



Universidade Federal de Juiz de Fora  
Faculdade de Engenharia  
Graduação em Engenharia Elétrica

## Controle de Temperatura de Forno para Solda SMD

**Alunos:** Gustavo de Oliveira Santos  
Higor Resende Rotondo Rocha  
Michael Santos Nepomuceno  
Thiago Duque Saber de Lima

Juiz de Fora  
27 de agosto de 2021



Universidade Federal de Juiz de Fora  
Faculdade de Engenharia  
Graduação em Engenharia Elétrica

## Controle de Temperatura de Forno para Solda SMD

Relatório referente ao trabalho da disciplina de Laboratório de Controle, do curso de Engenharia Elétrica, Universidade Federal de Juiz de Fora.

**Alunos:** Gustavo dos Santos Oliveira  
Higor Resende Rotondo Rocha  
Michael Santos Nepomuceno  
Thiago Duque Saber de Lima

**Professor:** André Augusto Ferreira

Juiz de Fora  
27 de agosto de 2021

# Sumário

<b>1</b>	<b>Introdução</b>	<b>9</b>
<b>2</b>	<b>Fundamentação teórica</b>	<b>9</b>
2.1	Curva de Soldagem SMD . . . . .	10
<b>3</b>	<b>Modelagem do Sistema Dinâmico</b>	<b>11</b>
3.1	Linearização - Aproximação de Padé . . . . .	12
<b>4</b>	<b>Modelagem do Sistema no Espaço de Estados</b>	<b>12</b>
<b>5</b>	<b>Função de transferência da planta</b>	<b>13</b>
5.1	Ziegler e Nichols . . . . .	13
<b>6</b>	<b>Análise de Polos e Zeros</b>	<b>17</b>
6.1	Sistema sem atraso . . . . .	18
6.2	Sistema com atraso . . . . .	20
<b>7</b>	<b>Análise na Frequênci</b> a	<b>22</b>
7.1	Sistema sem atraso . . . . .	22
7.2	Sistema com atraso . . . . .	27
<b>8</b>	<b>Controladores</b>	<b>32</b>
8.1	Controlador Proporcional . . . . .	32
8.1.1	Análise de Estabilidade . . . . .	36
8.1.1.1	Critério de Routh . . . . .	36
8.1.1.2	Diagrama de Nyquist . . . . .	37
8.1.1.3	Lugar das Raízes . . . . .	38
8.2	Controlador Integral . . . . .	41
8.2.1	Análise de Estabilidade . . . . .	42
8.2.1.1	Critério de Routh . . . . .	42
8.2.1.2	Diagrama de Nyquist . . . . .	42
8.3	Controlador Proporcional-Integral . . . . .	47
8.3.1	Análise de Estabilidade . . . . .	47
8.3.1.1	Critério de Routh . . . . .	48
8.3.1.2	Diagrama de Nyquist . . . . .	49
8.4	Análise dos Controladores . . . . .	55
<b>9</b>	<b>Compensadores</b>	<b>56</b>
9.1	Compensador por Atraso de Fase . . . . .	57
9.1.1	Projeto através do Lugar das Raízes . . . . .	57
9.1.2	Projeto através da Análise na Frequênci	60
9.2	Compensador por Avanço de Fase . . . . .	64
9.2.1	Projeto através do Lugar das Raízes . . . . .	65
9.2.2	Projeto através da Análise na Frequênci	67

<b>10 Controlador em Espaço de Estados</b>	<b>70</b>
10.1 Projeto por Alocação de Polos . . . . .	71
10.2 Análise dos Resultados . . . . .	74
<b>11 PID com Predictor de Smith</b>	<b>76</b>
11.1 Predictor de Smith . . . . .	76
11.2 Controlador PID . . . . .	78
11.2.1 Ziegler-Nichols . . . . .	78
11.2.2 ITAE . . . . .	79
11.2.3 pidtuner . . . . .	80
11.2.4 Análise dos Controladores . . . . .	81
<b>12 Conclusão</b>	<b>83</b>
<b>Códigos</b>	<b>85</b>
<b>Referências</b>	<b>111</b>

# Listas de Figuras

1	Modelo de um forno. . . . .	9
2	Forno para solda SMD. . . . .	10
3	Curva de aquecimento ideal. . . . .	10
4	Método de Ziegler e Nichols para identificação de parâmetros. . . . .	14
5	Curva de reação em função de uma resposta ao degrau para um ponto de operação de 120 °C. . . . .	15
6	Resposta ao degrau em malha aberta de $G(s)$ comparada ao modelo. . . . .	16
7	Respostas do sistema em malha aberta. . . . .	17
8	Mapa de polos e zeros sem atraso. . . . .	18
9	Mapa de polos e zeros com atraso. . . . .	18
10	Resposta ao degrau sem atraso. . . . .	19
11	Resposta ao degrau com e sem atraso. . . . .	20
12	Comparação entre a resposta com atraso e Aproximação de Padé. . . . .	21
13	Resposta ao degrau com atraso. . . . .	21
14	Resposta ao degrau com Aproximação de Padé. . . . .	22
15	Diagrama de bode para o termo proporcional sem atraso. . . . .	23
16	Diagrama de bode para o polo sem atraso. . . . .	23
17	Diagrama de Módulo para o sistema sem atraso. . . . .	24
18	Diagrama de Fase para o sistema sem atraso. . . . .	24
19	Entrada e saída do sistema sem atraso para $\omega = 1 \times 10^{-4} \text{ rad s}^{-1}$ . . . . .	25
20	Entrada e saída do sistema sem atraso para $\omega = 1 \times 10^{-3} \text{ rad s}^{-1}$ . . . . .	26
21	Entrada e saída do sistema sem atraso para $\omega = 1 \times 10^{-1} \text{ rad s}^{-1}$ . . . . .	26
22	Diagrama de bode para o 2º polo (com atraso). . . . .	27
23	Diagrama de bode para o zero. . . . .	28
24	Diagrama de Módulo para o sistema com atraso. . . . .	29
25	Diagrama de Fase para o sistema com atraso. . . . .	29
26	Entrada e saída do sistema com atraso para $1 \times 10^{-5} \text{ rad s}^{-1}$ . . . . .	30
27	Entrada e saída do sistema com atraso para $1 \times 10^{-2} \text{ rad s}^{-1}$ . . . . .	31
28	Entrada e saída do sistema com atraso para $1 \times 10^1 \text{ rad s}^{-1}$ . . . . .	31
29	Diagrama de blocos com o controlador proporcional. . . . .	32
30	Rootlocus para o controlador proporcional. . . . .	33
31	Diagrama de Módulo com o controlador proporcional. . . . .	34
32	Diagrama de Fase com o controlador proporcional. . . . .	34
33	Diagrama de blocos com os controladores proporcional. . . . .	35
34	Respostas em malha fechada com o controlador proporcional. . . . .	35
35	Comparação entre as respostas com Padé e Atraso Puro utilizando controlador proporcional. . . . .	36
36	Diagrama de blocos para o Critério de Routh com controlador proporcional. . .	36
37	Diagrama de Nyquist para o sistema com controlador proporcional. . . . .	38
38	Root Locus para estabilidade do controlador proporcional. . . . .	39
39	Comparação entre as respostas com ganho marginalmente estável. . . . .	40
40	Root Locus para Padé de 10ª ordem. . . . .	41
41	Diagrama de blocos com o controlador integral. . . . .	41
42	Diagrama de Nyquist para o sistema com controlador integral. . . . .	43
43	Root locus para o controlador integrador. . . . .	44

44	Diagrama de bode inicial com controlador integral. . . . .	45
45	Diagrama de bode com controlador integral para $K_i = 0.00165453$ . . . . .	45
46	Comparação entre as respostas obtidas utilizando um controlador integrador. .	46
47	Erro entre as respostas com Aproximação de Padé e atraso no tempo utilizando o controlador integral. . . . .	46
48	Diagrama de blocos para o controlador proporcional integral. . . . .	47
49	Relação entre o $K_p$ e o $K_i$ máximo para o controlador proporcional-integral. .	49
50	Diagrama de Nyquist para o sistema com controlador proporcional-integral. .	50
51	Root locus para o controlador proporcional-integral. . . . .	51
52	Diagrama de bode inicial com controlador proporcional-integral. . . . .	52
53	Diagrama de bode com controlador proporcional-integral com $K_p = 3,398$ . . .	53
54	Comparação entre Padé e Atraso puro no tempo com o controlador proporcional-integral. . . . .	54
55	Comparação entre as respostas obtidas utilizando um controlador proporcional-integral. . . . .	54
56	Erro entre as respostas com Aproximação de Padé e atraso no tempo utilizando o controlador proporcional-integral. . . . .	55
57	Comparação entre as respostas do sistema com os controladores. . . . .	55
58	Comparação entre as respostas com Aproximação de Padé a atraso no tempo para todos os controladores. . . . .	56
59	Polos de malha fechada desejados pro sistema sem compensador por atraso de fase. . . . .	58
60	Root Locus e polos de malha fechada desejados pro sistema com compensador por atraso de fase. . . . .	59
61	Diagrama de bode pro sistema não compensado. . . . .	60
62	Diagrama de bode pro sistema com o ganho $K_c$ . . . . .	61
63	Diagrama de bode do sistema com compensador de atraso de fase. . . . .	62
64	Resposta ao degrau do sistema com compensador atraso de fase. . . . .	63
65	Resposta à rampa do sistema com compensador atraso de fase. . . . .	64
66	Comparação entre as respostas utilizando compensador e PI. . . . .	64
67	Mapa do sistema não compensado. . . . .	65
68	Root locus do sistema com compensador de avanço de fase. . . . .	66
69	Diagrama de bode pro sistema não compensado. . . . .	67
70	Diagrama de bode do sistema com compensador por avanço de fase. . . . .	69
71	Resposta ao degrau com os compensadores por avanço de fase. . . . .	70
72	Resposta a rampa com os compensadores por avanço de fase. . . . .	70
73	Diagrama de blocos para o Espaço de Estados . . . . .	71
74	Resposta ao degrau com Alocação de Polos e $k_r = 1$ . . . . .	74
75	Resposta ao degrau com Alocação de Polos e $k_r = 0,01$ . . . . .	74
76	Comparação entre os controlador PI e por Alocação de Polos. . . . .	75
77	Sistema de controle com tempo morto. . . . .	76
78	Sistema de controle com Predictor de Smith. . . . .	77
79	Diagrama de blocos equivalente ao sistema com o Predictor de Smith. . . . .	77
80	Diagrama de blocos para o controlador PID. . . . .	78
81	Diagrama de blocos para o controlador PID com planta sem atraso. . . . .	81
82	Resposta ao degrau do sistema com controladores PID e PI. . . . .	81
83	Diagrama de blocos do controlador PI sintonizado e com Predictor de Smith. .	82

84	Resposta ao degrau do sistema com controlador PI e Predictor de Smith. . . . .	82
85	Controle de temperatura do forno de acordo com a curva de aquecimento desejada.	83

# **Lista de Códigos**

1	Resposta da planta ao degrau. . . . .	85
2	Mapa de polos e zeros sem atraso. . . . .	85
3	Mapa de polos e zeros sem atraso. . . . .	86
4	Resposta em malha aberta sem atraso. . . . .	87
5	Resposta ao degrau com e sem atraso. . . . .	88
6	Padé e Atraso. . . . .	88
7	Resposta ao degrau com atraso. . . . .	89
8	Resposta ao degrau com Aproximação de Padé. . . . .	90
9	Diagrama de Bode para o termo proporcional sem atraso. . . . .	91
10	Diagrama de Bode para o polo sem atraso. . . . .	92
11	Diagrama de Bode para o sistema sem atraso. . . . .	93
12	Respostas sem atraso para uma entrada senoidal. . . . .	94
13	Diagrama de bode para o 2º polo (com atraso). . . . .	96
14	Diagrama de bode para o zero. . . . .	96
15	Diagrama de Bode para o sistema com atraso. . . . .	97
16	Resposta com atraso para uma entrada senoidal. . . . .	98
17	Root locus para o controlador proporcional. . . . .	100
18	Diagrama de Bode com o controlador proporcional. . . . .	101
19	respostas em malha fechada com o controlador proporcional. . . . .	102
20	Comparação entre as resposta com Padé e Atraso Puro utilizando controlador proporcional. . . . .	103
21	Diagrama de Nyquist para o sistema com atraso de transporte. . . . .	104
22	Root Locus para estabilidade do controlador proporcional. . . . .	104
23	Root Locus para Padé de 10ª ordem. . . . .	105
24	Diagrama de Nyquist para o sistema com controlador integral. . . . .	107
25	Root locus para o controlador integrador. . . . .	108
26	Análise da resposta para um controlador integral. . . . .	109

## **Lista de Tabelas**

1	Tabela para o Critério de Routh utilizando controlador proporcional.	37
2	Tabela para o Critério de Routh utilizando controlador integral.	42
3	Tabela para o Critério de Routh utilizando controlador proporcional-integral.	48
4	Comparação entre o Controlador PI e Alocação de Polos.	75
5	Tabela com parâmetros para o primeiro método de Ziegler-Nichols.	79
6	Tabela com parâmetros para o método ITAE.	80

## **Lista de Abreviações e Siglas**

**SMD** Surface Mounted Device

**PTH** Pin Througt Hole

**MTBF** Tempo Médio entre Falhas

**PCB** Printed Circuit Board

**PID** Proporcional Integral Derivativo

**EDO** Equação Diferencial Ordinária

## 1 Introdução

Na industria de eletrônicos é cada vez mais comum a utilização de componentes Surface Mounted Device (SMD) em detrimento de Pin Throught Hole (PTH), uma vez que esse tipo de componente oferece uma redução de espaço, tornando o produto final mais leve e compacto, garante uma economia de energia elétrica e possui um Tempo Médio entre Falhas (MTBF) superior [Prasad, 1997]. Em projetos eletrônicos, a soldagem desses componentes se torna necessária em algum ponto, porém, a soldagem convencional, feita com um ferro de solda de baixa potência ou ar quente e uma solda fina, é muito difícil devido à sua pequena dimensão e dificuldades de posicionamento. Devido a essas dificuldades, existe a alternativa do uso de um forno específico para essa aplicação.

Dessa forma, este trabalho tem como objetivo a revisão bibliográfica de projetos que visam à resolução de problemas inerentes à construção de processos térmicos de downscaling e seu controle de temperatura com o intuito de possibilitar a soldagem de componentes SMD para projetos eletrônicos a partir do controle de temperatura de um forno. Por conta disso, é importante que a câmara interna do forno escolhido possua dimensões tão pequenas quanto possível desde que caiba as placas a serem soldadas, pois, assim, haverá um aumento mais rápido de temperatura e, também, um barateamento do projeto [Cordeiro, 2016].

## 2 Fundamentação teórica

O forno para soldagem de componentes SMD pode ser compreendido como um sistema térmico, que são aqueles que envolvem a transferência de calor de uma substância para outra [Ogata, 2003]. Sistemas desse tipo possuem uma resposta livre típica de um sistema de segunda ordem amortecido, porém, é comum modelar como um sistema de primeira ordem com atraso [Astrom and HÄGGLUND, 1995].

A Figura 1 apresenta o modelo de um forno elétrico resistivo, o qual é constituído de uma câmara de aquecimento, resistências elétricas e uma carcaça metálica. Além disso, é necessária toda a parte eletrônica responsável pelo controle, não demonstrada nesse exemplo.

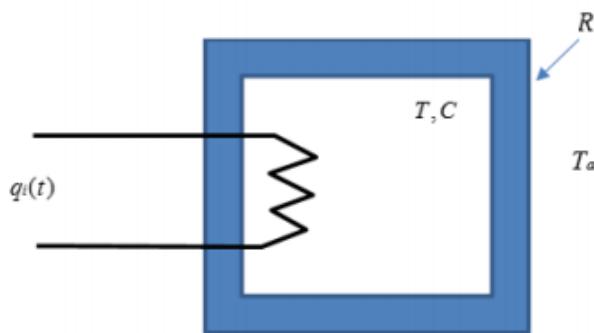


Figura 1: Modelo de um forno.

Fonte: [Sena, 2017]

A Figura 2 demonstra o forno elétrico resistivo no qual será baseado esse relatório. Nessa figura é possível observar os elementos citados anteriormente, como a Printed Circuit Board

(PCB) com os elementos responsáveis por toda a parte de controle, a carcaça metálica com a câmara de aquecimento e a resistência, não visível na imagem, em seu interior [Cordeiro, 2016].

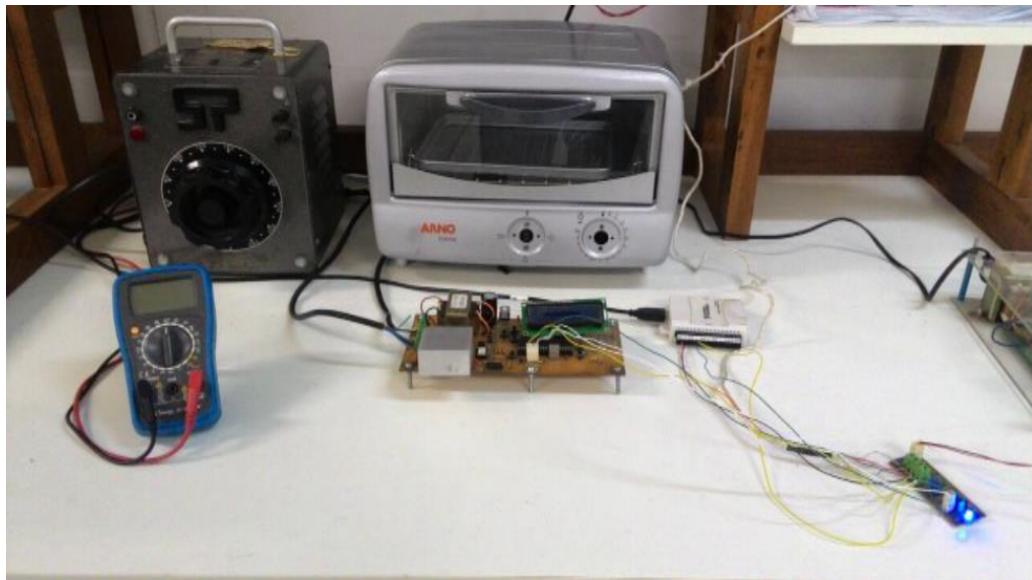


Figura 2: Forno para solda SMD.

Fonte: [Cordeiro, 2016]

## 2.1 Curva de Soldagem SMD

O procedimento recomendado para a utilização de um forno de solda SMD depende do controle da umidade do local, limpeza da placa e, principalmente, do processo de aquecimento da placa [Standard, 2014].

A Figura 3 demonstra a curva de aquecimento adequada para essa aplicação, a qual pode ser vista como a temperatura deve se comportar com o passar do tempo.

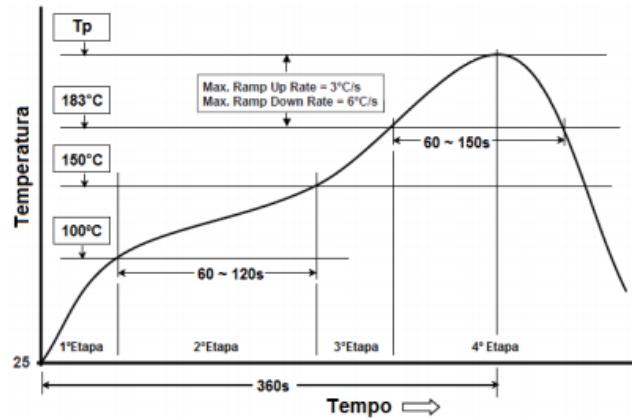


Figura 3: Curva de aquecimento ideal.

Fonte: [Meceneiro et al., 2016]

Podemos discretizar a curva de aquecimento em quatro etapas, conforme citadas abaixo.

- **1<sup>a</sup> Etapa:** Degrau para pré-aquecimento do forno da temperatura ambiente para 100 °C [Meceneiro et al., 2016].
- **2<sup>a</sup> Etapa:** Rampa de aquecimento de 100 °C a 150 °C, por um tempo de 60 s a 120 s [Meceneiro et al., 2016].
- **3<sup>a</sup> Etapa:** Rampa de aquecimento de 150 °C a 183 °C [Meceneiro et al., 2016].
- **4<sup>a</sup> Etapa:** Rampa de aquecimento de 150 °C a 220 °C subindo até  $3\text{ }^{\circ}\text{C s}^{-1}$ , descendo até  $6\text{ }^{\circ}\text{C s}^{-1}$ , por um tempo de 60 s a 150 s [Meceneiro et al., 2016].

### 3 Modelagem do Sistema Dinâmico

O sistema de controle proposto para realizar o projeto do controlador Proporcional Integral Derivativo (PID) é ilustrado na Figura 1. A planta consiste em um forno resistivo cuja temperatura interna varia de acordo com a quantidade de calor fornecido ao resistor de aquecimento.

A modelagem matemática do sistema descreve o comportamento da temperatura interna do forno, que é a variável a ser controlada, em função da quantidade de calor que é fornecida pelo resistor de aquecimento. É possível modelar o forno utilizando o conceito de Resistência Térmica ( $R$ ), Capacitância Térmica ( $C$ ) e através da temperatura externa do forno ( $T_a$ ).

O fluxo de calor, na transferência de calor por convecção ou condução, é dado pela Equação 1 [Ogata, 2003]:

$$q = k \cdot \Delta T \quad (1)$$

Assim, manipulando-a, iremos obter a Equação 4, que relaciona a temperatura interna do forno com a quantidade de calor fornecida pelo resistor de aquecimento.

$$q_o(t) = \frac{1}{R}(T(t) - T_a) \quad (2)$$

$$T(t) = \frac{1}{C} \left( q_i(t) - \frac{1}{R}(T(t) - T_a) \right) \quad (3)$$

$$RC \frac{dT(t)}{dt} + T(t) = R \cdot q_i(t) + T_a \quad (4)$$

Aplicando a Transformada de Laplace na Equação Diferencial Ordinária (EDO) encontrada na Equação Equação 4, é possível encontrar a seguinte equação, demonstrada na Equação 5 [Castaño, 2020].

$$sRCT(s) + T(s) = RQ_i(s) + \frac{T_a}{s} \quad (5)$$

Por ser um processo que acontece de forma rápida, conforme demonstrado na Figura 3, as flutuações de temperatura no ar de entrada serão desprezadas para facilitar no desenvolvimento do projeto. Através de manipulações, é possível encontrar a modelagem para o sistema térmico, demonstrado em Equação 7.

$$T(s)(sRC + 1) = RQ_i(s) \quad (6)$$

$$\boxed{\frac{T(s)}{Q_i(s)} = \frac{R}{sRC + 1}} \quad (7)$$

A Equação 7 pode ser reescrita para um modelo de primeira ordem, como demonstrada na Equação 8 [Ferreira, 2020].

$$G(s) = \frac{K}{\tau \cdot s + 1} \quad (8)$$

$$\begin{cases} K = R \\ \tau = RC \end{cases} \quad (9)$$

Porém, devido ao atraso de propagação do calor, é necessário fazer modificações na função de transferência encontrada na Equação 8, sendo assim necessário incluir um atraso, conforme a Equação 10.

$$\boxed{G(s) = \frac{K}{\tau \cdot s + 1} \cdot e^{-\theta s}} \quad (10)$$

### 3.1 Linearização - Aproximação de Padé

A Equação 10 apresenta a equação de um sistema não-linear. Esse tipo de sistema não é interessante de ser tratado, devido a sua alta complexidade. Para resolver isso, são aplicados métodos de linearização, como por exemplo a **Aproximação de Padé**. De maneira geral, podemos aproximar a exponencial para uma aproximação de primeira ordem da seguinte forma, conforme demonstrado em Equação 11 [Aguirre, 2020].

$$e^{-\theta s} = \frac{-s + \frac{2}{\theta}}{s + \frac{2}{\theta}} \quad (11)$$

Com isso, podemos reescrever a Equação 10 aplicando a aproximação demonstrada na Equação 11, encontrando a Equação 12

$$\boxed{G(s) = \frac{K \left( -s + \frac{2}{\theta} \right)}{(\tau s + 1) \cdot \left( s + \frac{2}{\theta} \right)}} \quad (12)$$

## 4 Modelagem do Sistema no Espaço de Estados

Para a modelagem no espaço de estados foi utilizada a função de transferência do sistema exposta na Equação 12. Reescrevendo-a, foi encontrada a Equação 15.

$$(\tau s + 1) \cdot \left( s + \frac{2}{\theta} \right) \cdot T(s) = K \cdot \left( -s + \frac{2}{\theta} \right) \cdot Q_i(s) \quad (13)$$

$$\left[ \tau s^2 + \left( \frac{2\tau}{\theta} + 1 \right) s + \frac{2}{\theta} \right] T(s) = \left[ -Ks + \frac{2K}{\theta} \right] Q_i(s) \quad (14)$$

$$\left[ s^2 + \left( \frac{2}{\theta} + \frac{1}{\tau} \right) s + \frac{2}{\tau\theta} \right] T(s) = \left[ -Ks + \frac{2K}{\theta} \right] Q_i(s) \quad (15)$$

Foi aplicada a Transformada Inversa de Laplace na Equação 15, considerando as condições iniciais nulas [Carneiro, ]. Com isso, foi encontrada a equação exposta em Equação 16.

$$\ddot{T} + \left( \frac{2}{\theta} + \frac{1}{\tau} \right) \dot{T} + \frac{2}{\tau\theta} T = -K\dot{Q}_i + \frac{2K}{\theta} Q_i \quad (16)$$

A partir da Equação 16 é possível encontrar as variáveis de fase, em que cada variável de estado subsequente é a derivada da variável de estado anterior[Carneiro, ]. Essas variáveis foram expressa em Equação 17.

$$\begin{cases} x_1 = T \\ x_2 = \dot{T} \\ y_1 = Q_i \end{cases} \quad (17)$$

Derivando as variáveis de fase expostas em Equação 17, foram obtidos os valores demonstrados em Equação 18.

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{y}_1 = \dot{Q}_i \\ \dot{x}_2 = \ddot{T} = -\frac{2}{\tau\theta}x_1 - \left( \frac{2}{\theta} + \frac{1}{\tau} \right)x_2 - K\dot{Q}_i + \frac{2K}{\theta}Q_i \end{cases} \quad (18)$$

Com os valores demonstrados nas equações Equação 17 e Equação 18 foi possível obter a modelagem no espaço de estados, demonstrado em Equação 19.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{2}{\tau\theta} & -\left( \frac{2}{\theta} + \frac{1}{\tau} \right) \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ K \end{bmatrix} \cdot q_i(t)$$

$$y = \begin{bmatrix} \frac{2}{\theta} & -1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

(19)

## 5 Função de transferência da planta

Por ser uma aplicação real, em que as características variam de acordo com as especificidades de cada forno, os parâmetros  $R$  e  $C$  da planta são difíceis de serem determinados de forma analítica. Sendo assim, o modelo da planta foi obtida a partir da identificação do sistema por meio de sinais de teste [Sena, 2017].

De maneira geral, é possível determinar os parâmetros de uma planta aplicando um sinal de entrada e observando o comportamento da saída. A partir dos resultados obtidos, são aplicáveis alguns métodos para determinar os parâmetros, como a aproximação de uma curva utilizando o método dos mínimos quadrados ou métodos baseados em gráficos da curva de reação a partir de um ponto de operação [Sena, 2017]. Nesse trabalho, será utilizado a segunda opção, mais especificamente utilizando o método de **Ziegler e Nichols**.

### 5.1 Ziegler e Nichols

Nesse método, as variáveis do sistema podem ser encontradas de acordo com o demonstrado na Figura 4. A reta em vermelho é traçada de acordo com o ponto de máxima inclinação da entrada ao degrau. O valor de  $\theta$  é encontrado na interseção dessa reta com o eixo x e o valor de  $\tau$  é a diferença entre o ponto em que se atinge 63,2 % do valor de regime e a interseção da

reta. O ganho K do sistema é dado pela Equação 20 , em que  $y_f - y(0)$  corresponde a variação da saída e  $u_f - u(0)$  equivaale à amplitude do degrau aplicado na entrada.

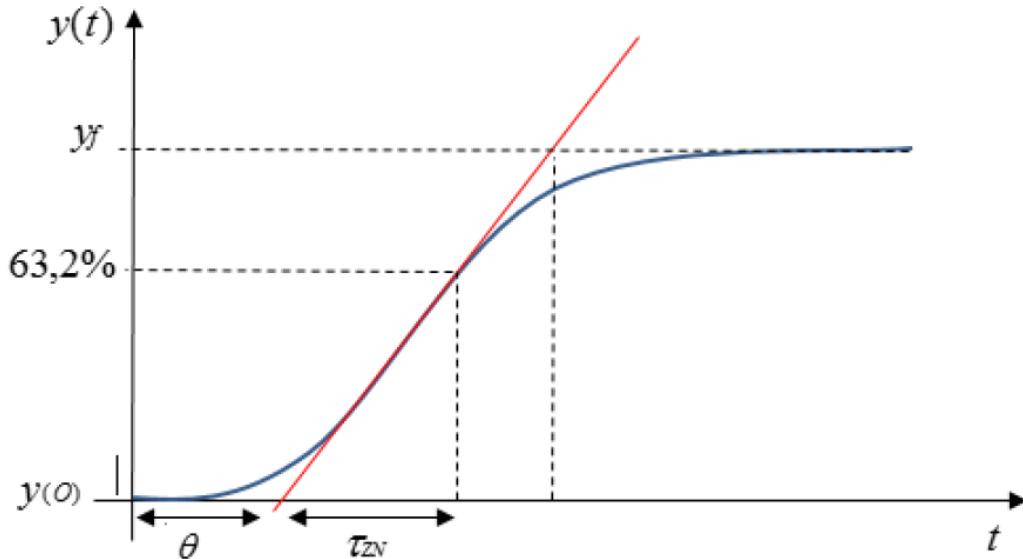


Figura 4: Método de Ziegler e Nichols para identificação de parâmetros.

Fonte: [Sena, 2017]

$$K = \frac{y_f - y(0)}{u_f - u(0)} \quad (20)$$

Para a identificação do modelo, foram utilizados os dados obtidos em [Sena, 2017], demonstrados na Figura 5, em que efetuou-se o degrau na entrada de 120 °C - equivalente a 60% da escala.

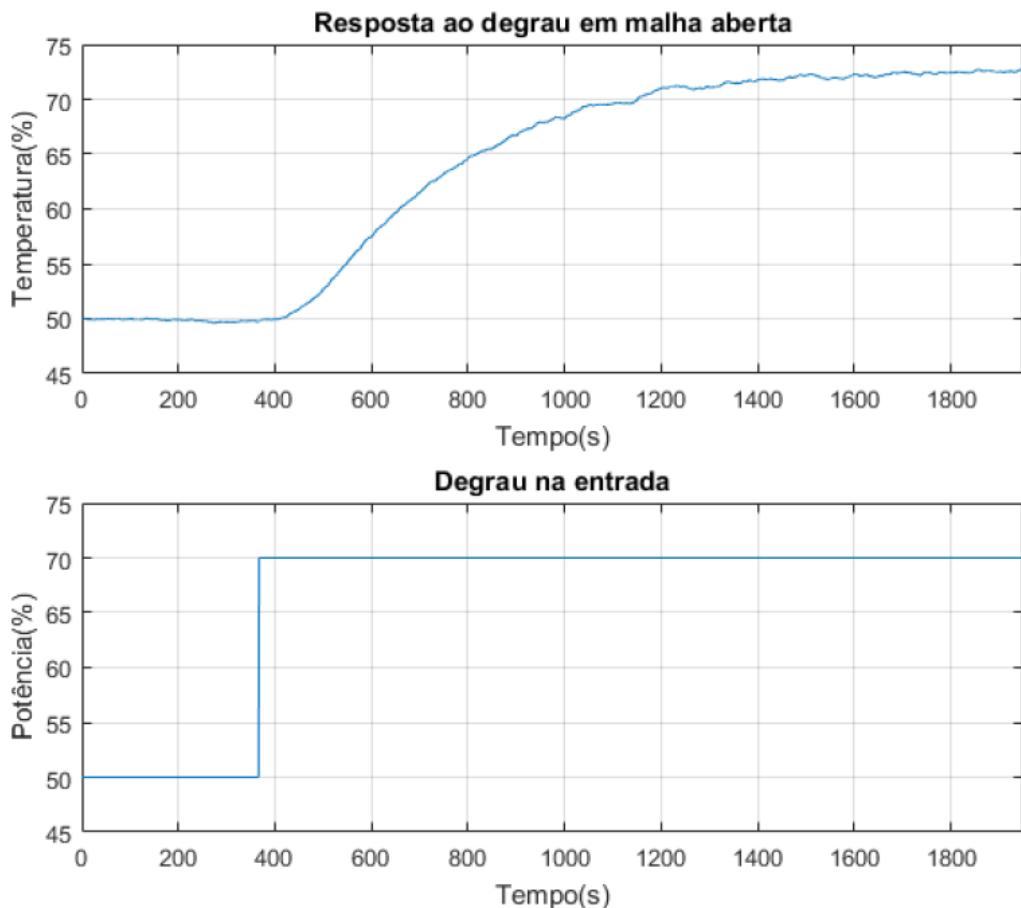


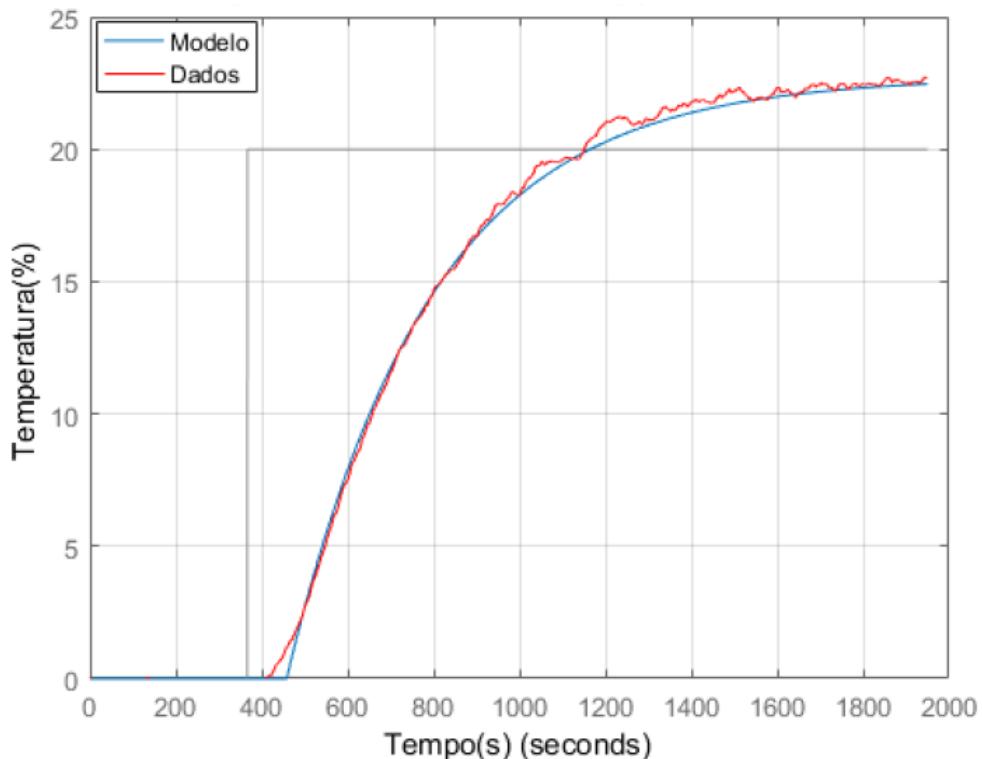
Figura 5: Curva de reação em função de uma resposta ao degrau para um ponto de operação de 120 °C.

Fonte: [Sena, 2017]

Através do método de obtenção dos parâmetros citados na Figura 4 e Equação 20, foram obtidos os valores expostos na Equação 21.

$$\begin{cases} K = 1,13 \\ \tau = 443 \\ \theta = 84 \end{cases} \quad (21)$$

Tais valores demonstram ser uma boa aproximação, conforme exposto na Figura 6, feita pelo autor do texto base para a escrita dessa trabalho [Sena, 2017].

Figura 6: Resposta ao degrau em malha aberta de  $G(s)$  comparada ao modelo.

Fonte: [Sena, 2017]

Sendo assim, temos a seguinte função de transferência da planta, demonstrada na Equação 22.

$$G(s) = \frac{-1,13(s - \frac{2}{84})}{(443s + 1) \cdot (s + \frac{2}{84})} \approx \frac{-1,13s + 0,0269047619047619}{443s^2 + 11,547619047619s + 0,0238095238095238} \quad (22)$$

A Figura 7 demonstra a resposta do sistema a uma entrada ao degrau. Para verificar a atuação do atraso de propagação no sistema, foram plotados os resultados para as funções de transferência considerando o atraso de propagação (linha em verde) e sem considerar o atraso de propagação (linha em vermelho), com o objetivo de comparar as respostas. A função de transferência para saída sem atraso é demonstrada em Equação 23. Para realizar tal plot, foi utilizado o código exposto em Código 1, feito em Python.

$$G(s) = \frac{1,13}{443s + 1} \quad (23)$$

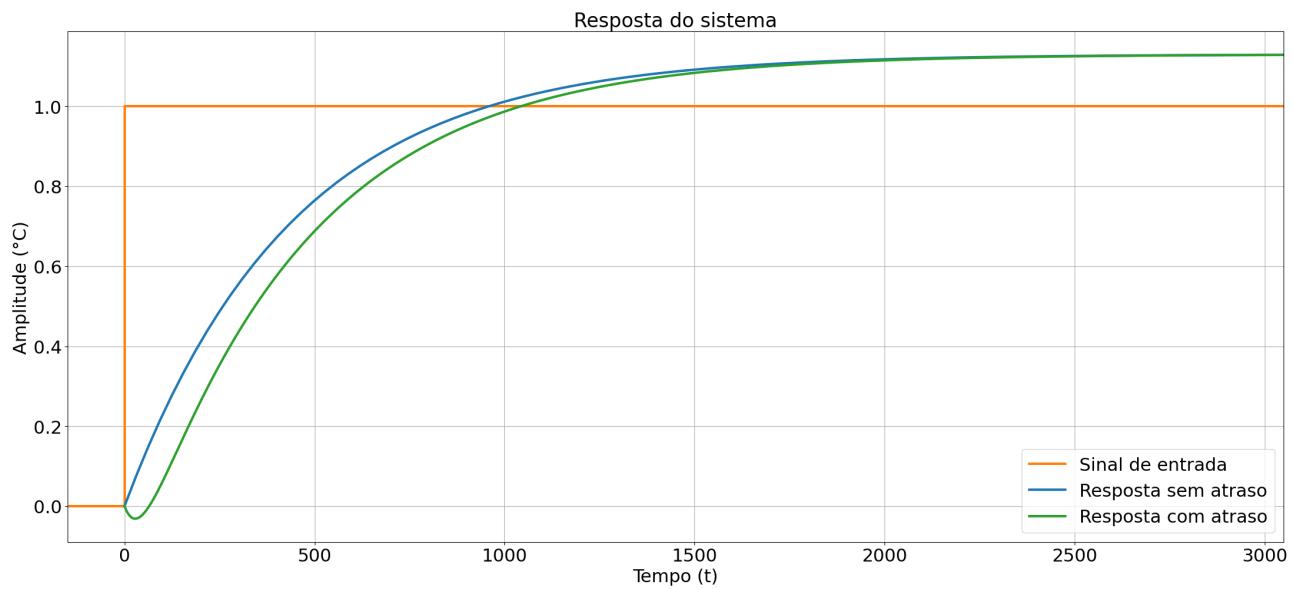


Figura 7: Respostas do sistema em malha aberta.

Fonte: Autoria Própria.

É possível notar uma diferença entre o gráfico obtido na Figura 6 e os da Figura 7 em um curto período de tempo que sucede o degrau de entrada. A linha em azul representa uma resposta sem considerar atraso de propagação, sendo assim, não é visto a demora para resposta do gráfico. Já a linha em verde demonstra a resposta considerando o atraso de propagação utilizando a Aproximação de Padé, a qual injeta um zero no semiplano direito tornando o sistema de fase não mínima, visível através do decrescimento inicial do gráfico.

A seguir, na Seção 6, serão feitas as análises de polos e zeros utilizando ambas considerações.

## 6 Análise de Polos e Zeros

As figuras Figura 8 e Figura 9 apresentam os mapas de polo e zero da função de transferência planta não considerando e considerando o atraso de propagação, demonstradas nas equações Equação 23 e Equação 22, respectivamente. A Figura 8 apresenta somente um polo no semiplano esquerdo e no eixo real em  $p_1 = -0,00225734 + j0$ , o que já era esperado devido ao sistema ser de primeira ordem. A Figura 9 apresenta dois polos  $p_2 = -0,00225734 + j0$ , resultante do sistema e  $p_1 = -0,02380952 + j0$ , resultante da Aproximação de Padé para o atraso de tempo. Além disso, a Figura 9 apresenta, conforme dito anteriormente, um zero no semiplano direito. Os códigos utilizados para gerar os plots das figuras Figura 8 e Figura 9 foram expostos em Código 2 e Código 3.

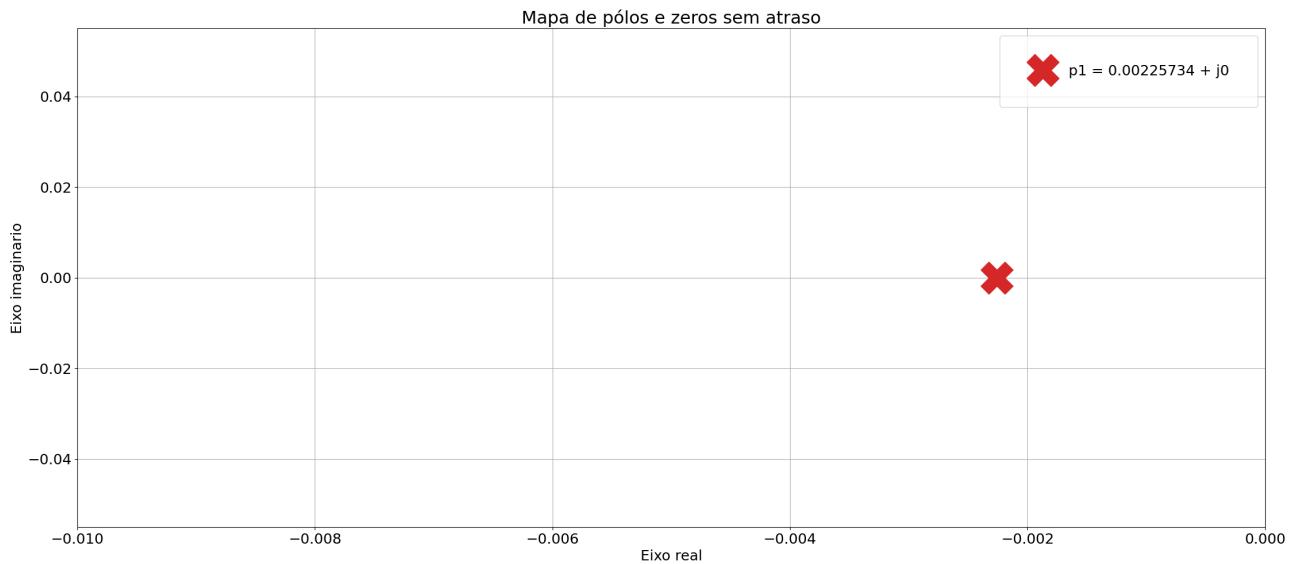


Figura 8: Mapa de polos e zeros sem atraso.

Fonte: Autoria Própria.

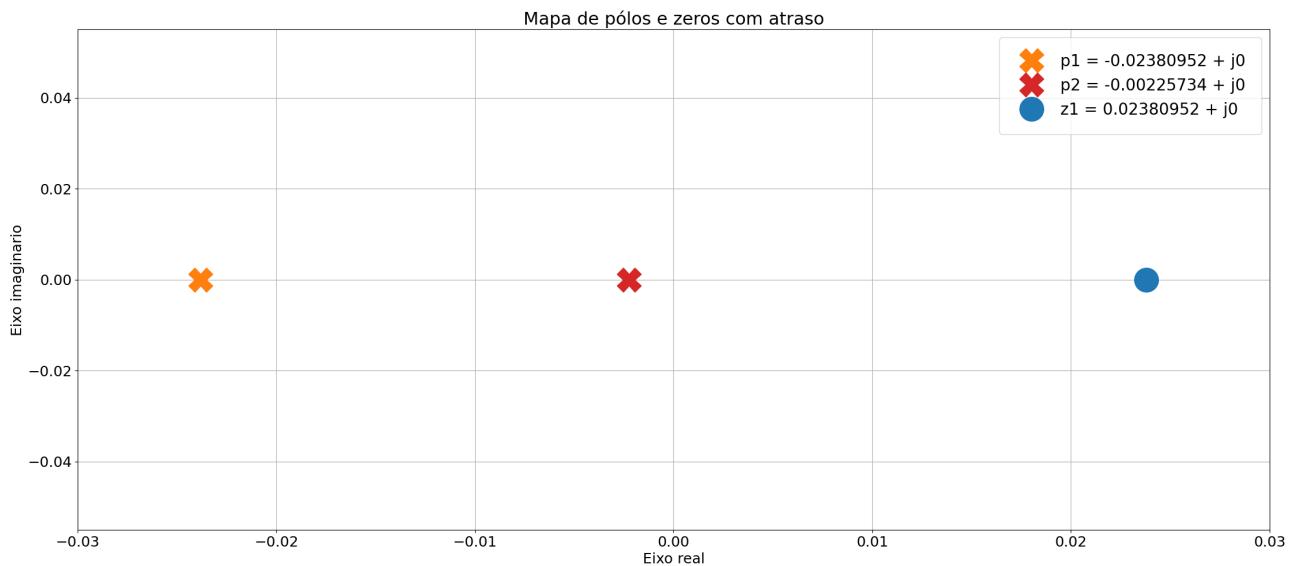


Figura 9: Mapa de polos e zeros com atraso.

Fonte: Autoria Própria.

A seguir, nas subseções Subseção 6.1 e Subseção 6.2 serão feitas as análises dos parâmetros de caracterização do sistema de maneira separada.

## 6.1 Sistema sem atraso

Para o sistema sem atraso, foi considerada a função de transferência exposta na Equação 23 e o mapa de polos e zeros da Figura 8.

O sistema sem atraso pode ser modelado, conforme dito anteriormente, como um **sistema de primeira ordem**. O ganho estático para um degrau unitário representa o valor que a

resposta assume em  $t \rightarrow \infty$ . É possível encontrar esse valor utilizando a equação para ganho estático, conforme visto nas equações Equação 24 e Equação 25.

$$\begin{aligned}\lim_{t \rightarrow \infty} y(t) &= \lim_{s \rightarrow 0} G(s) \\ \lim_{t \rightarrow \infty} y(t) &= \lim_{s \rightarrow 0} \frac{1,13}{443s + 1} \\ \lim_{t \rightarrow \infty} y(t) &= 1,13\end{aligned}\tag{24}$$

$$K = 1,13\tag{25}$$

Para um sistema de primeira ordem, a **constante de tempo (T)** é dada pelo termo que acompanha o fator s do denominador.

$$T = \tau = 433 \text{ s}\tag{26}$$

O **tempo de acomodação ( $T_a$ )** pode ser definido pelo tempo que o sistema demora até atingir 95% ( $T_{a5\%}$ ) ou 98% ( $T_{a2\%}$ ) do seu regime permanente, sem oscilar para fora desse limite. Essa variável pode ser definida através de um múltiplo da constante de tempo, conforme pode ser vista na equação Equação 27 [Ogata, 2003].

$$\begin{aligned}T_{a5\%} &= 3 \cdot T \\ T_{a2\%} &= 4 \cdot T\end{aligned}\tag{27}$$

Sendo assim, obtemos os valores para o tempo de acomodação descritos Equação 28.

$$\begin{aligned}T_{a5\%} &= 1299 \text{ s} \\ T_{a2\%} &= 1772 \text{ s}\end{aligned}\tag{28}$$

Os valores obtidos para o ganho estático, a constante de tempo e o tempo de acomodação podem ser visualizados no gráfico da figura Figura 10. O código para gerar esse gráfico está exposto em Código 4.

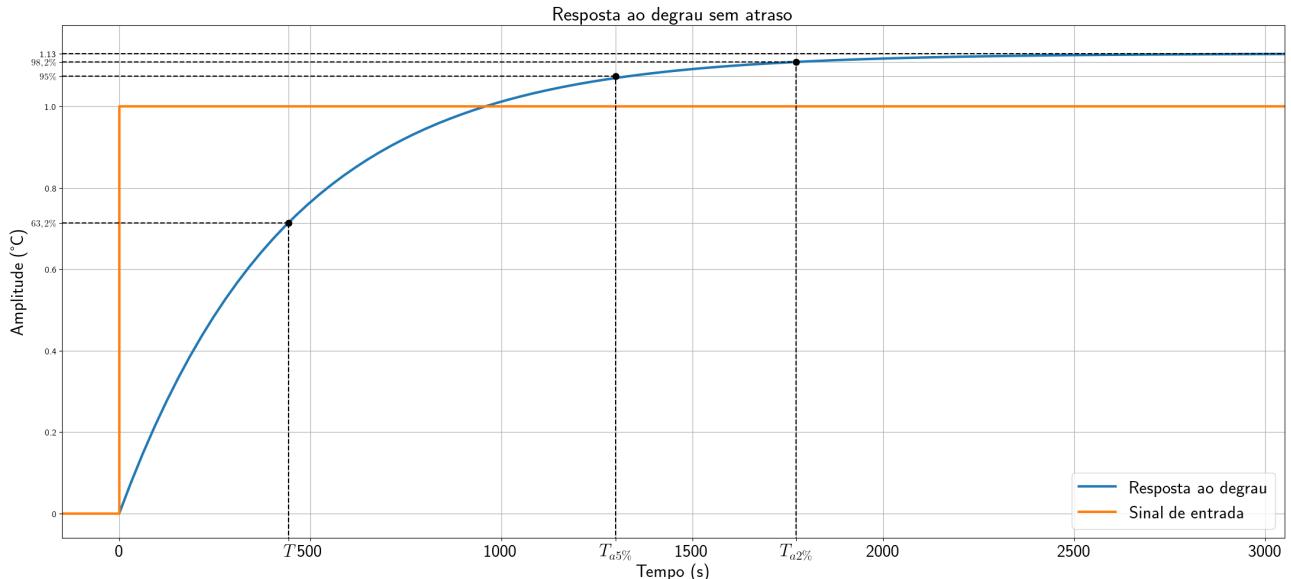


Figura 10: Resposta ao degrau sem atraso.

Fonte: Autoria Própria.

## 6.2 Sistema com atraso

Para o sistema com atraso, foi considerada a função de transferência exposta na Equação 29.

$$G(s) = \frac{1,13}{443s + 1} \cdot e^{-84s} \quad (29)$$

A função de transferência exposta em Equação 29 demonstra uma saída ao degrau igual a do gráfico da Figura 10 deslocado 84 s no tempo, conforme pode ser visto na Figura 11. O código para gerar esse gráfico está exposto em Código 5.

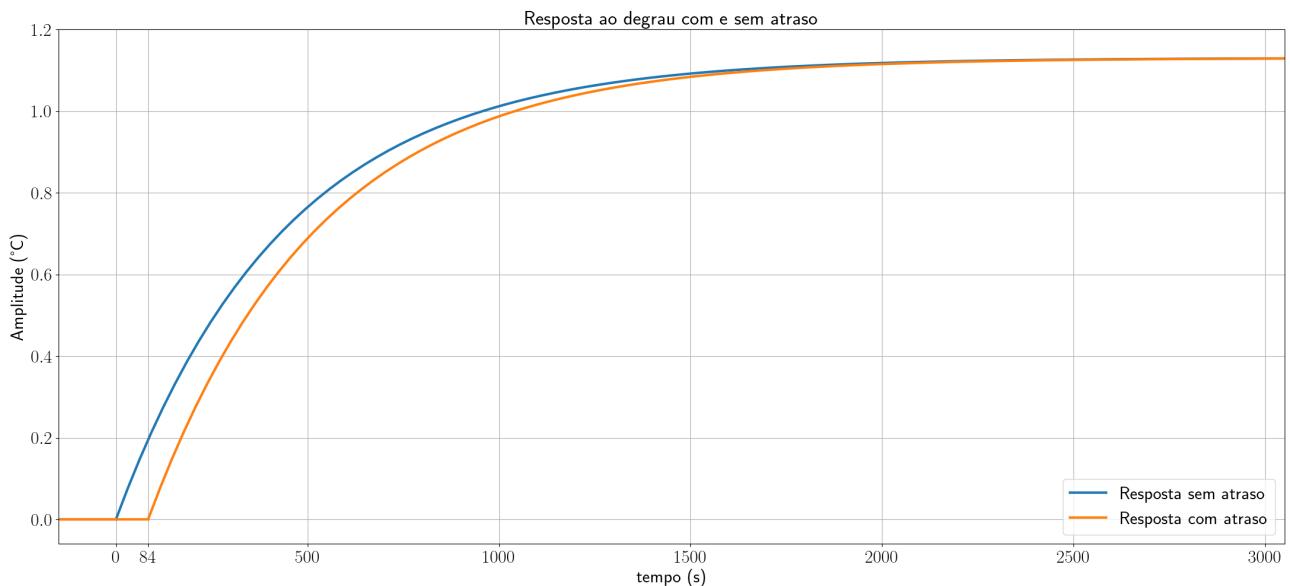


Figura 11: Resposta ao degrau com e sem atraso.

Fonte: Autoria Própria.

Sendo assim, para o sistema com atraso, é possível falar que os parâmetros de tempo se deslocam 84 s dos parâmetros originais e o ganho estático permanece o mesmo, conforme demonstrado na Equação 30.

$K_d = K = 1,13$	
$T_d = T + 84 = 527 \text{ s}$	
$T_{da5\%} = T_{a5\%} + 84 = 1383 \text{ s}$	
$T_{da2\%} = T_{a2\%} + 84 = 1856 \text{ s}$	

(30)

A Aproximação de Padé para o sistema, exposta em Equação 22 representa uma aproximação bastante razoável para o sistema com atraso, conforme demonstrado na Figura 12, em que é possível observar um erro pequeno decorrente do zero de fase não mínima nos segundos iniciais e uma representação bastante fiel ao real após os segundos iniciais. O código para gerar esse gráfico está exposto em Código 6.

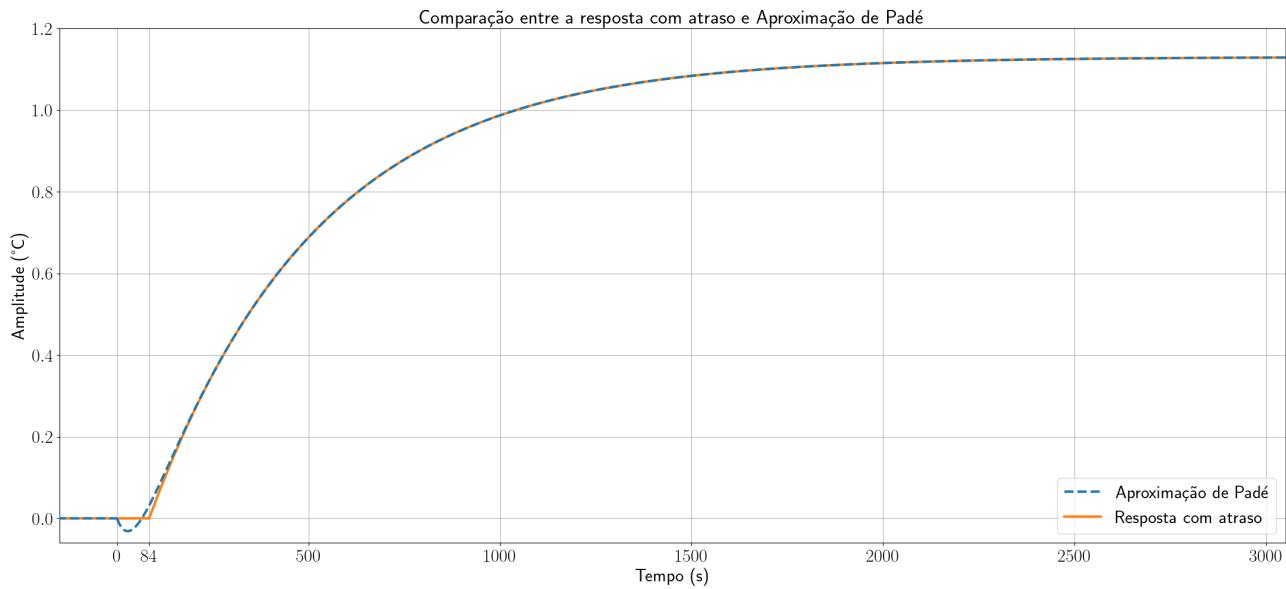


Figura 12: Comparação entre a resposta com atraso e Aproximação de Padé.

Fonte: Autoria Própria.

As figuras Figura 13 e Figura 14 apresentam os parâmetros encontrados na Equação 30 nos gráficos de resposta ao degrau do sistema com atraso puro no tempo e com a Aproximação de Padé. Os códigos, feitos em Python, utilizados para gerar os gráficos estão expostos em Código 7 e Código 8, respectivamente.

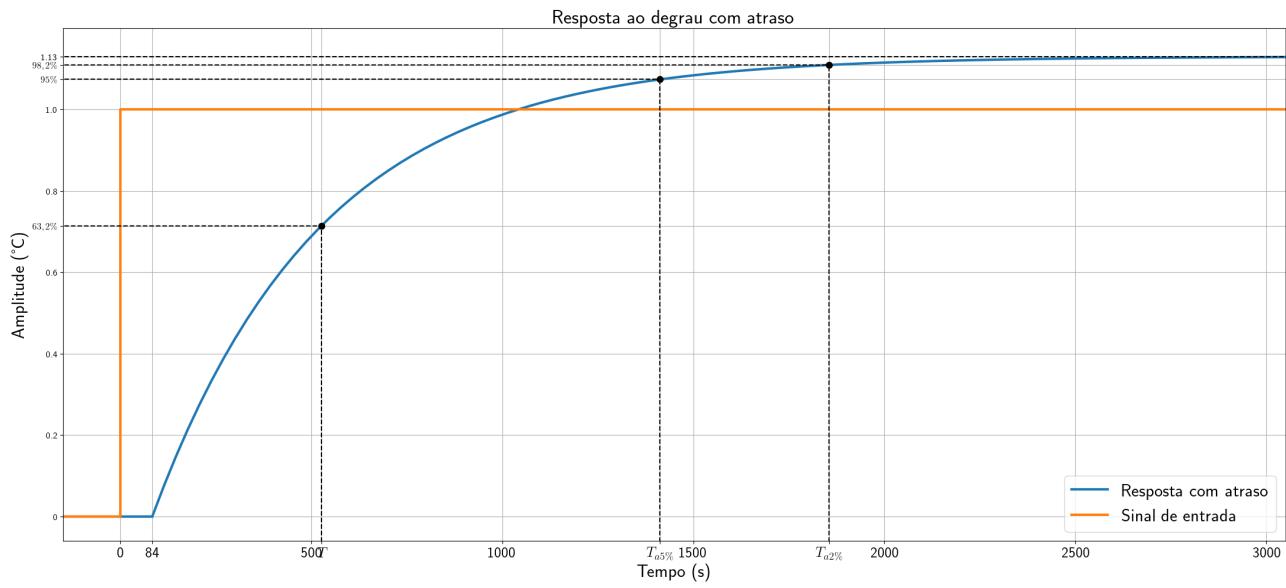


Figura 13: Resposta ao degrau com atraso.

Fonte: Autoria Própria.

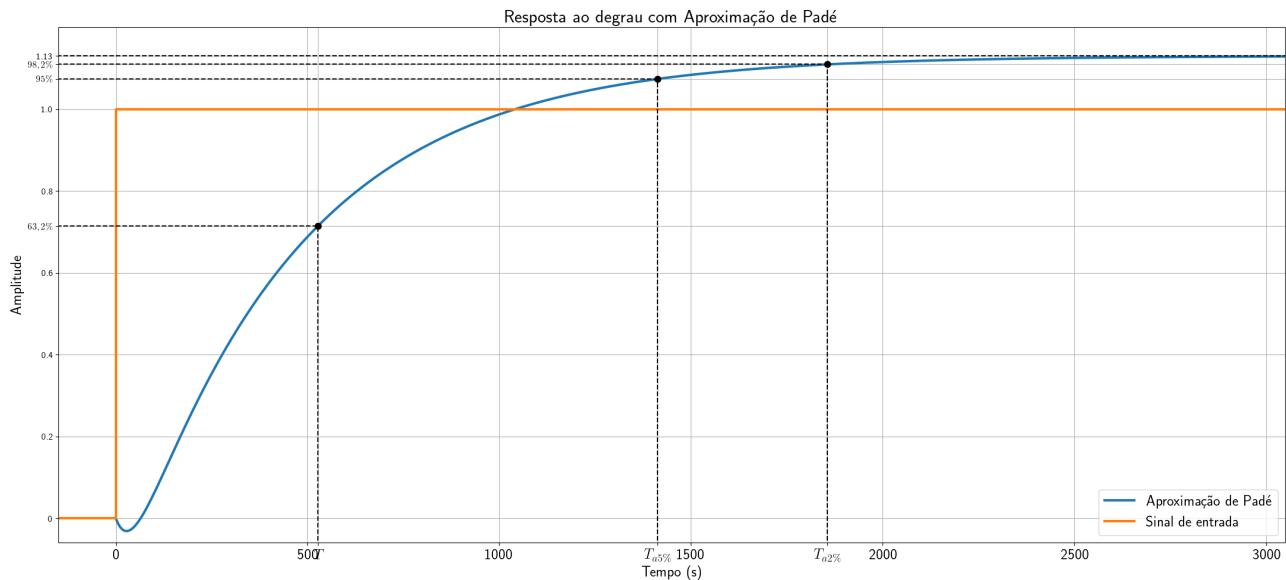


Figura 14: Resposta ao degrau com Aproximação de Padé.

Fonte: Autoria Própria.

## 7 Análise na Frequência

Para fazer a análise na frequência e o diagrama de bode, iremos utilizar as funções de transferência sem atraso e com atraso considerando a aproximação de Padé, descritas, respectivamente, nas equações Equação 23 e Equação 22.

### 7.1 Sistema sem atraso

Para a análise na frequência, iremos considerar a função de transferência exposta na Equação 23 explicitada da maneira demonstrada em Equação 31.

$$G(s) = \underbrace{1,13}_{\text{Proporcional}} \cdot \underbrace{\frac{1}{443s + 1}}_{\text{Polo}} \quad (31)$$

A equação Equação 31 foi explicitada de maneira a demonstrar de forma fácil os termos da função de transferência, em que é possível notar um termo proporcional e um polo. Será feito a análise de cada termo separado e, logo após, somados. A partir da Equação 32 serão feitos os cálculos para o termo proporcional  $G_1(s)$ .

$$G_1(s) = 1,13 \quad (32)$$

$$G_1(jw) = 1,13 \quad (33)$$

A Equação 34 apresenta os termos de módulo e fase de acordo com a frequência para o termo proporcional. A Figura 15 demonstra o diagrama de bode para esse termo. O código, feito em Python, para o gráfico da Figura 15 está exposto em Código 9.

$$|G_1(jw)|_{dB} = 1,062 \quad \angle G_1(jw) = 0^\circ \quad (34)$$

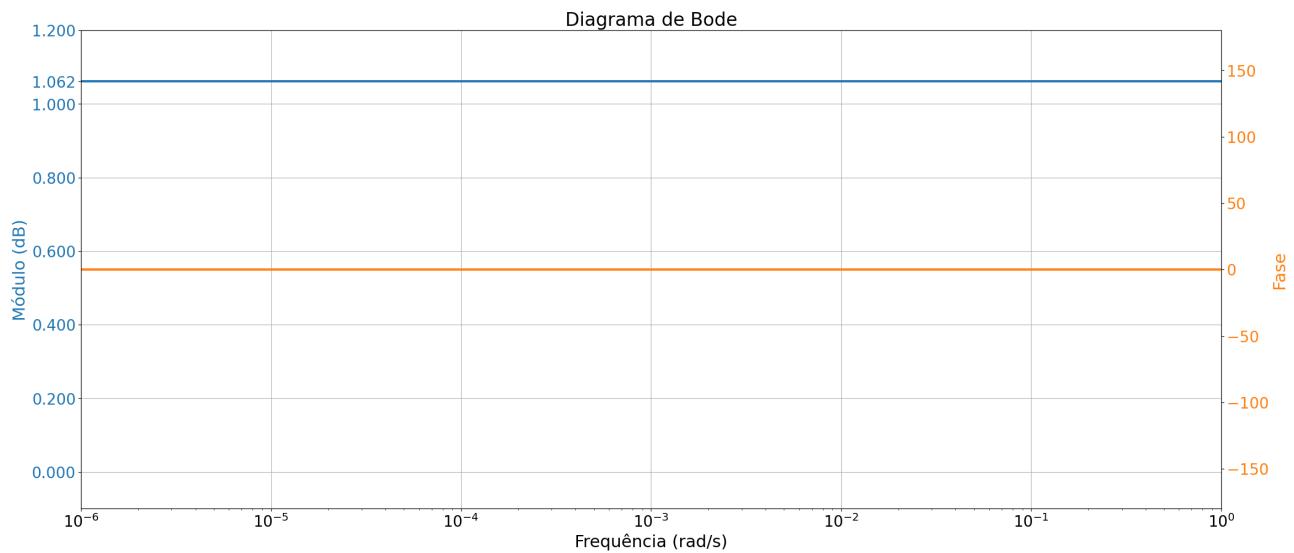


Figura 15: Diagrama de bode para o termo proporcional sem atraso.

Fonte: Autoria Própria.

A Equação 35 representa o polo explicitado na Equação 31. A partir da Equação 35 serão feitos os cálculos para o polo  $G_2(s)$ .

$$G_2(s) = \frac{1}{443s + 1} \quad (35)$$

$$G_2(jw) = \frac{1}{433jw + 1} \quad (36)$$

$$|G_2(jw)|_{dB} = -10 \cdot \log(443^2 w^2 + 1) \quad \angle G_2(jw) = -\arctan(443w) \quad (37)$$

A Equação 37 apresenta os termos de módulo e fase de acordo com a frequência para o polo. A Figura 16 demonstra o diagrama de bode para esse termo. O código, feito em Python, para o gráfico da Figura 16 está exposto em Código 10.

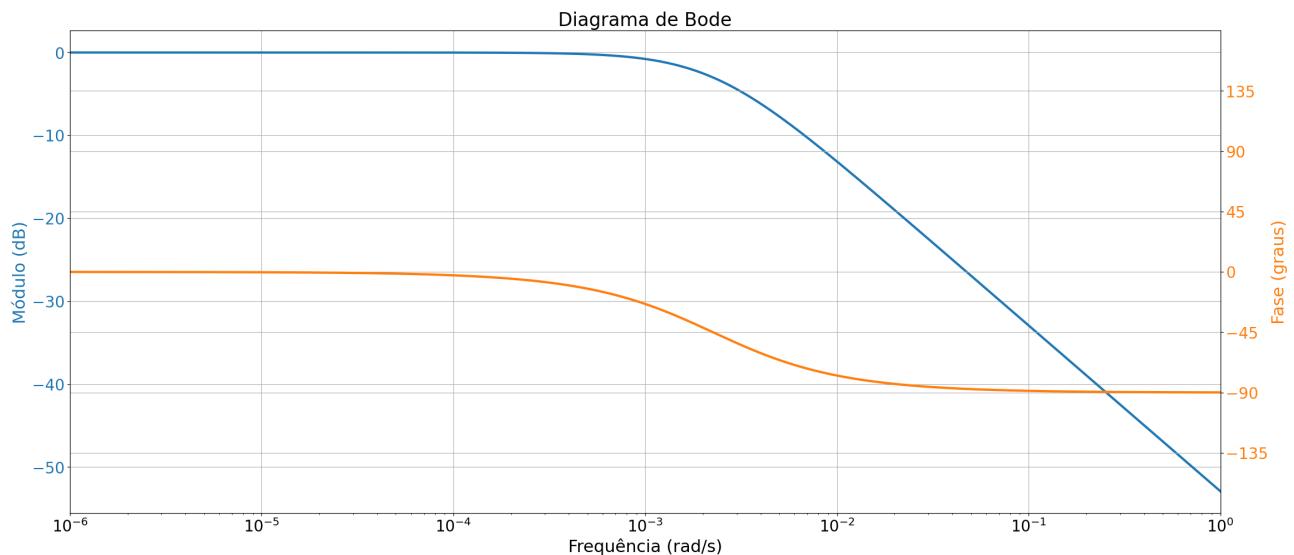


Figura 16: Diagrama de bode para o polo sem atraso.

Fonte: Autoria Própria.

Para conseguir o Diagrama de Bode completo do sistema, basta somar os diagramas obtidos nas figuras Figura 15 e Figura 16. O Diagrama de Módulo para o sistema está exposto na Figura 17 e o de Fase na Figura 18, sendo ambos gráficos gerados pelo Código 11, feito em Python.

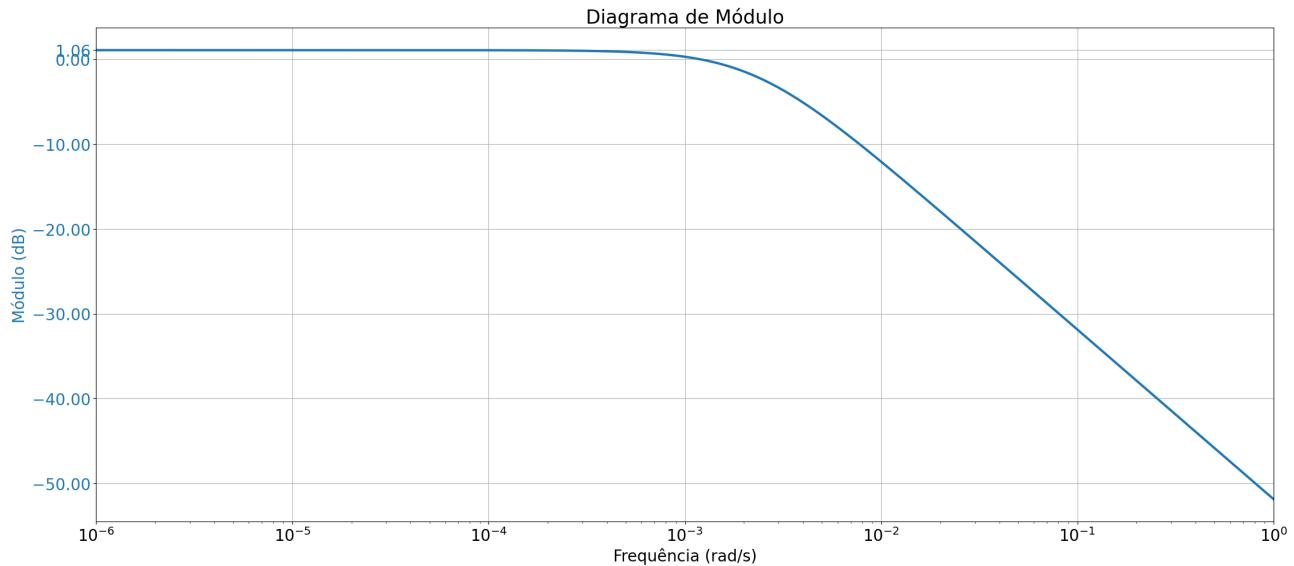


Figura 17: Diagrama de Módulo para o sistema sem atraso.

Fonte: Autoria Própria.

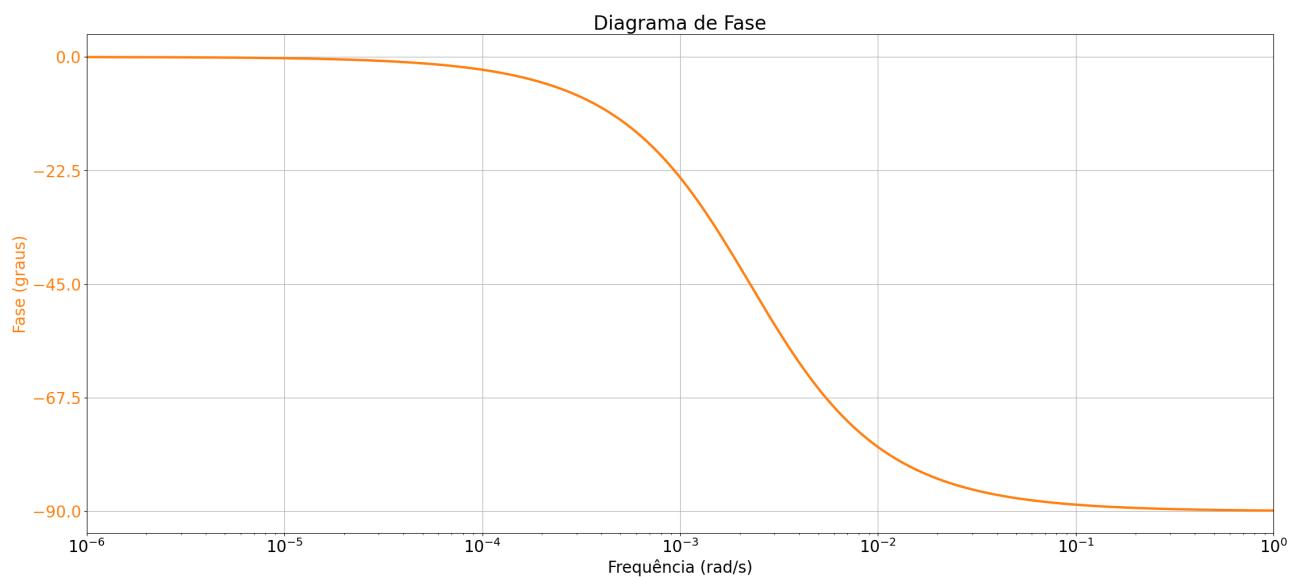


Figura 18: Diagrama de Fase para o sistema sem atraso.

Fonte: Autoria Própria.

Através dos gráficos das figuras Figura 17 e Figura 18 é possível obter a frequência de corte e as frequências e margens de fase e ganho, conforme demonstrada na Equação 38.

$$\begin{aligned} w_{\Phi_m} &= 1,23 \times 10^{-3} \text{ rad s}^{-1} & w_{G_m} &= \infty & w_c &= 2,84 \times 10^{-3} \text{ rad s}^{-1} \\ \Phi_m &= 151,403^\circ & G_m &= \infty \end{aligned} \quad (38)$$

Através do Diagrama de Bode obtidos nas figuras Figura 17 e Figura 18, foram escolhidas três frequências para serem analisadas os sinais de saída  $w_1 = 1 \times 10^{-4} \text{ rad s}^{-1}$ ,  $w_2 = 1 \times 10^{-3} \text{ rad s}^{-1}$  e  $w_3 = 1 \times 10^{-1} \text{ rad s}^{-1}$ . Para a frequência  $w_1$ , nos gráficos das figuras Figura 17 e Figura 18, observa-se uma ganho próximo a 1,06 dB e uma defasagem bem pequena, sendo assim, espera-se que a resposta do sistema para uma entrada senoidal de amplitude  $A = 1$  e  $\omega = w_1$  seja bem parecida com a entrada, com uma amplitude um pouco superior e leve defasagem. A Figura 19 demonstra os gráficos de entrada e saída do sistema para  $\omega = w_1$ .

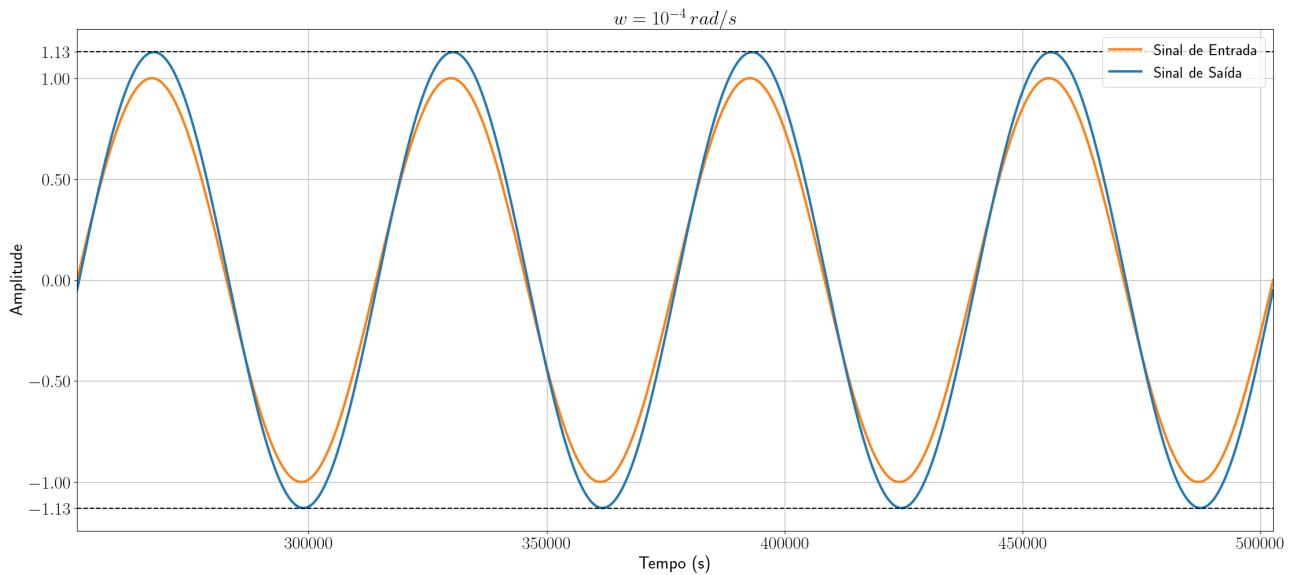
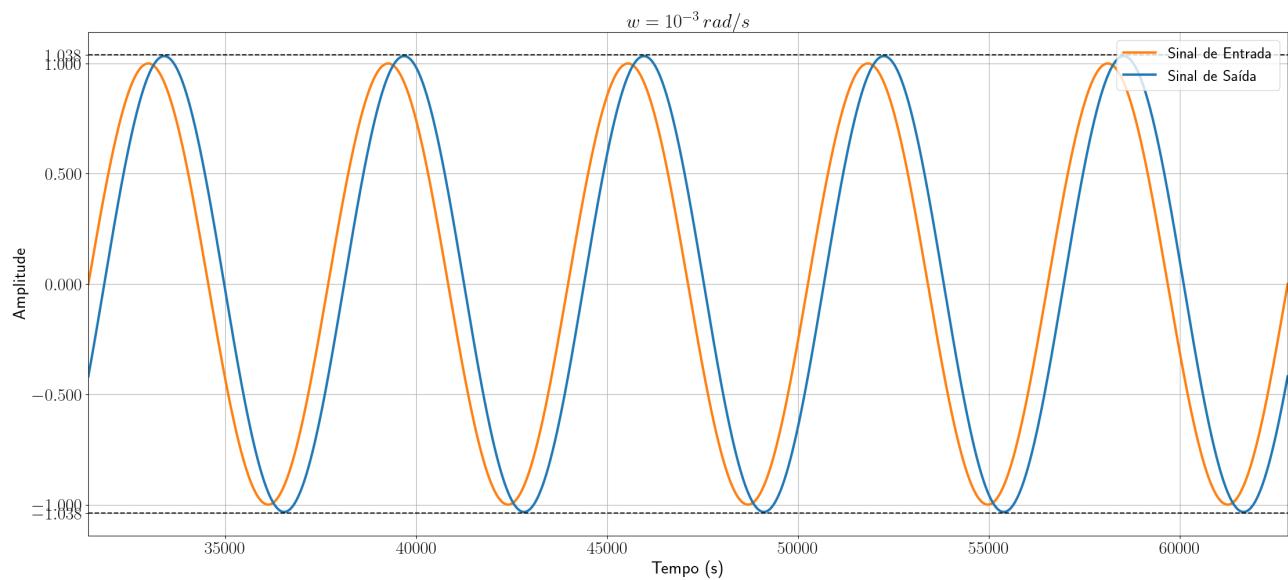


Figura 19: Entrada e saída do sistema sem atraso para  $\omega = 1 \times 10^{-4} \text{ rad s}^{-1}$ .

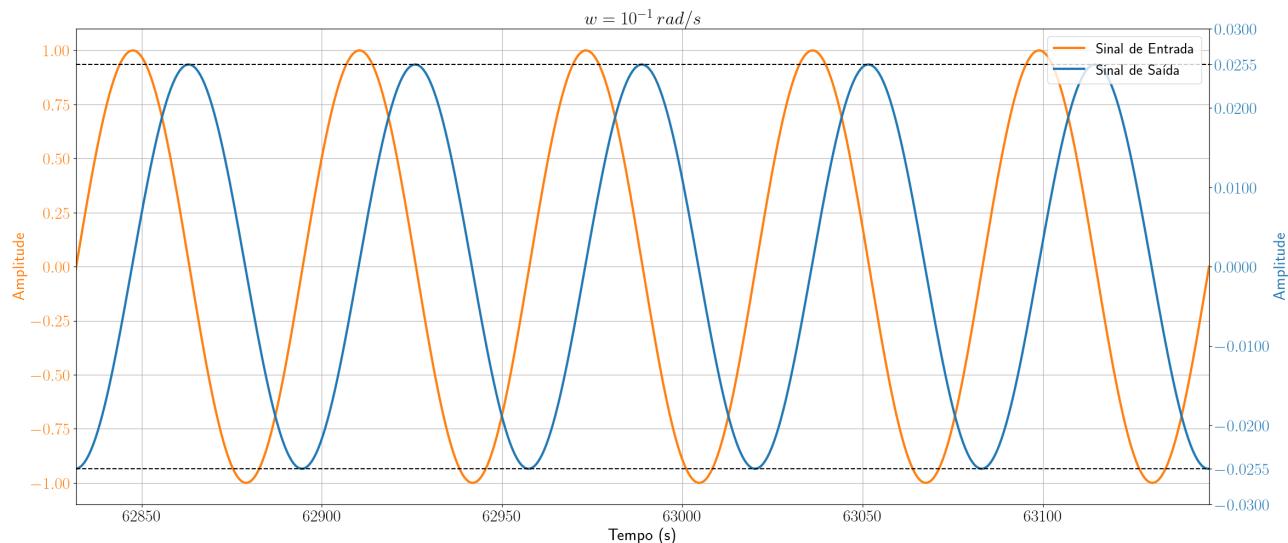
Fonte: Autoria Própria.

Para a frequência  $w_2$ , observa-se uma ganho próximo a 0 dB e uma defasagem considerável, próxima a 22,5°, sendo assim, espera-se que a resposta do sistema para uma entrada senoidal de amplitude  $A = 1$  e  $\omega = w_1$  tenha uma amplitude levemente superior a de entrada e uma defasagem considerável. A Figura 20 demonstra os gráficos de entrada e saída do sistema para  $\omega = w_2$ .

Figura 20: Entrada e saída do sistema sem atraso para  $\omega = 1 \times 10^{-3} \text{ rad s}^{-1}$ .

Fonte: Autoria Própria.

Para a frequência  $w_3$ , observa-se uma ganho próximo a  $-33 \text{ dB}$  e uma alta defasagem, próxima a  $90^\circ$ , sendo assim, espera-se que a resposta do sistema para uma entrada senoidal de amplitude  $A = 1$  e  $\omega = w_1$  tenha uma amplitude muito atenuada e uma defasagem de  $-90^\circ$ . A Figura 21 demonstra os gráficos de entrada e saída do sistema para  $\omega = w_3$ .

Figura 21: Entrada e saída do sistema sem atraso para  $\omega = 1 \times 10^{-1} \text{ rad s}^{-1}$ .

Fonte: Autoria Própria.

Para realizar o plot das figuras Figura 19, Figura 20 e Figura 21 foi utilizado o código, em Python, exposto em Código 12.

## 7.2 Sistema com atraso

Para o sistema com atraso, foi considerada a função de transferência com a Aproximação de Padé, demonstrada em Equação 22. Essa equação foi explicitada da maneira conforme vista em Equação 39.

$$G(s) = \underbrace{1,13}_{\text{Proporcional}} \cdot \underbrace{\frac{1}{443s+1}}_{\text{Polo}} \cdot \underbrace{\frac{1}{\frac{84}{2}s+1}}_{2^{\text{o}} \text{ Polo}} \cdot \underbrace{\left( -\frac{84}{2}s + 1 \right)}_{\text{Zero}} \quad (39)$$

Os termos **Proporcional** e **Polo** são termos provenientes da planta, sendo os cálculos e gráficos necessários para eles já expostos em Subseção 7.1, sendo assim não necessário refazê-los. Será feito a análise de **2º Polo** e **Zero**. A partir da Equação 40 serão feitos os cálculos para o termo **2º Polo**  $G_3(s)$ .

$$G_3(s) = \frac{1}{\frac{84}{2}s + 1} \quad (40)$$

$$G_3(jw) = \frac{1}{\frac{84}{2}jw + 1} \quad (41)$$

$$|G_3(jw)|_{dB} = -10 \log (42^2 w^2 + 1) \quad /G_3(jw) = -\arctan (42w) \quad (42)$$

A Equação 42 apresenta os termos de módulo e fase de acordo com a frequência para o **2º polo**. A Figura 22 demonstra o diagrama de bode para esse termo. O código, feito em Python, para o gráfico da Figura 22 está exposto em Código 13.

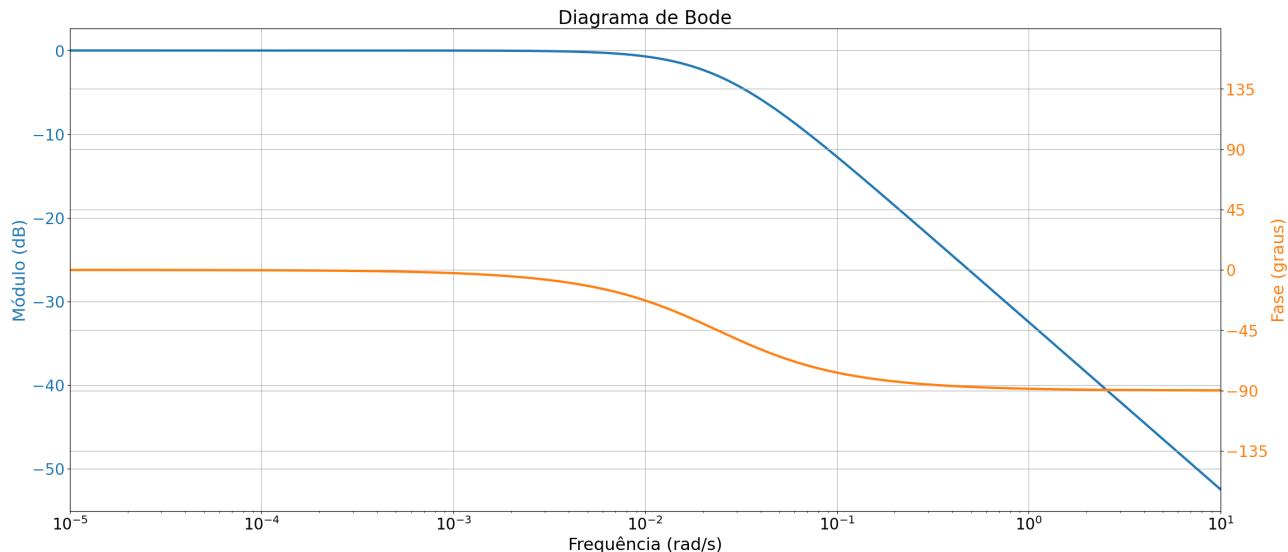


Figura 22: Diagrama de bode para o **2º polo** (com atraso).

Fonte: Autoria Própria.

A Equação 43 representa o zero explicitado na Equação 39. A partir da Equação 43 serão feitos os cálculos para o zero  $G_4(s)$ .

$$G_4(s) = -\frac{84}{2}s + 1 \quad (43)$$

$$G_4(jw) = -\frac{84}{2}jw + 1 \quad (44)$$

$$|G_4(jw)|_{dB} = 10 \cdot \log (42^2 w^2 + 1) \quad /G_4(jw) = -\arctan (42w) \quad (45)$$

A Equação 45 apresenta os termos de módulo e fase de acordo com a frequência para o zero. A Figura 23 demonstra o diagrama de bode para esse termo. O código, feito em Python, para o gráfico da Figura 23 está exposto em Código 14.

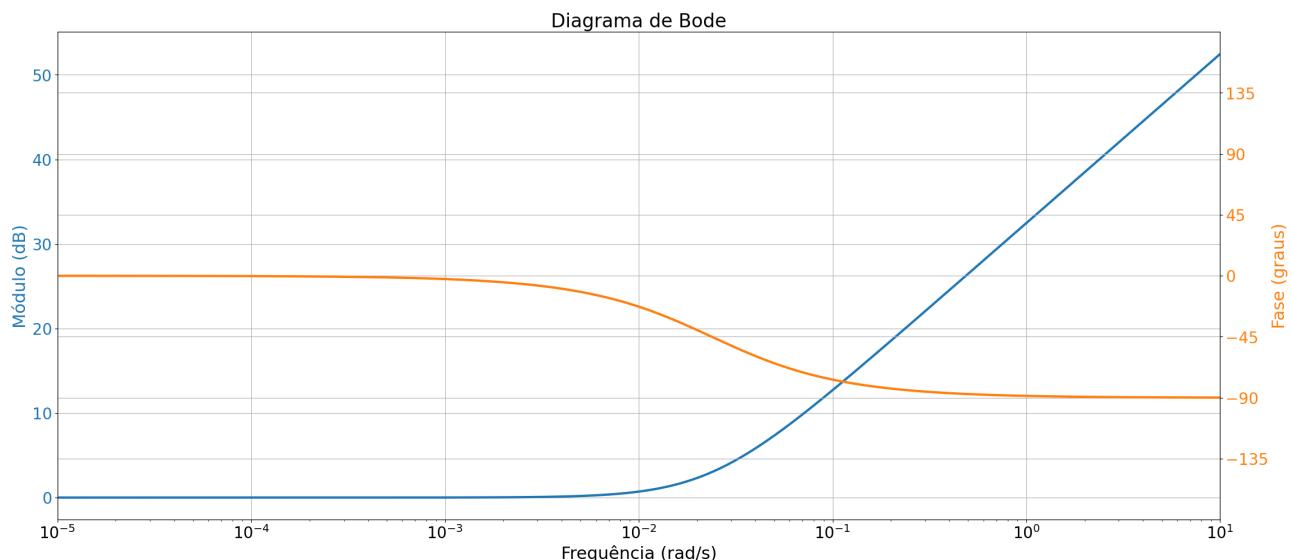


Figura 23: Diagrama de bode para o zero.

Fonte: Autoria Própria.

É interessante notar que o termo **Zero** representa um zero localizado no semi-plano direito do plano complexo, sendo assim um **zero de fase não-miníma**. Os zeros de fase não-miníma apresentam uma contribuição no gráfico de modulo em altas frequências de  $20 \text{ dB dec}^{-1}$  porém se difere dos zeros no semi-plano esquerdo no gráfico de fase, em que atuam de forma semelhante a um polo localizado no semiplano esquerdo, contribuindo com um decaimento de  $-90^\circ$  em altas frequências. Esse comportamento pode ser observado na Figura 23.

Para conseguir o Diagrama de Bode completo do sistema, basta somar os diagramas obtidos nas figuras Figura 15, Figura 16, Figura 22 e Figura 23. O Diagrama de Módulo para o sistema está exposto na Figura 24 e o de Fase na Figura 25, sendo ambos gráficos gerados pelo Código 15, feito em Python.

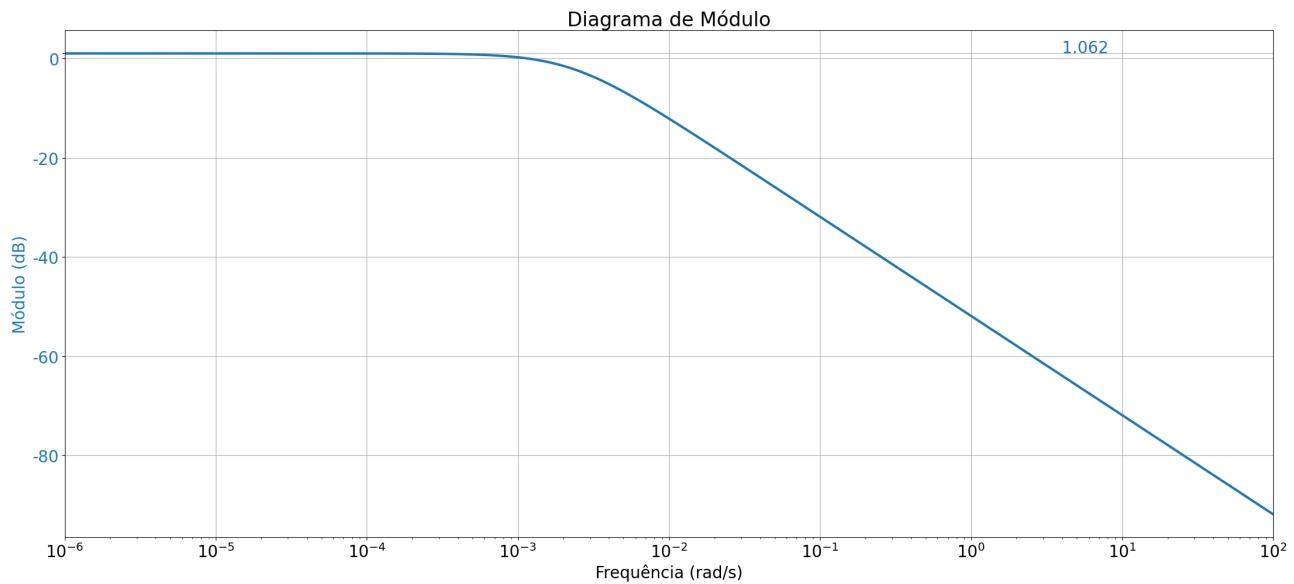


Figura 24: Diagrama de Módulo para o sistema com atraso.

Fonte: Autoria Própria.

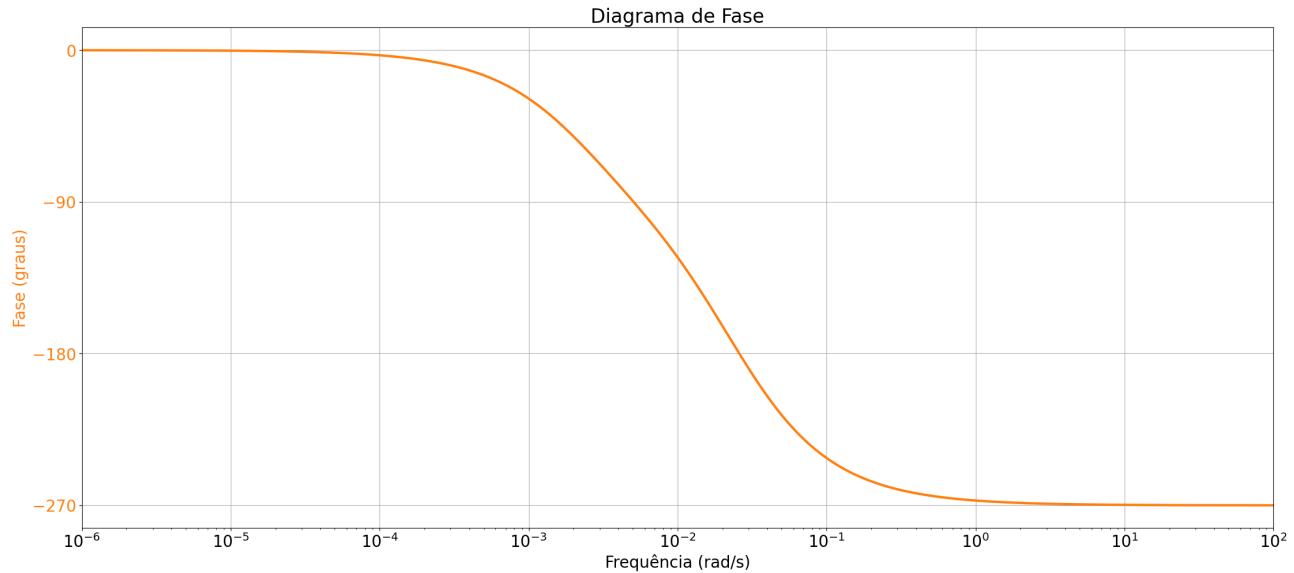


Figura 25: Diagrama de Fase para o sistema com atraso.

Fonte: Autoria Própria.

Através dos gráficos das figuras Figura 24 e Figura 25 é possível obter a frequência de corte e as frequências e margens de fase e ganho, conforme demonstrada na Equação 46.

$$\begin{aligned} w_{\Phi_m} &= 1,23 \times 10^{-3} \text{ rad s}^{-1} & w_{G_m} &= 2,59 \times 10^{-2} \text{ rad s}^{-1} & w_c &= 2,84 \times 10^{-3} \text{ rad s}^{-1} \\ \Phi_m &= 145,49^\circ & G_m &= 22,376 \text{ dB} \end{aligned} \quad (46)$$

Através do Diagrama de Bode obtidos nas figuras Figura 24 e Figura 25, foram escolhidas três frequências para serem analisadas os sinais de saída  $w_4 = 1 \times 10^{-5} \text{ rad s}^{-1}$ ,  $w_5 =$

$1 \times 10^{-2} \text{ rad s}^{-1}$  e  $w_6 = 1 \times 10^1 \text{ rad s}^{-1}$ . Para a frequência  $w_4$ , nos gráficos das figuras Figura 24 e Figura 25, observa-se uma ganho próximo a 1,06 dB e uma defasagem nula. Sendo assim, espera-se que a resposta do sistema para uma entrada senoidal de amplitude  $A = 1$  e  $\omega = w_4$  seja bem parecida com a entrada, com uma amplitude um pouco superior e sem defasagem. A Figura 26 demonstra os gráficos de entrada e saída do sistema para  $\omega = w_4$ .

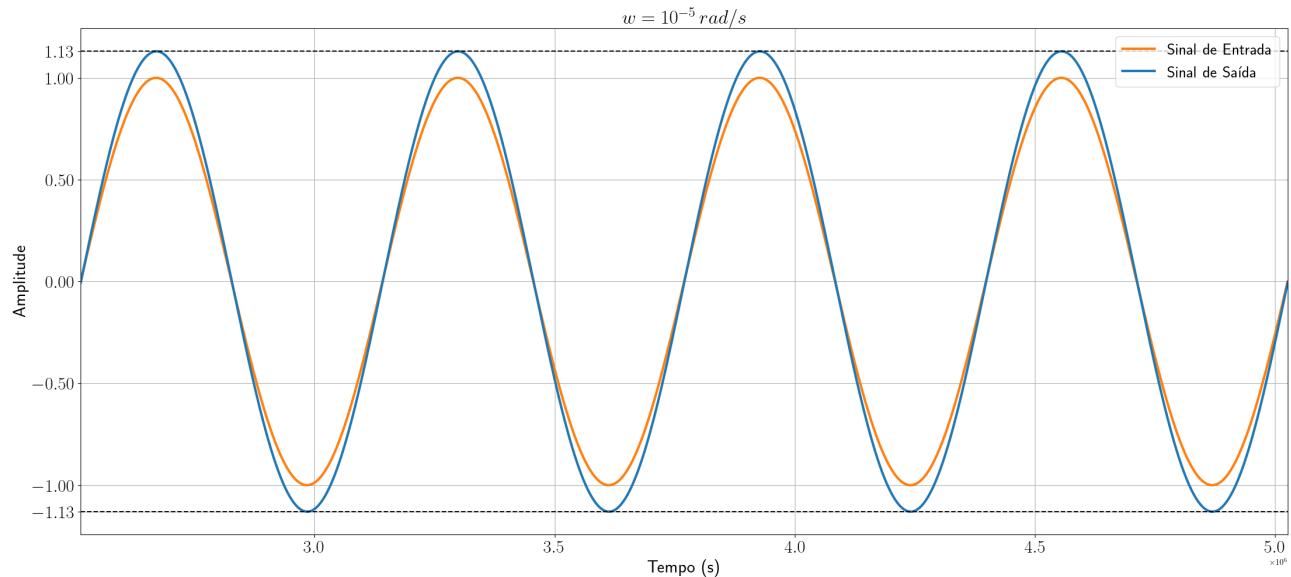


Figura 26: Entrada e saída do sistema com atraso para  $1 \times 10^{-5} \text{ rad s}^{-1}$ .

Fonte: Autoria Própria.

Para a frequência  $w_5$ , observa-se uma ganho próximo a  $-10 \text{ dB}$  e uma defasagem alta, próxima a  $123,4^\circ$ . Sendo assim, espera-se que a resposta do sistema para uma entrada senoidal de amplitude  $A = 1$  e  $\omega = w_5$  tenha uma amplitude de  $1/4$  da de entrada e uma defasagem alta. A Figura 27 demonstra os gráficos de entrada e saída do sistema para  $\omega = w_4$ .

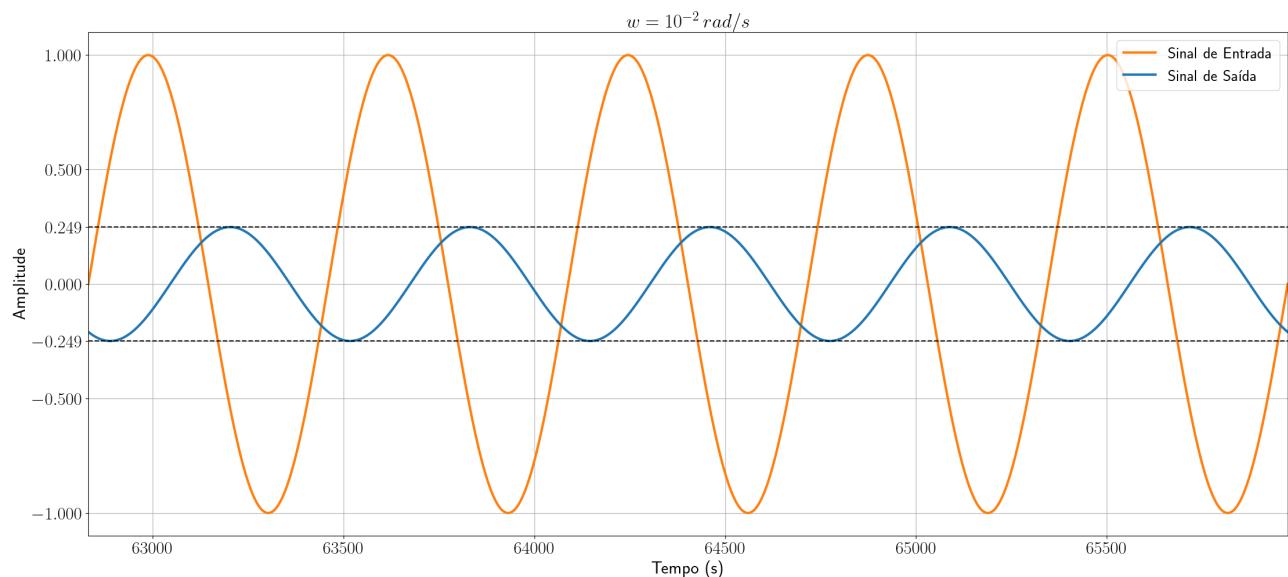


Figura 27: Entrada e saída do sistema com atraso para  $1 \times 10^{-2} \text{ rad s}^{-1}$ .

Fonte: Autoria Própria.

Para a frequência  $w_6$ , observa-se uma ganho próximo a  $-71,4 \text{ dB}$  e uma alta defasagem, próxima a  $-270^\circ$ . Sendo assim, espera-se que a resposta do sistema para uma entrada senoidal de amplitude  $A = 1$  e  $\omega = w_6$  tenha uma amplitude muito atenuada a de entrada e uma defasagem de  $270^\circ$ . A Figura 28 demonstra os gráficos de entrada e saída do sistema para  $\omega = w_6$ .

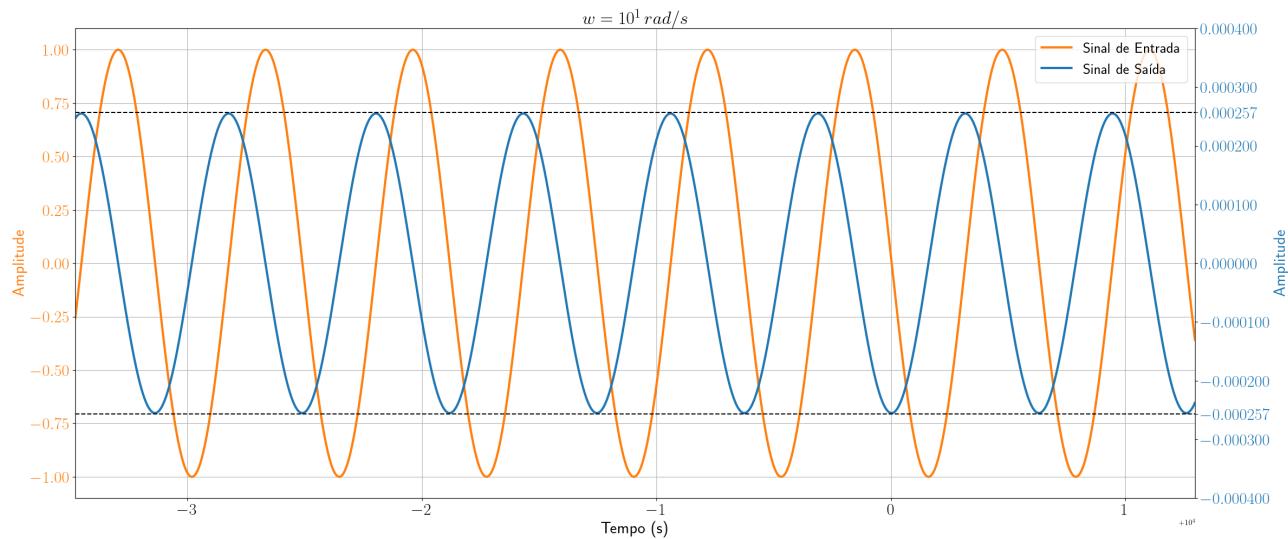


Figura 28: Entrada e saída do sistema com atraso para  $1 \times 10^1 \text{ rad s}^{-1}$ .

Fonte: Autoria Própria.

Para realizar o plot das figuras Figura 26, Figura 27 e Figura 28 foi utilizado o código, em Python, exposto em Código 16.

## 8 Controladores

O controlador é uma parte fundamental na engenharia de controle, pois ele garante que o sistema atue da maneira desejada, especificada por cada projeto. Nessa seção serão apresentados controladores a fim de melhorar a resposta do nosso forno para solda SMD.

### 8.1 Controlador Proporcional

A seguir, será feito o controle proporcional do sistema utilizando o **Método do Lugar das Raízes** (*root locus*) e através do método da resposta em frequência, nos quais deseja-se obter, respectivamente, um coeficiente de amortecimento  $\zeta = 0,5$  e margem de fase  $\Phi_m \geq 50^\circ$ .

Para o Método do Lugar das Raízes será utilizado o diagrama de blocos exposto na Figura 29, em que foi utilizada a função da transferência da planta com aproximação de padé exposta na Equação 22.

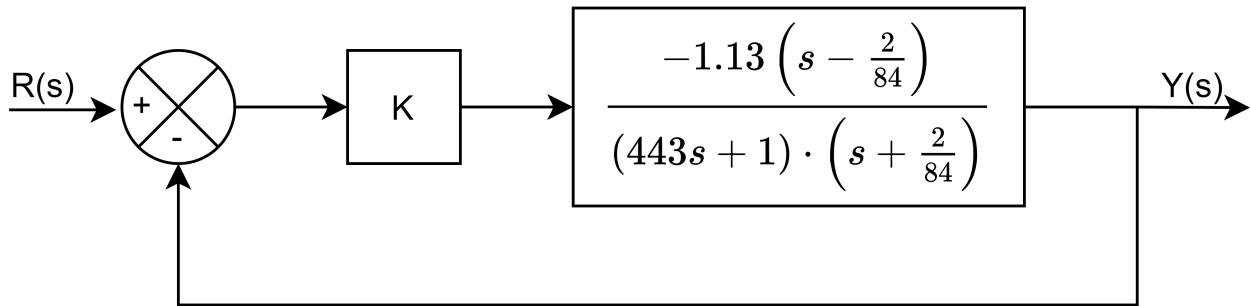


Figura 29: Diagrama de blocos com o controlador proporcional.

Fonte: Autoria Própria.

Através do diagrama de blocos exposto na Figura 29 é possível obter a função de transferência de malha fechada para o sistema com controlador proporcional e a equação característica, a qual será analisada para realizar o gráfico do lugar das raízes. Essas equações podem ser vistas em Equação 47 e Equação 48 respectivamente, em que  $G(s)$  é a função de transferência da planta com a Aproximação de Padé.

$$\frac{Y(s)}{R(s)} = \frac{K \cdot G(s)}{1 + K \cdot G(s)} = K \cdot \frac{-1,13s + 0,0269}{443s^2 + (11,55 - 1,13k)s + (0,02384 + 0,0269k)} \quad (47)$$

$$1 + K \cdot G(s) = 0 \quad (48)$$

Para fazer o gráfico do lugar das raízes é possível partir da Equação 48 e, através da análise de igualdade de uma função complexa, em que o módulo e fase dos dois lados da equação devem ser iguais, verificar o comportamento dos polos para todos os valores de  $K$ , ou podemos fazer uso de métodos, *softwares* computacionais ou linguagens de programação que facilitam nesse desenvolvimento. Nesse relatório foi utilizado o último caso.

Para realizar o *root locus* foi utilizado a função **rlocus** da biblioteca **control** do Python. O gráfico do *root locus* pode ser visto na Figura 30 e o código utilizado para gerar esse plot em Código 17.

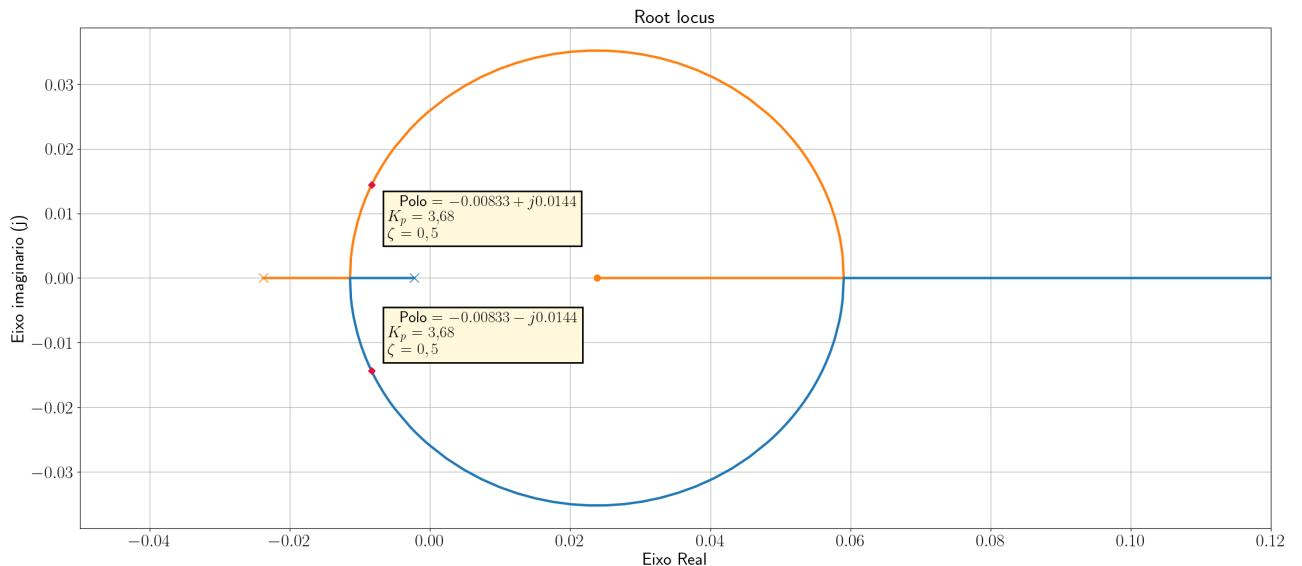


Figura 30: Rootlocus para o controlador proporcional.

Fonte: Autoria Própria.

Através do Figura 30, é possível encontrar  $K$  tal que  $\zeta = 0,5$ , conforme expressa na Equação 49.

$$K = 3,68 \quad (49)$$

As figuras Figura 24 e Figura 25 demonstram o Diagrama de Bode para o sistema com atraso, conforme demonstrado na equação Equação 22. A margem de ganho para esse sistema é de  $\Phi_m = 145,49^\circ$ , expressa na Equação 46. Para que  $\Phi_m = 50^\circ$  basta projetar um  $K$  tal que sua contribuição leve o gráfico de módulo a cruzar o eixo quando a fase  $\Phi = 130^\circ$ . Para isso, inicialmente, iremos encontrar a frequência  $\omega_i$  para  $\angle G(j\omega_i) = 130^\circ$ , conforme demonstrado nas equações Equação 50 e Equação 51.

$$\angle G(j\omega_i) = -2 \arctan(42\omega_i) - \arctan(443\omega_i) \quad (50)$$

$$\omega_i = 1,14 \times 10^{-2} \text{ rad s}^{-1} \quad (51)$$

Após encontrado  $\omega_i$ , foi encontrado o  $|G(j\omega_i)|_{dB}$ , conforme demonstrado nas equações Equação 52 e Equação 53.

$$|G(j\omega_i)|_{dB} = 1,062 + 10 \log(42^2 \omega_i^2 + 1) \quad (52)$$

$$|G(j\omega_i)|_{dB} = -13,1712 \text{ dB} \quad (53)$$

Sendo assim, é preciso um ganho  $K$  tal que sua contribuição no diagrama de módulo seja o positivo do ganho demonstrado em Equação 53, conforme exposto nas equações Equação 54 e Equação 55.

$$|G(j\omega_i)|_{dB} = 20 \log(K) \quad (54)$$

$$K = 4.55575 \quad (55)$$

As figuras Figura 31 e Figura 32 demonstram o diagrama de bode do sistema com o controlador proporcional. O código para o plot dos graficos está exposto em Código 18.

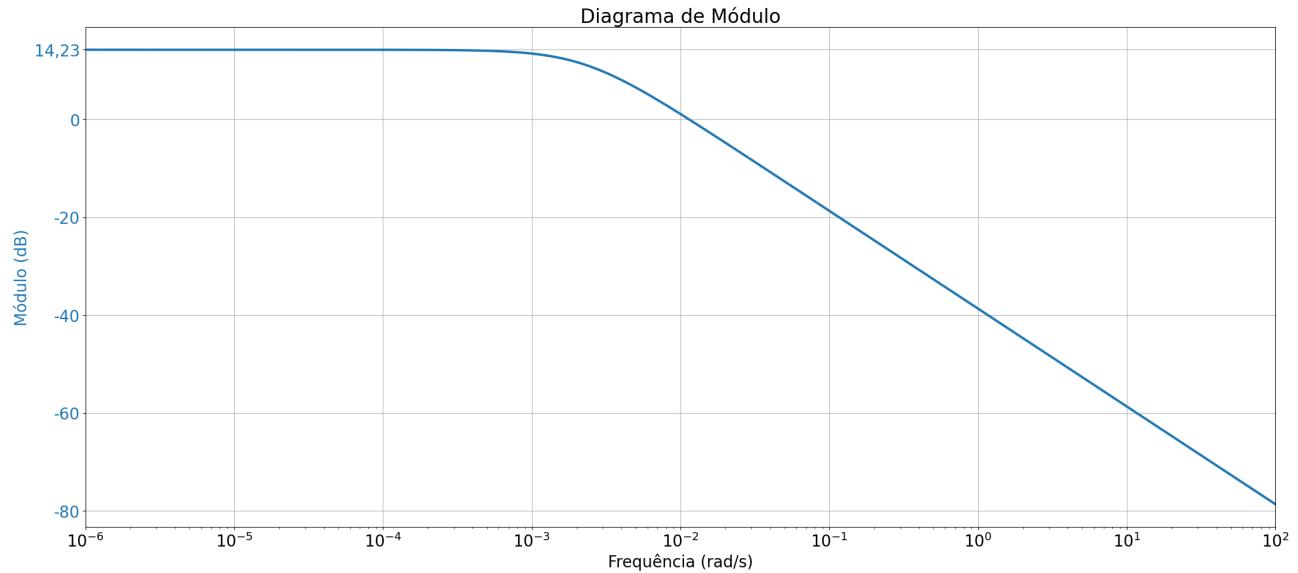


Figura 31: Diagrama de Módulo com o controlador proporcional.

Fonte: Autoria Própria.

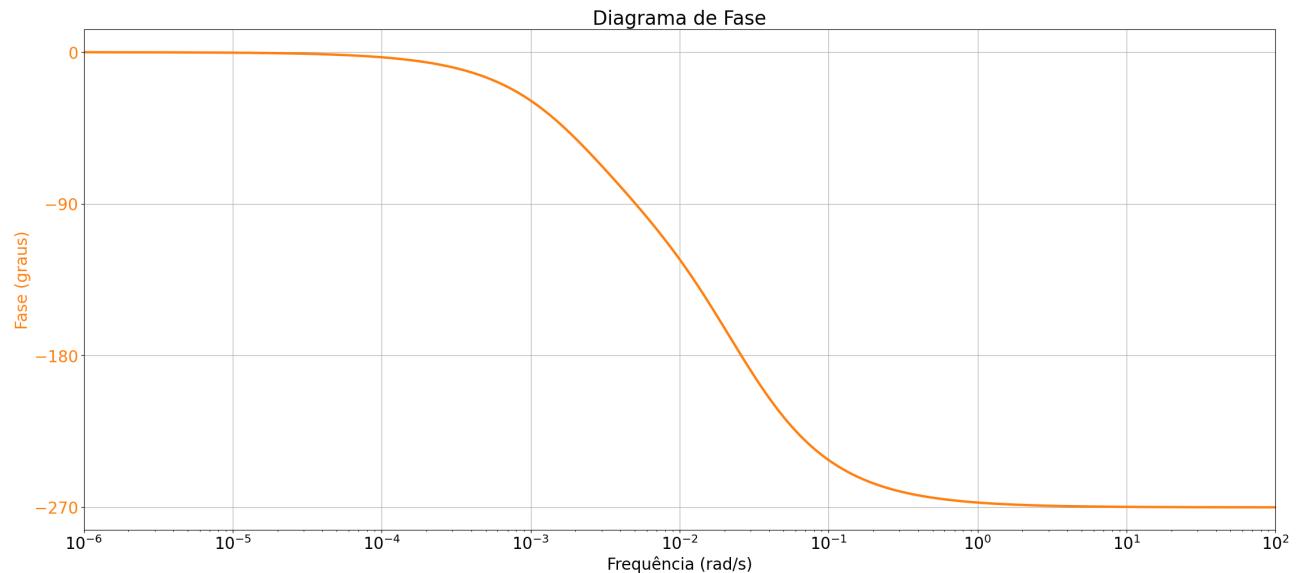


Figura 32: Diagrama de Fase com o controlador proporcional.

Fonte: Autoria Própria.

Com os valores de ganho obtidos nas equações Equação 49 e Equação 55 foram esboçados os diagramas de bloco para o sistema com controlador proporcional obtido pelo *root locus* e pela análise na frequência, conforme demonstrado na Figura 33.

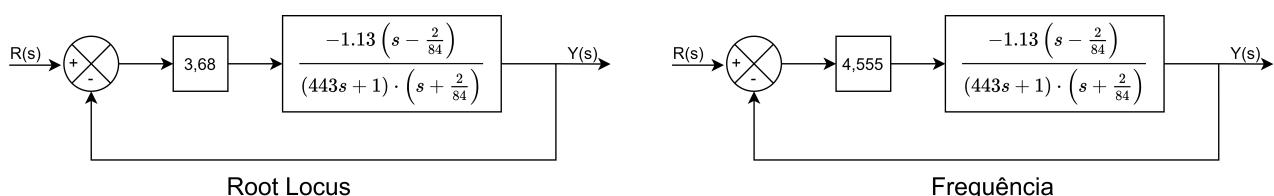


Figura 33: Diagrama de blocos com os controladores proporcional.

Fonte: Autoria Própria.

Através dos diagramas expostos na Figura 33, é possível obter a função de malha fechada do sistema para cada caso, exposta em Equação 56.

$$\underbrace{\frac{Y(s)}{R(s)} = \frac{-4.158s + 0.09901}{443s^2 + 7.389s + 0.1228}}_{\text{Root Locus}} \quad \underbrace{\frac{Y(s)}{R(s)} = \frac{-5.148s + 0.1226}{443s^2 + 6.4s + 0.1464}}_{\text{Frequência}} \quad (56)$$

A Figura 34 demonstra a resposta ao degrau unitário dos sistemas em malha fechada demonstrados na Equação 56. Para realizar o plot das Figura 34 foi utilizado o código, em Python, exposto em Código 19.

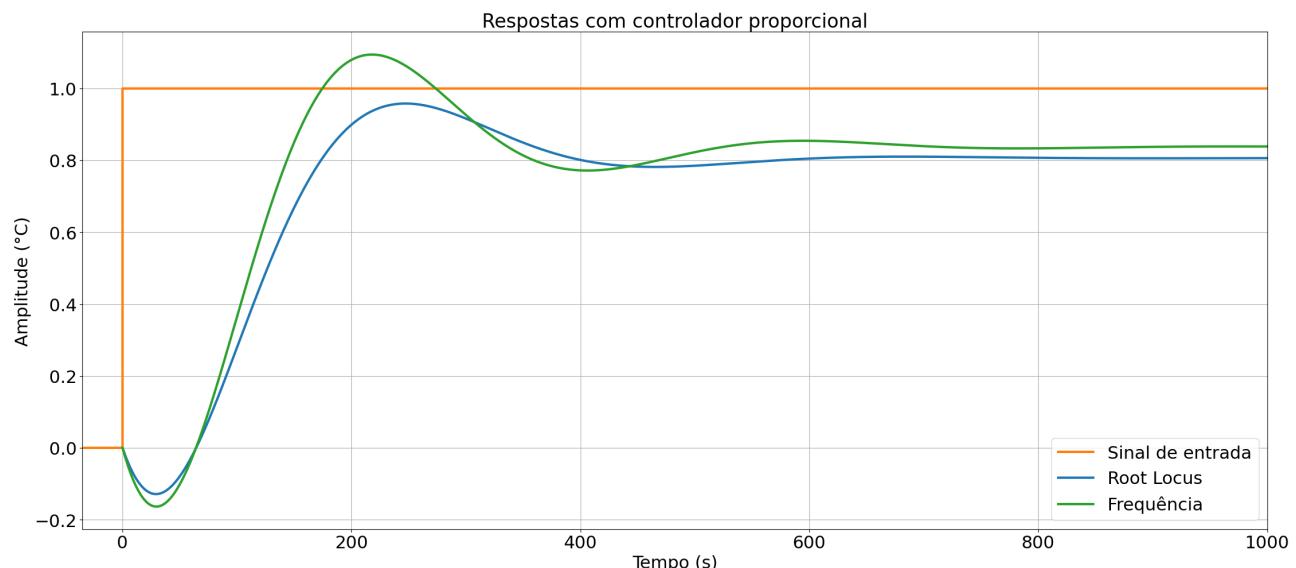


Figura 34: Respostas em malha fechada com o controlador proporcional.

Fonte: Autoria Própria.

Para comprovar a precisão da aproximação de padé de primeira ordem utilizada, foi feito o plot da resposta do sistema com controlador proporcional utilizando a aproximação de padé e com o atraso puro de tempo, conforme demonstrado na Figura 35. É possível notar que a aproximação se comporta de maneira similar ao gráfico real tanto no tempo transitório quanto no regime permanente, sendo assim considerada uma boa estimativa. Para a Figura 35 foi utilizado o *simulink* para obtenção dos dados e o Código 20, em Python, para o plot do gráfico.

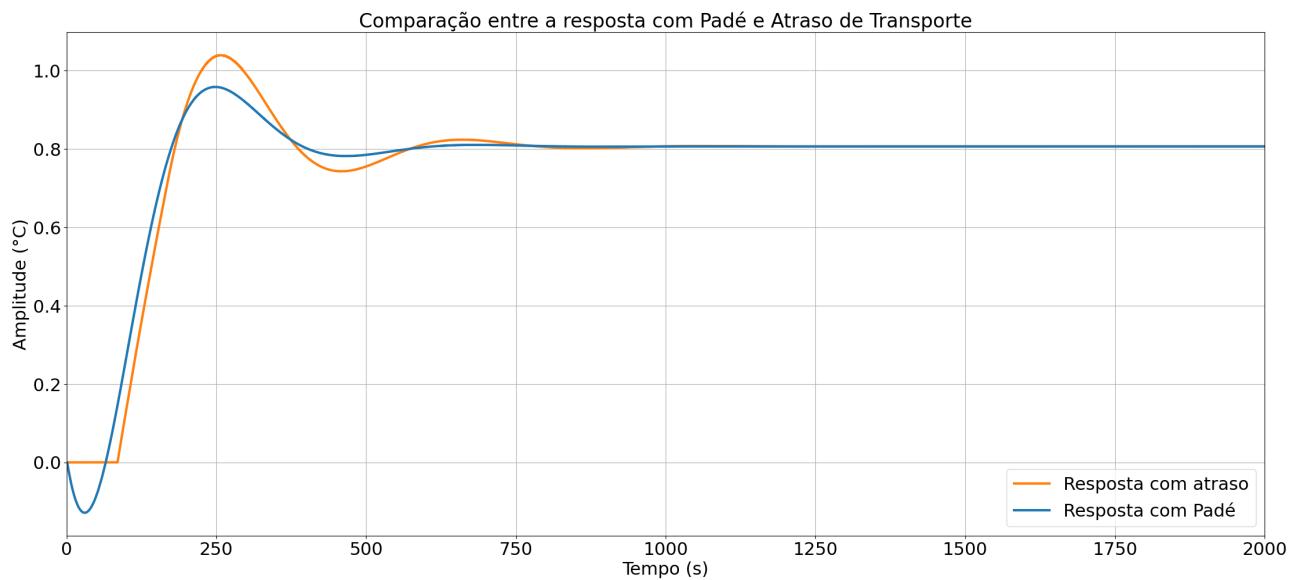


Figura 35: Comparação entre as respostas com Padé e Atraso Puro utilizando controlador proporcional.

Fonte: Autoria Própria.

### 8.1.1 Análise de Estabilidade

Um passo importante no projeto de controladores é a análise de estabilidade, uma vez que o uso de um controlador pode levar o sistema à instabilidade. Foi feita a análise de estabilidade para o sistema com controlador proporcional utilizando três métodos:

- Critério de Routh
- Diagrama de Nyquist
- Lugar das Raízes

#### 8.1.1.1 Critério de Routh

A Figura 36 demonstra o diagrama de blocos para o circuito com o controlador proporcional

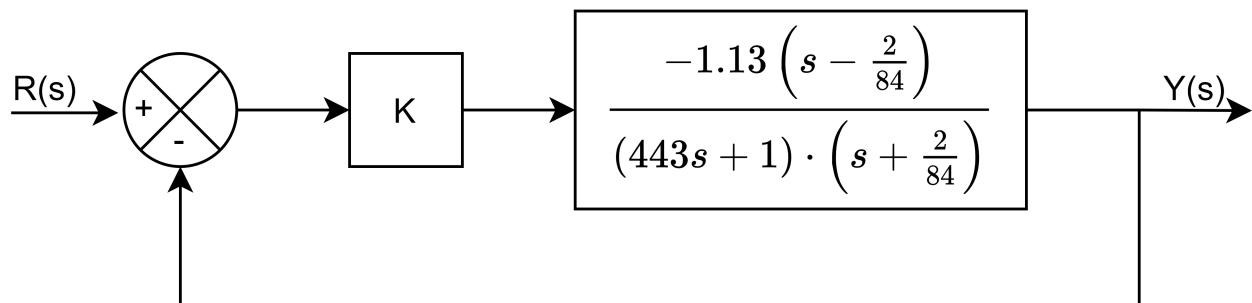


Figura 36: Diagrama de blocos para o Critério de Routh com controlador proporcional.

Fonte: Autoria Própria.

Através do diagrama de blocos da Figura 36, foi encontrada a função de transferência de malha fechada exposta em Equação 57.

$$\frac{Y(s)}{R(s)} = \frac{-1,13ks + 0,0269k}{443s^2 + (11,55 - 1,13k)s + (0,02384 + 0,0269k)} \quad (57)$$

Com a função de transferência exposta em Equação 57 foi possível montar a tabela para o Critério de Routh, conforme demonstrada na Tabela 1.

$s^2$	443	$0,02384 + 0,0269k$
$s^1$	$11,55 - 1,13k$	0
$s^0$	$\left( \frac{(11,55 - 1,13k) \cdot (0,02384 + 0,0269k)}{(11,55 - 1,13k)} \right)$	

Tabela 1: Tabela para o Critério de Routh utilizando controlador proporcional.

Sabendo que para que o sistema seja estável não pode haver mudança de sinal nos elementos da primeira coluna, é necessário que o segundo e terceiro termo também seja positivo. Sendo assim, chegamos na Equação 58.

$$\begin{cases} 11,55 - 1,13k & > 0 \\ 0,02384 + 0,0269k & > 0 \end{cases} \quad (58)$$

Resolvendo o sistema de inequações demonstrado em Equação 58, chegamos em Equação 59 e, finalmente, encontramos o valor máximo para estabilidade do ganho K em Equação 60.

$$-0,8863 < K < 10,2212 \quad (59)$$

$$0 < K < 10,2212 \quad (60)$$

### 8.1.1.2 Diagrama de Nyquist

O Critério de Nyquist relaciona a estabilidade de um sistema em malha fechada com a resposta em frequência em malha aberta e a posição dos polos de malha aberta [Nise, 2017]. O Diagrama de Nyquist comprehende um mapeamento de todo o semiplano direito do plano s a procura de zeros e polos e, através disso, traça um gráfico no plano GH, em que o número de envoltórias no -1 compreende uma relação entre o número de polos e zeros no semiplano direito, conforme demonstrado na equação Equação 61.

$$N = Z - P \quad (61)$$

Em que [Aguirre, 2016]:

- **N** é o número de envoltórias no -1
- **Z** é o número de polos de malha fechada envoltos por  $\Gamma$ .
- **P** é o número de polos de malha aberta envoltos por  $\Gamma$  (é conhecido a priori).

Sabendo que  $P$  é o número de polos de malha aberta envoltos no mapeamento de todo o semiplano direito, através da Figura 9 é possível obter  $P = 0$ . A Figura 37 representa o Diagrama de Nyquist para o sistema com a Aproximação de Padé. Para a Figura 37 foi utilizado o *matlab* para obtenção dos dados e o Código 21, em Python, para o plot do gráfico.

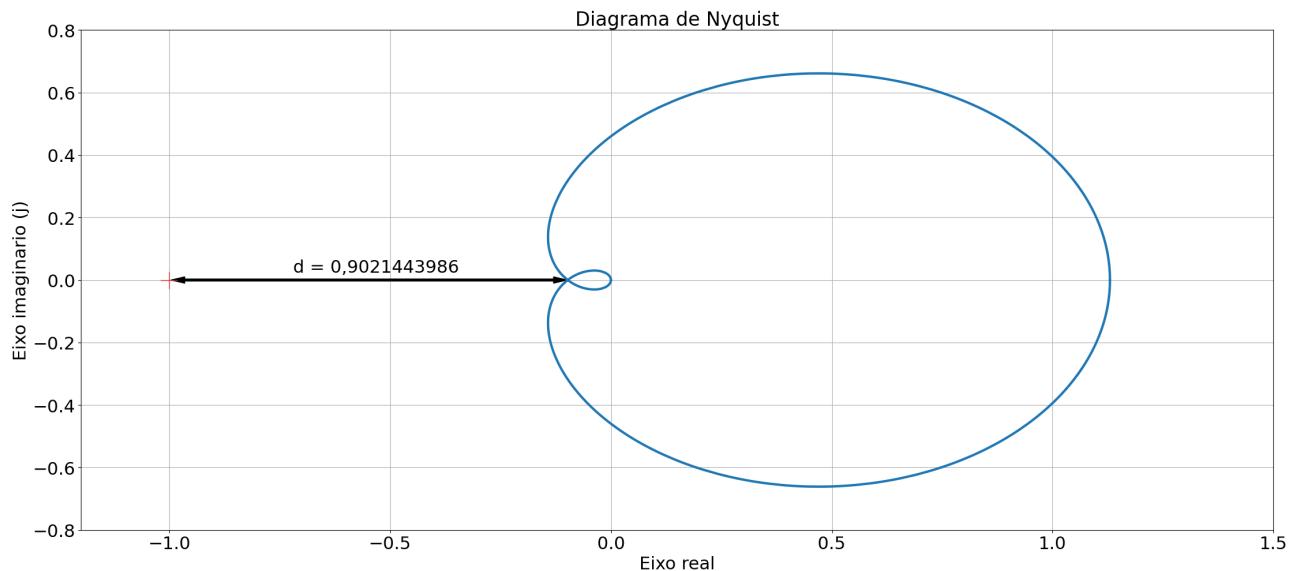


Figura 37: Diagrama de Nyquist para o sistema com controlador proporcional.

Fonte: Autoria Própria.

O gráfico da Figura 37 demonstra que não há nenhuma envoltória em  $-1$ , porém é sabido que o incremento de um ganho  $K$  age como um fator de escala para esse gráfico, amplificando-o de tal forma que o faça envolver duas vezes o  $-1$ . Nesse caso, aplicando o valor de  $N = 2$ , temos que o número de polos de malha fechada no semiplano direito é 2, levando o sistema a instabilidade. Esse caso é comprovado nas equações Equação 62.

$$\begin{aligned} 2 &= Z - 0 \\ Z &= 2 \end{aligned} \tag{62}$$

O ganho  $K$  mínimo para esse caso foi encontrado através da Equação 63, em que  $d$  é a distância entre o ponto  $-1 + j0$  e o ponto em que o Diagrama de Nyquist cruza o eixo real, conforme demonstrado na Figura 37. Sendo assim, para manter o sistema estável,  $k$  deve seguir o que foi demonstrado na Equação 64.

$$1 = (1 - d) \cdot k \tag{63}$$

$$K < 10,2191 \tag{64}$$

### 8.1.1.3 Lugar das Raízes

Uma outra forma de fazer a análise de estabilidade é através do Lugar das Raízes, em que o ponto de cruzamento do gráfico no eixo  $y$  nos da o ganho máximo do sistema, que é quando o sistema se torna marginalmente estável. Nesse ponto, temos que  $\zeta = 0$ .

Para essa análise, foi traçado o *root locus* do sistema, conforme demonstrado na Figura 38. O código, feito em Python, para gerar essa figura está exposto em Código 22.

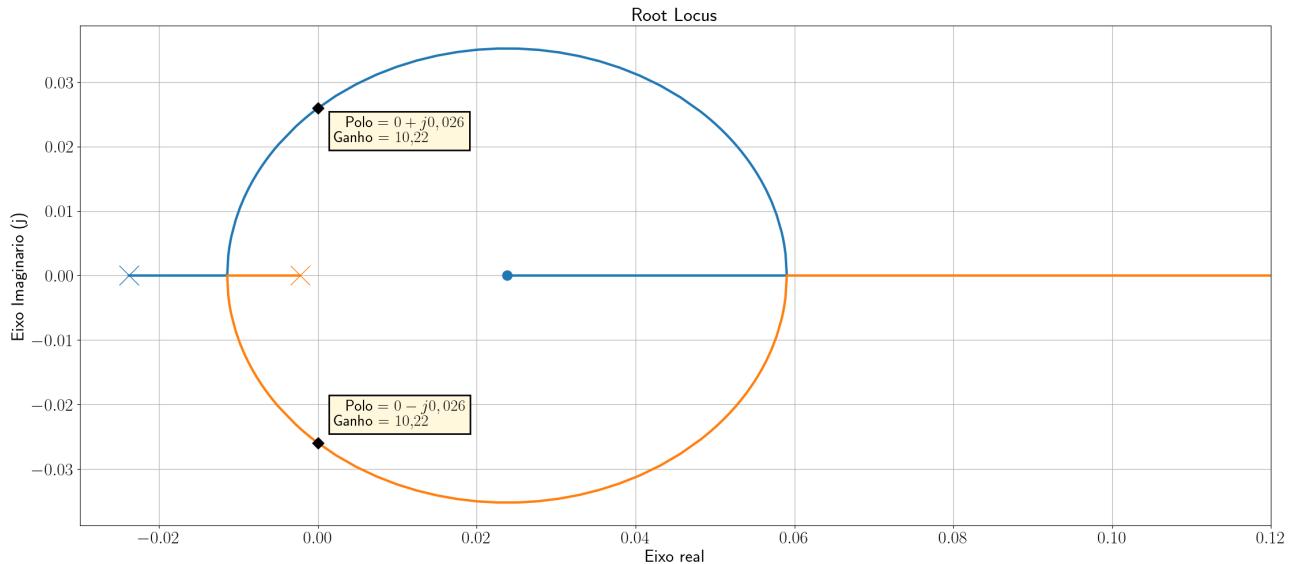


Figura 38: Root Locus para estabilidade do controlador proporcional.

Fonte: Autoria Própria.

Através do Lugar das Raízes demonstrado no gráfico da Figura 38, é possível encontrar os valores para o ganho  $K$  que não levam o sistema à instabilidade, demonstrados em Equação 65.

$$K \lesssim 10,22 \quad (65)$$

\*\*\*

Nessa seção, foram feitas três análises de estabilidades diferentes, através do Critério de Routh, Diagrama de Nyquist e Lugar das Raízes, sendo o resultado obtido por todos bem semelhantes e próximos a  $K < 10,22$ . Para validar esse valor, iremos utilizar  $K = 10,22$  e esperamos ver um sistema marginalmente estável, ou seja, que oscila indefinidamente sem amortecimento. Para isso, iremos utilizar o sistema com o atraso simulado pela Aproximação de Padé e outro com atraso puro no tempo, conforme visto na Figura 39.

Na Figura 39 é possível observar que o sistema com Aproximação de padé - exposto em laranja - se comporta de maneira esperada, oscilando indefinidamente sem amortecimento, porém, a curva para o Atraso de Transporte - exposta em azul - já demonstra a instabilidade, apresentando oscilação e crescimento indevido do seu valor de amplitude. Através de alguns testes heurísticos, foi percebido que a instabilidade para o sistema com atraso puro no tempo começa com  $k \approx 8$ .

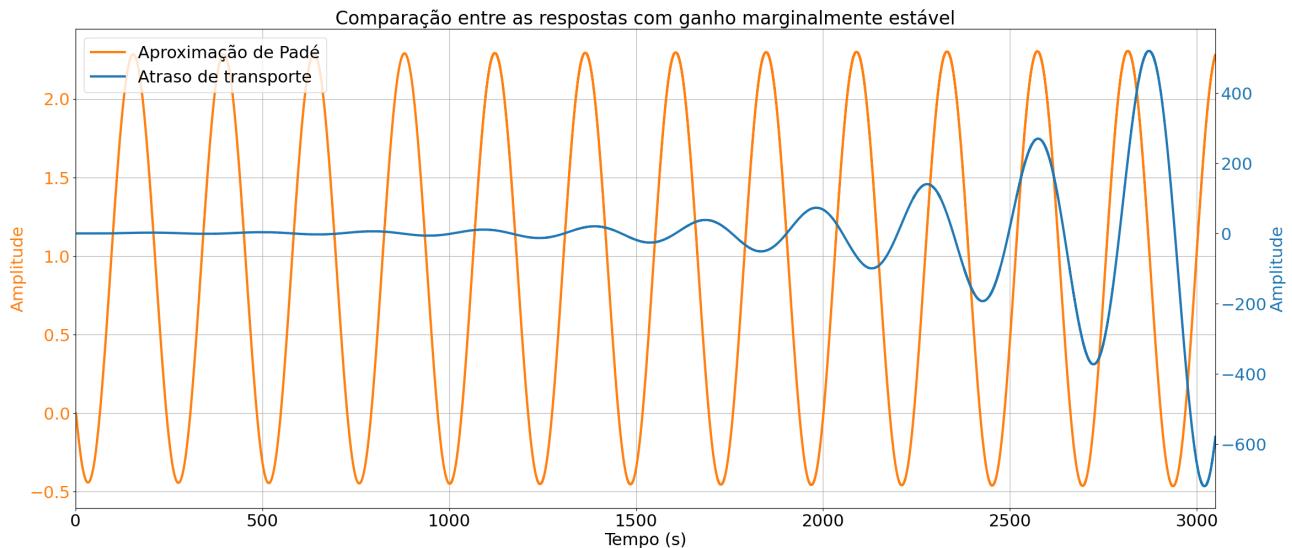


Figura 39: Comparaçāo entre as respostas com ganho marginalmente estável.

Fonte: Autoria Própria.

Para entender melhor esse problema, foi feita a análise do Lugar das Raízes utilizando uma aproximação de padé de decima ordem, conforme demonstrada na Equação 66.

$$e_{10}^{-84s} \approx \frac{s^{10} - \frac{55s^9}{42} + \frac{165s^8}{196} - \frac{715s^7}{2058} + \frac{715s^6}{7056} - \frac{715s^5}{32928} + \frac{993435986107611s^4}{288230376151711744}}{s^{10} + \frac{55s^9}{42} + \frac{165s^8}{196} + \frac{715s^7}{2058} + \frac{715s^6}{7056} + \frac{715s^5}{32928} + \frac{993435986107611s^4}{288230376151711744}} + \dots$$

$$- \frac{3676388955391431s^3}{9223372036854775808} + \frac{4726785799788983s^2}{147573952589676412928} + \dots$$

$$+ \frac{3676388955391431s^3}{9223372036854775808} + \frac{4726785799788983s^2}{147573952589676412928}$$

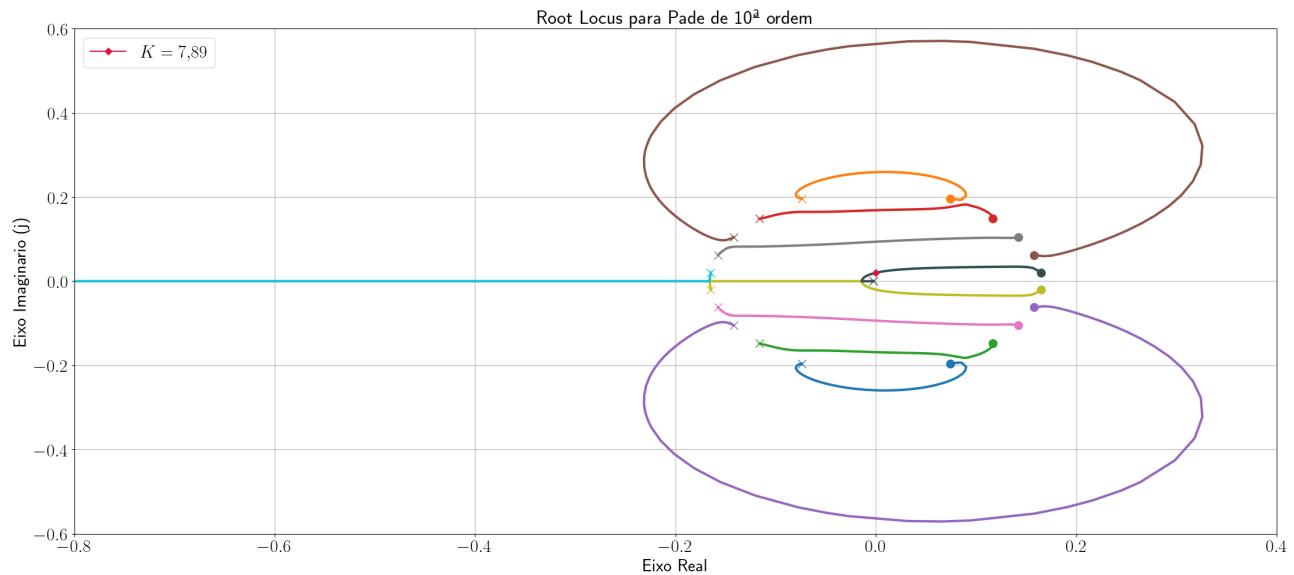
$$- \frac{950359049693023s}{590295810358705651712} + \frac{1448166170960797}{37778931862957161709568}$$

$$+ \frac{950359049693023s}{590295810358705651712} + \frac{1448166170960797}{37778931862957161709568}$$
(66)

Fazendo uma análise pelo Lugar das Raízes utilizando a Aproximação de Padé de decima ordem, foi constatado um ganho máximo para estabilidade de  $K = 7,89$ , concordante com o valor que observado anteriormente. A Figura 40 representa o *Root Locus* do sistema utilizando a aproximação de Padé de decima ordem. Para a Figura 40 foi utilizado o *matlab* para obtenção dos dados e o Código 23, em Python, para o plot do gráfico.

Sendo assim, apesar da Aproximação de Padé de primeira ordem oferecer um bom indicativo da resposta do sistema, ela falha no quesito de análise da estabilidade do sistema, uma vez que forneceu um  $K$  com erro relativo de  $e = 29.53\%$ .

Um outro método para a análise de estabilidade de sistemas com atraso de transporte é através do Critério de Lyapunov-Krasovskii, que poderá mais para frente ser demonstrada nesse relatório, caso necessário.

Figura 40: Root Locus para Padé de 10<sup>a</sup> ordem.

Fonte: Autoria Própria.

## 8.2 Controlador Integral

O controlador do tipo integral é capaz de eliminar o erro em regime permanente do sistema através da adição de um polo de malha aberta na origem, porém esse tipo de controlador prejudica a estabilidade do sistema e o seu comportamento no regime transitório, pois tende a modificar drasticamente e forçar a trajetória do lugar das raízes para o semiplano direito [Henriques, 2020b]. Nessa seção será projetado um controlador do tipo integral que force o sistema a ter um coeficiente de amortecimento  $\zeta = 0,5$  através do lugar das raízes e outro controlador que leve o sistema a ter uma margem de fase  $\Phi_m \geq 50^\circ$  através da análise na frequência. A Figura 41 demonstra o diagrama de blocos da planta com um controlador integral.

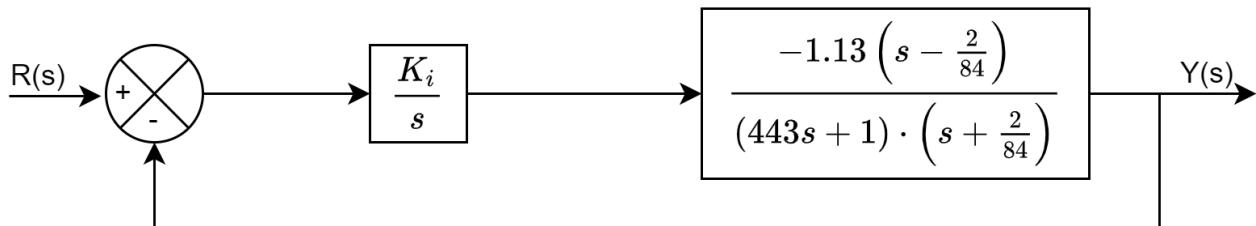


Figura 41: Diagrama de blocos com o controlador integral.

Fonte: Autoria Própria.

Resolvendo a malha de realimentação, foi encontrada a função de transferência da planta com o controlador integral, demonstrada na Equação 67.

$$\frac{Y(s)}{R(s)} = \frac{-1,13K_i s + 0,0269K_i}{443s^2 + 11,55s^2 + (-1,13K_i + 0,0238)s + 0,0269K_i} \quad (67)$$

Na Subsubseção 8.2.1 será feita a análise de estabilidade do sistema com o controlador integral.

### 8.2.1 Análise de Estabilidade

Um passo importante no projeto de controladores é a análise de estabilidade, uma vez que o uso de um controlador pode levar o sistema a instabilidade. Foi feita a análise de estabilidade para o sistema com controlador integral utilizando dois métodos:

- Critério de Routh
- Diagrama de Nyquist

#### 8.2.1.1 Critério de Routh

Através do diagrama de blocos exposto na Figura 41 e da função de transferência em Equação 67, foi possível montar a tabela para o Critério de Routh, conforme demonstrado na Tabela 2.

$s^3$	443	$-1,13K_i + 0,0238$
$s^2$	11,55	$0,0269K_i$
$s^1$	$-2,161K_i + 0,0238$	0
$s^0$	0,0269 $K_i$	

Tabela 2: Tabela para o Critério de Routh utilizando controlador integral.

Sabendo que para que o sistema seja estável não pode haver mudança de sinal nos elementos da primeira coluna, é necessário que o terceiro e quarto termo também seja positivo. Sendo assim, chegamos na Equação 68.

$$\begin{cases} -2,161K_i + 0,0238 > 0 \\ 0,0269K_i > 0 \end{cases} \quad (68)$$

Resolvendo o sistema de inequações demonstrado em Equação 68, encontramos os valores mínimo e máximo para estabilidade de  $K_i$  em Equação 69.

$$0 < K_i < 0,011 \quad (69)$$

#### 8.2.1.2 Diagrama de Nyquist

O Critério de Nyquist relaciona a estabilidade de um sistema em malha fechada com a resposta em frequência em malha aberta e a posição dos polos de malha aberta [Nise, 2017]. O Diagrama de Nyquist compreende um mapeamento de todo o semiplano direito do plano s a procura de zeros e polos e, através disso, traça um gráfico no plano GH, em que o número de envoltórias no -1 compreende uma relação entre o número de polos e zeros no semiplano direito, conforme demonstrado na equação Equação 70.

$$N = Z - P \quad (70)$$

Em que [Aguirre, 2016]:

- $N$  é o número de envoltórias no -1
- $Z$  é o número de polos de malha fechada envoltos por  $\Gamma$ .
- $P$  é o número de polos de malha aberta envoltos por  $\Gamma$  (é conhecido a priori).

Sabendo que  $P$  é o número de polos de malha aberta envoltos no mapeamento de todo o semiplano direito, é possível obter  $P = 0$ , uma vez que o mapa de polos e zeros do nosso sistema em malha aberta é igual ao da Figura 9 com a adição de um polo na origem. A Figura 42 representa o Diagrama de Nyquist para o sistema com atraso de transporte. Para a Figura 42 foi utilizado o *software* MATLAB para obtenção dos dados e, para o plot do gráfico, o Código 24, em Python.

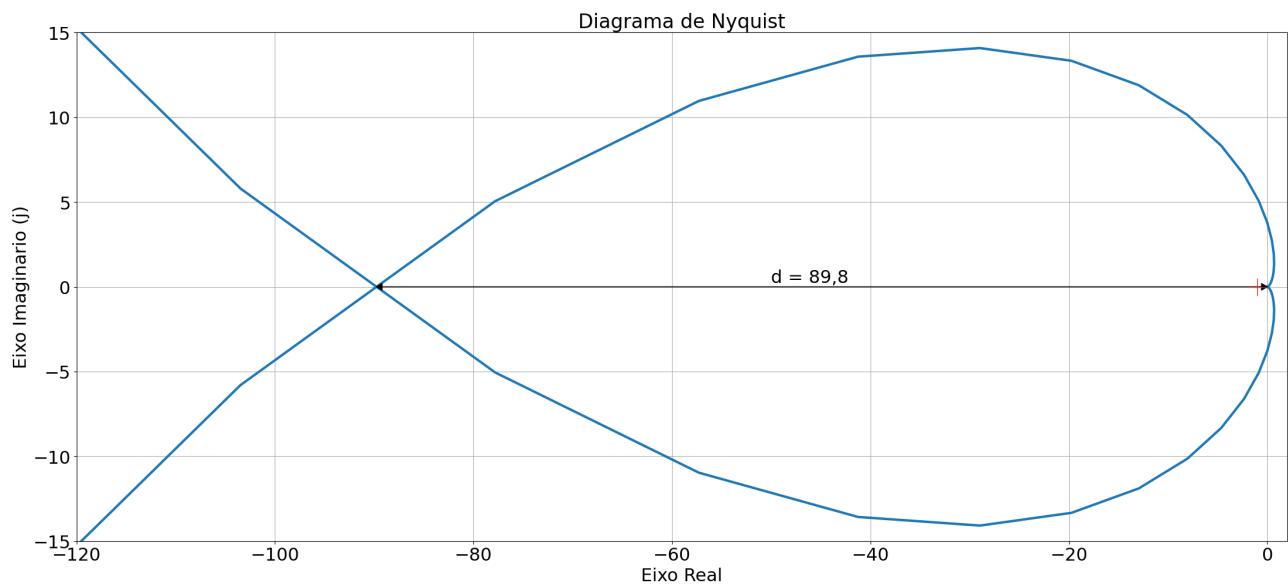


Figura 42: Diagrama de Nyquist para o sistema com controlador integral.

Fonte: Autoria Própria.

O gráfico da Figura 42 demonstra que há envoltórias em -1, o que significa que com  $K_i = 1$  o nosso sistema já é instável. Sendo assim, será reduzido  $K_i$  de tal forma que o Diagrama de Nyquist não envolva -1, ja que o decremento de  $K_i$  age como um fator de escala para esse gráfico, reduzindo-o de tal forma que não leve o sistema a instabilidade. Nesse caso, aplicando o valor de  $N = 0$ , temos que o número de polos de malha fechada no semiplano direito é 0, levando o sistema a estabilidade. Esse caso é comprovado nas equações Equação 71.

$$\begin{aligned} 0 &= Z - 0 \\ Z &= 0 \end{aligned} \tag{71}$$

O ganho  $K_i$  máximo para esse caso foi encontrado através da Equação 72, em que  $d$  é a distância entre a origem e o ponto em que o Diagrama de Nyquist cruza o eixo real, conforme demonstrado na Figura 42. Sendo assim, para manter o sistema estável,  $K_i$  deve seguir o que foi demonstrado na Equação 73.

$$1 = d \cdot K_i \tag{72}$$

$$K_i < 0,011 \quad (73)$$

\*\*\*

Para o projeto do controlador com  $\zeta = 0,5$  foi realizado o root locus através da função **rlocus** da biblioteca **control** do Python. O gráfico do root locus pode ser visto na Figura 43 e o código utilizado para gerar esse plot em Código 25.

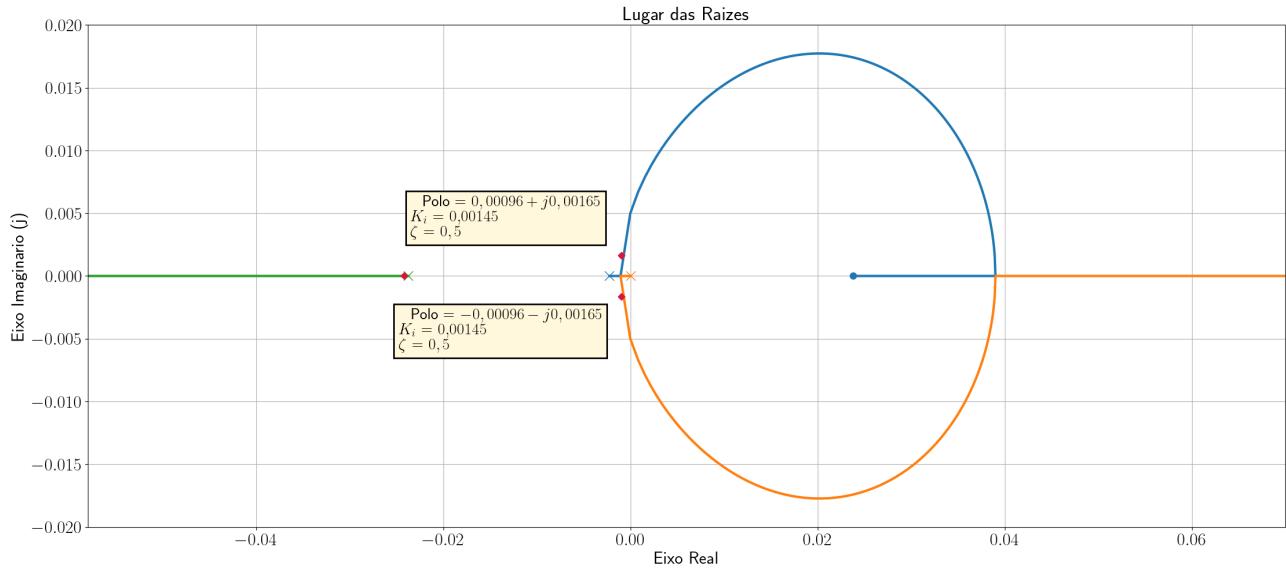


Figura 43: Root locus para o controlador integrador.

Fonte: Autoria Própria.

Através da Figura 43 é possível obter o valor de  $K_i$  para que o coeficiente de amortecimento  $\zeta = 0,5$ . A Equação 74 demonstra esse  $K_i$ .

$$K_i = 0,00145 \quad (74)$$

A seguir, será feito o controlador utilizando a análise na frequência. A Figura 44 apresenta o Diagrama de Bode para  $K_i = 1$ .

Através da Figura 44 percebemos que o sistema é instável para esse  $K_i$ , uma vez que apresenta uma margem de fase  $\Phi_m = -127^\circ$ . Sendo assim, precisamos modificar o  $K_i$  de tal forma que desloque o gráfico de fase para a direita a fim de levar o sistema a ter  $\Phi_m > 50^\circ$ .

Para o sistema com integrador, temos os seguintes valores para módulo e fase, expressos pela Equação 75.

$$\begin{aligned} |G(jw)|_{dB} &= 1,062 - 10 \log(443\omega^2 + 1) + 20 \log(K_i) - 20 \log(\omega) \\ \underline{|G(jw)|} &= -\arctan(443\omega) - \arctan(42\omega) - \arctan(42\omega) - 90^\circ \end{aligned} \quad (75)$$

O primeiro passo é achar  $\omega_i$  tal que  $\underline{|G(j\omega_i)|} = -130^\circ$ , demonstrado nas equações Equação 76 e Equação 77.

$$-130^\circ = -\arctan(443\omega) - \arctan(42\omega) - \arctan(42\omega) - 90^\circ \quad (76)$$

$$\omega_i = 1,463 \times 10^{-3} \text{ rad s}^{-1} \quad (77)$$

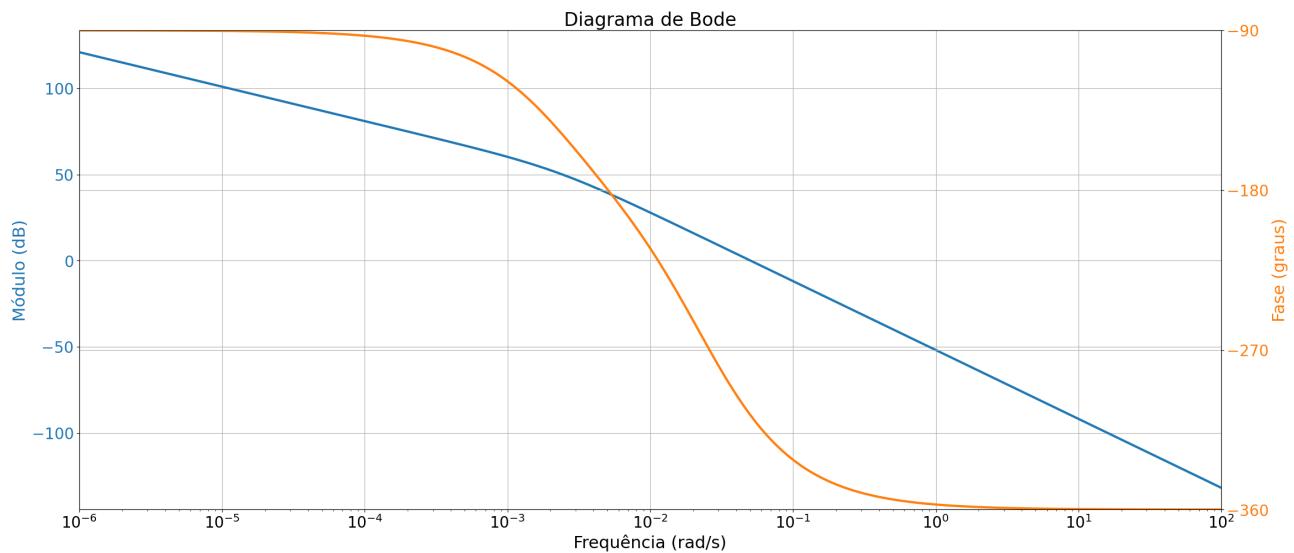


Figura 44: Diagrama de bode inicial com controlador integral.

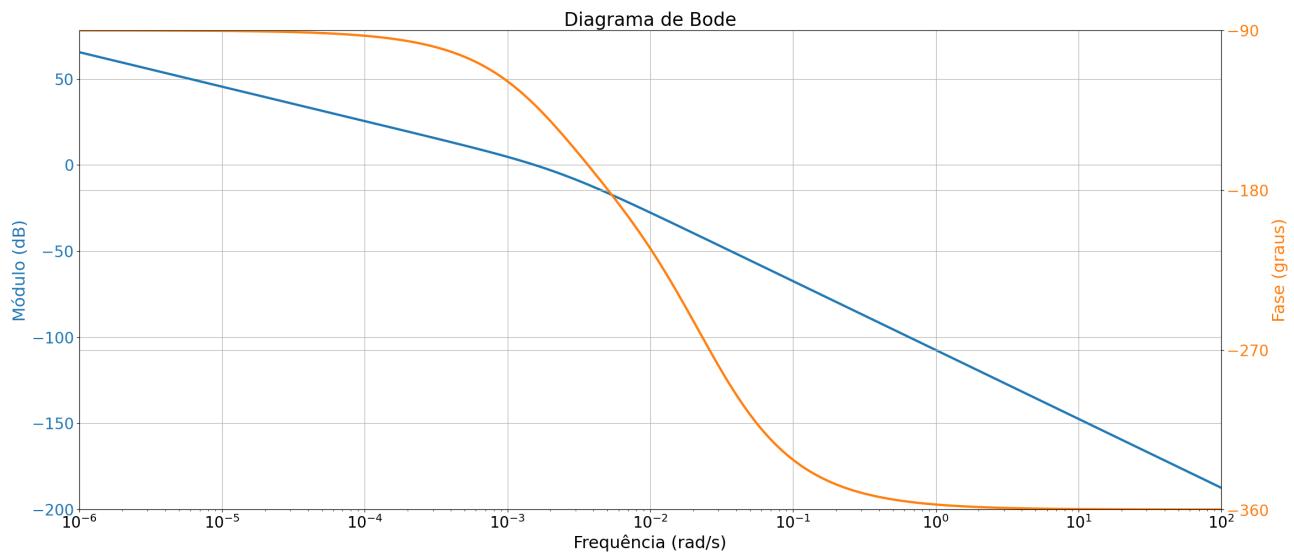
Fonte: Autoria Própria.

Encontrada  $\omega_i$ , iremos achar  $K_i$  tal que  $|G(j\omega_i)|_{dB} = 0$ , conforme a Equação 78 e Equação 79.

$$0 = 1,062 - 10 \log(443^2 \omega_i^2 + 1) + 20 \log(K_i) - 20 \log(\omega_i) \quad (78)$$

$$K_i = 0.00165453 \quad (79)$$

A Figura 45 apresenta o novo Diagrama de Bode para  $K_i = 0.00165453$ . É possível notar uma mudança entre essa figura e a Figura 44, sendo o sistema agora estável. Além disso, a margem de fase agora é a especificada pelo projeto  $\Phi_m = 50^\circ$ .

Figura 45: Diagrama de bode com controlador integral para  $K_i = 0.00165453$ .

Fonte: Autoria Própria.

Para a análise dos resultados foi plotado o gráfico da Figura 46, que demonstra as respostas ao degrau do sistema utilizando a Aproximação de Padé com os valores de  $K_i$  encontrados nas equações Equação 74 e Equação 79 e as respostas utilizando o atraso de transporte com os mesmos  $K_i$ . Nota-se que a Aproximação de Padé foi extremamente precisa, resultando em respostas idênticas.

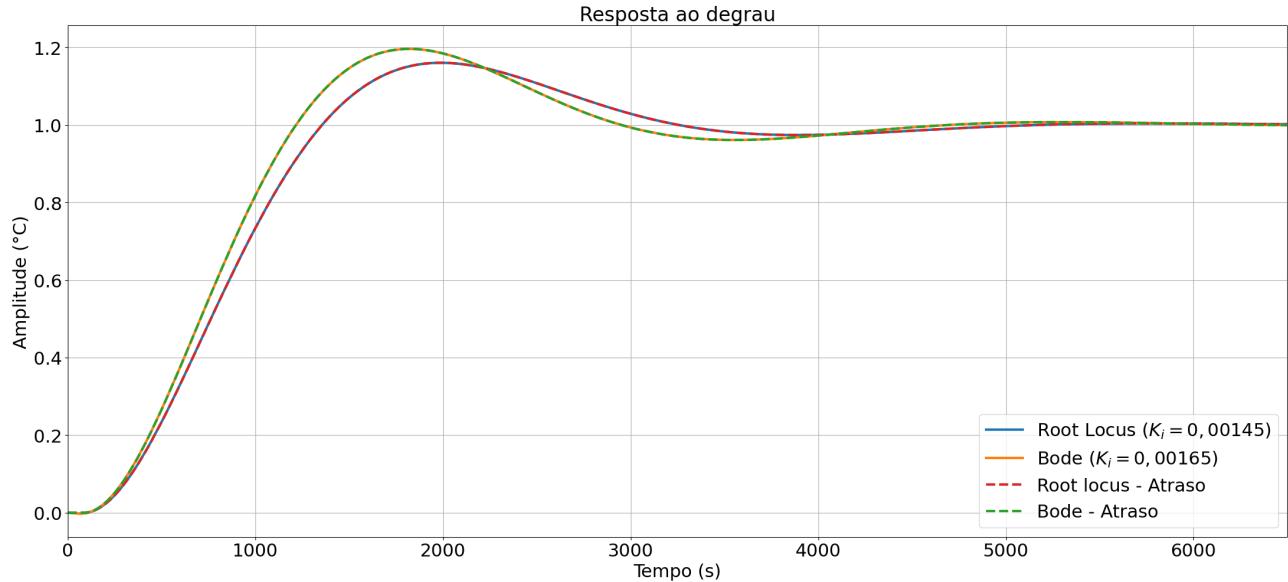


Figura 46: Comparação entre as respostas obtidas utilizando um controlador integrador.

Fonte: Autoria Própria.

Uma outra forma de analisar a precisão da Aproximação de Padé através de um gráfico de erro, que subtrai o valor obtido pela aproximação do valor real e demonstra seu módulo. Tal gráfico está demonstrado na Figura 47, em que é possível notar um erro mínimo entre os gráficos, sendo o maior erro próximo do  $t = 0$ s, decorrente do zero de fase não-miníma.

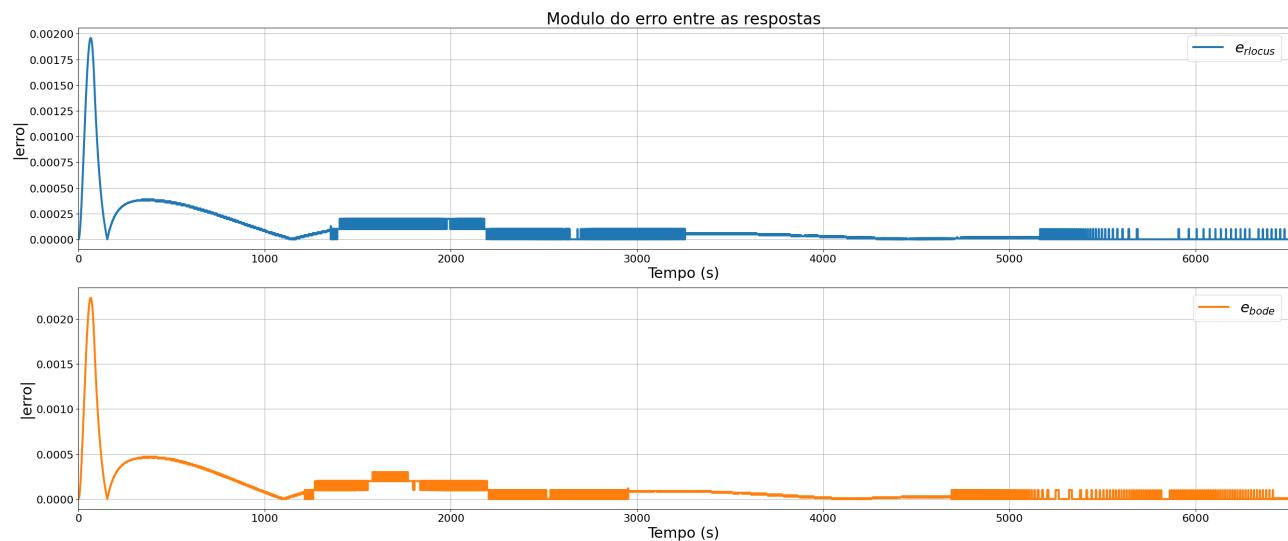


Figura 47: Erro entre as respostas com Aproximação de Padé e atraso no tempo utilizando o controlador integral.

Fonte: Autoria Própria.

Para gerar as Figuras 46 e 47 foi feito um código em *matlab* para a obtenção dos dados e outro, em Python, para gerar o plot. Tal código está demonstrado em Código 26.

### 8.3 Controlador Proporcional-Integral

Conforme dito na Subseção 8.2, o controlador do tipo integral é capaz de eliminar o erro em regime permanente do sistema porém prejudica a estabilidade e o comportamento no regime transitório. Um jeito de minimizar esse problema é através do controlador proporcional-integral que, além da inserção de um polo na origem, adiciona um zero bem próximo tal que atraia o polo inserido e que não modifique significativamente o lugar das raízes dos outros polos [Henriques, 2020b]. A Figura 48 demonstra o diagrama de blocos da planta com um controlador proporcional integral.

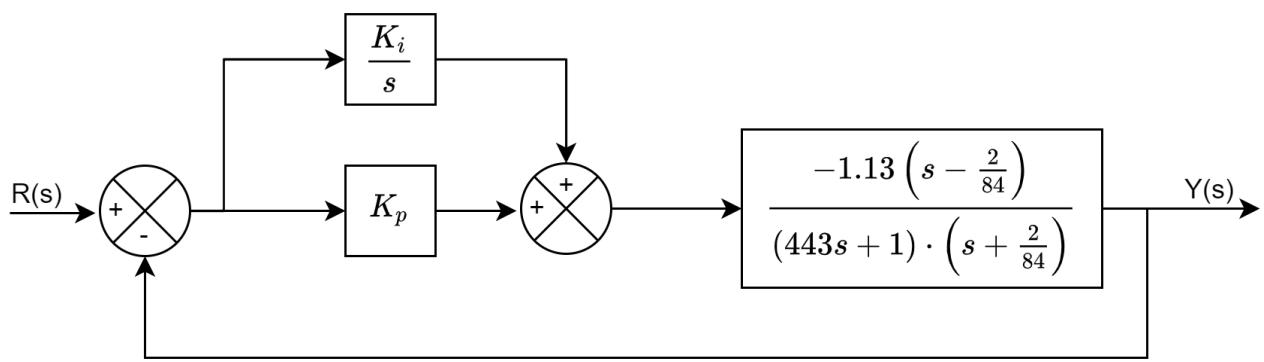


Figura 48: Diagrama de blocos para o controlador proporcional integral.

Fonte: Autoria Própria.

Resolvendo a malha de realimentação, encontramos a função de transferência da planta com o controlador proporcional-integral, demonstrada na Equação 80.

$$\frac{Y(s)}{R(s)} = \frac{(-1.13s + 0.0269) \cdot (K_p s + K_i)}{443s^3 + (11,55 - 1.13K_p)s^2 + (0,0238 - 1,13K_i + 0,0269K_p)s + 0,0269K_i} \quad (80)$$

Na Subsubseção 8.3.1 será feita a análise de estabilidade do sistema com o controlador proporcional-integral.

#### 8.3.1 Análise de Estabilidade

Um passo importante no projeto de controladores é a análise de estabilidade, uma vez que o uso de um controlador pode levar o sistema a instabilidade. Foi feita a análise de estabilidade para o sistema com controlador proporcional-integral utilizando dois métodos:

- Critério de Routh
- Diagrama de Nyquist

### 8.3.1.1 Critério de Routh

Através do diagrama de blocos exposto na Figura 48 e da função de transferência em Equação 80 foi possível montar a tabela para o Critério de Routh, conforme demonstrado na Tabela 3.

$s^3$	443	$(0,0238 - 1,13K_i + 0,0269K_p)$
$s^2$	$11,55 - 1,13K_p$	$0,0269K_i$
$s^1$	$\frac{-0,03K_p^2 + 0,284K_p + 1,276K_pK_i - 24,967K_i + 0,27489}{11,55 - 1,13K_p}$	0
$s^0$	$0,0269K_i$	

Tabela 3: Tabela para o Critério de Routh utilizando controlador proporcional-integral.

Para que o sistema seja estável não pode haver mudança de sinal nos elementos da primeira coluna, sendo assim, é necessário que o segundo, terceiro e quarto termo também sejam positivos. Posto isso, obtém-se a Equação 81.

$$\begin{cases} 11,55 - 1,13K_p > 0 \\ \frac{-0,03K_p^2 + 0,284K_p + 1,276K_pK_i - 24,967K_i + 0,27489}{11,55 - 1,13K_p} > 0 \\ 0,0269K_i > 0 \end{cases} \quad (81)$$

A segunda inequação exposta na Equação 81 representa uma região bem complexa, sendo assim, diferentemente do que foi exposto nas Subseção 8.1 e Subseção 8.2, deixaremos os valores de  $K_i$  em função de  $K_p$ . Nesse sentido, é possível encontrar os valores de  $K_i$  e  $K_p$ , expostos na Equação 82 e Equação 83.

$$K_p \leq 10,22 \quad (82)$$

$$\begin{cases} K_i \geq 0 \\ K_i \leq \frac{0,03K_p^2 - 0,284K_p - 0,27489}{1,276K_p - 24,967}, \quad \forall K_p \leq 10,22 \end{cases} \quad (83)$$

A Figura 49 representa o gráfico da segunda inequação expressa na Equação 83, em que a curva em azul representa o valor máximo de  $K_i$  para cada  $K_p$  e a região, também em azul, representa todos os valores de  $K_i$  possíveis para cada valor de  $K_p$ .

Através do gráfico da Figura 49, é possível observar o que o valor máximo de  $K_i \forall K_p \leq 10,22$  é sempre positivo nesse intervalo. Sendo assim, é possível escrever os valores de  $K_p$  e  $K_i$  conforme a Equação 84.

$$\boxed{\begin{aligned} K_p &\leq 10,22 \\ 0 \leq K_i &\leq \frac{0,03K_p^2 - 0,284K_p - 0,27489}{1,276K_p - 24,967}, \quad \forall K_p \leq 10,22 \end{aligned}} \quad (84)$$

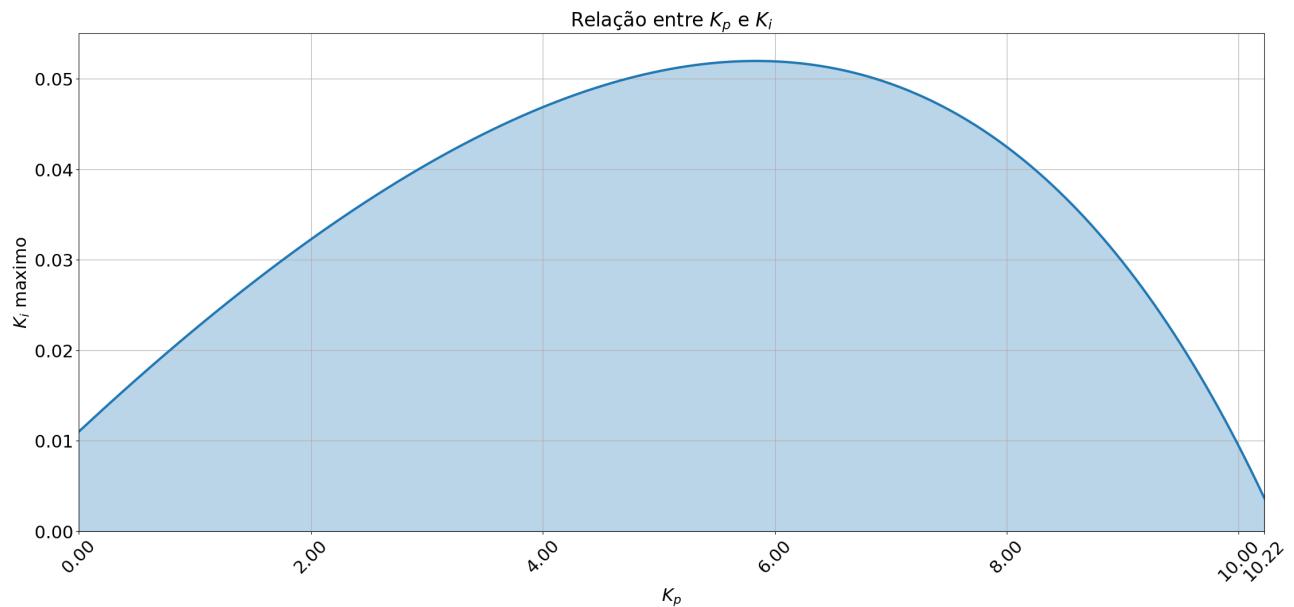


Figura 49: Relação entre o  $K_p$  e o  $K_i$  máximo para o controlador proporcional-integral.

Fonte: Autoria Própria.

### 8.3.1.2 Diagrama de Nyquist

Para a análise de estabilidade do sistema usando o Critério de Nyquist, iremos considerar a função de transferência do controlador conforme demonstrada na Equação 85.

$$G_c(s) = K_p + \frac{K_i}{s} = \frac{K_p s + K_i}{s} = K_p \frac{\left(s + \frac{K_i}{K_p}\right)}{s} \quad (85)$$

Através da Equação 85 é possível perceber que o controlador proporcional-integral insere um novo zero de malha aberta localizado em  $z = -\frac{K_i}{K_p} + j0$ , um polo também de malha aberta na origem e acrescenta um ganho  $K_p$ . Para fazer a simplificação do sistema, foi feito um cancelamento de polos e zeros utilizando o polo mais lento do sistema. A Figura 9 demonstra o mapa de polos e zeros do sistema com atraso de transporte, em que obtemos o polo mais lento  $p_2 = -0,00225 + j0$ . Sendo assim, teremos  $\frac{K_i}{K_p} = 0,00225$ .

Desse modo, é possível obter a nova função de transferência de malha aberta do sistema dependente somente de  $K_p$ , conforme a Equação 86.

$$FTMA(s) = \frac{K_p \left(s + \frac{1}{443}\right)}{s} \cdot \frac{-1,13 \left(s - \frac{2}{84}\right)}{(443s + 1) \cdot \left(s + \frac{2}{84}\right)} \quad (86)$$

$$FTMA(s) = \frac{K_p \cancel{\left(s + \frac{1}{443}\right)}}{s} \cdot \frac{-1,13 \left(s - \frac{2}{84}\right)}{443 \cdot \cancel{\left(s + \frac{1}{443}\right)} \cdot \left(s + \frac{2}{84}\right)} \quad (87)$$

$$FTMA(s) = -0.00255 K_p \frac{\left(s - \frac{2}{84}\right)}{s \left(s + \frac{2}{84}\right)} \quad (88)$$

Desse modo, foi traçado o Diagrama de Nyquist para  $K_p = 1$ , conforme demonstrado na Figura 50.

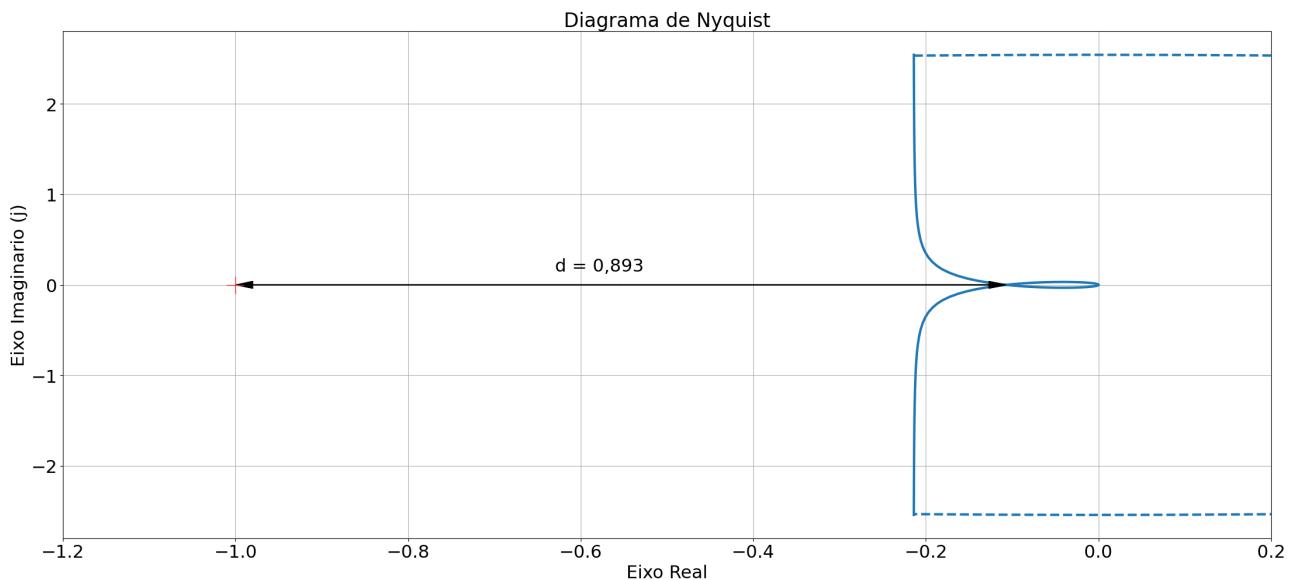


Figura 50: Diagrama de Nyquist para o sistema com controlador proporcional-integral.

Fonte: Autoria Própria.

O gráfico da Figura 50 demonstra que não há envoltórias no  $-1$ , o que significa que com  $K_p = 1$  o sistema é estável. É sabido que o incremento de um ganho  $K_p$  age como um fator de escala para esse gráfico, amplificando-o de tal forma que o faça envolver duas vezes o  $-1$ . Nesse caso, aplicando o valor de  $N = 2$  no Critério de Nyquist para o Diagrama de Nyquist modificado - sem considerar zeros e polos na origem -, em que temos  $P = 0$ , obtemos que o número de polos de malha fechada no semiplano direito é 2, levando o sistema a instabilidade. Esse caso é comprovado nas equações Equação 89.

$$\begin{aligned} 2 &= Z - 0 \\ Z &= 2 \end{aligned} \tag{89}$$

O ganho  $K_p$  mínimo para esse caso é encontrado através da Equação 90, em que  $d$  é a distância entre o ponto  $p = -1 + j0$  e o ponto em que o Diagrama de Nyquist cruza o eixo real, conforme demonstrado na Figura 50. Sendo assim, para manter o sistema estável,  $K_p$  deve seguir o que foi demonstrado na Equação 91.

$$1 > (1 - d) \cdot K_p \tag{90}$$

$$K_p \leq 9,345 \tag{91}$$

Através da Equação 91 é possível encontrar os valores  $K_p$  para a estabilidade, porém, para a análise de estabilidade do controlador proporcional-integral, o Critério de Nyquist peca em comparação ao Critério de Routh, uma vez que para simplificar expressivamente os cálculos foi necessário uma adaptação, que tira a generalização dos valores encontrados. Sendo assim, foram perdidas a análise dos valores de  $K_i$  para a estabilidade.

\*\*\*

Inicialmente, considerando a função de transferência do controlador proporcional-integral exposta na Equação 92, para facilitar o projeto, foi feito um **cancelamento de zeros e polos**, alocando o zero do controlador no polo mais lento do sistema não controlado que, de acordo com a Figura 9, se da por  $p_2 = -0,00225 + j0$ . Sendo assim, teremos  $\frac{K_i}{K_p} = 0,00225$ .

$$G_c(s) = K_p + \frac{K_i}{s} = \frac{K_p s + K_i}{s} = K_p \frac{\left(s + \frac{K_i}{K_p}\right)}{s} \quad (92)$$

Aplicando  $\frac{K_i}{K_p} = 0,00225$  no controlador, obtemos a nova função de transferência em malha aberta do sistema, exposto em Equação 95.

$$FTMA(s) = \frac{K_p \left(s + \frac{1}{443}\right)}{s} \cdot \frac{-1,13 \left(s - \frac{2}{84}\right)}{(443s + 1) \cdot \left(s + \frac{2}{84}\right)} \quad (93)$$

$$FTMA(s) = \frac{K_p \left(s + \frac{1}{443}\right)}{s} \cdot \frac{-1,13 \left(s - \frac{2}{84}\right)}{443 \cdot \left(s + \frac{1}{443}\right) \cdot \left(s + \frac{2}{84}\right)} \quad (94)$$

$$\boxed{FTMA(s) = -0.00255K_p \frac{\left(s - \frac{2}{84}\right)}{s \left(s + \frac{2}{84}\right)}} \quad (95)$$

Para o projeto do controlador com  $\zeta = 0,5$  foi utilizado o método do lugar das raízes. O gráfico do root locus pode ser visto na Figura 51, em que é possível se obter o valor de  $K_p$ .

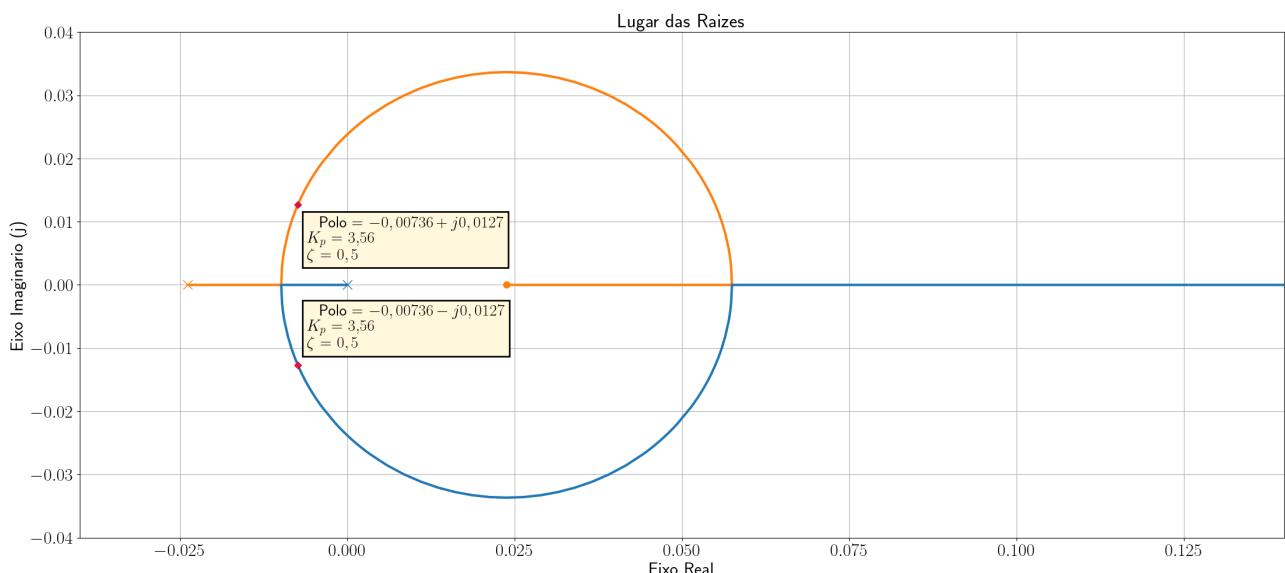


Figura 51: Root locus para o controlador proporcional-integral.

Fonte: Autoria Própria.

Através da Figura 51 é possível encontrar o valor de  $K_p$ , expresso na Equação 96. Descobrindo o valor de  $K_p$ , através da Equação 97 - que foi explicitada no início dessa seção - é possível encontrar o valor de  $K_i$ , exposto Equação 98.

$$\boxed{K_p = 3,56} \quad (96)$$

$$\frac{K_i}{K_p} = 0,00225 \quad (97)$$

$$K_i = 0.00801 \quad (98)$$

A seguir, será feito o controlador utilizando a análise na frequência para encontrar uma margem de fase  $\Phi_m > 50^\circ$ . Para isso, foi utilizada a função de transferência de malha aberta exposta em Equação 95. A Figura 52 apresenta o Diagrama de Bode para  $K_p = 1$ .

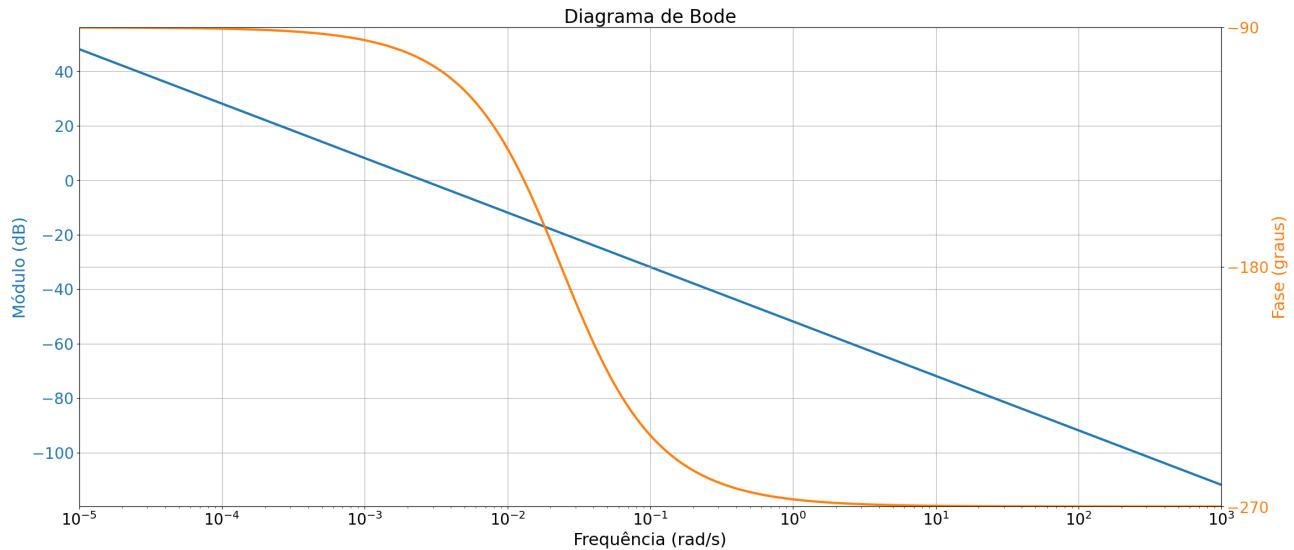


Figura 52: Diagrama de bode inicial com controlador proporcional-integral.

Fonte: Autoria Própria.

A Figura 52 demonstra o Diagrama de Bode para o sistema com atraso, conforme demonstrado na equação Equação 95. A margem de ganho para esse sistema é de  $\Phi_m = 77,92^\circ$ . Para que  $\Phi_m = 50^\circ$  basta projetar um  $K_p$  tal que sua contribuição leve o gráfico de módulo a cruzar o eixo quando a fase  $\Phi = -130^\circ$ . Para isso, iremos encontrar a frequência  $\omega_i$  tal que  $\angle G(j\omega_i) = -130^\circ$ , conforme demonstrado nas equações Equação 99 e Equação 100.

$$\angle G(j\omega_i) = -90 - 2 \arctan(42\omega_i) \quad (99)$$

$$\omega_i = 8,665 \times 10^{-3} \text{ rad s}^{-1} \quad (100)$$

Após encontrado  $\omega_i$ , foi encontrado o  $|G(j\omega_i)|_{dB}$ , conforme demonstrado nas equações Equação 101 e Equação 102.

$$|G(j\omega_i)|_{dB} = -51,869 - 20 \log(\omega_i) \quad (101)$$

$$|G(j\omega_i)|_{dB} = -10,6253 \text{ dB} \quad (102)$$

Sendo assim, é preciso um ganho  $K_p$  tal que sua contribuição no diagrama de módulo seja o positivo do ganho demonstrado em Equação 102, conforme exposto nas equações Equação 103 e Equação 104.

$$|G(j\omega_i)|_{dB} = 20 \log (K_p) \quad (103)$$

$$K_p = 3,398 \quad (104)$$

Com o  $K_p$  encontrado na Equação 104, é possível encontrar o valor de  $K_i$  através da Equação 97. O valor de  $K_i$  está exposto na Equação 105

$$K_i = 0,00764 \quad (105)$$

A Figura 53 apresenta o novo diagrama de bode com  $K_p = 3,398$ , em que é possível ver o controlador atuando modificando a margem de fase conforme desejado.

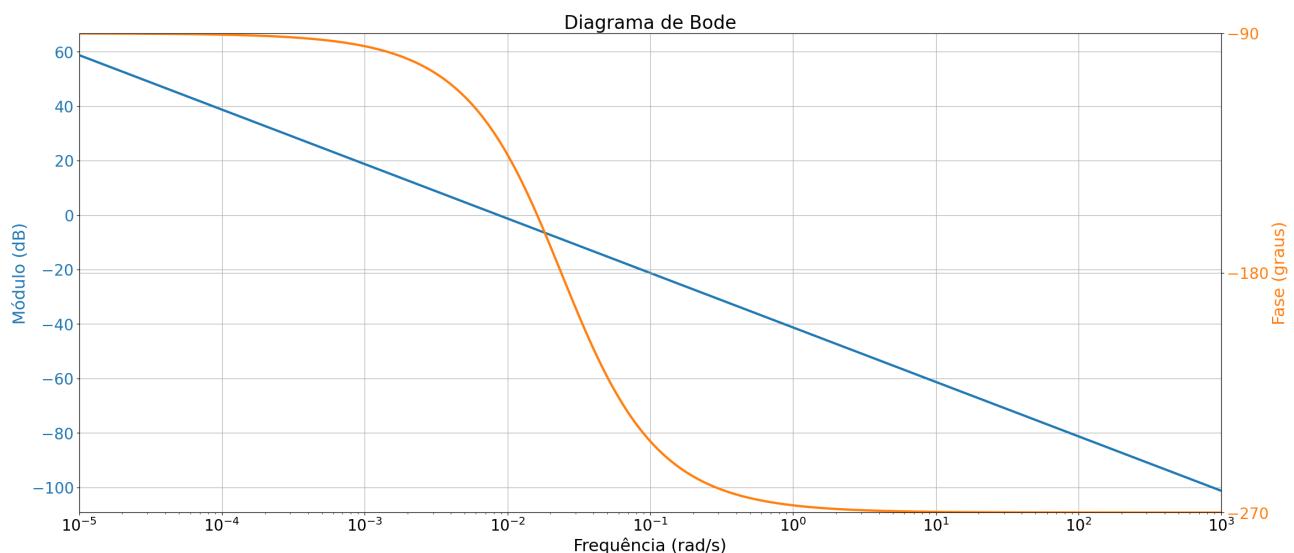


Figura 53: Diagrama de bode com controlador proporcional-integral com  $K_p = 3,398$ .

Fonte: Autoria Própria.

Para a análise dos resultados foi feita a comparação entre as respostas dos sistema com Aproximação de Padé e o atraso puro de tempo, utilizando os valores de  $K_p$  e  $K_i$  obtidos através do lugar das raízes e da análise em frequência, conforme demonstrado no diagrama de blocos da Figura 54. O resultado para essa análise pode ser visto na Figura 55, em que é possível observar que existe um erro em regime transitório das respostas com Aproximação de Padé e atraso no tempo, porém a aproximação ainda se demonstra eficaz. Para comprovar essa afirmação, foi feito um gráfico para a análise do erro, exposto na Figura 56, que demonstra um erro máximo  $e \approx 0,13$ ; aceitável para a nossa aplicação.

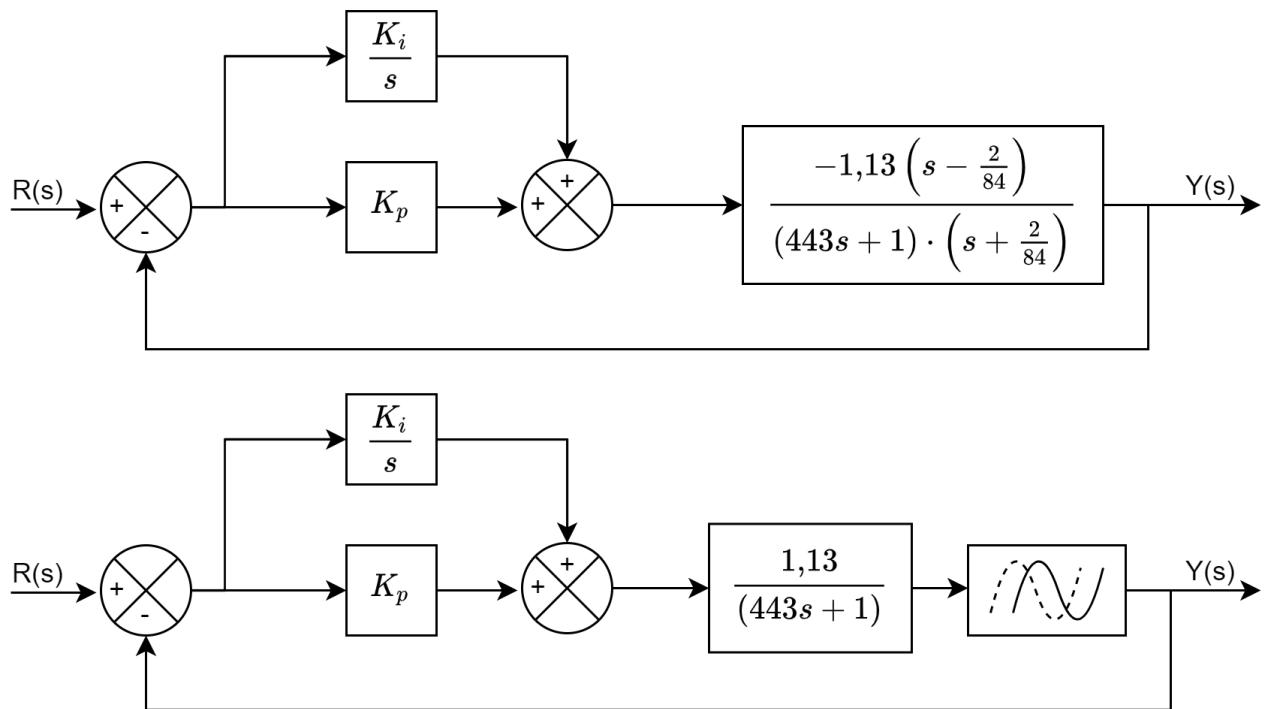


Figura 54: Comparaçāo entre Padé e Atraso puro no tempo com o controlador proporcional-integral.

Fonte: Autoria Própria.

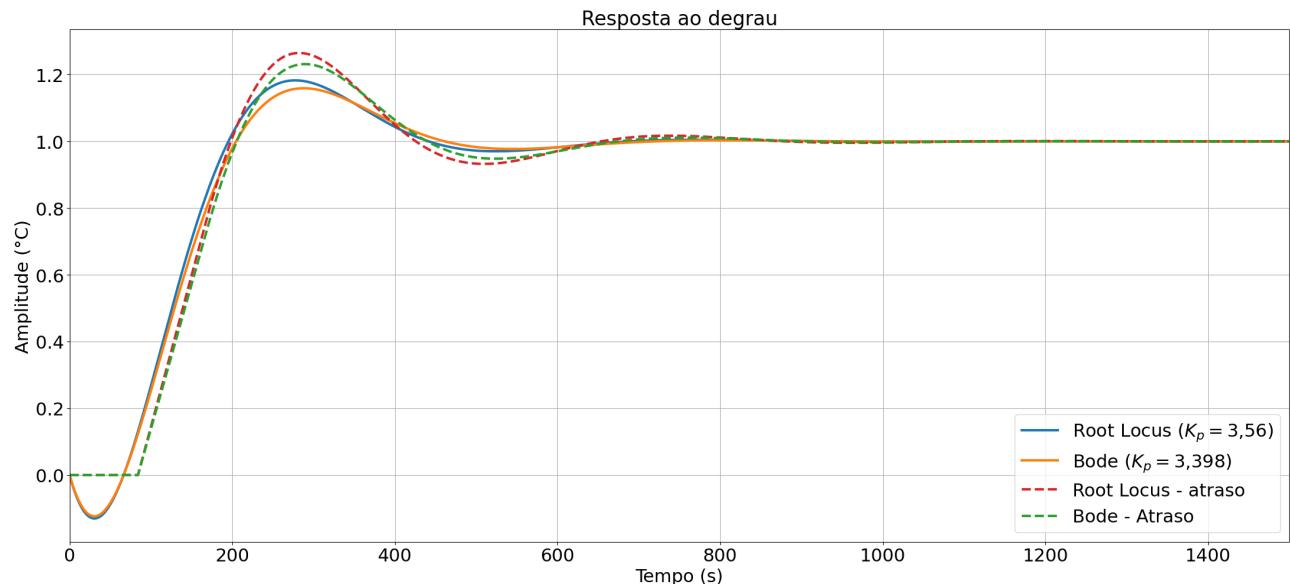


Figura 55: Comparaçāo entre as respostas obtidas utilizando um controlador proporcional-integral.

Fonte: Autoria Própria.

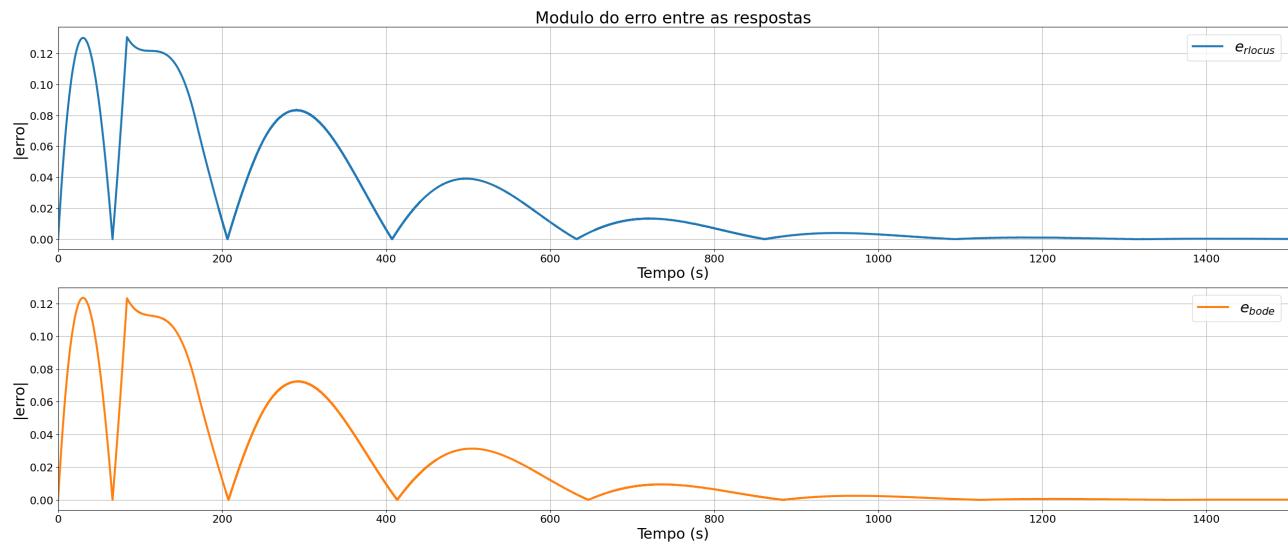


Figura 56: Erro entre as respostas com Aproximação de Padé e atraso no tempo utilizando o controlador proporcional-integral.

Fonte: Autoria Própria.

## 8.4 Análise dos Controladores

Nessa seção será feita a análise e comparação entre as respostas obtidas com cada tipo de controlador. Para isso, escolhemos os parâmetros para os controladores obtidos através do lugar das raízes e plotamos suas respostas na Figura 57.

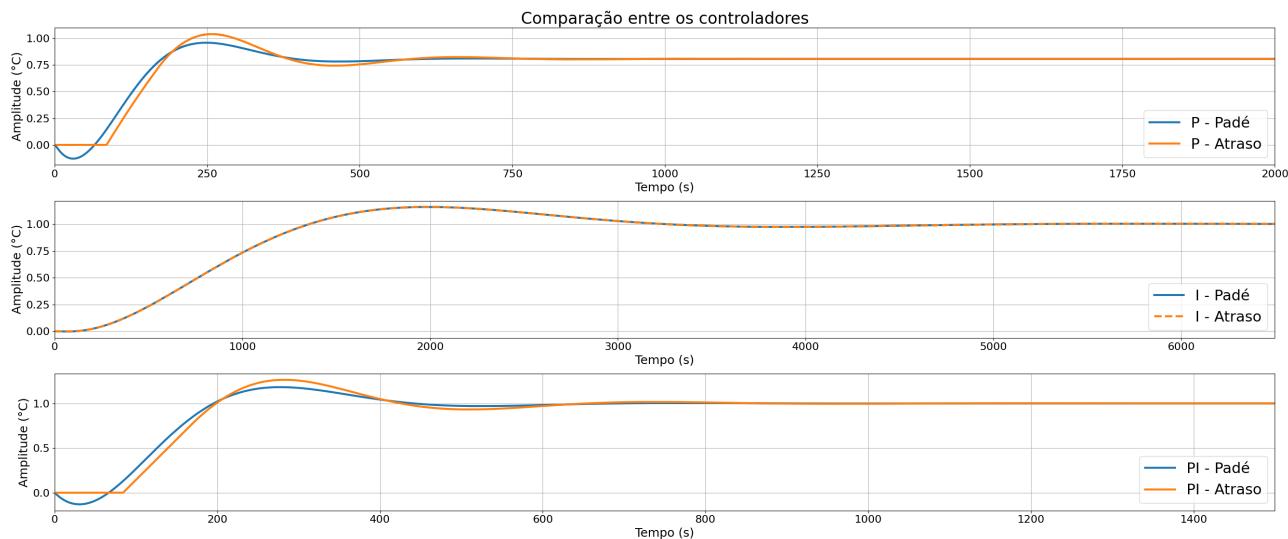


Figura 57: Comparação entre as respostas do sistema com os controladores.

Fonte: Autoria Própria.

Na Figura 57 é perceptível a ação e comportamento de cada tipo de controlador no sistema em comparação com a resposta do sistema em malha aberta (Figura 13), em que o controlador proporcional torna o sistema mais rápido, diminuindo o tempo de acomodação, porém não é capaz de eliminar o erro de regime permanente. O primeiro gráfico da Figura 57 demonstra

esse comportamento. O controlador do tipo integral elimina o erro de regime permanente, mas prejudica categoricamente o regime transitório do sistema, elevando consideravelmente o tempo de acomodação, conforme visto no segundo gráfico da Figura 57. Já o controlador do tipo proporcional-integral consegue unir as características positivas do dois outros controladores, tornando o sistema mais rápido e eliminando o erro de regime permanente, o que pode ser observado no terceiro gráfico da Figura 57.

Uma outra análise que é interessante de ser feita é como o comportamento do sistema com a Aproximação de Padé se assemelha ao do sistema real com atraso puro no tempo, uma vez que todos os cálculos são feitos utilizando uma aproximação de padé de primeira ordem e que, por ser uma aproximação, não representa com exatidão o modelo real, sendo necessário então um critério pra saber se essa aproximação é satisfatória ou não. Para isso, foi plotado o gráfico da Figura 58, que demonstra o módulo da diferença entre os valores obtidos com atraso e com a aproximação para cada tempo. O primeiro gráfico dessa figura demonstra o erro para o sistema com controlador proporcional, em que há uma diferença maior nos tempos iniciais e zera no regime estacionário, sendo que esse erro inicial não representa um grande problema para o projeto do forno em específico. O segundo gráfico demonstra o erro para o sistema com controlador integral, em que é possível observar um erro mínimo entre a simulação dos dois sistemas, o que demonstra que a aproximação, para esse caso, foi excelente. Já o terceiro gráfico da Figura 58 demonstra o erro para o sistema com controlador proporcional-integral, em que é possível observar, análogo ao que foi obtido com o controlador proporcional, uma diferença maior nos tempos iniciais e zera no regime estacionário, porém esse erro não representa problemas para o projeto.

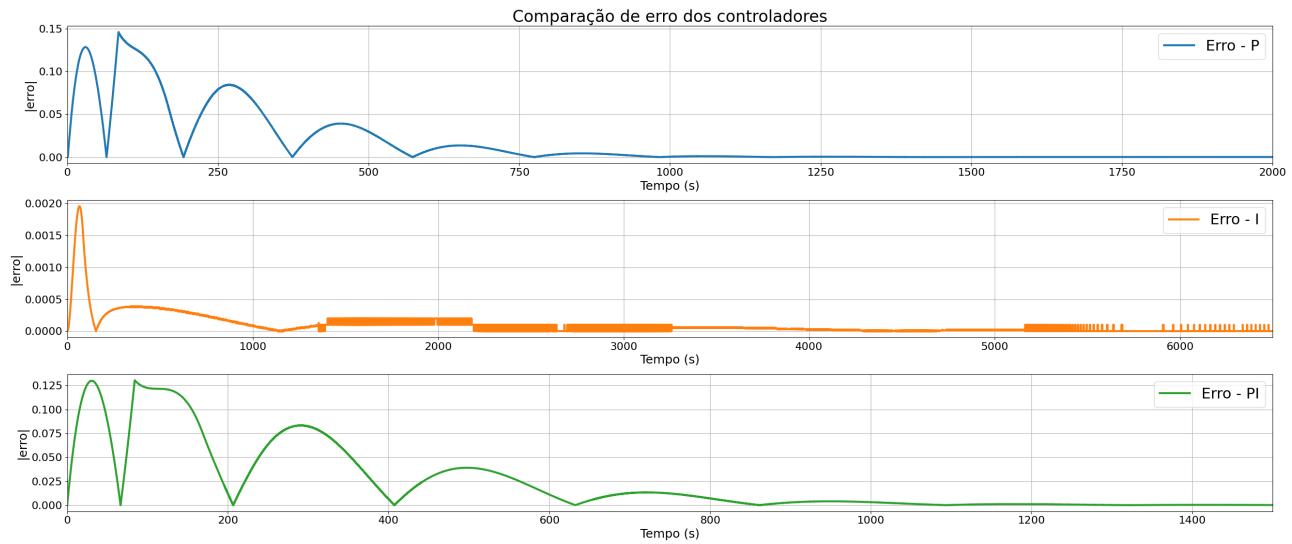


Figura 58: Comparação entre as respostas com Aproximação de Padé a atraso no tempo para todos os controladores.

Fonte: Autoria Própria.

## 9 Compensadores

Durante a construção de um sistema de controle, sabemos que a modificação adequada na dinâmica da planta é uma maneira simples de atender as especificações do projeto. Porém,

muitas vezes a planta é fixa e não pode ser modificada. Através de análises como a do lugar das raízes, pode ser observado que um simples ajuste no ganho pode não ser suficiente para o desempenho desejado. Sendo assim, é possível melhorar o desempenho do sistema por meio da inclusão de um compensador, que é o projeto de um filtro cujas características tendem a compensar as caracterizações indesejadas e inalteráveis da planta [Henriques, 2020a].

Nessa seção serão feitas a compensação do sistema utilizando dois tipos de compensadores:

- Compensador por Atraso de Fase
- Compensador por Avanço de Fase

## 9.1 Compensador por Atraso de Fase

Nessa subseção será projetado um compensador de atraso de fase pelo método do lugar das raízes para se obter coeficiente de amortecimento superior a  $\zeta > 0,5$  e um outro pelo método da resposta em frequência para que a margem de fase  $\Phi_m > 50^\circ$ . Para ambos compensadores, a constante de erro estático de velocidade deve ser pelo menos 10 vezes superior ao de malha aberta. A função de transferência da planta com a aproximação de padé para o atraso no tempo dada na Equação 22 é um do tipo 0, sendo assim, para serem feitos os cálculos, foi adicionado um integrador, de forma que obtemos a função de transferência exposta na Equação 106.

$$G(s) = \frac{0,00145}{s} \cdot \frac{-1.13s + 0.0269}{443s^2 + 11,55s + 0.0238} = \frac{-0.00163s + 3.9005e - 05}{s(443s^2 + 11,55s + 0.0238)} \quad (106)$$

Com a nova função de transferência exposta na Equação 106 é possível encontrar a constante de erro estático de velocidade  $K_v$  inicial através da Equação 107. Com isso,  $K_v$  é encontrado na Equação 108

$$K_v = \lim_{s \rightarrow 0} sG(s) \quad (107)$$

$$K_v = 0,00163 \quad (108)$$

A Equação 109 apresenta as funções de transferência de controladores por atraso de fase que serão utilizadas nos projetos.

$$\begin{aligned} G_c(s) &= K_c \frac{s+z}{s+p} \\ G_c(s) &= K_c \frac{s+\frac{1}{T}}{s+\frac{1}{\beta T}} \\ G_c(s) &= K_c \frac{Ts+1}{\alpha Ts+1} \end{aligned} \quad (109)$$

A seguir, serão feitos os projetos dos compensadores através do lugar das raízes, exposto na Subsubseção 9.1.1 e através da análise na frequência, exposto na Subsubseção 9.1.2.

### 9.1.1 Projeto através do Lugar das Raízes

A Figura 59 representa o lugar das raízes do sistema não compensado com o integrador. Na figura, é possível obter os polos de malha fechada desejados, que resultam em um  $\zeta \approx 0,5$ . Esses polos estão demonstrados na Equação 110.

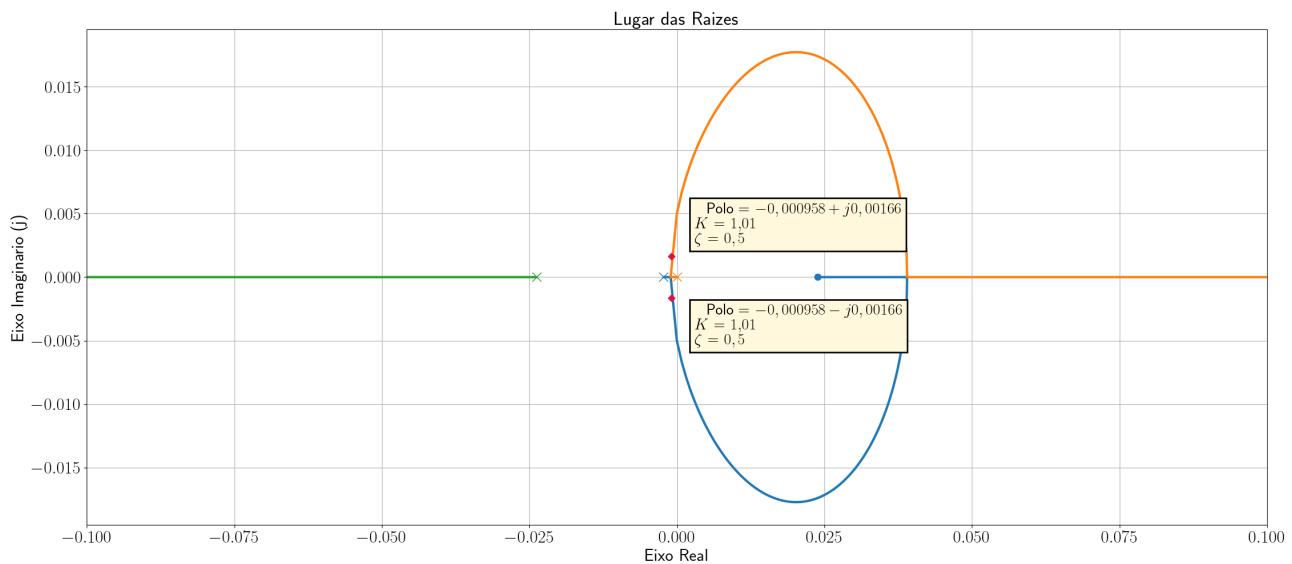


Figura 59: Polos de malha fechada desejados pro sistema sem compensador por atraso de fase.

Fonte: Autoria Própria.

$$\begin{aligned} p_1 &= -0.00096 + j0.00165 \\ p_2 &= -0.00096 - j0.00165 \\ \zeta &= 0,503 \end{aligned} \quad (110)$$

Para o projeto do compensador por atraso de fase utilizando pelo método do lugar das raízes, será utilizada a segunda função de transferência exposta Equação 109, que pode ser vista novamente na Equação 111.

$$G_c(s) = K_c \frac{s + \frac{1}{T}}{s + \frac{1}{\beta T}} \quad (111)$$

Inicialmente, iremos considerar  $K_c = 1$ . Sendo assim, para aumentar 10 vezes o valor da constante de erro estático de velocidade, é necessário atender a condição exposta na Equação 112. Sendo assim, foi encontrado o valor de  $\beta$  em Equação 113.

$$\lim_{s \rightarrow 0} \left( \frac{s + \frac{1}{T}}{s + \frac{1}{\beta T}} \right) = 10 \quad (112)$$

$$\boxed{\beta = 10} \quad (113)$$

Com  $\beta$  encontrado, é preciso projetar o controlador de tal forma que os zeros e o polos não alterem de forma expressiva a condição de fase do lugar das raízes. Uma forma de fazer isso é alocando o zero bem próximo a origem, de tal forma que o polo, apesar de estar 10 vezes mais próximo da origem, contribua no critério da fase de forma similar ao zero, assim não modificando drasticamente o lugar das raízes inicial. Sendo assim, escolhemos o zero em  $z = -0.0001$ . Com o zero definido, sabendo que  $\beta = 10$ , encontramos o polo em  $p = -0.00001$ . Assim, é possível montar a função de transferência do controlador dependente somente do ganho  $K_c$ , conforme demonstrado na Equação 114.

$$G_c(s) = K_c \frac{s + 0.0001}{s + 0.00001} \quad (114)$$

Com a função de transferência do compensador encontrada em Equação 114 é possível traçar o lugar das raízes do sistema compensado e encontrar e encontrar os polos que atendem a nossa condição e, através deles, encontrar o ganho  $K_c$ , o qual desejamos que esteja o mais próximo de 1 possível, para obtermos o novo  $K_v$  10 vezes maior. A Figura 60 apresenta o *root locus* do sistema compensando, em que é possível observar os novos polos de malha fechada, definidos na Equação 115.

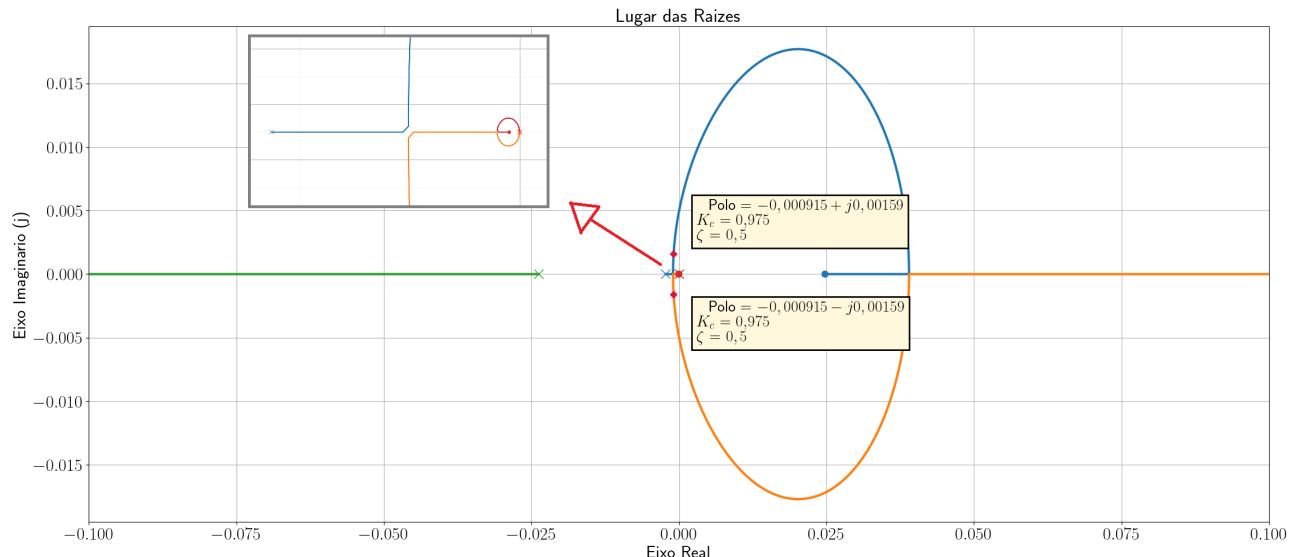


Figura 60: Root Locus e polos de malha fechada desejados pro sistema com compensador por atraso de fase.

Fonte: Autoria Própria.

$$\begin{aligned} p_1 &= -0.00091 + j0.00162 \\ p_2 &= -0.00091 - j0.00162 \\ \zeta &= 0,49 \end{aligned} \quad (115)$$

Para obter esses polos, é necessário um ganho  $K_c$  conforme demonstrado na Figura 60. Esse ganho está exposto em Equação 116.

$$K_c = 0.975 \quad (116)$$

Função de transferência do compensador: Com todos os parâmetros calculado, é possível obter a função de transferência do compensador, exposta na Equação 117.

$$G_c(s) = 0.975 \frac{(s + 0.0001)}{(s + 0.00001)} \quad (117)$$

Com a função de transferência do controlador calculada, é possível encontrar a nova função de transferência do sistema e a nova constante de erro estático de velocidade, expostas nas equações Equação 118 e Equação 119, respectivamente.

$$G(s) = \frac{0,00158925(s - 0.0238)(s + 0.0001)}{s(s + 0.0238)(s + 0.0226)(s + 0.00001)} \quad (118)$$

$$K_v = \lim_{s \rightarrow 0} s \cdot G_c(s) \cdot G(s) = 0,0158925 \quad (119)$$

### 9.1.2 Projeto através da Análise na Frequência

Para o projeto na frequência, queremos uma margem de fase  $\Phi_m > 50^\circ$  e a constante de erro estático de velocidade  $K_v$  pelo menos 10 vezes maior ao de malha aberta. Assim, iremos utilizar a mesma função de transferência da planta com atraso de propagação modelado pela Aproximação de Padé e com o integrador exposta na Equação 106. Essa função pode ser revista na Equação 120.

$$G(s) = \frac{0,00145}{s} \cdot \frac{-1.13s + 0.0269}{443s^2 + 11,55s + 0.0238} = \frac{-0.00163s + 3.9005e - 05}{s(443s^2 + 11,55s + 0.0238)} \quad (120)$$

Para a função de transferência exposta na Equação 120, temos a constante de erro estático de velocidade e margem de fase conforme expostas na Equação 121.

$$\begin{aligned} K_v &= \lim_{s \rightarrow 0} s \cdot G(s) = 0,00163 \\ \Phi_m &= 51,47^\circ \end{aligned} \quad (121)$$

A Figura 61 demonstra o diagrama de bode para o sistema não compensado, conforme exposto na Equação 120.

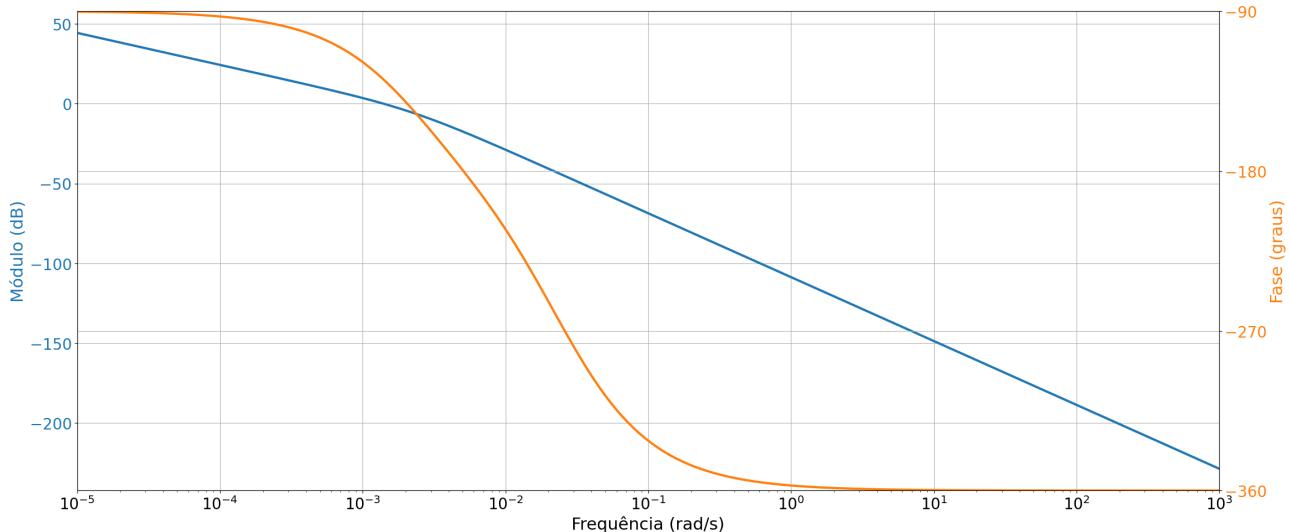


Figura 61: Diagrama de bode pro sistema não compensado.

Fonte: Autoria Própria.

Para a análise na frequência, iremos considerar um compensador por atraso de fase conforme demonstrado na terceira equação exposta Equação 109. Esse compensador pode ser revisto na Equação 122.

$$G_c(s) = K_c \frac{Ts + 1}{\alpha Ts + 1} \quad (122)$$

No projeto na frequência, devido a uma não exatidão obtida no projeto utilizando o *root locus*, iremos considerar uma folga. Sendo assim, será projetado um compensando com o

objetivo de obter um novo  $K_v$  11 vezes superior que o  $K_v$  inicial. Dessa maneira, foi encontrado um  $K_c$  que atenda essa necessidade, demonstrado na Equação 125.

$$K_v = \lim_{s \rightarrow 0} s \cdot G_c(s) \cdot G(s) \quad (123)$$

$$11 \cdot 0.00163 = 0.00163K_c \quad (124)$$

$$K_c = 11 \quad (125)$$

Com o ganho  $K_c$  inserido no sistema, iremos projetar o sistema com uma folga, com a margem de fase  $\Phi_m = 60^\circ$ . Para isso, vamos encontrar a frequência  $\omega_i$  que de essa margem. A Figura 62 apresenta o diagrama de bode pro sistema com o ganho  $K_c$  e, através dela, é possível encontrar  $\omega_i$ , exposta na Equação 126.

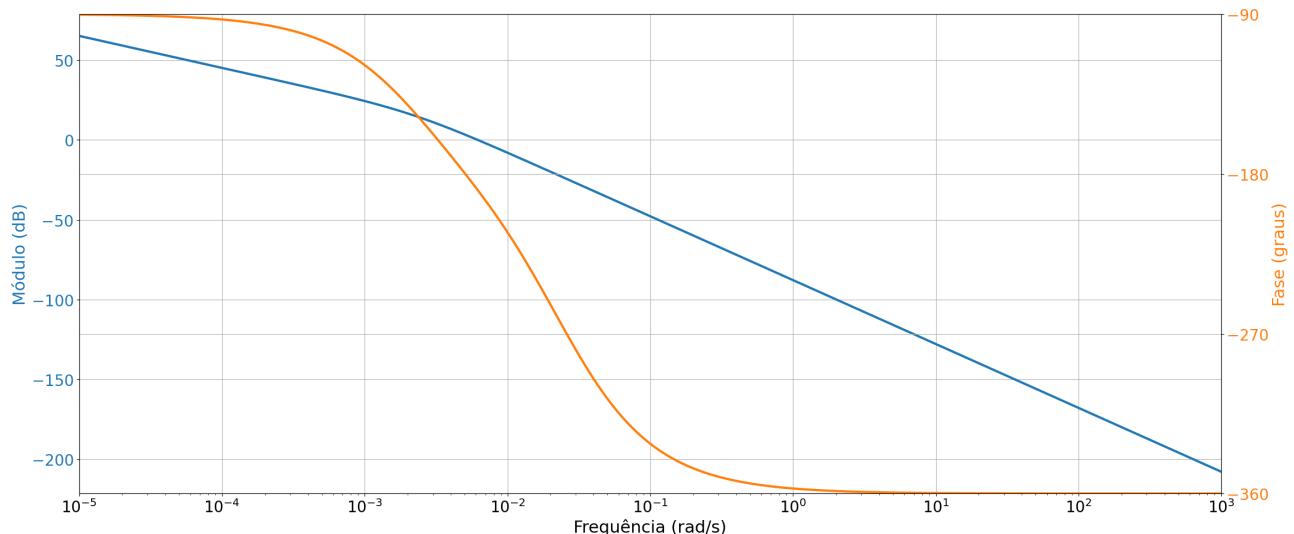


Figura 62: Diagrama de bode pro sistema com o ganho  $K_c$ .

Fonte: Autoria Própria.

$$\omega_i = 0,001\ 04 \text{ rad s}^{-1} \quad (126)$$

Com a frequência  $\omega_i$  calculada, foi encontrado o valor do gráfico de modulo nessa frequência, exposto na Equação 127

$$|G(j\omega_i)|_{dB} = -23,9 \text{ dB} \quad (127)$$

Com isso, temos que a assintota de alta frequência do controlador deve contribuir o negativo do encontrado anteriormente, para zerar o gráfico nesse ponto. Dessa maneira é possível encontrar o valor do parâmetro  $\alpha$  através da equação exposta em Equação 128. O valor de  $\alpha$  está demonstrado na Equação 129.

$$20 \log \left( \frac{1}{a} \right) = -23,9 \quad (128)$$

$$a = 15,6675 \quad (129)$$

Para o projeto do compensador, iremos definir a posição do polo uma década abaixo da frequência que escolhemos para ser a frequência de cruzamento ( $\omega_i$ ). Sendo assim, é possível obter o valor de  $T$  através da Equação 130. O valor do parâmetro  $T$  está demonstrado em Equação 131.

$$\frac{1}{T} = 0,000104 \quad (130)$$

$$T = 9615,38 \quad (131)$$

Com todos os parâmetros encontrados, é possível montar a função de transferência do compensador, exposta na Equação 132.

$$G_c(s) = 11 \cdot \frac{9615,38s + 1}{150648,96s + 1} \quad (132)$$

Com isso, é possível obter a nova função de transferência do sistema em malha aberta, exposta nas equações Equação 133 e Equação 134.

$$G(s) = 0,702 \frac{\left(s + \frac{1}{9615,38}\right)}{\left(s + \frac{1}{150648,96}\right)} \cdot 0,00163 \frac{(s - 0,0238)}{s(s + 0,0238)(s + 0,0226)} \quad (133)$$

$$G(s) = 0,00114 \cdot \frac{\left(s + \frac{1}{9615,38}\right)(s - 0,0238)}{s\left(s + \frac{1}{150648,96}\right)(s + 0,0238)(s + 0,0226)} \quad (134)$$

A Figura 63 demonstra o novo diagrama de bode para o sistema com compensador por atraso de fase, em que é possível notar uma mudança no gráficos de modulo e fase em comparação com o diagrama original, exposto na Figura 61. Com isso, obtemos os novos valores de margem de fase  $\Phi_m$  e  $K_v$  conforme exposto na Equação 135.

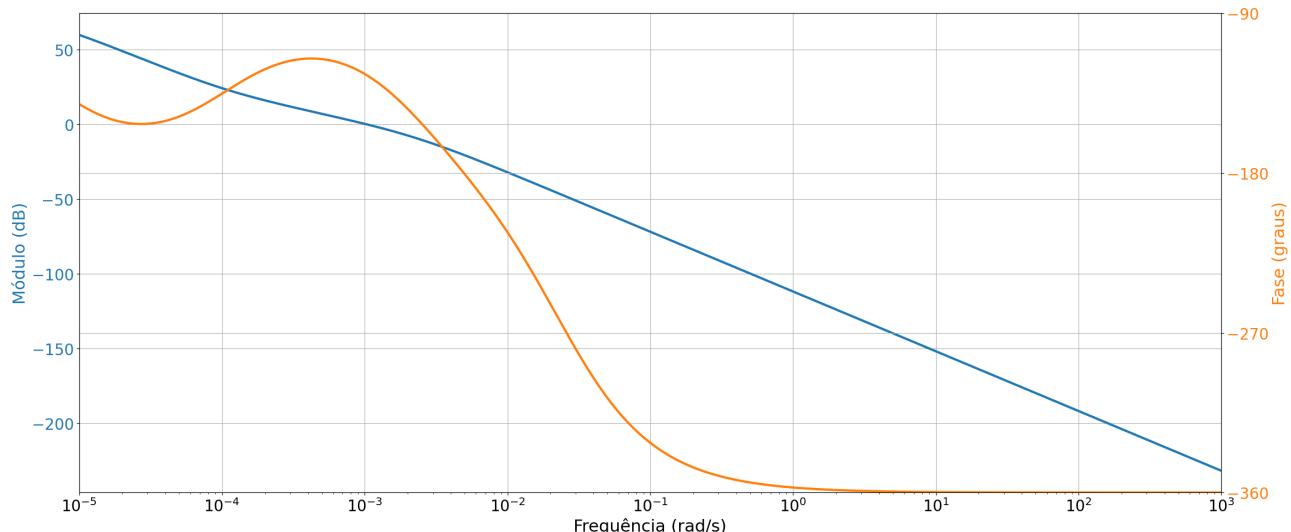


Figura 63: Diagrama de bode do sistema com compensador de atraso de fase.

Fonte: Autoria Própria.

$$\boxed{\begin{aligned}\Phi_m &= 55^\circ \\ K_v &= 0.0179 \quad (11x \text{ maior})\end{aligned}} \quad (135)$$

\*\*\*

Com os controladores obtidos pelo método do lugar das raízes e pela análise na frequência, foram plotados os gráficos da resposta ao degrau e resposta a rampa do sistema com integrador e compensado por atraso de fase.

A Figura 64 demonstra a resposta ao degrau dos sistemas com compensadores projetados através do *Root Locus* e através do Diagrama de Bode. É possível perceber que o sistema agora é do tipo 1, sendo assim, o erro em regime permanente para uma entrada ao degrau é nulo. Devido a implantação do compensador bem próximo a origem, o sistema se torna extremamente lento, tendo um tempo de acomodação  $T_{a2\%} \approx 15\,000 \text{ s}$ .

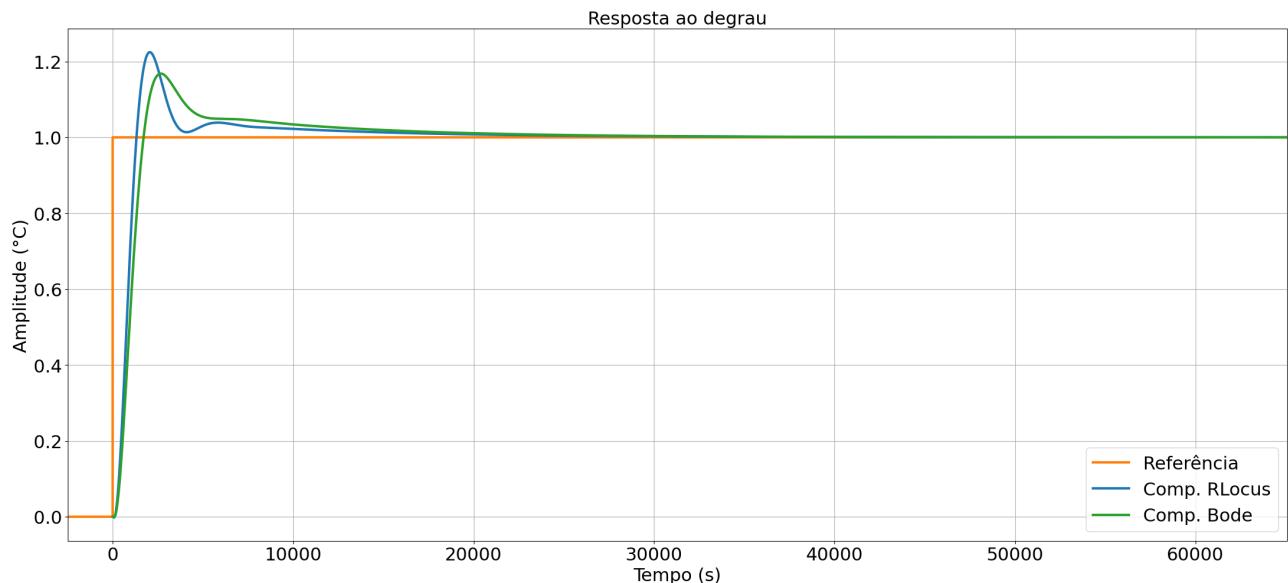


Figura 64: Resposta ao degrau do sistema com compensador atraso de fase.

Fonte: Autoria Própria.

A Figura 65 demonstra a resposta a rampa dos sistemas com compensadores através do *Root Locus* e através do Diagrama de Bode. É possível perceber que o sistema agora é do tipo 1, sendo assim, existe um erro em regime permanente para a entrada em rampa.

A Figura 66 demonstra a resposta do sistema a uma entrada em rampa utilizando o controlador proporcional-integral obtido na Subseção 8.3 e utilizando o compensador por atraso de fase obtido através do método do lugar das raízes. Nessa figura é possível perceber que o controlador PI possui um melhor resultado, pois é capaz de seguir a referência com menor oscilação e erro no regime transitório.

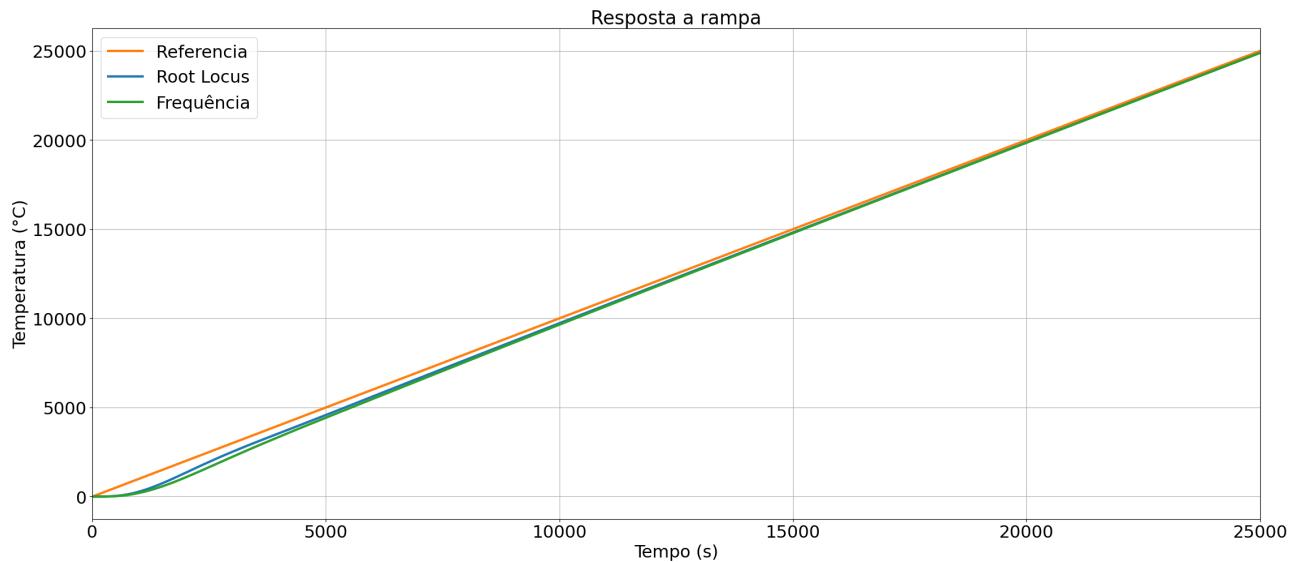


Figura 65: Resposta à rampa do sistema com compensador atraso de fase.

Fonte: Autoria Própria.

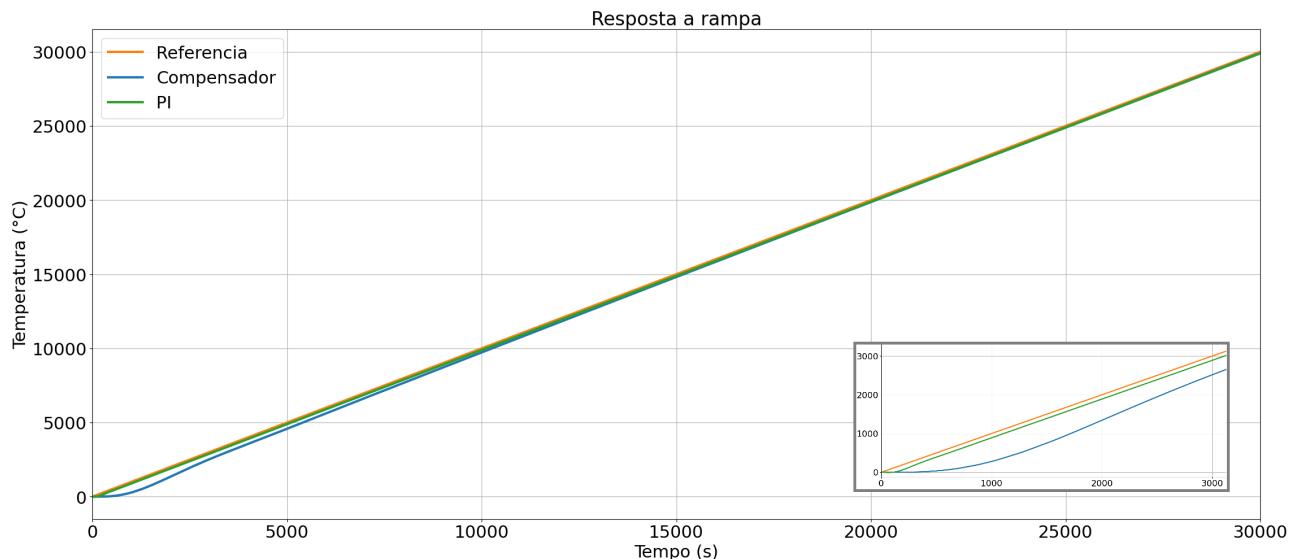


Figura 66: Comparação entre as respostas utilizando compensador e PI.

Fonte: Autoria Própria.

## 9.2 Compensador por Avanço de Fase

Nessa subseção será projetado um compensador de avanço de fase pelo método do lugar das raízes para se obter coeficiente de amortecimento superior a  $\zeta > 0,5$  e um outro pelo método da resposta em frequência para que a margem de fase  $\Phi_m > 50^\circ$ . Para ambos compensadores, a constante de erro estático de velocidade deve ser pelo menos 10 vezes superior ao de malha aberta. A função de transferência da planta com a aproximação de padé para o atraso no tempo dada na Equação 22 é um do tipo 0, sendo assim, para serem feitos os cálculos, foi adicionado um integrador, de forma que obtemos a função de transferência exposta na Equação 136.

$$G(s) = \frac{0,00145}{s} \cdot \frac{-1.13s + 0.0269}{443s^2 + 11,55s + 0.0238} = \frac{-0.00163s + 3.9005e - 05}{s(443s^2 + 11,55s + 0.0238)} \quad (136)$$

Igualmente ao que foi feito na Subseção 9.1, é possível encontrar o valor inicial da constante de erro de velocidade através da Equação 137.

$$K_v = \lim_{s \rightarrow 0} s \cdot G(s) = 0,00163 \quad (137)$$

Apesar do controlador por avanço de fase não ser o ideal para ajustes no regime permanente, foi projetado um controlador por avanço de fase que eleve em 12 vezes a constante de erro de velocidade. Esse controlador está explicitado na Equação 138.

$$G_c(s) = K_c \cdot \frac{s + z}{s + p}, \quad z < p \quad (138)$$

A seguir, serão feitos os projetos dos compensadores através do lugar das raízes, exposto na Subsubseção 9.2.1 e através da análise na frequência, exposto na Subsubseção 9.2.2.

### 9.2.1 Projeto através do Lugar das Raízes

Inicialmente, vamos reescrever a Equação 138 conforme visto na Equação 139.

$$G_c(s) = K_c \cdot \frac{s + \frac{1}{T}}{s + \frac{1}{\alpha T}} \quad (139)$$

Da equação Equação 139, para termos uma constante de erro estático de velocidade 12x maior que a original, chegamos em Equação 140.

$$K_c \cdot \alpha \geq 12 \quad (140)$$

O sistema não compensado possui o mapa de polos e zeros da Figura 67.

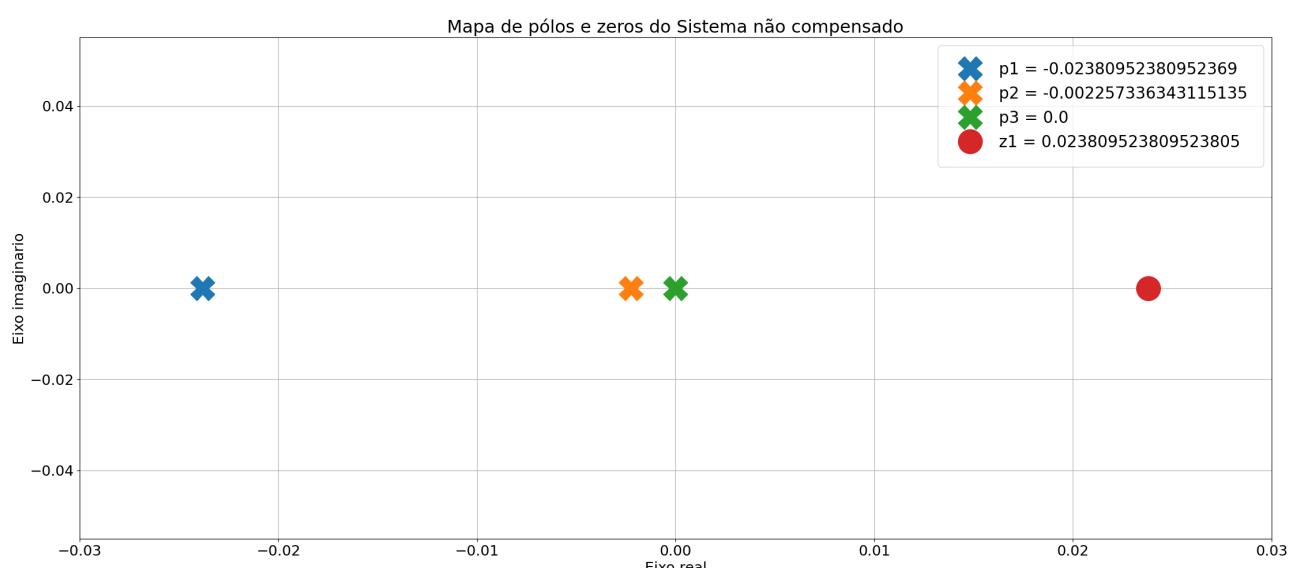


Figura 67: Mapa do sistema não compensado.

Fonte: Autoria Própria.

Dessa maneira, foi projetado um compensador por avanço de fase tal que o polo e zero sejam incluído entre os polos em azul (p1) e o em amarelo(p2), uma vez que queremos modificar o lugar de p2 e p3 em malha fechada. Sendo assim, para que a contribuição desse novo polo e zero na constante de erro estática de velocidade seja o maior possível e esse controlador consiga atuar levando o lugar das raízes para a esquerda, o novo polo deve ser posicionado o mais perto possível de p1. Já o zero, para melhor contribuir com o lugar das raízes, deve ser o mais próximo possível de p2. Sendo assim, obtivemos o seguinte valor para  $z$ ,  $p$ ,  $\alpha$  e  $T$ .

$$\begin{aligned} z &= 0,0035 \\ p &= 0,0238 \\ \alpha &= 0,147 \\ T &= 285,714 \end{aligned} \tag{141}$$

Com esses valores, foi obtida a função de transferência exposta na Equação 142 para o compensador por avanço de fase utilizando o lugar das raízes.

$$G_c(s) = K_c \cdot \frac{s + 0,0035}{s + 0,0238} \tag{142}$$

Com o compensador obtido na Equação 142, foi obtido o seguinte gráfico de lugar das raízes, exposto na Figura 68.

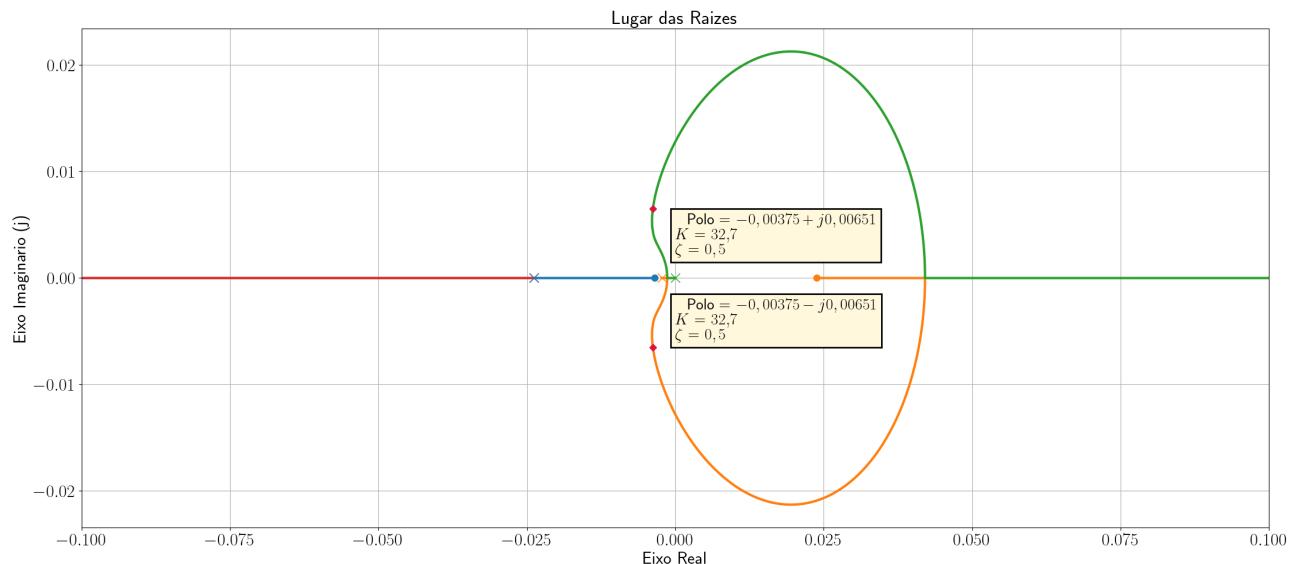


Figura 68: Root locus do sistema com compensador de avanço de fase.

Fonte: Autoria Própria.

Através da Figura 68, obtemos o valor de  $K_c$ , exposto na Equação 143.

$$K_c = 32,7 \tag{143}$$

Com isso, utilizando os valores de  $\alpha$  encontrado em Equação 141 e  $K_c$  em Equação 143, é possível encontrar o multiplicador da constante de erro estático de velocidade, exposto em Equação 144.

$$K_c \cdot \alpha = 32,7 \cdot 0,0147 = 4.8069 \quad (144)$$

Sendo assim, foi possível obter a constante de amortecimento  $\zeta = 0,5$  mas não foi possível obter uma constante de erro de velocidade  $K_v$  12x superior. Dessa maneira, o controlador não é capaz de atender os critérios impostos para o nosso sistema. A Equação 145 demonstra a função de transferência completa desse compensador.

$$G_c(s) = 32,7 \cdot \frac{s + 0,0035}{s + 0,0238} \quad (145)$$

### 9.2.2 Projeto através da Análise na Frequência

Para o projeto na frequência, queremos uma margem de fase  $\Phi_m > 50^\circ$  e a constante de erro estático de velocidade  $K_v$  pelo menos 10 vezes maior ao de malha aberta. O primeiro passo é olhar o diagrama de bode para o sistema não compensado, exposto na Figura 69.

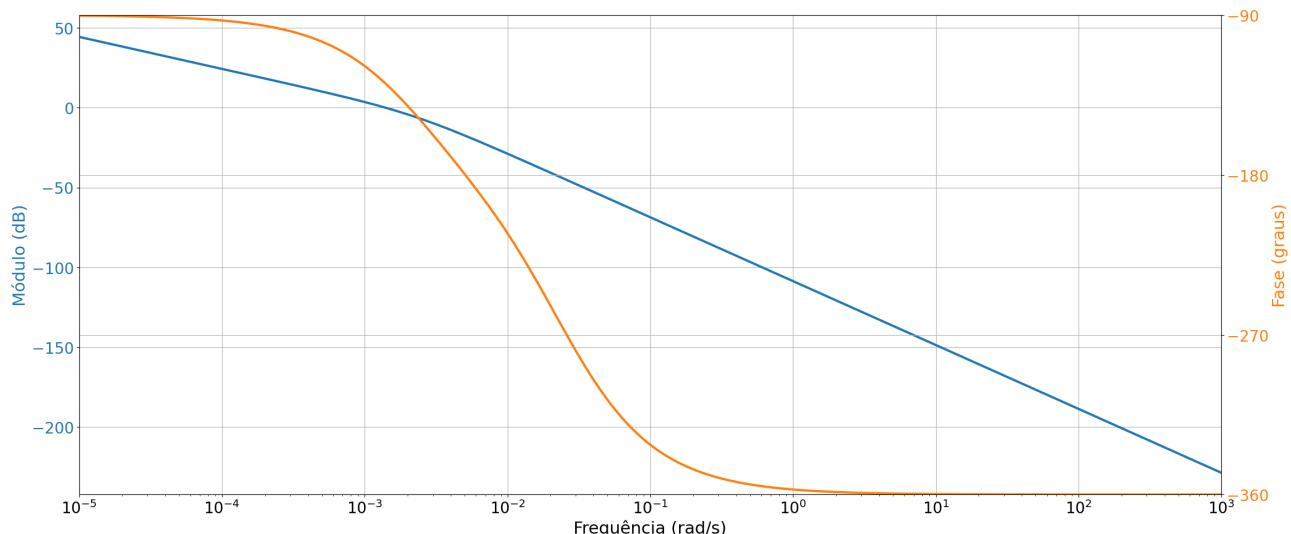


Figura 69: Diagrama de bode pro sistema não compensado.

Fonte: Autoria Própria.

Inicialmente, considerando a função de transferência planta conforme demonstrado na Equação 136, temos a seguinte constante de erro estático de velocidade e margem de fase do sistema, expostos na Equação 146.

$$\begin{aligned} K_v &= \lim_{s \rightarrow 0} s \cdot G(s) = 0,00163 \\ \Phi_m &= 51,47^\circ \end{aligned} \quad (146)$$

Para a análise na frequência, iremos considerar a seguinte função de transferência do compensador em avanço de fase, demonstrada em Equação 147.

$$G_c(s) = K_c \cdot \frac{Ts + 1}{\alpha Ts + 1} \quad (147)$$

Para que o sistema tenha uma constante de erro estático de velocidade 12x maior, precisamos que  $K_c = 12$ . Com isso, achamos o valor em dB de  $K_c$  em Equação 148.

$$|K_c|_{dB} = 20 \cdot \log(K_c) = 21,583 \quad (148)$$

O próximo passo é achar a frequência tal que  $|G(j\omega_i)|_{dB} = -|K_c|_{dB}$ , que irá ser a nova frequência de cruzamento de fase após a inclusão do compensador. Essa frequência ( $\omega_i$ ) está definida na Equação 149.

$$\begin{aligned} |G(j\omega_i)|_{dB} &= -21,583 \\ \omega_i &= 6,47 \times 10^{-3} \text{ rad s}^{-1} \end{aligned} \quad (149)$$

Com isso, foi encontrada a fase nessa frequência, demonstra na Equação 150.

$$\angle G(j\omega_i) = -191,18^\circ \quad (150)$$

Para que  $\Phi_m > 50^\circ$ , precisamos que em  $\omega_i$  o sistema tenha  $\angle G(j\omega_i) = -130^\circ$ . Sendo assim, precisamos que o nosso controlador tenha uma contribuição conforme a Equação 151.

$$-130^\circ = -191,18^\circ + \angle G_c(j\omega_i) \quad (151)$$

Considerando uma margem de segurança de  $5^\circ$ , encontramos  $\angle G_c(j\omega_i)$  em Equação 152.

$$\angle G_c(j\omega_i) = 66,18^\circ = \Phi \quad (152)$$

Utilizando o valor encontrado de  $\Phi$  em Equação 152 e em Equação 153, encontramos o valor de  $\alpha$  em Equação 154.

$$\alpha = \frac{1 - \sin \Phi}{1 + \sin \Phi} \quad (153)$$

$$\alpha = 0,044 \quad (154)$$

A seguir, iremos calcular o ganho do controlador em uma frequência  $\omega_n$ . Nessa frequência, temos o ganho dado pela Equação 156.

$$|G_c(j\omega_n)|_{dB} = 20 \cdot \log \frac{K_c}{\sqrt{a}} \quad (155)$$

$$|G_c(j\omega_n)|_{dB} = 35,149 \quad (156)$$

Sendo assim, para que  $\omega_n$  seja a nossa frequência de cruzamento, temos que  $\omega_n$  tem que ser tal que  $|G(j\omega_n)|_{dB} = -35,149$ . Com isso, encontramos  $\omega_n$ .

$$\omega_n = 1,438 \times 10^{-2} \text{ rad s}^{-1} \quad (157)$$

Com  $\omega_n$  definido, podemos encontrar  $T$  através da Equação 158. O valor de  $T$  está exposto na Equação 159.

$$T = \frac{1}{\sqrt{\alpha}\omega_n} \quad (158)$$

$$T = 329,710 \quad (159)$$

Com isso, encontramos a função de transferência do compensador por atraso de fase, demonstrada em Equação 160.

$$G_c(s) = 12 \cdot \frac{329,710s + 1}{14,667s + 1} \quad (160)$$

A Figura 70 demonstra o diagrama de bode do sistema com o compensador por avanço de fase.

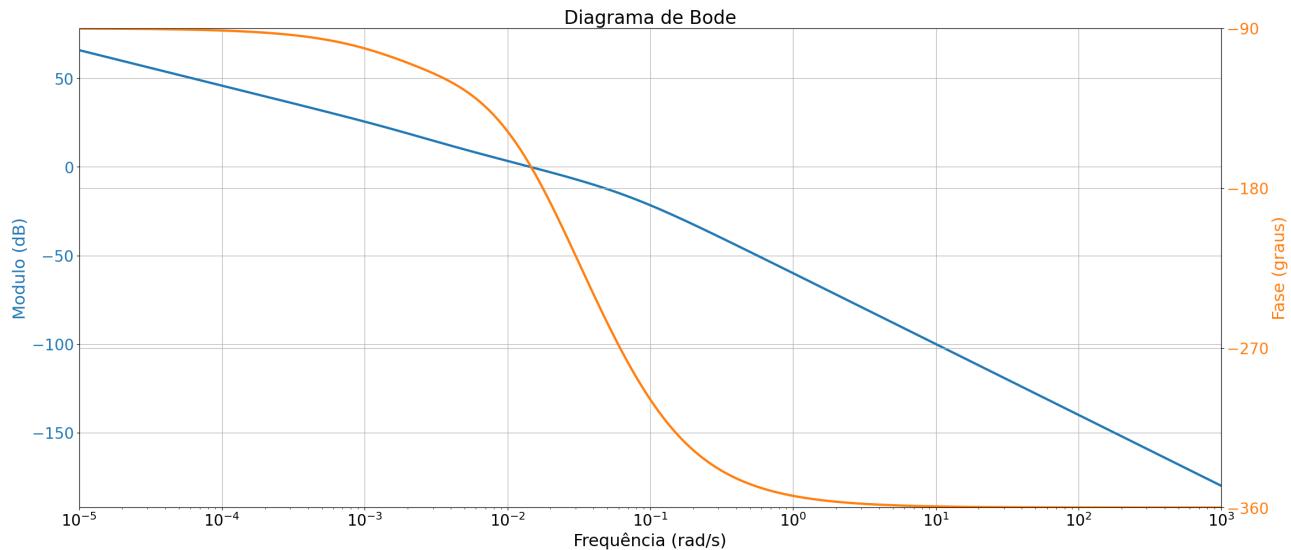


Figura 70: Diagrama de bode do sistema com compensador por avanço de fase.

Fonte: Autoria Própria.

\*\*\*

Com os controladores obtidos pelo método do lugar das raízes e pela análise na frequência, foram plotados os gráficos da resposta ao degrau e resposta a rampa do sistema com integrador e compensado por avanço de fase.

A Figura 71 demonstra a resposta ao degrau dos sistemas com compensadores projetados através do Root Locus e através do Diagrama de Bode. É possível perceber que o sistema agora é do tipo 1, sendo assim, o erro em regime permanente para uma entrada ao degrau é nulo. Em comparação com o sistema com atraso de fase, a resposta com o compensador por avanço é mais rápida, porém esse controlador não é o ideal para reduzir a constante de erro estático de velocidade.

A Figura 72 demonstra a resposta a rampa dos sistemas com compensadores projetados por Root Locus e através do Diagrama de Bode. É possível perceber que o sistema agora é do tipo 1, sendo assim, o sistema consegue acompanhar uma referência do tipo rampa, porém existe um erro em regime permanente.

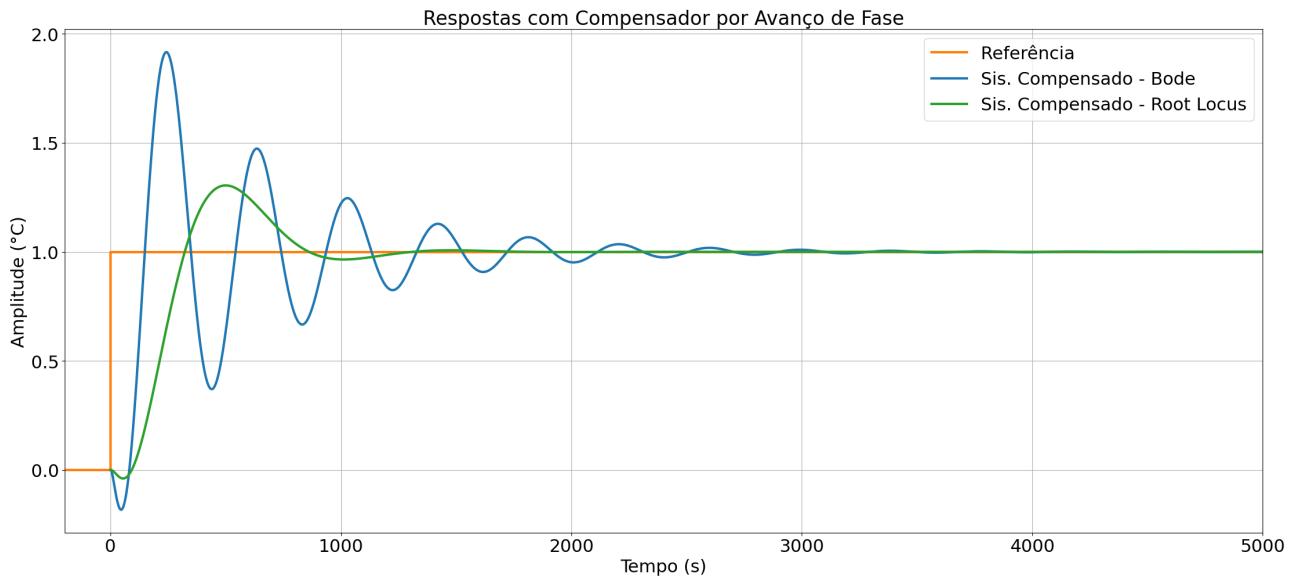


Figura 71: Resposta ao degrau com os compensadores por avanço de fase.

Fonte: Autoria Própria.

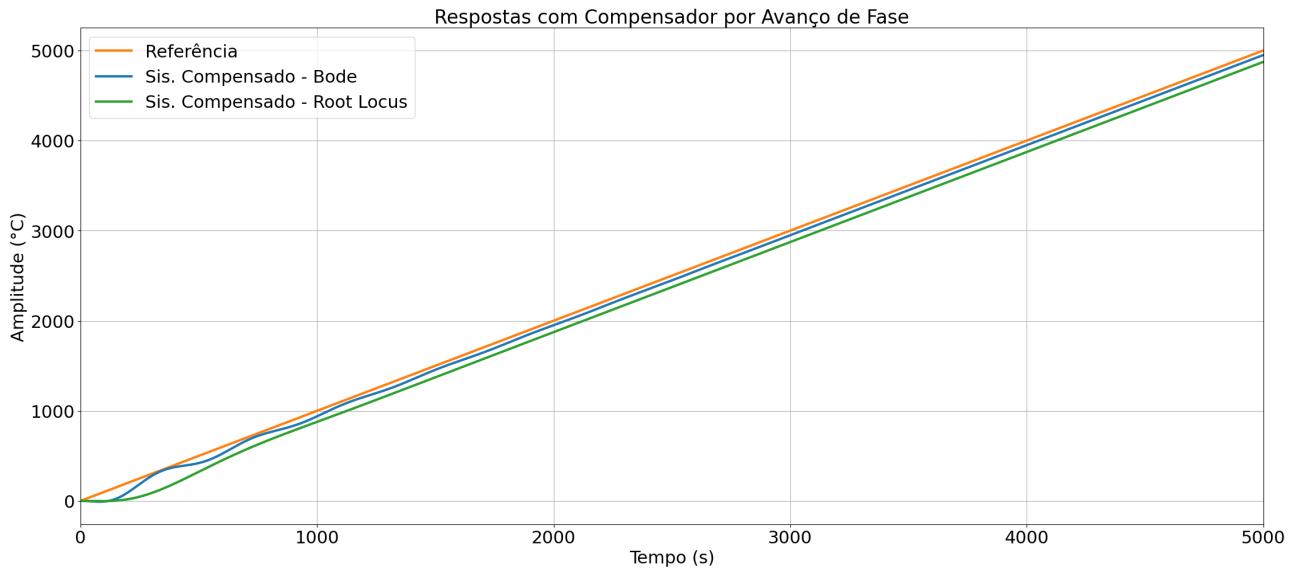


Figura 72: Resposta a rampa com os compensadores por avanço de fase.

Fonte: Autoria Própria.

## 10 Controlador em Espaço de Estados

Nessa seção será feito um controlador utilizando o espaço de estado através do método de alocação de polos, de forma que o tempo de acomodação seja compatível com o melhor dos controladores já projetados e que a resposta ao degrau do sistema não possua erro de regime permanente. considerando a planta definida em Equação 19. A principio, foram substituídos

os valores de  $\theta$ ,  $\tau$  e  $K$  nas matrizes, demonstrado na Equação 161.

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -5,374e-05 & -0,02601 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1,13 \end{bmatrix} \cdot q_i(t) \\ y &= [0,0238 \quad -1] \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \end{aligned} \quad (161)$$

## 10.1 Projeto por Alocação de Polos

Para a estrutura de controle em espaço de estados, iremos considerar o diagrama de blocos da Figura 73. Nesse diagrama estamos medindo todos os estados e aplicando um ganho para cada. A saída do bloco de ganho somado com o sinal de referência multiplicada por um ganho, que pondera a referência, forma o sinal de controle, conforme a Equação 162 [Cardoso, 2020].

$$u = -[k_1 \quad k_2] \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + k_r \cdot r \quad (162)$$

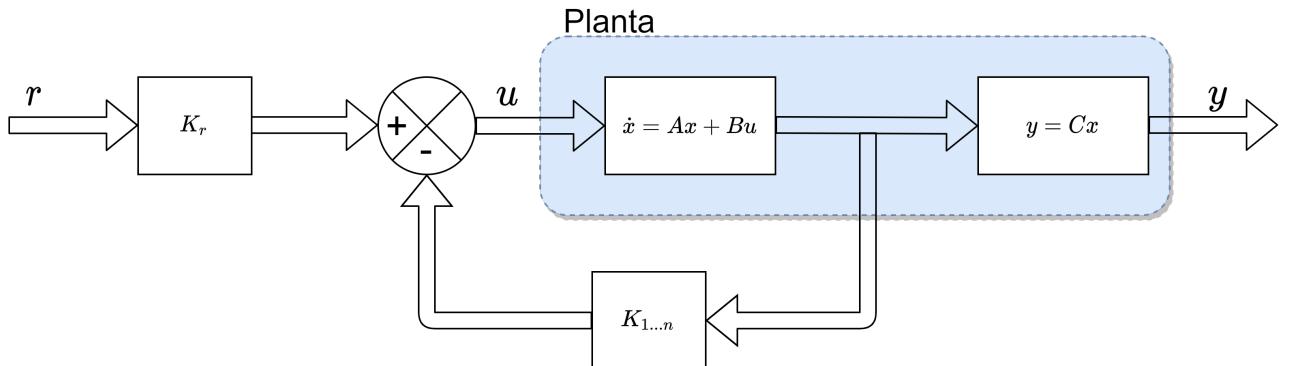


Figura 73: Diagrama de blocos para o Espaço de Estados

Fonte: Adaptado de [Cardoso, 2020].

Para simplificar, iremos considerar a matriz de ganhos de  $\mathbf{K}$ , conforme a Equação 163.

$$\mathbf{K} = [k_1 \quad k_2] \quad (163)$$

Com isso, obtém-se a equação de controle  $u$  conforme demonstrado na Equação 164.

$$u = -\mathbf{K} \cdot \mathbf{x} + k_r \cdot r \quad (164)$$

Substituindo a lei de controle na equação dinâmica do sistema, obtemos a representação do sistema em malha fechada, conforme demonstrado na Equação 165.

$$\begin{aligned} \dot{\mathbf{x}} &= (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x} + \mathbf{B}k_r \cdot r \\ y &= \mathbf{C}\mathbf{x} \end{aligned} \quad (165)$$

Na representação descrita na Equação 165, temos as seguintes interpretações:

- Autovalores de  $(\mathbf{A} - \mathbf{B}\mathbf{K}) \Rightarrow$  Polos de malha fechada do sistema.
- Determinação de  $\mathbf{K} \Rightarrow$  Através do método de alocação de polos.

- Determinação de  $k_r \Rightarrow$  Inverso do ganho CC do sistema em malha fechada para  $k_r = 1$ .

Inicialmente, temos que os autovalores de  $\mathbf{A}$  representam os polos de malha aberta do sistema, que devem ser os mesmos demonstrado na Figura 9. Com isso, encontramos os polos em Equação 166.

$$\begin{aligned} p_1 &= -0,0023 + j0 \\ p_2 &= -0,0238 + j0 \end{aligned} \quad (166)$$

A fim de comparação, foi encontrada o sistema sem compensador em malha fechada, que é dado através das equações demonstradas em Equação 167. Com isso, obtemos a seguinte representação, exposta em Equação 168.

$$\begin{cases} \dot{\mathbf{x}} = (\mathbf{A}_{mf} - \mathbf{B}_{mf}\mathbf{C}_{mf})\mathbf{x} + \mathbf{B}_{mf} \cdot r \\ y = \mathbf{C}_{mf}\mathbf{x} \end{cases} \quad (167)$$

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -0,0270 & 1,1039 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1,13 \end{bmatrix} \cdot q_i(t) \\ y &= [0,0238 \quad -1] \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \end{aligned} \quad (168)$$

Para o sistema exposto em Equação 168, temos os autovalores de  $\mathbf{A}_{mf}$  demonstrados em Equação 169, que, por serem positivos, demonstram um sistema instável, sendo necessário então um controlador que force o sistema a ter o comportamento desejado e o leve a estabilidade.

$$\begin{aligned} p_1 &= 0,0250 + j0 \\ p_2 &= 1,0789 + j0 \end{aligned} \quad (169)$$

Para o controle pela alocação de polos, iremos escolher polos que antedem nossos critérios. Sendo assim, para fins de comparação, iremos escolher polos de malha fechada bem próximos ao obtidos para o **controlador PI**. Os polos de malha fechada escolhidos estão expostos na Equação 170.

$$\begin{aligned} p_1 &= -0,01 + j0,0130 \\ p_2 &= -0,01 - j0,0130 \end{aligned} \quad (170)$$

Com os polos de malha fechada desejados definidos em Equação 170, é possível montar o polinômio característico, conforme demonstrado nas equações Equação 171 e Equação 172.

$$\phi(s) = (s - p_1) \cdot (s - p_2) \quad (171)$$

$$\phi(s) = s^2 + 0,02s + 0,000269 \quad (172)$$

Para achar a matriz utilizada na **Formula de Ackermann**, foi substituindo  $s$  pela matriz  $\mathbf{A}$ . Com isso, encontramos a equação descrita na Equação 173.

$$\phi(A) = \begin{bmatrix} 0 & 1 \\ -5,374e-05 & -0,02601 \end{bmatrix} - 0,02 \cdot \begin{bmatrix} 0 & 1 \\ -5,374e-05 & -0,02601 \end{bmatrix} + 0,000269 \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (173)$$

Com a Equação 173 definida, iremos aplicá-la na **Formula de Ackermann** para encontrar os valores dos ganhos  $\mathbf{K}$ . A **Formula de Ackermann** está exposta na Equação 174 e os ganhos  $k_1$  e  $k_2$  na Equação 175.

$$\mathbf{K} = [0 \ 1] [\mathbf{B} \ \mathbf{AB}]^{-1} \cdot \phi(A) \quad (174)$$

$\mathbf{K} = [k_1 \ k_2]$	
$k_1 = 0,0001904$	
$k_2 = -0.0053689$	

(175)

Com o valor dos ganhos já definido em Equação 175, é necessário ainda definir o ganho de entrada  $k_r$ . Para isso, iremos voltar na Equação 165 substituindo  $\mathbf{K}$  e considerando, inicialmente,  $k_r = 1$ . Dessa forma, foi obtida a seguinte representação no espaço de estados, demonstrado na Equação 176.

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -0,000269 & -0,0200 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1,13 \end{bmatrix} \cdot q_i(t) \\ y &= [0,0238 \ -1] \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \end{aligned} \quad (176)$$

Com a matriz  $\mathbf{A}$  obtida na Equação 176 é possível encontrar a função de transferência do sistema em malha fechada e, substituindo  $s = 0$ , encontramos o valor do ganho CC do sistema, conforme demonstrado nas equações 177 e 178.

$$G_{CC} = \lim_{s \rightarrow 0} G_{MFC}(s) = \lim_{s \rightarrow 0} 1,13 \cdot \frac{-s + 0,0238}{s^2 + 0,02s + 0,000269} \quad (177)$$

$$G_{CC} = 99,977 \quad (178)$$

O valor de  $k_r$  é dado pelo inverso de  $G_{CC}$ . Sendo assim, é possível obter o valor de  $k_r$  através da Equação 179. O valor de  $K_r$  está definido em Equação 180.

$$k_r = \frac{1}{G_{CC}} \quad (179)$$

$k_r = 0,0100$	
----------------	--

(180)

Com os valores de  $K_r$  e  $\mathbf{K}$  definidos, finalmente temos a representação do sistema compensado em espaço de estados, demonstrado na Equação 181.

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -0,000269 & -0,0200 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0,011298 \end{bmatrix} \cdot q_i(t) \\ y &= [0,0238 \ -1] \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \end{aligned} \quad (181)$$

## 10.2 Análise dos Resultados

Para a análise dos resultados, foi plotada a resposta ao degrau utilizando a representação em espaço de estados com  $k_r = 1$  e  $k_r = 0,0100$ , demonstradas, respectivamente, pelas Figuras 74 e 75.

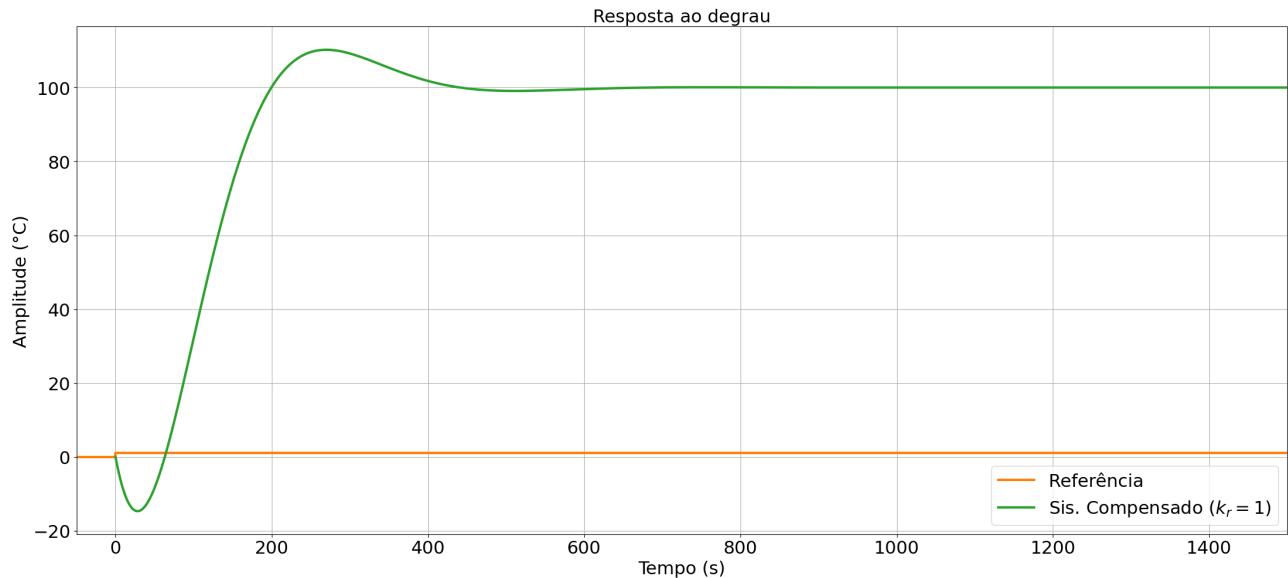


Figura 74: Resposta ao degrau com Alocação de Polos e  $k_r = 1$ .

Fonte: Autoria Própria.

A Figura 74 demonstra o sistema compensado por alocação de polos e  $k_r = 1$ . Nele é possível observar que o sistema se comporta de maneira rápida, porém, devido ao alto ganho de entrada, é obtido um erro expressivo em estado permanente. É possível obter uma saída sem erro através do ajuste do ganho  $k_r$ , como o que foi feito para o controle utilizado na Figura 75.

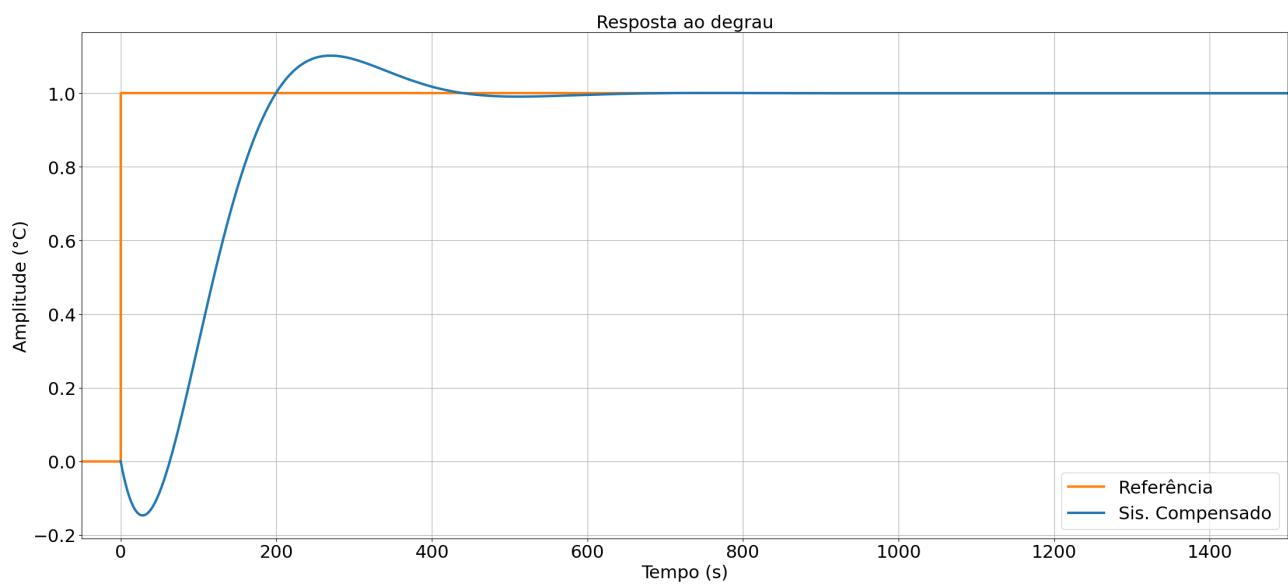


Figura 75: Resposta ao degrau com Alocação de Polos e  $k_r = 0,01$ .

Fonte: Autoria Própria.

A Figura 75 demonstra o sistema compensado por alocação de polos e  $k_r = 0,01$ . Esse sistema apresenta um regime transitório mais rápido em comparação com os outros controladores e compensadores já projetados nesse trabalho, e possui um erro de estado estacionário nulo. A Figura 76 apresenta a saída ao degrau do sistema utilizando o controlador PI e o controlador por alocação de polos.

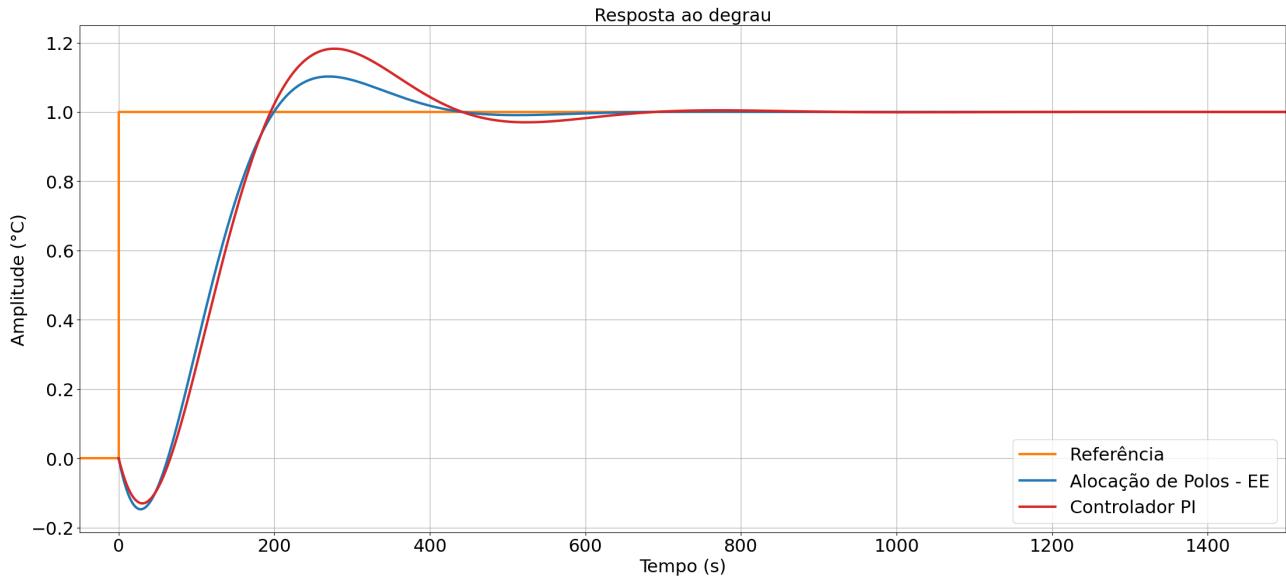


Figura 76: Comparação entre os controlador PI e por Alocação de Polos.

Fonte: Autoria Própria.

Através da Figura 76, é possível perceber que o sistema compensado utilizando a análise por espaço de estados apresenta um melhor resultado, obtendo um tempo de acomodação menor e menos *overshoot*. A Equação 182 apresenta o tempo de acomodação para 2%  $t_a$  e o máximo *overshoot*  $M_p$  para o sistema por alocação de polos.

$$\begin{aligned} T_a &\approx 389 \text{ s} \\ M_p &\approx 10,25\% \end{aligned} \tag{182}$$

A Tabela 4 apresenta os valores de  $t_a$  e  $M_p$  para o controlador PI e pelo controle feito pelo Método de Alocação de Polos. Através dela, é possível ver que o controlador através do espaço de estados apresenta melhor desempenho.

	Controlador PI	Espaço de Estados
$T_a$	591 s	389 s
$M_p$	18,3%	10,25%

Tabela 4: Comparação entre o Controlador PI e Alocação de Polos.

Infelizmente, esse tipo de controlador é impossível de ser aplicado na prática na nossa planta, uma vez que um dos nossos estados corresponde ao atraso de transporte, impossível de ser medido através de um sensor ou controlado. Sendo assim, apesar desse controle oferecer melhores resultados e ser matematicamente utilizável, esse controle não é aplicável ao nosso problema real.

## 11 PID com Predictor de Smith

Nessa seção, será feito o projeto final do controlador para o forno. Conforme demonstrado na Figura 3, a temperatura ideal para o forno de solda SMD deve seguir referências do tipo rampa. Dessa forma, o controlador necessita seguir uma referência em rampa sem erro ou com erro limitado. Para simplificar os cálculos, foi feito um controlador PID, que torna o sistema tipo 1. Como em rampa o erro pode ser visto como um atraso, será feita a compensação desse erro aumentando o tempo da referência para cada etapa. Além disso, conforme demonstrado na Equação 29, a planta contém um atraso de propagação, sendo necessário então o uso da técnica de Predictor de Smith. Sendo assim, será feito o projeto de um **Controle PID com Predictor de Smith**. Na Subseção 11.1 será apresentado o método de controle por Predictor de Smith.

### 11.1 Predictor de Smith

O Predictor de Smith é um esquema projetado por Otto J.M. Smith para controlar plantas com tempo morto. Inicialmente, iremos considerar a planta demonstrada pela Figura 77. Essa planta envolve um tempo morto de  $T$  segundos e um controlador  $G_c(s)$ , que é o controlador com Predictor de Smith.

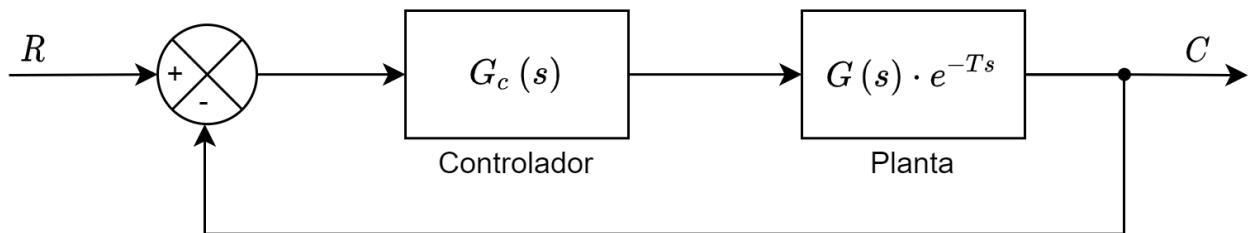


Figura 77: Sistema de controle com tempo morto.

Fonte: Adaptado de [Ogata, 2003].

O bloco do controlador  $G_c(s)$  é demonstrado na Equação 183, em que  $\hat{G}_c(s)$  é o controlador projetado para atender as especificações de projeto, considerando o sistema sem o atraso do tempo.

$$G_c(s) = \frac{\hat{G}_c(s)}{1 + (1 - e^{-Ts}) \hat{G}_c(s) G(s)} \quad (183)$$

A Equação 183 demonstra a função de transferência do bloco Predictor de Smith. Esse função possui uma similaridade com uma equação de um sistema realimentado. Sendo assim, é possível modificar o diagrama de blocos da Figura 77 para que o bloco do controlador seja uma realimentação, conforme demonstrado em Figura 78.

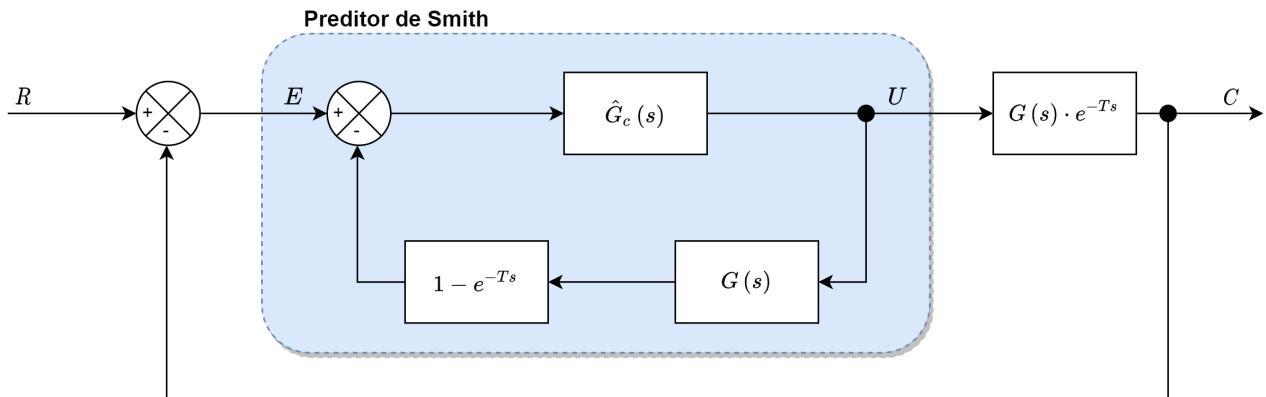


Figura 78: Sistema de controle com Predictor de Smith.

Fonte: Adaptado de [Ogata, 2003].

A função de transferência da realimentação interna  $U(s)/E(s)$  é dado pelo Predictor, e deve ser igual a função demonstrada em Equação 183. Essa função pode ser vista na Equação 184.

$$\frac{U(s)}{E(s)} = \frac{\hat{G}_c(s)}{1 + (1 - e^{-Ts}) \hat{G}_c(s) G(s)} \quad (184)$$

Sendo assim, podemos encontrar a função de transferência do sistema  $C(s)/R(s)$ , conforme demonstrada na Equação 185. A função de transferência encontrada está exposta na Equação 186.

$$\frac{C(s)}{R(s)} = \frac{\hat{G}_c(s) \cdot G(s) \cdot e^{-Ts}}{1 + (1 - e^{-Ts}) \hat{G}_c(s) G(s) + \hat{G}_c(s) G(s) e^{-Ts}} \quad (185)$$

$$\boxed{\frac{C(s)}{R(s)} = \frac{\hat{G}_c(s) \cdot G(s)}{1 + \hat{G}_c(s) \cdot G(s)} \cdot e^{-Ts}} \quad (186)$$

Dessa forma, é possível simplificar o diagrama de blocos, conforme visto na Figura 79, a onde temos o atraso de transporte fora da malha de realimentação, o que faz com que o sistema se comporte igualmente a um sistema sem atraso deslocado no tempo. Essa técnica é interessante pois facilita o projeto e sintonia do controlador, uma vez que este pode ser feito considerando o sistema sem o atraso de transporte.

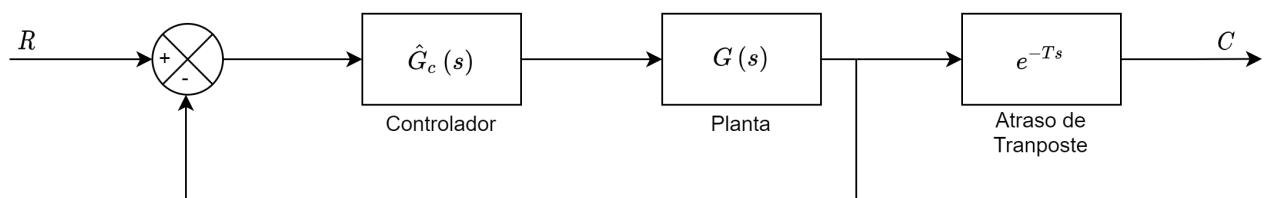


Figura 79: Diagrama de blocos equivalente ao sistema com o Predictor de Smith.

Fonte: Adaptado de [Ogata, 2003].

Sendo assim, na Subseção 11.2 será projetado o controlador PID que atenda as especificações do sistema, para que seja incluído no controlador com Predictor de Smith.

## 11.2 Controlador PID

O controlador PID é feito pela junção dos controladores proporcional, integral e derivativo, podendo ser exposto através da forma de diagrama de blocos, conforme demonstrado na Figura 80. A função de transferência desse controlador pode ser vista na Equação 187.

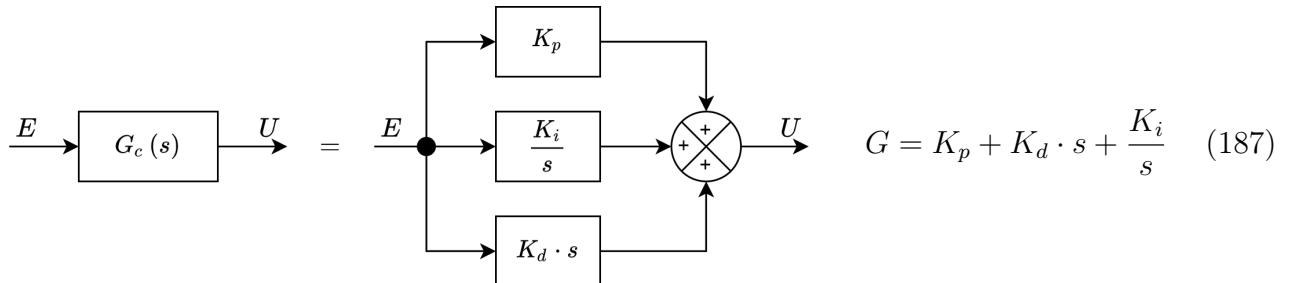


Figura 80: Diagrama de blocos para o controlador PID.

Fonte: Autoria Própria.

Nessa subseção serão projetados controladores PID através dos métodos listados abaixo e, logo após, comparados de forma a se utilizar o melhor controlador para a nossa aplicação.

- **Ziegler-Nichols.**
- **ITAE.**
- Utilizando a ferramenta **pidtuner**, do *MATLAB*.

### 11.2.1 Ziegler-Nichols

Nessa subseção será abordado primeiro método de sintonia de controladores proposto por Ziegler e Nichols. Esse método consiste em, através dos parâmetros de uma planta de primeira ordem com atraso (ou que pode ser modelado dessa forma), obter os valores para os parâmetros do controlador PID. Inicialmente, a função de transferência do controlador, demonstrada em Equação 187, foi reescrita, conforme vista em Equação 188 [Castaño, 2019].

$$G_c(s) = K_p \left( 1 + \frac{1}{T_i \cdot s} + T_d \cdot s \right) \quad (188)$$

Antes de fazer a sintonia, foi calculado o fator de controlabilidade da planta pelo método de Ziegler-Nichols. Esse fator pode ser calculado através da Equação 189.

$$F_c = \frac{\theta}{\tau} \quad (189)$$

Para que o sistema seja bem controlado pelo método de Ziegler-Nichols é necessário que o fator de controlabilidade atenda os critérios expostos em Equação 190. Como o sistema se mostra controlável, foi prosseguido a sintonia do controlador PID.

$$0,1 \leq \underbrace{0,189}_{F_c} \leq 0,3 \quad \Leftrightarrow \text{Sistema controlável por Ziegler-Nichols.} \quad (190)$$

Com a função de transferência escrita na Equação 188, é possível encontrar os valores dos parâmetros  $K_p$ ,  $T_i$  e  $T_d$  através dos parâmetros da nossa planta e das equações obtidas através da tabela exposta em Tabela 5.

Controlador	$K_p$	$T_i$	$T_d$
P	$\frac{\tau}{K \cdot \theta}$	$\infty$	0
PI	$0,9 \frac{\tau}{K \cdot \theta}$	$\frac{L}{0,3}$	0
PID	$1,2 \frac{\tau}{K \cdot \theta}$	$2 \cdot \theta$	$0,5 \cdot \theta$

Tabela 5: Tabela com parâmetros para o primeiro método de Ziegler-Nichols.

Dessa forma, foram encontrados os valores dos parâmetros, expostos na Equação 191. A função de transferência do controlador, oferecido originalmente na forma da Equação 187, está exposta em Equação 192.

$$\begin{aligned} K_p &= 5.600 \\ T_i &= 168 \\ T_d &= 42.0 \end{aligned} \quad (191)$$

$$G_c(s) = 5,600 + 235,221 \cdot s + \frac{0.0333}{s} \quad (192)$$

### 11.2.2 ITAE

O método ITAE utiliza os critérios de desempenho da integral do erro por um período de tempo suficiente para poder eliminar o erro em estado estacionário. Esse método utiliza a integral do erro de acordo com o tempo, sendo erros em tempo mais alto mais penalizados. A Equação 193 demonstra esse comportamento [Castaño, 2021].

$$ITAE = \int_0^T t \cdot |e(t)| dt \quad (193)$$

Esse método possui duas formas, sendo a forma de **Lopez** para perturbações de carga (problema regulatório) e a forma de **Rovira** para seguir referências (servossistemas). Como foram desconsideradas os efeitos de distúrbios no sistema e é necessário que ele siga uma referência, foi utilizada a **forma de Rovira**.

A princípio, foi reescrita a função de transferência do controlador, exposta na Equação 187, para a forma conforme demonstra na Equação 194.

$$G_c(s) = K_p \left( 1 + \frac{1}{T_i \cdot s} + T_d \cdot s \right) \quad (194)$$

Para utilizar o método ITAE, é preciso ter um sistema de primeira ordem com atraso de propagação. Esse é o caso exposto nesse relatório, e cada parâmetro da planta está exposto na Equação 21. O método dispõe de equações para definir os valores dos parâmetros  $K_p$ ,  $T_i$  e  $T_d$ , conforme demonstrada em Equação 195 e uma tabela com valores de parâmetros  $A$ ,  $B$ ,  $C$ ,  $D$ ,  $E$  e  $F$ , expostos na Tabela 6.

$$K_p = \frac{1}{K} \left( A \cdot \left( \frac{\theta}{\tau} \right)^B \right) \quad T_i = \frac{\tau}{(C + D \cdot (\frac{\theta}{\tau}))} \quad T_d = \tau \cdot \left( E \left( \frac{\theta}{\tau} \right)^F \right) \quad (195)$$

Controlador	Critério	$A$	$B$	$C$	$D$	$E$	$F$
PI	IAE	0,758	-0,861	1,02	-0,323	-	-
PI	ITAE	0,586	-0,916	1,03	-0,165	-	-
PID	IAE	1,086	-0,869	0,740	-0,130	0,348	0,914
PID	ITAE	0,965	-0,850	0,796	-0,147	0,308	0,929

Tabela 6: Tabela com parâmetros para o método ITAE.

Utilizando as formulas demonstradas em Equação 195 e o valor dos parâmetros em Tabela 6, é possível encontrar os valores de  $K_p$ ,  $T_i$  e  $T_d$ . Esses valores estão expressos na Equação 196. A função de transferência do controlador, oferecido originalmente na forma Equação 187, está exposta em Equação 197.

$$\begin{aligned} K_p &= 3,509 \\ T_i &= 576,727 \\ T_d &= 29,113 \end{aligned} \quad (196)$$

$$G_c(s) = 3,509 + 102,177 \cdot s + \frac{0,00608}{s}$$

(197)

### 11.2.3 pidtuner

O **pidtuner** é uma ferramenta do *software MATLAB* que é capaz de projetar controladores de diversos tipos seguindo características de velocidade e robustez definidas pelo projetista. Através de testes heurísticos, foram obtidos os melhores resultados utilizando um controlador PID que contem os parâmetros definidos pela Equação 198.

$$\begin{aligned} K_p &= 14,5096 \\ K_i &= 0,15743 \\ K_d &= 0 \end{aligned} \quad (198)$$

Com os parâmetros demonstrado em Equação 198, percebe-se que os testes resultaram em um controlador PI. Esse resultado será analisado na próxima subsubseção, em que será feita a comparação entre todos os controladores obtidos. A função de transferência da planta para o controlador obtido pelo pidtuner pode ser vista na Equação 199.

$$G_c(s) = 14,5096 + \frac{0,15743}{s}$$

(199)

### 11.2.4 Análise dos Controladores

Nessa subsubseção será feita a análise de resultado dos controladores obtidos. Para isso, foi feita a simulação de uma entrada de degrau no sistema com os controladores e com a planta sem o atraso de transporte, conforme demonstrado no diagrama de blocos da Figura 81.

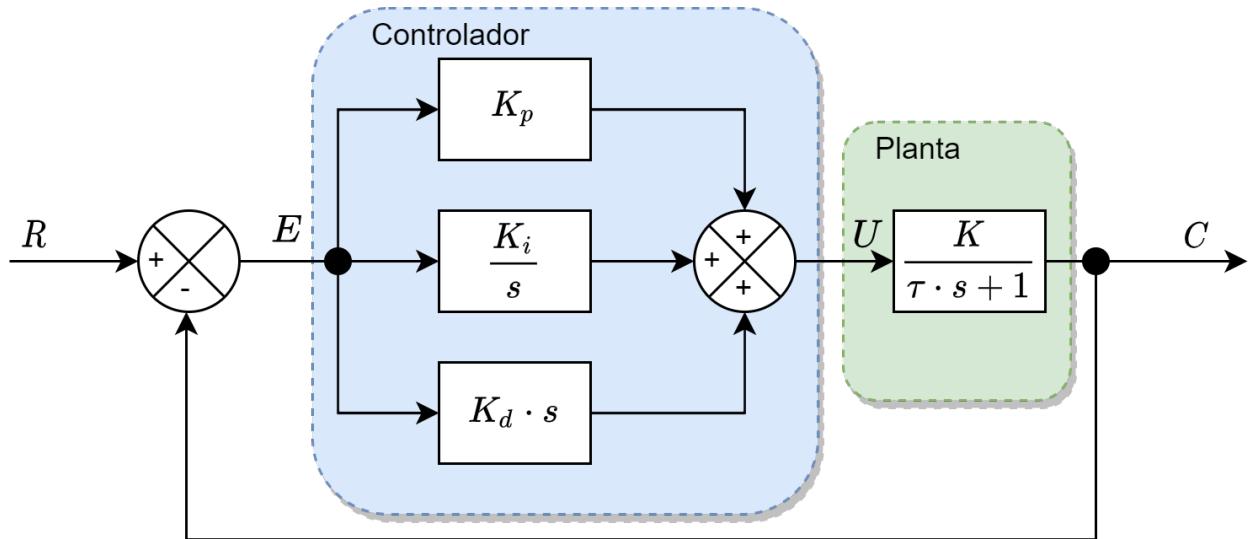


Figura 81: Diagrama de blocos para o controlador PID com planta sem atraso.

Fonte: Autoria Própria.

Com o diagrama de blocos da Figura 81 e os parâmetros pro controlador obtidos em Subsubseção 11.2.1, Subsubseção 11.2.2 e Subsubseção 11.2.3, foi conferida a resposta ao degrau, conforme vista na Figura 82.

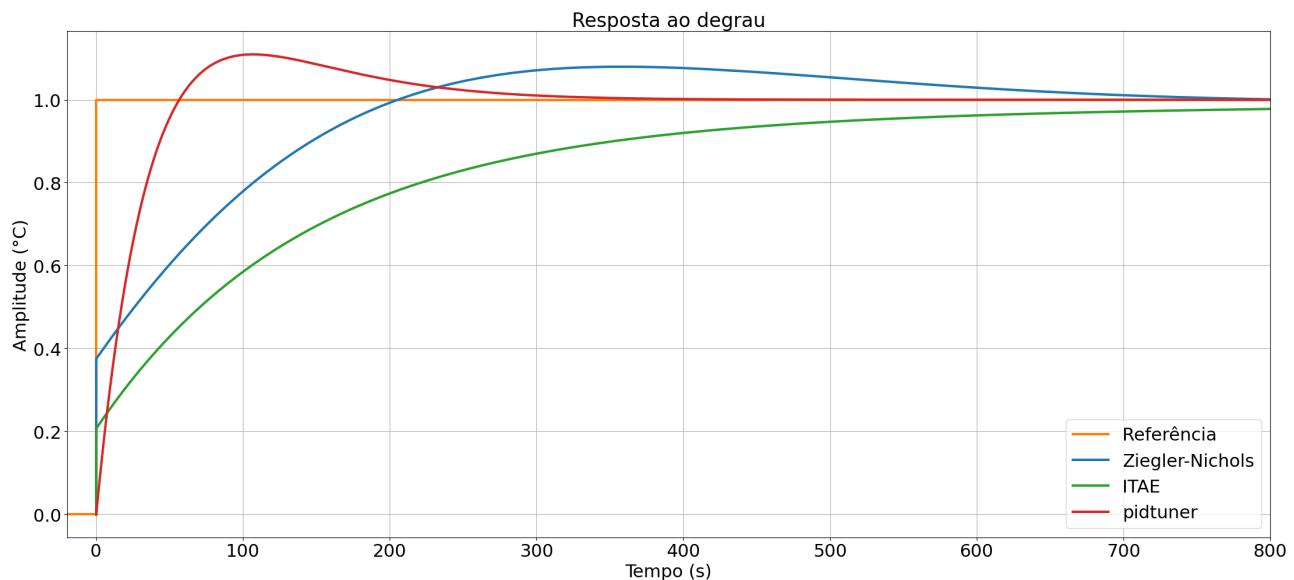


Figura 82: Resposta ao degrau do sistema com controladores PID e PI.

Fonte: Autoria Própria.

Conforme demonstrado na Figura 82, é possível perceber que o controlador sintonizado pelo **pidtuner** apresenta melhor resposta para o sistema, com um *overshoot* aceitável e melhor tempo de acomodação. Sendo assim, esse controlador será utilizado no sistema controlado com o Predictor de Smith.

\*\*\*

Com o controlador projetado e sintonizado, é obtido o diagrama de blocos para o sistema com o Predictor de Smith, exposto na Figura 83.

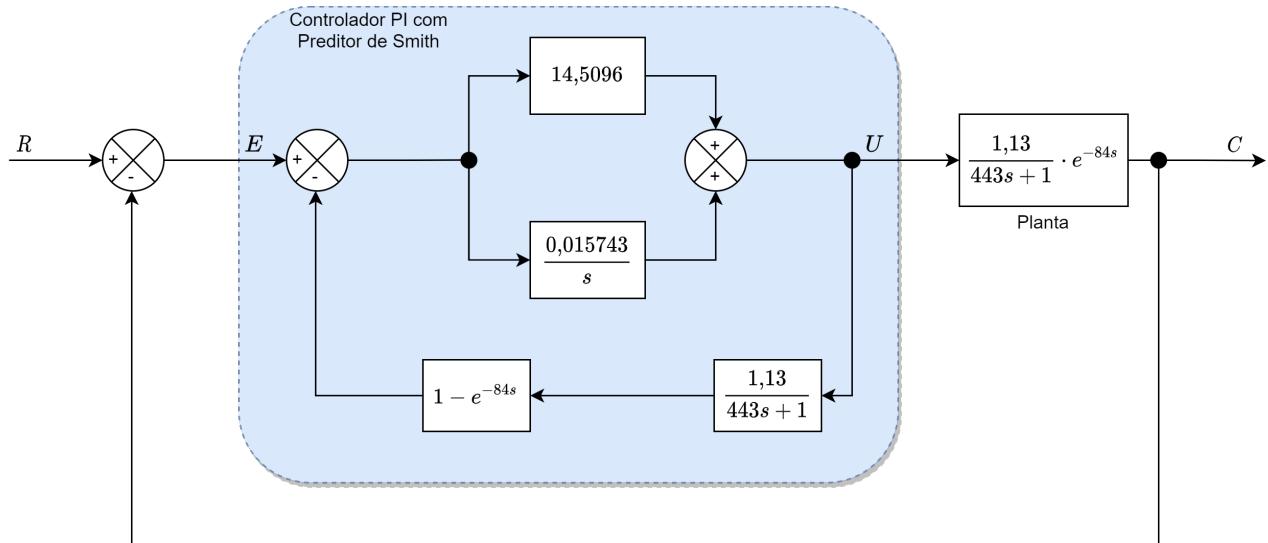


Figura 83: Diagrama de blocos do controlador PI sintonizado e com Predictor de Smith.

Fonte: Autoria Própria.

A resposta do sistema ao degrau, exposto pelo diagrama de blocos da Figura 83, pode ser visto na Figura 84, em que é possível perceber que o sistema de comporta exatamente igual ao projeto, demonstrado na Figura 82, deslocado 84s no tempo.

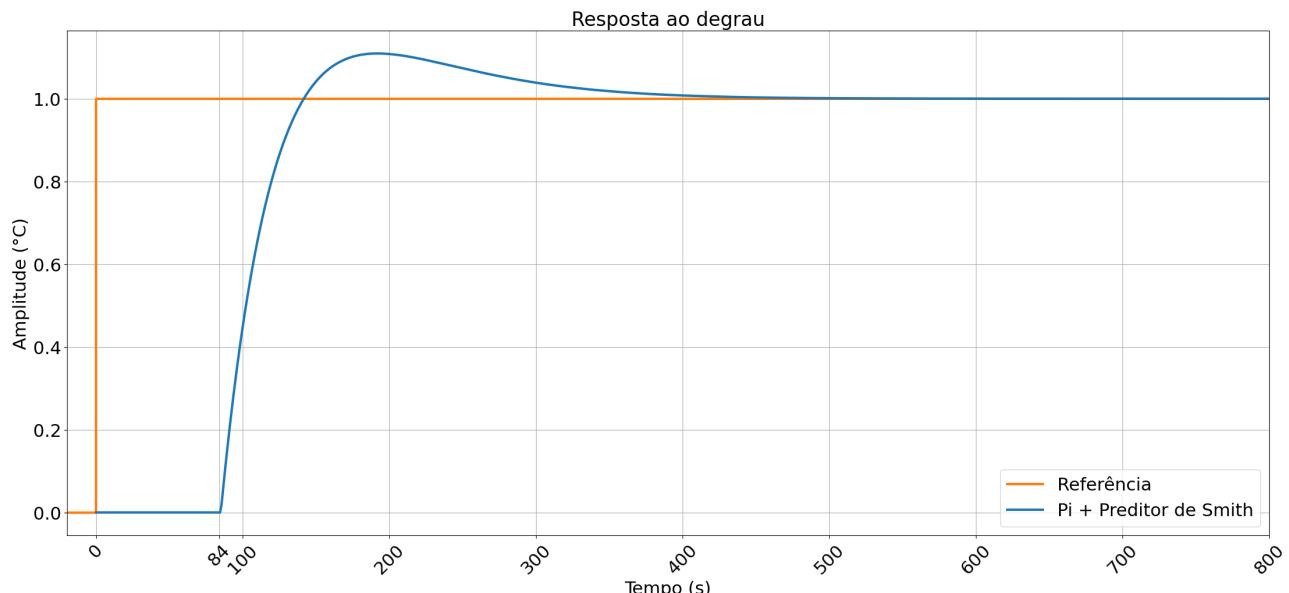


Figura 84: Resposta ao degrau do sistema com controlador PI e Predictor de Smith.

Fonte: Autoria Própria.

Com o controlador projetado e testado, é possível fazer o controle de temperatura do forno de maneira desejada, conforme demonstrado na curva de aquecimento da Figura 3. A Figura 85 apresenta o comportamento do forno, em que em laranja temos a referência desejada, que foi levemente modificada da original, e em azul a temperatura real do forno. É possível notar que o forno consegue atingir as temperaturas desejadas em cada etapa, sendo assim, o controlador projetado atende a necessidade do projeto e pode ser utilizado.

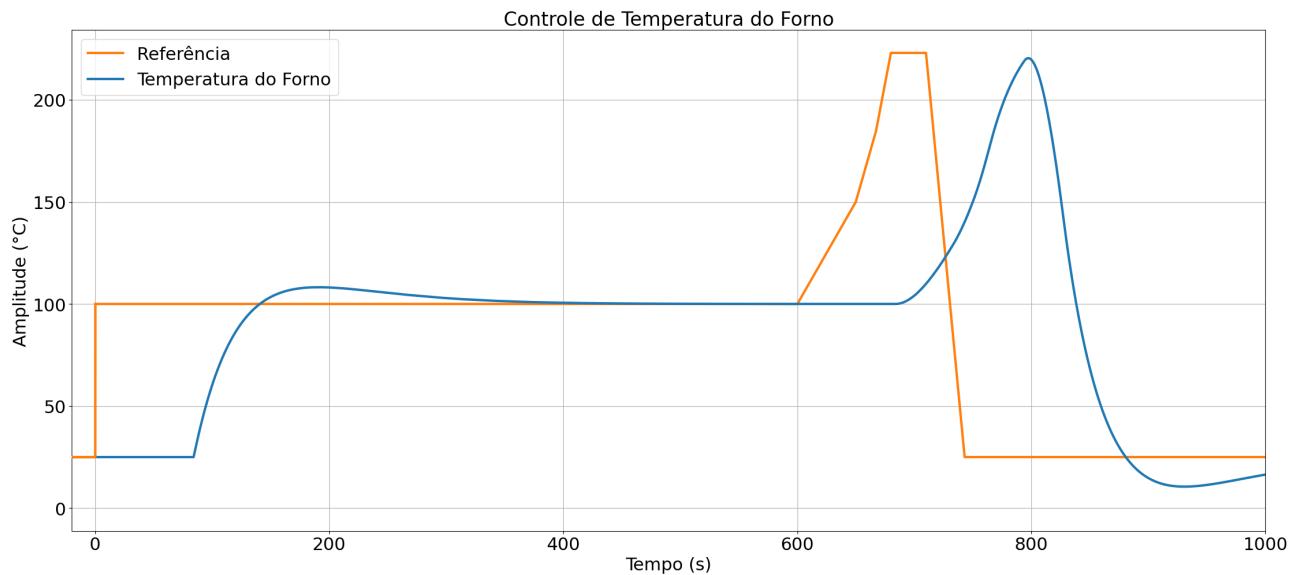


Figura 85: Controle de temperatura do forno de acordo com a curva de aquecimento desejada.

Fonte: Autoria Própria.

## 12 Conclusão

Durante uma parte do trabalho foram expostos os códigos utilizados para cálculos e geração de gráficos, porém, devido a repetibilidade desses códigos e o aumento da extensão do relatório, essa parte foi suprimida, sem grandes perdas para o leitor.

Nesse trabalho foi apresentado a fundamentação teórica, em que foi descrito o funcionamento e curva de aquecimento para um forno de solda SMD, a modelagem de um sistema dinâmico para um sistema térmico - que é o forno - utilizando a equação de calor e fazendo uma linearização utilizando a Aproximação de Padé, a modelagem do sistema no Espaço de Estados e a aplicação do método de Ziegler e Nichols para a descoberta dos parâmetros e função de transferência da planta. Após realizado todos esses passos, se comprovou, através das análises do mapa de polos e zeros e análise do sistema no domínio de Laplace e da frequência, que o sistema em malha aberta é estável porém muito lento, sendo necessário então, a implementação de um controlador para melhora dos parâmetros.

Na Seção 8 foram projetados vários tipos de controladores através da análise de estabilidade e resposta do sistema a um degrau unitário, em que foi obtida uma resposta rápida para o controlador proporcional porém que não elimina o erro de regime permanente, uma resposta extremamente lenta para o controlador integral e que elimina o erro de regime permanente e o controlador proporcional-integral, que possui um bom tempo de resposta e zera o erro de regime permanente.

Na Seção 9 foram projetados compensadores por avanço e por atraso de fase, com o objetivo de obter uma margem de fase  $\Phi_m \approx 50^\circ$  e melhorar a constante de erro estático de velocidade do sistema já com um controlador integral. Nessa seção, verificou-se que o compensador por atraso de fase conseguiu atender as especificações de projeto, porém não foi possível obter o mesmo sucesso com o compensador por avanço de fase, uma vez que esse controlador é projetado para garantir principalmente as questões do regime transitório, e não foi possível assegurar os requisitos para o regime permanente no nosso caso.

Na Seção 10 foi apresentado o controlador em espaço de utilizando, utilizando o método de alocação de polos. Observou-se uma resposta ao degrau satisfatório, com baixo sobressinal e bom tempo de acomodação, porém, infelizmente esse tipo de controlador é impossível de ser aplicado na prática a nossa planta, uma vez que um dos estados corresponde ao atraso de transporte, impossível de ser medido através de um sensor ou controlado. Dessa forma, esse controlador não é aplicável ao nosso sistema, mas ainda assim oferece uma abordagem matemática interessante, e o seu projeto foi de grande aprendizado para o grupo, uma vez que esse tema foi pouco debatido nas disciplinas anteriores.

Apesar de analisados todos esses controladores, verificou-se a necessidade da implementação de um controlador mais robusto para o projeto, uma vez que o sistema tratado em si já é um sistema complexo, possuindo um atraso puro de tempo e necessitando de algumas características mais específicas, como uma resposta mais rápida, fato que não é muito usual e fácil de ser obtido para sistemas térmicos.

Dessa forma, na Seção 11 foi feito o projeto de um controlador, inicialmente PID, com Predictor de Smith. Essa técnica foi interessante pois facilita o projeto e sintonia do controlador, uma vez que pode ser feito considerando o sistema sem o atraso de transporte. Dessa forma, foi possível projetar um controlador capaz de atender as necessidades do projeto de forma simples, conforme visto em toda seção. Ao utilizar esse controlador em conjunto com o sistema preditor, foi possível atender a necessidade do forno, ou seja, seguir a curva de aquecimento necessário para a solda de componentes SMD. Esse comportamento pode ser visto na Figura 85.

Dessa forma, podemos assumir que foi cumprido o objetivo proposto inicialmente e, mais do que isso, o trabalho foi capaz de proporcionar um grande aprendizado na área de teoria de controle, revisando e aprofundando conceitos vistos anteriormente em outras disciplinas e apresentando novas técnicas.

## Códigos

Código 1: Resposta da planta ao degrau.

```
from control.matlab import *
import matplotlib.pyplot as plt

plt.figure(figsize = (2560/96, 1080/96))
degrau = [[-200, 0, 0.0000001, 3300], [0, 0, 1, 1]]
plt.plot(degrau[0], degrau[1], label = 'Sinal de entrada')

num = [1.13]
den = [443, 1]
G = tf(num, den)
y, t = step(G)
plt.plot(t, y, label = 'Resposta sem atraso')

numD = [-1.13, 0.0269047619047619]
denD = [443, 11.547619047619, 0.0238095238095238]
GD = tf(numD, denD)
yD, tD = step(GD)
plt.plot(tD, yD, label = 'Resposta com atraso')

plt.xlabel('tempo (t)', fontsize=18)
plt.ylabel('Amplitude', fontsize=18)
plt.title('Resposta do sistema', fontsize = 22)
plt.xticks(fontsize=16)
plt.yticks(fontsize=16)
plt.xlim(-150, 3050)
plt.legend(loc = 4, fontsize=18)
plt.grid()
plt.savefig("test.png", bbox_inches='tight')
plt.show()
```

Código 2: Mapa de polos e zeros sem atraso.

```
from control.matlab import *
import matplotlib.pyplot as plt

num = [1.13]
den = [443, 1]
G = tf(num, den)
polos, zeros = pzmap(G, plot=False)
print(f'polos: {polos}, zeros: {zeros}')
plt.figure(figsize = (2560/96, 1080/96))
```

```
plt.plot(poles, [0], 'x', color = 'r', label = 'p1 = 0.00225734 + j0',
         markersize=40)
plt.grid()
plt.xlim(-.01, 0)
plt.legend(borderpad=2, loc= 'upper right', fontsize = 18)
plt.xticks(fontsize = 18)
plt.yticks(fontsize = 18)
plt.xlabel('Eixo real', fontsize=18)
plt.ylabel('Eixo imaginario', fontsize = 18)
plt.title('Mapa de polos e zeros sem atraso', fontsize= 22)
plt.savefig("mapaSemAtraso.png",bbox_inches='tight')
plt.show()
```

Código 3: Mapa de polos e zeros sem atraso.

```
from control.matlab import *
import matplotlib.pyplot as plt

num = [-1.13, 0.0269047619047619]
den = [443, 11.547619047619,0.0238095238095238 ]
G = tf(num, den)
poles, zeros = pzmap(G, plot= False)
print(f'polos: {poles}, zeros: {zeros}')
plt.figure(figsize = (2560/96, 1080/96))
plt.plot(poles[0], [0], 'x', color = 'r', label = 'p1 = -0.02380952
+ j0', markersize=30)
plt.plot(poles[1], [0], 'x', color = 'orange', label = 'p2 =
-0.00225734 + j0', markersize=30)
plt.plot(zeros, [0], 'o', color ='b', label = 'z1 = 0.02380952 + j0
', markersize=30)

plt.xlim(-.03, .03)
plt.legend(borderpad=1, loc= 'upper right', fontsize = 20)
plt.xticks(fontsize = 18)
plt.yticks(fontsize = 18)
plt.xlabel('Eixo real', fontsize=18)
plt.ylabel('Eixo imaginario', fontsize = 18)
plt.title('Mapa de polos e zeros com atraso', fontsize= 22)
plt.grid()
plt.savefig("mapaComAtraso.png",bbox_inches='tight')
plt.show()
```

Código 4: Resposta em malha aberta sem atraso.

```
from control.matlab import *
import matplotlib.pyplot as plt
import matplotlib as mpl
num = [1.13]
den = [443, 1]
G = tf(num, den)
y, t = step(G)
degrau = [[-200, 0, 0.0000001, 3300], [0, 0, 1, 1]]
plt.figure(figsize = (2560/96, 1080/96))
mpl.rcParams('text', usetex = True)
plt.plot(t,y, label = 'Resposta ao degrau', linewidth = 3)
plt.plot(degrau[0], degrau[1], label = 'Sinal de entrada', linewidth = 3)
plt.plot([443, 443], [-1, 0.7142962314762701], '--', color = 'black')
plt.plot([-1000, 443], [0.7142962314762701, 0.7142962314762701], '--', color = 'black')
plt.plot([-1000, 4000], [1.13, 1.13], '--', color = 'black')
plt.plot([1299, 1299], [-1, 1.0737406127443136], '--', color = 'black')
plt.plot([-1000, 1299], [1.0737406127443136, 1.0737406127443136], '--', color = 'black')
plt.plot([1772, 1772], [-1, 1.1093033280557303], '--', color = 'black')
plt.plot([-1000, 1772], [1.1093033280557303, 1.1093033280557303], '--', color = 'black')
plt.plot(443, 0.7142962314762701, '.', color='black', markersize=15)
plt.plot(1299, 1.0737406127443136, '.', color='black', markersize=15)
plt.plot(1772, 1.1093033280557303, '.', color='black', markersize=15)
plt.xlabel('tempo (t)', fontsize=18)
plt.ylabel('Amplitude', fontsize=18)
plt.title('Resposta ao degrau sem atraso', fontsize = 22)
plt.xticks([0, 443, 500, 1000, 1299, 1500, 1772, 2000, 2500, 3000], [0, r'$T$', 500, 1000, r'$T_{a5\%}$', 1500, r'$T_{a2\%}$', 2000, 2500, 3000], fontsize=18)
plt.yticks([0, 0.2, 0.4, 0.6, 0.7142962314762701, 0.8, 1.0, 1.0737406127443136, 1.1093033280557303, 1.13], [0, 0.2, 0.4, 0.6, f'$63,2\%$', 0.8, 1.0, r'$95\%$', r'$98,2\%$', 1.13], fontsize=11)
plt.xlim(-150, 3050)
plt.ylim(-.06, 1.2)
plt.legend(loc = 4, fontsize=18)
plt.grid()
plt.savefig("respostaSemAtraso.png", bbox_inches='tight')
plt.show()
```

Código 5: Resposta ao degrau com e sem atraso.

```
from control.matlab import *
import matplotlib.pyplot as plt
import matplotlib as mpl
num = [1.13]
den = [443, 1]
G = tf(num, den)
y, t = step(G)
t1 = [x+84 for x in t]
plt.figure(figsize = (2560/96, 1080/96))
mpl.rcParams['text', usetex = True]
plt.plot([-300, 0],[0,0], linewidth = 3, color = 'tab:blue')
plt.plot(t, y, label = 'Resposta sem atraso', linewidth = 3, color =
    'tab:blue')
plt.plot([-300, 84],[0,0], linewidth = 3, color = 'tab:orange')
plt.plot(t1, y, label = 'Resposta com atraso', linewidth = 3, color =
    'tab:orange')
plt.xlabel('tempo (t)', fontsize=18)
plt.ylabel('Amplitude', fontsize=18)
plt.title('Resposta ao degrau com e sem atraso', fontsize = 22)
plt.xticks([0,84,500,1000,1500,2000,2500,3000], fontsize=18)
plt.yticks(fontsize=18)
plt.xlim(-150, 3050)
plt.ylim(-.06, 1.2)
plt.legend(loc = 4, fontsize=18)
plt.grid()
plt.savefig("FTPPlantaExponencial.png",bbox_inches='tight')
plt.show()
```

Código 6: Padé e Atraso.

```
from control.matlab import *
import matplotlib.pyplot as plt
import matplotlib as mpl
num = [1.13]
den = [443, 1]
numD= [-1.13, 0.0269047619047619]
denD = [443, 11.547619047619, 0.0238095238095238]
G = tf(num, den)
GD = tf(numD, denD)
y, t = step(G)
yD, tD = step(GD)
t = [x+84 for x in t]
plt.figure(figsize = (2560/96, 1080/96))
mpl.rcParams['text', usetex = True]
plt.plot([-300, 0],[0,0], linewidth = 1, color = 'tab:blue')
plt.plot(tD, yD, label = 'Aproximacao de Padé', linewidth = 1, color
```

```

    = 'tab:blue')
plt.plot([-300, 84],[0,0], linewidth = 1, color = 'tab:orange')
plt.plot(t, y, label = 'Resposta com atraso', linewidth = 1, color =
    'tab:orange')
plt.title('Comparacao entre a resposta com atraso e Aproximacao de
    Pade', fontsize = 22)
plt.xlabel('Tempo (s)', fontsize=18)
plt.ylabel('Amplitude', fontsize=18)
plt.xticks([0,84,500,1000,1500,2000,2500,3000], fontsize=18)
plt.yticks(fontsize=18)
plt.xlim(-150, 3050)
plt.ylim(-.06, 1.2)
plt.legend(loc = 4, fontsize=18)
plt.grid()
plt.savefig("PadeEAtraso.png",bbox_inches='tight')
plt.show()

```

Código 7: Resposta ao degrau com atraso.

```

from control.matlab import *
import matplotlib.pyplot as plt
import matplotlib as mpl

num = [1.13]
den = [443, 1]
G = tf(num, den)
y, t = step(G)
t = [x+84 for x in t]
degrau = [[-200,0,0,0.0000001,3300], [0, 0, 1, 1]]
plt.figure(figsize = (2560/96, 1080/96))
mpl.rcParams['text', usetex = True]
plt.plot([-300, 84], [0,0], color = 'tab:blue', linewidth = 3)
plt.plot(t,y, label = 'Resposta com atraso', linewidth = 3, color =
    'tab:blue')
plt.plot(degrau[0], degrau[1], label = 'Sinal de entrada', linewidth
    = 3, color = 'tab:orange')
plt.plot([527, 527], [-1, 0.7142962314762701], '—', color = 'black'
    )
plt.plot([-300, 527], [0.7142962314762701, 0.7142962314762701], '—',
    , color = 'black')
plt.plot([1413, 1413], [-1, 1.0737406127443136], '—', color = 'black')
plt.plot([-300, 1413], [1.0737406127443136, 1.0737406127443136], '—',
    , color = 'black')
plt.plot([1856, 1856], [-1, 1.1093033280557303], '—', color = 'black')
plt.plot([-300, 1856], [1.1093033280557303, 1.1093033280557303], '—',
    , color = 'black')

```

```

plt.plot([-300, 3500], [1.13, 1.13], '—', color = 'black')
plt.plot(527, 0.7142962314762701, '.', markersize = 15, color = 'black')
plt.plot(1413, 1.0737406127443136, '.', markersize = 15, color = 'black')
plt.plot(1856, 1.1093033280557303, '.', markersize = 15, color = 'black')
plt.title('Resposta ao degrau com atraso', fontsize = 22)
plt.xlabel('Tempo (s)', fontsize=18)
plt.ylabel('Amplitude', fontsize=18)
plt.xticks([0, 84, 500, 527, 1000, 1413, 1500, 1856, 2000, 2500, 3000], [0, 84, 500, r'$T$', 1000, r'$T_{a5}\%$', 1500, r'$T_{a2}\%$', 2000, 2500, 3000], fontsize=18)
plt.yticks([0, 0.2, 0.4, 0.6, 0.7142962314762701, 0.8, 1.0, 1.0737406127443136, 1.1093033280557303, 1.13], [0, 0.2, 0.4, 0.6, r'$63,2\%$', 0.8, 1.0, r'$95\%$', r'$98,2\%$', 1.13], fontsize=11)
plt.xlim(-150, 3050)
plt.ylim(-.06, 1.2)
plt.legend(loc = 4, fontsize=18)
plt.grid()
plt.savefig("respostaComAtraso.png", bbox_inches='tight')
plt.show()

```

Código 8: Resposta ao degrau com Aproximação de Padé.

```

from control.matlab import *
import matplotlib.pyplot as plt
import matplotlib as mpl

num = [-1.13, 0.0269047619047619]
den = [443, 11.547619047619, 0.0238095238095238]
G = tf(num, den)
y, t = step(G)
degrau = [[-200, 0, 0.0000001, 3300], [0, 0, 1, 1]]
plt.figure(figsize = (2560/96, 1080/96))
mpl.rcParams['text', usetex = True]
plt.plot(t,y, label = 'Aproximacao de Pade', linewidth = 3)
plt.plot(degrau[0], degrau[1], label = 'Sinal de entrada', linewidth = 3)
plt.plot([527, 527], [-1, 0.7142962314762701], '—', color = 'black')
plt.plot([-300, 527], [0.7142962314762701, 0.7142962314762701], '—', color = 'black')
plt.plot([1413, 1413], [-1, 1.0737406127443136], '—', color = 'black')
plt.plot([-300, 1413], [1.0737406127443136, 1.0737406127443136], '—', color = 'black')

```

```

plt.plot([1856, 1856], [-1, 1.1093033280557303], '—', color = 'black')
plt.plot([-300, 1856], [1.1093033280557303, 1.1093033280557303], '—',
          color = 'black')
plt.plot([-300, 3500], [1.13, 1.13], '—', color = 'black')
plt.plot(527, 0.7142962314762701, '.', markersize = 15, color = 'black')
plt.plot(1413, 1.0737406127443136, '.', markersize = 15, color = 'black')
plt.plot(1856, 1.1093033280557303, '.', markersize = 15, color = 'black')
plt.title('Resposta ao degrau com Aproximacao de Pade', fontsize = 22)
plt.xlabel('Tempo (s)', fontsize=18)
plt.ylabel('Amplitude', fontsize=18)
plt.xticks([0, 500, 527, 1000, 1413, 1500, 1856, 2000, 2500,
           3000], [0, 500, r'$T$', 1000, r'$T_{a5\%}$', 1500, r'$T_{a2\%}$',
           2000, 2500, 3000], fontsize=18)
plt.yticks([0, 0.2, 0.4, 0.6, 0.7142962314762701, 0.8, 1.0,
           1.0737406127443136, 1.1093033280557303, 1.13], [0, 0.2, 0.4, 0.6,
           r'$63,2\%$', 0.8, 1.0, r'$95\%$', r'$98,2\%$', 1.13], fontsize = 11)
plt.xlim(-150, 3050)
plt.ylim(-.06, 1.2)
plt.legend(loc = 4, fontsize=18)
plt.grid()
plt.savefig("RespostasParametrosPade.png", bbox_inches='tight')
plt.show()

```

Código 9: Diagrama de Bode para o termo proporcional sem atraso.

```

from control.matlab import *
import numpy as np
import matplotlib.pyplot as plt

num = [1.13]
den = [1]
w = np.logspace(-6,0,10000)
G = tf(num, den)
dbode = bode(G, w, plot = False)
GdB = 20*np.log10(dbode[0])

fig, ax1 = plt.subplots(figsize = (2560/96, 1080/96))
ax1.set_xlabel('Frequencia (rad/s)', fontsize = 22)
ax1.set_xlim([10**-6, 10**0])
ax1.set_ylabel('Modulo (dB)', color='tab:blue', fontsize = 22)
ax1.xaxis.set_tick_params(labelsize=20)
ax1.yaxis.set_tick_params(labelsize=20)

```

```

ax1.plot(w, GdB, color='tab:blue', linewidth = 3)
ax1.tick_params(axis='y', labelcolor='tab:blue')
ax1.set_ylimits([-1,1.2])
ax1.set_xscale('log')
ax1.set_yticks([0, .2, .4, .6, .8, 1, 1.0615688696683936, 1.2])
ax1.grid()

ax2 = ax1.twinx()
ax2.set_ylabel('Fase', color='tab:orange', fontsize = 22)
ax2.plot(w, dbode[1], color='tab:orange', linewidth = 3)
ax2.tick_params(axis='y', labelcolor='tab:orange')
ax2.yaxis.set_tick_params(labelsize=20)
ax2.set_ylimits([-180,180])
ax2.set_xscale('log')

plt.title('Diagrama de Bode', fontsize = 24)
plt.savefig("ProporcionalSemAtraso.png",bbox_inches='tight')
plt.show()

```

Código 10: Diagrama de Bode para o polo sem atraso.

```

from control.matlab import *
import numpy as np
import matplotlib.pyplot as plt

num = [1]
den = [443, 1]
w = np.logspace(-6,0,10000)
G = tf(num, den)
dbode = bode(G, w, plot = False)
GdB = 20*np.log10(dbode[0])
phase = np.rad2deg(dbode[1])
fig, ax1 = plt.subplots(figsize = (2560/96, 1080/96))
ax1.set_xlabel('Frequencia (rad/s)', fontsize = 22)
ax1.set_xlim([10**-6, 10**0])
ax1.set_ylabel('Modulo (dB)', color='tab:blue', fontsize = 22)
ax1.xaxis.set_tick_params(labelsize=20)
ax1.yaxis.set_tick_params(labelsize=20)
ax1.plot(w, GdB, color='tab:blue', linewidth = 3)
ax1.tick_params(axis='y', labelcolor='tab:blue')
ax1.set_xscale('log')
ax1.grid()

ax2 = ax1.twinx()
ax2.set_ylabel('Fase (graus)', color='tab:orange', fontsize = 22)
ax2.plot(w, phase, color='tab:orange', linewidth = 3)
ax2.tick_params(axis='y', labelcolor='tab:orange')
ax2.yaxis.set_tick_params(labelsize=20)

```

```
ax2.set_ylim([-180,180])
ax2.set_xscale('log')
ax2.set_yticks([-135, -90, -45, 0, 45, 90, 135])
ax2.grid()

plt.title('Diagrama de Bode', fontsize = 24)
plt.savefig("PoloSemAtraso.png",bbox_inches='tight')
plt.show()
```

Código 11: Diagrama de Bode para o sistema sem atraso.

```
from control.matlab import *
import numpy as np
import matplotlib.pyplot as plt

num = [1.13]
den = [443, 1]
w = np.logspace(-6,0,10000)
G = tf(num, den)
dbode = bode(G, w, plot = False)
GdB = 20*np.log10(dbode[0])
phase = np.rad2deg(dbode[1])
plt.figure(figsize=(2560/96, 1080/96))
plt.plot(w, GdB, linewidth = 3)
plt.xscale('log')
plt.xlim([10**-6, 10**0])
plt.xticks(fontsize = 20)
plt.yticks([-50, -40, -30, -20, -10, 0, 1.062], fontsize = 20, color = 'tab:blue')
plt.xlabel('Frequencia (rad/s)', fontsize = 20)
plt.ylabel('Modulo (dB)', fontsize = 20, color = 'tab:blue')
plt.title('Diagrama de Modulo', fontsize = 24)
plt.grid()
plt.savefig("DiagramaModuloSemAtraso.png",bbox_inches='tight')
plt.show()

plt.figure(2, figsize=(2560/96, 1080/96))
plt.plot(w, phase, linewidth = 3, color = 'tab:orange')
plt.xlim([10**-6, 10**0])
plt.xticks(fontsize = 20)
plt.yticks([-90, -67.5, -45, -22.5, 0], fontsize = 20, color = 'tab:orange')
plt.xlabel('Frequencia (rad/s)', fontsize = 20)
plt.ylabel('Fase (graus)', fontsize = 20, color = 'tab:orange')
plt.title('Diagrama de Fase', fontsize = 24)
plt.xscale('log')
plt.grid()
plt.savefig('DiagramaFaseSemAtraso', bbox_inches = 'tight')
```

```
plt.show()
```

Código 12: Respostas sem atraso para uma entrada senoidal.

```
from control.matlab import *
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import lti, lsim
import matplotlib as mpl

mpl.rcParams('text', usetex = True)
num = [1.13]
den = [443, 1]
G = lti(num, den)
w_baixa = 10**-4
w_media = 10**-3
w_alta = 10**-1

# _____ w_baixa _____
t = np.linspace(0, 2*251327.41228718346, 100000)
u_baixa = 1*np.sin(w_baixa*t)
tout_baixa, y_baixa, x_baixa = lsim(G, u_baixa, t)
plt.figure(figsize=(2560/96, 1080/96))
plt.plot(t, u_baixa, linewidth = 3, color = 'tab:orange', label = 'Sinal de Entrada')
plt.plot(t, y_baixa, linewidth = 3, color = 'tab:blue', label = 'Sinal de Saída')
plt.plot([251327.41228718346, 2*251327.41228718346], [1.13, 1.13], '—', color = 'black')
plt.plot([251327.41228718346, 2*251327.41228718346], [-1.13, -1.13], '—', color = 'black')
plt.xlabel('Tempo (s)', fontsize = 22)
plt.ylabel('Amplitude', fontsize = 22)
plt.title(r'w = 10^{-4} \ rad/s$', fontsize = 24)
plt.legend(loc = 1, fontsize = 20)
plt.xticks(fontsize = 22)
plt.yticks([-1.13, -1, -.5, 0, .5, 1, 1.13], fontsize = 22)
plt.xlim([251327.41228718346, 2*251327.41228718346])
plt.grid()
plt.savefig("w10^-4SemAtraso.png", bbox_inches='tight')

# _____ w_media _____
t = np.linspace(0, 2*31415.92653589793, 100000)
u_media = 1*np.sin(w_media*t)
tout_media, y_media, x_media = lsim(G, u_media, t)
plt.figure(figsize=(2560/96, 1080/96))
plt.plot(t, u_media, linewidth = 3, color = 'tab:orange', label = 'Sinal de Entrada')
```

```
plt.plot(t, y_media, linewidth = 3, color = 'tab:blue', label = 'Sinal de Saída')
plt.plot([31415.92653589793, 2*31415.92653589793], [1.0379249317835233, 1.0379249317835233], '—', color = 'black')
plt.plot([31415.92653589793, 2*31415.92653589793], [-1.0379249317835233, -1.0379249317835233], '—', color = 'black')
plt.xlabel('Tempo (s)', fontsize = 22)
plt.ylabel('Amplitude', fontsize = 22)
plt.title(r'w = 10^{-3} \ rad/s$', fontsize = 24)
plt.legend(loc = 1, fontsize = 20)
plt.xticks(fontsize = 22)
plt.yticks([-1.0379249317835233, -1, -.5, 0, .5, 1, 1.0379249317835233], fontsize = 22)
plt.xlim([31415.92653589793, 2*31415.92653589793])
plt.grid()
plt.savefig("w10^-3SemAtraso.png", bbox_inches='tight')

# _____ w_alta _____
t = np.linspace(0, 63146.01233715484, 100000)
u_alta = 1*np.sin(w_alta*t)
tout_alta, y_alta, x_alta = lsim(G, u_alta, t)
fig, ax1 = plt.subplots(figsize = (2560/96, 1080/96))
ax1.set_xlabel('Tempo (s)', fontsize = 22)
ax1.set_xlim([62831.853071795864, 63146.01233715484])
ax1.set_ylabel('Amplitude', color='tab:orange', fontsize = 22)
ax1.xaxis.set_tick_params(labelsize=22)
ax1.yaxis.set_tick_params(labelsize=22)
entrada, = ax1.plot(t, u_alta, color='tab:orange', linewidth = 3,
                     label = 'Sinal de Entrada')
ax1.tick_params(axis='y', labelcolor='tab:orange')

ax1.grid()

ax2 = ax1.twinx()
ax2.set_ylabel('Amplitude', color='tab:blue', fontsize = 22)
saida, = ax2.plot(t, y_alta, color='tab:blue', linewidth = 3, label = 'Sinal de Saída')
ax2.plot([62831.853071795864, 63146.01233715484],
         [-0.02549292980957059, -0.02549292980957059], '—', color = 'black')
ax2.plot([62831.853071795864, 63146.01233715484],
         [0.02549292980957059, 0.02549292980957059], '—', color = 'black')
ax2.tick_params(axis='y', labelcolor='tab:blue')
ax2.yaxis.set_tick_params(labelsize=22)
ax2.set_ylim([-0.03, 0.03])
```

```

plt.legend([entrada, saida], [ 'Sinal de Entrada' , 'Sinal de Saída' ],
           fontsize = 20, loc = 1)
ax2.set_yticks([-0.03, -0.02549292980957059, -0.02, -0.01, 0, 0.01,
                0.02, 0.02549292980957059, 0.03])

plt.title(r 'w = 10^{-1} \ rad/s$', fontsize = 24)
plt.savefig("w10^-1SemAtraso.png",bbox_inches='tight')
plt.show()

```

Código 13: Diagrama de bode para o 2º polo (com atraso).

```

from control.matlab import *
import numpy as np
import matplotlib.pyplot as plt

num = [1]
den = [84/2, 1]
w = np.logspace(-5,1,100000)
G = tf(num, den)
dbode = bode(G, w, plot = False)
GdB = 20*np.log10(dbode[0])
phase = np.rad2deg(dbode[1])
fig, ax1 = plt.subplots(figsize = (2560/96, 1080/96))
ax1.set_xlabel('Frequencia (rad/s)', fontsize = 22)
ax1.set_xlim([10**-5, 10**1])
ax1.set_ylabel('Modulo (dB)', color='tab:blue', fontsize = 22)
ax1.xaxis.set_tick_params(labelsize=20)
ax1.yaxis.set_tick_params(labelsize=20)
ax1.plot(w, GdB, color='tab:blue', linewidth = 3)
ax1.tick_params(axis='y', labelcolor='tab:blue')
ax1.set_xscale('log')
ax1.grid()

ax2 = ax1.twinx()
ax2.set_ylabel('Fase (graus)', color='tab:orange', fontsize = 22)
ax2.plot(w, phase, color='tab:orange', linewidth = 3)
ax2.tick_params(axis='y', labelcolor='tab:orange')
ax2.yaxis.set_tick_params(labelsize=20)
ax2.set_ylim([-180,180])
ax2.set_xscale('log')
ax2.set_yticks([-135, -90, -45, 0, 45, 90, 135])
ax2.grid()

plt.title('Diagrama de Bode', fontsize = 24)
plt.savefig("2PoloComAtraso.png",bbox_inches='tight')
plt.show()

```

Código 14: Diagrama de bode para o zero.

```

from control.matlab import *
import numpy as np
import matplotlib.pyplot as plt

num = [-84/2, 1]
den = [1]
w = np.logspace(-5, 1, 100000)
G = tf(num, den)
dbode = bode(G, w, plot = False)
GdB = 20*np.log10(dbode[0])
phase = np.rad2deg(dbode[1])
fig, ax1 = plt.subplots(figsize = (2560/96, 1080/96))
ax1.set_xlabel('Frequencia (rad/s)', fontsize = 22)
ax1.set_xlim([10**-5, 10**1])
ax1.set_ylabel('Modulo (dB)', color='tab:blue', fontsize = 22)
ax1.xaxis.set_tick_params(labelsize=20)
ax1.yaxis.set_tick_params(labelsize=20)
ax1.plot(w, GdB, color='tab:blue', linewidth = 3)
ax1.tick_params(axis='y', labelcolor='tab:blue')
ax1.set_xscale('log')
ax1.grid()

ax2 = ax1.twinx()
ax2.set_ylabel('Fase (graus)', color='tab:orange', fontsize = 22)
ax2.plot(w, phase, color='tab:orange', linewidth = 3)
ax2.tick_params(axis='y', labelcolor='tab:orange')
ax2.yaxis.set_tick_params(labelsize=20)
ax2.set_ylim([-180,180])
ax2.set_xscale('log')
ax2.set_yticks([-135, -90, -45, 0, 45, 90, 135])
ax2.grid()

plt.title('Diagrama de Bode', fontsize = 24)
plt.savefig("ZeroComAtraso.png", bbox_inches='tight')
plt.show()

```

Código 15: Diagrama de Bode para o sistema com atraso.

```

from control.matlab import *
import numpy as np
import matplotlib.pyplot as plt

num = [-1.13, 0.0269047619047619]
den = [443, 11.547619047619, 0.0238095238095238]
w = np.logspace(-6, 2, 10000)
G = tf(num, den)
dbode = bode(G, w, plot = False)
GdB = 20*np.log10(dbode[0])

```

```

phase = np.rad2deg(dbode[1])
plt.figure(figsize=(2560/96, 1080/96))
plt.plot(w, GdB, linewidth = 3)
plt.xscale('log')
plt.xlim([10**-6, 10**2])
plt.xticks(fontsize = 20)
plt.yticks([-80, -60, -40, -20, 0, 1.062], [-80, -60, -40, -20, 0, ''], fontsize = 20, color = 'tab:blue')
plt.text(10**-6+4, 1.15, '1.062', fontsize = 20, color = 'tab:blue')
plt.xlabel('Frequencia (rad/s)', fontsize = 20)
plt.ylabel('Modulo (dB)', fontsize = 20, color = 'tab:blue')
plt.title('Diagrama de Modulo', fontsize = 24)
plt.grid()
plt.savefig("DiagramaModuloComAtraso.png", bbox_inches='tight')

plt.figure(2, figsize=(2560/96, 1080/96))
plt.plot(w, phase, linewidth = 3, color = 'tab:orange')
plt.xlim([10**-6, 10**2])
plt.xticks(fontsize = 20)
plt.yticks([-270, -180, -90, 0], fontsize = 20, color = 'tab:orange')
plt.xlabel('Frequencia (rad/s)', fontsize = 20)
plt.ylabel('Fase (graus)', fontsize = 20, color = 'tab:orange')
plt.title('Diagrama de Fase', fontsize = 24)
plt.xscale('log')
plt.grid()
plt.savefig('DiagramaFaseComAtraso', bbox_inches = 'tight')
plt.show()

```

Código 16: Resposta com atraso para uma entrada senoidal.

```

from control.matlab import *
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import lti, lsim
import matplotlib as mpl

mpl.rcParams['text', usetex = True]
num = [-1.13, 0.0269047619047619]
den = [443, 11.547619047619, 0.0238095238095238]
G = lti(num, den)
w_baixa = 10**(-5)
w_media = 10**(-2)
w_alta = 10

# _____ w_baixa _____
t = np.linspace(0, 2*2513274.1228718343, 100000)
u_baixa = 1*np.sin(w_baixa*t)

```

```
tout_baixa, y_baixa, x_baixa = lsim(G, u_baixa, t)
plt.figure(figsize=(2560/96, 1080/96))
plt.plot(t, u_baixa, linewidth = 3, color = 'tab:orange', label = 'Sinal de Entrada')
plt.plot(t, y_baixa, linewidth = 3, color = 'tab:blue', label = 'Sinal de Saída')
plt.plot([2513274.1228718343, 2*2513274.1228718343], [1.13, 1.13], '--', color = 'black')
plt.plot([2513274.1228718343, 2*2513274.1228718343], [-1.13, -1.13], '--', color = 'black')
plt.xlabel('Tempo (s)', fontsize = 22)
plt.ylabel('Amplitude', fontsize = 22)
plt.title(r'$w = 10^{-5} \ rad/s$', fontsize = 24)
plt.legend(loc = 1, fontsize = 20)
plt.xticks(fontsize = 22)
plt.yticks([-1.13, -1, -.5, 0, .5, 1, 1.13], fontsize = 22)
plt.xlim([2513274.1228718343, 2*2513274.1228718343])
plt.grid()
plt.savefig("w10^-5SemAtraso.png",bbox_inches='tight')

# _____ w_media _____
t = np.linspace(0, 65973.44572538565, 100000)
u_media = 1*np.sin(w_media*t)
tout_media, y_media, x_media = lsim(G, u_media, t)
plt.figure(figsize=(2560/96, 1080/96))
plt.plot(t, u_media, linewidth = 3, color = 'tab:orange', label = 'Sinal de Entrada')
plt.plot(t, y_media, linewidth = 3, color = 'tab:blue', label = 'Sinal de Saída')
plt.plot([62831.85307179586, 65973.44572538565],
[0.24881751098004878, 0.24881751098004878], '--', color = 'black')
plt.plot([62831.85307179586, 65973.44572538565],
[-0.24881751098004878, -0.24881751098004878], '--', color = 'black')
plt.xlabel('Tempo (s)', fontsize = 22)
plt.ylabel('Amplitude', fontsize = 22)
plt.title(r'$w = 10^{-2} \ rad/s$', fontsize = 24)
plt.legend(loc = 1, fontsize = 20)
plt.xticks(fontsize = 22)
plt.yticks([-1, -.5, -0.24881751098004878, 0, 0.24881751098004878,
.5, 1], fontsize = 22)
plt.xlim([62831.85307179586, 65973.44572538565])
plt.ylim([-1.1, 1.1])
print(y_media[70000:].max())
plt.grid()
plt.savefig("w10^-2SemAtraso.png",bbox_inches='tight')
```

```

# -----
t = np.linspace(0, 10002, 10000000)
u_alta = 1*np.sin(w_alta*t)
tout_alta, y_alta, x_alta = lsim(G, u_alta, t)
fig, ax1 = plt.subplots(figsize = (2560/96, 1080/96))
ax1.set_xlabel('Tempo (s)', fontsize = 22)
ax1.set_xlim([9996.521975601947, 10001.296623894705])
ax1.set_ylabel('Amplitude', color='tab:orange', fontsize = 22)
ax1.xaxis.set_tick_params(labelsize=22)
ax1.yaxis.set_tick_params(labelsize=22)
entrada, = ax1.plot(t, u_alta, color='tab:orange', linewidth = 3,
                     label = 'Sinal de Entrada')
ax1.tick_params(axis='y', labelcolor='tab:orange')
ax1.grid()

ax2 = ax1.twinx()

ax2.set_ylabel('Amplitude', color='tab:blue', fontsize = 22)
saida, = ax2.plot(t, y_alta, color='tab:blue', linewidth = 3, label
                  = 'Sinal de Saída')
ax2.plot([9996.521975601947, 10001.296623894705],
          [-0.0002570395782768865, -0.0002570395782768865], '—', color =
          'black')
ax2.plot([9996.521975601947, 10001.296623894705],
          [0.0002570395782768865, 0.0002570395782768865], '—', color =
          'black')
ax2.tick_params(axis='y', labelcolor='tab:blue')
ax2.yaxis.set_tick_params(labelsize=22)
ax2.set_ylim([-0.0004, 0.0004])
plt.legend([entrada, saida], ['Sinal de Entrada', 'Sinal de Saída'],
           fontsize = 20, loc = 1)
ax2.set_yticks([-0.0004, -0.0003, -0.0002570395782768865, -0.0002,
                -0.0001, 0, .0001, .0002, 0.0002570395782768865, .0003, .0004])

plt.title(r'$w = 10^{1} \ \backslash, rad/s$', fontsize = 24)
plt.savefig("w10^1SemAtraso.png", bbox_inches='tight')
plt.show()

```

Código 17: Root locus para o controlador proporcional.

```

from control.matlab import *
import numpy as np
import matplotlib.pyplot as plt
import matplotlib as mpl

mpl.rcParams['text', usetex = True]
num = [-1.13, 0.0269047619047619]

```

```

den = [443, 11.547619047619, 0.0238095238095238]
G = tf(num, den)
rlist, klist = rlocus(G, plot = False)
realPolo1 = []
realPolo2 = []
imagPolo1 = []
imagPolo2 = []
for i in range(len(rlist)):
    realPolo1.append(np.real(rlist[i][0]))
    realPolo2.append(np.real(rlist[i][1]))
    imagPolo1.append(np.imag(rlist[i][0]))
    imagPolo2.append(np.imag(rlist[i][1]))

plt.figure(figsize=(2560/96, 1080/96))
plt.plot(realPolo1, imagPolo1, color = 'tab:orange', linewidth = 3)
plt.plot(realPolo2, imagPolo2, color = 'tab:blue', linewidth = 3)
plt.plot(realPolo1[0], imagPolo1[0], marker = 'x', markersize = 12,
         color = 'tab:orange')
plt.plot(realPolo2[0], imagPolo2[0], marker = 'x', markersize = 12,
         color = 'tab:blue')
plt.plot(realPolo1[-1], imagPolo1[-1], marker = 'o', markersize = 8,
         color = 'tab:orange')
plt.plot(-0.00833, 0.0144, 'D', color = 'crimson', markersize = 6)
plt.plot(-0.00833, -0.0144, 'D', color = 'crimson', markersize = 6)
plt.xlim([-0.05, 0.12])
plt.xlabel('Eixo Real', fontsize = 22)
plt.ylabel('Eixo imaginario (j)', fontsize = 22)
plt.title('Root locus', fontsize = 24)
plt.xticks(fontsize = 22)
plt.yticks(fontsize = 22)
bbox_props = dict(boxstyle="square", pad=0.3, fc="cornsilk", ec=
    "black", lw=2)
plt.annotate(r'Polo = $-0.00833 + j0.0144$ \\ $K_p$ = $3\{,\}68$ \\ $zeta$ = $0,5$', xy=(-0.006, 0.011), xycoords='data', bbox =
    bbox_props, fontsize = 20)
plt.annotate(r'Polo = $-0.00833 - j0.0144$ \\ $K_p$ = $3\{,\}68$ \\ $zeta$ = $0,5$', xy=(-0.006, -0.007), xycoords='data', bbox =
    bbox_props, fontsize = 20)
plt.grid()
plt.savefig('ControleProporcionalRootlocus.png', bbox_inches =
    'tight')
plt.show()

```

Código 18: Diagrama de Bode com o controlador proporcional.

```

from control.matlab import *
import numpy as np
import matplotlib.pyplot as plt

```

```

num = [-5.147997499999999, 0.12257136904761902]
den = [443, 11.547619047619, 0.0238095238095238]
w = np.logspace(-6, 2, 10000)
G = tf(num, den)
dbode = bode(G, w, plot = False)
GdB = 20*np.log10(dbode[0])
phase = np.rad2deg(dbode[1])
plt.figure(figsize=(2560/96, 1080/96))
plt.plot(w, GdB, linewidth = 3)
plt.xscale('log')
plt.xlim([10**-6, 10**2])
plt.xticks(fontsize = 20)
plt.yticks([-80, -60, -40, -20, 0, 14.2332], [-80, -60, -40, -20, 0,
    '14,23'], fontsize = 20, color = 'tab:blue')
plt.xlabel('Frequencia (rad/s)', fontsize = 20)
plt.ylabel('Modulo (dB)', fontsize = 20, color = 'tab:blue')
plt.title('Diagrama de Modulo', fontsize = 24)
plt.grid()
plt.savefig("DiagramaModuloProporcional.png", bbox_inches='tight')

plt.figure(2, figsize=(2560/96, 1080/96))
plt.plot(w, phase, linewidth = 3, color = 'tab:orange')
plt.xlim([10**-6, 10**2])
plt.xticks(fontsize = 20)
plt.yticks([-270, -180, -90, 0], fontsize = 20, color = 'tab:orange')
plt.xlabel('Frequencia (rad/s)', fontsize = 20)
plt.ylabel('Fase (graus)', fontsize = 20, color = 'tab:orange')
plt.title('Diagrama de Fase', fontsize = 24)
plt.xscale('log')
plt.grid()
plt.savefig('DiagramaFaseProporcional', bbox_inches = 'tight')
plt.show()

```

Código 19: espostas em malha fechada com o controlador proporcional.

```

from control.matlab import *
import matplotlib.pyplot as plt
import matplotlib as mpl
import numpy as np

numRL = [-4.1584, 0.0990095238095238]
denRL = [443, 11.547619047619, 0.0238095238095238]
numF = [-5.147997499999999, 0.12257136904761902]
denF = [443, 11.547619047619, 0.0238095238095238]
GRL = tf(numRL, denRL)
GF = tf(numF, denF)

```

```

GRL = feedback(GRL, 1)
GF = feedback(GF, 1)
yRL, tRL = step(GRL, T = np.linspace(0, 1000, 20000) )
yF, tF = step(GF, T = np.linspace(0, 1000, 20000))

degrau = [[-200, 0, 0.0000001, 1000], [0, 0, 1, 1]]
plt.figure(figsize = (2560/96, 1080/96))
plt.plot(degrau[0], degrau[1], label = 'Sinal de entrada', linewidth = 3, color = 'tab:orange')
plt.plot(tRL, yRL, label = 'Root Locus', linewidth = 3, color = 'tab:blue')
plt.plot(tF, yF, label = 'Frequencia', linewidth = 3, color = 'tab:green')
plt.title('Respostas com controlador proporcional', fontsize = 24)
plt.xlabel('Tempo (s)', fontsize=22)
plt.ylabel('Amplitude', fontsize=22)
plt.xticks(fontsize=22)
plt.yticks(fontsize=22)
plt.xlim(-35, 1000)
plt.legend(loc = 4, fontsize=18)
plt.grid()
plt.savefig("RespostaControleProporcional.png",bbox_inches='tight')
plt.show()
plt.show()

```

Código 20: Comparaçāo entre as resposta com Padé e Atraso Puro utilizando controlador proporcional.

```

import pandas as pd
import matplotlib.pyplot as plt

data = pd.read_csv('plotRootLocusComPadeEAtraso.csv', sep=',')
print(data)
plt.figure(figsize = (2560/96, 1080/96))
plt.plot(data['tempo'], data['pade'], label = 'Resposta com Padé',
          linewidth = 3, color = 'tab:blue')
plt.plot(data['tempo'], data['atraso'], label = 'Resposta com atraso',
          linewidth = 3, color = 'tab:orange')
plt.grid()
plt.xlim([0, 2000])
plt.xticks(fontsize = 22)
plt.yticks(fontsize = 22)
plt.title('Comparacāo entre a resposta com Padé e Atraso de Transporte', fontsize = 24)
plt.legend(loc = 'lower right', fontsize = 22)
plt.xlabel('Tempo (s)', fontsize = 22)
plt.ylabel('Amplitude (C)', fontsize = 22)

```

```

plt.savefig(
    'ComparacaoPadeEAtrasoDeTransporteControladorProporcional.png',
    bbox_inches='tight')
plt.show()

```

Código 21: Diagrama de Nyquist para o sistema com atraso de transporte.

```

from control.freqplot import nyquist_plot
from control.matlab import *
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

dados = pd.read_csv('Nyquist.csv', sep=',')
num = [-1.13, 0.0269047619047619]
den = [443, 11.547619047619, 0.0238095238095238]
plt.figure(figsize=(2560/96, 1080/96))
plt.plot(dados['real'], dados['imag'], linewidth=3, color='tab:blue', zorder=2.7)
plt.plot(dados['real'], -1*dados['imag'], linewidth=3, color='tab:blue', zorder=2.7)
plt.plot(-1, 0, marker='+', markersize=22, color='tab:red', zorder=2.7)
plt.arrow(x=-.5, y=0, dx=-.465, dy=0, width=0.007, color='black', zorder=2.6)
plt.arrow(x=-.5, y=0, dx=.37, dy=0, width=0.007, color='black', zorder=2.6)
plt.annotate(text='d = 0,9021443986', xy=(-0.72, 0.025), fontsize=22)
plt.xlim([-1.2, 1.5])
plt.ylim([-0.8, 0.8])
plt.title('Diagrama de Nyquist', fontsize=24)
plt.xlabel('Eixo real', fontsize=22)
plt.ylabel('Eixo imaginario (j)', fontsize=22)
plt.xticks(fontsize=22)
plt.yticks(fontsize=22)
plt.grid()
plt.savefig('nyquistProporcional', bbox_inches='tight')
plt.show()

```

Código 22: Root Locus para estabilidade do controlador proporcional.

```

from control.matlab import *
from matplotlib.transforms import Bbox
import numpy as np
import matplotlib.pyplot as plt
import matplotlib as mpl

```

```

mpl.rc('text', usetex = True)
num = [-1.13, 0.0269047619047619]
den = [443, 11.547619047619, 0.0238095238095238]
G = tf(num, den)
rlist, klist = rlocus(G, plot = False, print_gain= True)
realPolo1 = []
imagPolo1 = []
realPolo2 = []
imagPolo2 = []
for i in range(len(rlist)):
    realPolo1.append(np.real(rlist[i][0]))
    imagPolo1.append(np.imag(rlist[i][0]))
    realPolo2.append(np.real(rlist[i][1]))
    imagPolo2.append(np.imag(rlist[i][1]))

fig = plt.figure(figsize=(2560/96, 1080/96))
plt.plot(realPolo1, imagPolo1, linewidth = 3)
plt.plot(realPolo2, imagPolo2, linewidth = 3)
plt.plot(realPolo1[0], imagPolo1[0], marker = 'x', markersize = 24,
         color = 'tab:blue')
plt.plot(realPolo2[0], imagPolo2[0], marker = 'x', markersize = 24,
         color = 'tab:orange')
plt.plot(realPolo1[-1], imagPolo1[-1], marker = '.', markersize =
         24, color = 'tab:blue')
plt.plot(0, 0.026, marker = 'D', markersize = 10, color = 'black')
plt.plot(0, -0.026, marker = 'D', markersize = 10, color = 'black')
bbox_props = dict(boxstyle="square", pad=0.3, fc="cornsilk", ec="black",
                  lw=2)
plt.annotate(r'Polo = $0 + j0,026$ \\\ Ganho = $10{,}22$', xy
            =(0.002, 0.023), xycoords='data', bbox = bbox_props, fontsize =
            20)
plt.annotate(r'Polo = $0 - j0,026$ \\\ Ganho = $10{,}22$', xy
            =(0.002, -0.021), xycoords='data', bbox = bbox_props, fontsize =
            20)
plt.xlim([-0.03, .12])
plt.yticks(fontsize = 22)
plt.xticks(fontsize = 22)
plt.xlabel('Eixo real', fontsize = 22)
plt.ylabel('Eixo Imaginario (j)', fontsize = 22)
plt.title('Root Locus', fontsize = 24)
plt.grid()
plt.savefig("RootlocusEstabilidadeProporcional.png",bbox_inches='tight')
plt.show()

```

Código 23: Root Locus para Padé de 10<sup>a</sup> ordem.

```
import matplotlib.pyplot as plt
```

```
import pandas as pd
import matplotlib as mpl

mpl.rcParams['text', usetex = True)
real = pd.read_csv('real.csv', sep=',')
imag = pd.read_csv('imag.csv', sep=',')
plt.figure(figsize = (2560/96, 1080/96))
plt.plot(real['polo0'], imag['polo0'], linewidth = 3)
plt.plot(real['polo1'], imag['polo1'], linewidth = 3)
plt.plot(real['polo2'], imag['polo2'], linewidth = 3)
plt.plot(real['polo3'], imag['polo3'], linewidth = 3)
plt.plot(real['polo4'], imag['polo4'], linewidth = 3)
plt.plot(real['polo5'], imag['polo5'], linewidth = 3)
plt.plot(real['polo6'], imag['polo6'], linewidth = 3)
plt.plot(real['polo7'], imag['polo7'], linewidth = 3)
plt.plot(real['polo8'], imag['polo8'], linewidth = 3)
plt.plot(real['polo9'], imag['polo9'], linewidth = 3)
plt.plot(real['polo10'], imag['polo10'], color = 'darkslategray',
         linewidth = 3)
# ----- Polos -----
plt.plot(-0.0740218150869702, 0.196016653787454, marker = 'x',
         markersize = 10, color = 'tab:orange')
plt.plot(-0.0740218150869702, -0.196016653787454, marker = 'x',
         markersize = 10, color = 'tab:blue')
plt.plot(-0.116338561114989, 0.14821394005926, marker = 'x',
         markersize = 10, color = 'tab:red')
plt.plot(-0.116338561114989, -0.14821394005926, marker = 'x',
         markersize = 10, color = 'tab:green')
plt.plot(-0.142084007825554, 0.1044035044962, marker = 'x',
         markersize = 10, color = 'tab:brown')
plt.plot(-0.142084007825554, -0.1044035044962, marker = 'x',
         markersize = 10, color = 'tab:purple')
plt.plot(-0.157506927746682, 0.0621801885970175, marker = 'x',
         markersize = 10, color = 'tab:gray')
plt.plot(-0.157506927746682, -0.0621801885970175, marker = 'x',
         markersize = 10, color = 'tab:pink')
plt.plot(-0.164810592987709, 0.0206586951194744, marker = 'x',
         markersize = 10, color = 'tab:cyan')
plt.plot(-0.164810592987709, -0.0206586951194744, marker = 'x',
         markersize = 10, color = 'tab:olive')
plt.plot(-0.00225733634311513, 0, marker = 'x', markersize = 10,
         color = 'darkslategray')
# ----- Zeros -----
plt.plot(0.074021815086949, 0.19601665378745, marker = 'o',
         markersize = 10, color = 'tab:orange')
```

```

plt.plot(0.074021815086949, -0.19601665378745, marker = 'o',
         markersize = 10, color = 'tab:blue')
plt.plot(0.116338561115311, 0.148213940059136, marker = 'o',
         markersize = 10, color = 'tab:red')
plt.plot(0.116338561115311, -0.148213940059136, marker = 'o',
         markersize = 10, color = 'tab:green')
plt.plot(0.142084007824134, 0.10440350449759, marker = 'o',
         markersize = 10, color = 'tab:gray')
plt.plot(0.142084007824134, -0.10440350449759, marker = 'o',
         markersize = 10, color = 'tab:pink')
plt.plot(0.157506927749218, 0.0621801885918674, marker = 'o',
         markersize = 10, color = 'tab:brown')
plt.plot(0.157506927749218, -0.0621801885918674, marker = 'o',
         markersize = 10, color = 'tab:purple')
plt.plot(0.164810592986293, 0.0206586951288776, marker = 'o',
         markersize = 10, color = 'darkslategray')
plt.plot(0.164810592986293, -0.0206586951288776, marker = 'o',
         markersize = 10, color = 'tab:olive')
# ----- Marker -----
plt.plot(0, 0.020005, marker = 'D', markersize = 4, color = 'crimson',
         label = r'$K = 7\{,\}89$')
plt.xlim([- .8, .4])
plt.ylim([- .6, .6])
plt.yticks(fontsize = 22)
plt.xticks(fontsize = 22)
plt.xlabel('Eixo Real', fontsize = 22)
plt.ylabel('Eixo Imaginario (j)', fontsize = 22)
plt.title('Diagrama de Nyquist para Pade de 10 ordem', fontsize =
24)
plt.legend(loc = 'upper left', fontsize = 22)
plt.grid()
plt.savefig("Pade10Ordem.png", bbox_inches='tight')
plt.show()

```

Código 24: Diagrama de Nyquist para o sistema com controlador integral.

```

from control import *
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

real = pd.read_csv('real.csv', sep=',')
imag = pd.read_csv('imag.csv', sep=',')

plt.figure(figsize=(2560/96, 1080/96))
plt.plot(real['real'], imag['imag'], color = 'tab:blue', linewidth =
3)

```

```

plt.plot(real[ 'real' ], -1*imag[ 'imag' ], color = 'tab:blue', linewidth = 3)
plt.plot(-1, 0, marker = '+', markersize = 22, color = 'tab:red', zorder = 2.7)
plt.arrow(x = -44.9, y = 0, dx = -44.9, dy = 0, width = 0.007, color = 'black', zorder = 2.5, length_includes_head = True, head_width = .4)
plt.arrow(x = -44.9, y = 0, dx = 44.9, dy = 0, width = 0.007, color = 'black', zorder = 2.5, length_includes_head = True, head_width = .4)
plt.annotate(text = 'd = 89,8', xy = (-50, 0.25), fontsize = 22)
plt.xlabel('Eixo Real', fontsize = 22)
plt.ylabel('Eixo Imaginario (j)', fontsize = 22)
plt.title('Diagrama de Nyquist', fontsize = 24)
plt.xticks(fontsize = 22)
plt.yticks(fontsize = 22)
plt.xlim([-120, 2])
plt.ylim([-15, 15])
plt.grid()
plt.savefig('nyquistIntegral', bbox_inches = 'tight')
plt.show()

```

Código 25: Root locus para o controlador integrador.

```

from control import *
import matplotlib.pyplot as plt
from matplotlib.transforms import Bbox
import numpy as np
import matplotlib as mpl

mpl.rcParams['text', usetex = True]
num = [-1.13, 0.0269047619047619]
den = [443, 11.547619047619, 0.0238095238095238]
G = tf(num, den)
Gc = tf([1], [1, 0])
FTMA = Gc*G
rlist, klist = rlocus(FTMA, plot = False)
realPolo1 = []
realPolo2 = []
realPolo3 = []
imagPolo1 = []
imagPolo2 = []
imagPolo3 = []
for i in range(len(rlist)):
    realPolo1.append(np.real(rlist[i][0]))
    realPolo2.append(np.real(rlist[i][1]))
    realPolo3.append(np.real(rlist[i][2]))
    imagPolo1.append(np.imag(rlist[i][0]))

```

```

    imagPolo2.append(np.imag(rlist[i][1]))
    imagPolo3.append(np.imag(rlist[i][2]))
plt.figure(figsize = (2560/96, 1080/96))
plt.plot(realPolo1, imagPolo1, color = 'tab:green', linewidth = 3)
plt.plot(realPolo2, imagPolo2, color = 'tab:blue', linewidth = 3)
plt.plot(realPolo3, imagPolo3, color = 'tab:orange', linewidth = 3)
plt.plot(realPolo1[0], imagPolo1[0], marker = 'x', markersize = 12,
         color = 'tab:green')
plt.plot(realPolo2[0], imagPolo2[0], marker = 'x', markersize = 12,
         color = 'tab:blue')
plt.plot(realPolo3[0], imagPolo3[0], marker = 'x', markersize = 12,
         color = 'tab:orange')
plt.plot(realPolo2[-1], imagPolo2[-1], marker = 'o', markersize = 8,
         color = 'tab:blue')
plt.plot(-0.00096, 0.00165, marker = 'D', markersize = 6, color =
         'crimson')
plt.plot(-0.00096, -0.00165, marker = 'D', markersize = 6, color =
         'crimson')
plt.plot(-0.0242, 0, marker = 'D', markersize = 6, color = 'crimson')
plt.xlabel('Eixo Real', fontsize = 22)
plt.ylabel('Eixo Imaginario (j)', fontsize = 22)
plt.xticks(fontsize = 22)
plt.yticks(fontsize = 22)
plt.title("Lugar das Raizes", fontsize = 24)
bbox_props = dict(boxstyle="square", pad=0.3, fc="cornsilk", ec=
                 "black", lw=2)
plt.annotate(r'Polo = $0,00096 + j0,00165$ \\ $K_i$ = $0{,}00145$',
            r'$\zeta$ = $0,5$', xy=(-0.0235, 0.0055), xycoords='data', bbox
            = bbox_props, fontsize = 20)
plt.annotate(r'Polo = $-0,00096 - j0,00165$ \\ $K_i$ = $0{,}00145$',
            r'$\zeta$ = $0,5$', xy=(-0.0248, -0.0035), xycoords='data', bbox
            = bbox_props, fontsize = 20)
plt.xlim([-0.058, 0.07])
plt.ylim([-0.02, .02])
plt.grid()
plt.savefig('rootlocusIntegral', bbox_inches='tight')
plt.show()

```

Código 26: Análise da resposta para um controlador integral.

```

from control.matlab import *
import matplotlib.pyplot as plt
import pandas as pd

data = pd.read_csv('data.csv', sep=',')
plt.figure(figsize = (2560/96, 1080/96))
plt.plot(data['tempo'], data['krootlocus'], linewidth = 3, zorder =

```

```
2.5, color = 'tab:blue', label = 'Root Locus ($K_i = 0,00145$)')
plt.plot(data['tempo'], data['kibode'], linewidth = 3, zorder = 2.5,
          color = 'tab:orange', label = 'Bode ($K_i = 0,00165$)')
plt.plot(data['tempo'], data['kiatrasorootlocus'], '--', linewidth =
          3, zorder = 2.5, color = 'tab:red', label = 'Root locus - Atraso')
plt.plot(data['tempo'], data['kiatrasobode'], '--', linewidth = 3,
          zorder = 2.5, color = 'tab:green', label = 'Bode - Atraso')

plt.legend(loc= 'lower right', fontsize = 22)
plt.title('Resposta ao degrau', fontsize = 24)
plt.xticks(fontsize = 22)
plt.yticks(fontsize = 22)
plt.xlabel('Tempo (s)', fontsize = 22)
plt.ylabel('Amplitude', fontsize = 22)
plt.xlim([0, 6500])
plt.grid()
plt.savefig('RespostaIntegrador.png', bbox_inches = 'tight')

fig, axs = plt.subplots(2, 1, figsize = (2560/96, 1080/96))
eRlocus = abs(data['kirootlocus'] - data['kiatrasorootlocus'])
eBode = abs(data['kibode'] - data['kiatrasobode'])
axs[0].plot(data['tempo'], eRlocus, linewidth = 3, color = 'tab:blue',
             label = '$|e_{rlocus}|$')
axs[0].set_xlim(0, 6500)
axs[0].set_xlabel('Tempo (s)', fontsize = 22)
axs[0].set_ylabel('$|e|$', fontsize = 22)
axs[0].grid(True)
axs[0].tick_params(axis = 'both', which = 'major', labelsize = 16)
axs[0].legend(fontsize = 22)
axs[0].set_title('Modulo do erro entre as respostas', fontsize = 24)

axs[1].plot(data['tempo'], eBode, linewidth = 3, color = 'tab:orange',
             label = '$|e_{bode}|$')
axs[1].set_xlim(0, 6500)
axs[1].set_xlabel('Tempo (s)', fontsize = 22)
axs[1].set_ylabel('$|e|$', fontsize = 22)
axs[1].grid()
axs[1].tick_params(axis = 'both', which = 'major', labelsize = 16)
axs[1].legend(fontsize = 22)

fig.tight_layout()
plt.savefig('graficodosErrosIntegral.png', bbox_inches = 'tight')
plt.show()
```

# Referências

- [Aguirre, 2016] Aguirre, L. A. (2016). Critério de nyquist (elt009, elt013, elt035). <https://www.youtube.com/watch?v=502A03Atcts>.
- [Aguirre, 2020] Aguirre, L. A. (2020). Aproximação de padé do atraso puro de tempo. <https://www.youtube.com/watch?v=zDRTsU2DZSk>.
- [Astrom and HÄGGLUND, 1995] Astrom, K. and HÄGGLUND, T. (1995). *PID Controllers: Theory, Design, and Tuning*. Instrument Society of America, 2 edition.
- [Cardoso, 2020] Cardoso, R. (2020). Sistemas de controle 2 - aula 12 - projeto de controladores em espaço de estados - parte 1.
- [Carneiro, ] Carneiro, M. L. Sistemas de controle - resumo espaço dos estados.
- [Castaño, 2019] Castaño, S. (2019). Ziegler e nichols malha aberta [controle pid] parte 1. <https://www.youtube.com/watch?v=wmyQJnTp0ls>.
- [Castaño, 2020] Castaño, S. (2020). Sistemas de primeira ordem controle [com e sem atraso] #001. [https://www.youtube.com/watch?v=5qQtN8G8G\\_g](https://www.youtube.com/watch?v=5qQtN8G8G_g).
- [Castaño, 2021] Castaño, S. (2021). Controlador pid via integral do erro [ajuste]. <https://www.youtube.com/watch?v=RkMRCWqXxAE>.
- [Cordeiro, 2016] Cordeiro, D. A. S. (2016). Uso do compensador pid no controle de temperatura de um forno elétrico a resistencia para soldagem de componentes smd. Monografia, Universidade Federal de Ouro Preto.
- [Ferreira, 2020] Ferreira, A. A. (2020). Cdmcc 004 - resposta transitória do motor cc. [https://www.youtube.com/watch?v=QSb4jG0UPVc&list=PLBeyF1M\\_iECLmoYC23Ml1wRUqeZnK2EZT&index=3](https://www.youtube.com/watch?v=QSb4jG0UPVc&list=PLBeyF1M_iECLmoYC23Ml1wRUqeZnK2EZT&index=3).
- [Henriques, 2020a] Henriques, R. M. (2020a). Teoria de controle i - cel038 - aula 14.
- [Henriques, 2020b] Henriques, R. M. (2020b). Teoria de controle i - cel038 - aula 18.
- [Meceneiro et al., 2016] Meceneiro, G. D., de Souza Conrado, B. T., Ximenes, R. L., and dos Santos, T. S. (2016). Desenvolvimento de forno para solda de componentes smd. *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, 4(1).
- [Nise, 2017] Nise, N. S. (2017). *Engenharia de Sistemas de Controle*. Livros Técnicos e Científicos Editora Ltda., 7 edition.
- [Ogata, 2003] Ogata, K. (2003). *Engenharia de Controle Moderno*. Pearson Prentice Hall, 4 edition.
- [Prasad, 1997] Prasad, R. P. (1997). *Surface Mount Technology: Principles and Practice*. Chapman & Hall, 2 edition.
- [Sena, 2017] Sena, R. L. (2017). Controlador pid aplicado a uma malha de controle de temperatura de forno resistivo. Monografia, Centro Tecnológico da Universidade Federal do Espírito Santo.

[Standard, 2014] Standard, J. I. (2014). Moisture/reflow sensitivity classification for nonhermetic surface mount devices.