



UFES - UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

CT - Centro Tecnológico

Departamento de Engenharia Elétrica

Disciplina: Sistemas Realimentados

Professor: José Leandro Félix Salles

Alunos: Arthur Sorrentino Ferreira e Higor David Oliveira

Trabalho Computacional I

1.1 Uma empilhadeira eletro-hidráulica potente pode ser usada para erguer material paletizado pesando várias toneladas no topo de uma plataforma de 10 m em um canteiro de obras. O sistema de controle com realimentação negativa possui uma função de transferência em malha aberta (atuador+planta+sensor) dada por:

$$G_{MA} = \frac{s^2 + Ns + 10}{s^2(s + \sqrt{N})}$$

Qual controlador você escolheria (P,PD, ou PI) para que a resposta ao degrau seja a mais rápida possível, com o menor sobressinal e erro em regime à rampa seja zero ? Justifique sua resposta usando o método do Lugar das Raízes e projete este controlador escolhido. Justifique, usando o método do lugar das raízes, os motivos para não usar os outros controladores.

Pela função de malha aberta do sistema, sabemos que o processo é do tipo 2. Pela ordem do sistema, sabemos a resposta característica ao degrau, rampa e parábola, (tabela 1 abaixo).

	ess (tipo 0)	ess (tipo 1)	ess (tipo 2)
Degrau	$1/(1+k)$	0	0
Rampa	∞	$1/k$	0
Parábola	∞	∞	$1/k$

Figura 1 - Erro em regime estacionário para sistemas contínuos de tipo 0, 1 e 2.

Como nosso sistema trata-se de um tipo 2, o erro em regime de resposta à rampa é 0, atendendo o último requisito do controlador.

Assim também, como não há necessidade de aumentar o tipo do sistema para reduzir o erro estacionário, adicionar um controle Proporcional-Integrativo só aumentará a instabilidade do sistema, pois adicionaria um pólo na origem.

Como os requisitos do projeto é puramente a resposta mais rápida para o menor sobressinal, isso determina que entre a escolha de um controlador P e PD, é melhor este último, que é conhecido por menor sobressinal e respostas mais rápidas. Por isso, vamos projetar direto o controlador PD.

Para o controlador PD, temos dois parâmetros que podem variar, Kp e Kd. Kp será determinado encontrando a melhor resposta para o controlador P e então, Kd será determinado usando Kp do controlador P.

Fazendo RL de Gma, temos o seguinte RL:

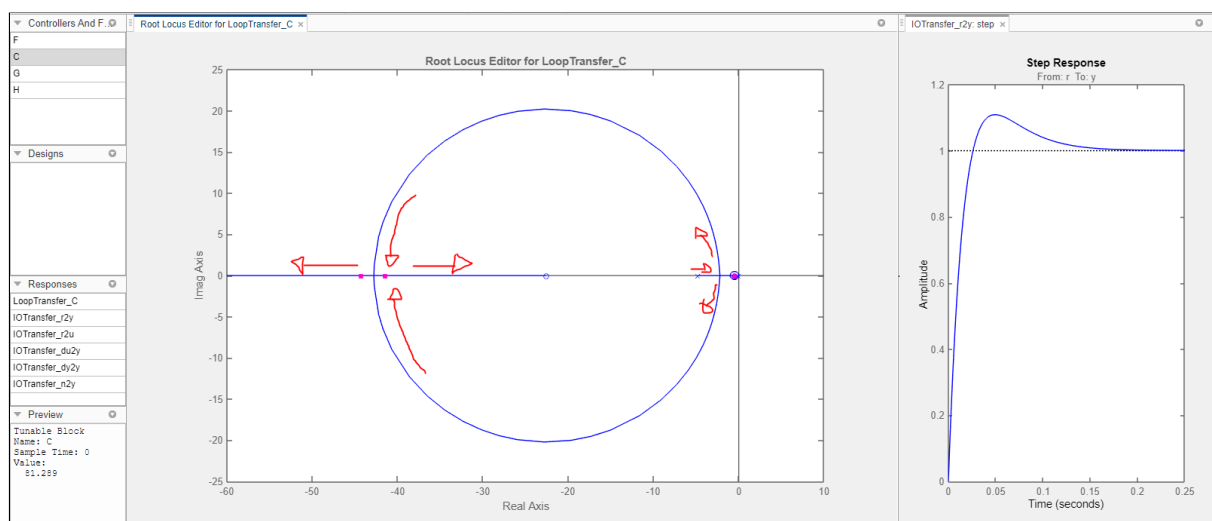


Figura 2 - Lugar das raízes de Kp.

Para escolher o Kp, usamos o critério das raízes estarem sobre o eixo real, para a resposta ser amortecida, reduzindo sobressinal e estarem o mais distante da origem. Então, foi escolhido **Kp = 81.30**, ponto próximo ao demonstrado acima.

Tendo Kp, agora podemos reescrever a equação para fazer o RL de Kd.

$$1 + (sKd + 80.30) \frac{Num}{Den} = 0 \rightarrow 1 + Kd \cdot \frac{Num'}{Den'} = 0$$

Com

$$\frac{Num'}{Den'} = Gma';$$

Com a ajuda do matlab, simplificamos:

$$Den + (sKd + 81.30) * Num = 0$$

...

$$s^3(Kd + 1) + s^2(23Kd + \sqrt{23} + 81.3) + s(10Kd + 23 \cdot 81.3) + 813 = 0$$

$$s^3 + s^2(86.0958) + s(1.8699 \cdot 10^3) + 813 + Kd \cdot (s^3 + 23s^2 + 10s) = 0$$

$$Gma' = \frac{Num'}{Den'} \approx \frac{s^3 + 23s^2 + 10s}{s^3 + 86.1s^2 + 1870s + 813};$$

Então, fazendo RL para Kd, temos o seguinte resultado:

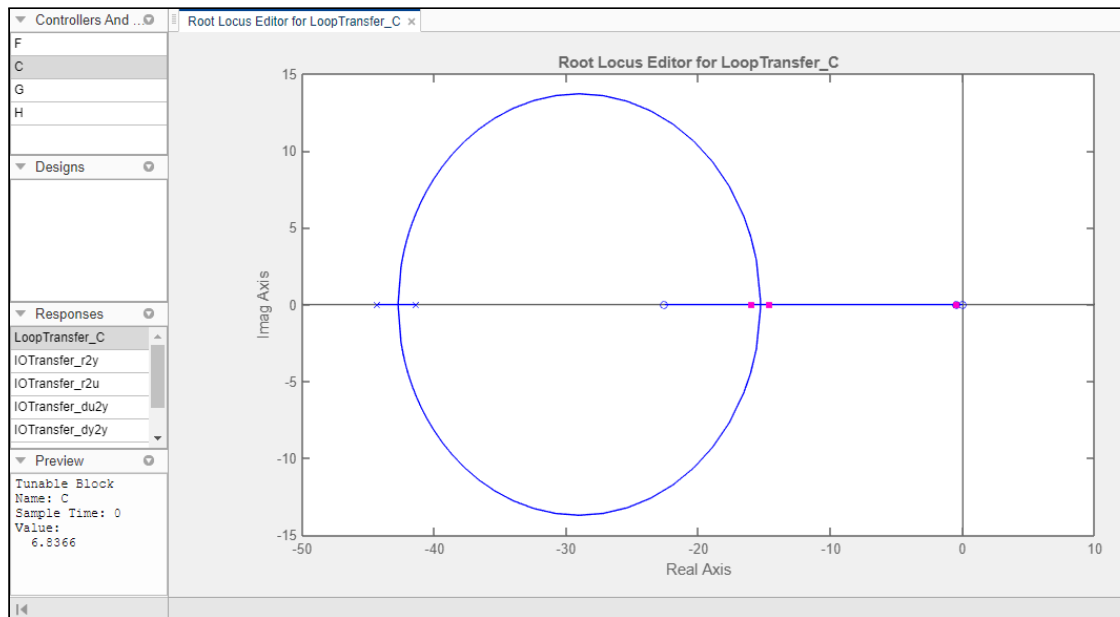


Figura 3 - Lugar das raízes de K_D .

A resposta ao degrau da ferramenta `rltools()` não funcionou da forma esperada, então, serão selecionados valores de K_d e analisada a resposta ao degrau.

Considerando que queremos a resposta mais rápida sem sobressinal, iremos escolher os pontos nas quais as raízes estão sobre o eixo real, à direita e à esquerda do círculo do RL. São esses pontos aproximadamente **$K_d = 0.003$** e **$K_d = 6.8$** . Como a adição de um zero na origem muda a resposta do sistema, apesar das raízes estarem sobre o eixo real, precisamos simular para confirmar que não temos sobressinal.

Antes de simularmos as respostas ao degrau, vale destacar que para a resposta mais rápida no controlador PD, queremos que o zero introduzido pelo controlador derivativo esteja mais próximo da origem, que, no nosso caso, significaria um K_d infinito, o que é inviável. Então, a escolha do valor final será feita pela análise da simulação da entrada em degrau.

As respostas ao degrau para esses dois pontos estão abaixo:

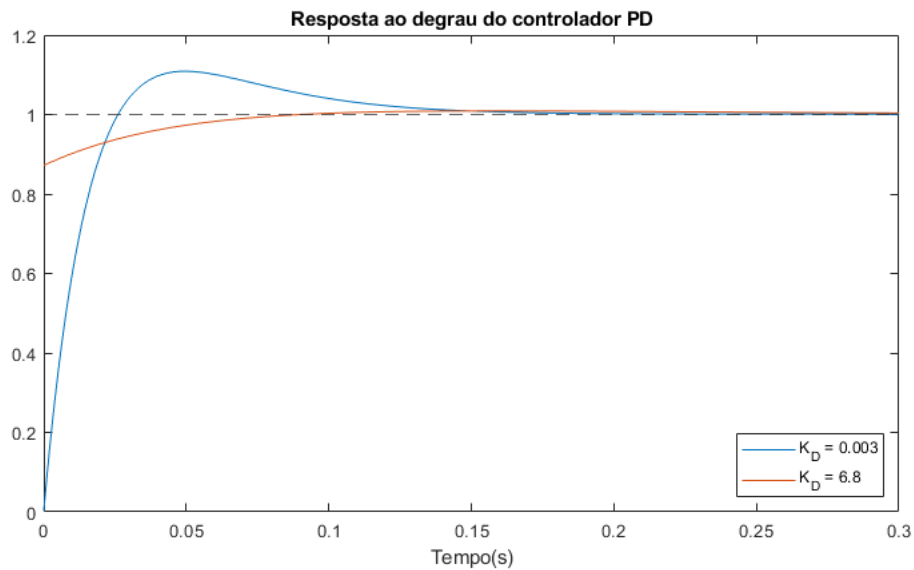


Figura 4 - Resposta ao degrau do controlador PD variando K_D .

Assim, pela resposta ao degrau, conseguimos escolher $K_d = 7$, que representa uma resposta rápida.

Para confirmar o requisito de rampa, iremos simular a resposta a rampa. A resposta para ambos K_d estão abaixo.

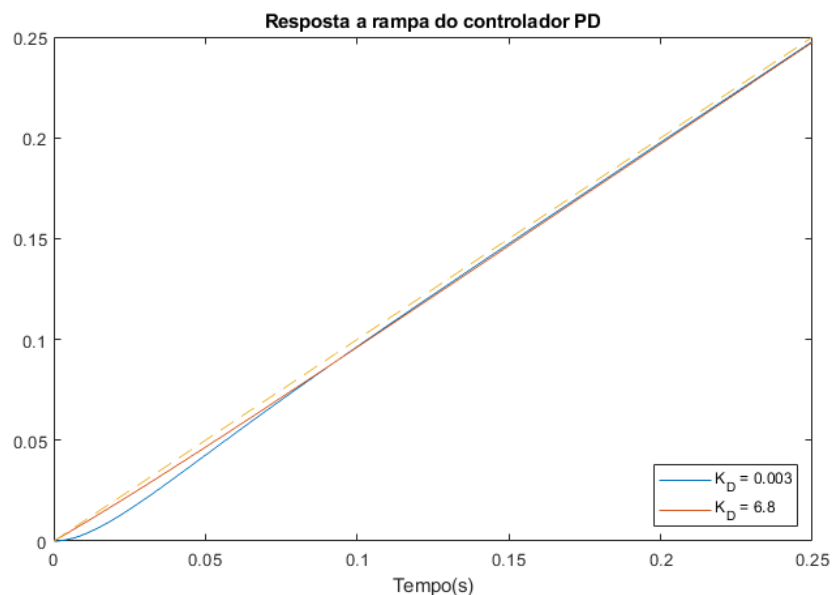


Figura 5 - Resposta a rampa do controlador PD variando K_D .

Pela resposta, conseguimos ver que a resposta segue a rampa, sendo o erro estacionário 0, conforme o esperado. Também, confirmamos que $K_d = 6.8$ apresenta uma resposta mais rápida em relação a $K_d = 0.003$.

Então, como conclusão, para a melhor resposta do sistema dado as especificações de resposta sem sobressinal, com menor tempo de resposta possível e erro estacionário a rampa nulo, temos um **controlador PD** em que os ganhos podem ser **$K_p = 80.3$** e **$K_d = 6.8$** ;

Questão 1

```
% Trabalho Computacional I - Sistemas Realimentados
% Arthur Sorrentino Ferreira e Higor David Oliveira
% N = 23

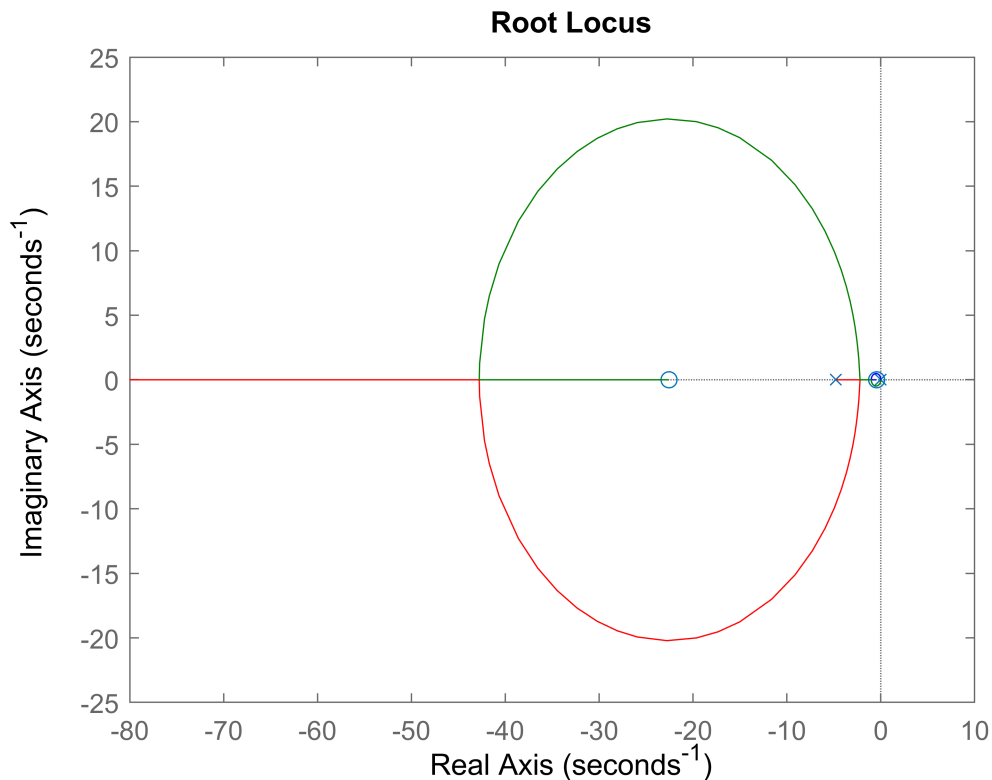
% Especificações:
% - Resposta ao degrau mais rápida possível
% - Menor sobressinal
% - Erro em regime à rampa igual a zero

clc, clear all

% ----- Variáveis principais -----
syms s; N = 23; % número da dupla
num = [1 N 10]; den = conv([1 0 0], [1 sqrt(N)]); % Definição da GMA

% ----- Variáveis secundárias -----
t=0:.001:2;

% ----- Definições -----
gp = tf(num, den); % Gma processo
rlocus(gp);
```

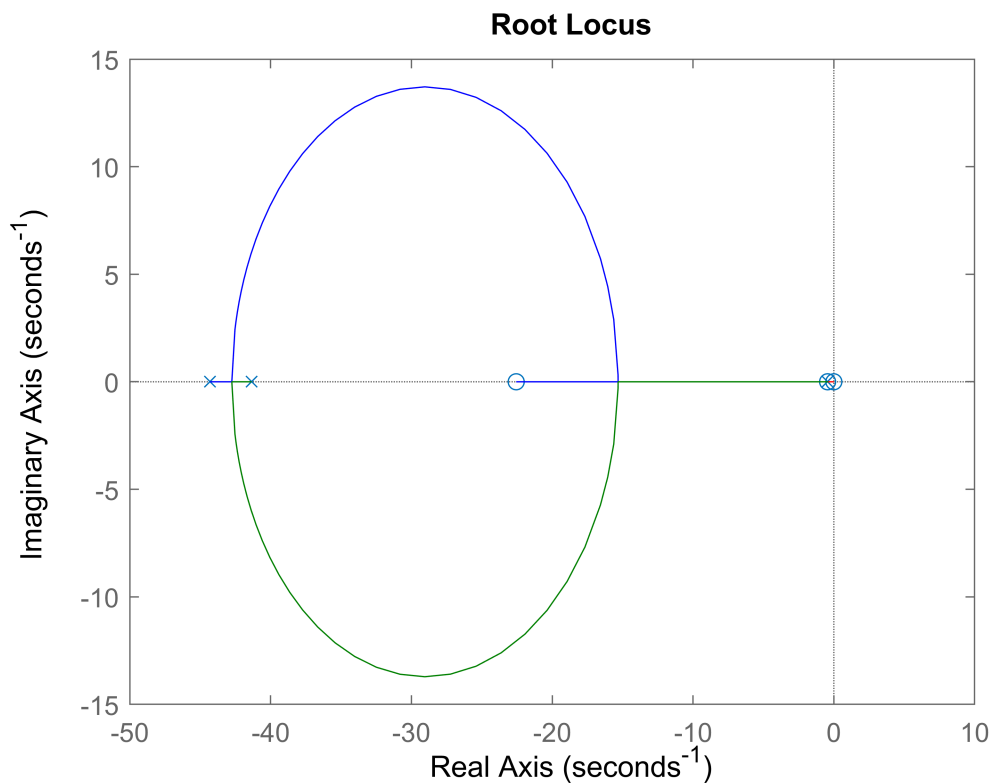


```
kp = 81.30; % Escolha justificada no doc

% ----- RL de Kd -----
num_ = conv(num, [1 0]); % multiplicação por s em cima
den_ = poly2sym(den,s) + poly2sym(num,s)*kp;
```

```
den_ = sym2poly(den_);

figure();
gma_ = tf(num_, den_);
rlocus(gma_)
```



```
kd_1 = 0.003; kd_2 = 6.8; kd_3 = 20; % Valores escolhidos de Kd
```

```
% ----- Respostas ao degrau de Kd -----
gc_pd1 = tf([kd_1 kp],1); % G controlador PD_1
gma_pd1 = gc_pd1*gp % Gma PD
```

```
gma_pd1 =
```

```
0.003 s^3 + 81.37 s^2 + 1870 s + 813
-----
s^3 + 4.796 s^2
```

Continuous-time transfer function.

```
gmf_pd1 = feedback(gma_pd1,1); % Gmf PD
```

```
gc_pd2 = tf([kd_2 kp],1); % G controlador PD_2
gma_pd2 = gc_pd2*gp % Gma PD
```

```
gma_pd2 =
```

```
6.8 s^3 + 237.7 s^2 + 1938 s + 813
-----
s^3 + 4.796 s^2
```

Continuous-time transfer function.

```
gmf_pd2 = feedback(gma_pd2,1); % Gmf PD  
  
gc_pd3 = tf([kd_3 kp],1); % G controlador PD_2  
gma_pd3 = gc_pd3*gp % Gma PD
```

gma_pd3 =

$$\frac{20 s^3 + 541.3 s^2 + 2070 s + 813}{s^3 + 4.796 s^2}$$

Continuous-time transfer function.

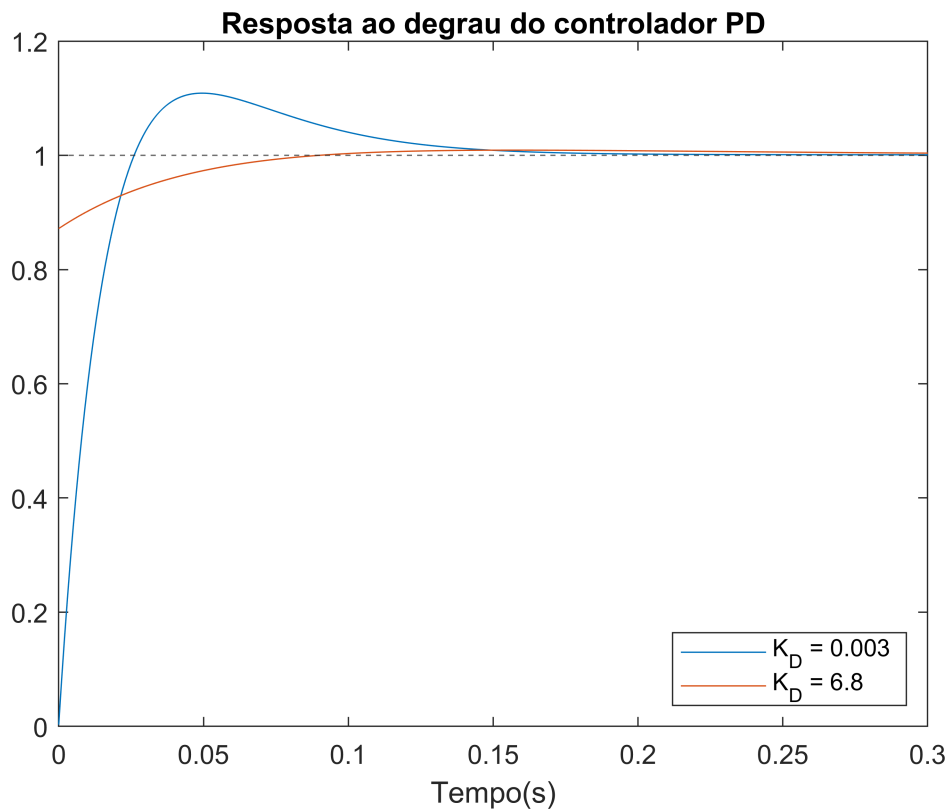
```
gmf_pd3 = feedback(gma_pd3,1); % Gmf PD  
  
y1 = step(gmf_pd1, t); y2 = step(gmf_pd2, t); %y3 = step(gmf_pd3, t);  
stp1 = stepinfo(y1,t);  
stp2 = stepinfo(y2,t);  
%stp3 = stepinfo(y3,t);  
fprintf("Para K_D = %.2d temos UP = %.2d e Ts = %.2d[s]", kd_1, stp1.Overshoot, stp1.SettlingTime)
```

Para K_D = 3.00e-03 temos UP = 1.08e+01 e Ts = 1.23e-01[s]

```
fprintf("E para K_D = %.2d temos UP = %.2d e Ts = %.2d[s]", kd_2, stp2.Overshoot, stp2.SettlingTime)
```

E para K_D = 6.80e+00 temos UP = 8.60e-01 e Ts = 5.73e-02[s]

```
%fprintf("E para K_D = %.2d temos UP = %.2d e Ts = %.2d[s]", kd_3, stp3.Overshoot, stp3.SettlingTime)  
figure()  
plot(t,y1); hold on; plot(t,y2); yline(1, 'LineStyle', '--'); % plot(t,y3);  
title('Resposta ao degrau do controlador PD'), xlabel('Tempo(s)'), legend('K_D = 0.003', 'K_D = 6.800'),  
xlim([0 0.3]);
```



```
% ----- Respostas a rampa de Kd -----
```

```
den_rampa_1 = gmf_pd1.denominator{1}; % recuperando denominador da Gmf
den_rampa_1 = conv(den_rampa_1, [1 0]); % adicionando s para simular uma rampa
gmf_pd1.denominator = {den_rampa_1}; % atualizando gmf1
```

```
den_rampa_2 = gmf_pd2.denominator{1}; % recuperando denominador da Gmf
den_rampa_2 = conv(den_rampa_2, [1 0]); % adicionando s para simular uma rampa
gmf_pd2.denominator = {den_rampa_2}; % atualizando gmf1
```

```
y1_1 = step(gmf_pd1, t); y2_1 = step(gmf_pd2, t);
```

```
stp1 = stepinfo(y1_1,t);
```

```
stp2 = stepinfo(y2_1,t);
```

```
fprintf("Para rampa com K_D = %.2d temos UP = %.2d e Ts = %.2d[s]", kd_1, stp1.Overshoot, stp1.SettlingTime);
```

```
Para rampa com K_D = 3.00e-03 temos UP = 00 e Ts = 1.96e+00[s]
```

```
fprintf("E para K_D = %.2d temos UP = %.2d e Ts = %.2d[s]", kd_2, stp2.Overshoot, stp2.SettlingTime);
```

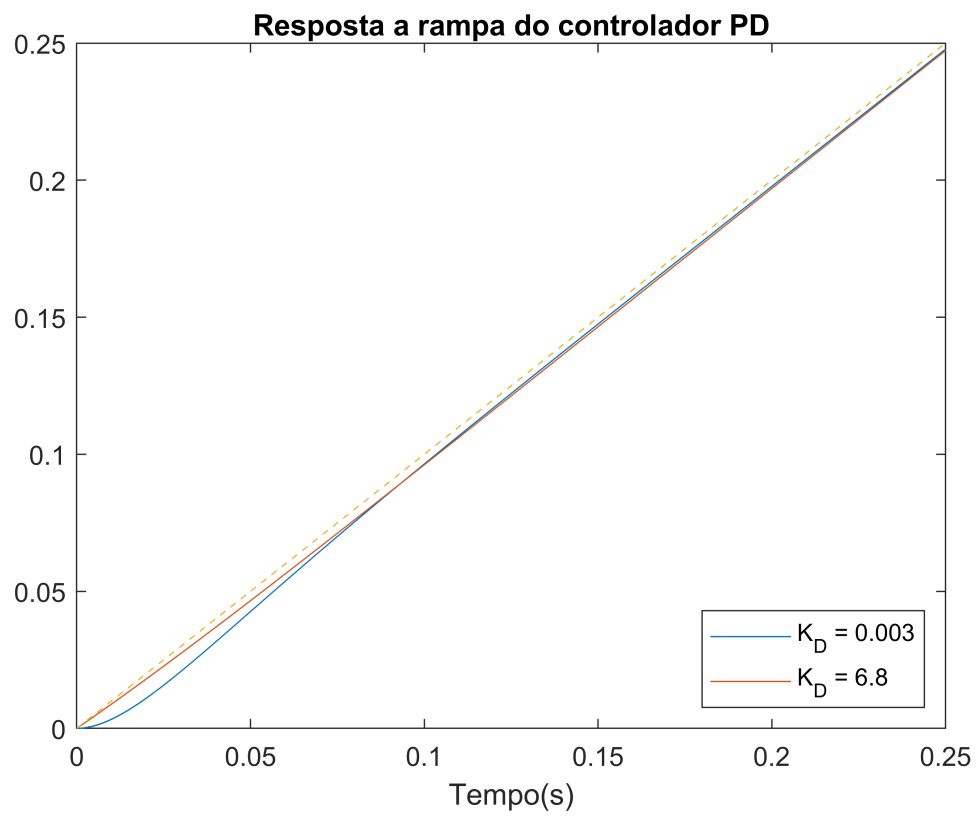
```
E para K_D = 6.80e+00 temos UP = 00 e Ts = 1.96e+00[s]
```

```
figure()
```

```
plot(t,y1_1); hold on; plot(t,y2_1); plot(t, t, 'LineStyle', '--');
```

```
title('Resposta a rampa do controlador PD'), xlabel('Tempo(s)'), legend('K_D = 0.003', 'K_D = 6.8');
```

```
xlim([0 0.25]);
```

1.2 Considere o sistema de controle digital apresentado na Figura 13.26 da pagina 599 do livro do Dorf. Considere o período de amostragem igual a 0,1 segundos e

$$G_p(s) = \frac{Ne^{-(0,1N)s}}{Ns + 10}$$

Use a aproximação de padé para o atraso de transporte:

$$e^{-Ts} = \frac{1 - Ts/2}{1 + Ts/2}$$

E considere que o controlador digital PI seja dado pela seguinte expressão:

$$D(z) = k \frac{z - a}{z - 1}$$

Verifique se é possível obter os parâmetros a e k , usando o método do lugar das raízes, para que a resposta a entrada degrau seja rápida e tenha sobressinal menor possível, com erro em regime à entrada rampa menor que 1. Caso seja possível, especifique-os. Caso contrário explique com detalhes o motivo de não ser possível obter estes parâmetros

Para projetar o controlador PI do sistema acima, primeiramente, vamos fazer análise do RL para um controlador P, determinando o valor de K para um sistema estável e com resposta mais rápida. Em seguida, iremos aplicar a fórmula do controlador PI, usando o valor de K do controlador P e em seguida achar valor de a para atender os requisitos do controlador. Porém, antes de tudo, iremos expandir a função de transferência dado usando a aproximação de Padé, sendo $N = 23$.

$$G_p(s) = \frac{23 \cdot e^{-2.3s}}{23 \cdot s + 10}; \quad \& \quad e^{-2.3s} \simeq \frac{1 - 2.3/2}{1 + 2.3/2}$$

Com ajuda do matlab, desenvolvendo as equações, chegamos em:

$$G_p(s) \simeq \frac{-26.45s + 23}{26.45s^2 + 34.5s + 10}$$

Transferindo para o domínio discreto com tempo de amostragem $T_s = 0.1$, temos:

$$G_p(z) \simeq \frac{0.08953z + 0.09768}{z^2 - 1.874z + 0.8777}$$

Então, primeiramente, com RL de G_p , conseguimos encontrar o valor de K para uma resposta estável.

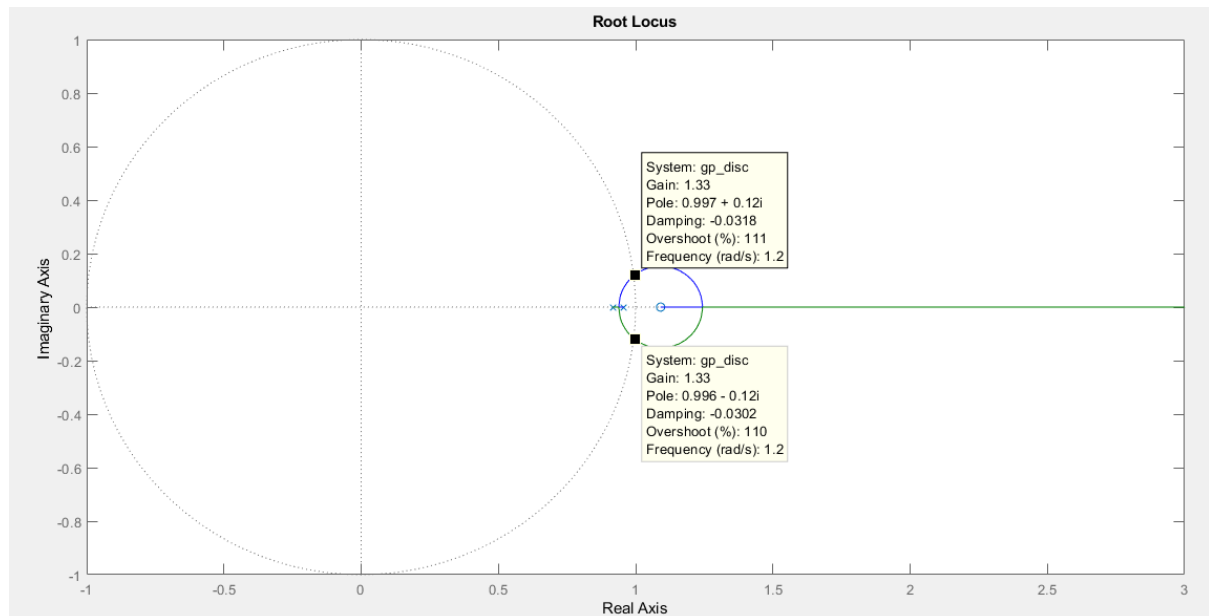


Figura 6 - Lugar das raízes para controlador P discreto.

Com RL, temos que K tem que ser menor que 1.33 para que o sistema seja estável no controlador P. Agora, vamos adicionar o controlador PI e então, por meio de uma análise da equação do erro estacionário e dos pólos e zeros da equação do processo, escolher um valor de a e calcular o valor de K mínimo para atender o requisito de erro em regime menor que 1 para a entrada rampa.

A função de transferência do controlador PI e de malha aberta completa do sistema é dado por:

$$G_C(z) = -\frac{k \cdot (a - z)}{z - 1} \quad \& \quad G_{ma}(z) \simeq \frac{k \cdot (0.895z - 0.0977) \cdot (a - z)}{(z - 1) \cdot (z^2 - 1.874z + 0.8777)}$$

Do RL do processo mostrado anteriormente, vemos que a função tem polos em 0.9575 e 0.9167. Então, primeiramente faremos $a = 0.9575$. Com isso, usando ferramentas do matlab de limite, encontramos o par:

$$K \geq 1.023 \quad \forall \quad a = 0.9575$$

Como o valor mínimo encontrado para K é muito próximo do limite estável do sistema, testaremos posicionar o zero do controlador sobre o outro polo, sendo assim, $a = 0.9167$. Fazendo isso, encontramos o segundo par:

$$K \geq 0.5219 \quad \forall \quad a = 0.9167$$

Então, vamos aplicar o valor possível de “a” e analisar o novo RL e as respostas ao degrau, alterando o valor de K para atender o requisito de menor sobressinal.

Aplicando $a = 0.9167$ no controlador e aplicando o controlador na função de transferência da planta e fazendo um novo RL, temos o seguinte resultado:

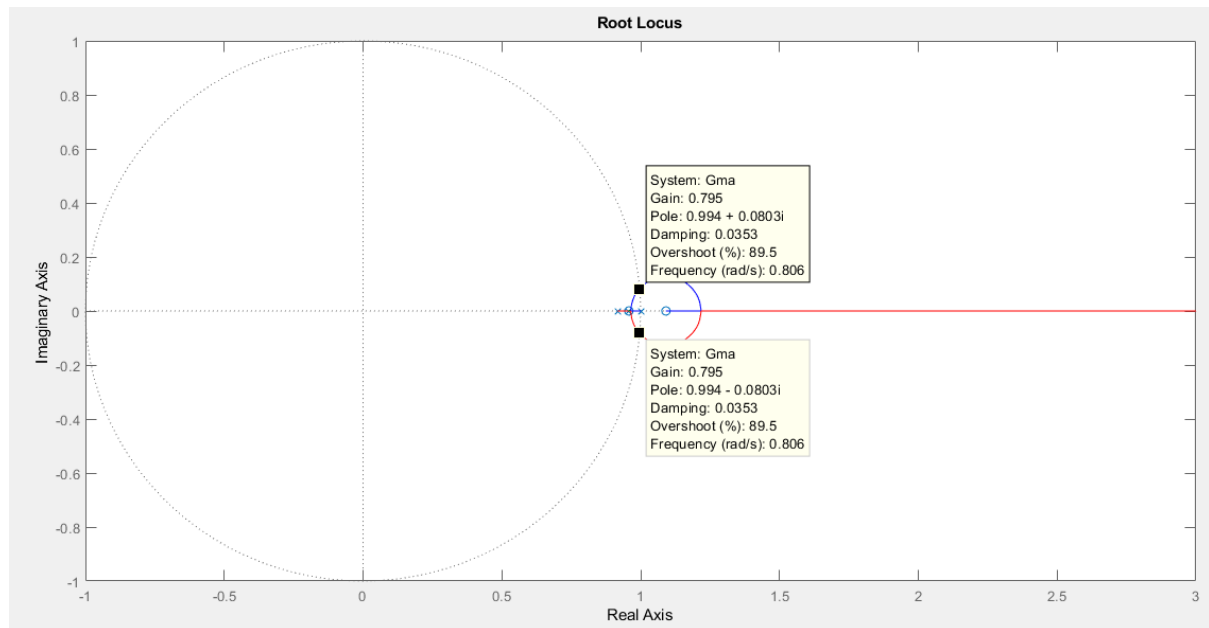


Figura 7 - Lugar das raízes para controlador PI discreto com $a = 0.9575$.

O que nos mostra que com a escolha de a , agora o ganho K tem de ser menor que 0.795 para que o sistema seja estável.

Do requisito de erro a rampa menor que 1, tínhamos que $K \geq 1.023$, porém, para atender o requisito de estabilidade, $K \leq 0.795$. Logo, para $a = 0.9575$, não há interseção dos valores de K que atendam os dois requisitos simultaneamente do controlador.

Fazendo essa mesma análise, agora para $a = 0.9167$, temos o seguinte RL:

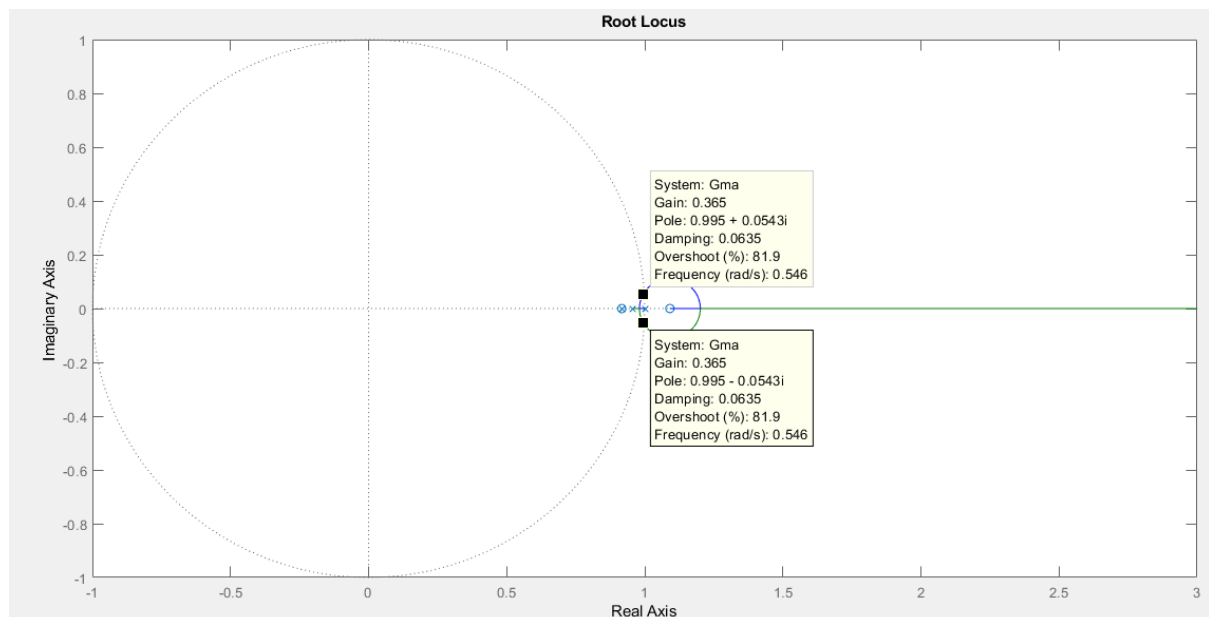


Figura 8 - Lugar das raízes para controlador PI discreto com $a = 0.9167$.

Então, para $a = 0.9167$, $K \leq 0.366$, para que atenda o requisito de estabilidade.

Porém, do requisito de erro a rampa menor que 1, tínhamos que $K \geq 0.5219$. Assim, de forma similar, concluímos que não há interseção nos valores de K que atendam ambos requisitos de erro em regime para a resposta rampa menor que 1 e estabilidade.

Dessa forma, como não existe intervalo de K que o controlador atende todos os requisitos do projeto, concluímos que o **controlador não é possível**.

Mesmo o controlador não sendo possível, analisar a resposta ao degrau variando o valor de K e encontrar qual resulta na resposta mais rápida e com menor sobressinal, os dois primeiros requisitos do controlador, deixando de lado o terceiro requisito, de erro a rampa menor que 1.

Fazendo primeiro para $a = 0.9575$, temos a resposta ao degrau para diferentes valores de K no gráfico abaixo.

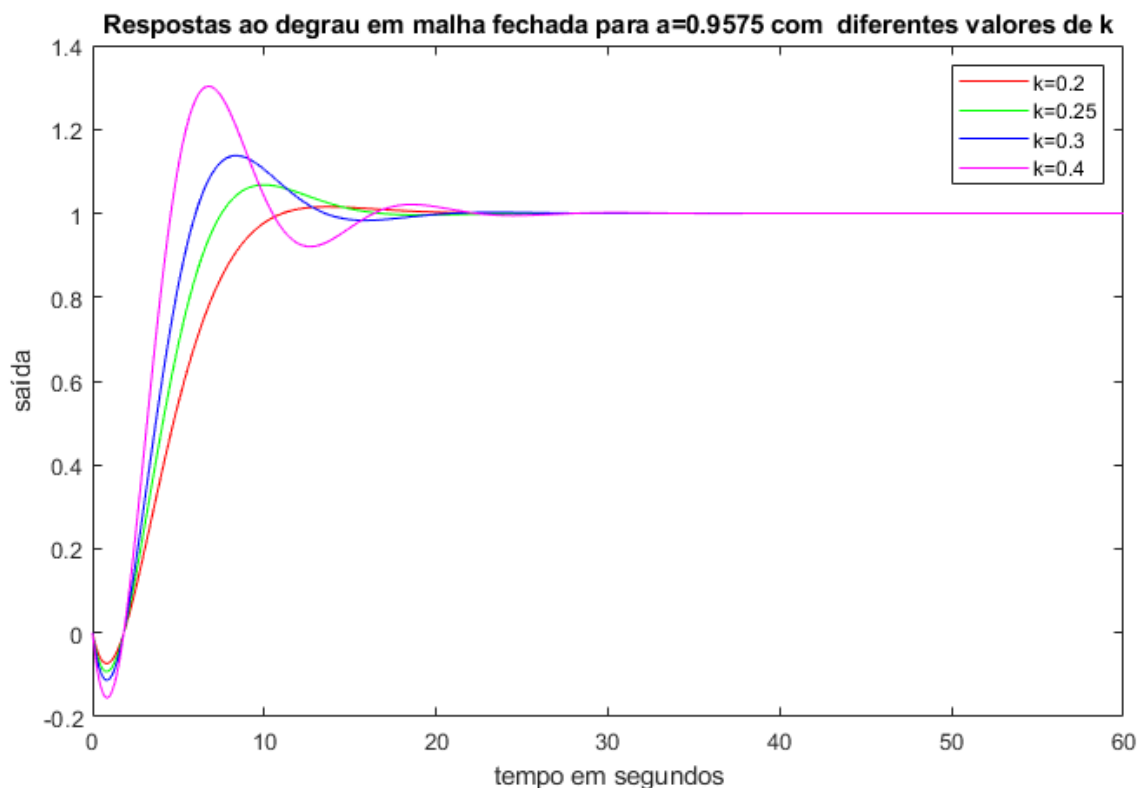


Figura 9 - Resposta ao degrau para diferentes valores de K e $a = 0.9575$.

Sendo assim, teríamos $K = 0.2$ e $a = 0.9575$ como resposta mais rápida e ao mesmo tempo com menor sobressinal.

Agora, com $a = 0.9167$, temos a resposta ao degrau de alguns valores de K abaixo:

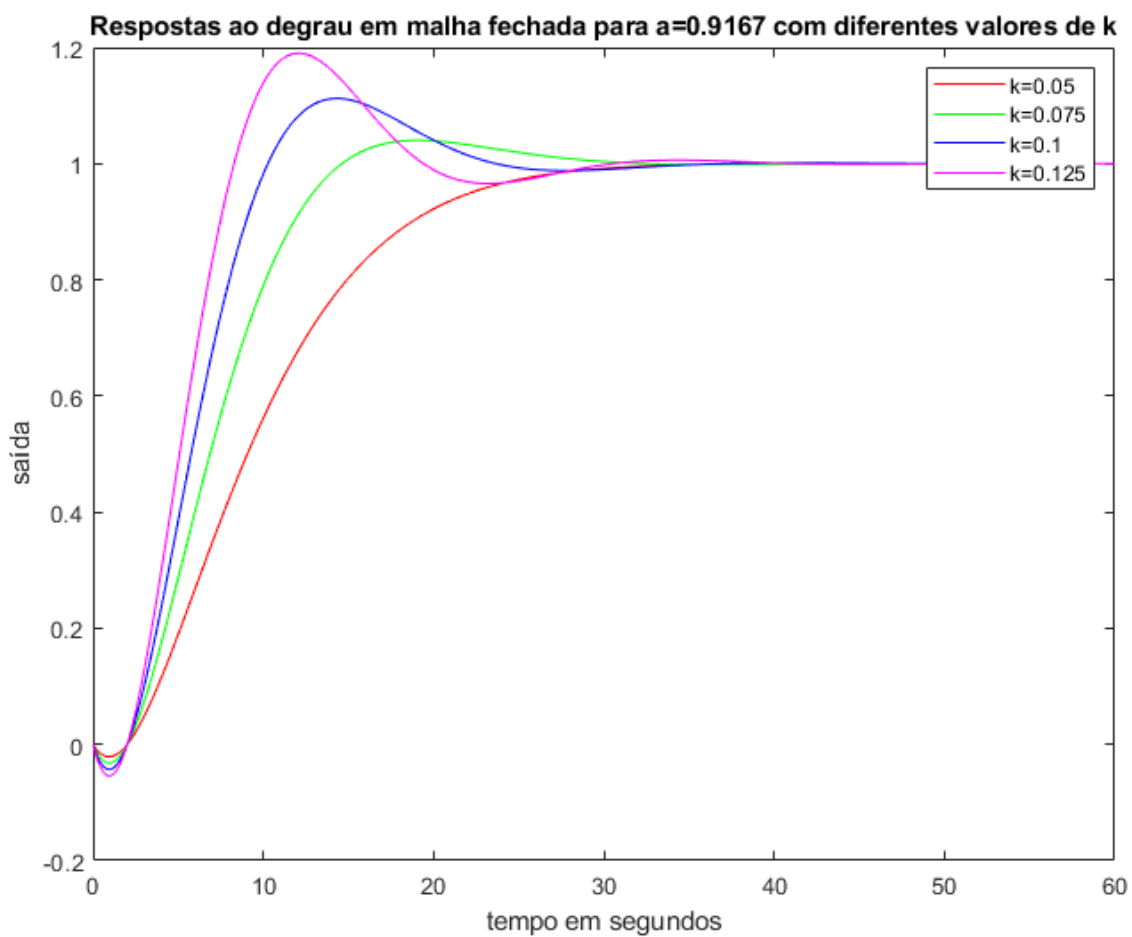


Figura 10 - Resposta ao degrau para diferentes valores de K e $a = 0.9167$.

Analisando as respostas ao degrau, vemos que para $a = 0.9167$, $K = 0.1$ corresponde a resposta mais rápida, com sobressinal de 11,2%.

Questão 2

```
% Trabalho Computacional I - Sistemas Realimentados
% Arthur Sorrentino Ferreira e Higor David Oliveira
% N = 23

% Especificações:
% - Resposta ao degrau rápida com menor sobressinal possível
% - Erro em regime à rampa menor que 1
% - Período de amostragem igual a 0,1 seg

clc, clear all

% ----- Variáveis principais -----
syms s; N = 23; % número da dupla
% Numerador e Denominador da Gp, considerando a aproximação de Padé:
num = {N*(1-(0.1*N/2)*s)}; den = {(N*s + 10)*(1+(0.1*N/2)*s)};
num = sym2poly(num{1});
den = sym2poly(den{1});

% ----- Variáveis secundárias -----
ts = 0.1; % tempo de amostragem
t=0:ts:60;

% ----- Definições -----
gp = tf(num, den) % G processo
```

```
gp =

      -26.45 s + 23
      -----
      26.45 s^2 + 34.5 s + 10

Continuous-time transfer function.
```

```
gp_disc = c2d(gp,ts) % contínuo para discreto
```

```
gp_disc =

      -0.08953 z + 0.09768
      -----
      z^2 - 1.874 z + 0.8777

Sample time: 0.1 seconds
Discrete-time transfer function.
```

```
zpk(gp_disc) % Pólos e Zeros da Gp em evidência
```

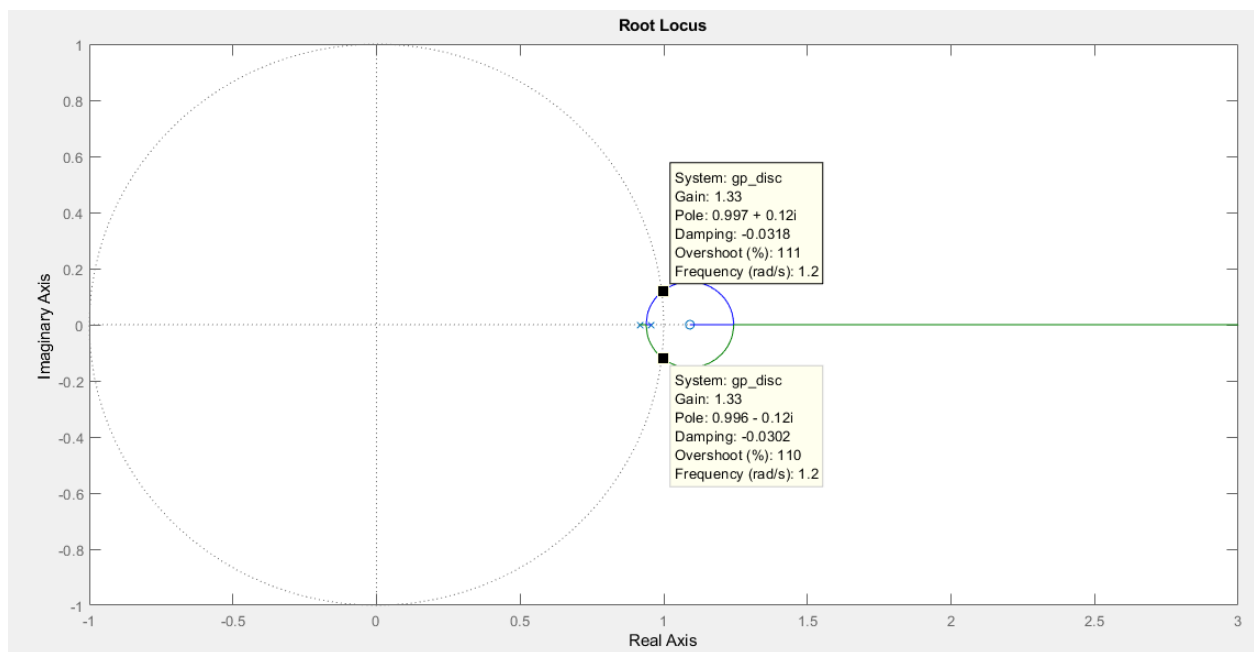
```
ans =

      -0.08953 (z-1.091)
      -----
      (z-0.9575) (z-0.9167)

Sample time: 0.1 seconds
Discrete-time zero/pole/gain model.
```

```
%rlocus(gp_disc)
%rltool(gp_disc)
```

% Observando o LGR: para a resposta ser estável, K deve ser menor que 1.33



% Controlador Digital PI

```
syms a; syms k; syms z;
D = k*((z-a)/(z-1)) % Formato simbólico
```

D =

$$-\frac{k(a-z)}{z-1}$$

% Cálculo do Erro em Regime:

```
R = (ts*z)/((z-1)^2) % Entrada
```

R =

$$\frac{z}{10(z-1)^2}$$

```
G = (poly2sym(gp_disc.Numerator{1},z)/poly2sym(gp_disc.Denominator{1},z)) * (D)
```

G =

$$\frac{k \left(\frac{6451335154741075}{72057594037927936} z - \frac{219956055047351}{2251799813685248} \right) (a-z)}{(z-1) \left(z^2 - \frac{16485376475007}{8796093022208} z + \frac{3952871242147603}{4503599627370496} \right)}$$

```
erro = R/(1+G)
```

erro =

$$10 \frac{\frac{z}{k \left(\frac{6451335154741075 z - 219956055047351}{72057594037927936} - \frac{219956055047351}{2251799813685248} \right) (a - z)}{\left((z - 1) \left(z^2 - \frac{16485376475007 z}{8796093022208} + \frac{3952871242147603}{4503599627370496} \right) + 1 \right)} (z - 1)^2}$$

```
erro_ss = (1-z^(-1))*erro
```

```
erro_ss =
```

$$-\frac{z \left(\frac{1}{z} - 1 \right)}{10 \frac{\frac{z}{k \left(\frac{6451335154741075 z - 219956055047351}{72057594037927936} - \frac{219956055047351}{2251799813685248} \right) (a - z)}{\left((z - 1) \left(z^2 - \frac{16485376475007 z}{8796093022208} + \frac{3952871242147603}{4503599627370496} \right) + 1 \right)} (z - 1)^2}}$$

```
erro_ss_lim = limit(erro_ss, z, 1)
```

```
erro_ss_lim =
```

$$-\frac{1}{10 k \left(\frac{587258606774157 a - 587258606774157}{255329829032240} - \frac{587258606774157}{255329829032240} \right)}$$

```
e_ss = 1; % erro estacionário deve ser menor que 1
erro_ss_lim1 = subs(erro_ss_lim, a, 0.9575)
```

```
erro_ss_lim1 =
```

$$\frac{10213193161289600}{9983396315160669 k}$$

```
k_1 = subs(erro_ss_lim1,k,e_ss)
```

```
k_1 =
```

$$\frac{10213193161289600}{9983396315160669}$$

```
k_1 = double(k_1)
```

```
k_1 = 1.0230
```

```
% Para a=0.9575, K deve ser maior que 1.023 para que o erro seja menor que 1
erro_ss_lim2 = subs(erro_ss_lim, a, 0.9167)
```

```
erro_ss_lim2 =
```

$$\frac{255329829032240000}{489186419442872781 k}$$

```
k_2 = subs(erro_ss_lim2,k,e_ss)
```

```
k_2 =
```

$$\frac{255329829032240000}{489186419442872781}$$

```
k_2 = double(k_2)
```

```
k_2 = 0.5219
```

% Para $a=0.9167$, K deve ser maior que 0.5219 para que o erro seja menor que 1

```
% Para a=0.9575
```

```
a = 0.9575;
```

```
g_controlador = tf([1 -a],[1 -1],ts)
```

```
g_controlador =
```

```
z - 0.9575
```

```
-----
```

```
z - 1
```

```
Sample time: 0.1 seconds
```

```
Discrete-time transfer function.
```

```
Gma = gp_disc * g_controlador
```

```
Gma =
```

```
-0.08953 z^2 + 0.1834 z - 0.09353
```

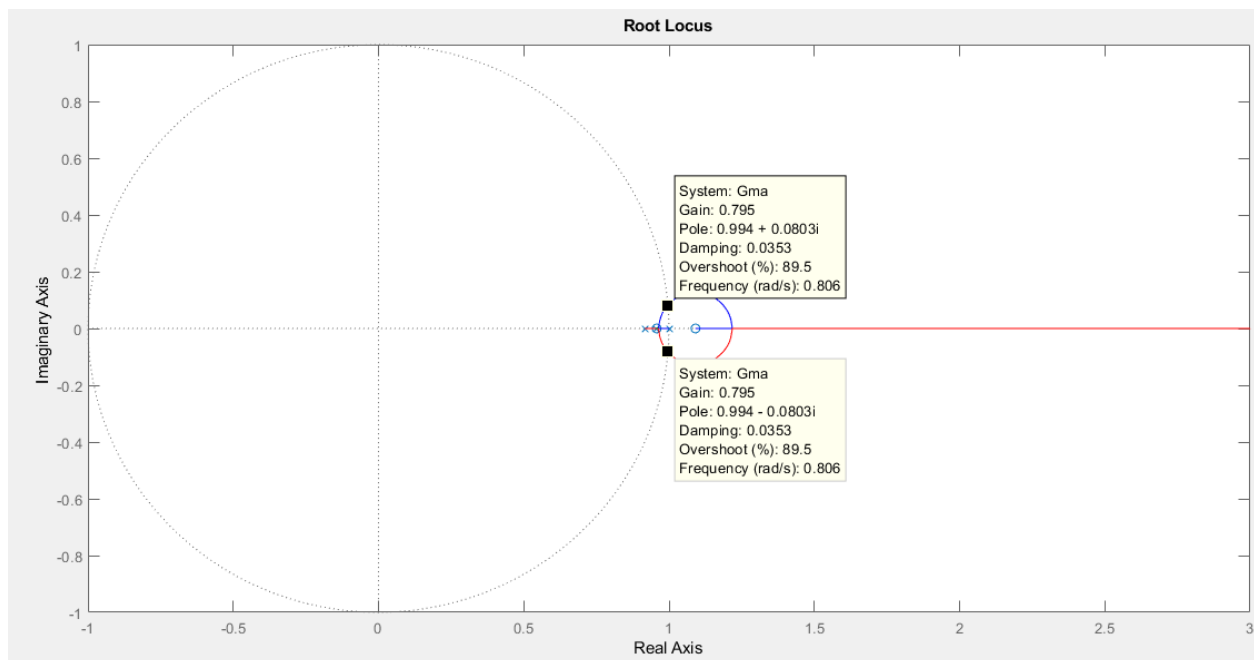
```
-----
```

```
z^3 - 2.874 z^2 + 2.752 z - 0.8777
```

```
Sample time: 0.1 seconds
```

```
Discrete-time transfer function.
```

% Resposta estável para $K \leq 0.795$



```
K = 0.2 ;
```

```
Gmf1 = feedback(K*Gma, 1);
```

```
y1 = step(Gmf1,t);
```

```
stepinfo(y1,t)
```

```
ans = struct with fields:
    RiseTime: 5.7459
    TransientTime: 10.0434
    SettlingTime: 10.0969
    SettlingMin: 0.9041
    SettlingMax: 1.0153
    Overshoot: 1.5342
    Undershoot: 7.3056
    Peak: 1.0153
    PeakTime: 13.7000
```

```
K = 0.25;
Gmf2 = feedback(K*Gma, 1);
y2 = step(Gmf2,t);
stepinfo(y2,t)
```

```
ans = struct with fields:
    RiseTime: 4.0517
    TransientTime: 13.9360
    SettlingTime: 14.0909
    SettlingMin: 0.9109
    SettlingMax: 1.0676
    Overshoot: 6.7625
    Undershoot: 9.2657
    Peak: 1.0676
    PeakTime: 10
```

```
K = 0.3;
Gmf3 = feedback(K*Gma, 1);
y3 = step(Gmf3,t);
stepinfo(y3,t)
```

```
ans = struct with fields:
    RiseTime: 3.1078
    TransientTime: 12.4793
    SettlingTime: 12.5673
    SettlingMin: 0.9059
    SettlingMax: 1.1373
    Overshoot: 13.7339
    Undershoot: 11.2824
    Peak: 1.1373
    PeakTime: 8.4000
```

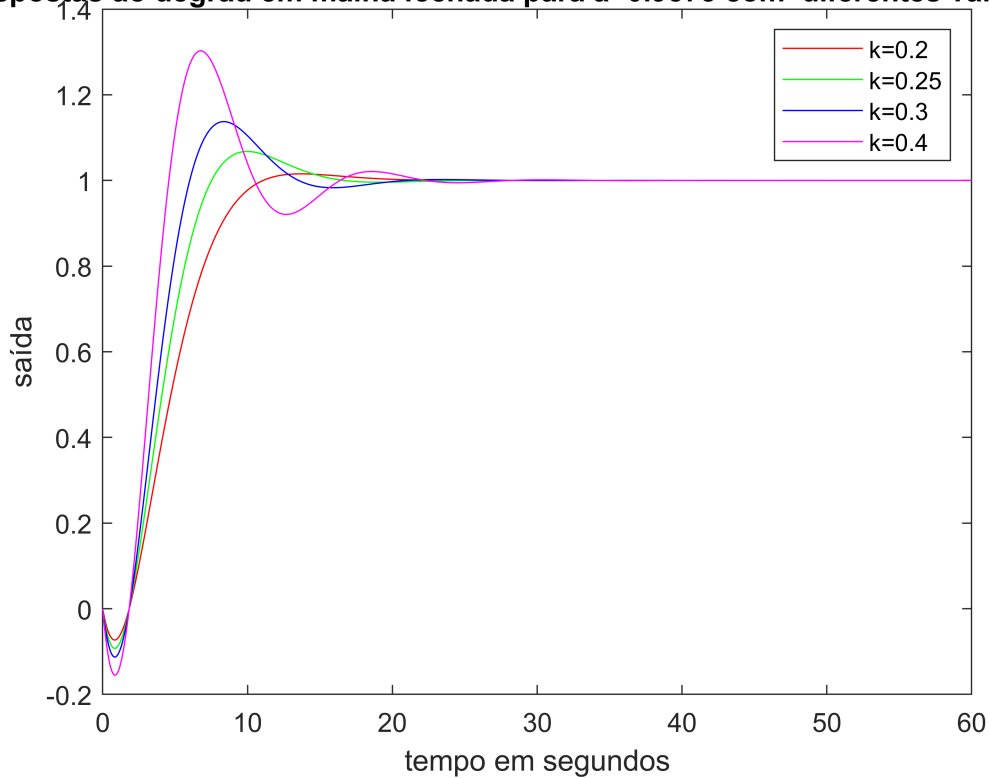
```
K = 0.4;
Gmf4 = feedback(K*Gma, 1);
y4 = step(Gmf4,t);
stepinfo(y4,t)
```

```
ans = struct with fields:
    RiseTime: 2.0993
    TransientTime: 15.4069
    SettlingTime: 19.0872
    SettlingMin: 0.9196
    SettlingMax: 1.3026
    Overshoot: 30.2631
    Undershoot: 15.4920
    Peak: 1.3026
    PeakTime: 6.8000
```

```
plot(t,y1,'r',t,y2,'g',t,y3,'b', t, y4,'m')
legend('k=0.2', 'k=0.25', 'k=0.3', 'k=0.4');
```

```
title('Respostas ao degrau em malha fechada para a = 0.9575 com diferentes valores de k');
xlabel('tempo em segundos');
ylabel('saída');
```

Respostas ao degrau em malha fechada para a=0.9575 com diferentes valores d



```
% Para a=0.9167
a = 0.9167;
g_controlador = tf([1 -a],[1 -1],ts)
```

```
g_controlador =
```

$$\frac{z - 0.9167}{z - 1}$$

```
Sample time: 0.1 seconds
Discrete-time transfer function.
```

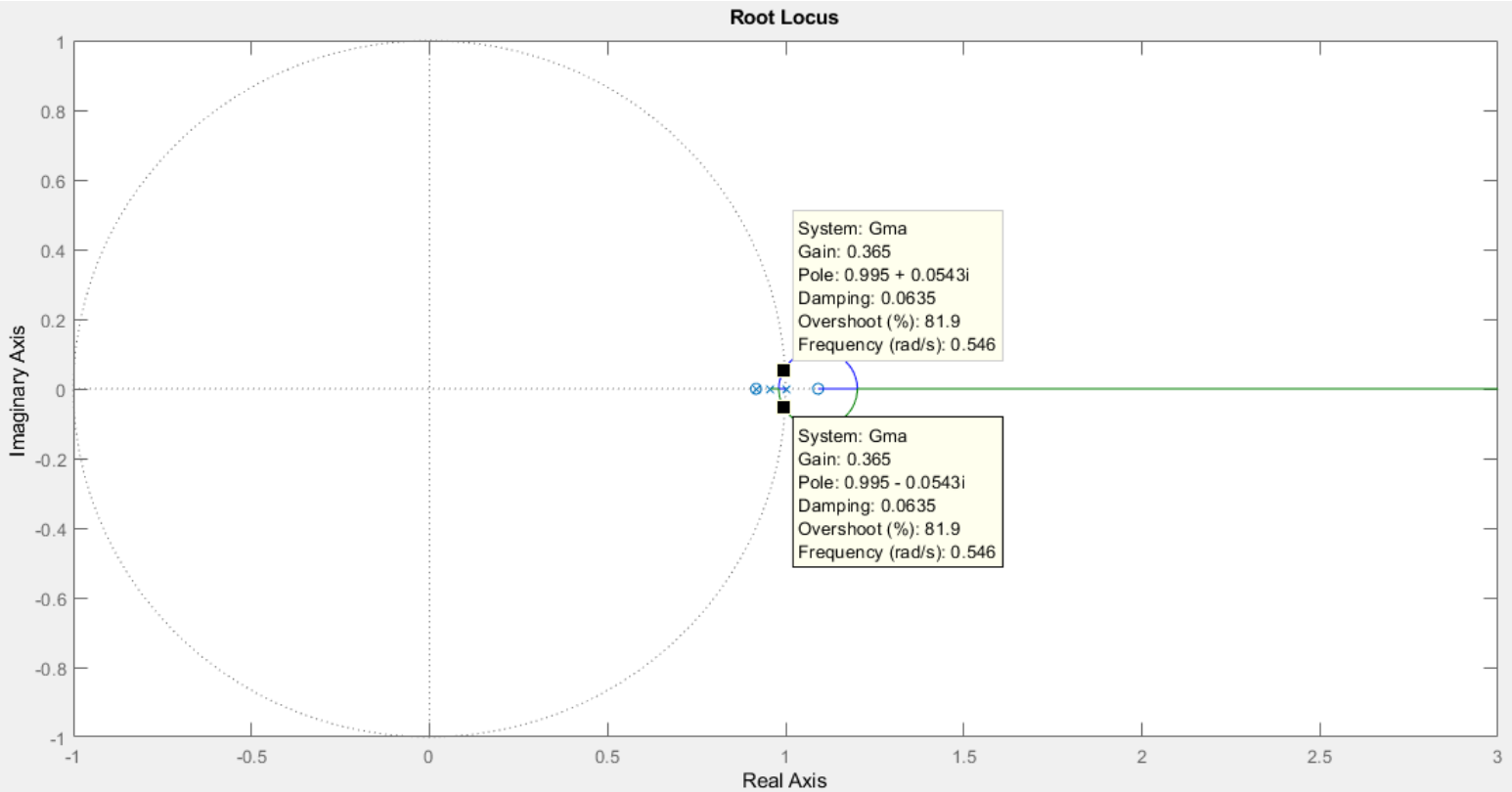
```
Gma = gp_disc * g_controlador
```

```
Gma =
```

$$\frac{-0.08953 z^2 + 0.1798 z - 0.08954}{z^3 - 2.874 z^2 + 2.752 z - 0.8777}$$

```
Sample time: 0.1 seconds
Discrete-time transfer function.
```

```
% Resposta estável para K<=0.365
```



```
K = 0.05;
Gmf5 = feedback(K*Gma, 1);
y5 = step(Gmf5,t);
stepinfo(y5,t)
```

```
ans = struct with fields:
    RiseTime: 14.9190
    TransientTime: 26.2913
    SettlingTime: 26.3820
    SettlingMin: 0.9015
    SettlingMax: 1.0002
    Overshoot: 0.0121
    Undershoot: 2.1114
    Peak: 1.0002
    PeakTime: 46.7000
```

```
K = 0.075;
Gmf6 = feedback(K*Gma, 1);
y6 = step(Gmf6,t);
stepinfo(y6,t)
```

```
ans = struct with fields:
    RiseTime: 8.5067
    TransientTime: 24.6952
    SettlingTime: 24.8428
    SettlingMin: 0.9004
    SettlingMax: 1.0402
    Overshoot: 4.0148
    Undershoot: 3.1933
    Peak: 1.0402
    PeakTime: 19.1000
```

```
K = 0.1;
Gmf7 = feedback(K*Gma, 1);
y7 = step(Gmf7,t);
stepinfo(y7,t)
```

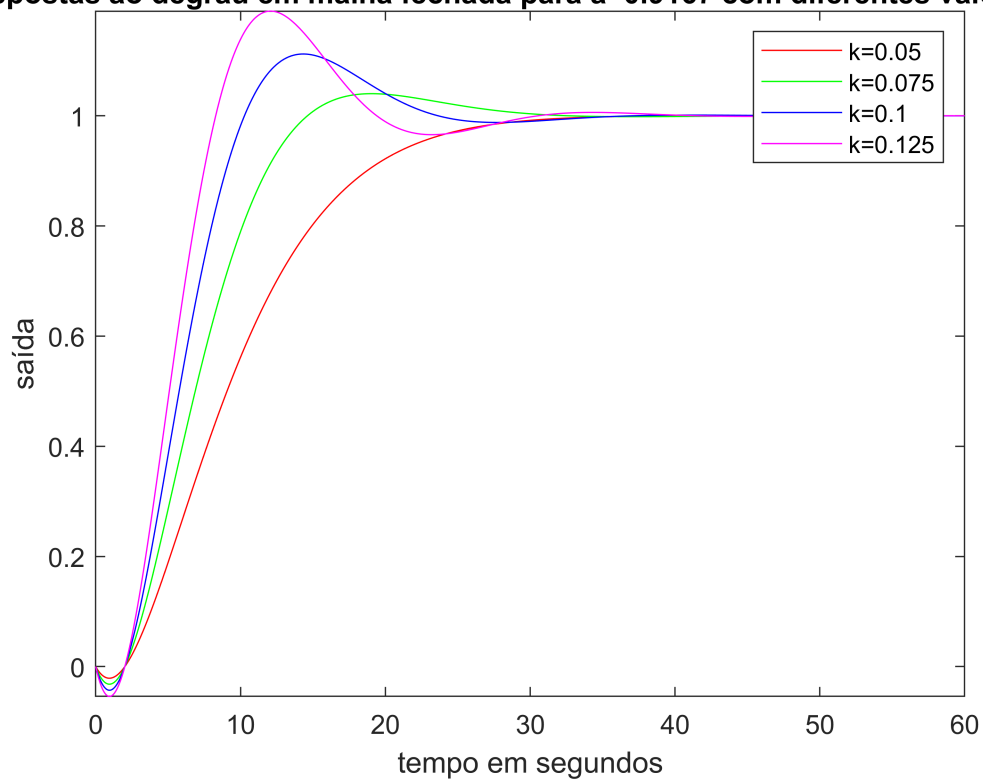
```
ans = struct with fields:
    RiseTime: 5.9757
    TransientTime: 21.4448
    SettlingTime: 21.5164
    SettlingMin: 0.9019
    SettlingMax: 1.1122
    Overshoot: 11.2274
    Undershoot: 4.2936
    Peak: 1.1122
    PeakTime: 14.3000
```

```
K = 0.125;
Gmf8 = feedback(K*Gma, 1);
y8 = step(Gmf8,t);
stepinfo(y8,t)
```

```
ans = struct with fields:
    RiseTime: 4.6496
    TransientTime: 26.5594
    SettlingTime: 26.7368
    SettlingMin: 0.9040
    SettlingMax: 1.1901
    Overshoot: 18.9937
    Undershoot: 5.4137
    Peak: 1.1901
    PeakTime: 12.1000
```

```
plot(t,y5,'r',t,y6,'g',t,y7,'b', t, y8,'m')
legend('k=0.05','k=0.075', 'k=0.1', 'k=0.125');
title('Respostas ao degrau em malha fechada para a=0.9167 com diferentes valores de k');
xlabel('tempo em segundos');
ylabel('saída');
```

Respostas ao degrau em malha fechada para $a=0.9167$ com diferentes valores de k



Questão 3

```
% Trabalho Computacional I - Sistemas Realimentados
% Arthur Sorrentino Ferreira e Higor David Oliveira
% N = 23

% Especificações:
% - Sobressinal à entrada degrau menor que 20%
% - Erro em regime à entrada rampa menor que 1 com o menor tempo
% de subida possível.
```

```
clc, clear all
```

```
% ----- Variáveis principais -----
```

```
syms s; N = 23; % número da dupla
% Numerador e Denominador da Gp:
num = 10*N ; den = {(s+(N/sqrt(N)))*(s+sqrt(N))*(s+N)};
den = sym2poly(den{1});
```

```
% ----- Variáveis secundárias -----
```

```
ts = 0.1; % tempo de amostragem
t=0:ts:80;
```

```
% ----- Definições -----
```

```
gp = tf(num, den) % G processo
```

gp =

$$\frac{230}{s^3 + 32.59 s^2 + 243.6 s + 529}$$

Continuous-time transfer function.

```
% Projeto do PI. Obter os parâmetros Kii e Kpi
```

```
syms s; syms Kpi; syms Kii;
gp = poly2sym(gp.Numerator{1},s)/poly2sym(gp.Denominator{1},s);
Kv = limit((s*(Kpi+Kii/s))*gp,s,0)
```

Kv =

$$\frac{10 K_{ii}}{23}$$

```
ess = 1/Kv % deve ser menor que 1, logo Kii deve ser maior ou igual a 2,3
```

ess =

$$\frac{23}{10 K_{ii}}$$

```
kii = 2.3;
kpi = 0.25
```



```
kpi = 0.2500
```

```
z= kii/kpi;  
% = kii/kpi; obtido usando o LR para obter a resposta mais rápida e estável  
num_pi=[1 z];  
den_pi=[1 0];  
gpi=tf(num_pi,den_pi)
```

```
gpi =
```

```
      s + 9.2  
-----  
      s
```

Continuous-time transfer function.

```
gp = tf(num, den) % retorna gp ao formato numérico
```

```
gp =
```

```
          230  
-----  
s^3 + 32.59 s^2 + 243.6 s + 529
```

Continuous-time transfer function.

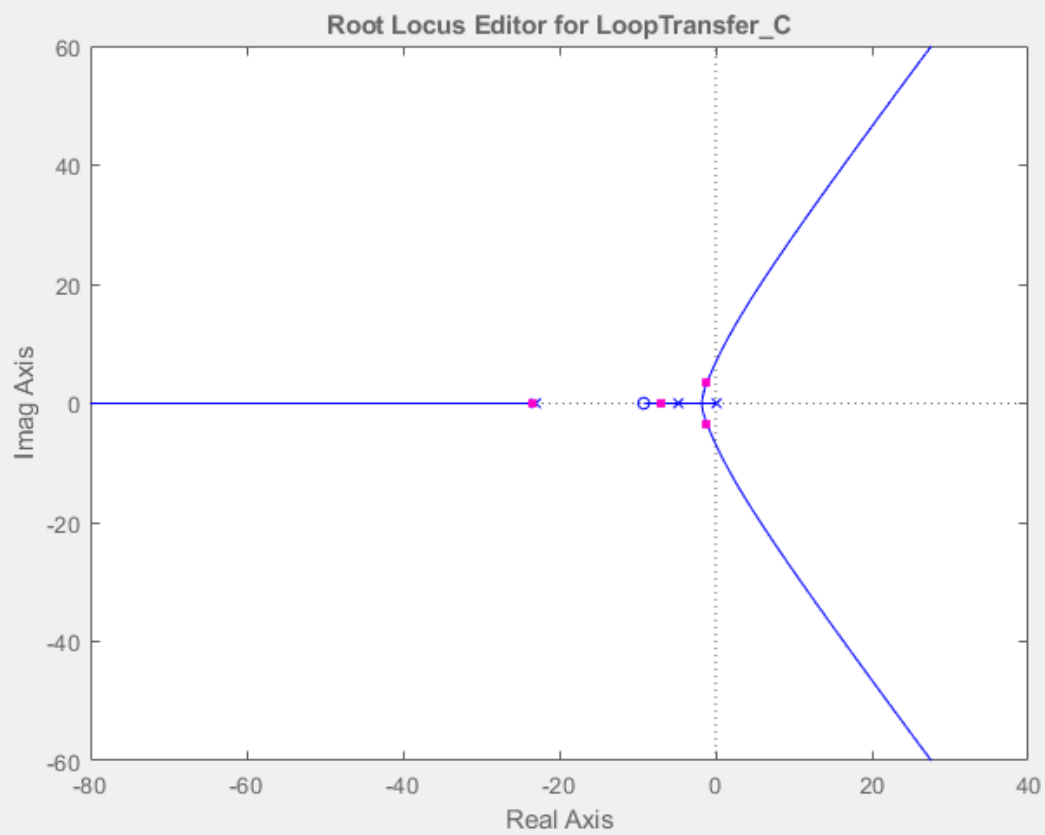
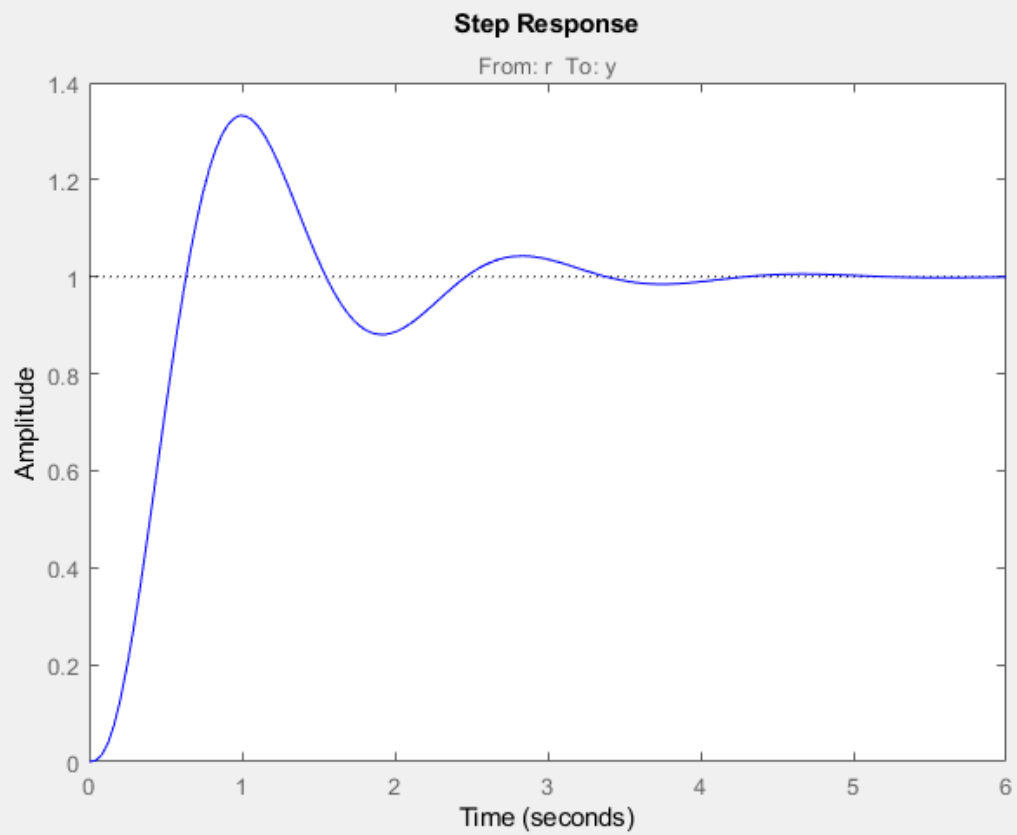
```
gma=gpi*gp
```

```
gma =
```

```
      230 s + 2116  
-----  
s^4 + 32.59 s^3 + 243.6 s^2 + 529 s
```

Continuous-time transfer function.

```
rltool(gma)
```



```
% Simulando o PI com a planta em malha fechada
```

```
gma=kpi*gma;  
gmf1=feedback(gma,1);  
t=0:0.01:3;  
y1=step(gmf1,t);
```

```
% Projeto do PD. Obter o parâmetro Kd
```

```
gp = poly2sym(gp.Numerator{1},s)/poly2sym(gp.Denominator{1},s);  
gp_ = (kpi*(1+z*s)*gp)/s
```

```
gp_ =
```

$$\frac{230 \left(\frac{23s}{10} + \frac{1}{4} \right)}{s \left(s^3 + \frac{2293434399252611 s^2}{70368744177664} + \frac{267850103576751 s}{1099511627776} + 529 \right)}$$

```
gma = (s*gp_)/(1+gp)
```

```
gma =
```

$$\frac{230 \left(\frac{23s}{10} + \frac{1}{4} \right)}{\left(\frac{230}{s^3 + \frac{2293434399252611 s^2}{70368744177664} + \frac{267850103576751 s}{1099511627776} + 529} + 1 \right) \left(s^3 + \frac{2293434399252611 s^2}{70368744177664} + \frac{267850103576751 s}{1099511627776} + 529 \right)}$$

```
num_gma = simplify(s*gp_);  
den_gma = simplify(1+gp);  
gma = num_gma/den_gma
```

```
gma =
```

$$\frac{230 \left(\frac{23s}{10} + \frac{1}{4} \right)}{\left(\frac{230}{s^3 + \frac{2293434399252611 s^2}{70368744177664} + \frac{267850103576751 s}{1099511627776} + 529} + 1 \right) \left(s^3 + \frac{2293434399252611 s^2}{70368744177664} + \frac{267850103576751 s}{1099511627776} + 529 \right)}$$

```
gma = simplify(gma)
```

```
gma =
```

$$\frac{37225065669984256 s + 4046202790215680}{70368744177664 s^3 + 2293434399252611 s^2 + 17142406628912064 s + 53409876830846976}$$

```
num_gma = [37225065669984256 4046202790215680];  
den_gma = [70368744177664 2293434399252611 17142406628912064 53409876830846976];  
gma = tf(num_gma, den_gma)
```

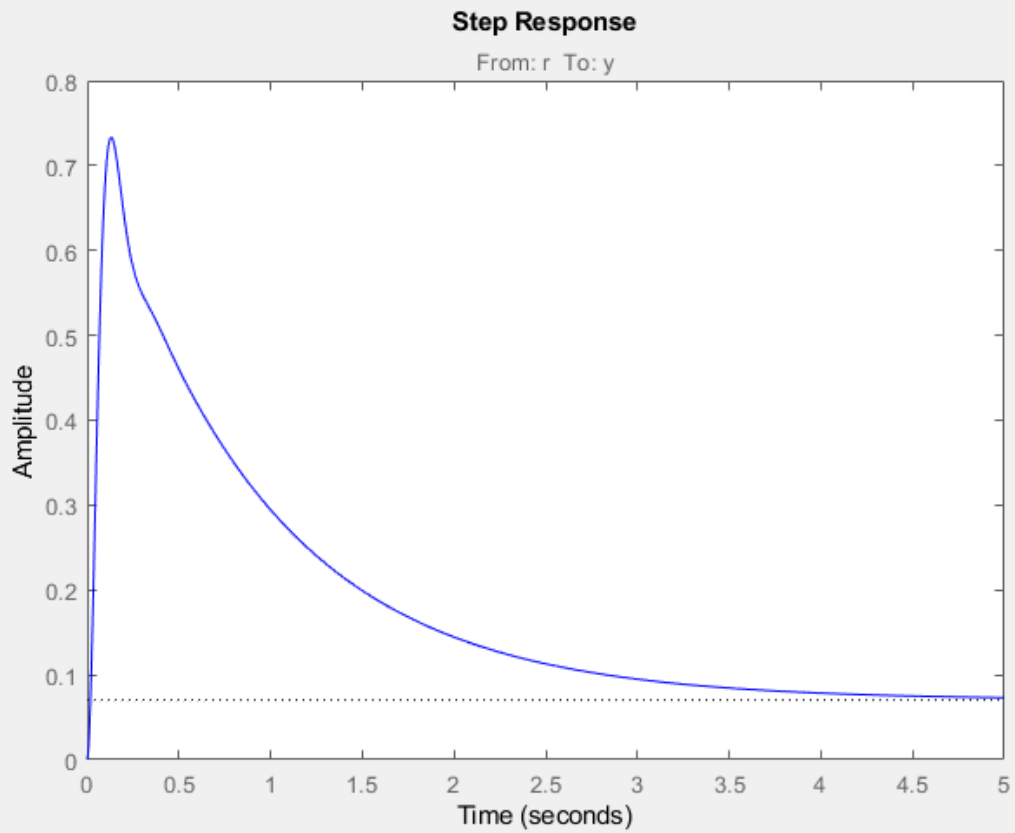
```
gma =
```

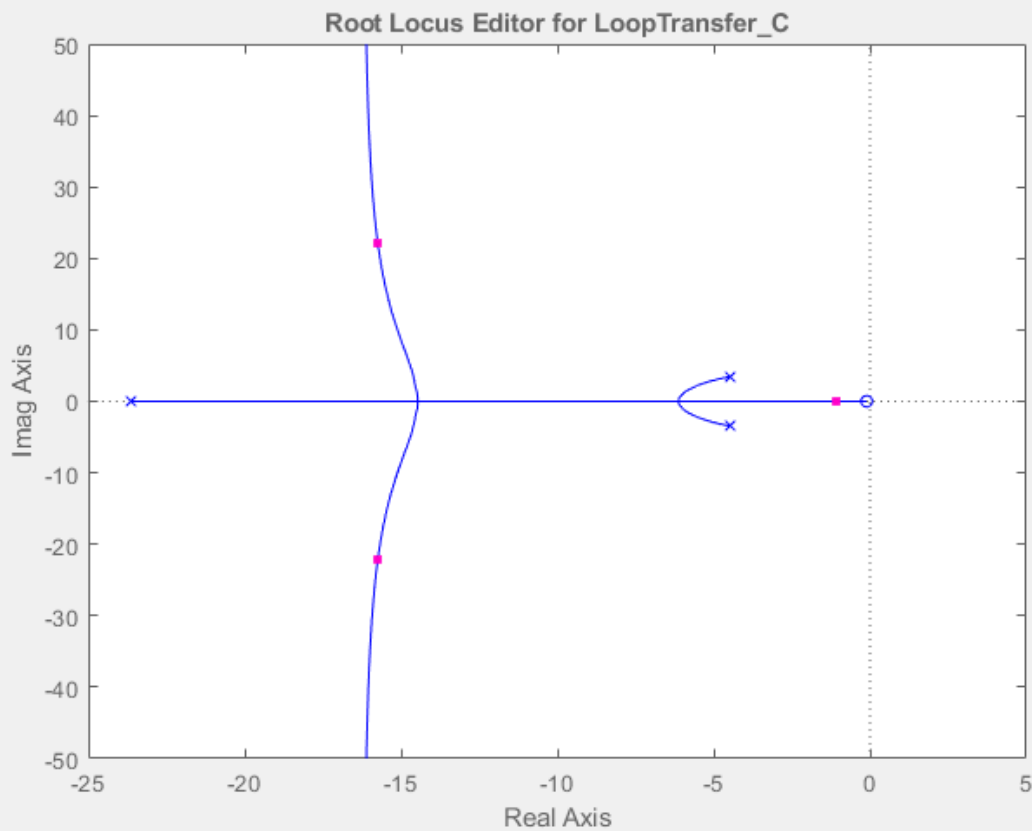
$$3.723e16 s + 4.046e15$$

$$7.037e13 s^3 + 2.293e15 s^2 + 1.714e16 s + 5.341e16$$

Continuous-time transfer function.

```
rltool(gma)
```





```
%rlocus(gma)
% Por meio do rltool e rlocus, escolhe-se Kd = 1.8 para atender sobressinal<20%

% Simulando o controlador PID em malha fechada com a planta
kd1 = 1.8;
num_pd=[kd1 1];
num_pid=conv(num_pi,num_pd);
den_pid=den_pi;
gc=tf(num_pid,den_pid);
gp = tf(num, den) % retorna gp ao formato numérico
```

gp =

$$\frac{230}{s^3 + 32.59 s^2 + 243.6 s + 529}$$

Continuous-time transfer function.

```
gg=gc*gp;
gmf2=feedback(gg,1);
t=0:0.01:3;
y2=step(gmf2,t);

kd1 = 1.4;
num_pd=[kd1 1];
num_pid=conv(num_pi,num_pd);
den_pid=den_pi;
```

```

gc=tf(num_pid,den_pid);
gg=gc*gp;
gmf3=feedback(gg,1);
t=0:0.01:3;
y3=step(gmf3,t);
plot(t,y1,'r',t,y2,'g',t,y3,'b');
legend('Resposta com PI', 'Resposta com PID para kd1=1.9', 'Resposta com PID para kd1=1.5');
ylabel('Resposta à entrada degrau');
xlabel('tempo')

```

