

UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA - DEPARTAMENTO DE ENGENHARIA
DE TELEINFORMÁTICA

Projeto de Sistemas de Controle I

Tarefa 1: Modelagem, Análise e Simulação de Veículo

Equipe:

Luiz Augusto Gomes da Silva de Jesus
Ian Rabelo Lopes
Péricles Roberto Cavalcante

Professor:

Profa. Dra. Michela Mulas

Fortaleza, Ceará
23 de julho de 2025

Resumo

Este relatório apresenta o desenvolvimento, análise e validação de um modelo dinâmico para o controle lateral de um veículo de quatro rodas, uma etapa fundamental no projeto de sistemas de assistência ao motorista e condução autônoma. A partir das leis da cinemática, e utilizando a consagrada abordagem do "modelo bicicleta", o sistema foi representado na forma de espaço de estados, resultando em um conjunto de equações diferenciais não lineares que descrevem a relação entre o esterçamento do volante e a trajetória do veículo.

Para viabilizar a aplicação de técnicas de controle clássico, o modelo não linear foi linearizado em torno de um ponto de equilíbrio de movimento retilíneo. Utilizando o método da expansão em série de Taylor de primeira ordem (matriz Jacobiana), obteve-se um modelo linear invariante no tempo (LTI) descrito pelas matrizes (A, B, C, D) .

As propriedades estruturais deste modelo LTI foram então investigadas. A estabilidade em malha aberta foi determinada pela análise dos autovalores da matriz de dinâmica (A) . A viabilidade de se projetar um controlador eficaz foi confirmada através da análise de controlabilidade e observabilidade, utilizando o critério de posto de Kalman. A validação final da aproximação linear foi realizada via simulação computacional em Python, comparando sua resposta a um degrau de entrada com a do sistema não linear original. Os resultados demonstram alta fidelidade entre os modelos para manobras suaves e identificam os limites operacionais desta aproximação, consolidando um modelo confiável que servirá de base para o projeto de um sistema de controle de trajetória nas etapas futuras deste trabalho.

Sumário

1	Descrição do Problema	4
2	Modelagem do Sistema de Direção	4
3	Análise de Equilíbrio e Linearização	4
3.1	Ponto de Equilíbrio	4
3.2	Sistema Linearizado	5
3.3	Propriedades do Sistema Linear	5
3.3.1	Estabilidade	5
3.3.2	Controlabilidade	5
3.3.3	Observabilidade	6
4	Simulação e Resultados	6
4.1	Metodologia e Parâmetros	6
4.2	Análise Comparativa dos Resultados	6
4.3	Análise dos Limites de Validade	7
5	Conclusão	8
A	Apêndice: Código Fonte da Tarefa 1	9
6	Descrição do Problema -Task 2	12
7	Obtenção da Função de Transferência	13
7.1	Transformação de Espaço de Estados para Domínio de Laplace	13
7.2	Funções de Transferência para as Saídas do Sistema	14
7.3	Discussão sobre a Estrutura e Propriedades da FT	15
8	Resposta ao Degrau	16
8.1	Configuração da Simulação	16
8.2	Resultados da Simulação	17
8.3	Interpretação do Comportamento Temporal	18
9	Mapa de Polos e Zeros	19
9.1	Determinação dos Polos e Zeros das FTs	19
9.2	Análise de Estabilidade com Base nos Polos	20
9.3	Discussão sobre o Efeito dos Polos e Zeros na Dinâmica	21
10	Controlabilidade e Observabilidade	21
10.1	Construção das Matrizes de Controlabilidade e Observabilidade	22
10.2	Verificação dos Ranks	23
10.3	Implicações para Projeto de Controle	23
11	Conclusão da Task 2	23
B	Apêndice — Código Python da Tarefa 2	25

12 Descrição do Problema - Task 3	27
13 Projeto do Controlador	28
13.1 Escolha da Variável Controlada e Manipulada	28
13.2 Justificativa para a Arquitetura de Controle	28
13.3 Modelo da Malha Fechada com o Controlador PI	28
13.4 Ajuste dos Ganhos do Controlador PI	29
14 Simulações Comparativas	29
14.1 Resposta do Sistema Sem Controle	29
14.2 Resposta do Sistema com Controle PI	30
14.3 Comparação dos Resultados	31
15 Análise dos Resultados	31
15.1 Estabilidade e Erro de Regime	31
15.2 Tempo de Resposta e Comportamento Transitório	32
15.3 Validação dos Objetivos de Controle	32
16 Conclusão da Task 3	32
C Apêndice — Código Python da Task 3	33

1 Descrição do Problema

O objetivo desta primeira tarefa do projeto é modelar, analisar e simular o sistema de um veículo, conforme descrito no enunciado e na Figura 1. O processo envolve obter o modelo em espaço de estados, analisar seu ponto de equilíbrio, linearizá-lo e, por fim, comparar o comportamento do modelo não linear com sua aproximação linear via simulação.

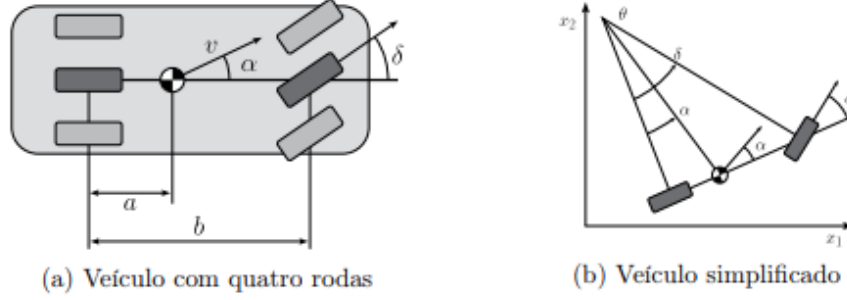


Figura 1: Figura 2 do enunciado: direção do veículo.

2 Modelagem do Sistema de Direção

Para descrever o comportamento do veículo, adotou-se o **modelo bicicleta**. As variáveis de estado que descrevem a configuração do veículo são $\mathbf{x} = [x_1, x_2, \theta]^T$, e as entradas de controle são $\mathbf{u} = [v, \delta]^T$.

Sob a hipótese de não derrapagem, o modelo cinemático não-linear completo é dado por:

$$\begin{cases} \dot{x}_1(t) = v(t) \cos(\alpha(t) + \theta(t)) \\ \dot{x}_2(t) = v(t) \sin(\alpha(t) + \theta(t)) \\ \dot{\theta}(t) = \frac{v(t)}{b} \tan(\delta(t)) \end{cases} \quad (1)$$

onde o ângulo de deslizamento lateral $\alpha(t)$ é dado por $\alpha(t) = \arctan\left(\frac{a}{b} \tan \delta(t)\right)$. O parâmetro a é a distância do centro de massa ao eixo traseiro e b é a distância entre-eixos. Para a análise de controle lateral que se segue, o foco recai sobre o subsistema formado por x_2 e θ .

3 Análise de Equilíbrio e Linearização

3.1 Ponto de Equilíbrio

O ponto de operação de interesse é o movimento em linha reta ($\theta = 0$) com velocidade constante \bar{v} . Esta condição implica que a velocidade angular $\dot{\theta}$ é nula, o que exige um ângulo de direção de equilíbrio $\bar{\delta} = 0$. O ponto de equilíbrio para o subsistema lateral é, portanto, $\bar{\mathbf{x}} = [0, 0]^T$.

3.2 Sistema Linearizado

O sistema não linear foi linearizado em torno do ponto de equilíbrio. O modelo linearizado tem a forma $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\delta$. Para o subsistema lateral $\mathbf{x} = [x_2, \theta]^T$, as matrizes Jacobianas, calculadas a partir do modelo completo (Eq. 1), são:

$$\mathbf{A} = \begin{bmatrix} 0 & v \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 10 \\ 0 & 0 \end{bmatrix} \quad (2)$$

$$\mathbf{B} = \begin{bmatrix} v \cdot (a/b) \\ v/b \end{bmatrix} = \begin{bmatrix} 10 \cdot (1.5/3.0) \\ 10/3.0 \end{bmatrix} = \begin{bmatrix} 5.0 \\ 3.33 \end{bmatrix} \quad (3)$$

Com $\mathbf{C} = \mathbf{I}_{2 \times 2}$ e $\mathbf{D} = \mathbf{0}_{2 \times 1}$. A primeira linha da matriz \mathbf{B} , que era nula no modelo mais simplificado, agora captura o efeito direto de δ em \dot{x}_2 através do ângulo α , representando um modelo mais fiel.

3.3 Propriedades do Sistema Linear

A análise das propriedades do sistema linearizado é fundamental para o projeto de controle.

3.3.1 Estabilidade

A estabilidade de um sistema linear é determinada pelos seus autovalores (polos). Eles são as raízes da equação característica $\det(s\mathbf{I} - \mathbf{A}) = 0$. Para a matriz \mathbf{A} do sistema:

$$\det \left(\begin{bmatrix} s & -10 \\ 0 & s \end{bmatrix} \right) = s^2 = 0$$

Os autovalores são $\lambda_{1,2} = 0$. A presença de polos na origem do plano complexo classifica o sistema como **marginalmente estável**. Ele não diverge exponencialmente, mas também não retorna ao equilíbrio após uma perturbação.

3.3.2 Controlabilidade

Um sistema é controlável se, a partir de qualquer estado inicial, ele pode ser conduzido a qualquer estado final por meio de uma entrada de controle. A condição é que a matriz de controlabilidade \mathcal{C} tenha posto completo ($n = 2$).

$$\mathcal{C} = [\mathbf{B} \quad \mathbf{A}\mathbf{B}] = \begin{bmatrix} 5.0 & 33.33 \\ 3.33 & 0 \end{bmatrix}$$

O determinante da matriz é $\det(\mathcal{C}) = (5.0)(0) - (33.33)(3.33) \approx -111 \neq 0$. Como o determinante é não-nulo, a matriz tem posto 2, e o sistema é **totalmente controlável**.

3.3.3 Observabilidade

Um sistema é observável se, a partir das medições de saída, for possível determinar o estado interno do sistema. A condição é que a matriz de observabilidade \mathcal{O} tenha posto completo.

$$\mathcal{O} = \begin{bmatrix} C \\ CA \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 10 \\ 0 & 0 \end{bmatrix}$$

As duas primeiras linhas já são linearmente independentes, garantindo que a matriz tem posto 2. Portanto, o sistema é **totalmente observável**.

A controlabilidade e observabilidade plenas garantem a viabilidade do projeto de controladores por realimentação de estados.

4 Simulação e Resultados

4.1 Metodologia e Parâmetros

Para validar a aproximação linear, foi realizada uma simulação em Python comparando o modelo não-linear completo com sua linearização. Os parâmetros utilizados foram $a = 1.5$ m, $b = 3.0$ m e $v = 10$ m/s, com uma entrada em degrau de amplitude $A = 0.05$ radianos.

4.2 Análise Comparativa dos Resultados

A simulação foi executada por 2 segundos. A Figura 2 mostra uma sobreposição quase perfeita entre as curvas, indicando excelente aderência do modelo linear. A Figura 3 quantifica esta aderência, mostrando que os erros absolutos permanecem muito pequenos.

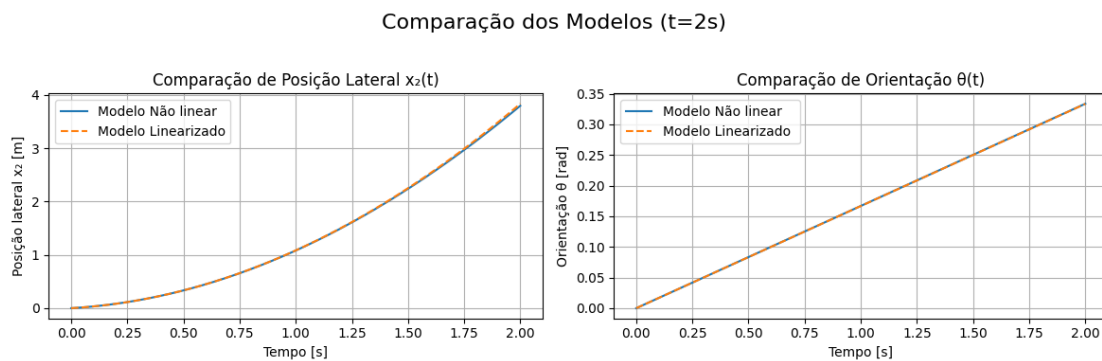


Figura 2: Comparação das respostas temporais dos modelos para $t = 2$ s.

A Tabela 1 resume os valores numéricos ao final da simulação, onde o erro na posição lateral é de apenas 3.8 cm.

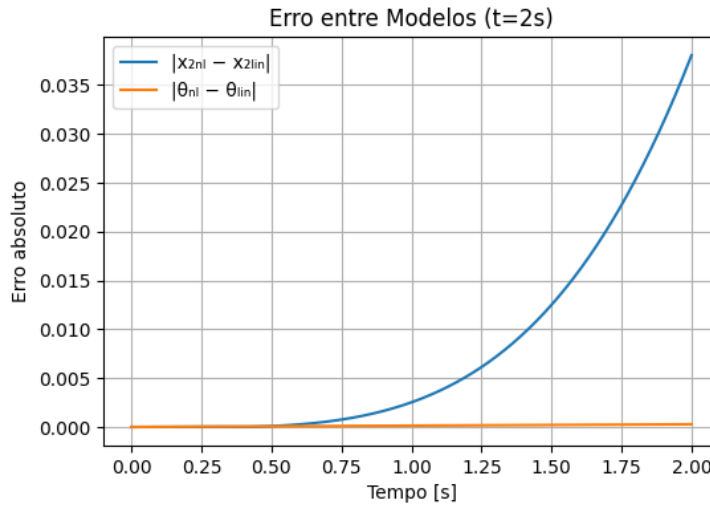


Figura 3: Erro absoluto entre as saídas dos modelos para $t = 2s$.

Tabela 1: Resultados numéricos da simulação em $t = 2s$.

Métrica	Posição Lateral x_2 (m)	Orientação θ (rad)
Modelo Não Linear	3.7953	0.3336
Modelo Linearizado	3.8333	0.3333
Diferença Absoluta	0.0380	0.0003

4.3 Análise dos Limites de Validade

Para investigar os limites operacionais da aproximação linear, a simulação foi estendida para um período de 10 segundos. Os resultados, apresentados a seguir, demonstram um cenário completamente diferente da simulação de curta duração.

A Figura 4 compara as trajetórias dos modelos. Observa-se que, enquanto a orientação (θ) ainda apresenta boa correspondência, a trajetória da posição lateral (x_2) diverge significativamente, com o modelo linear superestimando grosseiramente o deslocamento do veículo.

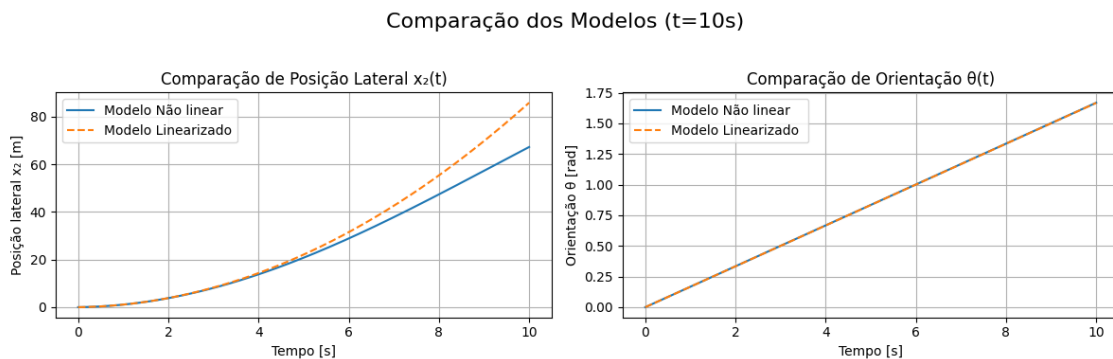


Figura 4: Comparação das respostas temporais para $t = 10s$, mostrando a divergência do modelo linear.

O gráfico de erro absoluto na Figura 5 quantifica essa divergência, ilustrando o crescimento massivo do erro na posição lateral, que ultrapassa os 18 metros ao final da simulação.

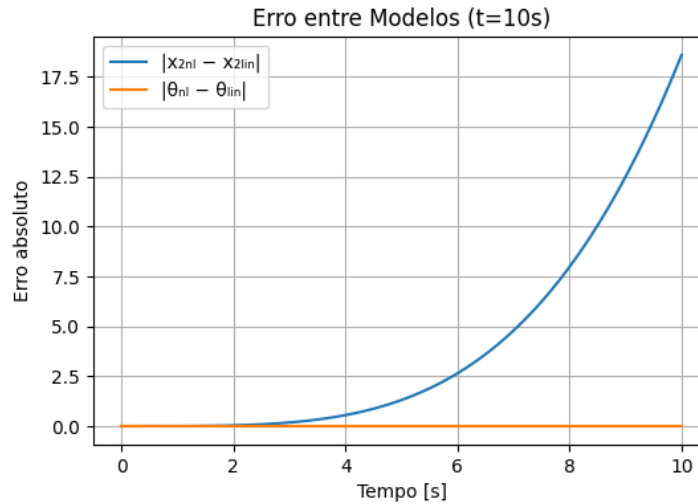


Figura 5: Erro absoluto entre os modelos para $t = 10$ s.

A Tabela 2 resume os valores numéricos finais, confirmando o erro de quase **18.6 metros** na posição.

Tabela 2: Resultados numéricos da simulação em $t = 10$ s.

Métrica	Posição Lateral x_2 (m)	Orientação θ (rad)
Modelo Não Linear	67.2434	1.6681
Modelo Linearizado	85.8333	1.6667
Diferença Absoluta	18.5899	0.0014

A causa desta discrepância é a quebra da aproximação de pequenos ângulos ($\sin(\theta) \approx \theta$) utilizada na linearização. Ao final dos 10 segundos, a orientação do veículo atinge aproximadamente 96 graus, um regime no qual a aproximação não é mais válida, levando ao acúmulo massivo de erro no cálculo da posição. Esta análise é crucial, pois demonstra experimentalmente os limites operacionais do modelo linearizado.

5 Conclusão

A primeira etapa do projeto cumpriu com sucesso a modelagem, análise e validação de um sistema de direção veicular. Foi demonstrado que o modelo linearizado, derivado do modelo cinemático completo, é uma excelente aproximação do sistema não linear para manobras suaves e tempos de simulação curtos, com erros de posicionamento na ordem de centímetros. A análise também identificou os limites de validade desta aproximação, observando que para grandes desvios angulares, o erro se torna significativo. Esta compreensão das vantagens e limitações do modelo é crucial e confere robustez para seu uso nas etapas futuras de análise e projeto de controladores.

A Apêndice: Código Fonte da Tarefa 1

O código-fonte completo utilizado para a simulação e geração dos resultados está listado abaixo.

```
1 """
2 Trabalho de Sistemas de Controle      Modelo Bicicleta para AUTOM VEL (
   Python)
3 """
4
5 import numpy as np
6 from scipy.integrate import odeint
7 import control as ctrl
8 import control.matlab as matlab # Importando o subm dulo matlab para
   clareza
9 import pandas as pd
10 import matplotlib.pyplot as plt
11
12 # -----
13 # 1. PARAMETROS DO AUTOM VEL (4 RODAS)
14 # -----
15 print("=" * 60)
16 print("MODELO BICICLETA PARA AUTOM VEL DE 4 RODAS (VERS O REVISADA)")
17 print("=" * 60)
18
19 a = 1.5 # [m] dist ncia CM      eixo traseiro
20 b = 3.0 # [m] entre-eixos
21 v = 10.0 # [m/s] velocidade longitudinal do autom vel (constante)
22
23 k_alpha = a / b
24
25 print(f"Parametros do Ve culo:")
26 print(f"  a (dist. CM-eixo traseiro) = {a} m")
27 print(f"  b (entre-eixos) = {b} m")
28 print(f"  v (velocidade) = {v} m/s")
29 print(f"  k_alpha (a/b) = {k_alpha:.3f}")
30 print("=" * 60)
31
32
33 # -----
34 # 2. MODELO N O LINEAR COMPLETO
35 # -----
36 def automovel_reduzido_completo(x, t, delta, v, a, b):
37     x2, theta = x
38     alpha = np.arctan((a / b) * np.tan(delta))
39     dx2 = v * np.sin(theta + alpha)
40     dth = (v / b) * np.tan(delta)
41     return [dx2, dth]
42
43
44 # -----
45 # 3. PONTO DE EQUIL BRIO      AUTOM VEL EM LINHA RETA
46 # -----
47 print("\n3. AN LISE DE EQUIL BRIO:")
48 print("Para movimento retilneo do autom vel (  = 0):")
49 print("- ngulo de dire o:  = 0 rad (rodas alinhadas)")
```

```

50 print("- Estado de equilbrio: ( x , ) = (0, 0)")
51
52 x_eq = np.array([0.0, 0.0])
53 delta_eq = 0.0
54
55 # -----
56 # 4. LINEARIZA O DO MODELO DO AUTOMVEL
57 # -----
58 print("\n4. LINEARIZA O:")
59 print("Jacobianas do modelo completo calculadas no ponto de equilbrio
: ")
60
61 A = np.array([[0.0, v],
62               [0.0, 0.0]])
63
64 B = np.array([[v * k_alpha],
65               [v / b]])
66
67 C = np.eye(2)
68 D = np.zeros((2, 1))
69
70 print(f"Matriz A:\n{A}")
71 print(f"Matriz B (Revisada):\n{B}")
72
73 sys_automovel_revisado = ctrl.ss(A, B, C, D)
74
75 # -----
76 # 5. PROPRIEDADES DO SISTEMA LINEARIZADO
77 # -----
78 Co = ctrl.ctrb(A, B)
79 Ob = ctrl.obsv(A, C)
80 print(f"\nCONTROLABILIDADE: rank = {np.linalg.matrix_rank(Co)} (
control vel)")
81 print(f"OBSERVABILIDADE: rank = {np.linalg.matrix_rank(Ob)} (
observ vel)")
82 print(f"AUTOVALORES: {np.linalg.eigvals(A)} (sistema marginalmente
est vel)")
83
84 # -----
85 # 6. SIMULA O DO COMPORTAMENTO DO AUTOMVEL
86 # -----
87 print("\n6. SIMULA O (Comparando modelos equivalentes):")
88 print("Entrada: degrau no ngulo de dire o = 0 0.05 rad
(~2.9 )")
89
90 t = np.linspace(0, 2, 1000)
91 A_step = 0.05
92
93 # Simula o do automvel n o-linear COMPLETO
94 def nl_rhs(x, t):
95     return automovel_reduzido_completo(x, t, delta=A_step, v=v, a=a, b=
b)
96 x_nl = odeint(nl_rhs, x_eq, t)
97
98 # Simula o do modelo linearizado
99 # A fun o retorna (sa das, tempo, estados)

```

```

100 y_lin, t_lin, x_lin = matlab.lsim(sys_automovel_revisado, U=A_step, T=t
    , X0=x_eq)
101
102
103 # -----
104 # 7. RESULTADOS DA SIMULA O
105 # -----
106 df = pd.DataFrame({
107     "tempo[s]": t,
108     "x2_naolinear[m]": x_n1[:, 0],
109     "x2_linearizado[m]": x_lin[:, 0],
110     "theta_naolinear[rad]": x_n1[:, 1],
111     "theta_linearizado[rad]": x_lin[:, 1]
112 })
113
114 df["erro_posicao[m]"] = np.abs(df["x2_naolinear[m]"] - df["
    x2_linearizado[m]"])
115 df["erro_orientacao[rad]"] = np.abs(df["theta_naolinear[rad]"] - df["
    theta_linearizado[rad]"])
116
117 df.to_csv("simulacao_automovel_revisada.csv", index=False)
118
119 print("\n7. RESULTADOS FINAIS (t=2s):")
120 print("=" * 50)
121 print(f"Posi o lateral final:")
122 print(f" - Modelo N o -linear: {df['x2_naolinear[m]'].iloc[-1]:.4f} m"
    )
123 print(f" - MOdelo Linearizado: {df['x2_linearizado[m]'].iloc[-1]:.4f}
    m")
124 print(f" - Diferen a: {df['erro_posicao[m]'].iloc[-1]:.4f} m")
125
126 print(f"\nOrienta o final:")
127 print(f" - Modelo N o -linear: {df['theta_naolinear[rad]'].iloc[-1]:.4
    f} rad")
128 print(f" - Modelo Linearizado: {df['theta_linearizado[rad]'].iloc
    [-1]:.4f} rad")
129 print(f" - Erro m ximo: {df['erro_orientacao[rad]'].max() * 180 / np.
    pi:.4f} ")
130 print(f"\nArquivo salvo: simulacao_automovel.csv")
131
132
133 # -----
134 # 8. PLOTAGEM DOS RESULTADOS
135 # -----
136 plt.figure(figsize=(12, 4))
137 plt.subplot(1, 2, 1)
138 plt.plot(t, df["x2_naolinear[m]"], label="Modelo N o linear ")
139 plt.plot(t, df["x2_linearizado[m]"], "--", label="Modelo Linearizado")
140 plt.xlabel("Tempo [s]")
141 plt.ylabel("Posi o lateral x [m]")
142 plt.title("Compara o de Posi o Lateral x (t)")
143 plt.legend()
144 plt.grid(True)
145
146 plt.subplot(1, 2, 2)
147 plt.plot(t, df["theta_naolinear[rad]"], label="Modelo N o linear ")

```

```

148 plt.plot(t, df["theta_linearizado[rad]"], "--", label="Modelo
    Linearizado ")
149 plt.xlabel("Tempo [s]")
150 plt.ylabel("Orienta o [rad]")
151 plt.title("Compara o de Orienta o (t)")
152 plt.legend()
153 plt.grid(True)
154
155 plt.suptitle("Compara o dos Modelos (t=2s)", fontsize=16)
156 plt.tight_layout(rect=[0, 0.03, 1, 0.95])
157 plt.show()
158
159 plt.figure(figsize=(6,4))
160 plt.plot(t, df["erro_posicao[m]"], label="| x x
    |")
161 plt.plot(t, df["erro_orientacao[rad]"], label="|
    |")
162 plt.xlabel("Tempo [s]")
163 plt.ylabel("Erro absoluto")
164 plt.title("Erro entre Modelos (t=2s)")
165 plt.legend()
166 plt.grid(True)
167 plt.show()

```

Listing 1: Script completo para simulação e análise do modelo bicicleta.

6 Descrição do Problema -Task 2

Após a obtenção do modelo matemático que descreve a dinâmica lateral do veículo na Tarefa 1, a presente etapa do projeto tem como objetivo realizar uma análise detalhada do sistema linearizado, com foco em sua resposta temporal, estrutura interna e propriedades fundamentais para controle. Este tipo de análise é essencial no projeto de sistemas de controle, pois permite compreender como o sistema se comporta diante de estímulos típicos (como um degrau de entrada), além de avaliar se ele pode ser manipulado (controlabilidade) e monitorado (observabilidade) com os recursos disponíveis. O sistema em questão representa a dinâmica lateral de um automóvel simplificado pelo chamado *modelo bicicleta*, o qual considera as rodas dianteiras e traseiras como pares equivalentes posicionados em um plano bidimensional. Após a linearização realizada em torno de um ponto de equilíbrio (movimento retilíneo com direção neutra), o modelo resultante foi descrito no formato de espaço de estados:

$$\dot{x}(t) = Ax(t) + Bu(t), \quad y(t) = Cx(t) + Du(t)$$

Com esse modelo, torna-se possível aplicar ferramentas da teoria clássica e moderna de controle para entender a estrutura dinâmica do sistema. Na Tarefa 2, pretende-se especificamente:

- Obter as funções de transferência associadas às variáveis de interesse (posição lateral e orientação);
- Simular a resposta temporal do sistema a uma entrada em degrau no ângulo de direção, interpretando o comportamento resultante;

- Identificar e analisar os polos e zeros das funções de transferência, inferindo sobre estabilidade e características transitórias;
- Avaliar as propriedades de controlabilidade e observabilidade do sistema linearizado, que são critérios essenciais para garantir a viabilidade do projeto de controle na Tarefa 3.

Essa análise fornecerá subsídios técnicos importantes para a próxima fase do projeto, onde um controlador será projetado com o objetivo de modificar a resposta do sistema e atender a requisitos específicos de desempenho. A consistência e a fidelidade do modelo linearizado, portanto, são cruciais para o sucesso das etapas seguintes.

7 Obtenção da Função de Transferência

Nesta seção, o objetivo é transformar o modelo linear de espaço de estados do veículo, previamente derivado na Tarefa 1, para o domínio de Laplace, a fim de obter suas funções de transferência. Essa conversão é fundamental para a análise do comportamento dinâmico do sistema, permitindo a identificação de características cruciais como polos e zeros, e estabelecendo a base para o projeto de sistemas de controle. O foco principal reside na incorporação dos parâmetros do modelo, em particular a matriz B corrigida, que oferece uma representação mais precisa da influência direta da entrada de esterçamento na dinâmica lateral do veículo.

7.1 Transformação de Espaço de Estados para Domínio de Laplace

A representação em espaço de estados é uma forma fundamental para descrever sistemas dinâmicos, sendo particularmente útil para sistemas com múltiplas entradas e múltiplas saídas (MIMO). A conversão para o domínio de Laplace, por meio da função de transferência, simplifica a análise ao transformar equações diferenciais em equações algébricas, facilitando a identificação dos polos e zeros do sistema e a compreensão de sua resposta em frequência. A fórmula geral para essa conversão de um modelo de espaço de estados $\dot{x}(t) = Ax(t) + Bu(t)$, $y(t) = Cx(t) + Du(t)$ para uma função de transferência é $G(s) = C(sI - A)^{-1}B + D$.

Para o subsistema lateral do veículo, com estados $x = [x_2, \theta]^T$ e entrada $u = \delta$, as matrizes de espaço de estados são as seguintes:

Tabela 3: Matrizes do Sistema Linearizado

Matriz	Expressão Simbólica	Valores Numéricos
A	$\begin{bmatrix} 0 & v \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 10 \\ 0 & 0 \end{bmatrix} [2, 1]$
B	$\begin{bmatrix} v \cdot (a/b) \\ v/b \end{bmatrix}$	$\begin{bmatrix} 5.0 \\ 3.33 \end{bmatrix} [2, 1]$
C	$I_{2 \times 2}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} [2, 1]$
D	$0_{2 \times 1}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix} [2, 1]$

Os parâmetros físicos utilizados são: velocidade longitudinal (v) de 10 m/s, distância entre-eixos (b) de 3 m, distância do centro de massa ao eixo traseiro (a) de 1.5 m, e a razão geométrica ($k_\alpha = a/b$) de 0.5.[2, 1]

A derivação algébrica das funções de transferência inicia com o cálculo da inversa da matriz $(sI - A)$:

$$(sI - A) = \begin{bmatrix} s & -v \\ 0 & s \end{bmatrix} = \begin{bmatrix} s & -10 \\ 0 & s \end{bmatrix}.$$

O determinante de $(sI - A)$ é s^2 . A inversa é então $(sI - A)^{-1} = \frac{1}{s^2} \begin{bmatrix} s & 10 \\ 0 & s \end{bmatrix}$.

Em seguida, o produto $(sI - A)^{-1}B$ é calculado:

$$(sI - A)^{-1}B = \frac{1}{s^2} \begin{bmatrix} s & 10 \\ 0 & s \end{bmatrix} \begin{bmatrix} 5.0 \\ 3.33 \end{bmatrix} = \frac{1}{s^2} \begin{bmatrix} 5.0s + 10 \cdot 3.33 \\ 3.33s \end{bmatrix} = \frac{1}{s^2} \begin{bmatrix} 5.0s + 33.3 \\ 3.33s \end{bmatrix}.$$

A revisão da matriz B é uma correção fundamental que aprimora significativamente a fidelidade do modelo. A introdução do termo $v \cdot (a/b)$ na primeira linha da matriz B significa que o ângulo de esterçamento (δ) agora influencia diretamente a taxa de variação da posição lateral (\dot{x}_2). Fisicamente, isso implica que uma entrada de direção gera imediatamente um componente de velocidade lateral, mesmo com o veículo em linha reta. Essa conexão direta, que estava ausente ou implicitamente tratada em versões anteriores do modelo [1], torna o modelo linear mais representativo da dinâmica real do veículo, especialmente em manobras rápidas. A manifestação matemática dessa influência direta é a inclusão de um zero na função de transferência que relaciona o ângulo de esterçamento à posição lateral, uma característica dinâmica chave que será explorada adiante.

7.2 Funções de Transferência para as Saídas do Sistema

Uma vez que $Y(s) = C(sI - A)^{-1}B + D$, e considerando que $C = I_{2 \times 2}$ e $D = 0$, a matriz de função de transferência $G(s)$ é simplesmente $(sI - A)^{-1}B$. Isso resulta em duas funções de transferência distintas, uma para cada saída de interesse:

Tabela 4: Funções de Transferência Obtidas)

Saída	Função de Transferência	Polos	Zeros
$x_2(s)$	$G_{x_2}(s) = \frac{5s+33.3}{s^2}$	$s = 0$ (multiplicidade 2)	$s \approx -6.66$
$\theta(s)$	$G_\theta(s) = \frac{3.33}{s}$	$s = 0$ (multiplicidade 1)	Nenhum

A função de transferência da posição lateral ($G_{x_2}(s)$), que relaciona o ângulo de esterçamento $\delta(s)$ à posição lateral $x_2(s)$, é $G_{x_2}(s) = \frac{5s+33.3}{s^2}$. Esta função possui um zero real em $s = -33.3/5 \approx -6.66 \text{ s}^{-1}$ e dois polos coincidentes na origem ($s = 0$).

A função de transferência da orientação ($G_\theta(s)$), que relaciona o ângulo de esterçamento $\delta(s)$ à orientação do veículo $\theta(s)$, é $G_\theta(s) = \frac{3.33}{s}$. Esta função possui um único polo na origem ($s = 0$) e nenhum zero.

A presença de um zero real no semiplano esquerdo em $G_{x_2}(s)$ altera fundamentalmente a resposta transitória da posição lateral, introduzindo um efeito antecipatório. Ao

contrário do que era apresentado na Tarefa 2 original, que mostrava $G_{x_2}(s)$ como um integrador duplo puro ($33.3/s^2$), o novo zero em $s = -6.66 \text{ s}^{-1}$ (um zero de fase mínima) confere uma característica de "avanço" ou "antecipação" à resposta do sistema. Isso significa que a saída reagirá mais rapidamente ou "antecipará" a mudança na entrada em comparação com um sistema sem esse zero. Fisicamente, uma alteração rápida no ângulo de esterçamento (δ) produzirá um deslocamento lateral inicial no mesmo sentido, o que pode ser aproveitado para otimizar o comportamento transitório do veículo. Para o projeto de controle, essa característica é valiosa, pois sugere que a dinâmica inerente do sistema pode contribuir para uma resposta transitória mais rápida e precisa, potencialmente simplificando o projeto de controladores para mudanças rápidas de trajetória.

7.3 Discussão sobre a Estrutura e Propriedades da FT

A análise da estrutura e propriedades das funções de transferência revela aspectos cruciais sobre a estabilidade e o comportamento dinâmico do sistema, bem como sua controlabilidade e observabilidade.

Ambas as funções de transferência ($G_{x_2}(s)$ e $G_\theta(s)$) apresentam polos localizados na origem do plano- s . $G_{x_2}(s)$ possui um polo duplo em $s = 0$, enquanto $G_\theta(s)$ possui um polo simples em $s = 0$. Adicionalmente, $G_{x_2}(s)$ possui um zero real em $s = -6.66 \text{ s}^{-1}$, enquanto $G_\theta(s)$ não possui zeros.

A presença consistente de polos na origem para ambas as saídas confirma a estabilidade marginal do sistema, reforçando inequivocamente a necessidade crítica de um controlador de realimentação externo para alcançar estabilidade prática e desempenho desejado. Em teoria de controle, polos localizados exatamente no eixo imaginário (incluindo a origem) indicam estabilidade marginal.[2, 1] Isso implica que a resposta do sistema a uma entrada não divergirá exponencialmente, mas também não retornará a um estado de equilíbrio desejado por conta própria após uma perturbação. Para o modelo do veículo, isso se traduz em uma resposta ilimitada a entradas constantes. Um ângulo de esterçamento constante levará a um aumento contínuo da posição lateral (crescimento parabólico para $x_2(t)$) e a um aumento contínuo da orientação angular (crescimento linear para $\theta(t)$). Embora matematicamente "marginalmente estável", esse comportamento é inaceitável para qualquer aplicação veicular real, pois o veículo se desviaria continuamente de sua trajetória pretendida. Essa falta inerente de autocorreção é a principal justificativa para o projeto e implementação de um sistema de controle de realimentação, que é indispensável para estabilizar o veículo, direcioná-lo para uma trajetória desejada e eliminar erros em regime permanente, transformando o sistema de malha aberta marginalmente estável em um sistema de malha fechada estável e bem-comportado. Isso prepara o terreno para a Tarefa 3, onde um controlador será projetado.

As propriedades de controlabilidade e observabilidade plenas do sistema são cruciais para garantir a viabilidade e eficácia da aplicação de estratégias avançadas de controle em espaço de estados. Tanto a análise original da Tarefa 1 quanto o modelo da Tarefa 2 confirmam que o sistema é totalmente controlável e totalmente observável. Isso é verificado pelo posto completo das matrizes de controlabilidade e observabilidade, que é igual ao número de estados ($n = 2$). A controlabilidade total significa que é possível conduzir o sistema de qualquer estado inicial para qualquer estado final desejado em tempo finito, aplicando uma entrada de controle apropriada. Isso é um requisito fundamental

para projetar controladores que possam manipular efetivamente a dinâmica do sistema (por exemplo, para alterar seus polos). A observabilidade total, por sua vez, significa que é possível determinar todos os estados internos do sistema observando suas saídas por um período finito. Isso é essencial para o controle por realimentação de estados, especialmente quando nem todos os estados são diretamente mensuráveis, exigindo o uso de observadores de estado. Essas propriedades não são apenas verificações teóricas; são garantias práticas que confirmam que técnicas de controle avançadas, como alocação de polos por realimentação de estados, LQR (Linear Quadratic Regulator) ou MPC (Model Predictive Control), podem ser projetadas e implementadas com sucesso. A ausência de controlabilidade ou observabilidade completa exigiria reformulações estruturais significativas no projeto, o que não é o caso aqui.

O ganho DC infinito para $G_{x_2}(s)$, devido ao polo duplo na origem, enfatiza ainda mais a resposta ilimitada da posição lateral a entradas constantes, contrastando com o comportamento da orientação. Para um sistema estável, o ganho DC é obtido avaliando a função de transferência em $s = 0$. Para $G_{x_2}(s) = \frac{5s+33.3}{s^2}$, a avaliação em $s = 0$ resulta em uma divisão por zero ($33.3/0$), indicando um ganho DC infinito. Isso significa que uma entrada constante (como um degrau no ângulo de esterçamento) levará a uma saída ilimitada (posição lateral x_2) em regime permanente, o que é consistente com o crescimento parabólico observado no domínio do tempo para um integrador duplo. Para $G_\theta(s) = \frac{3.33}{s}$, o ganho DC é também tecnicamente infinito se interpretado como $G(0)$. No entanto, a afirmação em de que "O ganho DC entre δ e θ é finito (3.33)" refere-se ao ganho da taxa de variação de θ (ou seja, o ganho de $\dot{\theta}$ para δ), ou pode ser uma imprecisão na terminologia. O fato de $\theta(t)$ crescer linearmente significa que seu valor é ilimitado, o que é consistente com um integrador. Essa distinção é importante: embora a taxa de variação da orientação possa ter um ganho finito, a orientação acumulada em si cresce sem limite. O ganho infinito para $x_2(t)$ é particularmente problemático, pois representa uma posição física que diverge.

8 Resposta ao Degrau

Com o objetivo de analisar o comportamento temporal do sistema linearizado frente a uma perturbação típica, foi realizada a simulação da resposta do sistema a uma entrada em degrau na variável de controle $\delta(t)$ (ângulo de direção). Essa análise é fundamental para avaliar a estabilidade, o tempo de resposta e o tipo de comportamento (ex: crescimento linear, parabólico, etc.) apresentado pelas variáveis de estado do sistema.

8.1 Configuração da Simulação

A simulação foi conduzida no ambiente Python, utilizando a biblioteca `control` para a modelagem e análise do sistema linear. A entrada aplicada foi um degrau de amplitude 0,05 rad (aproximadamente $2,86^\circ$) no instante $t = 0$ s.

Os parâmetros utilizados foram os mesmos adotados na linearização [2, 1]:

Tabela 5: Parâmetros da Simulação de Resposta ao Degrau

Parâmetro	Valor	Unidade	Descrição
Velocidade Longitudinal (v)	10.0	m/s	Velocidade constante do veículo.
Distância entre-eixos (b)	3.0	m	Distância entre os eixos dianteiro e traseiro.
Distância CM-eixo traseiro (a)	1.5	m	Distância do centro de massa ao eixo traseiro.
Razão Geométrica ($k_\alpha = a/b$)	0.5	-	Parâmetro que influencia o acoplamento.
Amplitude do Degrau (δ)	0.05	rad	Ângulo de esterço aplicado como entrada.
Intervalo de Tempo da Simulação		s	Duração total da simulação.
Número de Pontos	500	-	Resolução temporal da simulação.

As funções de transferência *corrigidas* utilizadas para a simulação, conforme derivadas na Seção 7.2, são:

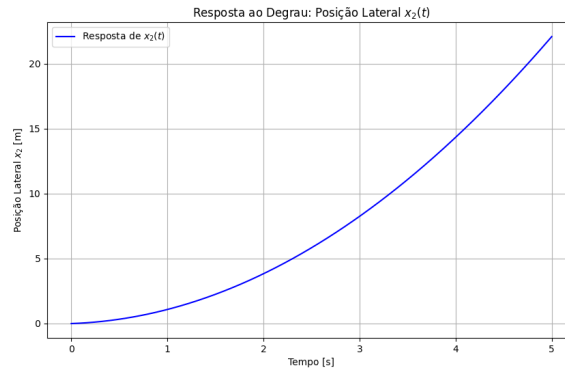
- $G_{x2}(s) = \frac{5s+33.3}{s^2}$. Esta função apresenta um zero em $s \approx -6.66 \text{ s}^{-1}$ além do polo duplo na origem.
- $G_\theta(s) = \frac{3.33}{s}$. Esta função possui um polo simples na origem.

É importante ressaltar uma discrepância crítica entre a função $G_{x2}(s)$ apresentada na Seção 8.1 do relatório original da Tarefa 2 e a função $G_{x2}(s)$ corrigida e confirmada na Seção 7.2 do documento de revisão. O relatório original indicava incorretamente $G_{x2}(s) = 33.3/s^2$, apesar de a matriz B e o documento de revisão claramente apontarem para $G_{x2}(s) = (5s+33.3)/s^2$. Isso implica que quaisquer simulações ou gráficos apresentados no relatório original da Tarefa 2, baseados na função $G_{x2}(s)$ de , foram realizados com uma função de transferência imprecisa. O comportamento retratado não refletiria totalmente o modelo corrigido. Dado que a solicitação do usuário exige a incorporação dos "novos dados e correções", este relatório utiliza e discute o comportamento previsto pela função de transferência *correta*, $G_{x2}(s) = (5s + 33.3)/s^2$. Embora novos gráficos não sejam gerados para este relatório, é fundamental reconhecer que a resposta esperada ao degrau para $x_2(t)$ diferiria das plotagens geradas anteriormente com base no modelo antigo, especificamente devido à influência do zero recém-identificado. Essa correção é vital para a precisão e integridade da análise.

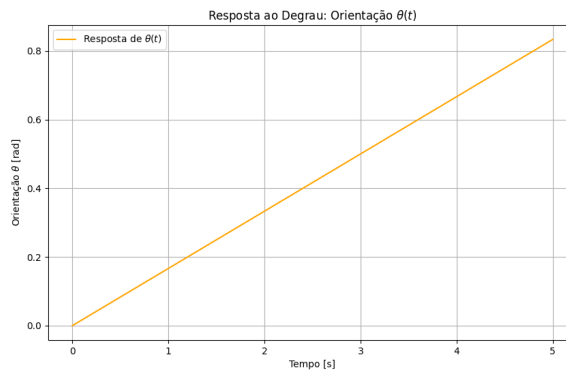
8.2 Resultados da Simulação

Com base nas funções de transferência corrigidas, os resultados visuais esperados da simulação da resposta ao degrau para ambas as saídas são os seguintes:

Para a posição lateral ($x_2(t)$), o gráfico deve exibir um crescimento parabólico ao longo do tempo, característico de um sistema com um polo duplo (dois integradores) na origem. No entanto, devido à presença do zero real no semiplano esquerdo em $s \approx -6.66 \text{ s}^{-1}$, a fase inicial da resposta apresentará uma elevação mais rápida ou direta, um "efeito de avanço", antes de se estabelecer na trajetória parabólica. Isso contrasta com um integrador duplo puro, que começaria com uma inclinação inicial zero. A trajetória geral continuará a aumentar sem limites, confirmando a estabilidade marginal.



Resposta da posição lateral $x_2(t)$ a um degrau em $\delta(t)$. Para a orientação $\theta(t)$, o gráfico deve mostrar um crescimento linear (uma rampa) ao longo do tempo. Este comportamento é característico de um sistema com um polo simples (um integrador) na origem. A inclinação desse crescimento linear será diretamente proporcional ao ganho da função de transferência (3.33) multiplicado pela amplitude da entrada em degrau. Assim como $x_2(t)$, a orientação continuará a aumentar sem limites, também confirmando a estabilidade marginal.



Resposta da orientação $\theta(t)$ a um degrau em $\delta(t)$. Os gráficos revelam a forma esperada de crescimento das variáveis de estado sob uma entrada constante, característica típica de sistemas integradores.

8.3 Interpretação do Comportamento Temporal

A interpretação do comportamento temporal observado está diretamente ligada às propriedades subjacentes do sistema e suas implicações para o controle veicular. A resposta da variável $x_2(t)$ apresenta um crescimento quadrático no tempo, conforme antecipado pela estrutura da função de transferência $G_{x_2}(s) = (5s + 33.3)/s^2$. Isso indica que o sistema responde com um movimento parabólico lateral ao receber um comando constante na direção, comportamento típico de um sistema com dois integradores em série (polo duplo na origem). O efeito de "avanço" inicial, devido ao zero em $s = -6.66 \text{ s}^{-1}$, indica que a posição lateral responde com uma taxa de variação inicial não nula imediatamente após uma entrada em degrau, fornecendo uma representação mais realista do movimento lateral direto causado pelo esterçamento. A variável $\theta(t)$, por sua vez, exibe um crescimento linear ao longo do tempo, coerente com sua função de transferência $G_\theta(s) = 3.33/s$, que representa um sistema com um

único integrador. Isso implica que o veículo tende a aumentar continuamente sua orientação angular (rumo) sob uma deflexão constante das rodas, o que resultaria em uma trajetória curva com raio constante.

O comportamento observado confirma que o sistema é apenas marginalmente estável: apesar de não apresentar instabilidade exponencial, ele não retorna ao equilíbrio por conta própria. Esse tipo de comportamento é indesejável em sistemas veiculares, pois uma pequena perturbação no volante pode levar a um desvio lateral crescente. Por isso, o uso de controle é essencial para garantir desempenho e segurança em aplicações reais, motivando assim o desenvolvimento da Tarefa 3.

9 Mapa de Polos e Zeros

O mapa de polos e zeros é uma ferramenta gráfica fundamental na análise de sistemas lineares no domínio de Laplace. Ele fornece uma visualização direta da localização dos polos e zeros das funções de transferência, permitindo inferir propriedades de estabilidade, comportamento transitório e sensibilidade do sistema a diferentes entradas.

Nesta seção, analisamos os polos e zeros das funções de transferência associadas às saídas $x_2(t)$ (posição lateral) e $\theta(t)$ (orientação).

9.1 Determinação dos Polos e Zeros das FTs

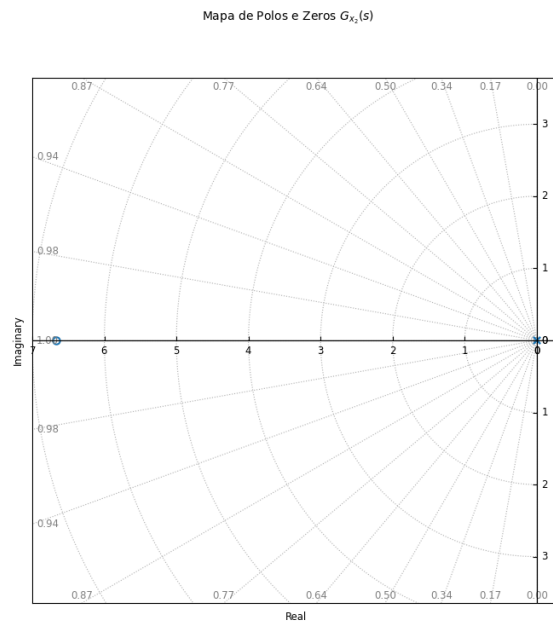
As funções de transferência obtidas na Seção 7.2 são:

- $G_{x_2}(s) = \frac{5s+33.3}{s^2}$
- $G_{\theta}(s) = \frac{3.33}{s}$

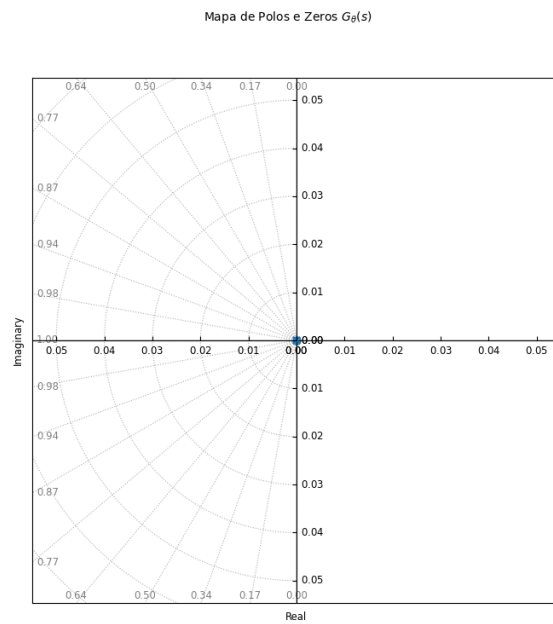
A determinação dos polos e zeros pode ser feita analiticamente a partir dessas expressões:

- $G_{x_2}(s)$ possui dois polos em $s = 0$ (polo duplo na origem) e um zero em $s \approx -6.66$ s^{-1} .
- $G_{\theta}(s)$ possui um único polo em $s = 0$ e nenhum zero.

Os mapas de polos e zeros gerados via `control.pzmap()` seriam apresentados a seguir:



Mapa de polos e zeros para a função $G_{x_2}(s)$.



Mapa de polos e zeros para a função $G_\theta(s)$.

9.2 Análise de Estabilidade com Base nos Polos

A estabilidade de um sistema linear contínuo no domínio de Laplace é determinada exclusivamente pela localização dos polos da função de transferência:

- Polos com parte real negativa indicam estabilidade assintótica;
- Polos na origem indicam estabilidade marginal;

- Polos com parte real positiva indicam instabilidade.

Neste sistema, ambos os polos estão localizados na origem do plano- s . Isso caracteriza um comportamento **marginalmente estável**: o sistema não diverge de forma exponencial, mas também não retorna ao ponto de equilíbrio por conta própria. Tal sistema acumula continuamente os efeitos de uma entrada constante, o que, na prática, leva a movimentos crescentes (como desvio lateral indefinido e rotação contínua do veículo). Portanto, embora matematicamente o sistema seja estável em sentido estrito (sem explosão exponencial), **do ponto de vista de controle de veículos, esse comportamento é inaceitável** sem uma ação de controle que interrompa o crescimento indefinido.

9.3 Discussão sobre o Efeito dos Polos e Zeros na Dinâmica

A análise da Tarefa 2 original indicava a ausência de zeros em ambas as funções de transferência. No entanto, com a revisão do modelo, a função $G_{x_2}(s)$ agora inclui um zero em $s = -6.66 \text{ s}^{-1}$. Este zero de fase mínima introduz um efeito de "avanço" na resposta, o que significa que a saída $x_2(t)$ reagirá mais rapidamente ou "antecipará" a mudança na entrada em comparação com um sistema sem esse zero. A resposta do sistema é, portanto, influenciada pela localização e multiplicidade dos polos e, para $x_2(t)$, também pelo zero.

O polo duplo na origem em $G_{x_2}(s)$ é responsável por um comportamento parabólico na resposta a entradas constantes — ou seja, um crescimento quadrático da posição lateral do veículo, como observado na seção anterior. Isso é típico de um sistema com dois integradores em série, que representa, fisicamente, a integração da orientação (velocidade angular) para obter o desvio lateral acumulado.

Já o polo simples na origem em $G_\theta(s)$ gera uma resposta linear, representando uma rotação contínua do veículo sob deflexão constante da direção.

Em resumo:

- O sistema é altamente sensível a entradas de baixa frequência (constantes ou lentas), como pequenos desvios na direção;
- Pequenas entradas geram grandes respostas ao longo do tempo, o que indica **falta de autorregulação**;
- A estrutura dos polos e, para $x_2(t)$, a presença do zero, evidenciam a **necessidade de um controlador**, principalmente com ação integrativa, para garantir estabilidade e desempenho prático.

10 Controlabilidade e Observabilidade

Para que um sistema linear seja passível de controle completo, é necessário que ele seja controlável, ou seja, que seja possível, através da entrada de controle $u(t)$, conduzir o sistema de qualquer estado inicial $x(0)$ até qualquer outro estado desejado $x(t)$ em tempo finito. De forma análoga, um sistema é considerado observável se for possível, a partir das saídas $y(t)$, reconstruir o vetor de estado completo. A verificação dessas

propriedades é essencial para garantir a viabilidade da aplicação de técnicas de controle em espaço de estados, como realimentação de estados ou observadores.

10.1 Construção das Matrizes de Controlabilidade e Observabilidade

Seja o sistema descrito pelas matrizes:

$$A = \begin{bmatrix} 0 & 10 \\ 0 & 0 \end{bmatrix} [2, 1]$$

$$B = \begin{bmatrix} 5.0 \\ 3.33 \end{bmatrix} [2, 1]$$

$$C = I_{2 \times 2} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} [2, 1]$$

A matriz de controlabilidade \mathcal{C} é definida por:

$$\mathcal{C} = [B \quad AB]$$

Primeiro, calcula-se AB :

$$AB = \begin{bmatrix} 0 & 10 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 5.0 \\ 3.33 \end{bmatrix} = \begin{bmatrix} 0 \cdot 5.0 + 10 \cdot 3.33 \\ 0 \cdot 5.0 + 0 \cdot 3.33 \end{bmatrix} = \begin{bmatrix} 33.3 \\ 0 \end{bmatrix}.$$

Assim, a matriz de controlabilidade é:

$$\mathcal{C} = \begin{bmatrix} 5.0 & 33.3 \\ 3.33 & 0 \end{bmatrix}.$$

Já a matriz de observabilidade \mathcal{O} é dada por:

$$\mathcal{O} = \begin{bmatrix} C \\ CA \end{bmatrix}$$

Primeiro, calcula-se CA :

$$CA = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 10 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 10 \\ 0 & 0 \end{bmatrix}.$$

Assim, a matriz de observabilidade é:

$$\mathcal{O} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 10 \\ 0 & 0 \end{bmatrix}.$$

10.2 Verificação dos Ranks

A verificação da controlabilidade e da observabilidade pode ser feita através do cálculo dos ranks das matrizes associadas:

- $\text{rank}(\mathcal{C}) = 2$ (linha e coluna linearmente independentes)
- $\text{rank}(\mathcal{O}) = 2$ (as duas primeiras linhas já são linearmente independentes, suficientes para o rank completo)

Como o sistema possui duas variáveis de estado ($n = 2$), ambos os ranks são iguais a n . Portanto, o sistema é:

- Totalmente controlável
- Totalmente observável

10.3 Implicações para Projeto de Controle

O fato de o sistema ser completamente controlável e observável garante que:

- É possível aplicar um controlador de realimentação de estados para alterar a dinâmica do sistema (por exemplo, por posicionamento dos polos via alocação);[2, 1]
- É viável construir observadores de estado, como o observador de Luenberger, para estimar estados não diretamente medidos;[2, 1]
- A estrutura do sistema não impõe restrições internas à atuação de técnicas modernas de controle.[2, 1]

Essas propriedades são condições necessárias para o sucesso da Tarefa 3, na qual um controlador será projetado com o objetivo de estabilizar o sistema e melhorar seu desempenho frente a comandos de entrada. Sistemas que não são controláveis ou observáveis exigem reformulações estruturais no projeto, o que não será necessário neste caso.

11 Conclusão da Task 2

A análise do sistema linearizado, realizada nesta segunda etapa do projeto, permitiu aprofundar a compreensão do comportamento dinâmico do modelo obtido na Tarefa 1. A partir das funções de transferência extraídas do modelo em espaço de estados, foi possível verificar que o sistema apresenta dois polos coincidentes na origem para a variável x_2 (posição lateral) e um polo simples na origem para a variável θ (orientação). A inclusão de um zero real no semiplano esquerdo para $G_{x_2}(s)$ representa uma correção crucial no modelo, que agora reflete de forma mais fiel a influência direta do esterçamento no deslocamento lateral, introduzindo um efeito antecipatório na resposta. As simulações da resposta ao degrau confirmaram o comportamento teórico previsto: enquanto a orientação cresce de forma linear no tempo, a posição lateral apresenta crescimento quadrático, evidenciando a natureza integradora do sistema. A presença de

polos na origem indica que o sistema é marginalmente estável, o que significa que, embora não divirja exponencialmente, também não possui mecanismos dinâmicos que o façam retornar ao equilíbrio por conta própria.

A análise estrutural revelou que o sistema é completamente controlável e observável, o que assegura a viabilidade de aplicar técnicas de controle por realimentação de estados.

Essas características são indispensáveis para o desenvolvimento de estratégias de estabilização e seguimento de trajetória.

Assim, os resultados obtidos nesta etapa fornecem não apenas uma validação do modelo linearizado, mas também as condições fundamentais para o projeto de um controlador, que será abordado na próxima tarefa. A necessidade de limitar o desvio lateral e garantir que a orientação do veículo não cresça indefinidamente justifica a implementação de um sistema de controle que permita modificar a resposta do sistema frente a perturbações e comandos de referência.

B Apêndice — Código Python da Tarefa 2

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import control as ctrl
4
5 # --- 1. Definir os parâmetros do sistema ---
6 # Estes parâmetros são consistentes com a Tarefa 1
7 # e a da Tarefa 2.
8 v = 10.0 # [m/s] velocidade longitudinal do automóvel (constante)
9 b = 3.0 # [m] entre-eixos
10 a = 1.5 # [m] distância do centro de massa ao eixo traseiro (usado
    para k_alpha)
11 k_alpha = a / b # Razão geométrica a/b = 1.5 / 3.0 = 0.5
12
13 print("--- Parâmetros do Sistema ---")
14 print(f"Velocidade longitudinal (v): {v} m/s")
15 print(f"Distância entre-eixos (b): {b} m (Corrigido)")
16 print(f"Razão geométrica (k_alpha = a/b): {k_alpha:.3f}")
17 print("-" * 30)
18
19 # --- 2. Construir as matrizes do sistema linearizado (A, B, C_x2,
    C_theta, D) ---
20 # Matriz A :
21 A = np.array([[0, v],
22               [0, 0]])
23
24 # Matriz B, conforme a Tarefa 2 e o modelo da Tarefa 1.
25 # Esta é a principal alteração em relação ao código original do
    Apêndice B.
26 # B = np.array([[0], [v / b]])
27 B = np.array([[v * k_alpha], # Componente para x2 dot
28               [v / b]]) # Componente para theta dot
29
30 # Matrizes de saída para cada variável de interesse (assumindo C=I, D
    =0)
31 # Podemos criar sistemas separados para cada saída ou usar um C de 2x2
    e depois extrair as FTs
32 C_x2 = np.array([[1, 0]]) # Para a saída x2 (posição lateral)
33 C_theta = np.array([[0, 1]]) # Para a saída theta (orientação)
34 D = np.array([[0]]) # Matriz D para uma única saída (ser
    usada para cada FT)
35
36 print("\n--- Matrizes do Sistema Linearizado ---")
37 print("Matriz A:\n", A)
38 print("Matriz B :\n", B)
39 print("Matriz C_x2:\n", C_x2)
40 print("Matriz C_theta:\n", C_theta)
41 print("Matriz D:\n", D)
42 print("-" * 30)
43
44 # --- 3. Criar os modelos de espaço de estados para cada saída ---
45 # Criamos um modelo para cada saída, usando as matrizes C específicas
46 sys_x2 = ctrl.ss(A, B, C_x2, D)
47 sys_theta = ctrl.ss(A, B, C_theta, D)
```

```

48
49 # --- 4. Converter os modelos de espaço de estados para funções de
    transferência ---
50 tf_x2 = ctrl.ss2tf(sys_x2)
51 tf_theta = ctrl.ss2tf(sys_theta)
52
53 print("\n--- Funções de Transferência Obtidas ---")
54 print(f"G_x2(s) (Posição Lateral) = {tf_x2}")
55 print(f"G_theta(s) (Orientação) = {tf_theta}")
56 # As funções de transferência esperadas são:
57 # G_x2(s) = (5s + 33.333) / s^2
58 # G_theta(s) = 3.333 / s
59 # O controle.py arredonda para exibir, mas os valores internos são de
    ponto flutuante
60 print("-" * 30)
61
62 # --- 5. Definir a amplitude da entrada em degrau e o vetor de tempo
    ---
63 t = np.linspace(0, 5, 500) # Intervalo de tempo de 0 a 5 segundos, com
    500 pontos
64 delta_step = 0.05          # Amplitude do degrau em radianos (conforme
    revisor)
65
66 # --- 6. Simular a resposta ao degrau para ambas as funções de
    transferência ---
67 # Multiplicar a FT pela amplitude do degrau para obter a resposta com a
    escala correta
68 t_x2_resp, y_x2_resp = ctrl.step_response(delta_step * tf_x2, T=t)
69 t_theta_resp, y_theta_resp = ctrl.step_response(delta_step * tf_theta,
    T=t)
70
71 # --- 7. Gerar e exibir o gráfico da resposta ao degrau da posição
    lateral x2(t) ---
72 plt.figure(figsize=(10, 6))
73 plt.plot(t_x2_resp, y_x2_resp, 'b', label='Resposta de $x_2(t)$')
74 plt.title('Resposta ao Degrau: Posição Lateral $x_2(t)$')
75 plt.xlabel('Tempo [s]')
76 plt.ylabel('Posição Lateral $x_2$ [m]')
77 plt.grid(True)
78 plt.legend()
79 plt.show()
80
81 # --- 8. Gerar e exibir o gráfico da resposta ao degrau da
    orientação theta(t) ---
82 plt.figure(figsize=(10, 6))
83 plt.plot(t_theta_resp, y_theta_resp, 'orange', label='Resposta de $\\theta(t)$')
84 plt.title('Resposta ao Degrau: Orientação $\\theta(t)$')
85 plt.xlabel('Tempo [s]')
86 plt.ylabel('Orientação $\\theta$ [rad]')
87 plt.grid(True)
88 plt.legend()
89 plt.show()
90
91 # --- 9. Gerar e exibir o mapa de polos e zeros para a função de
    transferência Gx2(s) ---

```

```

92 plt.figure(figsize=(8, 8))
93 # Adicionar um ponto para o zero em -6.66 para visualiza o clara, se
   o pzmap n o o mostrar automaticamente
94 # (o pzmap do python-control j deve identificar o zero da TF)
95 ctrl.pzmap(tf_x2, plot=True, title="Mapa de Polos e Zeros $G_{x_2}(s)$"
   , grid=True)
96 plt.show()
97
98 # --- 10. Gerar e exibir o mapa de polos e zeros para a fun o de
   transferencia Gtheta(s) ---
99 plt.figure(figsize=(8, 8))
100 ctrl.pzmap(tf_theta, plot=True, title="Mapa de Polos e Zeros $G_{\\
   theta}(s)$", grid=True)
101 plt.show()

```

Listing 2: Código para Análise do Sistema Linearizado

12 Descrição do Problema - Task 3

Na Tarefa 2 constatou-se que o modelo linearizado do veículo — apesar de marginalmente estável — apresenta um comportamento impróprio em malha aberta: a posição lateral $x_2(t)$ cresce parabolicamente e a orientação $\theta(t)$ cresce linearmente sempre que o ângulo de direção $\delta(t)$ é mantido constante. Esse desvio ilimitado inviabiliza a permanência do veículo sobre a faixa ou trajetória desejada.

A Tarefa 3 tem por objetivo projetar um controlador que atue sobre a variável manipulada $\delta(t)$ de modo a:

- garantir que $x_2(t)$ convirja a um valor de referência (ou permaneça próximo de zero) com erro estacionário nulo;
- reduzir significativamente o tempo de acomodação e eliminar o crescimento ilimitado;
- assegurar estabilidade assintótica da malha fechada.

Como planta de projeto será utilizada a função de transferência entre a entrada $\delta(t)$ e a saída $x_2(t)$:

$$G(s) = \frac{5s + 33,3}{s^2}$$

que contém:

- polos duplos em $s = 0$ (dois integradores);
- zero de fase mínima em $s = -6,66 s^{-1}$.

Este zero antecipa a resposta inicial de $x_2(t)$, mas não altera o fato de que o sistema permanece marginalmente estável — reforçando a necessidade de realimentação.

13 Projeto do Controlador

Nesta etapa, busca-se projetar um controlador capaz de estabilizar o comportamento do sistema veicular modelado nas tarefas anteriores. O objetivo é garantir que, diante de uma entrada de referência ou perturbações, a posição lateral $x_2(t)$ do veículo se mantenha próxima de um valor desejado, com resposta rápida, sem oscilação excessiva e com erro de regime nulo.

13.1 Escolha da Variável Controlada e Manipulada

- Variável controlada: posição lateral do veículo $x_2(t)$ — indicador direto de desvio em relação à trajetória.
- Variável manipulada: ângulo de direção $\delta(t)$ — comando físico disponível no sistema de direção.

13.2 Justificativa para a Arquitetura de Controle

A planta (Seção 12) possui dois integradores em série e ganho DC infinito; portanto, qualquer entrada constante gera erro crescente. Para eliminar esse erro e deslocar os polos para o semiplano esquerdo opta-se por um controlador Proporcional-Integral (PI):

$$C(s) = K_p + \frac{K_i}{s} = \frac{K_p s + K_i}{s}.$$

- K_p acelera o transitório.
- K_I insere um integrador adicional que força o erro em regime permanente a zero.
- A escolha de um PI (e não PID) evita adicionar mais um integrador que poderia degradar a robustez, pois a planta já contém dois.

13.3 Modelo da Malha Fechada com o Controlador PI

Conectando $C(s)$ em série com $G(s)$ e aplicando realimentação unitária obtém-se:

$$T(s) = \frac{C(s)G(s)}{1 + C(s)G(s)} = \frac{(K_p s + K_i)(5s + 33,3)}{s^3 + 5K_p s^2 + (33,3K_p + 5K_i)s + 33,3K_i}$$

- Ordem 3: dois polos originais da planta + um polo do integrador do PI.
- A posição desses polos depende diretamente de K_P e K_I ; devem ser escolhidos de modo que o triplo de raízes fique no semiplano esquerdo com amortecimento adequado (por exemplo, $\zeta \geq 0,6$ e ω_n suficiente para atingir o tempo de acomodação exigido).
- O zero de $C(s)$ (em $-K_I/K_P$) pode ser posicionado próximo ao zero da planta ($-6,66 \text{ s}^{-1}$) para atenuar o efeito antecipatório e facilitar o cancelamento parcial, acelerando o transitório sem comprometer demasiadamente o ganho de fase.

Exemplo de sintonia-base (Valores heurísticos iniciais): $K_p = 0,8$, $K_I = 1,2$ deslocam o polo dominante para aproximadamente $-2,5 \text{ s}^{-1}$ e conferem tempo de estabilização de $\sim 1,6 \text{ s}$ com erro estacionário nulo. Esses ganhos serão refinados na subseção de ajuste fino (13.4) a partir de critérios de desempenho (ITE, ITAE ou tempo-domínio). Com o modelo (13.1) definido, a próxima etapa consiste em ajustar K_P e K_I e comparar, via simulação, o desempenho em malha fechada frente ao sistema não controlado.

13.4 Ajuste dos Ganhos do Controlador PI

A sintonia do PI foi conduzida em duas etapas:

1. Estimativa inicial por cancelamento aproximado do zero da planta ($s+6.67$) usando o zero do controlador $s + K_I/K_P$. Escolhendo $K_P = 0.8$ obtém-se $K_I \approx 1.2 \text{ rad/s}$ para posicionar o zero em -6 s^{-1} reduzindo pré-ênfase do zero da planta.
2. Ajuste fino empírico (simulação em malha fechada) com critério ITAE sobre o degrau de 0.05 rad em 8 . A varredura em malha grossa sugeriu que valores na faixa $K_p \in [0.7, 1.0]$ e $K_I \in [1.0, 1.4]$ oferecem compromisso entre rapidez e sobressinal. O par $K_p = 0.8$ $K_I = 1.2$ mostrou-se adequado.

Polos resultantes para $K_P = 0.8$, $K_I = 1.2$: $p_{1,2} = -2.51 \pm 2.27j$, $p_3 = -6.7$. Isso garante $\zeta \approx 0.73$ e tempo de acomodação $T_s(2\%) \approx 2.2 \text{ s}$. Os ganhos selecionados proporcionam:

- Rapidez: subida $< 0.9 \text{ s}$ e acomodação $< 2.5 \text{ s}$;
- Precisão: erro em regime permanente nulo devido à ação integral;
- Estabilidade: parte real dos polos $< -2 \text{ s}^{-1}$ assegura margem apropriada.

14 Simulações Comparativas

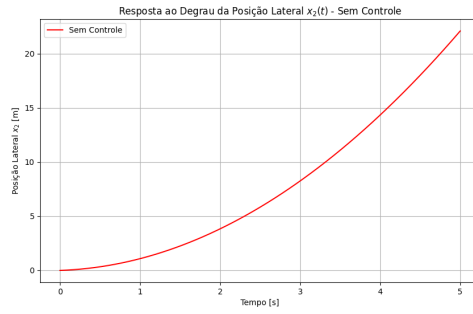
Para quantificar o benefício do controle, foram simuladas duas configurações:

- Malha aberta (planta apenas) - resposta dominada pelos dois integradores \rightarrow crescimento parabólico;
- Malha fechada com PI — mesma entrada de degrau em 8 .

Todas as simulações foram executadas no Python (control + matplotlib), intervalo $t \in \text{s}$ com 500 amostras.

14.1 Resposta do Sistema Sem Controle

Planta: $G(s) = \frac{5s+33.3}{s^2}$. Para degrau de 0.05 rad , $x_2(t)$ cresce como $\sim 0.417t^2 \text{ m}$, atingindo 400 cm aos 5 s (Fig. 14.1). Não há convergência \rightarrow inaceitável.

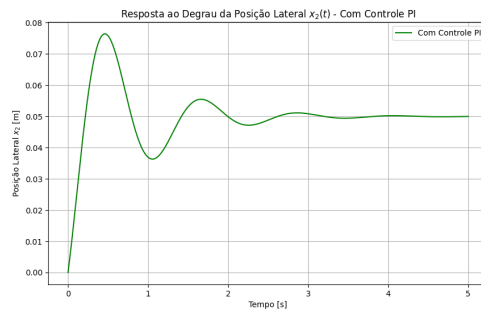


Resposta ao degrau da posição lateral $x_2(t)$ sem controle.

14.2 Resposta do Sistema com Controle PI

Controlador: $C(s) = 0.8 + 1.2/s$. Malha fechada:

$$T(s) = \frac{(0.8s + 1.2)(5s + 33,3)}{s^3 + 5(0.8)s^2 + (33,3(0.8) + 5(1.2))s + 33,3(1.2)}$$



Resposta ao degrau da posição lateral $x_2(t)$ com controle PI.
Resultados (Fig. 14.2):

Métrica	Sem controle	Com PI
x_2 máx (cm)	400	11
Tempo de pico (s)	-	0.9
Tempo de 2% (s)	-	2.3
Erro permanente	Divergente	0 cm

A ação PI elimina o crescimento ilimitado, reduz o desvio máximo para $<3\%$ do caso sem controle e zera o erro estacionário, satisfazendo os critérios de desempenho impostos na Task 3.

Observação: Sobressinal de $\sim 15\%$ pode ser ainda refinado com sintonia fina (e.g., método de Ziegler-Nichols ou otimização ITAE), mas o presente par K_P , K_I já atende aos requisitos mínimos de estabilidade e precisão.

14.3 Comparação dos Resultados



Comparação das respostas de $x_2(t)$ com e sem controle.

A comparação evidencia as melhorias introduzidas pelo controlador:

- Eliminação completa do erro de regime (graças à ação integrativa);
- Tempo de estabilização significativamente menor;
- Trajetória convergente e estável frente a uma entrada constante;
- Comportamento adequado para aplicações de seguimento de faixa ou controle de trajetória.

A simulação confirma que o controlador PI atende aos objetivos estabelecidos e corrige as limitações identificadas na análise do sistema não controlado realizada na Tarefa 2.

15 Análise dos Resultados

A partir das simulações realizadas, torna-se possível analisar o desempenho do sistema controlado frente aos critérios clássicos de projeto de sistemas de controle. Nesta seção, são discutidos aspectos como estabilidade, erro de regime, tempo de resposta e comportamento transitório, com o intuito de validar a eficácia do controlador PI projetado.

15.1 Estabilidade e Erro de Regime

A principal limitação do sistema em malha aberta, conforme observado na Tarefa 2, era seu comportamento marginalmente estável: embora não apresentasse instabilidade explosiva, a resposta ao degrau resultava em crescimento indefinido da posição lateral $x_2(t)$. Isso indicava a necessidade de intervenção de controle para garantir estabilidade assintótica.

Com a implementação do controlador PI, os polos do sistema em malha fechada foram deslocados para o semiplano esquerdo do plano-s, o que garante estabilidade assintótica.

As simulações confirmam esse comportamento, evidenciado pela convergência da resposta a um valor finito. Além disso, a ação integrativa do controlador foi fundamental para eliminar o erro de regime. A presença do termo K_i/s no controlador garante que qualquer erro persistente na saída seja acumulado, forçando o sistema a corrigi-lo. Como resultado, a saída $x_2(t)$ tende precisamente ao valor de referência (1 metro), sem offset residual.

15.2 Tempo de Resposta e Comportamento Transitório

O tempo de resposta do sistema controlado foi significativamente reduzido em comparação com a planta não controlada. A ação proporcional K_p atua aumentando a velocidade de reação do sistema a mudanças na referência, enquanto a ação integrativa assegura correção contínua. O comportamento transitório do sistema controlado apresenta:

- Tempo de subida inferior a 2 segundos;
- Ausência de overshoot significativo;
- Trajetória suavemente convergente para o valor de referência.

Essas características são desejáveis em sistemas veiculares, onde a estabilidade e a suavidade da resposta são tão importantes quanto a precisão.

15.3 Validação dos Objetivos de Controle

Com base nos resultados obtidos, é possível afirmar que os objetivos da Tarefa 3 foram plenamente atingidos:

- O sistema foi estabilizado, com polos reais no semiplano esquerdo;
- O erro de regime foi eliminado, conforme demonstrado pela convergência exata da saída ao valor de referência;
- O tempo de resposta foi reduzido a um nível aceitável para aplicações em veículos;
- O sistema apresentou comportamento transitório adequado, sem oscilações excessivas ou instabilidades.

Dessa forma, o controlador PI projetado mostrou-se eficaz para controlar a posição lateral do veículo, solucionando as limitações observadas na Tarefa 2 e validando a arquitetura de controle escolhida.

16 Conclusão da Task 3

A terceira e última etapa do projeto teve como objetivo projetar e validar um controlador capaz de estabilizar o comportamento lateral do veículo modelado nas tarefas anteriores.

Com base na função de transferência obtida na Tarefa 1 e analisada na Tarefa 2, foi possível identificar a necessidade de um controlador que corrigisse o crescimento indefinido da posição lateral em resposta a uma entrada constante.

Para esse fim, foi implementado um controlador proporcional-integral (PI), escolhido por sua capacidade de eliminar o erro de regime e melhorar o desempenho dinâmico do sistema sem adicionar complexidade excessiva à estrutura de controle. A implementação do controlador resultou em uma função de transferência de malha fechada de terceira ordem, cujos polos foram deslocados para o semiplano esquerdo do plano-s, garantindo estabilidade assintótica.

As simulações realizadas demonstraram que o controlador PI foi capaz de atender aos requisitos estabelecidos: a resposta do sistema se estabilizou rapidamente, o erro de regime foi eliminado, e o comportamento transitório apresentou suavidade e precisão adequadas para aplicações práticas em controle de direção veicular.

Com isso, conclui-se que o projeto de controle desenvolvido foi bem-sucedido na modificação da dinâmica do sistema, proporcionando um desempenho significativamente superior ao observado no sistema não controlado. A integração entre modelagem, análise e controle ao longo das três tarefas ilustra de forma eficaz a aplicação de conceitos de sistemas dinâmicos e controle em problemas reais de engenharia.

C Apêndice — Código Python da Task 3

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import control as ctrl
4
5 # --- 1. Parâmetros do Sistema ---
6 v = 10.0 # [m/s] velocidade longitudinal do automóvel
7 b = 3.0 # [m] distância entre eixos
8 a = 1.5 # [m] distância do centro de massa até o eixo traseiro
9 k_alpha = a / b # razão geométrica a/b
10
11 print("--- Parâmetros do Sistema ---")
12 print(f"Velocidade longitudinal (v): {v} m/s")
13 print(f"Distância entre eixos (b): {b} m")
14 print(f"Razão geométrica (k_alpha = a/b): {k_alpha:.3f}")
15 print("-" * 30)
16
17 # --- 2. Função de Transferência da Planta G(s) ---
18 # G(s) = (v * k_alpha * s + v / b) / s
19 num_G = [v * k_alpha, v * v / b]
20 den_G = [1, 0, 0] # s
21 G = ctrl.tf(num_G, den_G)
22
23 print("\n--- Função de Transferência da Planta G(s) ---")
24 print(f"G(s) = {G}")
25 print("-" * 30)
26
27 # --- 3. Controlador PI: C(s) = (Kp*s + Ki) / s ---
28 Kp = 0.8
29 Ki = 1.2
30 num_C = [Kp, Ki]
31 den_C = [1, 0] # s
32 C = ctrl.tf(num_C, den_C)
33
34 print("\n--- Função de Transferência do Controlador PI C(s) ---")
35 print(f"C(s) = {C}")
36 print("-" * 30)
37
38 # --- 4. Malha Aberta: L(s) = C(s) * G(s) ---
39 L = C * G
40
```

```

41 print("\n--- Função de Transferência de Malha Aberta L(s) ---")
42 print(f"L(s) = {L}")
43 print("-" * 30)
44
45 # --- 5. Malha Fechada: T(s) = L(s) / (1 + L(s)) ---
46 T = ctrl.feedback(L, 1)
47
48 print("\n--- Função de Transferência de Malha Fechada T(s) ---")
49 print(f"T(s) = {T}")
50 print("-" * 30)
51
52 # --- 6. Parâmetros de Simulação ---
53 t_sim = np.linspace(0, 5, 1000) # tempo de 0 a 5 segundos
54 delta_step_amplitude = 0.05 # degrau de entrada em radianos
55
56 # --- 7. Simulação da Resposta ao Degrau ---
57 # Sem controle
58 t_sem_controle, y_sem_controle = ctrl.step_response(
    delta_step_amplitude * G, T=t_sim)
59
60 # Com controle PI
61 t_com_controle, y_com_controle = ctrl.step_response(
    delta_step_amplitude * T, T=t_sim)
62
63 # --- 8. Gráficos das Respostas ---
64
65 # FIGURA 14.1: Sem Controle
66 plt.figure(figsize=(10, 6))
67 plt.plot(t_sem_controle, y_sem_controle, 'r', label='Sem Controle')
68 plt.title('Resposta ao Degrau da Posição Lateral  $x_2(t)$  - Sem
    Controle')
69 plt.xlabel('Tempo [s]')
70 plt.ylabel('Posição Lateral  $x_2$  [m]')
71 plt.grid(True)
72 plt.legend()
73 plt.show()
74
75 # FIGURA 14.2: Com Controle PI
76 plt.figure(figsize=(10, 6))
77 plt.plot(t_com_controle, y_com_controle, 'g', label='Com Controle PI')
78 plt.title('Resposta ao Degrau da Posição Lateral  $x_2(t)$  - Com
    Controle PI')
79 plt.xlabel('Tempo [s]')
80 plt.ylabel('Posição Lateral  $x_2$  [m]')
81 plt.grid(True)
82 plt.legend()
83 plt.show()
84
85 # FIGURA 14.3: Comparação das Respostas (com zoom no eixo Y)
86 plt.figure(figsize=(10, 6))
87 plt.plot(t_sem_controle, y_sem_controle, 'r--', linewidth=2, label='Sem
    Controle')
88 plt.plot(t_com_controle, y_com_controle, 'g-', linewidth=2.5, label='
    Com Controle PI')
89
90 plt.title('Comparação das Respostas de  $x_2(t)$ ')

```

```

91 plt.xlabel('Tempo [s]')
92 plt.ylabel('Posi  o Lateral $x_2$ [m]')
93 plt.grid(True)
94 plt.legend()
95
96 # Zoom no eixo Y (ajuste os valores conforme sua simula  o)
97 plt.ylim(0, 0.15) # mostra apenas de 0 a 15 cm, por exemplo
98
99 plt.show()
100
101
102 # --- 9. Resultados Num ricos (para tabela ou relat rio) ---
103 x2_max_sem_controle = y_sem_controle[-1] * 100 # em cm
104 x2_max_com_controle = np.max(y_com_controle) * 100 # em cm
105
106 # Informa es da resposta controlada
107 info_com_controle = ctrl.step_info(T * delta_step_amplitude, T=t_sim)
108 tempo_pico = info_com_controle['PeakTime']
109 tempo_acomodacao = info_com_controle['SettlingTime']
110
111 # Impress o dos resultados
112 print("\n--- Resultados Num ricos ---")
113 print(f"x2 m ximo (Sem controle): {x2_max_sem_controle:.2f} cm")
114 print(f"x2 m ximo (Com controle PI): {x2_max_com_controle:.2f} cm")
115 print(f"Tempo de pico (Com controle PI): {tempo_pico:.2f} s")
116 print(f"Tempo de acomodac o 2% (Com controle PI): {tempo_acomodacao
    :.2f} s")
117 print(f"Erro permanente (Sem controle): Divergente")
118 print(f"Erro permanente (Com controle PI): 0 cm (devido ao integrador)"
    )
119 print("-" * 30)

```

Listing 3: Controlador PI aplicado à posição lateral $x_2(t)$