PROGETTAZIONE DI UN REGOLATORE NEL TEMPO DISCRETO

Si vuole progettare il sistema di controllo per un attuatore idraulico. L'ingresso di controllo è la tensione di alimentazione dell'elettrovalvola, mentre l'uscita da controllare è la forza esercitata dall'attuatore sull'ambiente.
La funzione di trasferimento del sistema è la seguente:

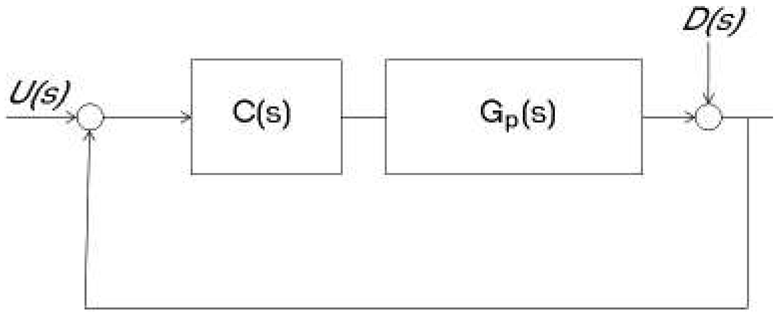
$$G_p(s) = \frac{2 \cdot 10^4}{(1+0.035s)(1+40s)(\frac{s^2}{2.5 \cdot 10^4} + \frac{35}{2.5 \cdot 10^4}s + 1}$$

Sull'attuatore agisce una forza esterna modellata come disturbo sull'uscita.
Si progetti un regolatore che garantisca il soddisfacimento delle seguenti specifiche:

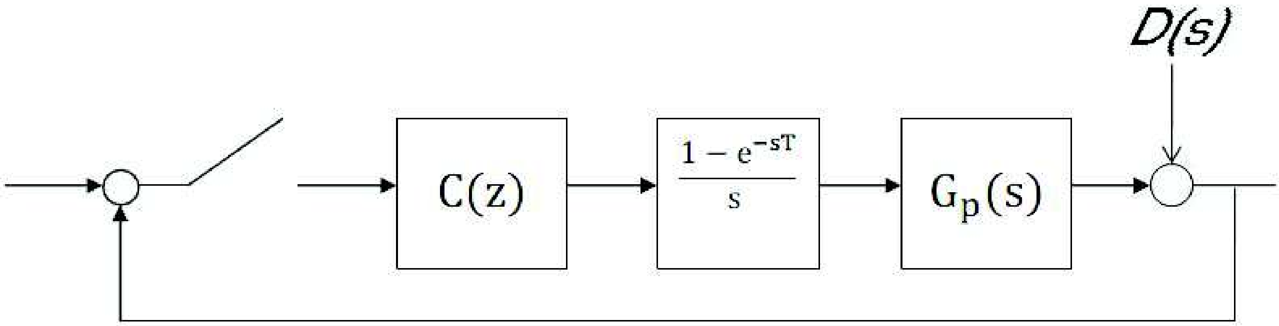
- Errore nullo con disturbo sull'uscita a gradino di ampiezza 500*N*
- Pulsazione di attraversamento: $\omega_c > 10$ *rad/s*
- Margine di fase: $M_\varphi \geq 80°$

1.Schema equivalente e tempo di campionamento

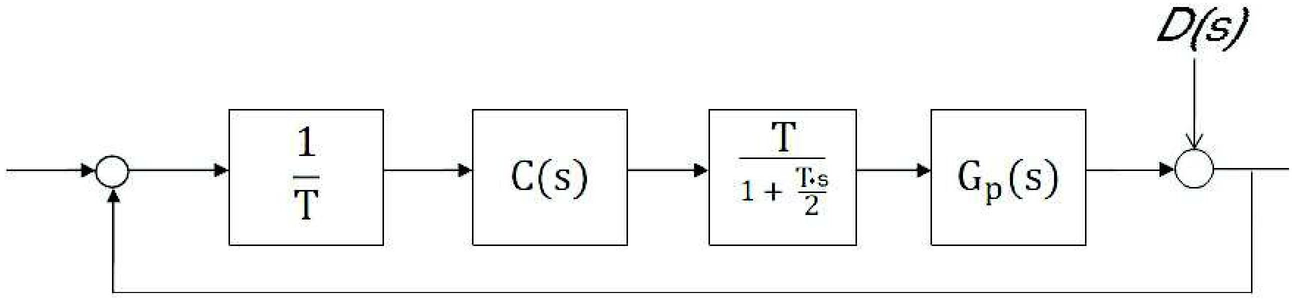
Lo schema di controllo analogico è il seguente:



Osserviamo il sistema di controllo nel tempo continuo e cominciamo a fare gli opportuni passaggi per la progettazione nel tempo discreto del controllore C(z) seguendo lo schema:



Noto il tempo di campionamento è possibile, tramite l'approssimazione di Padè, ricavare lo schema equivalente tempo continuo che segue:



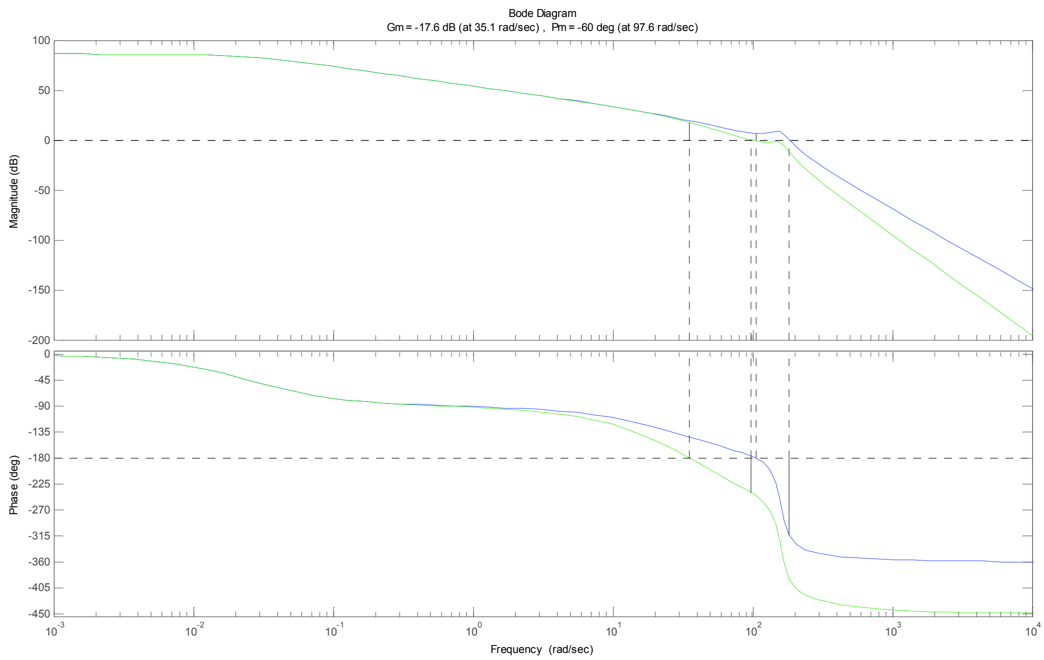
Trovo il tempo di campionamento T_c secondo la formula:

$$T_C = \frac{2\pi}{3\,\omega_t}$$

$\omega_t = 150$ ed è stata scelta in base alla seguente: $10 \cdot \omega_B < \omega_t < 20 \cdot \omega_B$ (dove ω_B è la banda passante=10 r/s).Trovo che T_c=0.014 sec
Faccio la serie con tra G e T e troviamo:

```
>> Gt=series(T,G)

Transfer function:
                        20000
---------------------------------------------------------------------
1.17e-006 s^5 + 0.0001304 s^4 + 0.03399 s^3 + 2.293 s^2 + 40.05 s + 1
```

facendo un confronto tra G e G_t si evince che a 10 r/sec c'è uno sfasamento di 12°



2.Analisi delle specifiche statiche

Per avere un errore nullo a un disturbo di tipo gradino, l'impianto deve essere di '*tipo 1*', quindi il controllore deve introdurre un polo nell'origine, pertanto avrà una funzione di trasferimento del tipo:

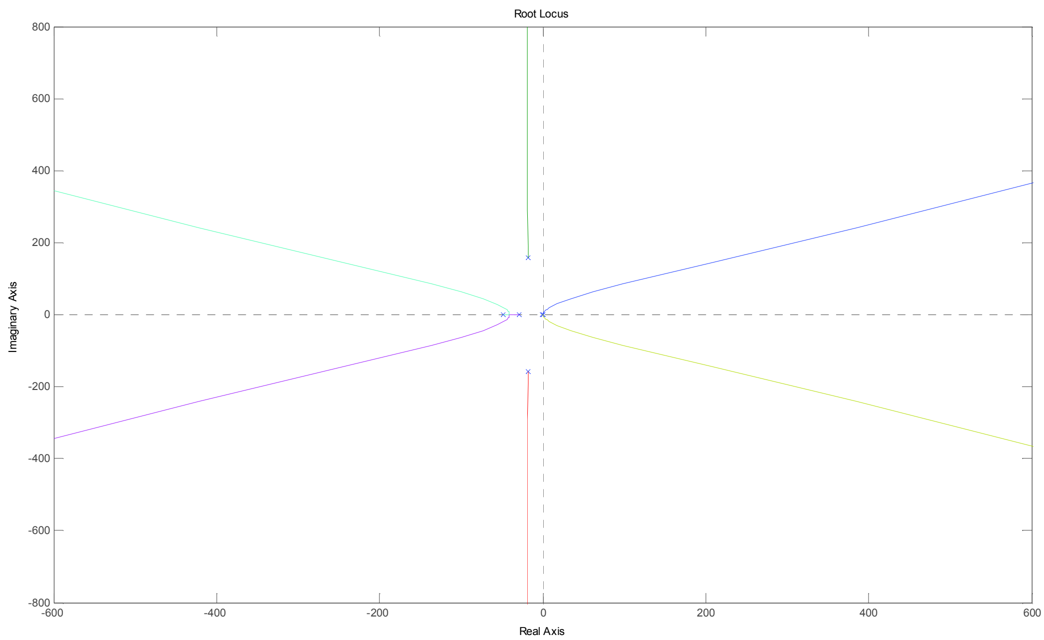
$$C = \frac{K}{s}$$

Per il momento poniamo K=1 e facciamo la serie tra la G_r e la C:

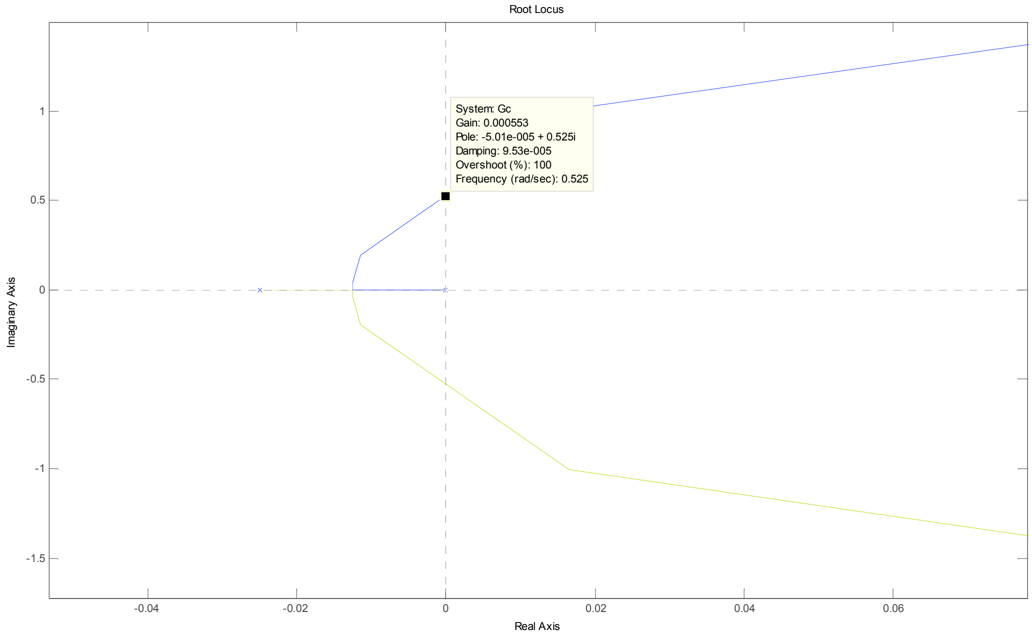
$$C = \frac{1}{s}$$

```
>> Gc=series(C,Gt)

Transfer function:
                               20000
----------------------------------------------------------------------
1.17e-006 s^6 + 0.0001304 s^5 + 0.03399 s^4 + 2.293 s^3 + 40.05 s^2 + s
```

faccio il luogo delle radici



Facciamo uno zoom in prossimità dell'origine e vediamo per quali valori del guadagno, i poli stanno sul semipiano sinistro. Si scopre che per avere stabilità, il guadagno statico deve essere inferiore a 0.000553:



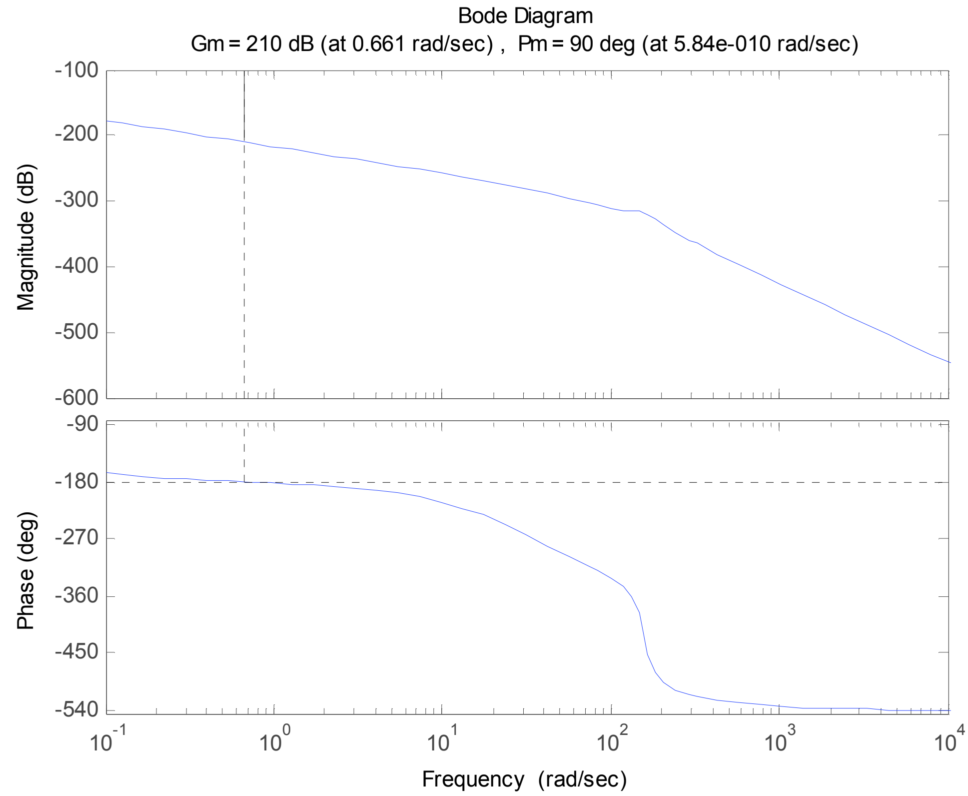
quindi scelgo k=2.92*10^(-14)

```
>> Gr=2.92*10^-14*Gc

Transfer function:
                          5.84e-010
----------------------------------------------------------------------
1.17e-006 s^6 + 0.0001304 s^5 + 0.03399 s^4 + 2.293 s^3 + 40.05 s^2 + s

>> zpk(Gr)

Zero/pole/gain:
                  0.00049897
-----------------------------------------------------
s (s+47.85) (s+28.56) (s+0.02501) (s^2 + 35s + 2.5e004)
```

ANALISI DELLE SPECIFICHE DINAMICHE

Ottenuta la funzione di trasferimento, andiamo a vedere se le specifiche dinamica (in termini di pulsazione di attraversamento e margine di fase) rispettano i requisiti richiesti.
Per fare ciò, andiamo a visualizzare i diagrammi di Bode

```
>> bode(Gr)
```



Come risulta dal grafico, il sistema presenta una pulsazione di attraversamento bassa rispetto a quella richiesta.
Si progetta allora una rete ad anticipo del tipo:

$$R = \frac{1 + \tau \cdot s}{1 + \alpha\tau \cdot s}$$

$$\alpha = \frac{1 - sen\Delta\phi}{1 + sen\Delta\phi} = 0.217 \qquad \tau = \frac{M - \cos\Delta\phi}{\omega_c^* \cdot sen\Delta\phi}$$

Dove $M = \frac{1}{10^{\frac{261dB}{20}}}$ rappresenta il modulo della funzione di trasferimento dell'impianto alla pulsazione $\omega_c^* = 12rad/s$

Inseriamo questa rete correttrice in *Matlab*:

```
n1 =

  1.0e+013 *

    1.1200    0.0000

>> d1=[2.32*10^12 1]
```

```
d1 =

  1.0e+012 *

    2.3200    0.0000

>> tf1=tf(n1,d1)

Transfer function:
1.12e013 s + 1
------------------
2.32e012 s + 1
```

E si calcola la serie con la G_c

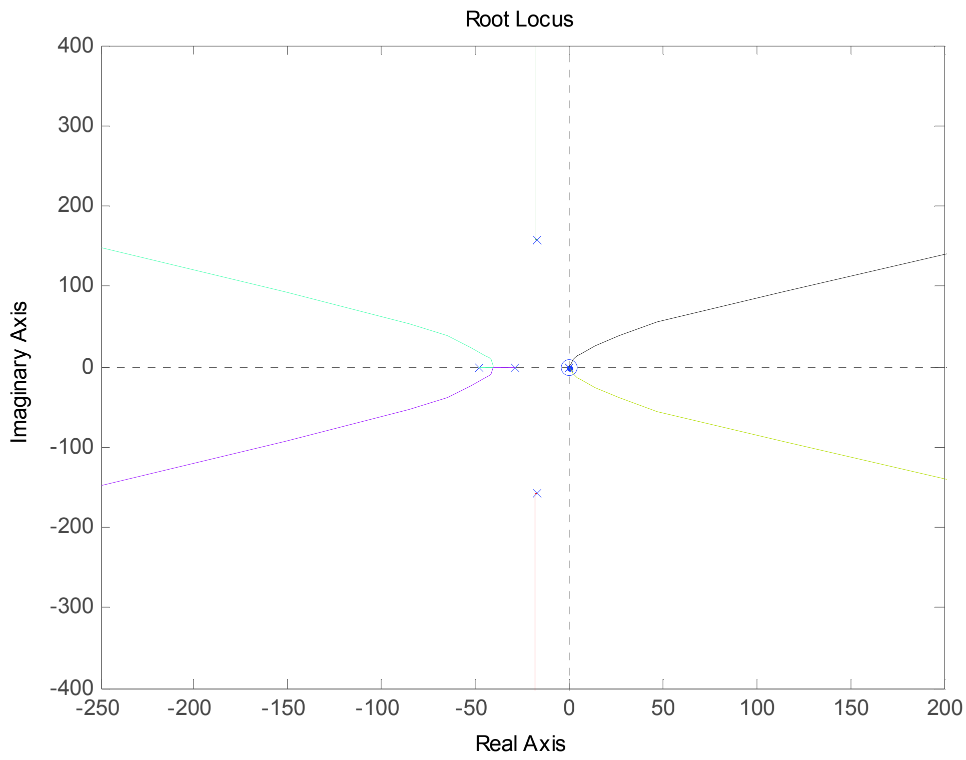
```
>> Gr1=series(tf1,Gr)

Transfer function:

                        6541 s + 5.84e-010
----------------------------------------------------------------------
2.715e006 s^7 + 3.026e008 s^6 + 7.886e010 s^5 + 5.319e012 s^4

                                  + 9.291e013 s^3 + 2.32e012 s^2 + s
```

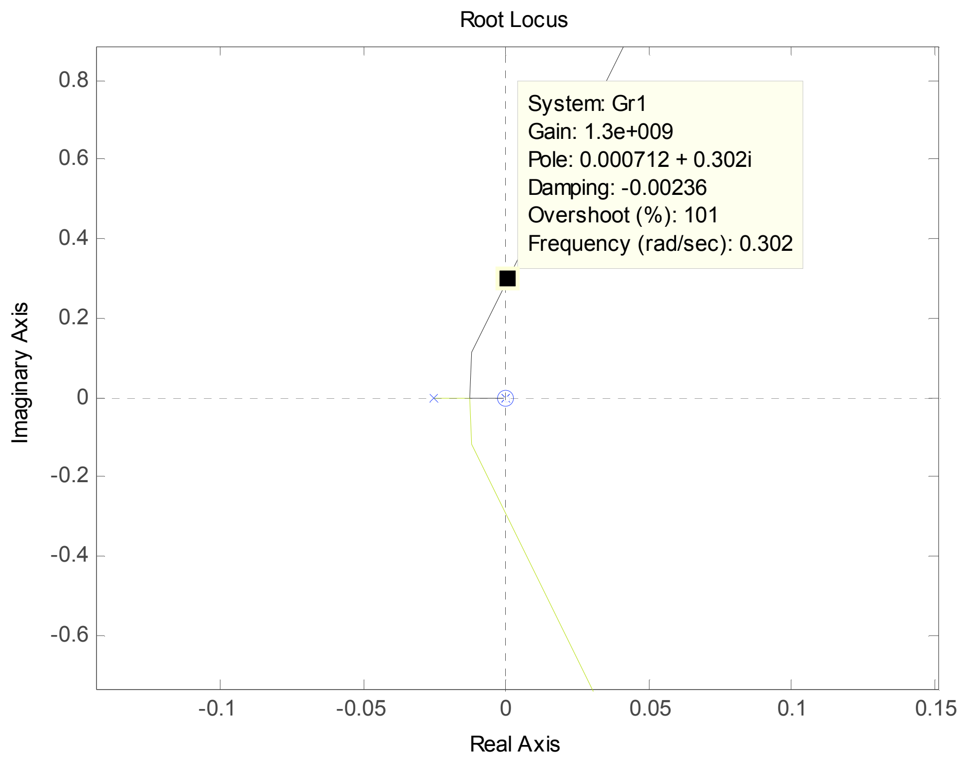
a questo punto vediamo se è possibile aumentare il K (senza perdere stabilità e guadagnando in pulsazione di attraversamento). Si trova che:

```
>> zpk(Gr1)

Zero/pole/gain:
                    0.0024088 s
---------------------------------------------------------
s^2 (s+47.85) (s+28.56) (s+0.02501) (s^2 + 35s + 2.5e004)
```

Attraverso *rlocus(Gr1)* si scopre che è possibile portare il guadagno fino a 1.3*10^9 senza perdere stabilità:

```
>> rlocus(Gr1)
```



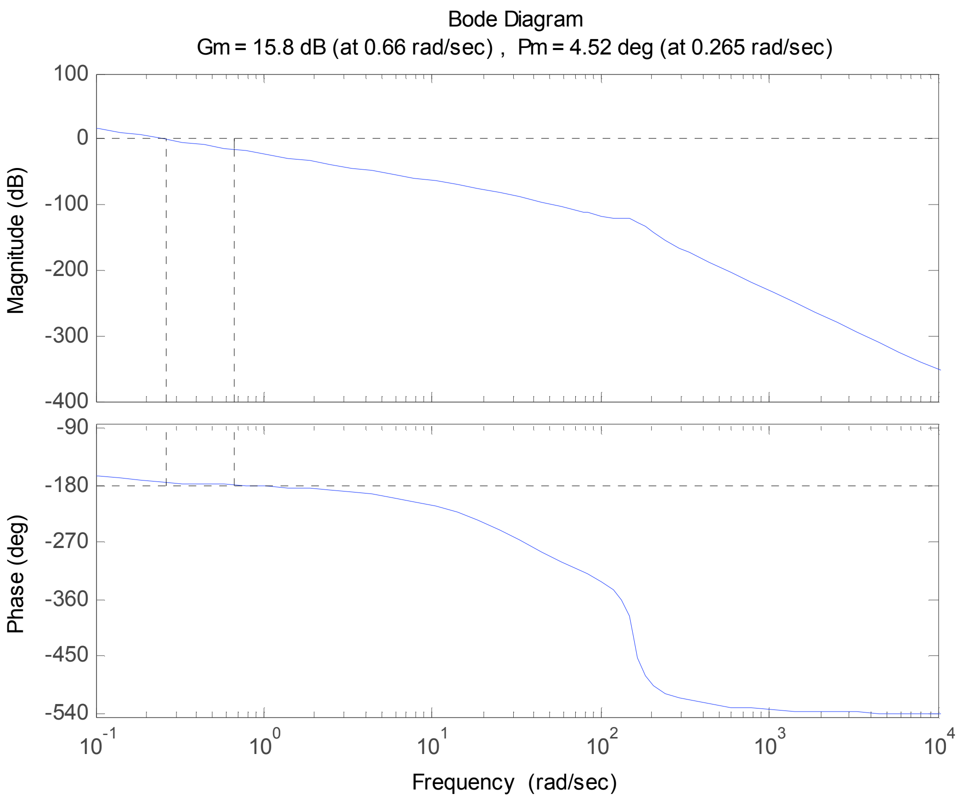



```
>> 1.21*10^9/0.0024088

ans =

  5.0232e+011
```

cioè è possibile moltiplicare per un k=5.0232e+011 senza perdere stabilità:

Per evitare di perdere troppo margine di fase, si opta per un K=10^9 e si ottiene:

```
>> Gr2=10^9*Gr1

Transfer function:

                    6.541e012 s + 0.584
----------------------------------------------------------------------
2.715e006 s^7 + 3.026e008 s^6 + 7.886e010 s^5 + 5.319e012 s^4

                    + 9.291e013 s^3 + 2.32e012 s^2 + s

>> margin(Gr2)
```

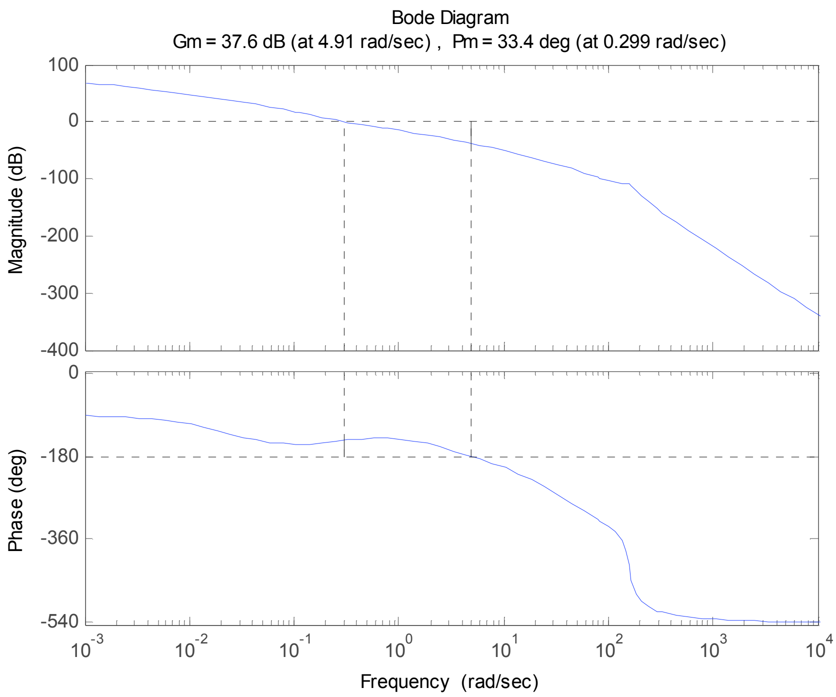


Ancora non va bene, dobbiamo progettare almeno un'altra rete anticipatrice per aumentare il margine di fase, che ora è calato a causa del K inserito, e la pulsazione di attraversamento.

Il procedimento è simile al precedente, cioè prendiamo sempre $\alpha = 0.217$ e $\tau = \frac{1}{\omega^* \sqrt{0.217}} = 2.72$

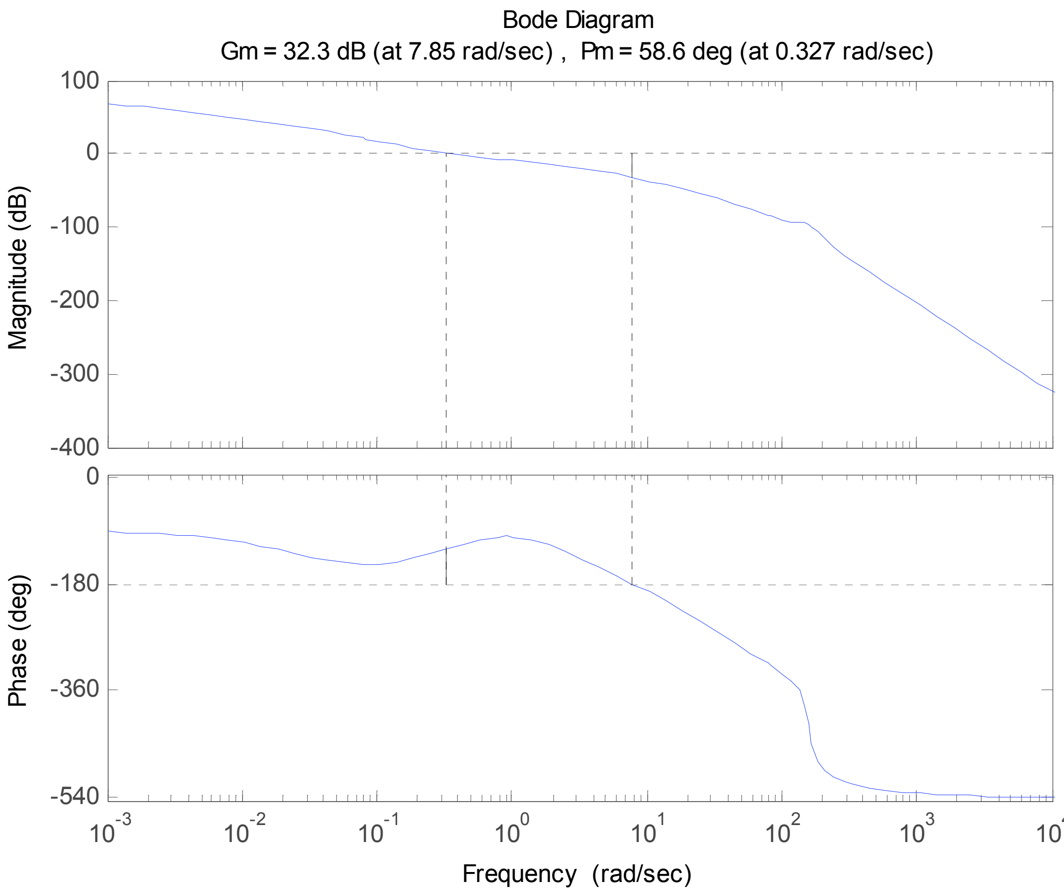
dove ω^* è la pulsazione alla quale il modulo di G_{r2} vale -19dB (cioè $\frac{\alpha}{2}$ in decibel):

```
d1 =

   0.5600    1.0000

>> n1=[2.72 1]

n1 =

   2.7200    1.0000

>> R=tf(n1,d1)

Transfer function:
2.72 s + 1
----------
0.56 s + 1

>> Gr3=series(R,Gr2)

Transfer function:

              1.779e013 s^2 + 6.541e012 s + 0.584
----------------------------------------------------------------------
1.521e006 s^8 + 1.722e008 s^7 + 4.447e010 s^6 + 3.057e012 s^5

                + 5.735e013 s^4 + 9.421e013 s^3 + 2.32e012 s^2 + s


>> margin(Gr3)
```



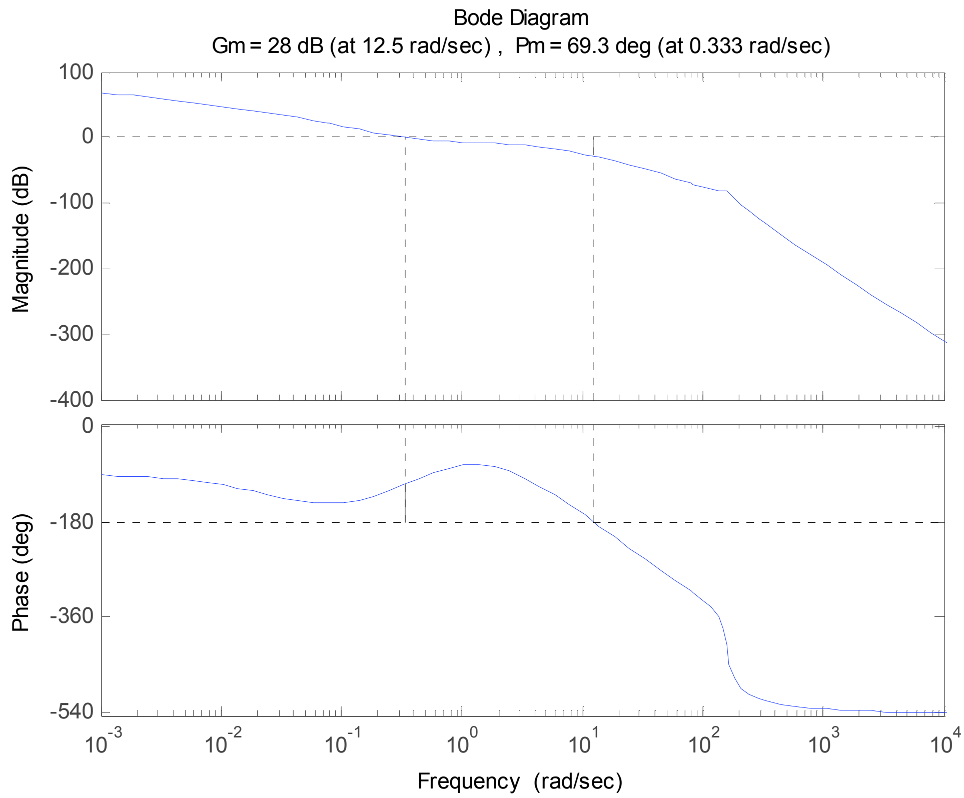
Abbiamo guadagnato margine di fase, ma non basta (bisogna arrivare almeno ad 80), quindi progetto un'altra rete ad anticipo allo stesso modo di prima, con $\alpha = 0.217$ e $\tau = 1.83$

```
>> R=tf(n1,d1)

Transfer function:
1.83 s + 1
----------
0.37 s + 1

>> Gr4=series(R,Gr3)

Transfer function:

        3.256e013 s^3 + 2.976e013 s^2 + 6.541e012 s + 0.584
----------------------------------------------------------------------
5.626e005 s^9 + 6.522e007 s^8 + 1.662e010 s^7 + 1.176e012 s^6

     + 2.428e013 s^5 + 9.221e013 s^4 + 9.507e013 s^3 + 2.32e012 s^2

                                     + s

>> margin(Gr4)
```



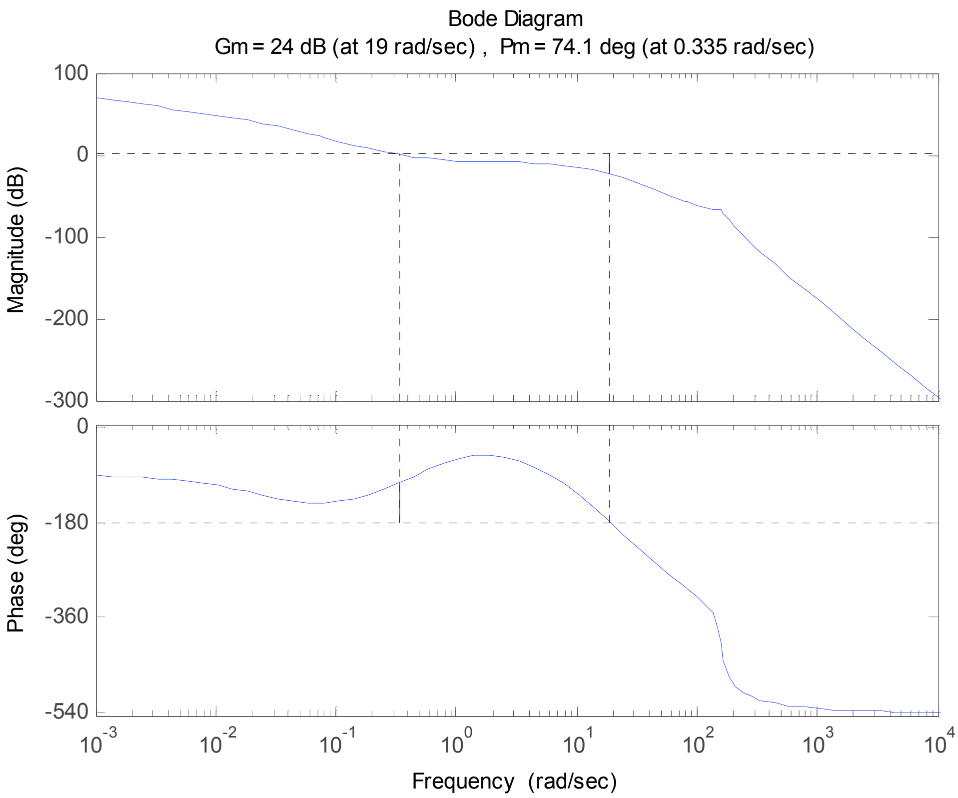
Progettiamo un’altra rete ad anticipo con gli stessi criteri:

```
>> R=tf(n1,d1)

Transfer function:
0.68 s + 1
----------
0.14 s + 1

>> Gr5=series(R,Gr4)

Transfer function:

  2.214e013 s^4 + 5.279e013 s^3 + 3.421e013 s^2 + 6.541e012 s + 0.584
----------------------------------------------------------------------
7.877e004 s^10 + 9.694e006 s^9 + 2.393e009 s^8 + 1.812e011 s^7

      + 4.574e012 s^6 + 3.719e013 s^5 + 1.055e014 s^4 + 9.539e013 s^3

                                 + 2.32e012 s^2 + s

>> margin(Gr5)
```



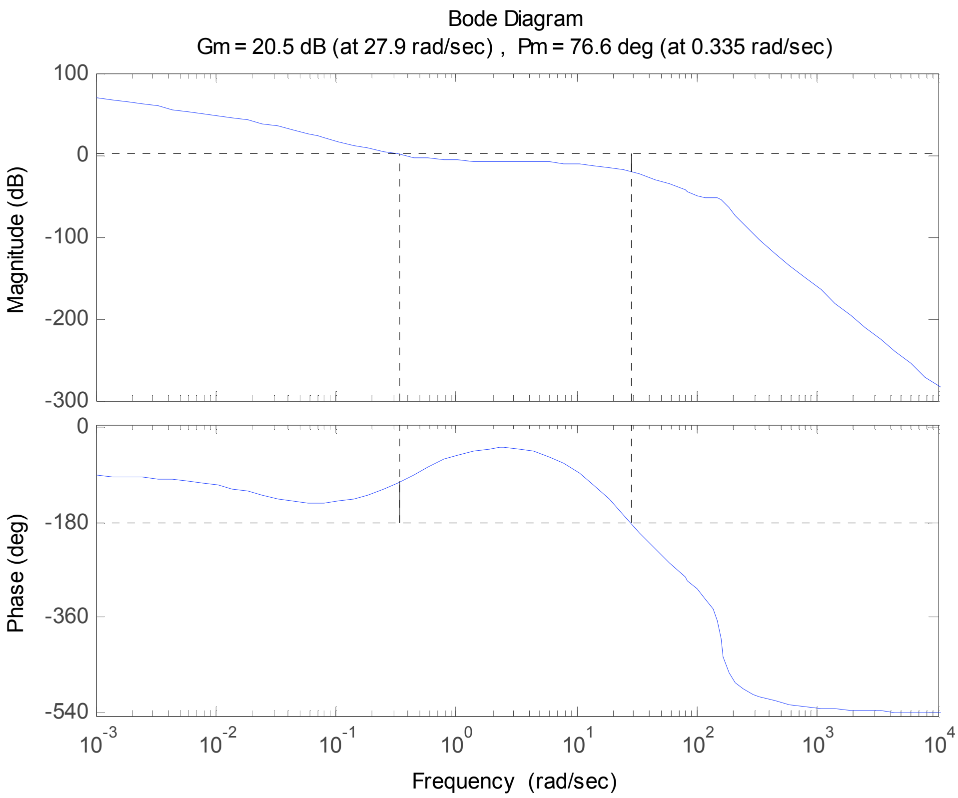
Altra rete ad anticipo:

```
>> R=tf(n1,d1)

Transfer function:
0.31 s + 1
-----------
0.064 s + 1

>> Gr6=series(R,Gr5)

Transfer function:

6.863e012 s^5 + 3.851e013 s^4 + 6.34e013 s^3 + 3.624e013 s^2 + 6.541e012 s

                                                 + 0.584

-------------------------------------------------------------------------
5041 s^11 + 6.992e005 s^10 + 1.628e008 s^9 + 1.399e010 s^8 + 4.74e011 s^7

     + 6.954e012 s^6 + 4.394e013 s^5 + 1.116e014 s^4 + 9.554e013 s^3

                                         + 2.32e012 s^2 + s


>> margin(Gr6)
```

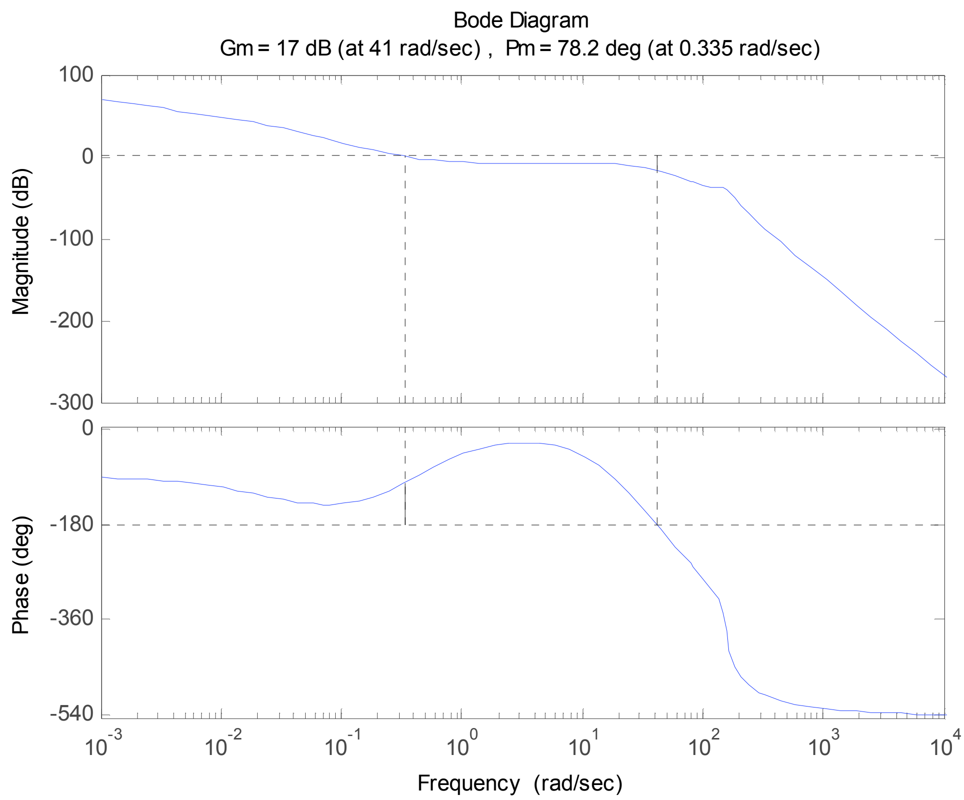


Ripetiamo gli stessi passaggi inserendo altre due reti:

```
>> R=tf(n1,d1)

Transfer function:
0.16 s + 1
-----------
0.032 s + 1

>> Gr7=series(R,Gr6)

Transfer function:

1.098e012 s^6 + 1.302e013 s^5 + 4.865e013 s^4 + 6.92e013 s^3 + 3.728e013 s^2

                                        + 6.541e012 s + 0.584

--------------------------------------------------------------------------------
161.3 s^12 + 2.741e004 s^11 + 5.91e006 s^10 + 6.105e008 s^9 + 2.916e010 s^8

      + 6.965e011 s^7 + 8.36e012 s^6 + 4.751e013 s^5 + 1.147e014 s^4

                              + 9.561e013 s^3 + 2.32e012 s^2 + s


>> margin(Gr7)
```

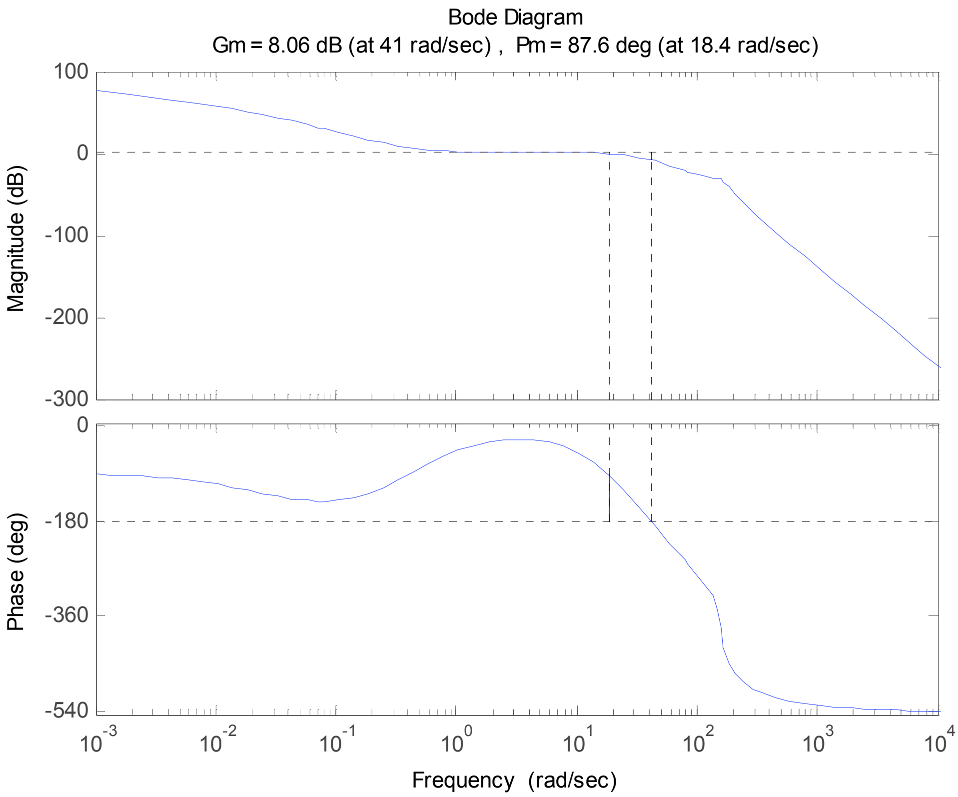


```
>> R=tf(n1,d1)

Transfer function:
 0.08 s + 1
-----------
0.017 s + 1

>> Gr8=series(R,Gr7)

Transfer function:

1.098e011 s^7 + 2.4e012 s^6 + 1.789e013 s^5 + 5.557e013 s^4 + 7.293e013 s^3

                              + 3.794e013 s^2 + 6.541e012 s + 0.584

-------------------------------------------------------------------------
 2.742 s^13 + 627.4 s^12 + 1.279e005 s^11 + 1.629e007 s^10 + 1.106e009 s^9

        + 4.1e010 s^8 + 8.386e011 s^7 + 9.168e012 s^6 + 4.946e013 s^5

                    + 1.163e014 s^4 + 9.565e013 s^3 + 2.32e012 s^2 + s


>> margin(Gr8)
```



Possiamo dire di essere quasi giunti al nostro obiettivo perché a questo punto per aumentare la pulsazione ci basterà moltiplicare per un K piccolo (sempre stando attenti a non perdere in stabilità, osservando il luogo delle radici): scegliamo K=2.8 convenientemente ed otterremo:

```
>> GrTOT=(2.8*Gr8)

Transfer function:

 3.075e011 s^7 + 6.721e012 s^6 + 5.009e013 s^5 + 1.556e014 s^4

              + 2.042e014 s^3 + 1.062e014 s^2 + 1.831e013 s + 1.635

------------------------------------------------------------------------
2.742 s^13 + 627.4 s^12 + 1.279e005 s^11 + 1.629e007 s^10 + 1.106e009 s^9

      + 4.1e010 s^8 + 8.386e011 s^7 + 9.168e012 s^6 + 4.946e013 s^5

               + 1.163e014 s^4 + 9.565e013 s^3 + 2.32e012 s^2 + s


>> margin(GrTOT)
```



che soddisfa le specifiche richieste.

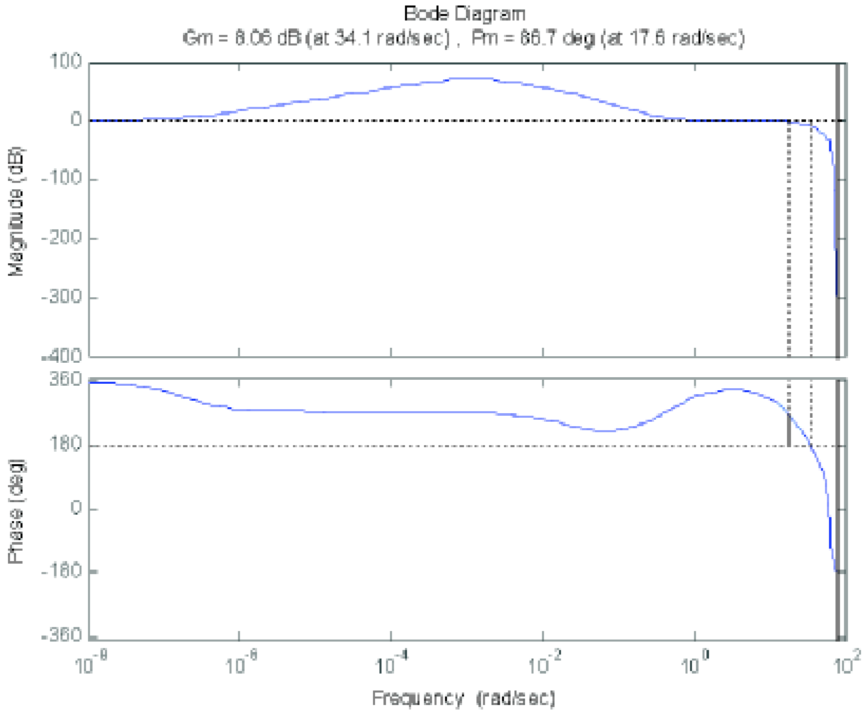
3.Discretizzazione del controllore

La discretizzazione del regolatore può essere effettuata attraverso la tecnica Tustin, in questo modo poi verificheremo se anche nel tempo discreto le specifiche sono rispettate:

```
>> Gd=c2d(GrTOT,0.014,'tustin')

Transfer function:

0.05498 z^13 - 0.01171 z^12 - 0.318 z^11 + 0.08353 z^10 + 0.7682 z^9

     - 0.239 z^8 - 0.9919 z^7 + 0.3553 z^6 + 0.7222 z^5 - 0.2913 z^4

                - 0.2811 z^3 + 0.1255 z^2 + 0.04568 z - 0.02227

----------------------------------------------------------------------
z^13 - 4.924 z^12 + 8.993 z^11 - 6.3 z^10 - 0.5229 z^9 + 0.9915 z^8

     + 4.387 z^7 - 6.484 z^6 + 3.82 z^5 - 1.082 z^4 + 0.1173 z^3

                    + 0.005565 z^2 - 0.001566 z + 1.451e-005
```

Sampling time: 0.014

Leggiamo i diagrammi di Bode:



Quindi il sistema soddisfa le specifiche richieste sia in termini di margine di fase chè in quelli di pulsazione di attraversamento (questi valori si discostano di poco rispetto a quelli osservati nel tempo continuo).
Andando a ricostruire il controllore C(s) nel tempo discreto dobbiamo fare la serie di tutte le reti correttrici introdotte e poi discretizzarle. Viene fuori che nel tempo continuo C(s) vale:

```
1.23e007 s^7 + 2.996e008 s^6 + 2.368e009 s^5 + 7.586e009 s^4 + 1.011e010 s^3

                       + 5.293e009 s^2 + 9.157e008 s + 8.176e-005

-----------------------------------------------------------------------------
2.343e006 s^8 + 2.749e008 s^7 + 1.057e010 s^6 + 1.65e011 s^5 + 1.072e012 s^4

                         + 2.745e012 s^3 + 2.32e012 s^2 + s
```

E nel tempo discreto:

```
>> Cz=c2d(C10,0.014)

Transfer function:

0.03846 z^7 - 0.2568 z^6 + 0.7348 z^5 - 1.167 z^4 + 1.112 z^3 - 0.635 z^2

                                 + 0.2014 z - 0.02736

-----------------------------------------------------------------------------
z^8 - 6.731 z^7 + 19.68 z^6 - 32.61 z^5 + 33.47 z^4 - 21.76 z^3 + 8.744 z^2

                                  - 1.981 z + 0.1935
```

Sampling time: 0.014

In definitiva:

$$\boxed{C(z) = \frac{0.03846z^7 - 0.2568z^6 + 0.7348z^5 - 1.167z^4 + 1.112z^3 - 0.635z^2 + 0.2014z - 0.02736}{z^8 - 6.731z^7 + 19.68z^6 - 32.61z^5 + 33.47z^4 - 21.76z^3 + 8.744z^2 - 1.981z + 0.1935}}$$

4.Risposta ad un ingresso di tipo gradino

Possiamo analizzare la risposta dell'apparato ad un segnale di tipo gradino, giusto per avere delle informazioni su quelle che sono le dinamiche di assestamento del sistema:

```
>>Gdf=feedback(Gd,1)

>>step(Gdf)
```