SOLUZIONE *Prova MATLAB Tipo – C*

Esame di “FONDAMENTI DI AUTOMATICA” (9 CFU)

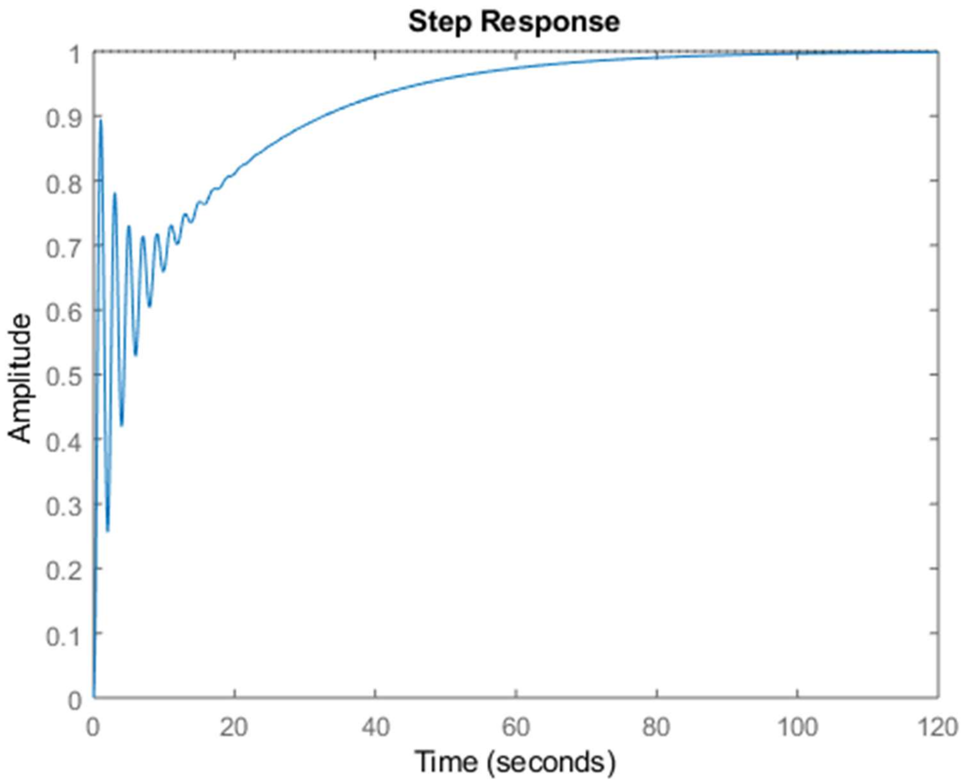
ESERCIZIO 1

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

>> InitAutomaticaTipoC

A =
   -0.3000   50.0000         0
   -0.1000   -0.4000         0
         0   50.0000         0
B =
    0.2000
    0.1000
         0
C =
     0     0     1

D =
     0

%% Es 1-A fdt del sistema
sys = ss(A,B,C,D);
G = tf(sys)
G =
        5 s + 0.5
  ----------------------
  s^3 + 0.7 s^2 + 5.12 s

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

% poli di G e autovalori di A coincidono, quindi il
% sistema è completamente controllabile e osservabil

p =
   0.0000 + 0.0000i
  -0.3500 + 2.2355i
  -0.3500 - 2.2355i
```

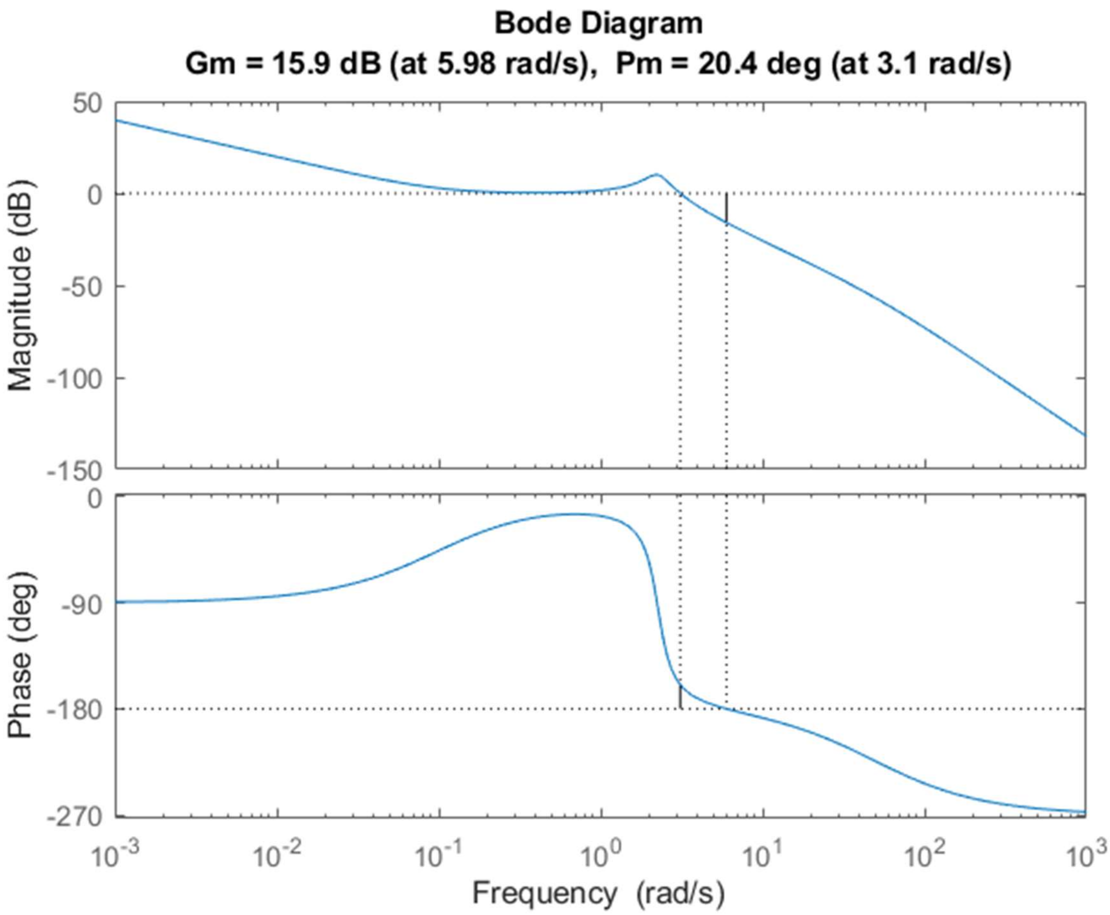
```
ev =
   0.0000 + 0.0000i
  -0.3500 + 2.2355i
  -0.3500 - 2.2355i
```

ESERCIZIO 2

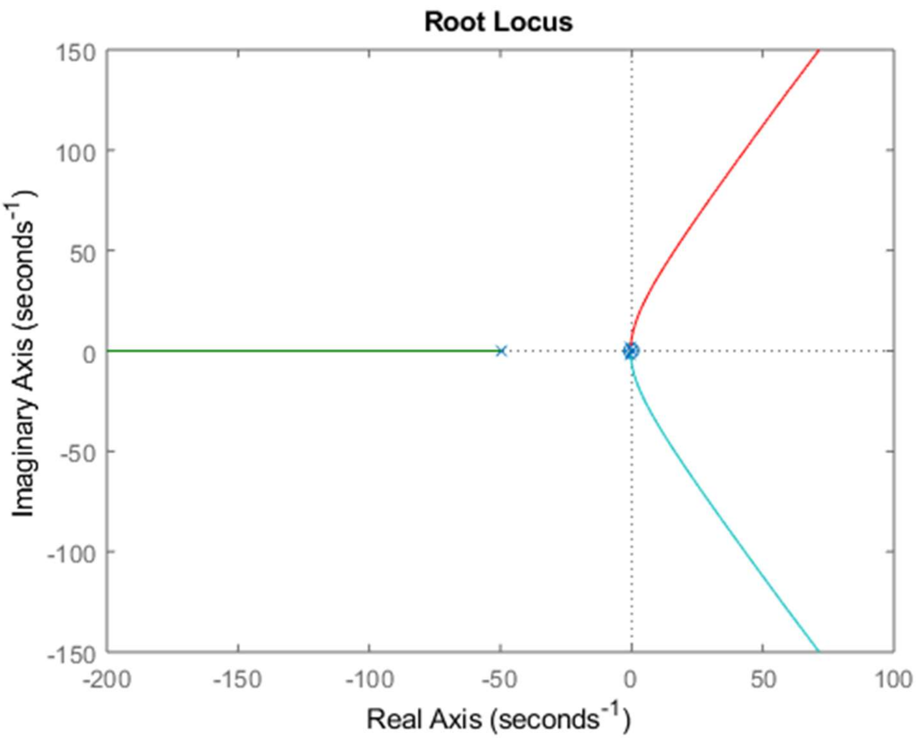
```
%% Es 2-a risposta al gradino ad anello chiuso
s=tf('s');
H=1/(1+s/50);
Gcl = feedback(G,H);
figure, step(Gcl) % sistema stabile
```

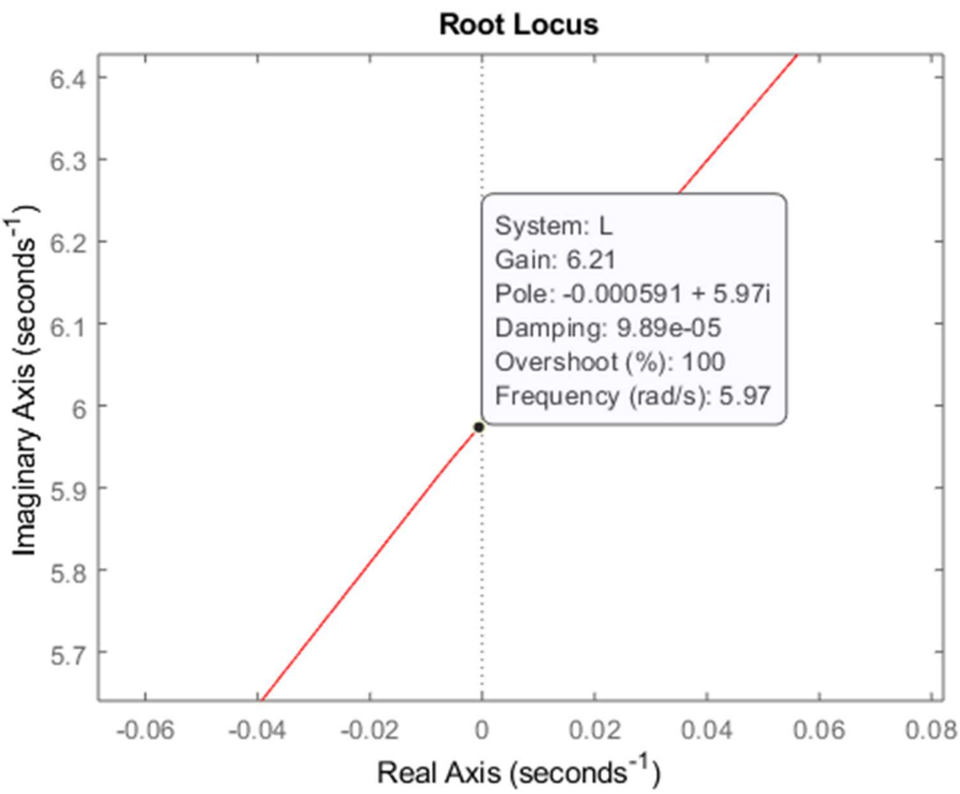



```
%% Es 2-b Calcolo del guadagno critico o limite per la
stabilità ad anello chiuso

% SOLUZIONE 1: dal margine di ampiezza

L = G*H
figure, margin(L)

Klim = db2mag(15.9)
```

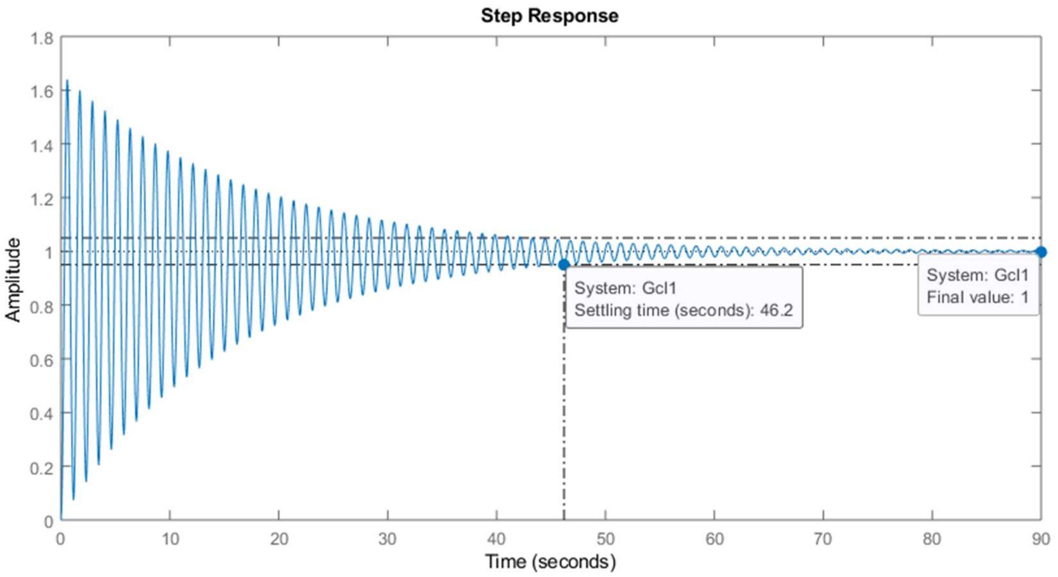



```
% oppure anche direttamente

Klim = margin(L)

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

% SOLUZIONE 2: plot del luogo delle radici

L = G*H
figure, rlocus(L)
Klim=6.21; % valore selezionato dal grafico
```

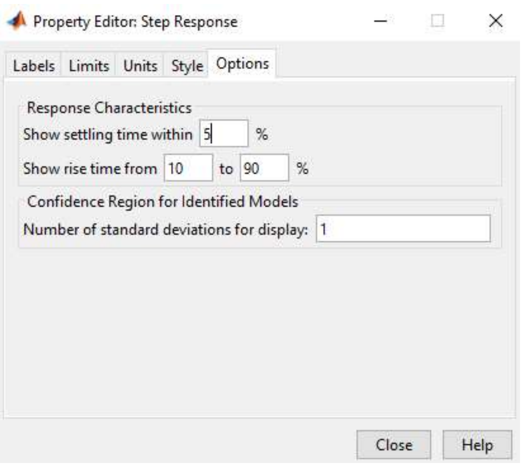

Root Locus
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})
150
100
50
0
-50
-100
-150
-200
-150
-100
-50
0
50
100


Root Locus
System: L
Gain: 6.21
Pole: -0.000591 + 5.97i
Damping: 9.89e-05
Overshoot (%): 100
Frequency (rad/s): 5.97
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})
6.4
6.3
6.2
6.1
6
5.9
5.8
5.7
-0.06
-0.04
-0.02
0
0.02
0.04
0.06
0.08

```
%% Es 2-c risposta al gradino, tempo di assestamento e
errore a regime
Gcl1 = feedback(0.8*Klim*G,H);
figure, step(Gcl1);

% Valore finale della risposta = 1, quindi errore a
% regime NULLO
```



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



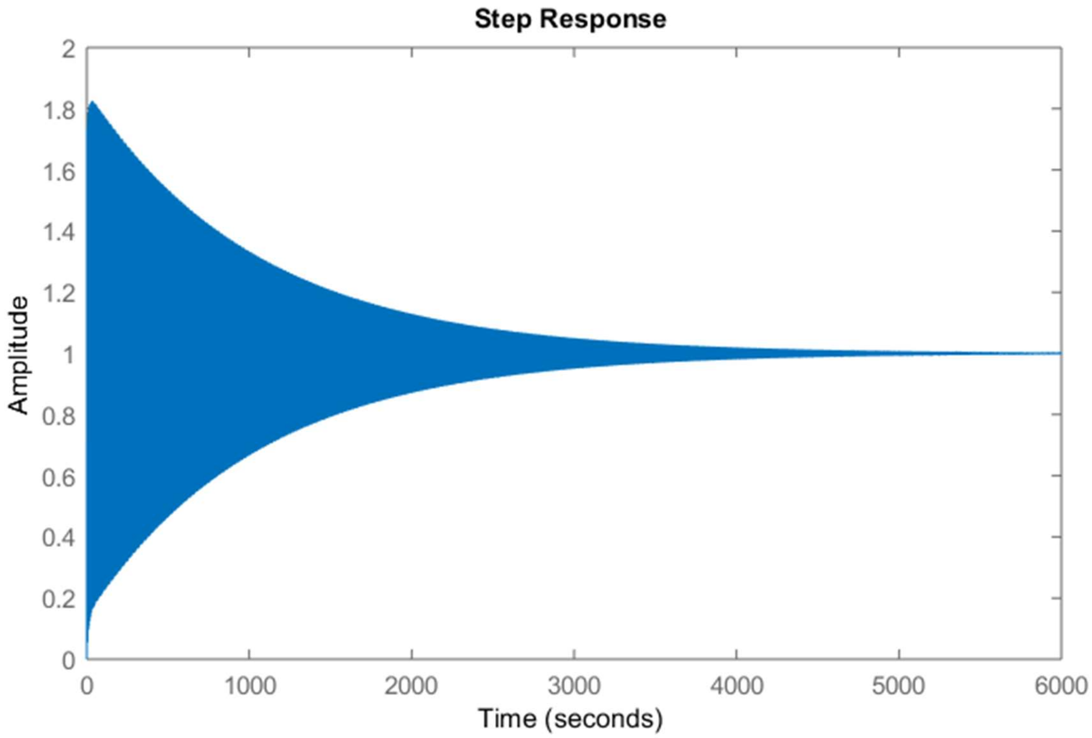
Oppure tramite i comandi:

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

figure,step(Gcl1,Popt)
```

ESERCIZIO 3

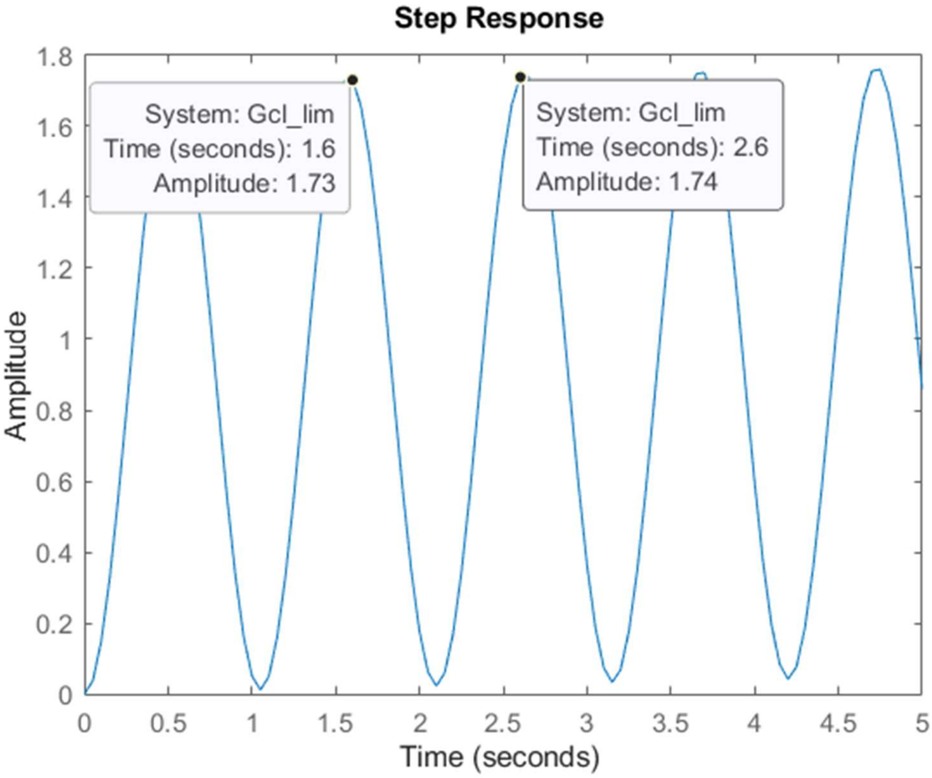
NOTA: il guadagno critico K_0 considerato dal metodo di Ziegler-Nichols corrisponde al `Klim` appena determinato. Per ottenere il parametro T_0, invece, è necessario analizzare la risposta al gradino del sistema chiuso in retroazione con tale guadagno critico, al fine di determinare da tale risposta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

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

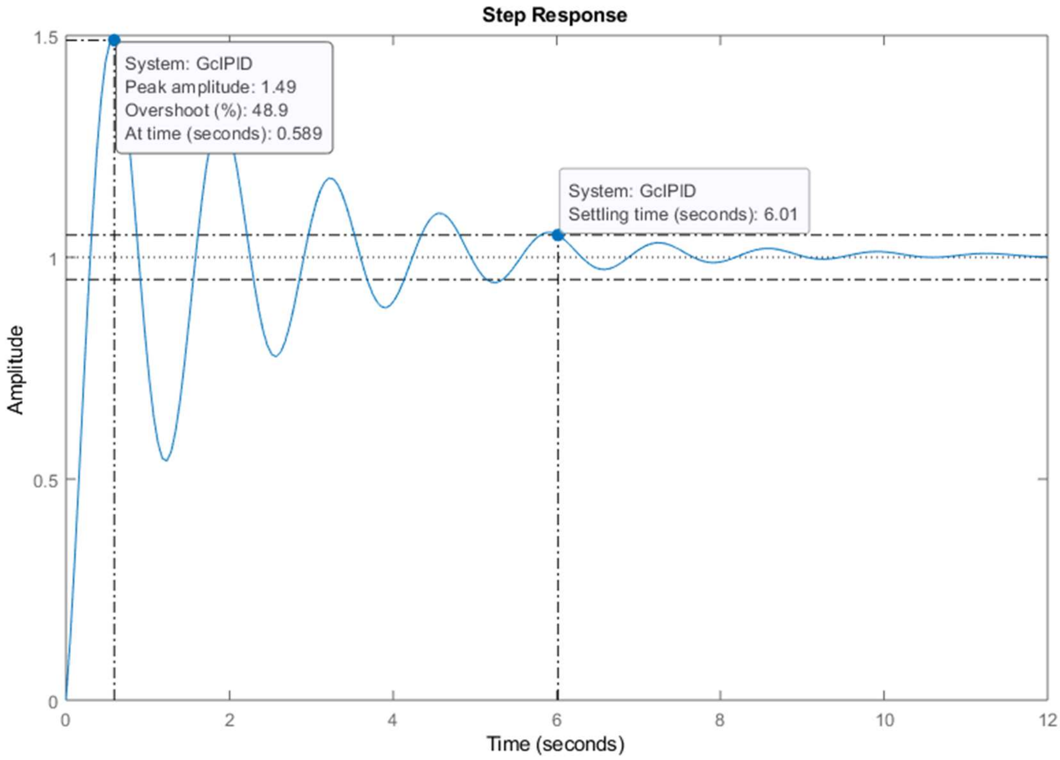


NOTA: la figura evidenzia che, a causa di approssimazioni numeriche inevitabili, è pressochè impossibile imporre esattamente la condizione di stabilità semplice. Infatti, le oscillazioni della risposta tendono a zero, seppure nel corso di un transitorio mostrato dal grafico di almeno 6000 secondi (100 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 = 2.6 - 1.6;
% periodo oscillazioni ricavato graficamente
```

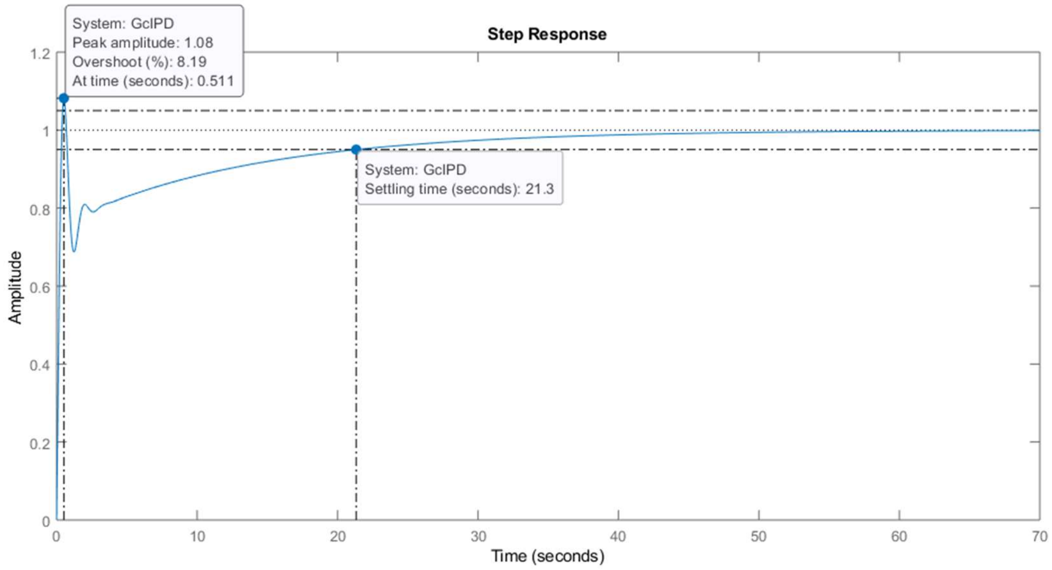



```
%% Taratura Ziegler-Nichols, regolatore PID
Kp = 0.6*Klim
Ti = 0.5*T0
Td = 0.125*T0

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

Kp =
    3.7260
Ti =
    0.5000
Td =
    0.1250

GcPID =
  0.2329 s^2 + 1.863 s + 3.726
  ----------------------------
             0.5 s

Continuous-time transfer function.
```



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

```
%% Taratura Ziegler-Nichols, regolatore PD
Kp=0.5*Klim
Td=0.2*T0
GcPD=Kp*(1+Td*s)
GclPD = feedback(GcPD*G,H);
figure,step(GclPD)

Kp =
    3.1050

Td =
    0.2000


GcPD =
  0.621 s + 3.105
```

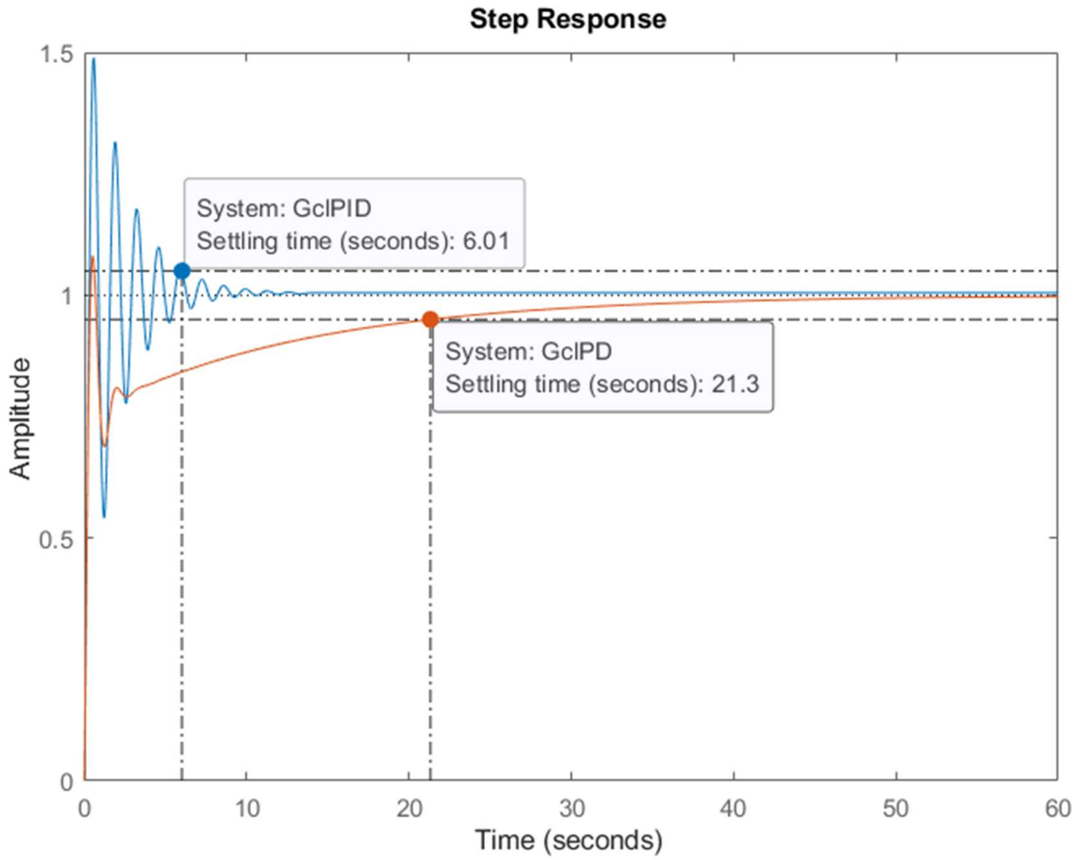


NOTA BENE: impostare la visualizzazione del tempo di assestamento al 5% …

RIASSUMENDO: In questo caso, il regolatore PID è quello che garantisce il minor tempo di assestamento, mentre il regolatore PD è quello che permette di ottenere la minor sovraelongazione nel transitorio. Entrambe le scelte possono quindi essere accettabili come risposta finale.

Si noti inoltre che la funzione di trasferimento G(s) del sistema fisico ha un polo in 0, il che significa che l'errore a regime dell'anello di retroazione è comunque nullo anche con un regolatore senza termine integrale (i.e. P o PD).

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

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


Step Response
System: GclPID
Settling time (seconds): 6.01
System: GclPD
Settling time (seconds): 21.3
Amplitude
Time (seconds)
1.5
1
0.5
0
0
10
20
30
40
50
60