

## Soluzione dell'Esercizio in Laboratorio 8 maggio 2017

Nell'anello digitale viene inserito il modello descritto dalla seguente funzione di trasferimento:

$$G(s) = \frac{0.1}{s(s+0.1)}$$

in cui è presente anche il dispositivo di tenuta di ordine zero (ZOH), con tempo di campionamento  $T = 1s$ .

Si prova inizialmente un semplice regolatore proporzionale,  $D(z) = K$ , per soddisfare le seguenti specifiche sul sistema in catena chiusa:

$$\begin{cases} S\% \leq 18\% \quad (\delta \geq 0.5) \\ T_a \leq 7s \end{cases}$$

in risposta al gradino unitario di riferimento.

Si inseriscono quindi la funzione di trasferimento  $G(s)$  ed il tempo di campionamento  $T$ :

```
>> s=tf('s')
```

```
Transfer function:
```

```
s
```

```
>> T=1
```

```
T =
```

```
1
```

```
>> Gs=0.1/(s*(s+0.1))
```

```
Transfer function:
```

$$\frac{0.1}{s^2 + 0.1 s}$$

Dovendo effettuare il progetto *diretto* del regolatore a tempo discreto, si calcola l'equivalente a tempo discreto  $G(z)$  della funzione di trasferimento  $G(s)$  usando la funzione Matlab `c2d` con l'opzione `'zoh'`:

```
>> Gz=c2d(Gs,T,'zoh')
```

```
Transfer function:
```

$$\frac{0.04837 z + 0.04679}{z^2 - 1.905 z + 0.9048}$$

```
Sampling time (seconds): 1
```

```
>> [numGs,denGs]=tfdata(Gs,'v')
```

```
numGs =
```

$$\begin{matrix} 0 & 0 & 0.1000 \end{matrix}$$

```
denGs =
```

$$\begin{matrix} 1.0000 & 0.1000 & 0 \end{matrix}$$

```
>> [numGz,denGz]=tfdata(Gz,'v')
```

```
numGz =
```

$$\begin{matrix} 0 & 0.0484 & 0.0468 \end{matrix}$$

```
denGz =
```

$$\begin{matrix} 1.0000 & -1.9048 & 0.9048 \end{matrix}$$

```
>> roots(denGz)
```

ans =

1.0000

0.9048

Si è calcolata quindi la funzione di trasferimento a tempo discreto del sistema da controllare, ovvero:

$$G(z) = 0.0484 \frac{z + 0.9672}{(z - 1)(z - 0.9048)}$$

Col luogo delle radici a tempo discreto in Matlab disegnato per la funzione  $G(z)$ , si verifica se esiste un  $K$  in grado di verificare le specifiche, ed in particolare per cui risulti un  $\delta \geq 0.5$  ( $S\% \leq 18\%$ ) che porti a soddisfare anche la specifica del tempo di assestamento  $T_a \leq 7s$ .

Il comando:

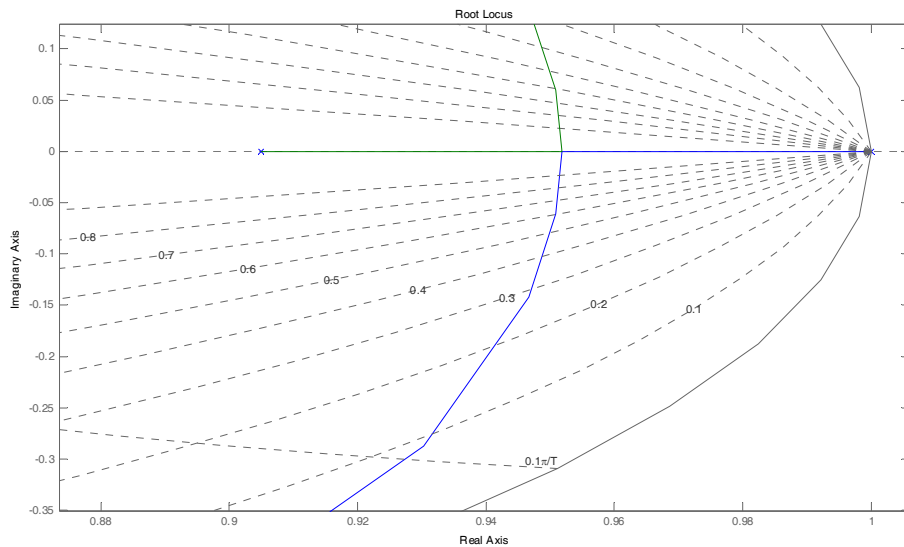
```
>> rlocus(Gz)
```

e successivamente:

```
>> K = rlocfind(Gz)
```

Select a point in the graphics window

porta al grafico seguente, opportunamente ingrandito in prossimità del valore di  $\delta$  cercato:



selezionando quindi il punto come in figura, dopo aver chiamato la funzione Matlab `rlocfind` per  $G(z)$ , si ottiene il risultato seguente, se si sceglie il punto di intersezione del luogo delle radici con il cerchio di raggio unitario:

```
selected_point =  
  
0.8940 + 0.4471i
```

per il valore di  $K$  che corrisponde al valore del guadagno critico. Il guadagno del regolatore deve essere perciò inferiore a questo valore massimo:

```
K =  
  
2.2191
```

Se invece viene chiamata la funzione Matlab `rlocfind` per  $G(z)$ , andando a scegliere il punto di intersezione più vicino del luogo delle radici di  $G(z)$  con il luogo dei punti a  $\delta$  costante e uguale a 0.5, si ottiene il valore di primo tentativo di  $K$  :

```
>> K  
  
K =
```

```
0.0888
```

```
>>
```

che dopo alcuni tentativi (guardando ad esempio come cambiano i valori della risposta per quel valore di  $K$  e il successivo ad esempio pari a  $K = 0.1$ ):

```
>> lsiminfo(yc1,t)
```

```
ans =
```

```
SettlingTime: 85.0166  
           Min: 0  
           MinTime: 0  
           Max: 1.1578  
           MaxTime: 39
```

```
>> K
```

```
K =
```

```
0.0888
```

```
>> K=0.1
```

```
K =
```

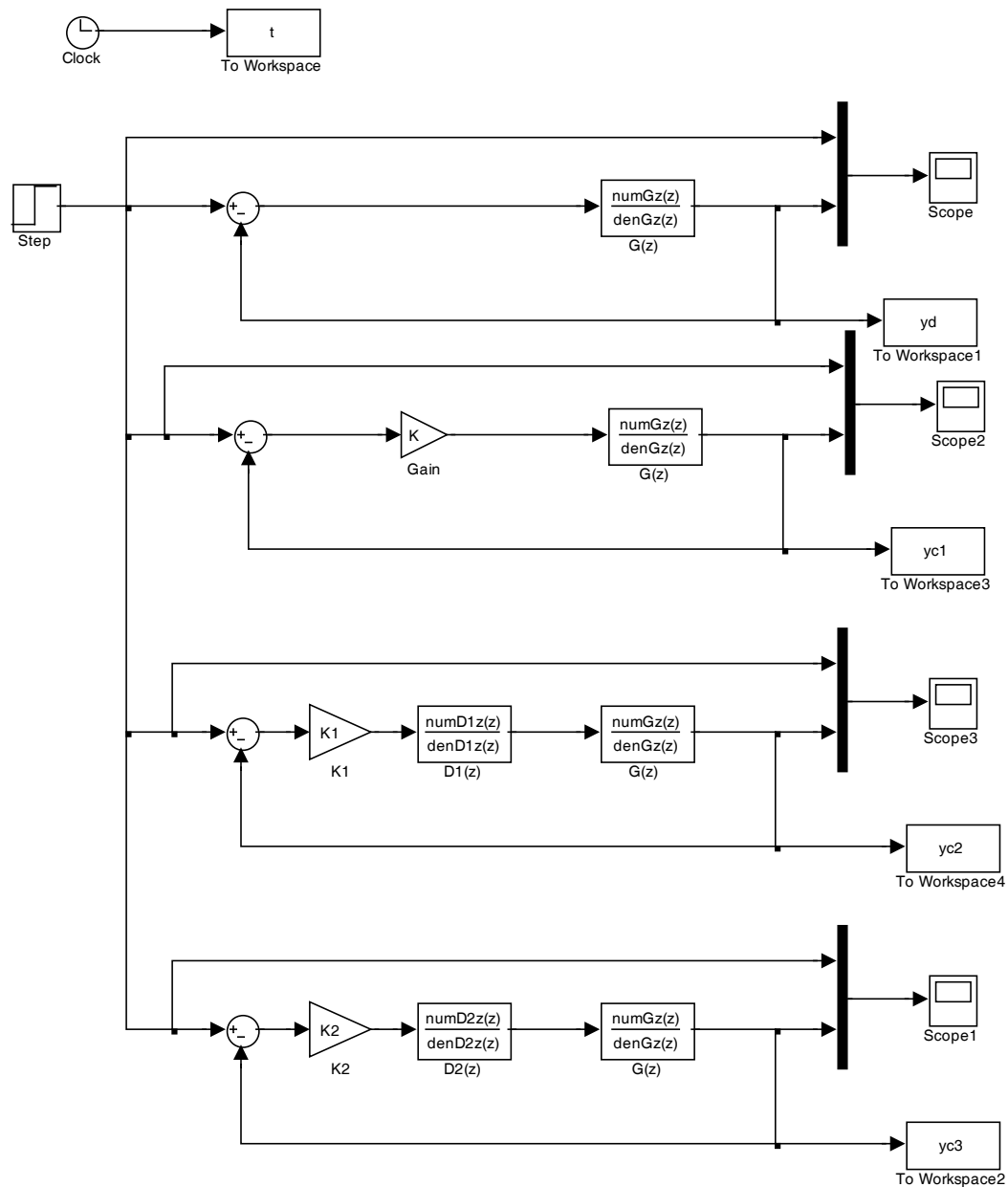
```
0.1000
```

```
>> lsiminfo(yc1,t)
```

```
ans =
```

```
SettlingTime: 82.5946  
           Min: 0  
           MinTime: 0  
           Max: 1.1837  
           MaxTime: 36
```

Tali risultati sono stati ottenuti attraverso lo schema di controllo progettato in Simulink sotto:



Si può concludere quindi che un semplice regolatore proporzionale non è sufficiente per ottenere le specifiche richieste. Si procede quindi a provare con la prima rete correttiva fornita per il progetto:

$$D_1(z) = K_1 \frac{z - 0.9048}{z - 0.4}$$

per la quale sarà necessario determinare il guadagno  $K_1$ .

```
>>
```

```
>> z=tf('z',T)
```

```
Transfer function:
```

```
z
```

```
Sampling time (seconds): 1
```

```
>> D1z=(z-0.9048)/(z-0.4)
```

```
Transfer function:
```

```
z - 0.9048
```

```
-----
```

```
z - 0.4
```

```
Sampling time (seconds): 1
```

```
>>
```

```
>> [numD1z,denD1z]=tfdata(D1z,'v')
```

```
numD1z =
```

```
1.0000 -0.9048
```

```
denD1z =
```

```
1.0000 -0.4000
```

```
>>
```

Si calcola il guadagno di anello per la prima rete correttiva a tempo discreto  $G_{a1}(z) = D_1(z)G(z)$ :

```
>> Ga1z=D1z*Gz
```

```
Transfer function:
```

```
0.04837 z^2 + 0.003019 z - 0.04233
```

```
-----
```

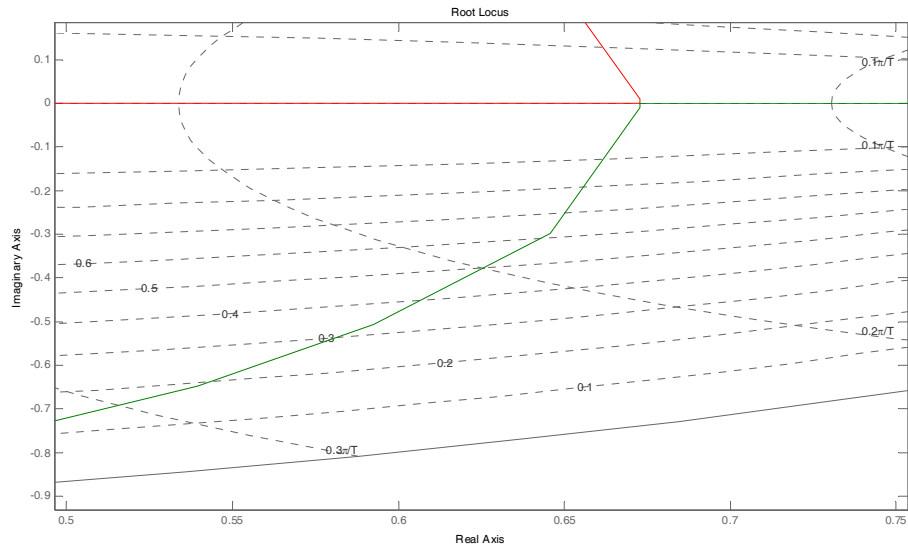
$$z^3 - 2.305 z^2 + 1.667 z - 0.3619$$

Sampling time (seconds): 1

E si calcola il nuovo luogo delle radici per il sistema costituito dalla cascata di  $D_1(z)$  e  $G(z)$ :

```
>> rlocus(Ga1z)
>> zgrid
```

Si effettua un ingrandimento in prossimità del punto di interesse, e si visualizza la seguente figura:



Scelto quindi un punto vicino all'intersezione del luogo delle radici di  $G_{a1}(z)$  ed il luogo a  $\delta$  costante con  $\delta = 0.5$  si ottiene:

```
>> K1=rlocfind(Ga1z)
Select a point in the graphics window
```

selected\_point =

$$0.6270 - 0.3714i$$

K1 =



2.8934

che porta a questi risultati:

```
>> lsiminfo(yc2,t)
```

ans =

```
SettlingTime: 13.0152
           Min: 0
           MinTime: 0
           Max: 1.1589
           MaxTime: 6
```

```
>>
```

Essendo soddisfatta la specifica sulla sovraelongazione ( $S\% < 18\%$ ), ma rimanendo violata invece quella sul tempo di assestamento, posso provare ad aumentare il valore di  $K_1$  di primo tentativo per cercare se esiste un valore ottimale. Purtroppo, provando diversi valori (ad esempio, 3 e 3.5), si vede che si arriva a violare la specifica sulla sovraelongazione, senza soddisfare quella sul tempo di assestamento:

```
>> K1=3
```

K1 =

3

```
>> lsiminfo(yc2,t)
```

ans =

```
SettlingTime: 13.0260
           Min: 0
           MinTime: 0
           Max: 1.1699
           MaxTime: 6
```

```
>> K1=3.5
```

```
K1 =
```

```
3.5000
```

```
>> lsiminfo(yc2,t)
```

```
ans =
```

```
SettlingTime: 12.5074
```

```
Min: 0
```

```
MinTime: 0
```

```
Max: 1.2269
```

```
MaxTime: 5
```

```
>>
```

Si conclude quindi che il regolatore assegnato non è in grado di portare a soddisfare le specifiche per nessun valore di  $K_1$  cercato. Si continua quindi provando il secondo regolatore proposto, che evidentemente tenta di far aumentare la velocità della risposta (ovvero a migliorare il tempo di assestamento) senza peggiorare la sovralongazione. Ti inserisce quindi la rete correttiva che segue:

$$D_2(z) = K_2 \frac{z - 0.88}{z + 0.5}$$

e si vede di determinare un nuovo valore di guadagno  $K_2$ .

```
>> D2z=(z-0.88)/(z+0.5)
```

```
Transfer function:
```

```
z - 0.88
```

```
-----
```

```
z + 0.5
```

```
Sampling time (seconds): 1
```

```
>>
```

```
>> [numD2z,denD2z]=tfdata(D2z,'v')
```

```
numD2z =
```

```
1.0000 -0.8800
```

```
denD2z =
```

```
1.0000 0.5000
```

Si calcola il nuovo guadagno di anello  $G_{a2}(z) = D_2(z)G(z)$ :

```
>>
```

```
>> Ga2z=D2z*Gz
```

```
Transfer function:
```

```
0.04837 z^2 + 0.004219 z - 0.04117  
-----  
z^3 - 1.405 z^2 - 0.04758 z + 0.4524
```

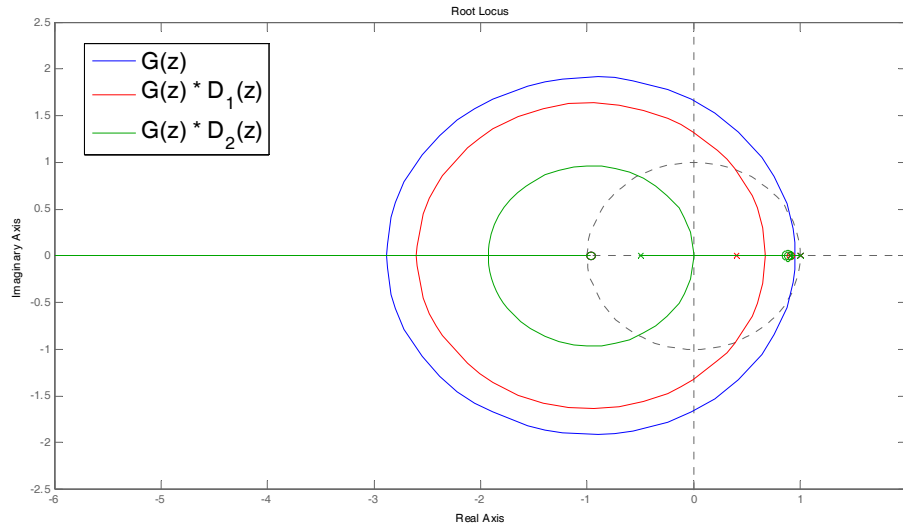
```
Sampling time (seconds): 1
```

```
>>
```

Si osservi l'effetto delle reti correttive introdotte attraverso il comando:

```
>> rlocus(Gz,'b',Ga1z,'r',Ga2z,'g')
```

```
>> legend('G(z)', 'G(z) * D_1(z)', 'G(z) * D_2(z)')
```



Si osservi come partendo da un sistema non compensato, con luogo delle radici molto vicino al cerchio di raggio unitario ( $z = 1$ ), si passa ad un sistema il cui luogo via via che si avvicina al punto  $z = 0$ .

Calcolando successivamente il luogo delle radici per il nuovo sistema compensato dalla seconda rete correttiva  $D_2(z)$ , ovvero  $G_{a2}(z) = D_2(z)G(z)$ , si ottiene:

```
>> rlocus(Ga2z)
>> zgrid
```

Si cerca poi il solito punto del luogo delle radici di  $G_{a2}(z)$  al luogo dei punti con  $\delta = 0.5$  costante:

```
>> K2=rlocfind(Ga2z)
Select a point in the graphics window
```

Select a point in the graphics window

```
selected_point =

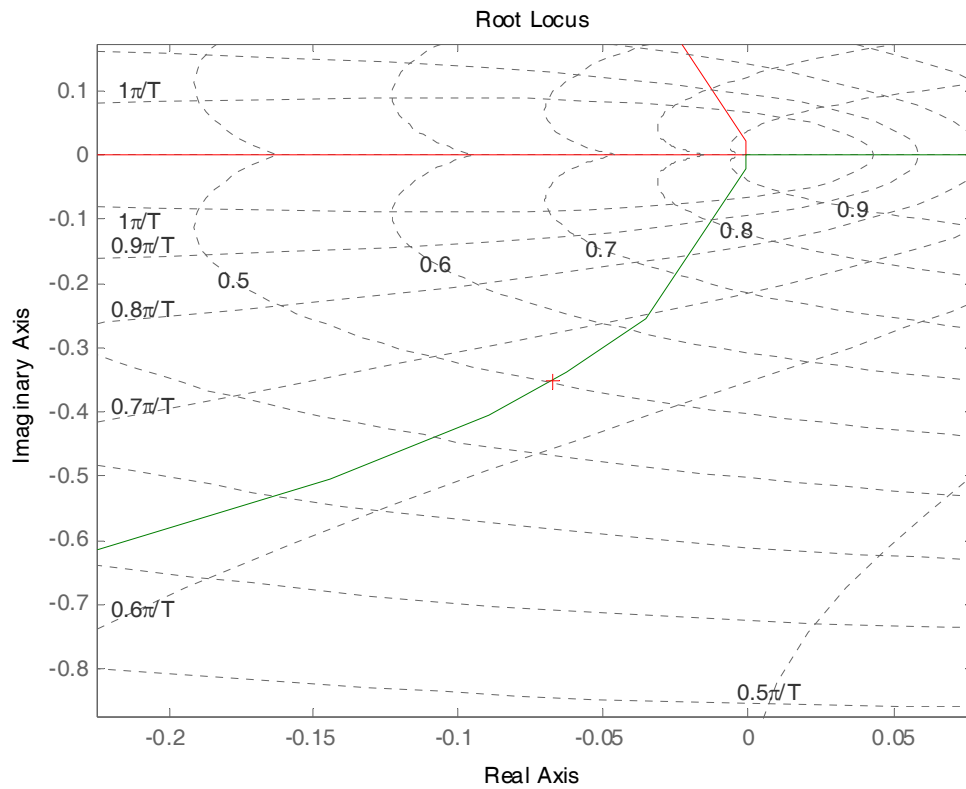
    -0.0677 - 0.3527i
```

```
K2 =
```

13.7317

>>

riportato nel grafico sotto:



a cui corrisponde quel  $K_2$  di primo tentativo, che porta alle seguenti prestazioni:

```
>> lsiminfo(y3,t)
```

ans =

```
SettlingTime: 6.1076  
Min: 0  
MinTime: 0  
Max: 1.2141  
MaxTime: 2
```

>>

Dal momento che la specifica del tempo di assestamento è soddisfatta ( $T_a \leq 7s$ ), mentre quella della sovraelongazione è violata ( $S\% \leq 18\%$ ), si prova a diminuire il guadagno della rete come segue:

```
>> K2=13
```

```
K2 =
```

```
13
```

```
>>
```

e si ottengono le seguenti caratteristiche della risposta al gradino:

```
>> lsiminfo(yC3,t)
```

```
ans =
```

```
SettlingTime: 6.0487
```

```
Min: 0
```

```
MinTime: 0
```

```
Max: 1.1717
```

```
MaxTime: 2
```

```
>>
```

che soddisfano le specifiche richieste dal problema. La rete correttiva da implementare è la seguente:

$$D_2(z) = 13 \frac{z - 0.88}{z + 0.5}$$