SOLUZIONE ***Prova MATLAB Tipo – B***

Esame di “FONDAMENTI DI AUTOMATICA” (9 CFU)

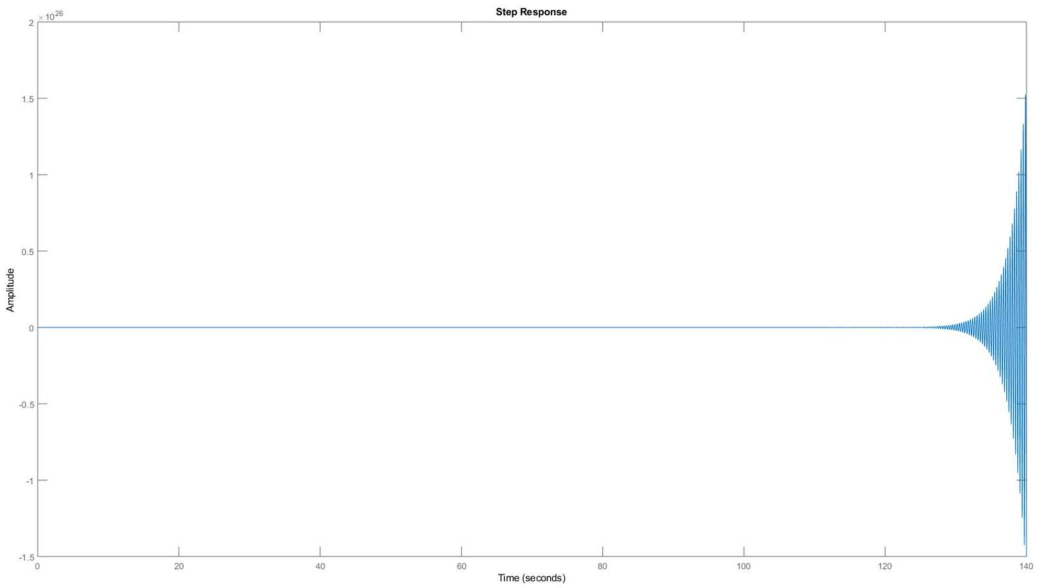
ESERCIZIO 1

```
%% Es 1-a: esecuzione dello script InitAutomaticaTipoB.m
%  ATTENZIONE a maiuscole/minuscole nel nome file
>> InitAutomaticaTipoB
A =

         0    1.0000   -1.0000

   -2.0000   -0.8000    0.8000

    2.0000    0.8000   -0.8000


B =

     0

   500

     0


C =

     0     0     1


D =

     0

%% Es 1-a Funzione di trasferimento
sys = ss(A,B,C,D);
G = tf(sys)

G =

               400 s + 1000
  -------------------------------
  s^3 + 1.6 s^2 + 4 s - 7.216e-33
```

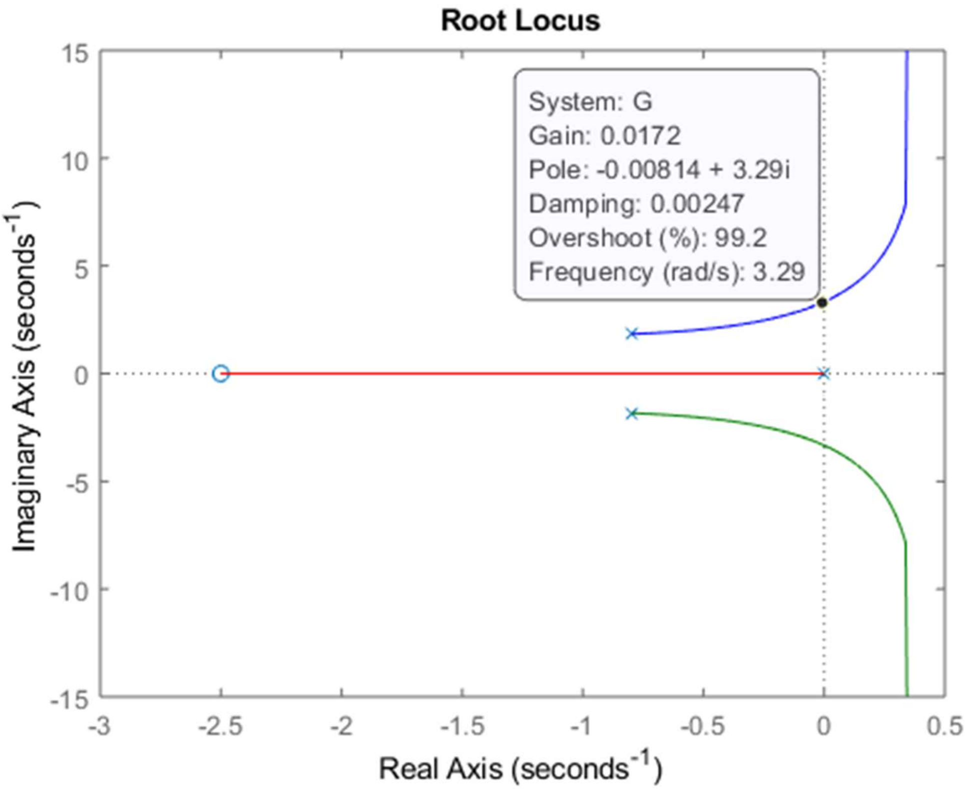
```
%% Es 1-b poli di G e autovalori di A
p = pole(G);
ev = eig(A);

p =

  -0.8000 + 1.8330i
  -0.8000 - 1.8330i
   0.0000 + 0.0000i
ev =

  -0.8000 + 1.8330i
  -0.8000 - 1.8330i
   0.0000 + 0.0000i

% poli e autovalori coincidono, quindi il sistema è
% completamente controllabile e osservabile
```

ESERCIZIO 2

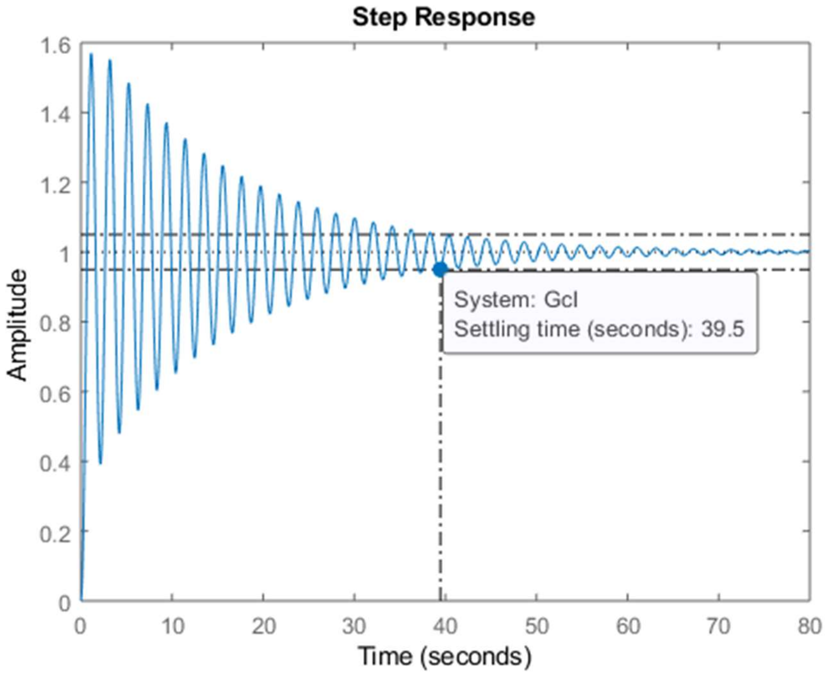
```
%% Es 2-a risposta al gradino ad anello chiuso

K=1;
Gcl = feedback(K*G,1);
figure, step(Gcl) % sistema instabile
```



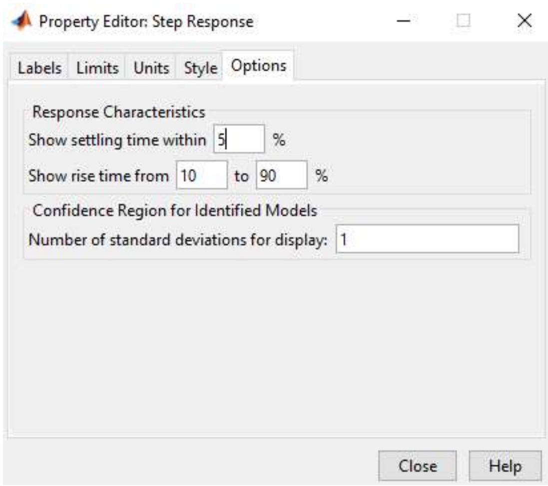
```
%% Es 2-b Calcolo del guadagno critico o limite per la
stabilità ad anello chiuso

% plot del luogo delle radici
figure, rlocus(G)
% valore selezionato dal grafico
Klim = 0.017;
```




```
%% Es 2-c risposta al gradino con guadagno 80% del
% guadagno limite

K1 = 0.8*Klim;

>> K1

K1 =

   0.0136

Gcl1 = feedback(K1*G,1);% FDT anello chiuso
figure, step(Gcl1); % Sistema stabile
```



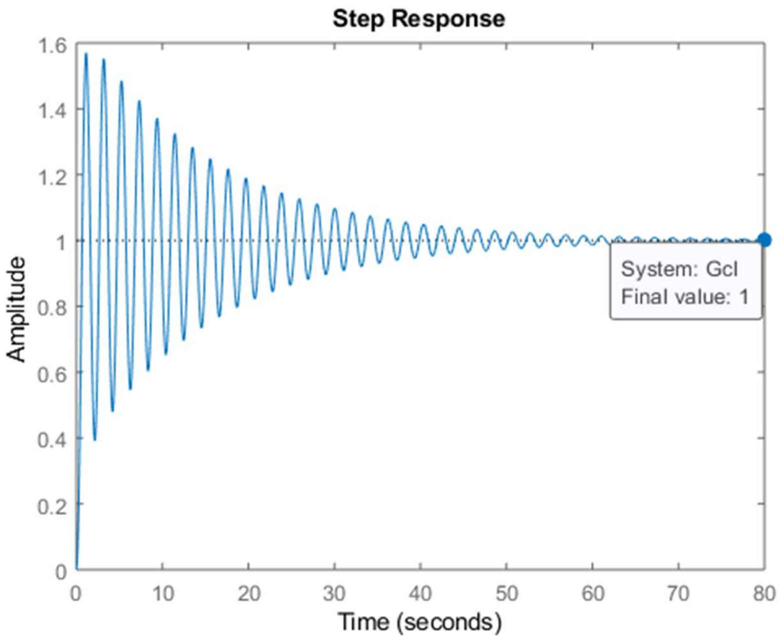
NOTA BENE: impostare la visualizzazione del tempo di assestamento al 5% tramite il menu ottenuto con mouse right-click sul plot della risposta:



Oppure tramite i comandi:

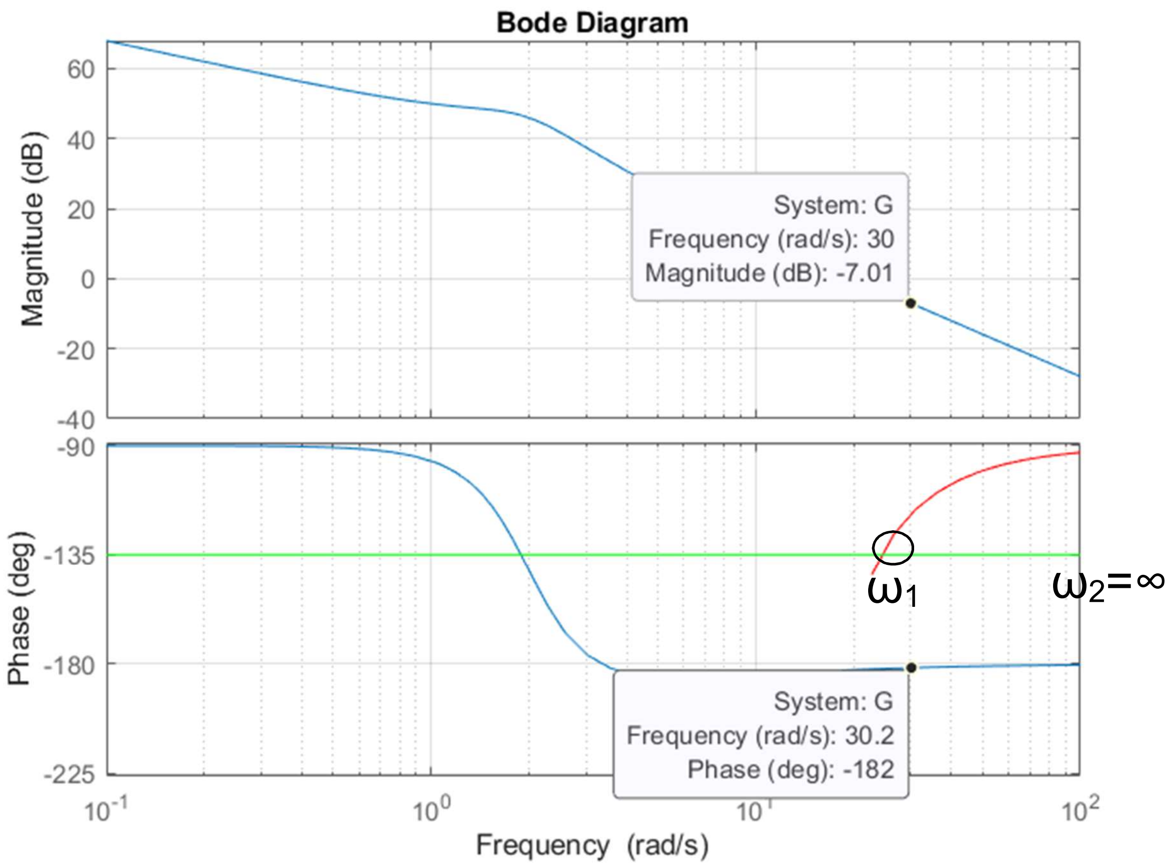
```
Popt=timeoptions;
Popt.SettleTimeThreshold=0.05;

figure,step(Gcl1,Popt)

%% Es 2-d considerazioni su errore a regime
p=pole(K1*G)
p =
  -0.8000 + 1.8330i
  -0.8000 - 1.8330i
   0.0000 + 0.0000i
```

```
% come evidenziato dalla caratteristica “Steady State”
% nel grafico ottenuto con step(), l'uscita tende a 1 al
% crescere di t, infatti il sistema K1*G è un sistema di
% tipo 1 (1 polo nell'origine p(3)=0). Per questo tipo di
% sistemi l'errore a regime nella risposta al gradino è
% nullo
```



ESERCIZIO 3

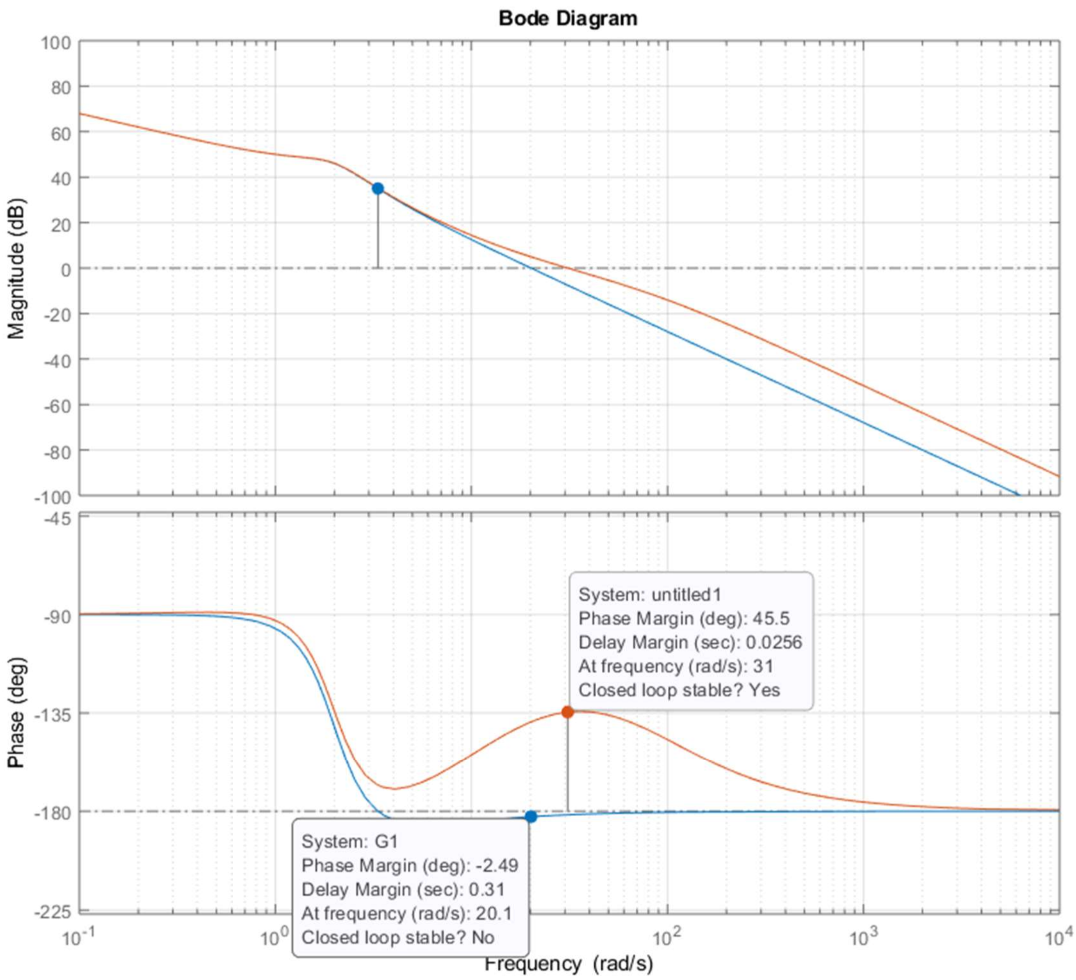
```
%% Es 3-a progetto della rete con formule d’inversione.

G1 = K*G;
Mf = 45;
leadNetDesignBode(G1,Mf);
% grafici per la verifica della realizzabilità della rete
```



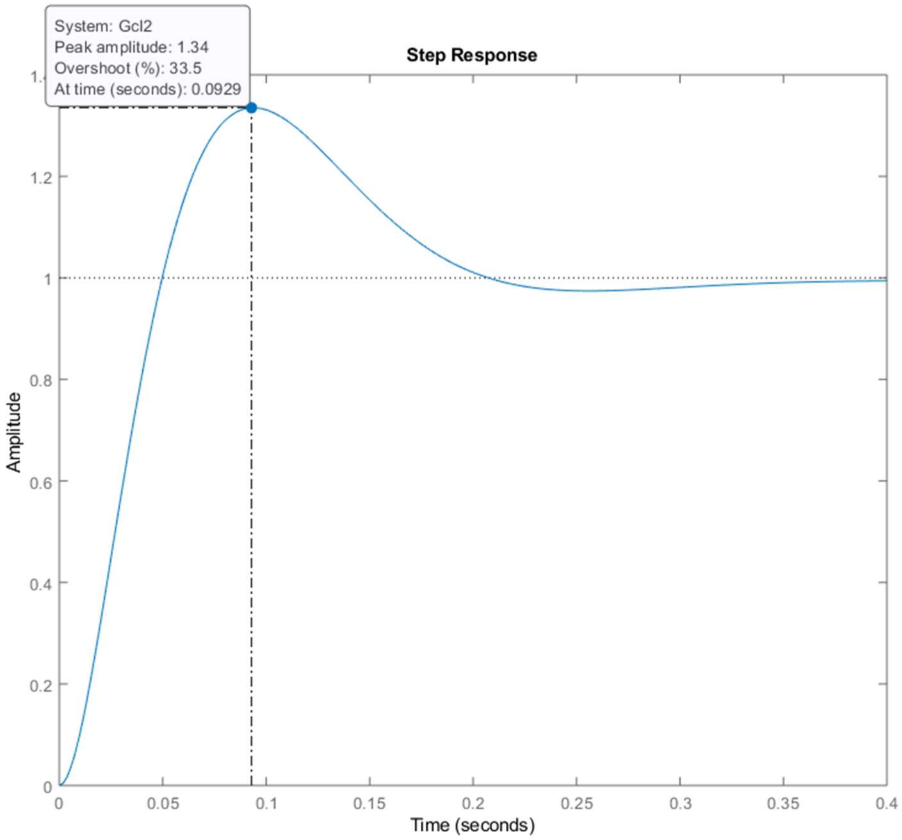
NOTA: condizioni di realizzabilità della rete ritardatrice rispettate tra ω_1 e ω_2, vedere al proposito i suggerimenti mostrati digitando **`help leadNetDesignBode`** (funzione Matlab scaricabile dalle pagine Classroom del corso di Fondamenti di Automatica)

```
% valori selezionati dal grafico
omega=30;
M = db2mag(7.35); % anticipatrice: amplifica
phi = 182-(180-Mf); % anticipatrice: anticipa la fase

tau1 = (M-cosd(phi))/(omega*sind(phi));
tau2 = (cosd(phi)-1/M)/(omega*sind(phi));

s=tf('s');
Gc=(1+tau1*s)/(1+tau2*s)
Gc =
   0.07515 s + 1
  -------------
  0.01153 s + 1
% verifica che sia anticipatrice:
alpha = tau2/tau1 % < 1
alpha =
    0.1534
```

```
%% Es 3-b diagrammi di bode e margini di fase
figure,bode(G1)
grid on
hold on
bode(Gc*G1) % verificare i margini
```



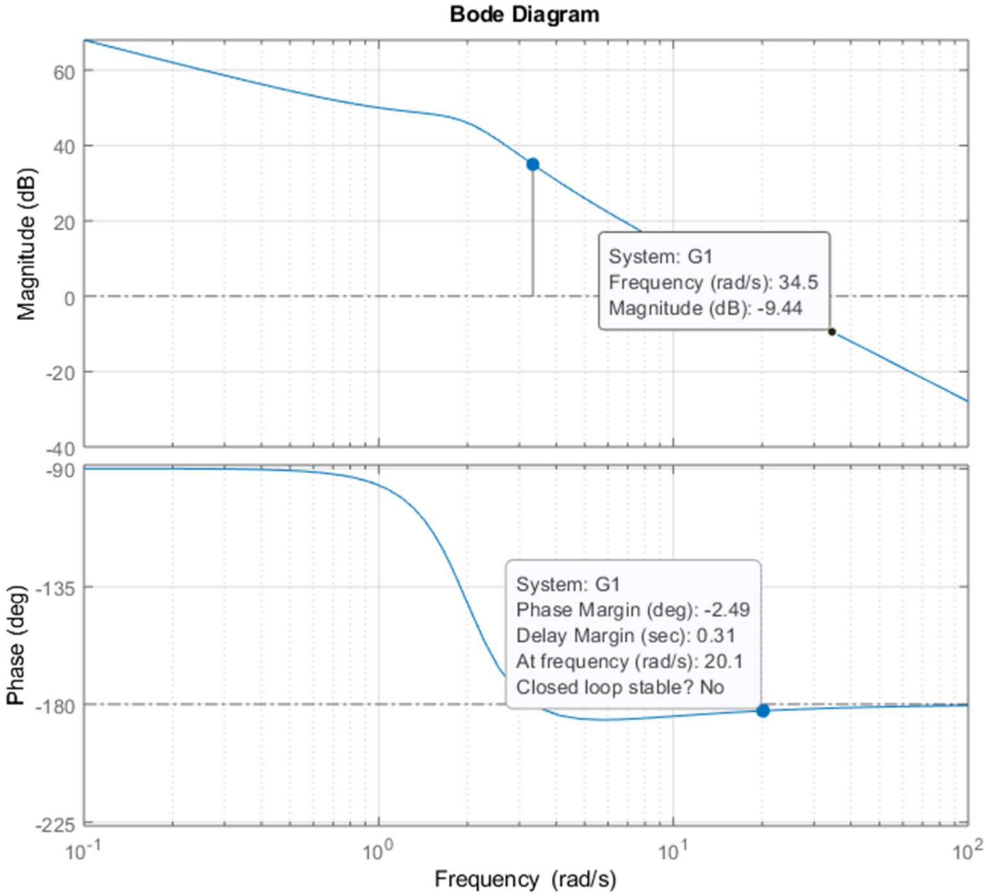

```
%% Es 3-c risposta al gradino e sovraelongazione
Gcl2 = feedback(Gc*G1,1);
figure, step(Gcl2) % overshoot 33.5%
```



ESERCIZIO 3 soluzione alternativa

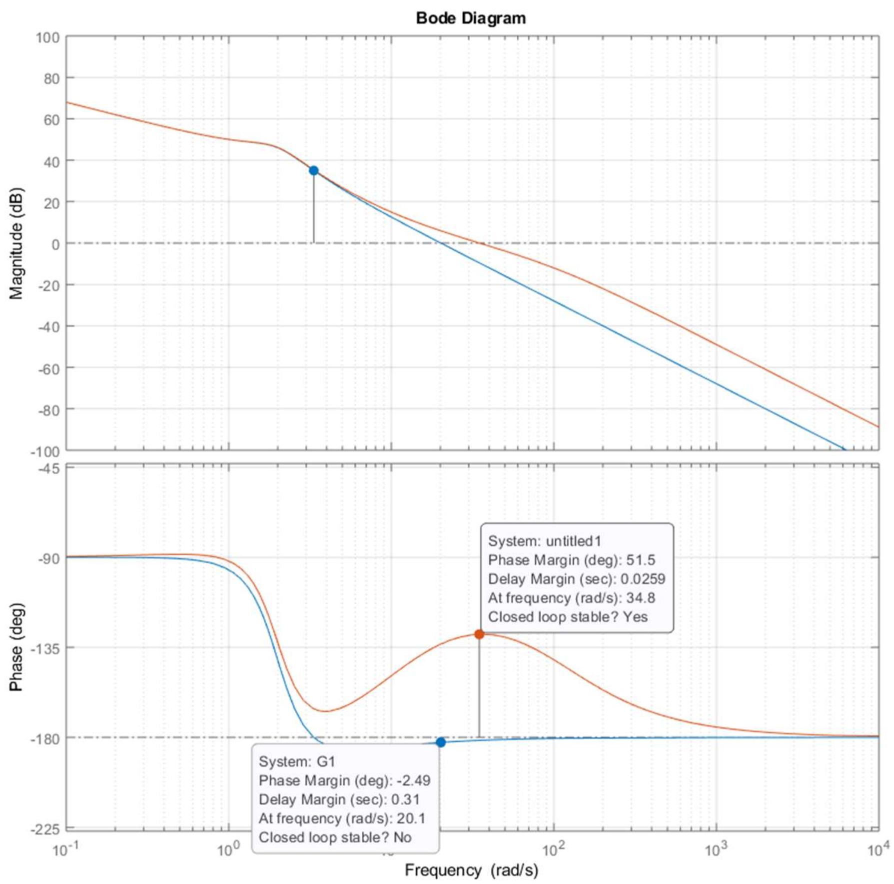
```
%% Es 3-a progetto della rete con procedura empirica

G1 = K*G;
Mf = 45;

figure, bode(G1)
grid on

% passo 1: determinare l'attuale margine di fase

% passo 2: determinare l'anticipo di fase necessario per
% ottenere il margine di fase voluto più un margine di
% sicurezza

% passo 3: determinare alpha secondo la regola
% alpha = (1-sind(phi))/(1+sind(phi))

% passo 4: determinare graficamente la pulsazione omega a
% cui il modulo del sistema non compensato vale
```

```
% -20*log10(1/sqrt(alpha))

% passo 5: ricavare tau dalla relazione
% tau = 1/(omega*sqrt(alpha))
```




```
Mf = 45;
% -2.5 (circa 3) = margine di fase attuale
% serve quindi un anticipo di fase di almeno
% Mf (desiderato) + 3 + il margine di sicurezza (altri 5)
phi = 3 + Mf + 5;
alpha = (1 - sind(phi))/(1 + sind(phi));

alpha =
    0.1120

Mm = -20*log10(1/sqrt(alpha));

Mm =
  -9.5096
```

```
omega=34.5; % valore selezionato dal grafico di ampiezza
tau = 1/(sqrt(alpha)*omega);


tau =
    0.0866


s = tf('s');
Gc = (1+tau*s)/(1+alpha*tau*s);

Gc =
  0.08663 s + 1
  -------------
  0.009698 s + 1

%% Es 3-b diagrammi di bode e margini di fase

figure, bode(G1)
grid on
hold on
bode(Gc*G1) % verificare i margini
```



```
%% Es 3-c risposta al gradino e sovraelongazione
Gcl2 = feedback(Gc*G1,1);
figure, step(Gcl2) % overshoot 27%
```