Sistemas Realimentados - Turma 2

EP10 - Projeto do Controlador PI

Nome: Gabriel Gatti da Silva e Thiago Felippe Neitzke Lahass

Seja o sistema de controle mostrado na Fig. 1, onde $J = 0.5$ **e,** $b = 2$**.**

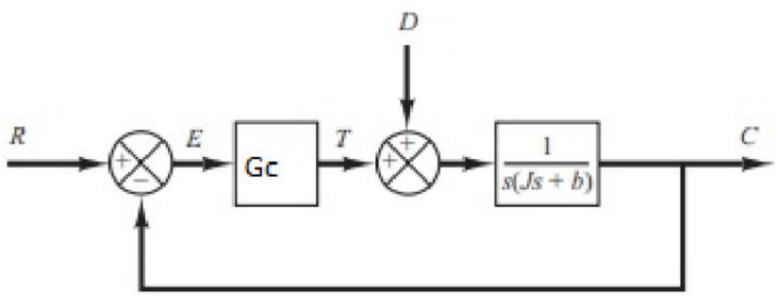


Figura 1

1) Projete um controlador Proporcional via método do Lugar das Raízes para que a resposta à entrada degrau possua sobressinal menor ou igual a 10%, com D=0, e possua o menor erro em regime ao degrau de distúrbio unitário, com R=0.

Para $J = 0.5$ e $b = 2$, temos que a FT da planta é:

$G(s) = \frac{1}{s(0.5s + 2)} = \frac{1}{0.5s^2 + 2s} = \frac{2}{s^2 + 4s}$, cujos polos são $P_1 = 0$ e $P_2 = -\frac{b}{J} = -4$, e não possui zeros. Importante ressaltar que $G(s)$ é do tipo 1, logo o erro em regime para entrada degrau sempre será 0.

Fechando a malha temos que as FTs ficam:

- $\frac{C(s)}{R(s)} = \frac{G_c(s)G(s)}{1 + G_c(s)G(s)}$ (1);
- $\frac{C(s)}{D(s)} = \frac{G(s)}{1 + G_c(s)G(s)}$ (2).

Como nessa primeira questão temos um controlador proporcional, então temos $G_c(s) = K$, e as FTs ficam:

$$\frac{C(s)}{R(s)} = \frac{KG(s)}{1 + KG(s)}$$ (3);

$$\frac{C(s)}{D(s)} = \frac{G(s)}{1 + KG(s)}$$ (4).

Onde suas equações características são iguais: $1 + KG(s) = 0$ (5).

Para ter um sobressinal menor ou igual a 10% à entrada degrau, é necessário que:

$$e^{\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}} \leq 0.1 \Rightarrow \zeta \geq 0.591$$.

Fazendo a análise do Lugar da Raízes de $G(s)$, temos:

- $\phi = \pm\frac{180}{2}(2r + 1) = \begin{cases} r = 0 \Rightarrow \phi = 90° \\ r = 1 \Rightarrow \phi = 270° \end{cases}$.
- $\sigma_a = \frac{(0 - 4) - (0)}{2} = -2$.
- Pontos de Sela: $N'(s)D(s) - N(s)D'(s) = 0 \Rightarrow 2(2s + 4) = 0 \Rightarrow s^* = -2.$

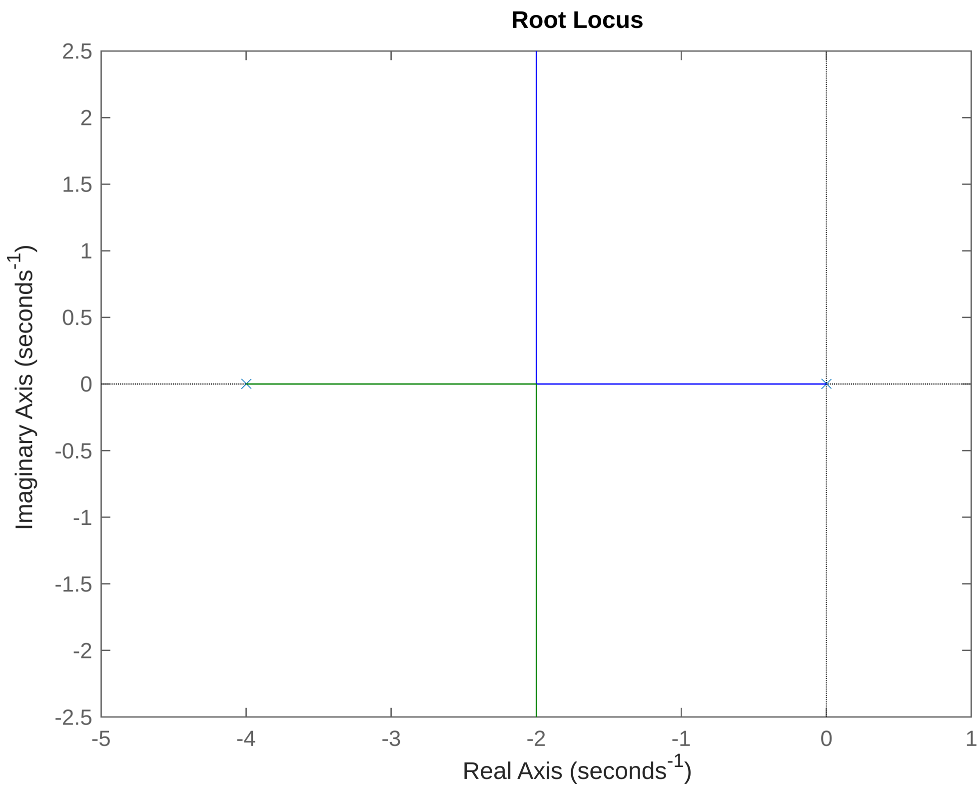
Com essas informações podemos desenhar o LR de $G(s)$ à mão, mas podemos também utilizar o *rlocus* ou o *rltool* para plotar isso de forma direta.

Utilizando o *rlocus* para visualizar os dados obtidos acima, temos:

```
J = 0.5;
b = 2;
s = tf('s');
G = 1 / (s* (J*s + b))
```

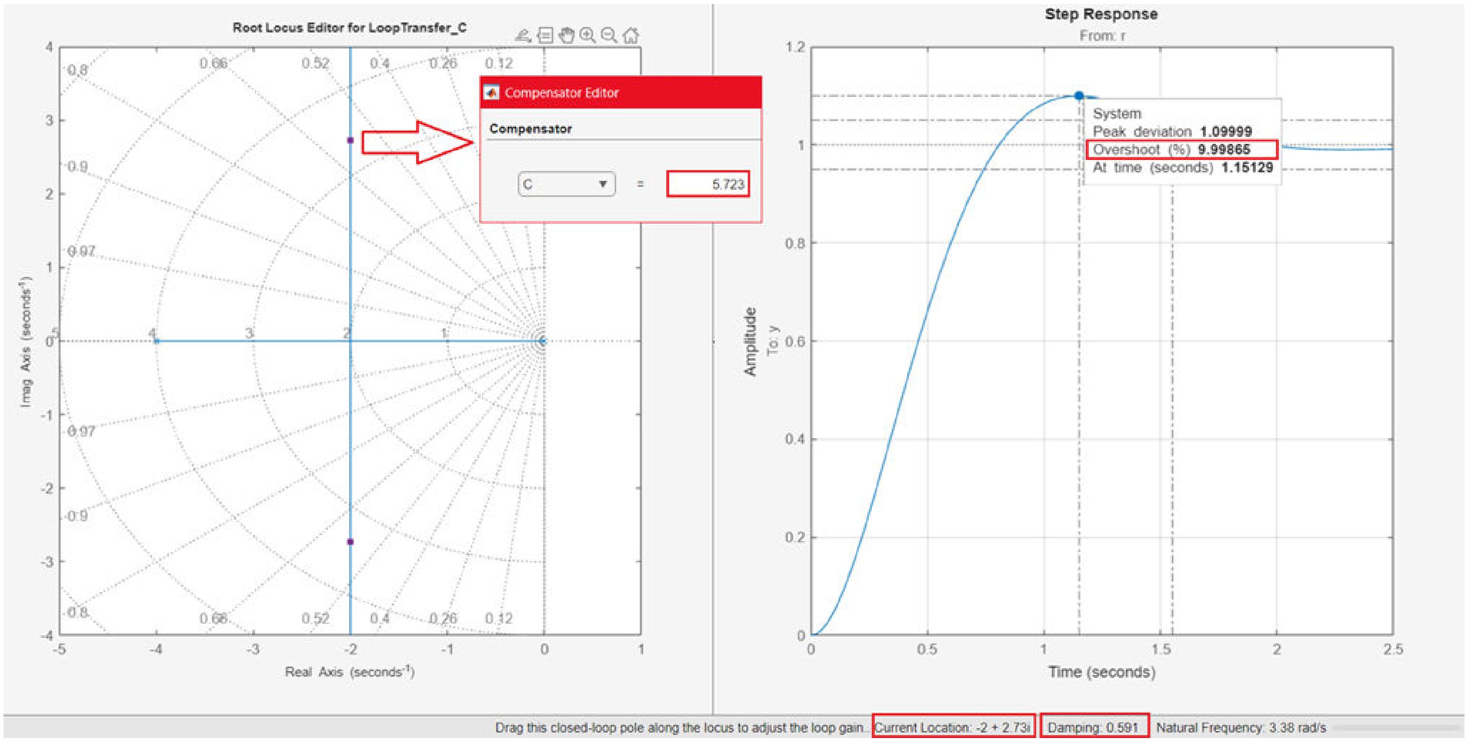
```
G =

        1
  -------------
  0.5 s^2 + 2 s

Continuous-time transfer function.
Model Properties
```

```
rlocus(G)
hold off
```



Podemos agora utilizar também o *rltool* para analisar os valores de K que satifazem a exigência de sobressinal vista ($\zeta \geq 0.591$):

```
% rltool(G)
```



Assim, podemos ver que para $\zeta = 0.591$ temos $K = 5.723$, que é o valor máximo que K pode assumir para que tenhamos um sobressinal menor ou igual a 10%.

Logo $K \leq 5.723$.

Para atendermos à segunda exigência (possuir o menor erro em regime ao degrau de distúrbio unitário, com $R = 0$), devemos analisar a resposta usando o TVF.

Por (4), temos:

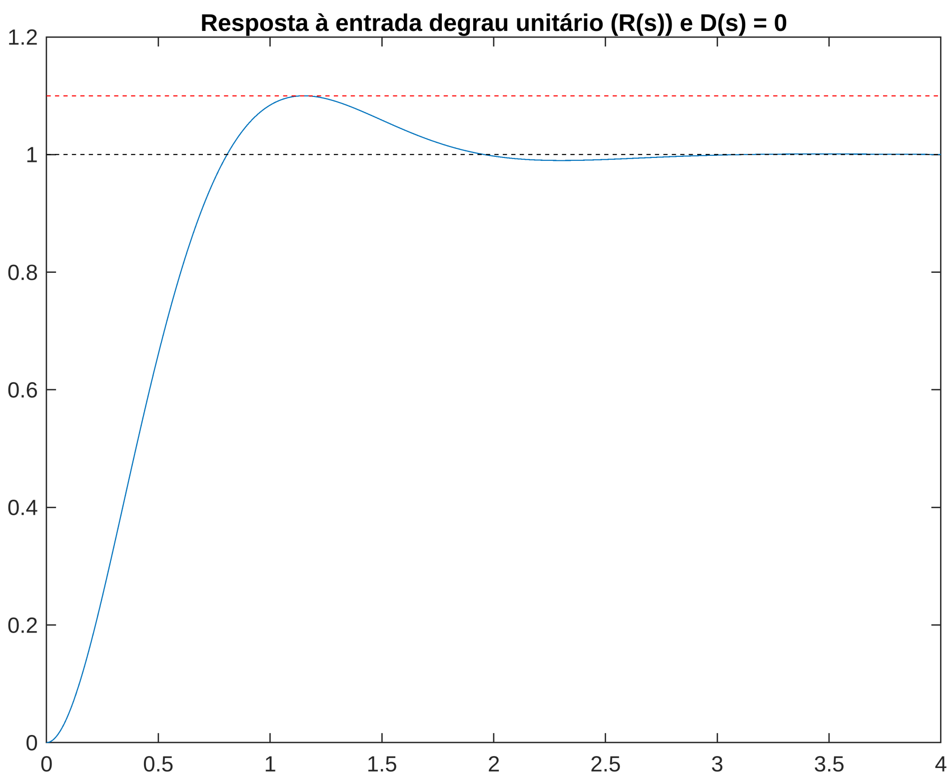
$\frac{C(s)}{D(s)} = \frac{G(s)}{1 + KG(s)}$, logo para $D(s) = \frac{1}{s}$, temos que valor da saída $C(s)$, em regime, é:

$$C_{ss} = \lim_{s \to 0} s \frac{D(s)G(s)}{1 + KG(s)} = \lim_{s \to 0} s \frac{1}{s} \frac{\frac{2}{s^2 + 4s}}{1 + K \frac{2}{s^2 + 4s}} = \lim_{s \to 0} \frac{\frac{2}{s^2 + 4s}}{\frac{s^2 + 4s + 2K}{s^2 + 4s}} = \frac{2}{2K} = \frac{1}{K}.$$

Como o erro em regime ao distúrbio unitário é dado por $C_{ss} - 0 = C_{ss} = \frac{1}{K}$, para chegarmos ao menor valor para ele, devemos ter o maior K possível, que, como vimos anteriormente, é $K = 5.723$.

Plotando as respostas ao degrau na entrada e no distúrbio, temos:

```
K = 5.723;
Gmf = feedback(K*G, 1);

t = 0:0.01:6;           % vetor de tempo
U_t = ones(size(t));    % Degrau unitário
y1_t = lsim(Gmf, U_t, t);

% Plot da saida
plot(t, y1_t);
hold on;
% Plot da referencia (degrau unitario)
plot(t, U_t, 'k--');
line([0, t(end)], [1.1, 1.1], 'Color', 'r', 'LineStyle', '--',
'DisplayName', '+ 10%');
% Limitando o plot para até 4 segundos no eixo x
xlim([0, 4]);
title('Resposta à entrada degrau unitário (R(s)) e D(s) = 0');
hold off;
```



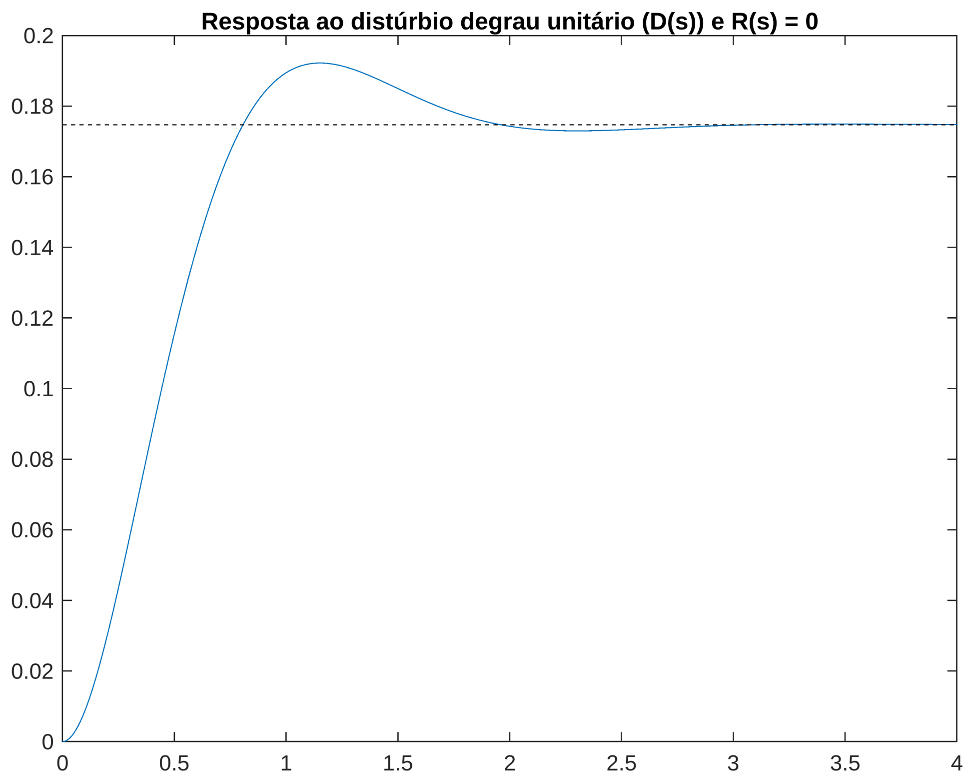
Vamos encontrar o tempo de acomodação para o intervalo de $\pm 5\%$ que será usado no item 2:

```
t_acomodacao = stepinfo(Gmf,SettlingTimeThreshold=0.05).SettlingTime
```

```
t_acomodacao = 1.5510
```

```
Gdf = feedback(G, K);
```

```
yd1_t = lsim(Gdf, U_t, t);

% Plot da saida
plot(t, yd1_t);
hold on;
% Plot da referencia (degrau unitario)
plot(t, U_t*(1/K), 'k--');
% Limitando o plot para até 4 segundos no eixo x
xlim([0, 4]);
title('Resposta ao distúrbio degrau unitário (D(s)) e R(s) = 0');
hold off;
```



2) Projete um controlador PI via método do Lugar das Raízes para que o sistema em malha fechada tenha sobressinal menor ou igual a 10% e possua tempo de acomodação menor ou igual ao do sistema de controle projetado no item 1.

Para um controlador PI, a configuração de $G_c(s)$ passa a ser:

$G_c(s) = K_p + \frac{K_i}{s} = \frac{K_p\left(s + \frac{K_i}{K_p}\right)}{s}$, assim notamos que o novo controlador introduz um polo em 0, e um zero em $z = -\frac{K_i}{K_p}$.

Dessa forma, a nova equação característica do sistema em malha fechada utilizando o controlador PI passa a ser:

$$1 + K_p\left(\frac{s + \frac{K_i}{K_p}}{s}\right)G(s) = 0 \Rightarrow 1 + K_p\left(\frac{s + \frac{K_i}{K_p}}{s}\right)\frac{2}{s^2 + 4s} = 0.$$

É importante ressaltar que como o controlador PI introduz um polo na origem, o erro em regime à entrada degrau sempre será nulo (o que já era garantido por $G(s)$ ser do tipo 1).

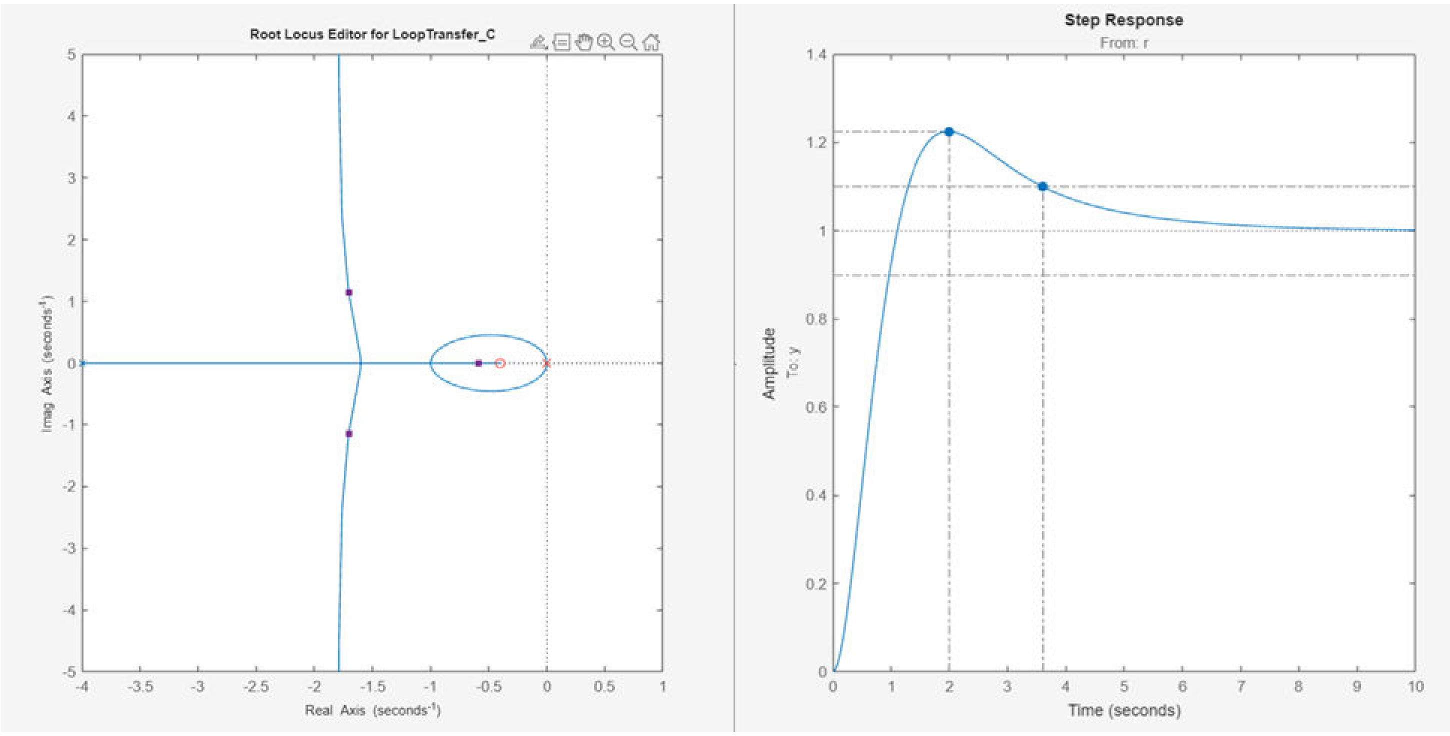
O primeiro passo do projeto é obter $z = -\frac{K_i}{K_p}$ próximo à origem e maior que a parte real do polo mais dominante da planta ($p_2 = -4$) para garantirmos estabilidade, pórem nada impede que a parte real do zero seja menor que a do polo, caso tenhamos estabilidade. Por outro lado, um zero muito próximo à origem tende a deixar a resposta lenta, assim podemos querer afastá-lo para deixar a resposta mais rápida.

É possível analisar o LR e as respostas no tempo variando o valor do zero inserido, analisando os valores de K_p que atendam ao sobressinal, assim como os tempos de estabelecimento, para, então, ver se estão de acordo com a especificação de menor ou igual ao do sistema projetado no item 1, ou seja, $t_{\text{estabelecimento}} \leq 1.55\ s$.

Um bom valor inicial para testes é fazer o valor de zero sendo 10% da parte real do polo mais próximo à origem $(p_2 = -4)$, isto é, $z = -0.4$, e a partir disso aumentar o valor em módulo do zero inserido e verificar os resultados obtidos. Usando o *rltool*, temos:

- Para $z = -0.4$:

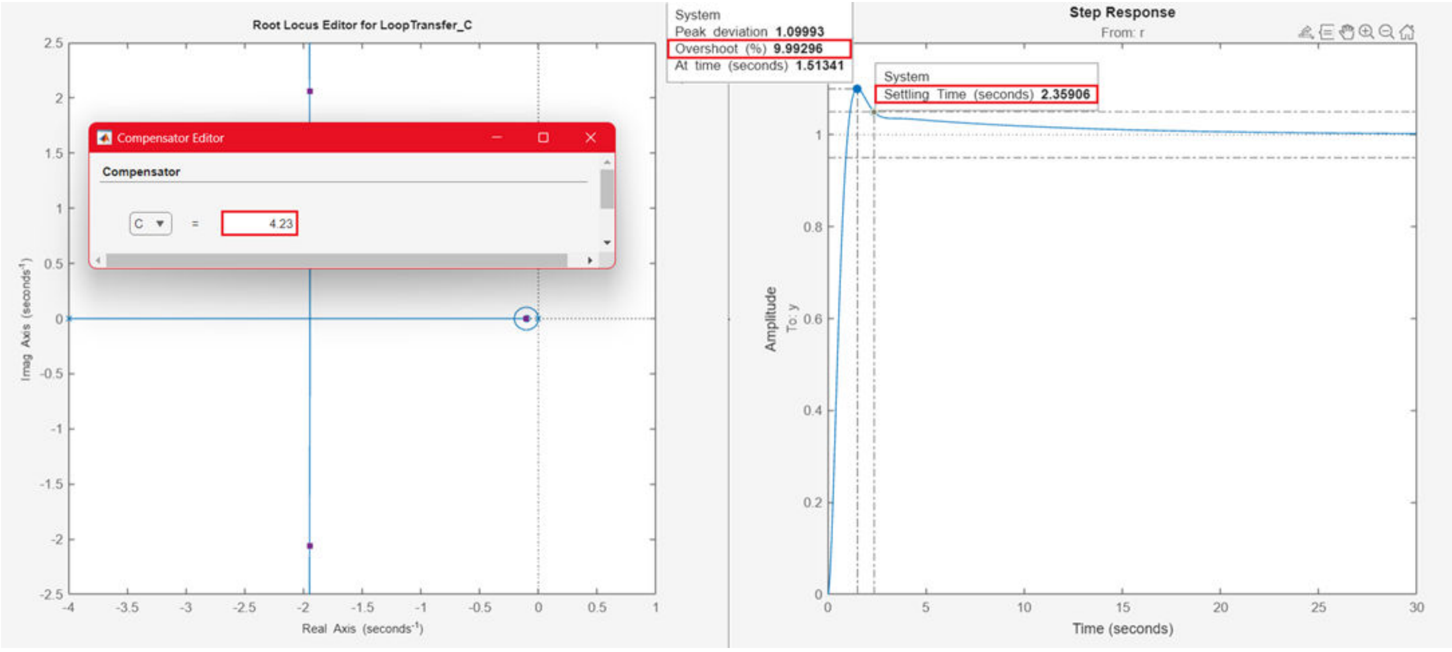
```
s = tf('s');
z = -0.4;
gma_PI = ( (s - z) / s ) * G;
%rltool(gma_PI)
```



Analisando a nova FT de malha aberta pelo *rltool*, para esse valor de z, não foi possível encontrar algum valor para K_p tal que o sobressinal fosse menor do que 10%. Nesse caso, devemos aproximar o zero da origem, para aumentar seu efeito. Nesse caso, podemos escolher, por exemplo, $z = -0.1$:

- Para $z = -0.1$:

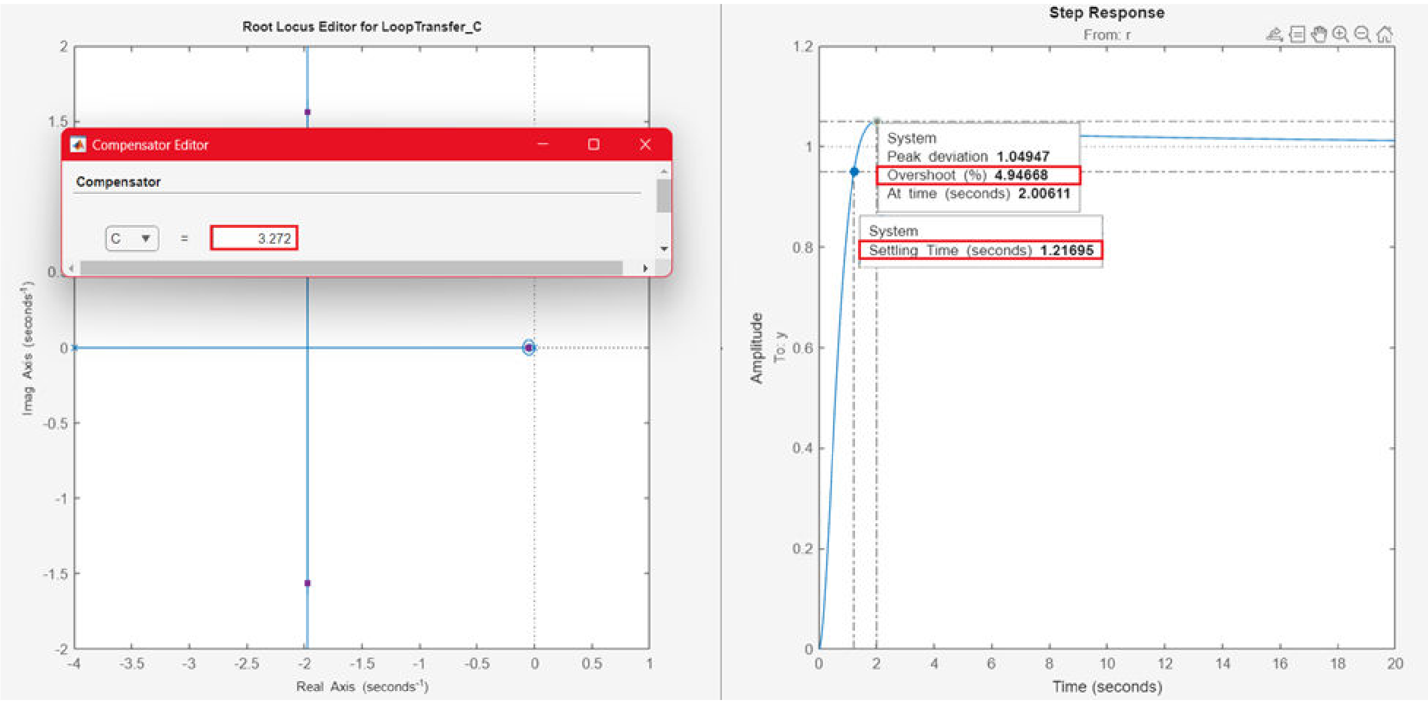
```
s = tf('s');
z = -0.1;
gma_PI = ( (s - z) / s ) * G;
%rltool(gma_PI)
```



Dessa vez foi possível encontrar valores para K_p tal que o sobressinal fosse menor do que 10%. Para esse caso, o menor tempo de estabelecimento possível respeitando o sobressinal foi $t_s = 2.36$ s, obtido para $K_p = 4.23$, que é maior que o tempo de estabelecimento encontrado no item 1. Podemos diminuir ainda mais o valor de z e analisar o que acontece. Por exemplo, podemos verificar o que acontece para $z = -0.05$:

- Para $z = -0.05$:

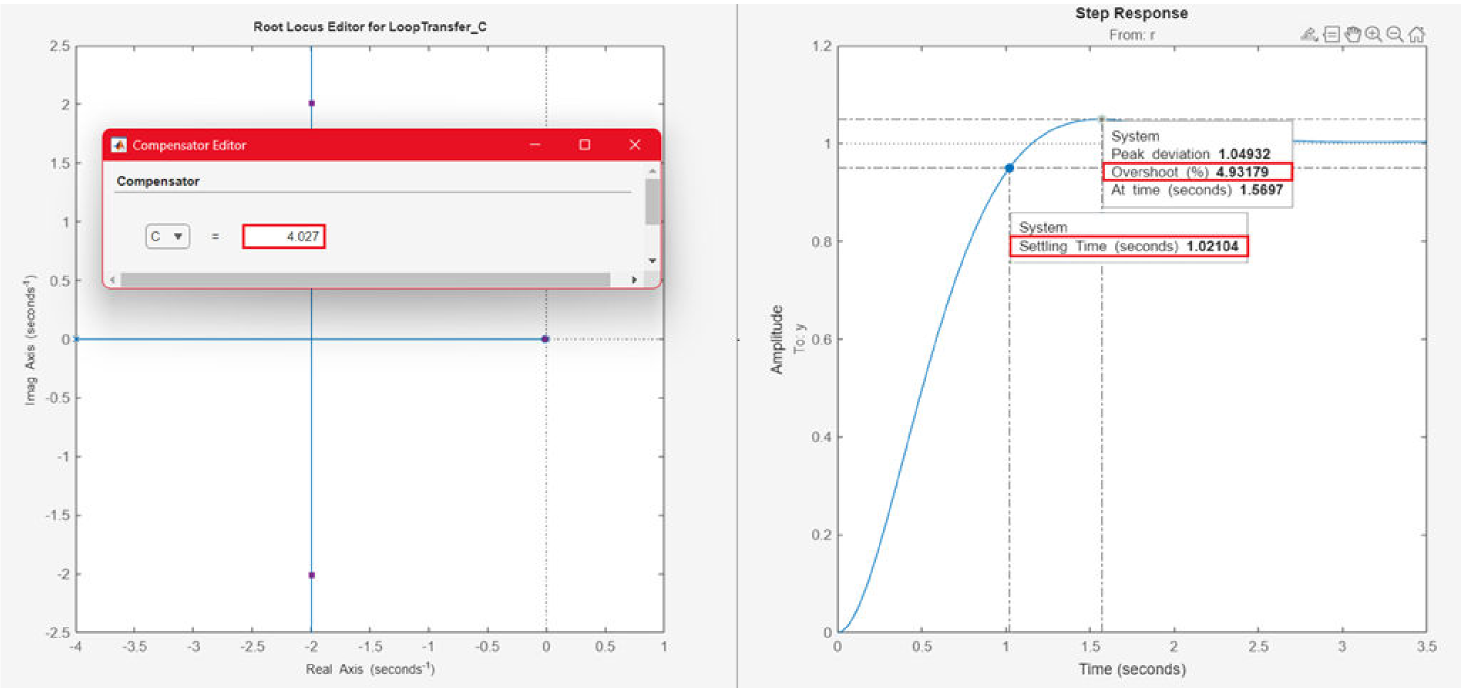
```
s = tf('s');
z = -0.05;
gma_PI = ( (s - z) / s ) * G;
%rltool(gma_PI)
```



Dessa vez também foi possível encontrar valores para K_p tal que o sobressinal fosse menor do que 10%. Para esse caso, o menor tempo de estabelecimento possível foi $t_s = 1.22$ s, obtido para $K_p = 3.272$, que já atende a especificação, visto que 1.22 s < 1.55 s. Podemos diminuir ainda mais o valor de z e analisar o que acontece. Por exemplo, podemos verificar o que acontece para $z = -0.01$:

- Para $z = -0.01$:

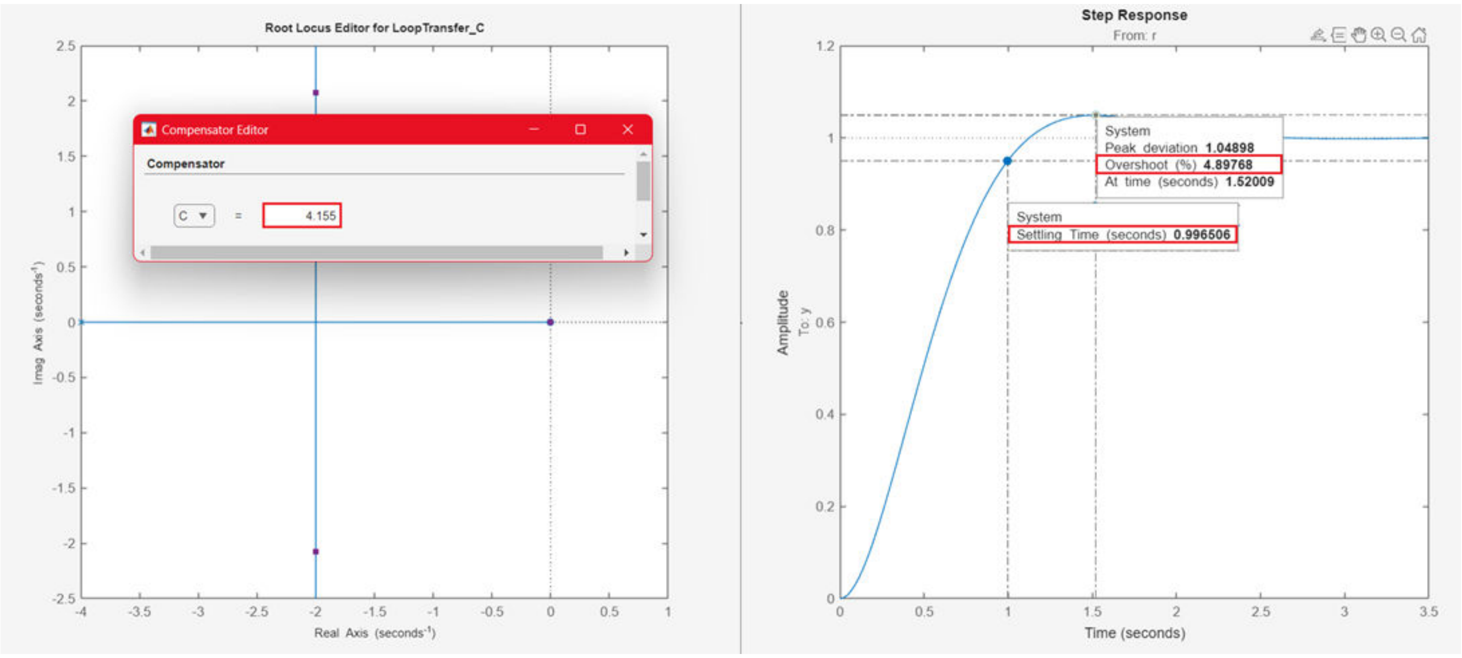
```
s = tf('s');
z = -0.01;
gma_PI = ( (s - z) / s ) * G;
%rltool(gma_PI)
```



Dessa vez também foi possível encontrar valores para K_p tal que o sobressinal fosse menor do que 10%. Para esse caso, o menor tempo de estabelecimento possível foi $t_s = 1.02$ s, obtido para $K_p = 4.027$. Podemos diminuir ainda mais o valor de z e analisar o que acontece. Por exemplo, podemos verificar o que acontece para $z = -0.001$:

- Para $z = -0.001$:

```
s = tf('s');
z = -0.001;
gma_PI = ( (s - z) / s ) * G;
%rltool(gma_PI)
```



Nesse caso, também foi possível encontrar valores para K_p tal que o sobressinal fosse menor do que 10%. Para esse caso, o menor tempo de estabelecimento possível foi aproximadamente $t_s = 1.00$ s, obtido para $K_p = 4.155$.

Para valores ainda menores de z não foram obtidas mudanças significativas no tempo de estabelecimento. Dessa forma, o menor tempo de estabelecimento obedecendo o sobressinal foi $t_s = 1.00$ s, obtido para $z = -0.001$ e $K_p = 4.155$. Como quanto mais próximo da origem o zero estiver, mais lenta fica a resposta, vamos utilizar o zero mais à esquerda que atendeu à especificação de tempo de estabelecimento, que foi $z = -0.05$. Portanto, podemos obter o valor de K_i a partir da relação:

$$z = -\frac{K_i}{K_p} \Rightarrow K_i = -z * K_p \Rightarrow K_i = 0.1636 \ .$$

Tendo os parâmetros do controlador PI, podemos fechar a malha e plotar sua a resposta ao degrau:

```
z = -0.05;
Kp = 3.272;
Ki = Kp * (-z)
```

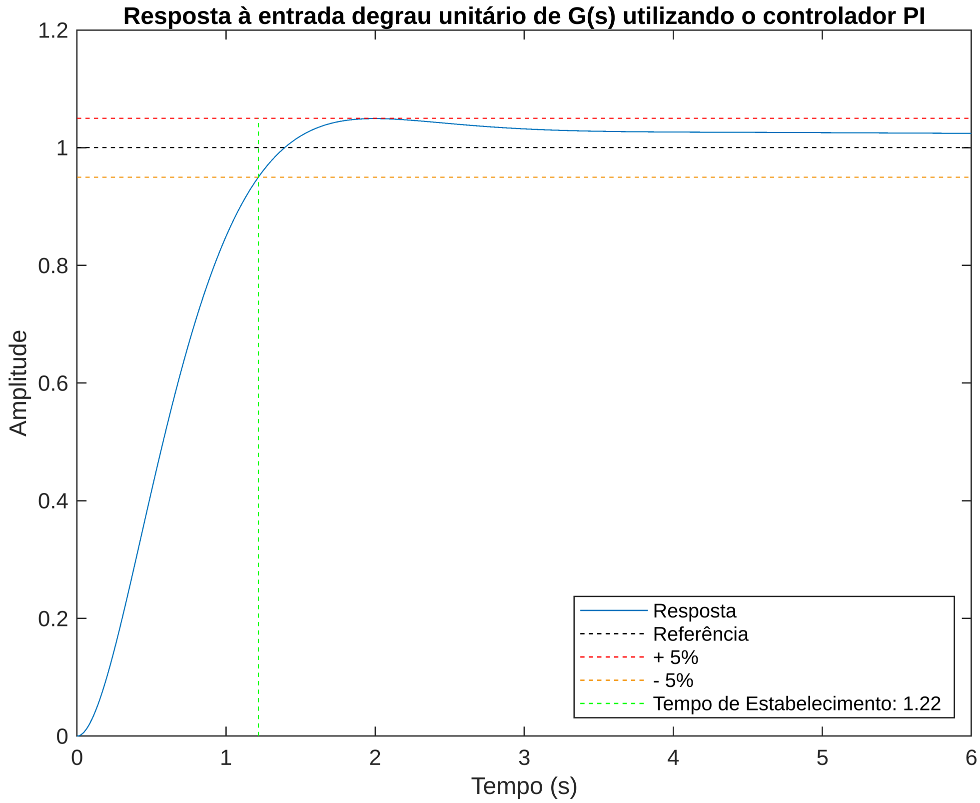
```
Ki = 0.1636
```

```
Gc = Kp + Ki / s
```

```
Gc =

  3.272 s + 0.1636
  ----------------
         s
```

```
Continuous-time transfer function.
Model Properties
```

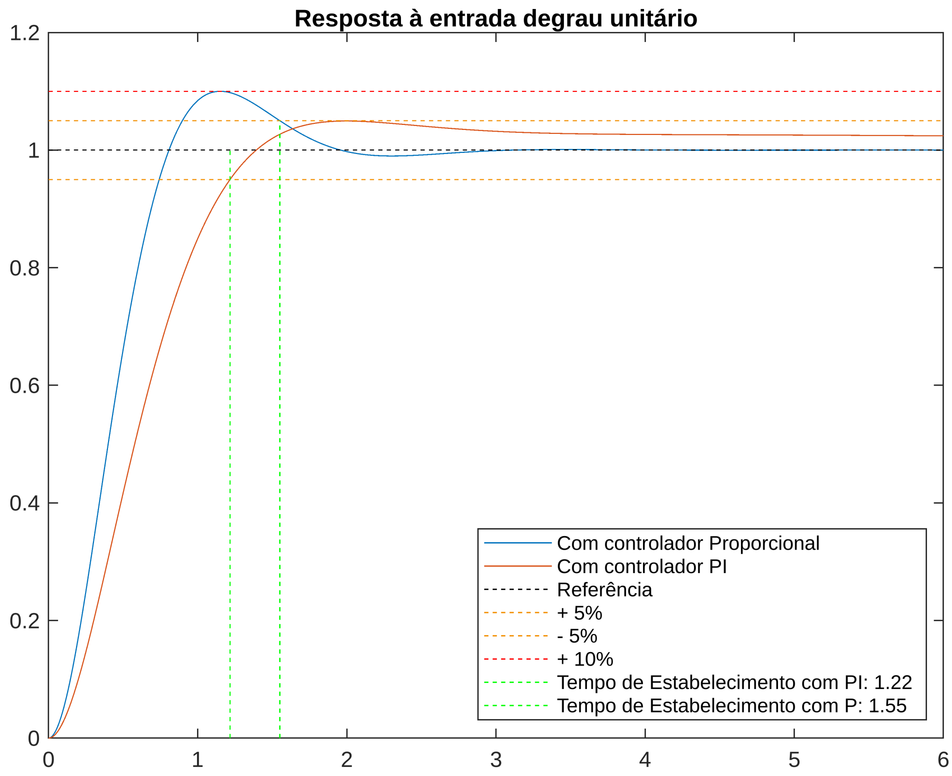
```
Gmf_PI = feedback(Gc*G, 1);

t = 0:0.01:6;            % vetor de tempo
U_t = ones(size(t));     % Degrau unitário
y2_t = lsim(Gmf_PI, U_t, t);

% Plot da saída e da referência (degrau unitário)
plot(t, y2_t, 'DisplayName', 'Resposta');
hold on;
plot(t, U_t, 'k--', 'DisplayName', 'Referência');
title('Resposta à entrada degrau unitário de G(s) utilizando o controlador
PI');
xlabel('Tempo (s)');
ylabel('Amplitude');

% Encontrando o valor final ou de regime da resposta
final_value = 1;
% Calculando 5% do valor final
tolerance = 0.05 * final_value;

% Encontrando o tempo de estabelecimento à 5%
settling_time_PI = stepinfo(Gmf_PI,SettlingTimeThreshold=0.05).SettlingTime;

% Adicionando linha com informações
line([0, t(end)], [1.05, 1.05], 'Color', 'r', 'LineStyle', '--',
'DisplayName', '+ 5%');
line([0, t(end)], [0.95, 0.95], 'Color', '#F38F00', 'LineStyle', '--',
'DisplayName', '- 5%');
line([settling_time_PI, settling_time_PI], [0, 1.05], 'Color', 'g',
'LineStyle', '--', 'DisplayName', ['Tempo de Estabelecimento: '
num2str(settling_time_PI, '%.2f')]);
legend('Location', 'southeast');
hold off;
```



3) Compare as resposta transitórias e em regime à entrada degrau, com D=0, e as respostas em regime à entrada distúrbio, com R=0, de ambos controladores projetados nos itens 1 e 2. Justifique a sua resposta.

```
% Encontrando o tempo de estabelecimento à 5% do P
settling_time_P = stepinfo(Gmf,SettlingTimeThreshold=0.05).SettlingTime;

% Plot da saida
plot(t, y1_t, 'DisplayName', 'Com controlador Proporcional');
hold on;
% Plot da referencia (degrau unitario)
plot(t, y2_t, 'DisplayName', 'Com controlador PI');
plot(t, U_t, 'k--', 'DisplayName', 'Referência');

% Adicionando linha com informações
line([0, t(end)], [1.05, 1.05], 'Color', '#F38F00', 'LineStyle', '--',
'DisplayName', '+ 5%');
line([0, t(end)], [0.95, 0.95], 'Color', '#F38F00', 'LineStyle', '--',
'DisplayName', '- 5%');
line([0, t(end)], [1.1, 1.1], 'Color', 'r', 'LineStyle', '--',
'DisplayName', '+ 10%');

line([settling_time_PI, settling_time_PI], [0, 1], 'Color', 'g',
'LineStyle', '--', 'DisplayName', ['Tempo de Estabelecimento com PI: '
num2str(settling_time_PI, '%.2f')]);
```

```
line([settling_time_P, settling_time_P], [0, 1.05], 'Color', 'g',
'LineStyle', '--', 'DisplayName', ['Tempo de Estabelecimento com P: '
num2str(settling_time_P, '%.2f')]);

title('Resposta à entrada degrau unitário');
legend('Location', 'southeast');
hold off;
```



Analisando as respostas transitórias à entrada ao degrau com D = 0, percebe-se que a do sistema com controlador PI possui um menor sobressinal e um menor tempo de estabelecimento em relação ao sistema com controlador proporcional, seguindo assim a especificação do projeto do PI, porém demora mais para chegar ao valor estacionário (zero próximo à origem introduzido pelo PI torna a resposta mais lenta). Quanto à resposta em regime, nota-se que nenhuma apresenta erro estacionário, o que é esperado, pois a função de transferência da malha fechada do sistema do item 1 é do tipo 1, e a do item 2 é do tipo 2, visto que o controlador PI introduz um polo na origem.

Para comparar as duas respostas em regime à entrada degrau de distúrbio unitário com R = 0, vamos fazer a análise do erro estacionário para o sistema do item 2, assim como já foi feito para o sistema do item 1. Partindo da equação (2), temos:

$$\frac{C(s)}{D(s)} = \frac{G_p(s)}{1 + G_c(s)G_p(s)} = \frac{\frac{2}{s(s+4)}}{1 + \frac{K_p\left(s + \frac{K_i}{K_p}\right)}{s} \cdot \frac{2}{s(s+4)}} = \frac{2s}{s^3 + 4s^2 + 2K_p s + 2K_i}.$$

A entrada é um degrau unitário, portanto $D(s) = \frac{1}{s}$. Logo, $C(s) = \frac{2s}{s^3 + 4s^2 + 2K_p s + 2K_i} \cdot \frac{1}{s}$

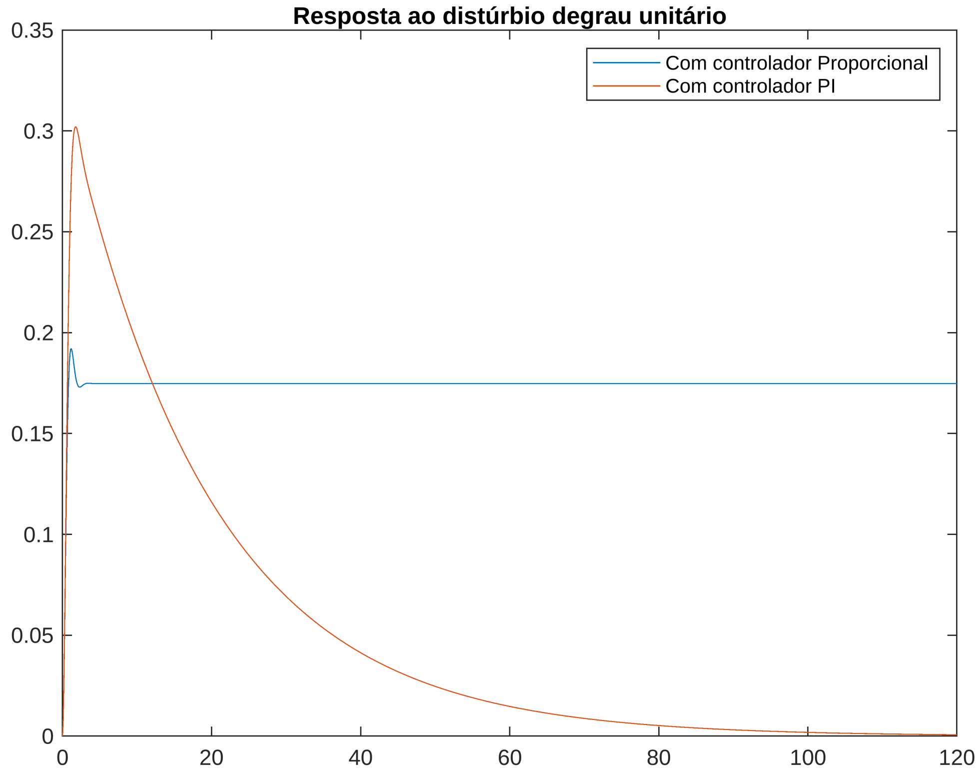
Agora, usamos o teorema do valor final para encontrar o valor de $C(s)$ em regime :

$$c_{ss} = \lim_{s \to 0} sC(s) = \lim_{s \to 0} s \cdot \frac{2s}{s^3 + 4s^2 + 2K_p s + 2K_i} \cdot \frac{1}{s} = 0 .$$

Dessa forma, é esperado que o controlador PI rejeite o distúrbio com entrada degrau.

Plotando as respostas, obtemos o seguinte gráfico:

```
% Plot da saida
t = 0:0.1:120;            % Vetor de tempo
U_t = ones(size(t));      % Degrau unitário
yd1_t = lsim(Gdf, U_t, t);
plot(t, yd1_t, 'DisplayName', 'Com controlador Proporcional');
hold on;

Gdf2 = feedback(G, Gc);
yd2_t = lsim(Gdf2, U_t, t);

plot(t, yd2_t, 'DisplayName', 'Com controlador PI');

title('Resposta ao distúrbio degrau unitário');
legend();
hold off;
```



Percebe-se que a resposta com controlador PI vai para zero, como esperado, e leva um tempo muito maior para chegar a esse valor final em comparação ao tempo que o sistema com controlador proporcional leva para se estabilizar.