



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE TELEINFORMÁTICA
DOUTORADO EM ENGENHARIA DE TELEINFORMÁTICA

IGOR ROCHA DE SOUSA

**CONTRIBUIÇÕES À SÍNTESE DE CONTROLADORES FRACIONÁRIOS:
SIMULAÇÃO NUMÉRICA, IDENTIFICAÇÃO, SINTONIA DE PARÂMETROS,
ANÁLISE DE ESTABILIDADE E IMPLEMENTAÇÃO EMBARCADA**

FORTALEZA

2025

IGOR ROCHA DE SOUSA

CONTRIBUIÇÕES À SÍNTESE DE CONTROLADORES FRACIONÁRIOS: SIMULAÇÃO
NUMÉRICA, IDENTIFICAÇÃO, SINTONIA DE PARÂMETROS, ANÁLISE DE
ESTABILIDADE E IMPLEMENTAÇÃO EMBARCADA

Tese apresentada ao Programa de Pós-Graduação
em Engenharia de Teleinformática do Centro de
Tecnologia da Universidade Federal do Ceará,
como requisito parcial à obtenção do título de
doutor em Engenharia de Teleinformática. Área de
Concentração: Sinais e Sistemas

Orientador: Prof. Dr. Guilherme de Alencar Barreto

Coorientador: Prof. Dr. Kléber Zuza Nóbrega

FORTALEZA

2025

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

S696c Sousa, Igor Rocha de.

Contribuições à síntese de controladores fracionários : simulação numérica, identificação, sintonia de parâmetros, análise de estabilidade e implementação embarcada / Igor Rocha de Sousa. – 2025.
184 f. : il. color.

Tese (doutorado) – Universidade Federal do Ceará, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia Civil: Recursos Hídricos, Fortaleza, 2025.

Orientação: Prof. Dr. Guilherme de Alencar Barreto.

Coorientação: Prof. Dr. Kléber Zuza Nóbrega.

1. controlador PID fracionário. 2. análise de estabilidade. 3. função Lambert-Tsallis. 4. identificação de sistemas. 5. sistemas embarcados. I. Título.

CDD 627

IGOR ROCHA DE SOUSA

CONTRIBUIÇÕES À SÍNTESE DE CONTROLADORES FRACIONÁRIOS: SIMULAÇÃO
NUMÉRICA, IDENTIFICAÇÃO, SINTONIA DE PARÂMETROS, ANÁLISE DE
ESTABILIDADE E IMPLEMENTAÇÃO EMBARCADA

Tese apresentada ao Programa de Pós-Graduação em Engenharia de Teleinformática do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do título de doutor em Engenharia de Teleinformática. Área de Concentração: Sinais e Sistemas

Aprovada em: 20 de Junho de 2025.

BANCA EXAMINADORA

Prof. Dr. Guilherme de Alencar Barreto (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Kléber Zuza Nóbrega (Coorientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Bismark Claure Torrico
Universidade Federal do Ceará (UFC)

Prof. Dr. Cláudio Marques de Sá Medeiros
Instituto Federal do Ceará (IFCE)

Prof. Dr. Pedro Luis Dias Peres
Universidade Estadual de Campinas (UNICAMP)

Prof. Dr. Julio Elias Normey Rico
Universidade Federal de Santa Catarina (UFSC)

À minha família.

AGRADECIMENTOS

Ao meu orientador, Prof. Dr. Guilherme de Alencar Barreto, pelas orientações e opiniões sempre precisas. Sou extremamente grato pela disposição e generosidade em atender às minhas solicitações, tanto acadêmicas quanto pessoais. Agradeço também pelo vasto conhecimento técnico que permeia toda esta tese e pela confiança depositada no desenvolvimento deste trabalho sobre o instigante tema de sistemas fracionários.

Ao meu coorientador, Prof. Dr. Kléber Zuza Nóbrega, cuja marcante paixão pela matemática contagia e inspira todos ao seu redor, sou grato pelos valiosos ensinamentos e pela ajuda nos desenvolvimentos relacionados à função Lambert-Tsallis.

Agradeço à Universidade Federal do Ceará (UFC) pela estrutura, suporte e ambiente acadêmico que possibilitaram a realização deste trabalho. Em especial, ao laboratório SPIRAL e aos seus integrantes, pelo ambiente colaborativo e pelas discussões enriquecedoras ao longo do desenvolvimento desta pesquisa.

A presente tese foi realizada com apoio financeiro da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), Brasil. Processo nº 2023/12842-8. Adicionalmente, contou com o suporte do Centro de Referência em Inteligência Artificial (CEREIA), por meio dos convênios 23067.055633/2021-19 (UFC) e 2020/09706-7 (FAPESP), e da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) – Código de Financiamento 001.

Agradeço imensamente à minha família, especialmente aos meus pais, Izaias Carlos de Sousa e Ivonete Rocha de Sousa, e à minha noiva, Patrícia Tavares Leitão, pelo apoio incondicional à minha decisão de seguir na pesquisa e pelo suporte nos momentos de exaustão.

“There is a universe of mathematics lying
between the complete differentiations and
integrations, and that fractional operators push
themselves forward sometimes, and are just as
real as others.” (Oliver Heaviside, 1871)

RESUMO

Nesta tese, são desenvolvidas e implementadas propostas inovadoras para diferentes etapas que compõem o projeto eficiente de controladores de ordem fracionária, a saber: simulação numérica, identificação de sistemas, análise analítica de estabilidade, sintonia de parâmetros e implementação embarcada. Para a etapa de análise de estabilidade, desenvolve-se uma proposta completamente inédita com o auxílio da função Lambert-Tsallis para determinar os polos de sistemas de controle fracionário cuja equação característica assume a forma genérica de um trinômio dado por $a_n s^n + a_m s^m + a_0 = 0$, $\forall n, m \in \mathbb{Q}^*$ e $a_n a_m a_0 \neq 0$. Como consequência, desenvolve-se também uma equação analítica capaz de definir os limites dos parâmetros do controlador que asseguram a estabilidade de tais sistemas. A fim de ilustrar o uso das ferramentas desenvolvidas, são considerados tanto plantas hipotéticas extraídas da literatura quanto plantas reais. Essa capacidade analítica é estendida para a sintonia de parâmetros, de modo que o espaço de busca de um algoritmo de otimização assume o hipervolume correspondente à região de estabilidade do sistema de controle em questão. Para validar a abordagem em um contexto de aplicação real, é desenvolvido um controlador fracionário discreto embarcado em uma incubadora neonatal para controle de temperatura. O modelo da planta foi obtido através de um algoritmo proposto nesta tese para identificação recursiva de sistemas fracionários, baseado no algoritmo LMS (*least mean square*). O modelo de ordem fracionária alcançou VAF (*variance accounted for*) de 97,84%. Durante a sintonia do controlador, as simulações computacionais são realizadas através de equações discretas desenvolvidas nesta tese, apresentando tempo de processamento da ordem de 250 vezes menor em relação às *toolboxes* disponíveis na literatura da área. Por fim, propõe-se uma implementação deste controlador no microcontrolador de 8 *bits* da incubadora, através de sua representação por uma rede neural MLP (*multilayer perceptron*) com o objetivo de mitigar a perda de otimalidade da sintonia de parâmetros que ocorre nos métodos tradicionais de discretização destes controladores. A proposta obteve RMSE (*root mean square error*) = 0,0715, frente aos 0,5032 da técnica de Merrikh-Bayat. No sistema embarcado, com apenas 4 neurônios na camada oculta, o controlador fracionário neural demandou apenas 6,511 ms para execução, ocupando 1556 *bytes* de memória de programa e 241 *bytes* de RAM. A resposta do sistema de controle apresentou um sobressinal de 1,66% e um tempo de acomodação de 8.183 s.

Palavras-chave: controlador PID fracionário; análise de estabilidade; função Lambert-Tsallis; identificação de sistemas; simulação numérica; regressão não linear; sistemas embarcados.

ABSTRACT

In this thesis, innovative proposals are developed and implemented for different stages that constitute the efficient design of fractional-order controllers, namely: numerical simulation, system identification, analytical stability analysis, parameter tuning, and embedded implementation. For the stability analysis stage, a completely novel approach is developed using the Lambert-Tsallis function to determine the poles of fractional-order control systems whose characteristic equation assumes the generic form of a trinomial given by $a_n s^n + a_m s^m + a_0 = 0$, $\forall, n, m \in \mathbb{Q}^*$ and $a_n a_m a_0 \neq 0$. As a consequence, an analytical equation is also developed to define the bounds of the controller parameters that ensure the stability of such systems. To illustrate the use of the developed tools, both hypothetical plants from the literature and real-world plants are considered. This analytical capability is extended to parameter tuning, so that the search space of an optimization algorithm is constrained to the hypervolume corresponding to the stability region of the control system under consideration. To validate the approach in a real application context, a discrete fractional-order controller is developed and embedded in a neonatal incubator for temperature control. The plant model was obtained using a recursive identification algorithm for fractional-order systems proposed in this thesis, based on the LMS (least mean square) algorithm. The resulting fractional-order model achieved a VAF (variance accounted for) of 97.84%. During the controller tuning process, the computational simulations are carried out using discrete equations developed in this thesis, resulting in a processing time approximately 250 times shorter than the elapsed time required by toolboxes available in the literature. Finally, an implementation of this controller is proposed for the 8-bit microcontroller of the incubator, using an MLP (multilayer perceptron) neural network representation to mitigate the loss of optimality in parameter tuning observed in traditional discretization methods for such controllers. The proposed approach achieved an RMSE (root mean square error) of 0.0715, compared to 0.5032 obtained using the Merrikh-Bayat technique. In the embedded system, with only 4 neurons in the hidden layer, the neural fractional-order controller required just 6.511 ms for execution, occupying 1556 bytes of program memory and 241 bytes of RAM. The response of the control system exhibited an overshoot of 1.66% and a settling time of 8183 s.

Keywords: fractional-order PID controller; stability analysis; Lambert-Tsallis function; system identification; numerical simulation; nonlinear regression; embedded systems.

LISTA DE FIGURAS

Figura 1 – Publicações da Elsevier com os termos <i>fractional-order</i> e <i>fractional-order controller</i>	26
Figura 2 – Possibilidades de atuação do controlador $PI^\lambda D^\mu$	39
Figura 3 – Típico sistema de controle em malha fechada.	40
Figura 4 – Análise da região de instabilidade no plano complexo transformado $\sigma = s^{\frac{1}{\gamma}}$	42
Figura 5 – Polos falsos e genuínos nos planos σ e s para o Exemplo 2.5.1.	46
Figura 6 – Transformação $\sigma = s^{\frac{1}{\gamma}}$ em uma superfície de Riemann para o Exemplo 2.5.1, $\gamma = 2$	46
Figura 7 – Polos falsos e genuínos nos planos σ e s para o Exemplo 2.5.2.	49
Figura 8 – Transformação $\sigma = s^{\frac{1}{\gamma}}$ em uma superfície de Riemann para o Exemplo 2.5.2, $\gamma = 10$	49
Figura 9 – Polos falsos e genuínos nos planos σ e s para o Exemplo 2.5.3.	51
Figura 10 – Transformação $\sigma = s^{\frac{1}{\gamma}}$ em uma superfície de Riemann para o Exemplo 2.5.3, $\gamma = 10$	52
Figura 11 – Gráficos de $W_r(z) \times z \subset \mathbb{R} \times \mathbb{R}$, $r \in \mathbb{Q}_+^*$	60
Figura 12 – Evolução de $W_r(z) \times z \subset \mathbb{R} \times \mathbb{R}$ para diferentes valores de $r \in \mathbb{Q}_+^*$	62
Figura 13 – Comparação das três morfologias de $W_r(z) \in \mathbb{R}$ para valores próximos de r	63
Figura 14 – Funções Lambert e Lambert-Tsallis para diferentes valores de r	64
Figura 15 – Comparação entre as funções Lambert e Lambert-Tsallis para r do tipo 1.	64
Figura 16 – Superfícies de Riemann de $W_r(z) \times z \subset \mathbb{C} \times \mathbb{C}$, $r \in \mathbb{Q}_+^*$	66
Figura 17 – Superfícies de Riemann de $W_r(z) \times z \subset \mathbb{R} \times \mathbb{C}$, $r \in \mathbb{Q}_+^*$	67
Figura 18 – Diferentes gráficos de $W_r(z) \times z \subset \mathbb{C} \times \mathbb{R}$, $r \in \mathbb{Q}_+^*$	68
Figura 19 – Evolução de $W_r(z) \times z \subset \mathbb{C} \times \mathbb{R}$ para diferentes valores de $r \in \mathbb{Q}_+^*$	69
Figura 20 – Planos $W_r(z)$, s e σ para o Exemplo 4.2.1.	77
Figura 21 – Resumo do método proposto em comparação com o método clássico.	78
Figura 22 – Busca de z_c para o Exemplo 4.2.2.	81
Figura 23 – Teste de estabilidade <i>a posteriori</i> do Exemplo 4.2.3.	83
Figura 24 – Exemplos pontuais da busca de z_c para o Exemplo 4.2.4.	86
Figura 25 – Busca de z_c em $\lambda = 0,898$ para Exemplo 4.2.4.	87
Figura 26 – Gráfico $z_c \times \lambda$ para caso 9 da Tabela 6 com $\theta_c = 112,5^\circ$	87
Figura 27 – Volume de estabilidade do Exemplo 4.2.4.	88

Figura 28 – Teste de estabilidade <i>a posteriori</i> do Exemplo 4.2.4, com passo de 0,01. . . .	88
Figura 29 – Respostas ao degrau unitário de $G(s) = \frac{5}{s^{\alpha+5}}$ para diferentes valores de α	93
Figura 30 – Resposta ao degrau da integral fracionária de ordem 0,5 para diferentes janelas de memória.	94
Figura 31 – Respostas ao degrau unitário dos diferentes métodos de simulação de sistemas de ordem fracionária.	97
Figura 32 – Exemplo de mau uso da <i>toolbox</i> FOTF para resposta ao degrau.	99
Figura 33 – Simulação de um sistema com duas diferentes discretizações do controlador $PI^{\lambda}D^{\mu}$	108
Figura 34 – Diagrama de blocos da Equação 6.3.	112
Figura 35 – Diagrama de blocos de Equação 6.3 em função de α	114
Figura 36 – Comparação entre dados de identificação e modelo estimado do caso 1. . . .	115
Figura 37 – Diagrama de blocos da Equação 6.17.	116
Figura 38 – Comparação entre dados de identificação e modelo estimado do caso 2. . . .	118
Figura 39 – Diagrama de blocos da Equação 6.28.	119
Figura 40 – Comparação entre dados de identificação e modelo estimado do caso 3. . . .	121
Figura 41 – Diagrama de blocos da Equação 6.46.	124
Figura 42 – Fluxograma da metodologia de sintonia de parâmetros usando PSO conjuntamente com a função de Lambert-Tsallis.	132
Figura 43 – Implementação do sistema de controle com a aproximação de Merrikh-Bayat. . .	136
Figura 44 – Diagrama do sistema de controle usado para o treinamento da rede neural. .	137
Figura 45 – Implementação do sistema de controle fracionário com o controlador $PI^{\lambda}D^{\mu}$ neural.	137
Figura 46 – Etapas utilizadas para o projeto de sistema de controle fracionário.	140
Figura 47 – Resultados da identificação dos dados de aquisição da incubadora.	142
Figura 48 – FAC dos resíduos.	143
Figura 49 – Gráfico $z_c \times \lambda$ para caso 9 da Tabela 6 com $\theta_c = 99^{\circ}$	145
Figura 50 – Volume de estabilidade do sistema de controle fracionário proposto.	145
Figura 51 – Teste de estabilidade <i>a posteriori</i> do sistema proposto, com passo de 0,01. . .	146
Figura 52 – Desempenho das 10 partículas g^{best}	148
Figura 53 – Evoluções dos parâmetros da melhor partícula g^{best}	149
Figura 54 – Resposta temporal do sistema de controle para 100.000 segundos de simulação. .	150

Figura 55 – Planos s e σ para o sistema de controle fracionário proposto.	151
Figura 56 – Exemplo de saturação dinâmica da partícula g^{best}	152
Figura 57 – Exemplo de resultados instáveis com a utilização do algoritmo PSO.	153
Figura 58 – Comparação entre as respostas dos controladores.	154
Figura 59 – Comparação entre a saída da rede neural e a saída desejada.	156
Figura 60 – Comparação entre as discretizações do controlador PI^λ	157
Figura 61 – Estágios do sistema embarcado no microcontrolador ATmega328p.	158
Figura 62 – Respostas temporais reais dos sistemas de controle fracionários embarcados.	159
Figura 63 – Incubadora neonatal C186 TS, FANEM®.	178
Figura 64 – Esquema eletrônico do acionamento do resistor de aquecimento.	178
Figura 65 – Comportamento da tensão em carga resistiva por ângulo de disparo em TRIAC.	179
Figura 66 – Microcontrolador utilizado e suas conexões com a placa eletrônica da incubadora.	179
Figura 67 – Ensaio da tensão média por registrador do <i>timer</i> do microcontrolador.	180
Figura 68 – Normalização entre tensão média aplicada e registrador do <i>timer</i>	181
Figura 69 – Ensaio realizado para identificação da planta experimental (incubadora neonatal).	182
Figura 70 – Conjunto de dados para identificação da planta experimental (incubadora neonatal).	182
Figura 71 – Sistema de controle de temperatura sugerido para a incubadora neonatal.	183

LISTA DE TABELAS

Tabela 1 – Valores de σ_j e $ \arg(\sigma_j) $ para o Exemplo 2.5.2.	48
Tabela 2 – Valores das raízes e dos resíduos dos polinômios para o Exemplo 2.5.2. . . .	48
Tabela 3 – Valores de σ_j e $ \arg(\sigma_j) $ para o Exemplo 2.5.3.	51
Tabela 4 – Valores das raízes e dos resíduos dos polinômios para o Exemplo 2.5.3. . . .	51
Tabela 5 – Sequência periódica dos tipos 1, 2 e 3 em função do número de casas decimais (x).	64
Tabela 6 – Exemplos de sistemas de controle adequados para análise via função Lambert- Tsallis.	73
Tabela 7 – Comparação entre os resultados obtidos para o Exemplo 4.2.1.	76
Tabela 8 – Valores de RMSE para diferentes tamanhos da janela L_0	94
Tabela 9 – Modelos utilizados para comparação presente na Figura 31 e Tabela 10. . . .	98
Tabela 10 – Métricas dos diferentes métodos de simulação de sistemas de ordem fracionária.	98
Tabela 11 – Média e desvio padrão do tempo de execução.	108
Tabela 12 – Resultados da identificação do caso 1.	115
Tabela 13 – Resultados da identificação do caso 2.	118
Tabela 14 – Resultados da identificação do caso 3.	121
Tabela 15 – Métricas e parâmetros da identificação da incubadora.	143
Tabela 16 – Parâmetros ótimos do controlador PI^λ	148
Tabela 17 – Resultados obtidos para o sistema de controle fracionário proposto.	150
Tabela 18 – Comparação do desempenho dos controladores PI^λ e PI.	154
Tabela 19 – Parâmetros da discretização de Merrikh-Bayat.	155
Tabela 20 – Desempenho da rede neural para diferentes neurônios na camada oculta. . .	156
Tabela 21 – Desempenho computacional do sistema embarcado.	158

LISTA DE ALGORITMOS

Algoritmo 1	–	Resolução da Equação da Função Lambert-Tsallis.	59
Algoritmo 2	–	Determinação do(s) valor(es) de $W_r^*(z)$ e de s^*	76
Algoritmo 3	–	Determinação do valor de z_c	80
Algoritmo 4	–	Obtenção do tipo de r	85
Algoritmo 5	–	Correção do valor de λ	86

LISTA DE ABREVIATURAS E SIGLAS

AR	Autorregressivos
ARX	Autorregressivo com entradas exógenas
CEREIA	Centro de Referência em Inteligência Artificial
FAC	Função de autocorrelação
FAPESP	Fundação de Amparo à Pesquisa do Estado de São Paulo
GL	Grünwald-Letnikov
ITSE	Integral of time-weighted squared error
$LD\text{-}PI^\lambda D^\mu$	$PI^\lambda D^\mu$ discreto de memória longa
LMI	Linear matrix inequalities
LMS	Least mean square
LT	Lambert-Tsallis
MBE	Erro médio de viés
MLP	Multilayer Perceptron
P	Proporcional
PD	Proporcional-derivativo
PD^μ	Proporcional-derivativo de ordem fracionária
PI	Proporcional-integrativo
PI^λ	Proporcional-integrativo de ordem fracionária
$PI^\lambda D^\mu$	Proporcional-integrativo-derivativo de ordem fracionária
PID	Proporcional-integrativo-derivativo
PRAS	Pseudo-random analog signal
PRBS	Pseudo-random binary signal
PSO	Particle swarm optimization
PSO-LT	PSO - Lambert-Tsallis
RC	Circuito resistor-capacitor
RL	Riemann-Liouville

RMSE	Root mean square error
RT	Resposta transitória
SISFRAC	Simulação computacional de sistemas de ordem fracionária
SISO	Single-input single-output
SP	Setpoint
SPIRAL	Signal and information processing for data analysis and learning systems
TRIAC	Triode for alternating current
UFC	Universidade Federal do Ceará
VAF	Variance accounted for

LISTA DE SÍMBOLOS

${}_{t_0}\mathcal{D}_t^\alpha f(t)$	Derivada fracionária de ordem α no instante t .
${}_t^C\mathcal{D}_t^\alpha f(t)$	Derivada fracionária de Caputo de ordem α no instante t .
${}_0^{GL}\mathcal{D}_t^\alpha f(t)$	Derivada fracionária de Grünwald-Letnikov de ordem α no instante t .
${}_{t_0}^{RL}\mathcal{D}_t^\alpha f(t)$	Derivada fracionária de Riemann-Liouville de ordem α no instante t .
a_0	Coeficiente do monômio independente de um trinômio fracionário.
a_{0_c}	a_0 crítico.
a_m	Coeficiente do monômio com menor expoente de um trinômio fracionário.
a_{m_c}	a_m crítico.
a_n	Coeficiente do monômio com maior expoente de um trinômio fracionário.
α	Ordem da derivada fracionária.
β	Ordem da derivada fracionária.
b	Ganho estático da malha de retroalimentação.
γ	Denominador comum dos expoentes fracionários para o teorema de Matignon.
γ	Ordem da derivada fracionária.
δ	Ordem da derivada fracionária.
c_1	Coeficiente constante de aprendizagem cognitiva (PSO).
c_2	Coeficiente constante de aprendizagem social (PSO).
$\cos(\cdot)$	Função trigonométrica cosseno.
$d_k(t)$	Rótulo do neurônio k .
e	Número de Euler.
$e[k]$	Sinal de erro discreto do sistema de controle.
e_q	q -exponencial, ou exponencial de Tsallis.
$E(s)$	Sinal de erro do sistema de controle.
E_{ep}	Erro de estado permanente.
$f(t)$	Função genérica no tempo.
$G_c(s)$	Função de transferência de um controlador.

$G_{cl}(s)$	Função de transferência do sistema de controle em malha fechada.
$G_p(s)$	Função de transferência de uma planta.
$\Gamma(\cdot)$	Função Gama.
gcd	Greatest common divisor (máximo divisor comum).
$H(s)$	Função de transferência da malha de retroalimentação.
h	Passo de cálculo presente na definição de derivada fracionária de Grünwald-Letnikov.
J	Função custo.
J_{total}	Função objetivo utilizada na otimização do controlador.
J_{RT}	Função objetivo com termos de ponderação.
k	Índice de iteração.
k_d	Ganho derivativo.
k_i	Ganho integrativo.
$k_i^{(j)}$	Valor de k_i da j -ésima partícula do enxame.
$k_i^{(min)}$	Valor mínimo de k_i .
$k_i^{(max)}$	Valor máximo de k_i .
k_{i_c}	k_i crítico.
$k_{i_c}^{(j)}[k]$	Valor de k_{i_c} da j -ésima partícula na iteração k .
k_p	Ganho proporcional.
$k_p^{(j)}$	Valor de k_p da j -ésima partícula do enxame.
$k_p^{(min)}$	Valor mínimo de k_p .
$k_p^{(max)}$	Valor máximo de k_p .
k_{p_c}	k_p crítico.
K_p	Valor de k_p na discretização de Merrikh-Bayat.
K_i	Valor de k_i na discretização de Merrikh-Bayat.
K_d	Valor de k_d na discretização de Merrikh-Bayat.
L	Quantidade de dados utilizados para identificação.
L_0	Profundidade da janela de memória.

$L1$	Norma $L1$ (soma dos valores absolutos).
λ	Ordem de integração.
$\lambda^{(j)}$	Valor de λ da j -ésima partícula do enxame.
$\lambda^{(min)}$	Valor mínimo de λ .
$\lambda^{(max)}$	Valor máximo de λ .
λ_{max}	Maior autovalor da matriz de autocorrelação \mathbf{R} .
μ	Ordem de derivação.
μ	Taxa de aprendizado do algoritmo LMS.
μ_{reg}	Parâmetro de ponderação associado ao termo de regularização.
$\mu_{inicial}$	Taxa de aprendizado inicial do algoritmo LMS.
$\mu[k]$	Vetor de taxas de aprendizagem do algoritmo LMS na iteração k .
M	Tamanho da memória na aproximação de Merrikh-Bayat.
M_s	Máximo sobressinal.
m	Menor expoente de um trinômio fracionário.
m_{ki}	Peso sináptico que conecta o neurônio oculto i ao neurônio de saída k .
n	Maior expoente de um trinômio fracionário.
$Q(s)$	Equação característica do sistema em malha fechada em s .
$Q(\sigma)$	Equação característica do sistema em malha fechada em σ .
q	Parâmetro da exponencial de Tsallis.
R	Coeficiente de correlação.
\mathbf{R}	Matriz de autocorrelação do algoritmo LMS.
$R(s)$	Sinal de referência de controle.
R^2	Coeficiente de determinação.
r	Parâmetro da exponencial de Tsallis ($r = \frac{1}{1-q}$).
s	Variável complexa contínua.
s	Segundos (unidade de tempo).
s^*	Raízes genuínas no plano s .
σ	Variável na transformação polinomial.

σ^*	Raízes no plano σ que levam a raízes genuínas em s .
σ_x^2	Variância do regressor.
σ_R	Desvio padrão de R .
T	Passo de cálculo na discretização de Euler.
T_a	Tempo de acomodação.
T_s	Tempo de subida.
t_{final}	Tempo de simulação.
$\tan(\cdot)$	Função tangente.
$u[k]$	Saída discreta do controlador fracionário.
$U(s)$	Saída do controlador.
V_{med}	Tensão média.
V_s	Valor eficaz da tensão de entrada.
w	Fator de peso inercial (PSO).
w_i	i -ésimo coeficiente da definição de Grünwald-Letnikov.
α_{w_i}	i -ésimo coeficiente w da definição de derivada fracionária de ordem α .
β_{w_i}	i -ésimo coeficiente w da definição de derivada fracionária de ordem β .
γ_{w_i}	i -ésimo coeficiente w da definição de derivada fracionária de ordem γ .
δ_{w_i}	i -ésimo coeficiente w da definição de derivada fracionária de ordem δ .
w_0, \dots, w_6	Constantes da discretização de Merrikh-Bayat.
w_b	Valor de $W_r(z_b)$.
w_{final}	Valor final de w (PSO).
$w_{inicial}$	Valor inicial de w (PSO).
w_{ij}	Peso sináptico que conecta a entrada j ao neurônio oculto i .
$W(z)$	Função de Lambert.
$W_q(z)$	Função de Lambert-Tsallis.
$W_r(z)$	Função de Lambert-Tsallis (em termos de r).
$W_r(z)^*$	Valores de $W_r(z)$ que levam a raízes genuínas em s .
x	Número de casas decimais consideradas em r .

$y[k]$	Saída discreta da planta.
$\hat{y}[k]$	Estimativa de $y[k]$.
$Y(s)$	Saída da planta e do sistema de controle.
z	Variável complexa discreta.
z	Argumento da função de Lambert.
z_b	Ponto de ramificação da função Lambert-Tsallis.
z_c	z crítico.
$\delta_k^{(o)}(t)$	Gradiente local do neurônio k .
$\delta_i^{(h)}(t)$	Gradiente local do neurônio oculto i .
θ_c	Ângulo crítico para estabilidade no plano $W_r(z)$.
θ_d	Ângulo de disparo do TRIAC.
$\theta_i(t)$	Limiar do neurônio oculto i .
$\theta_k^{(o)}(t)$	Limiar do neurônio de saída k .
θ_s	Ângulo auxiliar no plano s .
θ_σ	Ângulo auxiliar no plano σ .
$\eta(t)$	Taxa de aprendizado da rede neural.
ρ	Fator de constrição (PSO).
τ	Total de iterações do processo de identificação.
$\varphi_i(\cdot)$	Função tangente hiperbólica.
ψ_1, ψ_2	Variáveis aleatórias uniformes em $(0,1)$.
ω_c	Ganho de frequência cruzada.
\circ	Produto de Hadamard.
$g^{best}[k]$	Melhor posição do enxame (PSO).
$p_j[k]$	Posição da j -ésima partícula (PSO).
$p_j^{best}[k]$	Melhor posição da partícula j (PSO).
p_{min}	Valores mínimos para a partícula j .
$p_j^{(max)}[k]$	Valores máximos para a partícula j .
W	Vetor de parâmetros w .

${}^{\alpha}\mathbf{W}[k]$	Vetor de parâmetros w para derivada de ordem α .
${}^{\delta}\mathbf{W}[k]$	Vetor de parâmetros w para derivada de ordem δ .
\mathbf{Y}	Vetor de valores passados de $y[k]$.
$\hat{\Theta}[k]$	Vetor de estimativas dos parâmetros do modelo.
$\mathbf{X}[k]$	Matriz de regressores (LMS).
\mathbf{K}	Vetor de ganhos do controlador.

SUMÁRIO

1	INTRODUÇÃO	25
1.1	Contextualização e justificativa	26
1.2	Objetivos	29
1.2.1	<i>Objetivo geral</i>	<i>29</i>
1.2.2	<i>Objetivos específicos</i>	<i>29</i>
1.3	Produção acadêmica	29
1.4	Organização da tese	31
2	DERIVADAS E INTEGRAIS FRACIONÁRIAS EM SISTEMAS DE CONTROLE	33
2.1	Definições de derivada e integral fracionárias	33
2.1.1	<i>Demonstração da definição de Grünwald-Letnikov</i>	<i>35</i>
2.2	Controladores de ordem fracionária	37
2.3	Plantas de ordem fracionária	39
2.4	Sistemas de controle em malha fechada	40
2.5	Estabilidade de sistemas de controle de ordem fracionária	41
<i>Exemplo 2.5.1</i>	<i>caso $Q(s) = s^2 + s^{1,5} - 1$</i>	<i>43</i>
<i>Exemplo 2.5.2</i>	<i>caso $Q(s) = s^2 + s^{0,1} + 1$</i>	<i>47</i>
<i>Exemplo 2.5.3</i>	<i>caso $Q(s) = s^{1,1} + s^{0,5} + 1$</i>	<i>50</i>
2.5.1	<i>Considerações sobre o Teorema de Matignon</i>	<i>52</i>
3	A FUNÇÃO LAMBERT-TSALLIS	54
3.1	A função Lambert-Tsallis, $W_q(z)$	54
3.1.1	<i>Propriedades de $W_r(z)$</i>	<i>56</i>
3.1.2	<i>Algoritmo para determinação de $W_r(z)$</i>	<i>58</i>
3.2	Polimorfismo de $W_r(z)$	59
3.3	Um novo desenvolvimento: análise de $W_r(z) \in \mathbb{C}$	65
4	ESTABILIDADE DE SISTEMAS FRACIONÁRIOS: UMA NOVA PROPOSTA	70
4.1	Solução analítica de equações trinômiais generalizadas	70
4.2	Análise de estabilidade de sistemas pela função $W_r(z)$	72
<i>Exemplo 4.2.1</i>	<i>Encontrando as raízes da equação característica de ordem fracionária</i>	<i>74</i>

4.2.1	<i>Solução analítica para os limites dos parâmetros do controlador via $W_r(z)$</i>	78
Exemplo 4.2.2	<i>Condições de estabilidade para controlador PD e planta $G_{p2}(s)$</i>	79
Exemplo 4.2.3	<i>Condições de estabilidade para controlador PD e planta $G_{p3}(s)$</i>	82
Exemplo 4.2.4	<i>Condições de estabilidade para controlador PI^λ com planta $G_{p2}(s)$</i>	83
4.3	Considerações sobre os métodos propostos	89
5	SIMULAÇÃO DE SISTEMAS DINÂMICOS DE ORDEM FRACIONÁRIA	90
5.1	Equação discreta de um sistema de ordem fracionária	90
5.2	Sobre <i>softwares</i> para simulação de sistemas de ordem fracionária	95
5.3	Desenvolvimento de uma equação discreta de uso geral	100
5.4	Discretização do controlador $PI^\lambda D^\mu$ — versão 1	103
5.5	Discretização do controlador $PI^\lambda D^\mu$ — versão 2	104
5.6	Comparação entre as duas versões de discretização do controlador $PI^\lambda D^\mu$	107
5.7	Considerações finais	108
6	IDENTIFICAÇÃO DE SISTEMAS DE ORDEM FRACIONÁRIA	109
6.1	Inspiração para a criação do algoritmo de identificação proposto	111
6.2	Caso 1: planta $G(s) = \frac{1}{s^\alpha}$	111
6.2.1	<i>Validação do algoritmo proposto para o caso 1</i>	115
6.3	Caso 2: planta $G(s) = \frac{1}{s^\alpha + a}$	116
6.3.1	<i>Validando o algoritmo proposto para o caso 2</i>	117
6.4	Caso 3: planta $G(s) = \frac{c}{s^\alpha + a}$	118
6.4.1	<i>Validando o algoritmo proposto para o caso 3</i>	120
6.5	Caso 4: planta generalizada $G(s) = \frac{fs^\gamma + \dots + gs^\delta + c}{s^\alpha + \dots + ds^\beta + a}$	122
6.6	Condições de convergência do algoritmo	126
6.7	Considerações finais sobre o algoritmo proposto	127
7	SÍNTESE DE CONTROLADORES DE ORDEM FRACIONÁRIA	128
7.1	Sintonia de parâmetros do controlador de ordem fracionária	128
7.1.1	<i>Fundamentos do algoritmo PSO</i>	128
7.1.2	<i>Sintonia estável de sistemas ordem fracionária via PSO e função Lambert-Tsallis</i>	130
7.2	Implementação embarcada de controladores de ordem fracionária	132
7.2.1	<i>Aproximação discreta de controladores $PI^\lambda D^\mu$</i>	133
7.2.2	<i>Aplicação da rede neural MLP</i>	136

7.2.2.1	<i>Avaliação do sistema simulado em uma plataforma embarcada</i>	137
8	ESTUDO DE CASO: INCUBADORA NEONATAL	139
8.1	Identificação da incubadora neonatal	140
8.2	Sintonia dos parâmetros de um controlador de ordem fracionária	143
8.2.1	<i>Estabilidade a priori via função Lambert-Tsallis</i>	143
8.2.2	<i>Aplicação do algoritmo PSO-LT</i>	147
8.3	Implementação embarcada do controlador PI^λ	155
9	CONCLUSÕES	160
9.1	Contribuições da Tese	160
9.2	Trabalhos Futuros	162
	REFERÊNCIAS	163
	APÊNDICES	174
	APÊNDICE A –DESCRIÇÃO DA MÁQUINA UTILIZADA PARA SI- MULAÇÕES COMPUTACIONAIS	174
	APÊNDICE B – REGRESSÃO BASEADA EM MLP	175
	APÊNDICE C –RETROFIT DE UMA INCUBADORA NEONATAL . .	177
C.1	Definição do sinal de controle	180
C.2	Aquisição de dados	181
C.3	Sistema de controle	181

1 INTRODUÇÃO

O cálculo fracionário tem atraído significativamente a atenção das comunidades acadêmica e industrial pela capacidade de modelar dinâmicas de ordem não inteira, como a derivada de ordem 1,23 de $f(t)$, representada por $\frac{d^{1,23}f(t)}{dt^{1,23}}$. Seu uso pode generalizar técnicas matemáticas clássicas usadas em processamento de sinais, tais como medidas de entropia (LOPES; MACHADO, 2020), transformada de Fourier (GOLMANKHANEH *et al.*, 2021), *wavelet* (ONG; NG, 2025) e filtro de Kalman (SOLOMON *et al.*, 2022), entre outros.

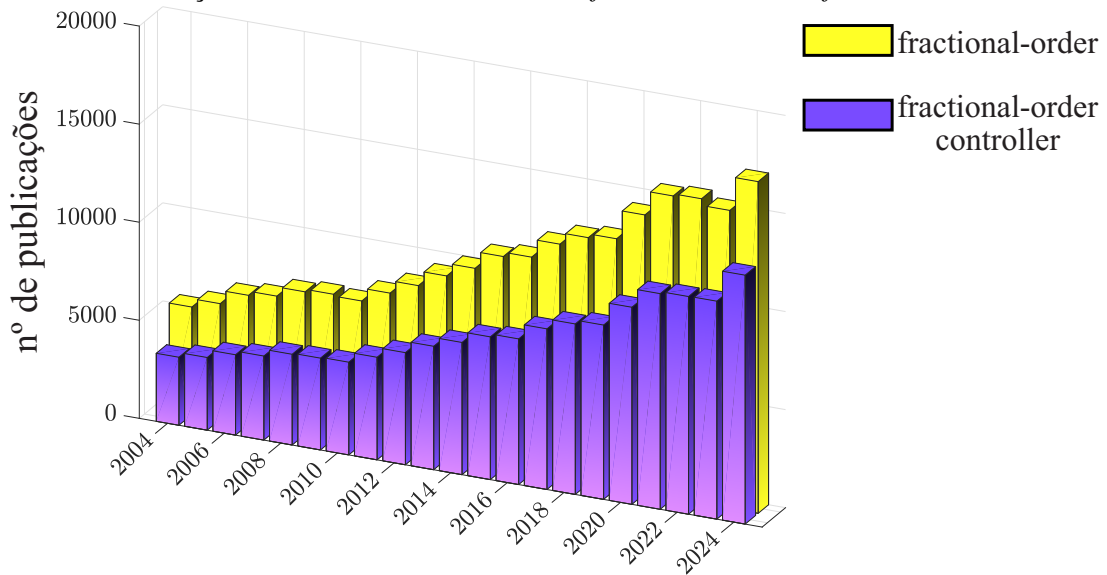
Atualmente, tem-se observado um número crescente de aplicações do cálculo fracionário em diversas aplicações em diversas áreas da ciência e da engenharia, tais como nanotecnologia (ABBAS *et al.*, 2024), medicina (UÇAR; ÖZDEMİR, 2023), economia (LIU *et al.*, 2024), aprendizado de máquinas (MA; YANG, 2024; HOA; PHU, 2024) e eletromagnetismo (YU *et al.*, 2023). No campo da engenharia, destaca-se o uso em aplicações em filtragem (DIMITRAKAKIS *et al.*, 2025) e sistemas de controle, nos quais é majoritariamente aplicado na modelagem de sistemas dinâmicos (AI *et al.*, 2024) e na síntese de controladores (TEPLJAKOV *et al.*, 2021).

Em decorrência do crescente interesse, a pesquisa sobre o tema tem atraído atenção da comunidade acadêmica, conforme o levantamento feito para esta tese na base de dados da Elsevier ilustrado na Figura 1, o qual revela a quantidade expressiva de aproximadamente 17.000 publicações com o termo *fractional order* somente no ano de 2024. Parte deste total diz respeito a controladores de ordem fracionária, tema este que acompanha a tendência de crescimento do número de publicações, com cerca de 13.000 no mesmo ano.

Devido à simplicidade de projeto e implementação do controlador PID (proporcional-integrativo-derivativo) tradicional, muitas linhas de pesquisa buscam aprimorar o desempenho desse controlador por meio de novas estruturas e métodos de sintonia (MERRIKH-BAYAT *et al.*, 2015). Uma dessas iniciativas, proposta por Podlubny (1994a), resultou em uma nova geração de controladores PID, conhecidos como controladores PID de ordem fracionária, ou $PI^\lambda D^\mu$, cuja função de transferência no domínio s é dado por $C(s) = k_p + \frac{k_i}{s^\lambda} + k_d s^\mu$, em que λ e μ são as ordens fracionárias de integração e derivação, respectivamente. Neste contexto, o controlador PID tradicional pode ser considerado um caso particular do controlador $PI^\lambda D^\mu$, em que $\lambda = \mu = 1$.

Devido aos valores fracionários das ordens de diferenciação e integração desses controladores, que em tese oferecem graus adicionais de liberdade ao controlador, seu desempenho pode ser potencialmente aprimorado (SHAH; AGASHE, 2016; DASTJERDI *et al.*, 2019). Como

Figura 1 – Publicações da Elsevier com os termos *fractional-order* e *fractional-order controller*.



Fonte: elaborada pelo autor.

consequência, controladores de ordem fracionária frequentemente alcançam desempenho superior em comparação aos controladores de ordem inteira (JAMIL *et al.*, 2022). Como exemplos de superioridade no desempenho, pode-se citar o clássico problema do robô manipulador com braço flexível (FELIU-TALEGON; FELIU-BATLLE, 2022) e o controle de temperatura (JAMIL *et al.*, 2022). Além disso, a superioridade do controlador $PI^\lambda D^\mu$ em relação ao PID convencional é evidenciada na literatura frente a diferentes desafios: em Kumar e Rana (2017) destaca-se seu melhor desempenho sob diversos tipos de ruído aleatório, em Liu *et al.* (2018) observa-se maior robustez frente a variações paramétricas, e em Jamil *et al.* (2022) destaca-se sua eficácia diante de perturbações externas.

1.1 Contextualização e justificativa

Apesar dos aspectos positivos mencionados, um desafio significativo na aplicação de sistemas de controle de ordem fracionária é a determinação de sua estabilidade, cuja análise não é tão trivial quanto em sistemas de ordem inteira. Dentre os métodos existentes, destacam-se o uso de LMIs (*linear matrix inequalities*) (LIN *et al.*, 2024) e o teorema de estabilidade de Matignon (MATIGNON, 1996). Para sistemas fracionários SISO (*single input single output*) representados por funções de transferência, o teorema de estabilidade de Matignon é amplamente utilizado, o qual envolve uma transformação de variáveis para converter o polinômio característico de ordem fracionária em um polinômio de ordem inteira. Contudo, este método não fornece técnicas analíticas para a solução da equação característica e não estabelece relações de estabilidade a

priori entre os parâmetros da planta e do controlador.

Estabelecer relações analíticas de estabilidade para sistemas de controle de ordem fracionária é altamente desejável do ponto de vista da síntese desses controladores. Tais relações poderiam ser utilizadas para orientar a otimização dos parâmetros de controladores fracionários em algoritmos de otimização usados em sintonia ótima dos parâmetros do controlador. De modo geral, essas técnicas baseiam-se na resposta ao degrau do sistema de controle para guiar o processo de ajuste dos parâmetros. Para isso, são necessárias simulações computacionais da resposta ao degrau, executadas inúmeras vezes ao longo da otimização.

Entretanto, entre os métodos atualmente utilizados para a simulação computacional de sistemas de ordem fracionária, alguns empregam aproximações por modelos de ordem inteira ([OUSTALOUP et al., 2000](#)), enquanto outros se baseiam diretamente nas definições de derivada fracionária ([XUE, 2017](#)). No entanto, estes métodos apresentam limitações, como baixa precisão ou tempo de simulação elevado, o que pode comprometer o processo de síntese de controladores fracionários por meio de algoritmos de otimização.

Devido à complexidade inerente ao cálculo fracionário, a implementação prática destes controladores ainda enfrenta desafios, especialmente porque os operadores de derivada e integral de ordem fracionária podem ser interpretados como filtros lineares de dimensão infinita ([PODLUBNY, 2001](#)). Técnicas de aproximação têm sido propostas para contornar tal dificuldade, como a apresentada em [Merrikh-Bayat et al. \(2015\)](#), que discretiza o controlador fracionário por meio de uma expansão por série de potências. Embora eficazes em diversos cenários, essas abordagens podem introduzir discrepâncias durante o regime transitório, afetando, por exemplo, o valor do máximo sobressinal e da frequência amortecida.

Por fim, a identificação da planta a ser controlada é uma etapa fundamental na síntese de controladores, uma vez que fornece o modelo matemático representativo do comportamento dinâmico do sistema ([DAS, 2011](#)). Métodos específicos para a identificação de sistemas fracionários como o apresentado em [Poinot e Trigeassou \(2004\)](#), permitem estimar tanto os coeficientes quanto os expoentes fracionários do modelo. Porém, os operadores fracionários identificados no modelo são aproximados por uma série de filtros de ordem inteira, o que recai na baixa precisão mencionada. Além disso, a ordem fracionária do modelo não é estimada de modo direto, mas inferida indiretamente a partir de parâmetros dos filtros utilizados na aproximação dos operadores fracionários. Todas estas condições resultam na não preservação de uma característica fundamental dos operadores fracionários, que é a memória infinita.

Diante dos pontos abordados, propõem-se novas técnicas para a realização da síntese de controladores de ordem fracionária, contemplando todas as etapas fundamentais do projeto de sistemas de controle fracionário: identificação da planta, simulação computacional, análise de estabilidade, síntese e implementação embarcada dos controladores. O objetivo é superar as limitações dos métodos atuais, promovendo maior precisão e eficiência em cada uma dessas etapas.

Em primeiro lugar, introduz-se um novo paradigma para resolução de equações características trinômiais generalizadas de ordem fracionária, fundamentado na recente função de Lambert-Tsallis ([Silva; RAMOS, 2019](#)), uma generalização da função Lambert. Além de proporcionar soluções analíticas para essas equações, a abordagem proposta também oferece um discriminante analítico e generalizado da estabilidade do sistema, levando em consideração todos os parâmetros da planta e do controlador. Isso permite ao projetista explorar os parâmetros do controlador de maneira a atender às condições de estabilidade. Até onde se tem conhecimento, nenhum outro método atualmente oferece tal possibilidade.

Posteriormente, estas condições analíticas de estabilidade são incorporadas a uma técnica de otimização metaheurística de modo a restringir a busca a regiões do espaço de parâmetros que garantam a estabilidade do sistema, evitando buscas desnecessárias em regiões de instabilidade.

As simulações computacionais necessárias para obter a resposta ao degrau, utilizadas na otimização metaheurística, são realizadas a partir da técnica SISFRAC (sigla para *simulação computacional de sistemas de ordem fracionária*), proposta que visa garantir a exatidão matemática da definição de derivada fracionária, ao mesmo tempo em que busca reduzir o tempo de processamento.

Para além da simulação computacional, propõe-se ainda uma implementação embarcada para os controladores fracionários através de um modelo computacional capaz de reproduzir o seu comportamento dinâmico; no caso, a rede MLP (*multilayer Perceptron*). Esta abordagem busca mitigar as distorções introduzidas por técnicas de discretização atualmente utilizadas, como a apresentada em [Merrikh-Bayat et al. \(2015\)](#).

Por fim, mas não menos importante, desenvolve-se uma técnica de identificação de sistemas fracionários capaz de estimar de forma direta os coeficientes e a ordem da planta utilizando a regra LMS (*least mean square*) em conjunto com a definição rigorosa dos operadores de derivada e integral fracionária.

1.2 Objetivos

Para tratar dos problemas mencionados na seção anterior no escopo da síntese de controladores fracionários, são definidos os seguintes objetivos geral e específicos a seguir.

1.2.1 *Objetivo geral*

Estabelecer um novo arcabouço matemático-computacional voltado para a síntese eficiente de controladores PID de ordem fracionária.

1.2.2 *Objetivos específicos*

Para alcançar o objetivo geral citado, estabeleceram-se os seguintes objetivos específicos:

- Desenvolver um método analítico de estabilidade para sistemas de controle de ordem fracionária, baseado na função de Lambert-Tsallis, para uma classe de sistemas cuja equação característica é dada por um trinômio.
- Propor uma técnica de simulação computacional eficiente que combine precisão matemática e baixo tempo de processamento.
- Apresentar um método de identificação recursiva de sistemas fracionários utilizando a regra LMS e operadores fracionários precisos.
- Realizar a síntese de controladores fracionários estáveis através da integração de técnicas de otimização metaheurísticas e condições analíticas de estabilidade.
- Implementar controladores fracionários em sistemas embarcados usando redes neurais para reproduzir o comportamento dinâmico do controlador.

1.3 Produção acadêmica

Durante a realização da tese, foram publicados os seguintes trabalhos:

- SOUSA, I.R.; NÓBREGA, K. Z. BARRETO, G. A. (2024) **An analytical approach to the stability conditions of a new class of fractional-order control systems by the Lambert-Tsallis function**. IEEE Access, v. 1, p. 1-1, 2024. DOI: 10.1109/ACCESS.2024.3503562. Disponível em: <https://ieeexplore.ieee.org/abstract/document/10759801>.
- SOUSA, I.R.; BARRETO, G. A.; NOGUEIRA, F. G. (2024) **Algoritmo Recursive Least**

M-estimate para identificação robusta de sistemas ARX LPV com ruído impulsivo. XXV Congresso Brasileiro de Automática, (CBA 2024), 15/10 a 18/10/24, Rio de Janeiro - Rio de Janeiro.

Disponível em: https://www.sba.org.br/cba2024/papers/paper_2761.pdf.

- SOUSA, I.R.; BARRETO, G. A. (2024) **Neural fractional order PID controller embedded in an 8-bit microcontroller for neonatal incubator.** International Joint Conference on Neural Networks (IJCNN 2024), presente no IEEE World Congress on Computational Intelligence (IEEE WCCI 2024), 30/06 a 05/07/24, Yokohama - Japão. DOI: 10.1109/IJCNN60899.2024.10651507.
Disponível em: <https://ieeexplore.ieee.org/abstract/document/10651507>.
- SOUSA, I.R.; BARRETO, G. A. (2023) **Identificação ARX LPV robusta a outliers via Estimador-M.** XVI Simpósio Brasileiro de Automação Inteligente (SBAI 2023), 15/10 a 18/10/23, Manaus - Amazonas. DOI: <https://doi.org/10.20906/SBAI-SBSE-2023/3909>.
Disponível em: https://www.sba.org.br/open_journal_systems/index.php/sbai/article/view/3909.
- SOUSA, I.R.; BARRETO, G. A. (2023) **Desenvolvimento de um controlador $PI^{\lambda}D^{\mu}$ discreto otimizado e embarcado em incubadora neonatal.** XVI Congresso Brasileiro de Inteligência Computacional (CBIC2023), 8/10 a 11/10/23, Salvador-Bahia. DOI: 10.21528/CBIC2023-049.
Premiado como o Melhor artigo na área Evolutionary, Swarm, and Nature-Inspired Algorithms.
Disponível em: https://sbic.org.br/wp-content/uploads/2023/10/pdf/CBIC_2023_paper049.pdf
- SOUSA, I.R.; MEDEIROS, C. M. S.; BARRETO, G. A. (2023) **Fuzzy control of a high-performance boost converter in discontinuous conduction mode and its application to a photovoltaic pumping system.** Learning & Nonlinear Models, 21(1), 47–59. DOI: 10.21528/lnlm-vol21-no1-art4.
Disponível em: <https://sbic.org.br/lnlm/publicacoes/vol21-no1/vol21-no1-art4/>
- SOUSA, I.R.; VASCONCELOS, F. J. S.; NOGUEIRA, F. G.; BARRETO, G. A. (2022) **Identificação e Controle LPV de uma Incubadora Neonatal.** XXIV Congresso Brasileiro de Automática (CBA 2022), 16/10 a 19/10/22, Fortaleza - Ceará.
DOI: <https://doi.org/10.20906/CBA2022/3751>.
Disponível em: https://www.sba.org.br/open_journal_systems/index.php/cba/article/view/3751
- SOUSA, I.R.; FIGUEIREDO, J. C.; BARRETO, G. A. (2021) **Revisitando o Uso da**

Transformada de Fourier no Reconhecimento de Voz para Robótica Móvel. XV Congresso Brasileiro de Inteligência Computacional (CBIC2021), 03/10 a 06/10/21, Joinville - Santa Catarina. DOI: 10.21528/CBIC2021-40.

Disponível em: https://sbic.org.br/wp-content/uploads/2021/09/pdf/CBIC_2021_paper_40.pdf

O seguinte trabalho foi aceito para publicação:

- SOUSA, I.R.; BARRETO, G. A. SOUZA FILHO, F. A. (2025) **Modelo autoregressivo LPV federado para previsão de vazão em reservatórios.** XVII Simpósio Brasileiro de Automação Inteligente (SBAI 2025), 29/07 a 01/08/25, São João del Rei - Minas Gerais.

Os seguintes trabalhos foram submetidos e estão aguardando análise dos pares:

- SOUSA, I.R.; NÓBREGA, K. Z. BARRETO, G. A. (2025) **Uso da Função de Lambert-Tsallis, $W_q(z)$, em sistemas de controle de ordem fracionária.** XLIV Congresso Nacional de Matemática Aplicada e Computacional (CNMAC 2025), 15/09 a 19/09/25, Rio de Janeiro, Rio de Janeiro (submetido).
- G. A. BARRETO, I. R. SOUSA, P. W. OLIVEIRA, G. A. P. THÉ (2025). **On the Synthesis of Self-Configurable Fractional Order PID Controllers,** Soft Computing (submetido).
- G. A. BARRETO, O. MENDES, I. R. SOUSA, M. MORALES (2025). **Optimal Tuning of PID and FOPID Controllers for Aircraft Models Using Swarm Intelligence,** Swarm and Evolutionary Computation (submetido).

1.4 Organização da tese

O restante da tese está organizada da seguinte forma. O [Capítulo 2](#) apresenta os conceitos que envolvem a aplicação do cálculo fracionário em sistemas de controle, como definições, plantas e controladores fracionários, e sua estabilidade. No [Capítulo 3](#), é apresentada a função Lambert-Tsallis, sua análise e propriedades, enquanto que no [Capítulo 4](#), desenvolve-se a proposta de análise de estabilidade de sistemas de ordem fracionária e um método analítico para

calcular os limites dos parâmetros do controlador que garantam a estabilidade. No [Capítulo 5](#), é proposta uma técnica de simulação computacional de sistemas de ordem fracionária, enquanto que no [Capítulo 6](#), apresenta-se uma proposta de algoritmo para identificação de sistemas de ordem fracionária. O [Capítulo 7](#) apresenta a síntese de controladores de ordem fracionária, composta da sintonia e da implementação embarcada. Por fim, as propostas apresentadas nesta tese são aplicadas a um estudo de caso no [Capítulo 8](#): o controle de temperatura de uma incubadora neonatal. As conclusões e sugestões para trabalhos futuros são apresentados no [Capítulo 9](#).

2 DERIVADAS E INTEGRAIS FRACIONÁRIAS EM SISTEMAS DE CONTROLE

Neste capítulo, são apresentados os conceitos que envolvem a aplicação do cálculo fracionário em sistemas de controle, como as definições de derivadas e integrais fracionárias na [Seção 2.1](#), exemplos de controladores na [Seção 2.2](#) e de sistemas físicos de ordem fracionária na [Seção 2.3](#), sistemas de controle de ordem fracionária em malha fechada na [Seção 2.4](#) e sua estabilidade na [Seção 2.5](#).

2.1 Definições de derivada e integral fracionárias

Pode-se afirmar com relativa certeza que o início do cálculo fracionário remonta ao ano de 1695, quando em uma troca de cartas com o Marquês de l'Hôpital, [Leibniz \(1695\)](#) versa sobre o significado de $n = \frac{1}{2}$ em sua notação $\frac{d^n y(x)}{dx^n}$ escrevendo que, em tradução livre do francês: “Segue que $d^{\frac{1}{2}}$ será igual a $x\sqrt{dx} : x$. Este é aparentemente um paradoxo do qual, um dia, serão extraídas consequências úteis, pois raramente há paradoxos sem utilidade”¹. Como mencionado no [Capítulo 1](#) e também visto ao longo do restante desta tese, diversas são as utilidades e aplicações do conceito de derivada fracionária.

A partir dessa troca de cartas até os dias atuais, passando por [Euler \(1738\)](#), [Lagrange \(1772\)](#), [Laplace \(1812\)](#), [Fourier \(1822\)](#), [Liouville \(1832\)](#), [Grünwald \(1867\)](#), [Letnikov \(1868\)](#), [Riemann \(1876\)](#), [Heaviside \(1893\)](#), [Caputo \(1967\)](#), e muitos outros, várias contribuições a respeito do tema foram publicadas. Nesta tese, apenas as definições de Riemann–Liouville (RL), Caputo e Grünwald-Letnikov (GL) são abordadas. Porém, mais informações sobre as contribuições e os desenvolvimentos das definições de derivada e integral de ordem fracionária podem ser encontradas em [Ross \(1977\)](#), [Miller e Ross \(1993\)](#), [Samko et al. \(1993\)](#), [Oldham e Spanier \(2006\)](#), [Valério et al. \(2014\)](#) e [Teodoro et al. \(2019\)](#), incluindo os acontecimentos ano a ano de cada contribuição. Nesta tese, será utilizada a notação ${}_t \mathcal{D}_t^\alpha f(t)$ para derivada fracionária de ordem $\alpha \in \mathbb{R}$ da função $f(t)$ proposta por [Davis \(1936\)](#), em que t_0 e t são os dois limites da operação.

A unificação das definições de Riemann e Liouville foi inicialmente proposta em [Laurent \(1884\)](#), passando então a ser denominada de Riemann–Liouville, e sendo atualmente

¹Posteriormente, [Lacroix \(1819\)](#), utilizando a função Gama em seu estudo sobre potência de funções, encontrou que $\frac{d^{1/2}x}{dx^{1/2}} = 2\sqrt{\frac{x}{\pi}}$, correspondendo ao mesmo valor obtido com as definições matemáticas de derivada de ordem fracionária utilizadas nos dias atuais.

expressa como

$${}^{RL}_{t_0} \mathcal{D}_t^\alpha f(t) = \frac{1}{\Gamma(n-\alpha)} \frac{d^n}{dt^n} \int_{t_0}^t \frac{f(\tau)}{(t-\tau)^{\alpha-n+1}} d\tau, \quad (2.1)$$

em que $(n-1) \leq \alpha < n \mid \alpha \in \mathbb{R}, n \in \mathbb{Z}$, e $\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt$ para $\text{Re}(z) > 0$ é a função Gama, uma extensão do conceito de fatorial para números reais que, no caso inteiro positivo, se reduz a $\Gamma(n) = (n-1)!$ (MONJE *et al.*, 2010). Esta é a definição mais conhecida, sendo mais utilizada na área da matemática pura, devido à sua simplicidade em relação às condições sobre a função $f(t)$ (PODLUBNY, 1998).

A definição de Caputo, também conhecida como Caputo-Dzhrbashjan, é utilizada por proporcionar interpretações físicas para as condições iniciais (VALÉRIO *et al.*, 2014), sendo expressa como

$${}^C_{t_0} \mathcal{D}_t^\alpha f(t) = \frac{1}{\Gamma(n-\alpha)} \int_{t_0}^t \frac{\frac{d^n}{d\tau^n} f(\tau)}{(t-\tau)^{\alpha-n+1}} d\tau, \quad (2.2)$$

em que $(n-1) \leq \alpha < n \mid \alpha \in \mathbb{R}, n \in \mathbb{Z}$ (MONJE *et al.*, 2010). É importante salientar que as definições de Caputo e Riemann–Liouville não são compatíveis entre si. Uma grande diferença entre estas definições é que a derivada de Caputo de uma constante C é 0, enquanto que a de Riemann–Liouville é $\frac{Ct^{-\alpha}}{\Gamma(1-\alpha)}$ (PODLUBNY, 1998).

Por fim, as definições de Grünwald e Letnikov foram unificadas por Liouville (1832). Atualmente denominada Grünwald-Letnikov, esta definição é comumente utilizada para simulação computacional devido à sua formulação em somatório, expressa como (XUE; BAI, 2024)

$${}^{GL}_0 \mathcal{D}_t^\alpha f(t) = \lim_{h \rightarrow 0} \frac{1}{h^\alpha} \sum_{i=0}^{\infty} (-1)^i \binom{\alpha}{i} f(t - ih), \quad (2.3)$$

em que a expressão contendo a expansão binomial é dada por

$$w_i = (-1)^i \binom{\alpha}{i} = (-1)^i \frac{\Gamma(\alpha+1)}{\Gamma(i+1)\Gamma(\alpha-i+1)}. \quad (2.4)$$

Do ponto de vista numérico, a definição de Grünwald-Letnikov é equivalente à de Riemann–Liouville, e é a primeira unificação de um operador integro-diferencial (PODLUBNY, 1998), dado por

$${}_{t_0}\mathcal{D}_t^\alpha f(t) = \begin{cases} \frac{d^\alpha}{dt^\alpha} f(t), & \alpha > 0 \\ f(t), & \alpha = 0 \\ \int_{t_0}^t f(\tau) d\tau^{-\alpha}, & \alpha < 0 \end{cases}. \quad (2.5)$$

Por ser um operador integro-diferencial, o valor de α determina se a operação é uma derivada de ordem fracionária ($\alpha > 0$) ou uma integral de ordem fracionária ($\alpha < 0$). De acordo com [Xue \(2017\)](#), se o passo de cálculo h for pequeno o suficiente, a definição de Grünwald-Letnikov pode ser aproximada por

$${}_0\mathcal{D}_t^\alpha f(t) \approx \frac{1}{h^\alpha} \sum_{i=0}^{\infty} (-1)^i \binom{\alpha}{i} f(t - ih), \quad (2.6)$$

mais usualmente utilizada para simulações computacionais. A função Gama, que aparece na [Equação 2.4](#), pode ser evitada ao utilizar um algoritmo recursivo para calcular os coeficientes w_i ,

$$w_i = \left(1 - \frac{\alpha + 1}{i}\right) w_{i-1}, \quad w_0 := 1. \quad (2.7)$$

2.1.1 Demonstração da definição de Grünwald-Letnikov

Originalmente, a abordagem tomada para se chegar na definição de Grünwald-Letnikov foi baseada na expressão ([SAMKO et al., 1993](#))

$$\mathcal{D}^\alpha f(t) = \lim_{h \rightarrow 0} \frac{\Delta_h^\alpha f(x)}{h^\alpha}. \quad (2.8)$$

Porém, nesta seção, será demonstrado que pode-se chegar no mesmo resultado por outro caminho. Isto posto, uma forma alternativa de se obter a definição de derivada fracionária de Grünwald-Letnikov é através do teorema binomial, ou seja

$$(x + y)^n = \sum_{i=0}^n \binom{n}{i} x^{n-i} y^i, \quad (2.9)$$

$$\binom{n}{i} = \frac{n!}{i!(n-i)!} \quad | \quad n, i \in \mathbb{Z}^+, \quad (2.10)$$

e do limite da discretização de Euler *a priori*, dado por

$$s = \lim_{T \rightarrow 0} \frac{1 - z^{-1}}{T}. \quad (2.11)$$

A partir da [Equação 2.11](#),

$$s^\alpha = \left(\lim_{T \rightarrow 0} \frac{1 - z^{-1}}{T} \right)^\alpha = \lim_{T \rightarrow 0} \frac{(1 - z^{-1})^\alpha}{T^\alpha}. \quad (2.12)$$

Nota-se que não é possível aplicar o teorema binomial no numerador da [Equação 2.12](#), uma vez que, por definição, $\alpha \in \mathbb{R}$. Faz-se necessário utilizar o teorema binomial generalizado e o coeficiente binomial generalizado, descritos respectivamente como

$$(x + y)^\alpha = \sum_{i=0}^{\infty} \binom{\alpha}{i} x^{\alpha-i} y^i, \quad (2.13)$$

$$\binom{\alpha}{i} = \frac{\Gamma(\alpha + 1)}{\Gamma(i + 1)\Gamma(\alpha - i + 1)} \quad | \quad \alpha \in \mathbb{R}, i \in \mathbb{Z}^+. \quad (2.14)$$

Fazendo $x = 1$ e $y = -z^{-1}$ na [Equação 2.13](#), a [Equação 2.12](#) pode ser desenvolvida como

$$s^\alpha = \lim_{T \rightarrow 0} \frac{(1 - z^{-1})^\alpha}{T^\alpha} = \lim_{T \rightarrow 0} \frac{1}{T^\alpha} \sum_{i=0}^{\infty} \frac{\Gamma(\alpha + 1)}{\Gamma(i + 1)\Gamma(\alpha - i + 1)} 1^{\alpha-i} (-z^{-1})^i \quad (2.15)$$

$$s^\alpha = \lim_{T \rightarrow 0} \frac{1}{T^\alpha} \sum_{i=0}^{\infty} \frac{\Gamma(\alpha + 1)}{\Gamma(i + 1)\Gamma(\alpha - i + 1)} 1^{\alpha-i} (-1)^i (z^{-1})^i \quad (2.16)$$

$$s^\alpha = \lim_{T \rightarrow 0} \frac{1}{T^\alpha} \sum_{i=0}^{\infty} \frac{\Gamma(\alpha + 1)}{\Gamma(i + 1)\Gamma(\alpha - i + 1)} (-1)^i z^{-i} \quad (2.17)$$

$$s^\alpha = \lim_{T \rightarrow 0} \frac{1}{T^\alpha} \sum_{i=0}^{\infty} w_i z^{-i}. \quad (2.18)$$

Observa-se que a [Equação 2.18](#) é uma forma compacta da definição original da definição de Grünwald-Letnikov de derivada fracionária, presente na [Equação 2.3](#). De fato,

$$s^\alpha F(s) = \lim_{T \rightarrow 0} \frac{1}{T^\alpha} \sum_{i=0}^{\infty} w_i z^{-i} F(z) \iff {}_0^{GL} \mathcal{D}_t^\alpha f(t) = \lim_{T \rightarrow 0} \frac{1}{T^\alpha} \sum_{i=0}^{\infty} w_i f(t - Th). \quad (2.19)$$

Como mencionado, o coeficiente w_i pode ser obtido de forma recursiva. O desenvolvimento de tal expressão é apresentado a seguir. Utilizando a definição de w_i apresentada na [Equação 2.4](#),

$$\frac{w_i}{w_{i-1}} = \frac{\frac{\Gamma(\alpha+1)}{\Gamma(i+1)\Gamma(\alpha-i+1)}(-1)^i}{\frac{\Gamma(\alpha+1)}{\Gamma(i-1+1)\Gamma(\alpha-i+1+1)}(-1)^{i-1}} = \frac{\frac{\Gamma(\alpha+1)}{\Gamma(i+1)\Gamma(\alpha-i+1)}(-1)}{\frac{\Gamma(\alpha+1)}{\Gamma(i)\Gamma(\alpha-i+2)}} \quad (2.20)$$

$$\frac{w_i}{w_{i-1}} = \frac{\cancel{\Gamma(\alpha+1)}}{\Gamma(i+1)\Gamma(\alpha-i+1)} \frac{\Gamma(i)\Gamma(\alpha-i+2)}{\cancel{\Gamma(\alpha+1)}}(-1) = \frac{-\Gamma(i)\Gamma(\alpha-i+2)}{\Gamma(i+1)\Gamma(\alpha-i+1)}. \quad (2.21)$$

Aplicando a propriedade $\Gamma(z+1) = z\Gamma(z)$, logo $\Gamma(\alpha-i+2) = (\alpha-i+1)\Gamma(\alpha-i+1)$. Assim,

$$\frac{w_i}{w_{i-1}} = \frac{\cancel{-\Gamma(i)}(\alpha-i+1)\cancel{\Gamma(\alpha-i+1)}}{i\cancel{\Gamma(i)}\cancel{\Gamma(\alpha-i+1)}} = \frac{-(\alpha-i+1)}{i} = \frac{i-\alpha-1}{i} \quad (2.22)$$

$$\frac{w_i}{w_{i-1}} = 1 - \frac{\alpha+1}{i} \quad (2.23)$$

$$w_i = \left(1 - \frac{\alpha+1}{i}\right) w_{i-1}, \quad w_0 := 1. \quad (2.24)$$

2.2 Controladores de ordem fracionária

No início dos anos de 1960, o estudo de cálculo fracionário começou a ser expandido para os campos da ciência e engenharia. Em controle, [Manabe \(1960\)](#) estendeu o conceito de *ordem não inteira*, introduzindo os sistemas de controle de ordem não inteira, o que culminou na

proposta de Podlubny (1994a) para o controlador PID de ordem fracionária, também denotado por controlador $PI^\lambda D^\mu$, cuja função de transferência é definida como

$$G_c(s) = \frac{U(s)}{E(s)} = k_p + \frac{k_i}{s^\lambda} + k_d s^\mu, \quad (2.25)$$

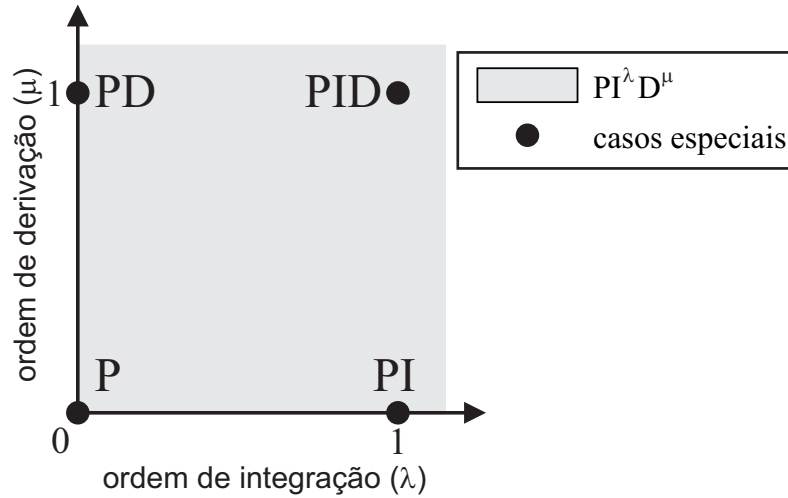
em que λ e μ são as ordens fracionárias de integração e derivação, respectivamente, enquanto que $U(s)$ e $E(s)$ denotam a saída do controlador e o sinal de erro do sistema de controle, respectivamente.

Diferentemente do PID convencional, que exige a introdução de um filtro no termo derivativo para evitar a amplificação de ruídos de alta frequência, o controlador $PI^\lambda D^\mu$, ao empregar uma ordem efetivamente inferior a 1 na derivada fracionária, possui uma capacidade intrínseca de suprimir ruídos (KUMAR; RANA, 2017). Essa característica suavizada do termo derivativo torna desnecessária, em muitos casos, a aplicação de filtros adicionais, mantendo a estrutura do controlador mais fiel à formulação teórica e simplificando sua implementação.

A partir da Equação 2.25, nota-se que os controladores PID convencionais são casos particulares do controlador $PI^\lambda D^\mu$, em que $\lambda = \mu = 1$. Assim, além dos 3 parâmetros usuais do controlador PID, representados pelos ganhos k_p , k_i , k_d , acrescentam-se 2 outros parâmetros a serem sintonizados, a saber, a ordem de integração λ e a ordem de derivação μ , conferindo graus de liberdade adicionais ao modelo do controlador, o que pode potencialmente melhorar o desempenho de controle de sistemas dinâmicos complexos (SHAH; AGASHE, 2016). Na Figura 2, está a gama de representações do controlador $PI^\lambda D^\mu$. Observa-se que os pontos em destaque são casos especiais em que o controlador PID pode atuar, seja como controlador P, PI, PD ou PID. Por outro lado, o controlador $PI^\lambda D^\mu$ pode trabalhar em qualquer ponto de operação entre os pontos em destaque, caracterizando-se como um controlador com uma riqueza em termos de dinâmicas bem maior que a do clássico PID de ordem inteira.

Atualmente, os controladores $PI^\lambda D^\mu$ vêm sendo utilizados com sucesso em um número crescente de aplicações, uma vez que podem alcançar melhores desempenhos que os controladores PID clássicos (JAMIL *et al.*, 2022). Em controle de temperatura por exemplo, Feliu-Batlle *et al.* (2014) realizaram o controle de um forno de reaquecimento com um controlador $PI^\lambda D^\mu$, apresentando melhores resultados que o PID convencional. Petráš e Vinagre (2002), realizaram o controle de um bloco de aquecimento por controlador $PI^\lambda D^\mu$ e PID, em que o $PI^\lambda D^\mu$ também demonstrou melhor desempenho. Outro controlador bastante comum é o avanço-atraso de fase, cuja versão de ordem fracionária, utilizada por El-Khazali (2018), Safaei e Tavakoli

Figura 2 – Possibilidades de atuação do controlador $PI^\lambda D^\mu$.



Fonte: elaborada pelo autor.

(2018) e [Kapoulea et al. \(2020\)](#), é definida como

$$G_c(s) = \frac{ds^\alpha + c}{s^\alpha + a}, \quad (2.26)$$

em que as constantes a , c , d são parâmetros do controlador e α é um expoente fracionário.

2.3 Plantas de ordem fracionária

Para além da questão do uso de derivadas/integrais fracionárias em controladores, cabe destacar que o cálculo fracionário também é usado para a construção de modelos de sistemas dinâmicos de um modo geral. Os operadores de derivada fracionária são operadores não locais, e esta propriedade permite modelar a dinâmica de muitos processos complexos nas ciências aplicadas e engenharia ([Vázquez et al., 2011](#)).

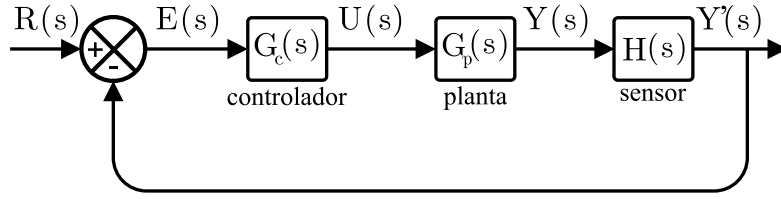
Existem diversos sistemas físicos representados por funções de transferência de ordem fracionária. Entre estes, destacam-se os circuitos elétricos com capacitores e indutores fractais, como o circuito RC^α em [Bošković et al. \(2017\)](#), a transferência de calor em [Poinot et al. \(2002\)](#) e [Petráš \(2011\)](#), e o carregamento de baterias de íon-lítio em [Solomon et al. \(2022\)](#), que podem ser representados pela função de transferência do tipo

$$G_p(s) = \frac{c}{s^\alpha + a}, \quad (2.27)$$

a reação cinética química em um reator batelada em [Qureshi e Aziz \(2020\)](#),

$$G_p(s) = \frac{s^\alpha}{s^\beta + as}, \quad (2.28)$$

Figura 3 – Típico sistema de controle em malha fechada.



Fonte: elaborada pelo autor.

e o modelo de pequeno sinais de um conversor eletrônico do tipo *boost* em [Jia e Liu \(2018\)](#),

$$G_p(s) = \frac{es^\alpha + c}{s^\beta + ds^\alpha + a}. \quad (2.29)$$

Além disso, destacam-se as seguintes aplicações: modelagem de materiais visco-elásticos ([HAN *et al.*, 2024](#)), a dinâmica de uma barra flexível ([BENFTIMA *et al.*, 2023](#)), o escoamento de um fluido não newtoniano ([SARWAR *et al.*, 2024](#)), a dinâmica da glicose e da insulina no corpo humano ([NISAR; FARMAN, 2025](#)), o modelo de um memristor ([FORTULAN *et al.*, 2024](#)) e a impedância do sistema respiratório humano ([WANG *et al.*, 2024](#)).

2.4 Sistemas de controle em malha fechada

Um sistema de controle em malha fechada típico de tempo contínuo SISO é ilustrado na [Figura 3](#), em que

$$G_c(s) = \frac{A(s)}{B(s)}, \quad G_p(s) = \frac{C(s)}{D(s)} \quad (2.30)$$

e $H(s)$ representam as funções de transferência do controlador, da planta e da malha de retroalimentação, respectivamente. Além disso, $R(s)$ representa o sinal de referência de controle, $E(s)$ é o erro entre a referência e a saída do sensor, $U(s)$ é a saída do controlador, $Y(s)$ é a saída da planta e $Y'(s)$ é o sinal medido pelo sensor. Assim, a função de transferência do sistema de controle em malha fechada $G_{cl}(s)$ é definida como

$$G_{cl}(s) = \frac{Y(s)}{R(s)} = \frac{G_c(s)G_p(s)}{1 + G_c(s)G_p(s)H(s)} = \frac{A(s)C(s)}{B(s)D(s) + A(s)C(s)H(s)}. \quad (2.31)$$

Na prática, a função de transferência na [Equação 2.31](#) para sistemas de ordem fracionária é expressa na forma de polinômios em s , como

$$G_{cl}(s) = \frac{P(s)}{Q(s)} = \frac{\sum_{i=0}^M a_i s^{\alpha_i}}{\sum_{j=0}^N b_j s^{\beta_j}}, \quad (2.32)$$

em que $a_i, b_j \in \mathbb{R}$, $\alpha_i, \beta_j \in \mathbb{R}_+$, e $Q(s) = \sum_{j=0}^N b_j s^{\beta_j}$ é a equação característica da função de transferência do sistema de controle em malha fechada. As raízes dos polinômios² $P(s) = 0$ e $Q(s) = 0$ estão relacionadas aos zeros e aos polos do sistema, respectivamente. A análise de polos é de fundamental importância para a estabilidade do sistema de controle, seja este fracionário ou não.

2.5 Estabilidade de sistemas de controle de ordem fracionária

Para analisar a estabilidade de um sistema de controle descrito pela [Equação 2.32](#), é necessário encontrar as raízes de $Q(s)$, ou seja, resolver a equação $Q(s) = 0$. Entretanto, quando a equação característica possui ordem fracionária, a solução não é trivial, uma vez que não existe solução analítica para resolver polinômios de ordem fracionária, exceto em raras exceções.

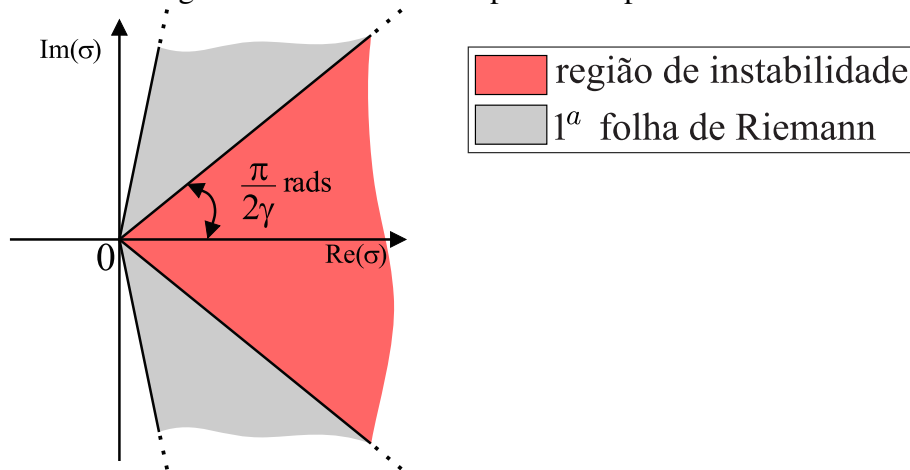
Dentre as exceções com solução analítica, destaca-se o trabalho de [Radwan et al. \(2009\)](#), em que um polinômio com três termos (trinômio) na forma $s^{2\alpha} + as^\alpha + b$ é convertido para um polinômio do segundo grau através de uma mudança de variável. Assim, a análise de estabilidade pode ser realizada utilizando a fórmula de Bhaskara nessa nova variável. Porém, é importante enfatizar que este método é aplicável apenas a um número bastante limitado de sistemas de controle, pois as restrições exigem que a equação característica trinomial inclua um termo com um expoente de ordem fracionária exatamente duas vezes o de outro termo, e não com coeficientes arbitrários. Nesta tese, é proposta no [Capítulo 4](#) uma metodologia pioneira para encontrar analiticamente os polos de um sistema de controle de ordem fracionária em que a função característica trinomial possui coeficientes e expoentes arbitrários.

Atualmente, a estabilidade de sistemas SISO de ordem fracionária, como o descrito na [Equação 2.32](#), é analisada principalmente através do teorema de [Matignon \(1996\)](#). Para aplicar tal teorema, realiza-se inicialmente a mudança de variável $\sigma = s^{\frac{1}{\gamma}}$, em que $\gamma \in \mathbb{N}^*$ representa o denominador comum dos expoentes fracionários de $Q(s)$. Com isso, a equação característica $Q(s)$ pode ser reescrita como

$$Q(\sigma) = \sum_{j=1}^N b_j \sigma^{\delta_j}, \quad (2.33)$$

²Por definição, *polinômios de ordem fracionária* não são de fato polinômios. Polinômios são entidades matemáticas com propriedades rigorosamente estabelecidas. Uma propriedade fundamental dos polinômios é a de possuir expoentes inteiros não negativos, o que os torna funções univariadas ([BARBEAU, 2013](#)). Assim, a nomenclatura correta seria a de *funções algébricas*. Porém, nesta tese, estas funções algébricas são chamadas de *polinômios fracionários* devido à vasta utilização do termo no contexto de ciência e engenharia. O mesmo vale para as expressões *trinômios* e *monômios* de ordem fracionária.

Figura 4 – Análise da região de instabilidade no plano complexo transformado $\sigma = s^{\frac{1}{\gamma}}$.



Fonte: elaborada pelo autor.

em que $\delta_j \in \mathbb{Z}_+$. Assim, este expediente matemático permite a criação de um novo polinômio, agora de ordem inteira, no plano σ . Dessa forma, o estudo da estabilidade pode ser realizado a partir da análise das raízes de $Q(\sigma)$. A seguir, apresenta-se o enunciado do Teorema de Matignon (MATIGNON, 1996):

Teorema 2.5.1: Primeiro Teorema de Matignon - Estabilidade de Sistemas Fracionários (1996)

Considere um sistema linear invariante no tempo SISO de ordem fracionária cuja função de transferência possui equação característica $Q(\sigma)$. O sistema é assintoticamente estável se, e somente se, todas as raízes σ_j do polinômio $Q(\sigma)$ satisfazem

$$|\arg(\sigma_j)| > \frac{\pi}{2\gamma}, \quad \forall j, \quad (2.34)$$

em que $|\arg(\sigma_j)|$ é ângulo absoluto, em radianos, entre o número complexo σ_j e o eixo real positivo. A região de instabilidade no plano σ é ilustrada na Figura 4. É interessante notar que para $\gamma = 1$, a região de instabilidade é a mesma para sistemas de ordem inteira. Porém, ressalta-se que o teorema de estabilidade de Matignon refere-se ao plano $s^{\frac{1}{\gamma}}$ e não ao plano s . A região de estabilidade no plano s continua sendo o semiplano esquerdo.

Embora a transformação $\sigma = s^{\frac{1}{\gamma}}$ auxilie na resolução do problema, sua utilização traz como consequência o surgimento de raízes falsas³ em $Q(s)$, uma vez que se trata de uma função multivalorada. Isso significa que para cada valor de s , têm-se γ soluções distintas em σ . A função raiz enésima, como no caso de $x = \sqrt[4]{1}$, é um exemplo de função multivalorada, em que a resposta completa é $x = \{+1, -1, +i, -i\}$. Além disso, o número de soluções de $Q(\sigma)$

³Os termos raízes *genuínas* e *falsas* são alternativas aos termos *feasible* e *unfeasible*, bastante utilizados na literatura estrangeira, e que podem ser traduzidos como *realizáveis* e *não realizáveis*. Os termos referem-se às raízes que estão dentro e fora da primeira folha de Riemann, respectivamente.

depende diretamente no valor de γ . Portanto, a análise das raízes de $Q(\sigma)$ e $Q(s)$ requer o uso de conceitos matemáticos mais adequados, como a superfície de Riemann, em que é possível visualizar uma função multivalorada de forma monovalorada através de uma estrutura geométrica compostas pelas folhas de Riemann conectadas (BROWN; CHURCHILL, 2015).

A função $\sigma = s^{\frac{1}{\gamma}}$ possui uma superfície de Riemann com γ folhas, ao longo das quais suas raízes se distribuem. Cada uma dessas folhas corresponde a um intervalo angular exclusivo no plano complexo. A extensão angular θ_{R_k} da k -ésima folha é dada por

$$\frac{-3\pi + 2\pi k}{\gamma} < \theta_{R_k} \leq \frac{-\pi + 2\pi k}{\gamma}. \quad (2.35)$$

A análise destas folhas é de crucial importância, uma vez que, de acordo com Gross e Braga (1961), apenas a primeira folha de Riemann tem significado físico em funções multivaloradas. Isso significa que as raízes que estão fora da primeira folha não possuem sentido físico e não influenciam na estabilidade ou instabilidade de um sistema de controle. Consequentemente, nem todas as soluções de $Q(\sigma)$ são raízes genuínas do problema original $Q(s)$. Analogamente à Equação 2.34, pode-se caracterizar as raízes de $Q(\sigma)$ que pertencem à primeira folha de Riemann por meio do segundo teorema de Matignon:

Teorema 2.5.2: Segundo Teorema de Matignon - Raízes da Primeira Folha de Riemann (1996)

As raízes de $Q(\sigma)$ que estão na primeira folha de Riemann são aquelas em que

$$|\arg(\sigma_j)| < \frac{\pi}{\gamma}, \quad \forall j. \quad (2.36)$$

A região da primeira folha de Riemann também é ilustrada na Figura 4. Os detalhes para a correta aplicação dos Teoremas 2.5.1 e 2.5.2 são explorados em três exemplos a seguir.

Exemplo 2.5.1: caso $Q(s) = s^2 + s^{1,5} - 1$

Seja um sistema de controle cuja equação característica é

$$Q(s) = s^2 + s^{1,5} - 1. \quad (2.37)$$

Existem infinitos valores de γ que podem ser utilizados, mas uma forma que reduz o número de raízes falsas é reescrever a equação com os termos fracionários representados por frações,

$$Q(s) = s^{\frac{2}{1}} + s^{\frac{3}{2}} - 1, \quad (2.38)$$

e em seguida escolher γ como o mínimo múltiplo comum dos denominadores dos expoentes fracionários. Logo, para este exemplo, $\gamma = 2$. Aplicando a transformação $\sigma = s^{\frac{1}{2}}$, tem-se que

$$Q(\sigma) = \sigma^4 + \sigma^3 - 1. \quad (2.39)$$

Assim, pode-se utilizar qualquer linguagem de programação que possua a função `roots()`⁴ para resolver $Q(\sigma)$. Nesta tese, utiliza-se o MATLAB para todas os cálculos computacionais e simulações. Com o auxílio da função `roots()`, as raízes de $Q(\sigma)$ são

$$\begin{aligned}\sigma_1 &= -1,380277 \\ \sigma_{2,3} &= -0,219447 \pm 0,914473i \\ \sigma_4 &= 0,819172.\end{aligned}\tag{2.40}$$

Para verificar se o sistema é estável, utiliza-se o teorema de Matignon na [Equação 2.34](#). Assim, os valores de $|\arg(\sigma_j)|, \forall j$ devem ser maiores que $\frac{\pi}{2\gamma} = 0,785398$. Os valores calculados são

$$\begin{aligned}|\arg(\sigma_1)| &= 3,141592 \\ |\arg(\sigma_{2,3})| &= 1,806314 \\ |\arg(\sigma_4)| &= 0.\end{aligned}\tag{2.41}$$

Nota-se que uma das quatro raízes de $Q(\sigma)$ não satisfaz a condição de estabilidade, o que significa que o sistema é instável. Para saber quais raízes estão na primeira folha, compara-se os valores obtidos com o valor referenciado na [Equação 2.36](#), ou seja, $\frac{\pi}{\gamma} = 1,570796$. Assim, conclui-se que o problema possui apenas uma raiz genuína⁵. Outra forma de obter as raízes genuínas de $Q(s)$ é aplicar as raízes s_j no polinômio característico. Para calcular as raízes s_j , aplica-se a transformação inversa, $s = \sigma^\gamma$, obtendo

$$\begin{aligned}s_1 &= 1,905166 \\ s_{2,3} &= -0,7881048 \mp 0,401358i \\ s_4 &= 0,671044.\end{aligned}\tag{2.42}$$

Por fim, aplicando estas soluções na função característica,

$$\begin{aligned}Q(s_1) &= 5,259316 \\ Q(s_{2,3}) &= -1,0799576 \pm 1,265248 \\ Q(s_4) &= 0.\end{aligned}\tag{2.43}$$

⁴A função `roots()` só pode ser utilizada em polinômios de ordem inteira.

⁵Deve-se atentar à expressão *genuína* aplicada às raízes σ . Aqui, σ_4 não é uma raiz genuína, pois não zera $Q(s)$. Todas os valores de σ zeram $Q(\sigma)$, mas não zeram $Q(s)$. Nesse contexto, σ_4 é chamada de genuína por estar dentro da primeira folha de Riemann. A transformação inversa de σ_4 , ou seja, $s_4 = \sigma_4^\gamma$ é a verdadeira raiz genuína de $Q(s)$.

Como previsto, apenas a solução 4 é genuína. É importante ressaltar que as raízes falsas não existem de fato. São criadas a partir da transformação $\sigma = s^{\frac{1}{\gamma}}$. Isso se reflete no fato de que, para diferentes valores de γ , são gerados diferentes polinômios $Q(\sigma)$, com diferentes números de raízes falsas, necessárias para respeitar o número de raízes do polinômio de ordem inteira. Porém, para todo e qualquer $Q(\sigma)$, o número de raízes genuínas, assim como seus valores, são constantes para qualquer valor de γ .

As localizações dos polos falsos e genuínos nos planos σ e s , assim como as regiões de instabilidade em cada plano e as folhas de Riemann são ilustradas na [Figura 5](#). É interessante notar que o trinômio $s^2 + s^{1.5} + 1$ possui apenas uma raiz genuína. Intuitivamente, o maior coeficiente (grau 2) induz o leitor a esperar duas raízes, sejam estas reais ou complexo-conjugadas. Porém, como pode ser notado na [Figura 5.b](#), a segunda raiz real do polinômio está na segunda folha de Riemann, tornando-se uma raiz não física (falsa). A interpretação dessas informações se torna mais clara com o auxílio do gráfico da superfície de Riemann, ilustrado na [Figura 6](#), complementando as informações da [Figura 5](#).

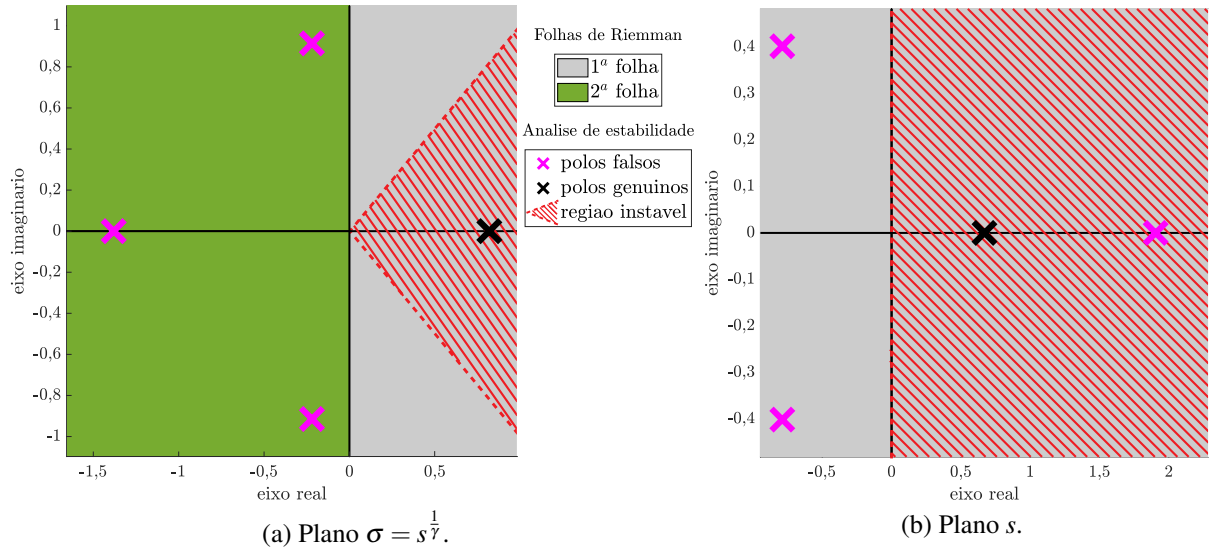
A superfície de Riemann é ilustrada em quatro dimensões, em que as coordenadas x , y e z correspondem respectivamente à parte real de s , à parte de imaginária de s e à parte real de σ . A quarta dimensão corresponde à parte imaginária de σ e é apresentada de forma cromática.

Para compreender como acontece a transformação $s^{\frac{1}{\gamma}}$, sugere-se um exercício mental que consiste em analisar o comportamento geométrico a partir de uma trajetória angular. Sejam θ_s e θ_σ os ângulos auxiliares deste exercício mental. Os dois ângulos são equivalentes, mas θ_s pertence ao plano s enquanto que θ_σ pertence ao plano σ . As duas folhas de Riemann no plano σ (plano superior à superfície de Riemann) compreendem os espaços de $-\frac{\pi}{2}$ a $\frac{\pi}{2}$ e de $\frac{\pi}{2}$ a $\frac{3\pi}{2}$, respectivamente. Enquanto isso, no plano s (plano inferior à superfície de Riemann), as duas folhas de Riemann compreendem os espaços de $-\pi$ a π e de π a 3π , respectivamente.

Iniciando a análise na primeira folha, os pontos em $\theta_\sigma = -\frac{\pi}{2}$ no plano σ equivalem aos pontos em $\theta_s = -\pi$ no plano s . É interessante notar que esta relação é de $1 : \gamma$. À medida que θ_σ cresce no sentido anti-horário, alcança-se a raiz genuína após percorrer $\frac{\pi}{2}$ rads, enquanto que no plano s essa distância percorrida equivale a π rads. Percorrendo mais $\frac{\pi}{2}$ rads no plano σ , tem-se $\theta_\sigma = \frac{\pi}{2}$, equivalente a $\theta_s = \pi$ rads, chegando ao fim da primeira folha de Riemann.

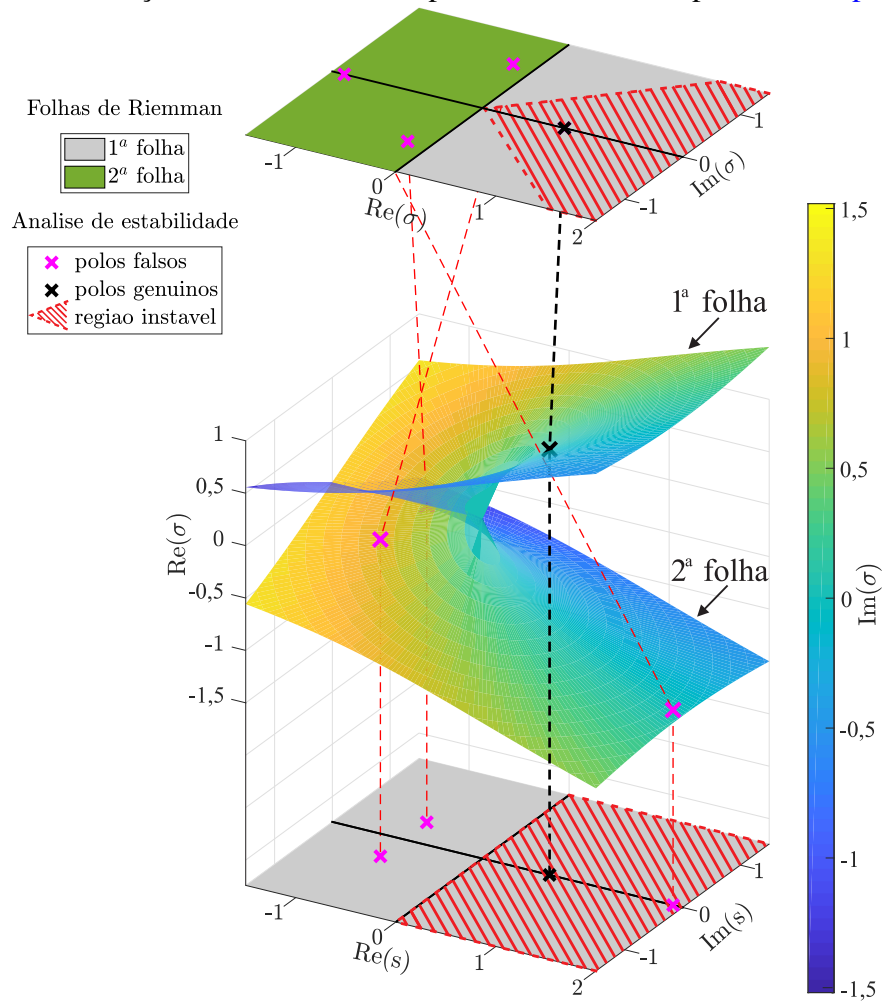
Neste momento, os pontos no plano σ e no plano s estão localizados na interseção entre a primeira e segunda folha da superfície de Riemann. Ao continuar incrementando θ_σ no sentido anti-horário, desliza-se para a segunda folha de Riemann (a folha inferior da

Figura 5 – Polos falsos e genuínos nos planos σ e s para o Exemplo 2.5.1.



Fonte: elaborada pelo autor.

Figura 6 – Transformação $\sigma = s^{\frac{1}{\gamma}}$ em uma superfície de Riemann para o Exemplo 2.5.1, $\gamma = 2$.



Fonte: elaborada pelo autor.

superfície), o que é feito de modo suave, como ilustrado. A primeira raiz falsa que se encontra é $\sigma = -0,219 + 0,914i$. Observa-se que esta raiz, que possui parte imaginária positiva no plano σ , passa a possuir parte imaginária negativa no plano s . Isso porque a segunda folha de Riemann no plano σ está defasada π rads da primeira folha (mais uma vez em razão de $\gamma = 2$).

Dando continuidade ao incremento angular no plano σ , encontra-se a raiz $\sigma = -1,38$ em $\theta_\sigma = \pi$. No plano s , estes valores equivalem a $s = 1,905$ em $\theta_s = 2\pi$. Percebe-se então que, desde que iniciou-se o exercício mental, foram percorridos $\frac{3\pi}{2}$ rads no plano σ , mas 3π rads no plano s , o que posiciona esta raiz no eixo real positivo do plano s . Nota-se que não faz sentido afirmar que esta raiz está na região de instabilidade do plano s , uma vez que ela pertence à segunda folha de Riemann, enquanto que a instabilidade está associada a uma região específica dentro da primeira folha. Por fim, com a continuação do incremento angular de θ_σ , passa-se pela raiz $\sigma = -0,219 - 0,914i$ e posteriormente, retorna-se à origem da primeira folha em $\theta_\sigma = \frac{3\pi}{2}$ e $\theta_s = 3\pi$. As duas folhas de Riemann se entrelaçam, de forma que a primeira se conecta à segunda e o fim da segunda folha se conecta com o início da primeira.

Exemplo 2.5.2: caso $Q(s) = s^2 + s^{0,1} + 1$

Seja um sistema de controle com equação característica

$$Q(s) = s^2 + s^{0,1} + 1. \quad (2.44)$$

Aplicando os mesmos passos do exemplo anterior, tem-se $\gamma = 10$, o que resulta em

$$Q(\sigma) = \sigma^{20} + \sigma + 1. \quad (2.45)$$

A diferença do polinômio deste exemplo para o exemplo anterior é apenas a ordem do segundo monômio. Porém, observa-se que há um total de 20 raízes, enquanto que no exemplo anterior, existem apenas quatro. Isso se dá pelo fato dos expoentes do polinômio anterior possuírem o fator 0,5 ($\gamma = 2$) em comum, ao passo que neste exemplo, o mínimo múltiplo comum é 0,1 ($\gamma = 10$). Para este caso, a condição de estabilidade é $|\arg(\sigma_j)|, \forall j$ serem maiores que $\frac{\pi}{2\gamma} = 0,157079$, enquanto que a condição para dada raiz estar na primeira folha de Riemann é $|\arg(\sigma_j)|$ ser menor que $\frac{\pi}{\gamma} = 0,314159$. Os valores de σ_j , $|\arg(\sigma_j)|$, $Q(\sigma_j)$, s_j e $Q(s_j)$ são exibidos na [Tabela 1](#) e na [Tabela 2](#).

Nota-se que, seguindo os critérios do teorema de [Matignon \(1996\)](#), todas as raízes estão fora da região de instabilidade, o que significa que o sistema é estável. Além disso, apenas as raízes $\sigma_{1,2}$ estão na primeira folha de Riemann. As localizações dos polos falsos e genuínos

Tabela 1 – Valores de σ_j e $|\arg(\sigma_j)|$ para o [Exemplo 2.5.2](#).

σ_j	$ \arg(\sigma_j) $
$\sigma_{1,2} = 1,0226 \pm 0,1663i$	$ \arg(\sigma_{1,2}) = 0,161180$
$\sigma_{3,4} = 0,9160 \pm 0,4810i$	$ \arg(\sigma_{3,4}) = 0,483537$
$\sigma_{5,6} = 0,7144 \pm 0,7443i$	$ \arg(\sigma_{5,6}) = 0,805878$
$\sigma_{7,8} = 0,4400 \pm 0,9283i$	$ \arg(\sigma_{7,8}) = 1,128187$
$\sigma_{9,10} = 0,1226 \pm 1,0135i$	$ \arg(\sigma_{9,10}) = 1,450435$
$\sigma_{11,12} = -0,2028 \pm 0,9916i$	$ \arg(\sigma_{11,12}) = 1,772558$
$\sigma_{13,14} = -0,5000 \pm 0,8660i$	$ \arg(\sigma_{13,14}) = 2,094395$
$\sigma_{15,16} = -0,7347 \pm 0,6525i$	$ \arg(\sigma_{15,16}) = 2,415427$
$\sigma_{17,18} = 0,8765 \pm 0,3793i$	$ \arg(\sigma_{17,18}) = 2,733159$
$\sigma_{19,20} = -0,9015 \pm 0,1052i$	$ \arg(\sigma_{19,20}) = 3,025439$

Fonte: elaborada pelo autor.

Tabela 2 – Valores das raízes e dos resíduos dos polinômios para o [Exemplo 2.5.2](#).

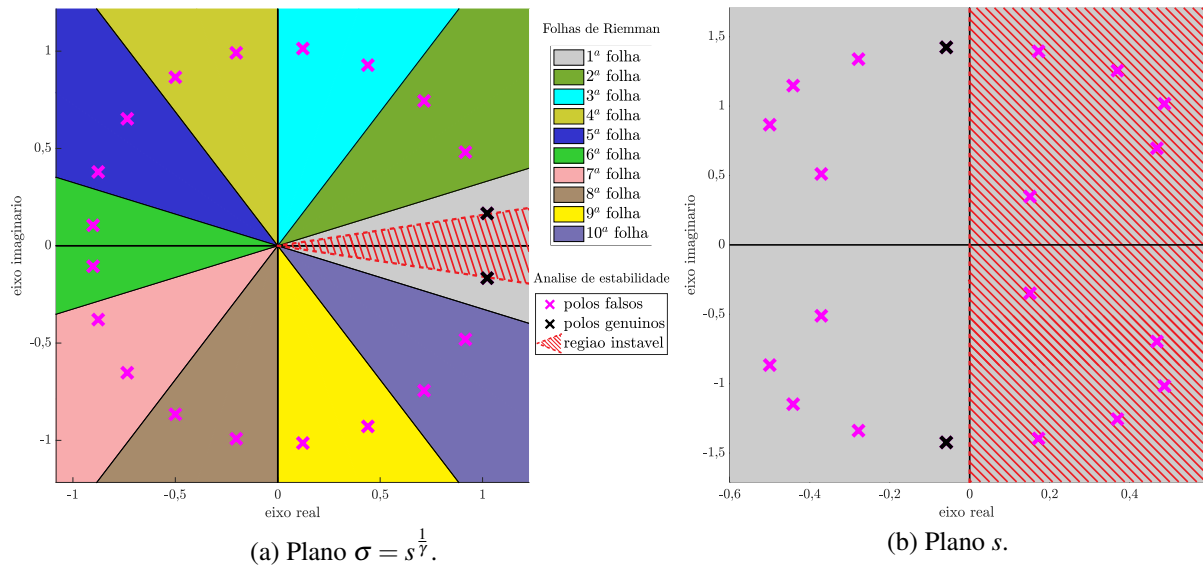
σ_j	$Q(\sigma_j)$	s_j	$Q(s_j)$
$\sigma_{1,2} = 1,0226 \pm 0,1663i$	0	$s_{1,2} = -0,0584 \pm 1,4234i$	0
$\sigma_{3,4} = 0,9160 \pm 0,4810i$	0	$s_{3,4} = 0,1724 \mp 1,3949i$	$0,1078 \mp 0,6303i$
$\sigma_{5,6} = 0,7144 \pm 0,7443i$	0	$s_{5,6} = -0,2780 \pm 1,3386i$	$0,3010 \mp 0,5621i$
$\sigma_{7,8} = 0,4400 \pm 0,9283i$	0	$s_{7,8} = 0,3696 \mp 1,2556i$	$0,5789 \mp 1,0599i$
$\sigma_{9,10} = 0,1226 \pm 1,0135i$	0	$s_{9,10} = -0,4415 \pm 1,1479i$	$0,8792 \mp 0,8169i$
$\sigma_{11,12} = -0,2028 \pm 0,9916i$	0	$s_{11,12} = 0,4874 \mp 1,0172i$	$1,2085 \mp 1,1051i$
$\sigma_{13,14} = -0,5000 \pm 0,8660i$	0	$s_{13,14} = -0,5000 \pm 0,8660i$	$1,4781 \mp 0,6581i$
$\sigma_{15,16} = -0,7347 \pm 0,6525i$	0	$s_{15,16} = 0,4685 \mp 0,6963i$	$1,7126 \mp 0,7485i$
$\sigma_{17,18} = 0,8765 \pm 0,3793i$	0	$s_{17,18} = -0,3711 \pm 0,5110i$	$1,8086 \mp 0,1710i$
$\sigma_{19,20} = -0,9015 \pm 0,1052i$	0	$s_{19,20} = 0,1510 \mp 0,3482i$	$1,8031 \mp 0,2104i$

Fonte: elaborada pelo autor.

nos planos σ e s , assim como as regiões de instabilidade em cada plano e as folhas de Riemann são ilustradas na [Figura 7](#), enquanto que a superfície de Riemann e suas conexões com os dois planos são apresentados na [Figura 8](#).

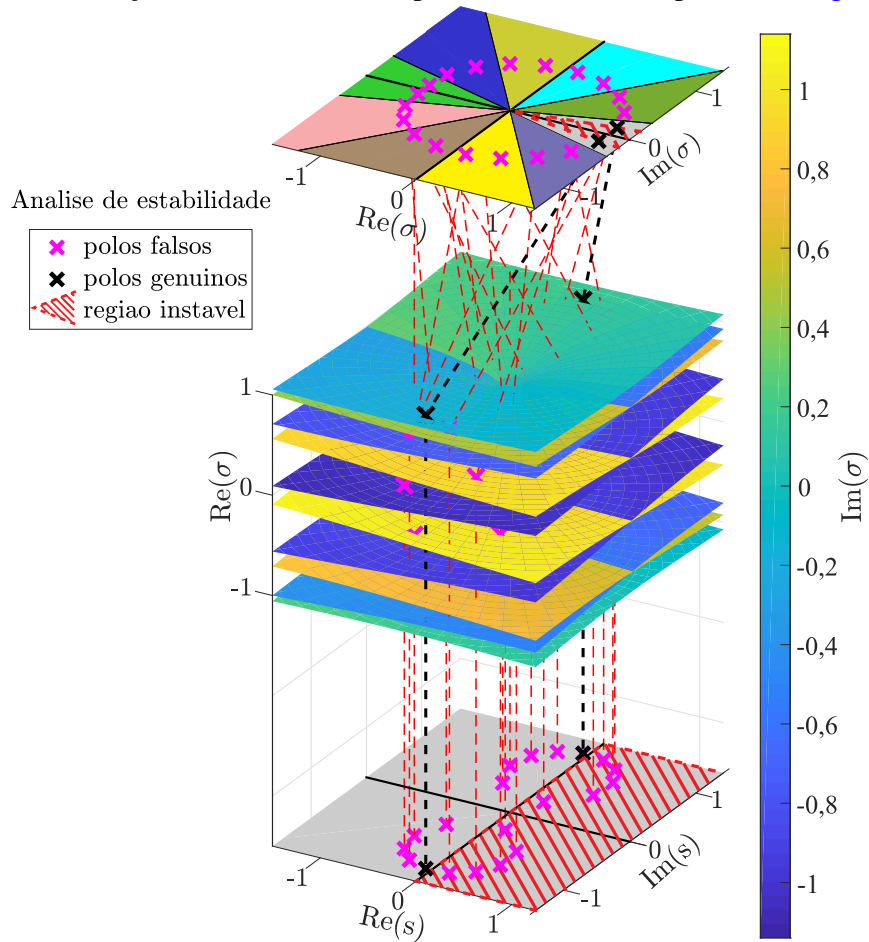
Pode-se observar as 10 folhas de Riemann presentes na [Figura 7.a](#), assim como a região de instabilidade, proporcional a $\frac{\pi}{2\gamma} = 0,15707$ rads (vide [Figura 4](#)). É interessante enfatizar que, como mencionado no exemplo anterior, as raízes aparentemente localizadas no semiplano direito do plano s não tornam o sistema instável, uma vez que, na realidade, encontram-se em outras folhas de Riemann e não constituem raízes físicas. Portanto, o sistema é estável e com apenas duas raízes. Observa-se, a partir da superfície de Riemann, que valores elevados de γ , *e.g.* $\gamma = 10$, dificultam a visualização dos detalhes da transformação $\sigma = s^{\frac{1}{\gamma}}$. Porém, ressalta-se que ocorrem os mesmos fenômenos observados no exemplo anterior.

Figura 7 – Polos falsos e genuínos nos planos σ e s para o Exemplo 2.5.2.



Fonte: elaborada pelo autor.

Figura 8 – Transformação $\sigma = s^{\frac{1}{7}}$ em uma superfície de Riemann para o Exemplo 2.5.2, $\gamma = 10$.



Fonte: elaborada pelo autor.

Exemplo 2.5.3: caso $Q(s) = s^{1,1} + s^{0,5} + 1$

Por fim, como último exemplo da aplicação do teorema de [Matignon \(1996\)](#), apresenta-se um caso peculiar, cuja equação característica é

$$Q(s) = s^{1,1} + s^{0,5} + 1. \quad (2.46)$$

Aplicando os mesmos passos do exemplo anterior, tem-se $\gamma = 10$, o que resulta em

$$Q(\sigma) = \sigma^{11} + \sigma^5 + 1. \quad (2.47)$$

Para este caso, a condição de estabilidade também é $|\arg(\sigma_j)|, \forall j$ serem maiores que $\frac{\pi}{2\gamma} = 0,157079$, enquanto que a condição para dada raiz estar na primeira folha de Riemann também é $|\arg(\sigma_j)|$ ser menor que $\frac{\pi}{\gamma} = 0,314159$. Os valores de σ_j , $|\arg(\sigma_j)|$, $Q(\sigma_j)$, s_j e $Q(s_j)$ são exibidos na [Tabela 3](#) e [Tabela 4](#). As localizações dos polos nos planos σ e s , as regiões de instabilidade em cada plano e as folhas de Riemann são ilustradas na [Figura 9](#), enquanto que a superfície de Riemann e suas conexões com os dois planos são apresentadas na [Figura 10](#).

Nota-se que, seguindo os critérios do teorema de [Matignon \(1996\)](#), todas as raízes estão fora da região de instabilidade, o que significa que o sistema é estável. Porém, não há raízes na primeira folha de Riemann. Daí, um questionamento que pode surgir é: como é possível um sistema ser estável se não há raízes na primeira folha?

Analisando o teorema de [Matignon \(1996\)](#) do ponto de vista lógico-matemático, conclui-se que o que garante a estabilidade do sistema é a negação da condição suficiente para a instabilidade. Para o sistema ser instável, obrigatoriamente deve haver ao menos uma raiz na primeira folha, uma vez que a zona de instabilidade está delimitada dentro da primeira folha. Logo, se não há raízes na primeira folha, não há como o sistema estar na região de instabilidade. Observa-se que este é um sistema estável mas com polos não físicos, resultando em um sistema não físico. A aparente conclusão que se pode inferir é a de que este não é um sistema fisicamente realizável, existindo apenas a sua representação matemática abstrata, pois as limitações físicas/naturais impedem o sistema de existir no mundo real.

Tabela 3 – Valores de σ_j e $|\arg(\sigma_j)|$ para o Exemplo 2.5.3.

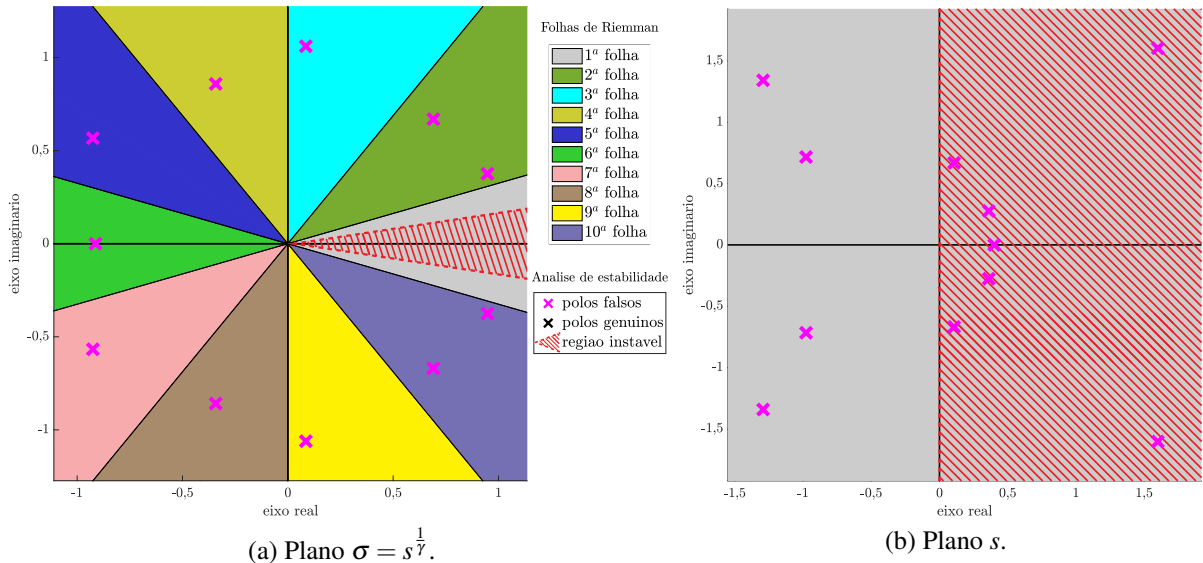
σ_j	$ \arg(\sigma_j) $
$\sigma_{1,2} = 0,9477 \pm 0,3756i$	$ \arg(\sigma_{1,2}) = 0,377395$
$\sigma_{3,4} = 0,6908 \pm 0,6692i$	$ \arg(\sigma_{3,4}) = 0,769494$
$\sigma_{5,6} = 0,0855 \pm 1,0607i$	$ \arg(\sigma_{5,6}) = 1,490363$
$\sigma_{7,8} = -0,9254 \pm 0,5669i$	$ \arg(\sigma_{7,8}) = 2,591917$
$\sigma_9 = -0,9127$	$ \arg(\sigma_9) = 3,141593$
$\sigma_{10,11} = -0,3423 \pm 0,8585i$	$ \arg(\sigma_{10,11}) = 1,950158$

Fonte: elaborada pelo autor.

Tabela 4 – Valores das raízes e dos resíduos dos polinômios para o Exemplo 2.5.3.

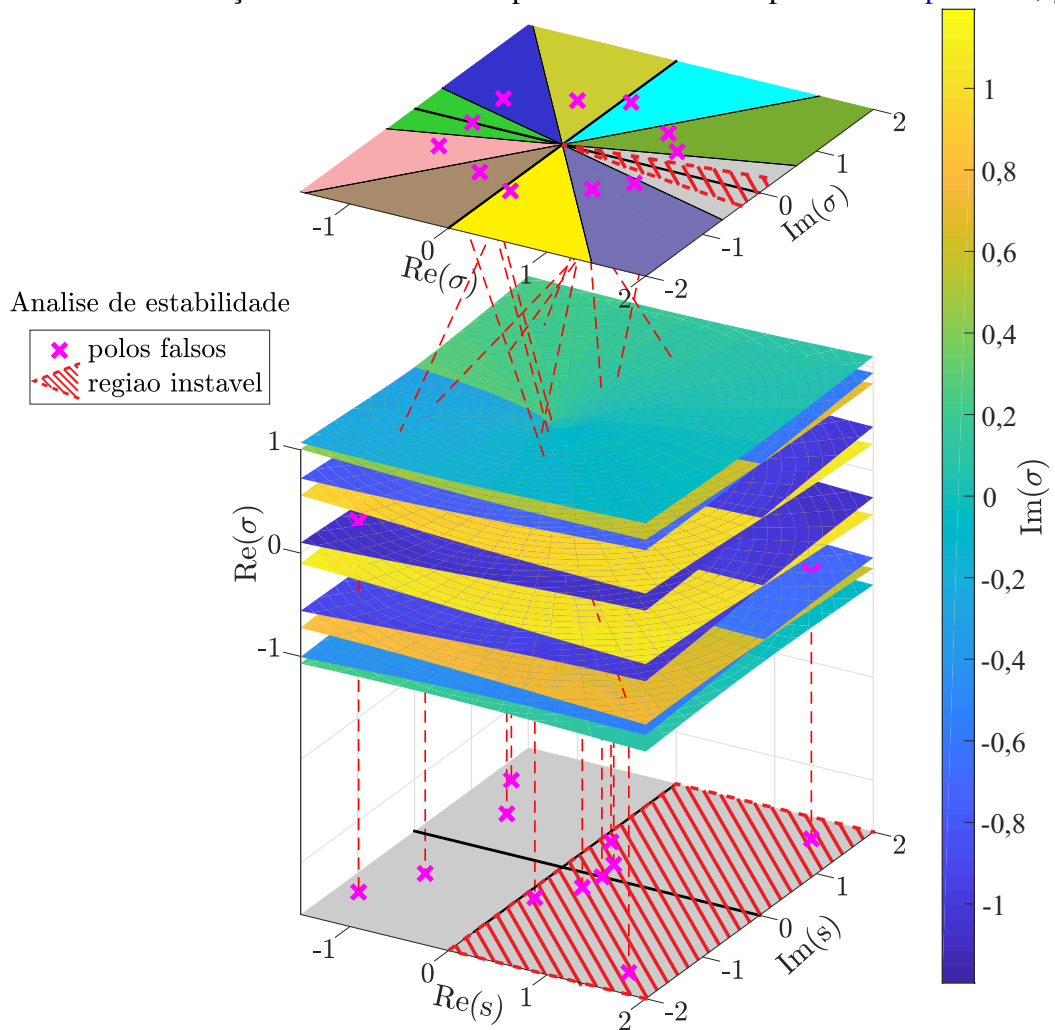
σ_j	$Q(\sigma_j)$	s_j	$Q(s_j)$
$\sigma_{1,2} = 0,9477 \pm 0,3756i$	0	$s_{1,2} = -0,9779 \mp 0,7165i$	$0,1952 \mp 1,5066i$
$\sigma_{3,4} = 0,6908 \pm 0,6692i$	0	$s_{3,4} = 0,1073 \pm 0,6689i$	$1,6380 \pm 1,1855i$
$\sigma_{5,6} = 0,0855 \pm 1,0607i$	0	$s_{5,6} = -1,2923 \pm 1,3422i$	$-0,1344 \pm 2,3271i$
$\sigma_{7,8} = -0,9254 \pm 0,5669i$	0	$s_{7,8} = 1,6005 \pm 1,6038i$	$3,9852 \pm 2,4484i$
$\sigma_9 = -0,9127$	0	$s_9 = 0,4014$	2,0000
$\sigma_{10,11} = -0,3423 \pm 0,8585i$	0	$s_{10,11} = 0,3617 \pm 0,2761i$	$1,9560 \pm 0,4925i$

Fonte: elaborada pelo autor.

Figura 9 – Polos falsos e genuínos nos planos σ e s para o Exemplo 2.5.3.

Fonte: elaborada pelo autor.

Figura 10 – Transformação $\sigma = s^{\frac{1}{\gamma}}$ em uma superfície de Riemann para o Exemplo 2.5.3, $\gamma = 10$.



Fonte: elaborada pelo autor.

2.5.1 Considerações sobre o Teorema de Matignon

O teorema de [Matignon \(1996\)](#) representa um marco histórico no estudo de sistemas de ordem fracionária, por trazer um critério formal de análise da estabilidade destes sistemas. Contudo, este teorema não traz uma resposta analítica para a solução da equação característica $Q(s)$, visto que na mudança de variável $\sigma = s^{\frac{1}{\gamma}}$, é necessário um método numérico para solucionar $Q(\sigma)$.

Esta necessidade ocorre devido o teorema de Abel-Ruffini, apresentado a seguir:

Teorema 2.5.3: Teorema de Abel-Ruffini (1824) *Não existe uma fórmula geral, composta apenas por radicais e operações algébricas elementares, que permita resolver todas as equações polinomiais de grau cinco ou superior com coeficientes racionais.*

Os três exemplos de aplicação do teorema de [Matignon \(1996\)](#) neste capítulo ilustram

que a ordem de γ aumenta consideravelmente a ordem do polinômio $Q(\sigma)$. Se um eventual projeto de controladores de ordem fracionária considerar a ordem de integração λ e/ou derivação μ com truncamento em duas casas decimais, a ordem do polinômio $Q(\sigma)$ pode chegar facilmente na casa das centenas.

Com uma solução analítica do polinômio $Q(s)$, relações diretas de estabilidade entre os parâmetros da planta e do controlador poderiam ser traçadas, delimitando os valores de ganhos do controlador que resultam em um sistema estável. O próximo capítulo desta tese aborda esta ideia, introduzindo, pela primeira vez, uma metodologia para determinar os polos de um sistema de controle em malha fechada cuja equação característica é composta por uma certa classe de polinômios fracionários: os trinômios de ordem fracionária.

3 A FUNÇÃO LAMBERT-TSALLIS

Neste capítulo é apresentada a função Lambert-Tsallis na [Seção 3.1](#), juntamente com um algoritmo capaz de calcular todos os seus valores para um dado argumento, incluindo aqueles pertencentes a ramos complexos. Uma análise da morfologia da função é exposta na [Seção 3.2](#) para valores reais e na [Seção 3.3](#) para valores complexos. A função de Lambert-Tsallis é utilizada no desenvolvimento de um método analítico para estabilidade de uma classe de sistemas de controle fracionário, apresentado no próximo capítulo, que explora a capacidade dessa função de solucionar analiticamente determinadas equações características de ordem fracionária.

3.1 A função Lambert-Tsallis, $W_q(z)$

Antes de definir a função de Lambert-Tsallis, é preciso definir a função de Lambert, $W(z)$. Esta é uma função transcendental introduzida por [Lambert \(1758\)](#) e [Euler \(1783\)](#), e que aparece na seguinte expressão, conhecida como Equação de Lambert,

$$W(z)e^{W(z)} = z, \quad (3.1)$$

em que $W(z) \in \mathbb{C}$, $\forall z \in \mathbb{C}$ ¹. A [Equação 3.1](#) revela que o produto de $W(z)$ por uma operação não linear sobre $W(z)$, no caso, a exponencial de $W(z)$, produz o próprio argumento da função $W(z)$. Uma consequência interessante da [Equação 3.1](#) é que a função de Lambert $W(z)$ em si não pode ser isolada, ou seja, a função de Lambert é definida *implicitamente* pela equação de Lambert.

A função Lambert possui presença importante na física ([COUTO et al., 2025](#)) e química ([DUBINOV, 2025](#)), pois surge de maneira recorrente em diversos problemas aplicados. Mais recentemente, surge em aplicações na engenharia, como no cálculo analítico da corrente em um modelo de célula solar ([CALASAN et al., 2024](#)) e na solução analítica de determinados sistemas com atraso ([BIN; FUMING, 2021](#)).

Em 2019, motivados pelos seus estudos sobre teoria quântica, [Silva e Ramos \(2019\)](#) propuseram incorporar a exponencial de Tsallis ([UMAROV et al., 2008](#)), também chamada de função q -exponencial, na equação de Lambert. Consequentemente, [Silva e Ramos \(2019\)](#) introduziram uma nova equação cuja solução é uma generalização da função Lambert, que passou a ser chamada de função Lambert-Tsallis, $W_q(z)$, sendo expressa como

$$W_q(z)e_q^{W_q(z)} = z, \quad (3.2)$$

¹Nessa expressão, z representa o argumento da função $W(z)$, não devendo ser confundido com a variável normalmente utilizada na transformada z , aplicada a sistemas de controle discretos.

que pode ser reescrita como

$$W_q(z)[1 + (1 - q)W_q(z)]^{\frac{1}{1-q}} = z, \quad (3.3)$$

em que, por definição, a função q -exponencial é dada por

$$e_q^z \equiv [1 + (1 - q)z]^{\frac{1}{1-q}}, \quad (3.4)$$

com $q \in \mathbb{R}$. Na função q -exponencial, o parâmetro q é um termo crucial que determina a natureza do crescimento da função. Tal parâmetro introduz uma não-linearidade que pode modelar uma gama mais ampla de comportamentos de crescimento, dependendo de seu valor. Para $q = 1$, a função q -exponencial se reduz à função exponencial padrão e^z . Quando $q > 1$, a função apresenta crescimento super-exponencial, aumentando mais rapidamente do que a exponencial padrão. Por outro lado, para $q < 1$, a função demonstra crescimento sub-exponencial. Assim, o parâmetro q permite uma representação flexível do comportamento exponencial, adaptando a função a diferentes propriedades de escala e complexidades para um determinado problema.

Diversos trabalhos, como [Andrade et al. \(2022\)](#), [Guedes et al. \(2022\)](#), [Almeida e Ramos \(2023\)](#) e [Silva \(2024\)](#), seguem [Silva e Ramos \(2019\)](#) em sua notação sobre a função Lambert-Tsallis e sua parametrização em termos de q . Porém, referir-se às funções q -exponencial e à Lambert-Tsallis em termos do parâmetro $r = \frac{1}{1-q}$, $r \in \mathbb{R}$, em vez de q , simplifica as manipulações algébricas e revela propriedades matemáticas importantes. Assim, pode-se reescrever as [Equações 3.2 e 3.3](#), respectivamente, como

$$W_r(z) \exp_r[W_r(z)] = z \quad (3.5)$$

e

$$W_r(z) \left[1 + \frac{W_r(z)}{r} \right]^r = z, \quad (3.6)$$

em que, por definição,

$$\exp_r(z) \equiv \left(1 + \frac{z}{r} \right)^r. \quad (3.7)$$

Nota-se que à medida que $r \rightarrow \infty$, $\exp_r(z) \rightarrow e^z$, visto que uma das definições do número de Euler, e , é

$$e = \lim_{n \rightarrow \infty} \left(1 + \frac{1}{n} \right)^n. \quad (3.8)$$

Como resultado, têm-se que

$$\lim_{r \rightarrow \infty} W_r(z) = W(z). \quad (3.9)$$

3.1.1 Propriedades de $W_r(z)$

Existem inúmeras propriedades de $W_r(z)$, a maioria das quais surge como consequência da [Equação 3.6](#) e da definição de $\exp_r(z)$, bem como da interação entre as duas expressões. Algumas propriedades são de grande relevância e, por isso, fazem-se necessárias suas apresentações a seguir.

Propriedade 3.1: $1 + z = \exp_1(z)$

Demonstração: Aplicando diretamente a [Equação 3.7](#),

$$\begin{aligned}\exp_r(z) &= \left(1 + \frac{z}{r}\right)^r \\ \exp_1(z) &= \left(1 + \frac{z}{1}\right)^1\end{aligned}\tag{3.10}$$

$$\exp_1(z) = 1 + z.\tag{3.11}$$

O mesmo pode ser feito para $1 - z = \exp_1(-z)$. \square

Propriedade 3.2: $[\exp_r(z)]^\alpha = \exp_{r\alpha}(z\alpha)$

Demonstração:

$$[\exp_r(z)]^\alpha = \left[\left(1 + \frac{z}{r}\right)^r\right]^\alpha\tag{3.12}$$

$$= \left(1 + \frac{z}{r}\right)^{r\alpha}\tag{3.13}$$

$$= \left(1 + \frac{z\alpha}{r\alpha}\right)^{r\alpha}.\tag{3.14}$$

Fazendo $r\alpha = \beta$, e posteriormente, aplicando a definição de $\exp_r(z)$ na [Equação 3.7](#),

$$[\exp_r(z)]^\alpha = \left(1 + \frac{z\alpha}{\beta}\right)^\beta\tag{3.15}$$

$$[\exp_r(z)]^\alpha = \exp_\beta(z\alpha)\tag{3.16}$$

$$[\exp_r(z)]^\alpha = \exp_{r\alpha}(z\alpha).\tag{3.17}$$

\square

Propriedade 3.3: $\left[\frac{W_r(z)}{z}\right]^{\frac{1}{r}} = \frac{1}{1 + \frac{W_r(z)}{r}}$

Demonstração: A partir da [Equação 3.6](#),

$$W_r(z) \left[1 + \frac{W_r(z)}{r} \right]^r = z$$

$$W_r(z)^{\frac{1}{r}} \left[1 + \frac{W_r(z)}{r} \right] = z^{\frac{1}{r}} \quad (3.18)$$

$$\frac{W_r(z)^{\frac{1}{r}}}{z^{\frac{1}{r}}} = \frac{1}{1 + \frac{W_r(z)}{r}} \quad (3.19)$$

$$\left[\frac{W_r(z)}{z} \right]^{\frac{1}{r}} = \frac{1}{1 + \frac{W_r(z)}{r}}. \quad (3.20)$$

□

Propriedade 3.4: $1 + \frac{W_r(z)}{r} = - \frac{W_{\frac{1}{r}} \left[-\frac{1}{r} \left(-\frac{z}{r} \right)^{\frac{1}{r}} \right]}{\frac{1}{r}}$

Demonstração: Partindo das [Equações 3.26](#), [3.27](#) e [3.28](#), define-se $q = 1 + y$, o que resulta em $q^r(1 - q) = -\frac{z}{r}$. Aplicando a [Propriedade 3.1](#), tem-se que $q^r \exp_1(-q) = -\frac{z}{r}$. Elevando ambos os lados da equação a $\alpha = \frac{1}{r}$,

$$q^{\alpha} \exp_{\alpha}(-\alpha q) = \left(-\frac{z}{r} \right)^{\alpha}. \quad (3.21)$$

Multiplicando ambos os lados por $-\alpha$,

$$-\alpha q \exp_{\alpha}(-\alpha q) = -\alpha \left(-\frac{z}{r} \right)^{\alpha}. \quad (3.22)$$

Observa-se o paralelo traçado com a equação de Lambert-Tsallis, descrita na [Equação 3.5](#) como $W_r(z) \exp_r[W_r(z)] = z$, de forma que

$$q = \frac{W_{\alpha} \left[-\alpha \left(-\frac{z}{r} \right)^{\alpha} \right]}{-\alpha}. \quad (3.23)$$

Por fim, aplicando este resultado na definição $q = 1 + y$, na [Equação 3.27](#), e substituindo $\alpha = \frac{1}{r}$,

$$1 + \frac{W_r(z)}{r} = - \frac{W_{\frac{1}{r}} \left[-\frac{1}{r} \left(-\frac{z}{r} \right)^{\frac{1}{r}} \right]}{\frac{1}{r}}. \quad (3.24)$$

□

3.1.2 Algoritmo para determinação de $W_r(z)$

Assim como a função Lambert, a função Lambert-Tsallis é uma função transcendental. Além disso, como pode ser visto na [Equação 3.6](#), a função de Lambert-Tsallis é multivalorada. No entanto, devido à q -exponencial de Tsallis, a natureza dessa equação é simples, podendo suas soluções, ou seja, os valores de $W_r(z)$, serem encontradas sem muito esforço. A seguir, é apresentado um algoritmo para calculá-las, podendo ser implementada em linguagens de programação como Matlab/Octave e Python. Cabe destacar que, embora a função Lambert-Tsallis já tenha sido explorada em diversas aplicações na literatura especializada, esta é a primeira vez que um algoritmo para calcular todos os valores de $W_r(z)$, incluindo os complexos, é apresentado.

Substituindo $r = \frac{n}{d} > 0$, com $\gcd(n, d) = 1$ (i.e., n e d são coprimos e \gcd é a sigla de *greatest common divisor*) na [Equação 3.6](#), tem-se que

$$W_r(z) \left[1 + \frac{d}{n} W_r(z) \right]^{\frac{n}{d}} = z \quad (3.25)$$

$$\frac{d}{n} W_r(z) \left[1 + \frac{d}{n} W_r(z) \right]^{\frac{n}{d}} = \frac{d}{n} z. \quad (3.26)$$

O próximo passo envolve uma segunda substituição,

$$y = W_r(z) \frac{d}{n}, \quad (3.27)$$

na [Equação 3.26](#) para obter

$$y(1+y)^{\frac{n}{d}} = \frac{d}{n} z. \quad (3.28)$$

Finalmente, como último passo, define-se

$$k = (1+y)^{\frac{1}{d}}, \quad (3.29)$$

e a [Equação 3.28](#) pode ser reescrita como

$$(k^d - 1)k^n = \frac{d}{n} z \quad (3.30)$$

$$k^{n+d} - k^n - \frac{d}{n} z = 0. \quad (3.31)$$

Nota-se que a [Equação 3.31](#) é uma equação polinomial de grau inteiro $n + d$, ou seja, existirão $n + d$ soluções em k . Após encontrar as soluções k , pode-se calcular $W_r(z)$ através da redefinição da [Equação 3.29](#) e da [Equação 3.27](#) como

$$y = k^d - 1 \quad (3.32)$$

$$W_r(z) = y \frac{n}{d}. \quad (3.33)$$

Algoritmo 1: Resolução da Equação da Função Lambert-Tsallis.

Dados: $r \in \mathbb{Q}_+^*$, $z \in \mathbb{C}$

Resultado: $W_r(z)$

$\frac{n}{d} \leftarrow r$;

Defina e resolva: $k^{n+d} - k^n - \frac{d}{n}z = 0$;

$y \leftarrow k^d - 1$;

$W_r(z) \leftarrow y \frac{n}{d}$;

retorna $W_r(z)$;

Portanto, para $r = \frac{n}{d} > 0$, $W_r(z)$ possui $n + d$ valores. Ressalta-se que todos estes procedimentos podem ser facilmente implementados utilizando linguagens de programação que possuam funções equivalentes à função `roots()` do Matlab/Octave. No [Algoritmo 1](#), é apresentado a resolução da equação de Lambert-Tsallis para obter os valores de $W_r(z)$.

Embora tenha uma definição simples, a função $W_r(z)$ possui aspectos importantes que a tornam matematicamente rica, como ser uma função multivalorada, possuir dois parâmetros e capacidade de modelar fenômenos polinomiais.

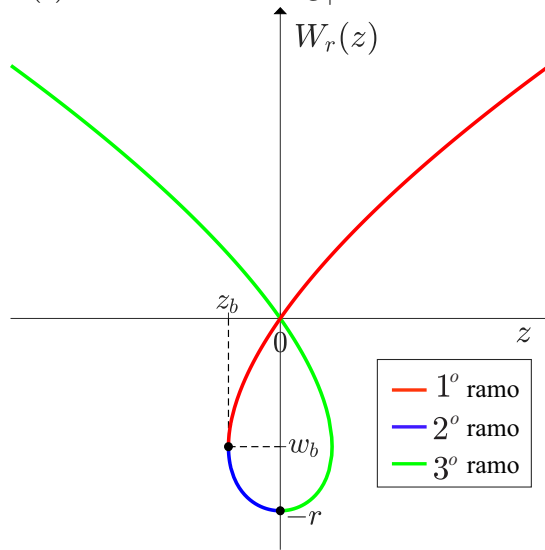
3.2 Polimorfismo de $W_r(z)$

Uma vez que, durante a solução de $W_r(z)$ no [Algoritmo 1](#), há a conversão de $r \in \mathbb{Q}_+^*$ para a fração n/d , pode-se classificar a função $W_r(z)$ em três tipos, conforme a paridade da fração correspondente de r :

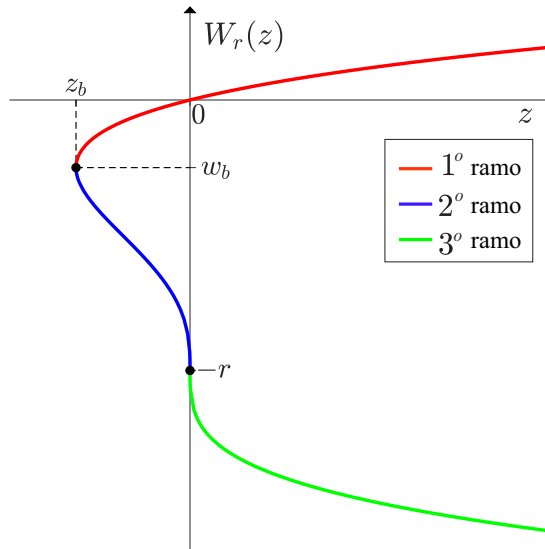
- **tipo 1:** $\frac{n}{d} = \frac{2\ell+1}{2\vartheta}$, $\ell, \vartheta \in \mathbb{Z}_+^*$, ou seja, r é composto pela razão de um *ímpar* sobre *par*.
- **tipo 2:** $\frac{n}{d} = \frac{2\ell+1}{2\vartheta+1}$, $\ell, \vartheta \in \mathbb{Z}_+^*$, ou seja, r é composto pela razão de um *ímpar* sobre *ímpar*.
- **tipo 3:** $\frac{n}{d} = \frac{2\ell}{2\vartheta+1}$, $\ell, \vartheta \in \mathbb{Z}_+^*$, ou seja, r é composto pela razão de um *par* sobre *ímpar*.

Na resolução de problemas aplicados, as funções Lambert e Lambert-Tsallis são frequentemente relacionadas a grandezas físicas mensuráveis, restritas ao domínio dos números reais ([RAMOS, 2022](#)). Por este motivo, a análise dos valores reais de $W_r(z)$ possui grande valor. Na [Figura 11](#), são ilustrados os gráficos de $W_r(z) \in \mathbb{R}$, $\forall z \in \mathbb{R}$, $r \in \mathbb{Q}_+^*$ para os três tipos descritos. Nota-se que para cada tipo, há um gráfico com forma bem definida.

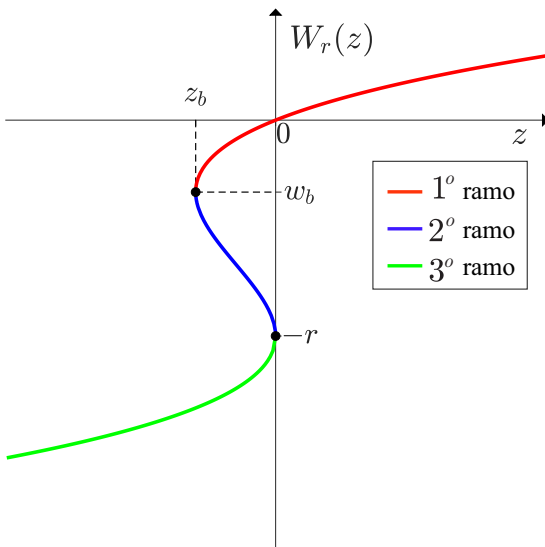
Figura 11 – Gráficos de $W_r(z) \times z \subset \mathbb{R} \times \mathbb{R}$, $r \in \mathbb{Q}_+^*$.



(a) Tipo 1, $r = 1/2$.



(b) Tipo 2, $r = 3/1$.



(c) Tipo 3, $r = 2/1$.

Fonte: elaborada pelo autor.

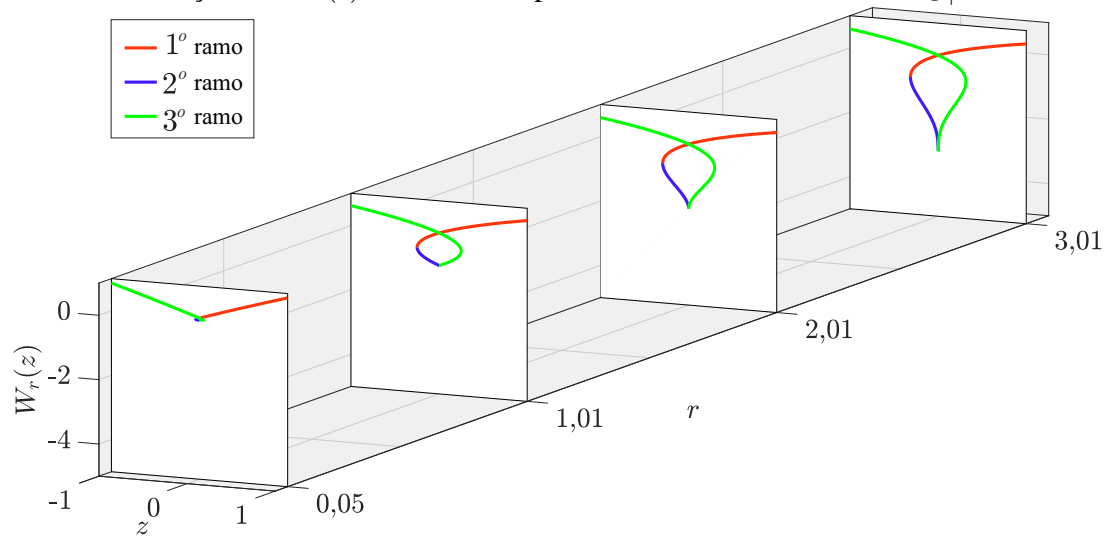
Nota-se que há, no máximo, três valores reais de $W_r(z)$, $\forall z \in \mathbb{R}$. Ao contrário do que ocorre com a função Lambert, a função Lambert-Tsallis possui três ramos reais, os quais são contínuos dentro de três intervalos distintos. Esses intervalos surgem devido a dois pontos de ramificação em $z = \{0, z_b\}$, com $z_b = -\left(\frac{r}{r+1}\right)^{r+1}$ para $r \neq -1$. O correspondente $W_r(z_b) = w_b = -\frac{r}{r+1}$ é analítico com duplicidade e, à medida que $r \rightarrow \infty$, esses valores de w_b e z_b se aproximam dos da função Lambert. Em outras palavras, em um modelo baseado na função de Lambert-Tsallis, há discriminantes analíticos $z = \{0, z_b\}$ que dependem apenas de r .

Estas características descritas, juntamente com a [Figura 11](#), sugerem que informações importantes podem ser extraídas a partir dos pontos de ramificação. Por exemplo, se $r = 4$, há cinco raízes na equação polinomial descrita pela [Equação 3.6](#). Essas raízes apresentam características distintas em três intervalos de z : $(-\infty, z_b]$, $[z_b, 0]$ e $[0, \infty)$. No primeiro e último intervalo, há dois pares de raízes complexas conjugadas e uma raiz real, enquanto no intervalo intermediário há um par de raízes complexas conjugadas e três raízes reais. Como visto no capítulo anterior, as raízes do polinômio característico $Q(s)$ estão associadas a dinâmicas internas do sistema. Consequentemente, se $Q(s)$ puder ser representada em função de $W_r(z)$, e se os pontos de ramificação de $W_r(z)$ forem conhecidos, têm-se uma ferramenta poderosa para analisar analiticamente a estabilidade de sistemas de controle de ordem fracionária.

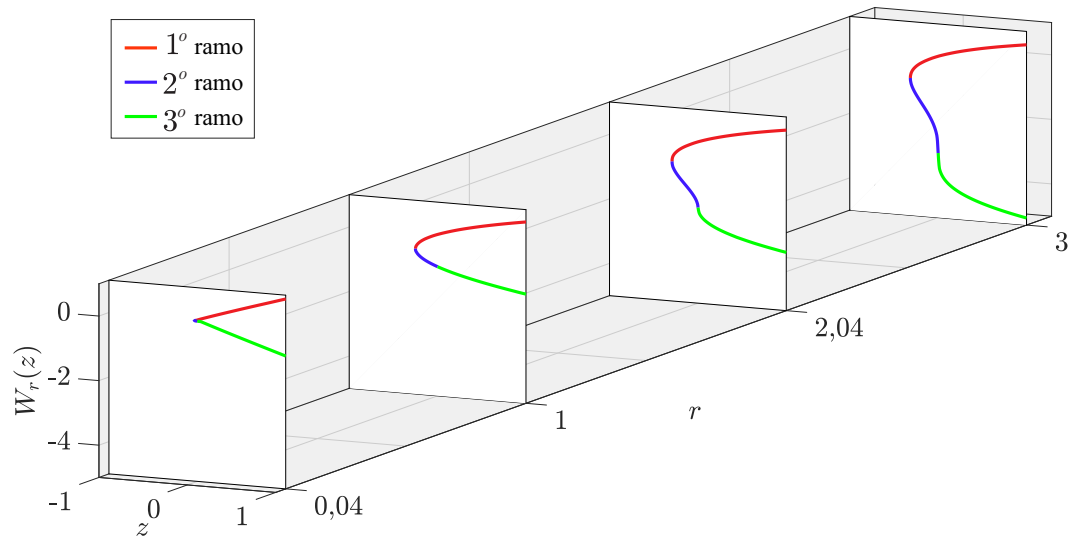
Uma outra característica da função $W_r(z)$, é que as três morfologias dos gráficos vistos na [Figura 11](#) não são estáticas, variando conforme o valor de r , como ilustrado na [Figura 12](#). É pertinente questionar se o fato de uma variação mínima de r ser capaz de mudar a morfologia de $W_r(z)$ não torna toda a análise sensível em relação a r . Porém, observa-se na [Figura 12](#) que, para r bastante próximos e de tipos diferentes, como 1, 1,01 e 1,04, ou 3, 3,01 e 3,04, a diferença entre as três formas está apenas no posicionamento do terceiro ramo (em verde). A princípio, isso não é um problema, uma vez a função de Lambert-Tsallis apresenta informações úteis para a resolução de problemas de aplicação majoritariamente no primeiro e segundo ramo, como em [Guedes et al. \(2022\)](#) e [Ramos \(2022\)](#).

De fato, se considerarmos uma mudança mínima em r , como uma pequena variação na terceira casa decimal presente na [Figura 13](#), pode-se ter os 3 tipos de gráfico, nos quais os dois primeiros ramos são idênticos. Portanto, variações mínimas de r , embora possam alterar a morfologia do gráfico, resultam em variações também mínimas de $W_r(z)$ nos primeiros dois ramos, que estão relacionados aos dois únicos ramos reais da função Lambert.

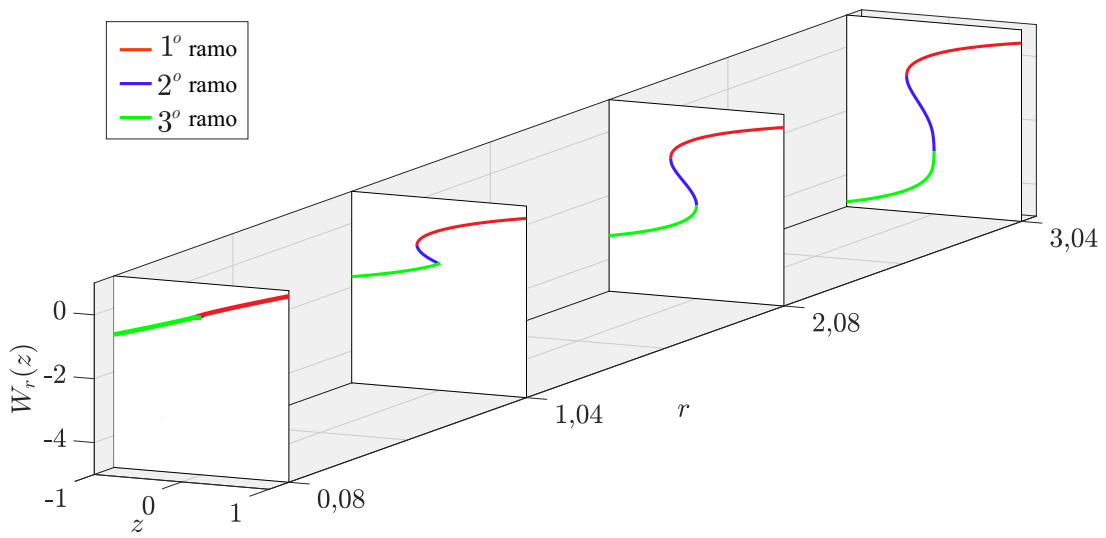
Figura 12 – Evolução de $W_r(z) \times z \subset \mathbb{R} \times \mathbb{R}$ para diferentes valores de $r \in \mathbb{Q}_+^*$.



(a) Tipo 1.



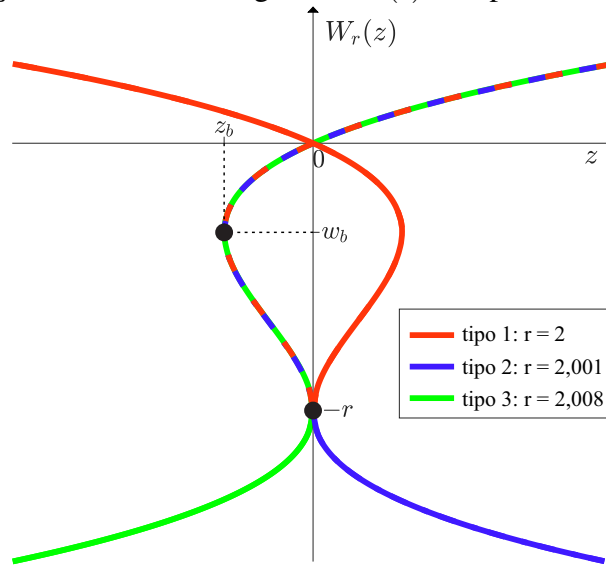
(b) Tipo 2.



(c) Tipo 3.

Fonte: elaborada pelo autor.

Figura 13 – Comparação das três morfologias de $W_r(z) \in \mathbb{R}$ para valores próximos de r .



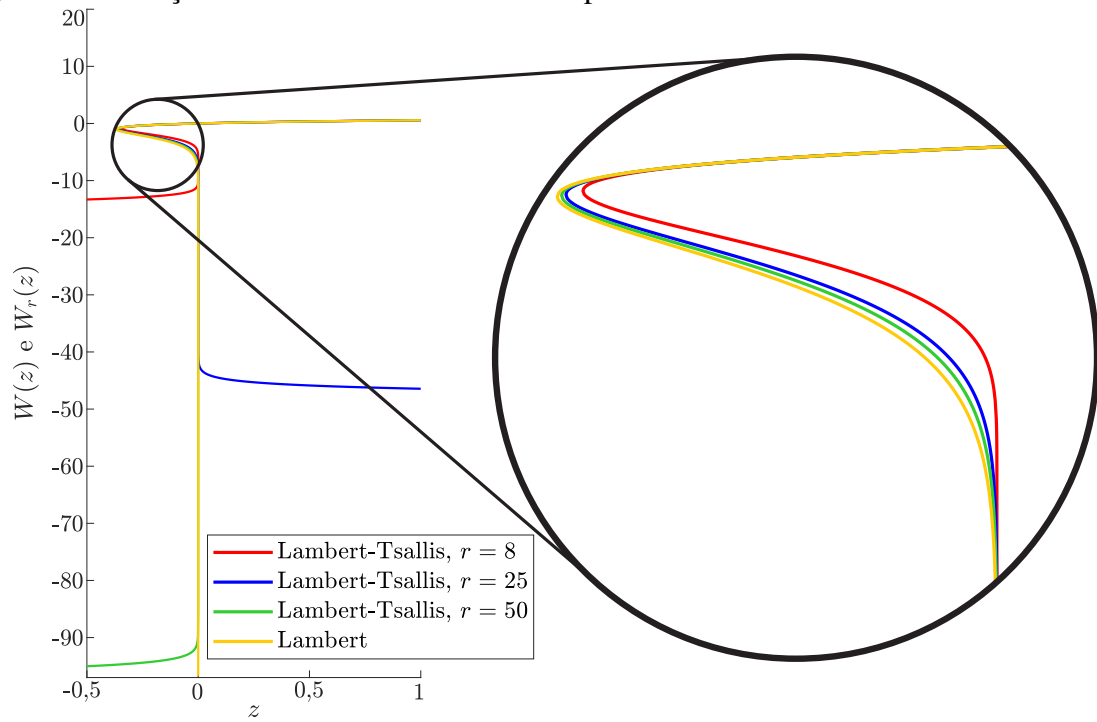
Fonte: elaborada pelo autor.

Uma comparação entre a função de Lambert-Tsallis e a função de Lambert é ilustrada na Figura 14 para diferentes valores de r . Observa-se que à medida que r aumenta, os primeiros ramos da Lambert-Tsallis se aproximam dos ramos da função Lambert, cujo segundo ramo toca o eixo das ordenadas em $W(z) = -\infty$. Na Lambert-Tsallis, o segundo ramo toca o eixo em $W_r(z) = -r$. Nota-se que o aumento de r faz o ponto em que o segundo ramo toca o eixo das ordenadas ser cada vez menor. De fato, $\lim_{r \rightarrow \infty} W_r(z) = W(z)$, como descrito pela Equação 3.8 e Equação 3.9, fazendo o terceiro ramo desaparecer.

Porém, é importante ressaltar que mesmo para valores de r bastante elevados, mas que se enquadrem no tipo 1, o terceiro ramo, que se inicia em um valor tão negativo quanto $-r$, estende-se, fazendo $W_r(z)$ cruzar novamente o eixo das abscissas e voltando a ter valores positivos, conforme ilustrado na Figura 15. O estranho comportamento do terceiro ramo para diferentes tipos de r , aparentemente errático e aleatório, segue um padrão que depende do número de casas decimais de r que se leva em consideração. Uma análise empírica revelou que para um número x de casas decimais consideradas, a sequência de aparição dos tipos de r , e consequentemente de $W_r(z)$, apresenta exatamente $N = 2^x - 1$ aparições consecutivas do tipo 1, e, entre estas aparições do tipo 1, os tipos 2 e 3 surgem de forma unitária e alternadas. Exemplos deste padrão são apresentados na Tabela 5. A periodicidade observada na sequência é uma manifestação da interação entre a função totiente² de Euler e a estrutura aritmética da base decimal, fenômenos associados à teoria dos números.

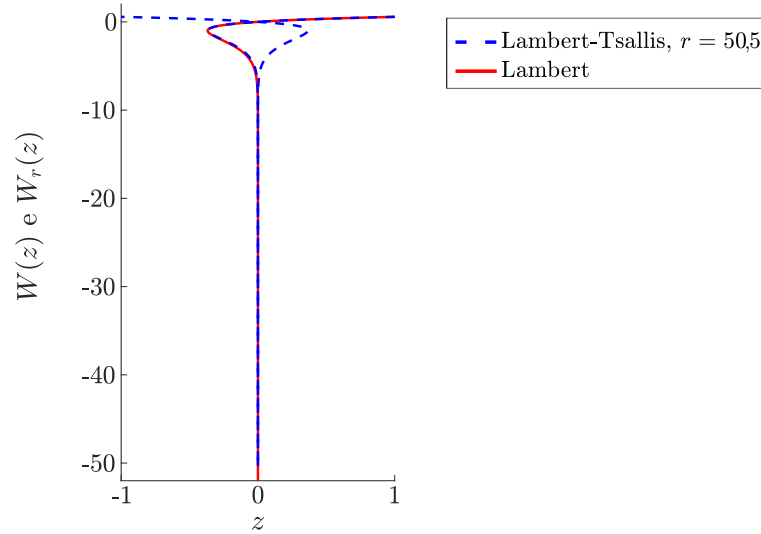
²A função totiente de Euler $\phi(\cdot)$ determina a quantidade de inteiros positivos menores ou iguais a um número que são coprimos deste número (SHAPIRO, 2008). Consequentemente, é possível contabilizar quantas frações $r = n/d$ são irredutíveis, influenciando na sequência de tipos 1, 2 e 3.

Figura 14 – Funções Lambert e Lambert-Tsallis para diferentes valores de r .



Fonte: elaborada pelo autor.

Figura 15 – Comparação entre as funções Lambert e Lambert-Tsallis para r do tipo 1.



Fonte: elaborada pelo autor.

Tabela 5 – Sequência periódica dos tipos 1, 2 e 3 em função do número de casas decimais (x).

casas decimais		sequência											$2^x - 1$
$x = 1$	r	... 1,	1,1,	1,2,	1,3,	1,4,	1,5,	1,6,	1,7,	1,8,	1,9	2 ...	1
	tipo	... 2	1	3	1	2	1	3	1	2	1	3 ...	
$x = 2$	r	... 1,	1,01,	1,02,	1,03,	1,04,	1,05,	1,06,	1,07,	1,08,	1,09,	1,1...	3
	tipo	... 2	1	1	1	3	1	1	1	2	1	1...	
$x = 3$	r	... 1,	1,001,	1,002,	1,003,	1,004,	1,005,	1,006,	1,007,	1,008,	1,009,	1,01...	7
	tipo	... 2	1	1	1	1	1	1	1	3	1	1...	

3.3 Um novo desenvolvimento: análise de $W_r(z) \in \mathbb{C}$

Embora a função Lambert-Tsallis seja comumente associada a grandezas físicas em problemas aplicados, o que restringe sua análise aos ramos reais, não se pode ignorar a relevância dos ramos complexos em contextos específicos. Ainda que todos os trabalhos anteriores com aplicações da função Lambert-Tsallis tenham realizado apenas análises de seus ramos reais, pode-se traçar paralelos com a função $W(z)$ de Lambert para aplicações em que os ramos complexos são de extrema relevância e importância, como [Das et al. \(2022\)](#). Contudo, convém destacar que nesta tese é apresentada a primeira aplicação da Lambert-Tsallis com foco em $W_r(z) \in \mathbb{C}$.

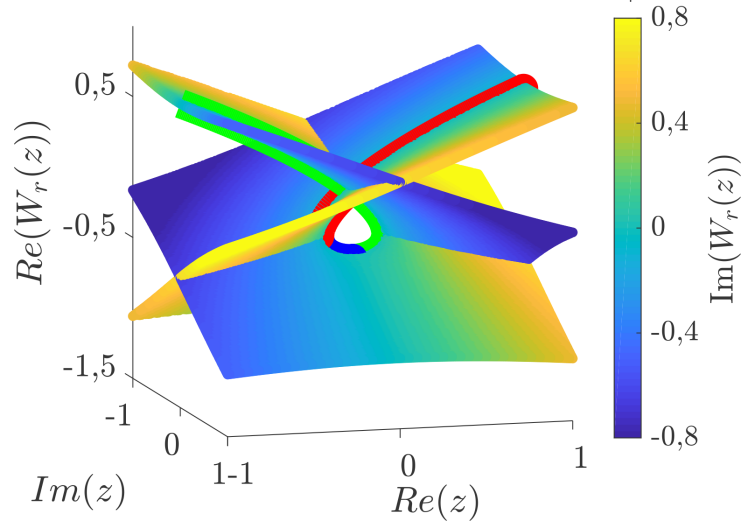
A superfície de Riemann da função $W_r(z) \in \mathbb{C}$, $\forall z \in \mathbb{C}$, é ilustrada na [Figura 16](#), em que os eixos x , y e z correspondem respectivamente à parte real de z , à parte imaginária de z e à parte real de $W_r(z)$. A quarta dimensão do gráfico corresponde à parte imaginária de $W_r(z)$ e é apresentada de forma cromática. Nota-se a morfologia similar àquela dos gráficos $W_r(z) \times z \subset \mathbb{R} \times \mathbb{R}$ presentes na [Figura 11](#). Observa-se que os três ramos reais encontram-se em diferentes folhas de Riemann. O conceito de *ramos reais* perde o sentido para $z \in \mathbb{C}$, pois no domínio complexo, não há apenas curvas mas sim superfícies reais, um subconjunto da superfície de Riemann que contém todos os valores reais de $W_r(z)$, conforme ilustrado na [Figura 17](#).

Embora seja possível a análise de $W_r(z)$ para $z \in \mathbb{C}$, a natureza do problema de sistemas de controle requer prioritariamente $z \in \mathbb{R}$, apenas³. Isso significa que apenas um recorte da superfície de Riemann é preciso para a análise; a saber, a seção transversal em que $\text{Im}(z) = 0$. O gráfico de $W_r(z) \in \mathbb{C}$, $\forall z \in \mathbb{R}$, é ilustrado na [Figura 18](#) para os três tipos de r , em que, para melhor visualização, os eixos x , y e z foram reorganizados de forma a representar, respectivamente, a parte imaginária de $W_r(z)$, os valores reais de z e a parte real de $W_r(z)$.

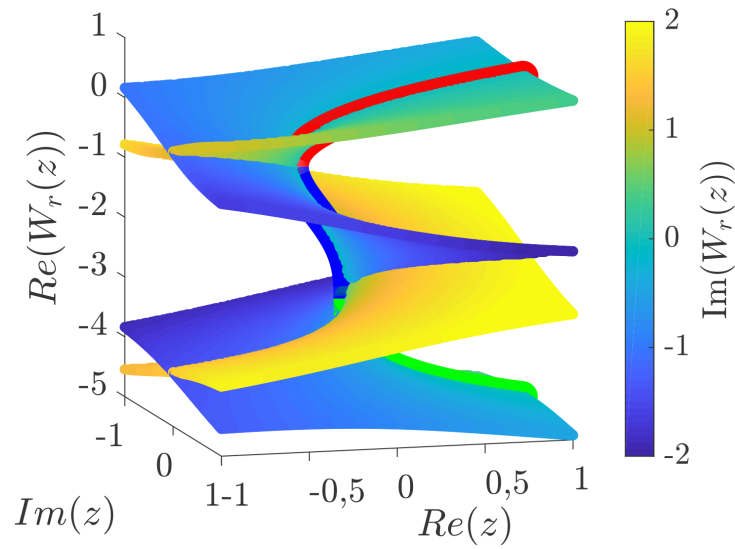
Observa-se que, além dos ramos reais, apresentados anteriormente, ramos complexos surgem nos pontos de ramificação conhecidos, $\{z_b, w_b\}$ e $\{0, -r\}$. Além disso, na [Figura 18.a](#), nota-se o surgimento de um ramo complexo no ponto $\{-z_b, w_b\}$. De maneira análoga à [Figura 12](#), a evolução de $W_r(z) \times z \subset \mathbb{C} \times \mathbb{R}$ é apresentada na [Figura 19](#). Nota-se que os únicos ramos que sempre existem são os ramos reais. De fato, os ramos complexos surgem da necessidade de complementar o número de raízes $n + d$ de $W_r(z)$. O número de pares de ramos complexo-conjugados pode-se ser calculado como $n + d - 1$. Por fim, nota-se que ramos complexos também podem surgir em $\{0, 0\}$, como ilustrado na [Figura 19](#) para $r = 0,2$, $r = 0,25$ e $r = 0,4$.

³Como será discutido neste capítulo, a modelagem do problema pela Lambert-Tsallis relaciona z aos parâmetros da planta e do controlador, que nesta tese assumem apenas valores reais. Embora existam condições para $z \in \mathbb{C}$ resultar em parâmetros reais, esta não é a regra. A análise de $z \in \mathbb{C}$ será realizada em um exemplo demonstrativo.

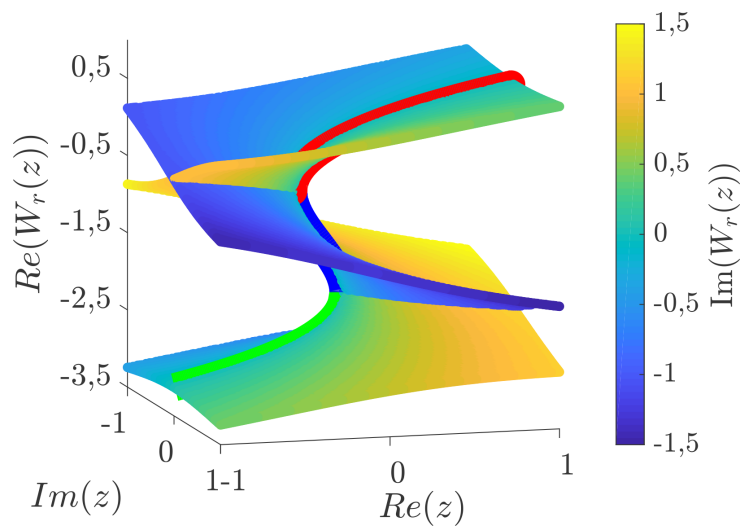
Figura 16 – Superfícies de Riemann de $W_r(z) \times z \subset \mathbb{C} \times \mathbb{C}$, $r \in \mathbb{Q}_+^*$.



(a) Tipo 1, $r = 1/2$.



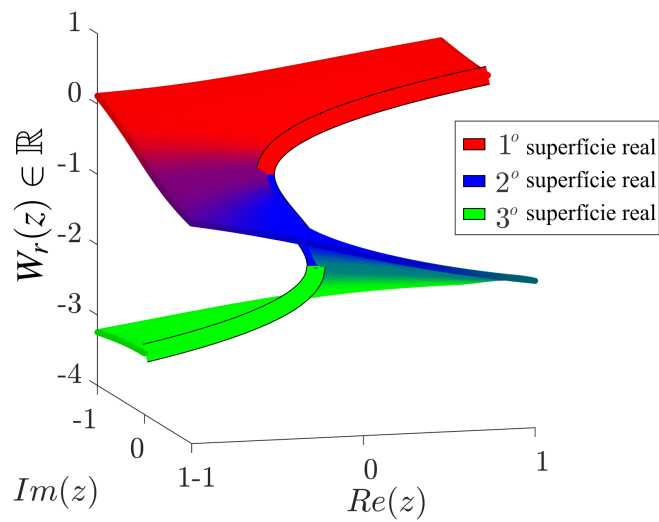
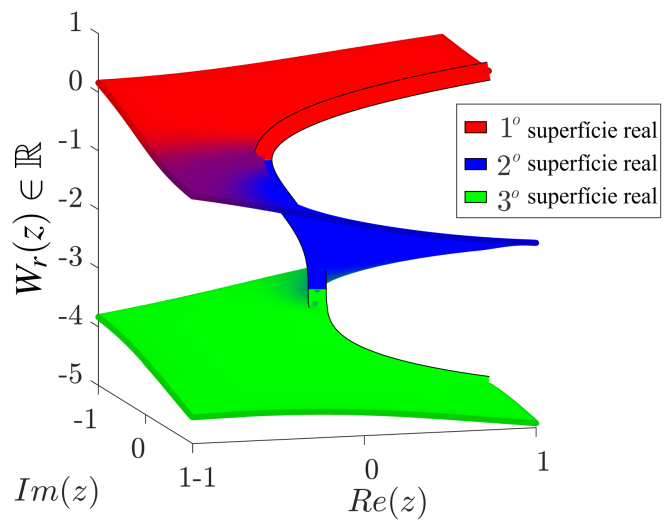
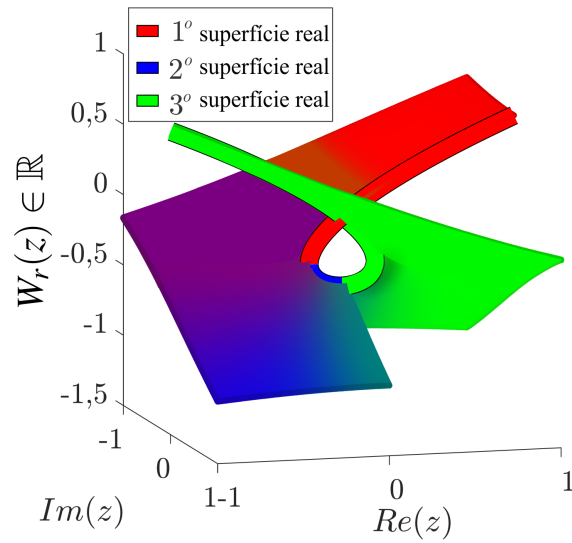
(b) Tipo 2, $r = 3/1$.



(c) Tipo 3, $r = 2/1$.

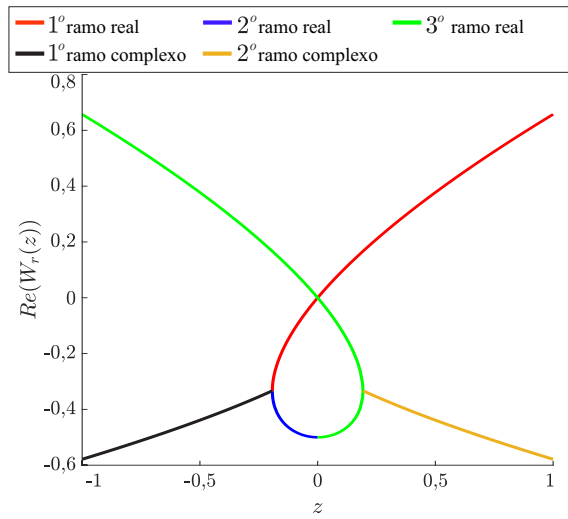
Fonte: elaborada pelo autor.

Figura 17 – Superfícies de Riemann de $W_r(z) \times z \subset \mathbb{R} \times \mathbb{C}$, $r \in \mathbb{Q}_+^*$.

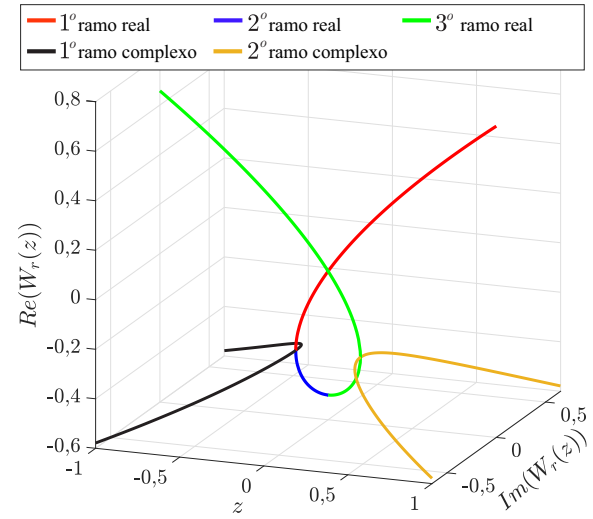


Fonte: elaborada pelo autor.

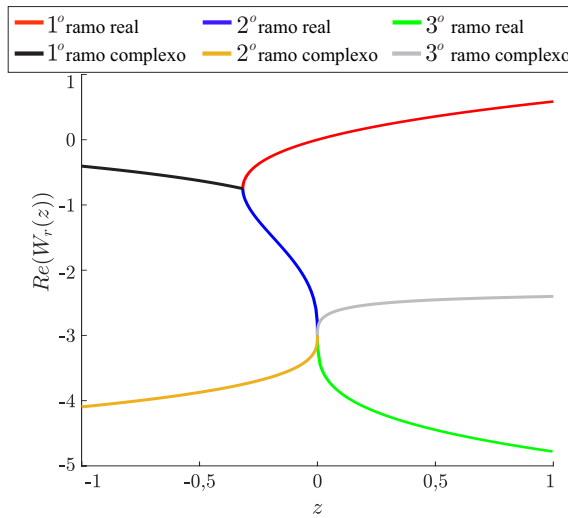
Figura 18 – Diferentes gráficos de $W_r(z) \times z \subset \mathbb{C} \times \mathbb{R}$, $r \in \mathbb{Q}_+^*$.



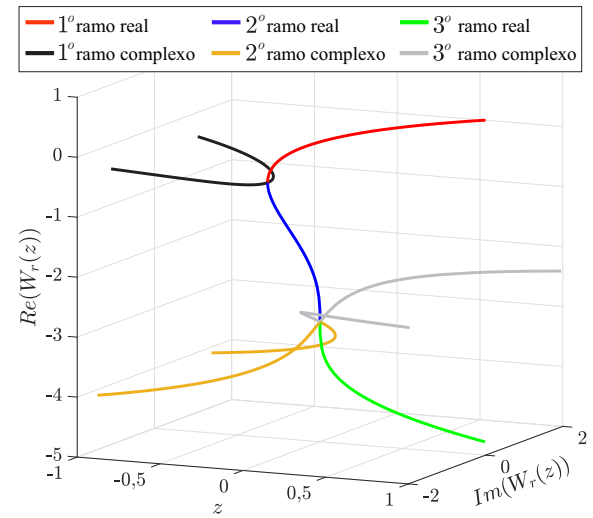
(a) Tipo 1, $r = 1/2$, vista lateral.



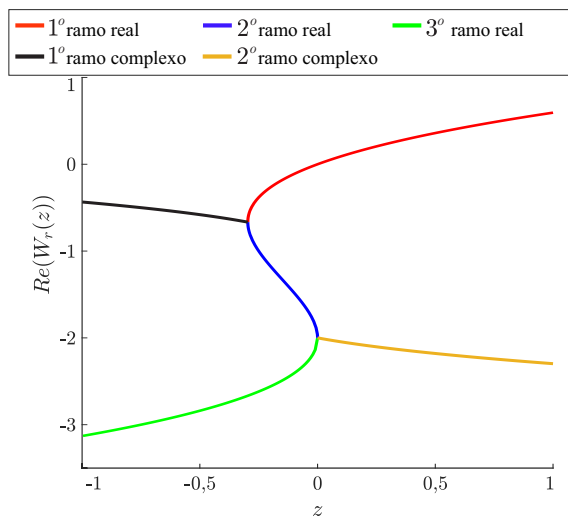
(b) Tipo 1, $r = 1/2$, vista tridimensional.



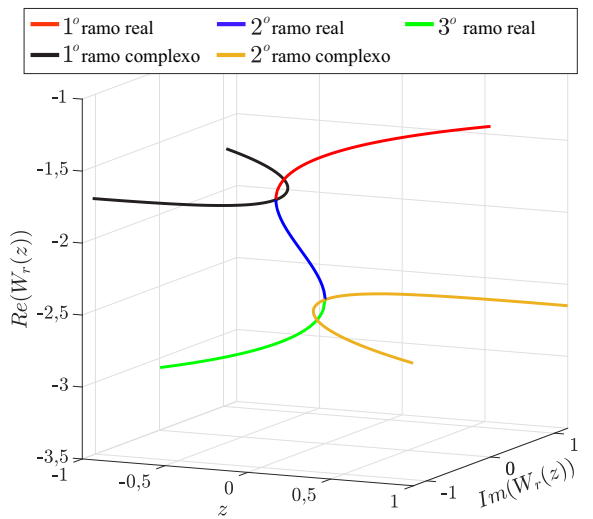
(c) Tipo 2, $r = 3/1$, vista lateral.



(d) Tipo 2, $r = 3/1$, vista tridimensional.



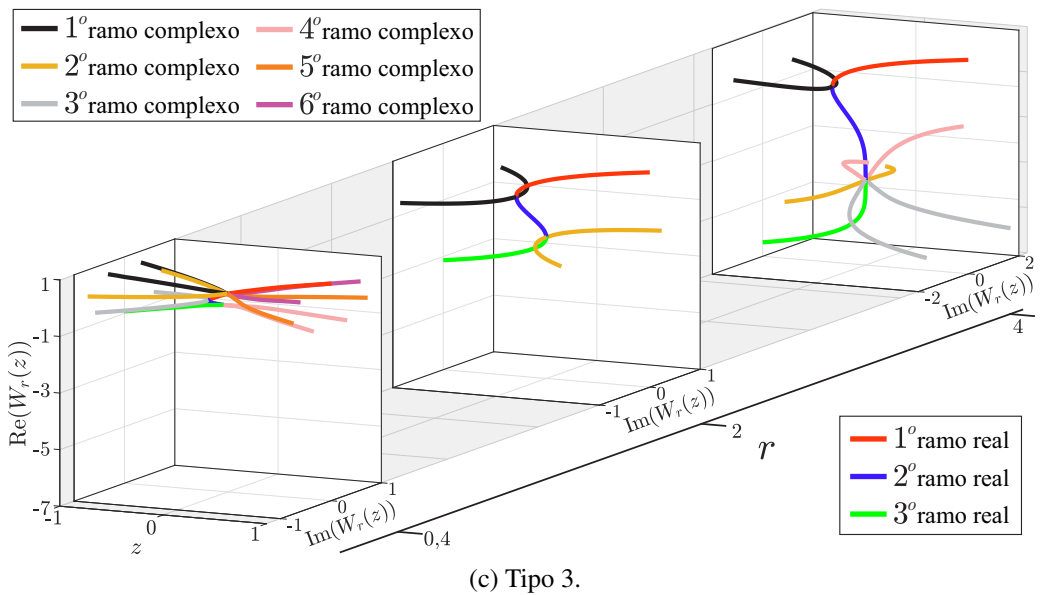
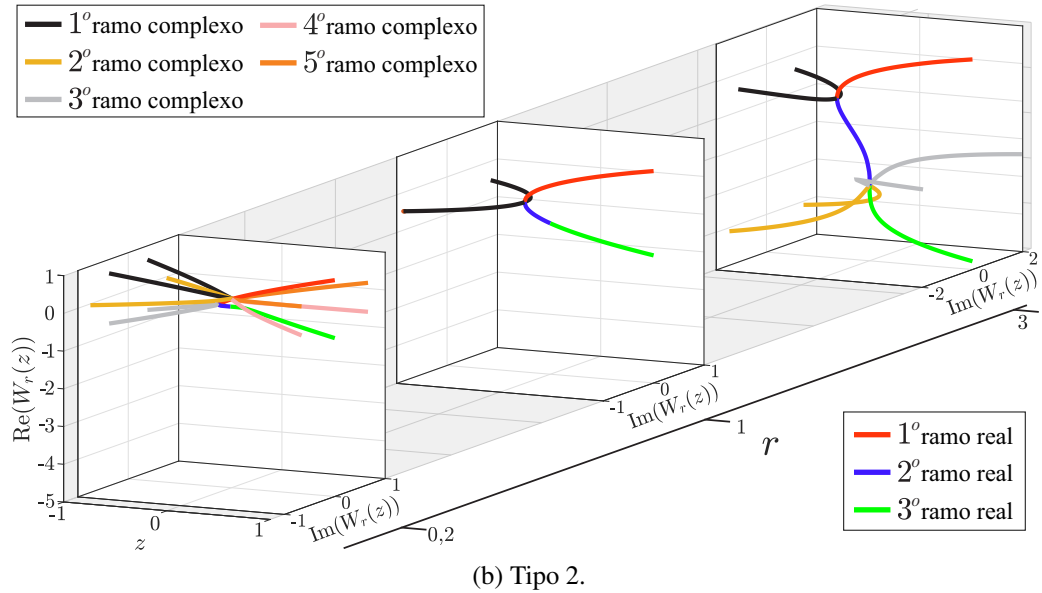
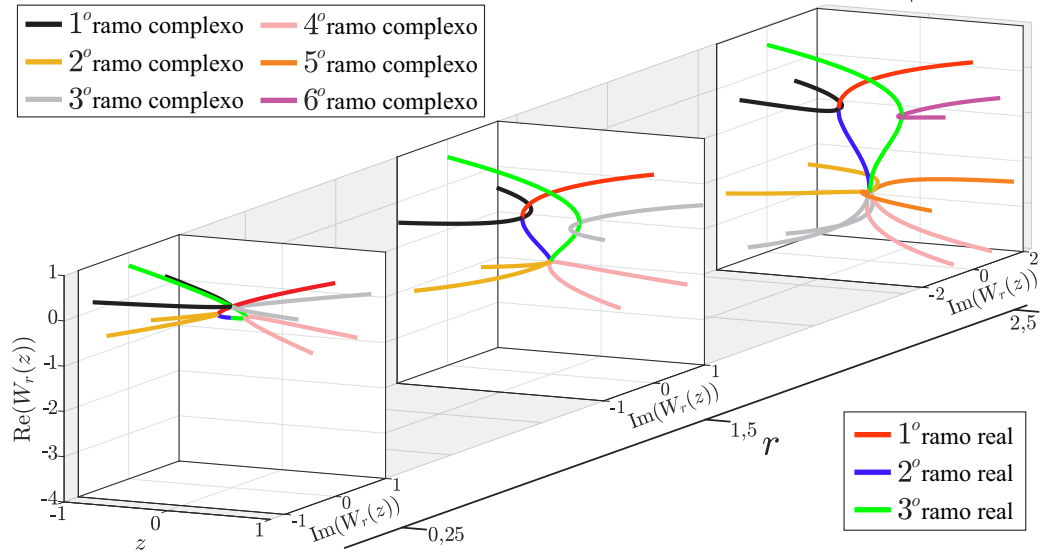
(e) Tipo 3, $r = 2/1$, vista lateral.



(f) Tipo 3, $r = 2/1$, vista tridimensional.

Fonte: elaborada pelo autor.

Figura 19 – Evolução de $W_r(z) \times z \subset \mathbb{C} \times \mathbb{R}$ para diferentes valores de $r \in \mathbb{Q}_+^*$.



Fonte: elaborada pelo autor.

4 ESTABILIDADE DE SISTEMAS FRACIONÁRIOS: UMA NOVA PROPOSTA

Neste capítulo, aborda-se novamente a questão da estabilidade de sistemas fracionários, desta vez com a proposição de um método alternativo ao Teorema de Matignon para uma classe de sistemas fracionários, a saber, sistemas que podem ser descritos por meio de funções de transferência cujos polinômios característicos são trinômios. É apresentada a solução analítica de equações trinômiais generalizadas na [Seção 4.1](#). A proposta de análise de estabilidade de sistemas de ordem fracionária é apresentada com exemplos de aplicação na [Seção 4.2](#), juntamente com um método analítico para calcular os limites dos parâmetros do controlador que garantam a estabilidade. Por fim, considerações sobre a proposta apresentada são expostas na [Seção 4.3](#).

4.1 Solução analítica de equações trinômiais generalizadas

Nesta seção, é demonstrada a capacidade da função Lambert-Tsallis de expressar soluções analíticas para equações trinômiais sem depender de séries de potências, restrições de convergência ou intervalos específicos de argumentos, como em [Glasser \(2000\)](#) e em [Belkić \(2019\)](#). Desta forma, pode-se lidar diretamente com trinômios de ordem fracionária em seu domínio original, sem a necessidade de transformações polinomiais.

Seja o trinômio

$$a_n s^n + a_m s^m + a_0 = 0, \quad (4.1)$$

$\forall n, m \in \mathbb{Q}^*$ com $m < n$ e $a_n a_m a_0 \neq 0$. Para demonstrar a solução do trinômio pela função Lambert-Tsallis, primeiramente isola-se a_0 , destaca-se a potência do segundo monômio, e aplica-se a [Propriedade 3.1](#), *i.e.*,

$$a_n s^n + a_m s^m = -a_0 \quad (4.2)$$

$$a_m s^m \left(1 + \frac{a_n}{a_m} s^{n-m} \right) = -a_0 \quad (4.3)$$

$$s^m \left(1 + \frac{a_n}{a_m} s^{n-m} \right) = -\frac{a_0}{a_m} \quad (4.4)$$

$$\therefore s^m \exp_1 \left(\frac{a_n}{a_m} s^{n-m} \right) = -\frac{a_0}{a_m}. \quad (4.5)$$

O próximo passo é estabelecer r , pois este será responsável por igualar as potências de s dentro e fora de $\exp_r(z)$. Isto é feito definindo

$$r = \frac{n-m}{m}. \quad (4.6)$$

Neste momento, deve-se evitar substituir r por um valor n merico, pois a manipula  o simb lica facilita o uso das propriedades apresentadas. Uma vez que r esteja estabelecido, o pr ximo passo   elevar a [Equa  o 4.5](#) ao expoente r com o aux lio da [Propriedade 3.2](#) resultando em

$$\left[s^m \exp_1 \left(\frac{a_n}{a_m} s^{n-m} \right) \right]^r = \left(-\frac{a_0}{a_m} \right)^r \quad (4.7)$$

$$s^{mr} \exp_r \left(r \frac{a_n}{a_m} s^{n-m} \right) = \left(-\frac{a_0}{a_m} \right)^r. \quad (4.8)$$

Pode-se perceber, na [Equa  o 4.8](#), que ambos os expoentes de s s o equivalentes, visto que $r = \frac{n-m}{m}$. Assim, a  nica diferen a entre estes termos   a constante que os acompanha. O pr ximo passo consiste em igualar esta constante, o que pode ser feito com a multiplica  o de $r \frac{a_n}{a_m}$ em ambos os lados da equa  o,

$$\underbrace{r \frac{a_n}{a_m}}_{W_r(z)} \underbrace{s^{mr} \exp_r \left(r \frac{a_n}{a_m} s^{n-m} \right)}_{\exp_r[W_r(z)]} = \underbrace{r \frac{a_n}{a_m} \left(-\frac{a_0}{a_m} \right)^r}_z. \quad (4.9)$$

Observa-se o paralelo tra ado com a equa  o de Lambert-Tsallis, descrita na [Equa  o 3.5](#) como $W_r(z) \exp_r[W_r(z)] = z$. Al m disso, nota-se uma rela  o entre s e $W_r(z)$ com

$$z = r \frac{a_n}{a_m} \left(-\frac{a_0}{a_m} \right)^r. \quad (4.10)$$

No entanto, dependendo de como a express o final de s for escrita, propriedades interessantes ou caracter sticas importantes sobre a natureza de s podem ser deixadas para tr s. Tendo isso em mente, sugere-se escrever a [Equa  o 4.9](#) de tr s maneiras diferentes,

$$s^{mr} = \frac{W_r(z)}{r \frac{a_n}{a_m}}, \quad (4.11)$$

$$s^m = \frac{-\frac{a_0}{a_m}}{1 + \frac{W_r(z)}{r}}, \quad (4.12)$$

$$s^n = \frac{-\frac{a_0}{a_n} \frac{W_r(z)}{r}}{1 + \frac{W_r(z)}{r}}. \quad (4.13)$$

Se as condições $m = 1$ ou $n = 1$ forem atendidas na [Equação 4.12](#) ou [Equação 4.13](#), respectivamente, s é facilmente determinado a partir de uma relação direta com $W_r(z)$ através de operações matemáticas básicas, o que facilita a análise de aspectos relacionados a s . Para os demais casos, recomenda-se o uso da [Equação 4.11](#).

Em resumo, a abordagem desenvolvida nesta seção não apenas descreve as etapas e estratégias para trabalhar com a função Lambert-Tsallis a partir da equação inicial, mas também revela que a forma como a solução final é escrita pode impactar significativamente as informações extraídas. Além disso, o estudo apresentado nesta seção demonstrou que é possível estabelecer uma relação entre os argumentos de $W_r(z)$, a saber, r e z (vide a [Equação 4.10](#)), que pode revelar características importantes do problema a ser trabalhado, o que será visto a seguir. Assim, a função Lambert-Tsallis deve ser considerada mais do que apenas uma ferramenta analítica para resolver uma equação trinomial de ordem fracionária, mas uma ferramenta capaz de fornecer *insights* sobre a variável e os parâmetros da equação que são fundamentais para entender a natureza do problema. Por fim, ressalta-se que esta é a primeira ferramenta capaz de solucionar a equação trinomial generalizada para parâmetros quaisquer e independentes entre si.

4.2 Análise de estabilidade de sistemas pela função $W_r(z)$

Para utilizar a Lambert-Tsallis na análise de estabilidade de sistemas de controle de ordem fracionária, a equação característica deve ser um trinômio. Portanto, para uma estrutura de controle típica, conforme mostrado na [Figura 3](#), deve-se utilizar uma planta, um controlador e um sensor que atendam a esse requisito. Casos para os quais essa restrição é satisfeita estão detalhados na [Tabela 6](#), considerando um típico sensor com ganho estático $H(s) = b$.

Tabela 6 – Exemplos de sistemas de controle adequados para análise via função Lambert-Tsallis.

caso	controlador de ordem inteira	planta de ordem fracionária	$Q(s)$	condições
1	$G_c(s) = k_p$	$G_p(s) = \frac{ds^\beta + c}{s^\alpha + a}$	$s^\alpha + bdk_p s^\beta + (a + bck_p)$	$\alpha, \beta \in \mathbb{Q}_+^*$ $a, b, c, d \in \mathbb{R}, k_p \in \mathbb{R}_+^*$ $\alpha > \beta$
2		$G_p(s) = \frac{c}{s^\beta + ds^\alpha + a}$	$s^\beta + ds^\alpha + (a + bck_p)$	$\alpha, \beta \in \mathbb{Q}_+^*$ $a, b, c, d \in \mathbb{R}, k_p \in \mathbb{R}_+^*$
3		$G_p(s) = \frac{es^\alpha + c}{s^\beta + ds^\alpha + a}$	$s^\beta + (d + bek_p)s^\alpha + (a + bck_p)$	$\alpha, \beta \in \mathbb{Q}_+^*$ $a, b, c, d, e \in \mathbb{R}, k_p \in \mathbb{R}_+^*$ $\beta \neq \alpha$
4	$G_c(s) = k_p + \frac{k_i}{s}$	$G_p(s) = \frac{c}{s^\alpha + a}$	$s^{1+\alpha} + (a + bck_p)s + bck_i$	$\alpha \in \mathbb{Q}_+^*$ $a, b, c \in \mathbb{R}, k_p, k_i \in \mathbb{R}_+^*$
5	$G_c(s) = k_p + k_d s$	$G_p(s) = \frac{c}{s^\alpha + a}$	$s^\alpha + bck_d s + (a + bck_p)$	$\alpha \in \mathbb{Q}_+^*$ $a, b, c \in \mathbb{R}, k_p, k_d \in \mathbb{R}_+^*$
6		$G_p(s) = \frac{c}{s + ds^\alpha + a}$	$ds^\alpha + (1 + bck_d)s + (a + bck_p)$	$\alpha \in \mathbb{Q}_+^*$ $a, b, c, d \in \mathbb{R}, k_p, k_d \in \mathbb{R}_+^*$
	controlador de ordem fracionária	planta de ordem inteira	$Q(s)$	condições
7	$G_c(s) = k_p + \frac{k_i}{s^\lambda}$	$G_p(s) = \frac{c}{s^n + a}$	$s^{n+\lambda} + (a + bck_p)s^\lambda + bck_i$	$\lambda \in \mathbb{Q}_+^*$ $n \in \mathbb{Z}_+^*$ $a, b, c \in \mathbb{R}, k_p, k_i \in \mathbb{R}_+^*$
8	$G_c(s) = k_p + k_d s^\mu$	$G_p(s) = \frac{c}{s^n + a}$	$s^n + bck_d s^\mu + (a + bck_p)$	$\mu \in \mathbb{Q}_+^*$ $n \in \mathbb{Z}_+^*$ $a, b, c \in \mathbb{R}, k_p, k_d \in \mathbb{R}_+^*$
	controlador de ordem fracionária	planta de ordem fracionária	$Q(s)$	condições
9	$G_c(s) = k_p + \frac{k_i}{s^\lambda}$	$G_p(s) = \frac{c}{s^\alpha + a}$	$s^{\lambda+\alpha} + (a + bck_p)s^\lambda + bck_i$	$\alpha, \lambda \in \mathbb{Q}_+^*$ $a, b, c \in \mathbb{R}, k_p, k_i \in \mathbb{R}_+^*$
10	$G_c(s) = k_p + k_d s^\mu$	$G_p(s) = \frac{c}{s^\alpha + a}$	$s^\alpha + bck_d s^\mu + (a + bck_p)$	$\alpha, \mu \in \mathbb{Q}_+^*$ $a, b, c \in \mathbb{R}, k_p, k_d \in \mathbb{R}_+^*$
11	$G_c(s) = k_p + k_d s^\alpha$	$G_p(s) = \frac{c}{s^\beta + ds^\alpha + a}$	$s^\beta + (d + bck_d)s^\alpha + (a + bck_p)$	$\alpha, \beta \in \mathbb{Q}_+^*$ $a, b, c, d \in \mathbb{R}, k_p, k_d \in \mathbb{R}_+^*$
12	$G(s) = k_p + \frac{k_i}{s^\lambda} + k_d s^\alpha$	$G_p(s) = \frac{c}{s^\alpha + a}$	$(1 + bck_d)s^{\lambda+\alpha} + (a + bck_p)s^\lambda + bck_i$	$\alpha, \lambda \in \mathbb{Q}_+^*$ $a, b, c \in \mathbb{R}, k_p, k_i, k_d \in \mathbb{R}_+^*$
13	$G_c(s) = \frac{ds^\alpha + c}{s^\alpha + a}$	$G_p(s) = \frac{e}{s^\alpha + f}$	$s^{2\alpha} + (a + bd + f)s^\alpha + (af + bce)$	$\alpha \in \mathbb{Q}_+^*$ $a, b, c, d, e, f \in \mathbb{R}$
14		$G_p(s) = \frac{es^\alpha + g}{s^\alpha + f}$	$(1 + bde)s^{2\alpha} + (a + f + bdg + bce)s^\alpha + (bcg + af)$	$\alpha \in \mathbb{Q}_+^*$ $a, b, c, d, e, f, g \in \mathbb{R}$

Os casos 1 a 6 utilizam controladores tradicionais de ordem inteira, como P, PI e PD. Os casos 7 a 10 utilizam controladores de ordem fracionária como PI^λ e PD^μ . Os casos 11 e 12, utilizam PD^μ e $PI^\lambda D^\mu$ em que a ordem de derivação é imposta como a mesma da planta de ordem fracionária, α . Por fim, os casos 13 e 14 utilizam um controlador por avanço ou atraso de fase de ordem fracionária, em que a ordem do controlador é imposta como sendo a mesma da planta fracionária.

Sistemas de controle cujas equações características $Q(s)$ são trinômios, mas não possuem ordem fracionária (nos quais plantas e controladores de ordem inteira são utilizados), não estão destacados na tabela. Embora tais casos também possam ser resolvidos utilizando a função Lambert-Tsallis, esses casos não são discutidos, pois sistemas de controle de ordem inteira não estão no escopo desta tese.

Para exemplificar a nova abordagem proposta, é utilizado como exemplo, o sistema de controle de uma planta de aquecimento elétrico descrita em [Petráš \(2011\)](#), o qual se enquadra no caso 5 da [Tabela 6](#).

Exemplo 4.2.1: Encontrando as raízes da equação característica de ordem fracionária

Neste exemplo, a equação característica é resolvida utilizando o teorema de [Matignon \(1996\)](#) (vide [Seção 2.5](#)) e pelo método proposto, que utiliza a função Lambert-Tsallis. As funções de transferência da planta $G_{p1}(s)$ e do controlador PD $G_{c1}(s)$ são dadas por

$$G_{p1}(s) = \frac{1}{39,96s^{1,25} + 0,598} \quad (4.14)$$

$$G_{c1}(s) = 64,47 + 12,46s, \quad (4.15)$$

tal que a função de transferência de malha fechada $G_{cl1}(s)$ é dada por

$$G_{cl1}(s) = \frac{12,46s + 64,47}{39,96s^{1,25} + 12,46s + 65,068}, \quad (4.16)$$

considerando que a realimentação em malha fechada tem ganho unitário ($b = 1$).

Aplicando o teorema de Matignon, escolhe-se $\gamma = 4$ com base no procedimento descrito na [Seção 2.5](#). Deve-se atentar ao surgimento de polos falsos devido à transformação $\sigma = s^{\frac{1}{\gamma}}$. Esses polos falsos satisfazem $Q(\sigma) = 0$, mas não satisfazem $Q(s) = 0$. Somente os polos genuínos encontrados na primeira folha de Riemann satisfazem $Q(s) = 0$. De acordo com o teorema de Matignon, todos os polos de $Q(\sigma)$ satisfazem a condição descrita na [Equação 2.34](#),

o que torna o sistema de controle da [Equação 4.16](#) estável. Para encontrar as raízes genuínas no plano s , deve-se aplicar $s = \sigma^\gamma$ aos valores de σ_i presentes na primeira folha de Riemann.

A proposta de solução do problema utilizando a função Lambert-Tsallis é detalhada no procedimento a seguir. Pelo princípio da identidade de polinômios entre a equação característica de $G_{cl1}(s)$ e a [Equação 4.1](#), conclui-se que $a_n = 39,96$, $a_m = 12,46$, $a_0 = 65,068$, $m = 1$ e $n = 1,25$. Aplicando a [Equação 4.6](#) e a [Equação 4.10](#), resulta que $r = \frac{1}{4}$, e $z = 0,8570 + 0,8570i$ ¹. A partir destes dois parâmetros obtidos, r e z , os valores de $W_r(z)$ são obtidos através da aplicação do [Algoritmo 1](#), que retorna

$$W_r(z) = \begin{bmatrix} -0,9402 \\ -0,3243 + 0,8349i \\ -0,3243 - 0,8349i \\ 0,6694 + 0,5167i \\ 0,6694 - 0,5167i \end{bmatrix}. \quad (4.17)$$

Embora tenha-se $m = 1$, o que significa que é possível obter s diretamente da [Equação 4.12](#), será utilizada a [Equação 4.11](#), para efeitos didáticos, uma vez que esta é a versão mais apropriada para quaisquer valores de $m, n > 0$. Assim,

$$s = \left(\frac{W_r(z)}{r \frac{a_n}{a_m}} \right)^{\frac{1}{mr}} \quad (4.18)$$

$$s = \left(\frac{W_r(z)}{0,25 \frac{39,96}{12,46}} \right)^{\frac{1}{0,25}}. \quad (4.19)$$

Porém, é importante entender que nem todos os valores de $W_r(z)$ levam a raízes genuínas de s , de forma semelhante ao que acontece com a transformação $\sigma = s^{\frac{1}{\gamma}}$. Os valores de $W_r(z)$ que levam a raízes genuínas de s , doravante denominados $W_r^*(z)$ e s^* , podem ser encontrados conforme o [Algoritmo 2](#).

¹Neste caso, $z \in \mathbb{C}$ devido ao valor de r ser do tipo 1, discussão já realizada na [Seção 3.2](#). No cálculo de polos do sistema com um dado $Q(s)$ trinomial, $z \in \mathbb{C}$ não se apresenta como um problema. Porém, na solução analítica para os limites dos parâmetros do controlador, proposta a ser apresentada na [Subseção 4.2.1](#), é preferível ter $z \in \mathbb{R}$ para evitar ganhos complexos, como mencionado no início da [Seção 3.3](#). Nos casos em que r é do tipo 1, recomenda-se a aplicação da [Propriedade 3.4](#) para utilizar a Lambert-Tsallis com $\frac{1}{r}$ do tipo 3.

Algoritmo 2: Determinação do(s) valor(es) de $W_r^*(z)$ e de s^* .

Dados: $m, r \in \mathbb{Q}_+^*$, $W_r(z) \in \mathbb{C}$, $Q(s)$
Resultado: $W_r^*(z)$, s^*
 $\frac{n}{d} \leftarrow r$;

 $j = 0$;

para $i = 1 : n + d$ **faça**

$$\text{Calcule } s_i = \left[\frac{W_{r_i}(z)}{r \frac{a_n}{a_m}} \right]^{\frac{1}{mr}} ;$$

 Calcule o resíduo $Q(s_i)$;

se $Q(s_i) == 0$ **então**
 $j = j + 1$;

 $W_{r_j}^*(z) = W_{r_i}(z)$;

 $s_j^* = s_i$;

fim
fim
retorna vetores $W_r^*(z)$ e s^* ;

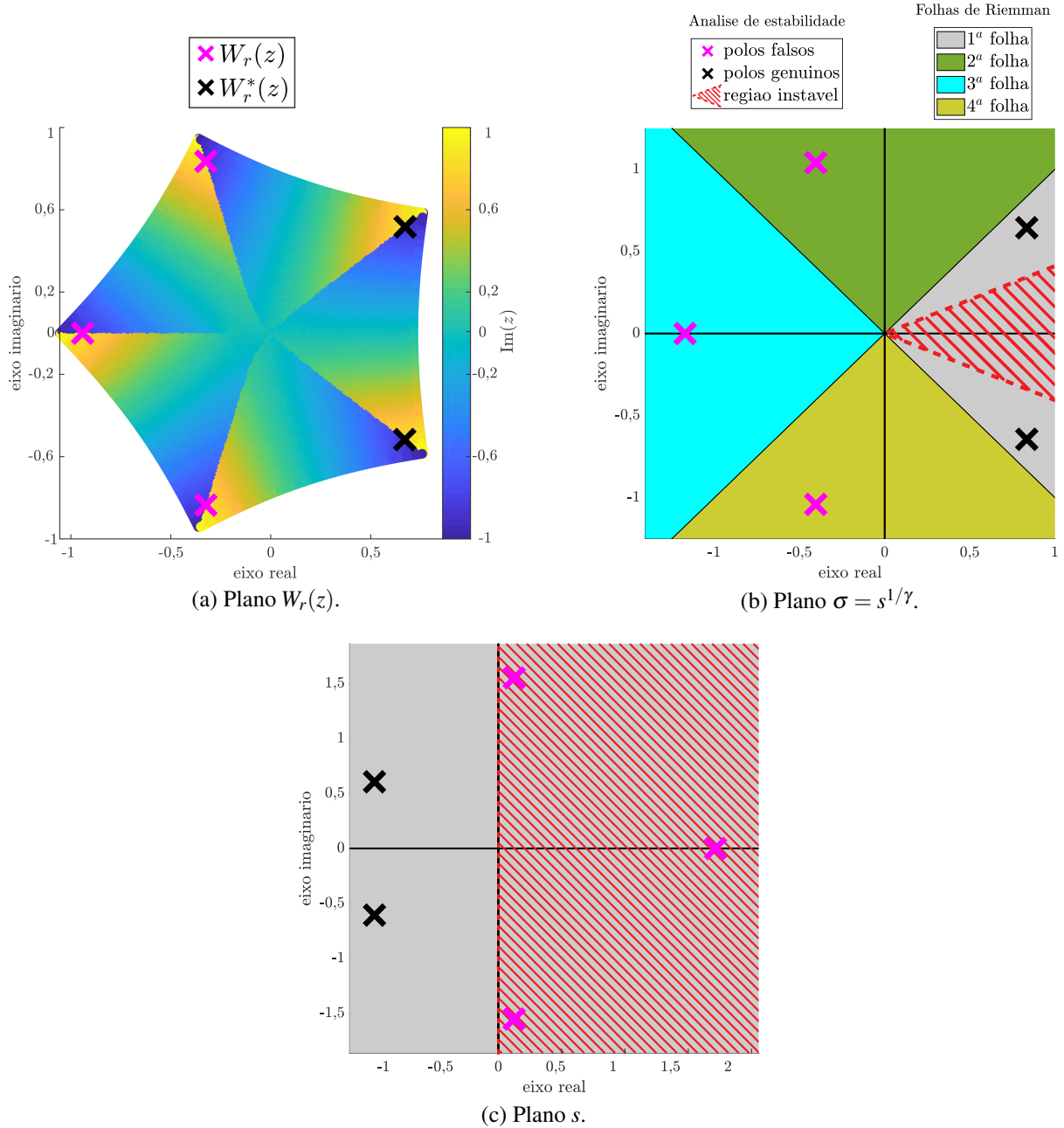
 Tabela 7 – Comparação entre os resultados obtidos para o [Exemplo 4.2.1](#).

Lambert-Tsallis		teorema de Matingon	
$W_r(z)$	s	$\sigma = s^{1/4}$	s
-0,9402	1,8914	-1,1727	1,8914
-0,3243 + 0,8349i	0,1381 + 1,5514i	-0,4045 + 1,0413i	0,1381 + 1,5514i
-0,3243 - 0,8349i	0,1381 - 1,5514i	-0,4045 - 1,0413i	0,1381 - 1,5514i
0,6694 + 0,5167i	-1,0791 + 0,6065i	0,8350 + 0,6445i	-1,0791 + 0,6065i
0,6694 - 0,5167i	-1,0791 - 0,6065i	0,8350 - 0,6445i	-1,0791 - 0,6065i

Na [Tabela 7](#) são apresentados os resultados de $W_r(z)$ e de s referentes à aplicação da Lambert-Tsallis, assim como os valores de σ e s referentes ao método de [Matignon \(1996\)](#). Os resultados em negrito representam os valores de s^* , de $W_r^*(z)$ e as raízes genuínas de $Q(\sigma)$, que levam aos valores genuínos de s . Os valores presentes na [Tabela 7](#) são ilustrados na [Figura 20](#).

O plano $W_r(z)$ presente na [Figura 20.a](#) foi construído a partir da superfície de Riemann composta pelos eixos x , y e z como, respectivamente a parte real de $W_r(z)$, a parte real de z e a parte imaginária de $W_r(z)$. A parte imaginária de z é representada em uma quarta dimensão de forma cromática. Nota-se a semelhança entre as posições de $W_r(z)$ e σ . De fato, ambos possuem os mesmos argumentos (ângulos), incluindo os valores de $W_r^*(z)$ e σ^* . Por fim, constatou-se que os resultados obtidos a partir dos dois métodos coincidem com precisão

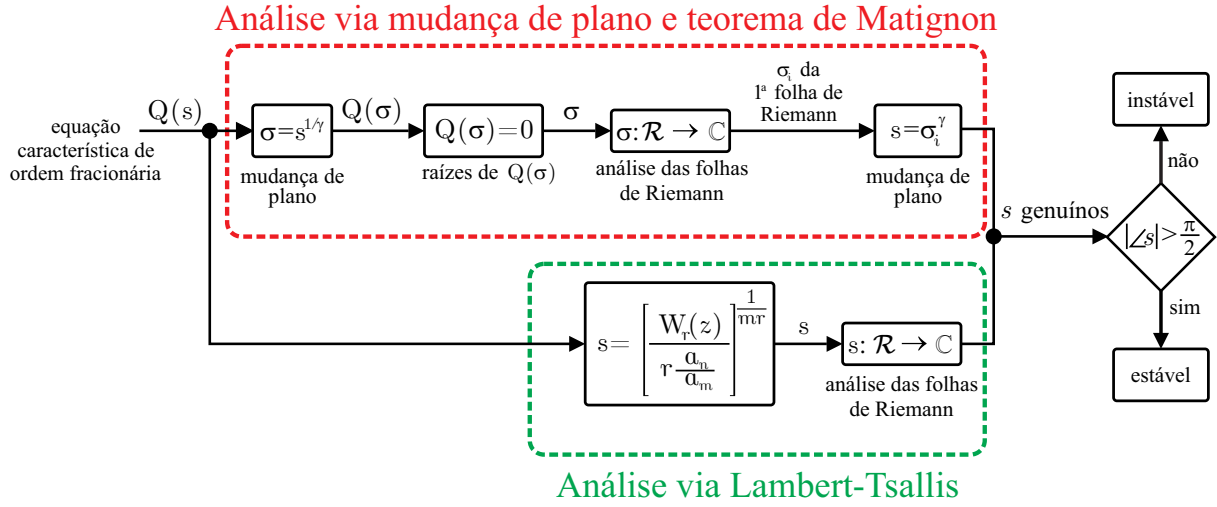
Figura 20 – Planos $W_r(z)$, s e σ para o Exemplo 4.2.1.



Fonte: elaborada pelo autor.

entre a 12ª e a 15ª casa decimal, sendo numericamente indistinguíveis. A Figura 21 resume a abordagem proposta enquanto também mostra as etapas intermediárias necessárias do método clássico de Matignon (1996). Com base neste exemplo e nesta figura, pode-se observar que a abordagem proposta envolve simplificações e menos etapas.

Figura 21 – Resumo do método proposto em comparação com o método clássico.



Fonte: elaborada pelo autor.

4.2.1 Solução analítica para os limites dos parâmetros do controlador via $W_r(z)$

A utilização da função Lambert-Tsallis em sistemas de controle de ordem fracionária não se resume apenas à solução de $Q(s)$ para determinar se um sistema é estável ou não. Visto que a ferramenta é analítica, é possível estender sua aplicação para determinar quais os limites dos ganhos do controlador que fazem o sistema ser estável.

Neste sentido, um sistema é dito estável se e somente se todas as raízes genuínas de seu polinômio característico estiverem à esquerda do plano s , o que significa que os módulos de seus argumentos devem ser maiores que $\frac{\pi}{2}$. Assim, dado que a solução de $Q(s)$ pela Lambert-Tsallis é analítica, é possível estabelecer condições suficientes para garantia de estabilidade.

A partir da [Equação 4.11](#), tem-se que

$$|\angle s^{mr}| = \left| \angle \frac{W_r(z)}{r \frac{a_n}{a_m}} \right| \quad (4.20)$$

$$|mr \angle s| = |\angle W_r(z) - 0| \quad (4.21)$$

$$|mr \angle s| = |\angle W_r(z)|. \quad (4.22)$$

Observa-se que o argumento de $r \frac{a_n}{a_m}$ é 0, uma vez que o resultado é um número real positivo (vide [Tabela 6](#)). Como o ângulo que define o limiar de estabilidade no plano s é $\frac{\pi}{2}$, consequentemente o valor crítico de $W_r(z)$ para a estabilidade é obtida quando

$$|\angle W_r(z)| = mr \frac{\pi}{2} = \theta_c, \quad (4.23)$$

em que θ_c é o ângulo crítico para a estabilidade no plano $W_r(z)$. Por consequência, este valor crítico de $W_r(z)$ está relacionado aos seus dois parâmetros, r e z , com r definido pelos expoentes fracionários da planta e do controlador, enquanto z , doravante denominado z crítico, z_c , é definido pela [Equação 4.10](#) como

$$z_c = r \frac{a_n}{a_m} \left(-\frac{a_0}{a_m} \right)^r$$

$$z_c = \frac{r a_n (-a_0)^r}{a_m^{r+1}}. \quad (4.24)$$

Uma vez que a [Tabela 6](#) define os valores de a_n , a_m e a_0 em relação aos parâmetros da planta e ganhos do controlador, tem-se uma poderosa ferramenta para calcular de forma analítica os limites e as relações entre tais parâmetros que garantem a estabilidade do sistema de controle. A partir da [Equação 4.24](#), pode-se isolar a_m de forma que, dados r , z_c , a_n e a_0 , o valor crítico de a_m , a_{m_c} , pode ser obtido como

$$a_{m_c} = \left(\frac{r a_n (-a_0)^r}{z_c} \right)^{\frac{1}{r+1}}. \quad (4.25)$$

O mesmo pode ser feito para o valor crítico de a_0 , a_{0_c} , para dado r , z_c , a_n e a_m ,

$$a_{0_c} = - \left(\frac{z_c a_m^{r+1}}{r a_n} \right)^{\frac{1}{r}}. \quad (4.26)$$

As três equações que definem z_c , a_{m_c} e a_{0_c} podem ser de grande utilidade para o projeto de controladores. Todavia, obter o valor de z_c não é uma tarefa trivial, uma vez que atualmente não existe uma maneira de calcular a inversa de $W_r(z)$, $W_r^{-1}(z)$, para se obter o valor de z_c a partir dos argumentos de $W_r(z_c)$. A seguir, introduz-se no [Algoritmo 3](#) a primeira abordagem para estimar z_c . É importante ressaltar que o algoritmo é direcionado à aplicação de sistemas de controle e que seu uso em outras aplicações deve ser adaptado. Nota-se que os coeficientes a_n , a_m , e a_0 não são necessários para a obtenção de z_c , apenas os valores dos expoentes n e m . De fato, z_c é uma propriedade destes expoentes e é a partir deste que podem ser traçadas relações entre tais coeficientes. A seguir, a aplicabilidade de z_c é explorada em três exemplos.

Exemplo 4.2.2: Condições de estabilidade para controlador PD e planta $G_{p2}(s)$

No [Exemplo 4.2.1](#), foi demonstrado que o sistema é estável, dados os ganhos k_p e k_d . Utilizando a mesma planta e tipo de controlador, o propósito deste exemplo é ir além de

Algoritmo 3: Determinação do valor de z_c .

Dados: $m, n \in \mathbb{Q}_+$
Resultado: z_c

 Calcule $r \leftarrow \frac{n-m}{m}$;

 Calcule $\theta_c \leftarrow mr \frac{\pi}{2}$;

 Inicialize $z \leftarrow z_0$;

enquanto $|\arg[W_r^*(z)]| > \theta_c$ **faça**

 Incrementar z ;

 Calcular os valores de $W_r(z)$ através do [Algoritmo 1](#);

 Filtrar $W_r^*(z)$ através do [Algoritmo 2](#);

fim
retorna z ;

descobrir se o sistema é ou não estável, mas calcular analiticamente limitantes para k_p e k_d que garantem que o sistema é estável. A planta é reescrita de forma a corresponder ao caso 5 da [Tabela 6](#), de forma que

$$G_{p2}(s) = \frac{1}{39,96s^{1,25} + 0,598} = \frac{0,025}{s^{1,25} + 0,015} . \quad (4.27)$$

Para realizar a análise, será utilizado o controlador PD, $G_{c2}(s)$, dado por

$$G_{c2}(s) = k_p + k_d s . \quad (4.28)$$

Assim, a função de transferência de malha fechada $G_{cl2}(s)$ é

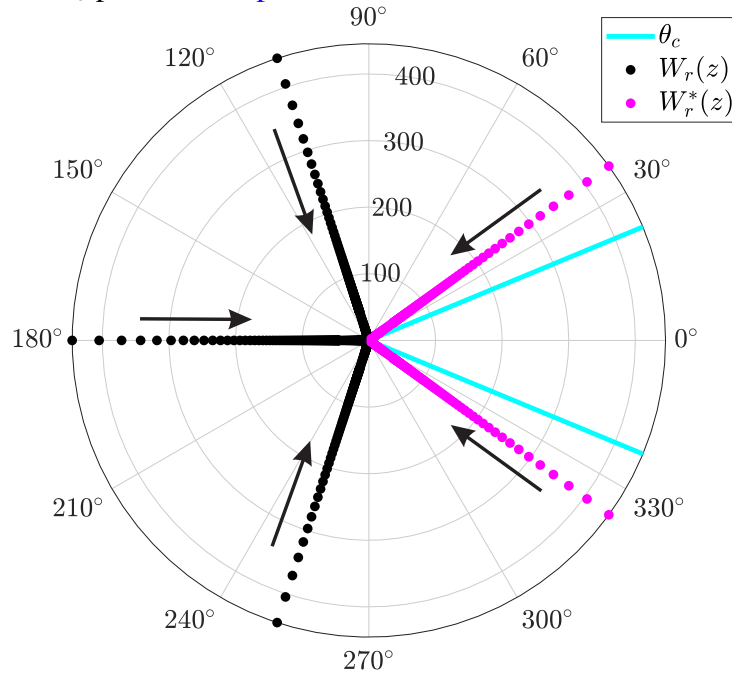
$$G_{cl2}(s) = \frac{0,025k_d s + 0,025k_p}{s^{1,25} + 0,025k_d s + (0,015 + 0,025k_p)} . \quad (4.29)$$

Pelo princípio da identidade de polinômios entre a equação característica de $G_{cl2}(s)$ e a [Equação 4.1](#), conclui-se que

$$\begin{aligned} n &= 1,25 \\ m &= 1 \\ a_n &= 1 \\ a_m &= 0,025k_d \\ a_0 &= 0,015 + 0,025k_p . \end{aligned} \quad (4.30)$$

A partir dos valores de n e m aplicados ao [Algoritmo 3](#) para a determinação do z crítico, obtém-se $z_c = 0$. O gráfico do processo é ilustrado na [Figura 22](#). As setas indicam o sentido dos valores de $W_r^*(z)$ e $W_r(z)$ à medida que se incrementa o valor de z . Percebe-se que

Figura 22 – Busca de z_c para o Exemplo 4.2.2.



Fonte: elaborada pelo autor.

$W_r^*(z)$ só toca a reta do ângulo crítico na origem $\{0,0\}$, o que resulta em $z_c = 0$. Aplicando $r = 0,25$, conforme a Equação 4.6, e os valores presentes na Equação 4.30 na Equação 4.24 que define z_c , tem-se que

$$0 = \frac{0,25(-0,015 - 0,025k_p)^{0,25}}{(0,025k_d)^{1,25}}. \quad (4.31)$$

Como o membro direito da equação trata-se de uma fração, isto significa que as únicas possibilidades de um resultado de valor nulo é se o denominador tender ao infinito, o que só é possível se $k_d \rightarrow \infty$, ou se o numerador assumir o valor zero. Contudo, é importante ressaltar que para zerar o numerador, o fator “ $-0,015 - 0,025k_p$ ” deve ser zero, o que implica que k_p precisa assumir valores negativos, violando as restrições da Tabela 6. Estes resultados também podem ser alcançados com a aplicação da Equação 4.25 e Equação 4.26. Em relação a a_{m_c} ,

$$a_{m_c} = \left[\frac{0,25(-0,015 - 0,025k_p)^{0,25}}{0} \right]^{\frac{1}{1,25}}. \quad (4.32)$$

A presença do valor zero no denominador significa que $a_{m_c} \rightarrow \infty$. Como $am = 0,025k_d$ (vide Equação 4.30), conclui-se que $k_d \rightarrow \infty$. Para a_{0_c} ,

$$a_{0_c} = - \left[\frac{0(0,025k_d)^{1,25}}{0,25} \right]^{\frac{1}{0,25}}. \quad (4.33)$$

A presença do valor zero no numerador faz $a_{0_c} = 0$. Como $a_0 = 0,015 + 0,025k_p$ (vide Equação 4.30), conclui-se que $k_p = -0,6$, o que viola as restrições da Tabela 6. Conclui-se

que o sistema será sempre estável para qualquer valor de $k_p, k_d > 0$. De fato, a aplicação do teorema de [Matignon \(1996\)](#) (vide [Seção 2.5](#)) comprova que para diferentes valores de k_p e k_d , incluindo valores elevados, o sistema é sempre estável.

Exemplo 4.2.3: Condições de estabilidade para controlador PD e planta $G_{p3}(s)$

Neste exemplo, a planta utilizada anteriormente é modificada para ser naturalmente instável. Posteriormente, limites de k_p e k_d que garantem que o sistema é estável são calculados analiticamente. Para isso, é trocado o sinal de um parâmetro da planta,

$$G_{p3}(s) = \frac{1}{39,96s^{1,25} - 0,598} = \frac{0,025}{s^{1,25} - 0,015}. \quad (4.34)$$

A função de transferência de malha fechada $G_{cl3}(s)$ é

$$G_{cl3}(s) = \frac{0,025k_d s + 0,025k_p}{s^{1,25} + 0,025k_d s + (-0,015 + 0,025k_p)}. \quad (4.35)$$

Pelo princípio da identidade de polinômios entre a equação característica de $G_{cl3}(s)$ e a [Equação 4.1](#), conclui-se que $n = 1,25$, $m = 1$, $a_n = 1$, $a_m = 0,025k_d$ e $a_0 = -0,015 + 0,025k_p$. Observa-se que r é o mesmo do exemplo anterior, pois n ou m não foram modificados. Consequentemente, z crítico também é o mesmo, ou seja, $z_c = 0$, uma vez que z_c só depende de m e r (vide [Algoritmo 3](#)). A única alteração acontece em a_0 , uma vez que o sinal do parâmetro da planta foi trocado. Aplicando os parâmetros obtidos na [Equação 4.25](#), chega-se a

$$a_{m_c} = \left[\frac{0,25(+0,015 - 0,025k_p)^{0,25}}{0} \right]^{\frac{1}{1,25}}, \quad (4.36)$$

o que, novamente, indica $k_d \rightarrow \infty$. O cálculo de a_{0_c} na [Equação 4.26](#),

$$a_{0_c} = - \left[\frac{0(0,025k_d)^{1,25}}{0,25} \right]^{\frac{1}{0,25}}, \quad (4.37)$$

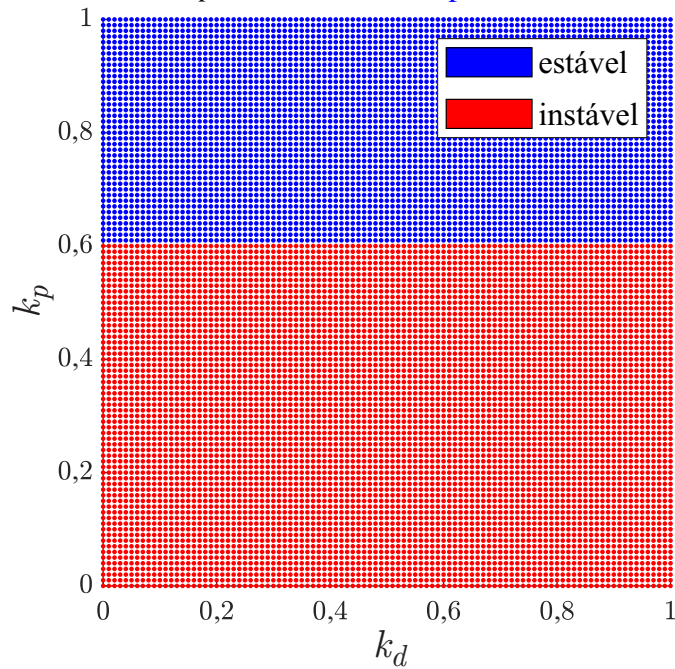
é o mesmo do exemplo anterior, também resultando em $a_{0_c} = 0$. Porém, desta vez, tem-se

$$-0,015 + 0,025k_p = a_{0_c} \quad (4.38)$$

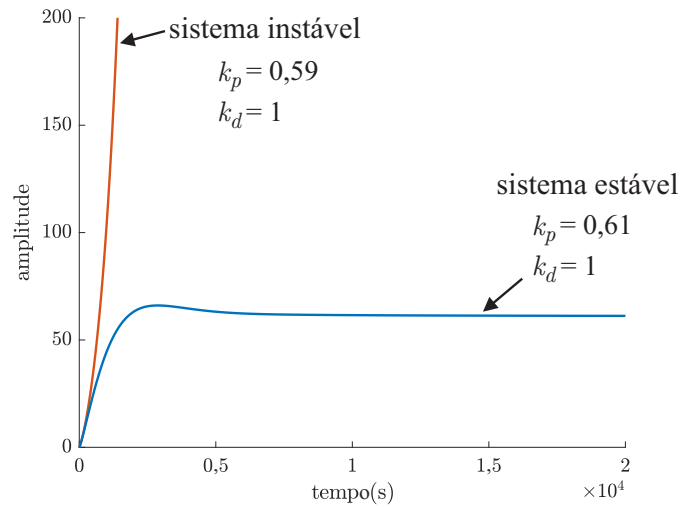
$$-0,015 + 0,025k_p = 0, \quad (4.39)$$

o que resulta em um limiar em $k_p = 0,6$. Para validar estes resultados, é realizada uma verificação *a posteriori* utilizando o teorema de [Matignon \(1996\)](#) (vide [Seção 2.5](#)) no plano $k_p \times k_d$ e a

Figura 23 – Teste de estabilidade *a posteriori* do Exemplo 4.2.3.



(a) $k_p \times k_d$ com $k_p, k_d \in [0, 1]$ e passo de 0,01.



(b) Respostas ao degrau do sistema de controle de malha fechada para $k_d = 1$ e $k_p \in \{0,59, 0,61\}$.

Fonte: elaborada pelo autor.

análise da resposta temporal, cujos resultados são ilustrados na Figura 23 e validam os limites obtidos com o método proposto.

Percebe-se que no Exemplo 4.2.2 e Exemplo 4.2.3, k_p e k_d não possuem relação entre si, sendo independentes. No próximo exemplo, é abordado um caso diferente da Tabela 6, de forma a expandir ainda mais as possibilidades de aplicação da Lambert-Tsallis em sistema de controle de ordem fracionária e do conceito de z_c .

Exemplo 4.2.4: Condições de estabilidade para controlador PI^λ com planta $G_{p2}(s)$

Neste exemplo, é utilizada a planta $G_{p2}(s)$ do [Exemplo 4.2.2](#), porém com um controlador PI de ordem fracionária, denotado por PI^λ , cuja função de transferência é

$$G_{c3}(s) = k_p + \frac{K_i}{s^\lambda}. \quad (4.40)$$

Desta forma, este exemplo encaixa-se no caso 9 da [Tabela 6](#), com função de transferência de malha fechada $G_{cl4}(s)$, dada por

$$G_{cl4}(s) = \frac{0,025k_p s^\lambda + 0,025k_i}{s^{1,25+\lambda} + (0,015 + 0,025k_p)s^\lambda + 0,025k_i}. \quad (4.41)$$

Pelo princípio da identidade de polinômios entre a equação característica de $G_{cl4}(s)$ e a [Equação 4.1](#), conclui-se que $n = 1,25 + \lambda$, $m = \lambda$, $a_n = 1$, $a_m = 0,015 + 0,025k_p$ e $a_0 = 0,025k_i$. Aplicando a [Equação 4.6](#), tem-se que

$$\begin{aligned} r &= \frac{n-m}{m} \\ r &= \frac{1,25}{\lambda}. \end{aligned} \quad (4.42)$$

Uma vez que o valor de r não é definido, não é possível obter um valor para z_c . Como então obter os valores limites dos parâmetros do controlador para λ qualquer? Neste sentido, percebe-se que para cada valor de λ , existe um valor de r e, conseqüentemente, um valor de z_c . Desenvolvendo as [Equações 4.25](#) e [4.26](#) voltadas para o caso 9 da [Tabela 6](#), tem-se os valores críticos de k_p e k_i dados por

$$k_{pc} = \frac{1}{bc} \left\{ \left[\frac{\alpha(-bck_i)^{\frac{\alpha}{\lambda}}}{\lambda z_c} \right]^{\frac{\lambda}{\alpha+\lambda}} - a \right\}, \quad (4.43)$$

$$k_{ic} = -\frac{1}{bc} \left[\frac{z_c \lambda (a + bck_p)^{\frac{\alpha+\lambda}{\lambda}}}{\alpha} \right]^{\frac{\lambda}{\alpha}}. \quad (4.44)$$

Aplicando estas equações no problema em questão, chega-se a

$$k_{pc} = \frac{1}{0,025} \left\{ \left[\frac{1,25(-0,025k_i)^{\frac{1,25}{\lambda}}}{\lambda z_c} \right]^{\frac{\lambda}{1,25+\lambda}} - 0,015 \right\} \quad (4.45)$$

$$k_{ic} = -\frac{1}{0,025} \left[\frac{z_c \lambda (0,015 + 0,025k_p)^{\frac{1,25+\lambda}{\lambda}}}{1,25} \right]^{\frac{\lambda}{1,25}}. \quad (4.46)$$

Algoritmo 4: Obtenção do tipo de r .

Dados: $r \in \mathbb{Q}_+^*$
Resultado: tipo
 Calcule $\frac{n}{d} \leftarrow r$;
se $\text{rem}(n,2)=0 \ \&\& \ \text{rem}(d,2)=1$ **então**
 | tipo = 3 ;
senão se $\text{rem}(n,2)=1 \ \&\& \ \text{rem}(d,2)=1$ **então**
 | tipo = 2 ;
senão se $\text{rem}(n,2)=1 \ \&\& \ \text{rem}(d,2)=0$ **então**
 | tipo = 1 ;
fim
retorna tipo

Observa-se que a partir da Lambert-Tsallis, obtêm-se equações analíticas dos limites dos parâmetros do controlador que faz o sistema de controle de ordem fracionária ser estável. Tais ganhos são função de um dado ganho k_p ou k_i e, sobretudo, de λ e z_c , que pode ser visto como $z_c(\lambda)$. Portanto, se faz necessário obter o valor de z_c para cada valor de λ considerado.

Assim, definindo os valores máximo e mínimo de λ que se deseja abordar, e o número de casas decimais considerado, aplica-se o [Algoritmo 3](#) para obter z_c de cada λ_i específico. Deve-se atentar para a aparição de valores de r do tipo 1, o que implica em valores complexos de z_c (vide [Exemplo 4.2.1](#)). O tipo de r pode ser verificado pela aplicação do [Algoritmo 4](#). Como dito anteriormente, nestes casos pode-se aplicar a [Propriedade 3.4](#) para utilizar a Lambert-Tsallis com $\frac{1}{r}$ do tipo 3. Outra possibilidade, que será aplicada neste exemplo, é alterar o valor de λ para o valor mais próximo de forma a ter um r do tipo desejado, conforme o número de casa decimais considerado (vide a [Tabela 5](#)), de forma semelhante ao realizado na [Figura 13](#). Caso deseje-se realizar esta alteração, um método é proposto no [Algoritmo 5](#). Neste algoritmo, o número de casas decimais é denotado por x , assim como na [Tabela 5](#).

De forma semelhante à [Figura 22](#), o processo de busca de z_c para o caso 9 da [Tabela 6](#), obtidos pelo [Algoritmo 3](#), é ilustrado na [Figura 24](#) para alguns valores de λ . Para o caso da [Figura 24.a](#), o argumento de $W_r^*(z)$ à medida que z é incrementado é ilustrado na [Figura 25](#). É possível observar o momento em que $\arg[W_r^*(z)]$ cruza o ângulo crítico θ_c . O valor de z_c é obtido através da interpolação linear entre os pontos imediatamente anterior e posterior a θ_c .

Considerando uma faixa de λ de 0,76 a 2, o gráfico $z_c \times \lambda$ para o caso 9 da [Tabela 6](#) com $\theta_c = 112,5^\circ$ é ilustrado na [Figura 26](#). Observou-se os valores de $z_c = 123.744,7$ para

Algoritmo 5: Correção do valor de λ .

Dados: λ , α , x , $tipo_desejado$
Resultado: λ corrigido

para $i = 1 : length(\lambda)$ **faça**

 Calcule $r_i \leftarrow \frac{\alpha}{\lambda_i}$;

 enquanto $tipo \neq tipo_desejado$ **faça**

 $r_i \leftarrow$ arredondamento na casa decimal $(x + 1)$ de $r_i + 10^{-x-1}$;

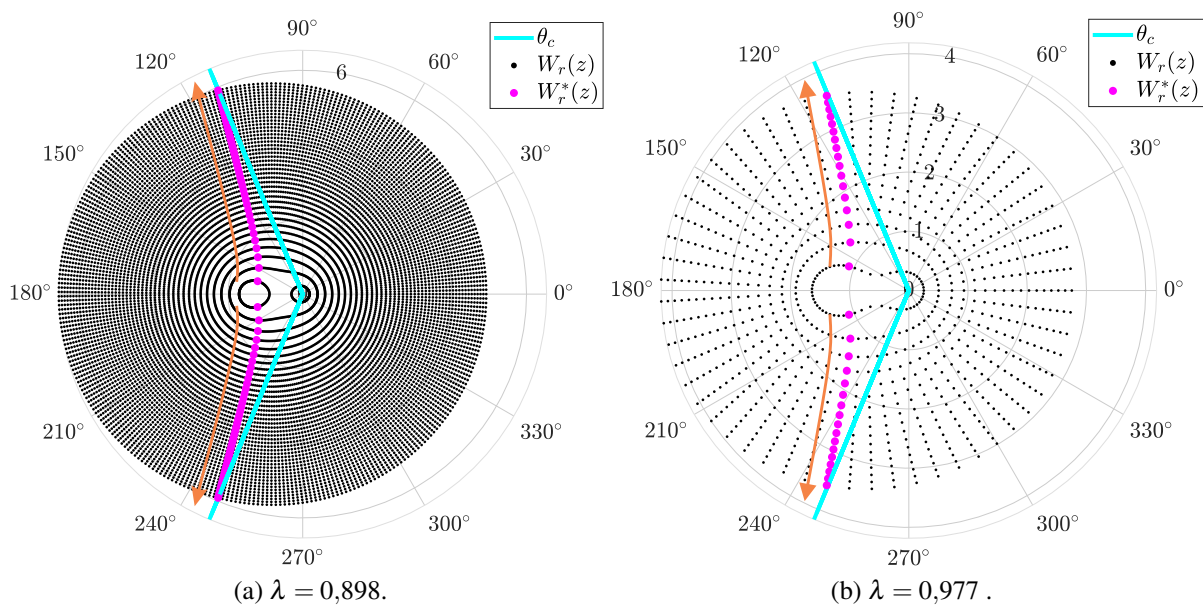
 Obtenha o $tipo$ de r_i através do [Algoritmo 4](#) ;

 fim

 $\lambda_i \leftarrow \frac{\alpha}{r_i}$;

fim
retorna λ ;

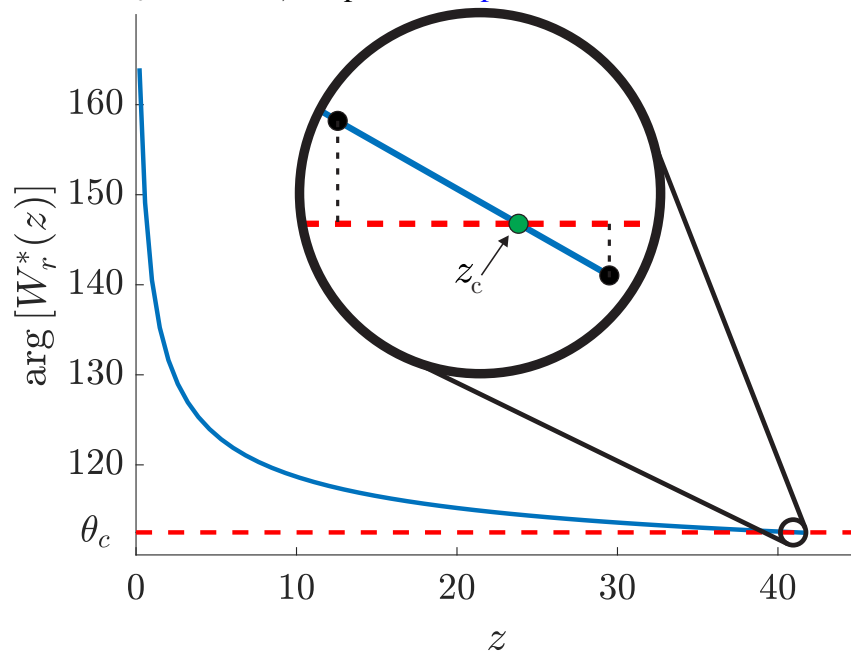
Figura 24 – Exemplos pontuais da busca de z_c para o [Exemplo 4.2.4](#).



Fonte: elaborada pelo autor.

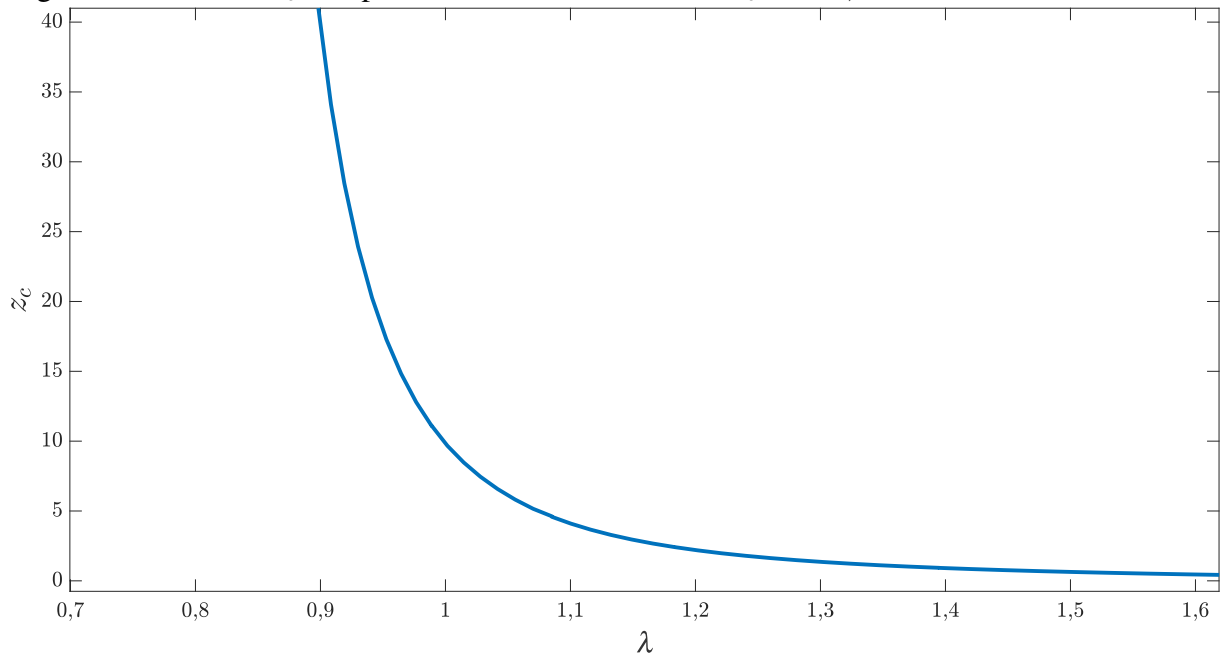
$\lambda = 0,758$ e $z_c = 0,0486$ com $\lambda = 1,953$. É importante ressaltar que, como z_c só depende de m e n , os valores obtidos podem ser aplicados a qualquer problema de controle que se encaixe no caso 9 para $\theta_c = 112,5^\circ$. O volume tridimensional de estabilidade, a partir dos parâmetros λ , k_p e k_i , é ilustrado na [Figura 27](#) para os limites dos parâmetros considerados. Percebe-se que a região de instabilidade possui um maior volume em relação à região estável. Este tipo de informação pode ser de grande utilidade para o projeto de controladores.

Figura 25 – Busca de z_c em $\lambda = 0,898$ para Exemplo 4.2.4.



Fonte: elaborada pelo autor.

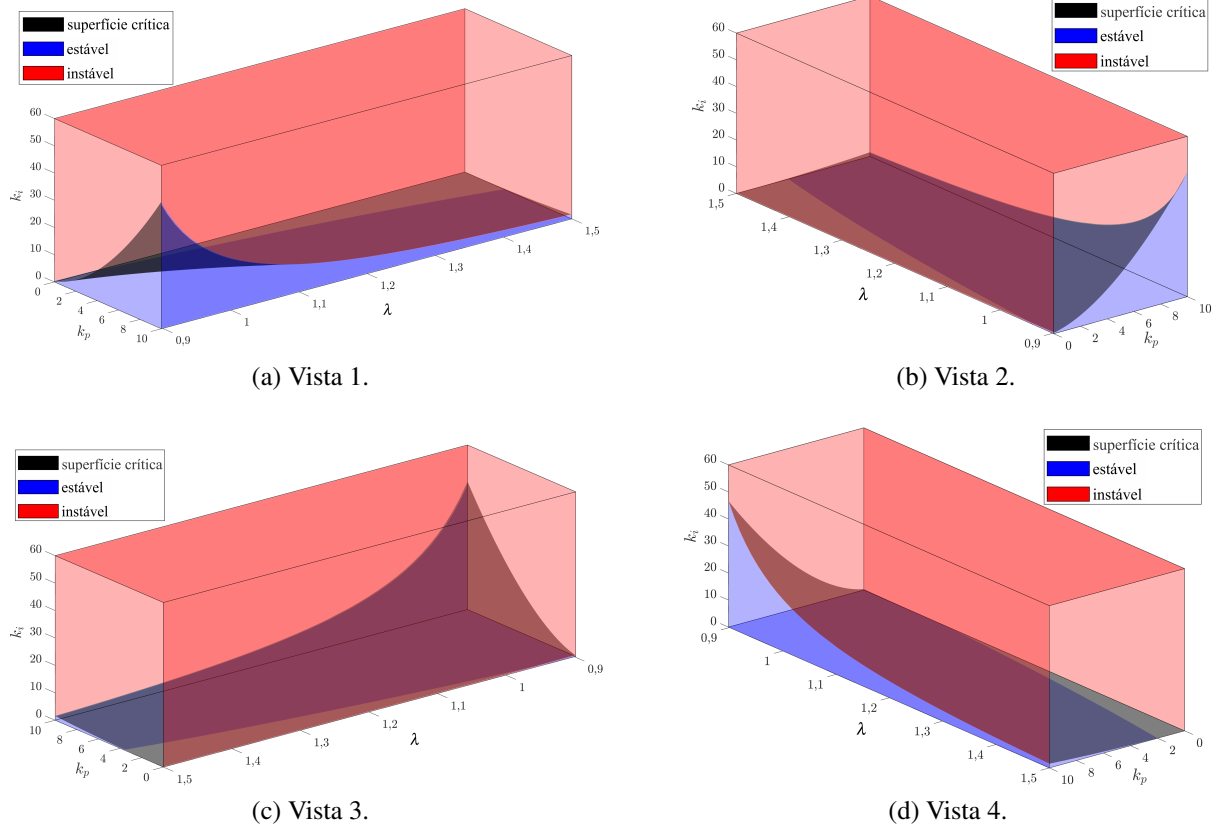
Figura 26 – Gráfico $z_c \times \lambda$ para caso 9 da Tabela 6 com $\theta_c = 112,5^\circ$.



Fonte: elaborada pelo autor.

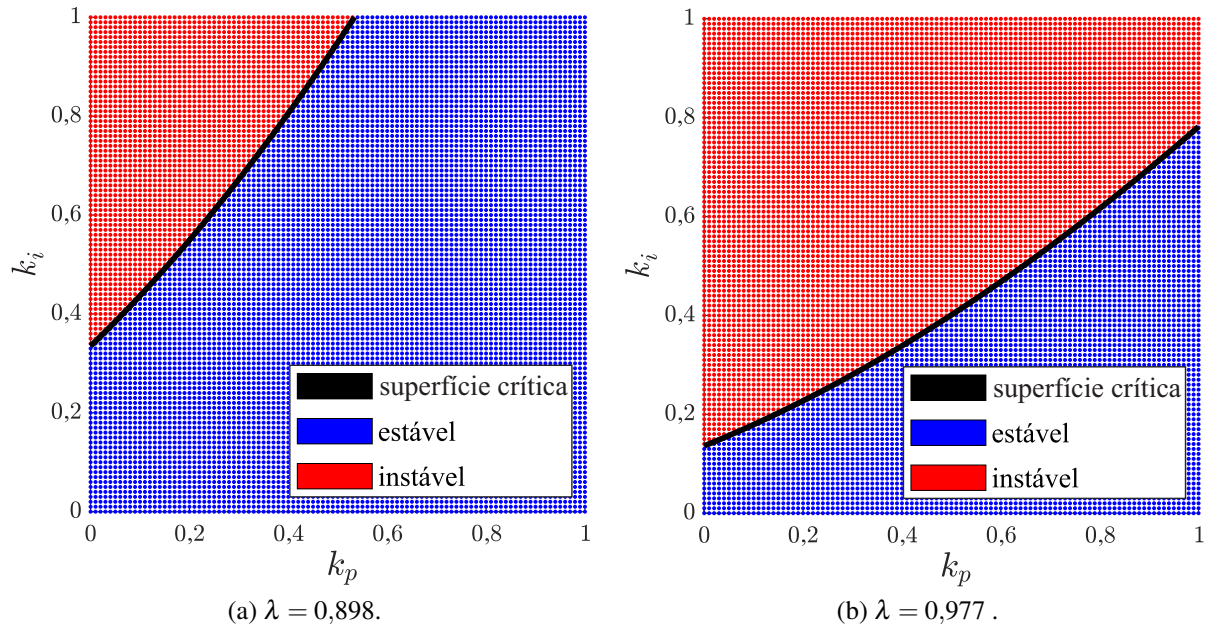
Para validar os resultados da Figura 27, é realizada uma verificação *a posteriori* utilizando o teorema de Matignon (1996), cujos resultados são ilustrados na Figura 28 para alguns valores de λ e que validam os limites obtidos com o método proposto.

Figura 27 – Volume de estabilidade do Exemplo 4.2.4.



Fonte: elaborada pelo autor.

Figura 28 – Teste de estabilidade *a posteriori* do Exemplo 4.2.4, com passo de 0,01.



Fonte: elaborada pelo autor.

4.3 Considerações sobre os métodos propostos

Este capítulo apresentou contribuições inéditas relacionadas à função Lambert-Tsallis, como a análise aprofundada do polimorfismo da função para $W_r(z) \in \mathbb{R}$ e $W_r(z) \in \mathbb{C}$, sua classificação em apenas três tipos para $r \in \mathbb{Q}_+^*$ com base no terceiro ramo real, e a identificação de um padrão periódico associado às casas decimais de r . Além disso, foi proposta a formulação do [Algoritmo 1](#), capaz de calcular todos os valores de $W_r(z)$, inclusive os complexos. Com a aplicação deste algoritmo, torna-se possível, pela primeira vez, saber exatamente quantas raízes e quantos ramos complexos $W_r(z)$ terá para qualquer valor de $r \in \mathbb{Q}_+^*$. Estes conhecimentos foram desenvolvidos a partir da necessidade de compreender o comportamento de $W_r(z)$ no contexto do problema abordado, uma vez que em sistemas de controle e sistemas dinâmicos, a variável s de Laplace pertence ao conjunto dos números complexos.

Ao contrário de qualquer outra técnica utilizada no contexto da análise numérica do problema matemático em consideração, a abordagem utilizando a Lambert-Tsallis é a única solução atualmente conhecida capaz de solucionar analiticamente trinômios generalizados. Desta forma, é possível analisar os polos do sistema diretamente no plano s , sem a necessidade de transformações polinomiais, como em abordagens usualmente utilizadas. Consequentemente, a utilização da Lambert-Tsallis juntamente com o conceito de z_c (z crítico) também é a única a fornecer discriminantes generalizados de estabilidade, compostas pelas relações entre os ganhos do controlador e todos os parâmetros da planta.

Uma possível aplicação para estes discriminantes, *e.g.* a [Equação 4.43](#) e a [Equação 4.44](#), é seu uso em conjunto com algoritmos de otimização, tais como gradiente descendente e meta-heurísticas. Com base na [Figura 27](#), percebe-se que, ao impor os limites de busca a partir dos valores críticos obtidos com a utilização da $W_r(z)$ e z_c , o algoritmo de otimização, além de evitar buscas locais que levam à instabilidade, terá como resultado um sistema de controle de ordem fracionária estável *a priori*, pois o espaço de busca será sempre estável. Esta possibilidade de aplicação da Lambert-Tsallis junto a algoritmos de otimização será proposta e explorada no próximo capítulo, que trata do projeto de um sistema de controle de ordem fracionária de uma planta real, a saber, uma incubadora neonatal.

5 SIMULAÇÃO DE SISTEMAS DINÂMICOS DE ORDEM FRACIONÁRIA

Das três definições de derivada fracionária apresentadas no [Capítulo 2](#), a mais conveniente para simulação é a de Grünwald-Letnikov, uma vez que sua expressão originária envolve operações discretas como o somatório.

Embora as outras definições citadas anteriormente possuam versões modificadas visando a discretização, como as de Riemann–Liouville e Caputo em [Xue \(2017\)](#) e [Aburakhis et al. \(2022\)](#), e embora as definições de Grünwald-Letnikov e Riemann–Liouville sejam equivalentes sob determinadas circunstâncias ([PODLUBNY, 1998](#)), o resultado da simulação computacional da derivada de Riemann–Liouville pode não ser tão bom quanto a de Grünwald-Letnikov, chegando a ser bem diferentes entre si ([XUE, 2017](#)). Este é um importante fator no momento de escolha da definição a se utilizar. Para exemplificar a simulação de um sistema de ordem fracionária, a seguir será desenvolvida a equação em tempo discreto de um sistema fracionário simples, com base na definição de Grünwald-Letnikov.

5.1 Equação discreta de um sistema de ordem fracionária

Seja o sistema $G(s) = \frac{c}{s^\alpha + a}$, em que c e a são constantes reais e α é o expoente fracionário. Aplicando a discretização de Euler *a priori*, $s = \frac{1 - z^{-1}}{T}$, em que z é a variável complexa da transformada Z, e em seguida a definição compacta de Grünwald-Letnikov de derivada fracionária desenvolvida na [Subseção 2.1.1](#) (vide [Equação 2.18](#)), obtém-se

$$G(z) = \frac{Y(z)}{U(z)} = \frac{c}{\left(\frac{1 - z^{-1}}{T}\right)^\alpha + a} = \frac{cT^\alpha}{(1 - z^{-1})^\alpha + aT^\alpha} = \frac{cT^\alpha}{\left(\sum_{i=0}^{\infty} {}^\alpha w_i z^{-i}\right) + aT^\alpha}, \quad (5.1)$$

em que ${}^\alpha w_i$ são os coeficientes relacionados à derivada fracionária de ordem α (vide [Equação 2.7](#)). Multiplicando os termos da expressão anterior de forma cruzada, chega-se a

$$Y(z) \left(\sum_{i=0}^{\infty} {}^\alpha w_i z^{-i} \right) + aT^\alpha Y(z) = cT^\alpha U(z). \quad (5.2)$$

Extraindo o primeiro termo do somatório e aplicando a transformada Z inversa, tem-se

$${}^\alpha w_0 y[k] + \left(\sum_{i=1}^{\infty} {}^\alpha w_i y[k - i] \right) + aT^\alpha y[k] = cT^\alpha u[k]. \quad (5.3)$$

Por fim, isolando $y[k]$, obtêm-se

$$({}^{\alpha}w_0 + aT^{\alpha})y[k] = cT^{\alpha}u[k] - \left(\sum_{i=1}^{\infty} {}^{\alpha}w_i y[k-i] \right) \quad (5.4)$$

$$y[k] = \frac{1}{({}^{\alpha}w_0 + aT^{\alpha})} \left\{ cT^{\alpha}u[k] - \left(\sum_{i=1}^{\infty} {}^{\alpha}w_i y[k-i] \right) \right\}, \quad (5.5)$$

em que $w_i = \left(1 - \frac{\alpha+1}{i}\right) w_{i-1}$ e $w_0 := 1$, como demonstrado na [Equação 2.24](#). Intuitivamente percebe-se que, uma vez que esta equação funciona para $\alpha \in \mathbb{R}$, também deverá funcionar para $\alpha \in \mathbb{Z}$. Pode-se demonstrar que a [Equação 5.5](#) é uma generalização da equação discreta de Euler *a priori* para sistemas de ordem inteira que possuam essa mesma estrutura. Para isso, pode-se comparar a [Equação 5.5](#), que generaliza o valor de α , com a equação temporal discreta para sistema de primeira ordem:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{c}{s+a} \quad (5.6)$$

$$G(z) = \frac{Y(z)}{U(z)} = \frac{c}{\left(\frac{1-z^{-1}}{T}\right) + a} = \frac{cT}{1-z^{-1} + aT} \quad (5.7)$$

$$Y(z)(1+aT) - z^{-1}Y(z) = cTU(z) \quad (5.8)$$

$$(1+aT)y[k] - y[k-1] = cTu[k] \quad (5.9)$$

$$y[k] = \frac{1}{(1+aT)} \left\{ cTu[k] + y[k-1] \right\}. \quad (5.10)$$

Calculando os coeficientes ${}^{\alpha}w_i$ da [Equação 5.5](#) para $\alpha = 1$, em que $w_i = \left(1 - \frac{\alpha+1}{i}\right) w_{i-1}$ e $w_0 := 1$ (vide [Equação 2.24](#)), tem-se

$${}^{\alpha}w_1 = \left(1 - \frac{1+1}{1}\right) 1 = -1$$

$${}^{\alpha}w_2 = \left(1 - \frac{1+1}{2}\right) (-1) = 0$$

$${}^{\alpha}w_3 = \left(1 - \frac{1+1}{3}\right) 0 = 0$$

\vdots

$${}^{\alpha}w_{\infty} = 0.$$

Observa-se que apenas o termo α_{w_1} é não nulo. Isso significa que, no somatório presente na [Equação 5.5](#), apenas o termo $\alpha_{w_1}y[k-1]$ é diferente de zero, pois consideram-se todos os valores iniciais nulos ($y[k < i] = 0$). Assim, aplicando estes valores de α_{w_i} na [Equação 5.5](#), chega-se a

$$y[k] = \frac{1}{(\alpha_{w_0} + aT^\alpha)} \left\{ cT^\alpha u[k] - \left(\sum_{i=1}^{\infty} \alpha_{w_i} y[k-i] \right) \right\} \quad (5.11)$$

$$y[k] = \frac{1}{(1 + aT^1)} \left\{ cT^1 u[k] - \alpha_{w_1} y[k-1] \right\} \quad (5.12)$$

$$y[k] = \frac{1}{(1 + aT)} \left\{ cT u[k] + y[k-1] \right\}, \quad (5.13)$$

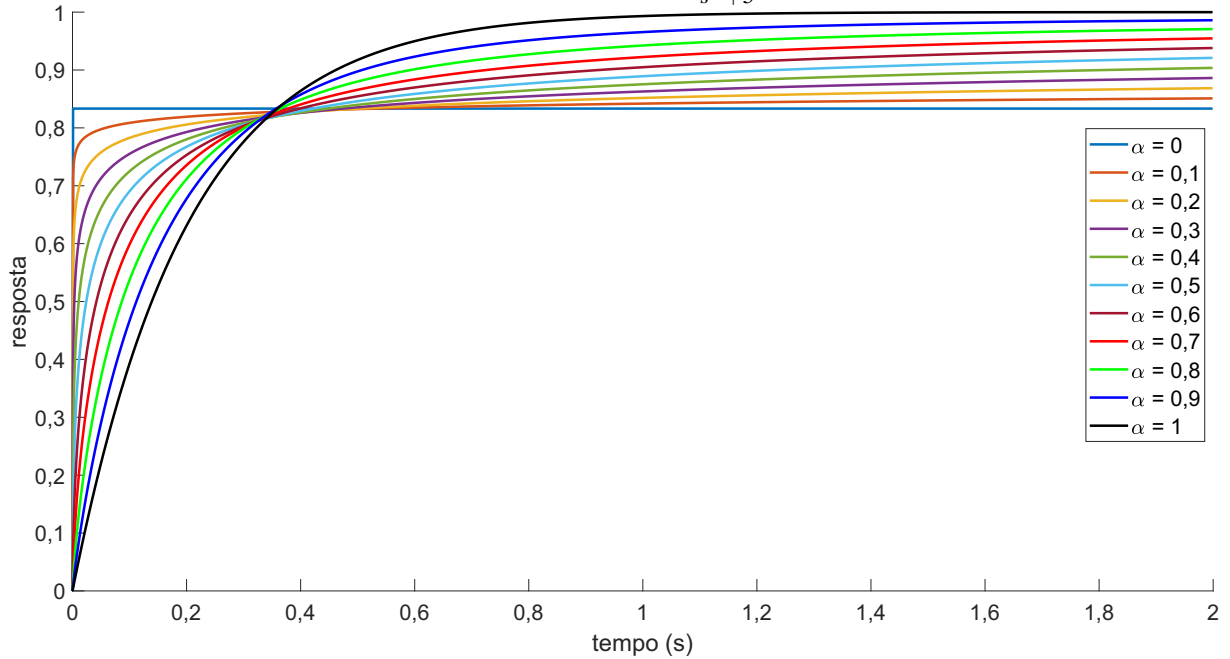
resultando exatamente na mesma expressão da [Equação 5.10](#), que representa a evolução de um sistema de primeira ordem (inteira). A mesma comparação pode ser feita para qualquer sistema de ordem inteira, como sistemas de segunda ordem, em que haverá apenas dois termos diferentes de zero no somatório: $\alpha_{w_1}y[k-1]$ e $\alpha_{w_2}y[k-2]$. Aos sistemas em que a ordem não é um valor inteiro, haverá infinitos valores no somatório. Conclui-se então que as expressões desenvolvidas a partir da definição de Grünwald-Letnikov de derivada fracionária são expressões generalistas em relação à ordem do sistema.

A evolução temporal discreta presente na [Equação 5.5](#) utiliza o valor da derivada fracionária mais próxima possível da definição de Grünwald-Letnikov, pois são utilizados todos os valores passados da função. Porém, a definição de Grünwald-Letnikov possui infinitos termos, o que impede sua aplicação computacional. A estratégia utilizada nesta tese para que a implementação dessa equação seja possível é considerar os valores iniciais nulos ($y[k < i] = 0$), evitando assim infinitos termos no somatório. Visando a simulação computacional, considerando nulos os valores em que $i > k$, a [Equação 5.5](#) pode ser rescrita como

$$y[k]_{k=2:t_{final}} = \frac{1}{(\alpha_{w_0} + aT^\alpha)} \left\{ cT^\alpha u[k] - \left(\sum_{i=1}^{k-1} \alpha_{w_i} y[k-i] \right) \right\}. \quad (5.14)$$

Na [Figura 29](#) são ilustradas as respostas ao degrau unitário do sistema $G(s) = \frac{5}{s^\alpha + 5}$ para $T = 1$ e diferentes valores de α , calculadas a partir da [Equação 5.14](#). É interessante notar o quão versátil pode ser a resposta ao degrau apenas por mudar a ordem do sistema. Tal versatilidade pode ser bastante interessante especialmente para identificação e controle de sistemas, em que pode-se alcançar resultados melhores que os obtidos por modelos de ordem inteira.

Figura 29 – Respostas ao degrau unitário de $G(s) = \frac{5}{s^{\alpha}+5}$ para diferentes valores de α .



Fonte: elaborada pelo autor.

Como mencionado anteriormente, nos sistemas em que a ordem não possui um valor inteiro, haverá infinitos valores no somatório presente na definição de Grünwald-Letnikov de derivada fracionária. A princípio, isso pode ser um problema, visto que sistemas simulados possuem limitação de memória. Pode-se pensar em limitar a quantidade de elementos no somatório, imaginando que seu efeito será similar ao de truncar um número real em determinada casa decimal. Este método é conhecido como efeito de memória curta (*short-memory effect*) (PODLUBNY, 1994b; PODLUBNY, 1998), definido como

$${}_{t_0}\mathcal{D}_t^\alpha f(t) \approx \frac{1}{T^\alpha} \sum_{i=0}^{L_0} w_i f[k-i], \quad (5.15)$$

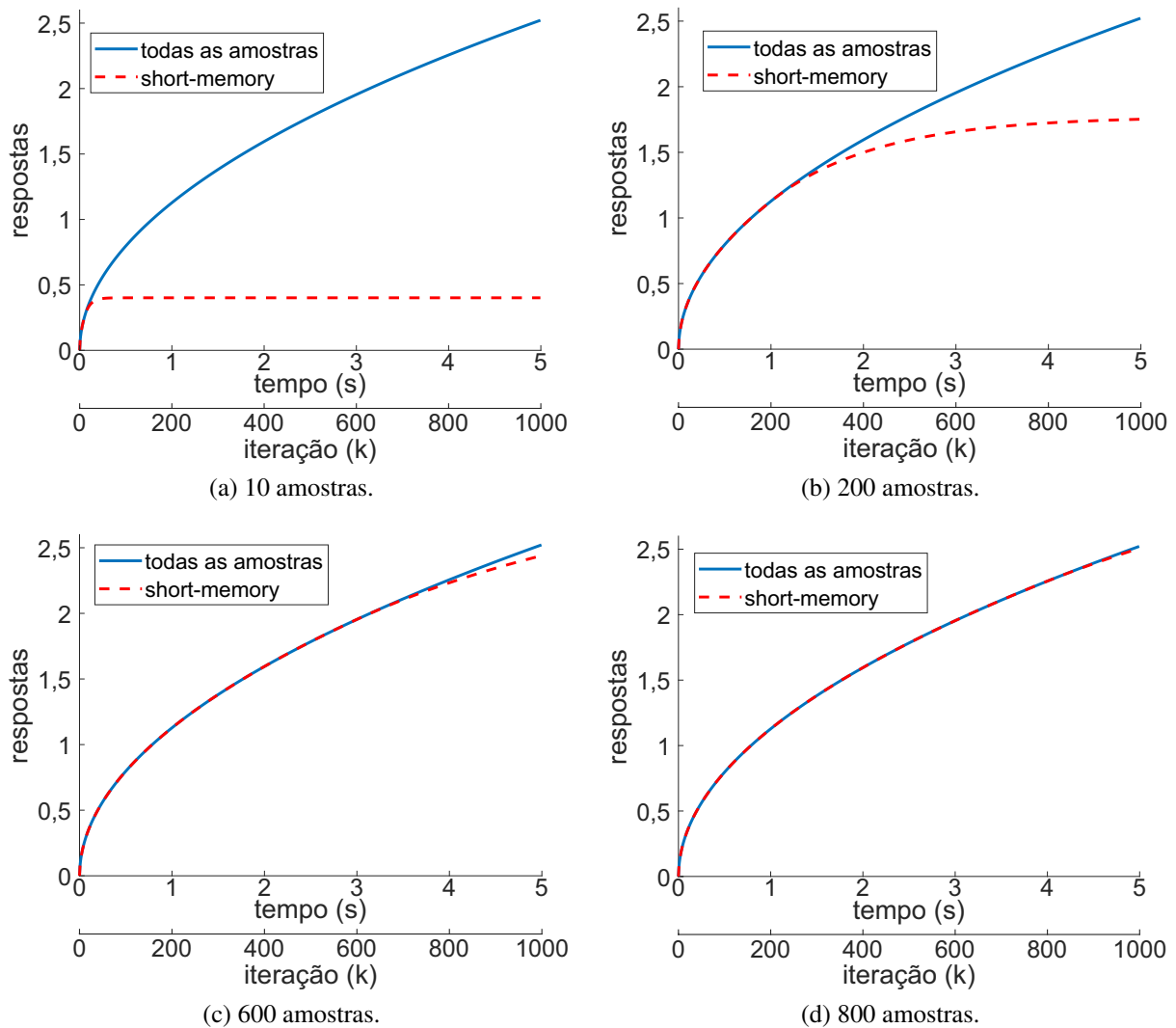
em que L_0 é a profundidade da janela de memória. Em outras palavras, a derivada de ordem fracionária no instante k é aproximada por uma janela móvel no intervalo $[k-L_0, k]$.

Porém, essa ação pode ter efeitos deletérios, pois com esta simplificação, aumenta-se a imprecisão da estimativa da derivada. Como exemplo, na Figura 30 são ilustradas as respostas ao degrau unitário de uma integral fracionária de ordem 0,5, ou seja $G(s) = \frac{1}{s^{0,5}}$, para diferentes valores de L_0 . São calculados os valores de RMSE (*root mean square error*), definido como

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}, \quad (5.16)$$

em que y_i representa os valores obtidos pela definição de Grünwald-Letnikov com memória infinita e \hat{y}_i os valores da versão com memória curta. Os resultados são apresentados na Figura 8.

Figura 30 – Resposta ao degrau da integral fracionária de ordem 0,5 para diferentes janelas de memória.



Fonte: elaborada pelo autor.

Tabela 8 – Valores de RMSE para diferentes tamanhos da janela L_0 .

tamanho da janela L_0 (memória curta)	RMSE
10	1,410708
200	0,359634
600	0,024221
800	0,003162

Fonte: elaborada pelo autor.

Percebe-se que há diferenças visíveis entre as duas respostas, mesmo que os valores de $\alpha_{w_{i>200}}$ para $\alpha = 0,5$ sejam da ordem de 10^{-5} . Portanto, utilizar a aproximação de memória-curta pode não ser uma boa ideia, principalmente para aplicações embarcadas, como controle de sistemas. Por outro lado, Podlubny (1994b) e Podlubny (1998) demonstram que, para se obter o erro de aproximação menor que uma tolerância ε , dado que $|f[k]|$ é menor que um certo valor M dentro

do intervalo de tempo da simulação, L_0 deve ser tal que

$$L_0 > \left(\frac{M}{\varepsilon |\Gamma(1 - \alpha)|} \right)^{\frac{1}{\alpha}}. \quad (5.17)$$

Contudo, [Xue \(2017\)](#) evidencia que, mesmo para uma precisão baixa de $\varepsilon = 10^{-2}$, com $\alpha = 0,1$ e $M = 1$, o valor de L_0 deve ser da ordem de 10^{19} , o que é inviável. Uma solução para simulação computacional é proposta na próxima seção, enquanto que uma solução para aplicações embarcadas é proposta na [Seção 7.2](#).

5.2 Sobre *softwares* para simulação de sistemas de ordem fracionária

Diversos autores elaboraram funções e *toolboxes* principalmente para o *software* MATLAB na tentativa de obter boas implementações de sistemas de ordem fracionária. Entre essas ferramentas, destacam-se, cronologicamente, a CRONE ([OUSTALOUP, 1991](#); [OUSTALOUP et al., 2000](#)), a Ninteger ([VALÉRIO, 2001](#); [VALÉRIO; COSTA, 2004](#)), a FOTF ([CHEN et al., 2009](#); [XUE et al., 2017](#); [PETRÁŠ, 2019](#)) e a FOMCON ([TEPLJAKOV, 2011](#); [TEPLJAKOV et al., 2019](#)). É pertinente mencionar que os autores da FOMCON implementaram também a FOMCONpy ([ONYEDI et al., 2020](#)), uma adaptação da *toolbox* original para linguagem *Python*. Outras referências para o cálculo de derivadas e integrais de ordem fracionária em *Python* podem ser encontradas em [Midya et al. \(2018\)](#) e [Miles \(2019\)](#).

A *toolbox* CRONE emprega o filtro de Oustaloup, que utiliza funções de transferência convencionais para representar uma aproximação com banda limitada de um operador de ordem fracionária. Se os parâmetros do filtro forem escolhidos de maneira apropriada, o filtro aproximará a derivada de Riemann–Liouville ([PETRÁŠ, 2019](#)). Porém, a necessidade de parametrizar o filtro pode desestimular o usuário a utilizar esta técnica. Alguns trabalhos abordam o problema de como selecionar estes parâmetros, como [Merrikh-Bayat \(2012\)](#) que apresenta regras de seleção, e [Gao e Liao \(2012\)](#), que utilizam o algoritmo de otimização PSO para parametrização ótima do filtro.

A *toolbox* Ninteger possui dois propósitos principais: a identificação de sistemas de ordem fracionária e a aproximação de sistemas de ordem fracionária com modelos de ordem inteira ([XUE; BAI, 2024](#)). A respeito da aproximação de sistemas de ordem fracionária, a *toolbox* essencialmente implementa as aproximações de Oustaloup, de Carlson ([CARLSON; HALIJAK, 1960](#); [CARLSON; HALIJAK, 1964](#)) e de Matsuda ([MATSUDA; FUJII, 1993](#)). A *toolbox* também conta com um conjunto de funções, modelos e interfaces para o projeto e análise

destes sistemas no MATLAB e *Simulink*.

A FOMCON encapsula alguma das principais funcionalidades das três *toolboxes* mencionadas anteriormente (CRONE, Ninteger e FOTF) e constrói uma interface gráfica GUI com o objetivo de aprimorar os esquemas clássicos de controle para uso no projeto de controladores de ordem fracionária e para realizar a identificação, análise e projeto de controle destes sistemas (LI *et al.*, 2017; XUE; BAI, 2024).

Porém, de acordo com Midya *et al.* (2018), os métodos de aproximação como a do princípio da memória-curta (*short-memory*), filtro de Oustaloup, Carlson, Matsuda, entre outros presentes nas *toolboxes* CRONE, Ninteger e FOMCON são utilizados para reduzir o custo computacional e o tempo de processamento de tais operações. No entanto, tais aproximações frequentemente fornecem resultados imprecisos. Além disso, muitos pesquisadores preferem trabalhar com as funções de transferência de ordem fracionária nas suas formas originais, sem aproximações por modelos de ordem inteira (MOROZ; BOROVETS, 2017).

Uma *toolbox* que trabalha diretamente com estas funções de transferência em suas formas originais é a FOTF, que usa a técnica de programação por sobrecarga de funções para permitir que as funções nativas do MATLAB e *Simulink* tratem modelos de ordem fracionária (LI *et al.*, 2017), além de contar com várias outras funções para cálculo fracionário e controle. Desde 2017, conta com suporte a sistemas multivariáveis de ordem fracionária (XUE; BAI, 2024).

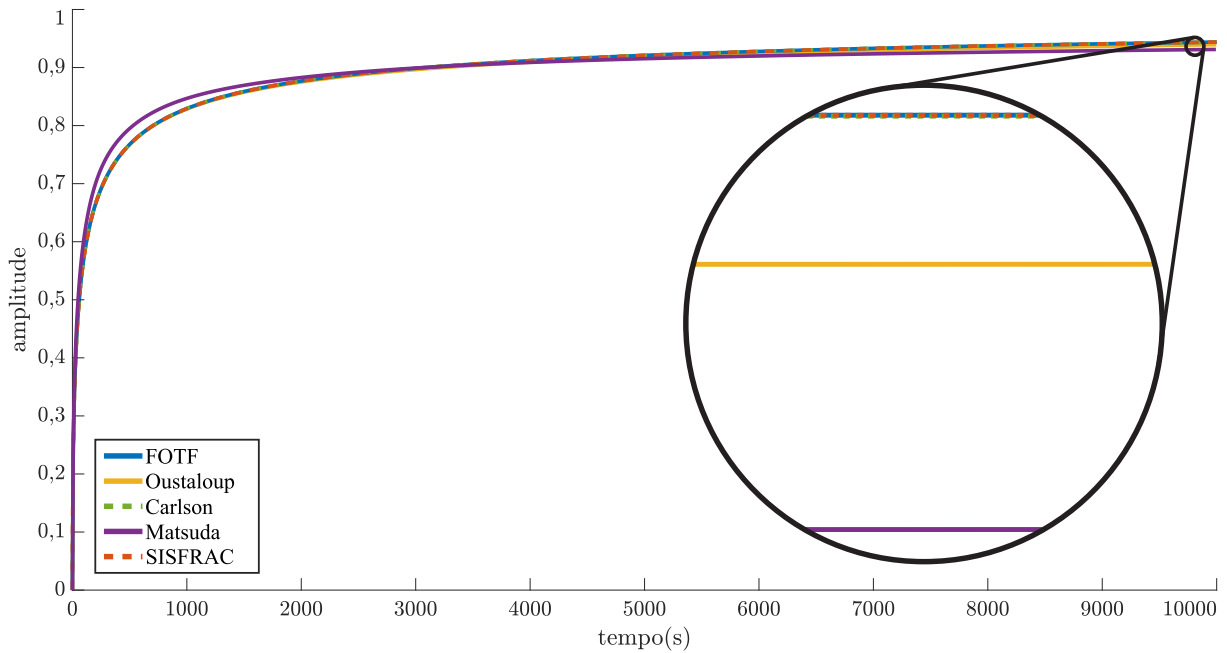
Apesar da variedade de funções e *toolboxes* de ordem fracionária existente atualmente, Petráš (2019) e Xue e Bai (2024) alertam que apesar dos entusiastas dos sistemas de ordem fracionária serem encorajados a procurar por *toolboxes* no *File Exchange*¹, os mesmos devem ter cuidado para selecionar ferramentas confiáveis, uma vez que os métodos de implementação podem ser significativamente diferentes, e alguns possuem baixa qualidade, que pode levar a resultados incorretos.

A *toolbox* FOTF é bastante interessante, uma vez que utiliza uma solução em forma fechada (vide Zhao e Xue (2008)) para computar a saída de um sistema de ordem fracionária. Esta solução em forma fechada é a mesma apresentada na Equação 5.5 desenvolvida nesta seção. Ambas utilizam a definição de Grünwald-Letnikov de derivada fracionária sem *short-memory*, que é a forma mais acurada de computação da derivada de ordem fracionária.

A fim de avaliar os diferentes métodos mencionados (FOTF, filtros de Oustaloup,

¹O *File Exchange* é um repositório online dentro da plataforma MATLAB Central, no qual usuários podem compartilhar, baixar e avaliar arquivos, códigos, funções e outras ferramentas da MathWorks desenvolvidas pela comunidade de usuários do MATLAB.

Figura 31 – Respostas ao degrau unitário dos diferentes métodos de simulação de sistemas de ordem fracionária.



Fonte: elaborada pelo autor.

Carlson e Matsuda), uma simulação de resposta ao degrau do sistema $G = \frac{0,1}{s^{0,5} + 0,1}$ com $T = 1$ é realizada com o intuito de comparar os tempos de execução e o índice RMSE (apresentado na [Equação 5.16](#)). São realizadas 30 execuções para obtenção do tempo médio e seu desvio-padrão. As especificações da máquina utilizada para as simulações nesta tese são apresentadas no [Apêndice A](#). Uma vez que a *toolbox* FOTF é a única que possui uma solução em forma fechada, e que utiliza todos os dados anteriores para o cálculo da derivada fracionária, esta será utilizada como resposta de referência com a qual as outras respostas serão comparadas para cálculo do RMSE. A solução fechada desenvolvida na [Equação 5.14](#) também será comparada, doravante chamada de SISFRAC (simulação computacional de sistemas de ordem fracionária).

As respostas ao degrau unitário dos métodos citados são ilustradas na [Figura 31](#) para tempo de simulação de 10.000 segundos, enquanto que os modelos utilizados nesta comparação são apresentados na [Tabela 9](#). Por fim, as métricas mencionadas são apresentadas na [Tabela 10](#).

Observa-se que as repostas ao degrau são bastante próximas. Entre os métodos de aproximação por filtro, a melhor resposta é de Carlson, com menor RMSE, enquanto que a pior é a de Matsuda, com o pior valor de RMSE e uma visível diferença em sua resposta transitória. Os tempos médios de execução das aproximações por filtro são os mesmos, cerca de 1,5ms. Os modelos aproximados de Oustaloup, Carlson e Matsuda possuem ordem 31 e apresentam parâmetros de altíssimas ordens. Manipular modelos de ordem tão grande com parâmetros

Tabela 9 – Modelos utilizados para comparação presente na Figura 31 e Tabela 10.

método	modelo
FOTF	$\frac{0,1}{s^{0,5} + 0,1}$
Oustaloup	$\frac{0,0003s^{31} + 50s^{30} + 2,6 \cdot 10^6 s^{29} + 5,3 \cdot 10^{10} s^{28} + 5 \cdot 10^{14} s^{27} + 2,2 \cdot 10^{18} s^{26} + 4,5 \cdot 10^{21} s^{25} + 4,4 \cdot 10^{24} s^{24} + 2 \cdot 10^{27} s^{23} + 4,4 \cdot 10^{29} s^{22} + 4,6 \cdot 10^{31} s^{21} + 2,3 \cdot 10^{33} s^{20} + 5,3 \cdot 10^{34} s^{19} + 6 \cdot 10^{35} s^{18} + 3 \cdot 10^{36} s^{17} + 8 \cdot 10^{36} s^{16} + 10^{37} s^{15} + 5,5 \cdot 10^{36} s^{14} + 1,5 \cdot 10^{36} s^{13} + 2 \cdot 10^{35} s^{12} + 1,2 \cdot 10^{34} s^{11} + 3,5 \cdot 10^{32} s^{10} + 5 \cdot 10^{30} s^9 + 3,3 \cdot 10^{28} s^8 + 10^{26} s^7 + 1,5 \cdot 10^{23} s^6 + 10^{20} s^5 + 3,6 \cdot 10^{16} s^4 + 5,6 \cdot 10^{12} s^3 + 3,8 \cdot 10^8 s^2 + 10^4 s + 0,1}{s^{31} + 10^5 s^{30} + 3,8 \cdot 10^9 s^{29} + 5,6 \cdot 10^{13} s^{28} + 3,6 \cdot 10^{17} s^{27} + 10^{21} s^{26} + 1,5 \cdot 10^{24} s^{25} + 10^{27} s^{24} + 3,3 \cdot 10^{29} s^{23} + 5 \cdot 10^{31} s^{22} + 3,6 \cdot 10^{33} s^{21} + 1,2 \cdot 10^{35} s^{20} + 2 \cdot 10^{36} s^{19} + 1,5 \cdot 10^{37} s^{18} + 5,8 \cdot 10^{37} s^{17} + 10^{38} s^{16} + 9 \cdot 10^{37} s^{15} + 3,7 \cdot 10^{37} s^{14} + 7,4 \cdot 10^{36} s^{13} + 7,3 \cdot 10^{35} s^{12} + 1,2 \cdot 10^{34} s^{11} + 3,5 \cdot 10^{32} s^{10} + 5 \cdot 10^{30} s^9 + 3,3 \cdot 10^{28} s^8 + 10^{26} s^7 + 1,5 \cdot 10^{23} s^6 + 10^{20} s^5 + 3,6 \cdot 10^{16} s^4 + 5,6 \cdot 10^{12} s^3 + 3,8 \cdot 10^8 s^2 + 10^4 s + 0,1}$
Carlson	$\frac{0,04s^{31} + 0,16s^{30} + 0,2s^{29} + 0,13s^{28} + 0,05s^{27} + 0,01s^{26} + 0,002s^{25} + 0,0002s^{24} + 2,8 \cdot 10^{-5} s^{23} + 2,2 \cdot 10^{-6} s^{22} + 1,4 \cdot 10^{-7} s^{21} + 7,2 \cdot 10^{-9} s^{20} + 3 \cdot 10^{-10} s^{19} + 10^{-11} s^{18} + 3 \cdot 10^{-13} s^{17} + 6,8 \cdot 10^{-15} s^{16} + 1,3 \cdot 10^{-16} s^{15} + 2 \cdot 10^{-18} s^{14} + 2,8 \cdot 10^{-20} s^{13} + 3 \cdot 10^{-22} s^{12} + 3 \cdot 10^{-24} s^{11} + 2 \cdot 10^{-26} s^{10} + 10^{-28} s^9 + 4 \cdot 10^{-31} s^8 + 10^{-33} s^7 + 3 \cdot 10^{-36} s^6 + 6 \cdot 10^{-39} s^5 + 7 \cdot 10^{-42} s^4 + 5 \cdot 10^{-45} s^3 + 2 \cdot 10^{-48} s^2 + 6 \cdot 10^{-52} s + 5 \cdot 10^{-56}}{s^{31} + 2,19s^{30} + 1,95s^{29} + 0,97s^{28} + 0,304s^{27} + 0,06s^{26} + 0,01s^{25} + 0,00121s^{24} + 1,1 \cdot 10^{-4} s^{23} + 8 \cdot 10^{-6} s^{22} + 4,6 \cdot 10^{-7} s^{21} + 2,2 \cdot 10^{-8} s^{20} + 8,6 \cdot 10^{-10} s^{19} + 2,8 \cdot 10^{-11} s^{18} + 7,5 \cdot 10^{-13} s^{17} + 1,6 \cdot 10^{-14} s^{16} + 3 \cdot 10^{-16} s^{15} + 4,6 \cdot 10^{-18} s^{14} + 5,7 \cdot 10^{-20} s^{13} + 6 \cdot 10^{-22} s^{12} + 5 \cdot 10^{-24} s^{11} + 3 \cdot 10^{-26} s^{10} + 2 \cdot 10^{-28} s^9 + 7 \cdot 10^{-31} s^8 + 2 \cdot 10^{-33} s^7 + 5 \cdot 10^{-36} s^6 + 8 \cdot 10^{-39} s^5 + 10^{-41} s^4 + 7 \cdot 10^{-45} s^3 + 3 \cdot 10^{-48} s^2 + 7 \cdot 10^{-52} s + 5 \cdot 10^{-56}}$
Matsuda	$\frac{3 \cdot 10^{-7} s^{31} + 1,3 \cdot 10^4 s^{30} + 4 \cdot 10^{13} s^{29} + 2 \cdot 10^{22} s^{28} + 2 \cdot 10^{30} s^{27} + 3 \cdot 10^{37} s^{26} + 1,3 \cdot 10^{44} s^{25} + 10^{50} s^{24} + 2,6 \cdot 10^{55} s^{23} + 10^{60} s^{22} + 1,3 \cdot 10^{64} s^{21} + 3,3 \cdot 10^{67} s^{20} + 1,8 \cdot 10^{70} s^{19} + 2,3 \cdot 10^{72} s^{18} + 6,5 \cdot 10^{73} s^{17} + 4,2 \cdot 10^{74} s^{16} + 6 \cdot 10^{74} s^{15} + 2 \cdot 10^{74} s^{14} + 1,5 \cdot 10^{73} s^{13} + 2,7 \cdot 10^{71} s^{12} + 10^{69} s^{11} + 9,4 \cdot 10^{65} s^{10} + 2 \cdot 10^{62} s^9 + 8,2 \cdot 10^{57} s^8 + 8 \cdot 10^{52} s^7 + 2 \cdot 10^{47} s^6 + 9,3 \cdot 10^{40} s^5 + 10^{34} s^4 + 2,5 \cdot 10^{26} s^3 + 1,2 \cdot 10^{18} s^2 + 10^9 s + 0,1}{s^{31} + 9,3 \cdot 10^9 s^{30} + 10^{19} s^{29} + 2 \cdot 10^{27} s^{28} + 9 \cdot 10^{34} s^{27} + 8 \cdot 10^{41} s^{26} + 1,6 \cdot 10^{48} s^{25} + 7 \cdot 10^{53} s^{24} + 7 \cdot 10^{58} s^{23} + 1,6 \cdot 10^{63} s^{22} + 8,3 \cdot 10^{66} s^{21} + 9,7 \cdot 10^{69} s^{20} + 2,5 \cdot 10^{72} s^{19} + 1,5 \cdot 10^{74} s^{18} + 2,1 \cdot 10^{75} s^{17} + 6,6 \cdot 10^{75} s^{16} + 4,8 \cdot 10^{75} s^{15} + 8,5 \cdot 10^{74} s^{14} + 3,8 \cdot 10^{73} s^{13} + 4,5 \cdot 10^{71} s^{12} + 1,4 \cdot 10^{69} s^{11} + 10^{66} s^{10} + 2 \cdot 10^{62} s^9 + 8,7 \cdot 10^{57} s^8 + 8,6 \cdot 10^{52} s^7 + 2 \cdot 10^{47} s^6 + 9,6 \cdot 10^{40} s^5 + 10^{34} s^4 + 2,5 \cdot 10^{26} s^3 + 1,2 \cdot 10^{18} s^2 + 10^9 s + 0,1}$
SISFRAC	$\frac{0,1}{s^{0,5} + 0,1}$

Fonte: elaborada pelo autor.

Tabela 10 – Métricas dos diferentes métodos de simulação de sistemas de ordem fracionária.

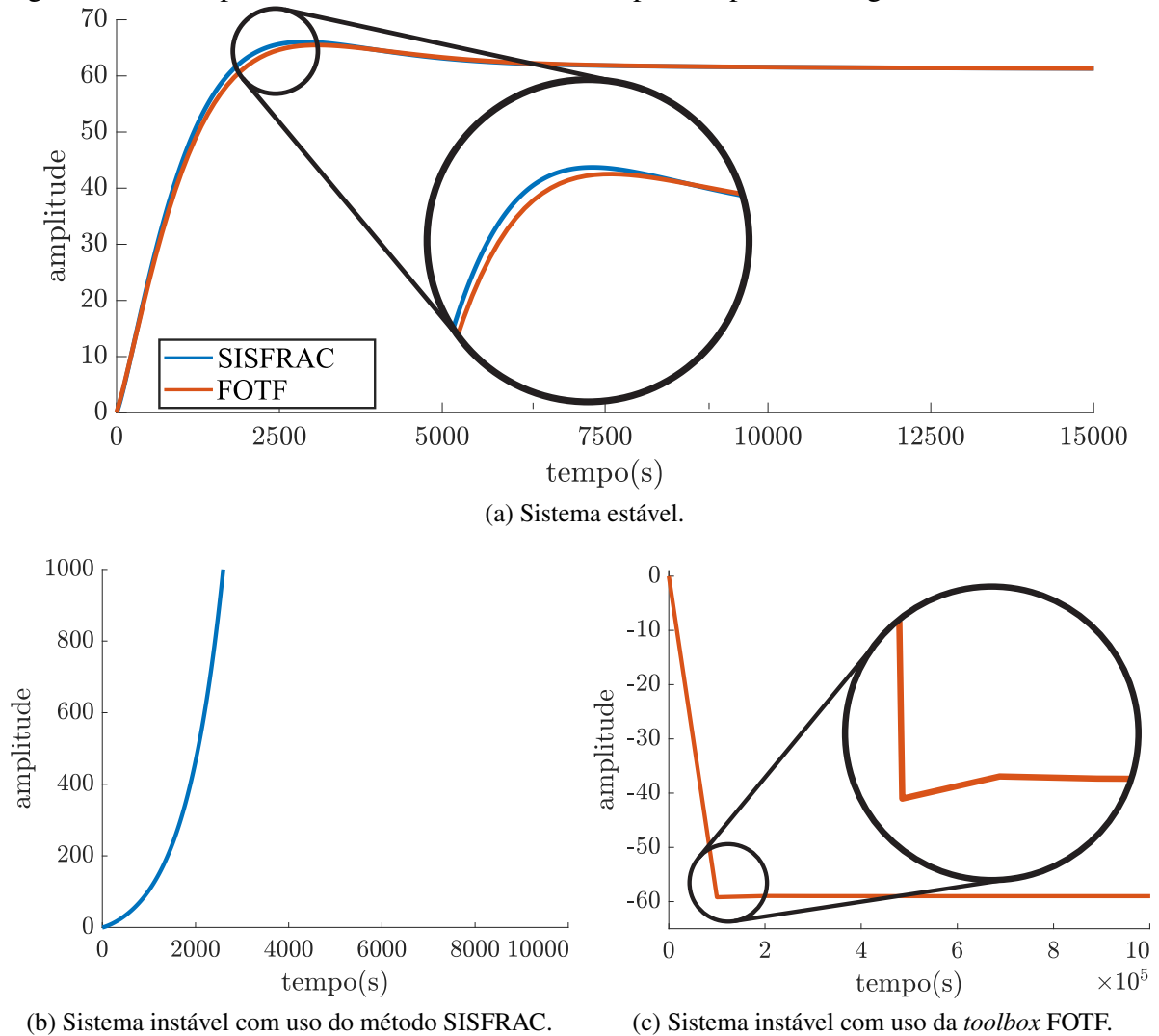
método	FOTF	Oustaloup	Carlson	Matsuda	SISFRAC
média (s)	39,74	0,0015	0,0015	0,0015	0,1559
desvio(s)	0,2333	0,0001	0,0001	0,0001	0,0047
RMSE	0	0,0031	0,0004	0,0118	$8 \cdot 10^{-16}$

Fonte: elaborada pelo autor.

astronômicos para fins de controle é uma tarefa árdua, sensível, e muitas vezes inviável. Por esta razão, o uso de filtros para aproximar funções de transferências de ordem fracionária é descartado nesta tese.

A resposta obtida pelo método SISFRAC possui RMSE nulo, indicando que a solução fechada apresentada na Equação 5.14 é exatamente a mesma da FOTF. Porém, o tempo médio de execução do método proposto é cerca de 255 vezes menor que o alcançado pela FOTF para o tempo de simulação utilizado. É importante salientar que esta diferença não é constante e depende do tempo de simulação. Diferenças ainda mais significativas podem ser observadas para tempos de simulação superiores. A razão da diferença entre esses dois métodos deve-se à equação utilizada na SISFRAC para a solução fechada ser específica para a função de transferência em questão. Este é um ganho substancial, uma vez que a quantidade de simulações de um sistema

Figura 32 – Exemplo de mau uso da *toolbox* FOTF para resposta ao degrau.



Fonte: elaborada pelo autor.

de controle de ordem fracionária durante sua fase de projeto pode ser bastante significativa e, consequentemente, muito dispendioso em custo computacional e tempo de simulação.

Por este motivo, o método proposto SISFRAC será utilizado neste trabalho para as simulações computacionais de resposta temporal para sistemas de ordem fracionária. Para o leitor que preferir utilizar a *toolbox* FOTF, recomenda-se bastante cautela e conhecimento sobre a mesma, pois caso mal utilizada, a FOTF também pode levar a resultados errados. Como exemplo do mau uso da *toolbox*, são ilustrados dois casos na Figura 32, em que foi constatado o uso incorreto da função `step()`.

Na Figura 32a, são apresentadas as respostas ao degrau de um mesmo sistema de controle estável. Observa-se que mesmo próximas, a resposta da *toolbox* diverge um pouco da resposta do método SISFRAC. Essa divergência é agravada nas Figura 32b e Figura 32c, que

apresentam as respostas ao degrau de um mesmo sistema de controle instável. A resposta correta do sistema instável, obtida pelo método SISFRAC, diverge da resposta fornecida pela FOTF.

Visando a simulação computacional, a equação temporal discreta do sistema simples apresentada na [Seção 5.1](#) é estendida para uma função de transferência generalista, cujo desenvolvimento é apresentado a seguir. Por ser uma função de transferência generalista, a mesma pode ser utilizada para simulação de plantas e controladores de ordem fracionária, com as devidas adaptações. Baseado nesta função generalista, é derivada uma equação temporal discreta do controlador $PI^\lambda D^\mu$ na [Seção 5.4](#). Um segundo desenvolvimento para este controlador é realizado na [Seção 5.5](#), abordando o uso de um somatório duplo. Por fim, a comparação entre as duas versões de equação discreta do $PI^\lambda D^\mu$ é realizada na [Seção 5.6](#).

5.3 Desenvolvimento de uma equação discreta de uso geral

Seja a função de transferência de ordem fracionária genérica

$$\frac{Y(s)}{U(s)} = \frac{fs^\gamma + \dots + gs^\delta + c}{bs^\alpha + \dots + ds^\beta + a}, \quad (5.18)$$

em que $\gamma \dots \delta, \alpha \dots \beta \in \mathbb{R}^+$ e $f \dots g, c, b \dots d, a \in \mathbb{R}$. Aplicando a definição de Grünwald-Letnikov de derivada fracionária desenvolvida na [Subseção 2.1.1](#) (vide [Equação 2.18](#)), tem-se

$$\frac{Y(z)}{U(z)} = \frac{f \left(\frac{1-z^{-1}}{T} \right)^\gamma + \dots + g \left(\frac{1-z^{-1}}{T} \right)^\delta + c}{b \left(\frac{1-z^{-1}}{T} \right)^\alpha + \dots + d \left(\frac{1-z^{-1}}{T} \right)^\beta + a} \quad (5.19)$$

$$\frac{Y(z)}{U(z)} = \frac{\frac{fT^{\delta+\dots} (1-z^{-1})^\gamma + \dots + gT^{\gamma+\dots} (1-z^{-1})^\delta + cT^{\gamma+\dots+\delta}}{T^{\gamma+\dots+\delta}}}{\frac{bT^{\beta+\dots} (1-z^{-1})^\alpha + \dots + dT^{\alpha+\dots} (1-z^{-1})^\beta + aT^{\alpha+\dots+\beta}}{T^{\alpha+\dots+\beta}}} \quad (5.20)$$

$$\frac{Y(z)}{U(z)} = \frac{fT^{\delta+\dots} (1-z^{-1})^\gamma + \dots + gT^{\gamma+\dots} (1-z^{-1})^\delta + cT^{\gamma+\dots+\delta}}{bT^{\beta+\dots} (1-z^{-1})^\alpha + \dots + dT^{\alpha+\dots} (1-z^{-1})^\beta + aT^{\alpha+\dots+\beta}} T^{\alpha+\dots+\beta-\gamma-\dots-\delta} \quad (5.21)$$

$$\frac{Y(z)}{U(z)} = \frac{fT^{\alpha+\dots+\beta-\dots-\gamma} (1-z^{-1})^\gamma + \dots + gT^{\alpha+\dots+\beta-\dots-\delta} (1-z^{-1})^\delta + cT^{\alpha+\dots+\beta}}{bT^{\beta+\dots} (1-z^{-1})^\alpha + \dots + dT^{\alpha+\dots} (1-z^{-1})^\beta + aT^{\alpha+\dots+\beta}}. \quad (5.22)$$

Aplicando a multiplicação cruzada,

$$\begin{aligned} & \left[bT^{\beta+\dots} (1-z^{-1})^{\alpha} + \dots + dT^{\alpha+\dots} (1-z^{-1})^{\beta} + aT^{\alpha+\dots+\beta} \right] Y(z) = \\ & \left[fT^{\alpha+\dots+\beta+\dots-\gamma} (1-z^{-1})^{\gamma} + \dots + gT^{\alpha+\dots+\beta+\dots-\delta} (1-z^{-1})^{\delta} + cT^{\alpha+\dots+\beta} \right] U(z) \end{aligned} \quad (5.23)$$

$$\begin{aligned} & bT^{\beta+\dots} (1-z^{-1})^{\alpha} Y(z) + \dots + dT^{\alpha+\dots} (1-z^{-1})^{\beta} Y(z) + aT^{\alpha+\dots+\beta} Y(z) = \\ & fT^{\alpha+\dots+\beta+\dots-\gamma} (1-z^{-1})^{\gamma} U(z) + \dots + gT^{\alpha+\dots+\beta+\dots-\delta} (1-z^{-1})^{\delta} U(z) + cT^{\alpha+\dots+\beta} U(z). \end{aligned} \quad (5.24)$$

Aplicando a forma compacta da definição de Grünwald-Letnikov de derivada fracionária desenvolvida na [Subseção 2.1.1](#) (vide [Equação 2.18](#)), obtêm-se

$$\begin{aligned} & bT^{\beta+\dots} \left(\sum_{i=0}^{\infty} \alpha_{w_i} z^{-i} \right) Y(z) + \dots + dT^{\alpha+\dots} \left(\sum_{i=0}^{\infty} \beta_{w_i} z^{-i} \right) Y(z) + aT^{\alpha+\dots+\beta} Y(z) = \\ & fT^{\alpha+\dots+\beta+\dots-\gamma} \left(\sum_{i=0}^{\infty} \gamma_{w_i} z^{-i} \right) U(z) + \dots + gT^{\alpha+\dots+\beta+\dots-\delta} \left(\sum_{i=0}^{\infty} \delta_{w_i} z^{-i} \right) U(z) + cT^{\alpha+\dots+\beta} U(z) \end{aligned} \quad (5.25)$$

$$\begin{aligned} & bT^{\beta+\dots} \left(\alpha_{w_0} + \sum_{i=1}^{\infty} \alpha_{w_i} z^{-i} \right) Y(z) + \dots + dT^{\alpha+\dots} \left(\beta_{w_0} + \sum_{i=1}^{\infty} \beta_{w_i} z^{-i} \right) Y(z) + aT^{\alpha+\dots+\beta} Y(z) = \\ & fT^{\alpha+\dots+\beta+\dots-\gamma} \left(\gamma_{w_0} + \sum_{i=1}^{\infty} \gamma_{w_i} z^{-i} \right) U(z) + \dots + \\ & + gT^{\alpha+\dots+\beta+\dots-\delta} \left(\delta_{w_0} + \sum_{i=1}^{\infty} \delta_{w_i} z^{-i} \right) U(z) + cT^{\alpha+\dots+\beta} U(z) \end{aligned} \quad (5.26)$$

$$\begin{aligned} & \alpha_{w_0} bT^{\beta+\dots} Y(z) + bT^{\beta+\dots} \left(\sum_{i=1}^{\infty} \alpha_{w_i} z^{-i} Y(z) \right) + \dots + \\ & \beta_{w_0} dT^{\alpha+\dots} Y(z) + dT^{\alpha+\dots} \left(\sum_{i=1}^{\infty} \beta_{w_i} z^{-i} Y(z) \right) + aT^{\alpha+\dots+\beta} Y(z) = \\ & \gamma_{w_0} fT^{\alpha+\dots+\beta+\dots-\gamma} U(z) + fT^{\alpha+\dots+\beta+\dots-\gamma} \left(\sum_{i=1}^{\infty} \gamma_{w_i} z^{-i} U(z) \right) + \dots + \\ & \delta_{w_0} gT^{\alpha+\dots+\beta+\dots-\delta} U(z) + gT^{\alpha+\dots+\beta+\dots-\delta} \left(\sum_{i=1}^{\infty} \delta_{w_i} z^{-i} U(z) \right) + cT^{\alpha+\dots+\beta} U(z) \end{aligned} \quad (5.27)$$

$$\begin{aligned}
& \left({}^{\alpha}w_0 b T^{\beta+\dots} + \dots + {}^{\beta}w_0 d T^{\alpha+\dots} + a T^{\alpha+\dots+\beta} \right) Y(z) + \\
& + b T^{\beta+\dots} \left(\sum_{i=1}^{\infty} {}^{\alpha}w_i z^{-i} Y(z) \right) + \dots + d T^{\alpha+\dots} \left(\sum_{i=1}^{\infty} {}^{\beta}w_i z^{-i} Y(z) \right) = \\
& \left(\gamma_{w_0} f T^{\alpha+\dots+\beta-\dots-\gamma} + \dots + \delta_{w_0} g T^{\alpha+\dots+\beta-\dots-\delta} + c T^{\alpha+\dots+\beta} \right) U(z) + \\
& + f T^{\alpha+\dots+\beta-\dots-\gamma} \left(\sum_{i=1}^{\infty} \gamma_{w_i} z^{-i} U(z) \right) + \dots + g T^{\alpha+\dots+\beta-\dots-\delta} \left(\sum_{i=1}^{\infty} \delta_{w_i} z^{-i} U(z) \right),
\end{aligned} \tag{5.28}$$

em que ${}^{\alpha}w_i$, ${}^{\beta}w_i$, γ_{w_i} e δ_{w_i} são os parâmetros relacionados às derivadas fracionárias de ordem α , β , γ e δ , respectivamente. Aplicando a transformada para tempo discreto, chega-se a

$$\begin{aligned}
& \left({}^{\alpha}w_0 b T^{\beta+\dots} + \dots + {}^{\beta}w_0 d T^{\alpha+\dots} + a T^{\alpha+\dots+\beta} \right) y[k] + \\
& + b T^{\beta+\dots} \left(\sum_{i=1}^{\infty} {}^{\alpha}w_i y[k-i] \right) + \dots + d T^{\alpha+\dots} \left(\sum_{i=1}^{\infty} {}^{\beta}w_i y[k-i] \right) = \\
& \left(\gamma_{w_0} f T^{\alpha+\dots+\beta-\dots-\gamma} + \dots + \delta_{w_0} g T^{\alpha+\dots+\beta-\dots-\delta} + c T^{\alpha+\dots+\beta} \right) u[k] + \\
& + f T^{\alpha+\dots+\beta-\dots-\gamma} \left(\sum_{i=1}^{\infty} \gamma_{w_i} u[k-i] \right) + \dots + g T^{\alpha+\dots+\beta-\dots-\delta} \left(\sum_{i=1}^{\infty} \delta_{w_i} u[k-i] \right)
\end{aligned} \tag{5.29}$$

$$\begin{aligned}
& \left({}^{\alpha}w_0 b T^{\beta+\dots} + \dots + {}^{\beta}w_0 d T^{\alpha+\dots} + a T^{\alpha+\dots+\beta} \right) y[k] = \\
& \left(\gamma_{w_0} f T^{\alpha+\dots+\beta-\dots-\gamma} + \dots + \delta_{w_0} g T^{\alpha+\dots+\beta-\dots-\delta} + c T^{\alpha+\dots+\beta} \right) u[k] + \\
& + f T^{\alpha+\dots+\beta-\dots-\gamma} \left(\sum_{i=1}^{\infty} \gamma_{w_i} u[k-i] \right) + \dots + g T^{\alpha+\dots+\beta-\dots-\delta} \left(\sum_{i=1}^{\infty} \delta_{w_i} u[k-i] \right) + \\
& - b T^{\beta+\dots} \left(\sum_{i=1}^{\infty} {}^{\alpha}w_i y[k-i] \right) - \dots - d T^{\alpha+\dots} \left(\sum_{i=1}^{\infty} {}^{\beta}w_i y[k-i] \right).
\end{aligned} \tag{5.30}$$

Assim, isolando $y[k]$, obtém-se

$$\begin{aligned}
y[k] = & \frac{1}{\left({}^{\alpha}w_0 b T^{\beta+\dots} + \dots + {}^{\beta}w_0 d T^{\alpha+\dots} + a T^{\alpha+\dots+\beta} \right)} \times \left[\right. \\
& \left(\gamma_{w_0} f T^{\alpha+\dots+\beta-\dots-\gamma} + \dots + \delta_{w_0} g T^{\alpha+\dots+\beta-\dots-\delta} + c T^{\alpha+\dots+\beta} \right) u[k] + \\
& + f T^{\alpha+\dots+\beta-\dots-\gamma} \left(\sum_{i=1}^{\infty} \gamma_{w_i} u[k-i] \right) + \dots + g T^{\alpha+\dots+\beta-\dots-\delta} \left(\sum_{i=1}^{\infty} \delta_{w_i} u[k-i] \right) + \\
& \left. - b T^{\beta+\dots} \left(\sum_{i=1}^{\infty} {}^{\alpha}w_i y[k-i] \right) - \dots - d T^{\alpha+\dots} \left(\sum_{i=1}^{\infty} {}^{\beta}w_i y[k-i] \right) \right].
\end{aligned} \tag{5.31}$$

Analogamente à [Equação 5.14](#), a [Equação 5.31](#) pode ser reescrita (visando a simulação computacional) considerando nulos os infinitos valores passados em que $i > k$, chegando-se ao seguinte resultado:

$$y[k]_{k=2:t_{final}} = \frac{1}{(\alpha_{w_0} b T^{\beta+\dots} + \dots + \beta_{w_0} d T^{\alpha+\dots} + a T^{\alpha+\dots+\beta})} \times \left[\begin{aligned} & (\gamma_{w_0} f T^{\alpha+\dots+\beta+\dots-\gamma} + \dots + \delta_{w_0} g T^{\alpha+\dots+\beta+\dots-\delta} + c T^{\alpha+\dots+\beta}) u[k] + \\ & + f T^{\alpha+\dots+\beta+\dots-\gamma} \left(\sum_{i=1}^{k-1} \gamma_{w_i} u[k-i] \right) + \dots + g T^{\alpha+\dots+\beta+\dots-\delta} \left(\sum_{i=1}^{k-1} \delta_{w_i} u[k-i] \right) + \\ & - b T^{\beta+\dots} \left(\sum_{i=1}^{k-1} \alpha_{w_i} y[k-i] \right) - \dots - d T^{\alpha+\dots} \left(\sum_{i=1}^{k-1} \beta_{w_i} y[k-i] \right) \end{aligned} \right]. \quad (5.32)$$

5.4 Discretização do controlador $PI^\lambda D^\mu$ — versão 1

Uma vez que a [Equação 5.18](#) é uma função de transferência genérica, a mesma pode ser utilizada para representar diversos sistemas, como plantas, filtros e controladores, desde que os coeficientes sejam adequados. Como exemplo, a equação do controlador $PI^\lambda D^\mu$ será ajustada conforme

$$G_{PI^\lambda D^\mu}(s) = k_p + \frac{k_i}{s^\lambda} + k_d s^\mu = \frac{k_p s^\lambda + k_i + k_d s^{\mu+\lambda}}{s^\lambda}. \quad (5.33)$$

Para ajustar a [Equação 5.33](#) à [Equação 5.18](#), é necessário fazer

$$\frac{k_p s^\lambda + k_i + k_d s^{\mu+\lambda}}{s^\lambda} \longrightarrow \frac{f s^\gamma + \dots + g s^\delta + c}{b s^\alpha + \dots + d s^\beta + a}, \quad (5.34)$$

em que

$$\begin{aligned} a &= 0 & b &= 1 & c &= k_i & d &= 0 & f &= k_p \\ g &= k_d & \alpha &= \lambda & \beta &= 0 & \gamma &= \lambda & \delta &= \mu + \lambda, \end{aligned}$$

e todos os demais coeficientes da versão genérica são nulos. Aplicando estas condições na [Equação 5.32](#), e considerando a entrada e saída do controlador respectivamente como o sinal de erro $e[k]$ e a variável de controle $u[k]$,

$$\begin{aligned}
u[k]_{k=2:t_{final}} = & \frac{1}{\left(\lambda_{w_0} 1 + \cancel{0_{w_0} 0 T^{\lambda+0}} + \cancel{0 T^{\lambda+0}} \right)^0} \left[\left(\lambda_{w_0} k_p \cancel{T^{\lambda+0-\lambda}} + \mu+\lambda_{w_0} k_d \cancel{T^{\lambda+0-\mu-\lambda}} + k_i T^{\lambda+0} \right) e[k] + \right. \\
& \left. + k_p \cancel{T^{\lambda+0-\lambda}} \left(\sum_{i=1}^{k-1} \lambda_{w_i} e[k-i] \right) + k_d \cancel{T^{\lambda+0-\mu-\lambda}} \left(\sum_{i=1}^{k-1} \mu+\lambda_{w_i} e[k-i] \right) - \cancel{1 T^0} \left(\sum_{i=1}^{k-1} \lambda_{w_i} u[k-i] \right) - \cancel{0 T^{\lambda}} \left(\sum_{i=1}^{k-1} \cancel{0_{w_i} u[k-i]} \right) \right]^0
\end{aligned} \tag{5.35}$$

$$\begin{aligned}
u[k]_{k=2:t_{final}} = & \frac{1}{\lambda_{w_0}} \left[\left(\lambda_{w_0} k_p + \mu+\lambda_{w_0} \frac{k_d}{T^\mu} + k_i T^\lambda \right) e[k] + k_p \left(\sum_{i=1}^{k-1} \lambda_{w_i} e[k-i] \right) + \right. \\
& \left. + \frac{k_d}{T^\mu} \left(\sum_{i=1}^{k-1} \mu+\lambda_{w_i} e[k-i] \right) - \left(\sum_{i=1}^{k-1} \lambda_{w_i} u[k-i] \right) \right].
\end{aligned} \tag{5.36}$$

Os parâmetros λ_{w_i} e $\mu+\lambda_{w_i}$ são coeficientes relacionados às derivadas fracionárias de ordem λ e $\mu + \lambda$, respectivamente, e podem ser calculados recursivamente conforme a [Equação 2.24](#).

5.5 Discretização do controlador $\text{PI}^\lambda \text{D}^\mu$ — versão 2

Aplicando a definição compacta de Grünwald-Letnikov de derivada fracionária desenvolvida na [Subseção 2.1.1](#) (vide [Equação 2.18](#)) diretamente na expressão do $\text{PI}^\lambda \text{D}^\mu$,

$$\frac{U(s)}{E(s)} = k_p + \frac{k_i}{s^\lambda} + k_d s^\mu,$$

$$\frac{U(z)}{E(z)} = k_p + \frac{k_i}{\left(\frac{1-z^{-1}}{T} \right)^\lambda} + k_d \left(\frac{1-z^{-1}}{T} \right)^\mu = k_p + \frac{k_i T^\lambda}{(1-z^{-1})^\lambda} + \frac{k_d (1-z^{-1})^\mu}{T^\mu}, \tag{5.37}$$

$$\frac{U(z)}{E(z)} = k_p + \frac{k_i T^\lambda}{\sum_{i=0}^{\infty} \lambda_{w_i} z^{-i}} + \frac{k_d}{T^\mu} \sum_{i=0}^{\infty} \mu_{w_i} z^{-i} \tag{5.38}$$

$$\frac{U(z)}{E(z)} = k_p + \frac{k_i T^\lambda}{\lambda_{w_0} + \sum_{i=1}^{\infty} \lambda_{w_i} z^{-i}} + \frac{k_d}{T^\mu} \left(\mu_{w_0} + \sum_{i=1}^{\infty} \mu_{w_i} z^{-i} \right) \quad (5.39)$$

$$\frac{U(z)}{E(z)} = \frac{k_p T^\mu \left(\lambda_{w_0} + \sum_{i=1}^{\infty} \lambda_{w_i} z^{-i} \right) + k_i T^\lambda T^\mu + k_d \left(\mu_{w_0} + \sum_{i=1}^{\infty} \mu_{w_i} z^{-i} \right) \left(\lambda_{w_0} + \sum_{i=1}^{\infty} \lambda_{w_i} z^{-i} \right)}{\left(\lambda_{w_0} + \sum_{i=1}^{\infty} \lambda_{w_i} z^{-i} \right) T^\mu} \quad (5.40)$$

Multiplicando de forma cruzada, chega-se a

$$\begin{aligned} \lambda_{w_0} T^\mu U(z) + T^\mu \left(\sum_{i=1}^{\infty} \lambda_{w_i} z^{-i} U(z) \right) &= \left(\lambda_{w_0} k_p T^\mu + k_i T^\lambda T^\mu \right) E(z) + k_p T^\mu \left(\sum_{i=1}^{\infty} \lambda_{w_i} z^{-i} E(z) \right) + \\ &+ k_d E(z) \left(\mu_{w_0} + \sum_{i=1}^{\infty} \mu_{w_i} z^{-i} \right) \left(\lambda_{w_0} + \sum_{i=1}^{\infty} \lambda_{w_i} z^{-i} \right) \end{aligned} \quad (5.41)$$

$$\begin{aligned} \lambda_{w_0} T^\mu U(z) &= \left(\lambda_{w_0} k_p T^\mu + k_i T^{\lambda+\mu} \right) E(z) + k_p T^\mu \left(\sum_{i=1}^{\infty} \lambda_{w_i} z^{-i} E(z) \right) + k_d E(z) \left[\mu_{w_0} \lambda_{w_0} + \mu_{w_0} \left(\sum_{i=1}^{\infty} \lambda_{w_i} z^{-i} \right) + \right. \\ &\quad \left. + \lambda_{w_0} \left(\sum_{i=1}^{\infty} \mu_{w_i} z^{-i} \right) + \left(\sum_{i=1}^{\infty} \mu_{w_i} z^{-i} \right) \left(\sum_{i=1}^{\infty} \lambda_{w_i} z^{-i} \right) \right] - T^\mu \left(\sum_{i=1}^{\infty} \lambda_{w_i} z^{-i} U(z) \right) \end{aligned} \quad (5.42)$$

$$\begin{aligned} \lambda_{w_0} T^\mu U(z) &= \left(\lambda_{w_0} k_p T^\mu + k_i T^{\lambda+\mu} + k_d \mu_{w_0} \lambda_{w_0} \right) E(z) + k_p T^\mu \left(\sum_{i=1}^{\infty} \lambda_{w_i} z^{-i} E(z) \right) + k_d E(z) \left[\mu_{w_0} \left(\sum_{i=1}^{\infty} \lambda_{w_i} z^{-i} \right) + \right. \\ &\quad \left. + \lambda_{w_0} \left(\sum_{i=1}^{\infty} \mu_{w_i} z^{-i} \right) + \left(\sum_{i=1}^{\infty} \sum_{j=1}^{\infty} \mu_{w_i} \lambda_{w_j} z^{-i-j} \right) \right] - T^\mu \left(\sum_{i=1}^{\infty} \lambda_{w_i} z^{-i} U(z) \right), \end{aligned} \quad (5.43)$$

em que λ_{w_i} e μ_{w_i} são os parâmetros relacionados às derivadas fracionárias de ordem λ e μ , respectivamente. Aplicando a transformada para tempo discreto,

$$\begin{aligned} \lambda_{w_0} T^\mu u[k] = & \left(\lambda_{w_0} k_p T^\mu + k_i T^{\lambda+\mu} + \frac{k_d \mu_{w_0} \lambda_{w_0}}{T^\mu} \right) e[k] + k_p T^\mu \left(\sum_{i=1}^{\infty} \lambda_{w_i} e[k-i] \right) + \frac{k_d}{T^\mu} \left[\mu_{w_0} \left(\sum_{i=1}^{\infty} \lambda_{w_i} e[k-i] \right) + \right. \\ & \left. + \lambda_{w_0} \left(\sum_{i=1}^{\infty} \mu_{w_i} e[k-i] \right) + \left(\sum_{i=1}^{\infty} \sum_{j=1}^{\infty} \mu_{w_i} \lambda_{w_j} e[k-i-j] \right) \right] - T^\mu \left(\sum_{i=1}^{\infty} \lambda_{w_i} u[k-i] \right) \end{aligned} \quad (5.44)$$

Por fim, chega-se a

$$\begin{aligned} u[k] = \frac{1}{\lambda_{w_0}} \left\{ \left(\lambda_{w_0} k_p + k_i T^\lambda + \frac{k_d \mu_{w_0} \lambda_{w_0}}{T^\mu} \right) e[k] + k_p \left(\sum_{i=1}^{\infty} \lambda_{w_i} e[k-i] \right) + \frac{k_d}{T^\mu} \left[\mu_{w_0} \left(\sum_{i=1}^{\infty} \lambda_{w_i} e[k-i] \right) + \right. \right. \\ \left. \left. + \lambda_{w_0} \left(\sum_{i=1}^{\infty} \mu_{w_i} e[k-i] \right) + \left(\sum_{i=1}^{\infty} \sum_{j=1}^{\infty} \mu_{w_i} \lambda_{w_j} e[k-i-j] \right) \right] - \left(\sum_{i=1}^{\infty} \lambda_{w_i} u[k-i] \right) \right\}. \end{aligned} \quad (5.45)$$

A [Equação 5.45](#) considera todos os infinitos valores passados do erro $e[k]$ e da saída de controle $u[k]$. Em relação à simulação computacional, os termos $e[i > k]$, $e[i + j > k]$ e $u[i > k]$ não são possíveis de serem computados. Como mencionado anteriormente, pode-se considerar nulos os infinitos valores passados em que isto acontece. Assim, o sistema só parte do estado nulo no início da simulação. Desta forma, a [Equação 5.45](#) pode ser reescrita, visando a simulação computacional, como

$$\begin{aligned} u[k]_{k=3:t_{final}} = \frac{1}{\lambda_{w_0}} \left\{ \left(\lambda_{w_0} k_p + k_i T^\lambda + \frac{k_d \mu_{w_0} \lambda_{w_0}}{T^\mu} \right) e[k] + k_p \left(\sum_{i=1}^{k-1} \lambda_{w_i} e[k-i] \right) + \frac{k_d}{T^\mu} \left[\mu_{w_0} \left(\sum_{i=1}^{k-1} \lambda_{w_i} e[k-i] \right) + \right. \right. \\ \left. \left. + \lambda_{w_0} \left(\sum_{i=1}^{k-1} \mu_{w_i} e[k-i] \right) + \left(\sum_{i=1}^{\lceil \frac{k-2}{2} \rceil} \sum_{j=1}^{\lceil \frac{k-2}{2} \rceil} \mu_{w_i} \lambda_{w_j} e[k-i-j] \right) \right] - \left(\sum_{i=1}^{k-1} \lambda_{w_i} u[k-i] \right) \right\}. \end{aligned} \quad (5.46)$$

Os parâmetros λ_{w_i} e μ_{w_i} são calculados recursivamente conforme a [Equação 2.24](#).

5.6 Comparação entre as duas versões de discretização do controlador $PI^\lambda D^\mu$

As duas versões do controlador $PI^\lambda D^\mu$ desenvolvidas, a saber, [Equação 5.36](#) e [Equação 5.46](#), são equivalentes entre si. Tem-se também que a equação do PID convencional discretizada por Euler *a priori* é um caso particular destas duas equações, como era de se esperar (vide desenvolvimento da [Equação 5.10](#) e [Equação 5.13](#)).

Para comparar o desempenho das duas versões, é realizada a simulação do seguinte sistema de controle de ordem fracionária. Seja a planta $G_p(z)$ dada por

$$G_p(z) = \frac{0,0044 + 0,0455z^{-1} - 0,0476z^{-2}}{1 - 1,5531z^{-1} + 0,5621z^{-2}}, \quad (5.47)$$

e o controlador $PI^\lambda D^\mu$ cujos parâmetros são

$$k_p = 0,0022 \quad k_i = 0,0959 \quad k_d = 0,0973 \quad (5.48)$$

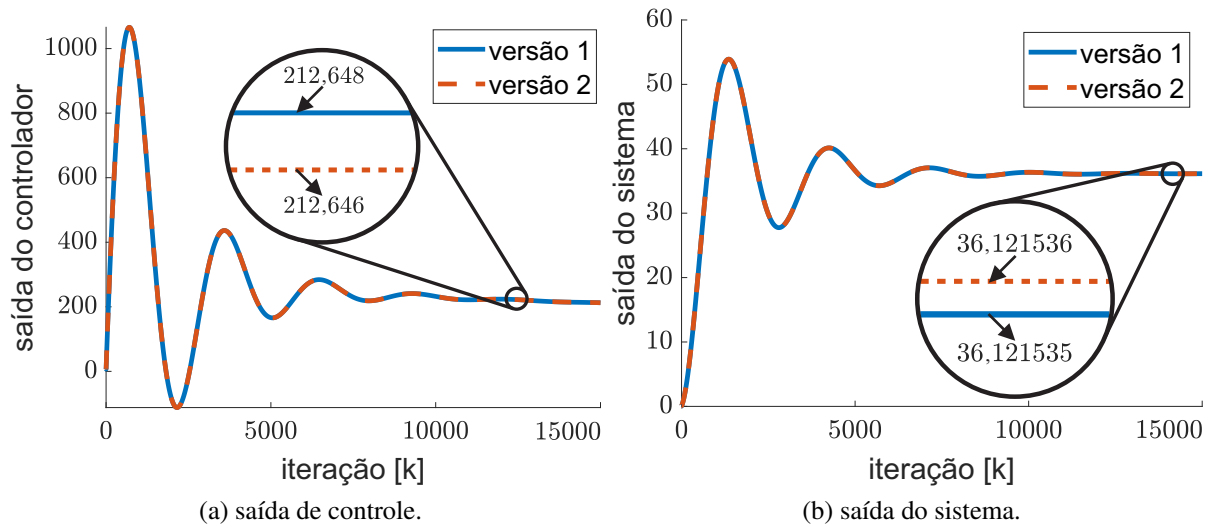
$$\lambda = 0,9519 \quad \mu = 0,9999, \quad (5.49)$$

em que o passo de cálculo é 1 s e o ganho de realimentação do sistema de controle é unitário. As saídas dos controladores e as respostas do sistema submetido a um degrau são ilustradas na [Figura 33](#), em que o sistema é submetido às duas versões do controlador $PI^\lambda D^\mu$ discretizado.

Conforme esperado, as respostas dos sistemas e suas saídas de controle são bastante similares. De fato, o RMSE entre as saídas de controle é de 0,1279, enquanto que o RMSE das respostas do sistema é de 0,0043. Isso equivale a uma diferença na terceira casa decimal para a saída de controle e na sexta casa decimal na saída do sistema. Estas pequenas diferenças se devem ao uso de $\lceil \cdot \rceil$ (arredondamento para cima) na [Equação 5.46](#). O limite superior $\lceil \frac{k-2}{2} \rceil$ é uma estratégia que evita índices inválidos, $e[i+j > k]$, garantindo que os índices de $e[\cdot]$ sejam todos válidos. Porém, esta estratégia pode excluir termos que seriam relevantes para a dinâmica completa do sistema. Essas variações discretas no número de termos podem levar a pequenos erros em relação ao comportamento desejado do sistema.

O sistema de controle foi executado 30 vezes. O tempo médio em segundos e seu desvio padrão são apresentados na [Tabela 11](#). As especificações da máquina utilizada para as simulações nesta tese são apresentadas no [Apêndice A](#). A grande diferença entre as duas versões está no tempo de processamento. A versão 1 contém um somatório duplo, o que é bastante dispendioso computacionalmente. A razão entre os tempos de processamento para esta simulação é de cerca de 950 vezes. Isso faz bastante diferença, pois como dito anteriormente, a quantidade de simulações de um sistema de controle de ordem fracionária durante sua fase de projeto pode

Figura 33 – Simulação de um sistema com duas diferentes discretizações do controlador $PI^\lambda D^\mu$.



Fonte: elaborada pelo autor.

Tabela 11 – Média e desvio padrão do tempo de execução.

versão da discretização	média e desvio padrão (s)
versão 1	$707,692 \pm 5,978$
versão 2	$0,748 \pm 0,012$

Fonte: elaborada pelo autor.

ser bastante significativa e, conseqüentemente, muito dispendioso em custo computacional e tempo de simulação. Por este motivo, todas as simulações de quaisquer funções de transferência de ordem fracionária presentes neste trabalho, a partir deste ponto, serão executadas a partir da equação genérica (vide [Equação 5.18](#)) adaptada à função de transferência que se deseja simular, sejam estas controladores $PI^\lambda D^\mu$ (vide [Equação 5.36](#)), plantas ou sistemas em malha fechada.

5.7 Considerações finais

As simulações realizadas neste capítulo permitiram desenvolver soluções computacionalmente viáveis para sistemas dinâmicos de ordem fracionária, com destaque para o método proposto SISFRAC, que apresentou desempenho superior em tempo de execução e precisão quando comparado a *softwares* disponíveis na literatura da área.

Os resultados obtidos reforçam a aplicabilidade do método SISFRAC em diversas etapas envolvidas na síntese de controladores fracionários, incluindo a simulação da resposta ao degrau de sistemas fracionários em malha fechada realizada durante a sintonia dos parâmetros do controlador, as simulações voltadas à implementação embarcada e, por fim, os procedimentos de identificação da planta, tema abordado no próximo capítulo.

6 IDENTIFICAÇÃO DE SISTEMAS DE ORDEM FRACIONÁRIA

A identificação de sistemas é uma etapa fundamental na prática de controle, responsável pela determinação de um modelo matemático representativo do comportamento do sistema. Em sistemas térmicos com convecção, essa etapa torna-se particularmente desafiadora devido à presença de parâmetros físicos, como o coeficiente de transferência de calor, que raramente são conhecidos com precisão e podem variar ao longo do tempo em razão de alterações físicas ou químicas na superfície de troca térmica, a capacidade térmica do corpo condutor, sua área superficial e sua difusividade térmica (DAS, 2011). Diante dessas dificuldades, abordagens de identificação baseadas em modelos de caixa-preta, que evitam a estimação direta de tais parâmetros, mostram-se vantajosas.

Para representar com maior fidelidade o comportamento dinâmico de sistemas físicos reais, generalizações de modelos têm sido propostas ao substituir as derivadas de ordem inteira por derivadas fracionárias, como as apresentadas na Seção 2.3, refletindo características intrínsecas de certos materiais e ampliando a capacidade descritiva dos modelos, como discutido em Torvik e Bagley (1984), que dissertam sobre o surgimento de derivadas fracionárias no comportamento dos materiais. Em relação à transferência de calor, as equações generalizadas para ordens fracionárias são apresentadas em Vázquez *et al.* (2011) sob a ótica da segunda lei da termodinâmica. Diversos autores realizam a modelagem de sistemas térmicos por modelos de ordem fracionária, como Petráš (2011) e Oprzędkiewicz *et al.* (2021).

Em geral, as principais metodologias de identificação de sistemas fracionários são divididas em modelos baseados no erro da equação (*equation-error-based models*) e modelos baseados no erro da saída (*output-error-based models*). A principal limitação dos modelos baseados em erro de equação é a necessidade de conhecimento prévio das ordens fracionárias, enquanto os modelos baseados em erro de saída podem identificar simultaneamente as ordens fracionárias e os coeficientes. (TASHAKORI *et al.*, 2024).

Entre os métodos baseados no erro da saída, destaca-se o apresentado em Poinot e Trigeassou (2004), que propõe a estimação dos coeficientes e das ordens fracionárias. Tal método atualiza os parâmetros através de cálculos recursivos que envolvem as matrizes jacobiana e hessiana resultante da otimização de uma função custo por alguma técnica de programação não linear.

Porém, os operadores fracionários presentes no modelo são aproximados por uma cascata de N filtros avanço de fase, correspondendo a operadores inteiros. Esta abordagem

possui algumas limitações, como o fato de o operador fracionário não ser implementado com base em definições matemáticas rigorosas de operadores fracionários, como as apresentadas na [Seção 2.1](#), mas sim por meio de uma aproximação composta por operadores inteiros de ordem N , cuja precisão depende diretamente deste parâmetro¹. Além disso, como os próprios autores apontam, a validade da aproximação é limitada a uma faixa de frequência restrita, cujos extremos também são tratados como parâmetros a serem estimados. A ordem fracionária do modelo, portanto, não é estimada de modo direto, mas inferida indiretamente a partir destes limites de frequência estimados. Ademais, para modelos que possuem mais de um operador fracionário, é preciso assumir condições *a priori* a respeito dos expoentes fracionários, o que exige um conhecimento prévio da planta. Por fim, devido à natureza local da aproximação utilizada, o uso dessa abordagem consequentemente resulta na não preservação de uma característica fundamental dos operadores fracionários: a memória infinita.

Um método amplamente utilizado em identificação de sistemas de ordem inteira é o algoritmo LMS (*least mean squares*), reconhecido pela sua robustez, eficácia e sua simplicidade computacional, com atualizações recursivas baseadas apenas em operações aritméticas simples. Uma de suas principais vantagens é a capacidade de estimar diretamente os parâmetros do modelo, sem depender de aproximações intermediárias. Essa característica torna o LMS especialmente atrativo para aplicações em que se busca um processo de estimação eficaz e com baixo custo computacional ([HAYKIN, 2013](#)).

Motivado pela eficácia e simplicidade computacional do algoritmo LMS, investigou-se a possibilidade de estimar os parâmetros do modelo fracionário por meio desse algoritmo de forma direta, sem a necessidade de aproximações intermediárias. O restante desta seção está dividido da seguinte forma. Na [Seção 6.1](#), apresentam-se a inspiração e os aspectos conceituais que deram origem ao algoritmo proposto. Nas [seções 6.2 a 6.4](#), a proposta é apresentada e desenvolvida gradualmente em relação à complexidade do modelo da planta, culminando na [Seção 6.5](#), que apresenta o algoritmo para um modelo de uso geral. Considerações sobre a convergência do algoritmo proposto e considerações finais são apresentadas nas [seções 6.6 e 6.7](#), respectivamente.

¹Como já visto na [Seção 5.2](#), vide [Figura 31](#) e [Tabela 9](#), aproximações por modelos de ordem inteira podem trazer resultados imprecisos.

6.1 Inspiração para a criação do algoritmo de identificação proposto

Durante o desenvolvimento do método SISFRAC no [Capítulo 5](#) para simulação de sistemas e controladores de ordem fracionária, observou-se que o operador de Grünwald-Letnikov para derivada fracionária de ordem α de $y(t)$, ou seja, $s^\alpha Y(s)$, pode ser representado de forma discreta como um produto interno entre um vetor composto pelos pesos ${}^\alpha w_i$ e um vetor composto por amostras passadas de $y(t)$, para T suficientemente pequeno. Por exemplo, considerando $L_0 = 3$ na [Equação 5.15](#), tem-se a expressão

$$s^\alpha Y(s) \approx \frac{1}{T^\alpha} ({}^\alpha w_0 y[k] + {}^\alpha w_1 y[k-1] + {}^\alpha w_2 y[k-2] + {}^\alpha w_3 y[k-3]) \quad (6.1)$$

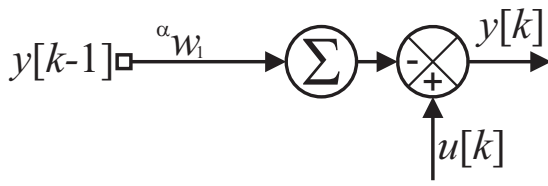
$$s^\alpha Y(s) \approx {}^{\alpha w'_0} y[k] + {}^{\alpha w'_1} y[k-1] + {}^{\alpha w'_2} y[k-2] + {}^{\alpha w'_3} y[k-3], \quad (6.2)$$

em que ${}^{\alpha w'} = \frac{{}^\alpha w}{T^\alpha}$. A [Equação 6.2](#) apresenta notável semelhança estrutural com modelos autor-regressivos (AR), cujos parâmetros podem ser eficientemente estimados pelo algoritmo LMS. Indo além, especulou-se sobre a possibilidade de um sistema fracionário SISO ser representado por um modelo ARX (autorregressivo com entradas exógenas) de ordem infinita. Partindo da premissa que esta representação é verdadeira, formulou-se a hipótese de identificação um sistema fracionário SISO através do algoritmo LMS, cuja formulação matemática é apresentada a seguir.

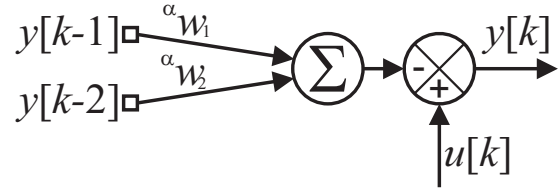
6.2 Caso 1: planta $G(s) = \frac{1}{s^\alpha}$

A princípio, para explorar a possibilidade da ideia proposta, desenvolve-se o algoritmo LMS para a função de transferência de ordem fracionária mais simples possível: a integral fracionária, que possui apenas a variável α , a qual representa a ordem da integração. Utilizando-se da discretização apresentada na [Capítulo 5](#), com a definição de Grünwald-Letnikov para derivadas fracionárias, tem-se

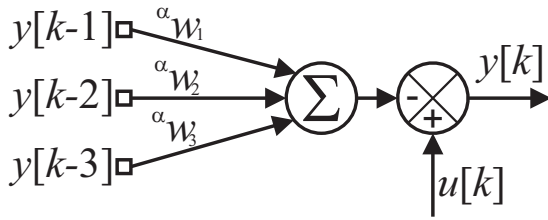
Figura 34 – Diagrama de blocos da Equação 6.3.



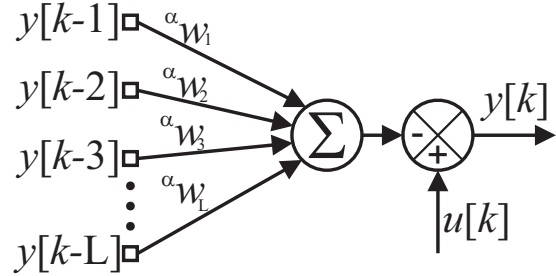
(a) Iteração em $k = 2$.



(b) Iteração em $k = 3$.



(c) Iteração em $k = 4$.



(d) Iteração em $k = L$.

Fonte: elaborada pelo autor.

$$U(s) = s^\alpha Y(s)$$

$$u[k] = \sum_{i=0}^L \alpha w_i y[k-i]$$

$$u[k] = \overset{1}{\alpha w_0 y[k]} + \sum_{i=1}^L \alpha w_i y[k-i]$$

$$y[k] = u[k] - \sum_{i=1}^L \alpha w_i y[k-i], \quad (6.3)$$

em que L é a quantidade de dados utilizado para identificação e os parâmetros αw_i são calculados recursivamente conforme a Equação 2.24. Nota-se que o número de componentes do somatório aumenta a cada iteração, conforme ilustrado na Figura 34, que apresenta um diagrama de blocos da Equação 6.3 para diferentes iterações.

Para facilitar a manipulação algébrica, a Equação 6.3 é reescrita como

$$y[k] = u[k] - \mathbf{W}^T[k] \mathbf{Y}[k], \quad (6.4)$$

em que $\mathbf{W}[k]$ e $\mathbf{Y}[k]$ são definidos como

$$\mathbf{W}[k] = \begin{bmatrix} \alpha w_1 \\ \alpha w_2 \\ \vdots \\ \alpha w_{k-1} \end{bmatrix}, \quad \mathbf{Y}[k] = \begin{bmatrix} y[k-1] \\ y[k-2] \\ \vdots \\ y[1] \end{bmatrix}, \quad k = 2, 3, \dots, L. \quad (6.5)$$

É importante ressaltar que, o número de elementos de $\mathbf{W}[k]$ e $\mathbf{Y}[k]$ também aumenta a cada iteração. Usualmente, utiliza-se algoritmo LMS para estimação dos parâmetros \mathbf{W} minimizando o erro quadrático instantâneo $J = \frac{1}{2}e^2$, em que $e = r - \hat{y}$ é o erro cometido entre os valores reais r e o estimado pelo modelo $\hat{y} = \mathbf{u} - \hat{\mathbf{W}}^T \mathbf{Y}$ utilizando o a expressão

$$\hat{\mathbf{W}}[k+1] = \hat{\mathbf{W}}[k] + \mu \frac{dJ[k]}{d\mathbf{W}}, \quad (6.6)$$

em que, pela regra da cadeia, o gradiente de J é fatorado como

$$\frac{dJ}{d\mathbf{W}} = \frac{\partial J}{\partial e} \frac{\partial e}{\partial y} \frac{dy}{d\mathbf{W}}. \quad (6.7)$$

Porém, devido ao fato de \mathbf{W} aumentar seus elementos a cada iteração, o processo de estimação torna-se progressivamente mais desafiador, em razão do acréscimo sucessivo de parâmetros ao modelo. Outra razão para não utilizar a busca direta do vetor \mathbf{W} é o fato que, desta forma, os valores de α_{w_i} que o compõe poderiam assumir quaisquer valores, uma vez que seriam independentes entre si. Porém, ainda que obtenha-se um conjunto de parâmetros α_{w_i} por essa abordagem, isso não garantiria a natureza fracionária do modelo, pois os parâmetros α_{w_i} devem seguir uma relação entre si, definida na [Equação 2.24](#) e reescrita aqui para facilitar o entendimento do processo:

$$w_i = \left(1 - \frac{\alpha + 1}{i}\right) w_{i-1}, \quad w_0 := 1.$$

Para contornar este problema, o diagrama de blocos presente na [Figura 34](#) pode ser redesenhado conforme a [Figura 35](#), em que todos os parâmetros α_{w_i} são escritos em função de α . Aproveitando-se da existência da expressão recursiva de α_{w_i} reapresentada acima, em que α_{w_i} depende de $\alpha_{w_{i-1}}$ e α , pode-se remodelar a atualização LMS nas Equações 6.6 e 6.7 para atualizar apenas o parâmetro α como

$$\hat{\alpha}[k+1] = \hat{\alpha}[k] + \mu \frac{dJ[k]}{d\alpha}, \quad (6.8)$$

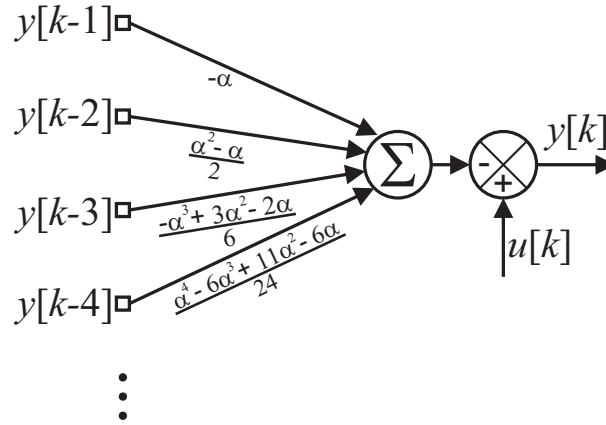
em que

$$\frac{dJ}{d\alpha} = \frac{\partial J}{\partial e} \frac{\partial e}{\partial y} \frac{dy}{d\mathbf{W}} \frac{d\mathbf{W}}{d\alpha}, \quad (6.9)$$

e

$$\frac{\partial J}{\partial e} = e, \quad \frac{\partial e}{\partial y} = -1, \quad \frac{\partial y}{\partial \mathbf{W}} = -\mathbf{Y}. \quad (6.10)$$

Figura 35 – Diagrama de blocos de Equação 6.3 em função de α .



Fonte: elaborada pelo autor.

A derivada $\frac{d\mathbf{W}}{d\alpha}$ pode ser obtida a partir da Equação 2.24, como

$$\frac{dw_i}{d\alpha} = \frac{d}{d\alpha} \left[\left(1 - \frac{\alpha+1}{i} \right) w_{i-1} \right], \quad w_0 := 1$$

$$\frac{dw_i}{d\alpha} = \left(0 - \frac{1}{i} \right) w_{i-1} + \left(1 - \frac{\alpha+1}{i} \right) \frac{d}{d\alpha} w_{i-1}$$

$$\frac{dw_i}{d\alpha} = \frac{-w_{i-1}}{i} + \left(1 - \frac{\alpha+1}{i} \right) \frac{d}{d\alpha} w_{i-1}, \quad \frac{d}{d\alpha} w_0 := 0. \quad (6.11)$$

Aplicando a Equação 6.10 e Equação 6.11 na Equação 6.8 e na Equação 6.9, chega-se

a

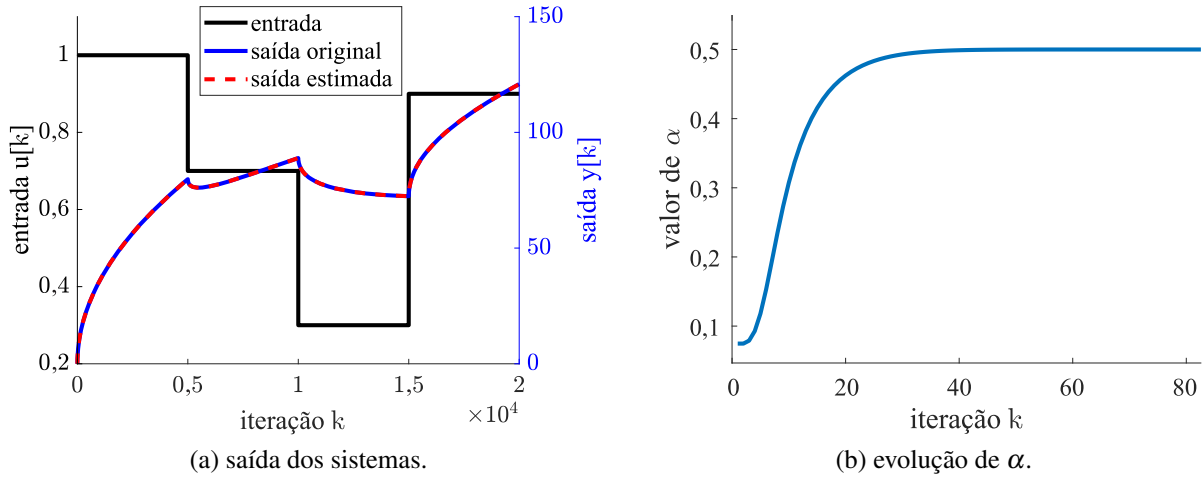
$$\hat{\alpha}[k+1] = \hat{\alpha}[k] + \mu[k]e[k]\mathbf{Y}^T[k] \frac{d\mathbf{W}[k]}{d\alpha}, \quad (6.12)$$

em que

$$\frac{d\mathbf{W}[k]}{d\alpha} = \begin{bmatrix} \frac{d}{d\alpha} \alpha_{w_1} \\ \frac{d}{d\alpha} \alpha_{w_2} \\ \vdots \\ \frac{d}{d\alpha} \alpha_{w_{k-1}} \end{bmatrix}. \quad (6.13)$$

Como evidenciado na Capítulo 5 pela Figura 30 e Figura 8, a melhor escolha para o valor de L é utilizar todas as amostras disponíveis.

Figura 36 – Comparação entre dados de identificação e modelo estimado do caso 1.



Fonte: elaborada pelo autor.

Tabela 12 – Resultados da identificação do caso 1.

parâmetro	inicial	final	real	RMSE	VAF
α	0,074679	0,499999	0,5	$1,516194 \cdot 10^{-6}$	99,999999%

Fonte: elaborada pelo autor.

6.2.1 Validação do algoritmo proposto para o caso 1

Para avaliar a ideia proposta, o algoritmo LMS na [Equação 6.12](#) é submetido a um conjunto de dados discretos de uma integral fracionária de ordem 0,5, ou seja, $G(s) = \frac{1}{s^{0,5}}$. A entrada $u[k]$ utilizada é um sinal PRAS (*pseudo random analog signal*). O valor de saída $y[k]$ é calculado conforme a discretização na [Equação 6.3](#) para $T = 1$. O valor de α é iniciado entre 0 e 1. A taxa de aprendizado μ possui decaimento linear conforme

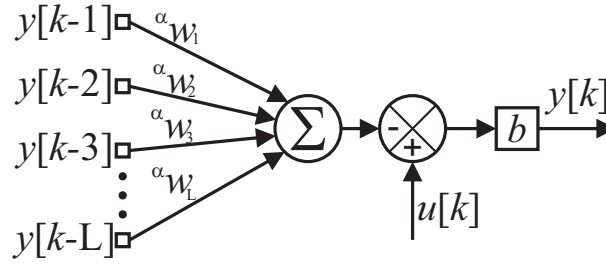
$$\mu[k] = \mu_{inicial} \left(1 - \frac{k}{\tau} \right), \quad (6.14)$$

em que $\mu_{inicial}$ é a taxa de aprendizagem inicial e τ é o número total de iterações. É utilizado $\mu_{inicial} = 0,01$ para este caso. Os resultados da identificação são apresentados na [Figura 36](#) e na [Tabela 12](#), em que são calculados os valores de RMSE (vide [Equação 5.16](#)) e do índice VAF (*variance accounted for*), definido como

$$\text{VAF} = 100\% \cdot \max \left(1 - \frac{\text{var}(y[t] - \hat{y}[t])}{\text{var}(y[t])}, 0 \right). \quad (6.15)$$

O algoritmo foi executado em 0,037465 s no computador cujas características são apresentadas no [Apêndice A](#). Observa-se que com poucas iterações, o parâmetro α converge para o valor correto. É interessante notar a dinâmica na evolução de α , suave e monotonicamente crescente, em direção ao valor correto do parâmetro, de modo que o modelo resultante obtém

Figura 37 – Diagrama de blocos da Equação 6.17.



Fonte: elaborada pelo autor.

VAF de 99,999%. Este resultado satisfatório evidencia a capacidade do algoritmo LMS de identificar sistemas de ordem fracionária. A seguir, aumenta-se a complexidade do modelo e desenvolve-se o algoritmo LMS do mesmo.

6.3 Caso 2: planta $G(s) = \frac{1}{s^\alpha + a}$

Para a planta $G(s) = \frac{1}{s^\alpha + a}$, pode-se desenvolver o algoritmo de identificação de forma análoga ao que foi desenvolvido na Seção 6.2, de forma que

$$y[k] = \left(\frac{1}{a+1} \right) (u[k] - \mathbf{W}^T[k] \mathbf{Y}[k]) \quad (6.16)$$

$$y[k] = b (u[k] - \mathbf{W}^T[k] \mathbf{Y}[k]) . \quad (6.17)$$

O diagrama de blocos da discretização presente na Equação 6.17 é ilustrado na Figura 37. De modo análogo ao desenvolvimento do caso anterior, as equações LMS para o problema são definidas como

$$\begin{cases} \hat{\alpha}[k+1] = \hat{\alpha}[k] + \mu_\alpha \frac{dJ[k]}{d\alpha} \\ \hat{b}[k+1] = \hat{b}[k] + \mu_b \frac{dJ[k]}{db} \end{cases} \quad (6.18)$$

em que

$$\begin{cases} \frac{dJ}{d\alpha} = \frac{\partial J}{\partial e} \frac{\partial e}{\partial y} \frac{\partial y}{\partial \mathbf{W}} \frac{d\mathbf{W}}{d\alpha} \\ \frac{dJ}{db} = \frac{\partial J}{\partial e} \frac{\partial e}{\partial y} \frac{\partial y}{\partial b} \end{cases} . \quad (6.19)$$

Aplicando

$$\frac{\partial J}{\partial e} = e, \quad \frac{\partial e}{\partial y} = -1, \quad \frac{\partial y}{\partial \mathbf{W}} = -b\mathbf{Y}, \quad \frac{\partial y}{\partial b} = u - \mathbf{W}^T \mathbf{Y} \quad (6.20)$$

nas Equações 6.18 e 6.19, por fim,

$$\begin{cases} \hat{\alpha}[k+1] = \hat{\alpha}[k] + \mu_{\alpha} b[k] e[k] \mathbf{Y}^T[k] \frac{d\mathbf{W}[k]}{d\hat{\alpha}} \\ \hat{b}[k+1] = \hat{b}[k] + \mu_b e[k] (\mathbf{W}^T[k] \mathbf{Y}[k] - u[k]) \end{cases}, \quad (6.21)$$

em que

$$\hat{a}[k+1] = \frac{1}{\hat{b}[k+1]} - 1. \quad (6.22)$$

De forma vetorial, pode-se escrever

$$\hat{\Theta}[k+1] = \hat{\Theta}[k] + e[k] \boldsymbol{\mu}[k] \circ \mathbf{X}[k] \quad (6.23)$$

em que

$$\hat{\Theta}[k] = \begin{bmatrix} \hat{\alpha}[k] \\ \hat{b}[k] \end{bmatrix}, \quad \boldsymbol{\mu}[k] = \begin{bmatrix} \mu_{\alpha}[k] \\ \mu_b[k] \end{bmatrix}, \quad \mathbf{X}[k] = \begin{bmatrix} b[k] \mathbf{Y}^T[k] \frac{d\mathbf{W}[k]}{d\hat{\alpha}} \\ \mathbf{W}^T[k] \mathbf{Y}[k] - u[k] \end{bmatrix}, \quad (6.24)$$

e \circ denota o produto de Hadamard. Ou ainda

$$\hat{\Theta}[k+1] = \hat{\Theta}[k] + e[k] \mathbf{X}[k] \boldsymbol{\mu}[k] \quad (6.25)$$

em que

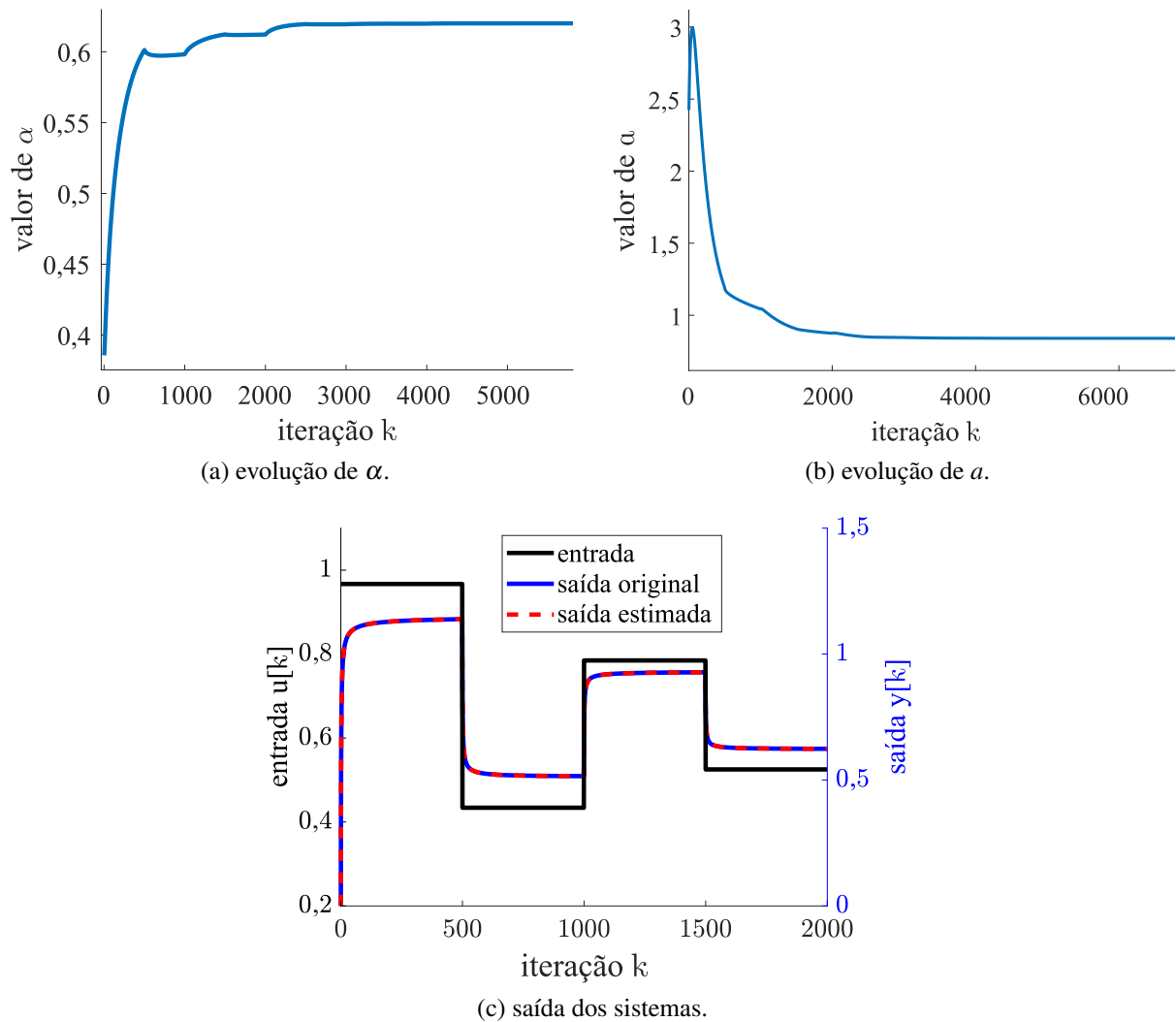
$$\hat{\Theta}[k] = \begin{bmatrix} \hat{\alpha}[k] \\ \hat{b}[k] \end{bmatrix}, \quad \boldsymbol{\mu}[k] = \begin{bmatrix} \mu_{\alpha}[k] \\ \mu_b[k] \end{bmatrix}, \quad \mathbf{X}[k] = \begin{bmatrix} b[k] \mathbf{Y}^T[k] \frac{d\mathbf{W}[k]}{d\hat{\alpha}} & 0 \\ 0 & \mathbf{W}^T[k] \mathbf{Y}[k] - u[k] \end{bmatrix}. \quad (6.26)$$

6.3.1 Validando o algoritmo proposto para o caso 2

Para avaliar a ideia proposta, o algoritmo LMS desenvolvido para o caso 2 é submetido a um conjunto de dados discretos do sistema $G(s) = \frac{1}{s^{0,62} + 0,84}$. A entrada $u[k]$ é um sinal PRAS. O valor de saída $y[k]$ é calculado conforme a discretização na Equação 6.17 para $T = 1$. Os parâmetros α e b são iniciados com valores entre 0 e 1. A taxa de aprendizado μ possui decaimento linear conforme a Equação 6.14 com $\mu_{inicial} = 0,01$ para este problema. O processo de treinamento foi realizado ao longo de 300 épocas (ciclos completos de dados). Os resultados da identificação são apresentados na Figura 38 e na Tabela 13. O algoritmo foi executado em 7,879 s no computador cujas características são apresentadas no Apêndice A.

Observa-se que foram necessários mais iterações para que os parâmetros convergissem para os valores corretos, evidenciando uma maior complexidade do sistema em questão.

Figura 38 – Comparação entre dados de identificação e modelo estimado do caso 2.



Fonte: elaborada pelo autor.

Tabela 13 – Resultados da identificação do caso 2.

parâmetro	inicial	final	real	RMSE	VAF
α	0,385728	0,620000	0,62	$1,455109 \cdot 10^{-8}$	99,999999%
a	2,424050	0,839999	0,84		

Fonte: elaborada pelo autor.

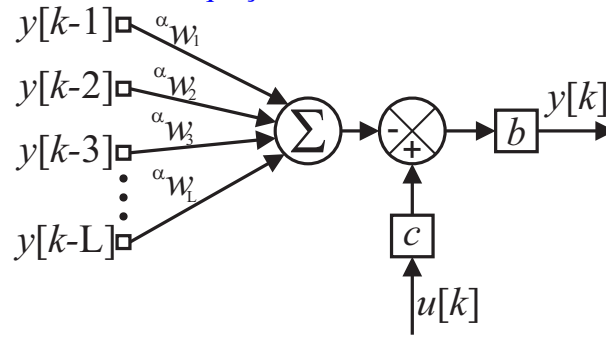
6.4 Caso 3: planta $G(s) = \frac{c}{s^\alpha + a}$

Aumentando ainda mais a complexidade, a seguir é desenvolvido o algoritmo LMS para a planta $G(s) = \frac{c}{s^\alpha + a}$ de forma análoga aos casos 1 e 2, de forma que

$$y[k] = \left(\frac{1}{a+1} \right) (cu[k] - \mathbf{W}^T[k] \mathbf{Y}[k]) \quad (6.27)$$

$$y[k] = b (cu[k] - \mathbf{W}^T[k] \mathbf{Y}[k]) . \quad (6.28)$$

Figura 39 – Diagrama de blocos da Equação 6.28.



Fonte: elaborada pelo autor.

O diagrama de blocos da discretização presente na Equação 6.28 é ilustrado na Figura 39.

As equações LMS para o problema são definidas como

$$\begin{cases} \hat{\alpha}[k+1] = \hat{\alpha}[k] + \mu_{\alpha} \frac{dJ[k]}{d\alpha} \\ \hat{b}[k+1] = \hat{b}[k] + \mu_b \frac{dJ[k]}{db}, \\ \hat{c}[k+1] = \hat{c}[k] + \mu_c \frac{dJ[k]}{dc} \end{cases} \quad (6.29)$$

em que

$$\begin{cases} \frac{dJ}{d\alpha} = \frac{\partial J}{\partial e} \frac{\partial e}{\partial y} \frac{\partial y}{\partial \mathbf{W}} \frac{d\mathbf{W}}{d\alpha} \\ \frac{dJ}{db} = \frac{\partial J}{\partial e} \frac{\partial e}{\partial y} \frac{\partial y}{\partial b} \\ \frac{dJ}{dc} = \frac{\partial J}{\partial e} \frac{\partial e}{\partial y} \frac{\partial y}{\partial c} \end{cases} \quad (6.30)$$

Aplicando

$$\frac{\partial J}{\partial e} = e, \quad \frac{\partial e}{\partial y} = -1, \quad \frac{\partial y}{\partial \mathbf{W}} = -b\mathbf{Y}, \quad \frac{\partial y}{\partial b} = cu - \mathbf{W}^T \mathbf{Y}, \quad \frac{\partial y}{\partial c} = bu \quad (6.31)$$

nas Equações 6.29 e 6.30, por fim,

$$\begin{cases} \hat{\alpha}[k+1] = \hat{\alpha}[k] + \mu_{\alpha} b[k] e[k] \mathbf{Y}^T[k] \frac{d\mathbf{W}[k]}{d\hat{\alpha}} \\ \hat{b}[k+1] = \hat{b}[k] + \mu_b e[k] (\mathbf{W}^T[k] \mathbf{Y}[k] - cu[k]) \\ \hat{c}[k+1] = \hat{c}[k] - \mu_c e[k] b[k] u[k] \end{cases}, \quad (6.32)$$

em que

$$\hat{a}[k+1] = \frac{1}{\hat{b}[k+1]} - 1. \quad (6.33)$$

De forma vetorial, pode-se escrever

$$\hat{\Theta}[k+1] = \hat{\Theta}[k] + e[k] \boldsymbol{\mu}[k] \circ \mathbf{X}[k], \quad (6.34)$$

em que

$$\hat{\Theta}[k] = \begin{bmatrix} \hat{\alpha}[k] \\ \hat{b}[k] \\ \hat{c}[k] \end{bmatrix}, \quad \boldsymbol{\mu}[k] = \begin{bmatrix} \mu_{\alpha}[k] \\ \mu_b[k] \\ \mu_c[k] \end{bmatrix}, \quad \mathbf{X}[k] = \begin{bmatrix} b[k] \mathbf{Y}^T[k] \frac{d\mathbf{W}[k]}{d\hat{\alpha}} \\ \mathbf{W}^T[k] \mathbf{Y}[k] - cu[k] \\ -b[k] u[k] \end{bmatrix}, \quad (6.35)$$

e \circ denota o produto de Hadamard. Ou ainda

$$\hat{\Theta}[k+1] = \hat{\Theta}[k] + e[k] \mathbf{X}[k] \boldsymbol{\mu}[k], \quad (6.36)$$

em que

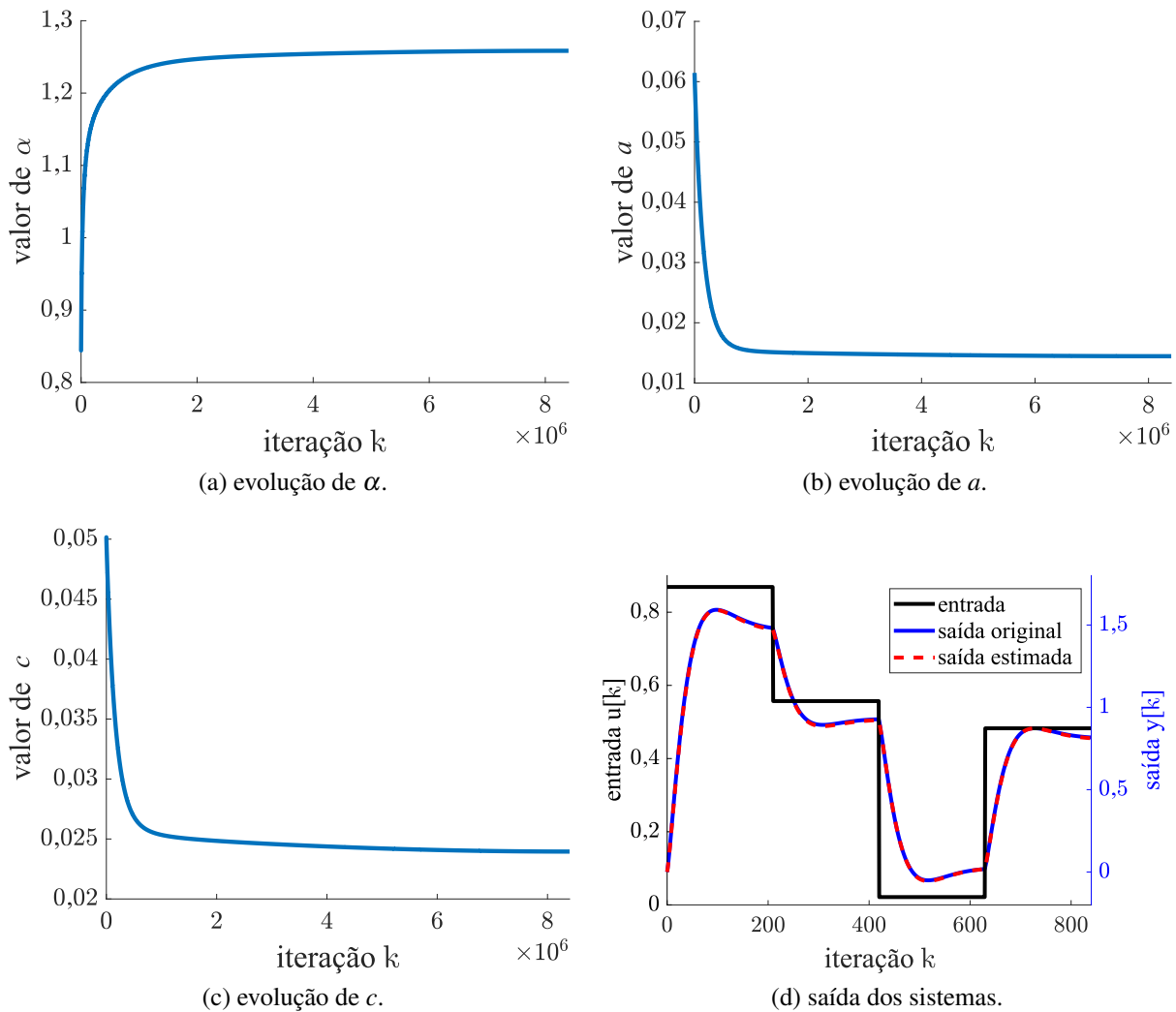
$$\hat{\Theta}[k] = \begin{bmatrix} \hat{\alpha}[k] \\ \hat{b}[k] \\ \hat{c}[k] \end{bmatrix}, \quad \boldsymbol{\mu}[k] = \begin{bmatrix} \mu_{\alpha}[k] \\ \mu_b[k] \\ \mu_c[k] \end{bmatrix}, \quad \mathbf{X}[k] = \begin{bmatrix} b[k] \mathbf{Y}^T[k] \frac{d\mathbf{W}[k]}{d\hat{\alpha}} & 0 & 0 \\ 0 & \mathbf{W}^T[k] \mathbf{Y}[k] - cu[k] & 0 \\ 0 & 0 & -b[k] u[k] \end{bmatrix}. \quad (6.37)$$

6.4.1 Validando o algoritmo proposto para o caso 3

Para avaliar a ideia proposta, o algoritmo LMS desenvolvido para o caso 3 é submetido a um conjunto de dados discretos do sistema de ordem fracionária apresentado em Petrás (2011), cujo modelo representa uma planta de aquecimento elétrico dada por

$$G(s) = \frac{1}{39,96s^{1,25} + 0,598} = \frac{0,025}{s^{1,25} + 0,015}. \quad (6.38)$$

Figura 40 – Comparação entre dados de identificação e modelo estimado do caso 3.



Fonte: elaborada pelo autor.

Tabela 14 – Resultados da identificação do caso 3.

parâmetro	inicial	final	real	RMSE	VAF
α	0,844125	1,25844	1,25	$5,019832 \cdot 10^{-3}$	99,997414%
a	0,061452	0,0144542	0,015		
c	0,050157	0,0239523	0,025		

Fonte: elaborada pelo autor.

A entrada $u[k]$ é um sinal PRAS. O valor de saída $y[k]$ é calculado conforme a discretização na Equação 6.28. O valor de α é iniciado entre 0,5 e 1, enquanto que os valores de a e c são iniciados entre 0,01 e 0,1. A taxa de aprendizado μ possui decaimento linear conforme a Equação 6.14 com $\mu_{inicial} = \begin{bmatrix} 10^{-2} & 10^{-5} & 10^{-5} \end{bmatrix}^T$. O processo de treinamento foi realizado ao longo de 10.000 épocas. Os resultados da identificação são apresentados na Figura 40 e na Tabela 14. O algoritmo foi executado em 268,76 s no computador cujas características são apresentadas no Apêndice A.

Observou-se a necessidade de uma quantidade ainda maior de iterações para que os parâmetros convergissem para os valores corretos, o que mostra uma maior complexidade para o sistema em questão. O resultado de $\text{VAF} = 99,997\%$ é considerável aceitável e demonstra que o algoritmo proposto é capaz de identificar modelos do tipo $\frac{c}{s^\alpha + a}$, como a planta de aquecimento elétrico de [Petráš \(2011\)](#).

A seguir, é realizado o desenvolvimento do algoritmo LMS de identificação de uma planta de ordem fracionária de uso geral.

6.5 Caso 4: planta generalizada $\frac{fs^\gamma + \dots + gs^\delta + c}{s^\alpha + \dots + ds^\beta + a}$

O desenvolvimento de uma expressão para estimação via LMS de um caso geral é descrito a seguir. Aplicando a multiplicação cruzada na função de transferência de uso geral,

$$(fs^\gamma + \dots + gs^\delta + c)U(s) = (s^\alpha + \dots + ds^\beta + a)Y(s) \quad (6.39)$$

$$fs^\gamma U(s) + \dots + gs^\delta U(s) + cU(s) = s^\alpha Y(s) + \dots + ds^\beta Y(s) + aY(s). \quad (6.40)$$

Aplicando a forma compacta da definição de Grünwald-Letnikov de derivada fracionária desenvolvida na [Subseção 2.1.1](#) (vide [Equação 2.18](#)), e posteriormente a transformada para tempo discreto, obtêm-se

$$f \sum_{i=0}^L \gamma w_i u[k-i] + \dots + g \sum_{i=0}^L \delta w_i u[k-i] + cu[k] = \sum_{i=0}^L \alpha w_i y[k-i] + \dots + d \sum_{i=0}^L \beta w_i y[k-i] + ay[k]. \quad (6.41)$$

Extraindo o primeiro termo de cada somatório, tem-se

$$\begin{aligned} f \cancel{\gamma w_0 u[k]} + f \sum_{i=1}^L \gamma w_i u[k-i] + \dots + g \cancel{\delta w_0 u[k]} + g \sum_{i=1}^L \delta w_i u[k-i] + cu[k] &= \cancel{\alpha w_0 y[k]} + \\ &+ \sum_{i=1}^L \alpha w_i y[k-i] + \dots + d \cancel{\beta w_0 y[k]} + d \sum_{i=1}^L \beta w_i y[k-i] + ay[k] \end{aligned} \quad (6.42)$$

$$\begin{aligned} (f + \dots + g + c)u[k] + f \sum_{i=1}^L \gamma w_i u[k-i] + \dots + g \sum_{i=1}^L \delta w_i u[k-i] &= (1 + \dots + d + a)y[k] + \\ &+ \sum_{i=1}^L \alpha w_i y[k-i] + \dots + d \sum_{i=1}^L \beta w_i y[k-i] \end{aligned} \quad (6.43)$$

Isolando $y[k]$, chega-se a

$$y[k] = \left(\frac{1}{1 + \dots + d + a} \right) \left\{ (f + \dots + g + c)u[k] + f \sum_{i=1}^L \gamma_{w_i} u[k-i] + \dots + g \sum_{i=1}^L \delta_{w_i} u[k-i] + \right. \\ \left. - \sum_{i=1}^L \alpha_{w_i} y[k-i] - \dots - d \sum_{i=1}^L \beta_{w_i} y[k-i] \right\} \quad (6.44)$$

Para facilitar a manipulação algébrica, a Equação 6.44 é reescrita como

$$y[k] = b \left\{ (f + \dots + g + c)u[k] + f \mathbf{W}^T[k] \mathbf{U}[k] + \dots + g \delta \mathbf{W}^T[k] \mathbf{U}[k] - \alpha \mathbf{W}^T[k] \mathbf{Y}[k] - \dots - d \beta \mathbf{W}^T[k] \mathbf{Y}[k] \right\} \quad (6.45)$$

$$y[k] = b \left\{ (f + \dots + g + c)u[k] + (f \mathbf{W}^T[k] + \dots + g \delta \mathbf{W}^T[k]) \mathbf{U}[k] - (\alpha \mathbf{W}^T[k] + \dots + d \beta \mathbf{W}^T[k]) \mathbf{Y}[k] \right\}, \quad (6.46)$$

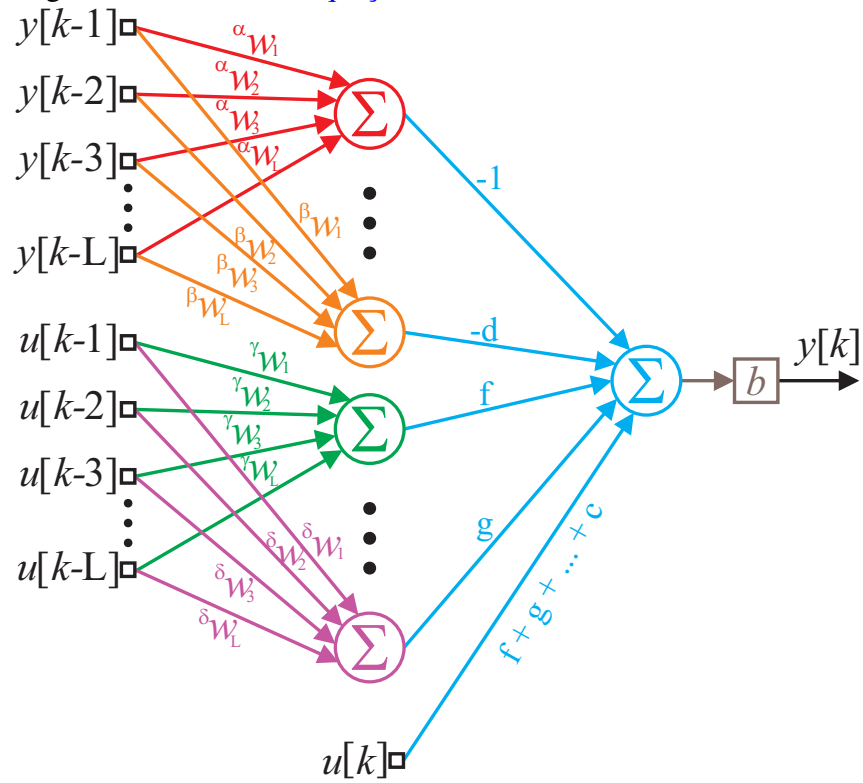
em que $\alpha \mathbf{W}[k]$, \dots , $\delta \mathbf{W}[k]$ e $\mathbf{U}[k]$ são definidos como

$$\alpha \mathbf{W}[k] = \begin{bmatrix} \alpha_{w_1} \\ \alpha_{w_2} \\ \vdots \\ \alpha_{w_{k-1}} \end{bmatrix}, \dots, \delta \mathbf{W}[k] = \begin{bmatrix} \delta_{w_1} \\ \delta_{w_2} \\ \vdots \\ \delta_{w_{k-1}} \end{bmatrix}, \quad \mathbf{U}[k] = \begin{bmatrix} u[k-1] \\ u[k-2] \\ \vdots \\ u[1] \end{bmatrix}, \quad k = 2, 3, \dots, L.$$

O diagrama de blocos da Equação 6.46 é ilustrado na Figura 41. Assim, as equações LMS para cada parâmetro do problema são definidas como

$$\left\{ \begin{array}{l} \hat{\alpha}[k+1] = \hat{\alpha}[k] + \mu_{\alpha} \frac{dJ[k]}{d\alpha} \\ \vdots \\ \hat{\beta}[k+1] = \hat{\beta}[k] + \mu_{\beta} \frac{dJ[k]}{d\beta} \\ \hat{\gamma}[k+1] = \hat{\gamma}[k] + \mu_{\gamma} \frac{dJ[k]}{d\gamma} \\ \vdots \\ \hat{\delta}[k+1] = \hat{\delta}[k] + \mu_{\delta} \frac{dJ[k]}{d\delta} \end{array} \right\}, \quad \left\{ \begin{array}{l} \hat{b}[k+1] = \hat{b}[k] + \mu_b \frac{dJ[k]}{db} \\ \vdots \\ \hat{d}[k+1] = \hat{d}[k] + \mu_d \frac{dJ[k]}{dd} \\ \hat{c}[k+1] = \hat{c}[k] + \mu_c \frac{dJ[k]}{dc} \\ \hat{f}[k+1] = \hat{f}[k] + \mu_f \frac{dJ[k]}{df} \\ \vdots \\ \hat{g}[k+1] = \hat{g}[k] + \mu_g \frac{dJ[k]}{dg} \end{array} \right\}, \quad (6.47)$$

Figura 41 – Diagrama de blocos da Equação 6.46.



Fonte: elaborada pelo autor.

em que

$$\left\{ \begin{array}{l} \frac{dJ}{d\alpha} = \frac{\partial J}{\partial e} \frac{\partial e}{\partial y} \frac{\partial y}{\partial \alpha \mathbf{W}} \frac{d\alpha \mathbf{W}}{d\alpha} \\ \vdots \\ \frac{dJ}{d\beta} = \frac{\partial J}{\partial e} \frac{\partial e}{\partial y} \frac{\partial y}{\partial \beta \mathbf{W}} \frac{d\beta \mathbf{W}}{d\beta} \\ \vdots \\ \frac{dJ}{d\gamma} = \frac{\partial J}{\partial e} \frac{\partial e}{\partial y} \frac{\partial y}{\partial \gamma \mathbf{W}} \frac{d\gamma \mathbf{W}}{d\gamma} \\ \vdots \\ \frac{dJ}{d\delta} = \frac{\partial J}{\partial e} \frac{\partial e}{\partial y} \frac{\partial y}{\partial \delta \mathbf{W}} \frac{d\delta \mathbf{W}}{d\delta} \end{array} \right\}, \quad \left\{ \begin{array}{l} \frac{dJ}{db} = \frac{\partial J}{\partial e} \frac{\partial e}{\partial y} \frac{\partial y}{\partial b} \\ \vdots \\ \frac{dJ}{dd} = \frac{\partial J}{\partial e} \frac{\partial e}{\partial y} \frac{\partial y}{\partial d} \\ \vdots \\ \frac{dJ}{dc} = \frac{\partial J}{\partial e} \frac{\partial e}{\partial y} \frac{\partial y}{\partial c} \\ \vdots \\ \frac{dJ}{df} = \frac{\partial J}{\partial e} \frac{\partial e}{\partial y} \frac{\partial y}{\partial f} \\ \vdots \\ \frac{dJ}{dg} = \frac{\partial J}{\partial e} \frac{\partial e}{\partial y} \frac{\partial y}{\partial g} \end{array} \right. \quad (6.48)$$

Aplicando

$$\begin{aligned}
 \frac{\partial J}{\partial e} &= e, & \frac{\partial e}{\partial y} &= -1 \\
 \frac{\partial y}{\partial \alpha \mathbf{W}} &= -b\mathbf{Y} \quad \dots \quad \frac{\partial y}{\partial \beta \mathbf{W}} = -bd\mathbf{Y}, & \frac{\partial y}{\partial \gamma \mathbf{W}} &= bf\mathbf{U} \quad \dots \quad \frac{\partial y}{\partial \delta \mathbf{W}} = bg\mathbf{U} \\
 \frac{\partial y}{\partial b} &= (f + \dots + g + c)u + (f\gamma \mathbf{W}^T + \dots + g\delta \mathbf{W}^T)\mathbf{U} - (\alpha \mathbf{W}^T + \dots + d\beta \mathbf{W}^T)\mathbf{Y} \\
 &\vdots \\
 \frac{\partial y}{\partial d} &= -b\beta \mathbf{W}^T \mathbf{Y} \\
 \frac{\partial y}{\partial c} &= bu \\
 \frac{\partial y}{\partial f} &= b(u + \gamma \mathbf{W}^T \mathbf{U}) \quad \dots \quad \frac{\partial y}{\partial g} = b(u + \delta \mathbf{W}^T \mathbf{U})
 \end{aligned} \tag{6.49}$$

nas Equações 6.47 e 6.48, chega-se por fim às seguintes equações

$$\left\{ \begin{aligned}
 \hat{\alpha}[k+1] &= \hat{\alpha}[k] + \mu_{\alpha} e[k] b[k] \mathbf{Y}^T[k] \frac{d\alpha \mathbf{W}[k]}{d\hat{\alpha}} \\
 &\vdots \\
 \hat{\beta}[k+1] &= \hat{\beta}[k] + \mu_{\beta} e[k] b[k] d[k] \mathbf{Y}^T[k] \frac{d\beta \mathbf{W}[k]}{d\hat{\beta}} \\
 \hat{\gamma}[k+1] &= \hat{\gamma}[k] - \mu_{\gamma} e[k] b[k] f[k] \mathbf{U}^T[k] \frac{d\gamma \mathbf{W}[k]}{d\hat{\gamma}} \\
 &\vdots \\
 \hat{\delta}[k+1] &= \hat{\delta}[k] - \mu_{\delta} e[k] b[k] g[k] \mathbf{U}^T[k] \frac{d\delta \mathbf{W}[k]}{d\hat{\delta}} \\
 \hat{b}[k+1] &= \hat{b}[k] + \mu_b e[k] \left\{ (\alpha \mathbf{W}^T[k] + \dots + d\beta \mathbf{W}^T[k]) \mathbf{Y}[k] - (f\gamma \mathbf{W}^T[k] + \dots + g\delta \mathbf{W}^T[k]) \mathbf{U}[k] - (f + \dots + g + c)u[k] \right\}, \\
 &\vdots \\
 \hat{d}[k+1] &= \hat{d}[k] + \mu_d e[k] b[k] \beta \mathbf{W}^T[k] \mathbf{Y}[k] \\
 \hat{c}[k+1] &= \hat{c}[k] - \mu_c e[k] b[k] u[k] \\
 \hat{f}[k+1] &= \hat{f}[k] - \mu_f e[k] b[k] (u[k] + \gamma \mathbf{W}^T[k] \mathbf{U}[k]) \\
 &\vdots \\
 \hat{g}[k+1] &= \hat{g}[k] - \mu_g e[k] b[k] (u[k] + \delta \mathbf{W}^T[k] \mathbf{U}[k])
 \end{aligned} \right.$$

em que

$$a[k+1] = \frac{1 - (1 + \dots + d[k+1])b[k+1]}{b[k+1]}. \tag{6.50}$$

De forma vetorial,

$$\hat{\Theta}[k+1] = \hat{\Theta}[k] + e[k]\mu[k] \circ \mathbf{X}[k], \quad (6.51)$$

em que

$$\hat{\Theta}[k] = \begin{bmatrix} \hat{\alpha}[k] \\ \vdots \\ \hat{\beta}[k] \\ \hat{\gamma}[k] \\ \vdots \\ \hat{\delta}[k] \\ \hat{b}[k] \\ \vdots \\ \hat{d}[k] \\ \hat{c}[k] \\ \hat{f}[k] \\ \vdots \\ \hat{g}[k] \end{bmatrix}, \quad \mu = \begin{bmatrix} \mu_{\alpha} \\ \vdots \\ \mu_{\beta} \\ \mu_{\gamma} \\ \vdots \\ \mu_{\delta} \\ \mu_b \\ \vdots \\ \mu_d \\ \mu_c \\ \mu_f \\ \vdots \\ \mu_g \end{bmatrix}, \quad \mathbf{X}[k] = \begin{bmatrix} b[k]\mathbf{Y}^T[k] \frac{d^{\alpha}\mathbf{W}[k]}{d\hat{\alpha}} \\ \vdots \\ b[k]d[k]\mathbf{Y}^T[k] \frac{d^{\beta}\mathbf{W}[k]}{d\hat{\beta}} \\ -b[k]f[k]\mathbf{U}^T[k] \frac{d^{\gamma}\mathbf{W}[k]}{d\hat{\gamma}} \\ \vdots \\ -b[k]g[k]\mathbf{U}^T[k] \frac{d^{\delta}\mathbf{W}[k]}{d\hat{\delta}} \\ (\alpha\mathbf{W}^T[k] + \dots + d^{\beta}\mathbf{W}^T[k])\mathbf{Y}[k] - (f^{\gamma}\mathbf{W}^T[k] + \dots + g^{\delta}\mathbf{W}^T[k])\mathbf{U}[k] - (f + \dots + g + c)u[k] \\ \vdots \\ b[k]^{\beta}\mathbf{W}^T\mathbf{Y} \\ -b[k]u[k] \\ -b[k](u[k] + \gamma\mathbf{W}^T\mathbf{U}) \\ \vdots \\ -b[k](u[k] + \delta\mathbf{W}^T\mathbf{U}) \end{bmatrix}, \quad (6.52)$$

e \circ denota o produto de Hadamard.

6.6 Condições de convergência do algoritmo

A convergência de algoritmos baseados no LMS é determinada por diferentes fatores, sendo a taxa de aprendizagem μ um dos mais relevantes. De maneira geral, deve-se escolher uma taxa de aprendizagem tal que

$$0 < \mu < \frac{2}{\lambda_{\max}(\mathbf{R})}, \quad (6.53)$$

em que λ_{\max} é o maior autovalor da matriz de autocorrelação \mathbf{R} do algoritmo LMS. Embora o algoritmo proposto utilize todas as amostras passadas disponíveis na iteração k , o que poderia implicar em um problema de condicionamento da matriz \mathbf{R} , no algoritmo proposto, cada

parâmetro tem sua própria equação LMS, e sua própria taxa de aprendizagem, de forma que a matriz de regressão é um escalar. Assim, deve-se escolher uma taxa de aprendizagem tal que

$$0 < \mu < \frac{2}{\sigma_x^2}, \quad (6.54)$$

em que σ_x^2 é a variância do regressor. Apesar da separação das equações do LMS eliminar o problema do condicionamento matricial, esta separação também introduz alguns desafios, como a solução para a convergência assíncrona dos parâmetros, o que implica na escolha adequada da taxa μ de cada equação LMS. Este efeito é mitigado ao se adotar uma taxa decrescente no tempo, como a apresentada na [Equação 6.14](#).

É importante mencionar que, embora cada parâmetro tenha sua própria equação LMS, as equações LMS não são desacopladas, visto que os parâmetros aparecem nos regressores de equações LMS de outros parâmetros, como visto claramente na [Equação 6.37](#) e na [Equação 6.52](#). A escolha não trivial adequada dos valores de μ para solução dos desafios mencionados pode ser sanada com a aplicação do algoritmo NLMS (*normalized least mean squares*), uma variação do algoritmo LMS tradicional que normaliza o passo de aprendizagem μ com base na norma quadrática do vetor de regressores.

6.7 Considerações finais sobre o algoritmo proposto

Dada a devida atenção aos pontos levantados anteriormente sobre as condições de convergência, ressalta-se que o algoritmo proposto oferece diversas vantagens, como a simplicidade matemática. As atualizações recursivas são realizadas exclusivamente por meio de operações aritméticas simples. Embora alguns termos envolvam derivadas de parâmetros, como $\frac{d^\alpha \mathbf{W}[k]}{d\alpha}$, cabe ressaltar que essas derivadas são obtidas recursivamente, conforme apresentado na [Equação 6.11](#), que também é composta por operações aritméticas simples.

Uma característica interessante e pouco convencional do algoritmo proposto é que, ao contrário do algoritmo LMS tradicionalmente utilizado em sistemas de ordem inteira, os parâmetros obtidos por meio deste algoritmo referem-se a um modelo de ordem contínua, no domínio s de Laplace. Esta característica é essencial para que se possa aplicar análises de estabilidade como a proposta no [Capítulo 3](#). Por fim, recomenda-se o uso de entradas com espectro plano ou próximo disso, como o sinal PRAS, a fim de garantir uma excitação adequada das dinâmicas do sistema.

7 SÍNTESE DE CONTROLADORES DE ORDEM FRACIONÁRIA

Neste capítulo, são apresentadas duas propostas que integram a síntese de controladores de ordem fracionária: a sintonia de parâmetros do controlador fracionário na [Seção 7.1](#), e a sua implementação embarcada na [Seção 7.2](#). Desta forma, este capítulo é dividido da seguinte forma. Na [Subseção 7.1.1](#), são apresentados os fundamentos do algoritmo de otimização metaheurística PSO (*particle swarm optimization*), utilizado em problemas de otimização. Na [Subseção 7.1.2](#) é proposto o algoritmo PSO-LT, que introduz os conceitos desenvolvidos no [Capítulo 4](#) sobre estabilidade a partir da função Lambert-Tsallis com o intuito de restringir a busca por parâmetros ótimos a um espaço de soluções com garantia de estabilidade do sistema de controle. Uma técnica da literatura para aproximação discreta de controladores $PI^\lambda D^\mu$ é descrita na [Subseção 7.2.1](#), enquanto a proposta de regressão não linear visando uma implementação embarcada eficiente desse tipo de controlador é apresentada na [Subseção 7.2.2](#).

7.1 Sintonia de parâmetros do controlador de ordem fracionária

Entre os métodos de sintonia de controladores de ordem fracionária, destacam-se os métodos de otimização meta-heurística, como a aplicação de algoritmos genéticos ([SARIR et al., 2021](#)), evolução diferencial ([GAO et al., 2013](#)), colônia de abelhas ([DJABALLAH et al., 2024](#)) e otimização por enxame de partículas ([NASSEF et al., 2023](#)).

A metaheurística PSO ([KENNEDY; EBERHART, 1995](#)) tornou-se um dos principais métodos de inteligência computacional em resolver problemas de otimização global ([SHAMI et al., 2022](#)). Este método provou ser uma poderosa ferramenta na resolução de problemas com não diferenciabilidade e não-convexidade da função objetivo, bem como alta dimensionalidade do vetor-solução. As vantagens do PSO são sua relativa simplicidade e convergência estável com boa eficiência computacional ([WANG et al., 2017](#)). No projeto do controlador de ordem fracionária deste capítulo, o controlador PI^λ é sintonizado via PSO.

7.1.1 Fundamentos do algoritmo PSO

A metaheurística de otimização PSO é um algoritmo baseado em população de soluções e sua busca pela solução ótima se dá em múltiplas direções pelas várias partículas, em vez de uma única solução. O algoritmo é inspirado no comportamento social e na auto-organização de grupos de pássaros migratórios e de cardume de peixes ([KENNEDY; EBERHART,](#)

1995). Entre suas vantagens, o algoritmo PSO possui memória que retém as boas soluções alcançadas por todas as partículas. Além disso, o algoritmo tem cooperação entre as partículas do enxame, compartilhando informações (HUNG *et al.*, 2008).

Para a otimização em um espaço de busca D -dimensional, a posição da j -ésima partícula do enxame é definida como $\mathbf{p}_j = [p_{j1}, p_{j1}, \dots, p_{jD}]$. Na iteração k , a melhor posição encontrada pela partícula \mathbf{p}_j é definida como $\mathbf{p}_j^{best}[k]$, enquanto que a melhor posição encontrada pelo enxame como um todo na iteração k é definida como $\mathbf{g}^{best}[k]$. Estas posições são utilizadas na atualização das velocidades \mathbf{v}_j e das posições $\mathbf{p}_j[k]$ de cada partícula, conforme

$$\zeta_1[k] = c_1 \psi_1[k] (\mathbf{p}_j^{best}[k] - \mathbf{p}_j[k]) \quad (7.1)$$

$$\zeta_2[k] = c_2 \psi_2[k] (\mathbf{g}^{best}[k] - \mathbf{p}_j[k]) \quad (7.2)$$

$$\mathbf{v}_j[k+1] = \rho (w[k]\mathbf{v}_j[k] + \zeta_1[k] + \zeta_2[k]) \quad (7.3)$$

$$\mathbf{p}_j[k+1] = \mathbf{p}_j[k] + \mathbf{v}_j[k+1], \quad (7.4)$$

em que $c_1, c_2 > 0$ são coeficientes constantes de aprendizagem cognitiva e social, respectivamente e ψ_1 e ψ_2 são variáveis aleatórias uniformemente distribuídas no intervalo $(0,1)$. O coeficiente $\rho > 0$ é um fator de constrição usado para estabilizar o processo de convergência. Por fim, $w[k]$ é o fator de peso inercial decrescente no instante k e visa evitar problemas relacionados à velocidade das partículas. Neste trabalho, este fator é definido como

$$w[k] = w_{final} + (w_{inicial} - w_{final}) \left(\frac{N_{it} - k}{N_{it} - 1} \right), \quad (7.5)$$

em que $w_{inicial}$ e w_{final} são, respectivamente, o valor inicial e final de w , enquanto que N_{it} é o número de iterações do algoritmo PSO.

Por fim, o desempenho de cada partícula $\mathbf{p}_j[k]$ e as posições $\mathbf{p}_j^{best}[k]$ e $\mathbf{g}^{best}[k]$ são avaliadas por meio de uma função objetivo. A solução ótima obtida depende da função objetivo estabelecida, que é fundamental no projeto. Os índices de desempenho baseados em erro são bastante utilizados nas funções objetivo em otimização de parâmetros de sistemas de controle em malha fechada, como a integral do erro quadrático multiplicado pelo tempo (*integral of time-weighted squared error* - ITSE) utilizada em Setiawan *et al.* (2025). No entanto, esses critérios focam principalmente na avaliação geral da resposta, em vez de enfatizarem ou inibirem certas características da resposta dinâmica do sistema de controle (ZHANG; YANG, 2024).

Neste sentido, uma abordagem alternativa para índice de desempenho é apresentada em Aghababa (2015), que utiliza critérios da resposta transitória (RT) do sistema de controle,

como tempo de subida T_s , tempo de acomodação T_a , máximo sobressinal M_s e erro de estado permanente E_{ep} , ponderados por meio de uma combinação convexa como

$$J_{RT} = \frac{1}{1+e^{-v}} (T_s + T_a) + \frac{e^{-v}}{1+e^{-v}} (M_s + E_{ep}) , \quad (7.6)$$

em que $-\infty < v < \infty$ é um expoente usado nos termos de ponderação. Otimizações em sistemas de controle que utilizam J_{RT} buscam minimizar este índice. Se $v \rightarrow \infty$, as grandezas T_s e T_a são priorizadas. Para $v \rightarrow -\infty$, as grandezas M_s e E_{ep} são priorizadas na resposta dinâmica do sistema.

A função objetivo utilizada na otimização do controlador nesta tese é uma combinação de J_{RT} com um termo de regularização baseado na norma $L1$ dos ganhos do controlador, de forma que

$$J_{total} = J_{RT} + \mu_{reg} \|\mathbf{K}\|_1 , \quad (7.7)$$

em que μ_{reg} é o parâmetro de ponderação associado ao termo de regularização e \mathbf{K} é o vetor de ganhos do controlador, definido, por exemplo, como $\mathbf{K} = [k_p \ k_i]$ para o controlador PI ou PI^λ . A norma $L1$ foi escolhida para favorecer modelos esparsos e a manutenção de ganhos baixos, evitando valores excessivamente altos para os parâmetros.

7.1.2 Sintonia estável de sistemas ordem fracionária via PSO e função Lambert-Tsallis

De forma geral, o hiperespaço de busca de um algoritmo de otimização meta-heurística, como o PSO, é um hiperparalelepípedo formado pelos limites preestabelecidos, também conhecido como restrições do tipo caixa. Um desafio recorrente na aplicação do PSO à sintonia de controladores de ordem fracionária é a definição dos limites de busca que garantam a estabilidade, pois os intervalos viáveis dos parâmetros geralmente não são conhecidos *a priori*, o que exige muita experimentação.

Nesta seção, este problema é atacado propondo a aplicação da função Lambert-Tsallis em conjunto com o algoritmo PSO para definir limites dinâmicos de busca para as partículas, doravante denominado de PSO-LT. Tais limites dinâmicos, calculados iterativamente para cada partícula, delinham a superfície crítica, como ilustrada na [Figura 27](#), transformando o espaço de busca no hipervolume correspondente à região de estabilidade do sistema de controle em questão. Assim, garante-se que o algoritmo PSO explore exclusivamente a região estável do sistema de controle, e que toda e qualquer solução será uma solução válida do ponto de vista da

estabilidade. Como visto na [Figura 27](#), a região, ou volume, estável de um determinado sistema de controle pode ser significativamente menor que a região instável. Permitir que o PSO explore regiões de instabilidade resulta em avaliações desperdiçadas, que poderiam ser direcionadas à exploração de regiões estáveis.

Para os casos em que uma partícula ultrapasse a superfície crítica (ou seja, caia em uma região instável), é aplicada uma saturação dinâmica, de forma que a partícula é imediatamente reposicionada no limite da superfície, garantindo que o algoritmo PSO explore apenas regiões estáveis do sistema de controle. Essa estratégia, além de garantir a estabilidade, acelera a convergência ao restringir o algoritmo PSO apenas a regiões relevantes. Para ilustrar essa abordagem, a seguir é apresentado um exemplo aplicado ao controlador PI^λ . O vetor de posições da j -ésima partícula para este controlador é tridimensional, definido como

$$\mathbf{p}_j = \begin{bmatrix} k_p^{(j)} & k_i^{(j)} & \lambda^{(j)} \end{bmatrix}. \quad (7.8)$$

Para realizar a saturação dinâmica das partículas que ultrapassarem a superfície crítica, optou-se por limitar exclusivamente o parâmetro $k_i^{(j)}$. Desta forma, os parâmetros λ e k_p evoluem livremente dentro de seus respectivos limites pré-definidos, enquanto k_i é constantemente monitorado e restringido para garantir que a partícula permaneça em uma região estável do espaço de busca. A saturação dinâmica das partículas sugerida nesta tese é dada após a atualização de suas posições e é dada como

$$\mathbf{p}_j[k+1] = \min \left\{ \max(\mathbf{p}_j[k+1], \mathbf{p}_{min}), \mathbf{p}_j^{(max)}[k+1] \right\}, \quad (7.9)$$

em que, para o controlador PI^λ ,

$$\mathbf{p}_{min} = \begin{bmatrix} k_p^{(min)} & k_i^{(min)} & \lambda^{(min)} \end{bmatrix}, \quad (7.10)$$

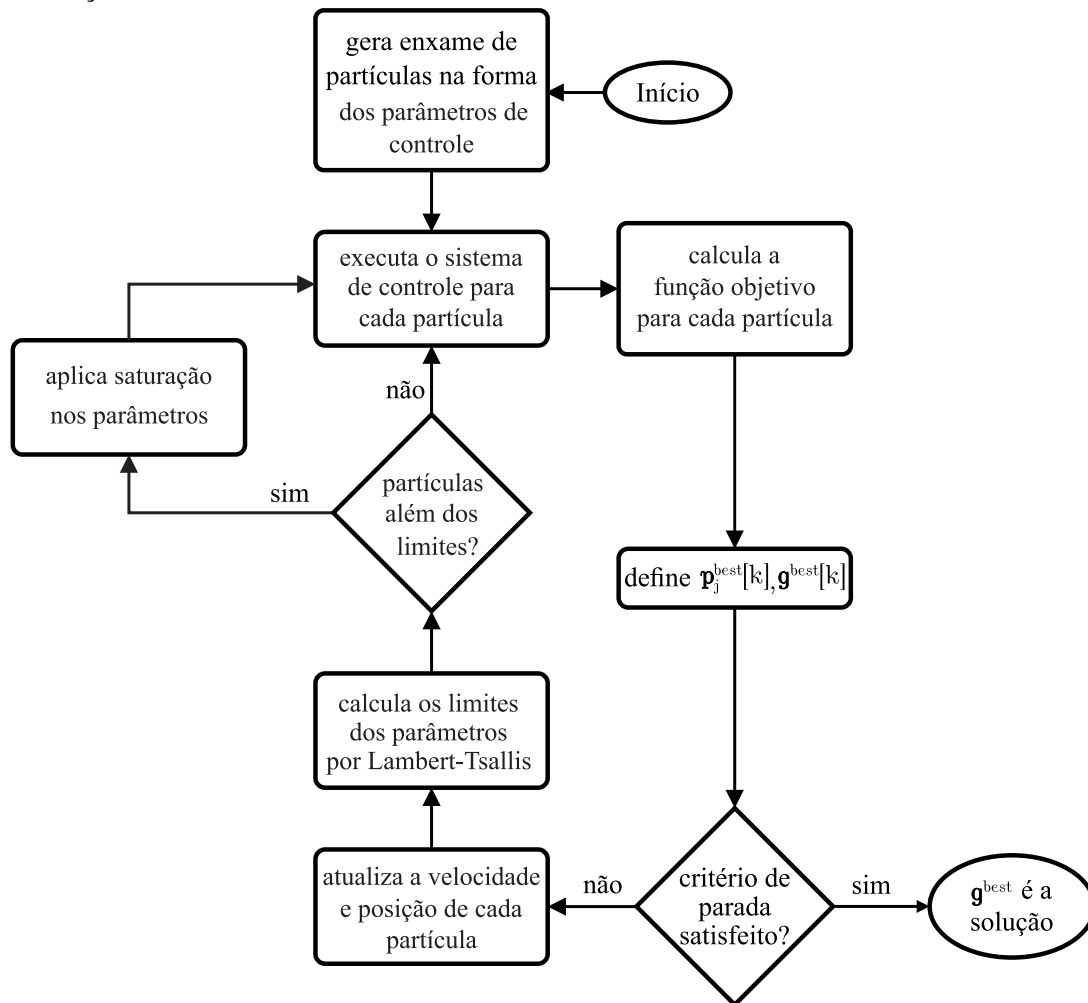
$$\mathbf{p}_j^{(max)}[k+1] = \begin{bmatrix} k_p^{(max)} & k_{i_c}^{(j)}[k+1] & \lambda^{(max)} \end{bmatrix} \quad (7.11)$$

representam os valores mínimos e máximos permitidos de \mathbf{p}_j , respectivamente. Para o caso 9 da [Tabela 6](#), $k_{i_c}^{(j)}[k]$ pode ser definido a partir da [Equação 4.44](#), para cada partícula no instante k , como

$$k_{i_c}^{(j)}[k] = -\frac{1}{bc} \left[\frac{\lambda^{(j)}[k] z_c(\lambda^{(j)}[k]) (a + bck_p^{(j)}[k])^{\frac{\alpha + \lambda^{(j)}[k]}{\lambda^{(j)}[k]}}}{\alpha} \right]^{\frac{\lambda^{(j)}[k]}{\alpha}}. \quad (7.12)$$

O fluxograma do processo de otimização proposto é ilustrado na [Figura 42](#).

Figura 42 – Fluxograma da metodologia de sintonia de parâmetros usando PSO conjuntamente com a função de Lambert-Tsallis.



Fonte: elaborada pelo autor.

7.2 Implementação embarcada de controladores de ordem fracionária

Um dos maiores problemas que surge na implementação de controladores $PI^\lambda D^\mu$ reside justamente em sua teoria matemática subjacente, o cálculo fracionário, que é bem mais complexo que o cálculo diferencial e integral de ordem inteira (PODLUBNY, 2001). Ressalta-se que, como visto na Seção 2.1, os operadores de derivada e integral fracionária são filtros lineares de dimensão infinita e todos os métodos de implementação são aproximações de dimensão finita.

É bastante comum que primeiro seja projetado o controlador $PI^\lambda D^\mu$ contínuo na Equação 2.25 e posteriormente seja sintetizada uma versão discreta aproximada. Porém, a otimalidade alcançada durante a fase de projeto é perdida após essa aproximação, uma vez que, na prática, qualquer sistema discreto possui limitações de espaço em memória. De fato, um

controlador ótimo projetado no tempo contínuo (*i.e.*, o controlador analógico) não será mais ótimo após a aproximação no tempo discreto (MERRIKH-BAYAT *et al.*, 2015).

O objetivo desta seção é investigar a hipótese de que é possível obter um modelo computacional capaz de reproduzir o comportamento dinâmico de um controlador $PI^\lambda D^\mu$ a partir de suas entradas reais: o erro atual e passados entre o *setpoint* do sistema de controle e a saída da planta. Além disso, a saída do modelo deve ser a variação do sinal de controle, o que proporciona maior suavidade à resposta do sistema, uma vez que essa condição de saída possui um integrador natural, capaz de eliminar o erro em regime permanente (PERRY *et al.*, 2007). O propósito dessa abordagem é mitigar a perda de otimalidade que ocorre nos métodos de discretização de controladores $PI^\lambda D^\mu$. Posteriormente, o modelo computacional do controlador $PI^\lambda D^\mu$ deve ser embarcado em um microcontrolador para realizar o controle do sistema, e os resultados obtidos devem ser comparados com um método de discretização da literatura.

Uma técnica amplamente utilizada em regressão não linear é a rede neural MLP (do inglês *multilayer perceptron*), um aproximador universal de funções multidimensionais (HORNIK *et al.*, 1989). Diversos autores utilizam redes neurais em sistemas de controle, como em Asgharnia *et al.* (2020), em que uma rede neural treinada seleciona os parâmetros do controlador $PI^\lambda D^\mu$ para cada cenário através de variáveis de *scheduling*. Munagala e Jatoth (2023) utilizaram uma MLP para aprender o comportamento de um controlador $PI^\lambda D^\mu$. Porém, as entradas do modelo são o *setpoint* de controle e a saída da planta, enquanto que a saída é o sinal de controle. Embora existam aplicações de redes neurais com controladores fracionários, estes trabalhos não abordaram a questão da implementação embarcada da forma que se propõe a fazer nesta tese, visando uma resposta dinâmica mais eficiente do controlador discretizado.

7.2.1 Aproximação discreta de controladores $PI^\lambda D^\mu$

Entre as formas existentes de implementação, destaca-se o controlador $PI^\lambda D^\mu$ discreto de memória longa (LD- $PI^\lambda D^\mu$, na sigla em inglês), proposto por Merrikh-Bayat *et al.* (2015), que consiste na discretização da variável contínua s na Equação 2.25 para z utilizando a transformação de Tustin e então realizando uma expansão em série de potência, que é truncada devido à limitação de memória. Nesta seção, a aproximação de Merrikh-Bayat é utilizada para fins de comparação com a proposta apresentada.

A aproximação proposta por Merrikh-Bayat *et al.* (2015) é baseada no mapeamento

modificado de Tustin, que aplicado à derivada fracionária s^μ , resulta em

$$s^\mu = \left[\frac{\omega_c}{\tan\left(\frac{\omega_c T}{2}\right)} \cdot \frac{1 - z^{-1}}{1 + z^{-1}} \right]^\mu = \alpha^\mu \left(\frac{1 - z^{-1}}{1 + z^{-1}} \right)^\mu, \quad (7.13)$$

em que $\alpha \triangleq \frac{\omega_c}{\tan\left(\frac{\omega_c T}{2}\right)}$, ω_c é a frequência de cruzamento de ganho da função de transferência

de malha aberta e $\tan(\cdot)$ é a função trigonométrica tangente. A expansão em série de potência da [Equação 7.13](#) é dada por

$$\alpha^\mu \left(\frac{1 - z^{-1}}{1 + z^{-1}} \right)^\mu = \alpha^\mu \sum_{k=0}^{\infty} f_k(\mu) (z^{-1})^k, \quad (7.14)$$

em que

$$f_k(\mu) = \frac{1}{k!} \cdot \frac{d^k}{d(z^{-1})^k} \left(\frac{1 - z^{-1}}{1 + z^{-1}} \right) \Big|_{z^{-1}=0}. \quad (7.15)$$

Inserindo a [Equação 7.13](#) e a [Equação 7.14](#) na [Equação 2.25](#), com $s^{-\lambda} = \left(\frac{1}{s}\right) s^{1-\lambda}$, a expressão de aproximação do controlador $PI^\lambda D^\mu$ resulta em

$$C(z) = K_p + K_i \frac{1+z^{-1}}{1-z^{-1}} \sum_{k=0}^M f_k(1-\lambda) z^{-k} + K_d \sum_{k=0}^M f_k(\mu) z^{-k}, \quad (7.16)$$

em que

$$K_p = k_p, \quad K_i = k_i \alpha^{-\lambda}, \quad K_d = k_d \alpha^\mu, \quad (7.17)$$

e M é o tamanho da memória considerada na aproximação¹. Se a [Equação 7.16](#) for desenvolvida, pode-se reescrevê-la como

$$C(z) = \frac{w_0 + w_1 z^{-1} + w_2 z^{-2} + \dots + w_{M+1} z^{-(M+1)}}{1 - z^{-1}}, \quad (7.18)$$

em que w_0, \dots, w_6 são constantes resultantes do desenvolvimento da equação. Os autores da proposta de aproximação discreta não as desenvolvem, apresentando apenas a expansão em série de potência mostrada na [Equação 7.15](#). Tentativas de desenvolvimento destas equações são descritas em [El-Shafei et al. \(2017\)](#) e [Santos et al. \(2022\)](#), porém as equações nestas referências apresentam erros. O desenvolvimento correto dessas equações, feitas de próprio punho, é

¹É importante ressaltar que k_p, k_i e k_d são os ganhos do controlador no domínio do tempo contínuo, ou seja, o controlador analógico apresentado na [Equação 2.25](#). As constantes K_p, K_i e K_d são as versões destes ganhos na aproximação discreta proposta por [Merrikh-Bayat et al. \(2015\)](#).

apresentado a seguir. Posteriormente, para fins de verificação, os resultados foram confirmados com o uso da *toolbox* de matemática simbólica no *software* MATLAB®.

Para a aproximação do controlador $PI^\lambda D^\mu$, a [Equação 7.16](#) é desenvolvida com $M = 5$, valor utilizado por [Merrikh-Bayat et al. \(2015\)](#) e [El-Shafei et al. \(2017\)](#). Assim, obtém-se a [Equação 7.18](#) com

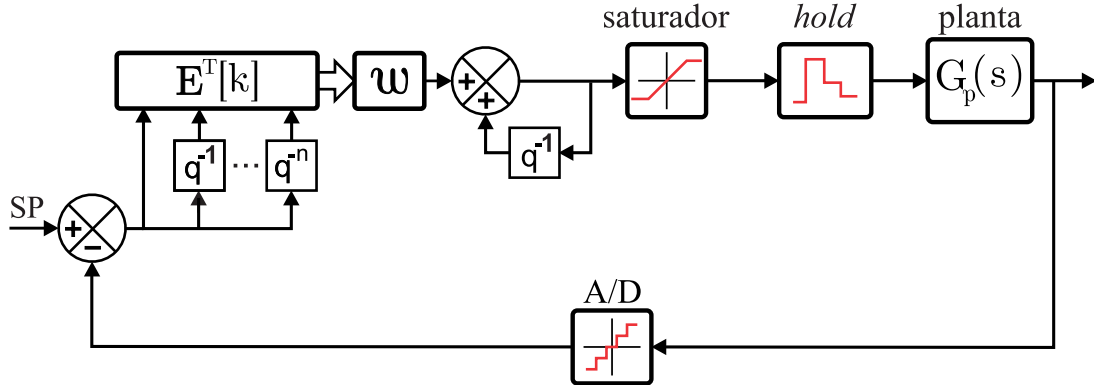
$$\begin{aligned}
 w_0 &= K_p + K_i + K_d \\
 w_1 &= -\{K_p + K_d(2\mu + 1) + K_i[2(1-\lambda) - 1]\} \\
 w_2 &= 2\{K_d(\mu^2 + \mu) + K_i[(1-\lambda)^2 - (1-\lambda)]\} \\
 w_3 &= -2\left\{K_d\left(\frac{2}{3}\mu^3 + \mu^2 + \frac{1}{3}\mu\right) + K_i\left[\frac{2}{3}(1-\lambda)^3 - (1-\lambda)^2 + \frac{1}{3}(1-\lambda)\right]\right\} \\
 w_4 &= 2\left\{K_d\left(\frac{1}{3}\mu^4 + \frac{2}{3}\mu^3 + \frac{2}{3}\mu^2 + \frac{1}{3}\mu\right) + \right. \\
 &\quad \left. + K_i\left[\frac{1}{3}(1-\lambda)^4 - \frac{2}{3}(1-\lambda)^3 + \frac{2}{3}(1-\lambda)^2 - \frac{1}{3}(1-\lambda)\right]\right\} \\
 w_5 &= -2\left\{K_d\left(\frac{2}{15}\mu^5 + \frac{1}{3}\mu^4 + \frac{2}{3}\mu^3 + \frac{2}{3}\mu^2 + \frac{1}{5}\mu\right) + \right. \\
 &\quad \left. + K_i\left[\frac{2}{15}(1-\lambda)^5 - \frac{1}{3}(1-\lambda)^4 + \frac{2}{3}(1-\lambda)^3 - \frac{2}{3}(1-\lambda)^2 + \frac{1}{5}(1-\lambda)\right]\right\} \\
 w_6 &= 2\left\{K_d\left(\frac{2}{15}\mu^5 + \frac{2}{3}\mu^3 + \frac{1}{5}\mu\right) - K_i\left[\frac{2}{15}(1-\lambda)^5 + \frac{2}{3}(1-\lambda)^3 + \frac{1}{5}(1-\lambda)\right]\right\}.
 \end{aligned} \tag{7.19}$$

Por fim, a [Equação 7.18](#) para $M = 5$ pode ser reescrita como

$$u[k] = u[k-1] + \mathbf{w}^T \mathbf{E}[k], \tag{7.20}$$

em que $\mathbf{w} = [w_0 \dots w_6]^T$ e $\mathbf{E}[k] = [E[k] \dots E[k-6]]^T$. A implementação do sistema de controle com o controlador $PI^\lambda D^\mu$ aproximado pela técnica de Merrikh-Bayat é ilustrada na [Figura 43](#).

Figura 43 – Implementação do sistema de controle com a aproximação de Merrikh-Bayat.



Fonte: elaborada pelo autor.

7.2.2 Aplicação da rede neural MLP

Após a obtenção dos parâmetros do controlador $PI^\lambda D^\mu$ por meio do PSO, uma rede neural MLP com uma camada oculta é treinada. Detalhes sobre a arquitetura de rede utilizada neste capítulo estão disponíveis no [Apêndice B](#). O diagrama do sistema de controle proposto para o treinamento da rede neural é apresentado na [Figura 44](#), em que SP representa o *setpoint* do sistema de controle. Para garantir uma comparação justa, a rede neural deve receber como entradas o erro atual e um determinado número de valores passados do sistema de controle com o controlador $PI^\lambda D^\mu$, de forma que o número total de entradas seja igual ao utilizado na aproximação de Merrikh-Bayat (vide [Equação 7.20](#)). Para gerar os dados utilizados para o treinamento da rede neural, o sistema de controle, o que inclui os sinais de entrada e saída do controlador $PI^\lambda D^\mu$, pode ser simulado através do método SISFRAC proposto no [Capítulo 5](#).

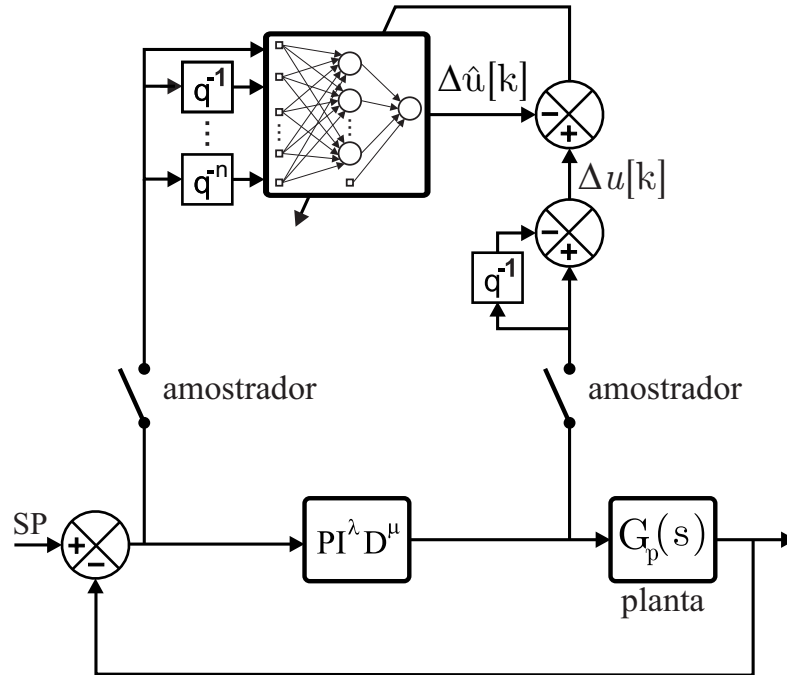
A saída da rede neural é uma estimativa da variação do sinal de controle $\Delta \hat{u}[k]$, e não a estimativa do próprio sinal de controle $\hat{u}[k]$. Essa escolha permite a implementação de um integrador natural, contribuindo para a eliminação do erro no estado permanente. Como o controlador $PI^\lambda D^\mu$ é um filtro, é importante que os dados usados para treinar a MLP não sejam distorcidos por um saturador.

A implementação do sistema de controle com o controlador $PI^\lambda D^\mu$ neural é ilustrada na [Figura 45](#). Como a saída da rede neural é a estimativa da variação do sinal de controle $\Delta \hat{u}$, a estimativa atual do sinal de controle \hat{u} que é enviada para a planta é obtida recursivamente como

$$\hat{u}[k] = \hat{u}[k-1] + \Delta \hat{u}[k]. \quad (7.21)$$

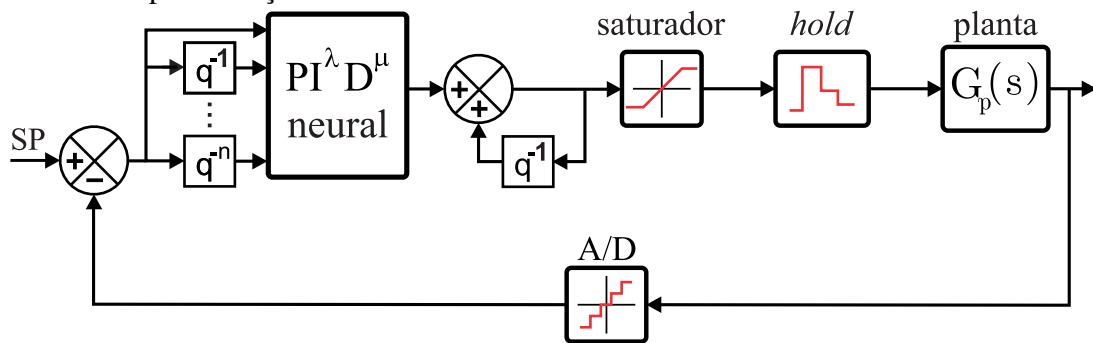
A implementação deste sistema de controle possui o mesmo saturador presente durante a etapa de busca de parâmetros com o algoritmo PSO (vide [Subseção 7.1.2](#)), necessário

Figura 44 – Diagrama do sistema de controle usado para o treinamento da rede neural.



Fonte: elaborada pelo autor.

Figura 45 – Implementação do sistema de controle fracionário com o controlador $PI^\lambda D^\mu$ neural.



Fonte: elaborada pelo autor.

para garantir que o sinal de controle permaneça dentro dos valores previamente estabelecidos para a implementação real. O mesmo se aplica ao sistema de controle com o controlador aproximado de Merrikh-Bayat.

7.2.2.1 Avaliação do sistema simulado em uma plataforma embarcada

Para avaliar a melhor configuração da rede neural, propõe-se analisar cinco índices: o RMSE, presente na [Equação 5.16](#), o coeficiente de correlação R , o desvio padrão de R , σ_R , o

coeficiente de determinação R^2 e o erro médio de viés (MBE), definidos como

$$R = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n(\sum x^2) - (\sum x)^2][n(\sum y^2) - (\sum y)^2]}} \quad (7.22)$$

$$\sigma_R = \sqrt{\frac{1}{n} \sum_{i=1}^n (R_i - \bar{R})^2}, \quad (7.23)$$

$$\text{MBE} = \frac{1}{n} \sum_{i=1}^n (y_i - x_i), \quad (7.24)$$

em que x_i e y_i representam $\Delta \hat{u}$ e Δu , respectivamente.

Sugere-se que o sistema de controle embarcado utilize a seguinte sequência de operações: o microcontrolador solicita a informação atual do sensor, associada à variável de processo, calcula o erro atual, realiza os cálculos da MLP gerando a variação do sinal de controle $\Delta \hat{u}[k]$, e finalmente atualiza o sinal de controle.

Por fim, o tempo de execução, o uso de memória do programa e o uso de RAM podem ser medidos e analisados em cada uma das etapas envolvidas no processo: resposta do sensor, atualização das entradas e normalização dos dados, somatório da camada oculta da MLP, funções de ativação da camada oculta, somatório da camada de saída, desnormalização dos dados e atualização da saída do controlador.

No próximo capítulo, a proposta de implementação de um controlador fracionário através de uma rede neural MLP é testada através do estudo de caso de um sistema real, em que o controlador neural é embarcado em um microcontrolador de 8 bits a fim de avaliar seu desempenho em uma planta real. Os índices mencionados anteriormente, assim como o tempo de execução e uso de memória são avaliados.

8 ESTUDO DE CASO: INCUBADORA NEONATAL

Neste capítulo, os conceitos abordados nos capítulos anteriores são aplicados didaticamente em um caso real de sistema de controle fracionário, cuja planta é uma incubadora neonatal comercial. A incubadora pertence ao laboratório SPIRAL (*signal and information processing for data analysis and learning systems*), situado no *campus* do Pici, da Universidade Federal do Ceará - UFC. Este laboratório coordena a linha de pesquisa em *monitoramento remoto de pacientes* no âmbito do projeto CEREIA (centro de referência em inteligência artificial), contemplado no edital FAPESP – MCTIC - CGI.BR para Centros de Pesquisas Aplicadas em Inteligência Artificial.

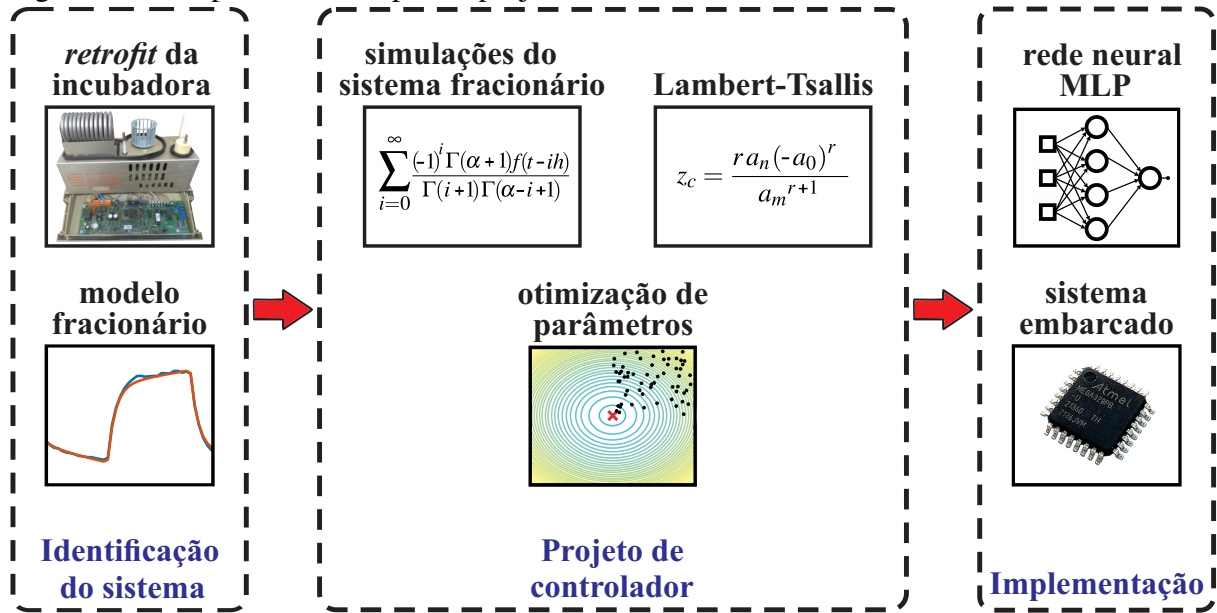
Neste estudo de caso, o sistema de controle deve regular a temperatura interna de uma incubadora neonatal por meio da atuação na tensão média aplicada a um resistor de aquecimento, controlado por um conversor monofásico CA/CC. O objetivo é manter a temperatura em $36,1^{\circ}\text{C}$, valor escolhido por oferecer maior segurança contra *overshoot* térmico durante os testes. Esta temperatura está dentro de faixas encontradas na literatura técnica, como o estudo de [Amadi et al. \(2017\)](#), que propõe valores entre $35,5^{\circ}\text{C}$ e $36,5^{\circ}\text{C}$ para o chamado *warm-up setpoint*, ou ponto de ajuste inicial de aquecimento do sistema. As etapas utilizadas no projeto de sistema de controle fracionário da incubadora, envolvendo todos os conceitos propostos nos capítulos anteriores são ilustradas na [Figura 46](#).

Por se tratar de um equipamento comercial, a incubadora não permite acesso ao sistema de potência, o que impede que usuários externos realizem o controle de temperatura de forma autônoma. Consequentemente, é necessário realizar um *retrofit* na placa eletrônica, ignorando o microcontrolador original e substituindo-o por um mais moderno, de modo a ter acesso ao atuador do equipamento e viabilizar intervenções no sistema para a realização de ensaios de identificação e controle da planta. O *retrofit* realizado na incubadora está documentado no [Apêndice C](#), com detalhes da intervenção feita na placa eletrônica, da definição do sinal de controle, da aquisição de dados para identificação do sistema e do diagrama de blocos detalhado do sistema de controle a ser implementado (vide [Figura 71](#)).

Com o *retrofit* concluído, é realizada a identificação da planta por função de transferência. Esta etapa tem como objetivo obter um modelo matemático de função de transferência representativo do comportamento dinâmico do sistema, que será utilizado na etapa de sintonia dos parâmetros do controlador.

Para controlar o sistema, opta-se por um controlador de ordem fracionária do tipo

Figura 46 – Etapas utilizadas para o projeto de sistema de controle fracionário.



Fonte: elaborada pelo autor.

PI^λ , com o objetivo de eliminar o erro em regime permanente do sistema. Para a obtenção dos ganhos do controlador, usa-se o algoritmo PSO-LT desenvolvido no capítulo anterior a fim de aplicar limites de busca dinâmicos que garantam a permanência das partículas dentro do hiperespaço estável. Por fim, o controlador resultante da otimização é implementado no sistema embarcado do *retrofit* da incubadora através do método proposto também no capítulo anterior que utiliza regressão não linear via redes neurais artificiais.

Dessa forma, este capítulo é dividido em três seções. Na [Seção 8.1](#), são apresentados os resultados referentes à identificação de um modelo por função de transferência que represente a planta. Na [Seção 8.2](#), são apresentados os resultados a respeito da aplicação da proposta de sintonia de controladores via algoritmo PSO-LT. Finalmente, resultados da implementação embarcada de controladores de ordem fracionária são apresentados na [Seção 8.3](#).

8.1 Identificação da incubadora neonatal

Com base nos trabalhos de [Poinot et al. \(2002\)](#) e [Petráš \(2011\)](#), que modelaram sistemas térmicos utilizando uma estrutura de ordem fracionária do tipo $G_p(s) = \frac{c}{s^\alpha + a}$, decidiu-se avaliar se este modelo é capaz de representar a planta em estudo. Os dados utilizados para a identificação estão ilustrados na [Figura 70](#) do [Apêndice C](#), que possuem uma entrada $u[k]$ do tipo PRBS (*pseudo-random binary signal*) com o objetivo de excitar adequadamente as dinâmicas internas do sistema e satisfazer a condição de persistência de excitação.

Para a realização da identificação do modelo proposto, é utilizado o algoritmo de identificação de sistemas de ordem fracionária proposto no [Capítulo 6](#), mais especificamente na [Seção 6.4](#). O valor do parâmetro α é iniciado entre 1 e 1,5, enquanto que os parâmetros a e c são iniciados com valores da ordem de 10^{-3} e 10^{-6} respectivamente. A taxa de aprendizado μ possui decaimento linear conforme a [Equação 6.14](#) com $\mu_{inicial} = \begin{bmatrix} 10^{-5} & 10^{-11} & 1,5 \cdot 10^{-11} \end{bmatrix}^T$. Foram utilizadas 40.000 épocas de treinamento. A grande quantidade de épocas utilizadas se deve aos valores extremamente pequenos de μ e da natureza do problema, uma vez que a derivada fracionária depende de todos os seus valores passados, exigindo um processo de identificação mais prolongado. Os valores mencionados foram definidos com base em testes preliminares, com o objetivo de garantir a convergência dos parâmetros. Os resultados são apresentados na [Figura 47](#), [Figura 48](#) e [Tabela 15](#).

Embora a função de autocorrelação (FAC) dos resíduos da identificação ilustrada na [Figura 48](#) demonstre que existam componentes não desprezíveis nas defasagens 1 e 2, os valores finais de RMSE e VAF apontam que o algoritmo de identificação conseguiu encontrar um conjunto de parâmetros que representa de maneira satisfatória os dados de aquisição, explicando 97,84% da variância dos dados, como indica o índice VAF. Observa-se que os parâmetros convergiram, com evolução suave. O algoritmo de identificação proposto foi excepcionalmente rápido, com tempo de execução de apenas 20,27 segundos, evidenciando a simplicidade matemática do algoritmo composto apenas de operações simples, como soma e multiplicação, vantagens herdadas do algoritmo LMS. Por fim, baseado nos resultados apresentados na [Tabela 15](#), o modelo identificado é

$$G_p(s) = \frac{1,9922 \cdot 10^{-5}}{s^{1,10} + 1,4176 \cdot 10^{-4}}. \quad (8.1)$$

É importante ressaltar que o algoritmo proposto resulta em um modelo contínuo, ou seja, no domínio s de Laplace, o que difere da abordagem convencional dos algoritmos baseados no LMS, que geralmente são utilizados para identificação de modelos discretos. Por ser um modelo contínuo, o mesmo pode ser utilizado na análise de estabilidade desenvolvida no [Capítulo 4](#). Esta análise é realizada na seção a seguir.

Figura 47 – Resultados da identificação dos dados de aquisição da incubadora.

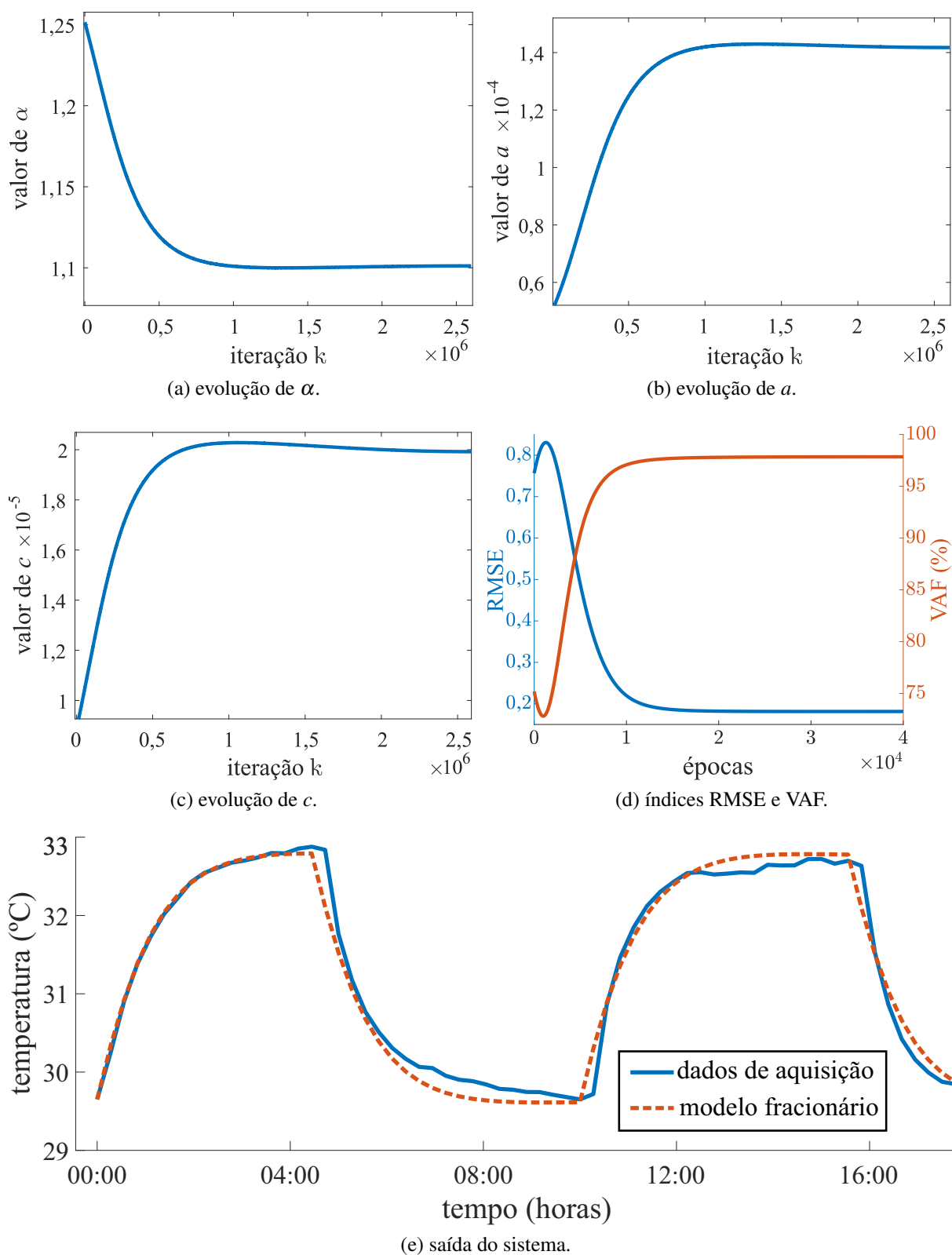
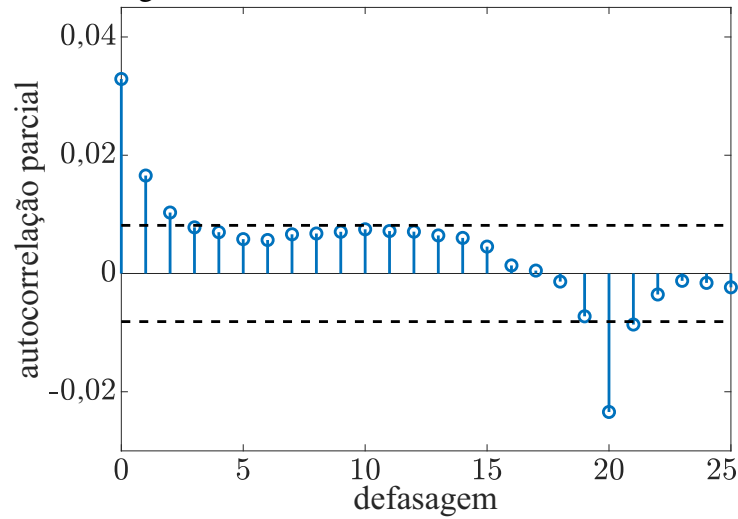


Figura 48 – FAC dos resíduos.



Fonte: elaborada pelo autor.

Tabela 15 – Métricas e parâmetros da identificação da incubadora.

parâmetro	inicial	final	RMSE	VAF
α	1,251520	1,101190	0,181366	97,842567%
a	$5,01896 \cdot 10^{-3}$	$1,4176 \cdot 10^{-4}$		
c	$8,85424 \cdot 10^{-6}$	$1,9922 \cdot 10^{-5}$		

Fonte: elaborada pelo autor.

8.2 Sintonia dos parâmetros de um controlador de ordem fracionária

Como destacado no início deste capítulo, optou-se pelo controlador PI^λ para o projeto de sistema de controle de ordem fracionária. A combinação deste controlador juntamente com o modelo identificado na [Equação 8.1](#) posiciona este sistema de controle no caso 9 da [Tabela 6](#). Desta forma, o sistema está elegível para análise via função de Lambert-Tsallis. Esta seção está subdividida da seguinte forma. A [Subseção 8.2.1](#) apresenta a análise *a priori* da estabilidade do sistema de controle da incubadora neonatal, enquanto que a [Subseção 8.2.2](#) apresenta os resultados da aplicação do algoritmo de sintonia proposto, que utiliza a Lambert-Tsallis para garantir a estabilidade *a priori* do espaço de busca do algoritmo PSO.

8.2.1 Estabilidade *a priori* via função Lambert-Tsallis

A função de transferência de malha fechada do sistema de controle fracionário composto pelo modelo da incubadora, controlador PI^λ e ganho de realimentação unitário é dada por

$$G_{cl}(s) = \frac{1,9922 \cdot 10^{-5} k_p s^\lambda + 1,9922 \cdot 10^{-5} k_i}{s^{1,10+\lambda} + (1,4176 \cdot 10^{-4} + 1,9922 \cdot 10^{-5} k_p) s^\lambda + 1,9922 \cdot 10^{-5} k_i}. \quad (8.2)$$

Pelo princípio da identidade de polinômios entre a equação característica de $G_{cl}(s)$ e a [Equação 4.1](#), conclui-se que

$$\begin{aligned} n &= 1,10 + \lambda \\ m &= \lambda \\ a_n &= 1 \\ a_m &= 1,4176 \cdot 10^{-4} + 1,9922 \cdot 10^{-5} k_p \\ a_0 &= 1,9922 \cdot 10^{-5} k_i, \end{aligned} \quad (8.3)$$

e conseqüentemente,

$$\begin{aligned} r &= \frac{n-m}{m} \\ r &= \frac{1,10}{\lambda}. \end{aligned} \quad (8.4)$$

Para obter as equações que descrevem os ganhos críticos de k_p e k_i , ou seja, k_{pc} e k_{ic} , aplicam-se estes valores nas [Equações 4.43](#) e [4.44](#), de modo que

$$k_{pc} = \frac{1}{1,9922 \cdot 10^{-5}} \left\{ \left[\frac{1,10 (-1,9922 \cdot 10^{-5} k_i)^{\frac{1,10}{\lambda}}}{\lambda z_c} \right]^{\frac{\lambda}{1,10+\lambda}} - 1,4176 \cdot 10^{-4} \right\} \quad (8.5)$$

e

$$k_{ic} = -\frac{1}{1,9922 \cdot 10^{-5}} \left[\frac{z_c \lambda (1,4176 \cdot 10^{-4} + 1,9922 \cdot 10^{-5} k_p)^{\frac{1,10+\lambda}{\lambda}}}{1,10} \right]^{\frac{\lambda}{1,10}}. \quad (8.6)$$

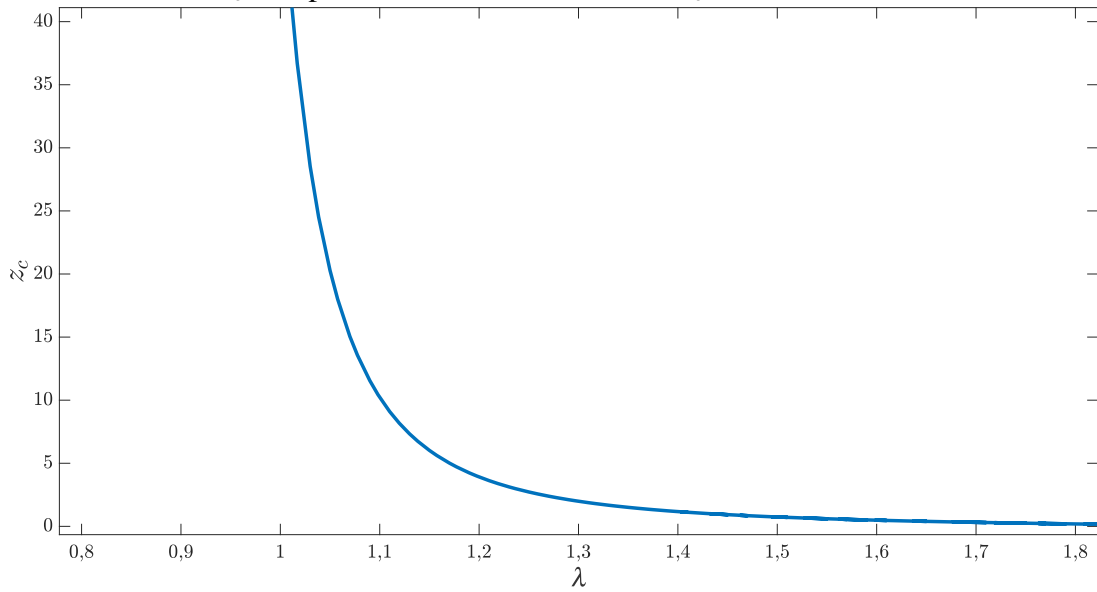
Para o cálculo de z_c (vide [Algoritmo 3](#)) é necessário calcular o valor de θ_c . Aplicando os dados do sistema em questão na [Equação 4.23](#), tem-se que

$$\theta_c = mr \frac{\pi}{2} = 99^\circ. \quad (8.7)$$

De forma análoga à [Figura 26](#), o gráfico de $z_c \times \lambda$ para o caso 9 da [Tabela 6](#) com $\theta_c = 99^\circ$ foi construído a partir do [Algoritmo 3](#) para o sistema de controle em questão e é ilustrado na [Figura 49](#). Observa-se que o mesmo é diferente do gráfico do [Exemplo 4.2.4](#), uma vez que os ângulos críticos θ_c são diferentes.

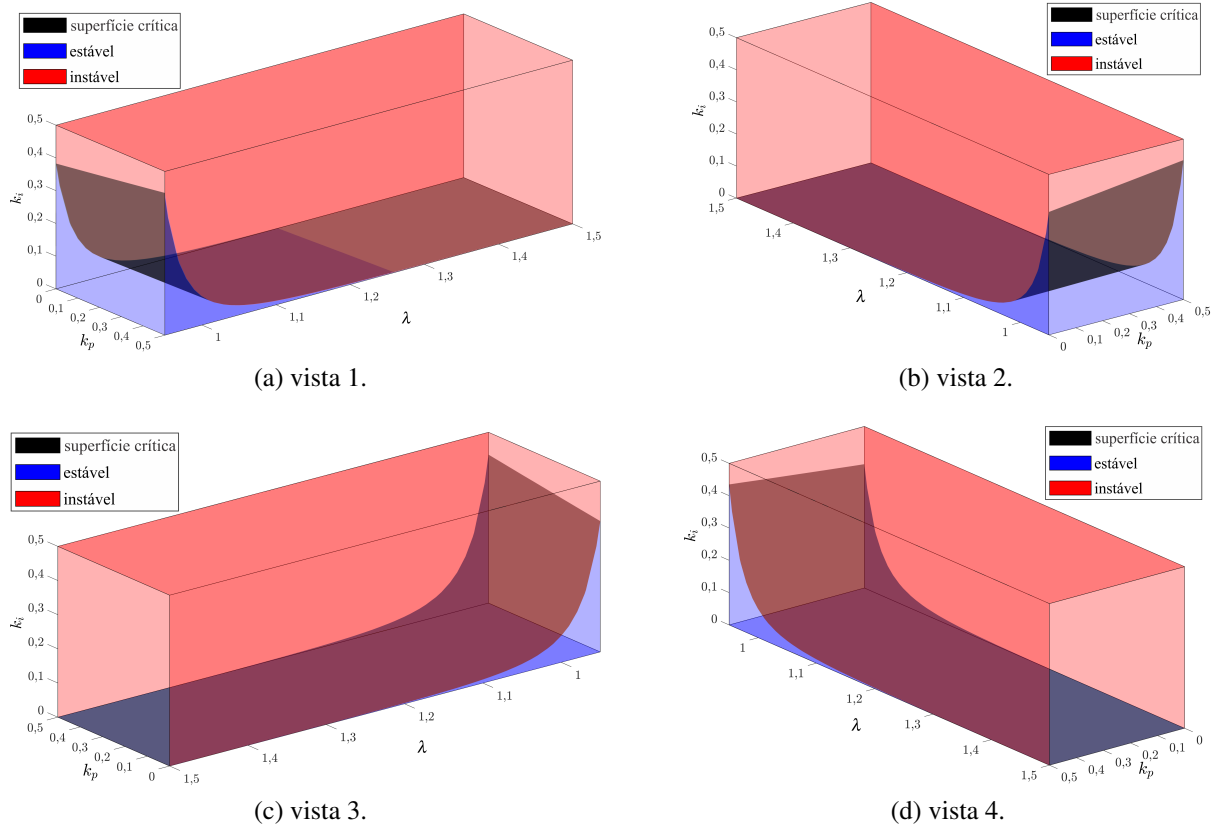
A partir de k_{pc} e k_{ic} e dos valores de $z_c \times \lambda$ presentes na [Figura 49](#), é possível obter o volume tridimensional de estabilidade do sistema de controle em questão de forma análoga ao da [Figura 27](#). O resultado é ilustrado na [Figura 50](#). O volume de estabilidade é validado *a posteriori*

Figura 49 – Gráfico $z_c \times \lambda$ para caso 9 da Tabela 6 com $\theta_c = 99^\circ$.



Fonte: elaborada pelo autor.

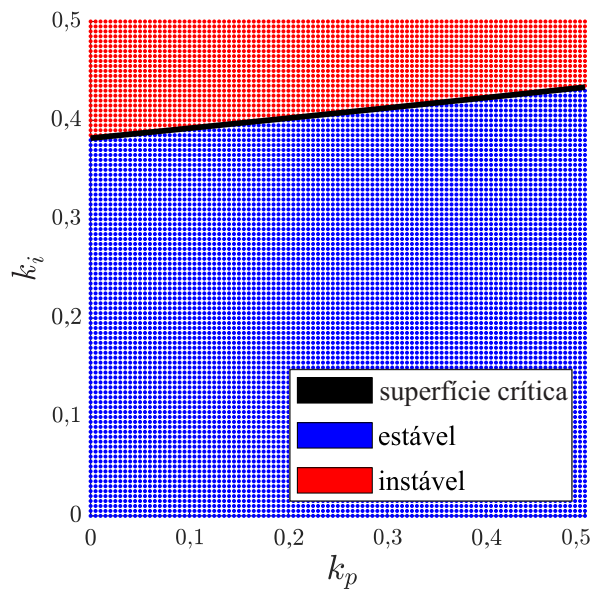
Figura 50 – Volume de estabilidade do sistema de controle fracionário proposto.



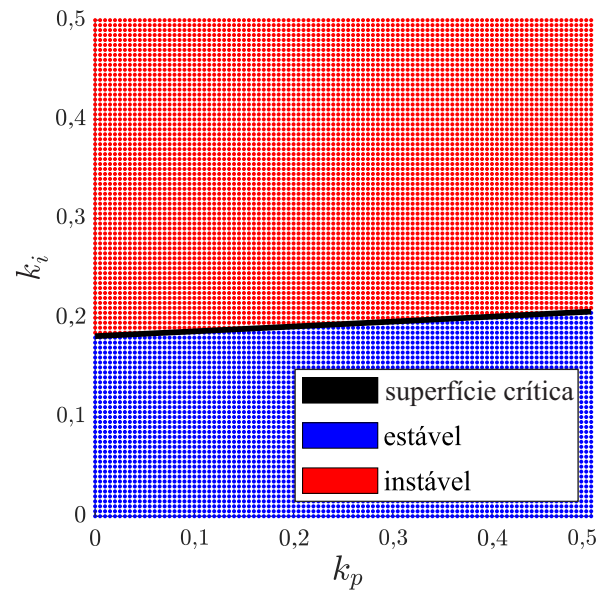
Fonte: elaborada pelo autor.

pela utilização do teorema de [Matignon \(1996\)](#), cujos resultados são ilustrados na [Figura 51](#) para alguns valores de λ . Este volume de estabilidade será o espaço de busca do algoritmo proposto para sintonia de controladores fracionários, cujos resultados são apresentados na próxima seção.

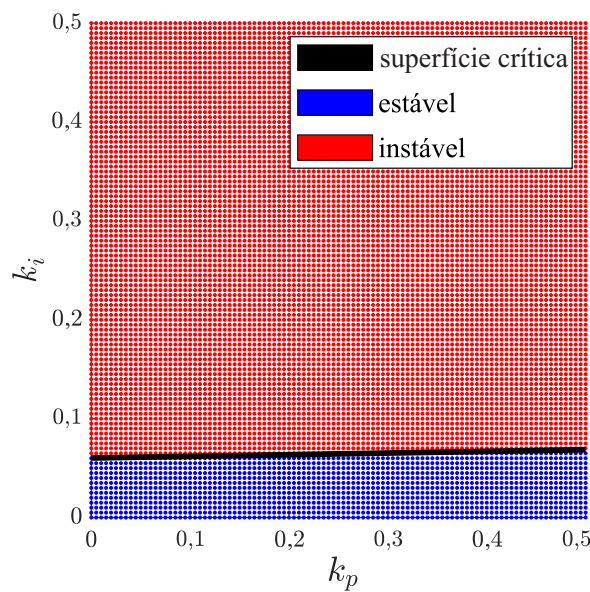
Figura 51 – Teste de estabilidade *a posteriori* do sistema proposto, com passo de 0,01.



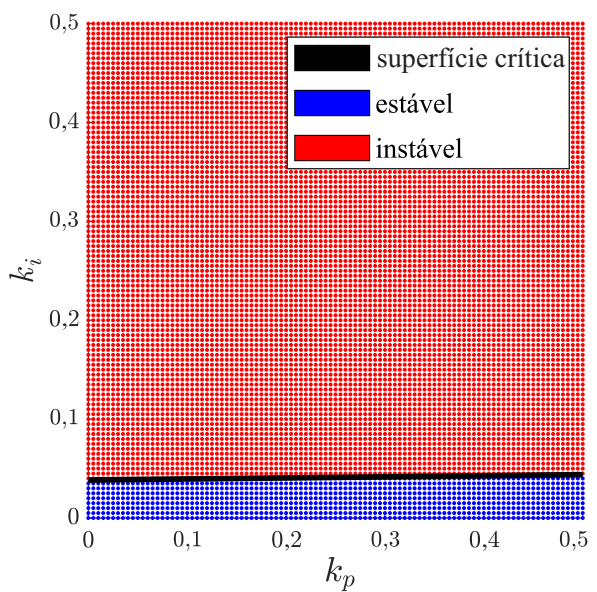
(a) $\lambda = 0.95$.



(b) $\lambda = 0.97$.



(c) $\lambda = 1.01$.



(d) $\lambda = 1.03$.

Fonte: elaborada pelo autor.

8.2.2 Aplicação do algoritmo PSO-LT

Uma vez que o controlador a ser utilizado é o PI^λ , o vetor de posições da j -ésima partícula é tridimensional e é definido pela [Equação 7.8](#), ou seja, $p_j = [k_p^{(j)} \quad k_i^{(j)} \quad \lambda^{(j)}]$.

Para garantir que o sistema projetado permaneça estável durante todo o processo de otimização, o espaço de busca foi restrito ao volume de estabilidade calculado analiticamente na [Figura 50](#). Para isso, optou-se pela manipulação do parâmetro $k_i^{(j)}$. Desta forma, os parâmetros λ e k_p são atualizados livremente, dentro de limites pré-estabelecidos, enquanto k_i é constantemente monitorado. Para cada valor de λ e k_p , o ganho k_i deve obedecer a uma condição do tipo $k_i \leq f(\lambda, k_p)$. Nesta tese, adota-se $f(\lambda, k_p)$ como o valor crítico de k_i , ou seja, k_{ic} , apresentado na [Equação 7.12](#) para cada partícula no instante k .

O algoritmo PSO-LT foi executado com os hiperparâmetros descritos a seguir. Foram utilizadas 10 realizações, em que cada realização é formada por um enxame de 12 partículas. É adotada a taxa decrescente para o fator de peso inercial descrita na [Equação 7.5](#), cujos valores inicial e final são $w_{inicial} = 0,01$ e $w_{final} = 0,001$. Os coeficientes de aprendizagem cognitiva e social utilizados são $c_1 = 1,2$ e $c_2 = 1$. Após uma análise preliminar, os valores mínimos de cada dimensão da partícula foram definidos como

$$p_{min} = [k_p^{(min)} \quad k_i^{(min)} \quad \lambda^{(min)}] = [0 \quad 0 \quad 0,9], \quad (8.8)$$

enquanto que os valores máximos são definidos como

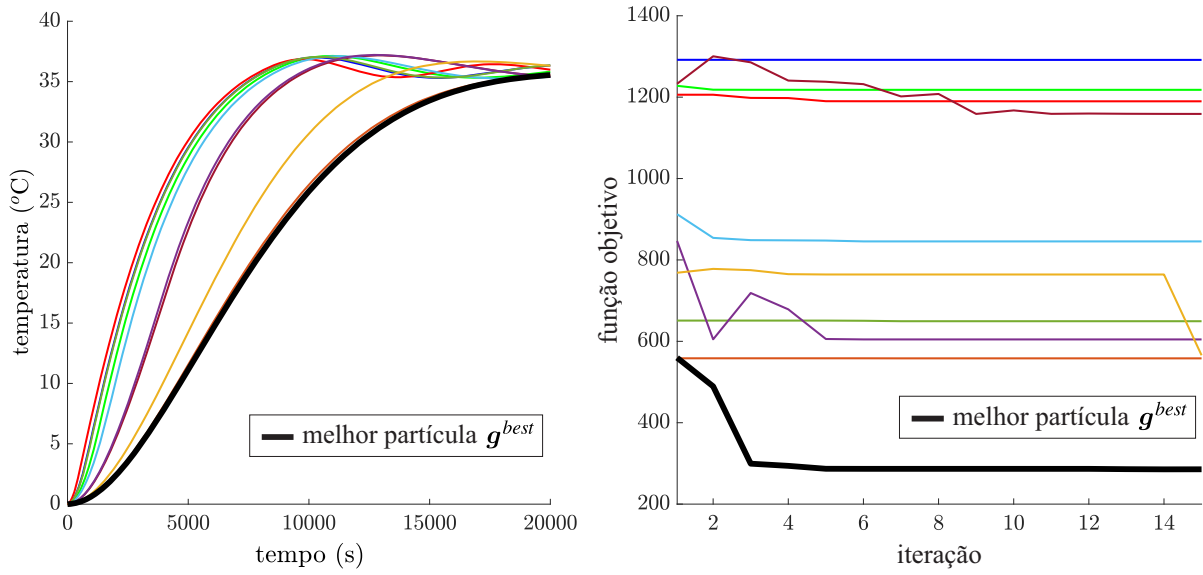
$$p_j^{(max)}[k+1] = [k_p^{(max)} \quad k_{ic}^{(j)}[k+1] \quad \lambda^{(max)}] = [0,01 \quad k_{ic}^{(j)}[k+1] \quad 1,3]. \quad (8.9)$$

Para avaliar o desempenho das partículas, utiliza-se a função objetivo apresentada na [Equação 7.7](#), que combina parâmetros da resposta ao degrau com a regularização dos ganhos $k_p^{(j)}$ e $k_i^{(j)}$ pela norma $L1$, com $v = -100$ e $\mu_{reg} = 50.000$, obtidos através de busca em grade.

Para a simulação da resposta ao degrau do sistema de controle de ordem fracionária, utiliza-se uma referência de $36,1^\circ \text{ C}$ e tempo de simulação $t_{final} = 20.000$ segundos. A simulação da saída da planta de ordem fracionária, $y[k]$, e da saída do controlador PI^λ , $u[k]$, são realizadas a partir do método SISFRAC (vide [Capítulo 5](#)), adaptadas diretamente da equação discreta de uso geral desenvolvida na [Equação 5.32](#) e da equação discreta do controlador $PI^\lambda D^\mu$ desenvolvida na [Equação 5.36](#), respectivamente, para o sistema de controle em questão como

$$y[k] = \frac{1}{(1,10w_0 + 1,4176 \cdot 10^{-4} T^{1,10})} \left\{ 1,9922 \cdot 10^{-5} T^{1,10} u[k] - \left(\sum_{i=1}^{k-1} 1,10 w_i y[k-i] \right) \right\} \quad (8.10)$$

Figura 52 – Desempenho das 10 partículas g^{best} .



(a) Resposta temporal do sistema de controle.

(b) Evolução da função objetivo.

Fonte: elaborada pelo autor.

Tabela 16 – Parâmetros ótimos do controlador PI^λ .

k_p	k_i	λ
0,0045647	0,0011333	0,97532

Fonte: elaborada pelo autor.

e

$$u[k]_{k=2:t_{final}} = \frac{1}{\lambda w_0} \left[(\lambda w_0 k_p + k_i T^\lambda) e[k] + k_p \left(\sum_{i=1}^{k-1} \lambda w_i e[k-i] \right) - \left(\sum_{i=1}^{k-1} \lambda w_i u[k-i] \right) \right] \quad (8.11)$$

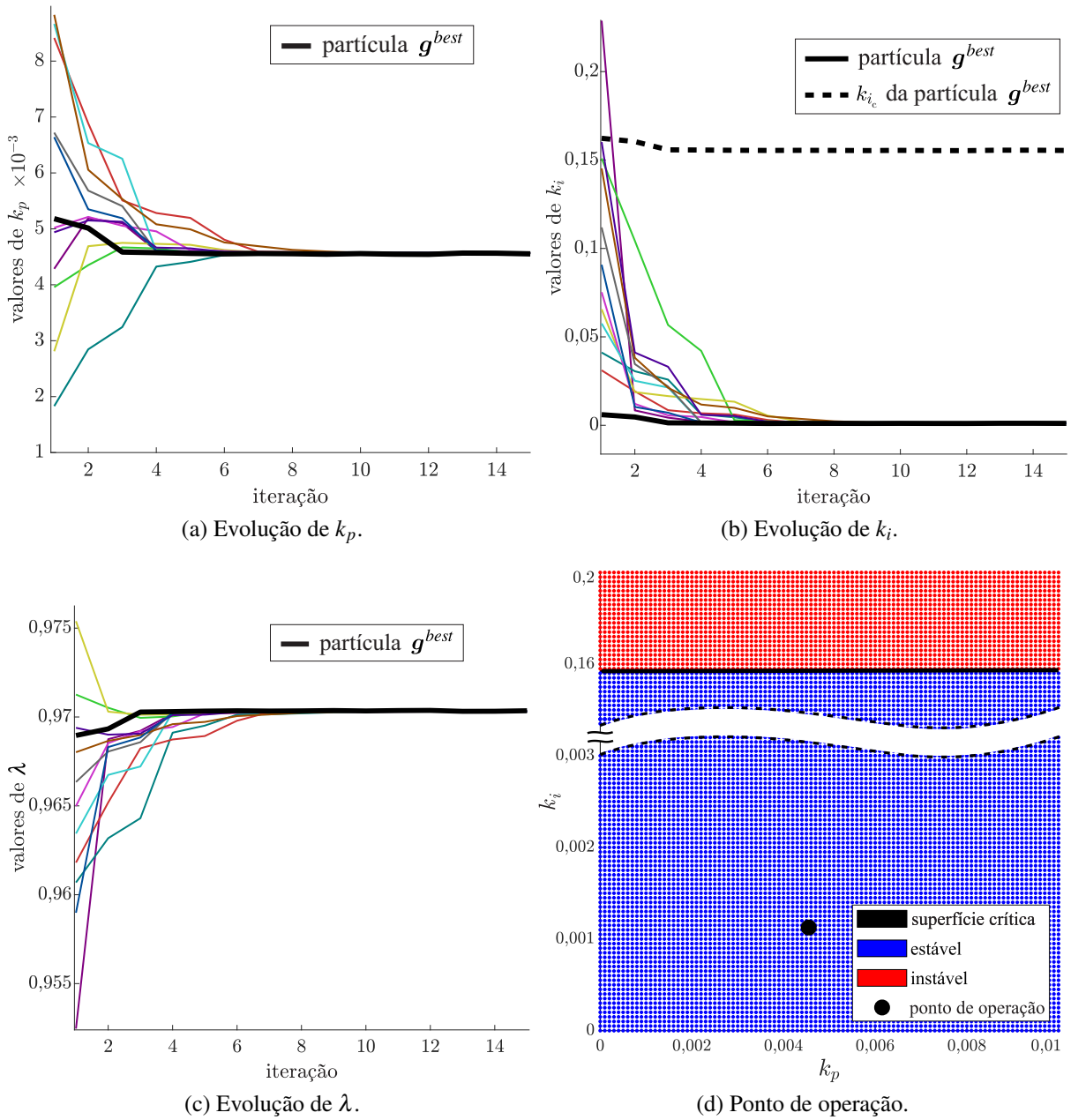
$$u[k]_{k=2:t_{final}} = \frac{1}{\lambda w_0} \left[(\lambda w_0 k_p + k_i T^\lambda) e[k] + \left(\sum_{i=1}^{k-1} \lambda w_i (k_p e[k-i] - u[k-i]) \right) \right]. \quad (8.12)$$

respectivamente. O período de amostragem T escolhido é de 1 segundo, considerado suficientemente baixo para o problema em questão. Os resultados são apresentados a seguir.

Na Figura 52.a, são ilustradas as respostas temporais dos sistemas de controle das 10 partículas $g^{best}[k]$, enquanto que na Figura 52.b são ilustradas as evoluções das suas respectivas funções objetivo. A resposta da partícula que obteve o menor valor da função objetivo é destacada. Os parâmetros do controlador PI^λ resultantes do processo de otimização são apresentados na Tabela 16.

As evoluções dos parâmetros da melhor partícula g^{best} são ilustradas na Figura 53. Observa-se a convergência das partículas nas três dimensões, como esperado no algoritmo PSO.

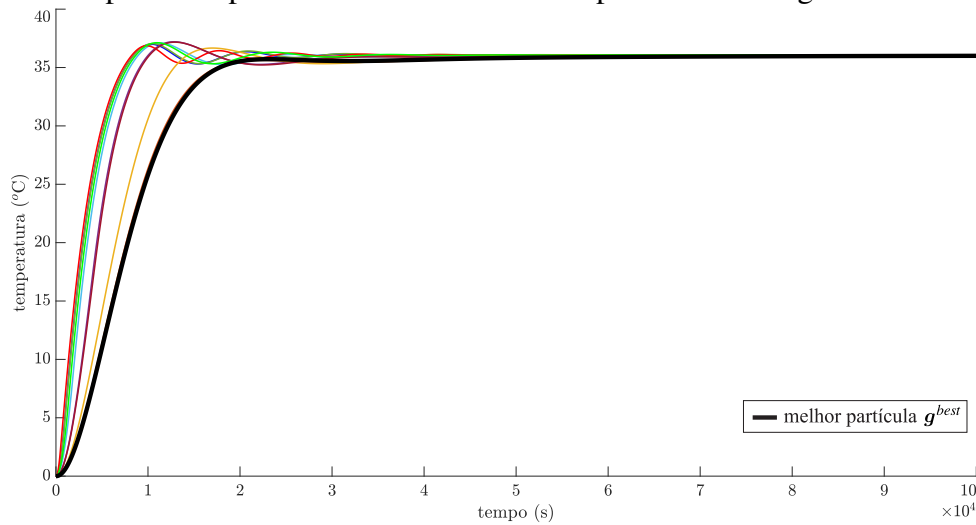
Figura 53 – Evoluções dos parâmetros da melhor partícula g^{best} .



Fonte: elaborada pelo autor.

A evolução do valor crítico de k_i , k_{i_c} para a partícula g^{best} é ilustrada na Figura 53.b. Percebe-se que durante todas as iterações, o valor de k_i da partícula g^{best} se manteve sempre abaixo de seu valor crítico, o que significa que, durante toda a execução do algoritmo, a partícula g^{best} esteve sempre na região estável. O ponto de operação do sistema de controle fracionário resultante é ilustrado na Figura 53.d, em que pode ser observado que o mesmo encontra-se longe da superfície crítica. As respostas ao degrau exibidas na Figura 52.a são ilustradas na Figura 54 com maior tempo de execução.

Figura 54 – Resposta temporal do sistema de controle para 100.000 segundos de simulação.



Fonte: elaborada pelo autor.

Tabela 17 – Resultados obtidos para o sistema de controle fracionário proposto.

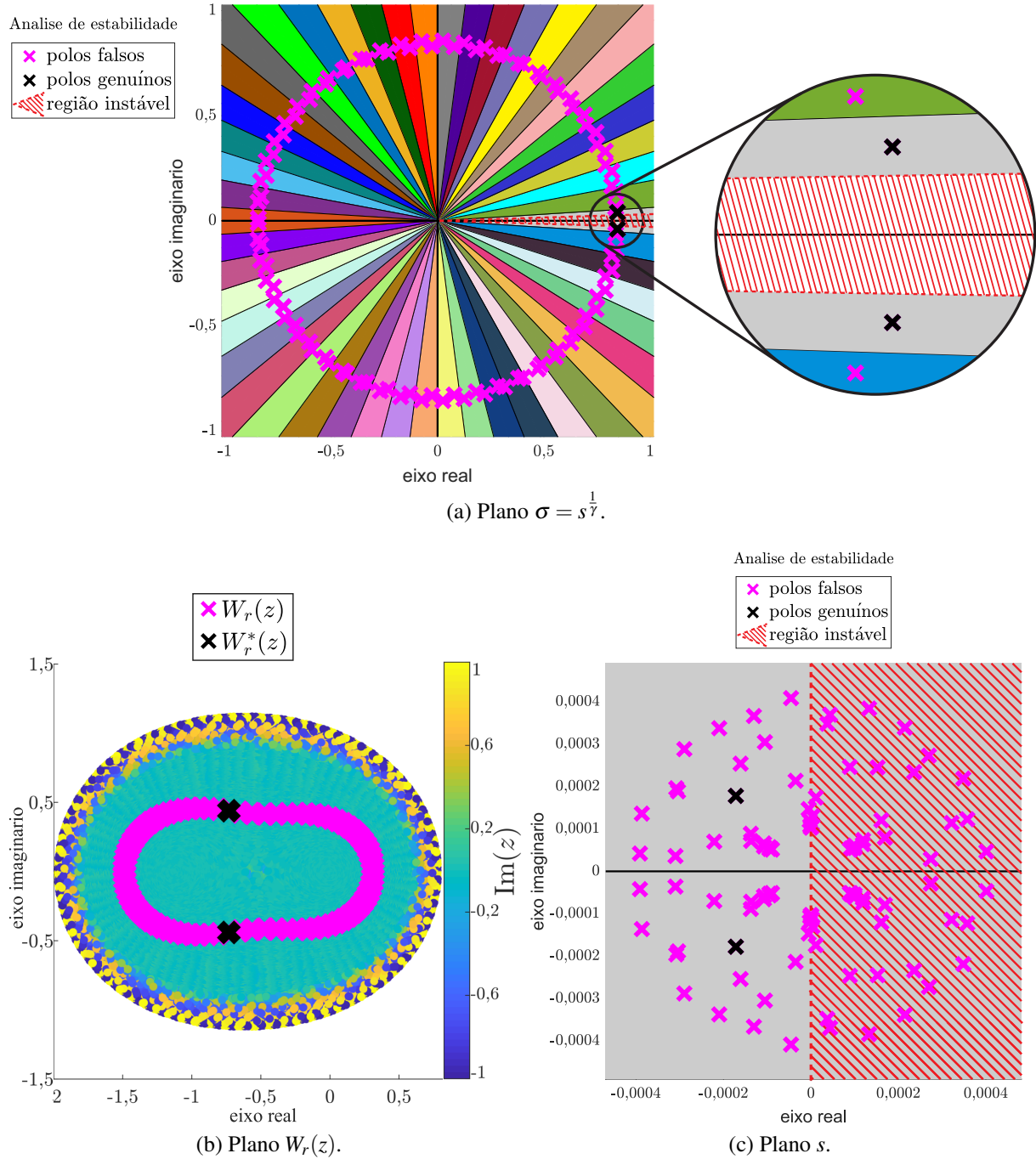
Lambert-Tsallis		teorema de Matignon	
$W_r^*(z)$	s	$\sigma = s^{1/50}$	s
$-0,7315 + 0,4398i$	$(-0,1755 + 0,1779i) \cdot 10^{-3}$	$0,8460 + 0,0396i$	$(-0,1755 + 0,1779i) \cdot 10^{-3}$
$-0,7315 - 0,4398i$	$(-0,1755 - 0,1779i) \cdot 10^{-3}$	$0,8460 - 0,0396i$	$(-0,1755 - 0,1779i) \cdot 10^{-3}$

Aplicando os parâmetros ótimos do controlador PI^λ presentes na Tabela 16 na Equação 8.13, a equação de transferência de malha fechada do sistema de controle fracionário é dada por

$$G_{cl}(s) = \frac{9,0938 \cdot 10^{-8} k_p s^{0,97532} + 2,2578 \cdot 10^{-8}}{s^{2,07532} + 1,4185 \cdot 10^{-4} s^{0,97532} + 2,2578 \cdot 10^{-8}}. \quad (8.13)$$

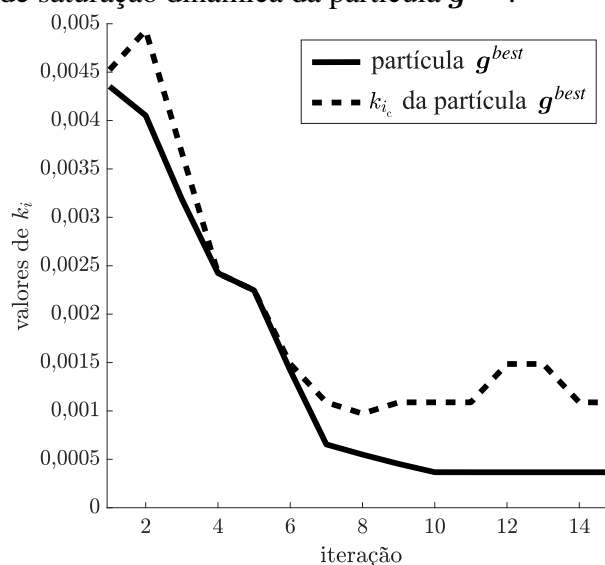
em que pelo princípio da identidade de polinômios entre a equação característica e a Equação 4.1, conclui-se que $a_n = 1$, $a_m = 1,4185 \cdot 10^{-4}$, $a_0 = 2,2578 \cdot 10^{-8}$, $m = 0,97532$ e $n = 2,07532$. Considerando $\lambda = 0,975$ e aplicando a Equação 4.6 e a Equação 4.10, resulta que $r = 1,128205$, e $z = -0,3795 - 0,1617i$. A partir destes dois parâmetros obtidos, r e z , os valores de $W_r(z)$ são obtidos através da aplicação do Algoritmo 1. Os valores numéricos de $W_r^*(z)$, ou seja, os valores de $W_r(z)$ que levam a raízes genuínas de s (obtidos através do Algoritmo 2), de σ e s genuínos são apresentadas na Tabela 17. O plano $\sigma = s^{\frac{1}{\gamma}}$ com $\gamma = 50$, o plano $W_r(z)$ e o plano s do sistema de controle fracionário em questão são ilustrados na Figura 55.

Figura 55 – Planos s e σ para o sistema de controle fracionário proposto.



Fonte: elaborada pelo autor.

Figura 56 – Exemplo de saturação dinâmica da partícula g^{best} .



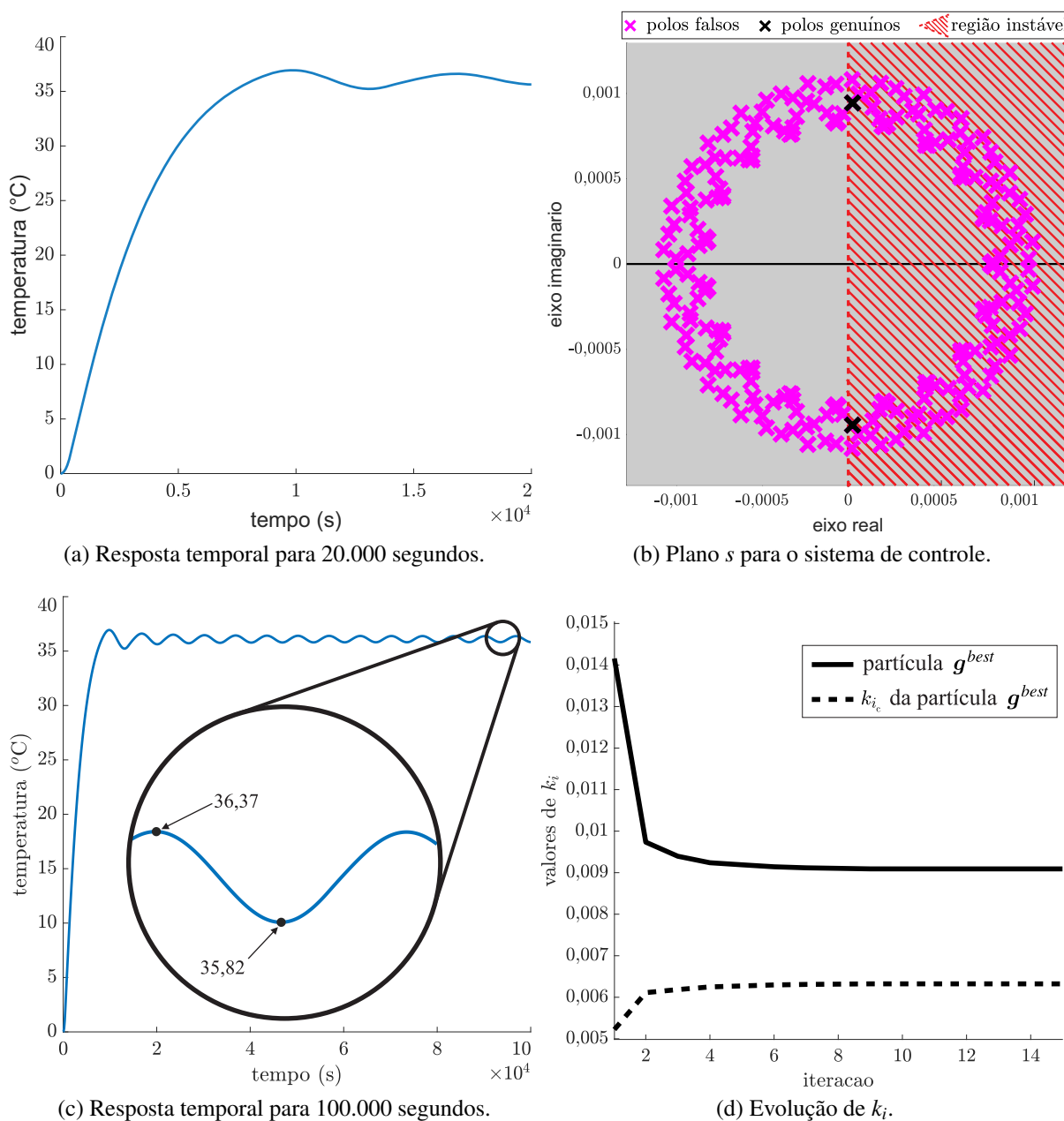
Fonte: elaborada pelo autor.

Como dito anteriormente, a partícula g^{best} em nenhum momento cruza a superfície de estabilidade, como visto na Figura 53.b. Um exemplo da saturação dinâmica de k_i foi observado durante execuções preliminares do algoritmo PSO-LT, e é ilustrado na Figura 56. Observa-se que, durante as iterações 4 e 5, a partícula g^{best} cruza a superfície, sendo saturada na mesma pelo algoritmo PSO-LT. Após a saturação, a partícula volta a realizar a busca na região de estabilidade, mostrando a efetividade do algoritmo PSO-LT proposto.

A efetividade do algoritmo PSO-LT também pode ser constatada no exemplo a seguir. Durante execuções preliminares do algoritmo PSO, sem a utilização da saturação dinâmica, observou-se um caso em que a partícula vencedora resultou em um sistema de controle instável. A resposta ao degrau durante o tempo de simulação escolhido para o cálculo da função objetivo é ilustrado na Figura 57.a, enquanto que o plano s da é ilustrada na Figura 57.b. Observa-se que os polos genuínos estão do lado direito do plano s , caracterizando o sistema como instável. A resposta ao degrau do sistema, aparentemente subamortecida, na verdade é resultado da saturação da saída de controle, imposta para manter o controlador nos níveis fisicamente realizáveis. A resposta do sistema para um maior tempo de simulação é ilustrada na Figura 57.c.

Os valores de k_i da partícula g^{best} durante a execução do algoritmo PSO são apresentados na Figura 57.d. Observa-se que desde a primeira iteração, o valor de k_i da partícula g^{best} encontra-se além do valor máximo permitido para a estabilidade, ou seja, desde a primeira iteração, a partícula encontra-se além da superfície crítica. A posição da partícula seria corrigida na primeira iteração caso a saturação dinâmica proposta não algoritmo PSO-LT estivesse sendo aplicada, como ilustrado na Figura 56.

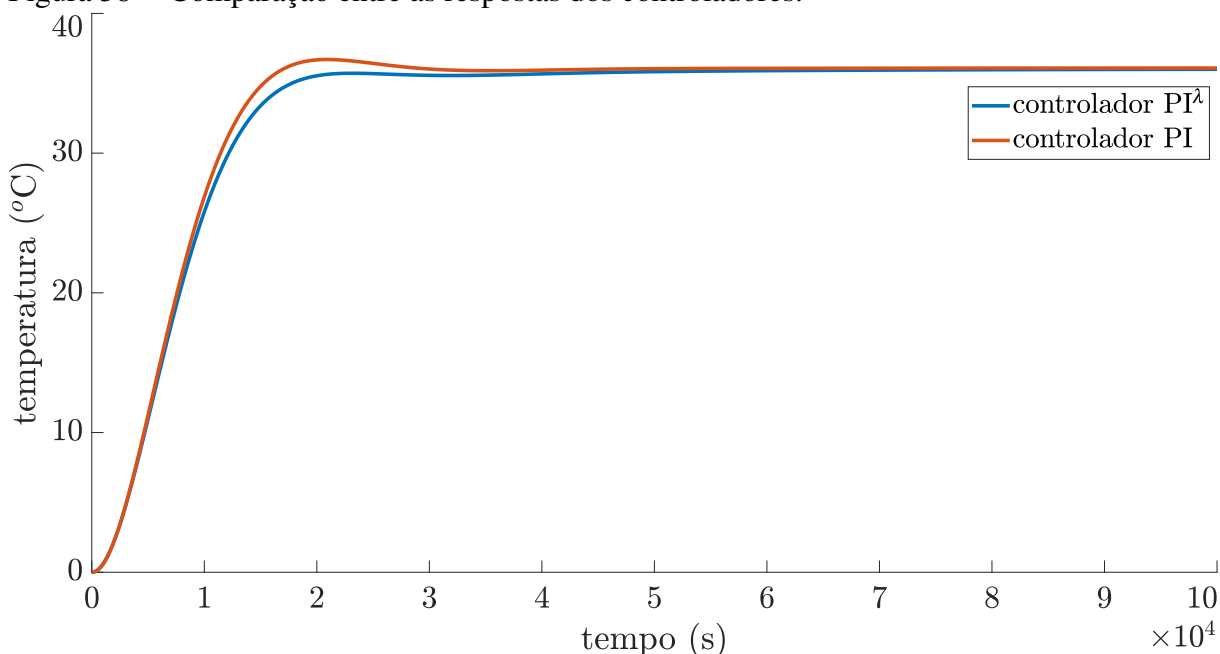
Figura 57 – Exemplo de resultados instáveis com a utilização do algoritmo PSO.



Fonte: elaborada pelo autor.

Este caso expõe a necessidade do projetista conhecer bem a capacidade do algoritmo PSO e da função objetivo utilizada. Observando a [Figura 57.a](#), nota-se que o tempo de simulação pode parecer curto, e que um tempo maior de simulação poderia contornar esse tipo de problema. Porém, um maior tempo de simulação implica em um maior tempo de execução do algoritmo PSO, visto que a resposta ao degrau deve ser calculada para cada partícula em cada iteração e em cada realização. Para os hiperparâmetros utilizados, a resposta ao degrau é calculada em um total de 1800 vezes.

Figura 58 – Comparação entre as respostas dos controladores.



Fonte: elaborada pelo autor.

Tabela 18 – Comparação do desempenho dos controladores PI^λ e PI.

controlador	parâmetros			tempo de acomodação	sobressinal
	k_p	k_i	λ		
PI^λ	0,0045647	0,0011333	0,97532	19.100 s	0%
PI	0,0018362	0,0009608	1	15.806 s	1,6%

Fonte: elaborada pelo autor.

Uma possível solução que não utiliza o algoritmo PSO-LT proposto é a adoção de termos na função objetivo que penalizem comportamentos oscilatórios, calculados, por exemplo, a partir do comprimento de arco da resposta do sistema de controle. Entretanto, considera-se a aplicação do algoritmo PSO-LT, com saturação dinâmica, suficiente e satisfatória, em face dos resultados apresentados nas Figuras 52 a 55.

Para fins de comparação com os resultados do controlador PI^λ , um controlador PI foi sintonizado pelo algoritmo PSO com os mesmos hiperparâmetros previamente descritos, para os quais o controlador PI^λ obteve melhor resultado. A combinação deste controlador PI com a planta de ordem fracionária identificada posiciona este sistema de controle no caso 4 da Tabela 6. As respostas temporais dos dois sistemas de controle são ilustradas na Figura 58, enquanto que a Tabela 18 apresenta os parâmetros dos controladores, o tempo de acomodação e o sobressinal das respostas. Nota-se que, embora a utilização do controlador PI^λ resulte em um tempo de acomodação superior em relação ao uso do controlador PI, o mesmo não apresenta sobressinal, enquanto que o uso do controlador PI proporciona sobressinal de 1,6%.

Tabela 19 – Parâmetros da discretização de Merrikh-Bayat.

w_0	w_1	w_2	w_3	w_4	w_5	w_6
0,056015	0,044346	-0,002477	-0,000785	-0,000806	-0,000467	-0,000509

Fonte: elaborada pelo autor.

8.3 Implementação embarcada do controlador PI^λ

O controlador PI^λ sintonizado na seção anterior é implementado conforme descrito na [Subseção 7.2.2](#). O resultado é comparado com a técnica de Merrikh-Bayat para discretização de controladores fracionários (vide [Subseção 7.2.1](#)). Para isso, o controlador PI^λ é discretizado para $T = 100$ s, a mesma taxa de amostragem utilizada para identificação do sistema na [Seção 8.1](#).

A partir dos parâmetros do controlador PI^λ apresentados na [Tabela 16](#), a discretização de Merrikh-Bayat para $M = 5$ e $T = 100$ s resulta em 6 parâmetros, descritos na [Tabela 19](#).

Em relação à implementação via rede neural, os dados utilizados para treinamento da MLP são obtidos através do diagrama ilustrado na [Figura 44](#). Para realizar uma comparação justa entre os dois métodos, são consideradas 7 entradas na rede neural, a mesma quantidade utilizada no método de Merrikh-Bayat. Com base em uma análise preliminar, os hiperparâmetros utilizados no treinamento da rede neural são descritos a seguir.

A MLP possui uma camada oculta, cujos neurônios utilizam a tangente hiperbólica como função de ativação. A camada de saída é composta por apenas um neurônio linear. Os dados são divididos em 80% para treinamento e 20% para teste. A rede foi treinada utilizando o algoritmo de retropropagação do erro apresentado no [Apêndice B](#). Heurísticas e testes sucessivos revelaram que a melhor opção é o uso de uma taxa de aprendizado com decaimento linear, com valor inicial de 0,5 e valor final de 0,001, juntamente com 600 épocas de treinamento. Os dados de entrada e saída são normalizados para o intervalo $[-1, +1]$. Os parâmetros das normalizações são armazenados para utilização posterior na etapa de implementação no sistema embarcado.

Para determinar o número ótimo de neurônios na camada oculta, foram realizadas 20 rodadas de treinamento para diferentes quantidades de neurônios, em que os índices descritos na [Subsubseção 7.2.2.1](#) foram analisados. Os resultados para o conjunto de teste são apresentados na [Tabela 20](#). Todas as configurações testadas apresentam baixos valores de MBE e RMSE, e elevados coeficientes de correlação e determinação, evidenciando a confiabilidade do treinamento. A configuração com 4 neurônios na camada oculta é escolhida como a melhor, por apresentar os melhores índices.

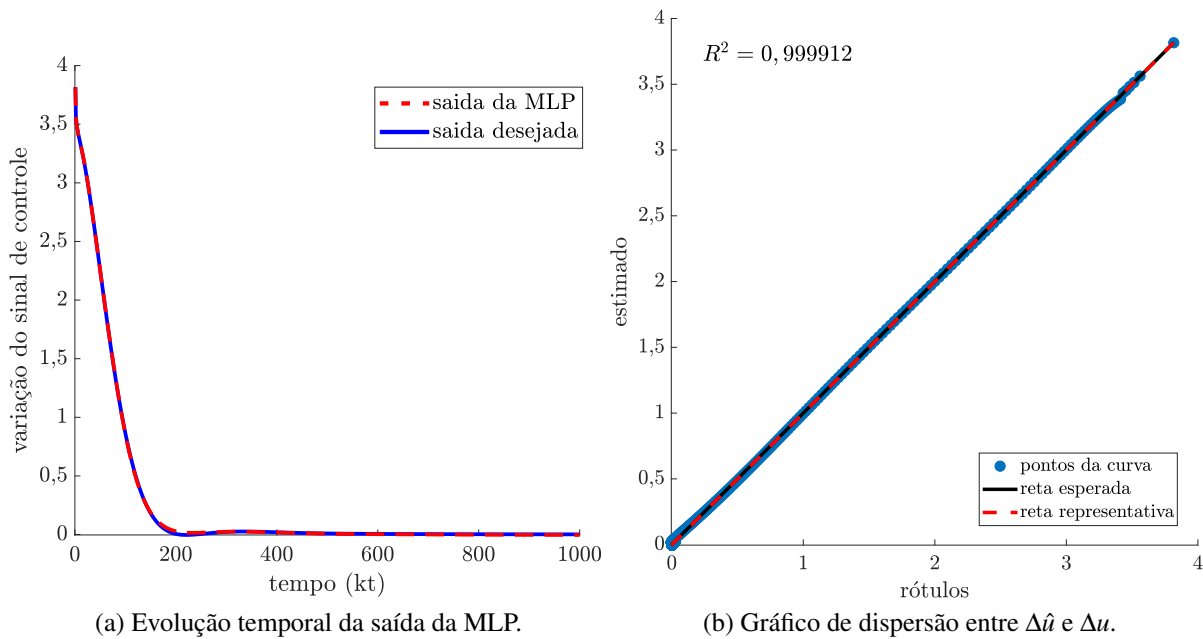
A saída da MLP, isto é, a variação estimada do sinal de controle $\Delta \hat{u}$ é apresentada

Tabela 20 – Desempenho da rede neural para diferentes neurônios na camada oculta.

neurônios ocultos	melhor			média
	R^2	MBE	RMSE	$R \pm \sigma_R$
3	0,9998	$4,76 \cdot 10^{-4}$	$3,26 \cdot 10^{-3}$	$0,9923 \pm 0,0118$
4	0,9999	$2,78 \cdot 10^{-4}$	$2,57 \cdot 10^{-3}$	$0,9966 \pm 0,0118$
5	0,9998	$6,59 \cdot 10^{-4}$	$3,34 \cdot 10^{-3}$	$0,9953 \pm 0,0117$
6	0,9998	$-6,44 \cdot 10^{-3}$	$3,12 \cdot 10^{-3}$	$0,9948 \pm 0,0092$
7	0,9998	$-1,07 \cdot 10^{-3}$	$2,71 \cdot 10^{-3}$	$0,9960 \pm 0,0085$

Fonte: elaborada pelo autor.

Figura 59 – Comparação entre a saída da rede neural e a saída desejada.



Fonte: elaborada pelo autor.

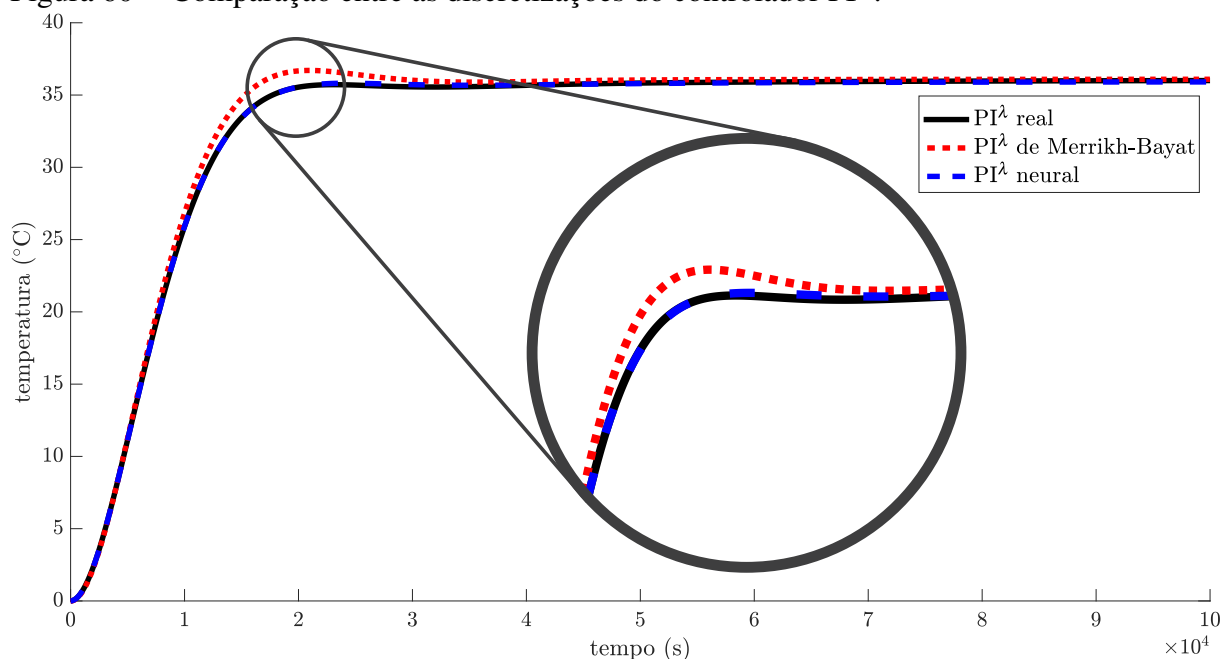
na [Figura 59.a](#), enquanto que o gráfico de dispersão entre os valores estimados pela MLP e os valores reais de Δu é ilustrada na [Figura 59.b](#).

Os resultados obtidos corroboram a hipótese de que uma rede MLP pode aprender o comportamento do controlador PI^λ , um filtro de ordem infinita, com elevada precisão a partir de poucos sinais de entrada.

A comparação entre as respostas simuladas dos sistemas de controle com o controlador PI^λ exato, o controlador PI^λ neural e o controlador PI^λ com aproximação de Merrikh-Bayat, é apresentada na [Figura 60](#). A saída do sistema resultante do controlador PI^λ neural é a mais próxima da saída do sistema com o controlador PI^λ exato, apresentando um sobressinal menor. De fato, o RMSE do controlador PI^λ neural é 0,0715, enquanto o RMSE da aproximação de Merrikh-Bayat é 0,5032.

Este resultado evidencia como verdadeira a hipótese levantada na [Seção 7.2](#) sobre a

Figura 60 – Comparação entre as discretizações do controlador PI^λ .



Fonte: elaborada pelo autor.

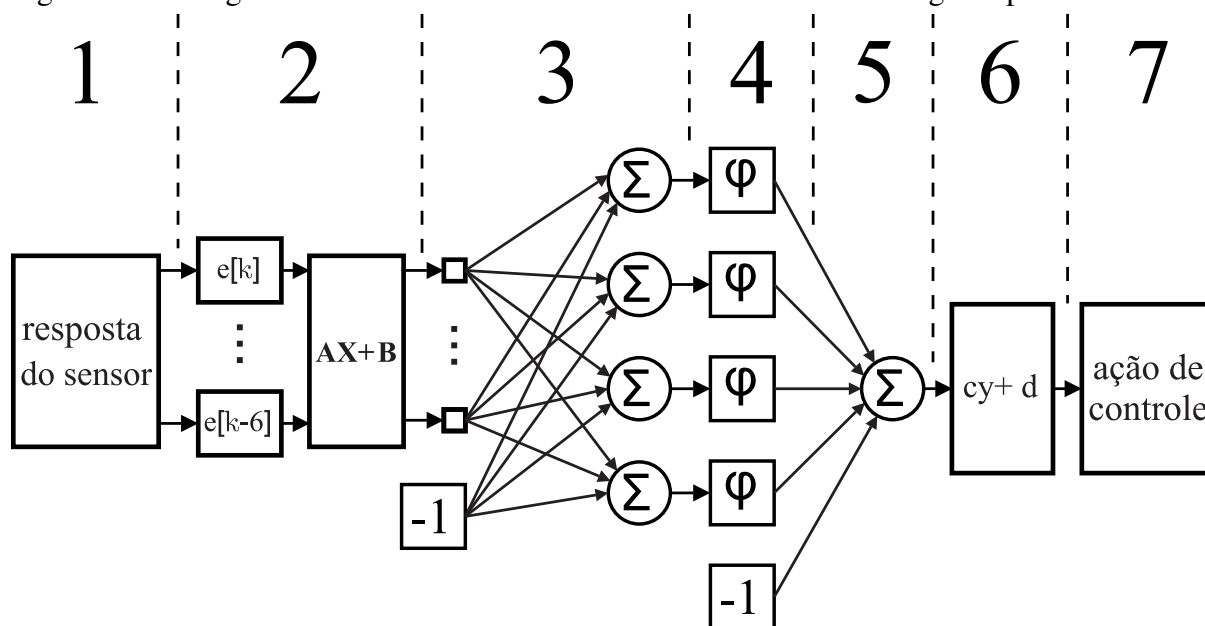
possibilidade de se obter um modelo computacional com poucas entradas ser capaz de reproduzir com aceitável acurácia o comportamento de um controlador fracionário, que por definição, possui memória infinita.

Para avaliar o desempenho prático do controlador PI^λ neural, a rede MLP foi embarcada no mesmo microcontrolador utilizado no *retrofit* e na aquisição de dados (vide [Apêndice C](#)). As medições de tempo de processamento e de uso de memória realizadas para cada estágio do sistema embarcado ilustrado na [Figura 61](#) são medidos com o auxílio de um osciloscópio e são apresentadas na [Tabela 21](#). O tempo total de processamento no sistema embarcado é de $7384,54 \mu s$ (7,4 s). Os estágios com maior tempo de processamento são a atualização e normalização das entradas e o cálculo das funções de ativação da camada oculta.

As respostas temporais dos sistemas embarcados com os controladores PI^λ neural e pela aproximação de Merrikh-Bayat são apresentadas na [Figura 62](#). A resposta real do sistema de controle apresentou um sobressinal de 1,66% e um tempo de acomodação de 8.183 s para o controlador PI^λ neural. Para a aproximação de Merrikh-Bayat, foi obtido um sobressinal de 3,68% e um tempo de acomodação de 20.120 s.

Os resultados reais dos dois controladores embarcados presentes na [Figura 62](#) confirmam os resultados de simulação na [Figura 60](#), de modo que o sistema de controle com o controlador PI^λ de Merrikh-Bayat apresenta maior sobressinal na resposta transitória em

Figura 61 – Estágios do sistema embarcado no microcontrolador ATmega328p.



Fonte: elaborada pelo autor.

Tabela 21 – Desempenho computacional do sistema embarcado.

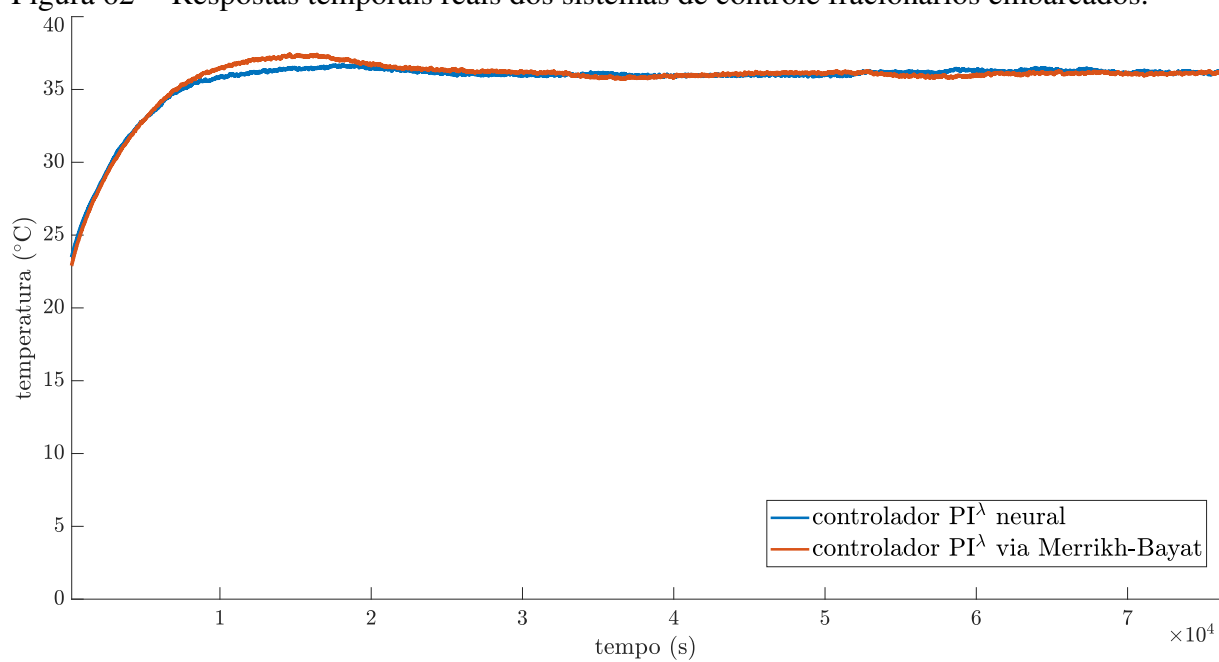
	Estágio	tempo de execução (μs)	memória de programa (bytes)	memória RAM (bytes)
1	resposta do sensor	834	998 (3,25%)	32 (1,56%)
2	atualização da entrada & normalização dos dados	3496	28 (0,09%)	8 (0,39%)
3	somatórios da camada oculta	874,5	858 (2,79%)	41 (2,0%)
4	funções de ativação da camada oculta	2000	122 (0,40%)	16 (0,78%)
5	somatório da camada de saída	123,5	258 (0,84%)	144 (7,03%)
6	desnormalização de dados	16,94	290 (0,94%)	32 (1,56%)
7	atualização da saída de controle	39,6	2932 (9,54%)	62 (3,03%)
total		7384,54	5486 (17,86%)	335 (16,36%)

Fonte: elaborada pelo autor.

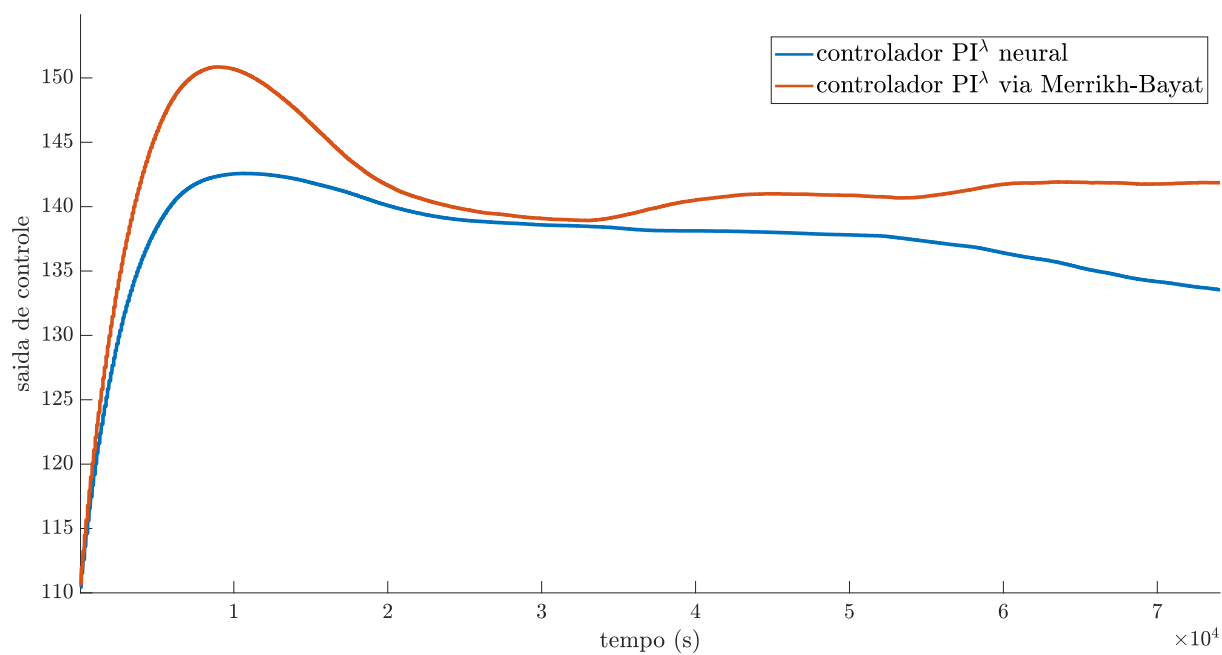
comparação com o controlador PI^{λ} neural.

Por fim, com base na Figura 62 e na Figura 60, observa-se que o controlador fracionário embarcado por meio de uma rede neural MLP apresenta grande proximidade com o desempenho do controlador fracionário desejado.

Figura 62 – Respostas temporais reais dos sistemas de controle fracionários embarcados.



(a) Saída do sistema.



(b) Sinal de controle.

Fonte: elaborada pelo autor.

9 CONCLUSÕES

A presente tese teve como objetivo principal estabelecer uma nova metodologia de síntese de controladores de ordem fracionária, de modo a garantir, de forma analítica e *a priori*, a estabilidade de sistemas de controle de ordem fracionária com equações características trinômiais generalizadas. Para isso, foram definidos objetivos específicos que abrangeram desde o desenvolvimento de fundamentos teóricos, como a análise de estabilidade baseada na função de Lambert-Tsallis, até aspectos práticos, como a simulação computacional, a identificação de sistemas fracionários, e a síntese de controladores via sintonia por otimização e a implementação embarcada eficiente com auxílio de redes neurais. Este capítulo apresenta as principais conclusões obtidas com o trabalho desenvolvido, discutindo suas contribuições, limitações e possibilidades para trabalhos futuras.

9.1 Contribuições da Tese

A análise de estabilidade baseada na função $W_r(z)$, apresentada na [Seção 4.2](#), evidenciou que o método proposto é mais simples e direto, conforme ilustrado na [Figura 21](#). Isso se deve ao fato da solução da equação característica ser expressa analiticamente em função de seus coeficientes e expoentes. Além disso, a extensão da abordagem para o cálculo *a priori* dos limites admissíveis dos parâmetros do controlador teve sua validade confirmada por um método *a posteriori* amplamente reconhecido na literatura, o critério proposto por [Matignon \(1996\)](#).

A região de estabilidade dos parâmetros obtida a partir dessa análise, representada por um volume tridimensional (vide [Figura 27](#) e [Figura 50](#)), mostrou-se bastante útil nos casos estudados. A integração dessa região de estabilidade *a priori* com o algoritmo de otimização metaheurística PSO resultou no algoritmo PSO-LT, que assegura que as partículas geradas conduzam exclusivamente a sistemas de controle estáveis. Tal comportamento é evidenciado nas [Figuras 52 - 57](#).

O método SISFRAC para simulação computacional de sistemas de ordem fracionária (sejam plantas ou controladores), apresentou vantagens significativas em relação a *toolboxes* tradicionais da literatura. Mantendo a precisão matemática rigorosa da definição de Grünwald-Letnikov para operadores fracionários, o método proporcionou uma redução expressiva no tempo de simulação, como demonstrado na [Figura 31](#), [Tabela 9](#) e [Tabela 10](#).

Quanto à implementação embarcada, a proposta de regressão não linear por meio de

uma rede neural MLP apresentou desempenho satisfatório. A MLP foi capaz de reproduzir com fidelidade o comportamento dinâmico do controlador fracionário, superando a abordagem de Merrikh-Bayat em termos de precisão. Notavelmente, uma simples arquitetura com apenas uma camada oculta composta por quatro neurônios e um neurônio linear na saída foi suficiente para alcançar esse resultado, com apresentado nas Figuras 59 - 62 e nas Tabelas 20 e 21.

Já o algoritmo proposto para identificação de sistemas de ordem fracionária apresentou resultados satisfatórios tanto em sistemas puramente matemáticos quanto em exemplos extraídos da literatura e no estudo de caso da incubadora neonatal. A hipótese levantada na Seção 6.1 — de que o algoritmo LMS seria capaz de identificar diretamente os parâmetros de um modelo de ordem fracionária, sem recorrer a aproximações que introduzem imprecisões — foi confirmada. A partir de um algoritmo composto apenas por operações simples, obteve-se uma identificação precisa, conforme verificado nas Figura 36, 38, 40 e 47, e nas Tabela 12, 13, 14 e 15.

Os resultados obtidos ao longo da tese, por meio de exemplos matemáticos, plantas da literatura e do estudo de caso da incubadora neonatal, demonstram que as propostas apresentadas são eficazes. As contribuições desenvolvidas, desde a formulação analítica da estabilidade via a função $W_r(z)$ até a implementação embarcada eficiente de controladores com auxílio redes neurais, representam um avanço significativo na síntese de controladores de ordem fracionária. A integração entre métodos analíticos, técnicas de simulação eficientes, estratégias de identificação diretas e aplicação prática comprova a viabilidade e a eficácia do paradigma proposto. Os resultados validados em diferentes cenários reforçam o potencial da abordagem desenvolvida como uma alternativa promissora para projetos de controle fracionário em contextos reais.

A análise de estabilidade baseada na função $W_r(z)$ restringe-se, em sua formulação atual, a sistemas de controle fracionário cuja equação característica assume a forma de um trinômio. Ainda que essa condição restrinja o escopo da metodologia, é importante destacar que ela abrange uma gama expressiva de configurações práticas entre planta e controlador, como ilustrado pelos 14 casos apresentados na Tabela 6.

Esses casos contemplam situações em que se combinam: i) plantas de ordem inteira com controladores de ordem fracionária; ii) plantas de ordem fracionária com controladores de ordem inteira; e iii) plantas e controladores ambos de ordem fracionária. Os arranjos compostos exclusivamente por plantas e controladores de ordem inteira não foram considerados, por não estarem no escopo desta tese.

9.2 Trabalhos Futuros

Como trabalhos futuros, propõem-se diferentes linhas de pesquisa com potencial para expandir o desenvolvimento de controladores de ordem fracionária:

- **Síntese de controladores fracionários por alocação de polos:** uma possível linha de pesquisa consiste em explorar a síntese de controladores fracionários via alocação de polos ([RASOULI et al., 2015](#)), sob a ótica da função Lambert-Tsallis, posicionando os polos diretamente sobre a superfície de Riemann associada ao plano s . Essa abordagem permitiria projetar controladores capazes de atender a requisitos de desempenho previamente especificados, como sobressinal e tempo de acomodação.
- **Estudo e transformação de equações quadrimiais fracionárias:** trabalhos futuros podem explorar a transformação de equações quadrimiais de ordem fracionária em formas trinômiais equivalentes, como em [Bluher \(2004\)](#). Essa abordagem ampliaria as possibilidades descritas na [Tabela 6](#), expandindo o conjunto de estruturas de controle compatíveis com a análise baseada na função de Lambert-Tsallis. A adaptação dessas técnicas ao contexto fracionário poderia abrir novas possibilidades para a síntese de controladores.
- **Síntese de controladores espaço de estados fracionários:** estender a análise de estabilidade de sistemas de controle de ordem fracionária pela função Lambert-Tsallis para a representação espaço de estados fracionários ([RAYNAUD; ZERGAÏNOH, 2000](#); [SABATIER, 2020](#)).
- **Controle distribuído de sistemas fracionários:** uma possível direção para trabalhos futuros é o desenvolvimento de estratégias de controle distribuído aplicadas a sistemas de controle de ordem fracionária ([GHANBARI; RAZZAGHI, 2022](#); [CHI et al., 2022](#)) visto a crescente adoção de arquiteturas em nuvem para aplicações industriais e laboratoriais.

REFERÊNCIAS

- ABBAS, S.; NAZAR, M.; GILLANI, S. F. F.; NAVEED, M.; AHMAD, M.; NISA, Z. U. A CPC fractional model of the heat and mass transport mechanism in carbon nanotubes with slip effects on velocity. **Modern Physics Letters B**, v. 38, n. 13, p. 2450100, 2024. Citado na página 25.
- ABURAKHIS, M.; ORDÓÑEZ, R.; DJANEYE-BOUNDJOU, O. A discrete fractional order adaptive law for parameter estimation and adaptive control. **IEEE Open Journal of Control Systems**, v. 1, p. 113–125, 2022. Citado na página 90.
- AGHABABA, M. P. Optimal design of fractional-order PID controller for five bar linkage robot using a new particle swarm optimization algorithm. **Soft Computing**, Springer Science and Business Media LLC, v. 20, n. 10, p. 4055–4067, 2015. Citado na página 129.
- AI, W.; LIN, X.; LUO, Y.; WANG, X. Fractional-order modeling and identification for an SCR denitrification process. **Fractal and Fractional**, MDPI AG, v. 8, n. 9, p. 524, 2024. Citado na página 25.
- ALMEIDA, F. J. L. de; RAMOS, R. V. Disentropy in astronomy. **The European Physical Journal Plus**, Springer Berlin Heidelberg, v. 138, n. 1, p. 20, 2023. Citado na página 55.
- AMADI, H.; ADESINA, C.; OLATEJU, E.; OMOKARO, S.; OKECHUKWU, A.; ALABI, P.; HARUNA, A. Validation of a novel technique that minimises early neonatal deaths—a comparative study. **J Pediatr Neonatal Care**, v. 6, n. 1, p. 00232, 2017. Citado na página 139.
- ANDRADE, J. S.; NOBREGA, K. Z.; RAMOS, R. V. Analytical solution of the current-voltage characteristics of circuits with power-law dependence of the current on the applied voltage using the Lambert-Tsallis W_q function. **IEEE Transactions on Circuits and Systems II: Express Briefs**, IEEE, v. 69, n. 3, p. 769–773, 2022. Citado na página 55.
- ASGHARNIA, A.; JAMALI, A.; SHAHNAZI, R.; MAHERI, A. Load mitigation of a class of 5-MW wind turbine with RBF neural network based fractional-order PID controller. **ISA Transactions**, Elsevier BV, v. 96, p. 272–286, 2020. Citado na página 133.
- BARBEAU, E. J. **Polynomials**. New York: Springer, 2013. (Problem Books in Mathematics). ISBN 0387406271; 9780387406275. Citado na página 41.
- BELKIĆ, D. All the trinomial roots, their powers and logarithms from the Lambert series, Bell polynomials and Fox–Wright function: illustration for genome multiplicity in survival of irradiated cells. **Journal of Mathematical Chemistry**, Springer, v. 57, p. 59–106, 2019. Citado na página 70.
- BENFTIMA, S.; GHARAB, S.; BATLLE, V. F. Fractional modeling and control of lightweight 1 DOF flexible robots robust to sensor disturbances and payload changes. **Fractal and Fractional**, v. 7, n. 7, 2023. Citado na página 40.
- BIN, F.; FUMING, P. Some results of finding the roots of time-delay systems by Lambert W function. In: **Proceedings of the 2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA'2021)**. Shenyang, China: IEEE, 2021. p. 199–204. Citado na página 54.
- BLUHER, A. W. On $xq+1+ax+b$. **Finite Fields and Their Applications**, Elsevier BV, v. 10, n. 3, p. 285–305, 2004. Citado na página 162.

- BOŠKOVIĆ, M. Č.; ŠEKARA, T. B.; LUTOVAC, B.; DAKOVIĆ, M.; MANDIĆ, P. D.; LAZAREVIĆ, M. P. Analysis of electrical circuits including fractional order elements. In: **IEEE. Proceedings of the 6th Mediterranean Conference on Embedded Computing (MECO'2017)**. Bar, Montenegro, 2017. p. 1–6. Citado na página 39.
- BROWN, J. W.; CHURCHILL, R. V. **Variáveis Complexas e Aplicações**. 9. ed. Porto Alegre: McGraw-Hill, 2015. ISBN 978-85-8055-517-2. Citado na página 43.
- CALASAN, M.; VUJOSEVIC, S.; MICEV, M.; ALRUWAILI, M.; WIJAYA, A. A. Triple-diode solar cell current optimization – an analytical solution based on the Lambert W function. **Alexandria Engineering Journal**, Elsevier BV, v. 104, p. 95–114, 2024. Citado na página 54.
- CAPUTO, M. Linear models of dissipation whose q is almost frequency independent—ii. **Geophysical Journal International**, v. 13, n. 5, p. 529–539, 11 1967. Citado na página 33.
- CARLSON, G.; HALIJAK, C. **Simulation of the fractional derivative operator \sqrt{s} and the fractional integral operator $1/\sqrt{s}$** . Tese (Doutorado) – Kansas State University, 1960. Citado na página 95.
- CARLSON, G.; HALIJAK, C. Approximation of fractional capacitors $(1/s)^{1/n}$ by a regular Newton process. **IEEE Transactions on Circuit Theory**, v. 11, n. 2, p. 210–213, 1964. Citado na página 95.
- CHEN, Y.; PETRAS, I.; XUE, D. Fractional order control - a tutorial. In: **2009 American Control Conference**. St. Louis, Missouri: IEEE, 2009. p. 1397–1411. Citado na página 95.
- CHI, C.; CAJO, R.; ZHAO, S.; LIU, G.-P.; IONESCU, C.-M. Fractional order distributed model predictive control of fast and strong interacting systems. **Fractal and Fractional**, MDPI AG, v. 6, n. 4, p. 179, 2022. Citado na página 162.
- COUTO, H. L.; AVELAR, A. T.; CARDOSO, W. B. Effective 1D-equation with a Lambert W function nonlinearity for cigar-shaped Bose–Einstein condensates. **Communications in Nonlinear Science and Numerical Simulation**, Elsevier, v. 140, p. 108447, 2025. Citado na página 54.
- DAS, A.; SIDHARTH, B. G.; ROBERTS, K.; VALLURI, S. R. Complex energies and the Lambert W function. **Theoretical and Mathematical Physics**, v. 211, n. 3, p. 817–825, 2022. Citado na página 65.
- DAS, S. **Functional fractional calculus**. Berlin: Springer, 2011. v. 1. Citado nas páginas 27 e 109.
- DASTJERDI, A. A.; VINAGRE, B. M.; CHEN, Y.; HOSSEINNIA, S. H. Linear fractional order controllers; a survey in the frequency domain. **Annual Reviews in Control**, Elsevier, v. 47, p. 51–70, 2019. ISSN 1367-5788. Citado na página 25.
- DAVIS, H. T. **The Theory of Linear Operators from the Standpoint of Differential Equations of Infinite Order**. Bloomington, Indiana: The Principia Press, 1936. 628 p. Citado na página 33.
- DIMITRAKAKIS, G. S.; GERMENIS, N. G.; SOKOS, E. N. Fractional order inverse filter for high sensitivity – low noise broadband seismometer. **IEEE Transactions on Instrumentation and Measurement**, Institute of Electrical and Electronics Engineers (IEEE), p. 1–1, 2025. Citado na página 25.

DJABALLAH, C. B.; NOUIBAT, W.; AYAD, R. Optimization of fractional-order PI-PID controllers for MIMO systems using artificial bee colony algorithm. **Soft Computing**, Springer Science and Business Media LLC, v. 28, n. 17–18, p. 10281–10299, 2024. Citado na página 128.

DUBINOV, A. E. Einstein-Smoluchowski-type relations for real gases. **Journal of Mathematical Chemistry**, v. 63, n. 4, p. 1116–1125, 2025. Citado na página 54.

EL-KHAZALI, R. Robust fractional-order lag/lead compensator. In: **Proceedings of the 61st IEEE International Midwest Symposium on Circuits and Systems (MWSCAS'2018)**. Wichita, Kansas: IEEE, 2018. p. 817–820. Citado na página 38.

EL-SHAFEI, M. A.; EL-HAWWARY, M. I.; EMARA, H. M. Implementation of fractional-order PID controller in an industrial distributed control system. In: **2017 14th International Multi-Conference on Systems, Signals & Devices (SSD)**. Marrakech, Morocco: IEEE, 2017. p. 713–718. Citado nas páginas 134 e 135.

EULER, L. De progressionibus transcendentibus seu quarum termini generales algebraice dari nequeunt. **Commentarii academiae scientiarum Petropolitanae**, p. 36–57, 1738. Citado na página 33.

EULER, L. De serie lambertina plurimisque eius insignibus proprietatibus. **Acta Academiae scientiarum imperialis petropolitanae**, p. 29–51, 1783. Citado na página 54.

FELIU-BATLLE, V.; RIVAS-PEREZ, R.; CASTILLO-GARCIA, F. J. Robust fractional-order temperature control of a steel slab reheating furnace with large time delay uncertainty. In: **IEEE. Proceedings of the 2014 International Conference on Fractional Differentiation and Its Applications (ICFDA'14)**. Catania, Italy, 2014. p. 1–6. Citado na página 38.

FELIU-TALEGON, D.; FELIU-BATLLE, V. Control of very lightweight 2-DOF single-link flexible robots robust to strain gauge sensor disturbances: A fractional-order approach. **IEEE Transactions on Control Systems Technology**, Institute of Electrical and Electronics Engineers (IEEE), v. 30, n. 1, p. 14–29, 2022. Citado na página 26.

FORTULAN, R.; KHEIRABADI, N. R.; RAEISI-KHEIRABADI, N.; NEZAMZADEH-EJHIEH, A.; CHIOLERIO, A.; ADAMATZKY, A. Fractional-order memristive dynamics in colloidal graphitic carbon nitride systems. **Phys. Rev. E**, American Physical Society, v. 110, p. 034607, Sep 2024. Citado na página 40.

FOURIER, J. B. J. **Théorie analytique de la chaleur**. Paris: Firmin Didot, 1822. Citado na página 33.

GAO, Q.; CHEN, J.; WANG, L.; XU, S.; HOU, Y. Multiobjective optimization design of a fractional order PID controller for a gun control system. **The Scientific World Journal**, Wiley, v. 2013, n. 1, 2013. Citado na página 128.

GAO, Z.; LIAO, X. Improved Oustaloup approximation of fractional-order operators using adaptive chaotic particle swarm optimization. **Journal of Systems Engineering and Electronics**, v. 23, n. 1, p. 145–153, 2012. Citado na página 95.

GHANBARI, G.; RAZZAGHI, M. Numerical solutions for distributed-order fractional optimal control problems by using generalized fractional-order chebyshev wavelets. **Nonlinear**

Dynamics, Springer Science and Business Media LLC, v. 108, n. 1, p. 265–277, 2022. Citado na página 162.

GLASSER, M. Hypergeometric functions and the trinomial equation. **Journal of Computational and Applied Mathematics**, Elsevier, v. 118, n. 1, p. 169–173, 2000. ISSN 0377-0427. Citado na página 70.

GOLMANKHANEH, A. K.; ALI, K.; YILMAZER, R.; KAABAR, M. Local fractal Fourier transform and applications. **Computational Methods for Differential Equations**, University of Tabriz, IR, n. Online First, 2021. Citado na página 25.

GROSS, B.; BRAGA, E. P. **Singularities of Linear System Functions**. New York: Elsevier Publishing Company, 1961. ISBN 9780444402639. Citado na página 43.

GRÜNWALD, A. K. Ueber begrenzte derivationen und deren anwendung. **Z. Math. Phys**, v. 12, n. 6, p. 441–480, 1867. Disponível em: https://gdz.sub.uni-goettingen.de/id/PPN599415665_0012. Citado na página 33.

GUEDES, V. F.; NOBREGA, K. Z.; RAMOS, R. V. Analytical solution of the space charge limited current using Lambert-Tsallis W_q function. **IEEE Transactions on Electron Devices**, IEEE, v. 69, n. 10, p. 5787–5791, 2022. Citado nas páginas 55 e 61.

HAN, B.; YIN, D.; AND, Y. G. The application of a novel variable-order fractional calculus on rheological model for viscoelastic materials. **Mechanics of Advanced Materials and Structures**, Taylor & Francis, v. 31, n. 28, p. 9951–9963, 2024. Citado na página 40.

HAYKIN, S. **Adaptive Filter Theory**. 5th. ed. Harlow: Pearson Education, 2013. Citado na página 110.

HEAVISIDE, O. On operators in physical mathematics. **Proceedings of the Royal Society of London**, The Royal Society London, v. 52, p. 504–529, 1893. Citado na página 33.

HOA, N. V.; PHU, N. D. Fuzzy discrete fractional calculus and fuzzy fractional discrete equations. **Fuzzy Sets and Systems**, v. 492, p. 109073, 2024. ISSN 0165-0114. Citado na página 25.

HORNIK, K.; STINCHCOMBE, M.; WHITE, H. Multilayer feedforward networks are universal approximators. **Neural Networks**, Elsevier BV, v. 2, n. 5, p. 359–366, 1989. Citado na página 133.

HUNG, H.-L.; HUANG, Y.-F.; YEH, C.-M.; TAN, T.-H. Performance of particle swarm optimization techniques on PAPR reduction for ofdm systems. In: **2008 IEEE International Conference on Systems, Man and Cybernetics**. Singapore: IEEE, 2008. p. 2390–2395. Citado na página 129.

JAMIL, A. A.; TU, W. F.; ALI, S. W.; TERRICHE, Y.; GUERRERO, J. M. Fractional-order PID controllers for temperature control: A review. **Energies**, MDPI, v. 15, n. 10, 2022. Citado nas páginas 26 e 38.

JIA, Z.; LIU, C. Fractional-order modeling and simulation of magnetic coupled boost converter in continuous conduction mode. **International Journal of Bifurcation and Chaos**, World Scientific, v. 28, n. 05, p. 1850061, 2018. Citado na página 40.

KAPOULEA, S.; TSIRIMOKOU, G.; PSYCHALINOS, C.; ELWAKIL, A. S. Employment of the Padé approximation for implementing fractional-order lead/lag compensators. **AEU - International Journal of Electronics and Communications**, v. 120, p. 153203, 2020. ISSN 1434-8411. Citado na página [39](#).

KENNEDY, J.; EBERHART, R. Particle swarm optimization. In: **Proceedings of ICNN'95 - International Conference on Neural Networks**. Perth, Australia: IEEE, 1995. (ICNN-95, v. 4), p. 1942–1948. Citado nas páginas [128](#) e [129](#).

KUMAR, V.; RANA, K. P. S. Comparative study on fractional order PID and PID controllers on noise suppression for manipulator trajectory control. In: _____. **Fractional Order Control and Synchronization of Chaotic Systems**. Cham: Springer International Publishing, 2017. p. 3–28. Citado nas páginas [26](#) e [38](#).

LACROIX, S. F. **Traité du calcul différentiel et du calcul intégral Tome 3**. Paris: Courcier, 1819. Citado na página [33](#).

LAGRANGE, J.-L. Sur une nouvelle espèce de calcul relatif à la différentiation et à l'intégration des quantités variables. **Nouveaux Mémoires de l'Académie royale des sciences et belles-lettres de Berlin**, v. 3, p. 441–476, 1772. Citado na página [33](#).

LAMBERT, J. H. Observationes variae in mathesin puram. **Acta Helvetica**, v. 3, n. 1, p. 128–168, 1758. Citado na página [54](#).

LAPLACE, P. S. **Théorie analytique des probabilités**. Paris: Courcier, 1812. Citado na página [33](#).

LAURENT, H. Sur le calcul des dérivées à indices quelconques. **Nouvelles annales de mathématiques: journal des candidats aux écoles polytechnique et normale**, v. 3, p. 240–252, 1884. Citado na página [33](#).

LEIBNIZ, G. W. Lettre de Leibniz à l'Hôpital du 30 septembre 1695. In: CAREIL, A. Foucher de (Ed.). **Correspondance Mathématique de Leibniz avec l'Hôpital**. Paris: Librairie de A. Franck, 1695. p. 297–302. Primenramente publicado em 1853. Citado na página [33](#).

LETNIKOV, A. V. Teoria da diferenciação com Índice arbitrário. **Matematicheskii Sbornik**, v. 3, n. 2, p. 85–112, 1868. Publicado originalmente em russo. Citado na página [33](#).

LI, Z.; LIU, L.; DEGHAN, S.; CHEN, Y.; XUE, D. A review and evaluation of numerical tools for fractional calculus and fractional order controls. **International Journal of Control**, Taylor & Francis, v. 90, n. 6, p. 1165–1181, 2017. Citado na página [96](#).

LIN, M.-S.; WU, J.-L.; ARUNKUMAR, A. Bounded real lemma for singular Caputo fractional-order systems. **IEEE Access**, Institute of Electrical and Electronics Engineers (IEEE), v. 12, p. 106303–106312, 2024. Citado na página [26](#).

LILOUVILLE, J. Mémoire sur le calcul des différentielles à indices quelconques. **Journal de l'École Polytechnique**, v. 13, p. 71–162, 1832. Citado nas páginas [33](#) e [34](#).

LIU, L.; ZHANG, S.; XUE, D.; CHEN, Y. Q. General robustness analysis and robust fractional-order PD controller design for fractional-order plants. **IET Control Theory & Applications**, Institution of Engineering and Technology (IET), v. 12, n. 12, p. 1730–1736, 2018. Citado na página [26](#).

LIU, X.; YE, G.; LIU, W.; GUO, Y.; SHI, F. On Atangana–Baleanu fractional granular calculus and its applications to fuzzy economic models in market equilibrium. **Journal of Computational and Applied Mathematics**, Elsevier, v. 450, p. 116017, 2024. ISSN 0377-0427. Citado na página 25.

LOPES, A. M.; MACHADO, J. A. T. A review of fractional order entropies. **Entropy**, MDPI AG, v. 22, n. 12, p. 1374, 2020. Citado na página 25.

MA, M.; YANG, J. Convergence analysis of novel fractional-order backpropagation neural networks with regularization terms. **IEEE Transactions on Cybernetics**, v. 54, n. 5, p. 3039–3050, 2024. Citado na página 25.

MANABE, S. The non-integer integral and its application to control systems(in japanese). **The Journal of the Institute of Electrical Engineers of Japan**, v. 80, n. 860, p. 589–597, 1960. Citado na página 37.

MATIGNON, D. Stability results for fractional differential equations with applications to control processing. In: **Proceedings of the 1996 Symposium on control, optimization and supervision, CESA'96 IMACS Multiconference, Computational Engineering in Systems Applications**. Lille, France: International Association for Mathematics and Computers in Simulation - IMACS, 1996. v. 2, n. 1, p. 963–968. Citado nas páginas 26, 41, 42, 47, 50, 52, 74, 76, 77, 82, 87, 145 e 160.

MATSUDA, K.; FUJII, H. H^∞ optimized wave-absorbing control - analytical and experimental results. **Journal of Guidance, Control, and Dynamics**, v. 16, n. 6, p. 1146–1153, 1993. Citado na página 95.

MERRIKH-BAYAT, F. Rules for selecting the parameters of oustaloup recursive approximation for the simulation of linear feedback systems containing $PI^\lambda D^\mu$ controller. **Communications in Nonlinear Science and Numerical Simulation**, v. 17, n. 4, p. 1852–1861, 2012. ISSN 1007-5704. Citado na página 95.

MERRIKH-BAYAT, F.; MIREBRAHIMI, N.; KHALILI, M. R. Discrete-time fractional-order PID controller: Definition, tuning, digital realization and some applications. **International Journal of Control, Automation and Systems**, Springer Science and Business Media LLC, v. 13, n. 1, p. 81–90, 2015. Citado nas páginas 25, 27, 28, 133, 134 e 135.

MIDYA, T.; GARAI, D.; DASGUPTA, T. A fast and accurate module for calculating fractional order derivatives and integrals in python. In: **2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)**. Bengaluru, India: IEEE, 2018. p. 1–6. Citado nas páginas 95 e 96.

MILES, P. **pyfod: a Python repository for performing fractional-order derivative operations**. San Francisco, CA: GitHub, 2019. <https://github.com/prmiles/pyfod>. Acessado em: 31 de janeiro de 2025. Citado na página 95.

MILLER, K. S.; ROSS, B. **An introduction to the fractional calculus and fractional differential equations**. New York: Wiley, 1993. (A Wiley-Interscience publication). Citado na página 33.

MONJE, C. A.; CHEN, Y.; VINAGRE, B. M.; XUE, D.; FELIU, V. **Fractional-order systems and controls: fundamentals and applications**. Berlin: Springer Science & Business Media, 2010. v. 1. Citado na página 34.

MOROZ, V.; BOROVETS, T. Using of fof toolbox and z-transform for fractional control systems. In: **2017 IEEE International Young Scientists Forum on Applied Physics and Engineering (YSF)**. Lviv, Ukraine: IEEE, 2017. p. 50–53. Citado na página 96.

MUNAGALA, V. K.; JATOTH, R. K. A novel approach for controlling DC motor speed using narxnet based fopid controller. **Evolving Systems**, Springer Science and Business Media LLC, v. 14, n. 1, p. 101–116, 2023. Citado na página 133.

NASSEF, A. M.; ABDELKAREEM, M. A.; MAGHRABIE, H. M.; BAROUTAJI, A. Metaheuristic-based algorithms for optimizing fractional-order controllers—a recent, systematic, and comprehensive review. **Fractal and Fractional**, MDPI AG, v. 7, n. 7, p. 553, 2023. Citado na página 128.

NISAR, K. S.; FARMAN, M. Investigation of fractional order model for glucose-insulin monitoring with PID and controllability. **Scientific Reports**, v. 15, n. 1, p. 8128, 2025. ISSN 2045-2322. Citado na página 40.

OLDHAM, K.; SPANIER, J. **The fractional calculus theory and applications of differentiation and integration to arbitrary order**. New York: Dover Publications, 2006. v. 1. ISBN 978-0-486-45001-8. Citado na página 33.

ONG, M. W. L.; NG, Z. C. A wavelet analysis of fractional calculus operators. **International Journal of Applied and Computational Mathematics**, Springer Science and Business Media LLC, v. 11, n. 2, 2025. Citado na página 25.

ONYEDI, T.; TEPLJAKOV, A.; PETLENKOV, E. Fomconpy: Fractional-order modelling and control library for python. In: **2020 43rd International Conference on Telecommunications and Signal Processing (TSP)**. Milão, Itália: IEEE, 2020. p. 239–245. Citado na página 95.

OPRZĘDKIEWICZ, K.; MITKOWSKI, W.; ROSÓŁ, M. Fractional order model of the two dimensional heat transfer process. **Energies**, MDPI AG, v. 14, n. 19, p. 6371, 2021. Citado na página 109.

OUSTALOUP, A. **La commande CRONE: commande robuste d'ordre non entier**. Paris: Hermes, 1991. ISBN 9782866012892. Citado na página 95.

OUSTALOUP, A.; LEVRON, F.; MATHIEU, B.; NANOT, F. Frequency-band complex noninteger differentiator: characterization and synthesis. **IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications**, v. 47, n. 1, p. 25–39, 2000. Citado nas páginas 27 e 95.

PERRY, A.; FENG, G.; LIU, Y.-F.; SEN, P. A design method for PI-like fuzzy logic controllers for DC–DC converter. **IEEE Transactions on Industrial Electronics**, Institute of Electrical and Electronics Engineers (IEEE), v. 54, n. 5, p. 2688–2696, 2007. Citado na página 133.

PETRÁŠ, I. **Fractional-order nonlinear systems: modeling, analysis and simulation**. Berlin: Springer Berlin, Heidelberg, 2011. v. 1. (Nonlinear Physical Science, v. 1). ISBN 9783642181009. Citado nas páginas 39, 74, 109, 120, 122 e 140.

PETRÁŠ, I.; VINAGRE, B. Practical application of digital fractional-order controller to temperature control. **Acta Montanistica Slovaca**, Citeseer, v. 7, n. 2, p. 131–137, 2002. Citado na página 38.

PETRÁŠ, I. **Handbook of Fractional Calculus with Applications**. Berlim, Alemanha: Walter De Gruyter Inc., 2019. v. 6, Applications in Control. Citado na página 95.

PODLUBNY, I. Fractional-order systems and fractional-order controllers. **Institute of Experimental Physics, Slovak Academy of Sciences, Kosice**, v. 12, n. 3, p. 1–18, 1994. Citado nas páginas 25 e 38.

PODLUBNY, I. Numerical methods of the fractional calculus. **Transactions of the Technical University of Kosice**, v. 4, n. 3-4, p. 200–208, 1994. Citado nas páginas 93 e 94.

PODLUBNY, I. **Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications**. Amsterdã, Países Baixos: Elsevier, 1998. v. 1. Citado nas páginas 34, 90, 93 e 94.

PODLUBNY, I. Geometric and physical interpretation of fractional integration and fractional differentiation. arXiv, 2001. Disponível em: <https://arxiv.org/abs/math/0110241>. Citado nas páginas 27 e 132.

POINOT, T.; TRIGEASSOU, J.; LIN, J. Parameter estimation of fractional models: Application to the modeling of diffusive systems. **IFAC Proceedings Volumes**, Elsevier, v. 35, n. 1, p. 319–324, 2002. ISSN 1474-6670. Citado nas páginas 39 e 140.

POINOT, T.; TRIGEASSOU, J.-C. Identification of fractional systems using an output-error technique. **Nonlinear Dynamics**, Springer Science and Business Media LLC, v. 38, n. 1–4, p. 133–154, 2004. Citado nas páginas 27 e 109.

QURESHI, S.; AZIZ, S. Fractional modeling for a chemical kinetic reaction in a batch reactor via nonlocal operator with power law kernel. **Physica A: Statistical Mechanics and its Applications**, Elsevier, v. 542, p. 123494, 2020. Citado na página 39.

RADWAN, A.; SOLIMAN, A.; ELWAKIL, A.; SEDEEK, A. On the stability of linear systems with fractional-order elements. **Chaos, Solitons & Fractals**, v. 40, n. 5, p. 2317–2328, 2009. ISSN 0960-0779. Citado na página 41.

RAMOS, R. V. Analytical solutions of cubic and quintic polynomials in micro and nanoelectronics using the lambert-tsallis wq function. **Journal of Computational Electronics**, Springer Science and Business Media LLC, v. 21, n. 2, p. 396–400, 2022. Citado nas páginas 59 e 61.

RASOULI, H.; FATEHI, A.; ZAMANIAN, H. Design and implementation of fractional order pole placement controller to control the magnetic flux in damavand tokamak. **Review of Scientific Instruments**, AIP Publishing, v. 86, n. 3, 2015. Citado na página 162.

RAYNAUD, H.-F.; ZERGAÏNOH, A. State-space representation for fractional order controllers. **Automatica**, Elsevier BV, v. 36, n. 7, p. 1017–1021, 2000. Citado na página 162.

RIEMANN, B. Versuch einer allgemeinen auffassung der integration und differentiation. In: **Gesammelte mathematische Werke und wissenschaftlicher Nachlass**. Leipzig: Teubner, 1876. p. 331–344. Citado na página 33.

ROSS, B. The development of fractional calculus 1695–1900. **Historia Mathematica**, v. 4, n. 1, p. 75–89, 1977. ISSN 0315-0860. Citado na página 33.

SABATIER, J. Fractional state space description: A particular case of the Volterra equations. **Fractal and Fractional**, MDPI AG, v. 4, n. 2, p. 23, 2020. Citado na página [162](#).

SAFAEI, M.; TAVAKOLI, S. Tuning of robust fractional-order phase-lead compensators using pole placement and pole-zero ratio minimization. **Journal of Vibration and Control**, SAGE Publications Sage UK: London, England, v. 24, n. 22, p. 5379–5390, 2018. Citado na página [39](#).

SAMKO, S. G.; KILBAS, A. A.; MARICHEV, O. I. Fractional integrals and derivatives. **Theory and applications**, Gordon and Breach, 1993. Citado nas páginas [33](#) e [35](#).

SANTOS, N. M. L.; MARTINS, P. R. O.; NETO, G. C. S.; JUNIOR, F. A. C. A. Projeto e implementação de um controlador FOPI em uma bancada de controle de nível. In: **Proceedings do XXIV Congresso Brasileiro de Automática**. Fortaleza, Ceará: SBA Sociedade Brasileira de Automática, 2022. (CBA2022). Citado na página [134](#).

SARIR, N.; MORSLI, S.; ALLAOUI, T.; DENAI, M. Optimal fractional-order PI control design for a variable speed PMSG-based wind turbine. **Journal European des Systemes Automatisés (JESA)**, International Information and Engineering Technology Association, v. 54, n. 6, p. 915–922, 2021. Citado na página [128](#).

SARWAR, S.; ALEEM, M.; IMRAN, M. A.; AKGÜL, A. A comparative study on non-newtonian fractional-order brinkman type fluid with two different kernels. **Numerical Methods for Partial Differential Equations**, v. 40, n. 1, p. e22688, 2024. Citado na página [40](#).

SETIAWAN, M. H.; MA'ARIF, A.; SAIFUDDIN, M. F.; SALAH, W. A. A comparative study of PID, FOPID, ISF, SMC, and FLC controllers for DC motor speed control with particle swarm optimization. **International Journal of Robotics and Control Systems**, ASCEE Publications, v. 5, n. 1, p. 640–660, 2025. Citado na página [129](#).

SHAH, P.; AGASHE, S. Review of fractional PID controller. **Mechatronics**, Elsevier, v. 38, p. 29–41, 2016. Citado nas páginas [25](#) e [38](#).

SHAMI, T. M.; EL-SALEH, A. A.; ALSWAIITI, M.; AL-TASHI, Q.; SUMMAKIEH, M. A.; MIRJALILI, S. Particle swarm optimization: A comprehensive survey. **IEEE Access**, Institute of Electrical and Electronics Engineers (IEEE), v. 10, p. 10031–10061, 2022. Citado na página [128](#).

SHAPIRO, H. N. **Introduction to the Theory of Numbers**. New York: Courier Corporation, 2008. Citado na página [63](#).

Silva, G.; RAMOS, R. The Lambert–Tsallis W_q function. **Physica A: Statistical Mechanics and its Applications**, Elsevier, v. 525, p. 164–170, 2019. ISSN 0378-4371. Citado nas páginas [28](#), [54](#) e [55](#).

SILVA, J. L. E. da. Geometric relations in classical and quantum information theory using the Lambert-Tsallis $W_q(x)$ function. **Journal of Computational and Applied Mathematics**, Elsevier, v. 441, p. 115664, 2024. ISSN 0377-0427. Citado na página [55](#).

SOLOMON, O. O.; ZHENG, W.; CHEN, J.; QIAO, Z. State of charge estimation of lithium-ion battery using an improved fractional-order extended Kalman filter. **Journal of Energy Storage**, Elsevier BV, v. 49, p. 104007, 2022. Citado nas páginas [25](#) e [39](#).

TASHAKORI, S.; SAN-MILLAN, A.; VAZIRI, V.; APHALE, S. S. Fast parameter identification of the fractional-order creep model. **Actuators**, v. 13, n. 12, 2024. Citado na página 109.

TEODORO, G. S.; MACHADO, J. T.; OLIVEIRA, E. C. D. A review of definitions of fractional derivatives and other operators. **Journal of Computational Physics**, v. 388, p. 195–208, 2019. Citado na página 33.

TEPLJAKOV, A. **Fractional-order calculus based identification and control of linear dynamic systems**. Dissertação (Mestrado) – Tallinn University of Technology, Tallinn, Estonia, 2011. Citado na página 95.

TEPLJAKOV, A.; ALAGOZ, B. B.; YEROGLU, C.; GONZALEZ, E. A.; HOSSEINNIA, S. H.; PETLENKOV, E.; ATES, A.; CECH, M. Towards industrialization of FOPID controllers: A survey on milestones of fractional-order control and pathways for future developments. **IEEE Access**, Institute of Electrical and Electronics Engineers (IEEE), v. 9, p. 21016–21042, 2021. Citado na página 25.

TEPLJAKOV, A.; PETLENKOV, E.; BELIKOV, J. FOMCON toolbox for modeling, design and implementation of fractional-order control systems. In: _____. **Volume 6 Applications in Control**. Berlin, Boston: De Gruyter, 2019. p. 211–236. ISBN 9783110571745. Disponível em: <https://doi.org/10.1515/9783110571745-010>. Citado na página 95.

TORVIK, P. J.; BAGLEY, R. L. On the appearance of the fractional derivative in the behavior of real materials. **Journal of Applied Mechanics**, v. 51, n. 2, p. 294–298, 1984. ISSN 0021-8936. Citado na página 109.

UMAROV, S.; TSALLIS, C.; STEINBERG, S. On a q -central limit theorem consistent with nonextensive statistical mechanics. **Milan journal of mathematics**, Springer, v. 76, n. 1, p. 307–328, 2008. Citado na página 54.

UÇAR, E.; ÖZDEMİR, N. New fractional cancer mathematical model via IL-10 cytokine and anti-PD-L1 inhibitor. **Fractal and Fractional**, v. 7, n. 2, 2023. ISSN 2504-3110. Citado na página 25.

VALÉRIO, D. Non-integer order robust control: an application. In: **Student Forum**. Porto, Portugal: European Union Control Association (EUCA), 2001. p. 25–28. Citado na página 95.

VALÉRIO, D.; COSTA, J. Ninteger: a non-integer control toolbox for Matlab. In: . Bordeaux, França: ENSEIRB, 2004. Citado na página 95.

VALÉRIO, D.; MACHADO, J. T.; KIRYAKOVA, V. Some pioneers of the applications of fractional calculus. **Fractional Calculus and Applied Analysis**, v. 17, n. 2, p. 552–578, 2014. Citado nas páginas 33 e 34.

VÁZQUEZ, L.; TRUJILLO, J. J.; VELASCO, M. P. Fractional heat equation and the second law of thermodynamics. **Fractional Calculus and Applied Analysis**, Springer Science and Business Media LLC, v. 14, n. 3, p. 334–342, 2011. Citado nas páginas 39 e 109.

WANG, D.; TAN, D.; LIU, L. Particle swarm optimization algorithm: an overview. **Soft Computing**, Springer Science and Business Media LLC, v. 22, n. 2, p. 387–408, 2017. Citado na página 128.

WANG, H.; LI, B.; MO, H.; XU, Y.; XU, Z.; XIE, Y. A novel fractional-order model of medium-frequency respiratory impedance. In: **2024 17th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)**. Shanghai, China: IEEE, 2024. p. 1–6. Citado na página [40](#).

XUE, D. **Fractional-order control systems: fundamentals and numerical implementations**. Berlim, Alemanha: Walter De Gruyter Inc., 2017. v. 1. Citado nas páginas [27](#), [35](#), [90](#) e [95](#).

XUE, D.; BAI, L. **Fractional Calculus: High-Precision Algorithms and Numerical Implementations**. Singapura: Springer Nature, 2024. Citado nas páginas [34](#), [95](#) e [96](#).

XUE, D.; LI, T.; LIU, L. A MATLAB toolbox for multivariable linear fractional-order control systems. In: **2017 29th Chinese Control And Decision Conference (CCDC)**. Changsha, China: IEEE, 2017. Citado na página [95](#).

YU, Y.; GAO, Q.; ZHAO, X.; JI, Y. 2-D modeling and analysis of time-domain electromagnetic anomalous diffusion with space-fractional derivative. **IEEE Transactions on Geoscience and Remote Sensing**, IEEE, v. 61, p. 1–13, 2023. Citado na página [25](#).

ZHANG, X.; YANG, Y. Optimization of PID controller parameters using a hybrid PSO algorithm. **International Journal of Dynamics and Control**, Springer Science and Business Media LLC, v. 12, n. 10, p. 3617–3627, 2024. Citado na página [129](#).

ZHAO, C.; XUE, D. Closed-form solutions to fractional-order linear differential equations. **Frontiers of Electrical and Electronic Engineering in China**, v. 3, n. 2, p. 214–217, 2008. ISSN 1673-3584. Citado na página [96](#).

APÊNDICE A – DESCRIÇÃO DA MÁQUINA UTILIZADA PARA SIMULAÇÕES COMPUTACIONAIS

Para realização de simulações computacionais, é utilizado o *notebook* G15 5511 da marca DELL, com processador i7-11800H de 4,60GHz, 16 GB de memória RAM e placa de vídeo NVIDIA GeForce RTX 3060.

APÊNDICE B – REGRESSÃO BASEADA EM MLP

A seguir, é descrito o funcionamento básico de uma MLP totalmente conectada e com uma camada oculta, treinada via algoritmo de retropropagação do erro. Na iteração t , a ativação i -ésimo neurônio oculto, $i = 1, \dots, Q$, é dada por

$$u_i^{(h)}(t) = \sum_{j=1}^P w_{ij}(t)x_j(t) - \theta_i(t) = \sum_{j=0}^P w_{ij}(t)x_j(t), \quad (\text{B.1})$$

em que w_{ij} é o peso sináptico que conecta a entrada j ao neurônio oculto i , $\theta_i(t)$ é o limiar do neurônio oculto i , $2 \leq Q < \infty$ é o número de neurônios ocultos e P é a dimensão do vetor de entrada, excluindo o limiar. Para simplificar, define-se $x_0(t) = -1$ e $w_{i0} = \theta_i^{(h)}(t)$. A saída do neurônio i é então definida por

$$y_i^{(h)}(t) = \varphi_i \left[u_i^{(h)}(t) \right] = \varphi_i \left[\sum_{j=0}^P w_{ij}(t)x_j(t) \right],$$

em que $\varphi_i(\cdot)$ é a função tangente hiperbólica. Da mesma forma, os valores de saída dos neurônios de saída são dados por

$$y_k^{(o)}(t) = \varphi_k \left[u_k^{(o)}(t) \right] = \varphi_k \left[\sum_{i=0}^Q m_{ki}(t)y_i^{(h)}(t) \right],$$

em que m_{ki} é o peso sináptico que conecta o neurônio oculto i ao neurônio de saída $k = 1, \dots, M$, e $M \geq 1$ é o número de neurônios da camada de saída. Define-se $y_0^{(h)}(t) = -1$ e $m_{k0} = \theta_k^{(o)}(t)$, onde $\theta_k^{(o)}(t)$ é o limiar do neurônio k .

A retropropagação começa na camada de saída propagando os sinais de erro, $e_k^{(o)}(t) = d_k(t) - y_k^{(o)}(t)$, onde $d_k(t)$ é o rótulo do neurônio k , em direção à camada oculta. O chamado *gradiente local* do neurônio k é dado por

$$\delta_k^{(o)}(t) = \varphi'_k \left[u_k^{(o)}(t) \right] e_k^{(o)}(t),$$

em que $\varphi'_k \left[u_k^{(o)}(t) \right] = \partial \varphi_k / \partial u_k^{(o)}$. Da mesma forma, o *gradiente local* $\delta_i^{(h)}(t)$ do neurônio oculto i é calculado como

$$\delta_i^{(h)}(t) = \varphi'_i \left[u_i^{(h)}(t) \right] \sum_{k=1}^M m_{ki}(t) \delta_k^{(o)}(t) = \varphi'_i \left[u_i^{(h)}(t) \right] e_i^{(h)}(t), \quad (\text{B.2})$$

em que o termo $e_i^{(h)}(t)$ desempenha o papel de um sinal de erro retropropagado ou projetado para o neurônio oculto i , uma vez que tais sinais de erro “ocultos” são combinações dos sinais de

erro “verdadeiros” calculados para os neurônios de saída. Finalmente, os pesos sinápticos dos neurônios de saída são atualizados de acordo com

$$m_{ki}(t+1) = m_{ki}(t) + \eta(t) \delta_k^{(o)}(t) y_i^{(h)}(t), \quad i = 0, \dots, Q,$$

em que $0 < \eta(t) < 1$ é a taxa de aprendizado. Os pesos dos neurônios ocultos são, por sua vez, ajustados por meio de uma regra de aprendizagem semelhante,

$$w_{ij}(t+1) = w_{ij}(t) + \eta(t) \delta_i^{(h)}(t) x_j(t), \quad j = 0, \dots, P.$$

Uma apresentação completa de todo o conjunto de treinamento durante o processo de aprendizagem é chamada de época. Muitas épocas podem ser necessárias até que a convergência do algoritmo de retropropagação aconteça. Uma maneira simples de avaliar a convergência é por meio do erro quadrático médio,

$$\varepsilon_{train} = \frac{1}{2N} \sum_{t=1}^N \sum_{k=1}^M \left[e_k^{(o)}(t) \right]^2 = \frac{1}{2N} \sum_{t=1}^N \sum_{k=1}^M \left[d_k(t) - y_k^{(o)}(t) \right]^2, \quad (\text{B.3})$$

calculado no final de uma execução de treinamento usando os vetores de dados de treinamento.

APÊNDICE C – RETROFIT DE UMA INCUBADORA NEONATAL

A incubadora neonatal utilizada é do modelo C186 TS da FANEM[®], ilustrada na [Figura 63.a](#). Um desenho esquemático desta planta térmica pode ser observado na [Figura 63.b](#). Mais detalhes da unidade controladora presente na incubadora estão ilustrados na [Figura 63.c](#).

Nota-se que a forma como a incubadora fornece calor para o ambiente interno é através de um resistor de aquecimento. Não há controle de umidade. Como o equipamento é de uso comercial, não há documentação pública de seu projeto. Por esse motivo, foi realizada a identificação visual do esquema eletrônico da placa da unidade controladora (vide [Figura 63.c](#)), em que o circuito relacionado ao acionamento do resistor de aquecimento é ilustrado na [Figura 64](#).

Percebe-se que o acionamento do resistor de aquecimento é realizado por um TRIAC (*triode for alternating current*), que por sua vez é acionado por ângulo de disparo. É essencial entender o conceito de ângulo de disparo para que se possa realizar o *retrofit* da placa eletrônica corretamente. A forma de onda de tensão aplicada a uma carga resistiva acionada por ângulo de disparo em TRIAC é ilustrada na [Figura 65.a](#), enquanto que a tensão média (V_{med}) aplicada nessa carga resistiva é ilustrada na [Figura 65.b](#) de acordo com

$$V_{med} = \frac{V_s}{2} [1 + \cos(\theta_d)] , \quad (C.1)$$

em que V_s é o valor eficaz da tensão de entrada e θ_d é o ângulo de disparo. Como pode ser observado, a relação entre V_{med} e θ_d é não linear, sendo necessária a linearização da curva para utilização de técnicas lineares de identificação e controle.

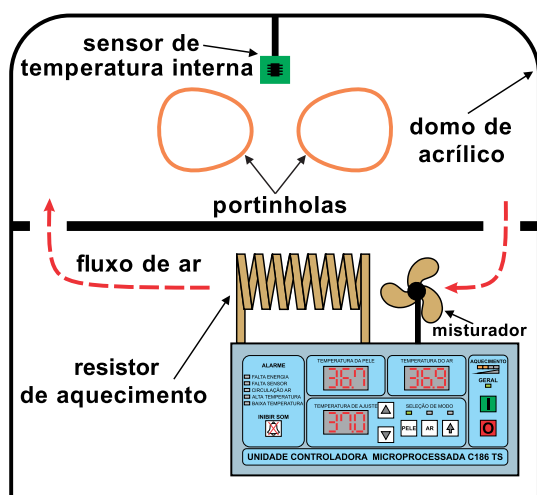
Para realizar a identificação e posteriormente o controle da planta, foram implementadas algumas modificações no *hardware* do equipamento, *i.e.*, na placa eletrônica da unidade de processamento. As conexões entre o microcontrolador original (8051), o acionamento do TRIAC e os relés de segurança foram desfeitas. Assim, um microcontrolador externo pôde ser acoplado e novas conexões puderam ser criadas, como a detecção da passagem por zero (*zero cross*) e o acionamento do TRIAC.

O microcontrolador utilizado é o ATMEGA328p, que possui 2 KB de SRAM e 32 KB de memória de programa. A função do microcontrolador no *retrofit* realizado é a de fornecer um meio de acionar o resistor de aquecimento, tornando possível a realização de ensaios de identificação e controle do sistema. É utilizado um sensor de temperatura interno à incubadora com interface digital. Os dados podem ser salvos em um cartão SD ou transmitidos via comunicação serial. Essas conexões podem ser observadas na [Figura 66](#).

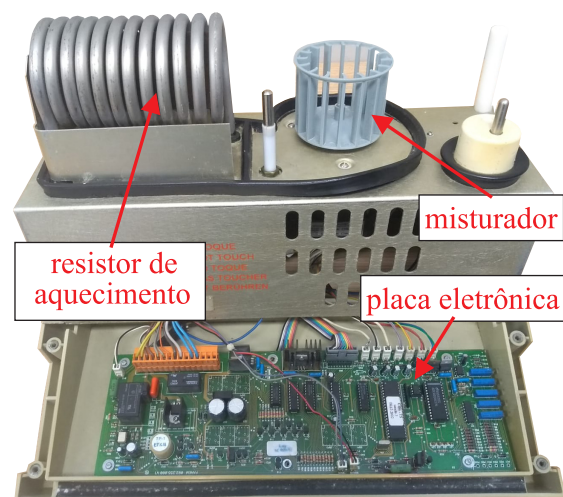
Figura 63 – Incubadora neonatal C186 TS, FANEM®.



(a) Planta experimental.



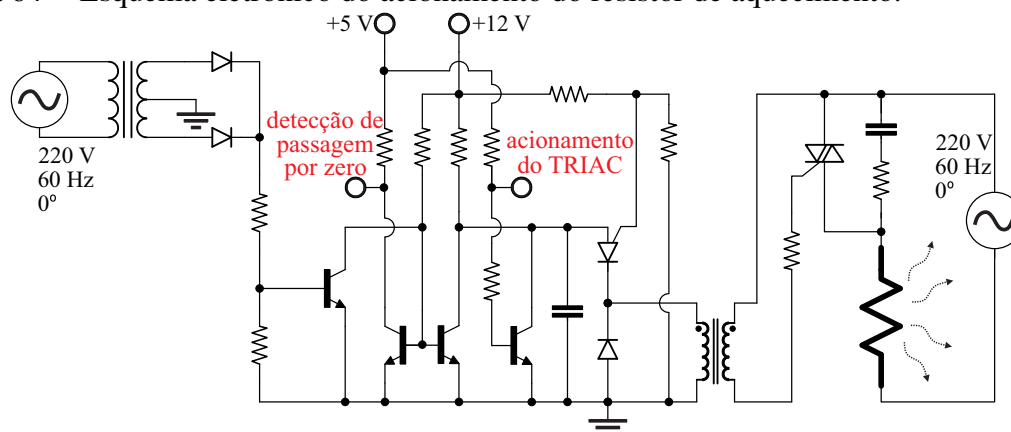
(b) Desenho esquemático da planta térmica.



(c) Unidade controladora da incubadora.

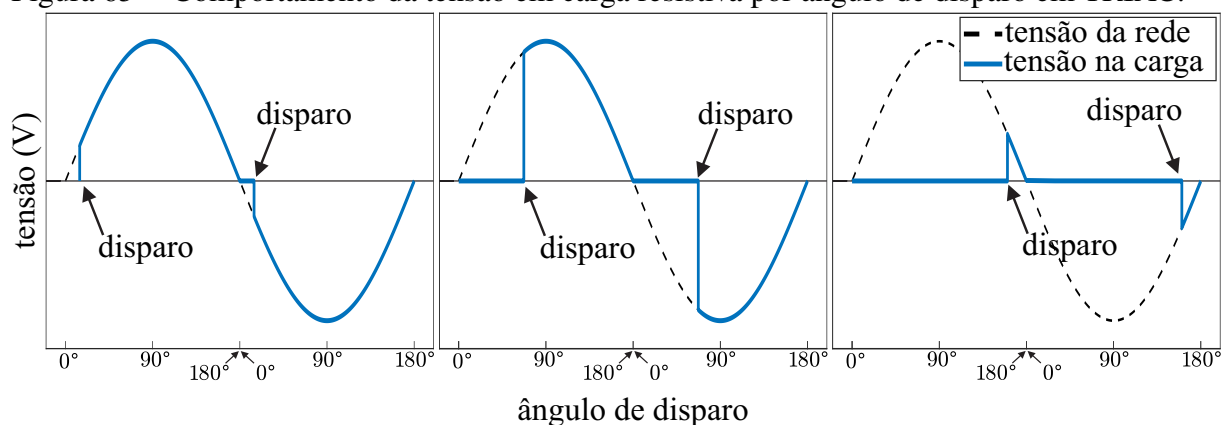
Fonte: elaborada pelo autor.

Figura 64 – Esquema eletrônico do acionamento do resistor de aquecimento.

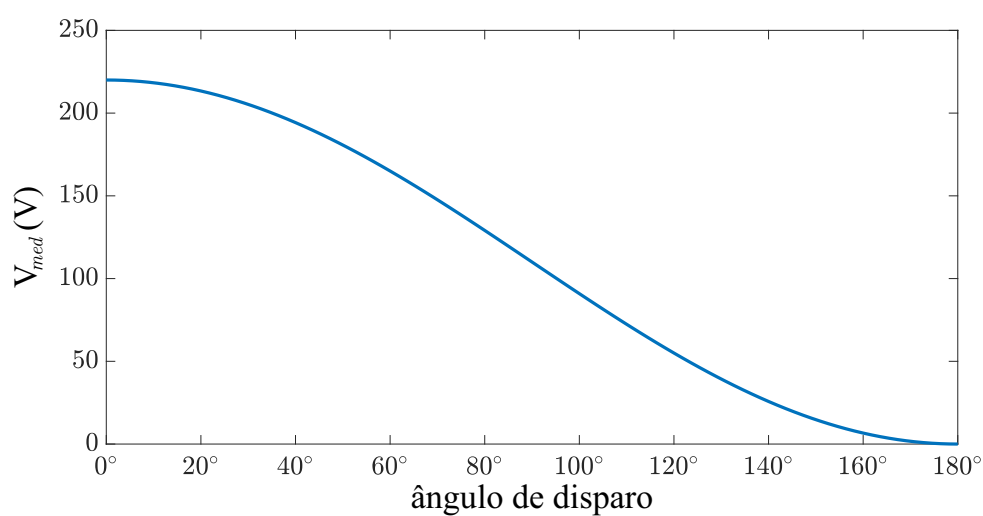


Fonte: elaborada pelo autor.

Figura 65 – Comportamento da tensão em carga resistiva por ângulo de disparo em TRIAC.



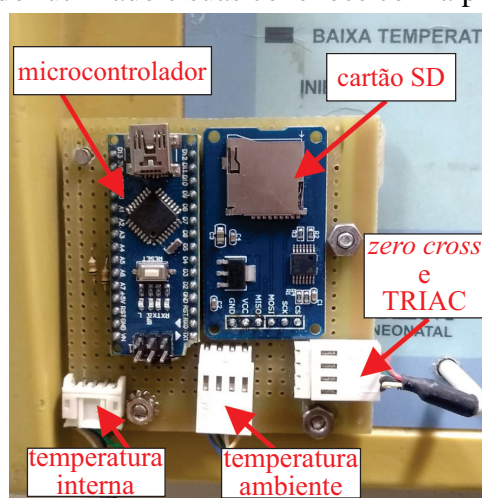
(a) Tensão aplicada à carga para diferentes ângulos de disparo.



(b) Tensão média por ângulo de disparo.

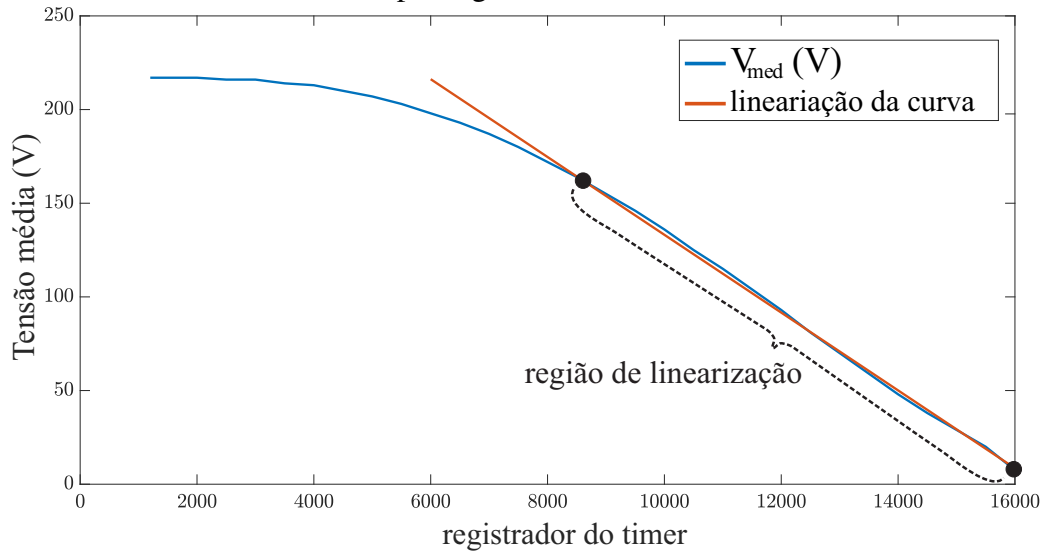
Fonte: elaborada pelo autor.

Figura 66 – Microcontrolador utilizado e suas conexões com a placa eletrônica da incubadora.



Fonte: elaborada pelo autor.

Figura 67 – Ensaio da tensão média por registrador do *timer* do microcontrolador.



Fonte: elaborada pelo autor.

C.1 Definição do sinal de controle

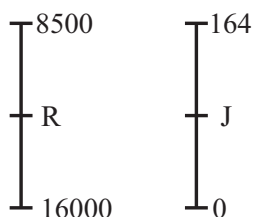
Para fornecer uma determinada tensão V_{med} à carga, o microcontrolador externo recebe o sinal da passagem por zero, contabiliza o tempo necessário para atingir o ângulo de disparo desejado e finalmente realiza o disparo do TRIAC (vide Figura 65.a). Como exemplo, pode-se mencionar que para um ângulo de 180° , deve-se contabilizar $8,33ms$ para realizar o disparo. As aquisições de dados realizadas utilizam o valor do registrador do *timer* do microcontrolador como entrada do sistema. Assim, o modelo da planta obtido capta as dinâmicas de temperatura interna da incubadora, do conversor de tensão CA-CA, seu circuito de disparo e finalmente o *timer* do microcontrolador utilizado. Um ensaio de tensão média (V_{med}) por registrador do *timer* foi realizado e pode ser observado na Figura 67.

Pode-se observar que a curva de tensão média é não linear. Para que um controlador linear funcione corretamente com este conversor, deve-se trabalhar apenas na região linear da curva, como ilustrado na Figura. Além disso, nota-se sua relação inversamente proporcional. Para corrigir este problema, é feita uma normalização dos valores do registrador R para a variável auxiliar J conforme a Figura 68. Os valores limites da variável J foram escolhidos como sendo os valores referentes às tensões médias aplicadas V_{med} . O resultado da normalização é escrito como

$$J = -0,021867R + 349,8667, \quad (C.2)$$

$$R = -45,73171J + 16000. \quad (C.3)$$

Figura 68 – Normalização entre tensão média aplicada e registrador do *timer*.



Fonte: elaborada pelo autor.

É interessante mencionar que a variável J é a variável de controle do sistema, sendo então a variável utilizada como entrada u no processo de identificação.

C.2 Aquisição de dados

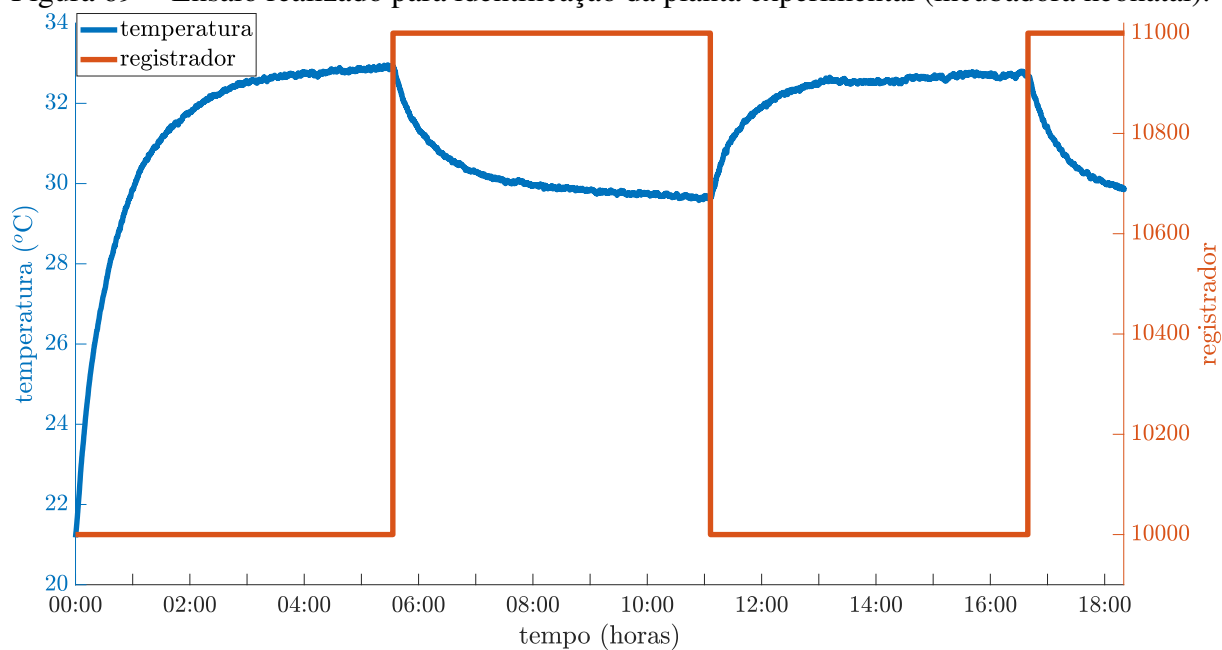
Um ensaio de aquisição foi realizado com taxa $0,1\text{ Hz}$ e é ilustrado na [Figura 69](#). O valor da taxa de aquisição foi escolhido com base em testes preliminares. O sinal de entrada PRBS (*pseudo-random binary signal*) está da faixa linearizada de tensão média aplicada (vide [Figura 67](#)). Pode-se observar a relação inversamente proporcional entre o valor do registrador do *timer* e a temperatura interna da incubadora. Com a aplicação da [Equação C.2](#) e uma reamostragem, o novo sinal passar a ter uma taxa de amostragem de 10 mHz , como ilustrado na [Figura 70](#). Este conjunto de dados é utilizado na etapa de identificação do sistema, cujos resultados são apresentados no estudo de caso no [Capítulo 8](#).

C.3 Sistema de controle

Com base nos conceitos apresentados neste apêndice, a [Figura 71](#) ilustra uma proposta de sistema de controle discreto de temperatura com controlador $\text{PI}^\lambda \text{D}^\mu$ utilizando o *retrofit* realizado. O microcontrolador calcula o erro $E(z)$ entre o *setpoint* e a medida do sensor, e o usa para gerar o sinal de controle $U(z)$. Esse sinal é saturado em $U_{sat}(z)$ para manter-se dentro dos limites permitidos (vide [Figura 67](#)) e, em seguida, passa por um *hold* de ordem zero, resultando em $U_{sat}(s)$. Por fim, o microcontrolador converte esse valor na variável $J(s)$ a partir da [Equação C.2](#), da utilização do periférico *timer* e do sinal de passagem por zero (vide [Figura 64](#)).

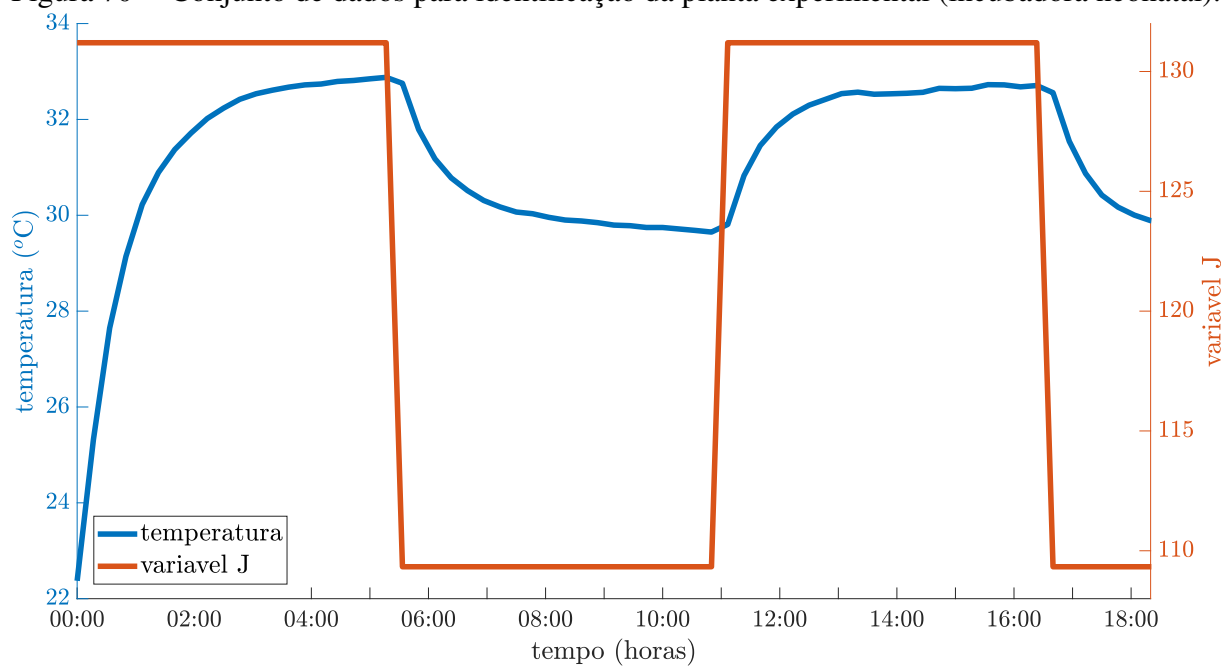
A partir de $J(s)$, o *driver* aciona o conversor CA/CC em um determinado ângulo de disparo $\theta_d(s)$, resultando na tensão média $V_{med}(s)$ (vide [Figura 64](#) e [Equação C.1](#)). A tensão $V_{med}(s)$ é aplicada ao resistor de aquecimento, que gera calor por efeito Joule e o transfere à câmara da incubadora na forma de fluxo térmico, representado por $\phi(s)$.

Figura 69 – Ensaio realizado para identificação da planta experimental (incubadora neonatal).



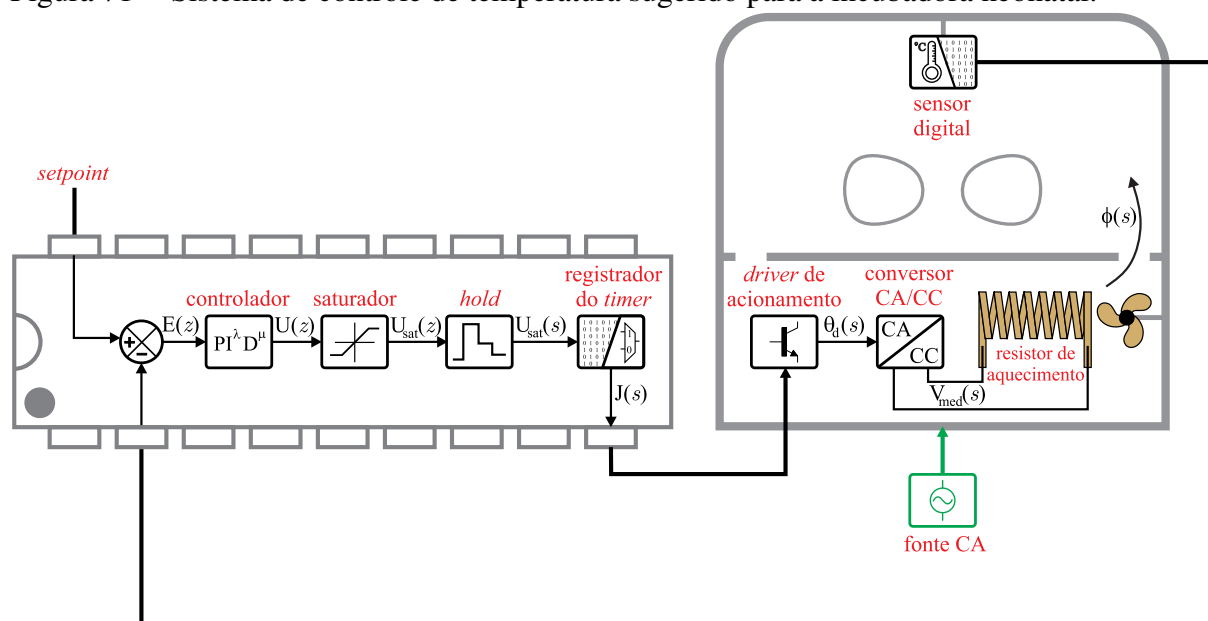
Fonte: elaborada pelo autor.

Figura 70 – Conjunto de dados para identificação da planta experimental (incubadora neonatal).



Fonte: elaborada pelo autor.

Figura 71 – Sistema de controle de temperatura sugerido para a incubadora neonatal.



Fonte: elaborada pelo autor.

Por fim, é importante destacar que os dados de aquisição apresentados na [Figura 70](#) podem ser utilizados para obter um modelo baseado em caixa preta. Esse modelo abrange diversas etapas complexas, desde o registrador do *timer* (periférico interno ao microcontrolador), passando pelo *driver* de acionamento, conversor CA/CC, resistor de aquecimento e a estrutura térmica da câmara da incubadora. Cada uma dessas etapas, se modelada isoladamente, envolve desafios significativos, como não linearidades e presença de tempo morto. Assim, a identificação de um modelo único, ajustado diretamente a partir dos dados reais de entrada e saída, oferece uma alternativa prática e eficiente para representar o comportamento global do sistema.