



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA BAHIA
CAMPUS VITÓRIA DA CONQUISTA
COORDENAÇÃO DO CURSO DE ENGENHARIA ELÉTRICA**

RAFAEL SILVA NOGUEIRA PACHECO

**DESENVOLVIMENTO DE UM ALGORITMO UTILIZANDO OS MÉTODOS DE
NEWTON-RAPHSON E DE LEVENBERG-MARQUARDT PARA A SINTONIA DE
CONTROLADORES PID COM BASE EM CRITÉRIOS DE DESEMPENHO**

Vitória da Conquista - BA

Dezembro de 2022

RAFAEL SILVA NOGUEIRA PACHECO

**DESENVOLVIMENTO DE UM ALGORITMO UTILIZANDO OS MÉTODOS DE
NEWTON-RAPHSON E DE LEVENBERG-MARQUARDT PARA A SINTONIA DE
CONTROLADORES PID COM BASE EM CRITÉRIOS DE DESEMPENHO**

Projeto de Final de Curso apresentado ao Curso de Engenharia Elétrica do Instituto Federal de Educação, Ciência e Tecnologia da Bahia, Campus Vitória da Conquista, como requisito parcial para a obtenção do título de Bacharel em Engenharia Elétrica.

Orientador: Prof. Me. Gislân Silveira Santos

Vitória da Conquista - BA

Dezembro de 2022

P116d PACHECO, Rafael Silva Nogueira.

Desenvolvimento de um algoritmo utilizando os métodos de Newton-Raphson e de Levenberg-Marquardt para a sintonia de controladores PID com base em critérios de desempenho. / Rafael Silva Nogueira Pacheco - - Vitória da Conquista: Instituto Federal de Educação, Ciência e Tecnologia da Bahia - IFBA, 2022.

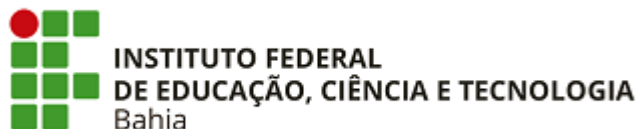
88f.

II. Color.

Projeto de final de curso, como requisito parcial para obtenção do título de Bacharel em Engenharia Elétrica, pelo Instituto Federal de Educação, Ciência e Tecnologia da Bahia – IFBA, campus Vitória da Conquista. Orientador Prof. Me. Gislan Silveira Santos.

1. Engenharia Elétrica – Projeto. 2. Desenvolvimento de algoritmo. 3. Sintonia de controladores PID. I. Instituto Federal de Educação, Ciência e Tecnologia da Bahia – IFBA. II. SANTOS, Gislan Silveira. III. Desenvolvimento de um algoritmo utilizando os métodos de Newton-Raphson e de Levenberg-Marquardt para a sintonia de controladores PID com base em critérios de desempenho

CDD: 621.3



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA BAHIA
Av. Sérgio Vieira de Mello, 3150 - Bairro Zabelê - CEP 45078-900 - Vitória da Conquista - BA - www.portal.ifba.edu.br

CARTA

DESENVOLVIMENTO DE UM ALGORITMO UTILIZANDO OS MÉTODOS DE NEWTON-RAPHSON E DE LEVENBERG-MARQUARDT PARA A SINTONIA DE CONTROLADORES PID COM BASE EM CRITÉRIOS DE DESEMPENHO

RAFAEL SILVA NOGUEIRA PACHECO

A presente Monografia, apresentada em sessão pública realizada em 15 de dezembro de 2022, foi avaliada como adequada para obtenção do Grau de Engenheiro Eletricista, julgada e aprovada em sua forma final pela Coordenação do Curso de Engenharia Elétrica do Instituto Federal de Educação, Ciência e Tecnologia da Bahia, *Campus* Vitória da Conquista.

BANCA EXAMINADORA

Prof. Me. Gislan Silveira Santos (Orientador)..... IFBA, *Campus* Vitória da Conquista
Prof. Me. Allan de Sousa Soares..... IFBA, *Campus* Vitória da Conquista
Prof. Me. Celton Ribeiro Barbosa.....IFBA, *Campus* Brumado

Vitória da Conquista - Bahia - Brasil
Dezembro, 2022



Documento assinado eletronicamente por **GISLAN SILVEIRA SANTOS, Professor(a) do Ensino Básico, Técnico e Tecnológico - EBTT**, em 19/12/2022, às 21:52, conforme decreto nº 8.539/2015.



Documento assinado eletronicamente por **ALLAN DE SOUSA SOARES, Professor(a) (Nível Superior)**, em 19/12/2022, às 21:58, conforme decreto nº 8.539/2015.



Documento assinado eletronicamente por **Celton Ribeiro Barbosa, Professor(a) do Ensino Básico, Técnico e Tecnológico - EBTT**, em 19/12/2022, às 22:08, conforme decreto nº 8.539/2015.



A autenticidade do documento pode ser conferida no site
[http://sei.ifba.edu.br/sei/controlador_externo.php?](http://sei.ifba.edu.br/sei/controlador_externo.php?acao=documento_conferir&acao_origem=documento_conferir&id_orgao_acesso_externo=0)
[acao=documento_conferir&acao_origem=documento_conferir&id_orgao_acesso_externo=0](http://sei.ifba.edu.br/sei/controlador_externo.php?acao=documento_conferir&acao_origem=documento_conferir&id_orgao_acesso_externo=0)
informando o código verificador **2681214** e o código CRC **A009348B**.

RESUMO

O presente trabalho propõe uma nova técnica de sintonia de controladores PID para o controle de processos LTI do tipo SISO, baseando-se em critérios de desempenho no domínio do tempo. Esse procedimento se dá por meio da aplicação dos métodos iterativos de Newton-Raphson e de Levenberg-Marquardt para a resolução dos sistemas de equações não lineares que descrevem o quão próximo os parâmetros da resposta ao degrau estão do desejado. Para tanto, um desenvolvimento algébrico foi conduzido para a determinação do jacobiano das funções de desempenho com relação aos coeficientes do controlador, permitindo a implementação dos métodos numéricos supracitados. Simulações numéricas do algoritmo proposto foram realizadas no software MATLAB para três funções de transferência de diferentes ordens. Os resultados demonstraram a viabilidade de utilização desse algoritmo de sintonia, desde que certas restrições sobre a função de transferência do processo sejam obedecidas, e uma comparação de performance entre os dois métodos numéricos evidenciou as vantagens e desvantagens de cada um. Além disso, mostrou-se que a taxa de convergência dos métodos pode ser amplificada quando os parâmetros de desempenho da curva de resposta inicial estão relativamente próximos do desejado, sugerindo que uma sintonia prévia por técnicas já estabelecidas aumenta as chances de sucesso do algoritmo.

Palavras-chave: sintonia; controladores PID; critérios de desempenho; Newton-Raphson; Levenberg-Marquardt.

ABSTRACT

This work proposes a new tuning technique for PID controllers to control SISO LTI processes, based on performance criteria in the time domain. This procedure is carried out through the application of the Newton-Raphson and the Levenberg-Marquardt iterative methods to solve the nonlinear set of equations that describe how close the step response parameters are to the desired ones. Therefore, an algebraic development was conducted to determine the jacobian of the performance functions with respect to the controller coefficients, allowing the implementation of the aforementioned numerical methods. Numerical simulations of the proposed algorithm were performed in the software MATLAB for three transfer functions of different orders. The results demonstrated the feasibility of using this tuning algorithm, as long as certain restrictions on the transfer function of the process are obeyed, and a performance comparison between the two numerical methods showed the advantages and disadvantages of each one. In addition, it was shown that the convergence rate of the methods can be amplified when the performance parameters of the initial response curve are relatively close to the desired ones, suggesting that a prior tuning by already established techniques increases the chances of success of the algorithm.

Keywords: tuning; PID controllers; performance criteria; Newton-Raphson; Levenberg-Marquardt.

LISTA DE ILUSTRAÇÕES

Figura 2.1	Representação em diagrama de blocos de um sistema.	17
Figura 2.2	Circuito RLC série.	18
Figura 2.3	Sinal contínuo aproximado como uma soma de pulsos retangulares de comprimento fixo.	22
Figura 2.4	Evolução temporal de uma exponencial complexa.	24
Figura 2.5	Diagrama de polos e zeros no plano s	28
Figura 2.6	Resposta do sistema com base na localização dos polos.	32
Figura 2.7	Diagrama de um sistema de controle em malha aberta.	33
Figura 2.8	Diagrama de um sistema de controle em malha fechada.	34
Figura 2.9	Interpretação gráfica dos parâmetros de desempenho na curva de resposta ao degrau.	39
Figura 2.10	Curvas de resposta ao degrau do processo $G(s) = (s^2 + 2s + 2)^{-1}$ em sistemas de malha aberta e de malha fechada com controle P para diferentes valores de ganho proporcional.	40
Figura 2.11	Curvas de resposta ao degrau do processo $G(s) = (s^2 + 2s + 2)^{-1}$ em sistemas de malha aberta e de malha fechada com controle PI para diferentes valores de ganho integral e $K_P = 5$ fixo.	41
Figura 2.12	Curvas de resposta ao degrau do processo $G(s) = (s^2 + 2s + 2)^{-1}$ em sistemas de malha aberta e de malha fechada com controle PD para diferentes valores de ganho derivativo e $K_P = 5$ fixo.	42
Figura 2.13	Comparação entre os diferentes tipos de controle em um mesmo processo.	43
Figura 2.14	Diagrama de blocos de um sistema de controle em malha fechada com controle PID.	43
Figura 2.15	Valor absoluto das funções de desempenho do processo $(s + 1)^{-1}$ para uma variação contínua dos coeficientes K_P e K_I com as especificações $t_r = 0,9$ s, $t_p = 1,5$ s, $M_p = 20\%$ e $t_s = 3,85$ s para $\alpha = 2\%$	48
Figura 3.1	Fluxograma do algoritmo proposto.	69
Figura 4.1	Caminho traçado pelo algoritmo nos métodos NR e LM para o sistema de primeira ordem, tendo $\mathbf{K}^{(0)} \in \{0,4; 1,8; 3,2; 4,6\} \times \{1,6; 7,2; 12,8; 18,4\}$ como pontos iniciais.	71

Figura 4.2	Curvas de resposta ao degrau alcançadas conforme a sintonia do algoritmo para o sistema de primeira ordem.	72
Figura 4.3	Convergência dos métodos no sistema de primeira ordem para 400 valores distintos dos critérios de desempenho em cada combinação, com $\mathbf{K}^{(0)} = (1; 2)$	73
Figura 4.4	Caminho traçado pelo algoritmo nos métodos NR e LM para o sistema de segunda ordem, tendo $\mathbf{K}^{(0)} \in \{0,1; 4,9\} \times \{0,1; 4,9\} \times \{0,1; 4,9\}$ como pontos iniciais.	74
Figura 4.5	Curvas de resposta ao degrau alcançadas conforme a sintonia do algoritmo para o sistema de segunda ordem.	75
Figura 4.6	Convergência dos métodos no sistema de segunda ordem para 500 valores distintos dos critérios de desempenho em cada combinação, com $\mathbf{K}^{(0)} = (1; 2; 0,5)$	76
Figura 4.7	Caminho traçado pelo algoritmo nos métodos NR e LM para o sistema de quarta ordem, tendo $\mathbf{K}^{(0)} \in \{0,1; 19,9\} \times \{0,1; 19,9\} \times \{0,1; 9,9\}$ como pontos iniciais.	77
Figura 4.8	Curvas de resposta ao degrau alcançadas conforme a sintonia do algoritmo para o sistema de quarta ordem.	78
Figura 4.9	Convergência dos métodos no sistema de quarta ordem para 500 valores distintos dos critérios de desempenho em cada combinação, com $\mathbf{K}^{(0)} = (67,3517; 84,1896; 13,4703)$	78

LISTA DE ABREVIATURAS E SIGLAS

PID	Proporcional Integral Derivativo
SISO	Single-Input Single-Output
LTI	Linear Time-Invariant
BIBO	Bounded-Input Bounded-Output
P	Proporcional
PI	Proporcional Integral
PD	Proporcional Derivativo
PSE	Polinômio Simétrico Elementar
NR	Newton-Raphson
LM	Levenberg-Marquardt

LISTA DE SÍMBOLOS

$x(t)$	Sinal de entrada
$y(t)$	Sinal de saída
$y_n(t)$	Componente natural da resposta
$y_f(t)$	Componente forçada da resposta
$\delta(t)$	Função impulso unitário
$h(t)$	Resposta ao impulso
$\mathcal{L}\{.\}$	Operador da transformada de Laplace
s	Variável do domínio de Laplace
j	Unidade imaginária
σ	Fator de amortecimento
ω	Frequência de oscilação
$\text{Arg}\{.\}$	Argumento do número complexo
$\mathcal{L}^{-1}\{.\}$	Operador da transformada inversa de Laplace
$H(s)$	Função de transferência
$N(s)$	Polinômio do numerador da função de transferência
$D(s)$	Polinômio do denominador da função de transferência
z_k	k -ésimo zero da função de transferência
p_k	k -ésimo polo da função de transferência
r_k	k -ésimo resíduo da função de transferência
$u(t)$	Função degrau unitário
$\text{Re}\{.\}$	Parte real do número complexo
$\text{Im}\{.\}$	Parte imaginária do número complexo
$G_c(s)$	Função de transferência do controlador
$R(s)$	Entrada de referência
$G(s)$	Função de transferência do processo
$E(s)$	Erro de rastreamento
e_{ss}	Erro de estado estacionário
y_{ss}	Valor da saída no estado estacionário
t_r	Tempo de subida
t_p	Tempo de pico

M_p	Ultrapassagem percentual
t_s	Tempo de assentamento
α	Margem do tempo de assentamento
f_{tr}	Função de desempenho do tempo de subida
f_{tp}	Função de desempenho do tempo de pico
f_{mp}	Função de desempenho da ultrapassagem percentual
f_{ts}	Função de desempenho do tempo de assentamento
K_P	Coefficiente proporcional
K_I	Coefficiente integral
K_D	Coefficiente derivativo
e_k	Polinômio simétrico elementar de grau k
F	Vetor das funções de desempenho
\mathbb{R}	Conjunto dos números reais
$\mathbf{x}^{(k)}$	Valor de \mathbf{x} na k -ésima iteração
J	Matriz jacobiana
$\ \cdot\ $	Norma euclidiana
λ_k	Parâmetro de damping
I_n	Matriz identidade de ordem n
K	Vetor de coeficientes do controlador
P	Vetor de polos
R	Vetor de resíduos
$M_{m \times n}(K)$	Conjunto das matrizes $m \times n$ sobre o corpo K
\mathbb{C}	Conjunto dos números complexos
A	Matriz de coeficientes do polinômio do denominador
B	Matriz de coeficientes do polinômio do numerador
H	Matriz hessiana
ϵ	Tolerância
N_{max}	Número máximo de iterações

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Objetivos	16
1.2	Justificativa	16
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	Modelos Matemáticos de Sistemas	17
2.1.1	Equações Diferenciais de Sistemas Elétricos	18
2.1.2	Sistemas LTI e a Integral de Convolução	20
2.1.3	Transformada de Laplace	23
2.1.4	Funções de Transferência	26
2.1.5	Resposta no Domínio do Tempo	28
2.2	Controle PID	32
2.2.1	Sistemas de Controle em Malha Aberta e em Malha Fechada	33
2.2.2	CrITÉrios de Desempenho no DomÍnio do Tempo	36
2.2.3	Controlador PID	39
2.2.4	Funções de Desempenho de um Controlador PI em Sistemas de Primeira Ordem	44
2.2.5	Funções de Desempenho de um Controlador PID em Sistemas de Maior Ordem	47
2.3	Métodos de Resolução de Sistemas de Equações Não Lineares	53
2.3.1	Método de Newton-Raphson	54
2.3.2	Método de Levenberg-Marquardt	56
3	METODOLOGIA	58
3.1	Jacobiano das Funções de Desempenho para Sistemas de Primeira Ordem	59
3.2	Jacobiano das Funções de Desempenho para Sistemas de Maior Ordem	61
3.3	Algoritmo de Sintonia de Controladores PID pelos Métodos de Newton-Raphson e Levenberg-Marquardt	66
4	RESULTADOS E DISCUSSÕES	70
4.1	Primeiro Caso: Sistema de Primeira Ordem	70
4.2	Segundo Caso: Sistema de Segunda Ordem	74

4.3 Terceiro Caso: Sistema de Quarta ordem	76
5 CONCLUSÃO	79
REFERÊNCIAS	81
APÊNDICE A	85
APÊNDICE B	86
APÊNDICE C	87

1 INTRODUÇÃO

O controle automático de processos é um dos pilares que sustenta a produtividade da indústria moderna, sendo essencial para o funcionamento de qualquer sistema que exija uma resposta dinâmica estável e um certo grau de rejeição de distúrbios. Essa descrição se aplica a uma vasta gama de aplicações, desde o controle de braços robóticos em sistemas de manufatura até o controle de voo e propulsão de aeronaves. Durante muito tempo, o projeto de controladores se baseou na experiência do projetista e em tentativa e erro [1], uma tradição que induziu o surgimento de um controlador considerado universal, que seria a primeira escolha a ser testada em qualquer sistema de controle em malha fechada: o controlador *proporcional integral derivativo* (PID).

O controle PID é o algoritmo de controle mais frequentemente aplicado e está presente em cerca de 90% dos processos industriais com sistemas de controle, mesmo nos casos em que outros controladores são mais apropriados [2–4]. Em seu primórdio, esse controlador era implementado através de sistemas elétricos com relés, sistemas pneumáticos ou sistemas hidráulicos, eventualmente sendo substituídos por sistemas digitais, como controladores lógicos programáveis e microcontroladores [5].

Apesar de ter um funcionamento simples e ser facilmente aplicável, o projeto de controladores PID ainda é alvo de diversos estudos de cunho teórico e prático, mesmo após tantas décadas de seu surgimento. Na estrutura interna do controlador, estão codificados três parâmetros que devem ser ajustados de maneira apropriada, visando estabilidade e uma resposta dinâmica satisfatória para o processo a ser controlado. Qualquer descuido na sintonia dos parâmetros pode ocasionar em consequências terríveis para a performance do sistema. Para se ter uma ideia da importância dessa problemática, algumas estimativas mostram que em 30% dos sistemas de controle com baixo desempenho, uma sintonia indevida do controlador é a principal causa das irregularidades [3].

Os primeiros passos para o estabelecimento de regras padronizadas de sintonia de controladores PID foram dados por Ziegler e Nichols em 1942, ao publicarem uma série de experimentos a serem feitos na planta com o intuito de determinar os parâmetros de controle a partir dos dados coletados [6]. Esse método heurístico foi extremamente importante para a época, mas sofre com inúmeras desvantagens por utilizar poucas informações acerca do processo, resultando em sistemas com baixa robustez [7].

Nas últimas décadas, diversas novas técnicas de sintonia foram trazidas à tona devido

aos avanços tecnológicos no ramo da computação. Dentre as principais estratégias emergentes, podem-se citar a sintonia automática, a sintonia ótima, a sintonia adaptativa em tempo real e a sintonia inteligente, esta última que usa de conceitos como redes neurais e lógica fuzzy [8]. Em [4], encontra-se uma coletânea de 1731 diferentes regras de sintonia, e novos métodos continuam a surgir em publicações acadêmicas recentes.

Em métodos de sintonia analíticos, é comum a adoção de índices de qualidade no domínio do tempo que direcionam a sintonia dos parâmetros para um modelo ideal pré-estabelecido da curva de resposta do processo. Os critérios de erro integral são os mais comumente escolhidos [9], como em [10, 11]. Outra possibilidade bastante explorada é visualizar o problema de sintonia do ponto de vista de otimizações numéricas e técnicas de aproximação de curvas. O método de Newton-Raphson [12, 13], o método de Levenberg-Marquardt [14] e as normas H_2 [15] e H_∞ [16, 17] são alguns exemplos de implementações. Porém, a predominância das especificações de desempenho adotadas está em características no domínio da frequência e em critérios de erro integral, sendo que critérios alternativos da curva de resposta como o tempo de subida, o tempo de pico, a ultrapassagem percentual e o tempo de assentamento recebem menos atenção.

O propósito deste trabalho foi desenvolver um algoritmo de sintonia de controladores PID aplicável a sistemas lineares e invariantes no tempo com uma única entrada e uma única saída, utilizando os métodos iterativos de Newton-Raphson e de Levenberg-Marquardt. Isso foi realizado estipulando o atendimento simultâneo de vários critérios de desempenho no domínio do tempo e definindo funções de desempenho do processo a partir de especificações analíticas na curva de resposta. Estudos similares foram conduzidos em [18, 19] por meio de algoritmos genéticos e em [20] pelo algoritmo *cuckoo search*, mas limitados a sistemas de primeira ou segunda ordem.

A estrutura deste trabalho está organizada da seguinte forma: a Seção 2 aborda a base teórica essencial para o entendimento do problema de sintonia e da solução proposta, partindo de conceitos básicos de sinais e sistemas, redigindo sobre a teoria de controle com realimentação e finalizando com métodos de resolução de equações não lineares; em seguida, a Seção 3 mostra o desenvolvimento algébrico que possibilitou a aplicação dos métodos numéricos e apresenta o algoritmo de sintonia proposto; na Seção 4, encontram-se os resultados dos testes realizados com o algoritmo em três funções de transferência com ordens diferentes, dispostos em tabelas com dados de performance do programa e em gráficos de convergência e das curvas de resposta

com o controlador sintonizado; por fim, na Seção 5, considerações finais são realizadas com base nos objetivos propostos e perspectivas de trabalhos futuros são apresentadas.

1.1 Objetivos

Objetivo Geral

Desenvolver um algoritmo de sintonia automática de controladores PID para sistemas lineares e invariantes no tempo com os métodos de Newton-Raphson e de Levenberg-Marquardt.

Objetivos Específicos

- Definir analiticamente funções de desempenho a partir de critérios sobre a curva de resposta ao degrau de um sistema;
- Obter expressões para os jacobianos das funções de desempenho com relação aos coeficientes do controlador;
- Formular um algoritmo que retorne os parâmetros do controlador sintonizados recebendo os dados do processo e os critérios de desempenho desejados como argumentos;
- Demonstrar a aplicabilidade, as vantagens e as desvantagens do algoritmo com testes em funções de transferência de diferentes ordens.

1.2 Justificativa

Preencher a lacuna existente no meio acadêmico quanto à aplicação de métodos numéricos de otimização para sintonia de controladores PID por critérios de desempenho como o tempo de subida, o tempo de pico, a ultrapassagem percentual e o tempo de assentamento em sistemas de qualquer ordem.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Modelos Matemáticos de Sistemas

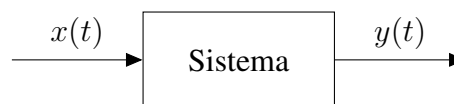
Segundo Haykin, um *senal* é uma função de uma ou mais variáveis que contém informações sobre a natureza de um fenômeno físico, enquanto um *sistema* é definido como uma entidade que produz sinais a partir da manipulação de outros sinais [21]. Sistemas físicos que evoluem com o tempo são chamados de sistemas dinâmicos, e essa atribuição é dada a um grande número de processos que interessam à engenharia de controle. Nesses sistemas, os sinais são modelados através de funções no tempo e o sistema em si é modelado por um conjunto de equações que estabelecem a relação entre os sinais de entrada e de saída.

Sistemas dinâmicos contínuos do tipo *Single-Input Single-Output* (SISO), ou seja, que recebem um único sinal como entrada e produzem um único sinal como saída, podem ser representados pela relação

$$x(t) \rightarrow y(t), \quad (2.1)$$

que associa cada sinal de entrada $x(t)$ a um sinal de saída $y(t)$, este último que também é chamado de *resposta* do sistema. Uma outra forma de caracterizar a relação entre esses dois sinais no tempo é através de um diagrama de blocos, como mostra a Figura 2.1.

Figura 2.1. Representação em diagrama de blocos de um sistema.



Fonte: Autoria própria.

O *modelo* de um sistema é uma equação matemática que quantifica as influências de cada uma das variáveis do sistema sobre as outras. Devido ao caráter dinâmico de sistemas físicos, tanto o valor instantâneo das variáveis de interesse quanto suas derivadas devem ser levadas em conta e, por esse motivo, a modelagem se dá por meio de equações diferenciais.

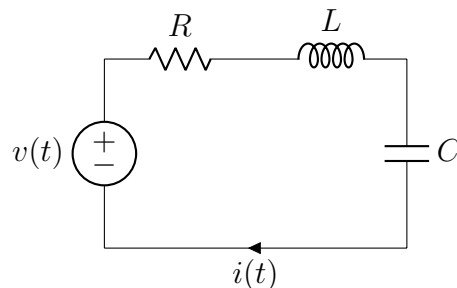
Os sistemas mais relevantes no contexto de sistemas de controle são aqueles que obedecem os princípios de linearidade e invariância no tempo ou que ao menos podem ser linearizados dentro de um intervalo específico. Neles, a transformada de Laplace pode ser utilizada para simplificar a manipulação das equações modeladas e extrair características acerca da resposta temporal. Além disso, todas as especificidades do processo físico em questão podem ser englobadas em uma função de transferência. As próximas seções irão abordar esses conceitos com mais detalhes.

2.1.1 Equações Diferenciais de Sistemas Elétricos

A dedução das equações que regem a natureza de um processo é feita utilizando as leis físicas do fenômeno, seja para sistemas mecânicos, elétricos, hidráulicos ou termodinâmicos [22]. A análise de sistemas elétricos, em específico, se baseia em grande parte nas Leis de Kirchhoff das tensões e das correntes, que estabelecem que a soma das tensões em uma malha e a soma das correntes em um nó devem ser iguais a zero. Essas ferramentas, quando combinadas com as propriedades eletromagnéticas conhecidas dos componentes circuitais, permitem a extração da equação diferencial ou sistema de equações diferenciais que ditam o comportamento das grandezas do circuito.

A Figura 2.2 mostra o diagrama de um circuito RLC, composto por uma fonte de excitação externa (fonte de tensão), um componente dissipador de energia (resistor), um componente de armazenamento indutivo (indutor) e um elemento de armazenamento capacitivo (capacitor).

Figura 2.2. Circuito RLC série.



Fonte: Autoria própria.

A principal propriedade de cada componente que se deve ter conhecimento para realizar análises em circuitos é a relação entre a tensão em seus terminais e a corrente que o atravessa. Nos componentes do circuito RLC, por exemplo: a tensão é proporcional à corrente no resistor; a tensão é proporcional à derivada da corrente no indutor; a corrente é proporcional à derivada da tensão no capacitor. Os coeficientes R , L e C são as constantes de proporcionalidade entre essas grandezas elétricas, chamadas de resistência, indutância e capacitância, respectivamente. Se o circuito for alimentado com uma tensão $v(t)$, pela Lei de Kirchhoff das tensões, a corrente de malha $i(t)$ satisfaz a equação

$$Ri(t) + L \frac{di(t)}{dt} + \frac{1}{C} \int_0^t i(\tau) d\tau = v(t),$$

que também pode ser escrita como uma equação diferencial de segunda ordem, na forma

$$L \frac{d^2 i(t)}{dt^2} + R \frac{di(t)}{dt} + \frac{1}{C} i(t) = \frac{dv(t)}{dt}. \quad (2.2)$$

O modelo matemático da Equação (2.2) contém, de maneira idealizada, todos os possíveis estados que as variáveis deste circuito podem assumir em um dado instante, mas vale ressaltar que diversos aspectos que modificam as características de circuitos na prática são desconsiderados, como perturbações externas e perdas por efeito Joule.

A resposta desse sistema irá depender de qual variável será considerada como saída do circuito, algo que pode ser atribuído à corrente que atravessa a malha ou à tensão em algum dos componentes, por exemplo. Se $i(t)$ for estabelecida como saída do sistema, a forma geral da resposta à entrada $v(t)$ será dada pelas soluções da equação diferencial em (2.2), que podem ser obtidas por métodos como o uso de fatores de integração ou o método dos coeficientes a determinar [22].

A solução de equações diferenciais lineares com coeficientes constantes possui duas componentes: $y_n(t)$, que é a solução geral da equação diferencial homogênea correspondente, e $y_f(t)$, que é uma solução particular. A resposta completa $y(t)$ é, então, escrita como a soma dessas duas funções,

$$y(t) = y_n(t) + y_f(t). \quad (2.3)$$

No contexto de circuitos elétricos, $y_n(t)$ é chamada de *resposta natural*, porque representa o comportamento da variável de saída quando nenhuma fonte externa de excitação é imposta, e $y_f(t)$ de *resposta forçada*, pois é produzida quando há aplicação de uma força externa [23].

Em circuitos modelados por uma equação diferencial de segunda ordem, como o da Figura 2.2, o comportamento da resposta natural será controlado pelos valores dos componentes passivos e como eles afetam a resolução da equação diferencial, e não sofrerá influência da fonte de tensão. A resposta forçada, em contrapartida, dependerá inteiramente do tipo de entrada $v(t)$ a qual o circuito é submetido.

2.1.2 Sistemas LTI e a Integral de Convolução

Um sistema é linear quando respeita os princípios da *aditividade* e da *homogeneidade*. Suponha um sistema conhecidamente linear caracterizado pelas relações de entrada/saída

$$x_1(t) \rightarrow y_1(t), \quad x_2(t) \rightarrow y_2(t).$$

A aditividade desse sistema estabelece que o efeito total de suas entradas pode ser determinado considerando uma entrada por vez, assumindo todas as outras entradas iguais a zero e somando os resultados. Ou seja,

$$x_1(t) + x_2(t) \rightarrow y_1(t) + y_2(t).$$

Por outro lado, a homogeneidade exige que a magnitude do fator de escala seja preservado ao passar pelo sistema:

$$kx_1(t) \rightarrow ky_1(t).$$

Essas duas propriedades podem ser combinadas em uma única, chamada de *superposição*, um princípio inerente a todos os sistemas lineares e caracterizado pela relação

$$k_1x_1(t) + k_2x_2(t) \rightarrow k_1y_1(t) + k_2y_2(t). \quad (2.4)$$

Sistemas cujos parâmetros não são alterados com o tempo são ditos *invariantes no tempo*. Nesses sistemas, um deslocamento no tempo no sinal de entrada produz o mesmo deslocamento no tempo no sinal de saída. Dessa forma, se

$$x(t) \rightarrow y(t)$$

para um sistema invariante no tempo, então,

$$x(t - t_0) \rightarrow y(t - t_0). \quad (2.5)$$

Quando um sistema respeita tanto o princípio da superposição quanto o princípio da invariância no tempo, ele é dito LTI (*Linear Time-Invariant*). O circuito RLC é um exemplo de sistema que entra nessa categoria. De fato, se cada componente que compõe um circuito é linear e invariante no tempo, com exceção de fontes independentes, então o circuito em si é

LTI [24]. Essa definição não se limita a circuitos elétricos, e em sua forma geral, sistemas LTI são modelados através de equações diferenciais ordinárias com coeficientes constantes [25]:

$$\frac{d^n y(t)}{dt^n} + a_1 \frac{d^{n-1} y(t)}{dt^{n-1}} + \dots + a_{n-1} \frac{dy(t)}{dt} + a_n y(t) = b_0 \frac{d^m x(t)}{dt^m} + b_1 \frac{d^{m-1} x(t)}{dt^{m-1}} + \dots + b_{m-1} \frac{dx(t)}{dt} + b_m x(t), \quad (2.6)$$

em que $a_1, \dots, a_n, b_0, \dots, b_m$ são constantes reais. A *ordem* dessa equação e, conseqüentemente, do sistema que ela representa, é definida pelo valor da derivada de maior ordem do sinal de saída presente, neste caso, n .

Com essas definições, torna-se possível elucidar a principal vantagem de trabalhar com sistemas LTI, em detrimento de sistemas que não cumprem os mesmos pré-requisitos. Um sinal contínuo no tempo pode ser decomposto como a combinação linear de sinais *elementares*, estes que são como blocos de construção para sinais mais complexos, dentre os quais estão as exponenciais e as senoides, por exemplo [21]. Ao tomar este sinal como entrada de um sistema LTI, o princípio da superposição dita que a resposta também será uma combinação linear e suas componentes serão a resposta do sistema LTI para o sinal elementar escolhido.

Seja $p_\Delta(t)$ a função pulso unitário definida da seguinte forma:

$$p_\Delta(t) = \begin{cases} \frac{1}{\Delta}, & \text{se } 0 \leq t \leq \Delta \\ 0, & \text{caso contrário.} \end{cases}$$

Essa definição é feita de maneira que a área delimitada pelo pulso é sempre igual a um, independentemente do comprimento de pulso Δ escolhido. Para alterar o instante em que o pulso se inicia, um deslocamento no tempo de $n\Delta$ unidades resulta na expressão

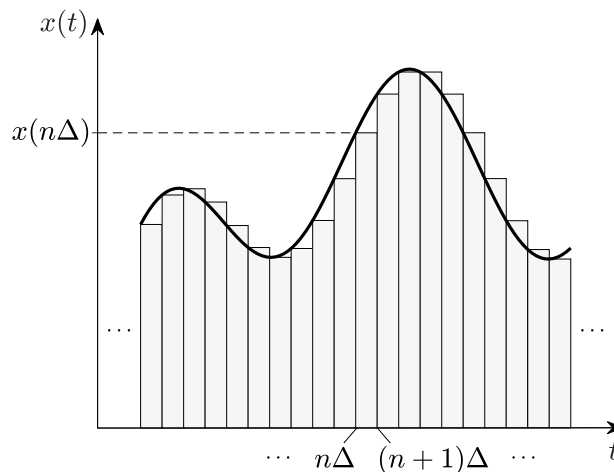
$$p_\Delta(t - n\Delta) = \begin{cases} \frac{1}{\Delta}, & \text{se } n\Delta \leq t \leq (n+1)\Delta \\ 0, & \text{caso contrário.} \end{cases}$$

Se $x(t)$ é um sinal contínuo arbitrário, ele pode ser aproximado em cada subintervalo $n\Delta \leq t \leq (n+1)\Delta$ para o seu valor na extremidade esquerda desse intervalo, o que possibilita o seguinte desenvolvimento:

$$\begin{aligned}
 x(t) &\approx x(n\Delta) \\
 &\approx x(n\Delta) \left(\frac{1}{\Delta}\right) \Delta \\
 &\approx x(n\Delta) p_{\Delta}(t - n\Delta) \Delta.
 \end{aligned}$$

Somando os resultados das aproximações de $x(t)$ em todos os subintervalos de comprimento Δ em $] -\infty, \infty[$, chega-se em uma aproximação desse sinal como uma soma de pulsos retangulares. O gráfico da Figura 2.3 ilustra essa decomposição.

Figura 2.3. Sinal contínuo aproximado como uma soma de pulsos retangulares de comprimento fixo.



Fonte: Autoria própria.

Quanto menor o comprimento dos pulsos, melhor a aproximação do sinal $x(t)$ por essa composição. Portanto, o valor exato de $x(t)$ em cada subintervalo é obtido no limite $\Delta \rightarrow 0$, onde cada pulso individual se torna uma função chamada de *impulso unitário*, também conhecida como função delta de Dirac, definida por

$$\delta(t) = \lim_{\Delta \rightarrow 0} p_{\Delta}(t) \implies \begin{cases} \delta(t) = 0, & \text{se } t \neq 0 \\ \int_{-\infty}^{\infty} \delta(t) = 1. \end{cases} \quad (2.7)$$

O impulso unitário também possui área unitária, mas ao contrário da função que a originou, ela está toda concentrada em um único ponto ao invés de um intervalo. Considerando $\tau = n\Delta$,

pode-se escrever $x(t)$ por meio da integral [25]

$$x(t) = \lim_{\Delta \rightarrow 0} \sum_{n=-\infty}^{\infty} x(n\Delta) p_{\Delta}(t - n\Delta) \Delta = \int_{-\infty}^{\infty} x(\tau) \delta(t - \tau) d\tau. \quad (2.8)$$

Agora, considere um sistema LTI que produz o sinal $y(t)$ como saída quando $x(t)$ é dado como entrada e seja $h_{\Delta}(t)$ a resposta do sistema para o pulso $p_{\Delta}(t)$. A partir da Equação (2.8), tem-se que a saída $y(t)$ é da forma [1, 25]

$$y(t) = \lim_{\Delta \rightarrow 0} \sum_{n=-\infty}^{\infty} x(n\Delta) h_{\Delta}(t - n\Delta) \Delta,$$

pois sabe-se que $x(n\Delta)$ é constante e $h_{\Delta}(t - n\Delta)$ é saída de $p_{\Delta}(t - n\Delta)$. Quando $\Delta \rightarrow 0$, a função $h_{\Delta}(t)$ se torna a *resposta ao impulso*, denotada por $h(t)$. Essa equação também pode ser escrita na forma integral e será chamada de *integral de convolução*, um operador linear que também é comumente representado pelo símbolo $*$.

$$y(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(\tau) h(t - \tau) d\tau \quad (2.9)$$

Assim, se uma forma de obter a resposta ao impulso de um dado sistema LTI for determinada, a resposta para qualquer sinal de entrada contínuo $x(t)$ será obtida através da convolução deste sinal com $h(t)$.

2.1.3 Transformada de Laplace

A *transformada de Laplace* é um operador que converte uma função no tempo em uma função na variável s , sendo s um número complexo. Ela é definida por

$$X(s) = \mathcal{L}\{x(t)\} = \int_{0^-}^{\infty} x(t) e^{-st} dt \quad (2.10)$$

em sua forma unilateral, e por

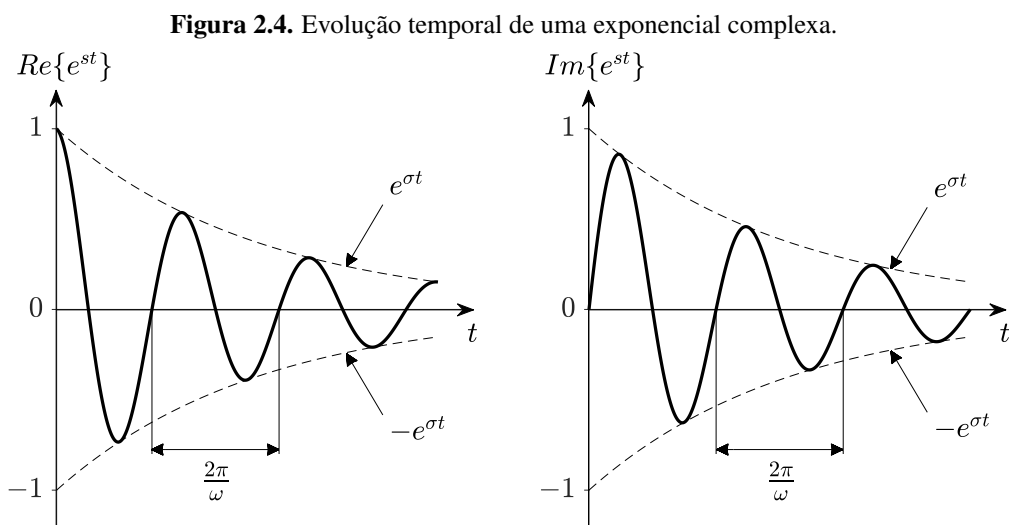
$$X(s) = \int_{-\infty}^{\infty} x(t) e^{-st} dt \quad (2.11)$$

em sua forma bilateral, em que a única diferença entre elas está no intervalo de integração. A notação para o limite de integração inferior na versão unilateral significa que a transformada do sinal pode ser determinada mesmo se $x(t)$ for descontínua em $t = 0$, vide a função impulso unitário [26, 27].

Para entender a aplicabilidade da transformada de Laplace em sistemas LTI, considere o sinal exponencial complexo e^{st} , com $s = \sigma + j\omega$, que pode ser expandido como

$$e^{st} = e^{\sigma t} \cos(\omega t) + je^{\sigma t} \sin(\omega t). \quad (2.12)$$

O coeficiente σ é chamado de *fator de amortecimento*, pois é responsável por delimitar o quão rápido o envelope exponencial, dentro do qual o sinal oscila, decai (ou aumenta) com o tempo, enquanto ω é a *frequência de oscilação* dos fatores senoidais. O gráfico da Figura 2.4 expõe o comportamento temporal das partes real e imaginária desse sinal.



Fonte: Autoria própria.

Agora, tome e^{st} como entrada de um sistema LTI arbitrário \mathcal{H} , com resposta ao impulso $h(t)$. A saída correspondente, a partir da Equação (2.9), é:

$$\begin{aligned} y(t) &= \int_{-\infty}^{\infty} h(\tau) e^{s(t-\tau)} d\tau \\ &= e^{st} \int_{-\infty}^{\infty} h(\tau) e^{-s\tau} d\tau. \end{aligned}$$

Percebe-se que essa integral é a transformada de Laplace da resposta ao impulso, logo, denotando-a por $H(s)$, obtém-se

$$y(t) = \mathcal{H}\{e^{st}\} = H(s)e^{st}. \quad (2.13)$$

Então, e^{st} é uma *autofunção* do sistema, enquanto a transformada de Laplace $H(s)$ é um *autovalor* do sistema [21].

As ações proporcionadas sobre a exponencial complexa podem ser melhor analisadas ao

expressar a transformada na forma polar $H(s) = |H(s)|e^{j \text{Arg}\{H(s)\}}$. A saída é reescrita como

$$y(t) = |H(s)|e^{\sigma t} \cos[\omega t + \text{Arg}\{H(s)\}] + j|H(s)|e^{\sigma t} \text{sen}[\omega t + \text{Arg}\{H(s)\}].$$

Portanto, ao passar por um sistema LTI, um sinal exponencial complexo tem sua magnitude multiplicada por um fator $|H(s)|$ e sua fase deslocada por um fator $\text{Arg}\{H(s)\}$, mas seu coeficiente de amortecimento σ e sua frequência de oscilação ω não são alterados.

De forma similar à decomposição de sinais como uma soma de impulsos unitários feita anteriormente, sinais podem ser representados como uma superposição ponderada de exponenciais complexas, afinal, a exponencial é um sinal elementar. A análise da resposta do sistema, então, torna-se subjugada à transformada de Laplace da resposta ao impulso. Essa expressão desejada pode ser deduzida analiticamente, a partir de (2.10), e toma a forma

$$x(t) = \mathcal{L}^{-1}\{X(s)\} = \frac{1}{2\pi j} \int_{\sigma-j\infty}^{\sigma+j\infty} X(s)e^{st} ds, \quad (2.14)$$

chamada de *transformada inversa de Laplace*.

Levando em consideração a resposta a uma exponencial complexa da Equação (2.13) e as definições em (2.10) e (2.14), é possível demonstrar que a resposta do sistema para qualquer sinal de entrada é calculada pela transformada inversa do produto das transformadas de $x(t)$ e $h(t)$, como segue:

$$\begin{aligned} y(t) &= \mathcal{H}\{x(t)\} \\ &= \mathcal{H}\left\{\frac{1}{2\pi j} \int_{\sigma-j\infty}^{\sigma+j\infty} X(s)e^{st} ds\right\} \\ &= \frac{1}{2\pi j} \int_{\sigma-j\infty}^{\sigma+j\infty} X(s)\mathcal{H}\{e^{st}\} ds \\ &= \frac{1}{2\pi j} \int_{\sigma-j\infty}^{\sigma+j\infty} X(s)H(s)e^{st} ds \\ &= \mathcal{L}^{-1}\{X(s)H(s)\}. \end{aligned} \quad (2.15)$$

Essa ideia pode ser generalizada para a convolução de dois sinais quaisquer, indicando que uma convolução no domínio do tempo é convertida em um produto no domínio de Laplace.

Outra propriedade extremamente útil dessa transformada permite determinar o valor de uma função $x(t)$ em zero e no infinito a partir de sua transformada $X(s)$, mesmo se a expressão

para $x(t)$ for desconhecida. Isso é realizado por meio do *teorema do valor inicial*,

$$x(0^+) = \lim_{s \rightarrow \infty} sX(s), \quad (2.16)$$

e do *teorema do valor final*,

$$\lim_{t \rightarrow \infty} x(t) = \lim_{s \rightarrow 0} sX(s). \quad (2.17)$$

Esses são apenas alguns exemplos dentre as diversas simplificações que surgem a partir das propriedades da transformada de Laplace. Para mais fórmulas, veja [21, 25].

2.1.4 Funções de Transferência

As ferramentas matemáticas apresentadas até aqui serão utilizadas para obter a resposta ao impulso de um sistema LTI a partir de sua equação diferencial. A propriedade da diferenciação da transformada de Laplace constata que

$$\mathcal{L} \left\{ \frac{d^n x(t)}{dt^n} \right\} = s^n X(s) - \sum_{k=1}^n s^{n-k} x^{(k-1)}(0^-). \quad (2.18)$$

Pode-se ver que a transformada converte uma diferenciação no domínio do tempo em um polinômio mônico em função de s com grau igual à ordem da derivada. Desse modo, a aplicação da transformada unilateral na Equação (2.6) resulta na equação algébrica

$$(s^n + a_1 s^{n-1} + \dots + a_{n-1} s + a_n)Y(s) + Y_0 = (b_0 s^m + b_1 s^{m-1} + \dots + b_{m-1} s + b_m)X(s) + X_0.$$

X_0 e Y_0 são combinações de todas as condições iniciais dos sinais $x(t)$ e $y(t)$, respectivamente, que surgem a partir da propriedade da diferenciação. Essas condições iniciais serão assumidas como nulas, pois se não o forem, o sistema se torna não linear [25]. Então,

$$\frac{Y(s)}{X(s)} = \frac{b_0 s^m + b_1 s^{m-1} + \dots + b_{m-1} s + b_m}{s^n + a_1 s^{n-1} + a_2 s^{n-2} + \dots + a_{n-1} s + a_n}. \quad (2.19)$$

A razão entre as transformadas da saída e da entrada é chamada de *função de transferência* do sistema. Essa expressão pode ser considerada uma notação simplificada para a equação diferencial, pois esta pode ser alcançada a partir da função de transferência por meio da transformada inversa de Laplace. Então, $Y(s)/X(s)$ representa uma descrição completa das características de entrada/saída de um sistema, independentemente da função de entrada [25].

Entretanto, pela Equação (2.15), pode-se deduzir que

$$Y(s) = H(s)X(s) \implies H(s) = \frac{Y(s)}{X(s)},$$

mostrando que a transformada de Laplace da resposta ao impulso de um sistema é igual à função de transferência desse sistema.

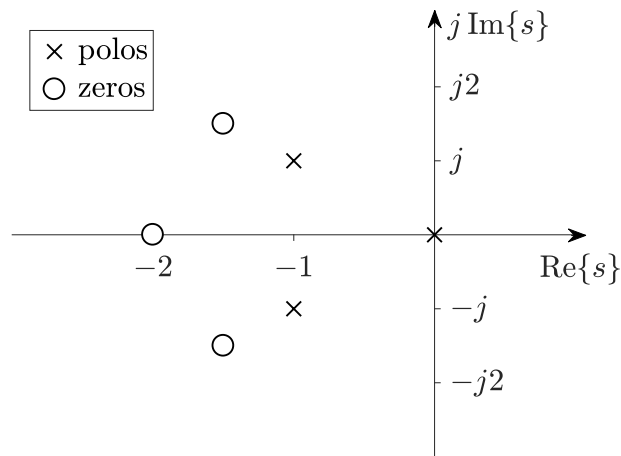
A função de transferência pode ser categorizada em três tipos, com base nos graus dos polinômios do denominador e do numerador: $H(s)$ será *própria* se $n \geq m$, estritamente própria se $n > m$ e *imprópria* se $n < m$. Existem motivos para restringir os estudos para funções de transferências estritamente próprias, dentre os quais pode-se citar o fato de que, em sistemas físicos, uma função de transferência imprópria implicaria na possibilidade de obter uma resposta infinita a uma entrada finita [1]. Logo, a função de transferência de um sistema LTI geral é a seguinte:

$$H(s) = \frac{b_0s^n + b_1s^{n-1} + b_2s^{n-2} + \dots + b_{n-1}s + b_n}{s^n + a_1s^{n-1} + a_2s^{n-2} + \dots + a_{n-1}s + a_n}. \quad (2.20)$$

Denota-se por $N(s)$ o polinômio do numerador e por $D(s)$ o polinômio do denominador de $H(s)$. Por se tratarem de funções polinomiais, elas podem ser escritas como o produto dos fatores primos correspondentes a cada uma de suas raízes, de acordo com o teorema fundamental da álgebra. Ademais, os coeficientes de $N(s)$ e $D(s)$ são todos reais, então deve haver uma simetria conjugada nesses conjuntos de raízes (ou são reais ou estão em pares de complexos conjugados) [24]. Essa decomposição será bastante útil ao realizar a inversão da transformada mais à frente. Assim, para $m \leq n$, tem-se

$$H(s) = \frac{N(s)}{D(s)} = \frac{K(s - z_1) \cdots (s - z_m)}{(s - p_1) \cdots (s - p_n)}. \quad (2.21)$$

z_1, \dots, z_m são as raízes de $N(s)$, e serão chamadas de *zeros* de $H(s)$, enquanto p_1, \dots, p_n são as raízes de $D(s)$, e serão chamados de *polos* de $H(s)$. A Figura 2.5 apresenta uma maneira de ilustrar uma função de transferência através da disposição de seus polos e zeros no plano complexo. O posicionamento desses termos terão influências diretas nas características transitórias e de estado estacionário da resposta do sistema.

Figura 2.5. Diagrama de polos e zeros no plano s .

Fonte: Autoria própria.

2.1.5 Resposta no Domínio do Tempo

O passo final para caracterizar um sistema LTI é determinar a sua resposta no domínio do tempo, ou seja, busca-se uma expressão para $y(t)$. Já se sabe que sua transformada é calculada pelo produto da função de transferência desse sistema com a transformada do sinal de entrada $X(s)$. Todavia, a aplicação direta da transformada inversa em $Y(s)$ pode ser bastante complicada, pois requer o cálculo de uma integral de linha no plano complexo. Por conta disso, a inversão de grande parte das funções é determinada com base nas propriedades da transformada de Laplace e/ou em uma tabela contendo algumas relações de transformação conhecidas.

Considere a função de transferência de sistemas LTI em (2.20). Se seus polos forem dois a dois distintos, essa função racional pode ser expandida em frações parciais da seguinte maneira:

$$\begin{aligned}
 Y(s) &= H(s)X(s) \\
 &= \frac{b_1 s^{n-1} + b_2 s^{n-2} + \dots + b_{n-1} s + b_n}{(s - p_1) \dots (s - p_n)} X(s) \\
 &= \frac{r_1}{s - p_1} + \dots + \frac{r_n}{s - p_n} + Y_f(s).
 \end{aligned} \tag{2.22}$$

O coeficiente complexo r_k é o *resíduo* associado ao polo p_k , com $k = 1, \dots, n$ e $Y_f(s)$ representa a soma dos termos dessa expansão originados dos polos do sinal de entrada $X(s)$.

Agora, torna-se fácil a aplicação da transformada inversa conforme a relação de transformação

$$\mathcal{L}\{r_k e^{p_k t} u(t)\} = \frac{r_k}{s - p_k}, \tag{2.23}$$

em que $u(t)$ é a função *degrau unitário* ou função de Heaviside, definida por

$$u(t) = \begin{cases} 1, & \text{se } t > 0 \\ 1/2, & \text{se } t = 0 \\ 0, & \text{se } t < 0. \end{cases} \quad (2.24)$$

Deste modo, a resposta será dada por

$$\begin{aligned} y(t) &= \mathcal{L}^{-1} \left\{ \frac{r_1}{s - p_1} + \cdots + \frac{r_n}{s - p_n} + Y_f(s) \right\} \\ &= (r_1 e^{p_1 t} + \cdots + r_n e^{p_n t}) u(t) + y_f(t), \end{aligned} \quad (2.25)$$

em que $y_f(t)$ é a transformada inversa de $Y_f(s)$. Vale lembrar que, no contexto a partir do qual esse procedimento se desdobrou, $y(t)$ é a solução da equação diferencial (2.6) e assim como o exemplo que foi dado para a resposta de um circuito RLC, ela poderá ser desmembrada em uma resposta natural e uma resposta forçada. Vê-se então que $y_f(t)$ em (2.25) é a resposta forçada do sistema, pois surge como consequência da aplicação de uma excitação externa (sinal de entrada), enquanto a soma dos outros termos compõem a resposta natural do sistema.

Com a aplicação da transformada inversa, o problema de resolver uma equação diferencial é reescrito como um problema de resolver uma equação algébrica, visto que basta determinar as raízes de uma equação polinomial de ordem n e os resíduos das frações parciais para chegar na resposta. Por conta disso, a transformada de Laplace é extensivamente utilizada para resolver equações diferenciais lineares com coeficientes constantes.

Os cálculos realizados para $y_f(t)$ dependerão unicamente do sinal de entrada e serão discutidos posteriormente. Por agora, o foco está na obtenção da resposta natural $y_n(t)$. Sistemas de primeira ordem têm apenas um polo, em que o mesmo deve ser necessariamente real, então se sabe que sua resposta natural será uma exponencial crescente ou decrescente. As ramificações no tipo de resposta começam a aparecer em sistemas de segunda ordem ou maior, portanto, o caso mais simples que se encaixa nessa descrição será analisado a seguir.

Considere um sistema de segunda ordem com polos p_1 e p_2 distintos entre si. A transformada da componente natural da resposta desse sistema será

$$Y_n(t) = \frac{r_1}{s - p_1} + \frac{r_2}{s - p_2}. \quad (2.26)$$

Primeiramente, se os polos são complexos conjugados, eles podem ser escritos na forma retangular $p_1 = \sigma + j\omega$ e $p_2 = \sigma - j\omega$, com σ e ω reais e $\omega \neq 0$. Ao realizar o processo inverso e juntar as frações parciais, a única maneira de obter um polinômio com coeficientes reais no numerador é se os resíduos também forem complexos conjugados, ou seja, $r_2 = \bar{r}_1$. No domínio do tempo, chega-se na expressão

$$y_n(t) = [r_1 e^{(\sigma+j\omega)t} + \bar{r}_1 e^{(\sigma-j\omega)t}] u(t). \quad (2.27)$$

Decompondo as exponenciais complexas como em (2.12):

$$\begin{aligned} y_n(t) &= [r_1 e^{(\sigma+j\omega)t} + \bar{r}_1 e^{(\sigma-j\omega)t}] u(t) \\ &= \{(\operatorname{Re}\{r_1\} + j \operatorname{Im}\{r_1\}) e^{\sigma t} [\cos(\omega t) + j \operatorname{sen}(\omega t)] \\ &\quad + (\operatorname{Re}\{r_1\} - j \operatorname{Im}\{r_1\}) e^{\sigma t} [\cos(\omega t) - j \operatorname{sen}(\omega t)]\} u(t) \\ &= \{2e^{\sigma t} [(\operatorname{Re}\{r_1\}) \cos(\omega t) - (\operatorname{Im}\{r_1\}) \operatorname{sen}(\omega t)]\} u(t). \end{aligned}$$

Utilizando a propriedade trigonométrica

$$\begin{cases} X_1 \cos \alpha - X_2 \operatorname{sen} \alpha = X_3 \cos(\alpha + \beta) \\ X_3 = \sqrt{X_1^2 + X_2^2} \\ \beta = \arctan\left(\frac{X_2}{X_1}\right), \end{cases} \quad (2.28)$$

fazendo $X_1 = \operatorname{Re}\{r_1\}$, $X_2 = \operatorname{Im}\{r_1\}$ e $\alpha = \omega t$, obtém-se

$$y_n(t) = [2|r_1| e^{\sigma t} \cos(\omega t + \operatorname{Arg}\{r_1\})] u(t). \quad (2.29)$$

O termo senoidal indica que a parte imaginária dos polos dá origem a oscilações no domínio do tempo, enquanto o sinal de σ determinará o formato do envelope exponencial. Se $\sigma < 0$, a amplitude da senoide sofre um decaimento exponencial e o sistema será *subamortecido*. Se $\sigma = 0$, o sinal se torna puramente senoidal e o sistema é *não amortecido*. Por último, quando $\sigma > 0$, a resposta do sistema aumenta sua amplitude exponencialmente com o tempo, logo é *instável*. A definição de estabilidade será formalizada mais à frente.

A segunda possibilidade surge quando os polos podem ser escritos como $p_1 = \sigma_1$ e

$p_2 = \sigma_2$, em que σ_1 e σ_2 são valores reais e distintos. A resposta natural, neste caso, será

$$y_n(t) = (r_1 e^{\sigma_1 t} + r_2 e^{\sigma_2 t}) u(t). \quad (2.30)$$

Nota-se, em (2.26), que os resíduos também devem ser valores reais. Se σ_1 e σ_2 são ambos negativos, a resposta será dada pela soma de duas exponenciais decrescentes, que resulta no caso *superamortecido*. Se ao menos um dos polos for positivo, a resposta será instável.

Na terceira possibilidade, tem-se $p_1 = p_2 = \sigma$, com σ necessariamente real. Como há a ocorrência de polos repetidos, a expansão por frações parciais assume um formato diferente,

$$Y_n(t) = \frac{r_1}{s + \sigma} + \frac{r_2}{(s + \sigma)^2}. \quad (2.31)$$

É possível mostrar que, em geral, quando $H(s)$ tem n polos repetidos,

$$\frac{N(s)}{(s - p)^n} = \frac{r_1}{s - p} + \frac{r_2}{(s - p)^2} + \dots + \frac{r_n}{(s - p)^n}, \quad (2.32)$$

e no domínio do tempo, aplicando a transformada inversa nessa expressão, tem-se

$$\mathcal{L}^{-1} \left\{ \frac{N(s)}{(s - p)^n} \right\} = \sum_{k=1}^n \frac{r_k}{(k - 1)!} t^{k-1} e^{pt}. \quad (2.33)$$

Assim sendo, a resposta natural para dois polos repetidos é

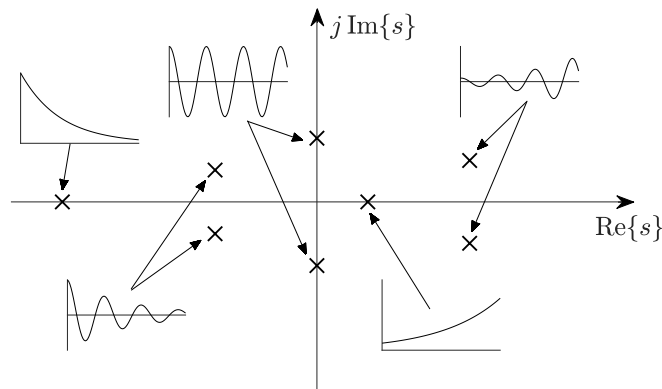
$$y_n(t) = [(r_1 + r_2 t) e^{\sigma t}] u(t), \quad (2.34)$$

que é o caso *criticamente amortecido* quando σ for negativo.

De acordo com a definição *Bounded-Input Bounded-Output* (BIBO) de estabilidade, um sistema será *estável* quando uma entrada limitada der origem a uma saída limitada. Evidentemente, em todos os cenários possíveis para a resposta natural de sistemas de segunda ordem, as exponenciais crescentes em $y_n(t)$ são fontes de instabilidade. Uma condição necessária e suficiente para garantir a estabilidade de um sistema LTI é que os polos de sua função de transferência estejam todos localizados no semiplano esquerdo, pois caso contrário, um sinal limitado pode resultar em uma resposta ilimitada [24,25,27]. Pela tamanha importância da equação $D(s) = 0$, cujas raízes são os polos de $H(s)$, dá-se um nome especial para a mesma: *equação característica do sistema*.

Estende-se essa noção de estabilidade para sistemas de ordem maior que dois ao notar que sua resposta natural será composta pela soma das componentes naturais de subsistemas de primeira e segunda ordem, desde que não haja a presença de três ou mais polos repetidos. A Figura 2.6 apresenta a contribuição de diferentes configurações de posicionamento dos polos para a resposta do sistema.

Figura 2.6. Resposta do sistema com base na localização dos polos.



Fonte: Autoria própria.

2.2 Controle PID

Controladores são sistemas projetados a partir da modelagem matemática de um processo que devem operar de forma a fornecer uma saída desejada para uma entrada específica. Um sistema de controle efetivo deve: certificar que os polos da função de transferência estejam no semiplano esquerdo para evitar instabilidades; fazer com que a saída do processo siga a referência o tanto quanto possível; rejeitar distúrbios e minimizar o erro; ter baixa sensibilidade a mudanças nos parâmetros do processo [1].

Os sistemas de controle em malha aberta são um tipo de configuração em que o sinal de saída é gerado diretamente a partir do sinal de entrada, enquanto sistemas de controle em malha fechada realizam uma retropropagação do sinal de saída para ações de correção e oferecem uma alternativa mais confiável e robusta com relação à malha aberta. O desempenho dos controladores pode ser avaliado submetendo-os a entradas de teste padronizadas e estudando as características da resposta temporal. Os critérios de desempenho mais comumente aplicados costumam trabalhar em cima da resposta ao degrau do sistema.

O controlador PID é o algoritmo de controle mais comum na indústria e combina as ações de três blocos operacionais distintos (proporcional, integral e derivativo) para promover melhorias na resposta transitória e no estado estacionário de um processo. O procedimento para

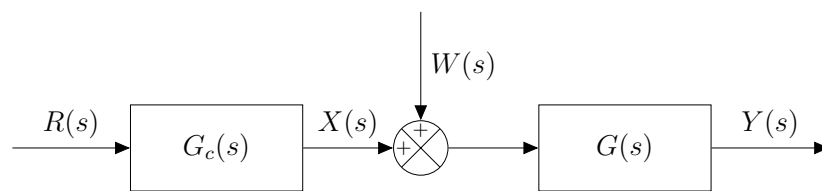
a determinação dos parâmetros do controlador PID, conhecido como sintonia do controlador, parte das especificações de desempenho exigidas para um dado processo. Os métodos de sintonia existentes variam muito de um para outro a depender do conhecimento que eles exigem do processo dinâmico.

Como será mostrado, em um sistema de controle em malha fechada atuando sobre um processo LTI do tipo SISO e com um controlador PID, certos critérios de desempenho podem ser estipulados e definidos analiticamente na curva de resposta. Dessa forma, obtém-se um sistema de equações não lineares que descrevem o quão próximo o sistema está do comportamento desejado.

2.2.1 Sistemas de Controle em Malha Aberta e em Malha Fechada

O controle em malha aberta está presente em um sistema quando as ações do controlador não são influenciadas pelo valor de saída. Noutras palavras, esses sistemas operam sem realimentação e geram diretamente a saída em resposta a um sinal de entrada [22]. Um sistema de controle em malha aberta consiste de um controlador $G_c(s)$ que atua em um sinal $R(s)$ chamado de *entrada de referência*, e cujo objetivo é tornar a saída a mais próxima possível de $R(s)$. O sinal $X(s)$, chamado de *sinal de controle*, é fornecido pelo controlador ao processo $G(s)$ junto a um sinal de perturbação $W(s)$ que surge naturalmente em qualquer caso prático. A Figura 2.7 ilustra esse sistema de controle em um diagrama de blocos.

Figura 2.7. Diagrama de um sistema de controle em malha aberta.



Fonte: Autoria própria.

A saída $Y(s)$ do sistema da Figura 2.7 é dada por

$$\begin{aligned} Y(s) &= [X(s) + W(s)]G(s) \\ &= R(s)G_c(s)G(s) + W(s)G(s). \end{aligned} \quad (2.35)$$

Visto que é desejável reduzir a diferença entre a referência e a saída com a implementação desse sistema de controle, pode-se definir o *erro de rastreamento* como um modo de avaliar a performance do controlador. No caso do controle em malha aberta, esse erro é calculado da

seguinte maneira:

$$\begin{aligned} E(s) &= R(s) - Y(s) \\ &= R(s)[1 - G_c(s)G(s)] - W(s)G(s). \end{aligned} \quad (2.36)$$

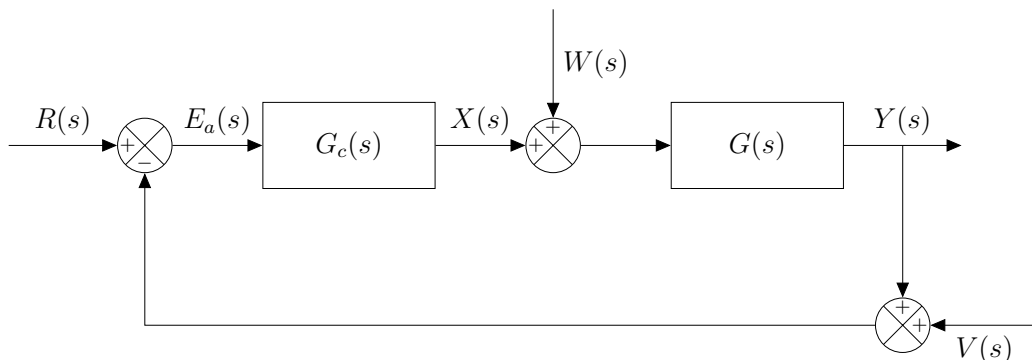
A função de transferência de malha aberta, também conhecida como *ganho de malha aberta*, pode ser obtida desconsiderando o sinal de perturbação e analisando a razão entre a saída e a entrada do sistema, como segue:

$$H(s) = \frac{Y(s)}{R(s)} = G_c(s)G(s). \quad (2.37)$$

O controlador $G_c(s)$ deve atuar de forma a rejeitar a perturbação $W(s)$ e impedir que ela atrapalhe a saída. Porém, percebe na Equação (2.36) que a influência da perturbação no erro de rastreamento age de forma independente de $G_c(s)$, e se $W(s)$ for grande o suficiente, não há nada que o controlador possa fazer para amenizar os efeitos negativos na saída. Esse é um dos diversos motivos que torna a implementação prática desse tipo de controle inviável.

A alternativa é usar um controle em malha fechada, onde o sistema possui um sensor que mede o valor instantâneo da saída e realimenta esse sinal para realizar uma ação de correção e tornar a saída igual ao sinal de referência. Neste caso, surge um novo sinal de perturbação $V(s)$ que interfere com a medição do sensor utilizado, como mostra o diagrama da Figura 2.8.

Figura 2.8. Diagrama de um sistema de controle em malha fechada.



Fonte: Autoria própria.

O princípio da superposição pode ser aplicado nesse sistema para obter a saída $Y(s)$ através da soma das influências individuais de cada entrada, obtendo [1, 22]

$$Y(s) = \frac{G_c(s)G(s)}{1 + G_c(s)G(s)}R(s) + \frac{G(s)}{1 + G_c(s)G(s)}W(s) - \frac{G_c(s)G(s)}{1 + G_c(s)G(s)}V(s). \quad (2.38)$$

O erro de rastreamento, então, será

$$\begin{aligned} E(s) &= R(s) - Y(s) \\ &= \frac{1}{1 + G_c(s)G(s)}R(s) - \frac{G(s)}{1 + G_c(s)G(s)}W(s) + \frac{G_c(s)G(s)}{1 + G_c(s)G(s)}V(s). \end{aligned} \quad (2.39)$$

Fazendo-se $W(s) \approx 0$ e $V(s) \approx 0$, obtém-se a função de transferência em malha fechada ou *ganho de malha fechada*,

$$H(s) = \frac{Y(s)}{R(s)} = \frac{G_c(s)G(s)}{1 + G_c(s)G(s)}. \quad (2.40)$$

A realimentação negativa amplia as possibilidades de ajuste no controlador para a contenção das perturbações. Na Equação (2.39), deduz-se que para reduzir a influência de $W(s)$, o ganho de malha aberta $G(s)G_c(s)$ deve ser o tão grande quanto possível. Ainda assim, o ruído $V(s)$ do sensor continuaria presente devido ao termo $G_c(s)G(s)$ no numerador da fração que o multiplica. Se uma análise mais a fundo for feita acerca das frequências presentes em cada uma dessas funções, revela-se que, de modo geral, sensores são mais susceptíveis a ruídos em altas frequências, enquanto os distúrbios da planta tendem a ocorrer em baixas frequências. Com isso, deve-se projetar $G_c(s)$ tal que o ganho de malha aberta seja maior em frequências baixas para suprimir $W(s)$, e menor em frequências altas para suprimir $V(s)$ [1].

Outra diferença importante entre esses sistemas de controle é na questão da estabilidade. Por (2.37), é evidente que quaisquer polos do processo $G(s)$ também serão polos da função de transferência em malha aberta, indicando que um processo instável operando sob esse tipo de controle resultará em uma resposta igualmente instável. No controle em malha fechada, porém, isso não é necessariamente verdade. Sejam $a(s)$, $b(s)$, $c(s)$ e $d(s)$ polinômios de forma que $G(s) = b(s)/a(s)$ e $G_c(s) = c(s)/d(s)$. Então, a equação característica da função de transferência em malha fechada é

$$\begin{aligned} 1 + G_c(s)G(s) &= 0 \\ 1 + \frac{b(s)c(s)}{a(s)d(s)} &= 0 \\ a(s)d(s) + b(s)c(s) &= 0. \end{aligned}$$

Essa equação mostra que é possível manipular os atributos do controlador de forma a eliminar os polos instáveis do sistema. Isto é, a presença de polos de $G(s)$ no semiplano direito não o impede de ser estabilizado pelo controle em malha fechada.

2.2.2 Critérios de Desempenho no Domínio do Tempo

Durante e após o projeto de sistemas de controle, testes devem ser realizados para verificar se os critérios de desempenho necessários para a operação da planta estão sendo atendidos ou podem ser otimizados com ajustes nos parâmetros do controlador. Isso é feito submetendo o sistema a uma entrada de teste e observando o comportamento da resposta para essa excitação. A função a ser utilizada como entrada de teste é de escolha do projetista, mas geralmente são escolhidas funções padronizadas seja qual for a aplicação, como impulsos, degraus, rampas, parábolas e senoides [26]. A Tabela 2.1 mostra os principais sinais de teste utilizados e as características que podem ser extraídas a partir da resposta do sistema.

Testes de desempenho padrões são normalmente definidos em termos da resposta ao degrau de um sistema, por ser o sinal mais fácil de ser gerado e avaliado [22]. A função degrau unitário, como definida em (2.24), tem $1/s$ como transformada de Laplace. Assim, a resposta ao degrau de um sistema com função de transferência $H(s)$ é

$$Y(s) = \frac{H(s)}{s}. \quad (2.41)$$

Aplicando o teorema do valor inicial da Equação (2.16) na resposta ao degrau:

$$y(0^+) = \lim_{s \rightarrow \infty} sY(s).$$

Mas $sY(s) = H(s)$ e como todas as condições iniciais de $y(t)$ devem ser nulas para conservar a linearidade do sistema, tem-se que

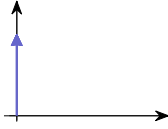
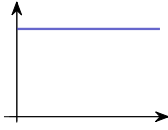
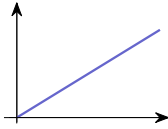
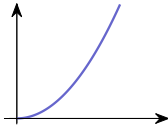
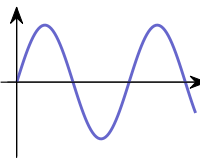
$$y(0^+) = \lim_{s \rightarrow \infty} H(s) = 0. \quad (2.42)$$

$H(s)$ é uma função racional e seu limite no infinito só será zero quando o grau do polinômio do denominador for maior que o grau do polinômio do numerador. Então, se $y(t)$ corresponde à saída de um sistema LTI, a função de transferência do sistema de controle deve ser estritamente própria.

A resposta ao degrau adiciona um polo na origem em $H(s)$, e ao fazer a decomposição em frações parciais, será obtido

$$Y(s) = Y_n(s) + \frac{C}{s},$$

Tabela 2.1. Formas de onda de teste utilizadas em sistemas de controle.

Entrada	Função	Esboço	Utilização
Impulso	$\delta(t)$		Resposta transitória, Modelagem
Degrau	$u(t)$		Resposta transitória, Erro em regime permanente
Rampa	$tu(t)$		Erro em regime permanente
Parábola	$\frac{1}{2}t^2u(t)$		Erro em regime permanente
Senoide	$\text{sen}(\omega t)u(t)$		Resposta transitória, Modelagem, Erro em regime permanente

Fonte: Nise [26].

em que C é uma constante real cujo valor dependerá do comportamento da resposta no estado estacionário. C/s é a componente forçada da resposta ao degrau e, no domínio do tempo, tem-se $y_f(t) = Cu(t)$.

O erro de estado estacionário e_{ss} é o valor de convergência do erro de rastreamento, e em um processo operando sob um sistema de controle em malha fechada, será calculado por

$$e_{ss} = \lim_{t \rightarrow \infty} e(t) = \lim_{t \rightarrow \infty} [r(t) - y(t)].$$

Supondo a ausência de polos instáveis em $H(s)$ e que o controlador atua de forma que $e_{ss} = 0$, a componente natural da resposta será composta por uma soma de exponenciais decrescentes, e o valor de $y(t)$ no infinito será determinado unicamente pela componente forçada. Então, fazendo-se $r(t) = u(t)$,

$$\lim_{t \rightarrow \infty} [u(t) - Cu(t)] = 0 \implies C = 1. \quad (2.43)$$

Diante disso, tem-se que $y_f(t) = u(t)$ é a componente de estado estacionário (ou forçada) da resposta ao degrau.

Os principais critérios de desempenho usados para avaliar o sistema de controle pela curva de resposta ao degrau são: o *tempo de subida*, o *tempo de pico*, a *ultrapassagem percentual* e o *tempo de assentamento*. Nas definições a seguir, $y(t)$ é a resposta ao degrau do sistema e y_{ss} é o valor de saída no estado estacionário (igual a 1, quando $e_{ss} = 0$).

O tempo de subida (t_r) é o tempo que o sistema leva para atingir as proximidades de seu valor de referência. Frequentemente é definida de duas maneiras: se o sistema for subamortecido e, portanto, possuir uma máxima ultrapassagem, t_r é medido entre 0% e 100% da referência; já se o sistema for superamortecido, logo sem ultrapassagem, mede-se t_r entre 10% e 90% da referência [22]. No primeiro caso, t_r é o instante em que a saída atinge o valor de estado estacionário, portanto, pode ser caracterizado pela equação

$$y(t_r) = y_{ss}. \quad (2.44)$$

O tempo de pico (t_p) é o instante em que a resposta atinge seu valor máximo e satisfaz a equação

$$y'(t_p) = 0. \quad (2.45)$$

A ultrapassagem percentual (M_p), também chamada de *overshoot*, é o valor pelo qual a forma de onda ultrapassa o valor em regime permanente no tempo de pico, expresso como uma porcentagem do valor em regime permanente. Assim, tem-se

$$M_p = \left(\frac{y(t_p) - y_{ss}}{y_{ss}} \right) \times 100\%. \quad (2.46)$$

Por fim, o tempo de assentamento (t_s) é o tempo necessário para os transitórios do sistema terminarem e satisfaz

$$y(t_s) = \left(1 \pm \frac{\alpha}{100\%} \right) y_{ss}. \quad (2.47)$$

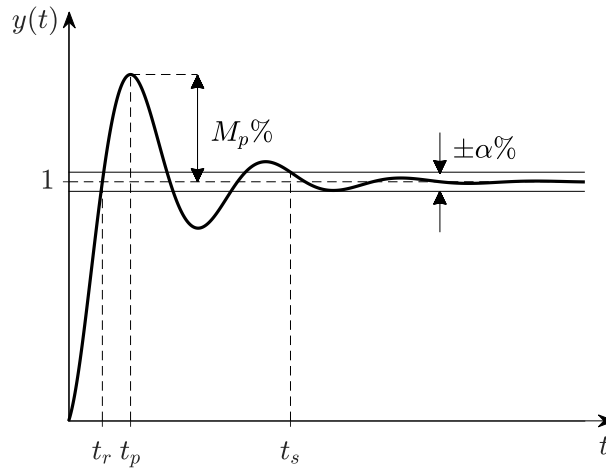
A margem delimitada por α dependerá da aplicação, sendo que 1%, 2% e 5% são alguns valores frequentemente usados.

Vale mencionar que a aplicação dessas equações deve ser feita com muito cuidado, pois para uma dada curva de resposta, t_r , t_p e t_s podem não ser os únicos instantes em que suas respectivas equações são válidas. Por exemplo, a derivada da resposta pode se anular em mo-

mentos diferentes de seu tempo de pico.

O tempo de subida e o tempo de pico são medidas da velocidade da resposta do sistema, enquanto a ultrapassagem percentual e o tempo de assentamento medem a estabilidade e a proximidade da resposta com o desejado. A Figura 2.9 mostra uma representação gráfica de cada um dos critérios de desempenho aqui explanados.

Figura 2.9. Interpretação gráfica dos parâmetros de desempenho na curva de resposta ao degrau.



Fonte: Autoria própria.

Sejam f_{tr} , f_{tp} , f_{mp} e f_{ts} funções advindas das Equações (2.44), (2.45), (2.46) e (2.47), respectivamente, definidas para $y_{ss} = 1$ por:

$$\begin{cases} f_{tr} = y(t_r) - 1 \\ f_{tp} = y'(t_p) \\ f_{mp} = y(t_p) - \left(1 + \frac{M_p}{100}\right) \\ f_{ts} = y(t_s) - \left(1 \pm \frac{\alpha}{100}\right). \end{cases} \quad (2.48)$$

Essas funções serão chamadas de *funções de desempenho* do controlador. Em síntese, especifica-se t_r , t_p , M_p , t_s e α , e analisa-se onde os polos e zeros de $H(s)$ precisam estar localizados para que a resposta real satisfaça as especificações desejadas. Os polos e zeros são modificados através de ajustes no controlador e, quando todos os critérios forem atendidos, tem-se o sistema de equações $f_{tr} = f_{tp} = f_{mp} = f_{ts} = 0$.

2.2.3 Controlador PID

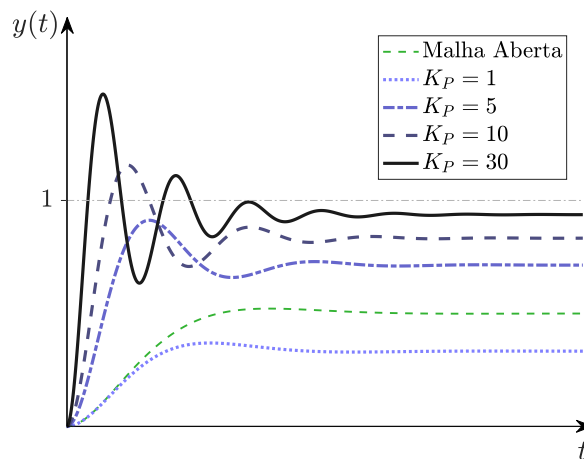
Suponha que se deseja desenvolver uma expressão matemática para um controlador que realize o rastreamento do sinal de entrada independentemente das características dinâmicas do

processo envolvido. Ele não precisa necessariamente ser o mais eficiente, mas deve cumprir as premissas básicas que se espera de um controle em malha fechada. Agora, considere a atuação desse controlador a cada instante: inicialmente o erro é máximo, portanto, é desejável que o sinal de controle também seja máximo, visando reduzir esse erro rapidamente; com essa ação, a saída começa a se aproximar do valor de referência e o sinal de controle deve reduzir conforme isso ocorre; por fim, a saída atinge a referência e o sinal de controle deve cessar juntamente ao erro. Essa descrição de funcionamento caracteriza o tipo mais simples de controle em malha fechada, o chamado controle *proporcional* ou simplesmente controle P. O mesmo atua com o objetivo de fornecer um sinal de controle proporcional ao erro do sistema, vinculando-os através de uma constante de proporcionalidade K_P , conhecida como coeficiente proporcional. Desse modo, sua equação no domínio do tempo é a seguinte:

$$x(t) = K_P e(t). \quad (2.49)$$

A Figura 2.10 compara a resposta ao degrau para diferentes valores de ganho proporcional no controle de um processo de segunda ordem. Observe que o aumento de K_P aproxima a resposta da referência no estado estacionário e eleva a velocidade de resposta, mas, em contrapartida, também aumenta a amplitude das oscilações e faz com que os transitórios persistam por mais tempo. Torna-se evidente que, apesar das vantagens desse controlador, como uma resposta rápida e baixa complexidade, há a necessidade de adicionar elementos dinâmicos ou compensadores à malha de controle para melhorar a estabilidade e as características de erro do sistema [1].

Figura 2.10. Curvas de resposta ao degrau do processo $G(s) = (s^2 + 2s + 2)^{-1}$ em sistemas de malha aberta e de malha fechada com controle P para diferentes valores de ganho proporcional.



Fonte: Autoria própria.

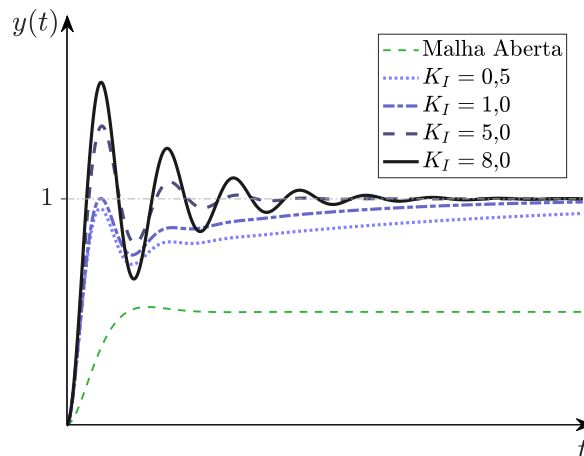
A principal falha do controle puramente proporcional é a não garantia da convergência do erro para zero, ou seja, há um erro de estado estacionário. Este problema é resolvido com a inclusão de um elemento que contabiliza os valores anteriores do erro do sistema para determinar se ainda há algum erro presente e que dê a devida compensação para zerá-lo. Essas características procuradas são inerentes a um integrador e, em combinação com o controle proporcional, formará o controlador *proporcional integral* (PI) ou ainda compensador de atraso, por levar em conta valores passados do sinal de erro. Sua atuação é descrita pela equação

$$x(t) = K_P e(t) + K_I \int_0^t e(t) dt, \quad (2.50)$$

na qual K_I é o coeficiente integral.

O gráfico da Figura 2.11 apresenta as respostas do mesmo processo anterior, agora sob o controle PI. Nota-se que o erro de estado estacionário é completamente eliminado e o valor de K_I determina o quão rápido essa convergência ocorre. Porém, assim como no controle proporcional, o aumento desse coeficiente causa uma resposta transitória abrupta e agrava o *overshoot*.

Figura 2.11. Curvas de resposta ao degrau do processo $G(s) = (s^2 + 2s + 2)^{-1}$ em sistemas de malha aberta e de malha fechada com controle PI para diferentes valores de ganho integral e $K_P = 5$ fixo.



Fonte: Autoria própria.

A problemática de conturbação transitória promovida pelos ganhos proporcional e integral pede por uma ação derivativa para ser corrigida. Um derivador fornece uma compensação negativa no sinal de controle caso o erro estiver reduzindo rapidamente, essencialmente suavizando a curva de resposta. O controle *proporcional derivativo* (PD) utiliza a taxa de variação instantânea do erro para calcular uma estimativa de valores futuros para a saída do sistema e dar

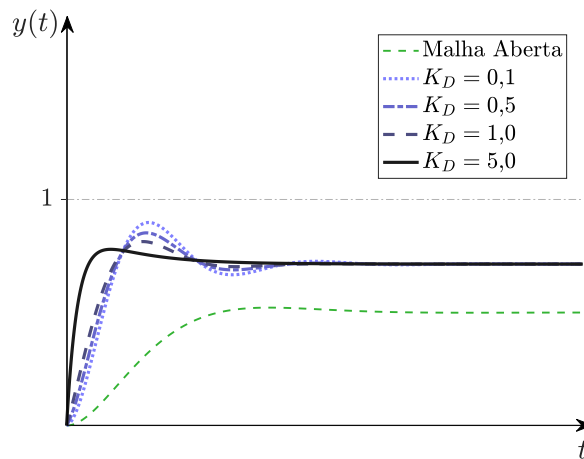
uma compensação para reduzir essa taxa conforme se aproxima da referência. Por esse motivo, também é chamado de compensador de avanço. Sua equação é da forma:

$$x(t) = K_P e(t) + K_D \frac{de(t)}{dt}, \quad (2.51)$$

em que K_D é o ganho derivativo.

Na Figura 2.12, são traçadas as curvas de resposta do mesmo processo sob o controle PD. O aumento de K_D reduz o *overshoot* e aumenta a velocidade de resposta, mas não corrige o erro de estado estacionário. Além disso, a implementação do controlador PD deve ser condicionada à ausência de uma quantidade significativa de ruídos no sistema pois, em sistemas muito ruidosos, as perturbações podem ser amplificadas pela ação derivativa e gerar instabilidade no processo. Essa desvantagem passa despercebida em um ambiente de simulação com processos ideais e deve ser levada em consideração na prática.

Figura 2.12. Curvas de resposta ao degrau do processo $G(s) = (s^2 + 2s + 2)^{-1}$ em sistemas de malha aberta e de malha fechada com controle PD para diferentes valores de ganho derivativo e $K_P = 5$ fixo.



Fonte: Autoria própria.

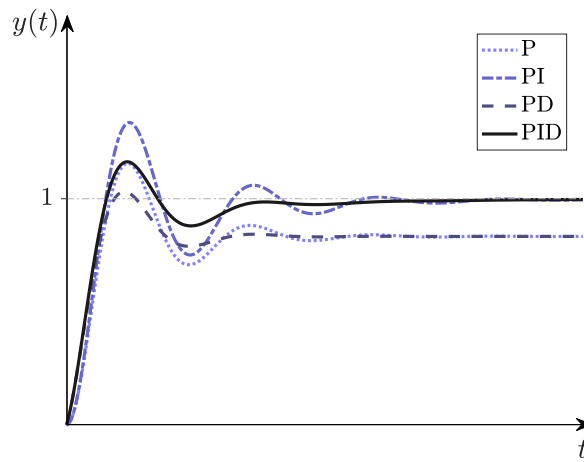
Incorporando as três ações de compensação em um só sistema, obtém-se o controle PID, cuja equação no domínio do tempo é

$$x(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt}. \quad (2.52)$$

Uma comparação entre as respostas dos controladores até agora tratados pode ser vista no gráfico da Figura 2.13. Partindo de um sistema de controle em malha aberta e comparando cada sistema de controle com o anterior, percebe-se que o ganho proporcional acelera a resposta e reduz o erro, o ganho integral elimina o erro de estado estacionário e o ganho derivativo melhora

a estabilidade do sistema.

Figura 2.13. Comparação entre os diferentes tipos de controle em um mesmo processo.



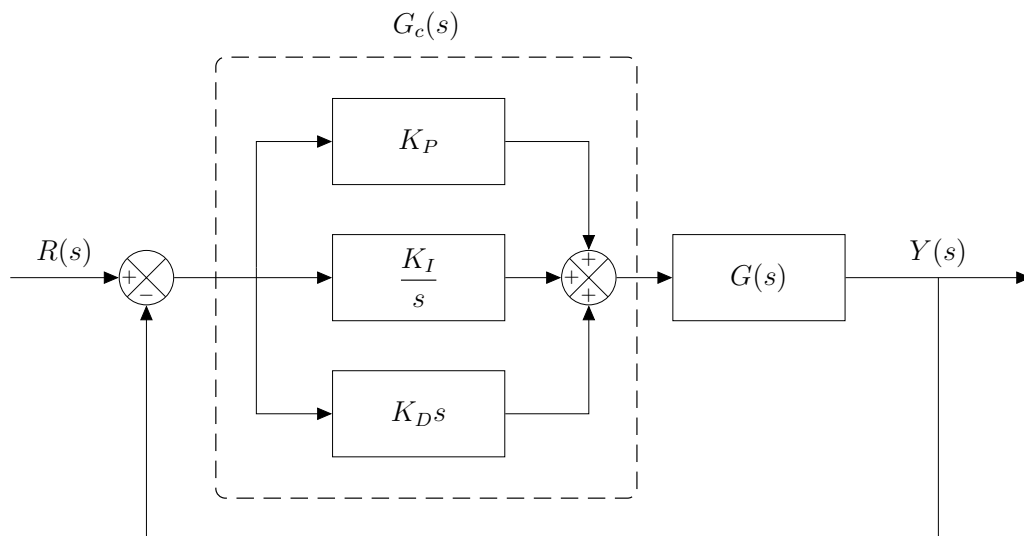
Fonte: Autoria própria.

A transformada de Laplace de (2.52) é calculada com base nas propriedades de derivação e integração, obtendo a função de transferência do controlador PID:

$$G_c(s) = \frac{X(s)}{E(s)} = K_P + \frac{K_I}{s} + K_D s. \quad (2.53)$$

A Figura 2.14 expõe o diagrama de blocos desse controlador em um sistema de controle com realimentação, enquanto a Tabela 2.2 apresenta a maneira como cada coeficiente influencia nas especificações de desempenho da resposta ao degrau.

Figura 2.14. Diagrama de blocos de um sistema de controle em malha fechada com controle PID.



Fonte: Autoria própria.

Tabela 2.2. Efeitos do aumento de cada coeficiente do controlador PID na resposta do sistema.

Parâmetro	t_r	t_p	M_p	t_s	e_{ss}	Estabilidade
K_P	Reduz	Reduz	Aumenta	Aumenta levemente	Reduz	Degrada
K_I	Reduz levemente	Não afeta	Aumenta	Aumenta	Elimina	Degrada
K_D	Reduz levemente	Reduz	Reduz	Reduz	Não afeta	Melhora para K_D pequeno

Fonte: Ang [28].

2.2.4 Funções de Desempenho de um Controlador PI em Sistemas de Primeira Ordem

Considere a equação diferencial que descreve um sistema de primeira ordem qualquer,

$$\frac{dy(t)}{dt} + a_1y(t) = b_0\frac{dx(t)}{dt} + b_1x(t), \quad (2.54)$$

em que a_1 , b_0 e b_1 são constantes reais e todas as condições iniciais de $x(t)$ e $y(t)$ são nulas. Se $G(s)$ é a função de transferência correspondente a este sistema, ela será escrita como

$$G(s) = \frac{b_0s + b_1}{s + a_1}. \quad (2.55)$$

Agora, será estudada a resposta ao degrau de $G(s)$ quando operando sob um controle em malha fechada, de forma que $G_c(s)$ é a função de transferência do controlador PI:

$$G_c(s) = K_P + \frac{K_I}{s}. \quad (2.56)$$

Substituindo (2.55) e (2.56) na equação de ganho de malha fechada em (2.40), chega-se na expressão

$$H(s) = \frac{b_0K_Ps^2 + (b_1K_P + b_0K_I)s + b_1K_I}{(1 + b_0K_P)s^2 + (a_1b_1K_P + b_0K_I)s + b_1K_I}. \quad (2.57)$$

Conforme (2.42), a condição inicial $y(0^+) = 0$ só será respeitada quando $H(s)$ for estritamente própria, o que ocorre somente para $b_0 = 0$. Portanto, a análise do desempenho do controlador PI será limitada para sistemas de primeira ordem do tipo

$$G(s) = \frac{b}{s + a}, \quad (2.58)$$

com $b \neq 0$, gerando a nova função de transferência em malha fechada,

$$H(s) = \frac{bK_P s + bK_I}{s^2 + (a + bK_P)s + bK_I}. \quad (2.59)$$

Com base na ordem da função de transferência obtida, supõe-se que a resposta desse sistema tenha características similares aos exemplos da Seção 2.1.5 de sistemas de segunda ordem. Se p_1 e p_2 são os polos de $H(s)$, tem-se que

$$(s - p_1)(s - p_2) = s^2 + (a + bK_P)s + bK_I.$$

Ao expandir o polinômio da esquerda da igualdade, obtém-se o sistema

$$\begin{cases} p_1 + p_2 = -(a + bK_P) \\ p_1 p_2 = bK_I. \end{cases} \quad (2.60)$$

Evidentemente, as soluções deste sistema serão as raízes da equação característica dada por $s^2 + (a + bK_P)s + bK_I = 0$:

$$p_1, p_2 = \frac{-(a + bK_P) \pm \sqrt{(a + bK_P)^2 - 4bK_I}}{2}. \quad (2.61)$$

O sinal do termo dentro da raiz em (2.61) determinará o formato da componente natural da resposta ao degrau do sistema, que será descrita por uma das Equações (2.29), (2.30) ou (2.34). Já a componente forçada da resposta ao degrau, como mostrado na Seção 2.2.2, é $y_f(t) = u(t)$. Desde que p_1 e p_2 estejam localizados no semiplano esquerdo, as seguintes afirmações podem ser constatadas:

- Se $(a + bK_P)^2 - 4bK_I < 0$, os polos são conjugados complexos e a resposta é subamortecida com equação

$$y(t) = [1 + 2|r_1|e^{\sigma t} \cos(\omega t + \text{Arg}\{r_1\})] u(t); \quad (2.62)$$

- Se $(a + bK_P)^2 - 4bK_I > 0$, os polos são reais e distintos e a resposta é superamortecida com equação

$$y(t) = (1 + r_1 e^{\sigma_1 t} + r_2 e^{\sigma_2 t}) u(t); \quad (2.63)$$

- Se $(a + bK_P)^2 - 4bK_I = 0$, os polos são reais e equivalentes e a resposta é criticamente amortecida com equação

$$y(t) = [1 + (r_1 + r_2 t)e^{\sigma t}] u(t). \quad (2.64)$$

Para determinar os valores dos resíduos r_1 e r_2 , deve-se voltar a atenção para a transformada da resposta ao degrau. Nos dois primeiros casos, os polos são distintos entre si e a resposta ao degrau $Y(s)$ poderá ser decomposta em frações parciais como segue:

$$Y(s) = \frac{r_1}{s - p_1} + \frac{r_2}{s - p_2} + \frac{1}{s}. \quad (2.65)$$

Uma vez que $H(s)$ e $Y(s)$ possuem o mesmo numerador, pode-se realizar o produto de cada resíduo pelos fatores lineares das outras frações, somar o resultado e igualar ao numerador de (2.59), obtendo

$$r_1 s(s - p_2) + r_2 s(s - p_1) + (s - p_1)(s - p_2) = bK_P s + bK_I.$$

Mas note que $(s - p_1)(s - p_2)$ é simplesmente o polinômio do denominador de $H(s)$, portanto

$$r_1 s(s - p_2) + r_2 s(s - p_1) + s^2 + (a + bK_P)s + bK_I = bK_P s + bK_I,$$

que resulta no sistema de equações

$$\begin{cases} r_1 + r_2 = -1 \\ r_1 p_2 + r_2 p_1 = a. \end{cases} \quad (2.66)$$

Tendo em mente que $p_1 \neq p_2$, a solução deste sistema será dada, na forma matricial, por:

$$\begin{bmatrix} r_1 \\ r_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ p_2 & p_1 \end{bmatrix}^{-1} \begin{bmatrix} -1 \\ a \end{bmatrix}. \quad (2.67)$$

Delineada uma forma de se calcular os polos e resíduos a partir dos coeficientes do controlador PI, as funções de desempenho da curva de resposta ao degrau serão utilizadas para avaliar a performance do controle. Aplicando a transformada inversa de Laplace na Equação

(2.65), obtém-se a resposta ao degrau

$$y(t) = (1 + r_1 e^{p_1 t} + r_2 e^{p_2 t})u(t), \quad (2.68)$$

e ao substituir $y(t)$ em (2.48), o seguinte sistema de equações é gerado:

$$\begin{cases} f_{tr}(K_P, K_I) = r_1 e^{p_1 t_r} + r_2 e^{p_2 t_r} & = 0 \\ f_{tp}(K_P, K_I) = r_1 p_1 e^{p_1 t_p} + r_2 p_2 e^{p_2 t_p} & = 0 \\ f_{mp}(K_P, K_I) = r_1 e^{p_1 t_p} + r_2 e^{p_2 t_p} - \frac{M_p}{100} & = 0 \\ f_{ts}(K_P, K_I) = r_1 e^{p_1 t_s} + r_2 e^{p_2 t_s} \mp \frac{\alpha}{100} & = 0. \end{cases} \quad (2.69)$$

Em suma, para um dado processo de primeira ordem $G(s)$, as características desejadas para a resposta são estabelecidas de antemão, e cabe ao projetista encontrar os valores de K_P e K_I que tornam a curva de resposta a mais próxima possível da desejada. Ou, com base na reformulação do problema aqui desenvolvida, busca-se a maneira mais eficiente de se resolver o sistema de equações não lineares em (2.69) em função de K_P e K_I . Os principais métodos a serem utilizados serão abordados na Seção 2.3.

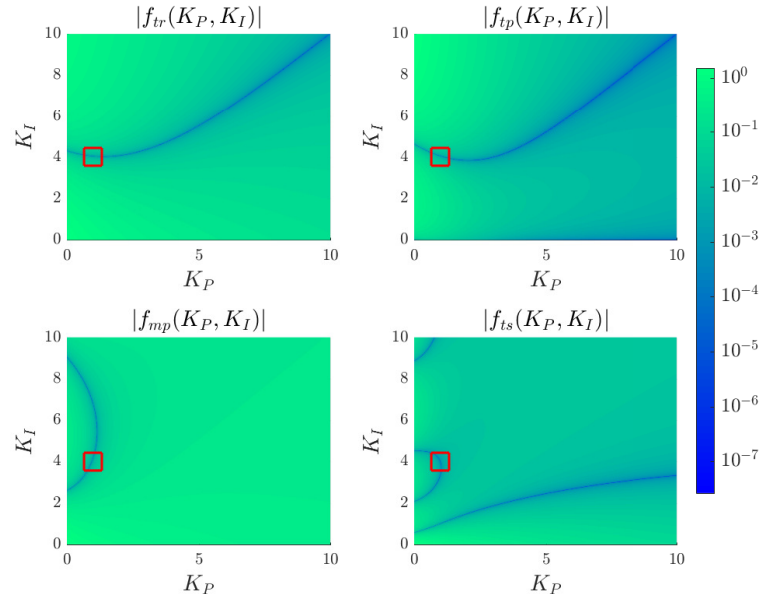
A Figura 2.15 mostra um exemplo dos valores das funções de desempenho em um processo de primeira ordem para cada valor dos coeficientes PI dentro de um intervalo específico. O retângulo vermelho em cada gráfico dá destaque para a região onde uma possível solução para o sistema de equações pode ser encontrada, para $K_P \approx 1$ e $K_I \approx 4$. Por se tratar de um sistema sobredeterminado (com mais equações que incógnitas), a probabilidade de existir uma solução dentro da precisão requerida é amplificada quando apenas dois critérios de desempenho são fixados.

2.2.5 Funções de Desempenho de um Controlador PID em Sistemas de Maior Ordem

Agora, procedimentos similares ao caso de primeira ordem serão feitos para sistemas de maior ordem, realizando as adaptações necessárias. Seja $G(s)$ um processo LTI de ordem $n > 1$ modelado pela equação diferencial (2.6) e com função de transferência

$$G(s) = \frac{b(s)}{a(s)} = \frac{b_0 s^n + b_1 s^{n-1} + \dots + b_{n-1} s + b_n}{s^n + a_1 s^{n-1} + a_2 s^{n-2} + \dots + a_{n-1} s + a_n}, \quad (2.70)$$

Figura 2.15. Valor absoluto das funções de desempenho do processo $(s + 1)^{-1}$ para uma variação contínua dos coeficientes K_P e K_I com as especificações $t_r = 0,9$ s, $t_p = 1,5$ s, $M_p = 20\%$ e $t_s = 3,85$ s para $\alpha = 2\%$.



Fonte: Autoria própria.

e seja $G_c(s)$ a função de transferência de um controlador PID,

$$G_c(s) = \frac{c(s)}{d(s)} = \frac{K_D s^2 + K_P s + K_I}{s}. \quad (2.71)$$

O ganho de malha fechada $H(s)$ deste sistema, conforme a Equação (2.40), será:

$$\begin{aligned} H(s) &= \frac{b(s)c(s)}{a(s)d(s) + b(s)c(s)} \\ &= \frac{(b_0 s^n + b_1 s^{n-1} + \dots + b_{n-1} s + b_n)(K_D s^2 + K_P s + K_I)}{(b_0 s^n + b_1 s^{n-1} + \dots + b_{n-1} s + b_n)(K_D s^2 + K_P s + K_I) + (s^n + a_1 s^{n-1} + a_2 s^{n-2} + \dots + a_{n-1} s + a_n)s}. \end{aligned}$$

Antes de realizar as devidas simplificações na expressão de $H(s)$, uma análise prévia dos coeficientes resultantes no numerador e no denominador pode revelar informações importantes quanto às restrições necessárias sobre o processo para possibilitar um controle efetivo. Supondo, em um primeiro momento, que todos os coeficientes a_k e b_k sejam não nulos, $H(s)$ será a razão entre dois polinômios de grau $(n + 2)$, cujo limite no infinito será

$$\lim_{s \rightarrow \infty} H(s) = \frac{b_0 K_D}{b_0 K_D}.$$

Porém, pela Equação (2.42), sabe-se que a condição inicial $y(0^+) = 0$ será validada somente

quando esse limite tender a zero. Então, deve-se ter $b_0 = 0$. Ainda assim, o limite se torna

$$\lim_{s \rightarrow \infty} H(s) = \frac{b_1 K_D}{1 + b_1 K_D},$$

o que também força $b_1 = 0$ e, finalmente, o limite é zero. Dessa forma, o polinômio $b(s)$ do processo deve possuir grau menor ou igual a $(n - 2)$. Outra maneira de interpretar essa restrição é que $G(s)$ deve ter uma quantidade máxima de zeros igual à quantidade de polos menos dois.

Expressando os polinômios de $H(s)$ em notação de somatório:

$$H(s) = \frac{N(s)}{D(s)} = \frac{\sum_{k=0}^{n+1} (K_P b_k + K_I b_{k-1} + K_D b_{k+1}) s^{n+1-k}}{\sum_{k=0}^{n+1} (a_k + K_P b_k + K_I b_{k-1} + K_D b_{k+1}) s^{n+1-k}}, \quad (2.72)$$

em que definem-se $a_0 = 1$, $a_k = 0$ se $k \notin [0, n]$ e $b_k = 0$ se $k \notin [2, n]$.

A equação característica $D(s) = 0$ é uma equação polinomial de grau $(n + 1)$, e se p_1, \dots, p_{n+1} são suas soluções, e conseqüentemente os polos de $H(s)$, então $D(s)$ é escrito como o produto dos fatores primos correspondentes a cada polo, como

$$D(s) = \prod_{k=1}^{n+1} (s - p_k). \quad (2.73)$$

A ordem do sistema condiciona a complexidade dos cálculos necessários para a obtenção desses polos, e quando n for grande o suficiente, deve-se recorrer para métodos numéricos de determinação das raízes de polinômios. Ainda assim, não está claro o modo como os ganhos do controlador afetam o posicionamento de cada um desses polos, logo, o que pode ser feito é buscar uma expressão genérica para cada um dos coeficientes do polinômio resultante do produtório em (2.73) e atingir um sistema de equações que relacionam esses polos com os coeficientes da planta e do controlador.

A partir deste ponto, um conceito da álgebra comutativa será usado para simplificar a notação durante os cálculos. O produto da Equação (2.73) gera um polinômio no qual cada um de seus coeficientes é um *polinômio simétrico elementar* (PSE), denotado por e_k e definido como a soma de todos os produtos de k polos distintos. Como exemplo, os três primeiros termos

e_k são

$$e_1(p_1, \dots, p_{n+1}) = \sum_{j_1=1}^{n+1} p_{j_1}, \quad e_2(p_1, \dots, p_{n+1}) = \sum_{j_1=1}^{n+1} \sum_{j_2=j_1+1}^{n+1} p_{j_1} p_{j_2},$$

$$e_3(p_1, \dots, p_{n+1}) = \sum_{j_1=1}^{n+1} \sum_{j_2=j_1+1}^{n+1} \sum_{j_3=j_2+1}^{n+1} p_{j_1} p_{j_2} p_{j_3}.$$

Trivialmente, sabe-se que $e_0 = 1$, enquanto que, para $k = 1, \dots, n+1$, a seguinte generalização pode ser obtida [29]:

$$e_k = e_k(p_1, \dots, p_{n+1}) = \sum_{1 \leq j_1 < \dots < j_k \leq n+1} p_{j_1} \cdots p_{j_k}. \quad (2.74)$$

Diz-se que $e_k(p_1, \dots, p_{n+1})$ é um PSE de grau k em $(n+1)$ polos. Utilizando esse conceito, escreve-se a expansão do produto em (2.73) como

$$D(s) = \sum_{k=0}^{n+1} (-1)^k e_k(p_1, \dots, p_{n+1}) s^{n+1-k}. \quad (2.75)$$

Uma equivalência entre (2.75) e o denominador de (2.72) resulta no sistema de $(n+1)$ equações não lineares abaixo:

$$\begin{cases} e_1 = -(a_1 + b_1 K_P + b_0 K_I + b_2 K_D) \\ e_2 = a_2 + b_2 K_P + b_1 K_I + b_3 K_D \\ \vdots \\ e_{n+1} = (-1)^{n+1} (a_{n+1} + b_{n+1} K_P + b_{n-1} K_I + b_{n+2} K_D). \end{cases} \quad (2.76)$$

Essas equações não oferecem muita utilidade no momento e a relação entre os polos e os coeficientes do controlador permanece vago. Porém, ao adentrar na problemática dos métodos numéricos de resolução, esse sistema se tornará de fundamental importância.

Se todos os polos forem dois a dois distintos, a resposta ao degrau do sistema é expandida em frações parciais no seguinte modo:

$$Y(s) = \frac{r_1}{s - p_1} + \dots + \frac{r_{n+1}}{s - p_{n+1}} + \frac{1}{s}. \quad (2.77)$$

O numerador da Equação (2.72) pode ser expresso em termos dos resíduos ao juntar as frações

parciais acima, chegando em

$$N(s) = \sum_{j=1}^{n+1} r_j N_j(s) + D(s), \quad (2.78)$$

em que cada $N_j(s)$ é um polinômio de grau $(n + 1)$ composto pelo produto de todos os fatores primos dos polos de $Y(s)$, com exceção do que corresponde ao coeficiente r_j . O termo $D(s)$ surge por conta da componente forçada $1/s$, que adiciona o produto $(s - p_1) \cdots (s - p_{n+1})$ ao numerador. Cada $N_j(s)$ será dado por

$$N_j(s) = s \prod_{\substack{k=1 \\ k \neq j}}^{n+1} (s - p_k). \quad (2.79)$$

A similaridade entre os produtórios das Equações (2.73) e (2.79) torna inevitável a realização de um paralelo entre as duas. Para algum $j = 1, \dots, n$, se o fator $(s - p_j)$ em $D(s)$ for substituído por s , obtém-se $N_j(s)$. Logo, ao expandir $N_j(s)$ na forma polinomial, o seu k -ésimo coeficiente deve conter a soma de todos os produtos dos $(k - 1)$ elementos p_k distintos, com $k \neq j$. Felizmente, há uma maneira simples de obter cada um desses termos pela derivada de e_k com relação a p_j , na qual será utilizada a notação

$$\frac{\partial e_k}{\partial p_j} = e_{k-1}(p_1, \dots, p_{j-1}, p_{j+1}, \dots, p_{n+1}) = e_{k-1}^{(j)} \quad (2.80)$$

para representar essa nova combinação [30]. Portanto, tem-se que

$$N_j(s) = \sum_{k=0}^{n+1} (-1)^k e_k^{(j)} s^{n+1-k}. \quad (2.81)$$

Substituindo (2.75) em (2.78) e expandindo $D(s)$ com o somatório do denominador em (2.72):

$$\begin{aligned} N(s) &= \sum_{j=1}^{n+1} \left[r_j \sum_{k=0}^{n+1} (-1)^k e_k^{(j)} s^{n+1-k} \right] + \sum_{k=0}^{n+1} (a_k + b_k K_P + b_{k-1} K_I + b_{k+1} K_D) s^{n+1-k} \\ &= \sum_{k=0}^{n+1} \left[\sum_{j=1}^{n+1} (-1)^k e_k^{(j)} r_j + a_k + b_k K_P + b_{k-1} K_I + b_{k+1} K_D \right] s^{n+1-k}. \end{aligned}$$

Uma associação entre a expressão encontrada para $N(s)$ e o numerador da Equação (2.72)

origina o sistema de equações

$$\left\{ \begin{array}{l} \sum_{j=1}^{n+1} r_j + a_0 + b_0 K_P + b_{-1} K_I + b_1 K_D = b_0 K_P + b_{-1} K_I + b_1 K_D \\ - \sum_{j=1}^{n+1} e_1^{(j)} r_j + a_1 + b_1 K_P + b_0 K_I + b_2 K_D = b_1 K_P + b_0 K_I + b_2 K_D \\ \vdots \\ (-1)^n \sum_{j=1}^{n+1} e_n^{(j)} r_j + a_n + b_n K_P + b_{n-1} K_I + b_{n+1} K_D = b_n K_P + b_{n-1} K_I + b_{n+1} K_D. \end{array} \right.$$

Então, a partir dos valores encontrados para os polos, calcula-se cada elemento $e_{k-1}^{(j)}$ conforme (2.80), para $k, j = 1, \dots, n+1$, e os resíduos serão determinados por:

$$\begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & \cdots & 1 \\ e_1^{(1)} & \cdots & e_1^{(n+1)} \\ \vdots & \ddots & \vdots \\ e_n^{(1)} & \cdots & e_n^{(n+1)} \end{bmatrix}^{-1} \begin{bmatrix} -a_0 \\ a_1 \\ \vdots \\ (-1)^{n+1} a_n \end{bmatrix}. \quad (2.82)$$

Enfim, a resposta ao degrau é calculada pela transformada inversa de (2.77),

$$y(t) = 1 + \sum_{k=1}^{n+1} r_k e^{p_k t}, \quad (2.83)$$

e com base nas definições em (2.48), as funções de desempenho para o controle PID de um sistema de ordem n serão:

$$\left\{ \begin{array}{l} f_{tr}(K_P, K_I, K_D) = \sum_{k=1}^{n+1} r_k e^{p_k t_r} = 0 \\ f_{tp}(K_P, K_I, K_D) = \sum_{k=1}^{n+1} r_k p_k e^{p_k t_p} = 0 \\ f_{mp}(K_P, K_I, K_D) = \sum_{k=1}^{n+1} r_k e^{p_k t_p} - \frac{M_p}{100} = 0 \\ f_{ts}(K_P, K_I, K_D) = \sum_{k=1}^{n+1} r_k e^{p_k t_s} \mp \frac{\alpha}{100} = 0. \end{array} \right. \quad (2.84)$$

2.3 Métodos de Resolução de Sistemas de Equações Não Lineares

Os procedimentos para encontrar a raiz de uma única função de uma variável real são simples. Se é conhecido um intervalo no qual a função possui sinais opostos nas extremidades, aplica-se um dos métodos fechados de resolução, que subdividem esse intervalo em partições cada vez menores e garantem a convergência para a raiz. Métodos como o da bisseção e *regula falsi* entram nessa categoria. Quando não se tem conhecimento de tal intervalo, é necessário aplicar um dos métodos abertos, que necessitam da escolha de um ou dois pontos iniciais para começar a busca pela raiz, e possuem certas condições para garantir a convergência. Dentre os principais métodos abertos, pode-se citar o método do ponto fixo, o método de Newton-Raphson e o método da secante [31].

Todavia, a resolução simultânea de várias equações não lineares adiciona diversas camadas de complicação. Não há como ter certeza que uma solução para essas equações exista até que ela seja encontrada, e ao contrário do caso unidimensional, não é possível limitar a busca pela solução a um intervalo específico sem incertezas quanto à convergência [32]. A estratégia para solucionar tais sistemas, então, é iniciar a procura em um ponto em que as funções são próximas de zero e utilizar um algoritmo que dê pequenos incrementos às variáveis até atingir o critério de convergência.

Sistemas de m equações e n incógnitas podem ser escritos, na notação vetorial, como

$$\mathbf{F}(\mathbf{x}) = 0, \quad (2.85)$$

em que $\mathbf{F}: X \rightarrow \mathbb{R}^m$ é uma função vetorial, com $X \subseteq \mathbb{R}^n$. Resolver o sistema é equivalente a encontrar algum $\mathbf{x} \in X$ que satisfaça a equação vetorial acima. Quando $m < n$, o sistema é dito *subdeterminado* e, quando $m > n$, o sistema é dito *sobredeterminado*. Sistemas sobredeterminados, por terem mais restrições que graus de liberdade, são considerados mais difíceis de serem resolvidos e é comum a não existência de soluções nesse caso.

Métodos iterativos são, sem dúvidas, a maneira mais efetiva de encontrar uma solução de sistemas não lineares. O processo iterativo consiste de um conjunto de procedimentos que são repetidos várias vezes para encontrar uma solução do sistema, caso existir. Isso é realizado através de aproximações sucessivas $\mathbf{x}^{(k)}$ para a solução desejada, iniciando-se de um “palpite”

inicial $\mathbf{x}^{(0)}$ e descrito por uma regra de iteração

$$\mathbf{x}^{(k+1)} = \mathbf{G}(\mathbf{x}^{(k)}),$$

em que $\mathbf{G}: Y \rightarrow \mathbb{R}^m$ é uma função vetorial, com $Y \subseteq \mathbb{R}^n$. A escolha de um bom “palpite” inicial é crucial para a convergência do método. Essa decisão costuma ser antecedida de uma análise detalhada acerca da natureza do sistema para obter uma boa aproximação para a solução logo de início. A interrupção do processo iterativo ocorre mediante a um critério de parada pré-estabelecido, que pode ser com base na convergência da variável para uma solução aproximada, na divergência da variável para valores ordens de magnitude acima do esperado, ou no número de iterações ao fixar um limite máximo para este.

A seguir, dois métodos serão apresentados para uma possível aplicação na resolução dos sistemas das funções de desempenho: o método de Newton-Raphson e o método de Levenberg-Marquardt.

2.3.1 Método de Newton-Raphson

O método de Newton-Raphson é um método numérico iterativo comumente utilizado para determinar as raízes de equações não lineares. Se $f: X \rightarrow \mathbb{R}$ é uma função de uma variável, em que $X \subseteq \mathbb{R}$, e $\bar{x} \in \mathbb{R}$ é um valor tal que $f(\bar{x}) = 0$, então, o método de Newton-Raphson de uma variável é caracterizado pelo processo iterativo

$$x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})},$$

para $k \geq 0$ inteiro e em que x_0 é um valor inicial que garante a convergência do processo desde que $|x_0 - \bar{x}|$ seja suficientemente pequeno [33].

O método numérico supracitado também possui uma versão mais genérica que é aplicável em sistemas de equações. Considere a Equação (2.85) e faça $m = n$, obtendo um sistema de n equações e n variáveis. Sejam $f_1, \dots, f_n: X \rightarrow \mathbb{R}$, com $X \subseteq \mathbb{R}^n$, tais que $\mathbf{F} = (f_1, \dots, f_n)$. Procura-se encontrar um vetor $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$ de forma que

$$\begin{cases} f_1(x_1, \dots, x_n) = 0 \\ \vdots \\ f_n(x_1, \dots, x_n) = 0. \end{cases}$$

O maior ponto fraco desse método está na quantidade de cálculos necessários para a inversão da matriz $\mathbf{J}_F(\mathbf{x})$ a cada iteração, que aumenta substancialmente com a ordem do sistema de equações. Outra desvantagem ocorre devido à provável não convergência caso o ponto inicial estiver distante de uma solução. Apesar disso, o método de Newton-Raphson continua sendo o principal método numérico aplicado em problemas de localização de raízes [34].

2.3.2 Método de Levenberg-Marquardt

O método de Levenberg-Marquardt pode ser visto como uma variação do método de Newton-Raphson, geralmente implementado em problemas de minimização por mínimos quadrados. A aplicação desse método requer uma reinterpretação do problema inicial de solucionar equações não lineares. Ao passar os dois termos da Equação (2.87) para o mesmo lado da igualdade, o objetivo se torna minimizar a função $S(\mathbf{x})$ obtida,

$$\min_{\mathbf{x} \in \mathbb{R}^n} S(\mathbf{x}) = \min_{\mathbf{x} \in \mathbb{R}^n} \frac{1}{2} \|\mathbf{F}(\mathbf{x}') + \mathbf{J}_F(\mathbf{x}')(\mathbf{x} - \mathbf{x}')\|^2, \quad (2.90)$$

que será chamada de função objetivo, sendo $\|\cdot\|$ a norma euclidiana no \mathbb{R}^n . Ela pode ser desenvolvida da seguinte forma:

$$\begin{aligned} S(\mathbf{x}) &= \frac{1}{2} [\mathbf{F}(\mathbf{x}') + \mathbf{J}_F(\mathbf{x}')(\mathbf{x} - \mathbf{x}')]^T [\mathbf{F}(\mathbf{x}') + \mathbf{J}_F(\mathbf{x}')(\mathbf{x} - \mathbf{x}')] \\ &= \frac{1}{2} [\mathbf{F}(\mathbf{x}')^T \mathbf{F}(\mathbf{x}') + (\mathbf{x} - \mathbf{x}')^T \mathbf{J}_F(\mathbf{x}')^T \mathbf{F}(\mathbf{x}') + \mathbf{F}(\mathbf{x}')^T \mathbf{J}_F(\mathbf{x}')(\mathbf{x} - \mathbf{x}') \\ &\quad + (\mathbf{x} - \mathbf{x}')^T \mathbf{J}_F(\mathbf{x}')^T \mathbf{J}_F(\mathbf{x}')(\mathbf{x} - \mathbf{x}')]. \end{aligned}$$

A derivada de $S(\mathbf{x})$ com relação a \mathbf{x} será igual a zero em seu ponto de mínimo, então:

$$\begin{aligned} \mathbf{F}(\mathbf{x}')^T \mathbf{J}_F(\mathbf{x}') + \mathbf{F}(\mathbf{x}')^T \mathbf{J}_F(\mathbf{x}') + (\mathbf{x} - \mathbf{x}')^T \mathbf{J}_F(\mathbf{x}')^T \mathbf{J}_F(\mathbf{x}') + (\mathbf{x} - \mathbf{x}')^T \mathbf{J}_F(\mathbf{x}')^T \mathbf{J}_F(\mathbf{x}') &= 0 \\ \mathbf{J}_F(\mathbf{x}')^T \mathbf{J}_F(\mathbf{x}') (\mathbf{x} - \mathbf{x}') &= -\mathbf{J}_F(\mathbf{x}')^T \mathbf{F}(\mathbf{x}'). \end{aligned}$$

A expressão obtida acima consiste de um outro método numérico de minimização, chamado de método de Gauss-Newton, que sofre com problemas de convergência similares aos do método de Newton, neste quando o produto $\mathbf{J}_F^T \mathbf{J}_F$ é singular. Pode-se contornar esse obstáculo ao introduzir um parâmetro λ_k nas diagonais da matriz $\mathbf{J}_F^T \mathbf{J}_F$, resultando na expressão

$$[\mathbf{J}_F(\mathbf{x}')^T \mathbf{J}_F(\mathbf{x}') + \lambda_k \mathbf{I}_n] (\mathbf{x} - \mathbf{x}') = -\mathbf{J}_F(\mathbf{x}')^T \mathbf{F}(\mathbf{x}'),$$

que é a equação do método de Levenberg-Marquardt [35, 36]. λ_k é chamado de parâmetro de *damping* e seu valor deve ser atualizado a cada iteração de forma que ele seja grande o suficiente para proporcionar um decréscimo na função objetivo. Além disso, deve se aproximar de zero nos estágios finais do algoritmo, tendendo à direção de Gauss-Newton [35]. Então, a regra de atualização por meio desse método se torna

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \left[\mathbf{J}_{\mathbf{F}}(\mathbf{x}^{(k)})^T \mathbf{J}_{\mathbf{F}}(\mathbf{x}^{(k)}) + \lambda_k \mathbf{I}_n \right]^{-1} \mathbf{J}_{\mathbf{F}}(\mathbf{x}^{(k)})^T \mathbf{F}(\mathbf{x}^{(k)}). \quad (2.91)$$

3 METODOLOGIA

Neste trabalho, realizou-se uma implementação computacional dos métodos de Newton-Raphson e de Levenberg-Marquardt apresentados na Seção 2.3 para solucionar os sistemas de equações das funções de desempenho de primeira ordem em (2.69) e o caso geral em (2.84). Por conseguinte, será demonstrada a existência de um método analítico de sintonia de controladores PID para o controle de sistemas LTI do tipo SISO de qualquer ordem com base em critérios de desempenho.

As equações em questão contém não linearidades originadas das funções exponenciais da resposta ao degrau e suas resoluções não podem ser feitas com relação aos resíduos r_k e aos polos p_k , afinal, seus valores operam sob as restrições impostas pelas Equações (2.76) e (2.82), que limitam a liberdade de posicionamento dos polos no plano complexo. Os coeficientes K_P , K_I e K_D , portanto, são as verdadeiras variáveis de interesse nas implementações numéricas que foram realizadas.

Ambos os métodos numéricos mencionados exigem o cálculo do jacobiano das funções de desempenho dos sistemas (2.69) e (2.84) com relação aos parâmetros que serão ajustados a cada iteração buscando a minimização dessas funções. Mas não há relações explícitas entre essas funções e os coeficientes PID. O que foi obtido, porém, são os sistemas de equações que relacionam os polos e resíduos com os parâmetros do controlador implicitamente. Visto que a forma como as funções variam a partir de mudanças nos polos e resíduos é facilmente alcançada, resta determinar como cada polo e resíduo varia a uma mudança nos coeficientes do controlador.

No algoritmo desenvolvido, primeiramente, são feitas as escolhas dos critérios de desempenho e do método numérico a ser aplicado. A quantidade de critérios deve ser igual ao número de coeficientes do controlador apropriado, para que componham um sistema com a mesma quantidade de equações e incógnitas. Depois, com um “palpite” inicial para os coeficientes do controlador, inicia-se a primeira iteração e calculam-se os polos e resíduos da resposta ao degrau. Após isso, devem ser calculados os jacobianos das funções de desempenho com relação aos polos e aos resíduos, e os jacobianos dos polos e dos resíduos com relação aos coeficientes do controlador. Enfim, determina-se o jacobiano das funções de desempenho com relação aos coeficientes do controlador pela regra da cadeia para funções de várias variáveis. No final do laço, aplica-se a regra de atualização do método numérico correspondente e o novo valor das funções de desempenho é analisado para determinar se o método deve ser interrompido por uma

possível convergência/divergência ou deve prosseguir para a próxima iteração. O algoritmo está disposto na forma de um fluxograma e de pseudocódigos com sintaxe similar à linguagem do MATLAB [37], software no qual se implementaram as simulações numéricas do algoritmo.

3.1 Jacobiano das Funções de Desempenho para Sistemas de Primeira Ordem

Levando em conta as considerações feitas para sistemas de primeira ordem da Seção 2.2.4, seja $\mathbf{F} = (f_1, f_2)$, em que $f_1, f_2 \in \{f_{tr}, f_{tp}, f_{mp}, f_{ts}\}$ e $f_1 \neq f_2$, e seja $\mathbf{K} = (K_P, K_I)$. Ao definir $\mathbf{P} = (p_1, p_2)$ e $\mathbf{R} = (r_1, r_2)$, tem-se que, o jacobiano de \mathbf{F} com relação a \mathbf{K} pode ser obtido pela regra da cadeia a seguir:

$$\mathbf{J}_{\mathbf{F}}(\mathbf{K}) = \mathbf{J}_{\mathbf{F}}(\mathbf{P})\mathbf{J}_{\mathbf{P}}(\mathbf{K}) + \mathbf{J}_{\mathbf{F}}(\mathbf{R})\mathbf{J}_{\mathbf{R}}(\mathbf{K}), \quad (3.1)$$

nos quais $\mathbf{J}_{\mathbf{F}}(\mathbf{P})$ e $\mathbf{J}_{\mathbf{F}}(\mathbf{R})$ são os jacobianos de \mathbf{F} com relação a \mathbf{P} e \mathbf{R} , respectivamente. Ademais, $\mathbf{J}_{\mathbf{P}}(\mathbf{K})$ é o jacobiano de \mathbf{P} com relação a \mathbf{K} e $\mathbf{J}_{\mathbf{R}}(\mathbf{K})$ é o jacobiano de \mathbf{R} com relação a \mathbf{K} .

Os jacobianos das funções de desempenho com relação aos polos e aos resíduos são obtidos prontamente a partir do sistema em (2.69). Basta estabelecer quais dos quatro critérios de desempenho serão adotados para aprimorar a performance do controle e formar as matrizes jacobianas a partir das derivadas de suas devidas funções. As derivadas individuais de f_{tr} , f_{tp} , f_{mp} e f_{ts} serão as linhas das matrizes

$$\begin{bmatrix} \frac{\partial f_{tr}}{\partial \mathbf{P}} \\ \frac{\partial f_{tp}}{\partial \mathbf{P}} \\ \frac{\partial f_{mp}}{\partial \mathbf{P}} \\ \frac{\partial f_{ts}}{\partial \mathbf{P}} \end{bmatrix} = \begin{bmatrix} r_1 t_r e^{p_1 t_r} & r_2 t_r e^{p_2 t_r} \\ (r_1 + r_1 p_1 t_p) e^{p_1 t_p} & (r_2 + r_2 p_2 t_p) e^{p_2 t_p} \\ r_1 t_p e^{p_1 t_p} & r_2 t_p e^{p_2 t_p} \\ r_1 t_s e^{p_1 t_s} & r_2 t_s e^{p_2 t_s} \end{bmatrix} \quad (3.2)$$

e

$$\begin{bmatrix} \frac{\partial f_{tr}}{\partial \mathbf{R}} \\ \frac{\partial f_{tp}}{\partial \mathbf{R}} \\ \frac{\partial f_{mp}}{\partial \mathbf{R}} \\ \frac{\partial f_{ts}}{\partial \mathbf{R}} \end{bmatrix} = \begin{bmatrix} e^{p_1 t_r} & e^{p_2 t_r} \\ p_1 e^{p_1 t_p} & p_2 e^{p_2 t_p} \\ e^{p_1 t_p} & e^{p_2 t_p} \\ e^{p_1 t_s} & e^{p_2 t_s} \end{bmatrix}. \quad (3.3)$$

Como um exemplo, se o projetista desejar obter um tempo de pico e uma ultrapassagem percentual específicos, deve-se fazer $\mathbf{F} = (f_{tp}, f_{mp})$, excluir a primeira e a quarta linhas das matrizes em (3.2) e em (3.3) e, assim, obter matrizes 2×2 que corresponderão a $\mathbf{J}_{\mathbf{F}}(\mathbf{P})$ e a $\mathbf{J}_{\mathbf{F}}(\mathbf{R})$, respectivamente.

Os jacobianos remanescentes, $\mathbf{J}_{\mathbf{P}}(\mathbf{K})$ e $\mathbf{J}_{\mathbf{R}}(\mathbf{K})$, podem ser encontrados ao fazer uma derivação implícita das equações que relacionam os polos e resíduos com os coeficientes PI. Derivando (2.60) com relação a K_P e a K_I e escrevendo o resultado na forma matricial, chega-se em

$$\begin{bmatrix} 1 & 1 \\ p_2 & p_1 \end{bmatrix} \underbrace{\begin{bmatrix} \frac{\partial p_1}{\partial K_P} & \frac{\partial p_1}{\partial K_I} \\ \frac{\partial p_2}{\partial K_P} & \frac{\partial p_2}{\partial K_I} \end{bmatrix}}_{\mathbf{J}_{\mathbf{P}}(\mathbf{K})} = \begin{bmatrix} -b & 0 \\ 0 & b \end{bmatrix},$$

que equivale a

$$\mathbf{J}_{\mathbf{P}}(\mathbf{K}) = \begin{bmatrix} 1 & 1 \\ p_2 & p_1 \end{bmatrix}^{-1} \begin{bmatrix} -b & 0 \\ 0 & b \end{bmatrix}. \quad (3.4)$$

Note que a matriz inversa acima é igual à atingida para calcular os valores dos resíduos, na Equação (2.67).

Por fim, derivando (2.66) com relação a K_P , obtém-se

$$\begin{bmatrix} 1 & 1 \\ p_2 & p_1 \end{bmatrix} \underbrace{\begin{bmatrix} \frac{\partial r_1}{\partial K_P} \\ \frac{\partial r_2}{\partial K_P} \end{bmatrix}}_{\frac{\partial \mathbf{R}}{\partial K_P}} + \begin{bmatrix} 0 & 0 \\ \frac{\partial p_2}{\partial K_P} & \frac{\partial p_1}{\partial K_P} \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} = 0.$$

A matriz em destaque representa a primeira coluna de $\mathbf{J}_{\mathbf{R}}(\mathbf{K})$, enquanto a segunda coluna vem

à tona ao trocar a variável de derivação para K_I . Portanto, ao isolar as derivadas, tem-se que

$$\left\{ \begin{array}{l} \frac{\partial \mathbf{R}}{\partial K_P} = - \begin{bmatrix} 1 & 1 \\ p_2 & p_1 \end{bmatrix}^{-1} \begin{bmatrix} 0 & 0 \\ \frac{\partial p_2}{\partial K_P} & \frac{\partial p_1}{\partial K_P} \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} \\ \frac{\partial \mathbf{R}}{\partial K_I} = - \begin{bmatrix} 1 & 1 \\ p_2 & p_1 \end{bmatrix}^{-1} \begin{bmatrix} 0 & 0 \\ \frac{\partial p_2}{\partial K_I} & \frac{\partial p_1}{\partial K_I} \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} \end{array} \right. , \quad (3.5)$$

e o jacobiano dos resíduos é

$$\mathbf{J}_R(\mathbf{K}) = \begin{bmatrix} \frac{\partial \mathbf{R}}{\partial K_P} & \frac{\partial \mathbf{R}}{\partial K_I} \end{bmatrix}. \quad (3.6)$$

3.2 Jacobiano das Funções de Desempenho para Sistemas de Maior Ordem

Nesta parte, será mostrada uma generalização dos cálculos da seção anterior para sistemas LTI de ordem $n > 1$, e valem as considerações da Seção 2.2.5. Seja $\mathbf{F} = (f_1, f_2, f_3)$, em que f_1, f_2 e f_3 pertencem a $\{f_{tr}, f_{tp}, f_{mp}, f_{ts}\}$ e são dois a dois distintos, e seja $\mathbf{K} = (K_P, K_I, K_D)$. Definindo $\mathbf{P} = (p_1, \dots, p_{n+1})$ e $\mathbf{R} = (r_1, \dots, r_{n+1})$, aplica-se a mesma regra da cadeia da Equação (3.1) para calcular o jacobiano de \mathbf{F} em relação a \mathbf{K} , com a diferença nas dimensões das matrizes envolvidas. Neste caso, tem-se que $\mathbf{J}_F(\mathbf{K}) \in M_{3 \times 3}(\mathbb{R})$, $\mathbf{J}_F(\mathbf{P}), \mathbf{J}_F(\mathbf{R}) \in M_{3 \times (n+1)}(\mathbb{C})$ e $\mathbf{J}_P(\mathbf{K}), \mathbf{J}_R(\mathbf{K}) \in M_{(n+1) \times 3}(\mathbb{C})$. Embora as matrizes resultantes da regra da cadeia possam ter elementos não reais, a simetria conjugada do produto entre elas irá garantir que o resultado final seja uma matriz real.

A matriz \mathbf{P} de polos é composta pelas raízes do polinômio $D(s)$ e serão determinadas numericamente pela função `ROOTS` do `MATLAB`. O cálculo da matriz \mathbf{R} , por sua vez, pode ter sua notação simplificada introduzindo o jacobiano do vetor de polinômios simétricos elementares (e_1, \dots, e_{n+1}) com relação a \mathbf{P} , definido por

$$\mathbf{J}_e(\mathbf{P}) = \begin{bmatrix} \frac{\partial e_1}{\partial p_1} & \cdots & \frac{\partial e_1}{\partial p_{n+1}} \\ \frac{\partial e_2}{\partial p_1} & \cdots & \frac{\partial e_2}{\partial p_{n+1}} \\ \vdots & \ddots & \vdots \\ \frac{\partial e_{n+1}}{\partial p_1} & \cdots & \frac{\partial e_{n+1}}{\partial p_{n+1}} \end{bmatrix} = \begin{bmatrix} 1 & \cdots & 1 \\ e_1^{(1)} & \cdots & e_1^{(n+1)} \\ \vdots & \ddots & \vdots \\ e_n^{(1)} & \cdots & e_n^{(n+1)} \end{bmatrix}, \quad (3.7)$$

que resume a Equação (2.82) para

$$\mathbf{R} = [\mathbf{J}_e(\mathbf{P})]^{-1} \mathbf{A}, \quad (3.8)$$

em que

$$\mathbf{A} = \begin{bmatrix} -a_0 \\ a_1 \\ \vdots \\ (-1)^{n+1} a_n \end{bmatrix}. \quad (3.9)$$

A existência da matriz inversa de $\mathbf{J}_e(\mathbf{P})$ está condicionada a uma distinção entre todos os $(n + 1)$ polos, isto porque

$$\det [\mathbf{J}_e(\mathbf{P})] = \prod_{1 \leq i < j \leq n+1} (p_i - p_j), \quad (3.10)$$

que é chamado de determinante de *Vandermonde*, cujo valor é zero se, e somente se, existir quaisquer $p_i = p_j$ para $i \neq j$. A demonstração dessa proposição vai além do escopo deste trabalho, mas caso o leitor se interessar por mais detalhes, confira o trabalho redigido por Lascoux e Pragacz em [38].

De maneira similar ao caso de primeira ordem, $\mathbf{J}_F(\mathbf{P})$ e $\mathbf{J}_F(\mathbf{R})$ serão formadas, respectivamente, pelas linhas das matrizes

$$\begin{bmatrix} \frac{\partial f_{tr}}{\partial \mathbf{P}} \\ \frac{\partial f_{tp}}{\partial \mathbf{P}} \\ \frac{\partial f_{mp}}{\partial \mathbf{P}} \\ \frac{\partial f_{ts}}{\partial \mathbf{P}} \end{bmatrix} = \begin{bmatrix} r_1 t_r e^{p_1 t_r} & \cdots & r_{n+1} t_r e^{p_{n+1} t_r} \\ (r_1 + r_1 p_1 t_p) e^{p_1 t_p} & \cdots & (r_{n+1} + r_{n+1} p_{n+1} t_p) e^{p_{n+1} t_p} \\ r_1 t_p e^{p_1 t_p} & \cdots & r_{n+1} t_p e^{p_{n+1} t_p} \\ r_1 t_s e^{p_1 t_s} & \cdots & r_{n+1} t_s e^{p_{n+1} t_s} \end{bmatrix} \quad (3.11)$$

e

$$\begin{bmatrix} \frac{\partial f_{tr}}{\partial \mathbf{R}} \\ \frac{\partial f_{tp}}{\partial \mathbf{R}} \\ \frac{\partial f_{mp}}{\partial \mathbf{R}} \\ \frac{\partial f_{ts}}{\partial \mathbf{R}} \end{bmatrix} = \begin{bmatrix} e^{p_1 t_r} & \dots & e^{p_{n+1} t_r} \\ p_1 e^{p_1 t_p} & \dots & p_{n+1} e^{p_{n+1} t_p} \\ e^{p_1 t_p} & \dots & e^{p_{n+1} t_p} \\ e^{p_1 t_s} & \dots & e^{p_{n+1} t_s} \end{bmatrix}, \quad (3.12)$$

de acordo com as derivadas das equações em (2.84). Agora, porém, somente uma das linhas é removida das matrizes para gerar os jacobianos, visto que três critérios de desempenho devem ser estipulados para o controlador.

Enquanto o sistema em (2.76) é não linear, ao derivar todas suas equações com relação a um dos coeficientes PID, por exemplo K_P , obtém-se equações lineares com relação às derivadas $\partial p_k / \partial K_P$, como mostrado abaixo:

$$\left\{ \begin{array}{l} \frac{\partial e_1}{\partial K_P} = \frac{\partial e_1}{\partial p_1} \frac{\partial p_1}{\partial K_P} + \dots + \frac{\partial e_1}{\partial p_{n+1}} \frac{\partial p_{n+1}}{\partial K_P} = -b_1 \\ \frac{\partial e_2}{\partial K_P} = \frac{\partial e_2}{\partial p_1} \frac{\partial p_1}{\partial K_P} + \dots + \frac{\partial e_2}{\partial p_{n+1}} \frac{\partial p_{n+1}}{\partial K_P} = b_2 \\ \vdots \\ \frac{\partial e_{n+1}}{\partial K_P} = \frac{\partial e_{n+1}}{\partial p_1} \frac{\partial p_1}{\partial K_P} + \dots + \frac{\partial e_{n+1}}{\partial p_{n+1}} \frac{\partial p_{n+1}}{\partial K_P} = (-1)^{n+1} b_{n+1}. \end{array} \right. \quad (3.13)$$

Se, ao invés de K_P , derivar-se cada e_k por K_I ou K_D , encontra-se um sistema similar a (3.13), tendo como diferenças a variável de diferenciação das derivadas de p_k e o coeficiente do lado direito de cada equação, que será b_{k-1} ou b_{k+1} na k -ésima equação. As derivadas $\partial e_k / \partial p_j$, portanto, permanecem as mesmas, de forma que os três sistemas obtidos podem ser expressos pelo produto entre os jacobianos $\mathbf{J}_e(\mathbf{P})$ e $\mathbf{J}_P(\mathbf{K})$ no lado esquerdo da igualdade e pela matriz

$$\mathbf{B} = \begin{bmatrix} -b_1 & -b_0 & -b_2 \\ b_2 & b_1 & b_3 \\ \vdots & \vdots & \vdots \\ (-1)^{n+1} b_{n+1} & (-1)^{n+1} b_n & (-1)^{n+1} b_{n+2} \end{bmatrix} \quad (3.14)$$

no lado direito. Dito isso, deduz-se que

$$\mathbf{J}_P(\mathbf{K}) = [\mathbf{J}_e(\mathbf{P})]^{-1} \mathbf{B}. \quad (3.15)$$

Neste ponto, ao continuar a seguir os procedimentos feitos nos sistemas de primeira ordem, deve-se derivar o sistema que associa os resíduos com os coeficientes do controlador com relação a \mathbf{K} . Esse sistema equivale à Equação (3.8) na forma matricial, mas ao aplicar a derivada nessa expressão, dentre os termos emergentes, estão a derivada de uma matriz com relação a um vetor, algo que necessitaria do uso de dimensões superiores e elementos tensoriais para manter o rigor matemático. Para evitar eventuais complicações, deriva-se (3.8) com relação a somente um dos coeficientes PID e deduz-se as outras duas a partir do resultado de uma, visto que as diferenças entre elas serão mínimas e evidentes. No final, $\mathbf{J}_R(\mathbf{K})$ será composta pela concatenação das derivadas resultantes ao longo de suas colunas.

Derivando (3.8) com relação a K_P :

$$\mathbf{J}_e(\mathbf{P}) \frac{\partial \mathbf{R}}{\partial K_P} + \frac{\partial \mathbf{J}_e(\mathbf{P})}{\partial K_P} \mathbf{R} = 0 \implies \frac{\partial \mathbf{R}}{\partial K_P} = -[\mathbf{J}_e(\mathbf{P})]^{-1} \frac{\partial \mathbf{J}_e(\mathbf{P})}{\partial K_P} \mathbf{R}. \quad (3.16)$$

O cálculo acima foi realizado primeiro passando a inversa do jacobiano para o lado esquerdo da igualdade e, depois, aplicando a derivada com relação a K_P pela regra do produto. As matrizes $\mathbf{J}_e(\mathbf{P})$ e \mathbf{R} já são conhecidas, enquanto a derivada do jacobiano situada entre eles ainda não foi determinada. Será mostrado a seguir que, ao derivar individualmente cada elemento desse jacobiano e desenvolver o resultado, a k -ésima linha dessa matriz pode ser expressa como a transposta do produto de uma matriz com todas as derivadas de segunda ordem do polinômio e_k com o vetor de derivadas de \mathbf{P} com relação a K_P :

$$\begin{aligned} \frac{\partial \mathbf{J}_e(\mathbf{P})}{\partial K_P} &= \begin{bmatrix} \frac{\partial}{\partial K_P} \left(\frac{\partial e_1}{\partial p_1} \right) & \cdots & \frac{\partial}{\partial K_P} \left(\frac{\partial e_1}{\partial p_{n+1}} \right) \\ \vdots & \ddots & \vdots \\ \frac{\partial}{\partial K_P} \left(\frac{\partial e_{n+1}}{\partial p_1} \right) & \cdots & \frac{\partial}{\partial K_P} \left(\frac{\partial e_{n+1}}{\partial p_{n+1}} \right) \end{bmatrix} \\ &= \begin{bmatrix} \frac{\partial^2 e_1}{\partial p_1^2} \frac{\partial p_1}{\partial K_P} + \cdots + \frac{\partial^2 e_1}{\partial p_{n+1} \partial p_1} \frac{\partial p_{n+1}}{\partial K_P} & \cdots & \frac{\partial^2 e_1}{\partial p_1 \partial p_{n+1}} \frac{\partial p_1}{\partial K_P} + \cdots + \frac{\partial^2 e_1}{\partial p_{n+1}^2} \frac{\partial p_{n+1}}{\partial K_P} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 e_{n+1}}{\partial p_1^2} \frac{\partial p_1}{\partial K_P} + \cdots + \frac{\partial^2 e_{n+1}}{\partial p_{n+1} \partial p_1} \frac{\partial p_{n+1}}{\partial K_P} & \cdots & \frac{\partial^2 e_{n+1}}{\partial p_1 \partial p_{n+1}} \frac{\partial p_1}{\partial K_P} + \cdots + \frac{\partial^2 e_{n+1}}{\partial p_{n+1}^2} \frac{\partial p_{n+1}}{\partial K_P} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{H}_{e_1}(\mathbf{P}) \frac{\partial \mathbf{P}}{\partial K_P} & \cdots & \mathbf{H}_{e_{n+1}}(\mathbf{P}) \frac{\partial \mathbf{P}}{\partial K_P} \end{bmatrix}^T, \end{aligned}$$

Cada $\mathbf{H}_{e_k}(\mathbf{P})$ será chamada de matriz *hessiana* de e_k com relação a \mathbf{P} e suas componentes são as derivadas de segunda ordem $\partial^2 e_k / \partial p_i \partial p_j$, com $i, j = 1, \dots, n+1$. Em primeiro lugar, se $i \neq j$, pode-se tomar como base o comportamento da primeira derivação na Equação (2.80) e

afirmar que uma segunda derivação resultaria na soma de todos os produtos de $(k-2)$ polos, dois a dois distintos, e diferentes de p_i e p_j , combinação esta que será denotada por $e_{k-2}^{(j,i)}$. Contudo, se $i = j$, estaria se derivando por p_i um polinômio em que todas as instâncias desse polo foram removidas durante a primeira derivação e seria obtido zero. Algebricamente:

$$\frac{\partial^2 e_k}{\partial p_i \partial p_j} = \frac{\partial}{\partial p_i} \left(e_{k-1}^{(j)} \right) = \begin{cases} 0, & \text{se } k = 1 \text{ ou } i = j \\ e_{k-2}^{(j,i)}, & \text{caso contrário.} \end{cases} \quad (3.17)$$

Logo, $\mathbf{H}_{e_1}(\mathbf{P})$ é a matriz nula $(n+1) \times (n+1)$, enquanto que, para $k = 2, \dots, n+1$,

$$\mathbf{H}_{e_k}(\mathbf{P}) = \begin{bmatrix} 0 & e_{k-2}^{(1,2)} & \cdots & e_{k-2}^{(1,n+1)} \\ e_{k-2}^{(2,1)} & 0 & \cdots & e_{k-2}^{(2,n+1)} \\ \vdots & \vdots & \ddots & \vdots \\ e_{k-2}^{(n+1,1)} & e_{k-2}^{(n+1,2)} & \cdots & 0 \end{bmatrix}. \quad (3.18)$$

A forma das matrizes hessianas é indiferente quanto à variável de diferenciação escolhida no início dos cálculos, pois uma alteração nesse quesito só afetaria o segundo termo durante a aplicação da regra da cadeia em $\partial \mathbf{J}_e(\mathbf{P}) / \partial K_P$. Em outras palavras, uma substituição de K_P por K_I ou por K_D na Equação (3.16) e no desenvolvimento da derivada de $\mathbf{J}_e(\mathbf{P})$ manteria a validade dessas igualdades. Assim sendo, obtém-se

$$\begin{cases} \frac{\partial \mathbf{J}_e(\mathbf{P})}{\partial K_P} = \left[\mathbf{H}_{e_1}(\mathbf{P}) \frac{\partial \mathbf{P}}{\partial K_P} \quad \cdots \quad \mathbf{H}_{e_{n+1}}(\mathbf{P}) \frac{\partial \mathbf{P}}{\partial K_P} \right]^T \\ \frac{\partial \mathbf{J}_e(\mathbf{P})}{\partial K_I} = \left[\mathbf{H}_{e_1}(\mathbf{P}) \frac{\partial \mathbf{P}}{\partial K_I} \quad \cdots \quad \mathbf{H}_{e_{n+1}}(\mathbf{P}) \frac{\partial \mathbf{P}}{\partial K_I} \right]^T \\ \frac{\partial \mathbf{J}_e(\mathbf{P})}{\partial K_D} = \left[\mathbf{H}_{e_1}(\mathbf{P}) \frac{\partial \mathbf{P}}{\partial K_D} \quad \cdots \quad \mathbf{H}_{e_{n+1}}(\mathbf{P}) \frac{\partial \mathbf{P}}{\partial K_D} \right]^T, \end{cases} \quad (3.19)$$

e, finalmente, a expressão para o jacobiano dos resíduos:

$$\mathbf{J}_R(\mathbf{K}) = \left[-[\mathbf{J}_e(\mathbf{P})]^{-1} \frac{\partial \mathbf{J}_e(\mathbf{P})}{\partial K_P} \mathbf{R} \quad -[\mathbf{J}_e(\mathbf{P})]^{-1} \frac{\partial \mathbf{J}_e(\mathbf{P})}{\partial K_I} \mathbf{R} \quad -[\mathbf{J}_e(\mathbf{P})]^{-1} \frac{\partial \mathbf{J}_e(\mathbf{P})}{\partial K_D} \mathbf{R} \right]. \quad (3.20)$$

3.3 Algoritmo de Sintonia de Controladores PID pelos Métodos de Newton-Raphson e Levenberg-Marquardt

O algoritmo de sintonia proposto recebe como argumentos a função de transferência da planta ($G(s)$), obedecendo às restrições impostas nas Seções 2.2.4 e 2.2.5, os critérios de desempenho desejados para a resposta ao degrau do sistema ($t_r, t_p, M_p, t_s, \alpha$), o valor inicial para os coeficientes do controlador PI ou PID ($\mathbf{K}^{(0)}$), o método de resolução das equações não lineares e algumas especificações do critério de parada. A estrutura do algoritmo compõe um laço no qual se realizam os incrementos do método iterativo escolhido sobre os coeficientes do controlador e, mediante ao atendimento de um dos critérios de parada, o valor de \mathbf{K} na última iteração é retornado, caso uma solução for encontrada. Caso contrário, o retorno é uma mensagem de erro indicando o motivo da falha.

Nos critérios de parada, adotou-se uma tolerância $\epsilon > 0$, para quando \mathbf{K} for tal que

$$\|\mathbf{F}(\mathbf{K})\| < \epsilon, \quad (3.21)$$

indicar que os critérios de desempenho estão suficientemente próximos do desejado. Além disso, limitou-se o tempo de execução do programa estabelecendo um número máximo de iterações $N_{max} > 0$. Se este for atingido, pode ser necessário alterar o “palpite” inicial ou reduzir a tolerância, sem garantia de sucesso quando não houver certeza da existência de uma solução.

A implementação computacional de grande parte do algoritmo é bastante direta em softwares especializados como o MATLAB. A maior complexidade, a nível de programação, está no cálculo de cada um dos PSE's que compõem as matrizes jacobianas e hessianas. Para descomplicar a obtenção desses polinômios, considere o exemplo abaixo de um PSE de grau 3 em 4 polos:

$$\begin{aligned} e_3(p_1, p_2, p_3, p_4) &= p_1 p_2 p_3 + p_1 p_2 p_4 + p_1 p_3 p_4 + p_2 p_3 p_4 \\ &= p_1 (p_2 p_3 + p_2 p_4 + p_3 p_4) + p_2 p_3 p_4 \\ &= p_1 e_2(p_2, p_3, p_4) + p_2 e_2(p_3, p_4). \end{aligned}$$

Nessa expressão, os polos p_1 e p_2 multiplicam PSE's de grau 2 contendo os respectivos polos que os sucedem. Esse comportamento segue nos PSE's de grau 2, obtendo-se o produto dos polos com PSE's de grau 1. Generalizando, um PSE de grau k contém nele PSE's de grau

$(k - 1)$, estes que contém PSE's de grau $(k - 2)$, e assim por diante. Com essa propriedade em mente, o Algoritmo 1 mostra uma implementação recursiva para o cálculo do PSE de grau k nas variáveis do vetor P .

Algoritmo 1 Polinômio Simétrico Elementar

```

1: function PSE(product, P, k)
2:   if k == 0 then
3:     return product
4:   sum = 0
5:   n = LENGTH(P)
6:   for i = 1 to n do
7:     sum = sum + PSE(product * P(i), P(i + 1 : n), k - 1)
8:   return sum

```

Fonte: Autoria própria.

Com o auxílio dessa função, o cálculo das matrizes jacobiana e hessianas torna-se trivial. Conforme a Equação (3.7), o kj -ésimo elemento de $\mathbf{J}_e(\mathbf{P})$ é um PSE de grau $(k - 1)$ no conjunto de polos obtido ao retirar o j -ésimo polo. Essa definição é explorada no Algoritmo 2, em que a variável J_e é inicializada como uma matriz $n \times n$ de zeros, e o cálculo de cada um de seus elementos envolve uma chamada da função PSE() com os argumentos apropriados. A matriz retornada corresponderá ao jacobiano $\mathbf{J}_e(\mathbf{P})$.

Algoritmo 2 Jacobiano dos Polinômios Simétricos Elementares

```

1: function PSEJACOBIAN(P)
2:   n = LENGTH(P)
3:   Je = ZEROS(n, n)
4:   for k = 1 to n do
5:     for j = 1 to n do
6:       S = [P(1 : j - 1); P(j + 1 : n)]
7:       Je(k, j) = PSE(1, S, k - 1)
8:   return Je

```

Fonte: Autoria própria.

De forma semelhante, no caso dos hessianos, para calcular o ij -ésimo elemento de $\mathbf{H}_{e_k}(\mathbf{P})$, retiram-se os polos de índice i e j e calcula-se o PSE de grau $(k - 2)$ sobre esse conjunto, vide a Equação (3.18). Ademais, pela propriedade de simetria das matrizes hessianas, $e_k^{(i,j)} = e_k^{(j,i)}$, os elementos abaixo da diagonal principal são equivalentes aos elementos que estão acima, o que reduz a quantidade de cálculos pela metade. O Algoritmo 3 inicializa He como uma matriz tridimensional com dimensões $n \times n \times n$, chama a função PSE() quando necessário e retorna as n matrizes hessianas dos PSE's.

Algoritmo 3 Hessiano dos Polinômios Simétricos Elementares

```

1: function PSEHESSIAN( $P$ )
2:    $n = \text{LENGTH}(P)$ 
3:    $He = \text{ZEROS}(n, n, n)$ 
4:   for  $k = 2$  to  $n$  do
5:     for  $i = 1$  to  $n$  do
6:       for  $j = 1$  to  $n$  do
7:         if  $i < j$  then
8:            $S = [P(1 : i - 1); P(i + 1 : j - 1); P(j + 1 : n)]$ 
9:            $He(i, j, k) = \text{PSE}(1, S, k - 2)$ 
10:        else if  $i > j$  then
11:           $He(i, j, k) = He(j, i, k)$ 
12:   return  $He(:, :, 1), \dots, He(:, :, n)$ 

```

Fonte: Autoria própria.

É importante salientar que a variável n dos Algoritmos 1, 2 e 3 não corresponde à ordem da planta, afinal, se $G(s)$ é de ordem n , a função de transferência em malha fechada operando sob um controle PID tem $(n + 1)$ polos, quantidade esta que define a ordem do jacobiano e dos hessianos.

Com relação aos métodos numéricos, uma opção entre duas regras de atualização do vetor \mathbf{K} são oferecidas pelo algoritmo, a primeira correspondente à Equação (2.89) do método de Newton-Raphson (NR), e a segunda correspondente à Equação (2.91) do método de Levenberg-Marquardt (LM). Após testes preliminares com alguns dos parâmetros de *damping* recomendados em [35, 39], optou-se pela expressão

$$\lambda_k = \mathbf{F}(\mathbf{K})^T \mathbf{F}(\mathbf{K}) \quad (3.22)$$

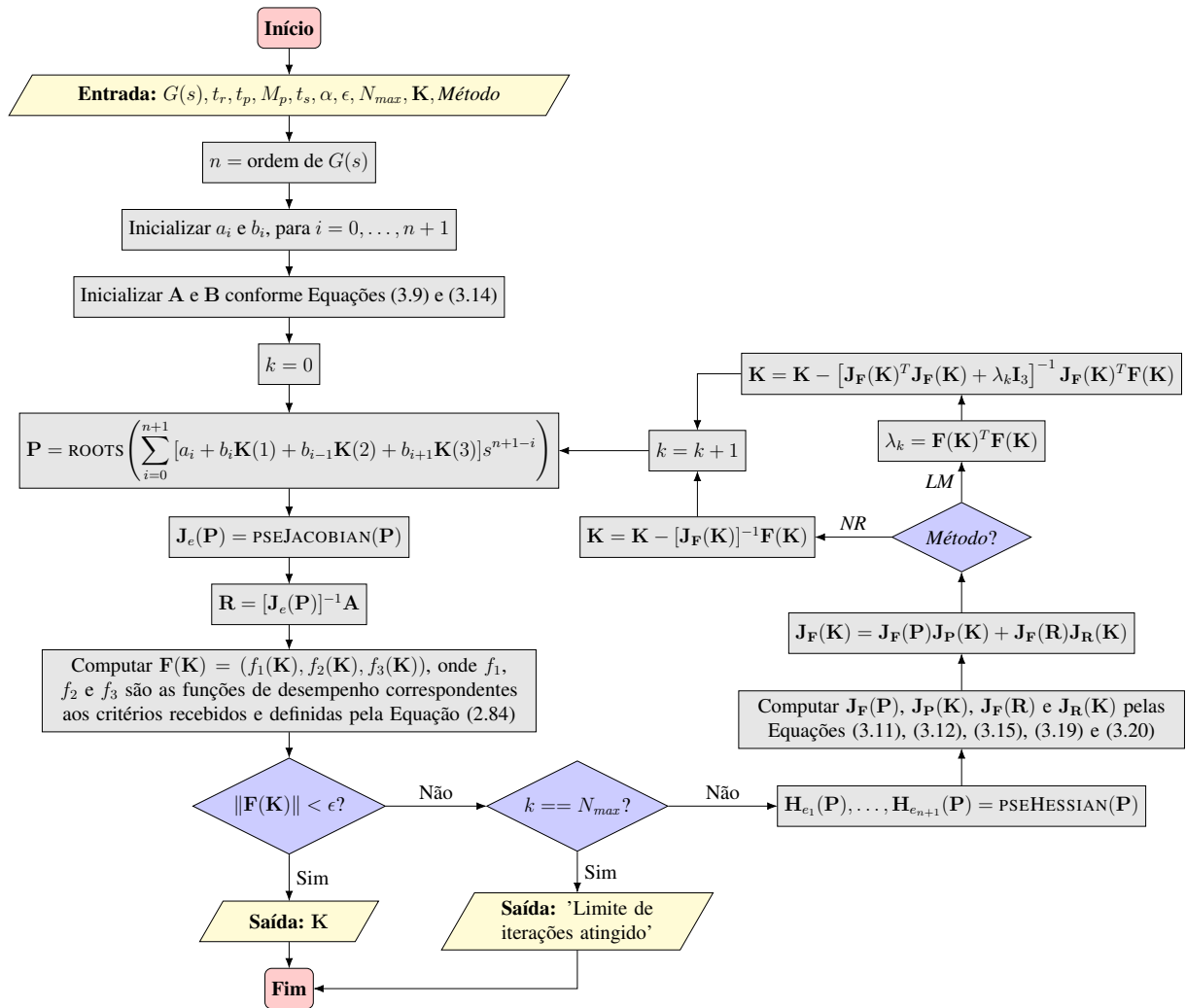
para integrar a regra de atualização do método LM.

A Figura 3.1 apresenta o fluxograma do algoritmo para a sintonia do controlador. Vale lembrar que somente dois (para um controlador PI) ou três (para um controlador PID) dos critérios de desempenho podem ser delimitados por vez. Uma forma simples de informar ao programa quais deles não devem ser considerados é atribuir um valor de referência que os parâmetros jamais assumiriam em um caso real, como 0 ou -1 , e adicionar esse condicional nos blocos constituintes que dependem dessa informação. A função `ROOTS` do MATLAB calcula as raízes do polinômio do denominador da Equação (2.72) e atribui o resultado ao vetor \mathbf{P} .

Apesar da teoria de sistemas de primeira ordem ter sido apresentada separadamente, a fim de facilitar o entendimento, não há necessidade de um algoritmo exclusivo para estes casos.

Isto porque, ainda que não há menção explícita dos PSE's e de seus jacobianos e hessianos, eles estão presentes nas equações da Seção 3.1, de forma análoga às equações de sistemas de ordem n . Essencialmente, basta fazer $n = 1$ nas equações do caso genérico e retirar a terceira coluna dos jacobianos com relação a \mathbf{K} , por conta da ausência do termo derivativo nos sistemas de primeira ordem.

Figura 3.1. Fluxograma do algoritmo proposto.



Fonte: Autoria própria.

4 RESULTADOS E DISCUSSÕES

Testes do algoritmo foram efetuados em três funções de transferência escolhidas arbitrariamente, de primeira, segunda e quarta ordem. Em cada teste, analisaram-se três combinações distintas de critérios de desempenho e fixou-se a margem do tempo de assentamento em $\alpha = 2\%$ para todos os casos. A tolerância optada foi de $\epsilon = 10^{-3}$ e o número máximo de iterações N_{max} foi escolhido de acordo com a ordem do sistema, sendo 10^4 para o sistema de primeira ordem e 200 para os sistemas de segunda e quarta ordem, devido à maior quantidade de cálculos realizados por iteração nesses dois últimos.

Para analisar a performance do algoritmo de sintonia, primeiramente, estabeleceram-se vários valores iniciais para os coeficientes do controlador e, em um espaço cartesiano, exibiu-se como esses coeficientes são alterados após cada iteração. Assim, o caminho traçado pelos métodos numéricos em cada combinação de critérios foi facilmente visualizado, o que proporciona uma intuição quanto às diferentes estratégias de otimização dos métodos NR e LM, como o tamanho do passo e a taxa de convergência. Tabelas com a precisão, o tempo de execução e o número de iterações em cada caso estão dispostas para uma análise quantitativa da performance do algoritmo. Além do mais, traçaram-se as curvas de resposta ao degrau em cada ponto de convergência obtido para determinar a corretude do resultado.

Em segundo lugar, estudou-se a taxa de convergência dos métodos quando a curva de resposta do sistema já possui parâmetros de desempenho próximos do desejado no ponto inicial. Com isso, buscaram-se indicações para um aumento na taxa de convergência ao fazer uma sintonia prévia do controlador por algum dos métodos clássicos, como tentativa e erro ou os métodos de Ziegler-Nichols, e usar o resultado dessa sintonia como “palpite” inicial do algoritmo desenvolvido, com o intuito de minimizar o erro.

4.1 Primeiro Caso: Sistema de Primeira Ordem

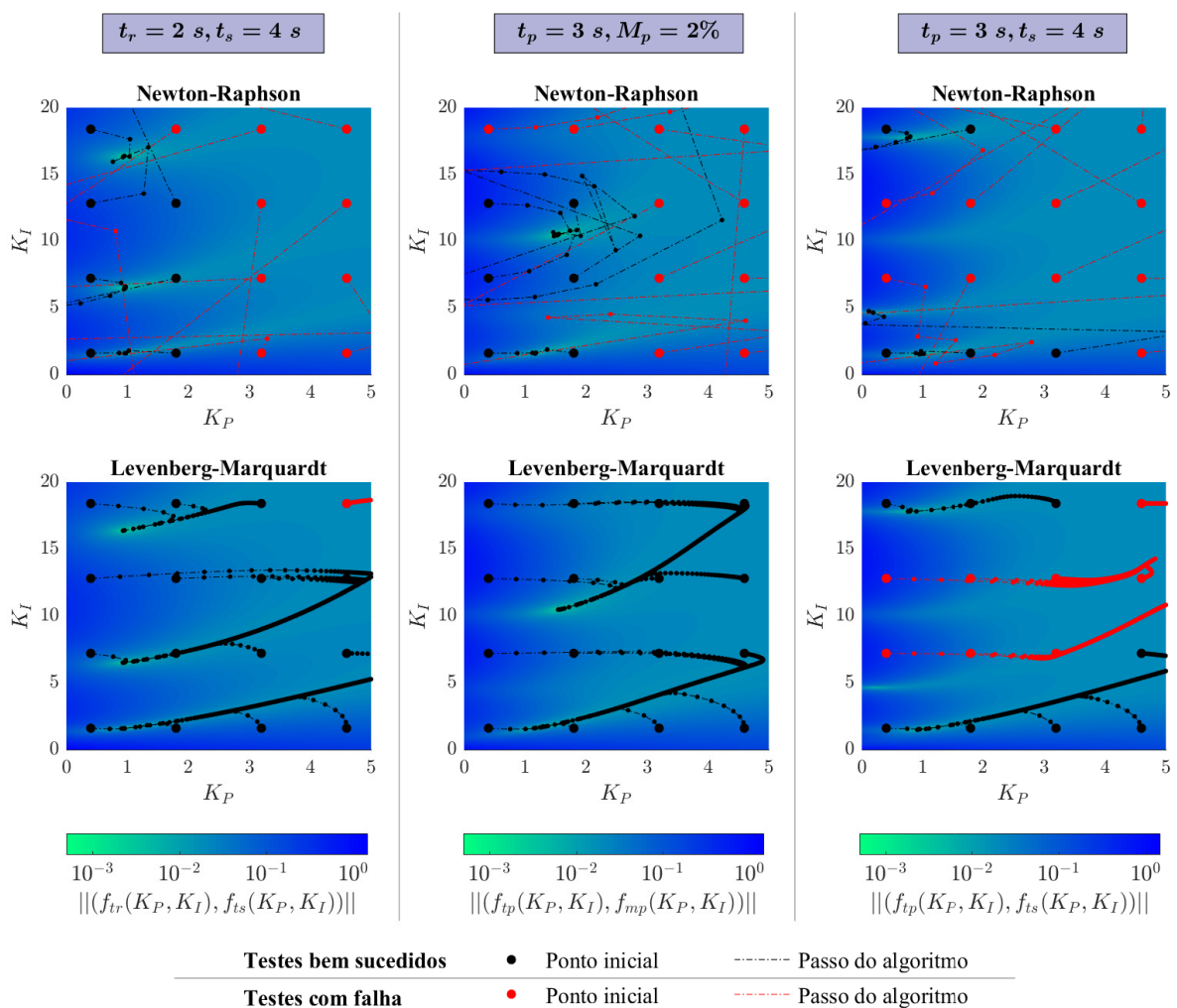
O primeiro sistema tem como função de transferência

$$G(s) = \frac{1}{s + 1}, \quad (4.1)$$

e a aplicação do algoritmo visou sintonizar um controlador PI operando sobre esse processo. As combinações de critérios de desempenho foram: tempo de subida e tempo de assentamento; tempo de pico e ultrapassagem percentual; tempo de pico e tempo de assentamento.

Os gráficos da Figura 4.1 mostram os resultados da simulação com \mathbf{K} iniciando a partir de dezesseis pontos uniformemente distribuídos, para os critérios de desempenho listados acima de cada gráfico. As linhas que conectam cada ponto representam o caminho traçado pelo algoritmo de uma iteração para a outra e o mapa de cor constitui o valor assumido pela norma das funções de desempenho para uma variação contínua de K_P e K_I . A tabela do Apêndice A expõe detalhadamente os dados coletados durante a execução do programa, enquanto a Tabela 4.1 mostra as informações gerais de performance do algoritmo.

Figura 4.1. Caminho traçado pelo algoritmo nos métodos NR e LM para o sistema de primeira ordem, tendo $\mathbf{K}^{(0)} \in \{0,4; 1,8; 3,2; 4,6\} \times \{1,6; 7,2; 12,8; 18,4\}$ como pontos iniciais.



Fonte: Autoria própria.

Pela análise dos gráficos, nota-se de imediato uma diferença intrínseca entre os passos dos dois métodos. A distância entre os pontos de iterações consecutivas do método NR tende a ser maior, e a convergência é atingida em menos iterações, em comparação ao método LM. Porém, por ser mais direto, o método NR é menos preciso, sendo comum divergir quando o ponto inicial está distante de uma solução. O método LM, em contrapartida, é mais robusto e

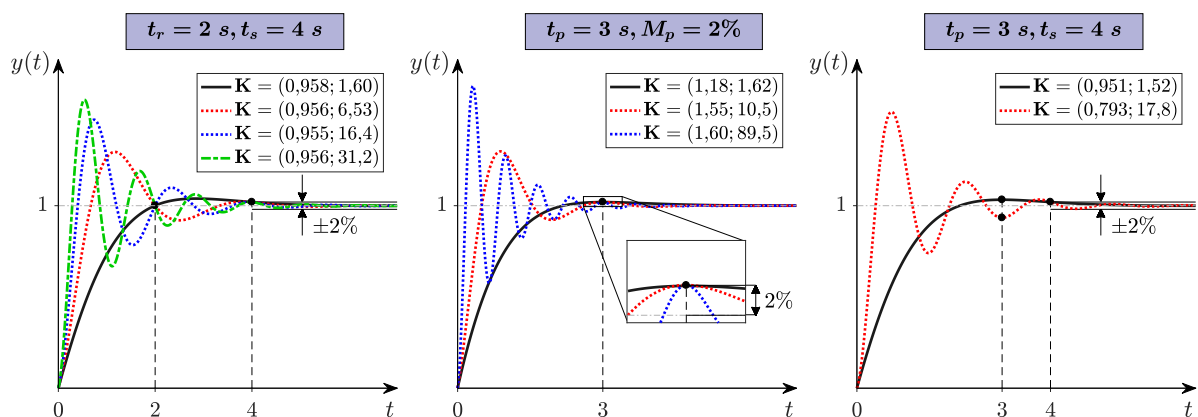
Tabela 4.1. Performance do algoritmo de sintonia do controlador no sistema de primeira ordem.

	$t_r = 2\text{ s}, t_s = 4\text{ s}$		$t_p = 3\text{ s}, M_p = 2\%$		$t_p = 3\text{ s}, t_s = 4\text{ s}$	
	NR	LM	NR	LM	NR	LM
Precisão	21/64	58/64	18/64	64/64	16/64	36/64
Tempo médio de execução (ms)	4,02	167	1,64	51,1	1,47	209
Número médio de iterações	4,90	1254	5,67	430	4,63	1564
Tempo médio de execução por iteração (ms/iter)	0,819	0,133	0,289	0,119	0,318	0,134

Fonte: Autoria própria.

tem um tempo de execução por iteração menor, provavelmente devido aos cálculos com valores menores na obtenção da inversa da regra de atualização, mas, no total, leva um tempo maior para atingir a convergência por conta do alto número de iterações. Nos pontos em que o método LM não encontra uma solução, como visto na terceira coluna de gráficos, um aumento no limite de iterações deve ser o suficiente para corrigir essa falha.

Ainda na Figura 4.1, é perceptível que cada um dos critérios gera um sistema de equações com múltiplas soluções para a tolerância escolhida, visível nas regiões mais claras dos mapas de cor. Entretanto, encontrar uma solução para o sistema de equações não significa que os critérios de desempenho são atendidos, já que as funções de desempenho dão espaço para uma interpretação errônea dos parâmetros na curva de resposta. Para esclarecer essa proposição, a Figura 4.2 mostra as curvas de resposta de todas as soluções encontradas a partir dos dezesseis pontos iniciais.

Figura 4.2. Curvas de resposta ao degrau alcançadas conforme a sintonia do algoritmo para o sistema de primeira ordem.

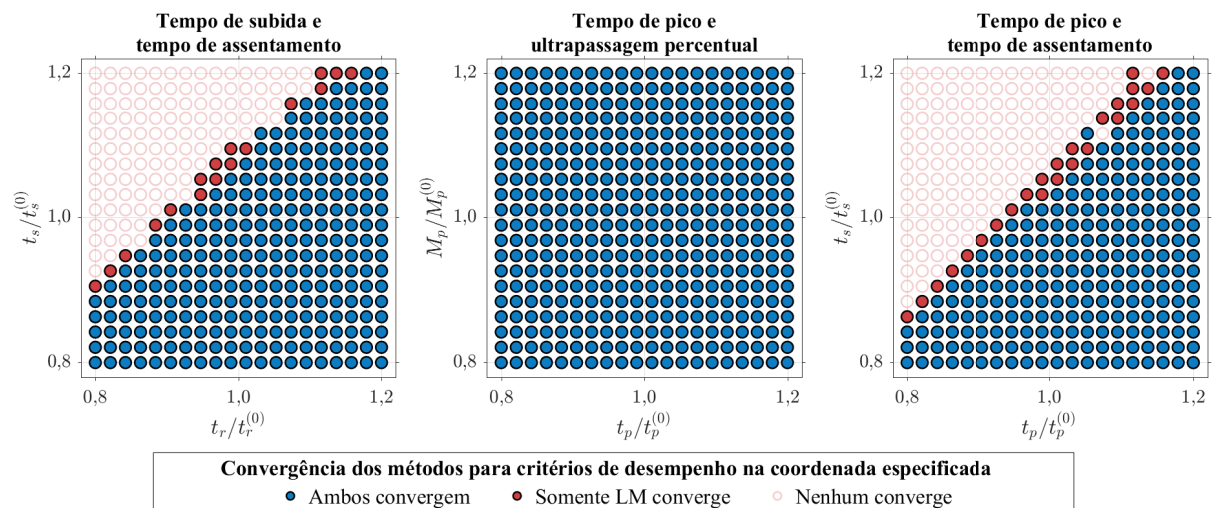
Fonte: Autoria própria.

Na primeira combinação de critérios, quatro diferentes curvas de resposta são obtidas pela sintonia do algoritmo, mas somente uma delas atende aos critérios corretamente. Todas as curvas permanecem dentro da margem de $\pm 2\%$ após o tempo de assentamento escolhido de 4 s , mas somente a curva em preto tem um tempo de subida de 2 s , pois as outras três já ultrapas-

saram a referência antes desse instante. Observe que as quatro curvas têm aproximadamente o mesmo coeficiente proporcional, mas coeficientes integrais distintos, e o aumento de K_I intensifica as oscilações e eleva a velocidade de resposta, fazendo com que cada curva passe pelo ponto (2; 1) em diferentes semiciclos da oscilação. Na segunda e terceira combinações, novamente, somente a curva em preto atende a todos os critérios adequadamente, já que o tempo de pico das outras curvas ocorrem antes do desejado.

Para concluir a análise do caso de primeira ordem, fixou-se $\mathbf{K}^{(0)} = (1; 2)$, que gera uma curva de resposta com as especificações $t_r^{(0)} = 1,57$ s, $t_p^{(0)} = 2,36$ s, $M_p^{(0)} = 6,70\%$ e $t_s^{(0)} = 3,73$ s. A convergência do algoritmo foi investigada quando os critérios estabelecidos estão dentro de uma margem de $\pm 20\%$ dos parâmetros de desempenho no ponto inicial. A Figura 4.3 apresenta os resultados desses testes.

Figura 4.3. Convergência dos métodos no sistema de primeira ordem para 400 valores distintos dos critérios de desempenho em cada combinação, com $\mathbf{K}^{(0)} = (1; 2)$.



Fonte: Autoria própria.

No primeiro e terceiro gráficos, soluções foram encontradas em grande parte dos pontos por ambos os métodos, com exceção de alguns nos quais somente o método LM convergiu, localizados na fronteira da região em que ambos falharam. No segundo gráfico, os dois métodos obtiveram soluções para todos os critérios testados. Obviamente que não se pode generalizar a alta taxa de convergência desse caso particular para quaisquer valores dos parâmetros de desempenho, mas pela simplicidade da função de transferência de primeira ordem e pelo alto grau de liberdade que o controlador PI oferece, sintonizar com antecedência o controlador até obter uma resposta não muito distante dos critérios desejados aumenta bastante a probabilidade de encontrar uma solução por ao menos um dos métodos.

4.2 Segundo Caso: Sistema de Segunda Ordem

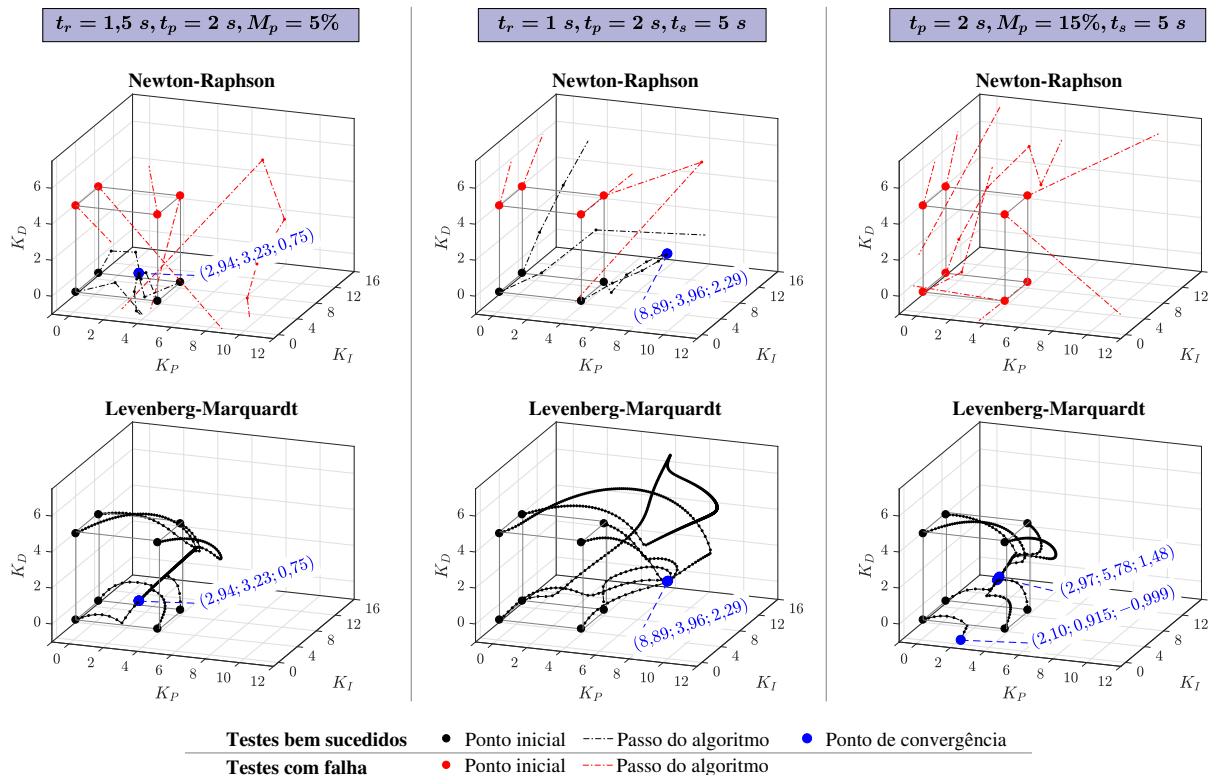
A função de transferência de segunda ordem escolhida é a seguinte:

$$G(s) = \frac{1}{s^2 + 2s + 2}. \quad (4.2)$$

Nesta situação, o algoritmo inclui o termo derivativo nos ajustes para possibilitar a restrição de três critérios de desempenho simultâneos e os gráficos estão dispostos em três dimensões. As combinações de critérios são: tempo de subida, tempo de pico e ultrapassagem percentual; tempo de subida, tempo de pico e tempo de assentamento; tempo de pico, ultrapassagem percentual e tempo de assentamento.

O gráfico tridimensional, agora necessário para visualização dos três coeficientes, impossibilita a inserção de um mapa de cor como na Figura 4.1, mas não torna inviável a análise dos métodos puramente pelos caminhos traçados. A Figura 4.4 mostra esses caminhos quando o algoritmo inicia em 8 diferentes pontos do gráfico $K_P \times K_I \times K_D$. Os detalhes dos testes simulados estão na tabela do Apêndice B e as informações gerais de performance estão na Tabela 4.2.

Figura 4.4. Caminho traçado pelo algoritmo nos métodos NR e LM para o sistema de segunda ordem, tendo $\mathbf{K}^{(0)} \in \{0,1; 4,9\} \times \{0,1; 4,9\} \times \{0,1; 4,9\}$ como pontos iniciais.



Fonte: Autoria própria.

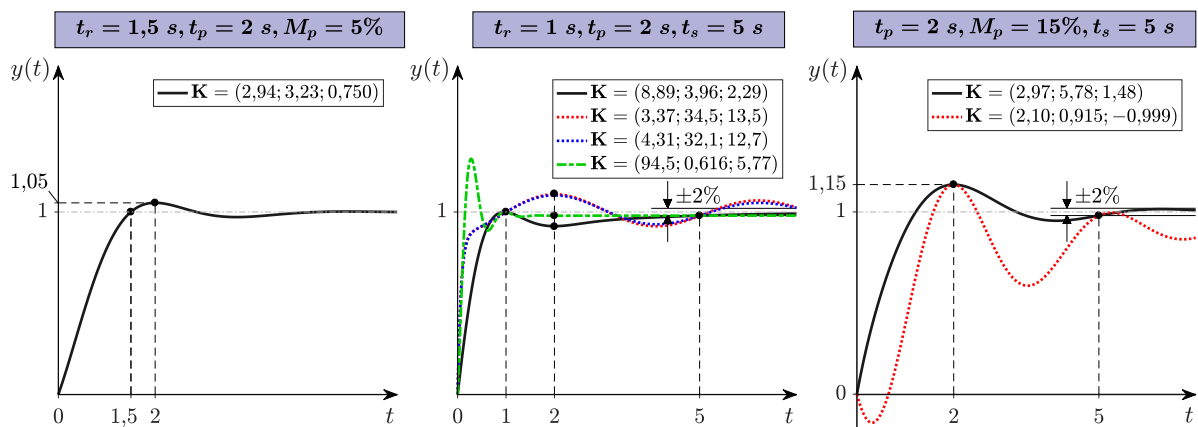
Tabela 4.2. Performance do algoritmo de sintonia do controlador no sistema de segunda ordem.

	$t_r = 1,5 s, t_p = 2 s, M_p = 5\%$		$t_r = 1 s, t_p = 2 s, t_s = 5 s$		$t_p = 2 s, M_p = 15\%, t_s = 5 s$	
	NR	LM	NR	LM	NR	LM
Precisão	22/64	52/64	29/64	44/64	6/64	64/64
Tempo médio de execução (ms)	398	1160	303	937	250	972
Número médio de iterações	5,95	54,9	7,28	43,5	5,17	45,5
Tempo médio de execução por iteração (ms/iter)	66,8	21,1	41,6	21,5	48,4	21,4

Fonte: Autoria própria.

As mesmas características observadas no caso anterior são vistas aqui. O método NR é mais veloz e menos preciso, enquanto o método LM sacrifica a rapidez por uma confiabilidade maior na obtenção do resultado. Mesmo limitando o número de iterações para um valor 50 vezes menor que os testes de primeira ordem, o tempo de execução total dos métodos é bem maior, dado que em toda iteração, as matrizes jacobianas envolvidas têm uma dimensão a mais.

Das múltiplas soluções obtidas (algumas fora do intervalo de visibilidade dos gráficos), nem todas condizem com os critérios desejados, como se vê na Figura 4.5. No primeiro gráfico, somente uma solução é encontrada e ela respeita seus respectivos critérios. No segundo, curiosamente, nenhuma das quatro respostas está correta. As curvas preta e verde não atendem o tempo de pico estabelecido, e as curvas azul e vermelha extrapolam a margem de 2% após o suposto tempo de assentamento. No terceiro gráfico, a curva preta tem parâmetros satisfatórios, enquanto a curva vermelha não atende o tempo de assentamento. Esta última curva também tem um comportamento distinto das outras, por originar de um K_D com valor negativo, que explica a presença de um *undershoot* inicial.

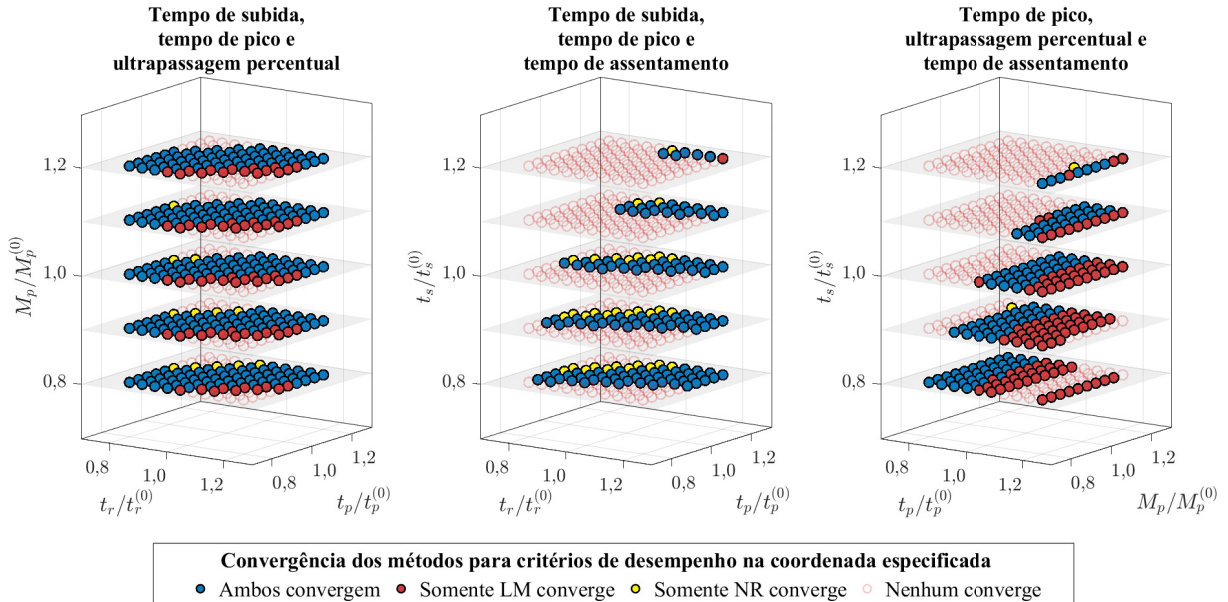
Figura 4.5. Curvas de resposta ao degrau alcançadas conforme a sintonia do algoritmo para o sistema de segunda ordem.

Fonte: Autoria própria.

Escolhendo-se o ponto inicial $\mathbf{K}^{(0)} = (1; 2; 0,5)$, cuja resposta é caracterizada por $t_r^{(0)} = 2,51 s$, $t_p^{(0)} = 3,46 s$, $M_p^{(0)} = 7,97\%$ e $t_s^{(0)} = 4,86 s$, os gráficos da Figura 4.6 ilustram os pontos

de convergência do algoritmo para critérios de desempenho dentro de uma margem de $\pm 20\%$ dos iniciais.

Figura 4.6. Convergência dos métodos no sistema de segunda ordem para 500 valores distintos dos critérios de desempenho em cada combinação, com $\mathbf{K}^{(0)} = (1; 2; 0,5)$.



Fonte: Autoria própria.

Na maioria dos pontos desses gráficos, ou ambos convergem ou somente o método LM converge. Os pontos amarelos do gráfico estão em minoria e provavelmente ocorrem devido à limitação no número de iterações, que prejudica o método LM por convergir mais lentamente.

4.3 Terceiro Caso: Sistema de Quarta ordem

O último sistema testado no algoritmo possui a função de transferência

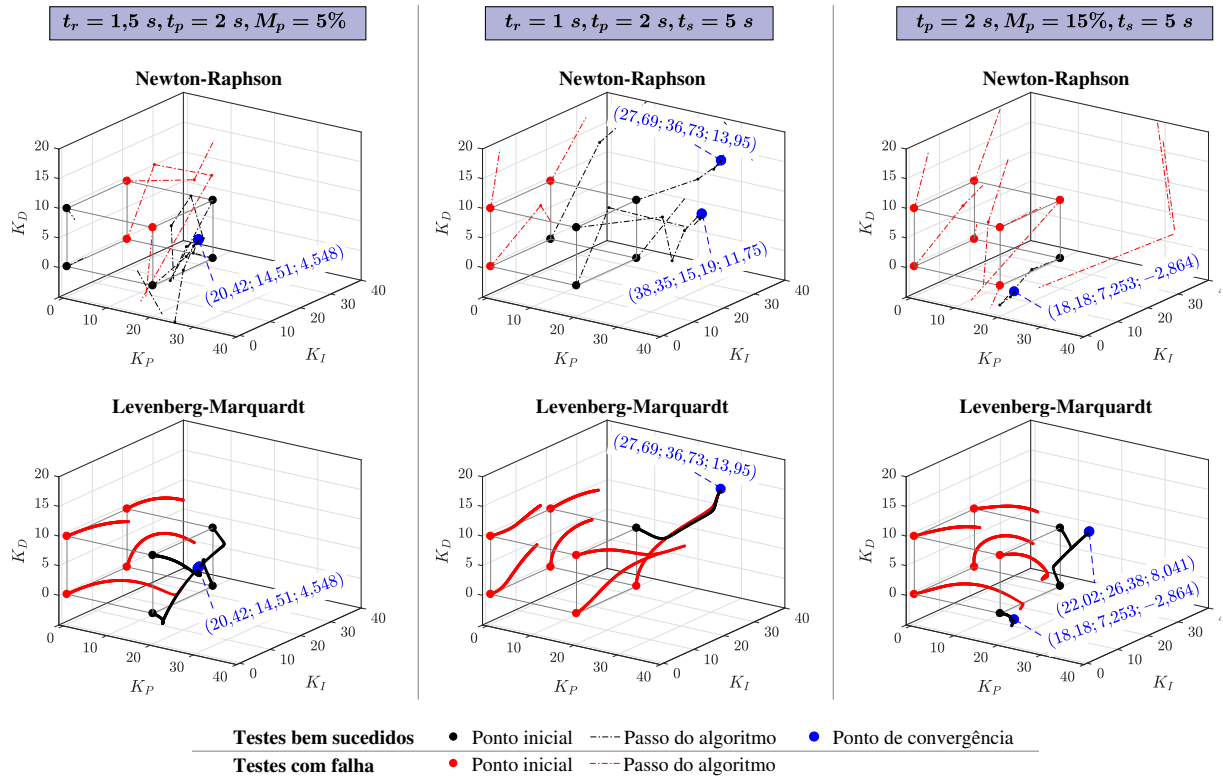
$$G(s) = \frac{s + 2,5}{(s + 1)(s + 2)(s + 3)(s + 4)}. \quad (4.3)$$

Os gráficos serão similares aos de segunda ordem, assim como as combinações de critérios.

Os gráficos com os caminhos percorridos pelo métodos iterativos com a função de transferência de quarta ordem se encontram na Figura 4.7, e os dados coletados durante os testes estão apresentados na tabela do Apêndice C e na Tabela 4.3.

Ao analisar esses gráficos e tabelas, a impressão inicial é que o método NR é mais eficiente neste caso de maior ordem, superando o método LM tanto em precisão quanto em velocidade. Mas isso se deve ao fato que o número máximo de iterações foi mantido o mesmo dos testes no sistema de segunda ordem, e claramente o método LM precisaria de um aumento considerável nesse limite, o que elevaria o tempo de execução. Na Figura 4.7, é possível ver

Figura 4.7. Caminho traçado pelo algoritmo nos métodos NR e LM para o sistema de quarta ordem, tendo $\mathbf{K}^{(0)} \in \{0,1; 19,9\} \times \{0,1; 19,9\} \times \{0,1; 9,9\}$ como pontos iniciais.



Fonte: Autoria própria.

Tabela 4.3. Performance do algoritmo de sintonia do controlador no sistema de quarta ordem.

	$t_r = 1,5 s, t_p = 2 s, M_p = 5\%$		$t_r = 1 s, t_p = 2 s, t_s = 5 s$		$t_p = 2 s, M_p = 15\%, t_s = 5 s$	
	NR	LM	NR	LM	NR	LM
Precisão	43/64	30/64	50/64	29/64	19/64	35/64
Tempo médio de execução (ms)	261	2060	285	2250	310	1750
Número médio de iterações	4	100	5,70	112	7,11	85,5
Tempo médio de execução por iteração (ms/iter)	65,4	20,5	50,1	20,1	43,6	20,4

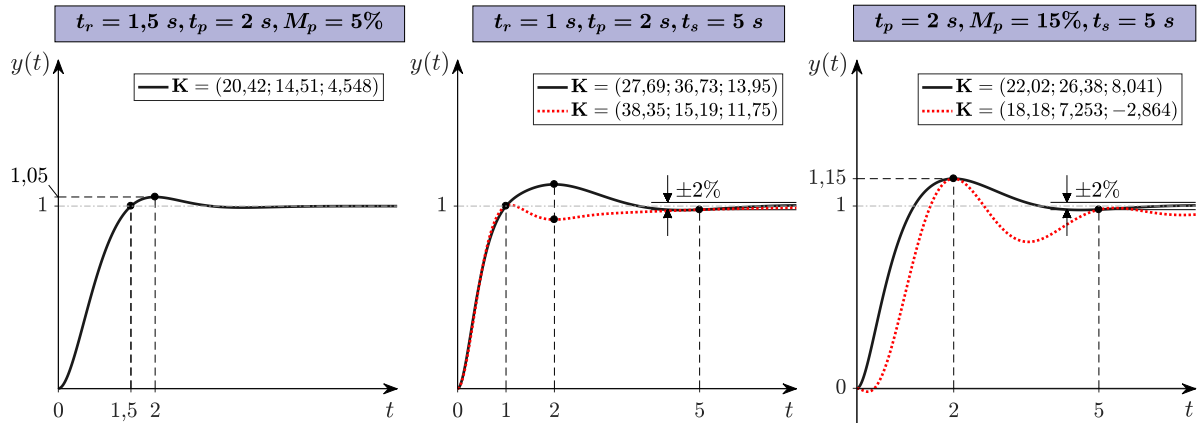
Fonte: Autoria própria.

que nos caminhos em que o método LM falha, os passos do algoritmo estavam direcionando os coeficientes para um dos pontos de convergência em um ritmo lento, até ser interrompido no meio do procedimento. Fora isso, todas as outras características observadas no caso de segunda ordem aparecem aqui, como a disparidade entre os comprimentos de passo e entre os tempos de execução dos métodos.

As curvas de resposta com os coeficientes sintonizados se encontram na Figura 4.8. Em todos os gráficos dessa figura, as curvas em preto atendem corretamente aos seus respectivos critérios, enquanto as curvas em vermelho falham em ao menos um dos critérios.

Para analisar a convergência de um único ponto inicial para diferentes critérios impostos, escolheu-se $\mathbf{K}^{(0)} = (67,3517; 84,1896; 13,4703)$, cuja resposta no domínio do tempo é caracte-

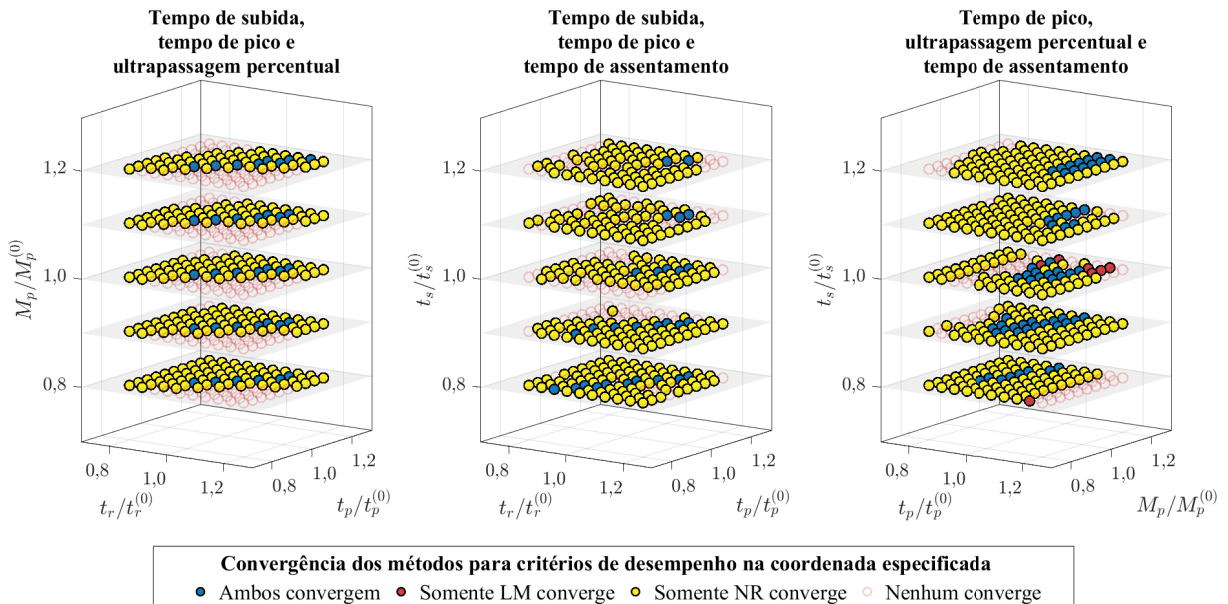
Figura 4.8. Curvas de resposta ao degrau alcançadas conforme a sintonia do algoritmo para o sistema de quarta ordem.



Fonte: Autoria própria.

rizada pelos parâmetros $t_r^{(0)} = 0,534$ s, $t_p^{(0)} = 0,938$ s, $M_p^{(0)} = 44,5\%$ e $t_s^{(0)} = 4,07$. O resultado é mostrado na Figura 4.9.

Figura 4.9. Convergência dos métodos no sistema de quarta ordem para 500 valores distintos dos critérios de desempenho em cada combinação, com $\mathbf{K}^{(0)} = (67,3517; 84,1896; 13,4703)$.



Fonte: Autoria própria.

Nesses gráficos, fica clara a superioridade do método NR dentro das particularidades desse caso em especial, com uma alta taxa de convergência. Porém, levando em conta as comparações realizadas até aqui, não é incorreto supor que quanto maior o valor atribuído a N_{max} , mais dos pontos amarelos da Figura 4.9 se tornariam azuis. Como uma única execução do programa já está na casa dos segundos quando o método LM é escolhido, como indica a Tabela 4.3, aumentar o limite de iterações inviabiliza a realização de centenas de testes consecutivos, mais é recomendado se o objetivo é sintonizar o controlador apenas para um caso particular.

5 CONCLUSÃO

Conforme as proposições iniciais, este trabalho realizou definições analíticas na curva de resposta ao degrau de sistemas LTI operando sob o controle PI/PID. Essas definições se deram a partir de critérios de desempenho no domínio do tempo, e o jacobiano das funções de desempenho com relação aos coeficientes do controlador integrou o passo iterativo dos métodos de Newton-Raphson e de Levenberg-Marquardt. Um algoritmo foi apresentado a fim de sintonizar o controlador automaticamente para um dado processo, com base nos critérios de desempenho estabelecidos e nas especificações do procedimento iterativo.

Os testes com o sistema de primeira ordem mostraram que ambos os métodos numéricos encontram uma solução com relativa rapidez devido à simplicidade dos cálculos, desde que a tolerância imposta não impacte negativamente na existência de uma solução em regiões próximas dos pontos iniciais. Nos sistemas de maior ordem, os três graus de liberdade do controlador foram aproveitados para instituir três critérios simultâneos sobre a resposta, mas isso elevou a exigência computacional e criou inconvenientes quando múltiplos testes são realizados de maneira sequencial, seja para vários pontos iniciais ou para várias combinações de critérios. O aumento da complexidade do sistema também pareceu condicionar a convergência do método LM para um valor mais alto no número máximo de iterações.

Em geral, certos cuidados devem ser tomados ao utilizar o algoritmo. O método NR tende a convergir rapidamente, mas estar próximo da solução é uma condição geralmente necessária para evitar uma divergência, o que mostra a importância de uma sintonia prévia para se aproximar da região de convergência. O método LM é mais robusto, porém necessita de mais iterações para atingir o resultado. Uma tolerância muito baixa prejudica a performance de ambos, pois reduz as chances de existir uma solução para o sistema. A verificação dos parâmetros de desempenho na curva de resposta de soluções individuais é essencial para confirmar o correto atendimento dos critérios, pois a existência de múltiplas soluções pode ser uma ocorrência comum em certos casos.

A primeira proposta para trabalhos futuros é encontrar uma forma de direcionar os passos iterativos do algoritmo para soluções que atendam corretamente aos critérios, e assim evitar falsos positivos. Isso provavelmente pode ser alcançado ao estabelecer restrições extras nas funções de desempenho, mas isto exigiria a utilização de métodos de otimização mais avançados que impõem essas restrições em seus cálculos.

Segundamente, há a possibilidade explorar a aplicação de métodos numéricos voltados

para resolução de sistemas sobredeterminados, como os métodos de região de confiança em [36]. Assim, seria possível impor quatro critérios de desempenho simultaneamente mesmo com os três graus de liberdade do controlador PID. Também seria interessante investigar o comportamento do algoritmo quando as funções de desempenho são redefinidas a partir de critérios diferentes dos aqui apresentados, como por exemplo, os critérios de erro integral em [10, 11].

Em terceiro lugar, devem-se buscar adaptações no algoritmo para problemas mal escalados, isto é, quando há uma grande desproporcionalidade entre as colunas ou linhas da matriz jacobiana. Essa adversidade ocorre frequentemente em funções de transferência modeladas a partir de casos práticos, e algumas sugestões para o método de Levenberg-Marquardt se encontram em [35].

Por último, o uso de modelos de redução de ordem pode reduzir significativamente a quantidade de cálculos realizados pelo programa em sistemas de ordem elevada. A aproximação por polos dominantes é um dos candidatos para esse empreendimento [1, 40].

REFERÊNCIAS

- [1] FRANKLIN, Gene F.; POWELL, J. David; EMAMI-NAEINI, Abbas. *Sistemas de Controle para Engenharia*. 6. ed. Porto Alegre: Bookman, 2013.
- [2] ÅSTRÖM, K.; HÄGGLUND, T. *PID Controllers: Theory, Design, and Tuning*. 2. ed. Research Triangle Park, NC: Instrument Society of America, 1995.
- [3] JELALI, Mohieddine. *Control Performance Management in Industrial Automation: Assessment, Diagnosis and Improvement of Control Loop Performance*. 3. ed. London: Springer, 2013.
- [4] O'DWYER, Aidan. *Handbook of PI and PID Controller Tuning Rules*. 3. ed. London: Imperial College Press, 2009.
- [5] LEVINE, William S. *The Control Handbook: Control System Fundamentals*. 2. ed. College Park, MD: CRC Press, 2011.
- [6] ZIEGLER, J. G.; NICHOLS, N. B. *Optimum Settings for Automatic Controllers*. Transactions of the ASME, Vol. 64, p. 759–768, 1942.
- [7] ÅSTRÖM, K.; HÄGGLUND, T. *The future of PID control*. Control Engineering Practice, Vol. 9, p. 1163-1175, 2001.
- [8] BORASE, Rakesh P.; MAGHADE, D. K.; SONDKAR, S. Y.; PAWAR, S. N. *A review of PID control, tuning methods and applications*. International Journal of Dynamics and Control, Vol. 9, p. 818-827, 2021.
- [9] MANN, G. K. I.; HU, B. -G.; GOSINE, R. G. *Time domain based design and analysis of new PID tuning rules*. IEE Proceedings - Control Theory and Applications, Vol. 148, N. 3, p. 251-261, 2001.
- [10] SAHRAIAN, Masood; KODIYALAM, Srinivas. *Tuning PID Controllers Using Error-Integral Criteria and Numerical Optimization*. American Institute of Aeronautics and Astronautics Meeting Papers on Disc, p. 237-246, 1996.
- [11] ŽÁN, Vilém; KUBÍČEK, Karel; ČECH, Martin. *Design of robust PI controller by combining robustness regions with time-domain criteria*. IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA), p. 1-8, 2022.

- [12] RAD, A. Besharati; LO, Wai Lun; TSANG, K. M. *Self-Tuning PID Controller Using Newton–Raphson Search Method*. IEEE Transactions on Industrial Electronics, Vol. 44, N. 5, p. 717-725, 1997.
- [13] ÅSTRÖM, K.; PANAGOPOULOS, H.; HÄGGLUND, T. *Design of PI Controllers based on Non-Convex Optimization*. Automatica, Vol. 34, N. 5, p. 585-601, 1998.
- [14] EL-HAMID, Ahmed S. Abd; EISSA, Ahmed H.; RADWAN, Aly M. *Levenberg-Marquardt's Algorithm used for PID Controller Parameters Optimization*. International Journal of Engineering Research, Vol. 4, N. 6, p. 286-290, 2015.
- [15] BAZANELLA, Alexandre Sanfelice; CAMPESTRINI, Lucíola; ECKHARD, Diego. *Data-Driven Controller Design: The H_2 Approach*. [S.l.]: Springer, 2012.
- [16] HO, Ming-Tzu. *Synthesis of H_∞ PID controllers: A parametric approach*. Automatica, Vol. 39, p. 1069-1075, 2003.
- [17] KIM, Tae-Hyoung; MARUTA, Ichiro; SUGIE, Toshiharu. *Robust PID controller tuning based on the constrained particle swarm optimization*. Automatica, Vol. 44, p. 1104-1110, 2008.
- [18] LEE, Chia-Ling; PENG, Chao-Chung. *Analytic Time Domain Specifications PID Controller Design for a Class of 2nd Order Linear Systems: A Genetic Algorithm Method*. IEEE Access, Vol. 9, p. 99266-99275, 2021.
- [19] LEE, Chia-Ling; PENG, Chao-Chung. *Performance Demands based Servo Motor Speed Control: A Genetic Algorithm Proportional-Integral Control Parameters Design*. International Symposium on Computer, Consumer and Control (IS3C), p. 469-472, 2020.
- [20] HORNG, Huey-Yang. *Design of the Second-Order Controller by Time-Domain Objective Functions Using Cuckoo Search*. In: DEL SER, Javier; VILLAR, Esther; OSABA, Eneko. *Swarm Intelligence - Recent Advances, New Perspectives and Applications*. London: In-techOpen, 2019. p. 101-113.
- [21] HAYKIN, Simon; VAN VEEN, Barry. *Sinais e Sistemas*. Porto Alegre: Bookman, 2001.
- [22] DORF, Richard C.; BISHOP, Robert H. *Sistemas de Controle Modernos*. 11. ed. Rio de Janeiro: LTC, 2011.

- [23] ALEXANDER, Charles K.; SADIKU, Matthew N. O. *Fundamentos de Circuitos Eléctricos*. 5. ed. Porto Alegre: AMGH, 2013.
- [24] SIEBERT, William McC. *Circuits, Signals, and Systems*. Cambridge, MA: MIT Press; New York: McGraw-Hill, 1986.
- [25] PHILLIPS, Charles L.; PARR, John M. *Signals, Systems, and Transforms*. 4. ed. Pearson Prentice Hall, 2008.
- [26] NISE, Norman S. *Engenharia de Sistemas de Controle*. 6. ed. Rio de Janeiro: LTC, 2013.
- [27] OPPENHEIM, Alan V.; WILLSKY, Alan S. *Sinais e Sistemas*. 2. ed. São Paulo: Pearson Prentice Hall, 2010.
- [28] ANG, Kiam Heong; CHONG, Gregory; LI, Yun. *PID Control System Analysis, Design, and Technology*. IEEE Transactions on Control Systems Technology, Vol. 13, N. 4, p. 559-576, 2005.
- [29] MACDONALD, I. G. *Symmetric Functions and Hall Polynomials*. 2 ed. London: Oxford University Press Inc., 1995.
- [30] BAKER, Frank B.; HARWELL, Michael R. *Computing Elementary Symmetric Functions and Their Derivatives: A Didactic*. Applied Psychological Measurement, Vol. 20, N. 2, p. 169-192, 1996.
- [31] CHAPRA, Steven C.; CANALE, Raymond P. *Numerical Methods for Engineers*. 7. ed. New York: McGraw-Hill, 2015.
- [32] PRESS, William H.; TEUKOLSKY, Saul A.; VETTERLING, William T.; FLANNERY, Brian P. *Numerical Recipes*. 3. ed. New York: Cambridge University Press, 2007.
- [33] FRANCO, Neide Bertoldi. *Cálculo Numérico*. São Paulo: Pearson Prentice Hall, 2006.
- [34] BURDEN, Richard L.; FAIRES, J. Douglas. *Numerical Analysis*. 9. ed. Boston, MA: Cengage Learning, 2010.
- [35] BENATTI, Kléber Aderaldo. *O Método de Levenberg-Marquardt para o Problema de Quadrados Mínimos Não Linear*. Dissertação (Mestrado em Matemática) - Setor de Ciências Exatas, Universidade Federal do Paraná. Curitiba, p. 106. 2017.

- [36] YUAN, Ya-xiang. *Trust Region Algorithms for Nonlinear Equations*. Conference on Scientific Computation, Hong Kong, 1994.
- [37] MATLAB R2022a. Natick, MA: The MathWorks, Inc., 2022. Disponível em: <https://www.mathworks.com/products/matlab.html>. Acesso em: 10 nov 2022.
- [38] LASCOUX, Alain; PRAGACZ, Piotr. *Jacobians of Symmetric Polynomials*. Annals of Combinatorics, Vol. 6, p. 169-172, 2002.
- [39] UMAR, A. O.; SULAIMAN, I. M.; MAMAT, M.; WAZIRI, M. Y.; ZAMRI, N. *On damping parameters of Levenberg-Marquardt algorithm for nonlinear least square problems*. Journal of Physics: Conference Series, Vol. 17340, 012058, 2021.
- [40] SCHILDERS, Wilhelmus H. A.; VORST, Henk A. van der; ROMMES, Joost. *Model Order Reduction: Theory, Research Aspects and Applications*. [S.l.]: Springer, 2008.

APÊNDICE A - Tabela com os dados coletados durante a execução do programa para o sistema de primeira ordem.

N	K ⁽⁰⁾	$t_r = 2\text{ s}, t_s = 4\text{ s}$						$t_p = 3\text{ s}, M_p = 2\%$						$t_p = 3\text{ s}, t_s = 4\text{ s}$					
		NR			LM			NR			LM			NR			LM		
		T_{exec}	N_{iter}	K'	T_{exec}	N_{iter}	K'	T_{exec}	N_{iter}	K'	T_{exec}	N_{iter}	K'	T_{exec}	N_{iter}	K'	T_{exec}	N_{iter}	K'
1	(0,4;1,6)	16,4	2	(0,958;1,6)	6,61	3	(0,953;1,59)	6,7	3	(1,18;1,62)	3,11	4	(1,18;1,62)	6,54	3	(0,95;1,52)	2,19	3	(0,951;1,52)
2	(1,1;1,6)	14,6	1	(0,95;1,6)	0,704	1	(0,969;1,61)	0,715	1	(1,17;1,62)	0,405	1	(1,17;1,62)	0,71	2	(0,949;1,52)	0,59	2	(0,95;1,52)
3	(1,8;1,6)	19,2	3	(0,956;1,6)	4,13	4	(0,956;1,59)	5,76	3	(1,18;1,62)	1,22	3	(1,17;1,61)	4,91	3	(0,945;1,52)	5,67	4	(0,948;1,52)
4	(2,5;1,6)	-	-	-	3,32	8	(0,956;1,6)	-	-	-	1,53	7	(1,18;1,62)	-	-	-	2,89	9	(0,948;1,52)
5	(3,2;1,6)	-	-	-	4,81	20	(0,957;1,6)	-	-	-	1,81	13	(1,17;1,62)	1,21	7	(0,19;4,62)	6,84	23	(0,943;1,52)
6	(3,9;1,6)	-	-	-	6,99	56	(0,955;1,59)	-	-	-	2,76	25	(1,18;1,62)	-	-	-	8,15	64	(0,948;1,52)
7	(4,6;1,6)	-	-	-	21,5	162	(0,956;1,59)	-	-	-	3,84	43	(1,18;1,62)	-	-	-	11,8	155	(0,948;1,52)
8	(5,3;1,6)	-	-	-	40,4	450	(0,94;1,59)	-	-	-	5,85	67	(1,16;1,61)	-	-	-	22,1	309	(0,951;1,52)
9	(0,4;4,4)	8,84	5	(0,955;6,53)	3,4	14	(0,956;6,53)	-	-	-	5,75	40	(1,15;1,6)	0,831	3	(0,19;4,62)	0,651	3	(0,19;4,62)
10	(1,1;4,4)	1,79	5	(0,956;6,53)	67,1	410	(0,956;1,6)	-	-	-	4,22	38	(1,18;1,62)	-	-	-	1,37	10	(0,19;4,62)
11	(1,8;4,4)	-	-	-	29,7	172	(0,957;1,6)	-	-	-	3,66	32	(1,17;1,62)	2,24	8	(0,191;4,62)	-	-	-
12	(2,5;4,4)	-	-	-	20,6	139	(0,955;1,59)	-	-	-	3,36	30	(1,18;1,62)	-	-	-	15,6	151	(0,952;1,52)
13	(3,2;4,4)	-	-	-	17	165	(0,955;1,59)	-	-	-	3,45	34	(1,17;1,61)	-	-	-	12	163	(0,949;1,52)
14	(3,9;4,4)	-	-	-	25,2	261	(0,956;1,6)	-	-	-	3,93	46	(1,18;1,62)	-	-	-	21,2	233	(0,947;1,52)
15	(4,6;4,4)	-	-	-	54,3	496	(0,955;1,59)	-	-	-	5,3	64	(1,18;1,62)	-	-	-	37,6	354	(0,949;1,52)
16	(5,3;4,4)	-	-	-	102	998	(0,957;1,6)	-	-	-	9,35	89	(1,18;1,62)	-	-	-	44,6	525	(0,952;1,52)
17	(0,4;7,2)	2,2	3	(0,956;6,53)	1,36	3	(0,965;6,54)	2,17	5	(1,55;10,5)	33,6	161	(1,18;1,62)	-	-	-	-	-	-
18	(1,1;7,2)	1,27	2	(0,957;6,53)	0,984	3	(0,955;6,53)	1,72	6	(1,55;10,5)	20,8	156	(1,18;1,62)	-	-	-	-	-	-
19	(1,8;7,2)	2,15	5	(0,959;6,53)	1,7	4	(0,955;6,53)	3,35	23	(1,6;89,5)	20,8	150	(1,16;1,6)	-	-	-	-	-	-
20	(2,5;7,2)	-	-	-	1,44	8	(0,955;6,53)	-	-	-	21,1	146	(1,17;1,62)	-	-	-	-	-	-
21	(3,2;7,2)	-	-	-	3,52	21	(0,956;6,53)	-	-	-	16,8	148	(1,17;1,62)	-	-	-	-	-	-
22	(3,9;7,2)	-	-	-	1600	8352	(0,955;1,59)	-	-	-	23,3	159	(1,18;1,62)	-	-	-	188	1673	(0,948;1,52)
23	(4,6;7,2)	-	-	-	559	5622	(0,955;1,59)	-	-	-	14,1	178	(1,16;1,61)	-	-	-	117	1271	(0,932;1,51)
24	(5,3;7,2)	-	-	-	591	6109	(0,956;1,59)	-	-	-	18,3	207	(1,16;1,61)	-	-	-	121	1412	(0,951;1,52)
25	(0,4;10)	2,92	8	(0,94;6,51)	13,7	123	(0,955;6,53)	0,564	4	(1,55;10,5)	0,658	4	(1,54;10,5)	-	-	-	-	-	-
26	(1,1;10)	1,72	7	(0,949;1,59)	13,6	131	(0,95;6,53)	0,54	2	(1,54;10,4)	0,542	3	(1,55;10,5)	-	-	-	-	-	-
27	(1,8;10)	-	-	-	19	145	(0,956;6,53)	0,391	2	(1,56;10,5)	0,44	2	(1,55;10,5)	-	-	-	-	-	-
28	(2,5;10)	-	-	-	18,6	169	(0,95;6,53)	0,824	5	(1,55;10,5)	0,643	4	(1,56;10,5)	-	-	-	1390	9591	(0,949;1,52)
29	(3,2;10)	-	-	-	28	213	(0,956;6,53)	0,729	4	(1,59;28)	1,75	15	(1,56;10,5)	-	-	-	1350	9711	(0,95;1,52)
30	(3,9;10)	-	-	-	34,6	295	(0,939;6,52)	-	-	-	60,2	492	(1,18;1,62)	-	-	-	1380	9720	(0,949;1,52)
31	(4,6;10)	-	-	-	48,2	456	(0,955;6,53)	-	-	-	58,2	485	(1,18;1,62)	-	-	-	1350	9417	(0,948;1,52)
32	(5,3;10)	-	-	-	108	853	(0,955;6,53)	-	-	-	60,1	510	(1,18;1,62)	-	-	-	429	3534	(0,951;1,52)
33	(0,4;12,8)	1,21	5	(0,955;16,4)	967	6850	(0,939;6,52)	0,782	5	(1,55;10,5)	1,56	13	(1,55;10,5)	-	-	-	-	-	-
34	(1,1;12,8)	1,13	5	(0,956;16,4)	709	5375	(0,955;6,53)	0,599	4	(1,55;10,5)	1,45	12	(1,54;10,5)	0,869	5	(0,791;17,8)	-	-	-
35	(1,8;12,8)	1,01	5	(0,956;31,2)	564	4469	(0,956;6,53)	0,615	3	(1,55;10,5)	1,41	12	(1,55;10,5)	-	-	-	-	-	-
36	(2,5;12,8)	-	-	-	512	3996	(0,95;6,53)	-	-	-	1,82	13	(1,55;10,5)	-	-	-	-	-	-
37	(3,2;12,8)	-	-	-	512	3870	(0,934;6,51)	-	-	-	2,48	17	(1,55;10,5)	-	-	-	-	-	-
38	(3,9;12,8)	-	-	-	508	4073	(0,955;6,53)	-	-	-	3,29	26	(1,54;10,5)	-	-	-	-	-	-
39	(4,6;12,8)	-	-	-	713	4642	(0,955;6,53)	-	-	-	7,29	56	(1,57;10,5)	-	-	-	-	-	-
40	(5,3;12,8)	-	-	-	758	5695	(0,945;6,52)	-	-	-	120	1152	(1,18;1,62)	-	-	-	-	-	-
41	(0,4;15,6)	0,907	3	(0,955;16,4)	0,67	3	(0,95;16,4)	-	-	-	14,4	147	(1,55;10,5)	0,573	3	(0,787;17,8)	-	-	-
42	(1,1;15,6)	0,64	2	(0,956;16,4)	0,472	2	(0,958;16,4)	1,59	11	(1,17;1,62)	14,3	132	(1,54;10,5)	0,804	5	(0,79;17,8)	-	-	-
43	(1,8;15,6)	0,937	5	(0,956;16,4)	0,733	5	(0,955;16,4)	-	-	-	14,9	122	(1,55;10,5)	-	-	-	-	-	-
44	(2,5;15,6)	3,46	19	(0,954;6,53)	3,81	23	(0,95;16,4)	-	-	-	14,9	118	(1,55;10,5)	0,686	4	(0,718;54)	-	-	-
45	(3,2;15,6)	-	-	-	-	-	-	-	-	-	14,1	122	(1,55;10,5)	-	-	-	-	-	-
46	(3,9;15,6)	-	-	-	-	-	-	-	-	-	16,4	134	(1,55;10,5)	-	-	-	-	-	-
47	(4,6;15,6)	-	-	-	-	-	-	-	-	-	19,9	161	(1,55;10,5)	-	-	-	-	-	-
48	(5,3;15,6)	-	-	-	-	-	-	-	-	-	43,3	219	(1,54;10,5)	-	-	-	-	-	-
49	(0,4;18,4)	0,777	3	(0,95;16,4)	1,19	8	(0,939;16,4)	-	-	-	62,1	580	(1,55;10,5)	0,473	3	(0,79;17,8)	0,587	3	(0,793;17,8)
50	(1,1;18,4)	0,817	3	(0,955;16,4)	1,32	11	(0,955;16,4)	-	-	-	61,8	598	(1,55;10,5)	0,479	3	(0,79;17,8)	0,634	4	(0,79;17,8)
51	(1,8;18,4)	-	-	-	1,7	15	(0,955;16,4)	-	-	-	68,9	590	(1,57;10,5)	0,818	7	(0,788;17,8)	0,623	4	(0,783;17,8)
52	(2,5;18,4)	-	-	-	2,5	23	(0,948;16,4)	-	-	-	62,3	563	(1,55;10,5)	-	-	-	1,11	9	(0,765;17,8)
53	(3,2;18,4)	-	-	-	4,42	41	(0,94;16,4)	-	-	-	66,4	551	(1,54;10,5)	-	-	-	4,48	35	(0,789;17,8)
54	(3,9;18,4)	-	-	-	9,88	92	(0,955;16,4)	-	-	-	59,4	567	(1,55;10,5)	-	-	-	-	-	-
55	(4,6;18,4)	-	-	-	-	-	-	-	-	-	75,4	612	(1,57;10,5)	-	-	-	-	-	-
56	(5,3;18,4)	-	-	-	-	-	-	-	-	-	76,5	694	(1,55;10,5)	-	-	-	-	-	-
57	(0,4;21,2)	1,05	6	(0,949;75,6)	60,2	531	(0,975;16,4)	0,823	7	(1,6;28)	231	1901	(1,54;10,5)	0,937	6	(0,79;17,8)	18,8	150	(0,765;17,8)
58	(1,1;21,2)	-	-	-	60,2	576	(0,97;16,4)	0,994	9	(1,59;28)	247	1965	(1,55;10,5)	0,744	6	(0,79;17,8)	17,3	166	(0,79;17,8)
59	(1,8;21,2)	1,24	6	(0,955;6,53)	63,4	632	(0,966;16,4)	0,655	5	(1,59;54,4)	242	2023	(1,54;10,5)	0,727	6	(0,95;1,52)	24,4	185	(0,79;17,8)
60	(2,5;21,2)	-	-	-	107	709	(0,974;16,4)	-	-	-	241	2072	(1,54;10,5)	-	-	-	25,4	213	(0,789;17,8)
61	(3,2;21,2)	-	-	-	101	823	(0,958;16,4)	-	-	-	240	2117	(1,56;10,5)	-	-	-	30,2	259	(0,77;17,8)
62	(3,9;21,2)	-	-	-	145	1004	(0,975;16,4)	-	-	-	263	2174	(1,55;10,5)	-	-	-	43,5	350	(0,783;17,8)
63	(4,6;21,2)	-	-	-	144	1321	(0,953;16,4)	-	-	-	269	2265	(1,55;10,5)	-	-	-	67,7	594	(0,789;17,8)
64	(5,3;21,2)	-	-	-	273	2088	(0,944;16,4)	-	-	-	300	2424	(1,55;10,5)	-	-	-	78,6	5976	(0,781;17,8)

Obs.: K' é o ponto retornado pelo algoritmo após a execução e T_{exec} está em ms.

APÊNDICE B - Tabela com os dados coletados durante a execução do programa para o sistema de segunda ordem.

N	K ⁽⁰⁾	$t_r = 1,5 s, t_p = 2 s, M_p = 5\%$						$t_r = 1 s, t_p = 2 s, t_x = 5 s$						$t_p = 2 s, M_p = 15\%, t_x = 5 s$					
		NR			LM			NR			LM			NR			LM		
		T_{exec}	N_{iter}	K'	T_{exec}	N_{iter}	K'	T_{exec}	N_{iter}	K'	T_{exec}	N_{iter}	K'	T_{exec}	N_{iter}	K'	T_{exec}	N_{iter}	K'
1	(0,1;0,1;0,1)	1550	7	(2,94;3,23;0,75)	398	12	(2,94;3,22;0,748)	347	9	(94,5;0,616;5,77)	747	32	(8,88;3,97;2,3)	-	-	-	376	11	(2,1;0,918;-0,999)
2	(2,5;0,1;0,1)	363	4	(2,94;3,23;0,75)	328	9	(2,93;3,21;0,742)	-	-	-	569	22	(8,87;3,96;2,29)	-	-	-	283	7	(2,1;0,914;-1)
3	(4,9;0,1;0,1)	1190	3	(2,94;3,23;0,751)	445	16	(2,94;3,23;0,75)	263	5	(8,89;3,96;2,29)	436	15	(8,87;3,95;2,29)	-	-	-	530	21	(2,96;5,73;1,46)
4	(7,3;0,1;0,1)	341	7	(2,94;3,22;0,748)	1110	53	(2,94;3,23;0,75)	228	4	(8,85;3,93;2,27)	368	11	(8,89;3,96;2,29)	-	-	-	1130	55	(2,98;5,8;1,49)
5	(0,1;2,5;0,1)	284	5	(2,94;3,22;0,748)	336	9	(2,94;3,22;0,749)	339	9	(3,82;35,3;13,9)	727	32	(8,92;4,02;2,32)	247	5	(2,1;0,91;-1)	416	14	(2,96;5,71;1,45)
6	(2,5;2,5;0,1)	216	3	(2,94;3,23;0,75)	231	4	(2,94;3,22;0,748)	318	8	(2,72;31,6;11,9)	609	25	(8,89;3,96;2,29)	-	-	-	347	10	(2,95;5,69;1,44)
7	(4,9;2,5;0,1)	249	4	(2,94;3,23;0,75)	321	9	(2,94;3,24;0,758)	229	4	(8,88;3,97;2,3)	409	13	(8,83;3,94;2,28)	-	-	-	405	13	(2,95;5,66;1,43)
8	(7,3;2,5;0,1)	-	-	-	877	40	(2,94;3,23;0,75)	237	4	(8,87;3,96;2,29)	323	8	(8,88;3,95;2,29)	-	-	-	904	38	(2,97;5,77;1,47)
9	(0,1;4,9;0,1)	245	5	(2,94;3,23;0,75)	368	12	(2,94;3,23;0,752)	314	7	(3,37;34,5;13,5)	-	-	-	-	-	-	320	9	(2,94;5,63;1,42)
10	(2,5;4,9;0,1)	267	4	(2,94;3,23;0,75)	297	8	(2,94;3,23;0,75)	299	7	(2,76;32,1;12,2)	-	-	-	-	-	-	273	6	(2,95;5,69;1,44)
11	(4,9;4,9;0,1)	242	4	(2,94;3,23;0,75)	378	13	(2,94;3,23;0,75)	282	5	(8,88;3,96;2,29)	500	19	(8,91;3,99;2,31)	-	-	-	313	8	(2,93;5,57;1,39)
12	(7,3;4,9;0,1)	-	-	-	1060	50	(2,94;3,23;0,75)	394	5	(8,85;3,91;2,27)	298	8	(8,89;3,97;2,3)	-	-	-	724	32	(2,97;5,78;1,48)
13	(0,1;7,3;0,1)	241	4	(2,94;3,23;0,75)	823	32	(2,94;3,23;0,75)	295	8	(3,25;34,4;13,3)	-	-	-	275	7	(1,99;25,1;8,83)	397	12	(2,97;5,76;1,47)
14	(2,5;7,3;0,1)	348	8	(2,94;3,22;0,749)	633	25	(2,94;3,23;0,752)	301	7	(2,77;31,9;12,1)	-	-	-	268	6	(1,99;25,1;8,83)	313	8	(2,97;5,78;1,48)
15	(4,9;7,3;0,1)	-	-	-	799	34	(2,94;3,23;0,754)	306	8	(8,88;3,93;2,28)	-	-	-	-	-	-	375	12	(2,99;5,88;1,52)
16	(7,3;7,3;0,1)	385	10	(2,94;3,23;0,749)	1990	100	(2,94;3,23;0,75)	269	5	(8,88;3,96;2,29)	496	18	(8,9;3,98;2,3)	-	-	-	826	37	(2,99;5,86;1,51)
17	(0,1;0,1;2,5)	-	-	-	564	22	(2,94;3,23;0,756)	-	-	-	994	46	(8,89;3,97;2,3)	-	-	-	592	24	(2,96;5,73;1,46)
18	(2,5;0,1;2,5)	-	-	-	477	17	(2,95;3,24;0,761)	338	8	(8,94;3,98;2,31)	609	24	(8,84;3,95;2,28)	-	-	-	511	20	(2,95;5,66;1,43)
19	(4,9;0,1;2,5)	-	-	-	649	27	(2,94;3,23;0,75)	-	-	-	421	15	(8,84;3,92;2,27)	-	-	-	642	27	(2,93;5,56;1,38)
20	(7,3;0,1;2,5)	-	-	-	1410	68	(2,94;3,23;0,75)	236	4	(8,89;3,95;2,29)	353	11	(8,89;3,96;2,29)	-	-	-	1280	64	(2,97;5,76;1,47)
21	(0,1;2,5;2,5)	293	6	(2,94;3,23;0,75)	448	16	(2,94;3,23;0,751)	-	-	-	875	41	(8,88;3,98;2,3)	-	-	-	447	16	(2,95;5,65;1,42)
22	(2,5;2,5;2,5)	385	11	(2,94;3,23;0,751)	294	8	(2,95;3,24;0,762)	461	17	(8,89;3,96;2,29)	556	22	(8,89;3,97;2,3)	-	-	-	374	12	(2,96;5,73;1,46)
23	(4,9;2,5;2,5)	-	-	-	404	14	(2,94;3,23;0,75)	284	6	(8,9;3,97;2,3)	357	12	(8,88;3,96;2,29)	-	-	-	422	15	(2,95;5,65;1,42)
24	(7,3;2,5;2,5)	-	-	-	925	43	(2,95;3,26;0,774)	215	3	(8,85;3,96;2,28)	265	6	(8,88;3,95;2,29)	-	-	-	875	41	(2,97;5,77;1,48)
25	(0,1;4,9;2,5)	279	6	(2,94;3,23;0,75)	563	21	(2,94;3,23;0,75)	-	-	-	-	-	-	-	-	-	384	12	(2,97;5,79;1,48)
26	(2,5;4,9;2,5)	280	5	(2,93;3,21;0,74)	357	11	(2,94;3,23;0,75)	314	8	(3,06;33,4;12,8)	-	-	-	-	-	-	247	5	(2,96;5,74;1,46)
27	(4,9;4,9;2,5)	-	-	-	448	14	(2,94;3,23;0,751)	365	12	(2,71;32,6;12,3)	535	20	(8,89;3,97;2,3)	-	-	-	602	8	(2,99;5,92;1,54)
28	(7,3;4,9;2,5)	377	7	(2,94;3,22;0,747)	990	45	(2,95;3,25;0,765)	211	3	(8,89;3,96;2,29)	261	5	(8,91;3,99;2,31)	-	-	-	391	29	(2,99;5,86;1,52)
29	(0,1;7,3;2,5)	314	5	(2,94;3,23;0,749)	1000	46	(2,94;3,23;0,75)	258	6	(3,36;35,7;13,9)	-	-	-	-	-	-	387	14	(2,98;5,82;1,5)
30	(2,5;7,3;2,5)	280	6	(2,94;3,23;0,75)	763	32	(2,94;3,23;0,75)	265	6	(3,23;34,6;13,4)	-	-	-	229	4	(2,96;5,75;1,47)	260	6	(3;5,93;1,54)
31	(4,9;7,3;2,5)	-	-	-	810	37	(2,95;3,25;0,77)	274	7	(2,55;31,7;11,9)	-	-	-	230	4	(2,97;5,77;1,48)	336	10	(2,98;5,83;1,5)
32	(7,3;7,3;2,5)	293	7	(2,94;3,23;0,751)	1660	82	(2,95;3,25;0,765)	-	-	-	561	23	(8,89;3,96;2,29)	-	-	-	701	30	(2,98;5,84;1,51)
33	(0,1;0,1;4,9)	-	-	-	1610	77	(2,95;3,26;0,772)	-	-	-	2120	108	(8,92;4,02;2,32)	-	-	-	1110	51	(2,97;5,78;1,48)
34	(2,5;0,1;4,9)	-	-	-	1290	64	(2,94;3,23;0,75)	-	-	-	992	46	(8,91;3,99;2,31)	252	5	(60,8;5,08;-0,204)	969	46	(2,97;5,76;1,47)
35	(4,9;0,1;4,9)	-	-	-	1700	82	(2,94;3,23;0,75)	-	-	-	642	27	(8,89;3,97;2,3)	-	-	-	1190	59	(2,99;5,88;1,52)
36	(7,3;0,1;4,9)	-	-	-	3410	157	(2,95;3,24;0,762)	-	-	-	490	19	(8,91;3,97;2,3)	-	-	-	2000	102	(2,97;5,77;1,47)
37	(0,1;2,5;4,9)	-	-	-	1170	55	(2,94;3,23;0,751)	-	-	-	2560	139	(8,94;4,04;2,33)	-	-	-	935	43	(2,99;5,89;1,53)
38	(2,5;2,5;4,9)	-	-	-	848	39	(2,94;3,23;0,75)	-	-	-	943	45	(8,9;3,98;2,3)	-	-	-	735	32	(2,99;5,89;1,53)
39	(4,9;2,5;4,9)	-	-	-	997	48	(2,94;3,23;0,75)	-	-	-	542	22	(8,9;3,99;2,31)	-	-	-	845	39	(2,99;5,88;1,52)
40	(7,3;2,5;4,9)	-	-	-	1860	97	(2,94;3,23;0,75)	-	-	-	380	13	(8,9;3,97;2,3)	-	-	-	1410	70	(2,97;5,8;1,49)
41	(0,1;4,9;4,9)	-	-	-	1290	63	(2,94;3,23;0,75)	-	-	-	-	-	-	-	-	-	632	27	(2,97;5,77;1,47)
42	(2,5;4,9;4,9)	-	-	-	1080	40	(2,94;3,23;0,75)	-	-	-	1130	54	(8,89;3,97;2,3)	-	-	-	505	20	(2,98;5,82;1,5)
43	(4,9;4,9;4,9)	-	-	-	1130	45	(2,94;3,23;0,756)	-	-	-	596	24	(8,9;3,97;2,3)	-	-	-	611	25	(2,97;5,8;1,49)
44	(7,3;4,9;4,9)	-	-	-	1730	90	(2,94;3,23;0,757)	-	-	-	385	13	(8,89;3,97;2,3)	-	-	-	1040	49	(2,99;5,86;1,51)
45	(0,1;7,3;4,9)	329	10	(2,94;3,22;0,749)	2060	109	(2,94;3,23;0,751)	-	-	-	-	-	-	-	-	-	758	34	(2,98;5,81;1,49)
46	(2,5;7,3;4,9)	-	-	-	1470	74	(2,94;3,23;0,75)	-	-	-	-	-	-	-	-	-	526	21	(2,97;5,77;1,48)
47	(4,9;7,3;4,9)	-	-	-	1520	78	(2,95;3,25;0,768)	284	7	(2,74;32,5;12,3)	-	-	-	-	-	-	609	24	(2,97;5,77;1,47)
48	(7,3;7,3;4,9)	-	-	-	2510	133	(2,94;3,23;0,751)	-	-	-	959	44	(8,89;3,97;2,3)	-	-	-	995	46	(2,97;5,76;1,47)
49	(0,1;0,1;7,3)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2220	117	(2,97;5,77;1,47)
50	(2,5;0,1;7,3)	-	-	-	-	-	-	-	-	-	2910	155	(8,89;3,97;2,3)	-	-	-	2100	108	(2,98;5,81;1,49)
51	(4,9;0,1;7,3)	-	-	-	-	-	-	355	11	(95,9;0,365;5,81)	1580	81	(8,89;3,96;2,29)	-	-	-	2490	128	(2,97;5,75;1,47)
52	(7,3;0,1;7,3)	-	-	-	-	-	-	-	-	-	1210	60	(8,92;4,01;2,32)	-	-	-	3490	189	(2,97;5,76;1,47)
53	(0,1;2,5;7,3)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2090	110	(2,99;5,89;1,53)
54	(2,5;2,5;7,3)	-	-	-	3110	167	(2,94;3,23;0,75)	-	-	-	3530	190	(8,92;4,01;2,32)	-	-	-	1820	89	(3;5,95;1,55)
55	(4,9;2,5;7,3)	-	-	-	3310	183	(2,94;3,23;0,75)	-	-	-	1640	79	(8,93;4,02;2,32)	-	-	-	1910	98	(2,97;5,79;1,48)
56	(7,3;2,5;7,3)	-	-	-	-	-	-	492	18	(21,6;5,49;0,407)	1090	53	(8,89;3,96;2,29)	-	-	-	2650	140	(2,98;5,85;1,51)
57	(0,1;4,9;7,3)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1510	75	(2,98;5,83;1,5)
58	(2,5;4,9;7,3)	-	-	-	2920	160	(2,94;3,23;0,75)	-	-	-	2240	115	(8,89;3,96;2,29)	-	-	-	1270	63	(2,98;5,84;1,51)
59	(4,9;4,9;7,3)	-	-	-	3080	165	(2,94;3,23;0,756)	-	-	-	1240	62	(8,89;3,96;2,29)	-	-	-	1400	71	(2,97;5,78;1,48)
60	(7,3;4,9;7,3)	-	-	-	-	-	-	-	-	-	978	45	(8,91;4,2,31)	-	-	-	2010	106	(2,97;5,78;1,48)
61	(0,1;7,3;7,3)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1490	76	(2,99;5,86;1,52)
62	(2,5;7,3;7,3)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1220	60	(2,99;5,86;1,51)
63	(4,9;7,3;7,3)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1300	65	(2,99;5,91;1,53)
64	(7,3;7,3;7,3)	-	-	-	-	-	-	-	-	-	1780	92	(8,9;3,99;2,31)	-	-	-	1960	100	(2,97;5,78;1,48)

Obs.: K' é o ponto retornado pelo algoritmo após a execução e T_{exec} está em ms.

APÊNDICE C - Tabela com os dados coletados durante a execução do programa para o sistema de quarta ordem.

N	K ⁽⁰⁾	$t_r = 1.5 s, t_p = 2 s, M_p = 5\%$						$t_r = 1 s, t_p = 2 s, t_d = 5 s$						$t_p = 2 s, M_p = 15\%, t_d = 5 s$					
		NR			LM			NR			LM			NR			LM		
		T_{exec}	N_{iter}	K'	T_{exec}	N_{iter}	K'	T_{exec}	N_{iter}	K'	T_{exec}	N_{iter}	K'	T_{exec}	N_{iter}	K'	T_{exec}	N_{iter}	K'
1	(0.1;0.1;0.1)	491	8	(20.4;14.5;4.55)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
2	(10;0.1;0.1)	301	4	(20.4;14.5;4.55)	2360	117	(20.3;14.4;4.45)	321	7	(232;42.8;482)	-	-	-	271	5	(18.2;7.25;-2.86)	1350	66	(18.2;7.34;-2.81)
3	(19.9;0.1;0.1)	266	3	(20.4;14.5;4.55)	1990	97	(20.3;14.4;4.42)	254	4	(38.4;15;11.7)	-	-	-	-	-	-	882	39	(18.2;7.23;-2.88)
4	(29.8;0.1;0.1)	279	3	(20.4;14.5;4.55)	3580	184	(20.5;14.6;4.64)	262	4	(38.4;15.1;11.7)	3060	151	(38.4;15.1;11.7)	-	-	-	-	-	-
5	(0.1;10;0.1)	289	5	(20.4;14.5;4.55)	3730	188	(20.4;14.5;4.53)	304	7	(150;28.8;88.3)	-	-	-	-	-	-	-	-	-
6	(10;10;0.1)	231	3	(20.4;14.5;4.56)	1340	59	(20.4;14.5;4.53)	250	4	(27.7;36.7;13.9)	-	-	-	309	7	(18.2;7.3;-2.84)	1300	62	(18.2;7.27;-2.85)
7	(19.9;10;0.1)	249	3	(20.4;14.5;4.55)	608	24	(20.4;14.5;4.53)	291	7	(11.5;64.4;13.3)	-	-	-	239	3	(18.2;7.26;-2.86)	571	21	(18.2;7.26;-2.86)
8	(29.8;10;0.1)	235	3	(20.4;14.5;4.55)	1930	84	(20.4;14.5;4.57)	259	4	(38.4;15.2;11.7)	2080	107	(38.2;15.3;11.8)	267	5	(18.2;7.25;-2.86)	3460	178	(22.1;26.5;8.11)
9	(0.1;19.9;0.1)	-	-	-	-	-	-	308	7	(27.7;36.7;13.9)	-	-	-	-	-	-	-	-	-
10	(10;19.9;0.1)	303	4	(20.4;14.5;4.55)	1470	69	(20.5;14.6;4.64)	380	12	(11.5;64.4;13.3)	-	-	-	287	6	(22.2;27;8.32)	2080	104	(22;26.3;8.02)
11	(19.9;19.9;0.1)	251	3	(20.4;14.5;4.55)	708	29	(20.5;14.7;4.69)	242	4	(11.5;64.3;13.3)	-	-	-	265	5	(18.2;7.25;-2.86)	1300	62	(22;26.4;8.04)
12	(29.8;19.9;0.1)	260	4	(20.4;14.5;4.55)	1900	94	(20.5;14.7;4.71)	245	4	(38.4;15.1;11.7)	2900	152	(38.4;15.2;11.7)	-	-	-	1680	83	(22;26.4;8.05)
13	(0.1;29.8;0.1)	-	-	-	-	-	-	316	7	(62.8;131;72.4)	-	-	-	-	-	-	-	-	-
14	(10;29.8;0.1)	252	4	(20.4;14.5;4.55)	-	-	-	257	4	(11.5;64.4;13.3)	3000	147	(27.7;36.7;13.9)	273	5	(22.2;26.9;8.32)	1510	73	(22;26.4;8.04)
15	(19.9;29.8;0.1)	254	4	(20.4;14.5;4.55)	3250	168	(20.4;14.5;4.57)	242	4	(11.5;64.3;13.3)	2260	116	(27.7;36.7;13.9)	-	-	-	831	36	(22;26.2;7.96)
16	(29.8;29.8;0.1)	250	4	(20.4;14.4;4.5)	-	-	-	247	4	(38.4;15.1;11.7)	2020	99	(27.8;36.5;13.9)	-	-	-	1350	63	(22.4;27.4;8.52)
17	(0.1;0.1;5)	275	5	(20.4;14.5;4.55)	-	-	-	-	-	-	-	-	-	300	7	(356;15.4;56.5)	-	-	-
18	(10;0.1;5)	259	4	(20.4;14.5;4.55)	3100	133	(20.4;14.5;4.53)	-	-	-	-	-	-	-	-	-	2370	120	(18.2;7.28;-2.84)
19	(19.9;0.1;5)	236	3	(20.4;14.5;4.52)	2070	99	(20.4;14.5;4.51)	267	5	(38.4;15.2;11.7)	-	-	-	-	-	-	2120	105	(18.2;7.39;-2.77)
20	(29.8;0.1;5)	239	3	(20.4;14.5;4.55)	-	-	-	259	4	(38.4;15.1;11.7)	2430	119	(38.4;15.1;11.7)	-	-	-	-	-	-
21	(0.1;10;5)	-	-	-	-	-	-	507	19	(11.5;64.4;13.3)	-	-	-	-	-	-	-	-	-
22	(10;10;5)	237	3	(20.4;14.5;4.55)	1320	64	(20.4;14.5;4.5)	346	10	(11.5;64.4;13.3)	-	-	-	394	7	(18.2;7.26;-2.86)	3110	160	(18.2;7.26;-2.86)
23	(19.9;10;5)	204	2	(20.4;14.5;4.54)	475	16	(20.4;14.5;4.53)	242	4	(38.3;15.2;11.7)	3410	165	(38.4;15.2;11.7)	-	-	-	-	-	-
24	(29.8;10;5)	227	3	(20.4;14.5;4.55)	1850	92	(20.5;14.7;4.68)	221	3	(38.4;15.2;11.7)	1510	68	(38.2;15.3;11.7)	-	-	-	3490	169	(22;26.3;8)
25	(0.1;19.9;5)	298	7	(20.4;14.5;4.54)	-	-	-	330	8	(63.1;131;72.3)	-	-	-	-	-	-	-	-	-
26	(10;19.9;5)	227	3	(20.4;14.5;4.49)	1570	77	(20.5;14.6;4.61)	257	5	(27.7;36.7;13.9)	-	-	-	272	4	(22.2;26.9;8.27)	1880	95	(22;26.4;8.04)
27	(19.9;19.9;5)	239	2	(20.4;14.5;4.55)	591	23	(20.4;14.5;4.57)	296	3	(27.7;36.6;13.9)	3820	190	(27.7;36.6;13.9)	206	2	(22.2;26.9;8.29)	788	34	(22.1;26.5;8.1)
28	(29.8;19.9;5)	243	3	(20.4;14.5;4.55)	2060	104	(20.4;14.5;4.57)	244	3	(38.3;15.2;11.7)	2210	105	(38.4;15.2;11.7)	-	-	-	1470	69	(22;26.4;8.03)
29	(0.1;29.8;5)	246	4	(20.4;14.5;4.56)	-	-	-	295	6	(27.6;36.8;13.9)	-	-	-	-	-	-	-	-	-
30	(10;29.8;5)	241	3	(20.4;14.5;4.55)	-	-	-	271	5	(27.7;36.7;13.9)	2630	129	(27.7;36.5;13.9)	348	9	(18.2;7.25;-2.87)	1420	68	(22.2;27.8;3.32)
31	(19.9;29.8;5)	238	3	(20.4;14.5;4.53)	3570	184	(20.4;14.6;4.59)	347	9	(150;28.5;88.3)	1610	76	(27.7;36.6;13.9)	-	-	-	472	16	(22.4;27.4;8.55)
32	(29.8;29.8;5)	247	4	(20.4;14.5;4.55)	-	-	-	431	14	(11.5;64.4;13.3)	1440	66	(27.7;36.6;13.9)	-	-	-	1110	50	(22.5;27.8;8.72)
33	(0.1;0.1;9.9)	296	7	(20.4;14.5;4.55)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
34	(10;0.1;9.9)	-	-	-	3670	192	(20.3;14.4;4.45)	288	5	(38.3;15.1;11.7)	-	-	-	-	-	-	-	-	-
35	(19.9;0.1;9.9)	-	-	-	2770	145	(20.3;14.4;4.45)	265	5	(38.3;15.1;11.7)	-	-	-	-	-	-	-	-	-
36	(29.8;0.1;9.9)	-	-	-	-	-	-	263	4	(38.4;15.1;11.7)	2190	108	(38.4;15.1;11.7)	-	-	-	-	-	-
37	(0.1;10;9.9)	256	4	(20.4;14.5;4.55)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
38	(10;10;9.9)	267	5	(20.4;14.5;4.55)	2000	100	(20.4;14.5;4.56)	288	6	(150;29.4;88.4)	-	-	-	-	-	-	-	-	-
39	(19.9;10;9.9)	278	6	(20.4;14.5;4.55)	857	37	(20.5;14.6;4.65)	261	4	(38.3;15.2;11.7)	3000	156	(38.3;15.2;11.8)	-	-	-	2810	143	(22.1;26.5;8.12)
40	(29.8;10;9.9)	247	4	(20.4;14.5;4.52)	2420	122	(20.5;14.7;4.67)	247	3	(38.4;15.2;11.7)	1120	51	(38.4;15.2;11.7)	-	-	-	3660	191	(22;26.3;8.01)
41	(0.1;19.9;9.9)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
42	(10;19.9;9.9)	226	3	(20.4;14.5;4.55)	2290	115	(20.4;14.6;4.6)	267	5	(38.4;15.2;11.7)	-	-	-	-	-	-	2090	106	(22;26.2;7.98)
43	(19.9;19.9;9.9)	246	4	(20.4;14.5;4.56)	1310	42	(20.4;14.6;4.58)	242	3	(27.7;36.7;13.9)	3660	195	(27.8;36.4;13.9)	-	-	-	826	34	(22.1;26.5;8.1)
44	(29.8;19.9;9.9)	230	3	(20.4;14.5;4.54)	2640	129	(20.5;14.6;4.65)	243	4	(38.4;15.1;11.7)	1650	82	(38.4;15.2;11.7)	-	-	-	1500	72	(22;26.3;8.01)
45	(0.1;29.8;9.9)	-	-	-	-	-	-	389	11	(38.4;15.1;11.7)	-	-	-	-	-	-	-	-	-
46	(10;29.8;9.9)	239	3	(20.4;14.5;4.55)	-	-	-	240	3	(27.7;36.7;13.9)	2690	139	(27.7;36.5;13.9)	578	23	(18.2;7.26;-2.86)	1840	92	(22.5;27.7;8.69)
47	(19.9;29.8;9.9)	242	4	(20.4;14.5;4.54)	-	-	-	233	3	(27.7;36.7;13.9)	1120	53	(27.8;36.5;13.9)	215	2	(22.1;26.7;8.19)	411	13	(22.4;27.6;8.64)
48	(29.8;29.8;9.9)	251	4	(20.4;14.5;4.52)	-	-	-	224	2	(27.6;36.6;13.9)	957	43	(27.7;36.6;13.9)	-	-	-	1130	51	(22.4;27.4;8.55)
49	(0.1;0.1;14.8)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
50	(10;0.1;14.8)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
51	(19.9;0.1;14.8)	-	-	-	-	-	-	438	14	(11.5;64.4;13.3)	-	-	-	-	-	-	-	-	-
52	(29.8;0.1;14.8)	-	-	-	-	-	-	266	5	(38.3;15.1;11.7)	2470	123	(38.4;15.1;11.7)	282	6	(132;26.9;17.7)	-	-	-
53	(0.1;10;14.8)	317	6	(20.4;14.5;4.51)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
54	(10;10;14.8)	-	-	-	-	-	-	-	-	-	-	-	-	430	15	(132;26.6;17.6)	-	-	-
55	(19.9;10;14.8)	-	-	-	2250	113	(20.5;14.7;4.66)	-	-	-	3570	188	(38.3;15.3;11.8)	-	-	-	-	-	-
56	(29.8;10;14.8)	-	-	-	-	-	-	276	4	(38.4;15.2;11.7)	1320	60	(38.3;15.2;11.7)	-	-	-	-	-	-
57	(0.1;19.9;14.8)	285	5	(20.4;14.5;4.55)	-	-	-	352	9	(95.5;23.7;24.4)	-	-	-	-	-	-	-	-	-
58	(10;19.9;14.8)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	3000	156	(22;26.3;7.98)
59	(19.9;19.9;14.8)	-	-	-	2180	110	(20.5;14.6;4.61)	-	-	-	-	-	-	-	-	-	1440	71	(22.1;26.6;8.13)
60	(29.8;19.9;14.8)	-	-	-	-	-	-	256	4	(38.3;15.2;11.8)	1710	83	(38.3;15.3;11.8)	-	-	-	2230	113	(22.4;27.5;8.6)
61	(0.1;29.8;14.8)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
62	(10;29.8;14.8)	296	7	(20.4;14.5;4.56)	-	-	-	272	4	(38.5;59.4;27.8)	3420	183	(27.7;36.7;13.9)	363	11	(18.2;7.2;-2.9)	3120	157	(22.5;27.7;8.67)
63	(19.9;29.8;14.8)	-	-	-	-	-	-	225	3	(27.7;36.7;13.9)	1240	57	(27.7;36.6;13.9)	308	7	(18.2;7.21;-2.89)	970	43	(22.5;27.7;8.71)
64	(29.8;29.8;14.8)	-	-	-	-	-	-	242	3	(27.7;36.7;13.9)	835	36	(27.8;36.4;13.9)	279	6	(22.2;27;8.32)	1600	78	(22.4;27.6;8.65)

Obs.: K' é o ponto retornado pelo algoritmo após a execução e T_{exec} está em ms.