Sistemas Realimentados - 2023/2

Nome: Dionatas Santos Brito

Data limite para entrega: 4/10, 6h da manhã .

Trabalho 3 - Projeto de controladores PID via método do lugar das raízes.

```
I=8; % Seu valor de I
[G1,G2,G3, iae_G1, iae_G3, ts_G2]=ini_t3(I);
datetime('now')
```

```
ans = datetime
   04-Oct-2023 07:07:17
```

Atividade 1: Projeto de um controlador PI para sistema de primeira ordem + tempo morto (G1(s)).

Projetar um controlador PI via método do lugar das raízes usando a FT G1. O controlador resultante C1 deve resultar em um valor de $IAE \leq iae_{G1}$ e erro nulo para entrada degrau.

Mostrar o controlador e o LR utilizado, explicar as escolhas da localização do zero do controlador para atender a especificação e a obtenção de Kp e Ki do LR.

```
G1
```

```
G1 =

                9
  exp(-7*s) * --------
              28 s + 1

Continuous-time transfer function.
Model Properties
```

```
iae_G1
```

```
iae_G1 = 15.6732
```

O controlador Proporcional-Integral (PI) é um tipo de controlador que combina as ações proporcional e integral. Ele possui o seguinte formato:

$C(s) = K * \frac{s + \frac{\mathrm{Ki}}{\mathrm{Kp}}}{s}$, onde a ação proporciona (Kp) ajusta o ganho do sistema, enquanto a ação integral (Ki) elimina o erro estático do sistema,

- O primeiro passo para criar um controlador PI com tempo morto via Lugar das Raízes é utilizar a apoximação de Pade:

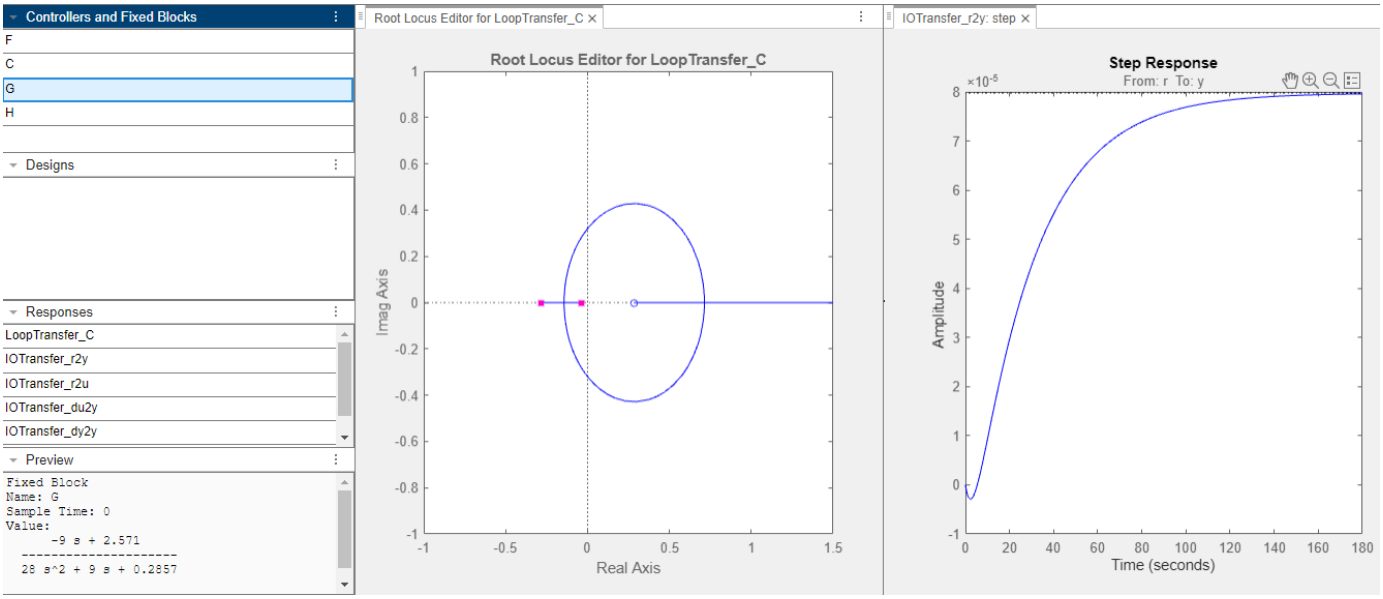
```
G1_MODEL=pade(G1,1)
```

```
G1_MODEL =

      -9 s + 2.571
  --------------------
  28 s^2 + 9 s + 0.2857

Continuous-time transfer function.
Model Properties
```

```
pole_G1_MODEL = pole(G1_MODEL)
```

```
pole_G1_MODEL = 2×1
   -0.2857
   -0.0357
```

Figura rltool referente ao LR de G1 com aproximação de Pade:

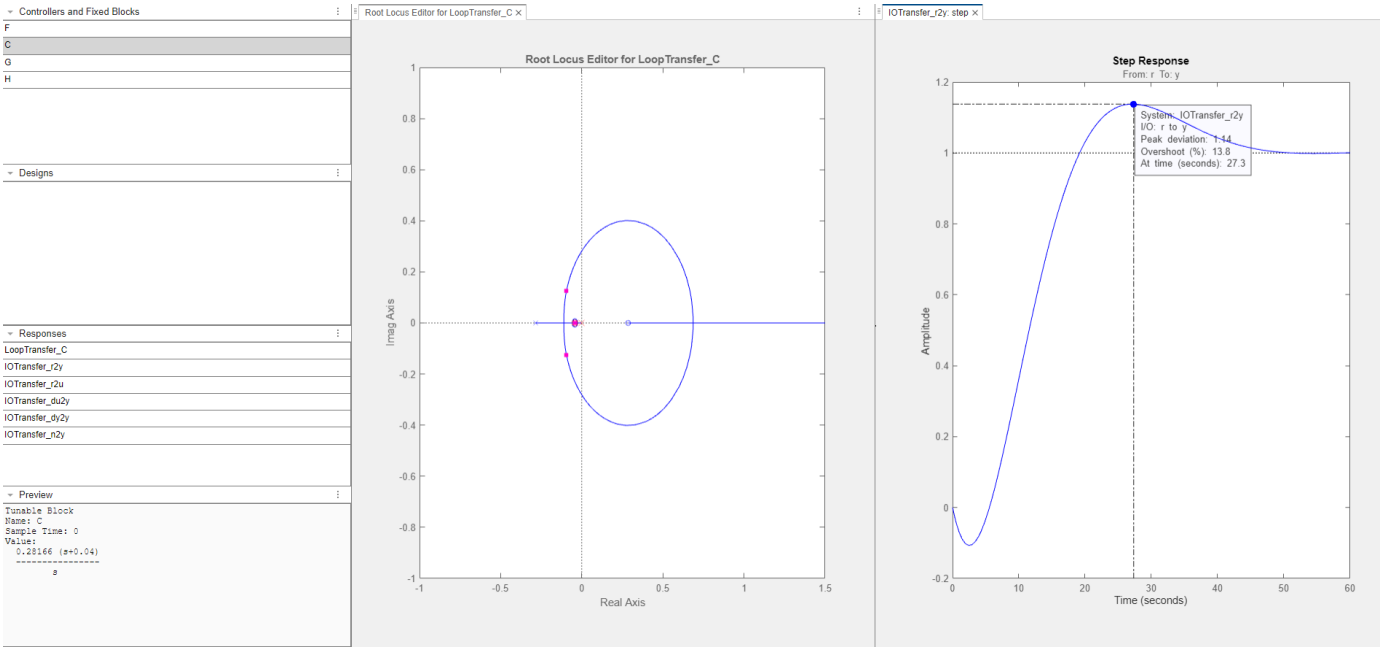


- O segundo passo é abrir o lugar das raízes (rltool), adicionar um polo na origem + um zero ajustável.

O polo do controlador PI está localizado em $S = 0$ (devido ao termo integral), e o zero está localizado em $-\frac{K_p}{K_i}$.

```
%Pontos retirados do rltool
s=tf('s');
KP_C1 = 0.28166;
C1=  KP_C1*(s+0.04)/s;
stepinfo(G1);
```

Figura referente ao LR ao controlador PI:




```
%Controlador PI final * G1 com aproximação de Pade (após adicionar um polo  em 0 e um zero em -0.5)
Final_Control= C1*G1_MODEL
```

```
Final_Control =

  -2.535 s^2 + 0.6229 s + 0.02897
  -------------------------------
     28 s^3 + 9 s^2 + 0.2857 s

Continuous-time transfer function.
Model Properties
```

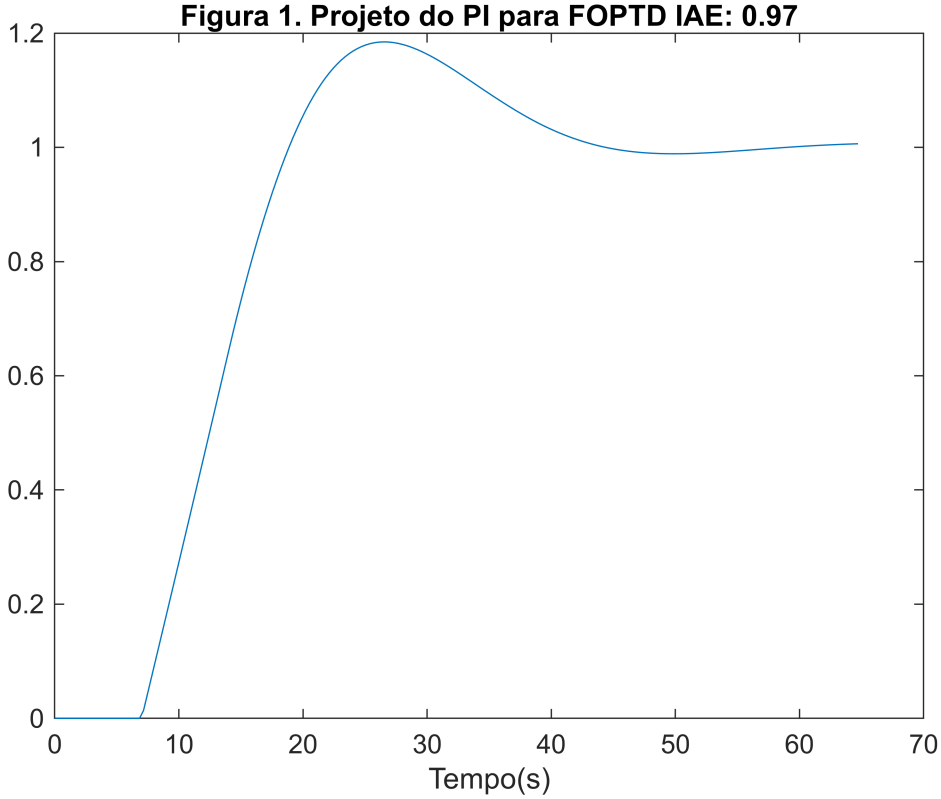
Apresentação do resultado: simulação e o cálculo do IAE usando o controlador C1 projetado.

```
M1=feedback(C1*G1,1);
[y,t]=step(M1);
erroG1=1-y(end)
```

```
erroG1 = -0.0062
```

```
t=linspace(0,max(t),200);
y=step(M1,t);
plot(t,y);xlabel('Tempo(s)');
iae1=trapz(t,abs(1-y));
ss=sprintf('Figura 1. Projeto do PI para FOPTD IAE: %3.2f', iae1/iae_G1);
title(ss);
```

Figura 1. Projeto do PI para FOPTD IAE: 0.97



Análise:

O projeto do PI via LR envolve escolher a localização do zero e o ganho Kp. Depois disto, calcula-se o ganho KI. Alguns efeitos:

- O zero do PI próximo à origem tende a deixar a resposta lenta
- O zero do PI muito longe da origem tende a deixar a resposta menos amortecida (maior sobreelevação)
- O ganho Kp deve ser aumentado de modo a conseguir respostas mais rápidas
- IAE é a integral do erro, quanto mais rápido o erro sumir e menor sobreelevação, menor será o valor de IAE

Para a condição de erro nulo: Adiconei um polo na origem (Integrador) e um zero em 0.04 para atender o erro em regime utilizando o rltool.

Olhando para o Kp, para valores maiores que 0.6 aproximadamente, a resposta em malha fechada tornou muito oscilatória, rápidamente instável a medida que se aumentava esse termo.

Olhando para o polo, foi testado tanto o zero próximo a origem quanto mais distante, e obtive o seguinte resultado:

- Próximo: Como se localizava mais perto do polo de valor -0.2857 (Polos entre a origem), gerou uma resposta mais lenta com baixa sobreelevação, entretanto, com o ajuste do Kp foi possível deixa-la com resposta mais rápida, com baixa sobreelevação (18.4%) e com um tempo de estabelecimento relativamente baixo (41s) ao se comparar com o ts de G1.

- Distante: Quanto mais distânte da origem (-0.4 por exemplo) a resposta gerava mais sobreelevação, aumentando o IAE, tempo de estabilização e consequentemente, o ero em regime.

Portanto para atender o projeto, a minha escolha de polo foi mais próximo a origem, obtendo:

- Kp = 0.28166
- Ki/Kp = -0.04
- Ki = 0.0113 (aproximadamente)
- erroG1 = -0.0062
- Sobreelevação: 18.4%
- Tempo de Estabelecimento: 41.26

Figura referente ao LR ao controlador PI pode ser vista mais a cima da Ánalise.

```
step_M1 = stepinfo(M1);
ts_M1= step_M1.SettlingTime
```

```
ts_M1 = 41.2644
```

```
up_M1 = step_M1.Overshoot
```

```
up_M1 = 18.4614
```

Atividade 2: Projeto de um controlador C2 tipo PD via método do LR para o modelo de ordem 2 G2(s) que permita obter o tempo de estabilização $ts \leq ts_{G2}$ e com sobreelevação menor que 1%.

Mostrar o controlador e o LR utilizado, explicar as escolhas da localização do zero do controlador para atender a especificação e a obtenção de Kp e Kd do LR.

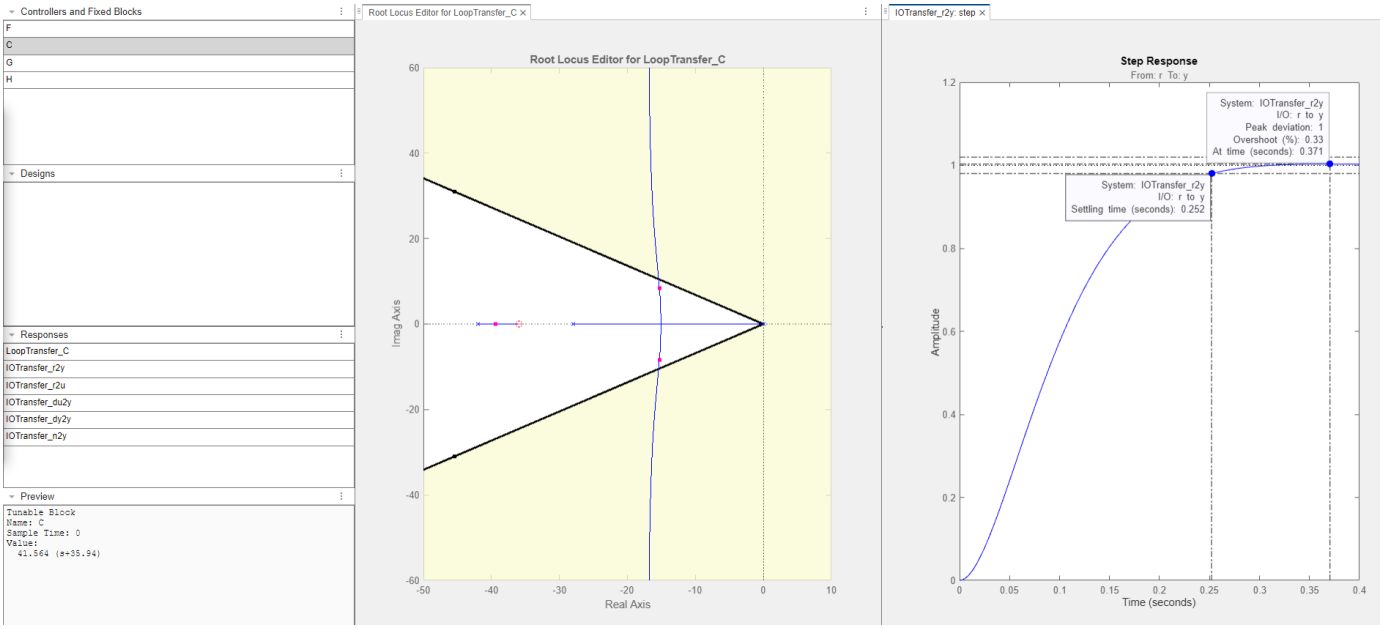
```
ts_G2
```

```
ts_G2 = 0.2778
```

```
C2 = 41.57*(s+35.94)
```

```
C2 =

  41.57 s + 1494

Continuous-time transfer function.
Model Properties
```

Abaixo a simulação com o controlador C2 projetado.




```
M2=feedback(C2*G2,1);
[y,t]=step(M2);
step_G2= stepinfo(M2);
ts_G2= step_G2.SettlingTime
```

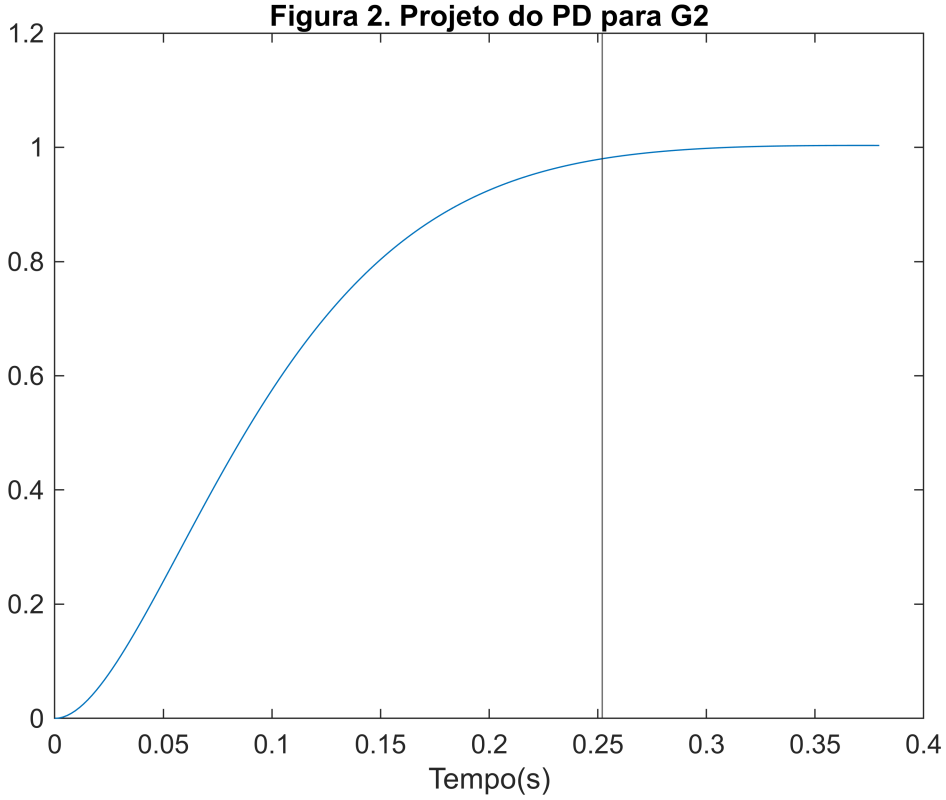
```
ts_G2 = 0.2520
```

```
up_G2 = step_G2.Overshoot
```

```
up_G2 = 0.3304
```

```
t=linspace(0,max(t),200);
y=step(M2,t);
plot(t,y);xline(ts_G2);
xlabel('Tempo(s)');title('Figura 2. Projeto do PD para G2')
```

Figura 2. Projeto do PD para G2



Ánalise:

Um controlador PD (Proporcional-Derivativo) é um tipo de controlador composto por dois termos: o termo proporcional (P) e o termo derivativo (D).

Efeito:

O PD ajuda a evitar/ reduzir o overshoot e melhorar a resposta transitória, permitindo uma resposta mais rápida com uma melhor estabilidade doque utilizar o controlador puramente Proporcional (P).

- Proximo a origem: Aproximar o zero do PD da origem aumenta o efeito derivativo, e permite assim aumentar o ganho proporcional, que torna a resposta mais rápida.
- Longe da origem: Quanto mais se movimentava o zero do PD para longe da origem, o Tempo de estabelecimento e o Overshoot aumentava consideravelmente, fugindo da especificação pedida de $ts \leq ts_{G2}$ e com sobreelevação menor que 1%. (Zero após o ultimo polo da Esquerda)

Portanto para atender o projeto, a minha escolha de Zero do PD foi o mais próximo a origem (antes do ultimo polo da esquerda), em -35.94. Nele obtive:

- Kp = 41.57
- Sobreelevação = 0.33%
- Tempo de Estabelecimento = 0.2520 s

Atividade 3: Projeto de um controlador C3 PI ou PID para o modelo de ordem 4 G3(s) tal que se tenha $IAE \leq iae_{G3}$.

Mostrar o(s) LR utilizado(s), explicar as escolhas para obter o controlador e atender a especificação e a obtenção dos ganhos do PID no LR.

```
G3
```

```
G3 =

                  400
  -------------------------------------
  s^4 + 28 s^3 + 294 s^2 + 1372 s + 2401

Continuous-time transfer function.
Model Properties
```

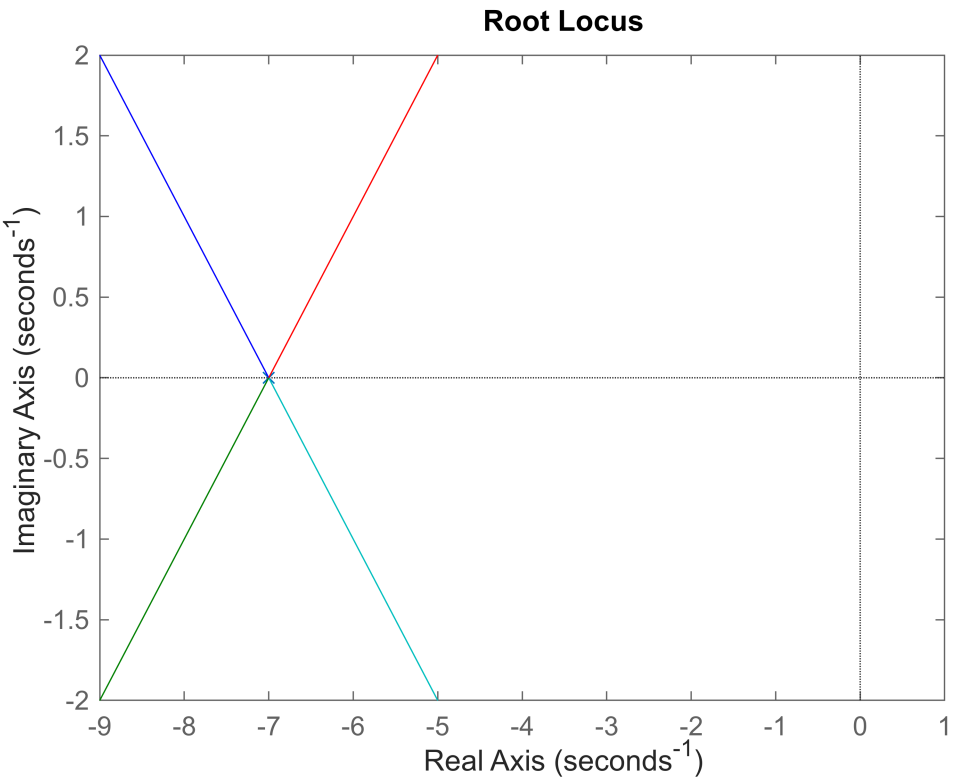
```
iae_G3
```

```
iae_G3 = 0.5122
```

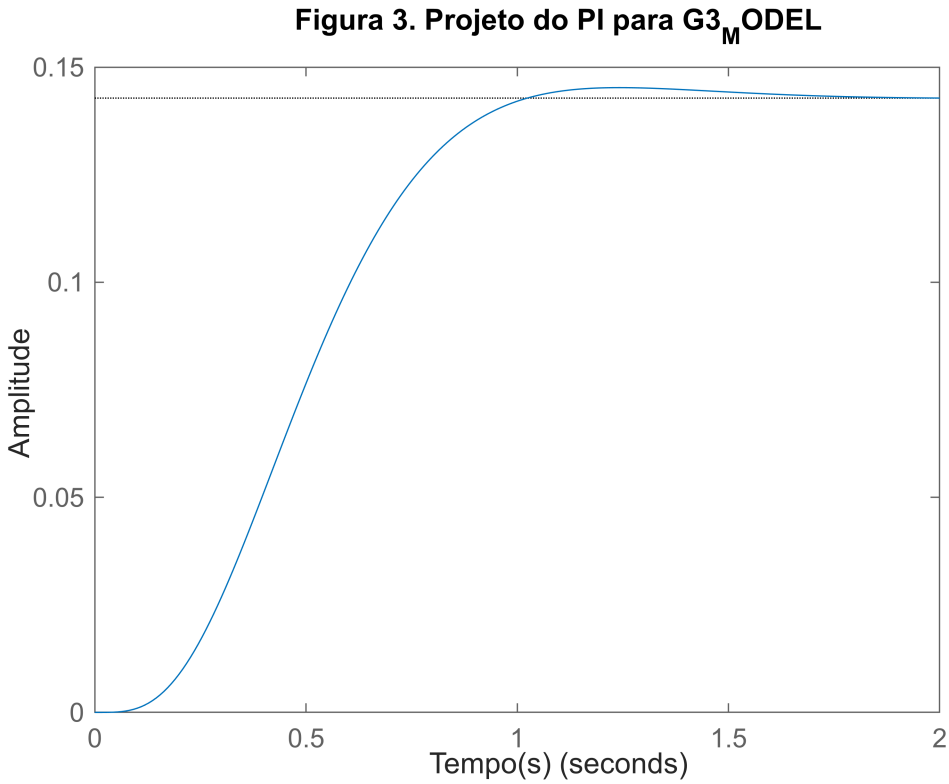
Para projetar um controlador de forma a garantir que o erro em regime seja praticamente nulo, que tenha $IAE \leq iae_{G3}$, e que tanto o tempo de estabelecimento quanto a sobrelevação sejam mínimos para que haja pouca discrepância entre a resposta e a referência de degrau unitário, é possível empregar um controlador PI.

Devemos iniciamente plotar o estado que precisamos aproximar para a específicação:

```
rlocus(G3)
```




```
G3_MODEL = feedback(G3,1);
step(G3_MODEL);xlabel('Tempo(s)');title('Figura 3. Projeto do PI para G3_MODEL')
```



Segundo a resposta ao degrau unitário em malha fechada temos:

```
G3_step = stepinfo(G3);

%Tempo de Estabelecimento
G3_ts = G3_step.SettlingTime
```

```
G3_ts = 1.2978
```

```
%Sobreelevação
G3_UP = G3_step.Overshoot
```

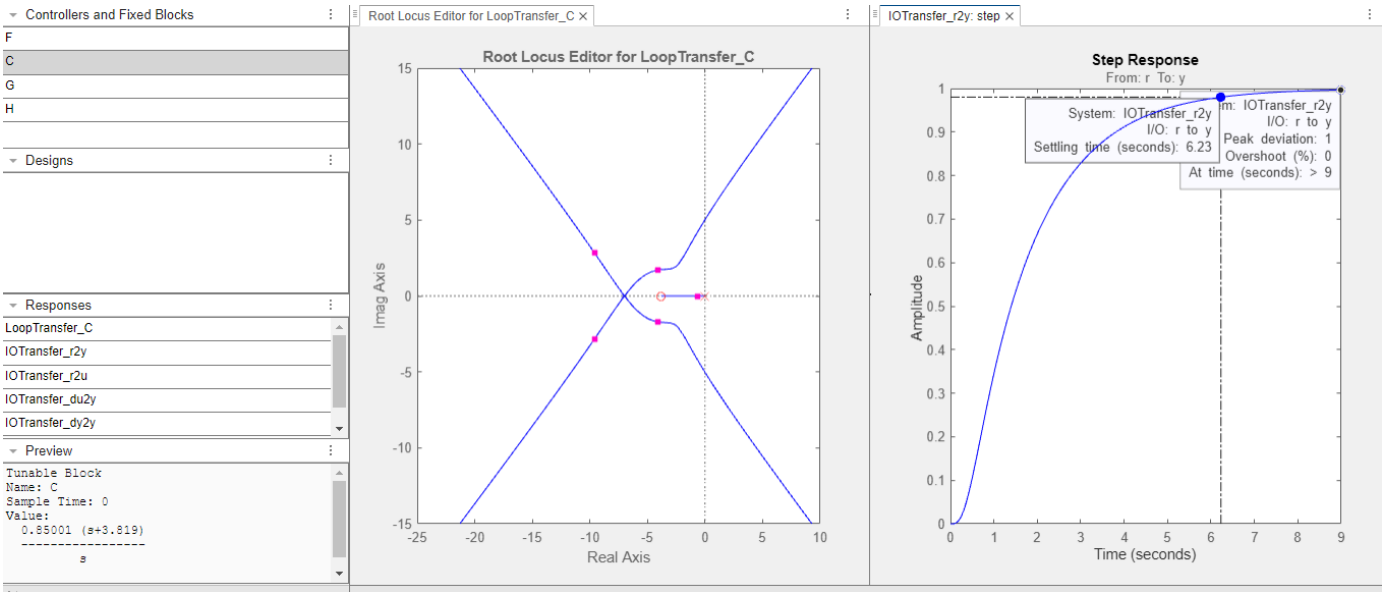
```
G3_UP = 0
```

Análise para o PI:

Dado essas informações, tomarei a hipósite projetar um controlador PI afim de eliminar o erro em regime, mantendo o valor tempo de estabelecimento e de sobreelevação o mais baixo possível.

O projeto do PI se dá como na questão 1, adicionando um polo na origem para deixar o erro próximo a nulo e um zero, variando o Kp.

Caso seja o zero seja adicionado próximo a origem, o sistema iria ficar lento e ao variar o Kp, iria torna-lo mais rápido.



Para a construção do PI acima, obtive os seguintes efeitos:

Ao distanciar o Zero do PI da origem, o tempo de estabelecimento ficava menor, enquanto a sobreelevação ficava maior (por conta que era necessário aumentar o Kp para atender as especificações). Dessa forma, os resultados obtidos não se distanciavam muito dos observados para o controlador PI com IAE, na resposta ao degrau em malha fechada, chegando próximo a 1.8438 com o erro de regime igual a 0.0036, que ao se plotar (utilizando o código do professor) o IAE_PI/ iaeg3, gera 3.60. Dito isso, é possivel notar que um controlador PI sozinho não consegue cumprir com a especificação e é necessário utilizar ele juntamente com o PD, formando o PID.

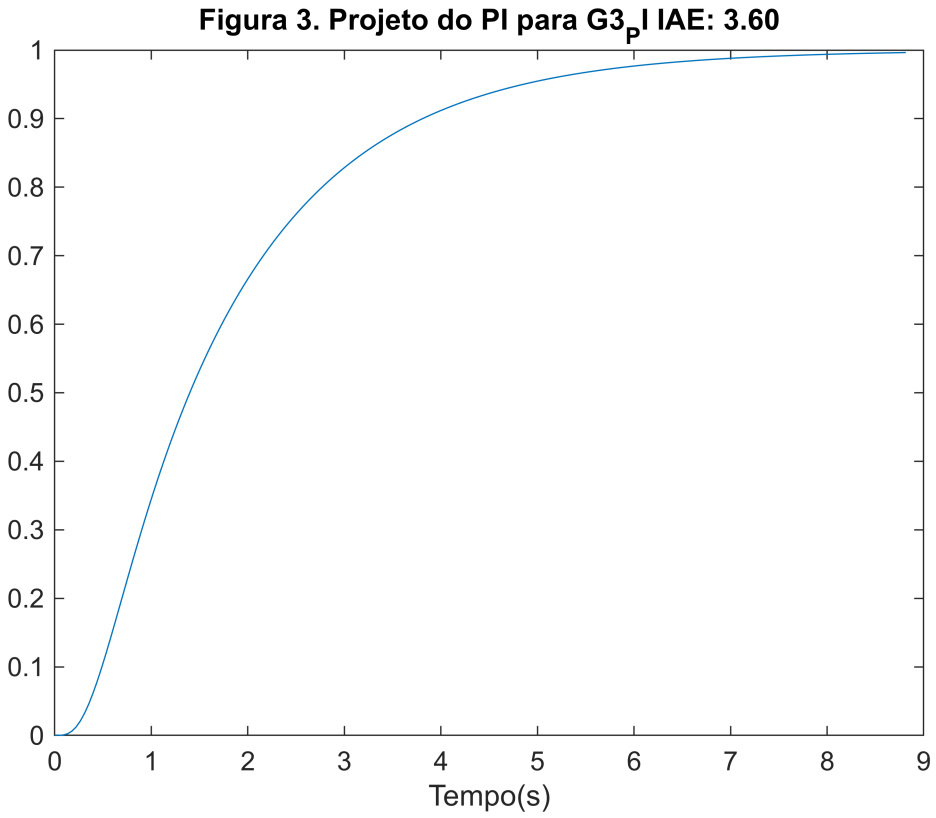
```
C3_PI=0.85*((s+3.819)/s);
M3_PI=feedback(C3_PI*G3,1);
[y,t]=step(M3_PI);
t=linspace(0,max(t),200);
y=step(M3_PI,t);
plot(t,y);xlabel('Tempo(s)');
iae3_PI=trapz(t,abs(1-y))
```

```
iae3_PI = 1.8438
```

```
ErroG3_PI=1-y(end)
```

```
ErroG3_PI = 0.0036
```

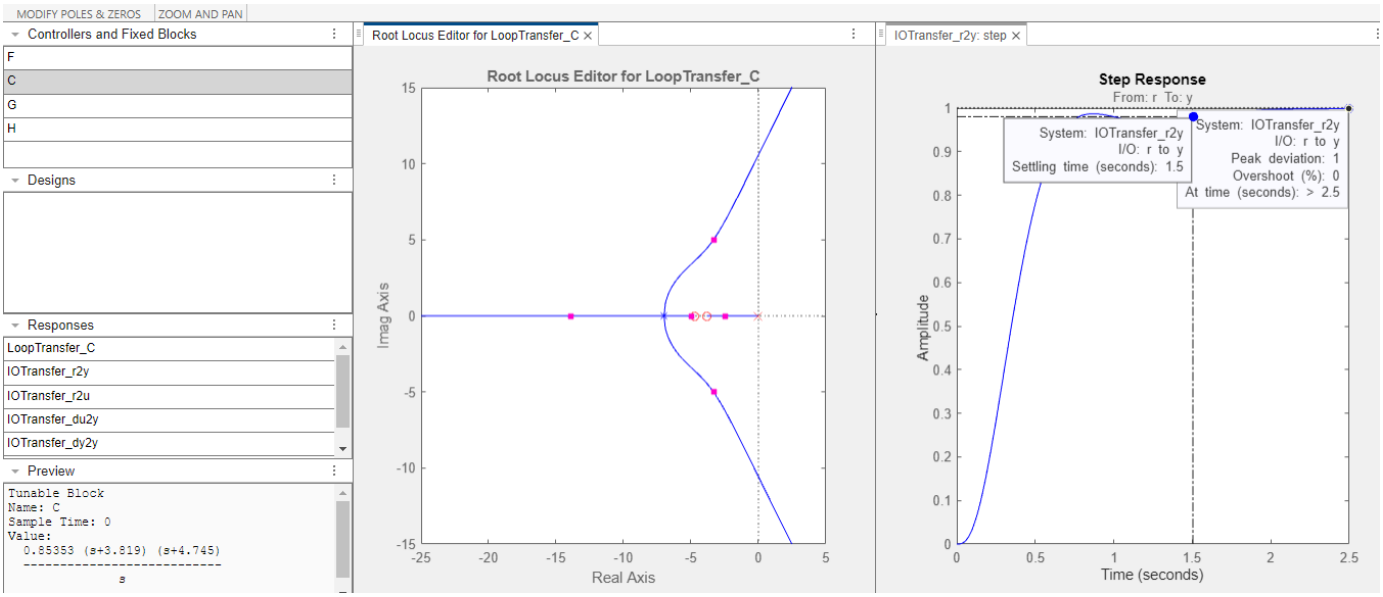
```
ss=sprintf('Figura 3. Projeto do PI para G3_PI IAE: %3.2f', iae3_PI/iae_G3);
title(ss);
```



Análise para o PID:

Como o projeto do controlador PI não atendeu as especificações de sobreelevação, neste caso o efeito derivativo deve ser incluído, resultando em um controlador PID.

O PD se dá como na questão 2, adicionando um zero do PD.




```
C3=0.85353*((s+3.819)*(s+4.745)/s);
```

Abaixo a simulação e o cálculo do IAE usando o controlador C3 projetado.

```
M3=feedback(C3*G3,1);
[y,t]=step(M3);
t=linspace(0,max(t),200);
y=step(M3,t);
plot(t,y);xlabel('Tempo(s)');
iae3=trapz(t,abs(1-y))
```

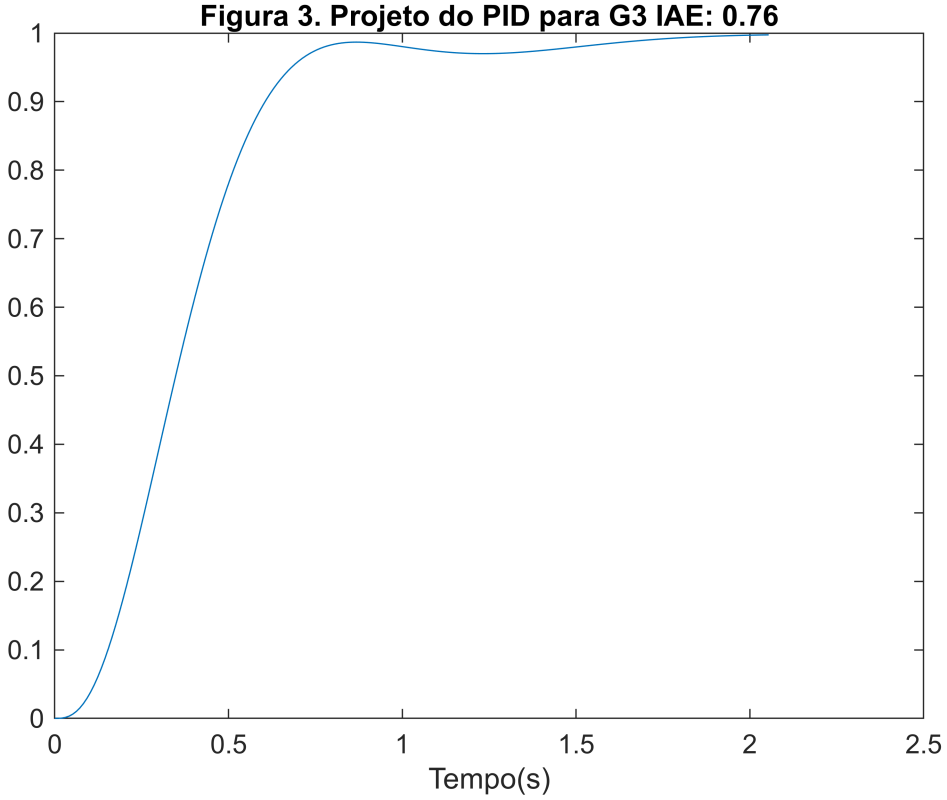
```
iae3 = 0.3868
```

```
erroG_13PID=1-y(end)
```

```
erroG_13PID = 0.0026
```

```
ss=sprintf('Figura 3. Projeto do PID para G3 IAE: %3.2f', iae3/iae_G3);
title(ss);
```

Figura 3. Projeto do PID para G3 IAE: 0.76



Ao adicionar o zero do PD em -4.745 eu atendi aos requisitos do controlador.

Efeitos:

Adicionado após o ponto de sela: No meu caso, ao mover o zero do PD para longe da origem (Após o ponto de sela), ocasionou em uma maior sobreelevação.

Adicionando antes do polo do PI: Nesse caso, o sistema ficou muito lento aumentando o tempo de estabilização

Adicionando entre o ponto de sela e o polo do PI: Aqui foi o lugar ideal (no meu caso), pois não houve muita sobreelevação e teve menor tempo de estabilização, ao se comparar com os outros locais, ficando justamente para ajustar o valor atráves do Kp.

Portanto, como na questão 2, aproximar o zero do PD da origem aumenta o efeito derivativo, e permite assim aumentar o ganho proporcional, que torna a resposta mais rápida.

Entretanto o melhor Ts e Overshoot se deu entre o ponto de sela e o polo do PI, obtendo os seguintes valores:

- Kp = 0.85353
- Sobreelevação = 0%
- Tempo de Estabelecimento = 1.5 s
- Erro em regime: 0.0026

Ao se comparar a análise do PI e do PID, é possivel notar que o PID gerou o menor IAE. Dito isso, foi possível atingir a especificação.

```
%Tempo de Estabelecimento
G3pi_ts = G3_pistep.SettlingTime
```

```
G3pi_ts = 1.5044
```

```
%Sobreelevação
G3_piUP = G3_pistep.Overshoot
```

```
G3_piUP = 0
```