

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE ENGENHARIA ELÉTRICA E ELETRÔNICA
CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Tito Bertoni Viegas

**DIMENSIONAMENTO E MODELAGEM DE TRANSFORMADORES ELÉTRICOS
MONOFÁSICOS CONSIDERANDO A SATURAÇÃO MAGNÉTICA**

Florianópolis

2026

Tito Bertoni Viegas

**DIMENSIONAMENTO E MODELAGEM DE TRANSFORMADORES ELÉTRICOS
MONOFÁSICOS CONSIDERANDO A SATURAÇÃO MAGNÉTICA**

Trabalho de Conclusão do Curso de Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal de Santa Catarina como requisito para a obtenção do título de Bacharel em Engenharia Elétrica.

Orientador: Prof. Jean Viane Leite, Dr..

Florianópolis

2026

Ficha de identificação da obra

Viegas, Tito Bertoni

Dimensionamento e modelagem de transformadores elétricos monofásicos considerando a saturação magnética / Tito Bertoni Viegas ; orientador, Jean Vianeí Leite, 2026.

62 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Centro Tecnológico, Graduação em Engenharia Elétrica, Florianópolis, 2026.

Inclui referências.

1. Engenharia Elétrica. 2. Transformadores Elétricos. Saturação Magnética. Modelagem Computacional.. I. Leite, Jean Vianeí. II. Universidade Federal de Santa Catarina. Graduação em Engenharia Elétrica. III. Título.

Tito Bertoni Viegas

**DIMENSIONAMENTO E MODELAGEM DE TRANSFORMADORES ELÉTRICOS
MONOFÁSICOS CONSIDERANDO A SATURAÇÃO MAGNÉTICA**

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de “Bacharel em Engenharia Elétrica” e aceito, em sua forma final, pelo Curso de Graduação em Engenharia Elétrica.

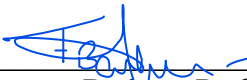
Florianópolis, 25 de março de 2026.

Prof. Roberto Francisco Coelho, Dr.
Coordenador do Curso de Graduação em Engenharia Elétrica

Banca Examinadora:

Prof. Jean Viane Leite, Dr.
Orientador
Universidade Federal de Santa Catarina

Prof. Mauricio Valencia Ferreira da Luz, Dr.
Universidade Federal de Santa Catarina


Prof. Filomena Barbosa Rodrigues Mendes, Dra.
Universidade Tecnológica Federal do Paraná

Este trabalho é dedicado aos meus colegas de classe e aos meus queridos pais.

AGRADECIMENTOS

Agradeço primeiramente à minha família, pelo apoio incondicional, pelo incentivo e por toda a compreensão durante as longas horas de estudo, pois sem o suporte e o sacrifício de vocês, a realização deste sonho e a conclusão desta etapa não seriam possíveis. Estendo meus sinceros agradecimentos ao meu orientador, Prof. Dr. Jean Viane Leite, pelas orientações, paciência e direcionamento técnico fundamentais para o desenvolvimento e conclusão deste Trabalho de Conclusão de Curso. Sou grato também aos docentes do Departamento de Engenharia Elétrica e Eletrônica da Universidade Federal de Santa Catarina (UFSC), pelo conhecimento compartilhado durante a minha formação acadêmica, bem como aos colegas de curso, pelo convívio e pela troca de experiências ao longo desses anos de estudo. Por fim, agradeço a todos que contribuíram, de maneira direta ou indireta, para a realização deste projeto.

RESUMO

Este trabalho aborda a modelagem computacional de transformadores elétricos, com foco na representação matemática da não linearidade dos materiais ferromagnéticos utilizados em seus núcleos. A eficiência, a qualidade de energia e o comportamento dinâmico destes equipamentos são diretamente influenciados pelo fenômeno da saturação magnética, o qual impõe desafios complexos à sua simulação digital. O objetivo principal desta pesquisa é desenvolver, simular e comparar modelos computacionais capazes de representar a curva de magnetização anisterética do aço silício, avaliando o impacto da saturação na forma de onda da corrente de excitação e na geração de distorções harmônicas. A metodologia adotada parte do dimensionamento paramétrico de um transformador de 60 VA, seguido da formulação de suas equações diferenciais acopladas. Para a resolução do sistema no domínio do tempo, emprega-se o método de integração numérica de Runge-Kutta de 4ª ordem (RK4) estruturado em ambiente Python. O escopo do estudo contempla a implementação de três abordagens analíticas distintas: o Modelo Linear, a Equação de Fröhlich e a função Tangente Hiperbólica. As simulações são estruturadas para avaliar o equipamento tanto em regime permanente quanto sob transientes severos, especificamente a corrente de *inrush*. Para testar os limites de validade e a robustez numérica de cada formulação, os modelos são submetidos a diferentes ângulos de chaveamento da rede e a condições extremas de sobretensão.

Palavras-chave: Transformadores Elétricos. Saturação Magnética. Modelagem Computacional. Corrente de *Inrush*. Método de Runge-Kutta.

ABSTRACT

This work addresses the computational modeling of electrical transformers, focusing on the mathematical representation of the non-linearity of ferromagnetic materials used in their cores. The efficiency, power quality, and dynamic behavior of these devices are directly influenced by the magnetic saturation phenomenon, which imposes complex challenges to their digital simulation. The main objective of this research is to develop, simulate, and compare computational models capable of representing the anhysteretic magnetization curve of silicon steel, evaluating the impact of saturation on the excitation current waveform and the generation of harmonic distortions. The adopted methodology starts with the parametric design of a 60 VA transformer, followed by the formulation of its coupled differential equations. To solve the system in the time domain, the 4th-order Runge-Kutta (RK4) numerical integration method is employed, structured in a Python environment. The scope of the study encompasses the implementation of three distinct analytical approaches: the Linear Model, the Fröhlich Equation, and the Hyperbolic Tangent function. The simulations are structured to evaluate the equipment both in steady-state and under severe transients, specifically the inrush current. To test the validity limits and numerical robustness of each formulation, the models are subjected to different grid switching angles and extreme overvoltage conditions.

Keywords: Electrical Transformers. Magnetic Saturation. Computational Modeling. Inrush Current. Runge-Kutta Method.

LISTA DE FIGURAS

Figura 1: Dimensões da lâmina padronizada	19
Figura 2: Dimensões do núcleo	20
Figura 3: Tensão de entrada com 90° de fase	30
Figura 4: Tensão de entrada com 0° de fase	31
Figura 5: Correntes no primário para 20 Vp (90°)	32
Figura 6: Curvas B x H para 20 Vp (90°)	32
Figura 7: FFTs para 20 Vp (90°)	33
Figura 8: Correntes no primário para 20 Vp (0°)	33
Figura 9: Curvas B x H para 20 Vp (0°)	34
Figura 10: FFTs para 20 Vp (0°)	34
Figura 11: Correntes no primário para 179,6 Vp (90°)	35
Figura 12: Curvas B x H para 179,6 Vp (90°)	36
Figura 13: FFTs para 179,6 Vp (90°)	36
Figura 14: Correntes no primário para 179,6 Vp (0° - primeiros ciclos)	37
Figura 15: Correntes no primário para 179,6 Vp (0° - últimos ciclos)	38
Figura 16: Curvas B x H para 179,6 Vp (0° - primeiros ciclos)	38
Figura 17: Curvas B x H para 179,6 Vp (0° - últimos ciclos)	39
Figura 18: FFTs para 179,6 Vp (0°)	39
Figura 19: Correntes no primário para 268 Vp (90°)	40
Figura 20: Curvas B x H para 268 Vp (90°)	41
Figura 21: FFTs para 268 Vp (90°)	42
Figura 22: Corrente no primário para 268 Vp (0° - primeiros ciclos)	42
Figura 23: Curvas B x H para 268 Vp (0° - primeiros ciclos)	43
Figura 24: Correntes no primário para 268 Vp (0° - últimos ciclos)	43
Figura 25: Curvas B x H para 268 Vp (0° - últimos ciclos)	44
Figura 26: FFTs para 268 Vp (0°)	44

LISTA DE TABELAS

Tabela 1: Tabela de medidas para lâminas padronizadas.	19
Tabela 2: Parâmetros do Transformador	23
Tabela 3: Parâmetros ajustados para curva de referência.	29

LISTA DE ABREVIATURAS E SIGLAS

AWG – American Wire Gauge (Padrão Americano de Bitola de Fios)

CC – Corrente Contínua

FFT – Fast Fourier Transform (Transformada Rápida de Fourier)

RK4 - Runge-Kutta de 4ª Ordem

VA – Volt-Ampère

Vp – Volts de pico

Vrms – Volts Root Mean Square (Valor Eficaz)

LISTA DE SÍMBOLOS

Letras Latinas:

- A_{fio} – Área da seção transversal do condutor
- A_g – Área geométrica bruta da seção do núcleo
- A_m – Área útil da seção magnética do núcleo
- a – Largura da coluna central da lâmina do núcleo
- B – Indução magnética instantânea
- B_{max} – Indução magnética máxima
- B_{sat} – Indução magnética de saturação teórica
- b – Altura do empilhamento magnético do núcleo
- d – Densidade de corrente elétrica
- f – Frequência da rede elétrica
- H – Intensidade de campo magnético
- h_{med} – Altura média do percurso magnético
- I_1, I_2 – Corrente elétrica eficaz (primário e secundário)
- i_1, i_2 – Corrente elétrica instantânea (primário e secundário)
- L_1, L_2 – Indutância de dispersão
- l_{espira} – Comprimento médio de uma espira
- l_m – Comprimento médio do caminho magnético
- N_1, N_2 – Número de espiras
- R_1, R_2 – Resistência elétrica ôhmica dos enrolamentos
- R_{carga} – Resistência elétrica da carga conectada ao secundário
- \mathfrak{R}_{din} – Relutância magnética dinâmica
- S – Potência aparente
- $v_1(t)$ – Tensão elétrica instantânea da fonte
- V_1, V_2 – Tensão elétrica eficaz
- w_{med} – Largura média do percurso magnético

Z_{base} – Impedância base do equipamento

Letras Gregas:

α – Parâmetro de inclinação da função Tangente Hiperbólica

β – Parâmetro de assíntota da função Tangente Hiperbólica

γ – Parâmetro linear da Equação de Fröhlich

δ – Parâmetro não linear da Equação de Fröhlich

Δt – Passo de integração temporal (algoritmo numérico)

ϕ – Fluxo magnético

μ_0 – Permeabilidade magnética do vácuo

μ_{dif} – Permeabilidade magnética diferencial

μ_{in} – Permeabilidade magnética inicial do material

μ_r – Permeabilidade magnética relativa

ω – Frequência angular da rede

ρ_{Cu} – Resistividade elétrica do material condutor (cobre)

τ – Constante de tempo do circuito elétrico (L/R)

SUMÁRIO

1 INTRODUÇÃO	15
1.1 INTRODUÇÃO GERAL	15
1.2 OBJETIVOS	16
1.2.1 Objetivo Geral	16
1.2.2 Objetivos Específicos	16
2 DESENVOLVIMENTO	17
2.1 MODELAGEM DO TRANSFORMADOR	17
2.2 MODELAGEM MATEMÁTICA DO MATERIAL DO NÚCLEO	23
2.2.1 Modelo de Fröhlich	23
2.2.2 Modelo de Tangente Hiperbólica	25
2.3 SOLUÇÃO NUMÉRICA E ALGORITMO	26
3 RESULTADOS	29
3.1 ANÁLISE EM TENSÃO ABAIXO DA NOMINAL	31
3.1.1 Conexão 20 Vp (90°)	31
3.1.2 Conexão 20 Vp (0°)	33
3.2 ANÁLISE EM TENSÃO NOMINAL	34
3.2.1 Conexão 179,6 Vp (90°)	35
3.2.2 Conexão 179,6 Vp (0°)	36
3.3 ANÁLISE EM SOBRETENSÃO	40
3.3.1 Conexão 268 Vp (90°)	40
3.3.2 Conexão 268 Vp (0°)	42
4 CONCLUSÃO	45
REFERÊNCIAS	47
APÊNDICE A – Código Fonte em Python para a Simulação Numérica	48

1 INTRODUÇÃO

1.1 INTRODUÇÃO GERAL

Os transformadores são dispositivos estáticos fundamentais nos sistemas de energia elétrica, permitindo a transferência de potência entre diferentes níveis de tensão com alta eficiência. O desempenho desses equipamentos está intrinsecamente ligado às propriedades do material ferromagnético que compõe seu núcleo. O comportamento de materiais como os aços elétricos de grão orientado (LANDGRAF, [s.d.]) é complexo e marcadamente não linear, apresentando fenômenos indissociáveis como a saturação e a histerese magnética.

A saturação magnética do núcleo, em particular, é um fenômeno crítico que ocorre quando o material se aproxima de seu limite físico de magnetização. Operar um transformador nesta região, seja por falhas no projeto ou por distúrbios na rede, resulta em uma corrente de excitação distorcida e de alta amplitude, o que compromete severamente a qualidade da energia, gera poluição harmônica e ameaça a integridade do sistema elétrico. Com a crescente exigência por estabilidade e eficiência energética, a modelagem computacional precisa desses comportamentos não lineares tornou-se uma etapa indispensável para o projeto e a análise de transformadores modernos.

Neste contexto, este trabalho de conclusão de curso dedica-se à simulação dinâmica de transformadores considerando a não linearidade do núcleo, com foco principal no efeito da saturação sob diferentes regimes de operação. Para isso, adota-se uma abordagem no domínio do tempo, fundamentada na solução de equações diferenciais acopladas, para comparar o desempenho de formulações matemáticas, como a Equação de Fröhlich e a Tangente Hiperbólica, na representação da curva de magnetização anisterética. Esta delimitação metodológica permite uma investigação focada e robusta do impacto numérico e físico da saturação na corrente de magnetização, avaliando desde a distorção harmônica em regime permanente até a severidade das correntes de *inrush* durante manobras de energização, servindo como base sólida para futuras investigações que englobem modelos dinâmicos de perdas.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Desenvolver e validar uma modelagem computacional dinâmica de transformadores elétricos, com foco na análise crítica dos efeitos da saturação magnética e na comparação de formulações matemáticas representativas do comportamento magnético do núcleo sob diferentes regimes de operação.

1.2.2 Objetivos Específicos

Para alcançar o objetivo geral, os seguintes objetivos específicos foram definidos:

- Revisar a fundamentação teórica de circuitos magnéticos não lineares e as metodologias clássicas de dimensionamento físico de transformadores.
- Estudar e comparar os principais modelos matemáticos para a representação da curva de magnetização anisterética, com ênfase na Equação de Fröhlich e na função Tangente Hiperbólica.
- Formular o sistema de equações diferenciais acopladas do transformador e implementá-lo computacionalmente utilizando o método de integração numérica de Runge-Kutta de 4ª ordem (RK4).
- Analisar o impacto da saturação na forma de onda e no espectro harmônico (FFT) da corrente de excitação em regime permanente, avaliando cenários de tensão nominal e sobretensão.
- Simular e avaliar a resposta transitória do equipamento frente a diferentes instantes de conexão à rede, evidenciando a dinâmica da corrente de *inrush* e a robustez numérica dos modelos matemáticos sob estresse extremo.

2 DESENVOLVIMENTO

Este capítulo detalha a metodologia adotada para o dimensionamento físico e a modelagem computacional do transformador objeto de estudo. Para garantir uma transição coerente entre o projeto real e a simulação numérica, o processo é estruturado em três etapas fundamentais.

O escopo inicia-se pela definição dos parâmetros físicos e geométricos do equipamento (como dimensões do núcleo e número de espiras), baseando-se em metodologias clássicas de projeto de máquinas elétricas. Em seguida, procede-se à derivação dos parâmetros do circuito elétrico e magnético equivalente, estabelecendo as grandezas necessárias para a representação analítica das não linearidades do aço silício. Por fim, apresenta-se a formulação do sistema de equações diferenciais acopladas e a estruturação do algoritmo de integração numérica, fundamentado no método de Runge-Kutta de 4ª ordem (RK4), que servirá de motor para as simulações dinâmicas discutidas no capítulo de resultados.

2.1 MODELAGEM DO TRANSFORMADOR

A modelagem computacional fidedigna de um transformador requer a determinação prévia de seus parâmetros construtivos fundamentais, seguindo metodologias consolidadas na literatura. Neste trabalho, o escopo limita-se a transformadores monofásicos de baixa potência, operando em frequência industrial, com núcleo do tipo encouraçado (shell-type), adaptando-se o roteiro de cálculo proposto por Martignoni (1991).

Como ponto de partida para o dimensionamento, definiram-se as especificações nominais do equipamento objeto de estudo. Optou-se por um transformador com potência aparente de 60 VA e relação de transformação de 127 V / 24 V, operando na frequência de 60 Hz. Para compensar as perdas térmicas e magnéticas inerentes à operação, adicionou-se uma margem de segurança de 10% na potência aparente exigida do enrolamento primário ($S_1 = 1,1 \cdot S_2$), totalizando 66 VA.

Definidas as especificações de tensão e potência, calculam-se as correntes nominais que circulam nos enrolamentos primário e secundário através das Equações (2.1) e (2.2):

$$I_1 = \frac{S_1}{V_1} = \frac{66}{127} \approx 0,52 \text{ A} \quad (2.1)$$

$$I_2 = \frac{S_2}{V_2} = \frac{60}{24} = 2,50 \text{ A} \quad (2.2)$$

Para transformadores com potências inferiores a 500 VA, Martignoni (1991) recomenda adotar uma densidade de corrente (d) nos condutores de 3 A/mm². Essa escolha baseia-se no equilíbrio entre a capacidade de dissipação térmica e o rendimento do dispositivo. A partir dessa densidade, determinam-se as seções transversais dos condutores pelas Equações (2.3) e (2.4):

$$A_{fio1} = \frac{I_1}{d} = \frac{0,52}{3} \approx 0,173 \text{ mm}^2 \quad (2.3)$$

$$A_{fio2} = \frac{I_2}{d} = \frac{2,50}{3} \approx 0,833 \text{ mm}^2 \quad (2.4)$$

Consultando a tabela de fios padronizados (AWG), selecionaram-se os condutores que melhor se aproximam dos valores calculados, sendo eles os fios AWG números 24 e 17, com seções transversais de 0,205 mm² e 1,04 mm² respectivamente. Essas escolhas resultam em densidades reais de corrente de 2,54 A/mm² e 2,4 A/mm², estando ambas dentro do limite estipulado de 3 A/mm².

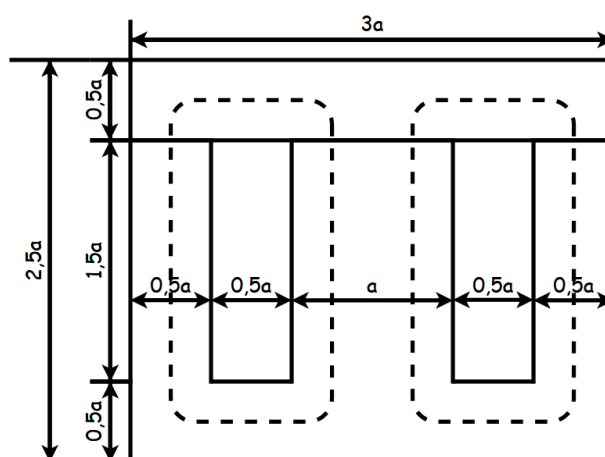
Uma vez definidas as seções transversais dos fios, prossegue-se para o dimensionamento do circuito magnético. Existe uma relação intrínseca entre o núcleo e os condutores: o dimensionamento ideal do núcleo deve resultar em uma janela de enrolamento suficiente para acomodar as bobinas. A seção magnética necessária (A_m) foi determinada pela relação empírica para transformadores de baixa potência, sendo função direta da potência aparente e da frequência, conforme a Equação (2.5):

$$A_m = 7,5 \cdot \sqrt{\frac{S_2}{f}} = 7,5 \cdot \sqrt{\frac{60}{60}} = 7,5 \text{ cm}^2 \quad (2.5)$$

Considerando um fator de empilhamento de 1,1 (devido ao isolamento entre lâminas) e uma seção de coluna central quadrada, a área geométrica bruta ($A_g = 1,1 \cdot A_m$) resulta em

8,25 cm². Para a definição das dimensões físicas, buscou-se uma seção transversal próxima à quadrada ($a \approx b$) para otimização do perímetro médio das espiras. A raiz quadrada de A_g sugere uma aresta ideal de aproximadamente 2,87 cm. Para a materialização do núcleo, optou-se pela utilização de lâminas padronizadas do tipo "E-I" (sem perdas), amplamente empregadas em transformadores de baixa potência. As proporções geométricas destas lâminas são fixas e dependem exclusivamente da largura da coluna central (a), conforme ilustrado na Figura 1. A Tabela 1 apresenta as dimensões para lâminas padrão, relacionando o número da lâmina com a largura da coluna central.

Figura 1: Dimensões da lâmina padronizada



Fonte: Elaborado pelo autor (2026) com base em Martignoni (1991).

Tabela 1: Tabela de medidas para lâminas padronizadas.

N.º	a (cm)	Seção da Janela (mm ²)	Peso do núcleo (kg/cm)
0	1,5	168	0,095
1	2	300	0,170
2	2,5	468	0,273
3	3	675	0,380
4	3,5	900	0,516
5	4	1200	0,674
6	5	1880	1,053

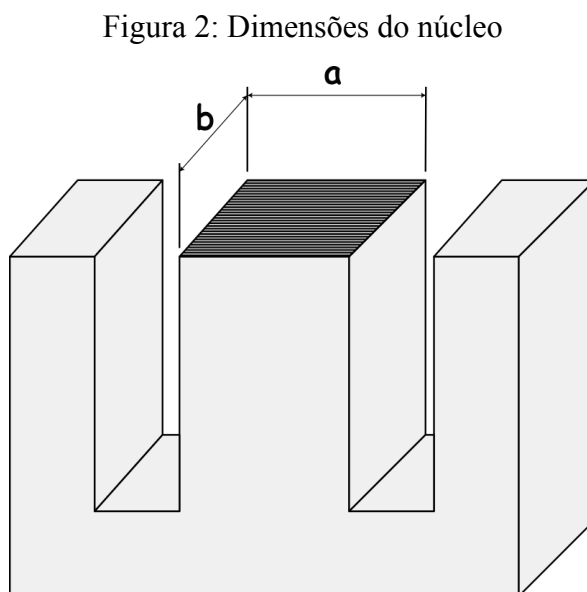
Fonte: Elaborado pelo autor (2026) com base em Martignoni (1991).

Consultando a tabela de lâminas comerciais, selecionou-se a Lâmina N° 3, que possui largura da coluna central fixada em ($a = 3,0 \text{ cm}$), valor comercial mais próximo do calculado teoricamente.

Uma vez definida a largura da lâmina (a), a espessura do empilhamento (b) necessária para atingir a área calculada é determinada pela Equação (2.6):

$$b = \frac{A_g}{a} = \frac{8,25}{3} = 2,75 \text{ cm} \quad (2.6)$$

Dessa forma, definem-se as dimensões efetivas da coluna central do núcleo para a simulação como $3,0 \text{ cm} \times 2,75 \text{ cm}$, totalizando a seção geométrica bruta de $8,25 \text{ cm}^2$. A Figura 2 ilustra a representação física das grandezas a e b na formação do pacote magnético.



Fonte: Elaborado pelo autor (2026) com base em Martignoni (1991).

Para o cálculo da relutância do núcleo na simulação, é necessário determinar o comprimento médio do caminho percorrido pelo fluxo magnético. Em transformadores do tipo encouraçado, o fluxo divide-se simetricamente entre as colunas laterais. Portanto, considera-se o perímetro médio de uma das malhas magnéticas.

Fundamentando-se nas proporções padronizadas para a Lâmina N° 3, conforme apresentado na Tabela 1, e assumindo uma altura de janela padrão de $1,5a$, determinam-se as dimensões efetivas do percurso magnético. Conforme o método aproximado estabelecido por Martignoni (1991), o comprimento do circuito magnético no ferro (l_m), observado em traçado

pontilhado na Figura 1, não é estritamente retangular. O cálculo de l_m incorpora os trechos lineares correspondentes à altura da janela e à largura da culatra, somados ao perímetro dos arcos de circunferência que representam a deflexão do campo magnético nas junções do núcleo E-I. Essa composição geométrica detalhada fornece uma estimativa mais rigorosa da impedância magnética equivalente, sendo matematicamente definida pela Equação (2.7):

$$l_m = 2 \cdot 0,5a + 2 \cdot 1,5a + 0,5a\pi = 5,6a \quad (2.7)$$

Substituindo o valor da largura da coluna central, obtém-se um caminho médio de 16,8 cm ou 0,168 m.

O número de espiras dos enrolamentos primário e secundário foi calculado de acordo com as Equações (2.8) e (2.9), fixando-se a indução magnética máxima (B_{max}) em 1,1 T, garantindo a operação na região linear do aço silício padrão.

$$N_1 = \frac{V_1 \cdot 10^8}{4,44 \cdot B_{max} \cdot A_m \cdot f} = \frac{127}{4,44 \cdot 1,1 \cdot 7,5 \times 10^{-4} \cdot 60} \approx 578 \text{ espiras} \quad (2.8)$$

$$N_2 = \frac{V_2 \cdot 10^8}{4,44 \cdot B_{max} \cdot A_m \cdot f} = \frac{24}{4,44 \cdot 1,1 \cdot 7,5 \times 10^{-4} \cdot 60} \approx 110 \text{ espiras} \quad (2.9)$$

Embora o roteiro de Martignoni (1991) defina os aspectos construtivos, a simulação dinâmica no domínio do tempo, regida pelas equações diferenciais apresentadas mais adiante neste capítulo, exige o conhecimento dos parâmetros do circuito elétrico equivalente: as resistências ôhmicas dos enrolamentos (R_1, R_2) e as indutâncias de dispersão (L_1, L_2).

Para determinar a resistência elétrica, calcula-se o comprimento total do condutor bobinado baseando-se no comprimento médio de uma espira (l_{espira}). Martignoni (1991, p. 86) fornece uma expressão empírica para este cálculo em pequenos transformadores, demonstrada na Equação (2.10):

$$l_{espira} = 2a + 2b + 0,5a\pi \quad (2.10)$$

Substituindo as dimensões do núcleo definidas obtém-se um comprimento médio de espira de 0,162 m.

A resistência de cada enrolamento é então calculada pela Segunda Lei de Ohm, utilizando a resistividade do cobre sem correção por temperatura ($\rho_{Cu} \approx 1,72 \times 10^{-8} \Omega \cdot m$) e as seções transversais (A_{fio}) dos fios AWG 24 e AWG 17 previamente selecionados de acordo com a Equação (2.11):

$$R = \rho_{Cu} \cdot \frac{N \cdot l_{espira}}{A_{fio}} \quad (2.11)$$

Aplicando os valores, obtemos resistências de 7,86 Ω e 0,29 Ω , para os fios AWG 24 e AWG 17, respectivamente.

O cálculo analítico preciso das indutâncias de dispersão requer a análise complexa da geometria do fluxo disperso na janela do transformador, o que foge ao escopo do método simplificado de Martignoni (1991). Para a modelagem dinâmica, estes valores foram estimados a partir da impedância base (Z_{base}) do equipamento, adotando-se uma reatância de dispersão percentual típica de 3% para transformadores desta classe de potência.

A impedância base do lado primário é dada pela Equação (2.12):

$$Z_{base1} = \frac{V_1^2}{S_1} = \frac{127^2}{66} \approx 244,38 \Omega \quad (2.12)$$

Assumindo $x\%=3\%$, a reatância indutiva de dispersão (X_{l1}) e a indutância correspondente (L_1) são calculadas de acordo com a Equação (2.13):

$$L_1 = \frac{Z_{base1} \cdot 0,03}{2\pi f} = \frac{244,38 \cdot 0,03}{377} \approx 19,4 \text{ mH} \quad (2.13)$$

Analogamente para o secundário obtém-se $Z_{base2} = 9,6 \Omega$ e $L_2 \approx 0,77 \text{ mH}$.

Dessa forma, os parâmetros finais consolidados para a alimentação do algoritmo numérico são apresentados na Tabela 2.

Tabela 2: Parâmetros do Transformador

Parâmetro	Valor	Unidade	Descrição
S1	66	VA	Potência aparente base de entrada
S2	60	VA	Potência aparente base de saída
V1	127	V	Tensão eficaz de entrada
V2	24	V	Tensão eficaz de saída
f	60	Hz	Frequência da rede
N1	578	espiras	Número de espiras do primário
N2	110	espiras	Número de espiras do secundário
R1	7,86	Ω	Resistência do enrolamento primário
R2	0,29	Ω	Resistência do enrolamento secundário
L1	0,0194	H	Indutância de dispersão do enrolamento primário
L2	0,00077	H	Indutância de dispersão do enrolamento secundário
Am	0,00075	m ²	Área da seção transversal útil do núcleo
lm	0,168	m	Comprimento médio do caminho magnético

Fonte: Autoria própria (2026).

2.2 MODELAGEM MATEMÁTICA DO MATERIAL DO NÚCLEO

Definidos os parâmetros físicos e elétricos do transformador na etapa de dimensionamento, o passo seguinte consiste na estruturação do modelo matemático que representará o comportamento dinâmico do componente. No entanto, esses valores constantes não são suficientes, por si sós, para descrever o comportamento não linear do transformador durante sua operação. Para capturar os fenômenos de saturação magnética e analisar as correntes de excitação no domínio do tempo, é necessário estruturar um modelo dinâmico que integre essas grandezas físicas às leis fundamentais do eletromagnetismo.

2.2.1 Modelo de Fröhlich

Um modelo escolhido para representar a característica não linear do núcleo ferromagnético foi a Equação de Fröhlich, que descreve a curva de magnetização anisterética. Esta função, apresentada na Equação (2.14), foi escolhida pela sua eficiência computacional e capacidade de aproximar a transição não linear inicial de aços elétricos.

$$B = \frac{H}{\gamma + \delta|H|} \quad , \quad (2.14)$$

Onde B é a indução magnética, H é o campo magnético, γ e δ são coeficientes a serem determinados a partir do material a ser modelado.

A determinação física dos coeficientes γ e δ fundamenta-se na análise assintótica da curva $B \times H$. Na região linear, onde o campo magnético é muito pequeno ($H \rightarrow 0$), o termo $\delta|H|$ torna-se desprezível, reduzindo a equação para $B = \frac{H}{\gamma}$. Dessa forma, conclui-se que γ representa o inverso da permeabilidade inicial do material $\left(\gamma = \frac{1}{\mu_r \cdot \mu_0}\right)$, onde μ_r é a permeabilidade magnética relativa do aço e μ_0 é a permeabilidade do vácuo.

Por outro lado, na região de saturação profunda, quando o campo tende ao infinito ($H \rightarrow \infty$), o termo γ torna-se irrelevante, simplificando a equação para $B = \frac{1}{\delta}$. Portanto, o coeficiente δ corresponde ao inverso da indução magnética de saturação máxima teórica B_{sat} $\left(\delta = \frac{1}{B_{sat}}\right)$.

Para a solução numérica do sistema, torna-se necessário o processo inverso, obter a intensidade de campo H (e conseqüentemente a corrente) a partir de uma indução magnética B conhecida. Isolando H na Equação (2.14), obtém-se a relação constitutiva inversa utilizada no algoritmo, apresentada na Equação (2.15):

$$H = \frac{\gamma \cdot B}{1 - \delta|B|} \quad . \quad (2.15)$$

Esta formulação matemática permite calcular a relutância dinâmica (R_{din}) a cada passo de tempo. Isso é feito obtendo-se a derivada da intensidade de campo em relação à indução magnética $\left(\frac{dH}{dB}\right)$ a partir da inversa da Equação de Fröhlich. Vale ressaltar que esta derivada representa o inverso da permeabilidade diferencial $\left(\frac{1}{\mu_{dif}}\right)$. Assim, à medida que o material satura e a permeabilidade decresce, o termo $\frac{dH}{dB}$ aumenta, elevando a relutância do circuito magnético conforme demonstrado na Equação (2.16):

$$\frac{dH}{dB} = \frac{1}{\mu_{dif}} = \frac{\gamma}{(1 - \delta|B|)^2} \quad . \quad (2.16)$$

2.2.2 Modelo de Tangente Hiperbólica

Como alternativa ao modelo de Fröhlich, implementou-se também o modelo baseado na função tangente hiperbólica. Este modelo é amplamente reconhecido na literatura pela modelagem de sua transição para a região de saturação. Diferentemente das formulações racionais, a sua derivada exponencial decai de forma assintótica, mantendo a permeabilidade diferencial com valores não nulos por uma faixa mais ampla de operação. Esta característica evita o colapso abrupto da derivada em saturação profunda, o que favorece a estabilidade de métodos numéricos de integração. A relação constitutiva entre a indução magnética B e a intensidade de campo H é descrita pela Equação (2.17):

$$B = \beta \cdot \tanh(\alpha H) \quad (2.17)$$

Onde β e α são parâmetros a serem determinados a partir do material a ser representado.

Nesta formulação, o parâmetro β define a assíntota horizontal da curva, correspondendo diretamente à indução magnética de saturação do material (B_{sat}). Por sua vez, o parâmetro α relaciona-se à inclinação da curva na origem. Sabendo que a permeabilidade magnética inicial (μ_{in}) é definida pela derivada da curva em $H = 0$, obtém-se a relação algébrica $\mu_{in} = \beta\alpha$. Dessa forma, os parâmetros intrínsecos do modelo podem ser determinados diretamente a partir das características físicas conhecidas do material, conforme expresso nas Equações (2.18) e (2.19):

$$\alpha = \frac{\mu_{in}}{B_{sat}} \quad (2.18)$$

$$\beta = B_{sat} \quad (2.19)$$

Para a aplicação no algoritmo de solução numérica proposto mais adiante na Equação 2.20, é necessário determinar a relutância dinâmica a cada passo de tempo. Isso

exige o cálculo da derivada $\frac{dH}{dB}$ em função da indução magnética instantânea B , que é a variável de estado conhecida na simulação.

Utilizando a identidade trigonométrica $\sec^2(x) = 1 - \tanh^2(x)$, é possível expressar a permeabilidade diferencial μ_{dif} diretamente em função da indução magnética B , eliminando a necessidade de calcular o campo H explicitamente. A dedução é apresentada a seguir pelas Equações (2.20), (2.21) e (2.22):

$$\frac{dB}{dH} = \beta\alpha \cdot \sec^2(\alpha H) = \beta\alpha \left[1 - \tanh^2(\alpha H) \right] \quad (2.20)$$

Substituindo $\tanh^2(\alpha H)$ por $\left(\frac{B}{B_{sat}}\right)^2$ e $\beta\alpha$ por μ_{in} :

$$\frac{dB}{dH} = \mu_{in} \left[1 - \left(\frac{B}{B_{sat}}\right)^2 \right] \quad (2.21)$$

Finalmente, o termo $\frac{dH}{dB}$ necessário para a atualização da relutância dinâmica na matriz do sistema é obtido pela inversão da permeabilidade diferencial:

$$\frac{dH}{dB} = \frac{1}{\mu_{in} \left[1 - \left(\frac{B}{B_{sat}}\right)^2 \right]} \quad (2.22)$$

2.3 SOLUÇÃO NUMÉRICA E ALGORITMO

O comportamento dinâmico do transformador é regido por um sistema de equações diferenciais acopladas que não possui solução analítica simples devido à variação temporal da permeabilidade. Portanto, a solução foi implementada numericamente estabelecendo-se primeiramente as leis fundamentais de cada circuito.

A modelagem dinâmica do transformador baseia-se na aplicação das Leis de Kirchhoff para os circuitos elétricos e da Lei de Faraday-Lenz para o acoplamento magnético (FITZGERALD et al., 2006). No enrolamento primário, a tensão da fonte $v_1(t)$ é equilibrada pela queda de tensão resistiva, dependente da corrente i_1 e da resistência intrínseca R_1 , pelas

tensões induzidas na indutância de dispersão L_1 e pela força eletromotriz mútua gerada pela variação temporal do fluxo magnético ϕ concatenado nas N_1 espiras, resultando na Equação (2.23):

$$-v_1(t) + R_1 \cdot i_1 + L_1 \frac{di_1}{dt} + N_1 \frac{d\phi}{dt} = 0 \quad (2.23)$$

De forma análoga, para o circuito secundário, considerando uma carga resistiva (R_{carga}), a equação de malha é dada pela Equação (2.24), onde i_2 é a corrente, R_2 é a resistência do enrolamento, L_2 é a indutância de dispersão, N_2 é o número de espiras do enrolamento secundário e ϕ é o fluxo resultante no núcleo:

$$-N_2 \frac{d\phi}{dt} + (R_2 + R_{carga}) \cdot i_2 + L_2 \frac{di_2}{dt} = 0 \quad (2.24)$$

A terceira equação de estado provém da derivada temporal da Lei de Hopkinson (FITZGERALD et al., 2006), onde a variação do fluxo é impulsionada pela soma algébrica das forças magnetomotrizes dos enrolamentos. Mantendo a consistência com as convenções de corrente adotadas nas Equações (2.23) e (2.24), tem-se a Equação (2.25):

$$\frac{d\phi}{dt} = \frac{1}{\mathfrak{R}_{din}} \cdot \left(N_1 \frac{di_1}{dt} - N_2 \frac{di_2}{dt} \right) \quad (2.25)$$

Em abordagens convencionais, este sistema seria organizado na forma matricial e resolvido via inversão numérica a cada passo de tempo. No entanto, tal método eleva o custo computacional, especialmente quando associado a integradores de múltiplo estágio como o Runge-Kutta. Visando a otimização do algoritmo e a execução em tempo real, optou-se por uma solução algébrica direta. Isolando-se os termos $\frac{di}{dt}$ das equações de malha dos enrolamentos, obtém-se as Equações (2.26) e (2.27) e substituindo-os na Equação (2.25), é possível desacoplar o sistema e obter uma expressão explícita para a derivada do fluxo magnético. Após as manipulações algébricas, chega-se à Equação (2.28) implementada no código:

$$\frac{di_1}{dt} = \frac{v_1 - R_1 \cdot i_1 - N_1 \frac{d\phi}{dt}}{L_1} \quad (2.26)$$

$$\frac{di_2}{dt} = \frac{N_2 \frac{d\phi}{dt} - (R_2 + R_{carga}) \cdot i_2}{L_2} \quad (2.27)$$

$$\frac{d\phi}{dt} = \frac{\frac{N_1}{L_1} (v_1 - R_1 \cdot i_1) - \frac{N_2}{L_2} (i_2 \cdot (R_2 + R_{carga}))}{\frac{N_1^2}{L_1} + \frac{N_2^2}{L_2} + \mathfrak{R}_{din}} \quad (2.28)$$

Uma vez determinado o valor instantâneo de $\frac{d\phi}{dt}$, as derivadas das correntes são calculadas por substituição direta nas Equações (2.26) e (2.27) apresentadas anteriormente. Esta abordagem reduz a complexidade da solução a operações aritméticas simples, eliminando a necessidade de inversão de matrizes e garantindo maior robustez numérica.

Para a resolução do sistema de equações diferenciais não lineares, empregou-se o método de integração numérica de Runge-Kutta de 4ª ordem (RK4). Este método é amplamente utilizado em simulações dinâmicas devido ao seu excelente balanço entre estabilidade e precisão computacional (CHAPRA; CANALE, 2008). Diferentemente do Método de Euler, que realiza uma extrapolação linear baseada exclusivamente na derivada do instante anterior, o método RK4 oferece maior precisão ao explorar o comportamento da função ao longo de todo o passo de integração. O algoritmo calcula a inclinação da curva em quatro pontos estratégicos dentro do mesmo intervalo de tempo (k_1, k_2, k_3, k_4) . Desta forma, o sistema é avaliado quatro vezes a cada passo e utiliza-se uma média ponderada destas inclinações para determinar a trajetória mais precisa. A atualização para o próximo instante $(k + 1)$ é feita pela Equação (2.29):

$$x_{k+1} = x_k + \frac{\Delta t}{6} (k_1 + 2k_2 + 2k_3 + k_4) \quad (2.29)$$

O método RK4 minimiza os erros de truncamento associados à integração numérica, permitindo a obtenção de soluções numéricas estáveis mesmo diante das não linearidades inerentes à saturação magnética.

3 RESULTADOS

Com a fundamentação teórica estabelecida e a implementação computacional concluída, este capítulo dedica-se à apresentação e análise dos resultados obtidos através da simulação do transformador monofásico de 60 VA. O objetivo central é validar a robustez do método numérico Runge-Kutta de 4ª ordem na solução das equações diferenciais acopladas e, subsequentemente, investigar o impacto da não linearidade do núcleo ferromagnético nas formas de onda de corrente.

As simulações foram realizadas aplicando-se os parâmetros físicos e elétricos determinados no Capítulo 2 (Tabela 1). Para garantir uma avaliação criteriosa do desempenho do modelo, a análise foi estruturada em três etapas progressivas. Inicialmente, o sistema é submetido a um regime de baixa excitação, onde se espera um comportamento predominantemente senoidal das correntes, servindo como validação base do algoritmo. Em seguida, analisa-se a operação em tensão nominal e em sobretensão, comparando-se as respostas dos modelos de saturação de Fröhlich e da Tangente Hiperbólica. Por fim, realiza-se a análise no domínio da frequência (FFT) em todos os casos para quantificar a distorção harmônica introduzida pela magnetização do núcleo.

Uma curva de magnetização inicial experimental, oriunda de um aço silício comercial, foi utilizada para servir de referência aos modelos. Os parâmetros dos modelos foram obtidos através de ajustes de curvas e são mostrados na Tabela 3.

Tabela 3: Parâmetros ajustados para curva de referência.

Modelo	Parâmetro	Valor Ajustado
Linear	μ_r	34202
Fröhlich	γ	8,378
Fröhlich	δ	0,536
Tangente Hiperbólica	α	0,0275
Tangente Hiperbólica	β	1,778

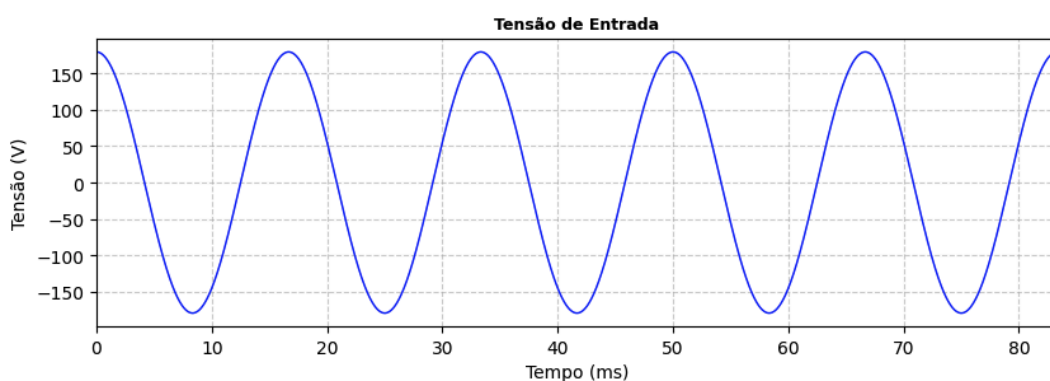
Fonte: Autoria própria (2026).

Visando isolar os efeitos da não linearidade magnética e focar exclusivamente no comportamento do núcleo, todas as simulações deste capítulo foram conduzidas considerando

o transformador operando a vazio. Computacionalmente, essa condição foi emulada conectando-se uma resistência de carga elevada, fixada em 10.000Ω , ao enrolamento secundário. Para a solução numérica das equações diferenciais, adotou-se um passo de integração temporal de 1×10^{-7} segundos. Esta resolução foi cuidadosamente escolhida para garantir a estabilidade e a convergência do método de Runge-Kutta diante de variações abruptas de corrente, mantendo, simultaneamente, o tempo de processamento computacional dentro de limites viáveis.

Um aspecto fundamental na dinâmica de transformadores reside na dependência da resposta transitória em relação ao instante de conexão à rede elétrica, evento que, na prática operacional, ocorre de forma aleatória. Visando avaliar a robustez do modelo matemático frente a diferentes condições iniciais, o estudo contemplou dois cenários distintos de chaveamento. O primeiro considera a conexão em ângulo de fase ótimo (90°), modelado pela função cosseno ($v(t) = V_p \cdot \cos(\omega t)$). Esta configuração alinha fisicamente o fluxo magnético inicial nulo com a derivada da tensão, eliminando a componente contínua de compensação. Sob esta condição, o sistema entra imediatamente em regime permanente, permitindo verificar se a corrente de excitação apresenta o comportamento senoidal e a defasagem de 90 graus previstos pela teoria de circuitos para indutores lineares. É possível visualizar a forma de onda da tensão com 90° de fase na Figura 3.

Figura 3: Tensão de entrada com 90° de fase

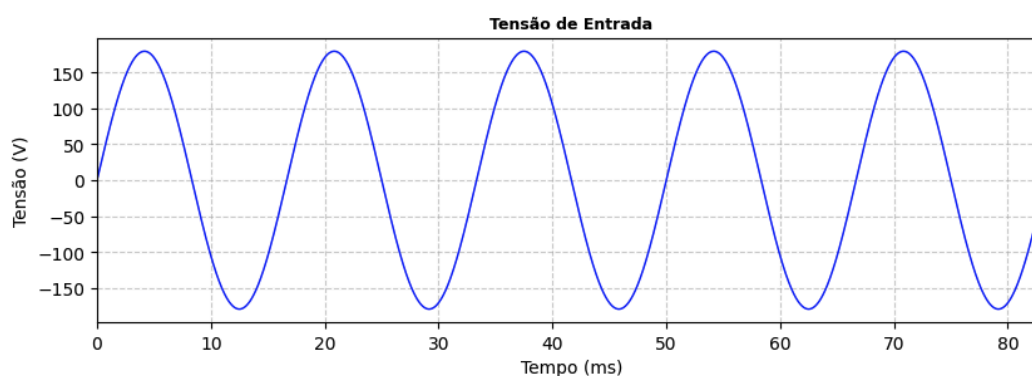


Fonte: Autoria própria (2026).

Em contrapartida, analisou-se também o "pior caso" de energização, representado pela função seno ($v(t) = V_p \cdot \text{sen}(\omega t)$), que corresponde ao fechamento do circuito no instante em que a tensão cruza o zero (0°). Conforme a Lei de Faraday, esta condição impõe

que o fluxo magnético atinja, teoricamente, o dobro de sua amplitude máxima no primeiro semiciclo, levando o núcleo a uma saturação profunda e unidirecional. A simulação deste cenário é indispensável para observar a geração da corrente de *inrush* e validar a capacidade do algoritmo em representar transientes severos e o decaimento da componente CC. A Figura 4 demonstra a forma de onda da tensão para o pior caso.

Figura 4: Tensão de entrada com 0° de fase



Fonte: Autoria própria (2026).

3.1 ANÁLISE EM TENSÃO ABAIXO DA NOMINAL

Em primeiras análises, é imprescindível verificar a consistência física e matemática do simulador em condições ideais. Para tanto, o transformador foi simulado com uma tensão de entrada reduzida (20 V_p), garantindo que a indução magnética máxima permanecesse muito abaixo do joelho da curva de saturação.

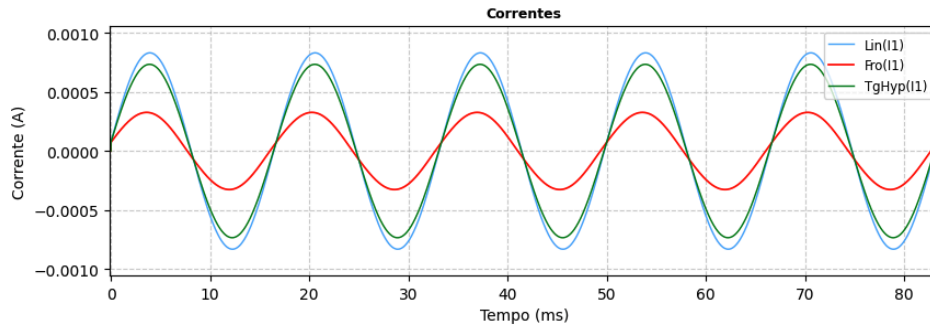
3.1.1 Conexão 20 V_p (90°)

As Figuras 5, 6, e 7 apresentam, respectivamente, as correntes de magnetização, as curvas $B \times H$ e o espectro de frequências para a condição de conexão em fase ótima (90°).

A Figura 5 confirma a validação do algoritmo em regime linear. Observa-se que as correntes de magnetização são perfeitamente senoidais, simétricas em relação ao eixo zero e com amplitudes reduzidas (na ordem de miliamperes). Além disso, verifica-se o atraso de fase de 90° em relação à tensão da Figura 3. Este comportamento é característico da operação a

vazio, onde a impedância de magnetização é muito superior à resistência do enrolamento primário, conferindo à carga um caráter predominantemente indutivo.

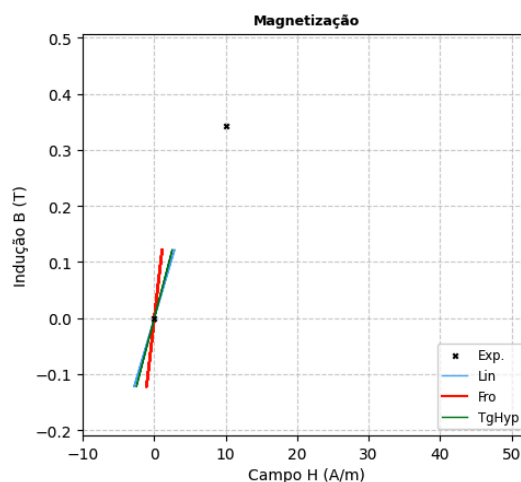
Figura 5: Correntes no primário para 20 Vp (90°)



Fonte: Autoria própria (2026).

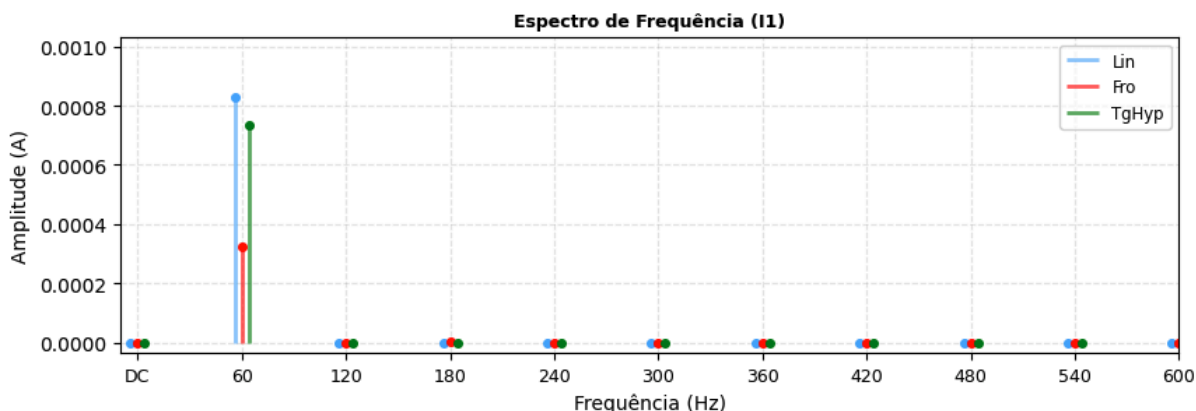
A análise do ciclo de magnetização na Figura 6 demonstra que, sob baixa excitação, a excursão do ponto de operação restringe-se a uma região estreita e linear próxima da origem, apresentando apenas um traçado reto. Neste contexto, nota-se que a amplitude da corrente estimada pelo modelo de Fröhlich é inferior às demais. Isto ocorre devido ao ajuste matemático da função, que apresenta uma inclinação inicial diferente (maior permeabilidade inicial). Conseqüentemente, o modelo exige uma menor intensidade de campo magnético H para alcançar a mesma indução B , o que se traduz diretamente na geração de uma corrente elétrica menor. A análise espectral via FFT, demonstrada na Figura 7, corrobora a linearidade, exibindo apenas a componente fundamental de 60 Hz, isenta de distorções harmônicas.

Figura 6: Curvas B x H para 20 Vp (90°)



Fonte: Autoria própria (2026).

Figura 7: FFTs para 20 Vp (90°)



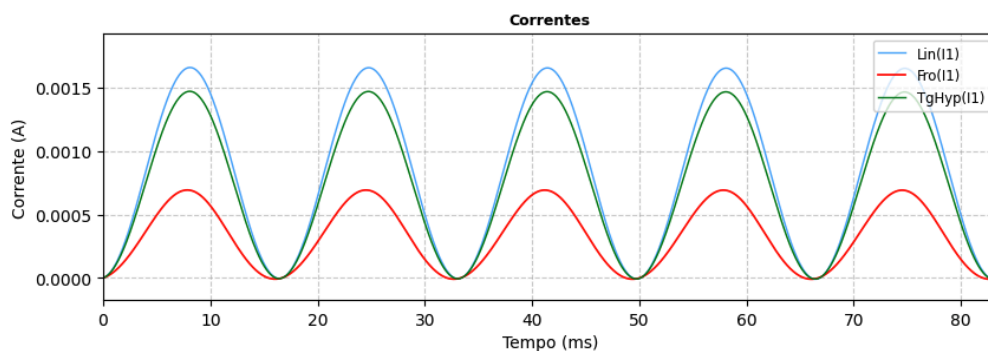
Fonte: Autoria própria (2026).

3.1.2 Conexão 20 Vp (0°)

As Figuras 8, 9 e 10 ilustram o comportamento do sistema no "pior caso" de energização (0°), simulado pela função seno.

Na Figura 8, pode-se observar o comportamento das correntes no primário para este caso. Diferentemente do caso de melhor conexão à rede, a corrente nasce deslocada do eixo zero, apresentando uma componente contínua (CC) significativa. Um aspecto crucial observado é o lento decaimento desta componente. Como o núcleo opera na região linear (não saturada), a indutância de magnetização (L_m) assume seu valor máximo e constante. Sendo a constante de tempo do circuito dada por $\tau = L_m/RI$, a elevada indutância resulta em um τ longo, justificando o amortecimento lento do transiente observado.

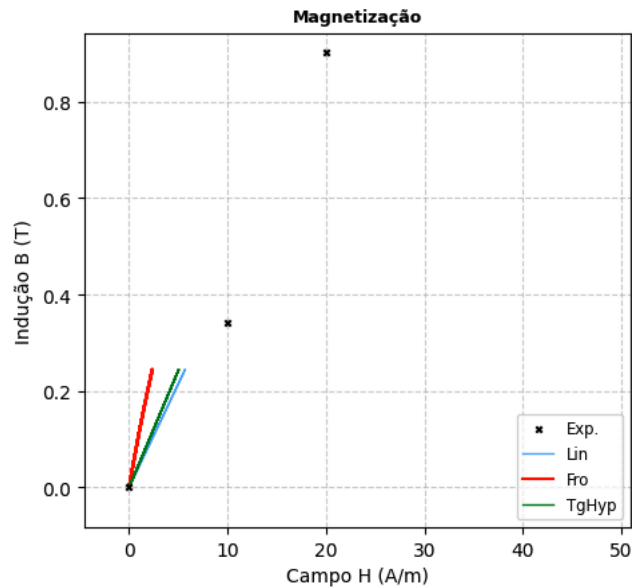
Figura 8: Correntes no primário para 20 Vp (0°)



Fonte: Autoria própria (2026).

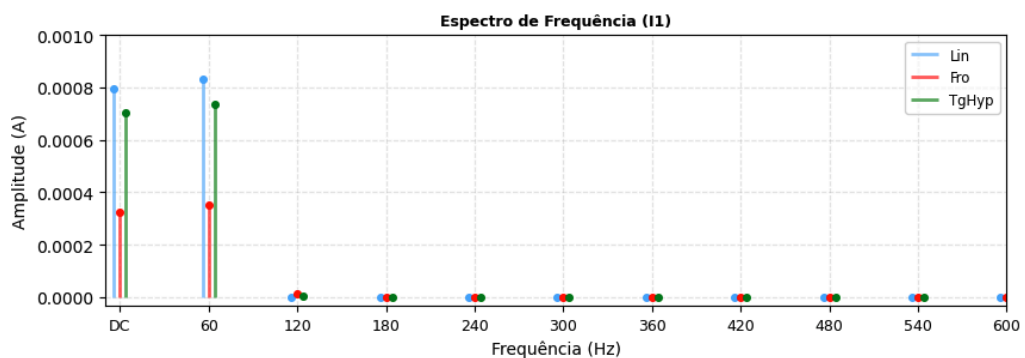
As curvas $B \times H$ na Figura 9 mostram o deslocamento do ciclo de operação para o primeiro quadrante, evidenciando a magnetização unidirecional típica do *inrush*. O espectro da Figura 10 reflete essa assimetria através do surgimento de componentes em nível CC.

Figura 9: Curvas $B \times H$ para 20 Vp (0°)



Fonte: Autoria própria (2026).

Figura 10: FFTs para 20 Vp (0°)



Fonte: Autoria própria (2026).

3.2 ANÁLISE EM TENSÃO NOMINAL

Na segunda etapa da avaliação, o transformador foi submetido à sua tensão nominal de projeto ($127 \text{ Vrms} \approx 179,6 \text{ Vp}$) e analisado em regime permanente, por meio do uso de

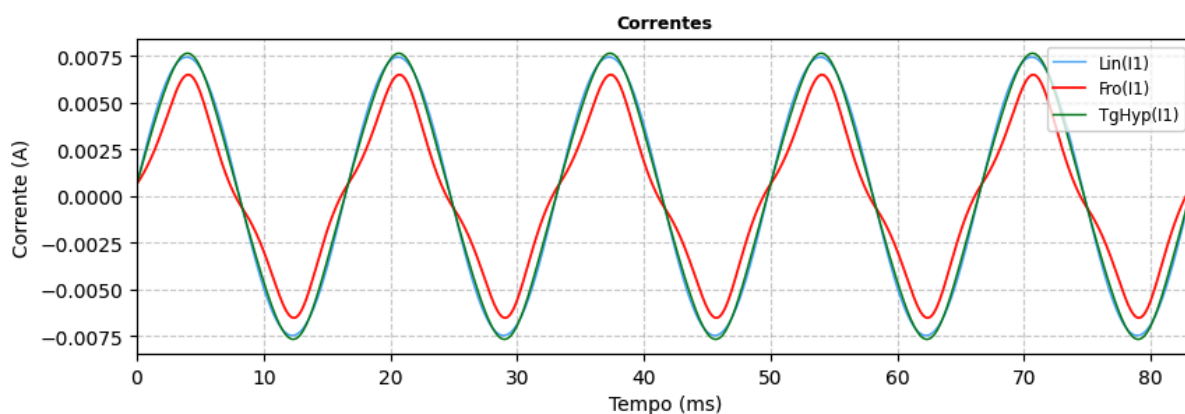
entrada em ângulo de fase ótimo e em regime transitório, utilizando entrada de energia no pior caso de ângulo de fase.

3.2.1 Conexão 179,6 Vp (90°)

As Figuras 11, 12 e 13 apresentam os resultados para a conexão em ângulo de fase ótimo (90°).

A análise visual da corrente de magnetização na Figura 11 revela que a forma de onda do modelo de Fröhlich deixa de ser puramente senoidal e começa a assumir um aspecto distorcido, enquanto os outros modelos ainda permanecem praticamente senoidais.

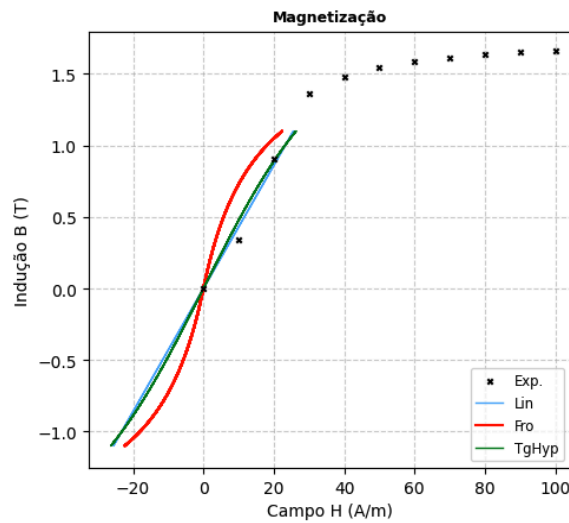
Figura 11: Correntes no primário para 179,6 Vp (90°)



Fonte: Autoria própria (2026).

A Figura 12 exibe os ciclos magnéticos dos modelos. Nota-se que o modelo de Fröhlich já apresenta o formato característico em "S", indicando uma excursão maior na região de saturação, ao passo que o modelo da Tangente Hiperbólica exibe uma curvatura mais discreta na extremidade superior.

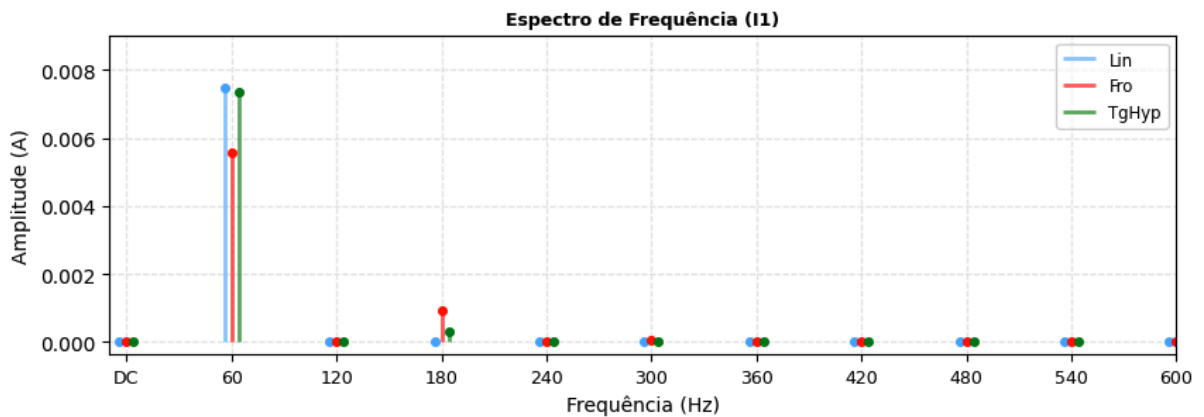
Figura 12: Curvas B x H para 179,6 Vp (90°)



Fonte: Autoria própria (2026).

A distribuição espectral é vista na Figura 13: enquanto o modelo linear apresenta apenas a fundamental (60 Hz), os modelos não lineares revelam o surgimento de harmônicos ímpares, com destaque para a 3ª harmônica (180 Hz) e a 5ª harmônica (300 Hz), componentes que, apesar de pequenas, impactam diretamente a qualidade da energia elétrica.

Figura 13: FFTs para 179,6 Vp (90°)



Fonte: Autoria própria (2026).

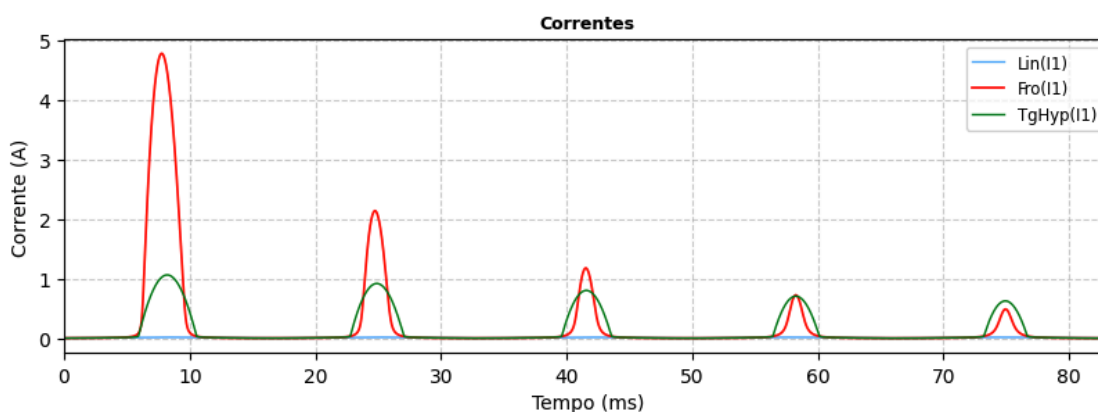
3.2.2 Conexão 179,6 Vp (0°)

As Figuras 14, 15, 16, 17 e 18 ilustram o comportamento crítico do sistema sob conexão no cruzamento por zero (0°), onde a excursão do fluxo magnético dobra, levando o

núcleo à saturação profunda. Devido à severidade deste fenômeno, a análise contempla tanto a resposta transitória imediata, focada na magnitude dos picos, quanto o comportamento de decaimento a longo prazo.

A Figura 14 detalha os primeiros cinco ciclos de operação e expõe uma divergência crítica de robustez numérica entre os modelos. O modelo de Fröhlich estima um pico de corrente de *inrush* extremamente elevado, próximo a 4,8 A. Matematicamente, isso ocorre devido à assíntota horizontal da função racional de Fröhlich; à medida que o fluxo se aproxima do parâmetro de saturação ajustado, a permeabilidade diferencial tende a valores mínimos abruptamente, fazendo a corrente calculada disparar. Em contrapartida, o modelo da Tangente Hiperbólica estima um pico consideravelmente mais conservador e fisicamente plausível, em torno de 1,1 A, refletindo o comportamento assintótico de sua derivada. Como a taxa de variação não colapsa abruptamente a zero na transição, a formulação exige valores menores de campo magnético para sustentar o fluxo, permitindo a estimativa de correntes de pico reduzidas e matematicamente estáveis. Já o modelo Linear falha em representar a severidade do fenômeno, mantendo a amplitude nominal apenas deslocada por um nível CC, o que levaria ao subdimensionamento de proteções.

Figura 14: Correntes no primário para 179,6 Vp (0° - primeiros ciclos)

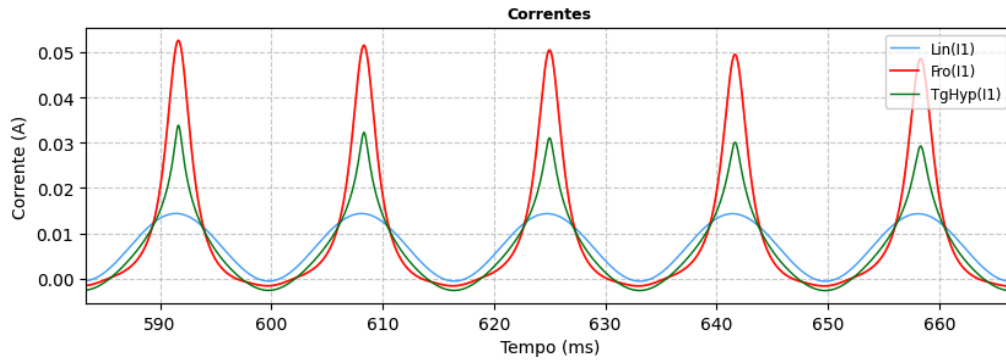


Fonte: Autoria própria (2026).

Avançando a janela de observação, a Figura 15 apresenta os últimos 5 ciclos, dos 40 simulados neste caso, permitindo analisar a dissipação de energia. Um fenômeno físico relevante observado é a variação dinâmica da velocidade de decaimento. O modelo Linear (azul) apresenta o decaimento mais lento de todos. Isso ocorre porque sua indutância é fixa e elevada, retendo a energia magnética no circuito por mais tempo ($\tau = L/R$ constante e alto). Já

os modelos não lineares apresentam um decaimento mais acelerado, pois, durante os picos de saturação, a indutância diferencial do núcleo reduz-se drasticamente, diminuindo a constante de tempo instantânea e permitindo que a energia excedente seja "drenada" mais rapidamente.

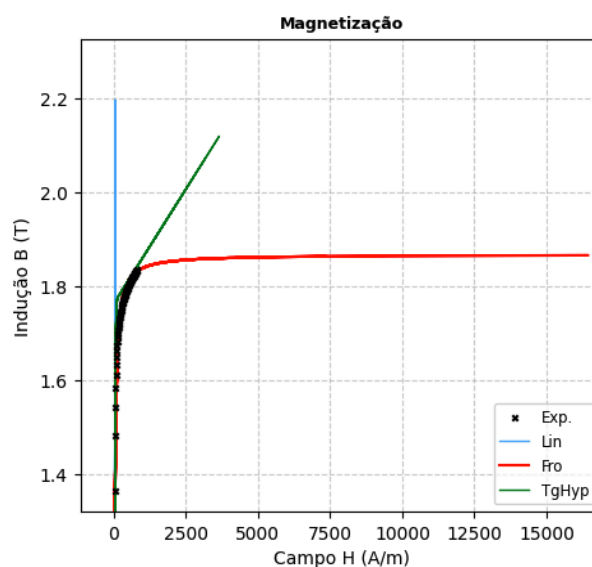
Figura 15: Correntes no primário para 179,6 Vp (0° - últimos ciclos)



Fonte: Autoria própria (2026).

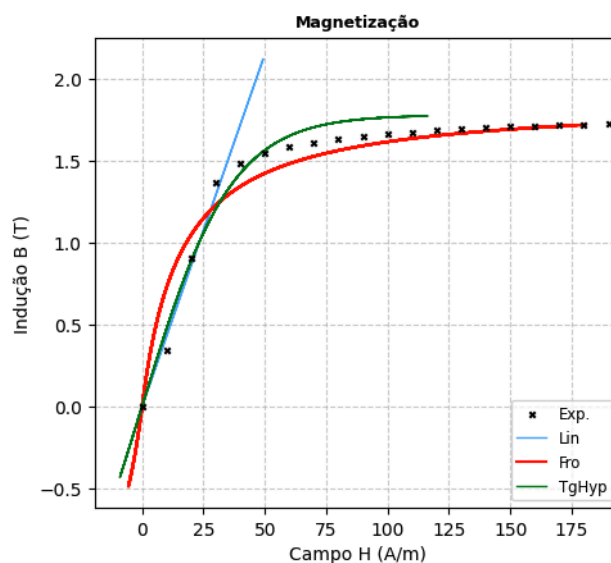
Já as Figura 16 e 17 ilustram as trajetórias dinâmicas do ponto de operação no plano $B \times H$. A visualização dos primeiros ciclos (Figura 16) evidencia a saturação extrema do núcleo magnético, confirmando as altas correntes de *inrush*. A visão dos últimos cinco ciclos (Figura 17) evidenciam uma considerável redução na saturação do núcleo em comparação aos primeiros picos de corrente, com uma tendência à operação mais próxima do joelho da curva, porém ainda saturados.

Figura 16: Curvas B x H para 179,6 Vp (0° - primeiros ciclos)



Fonte: Autoria própria (2026).

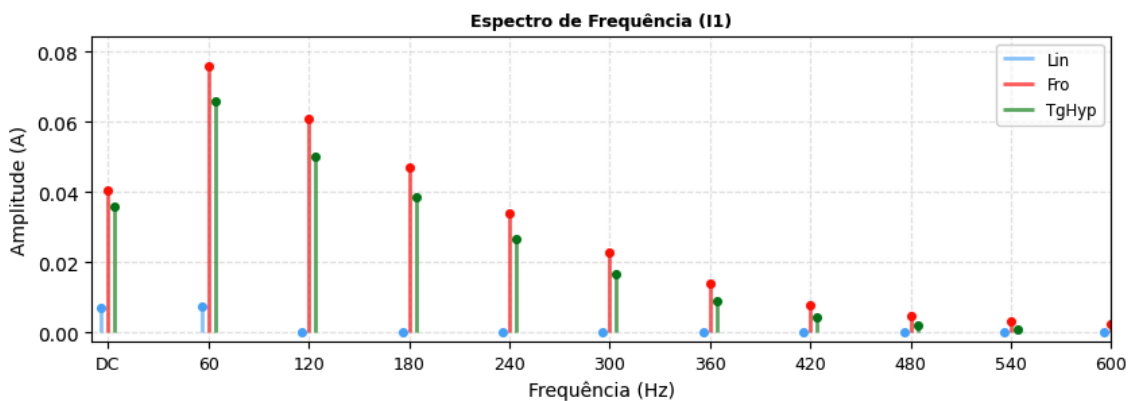
Figura 17: Curvas B x H para 179,6 Vp (0° - últimos ciclos)



Fonte: Autoria própria (2026).

Por fim, a Figura 18 demonstra a análise espectral englobando todos os 40 ciclos simulados. É possível perceber uma expressiva distorção harmônica gerada pelos modelos não lineares. Devido à assimetria severa da forma de onda durante o transiente unidirecional, o espectro é composto não apenas por harmônicos ímpares, mas também por fortes componentes pares (como a 2ª harmônica em 120 Hz). O modelo Linear, incapaz de representar tal distorção, exibe apenas a componente contínua (CC) injetada no início do transiente juntamente com a fundamental de 60 Hz.

Figura 18: FFTs para 179,6 Vp (0°)



Fonte: Autoria própria (2026).

3.3 ANÁLISE EM SOBRETENSÃO

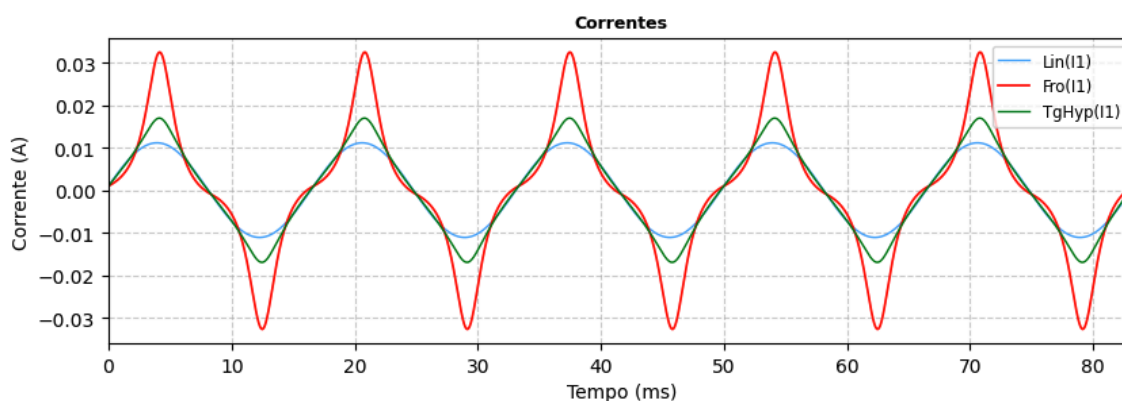
Para investigar os limites de estabilidade dos modelos e o comportamento do transformador sob condições de estresse magnético, realizou-se uma terceira bateria de simulações aplicando uma sobretensão de aproximadamente 50% acima do valor nominal (190 Vrms \approx 268 Vpico). Neste cenário, a indução magnética é forçada a realizar excursões profundas na região de saturação.

3.3.1 Conexão 268 Vp (90°)

As Figuras 19, 20 e 21 apresentam os resultados para a conexão em ângulo de fase ótimo (90°) sob sobretensão, correspondente a um aumento de aproximadamente 50% em relação ao valor nominal. A análise da corrente de magnetização na Figura 19 evidencia o distanciamento substancial entre as estimativas dos modelos matemáticos quando o núcleo é forçado a operar na região de saturação.

Diferentemente do cenário nominal, a forma de onda do modelo de Fröhlich (vermelho) exibe picos extremamente agudos e de alta amplitude, atingindo valores próximos a 0,03 A, o que representa o dobro da corrente estimada pelo modelo da Tangente Hiperbólica, de aproximadamente 0,015 A. Fisicamente, essa distorção acentuada ocorre porque o núcleo magnético permanece além do joelho da curva de magnetização durante uma grande parcela do semiciclo. Para forçar a variação senoidal do fluxo magnético exigida rigorosamente pela fonte de tensão, o circuito elétrico demanda pulsos de corrente de elevada amplitude nos instantes de pico.

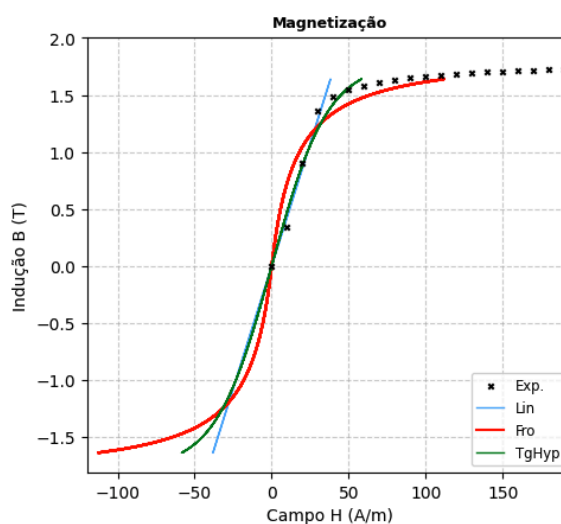
Figura 19: Correntes no primário para 268 Vp (90°)



Fonte: Autoria própria (2026).

A justificativa matemática e magnética para essa divergência de amplitudes é confirmada pela Figura 20. Observando o comportamento das curvas de magnetização no primeiro e terceiro quadrantes, nota-se que o modelo de Fröhlich demanda uma intensidade de campo magnético (H) consideravelmente maior para sustentar densidades de fluxo acima de 1,4 T. Por outro lado, o modelo da Tangente Hiperbólica apresenta uma transição geométrica distinta, exigindo menor força magnetomotriz para a mesma excursão de indução magnética, resultando em correntes de pico mais contidas.

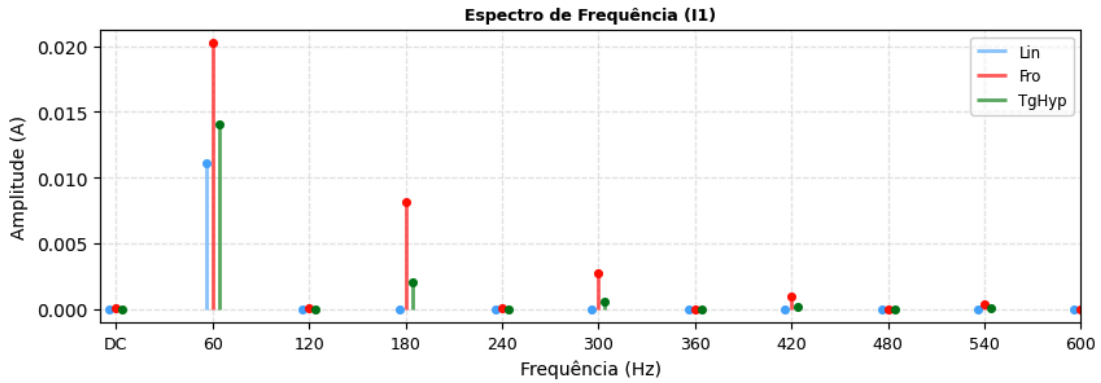
Figura 20: Curvas B x H para 268 Vp (90°)



Fonte: Autoria própria (2026).

A consequência desta operação sob forte não linearidade reflete-se nitidamente no espectro de frequências da Figura 21. A distorção harmônica atinge níveis críticos nos modelos não lineares, especialmente no modelo de Fröhlich, onde a magnitude da 3ª harmônica (180 Hz) aproxima-se da metade do valor da componente fundamental (60 Hz), além do surgimento proeminente de componentes de 5ª e 7ª ordem. Em uma aplicação prática, esse grau de poluição harmônica resultaria em perdas suplementares substanciais e aquecimento excessivo do componente. O modelo Linear, por sua vez, mantém seu espectro inalterado e restrito à fundamental, o que reafirma a sua total inadequação para a previsão do comportamento e da qualidade de energia de transformadores reais operando sob estresse elétrico.

Figura 21: FFTs para 268 Vp (90°)

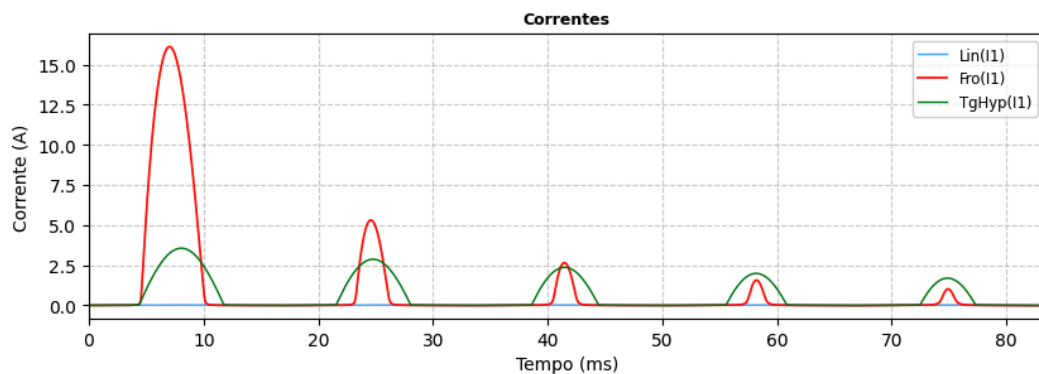


Fonte: Autoria própria (2026).

3.3.2 Conexão 268 Vp (0°)

A simulação do *inrush* sob sobretensão evidencia o agravamento das divergências nas estimativas das formulações matemáticas. A Figura 22 mostra que, em comparação ao cenário nominal, o modelo de Fröhlich apresenta picos de corrente consideravelmente maiores nesta condição, aproximando-se de 16,0 A. A Tangente Hiperbólica, por sua vez, resulta em uma estimativa de magnitude inferior para a corrente transitória, limitando o valor de pico a cerca de 3,5 A.

Figura 22: Corrente no primário para 268 Vp (0° - primeiros ciclos)

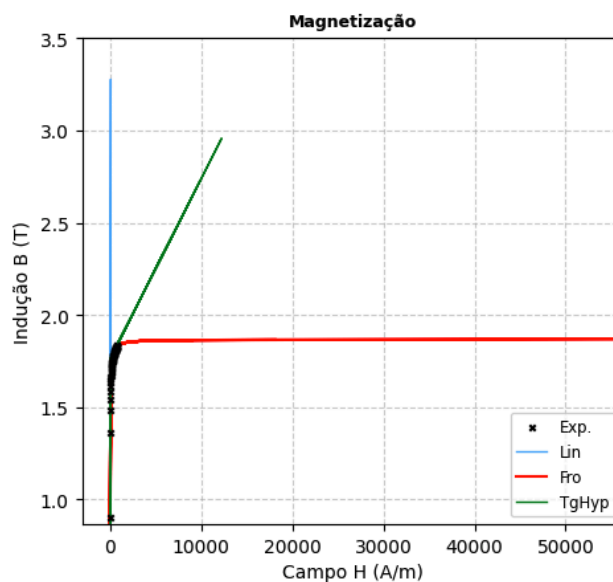


Fonte: Autoria própria (2026).

A divergência observada nas correntes reflete-se no campo magnético demandado pelo modelo de Fröhlich, que supera os 50.000 A/m, conforme ilustrado na Figura 23. Este comportamento resulta em uma curva de magnetização que se aproxima de uma reta horizontal nesta escala. A Tangente Hiperbólica apresenta uma transição geométrica distinta,

exigindo uma intensidade de campo magnético substancialmente menor para a mesma excursão de fluxo.

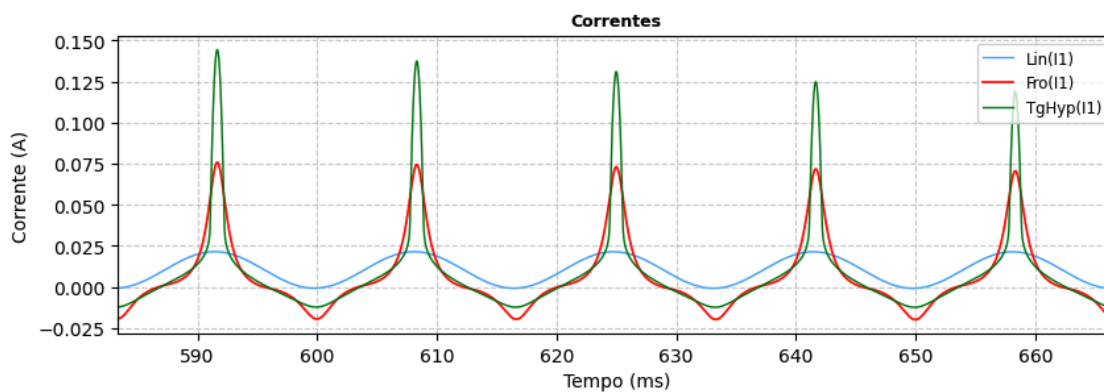
Figura 23: Curvas B x H para 268 Vp (0° - primeiros ciclos)



Fonte: Autoria própria (2026).

A Figura 24 ilustra as correntes no primário após cerca de 35 ciclos de simulação, momento em que o transiente se encontra atenuado. Nota-se que as formas de onda preservam uma distorção harmônica expressiva em comparação ao regime nominal, o que evidencia a operação contínua na região não linear do material, mesmo com os picos de corrente reduzidos à grandeza dos miliamperes.

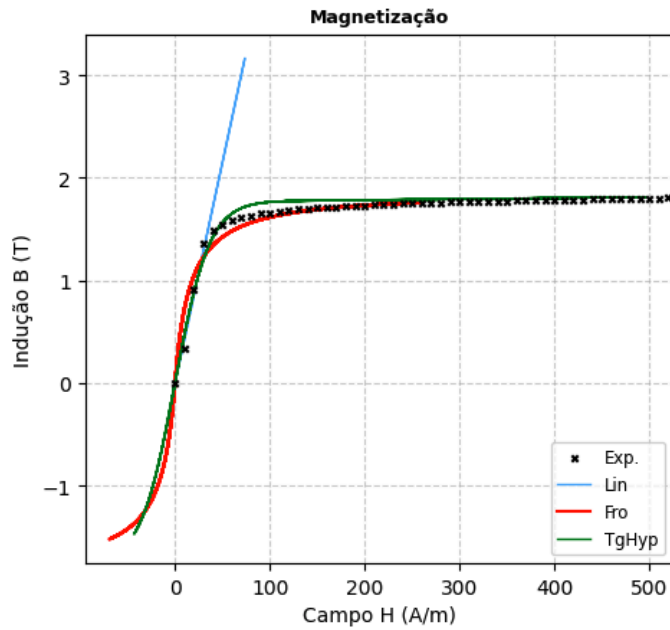
Figura 24: Correntes no primário para 268 Vp (0° - últimos ciclos)



Fonte: Autoria própria (2026).

As trajetórias dinâmicas nos ciclos finais simulados, mostradas na Figura 25, confirmam o retorno a um ciclo magnético estável, mas o sistema ainda opera fortemente na região saturada devido ao nível excessivo de tensão imposto pela rede.

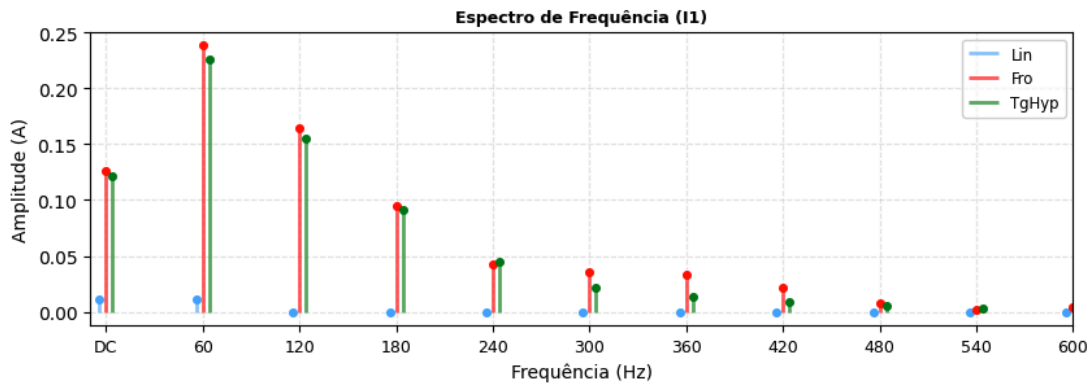
Figura 25: Curvas B x H para 268 Vp (0° - últimos ciclos)



Fonte: Autoria própria (2026).

Por fim, a Figura 26 reafirma o impacto da assimetria da onda através do surgimento de harmônicos pares, com a 2ª harmônica (120 Hz) apresentando forte magnitude no espectro transiente dos modelos não lineares, um comportamento de vital importância para projetos de proteção e que não é contemplado na formulação do modelo Linear.

Figura 26: FFTs para 268 Vp (0°)



Fonte: Autoria própria (2026).

4 CONCLUSÃO

Este trabalho atingiu seu objetivo principal ao desenvolver e validar uma modelagem computacional para transformadores elétricos, focada na análise crítica dos efeitos da saturação magnética e na comparação de modelos matemáticos representativos da curva anisterética do aço silício. A metodologia adotada permitiu transitar desde o dimensionamento físico do equipamento de 60 VA até a simulação dinâmica no domínio do tempo, revelando fenômenos intrínsecos à não linearidade dos materiais ferromagnéticos.

Do ponto de vista computacional, a formulação de uma solução algébrica direta para desacoplar as equações diferenciais, aliada ao método de integração numérica de Runge-Kutta de 4ª ordem, provou-se altamente eficiente. O algoritmo demonstrou estabilidade tanto nas excursões lineares sob baixa tensão quanto nas extremas variações de permeabilidade inerentes à saturação profunda, garantindo a fidelidade das formas de onda sem instabilidades.

As simulações evidenciaram que o Modelo Linear para o material é inadequado para representar a realidade física do equipamento operando em tensões nominais ou superiores. Ao ignorar a saturação, este modelo subestima a amplitude dos picos de corrente e falha em prever a distorção harmônica, tornando-se obsoleto para estudos de qualidade de energia e proteção.

Entre os modelos não lineares analisados em regime permanente, tanto a equação de Fröhlich quanto a da Tangente Hiperbólica conseguiram capturar o afinamento dos picos de corrente de excitação e a injeção de harmônicos ímpares na rede, dominados pela 3ª harmônica. Contudo, sob sobretensão severa, o modelo de Fröhlich, para a parametrização utilizada, apresentou picos acentuadamente mais agudos, demandando forças magnetomotrizes consideravelmente maiores.

O destaque da pesquisa foi a análise do comportamento transitório no pior caso de energização (cruzamento da tensão por zero). Nesse cenário, ficou evidente uma diferença significativa no comportamento das formulações matemáticas. O modelo de Fröhlich apresentou uma limitação importante: devido à assíntota horizontal da sua curva, a permeabilidade diferencial atinge valores extremamente baixos na região de saturação profunda. Como consequência, o modelo estimou picos de corrente muito elevados, chegando próximo a 16,0 A.

Em contrapartida, o modelo da Tangente Hiperbólica apresentou maior estabilidade numérica na estimativa dos picos de corrente. Graças à sua formulação matemática, este

modelo garante uma transição mais controlada para a região de saturação profunda, evitando quedas abruptas na permeabilidade magnética. Essa característica permitiu estimar correntes de *inrush* mais condizentes com os limites físicos de impedância do circuito (aproximadamente 4,8 A). Além disso, o modelo representou adequadamente a rápida atenuação do transiente e o surgimento das componentes harmônicas pares (2^a e 4^a), que são típicas da assimetria desse fenômeno.

Para trabalhos futuros, sugere-se expandir o modelo computacional para incluir o laço de histerese dinâmico, avançando além da limitação da curva anisterética. A incorporação das perdas no núcleo e a posterior validação dos resultados através de ensaios práticos de laboratório seriam passos importantes para aprimorar e consolidar a ferramenta de simulação desenvolvida neste estudo.

REFERÊNCIAS

LANDGRAF, F. J. G. Propriedades magnéticas de aços para fins elétricos. São Paulo: Instituto de Pesquisas Tecnológicas do Estado de São Paulo, [s.d.].

MARTIGNONI, A. Transformadores. 8. ed. São Paulo: Ed. Globo, 1991.

FITZGERALD, A. E.; KINGSLEY JR., C.; UMANS, S. D. Máquinas Elétricas. 6. ed. Porto Alegre: AMGH, 2006.

CHAPRA, S. C.; CANALE, R. P. Métodos Numéricos para Engenharia. 5. ed. São Paulo: McGraw-Hill, 2008.

APÊNDICE A – CÓDIGO FONTE EM PYTHON PARA A SIMULAÇÃO NUMÉRICA

```

import tkinter as tk
from tkinter import ttk, messagebox
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg,
NavigationToolbar2Tk
import threading
import sys
from scipy.optimize import curve_fit

# --- DADOS DE REFERENCIA ---
RAW_DATA = [
    [0, 0], [0.342, 10], [0.9035, 20], [1.3625, 30], [1.4807, 40], [1.5418,
50],
    [1.5813, 60], [1.6104, 70], [1.6316, 80], [1.6477, 90], [1.661, 100],
[1.6727, 110],
    [1.682, 120], [1.6912, 130], [1.6983, 140], [1.7057, 150], [1.7118, 160],
[1.716, 170],
    [1.7207, 180], [1.7262, 190], [1.7297, 200], [1.7339, 210], [1.7376,
220], [1.7416, 230],
    [1.7455, 240], [1.7479, 250], [1.7507, 260], [1.7549, 270], [1.7558,
280], [1.7596, 290],
    [1.7626, 300], [1.7639, 310], [1.7664, 320], [1.7684, 330], [1.7707,
340], [1.7732, 350],
    [1.7758, 360], [1.7771, 370], [1.779, 380], [1.7809, 390], [1.7834, 400],
[1.7843, 410],
    [1.7874, 420], [1.7873, 430], [1.7902, 440], [1.7912, 450], [1.7939,
460], [1.7943, 470],
    [1.7958, 480], [1.7971, 490], [1.7997, 500], [1.8005, 510], [1.8021,
520], [1.8038, 530],
    [1.8046, 540], [1.8058, 550], [1.8076, 560], [1.8092, 570], [1.8103,
580], [1.8106, 590],
    [1.8125, 600], [1.8138, 610], [1.8144, 620], [1.8154, 630], [1.8171,
640], [1.8189, 650],
    [1.819, 660], [1.8202, 670], [1.822, 680], [1.8224, 690], [1.8239, 700],
[1.8248, 710],
    [1.8256, 720], [1.8275, 730], [1.8277, 740], [1.8292, 750], [1.8302,
760], [1.8318, 770],
    [1.8317, 780], [1.8333, 790], [1.8339, 800]
]
DATA_B = np.array([p[0] for p in RAW_DATA])
DATA_H = np.array([p[1] for p in RAW_DATA])

# --- TOOLBAR PERSONALIZADA ---
class Toolbar(NavigationToolbar2Tk):
    toolitems = [t for t in NavigationToolbar2Tk.toolitems if t[0] != 'Save']

# --- GERENCIADOR DE CURSORES ---
class CursorManager:
    def __init__(self, canvas, ax):
        self.canvas = canvas
        self.ax = ax
        self.cursor_line = self.ax.axvline(x=0, color='gray', linestyle='--',
linewidth=1, alpha=0)
        self.cursor_text = self.ax.text(0.02, 0.95, '',
transform=self.ax.transAxes,
bbox=dict(boxstyle="round",
fc="white", alpha=0.9, ec='gray'),

```

```

                                                    fontweight='bold', fontsize=9,
visible=False, color='black')
    self.canvas.mpl_connect('motion_notify_event', self.on_mouse_move)

    def on_mouse_move(self, event):
        if not event.inaxes or event.inaxes != self.ax:
            self.cursor_line.set_alpha(0)
            self.cursor_text.set_visible(False)
            self.canvas.draw_idle()
            return

        x_mouse = event.xdata
        lines = [l for l in self.ax.get_lines() if l != self.cursor_line and
l.get_visible()]
        if not lines: return

        text_info = []
        text_info.append(f"X = {x_mouse:.4f}")

        found_data = False
        for line in lines:
            x_data = line.get_xdata()
            y_data = line.get_ydata()
            if len(x_data) < 2: continue
            if x_mouse < x_data[0] or x_mouse > x_data[-1]: continue

            y_val = np.interp(x_mouse, x_data, y_data)
            label = line.get_label()
            if label and not label.startswith('_'):
                lbl = label.replace("Linear", "Lin").replace("Fröhlich",
"Fro").replace("TgHyp", "TgH")
                text_info.append(f"{lbl}: {y_val:.2f}")
                found_data = True

        if found_data:
            self.cursor_line.set_xdata([x_mouse])
            self.cursor_line.set_alpha(0.8)
            self.cursor_text.set_text("\n".join(text_info))
            self.cursor_text.set_visible(True)
            self.canvas.draw_idle()

# --- APP PRINCIPAL ---
class SimuladorApp:
    def __init__(self, root):
        self.root = root
        self.root.title("Simulador - Transformador Monofásico - TCC")
        self.root.geometry("1360x950")
        self.root.resizable(False, False)
        self.root.protocol("WM_DELETE_WINDOW", self.on_closing)

        self.is_running = False
        self.last_results = None

        # --- LAYOUT ESQUERDO (Painel de Controle) ---
        frame_container = ttk.Frame(root)
        frame_container.place(x=10, y=10, width=350, height=930)

        canvas_scroll = tk.Canvas(frame_container)

```

```

        scrollbar = ttk.Scrollbar(frame_container, orient="vertical",
command=canvas_scroll.yview)
        self.scrollable_frame = ttk.Frame(canvas_scroll)

        self.scrollable_frame.bind(
            "<Configure>",
            lambda e:
canvas_scroll.configure(scrollregion=canvas_scroll.bbox("all"))
        )

        canvas_scroll.create_window((0, 0), window=self.scrollable_frame,
anchor="nw")
        canvas_scroll.configure(yscrollcommand=scrollbar.set)

        canvas_scroll.pack(side="top", fill="both", expand=True)
        scrollbar.place(relx=1.0, rely=0, relheight=1.0, anchor="ne")

# --- LAYOUT DIREITO (Gráficos) ---
        frame_right = ttk.Frame(root)
        frame_right.place(x=370, y=10, width=980, height=930)

        self.notebook = ttk.Notebook(frame_right)
        self.notebook.pack(fill=tk.BOTH, expand=True)

        self.tab_voltage = ttk.Frame(self.notebook)
        self.tab_current = ttk.Frame(self.notebook)
        self.tab_bxh = ttk.Frame(self.notebook)
        self.tab_fft = ttk.Frame(self.notebook)

        self.notebook.add(self.tab_voltage, text=" Tensão ")
        self.notebook.add(self.tab_current, text=" Correntes ")
        self.notebook.add(self.tab_bxh, text=" Magnetização ")
        self.notebook.add(self.tab_fft, text=" FFT ")

        self.setup_chart_voltage()
        self.setup_chart_current()
        self.setup_chart_bxh()
        self.setup_chart_fft()

# --- CONTROLES ---
        self.entries = {}

        def add_input(parent, label, val, key, r, c):
            ttk.Label(parent, text=label).grid(row=r, column=c*2, sticky="w",
padx=2, pady=2)
            entry = ttk.Entry(parent, width=8)
            entry.insert(0, str(val))
            entry.grid(row=r, column=c*2+1, sticky="w", padx=2, pady=2)
            self.entries[key] = entry

# 1. CIRCUITO
        lf_circ = ttk.LabelFrame(self.scrollable_frame, text="1. Circuito
Elétrico", padding=5)
        lf_circ.pack(fill="x", pady=5, padx=5)
        add_input(lf_circ, "R1 (Ω):", 7.86, 'R1', 0, 0) ;
add_input(lf_circ, "R2 (Ω):", 0.29, 'R2', 0, 1)
        add_input(lf_circ, "L1 (H):", 0.0194, 'L1', 1, 0) ;
add_input(lf_circ, "L2 (H):", 0.00077, 'L2', 1, 1)
        add_input(lf_circ, "N1:", 578, 'N1', 2, 0) ; add_input(lf_circ,
"N2:", 110, 'N2', 2, 1)

```

```

        ttk.Label(lf_circ, text="R Carga ( $\Omega$ ):").grid(row=3, column=0,
sticky="e", padx=2, pady=2)
        e_rc = ttk.Entry(lf_circ, width=8)
        e_rc.insert(0, "9.6")
        e_rc.grid(row=3, column=1, sticky="w", padx=2, pady=2)
        self.entries['Rcarga'] = e_rc

        # 2. GEOMETRIA
        lf_geo = ttk.LabelFrame(self.scrollable_frame, text="2. Geometria do
Núcleo", padding=5)
        lf_geo.pack(fill="x", pady=5, padx=5)
        add_input(lf_geo, "Área S ( $m^2$ ):", 0.00075, 'S', 0, 0)
        add_input(lf_geo, "Comp lm (m):", 0.168, 'lm', 0, 1)

        # 3. PARÂMETROS DOS MODELOS
        lbl_params = ttk.Label(self.scrollable_frame, text="3. Parâmetros dos
Modelos", font=("Arial", 9, "bold"))
        lbl_params.pack(anchor="w", padx=5, pady=(10, 0))

        param_notebook = ttk.Notebook(self.scrollable_frame)
        param_notebook.pack(fill="x", padx=5, pady=5)

        # Aba Linear
        tab_lin = ttk.Frame(param_notebook)
        param_notebook.add(tab_lin, text="Linear")
        ttk.Label(tab_lin, text="Linearidade ideal (Sem saturação)",
font=("Arial", 8, "italic"), foreground="#007acc").grid(row=0, column=0,
columnspan=4, sticky="w", pady=5)
        add_input(tab_lin, " $\mu_r$  ( $\mu_r$ ):", 5000, 'mur_lin', 1, 0)

        # Aba Fröhlich
        tab_fro = ttk.Frame(param_notebook)
        param_notebook.add(tab_fro, text="Fröhlich")
        ttk.Label(tab_fro, text="Ctes ' $\gamma$ ' e ' $\delta$ '", font=("Arial", 8,
"italic"), foreground="#cc0000").grid(row=0, column=0, columnspan=4,
sticky="w", pady=5)
        add_input(tab_fro, " $\mu_r$  ( $1/(\gamma*\mu_0)$ ):", 5000, 'mur_fro', 1, 0)
        add_input(tab_fro, "B Sat ( $1/\delta$ ):", 1.1, 'Bsat_fro', 1, 1)

        # Aba Hiperbólica
        tab_hip = ttk.Frame(param_notebook)
        param_notebook.add(tab_hip, text="TgHyp")
        ttk.Label(tab_hip, text="Ctes ' $\alpha$ ' e ' $\beta$ '", font=("Arial", 8,
"italic"), foreground="#009900").grid(row=0, column=0, columnspan=4,
sticky="w", pady=5)
        add_input(tab_hip, " $\mu_r$  ( $(\alpha*Bsat)/\mu_0$ )", 5000, 'mur_hip', 1, 0)
        add_input(tab_hip, "B Sat ( $\beta$ ):", 1.1, 'Bsat_hip', 1, 1)

        # BOTÃO DE CALIBRAÇÃO
        btn_calib = tk.Button(self.scrollable_frame, text="AJUSTE CURVA DE
REFERENCIA",
                                bg="#ffdd55", fg="black", font=('Arial', 9,
'bold'), command=self.fit_experimental_data)
        btn_calib.pack(fill="x", pady=5, padx=10)

        # 6. FONTE
        lf_src = ttk.LabelFrame(self.scrollable_frame, text="4. Fonte e
Análise", padding=5)
        lf_src.pack(fill="x", pady=5, padx=5)

```

```

        add_input(lf_src, "V Pico:", 179.6, 'Vpico', 0, 0) ;
add_input(lf_src, "Freq:", 60.0, 'f', 0, 1)
        add_input(lf_src, "Ciclos:", 20, 'cycles', 1, 0) ;
add_input(lf_src, "Passo:", 1e-6, 'dt', 1, 1)

        # --- SELETOR DE ONDA ---
        ttk.Label(lf_src, text="Fase Inicial:").grid(row=2, column=0,
sticky="e", padx=2, pady=2)
        self.wave_type = tk.StringVar(value="cos")
        frame_wave = ttk.Frame(lf_src)
        frame_wave.grid(row=2, column=1, sticky="w", columnspan=2)

        # Radiobuttons compactos
        rb1 = ttk.Radiobutton(frame_wave, text="90°",
variable=self.wave_type, value="cos")
        rb1.pack(side="left", padx=2)
        rb2 = ttk.Radiobutton(frame_wave, text="0°", variable=self.wave_type,
value="sin")
        rb2.pack(side="left", padx=2)

        # 7. VISUALIZAÇÃO
        lf_view = ttk.LabelFrame(self.scrollable_frame, text="5. Janela de
Tempo", padding=5)
        lf_view.pack(fill="x", pady=5, padx=5)
        self.view_mode = tk.StringVar(value="transient")
        ttk.Radiobutton(lf_view, text="Inicio", variable=self.view_mode,
value="transient", command=self.refresh_plot_if_ready).grid(row=0, column=0,
padx=2)
        ttk.Radiobutton(lf_view, text="Fim", variable=self.view_mode,
value="steady", command=self.refresh_plot_if_ready).grid(row=0, column=1,
padx=2)
        ttk.Radiobutton(lf_view, text="Tudo", variable=self.view_mode,
value="all", command=self.refresh_plot_if_ready).grid(row=0, column=2,
padx=2)

        # 8. CONTROLE DE PLOTAGEM
        lf_layers = ttk.LabelFrame(self.scrollable_frame, text="6. Controle
de Camadas", padding=5)
        lf_layers.pack(fill="x", pady=5, padx=5)

        ttk.Label(lf_layers, text="Modelo", font=("Arial", 8,
"bold")).grid(row=0, column=0, sticky="w")
        ttk.Label(lf_layers, text="I1", font=("Arial", 8,
"bold")).grid(row=0, column=1, padx=5)
        ttk.Label(lf_layers, text="I2", font=("Arial", 8,
"bold")).grid(row=0, column=2, padx=5)
        ttk.Label(lf_layers, text="Mag", font=("Arial", 8,
"bold")).grid(row=0, column=3, padx=5)

        # Linear
        self.var_lin_i1 = tk.BooleanVar(value=True)
        self.var_lin_i2 = tk.BooleanVar(value=False)
        self.var_lin_b = tk.BooleanVar(value=True)
        ttk.Label(lf_layers, text="Linear", foreground="#007acc",
font=("Arial", 9, "bold")).grid(row=1, column=0, sticky="w", pady=5)
        ttk.Checkbutton(lf_layers, variable=self.var_lin_i1,
command=self.refresh_plot_if_ready).grid(row=1, column=1)
        ttk.Checkbutton(lf_layers, variable=self.var_lin_i2,
command=self.refresh_plot_if_ready).grid(row=1, column=2)

```

```

        ttk.Checkbutton(lf_layers, variable=self.var_lin_b,
command=self.refresh_plot_if_ready).grid(row=1, column=3)

        # Fröhlich
        self.var_fro_i1 = tk.BooleanVar(value=True)
        self.var_fro_i2 = tk.BooleanVar(value=False)
        self.var_fro_b = tk.BooleanVar(value=True)
        ttk.Label(lf_layers, text="Fröhlich", foreground="#cc0000",
font=("Arial", 9, "bold")).grid(row=2, column=0, sticky="w", pady=5)
        ttk.Checkbutton(lf_layers, variable=self.var_fro_i1,
command=self.refresh_plot_if_ready).grid(row=2, column=1)
        ttk.Checkbutton(lf_layers, variable=self.var_fro_i2,
command=self.refresh_plot_if_ready).grid(row=2, column=2)
        ttk.Checkbutton(lf_layers, variable=self.var_fro_b,
command=self.refresh_plot_if_ready).grid(row=2, column=3)

        # Tg Hiperbólica
        self.var_hip_i1 = tk.BooleanVar(value=True)
        self.var_hip_i2 = tk.BooleanVar(value=False)
        self.var_hip_b = tk.BooleanVar(value=True)
        ttk.Label(lf_layers, text="Tg Hiperbólica", foreground="#009900",
font=("Arial", 9, "bold")).grid(row=3, column=0, sticky="w", pady=5)
        ttk.Checkbutton(lf_layers, variable=self.var_hip_i1,
command=self.refresh_plot_if_ready).grid(row=3, column=1)
        ttk.Checkbutton(lf_layers, variable=self.var_hip_i2,
command=self.refresh_plot_if_ready).grid(row=3, column=2)
        ttk.Checkbutton(lf_layers, variable=self.var_hip_b,
command=self.refresh_plot_if_ready).grid(row=3, column=3)

        # CHECKBOX DE PONTOS EXPERIMENTAIS
        self.var_show_exp = tk.BooleanVar(value=True)
        ttk.Checkbutton(lf_layers, text="Mostrar Pontos Exp.",
variable=self.var_show_exp, command=self.refresh_plot_if_ready).grid(row=4,
column=0, colspan=4, pady=5)

        # EXTRAS
        lf_extra = ttk.LabelFrame(self.scrollable_frame, text="Opções
Extras", padding=5)
        lf_extra.pack(fill="x", pady=5, padx=5)
        self.var_fft = tk.BooleanVar(value=False)
        chk_fft = ttk.Checkbutton(lf_extra, text="Calcular FFT",
variable=self.var_fft)
        chk_fft.pack(anchor="w")

        # Botão Rodar
        self.btn_run = tk.Button(self.scrollable_frame, text="RODAR
SIMULAÇÃO", bg="#333", fg="white", font=('Arial', 11, 'bold'),
command=self.start_thread)
        self.btn_run.pack(fill="x", pady=15, padx=5)

        # STATUS
        self.lbl_status = tk.Label(self.scrollable_frame, text="Pronto.",
fg="gray", font=("Arial", 9))
        self.lbl_status.pack(side=tk.BOTTOM, pady=5)

        # --- FUNÇÃO DE CALIBRAÇÃO ---
        def fit_experimental_data(self):
            try:

```

```

mu0 = 4 * np.pi * 1e-7
H_data = DATA_H
B_data = DATA_B

# --- A. AJUSTE LINEAR ---
mask_lin = H_data <= 25
h_lin = H_data[mask_lin]
b_lin = B_data[mask_lin]

def func_linear(h, mur):
    return mur * mu0 * h

popt_lin, _ = curve_fit(func_linear, h_lin, b_lin)
calc_mu_lin = popt_lin[0]

# --- B. AJUSTE FRÖHLICH ---
def func_frohlich(h, a, b):
    return h / (a + b * h)

p0_fro = [1/(5000*mu0), 1/1.8]
popt_fro, _ = curve_fit(func_frohlich, H_data, B_data, p0=p0_fro,
bounds=(0, np.inf))
calc_mu_fro = 1.0 / (popt_fro[0] * mu0)
calc_bsat_fro = 1.0 / popt_fro[1]

# --- C. AJUSTE TANH ---
def func_tanh(h, mur, bsat):
    return bsat * np.tanh( (mur * mu0 * h) / bsat )

p0_tanh = [5000, 1.8]
popt_tanh, _ = curve_fit(func_tanh, H_data, B_data, p0=p0_tanh,
bounds=(0, np.inf))
calc_mu_tanh = popt_tanh[0]
calc_bsat_tanh = popt_tanh[1]

# --- ATUALIZAR INTERFACE ---
# Linear
self.entries['mur_lin'].delete(0, tk.END);
self.entries['mur_lin'].insert(0, f"{calc_mu_lin:.0f}")

# Fröhlich
self.entries['mur_fro'].delete(0, tk.END);
self.entries['mur_fro'].insert(0, f"{calc_mu_fro:.0f}")
self.entries['Bsat_fro'].delete(0, tk.END);
self.entries['Bsat_fro'].insert(0, f"{calc_bsat_fro:.3f}")

# Tanh
self.entries['mur_hip'].delete(0, tk.END);
self.entries['mur_hip'].insert(0, f"{calc_mu_tanh:.0f}")
self.entries['Bsat_hip'].delete(0, tk.END);
self.entries['Bsat_hip'].insert(0, f"{calc_bsat_tanh:.3f}")

messagebox.showinfo("Sucesso",
    f"Calibração Concluída!\n\n"
    f"--- LINEAR ---\n  μr = {calc_mu_lin:.0f}\n\n"
    f"--- FRÖHLICH ---\n  μr = {calc_mu_fro:.0f}\n  Bsat =
{calc_bsat_fro:.3f}\n\n")

```

```

        f"--- TG HIPERBÓLICA ---\n   $\mu_r$  = {calc_mu_tanh:.0f}\n  Bsat =
{calc_bsath_tanh:.3f}")

    except Exception as e:
        messagebox.showerror("Erro na Calibração", str(e))

# --- PLOT SETUP ---
def setup_chart_voltage(self):
    self.fig_v = plt.Figure(figsize=(5, 4), dpi=100)
    self.ax_v = self.fig_v.add_subplot(111)
    self.ax_v.set_title("Tensão de Entrada", fontsize=10, weight='bold')
    self.ax_v.set_ylabel("Tensão (V)")
    self.ax_v.set_xlabel("Tempo (ms)")
    self.ax_v.grid(True, linestyle='--', alpha=0.6)
    self.canvas_v = FigureCanvasTkAgg(self.fig_v,
master=self.tab_voltage)
    self.canvas_v.draw()
    Toolbar(self.canvas_v, self.tab_voltage).pack(side=tk.BOTTOM,
fill=tk.X)
    self.canvas_v.get_tk_widget().pack(side=tk.TOP, fill=tk.BOTH,
expand=True)
    self.cursor_v = CursorManager(self.canvas_v, self.ax_v)

def setup_chart_current(self):
    self.fig_i = plt.Figure(figsize=(5, 4), dpi=100)
    self.ax_i = self.fig_i.add_subplot(111)
    self.ax_i.set_title("Correntes", fontsize=10, weight='bold')
    self.ax_i.set_ylabel("Corrente (A)")
    self.ax_i.set_xlabel("Tempo (ms)")
    self.ax_i.grid(True, linestyle='--', alpha=0.6)
    self.canvas_i = FigureCanvasTkAgg(self.fig_i,
master=self.tab_current)
    self.canvas_i.draw()
    Toolbar(self.canvas_i, self.tab_current).pack(side=tk.BOTTOM,
fill=tk.X)
    self.canvas_i.get_tk_widget().pack(side=tk.TOP, fill=tk.BOTH,
expand=True)
    self.cursor_i = CursorManager(self.canvas_i, self.ax_i)

def setup_chart_bxh(self):
    self.fig_b = plt.Figure(figsize=(5, 4), dpi=100)
    self.ax_b = self.fig_b.add_subplot(111)
    self.ax_b.set_title("Magnetização", fontsize=10, weight='bold')
    self.ax_b.set_ylabel("indução B (T)")
    self.ax_b.set_xlabel("Campo H (A/m)")
    self.ax_b.grid(True, linestyle='--', alpha=0.6)
    self.canvas_b = FigureCanvasTkAgg(self.fig_b, master=self.tab_bxh)
    self.canvas_b.draw()
    Toolbar(self.canvas_b, self.tab_bxh).pack(side=tk.BOTTOM, fill=tk.X)
    self.canvas_b.get_tk_widget().pack(side=tk.TOP, fill=tk.BOTH,
expand=True)
    self.cursor_b = CursorManager(self.canvas_b, self.ax_b)

def setup_chart_fft(self):
    self.fig_f = plt.Figure(figsize=(5, 4), dpi=100)
    self.ax_f = self.fig_f.add_subplot(111)
    self.ax_f.set_title("FFT (Requer 'Calcular FFT' ativo)", fontsize=10,
weight='bold')
    self.ax_f.set_ylabel("Amp")

```

```

self.ax_f.set_xlabel("Hz")
self.ax_f.grid(True, linestyle='--', alpha=0.6)
self.canvas_f = FigureCanvasTkAgg(self.fig_f, master=self.tab_fft)
self.canvas_f.draw()
Toolbar(self.canvas_f, self.tab_fft).pack(side=tk.BOTTOM, fill=tk.X)
self.canvas_f.get_tk_widget().pack(side=tk.TOP, fill=tk.BOTH,
expand=True)
self.cursor_f = CursorManager(self.canvas_f, self.ax_f)

def on_closing(self):
    self.is_running = False
    self.root.destroy()
    sys.exit(0)

def refresh_plot_if_ready(self):
    if self.last_results is not None:
        self.plot_results(self.last_results)

def start_thread(self):
    if self.is_running: return
    self.is_running = True
    self.btn_run.config(state=tk.DISABLED)
    self.lbl_status.config(text="CALCULANDO...", fg="blue")
    t = threading.Thread(target=self.run_simulation, daemon=True)
    t.start()

# --- SIMULAÇÃO ---
def run_simulation(self):
    try:
        def get_val(key): return float(self.entries[key].get())
        Vpico, f, dt = get_val('Vpico'), get_val('f'), get_val('dt')
        cycles = get_val('cycles')
        if cycles <= 0: cycles = 1.0
        tf = cycles / f

        R1, L1, N1 = get_val('R1'), get_val('L1'), get_val('N1')
        R2, L2, N2 = get_val('R2'), get_val('L2'), get_val('N2')
        Rcarga = get_val('Rcarga')

        # Parametros Compartilhados
        S, lm = get_val('S'), get_val('lm')
        clamp = 0.999
        mu0 = 4 * np.pi * 1e-7

        # Parametros Individuais
        mur_lin = get_val('mur_lin')

        mur_fro = get_val('mur_fro')
        Bsat_fro = get_val('Bsat_fro')

        mur_hip = get_val('mur_hip')
        Bsat_hip = get_val('Bsat_hip')

        num_passos = int(tf / dt) + 1
        t_vec = np.linspace(0, tf, num_passos)

        # --- SELEÇÃO DE ONDA ---
        if self.wave_type.get() == 'sin':
            e1_vec = Vpico * np.sin(2 * np.pi * f * t_vec)
        else:
            e1_vec = Vpico * np.cos(2 * np.pi * f * t_vec)

```

```

def run_rk4(model_type):
    i1, i2, phi = np.zeros(num_passos), np.zeros(num_passos),
np.zeros(num_passos)

    # Setup Específico de cada Modelo
    if model_type == 'frolich':
        pa = 1.0 / (mur_fro * mu0)
        pb = 1.0 / Bsat_fro

    elif model_type == 'linear':
        R_const = lm / (mur_lin * mu0 * S)

    elif model_type == 'tghip':
        beta = Bsat_hip
        mu_in = mur_hip * mu0

    def calc_derivs(i1_v, i2_v, phi_v, v_fonte):
        if model_type == 'frolich':
            B_val = phi_v / S
            limit = Bsat_fro * clamp
            if abs(B_val) >= limit: B_val = np.sign(B_val) *
limit

            den = (1.0 - pb * abs(B_val))
            if den < 1e-9: den = 1e-9
            dh = pa / (den * den)
            R_din = (lm / S) * dh

        elif model_type == 'linear':
            R_din = R_const

        elif model_type == 'tghip':
            B_val = phi_v / S
            limit = beta * clamp
            if abs(B_val) >= limit: B_val = np.sign(B_val) *
limit

            ratio = (B_val / beta)**2
            denom = mu_in * (1.0 - ratio)
            if denom < 1e-9: denom = 1e-9
            dh = 1.0 / denom
            R_din = (lm / S) * dh

        tA = N1 / L1
        tB = N2 / L2
        u1 = v_fonte - R1 * i1_v
        u2 = -(Rcarga + R2) * i2_v

        num = (tA * u1) - (tB * u2)
        den = (tA * N1) + (tB * N2) + R_din
        dphi = num / den
        di1 = (u1 - N1 * dphi) / L1
        di2 = (u2 + N2 * dphi) / L2
        return di1, di2, dphi

    for k in range(1, num_passos):
        if not self.is_running: break
        v_now = e1_vec[k-1]
        io1, io2, pho = i1[k-1], i2[k-1], phi[k-1]

```

```

        k1a, k1b, k1c = calc_derivs(io1, io2, pho, v_now)
        k2a, k2b, k2c = calc_derivs(io1 + 0.5*dt*k1a, io2 +
0.5*dt*k1b, pho + 0.5*dt*k1c, v_now)
        k3a, k3b, k3c = calc_derivs(io1 + 0.5*dt*k2a, io2 +
0.5*dt*k2b, pho + 0.5*dt*k2c, v_now)
        k4a, k4b, k4c = calc_derivs(io1 + dt*k3a, io2 + dt*k3b,
pho + dt*k3c, v_now)

        i1[k] = io1 + (dt/6.0)*(k1a + 2*k2a + 2*k3a + k4a)
        i2[k] = io2 + (dt/6.0)*(k1b + 2*k2b + 2*k3b + k4b)
        phi[k] = pho + (dt/6.0)*(k1c + 2*k2c + 2*k3c + k4c)

    return (i1, i2, (N1*i1 - N2*i2)/lm, phi/S)

res_lin = run_rk4('linear')
res_fro = run_rk4('frolich')
res_hip = run_rk4('tghip')

# FFT Opcional
fft_lin, fft_fro, fft_hip = (None,None), (None,None), (None,None)

if self.var_fft.get():
    def get_fft(vec):
        N = len(vec)
        # Downsample se for muito grande
        if N > 200000:
            fac = N // 200000
            vec = vec[::fac]
            dtt = dt * fac
            N = len(vec)
        else: dtt = dt
        yf = np.fft.fft(vec)
        xf = np.fft.fftfreq(N, dtt)[:N//2]
        mag = 2.0/N * np.abs(yf[0:N//2])
        mag[0] /= 2.0
        return xf, mag

    fft_lin = get_fft(res_lin[0])
    fft_fro = get_fft(res_fro[0])
    fft_hip = get_fft(res_hip[0])

self.last_results = {
    't': t_vec, 'el': el_vec,
    'lin': res_lin, 'fro': res_fro, 'hip': res_hip,
    'ffts': {'lin': fft_lin, 'fro': fft_fro, 'hip': fft_hip},
    'bsat': Bsat_fro
}

if self.is_running:
    self.root.after(0, self.plot_results, self.last_results)

except Exception as e:
    if self.is_running:
        self.root.after(0, self.show_error, str(e))

def plot_results(self, results):
    self.btn_run.config(state=tk.NORMAL)
    self.lbl_status.config(text="Concluído.", fg="green")
    self.is_running = False

mode = self.view_mode.get()

```

```

t = results['t']
e1 = results['e1']

# Recorte
total = len(t)
dt_val = float(self.entries['dt'].get())
freq = float(self.entries['f'].get())
pts_ciclo = int(1.0/(freq*dt_val))
visiveis = 5 * pts_ciclo

s, e = 0, total

if mode == 'transient':
    e = min(total, visiveis)
elif mode == 'steady':
    if total > visiveis: s = total - visiveis

step = max(1, (e-s)//5000)
sl = slice(s, e, step)
t_p = t[sl] * 1000
e1_p = e1[sl]

# 1. Tensão
self.ax_v.clear()
self.ax_v.plot(t_p, e1_p, 'b', lw=1)
self.ax_v.set_title(f"Tensão de Entrada", fontsize=9, weight='bold')
self.ax_v.set_ylabel("Tensão (V)")
self.ax_v.set_xlabel("Tempo (ms)")
self.ax_v.grid(True, linestyle='--', alpha=0.6)
if len(t_p) > 0: self.ax_v.set_xlim([t_p[0], t_p[-1]])
self.canvas_v.draw()

# 2. Corrente
self.ax_i.clear()

has_lines = False

# Lin
if self.var_lin_i1.get():
    self.ax_i.plot(t_p, results['lin'][0][sl], '#4da6ff',
label='Lin(I1)', lw=1)
    has_lines = True
if self.var_lin_i2.get():
    self.ax_i.plot(t_p, results['lin'][1][sl], '#4da6ff', ls='--',
label='Lin(I2)', lw=1)
    has_lines = True

# Fro
if self.var_fro_i1.get():
    self.ax_i.plot(t_p, results['fro'][0][sl], 'red',
label='Fro(I1)', lw=1.2)
    has_lines = True
if self.var_fro_i2.get():
    self.ax_i.plot(t_p, results['fro'][1][sl], 'darkred', ls='--',
label='Fro(I2)', lw=1)
    has_lines = True

# Hip
if self.var_hip_i1.get():
    self.ax_i.plot(t_p, results['hip'][0][sl], 'green',
label='TgHyp(I1)', lw=1)

```

```

        has_lines = True
        if self.var_hip_i2.get():
            self.ax_i.plot(t_p, results['hip'][1][sl], 'darkgreen', ls='--',
label='TgHyp(I2)', lw=1)
            has_lines = True

        self.ax_i.set_title("Correntes", fontsize=9, weight='bold')
        self.ax_i.set_ylabel("Corrente (A)")
        self.ax_i.set_xlabel("Tempo (ms)")
        if has_lines: self.ax_i.legend(loc='upper right', fontsize='small')
        self.ax_i.grid(True, ls='--', alpha=0.6)
        if len(t_p) > 0: self.ax_i.set_xlim([t_p[0], t_p[-1]])
        self.canvas_i.draw()

    # 3. BxH
    self.ax_b.clear()
    has_lines_b = False

    # PLOTAR PONTOS EXPERIMENTAIS
    if self.var_show_exp.get():
        self.ax_b.scatter(DATA_H, DATA_B, color='black', s=10,
marker='x', label='Exp.', zorder=5)
        has_lines_b = True

    if self.var_lin_b.get():
        self.ax_b.plot(results['lin'][2][sl], results['lin'][3][sl],
'#4da6ff', label='Lin', lw=1)
        has_lines_b = True
    if self.var_fro_b.get():
        self.ax_b.plot(results['fro'][2][sl], results['fro'][3][sl],
'red', label='Fro', lw=1.5)
        has_lines_b = True
    if self.var_hip_b.get():
        self.ax_b.plot(results['hip'][2][sl], results['hip'][3][sl],
'green', label='TgHyp', lw=1)
        has_lines_b = True

    self.ax_b.set_title("Magnetização", fontsize=9, weight='bold')
    self.ax_b.set_ylabel("Indução B (T)")
    self.ax_b.set_xlabel("Campo H (A/m)")
    if has_lines_b: self.ax_b.legend(loc='lower right', fontsize='small')
    self.ax_b.grid(True, ls='--', alpha=0.6)
    self.canvas_b.draw()

    # 4. FFT
    self.ax_f.clear()
    if not self.var_fft.get():
        self.ax_f.text(0.5, 0.5, "Cálculo FFT Desativado", ha='center',
transform=self.ax_f.transAxes)
    else:
        freq_fund = float(self.entries['f'].get())
        max_f = 600

        harmonics = np.arange(0, max_f + 1, freq_fund)
        self.ax_f.set_xticks(harmonics)

        labels = ["DC" if h == 0 else f"{int(h)}" for h in harmonics]
        self.ax_f.set_xticklabels(labels, rotation=0, fontsize=9)

    # Função interna para plotar com deslocamento
    def pfft(d, c, l, shift=0):

```

```

        if d[0] is None: return
        xf, yf = d

        idx = [np.abs(xf - h).argmin() for h in harmonics]

        # Filtrar indices validos
        valid_x = []
        valid_y = []
        for i in idx:
            if i < len(yf):
                valid_x.append(xf[i])
                valid_y.append(yf[i])

        # Aplicar o Shift apenas visualmente
        vx = np.array(valid_x) + shift
        vy = np.array(valid_y)

        if len(vx) > 0:
            self.ax_f.vlines(x=vx, ymin=0, ymax=vy, color=c,
alpha=0.6, linewidth=2, label=1)
            self.ax_f.plot(vx, vy, color=c, marker='o',
linestyle='None', markersize=4)

            # Chamadas com Deslocamento (Shift)
            # Azul um pouco pra esquerda, Vermelho no centro, Verde pra
direita
            if self.var_lin_il.get(): pfft(results['ffts']['lin'], '#4da6ff',
'Lin', shift=-4)
            if self.var_fro_il.get(): pfft(results['ffts']['fro'], 'red',
'Fro', shift=0)
            if self.var_hip_il.get(): pfft(results['ffts']['hip'], 'green',
'TgHyp', shift=+4)

            self.ax_f.set_xlim(left=-10, right=max_f)
            if has_lines: self.ax_f.legend(loc='upper right',
fontsize='small')

            self.ax_f.set_title("Espectro de Frequência (I1)", fontsize=9,
weight='bold')
            self.ax_f.set_ylabel("Amplitude (A)")
            self.ax_f.set_xlabel("Frequência (Hz)")
            self.ax_f.grid(True, ls='--', alpha=0.3)
            self.canvas_f.draw()

        # Refresh Cursors
        self.cursor_v = CursorManager(self.canvas_v, self.ax_v)
        self.cursor_i = CursorManager(self.canvas_i, self.ax_i)
        self.cursor_b = CursorManager(self.canvas_b, self.ax_b)
        self.cursor_f = CursorManager(self.canvas_f, self.ax_f)

    def show_error(self, msg):
        self.btn_run.config(state=tk.NORMAL)
        self.is_running = False
        self.lbl_status.config(text="Erro!", fg="red")
        messagebox.showerror("Erro", msg)

if __name__ == "__main__":
    root = tk.Tk()
    app = SimuladorApp(root)

```

```
root.mainloop()
```