

Laboratorio Matlab – Fondamenti di Automatica

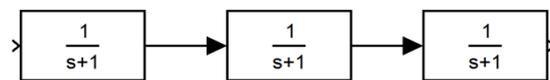
Controllo della portata di uscita di una rete di serbatoi – Soluzioni

Per risolvere questo esercizio occorre tradurre lo schema a blocchi dato nel testo in uno schema Simulink (digitare `simulink` al prompt di Matlab e creare un nuovo modello con Blank Model).

Sapendo che i blocchi 1, 2 e 3 hanno funzione di trasferimento data da:

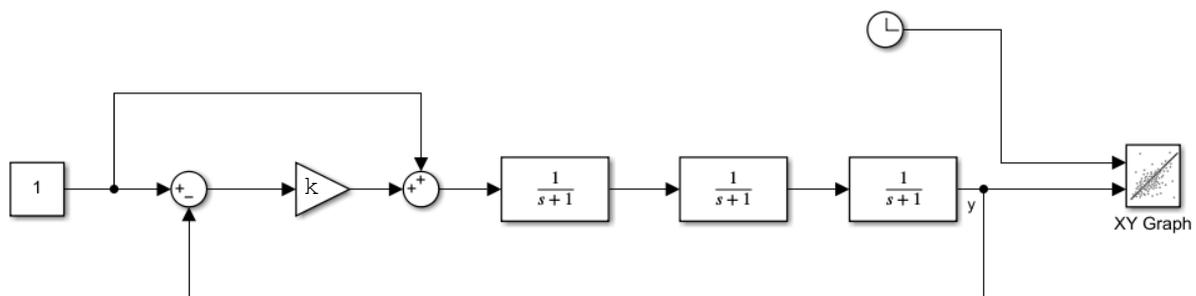
$$G(s) = \frac{1}{s+1}$$

per rappresentare lo schema precedente in Simulink si può utilizzare il blocco Transfer Fcn per tre volte connettendo i blocchi in serie (Library Browser → Simulink → Continuous → trascinare l'oggetto Transfer Fcn nel modello):



Per sommare o sottrarre due segnali esiste il nodo sommatore (Commonly Used Blocks → Sum); per cambiare i segni del nodo sommatore occorre entrare nelle proprietà dello stesso e specificarli nel campo “list of signs”.

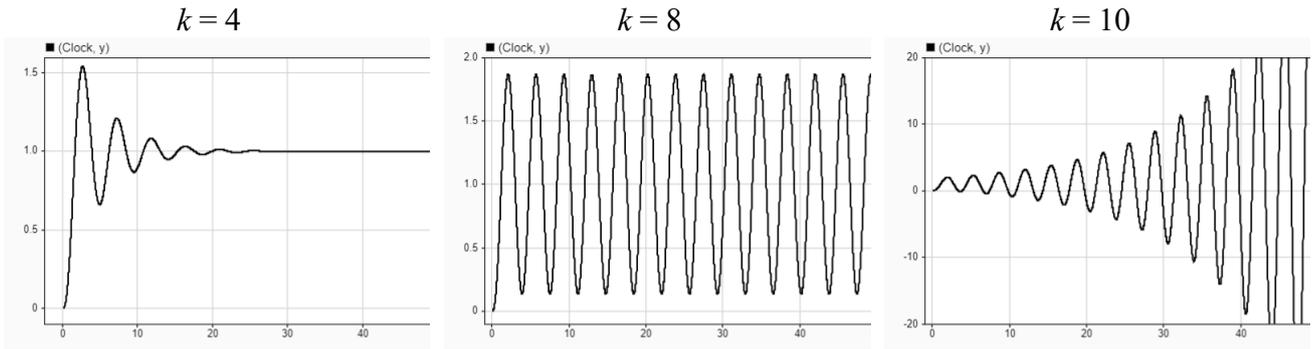
Infine, per moltiplicare un segnale (guadagno “k”) si utilizza un blocco denominato Gain (Commonly Used Blocks → Gain; entrare nelle proprietà del blocco e definire il valore pari a k (tale valore verrà poi definito di volta in volta nel workspace di Matlab)). L'ingresso, come richiesto dal testo, sarà una costante con valore 1 (Sources → Constant). Si può pertanto costruire il seguente schema:



Si noti che è stato inserito il blocco Clock (Sources → Clock) per generare il tempo da inserire nel blocco XY (Sinks → XYGraph) in cui verrà visualizzata la portata di uscita dal terzo serbatoio nel tempo. Simulare il sistema per 50 unità di tempo (Stop Time nella finestra Simulation).

La richiesta dell'esercizio è di confermare che i calcoli svolti a lezione siano corretti, ovvero che il limite di stabilità del sistema sia raggiunto per $k = 8$. Per fare ciò svolgiamo 3 simulazioni con 3 diversi valori di k ($k < 8$, $k = 8$, $k > 8$).

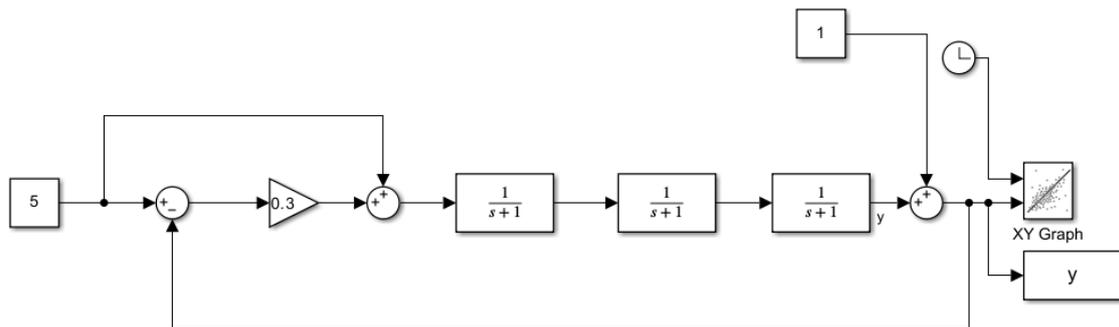
Si ottengono i seguenti risultati (definire $k=4$ al prompt di Matlab; simulare il modello con Run e visualizzare il risultato con doppio click su XY Graph; ripetere per $k=8$ e $k=10$).



(Per cambiare la visualizzazione dei grafici, selezionare la finestra `Format` del grafico)

Come si nota il sistema risulta asintoticamente stabile per $k < 8$, semplicemente stabile per $k = 8$ e perde di stabilità per $k > 8$.

Per valutare l'andamento dell'uscita all'equilibrio al variare di k (nel caso di asintotica stabilità), fissiamo l'ingresso pari a 5 e aggiungiamo in uscita (tramite un nodo sommatore) il disturbo costante pari a 1 (`Sources` → `Constant`). Aggiungiamo inoltre un blocco che ci permetta di ottenere nel `Workspace` di Matlab la portata di uscita (`Sinks` → `To Workspace`, nominando con y la variabile di uscita e selezionando `array` alla voce `Save format`).



Ripetiamo diverse simulazioni per differenti valori di k riportando il valore dell'uscita di regime corrispondente in una tabella. Il risultato ottenuto è il seguente:

k	0	0.3	0.6	1	2	3	4	5	6	7
<i>uscita</i>	6	5.769	5.625	5.5	5.33	5.25	5.2	5.166	5.143	5.125

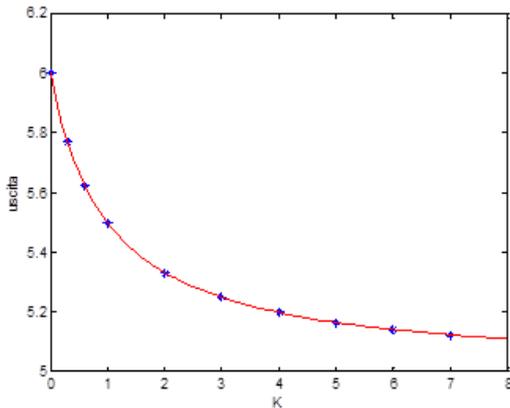
NOTA: per ottenere tali valori utilizzare i comandi:

```
uscita=out.y;
```

```
Yeq=uscita(length(uscita)) % ultimo punto della simulazione≈uscita di equilibrio
```

I valori ottenuti (asterischi blu) ben approssimano il risultato ottenuto in teoria (linea rossa)

$$\bar{y} = \bar{u} + \frac{\bar{d}}{1+k}$$



```
vettorek=[0 0.3 0.6 1 2 3 4 5 6 7];
uscita=[6 5.769 5.63 5.5 5.33 5.25 5.2 5.166 5.143 5.125];
figure; plot(vettorek,uscita,'b*'); hold on;

k=[0:0.1:8];
yeq=5+1./(1+k);
plot(k,yeq, 'r ');
xlabel('k'); ylabel('uscita');
```

Pertanto, per ridurre l'effetto del disturbo sull'uscita occorre portare k a valori elevati, avvicinandosi però così al limite di stabilità per il sistema.

Per valutare il tempo di risposta ($T_R = -\frac{5}{\Re(\lambda_D)}$, con λ_D = autovalore dominante, cioè l'autovalore con parte reale maggiore) occorre scrivere le equazioni che definiscono il sistema. Queste sono date da:

$$\begin{aligned}\dot{x}_1 &= -x_1 + u = -x_1 + y^0 + k(y^0 - y) = -x_1 + y^0 + k(y^0 - x_3 - d) \\ &= -x_1 - kx_3 + y^0(1+k) - kd\end{aligned}$$

$$\dot{x}_2 = -x_2 + x_1$$

$$\dot{x}_3 = -x_3 + x_2$$

$$y = x_3 + d$$

da cui $A = \begin{vmatrix} -1 & 0 & -k \\ 1 & -1 & 0 \\ 0 & 1 & -1 \end{vmatrix}$

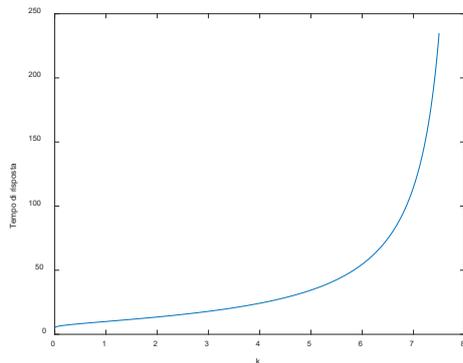
Valutiamo il tempo di risposta al variare del parametro k

```
par_k=[]; par_TR=[];
for k=0:0.01:7.5
    A=[-1 0 -k;1 -1 0;0 1 -1];
    reale_lD=max(real(eig(A)));
    TR=-5/reale_lD;
    par_k=[par_k;k];
    par_TR=[par_TR;TR];
end;
figure;
plot(par_k,par_TR);
xlabel('k'); ylabel('Tempo di risposta');
```

In alternativa, per ogni k in tabella, valutiamo A , valutiamo λ , $\Re(\lambda_D)$, $T_D = -\frac{1}{\Re(\lambda_D)}$ e $T_R = 5T_D$. Per esempio, per $k = 2$

```
k=2; A=[-1 0 -k;1 -1 0;0 1 -1];
reale_lD=max(real(eig(A)));
TD=-1/reale_lD;
TR=5*TD
```

k	0	1	2	3	4	5	6	7	7.5
T_R	5	10	13.51	17.93	24.23	34.48	54.68	124.85	234.93



Oppure, per punti:

```
vettorek=[0 1 2 3 4 5 6 7 7.5];
tempoR=[5 10 13.51 17.93 24.23 34.48 54.68 124.85 234.93];
figure; plot(vettorek,tempoR,'b*');
```

Pertanto, aumentando k , pur compensando meglio l'effetto del disturbo sull'uscita, il tempo di risposta aumenta (al limite della stabilità, la parte reale dell'autovalore dominante tende a zero e il tempo di risposta tende a infinito).