Controllo digitale

Introduzione alla simulazione di sistemi a tempo discreto in ambiente Matlab-Simulink

Alessandro Pisano
apisano@unica.it

Problema 1
Simulare il sistema descritto dalla equazione alle differenze:

$$y(k+1) - 1.2y(k) = u(k)$$

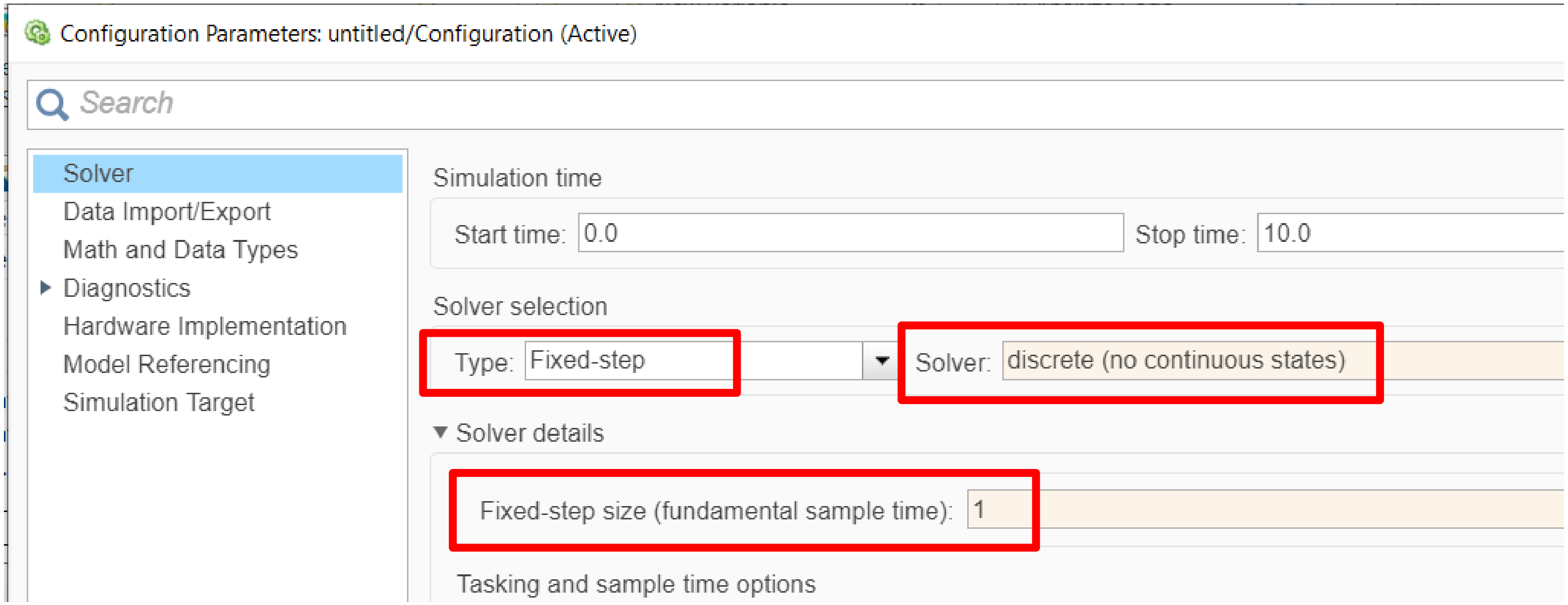
$$y(0) = 10$$

$$u(k) = 5\delta_{-1}(k)$$

(esempio discusso a lezione)

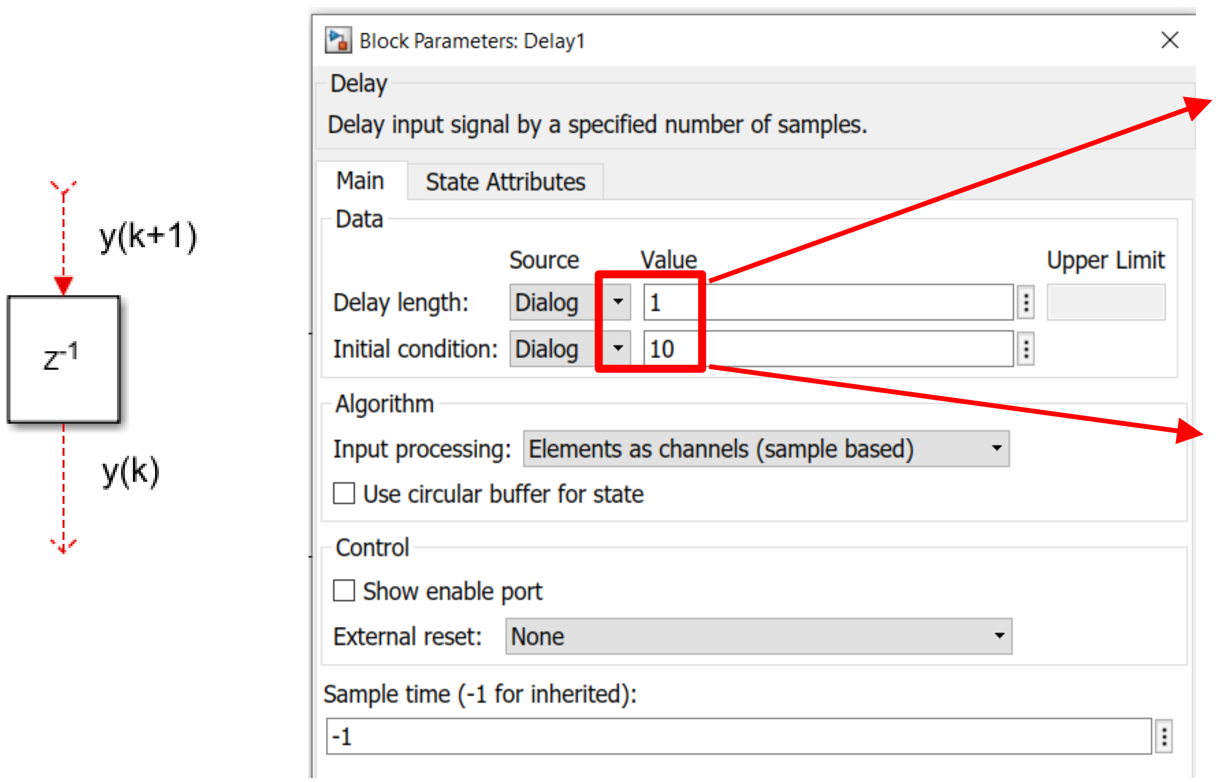
Guida passo-passo per la realizzazione del modello

Aprire un nuovo modello, e configurare i parametri del Solver (Model Settings) come segue



Il blocco fondamentale per la modellazione in Simulink di una equazione alle differenze è il blocco «delay» (libreria «Discrete»)

E' un blocco che applica un ritardo di m passi di campionamento (m è un parametro di configurazione del blocco).



Ritardo di 1 passo ($m = 1$)

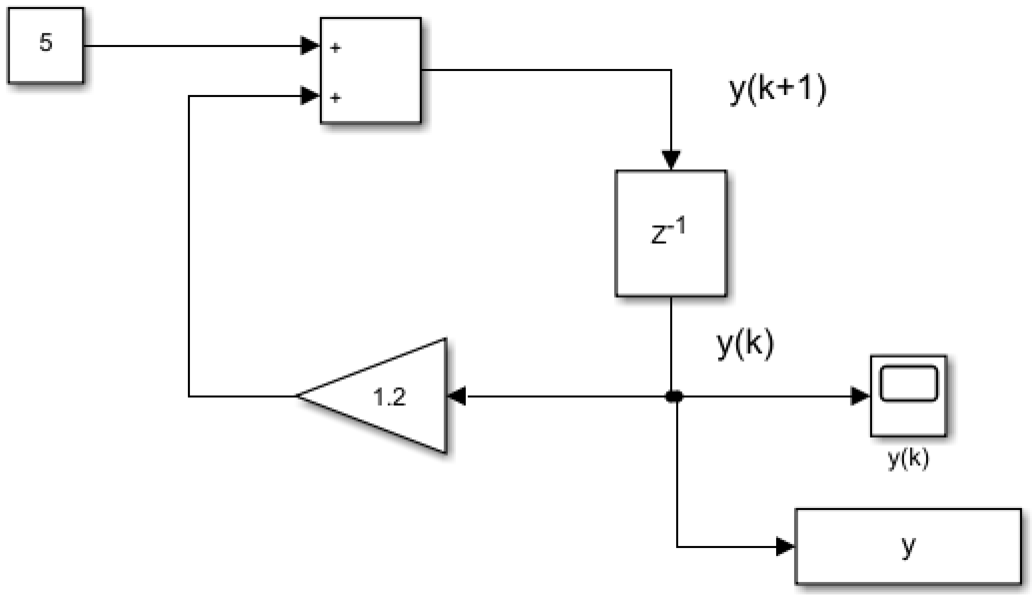
Condizione iniziale $y(0) = 10$ per il segnale in uscita dal blocco ($y(k)$)

L’idea è quella di costruire l’ingresso al blocco delay (la sequenza $y(k+1)$) combinando fra loro $y(k)$ ed $u(k)$ in modo da realizzare l’equazione alle differenze riscritta in forma esplicita:

$$y(k+1) = 1.2y(k) + u(k)$$

equazione alle differenze riscritta in forma esplicita:

Si realizzi il seguente schema:

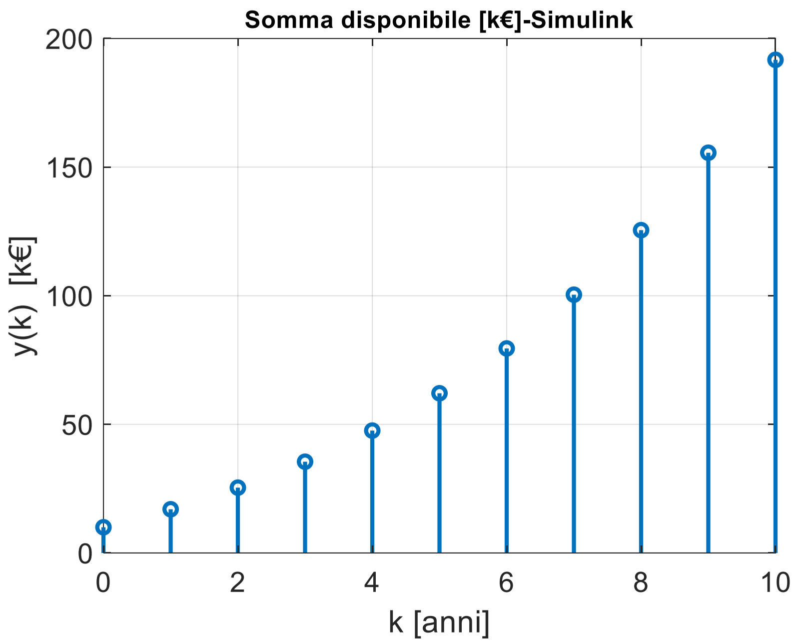


Ead01.slx

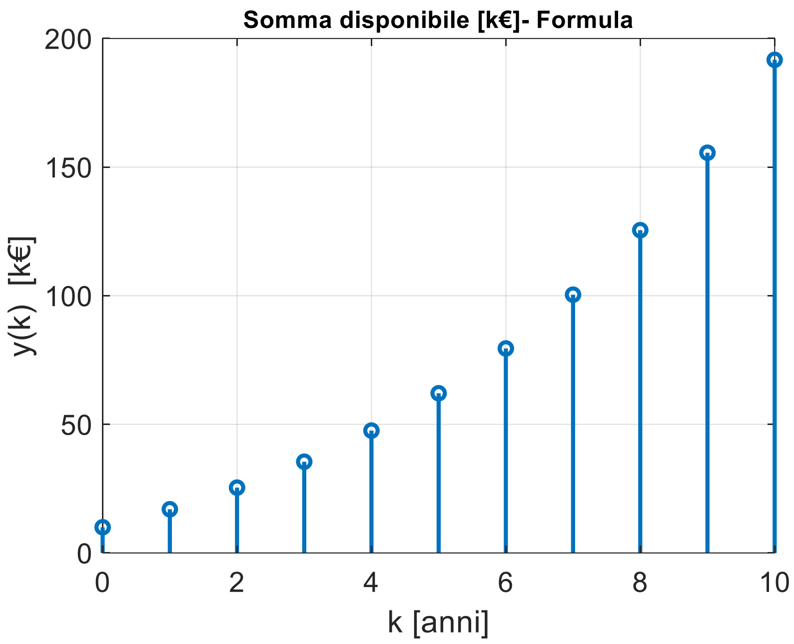
Confrontiamo attraverso un test esteso ai primi 10 campioni la risposta ottenuta mediante Simulink con la sua espressione analitica ricavata a lezione

$$y(k) = (1.2)^k y(0) + 5\frac{1-(1.2)^k}{1-1.2} = 35(1.2)^k - 25$$

```
figure(1)
stem(y.Time,y.Data,'LineWidth',2),grid
xlabel('k [anni]')
ylabel('y(k)  [k€]')
set(gca,'FontSize',14)
title('Somma disponibile [k€] - Simulink','FontSize',12)

k=0:10;
y_analytic=35*(1.2).^k-25;
figure(2)
stem(k,y_analytic,'LineWidth',2),grid
xlabel('k [anni]')
ylabel('y(k)  [k€]')
set(gca,'FontSize',14)
title('Somma disponibile - Matlab','FontSize',12)
```



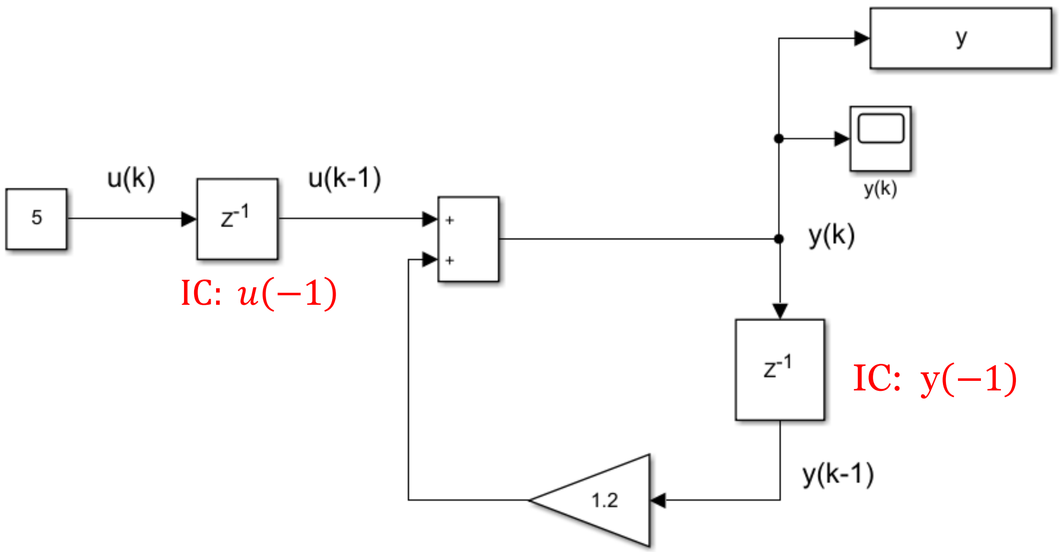
Somma disponibile [k€]-Simulink
200
150
100
50
0
y(k) [k€]
0
2
4
6
8
10
k [anni]



Somma disponibile [k€]- Formula
200
150
100
50
0
y(k) [k€]
0
2
4
6
8
10
k [anni]

Il modello può essere anche realizzato sfruttando l’equazione «all’indietro»

$$y(k) = 1.2y(k-1) + u(k-1)$$



Ead02.slx

In questa versione del modello, il blocco delay genera in uscita la sequenza $y(k-1)$, non più $y(k)$ come nel modello precedente. La condizione iniziale deve pertanto essere modificata rispetto a prima. All’istante iniziale $k=0$ la condizione iniziale deve essere $y(-1)$. Come determinarla ?

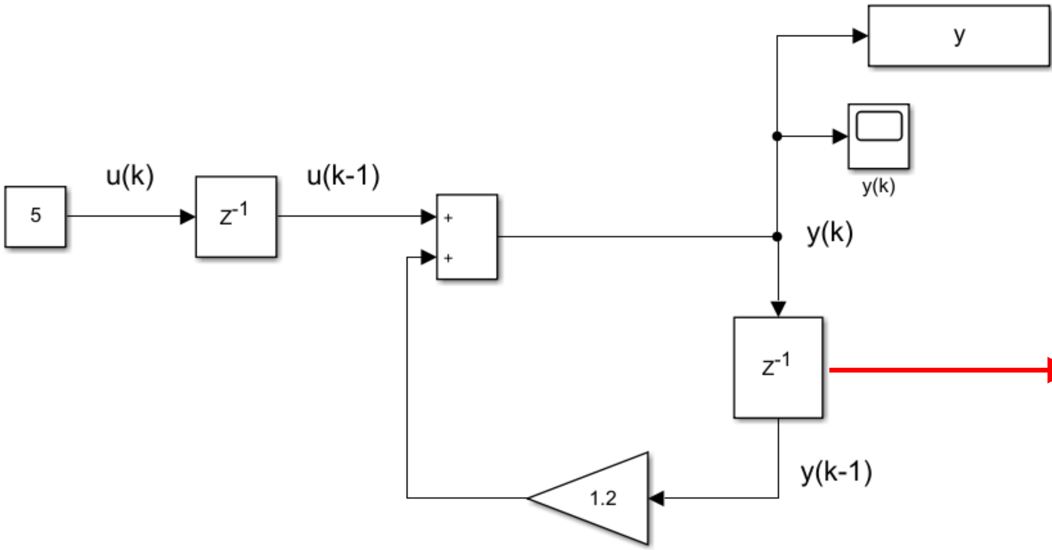
Il valore di $y(-1)$ lo ricaviamo ovviamente dalla equazione alle differenze:

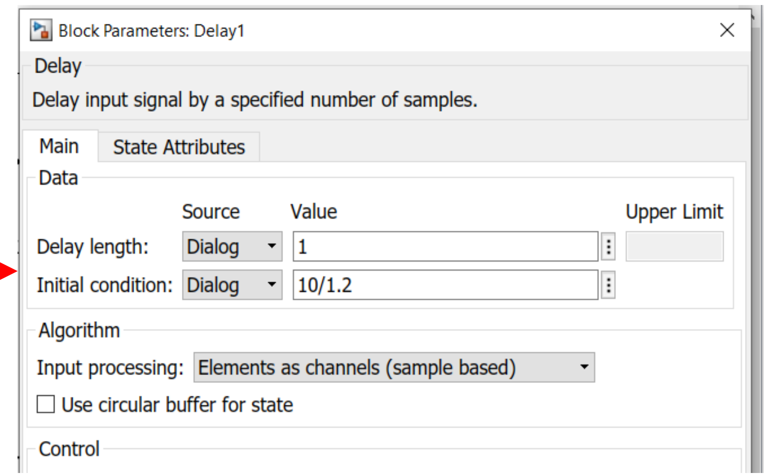
$$y(k) = 1.2y(k-1) + u(k-1)$$

La valutiamo per $k = 0$ $\qquad y(0) = 1.2y(-1) + \overbrace{u(-1)}^{=0} = 1.2y(-1)$

$$\boxed{y(-1) = \frac{y(0)}{1.2}}$$

Impostare correttamente la condizione iniziale



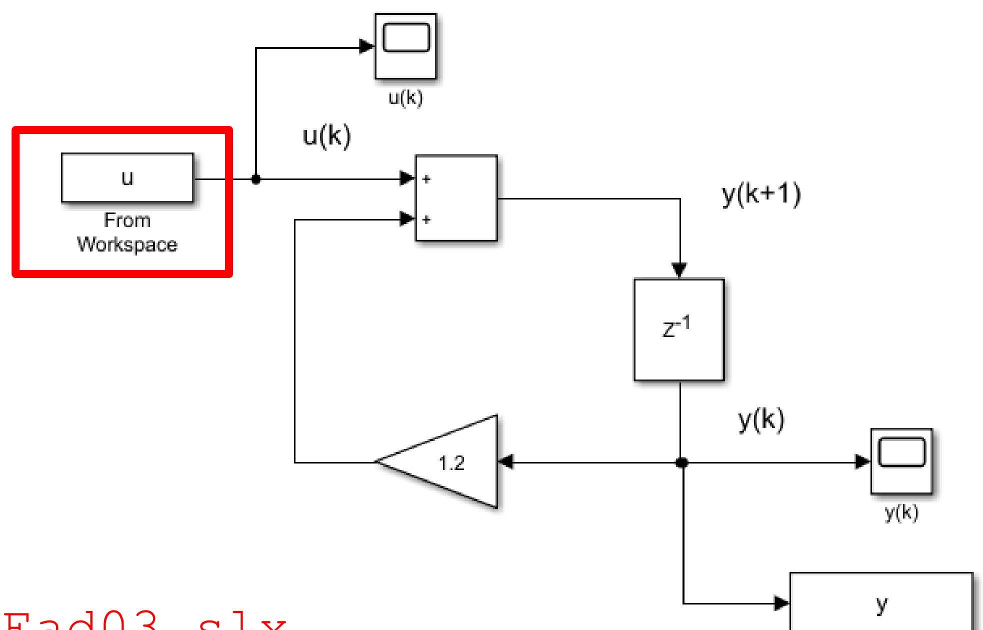


Ora vediamo come applicare in ingresso una sequenza $u(k)$ differente

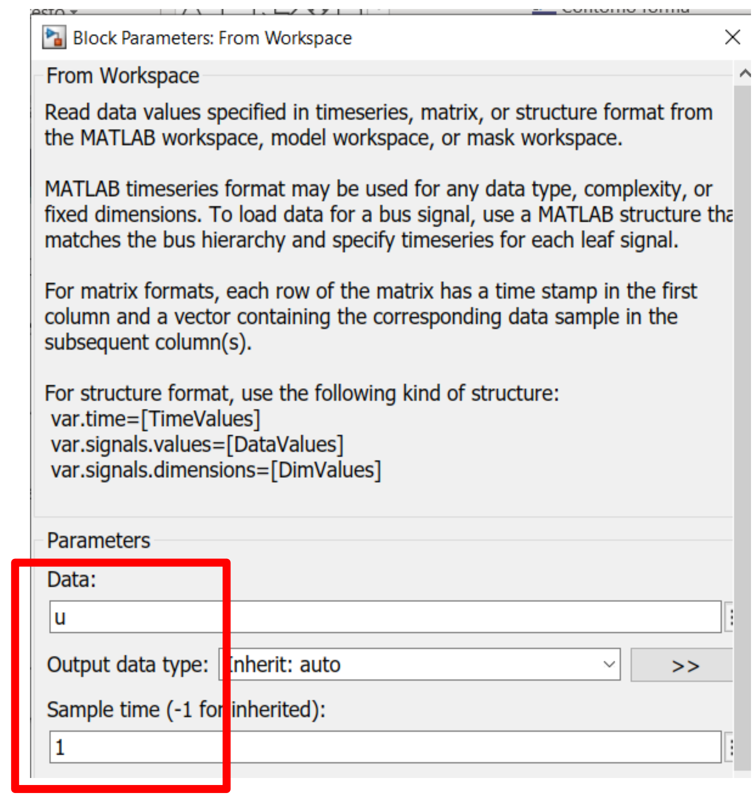
In luogo della sequenza costante $u(k) = 5\delta_{-1}(k)$ ora applichiamo una sequenza di ingresso variegata:

$$u(k) = [-5 \;\; 0 \;\; 5 \;\;\; 0 \; -20 \;\; -10]$$

Partiamo dalla versione del modello del file `Ead01.slx`. Il primo passo è sostituire il blocco `constant` che generava il segnale di ingresso costante pari a 5 con un blocco «`From Workspace`» (Libreria «`Sources`»)



Ead03.slx



Script da eseguire prima di avviare la simulazione

```
clc

k=[0:5];
inp=[-5 0 5 0 -20 -10];
u=[k' inp'];
```

Si deve creare nel workspace una matrice che si chiama `u` (lo stesso nome inserito nel campo `Data` della finestra di parametrizzazione del blocco `From workspace`.
Tale matrice deve avere nella prima colonna gli istanti discreti, ed affianco i corrispondenti valori di `u(k)`

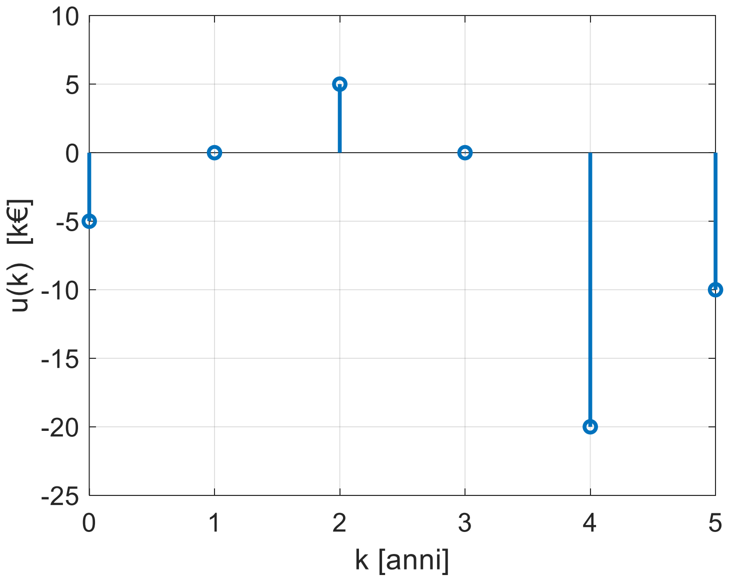
Nel caso in esame risulta:

$$u = \begin{bmatrix} 0 & -5 \\ 1 & 0 \\ 2 & 5 \\ 3 & 0 \\ 4 & -20 \\ 5 & -10 \end{bmatrix}$$

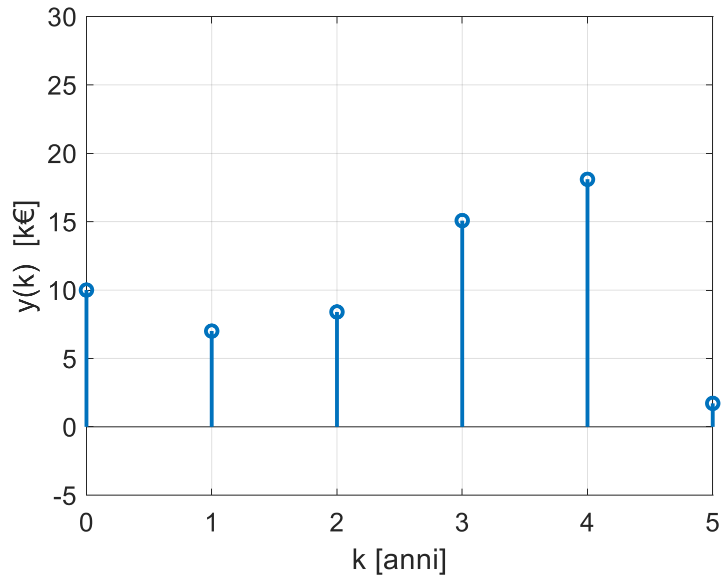
```
figure(1)
stem(k,inp,'LineWidth',2),
grid
xlabel('k [anni]')
ylabel('u(k)  [k€]')
set(gca,'FontSize',14)
axis([0 5 -25 10])
```

```
figure(2)
stem(y.Time,y.Data,'LineWidth',2),
grid
xlabel('k [anni]')
ylabel('y(k)  [k€]')
set(gca,'FontSize',14)
axis([0 5 -5 30])
```

$u(k)$



$y(k)$



Problema 2
Simulare il sistema descritto dalla equazione alle differenze:

$$y(k+2) + 0.5y(k+1) - 0.24\,y(k) = 2u(k+1) + u(k)$$

$$y(0) = 2 \qquad y(1) = 4 \qquad u(k) = 3\delta_{-1}(k)$$

Determinare l'espressione analitica della evoluzione libera

Realizzare il modello Simulink del sistema, e verificare che l'evoluzione libera ottenuta mediante simulazione coincida con l'espressione analitica calcolata

Visualizzare mediante Simulink l'evoluzione della sequenza di uscita in presenza dell'ingresso $u(k) = 3\delta_{-1}(k)$.

L'evoluzione libera è soluzione della equazione

$$y(k+2) + 0.5y(k+1) - 0.24\, y(k) = 0 \qquad y(0) = 2 \qquad y(1) = 4$$

Determiniamo i modi. Il polinomio caratteristico è:

$$\mathrm{P}(z) = z^2 + 0.5z - 0.24$$

Calcoliamone le radici mediante Matlab

```
>> roots([1 0.5 -0.24])

ans =

   -0.8000
    0.3000
```

Le radici sono entrambe reali e interne al disco unitario. Pertanto, tutti i modi sono stabili e la risposta libera tenderà a zero per $k \to \tau\infty$

Poiché una delle radici è negativa, ci attendiamo un transitorio con oscillazioni (ringing)

Modi: $(-0.8)^k$ $\quad (0.3)^k$

Evoluzione libera: $y(k) = A(0.3)^k + B(-0.8)^k$

Determiniamo i valori delle costanti A e B imponendo che siano soddisfatte le condizioni iniziali

$$y(0) = 2 \qquad y(1) = 4$$

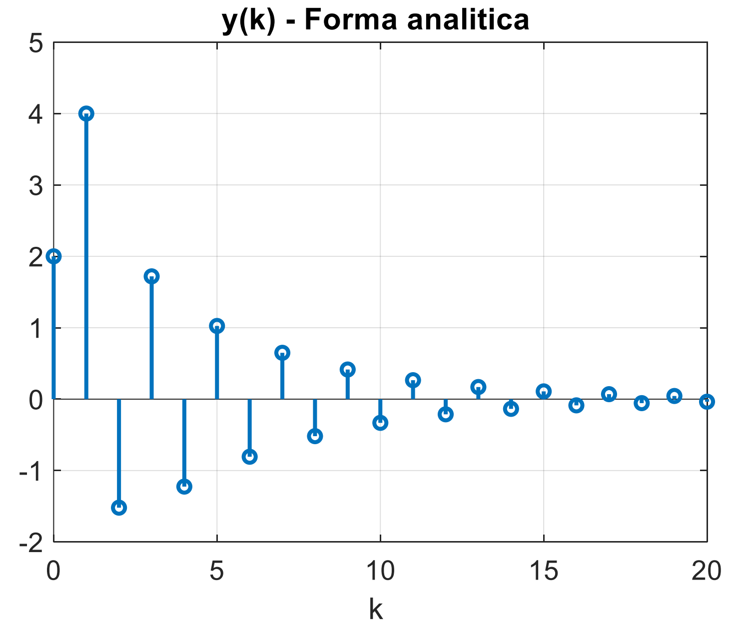
$$y(0) = A + B = 2$$

$$y(1) = 0.3A - 0.8B = 4$$

$$\Rightarrow \quad A = 5.09 \qquad B = -3.09$$

$$\Rightarrow \quad y(k) = 5.09(0.3)^k - 3.09(-0.8)^k$$

```
k=0:20;
y_libera=5.09*(0.3).^k-3.09*(-0.8).^k;

figure(1)
stem(k,y_libera,'LineWidth',2),grid
xlabel('k ')
title('y(k) - Forma analitica ')
set(gca,'FontSize',14)
axis([0 20 -2 5])
```



Realizziamo in Simulink direttamente il modello completo, comprensivo della sequenza di ingresso, alla quale si attribuirà valore identicamente nullo nel test della evoluzione libera.

Riscriviamo l'equazione alle differenze

$$y(k+2) + 0.5y(k+1) - 0.24\,y(k) = 2u(k+1) + u(k)$$

Dobbiamo trasformare l'equazione in modo che non compaiano termini del tipo $u(k+1)$.
Affinchè si possa realizzare in Simulink, l'equazione alle differenze può unicamente contenere il valore corrente u(k) dell'ingresso e valori passati $u(k-1), u(k-2)$, ecc.

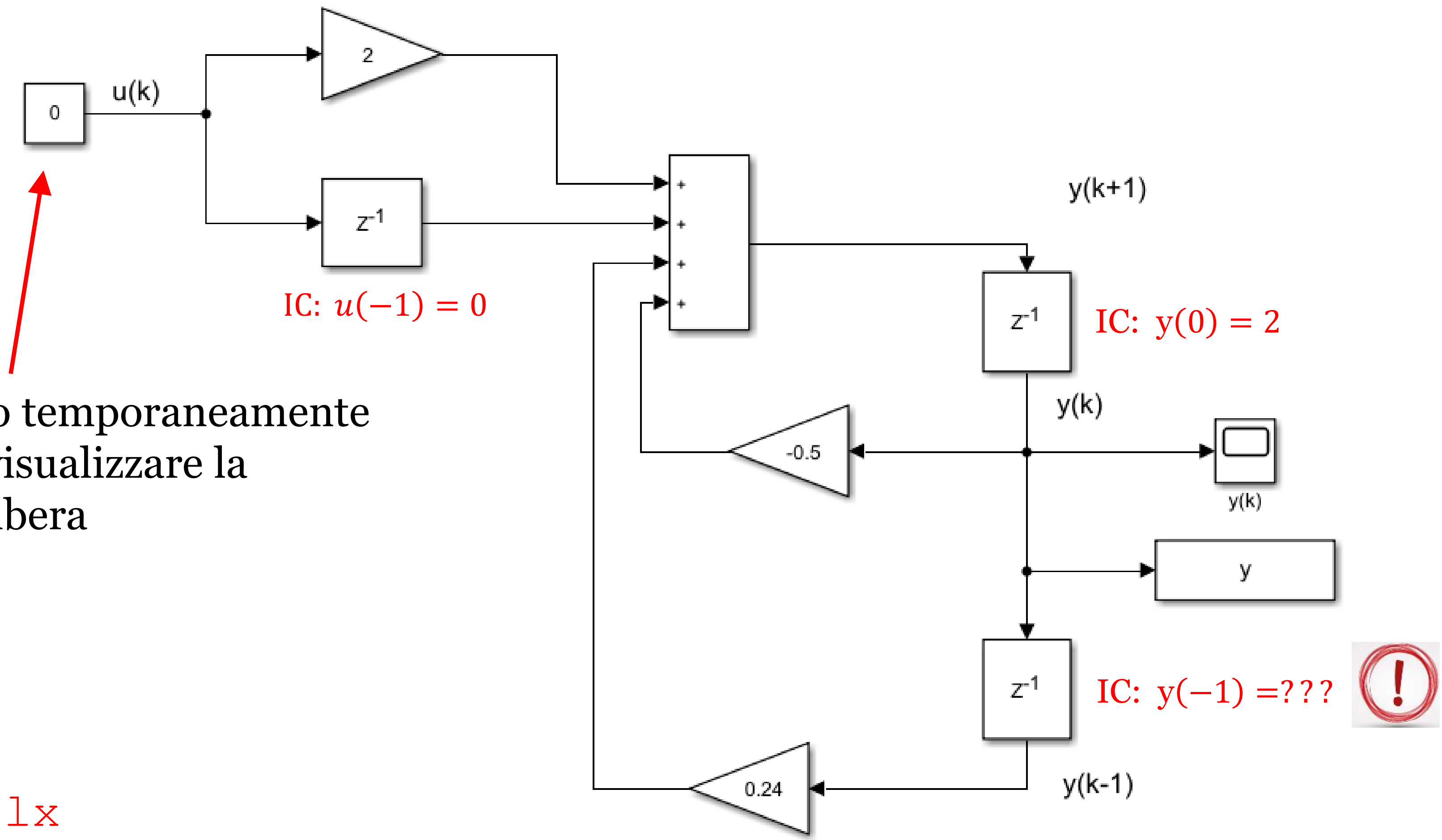
Trasliamo pertanto il modello di un passo all'indietro:

$$y(k+1) + 0.5y(k) - 0.24\,y(k-1) = 2u(k) + u(k-1)$$

e riscriviamolo in forma esplicita:

$$y(k+1) = -0.5y(k) + 0.24\,y(k-1) + 2u(k) + u(k-1)$$

$$y(k+1) = -0.5y(k) + 0.24\,y(k-1) + 2u(k) + u(k-1)$$



Inseriamo temporaneamente zero per visualizzare la risposta libera

Ead4.slx

Dobbiamo calcolare la condizione iniziale y(−1)

Per calcolare la condizione iniziale y(−1), possiamo valutare il modello seguente in corrispondenza del valore $k = 0$

$$y(k+1) = -0.5y(k) + 0.24\, y(k-1)$$

Si ottiene:

$$y(1) = -0.5y(0) + 0.24\, y(-1)$$

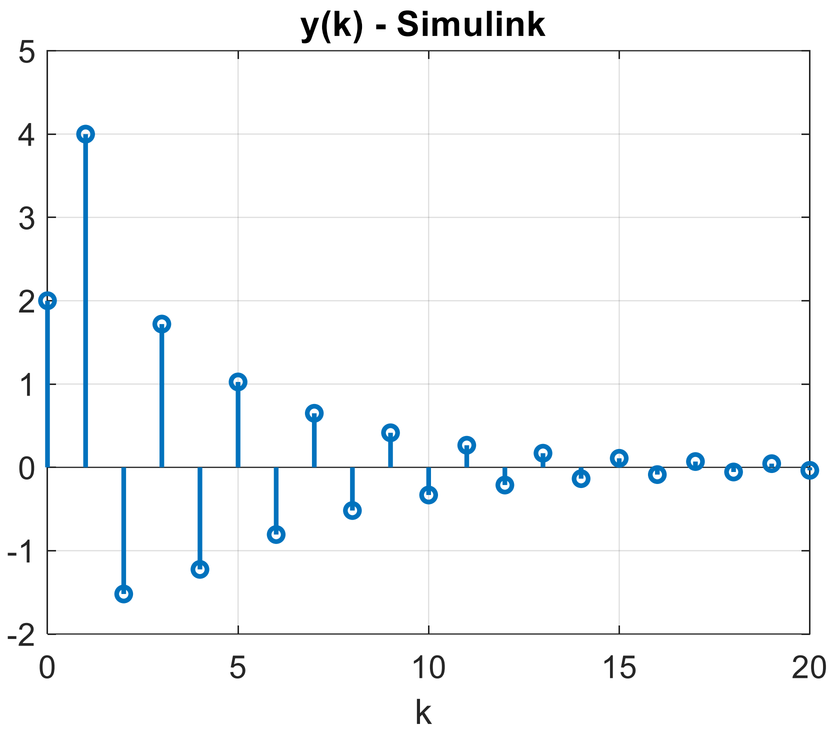
Sostituiamo i valori di $y(0) = 2, y(1) = 4$

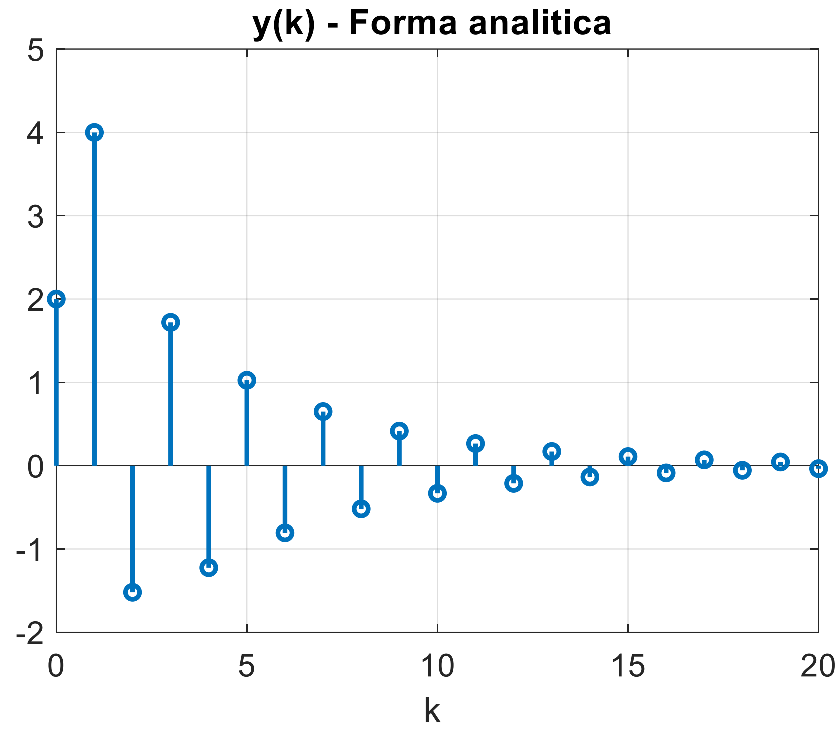
$$4 = -1 + 0.24\, y(-1)$$

$$y(-1) = \frac{4+1}{0.24} = 20.83$$

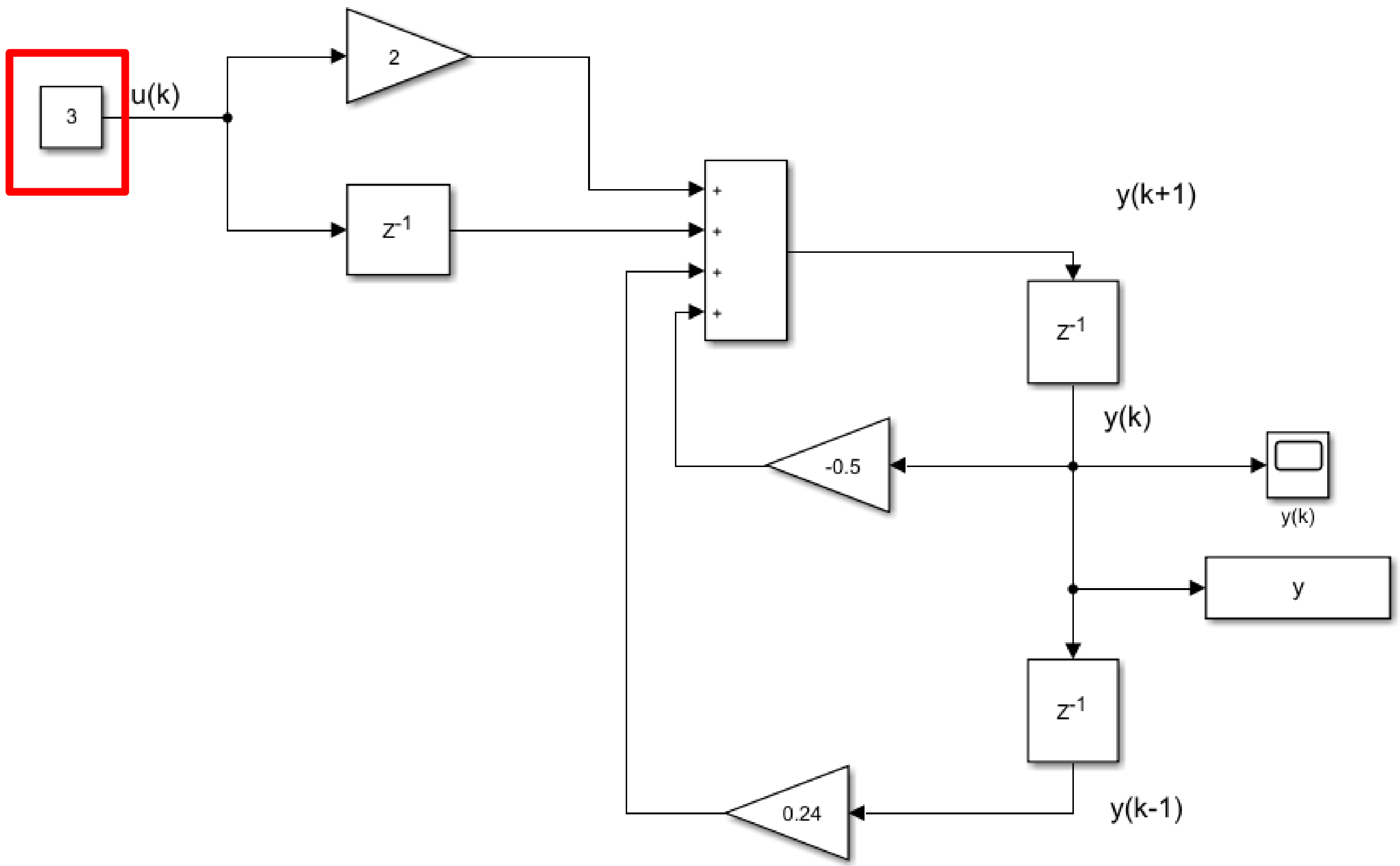
Inseriamo la condizione iniziale di 20.83 nel blocco Delay collocato piu in basso che produce in uscita la sequenza $y(k-1)$, ed avviamo la simulazione.

```
figure(1)
stem(y.Time,y.Data,'LineWidth',2),grid
xlabel('k ')
title('y(k) - Simulink ')
set(gca,'FontSize',14)
axis([0 20 -2 5])


k=0:20;
y_libera=5.09*(0.3).^k-3.09*(-0.8).^k;

figure(2)
stem(k,y_libera,'LineWidth',2),grid
xlabel('k ')
title('y(k) - Forma analitica ')
set(gca,'FontSize',14)
axis([0 20 -2 5])
```

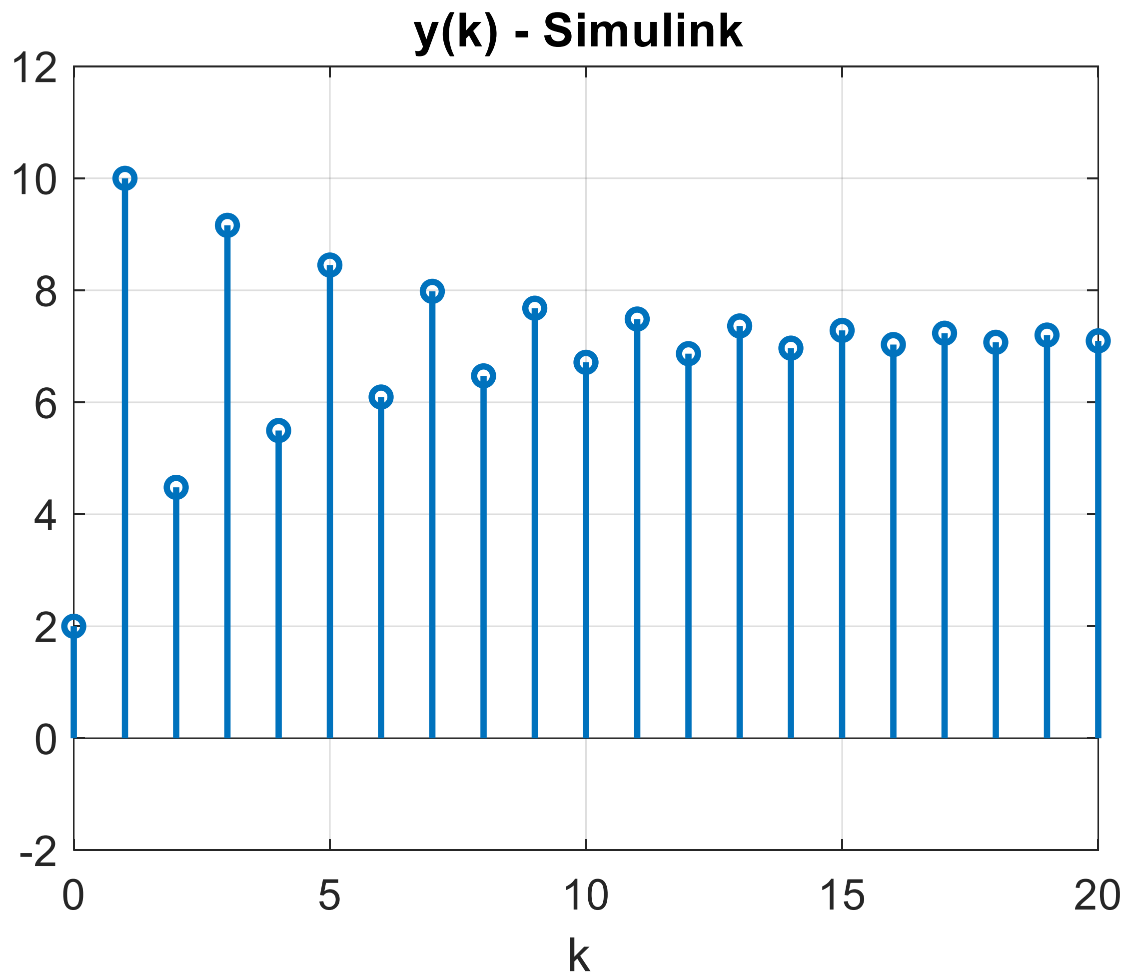

y(k) - Simulink
5
4
3
2
1
0
-1
-2
0
5
10
15
20
k


y(k) - Forma analitica
5
4
3
2
1
0
-1
-2
0
5
10
15
20
k

Ora inseriamo il valore non nullo per la sequenze in ingresso, e rieseguiamo la simulazione



```
figure(1)
stem(y.Time,y.Data,'LineWidth',2),grid
xlabel('k ')
title('y(k) - Simulink ')
set(gca,'FontSize',14)
axis([0 20 -2 12])
```



La sequenza di uscita non tende più a zero per effetto del termine forzante di ingresso.

Si ha però che la condizione iniziale $y(1) = 4$ non è più soddisfatta.

Come mai ?

Perché nel calcolare «all'indietro» la condizione iniziale y(−1) abbiamo in precedenza trascurato la presenza dell'ingresso in quanto ci stavamo occupando della risposta libera. Ripetiamo quindi la procedura considerando il modello completo

$$y(k+1) = -0.5y(k) + 0.24\,y(k-1) + 2u(k) + u(k-1)$$

e calcoliamo la condizione iniziale y(−1) valutandolo per $k = 0$

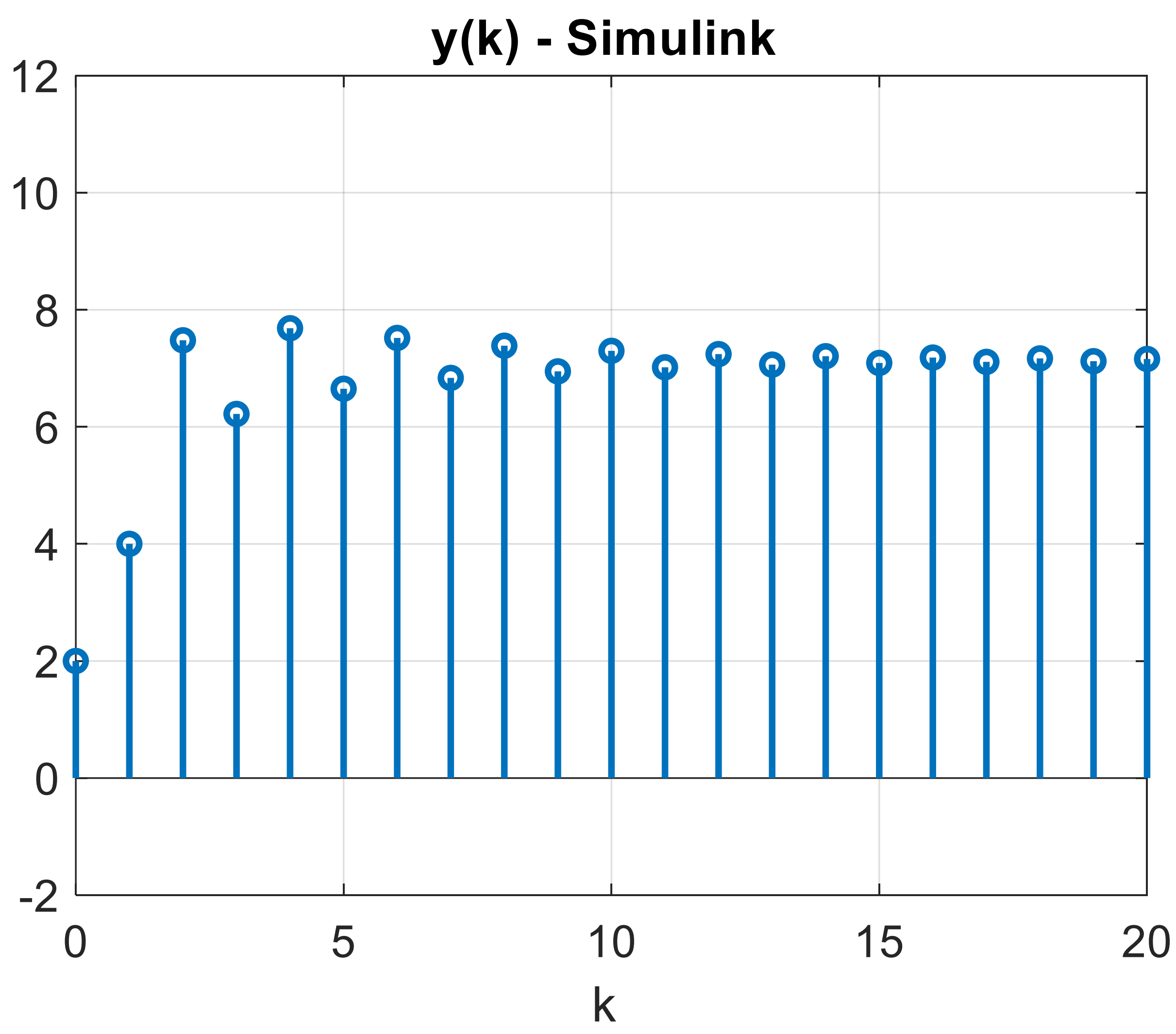
$$y(1) = -0.5y(0) + 0.24\,y(-1) + 2u(0) + u(-1)$$

Sostituiamo i valori di $y(1) = 4$ e $y(0) = 2$ (come prima) ma ora includiamo anche $u(0) = 3$ e $u(-1) = 0$

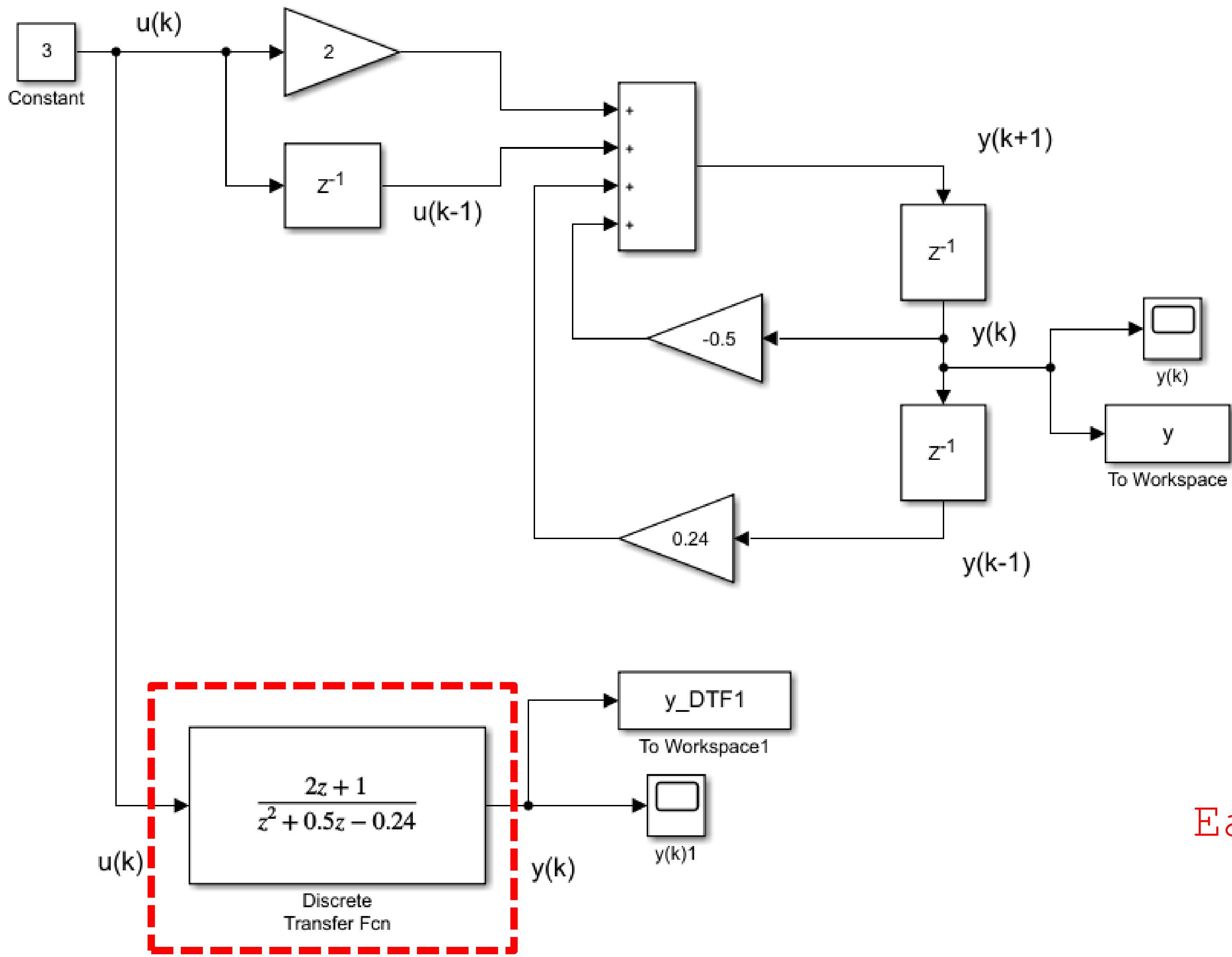
$$4 = -1 + 0.24\,y(-1) + 6 \qquad y(-1) = \frac{4+1-6}{0.24} = -4.16$$

Inseriamo la condizione iniziale di -4.16 nel blocco Delay che produce in uscita la sequenza $y(k-1)$, e ripetiamo la simulazione.

Ora l'evoluzione ottenuta è coerente con le condizioni iniziali $y(0) = 2, y(1) = 4$.

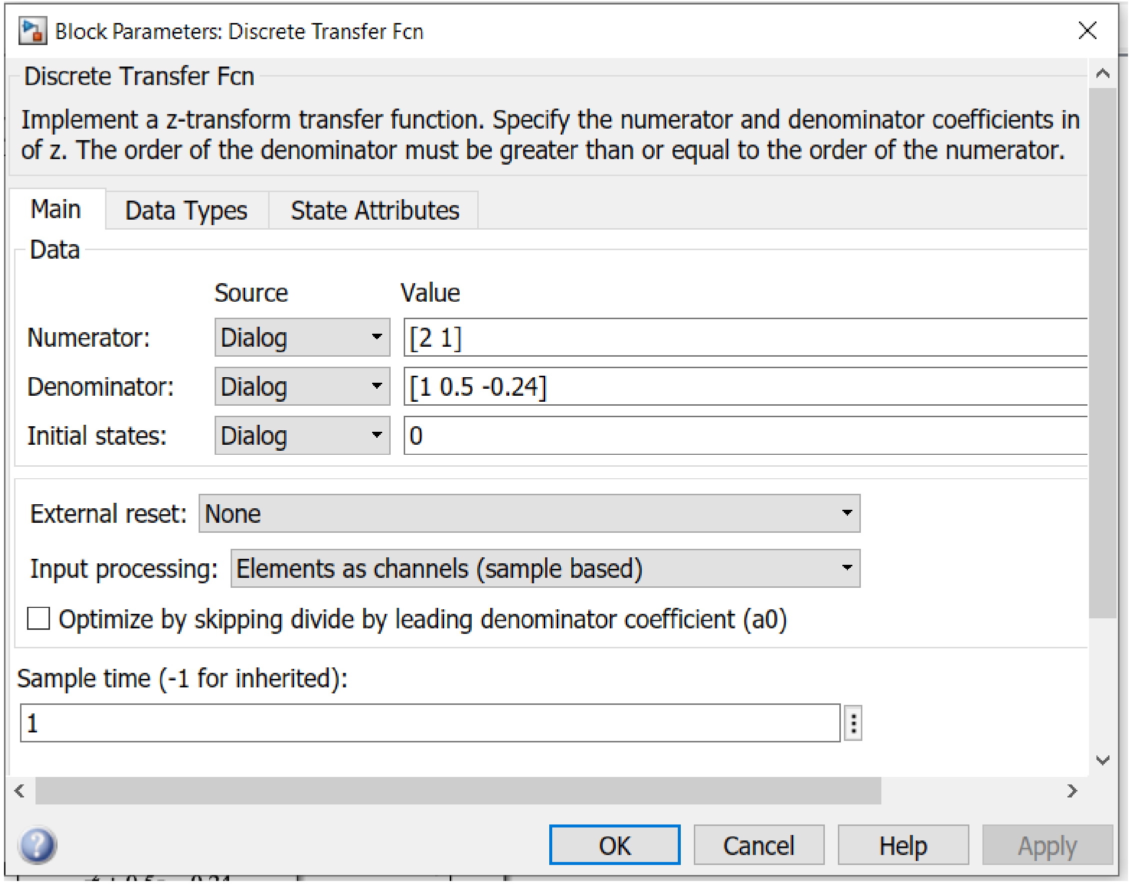


Il modello può essere realizzato in forma piu compatta impiegando un blocco più sofisticato, denominato «`Discrete Transfer Fcn`» che riceve in ingresso la sequenza di ingresso e produce in uscita la sequenza di uscita



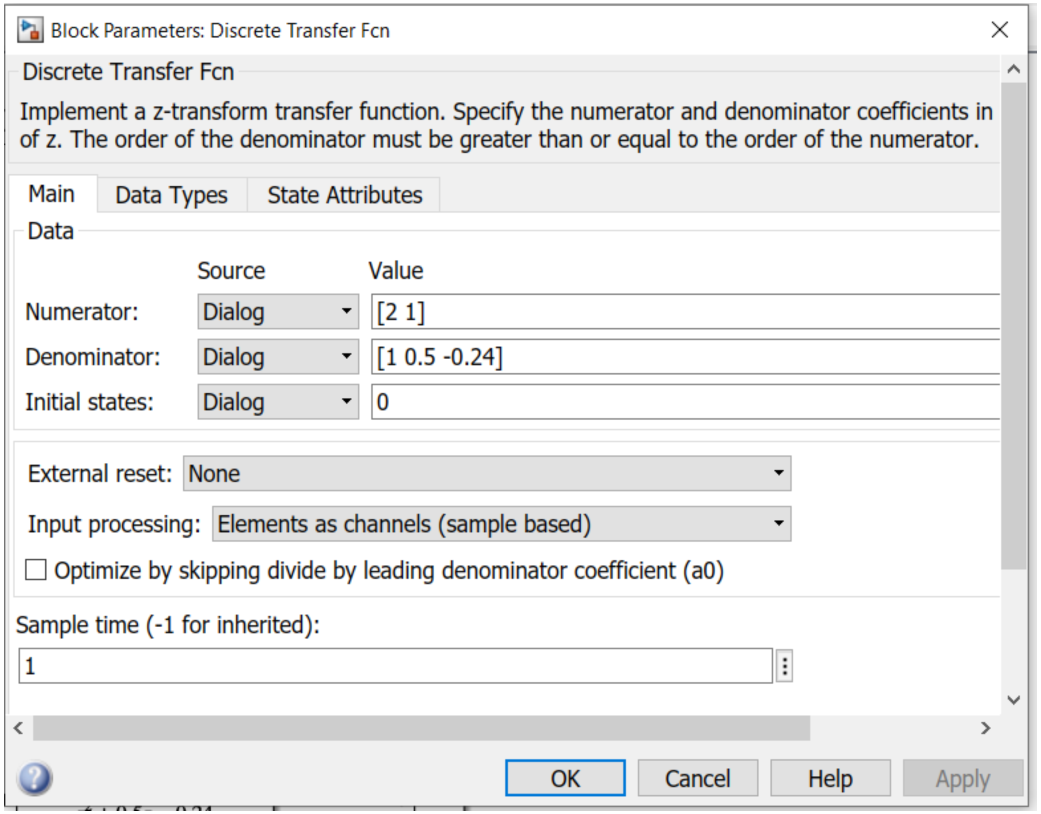
Ead5.slx

Nel modello `Ead5.slx` sono riportati, in maniera ridondante, sia il modello della equazione alle differenze da noi precedentemente realizzato che la sua forma «compatta» attraverso il blocco `Discrete Transfer Fcn`



Il blocco si parametrizza inserendo, nei campi «Numerator» e «Denominator» i coefficienti presenti nell’equazione alle differenze

$$y(k+1) + 0.5y(k) - 0.24\, y(k-1) = 2u(k) + u(k-1)$$



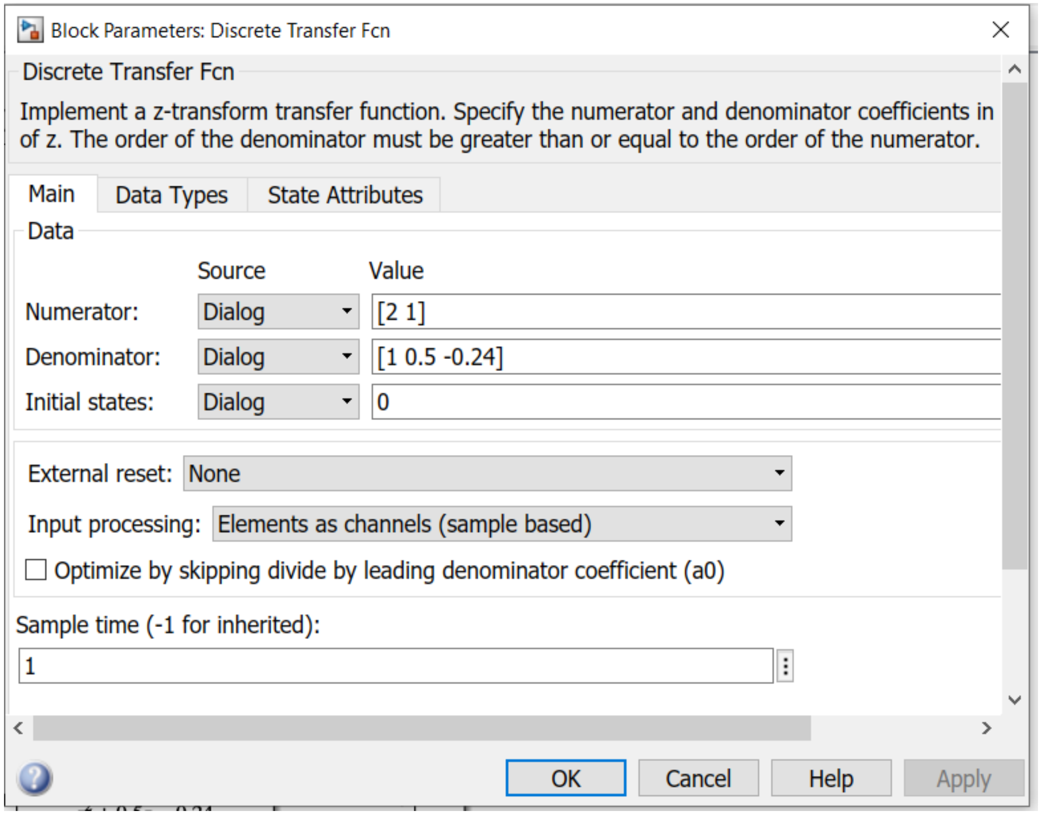
Il campo «`Denominator`, in particolare, deve contenere il vettore

$$[a_n \quad a_{n-1} \quad \dots a_0]$$

dei coefficienti che coinvolgono la sequenza di uscita, che in questo caso coincide con il vettore

$$[1 \quad 0.5 \quad -0.24]$$

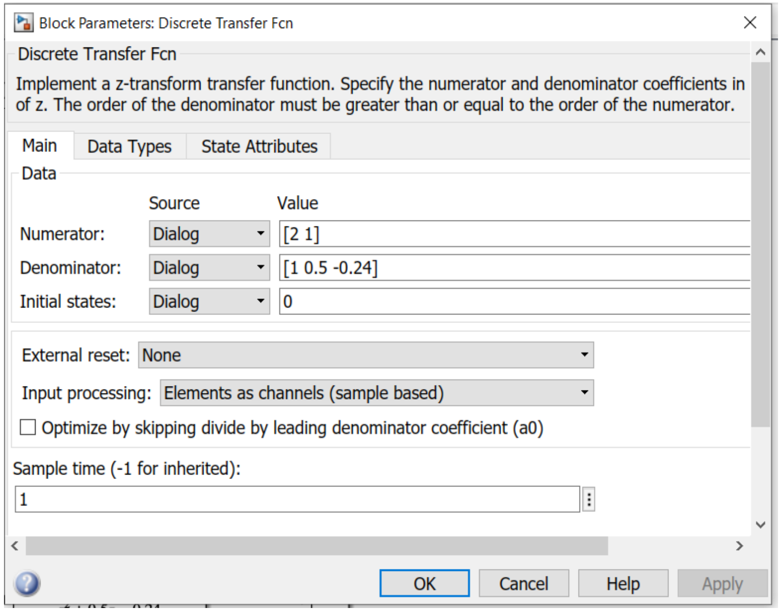
$$y(k+1) + 0.5y(k) - 0.24\, y(k-1) = 2u(k) + u(k-1)$$



Analogamente, il campo «`Numerator`» deve contenere il vettore

$$[b_m \quad b_{m-1} \quad \dots b_0]$$

dei coefficienti che coinvolgono la sequenza di ingresso, che in questo caso coincide con il vettore [2   1]



I problemi associati all'impiego di questo blocco nascono in relazione all'impostazione delle condizioni iniziali.

Se si desidera impiegare questo blocco per determinare una risposta forzata, cioè una risposta che parte da condizioni iniziali nulle, è sufficiente inserire 0 nel campo `Initial states` (come fatto in figura) e tutto funziona come desiderato, con evidenti vantaggi in termini di semplicità di modellazione.

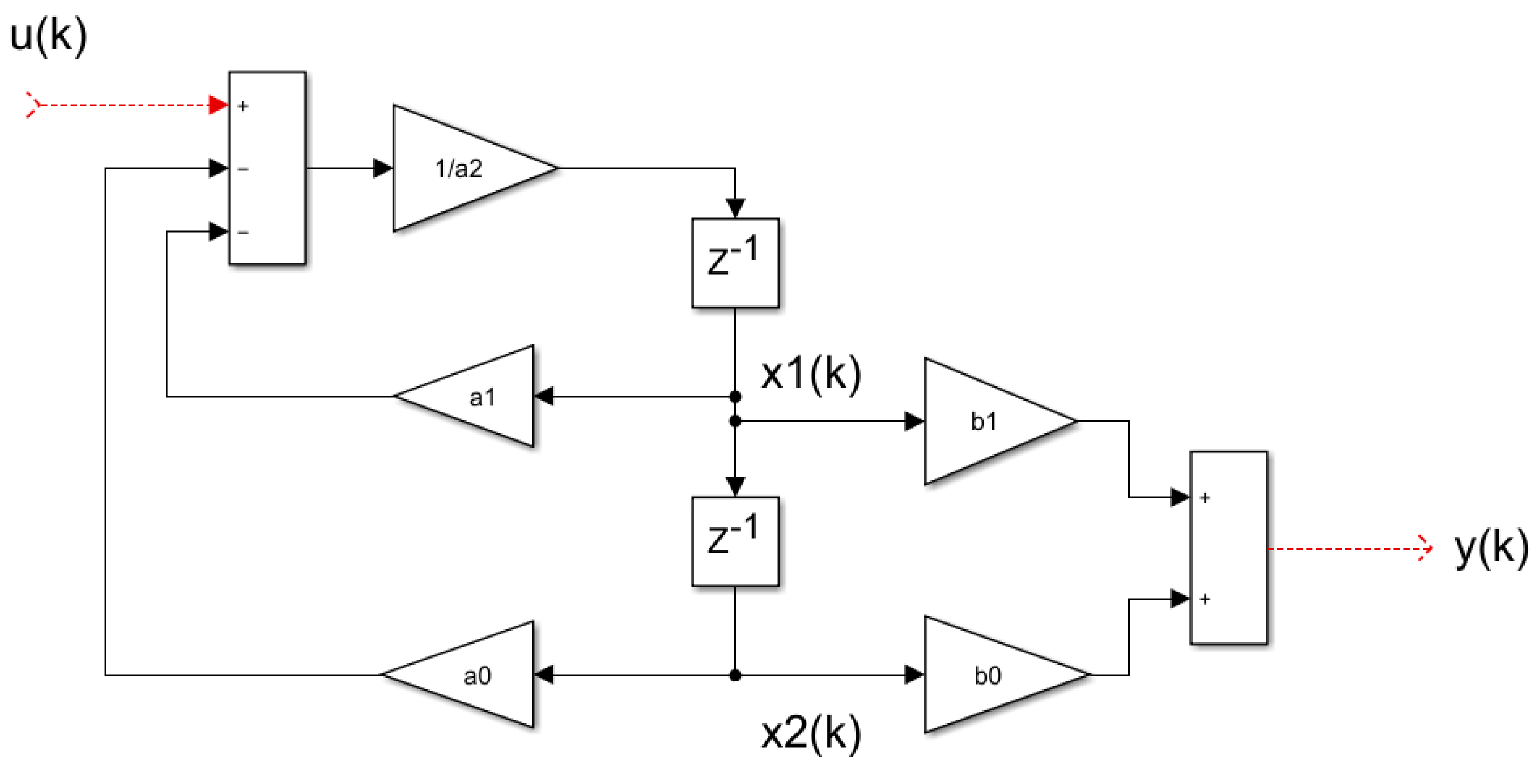
Ma se invece si desidera imporre determinate condizioni iniziali non nulle sui campioni $y(0), y(1), \dots, y(n-1)$ della sequenza di uscita, come nel caso del presente esercizio, la determinazione dei valori che devono essere inseriti nel campo `Initial states` diventa oltremodo complicata.

Discutiamo questo aspetto nella prossima slide.

Il blocco `Discrete Transfer Fcn` impiega una rappresentazione interna della equazione che risulta espressa, per una generica equazione alle differenze del secondo ordine

$$a_2 y(k+2) + a_1 y(k+1) + a_0 y(k) = b_1 u(k+1) + b_0 u(k)$$

dal seguente schema a blocchi:



Lo schema corrisponde alle seguenti relazioni

$$y(k) = b_1 x_1(k) + b_0 x_2(k)$$
$$x_1(k) = \frac{1}{a_2}[u(k-1) - a_1 x_1(k-1) - a_0 x_2(k-1)]$$
$$x_2(k) = x_1(k-1)$$

Gli stati dei quali si deve inserire la condizione iniziale nel campo `Initial states` sono le variabili $x_1(0)$ e $x_2(0)$

Tali quantità sono correlate alle condizioni iniziali $y(0)$ ed $y(1)$ dalle seguenti relazioni, che si ricavano facilmente dal modello soprariportato

$$y(0) = b_1 x_1(0) + b_0 x_2(0)$$

$$y(1) = b_1 x_1(1) + b_0 x_2(1) = \frac{b_1}{a_2}[u(0) - a_1 x_1(0) - a_0 x_2(0)] + b_0 x_1(0) =$$
$$= -\frac{b_1}{a_2}[u(0) + a_1 x_1(0) + a_0 x_2(0)] + b_0 x_1(0)$$

Sostituendo le condizioni iniziali $y(0) = 2, y(1) = 4$ del problema ed i valori dei parametri dell'equazione (nel sistema è coinvolto anche $u(0)$, il campione iniziale dell'ingresso, che vale 3) si ottiene:

$$2x_1(0) + x_2(0) = 2$$

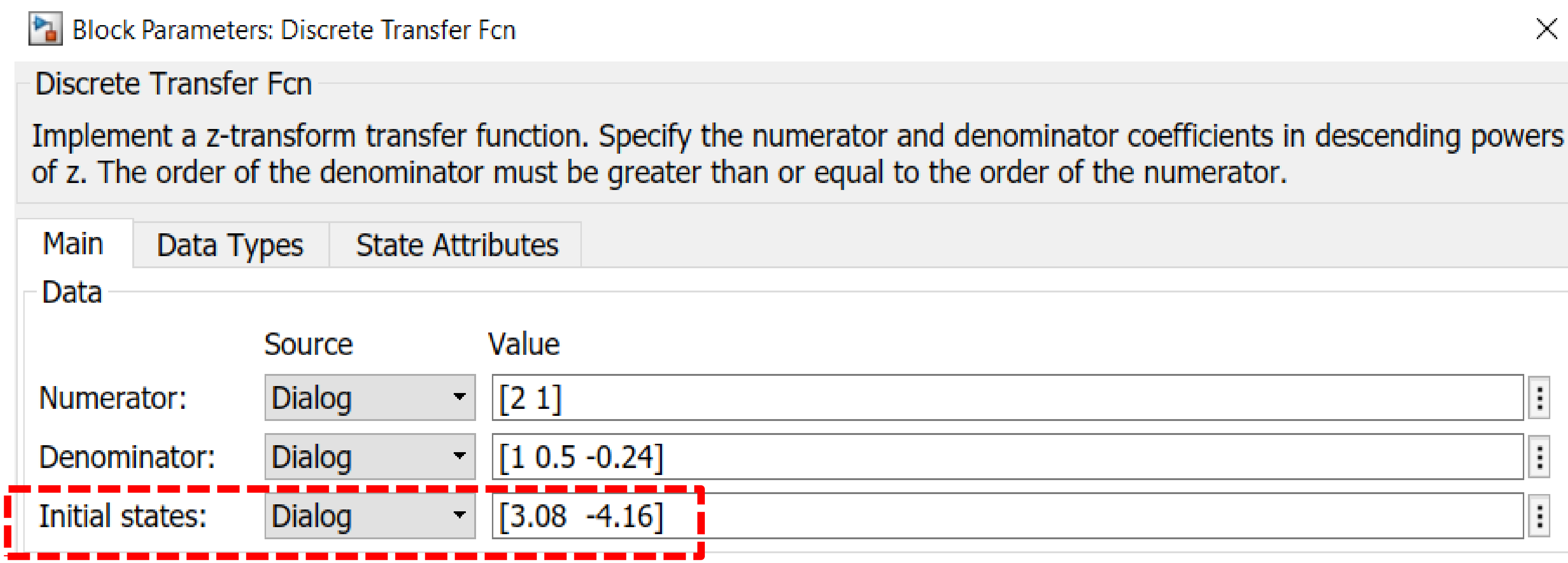
$$0.48x_2(0) = -2$$

Risolvendo il sistema si ricava pertanto come le condizioni iniziali $x_1(0)$ e $x_2(0)$ degli stati devono essere le seguenti:

$$x_1(0) = 3.08$$

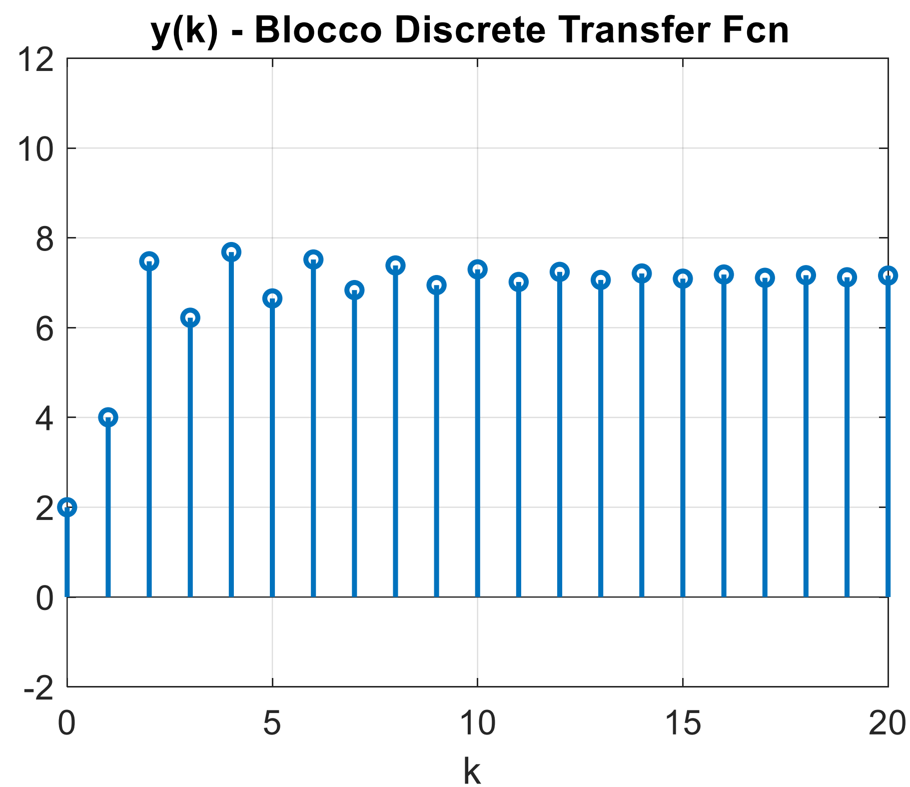
$$x_2(0) = 3.08$$

Inseriamo tali valori nel relativo campo, e rieseguiamo la simulazione



```
figure(1)
stem(y_DTF1.Time,y_DTF1.Data,'LineWidth',2),grid
xlabel('k ')
title('y(k) - Blocco Discrete Transfer Fcn ')
set(gca,'FontSize',14)
axis([0 20 -2 12])
```

Si può notare che ora la risposta ottenuta per la sequenza di uscita è coerente con le condizioni iniziali



La complessità di tale procedura di calcolo, ed in particolare la sua dipendenza dal valore dell’ingresso del blocco all’istante $k = 0$, suggeriscono come tale blocco sia indicato unicamente quando non siano rilevanti le condizioni iniziali dell’uscita, come peraltro si verifica nell’ambito della simulazione dei sistemi di controllo.

Passiamo ora ad un altro esercizio, che coinvolge stavolta un modello in variabili di stato

Esercizio 5. È data la rappresentazione in termini di variabili di stato di un sistema tempo-discreto lineare e stazionario automono

$$\begin{cases} x(k+1) &= A\,x(k) \\ y(k) &= C\,x(k) \end{cases}$$

dove

$$A = \begin{bmatrix} 0.5 & 1 \\ 0 & 0.5 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}.$$

(a) Si determini la dimensione dell'ingresso $u(k)$, dell'uscita $y(k)$ e dello stato $x(k)$.

(b) Si determini il polinomio caratteristico $P_A(s)$ della matrice A.

(c) Si calcolino gli autovalori e i modi della matrice A.

(d) Si calcoli la matrice di transizione dello stato a tempo discreto A^k mediante sviluppo di Sylvester.

(e) Si determini l'evoluzione dello stato e dell'uscita a partire dallo stato iniziale $x(0) = [2 \quad -1]^T$. A che valori tendono tali evoluzioni per $k \to \infty$ e perché?

(f) Verificare mediante Simulink la correttezza della risposta fornita al quesito (e) confrontando l'espressione analitica della evoluzione di stato e uscita con quelle fornite dal modello Simulink.

Soluzione dell'esercizio 5

(a) Si determini la dimensione dell'ingresso $u(k)$, dell'uscita $y(k)$ e dello stato $x(k)$.

L'ingresso ha dimensione nulla, in quanto si tratta di un sistema autonomo in cui non interviene nessuna sequenza di ingresso.

In conseguenza di questo fatto, le matrici B e D del modello in variabili di stato sono identicamente nulle.

(b) Si determini il polinomio caratteristico $P_A(s)$ della matrice A.

$$A = \begin{bmatrix} 0.5 & 1 \\ 0 & 0.5 \end{bmatrix}$$

Data la forma triangolare della matrice, possiamo già individuare i suoi autovalori, che coincidono con gli elementi diagonali.
La matrice in esame ha pertanto un autovalore $\lambda_1 = 0.5$ di molteplicità $\nu = 2$

Il polinomio caratteristico della matrice A si determina mediante l'espressione

$$P_{car}(\lambda) = \det(\lambda I - A)$$

$$\lambda I - A = \begin{bmatrix} \lambda - 0.5 & -1 \\ 0 & \lambda - 0.5 \end{bmatrix}$$

$$P_{car}(\lambda) = (\lambda - 0.5)^2$$

Il polinomio ottenuto è consistente con la conclusione tratta sugli autovalori della matrice triangolare A

(c) Si calcolino gli autovalori e i modi della matrice A.

Gli autovalori sono stati determinati nel quesito precedente.

I **modi** sono:

$$\mu_1(k) = (-0.5)^k \qquad \mu_2(k) = k(-0.5)^k$$

(d) Si calcoli la matrice di transizione dello stato a tempo discreto A^k mediante sviluppo di Sylvester.

Per il modello in esame, che ha ordine $n = 2$, lo sviluppo di Sylvester della matrice di transizione dello stato sarà:

$$A^k = \beta_0(k)I + \beta_1(k)A$$

Essendo $\lambda_1 = 0.5$ l'autovalore doppio della matrice A, avente molteplicità $\nu = 2$, il sistema per il calcolo delle funzioni $\beta_0(k)$ e $\beta_1(k)$ assume la forma

$$\begin{cases} \beta_0(k) + \lambda_1\beta_1(k) = {\lambda_1}^k \\ \beta_1(k) = k{\lambda_1}^{k-1} \end{cases} \Rightarrow \begin{cases} \beta_0(k) + 0.5\beta_1(k) = (0.5)^k \\ \beta_1(k) = k(0.5)^{k-1} = 2k(0.5)^k \end{cases}$$

Si ha quindi:

$$\beta_0(k) = (0.5)^k - k(0.5)^k$$

$$\beta_1(k) = 2k(0.5)^k$$

$$A^k = [(0.5)^k - k(0.5)^k]I + 2k(0.5)^k A$$

$$A = \begin{bmatrix} 0.5 & 1 \\ 0 & 0.5 \end{bmatrix}$$

Sviluppando i conti:

$$A^k = \begin{bmatrix} (0.5)^k - k(0.5)^k & 0 \\ 0 & (0.5)^k - k(0.5)^k \end{bmatrix} + \begin{bmatrix} k(0.5)^k & 2k(0.5)^k \\ 0 & k(0.5)^k \end{bmatrix} =$$

$$= \begin{bmatrix} (0.5)^k & 2k(0.5)^k \\ 0 & (0.5)^k \end{bmatrix}$$

Calcolo della matrice di transizione dello stato mediante il Symbolic Math Toolbox di Matlab

```
A = [ 0.5 1; 0 0.5 ];

syms k
assume(k,'integer')

Atok = simplify(A^k)
```

Definizione di una variabile simbolica k ristretta al campo dei valori interi

Calcolo della matrice A^k e successiva semplificazione

Si ottiene:

```
Atok =

[ 1/2^k, (2*k)/2^k]
[     0,     1/2^k]
```

L’istruzione `assume(k,'integer’)` restringe la variabile simbolica `k` a valori interi, e ciò agevola l’algoritmo di calcolo.

La funzione `simplify` applicata alla espressione `A^k` «forza» Matlab ad operare ulteriori semplificazioni alla soluzione trovata.

Se tale funzione non viene impiegata, la risposta che si ottiene è:

```
A = [ 0.5 1; 0 0.5 ]
syms k
assume(k,'integer')
Atok = (A^k)
```

```
Atok =

[ exp(-k*log(2)), 2*k*exp(-k*log(2))]
[              0,     exp(-k*log(2))]
```

che è corretta ma sicuramente meno maneggevole rispetto a quella ottenuta impiegando la funzione `simplify`.

Calcolo del polinomio caratteristico mediante il Symbolic Math Toolbox di Matlab

```
A = [ 0.5 1; 0 0.5 ]
syms lambda
Pcar=det(lambda*eye(2)-A);
Pcar=simplify(Pcar)
```

```
Pcar =

(2*lambda - 1)^2/4
```

Il Symbolic Math Toolbox restituisce il polinomio:

$$P_{car}(\lambda) = \frac{1}{4}(2\lambda - 1)^2$$

(e) Si determini l'evoluzione dello stato e dell'uscita a partire dallo stato iniziale $x(0) = [2 \quad -1]^T$. A che valori tendono tali evoluzioni per $k \to \infty$ e perché?

Le evoluzioni dello stato e dell'uscita in un sistema autonomo espresso in variabili di stato sono:

$$\boldsymbol{x}(k) = \boldsymbol{A}^k \boldsymbol{x}(0)$$

$$y(k) = C\boldsymbol{A}^k \boldsymbol{x}(0)$$

Possiamo già affermare, prima ancora di determinarle in forma analitica, che le evoluzioni di stato e uscita tenderanno a zero per $k \to \infty$, in conseguenza del fatto che gli autovalori della matrice A sono entrambi interni al disco unitario e quindi tutti i modi del sistema (che abbiamo determinato in precedenza) sono stabili.

Sviluppiamo i conti determinando in forma analitica le evoluzioni di stato e uscita

$$\boldsymbol{x}(k) = \boldsymbol{A}^k\boldsymbol{x}(0) = \begin{bmatrix} (0.5)^k & 2k(0.5)^k \\ 0 & (0.5)^k \end{bmatrix}\begin{bmatrix} 2 \\ -11 \end{bmatrix}$$

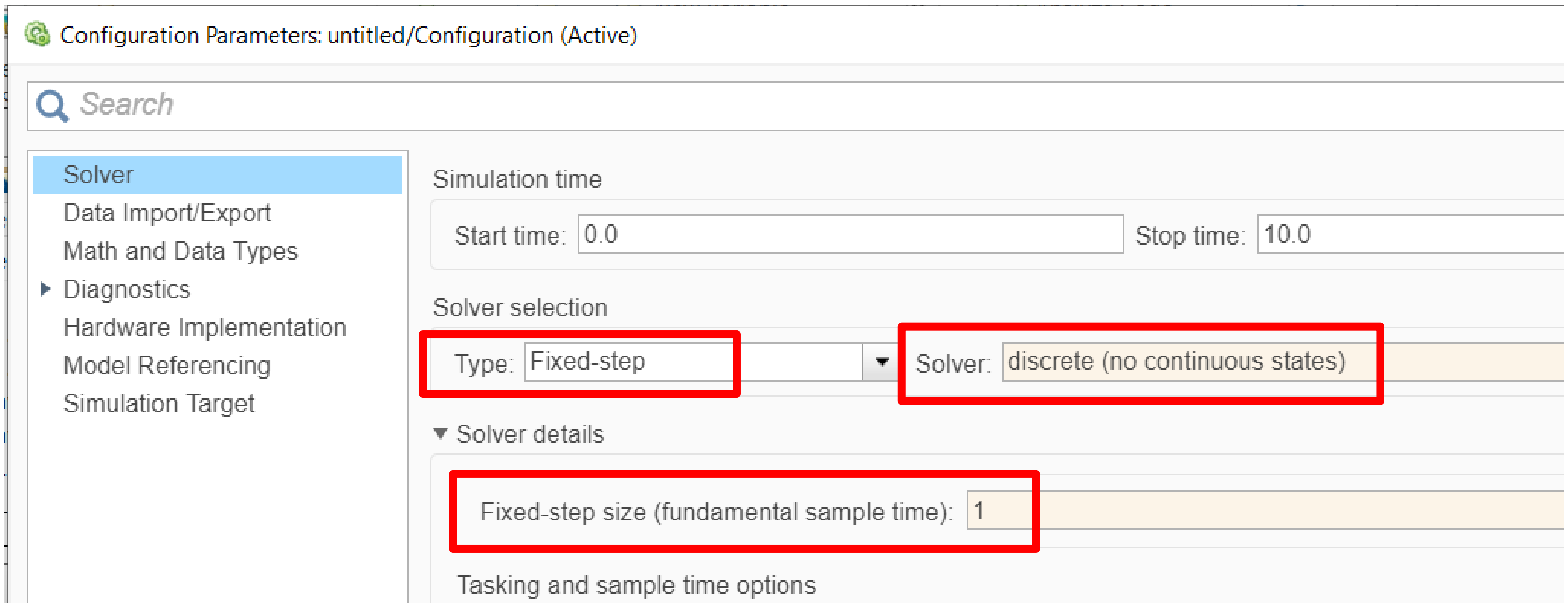
$$x_1(k) = 2(0.5)^k - 22k(0.5)^k$$

$$x_2(k) = -11(0.5)^k$$

$$y(k) = C\boldsymbol{A}^k\boldsymbol{x}(0) = \begin{bmatrix} 1 & 0 \end{bmatrix}\begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} = x_1(k) = 2(0.5)^k - 22k(0.5)^k$$

(f) Verificare mediante Simulink la correttezza della risposta fornita al quesito (e) confrontando l'espressione analitica della evoluzione di stato e uscita con quelle fornite dal modello Simulink.

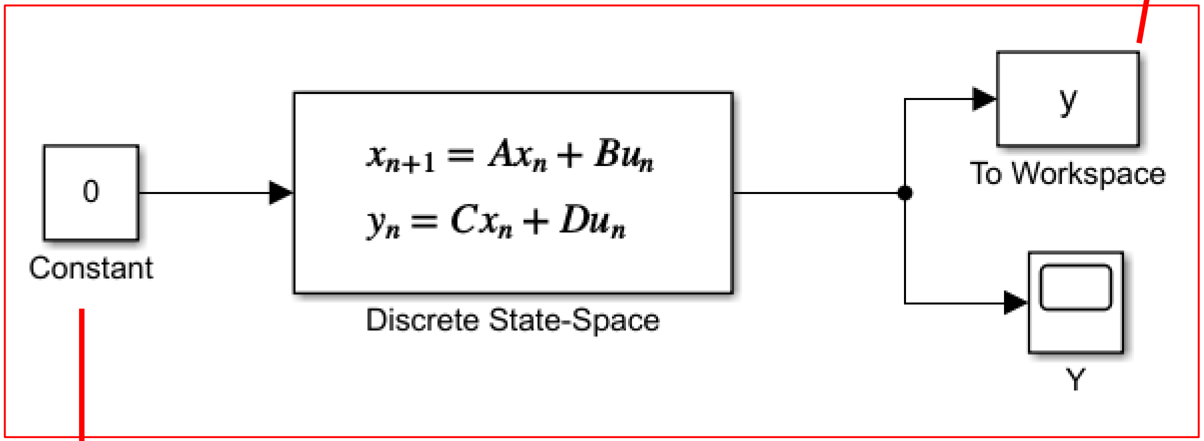
Aprire un nuovo modello, e configurare i parametri del Solver (Model Settings) come segue (in modo analogo a quanto fatto per le precedenti simulazioni con le equazioni alle differenze)



Per la modellazione di sistemi a tempo discreto in variabili di stato, il blocco fondamentale è il blocco **«`Discrete State-Space`»** (libreria discrete)

Si realizzi il seguente modello Simulink:

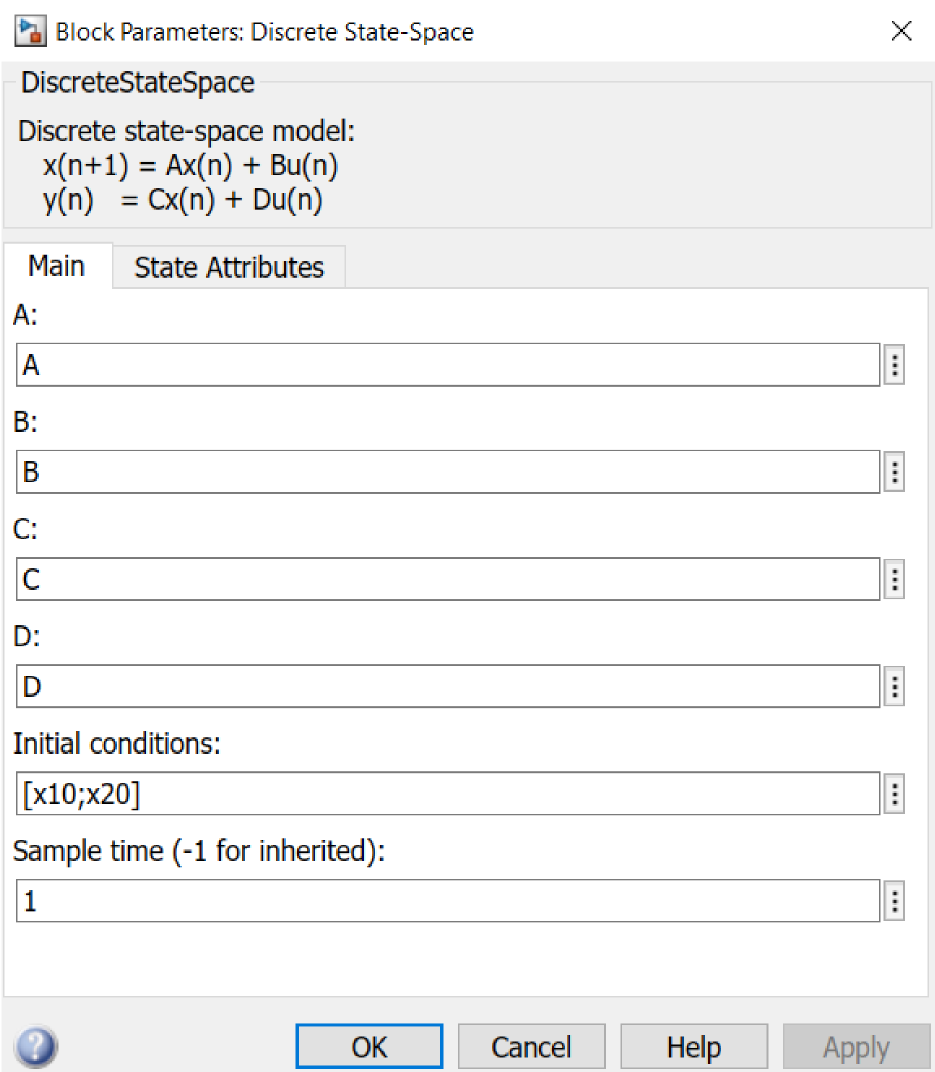
Nella finestra di parametrizzazione:
Save format = `timeseries`



Ingresso nullo in quanto il sistema è autonomo

`Ex5_es1_onlyoutput.slx`

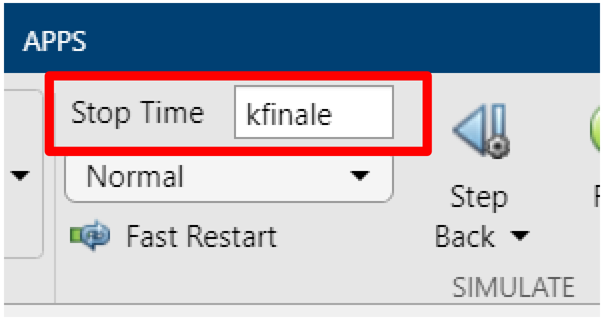
Il blocco «Discrete State-Space» riceve in ingresso u(k), e produce in uscita la sequenza di uscita y(k).



Lo parametrizziamo mediante variabili che definiremo all'interno in uno script.

Si devono impostare le 4 matrici A,B,C,D del modello, le condizioni iniziali dello stato, ed il passo di campionamento (Sample Time) **da scegliersi pari ad 1**

Nel modello Simulink viene parametrizzata anche la durata della simulazione, per mezzo della variabile `kfinale`



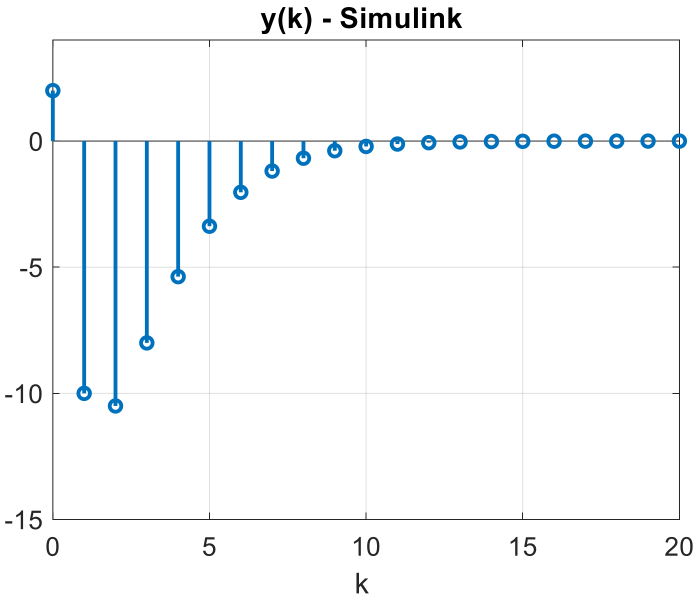
Script di parametrizzazione da eseguire prima di avviare la simulazione

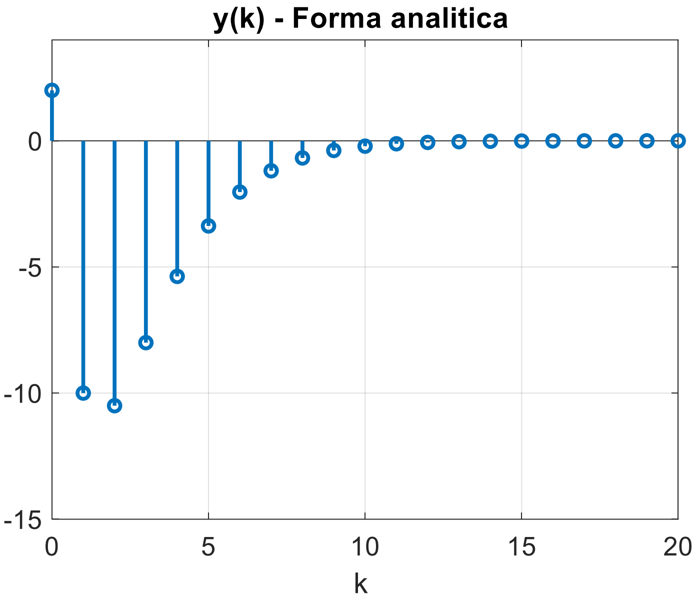
```
%% definizione parametri
kfinale=20;
A = [ 0.5 1; 0 0.5 ]
B=[0;0]
C=[1 0];
D=0;
x10=2;
x20=-11;
```

La sequenza di uscita puo essere visualizzata facendo doppio click sul blocco «Scope»

Inseriamo nello script ulteriori istruzioni volte a creare due grafici che confrontino fra loro la sequenza di uscita determinata con il modello Simulink con la relativa espressione analitica $y(k) = 2(0.5)^k - 22k(0.5)^k$ che abbiamo calcolato poc'anzi.

```
%% creazione grafici
figure(1)
stem(y.Time,y.Data,'LineWidth',2),grid
xlabel('k ')
title('y(k) - Simulink ')
set(gca,'FontSize',14)
axis([0 kfinale -15 4])

k=0:kfinale;
y_analytic=2*(0.5).^k-22.*k.*(0.5).^k

figure(2)
stem(k,y_analytic,'LineWidth',2),grid
xlabel('k ')
title('y(k) - Forma analitica ')
set(gca,'FontSize',14)
axis([0 kfinale -15 4])
```


y(k) - Simulink
0
-5
-10
-15
0
5
10
15
20
k


y(k) - Forma analitica
0
-5
-10
-15
0
5
10
15
20
k

Questo approccio alla modellazione non consente di visualizzare le evoluzioni delle variabili di stato, ma unicamente quella dell'uscita.

Presentiamone una variante che rende accessibili, all'interno del modello Simulink, anche le evoluzioni delle variabili di stato.

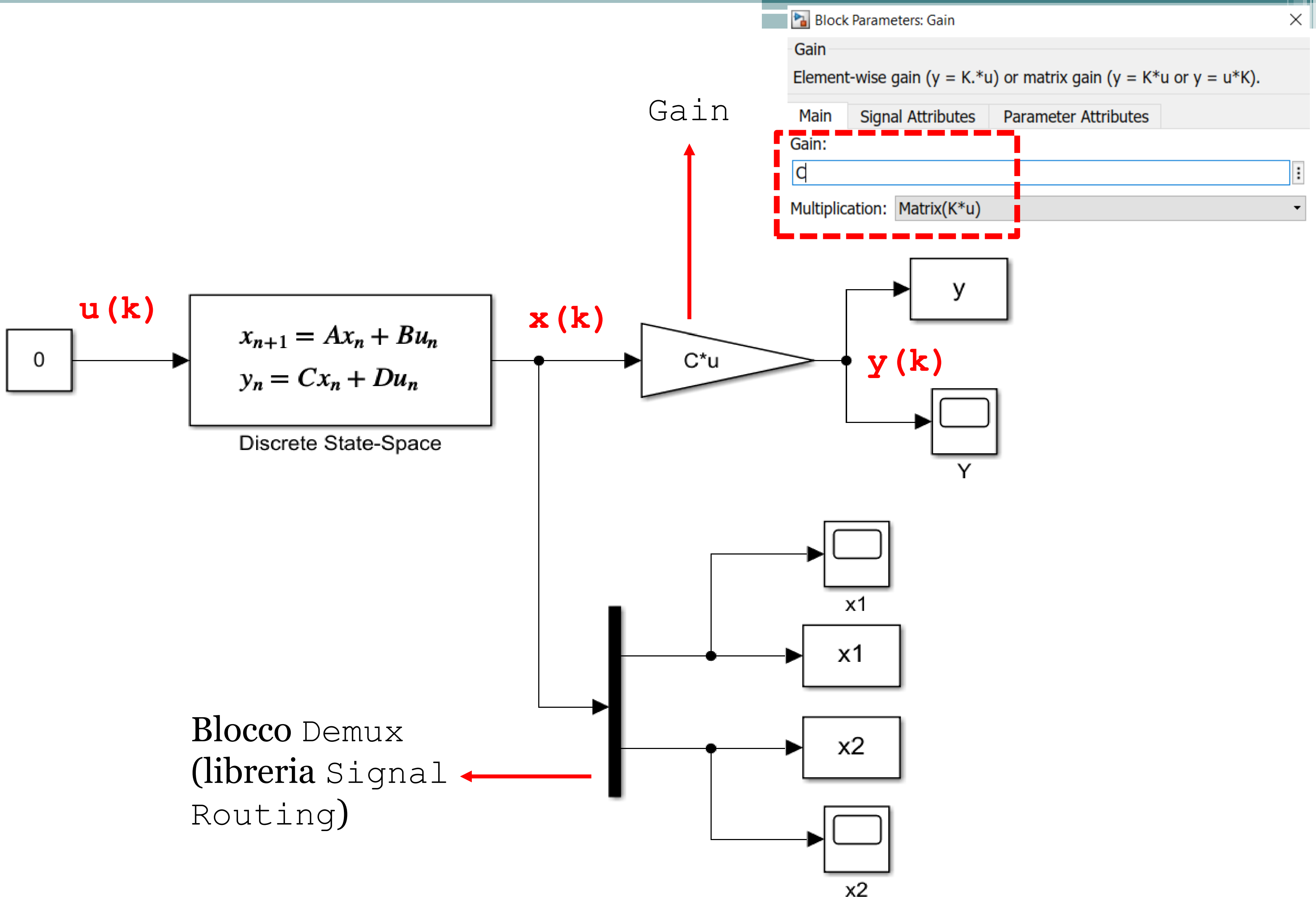
Il «trucco» è quello di considerare un modello in variabili di stato fittizio, nel quale la variabile di uscita coincida con lo stato, e successivamente, a valle di tale blocco, moltiplicare lo stato per la matrice C per ottenere l'uscita.

Tale modello fittizio assume la forma seguente

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \mathbf{B}\,\mathbf{u}(k)$$

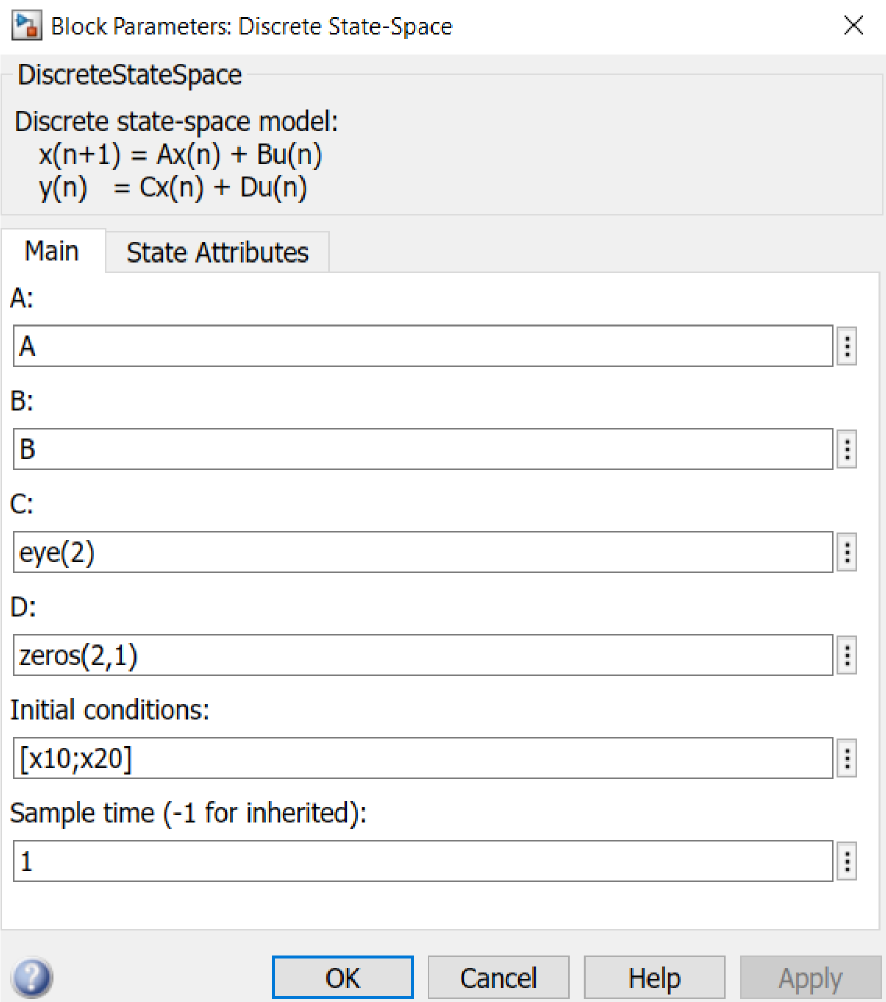
$$\mathbf{y}(k) = \boldsymbol{x}(k)$$

che corrisponde ad un modello in variabili di stato avente come matrice C la matrice identità di dimensione n, e come matrice D una matrice di zeri di dimensione opportuna.



Ex5_es1_fullstate.slx

Il blocco Discrete State Space è parametrizzato come segue

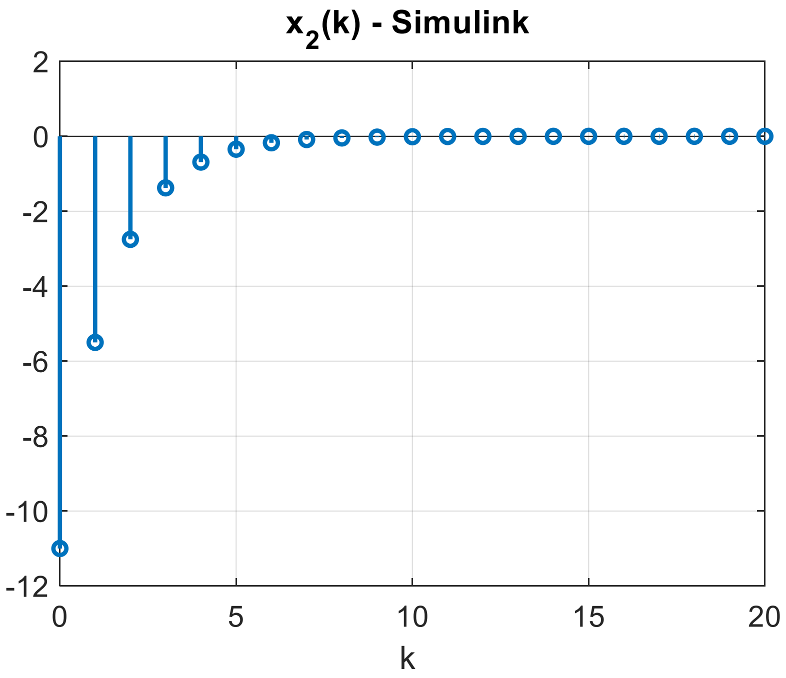


Script da eseguire prima di avviare il modello Simulink:

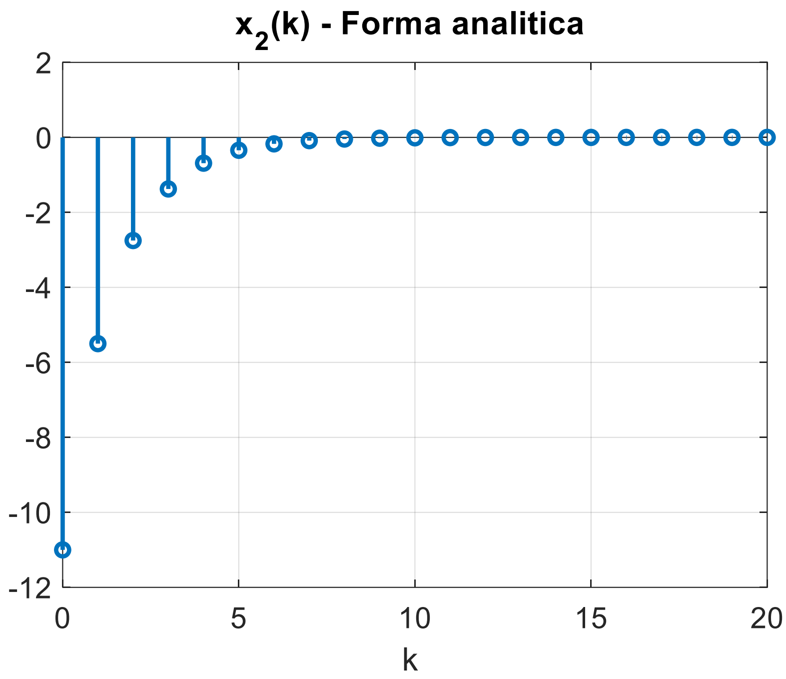
```
A = [ 0.5 1; 0 0.5 ]
B=[0;0]
C=[1 0];
x10=2;
x20=-11;
```

Ora possiamo ispezionare anche l'evoluzione dello stato $x_2(k)$, e verificare se la sua evoluzione coincide con quella in forma analitica $x_2(k) = -11(0.5)^k$ determinata in precedenza

```
%% creazione grafici
figure(1)
stem(x2.Time,x2.Data,'LineWidth',2),grid
xlabel('k ')
title('x_2(k) - Simulink ')
set(gca,'FontSize',14)
axis([0 kfinale -12 2])

k=0:kfinale;
x2_analytic=-11*(0.5).^k

figure(2)
stem(k,x2_analytic,'LineWidth',2),grid
xlabel('k ')
title('x_2(k) - Forma analitica ')
set(gca,'FontSize',14)
axis([0 kfinale -12 2])
```



x2(k) - Simulink
2
0
-2
-4
-6
-8
-10
-12
0
5
10
15
20
k



x2(k) - Forma analitica
2
0
-2
-4
-6
-8
-10
-12
0
5
10
15
20
k