Esame di “FONDAMENTI DI AUTOMATICA” (9 CFU)

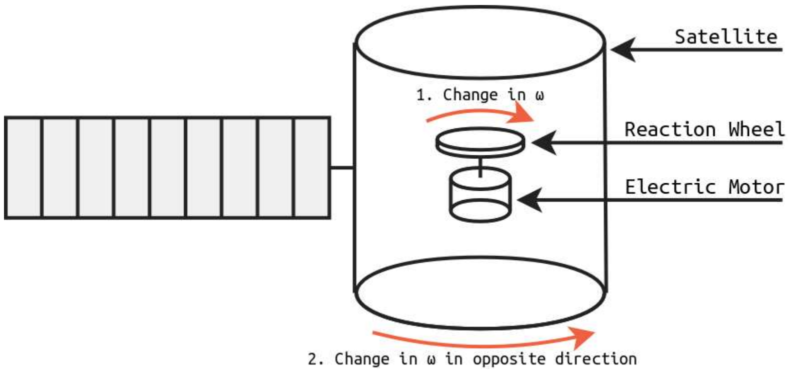
Prova MATLAB – 23 giugno 2022 – Testo B

Istruzioni per lo svolgimento: lo studente deve consegnare al termine della prova una cartella nominata Cognome_Nome, contenente:

1. Un Matlab script file (i.e. file di testo con estensione .m) riportante i comandi eseguiti e la risposta alle eventuali richieste teoriche sotto forma di commento (i.e. riga di testo preceduta dal simbolo %)
   NOTA: per copiare i comandi dalla Command History, visualizzarla tramite menu “Layout → Command History → Docked”, selezionare in tale finestra le righe di interesse tramite *Ctrl+mouse left-click* e dal menu visualizzato tramite *mouse right-click* selezionare “create script”
2. Le figure rilevanti per la dimostrazione dei risultati ottenuti in **formato JPEG o PNG** avendo cura di salvare i file delle figure quando queste mostrano le caratteristiche di interesse per la verifica del progetto (i.e. Settling Time, Stability Margins, ecc.).
   NOTA: per salvare una figura Matlab in formato PNG o JPG, usare il menu “File → Save as” dalla finestra della figura di interesse, assegnarle un nome e selezionare l’estensione *.PNG o *.JPG nel menu a tendina “salva come”, avendo cura che le figure siano salvate quando queste mostrano le caratteristiche di interesse per la verifica del progetto

INTRODUZIONE

Si consideri il sistema di controllo dell’orientamento di un satellite tramite ruota di reazione, schematizzato nella seguente figura:



ed il cui modello matematico è stato oggetto dei primi esercizi della prova scritta odierna (Testo A). Il modello, del tipo

$$\dot{x}(t) = Ax(t) + Bu(t);\; y(t) = Cx(t) + Du(t)$$

è inizializzato dallo script **`initAutomaticaTestoB.m`** fornito dal docente.

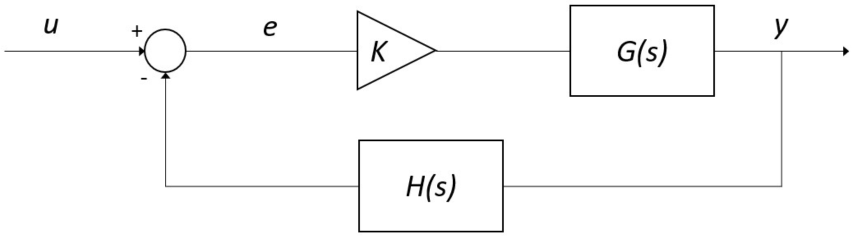
ESERCIZIO 1.

a) Dato il modello ottenuto nell’introduzione, si ricavi la funzione di trasferimento G(s) del sistema in esame.

b) Si determinino i poli della funzione di trasferimento e si verifichi se coincidono con gli autovalori di A. Descrivere il motivo di eventuali discrepanze tramite righe di commento (i.e. precedute dal simbolo %) sul file .m

ESERCIZIO 2

Si consideri il sistema in retroazione unitaria rappresentato in figura:

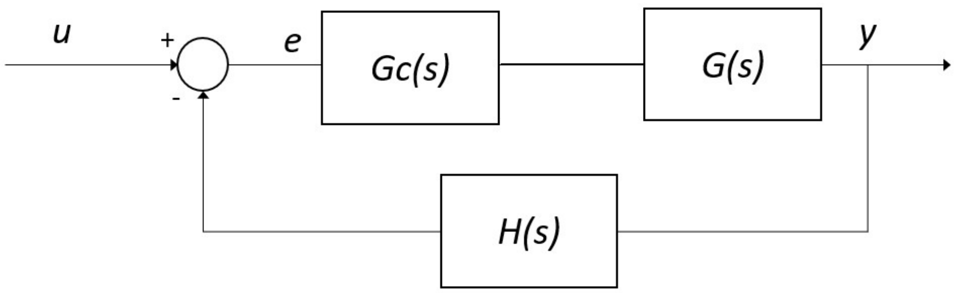


Con G(s) ricavata dall'Esercizio 1 e H(s) che è già stata inizializzata dallo script **`InitAutomaticaTestoB.p`**.

a) Si verifichi se il sistema ad anello chiuso, con guadagno $K = 1$, risulti o meno stabile tramite l'analisi della risposta $y(t)$ al gradino unitario.
b) Si determini, se esiste, il valore del guadagno K_{lim} per il quale il sistema risulta semplicemente stabile, utilizzando il grafico del luogo delle radici della funzione G(s)*H(s).
c) Si ponga $K_1 = 0.8\, K_{lim}$, si visualizzi l'andamento della risposta al gradino $y(t)$ del sistema chiuso in retroazione con tale guadagno e si determini il tempo d'assestamento al 5%.
d) Si determini il valore a regime della risposta al gradino $y(t)$ e si motivi il risultato tramite righe di commento (i.e. precedute dal simbolo %) sul file .m

ESERCIZIO 3

Si consideri il sistema rappresentato in figura



Con $G(s)$ ricavata dall'Esercizio 1 e H(s) come indicato nell'Esercizio 2.

a) Si determinino come possibili funzioni di trasferimento alternative per il controllore $G_c(s)$ quelle di un regolatore di tipo **PD** e di uno di tipo **PID**, considerati entrambi nella <u>formulazione classica</u> e con i parametri K_p, T_i, T_d tarati secondo il metodo di Ziegler-Nichols basato sull'oscillazione critica ad anello chiuso (**vedi tabella allegata**).
b) Si verifichi tramite l'analisi della risposta al gradino del sistema compensato e chiuso in retroazione quale tra i regolatori proposti sia il più efficace in termini di massima sovraelongazione percentuale e tempo di assestamento.

TIPO	K_p	T_i	T_d
PD	0.5 K_0	-	0.2 T_0
PID	0.6 K_0	0.5 T_0	0.125 T_0

NOTA:
K_0 = guadagno critico, di fatto corrispondente al guadagno K_{lim} determinato al punto b) dell'Esercizio 2, cioè tale per cui il sistema chiuso in retroazione risulti semplicemente stabile (i.e. con oscillazione persistente della risposta).
T_0 = periodo delle oscillazioni della risposta in condizione di stabilità semplice ad anello chiuso.

SOLUZIONE (traccia):

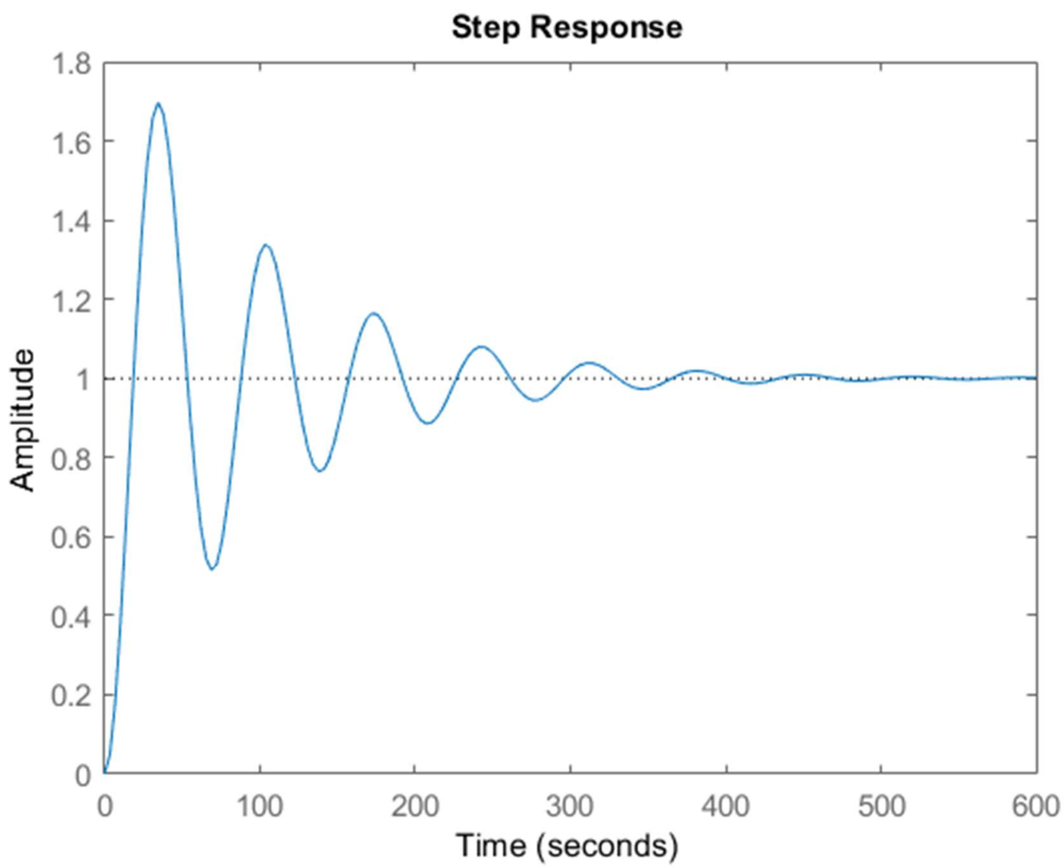
Contenuto di `initAutomaticaTestoB`

```
% Inizializzazione parametri
J1=120;
J2=80;
b=2;

% Inizializzazione matrici
A = [ 0, 1,     0;
      0, 0,  b/J1;
      0, 0, -b/J2]
B = [0;
    -1/J1;
    1/J2]
C=[-1 0 0]
D=0

% Inizializzazione ramo di feedback:
s=tf('s');
H=1/(1+s*0.5)
clear s
```

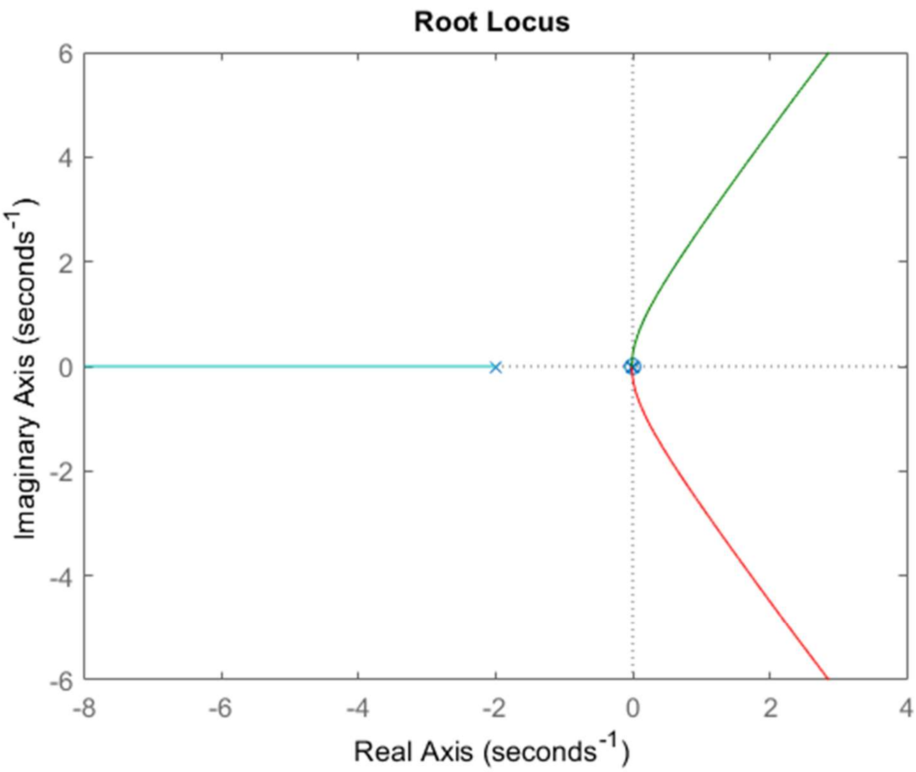
Svolgimento:

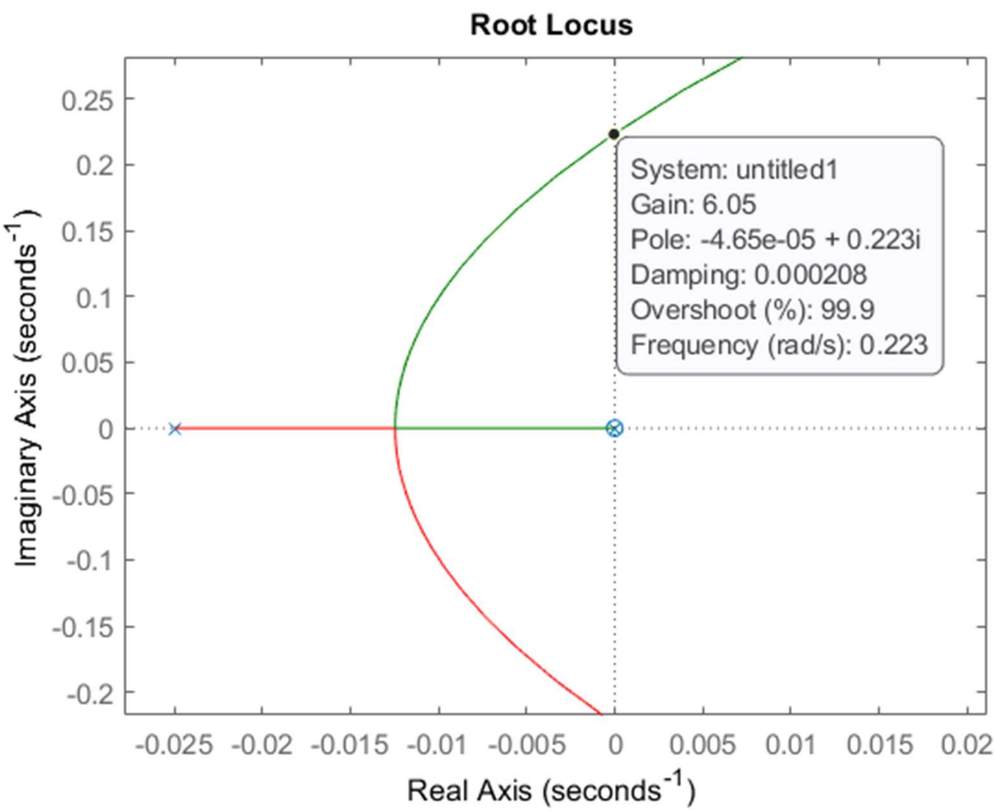
```
sys=ss(A,B,C,D)
G=tf(sys)

G =
```

```
   0.008333 s + 3.614e-20
   ----------------------
      s^3 + 0.025 s^2

pole(G)

ans =
         0
         0
   -0.0250
eig(A)

ans =
         0
         0
   -0.0250
% Poli e autovalori coincidono (sistema completamente controllabile e osservabile)

Gcl=feedback(G,H)

step(Gcl)
```



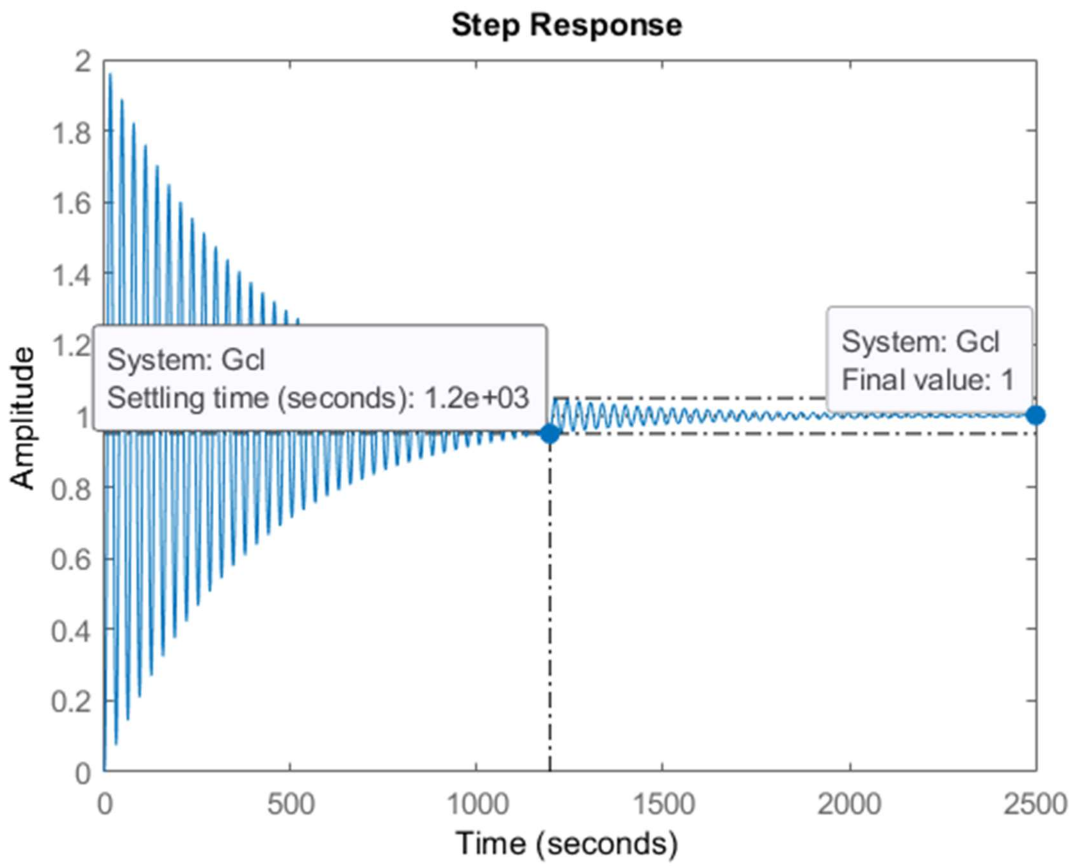
```
rlocus(G*H)
```






```
Klim = 6.05

Gcl=feedback(0.8*Klim*G,H)
```

```
step(Gcl)
```




```
% Valore a regime = 1 (errore nullo in risposta al gradino unitario), perché il sistema è di tipo 1, cioè ha un polo nell'origine
```

NOTA:
Nelle funzioni di trasferimento ottenuto si riscontrano sia un polo nullo che uno zero nullo, per i quali TEORICAMENTE si dovrebbero avere delle semplificazioni, ma per via di approssimazioni numeriche ciò non avviene sempre correttamente. In effetti, la funzione Gcl ottenuta nell'ultimo passaggio precedente è:

```
Gcl =
              0.02017 s^2 + 0.04033 s + 1.749e-19
  --------------------------------------------------------
  0.5 s^4 + 1.012 s^3 + 0.025 s^2 + 0.04033 s + 1.749e-19
```

che però è di fatto equivalente a:

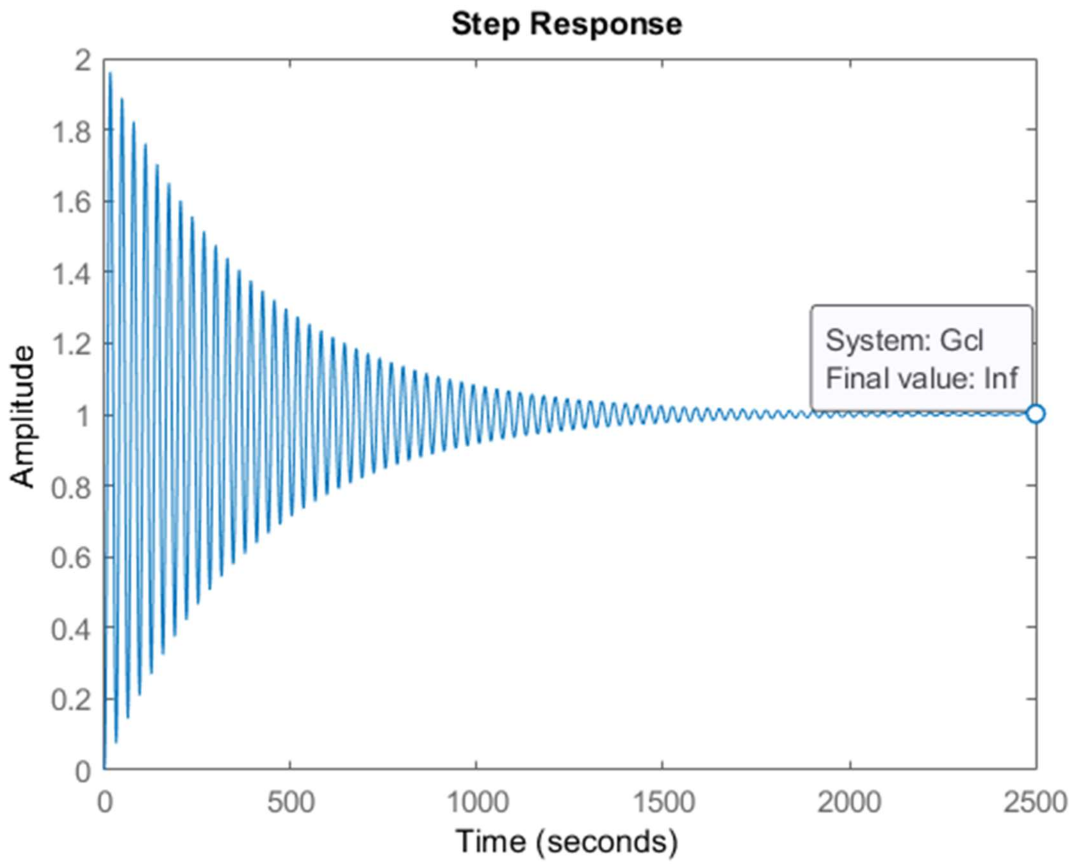
```
Gcl =
             0.02017 s + 0.04033
  ---------------------------------------
  0.5 s^3 + 1.012 s^2 + 0.025 s + 0.04033
```

Tuttavia, su alcuni PC del laboratorio dipartimentale (probabilmente per via di una versione di Matlab non recente), le stesse operazioni riportate qui hanno dato un risultato con il **segno**

dei termini costanti (i.e. quelli di ordine di grandezza 10^{-9}) **negativo sia a denominatore che a numeratore**, cioè ad esempio:

```
Gcl =

          0.02017 s^2 + 0.04033 s - 1.54e-19
  ------------------------------------------------------
  0.5 s^4 + 1.012 s^3 + 0.025 s^2 + 0.04033 s - 1.54e-19
```

La risposta al gradino di tale funzione ottenuta con il comando **`step()`** risulta però con limite per tempo infinito indeterminato (anche se indicato da Matlab come "Inf") e per questo Matlab non fornisce il valore corrispondente del tempo di assestamento:



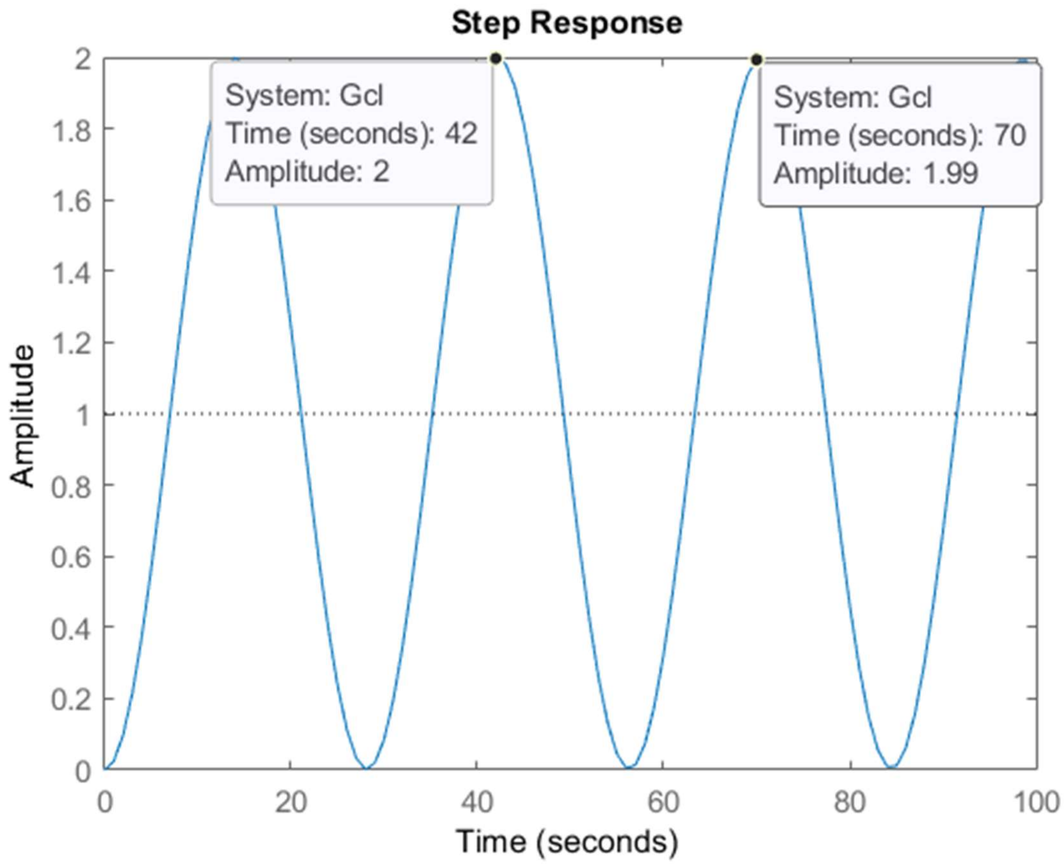
Si noti come l'aspetto del grafico sia di fatto identico al precedente e, soprattutto, come in realtà la risposta tenda chiaramente a un valore unitario a regime.
In tale caso, la consegna può essere rispettata salvando il grafico appena riportato e segnando nel file di riepilogo dei comandi usati che il tempo di assestamento non era visualizzato per problemi numerici di Matlab.
Si noti infine che in tale caso sarebbe ancora possibile ottenere questo parametro per altra via, utilizzando il comando **`stepinfo()`**, che però non fornisce un grafico ma solo i valori caratteristici della risposta a linea di comando:

```
stepinfo(Gcl,'SettlingTimeThreshold',0.05)
ans =
  struct with fields:
```

```
        RiseTime: 5.2621
     SettlingTime: 1.1954e+03
      SettlingMin: 0.0752
      SettlingMax: 1.9617
        Overshoot: 96.1725
       Undershoot: 0
             Peak: 1.9617
         PeakTime: 15.7178
```

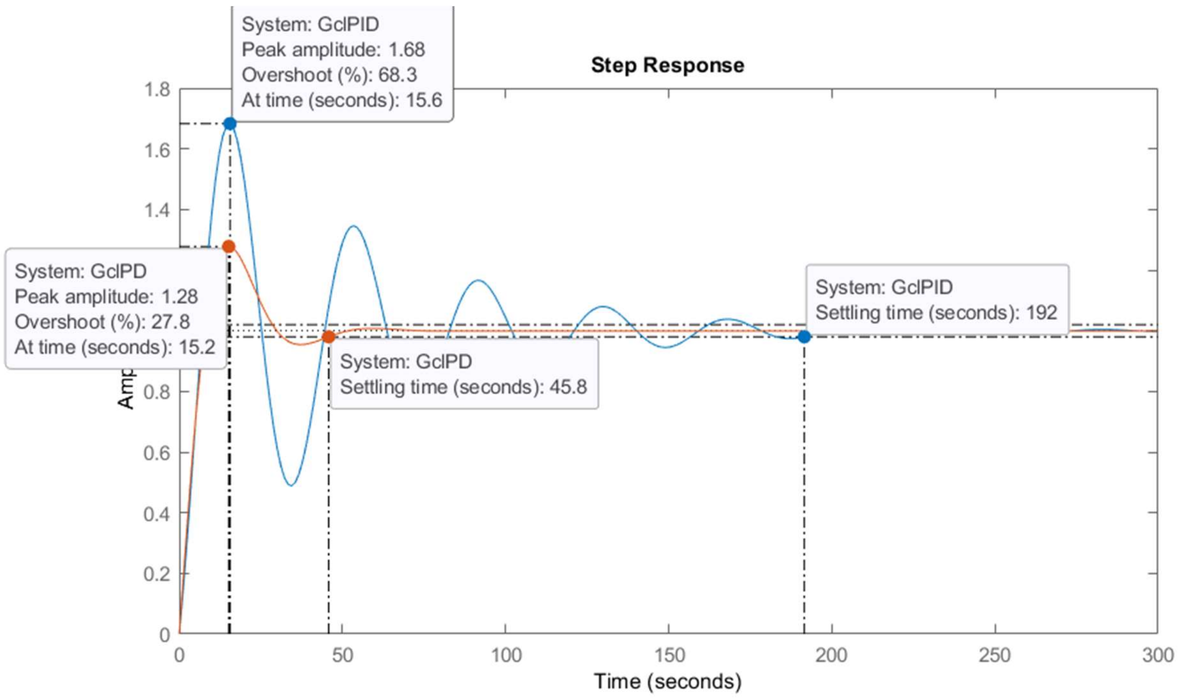
Si noti come il comando **`stepinfo()`** permetta di specificare direttamente come parametro opzionale la soglia al 5% per calcolare il tempo di assestamento.

```
Gcl=feedback(Klim*G,H)
step(Gcl)
% Riduco il tempo del grafico di risposta al gradino unitario per vedere meglio le oscillazioni
step(Gcl,100)
```




```
T0 = 70-42
K0 = Klim

% Costruisco il PID
Kp=0.6*K0
```

```
Ti=0.5*T0
Td=0.125*T0
s=tf('s')
PID=Kp*(1+1/Ti/s+Td*s)

% Costruisco il PD
Kp=0.5*K0
Td=0.2*T0
PD=Kp*(1+Td*s)

% Confronto PID vs PI
GclPID=feedback(PID*G,H)
GclPD=feedback(PD*G,H)

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




```
% Il PD fornisce una prestazione migliore in assoluto, cioè
sia in termini di tempo di assestamento che di overshoot,
entrambi minori di quelli ottenuti con il PID
```