Trabalho 5 - Projeto de controladores PID no domínio da frequência

Nome: Guilherme Goes Zanetti

Data limite para entrega: 24/11, 6h

Importante lembrar:

- Entrega após a data/horário acima: a nota será multiplicada por $1-e^{-30/h}$, onde h são as horas em atraso (Exemplo: 24h, multiplica por 0.71).
- O trabalho não é recebido por email
- Cabe a vocês garantir que o documento entregue é um arquivo pdf legível, e que não foi entregue com erro. Para isto, basta depositar e abrir para conferir.
- Código é apenas uma informação complementar, e não é considerada parte da solução para fins de avaliação.
- Caso não haja tempo de fazer todo o trabalho, entregue no prazo o que estiver pronto.

```
I=16; % Seu valor de I
[G, ts0, G1]=ini_t5(I)
```

```
G =
 
          27
  -------------------
  0.2 s^2 + 1.2 s + 1
 
Continuous-time transfer function.
Model Properties
ts0 = 1.5000
G1 =
 
  27
  ---
  s^2
 
Continuous-time transfer function.
Model Properties
```

```
datetime('now')
```

```
ans = datetime
   22-Nov-2023 20:31:02
```

Parte I: Projete um controlador PI C1 que atenda:

- Erro nulo em regime para entrada degrau
- Tempo de estabelecimento $\leq$ts0 segundos.

Parte II: Projete um controlador derivativo C2 de modo a obter um controlador PID=C1*C2 tal que as especificações anteriores sejam mantidas e adicionalmente se tenha

- Sobreelevação $\leq 4\%$
- Tempo de estabelecimento menor que ts0

Parte III: Estabilização: Usar critério de Nyquist para verificar se um controlador PD ou PI estabiliza G1(s)

O que apresentar:

Projeto do controlador PI (C1):

1) Gráfico de Bode de G e de C1G explicando a escolha do módulo adicionado, a localização do zero do PI, e as mudanças que o controlador C1 ocasionou nas margens de ganho e fase. Identifique a frequência de cruzamento de ganho por 0dB no gráfico de Bode com o controlador C1.

2) Resposta ao degrau em malha fechada e sinal de controle aplicado plotados no mesmo gráfico, mostrando o atendimento das especificações e explicitando os valores de Kp e Ki.

3) Resumir os passos do projeto do controlador PI (**Atenção: não são os comandos dados!**)

Projeto do controlador PD (C2):

4) Gráfico de Bode de C1G e de C1C2G identificando a localização do zero do PD, e as mudanças que o controlador C2 ocasionou na margem de fase em relação ao controlador C1G.

5) Resposta ao degrau em malha fechada mostrando o atendimento das especificações.

6) Sinal de controle devido ao controlador C1C2, explicitando o valor de Kd e o parâmetro Tf do filtro.

7) Resumir os passos do projeto do PD (**Atenção: não são os comandos dados!**)

Estabilização:

8) Verifique via critério de Nyquist se um controlador PD estabiliza G1(s): Fazer o gráfico de Nyquist e aplicar o critério

9) Verifique via critério de Nyquist se um controlador PI estabiliza G1(s): Fazer o gráfico de Nyquist e aplicar o critério

Parte I: Projeto do PI (C1)

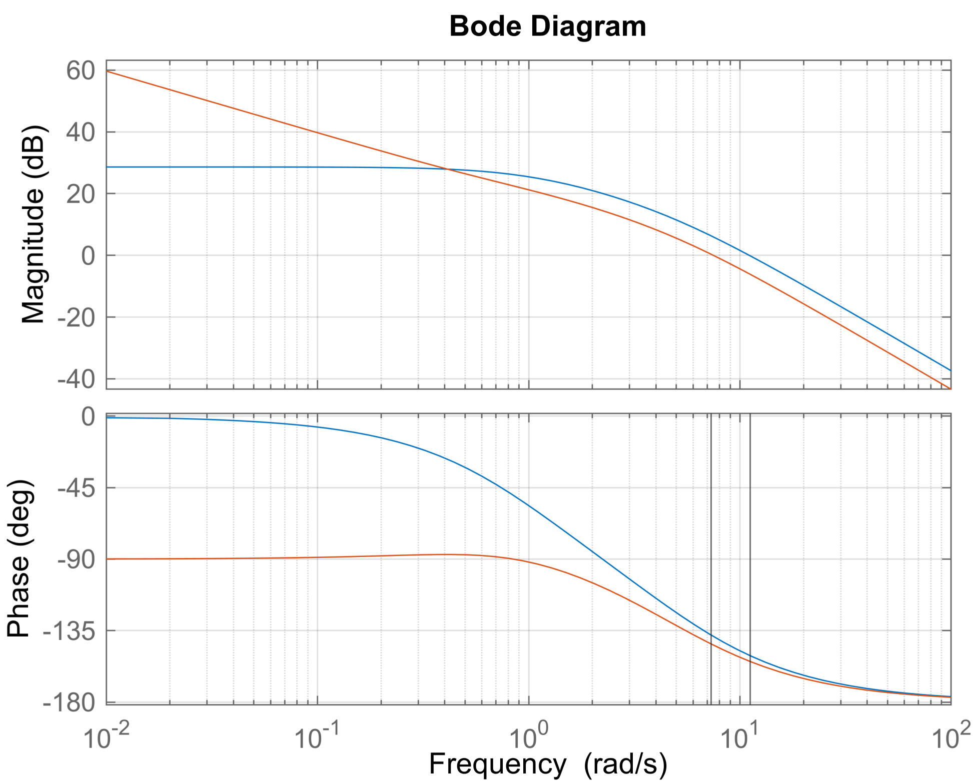
```
[C1 wg1] = projpi(G, -6)
```

```
C1 =

  0.5012 s + 0.3574
  -----------------
          s
```

```
Continuous-time transfer function.
Model Properties
wg1 = 7.1319
```

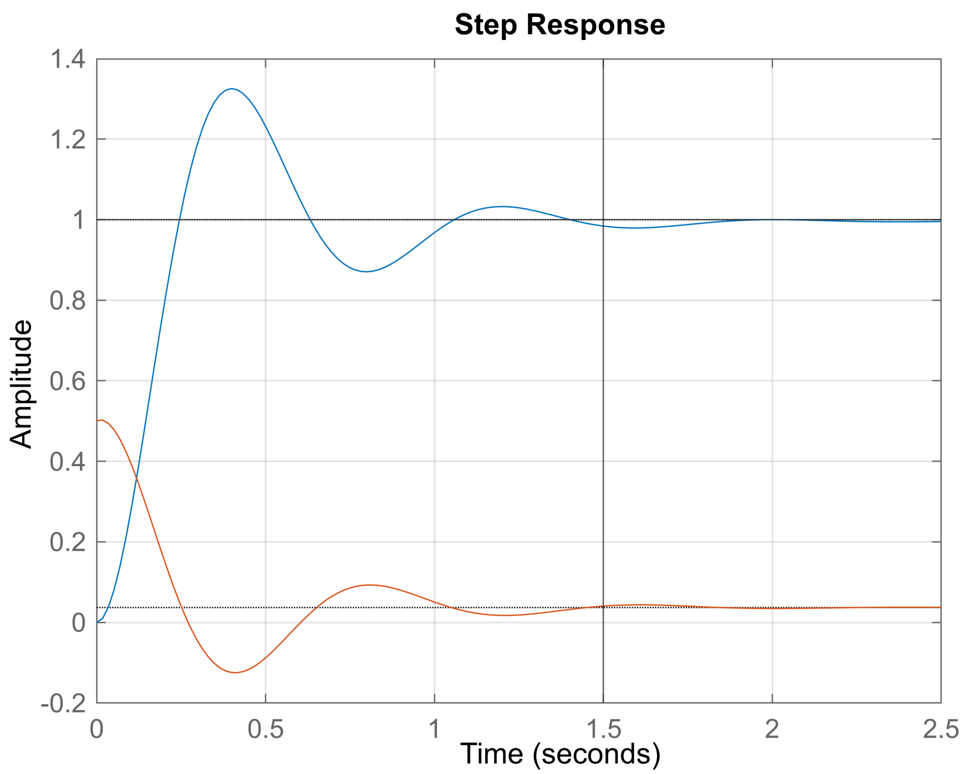
```
bode(G, C1*G); xline(11.2); xline(7.3);grid();
```



Analisando o gráfico de bode acima, onde a linha azul é G(s) e a linha laranja é C1(s)*G(s), vemos que a margem de fase original do sistema é de aproximadamente 30° e foi escolhido um controlador PI para aumentar a margem de fase para 40° e deixar o sistema menos oscilatório e cumprir o requisito de tempo de estabelecimento menor que 1,5s, que pode ser verificado na simulação de entrada ao degrau a seguir. Para realizar essa alteração na margem de fase, foi necessário diminuir o ganho em 6dB, que resulta em um Kp=0,5. Para alterar pouco a fase na frequência wg' escolhida, o zero do PI foi adicionado uma década antes de wg', ou seja, em -0,713. Portanto Ki = Kp*0,713 = 0,357. A margem de ganho em ambos os casos é infinita.

A seguir, temos o gráfico da resposta ao degrau do sistema com controlador e também o sinal de controle aplicado e analisamos o atendimento das especificações.

```
figure;
step(feedback(C1*G, 1), feedback(C1, G)); xline(ts0); yline(1); grid();
```

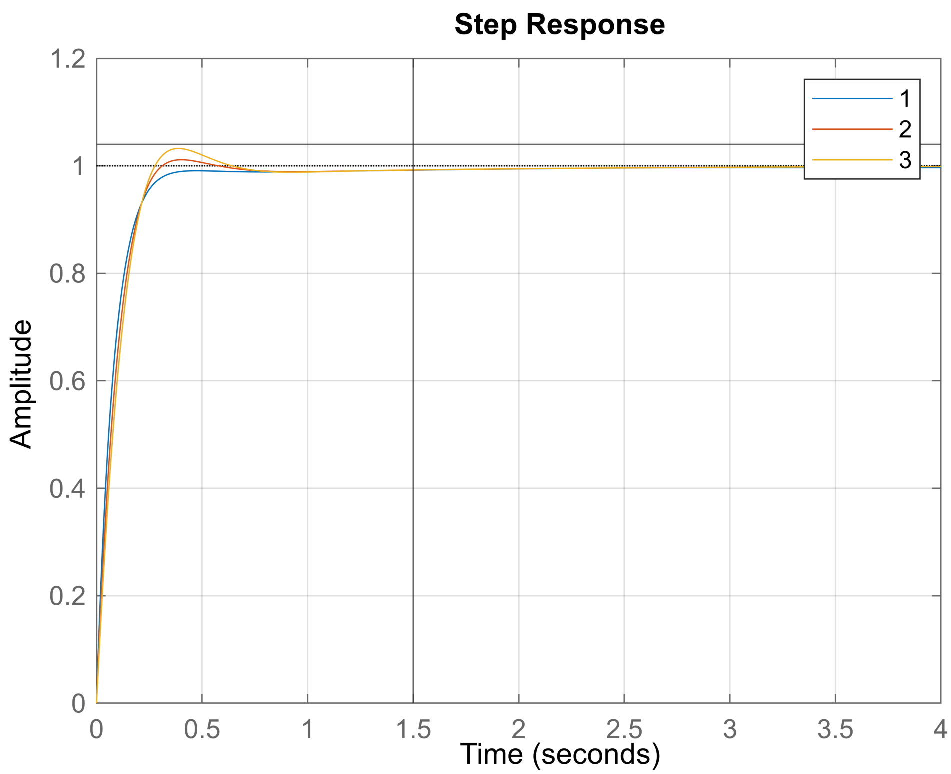


Podemos ver que o sistema com o controlador PI especificado tem tempo de estabelecimento muito próximo dos 1,5 segundos pedidos, e o erro de regime ao degrau nulo, já que o sistema se torna de tipo 1 e como pode-se ver no gráfico.

Parte II: Projeto do PD (C2)

```
figure;
C2_1 = projpd(C1*G, 0.8*wg1);
C2_2 = projpd(C1*G, wg1);
C2_3 = projpd(C1*G, 1.2*wg1);

figure;
step(feedback(C1*C2_1*G, 1), feedback(C1*C2_2*G, 1), feedback(C1*C2_3*G, 1));
xline(ts0); yline(1.04); legend('1', '2', '3'); grid();
```




```
C2 = C2_2
```

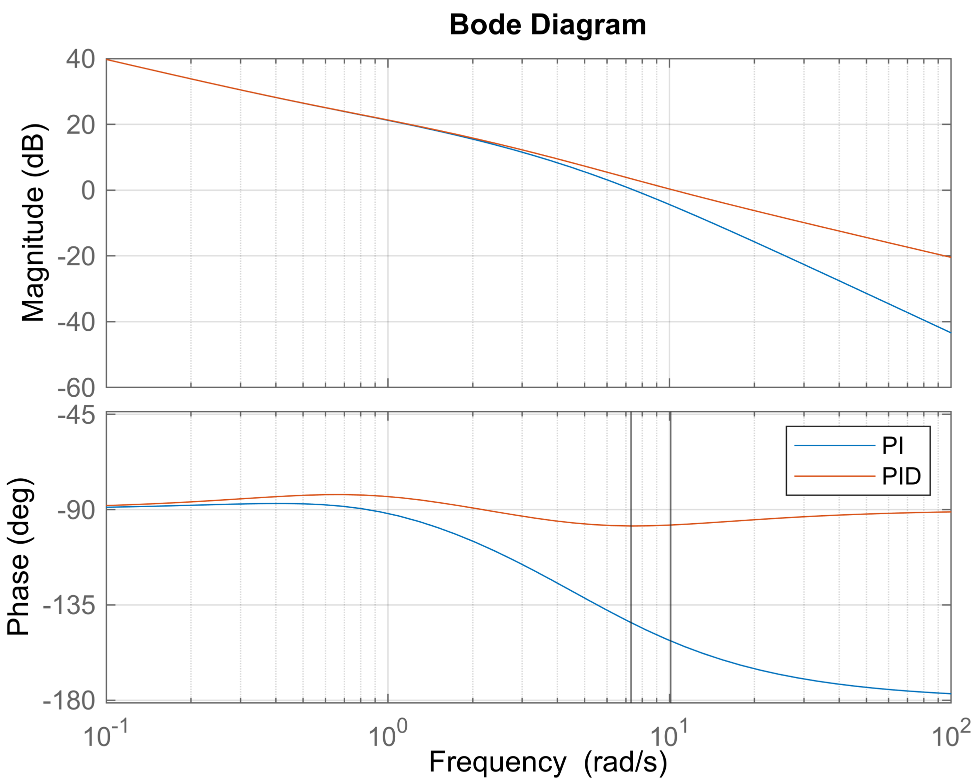
```
C2 =

  0.1402 s + 1

Continuous-time transfer function.
Model Properties
```

Com base no sistema com controlador PI C1*G, foi necessário escolher a localização do zero do PD a ser adicionado. Foram tentados 3 valores ao redor de wg, como visto no gráfico acima, e escolhido o zero que teve a melhor resposta ao degrau: wg, que é igual a -7,13 e resulta num Kd = 0,07. A seguir analisaremos a influência desse zero em -7,13 no gráfico de bode.

```
bode(C1*G, C1*C2*G); xline(7.3); xline(10.1); legend('PI', 'PID'); grid();
```



Podemos ver que a adição do zero em -7,13 fez com que a margem de fase aumentasse bastante e passasse de 40° para 80°. A seguir, escolhemos o valor do filtro Tf para tornar o PID realizável com o valor padrão de 1/100, criando um polo em -100, uma frequência bem mais alta que os outros polos e zeros do sistema. e comparamos a resposta em degrau com o PID sem o filtro.

```
Tf = 1/100;
PID = pid(0.5, 0.357, 0.07, Tf) % Kp, Ki, Kd, Tf
```

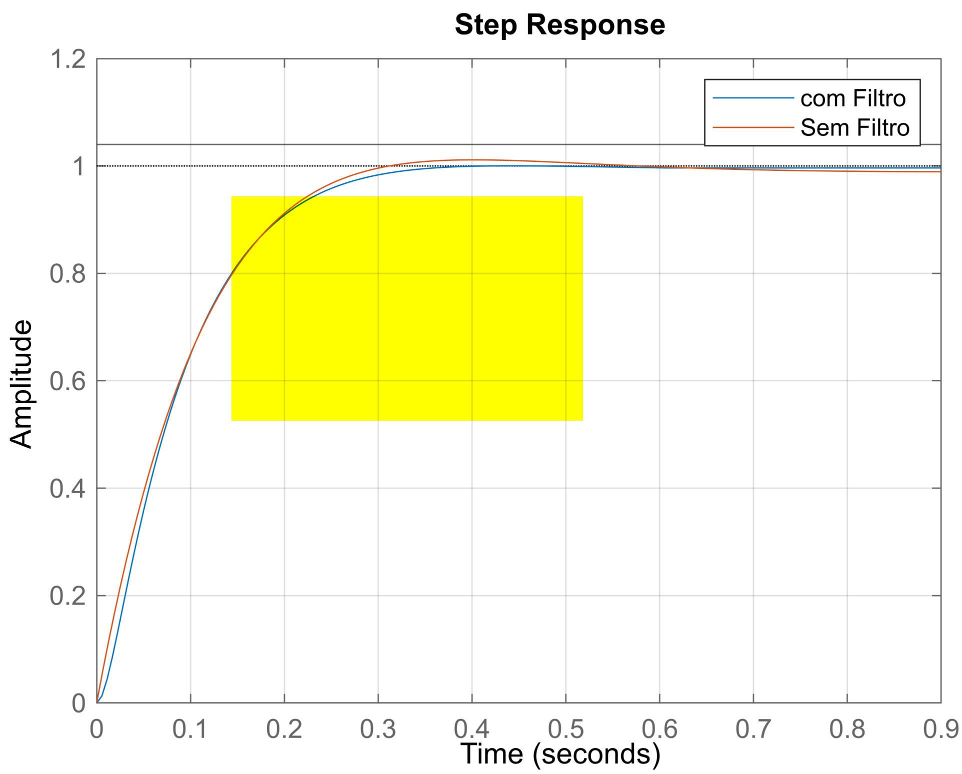
```
PID =

             1            s
  Kp + Ki * --- + Kd * --------
             s          Tf*s+1

  with Kp = 0.5, Ki = 0.357, Kd = 0.07, Tf = 0.01

Continuous-time PIDF controller in parallel form.
Model Properties
```

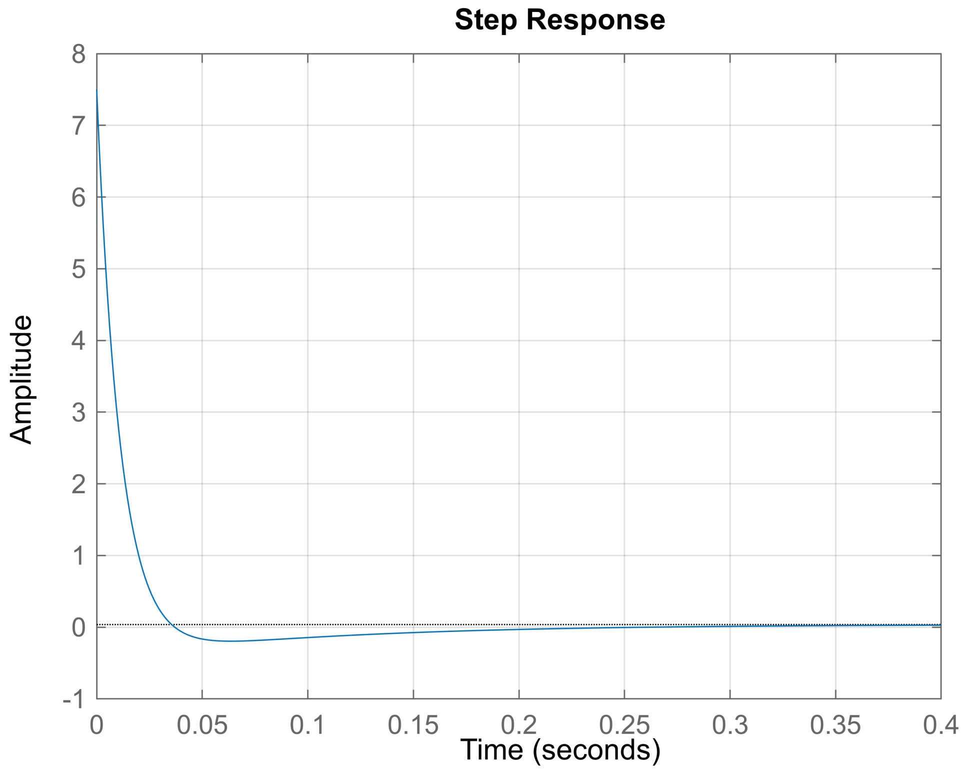
```
step(feedback(PID*G, 1), feedback(C1*C2*G, 1)); xline(ts0); yline(1.04);
legend('com Filtro', 'Sem Filtro'); grid();
```



Podemos ver na resposta em malha fechada a entrada de degrau, que o sistema atende às especificações: erro de regime nulo, baixa sobreelevação e tempo de estabelecimento quase 3 vezes menor do que o ts0 especificado de 1,5s.

A seguir exibimos o gráfico do sinal de controle em resposta a entrada em degrau.

```
step(feedback(PID, G)); grid();
```



Vemos que, para alcançar a resposta ao degrau obtida, é necessário que o sinal de controle assuma valores 8 vezes maiores do que a entrada, que pode ser um limitante no sistema real.

Parte III: Estabilização

Fazer os gráficos de Nyquist (à mão ou no Matlab) e aplicar o critério

Primeiro analisamos o sistema G1 sem controlador:

```
G1
```

```
G1 =

  27
  ---
  s^2

Continuous-time transfer function.
Model Properties
```

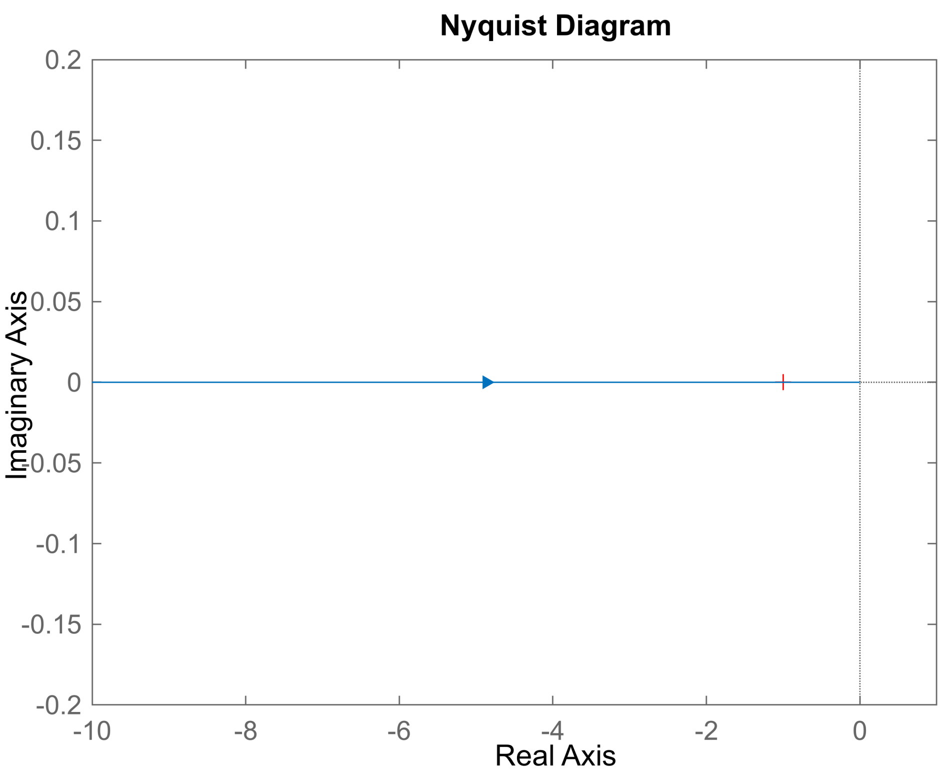
Pelo critério de Nyquist:

$$\phi = \left(Z_d - P_d - \frac{P_w}{2}\right)180^\circ$$

Analisando G1, temos Pw = 2 e Pd = 0. Portanto, para estabilidade (Zd = 0):

$$\phi = \left(0 - 0 - \frac{2}{2}\right)180° = -180°$$

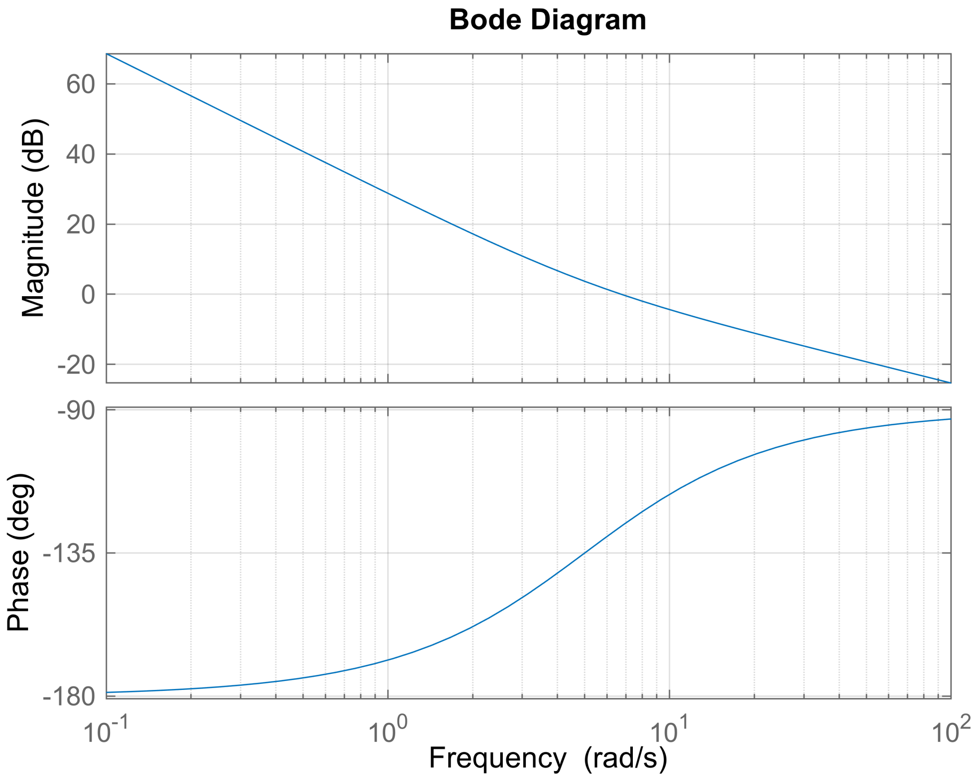
```
figure;
h=nyquistplot(G1);
set(h,'ShowFullContour','off');
ylim([-0.2 0.2]); xlim([-10 1]);
```



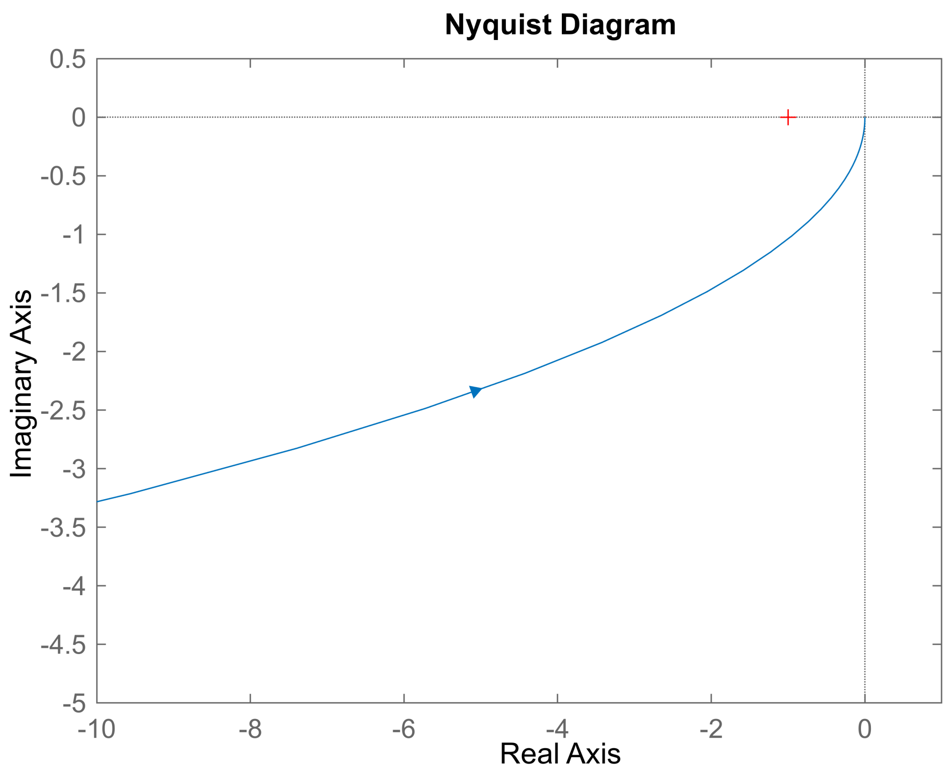
Pelo gráfico de Nyquist, o sistema seria estável se a curva passasse por baixo do ponto -1, porém, como a curva passa exatamente sobre o eixo real, o sistema sem controle é marginalmente estável.

Agora analisando se um PD pode estabilizar G1:

```
PD = projpd(G1, 5);
bode(PD*G1);grid();
```




```
figure;
h=nyquistplot(PD*G1);
set(h,'ShowFullContour','off');
ylim([-5 0.5]); xlim([-10 1]);
```



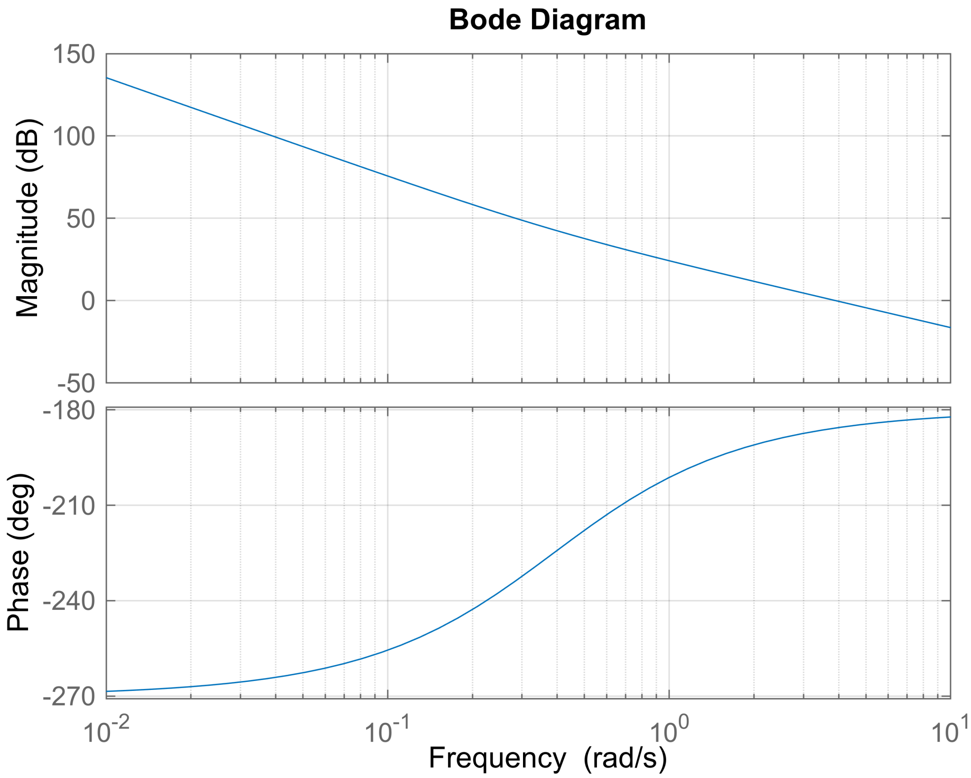
Ao se utilizar um PD com qualquer valor do zero adicionado, o critério de Nyquist permanece o mesmo, com ϕ=-180°, porém no gráfico de nyquist, a curva polar passa a ficar por baixo do ponto -1, fazendo com que o ϕ seja de fato -180° e tornando o sistema estável. Isso porque o gráfico polar se iniciará com fase -180° para frequências próximas de zero e passará para fase de -90° com a frequência tendendo ao infinito, sem outro polo para atrasar mais a fase e passar por cima do ponto -1.

Analisando para um PI:

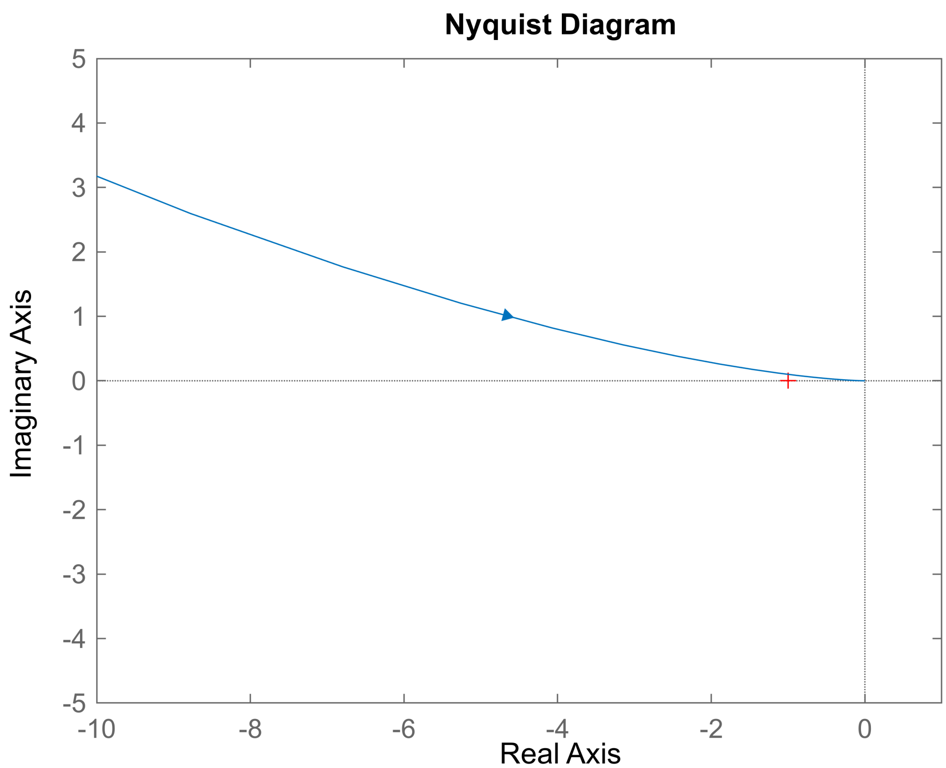
Para o PI, com a adição de mais um polo na origem, o critério de nyquist passa a ser:

$$\phi = \left(0 - 0 - \frac{3}{2}\right)180^\circ = -270^\circ$$

```
PI = projpi(G1, -5);
bode(PI*G1);grid();
```




```
figure;
h=nyquistplot(PI*G1);
set(h,'ShowFullContour','off');
ylim([-5 5]); xlim([-10 1]);
```



Para qualquer controlador PI projetado, o sistema nunca será estabilizado. Já que, para isso, a curva polar precisaria passar por debaixo do ponto -1 para ter ϕ=-270°, o que significa que a fase do sistema teria que estar mais avançada do que -180° em algum ponto. Entretando, como temos 3 polos na origem e apenas 1 zero, a fase estará sempre entre -270° e -180°, já que o zero irá avançar a fase em no máximo 90°. Também podemos ver isso no gráfico de bode, onde a margem de fase é sempre negativa com a adição de um polo na origem e um zero.