LABORATÓRIO DE CONTROLE

Relatório T1
Simulação de Sistemas de Controle

PUCRS

Autores:
Gabriel Eduardo Decian, Daniela Schardong

3 de novembro de 2025

Sumário

Lista de Scripts

Lista de Figuras

Lista de Equações

1 Introdução

Este trabalho apresenta a análise do comportamento de um sistema de controle PI (Proporcional-Integral) sem ação derivativa através de simulações computacionais no ambiente MATLAB/Simulink. O estudo tem como objetivo investigar os efeitos dos parâmetros do controlador PI na resposta ao degrau unitário de um sistema de segunda ordem.

O controlador PI é amplamente utilizado em aplicações industriais devido à sua capacidade de eliminar o erro de regime permanente através da ação integral, mantendo uma resposta dinâmica adequada através da ação proporcional. A análise paramétrica dos ganhos K_p (proporcional) e T_i (tempo integral) é fundamental para compreender o comportamento do sistema em malha fechada e estabelecer critérios de projeto adequados.

A função de transferência do controlador PI é dada por:

$$C(s) = K_p\left(1 + \frac{1}{T_i s}\right) = K_p + \frac{K_i}{s} \tag{1}$$

onde $K_i = K_p/T_i$ é o ganho integral.

O sistema em análise consiste em uma planta de segunda ordem com função de transferência:

$$G(s) = \frac{100}{s^2 + 6s + 100} \tag{2}$$

Este relatório investiga sistematicamente o comportamento do sistema em malha fechada para diferentes combinações dos parâmetros $K_p \in \{0.5, 1, 2, 10\}$ e $T_i \in \{\infty, 10, 1, 0.1\}$ segundos, analisando aspectos como estabilidade, tempo de resposta, overshoot e erro de regime permanente.

2 Modelo no Simulink

Para realizar as simulações e análises propostas neste trabalho, foi desenvolvido um modelo no ambiente Simulink que representa o sistema de controle PI em malha fechada. O modelo implementa a estrutura clássica de controle por realimentação unitária negativa, permitindo a variação sistemática dos parâmetros do controlador durante as simulações.

A Figura 1 apresenta a estrutura completa do modelo desenvolvido no Simulink.

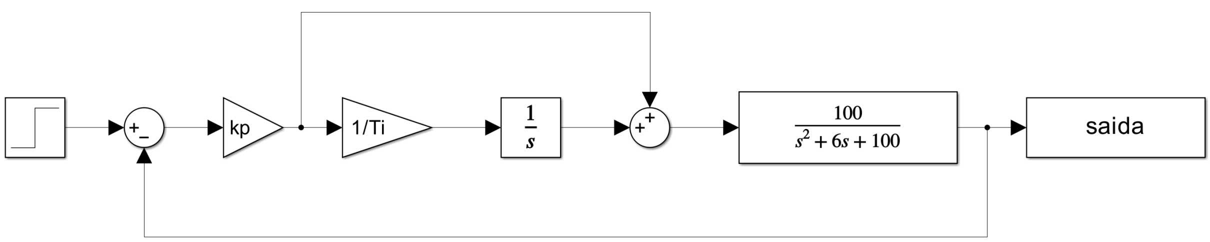

Figura 1: Modelo criado no Simulink baseado no modelo encontrado no enunciado

O modelo é composto pelos seguintes elementos principais:

- **Step:** Gerador de sinal degrau unitário que serve como referência para o sistema;
- **Sum:** Somador que calcula o erro entre a referência e a saída realimentada;
- **Gain (Kp):** Bloco que implementa o ganho proporcional do controlador;
- **Integrator (1/s):** Integrador que implementa a ação integral;

- **Gain1 (1/Ti):** Ganho que define o tempo integral do controlador;
- **Transfer Function:** Função de transferência da planta $G(s) = \frac{100}{s^2+6s+100}$;
- **To Workspace:** Bloco que armazena os dados de simulação no workspace do MATLAB.

A configuração permite a modificação automática dos parâmetros K_p e T_i através de scripts MATLAB, facilitando a análise paramétrica sistemática do comportamento do sistema. O caso especial $T_i = \infty$ (controlador puramente proporcional) é implementado configurando-se o ganho do bloco Gain1 para um valor próximo de zero (ε), conforme recomendado no enunciado.

Matematicamente, quando $T_i \to \infty$, o ganho integral é calculado como:

$$\text{Gain1} = \frac{1}{T_i} = \frac{1}{\frac{1}{\varepsilon}} = \varepsilon \tag{3}$$

Onde ε representa o menor número positivo representável no MATLAB (eps), efetivamente eliminando a ação integral do controlador.

3 Script com base do Módulo Simulink

Para automatizar a variação dos parâmetros do controlador PI e a geração dos gráficos de resposta ao degrau, foi desenvolvido um script em MATLAB que interage diretamente com o modelo Simulink. O script executa as simulações para todas as combinações de K_p e T_i especificadas no enunciado, armazena os resultados e gera os gráficos conforme solicitado.

A seguir, apresenta-se o código-fonte do script utilizado:

```matlab
% Valores especificados no enunciado
Kp_values = [0.5, 1, 2, 10];
Ti_values = [Inf, 10, 1, 0.1];

% Parametros da simulacao
t_final = 20;         % Duracao da simulacao (s)
step_size = 0.001;    % Passo fixo

% Nome do modelo Simulink
modelName = 'modeloPIsemD';
load_system(modelName);

% Configuracao de cores e estilos para os graficos
colors = [0 0.4470 0.7410;             % Azul (Kp=0.5)
          0.8500 0.3250 0.0980;        % Laranja (Kp=1)
          0.2 0.7 0.2;                 % Verde (Kp=2)
          0.6 0.1 0.6];                % Roxo (Kp=10)
styles = {'-', '--', ':', '-.'};     % Estilos de linha distintos
linewidths = [1.5, 2.0, 1.5, 2.0]; % Espessuras variadas

% Configura parametros do modelo
set_param(modelName, ...
    'StopTime', num2str(t_final), ...
    'Solver', 'ode45', ...
    'FixedStep', num2str(step_size), ...
    'MaxStep', num2str(step_size), ...
    'SaveOutput', 'on', ...
    'SaveTime', 'on', ...
```

```matlab
    'SaveFormat', 'StructureWithTime', ...
    'SignalLogging', 'on', ...
    'SignalLoggingName', 'logsout');

%% Gerar 4 figuras conforme enunciado: uma para cada Ti
for j = 1:length(Ti_values)
    Ti_val = Ti_values(j);

    % Tratamento do Ti infinito conforme enunciado
    if isinf(Ti_val)
        Ti = 1/eps;                 % Ti infinito
        Gain1_value = eps;          % 1/Ti = 1/(1/eps) = eps
        Ti_str = 'inf';
        Ti_display = '\infty';
    else
        Ti = Ti_val;
        Gain1_value = 1/Ti;         % 1/Ti normal
        Ti_str = strrep(num2str(Ti_val), '.', '_');
        Ti_display = num2str(Ti_val);
    end

    % Debug: mostrar valores configurados
    fprintf('Figura %d: Ti = %s, Gain1 = %.6f\n', j, Ti_display, Gain1_value);

    % Criar nova figura
    figure('Position', [100 + j*50, 100 + j*50, 800, 600], 'Color', 'white');
    hold on;

    for i = 1:length(Kp_values)
        Kp = Kp_values(i);

        % Configurar os ganhos no modelo Simulink
        set_param([modelName '/Gain'], 'Gain', num2str(Kp));         % Kp
        set_param([modelName '/Gain1'], 'Gain', num2str(Gain1_value)); % 1/Ti

        % Executar simulacao
        sim(modelName);

        % Extrair dados da simulacao
        t = saida.time;
        y = saida.signals.values;

        % Plotar curva
        plot(t, y, 'Color', colors(i,:), 'LineStyle', styles{i}, ...
             'LineWidth', linewidths(i), 'DisplayName', ['K_p = ' num2str(Kp)]);
    end

    % Formatacao do grafico
```

```
    if isinf(Ti_val)
        title_str = 'Resposta ao Degrau Unitario - T_i = \infty (Controlador P)';
        filename = 'grafico_Ti_inf';
    else
        title_str = ['Resposta ao Degrau Unitario - T_i = ' num2str(Ti_val) ' s'];
        filename = ['grafico_Ti_' Ti_str];
    end

    % Configurar fundo branco e textos pretos
    set(gcf, 'Color', 'white');  % Fundo da figura branco
    set(gca, 'Color', 'white');  % Fundo dos eixos branco

    title(title_str, 'FontSize', 14, 'FontWeight', 'bold', 'Color', 'black');
    xlabel('Tempo (s)', 'FontSize', 12, 'FontWeight', 'bold', 'Color', 'black');
    ylabel('Saida y(t)', 'FontSize', 12, 'FontWeight', 'bold', 'Color', 'black');

    % Configurar eixos e grid com cores pretas
    set(gca, 'XColor', 'black', 'YColor', 'black', 'FontSize', 11);
    set(gca, 'GridColor', 'black', 'GridAlpha', 0.3);

    legend('Location', 'best', 'FontSize', 11, 'TextColor', 'black', ...
        'EdgeColor', 'black', 'Color', 'white');
    grid on;
    xlim([0 t_final]);

    % Linha de referencia no setpoint
    yline(1, 'k--', 'LineWidth', 1, 'Alpha', 0.5, 'HandleVisibility', 'off');

    hold off;

    % Salvar grafico
    saveas(gcf, [filename '.png']);
    saveas(gcf, [filename '.fig']);

    fprintf('Figura %d salva: %s\n', j, filename);
end

%% Gerar 4 figuras adicionais conforme enunciado: uma para cada Kp
for i = 1:length(Kp_values)
    Kp = Kp_values(i);
    Kp_str = strrep(num2str(Kp), '.', '_');

    % Debug: mostrar Kp configurado
    fprintf('Figura %d: Kp = %.1f\n', i+4, Kp);

    % Criar nova figura
```

```
    figure('Position', [200 + i*50, 200 + i*50, 800, 600], 'Color', 'white');
    hold on;

    for j = 1:length(Ti_values)
        Ti_val = Ti_values(j);

        % Tratamento do Ti infinito conforme enunciado
        if isinf(Ti_val)
            Ti = 1/eps;                 % Ti infinito
            Gain1_value = eps;          % 1/Ti = 1/(1/eps) = eps
            Ti_display = '\infty';
        else
            Ti = Ti_val;
            Gain1_value = 1/Ti;         % 1/Ti normal
            Ti_display = num2str(Ti_val);
        end

        % Configurar os ganhos no modelo Simulink
        set_param([modelName '/Gain'], 'Gain', num2str(Kp));         % Kp
        set_param([modelName '/Gain1'], 'Gain', num2str(Gain1_value)); % 1/Ti

        % Executar simulacao
        sim(modelName);

        % Extrair dados da simulacao
        t = saida.time;
        y = saida.signals.values;

        % Plotar curva
        plot(t, y, 'Color', colors(j,:), 'LineStyle', styles{j}, ...
             'LineWidth', linewidths(j), 'DisplayName', ['T_i = ' Ti_display ' s']);
    end

    % Formatacao do grafico
    title_str = ['Resposta ao Degrau Unitario - K_p = ' num2str(Kp)];
    filename = ['grafico_Kp_' Kp_str];

    % Configurar fundo branco e textos pretos
    set(gcf, 'Color', 'white');  % Fundo da figura branco
    set(gca, 'Color', 'white');  % Fundo dos eixos branco

    title(title_str, 'FontSize', 14, 'FontWeight', 'bold', 'Color', 'black');
    xlabel('Tempo (s)', 'FontSize', 12, 'FontWeight', 'bold', 'Color', 'black');
    ylabel('Saida y(t)', 'FontSize', 12, 'FontWeight', 'bold', 'Color', 'black');

    % Configurar eixos e grid com cores pretas
```

```
    set(gca, 'XColor', 'black', 'YColor', 'black', 'FontSize', 11);
    set(gca, 'GridColor', 'black', 'GridAlpha', 0.3);

    legend('Location', 'best', 'FontSize', 11, 'TextColor', 'black', ...
           'EdgeColor', 'black', 'Color', 'white');
    grid on;
    xlim([0 t_final]);

    % Linha de referencia no setpoint
    yline(1, 'k--', 'LineWidth', 1, 'Alpha', 0.5, 'HandleVisibility', 'off');

    hold off;

    % Salvar grafico
    saveas(gcf, [filename '.png']);
    saveas(gcf, [filename '.fig']);

    fprintf('Figura %d salva: %s\n', i+4, filename);
end

% Fechar o modelo Simulink
close_system(modelName, 0);

fprintf('\n=== SIMULACAO CONCLUIDA ===\n');
fprintf('8 figuras geradas conforme enunciado:\n');
fprintf('Graficos por Ti (4 figuras):\n');
fprintf('- grafico_Ti_inf\n');
fprintf('- grafico_Ti_10\n');
fprintf('- grafico_Ti_1\n');
fprintf('- grafico_Ti_0_1\n');
fprintf('Graficos por Kp (4 figuras):\n');
fprintf('- grafico_Kp_0_5\n');
fprintf('- grafico_Kp_1\n');
fprintf('- grafico_Kp_2\n');
fprintf('- grafico_Kp_10\n');
```

Listing 1: Script MATLAB para simulação do sistema PI no Simulink

O script realiza as seguintes etapas principais:

- Carrega o modelo Simulink e configura os parâmetros de simulação (tempo, passo, solver);
- Varre os valores de K_p e T_i conforme o enunciado, ajustando os ganhos no modelo;
- Executa as simulações e armazena as respostas no workspace;
- Gera e salva os gráficos de resposta ao degrau, agrupando-os por T_i e por K_p;
- Facilita a análise paramétrica e a comparação dos resultados.

A utilização deste script garante reprodutibilidade, agilidade e precisão na análise dos efeitos dos parâmetros do controlador PI sobre o sistema em estudo.

4 Resultados e Análise

4.1 Gráficos com base em Ti

4.1.1 $T_i = \infty$ (Controlador P)

A Figura 2 apresenta a resposta ao degrau unitário do sistema com controlador puramente proporcional ($T_i = \infty$), onde a ação integral é eliminada. Este caso representa o comportamento fundamental de um controlador proporcional puro para diferentes ganhos K_p.

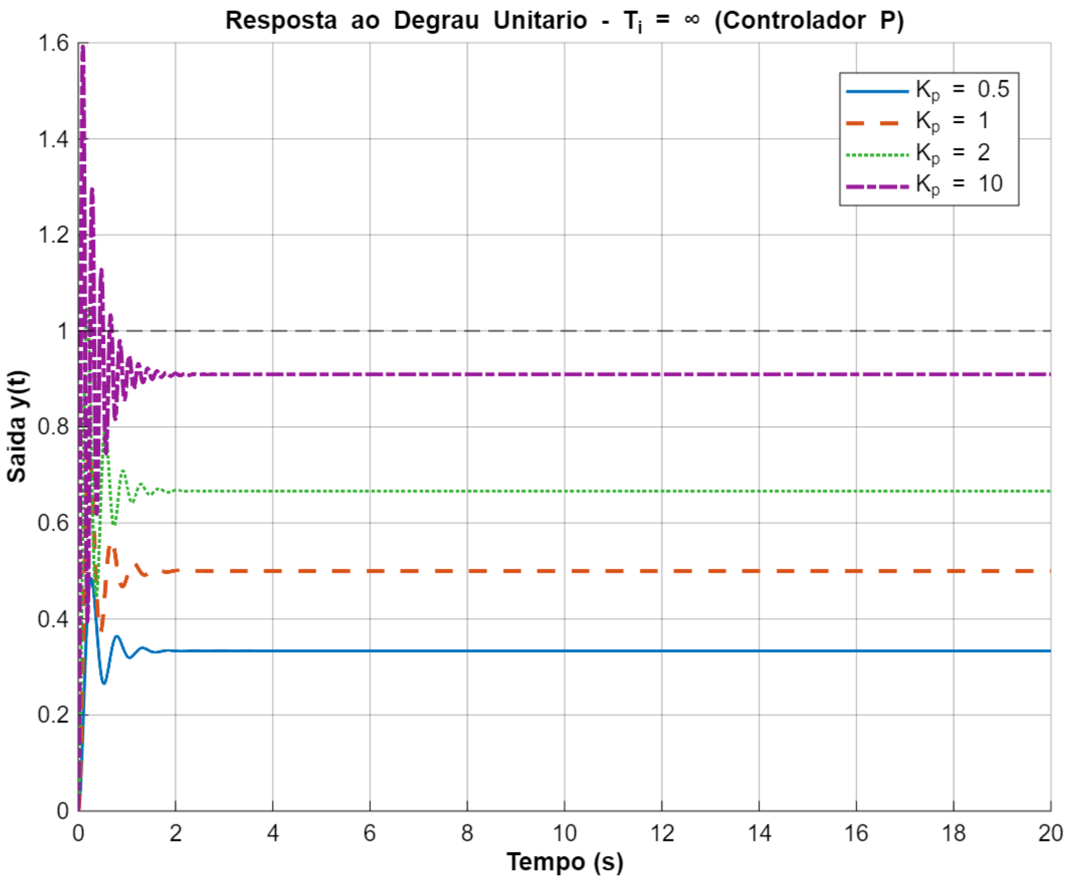

Figura 2: Resposta ao degrau unitário para $T_i = \infty$ (Controlador P)

A análise revela que, sem ação integral, o sistema apresenta erro de regime permanente para todos os valores de K_p testados. Para $K_p = 0{,}5$, observa-se resposta mais lenta com erro de aproximadamente 65%, enquanto $K_p = 1$ resulta em erro de cerca de 50%. O aumento do ganho proporcional para $K_p = 2$ reduz o erro para aproximadamente 35%, e $K_p = 10$ apresenta o menor erro (∼10%), porém com overshoot inicial significativo. Este comportamento confirma a limitação fundamental dos controladores proporcionais: a incapacidade de eliminar completamente o erro de regime permanente.

4.1.2 $T_i = 10$ s (Ação integral fraca)

A Figura 3 demonstra o comportamento do sistema com ação integral fraca ($T_i = 10$ s), onde o ganho integral é relativamente baixo, permitindo a observação da convergência gradual para o setpoint.

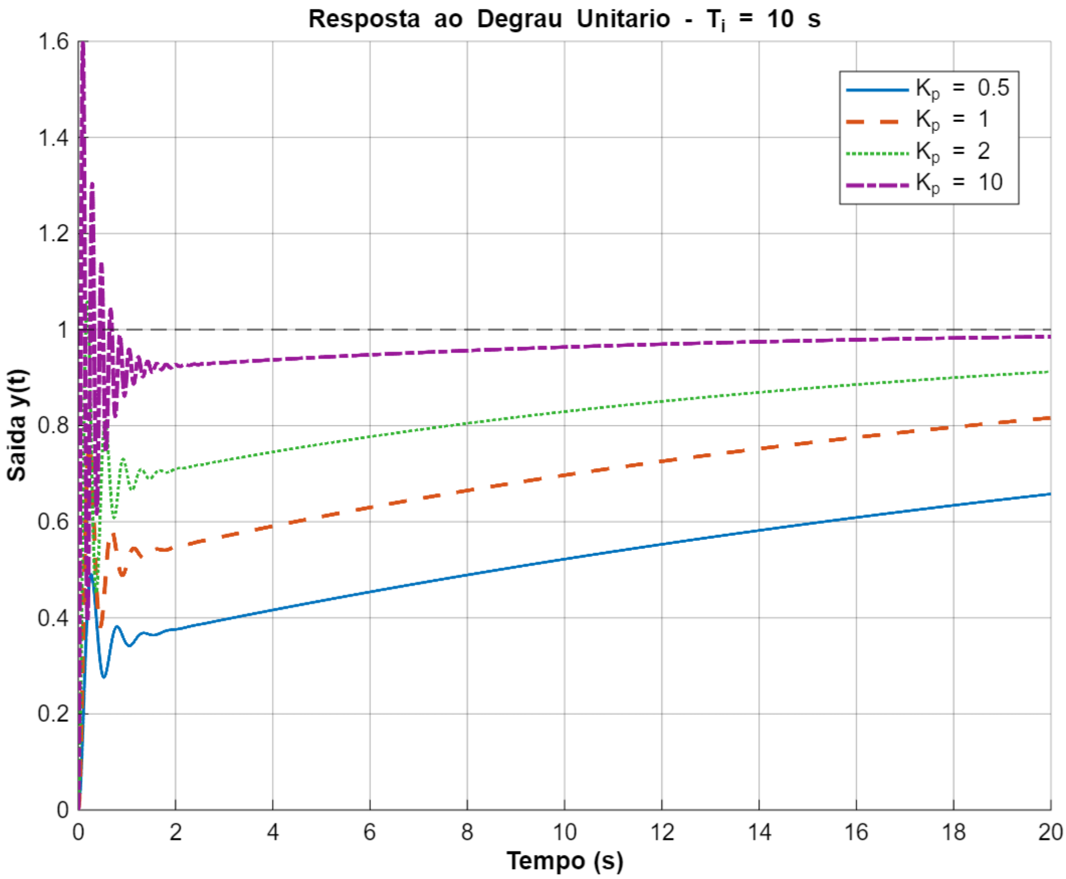

Figura 3: Resposta ao degrau unitário para $T_i = 10$ s

Os resultados mostram que a ação integral, mesmo que fraca, é capaz de eliminar o erro de regime permanente, com todas as curvas convergindo lentamente para o valor de referência $(y = 1)$. Ganhos proporcionais maiores resultam em resposta inicial mais rápida, mantendo a estabilidade do sistema. A convergência lenta é característica da ação integral fraca, que requer tempo considerável para acumular o erro e corrigi-lo completamente.

4.1.3 $T_i = 1$ s (Ação integral moderada)

A Figura 4 ilustra o comportamento com ação integral moderada ($T_i = 1$ s), representando um compromisso equilibrado entre velocidade de resposta e estabilidade do sistema.

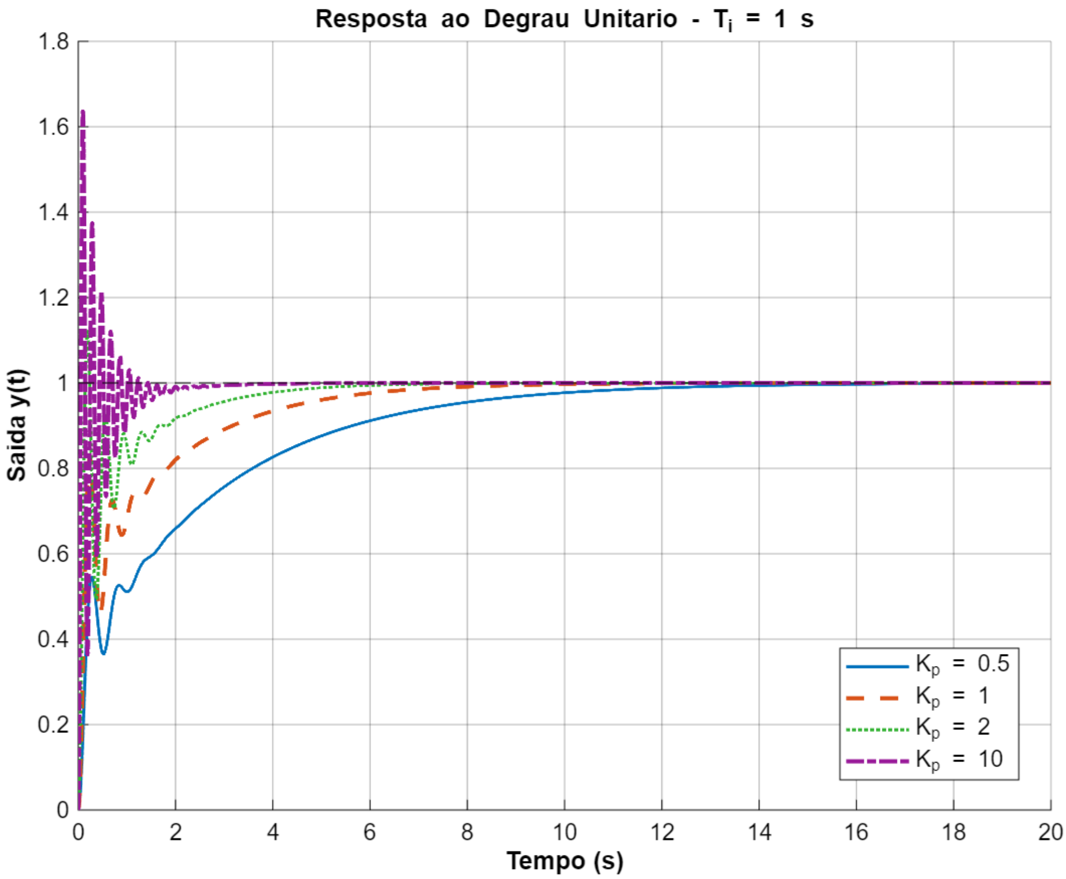

Figura 4: Resposta ao degrau unitário para $T_i = 1$ s

Esta configuração demonstra convergência mais rápida para o setpoint em comparação com $T_i = 10$ s. Para $K_p = 0{,}5$, observa-se resposta suave sem overshoot significativo. Os valores $K_p = 1$ e $K_p = 2$ apresentam pequeno overshoot com boa convergência, enquanto $K_p = 10$ resulta em overshoot mais pronunciado, mas ainda mantém estabilidade e convergência adequada. Este cenário representa um bom compromisso entre velocidade de resposta e estabilidade do sistema.

4.1.4 $T_i = 0{,}1$ s (Ação integral muito forte)

A Figura 5 revela o comportamento crítico do sistema com ação integral muito forte ($T_i = 0{,}1$ s), onde o ganho integral elevado pode comprometer a estabilidade, especialmente para ganhos proporcionais altos.

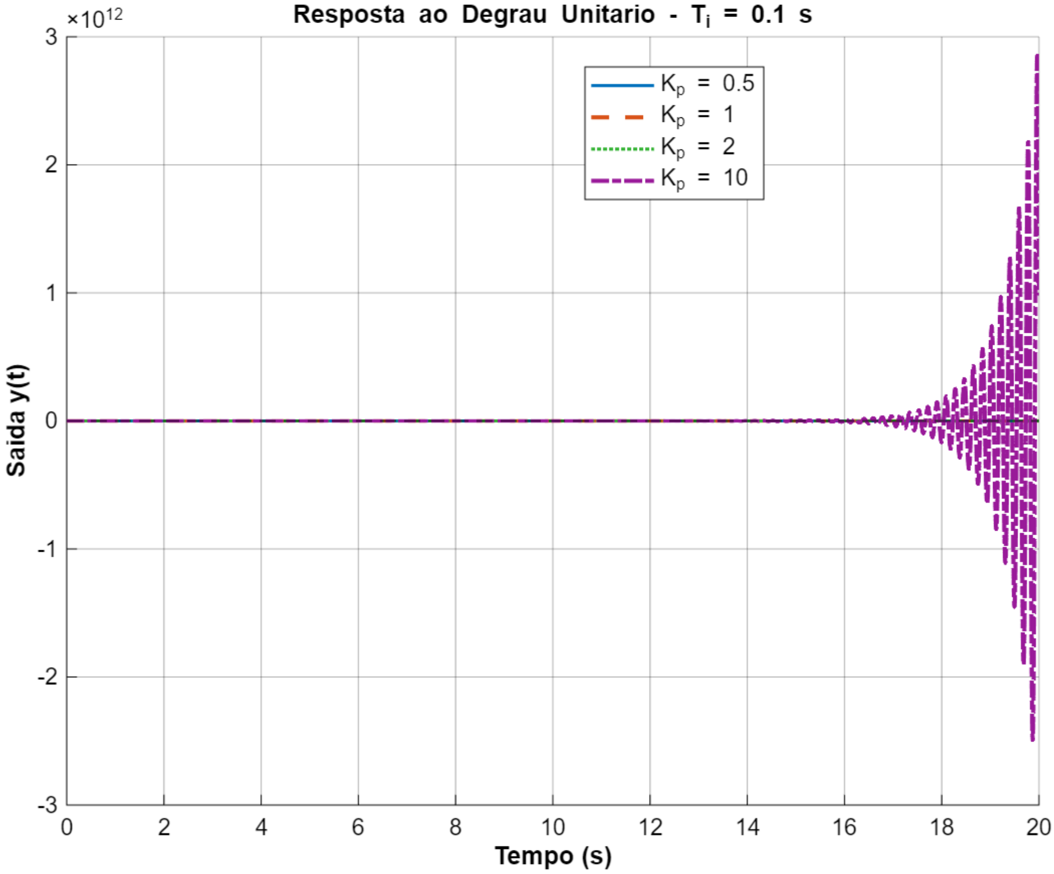

Figura 5: Resposta ao degrau unitário para $T_i = 0{,}1$ s

Os resultados evidenciam comportamento instável para $K_p = 10$, caracterizado por oscilações crescentes que levam o sistema à instabilidade. A escala do gráfico (10^{12}) indica valores extremos, confirmando a perda de controle do sistema. Este fenômeno demonstra que ação integral excessiva, combinada com ganho proporcional alto, pode desestabilizar completamente o sistema de controle, tornando esta combinação de parâmetros inadequada para aplicações práticas.

4.2 Gráficos com base em K_p

4.2.1 $K_p = 0{,}5$ (Ganho proporcional baixo)

A Figura 6 apresenta a resposta ao degrau unitário do sistema com ganho proporcional baixo ($K_p = 0{,}5$), permitindo a análise do efeito de diferentes tempos integrais em um sistema com resposta naturalmente conservadora.

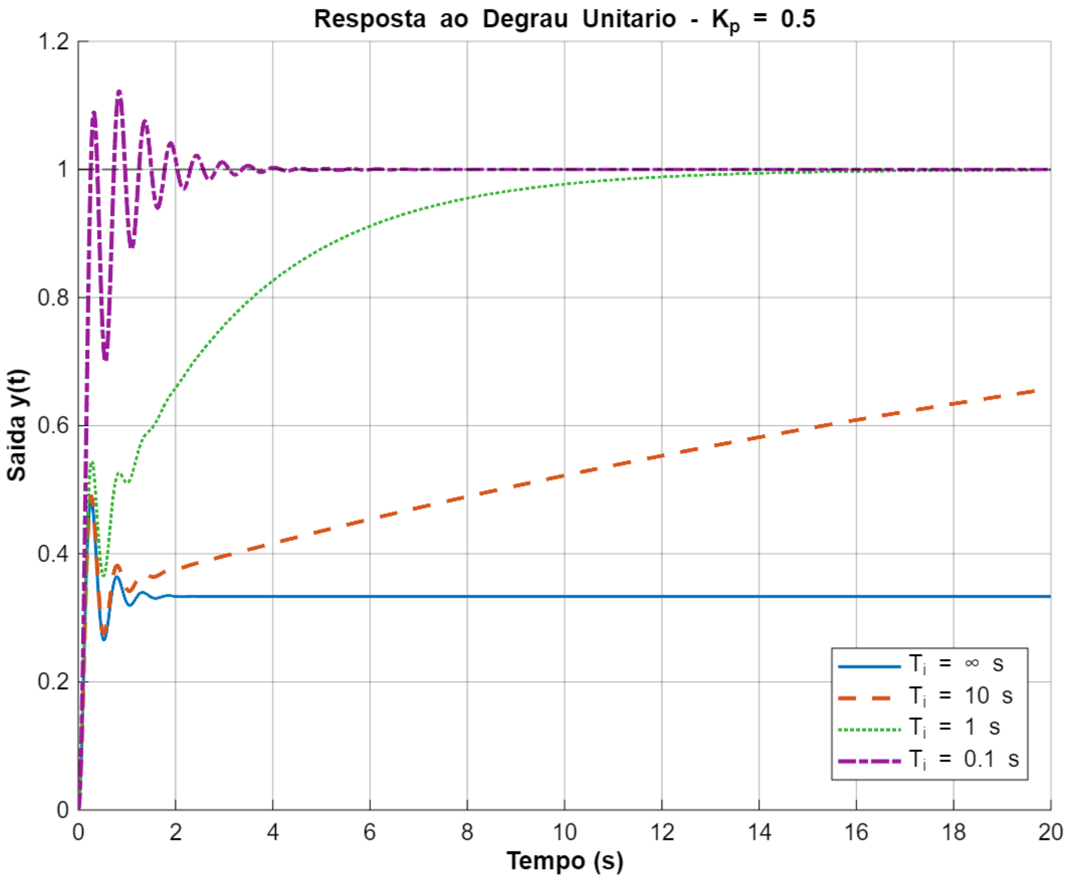

Figura 6: Resposta ao degrau unitário para $K_p = 0{,}5$

Com $K_p = 0{,}5$, o sistema apresenta comportamento estável para todos os valores de T_i testados. Para $T_i = \infty$, observa-se erro de regime permanente de aproximadamente 65% com resposta suave. O tempo integral $T_i = 10$ s proporciona convergência lenta mas estável para o setpoint. Com $T_i = 1$ s, obtém-se boa velocidade de convergência sem overshoot significativo, enquanto $T_i = 0{,}1$ s resulta em oscilações amortecidas que convergem rapidamente para o valor desejado. Este ganho proporcional baixo demonstra excelente margem de estabilidade mesmo com ação integral forte.

4.2.2 $K_p = 1$ (Ganho proporcional moderado)

A Figura 7 ilustra o comportamento do sistema com ganho proporcional moderado ($K_p = 1$), representando um compromisso entre velocidade de resposta e estabilidade do sistema.

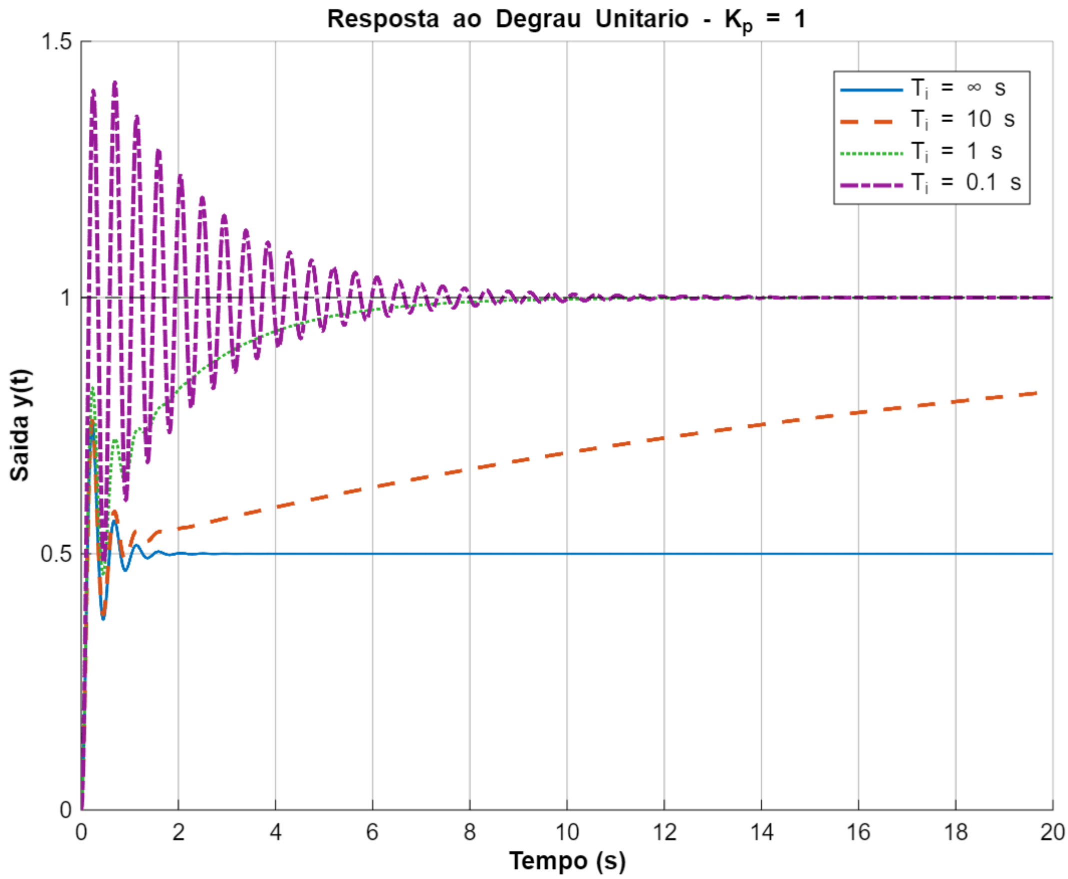

Figura 7: Resposta ao degrau unitário para $K_p = 1$

O sistema com $K_p = 1$ mantém estabilidade para a maioria das configurações. Para $T_i = \infty$, o erro de regime permanente reduz para aproximadamente 50%. Com $T_i = 10$ s e $T_i = 1$ s, o sistema apresenta convergência adequada com pequeno overshoot. Entretanto, para $T_i = 0{,}1$ s, começam a aparecer oscilações mais pronunciadas, indicando que a combinação de ganho proporcional moderado com ação integral muito forte aproxima o sistema dos limites de estabilidade.

4.2.3 $K_p = 2$ (Ganho proporcional elevado)

A Figura 8 demonstra o comportamento crítico do sistema com ganho proporcional elevado ($K_p = 2$), onde a combinação com ação integral forte pode levar à instabilidade.

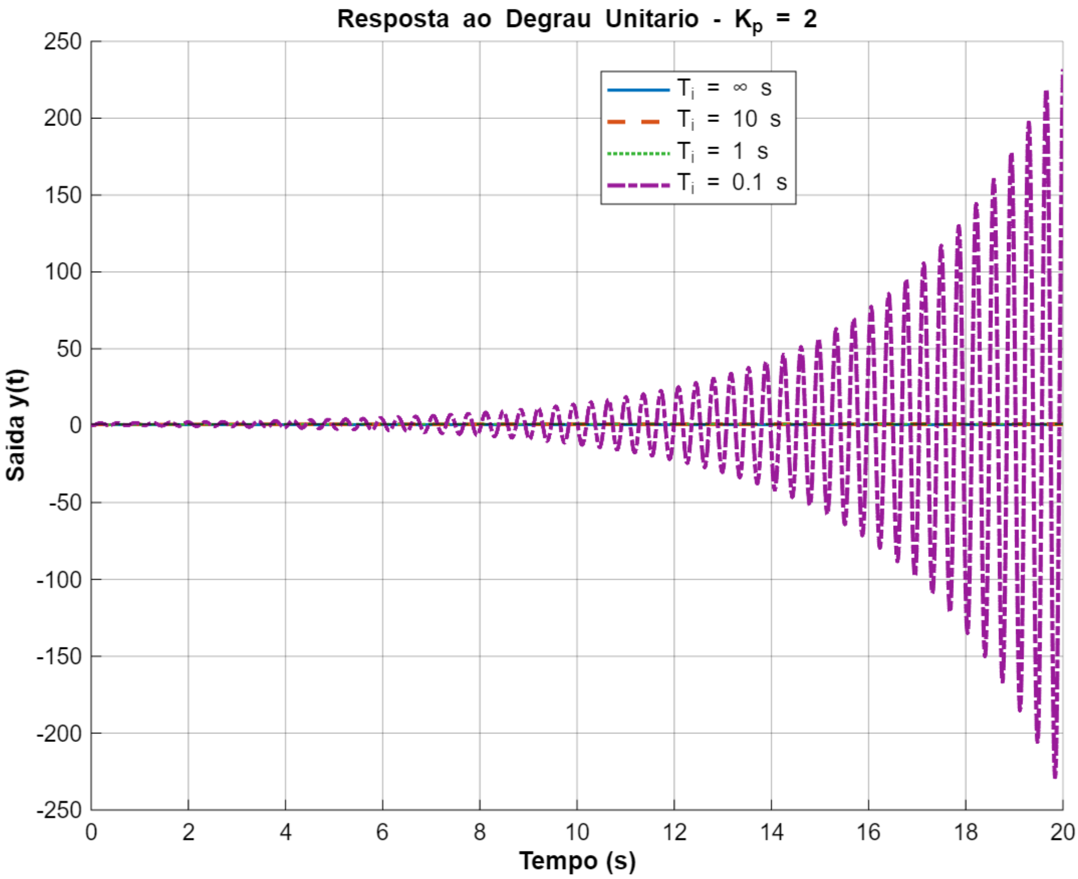

Figura 8: Resposta ao degrau unitário para $K_p = 2$

Com $K_p = 2$, o sistema ainda mantém estabilidade para $T_i = \infty$, $T_i = 10$ s e $T_i = 1$ s, apresentando erro de regime de aproximadamente 35% no caso proporcional puro. No entanto, para $T_i = 0{,}1$ s, o sistema torna-se claramente instável, com oscilações crescentes que atingem amplitudes de até 250 unidades. Este comportamento evidencia que ganhos proporcionais elevados requerem cuidado especial na escolha do tempo integral para manter a estabilidade do sistema.

4.2.4 $K_p = 10$ (Ganho proporcional muito elevado)

A Figura 9 revela o comportamento extremo do sistema com ganho proporcional muito elevado ($K_p = 10$), demonstrando instabilidade severa quando combinado com ação integral forte.

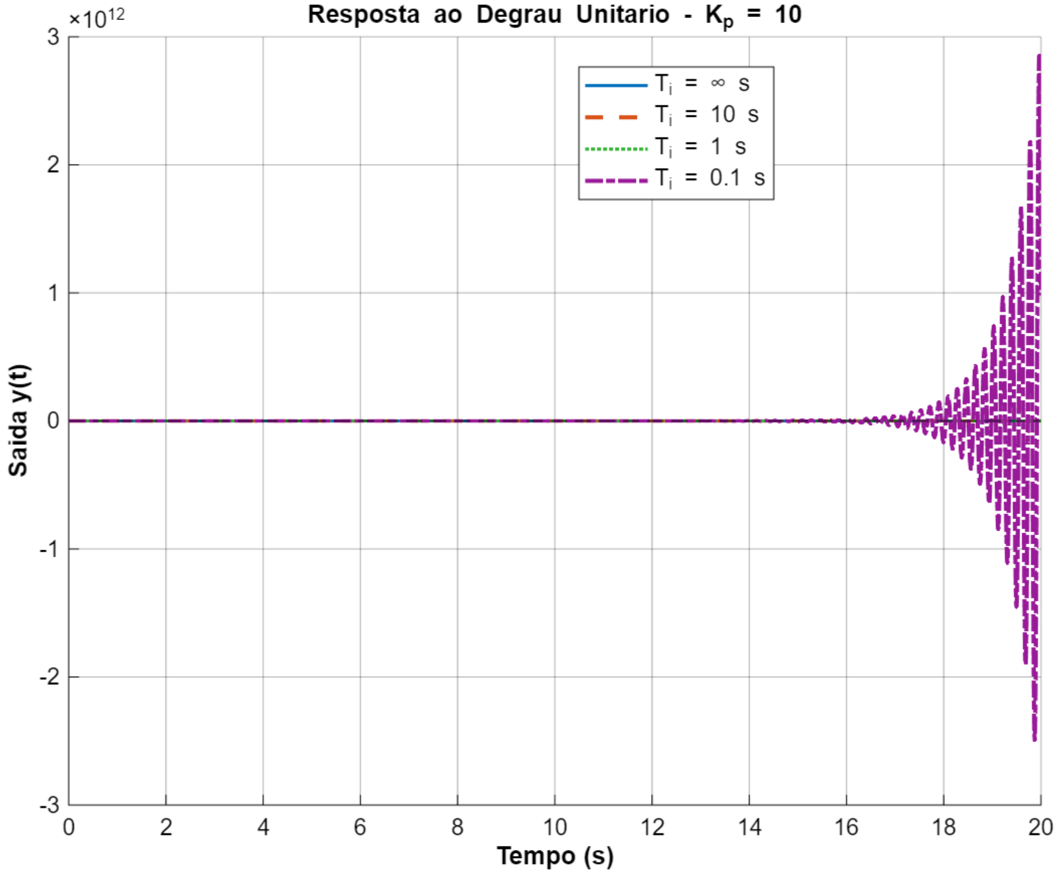

Figura 9: Resposta ao degrau unitário para $K_p = 10$

O sistema com $K_p = 10$ apresenta comportamento drasticamente diferente dependendo do tempo integral. Para $T_i = \infty$, mantém-se estável com erro de regime reduzido para aproximadamente 10%. Com $T_i = 10$ s e $T_i = 1$ s, o sistema ainda converge adequadamente, embora com overshoot mais pronunciado. Contudo, para $T_i = 0{,}1$ s, ocorre instabilidade catastrófica com oscilações que atingem a escala de 10^{12}, tornando o sistema completamente incontrolável. Este resultado confirma a importância crítica do dimensionamento adequado dos parâmetros do controlador PI.

4.2.5 Análise comparativa dos gráficos por K_p

A análise dos gráficos agrupados por K_p revela padrões consistentes no comportamento do sistema:

- **Estabilidade decrescente:** À medida que K_p aumenta, a margem de estabilidade diminui, especialmente para T_i baixos;
- **Erro de regime:** O controlador puramente proporcional ($T_i = \infty$) apresenta erro inversamente proporcional ao ganho K_p;
- **Limite crítico:** A combinação $K_p = 10$ e $T_i = 0{,}1$ s representa um limite crítico de instabilidade;
- **Compromisso ótimo:** Os valores $K_p = 1$ ou $K_p = 2$ com $T_i = 1$ s oferecem o melhor compromisso entre performance e estabilidade.