

Risoluzione Esercizio Progetto con Metodo Indiretto

Partendo dal sistema da controllare, che viene automaticamente definito dallo script file di Matlab, si ottiene la seguente funzione di trasferimento del terzo ordine:

$$G(s) = \frac{22.22}{s^3 + 100s^2 + 44.61s}$$

che risulta essere anche un *sistema di tipo 1*, quindi presenta errore a regime nullo in risposta al gradino unitario di riferimento. Il sistema di controllo da progettare per $G(s)$ deve essere tale da far soddisfare le seguenti specifiche:

$$\begin{cases} T_a \leq 0.5s \\ S\% \leq 35\% \quad (\delta \geq 0.33) \end{cases}$$

in risposta al gradino unitario di riferimento $r(t)$.

Lanciando lo script di Matlab viene automaticamente definita la funzione di trasferimento del sistema da controllare $G(s)$:

```
Gs_sys = ss(A,B(:,1),C(2,:),D(1,2));
```

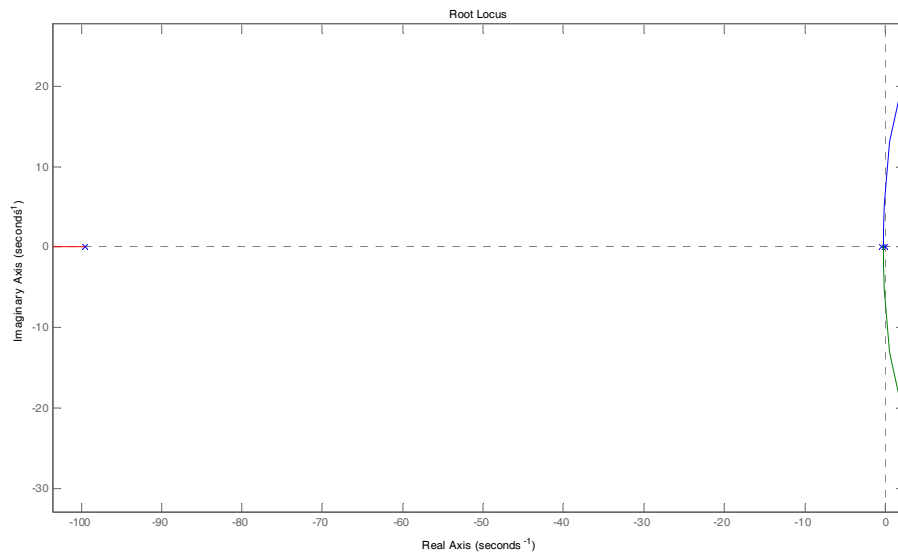
```
[numGs,denGs] = tfdata(Gs_sys,'v');
```

```
Gs = tf(numGs,denGs);
```

essendo già state definite le matrici (A , B , C , D) del modello nello spazio degli stati del motore elettrico il cui albero è accoppiato ad un carico con inerzia.

Si può visualizzare il luogo delle radici del sistema non compensato, che risulta svilupparsi in una zona del semipiano sinistro (RHP) relativamente limitata, come appare dalla figura seguente:

```
>>  
>> rlocus(Gs)  
>>
```



successivamente, mediante il comando Matlab `rlocfind`, si può verificare che comunque il sistema in retroazione unitaria risulta stabile per $0 \leq K \leq K_c$, con $K_c \cong 172.6$, come è evidente dai comandi sotto:

```
>> Kc=rlocfind(Gs)  
Select a point in the graphics window
```

```
selected_point =
```

```
0.0000 + 6.1929i
```

```
Kc =
```

```
172.5982
```

in corrispondenza del punto di intersezione del luogo delle radici di $G(s)$ con l'asse delle ordinate.

Se si calcolano poi le radici del polinomio al denominatore della funzione di trasferimento di $G(s)$, si vede anche che:

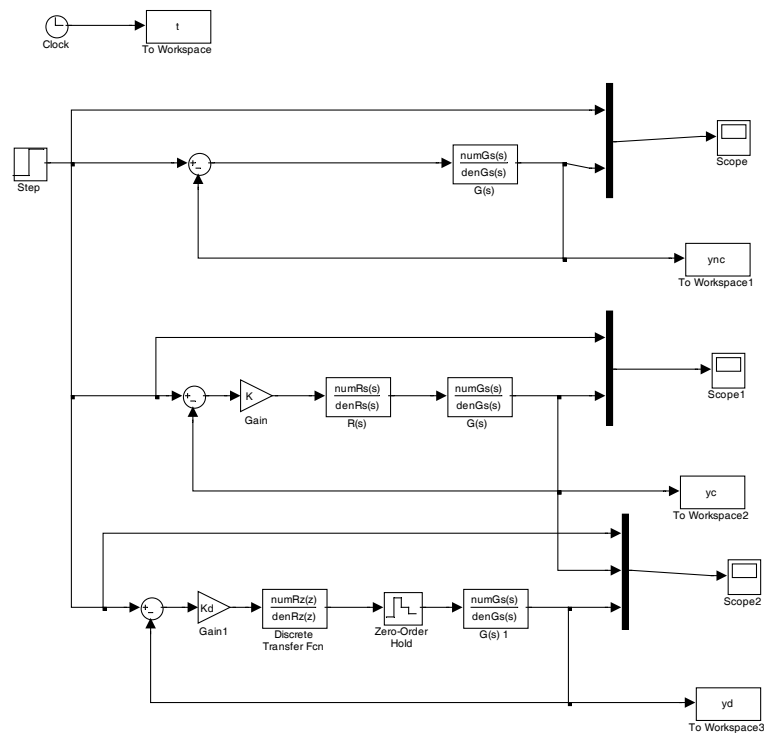
```
>> roots(denGs)
```

```
ans =
```

```
0  
-99.5536  
-0.4481
```

```
>>
```

che evidenzia come il sistema elettromeccanico da controllare sia intrinsecamente (semplicemente) stabile e di tipo 1. Inoltre, attraverso lo schema Simulink riportato nella figura successiva, in riferimento al primo schema:



si può verificare il tempo di assestamento e la sovravelongazione del sistema da controllare.

```
>> lsiminfo(ync,t)
```

```
ans =
```

```
SettlingTime: 17.5024
```

```
Min: 0
```

```
MinTime: 0
```

```
Max: 1.1861
```

```
MaxTime: 7.5500
```

```
>>
```

Questi non soddisfano le specifiche richieste, in particolare il tempo di assestamento T_a (che deve essere $\leq 0.5s$). Si può facilmente verificare che, anche se eventualmente venisse aumentato il guadagno proporzionale per $G(s)$, non sarebbe comunque possibile soddisfarle entrambe. Si procede quindi alla correzione di $G(s)$ attraverso rete di tipo *anticipatrice*, trattandosi di sistema già con tre poli.

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

```
>> Rs=(1+s/10)/(1+s/150)
```

```
>> [numRs,denRs]=tfdata(Rs,'v')
```

```
>>
```

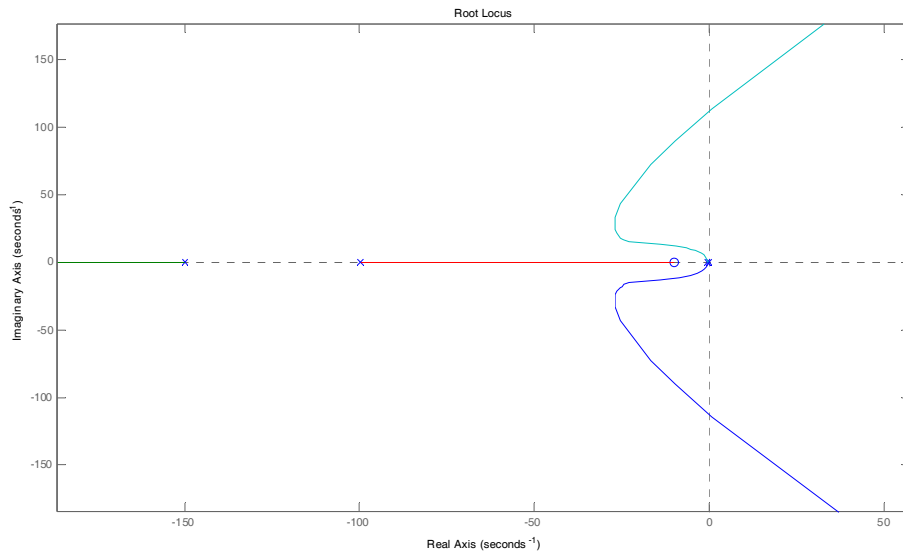
Si definisce quindi la funzione guadagno di anello $G_a(s)$ come segue:

```
>> Ga=Rs*Gs
```

e successivamente si disegna il luogo delle radici:

```
>> rlocus(Ga)
```

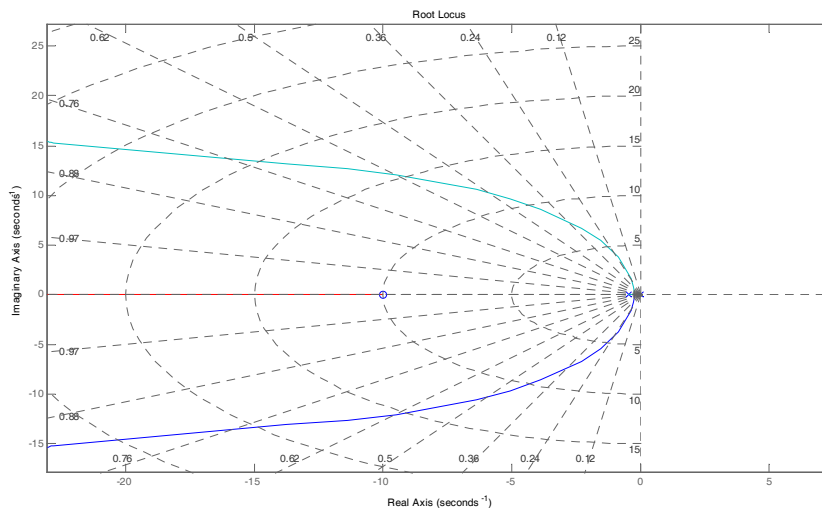
da cui ottengo:



Si osservi l'effetto dell'inserimento dello zero della rete correttiva, che è responsabile della deformazione e successivo spostamento del luogo delle radici iniziale del sistema compensato verso sinistra ("effetto attrattivo dello zero"), a cui corrisponde anche un miglioramento del tempo di assestamento T_a . Si ricordi infatti che il luogo dei punti a tempo di assestamento costante è descritto

dalla relazione $T_a \cong \frac{3}{\delta \omega_n}$, con δ parte reale dei poli complessi coniugati dominanti. Si disegna

quindi il luogo dei punti a δ costante con il comando `sgrid`, e si ottiene il seguente grafico ingrandito sul punto di interesse ($\delta \geq 0.3$):



e col comando Matlab `rlocfind`, col mouse vado a trovare l'intersezione tra il luogo del sistema compensato e la retta corrispondente a $\delta \approx 0.3$:

```
>> K=rlocfind(Ga)
```

da cui ottengo:

Select a point in the graphics window

```
selected_point =
```

```
-2.6707 + 7.1717i
```

```
K =
```

```
240.8861
```

```
>>
```

Se in Simulink si riduce la durata della simulazione a 5 secondi, si apprezza maggiormente il beneficio ottenuto, e si vanno a valutare le caratteristiche della risposta al gradino del sistema compensato con la rete correttiva per il valore di K così determinato:

```
>> lsiminfo(yc,t)
```

```
ans =
```

```
SettlingTime: 1.3845
```

```
Min: 0
```

```
MinTime: 0
```

```
Max: 1.4264
```

```
MaxTime: 0.3500
```

```
>>
```

A partire da quel valore iniziale di K di primo tentativo, provo ad aumentare il guadagno, ad esempio:

```
>> K=800
```

```
K =
```

```
800
```

```
>>
```

```
>> lsiminfo(yc,t)
```

```
ans =
```

```
SettlingTime: 0.4258
```

```
Min: 0
```

```
MinTime: 0
```

```
Max: 1.3210
```

```
MaxTime: 0.1500
```

Si osservi che si ottiene un miglioramento decisivo di T_a , anche se marginale, ma soddisfa le specifiche ($T_a \leq 0.5s$); la sovralongazione risulta invece pienamente soddisfatta ($S\% \leq 35\%$). Si può provare eventualmente un nuovo valore di K , ad esempio:

```
>> K=1300
```

```
K =
```

```
1300
```

```
>> lsiminfo(yc,t)
```

ans =

SettlingTime: **0.2387**

Min: 0

MinTime: 0

Max: **1.3250**

MaxTime: 0.0950

>>

Quindi per il valore di $K = 1300$ vengono verificate entrambe le specifiche. Posso ora progettare il regolatore a tempo discreto, una volta definito il tempo di campionamento più opportuno.

Ricordando la relazione empirica $T \cong \frac{T_a}{100}$, appare adeguata la scelta di $T = 0.005s$:

>> T=0.005

Posso discretizzare quindi il regolatore a tempo continuo $R(s)$, secondo il metodo di Tustin, attraverso la funzione c2d come segue:

>> Rz=c2d(Rs,T,'tu')

>> [numRz,denRz]=tfdata(Rz,'v')

Transfer function:

11.18 z - 10.64

z - 0.4545

Sampling time (seconds): 0.005

numRz =

11.1818 -10.6364

denRz =

1.0000 -0.4545

>>

Si ottiene quindi il seguente regolatore equivalente a tempo discreto secondo il metodo di Tustin, che si otterrebbe da $R(s)$ con la relazione:

$$R(z) = R(s) \Big|_{s=\frac{2}{T} \frac{z-1}{z+1}} = K_d \frac{11.18z - 10.64}{z - 0.4545}$$

e come primo tentativo si pone $K_d \equiv K$.

Si analizzano quindi le prestazioni del sistema a tempo discreto in termini di tempo di assestamento e massima sovraelongazione per $K_d \equiv K = 1300$:

>> lsiminfo(yd,t)

ans =

SettlingTime: 0.2106

Min: 0

MinTime: 0

Max: **1.3814**

MaxTime: 0.0950

>>

Si osservi come venga ora violata la specifica sulla sovraelongazione, evidentemente dovuto all'inserimento del dispositivo di tenuta di ordine zero, che produce un peggioramento dei margini di stabilità, e quindi ad un aumento della sovraelongazione.

Si procede quindi a diminuire il guadagno K_d , per provare a riottenere il soddisfacimento delle specifiche anche per il sistema a tempo discreto.

Si ricordi infatti che una riduzione del guadagno proporzionale comporta, in generale, una diminuzione della massima sovraelongazione, ma anche un aumento del tempo di assestamento. Tuttavia, il peggioramento di questo parametro non viola la condizione su T_a , con $T_a \leq 0.5s$, essendo $T_a = 0.21s$ per il sistema controllato a tempo discreto. Si prova quindi a ridurre il valore di K_d , e dopo pochi tentativi, si determina che con:

```
>> Kd=800
```

```
Kd =
```

```
800
```

```
>>
```

```
>> lsiminfo(yd,t)
```

```
ans =
```

```
SettlingTime: 0.4273
```

```
Min: 0
```

```
MinTime: 0
```

```
Max: 1.3501
```

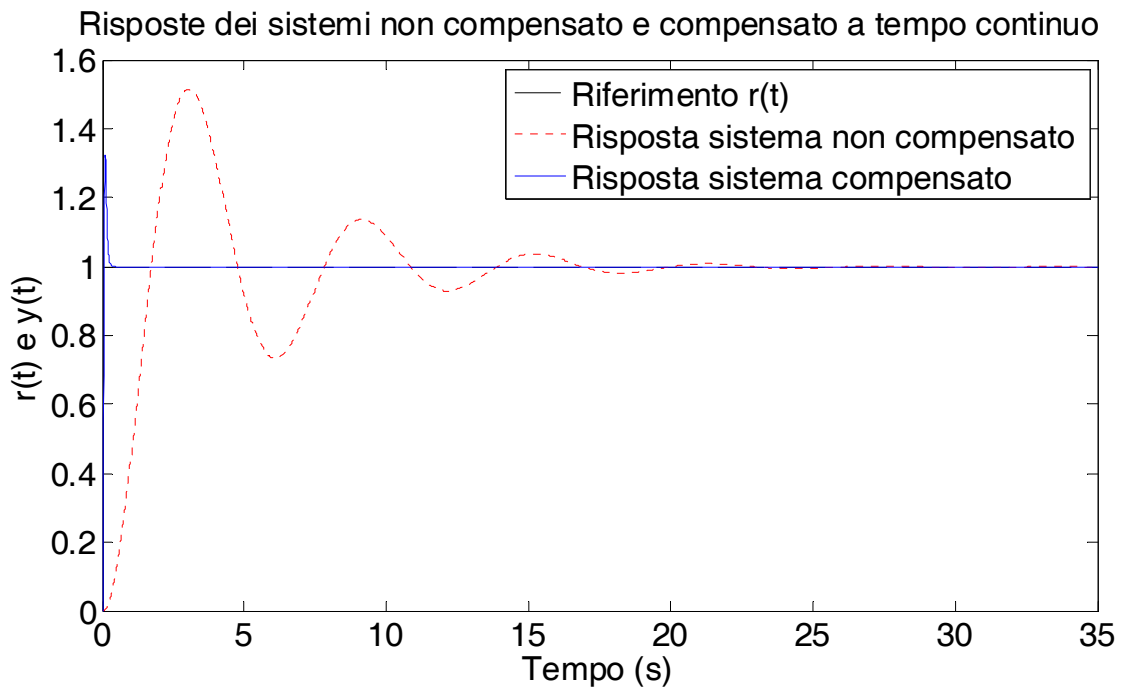
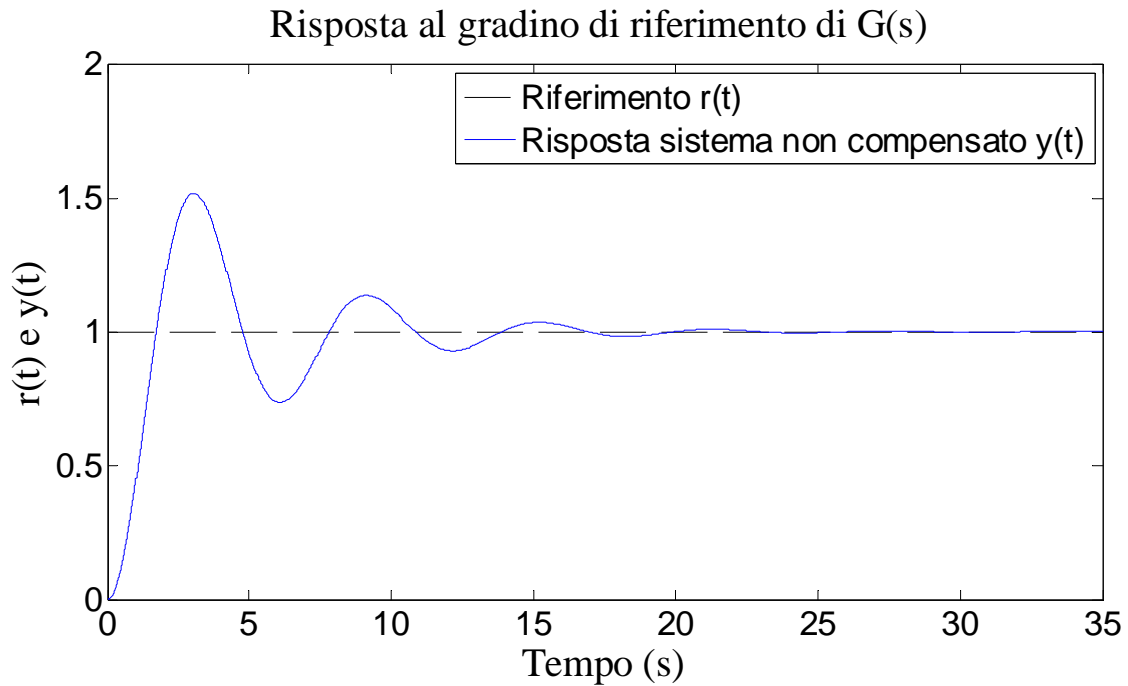
```
MaxTime: 0.1450
```

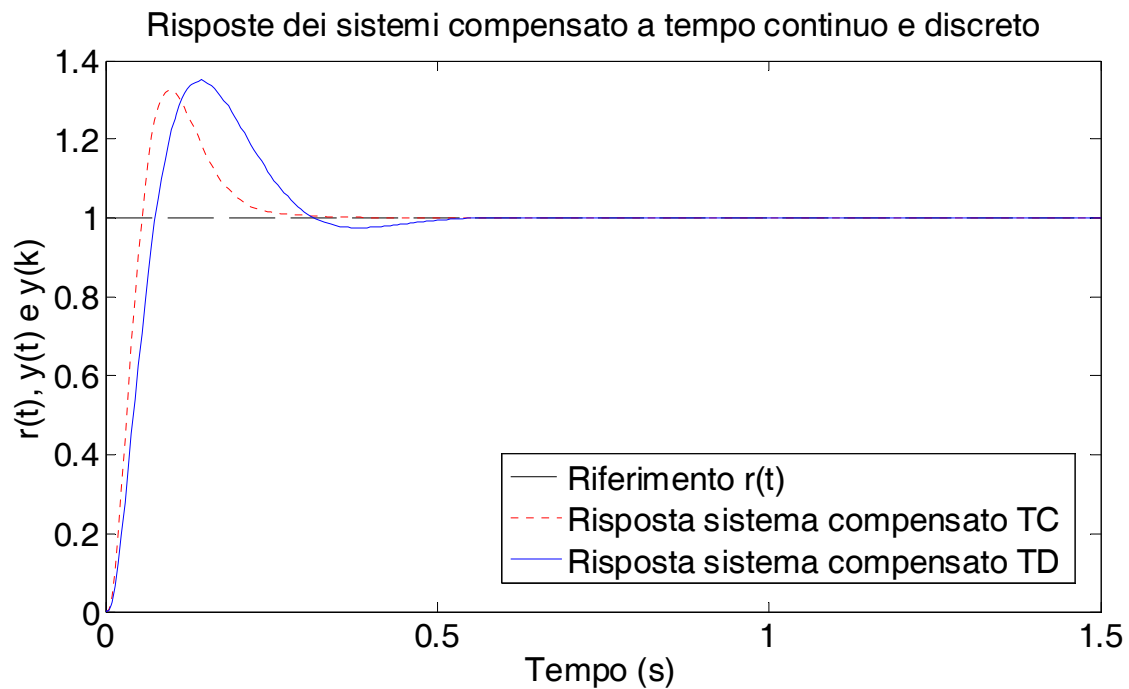
```
>>
```

```
>>
```

entrambe le specifiche sono soddisfatte.

I grafici delle risposte al gradino sono le seguenti, rispettivamente per il sistema non compensato, il sistema non compensato confrontato con il sistema compensato a tempo continuo, ed il sistema compensato a tempo continuo confrontato con il sistema compensato a tempo discreto.





Si conclude quindi che a volte risulta necessario *ricalibrare* il guadagno della rete correttiva a tempo discreto $R(z)$, quando il progetto viene inizialmente eseguito a tempo continuo con successiva discretizzazione del regolatore determinato a tempo continuo. Tale procedimento è noto come progetto *indiretto* del regolatore, che, se da un lato sfrutta il vantaggio della semplicità del metodo eseguito tutto a tempo continuo, dall'altra invece, attraverso la semplice discretizzazione del regolatore e relativa implementazione, non è in grado di tenere conto *a priori* della presenza del dispositivo di tenuta di ordine zero.