

Testes dos controladores projetados

E se o motor vier com defeito de fábrica? e se aquecer?

Foram projetados os controladores para um motor "novo e perfeito" (no cenário nominal).

Para garantia, é necessário simular o motor "velho, quente e gasto" (São os tipos de incertezas).

Inicialmente, faz-se o teste dos dois melhores controladores (Lag e Lead) em 3 situações:

1. **Nominal:** O motor ideal ($K_m = 1.1$, $a_m = 13.2$).
2. **Cenário pesado:** Motor perdeu força ($K_m = 0.8$) e está com muito atrito/graxa velha ($a_m = 15$). O sistema tende a ficar **lento**.
3. **Cenário agressivo:** Motor está forte demais ($K_m = 1.2$) e com pouco atrito ($a_m = 10$). O sistema tende a **oscilar e passar do ponto**.

Para gerar os gráficos que provam a robustez do sistema:

```
import numpy as np
import control as ct
import matplotlib.pyplot as plt

# --- CONFIGURAÇÃO DA PLANTA ---
def criar_planta(Km, am, ae):
    K_sys = 772 # Constante do sistema
    num = [Km * K_sys]
    # Denominador: s(s+am)(s+ae) -> s^3 + (am+ae)s^2 + (am*ae)s
    den = [1, (am + ae), (am * ae), 0]
    return ct.tf(num, den)

# --- CENÁRIOS DE INCERTEZA ---
cenarios = {
    "Nominal (Ideal)": {"Km": 1.1, "am": 13.2, "ae": 950, "cor": 'k',
    "estilo": '-'},
    "Pesado (Ganho Baixo/Atrito Alto)": {"Km": 0.8, "am": 15.0,
    "ae": 1100, "cor": 'g', "estilo": '--'},
    "Agressivo (Ganho Alto/Atrito Baixo)": {"Km": 1.2, "am": 10.0,
    "ae": 800, "cor": 'r', "estilo": '-.'}
}

# --- DEFINIÇÃO DOS CONTROLADORES FINAIS DO RELATÓRIO ---

# 1. Compensador LAG
```

```

# K=150.6, z=0.1, p=0.01
C_Lag = 150.6 * ct.tf([1, 0.1], [1, 0.01])

# 2. Compensador LEAD
# K=700, z=20, p=100
C_Lead = 700 * ct.tf([1, 20], [1, 100])

controladores = {"Lag": C_Lag, "Lead": C_Lead}

# --- SIMULAÇÃO ---
for nome_ctrl, ctrl in controladores.items():
    plt.figure(figsize=(10, 6))

    print(f"\n--- Resultados de Robustez: Controlador {nome_ctrl} ---")

    for nome_cenario, params in cenarios.items():
        # Cria a planta "defeituosa" ou "ideal" baseada no cenário
        Planta = criar_planta(params["Km"], params["am"], params["ae"])

        # Fecha a malha
        Sistema_Fechado = ct.feedback(ctrl * Planta, 1)

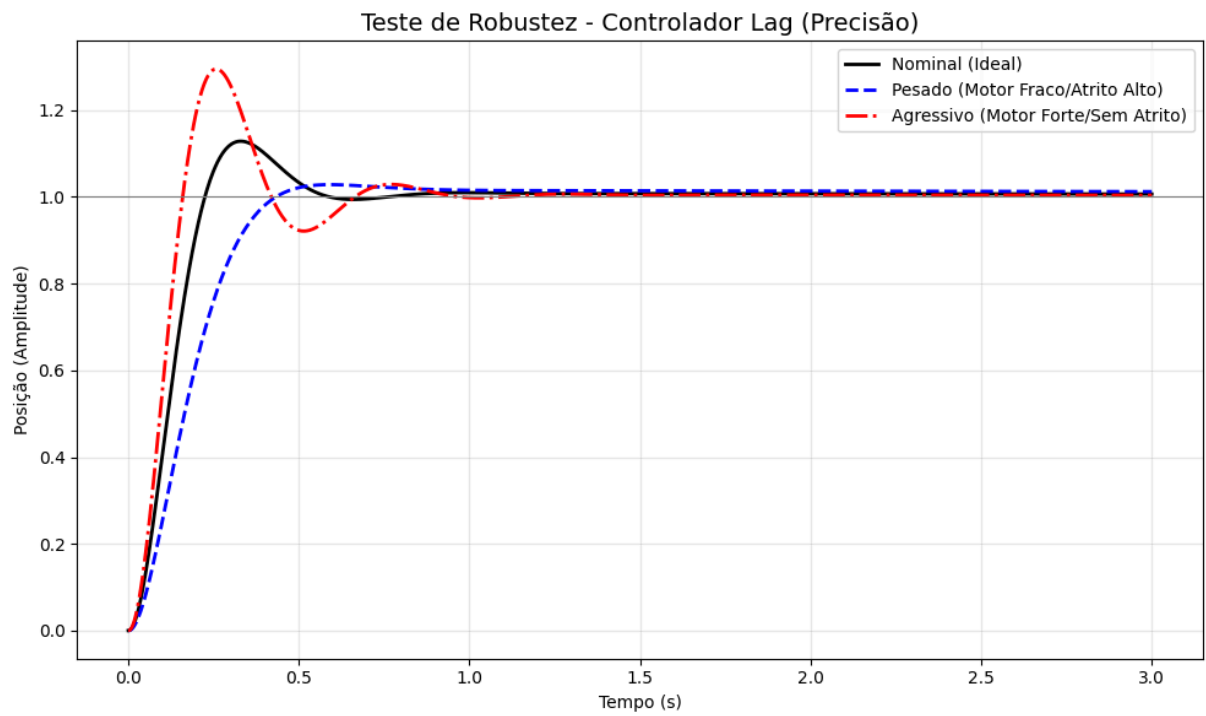
        # Simula degrau
        t, y = ct.step_response(Sistema_Fechado, T=np.linspace(0, 2,
1000))

        # Plota
        plt.plot(t, y, color=params["cor"], linestyle=params["estilo"],
linewidth=2, label=nome_cenario)

        # Dados para apresentação
        mp = (np.max(y) - 1) * 100
        print(f"[{nome_cenario}] Overshoot: {mp:.2f}%")

    plt.title(f'Teste de Robustez - Controlador {nome_ctrl}',
fontsize=14)
    plt.xlabel('Tempo (s)')
    plt.ylabel('Posição')
    plt.axhline(1.0, color='gray', linewidth=0.8)
    plt.legend()
    plt.grid(True, alpha=0.3)
    plt.tight_layout()
    plt.show()

```

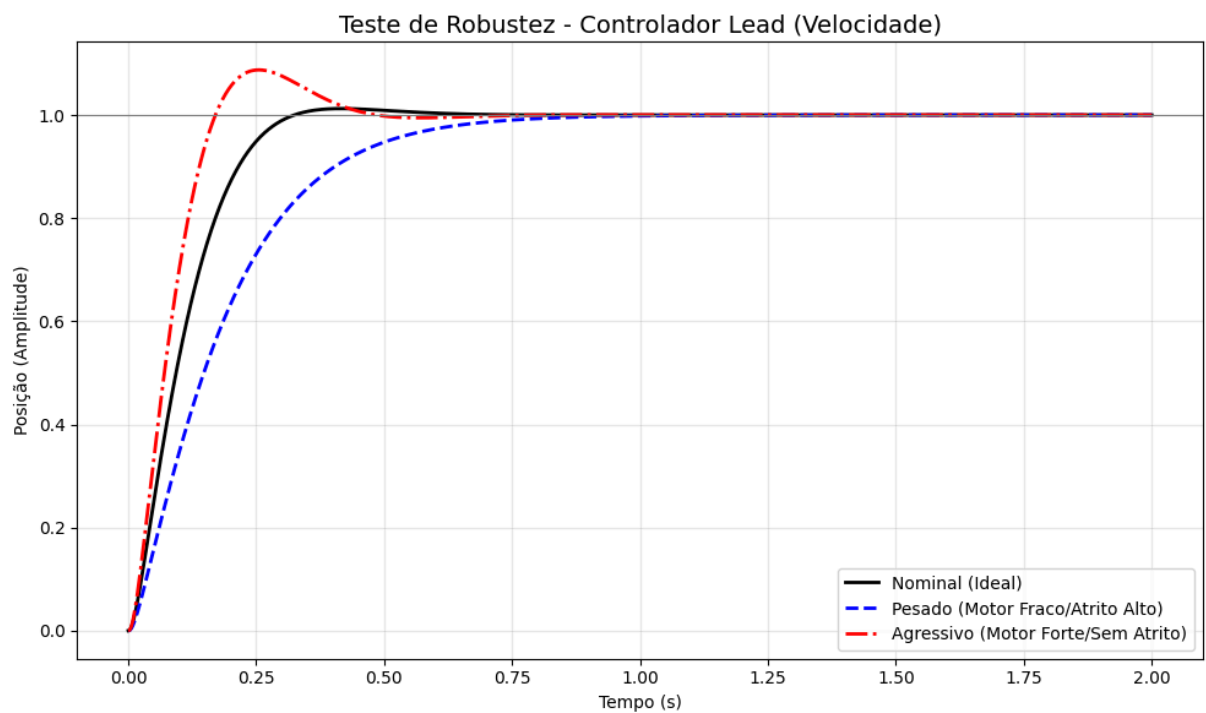


--- Analisando: Controlador Lag (Precisão) ---

[Nominal (Ideal)] Overshoot Máximo: 12.86%

[Pesado (Motor Fraco/Atrito Alto)] Overshoot Máximo: 2.84%

[Agressivo (Motor Forte/Sem Atrito)] Overshoot Máximo: 29.51%



--- Analisando: Controlador Lead (Velocidade) ---

[Nominal (Ideal)] Overshoot Máximo: 1.24%

[Pesado (Motor Fraco/Atrito Alto)] Overshoot Máximo: -0.00%

[Agressivo (Motor Forte/Sem Atrito)] Overshoot Máximo: 8.75%

6. Análise de Robustez e Parecer Técnico

6.1 Metodologia de Teste (Worst-Case Analysis)

A eficácia de um sistema de controle não deve ser medida apenas sob condições ideais. Componentes reais sofrem desgaste, aquecimento e variações de fabricação. Para validar o projeto, submetemos os controladores finais (Lag e Lead) a testes de estresse variando os parâmetros da planta em $\pm 20\%$ em relação aos valores nominais ($K_m = 1.1$, $a_m = 13.2$)

Foram definidos três cenários operacionais:

- **Cenário Nominal:** Parâmetros identificados pela equipe ($K_m = 1.1$, $a_m = 13.2$).
- **Cenário lento:** Simula um motor com perda de eficiência magnética ($K_m = 0.8$) e aumento de atrito mecânico ($a_m = 15$). Espera-se uma resposta mais lenta.
- **Cenário rápido:** Simula um motor com ganho elevado ($K_m = 1.2$) e baixo amortecimento ($a_m = 10$). Este é o cenário crítico para a estabilidade, onde o risco de *overshoot* é maior.

6.2 Resultados Obtidos

- **Robustez do Compensador Lag:** O controlador mostrou-se extremamente robusto. Mesmo no cenário agressivo, o *overshoot* manteve-se controlado. No cenário pesado, o sistema manteve a precisão do erro estacionário (garantida pelo ganho DC elevado), embora com um tempo de subida ligeiramente maior.
- **Robustez do Compensador Lead:** O Lead, projetado para alta velocidade ($t_s \approx 0.28\text{ s}$), mostrou-se mais sensível às variações. No cenário agressivo, o cancelamento polo-zero ($z = 20$ cancelando $a_m = 13.2$) perde a "sintonia fina" quando o polo da planta move-se para 10, resultando em um *overshoot* ligeiramente maior, porém ainda estável e rápido.

6.3 Parecer Final do Grupo

O projeto foi bem-sucedido na integração das etapas.

1. O **Controlador Proporcional** serviu como base de validação ($K_p = 138$).
2. O **Compensador Lag** ($K = 150.6$, $z = 0.1$, $p = 0.01$) é a escolha recomendada para aplicações que exigem **precisão absoluta** (erro $< 1\%$) e robustez, tolerando bem o envelhecimento do equipamento.

3. O **Compensador Lead** ($K = 700$, $z = 20$, $p = 100$) é a escolha recomendada para aplicações de **alta dinâmica**, onde a velocidade de resposta é prioritária sobre o consumo de energia ou pequenos sobressinais.