Por ser um FIR simétrico, a fase coincide com a reta de fase linear em toda a faixa, e o atraso de grupo é constante e igual a $(N-1)/2 = 1024$ amostras. A banda sombreada é a largura de canal do banco, $f_0/2 = 30\,\text{Hz}$. 65
Figura 16 – Diagrama de estados da máquina de controle do processador do banco. A saída `rst_proc_banco` vale 1 apenas em `S2`. 69
Figura 17 – Erros de magnitude, de fase e TVE por ordem harmônica: média e pior caso entre as dez frequências *off-nominal* ensaiadas. 76
Figura 18 – Erro médio por ordem harmônica em cada uma das dez frequências *off-nominal*, com a faixa correspondente ao modelo Python ao fundo. 77
Figura 19 – Divergência relativa entre o SAPHO e o modelo Python em cada etapa da cadeia, ao longo da janela analisada, no ensaio *off-nominal*. . 78
Figura 20 – Erros de magnitude, de fase e TVE por ordem harmônica, para rampa de frequência de 55 Hz a 65 Hz. 79
Figura 21 – Divergência relativa entre o SAPHO e o modelo Python em cada etapa da cadeia, ao longo da janela analisada, no ensaio de rampa. . . 80
Figura 22 – Erro da correção de fase na 50ª ordem harmônica, em relação à correção ideal, ao longo do ensaio de rampa. 81
Figura 23 – Erros de magnitude, de fase e TVE por ordem harmônica, para modulação de fase com $f_m = 0{,}5\,\text{Hz}$. 82
Figura 24 – Erros de magnitude, de fase e TVE por ordem harmônica, para modulação de fase com $f_m = 5\,\text{Hz}$. 83

Figura 25 – Erros de magnitude, de fase e TVE por ordem harmônica, para modulação de amplitude com $f_m = 0{,}5\,\text{Hz}$. 84
Figura 26 – Erros de magnitude, de fase e TVE por ordem harmônica, para modulação de amplitude com $f_m = 5\,\text{Hz}$. 85
Figura 27 – Divergência relativa entre o SAPHO e o modelo Python em cada etapa da cadeia, nos ensaios de modulação de fase. 86
Figura 28 – Divergência relativa entre o SAPHO e o modelo Python em cada etapa da cadeia, nos ensaios de modulação de amplitude. 86

LISTA DE TABELAS

Tabela 1 – Parâmetros do sistema implementado. 50
Tabela 2 – Atrasos compensados na cadeia. 60
Tabela 3 – Amostras descartadas antes de habilitar cada etapa. 61
Tabela 4 – Comportamento da máquina de estados de controle do banco. . 68
Tabela 5 – Resolução absoluta (ULP) do formato de 32 bits do SAPHO para as grandezas de frequência acumuladas na cadeia. 70
Tabela 6 – Erro do estado do filtro de Kalman após 14 336 atualizações em 32 bits, tomando como referência o mesmo cálculo em dupla precisão. . 71
Tabela 7 – Erro no cálculo do intervalo de meio ciclo em 32 bits, para $n_{zc} = 128$ amostras (60 Hz), e erro eficaz resultante em f_{zc} na faixa de 55 Hz a 65 Hz. 71
Tabela 8 – Ensaios realizados, equação do sinal de teste e respectivos parâmetros. 75

LISTA DE ABREVIAÇÕES E ACRÔNIMOS

DFT Transformada Discreta de Fourier (do inglês *Discrete Fourier Transform*)

FIFO Primeiro a Entrar, Primeiro a Sair (do inglês *First-In, First-Out*)

FIR Resposta ao Impulso Finita (do inglês *Finite Impulse Response*)

FPGA Arranjo de Portas Programável em Campo (do inglês *Field-Programmable Gate Array*)

IDFT Transformada Discreta de Fourier Inversa (do inglês *Inverse Discrete Fourier Transform*)

PMU Unidade de Medição Fasorial (do inglês *Phasor Measurement Unit - PMU*)

ROCOF Taxa de Variação da Frequência (do inglês *Rate of Change of Frequency*)

SOS Seções de Segunda Ordem (do inglês *Second-Order Sections*)

TVE Erro Vetorial Total (do inglês *Total Vector Error*)

SEP Sistema Elétrico de Potência

GPS Sistema de Posicionamento Global (do inglês *Global Positioning System*)

PDC Concentrador de Dados Fasoriais (do inglês *Phasor Data Concentrator*)

IFFT Transformada Inversa Rápida de Fourier (do inglês *Inverse Fast Fourier Transform*)

FE Erro de Frequência (do inglês *Frequency Error*)

RFE Erro da Taxa de Variação da Frequência (do inglês *Rate of Change of Frequency Error*)

LISTA DE SÍMBOLOS

f_0 Frequência nominal.

f_m Frequência de modulação.

F_s Frequência de amostragem.

T_s Período de amostragem.

h Ordem harmônica.

H Número total de harmônicos.

A_h Amplitude do harmônico de ordem h.

ϕ_{0h} Fase inicial do harmônico de ordem h.

k_x Índice de modulação em amplitude.

k_a Índice de modulação em fase.

$X_{h[n]}$ Fasor harmônico de referência.

λ Fator de reamostragem ($\lambda = f_{\text{ref}}/f_{\text{inst}}$).

N_{ppc} Número nominal de pontos por ciclo.

M Fator de decimação do banco polifásico ($M = N_{ppc}$).

q Intensidade do ruído de processo do filtro de Kalman.

r Variância do ruído de medição do filtro de Kalman.

$\nu[n]$ Ruído aditivo.

A_{RMS} Valor eficaz da amplitude.

V_m Amplitude de pico da senoide.

V_{rms} Valor eficaz da senoide ($V_{rms} = V_m/\sqrt{2}$).

Δf Desvio de frequência em relação à nominal ($\Delta f = f - f_0$).

n Índice discreto.

t Variável temporal contínua.

CONTEÚDO

1 Introdução 14
1.1 Motivação 15
1.2 Objetivos 16
1.2.1 Objetivos específicos 16
1.3 Contribuições 17
1.4 Publicações 18
1.5 Divisão do trabalho 18
2 Revisão Bibliográfica e Fundamentação Teórica 19
2.1 Fasores e Sincrofasores 19
2.1.1 Representação Fasorial 19
2.1.2 Sincrofasores e PMUs 21
2.2 Fasores harmônicos 23
2.3 Estimação fasorial 24
2.3.1 Estimação sob frequência off-nominal 25
2.3.2 Normas e métricas de avaliação 27
2.4 Trabalhos correlatos 29
2.4.1 Estimação de fasores harmônicos 29
2.4.2 Reamostragem dinâmica e sincronização por frequência estimada 29
2.4.3 Implementações embarcadas 30
2.4.4 Posicionamento deste trabalho 30
2.5 Revisão das Técnicas Utilizadas no Trabalho 31
2.5.1 Interpolação e reamostragem dinâmica 31
2.5.2 Interpolação B-Spline e estrutura de Farrow 32
2.5.3 Filtros Flat-Top 33
2.5.4 Decomposição polifásica e bancos de filtros 34
2.6 Conclusões Parciais 35
3 Metodologia 36
3.1 Visão geral do método proposto 36
3.2 Arquitetura funcional da metodologia 37
3.2.1 Estimação de frequência por Zero-Crossing 39
3.2.2 Suavização da frequência por filtro de Kalman 40
3.2.2.1 Tentativas iniciais baseadas em média móvel 40
3.2.2.2 Filtro de Kalman adotado 41
3.2.3 Reamostragem por interpolação B-Spline 42
3.2.3.1 Estrutura de Farrow 43
3.2.4 Banco de filtros Flat-Top 45
3.2.5 Decomposição polifásica 46

3.2.6 Correção de fase . . . 47
3.2.7 Métricas para avaliação . . . 48
3.3 Conclusões Parciais . . . 49
4 Implementação e Validação na Plataforma SAPHO . . . 50
4.1 Parâmetros do sistema implementado . . . 50
4.2 Visão geral da arquitetura de implementação . . . 51
4.3 Implementação do `proc_interp` . . . 52
4.3.1 Pré-filtro do Zero-Crossing em arquitetura SOS . . . 52
4.3.2 Zero Crossing de meio ciclo . . . 55
4.3.3 Suavização por Filtro de Kalman . . . 57
4.3.4 Interpolação B-spline com Pré-filtro . . . 58
4.3.5 Compensação dos Atrasos e Descarte de Transitórios . . . 60
4.3.6 Atrasos e descartes de amostras . . . 60
4.4 Implementação do `proc_banco` . . . 62
4.5 Interface entre os dois processadores . . . 65
4.5.1 FIFOs entre os processadores . . . 66
4.5.2 Máquina de estados de controle . . . 67
4.6 Decisões impostas pela precisão limitada . . . 69
4.7 Conclusões Parciais . . . 72
5 Resultados . . . 74
5.1 Modelo dos sinais de teste . . . 74
5.2 Avaliação segundo a norma . . . 75
5.2.1 Regime *off-nominal* . . . 75
5.2.2 Rampa de frequência . . . 78
5.2.3 Modulação . . . 81
5.3 Comparação entre o modelo Python e a implementação SAPHO . . . 87
5.4 Conclusões Parciais . . . 88
6 Conclusões . . . 90
6.1 Trabalhos futuros . . . 91
REFERÊNCIAS . . . 93
APPENDIX A – DECLARAÇÃO DE USO DE INTELIGÊNCIA ARTIFICIAL GENERATIVA . . . 95

1 Introdução

A tensão das redes elétricas modernas envolve sinais cada vez mais distantes do modelo senoidal ideal. Conversores eletrônicos, cargas não lineares, geração distribuída, sistemas de armazenamento e fontes renováveis que utilizam dispositivos de eletrônica de potência alteram a forma de onda das tensões e correntes, introduzindo componentes harmônicas e inter-harmônicas, variações de amplitude, desvios de frequência e ruído. Nesse cenário, a análise apenas da componente fundamental pode ser insuficiente para caracterizar o estado do sistema e apoiar aplicações de qualidade de energia, diagnóstico de distorções e monitoramento em tempo real.

Diante deste contexto, a estimação fasorial sincronizada tornou-se uma ferramenta importante para a observação de Sistema Elétrico de Potência (SEP), pois associa grandezas elétricas medidas em diferentes pontos a uma base temporal comum. As Unidade de Medição Fasorial (do inglês *Phasor Measurement Unit - PMU*) (PMU) são capazes de estimar fasores, frequência e Taxa de Variação da Frequência (do inglês *Rate of Change of Frequency*) (ROCOF) da componente fundamental, permitindo comparar medições obtidas em locais distintos da rede (PHADKE; THORP, 2008; IEEE, 2011). Entretanto, a extensão desse conceito para componentes harmônicas ainda é menos consolidada. Enquanto os sincrofasores da fundamental possuem requisitos normativos bem definidos, a estimação sincronizada de fasores harmônicas ainda carece de padronização completa, especialmente no que diz respeito a ensaios, limites de erro e condições dinâmicas (IEC, 2018).

Essa lacuna é relevante porque as harmônicas não apenas acrescentam componentes espectrais ao sinal; eles também tornam mais sensível o problema de sincronismo temporal. Quando a frequência fundamental se afasta da nominal, filtros e janelas projetados para uma grade fixa passam a observar o sinal em instantes ligeiramente desalinhados. Esse desalinhamento produz erro de magnitude e, principalmente, erro de fase. Em componentes harmônicas, o efeito angular se torna mais crítico, pois a fase associada à ordem h possui mudança de h vezes a fase da fundamental. Assim, pequenas imprecisões na frequência estimada, nos atrasos de processamento ou nos instantes de amostragem podem gerar erros expressivos nas ordens mais elevadas.

Outro aspecto relevante é o custo computacional. Bancos de filtros são atrativos para a estimação harmônica por permitirem extrair simultaneamente várias componentes; no entanto, implementar um filtro dedicado a cada harmônica pode exigir um elevado número de operações. Esse custo torna-se muito elevado quando se deseja monitorar muitas ordens harmônicas ou quando a implementação deve operar em plataformas embarcadas. Nesse sentido, uma das possibilidades é o uso da decomposição polifásica, que reorganiza o banco de filtros de modo a reduzir a carga computacional, preservando a interpretação de filtragem multicanal (MITRA; KUO, 2006). Complementarmente, os filtros Flat-Top

destacam-se por apresentarem resposta de magnitude plana ao redor da frequência central, o que reduz a sensibilidade da estimação a pequenos desvios de frequência (DUDA *et al.*, 2020).

Além da extração harmônica propriamente dita, é necessário lidar com a variação da frequência fundamental antes que o sinal chegue ao banco de filtros. A alternativa mais frequente na literatura é reprojetar ou selecionar os filtros de acordo com a frequência estimada; essa estratégia, contudo, exige atualizar continuamente os coeficientes do banco, o que eleva a complexidade e o custo computacional da implementação. Outra possibilidade é reamostrar dinamicamente o sinal, gerando amostras em instantes compatíveis com o período elétrico estimado. Nesse caso, o banco posterior continua operando em uma base nominal, enquanto a adaptação à frequência real ocorre na etapa de interpolação, deslocando o custo do reprojeto dos filtros para uma etapa de interpolação mais leve. A interpolação B-Spline, implementada por estrutura de Farrow, é uma alternativa adequada para esse tipo de reamostragem por permitir a geração de amostras em posições fracionárias com coeficientes fixos (ALEIXO *et al.*, 2022).

Este trabalho investiga uma metodologia de estimação de fasores harmônicas que combina três elementos principais: estimação da frequência fundamental por cruzamento por zero, reamostragem dinâmica por interpolação B-Spline e extração harmônica por banco de filtros Flat-Top com decomposição polifásica. A metodologia é levada de um modelo de referência em software até uma implementação em processadores *soft-core*, avaliada por simulação funcional. É nessa etapa que os principais gargalos da cadeia são identificados e ajustados, desde a precisão numérica das operações acumuladas até o sincronismo entre os blocos de processamento, de modo que a metodologia opere corretamente dentro do tempo de processamento disponível. A síntese e os testes em hardware Arranjo de Portas Programável em Campo (do inglês *Field-Programmable Gate Array*) (FPGA) constituem o desdobramento natural desse desenvolvimento e ficam como proposta de trabalhos futuros; a ênfase, nesta etapa, recai sobre a integração da cadeia de processamento e sobre a análise de como as limitações da plataforma embarcada afetam a estimação nas ordens harmônicas mais elevadas.

1.1 Motivação

A motivação deste trabalho surge da necessidade de estimadores harmônicas que sejam, simultaneamente, precisos e viáveis para implementação embarcada. Em um cenário prático, o sinal medido pode apresentar ruído, distorção, desvio de frequência e modulações. Um estimador útil não pode depender apenas de condições estacionárias ideais: é preciso preservar a coerência de fase e de magnitude mesmo quando a frequência fundamental se desloca da nominal.

É desse requisito que decorre o uso da reamostragem dinâmica. Quando o sinal está

em frequência diferente da nominal, uma janela fixa deixa de representar exatamente o mesmo intervalo elétrico. A etapa de interpolação corrige essa incompatibilidade temporal antes da extração dos fasores, reduzindo o erro causado por analisar o sinal sobre uma grade inadequada. Como consequência, o banco de filtros pode ser mantido em uma base nominal, o que simplifica o processamento posterior.

A par da precisão, pesa o custo computacional da estimação de múltiplas ordens harmônicas. Extrair muitas componentes por meio de filtros independentes é oneroso, sobretudo em uma implementação em tempo real. Para contornar esse obstáculo, a decomposição polifásica reorganiza o banco de filtros de modo a reduzir o número de operações, ao passo que o filtro Flat-Top é adotado por sua baixa sensibilidade de magnitude nas proximidades da frequência de interesse. O cerne deste trabalho está em integrar essas técnicas em uma única cadeia de estimação.

1.2 Objetivos

O objetivo geral deste trabalho é implementar, em tempo real e sobre uma plataforma de processadores *soft-core*, uma metodologia de estimação de fasores harmônicas para sinais com frequência fundamental variável, que combina estimação de frequência por cruzamento por zero, reamostragem por interpolação B-Spline e banco de filtros Flat-Top com decomposição polifásica. Isso envolve avaliar a cadeia de processamento, identificar seus gargalos e realizar os ajustes necessários para que opere corretamente dentro do tempo de processamento disponível.

1.2.1 Objetivos específicos

Os objetivos específicos deste trabalho são:

- estudar a representação de fasores, sincrofasores e fasores harmônicas no contexto de sinais elétricos com frequência variável;
- discutir métricas e ensaios usados em PMU, utilizando a IEC/IEEE 60255-118-1 como referência metodológica para a construção de cenários de teste;
- formular a etapa de estimação de frequência por cruzamento por zero, incluindo filtragem preliminar, interpolação do instante de cruzamento e suavização da estimativa;
- explicar o papel do alinhamento temporal entre sinal, frequência estimada, filtros e interpolador;
- aplicar interpolação B-Spline em estrutura de Farrow para realizar a reamostragem dinâmica do sinal;

- organizar a extração dos fasores harmônicas por meio de um banco de filtros Flat-Top;
- empregar decomposição polifásica para reduzir o custo computacional associado ao banco de filtros;
- definir as métricas de avaliação: Erro Vetorial Total (do inglês *Total Vector Error*) (TVE), erro de magnitude, erro de fase, erro de frequência e erro de ROCOF, e aplicá-las na análise da cadeia implementada;
- implementar a cadeia de processamento em uma plataforma de processadores *soft-core*, particionando-a entre os núcleos e coordenando-os para operação em tempo real;
- identificar e corrigir os gargalos da implementação embarcada, tanto os de precisão numérica quanto os de sincronismo entre os blocos, de modo que a metodologia opere dentro do tempo de processamento disponível;
- validar a implementação por simulação funcional, comparando os resultados embarcados com os de um modelo de referência em software.

1.3 Contribuições

As principais contribuições deste trabalho são:

- a integração, em uma única cadeia de estimação harmônica, do rastreamento de frequência, da reamostragem B-Spline e do banco Flat-Top polifásico;
- a implementação dessa cadeia em tempo real sobre uma plataforma de processadores *soft-core*, com o particionamento entre núcleos e o mecanismo de sincronismo que viabilizam sua operação;
- a identificação e a correção dos gargalos de precisão numérica e de sincronismo que condicionam o funcionamento da metodologia na plataforma embarcada;
- o esclarecimento do papel da reamostragem dinâmica na redução do desalinhamento temporal provocado por frequência fora da nominal, e a formulação dos atrasos e alinhamentos que mantêm a coerência entre a frequência estimada e o trecho de sinal reamostrado;
- a adaptação de sinais de teste inspirados na IEC/IEEE 60255-118-1 para a análise de componentes harmônicas;
- a avaliação, por simulação funcional, de como as limitações da plataforma embarcada afetam a estimação nas ordens harmônicas mais elevadas.

1.4 Publicações

Parte dos resultados desta dissertação foi divulgada em dois artigos aceitos para publicação, ambos decorrentes do desenvolvimento da cadeia de estimação de fasores harmônicas aqui descrita:

- R. R. Rapozo, V. R. da Silva, V. V. de Morais, L. R. M. Silva e C. A. Duque, "Estimação em Tempo Real de Fasores harmônicas com Banco de Filtros Flat-Top em FPGA usando Soft-Core", aceito para publicação no *Congresso Brasileiro de Automática* (CBA), 2026.
- V. M. Ribeiro, V. V. Morais, R. R. Rapozo, L. R. M. Silva e C. A. Duque, "Interpolated Filter Bank for Harmonic Phasor Estimation with Kalman Frequency Tracking", aceito para publicação no *International Conference on Harmonics and Quality of Power* (ICHQP), 2026.

O primeiro artigo concentra-se na implementação em tempo real do banco de filtros Flat-Top sobre processadores *soft-core* em FPGA, tema central desta dissertação. O segundo trata do banco de filtros interpolado para estimação de fasores harmônicas com rastreamento de frequência por filtro de Kalman, aspecto diretamente relacionado às etapas de suavização e reamostragem desenvolvidas neste trabalho.

1.5 Divisão do trabalho

O restante desta dissertação está organizado da seguinte forma.

O Capítulo 2 apresenta a revisão bibliográfica e a fundamentação teórica. São discutidos fasores, sincrofasores, estimação fasorial, fasores harmônicas, normas e métricas de avaliação, interpolação, B-Splines, estrutura de Farrow, filtros Flat-Top e decomposição polifásica.

O Capítulo 3 descreve a metodologia proposta. Nesse capítulo, os conceitos apresentados separadamente na fundamentação são integrados em uma cadeia composta por estimação de frequência, suavização, alinhamento temporal, reamostragem dinâmica, banco de filtros polifásico, correção de fase e métricas de avaliação.

O Capítulo 5 é reservado para resultados e discussões. Essa etapa deverá reunir, quando consolidados, os ensaios em condições estacionárias e dinâmicas, bem como a análise dos erros obtidos na estimação dos fasores harmônicas.

Por fim, a conclusão sintetiza os principais pontos desenvolvidos, discute limitações da etapa atual e indica os próximos passos relacionados à implementação em FPGA e à validação experimental do método.

2 Revisão Bibliográfica e Fundamentação Teórica

Este capítulo reúne os fundamentos necessários à compreensão da metodologia implementada nesta dissertação e a posiciona frente à literatura. Inicialmente, são introduzidas as representações fasorial e sincrofasorial, que definem como sinais elétricos senoidais são descritos por magnitude e fase. Essa descrição é então estendida às componentes harmônicas, evidenciando por que o erro de fase cresce com a ordem harmônica. Em seguida, aborda-se o problema de estimação a partir de sinais amostrados, o efeito da frequência off-nominal e as métricas usadas na avaliação. Por fim, após situar o trabalho em relação à literatura, apresentam-se as quatro técnicas que compõem a cadeia implementada: reamostragem dinâmica, interpolação B-Spline por estrutura de Farrow, filtros Flat-Top e decomposição polifásica, cuja integração é detalhada no Capítulo 3.

2.1 Fasores e Sincrofasores

Esta seção introduz as representações que fundamentam todo o trabalho. Primeiro, revisita-se o conceito de fasor, que descreve uma grandeza senoidal por magnitude e ângulo de fase, e sua extensão para condições de frequência variável. Em seguida, apresenta-se o sincrofasor, que associa essa representação a uma referência temporal comum, viabilizando a comparação de medições em pontos distintos do sistema elétrico por meio das PMU.

2.1.1 Representação Fasorial

Um fasor é uma representação complexa de uma grandeza senoidal em regime permanente. Essa representação é amplamente utilizada em sistemas elétricos porque permite substituir a descrição temporal completa de uma senoide por duas informações principais: magnitude e ângulo de fase. Com isso, a análise de circuitos e sistemas em corrente alternada torna-se mais simples, uma vez que a dependência temporal associada à frequência fundamental é separada da representação complexa da grandeza (PHADKE; THORP, 2008).

Considere o sinal senoidal

$$v(t) = V_m \cos(\omega t + \varphi), \tag{2.1}$$

em que V_m é a amplitude de pico, ω é a frequência angular e φ é o ângulo de fase em relação a uma referência temporal. O valor eficaz associado à senoide é $V_m/\sqrt{2}$. Assim, o fasor correspondente pode ser escrito como

$$\bar{V} = \frac{V_m}{\sqrt{2}} e^{j\varphi}. \tag{2.2}$$

Na forma polar, a mesma grandeza é representada por

$$\bar{V} = V_{rms}\angle\varphi, \tag{2.3}$$

em que V_{rms} é o valor eficaz da grandeza. A informação temporal ωt não aparece explicitamente no fasor, pois a representação assume uma frequência de referência conhecida. Assim, o fasor convencional não descreve diretamente a evolução temporal do sinal; ele representa o módulo e o atraso angular da senoide em relação à referência adotada. Essa característica é adequada para regime permanente, mas torna-se mais delicada quando a frequência varia ao longo do tempo.

A Figura 1 ilustra a relação entre uma senoide e sua representação no plano complexo. O ângulo do fasor indica a defasagem em relação à referência adotada, enquanto seu módulo está associado ao valor eficaz do sinal.

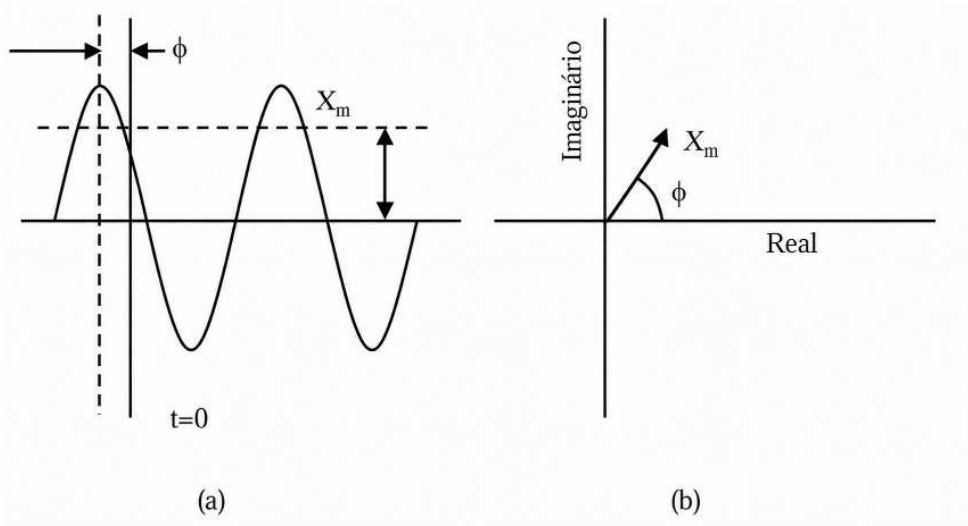


Figura 1 – Representação de uma senoide e de seu fasor associado no plano complexo, adaptada de (PHADKE; THORP, 2008).

A representação fasorial das Equações 2.2 e 2.3 pressupõe que tanto a amplitude quanto a frequência permaneçam fixas. Em um sistema elétrico em operação, no entanto, essas grandezas sofrem pequenas variações ao longo do tempo, de modo que resumir o sinal a um único fasor constante torna-se uma aproximação restrita ao regime permanente. Uma descrição mais geral mantém o ponto de partida no domínio do tempo, mas permite que a amplitude e a frequência sejam funções do tempo. Quando a frequência varia, o argumento do cosseno não pode mais ser escrito como o produto ωt, pois esse produto pressupõe uma taxa de giro constante. Em seu lugar, utiliza-se o ângulo de fase total efetivamente acumulado pelo sinal, obtido pela integral da frequência instantânea $f(t)$,

$$v(t) = V_m(t)\cos\left(2\pi \int_0^t f(\tau)\, d\tau + \varphi\right), \tag{2.4}$$

em que $V_m(t)$ é a amplitude de pico, agora variável, e o termo integral acumula a fase à medida que a frequência muda. É conveniente separar a frequência instantânea em duas parcelas: a frequência nominal f_0, conhecida e fixa, e um desvio $\Delta f(t)$ que reúne as variações decorrentes da operação do sistema, isto é, $f(t) = f_0 + \Delta f(t)$. Levando essa decomposição à Equação (2.4), o ângulo acumulado separa-se em uma parcela nominal e uma parcela de desvio,

$$v(t) = V_m(t)\cos\!\left(2\pi f_0 t + 2\pi \int_0^t \Delta f(\tau)\, d\tau + \varphi\right). \tag{2.5}$$

O termo $2\pi f_0 t$ representa o giro na frequência de referência, enquanto a integral de Δf concentra todo o efeito acumulado do afastamento em relação à nominal. Adotando a frequência nominal como referência girante e suprimindo o termo $2\pi f_0 t$, obtém-se o fasor dinâmico

$$\bar{V}(t) = V_{rms}(t)\, e^{\,j\left(2\pi \int_0^t \Delta f(\tau)\, d\tau + \varphi\right)}, \tag{2.6}$$

com $V_{rms}(t) = V_m(t)/\sqrt{2}$. Ao contrário do fasor estático, o fasor dinâmico carrega de forma explícita a dependência temporal introduzida pelo desvio de frequência: enquanto $\Delta f(\tau)$ for diferente de zero, o argumento da exponencial cresce continuamente e o fasor gira no referencial nominal, em vez de permanecer estacionário.

Um caso particular de interesse ocorre quando a amplitude é constante e o desvio de frequência também se mantém fixo, $\Delta f(t) = \Delta f$. A integral reduz-se então a $2\pi \Delta f\, t = (\omega - \omega_0)t$, e a Equação (2.6) assume a forma compacta

$$\bar{V}(t) = V_{rms}\, e^{\,j[(\omega-\omega_0)t+\varphi]}, \tag{2.7}$$

que descreve um fasor girando a uma taxa angular constante $(\omega - \omega_0)$ em torno da referência nominal. Essa forma é retomada na Seção 2.3 ao se analisar o efeito da frequência fora da nominal sobre a estimação fasorial.

2.1.2 Sincrofasores e PMUs

Um sincrofasor é um fasor associado a uma referência temporal comum, o que permite comparar de forma coerente medições realizadas em pontos geograficamente distintos de um sistema elétrico. Em sistemas de medição fasorial sincronizada, essa base de tempo é normalmente fornecida por fontes de sincronização de alta precisão, como o Sistema de Posicionamento Global (do inglês *Global Positioning System*) (GPS), e utilizada por unidades de medição fasorial, conhecidas como PMU (PHADKE; THORP, 2008; IEEE, 2011).

O que distingue um sincrofasor de um fasor convencional é justamente a associação explícita a uma estampa de tempo: como o ângulo de fase depende do instante tomado como referência, duas medições só podem ser comparadas diretamente se estiverem sincronizadas. Fasores medidos em barras distintas passam, assim, a ser tratados como observações de um mesmo estado elétrico, com a coerência angular preservada entre pontos afastados do sistema. Essa propriedade é fundamental em aplicações de monitoramento de grandes áreas, estimação de estado, proteção, controle e análise dinâmica de sistemas elétricos (PHADKE; THORP, 2008; MARTINS, 2012).

A Figura 2 apresenta uma representação simplificada de um sistema elétrico composto por geradores interligados por meio de barramentos. Em cada ponto de medição, a tensão local é uma senoide caracterizada por um fasor próprio, representado pelo vetor girante no plano complexo associado àquela forma de onda. Embora as tensões observadas nas diferentes barras apresentem magnitudes e, sobretudo, ângulos de fase distintos entre si, todos os fasores são referidos a uma mesma base temporal de sincronização. É justamente essa referência comum que permite comparar diretamente os ângulos medidos em barras geograficamente afastadas e relacioná-los ao estado operativo da rede; sem ela, cada ângulo estaria atrelado a uma origem de tempo local arbitrária e a comparação entre pontos distintos perderia o sentido.

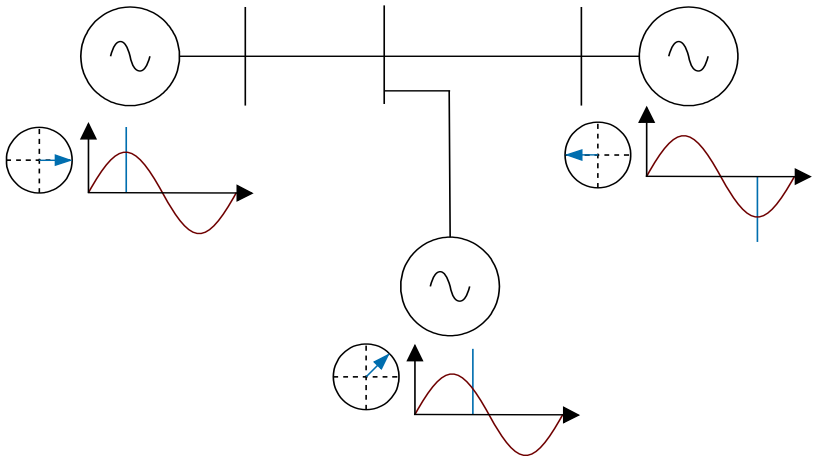


Figura 2 – Representação de medições fasoriais sincronizadas em diferentes barras de um sistema elétrico: cada ponto de medição possui um fasor de tensão próprio, com magnitude e ângulo distintos, todos referidos a uma base temporal comum.

De forma geral, uma PMU realiza a aquisição dos sinais de tensão e corrente, aplica etapas de condicionamento e processamento digital, estima fasores, frequência e taxa de variação da frequência, e envia essas informações a um concentrador de dados fasoriais, ou Concentrador de Dados Fasoriais (do inglês *Phasor Data Concentrator*) (PDC). Esse arranjo é ilustrado na Figura 3.

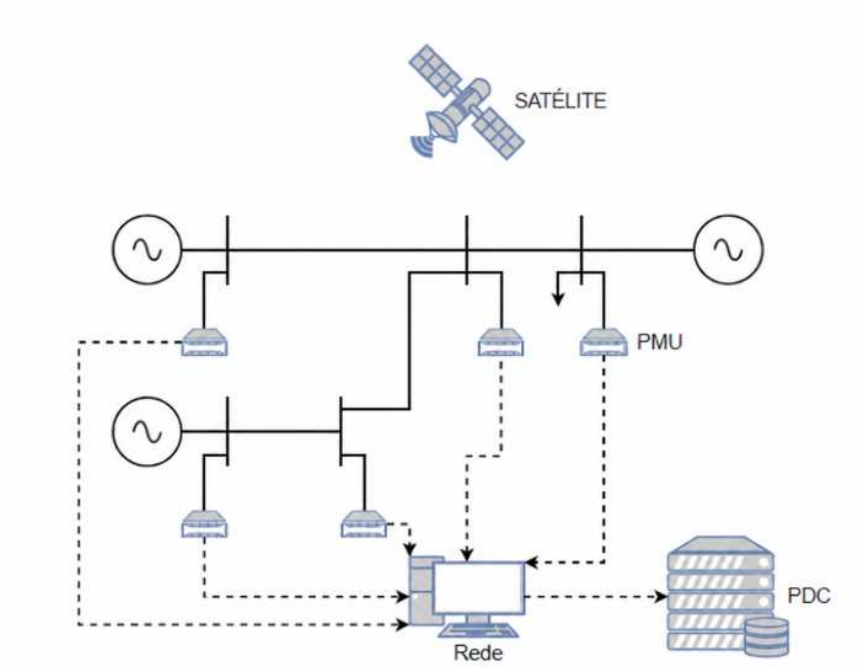


Figura 3 – Estrutura geral de um sistema de medição fasorial sincronizada, adaptada de (LUIZ, 2024).

2.2 Fasores harmônicos

Sinais elétricos reais podem conter componentes harmônicas decorrentes de cargas não lineares, conversores eletrônicos, saturação de equipamentos e outras fontes de distorção. Um sinal periódico não puramente senoidal pode ser representado como uma soma de componentes em múltiplos inteiros da frequência fundamental. Considerando uma frequência fundamental f_1, e seguindo a forma geral de sinais de ensaio adotada na IEC/IEEE 60255-118-1 (IEC, 2018), uma representação simplificada é

$$x(t) = A_1 \cos(2\pi f_1 t + \varphi_1) + \sum_{h=2}^{H} A_h \cos(2\pi h f_1 t + \varphi_h), \tag{2.8}$$

em que, A_1 é a amplitude de pico da componente fundamental, f_1 é a frequência fundamental e φ_1 é a fase inicial da fundamental. Para as componentes harmônicas, h representa a ordem harmônica, H é a maior ordem considerada, A_h é a amplitude de pico da componente de ordem h e φ_h é a fase inicial, ou defasagem angular, dessa componente. Assim, A_h define o tamanho da harmônica, enquanto φ_h define seu deslocamento angular em relação à referência adotada. A frequência da componente de ordem h é hf_1; por exemplo, a segunda harmônica está em $2f_1$ e a quinta harmônica está em $5f_1$.

O fasor associado à componente de ordem h pode ser escrito como

$$\bar{V}_h = \frac{A_h}{\sqrt{2}} e^{j\varphi_h}, \tag{2.9}$$

em que, o termo $A_h/\sqrt{2}$ corresponde ao valor eficaz da componente harmônica, pois A_h foi definido como amplitude de pico. O fator $e^{j\varphi_h}$ armazena a informação angular. Portanto, a amplitude e a fase representam aspectos diferentes do mesmo fasor: a primeira indica o módulo da componente e a segunda indica seu adiantamento ou atraso angular.

Quando a frequência fundamental varia, a fase da componente harmônica também passa a depender da trajetória temporal da frequência. Para deixar essa relação explícita, define-se $\theta_1(t)$ como o ângulo elétrico acumulado pela fundamental, sem incluir a fase inicial. Em frequência constante, $\theta_1(t) = 2\pi f_1 t$; em frequência variável, esse ângulo é obtido pela acumulação da frequência instantânea. A fase total da fundamental é, portanto, $\theta_1(t) + \varphi_1$. Para a componente harmônica de ordem h, a parte que gira no tempo é multiplicada por h, pois essa componente oscila h vezes mais rápido que a fundamental. Assim, sua fase instantânea pode ser escrita como

$$\theta_h(t) = h\theta_1(t) + \varphi_h, \tag{2.10}$$

em que, $h\theta_1(t)$ representa a evolução angular causada pela frequência da harmônica, enquanto φ_h representa apenas a defasagem inicial dessa harmônica. Se essa defasagem for interpretada como um atraso temporal τ_h, então, para frequência constante, tem-se $\varphi_h = -2\pi h f_1 \tau_h$. Isso mostra que atraso temporal e fase angular não são a mesma grandeza: o mesmo atraso em segundos produz uma defasagem angular maior nas harmônicas de ordem mais elevada.

Essa relação mostra por que a estimação de fasores harmônicos é sensível ao erro de frequência e ao desalinhamento temporal. Um pequeno erro na fase da componente fundamental tende a produzir erro proporcionalmente maior nas componentes de ordem elevada. Assim, técnicas de estimação harmônica precisam lidar simultaneamente com seletividade em frequência, precisão de fase e custo computacional.

A estimação sincronizada de fasores harmônicos ainda é um tema em desenvolvimento. Há aplicações potenciais em monitoramento de qualidade de energia, estimação de estado harmônico, localização de fontes de distorção, avaliação de filtros ativos e diagnóstico de fenômenos em redes com elevada penetração de eletrônica de potência. No entanto, diferentemente dos sincrofasores da componente fundamental, ainda não há, nas normas de PMU consideradas neste trabalho, requisitos de desempenho específicos para a estimação sincronizada de fasores harmônicos.

2.3 Estimação fasorial

Definidas as representações fasorial e harmônica, resta o problema de obtê-las na prática, a partir de um número finito de amostras. Esta seção trata desse problema em dois momentos. Primeiro, examina-se como o desvio da frequência fundamental em relação

à nominal degrada a estimação, efeito que se agrava nas componentes de ordem elevada. Em seguida, apresentam-se as normas e as métricas empregadas para quantificar o erro dessa estimação, que servem de base à avaliação conduzida nos capítulos posteriores.

2.3.1 Estimação sob frequência off-nominal

Na prática, os fasores não são obtidos a partir de expressões analíticas contínuas, mas de sinais discretos adquiridos por conversores analógico-digitais. O problema da estimação fasorial consiste, portanto, em estimar magnitude, fase, frequência e, quando necessário, ROCOF a partir de um conjunto finito de amostras.

Esse problema pode ser abordado por diferentes famílias de métodos, cada uma partindo de um princípio próprio de análise do sinal. Entre as técnicas mais consolidadas estão métodos baseados na Transformada Discreta de Fourier (do inglês *Discrete Fourier Transform*) (DFT), filtros digitais, filtros de Kalman, métodos de mínimos quadrados, a Transformada de Taylor-Fourier e algoritmos interpolados no domínio da frequência. Cada família apresenta compromissos próprios entre precisão, tempo de resposta, robustez a ruído, rejeição de interferências e custo computacional.

Em condições estacionárias e com frequência nominal, métodos baseados na DFT tendem a apresentar boa precisão e implementação simples. Entretanto, quando a frequência do sinal se desvia da frequência nominal, a janela de observação deixa de conter um número inteiro de ciclos. Esse desalinhamento provoca espalhamento espectral e afeta a estimação da magnitude e da fase (PHADKE; THORP, 2008; HARRIS, 1978). O problema se torna ainda mais relevante em aplicações harmônicas, pois o erro de tempo ou de fase aparece no argumento das componentes em múltiplos da frequência fundamental. Assim, uma mesma incerteza angular associada à base temporal da fundamental tende a produzir efeito mais severo nas componentes de ordem elevada. Esse comportamento está no contexto dos desafios tratados em métodos de estimação de fasores harmônicos sob frequência variável como (DUDA *et al.*, 2020; ALEIXO *et al.*, 2022).

Essa dificuldade pode ser observada diretamente a partir do fasor dinâmico introduzido em (2.7). Considere uma senoide de valor eficaz V_{rms}, com amplitude constante e frequência angular real ω fixa, porém afastada da nominal ω_0,

$$v(t) = \sqrt{2}\, V_{\mathrm{rms}} \cos(\omega t + \varphi). \tag{2.11}$$

Tomando a frequência nominal ω_0 como referência girante, conforme já discutido, a componente complexa associada ao sinal observada nesse referencial corresponde ao caso de desvio de frequência constante da Equação 2.7,

$$\bar{V}(t) = V_{\mathrm{rms}}\, e^{j[(\omega-\omega_0)t+\varphi]}. \tag{2.12}$$

Quando $\omega = \omega_0$, o termo dependente do tempo desaparece e o fasor permanece parado no referencial síncrono, com ângulo constante φ. Porém, quando $\omega \neq \omega_0$, surge uma rotação residual dada por

$$\varphi(t) = (\omega - \omega_0)t + \varphi. \tag{2.13}$$

Portanto, a frequência fora da nominal não produz apenas um erro estático: ela faz o ângulo estimado variar continuamente ao longo da janela de observação. Em um estimador projetado para a frequência nominal, o desvio de frequência faz com que o fasor estimado continue girando em relação à referência, de modo que a fase estimada varia continuamente ao longo do tempo. A magnitude também é afetada, pois a estimação fasorial pode ser interpretada como uma projeção do sinal sobre uma base complexa, ou, de forma equivalente, como a resposta de um filtro centrado em uma frequência específica. Quando a componente analisada coincide com essa frequência central, o ganho do estimador é aproximadamente unitário. Entretanto, quando sua frequência está deslocada da nominal, a componente deixa de projetar-se integralmente sobre a base utilizada pelo estimador, ocasionando espalhamento espectral e um ganho efetivo inferior à unidade. Como consequência, mesmo uma senoide pura, na ausência de harmônicos, pode apresentar erro de magnitude quando analisada por uma janela ou filtro centrado na frequência nominal.

Para componentes harmônicas, o mesmo raciocínio se torna mais crítico. Se a fundamental possui desvio $\Delta f = f - f_0$, a componente de ordem h aparece em torno de $h\Delta f$, enquanto o canal harmônico nominal é centrado em hf_0. O desvio visto por esse canal é, portanto,

$$\Delta f_h = h(f - f_0) = h\Delta f. \tag{2.14}$$

Dessa forma, a rotação residual e o desalinhamento espectral crescem com a ordem harmônica. Uma pequena diferença de frequência na fundamental pode se tornar um deslocamento significativo nas ordens mais elevadas, aumentando os erros de fase e de magnitude.

Esse contexto motiva o uso de técnicas que adaptem a análise à frequência real do sinal. Uma alternativa é reprojetar ou selecionar os filtros de acordo com a frequência estimada. Nesse caso, a janela pode manter o mesmo comprimento, mas a resposta do filtro deixa de estar centrada em f_0 e passa a ser centrada na frequência real estimada, ou em $h\hat{f}$ no caso de um canal harmônico. Isso reduz o erro porque a componente de interesse volta a ficar próxima da região de ganho unitário e de fase adequada do estimador. Tal ajuste, porém, não elimina todos os efeitos da janela: se a frequência variar dentro da própria janela, se houver ruído, harmônicos próximos ou atraso de estimação,

ainda haverá erro residual. Além disso, em um banco com muitas ordens harmônicas, atualizar continuamente os coeficientes de todos os canais pode aumentar a complexidade da implementação.

Outra possibilidade consiste em reamostrar dinamicamente o sinal, de modo que o estágio de estimação continue operando em uma base temporal nominal. Nesse procedimento, as amostras originais continuam sendo adquiridas pelo *clock* fixo do conversor analógico-digital. A partir da frequência estimada, determinam-se os instantes em que o sinal deveria ser observado para que cada ciclo elétrico seja representado pelo número nominal de amostras utilizado pelo banco de filtros. Como esses instantes de reamostragem normalmente situam-se entre duas amostras físicas, emprega-se a interpolação B-Spline para reconstruir o sinal em posições fracionárias. Dessa forma, o método não altera o fenômeno medido, mas ajusta a base temporal fornecida ao estimador. Como consequência, a rotação residual de fase causada pelo desvio de frequência é significativamente reduzida, permitindo que o banco de filtros opere em condições mais próximas daquelas para as quais foi projetado.

2.3.2 Normas e métricas de avaliação

As normas IEEE C37.118.1 e IEC/IEEE 60255-118-1 definem requisitos de desempenho para PMU na estimação de sincrofasores, frequência e ROCOF da componente fundamental (IEEE, 2011; IEEE, 2014; IEC, 2018). Essas normas estabelecem ensaios para condições estacionárias e dinâmicas, incluindo frequência fora da nominal, rampa de frequência, modulação de amplitude e modulação de fase.

Como essas normas tratam da componente fundamental, elas não são adotadas aqui como especificação normativa para fasores harmônicos. Seus ensaios são utilizados apenas como referência metodológica para a construção de cenários de teste contendo componentes harmônicas. Dessa forma, avalia-se o comportamento do método em condições inspiradas nos ensaios de PMU, mas sem atribuir à norma limites de desempenho para fasores harmônicos.

A métrica mais utilizada para avaliação fasorial é o TVE, que compara o fasor estimado com o fasor de referência no plano complexo. Na IEC/IEEE 60255-118-1, essa métrica é definida para o sincrofasor da componente fundamental por (IEC, 2018)

$$TVE_{\text{pu}} = \sqrt{\frac{\left(\hat{X}_r - X_r\right)^2 + \left(\hat{X}_i - X_i\right)^2}{X_r^2 + X_i^2}}, \tag{2.15}$$

em que $\hat{X}_r$ e $\hat{X}_i$ são as partes real e imaginária do fasor estimado, e X_r e X_i são as partes real e imaginária do fasor de referência. A Equação 2.15 fornece o TVE em valor normalizado, ou pu. Quando o resultado é apresentado em porcentagem, utiliza-se

$$TVE_{\%} = 100\,TVE_{\mathrm{pu}}. \tag{2.16}$$

De forma equivalente, o Anexo C da IEC/IEEE 60255-118-1 relaciona o TVE aos erros de magnitude e fase por

$$TVE_{\mathrm{pu}} = \sqrt{2(1+ME)\left[1-\cos(PE)\right]+ME^2}, \tag{2.17}$$

em que ME é o erro de magnitude em pu e PE é o erro de fase. Na Equação 2.17, o argumento do cosseno deve estar em unidade angular compatível com a função utilizada; na forma usual, emprega-se PE em radianos. O erro de magnitude pode ser escrito como

$$ME = \frac{\sqrt{\hat{X}_r^2+\hat{X}_i^2}-\sqrt{X_r^2+X_i^2}}{\sqrt{X_r^2+X_i^2}}, \tag{2.18}$$

e o erro de fase como

$$PE = \mathrm{atan2}(\hat{X}_i,\hat{X}_r) - \mathrm{atan2}(X_i,X_r). \tag{2.19}$$

Assim, a forma em coordenadas retangulares da Equação (2.15) e a forma em termos de ME e PE da Equação 2.17 representam a mesma métrica. Neste trabalho, quando essa métrica for aplicada a componentes harmônicas, seu uso deve ser entendido como uma extensão metodológica por componente, e não como um requisito normativo da IEC/IEEE 60255-118-1 para fasores harmônicos.

Além do TVE, a norma IEC/IEEE 60255-118-1 define, para a medição da componente fundamental, o erro de frequência, dado por (IEC, 2018)

$$FE = \hat{f} - f, \tag{2.20}$$

e o erro de ROCOF,

$$RFE = \widehat{ROCOF} - ROCOF. \tag{2.21}$$

Para interpretar o desempenho de estimadores harmônicos, também é útil separar o erro de magnitude e o erro de fase. O erro de magnitude indica a diferença entre os módulos dos fasores, enquanto o erro de fase indica o desalinhamento angular. Essa separação é importante porque dois estimadores podem apresentar TVE semelhante, mas por razões distintas: um pode ser limitado por erro de ganho, enquanto outro pode ser limitado por erro de fase.

2.4 Trabalhos correlatos

Esta seção situa o trabalho em relação à literatura, organizando-a nos três eixos que a metodologia proposta combina: a estimação de fasores harmônicos propriamente dita, a reamostragem dinâmica usada para lidar com desvios de frequência e as implementações embarcadas desses estimadores.

2.4.1 Estimação de fasores harmônicos

Os métodos de estimação fasorial costumam ser agrupados conforme o domínio em que operam. No domínio da frequência, predominam abordagens baseadas na DFT, que apresentam bom desempenho em regime permanente mas se degradam sob condições dinâmicas por causa do espalhamento espectral. A DFT interpolada e suas variantes aprimoradas foram propostas justamente para compensar esse espalhamento e a interferência entre componentes vizinhas (MACII; PETRI *et al.*, 2014; ROMANO; PAOLONE, 2019). Ainda assim, tais métodos permanecem sensíveis à presença de inter-harmônicos.

No caso específico das componentes harmônicas, o problema é mais severo, pois o desvio de frequência é multiplicado pela ordem h, como mostra a Equação 2.14. Diversas estratégias foram propostas para tratá-lo. Em (CHEN; KEZUNOVIC; LIAO, 2014), o algoritmo de estimação de sincrofasores é modificado para acomodar componentes harmônicas. Em (CHEN *et al.*, 2018), cada harmônica é modelada como soma ponderada de funções sinc, com a estimação realizada por um banco de filtros Resposta ao Impulso Finita (do inglês *Finite Impulse Response*) (FIR). Já (LI *et al.*, 2021) trata explicitamente o desvio de frequência na estimação dinâmica de fasores harmônicos.

Uma linha particularmente relevante para este trabalho é a que emprega **filtros e janelas Flat-Top**. A propriedade de resposta de magnitude plana em torno da frequência central torna a estimação de amplitude pouco sensível a pequenos desvios de frequência, característica explorada desde os estudos clássicos sobre janelas (HARRIS, 1978) e formalizada em projetos específicos de janelas Flat-Top (DUDA; ZIELIŃSKI; BARCZENTEWICZ, 2016). Sua aplicação direta à estimação de fasores harmônicos por meio de filtros FIR Flat-Top é apresentada em (DUDA *et al.*, 2020), e constitui o ponto de partida para o banco de filtros adotado nesta dissertação.

2.4.2 Reamostragem dinâmica e sincronização por frequência estimada

Uma alternativa a reprojetar continuamente os filtros em função da frequência medida é ajustar a *base temporal* do sinal, reamostrando-o de modo que o número de amostras por ciclo permaneça constante. Essa ideia aparece na literatura sob o nome de reamostragem coerente (BORKOWSKI; BIEŃ, 2009) e, no contexto de sincrofasores, como ajuste do valor amostrado (MAHARJAN *et al.*, 2017; MAHARJAN; PENG; MARTINEZ,

2016). Métodos de estimação da própria frequência, necessários para comandar esse ajuste, também foram desenvolvidos por interpolação sobre coeficientes de Fourier (ABOUTANIOS; MULGREW, 2010) e por correção da estimativa em frequências fora da nominal (GHAFARI *et al.*, 2016).

A realização prática da reamostragem exige um interpolador de atraso fracionário eficiente. As funções B-Spline, introduzidas em (SCHOENBERG, 1946), oferecem uma base adequada, e formulações causais apropriadas à implementação em tempo real são discutidas em (PETRINOVIC, 2008). A aplicação dessa estrutura especificamente à estimação de fasores harmônicos em tempo real é demonstrada em (ALEIXO *et al.*, 2022), trabalho que fundamenta o interpolador adotado nesta dissertação.

2.4.3 Implementações embarcadas

O terceiro eixo diz respeito à viabilidade de executar esses algoritmos em *hardware* dedicado. Implementações de estimadores baseados em DFT interpolada em FPGA são descritas em (ROMANO; PAOLONE, 2014), com análise de desempenho e de custo de recursos, e estendidas a um protótipo completo de PMU de baixo custo voltado a redes de distribuição em (ROMANO *et al.*, 2017). Esses trabalhos evidenciam que o gargalo, em geral, não está na formulação matemática, mas no número de operações por amostra que a plataforma consegue sustentar.

É nesse ponto que a decomposição polifásica se torna decisiva, ao reduzir o custo do banco de filtros de uma ordem proporcional a NM para $N + M \log_2 M$. Sua aplicação à estimação de fasores harmônicos foi investigada em trabalhos anteriores do mesmo grupo de pesquisa (LUIZ, 2024; VICTOR, 2024), que avaliaram o banco Flat-Top polifásico em ambiente de simulação.

2.4.4 Posicionamento deste trabalho

Os três eixos acima são, em geral, tratados separadamente na literatura. Trabalhos sobre filtros Flat-Top concentram-se na resposta em frequência do estimador; trabalhos sobre reamostragem tratam da sincronização temporal; e trabalhos sobre implementação embarcada partem, quase sempre, de estimadores baseados em DFT. A contribuição desta dissertação está em **integrar os três**: encadear a estimação de frequência, a reamostragem dinâmica por B-Spline e o banco Flat-Top em forma polifásica em uma única cadeia de processamento, e levá-la efetivamente a uma implementação em *hardware* descrita e validada por simulação, avaliando o quanto as limitações da plataforma embarcada afetam a estimação nas ordens harmônicas elevadas.

2.5 Revisão das Técnicas Utilizadas no Trabalho

Esta seção reúne os fundamentos das quatro técnicas que compõem a cadeia implementada, na ordem em que atuam sobre o sinal. Primeiro, a reamostragem dinâmica ajusta a base temporal do sinal à frequência estimada, de modo que o estágio seguinte opere sempre com o número nominal de pontos por ciclo; a interpolação em posições fracionárias que ela exige é realizada por B-Splines em estrutura de Farrow. O sinal reamostrado alimenta então um banco de filtros Flat-Top, escolhido por sua baixa sensibilidade de magnitude a pequenos desvios de frequência, cuja implementação eficiente se apoia na decomposição polifásica. Cada técnica é apresentada a seguir de forma individual; sua integração em um único fluxo de processamento é objeto do Capítulo 3.

2.5.1 Interpolação e reamostragem dinâmica

O banco de filtros polifásico é projetado assumindo um número fixo de pontos por ciclo, $N_{ppc} = F_s/f_0$. Como a aquisição ocorre em taxa fixa $F_s = f_0 N_{ppc}$, mas a frequência f do sinal varia em relação a f_0, o número real de amostras contidas em um ciclo deixa de ser N_{ppc}. Essa quebra de premissa se manifesta como desalinhamento espectral e aparece, sobretudo, como erro de fase, mais severo nos harmônicos de ordem elevada.

Reprojetar o banco a cada nova frequência seria inviável em hardware embarcado. A estratégia adotada pelo processo de reamostragem é deslocar a correção para uma etapa de reamostragem comandada pela frequência estimada $\hat{f}$. A ideia é converter a sequência adquirida no tempo, presa ao relógio de aquisição, em uma sequência observada na *fase* do fundamental estimado: cada amostra de saída avança sempre a mesma fração de ciclo, de modo que um ciclo elétrico corresponda sempre a N_{ppc} amostras, qualquer que seja $\hat{f}$. Note que a fase inicial do sinal não é eliminada, pois ela faz parte do fasor que se deseja medir; o que se corrige é o descompasso acumulativo entre a grade temporal nominal e a frequência real.

O passo de reamostragem decorre diretamente dessa exigência. Em $N_{ppc} = T_f/T_s$ há três grandezas: N_{ppc} que é um número fixo imposto pelo projeto do banco; o período do sinal $T_f = 1/\hat{f}$, que o sistema impõe e varia; e o passo de amostragem, único grau de liberdade. Mantendo N_{ppc} fixo, isola-se o passo de tempo que a saída deve ter,

$$T_{s_{\text{out}}} = \frac{T_f}{N_{ppc}} = \frac{1}{N_{ppc}\,\hat{f}}. \tag{2.22}$$

Comparando esse passo com o passo de entrada $T_s = 1/(N_{ppc} f_0)$, define-se o fator de reamostragem

$$\lambda = \frac{T_{s_{\text{out}}}}{T_s} = \frac{f_0}{\hat{f}}. \tag{2.23}$$

Portanto, λ não é um parâmetro arbitrário: ele decorre diretamente da imposição de N_{ppc} fixo. O passo de tempo da saída é o passo de amostragem modulado por λ e, por isso, acompanha as variações de $\hat{f}$, enquanto o passo de fase permanece constante.

O sentido do efeito segue dessa relação. Quando $\hat{f} > f_0$, tem-se $\lambda < 1$: as amostras do sinal amostrado de saída ficam mais juntas no eixo original e há mais amostras por ciclo, pois o ciclo real é mais curto. Para $f_0 = 60\,\text{Hz}$ e $\hat{f} = 61\,\text{Hz}$, por exemplo, um ciclo real ocupa cerca de $(60/61)N_{ppc}$ amostras físicas, e $\lambda = 60/61$ recupera exatamente N_{ppc} amostras por ciclo estimado. Quando $\hat{f} < f_0$, tem-se $\lambda > 1$ e o efeito se inverte. Em ambos os casos, a taxa do conversor A/D não muda.

Na implementação, é conveniente trabalhar com a posição em unidades de amostra original, $p_i[k] = F_s\, t_i[k]$, que avança a cada saída segundo

$$p_i[k+1] = p_i[k] + \lambda. \tag{2.24}$$

A reamostragem não edita nem descarta amostras do *buffer*: toda amostra de saída é um valor novo, interpolado. Para calculá-la, separa-se a posição em parte inteira $n_k = \lfloor p_i[k] \rfloor$, que indica o intervalo de entrada, e parte fracionária $\alpha_k = p_i[k] - n_k$, a fase fracionária dentro desse intervalo. O interpolador B-Spline, detalhado na Seção 2.5.2, usa α_k e as amostras vizinhas para estimar o valor do sinal naquele instante fracionário. Assim, a discrepância de frequência não é ignorada, mas usada para escolher os instantes de interpolação, reduzindo o desvio de fase causada por aplicar uma grade nominal fixa a um sinal de período diferente. O sucesso do procedimento depende da qualidade da estimativa de frequência, da precisão do interpolador e da compensação dos atrasos introduzidos pelos filtros anteriores à extração fasorial.

2.5.2 Interpolação B-Spline e estrutura de Farrow

As B-Splines são funções de base utilizadas em problemas de aproximação e interpolação, com propriedades de suavidade e suporte compacto. No contexto de estimação fasorial harmônica, a interpolação B-Spline é empregada para gerar amostras em instantes fracionários, permitindo a reamostragem dinâmica do sinal em função da frequência estimada (ALEIXO *et al.*, 2022).

Na forma cúbica, a interpolação pode ser implementada de maneira eficiente por uma estrutura de Farrow. Essa estrutura representa o interpolador como uma combinação polinomial de subfiltros fixos, forma usual para implementar atrasos fracionários variáveis sem alterar continuamente os coeficientes internos do filtro (ALEIXO *et al.*, 2022). Para um parâmetro fracionário α, a saída interpolada pode ser escrita de forma geral como

$$y[n,\alpha] = \sum_{p=0}^{P} H_p[n]\alpha^p, \tag{2.25}$$

em que $H_p[n]$ são as saídas dos subfiltros e P é a ordem do polinômio. No caso cúbico, $P = 3$. A principal vantagem dessa estrutura, já explorada em interpoladores para estimação fasorial, é que os coeficientes dos subfiltros permanecem fixos, enquanto o atraso fracionário é ajustado apenas pelo parâmetro α. Isso evita o reprojeto do filtro a cada nova frequência estimada (ALEIXO *et al.*, 2022).

Como a interpolação B-Spline cúbica não é naturalmente interpolante em todos os pontos amostrados, utiliza-se um pré-filtro para obter os coeficientes adequados da expansão. Esse pré-filtro compensa a suavização da base B-Spline e melhora a preservação das componentes do sinal. Para estimação harmônica, essa compensação é relevante porque pequenas alterações de amplitude e fase podem ser amplificadas nas componentes de ordem elevada.

2.5.3 Filtros Flat-Top

Filtros e janelas Flat-Top são projetados para apresentar resposta de magnitude muito plana em torno da frequência central. Essa propriedade é desejável quando o objetivo é estimar amplitudes com baixa sensibilidade a pequenos desvios de frequência (HARRIS, 1978; DUDA; ZIELIŃSKI; BARCZENTEWICZ, 2016). Em estimação de fasores harmônicos, o uso de filtros FIR Flat-Top é explorado para reduzir o erro de amplitude e preservar a informação de fase dentro da banda de interesse (DUDA *et al.*, 2020).

Um filtro Flat-Top pode ser construído a partir de uma combinação de cossenos, como

$$w[n] = \sum_{m=0}^{P} a_m \cos\left(\frac{2\pi mn}{N}\right), \tag{2.26}$$

seguida de normalização adequada. A partir do filtro protótipo, filtros centrados em diferentes frequências podem ser obtidos por modulação complexa. Quando essa estratégia é aplicada a várias componentes harmônicas, obtém-se um banco de filtros.

O uso direto de um filtro para cada componente pode exigir elevado número de operações, principalmente quando muitas ordens harmônicas são estimadas simultaneamente. A partir das propriedades dos filtros Flat-Top e da teoria de bancos polifásicos, a combinação entre essas técnicas torna-se atrativa: preserva as propriedades de estimação do filtro protótipo e reduz a carga computacional associada ao banco (DUDA *et al.*, 2020; MITRA; KUO, 2006).

A Figura 4 ilustra essa propriedade para o filtro protótipo utilizado neste trabalho. Na faixa de $\pm 1\,\mathrm{Hz}$ em torno da frequência central, o desvio de magnitude permanece inferior a $1 \times 10^{-6}\,\%$, e mesmo em $\pm 2\,\mathrm{Hz}$ não ultrapassa $5 \times 10^{-5}\,\%$. É essa planicidade

que torna a estimação de amplitude praticamente insensível a desvios de frequência dentro da faixa de interesse.

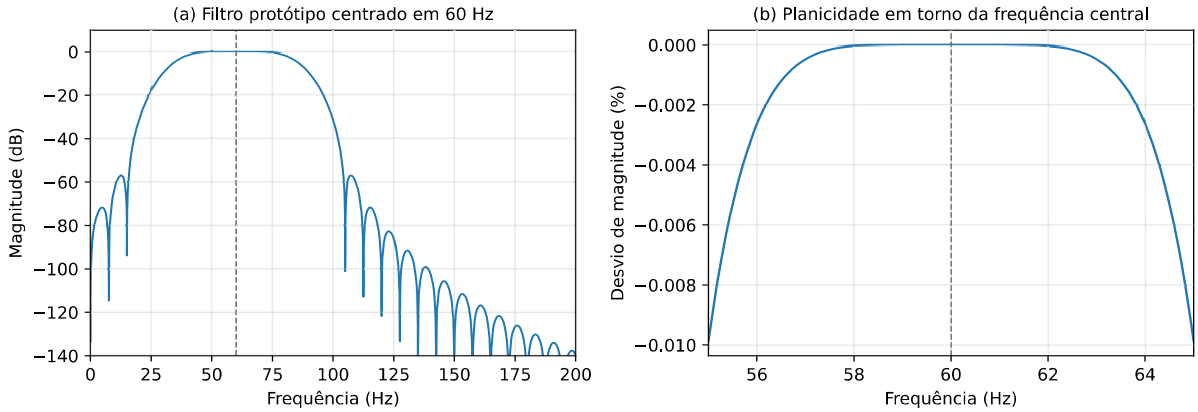


Figura 4 – Resposta em magnitude do filtro protótipo Flat-Top: (a) resposta completa do protótipo modulado para a frequência fundamental; (b) desvio de magnitude em torno da frequência central, evidenciando a planicidade.

2.5.4 Decomposição polifásica e bancos de filtros

A decomposição polifásica é uma ferramenta clássica de processamento multitaxa. Ela permite reescrever um filtro em componentes que operam em fases distintas da sequência de entrada, possibilitando implementações eficientes de decimadores, interpoladores e bancos de filtros (MITRA; KUO, 2006). Para um filtro $H(z)$, a decomposição polifásica do Tipo I é dada por

$$H(z) = \sum_{\ell=0}^{M-1} z^{-\ell} E_\ell(z^M), \tag{2.27}$$

em que M é o número de fases e $E_\ell(z)$ são os componentes polifásicos.

Em um banco de filtros uniforme, o filtro protótipo é deslocado em frequência para formar múltiplos canais. A implementação direta requer a filtragem individual de cada canal. Na forma polifásica, a filtragem é reorganizada em taxa reduzida e combinada por uma Transformada Discreta de Fourier Inversa (do inglês *Inverse Discrete Fourier Transform*) (IDFT), tipicamente calculada por um algoritmo de Transformada Inversa Rápida de Fourier (do inglês *Inverse Fast Fourier Transform*) (IFFT), o que reduz o número de operações. Para um filtro protótipo com N coeficientes e um banco com M canais, a complexidade aproximada pode ser reduzida de uma ordem proporcional a NM para uma ordem proporcional a

$$N + M \log_2(M). \tag{2.28}$$

Essa redução é um dos elementos centrais deste trabalho. Como a estimação de fasores harmônicos pode exigir dezenas de canais, a forma polifásica contribui para tornar

o método mais adequado a aplicações em tempo real e a implementações embarcadas. A metodologia proposta no Capítulo 3 utiliza essa estrutura para organizar o banco Flat-Top de forma computacionalmente eficiente.

A Figura 5 resume essa estrutura: a sequência de entrada é decimada por M e distribuída entre os componentes polifásicos $E_\ell(z^M)$ por uma linha de atrasos, e as saídas dos ramos são combinadas por uma IDFT de M pontos, que produz simultaneamente todos os canais do banco.

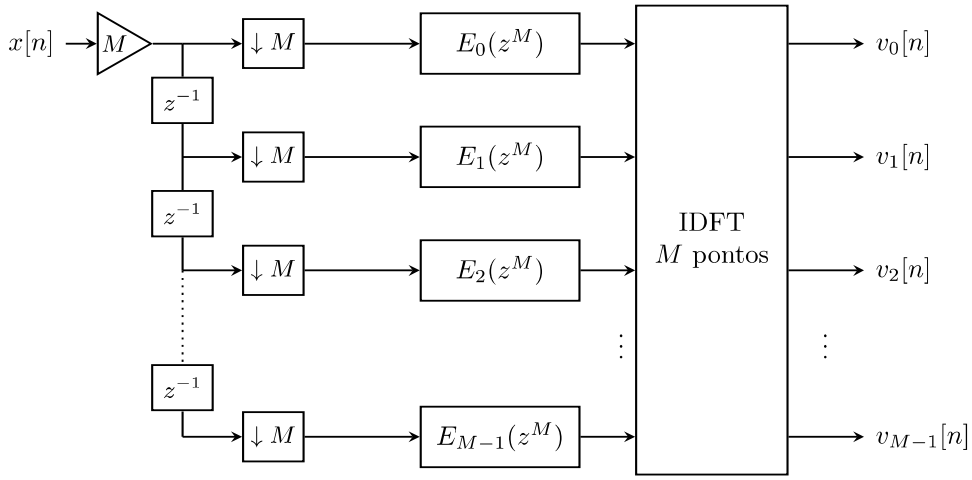


Figura 5 – Implementação polifásica de um banco de filtros uniforme: o ganho M na entrada precede a linha de atrasos, e cada ramo é decimado por M antes de alimentar o componente polifásico $E_\ell(z^M)$, cujas saídas são combinadas por uma IDFT de M pontos.

2.6 Conclusões Parciais

Este capítulo reuniu os fundamentos que sustentam a cadeia implementada nesta dissertação. A representação fasorial e sincrofasorial define o objeto de estimação; a discussão sobre fasores harmônicos evidencia a dificuldade central do problema, a amplificação do erro de fase com a ordem harmônica; a estimação fasorial e as métricas associadas estabelecem o problema a resolver e os critérios de avaliação; e a revisão dos trabalhos correlatos posiciona a proposta frente à literatura, mostrando que rastreamento de frequência, reamostragem e implementação embarcada costumam ser tratados separadamente.

As técnicas revisadas ao final do capítulo: reamostragem dinâmica, interpolação B-Spline por estrutura de Farrow, filtros Flat-Top e decomposição polifásica, são os blocos que, integrados, constituem essa cadeia. No capítulo seguinte, esses blocos são combinados em um único fluxo de processamento voltado à estimação de fasores harmônicos sob frequência variável, deslocando a ênfase da fundamentação individual de cada técnica para a forma como elas operam em conjunto e para as decisões que viabilizam sua execução em tempo real.

3 Metodologia

Este capítulo apresenta a metodologia proposta para estimação de fasores harmônicos em condições de frequência variável. A proposta é organizada como uma cadeia de processamento em tempo discreto, composta por estimação da frequência fundamental, reamostragem dinâmica por interpolação B-Spline e extração harmônica por banco de filtros Flat-Top em estrutura polifásica. O objetivo central é produzir uma sequência de amostras sincronizada com a frequência instantânea do sinal, de modo que o banco de filtros opere em uma base temporal mais adequada à estimação dos fasores harmônicos.

A metodologia descrita neste capítulo corresponde ao algoritmo que orienta a implementação em desenvolvimento. Assim, as equações e etapas apresentadas têm caráter metodológico: definem o processamento necessário, os sinais de entrada e saída de cada bloco e as métricas que serão utilizadas na avaliação posterior. Os resultados finais de desempenho e a validação completa em FPGA não são tratados neste capítulo.

3.1 Visão geral do método proposto

O método parte de um sinal discreto $x[n]$, obtido a partir da amostragem de um sinal elétrico de tensão ou corrente, que podem possuir frequência diferente da nominal. Em aplicações de estimação fasorial, esse desvio de frequência é crítico, pois a maior parte dos estimadores baseados em transformadas ou bancos de filtros assume uma relação bem definida entre frequência fundamental, janela de observação e pontos espectrais avaliados. Quando essa relação é perdida, ocorre desalinhamento temporal e espectral, o que afeta principalmente a fase dos fasores estimados e se torna mais severo em componentes harmônicas de ordem elevada.

Para contornar esse problema, a metodologia proposta emprega uma etapa de rastreamento de frequência por cruzamento por zero, seguida por uma etapa de reamostragem dinâmica. A frequência estimada alimenta um interpolador B-Spline implementado por estrutura de Farrow, que produz uma sequência reamostrada $x_i[k]$ com número de amostras por ciclo constante associado à frequência nominal. Em seguida, essa sequência alimenta um banco de filtros Flat-Top polifásico, responsável pela extração simultânea dos fasores da componente fundamental e das componentes harmônicas de interesse.

De forma resumida, a cadeia de processamento é composta pelas seguintes etapas:

- filtragem preliminar para aumentar a robustez da detecção de cruzamentos por zero;
- estimação da frequência fundamental por Zero-Crossing;
- suavização da frequência estimada por filtro de Kalman;
- alinhamento temporal entre sinal e frequência estimada;

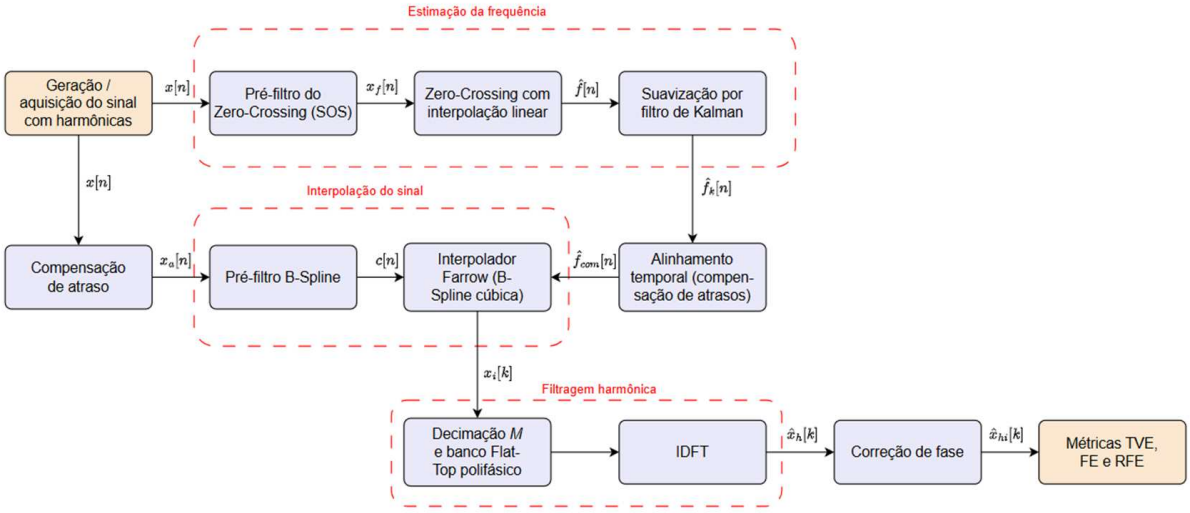


Figura 6 – Visão geral da cadeia de processamento completa do método proposto, do sinal de entrada às métricas de avaliação.

- reamostragem por interpolação B-Spline cúbica;
- extração harmônica por banco de filtros Flat-Top com decomposição polifásica;
- correção de fase e cálculo das métricas de avaliação.

3.2 Arquitetura funcional da metodologia

A Figura 6 resume graficamente essa cadeia de processamento. O sinal de entrada segue diretamente para o caminho de compensação de atraso, pré-filtragem B-Spline e interpolação Farrow. Em paralelo, a frequência estimada pelo método de Zero-Crossing é suavizada pelo filtro de Kalman e posteriormente compensada temporalmente. A frequência compensada $\hat{f}_{\text{comp}}[n]$ é utilizada no controle da reamostragem realizada pelo interpolador Farrow. O sinal de entrada $x[n]$, que possui componentes harmônicas, é inicialmente filtrado para aumentar a robustes da detecção dos cruzamentos por zero, dos quais se obtém uma estimativa da frequência fundamental $\hat{f}[n]$. Essa estimativa é suavizada pelo filtro de Kalman, produzindo $\hat{f}_s[n]$, que controla a reamostragem dinâmica após a compensação dos atrasos introduzidos pelos filtros (alinhamento temporal). A sequência reamostrada $x_i[k]$ alimenta o banco de filtros Flat-Top polifásico, que extrai os fasores harmônicos; por fim, aplica-se a correção de fase e calculam-se as métricas de avaliação (TVE, Erro de Frequência (do inglês *Frequency Error*) (FE) e Erro da Taxa de Variação da Frequência (do inglês *Rate of Change of Frequency Error*) (RFE)).

A escolha desses blocos está relacionada ao compromisso entre precisão, robustez e custo computacional. O estimador por cruzamento por zero é simples e adequado a operação em tempo real, desde que precedido de filtragem apropriada. A interpolação B-Spline permite deslocar o problema do reprojeto contínuo de filtros para uma etapa

de reamostragem controlada pela frequência estimada (ALEIXO *et al.*, 2022). Por fim, a decomposição polifásica reduz o esforço necessário para implementar o banco de filtros, preservando a estrutura de extração harmônica baseada em filtros Flat-Top (DUDA *et al.*, 2020; MITRA; KUO, 2006).

A metodologia pode ser organizada em dois níveis funcionais: sincronização temporal por estimação de frequência e reamostragem, e extração dos fasores harmônicos. Essa organização não representa uma arquitetura final de hardware, mas uma descrição algorítmica dos blocos necessários para que o estimador opere com sinais cuja frequência fundamental pode se afastar da nominal.

No primeiro nível, encontram-se as etapas responsáveis por estimar a frequência fundamental e ajustar a base temporal do sinal. Inicialmente, o sinal de entrada passa por um pré-filtro associado ao Zero-Crossing, com o objetivo de atenuar componentes harmônicas e ruído antes da detecção dos cruzamentos por zero. Em seguida, o instante de cruzamento é refinado por interpolação linear, reduzindo o erro causado pela amostragem discreta. A estimativa é atualizada a cada meio ciclo, o que melhora o acompanhamento de variações de frequência em ensaios dinâmicos, como variação em rampa.

A frequência obtida pelo Zero-Crossing é suavizada por um filtro de Kalman e alinhada temporalmente com o trecho do sinal ao qual ela se refere. Esse alinhamento é necessário porque o pré-filtro do Zero-Crossing, o filtro de Kalman e o pré-filtro B-Spline introduzem atrasos. Se esses atrasos não forem compensados, o interpolador pode usar uma frequência estimada para um trecho diferente daquele que está sendo reamostrado, produzindo erro de fase. Após a compensação, o sinal é processado pelo pré-filtro B-Spline e pela estrutura de Farrow, gerando a sequência reamostrada $x_i[k]$.

No segundo nível, a sequência reamostrada alimenta o banco de filtros harmônicos. O filtro Flat-Top é usado como protótipo por apresentar baixa sensibilidade de magnitude a pequenos desvios em torno da frequência central. Para reduzir o custo computacional, seus coeficientes são reorganizados por decomposição polifásica, e as saídas dos ramos são combinadas por uma IFFT para produzir simultaneamente os canais associados às componentes de interesse. A partir desses canais, obtêm-se os fasores harmônicos estimados.

As métricas de avaliação são calculadas apenas após a extração fasorial. Para cada componente, os fasores estimados serão comparados com fasores de referência por meio de TVE, erro de magnitude, erro de fase, erro de frequência e erro de ROCOF. Dessa forma, a metodologia separa claramente as etapas de sincronização e reamostragem, extração harmônica e avaliação, o que facilita a análise posterior dos erros introduzidos por cada bloco. Os sinais utilizados nessa avaliação são definidos no Capítulo 5.

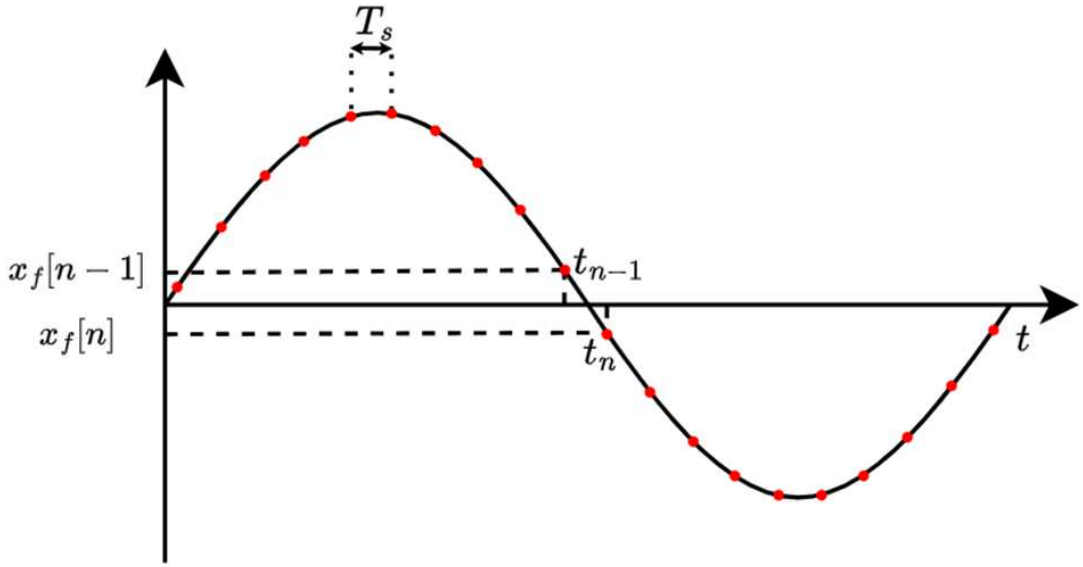


Figura 7 – Interpolação linear do instante de cruzamento por zero: a reta que liga as amostras adjacentes $x_f[n-1]$ e $x_f[n]$, nos instantes t_{n-1} e t_n, cruza o eixo no instante estimado $\hat{t}_{zc}$.

3.2.1 Estimação de frequência por Zero-Crossing

A primeira etapa dinâmica do método é a estimação da frequência fundamental. Para isso, emprega-se um detector de cruzamento por zero, escolhido por sua simplicidade e por sua baixa demanda computacional (RIBEIRO *et al.*, 2013). Antes da detecção, o sinal é processado por um filtro voltado à atenuação de harmônicos, inter-harmônicos e ruído. Essa etapa é importante porque componentes de alta frequência podem introduzir cruzamentos indesejados, fazendo com que o período estimado deixe de representar a componente fundamental.

Considere duas amostras consecutivas do sinal filtrado, $x_f[n-1]$ e $x_f[n]$, entre as quais ocorre mudança de sinal. O instante de cruzamento pode ser refinado por interpolação linear. Se t_{n-1} é o instante da amostra anterior, uma estimativa do instante de cruzamento é dada por

$$\hat{t}_{zc} = t_{n-1} + T_s \frac{x_f[n-1]}{x_f[n-1] - x_f[n]}. \tag{3.1}$$

O termo fracionário da Equação 3.1 reduz o erro de quantização temporal que ocorreria caso o cruzamento fosse associado diretamente ao índice da amostra. Esse refinamento é especialmente importante quando a frequência estimada será usada para controlar a reamostragem, pois pequenos erros de tempo podem se acumular na fase das componentes harmônicas. A Figura 7 ilustra esse refinamento: a reta que liga as amostras adjacentes $x_f[n-1]$ e $x_f[n]$, nos instantes t_{n-1} e t_n, cruza o eixo no instante estimado $\hat{t}_{zc}$, obtido pela semelhança de triângulos sobre essa reta.

No algoritmo adotado, a frequência é estimada a partir do intervalo entre cruzamentos separados por meio ciclo. Essa escolha permite atualizar a estimativa com maior frequência do que uma estratégia baseada no ciclo completo, o que é útil em cenários de rampa, nos quais a frequência varia durante o próprio ciclo. Se $\Delta\hat{t}_{zc}$ representa o intervalo estimado entre dois cruzamentos consecutivos de polaridade oposta, a frequência estimada é:

$$\hat{f}_{zc}[n] = \frac{1}{2\Delta\hat{t}_{zc}}. \tag{3.2}$$

A Equação 3.2 mostra que o estimador depende diretamente da precisão dos instantes de cruzamento. Por isso, a etapa de filtragem anterior ao Zero-Crossing e a interpolação linear do cruzamento são tratadas como partes essenciais do método, e não apenas como detalhes de implementação.

3.2.2 Suavização da frequência por filtro de Kalman

A sequência $\hat{f}_{zc}[n]$ obtida pelo detector de cruzamento por zero pode apresentar oscilações decorrentes de ruído, distorções residuais e erros numéricos. Como essa frequência é usada para controlar a reamostragem, suas variações bruscas se propagam diretamente para o parâmetro de interpolação. Por isso, é necessário suavizá-la antes de utilizá-la. Esta seção descreve a estratégia de suavização efetivamente adotada na implementação que consiste na utilização de um filtro de Kalman e registra, por completude, as abordagens anteriores baseadas em média móvel que foram avaliadas e descartadas.

3.2.2.1 Tentativas iniciais baseadas em média móvel

A primeira abordagem testada foi a utilização de um filtro de média móvel de comprimento N_w sobre a frequência estimada. As equações descrevem o filtro nas suas formas não recurssiva e recurssiva, respectivamente.

$$\hat{f}[n] = \frac{1}{N_w} \sum_{i=0}^{N_w-1} \hat{f}_{zc}[n-i]. \tag{3.3}$$

$$\hat{f}[n] = \hat{f}[n-1] + \frac{1}{N_w}\left(\hat{f}_{zc}[n] - \hat{f}_{zc}[n-N_w]\right). \tag{3.4}$$

Conceitualmente, o filtro média móvel é capaz de suavizar a frequência e reduz variações bruscas no parâmetro de reamostragem. Entretanto, sua realização embarcada apresentou dois problemas práticos relevantes.

Na **forma não recursiva**, mostrada em (3.3) a média era calculada mantendo-se as últimas N_w amostras em um *buffer* e percorrendo-se todo o *buffer* a cada novo instante de amostragem. Como N_w é da ordem do número de pontos por ciclo, essa varredura repetida

representava um custo de processamento elevado para o processador embarcado (SAPHO), comprometendo a operação em tempo real e tornando essa abordagem inadequada.

Na **forma recursiva**, mostrada em (3.4) a mesma média móvel é escrita como uma equação de diferenças que evita a soma completa a cada nova amostra, atualizando-se a saída anterior apenas com a amostra que entra e a que sai da janela. Essa forma, porém, apresenta um polo discreto sobre $z = 1$, idealmente cancelado por um zero na mesma posição. Combinada à quantização dos coeficientes e à aritmética de precisão limitada da implementação no processador embarcado, a imperfeição desse cancelamento polo-zero tornou o filtro **numericamente instável em determinados momentos**, produzindo divergências na frequência suavizada. Por esses motivos, a média móvel foi descartada como bloco de suavização.

3.2.2.2 Filtro de Kalman adotado

A solução adotada substitui a média móvel por um **filtro de Kalman** com vetor de estado bidimensional, escrito em forma recursiva e sem necessidade de *buffer*. O estado reúne a própria frequência e a sua taxa de variação,

$$\mathbf{x}[n] = \begin{bmatrix} f[n] \\ \dot{f}[n] \end{bmatrix} = \begin{bmatrix} f_0 + R_f \cdot T_s \cdot n \\ R_f \end{bmatrix}, \tag{3.5}$$

com modelo de transição de velocidade constante e medição dada pela frequência estimada pelo método de Zero-Crossing,

$$\mathbf{x}[n] = \mathbf{F} \cdot \mathbf{x}[k-1] + \mathbf{w}[n], \qquad z[n] = \mathbf{H} \cdot \mathbf{x}[n] + v[n], \tag{3.6}$$

em que

$$\mathbf{F} = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix}, \qquad \mathbf{H} = \begin{bmatrix} 1 & 0 \end{bmatrix}, \qquad z[k] = \hat{f}_{zc}[n], \tag{3.7}$$

sendo T_s o intervalo entre atualizações do filtro, correspondente a meio ciclo nominal da rede, uma vez que o método de Zero-Crossing produz uma nova medição a cada cruzamento por zero. O ruído de processo $\mathbf{w}[n]$ tem covariância $\mathbf{Q}$ e o ruído de medição $v[n]$ tem variância R,

$$\mathbf{Q} = q \begin{bmatrix} T_s^3/3 & T_s^2/2 \\ T_s^2/2 & T_s \end{bmatrix}, \qquad R = r, \tag{3.8}$$

em que q quantifica o desvio admitido em relação ao modelo de velocidade constante entre amostras e r representa a variância da medição. A cada nova medição, o filtro executa uma etapa de *predição*,

$$\hat{\mathbf{x}}^{-}[n] = \mathbf{F} \cdot \hat{\mathbf{x}}[n-1], \qquad \mathbf{P}^{-}[n] = \mathbf{F} \cdot \mathbf{P}[n-1]\,\mathbf{F}^{\top} + \mathbf{Q}, \tag{3.9}$$

seguida de uma etapa de *correção* a partir da inovação $e[n]$, de sua variância $S[n]$ e do ganho $\mathbf{K}[n]$,

$$e[n] = z[n] - \mathbf{H}\,\hat{\mathbf{x}}^{-}[n], \qquad S[n] = \mathbf{H}\,\mathbf{P}^{-}[n]\,\mathbf{H}^{\top} + R, \tag{3.10}$$

$$\mathbf{K}[n] = \mathbf{P}^{-}[n]\,\mathbf{H}^{\top} S^{-1}[n], \tag{3.11}$$

$$\hat{\mathbf{x}}[n] = \hat{\mathbf{x}}^{-}[n] + \mathbf{K}[k]\,e[n], \qquad \mathbf{P}[n] = (\mathbf{I} - \mathbf{n}[n]\,\mathbf{H})\,\mathbf{P}^{-}[n]. \tag{3.12}$$

A frequência suavizada utilizada pelos blocos seguintes é a primeira componente do estado corrigido, $\hat{f}[n] = \left[\hat{\mathbf{x}}[n]\right]_1$. Cabe destacar que, embora o estado contenha a taxa de variação $\dot{f}[n]$, essa componente existe apenas para compor o modelo de transição das equações (3.6) e (3.7).

Por operar de forma recursiva, com um número fixo e pequeno de operações por medição, o filtro de Kalman elimina a varredura de *buffer* da forma não recursiva da média móvel e não sofre da instabilidade observada em sua forma recursiva, sendo portanto adequado à implementação embarcada. Quanto ao atraso, o filtro não adiciona latência estrutural de janela: a estimativa é atualizada a cada nova medição do Zero-Crossing, isto é, a cada meio ciclo, e a defasagem residual introduzida pela dinâmica do filtro, regulada pelo par (q, r), mostrou-se desprezível para a sintonia adotada. O atraso do pré-filtro B-Spline, por sua vez, é compensado posteriormente. Cabe ainda observar que as primeiras estimativas, produzidas durante o transitório de convergência do filtro a partir das condições iniciais $\hat{\mathbf{x}}[0]$ e $\mathbf{P}[0]$, são descartadas por não serem representativas do regime permanente. O detalhamento da realização escalar desse filtro no processador, incluindo o tratamento da aritmética de precisão limitada, é apresentado no Capítulo 4.

3.2.3 Reamostragem por interpolação B-Spline

Após a estimação da frequência, o sinal é reamostrado para reduzir o desalinhamento entre a frequência real do sinal e a base de estimação do banco de filtros. A ideia é produzir uma sequência $x_i[k]$ que preserve o número nominal de pontos por ciclo associado a f_0, mesmo quando a frequência real se desvia da nominal. Para isso, define-se a razão instantânea de reamostragem

$$\lambda[n] = \frac{f_0}{\hat{f}[n]}. \tag{3.13}$$

Quando $\hat{f}[n] > f_0$, a razão $\lambda[n]$ torna-se menor que um; quando $\hat{f}[n] < f_0$, torna-se maior que um. Dessa forma, a taxa efetiva de geração das amostras interpoladas acompanha a frequência estimada, evitando que o banco de filtros opere sobre uma sequência temporal incompatível com sua base nominal.

O interpolador utilizado é baseado em B-Splines cúbicas com pré-filtragem, conforme discutido em (ALEIXO *et al.*, 2022). A interpolação B-Spline possui boas propriedades de

suavidade, mas a forma cúbica básica não passa necessariamente pelos valores originais das amostras. Por isso, emprega-se um pré-filtro que transforma a sequência de amostras em coeficientes adequados para a reconstrução interpolante.

No caso cúbico, o pré-filtro pode ser representado por uma função de transferência racional,

$$H_{pre}(z) = \frac{B(z)}{A(z)}, \tag{3.14}$$

de modo que a sequência pré-filtrada $c[n]$ seja obtida por

$$c[n] = H_{pre}(z) \cdot \{x[n]\}. \tag{3.15}$$

Na implementação adotada, os coeficientes desse pré-filtro são calculados a partir do polo associado à B-Spline cúbica, frequentemente escrito como

$$s = \sqrt{3} - 2. \tag{3.16}$$

O pré-filtro compensa a suavização inerente à base B-Spline e melhora a fidelidade da amostra interpolada. Essa etapa é particularmente relevante para harmônicos de ordem elevada, pois a atenuação ou o atraso de fase introduzidos na reconstrução temporal podem produzir erros crescentes com a ordem harmônica.

3.2.3.1 Estrutura de Farrow

A geração de amostras em instantes fracionários é realizada por uma estrutura de Farrow. Essa estrutura implementa um filtro de atraso fracionário variável como um polinômio em função do parâmetro fracionário α. Para a interpolação cúbica utilizada neste trabalho, a amostra interpolada pode ser escrita como:

$$x_i[k] = H_0[n]\alpha^3 + H_1[n]\alpha^2 + H_2[n]\alpha + H_3[n], \tag{3.17}$$

em que $\alpha \in [0,1)$ indica a posição fracionária da amostra desejada em relação às amostras disponíveis, e $H_0[n]$, $H_1[n]$, $H_2[n]$ e $H_3[n]$ são combinações lineares de quatro amostras consecutivas do sinal pré-filtrado. Considerando o vetor local

$$\mathbf{c}_n = \begin{bmatrix} c[n-3] & c[n-2] & c[n-1] & c[n] \end{bmatrix}^T, \tag{3.18}$$

os subfiltros cúbicos podem ser expressos como:

$$H_0[n] = -\frac{1}{6}c[n-3] + \frac{1}{2}c[n-2] - \frac{1}{2}c[n-1] + \frac{1}{6}c[n], \tag{3.19}$$

$$H_1[n] = \frac{1}{2}c[n-3] - c[n-2] + \frac{1}{2}c[n-1], \tag{3.20}$$

$$H_2[n] = -\frac{1}{2}c[n-3] + \frac{1}{2}c[n-1], \tag{3.21}$$

$$H_3[n] = \frac{1}{6}c[n-3] + \frac{2}{3}c[n-2] + \frac{1}{6}c[n-1]. \tag{3.22}$$

O parâmetro α é atualizado de acordo com a razão $\lambda[n]$. Enquanto $\alpha < 1$, novas amostras interpoladas são geradas. Quando α ultrapassa a unidade, subtrai-se um período de amostra para reposicionar o acumulador fracionário:

$$\alpha \leftarrow \alpha + \lambda[n] \tag{3.23}$$

$$\alpha \leftarrow \alpha - 1 \quad \text{quando } \alpha \geq 1. \tag{3.24}$$

A Figura 8 ilustra essa ideia de forma simples. Os círculos mostrados ao longo da linha do gráfico são as amostras originais $x_0[\cdot]$, disponíveis somente nos instantes múltiplos do período de amostragem; a curva cheia é a reconstrução contínua obtida a partir delas; e o quadrado é a amostra interpolada $x[l]$, avaliada a uma distância fracionária α da amostra $x_0[k]$. Adaptada de (ALEIXO *et al.*, 2022). As amostras originais continuam onde estão, com seus instantes inalterados; o interpolador apenas lê o valor da curva reconstruída no instante desejado, que cai entre duas amostras vizinhas. Quem escolhe esse instante é o acumulador α, comandado por $\lambda[n]$.

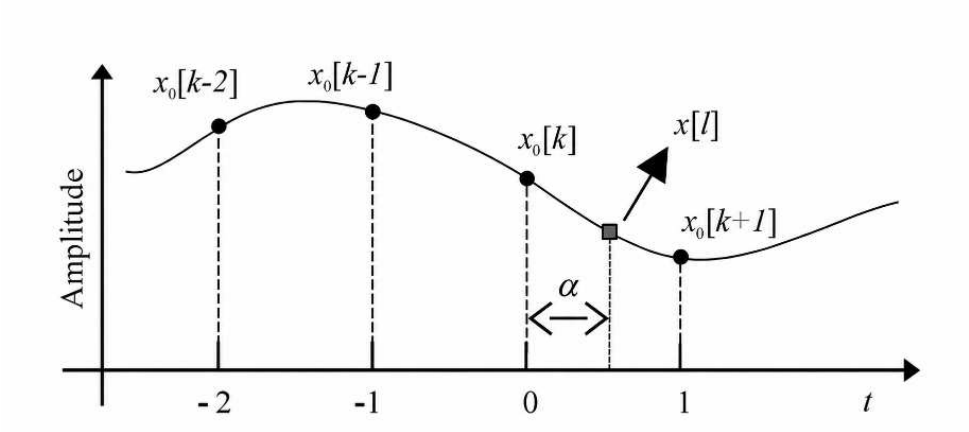


Figura 8 – Obtenção de uma amostra em instante fracionário pelo Bspline.

Os dois casos possíveis se resumem ao passo desse acumulador. Quando $\hat{f}[n] > f_0$, tem-se $\lambda[n] < 1$ e o passo da sequência reamostrada fica menor que o original: o acumulador

avança pouco a cada saída e pontos interpolados são gerados dentro do mesmo intervalo entre duas amostras consecutivas, exatamente a situação desenhada na figura. Quando $\hat{f}[n] < f_0$, tem-se $\lambda[n] > 1$ e o passo fica maior que o original: os instantes desejados ficam mais espaçados, o acumulador ultrapassa a unidade, o índice avança para a próxima amostra original e a interpolação é feita na fração que sobra. Em ambos os casos o efeito é o mesmo: manter o número de pontos por ciclo próximo do valor nominal associado a f_0.

Essa forma de implementação evita o recálculo dos coeficientes do filtro para cada novo atraso fracionário. Os coeficientes dos subfiltros permanecem fixos, e apenas as potências de α mudam ao longo do tempo. Essa característica torna a estrutura adequada para aplicações embarcadas e para implementação em arquiteturas de processamento com recursos limitados.

A Figura 9 ilustra a estrutura de Farrow para interpolação cúbica, mostrando os blocos de pré-filtragem, os subfiltros e a combinação polinomial controlada por α.

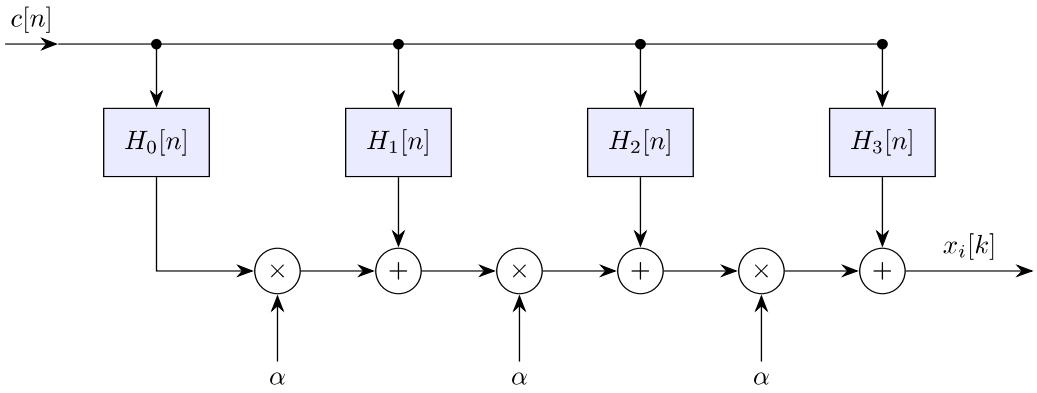


Figura 9 – Estrutura de Farrow para interpolação cúbica.

3.2.4 Banco de filtros Flat-Top

Após a reamostragem, a sequência $x_i[k]$ alimenta o banco de filtros responsável pela estimação dos fasores harmônicos. O filtro protótipo adotado é do tipo Flat-Top, cuja principal característica é apresentar resposta de magnitude bastante plana em torno da frequência central. Essa propriedade é desejável para estimação de amplitude, pois reduz a sensibilidade a pequenos desvios de frequência dentro da banda de passagem (DUDA; ZIELIŃSKI; BARCZENTEWICZ, 2016; DUDA *et al.*, 2020).

O filtro protótipo pode ser definido por uma janela cossenoidal Flat-Top normalizada. Uma forma geral para seus coeficientes é

$$w[n] = \frac{1}{C} \sum_{m=0}^{P} a_m \cos\left(\frac{2\pi m n}{N}\right), \tag{3.25}$$

em que a_m são os coeficientes da janela, N é o comprimento do filtro, P é a ordem da combinação cossenoidal e C é um fator de normalização escolhido para que o ganho na

frequencia zero seja unitário. A partir desse filtro base, o filtro associado à componente harmônica de ordem h é obtido por modulação complexa:

$$h_h[n] = w[n]e^{j2\pi h f_0 n/F_s}. \tag{3.26}$$

A saída do filtro associado à ordem h é então

$$X_h[k] = \sum_{n=0}^{N-1} h_h[n]x_i[k-n]. \tag{3.27}$$

A implementação direta da Equação (3.27) para muitas ordens harmônicas pode se tornar custosa. Se o banco possuir M canais e o filtro protótipo possuir N coeficientes, a forma direta exige uma quantidade de operações proporcional a NM para produzir uma saída completa do banco. Por esse motivo, utiliza-se a estrutura polifásica.

3.2.5 Decomposição polifásica

A decomposição polifásica permite particionar o filtro protótipo em M componentes, cada um operando em uma taxa reduzida. Para um filtro $H(z)$, sua decomposição polifásica do Tipo I pode ser escrita como (MITRA; KUO, 2006)

$$H(z) = \sum_{\ell=0}^{M-1} z^{-\ell} E_\ell(z^M), \tag{3.28}$$

em que

$$E_\ell(z) = \sum_{r=-\infty}^{\infty} h[rM+\ell]z^{-r}, \qquad 0 \leq \ell \leq M-1. \tag{3.29}$$

A decomposição polifásica reorganiza a resposta ao impulso do filtro protótipo em M subfiltros (componentes polifásicas), cada um operando sobre uma subsequência decimada do sinal. Quando aplicada a um banco de filtros DFT uniforme, no qual os filtros dos canais são versões moduladas (deslocadas em frequência) de um mesmo protótipo, as saídas das componentes polifásicas podem ser combinadas por uma IDFT, produzindo simultaneamente as saídas de todos os canais do banco. Em termos práticos, a estrutura polifásica desloca a filtragem para a taxa decimada e substitui a implementação direta dos diversos filtros modulados por uma combinação entre filtragem polifásica e IFFT.

Para um banco com M canais e filtro protótipo de comprimento N, a complexidade aproximada passa de NM multiplicações, na forma direta, para

$$N + M\log_2(M), \tag{3.30}$$

quando a etapa de modulação é realizada por uma IFFT. Essa redução é uma das principais motivações para o uso da decomposição polifásica no presente trabalho, pois a estimação simultânea de vários fasores harmônicos exige um banco com muitos canais.

A decomposição polifásica admite mais de um caso de implementação, conforme a relação escolhida entre o fator de decimação M, o comprimento do filtro protótipo e o número de canais do banco (LUIZ, 2024; VICTOR, 2024). Neste trabalho, adota-se o caso particular em que o fator de decimação é igual ao número de pontos por ciclo,

$$M = N_{ppc} = \frac{F_s}{f_0}, \tag{3.31}$$

que corresponde a um banco de filtros DFT uniforme *maximamente decimado*. A motivação dessa escolha é estrutural: ao igualar o número de ramos polifásicos ao fator de decimação, os *bins* da IDFT de tamanho M ficam espaçados de $F_s/M = f_0$, de modo que cada bin recai exatamente sobre uma componente harmônica hf_0. Esse caso também reduz a quantidade de amostras processadas e o número de cálculos: como o banco produz uma saída a cada M amostras de entrada, a etapa custosa, filtragem polifásica seguida da IDFT, é executada uma única vez a cada M amostras, e não a cada amostra, como ocorreria em um filtro operando à taxa cheia.

Uma consequência direta dessa decimação é a taxa efetiva de saída do banco. Sendo a taxa de amostragem $F_s = f_0 N_{ppc}$, a saída ocorre a $F_s/M = f_0$ fasores por segundo, ou seja, um fasor por ciclo. A norma IEC/IEEE 60255-118-1 estabelece que a taxa de reporte de uma PMU deve ser um submúltiplo ou múltiplo da frequência nominal, listando para sistemas de 60 Hz as taxas de 10, 12, 15, 20, 30, 60 e 120 quadros por segundo, ou saídas do algoritmo de estimação (IEC, 2018). A taxa de 60 quadros por segundo obtida com $M = N_{ppc}$ é, portanto, compatível com a norma. Vale ressaltar que a norma não impõe esse valor específico de M: outros fatores de decimação (por exemplo, $2N_{ppc}$ resultando em 30 quadros por segundo) também atenderiam às taxas previstas. A escolha $M = N_{ppc}$ decorre da estrutura do banco maximamente decimado descrita acima, e a compatibilidade com a taxa de reporte normativa é uma consequência conveniente, e não a causa.

3.2.6 Correção de fase

Mesmo após a reamostragem, a saída do banco de filtros necessita de correção de fase associada ao desvio acumulado entre a frequência estimada e a frequência nominal. Seja

$$\Delta f[k] = \hat{f}[k] - f_0. \tag{3.32}$$

Uma aproximação discreta da fase acumulada devido ao desvio de frequência pode ser obtida por integração numérica. Utilizando a regra trapezoidal, define-se

$$\Phi[k] = \Phi[k-1] + \pi\left(\Delta f[k] + \Delta f[k-1]\right) T_r, \tag{3.33}$$

em que T_r é o período efetivo entre saídas do banco após a decimação. Para a componente harmônica de ordem h, a correção é proporcional à ordem harmônica:

$$\Phi_h[k] = h\Phi[k]. \tag{3.34}$$

Assim, o fasor corrigido é calculado a partir da magnitude estimada e da fase ajustada,

$$\hat{X}_h[k] = |\hat{X}_h[k]| e^{j(\angle \hat{X}_h[k] + \Phi_h[k])}. \tag{3.35}$$

Essa etapa é importante porque pequenos desvios de fase na fundamental são amplificados nas ordens harmônicas. Portanto, erros de alinhamento temporal, quantização da frequência ou atrasos não compensados tendem a aparecer de forma mais evidente nos fasores harmônicos de maior ordem.

3.2.7 Métricas para avaliação

A avaliação do método será baseada em métricas usuais de estimação fasorial. A principal delas é o Erro Vetorial Total, ou *Total Vector Error* (TVE), definido como

$$TVE_h[k] = 100\sqrt{\frac{\left(\mathrm{Re}\{\hat{X}_h[k]\} - \mathrm{Re}\{X_h[k]\}\right)^2 + \left(\mathrm{Im}\{\hat{X}_h[k]\} - \mathrm{Im}\{X_h[k]\}\right)^2}{\left(\mathrm{Re}\{X_h[k]\}\right)^2 + \left(\mathrm{Im}\{X_h[k]\}\right)^2}}, \tag{3.36}$$

em que $\hat{X}_h[k]$ é o fasor estimado e $X_h[k]$ é o fasor de referência da componente de ordem h. O TVE combina erro de magnitude e erro de fase em uma única medida vetorial, o que o torna adequado para comparar o fasor estimado com a referência.

Também são considerados o erro de magnitude e o erro de fase, definidos por

$$\varepsilon_{A,h}[k] = |\hat{X}_h[k]| - |X_h[k]|, \tag{3.37}$$

$$\varepsilon_{\varphi,h}[k] = \angle \hat{X}_h[k] - \angle X_h[k]. \tag{3.38}$$

Essas duas métricas permitem interpretar a origem do TVE. Em sinais com variação de frequência, por exemplo, é comum que erros de alinhamento temporal se manifestem principalmente como erro de fase, e que esse efeito cresça com a ordem harmônica. Por fim, quando a frequência estimada também é avaliada, pode-se utilizar o erro de frequência

$$FE[k] = \hat{f}[k] - f[k], \tag{3.39}$$

e o erro de taxa de variação da frequência, associado ao ROCOF,

$$RFE[k] = \widehat{ROCOF}[k] - ROCOF[k]. \tag{3.40}$$

No contexto deste trabalho, essas métricas serão utilizadas para organizar a validação posterior do método em diferentes cenários de teste. Como os resultados finais ainda estão em desenvolvimento, este capítulo limita-se a estabelecer as definições e o encadeamento metodológico necessário para a avaliação.

3.3 Conclusões Parciais

Este capítulo apresentou a metodologia proposta para estimação de fasores harmônicos, com vistas à operação em tempo real em plataforma embarcada. O método combina rastreamento da frequência fundamental por Zero-Crossing, suavização dessa estimativa por filtro de Kalman, reamostragem dinâmica por interpolação B-Spline em estrutura de Farrow e extração harmônica por banco de filtros Flat-Top em estrutura polifásica combinada com uma IDFT. A ideia central é adaptar a base temporal do sinal antes da análise espectral, reduzindo o impacto de desvios da frequência fundamental sobre os fasores harmônicos estimados. Essa estratégia responde diretamente ao problema enunciado no início do capítulo, no qual o desalinhamento temporal se manifesta sobretudo como erro de fase e se agrava nas componentes harmônicas de ordem elevada.

Duas decisões de projeto merecem destaque. A primeira é a substituição da suavização por média móvel pelo filtro de Kalman: as formas não recursiva e recursiva da média móvel mostraram-se, respectivamente, custosa em varredura de *buffer* e numericamente instável na aritmética de precisão limitada do processador embarcado, ao passo que o filtro de Kalman opera de forma recursiva, com número fixo e reduzido de operações por medição e sem latência estrutural de janela. A segunda é a etapa de correção de fase proporcional à ordem harmônica, que compensa o desvio acumulado entre a frequência estimada e a nominal e é justamente o mecanismo que contém a amplificação do erro de fase nos harmônicos de maior ordem.

As etapas descritas também estabelecem a ponte entre o modelo em software e a implementação embarcada em desenvolvimento. A descrição foi mantida no nível metodológico, sem assumir validação final em FPGA ou resultados conclusivos. A fundamentação teórica dos blocos empregados foi revisada em parte no capítulo anterior e complementada ao longo deste capítulo, enquanto a análise de resultados será incorporada posteriormente, após a consolidação dos ensaios e a definição dos cenários de teste apresentados no Capítulo 5.

4 Implementação e Validação na Plataforma SAPHO

O Capítulo 3 descreveu a metodologia no nível algorítmico. Este capítulo trata da **implementação** dessa cadeia na plataforma SAPHO, que é o foco principal do trabalho. O algoritmo é primeiro prototipado em Python, constituindo um modelo de referência que funciona como *gêmeo* do sistema embarcado, e, em seguida, portado para o SAPHO, que gera os processadores *soft-core* descritos em Verilog. A validação apresentada neste trabalho é feita por **simulação** do Verilog gerado; a síntese e o embarque em FPGA não fazem parte do escopo e ficam como trabalho futuro. O encadeamento dos blocos segue uma estrutura de produtor–consumidor conectados por filas FIFO, conforme detalhado a seguir.

4.1 Parâmetros do sistema implementado

Antes de detalhar cada bloco, a Tabela 1 reúne os parâmetros numéricos adotados na implementação, que contextualizam as decisões descritas nas seções seguintes.

Tabela 1 – Parâmetros do sistema implementado.

Parâmetro	**Valor**
Frequência nominal f_0	60 Hz
Frequência de amostragem F_s	15 360 Hz
Pontos por ciclo N_{ppc}	256
Fator de decimação M	256 (igual a N_{ppc})
Filtro protótipo Flat-Top	$8N_{ppc} + 1$ coeficientes, organizados em 8 fases de 256
IDFT	256 pontos
Componentes extraídas	51 ordens ($h = 0$ a 50), partes real e imaginária
Amplitude das harmônicas	5 % da fundamental
Aritmética do núcleo SAPHO	ponto flutuante de 32 bits, mantissa de 23 bits e expoente de 8 bits
Escalas de fronteira	entrada dividida por 32 768; saídas multiplicadas por 10^6

A escolha $M = N_{ppc}$ é o que conecta diretamente o interpolador ao banco: como o interpolador entrega exatamente N_{ppc} amostras por ciclo da fundamental, o banco processa um bloco a cada ciclo, e o canal de índice h da IDFT corresponde à harmônica de ordem h.

As escalas de fronteira merecem registro porque a comunicação entre os processadores e o testbench é feita em inteiros de 32 bits: o sinal de entrada é normalizado por 32 768 e as saídas são multiplicadas por 10^6 antes de serem escritas. Esse fator define a resolução efetiva com que cada grandeza atravessa a fronteira entre os blocos.

4.2 Visão geral da arquitetura de implementação

A cadeia de processamento foi particionada em dois processadores *soft-core* como ilustra a Figura 10:

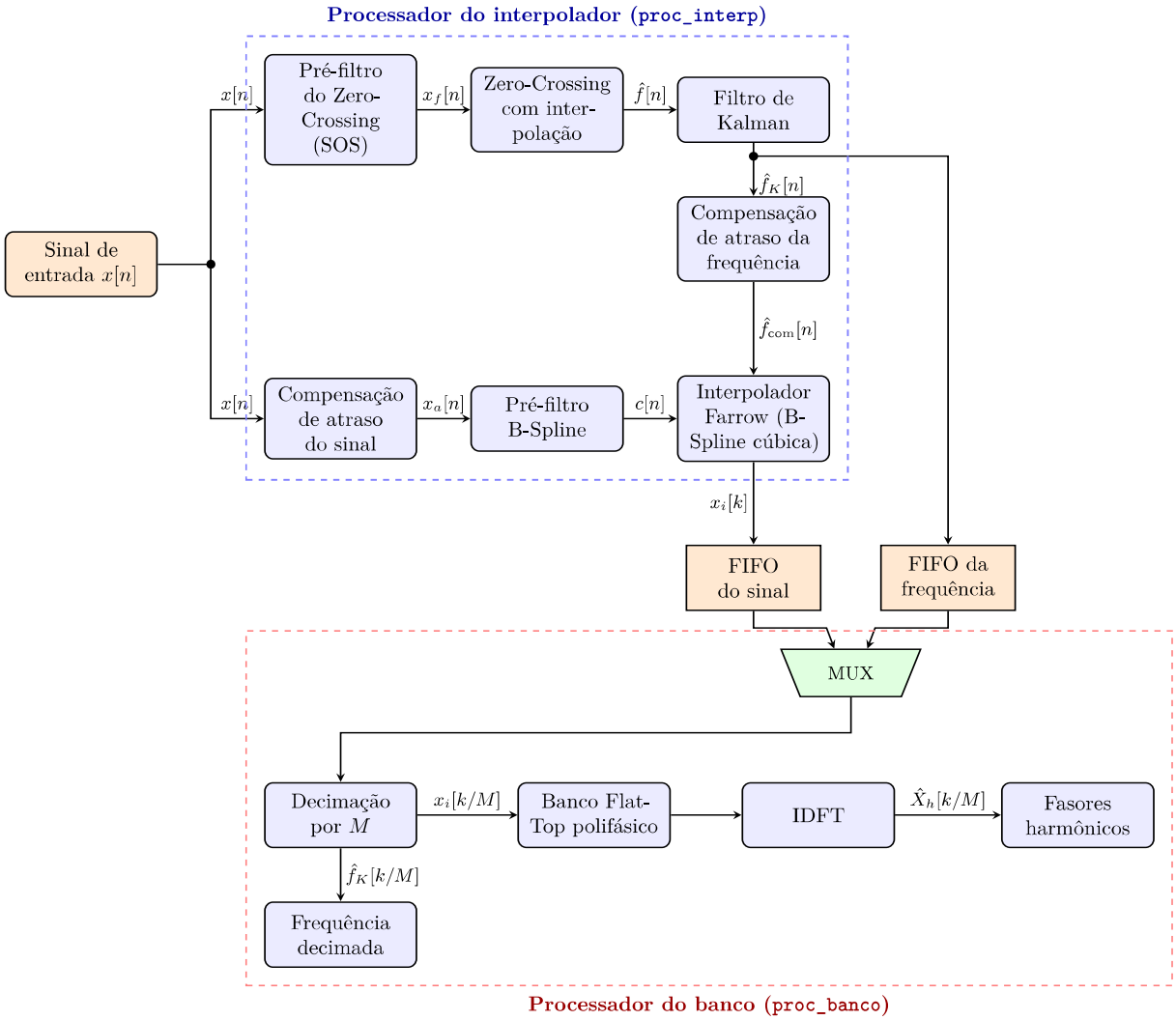


Figura 10 – Pipeline de implementação dividido em dois processadores *soft-core*, ligados por duas FIFOs — uma para o sinal reamostrado e outra para a frequência estimada — e um multiplexador na entrada do processador do banco. Como na Figura 6, o sinal e a frequência percorrem caminhos distintos a partir da entrada, cada um com a sua própria compensação de atraso: 255 amostras no sinal, antes do pré-filtro B-Spline, e 6 amostras na frequência, antes do interpolador Farrow (Tabela 2). No processador do banco os dois fluxos são decimados por M: a frequência encerra aí o seu percurso, emitida uma vez por quadro, enquanto o sinal segue para o banco polifásico e a IDFT, de onde saem os fasores $\hat{X}_h[k/M]$, um quadro a cada M amostras. Na seta entre o banco e a IDFT, $E_\ell(z)$ denota as M saídas das componentes polifásicas, que a transformada combina nos canais harmônicos.

Cada um dos processadores é responsável por uma etapa distinta do método, como descrito a seguir.

- **Processador do interpolador** (`proc_interp`): recebe o sinal de entrada $x[n]$ e executa o pré-filtro do Zero-Crossing, implementado na forma de Seções de Segunda Ordem (do inglês *Second-Order Sections*) (SOS), a detecção de cruzamento por zero com interpolação linear, a suavização da frequência por filtro de Kalman, os atrasos e descartes de amostras necessários ao alinhamento temporal, o pré-filtro B-Spline e a estrutura de Farrow. Suas saídas é a sequência reamostrada $x_i[k]$ e a frequência suavisada $f[n]$.

- **Processador do banco** (`proc_banco`): consome *dois* fluxos: o sinal reamostrado e a frequência estimada, organiza os coeficientes do filtro Flat-Top na forma polifásica, aplica a decimação por M, executa a filtragem polifásica e a IDFT, produzindo os fasores harmônicos.

A separação em dois processadores reflete a diferença de natureza entre as etapas da cadeia: o interpolador opera amostra a amostra, em taxa cheia, enquanto o banco executa um bloco de cálculo pesado (filtragem polifásica e IDFT) apenas uma vez a cada M amostras, por causa da decimação. Para desacoplar essas taxas, dois sinais produzidos pelo interpolador são armazenados em memórias FIFO, das quais o `proc_banco` os consome: como a carga do banco varia de forma intermitente, as FIFO se enchem enquanto ele executa as etapas mais pesadas e se esvaziam quando ele está ocioso. Uma máquina de estados coordena o disparo do processador do banco a partir do preenchimento dessas FIFO.

Ao final da simulação, os dados são lidos de volta no Python para o pós-processamento (ajuste de escala, junção das partes real e imaginária, remoção de componente DC, correção de fase e reconstrução dos fasores). Essa etapa também permite a validação cruzada entre o modelo de referência e a execução embarcada.

4.3 Implementação do `proc_interp`

4.3.1 Pré-filtro do Zero-Crossing em arquitetura SOS

O detector de cruzamento por zero precisa enxergar apenas a componente fundamental; harmônicos e ruído produziriam cruzamentos indesejados. A função do pré-filtro é, portanto, atenuar tudo acima da fundamental, o que se resume a posicionar adequadamente a frequência de corte de acordo com os coeficientes adotados. A Figura 11 mostra a resposta em magnitude obtida com os coeficientes efetivamente utilizados.

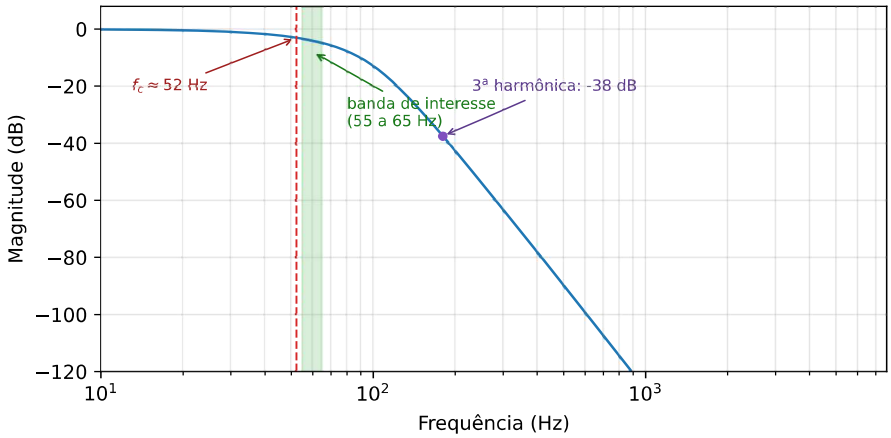


Figura 11 – Resposta em magnitude do pré-filtro Bessel do Zero-Crossing, em cascata de três seções de segunda ordem, com a frequência de corte e a banda de interesse destacadas.

A frequência de corte de -3 dB fica em torno de 52 Hz, e a banda de interesse — a faixa *off-nominal* de 55 Hz a 65 Hz — situa-se logo após o joelho da resposta. Já a terceira harmônica, em 180 Hz, é atenuada em cerca de 38 dB, e as ordens superiores em mais ainda. A atenuação da própria fundamental não é um problema aqui, pois o Zero-Crossing depende apenas dos *instantes* de cruzamento, e não da amplitude do sinal filtrado.

O filtro escolhido é do tipo Bessel. Essa escolha se justifica por duas razões. A primeira é a característica de **fase aproximadamente linear** na faixa de interesse: como a frequência estimada é depois alinhada temporalmente com o sinal (Seção 3.2.2 e seguinte), é desejável que o atraso introduzido pelo filtro seja bem comportado e previsível. A Figura 12 apresenta a resposta de fase e o atraso de grupo dos mesmos coeficientes.

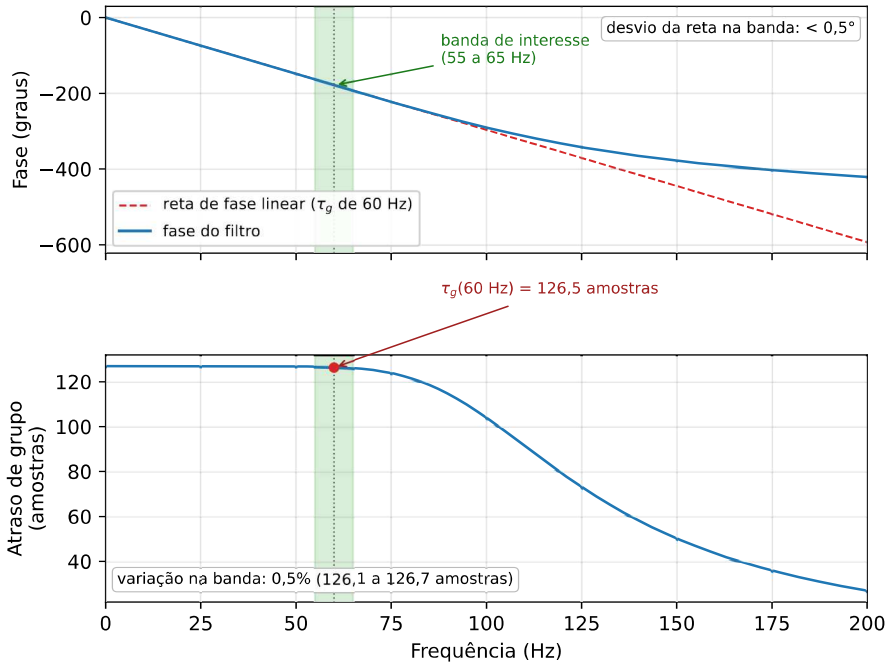


Figura 12 – Resposta de fase (acima) e atraso de grupo (abaixo) do pré-filtro Bessel do Zero-Crossing. Na banda de interesse a fase praticamente coincide com a reta de fase linear traçada com o atraso de grupo de 60 Hz, e o atraso permanece constante em torno de 126,5 amostras.

Na banda *off-nominal* a fase acompanha a reta de referência com desvio inferior a 0,5°, e o atraso de grupo permanece praticamente constante, variando apenas 0,5 % entre 55 Hz e 65 Hz. O atraso de grupo em 60 Hz é de 126,5 amostras, valor que corresponde às 127 amostras de compensação registradas na Tabela 2. Como essa variação é desprezível dentro da faixa de operação, compensar o atraso por um valor *fixo* de amostras é uma aproximação válida, e não uma simplificação grosseira — é justamente isso que a fase linear garante. A segunda é o **custo computacional**: por ser um filtro de resposta ao impulso infinita (IIR), o Bessel atinge a seletividade necessária com uma ordem baixa, exigindo menos operações por amostra do que um filtro de resposta finita (FIR) equivalente — vantagem relevante em um processador embarcado.

A dificuldade prática surgiu na realização numérica. Na **forma direta**, o filtro de ordem 6 é realizado como um único quociente de polinômios,

$$H(z) = \frac{b_0 + b_1 z^{-1} + \cdots + b_6 z^{-6}}{1 + a_1 z^{-1} + \cdots + a_6 z^{-6}}, \tag{4.1}$$

e os coeficientes b_i que dela resultam têm magnitude muito pequena. Sob a aritmética de precisão limitada do núcleo SAPHO, esses coeficientes diminutos provocam perda de precisão e degradação da resposta. A solução adotada foi fatorar a Equação (4.1) em uma **cascata de seções de segunda ordem** (SOS, *second-order sections*), isto é, escrever a mesma função de transferência como um produto de estágios de segunda ordem,

$$H(z) = \prod_{i=1}^{N_s} \frac{b_{0i} + b_{1i}z^{-1} + b_{2i}z^{-2}}{1 + a_{1i}z^{-1} + a_{2i}z^{-2}}, \tag{4.2}$$

com $N_s = 3$ seções para a ordem 6 adotada e com normalização do ganho distribuída entre elas. As Equações (4.1) e (4.2) descrevem o *mesmo* filtro — a diferença está apenas em como ele é calculado. Essa fatoração mantém os coeficientes de cada estágio em uma faixa numérica saudável, preservando a precisão e a estabilidade do filtro sob quantização. A arquitetura SOS é, assim, uma decisão de implementação motivada diretamente pela limitação numérica observada na forma direta.

A estabilidade da cascata pode ser verificada pelos coeficientes de cada seção: para polos complexos conjugados, $|p_i| = \sqrt{a_{2i}}$. Para $f_s = 15{,}36\,\text{kHz}$, obtêm-se $|p_1| \approx 0{,}9671$, $|p_2| \approx 0{,}9710$ e $|p_3| \approx 0{,}9804$, todos no interior do círculo unitário, como mostra a Figura 13. A proximidade da fronteira decorre da baixa frequência de corte normalizada ($\omega_c = 2\pi f_c/f_s \approx 0{,}0368\,\text{rad}$) e reforça a necessidade da forma SOS.

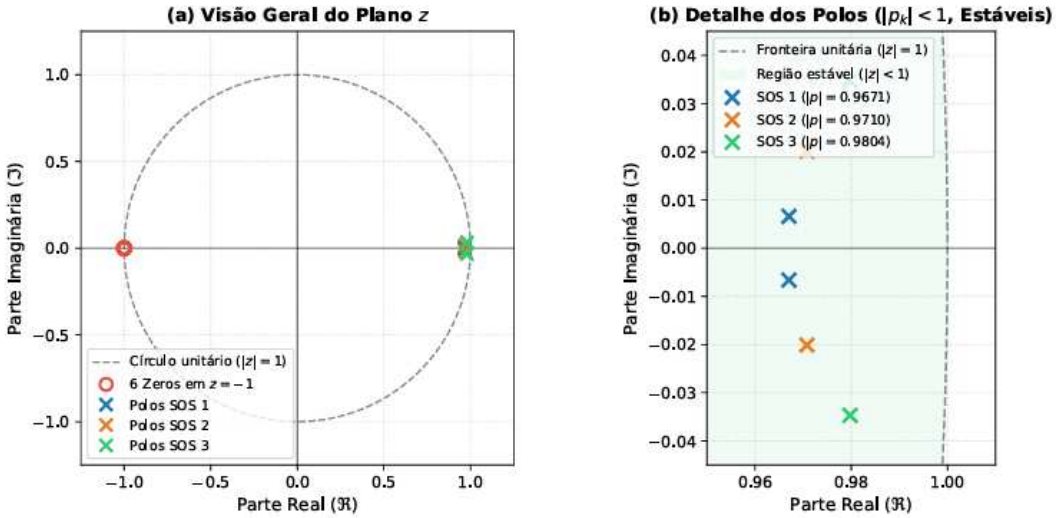


Figura 13 – Diagrama de polos e zeros do pré-filtro Bessel em SOS ($f_s = 15{,}36\,\text{kHz}$): (a) visão geral do plano z, com os seis zeros em $z = -1$; (b) detalhe dos três pares de polos, todos com $|p_i| < 1$.

4.3.2 Zero Crossing de meio ciclo

O detector recebe a saída do pré-filtro Bessel da Seção 4.3.1, chamada `acc` no código, e entrega uma estimativa de frequência a cada cruzamento. O Código 4.1 mostra a implementação.

A detecção é feita pelo produto `sig = va*acc` entre a amostra anterior e a atual: quando esse produto é negativo, as duas têm sinais opostos e houve um cruzamento entre elas. Não há teste de direção, de modo que o detector **dispara tanto na subida quanto na descida** da forma de onda. O intervalo medido é, assim, meio período, e é por isso que a frequência sai como `fzc = 1/(2*Tsc)`, exatamente a Equação 3.2 do Capítulo 3.

Código 4.1 – Detecção de cruzamento por zero de meio ciclo e estimação da frequência no `proc_interp`.

```
sig     = va * acc;          // acc é a saída do pré-filtro Bessel; produto < 0 indica
    ↪ cruzamento
cont_zc = cont_zc + 1;       // conta amostras inteiras desde o cruzamento anterior
Tsc     = cont_zc * Ts;      // intervalo bruto, ainda sem as parcelas fracionárias

if (sig < 0)
{
    Nb  = acc / (acc - va);  // fração do intervalo entre va e acc em que o zero ocorre
    T2  = Nb * Ts;           // resíduo entre o zero real e a amostra atual
    Tsc = Tsc + T1 - T2;     // meio ciclo medido de zero real a zero real
    fzc = 1 / (2*Tsc);       // meio período convertido em frequência
    T1  = T2;                // resíduo é repassado à próxima medição
    cont_zc = 0.0;           // reinicia a contagem de amostras
}

va = acc;                    // amostra anterior da próxima iteração
```

Medir meio ciclo em vez do ciclo completo dobra a taxa de atualização da estimativa, o que é relevante porque é essa frequência que controla a reamostragem.

O cruzamento real quase nunca coincide com uma amostra, e associá-lo diretamente ao índice da amostra introduziria um erro de até um período de amostragem. Por isso, o código calcula `Nb = acc/(acc - va)`, que é a fração do intervalo entre as duas amostras vizinhas em que a reta que as liga corta o eixo. É essa a construção desenhada na Figura 7: a semelhança de triângulos entre $x_f[n-1]$, $x_f[n]$ e o instante estimado $\hat{t}_{zc}$. Há apenas uma diferença de forma em relação à Equação (3.1): lá a fração é medida *a partir da amostra anterior*, enquanto no código ela é medida *a partir da amostra atual*, sendo uma o complemento da outra. É a mesma reta, lida do outro lado, e é por isso que `T2` aparece *subtraído* em `Tsc + T1 - T2`.

Essa leitura pelo lado da amostra atual é o que permite encadear as correções fracionárias. A variável `T2` é a sobra do cruzamento atual e `T1` é a sobra guardada do cruzamento anterior; somar `T1` e subtrair `T2` faz o intervalo ir de zero real a zero real, e não de amostra a amostra. Como `T1` recebe `T2` ao final, o resíduo é repassado à medição seguinte em vez de ser descartado, e o erro fracionário **não se acumula** de meio ciclo para meio ciclo.

Resta o ponto em que a implementação se afasta da forma direta. A maneira natural de medir o intervalo seria acumular o período de amostragem a cada amostra, $T_{sc} \leftarrow T_{sc} + T_s$. Em ponto flutuante de precisão simples, isso significa somar cerca de 128 parcelas pequenas a um total que cresce, e o arredondamento de cada soma se acumula ao longo do meio ciclo. A forma adotada substitui essa acumulação por uma **contagem inteira de amostras** seguida de **uma única multiplicação** no instante do cruzamento, $T_{sc} = \texttt{cont_zc} \cdot T_s$. O erro deixa, então, de depender do número de amostras percorridas. Essa é a primeira de várias decisões tomadas em função da aritmética do núcleo, e a

Seção 4.6 a retoma quantitativamente na Tabela 7.

4.3.3 Suavização por Filtro de Kalman

A frequência entregue pelo Zero-Crossing é ruidosa e muda em degraus, a cada meio ciclo. O filtro de Kalman a suaviza e, ao mesmo tempo, estima a sua derivada. O Código 4.2 mostra a implementação.

Código 4.2 – Filtro de Kalman escalar no `proc_interp`, com estado em desvio da nominal e soma compensada.

```
// ---------- Predição: propaga estado e covariância um passo à frente ----------
kah_u      = Ts * df - comp_dfreq;      // incremento Ts*rocof, já descontado o resíduo pendente
kah_t      = dfreq + kah_u;             // soma sujeita a arredondamento
comp_dfreq = (kah_t - dfreq) - kah_u;   // parcela perdida no arredondamento
dfreq_pred = kah_t;                     // desvio de frequência predito
df_pred    = df;                        // rocof predito: constante no modelo de rampa

p00_pred = p00 + Ts*p10 + Ts*p01 + Ts*Ts*p11 + q00;   // F*P*F' + Q, escrito termo a termo
p01_pred = p01 + Ts*p11 + q01;
p10_pred = p10 + Ts*p11 + q10;
p11_pred = p11 + q11;

// ---------- Correção: incorpora a medida do Zero-Crossing ----------
y_kalman    = fzc - f_nominal;          // medida convertida para desvio da nominal
erro        = y_kalman - dfreq_pred;    // inovação
S_incerteza = p00_pred + R;             // covariância da inovação
K0 = p00_pred / S_incerteza;            // ganho de Kalman: componente da frequência
K1 = p10_pred / S_incerteza;            // ganho de Kalman: componente do rocof

kah_u      = K0 * erro - comp_dfreq;    // correção, já descontado o resíduo pendente
kah_t      = dfreq_pred + kah_u;
comp_dfreq = (kah_t - dfreq_pred) - kah_u;
dfreq      = kah_t;                     // estado corrigido: desvio em relação a 60 Hz
df         = df_pred + K1 * erro;

p00 = (1.0 - K0) * p00_pred;            // (I - K*H)*P, com H = [1 0]
p01 = (1.0 - K0) * p01_pred;
p10 = p10_pred - K1 * p00_pred;
p11 = p11_pred - K1 * p01_pred;

fcc = f_nominal + dfreq;                // frequência absoluta entregue ao interpolador
```

Esse código deve produzir o mesmo resultado das Equações (3.9), (3.10), (3.11) e (3.12) da Seção 3.2.2, com $\mathbf{F}$, $\mathbf{H}$, $\mathbf{Q}$ e R como definidos nas Equações (3.7) e (3.8). A correspondência entre os nomes é direta: `p00_pred` a `p11_pred` são $\mathbf{P}^{-}[n]$; `erro` é a inovação $e[n]$; `S_incerteza` é $S[n]$; `K0` e `K1` são as duas componentes de $\mathbf{K}[n]$; e `fcc` é a frequência suavizada entregue ao interpolador. Ainda assim, três diferenças em relação à escrita matricial do Capítulo 3 saltam à vista no código, e vale explicar cada uma.

A primeira é a **forma escalar**. O núcleo SAPHO não dispõe de operações matriciais, de modo que a matriz de covariância foi aberta nos quatro escalares `p00`, `p01`, `p10` e `p11`, e o produto $\mathbf{FPF}^{\top} + \mathbf{Q}$ está escrito termo a termo. Trata-se de reescrita pura: as operações realizadas são exatamente as mesmas, apenas sem laço e sem matriz.

A segunda é que o estado guardado não é a frequência absoluta, e sim o **desvio em relação à nominal**, $\Delta f = f - 60\,\text{Hz}$. A medida entra como `y_kalman = fzc - f_nominal` e a saída volta a Hz absolutos como `fcc = f_nominal + dfreq`. Como a translação é constante, ela se cancela tanto na predição quanto na inovação, e o filtro é algebricamente o mesmo — $\mathbf{P}$, $\mathbf{Q}$, R, `K0` e `K1` ficam idênticos. O ganho é puramente numérico: em ponto flutuante de precisão simples, a resolução em torno de 3 Hz é cerca de 16 vezes mais fina do que em torno de 63 Hz.

A terceira é que a atualização do estado `dfreq` não é feita pela soma direta, e sim por uma **soma compensada**: a variável `comp_dfreq` guarda a parcela que o arredondamento descartou e a devolve na iteração seguinte, tanto na predição quanto na correção. A razão é que, depois que $\mathbf{P}$ converge, o ganho `K0` torna-se pequeno e a correção `K0*erro` passa a ser da ordem do próprio arredondamento da soma. Sem a compensação, essa correção seria simplesmente absorvida e o filtro deixaria de responder às novas medidas.

As duas últimas diferenças, portanto, não são escolhas de algoritmo: são **decisões impostas pela aritmética de precisão limitada** do núcleo, e ambas são retomadas em detalhe na Seção 4.6, que apresenta na Tabela 6 o erro do estado do filtro após 14 336 atualizações.

4.3.4 Interpolação B-spline com Pré-filtro

O interpolador é a etapa que absorve a variação de frequência: ele gera, a partir do sinal amostrado a taxa fixa, uma nova sequência com N_{ppc} amostras por ciclo elétrico, qualquer que seja a frequência da rede. A implementação segue a cadeia descrita em (ALEIXO *et al.*, 2022), que trata exatamente do mesmo problema — reamostragem em tempo real para estimação de fasores harmônicos — e é composta de pré-filtro causal truncado, avaliação B-Spline cúbica em estrutura de Farrow e acumulador de fase. As equações correspondentes já foram apresentadas no Capítulo 3.

A B-Spline cúbica é suave, mas não passa pelos valores das amostras originais. O pré-filtro corrige isso, convertendo as amostras nos coeficientes da estrutura de Farrow, conforme as Equações (3.14) e (3.15). O pré-filtro exato, porém, é **simétrico e de dois lados**: tem uma metade causal e uma metade anticausal, e esta última exigiria conhecer amostras futuras, o que é inviável em tempo real. A solução adotada em (ALEIXO *et al.*, 2022), e reproduzida aqui, é truncar a metade anticausal em um número finito de termos e realizá-la como um filtro FIR, aceitando em troca um atraso puro. O Código 4.3 mostra as duas metades.

O coeficiente `den_pre_BSP[1] = 0,267 949 19` é $-s$, com $s = \sqrt{3} - 2$ dado pela Equação (3.16): o termo recursivo é a metade causal do pré-filtro. Os seis coeficientes de `num_pre_BSP`, iguais a 1,607 695 15, −0,430 780 62, 0,115 427 32, −0,030 928 66, 0,008 287 31 e −0,002 220 58, formam a progressão geométrica $-6s^{r+1}$ truncada: é a metade anticausal,

Código 4.3 – Pré-filtro B-Spline: FIR truncado da metade anticausal seguido do polo causal.

```
buffer_entrada_prefiltro[0] = buffer_entrada_prefiltro[1];   // desloca o histórico de entrada
buffer_entrada_prefiltro[1] = buffer_entrada_prefiltro[2];
buffer_entrada_prefiltro[2] = buffer_entrada_prefiltro[3];
buffer_entrada_prefiltro[3] = buffer_entrada_prefiltro[4];
buffer_entrada_prefiltro[4] = buffer_entrada_prefiltro[5];
buffer_entrada_prefiltro[5] = x_atrasado;                    // amostra mais recente

// Metade anticausal do pré-filtro, truncada em 6 termos: coeficientes -6*s^(r+1)
dot_result = buffer_entrada_prefiltro[0] * num_pre_BSP[0] +
             buffer_entrada_prefiltro[1] * num_pre_BSP[1] +
             buffer_entrada_prefiltro[2] * num_pre_BSP[2] +
             buffer_entrada_prefiltro[3] * num_pre_BSP[3] +
             buffer_entrada_prefiltro[4] * num_pre_BSP[4] +
             buffer_entrada_prefiltro[5] * num_pre_BSP[5];

// Metade causal: polo único em s = sqrt(3) - 2
buffer_saida_prefiltro[1] = dot_result
                          - den_pre_BSP[1] * buffer_saida_prefiltro[0];
buffer_saida_prefiltro[0] = buffer_saida_prefiltro[1];       // estado do polo
```

aplicada com os índices invertidos no tempo. O preço dessa inversão é um atraso de 6 amostras, e é exatamente por isso que a frequência estimada precisa ser atrasada antes de entrar no interpolador — a linha correspondente da Tabela 2. Vale registrar que (ALEIXO *et al.*, 2022) utiliza oito termos no truncamento, enquanto a implementação adotou seis, número suficiente para a precisão pretendida e que economiza duas multiplicações por amostra.

Com os coeficientes da estrutura de Farrow disponíveis, a interpolação propriamente dita é feita pela estrutura de Farrow, mostrada no Código 4.4.

Os quatro subfiltros `H0` a `H3` são literalmente as Equações (3.19) a (3.22), com as frações $-1/6$, $1/2$ e $2/3$ escritas como constantes de 10 casas decimais. A avaliação `y = ((H0*alfa + H1)*alfa + H2)*alfa + H3` é a Equação (3.17) na forma de Horner: três multiplicações em lugar das seis da forma direta, sem que α^2 e α^3 precisem ser calculados explicitamente — vantagem de custo e também de precisão, já que elevar a potência amplifica o erro do próprio α. Como apenas quatro amostras são necessárias, o buffer é deslocado diretamente, sem índice circular; para uma janela tão curta, o deslocamento é mais barato que a aritmética de ponteiro.

O que torna a reamostragem assíncrona é o acumulador de fase. O passo `lambda_val = 60.0/freq_atrasada` é a razão da Equação (3.13), e o par `alfa += lambda_val` e `alfa -= 1.0` é a Equação (3.24). Como λ não é inteiro, o laço `while (alfa < 1.0)` emite **zero, uma ou duas** amostras de saída para cada amostra de entrada, conforme a frequência estimada esteja acima ou abaixo da nominal. É esse laço que mantém $N_{ppc} = 256$ amostras por ciclo na saída, independentemente da frequência da rede, e é por isso que o banco a jusante pode operar com coeficientes fixos, sem reprojeto.

Código 4.4 – Estrutura de Farrow e acumulador de fase do interpolador B-Spline cúbico.

```
buffer_entrada_farrow[0] = buffer_entrada_farrow[1];   // desloca as quatro amostras do
    ↪ interpolador
buffer_entrada_farrow[1] = buffer_entrada_farrow[2];
buffer_entrada_farrow[2] = buffer_entrada_farrow[3];
buffer_entrada_farrow[3] = buffer_saida_prefiltro[1];  // coeficiente de spline mais recente

freq_atrasada = buffer_freq[0];       // frequência alinhada ao atraso do pré-filtro
lambda_val    = 60.0 / freq_atrasada; // passo de fase: nominal sobre frequência estimada

while (alfa < 1.0)                    // emite zero, uma ou duas amostras por amostra de entrada
{
    H0 = (-0.1666666667*buffer_entrada_farrow[0]) + (0.5*buffer_entrada_farrow[1])
       + (-0.5*buffer_entrada_farrow[2])          + (0.1666666667*buffer_entrada_farrow[3]);
    H1 = (0.5*buffer_entrada_farrow[0]) - buffer_entrada_farrow[1]
       + (0.5*buffer_entrada_farrow[2]);
    H2 = (-0.5*buffer_entrada_farrow[0]) + (0.5*buffer_entrada_farrow[2]);
    H3 = (0.1666666667*buffer_entrada_farrow[0]) + (0.6666666667*buffer_entrada_farrow[1])
       + (0.1666666667*buffer_entrada_farrow[2]);

    y = ((H0 * alfa + H1) * alfa + H2) * alfa + H3;   // avaliação por Horner
    fout(0, y * 1000000);             // amostra reamostrada enviada à FIFO do proc_banco
    alfa = alfa + lambda_val;         // avança a fase fracionária
}
alfa = alfa - 1.0;                    // devolve a fase ao intervalo [0, 1)
```

4.3.5 Compensação dos Atrasos e Descarte de Transitórios

4.3.6 Atrasos e descartes de amostras

A cadeia introduz atrasos que precisam ser compensados para que a frequência estimada corresponda ao mesmo trecho de sinal que está sendo reamostrado. A Tabela 2 lista esses atrasos.

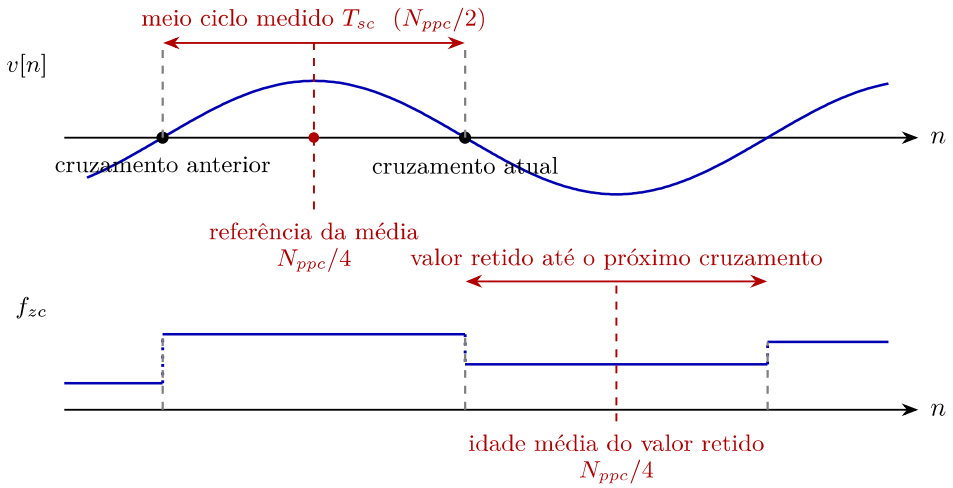
Tabela 2 – Atrasos compensados na cadeia.

Etapa	**Amostras**	**Taxa**
Pré-filtro IIR Bessel do Zero-Crossing, em 60 Hz	127	cheia
Latência de medição e retenção do Zero-Crossing	128	cheia
Atraso total aplicado a $x[n]$ antes do B-Spline	255	cheia
Frequência estimada, compensando o pré-filtro B-Spline	6	cheia

Nem toda parcela dessa tabela é um atraso de grupo, e vale distinguir os casos. As 127 amostras do pré-filtro Bessel são atraso de grupo no sentido estrito — há um filtro, com resposta ao impulso e resposta de fase, e o valor é lido diretamente dela (Seção 4.3.1). Já o Zero-Crossing não possui filtro nem *buffer*: ele apenas mede intervalos entre cruzamentos. Ainda assim, a frequência que ele entrega refere-se a um instante no passado, e por duas razões independentes, cada uma valendo $N_{ppc}/4 = 64$ amostras.

A primeira é que a estimativa é uma **média**, e não um valor instantâneo. O detector acumula o intervalo T_{sc} entre dois cruzamentos consecutivos e calcula $f_{zc} = 1/(2T_{sc})$, de

modo que o valor obtido caracteriza o meio ciclo inteiro que acabou de transcorrer. A referência temporal correta de uma média sobre um intervalo é o *ponto médio* desse intervalo, e não o instante em que ela fica disponível — daí $N_{ppc}/4$. A segunda é a **retenção**: f_{zc} só é recalculada quando ocorre um novo cruzamento, mas é emitida a cada amostra, mantida constante no intervalo entre cruzamentos. A idade média desse valor retido, ao longo do meio ciclo em que ele permanece válido, é novamente $N_{ppc}/4$. A Figura 14 representa as duas parcelas.



$$N_{ppc}/4 \;+\; N_{ppc}/4 \;=\; N_{ppc}/2 \;=\; 128 \text{ amostras}$$

Figura 14 – Origem das 128 amostras de latência do Zero-Crossing. Acima, a estimativa f_{zc} resulta de uma média sobre o meio ciclo entre dois cruzamentos, cuja referência temporal é o ponto médio do intervalo. Abaixo, esse valor é retido até o cruzamento seguinte, o que acrescenta uma idade média de mais um quarto de ciclo.

Somadas, as duas parcelas dão $N_{ppc}/2 = 128$ amostras, e com o pré-filtro chega-se a $127 + 128 = 255$ — exatamente o valor aplicado ao sinal de entrada nos dois projetos, tanto no modelo de referência em Python quanto no processador do interpolador. É essa compensação que garante que o trecho de sinal que entra no interpolador seja o mesmo a que se refere a frequência que o corrige.

Além da compensação de atraso, é necessário **descartar** amostras no início da execução, antes de habilitar cada etapa. A Tabela 3 apresenta esses descartes.

Tabela 3 – Amostras descartadas antes de habilitar cada etapa.

Etapa	**Amostras**	**Ciclos**
Antes de habilitar o filtro de Kalman	1024	4
Antes de habilitar o pré-filtro B-Spline e a estrutura de Farrow	307 200	1200
Até liberar a primeira saída da estrutura de Farrow	7	—

A comparação entre as duas tabelas revela um ponto importante: os atrasos compensados no interpolador somam pouco mais de duas centenas de amostras, mas o descarte efetivo antes da reamostragem é de 307 200 amostras, ou 1200 ciclos da fundamental — três ordens de grandeza maior. A razão é que **não basta compensar atraso**: é preciso esperar o **transitório do filtro de Kalman** convergir. O filtro parte de estado inicial nulo e de uma matriz de covariância arbitrária e leva centenas de ciclos até que a estimativa de frequência se estabilize dentro da tolerância desejada. Usar a saída do Kalman antes disso significa alimentar o interpolador com uma frequência incorreta, o que desloca a base temporal da sequência reamostrada e contamina a fase de *todas* as componentes harmônicas — efeito tanto mais severo quanto maior a ordem h. Por isso o descarte é dimensionado pelo transitório do Kalman, e não pela soma dos atrasos compensados.

4.4 Implementação do `proc_banco`

O `proc_banco` recebe pela FIFO (Seção 4.5.1) a sequência já reamostrada e entrega, uma vez a cada ciclo da fundamental, os 51 fasores harmônicos de ordem $h = 0$ a 50. As diretivas do núcleo declaram a aritmética e o tamanho da transformada: `#NUBITS 32`, `#NBMANT 23`, `#NBEXPO 8` e `#FFTSIZ 8`, este último igual a $\log_2 256$.

O banco precisa das últimas 256 amostras a cada quadro. Guardá-las deslocando um vetor custaria 255 cópias por amostra recebida, o que é proibitivo à taxa cheia. Em vez disso, o armazenamento é feito em um **buffer circular** de 256 posições com um ponteiro de escrita, e cada amostra custa uma única atribuição, como mostra o Código 4.5.

Código 4.5 – Buffer circular de entrada do `proc_banco`.

```
buffer_head = buffer_head - 1;              // ponteiro de escrita anda para trás
if (buffer_head < 0)
{
    buffer_head = M - 1;                    // retorno ao fim do vetor, sem operador módulo
}
buffer[buffer_head] = in(0) / 1000000.0;    // amostra reamostrada, lida da FIFO
```

Dois detalhes desse trecho merecem registro. O primeiro é que o retorno ao fim do vetor é feito por um `if` explícito, e não pelo operador módulo: o núcleo não dispõe de divisão inteira barata, ao passo que a comparação é imediata. O segundo é o sentido dos ponteiros. A escrita *decrementa* o índice, enquanto a leitura, feita adiante pelo índice `ii`, *incrementa* a partir de `buffer_head`. O vetor é assim percorrido da amostra mais recente para a mais antiga, que é exatamente a ordem $x[n], x[n-1], x[n-2], \ldots$ exigida pela convolução — a inversão temporal sai de graça no sentido de varredura, sem nenhuma operação adicional.

A filtragem propriamente dita está no Código 4.6.

Código 4.6 – Decimação por M e laço polifásico do banco Flat-Top.

```
if (sample_count == (M - 1))      // o bloco roda uma vez a cada M = 256 amostras
{
    sample_count = 0;
    mm = 0;                       // índice da fase
    ii = buffer_head;             // leitura começa na amostra mais recente

    while (mm < M)
    {
        E[mm][7] = E[mm][6];      // desloca a linha de atraso da fase mm
        E[mm][6] = E[mm][5];
        E[mm][5] = E[mm][4];
        E[mm][4] = E[mm][3];
        E[mm][3] = E[mm][2];
        E[mm][2] = E[mm][1];
        E[mm][1] = E[mm][0];
        E[mm][0] = buffer[ii];    // amostra que cabe a esta fase neste quadro

        E0[mm] = Ehh[mm][0]*E[mm][0] + Ehh[mm][1]*E[mm][1]   // os 8 taps da fase mm
               + Ehh[mm][2]*E[mm][2] + Ehh[mm][3]*E[mm][3]
               + Ehh[mm][4]*E[mm][4] + Ehh[mm][5]*E[mm][5]
               + Ehh[mm][6]*E[mm][6] + Ehh[mm][7]*E[mm][7];

        mm = mm + 1;
        ii = ii + 1;              // avança da amostra mais nova para a mais antiga
        if (ii >= M) { ii = 0; }
    }

    // E0 é o próprio vetor de trabalho da IFFT de M pontos, executada in place
}
else
{
    sample_count = sample_count + 1;   // nas demais amostras, apenas armazena e conta
}
```

A decimação é explícita: todo o bloco de filtragem está dentro do teste `sample_count == M-1`, com $M = N_{ppc} = 256$, de modo que ele é executado **uma vez a cada** 256 **amostras**. Nas outras 255 o processador apenas armazena a amostra e incrementa o contador. É a forma concreta da economia antecipada no Capítulo 3: a etapa cara roda na taxa decimada, e não na taxa cheia.

A matriz `Ehh[256][8]` é carregada do arquivo `Ehh.txt` na inicialização e permanece constante. São as 2048 amostras do protótipo Flat-Top de $8N_{ppc}$ coeficientes, já reorganizadas em 256 fases de 8 coeficientes cada — ou seja, a Equação (3.29) resolvida fora do processador, em tempo de projeto. Cada fase tem sua própria linha de atraso, `E[mm][0]` a `E[mm][7]`, e seu produto interno de oito termos escrito por extenso, sem laço interno. O resultado de cada fase é gravado diretamente em `E0[mm]`, que é o próprio vetor de trabalho da IFFT: não há cópia intermediária entre o banco e a transformada.

Vale explicar por que o protótipo tem justamente $8N_{ppc}$ coeficientes, e não menos. A decimação por $M = N_{ppc}$ põe a saída de cada canal a $F_s/M = f_0 = 60\,\text{Hz}$, de modo que, pelo teorema da amostragem, a banda útil de cada canal é de $\pm f_0/2 = \pm 30\,\text{Hz}$ em torno do seu centro hf_0. Tudo o que estiver fora dessa faixa é rebatido para dentro dela, e o rebatimento é ditado pelo próprio passo de decimação: uma componente afastada de kf_0

do centro do canal dobra *exatamente* sobre o mesmo ponto da banda. Os interferentes que importam, portanto, não estão espalhados pelo espectro — estão precisamente sobre os canais harmônicos vizinhos, $(h \pm 1)f_0$, $(h \pm 2)f_0$ e assim por diante. Disso decorrem duas exigências simultâneas sobre o protótipo: na passagem, ele precisa ser **plano** sobre a excursão de frequência dos ensaios, que percorrem de 55 Hz a 65 Hz, ou ±5 Hz em torno da nominal, faixa na qual o desvio de magnitude não passa de 0,01 %; na rejeição, ele precisa atenuar fortemente os **múltiplos de** f_0, e o protótipo adotado fica em −116 dB em ±60 Hz e abaixo de −62 dB em toda a vizinhança 60 Hz ± 5 Hz que pode dobrar sobre a banda útil. Na borda da banda útil, em ±30 Hz, o ganho do protótipo é de −9,6 dB.

Sobre `E0` roda em seguida uma IFFT de 256 pontos *in place*, com os fatores de rotação lidos do arquivo `wpv.txt`. Ao final, os 51 primeiros elementos são emitidos com parte real e parte imaginária intercaladas, precedidos da frequência estimada — formato que o script de pós-processamento em Python desempacota diretamente do arquivo de simulação.

Vale contrastar essa organização com a do modelo de referência. O Python trabalha com o sinal inteiro residente em memória: as fases saem de um fatiamento com passo sobre o protótipo, os atrasos $z^{-\ell}$ de cada ramo são obtidos criando M cópias deslocadas do sinal completo, a decimação é outro fatiamento, cada ramo é filtrado de uma vez sobre toda a subsequência e a IFFT devolve uma matriz complexa com todos os canais em todos os instantes. Isso é conveniente para validar o algoritmo, mas pressupõe memória proporcional ao comprimento do sinal, o que não existe no processador embarcado. No SAPHO cada uma dessas construções foi trocada pela sua versão amostra a amostra: as M cópias do sinal viraram *um* buffer circular de 256 posições; o fatiamento da decimação virou um contador; a filtragem de cada ramo virou o produto interno de oito termos acumulado sobre a linha de atraso da fase; e a matriz de saída virou um único vetor de 256 posições, reaproveitado a cada quadro. O algoritmo é o mesmo, mas com memória constante em vez de proporcional ao sinal — e é justamente essa equivalência que a comparação com o modelo de referência, apresentada no Capítulo 5, verifica.

O banco introduz um atraso próprio, e ele é o contraponto exato do pré-filtro Bessel da Seção 4.3.1: lá a fase é *aproximadamente* linear, aqui ela é *exatamente* linear. O protótipo Flat-Top é construído como uma soma de cossenos e com $N = 8N_{ppc} + 1 = 2049$ coeficientes. Por ser uma soma de cossenos de argumento simétrico, $h[n]$ resulta simétrico, o que caracteriza um filtro FIR de fase linear. O atraso de grupo é então rigorosamente constante em toda a faixa e vale

$$\tau_g = \frac{N-1}{2} = 1024 \text{ amostras}, \tag{4.3}$$

não como aproximação válida dentro de uma banda, mas como consequência algébrica da simetria. A Figura 15 mostra a fase e o atraso de grupo do protótipo.

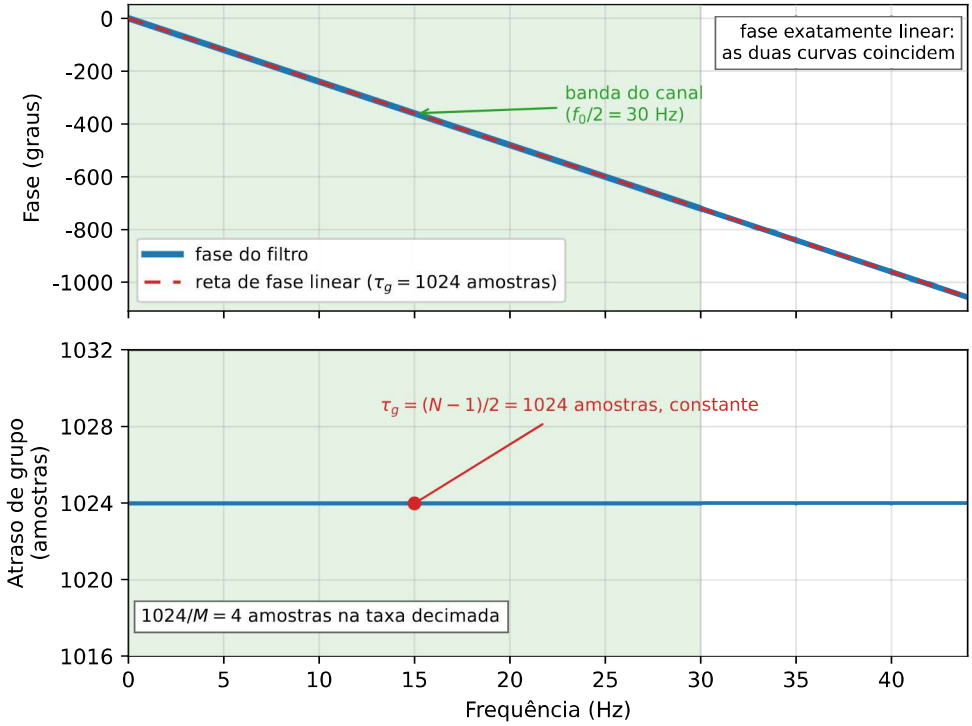


Figura 15 – Resposta de fase (acima) e atraso de grupo (abaixo) do filtro protótipo Flat-Top. Por ser um FIR simétrico, a fase coincide com a reta de fase linear em toda a faixa, e o atraso de grupo é constante e igual a $(N-1)/2 = 1024$ amostras. A banda sombreada é a largura de canal do banco, $f_0/2 = 30\,\text{Hz}$.

Como o banco decima por $M = N_{ppc} = 256$, essas 1024 amostras da taxa cheia correspondem a $1024/256 = 4$ amostras na taxa decimada, que é a forma em que o atraso é efetivamente compensado no pós-processamento — o mesmo atraso, portanto, lido em três unidades equivalentes: 1024 amostras da taxa cheia, 4 ciclos da fundamental ou 4 quadros de saída.

4.5 Interface entre os dois processadores

Os dois processadores soft-core operam em ritmos distintos: o interpolador produz uma amostra por vez, à taxa cheia, enquanto o banco consome e processa um bloco inteiro apenas a cada M amostras. Acoplá-los diretamente faria com que as amostras geradas pelo interpolador enquanto o banco processa fossem perdidas. A interface entre eles resolve isso com dois elementos: um par de memórias Primeiro a Entrar, Primeiro a Sair (do inglês *First-In, First-Out*) (FIFO), que retém os dados enquanto o banco está ocupado, desacoplando as duas taxas; e uma máquina de estados finita, que observa a ocupação das filas e o estado do banco para disparar cada nova rodada de processamento.

4.5.1 FIFOs entre os processadores

A comunicação entre o processador do interpolador e o do banco é feita por memórias do tipo FIFO. Optou-se por uma implementação própria, em vez de utilizar uma IP de FIFO proprietária, para manter independência da ferramenta, garantir portabilidade entre o ambiente de simulação e uma eventual síntese e ter controle total sobre o comportamento da fila.

O sistema emprega duas filas, e não uma. A primeira transporta a sequência reamostrada produzida pela estrutura de Farrow; a segunda, a frequência estimada. As duas são necessárias porque o processador do banco precisa das duas grandezas: o bloco de amostras a filtrar e a frequência correspondente a esse bloco, usada na correção de fase. Ambas têm palavra de 32 bits e profundidade de 16384 posições.

No fluxo do sistema, as filas são alimentadas pelas saídas que o processador do interpolador produz e são consumidas pelo processador do banco quando este é disparado. A ocupação de cada fila é justamente o que a máquina de estados (Seção 4.5.2) observa para decidir quando há dado suficiente para iniciar uma rodada do banco.

A profundidade das FIFO não é arbitrária: ela deve ser dimensionada a partir dos tempos de processamento dos dois processadores, que são diferentes. O interpolador produz uma amostra a cada intervalo correspondente ao seu tempo de processamento por amostra; o banco, por outro lado, fica ocupado por um tempo relativamente longo quando dispara, pois executa a filtragem polifásica e a IDFT de uma só vez. Enquanto o banco está ocupado, o interpolador continua produzindo amostras, que precisam ser armazenadas na fila para não serem perdidas.

O raciocínio de dimensionamento é, portanto, o seguinte: a fila deve comportar todas as amostras que o interpolador é capaz de gerar durante o intervalo em que o banco permanece ocupado processando o bloco anterior, sem transbordar (`full`) e sem ficar vazia (`empty`) no momento em que o banco volta a ler.

O raciocínio acima é suficiente para uma única fila, mas a arquitetura adotada tem duas, e é aí que reside a principal limitação do projeto. As duas filas não são preenchidas na mesma taxa: o sinal reamostrado é escrito a cada amostra produzida pela estrutura de Farrow, enquanto a frequência estimada é na mesma taxa de amostragem original. Como o consumo pelo banco é comum às duas, a fila alimentada mais rapidamente acumula um excedente que cresce ao longo do tempo e, em execução prolongada, acaba por transbordar.

A profundidade de 16 384 posições adotada em cada fila é, nesse sentido, uma medida de contenção: ela é bem maior que a exigida pelo desacoplamento entre T_{banco} e T_{interp}, e existe para absorver esse desbalanceamento durante o intervalo dos ensaios. A condição de disparo verificada pela máquina de estados (Seção 4.5.2) garante que o banco só seja iniciado quando houver dado nas *duas* filas, mas não impede que a diferença de

ocupação entre elas cresça.

A consequência deve ser registrada com clareza: a arquitetura atual não é a ideal, pois o descompasso entre as taxas de escrita pode, em execução suficientemente longa, levar ao transbordamento da fila alimentada mais rapidamente. Cabe uma ressalva, porém, sobre a gravidade desse efeito em cenários práticos. A análise acima supõe um desbalanceamento sistemático e de sentido fixo, mas a cadência de escrita da fila da frequência é ditada pelo Zero-Crossing, que depende da frequência real do sinal. Em operação prática, essa frequência oscila em torno da nominal, ora acima, ora abaixo, de modo que a fila da frequência ora se enche mais rápido, ora mais devagar em relação à do sinal. A ocupação relativa das duas filas, portanto, não cresce monotonicamente: ela varia de sentido ao longo do tempo, o que tende a atenuar o acúmulo e a postergar o transbordamento. Ainda assim, não há garantia de compensação exata entre os dois sentidos, e o problema permanece, em geral, menos severo do que o pior caso de desbalanceamento fixo sugere.

Para a duração dos ensaios apresentados no Capítulo 5 o dimensionamento é suficiente e não houve perda de amostras, o que valida os resultados obtidos. A equalização das taxas de escrita das duas filas. Por exemplo, escrevendo a frequência na mesma cadência do sinal, ou substituindo o par de filas por uma única fila de registros compostos.

4.5.2 Máquina de estados de controle

A coordenação entre os dois processadores é feita por uma máquina de estados finita (FSM). Ela não gerencia a leitura e a escrita internas das filas, tarefa que cabe ao próprio controle de ocupação das FIFO. Sua função é atuar como sequenciador de sincronismo (*handshake*): decidir o instante em que o processador do banco deve iniciar uma nova rodada de processamento. Para isso, verifica duas condições: se há dado disponível em ambas as filas e se o processador do banco está livre, conciliando assim o ritmo do interpolador, que produz uma amostra por vez, com o do banco, que processa um bloco a cada M amostras.

A máquina de estados trabalha com quatro sinais, descritos a seguir em linguagem simples:

- `usedw_sig` — a **quantidade de dados presente na fila do sinal** naquele instante, isto é, quantas amostras reamostradas já foram escritas pelo interpolador e ainda não foram consumidas pelo banco.
- `usedw_freq` — a mesma informação para a **fila da frequência**: quantos valores de frequência estimada estão aguardando na fila.
- `flag_proc_banco` — um aviso que vem do processador do banco (corresponde ao bit `out1_banco[0]`). É a **flag emitida ao final de cada iteração do código do**

banco, sinalizando que aquela rodada de processamento terminou. O código do banco a leva a `1` logo ao iniciar uma rodada, mantendo-a nesse valor enquanto está ocupado, e a devolve a `0` ao concluir. Assim, `0` indica banco livre e `1` indica banco ocupado.

- `rst_proc_banco` — a saída da FSM, que funciona como um "botão de *start*". Quando vai a `1` por um ciclo de *clock*, dá o pulso que dispara (reinicia) o processador do banco para consumir as filas.

A máquina possui quatro estados ativos (`S0` a `S3`) e é descrita com codificação "segura" (*safe*): caso, por alguma falha, ela caia em um estado ilegal, retorna sozinha ao estado inicial `S0`. A Tabela 4 resume o comportamento de cada estado, e a Figura 16 apresenta o diagrama de transições.

Tabela 4 – Comportamento da máquina de estados de controle do banco.

Estado	**Saída `rst_proc_banco`**	**Condição verificada**	**Transição / significado**
`S0` (espera ficar livre)	`0` (não dispara)	o banco está livre? (`flag_proc_banco == 0`)	se sim, vai para `S1`; senão, permanece aguardando o fim da rodada anterior
`S1` (espera ter dado)	`0`	há dado nas *duas* filas? (`usedw_sig > 0` **e** `usedw_freq > 0`)	se sim, vai para `S2`; senão, aguarda o interpolador alimentar as filas
`S2` (dispara)	`1` (pulso de *start*)	— (incondicional)	vai sempre para `S3`; este pulso é o que inicia o `proc_banco`
`S3` (confirma a partida)	`0`	o banco assumiu o disparo? (`flag_proc_banco == 1`)	se sim, volta para `S0`; senão, aguarda. Impede que um segundo pulso seja emitido antes de o banco reagir ao primeiro

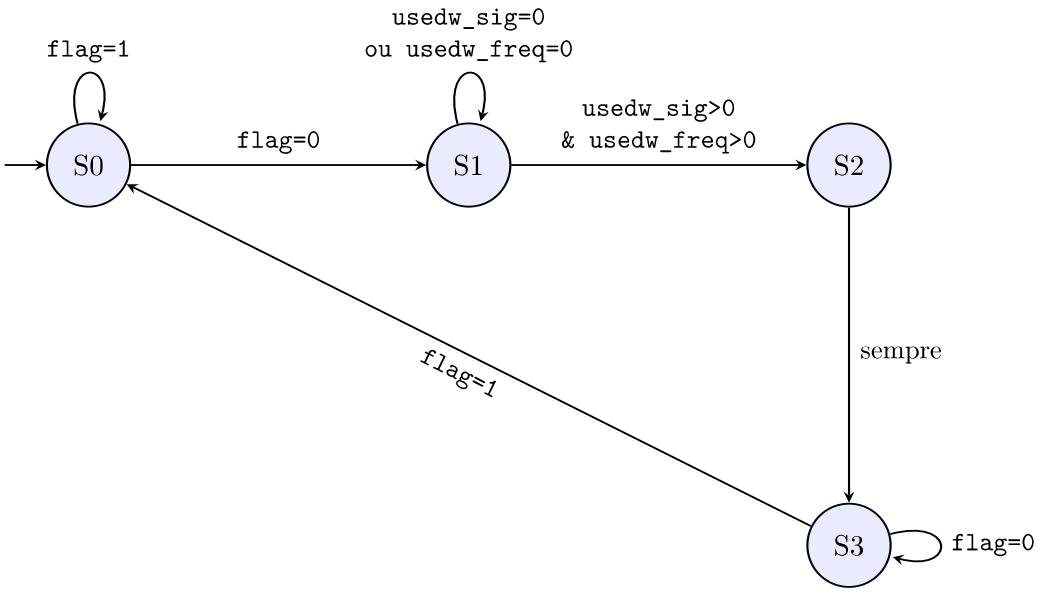


Figura 16 – Diagrama de estados da máquina de controle do processador do banco. A saída `rst_proc_banco` vale 1 apenas em `S2`.

Resumidamente, o funcionamento é: esperar o banco ficar livre, esperar chegar dado nas duas filas, apertar o *start* por um *clock*, esperar o banco confirmar que assumiu o disparo e repetir. Esse aperto de mão garante duas propriedades importantes: o banco **nunca** é iniciado sem que haja dado em ambas as filas e **nunca** é disparado enquanto ainda está ocupado com a rodada anterior. Com isso, evita-se qualquer condição de corrida entre os dois processadores.

Esse mecanismo se encaixa naturalmente com a decimação por M do banco de filtros polifásico. Como o banco produz uma saída a cada M amostras de entrada, ele executa a filtragem polifásica e a IDFT de uma só vez e permanece ocupado por um intervalo relativamente longo a cada rodada. Enquanto isso, o interpolador não interrompe a produção: as amostras que ele gera durante o processamento pesado do banco são retidas nas filas, em vez de perdidas. Quando o banco conclui a rodada, a máquina de estados verifica que há material acumulado e o dispara novamente, consumindo o que se juntou. Dessa forma, a FSM concilia o ritmo do interpolador, que produz uma amostra por vez, com o do banco, que processa um bloco a cada M amostras.

4.6 Decisões impostas pela precisão limitada

As duas últimas decisões de implementação deste capítulo nascem do formato numérico do núcleo, e por isso convém começar por ele. O SAPHO opera com palavras de 32 bits repartidas em 1 bit de sinal, 8 bits de expoente e 23 bits de mantissa, num arranjo similar ao da norma IEEE 754 para precisão simples. Diferentemente do IEEE 754, porém, o bit inteiro não é implícito: ele ocupa uma das 23 posições da mantissa, restando 22 bits para a parte fracionária. A precisão efetiva equivale, assim, a cerca de **sete dígitos**

decimais significativos contra os quinze do formato de dupla precisão usado no modelo Python de referência.

A palavra *significativos* é a que importa aqui. A mantissa guarda dígitos contados a partir do primeiro algarismo do número, e não casas decimais em posição fixa: são sempre os primeiros dígitos que se preservam, quer o número seja 60 ou 0,05. A consequência é que o menor incremento capaz de alterar o valor guardado, a unidade da última posição, ou ULP é **proporcional ao próprio número**: quanto maior o valor armazenado, mais grosseiro o passo. Uma parcela menor que meia ULP, somada a ele, simplesmente desaparece.

Daí decorre uma diretriz de projeto simples: **as duas parcelas de uma soma devem estar na mesma ordem de grandeza**. Somar repetidamente um valor pequeno a um acumulador grande é justamente o caso em que a precisão limitada se manifesta com mais força, e é o que ocorre nas etapas que acumulam ao longo de muitas amostras. A Tabela 5 quantifica o efeito para as grandezas de frequência que circulam nesta cadeia.

Tabela 5 – Resolução absoluta (ULP) do formato de 32 bits do SAPHO para as grandezas de frequência acumuladas na cadeia.

Grandeza acumulada	**Valor típico**	**ULP**
Frequência absoluta f	60 Hz	$7{,}6 \times 10^{-6}$ Hz
Desvio Δf, ensaio *off-nominal*	5 Hz	$9{,}5 \times 10^{-7}$ Hz
Desvio Δf, ensaio de rampa	0,5 Hz	$1{,}2 \times 10^{-7}$ Hz
Desvio Δf, regime próximo ao nominal	0,05 Hz	$7{,}5 \times 10^{-9}$ Hz

A leitura é direta: uma correção de Kalman de 1×10^{-6} Hz, magnitude corriqueira depois que o ganho do filtro assenta, somada a um estado de 60 Hz não altera nenhum bit, pois está abaixo de meia ULP; somada a um estado de 0,05 Hz, é registrada com três dígitos de folga. Duas etapas da cadeia foram reformuladas por causa disso.

A primeira é o **filtro de Kalman**, e a reformulação é a diretriz acima aplicada ao pé da letra: o estado passou a guardar o *desvio* $\Delta f = f - f_0$ em vez da frequência absoluta, com os 60 Hz somados apenas na saída. O acumulador encolhe algumas ordens de grandeza e passa a conviver com as correções que recebe. Em complemento, as duas recursões que atualizam o estado (a predição e a correção pelo ganho de Kalman) usam **somatório compensado** (algoritmo de Kahan), em que uma variável auxiliar guarda o resíduo de arredondamento de cada soma e o reinjeta na seguinte, ao custo de duas operações por atualização. A Tabela 6 mede o efeito das duas medidas sobre a recursão $\hat{x} \leftarrow \hat{x} + K_0\, e$, alimentada por 14 336 correções de magnitude típica 1×10^{-6} Hz — o equivalente a 1 s de ensaio a 15 360 Hz, descontado o transitório inicial de 1024 amostras que o código descarta.

Tabela 6 – Erro do estado do filtro de Kalman após 14 336 atualizações em 32 bits, tomando como referência o mesmo cálculo em dupla precisão.

Variável de estado	**Valor**	**Soma direta**	**Soma de Kahan**	**Ganho**
Frequência absoluta f	60,05 Hz	$1{,}5 \times 10^{-3}$ Hz	$1{,}1 \times 10^{-6}$ Hz	1435×
Desvio Δf	5 Hz	$6{,}4 \times 10^{-6}$ Hz	$1{,}8 \times 10^{-7}$ Hz	35×
Desvio Δf	0,5 Hz	$4{,}2 \times 10^{-6}$ Hz	$4{,}7 \times 10^{-9}$ Hz	894×
Desvio Δf	0,05 Hz	$1{,}7 \times 10^{-9}$ Hz	$1{,}7 \times 10^{-9}$ Hz	—

A melhora vem de dois mecanismos distintos. A mudança da variável de estado encolhe o acumulador e, com ele, a ULP — é o que separa a primeira linha das demais. O somatório compensado ataca a outra metade do problema: o erro da soma direta *cresce* com o número de iterações, enquanto o da soma compensada permanece limitado. Junto ao nominal, na última linha, o acumulador já é tão pequeno que a compensação deixa de fazer diferença mensurável; no ensaio *off-nominal*, porém, o desvio de 5 Hz devolve ao acumulador uma ULP grande o bastante para que ela continue valendo a pena. Como a duração dos ensaios não é fixada de antemão e o pior caso é justamente o *off-nominal*, as duas medidas foram mantidas em conjunto.

A segunda etapa é o **Zero-Crossing**, e o defeito é o mesmo: o tempo decorrido dentro de cada meio ciclo era obtido por acumulação do período de amostragem, $T_{sc} \leftarrow T_{sc} + T_s$ — outra soma repetida de uma parcela pequena. A acumulação foi substituída por um **contador inteiro de amostras** multiplicado pelo período, $T_{sc} = n_{zc}\,T_s$: uma única multiplicação, cujo erro é o de um único arredondamento, independentemente da duração do intervalo. A Tabela 7 compara as duas formas.

Tabela 7 – Erro no cálculo do intervalo de meio ciclo em 32 bits, para $n_{zc} = 128$ amostras (60 Hz), e erro eficaz resultante em f_{zc} na faixa de 55 Hz a 65 Hz.

Forma de cálculo	**Erro em T_{sc}**	**ULP**	**Erro eficaz em f_{zc}**
Acumulação, $T_{sc} \leftarrow T_{sc} + T_s$	$-5{,}2 \times 10^{-9}$ s	5,5	$3{,}7 \times 10^{-5}$ Hz
Produto, $T_{sc} = n_{zc}\,T_s$	$4{,}4 \times 10^{-10}$ s	0,47	$4{,}6 \times 10^{-6}$ Hz

Melhorou porque a acumulação soma 128 arredondamentos enviesados no mesmo sentido (a própria representação de $T_s = 65{,}104\,\mu$s erra $3{,}4 \times 10^{-12}$ s a cada parcela), ao passo que o produto arredonda uma única vez. O fator de 8 no erro eficaz de f_{zc}, contudo, subestima o que está em jogo: na forma acumulada o erro *médio* tem praticamente a mesma magnitude do valor eficaz, isto é, trata-se de **viés**, e não de ruído. Um desvio sistemático desse tipo atravessa o filtro de Kalman sem atenuação (o filtro modela a medida como não polarizada) e reapareceria integralmente no erro de frequência (FE) e no erro da taxa de variação (RFE) avaliados no Capítulo 5.

Ambas as decisões são específicas da implementação embarcada: no modelo Python, em dupla precisão, nenhuma delas seria necessária.

O desenvolvimento segue, então, o caminho: cada bloco da metodologia (pré-filtro SOS, Zero-Crossing, filtro de Kalman, pré-filtro B-Spline, Farrow no `proc_interp`; organização polifásica, decimação e IDFT no `proc_banco`) é primeiro validado no modelo Python e depois traduzido para o código executado pelo processador SAPHO. A verificação é feita por simulação: um sinal de teste gerado no Python é exportado, processado na simulação dos processadores e relido no Python, comparando-se o resultado embarcado com o do modelo de referência. Esse procedimento fecha o laço entre o protótipo e a implementação e sustenta a análise de resultados do capítulo seguinte.

4.7 Conclusões Parciais

Este capítulo tratou da implementação da metodologia do Capítulo 3 na plataforma SAPHO, partindo de um modelo de referência em Python e chegando aos dois processadores *soft-core* validados por simulação do Verilog gerado. A síntese e o embarque em FPGA permanecem como trabalho futuro; a ênfase, aqui, recaiu sobre a integração da cadeia e sobre as decisões que viabilizam sua operação dentro do tempo de processamento disponível.

A primeira decisão estruturante foi a partição da cadeia em dois processadores, o `proc_interp` e o `proc_banco`, que reflete a diferença de natureza entre suas etapas: o interpolador opera amostra a amostra, à taxa cheia, enquanto o banco executa a filtragem polifásica e a IDFT em bloco, apenas uma vez a cada M amostras. Essa mesma partição impôs os dois mecanismos que compõem a interface entre os núcleos: um par de memórias FIFO, que retém as amostras produzidas pelo interpolador enquanto o banco está ocupado, e uma máquina de estados finita, que observa a ocupação das filas e o estado do banco para disparar cada rodada sem condição de corrida. A escolha $M = N_{ppc}$ é o que amarra as duas pontas, fazendo com que o interpolador entregue exatamente um ciclo por bloco e que o canal de índice h da IDFT corresponda à harmônica de ordem h.

O alinhamento temporal entre o sinal e a frequência que o corrige mostrou-se um ponto central. Os atrasos introduzidos pelos filtros da cadeia foram identificados e compensados por um valor fixo de amostras, aproximação que a fase linear do pré-filtro Bessel e do banco Flat-Top justifica; distinguiu-se, além disso, a compensação de atraso do descarte do transitório do filtro de Kalman, três ordens de grandeza maior e dimensionado não pela soma dos atrasos, mas pela convergência da estimativa de frequência.

A segunda classe de decisões nasceu da aritmética de 32 bits do núcleo. Duas etapas foram reformuladas para conviver com a resolução limitada: o filtro de Kalman passou a acumular o desvio Δf em vez da frequência absoluta, com somatório compensado de Kahan, e o Zero-Crossing substituiu a acumulação do período por um produto entre contador inteiro e período de amostragem, eliminando um viés que atravessaria o filtro sem atenuação. Ambas são específicas do alvo embarcado e desnecessárias no modelo em dupla precisão.

Cabe registrar, por fim, a principal limitação da arquitetura atual. As duas filas são escritas em cadências distintas, e o descompasso entre elas pode, em execução suficientemente longa, levar ao transbordamento da fila alimentada mais rapidamente; em cenários práticos, nos quais a frequência oscila em torno da nominal, esse desbalanceamento tende a alternar de sentido e a postergar o problema, mas não a eliminá-lo. Para a duração dos ensaios do capítulo seguinte o dimensionamento adotado é suficiente e não houve perda de amostras, o que preserva a validade dos resultados; a equalização das taxas de escrita das duas filas permanece como o principal aperfeiçoamento a ser feito. Consolidada a implementação, o Capítulo 5 avalia seu desempenho frente ao modelo de referência e quantifica o quanto as limitações da plataforma afetam a estimação nas ordens harmônicas mais elevadas.

5 Resultados

5.1 Modelo dos sinais de teste

Os sinais utilizados para avaliar a metodologia são inspirados nos ensaios da norma IEC/IEEE 60255-118-1, que estabelece critérios para PMUs na estimação da componente fundamental, frequência e ROCOF (IEC, 2018). Como ainda não há uma norma equivalente consolidada para fasores harmônicos, os ensaios normativos são utilizados como referência metodológica para avaliar o comportamento do estimador quando o sinal contém componentes harmônicas.

De forma geral, considera-se um sinal composto pela componente fundamental e por componentes harmônicas, escrito como

$$x(t) = A_1(t)\cos\left(\theta_1(t)\right) + \sum_{h\in\mathcal{H}} A_h(t)\cos\left(h\theta_1(t) + \varphi_h\right) + \eta(t), \tag{5.1}$$

em que $A_1(t)$ é a amplitude da componente fundamental, $A_h(t)$ é a amplitude da componente harmônica de ordem h, $\theta_1(t)$ é a fase instantânea da fundamental, φ_h é a fase inicial da componente harmônica e $\eta(t)$ representa ruído aditivo. O conjunto $\mathcal{H}$ contém as ordens harmônicas avaliadas. Nos ensaios considerados neste trabalho, utilizou-se $\mathcal{H} = \{3, 4, \ldots, 50\}$, ou seja, sinais com harmônicos até a 50^{a} ordem, sem a componente de 2^{a} ordem.

Assim, diferentes cenários de teste podem ser descritos por diferentes escolhas de $A_1(t)$, $A_h(t)$ e $f(t)$. Em condição *off-nominal*, a frequência é constante, porém diferente da nominal. Em rampa de frequência, $f(t)$ varia linearmente no tempo. Nos ensaios de modulação de amplitude e modulação de fase, a amplitude ou a fase são perturbadas por sinais senoidais de baixa frequência. Essas condições permitem avaliar se a cadeia de estimação mantém coerência temporal quando o sinal deixa de ser estritamente estacionário.

No domínio discreto, o sinal amostrado é representado por

$$x[n] = x(nT_s), \tag{5.2}$$

em que $T_s = 1/F_s$ é o período de amostragem. Na implementação de referência em software, utiliza-se $F_s = f_0 N_{ppc}$, sendo f_0 a frequência nominal e N_{ppc} o número nominal de pontos por ciclo. Essa escolha facilita a conexão entre o interpolador e o banco de filtros, pois o fator de decimação do banco polifásico pode ser associado ao número nominal de amostras por ciclo.

Alguns parâmetros são comuns a todos os ensaios e não são repetidos adiante. A amplitude da fundamental é $X_m = 1$ pu e as fases iniciais são nulas, $\varphi_h = 0$. Cada componente harmônica do conjunto $\mathcal{H}$ é gerada com $A_h = 0{,}05$, isto é, 5 % da fundamental,

valor abaixo do limite de 10 % por harmônico previsto na Tabela 2 da norma para a classe M. Os ensaios foram executados sem ruído aditivo, $\eta(t) = 0$, de modo a isolar o efeito da variação de frequência sobre a cadeia de estimação. A frequência de amostragem é $F_s = 15\,360\,\mathrm{Hz}$, com $N_{ppc} = 256$. É a taxa de reporte que amarra o fator de decimação do banco: a decimação por $M = N_{ppc}$ entrega 60 quadros por segundo, uma das taxas normalizadas na Tabela 1 da norma para sistemas de 60 Hz (IEC, 2018).

A Tabela 8 resume os ensaios efetivamente executados, a equação que descreve cada sinal e os parâmetros adotados em cada um.

Tabela 8 – Ensaios realizados, equação do sinal de teste e respectivos parâmetros.

Ensaio	**Sinal**	**Parâmetros**	**Duração**
Frequência *off-nominal*	Eq. (5.3)	f_1 de 55 Hz a 65 Hz, passo de 1 Hz	280 *frames* (4,7 s)
Rampa de frequência	Eq. (5.5)	$R_f = 1{,}0\,\mathrm{Hz\,s^{-1}}$, 55 Hz $\rightarrow$ 65 Hz	600 *frames* (10 s)
Modulação de amplitude	Eq. (5.7)	$k_x = 0{,}1$; $k_a = 0$; $f_m = 0{,}5$ Hz e 5 Hz	280 *frames* (4,7 s)
Modulação de fase	Eq. (5.7)	$k_x = 0$; $k_a = 0{,}1$; $f_m = 0{,}5$ Hz e 5 Hz	280 *frames* (4,7 s)

5.2 Avaliação segundo a norma

Em todos os ensaios, as figuras apresentam duas curvas para cada métrica: o resultado do **modelo Python** de referência e o resultado do **SAPHO**. Cada curva é a média sobre todos os *frames* da janela analisada — 4,7 s nos ensaios de regime permanente e de modulação, 10 s na rampa — e a faixa sombreada se estende dessa média até o valor máximo observado. Registrar a duração não é um detalhe: como se verá na Seção 5.3, o erro de fase do SAPHO cresce ao longo do tempo, de modo que qualquer estatística depende do comprimento do trecho avaliado.

5.2.1 Regime *off-nominal*

O ensaio de frequência *off-nominal* corresponde ao ensaio de regime permanente da norma (*signal frequency test*), cujo sinal de entrada é dado pela Equação (12) de (IEC, 2018). Acrescentando o conteúdo harmônico adotado neste trabalho, o sinal aplicado é

$$x(t) = X_m \cos(2\pi f_1 t) + \sum_{h \in \mathcal{H}} A_h \cos(2\pi h f_1 t)\,, \tag{5.3}$$

em que f_1 é constante, porém diferente da nominal. Para os instantes de reporte $t = nT$, as Equações (21) a (24) da norma estabelecem as grandezas de referência

$$X_h(nT) = \frac{A_h}{\sqrt{2}} \angle\, 2\pi h \Delta f\, nT, \qquad f(nT) = f_0 + \Delta f, \qquad ROCOF(nT) = 0, \tag{5.4}$$

com $\Delta f = f_1 - f_0$. Varreu-se a faixa de 55 Hz a 65 Hz em passos de 1 Hz, excluída a nominal, totalizando dez ensaios. Os extremos correspondem a $\Delta f = \pm 5$ Hz, que é

exatamente o limite da faixa de classe M da Tabela 2 da norma (±5 Hz para taxas de reporte iguais ou superiores a 25 quadros por segundo), à qual corresponde um TVE máximo de 1 %. A duração de 5 s é o mínimo exigido na Subseção 6.3 da norma para ensaios de regime permanente.

A Figura 17 resume a campanha inteira: para cada métrica, a média das dez frequências e o quanto o pior caso entre elas se afasta dessa média. O erro de magnitude é praticamente idêntico entre o modelo de referência e a implementação embarcada, chegando a 0,47 % na 50ª ordem em ambos. O TVE médio também praticamente coincide: na 50ª ordem é de 0,50 % no SAPHO e 0,49 % no modelo Python, e mesmo o pior *frame* não passa de 0,78 %, de modo que nenhuma das duas implementações se aproxima do limite de 1 % em toda a faixa avaliada. A pequena diferença que resta entre as duas implementações está no erro de fase: 0,071° contra 0,057° na 50ª ordem.

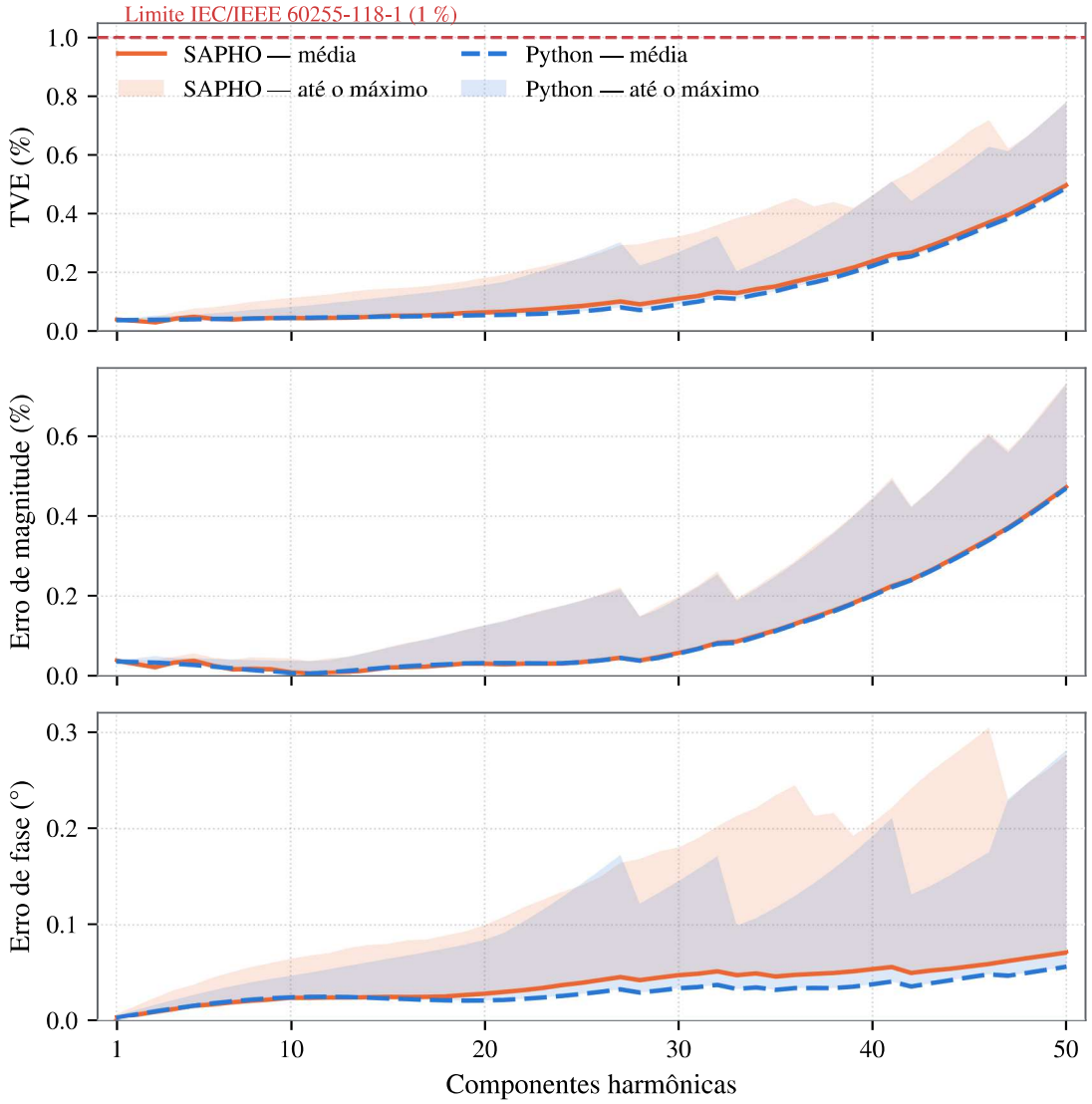


Figura 17 – Erros de magnitude, de fase e TVE por ordem harmônica: média e pior caso entre as dez frequências *off-nominal* ensaiadas.

A Figura 18 abre esse envelope e preserva a identidade de cada ensaio. Na 50ª ordem, o TVE do SAPHO vai de 0,34 %, em 55 Hz, a 0,67 %, em 65 Hz, e todas as dez curvas permanecem dentro da faixa do modelo Python, que fica entre 0,34 % e 0,64 % na 50ª ordem; em nenhuma frequência o limite de 1 % é atingido. O que a figura torna visível é que o erro **não acompanha** $|\Delta f|$: o TVE na 50ª ordem cresce de forma monótona de 55 Hz para 65 Hz, e 61 Hz apresenta erro maior que 55 Hz, ainda que esteja mais próxima da nominal. Isso já indica que o erro nas ordens elevadas não decorre simplesmente do afastamento em relação à frequência nominal.

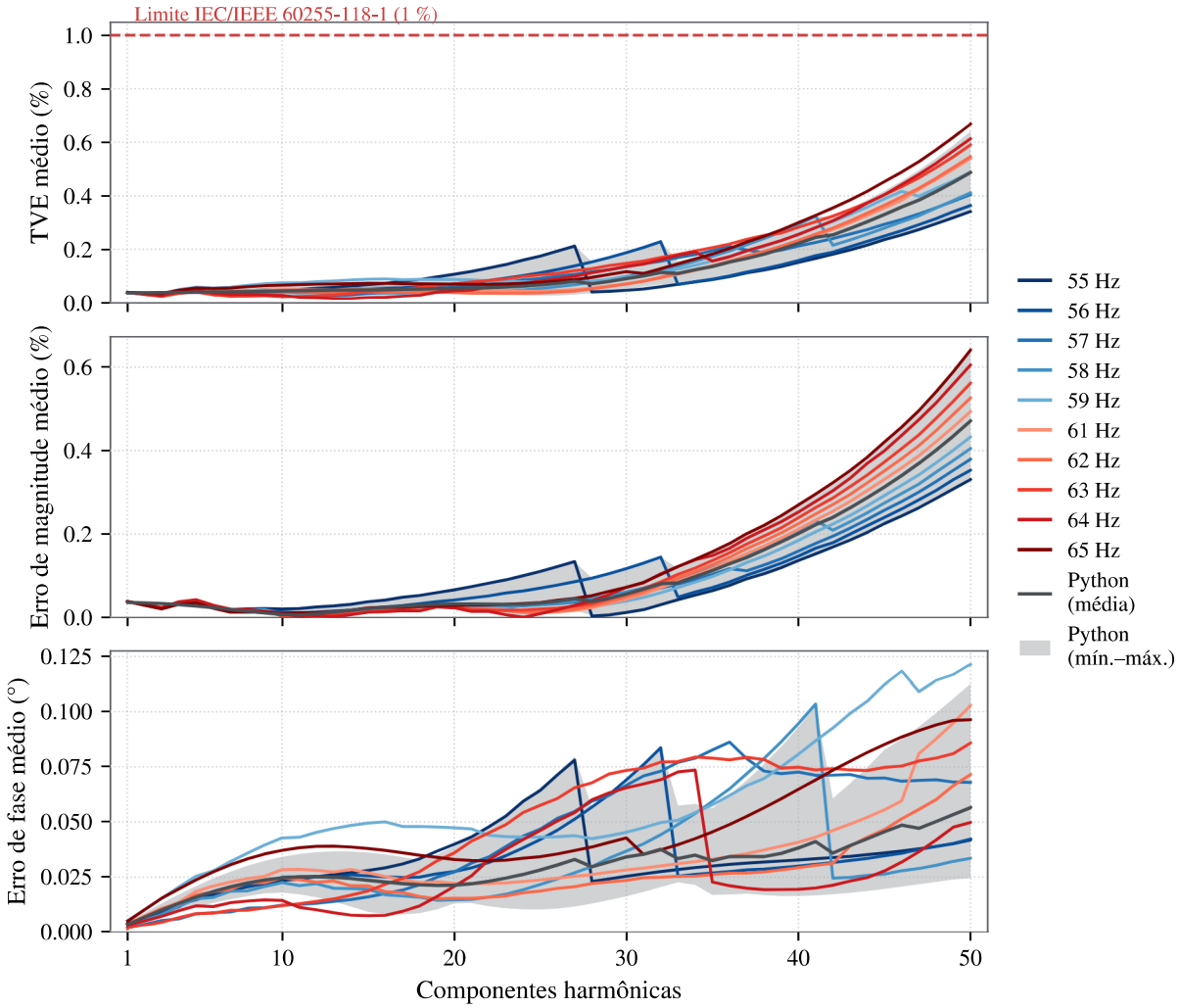


Figura 18 – Erro médio por ordem harmônica em cada uma das dez frequências *off-nominal*, com a faixa correspondente ao modelo Python ao fundo.

A Figura 19 localiza a origem dessa pequena diferença, comparando a saída do SAPHO com a do modelo Python etapa a etapa ao longo da cadeia. Como as três grandezas têm unidades diferentes, cada uma é normalizada por si mesma e apresentada como diferença relativa. As saídas do Zero-Crossing e do filtro de Kalman diferem do modelo de referência em $6{,}1 \times 10^{-6}$ e $1{,}2 \times 10^{-7}$, respectivamente, e essas diferenças permanecem **constantes** ao longo de todo o ensaio, o que confirma que a etapa de

estimação de frequência não introduz erro acumulativo. Já a saída do interpolador B-spline parte de $2{,}3 \times 10^{-5}$ e chega a $4{,}0 \times 10^{-4}$, um crescimento de cerca de 17 vezes em 5 s. A explicação está na natureza do bloco: o interpolador **integra** a diferença de frequência para formar a base de tempo da reamostragem, de modo que mesmo a diferença de 1×10^{-7} na saída do Kalman se acumula num pequeno deslocamento de tempo ao longo do ensaio. Trata-se de um deslocamento no **tempo**, e não de um erro de amplitude. Por isso o TVE da fundamental permanece em torno de 0,04 % apesar da diferença acumulada.

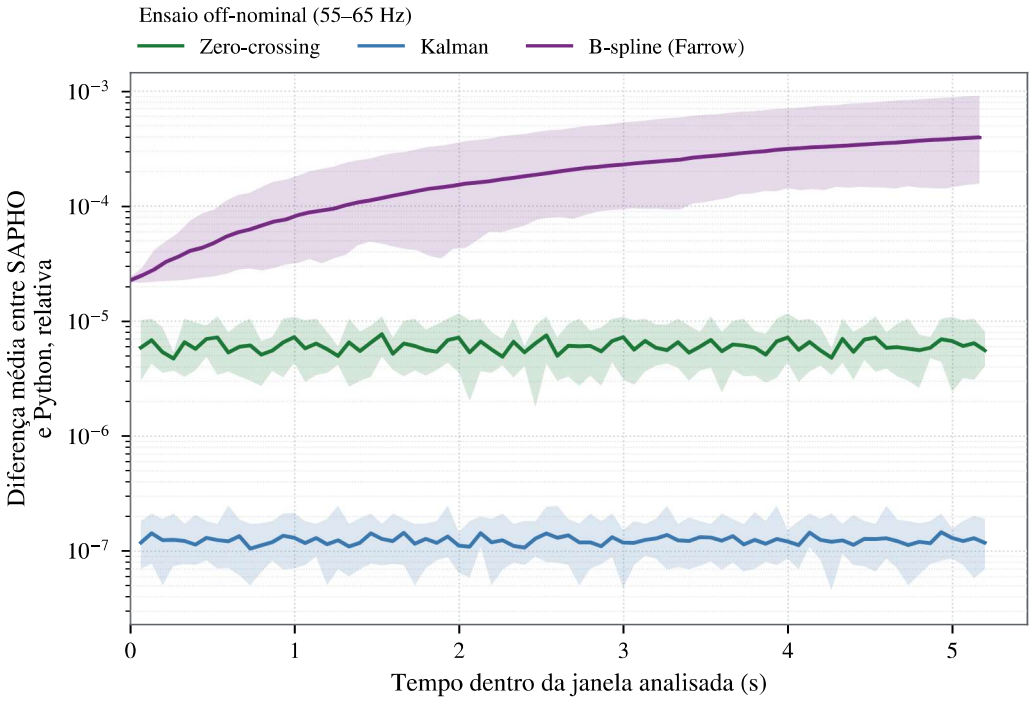


Figura 19 – Divergência relativa entre o SAPHO e o modelo Python em cada etapa da cadeia, ao longo da janela analisada, no ensaio *off-nominal*.

5.2.2 Rampa de frequência

No ensaio de rampa, a fase instantânea recebe um termo quadrático, o que faz a frequência variar linearmente no tempo. Seguindo a Equação (33) de (IEC, 2018) e acrescentando o conteúdo harmônico, o sinal aplicado é

$$x(t) = X_m \cos\left(2\pi f_a t + \pi R_f t^2\right) + \sum_{h \in \mathcal{H}} A_h \cos\left[h\left(2\pi f_a t + \pi R_f t^2\right)\right], \tag{5.5}$$

cujas grandezas de referência, pelas Equações (38) a (40) da norma, são

$$f(t) = f_a + R_f t, \qquad ROCOF(t) = R_f. \tag{5.6}$$

Adotou-se $R_f = 1{,}0\,\mathrm{Hz\,s^{-1}}$, exatamente a taxa de rampa prescrita na Tabela 6 da norma. A geração parte de $f_a = 35\,\mathrm{Hz}$, mas os primeiros 1200 ciclos, ou 20 s, são

descartados para a convergência do filtro de Kalman, conforme a Tabela 3 do Capítulo 4. A janela efetivamente avaliada começa, portanto, em $f_a + R_f \cdot 20\,\mathrm{s} = 55\,\mathrm{Hz}$ e percorre os 10 Hz seguintes até 65 Hz, que é a faixa de ± 5 Hz da classe M, com TVE máximo de 1 %.

O ensaio de rampa é o mais severo dos três. A Figura 20 mostra que, nesse caso, **o próprio modelo Python** ultrapassa o limite de 1 % de TVE a partir da 47ª ordem, atingindo 1,12 % na 50ª. A implementação embarcada acompanha o modelo de perto, cruzando o limite na mesma ordem e chegando a 1,10 %. O erro de magnitude, por sua vez, continua praticamente idêntico nas duas implementações, em torno de 0,47 % na 50ª ordem. Ou seja, sob rampa a limitação não é apenas de implementação: a própria metodologia perde precisão nas ordens elevadas, porque a frequência varia continuamente ao longo da janela de observação do banco.

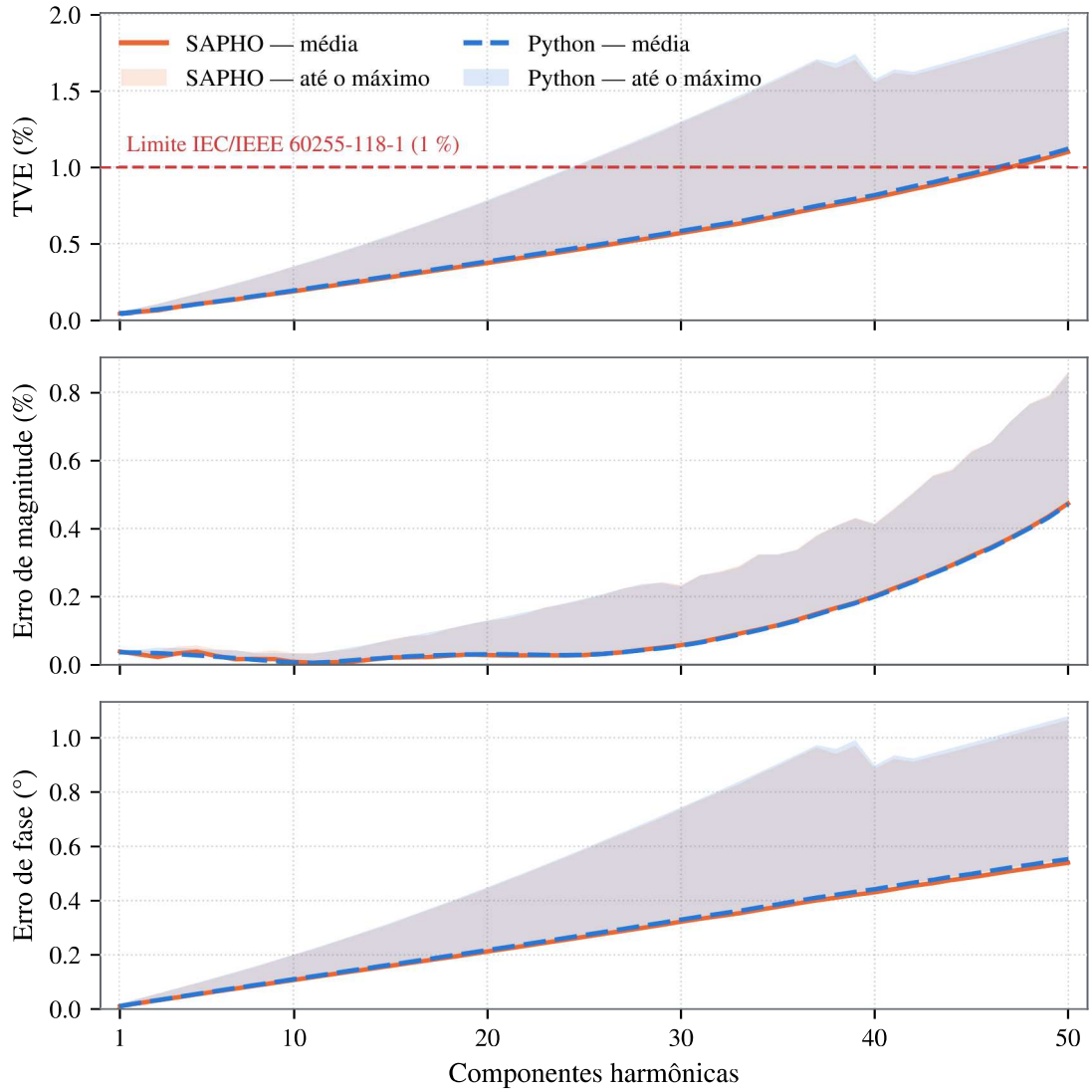


Figura 20 – Erros de magnitude, de fase e TVE por ordem harmônica, para rampa de frequência de 55 Hz a 65 Hz.

A Figura 21 esclarece esse ponto. A divergência do Zero-Crossing e a do filtro

de Kalman permanecem planas ao longo dos 10 s do ensaio, em torno de 6×10^{-6} e $1{,}3 \times 10^{-7}$ relativos, ao passo que o erro do B-spline **vai aumentando** durante todo o registro, de $2{,}2 \times 10^{-5}$ a $9{,}7 \times 10^{-4}$, cerca de duas vezes e meia o valor alcançado na campanha *off-nominal*. É justamente esse contraste que mostra como a ordem de grandeza dos erros das etapas anteriores interfere no resultado final: a frequência estimada pelo Zero-Crossing e suavizada pelo Kalman é a mesma que realimenta a reamostragem e, portanto, o integrador que forma a base de tempo do interpolador. Um erro pequeno, mas persistente, nessas etapas é somado ao longo de todo o ensaio, e é essa soma que aparece acumulada na saída do B-spline.

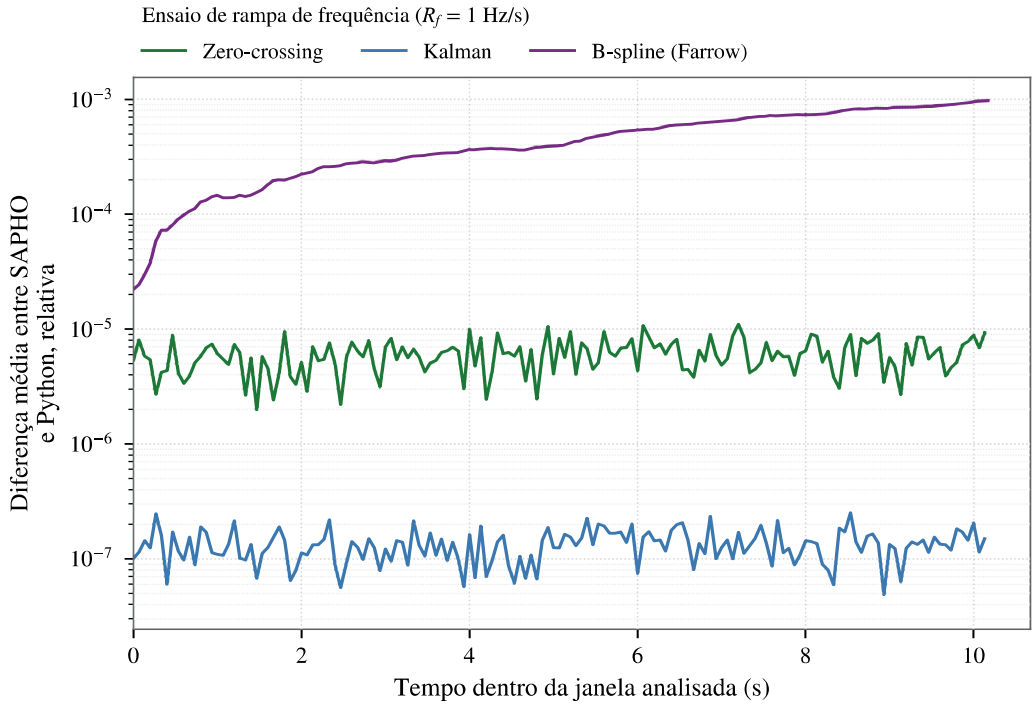


Figura 21 – Divergência relativa entre o SAPHO e o modelo Python em cada etapa da cadeia, ao longo da janela analisada, no ensaio de rampa.

O efeito do integrador fica explícito na Figura 22, que compara a correção de fase de cada implementação com a correção ideal (a integral da frequência verdadeira de referência) na 50ª ordem. O erro acumulado atinge cerca de $-27{,}5°$ no modelo Python e $-27{,}2°$ no SAPHO, com as duas curvas praticamente sobrepostas e a forma de parábola que se espera da integral de um desvio de frequência que varia linearmente. Cabe a ressalva de que essa é a *parcela* do erro devida à correção de fase, e não o erro total: o interpolador reamostra usando $\hat{f}$ e a correção desfaz essa reamostragem usando o *mesmo* $\hat{f}$, de modo que as duas parcelas entram com sinais contrários e se cancelam em boa parte.

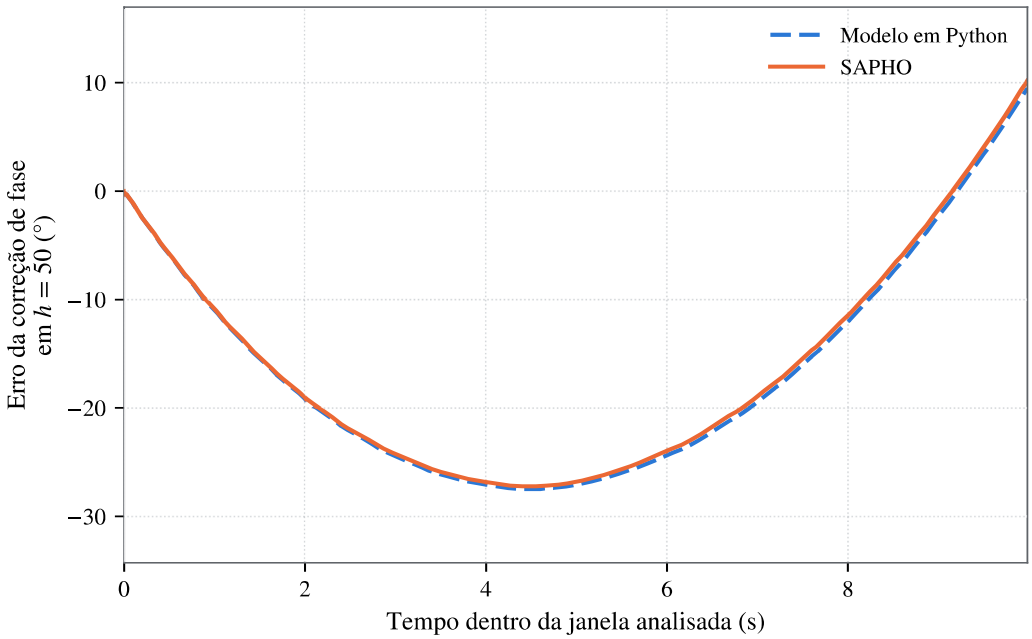


Figura 22 – Erro da correção de fase na 50ª ordem harmônica, em relação à correção ideal, ao longo do ensaio de rampa.

Há ainda um aspecto que distingue a rampa dos demais ensaios. O vetor de estado do filtro de Kalman, na Equação (3.5), reúne a frequência e a sua taxa de variação, $[f_0 + R_f T_s n,\, R_f]^\top$. Na rampa, $R_f = 1{,}0\,\mathrm{Hz\,s^{-1}}$; em todos os outros ensaios $R_f = 0$ e a segunda componente do estado é nula. A rampa é, portanto, a única condição em que o estimador opera com esse segundo estado permanentemente excitado e é essa mesma frequência que realimenta a reamostragem e o integrador da correção de fase, o que contribui para agravar o quadro descrito acima.

5.2.3 Modulação

Os ensaios de modulação correspondem ao ensaio de largura de banda de medição da norma, cujo sinal de sequência positiva é dado pelas Equações (25) e (28) de (IEC, 2018):

$$x(t) = X_m \left[1 + k_x \cos\left(2\pi f_m t\right)\right] \cos\left[2\pi f_0 t + k_a \cos\left(2\pi f_m t - \pi\right)\right], \tag{5.7}$$

em que f_m é a frequência de modulação, k_x é o fator de modulação de amplitude e k_a o fator de modulação de fase. As grandezas de referência de frequência e ROCOF são dadas pelas Equações (30) e (32) da norma:

$$f(t) = f_0 - k_a f_m \sin\left(2\pi f_m t - \pi\right), \qquad ROCOF(t) = -k_a \left(2\pi f_m^2\right) \cos\left(2\pi f_m t - \pi\right). \tag{5.8}$$

A Tabela 4 da norma prevê dois níveis de modulação, aplicados separadamente: $k_x = 0{,}1$ com $k_a = 0$, que corresponde à modulação de amplitude, e $k_x = 0$ com $k_a = 0{,}1$,

que corresponde à modulação de fase. Os dois níveis foram ensaiados, cada um com duas frequências de modulação. As componentes harmônicas foram geradas com a mesma modulação de fase da fundamental, e não com a fase multiplicada pela ordem h.

Avaliaram-se duas frequências de modulação, $f_m = 0{,}5\,\text{Hz}$ e $f_m = 5\,\text{Hz}$. O segundo valor corresponde ao limite superior da faixa de classe M: a norma o define como o menor entre um quinto da taxa de reporte e 5 Hz, o que resulta em 5 Hz para os 60 quadros por segundo adotados. O limite de TVE associado a esse ensaio é de 3 %.

Na modulação de fase, apresentada nas Figuras 23 e 24, o estimador apresenta bom desempenho em ambas as frequências.

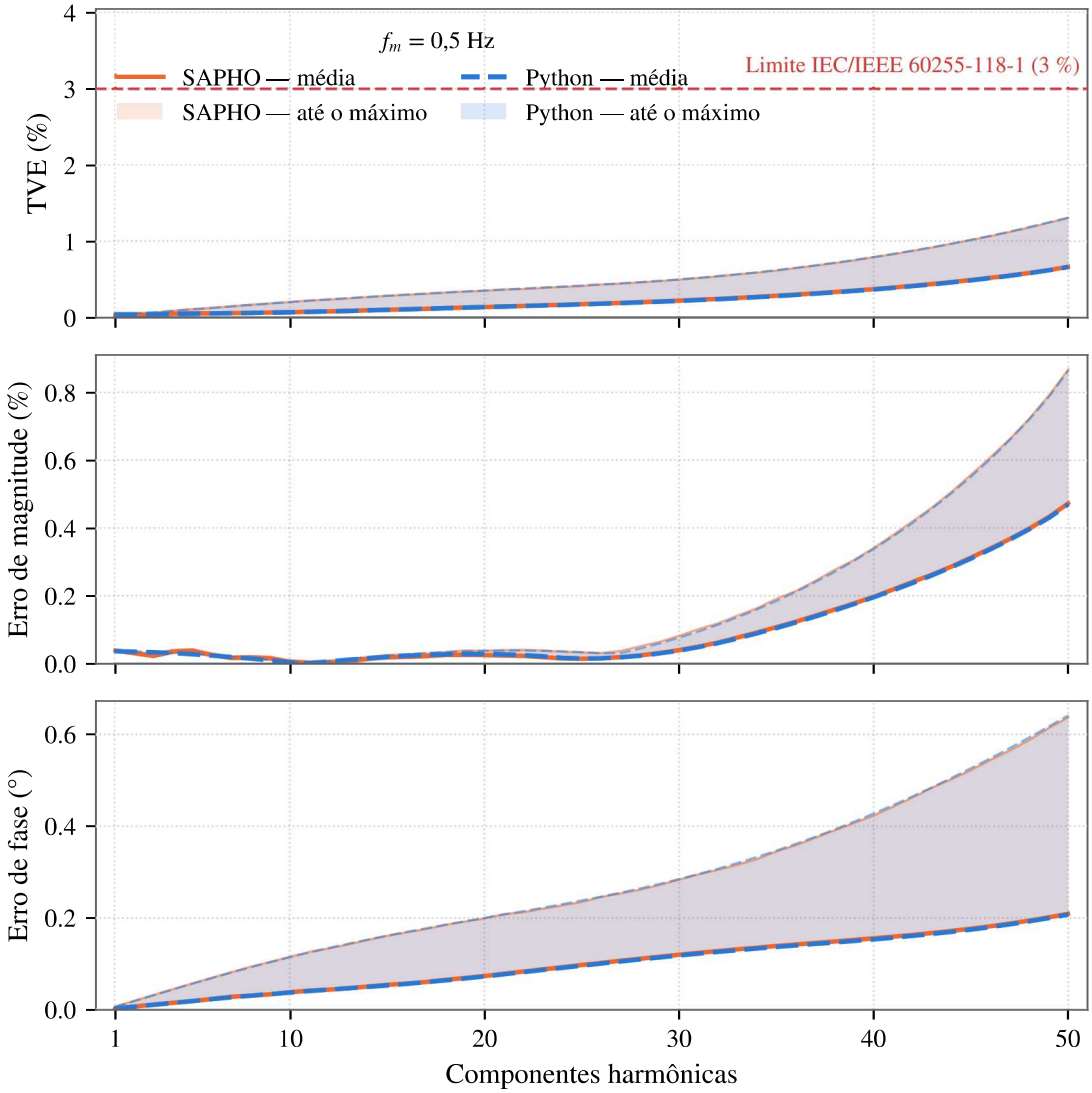


Figura 23 – Erros de magnitude, de fase e TVE por ordem harmônica, para modulação de fase com $f_m = 0{,}5\,\text{Hz}$.

Para $f_m = 0{,}5\,\text{Hz}$, o TVE médio na 50ª ordem é de 0,66 % tanto no SAPHO quanto no modelo Python. Para $f_m = 5\,\text{Hz}$, com a perturbação dez vezes mais rápida, esse valor sobe para 1,45 % nas duas implementações — bem abaixo do limite de 3 %. O crescimento

do erro em ambas as implementações, com o pior *frame* aproximando-se do limite nas últimas ordens, é coerente com o observado na rampa: quanto mais rápida a variação do sinal dentro da janela de observação, maior o erro nas ordens superiores. O erro de magnitude, por sua vez, permanece idêntico entre as duas implementações, como nos demais ensaios.

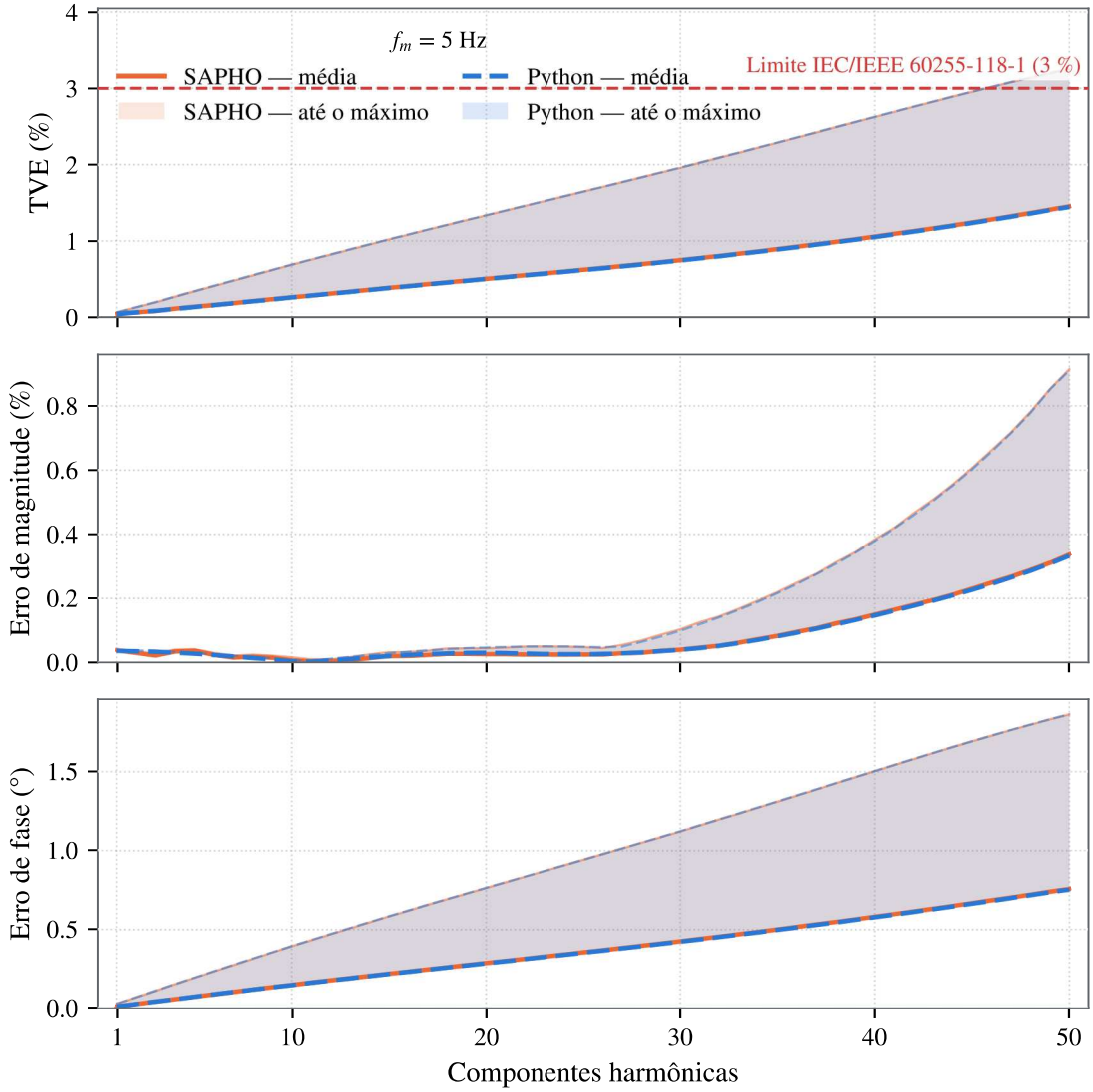


Figura 24 – Erros de magnitude, de fase e TVE por ordem harmônica, para modulação de fase com $f_m = 5\,\text{Hz}$.

Na modulação de amplitude, Figuras 25 e 26, a margem é ainda maior: o TVE médio na 50ª ordem fica em 0,10 % e 0,33 % no SAPHO, contra 0,11 % e 0,31 % no modelo Python, para $f_m = 0{,}5\,\text{Hz}$ e $f_m = 5\,\text{Hz}$ respectivamente. O erro de magnitude é o único ponto em que as duas implementações deixam de coincidir exatamente: ele apresenta uma ondulação característica ao longo das ordens, e as curvas do SAPHO e do modelo se afastam ligeiramente em alguns trechos. A diferença, porém, é inferior a 0,01 % e o erro absoluto não passa de 0,06 % em toda a faixa, o que não prejudica o resultado e o TVE continua dominado pelo erro de fase.

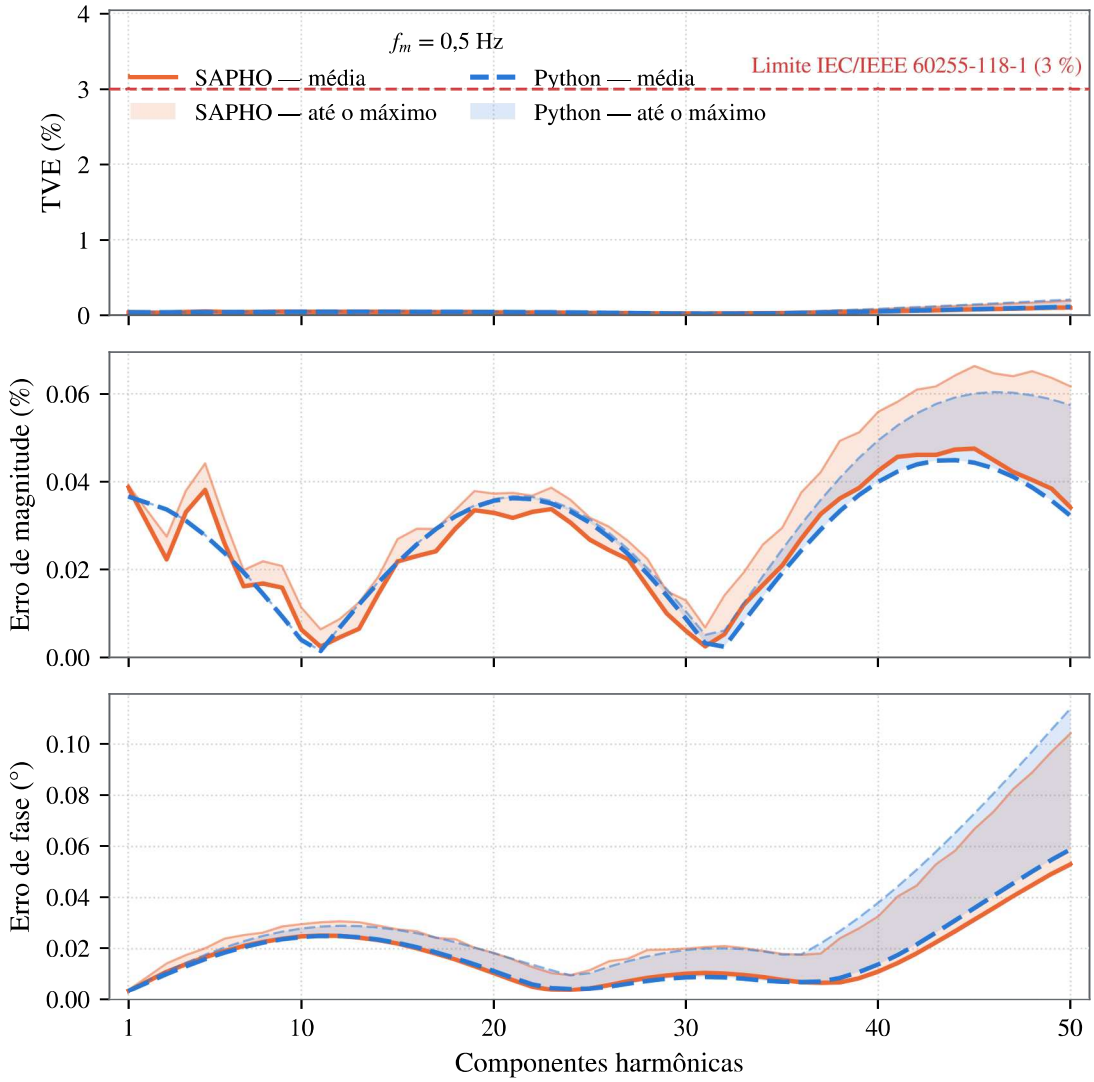


Figura 25 – Erros de magnitude, de fase e TVE por ordem harmônica, para modulação de amplitude com $f_m = 0{,}5\,\text{Hz}$.

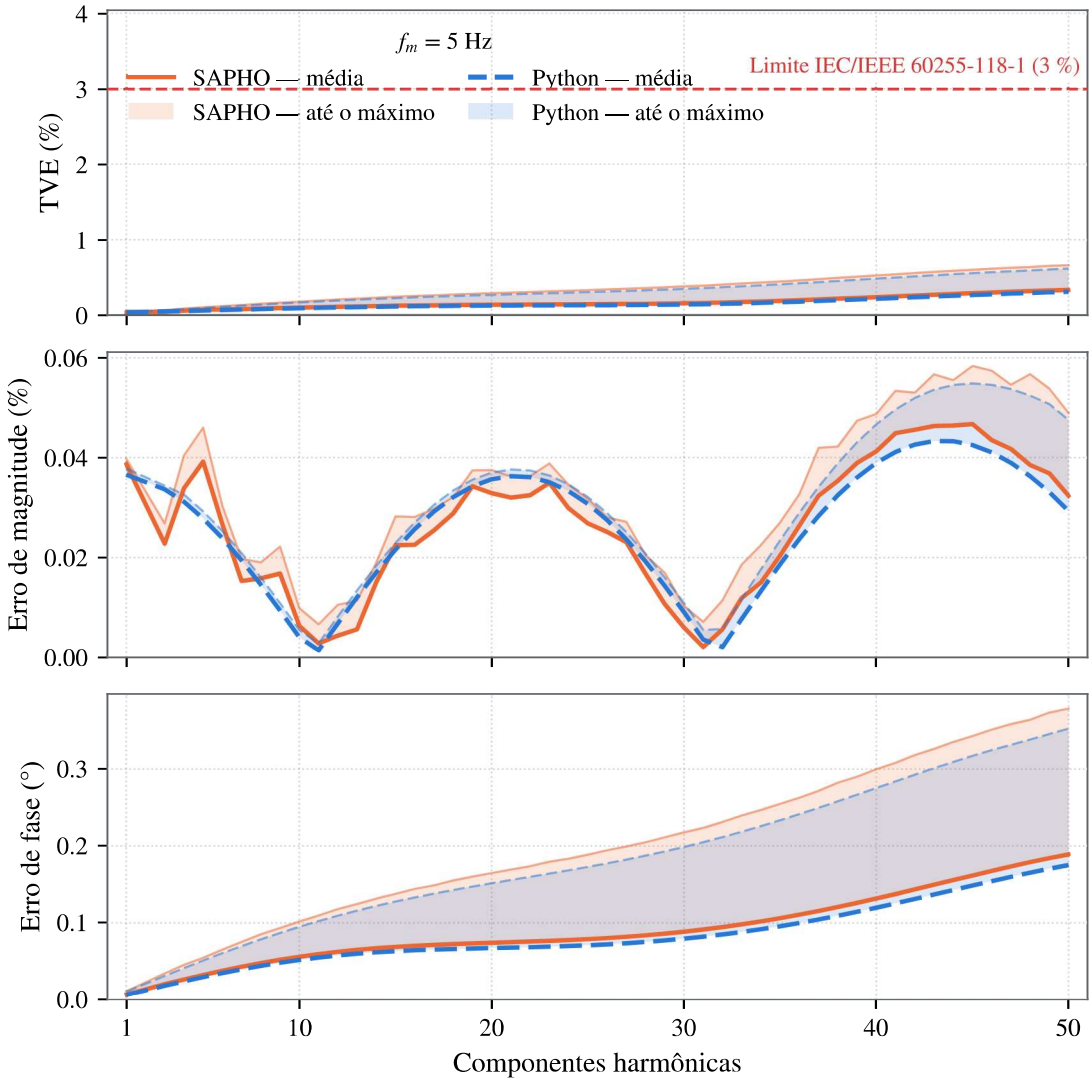


Figura 26 – Erros de magnitude, de fase e TVE por ordem harmônica, para modulação de amplitude com $f_m = 5\,\text{Hz}$.

A análise por estágio, nas Figuras 27 e 28, repete a assinatura dos ensaios anteriores: Zero-Crossing e Kalman permanecem planos ao longo do registro, enquanto o erro do B-spline cresce continuamente, até cerca de 5×10^{-4}. Esse valor é da mesma ordem do obtido na campanha *off-nominal* e cerca de metade do da rampa, o que é coerente com a folga observada nas duas modulações.

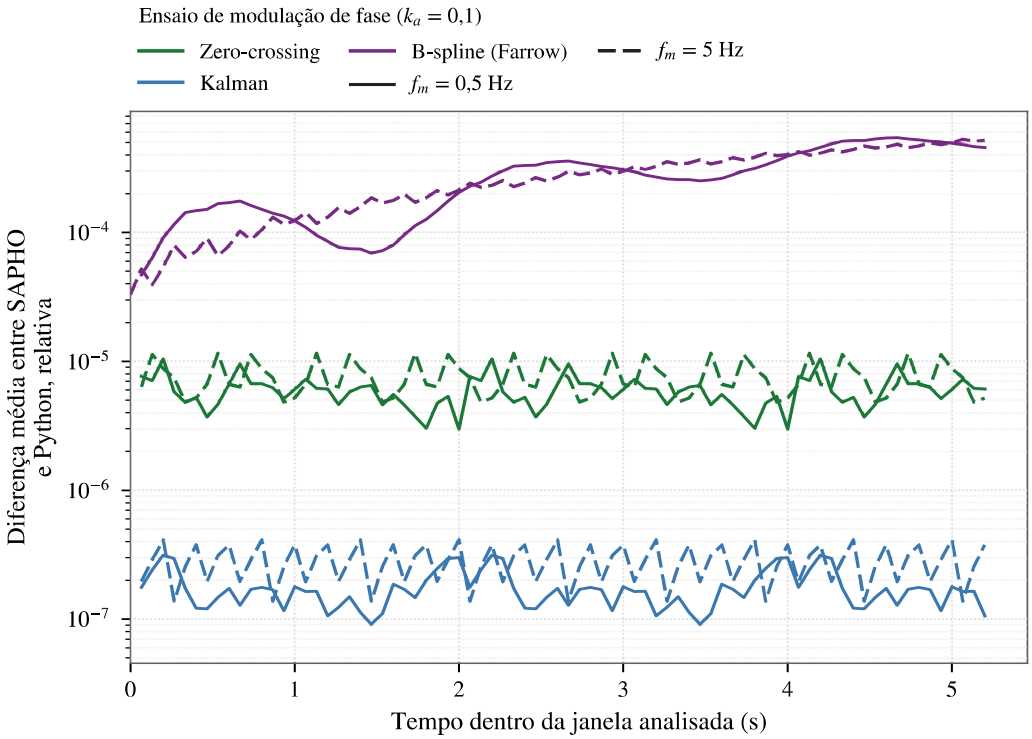


Figura 27 – Divergência relativa entre o SAPHO e o modelo Python em cada etapa da cadeia, nos ensaios de modulação de fase.

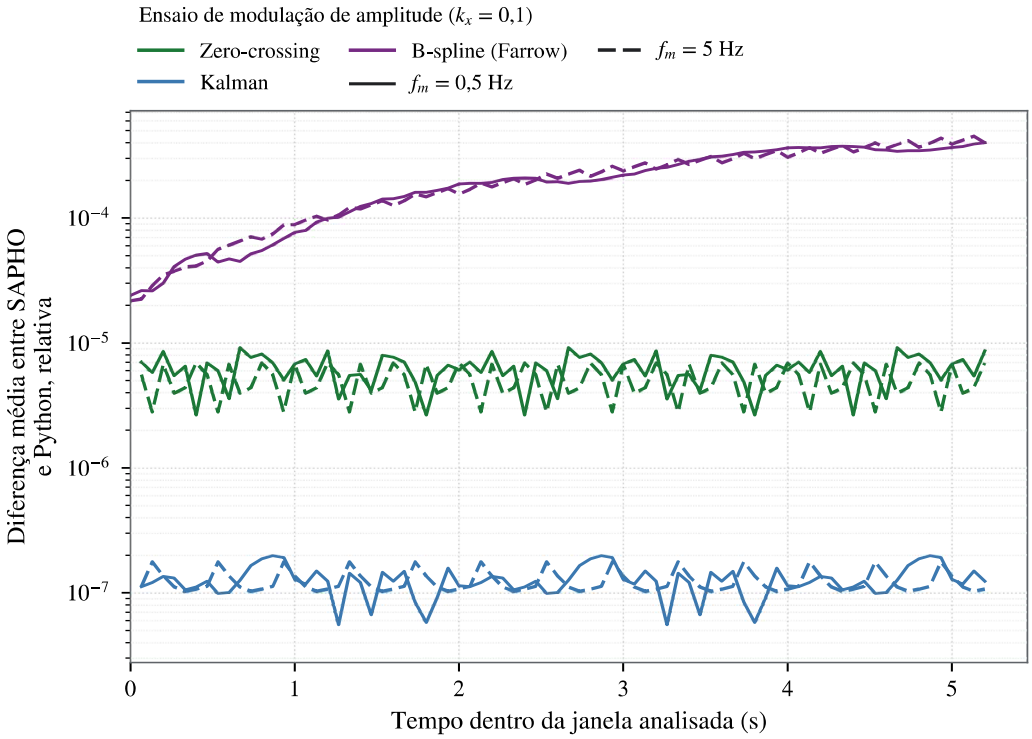


Figura 28 – Divergência relativa entre o SAPHO e o modelo Python em cada etapa da cadeia, nos ensaios de modulação de amplitude.

5.3 Comparação entre o modelo Python e a implementação SAPHO

Os ensaios da seção anterior revelam um padrão consistente, que merece análise à parte. Em todos os casos, o **erro de magnitude** do modelo Python e o da implementação embarcada praticamente coincidem, e o **TVE** do SAPHO acompanha de perto o de referência; a pequena diferença que resta concentra-se no **erro de fase** e tende a aumentar com a ordem harmônica.

Essa distinção é informativa. Se a origem da divergência fosse a quantização de amplitude ou a precisão limitada da aritmética do banco, o efeito apareceria primeiro no erro de magnitude e seria aproximadamente uniforme entre as ordens. Um erro que cresce proporcionalmente a h tem outra assinatura: é o comportamento característico de um **desalinhamento temporal**. Um desvio Δt entre o sinal reamostrado e a base de tempo à qual ele é referido produz um erro de fase $2\pi h f_1 \Delta t$, ou seja, proporcional à ordem harmônica; e como o TVE, para erro de magnitude desprezível, é aproximadamente igual ao erro de fase em radianos, o TVE herda essa proporcionalidade.

A comparação etapa a etapa entre o SAPHO e o modelo Python, apresentada nas Figuras 19, 21, 27 e 28, permite localizar onde ao longo da cadeia esse desalinhamento se forma. O quadro é o mesmo nos três ensaios:

- **Zero-Crossing.** A diferença relativa em relação ao modelo de referência é de aproximadamente 6×10^{-6} e permanece *constante* ao longo de todo o registro, sem tendência de crescimento.
- **Filtro de Kalman.** A diferença cai para a faixa de 1×10^{-7} a 3×10^{-7}, também sem tendência de crescimento. A suavização, portanto, não introduz erro acumulativo; ao contrário, reduz em mais de uma ordem de grandeza a divergência que recebe do Zero-Crossing.
- **Interpolador B-spline.** Aqui a diferença deixa de ser constante e passa a *crescer monotonicamente* durante todo o ensaio: cerca de 17 vezes na campanha *off-nominal* e cerca de 44 vezes na rampa, atingindo $9{,}7 \times 10^{-4}$.

A conclusão é direta: a divergência não nasce em nenhuma das etapas de estimação de frequência, cujos erros são pequenos e estacionários, mas se **acumula na reamostragem**. O interpolador integra a diferença de frequência para construir a sua base de tempo, de modo que um erro constante na frequência estimada se converte em um deslocamento temporal que cresce linearmente com a duração do ensaio e é esse deslocamento que produz um erro de fase proporcional a h, e portanto um TVE proporcional a h.

Vale registrar, por fim, que uma parcela da divergência é irredutível: o modelo de referência opera em precisão dupla, com mantissa de 52 bits, contra os 23 bits do

núcleo embarcado. Essa diferença estabelece um piso para a concordância entre as duas implementações e não deve ser confundida com erro de projeto. E é exatamente nesse piso que a frequência estimada pelo SAPHO se encontra: ajustando a inclinação do erro de fase em todas as ordens harmônicas, obtém-se um erro de frequência efetivo de 1,2 µHz, cerca de 0,17 ULP do formato de 23 bits, cujo valor a 60 Hz é de 7,2 µHz. O que ainda diferencia as ordens elevadas, portanto, não é a magnitude desse resíduo, que já está abaixo do limite da aritmética disponível, mas o fato de ele ser *integrado* pelo interpolador: a deriva do erro de fase na 50ª ordem é de até $0{,}044\,^\circ\,\mathrm{s}^{-1}$ no SAPHO contra $0{,}002\,^\circ\,\mathrm{s}^{-1}$ no modelo em precisão dupla. É um aspecto que admite melhoria direta na arquitetura.

5.4 Conclusões Parciais

Este capítulo avaliou a cadeia de estimação de fasores harmônicos sob os ensaios inspirados na IEC/IEEE 60255-118-1, comparando, para cada métrica, o modelo Python de referência com a implementação no SAPHO. Os ensaios cobriram regime *off-nominal*, rampa de frequência e modulação, e em todos eles a estimação de frequência mostrou-se precisa, contida em $\pm 2 \times 10^{-4}$ Hz no regime *off-nominal* e com erro sempre inferior a 1 mHz no rastreamento da rampa, o que afasta a etapa de frequência como origem dos erros observados nas ordens harmônicas.

O padrão que atravessa todos os ensaios é claro: o erro de magnitude e o TVE do modelo de referência e da implementação embarcada praticamente coincidem, restando apenas uma pequena diferença no erro de fase, que cresce com a ordem harmônica. A comparação etapa a etapa localiza a causa: as divergências do Zero-Crossing e do filtro de Kalman são pequenas e permanecem constantes ao longo dos ensaios, ao passo que a do interpolador B-spline cresce monotonicamente do início ao fim do registro. O erro não nasce, portanto, na estimação de frequência nem na aritmética do banco de filtros: ele se *acumula* na reamostragem, que integra a diferença de frequência para formar a sua base de tempo, convertendo um resíduo constante em um desalinhamento temporal crescente e daí em um erro de fase proporcional a h.

Cabe distinguir os dois regimes de limitação. No ensaio *off-nominal*, tanto a metodologia quanto a implementação embarcada permanecem dentro do limite de 1 % de TVE em toda a faixa avaliada, com TVE médio de cerca de 0,5 % na 50ª ordem. Já no ensaio de rampa, o mais severo, o próprio modelo de referência ultrapassa o limite a partir da 47ª ordem, o que mostra que, sob variação contínua de frequência, a perda de precisão nas altas ordens é intrínseca à metodologia, e não um artefato do alvo embarcado. Os ensaios de modulação, por sua vez, foram aprovados com folga nos dois níveis previstos pela norma, com TVE médio sempre abaixo de 2 % contra o limite de 3 %, e confirmam a mesma leitura: quanto mais rápida a variação do sinal dentro da janela de observação do banco, maior o erro nas ordens superiores.

Por fim, registrou-se que parte da divergência entre modelo e implementação é irredutível, decorrente da diferença de precisão numérica entre a dupla precisão do modelo Python e o formato de 32 bits do núcleo. Os resultados indicam, contudo, que a frequência estimada pelo SAPHO já se encontra nesse piso (o erro de frequência efetivo é de cerca de 0,17 ULP do formato de 23 bits), e que o fator determinante é a integração desse resíduo pelo interpolador. É um aspecto que admite melhoria direta na arquitetura, seja pela preservação da resolução da frequência ao longo de toda a cadeia, na linha do que já motivou o filtro de Kalman a operar sobre o desvio Δf (Seção 4.6), seja pela reancoragem periódica da base de tempo do interpolador, impedindo que o resíduo seja integrado indefinidamente.

6 Conclusões

Esta dissertação investigou uma metodologia de estimação de fasores harmônicos para sinais com frequência fundamental variável e a levou de um modelo de referência em software a uma implementação em processadores *soft-core* na plataforma SAPHO, validada por simulação funcional. A proposta integra, em uma única cadeia de processamento, três elementos que a literatura costuma tratar separadamente: o rastreamento da frequência fundamental por cruzamento por zero, a reamostragem dinâmica por interpolação B-Spline em estrutura de Farrow e a extração harmônica por banco de filtros Flat-Top em forma polifásica. A ideia que articula esses blocos é adaptar a base temporal do sinal antes da análise espectral, de modo que o banco opere sempre com o número nominal de pontos por ciclo e que o desalinhamento provocado por frequência fora da nominal, cujo efeito se amplifica com a ordem harmônica, seja contido antes da extração dos fasores.

No plano metodológico, a fundamentação e a formulação da cadeia mostraram por que a estimação de fasores harmônicos é mais sensível ao erro de frequência e ao desalinhamento temporal do que a estimação da componente fundamental: a fase da componente de ordem h evolui h vezes mais rápido, de modo que uma mesma incerteza temporal se traduz em erro de fase proporcional a h. Essa observação orientou tanto a escolha das técnicas quanto a etapa de correção de fase proporcional à ordem harmônica, que compensa o desvio acumulado entre a frequência estimada e a nominal.

No plano da implementação, que constitui o foco do trabalho, a cadeia foi particionada em dois processadores *soft-core* conectados por uma estrutura produtor–consumidor: o processador do interpolador, que opera amostra a amostra à taxa cheia, e o processador do banco, que executa a filtragem polifásica e a IDFT em bloco a cada M amostras. A escolha $M = N_{ppc}$ amarra os dois núcleos, fazendo com que cada bloco processado corresponda a um ciclo da fundamental e que o canal de índice h da IDFT recaia sobre a harmônica de ordem h. A coordenação entre os núcleos, por memórias FIFO e por uma máquina de estados de sincronismo, e o orçamento de atrasos e descartes necessários ao alinhamento temporal foram detalhados e implementados. Duas reformulações impostas pela aritmética de 32 bits do núcleo mereceram destaque: a operação do filtro de Kalman sobre o desvio Δf com somatório compensado, e a substituição da acumulação do período por um produto no estimador de cruzamento por zero, ambas voltadas a preservar resolução numérica em variáveis que, de outro modo, propagariam erro amplificado para as ordens harmônicas mais altas.

A avaliação seguiu ensaios inspirados na IEC/IEEE 60255-118-1: regime *off-nominal*, rampa de frequência e modulação, comparando o modelo Python de referência com as variantes embarcadas. Em todos os ensaios, a estimação de frequência mostrou-se precisa, o que afastou essa etapa como origem dos erros observados. O resultado mais

informativo foi o padrão comum a todos os cenários: o erro de magnitude do modelo e o da implementação praticamente coincidem, enquanto o TVE da variante que recebe a frequência pela fila se separa da referência com crescimento aproximadamente linear na ordem harmônica. Essa assinatura, característica de um desalinhamento temporal e não de quantização de amplitude, localizou a causa dominante não na aritmética do banco, mas no caminho percorrido pela frequência estimada até a correção de fase. Distinguiram-se, ainda, dois regimes de limitação: no ensaio *off-nominal*, a metodologia permanece dentro do limite de 1 % de TVE em toda a faixa avaliada, e o cruzamento desse limite ocorre apenas na implementação embarcada e nas ordens mais elevadas; já sob rampa, o próprio modelo de referência ultrapassa o limite nas altas ordens, evidenciando que, sob variação contínua de frequência, a perda de precisão é intrínseca à metodologia, e não um artefato do alvo embarcado.

Em conjunto, os resultados sustentam que a metodologia proposta é viável em plataforma embarcada e atende aos critérios normativos de referência nas ordens de interesse sob condições estacionárias, e delimitam com precisão onde estão suas fronteiras de exatidão: nas ordens harmônicas mais elevadas e sob variação rápida de frequência, com o alinhamento temporal entre o sinal e a frequência que o corrige como o fator determinante e, diretamente aperfeiçoável, da concordância entre modelo e implementação.

6.1 Trabalhos futuros

O desenvolvimento aqui apresentado abre várias frentes de continuidade, organizadas a seguir da consolidação do *hardware* aos aprimoramentos da própria metodologia.

- **Síntese e validação em FPGA.** A validação conduzida neste trabalho é feita por simulação funcional do Verilog gerado. O passo natural seguinte é a síntese em FPGA, com análise de fechamento de temporização e de utilização de recursos, seguida de validação experimental do estimador com sinais reais ou injetados por fonte controlada.
- **Correção do desalinhamento temporal na interface.** Como os resultados apontam o caminho da frequência estimada até o banco como fonte dominante de erro, o aperfeiçoamento de maior impacto é a reestruturação dessa interface: escrever a frequência na mesma cadência do sinal ou substituir o par de FIFOs por uma única fila de registros compostos, eliminando o desbalanceamento entre os dois fluxos e o transbordamento que dele decorre em execução prolongada. Junto a isso, cabe preservar a resolução do desvio Δf ao longo de toda a fila — evitando o deslocamento de escala que hoje a devolve à ordem de grandeza da frequência absoluta — e substituir o valor decimado de frequência por uma média sobre o bloco, atenuando o erro proporcional à taxa de variação sob rampa e modulação.

- **Robustez a ruído e a inter-harmônicos.** Os ensaios deste trabalho foram conduzidos sem ruído aditivo, para isolar o efeito da variação de frequência. A avaliação da cadeia sob ruído, distorção e componentes inter-harmônicas é necessária para caracterizar seu desempenho em condições de campo, com atenção particular ao estimador de frequência por cruzamento por zero, cuja detecção é sensível a essas perturbações.
- **Mitigação da perda de precisão sob variação rápida.** Como a limitação sob rampa e modulação mostrou-se intrínseca à metodologia, investigar estratégias que a atenuem, por exemplo, a modelagem da variação de frequência dentro da janela de observação do banco, ou o uso de um estimador de frequência de ordem superior, é uma via para estender a faixa de ordens harmônicas estimadas com exatidão sob condições dinâmicas.
- **Ampliação dos ensaios normativos.** A avaliação pode ser estendida aos demais ensaios previstos na referência normativa e a taxas de reporte distintas, bem como à comparação quantitativa com outras técnicas de estimação de fasores harmônicos embarcadas, situando o método frente ao estado da arte em termos de exatidão e custo computacional.

REFERÊNCIAS

ABOUTANIOS, E.; MULGREW, B. Iterative frequency estimation by interpolation on fourier coefficients. **IEEE Transactions on Signal Processing**, 2010.

ALEIXO, R. R.; LOMAR, T. S.; SILVA, L. R. M.; MONTEIRO, H. L. M.; DUQUE, C. A. Real-time b-spline interpolation for harmonic phasor estimation in power systems. **IEEE Transactions on Instrumentation and Measurement**, v. 71, p. 1–9, 2022.

BORKOWSKI, D.; BIEŃ, A. Improvement of accuracy of power system spectral analysis by coherent resampling. **IEEE Transactions on Power Delivery**, v. 24, n. 3, p. 1004–1013, 2009.

CHEN, K.; KEZUNOVIC, M.; LIAO, Y. An improved synchrophasor estimation algorithm for harmonic phasors. **IEEE Transactions on Power Delivery**, v. 29, n. 6, p. 2505–2513, 2014.

CHEN, L.; ZHAO, W.; WANG, Q.; WANG, F.; HUANG, S. Dynamic harmonic synchrophasor estimator based on sinc interpolation functions. **IEEE Transactions on Instrumentation and Measurement**, v. 68, n. 9, p. 3054–3065, 2018.

DUDA, K.; ZIELIŃSKI, T. P.; BARCZENTEWICZ, S. H. Perfectly flat-top and equiripple flat-top cosine windows. **IEEE Transactions on Instrumentation and Measurement**, v. 65, n. 7, p. 1558–1567, 2016.

DUDA, K.; ZIELIŃSKI, T. P.; BIEŃ, A.; BARCZENTEWICZ, S. H. Harmonic phasor estimation with flat-top fir filter. **IEEE Transactions on Instrumentation and Measurement**, v. 69, n. 5, p. 2039–2047, 2020.

GHAFARI, C.; ALMASALMA, H.; RAISON, B.; HADJSAID, N.; CAIRE, R.; MARTIN, E. Phasors estimation at off-nominal frequencies through an enhanced-sva method with a fixed sampling clock. **IEEE Transactions on Power Delivery**, v. 32, n. 4, p. 1766–1775, 2016.

HARRIS, F. J. On the use of windows for harmonic analysis with the discrete fourier transform. **Proceedings of the IEEE**, v. 66, n. 1, p. 51–83, 1978.

IEC. **IEC 60255-118-1: Measuring relays and protection equipment – Part 118-1: Synchrophasor measurements**. Geneva, 2018.

IEEE. **IEEE Std C37.118.1-2011: IEEE Standard for Synchrophasor Measurements for Power Systems**. New York, 2011.

IEEE. **IEEE Std C37.118.1a-2014: Amendment 1: Modification of Selected Performance Requirements**. New York, 2014.

LI, W.; ZHANG, G.; CHEN, M.; ZHONG, H.; GENG, Y. Dynamic harmonic phasor estimator considering frequency deviation. **IEEE Sensors Journal**, v. 21, n. 21, p. 24453–24461, 2021.

LUIZ, M. M. **Aplicação da Técnica de Decomposição Polifásica para Redução de Esforço Computacional de Filtros Empregados na Estimação de Fasores Harmônicos**. Monografia (Qualificação de Doutorado em Engenharia Elétrica) — Universidade Federal de Juiz de Fora, Juiz de Fora, 2024.

MACII, D.; PETRI, D. *et al.* Performance analysis of interpolation-based synchrophasor estimation methods. **IEEE Transactions on Instrumentation and Measurement**, 2014.

MAHARJAN, S.; PENG, J. C.-H.; MARTINEZ, J. E. Taylor-series based sample value adjustment for synchrophasor estimation. **IEEE Transactions on Power Delivery**, 2016.

MAHARJAN, S.; PENG, J. C.-H.; MARTINEZ, J. E.; XIAO, W.; HUANG, P.-H.; KIRTLEY, J. L. Improved sample value adjustment for synchrophasor estimation at off-nominal power system conditions. **IEEE Transactions on Power Delivery**, v. 32, n. 1, p. 33–44, 2017.

MARTINS, C. H. R. **Medição fasorial sincronizada aplicada a sistemas elétricos de potência**. 2012. Dissertação ou trabalho acadêmico.

MITRA, S. K.; KUO, Y. **Digital Signal Processing: A Computer-Based Approach**. New York: McGraw-Hill Higher Education, 2006. v. 2.

PETRINOVIC, D. Causal cubic splines: Formulations, interpolation properties and implementations. **IEEE Transactions on Signal Processing**, v. 56, n. 11, p. 5442–5453, 2008.

PHADKE, A. G.; THORP, J. S. **Synchronized Phasor Measurements and Their Applications**. New York: Springer, 2008.

RIBEIRO, P.; DUQUE, C.; {Da Silveira}, P.; CERQUEIRA, A. **Power systems signal processing for smart grids**. United States: Wiley, 2013. ISBN 978-1-119-99150-2.

ROMANO, P.; PAOLONE, M. Enhanced interpolated-dft for synchrophasor estimation in fpgas: Theory, implementation, and validation of a pmu prototype. **IEEE Transactions on Instrumentation and Measurement**, v. 63, n. 12, p. 2824–2836, 2014.

ROMANO, P.; PAOLONE, M. An enhanced interpolated dft algorithm for synchrophasor estimation in real-time applications. **IEEE Transactions on Instrumentation and Measurement**, 2019.

ROMANO, P.; PAOLONE, M.; CHAU, T.; JEPPESEN, B.; AHMED, E. A high-performance, low-cost pmu prototype for distribution networks based on fpga. In: **IEEE PowerTech**. [S.l.: s.n.], 2017.

SCHOENBERG, I. J. Contributions to the problem of approximation of equidistant data by analytic functions. part b. on the problem of osculatory interpolation. a second class of analytic approximation formulae. **Quarterly of Applied Mathematics**, v. 4, n. 2, p. 112–141, 1946.

VICTOR. **Decomposição polifásica aplicada à estimação de fasores harmônicos**. Monografia (Qualificação de Doutorado em Engenharia Elétrica) — Universidade Federal de Juiz de Fora, Juiz de Fora, 2024.

APPENDIX A – DECLARAÇÃO DE USO DE INTELIGÊNCIA ARTIFICIAL GENERATIVA

Ferramentas de inteligência artificial generativa fornecidas pela OpenAI, incluindo o ChatGPT e o Codex, bem como o Claude, da Anthropic, foram utilizadas como recursos de apoio durante a preparação desta dissertação. Seu uso limitou-se às seguintes atividades:

- revisão linguística, gramatical e estilística;
- sugestões para aprimorar a clareza, a concisão e a consistência terminológica;
- auxílio na formatação em LaTeX e na organização do documento; e
- revisão auxiliar de código-fonte e de descrições técnicas.

Todas as sugestões geradas foram criticamente revisadas, editadas e validadas pelo autor antes de serem incorporadas à dissertação. Essas ferramentas não foram consideradas autoras e não substituíram a formulação humana do problema de pesquisa, o projeto e a implementação da arquitetura proposta, os procedimentos experimentais, a análise e a interpretação dos resultados, a verificação das referências ou o desenvolvimento das conclusões. O autor permanece integralmente responsável pela originalidade, pela exatidão, pela atribuição de fontes e pela integridade ética do conteúdo apresentado nesta dissertação.