

**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO**  
**CENTRO TECNOLÓGICO**  
**CURSO DE ENGENHARIA DE COMPUTAÇÃO**



**ANDRÉ LOUIS SOUZA RIBEIRO (2019107791)**  
**ANDRÉ OLIVEIRA CUNHA (2019107756)**

**TRABALHO 2 DE SISTEMAS REALIMENTADOS**

**Vitória - ES**  
**2022**

SUMÁRIO

<b>Projeto 1</b>	<b>2</b>
Projeto 1.1.i	6
Projeto 1.1.ii	16
Projeto 1.2.i	29
Projeto 1.2.ii	41
Projeto 1.3	44

## Projeto 1

A dinâmica de um veículo lunar (ver Fig. P12.10 pág 571 do livro do Dorf e Bishop) é dada por:

$$G = \frac{500N}{s(s + (N + 3))(s + \frac{25}{\sqrt{N}})} e^{-(T)s}$$

Devemos projetar controlador PID, usando a resposta em frequência, para que o sistema atenda às seguintes especificações:

- I. Erro à entrada rampa  $\leq 1$
- II. Erro à entrada ao distúrbio  $\leq 1$
- III. Margem de fase  $\geq 60^\circ$
- IV. Largura de banda maior possível.

Sendo N neste caso, o número da dupla, que deve ser substituído na equação, no nosso caso o  $N = 4$ . Substituindo o N da dupla, ficará da seguinte forma a planta:

$$G = \frac{2000}{s(s + 7)(s + 12,5)} e^{-(T)s}$$

ou então, feita no Matlab:

$$\frac{2000}{s^3 + 19.5 s^2 + 87.5 s}$$

Podemos projetar um controlador PID, primeiro fazendo um controlador PI e depois um controlador PD. Em que esses tem característica de filtro passa baixa e filtro passa alta. Por esse motivo, o controlador PID tem característica de um filtro passa faixa. O controlador PID pode ser definido por:

$$PID = KP + KI/s + sKd = (1 + sKdd)(Kpi + kii/s)$$

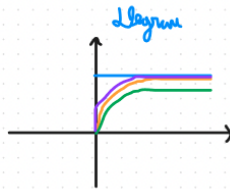
Vamos analisar primeiro, os **critérios** que o sistema deve obedecer. Para analisar o erro em regime permanente, podemos perceber que o sistema é do tipo 2 e entrada rampa, e podemos utilizar a seguinte tabela de erro para auxiliar nossos cálculos:

Temos algumas classificações de sistema:

Tipo 0  $\rightarrow s^0$

Tipo 1  $\rightarrow s^1$

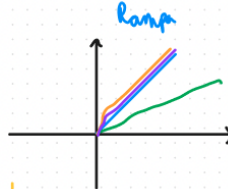
Tipo 2  $\rightarrow s^2$



Tipo 0:  $e(\infty) = \frac{1}{1+K_p}$ ,  $K_p = \lim_{s \rightarrow 0} G(s)$

Tipo 1:  $e(\infty) = 0$

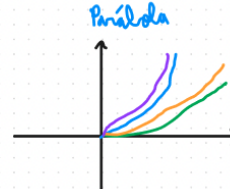
Tipo 2:  $e(\infty) = 0$



Tipo 0:  $e(\infty) = \infty$

Tipo 1:  $e(\infty) = \frac{1}{K_v}$ ,  $K_v = \lim_{s \rightarrow 0} sG(s)$

Tipo 2:  $e(\infty) = 0$



Tipo 0:  $e(\infty) = \infty$

Tipo 1:  $e(\infty) = \infty$

Tipo 2:  $e(\infty) = \frac{1}{K_a}$ ,  $K_a = \lim_{s \rightarrow 0} s^2 G(s)$

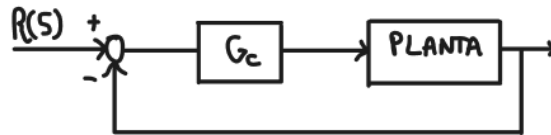
	tipo 0	tipo 1	tipo 2
Degrau	$1 / 1 + k_p$	0	0
Rampa	$\infty$	$1/k_v$	0
Parábola	$\infty$	$\infty$	$1/k_a$

Podemos comprovar que o erro em regime será zero por meio dos cálculos:

Dado o controlador PID com equação:

$$G_c = \frac{K_d s^2 + k_i s + K_p}{s}$$

Sabendo também que o diagrama de blocos do sistema é:



$$G_{MA} = \frac{K_d s^2 + k_i s + K_p}{s} \cdot \frac{2000}{s(s+7)(s+12,5)}$$

$$G_{MA} = \frac{2000(K_d s^2 + k_i s + K_p)}{s^2(s^2 + 19,5s + 87,5)}$$

Sabendo que a realimentação é unitária e conhecendo a  $G_{MA}$ , podemos calcular o erro a entrada rampa como:

$$K_v = \lim_{s \rightarrow 0} s \cdot G_{MA} \cdot H$$

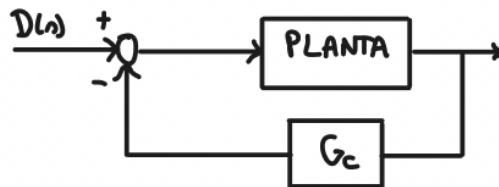
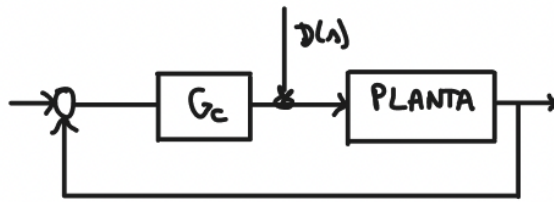
$$K_v = \lim_{s \rightarrow 0} s \cdot \frac{2000(K_d s^2 + k_i s + K_p)}{s^2(s^2 + 19,5s + 87,5)} \cdot 1$$

$$K_v = \infty$$

Então o erro em regime é:

$$ess(\infty) = \frac{1}{K_v} = \frac{1}{\infty} = 0$$

Para o distúrbio, o nosso diagrama de blocos fica:



Fazendo análise similar as anterior, porém com  $H = G_c$ , então:

$$K_v = \lim_{s \rightarrow 0} s \cdot G_{ma} \cdot H$$

$$K_v = \lim_{s \rightarrow 0} s \left( \frac{2000}{s(s+7)(s+12,5)} \cdot \frac{K_d s^2 + k_i s + K_p}{s} \right)$$

$$K_v = \lim_{s \rightarrow 0} s \cdot \frac{2000 (K_d s^2 + k_i s + K_p)}{s^2 (s^2 + 19,5s + 87,5)}$$

$$K_v = \infty$$

Então o erro em regime a entrada distúrbio é:

$$e_{std}(\infty) = \frac{1}{K_v} = \frac{1}{\infty} = 0$$

## Projeto 1.1.i

Devemos projetar um controlador com  $T=0$ , isso quer dizer que não tem atraso de transporte.

Para projetarmos o PID vamos seguir os passos fornecidos pelo professor, Sendo eles :

1. Determinar a  $GM$ ,  $PM$ , frequências de cruzamento de ganho e de fase e a largura de banda da Planta dada pela função de transferência  $GP$ ;
2. Obter a nova frequência de cruzamento de ganho ( $\omega_{0dB}$ ) através da relação:  
 $25^\circ = 180^\circ + \text{Fase de } GP(\omega_{0dB})$
3. Calcular o ganho proporcional do PI ( $KPI$ ) de maneira que  $GMA = GP \times KPI$  tenha a MF aproximadamente igual a  $25^\circ$  na nova frequência de cruzamento de ganho ( $\omega_{0dB}$ ). Ou seja escolher  $KPI$  tal que:  $20 \log(KP) + 20 \log[GP(\omega_{0dB})] = 0$ ;
4. Escolher a frequência de corte do PI tal que o atraso de fase do PI ocorra um pouco abaixo da nova frequência de cruzamento de ganho. Por exemplo:  $KI/KP = \omega_{0dB}/2$  ou  $KI/KP = \omega_{0dB}/5$  Em seguida, simular o PI com a planta, e verificar se a resposta é estável e rápida.
5. Projetar o PD dado por  $1 + sKdd$  considerando a FTMA  $GMA = GP \times [KPI \times (KII/s)]$  e que a MF especificada seja maior ou igual a  $60^\circ$  na nova frequência de cruzamento de ganho obtida no item 3. Isto pode ser obtido escolhendo a frequência de corte do PD tal que:  $(1/Kdd = \omega_{0dB})$
6. Simular o sistema realimentado com o controlador PID projetado para a entrada degrau.
7. Refazer o projeto caso a resposta ao degrau não seja satisfatória;

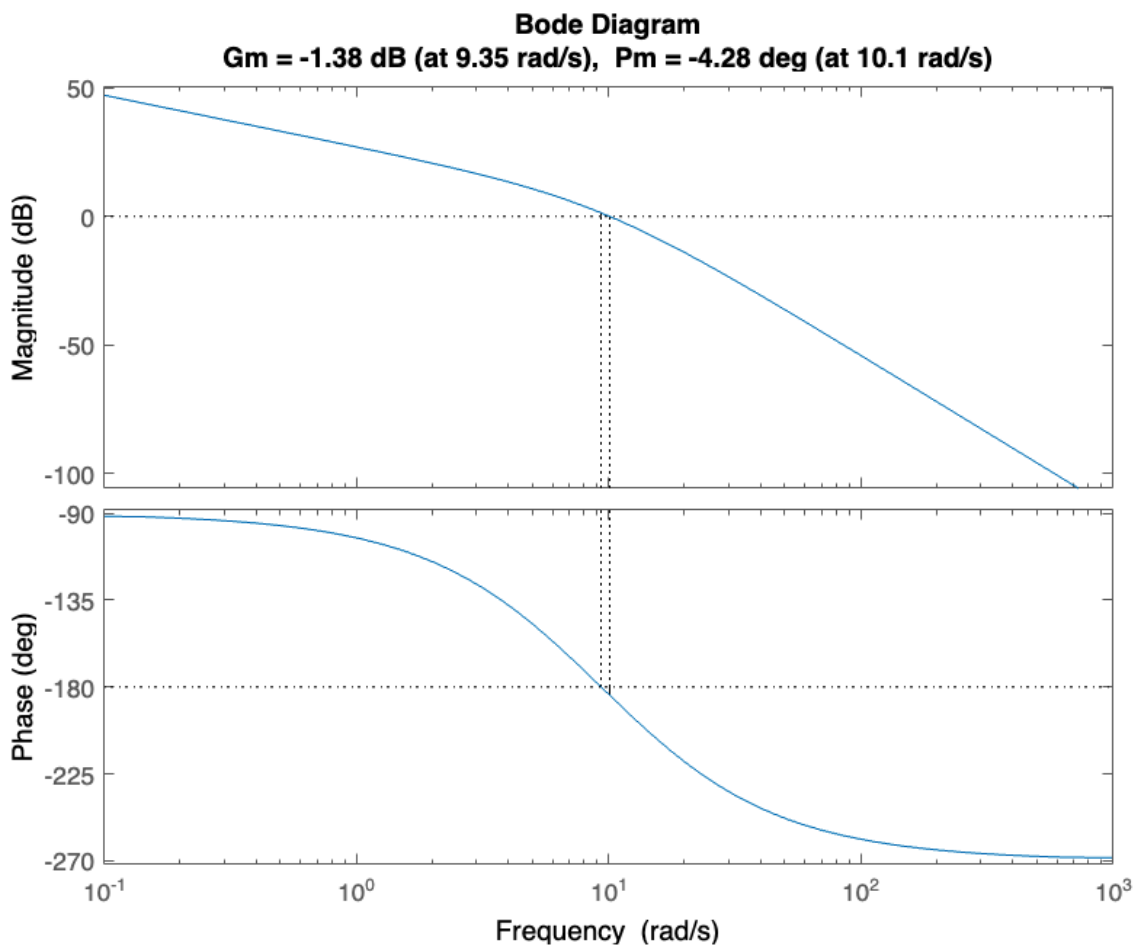
Inicialmente, vamos substituir na nossa planta os valores e simular a mesma, para que seja possível extrair alguns dados iniciais de margem de fase e margem de ganho, inclusive suas frequências e largura de banda. Para achar a largura de banda devemos olhar no gráfico de bode a frequência onde a amplitude corresponde a -3 dB. Fazemos isso com o seguinte trecho de código:

```

%%
syms s
T = 0;
N = 4;
num = 500*N;
den = conv([1 0], conv([1 3+N], [1 25/sqrt(N)]));
g_planta = tf(num, den, 'InputDelay', T);
[Gm,Pm,Wgm,Wpm] = margin(g_planta)

```

Vamos obter o seguinte gráfico de Bode:



**Gm = 0.8531; Pm = -4.2847; Wgm = 9.3541 rad/s ; Wpm = 10.1131 rad/s**

Podemos notar, pelo gráfico de margem e fase, que o gráfico de bode está instável.

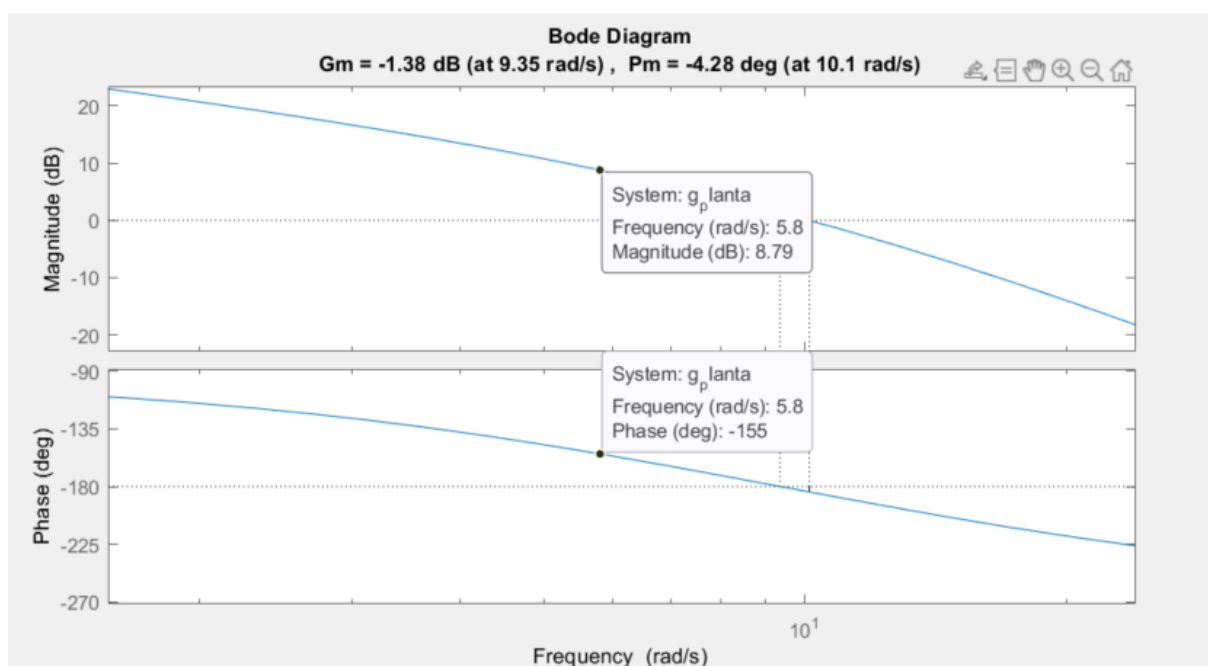


Para elaboração do controlador PI, devemos então, definir uma nova frequência de cruzamento de ganho. Utilizando a seguinte fórmula:

$$25^\circ = 180^\circ + \text{Fase de GP}(\omega_{0dB})$$

$$\text{Fase de GP}(\omega_{0dB}) = -155^\circ$$

Olhando o gráfico de bode para uma fase de  $-155^\circ$ , temos uma frequência  $\omega = 5,8$  (rad/s). Nessa mesma frequência, temos um ganho de aproximadamente 8.79 dB. Como podemos ver na seguinte imagem:



Agora, para que a nova frequência de cruzamento de ganho atenda a margem de fase especificada de  $25^\circ$ , devemos calcular o ganho do controlador proporcional do controlador PI, para modificar o gráfico da magnitude. Podemos encontrar o ganho proporcional do PI ( $K_{PI}$ ), utilizando log para converter o valor do dB.

$$20 \log K_{PI} = -8,79$$

$$K_{PI} = 10^{(-8,79/20)}$$

$$K_{PI} = 0,3635$$

Agora que temos o valor de KPI, podemos multiplicá-lo pela planta. Depois disso, simular e pegar as informações das novas margens de ganho e fase, a frequência de corte e visualizar o novo gráfico de bode.

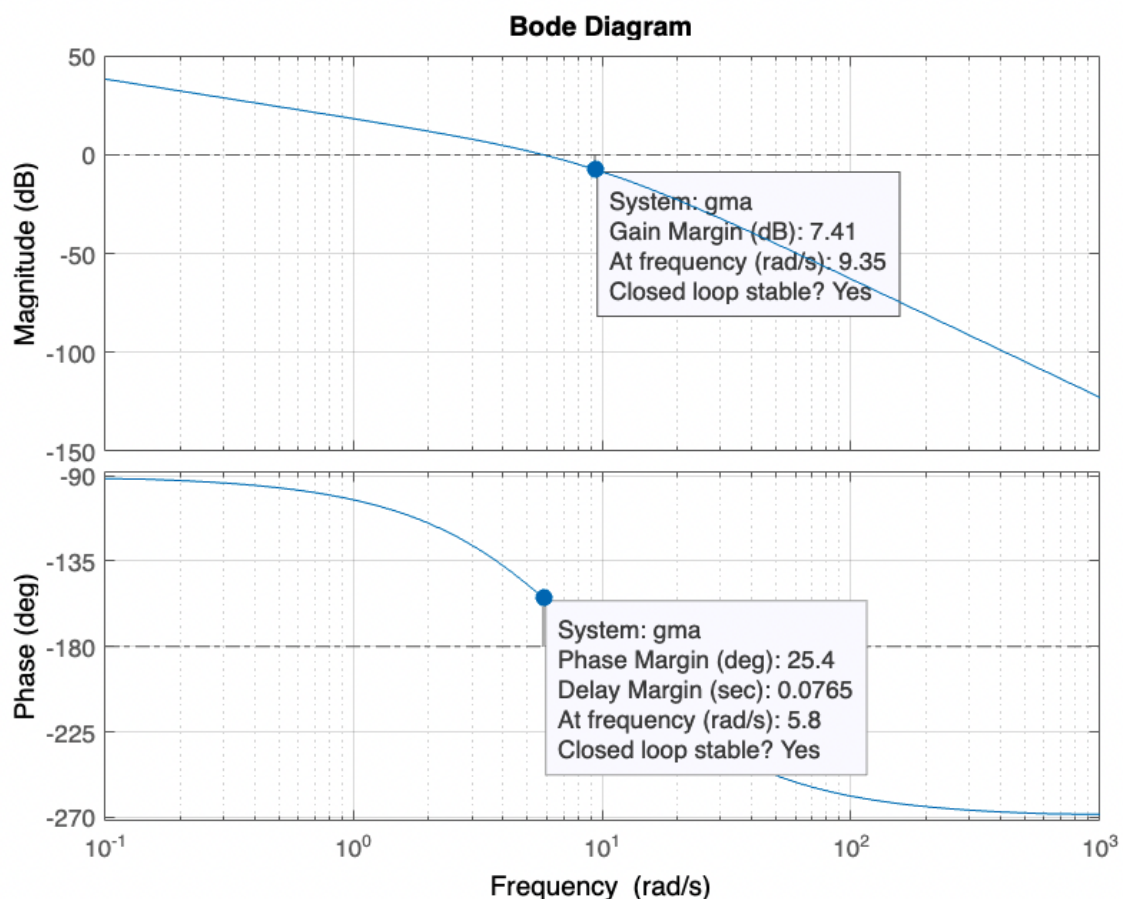
O código fica:

```
% Nova g_planta com o kpi
gma = g_planta*kpi;
[MGgma, MFgma, wGgma, wFgma] = margin(gma);%retorno do margin

% Novos valores para as margens
MGgma
MFgma
wGgma
wFgma

margin(gma); % Mostra o grafico de bode com a margem de fase e ganho
```

O gráfico do G da planta com o Kpi:



**Gm = 2.3570; Pm= 25.4461.; Wgm = 9.3541; Wpm = 5.8021;kpi =0.3635**

### Nova Gma:

$$\frac{727}{s^3 + 19.5 s^2 + 87.5 s}$$

Agora é necessário encontrar uma frequência de corte cujo atraso de fase do PI esteja um pouco abaixo da nova frequência de cruzamento de ganho. Para isso vamos utilizar a seguinte fórmula:

$$K_{ii} = k_{pi} * W_{0db} / X$$

Logo, serão realizados testes para diferentes valores de **X** para estipular um bom valor para  $K_{ii}$ .

Para achar um  $K_{ii}$  bom para a planta, vamos rodar um código em loop para adicionar alguns possíveis valores e armazená-los em um array. Posteriormente vamos fazer outro loop para gerar uma tabela com os valores de sobressinal, tempo de subida e estabelecimento para cada planta multiplicada por esses valores de  $K_{ii}$  salvos.

```
% Projetando o controlador PI
kii_array = []; % Possiveis valores de ki para o controlador PI
gmf_pi_array = []; % Possiveis novas plantas com o controlador PI
for i = [2.5 3 5 7 9 10 12 14 16 18 20]
    result_kii = kpi*wFgma/i;
    kii_array = [kii_array,result_kii];

    result_pi = tf([kpi result_kii], [1 0]);
    result_gma = g_planta*result_pi;

    result_gmf = feedback(result_gma,1);
    gmf_pi_array = [gmf_pi_array, result_gmf];
end

% gera a tabela para os valores de KII na planta
index = 1; % Contador para a tabela
col_pi = [];

for i = gmf_pi_array
    OS_pi=0;
    RT_pi=0;
    ST_pi=0;
```

```

st_info=stepinfo(i);
OS_pi(index)=st_info.Overshoot; % MP
RT_pi(index)=st_info.RiseTime; % Tr
ST_pi(index)=st_info.SettlingTime; % Ts

col_pi = [col_pi, [OS_pi(index) RT_pi(index) ST_pi(index)
kii_array(index) ]'];
index = index+1;
end

Tabela1=table(col_pi,'RowNames',{'Sobressinal (%)','T. Subida (s)', 'T.
Estabelecimento(s)', 'KII' });

```

Com esses resultados conseguimos gerar uma tabela e comparar os resultados obtidos.

### Valores de Kii:

X	2.5	3	5	7	9	10	12
Sobressinal(%)	105.6898	97.0766	78.7409	70.4354	65.7090	64.0364	61.5101
T. Subida(s)	0.1721	0.1759	0.1846	0.1890	0.1916	0.1926	0.1941
T. Estabelecimento(s)	30.5010	11.9626	5.2334	4.1616	3.7463	3.7356	3.7222
KII	0.8436	0.7030	0.4218	0.3013	0.2343	0.2109	0.1758

Com base na tabela acima, concluímos que a melhor escolha é o **Kii = 0.1758**, pois ele está com um o melhor tempo de assentamento e além disso com um baixo tempo de subida. Então avaliando esse conjunto, essa é a melhor escolha para o projeto.

Agora que projetamos o controlador PI, devemos então projetar o controlador PD (1 + sKdd). E com isso conseguiremos projetar nosso controlador PID.

Podemos utilizar a seguinte fórmula para encontrar **Kdd**:

$$Kdd = X / W0db$$

Podemos encontrar o melhor X, pegando cada um dos valores de Kdd, colocando na fórmula do PD que é  $1 + sK_{dd}$ , e multiplicando pelo controlador PI que foi projetado. Dessa forma, vamos ter o nosso controlador PID, multiplicando esse controlador pela planta vamos ter o resultado final do projeto. E com isso podemos analisar se os requisitos foram atendidos.

Fizemos um código no matlab que gera alguns possíveis valores para Kdd com a fórmula acima e salva em um array.

```
kdd_array = [];
gmf_pid_array = [];

for i = [1 1.1 1.2 1.3 1.4 1.5 1.6 1.7 2 2.5 3]
    result_kdd = i/wFgma;
    kdd_array = [kdd_array, result_kdd];

    % Controlador PID
    n_pid = conv([kpi kii_array(7)], [result_kdd 1]);
    d_pid = [1 0];
    result_pid = tf(n_pid, d_pid);

    result_gma_pid = g_planta*result_pid;

    result_gmf = feedback(result_gma_pid,1);
    gmf_pid_array = [gmf_pid_array, result_gmf];
end

col_pid = [];
index = 1;
for i = gmf_pid_array
    OS_pid=0;
    RT_pid=0;
    ST_pid=0;
    st_info=stepinfo(i);
    OS_pid(index)=st_info.Overshoot;
    RT_pid(index)=st_info.RiseTime;
    ST_pid(index)=st_info.SettlingTime;

    col_pid = [col_pid, [OS_pid(index) RT_pid(index) ST_pid(index)
kdd_array(index)]];
    index = index+1;
end
Tabela2=table(col_pid,'RowNames',{'Sobressinal (%)','T. Subida (s)',
'T. Estabelecimento(s)', 'KDD'});
```

A tabela abaixo mostra alguns possíveis candidatos de Kdd.

#### Valores de Kdd:

X	Kdd	Sobressinal(%)	tr	ta	Largura de Banda
---	-----	----------------	----	----	------------------

1.5	0.2585	6.8878	0.1340	2.5026	13.4 rad/s
1.6	0.2758	6.6852	0.1287	2.5220	14 rad/s
1.7	0.2930	6.6721	0.1238	2.5400	14.5 rad/s
2	0.3447	7.2115	0.1113	2.5852	16.1 rad/s
2.5	0.4309	9.2226	0.0952	2.6245	18.4 rad/s

**Tabela - Testes do Kdd**

Portanto, fazendo a análise da tabela, escolhemos o Kdd no valor de **0.2930**, sendo essa a melhor escolha dentre os outros valores, pois possui o menor sobressinal e os tempos de assentamento e subida são bons comparados com os outros.

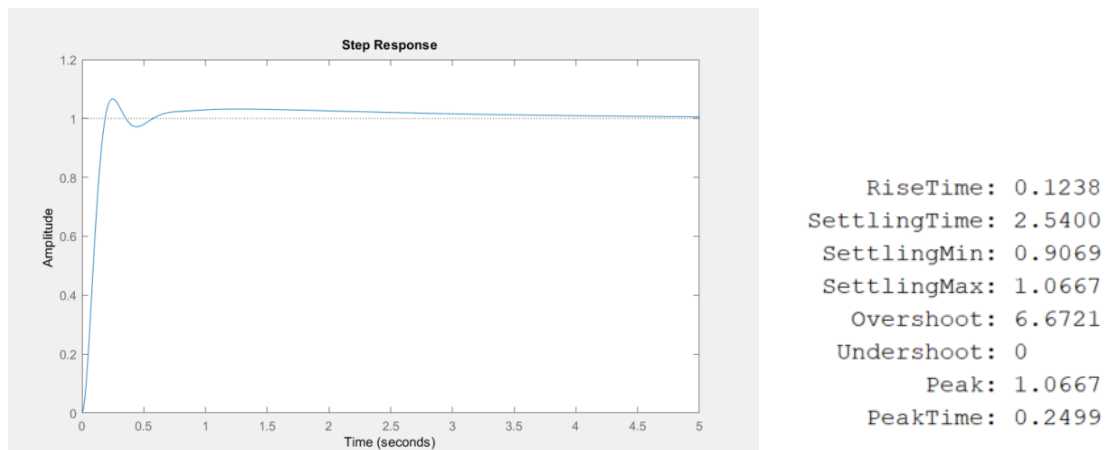
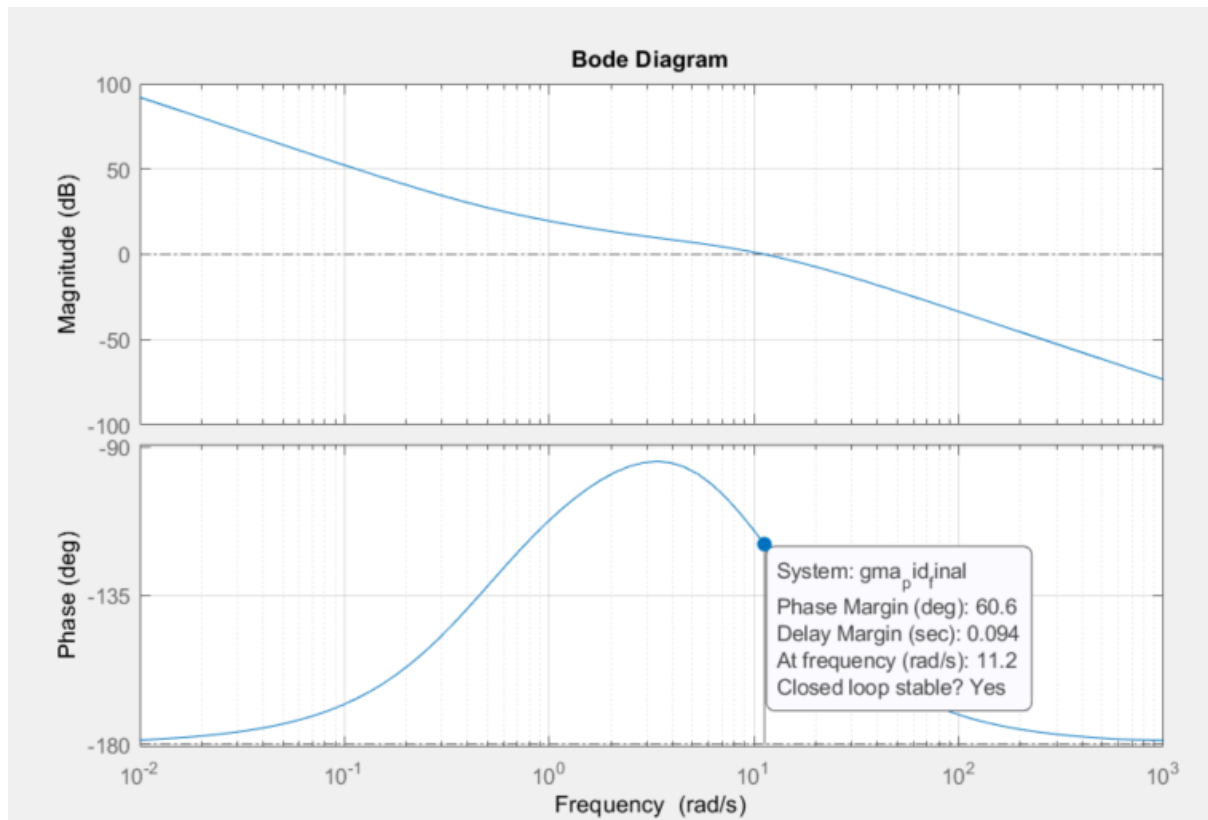
Dito isso, chegamos ao seguinte controlador **PID**:

$$\mathbf{PID = (1 + sK_{dd})(K_{pi} + k_{ii}/s)}$$

$$\mathbf{PID = (1 + s0.2930)(0.3635 + 0.1758/s)}$$

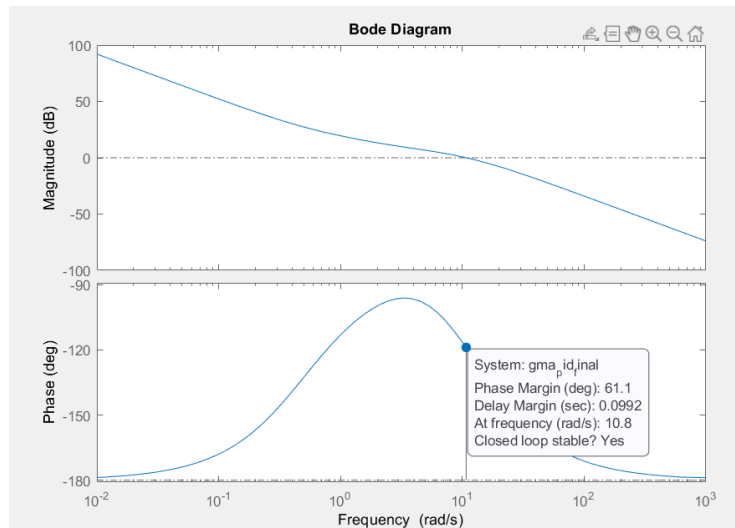
Podemos simular o sistema e verificar que os requisitos foram atendidos, incluindo o de obter uma MF > 60°. Analisando o gráfico de bode, da planta multiplicada pelo

controlador:

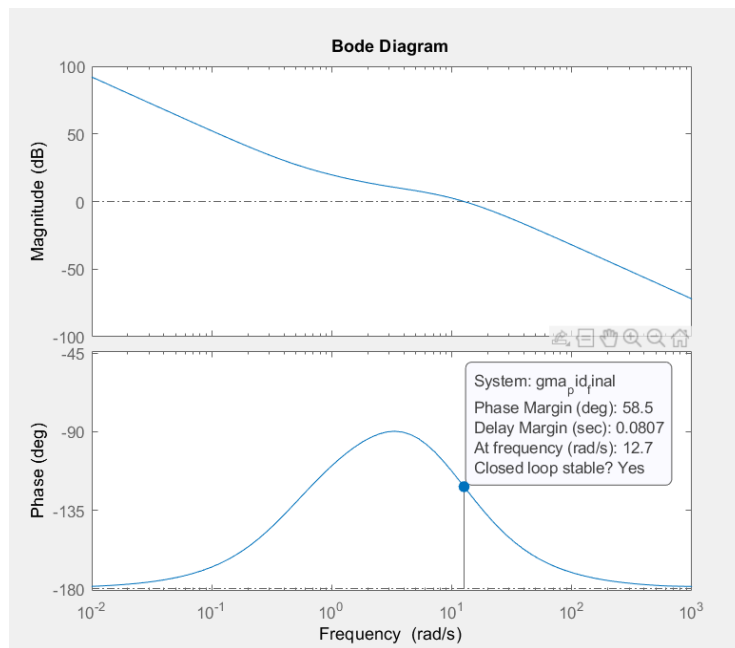


Podemos perceber que a margem de fase está dentro do requisito de ter a margem fase maior que 60 graus e erro menor que 1, logo, conseguimos projetar um controlador para esse sistema. Além disso, vemos que a resposta ao degrau do sistema tem bons resultados.

Podemos simular o que aconteceria se aumentássemos ou diminuíssemos o  $K_{dd}$  e quais Margens de fase iremos encontrar.



$K_{dd} = 0.2758$  (Menor do que o escolhido)



$K_{dd} = 0.3447$  (Maior do que o escolhido)

Logo, podemos verificar que a nossa escolha do  **$K_{dd} = 0.2930$**  foi a melhor escolha para o projeto, possuindo com os melhores resultados.

O código abaixo gera a Gma com o melhor PID projetado.

```
% A gma com o melhor PID
n_pid_final = conv([kpi kii_array(8)], [kdd_array(5) 1]);
d_pid_final = [1 0];
pid_final = tf(n_pid_final, d_pid_final);
```



```

gma_pid_final = g_planta * pid_final;

% Novos valores de margem para o controlador PID final
[MG_pid2, MF_pid2, wG_pid2, wF_pid2] = margin(gma_pid_final);

gmf_pid_final = feedback(gma_pid_final,1);
margin(gma_pid_final);

```

## Projeto 1.1.ii

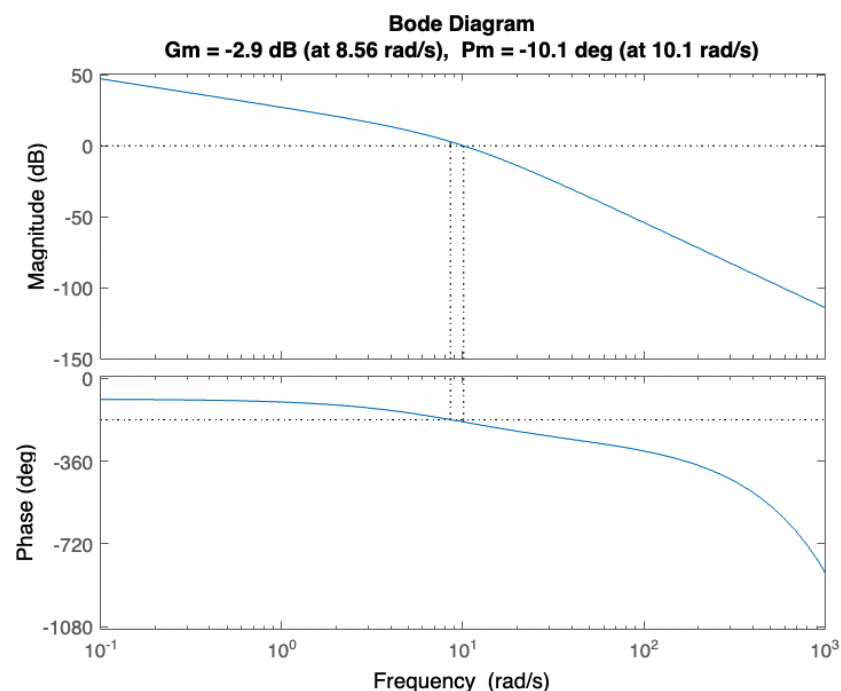
Para projetar o sistema com atraso de transporte de 0.01s, vamos seguir as mesmas etapas realizadas no projeto anterior.

Inicialmente, vamos substituir na nossa planta os valores e simular a mesma, para que seja possível extrair alguns dados iniciais de margem de fase e margem de ganho, inclusive suas frequências e largura de banda.

```

syms s
T = 0.01;
N = 4;
num = 500*N;
den = conv([1 0], conv([1 3+N], [1 25/sqrt(N)]));
g_planta = tf(num, den, 'InputDelay', T);
[Gm, Pm, Wgm, Wpm] = margin(g_planta)

```



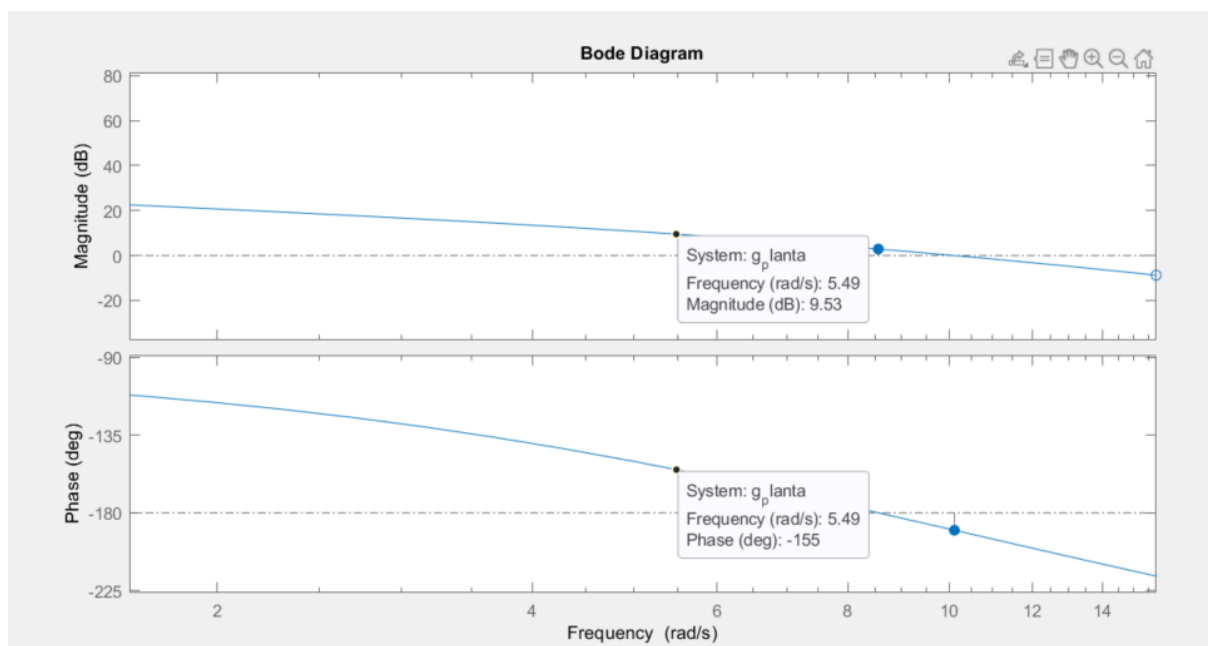
$$G_m = 0.7162; P_m = -10.0791; W_{gm} = 8.55 \text{ rad/s}; W_{pm} = 10.11 \text{ rad/s}$$

Para elaboração do controlador PI, devemos então, definir uma nova frequência de cruzamento de ganho. Utilizando a seguinte fórmula:

$$25^\circ = 180^\circ + \text{Fase de GP}(\omega_{0dB})$$

$$\text{Fase de GP}(\omega_{0dB}) = -155^\circ$$

Olhando o gráfico de bode para uma fase de  $-155^\circ$ , temos uma frequência  $w = 5,49$  (rad/s). Nessa mesma frequência, temos um ganho de aproximadamente 9.53 dB. Como podemos ver na seguinte imagem:



Agora para que a nova frequência de cruzamento de ganho atenda a margem de fase especificada de  $25^\circ$ , devemos calcular o ganho do controlador proporcional do controlador PI, para modificar o gráfico da magnitude. Podemos encontrar o ganho proporcional do PI (KPI), utilizando log para converter o valor do dB.

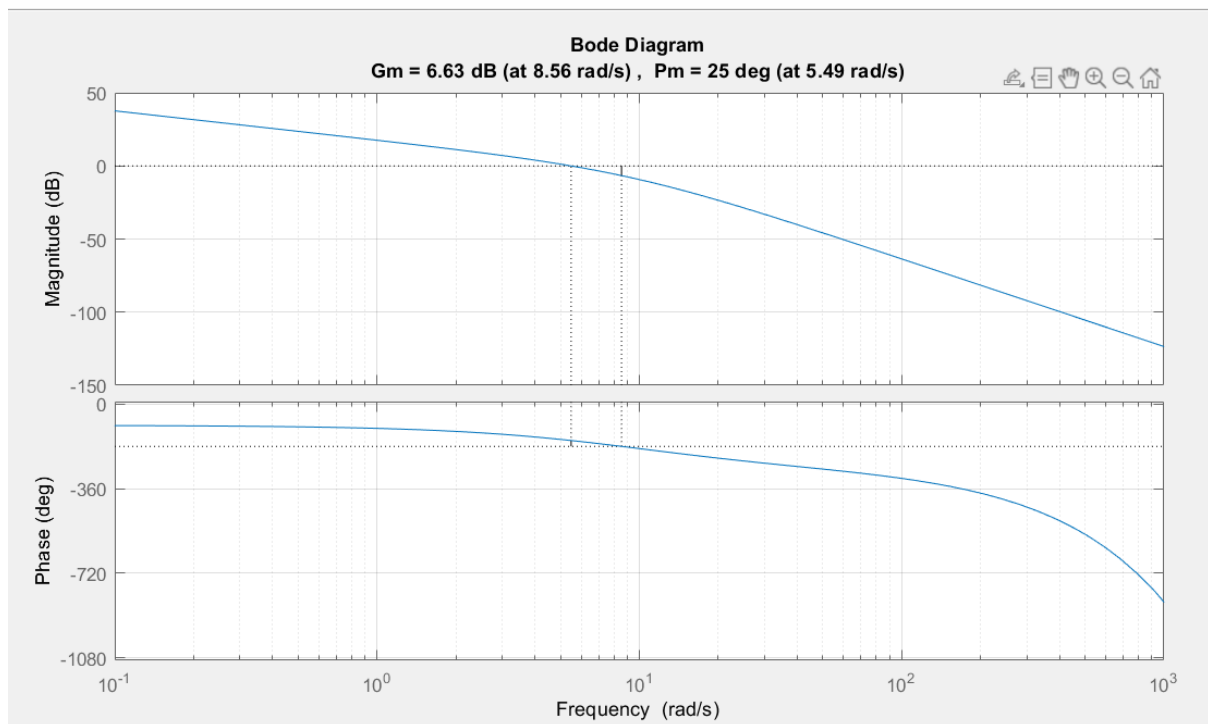
$$20 \log K_{pi} = -9,53$$

$$K_{pi} = 10^{(-9,53/20)}$$

$$K_{pi} = 0,3338$$

Agora que temos o valor de KPI, podemos multiplicá-lo pela planta. Depois disso, simular e pegar as informações das novas margens de ganho e fase, a frequência de corte e visualizar o novo gráfico de bode.

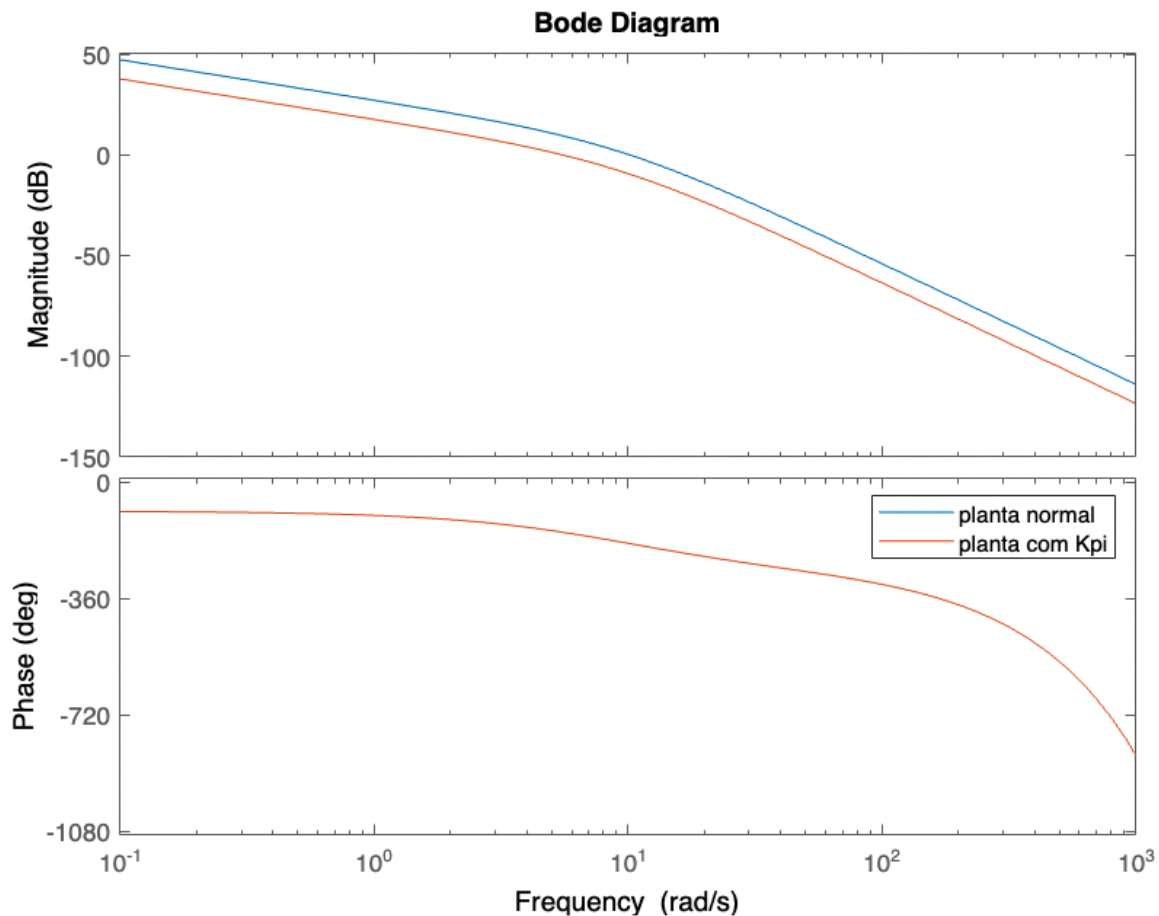
**O gráfico do G da planta com o Kpi:**



$$Gm = 2.1456; Pm = 24.9944; Wgm = 8.55 \text{ rad/s}; Wpm = 5.49 \text{ rad/s}$$

**Nova gma:**

$$\exp(-0.01*s) * \frac{667.6}{s^3 + 19.5 s^2 + 87.5 s}$$



Agora é necessário encontrar uma frequência de corte cujo atraso de fase do PI esteja um pouco abaixo da nova frequência de cruzamento de ganho. Para isso vamos utilizar a seguinte fórmula:

$$K_{ii} = k_{pi} * W_{0db} / X$$

Logo, serão realizados testes para diferentes valores de **X** para estipular um bom valor para  $K_{ii}$ .

Para achar um  $K_{ii}$  bom para a planta, vamos rodar um código em loop para adicionar alguns possíveis valores e armazená-los em um array. Posteriormente vamos fazer outro loop para gerar uma tabela com os valores de sobressinal, tempo de subida e estabelecimento para cada planta multiplicada por esses valores de  $K_{ii}$  salvos.

Gerando alguns possíveis valores de **Kii** e analisando o sobressinal, tempo de subida e de estabelecimento temos a tabela a seguir:

<b>X</b>	<b>Kii</b>	<b>Sobressinal(%)</b>	<b>tr</b>	<b>ta</b>
5	0.3668	80.7678	0.1929	7.1949
9	0.2038	67.4534	0.2002	3.9567
10	0.1834	65.7465	0.2010	3.9456
12	0.1528	63.1438	0.2024	3.9315
14	0.1310	61.2711	0.2035	3.9227

Um bom valor de **Kii**, com base na tabela gerada é de 0.1528, visto que ele tem um dos tempos de assentamento mais baixos e uma resposta de subida mais rápida. Portanto, avaliando esse conjunto, essa é a melhor escolha.

Agora que projetamos o controlador PI, devemos então projetar o controlador PD ( $1 + sK_{dd}$ ). E com isso conseguiremos projetar nosso controlador PID.

Podemos utilizar a seguinte fórmula para encontrar **Kdd**:

$$K_{dd} = X / W_{0db}$$

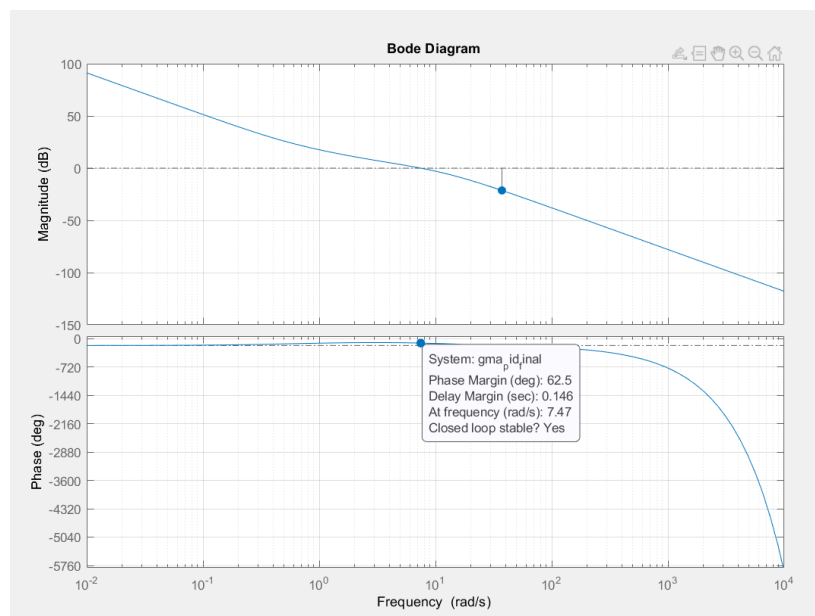
Podemos encontrar o melhor **X**, pegando cada um dos valores de **Kdd**, colocando na fórmula do PD que é  $1 + sK_{dd}$ , e multiplicando pelo controlador PI que foi projetado. Dessa forma, vamos ter o nosso controlador PID, multiplicando esse controlador pela planta vamos ter o resultado final do projeto. E com isso podemos analisar se os requisitos foram atendidos.

<b>Kdd</b>	<b>Sobressinal(%)</b>	<b>tr</b>	<b>ta</b>
0.2002	10.9518	0.1541	NaN
0.2093	10.4322	0.1506	NaN

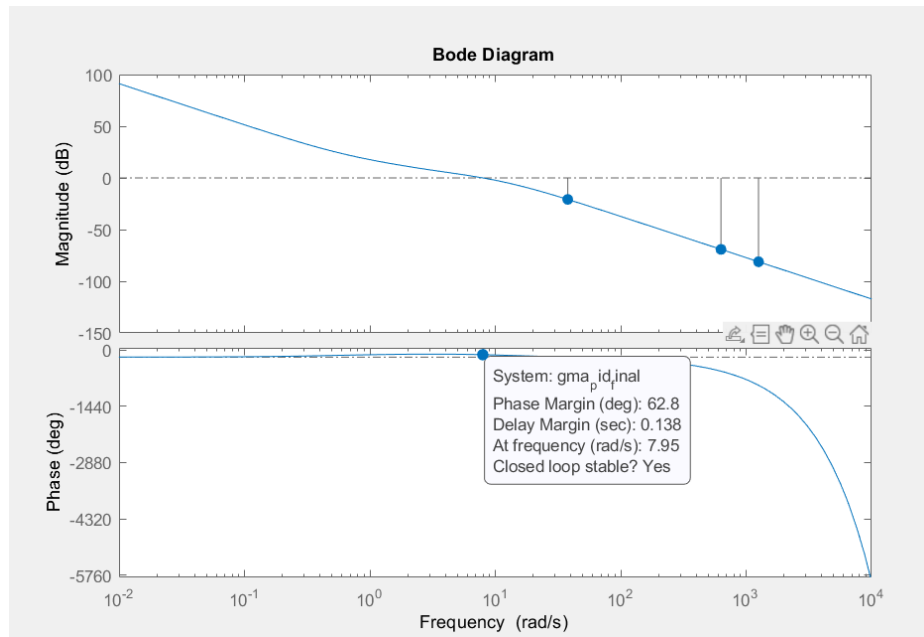
0.2184	10.0141	0.1472	NaN
0.2275	9.6823	0.1438	NaN
0.3640	11.2019	0.1047	NaN

Nos resultados obtidos percebemos a aparição de muitos valores **NaN** no tempo de assentamento, isso provavelmente está indicando que o sistema não possui um tempo finito de assentamento. Além disso, quando simulamos o nosso controlador PID multiplicado pela planta, chegamos a MF 62.5°, porém com uso desses valores NaN nas colunas do tempo de assentamento.

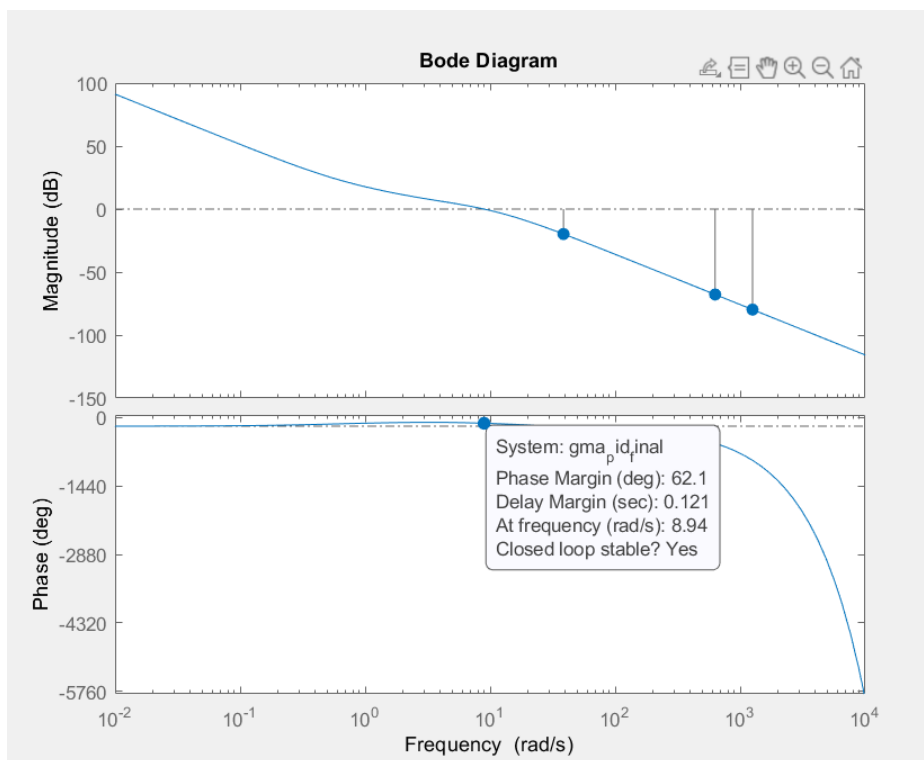
**Simulação Kdd = 0.2093; MF = 62.5°; Ta = NaN;**



**Simulação Kdd = 0.2184; MF = 62.8°; Ta = NaN;**



**Simulação  $K_{dd} = 0.3640$ ; MF =  $62.1^\circ$ ;  $T_a = NaN$ ;**



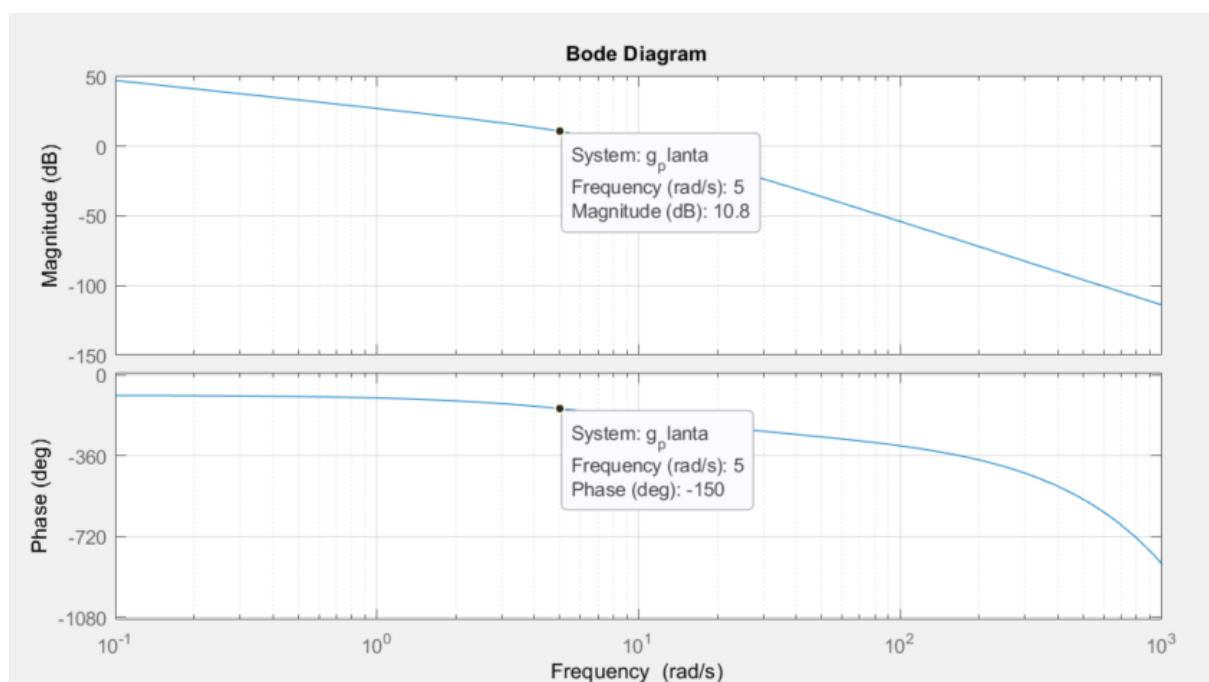
Também foram realizadas mudanças no **Kii** escolhido, mesmo assim o sistema continuou com esses valores NaN. E após alguns testes, concluímos que não é possível projetar esse controlador com esses requisitos com a escolha da nova margem de fase de  $25^\circ$ . Então, vamos mudar a nossa Margem de fase para  $30^\circ$  e

obter uma nova frequência de cruzamento de ganho para analisar se vai atender aos requisitos pedidos.

$$30^\circ = 180^\circ + \text{Fase de } GP(\omega_{0dB})$$

$$\text{Fase de } GP(\omega_{0dB}) = -150^\circ$$

Olhando o gráfico de bode para uma fase de  $-150^\circ$ , temos uma frequência  $\omega = 5$  (rad/s). Nessa mesma frequência, temos um ganho de aproximadamente 10.8 dB. Como podemos ver na seguinte imagem:



**Fase:  $-150^\circ$ , Magnitude: 10.8, Frequência: 5**

Agora, para que a nova frequência de cruzamento de ganho atenda a margem de fase especificada de  $30^\circ$ , devemos calcular o ganho do controlador proporcional do controlador PI, para modificar o gráfico da magnitude. Podemos encontrar o ganho proporcional do PI (KPI), utilizando log para converter o valor do dB.



$$20 \log K_{pi} = -10,8$$

$$K_{pi} = 10^{(-10,8/20)}$$

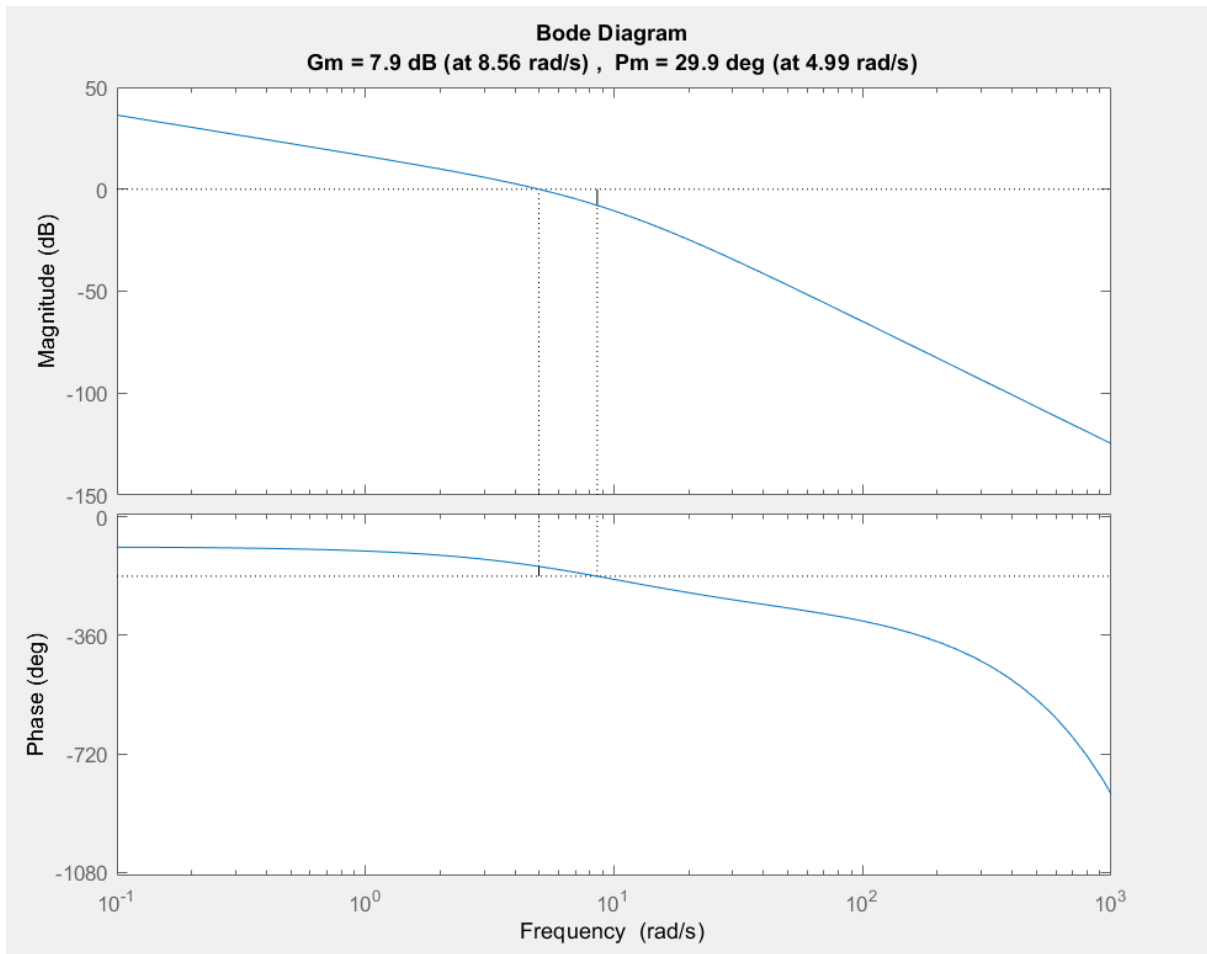
$$K_{pi} = 0,2884$$

Agora que temos o valor de KPI, podemos multiplicá-lo pela planta. Depois disso, simular e pegar as informações das novas margens de ganho e fase, a frequência de corte e visualizar o novo gráfico de bode.

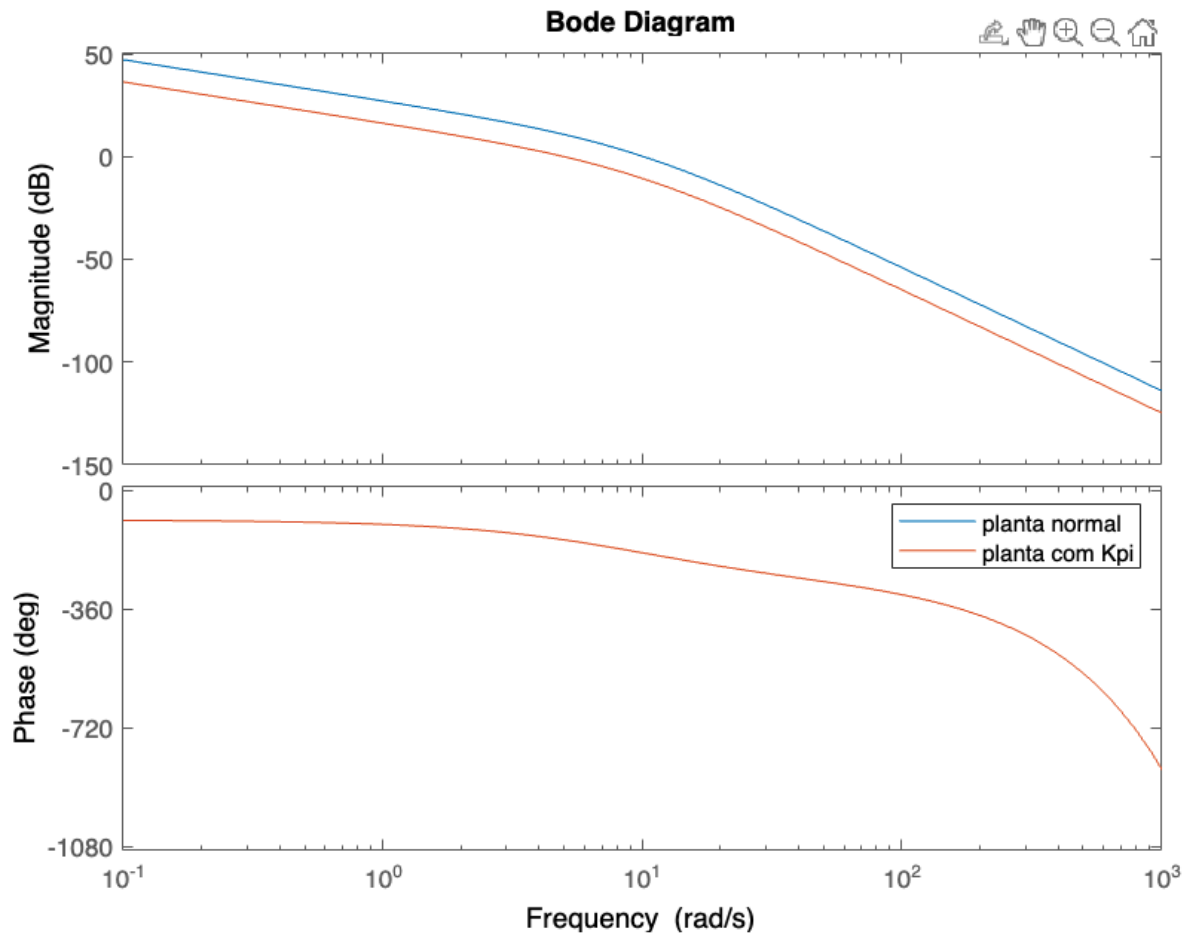
**Nova Gma:**

$$\exp(-0.01*s) * \frac{576.8}{s^3 + 19.5 s^2 + 87.5 s}$$

Continuous-time transfer function.



**Gm = 2.48; Pm = 29.92; Wgm = 8.55 rad/s ; Wpm = 4.98 rad/s**



Agora é necessário encontrar uma frequência de corte cujo atraso de fase do PI esteja um pouco abaixo da nova frequência de cruzamento de ganho. Para isso vamos utilizar a seguinte fórmula:

$$K_{ii} = k_{pi} * W_{0db} / X$$

Logo, serão realizados testes para diferentes valores de **X** para estipular um bom valor para **Kii**.

<b>X</b>	<b>Kii</b>	<b>Sobressinal(%)</b>	<b>tr</b>	<b>ta</b>
3	0.4794	90.0403	0.2004	7.2538
5	0.2876	72.0903	0.2103	4.2559
7	0.2055	64.0683	0.2164	3.1987
9	0.1598	59.4774	0.2196	3.1888

Um bom valor de  $K_{ii}$ , com base na tabela gerada é de **0.2055**, visto que ele tem um dos tempos de assentamento mais baixos e uma resposta de subida mais rápida. Portanto, avaliando esse conjunto, essa é a melhor escolha.

Agora que projetamos o controlador PI, devemos então projetar o controlador PD ( $1 + sK_{dd}$ ). E com isso conseguiremos projetar nosso controlador PID.

Podemos utilizar a seguinte fórmula para encontrar  **$K_{dd}$** :

$$K_{dd} = X / W_{0db}$$

Podemos encontrar o melhor  **$X$** , pegando cada um dos valores de  $K_{dd}$ , colocando na fórmula do PD que é  $1 + sK_{dd}$ , e multiplicando pelo controlador PI que foi projetado. Dessa forma, vamos ter o nosso controlador PID, multiplicando esse controlador pela planta vamos ter o resultado final do projeto. E com isso podemos analisar se os requisitos foram atendidos.

<b>X</b>	<b>K<sub>dd</sub></b>	<b>Sobressinal(%)</b>	<b>tr</b>	<b>ta</b>	<b>Largura de Banda</b>
1.1	0.2206	7.8818	0.1679	2.8399	11.3 rad/s
1.2	0.2406	7.0015	0.1590	NaN	10.8 rad/s
1.6	0.3208	6.3020	0.1333	2.9768	13.1 rad/s

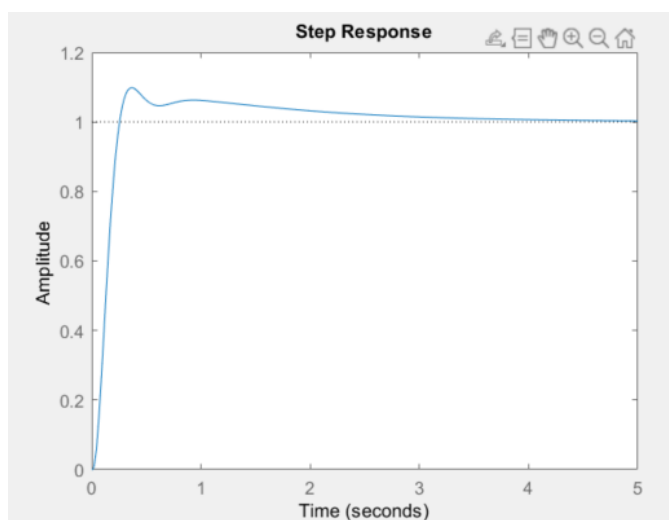
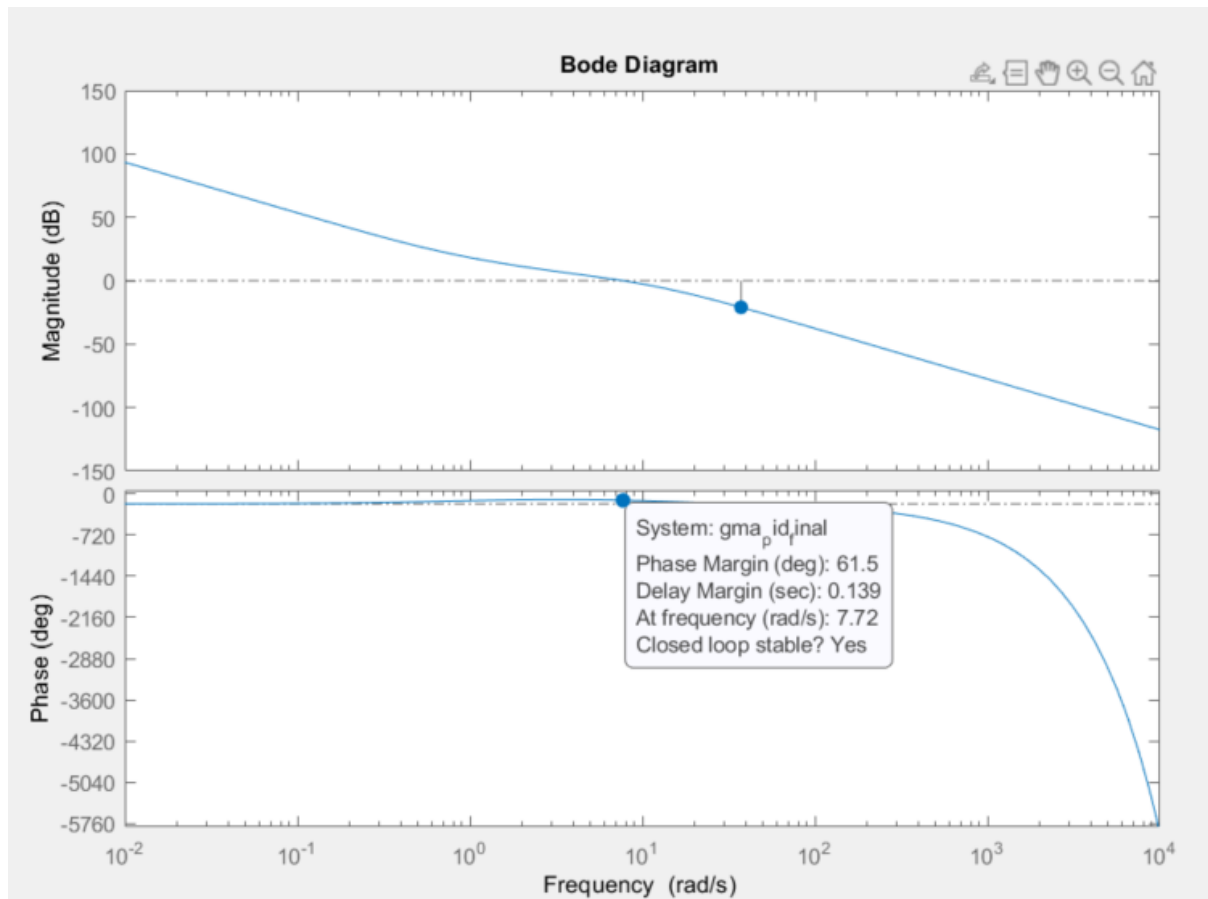
Um bom valor de  $K_{dd}$ , com base na tabela gerada é de **0.2206**, visto que ele tem o tempo de assentamento mais baixo. Além disso não tem NaN na em sua coluna que está sendo utilizada, os NaN diminuíram bastante com o aumento da margem de fase, aparecendo poucos dessa vez.

Dito isso, chegamos ao seguinte controlador PID:

$$PID = (1 + sK_{dd})(K_{pi} + k_{ii}/s)$$

$$PID = (1 + s0.2206)(0.2884 + 0.2055/s)$$

Podemos simular o sistema e verificar que os requisitos foram atendidos, incluindo o de obter uma MF > 60°. Analisando o gráfico de bode, da planta multiplicada pelo controlador:



RiseTime: 0.1644  
 SettlingTime: 2.5836  
 SettlingMin: 0.9614  
 SettlingMax: 1.0985  
 Overshoot: 9.8458  
 Undershoot: 0  
 Peak: 1.0985  
 PeakTime: 0.3681

Podemos perceber que a margem de fase está dentro do requisito (**MF = 61.5**), além disso, conseguimos bons valores de tempo de subida, assentamento e largura de banda. Logo, conseguimos projetar um controlador para esse sistema atendendo aos requisitos.

## Projeto 1.2.i

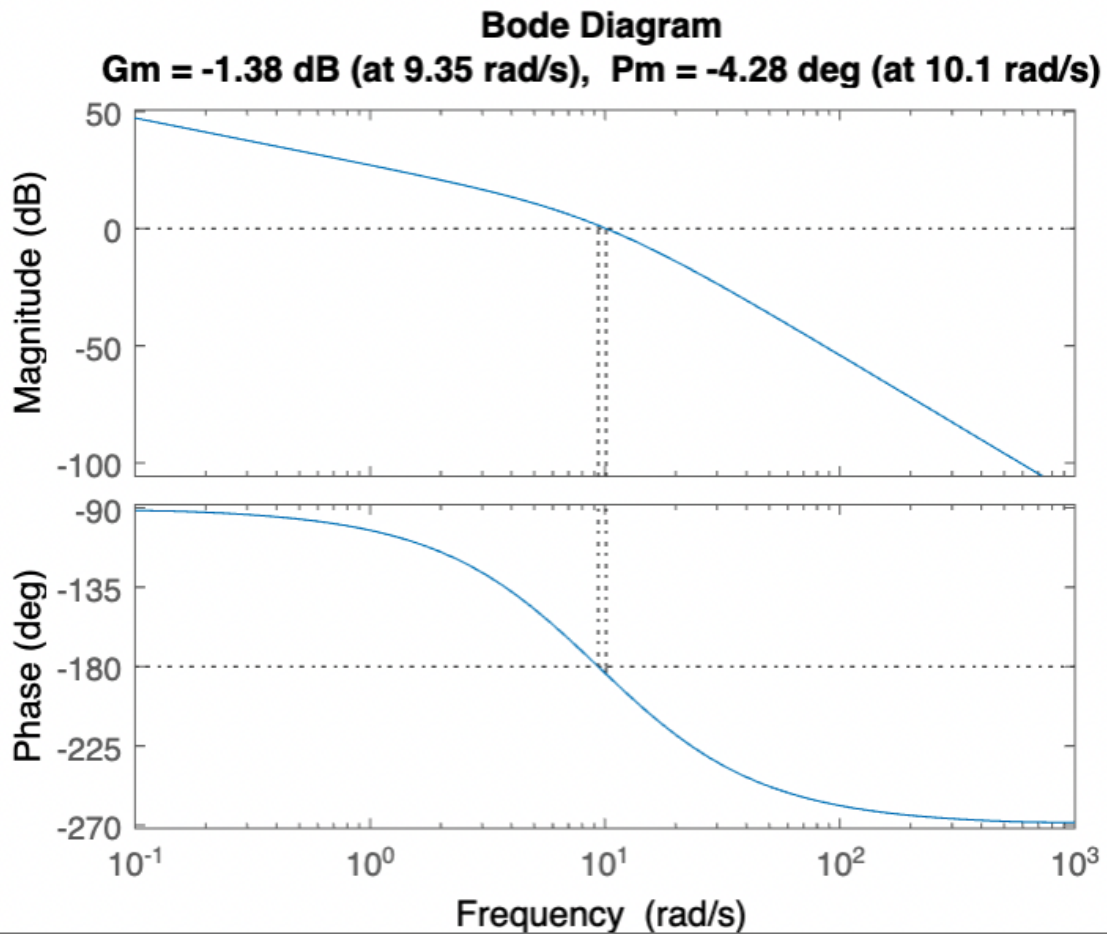
Nesse projeto, nós vamos repetir o projeto da 1.1, porém usando o controlador Atraso-Avanço de fase. Então, usando a mesma planta da questão anterior, vamos projetar um controlador que atenda ao erro à entrada rampa e ao distúrbio de rampa menores ou iguais a 1, largura de banda da função de transferência em malha aberta maior possível e margem de fase maior ou igual a 60 graus .

Para esse primeiro passo vamos fazer com  $T = 0$ , sem atraso de transporte. Seguindo o passo a passo do material do professor para projetar um controlador atraso-avanço, vamos:

- 1) Determinar o ganho  $K_1$  para atender as especificações do erro em regime e da largura de banda.

Para isso com o auxílio do Matlab, vamos gerar a nossa planta:

```
syms s
T = 0; % Tempo
N = 4; % Numero do grupo
num = 500*N;
den = conv([1 0], conv([1 3+N], [1 25/sqrt(N)]));
g_planta = tf(num, den, 'InputDelay', T);
margin(g_planta);
```



Depois usamos o numerador e denominador para calcular o erro em regime permanente a entrada rampa e o erro ao distúrbio:

```
num1 = poly2sym(conv(num,[1 0]),s);
den1 = poly2sym(den,s);
g1 = num1/den1;
% Entrada rampa
sEs = 1/(g1);
ess = double(limit(sEs,s,0,'right'));
ess
% Disturbio rampa
sEd = g1/(g1);
essd = double(limit(sEd,s,0,'right'));
essd
```

Obtivemos como resposta, para a nossa planta um erro em regime permanente a entrada rampa (ess) e o erro ao distúrbio (essd) de:

```
ess = 0.0437
```

```
essd = 1
```

Ou então por meio de cálculos feitos a mão chegamos em resultados bem próximos:

$$e_{ss}(\infty) = \lim_{s \rightarrow 0} \frac{1}{K_1 G_{MA}} < 1$$

$$e_{ss}(\infty) = \lim_{s \rightarrow 0} \frac{1}{K_1 \left( \frac{2000}{s(s+7)(s+12,5)} \right)} < 1$$

$$\frac{1}{22,85 K_1} < 1$$

$$K_1 > 0,04376$$

$$e_{ssd}(\infty) = \lim_{s \rightarrow 0} \frac{G_{MA}}{1 + K_1 G_{MA}} < 1$$

$$e_{ssd}(\infty) = \lim_{s \rightarrow 0} \frac{\left( \frac{2000}{s(s+7)(s+12,5)} \right)}{1 + K_1 \left( \frac{2000}{s(s+7)(s+12,5)} \right)} < 1$$

$$= \frac{22,85}{1 + 22,85 K_1} < 1$$

$$= K_1 > 0,956$$

Como devemos atender às duas situações, **ess** > 0.0437 e **essd** > 1, podemos escolher o valor do K1 (nosso K de atraso) com um valor igual a 2. Mas para conferir se ainda vai atender aos erros ser menor do que 1, fazemos:

```
%% Verificando o erro a rampa
k1 = 2;
sEs = 1/(k1*g1);
ess = double(limit(sEs,s,0,'right'))
```



```
%erro encontrado 0.02

% Disturbio rampa
sEd = g1/(k1*g1);
essd = double(limit(sEd,s,0,'right'))
%erro encontrado 0.500
```

E obtemos como resposta:

```
ess = 0.0219
essd = 0.5000
```

Atendendo aos requisitos do problema.

2) Dada a FT da planta ( $G_p$ ), projetaremos o controlador atraso de fase:

$$G_{Atraso} = k \frac{s + z_1}{s + p_1} = k_{atraso} \frac{1 + j\alpha_1\omega\tau_1}{1 + j\omega\tau_1}, \quad k_{atraso} = \frac{k}{\alpha_1}$$

para atender as especificações do erro em regime e largura de banda. Considerando que a MF após a aplicação deste controlador na planta seja entre 30 a 40 graus.

Multiplicando nossa planta com o K1 obtemos:

Margem de fase: -21.065872, Largura de banda: 13.835427

Como o ideal é que a MF após a aplicação do controlador seja entre 30 e 40 graus, vamos escolher o valor de 35 graus (que é um valor intermediário) e calcular então a nova frequência de cruzamento:

```
[mag,fase,wout] = bode(k1*g_planta);
mag = squeeze(mag);
fase = squeeze(phase);

% FaseGP = 35 - 180 = -145
nova_fase = 35 - 180;

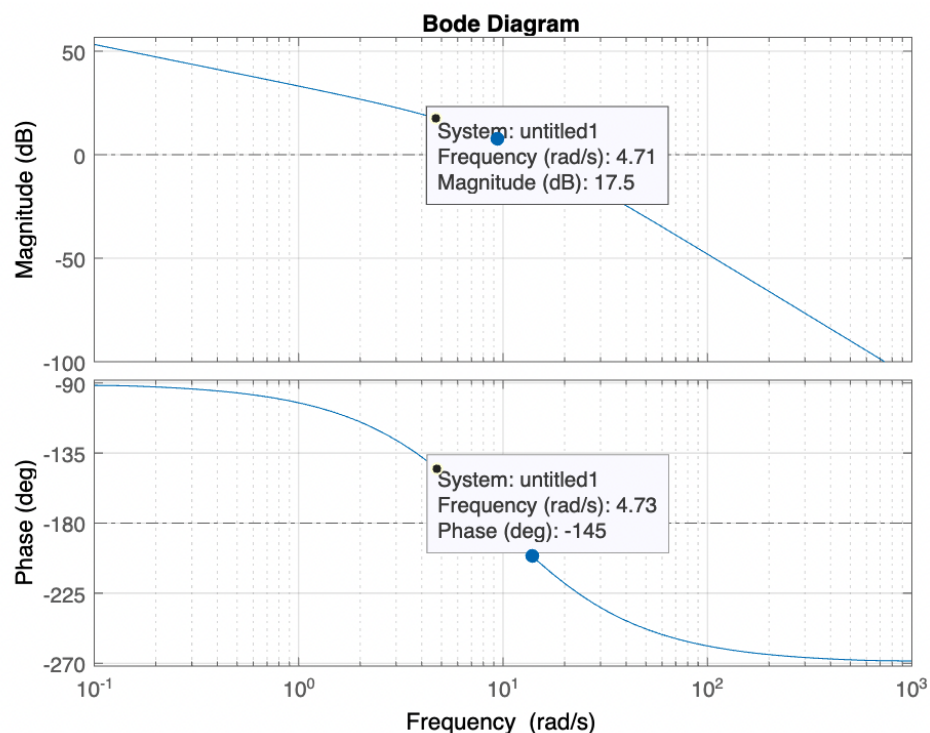
% Valor da frequência a partir da nova fase
w = interp1(fase, wout, nova_fase);

% Valor do ganho a partir da frequencia
mod = interp1(wout, 20*log10(mag), w);
```

```
mod
figure()
bode(k1*g_planta);
```

```
w(frequência) = 4.7595
mod = 17.4303
```

Podemos conferir isso também no gráfico esses valores do ganho e da frequência pelo gráfico de bode:



Logo, tanto por meio de código quanto pela análise do diagrama de bode, chegamos a conclusão de que a nova frequência de cruzamento de ganho deve ser 4.7 rad/s e o módulo nessa frequência é 17.5 dB.

Agora, obtendo a constante  $\alpha_1$  do controlador atraso de fase:

```
syms alfa1
equacao = 20*log10(mod) + 20*log10(alfa1);
alfa1 = double(solve(equacao));
```

```
alfa1 = 0.0574
```

Escolhendo a frequência de corte do zero do compensador uma década abaixo, ou seja,  $4.7/10 = 0.47$  da nova frequência de cruzamento de ganho, encontramos a frequência de corte do pólo do compensador:

```
syms tau1
equacao = 1/(alfa1*tau1) - w/10;
tau1 = double(solve(equacao))
```

O valor de tau1, do controlador de atraso encontrado foi de **36.62**.

A frequência de corte do zero do compensador ( $1/(alfa1*tau1)$ ) encontrada foi de **0.4759** e a frequência de corte do pólo do compensador ( $1/(tau1)$ ) de **0.0273**.

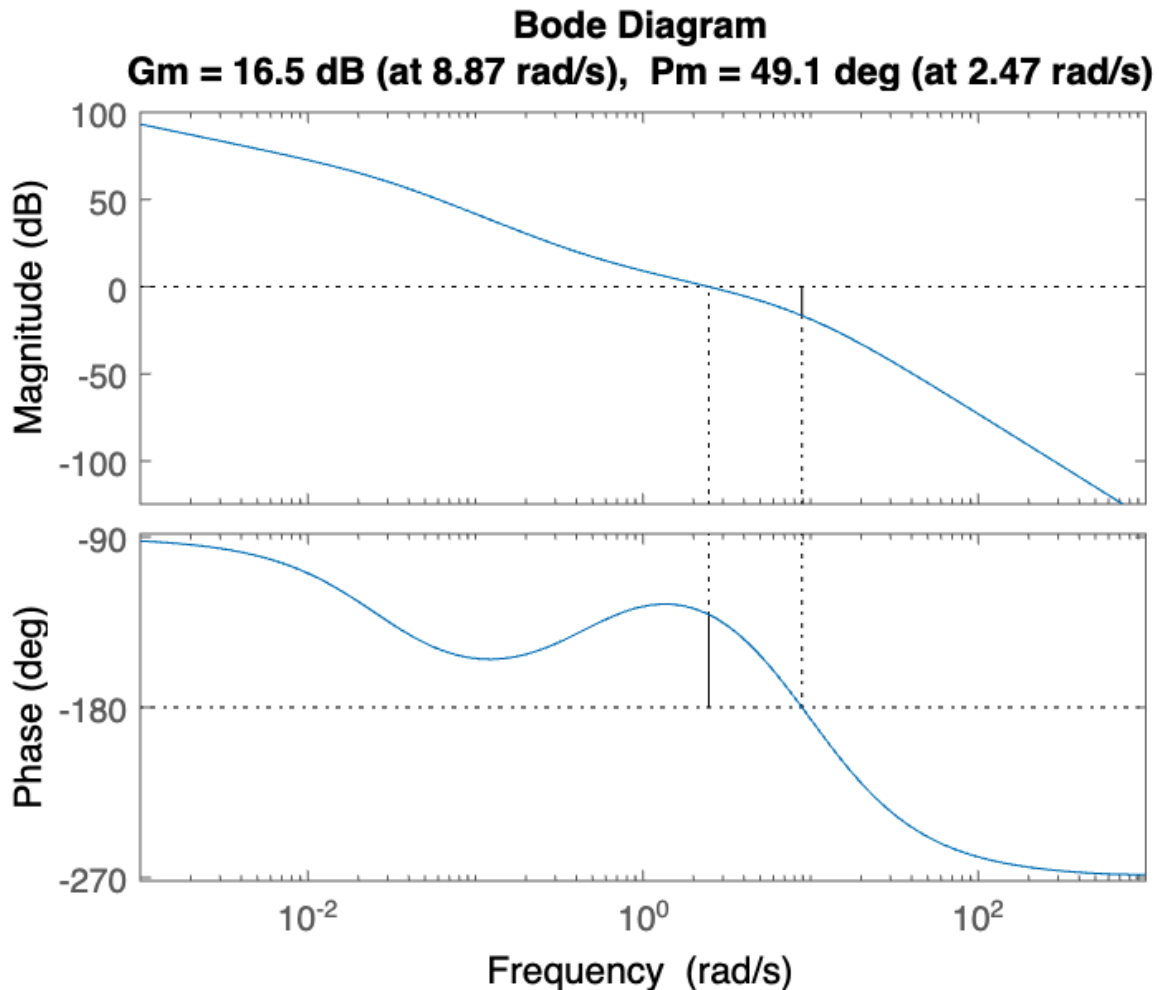
Com todos esses valores em mão, podemos agora montar nosso controlador atraso de fase:

```
syms jw
g_atrasos = k1*(1+jw*alfa1*tau1)/(1+jw*tau1);
[num,den]=numden(g_atrasos);
g_atraso = tf(sym2poly(num),sym2poly(den))
```

$$g_{\text{atraso}} = \frac{4.261e31 \text{ s} + 2.028e31}{3.714e32 \text{ s} + 1.014e31}$$

- 3) Vamos multiplicar a FT da planta pela FT do controlador Atraso de fase, obtido no passo anterior para obter  $G_{ma} = g_{\text{atraso}} * g_{\text{planta}}$  e determinar a MF usando esta FT em malha aberta.

```
%% Etapa 3 - Multiplicar atraso pela planta
g_planta_atrasada = g_atraso*g_planta;
[Gm,Pm1,Wgm,Wpm]=margin(g_planta_atrasada);
```



Margem de Ganho: 6.690666, Largura de banda: 2.470658  
 Margem de Fase: 49.108195, Frequência de Ganho: 8.874183

- 4) Considerando a FT em malha aberta  $G_{ma}$  obtida no passo 3, vamos projetar o controlador avanço de fase avanço:

$$G_{Avanço} = k_{avanço} \frac{1 + j\alpha_1\omega\tau_1}{1 + j\omega\tau_1} \text{ onde } k_{avanço} = \alpha_2 > 1$$

Escolhendo um  $K_{avanco}$  igual a 1 para atender a margem de fase especificada (60 graus) considerando uma folga de 10 graus na MF, vamos achar que o  $\alpha_2$  é:

```
% Etapa 4 - Projetar o controlador do avanço de fase
k_avanco = 1;
```

```
FaseGma = 70 - Pm1; % margem de fase especificada 60, +10 de folga
syms alfa2
equacao=(alfa2-1)/(2*sqrt(alfa2))==tand(FaseGma);
alfa2 = double(solve(equacao))
```

```
alfa2 = 2.1085
```

Com alfa2 em mãos, podemos calcular o módulo dele:

```
%% Calcular mod usando a formula
mod = -10*log10(alfa2)
```

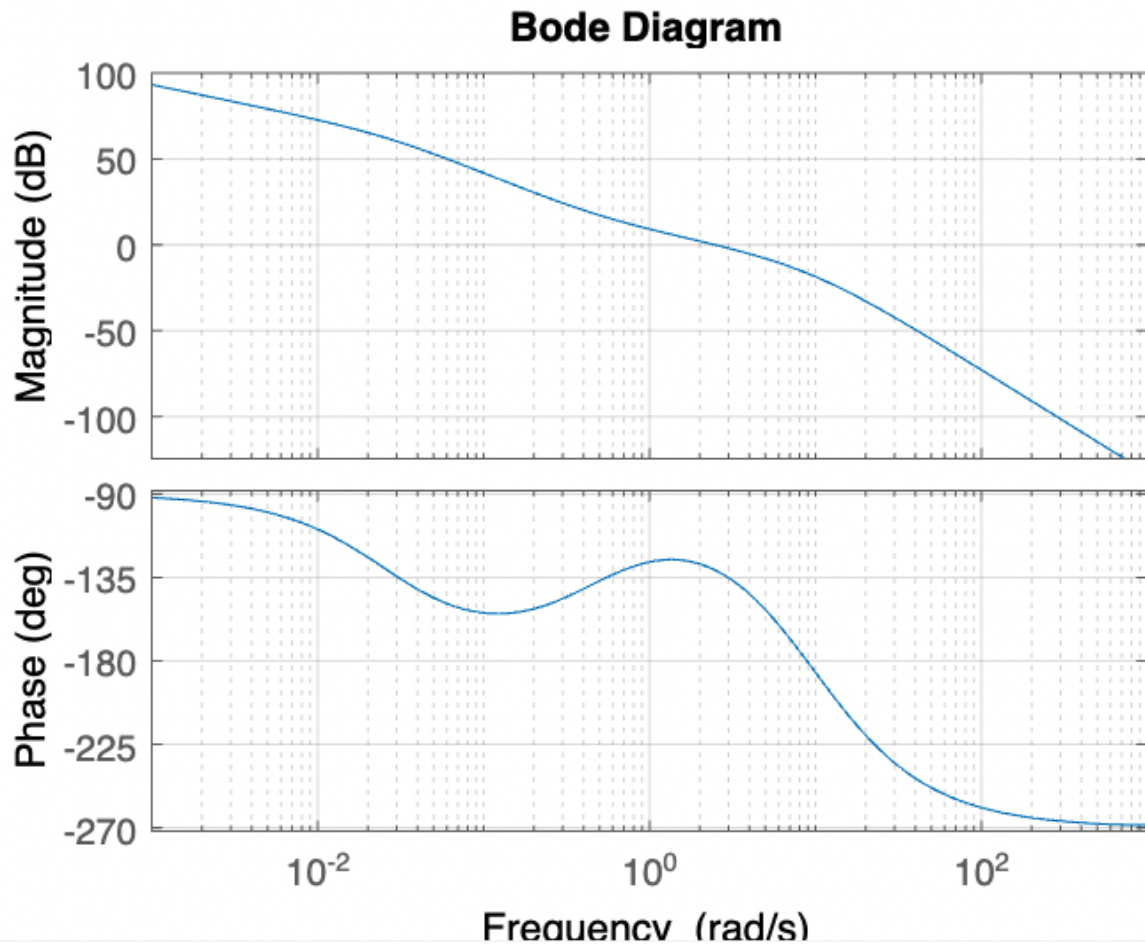
```
mod = -3.2398
```

E tanto pelo código abaixo, quanto analisando o diagrama de bode, conseguimos obter o valor da frequência:

```
bode(g_planta_atrasada)
[mag,phase,wout] = bode(g_planta_atrasada);
mag = squeeze(mag);

wm = interp1(20*log10(mag), wout, mod);
```

```
wm = 3.3566
```



Com a frequência de 3.35 rad/s encontrada, podemos encontrar o valor de tau2:

```
tau2 = 1/(wm*sqrt(alfa2));
1/(alfa2*tau2) % freq de corte do zero do compensador
1/(tau2) % freq de corte do polo do compensador
```

tau2 = 0.2052

Com os dados obtidos, podemos montar nosso controlador de avanço:

```
g_avancos = k_avanco*(1+jw*alpha2*tau2)/(1+jw*tau2);
[num,den]=numden(g_avancos);
g_avanco = tf(sym2poly(num),sym2poly(den));
```

g\_avanco =

$$\frac{4.387e30 \text{ s} + 1.014e31}{2.081e30 \text{ s} + 1.014e31}$$

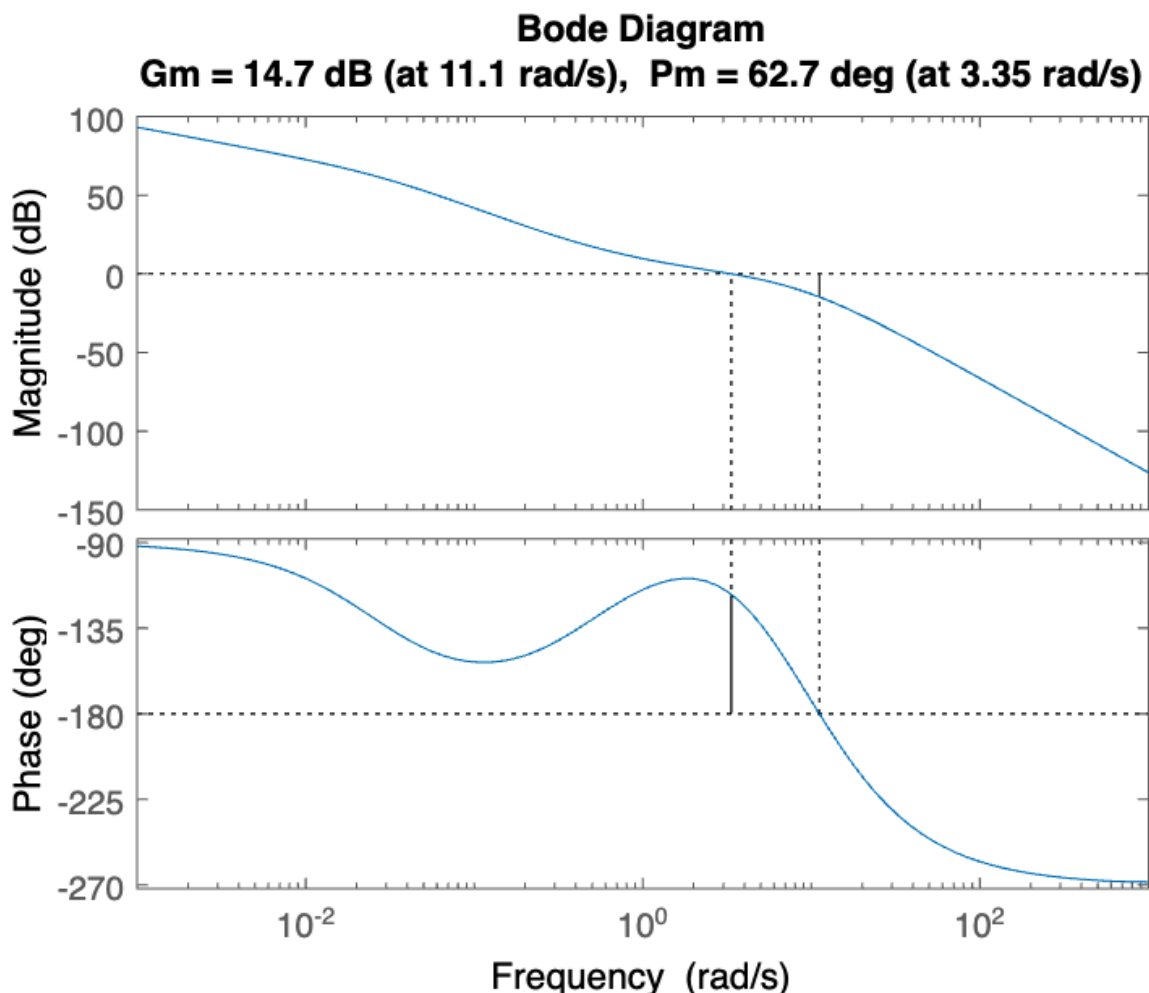
- 5) Agora vamos verificar se as especificações de MF e Largura de Banda da resposta em frequência em malha aberta de  $C_c * G_p = G_{atraso} * G_{avanco} * G_p$  foram atendidas.

```
nova_gma = g_avanco * g_atraso * g_planta;  
[Gm, Pm, Wgm, Wpm] = margin(nova_gma);  
margin(nova_gma)
```

Nossa nova Gma obtida com o controlador atraso-avanço é:

$$\frac{3.739e65 s^2 + 1.042e66 s + 4.114e65}{7.728e62 s^5 + 1.886e64 s^4 + 1.416e65 s^3 + 3.334e65 s^2 + 8.999e63 s}$$

E seu gráfico de bode:



Como é possível ver no diagrama de bode, a Margem de fase encontrada foi de 62.7 graus, sendo atendida. A largura de banda encontrada (Wpm) foi de 3.35 rad/s.

Os requisitos do projeto foram atendidos! Mas como a questão pede a maior largura de banda possível, vamos criar uma tabela com a variação de 30 a 40 graus feita no passo 2, abstraindo todos os passos anteriores e indo para o resultado final, visto que os cálculos e procedimentos são os mesmos.

MF (passo 2)	MF (final)	Largura de Banda (final)
30	61.320663	3.703089
31	61.615449	3.627795
32	61.906997	3.553993
33	62.181454	3.483504
34	62.447099	3.415283
35	62.705373	3.348859
36	62.950793	3.284814
37	63.198903	3.221308
38	63.439985	3.159419
39	63.674272	3.099082
40	63.905389	3.039752

Como é possível ver na tabela acima, o melhor projeto se encontra quando usamos 30 graus no passo 2, visto que a margem de fase final ficou mais próxima dos 60 graus e a largura de banda foi a maior dentre o intervalo de 30 a 40.

Escolhemos então, para o nosso projeto, um controlador atraso-avanço com as seguintes especificações:

$$k_1 = 2$$

$$\alpha_1 = 0.0623$$

$$\tau_1 = 30.3192$$



g\_atraso =

$$\frac{4.787e30 s + 2.535e30}{3.843e31 s + 1.268e30}$$

alpha2 = 2.3115

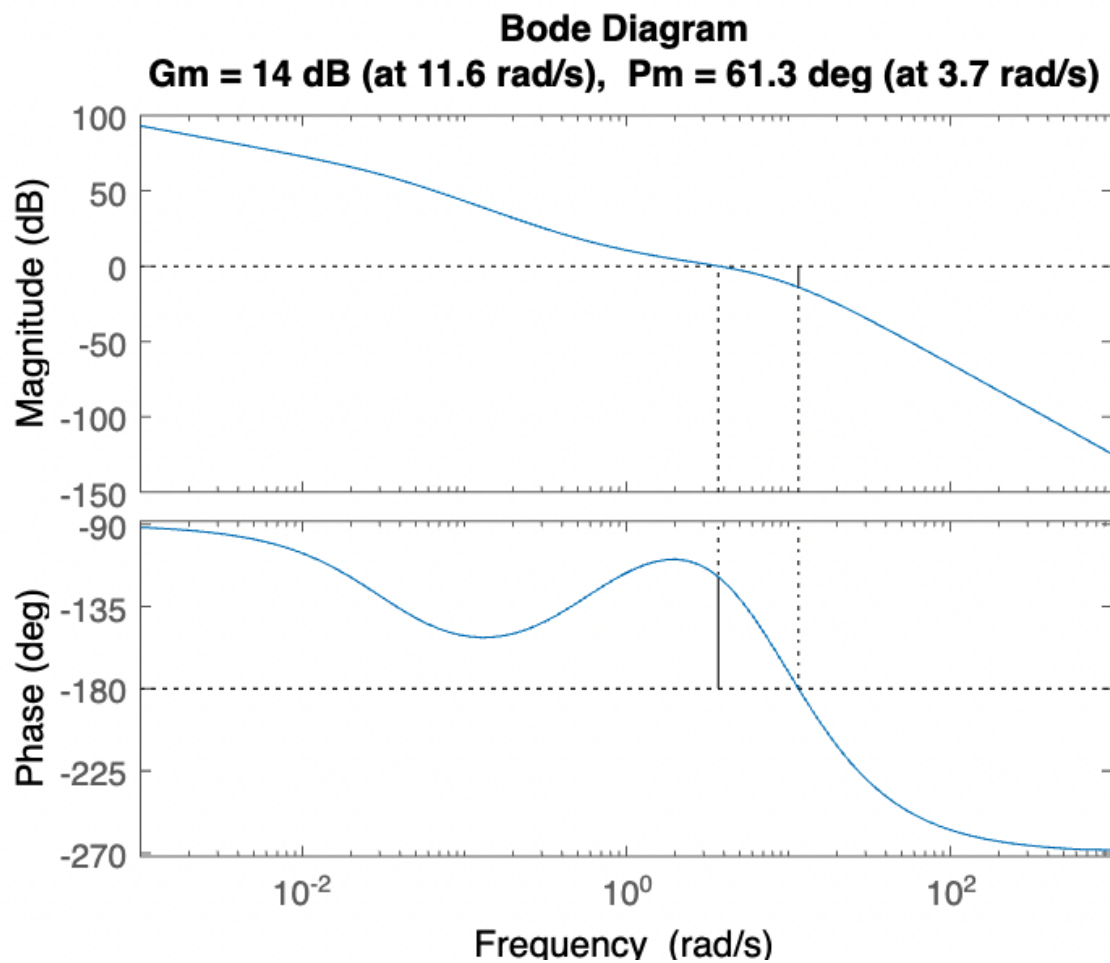
tau2 = 0.1773

g\_avanco =

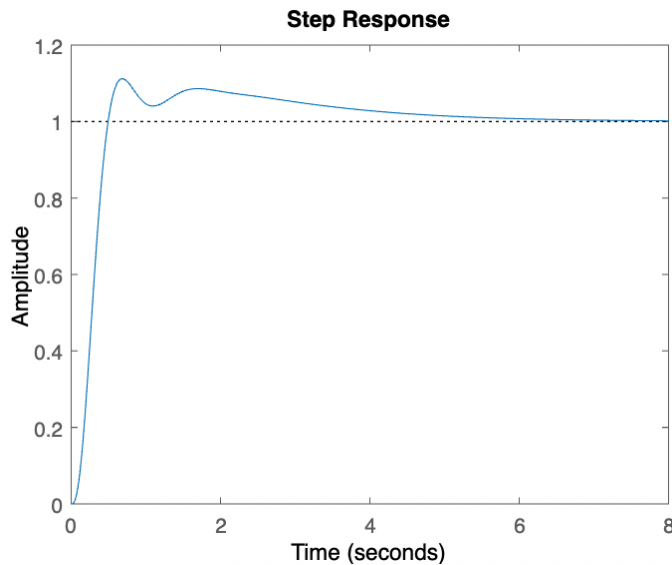
$$\frac{4.157e30 s + 1.014e31}{1.798e30 s + 1.014e31}$$

Nova Gma com controlador:

$$\frac{3.98e64 s^2 + 1.182e65 s + 5.142e64}{6.912e61 s^5 + 1.74e63 s^4 + 1.371e64 s^3 + 3.455e64 s^2 + 1.125e63 s}$$



E a resposta ao degrau:



RiseTime: 0.3071  
 TransientTime: 4.5467  
 SettlingTime: 4.5467  
 SettlingMin: 0.9074  
 SettlingMax: 1.1121  
 Overshoot: 11.2057  
 Undershoot: 0  
 Peak: 1.1121  
 PeakTime: 0.6838

Os passos foram os mesmos feitos com 35 graus no passo 2, só que agora com 30 graus. Por isso apresentamos apenas os resultados parciais e finais.

## Projeto 1.2.ii

Nesse projeto vamos fazer a mesma aplicação do atraso-avanço que o anterior, mas agora com o com atraso de transporte de 0,01 segundos.

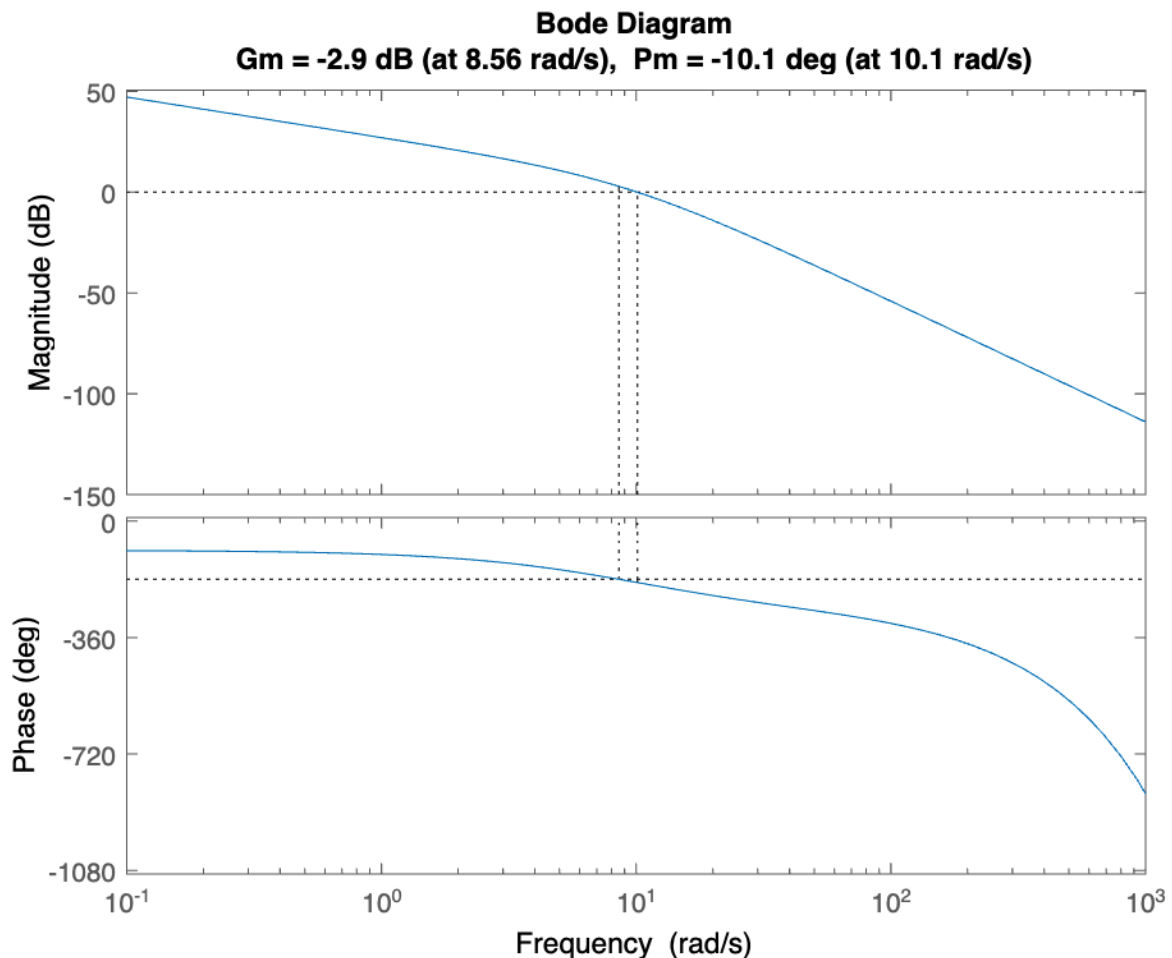
Não vamos precisar recalculer o erro em regime e o de distúrbio novamente, pois o ganho será igual já que o atraso de transporte só influencia na fase da malha. Portanto, nossos erros para essa questão é:

$ess = 0.0219$   
 $essd = 0.5000$

Nosso sistema com o atraso de transporte ficará:

```

syms s
T = 0.01; % Tempo
N = 4; % Numero do grupo
num = 500*N;
den = conv([1 0], conv([1 3+N], [1 25/sqrt(N)]));
g_planta = tf(num, den, 'InputDelay', T);
margin(g_planta)
  
```



Assim como feito no projeto atraso-avanço anteriormente, serão repetidos aqui. Com isso conseguimos obter os valores dos alfas, taus, além dos controladores de avanço, atraso e a planta final. Usando também 30 graus no passo 2, temos que os resultados obtidos foram:

`alfa1 = 0.0594`

`tau1 = 33.8100`

Com uma margem de fase de 30 graus (passo 2) obtemos nosso controlador atraso de fase:

`g_atraso =`

$$\frac{2.036e31 \, s + 1.014e31}{1.714e32 \, s + 5.071e30}$$

`alfa2 = 2.3148`

$\tau_2 = 0.1837$

Com uma MF de 70 graus, ou seja, 60 graus + 10 graus de folga temos que nosso controlador avanço de fase é:

$$g_{\text{avanco}} = \frac{3.449e31 s + 8.113e31}{1.49e31 s + 8.113e31}$$

Com isso, nossa planta inicial com atraso de transporte junto com o controlador de atraso-avanço tem:

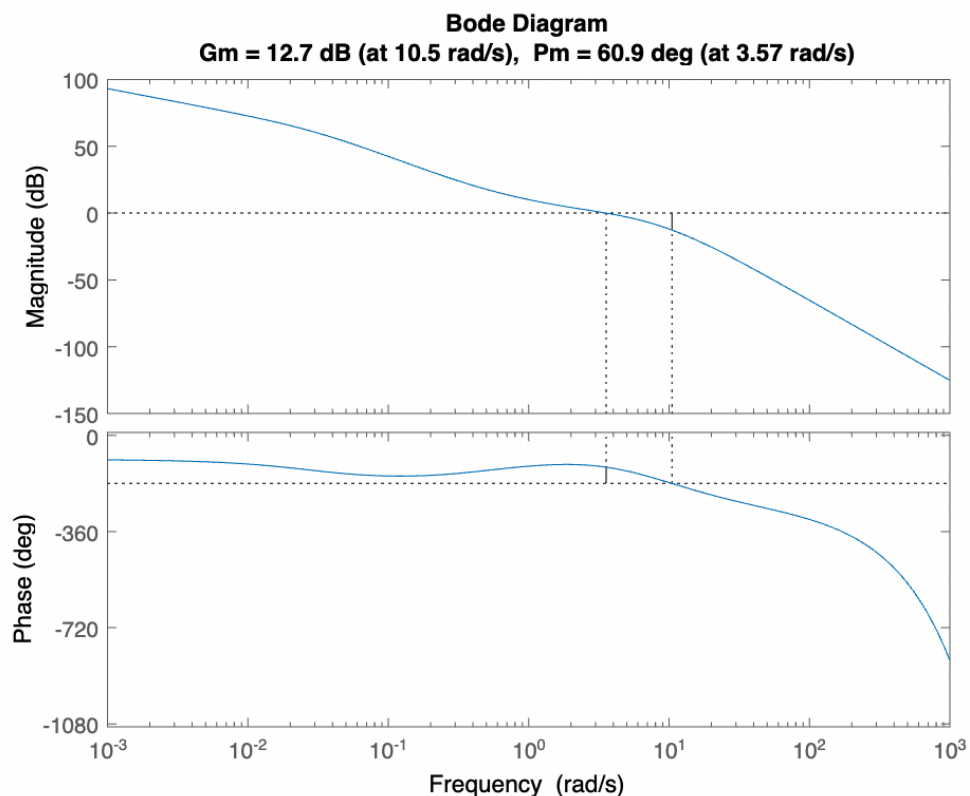
**Margem de fase: 60.910040**

**Largura de banda: 3.568297**

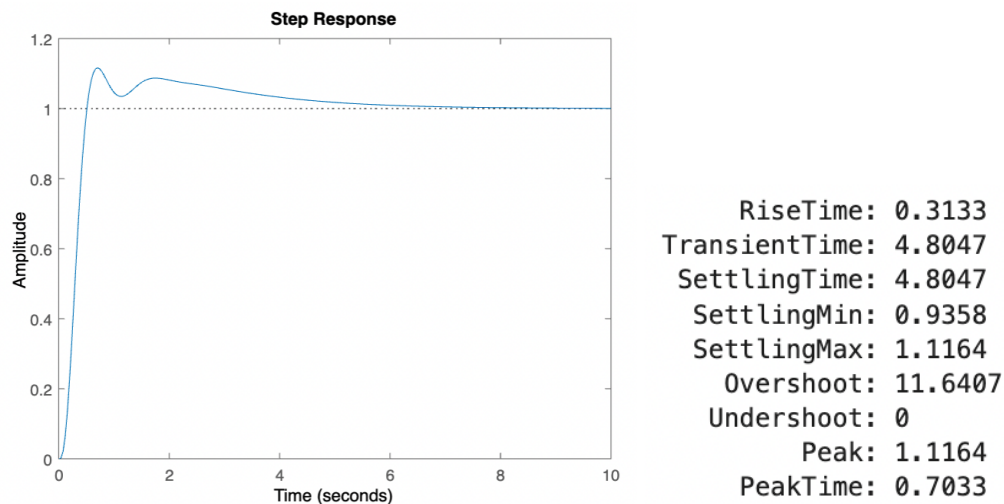
E portanto, nossa nova Gma é:

$$\exp(-0.01s) * \frac{1.405e66 s^2 + 4.003e66 s + 1.646e66}{2.555e63 s^5 + 6.38e64 s^4 + 4.966e65 s^3 + 1.232e66 s^2 + 3.6e64 s}$$

Seu diagrama de bode:



E a resposta ao degrau do sistema é:



Com isso conseguimos obter um controlador que atende as especificações desejadas. Percebemos que a presença do atraso de transporte na nossa planta não gerou muita influência no resultado final, sendo possível projetar o controlador com as mesmas especificações feitas sem a presença dele.

Nota-se que a largura de banda alcançada é menor que o controlador atraso-avanço obtido na planta sem atraso, isso muito se justifica no uso do valor mínimo de  $K_1$  para atender o erro em distúrbio, porque dessa forma quanto maior o valor de  $k_1$ , o sistema irá ter um ganho, aumentando a largura de banda (cortar o -3db mais para frente), assim deixando a largura de banda desde o primeiro passo baixa.

A análise com o atraso de transporte é importante, visto que os projetos reais, em sua maioria, costumam apresentá-lo.

## Projeto 1.3

1.3 - Compare as respostas à entrada degrau (sobressinal, tempo de subida) e o erro em regime às entradas rampa para os dois sistemas com os controladores obtidos nos itens 1.1 e 1.2 para  $T=0.0$  s.

Fazendo uma comparação entre o controlador PID feito na questão 1.1 e o controlador de atraso-avanço na questão 1.2, vimos que em relação ao erro em regime e o erro ao distúrbio para o PID é zero, visto que este adiciona um polo a origem do sistema, e como a

planta já tinha um polo acabou se tornando do tipo 2. Por ser do tipo 2 uma entrada rampa terá os dois erros iguais a zero, mas já no atraso-avanço não é (apesar de apresentar valores bem baixos). Na tabela abaixo é possível comparar alguns valores dos dois controladores.

Controlador	MF	MP	tr	ta	Largura de banda	ess	essd
PID	60.6	6.67%	0.1238	2.54	14.5 rad/s	0	0
atraso-avanço	60.9	11.64%	0.31	4.8	3.56 rad/s	0.0437	1

Podemos concluir analisando a tabela que esses controladores são parecidos em alguns quesitos, ficando bem próximos no caso da margem de fase e tempo de subida e overshoot. Porém, em uma situação real, o controlador escolhido seria o avanço-atraso, pois o PID ele acaba não feito na prática. Outro motivo da escolha do controlador atraso-avanço é que ele possui um menor largura de banda, portanto possui um maior tempo de subida.