

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

**LUÍS VÍTOR DA COSTA BOHRER**

**MODELAGEM DE SISTEMAS PASSIVOS VIA IMPOSIÇÃO DE PASSIVIDADE  
A PARTIR DO MÉTODO DAS FRAÇÕES POSITIVAS**

**CURITIBA**

**2025**

**LUÍS VÍTOR DA COSTA BOHRER**

**MODELAGEM DE SISTEMAS PASSIVOS VIA IMPOSIÇÃO DE PASSIVIDADE  
A PARTIR DO MÉTODO DAS FRAÇÕES POSITIVAS**

**Modeling of Passive Systems via Passivity Enforcement based on the  
Positive Fractions Method**

Trabalho de conclusão de curso de graduação  
apresentado como requisito para obtenção do  
título de Bacharel em Engenharia de Controle e  
Automação do curso de Engenharia de Controle  
e Automação da Universidade Tecnológica  
Federal do Paraná (UTFPR).

Orientador(a): Prof. Dr. Lucas Pioli Rehbein  
Kurten Ihlenfeld

**CURITIBA**

**2025**



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

**LUÍS VÍTOR DA COSTA BOHRER**

**MODELAGEM DE SISTEMAS PASSIVOS VIA IMPOSIÇÃO DE PASSIVIDADE  
A BASEADA NO MÉTODO DAS FRAÇÕES POSITIVAS**

Trabalho de conclusão de curso de graduação  
apresentado como requisito para obtenção do  
título de Bacharel em Engenharia de Controle e  
Automação do curso de Engenharia de Controle  
e Automação da Universidade Tecnológica  
Federal do Paraná (UTFPR).

Data de aprovação: 26 / novembro / 2025

---

Lucas Pioli Rehbein Kurten Ihlenfeld  
Doutorado  
Universidade Tecnológica Federal do Paraná

---

Narco Afonso Ravazzoli Maciejewski  
Doutorado  
Universidade Tecnológica Federal do Paraná

---

Victor Baptista Frencl  
Doutorado  
Universidade Tecnológica Federal do Paraná

**CURITIBA  
2025**

## RESUMO

Este trabalho de conclusão de curso apresenta uma análise comparativa entre duas abordagens de imposição de passividade para modelos de sistemas lineares intransientes no tempo (SLIT) oriundos de um processo de identificação de sistemas a partir de dados experimentais no domínio da frequência. Utilizando dados experimentais de resposta em frequência de um transformador, foram empregadas técnicas de identificação baseadas no algoritmo *Vector Fitting* para obtenção de representações racionais equivalentes. A partir desses modelos, avaliou-se a presença de violações de passividade e aplicaram-se dois métodos perturbativos para correção: o método baseado no conceito de *Positive Realness*, disponível no ambiente do MATLAB com a *Matrix Fitting Toolbox*, e o método das Frações Positivas, desenvolvido neste trabalho com o auxílio do pacote de otimização convexa CVX. Os resultados indicaram que ambos os métodos foram capazes de garantir a passividade dos modelos, sendo o método baseado na *Positive Realness* o que apresentou menor erro médio e maior aderência às medições originais. O método das Frações Positivas, por sua vez, apresentou erro superior. Ainda assim, a implementação demonstrou-se funcional em casos com uma entrada e uma saída (SISO) e contribuiu para o avanço das técnicas de modelagem passiva, oferecendo uma base para futuras aplicações em sistemas com múltiplas entradas e múltiplas saídas (MIMO) e integração com ferramentas de simulação.

Palavras-chave: passividade; *vector fitting*; frações positivas; *cvx*.

## **ABSTRACT**

This undergraduate thesis presents a comparative analysis between two approaches for enforcing passivity in linear time invariant (LTI) models obtained from system identification using experimental frequency domain data. Experimental frequency response data from a transformer was used, and identification techniques based on the Vector Fitting algorithm were employed to derive equivalent rational representations. From these models, the presence of passivity violations was assessed and two perturbative correction methods were applied: the Positive Realness based method available in the MATLAB environment with the Matrix Fitting Toolbox, and the Positive Fractions method, developed in this work using the convex optimization package CVX. The results showed that both methods successfully ensured the passivity of the models, with the Positive Realness based approach achieving lower average error and better adherence to the original measurements. The Positive Fractions method, on the other hand, exhibited higher error. Nevertheless, its implementation proved functional in single-input single-output (SISO) cases and contributes to the advancement of passive modeling techniques, providing a foundation for future applications in multiple-input multiple-output (MIMO) systems and integration with simulation tools.

Keywords: passivity; vector fitting; positive fractions; cvx.

## LISTA DE FIGURAS

<b>Figura 1 – Comparação de simulações de um modelo não passivo e um passivo ....</b>	<b>11</b>
<b>Figura 2 – Diagrama de medições de um VNA (DET representa o dispositivo em teste) .....</b>	<b>15</b>
<b>Figura 3 – Exemplo de uma função suave .....</b>	<b>18</b>
<b>Figura 4 – Exemplo de uma função não suave .....</b>	<b>19</b>
<b>Figura 5 – Exemplo de violação de passividade entre pontos de frequência medidos</b>	<b>21</b>
<b>Figura 6 – Representação da expansão usada no método das frações positivas .....</b>	<b>23</b>
<b>Figura 7 – Exemplo de implementação de mínimos quadrados no CVX .....</b>	<b>25</b>
<b>Figura 8 – Fluxograma do procedimento para obtenção de modelos passivos .....</b>	<b>26</b>
<b>Figura 9 – Código em MATLAB usado para a visualização das violações de passividade .....</b>	<b>27</b>
<b>Figura 10 – Resposta em frequência do transformador .....</b>	<b>28</b>
<b>Figura 11 – Autovalores da parte real dos dados de transformador .....</b>	<b>28</b>
<b>Figura 12 – Código em MATLAB usado para identificar um modelo e impor passividade a partir de rotinas de código aberto consolidadas .....</b>	<b>29</b>
<b>Figura 13 – Critério de escolha de Akaike .....</b>	<b>29</b>
<b>Figura 14 – Critério de escolha de Bayes .....</b>	<b>30</b>
<b>Figura 15 – AIC corrigido .....</b>	<b>30</b>
<b>Figura 16 – Modelo do sistema SISO com 26 polos .....</b>	<b>31</b>
<b>Figura 17 – Autovalores da parte real do modelo identificado .....</b>	<b>31</b>
<b>Figura 18 – Modelo do sistema SISO passivado pelo RPdriver .....</b>	<b>32</b>
<b>Figura 19 – Autovalores da parte real do modelo perturbado com o RPdriver .....</b>	<b>32</b>
<b>Figura 20 – Código em MATLAB usado para implementar o problema de otimização do método das frações positivas .....</b>	<b>33</b>
<b>Figura 21 – Modelo do sistema SISO passivado pelo método das frações positivas ..</b>	<b>34</b>
<b>Figura 22 – Autovalores da parte real do modelo perturbado com o método das frações positivas .....</b>	<b>34</b>
<b>Figura 23 – Erros ponto a ponto dos modelos .....</b>	<b>35</b>

## LISTA DE TABELAS

Tabela 1 – Síntese dos resultados dos modelos .....	34
---	----

## LISTA DE SIGLAS

AIC	Critério da Informação de Akaike (do inglês <i>Akaike's Information Criterion</i> )
AICc	AIC Corrigido
BIBO	Entrada limitada e saída limitada (do inglês <i>Bounded Input Bounded Output</i> )
BIC	Critério da Informação de Bayes (do inglês <i>Bayesian Information Criterion</i> )
CVX	Pacote de otimização do ambiente MATLAB (nomeado do inglês <i>convex</i> )
MATLAB	Laboratório de matrizes (do inglês <i>Matrix Laboratory</i> )
MIMO	Múltiplas entradas e múltiplas saídas (do inglês <i>Multiple Input Multiple Output</i> )
MISO	Múltiplas entradas e uma saída (do inglês <i>Multiple Input Single Output</i> )
PRL	Lema positivo real (do inglês <i>Positive Real Lemma</i> )
SIMO	Uma entrada e múltiplas saídas (do inglês <i>Single Input Multiple Output</i> )
SINTEF	Fundação para Pesquisa Industrial e Técnica (do norueguês <i>Stiftelsen for Industriell og Teknisk Forskning</i> )
SISO	Uma entrada e uma saída (do inglês <i>Single Input Single Output</i> )
SLIT	Sistema Linear Invariante no Tempo



## SUMÁRIO

1	INTRODUÇÃO .....	9
1.1	Problematização .....	9
1.2	Objetivos .....	11
1.2.1	Objetivo Geral .....	11
1.2.2	Objetivos Específicos .....	11
1.3	Justificativa .....	12
1.4	Metodologia da Pesquisa .....	13
2	DESENVOLVIMENTO .....	15
2.1	Aquisição de dados .....	15
2.2	<i>Vector Fitting</i> .....	16
2.3	Passividade .....	20
2.3.1	Verificação de Passividade .....	20
2.3.1.1	Métodos Dependentes da Frequência .....	20
2.3.1.2	Métodos Independentes da Frequência .....	21
2.3.2	Imposição de Passividade .....	22
2.3.2.1	Método Baseado na <i>Positive Realness</i> .....	22
2.3.2.2	Método Baseado em Frações Positivas .....	23
2.4	Pacote de Otimização Convexa (CVX) .....	25
2.5	Resumo dos Procedimentos .....	26
3	RESULTADOS E DISCUSSÕES .....	27
3.1	Análise Exploratória da Passividade dos Dados .....	27
3.2	Aplicação das Rotinas da <i>Matrix Fitting Toolbox</i> .....	28
3.3	Método das Frações Positivas .....	33
4	CONCLUSÃO .....	36
	REFERÊNCIAS .....	37

## 1 INTRODUÇÃO

### 1.1 Problematização

O processo de modelagem matemática consiste na tradução de problemas reais em relações matemáticas abstratas. Tais problemas podem ser de natureza física, biológica, química, econômica, sociológica, geológica, entre outras (Eck; Garcke; Knabner, 2017). Assim, modelos matemáticos são uma ferramenta poderosa para a análise de problemas complexos e generalistas. Além disso, suas aplicações são vastas, podendo ser implementados em áreas como simulação, previsão e projeto de sistemas de controle (Aguirre, 2007).

Sendo assim, uma grande área que faz o uso constante de conceitos de modelagem matemática é a teoria de sistemas. Para entender mais sobre essa área, é preciso compreender o que são sistemas e como eles podem ser caracterizados. Zadeh (1962) parte da definição do dicionário *Webster*, chegando à conclusão de que qualquer conjunto de elementos que interagem entre si pode ser chamado de sistema, desde partículas que se atraem até humanos em sociedade. Tal definição é bastante generalista, assim como a definição de modelagem de Eck, Garcke e Knabner (2017) abordada anteriormente.

Mais especificamente no cenário da área de controle, os sistemas podem ser vistos como processos que respondem a sinais de entrada e geram outros sinais de saída (Oppenheim; Willsky, 2010). A teoria de sistemas, então, apresenta maior preocupação em extrair relações matemáticas que podem ser estabelecidas nesses sistemas, independentemente de suas formas físicas. Dentre os tópicos de estudo dessa área, está a identificação de sistemas, que se preocupa em determinar a classe específica de um sistema baseando-se em observações de suas entradas e saídas (Zadeh, 1962).

De acordo com Aguirre (2007), as técnicas de identificação de sistemas podem ser separadas em três grandes grupos. Primeiramente, a modelagem caixa branca parte exclusivamente do uso de leis físicas e relações matemáticas que regem o sistema analisado. Para tal, é preciso ter amplo conhecimento sobre a natureza do processo e as variáveis que podem influenciá-la. Esse é, também, o grande limitador dessa categoria, visto que a dificuldade técnica para sua aplicação cresce junto com a complexidade do sistema estudado, tanto em relação ao entendimento da teoria necessária, quanto em relação ao tempo necessário para sua implementação.

No outro lado do espectro, a modelagem caixa preta é feita somente a partir da observação dos sinais de entrada e de saída do sistema. Assim, o levantamento do modelo é feito de modo empírico, sem a necessidade de qualquer conhecimento prévio sobre o sistema analisado. Finalmente, a modelagem caixa cinza dá-se pelo uso de dados ou informações que complementam as informações de entrada e saída do sistema, mas não o descrevem completamente por si só; logo, essa categoria pode ser vista como intermediária entre a modelagem caixa branca e modelagem caixa preta.

Neste trabalho será explorado um algoritmo de identificação caixa preta chamado *vector fitting*, tendo em vista sua robustez, eficiência e ampla disseminação na comunidade de identificação de sistemas no domínio da frequência (Gustavsen; Semlyen, 1999).

Assim, os sistemas podem apresentar diversas propriedades que os caracterizam e definem como suas respostas comportam-se em relação a entradas específicas. Neste trabalho será feita uma análise da passividade de sistemas. Para defini-la, pode-se partir do conceito de estabilidade. Segundo Nise (2013), a estabilidade pode ser definida usando a condição de entrada limitada e saída limitada (BIBO), que diz que um sistema será estável se, ao se fornecer entradas limitadas e finitas, as saídas também serão limitadas e finitas.

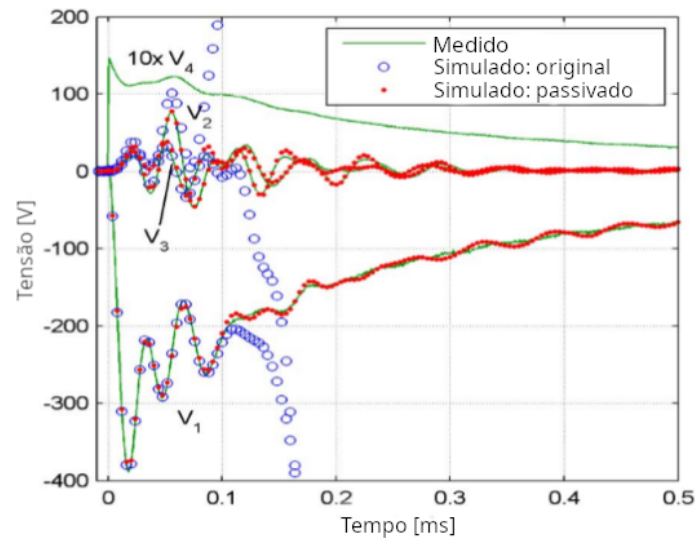
Ao contrário da passividade, a estabilidade não possui a propriedade de fechamento algébrico. Isso significa que a interligação de sistemas estáveis nem sempre resulta em outro sistema estável. Dessa forma, a passividade pode ser vista como uma generalização do conceito de estabilidade, motivo pelo qual ela pode ser chamada, também, de super-estabilidade (Udwadia, 2005). Um sistema é dito passivo quando ele apenas armazena ou dissipa energia, ou seja, é um sistema que não produz energia e, portanto, todo sistema passivo será estável, enquanto a recíproca não é verdadeira. Além disso, a passividade também garante o fechamento algébrico das operações com o sistema, ou seja, qualquer tipo de interligação de sistemas passivos sempre gera outros sistemas igualmente passivos (Willems, 2007).

Ademais, tendo em vista que a passividade implica tanto estabilidade quanto causalidade de sistemas, apesar da recíproca não ser verdadeira, a passividade pode ser vista como a propriedade mais forte e geral dentre as três. Entretanto, a verificação da passividade de um sistema é tecnicamente mais complexa e sua imposição, ao contrário da estabilidade e causalidade, não é feita na maior parte dos procedimentos de identificação de sistemas.

Modelos com violação de passividade podem apresentar uma boa resposta segundo métricas tradicionais, mas também levantar respostas muito adversas em alguns cenários, como demonstrado pelo exemplo da Figura 1. Nesse gráfico, pode-se observar que a resposta do modelo com violação de passividade (em azul) apresentou resultados instáveis na simulação, enquanto o modelo passivo (em vermelho) apresentou simulações mais coerentes com as medições reais (em verde).

A violação de passividade pode ser resolvida a partir de métodos de imposição de passividade. Tais procedimentos são formulados de maneira perturbativa ou não perturbativa. Métodos perturbativos procuram encontrar perturbações paramétricas mínimas que, quando aplicadas no modelo não passivo, tornam-o passivo. Métodos não perturbativos, por sua vez, procuram encontrar um novo modelo via estimação de parâmetros com restrições de passividade. Assim, embora ambos resultem em modelos passivos, métodos perturbativos possuem a característica de se aproximar de um modelo de base enquanto métodos não perturbativos aproximam-se de dados de medição. Com isso, neste trabalho serão analisados dois métodos perturbativos de imposição de passividade, o primeiro se baseia na perturbação de autovalores da função de transferência do sistema e possui uma implementação em código aberto (Gustavsen, 2009) na rotina *RPdriver.m*, em MATLAB. Essa implementação é consolidada e validada

**Figura 1 – Comparação de simulações de um modelo não passivo e um passivo**



**Fonte: Adaptado de Gustavsen (2008).**

por outros trabalhos acadêmicos, como em Achkar *et al.* (2025) e Mouhaidali *et al.* (2021). Para o segundo método, chamado de Método das Frações Positivas, será feita uma implementação própria em MATLAB, cuja resposta será comparada com a resposta do primeiro método.

## 1.2 Objetivos

### 1.2.1 Objetivo Geral

Aplicar os conceitos de verificação e imposição de passividade por meio da implementação e validação de um algoritmo de imposição de passividade baseado no método das frações positivas.

### 1.2.2 Objetivos Específicos

1. Revisar a bibliografia do estado da arte no campo de identificação de modelos passivos;
2. Análise exploratória da passividade dos dados disponíveis;
3. Realizar a identificação de sistemas para os dados disponíveis via aplicação do *vector fitting*;
4. Realizar a imposição de passividade de forma perturbativa por meio da rotina *RPdriver.m*;
5. Implementar um algoritmo de imposição de passividade de forma perturbativa a partir do método das frações positivas;

6. Validar a implementação desenvolvida, pela comparação dos modelos resultantes dos dois métodos;

### 1.3 Justificativa

Tendo em vista a importância das técnicas e processos de modelagem de sistemas dinâmicos para diversas áreas nos tempos atuais para o desenvolvimento tecnológico e científico, esse trabalho visa explorar a passividade de modelos matemáticos de sistemas.

A representação matemática precisa de sistemas físicos é um elemento fundamental para análises, simulações e estudos de desempenho em diversos ramos da engenharia. Como qualquer sistema dinâmico físico possui comportamento intrinsecamente passivo, ou seja, incapaz de gerar energia espontaneamente, as técnicas de verificação de passividade são ferramentas poderosas para filtragem de dados de boa qualidade, visto que elas podem explicitar erros de medição, ruídos, limitações do equipamento ou inadequações na ordem do modelo resultem em violações de passividade, comprometendo a estabilidade e a utilidade do modelo final.

Ademais, garantir a passividade de modelos a partir de técnicas de imposição de passividade em ambientes industriais e de pesquisa é essencial para evitar instabilidades numéricas em simulações de transitórios eletromagnéticos e para assegurar coerência física em aplicações avançadas, como gêmeos digitais (digital twins) e sombras digitais (digital shadows). Nessas aplicações, modelos matemáticos são constantemente atualizados ou validados a partir de dados reais, e devem refletir com alta fidelidade o comportamento físico do sistema representado. Um modelo não passivo pode levar a previsões incorretas, divergência numérica e falhas no processo de tomada de decisão automatizada.

Os gêmeos digitais, em especial, dependem de modelos estáveis e energeticamente coerentes para simular, prever e otimizar o funcionamento de ativos industriais em tempo real. Transformadores, inversores, motores, sistemas de distribuição e redes de alta tensão são frequentemente monitorados por meio de tais tecnologias. Sem a garantia de passividade, um gêmeo digital pode apresentar oscilações artificiais, regimes instáveis ou respostas fisicamente impossíveis, prejudicando análises preditivas e estratégias de manutenção. Assim, a verificação e imposição de passividade tornam-se elementos chave para a confiabilidade dos sistemas de monitoramento avançado.

Da mesma forma, sombras digitais, que funcionam como representações continuamente alimentadas por dados reais, mas com menor complexidade que os gêmeos digitais completos, também exigem modelos passivos. Nesses casos, a exigência é ainda mais crítica, pois as sombras digitais costumam operar com modelos reduzidos e atualizados de forma incremental — e pequenas violações de passividade podem ser amplificadas quando o modelo é utilizado para simulações rápidas ou integradas a malhas de controle em tempo real.

Diante desse cenário, justifica-se o desenvolvimento e a avaliação de métodos de verificação e imposição de passividade aplicados a modelos identificados no domínio da frequência. Mesmo quando o modelo identificado reproduz adequadamente o comportamento medido, qual-

quer violação de passividade pode inviabilizar seu uso em aplicações críticas. Assim, este trabalho se justifica tanto pela importância fundamental da passividade na modelagem de sistemas físicos quanto pela crescente necessidade de modelos confiáveis para aplicações avançadas baseadas em dados, em especial os gêmeos digitais e as sombras digitais, consolidando sua relevância científica, tecnológica e industrial.

#### 1.4 Metodologia da Pesquisa

De acordo com a definição de Azevedo e Ensslin (2020), este trabalho consiste em uma pesquisa teórica conceitual aplicada de natureza descritiva e lógica experimental. Primeiramente, foi feito um levantamento bibliográfico do estado da arte da modelagem de sistemas passivos.

Então, foram utilizados dados experimentais de admitância de um transformador, obtidos a partir de um analisador de rede vetorial (VNA). Os dados continham violações de passividade causadas pela má calibração do VNA, que foram explicitadas a partir da análise dos autovalores da parte real da resposta em frequência medida.

A partir desses dados, foi aplicada a rotina *VFdriver.m*, disponibilizada em código aberto pela Fundação para Pesquisa Industrial e Técnica (SINTEF) na *Matrix Fitting Toolbox* do software MATLAB. A ordem do modelo foi decidida com base nos critérios de escolha de Akaike (AIC) e Bayes (BIC), bem como a forma corrigida do critério de Akaike (AICc) (Ljung, 1999), que corrige a pontuação do AIC levando em conta a quantidade de dados usados. Eles são calculados da seguinte forma:

$$AIC = N \log_{10} \left[ \frac{1}{N} \sum_{i=1}^N |(y_i - \hat{y}_i)|^2 \right] + 2n_p + N[\log_{10}(2\pi) + 1], \quad (1)$$

$$BIC = N \log_{10} \left[ \frac{1}{N} \sum_{i=1}^N |(y_i - \hat{y}_i)|^2 \right] + n_p \log_{10}(N) + N[\log_{10}(2\pi) + 1], \quad (2)$$

$$AICc = AIC + 2n_p \frac{n_p + 1}{N - n_p - 1}. \quad (3)$$

em que  $N$  é a quantidade de pontos de frequência medidos,  $y_i$  é a resposta em frequência original,  $\hat{y}_i$  é a resposta em frequência do modelo e  $n_p$  é a ordem do modelo. Assim, esses critérios avaliam os modelos penalizando o erro apresentado e sua complexidade, com base na ordem, de forma que valores mais baixos indicam modelos que apresentam um melhor compromisso entre complexidade e precisão. Com isso, foi obtido um modelo estável e causal, mas ainda com as violações de passividade presentes nos dados.

Então, foi feita a aplicação da rotina *RPdriver.m*, também disponibilizada na *Matrix Fitting Toolbox*, que faz a imposição de passividade no modelo identificado. Também foi feita a

imposição de passividade utilizando a implementação desenvolvida com base no método das frações positivas. Os modelos resultantes foram comparados quantitativamente usando a raiz quadrada do erro quadrático médio (RMSE) e sua versão normalizada pela largura da faixa de respostas dos dados (NRMSE). Esses erros são calculados das seguintes formas:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (Y(\omega_i) - \hat{Y}(\omega_i))^2}; \quad (4)$$

$$NRMSE = \frac{RMSE}{|\max(Y(\omega)) - \min(Y(\omega))|}, \quad (5)$$

em que  $n$  é a quantidade de pontos de frequência,  $Y(\omega_i)$  é a resposta em frequência dos dados originais e  $\hat{Y}(\omega_i)$  é a resposta em frequência do modelo.

Feita essa escolha, serão usados dados externos no domínio da frequência e algoritmos de código aberto assim como uma implementação própria para o desenvolvimento de modelos matemáticos. Tais modelos serão analisados quantitativamente via simulações computacionais e ensaios experimentais de transitórios.

## 2 DESENVOLVIMENTO

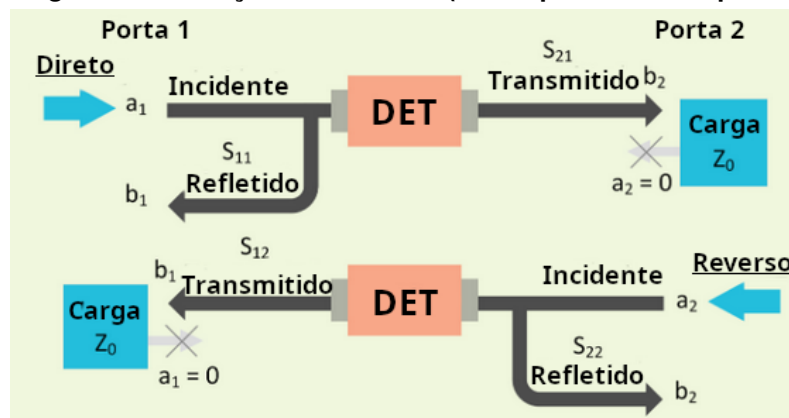
Nesse capítulo será feita uma introdução aos processos de aquisição de dados e ao algoritmo *vector fitting*, que será usado para a identificação de sistemas no domínio da frequência. Após isso, será apresentada a propriedade da passividade, as etapas de verificação e imposição de passividade e os métodos que serão usados nesse trabalho. Finalmente, será apresentado um pacote de otimização convexa no ambiente do MATLAB (CVX).

### 2.1 Aquisição de dados

Para a identificação de modelos no domínio da frequência, é necessário obter a resposta em frequência do sistema em uma banda de frequências. Para isso, usa-se um analisador de rede vetorial (VNA).

O VNA é conectado ao dispositivo em teste (DET) por meio de duas portas que servem como entradas e saídas do sistema. A partir dessas portas, o VNA gera sinais e mede suas reflexões e transmissões pelo equipamento. Como ambas as portas atuam como transmissores e receptores, são obtidas quatro medições, como mostra a Figura 2.

**Figura 2 – Diagrama de medições de um VNA (DET representa o dispositivo em teste)**



Fonte: Adaptado de Tektronix (2017).

Por conta da dificuldade de medir tensões e correntes em altas frequências, usam-se os parâmetros de espalhamento para caracterizar os equipamentos. Para um sistema de duas portas e considerando que os sinais são totalmente absorvidos pela carga de referência  $Z_0$ , eles são definidos pelas Equações (6), (7), (8) e (9) (Tektronix, 2017).

$$S_{11} = \frac{b_1}{a_1}, \quad (6)$$

$$S_{21} = \frac{b_2}{a_1}, \quad (7)$$

$$S_{12} = \frac{b_1}{a_2}, \quad (8)$$



$$S_{22} = \frac{b_2}{a_2}. \quad (9)$$

Em que  $S_{ij}$  é o espalhamento do sinal gerado pela porta  $j$  observado na porta  $i$ ,  $b_i$  é o sinal recebido na porta  $i$  e  $a_j$  é o sinal transmitido pela porta  $j$ . Vale constar que todos esse valores são fasores, ou seja, contém magnitudes e fases. A partir da matriz de espalhamento  $S$  e do valor da impedância usada na medição  $Z_0$ , pode-se fazer a conversão para a matriz de admitância  $Y$ .

$$Y = \left( \sqrt{Z_0} \right)^{-1} (I - S) (I + S)^{-1} \left( \sqrt{Z_0} \right). \quad (10)$$

Em que  $\sqrt{Z_0}$  é uma matriz diagonal cujos valores da diagonal são iguais à raiz quadrada da impedância de referência  $Z_0$ , comumente  $50 \Omega$ . De maneira análoga, também é possível obter a matriz de impedância  $Z$  do sistema, visto que  $Z = Y^{-1}$ . As matrizes de impedância e admitância trazem interpretações elétricas mais usuais, visto que elas relacionam as tensões e correntes do sistema.

$$I = YV, \quad (11)$$

$$V = ZI. \quad (12)$$

Em que  $I$  é o vetor de correntes e  $V$  é o vetor de tensões. Nesse trabalho, serão usados dados coletados por um VNA na forma de matrizes de admitância.

## 2.2 Vector Fitting

O *Vector Fitting* é um algoritmo de identificação de sistemas caixa preta no domínio da frequência e é o mais consolidado nessa área. Introduzido em Gustavsen e Semlyen (1998) e aprimorado em Gustavsen e Semlyen (1999), o *vector fitting* parte de um conjunto de dados coletados  $f(s)|_{s=j\omega_k}$ , sendo  $s$  a variável do domínio das frequências complexas e  $\omega$  a frequência. A medição é feita dentro de um conjunto de frequências  $\{s_k\}$  e assume-se que eles podem ser representados por uma função de transferência na forma da Equação (13).

$$f(s) \approx \sum_{n=1}^N \frac{c_n}{s - a_n} + sh + d. \quad (13)$$

A Equação (13) é formada por:

- $N$ : número de polos assumido a partir de uma análise inicial dos dados;
- $\{a_n, c_n, h, d\}$ : parâmetros do modelo;

A identificação dos polos  $a_n$  na Equação (13) se trata de um problema não linear, visto que os mesmos se encontram no denominador da equação. Assim, é útil usar um artifício para aproximá-la a uma função linear. Para isso, monta-se a equação  $\sigma(s)$ , que se relaciona com  $f(s)$  das seguintes maneiras:

$$\sigma(s) \approx \sum_{n=1}^N \frac{\tilde{c}_n}{s - \tilde{a}_n} + 1, \quad (14)$$

$$\sigma(s)f(s) \approx \sum_{n=1}^N \frac{c_n}{s - \tilde{a}_n} + sh + d = \sigma f(s). \quad (15)$$

Na Equação (14), é definido um conjunto de polos  $\tilde{a}_n$  que servirá como aproximação inicial do modelo. Visto isso, multiplicando  $f(s)$  na Equação (14), obtém-se:

$$\sum_{n=1}^N \frac{c_n}{s - \tilde{a}_n} + sh + d \approx \left( \sum_{n=1}^N \frac{\tilde{c}_n}{s - \tilde{a}_n} + 1 \right) f(s). \quad (16)$$

Na Equação (16), as variáveis desconhecidas são  $c_n$ ,  $h$ ,  $d$  e  $\tilde{c}_n$ , logo, trata-se de um problema linear. Para resolvê-lo, é feita a montagem matricial do problema:

$$\underbrace{\begin{bmatrix} \frac{1}{s_1 - \tilde{a}_1} & \dots & \frac{1}{s_1 - \tilde{a}_n} & s_1 & 1 & -\frac{f_1}{s_1 - \tilde{a}_1} & \dots & -\frac{f_1}{s_1 - \tilde{a}_n} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{s_k - \tilde{a}_1} & \dots & \frac{1}{s_k - \tilde{a}_n} & s_k & 1 & -\frac{f_k}{s_k - \tilde{a}_1} & \dots & -\frac{f_k}{s_k - \tilde{a}_n} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} c_1 \\ \vdots \\ c_n \\ h \\ d \\ \tilde{c}_1 \\ \vdots \\ \tilde{c}_n \end{bmatrix}}_{\mathbf{x}} \approx \underbrace{\begin{bmatrix} f_1 \\ \vdots \\ f_k \end{bmatrix}}_{\mathbf{y}}. \quad (17)$$

Tendo um problema do tipo  $\mathbf{A}\mathbf{x} \approx \mathbf{y}$ , usa-se o método dos mínimos quadrados para se obter o vetor de parâmetros do modelo  $\mathbf{x}$  que gera o resultado mais próximo possível de  $\mathbf{y}$ . Tendo em vista que  $\mathbf{A}$  não é uma matriz quadrada e, conseqüentemente, não é invertível, usa-se o operador pseudo-inversa ( $\dagger$ ) para se obter uma inversa à esquerda. Como  $\mathbf{A}$  possui colunas linearmente independentes, sua pseudo-inversa é calculada pela Equação (18) (Greville, 1959).

$$\mathbf{A}^\dagger = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \quad (18)$$

Assim,  $x \approx A^\dagger y$  traz os valores que melhor aproximam as variáveis desconhecidas da Equação (16) para o conjunto de dados  $y$ .

Com tais valores, observa-se que:

$$f(s) \approx \frac{\sigma f(s)}{\sigma(s)}. \quad (19)$$

Escrevendo as Equações (14) e (15) fatoradas em zeros e polos:

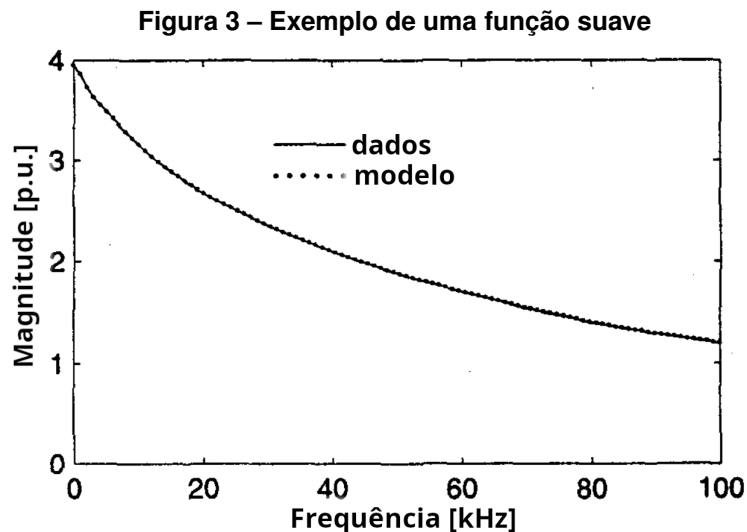
$$f(s) \approx \frac{\prod_{n=1}^{N+1} (s - z_n) \prod_{n=1}^N (s - \tilde{a}_n)}{\prod_{n=1}^N (s - \tilde{a}_n) \prod_{n=1}^N (s - \tilde{z}_n)} = \frac{\prod_{n=1}^{N+1} (s - z_n)}{\prod_{n=1}^N (s - \tilde{z}_n)}. \quad (20)$$

Nota-se, então, que os zeros de  $\sigma(s)$  são aproximações dos polos de  $f(s)$ . Com isso, pode-se fechar um laço iterativo usando o conjunto  $\tilde{z}_n$  como novos polos na Equação (16) e resolver o sistema linear novamente. Cada iteração leva o conjunto  $\tilde{z}_n$  mais próximo ao conjunto de polos verdadeiro  $a_n$ .

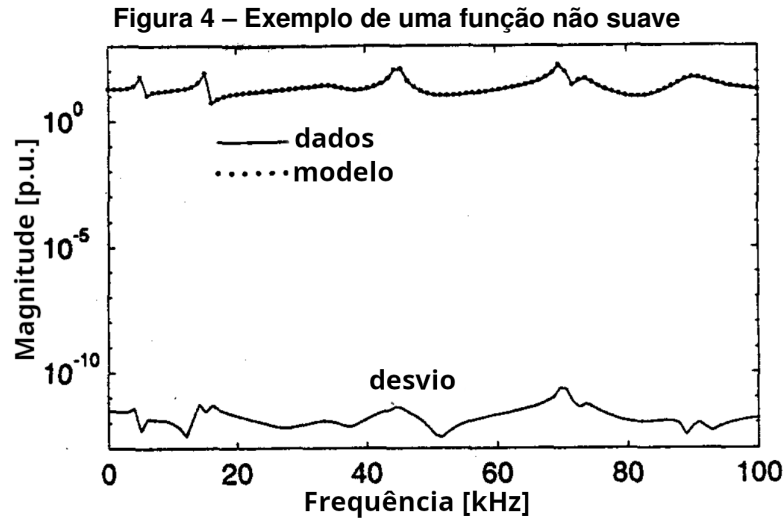
Quando se atinge a precisão desejada, basta usar o conjunto aproximado no lugar de  $a_n$  na Equação (13) e resolver o sistema linear uma última vez. Dessa forma, faz-se:

$$f(s) \approx \sum_{n=1}^N \frac{c_n}{s - \tilde{z}_n} + d + sh. \quad (21)$$

Para otimizar a convergência do *vector fitting*, é preciso fazer uma boa escolha do conjunto inicial de polos  $\tilde{a}_n$ . Isso é feito a partir da análise inicial dos dados de  $f(s)$ . Caso a função seja suave, isto é, sem picos de ressonância, como na Figura 3, polos reais espaçados linearmente no domínio da frequência são ideais. Caso haja picos de ressonância, como na Figura 4, é mais eficaz usar polos complexos levemente atenuados, com parte real cerca de 100 vezes menor que a parte imaginária, e espaçados em escala logarítmica nas frequências (Gustavsen; Semlyen, 1999).



Fonte: Adaptado de Gustavsen e Semlyen (1999).



Fonte: Adaptado de Gustavsen e Semlyen (1999).

No caso de funções não suaves, os polos iniciais indicados são pares de complexos conjugados  $p = \alpha \pm j\beta$ , com  $\beta$  distribuído linearmente dentro da banda de frequência de interesse e  $\alpha = \frac{\beta}{100}$ . Em funções suaves, os polos indicados são reais e espaçados linear ou logaritmicamente em função da frequência.

Além disso, uma análise inicial do sistema pode ajudar a obter um modelo com estabilidade e causalidade mais coerentes com o sistema analisado. Por exemplo, caso o sistema seja causal, zerar o termo  $h$  nas relações matemáticas garante que o modelo final também seja causal.

A Equação (21) aplica-se para o caso SISO. Porém, é possível fazer ajustes para satisfazer casos MIMO (Gustavsen, 2002). Para tal, busca-se aproximar não apenas uma função  $f(s)$ , mas uma matriz de funções  $\mathbf{F}(s)$ , em que  $Q$  representa o número de saídas e  $P$  representa o número de entradas.

$$\mathbf{F}(s) = \begin{bmatrix} f_{1,1}(s) & f_{1,2}(s) & \dots & f_{1,P}(s) \\ f_{2,1}(s) & f_{2,2}(s) & \dots & f_{2,P}(s) \\ \vdots & \vdots & \ddots & \vdots \\ f_{Q,1}(s) & f_{Q,2}(s) & \dots & f_{Q,P}(s) \end{bmatrix} = \sum_{n=1}^N \frac{\mathbf{C}_n}{s - \tilde{a}_n} + s\mathbf{H} + \mathbf{D}, \quad (22)$$

$$f_{ij}(s) = \sum_{n=1}^N \frac{c_{ijn}}{s - \tilde{a}_n} + sh_{ij} + d_{ij}. \quad (23)$$

Nas Equações (22) e (23),  $i$  refere-se ao índice da linha referida e  $j$  refere-se ao índice da coluna referida. Percebe-se que o único termo que independe de  $i$  e  $j$  é  $\tilde{a}_n$ . Isso significa que a aproximação resultante do *vector fitting* assume polos comuns para todas as entradas e saídas. Assim, é possível empilhar todos os termos de  $\mathbf{F}(s)$  em uma única coluna e realizar o mesmo procedimento da Equação (17) para identificar os componentes  $f_{ij}(s)$  individualmente. Outra abordagem para resolver o caso MIMO é realizar a identificação do sistema linha a linha,  $P$  sistemas MISO, ou coluna a coluna,  $Q$  sistemas SIMO, e remontar a realização MIMO.

Nesse trabalho, será utilizado o algoritmo em MATLAB disponibilizado publicamente pela Fundação para Pesquisa Industrial e Técnica (SINTEF) (Gustavsen, 2008). Nele, é feita uma aplicação do *vector fitting* otimizada para alocação mais rápida dos polos, baseada em Gustavsen (2006) e Deschrijver *et al.* (2008).

## 2.3 Passividade

Assegurar a passividade de um modelo é importante para manter suas propriedades condizentes com a realidade. Assim, as simulações e previsões baseadas nele são mais confiáveis. Além disso, a passividade apresenta a propriedade de fechamento algébrico. Isso significa que qualquer conexão de sistemas passivos resulta em um sistema passivo. Logo, é possível fazer integrações complexas envolvendo diferentes modelos passivos sem que essa propriedade seja perdida na topologia final. Ademais, a passividade é uma condição suficiente para a causalidade e estabilidade do modelo, enquanto a recíproca não é verdadeira. Logo, assegurar a passividade de um modelo também assegura sua causalidade e estabilidade.

Tendo em vista essas motivações, garantir a passividade de um modelo identificado é benéfico. Portanto, durante a aplicação do *vector fitting* é possível assegurar que o modelo resultante do ajuste seja causal, ao zerar a variável  $h$  da Equação (15), e estável, ao manter as partes reais dos polos alocados negativas. Porém, a passividade não é assegurada durante o processo de ajuste. Assim, o modelo obtido a partir do *vector fitting* pode apresentar passividade acidentalmente, mas não há garantia disso, logo, cabe ao usuário fazer a verificação e imposição de passividade em uma etapa de pós-processamento do modelo.

### 2.3.1 Verificação de Passividade

A passividade é uma propriedade definida a partir da energia de um sistema, mais especificamente, o sistema analisado pode somente dissipar energia das suas entradas e, logo, não pode gerar saídas com maior energia. O método de verificação de passividade de um sistema depende da maneira como o sistema é representado. Nesse trabalho, serão discutidos dois métodos: um no domínio da frequência, na forma de uma matriz de transferência e outro, independente da frequência, no espaço de estados.

#### 2.3.1.1 Métodos Dependentes da Frequência

Uma maneira simples de averiguar a passividade de um sistema se dá ao avaliar se a parte real da sua matriz de admitância é positivo definida. Isso ocorre quando todos os seus autovalores são positivos para todos os pontos do domínio de Laplace. Esse método de verificação de passividade é chamado de *positive realness*. Para o caso de uma matriz de admitância  $\mathbf{Y}(s)$ , isto é:

$$\lambda(\text{Re}\{Y(s)\}) = \lambda(G(s)) > 0, \quad \forall s \in \mathbb{C}. \quad (24)$$

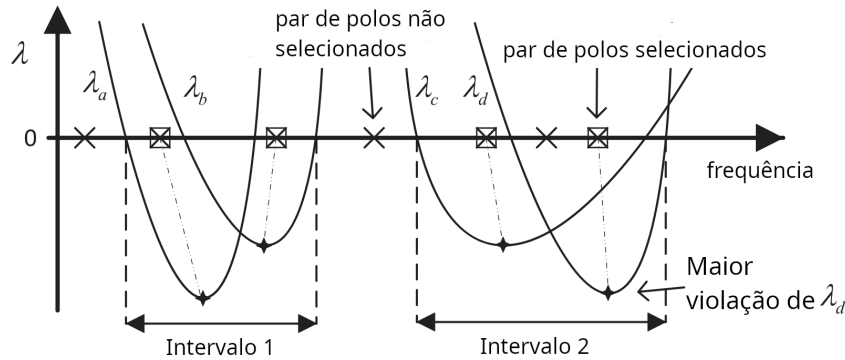
Na Equação (24)  $G(s)$  é a parte real da matriz de admitância e a função  $\lambda$  calcula seus autovalores. Em situações práticas, a avaliação de todos os pontos do domínio de Laplace é impossível, tendo em vista que o processamento é sempre discreto. Em conjuntos de dados de respostas em frequência finitos, porém, é possível aplicar o método de *positive realness*, visto que se precisa avaliar a matriz admissão apenas nas frequências medidas.

Dessa maneira, o método de *positive realness* pode ser usado para assegurar a passividade em medições de respostas em frequência. Com isso, é possível fazer a validação das medições antes mesmo de fazer a identificação de um modelo. Esse é um passo importante, pois realizar a imposição de passividade em modelos gerados a partir de dados não passivos não traz os resultados esperados e pode afetar a precisão do modelo final.

### 2.3.1.2 Métodos Independentes da Frequência

Como visto anteriormente, a Equação (24) pode ser usada quando se tem um conjunto discreto de pontos no domínio de Laplace. Porém, é desejável assegurar a passividade do modelo identificado em todo o domínio de Laplace, para garantir seu bom funcionamento em pontos não contemplados pelas medições iniciais. A Figura 5 representa a possibilidade da passividade ser violada entre pontos de frequência não observados. Isso é demonstrado pelos intervalos em que os autovalores calculados são negativos.

**Figura 5 – Exemplo de violação de passividade entre pontos de frequência medidos**



**Fonte: Adaptado de Morales et al. (2018).**

Assim, pode-se usar do lema positivo real (PRL) (Ferrante, 2005) para fazer a avaliação de uma realização em espaço de estados do modelo identificado. Ele é dado da seguinte forma:

$$\Psi = \begin{bmatrix} -A^T P - P A & -P B + C^T \\ -B^T P + C & D + D^T \end{bmatrix} = \begin{bmatrix} -A^T P - P A & -P B + C^T \\ (-P B + C^T)^T & D + D^T \end{bmatrix} \succ 0. \quad (25)$$

Na Equação (25), o símbolo  $\succ$  indica que todos os autovalores da matriz  $\Psi$  devem ser positivos, ou seja,  $\Psi$  deve ser positivo definida. As matrizes  $A$ ,  $B$ ,  $C$  e  $D$  são as matrizes da realização em espaço de estados e a matriz  $P$  é uma matriz simétrica positivo definida qualquer.

Vale ressaltar que, ao se encontrar uma matriz  $P$  que satisfaça a Equação (25), a passividade é garantida para todas as frequências, assim, o modelo será passivo até em pontos não contemplados pelos dados usados para identificação.

### 2.3.2 Imposição de Passividade

Após identificar um modelo a partir do *vector fitting* e verificar que a passividade é violada, pode-se implementar um pós-processamento do modelo, denominado imposição de passividade. Nesse trabalho, serão analisadas duas maneiras de fazer essa imposição. A primeira aplica pequenas perturbações determinísticas nas matrizes da Equação (22) de forma que elas passem a seguir as restrições do *positive realness*, vide Equação (24). O instituto SINTEF disponibilizou uma implementação desse método de imposição de passividade em MATLAB (Gustavsen, 2009). Essa implementação será usada no Capítulo 3 para a avaliação desses métodos.

A segunda maneira de imposição de passividade que será analisada é baseada no método das frações positivas. Ela baseia-se em fazer uma identificação dos resíduos do *vector fitting* de maneira que todas as frações da Equação (22) sejam positivas. Como será analisado posteriormente, esse método é menos custoso computacionalmente quando comparado ao método baseado em perturbações, mas pode levar a soluções sub ótimas.

#### 2.3.2.1 Método Baseado na *Positive Realness*

Dado um modelo na forma de funções racionais, de acordo com a Equação (22), que não obedece a condição de *positive realness*, da Equação (24), deseja-se encontrar uma matriz de perturbações  $\Delta Y$  tal que (Gustavsen, 2007):

$$\Delta Y = \sum_{n=1}^N \frac{\Delta C_n}{s - \tilde{a}_n} + \Delta D + s\Delta E \approx 0; \quad (26)$$

$$\lambda \left( \text{Re} \left\{ \sum_{n=1}^N \frac{\Delta C_n}{s - \tilde{a}_n} + \Delta D \right\} \right) > 0; \quad (27)$$

$$\lambda(D + \Delta D) > 0; \quad (28)$$

$$\lambda(E + \Delta E) > 0. \quad (29)$$

Nessas equações,  $D$ ,  $E$  e  $\tilde{a}_n$  são parâmetros do modelo e  $\Delta C_n$ ,  $\Delta D$  e  $\Delta E$  são as perturbações calculadas. Essas condições visam garantir que o modelo resultante após a

aplicação da perturbação  $\Delta Y$  em  $Y$  será passivo, enquanto se minimiza as alterações no modelo original. Tal problema é resolvido da seguinte forma:

$$\min_{\Delta x} \frac{1}{2} (\Delta x^\top A_{sys}^\top A_{sys} \Delta x), \quad \text{sujeito a} \quad (30)$$

$$B_{sys} \Delta x < c.$$

Nessa equação,  $\Delta x$  contem os valores das perturbações necessárias, ou seja, os valores de  $\Delta C_n$ ,  $\Delta D$  e  $\Delta E$ . Esse problema de otimização é resolvido via programação quadrática.

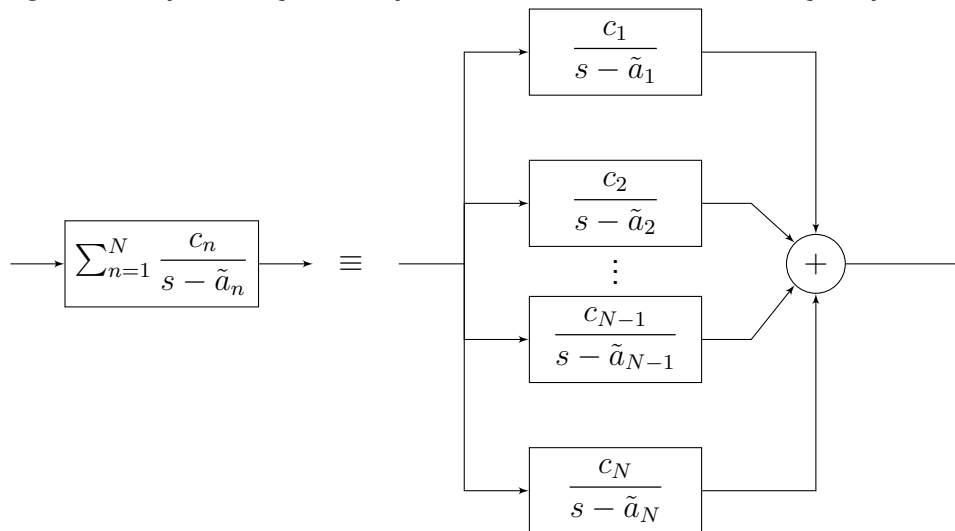
Esse método se baseia na condição de *positive realness*, que é uma condição suficiente e necessária. Isso significa que a solução do problema de otimização é, de fato, ótimo. Porém, para isso, é necessário otimizar muitos parâmetros em  $\Delta x$ , que escala com a ordem do sistema e sua quantidade de entradas e saídas, o que pode provar-se custoso computacionalmente.

Uma implementação desse método está disponível em código aberto na rotina de MATLAB *RPdriver.m* (Gustavsen, 2009). Tendo em vista que esse pacote é bastante validado e robusto, essa rotina será usada em conjunto com a rotina *VFdriver.m*, disponível no mesmo pacote, para se obter modelos que representam o estado da arte da imposição de passividade.

### 2.3.2.2 Método Baseado em Frações Positivas

Outra maneira de garantir que a função da Equação (22) seja passiva dá-se ao garantir que todos os seus termos sejam positivos. Então, pode-se fazer a expansão do somatório e impor a passividade em cada um dos termos individualmente. Isso é análogo a observar o sistema como um conjunto de subsistemas ligados em paralelo, como representado na Figura 6.

**Figura 6 – Representação da expansão usada no método das frações positivas**



**Fonte: Autoria própria.**

Assim, devido a propriedade de fechamento algébrico, pode-se impor a passividade em todas as  $N$  frações resultantes e realizar o somatório desses termos passivos para obter-se um



modelo passivo. A utilidade dessa estratégia torna-se evidente ao observar-se a simplicidade de realizar a imposição de passividade nos termos separadamente. Separando o conjunto de polos reais do conjunto de polos complexos conjugados, obtém-se (Tommasi *et al.*, 2008):

$$Y(s) = a_0 + \sum_{n=1}^{N_r} \frac{a_n}{s - p_n} + \sum_{m=1}^{N_c} \frac{sb_m + c_m}{s^2 + sd_m + e_m}. \quad (31)$$

Na Equação (31),  $N_r$  é o número de polos reais no modelo e  $N_c$  é o número de pares de polos complexos conjugados no modelo. Os termos individuais serão positivos caso  $a_n \geq 0$  para  $n \in \{1, 2, \dots, N_r\}$  e  $c_n \geq 0$  e  $b_n d_n \geq c_n$  para  $n \in \{1, 2, \dots, N_c\}$ . Assim, o método das frações positivas procura resolver o problema de otimização:

$$\begin{aligned} \min_{\{a_n, b_n, c_n\}} \sum_{k=1}^K (Y(s_k) - \tilde{Y}(s_k))^2, \quad \text{sujeito a} \\ a_n \geq 0 \quad \forall n \in \{1, 2, \dots, N_r\}, \\ c_n \geq 0 \quad \forall n \in \{1, 2, \dots, N_c\}, \\ b_n d_n \geq c_n \quad \forall n \in \{1, 2, \dots, N_c\}, \end{aligned} \quad (32)$$

em que  $Y(s_k)$  é o conjunto de dados original e  $\tilde{Y}(s_k)$  é o conjunto de resultados do modelo passivo. Então, visa-se minimizar o erro quadrático do modelo, enquanto se mantém todas as frações positivas. De maneira perturbativa, o problema de otimização torna-se:

$$\begin{aligned} \min_{\{\Delta a_n, \Delta b_n, \Delta c_n\}} ||\Delta R_n||, \quad \text{sujeito a} \\ a_n + \Delta a_n \geq 0 \quad \forall n \in \{0, 1, \dots, N_r\}, \\ c_n + \Delta c_n \geq 0 \quad \forall n \in \{1, 2, \dots, N_c\}, \\ (b_n + \Delta b_n)d_n \geq c_n + \Delta c_n \quad \forall n \in \{1, 2, \dots, N_c\}. \end{aligned} \quad (33)$$

em que  $\Delta R_n$  agrupa as perturbações  $\Delta a_n$ ,  $\Delta b_n$  e  $\Delta c_n$  aplicadas em cada resíduo. Sendo assim, é possível assegurar a passividade sem a inclusão de mais variáveis no problema de otimização.

Por conta disso, o método das frações positivas facilita o processo de otimização, porém, vale constar que ter todas as frações positivas é uma condição apenas suficiente, não necessária (Tommasi; Deschrijver; Dhaene, 2008). Isso significa que o resultado obtido por esse método pode não ser um resultado ótimo quando comparado com resultados obtidos a partir de métodos que se baseiam em condições suficientes e necessárias, como o método de perturbações apresentado anteriormente.

A partir desses conceitos, será desenvolvida uma rotina em MATLAB para implementar o método das frações positivas e fazer a imposição de passividade. As respostas dos modelos

resultantes serão comparadas às respostas obtidas com o método anterior, para se validar o algoritmo desenvolvido.

## 2.4 Pacote de Otimização Convexa (CVX)

Para resolver os problemas de otimização necessários, será utilizado o pacote de otimização convexa (CVX), baseado em Research (2012) e Grant e Boyd (2008), que é uma ferramenta disponível no *software* MATLAB que permite a construção em uma linguagem de mais alto nível, como apresentada pela Equação (32). A partir do problema declarado em linguagem de alto nível, o CVX faz as devidas transformações e aplica os solucionadores adequados, facilitando a resolução.

Para tornar isso possível, o CVX requer que o usuário siga um conjunto de regras específicas que garantem que o problema proposto é convexo e, assim, possui uma solução única. Esse conjunto é chamado de conjunto de regras de programação convexa disciplinada (do inglês *disciplined convex programming ruleset*). As regras relevantes para o problema de imposição de passividade são:

- uma função convexa multiplicada por uma função afim ou constante é convexa;
- funções objetivo de minimização devem possuir argumentos convexos;
- restrições do tipo maior ou igual ( $\geq$ ) devem possuir uma função convexa à esquerda e côncava à esquerda;

Vale comentar que o CVX já interpreta construções do tipo  $\mathbf{A} \geq c$ , em que  $\mathbf{A}$  é uma matriz e  $c$  é uma constante qualquer, como  $\mathbf{A} \succeq c$ , ou seja, todos os autovalores de  $\mathbf{A}$  são maiores ou iguais a  $c$ . Isso facilita a verificação de matrizes semipositivo definidas, que é importante para a generalização do processo de imposição de passividade para os casos MIMO.

Por isso, o CVX tornou-se uma ferramenta conceituada na área de otimização, sendo usado em diversos artigos como Wong e Zhou (2023) e Usman e Ramdhani (2019). Para a implementação prática, pode-se seguir os passos do manual produzido por Grant e Boyd (2024), como o exemplo da Figura 7.

**Figura 7 – Exemplo de implementação de mínimos quadrados no CVX**

```
cvx_begin
    variable x(n)
    minimize( norm(A*x-b) )
    subject to
        l <= x <= u
cvx_end
```

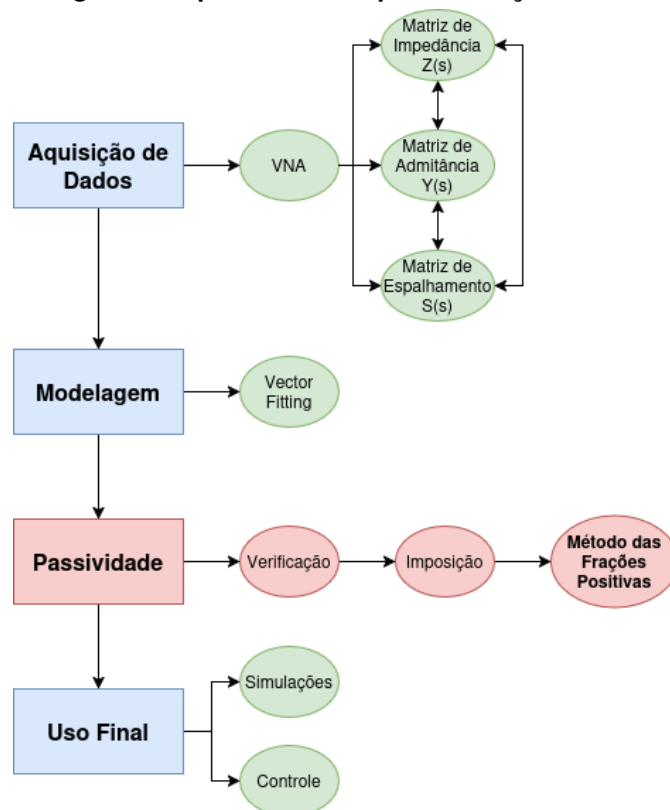
**Fonte: Grant e Boyd (2024).**

Esse exemplo demonstra uma aplicação de mínimos quadrados que encontra um vetor de parâmetros  $x$  que melhor aproxima a matriz  $A$  do vetor  $b$ , enquanto mantém os elementos de  $x$  entre os limites  $l$  e  $u$ , que podem ser constantes ou vetores de mesma dimensão de  $x$ . Assim, é possível perceber que um problema complexo pode ser resolvido de maneira trivial e com uma linguagem de alto nível com o CVX.

## 2.5 Resumo dos Procedimentos

A Figura 8 traz um resumo do procedimento de modelagem de sistemas passivos contemplado nesse trabalho. Primeiro, faz-se a aquisição de dados no domínio da frequência. Então faz-se o ajuste de um modelo inicial a esses dados com o *vector fitting*. Com o modelo, faz-se a verificação e a imposição de passividade, usando o método das frações positivas. Finalmente, obtém-se um modelo passivo que pode ser usado para aplicações de simulação e controle de sistemas.

**Figura 8 – Fluxograma do procedimento para obtenção de modelos passivos**



**Fonte: Autoria própria.**

Na Figura 8, em vermelho, é enfatizado o foco do presente trabalho, que é a implementação de técnicas de verificação e imposição de passividade em modelos no domínio da frequência, com maior atenção no método das frações positivas.

### 3 RESULTADOS E DISCUSSÕES

Nesse capítulo serão realizadas análises exploratórias de dados de resposta em frequência de um transformador, explicitando as violações de passividade nas medições. Posteriormente, também será feito ajuste de um modelo a essas medições a partir do *vector fitting*. Esse modelo terá sua passividade verificada e imposta usando a rotina disponibilizada pelo instituto SINTEF e a rotina baseada no método das frações positivas desenvolvida. Os modelos finais serão comparados com o objetivo de validar a implementação desenvolvida.

#### 3.1 Análise Exploratória da Passividade dos Dados

Para fazer a implementação do algoritmo de imposição de passividade baseado no método das frações positivas, foram usados dados de resposta em frequência de transformadores. Os conjuntos de dados escolhidos já apresentam violações de passividade intencionais para se observar seus efeitos.

Primeiramente, foi feita a verificação de passividade dos dados de acordo com o método de *positive realness*, descrito na Equação (24). Nessa etapa, espera-se encontrar autovalores negativos em frequências em que a passividade é violada. A Figura 9 revela o algoritmo usado para isso.

**Figura 9 – Código em MATLAB usado para a visualização das violações de passividade**

```
% Encontrando os autovalores da parte real da resposta em frequencia
realEigenValues = zeros(length(frequency), 1);
for i = 1:length(frequency)
    realEigenValues(i) = eig(real(response(i)));
end

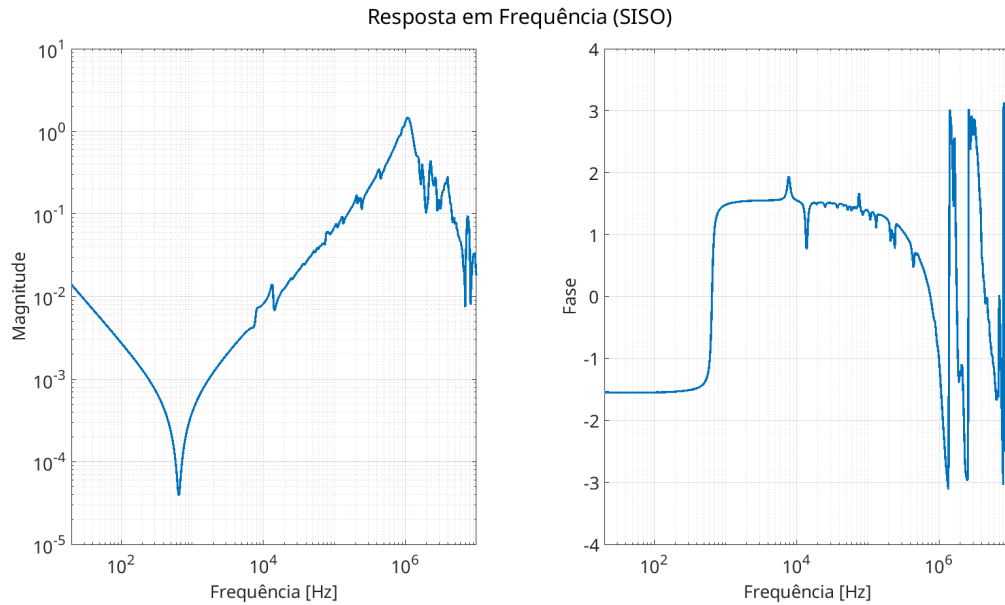
% Marcando os pontos em que há violação de passividade
for i = 1:length(realEigenValues)
    if(realEigenValues(i) < 0)
        plot(frequencyHz(i), realEigenValues(i), "r*");
    end
end
```

**Fonte: Autoria própria.**

Os dados com violações de passividade usados foram traçados no gráfico da Figura 10. Uma inspeção rápida dessa figura não levanta muitas suspeitas sobre o conjunto de dados, tendo em vista que parece adequada para um transformador. Assim, verificou-se os autovalores associados à parte real de cada medição.

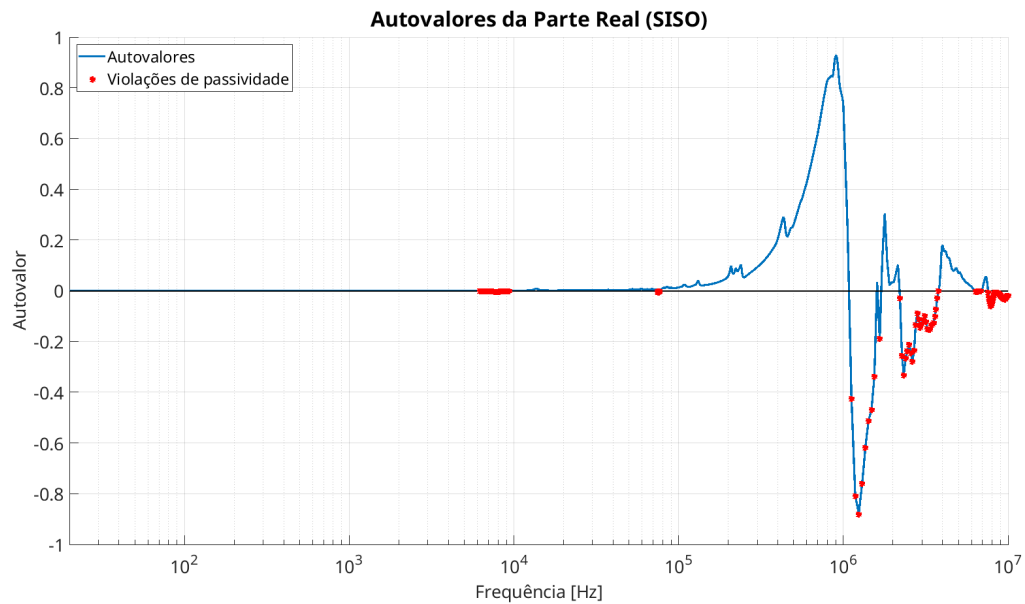
A Figura 11 traz os autovalores da parte real da resposta em frequência e marca, em vermelho, as frequências onde esses autovalores são negativos e, logo, há violações de passividade nos dados coletados. Grande parte das violações ocorre em frequências acima de 1 MHz, com a maior violação em 1,24 MHz, que apresenta um autovalor de -0,879. Tendo em vista que transformadores são elementos elétricos passivos, tais violações indicam que houve problemas durante a medição de sua resposta em frequência e esses dados apresentam incoerências.

**Figura 10 – Resposta em frequência do transformador**



Fonte: Autoria própria.

**Figura 11 – Autovalores da parte real dos dados de transformador**



Fonte: Autoria própria.

### 3.2 Aplicação das Rotinas da *Matrix Fitting Toolbox*

Para identificar o modelo inicial, foi usada a rotina *VFdriver.m*, presente na *Matrix Fitting Toolbox* (Gustavsen, 2009), que faz uma implementação otimizada do *Vector Fitting*. Com essa rotina, é possível especificar polos iniciais, número de iterações e pesos, que são parâmetros importantes para identificar um modelo que bem representa os dados.

Tendo em vista que os dados apresentam violações de passividade, espera-se que o modelo resultante da identificação também não seja passivo, então, a rotina *RPdriver.m*, da mesma *toolbox*, será aplicada para fazer a imposição de passividade a partir da perturbação de autovalores, de acordo com a Equação (30). O modelo resultante será visto como a referência de imposição de passividade e usado para validar o algoritmo baseado em frações positivas que será desenvolvido. Esses passos foram implementados de acordo com a Figura 12.

**Figura 12 – Código em MATLAB usado para identificar um modelo e impor passividade a partir de rotinas de código aberto consolidadas**

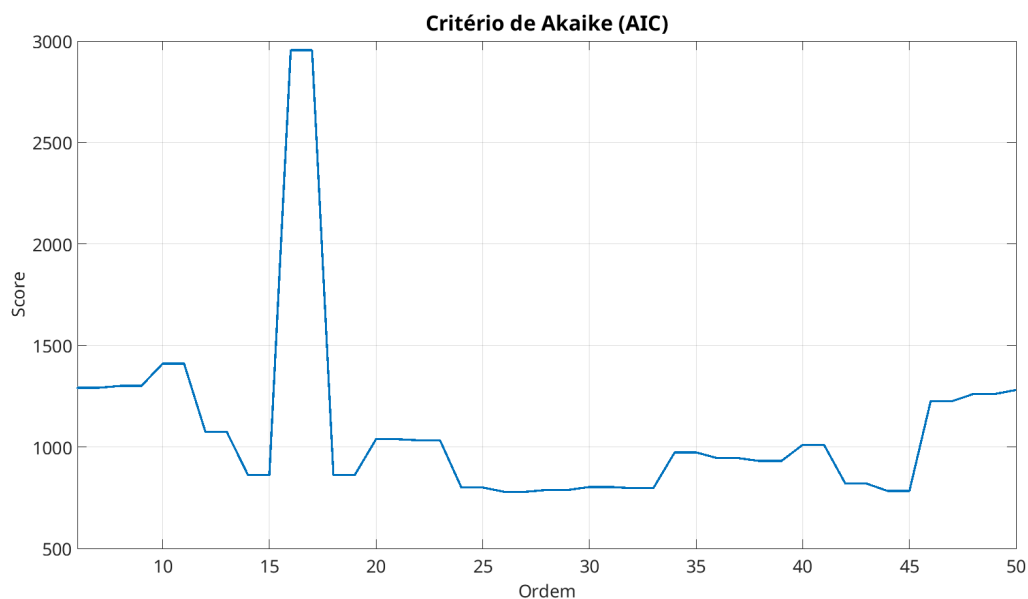
```
% Encontrando o modelo base
[vfDriverModel, ~, vfDriverResponse] = ...
    VFdriver(response, 1j*frequency, poles, opts);

% Impondo passividade
[rpDriverModel, rpDriverResponse, ~] = ...
    RPdriver(vfDriverModel, 1j*frequency, rpDriverOpts);
```

**Fonte: Autoria própria.**

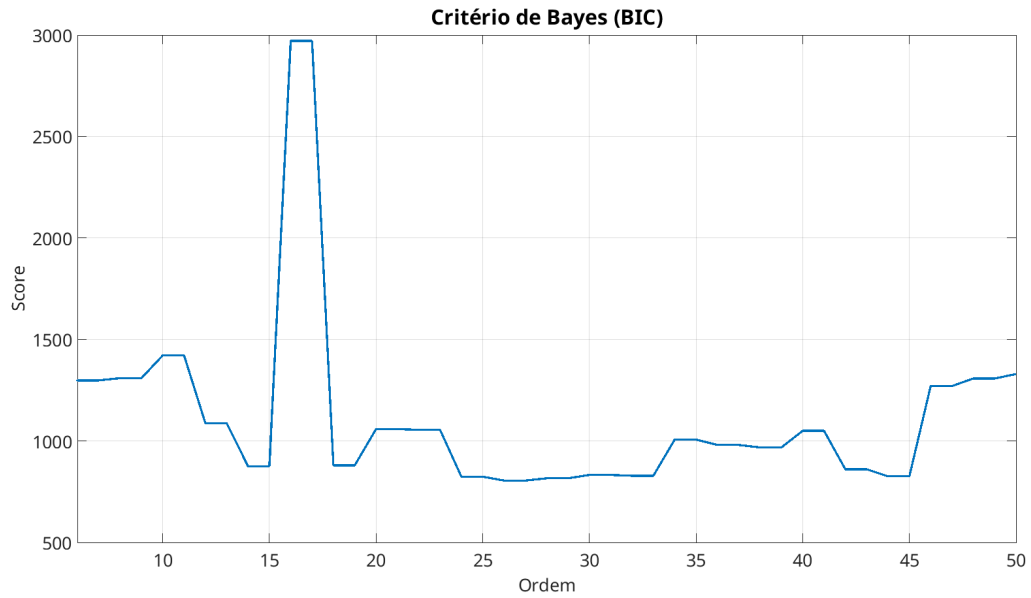
Foi necessário aplicar pesos inversamente proporcionais à magnitude das respostas para representar melhor a dinâmica do sistema nas frequências mais baixas, particularmente na largura de banda de 100 Hz a 10 kHz. Para encontrar a ordem que melhor representa o modelo, foram feitos testes com ordens variando de 5 a 100 e foram usados os critérios de escolha de Akaike (AIC), Bayes (BIC) e a forma corrigida do critério de Akaike (AICc) (Ljung, 1999). Os resultados são apresentados nas Figuras 13, 14 e 15.

**Figura 13 – Critério de escolha de Akaike**

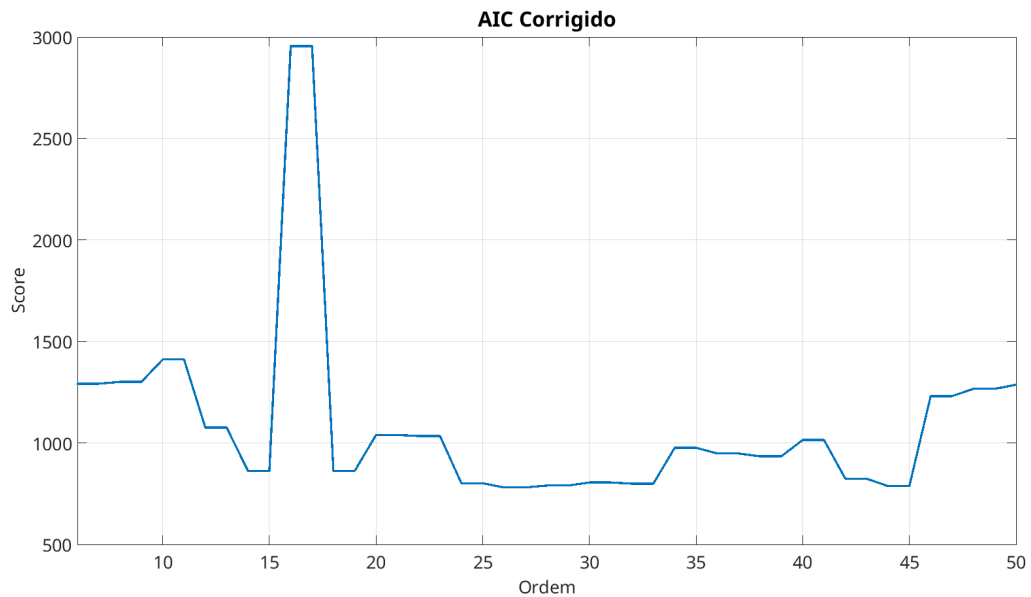


**Fonte: Autoria própria.**

De acordo com os resultados das Figuras 13, 14 e 15, foi escolhido usar um modelo com 26 polos, por apresentar menor pontuação e, logo, apresentar um bom compromisso entre precisão e complexidade.

**Figura 14 – Critério de escolha de Bayes**

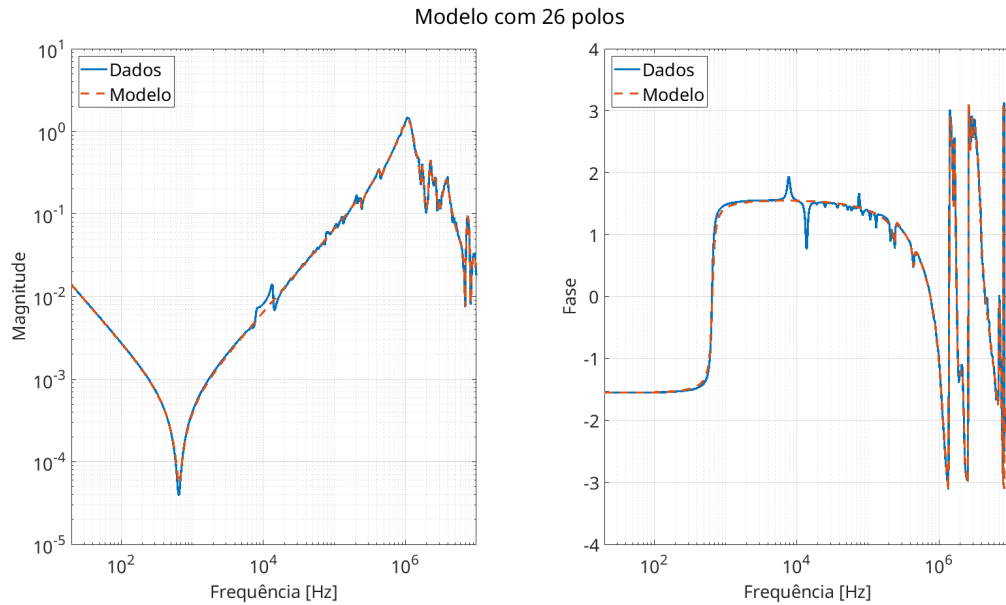
**Fonte: Autoria própria.**

**Figura 15 – AIC corrigido**

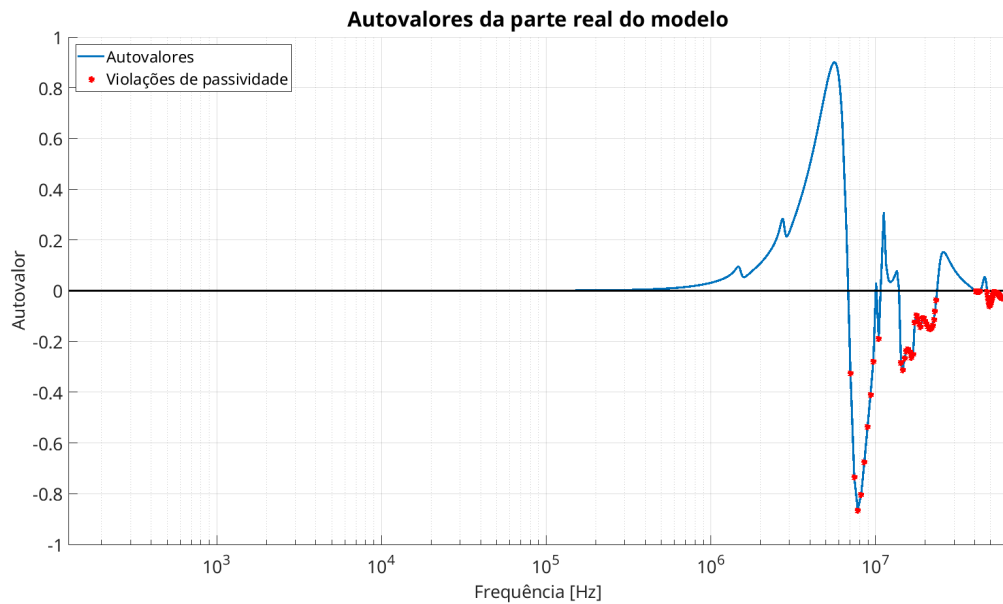
**Fonte: Autoria própria.**

Pode-se perceber que o modelo obtido na Figura 16 faz uma boa aproximação da dinâmica do sistema. Porém, tendo em vista que ele foi construído a partir de dados não passivos, espera-se que ele possua violações de passividade parecidas com aqueles apresentados na Figura 11.

A partir da análise da Figura 17, pode-se perceber que as violações de passividade presentes na Figura 11 nas bandas abaixo de 100 kHz foram suprimidas, enquanto as de maiores frequências foram mantidas, com a mudança sutil da maior violação de passividade sendo o

**Figura 16 – Modelo do sistema SISO com 26 polos**

Fonte: Autoria própria.

**Figura 17 – Autovalores da parte real do modelo identificado**

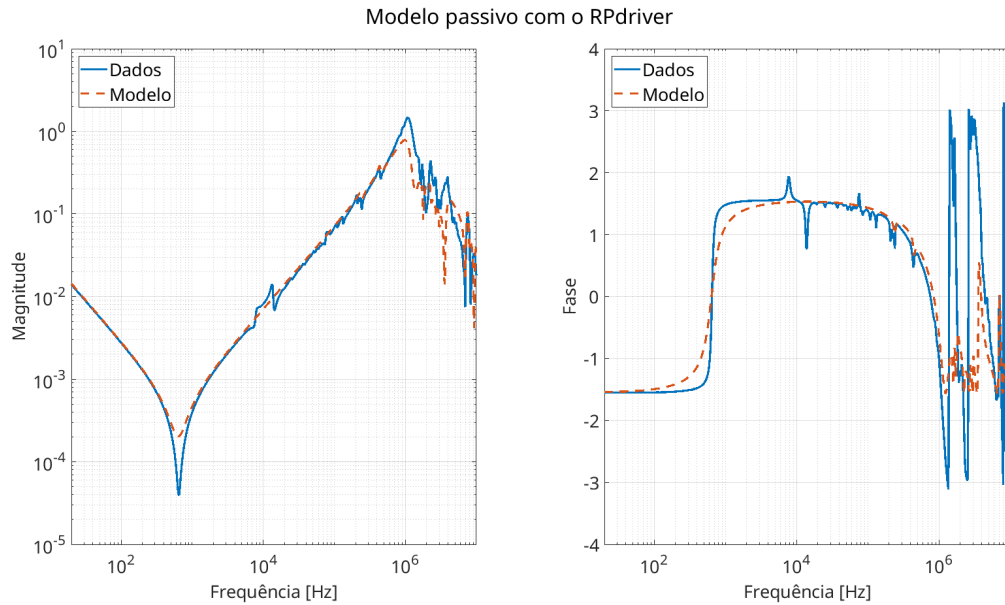
Fonte: Autoria própria.

autovalor de -0,853, ainda na frequência de 1,24 MHz. Isso confirma que o algoritmo do VF, apesar de garantir a estabilidade e causalidade do modelo, não garante sua passividade e até imita as violações apresentadas nos dados de origem.

Com isso, foi feita a aplicação da rotina RPdriver, disponibilizada pelo instituto SINTEF. Essa é uma ferramenta conceituada que faz a aplicação de perturbações nos autovalores não passivos de maneira iterativa até que a passividade seja alcançada.

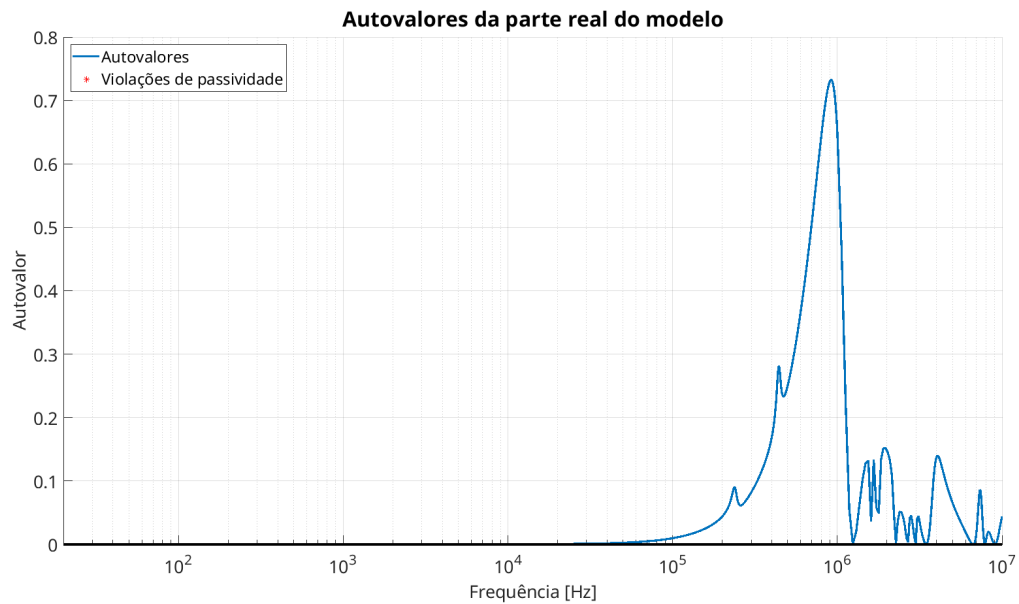


**Figura 18 – Modelo do sistema SISO passivado pelo RPdriver**



**Fonte: Autoria própria.**

**Figura 19 – Autovalores da parte real do modelo perturbado com o RPdriver**



**Fonte: Autoria própria.**

Observando a Figura 18, pode-se perceber que a resposta em frequência se manteve próxima dos dados originais.

Para a comparação quantitativa dos modelos, foram usados a raiz quadrada do erro quadrático médio (RMSE) e sua versão normalizada pela largura da faixa de respostas dos dados (NRMSE).

O RMSE do modelo não passivado foi de 0,0156 e seu NRMSE foi de 1,0642%. O RMSE do modelo passivo foi de 0,1597 e seu NRMSE foi de 10,8954%, cerca de dez vezes maior. Porém, como garantido pela Figura 19, o novo modelo é passivo em todas as frequências.

### 3.3 Método das Frações Positivas

Com um modelo passivado de referência, será feita a implementação baseada no método das frações positivas de acordo com a Equação (33). Será usada a abordagem perturbativa para que a comparação com o modelo passivo de referência seja mais coerente, tendo em vista que a rotina *RPdriver.m* também traz uma abordagem perturbativa. A Figura 20 demonstra a implementação dentro do ambiente do CVX.

**Figura 20 – Código em MATLAB usado para implementar o problema de otimização do método das frações positivas**

```
cvx_begin quiet
    variable delta0(size(a0));
    variable deltaR(size(residues)) complex;
    minimize(norm([delta0;deltaR]));
    subject to:
        a0 + delta0 >= 0;
        for i = 1:numberOfPoles
            -real(poles(i))*(real(residues(i)) + real(deltaR(i)))...
            - imag(poles(i))*(imag(residues(i)) + imag(deltaR(i))) >= 0;

            -real(poles(i))*(real(residues(i)) + real(deltaR(i)))...
            + imag(poles(i))*(imag(residues(i)) + imag(deltaR(i))) >= 0;
        end
cvx_end
```

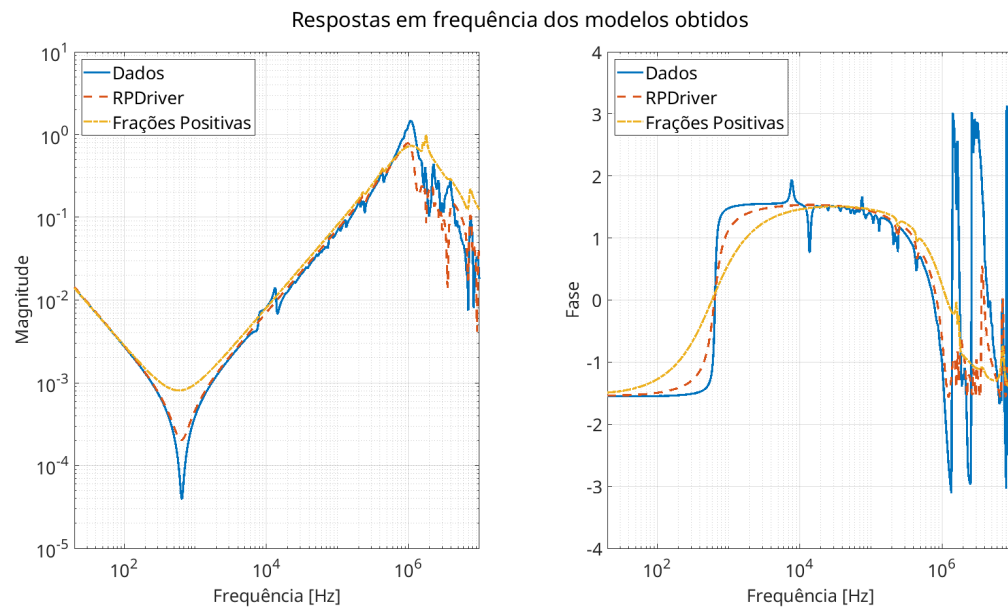
**Fonte: Autoria própria.**

A partir do uso do código presente na Figura 20, foi obtido um vetor de perturbações que, quando somado aos resíduos originais, tornam o modelo passivo. Os resultados foram:

Observando a Figura 21, pode-se perceber que a resposta em frequência do modelo das frações positivas ficou mais longe dos dados originais que o modelo passivado pelo *RPdriver*, apresentando um RMSE de 0,3156 e um NRMSE de 21,5357%, cerca de duas vezes maior que o modelo passivado pelo *RPdriver*. Isso era esperado, tendo em vista que o método das frações positivas não garante resultados ótimos, diferente do *RPdriver*. Dito isso, na Figura 22, pode-se conferir que o novo modelo é passivo em todas as frequências, logo, o resultado obtido satisfaz o proposto.

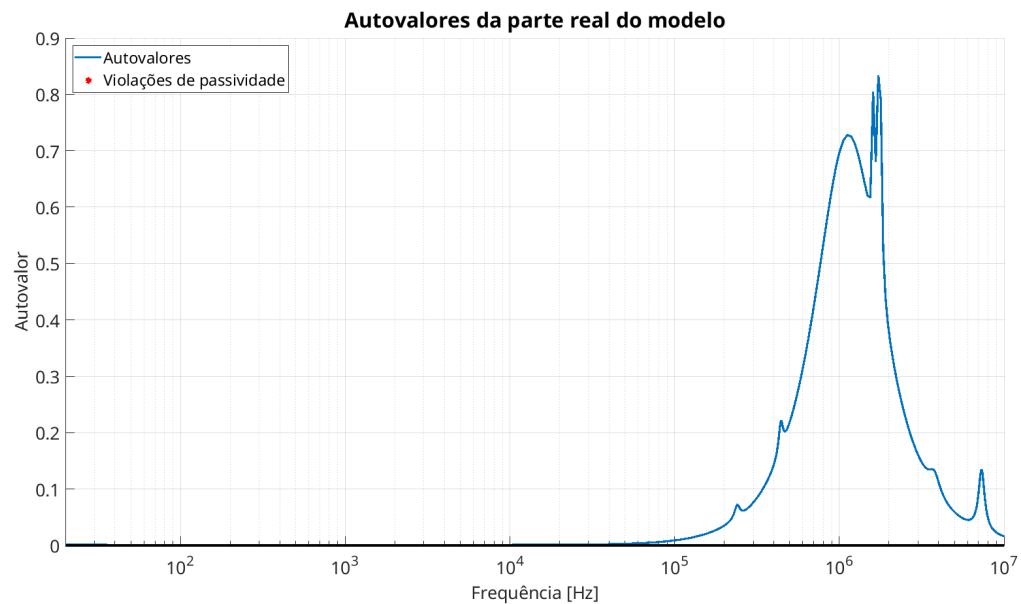
A Tabela 1 e a Figura 23 fazem a síntese dos resultados obtidos:

**Figura 21 – Modelo do sistema SISO passivado pelo método das frações positivas**



Fonte: Autoria própria.

**Figura 22 – Autovalores da parte real do modelo perturbado com o método das frações positivas**

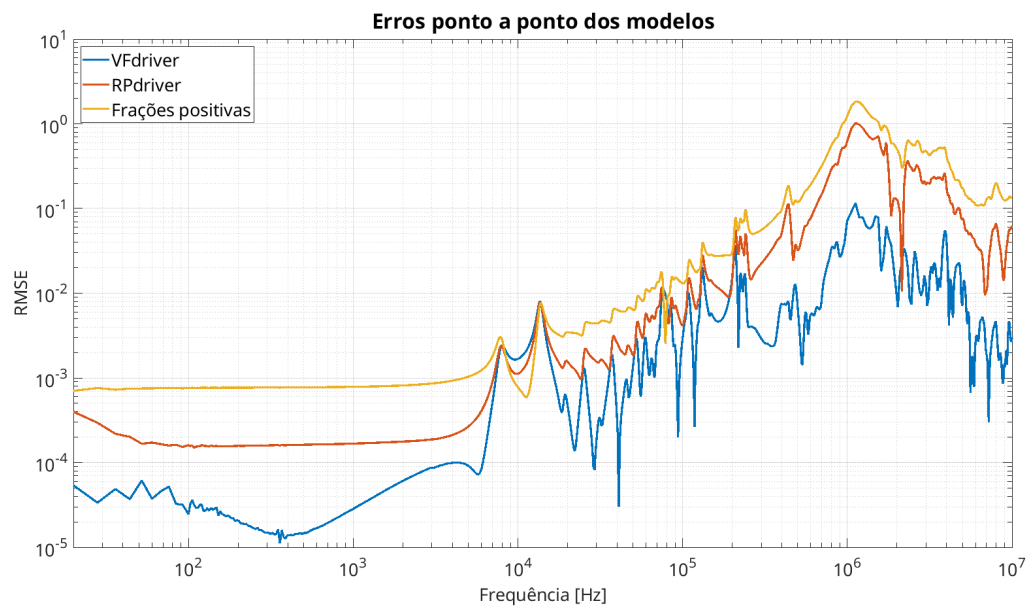


Fonte: Autoria própria.

**Tabela 1 – Síntese dos resultados dos modelos**

Método	RMSE	NRMSE (%)	AIC	BIC	AICc	Passivo
VFdriver	0,0156	1,0642	-1598,3	-1572,4	-1596,6	Não
RPdriver	0,1597	10,8954	240,3	266,8	242,0	Sim
Frações Positivas	0,3156	21,5357	778,9	804,7491	780,5693	Sim

Fonte: Autoria própria.

**Figura 23 – Erros ponto a ponto dos modelos**

**Fonte: Autoria própria.**

## 4 CONCLUSÃO

Neste trabalho de conclusão de curso foram aplicadas técnicas de identificação de sistemas no domínio da frequência, verificação de passividade e imposição de passividade, com o objetivo de desenvolver e avaliar uma implementação baseada no método das Frações Positivas. Foi possível analisar o comportamento de modelos matemáticos obtidos a partir de dados experimentais de resposta em frequência de um transformador, cuja natureza passiva é conhecida, mas cujos dados apresentavam violações de passividade.

A partir do uso da rotina *VFdriver.m* da *Matrix Fitting Toolbox*, obteve-se um modelo racional adequado para representar o sistema, preservando suas propriedades de estabilidade e causalidade, porém mantendo as violações de passividade presentes nos dados. Para restabelecer essa característica, foram comparados dois métodos perturbativos de imposição de passividade: o método baseado na *Positive Realness*, implementado por meio da rotina *RPdriver.m* da *Matrix Fitting Toolbox*, e o método das Frações Positivas, implementado neste trabalho em ambiente MATLAB com suporte ao pacote de otimização CVX.

Os resultados mostraram que ambos os métodos foram capazes de eliminar as violações de passividade, garantindo que os modelos finais fossem passivos em todo o espectro de frequências analisado. Entretanto, o método baseado na *Positive Realness* apresentou desempenho superior, tanto em termos de erro médio (RMSE e NRMSE) quanto em fidelidade às medições originais. O método das Frações Positivas resultou em maiores erros quando comparado ao método anterior, mas isso já era esperado por conta da natureza não ótima das frações positivas. Em termos de representabilidade dos dados originais, o método das frações positivas mostrou-se fidedigno e trouxe resultados próximos ao desejado.

A análise dos resultados também permitiu constatar que a imposição de passividade, embora essencial para a estabilidade e coerência física dos modelos, introduz inevitavelmente uma degradação na precisão da resposta em frequência. Essa degradação, no entanto, pode ser controlada pela escolha adequada da ordem do modelo e pela calibração dos parâmetros de ponderação aplicados no processo de identificação. Também foi possível observar que os pesos escolhidos e os métodos de normalização das matrizes de regressão usados são de grande importância para a obtenção de bons modelos, tanto na etapa de identificação quanto na imposição de passividade.

Com base nos experimentos e análises realizadas, conclui-se que a implementação do método das Frações Positivas desenvolvida é uma alternativa viável para imposição de passividade, embora ainda careça de refinamento para alcançar desempenho semelhante à rotina consolidada da *Matrix Fitting Toolbox*. Esse trabalho, contudo, contribui para o avanço das técnicas de modelagem passiva com base em algoritmos de otimização convexa e oferece um ponto de partida para o desenvolvimento de novos algoritmos baseados nas Frações Positivas. Tendo em vista que o trabalho contemplou apenas o caso SISO, sugere-se a expansão para o caso MIMO para trabalhos futuros.

## REFERÊNCIAS

- ACHKAR, O. *et al.* Tpkey: Using tpms signals for secure and usable intra-vehicle device authentication. *In: 2025 IEEE INTERNATIONAL CONFERENCE ON CYBER SECURITY AND RESILIENCE (CSR)*. 2025. **Anais [...]** [S.l.: s.n.], 2025. p. 586–591.
- AGUIRRE, L. A. **Introdução à Identificação de Sistemas – Técnicas Lineares e Não-Lineares Aplicadas a Sistemas Reais**. 3. ed. Belo Horizonte, MG, Brasil: Editora UFMG, 2007. ISBN 9788570415844.
- AZEVEDO, R. C.; ENSSLIN, L. **Metodologia da Pesquisa para Engenharias**. 1. ed. Belo Horizonte, MG, Brasil: PPGE/CEFET, 2020. ISBN 9786500102680.
- DESCHRIJVER, D. *et al.* Macromodeling of multiport systems using a fast implementation of the vector fitting method. **IEEE Microwave and Wireless Components Letters**, v. 18, n. 6, p. 383–385, 2008.
- ECK, C.; GARCKE, H.; KNABNER, P. **Mathematical modeling**. 3. ed. [S.l.]: Springer International Publishing Berlin, 2017. ISBN 978-3-662-54335-1.
- FERRANTE, A. Positive real lemma: necessary and sufficient conditions for the existence of solutions under virtually no assumptions. **IEEE Transactions on Automatic Control**, v. 50, n. 5, p. 720–724, 2005.
- GRANT, M.; BOYD, S. Graph implementations for nonsmooth convex programs. *In: BLONDEL, V.; BOYD, S.; KIMURA, H. (Ed.). Recent Advances in Learning and Control*. [S.l.]: Springer-Verlag Limited 2008 (Lecture Notes in Control and Information Sciences). p. 95–110. [http://stanford.edu/~boyd/graph\\_dcp.html](http://stanford.edu/~boyd/graph_dcp.html).
- GRANT, M. C.; BOYD, S. P. . **The CVX Users' Guide: Release 2.2**. [S.l.], 2024. Disponível em: <https://cvxr.com/cvx/doc/>. Acesso em: 11 jun. 2025.
- GREVILLE, T. N. E. The pseudoinverse of a rectangular or singular matrix and its application to the solution of systems of linear equations. **SIAM Review**, v. 1, n. 1, p. 38–43, 1959.
- GUSTAVSEN, B. Computer code for rational approximation of frequency dependent admittance matrices. **IEEE Transactions on Power Delivery**, v. 17, n. 4, p. 1093–1098, 2002.
- GUSTAVSEN, B. Fast passivity enforcement for pole-residue models by perturbation of residue matrix eigenvalues. **IEEE Transactions on Power Delivery**, v. 23, n. 4, p. 2278–2285, 2008.
- GUSTAVSEN, B. Fast passivity enforcement of rational macromodels by perturbation of residue matrix eigenvalues. *In: 2007 IEEE WORKSHOP ON SIGNAL PROPAGATION ON INTERCONNECTS*. 2007. **Anais [...]** [S.l.: s.n.], 2007. p. 71–74.
- GUSTAVSEN, B. Improving the pole relocating properties of vector fitting. **IEEE Transactions on Power Delivery**, v. 21, n. 3, p. 1587–1592, 2006.
- GUSTAVSEN, B. **MATRIX FITTING TOOLBOX**. Trondheim, Noruega, 2009. Disponível em: <https://www.sintef.no/en/software/vector-fitting/downloads/matrix-fitting-toolbox/#menu>.
- GUSTAVSEN, B. **VFIT3**. Trondheim, Noruega, 2008. Disponível em: <https://www.sintef.no/en/software/vector-fitting/downloads/vfit3/>.
- GUSTAVSEN, B.; SEMLYEN, A. Rational approximation of frequency domain responses by vector fitting. **IEEE Transactions on Power Delivery**, v. 14, n. 3, p. 1052–1061, 1999.

GUSTAVSEN, B.; SEMLYEN, A. Simulation of transmission line transients using vector fitting and modal decomposition. **IEEE Transactions on Power Delivery**, v. 13, n. 2, p. 605–614, 1998.

LJUNG, L. **System Identification: Theory for the user**. Upper Saddle River, NJ, USA: Prentice-Hall PTR, 1999.

MORALES, J. *et al.* Pole-selective residue perturbation technique for passivity enforcement of fdnes. **IEEE Transactions on Power Delivery**, v. 33, n. 6, p. 2746–2754, 2018.

MOUHAI DALI, A. *et al.* Electromagnetic transient analysis of transmission line based on rational krylov approximation. **IEEE Transactions on Power Delivery**, v. 36, n. 5, p. 2913–2920, 2021.

NISE, N. S. **Engenharia de Sistemas de Controle**. 6. ed. Pomona, California, USA: LTC, 2013. ISBN 9788521621362.

OPPENHEIM, A. V.; WILLISKY, A. S. **Sinais e Sistemas**. 2. ed. São Paulo, SP, Brasil: Pearson Education do Brasil Ltda., 2010. ISBN 9788543013800.

RESEARCH, I. C. **CVX: Matlab Software for Disciplined Convex Programming, version 2.0**, . 2012. <http://cvxr.com/cvx>.

TEKTRONIX . **Introduction to VNA Basics: Primer**. [S.l.], 2017. Disponível em: <https://tek.com/en/documents/primer/what-vector-network-analyzer-and-how-does-it-work>. Acesso em: 12 nov. 2025.

TOMMASI, L. D. *et al.* Validation of positive fraction vector fitting algorithm in the identification of passive immittances. **omega [rad/s]**, v. 10, n. 2, p. 100, 2008.

TOMMASI, L. de; DESCHRIJVER, D.; DHAENE, T. Single-input-single-output passive macromodeling via positive fractions vector fitting. *In: 2008 12TH IEEE WORKSHOP ON SIGNAL PROPAGATION ON INTERCONNECTS*. 2008. **Anais [...]** [S.l.: s.n.], 2008. p. 1–2.

UDWADIA, F. E. Boundary control, quiet boundaries, super-stability and super-instability. **Applied Mathematics and Computation**, v. 164, n. 2, p. 327–349, 2005. ISSN 0096-3003. 12th International Workshop on Dynamics and Control.

USMAN, K.; RAMDHANI, M. Comparison of classical interpolation methods and compressive sensing for missing data reconstruction. *In: 2019 IEEE INTERNATIONAL CONFERENCE ON SIGNALS AND SYSTEMS (ICSIGSYS)*. 2019. **Anais [...]** [S.l.: s.n.], 2019. p. 29–33.

WILLEMS, J. C. Dissipative dynamical systems. **European Journal of Control**, v. 13, n. 2, p. 134–151, 2007. ISSN 0947-3580.

WONG, W. K.; ZHOU, J. Using cvx to construct optimal designs for biomedical studies with multiple objectives. **Journal of Computational and Graphical Statistics**, ASA Website v. 32, n. 2, p. 744–753, 2023. Disponível em: <https://doi.org/10.1080/10618600.2022.2104858>.

ZADEH, L. A. From circuit theory to system theory. **Proceedings of the IRE**, v. 50, n. 5, p. 856–865, 1962.