SOLUZIONE *Prova MATLAB Tipo – D*

Esame di "FONDAMENTI DI AUTOMATICA" (9 CFU)

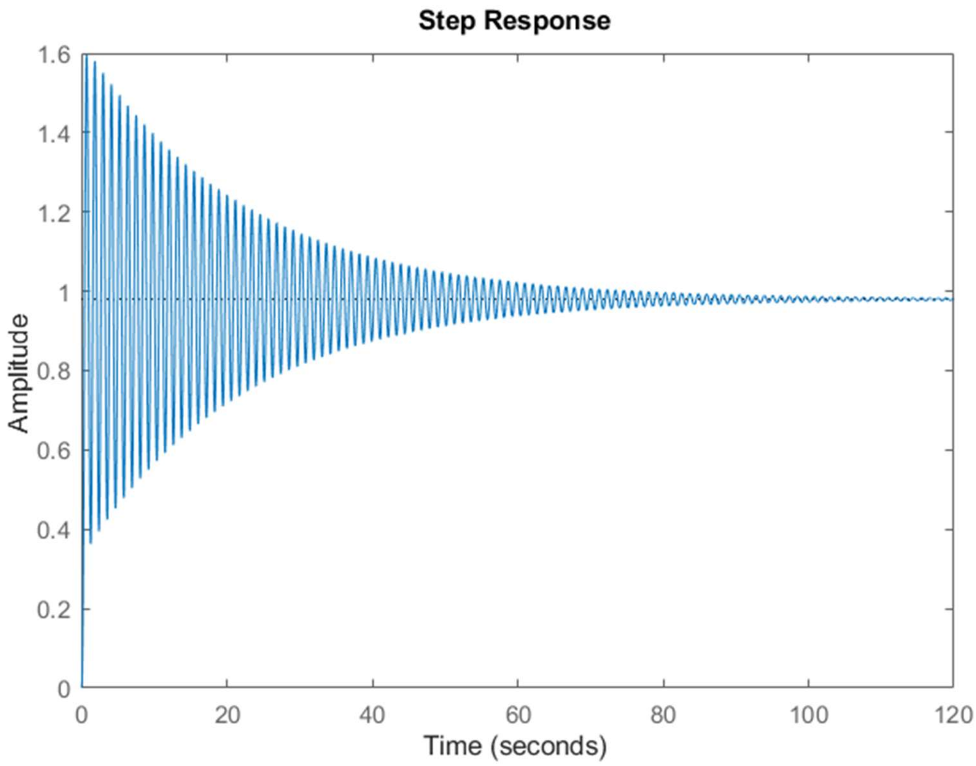
ESERCIZIO 1

```
%% Es 1-a: esecuzione dello script InitAutomaticaTipoD.m
%  ATTENZIONE a maiuscole/minuscole nel nome file

>> InitAutomaticaTipoC

A =
    -1    -6     0
     4    -4    -3
     0     1     0

B =
    0.6000
         0
         0

C =
     0     0     1

D =
     0

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

G =

           2.4
  ----------------------
  s^3 + 5 s^2 + 31 s + 3

%% Es 1-B verifica poli e autovalori
p = pole(G)
ev = eig(A)


% poli di G e autovalori di A coincidono, infatti il
% sistema è completamente osservabile (rank(Qt)=3)
```

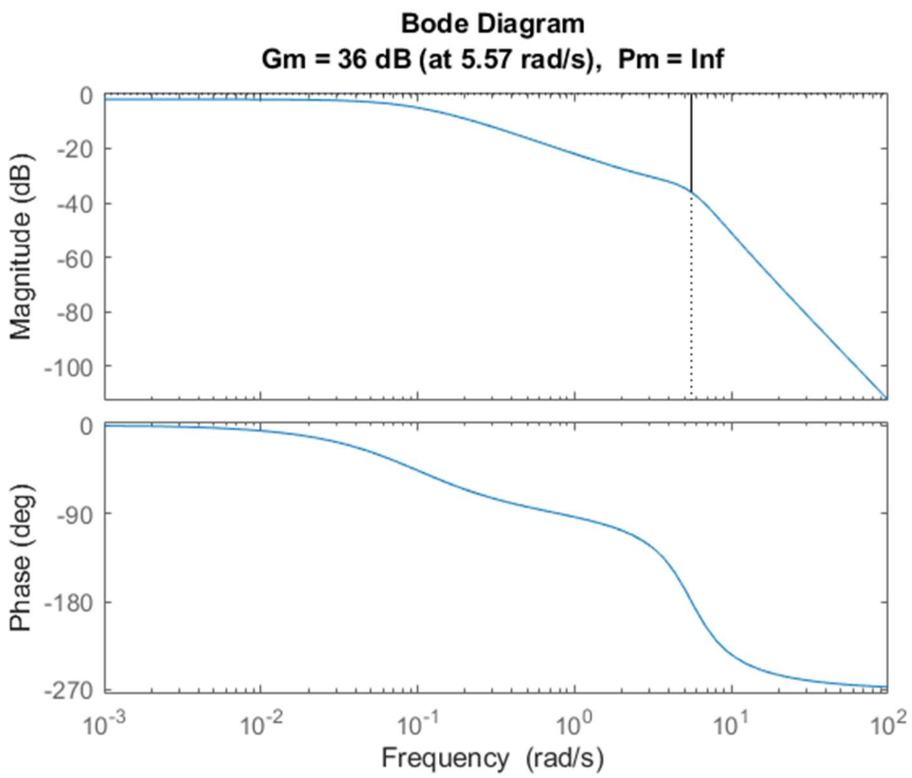
```
p =
  -2.4508 + 4.9509i
  -2.4508 - 4.9509i
  -0.0983 + 0.0000i
ev =
  -2.4508 + 4.9509i
  -2.4508 - 4.9509i
  -0.0983 + 0.0000i
```

ESERCIZIO 2

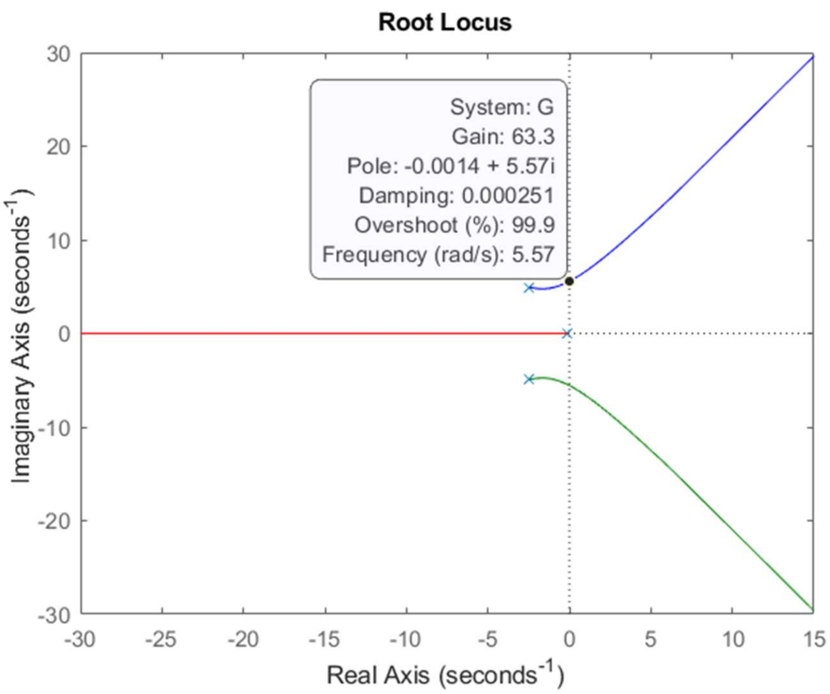
```
%% Es 2-a specifiche su errore a regime

syms K
% costante e errore di posizione
kp = K*dcgain(G);
ep = 1/(1+kp);
% calcolo di K
K = solve(ep==0.02, K)

K =
   245/4

K=double(K)

K =
   61.2500

%% Es 2-b Verifica della stabilità ad anello chiuso

Gcl = feedback(K*G,1);
figure, step(Gcl) % sistema stabile (di poco!!!)
```



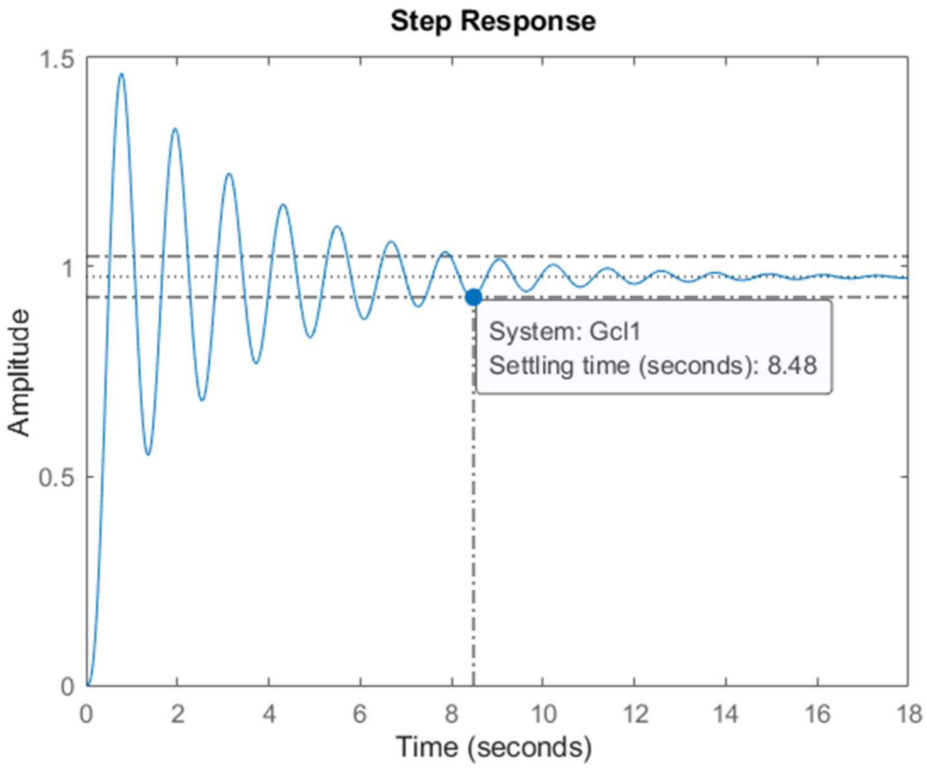

```
%% Es 2-c Calcolo del guadagno critico o limite per la stabilità ad anello chiuso

% SOLUZIONE 1: dal margine di ampiezza

figure, margin(G)

Klim = db2mag(36)

% oppure anche direttamente

Klim = margin(G)

% NOTA: il risultato di Klim=margin(G), senza creare il grafico, è già in valore assoluto, non in dB
```



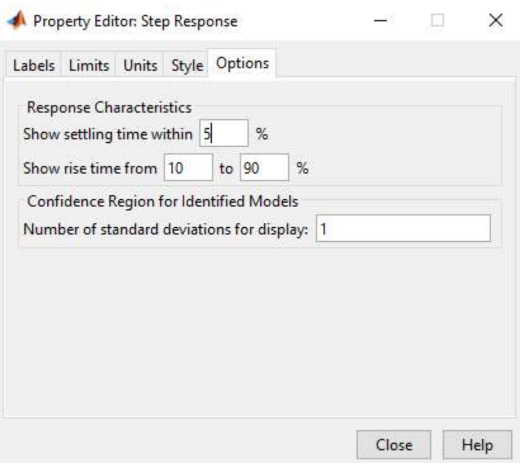

```
% SOLUZIONE 2: plot del luogo delle radici

figure, rlocus(G)

% valore selezionato dal grafico
Klim = 63.3;
```



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

K1 = 0.8*Klim;

K1 =

   50.64

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:



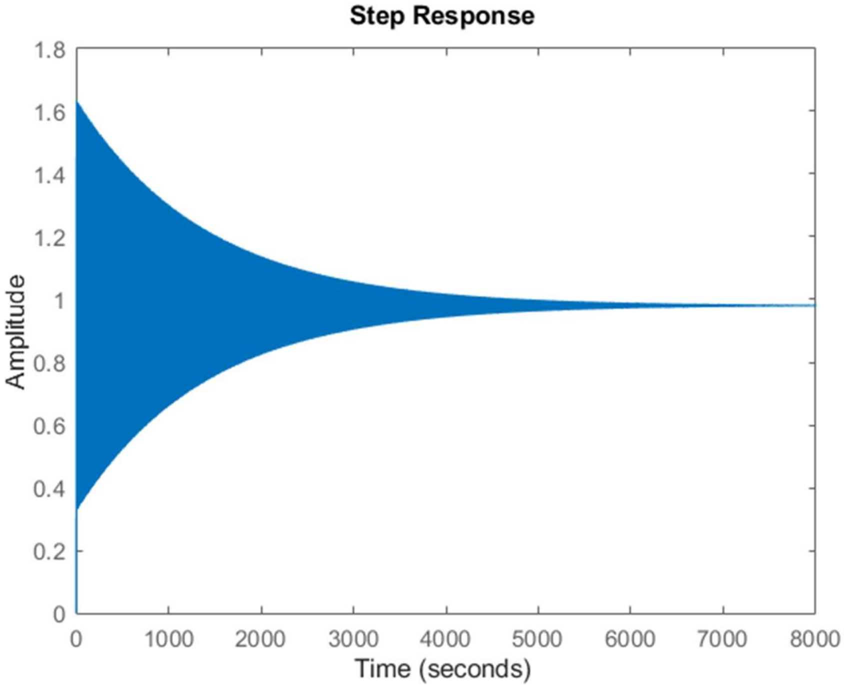
Oppure tramite i comandi:

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

figure,step(Gcl1,Popt)
```

ESERCIZIO 3

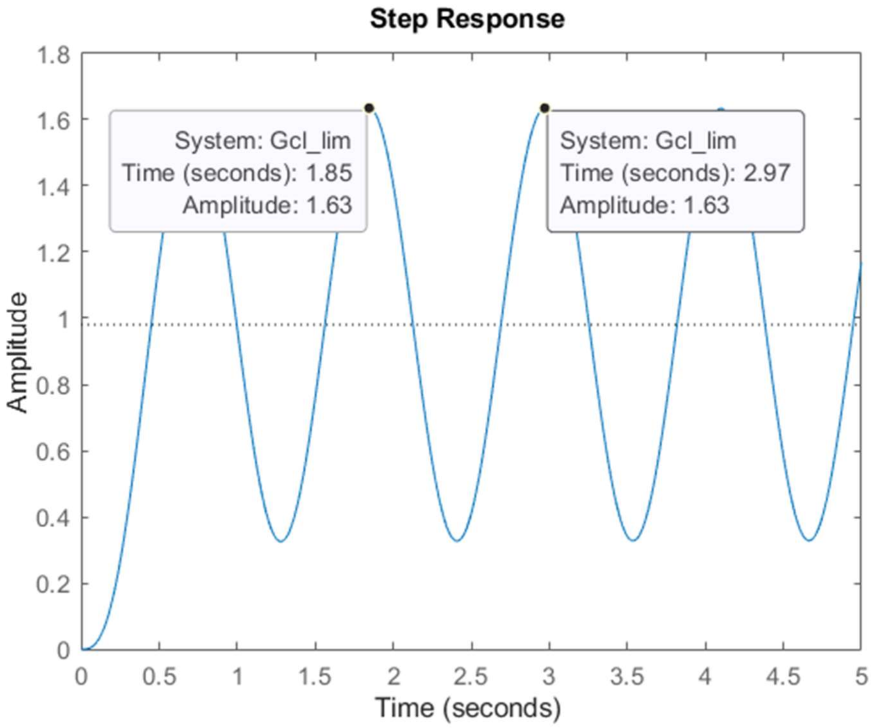
NOTA: il guadagno critico K_0 considerato dal metodo di Ziegler-Nichols corrisponde al `Klim` appena determinato. Per ottenere il parametro T_0, invece, è necessario analizzare la risposta al gradino del sistema chiuso in retroazione con tale guadagno critico, al fine di determinare da tale risposta il periodo delle oscillazioni.

```
%% Es 3-a progetto regolatore
Gcl_lim = feedback(Klim*G,1);
figure, step(Gcl_lim)
```



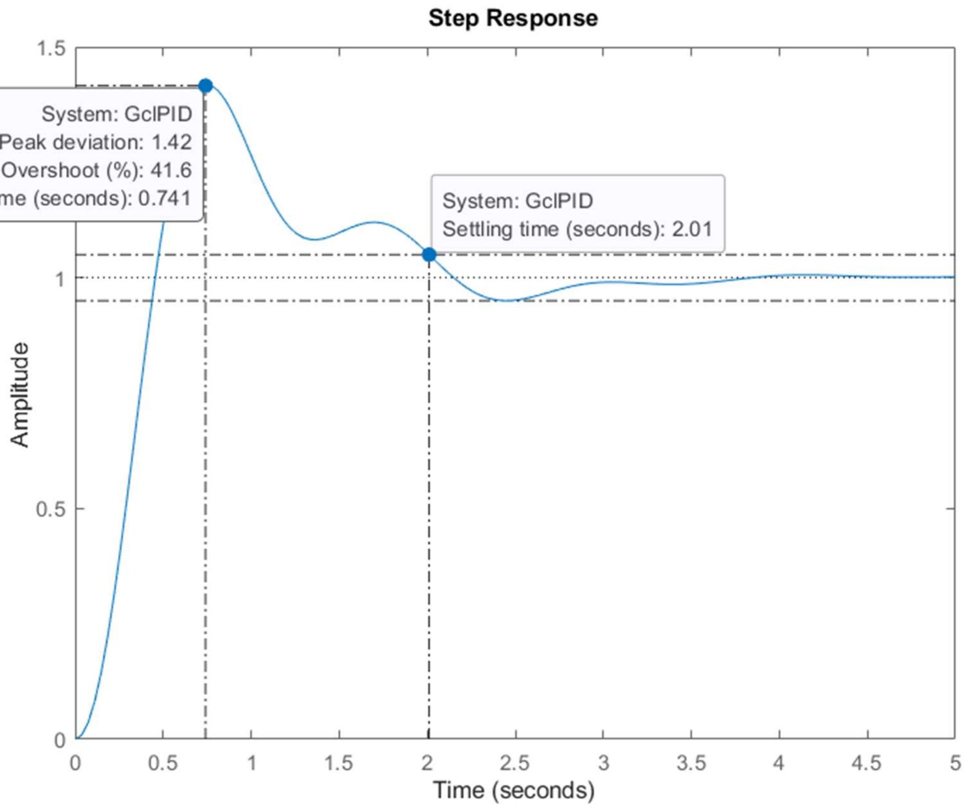
NOTA: la figura evidenzia che, a causa di approssimazioni numeriche inevitabili, è pressochè impossibile imporre esattamente la condizione di stabilità semplice. Infatti, le oscillazioni della risposta tendono a zero, seppure nel corso di un transitorio mostrato dal grafico di almeno 6000 secondi (100 minuti). Per gli scopi di progetto, tuttavia, la condizione di stabilità asintotica ma con decadimento esponenziale delle oscillazioni molto lento è sufficiente per identificare con ottima approssimazione il periodo delle oscillazioni stesse. Per quest'ultima operazione, è però necessario ingrandire opportunamente il grafico, visualizzando una porzione di tempo limitata, ad esempio i primi 5 secondi ed eventualmente ingrandire ulteriormente a mano, tramite il tasto di zoom (visualizzati posizionando il mouse in alto a destra del grafico).

```
figure,step(Gcl_lim,5);
T0=2.97-1.85;
```




```
% periodo oscillazioni ricavato graficamente

%% Taratura Ziegler-Nichols, regolatore PID
Kp = 0.6*Klim
Ti = 0.5*T0
Td = 0.125*T0

GcPID = Kp*(1+Td*s+1/(Ti*s))
GclPID = feedback(GcPID*G,1);
figure, step(GclPID)

Kp =
    37.9800
```

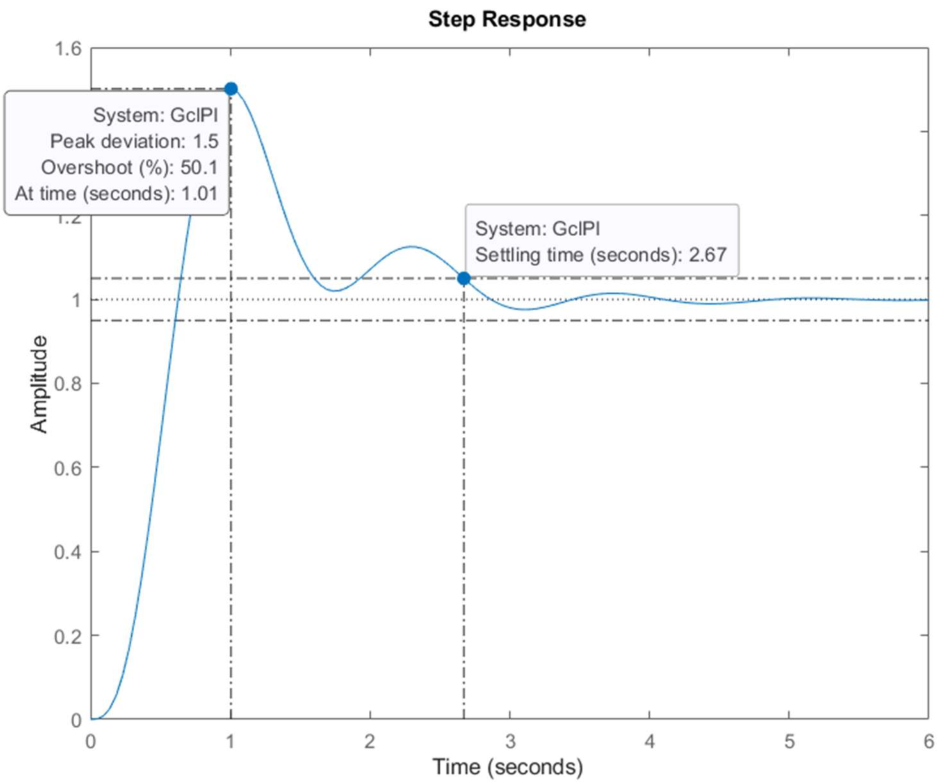
```
Ti =
    0.5600

Td =
    0.1400

GcPID =
  2.978 s^2 + 21.27 s + 37.98
  ---------------------------
            0.56 s

Continuous-time transfer function.
```



NOTA BENE: impostare la visualizzazione del tempo di assestamento al 5% …

```
%% Taratura Ziegler-Nichols, regolatore PI
Kp=0.45*Klim
Ti=0.85*T0
GcPI=Kp*(1+1/(Ti*s))
GclPI = feedback(GcPI*G,1);
figure,step(GclPI)

Kp =
```

```
    28.4850

Ti =
    0.9520

GcPI =
  27.12 s + 28.48
  ---------------
      0.952 s
```



NOTA BENE: impostare la visualizzazione del tempo di assestamento al 5% …

RIASSUMENDO: In questo caso, il regolatore PID è quello che garantisce sia il minor tempo di assestamento che la minor sovraelongazione nel transitorio, ed è pertanto preferibile al regolatore PI.

Entrambe le versioni sono comunque funzionali all'annullamento dell'errore a regime, condizione per la quale è appunto necessario un regolatore con un polo in 0, non essendo quest'ultimo presente nella funzione di trasferimento del sistema da controllare.

```
%% CONFRONTO CONCLUSIVO: grafici della risposta al
%  gradino dei due regolatori analizzati (PI e PID)
%  sovrapposti nello stesso plot

figure, step(GclPID)
hold on
step(GclPI)
```