Università degli Studi di Napoli Federico II
Dipartimento di Informatica e Sistemistica
Gruppo di ricerca SINCRO

Esercitazioni di Automazione Industriale

M. di Bernardo e S. Santini

a cura di A. Tierno

Sommario

Modulo 1: Sintesi di un controllo a retroazione di stato in Matlab

Come noto dalla teoria, dato un sistema completamente controllabile, è possibile attraverso una retroazione lineare dello stato piazzare i poli del sistema a ciclo chiuso dovunque nel piano complesso. In particolare considerato il sistema $\dot{x} = Ax + Bu$ e la retroazione lineare dello stato $u = -Kx$ il sistema a ciclo chiuso diventa $\dot{x} = (A - BK)x$ e la matrice K consente di assegnare i poli a ciclo chiuso.

Consideriamo il sistema descritto dalle seguenti equazioni:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -2x_2 + u \\ y &= x_1 \end{aligned}$$

Che, assunta x_1 un a posizione e x_2 una velocità, può ad esempio rappresentare il modello della sola parte meccanica di un motore a corrente continua.

Vogliamo piazzare, tramite retroazione dello stato, i poli del sistema a ciclo chiuso in $-2 \pm j$.

<u>**Svolgimento**</u>.

Cominciamo a studiare il sistema a ciclo aperto.

Studiamone innanzitutto la controllabilità. Possiamo farlo attraverso il comando **ctrb** (digitare dal prompt Matlab “help ctrb” per maggiori dettagli. La sintassi è la seguente:

```
>> A = [0 1; 0 -2];
>> B = [0 1]';
>> C = [1 0];
>> D = 0;
>> sys = ss(A,B,C,D)

a =
        x1  x2
   x1    0   1
   x2    0  -2

b =
        u1
   x1    0
   x2    1

c =
        x1  x2
   y1    1   0

d =
        u1
   y1    0

Continuous-time model.
>> co = ctrb(A,B)

co =

     0     1
     1    -2
```

Oppure in maniera equivalente e più veloce:

```
>> A = [0 1; 0 -2];
>> B = [0 1]';
>> co = ctrb(A,B)

co =

     0     1
     1    -2
```

In ogni caso il comando ctrb ci restituisce la matrice di controllabilità. In base al criterio di controllabilità ne deduciamo che il sistema è controllabile.

Possiamo inoltre calcolarne autovalori ed auto vettori mediante il comando **eig**, secondo la seguente sintassi:

```
>> [T,D] = eig(A)

T =

    1.0000   -0.4472
         0    0.8944


D =

     0     0
     0    -2
```

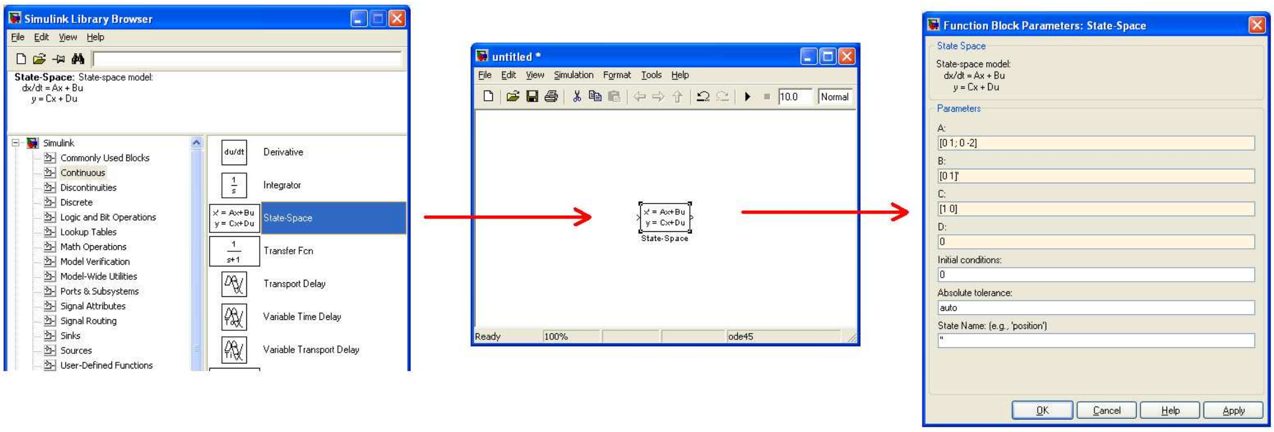
O più semplicemente:

```
>> eig(A)

ans =

     0
    -2
```

Possiamo inoltre valutare la risposta indiciale del sistema in esame (ovvero la risposta ad un segnale a gradino unitario posto in ingresso).

Prima di fare ciò realizziamo però anche il modello Simulink.

Avviamo Simulink e apriamo un nuovo documento. Per realizzare il modello Simulink nello spazio di stato possiamo usare il blocco “State-Space” della libreria “Continuous” e cliccando due volte su tale blocco specificare le matrici che descrivono il sistema in esame.



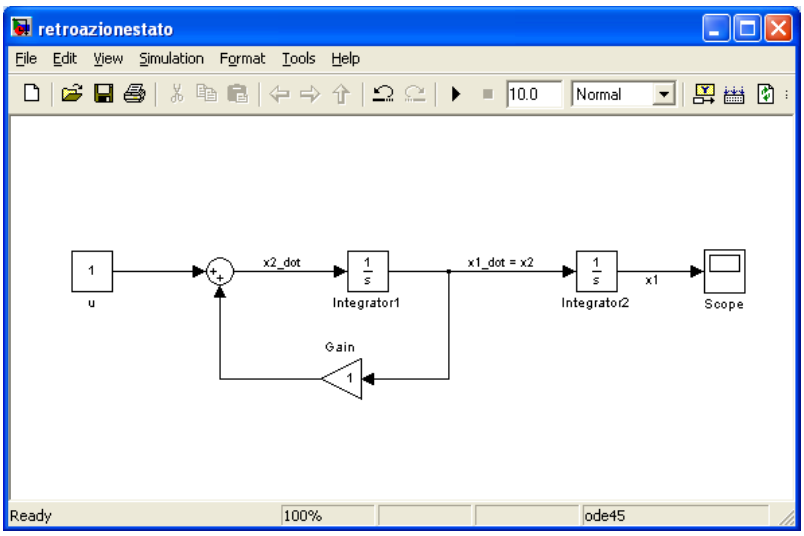
Oppure possiamo realizzare il modello mediante integratori.

Lo si può fare attraverso pochi semplici passi.

1. Aprire Simulink → Nuovo modello.

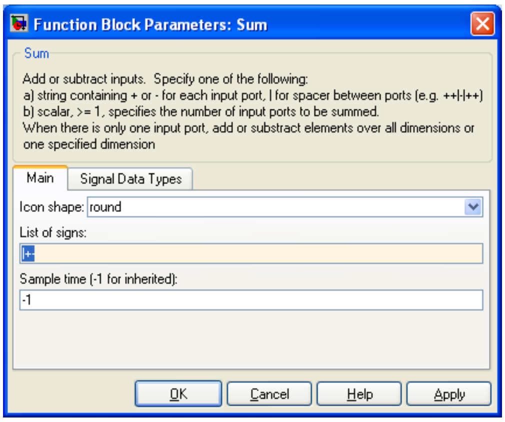
2. Collocazione di blocchi e linee.

    a. Dalla libreria "Continuous" trascinare il blocco "Integrator" (ne servono 2).
    b. Dalla libreria "Math operations" serve un blocco sommatore.
    c. Ancora dalla libreria "Math operations" trascinare un blocco "Gain" (si noti che nella figura che segue tale blocco è stato ruotato, puramente per un fatto estetico).
    d. Dalla libreria "Sources" trascinare un blocco "Constant" che costituirà il nostro ingresso (successivamente può essere sostituito).
    e. Dalla libreria "Sinks" trascinare il blocco "Scope" necessario per visualizzare luscita.
    f. A questo punto, in accordo con le equazioni che descrivono il sistema, collegare i blocchi come illustrato nella figura seguente:

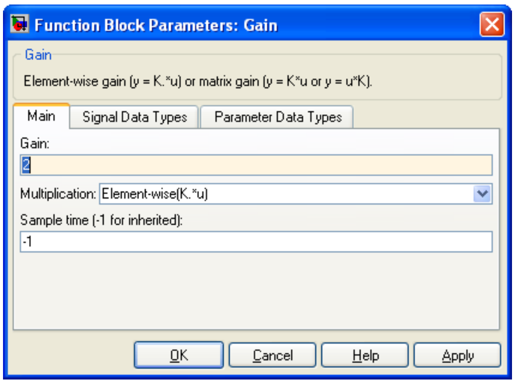


3. Modifica dei blocchi.

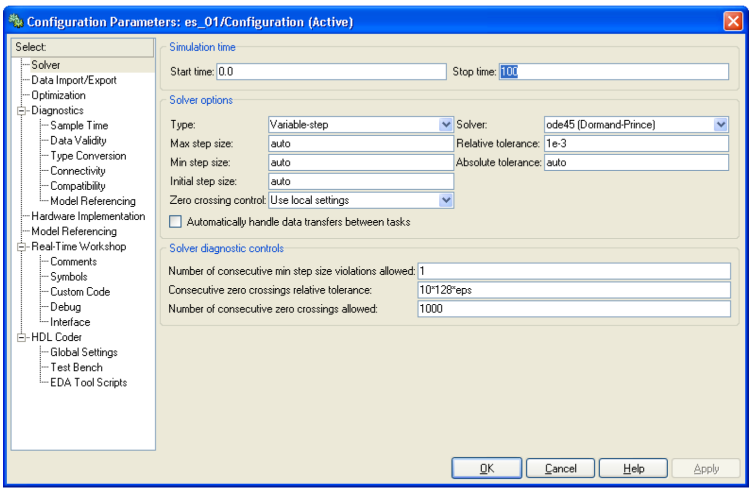
   a. Dobbiamo innanzitutto modificare il blocco sommatore: cliccandoci sopra possiamo cambiare la sequenza dei segni, a noi ne serve uno positivo ed uno negativo.



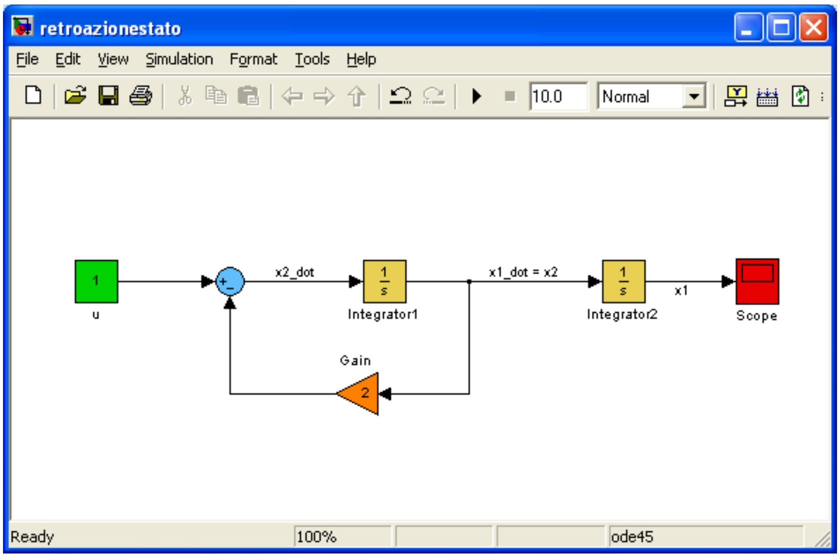
   b. Poi dobbiamo modificare il blocco Gain, sostituendo il valore di default 1 con il valore 2.



4. Impostare i parametri della simulazione (Simulation → Configuration Parameters…): possiamo per esempio allungare il tempo di simulazione per avere la certezza che il sistema vada a regime (si noti che il tempo di simulazione si può anche impostare direttamente dalla barra dei comandi).



Alla fine il sistema si presenta così (i blocchi costituenti sono stati colorati per aumentarne la leggibilità):

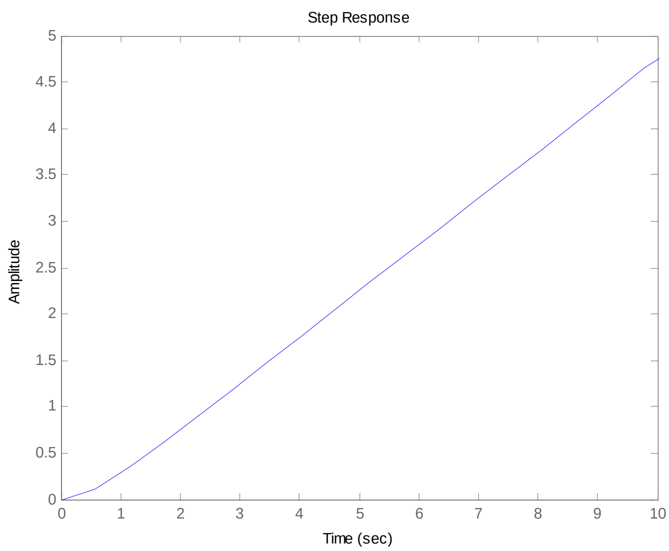


Il nostro obiettivo è valutare la risposta al gradino, possiamo fare ciò sia dal prompt di Matlab attraverso il comando **step**, che dal Simulink utilizzando il modello appena costruito.

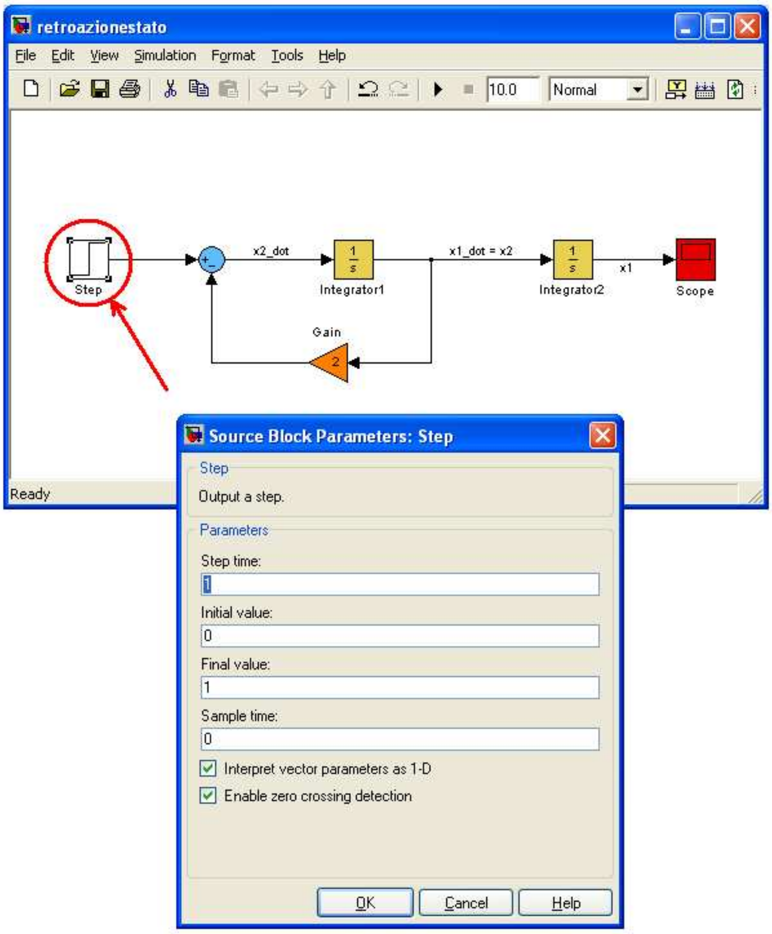
Dal Matlab la sintassi è la seguente:

```
>> step(sys)
```

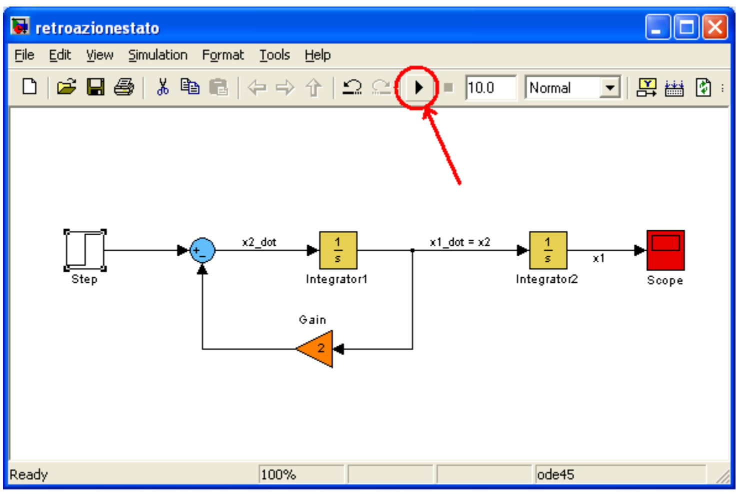
Otterremo così il diagramma della risposta indiciale; se gli assi non sono opportunamente tarati, possiamo modificare il plot a nostro piacimento usando il menu a tendina “Edit → Axes proprieties…”.



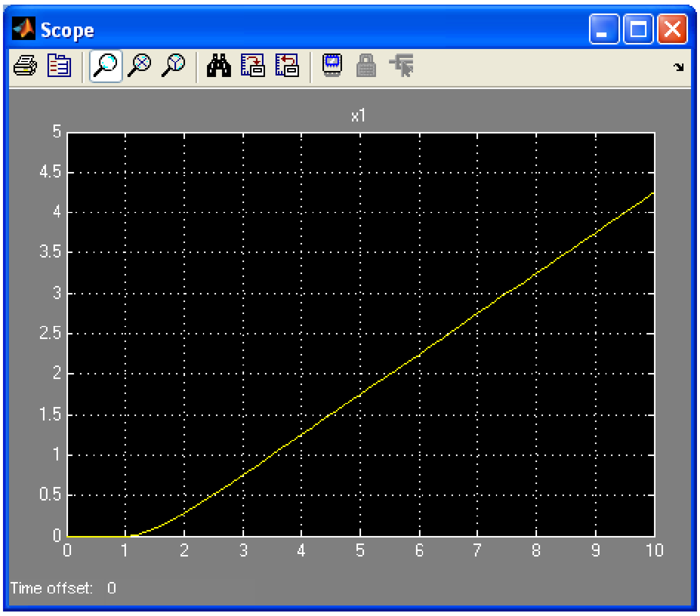
Per utilizzare invece il modello Simulink, dobbiamo prima sostituire il blocco in ingresso con il blocco “Step” reperibile dalla libreria “Sources”. Cliccando poi due volte su tale blocco appare una finestra di pop-up che ci permette di impostare a nostro piacimento i parametri del segnale a gradino:



Avviamo la simulazione (attraverso il menu a tendina "Simulation" oppure cliccando sul bottone "Start simulation" :



Cliccando sul blocco "Scope" possiamo visualizzare l'uscita:



A questo punto in genere si effettua la scelta degli auto valori per rientrare in opportune specifiche; nel nostro caso gli auto valori sono già assegnati: $-2 \pm j$.

Per il posizionamento dei poli i comandi più usati (vale anche per i sistemi discreti) sono:

```
K = PLACE(A,B,P)
K = ACKER(A,B,P)
```

che ci forniscono la matrice dei guadagni di retroazione K tali che i poli del sistema a ciclo chiuso coincidano con quelli specificati nel vettore P. In particolare il comando place si usa anche per sistemi con più ingressi (tutti considerati come ingressi di controllo), mentre acker solo per sistemi con un ingresso.

Usando per esempio il primo comando, si ottiene:

```
>> K = place([0 1; 0 -2], [0 1]', [-2+j -2-j])

K =

     5     2
```

Andiamo a verificare in Matlab/Simulink.

Il sistema adesso ha la forma seguente:

$$\dot{x} = (A - BK)x$$

dove la matrice $A - BK$ è calcolata di seguito:

```
>> A

A =

     0     1
     0    -2

>> B

B =

     0
     1

>> K = [5 2]

K =

     5     2

>> A-B*K

ans =

     0     1
    -5    -4
```

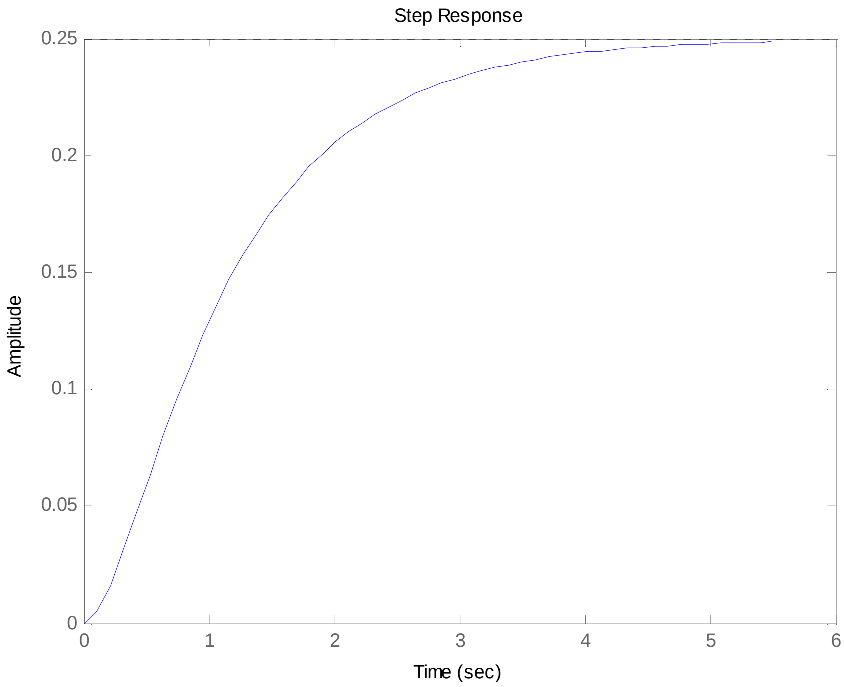
Valutiamo dunque gli autovalori:

```
>> eig(ans)

ans =

  -2.0000 + 1.0000i
  -2.0000 - 1.0000i
```

Sono proprio quelli che volevamo assegnare.

Ricalcoliamo la risposta indiciale.

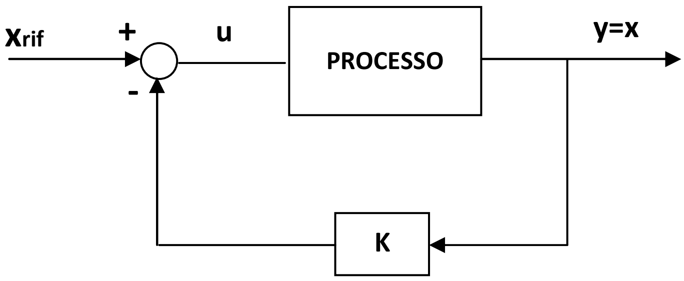
```
A =

     0     1
    -4    -5

>> B, C, D

B =

     0
     1


C =

     1     0


D =

     0

>> sys=ss(A,B,C,D)

a =
        x1   x2
   x1    0    1
   x2   -4   -5

b =
        u1
   x1    0
   x2    1

c =
        x1   x2
   y1    1    0

d =
        u1
   y1    0

Continuous-time model.
>> step(sys)
```



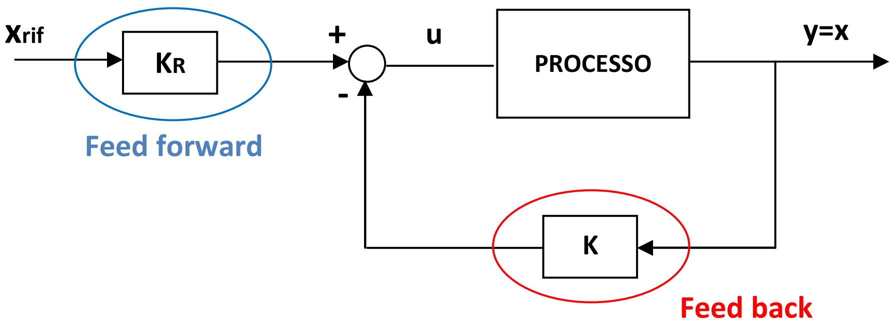
Anche se in questo caso non risulta necessario, in generale è possibile un ulteriore miglioramento della soluzione ottenuta.

Per esempio, quando il riferimento non è nullo, il *pole placement* non è sufficiente a soddisfare le specifiche a regime. In particolare a regime avremo un errore pari a:

$$e = x_{rif} - \overline{x} = x_{rif} + (A - BK)^{-1} B x_{rif} = [I + (A - BK)^{-1} B] x_{rif}$$



Per risolvere almeno in parte questo problema si può far uso di un'azione aggiuntiva in avanti (*Feed forward*):



In tal caso (FB+FF) avremo:

$$\dot{x} = Ax + Bu = Ax + B(K_R x_{rif} - Kx) = (A - BK)x + BK_R x_{rif}$$

Quindi a regime:

$$0 = (A - BK)\overline{x} + BK_R x_{rif} \Rightarrow \overline{x} = -(A - BK)^{-1} BK_R x_{rif}$$

Dunque scegliendo opportunamente K_R possiamo ridurre l'errore a regime.

Esercizi

1. Dato il sistema seguente

$$\begin{cases} \dot{x} = \begin{pmatrix} -1 & 0 \\ 2 & -3 \end{pmatrix} x + \begin{pmatrix} 1 \\ 0 \end{pmatrix} u \\ y = \begin{pmatrix} 1 & 0 \end{pmatrix} x \end{cases}$$

   progettare con il metodo di Ackermann un controllore che posizioni gli autovalori a -10 e -20.

2. Dato il sistema costituito da un doppio integratore, si progetti la retroazione di stato che porti i poli del sistema a ciclo chiuso in $-1 \pm j$.

3. Si consideri il sistema descritto dalle equazioni seguenti:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} -1 & 1 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} u$$

   e, utilizzando il comando *place* si determini il guadagno di retroazione di stato tale da portare i poli del sistema a ciclo chiuso in (-3, 2) e successivamente in (-1, -2). Perché il comando usato fornisce un segnale di errore soltanto nel secondo caso?

4. Si consideri il sistema:

$$x(k+1) = Ax(k) + Bu(k) = \begin{bmatrix} 0 & -\frac{1}{4} \\ -3 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ -\frac{1}{2} \end{bmatrix} u(k)$$

   Si desiderano assegnare gli auto valori ½ e ¼.

5. Si consideri il seguente sistema:

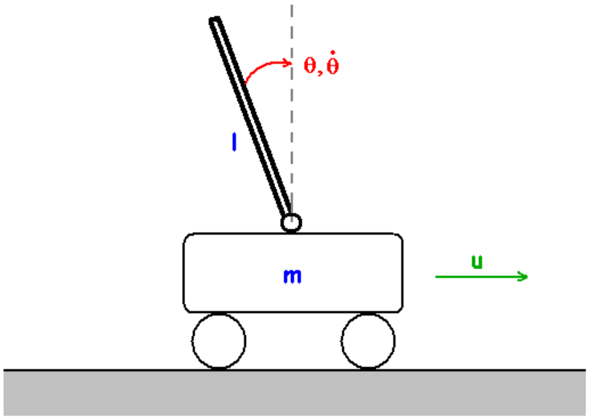
$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} -2 & 1 \\ 0 & -4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \end{pmatrix} u$$

$$y = (1 \quad -1) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

   Si progetti un controllore tale che il sistema a ciclo chiuso abbia poli in $(-1 \pm j, -10)$.

Modulo 2: Simulazione di un controllore CC in Simulink

Il pendolo inverso



Poniamo (trascurando gli stati del carrello):

$$x_1 = \delta\vartheta$$
$$x_2 = \delta\dot{\vartheta}$$

Risulta:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = 2x_1 + x_2 + u \end{cases}$$

Riscriviamo il modello del pendolo inverso in forma matriciale:

$$\begin{cases} \dot{x} = \begin{pmatrix} 0 & 1 \\ 2 & 1 \end{pmatrix} x + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u \\ y = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} x \end{cases}$$

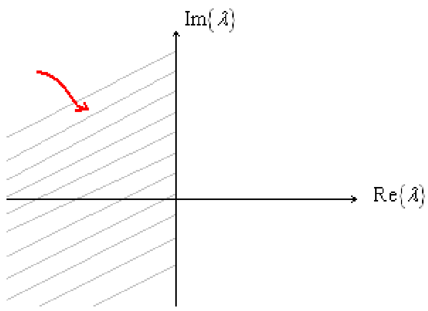
Supponiamo che il sistema a C.A. sia instabile. Vogliamo progettare un'azione di controllo u(x,t) tale da stabilizzare il pendolo nella posizione invertita, ovvero in modo che risulti: $\mathrm{Re}\left\{\hat{\lambda}_1, \hat{\lambda}_2\right\} < 0$.

Svolgimento.

Se sono disponibili le misure di tutto lo stato del sistema, possiamo generare l'ingresso di controllo moltiplicando le misure dello stato per un guadagno statico.

1. Tradurre le specifiche in termini degli autovalori di $\hat{A}$, ovvero in termini del suo polinomio caratteristico.

   Graficamente vogliamo che gli autovalori siano nel semipiano sinistro:



Se andiamo a calcolare gli auto valori della matrice dinamica, notiamo che tale specifica non è soddisfatta:

```
A =

     0     1
     2     1

>> A = [0 1; 2 1];
>> eig(A)

ans =

    -1
     2
```

2. Sintesi del controllore
    a. Scelta del tipo di controllore.

    Scegliamo:

$$u = -Kx = -[k_1 \quad k_2]\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = -k_1x_1 - k_2x_2;$$

    b. Analisi del C.C. (regolazione dei guadagni).

    Dobbiamo scegliere k_1 e k_2.

$$\begin{cases} \dot{x} = Ax - Bu \\ u = -kx \end{cases} \Rightarrow \dot{x} = (A - Bk)x = \hat{A}x$$

    In particolare possiamo scegliere k_1 e k_2 in modo da posizionare gli auto valori di $\hat{A}$ in una locazione desiderata. Supponiamo ad esempio di voler riposizionare i poli nei punti -1 e -10; ricorriamo nuovamente al comando pole:

```
>> A = [0 1; 2 1];
>> B = [0 1]';
>> C = [1 0; 0 1];
>> D = 0;
>> K = place([0 1; 2 1], [0 1]', [-1 -10])

K =

   12.0000   12.0000

>> Ac = A - B*K

Ac =

         0    1.0000
  -10.0000  -11.0000
```

    In tal modo fissiamo k_1 e k_2 in modo da rendere stabile un punto di equilibrio instabile.

Verifichiamo :

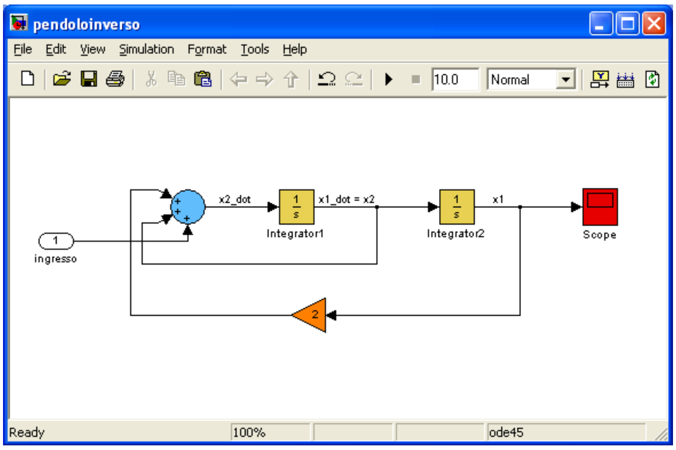
```
>> eig(Ac)

ans =

    -1
   -10
```

Dunque: $\hat{\lambda}_1 = -1, \hat{\lambda}_2 = -10$.

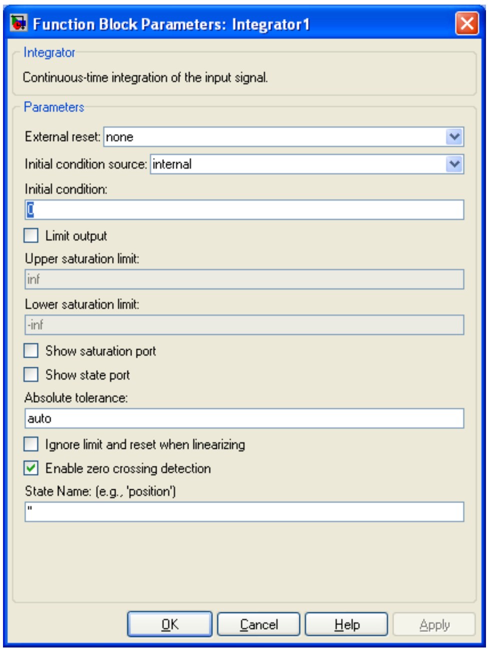
3. Simuliamo adesso il controllore CC in Simulink.

   Per la costruzione del modello si segue l'algoritmo visto in precedenza:

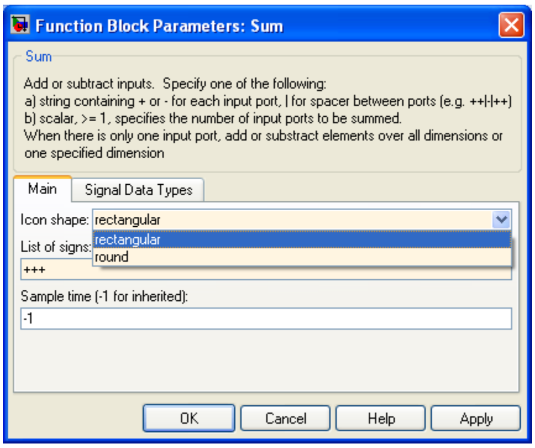
   a. Collocazione di blocchi e linee: in accordo con le equazioni del sistema, avremo bisogno di due blocchi integratore, di uno sommatore, di uno scope per la visualizzazione dell'uscita e di un blocco gain. Effettuando gli opportuni collegamenti di blocchi e linee, lo schema appare come nella figura che segue.



   b. Modifica dei blocchi.
      - Cliccando sui blocchi integratore due volte possiamo impostare le condizioni iniziali.
      - Per un fatto puramente estetico possiamo cambiare la forma del sommatore in rettangolare.
      - Possiamo aggiungere un blocco "clock" dalla libreria "Sources" che fornisce il tempo corrente di simulazione.
      - Per lavorare sulle variabili in gioco possiamo usare dei blocchi "To Workspace" della libreria "Sinks", che ci permettono di salvare le variabili nel Workspace di Matlab.
      - Possiamo mettere come ingresso al sistema un blocco step, reperibile nella libreria "Sources".

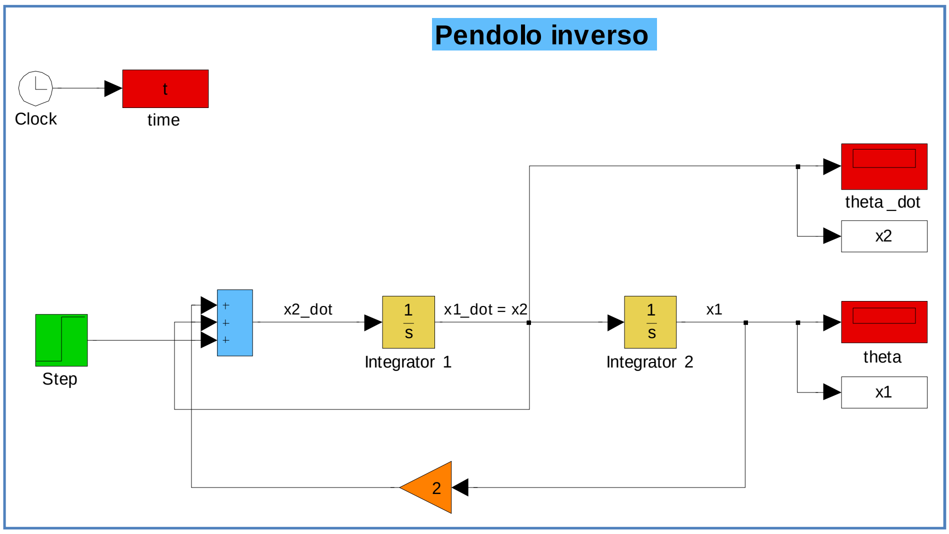


impostazione condizioni iniziali

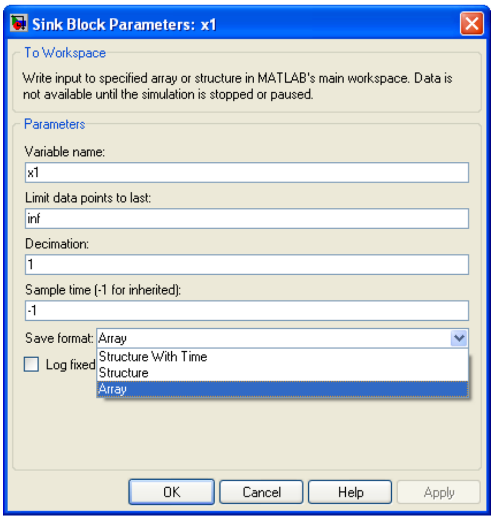


modifica blocco sommatore

Lo schema Simulink del sistema in esame appare dunque come segue:

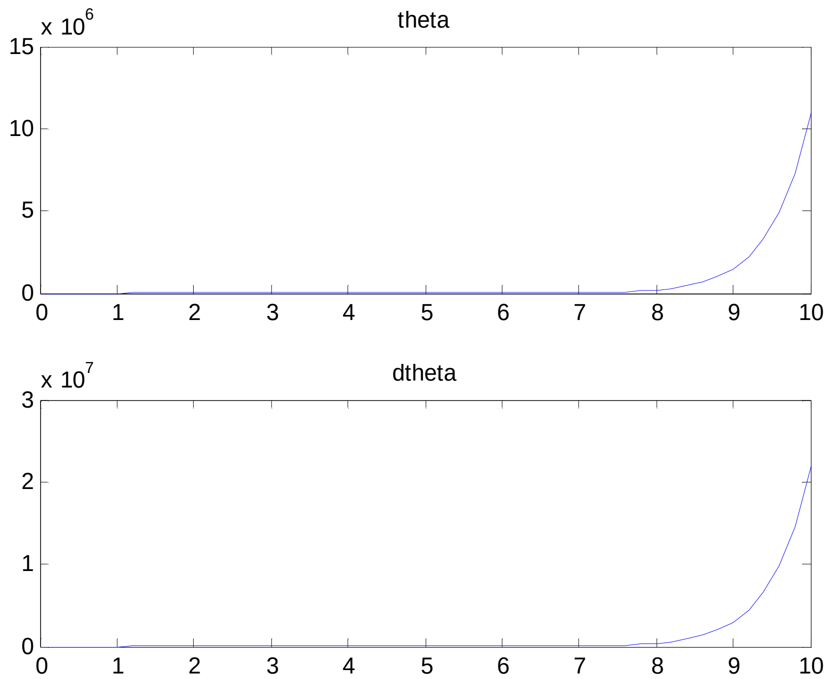


N.B: quando si usano i blocchi “To Workspace” ci si assicuri che il formato in cui vengono salvati i dati sia di tipo “array” (cliccare due volte sul blocco e selezionare la voce “Array” dal menu a tendina “Save format”):

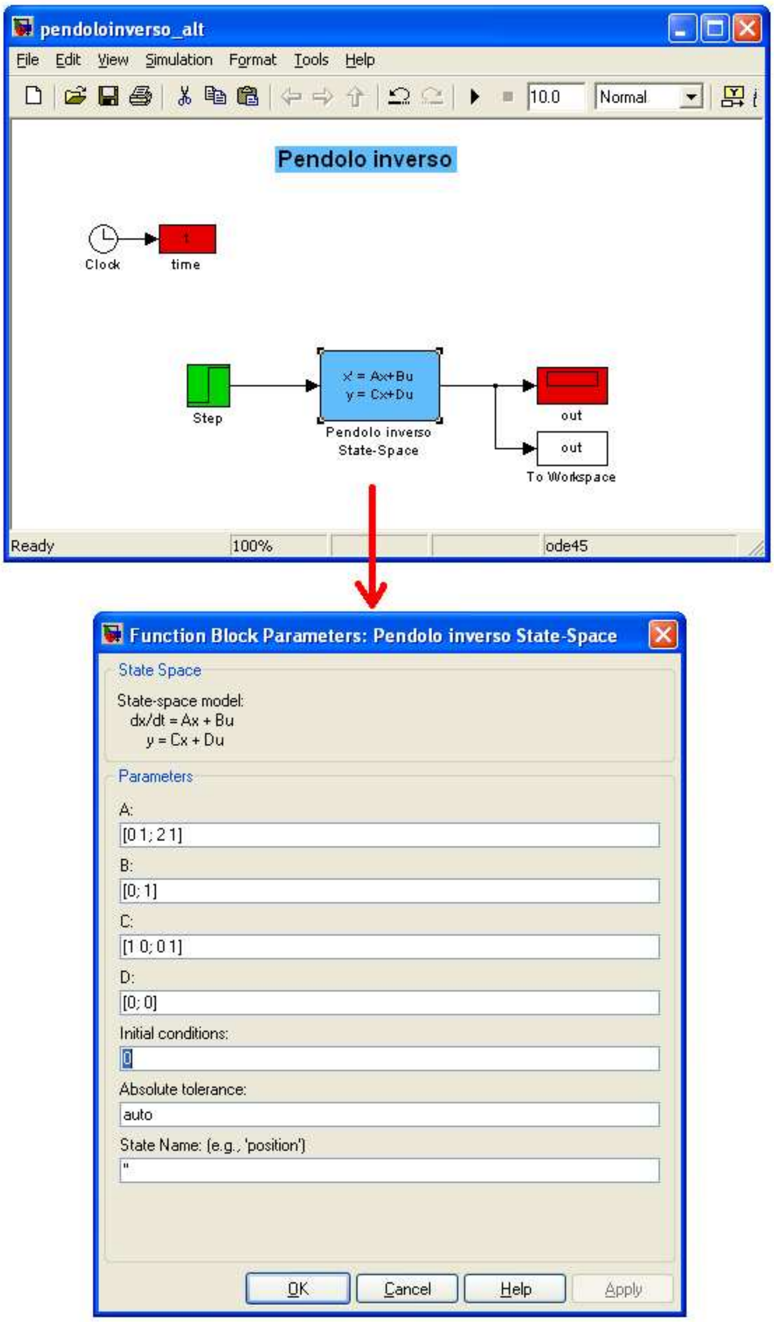


Avviando la simulazione per un ingresso a gradino unitario il sistema fornisce l’uscita seguente:

```
>> figure
>> subplot(2,1,1);
>> plot(t,x1)
>> title('theta')
>> subplot(2,1,2);
>> plot(t,x2)
>> title('dtheta')
```

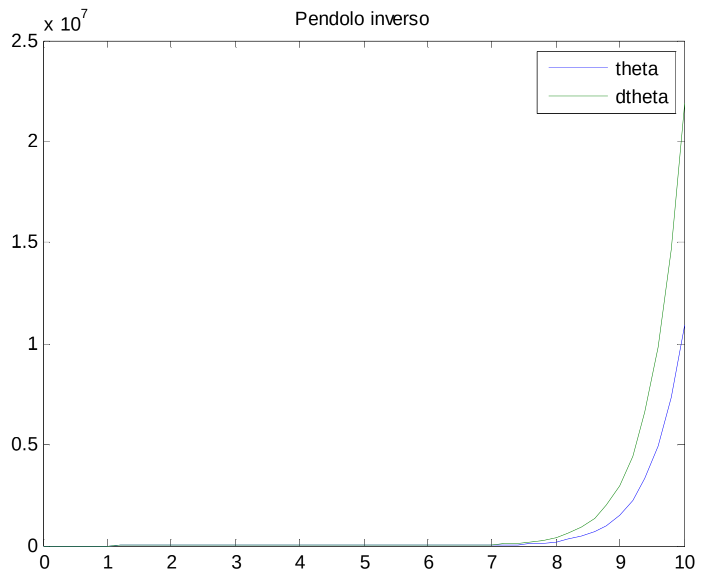


In alternativa a costruire lo schema con i blocchi integratori, possiamo usare il blocco "State-Space" della libreria "Continuous", che ci permette di specificare il sistema da modellare in termini delle matrici A, B, C e D:



Avviando la simulazione e diagrammando i risultati si osservano infatti gli stessi andamenti per le due uscite.

```
>> figure
>> plot(t,out)
>> legend('theta','dtheta')
>> title('Pendolo inverso')
```

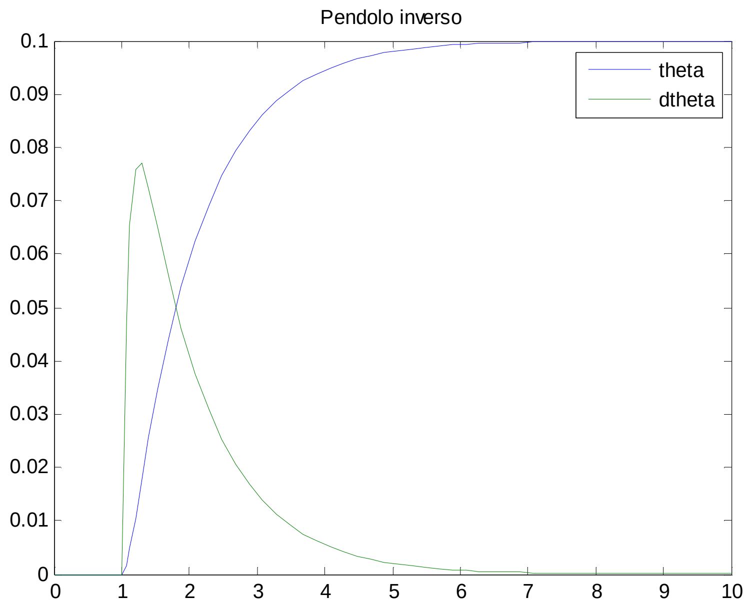


Considerando lo schema costruito con il blocco State-Space, andiamo a modificare la matrice dinamica, che in seguito al pole-placement è diventata:

$$\dot{x} = (A - Bk)x = \begin{pmatrix} 0 & 1 \\ -10 & -11 \end{pmatrix} x$$

Considerando ancora come ingresso un gradino unitario, vediamo come si comportano le uscite:

```
>> figure
>> plot(t,out)
>> legend('theta','dtheta')
>> title('Pendolo inverso')
```



Come si evince dai risultati, grazie al pole-placement abbiamo reso stabile un sistema che non lo era.

Esercizi

1. Dato il seguente sistema:

$$\begin{cases} \dot{x} = \begin{pmatrix} -2 & 1 \\ -2 & 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u \\ y = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} x \end{cases}$$

progettare in Simulink un controllore tale che gli autovalori del sistema a ciclo chiuso siano $\lambda_{1,2} = -3 \mp j25$.

2. Dato il sistema dinamico

$$\begin{aligned} \dot{x}_1 &= x_1^2 + x_2 \\ \dot{x}_2 &= -x_1^2 + x_2 + u \end{aligned}$$

linearizzando opportunamente il sistema, si progetti un controllo in retroazione di stato in modo che l'origine del sistema retroazionato sia asintoticamente stabile.

Modulo 3: Osservatori di Stato

Progetto di un osservatore

Progettare un osservatore per il sistema

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -2x_2 + u \\ y &= x_1 \end{aligned}$$

Svolgimento.

Il progetto di un controllore attraverso retroazione dello stato presuppone la conoscenza di tutto lo stato del sistema. Spesso nella pratica questa ipotesi non è verificata; in questi casi si rende necessario il progetto di un osservatore che a partire dalla conoscenza degli ingressi e delle uscite fornisca una stima dello stato. In termini di spazio di stato l'osservatore ha una forma del tipo:

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x}) - Du$$

Che può essere riscritta come:

$$\dot{\hat{x}} = (A - LC)\hat{x} + (B - LD)u + Ly$$

Verifichiamo l'osservabilità del sistema. Ciò può essere fatto tramite i comandi **obsv** e **rank**:

```
>> A=[0 1; 0 -2]

A =

     0     1
     0    -2

>> B=[0 1]'

B =

     0
     1

>> C=[1 0]

C =

     1     0

>> D=0

D =

     0

>> rank(obsv(A,C))

ans =

     2
```

Essendo il rango della matrice di osservabilità uguale all'ordine del sistema, si evince che il sistema è osservabile.

Calcoliamo gli autovalori:

```
>> eig(A)

ans =

     0
    -2
```

Vogliamo progettare un osservatore che consenta la stima della seconda variabile di stato (la prima coincide con l'uscita) e che abbia poli in $-10 \pm j$.

Effettuiamo prima il pole-placement:

```
>> L=place(A',C',[-10+j -10-j])'

L =

   18.0000
   65.0000
```

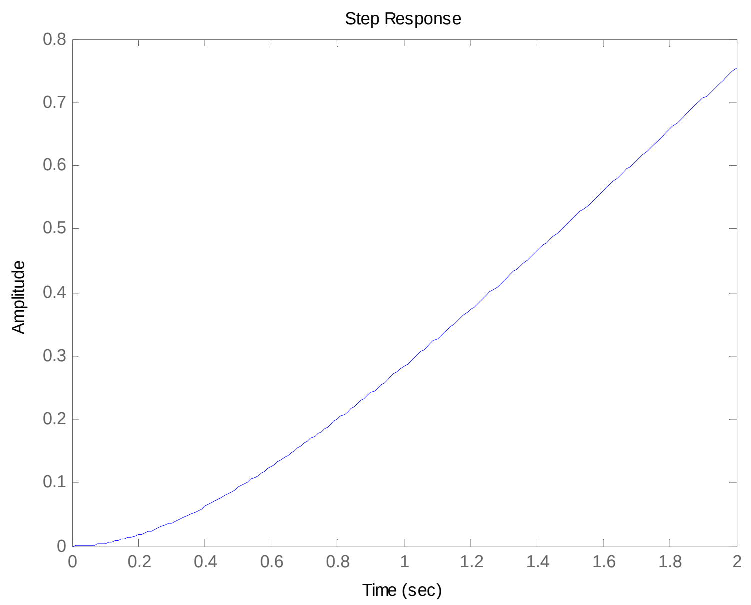
Con le istruzioni seguenti invece si assegnano le matrici dell'osservatore che ha come uscite la stima dello stato del sistema:

```
>> Ao=A-L*C

Ao =

  -18.0000    1.0000
  -65.0000   -2.0000

>> Bo=[B-L*D L]

Bo =

         0   18.0000
    1.0000   65.0000

>> Co=eye(2)

Co =

     1     0
     0     1

>> Do=zeros(2,2)

Do =

     0     0
     0     0
```

Per verificare le prestazioni dell'osservatore progettato consideriamo l'evoluzione dello stato del sistema e la sua stima in presenza di un ingresso a gradino di ampiezza unitaria.

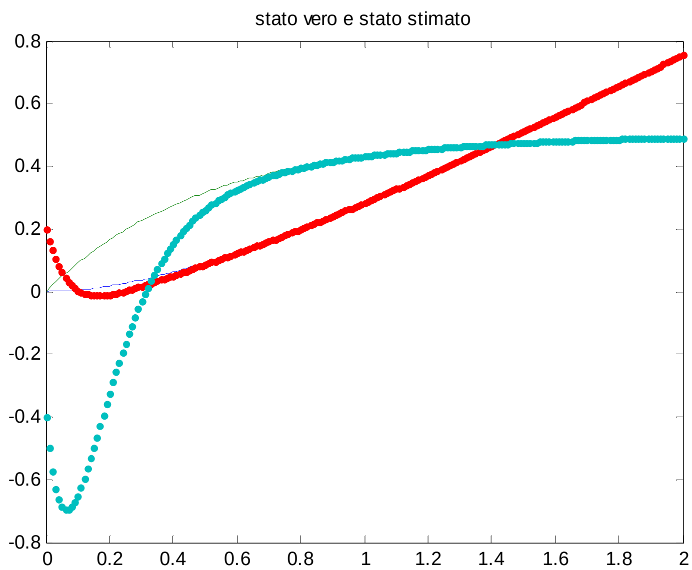
Per quanto riguarda il sistema:

```
>> step(A,B,C,D,1,0:0.01:2);
```



Assumendo come stato iniziale dell'osservatore il vettore (0.2, -0.4), lo stato dell'osservatore ed una rappresentazione dello stato vero e di quello stimato si possono ottenere con le seguenti istruzioni:

```
>> xo_obs=[0.2 -0.4];
>> [yo, xo]=lsim(Ao,Bo,Co,Do,[ones(size(t))' y],t,xo_obs);
>> plot(t,x,t,xo,'.')
>> title('stato vero e stato stimato')
```



Dove le variabili stimate sono rappresentate dai tratti punteggiati.

Esercizi

1. Valutare se il seguente sistema dinamico tempo-discreto è osservabile o meno, spiegandone il perchè:

$$\begin{cases} x_{k+1} = \begin{pmatrix} 0 & 1 \\ -0.2 & \sqrt{2} \end{pmatrix} x_k + \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix} u_k \\ y_k = \begin{pmatrix} 2 & 1 \\ 0 & -1 \end{pmatrix} x_k \end{cases}$$

2. Costruire uno schema Simulink di un osservatore che stimi lo stato del sistema:

$$\begin{cases} \dot{x} = \begin{pmatrix} -1 & 0 \\ 2 & -3 \end{pmatrix} x + \begin{pmatrix} 1 \\ 0 \end{pmatrix} u \\ y = (0 \quad 1) x \end{cases}$$

garantendo che l'errore vada a regime in meno di 10 secondi.

3. Con riferimento all'esercizio precedente, ricavare la risposta indiciale del sistema a ciclo chiuso.

4. Si consideri il sitema di equazioni

$$\begin{cases} \dot{x}(t) = \begin{pmatrix} -7 & 1 \\ -12 & 0 \end{pmatrix} x(t) + \begin{pmatrix} 1 \\ -1 \end{pmatrix} u(t) \\ y = (1 \quad 0) \end{cases}$$

   a) Si fornisca il grafico della risposta indiciale a ciclo aperto;
   b) Si valuti la controllabilità e l'osservabilità del sistema;
   c) Si progetti un controllore in retroazione di stato che induca nel sistema un comportamento oscillatorio caratterizzato da un tempo di assestamento pari a 25 secondi e periodo di oscillazione pari a ¼.
   d) Si progetti un'azione in feed-forward in modo che il guadagno del sistema ciclo chiuso sia unitario.
   e) Si produca uno schema a blocchi del sistema di controllo sviluppato al punto precedente.

5. Sviluppare opportunamente uno schema per la risoluzione numerica del problema al punto precedente con la piattaforma Matlab/Simulink. Si descrivano con cura tutti i dettagli informatici necessari a rendere il codice operativo.

6. Dato il sistema

$$\dot{y} - 0.1y = 2u$$

Realizzare in Simulink un controllore in retroazione di uscita che garantisca errore nullo a regime ed oscillazioni nel transitorio con periodo pari a circa 3s che decadano dopo circa 10s.

7. Si consideri il sistema di equazioni

$$\begin{cases} \dot{x}(t) = \begin{pmatrix} -7 & 1 \\ -12 & 0 \end{pmatrix} x(t) + \begin{pmatrix} 1 \\ -1 \end{pmatrix} u(t) \\ y = \begin{pmatrix} 1 & 0 \end{pmatrix} \end{cases}$$

Sviluppare opportunamente uno schema Simulink di un controllore in retroazione di uscita che induca nel sistema un comportamento oscillatorio caratterizzato da un tempo di assestamento pari a 25 secondi e periodo di oscillazione pari a ¼.

8. Si considerino due serbatoi in cascata, detti rispettivamente serbatoio 1 e 2. Nelle ipotesi semplificative in cui:
   - la superficie di ogni serbatoio sia unitaria;
   - la portata in uscita sia proporzionale al livello del liquido in ogni serbatoio secondo un coefficiente σ uguale per entrambi i serbatoi;
   - una portata u sia posta in ingresso al solo serbatoio 1;
   - sia misurabile solo il livello nel serbatoio 2;
     a) si fornisca un modello dinamico del sistema;
     b) si studi la stabilità del sistema al variare di σ

   Scelto un valore opportuno σ

   a) si dia un grafico qualitativo della risposta del sistema ad una portata in ingresso unitaria, considerando che inizialmente i serbatoi siano vuoti;
   b) si valuti la controllabilità e l'osservabilità del sistema;
   c) si progetti un controllo in retroazione di stato che posizioni gli autovalori del sistema a ciclo chiuso in $-2\sigma(1\pm j)$;
   d) si progetti un osservatore di stato che consenta l'implementazione dell'osservatore progettato al passo precedente.

Modulo 4: PID in Simulink

Il termometro

Progettare un controllore per il termometro descritto da:

$$\dot{y} + 0.2y = u$$

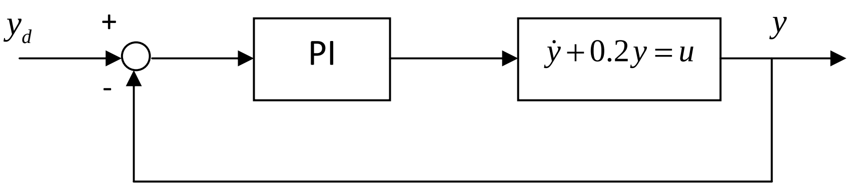
che garantisca un errore a regime nullo ed un tempo di assestamento inferiore ad 10s.

<u>Svolgimento</u>

Per avere errore nullo a regime usiamo un'azione integrale; per soddisfare la seconda specifica usiamo invece un'azione proporzionale:

$$u = K_P e + K_I \int_0^t e(\tau)d\tau$$

Nel sistema a ciclo chiuso risulta:



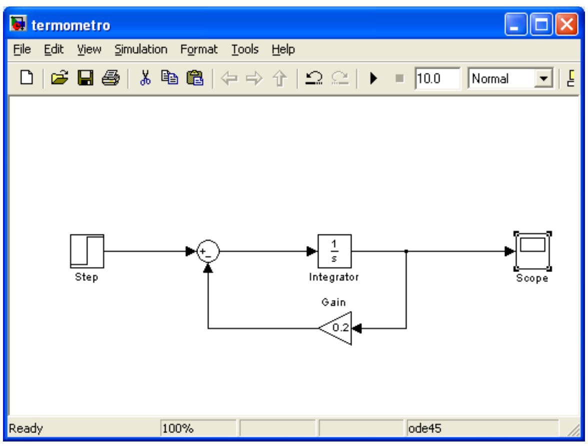
$$\dot{y} + 0.2y = K_P(y_d - y) + K_I \int_0^t e(y_d - y)d\tau \Rightarrow \ddot{y} + 0.2\dot{y} = K_P(y_d - y) + K_I(y_d - y)$$

Se y_d è costante la sua derivata sarà nulla e quindi ne deriva:

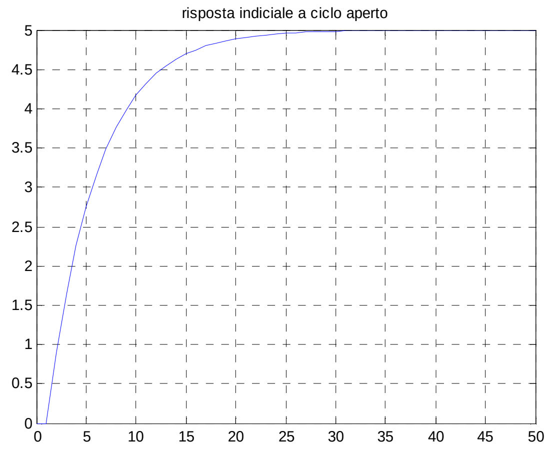
$$\ddot{y} + (0.2 + K_P)\dot{y} + K_I y = K_I y_d$$

Vediamone la realizzazione in Matlab/Simulink.

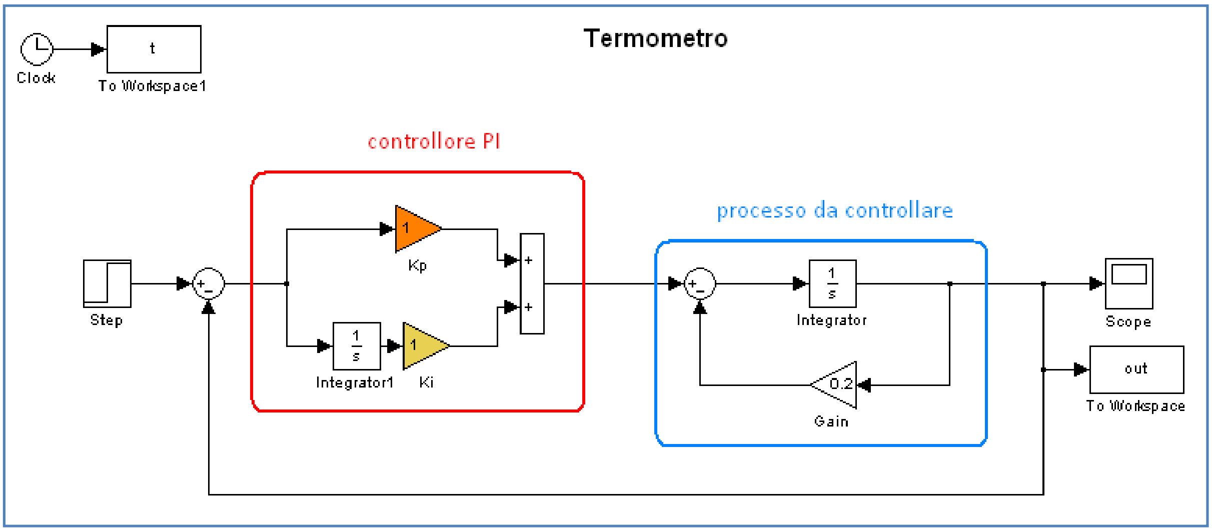
Il sistema in esame è così costituito (in ingresso abbiamo messo un gradino di ampiezza unitaria):



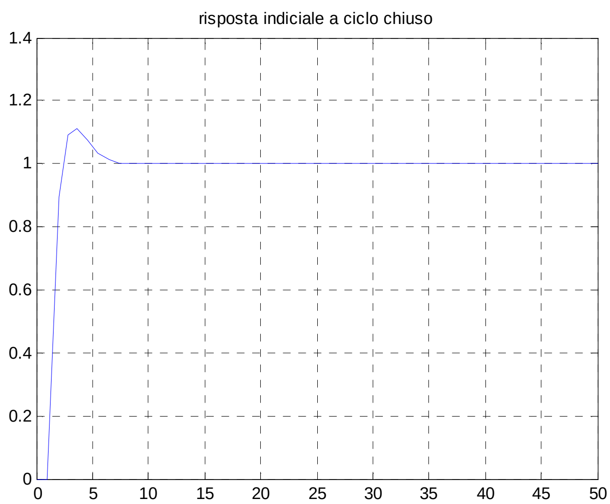
Diagrammiamo la risposta indiciale a ciclo aperto:

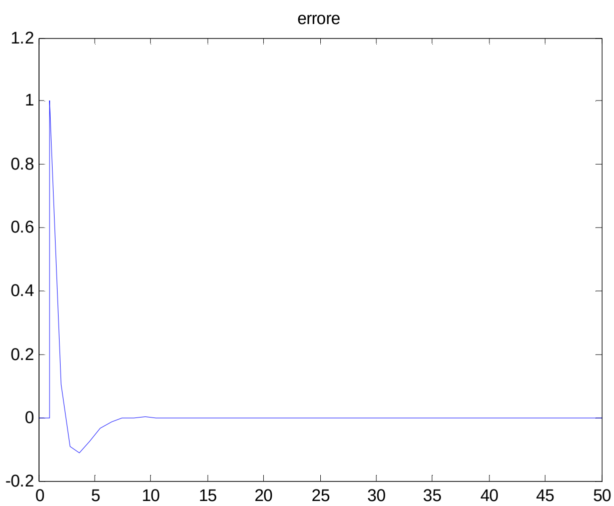


Andiamo a realizzarlo schema Simulink del controllore PI precedentemente calcolato in modo analitico:



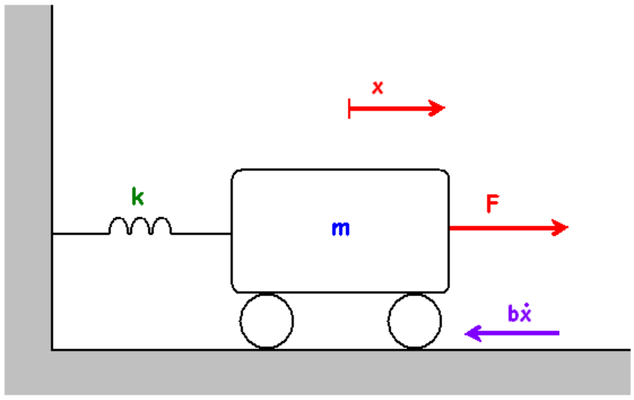
Scegliendo opportunamente i coefficienti proporzionale e integrale (per esempio entrambi pari a 1), riusciamo a soddisfare le specifiche:





Sistema massa-molla

Consideriamo un sistema composto da una massa su cui agiscono: una forza F, una molla con costante elastica k e l'attrito con coefficiente b.



$$m\ddot{x}(t)+b\dot{x}(t)+kx(t)=F(t)$$

Con:

m = 1kg
b = 10 N.s/m
k = 20 N/m
F(s) = 1

Mostrare come ciascun coefficente Kp, Ki e Kd influisce per ottenere:

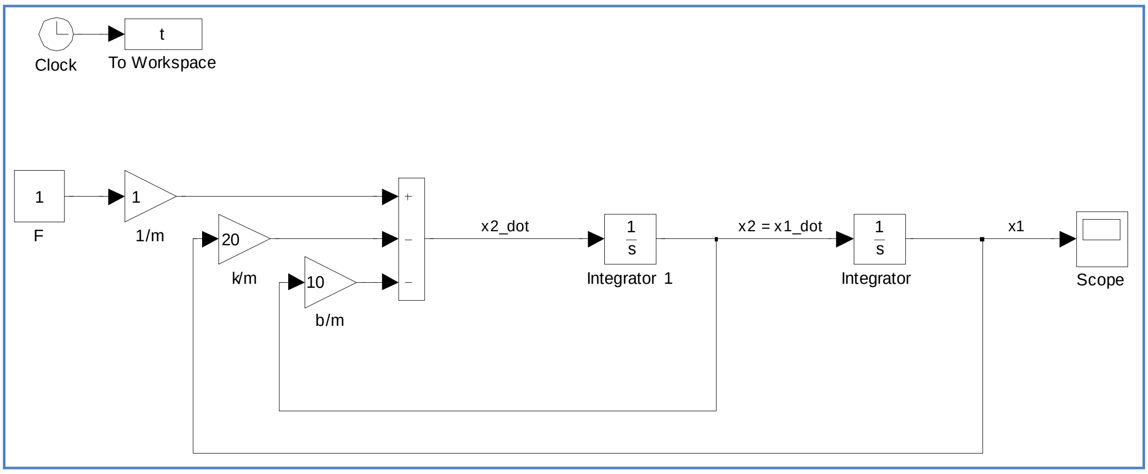
- Un rapido tempo di salita
- Minima Sovraelongazione
- Errore nullo in Regime Permanente

__Svolgimento__.

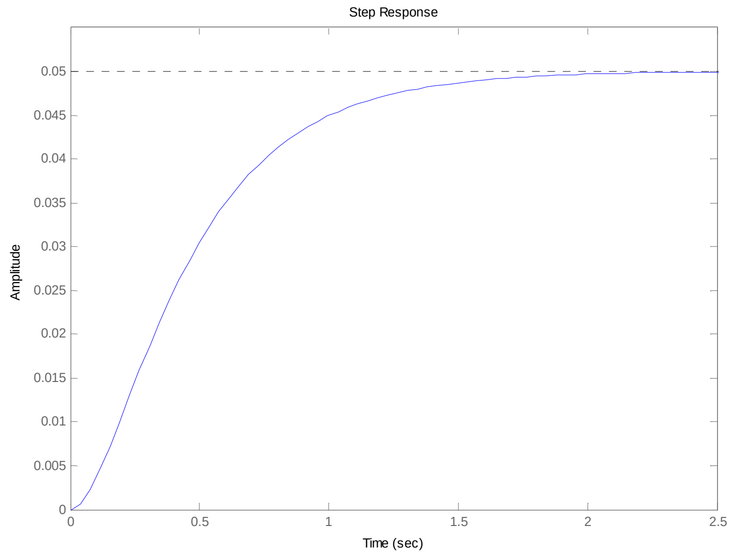
Posto $x_1 = x$ e $x_2 = \dot{x}_1$ il sistema può essere così descritto:

$$\begin{aligned}
&\dot{x}_1 = x_2\\
&\dot{x}_2 = -\frac{k}{m}x_1 - \frac{b}{m}x_2 + \frac{1}{m}F\\
&y = x_1
\end{aligned}$$

In Simulink si presenta come segue:



- Cominciamo ad analizzare la risposta a C.A: andiamo a graficare la risposta indiciale del sistema.

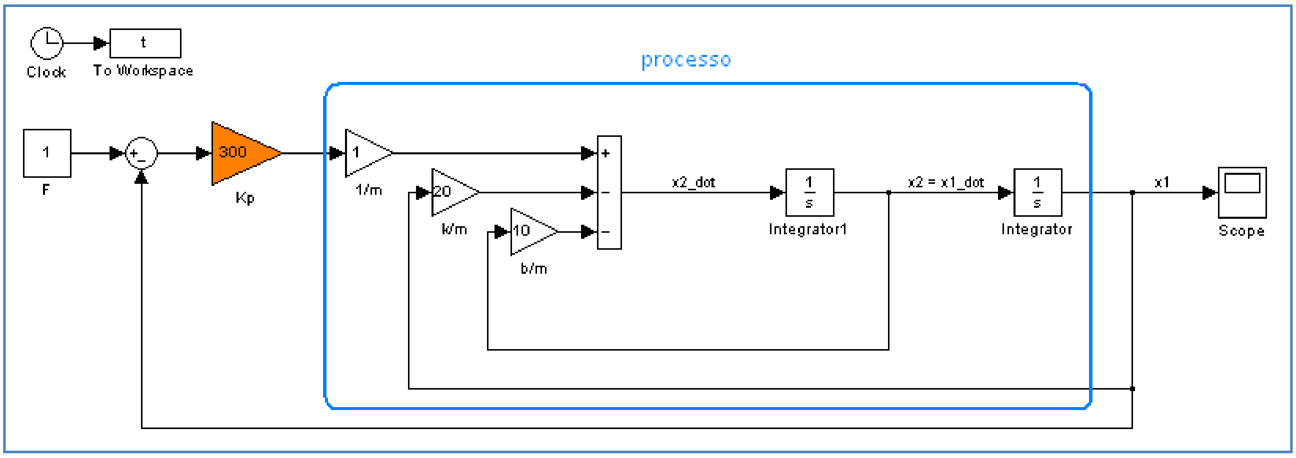


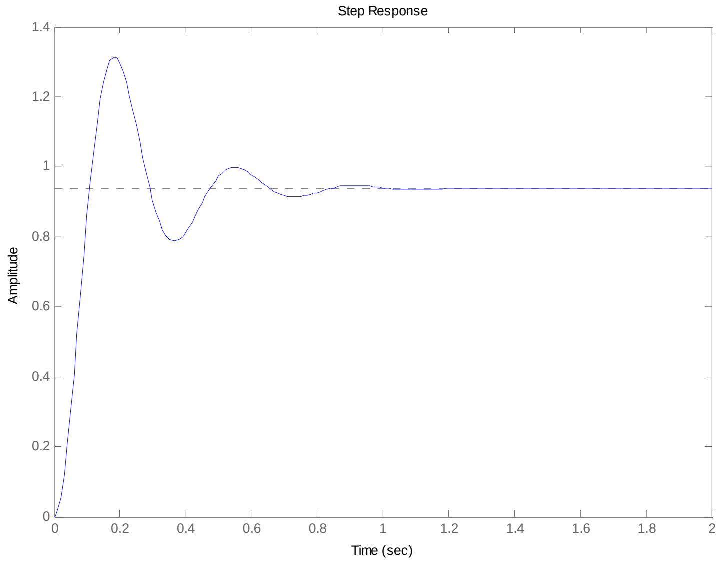
Si osserva un elevato errore a regime, inoltre il tempo di salita e' di circa un secondo, e il transitorio e' circa 1.5 secondi.

Costruiamo un controllore che riduca il tempo di salita e il transitorio, ed elimini l'errore a regime permanente.

- Controllore ad azione Proporzionale (P).

Dalla tabella precedentemente mostrata si può vedere che Kp riduce il tempo di salita, incrementa le sovraelongazioni e riduce l'errore in regime permanente. Poniamo Kp=300:

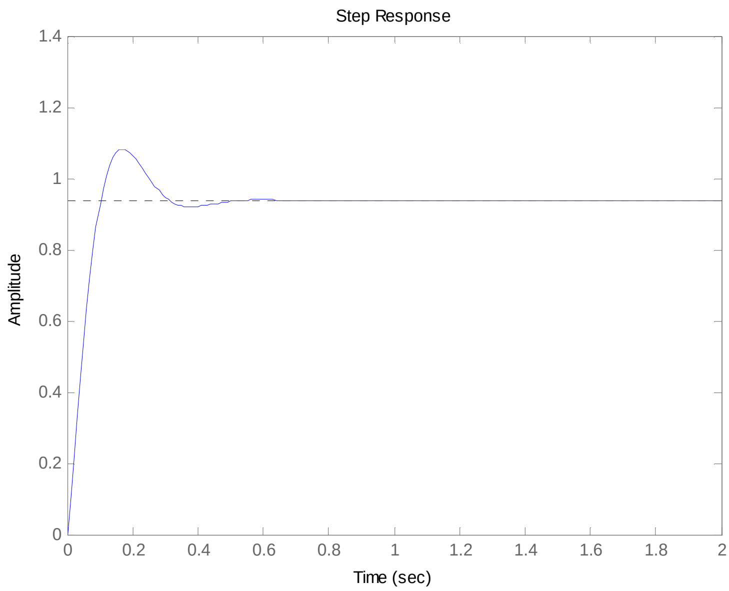




Il grafico mostra che il controllore Kp riduce sia il tempo di salita che l'errore a regime permanente, aumenta le oscillazioni e decresce leggermente il transitorio.

- Controllore ad azione Proporzionale e Derivativa (PD).

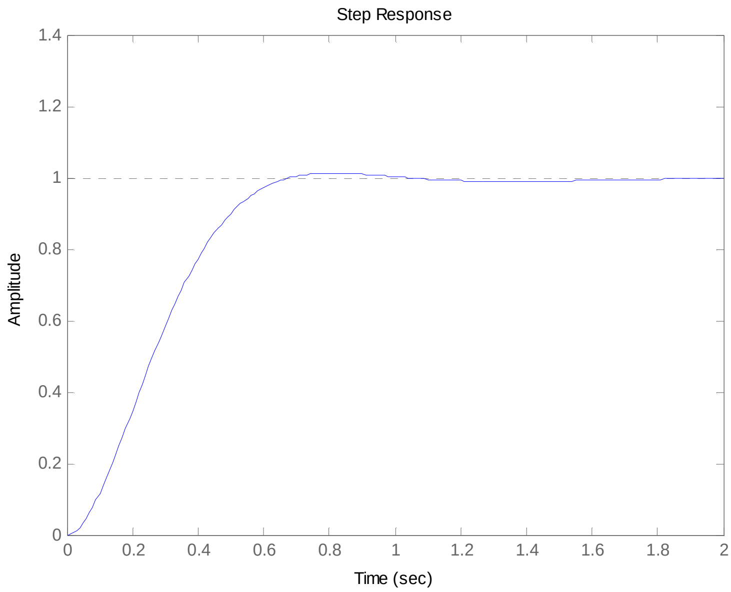
Dalla tabella vista si può vedere che il controllore Kd riduce le sovraelongazioni del transitorio. Imponendo Kp=300 e Kd=10 si ottiene il grafico seguente:



Il grafico mostra, come ci si attendeva, che l'azione derivativa diminuisce sia la sovraelongazione che il transitorio e influisce poco sul tempo di salita e sull'errore a regime permanente.

- Controllore ad azione Proporzionale ed Integrale (PI)

Ci si aspetta che il controllore Ki diminuisca il tempo di salita, incrementi sia le sovraelongazioni che il transitorio ed elimini l'errore a regime permanente. Riduciamo il Kp a 30 e consideriamo Ki=70. Si ottiene:

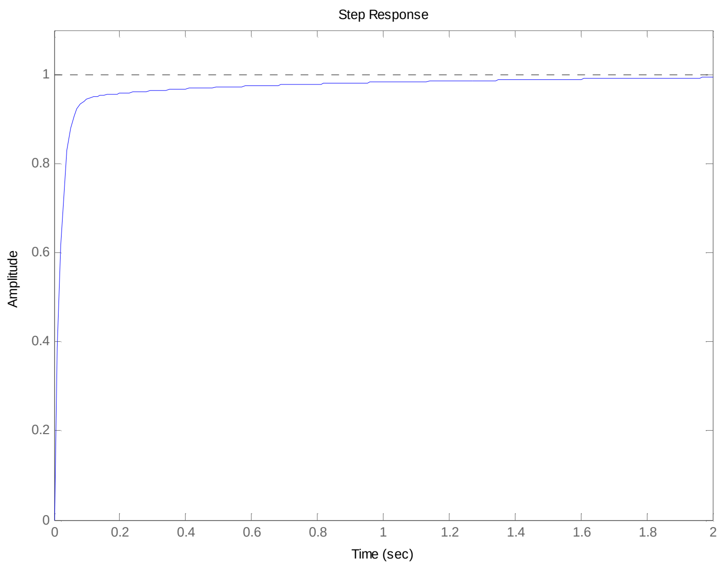


Si è ridotto il guadagno proporzionale Kp perché il controllore Ki stesso riduce il tempo di salita e incrementa le sovraelongazioni (anche il controllore Kp ha questi effetti). Si noti che l'aggiunta del controllore ad azione integrale ha eliminato l'errore a regime permanente.

- Controllore ad azione Proporzionale, Integrale e Derivativa (PID)

Dopo vari tentativi la risposta desiderata che è stata trovata con i seguenti valori dei guadagni:

Kp=350; Ki=300; Kd=50.



Abbiamo ottenuto un sistema che non ha oscillazioni, ha un basso tempo di salita ed ha un errore nullo in regime permanente.

Possiamo dire che un controllore proporzionale (Kp) avrà l'effetto di ridurre il tempo di salita (parametro che caratterizza la prontezza della risposta) e ridurre, ma non eliminare, l'errore a regime permanente. Un'azione Integratrice (regolatore I con cost. Ki) ha l'effetto di eliminare l'errore in Regime Permanente ma peggiora la risposta transitoria (noie con la stabilità). Un controllore derivativo (Kd) ha l'effetto di aumentare la stabilità del sistema, migliorando la risposta transitoria. Questo tipo di di controllore non è fisicamente realizzabile e viene abbinato con un controllore proporzionale ed integratore. Gli effetti di uni controllore in un sistema a ciclo chiuso sono riassunti nella seguente tabella:

CONTROLLORE	**TEMPO DI SALITA**	**SOVRAELONGAZIONI**	**TRANSITORI**	**ERRORE A REGIME**
Kp	Diminuisce	Aumentano	Non Influisce	Diminuisce
Ki	Diminuisce	Aumentano	Aumenta	Eliminato
Kd	Non Influisce	Diminuiscono	Diminuisce	Non Influisce

N.B: queste relazioni non sono accurate; infatti i tre controllori sono dipendenti l'uno dall'altro. Lo schema è indicativo nella determinazione dei valori di Kp, Ki e Kd.

Esercizi

1. Fornire il diagramma a blocchi e l'equazione a ciclo chiuso di un sistema di controllo proporzionale i cui componenti siano descritti come segue:

   Sistema: $\ddot{x}+6\dot{x}+9x=u$;

   Attuatori: $4u=w$;

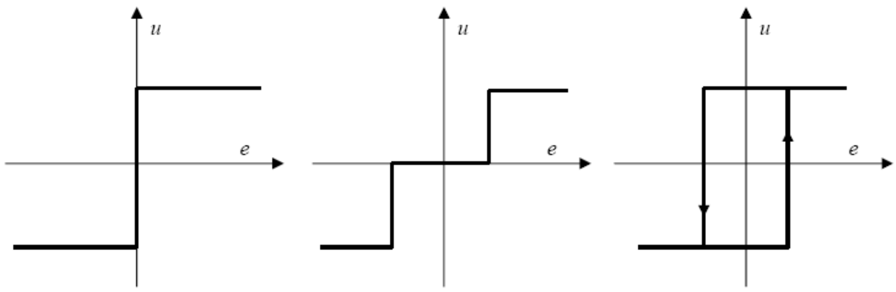
   Sensori: ideali.

2. Selezionare opportunamente il guadagno del controllore proporzionale per il sistema di cui al punto precedente in modo da ottenere una risposta oscillante con periodo pari a 2π .

3. Si consideri un sistema di due cisterne in cascata e si dicano $l_1(t)$ ed $l_2(t)$ i livelli di liquido nelle due cisterne nell'ipotesi in cui la fuoriuscita di liquido dalla prima cisterna sia pari a $0.6l_1(t)$ e quella dalla seconda cisterna a $0.8l_2(t)$.
   a) Ricavare il modello del sistema
   b) Progettare un opportuno sistema di controllo (decidendo a quale serbatoio aggiungere l'ingresso di controllo) in modo che a regime i due serbatoi siano riempiti ai livelli $l_1=10$ e $l_2=6$ e che il tempo di riempimento dei serbatoi sia inferiore a 5 s. (Illustrare con estrema cura la progettazione del controllore).
   c) Fornire lo schema a blocchi del sistema di controllo progettato al punto precedente.
   d) Diagrammare la risposta del sistema a ciclo chiuso.

Modulo 5: Controllo a relé (ON-OFF)

Nel controllo a relé (anche detto ON-OFF) il regolatore fa assumere alla variabile di ingresso dell'attuatore due soli valori: a seconda se l'uscita del sistema oltrepassa in alto o in basso un margine d'errore (detto gap). Il controllo On/off, a fronte della sua immediatezza e semplicità di implementazione, implica che viene sempre utilizzato il controllo al massimo delle sue possibilità.



On/Off ideale — On/Off con zona morta (necessita di 3 posizioni) — On/Off con isteresi e zona morta

Lo scaldabagno

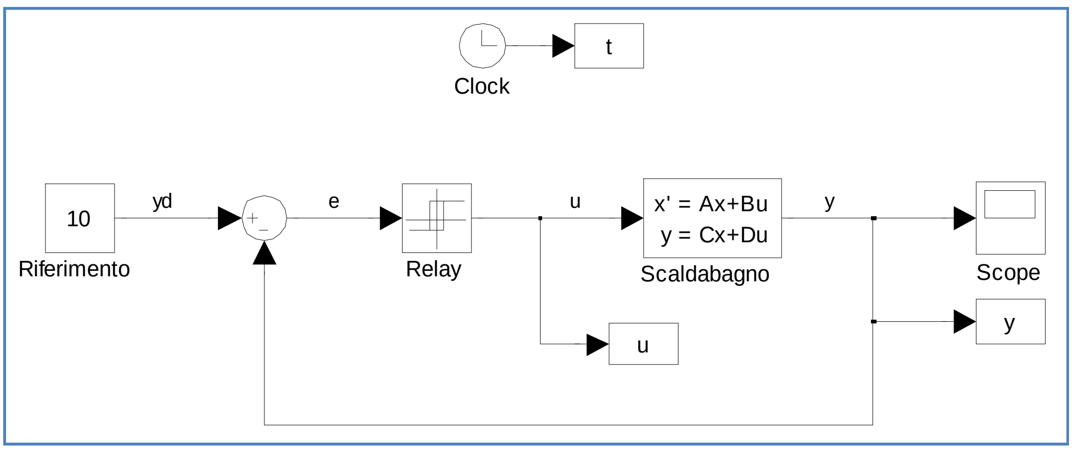
Progettare un sistema di controllo a relé in modo tale che il sistema

$$\dot{y} + 0.2y = u$$

esibisca oscillazioni di ampiezza inferiore al 5% intorno al valore desiderato $y_d = 10$.

<u>**Svolgimento**</u>.

Possiamo utilizzare un relé con isteresi, troviamo tale blocco nella libreria "Discontinuities". Lo schema di controllo assume la forma seguente:



Nel nostro caso se $u = U$ risulta:

$$\cancel{\dot{y}_\infty} + 0.2y_\infty = U \Rightarrow y_\infty = 5 \cdot U$$

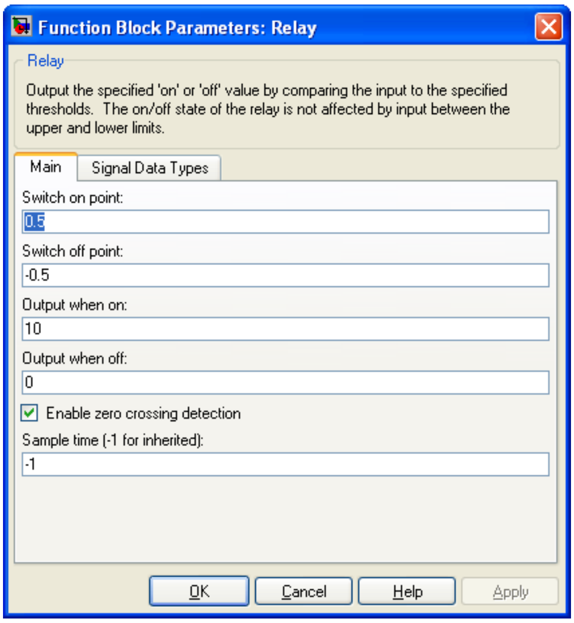
Perciò scegliamo U in modo tale che :

$$y_{\infty} = 5 \cdot U > 10$$

Scegliendo $U = 10 \Rightarrow y_{\infty} = 50$ e inoltre $t_a = 5\tau \Rightarrow t_a \simeq 25s$.

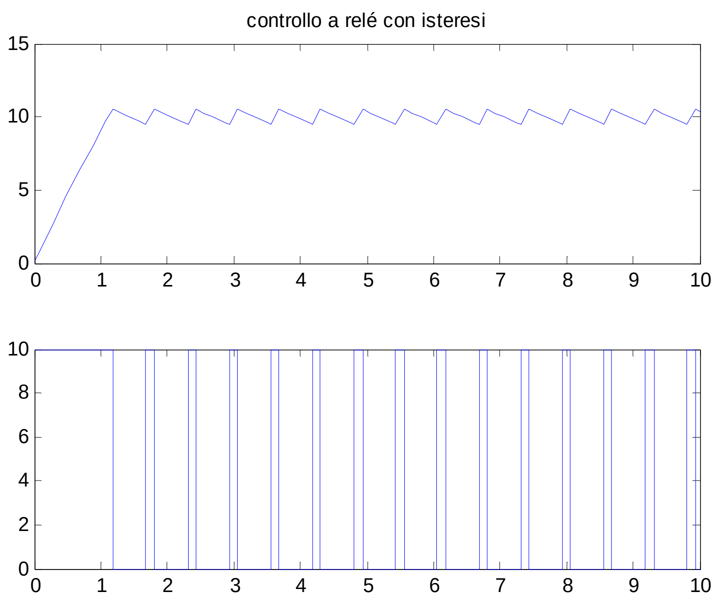
Inoltre poiché $y = 10 \pm 0.5 \Rightarrow \varepsilon = 0.5$

Dunque cliccando due volte sul blocco “Relay” impostiamo i parametri necessari:



Avviamo la simulazione e diagrammiamo i risultati:

```
>> figure
>> subplot(2,1,1);
>> plot(t,y)
>> title('controllo a relé con isteresi')
>> subplot(2,1,2);
>> plot(t,u)
```



Esercizi

1. Realizzare in Matlab/Simulink lo schema di controllo a relé per un frigorifero, con $y_d = -18°C$.