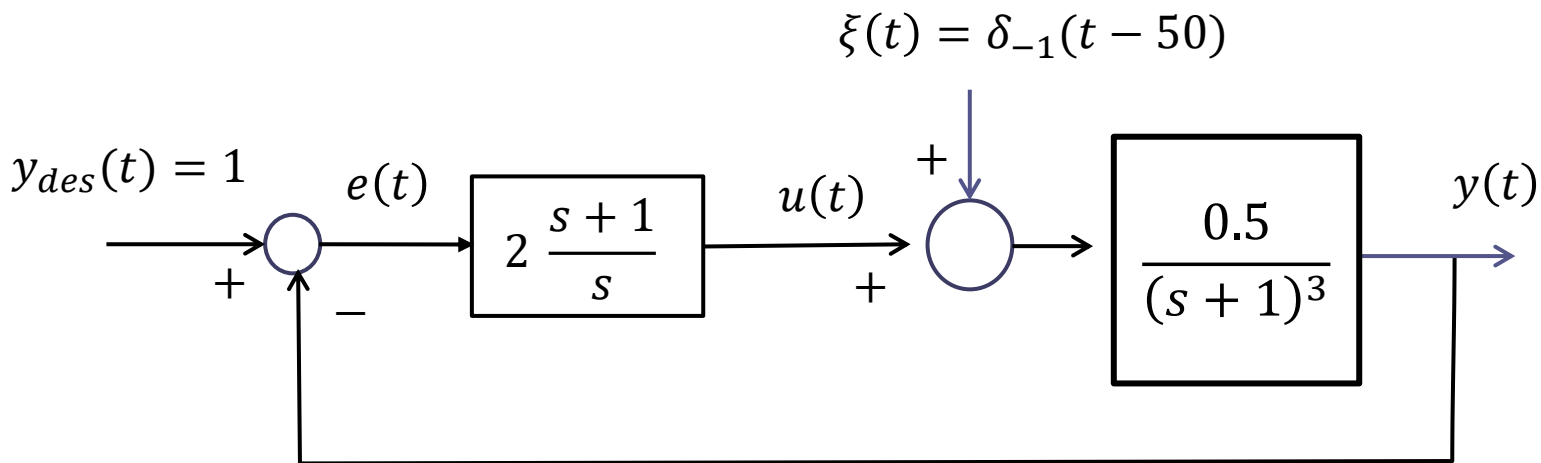


Controlli Automatici

Implementazione digitale dei controllori

Problema n. 1

Implementare per via numerica il seguente sistema di controllo e testarne le prestazioni in corrispondenza di diversi valori del periodo di campionamento



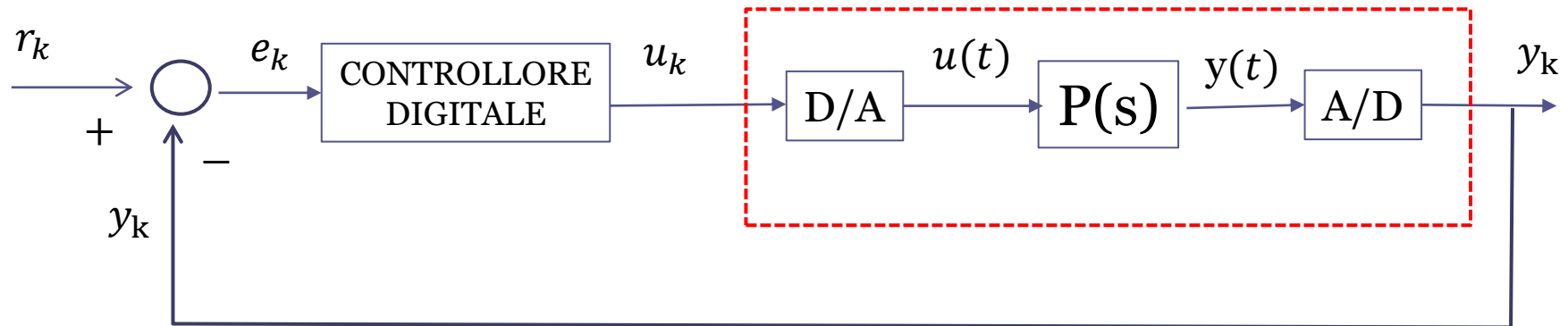
Regolatore PI con

$$K_P = 2$$

$$T_I = 1s$$

Disturbo di ampiezza unitaria che interviene a $t = 50$

Richiami

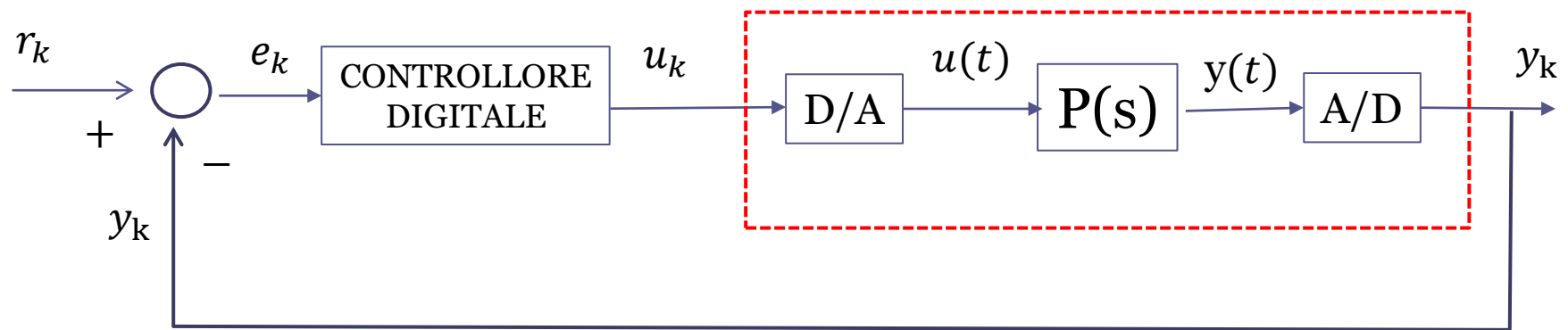


Il convertitore analogico digitale (A/D) estrae dal segnale $y(t)$ la sequenza numerica

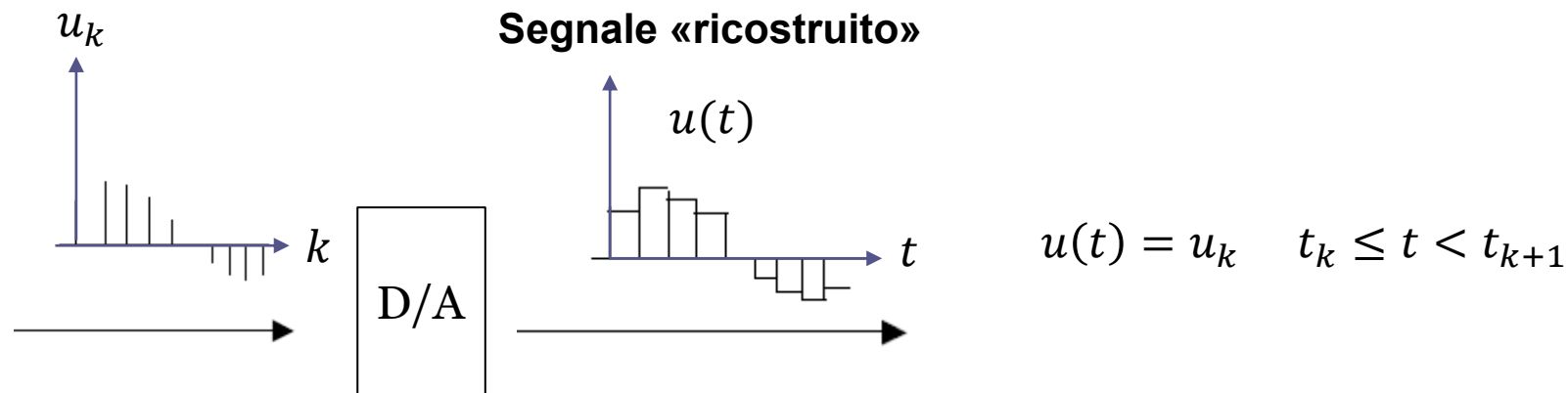
$$y_k = y(kT_c)$$

in cui T_c viene detto «**periodo di campionamento**»

Il blocco «**controllore digitale**» si attiva ogni T_c secondi effettuando il calcolo del segnale u_k ad ogni istante $t_k = kT_c$ con $k = 0, 1, 2, \dots$



Una volta che all'istante generico $t_k = kT_c$ il controllore digitale ha determinato la propria uscita u_k , questo valore transita attraverso un convertitore digitale/analogico (D/A) che converte la sequenza numerica u_k in un segnale $u(t)$ **costante a tratti**, di fatto «mantenendone costante» il valore fino al successivo istante di campionamento



Implementazione digitale di un controllore PID

$$e_{-1} = e_{-2} = u_{PID,-1} = 0$$

$$b_0 = K_P \left(1 + \frac{T_c}{T_I} + \frac{T_D}{T_c} \right) \quad b_1 = -K_P \left(1 + 2 \frac{T_D}{T_c} \right) \quad b_2 = \frac{K_P T_D}{T_c}$$

$$u_{PID,k} = b_0 e_k + b_1 e_{k-1} + b_2 e_{k-2} + u_{PID,k-1}$$

Implementazione digitale di un controllore PI

$$e_{-1} = u_{PI,-1} = 0$$

$$b_0 = K_P \left(1 + \frac{T_c}{T_I} \right)$$

$$u_{PI,k} = b_0 e_k - K_P e_{k-1} + u_{PI,k-1}$$

Controllore PID

Pseudo-codice di controllo

Inizializzazione: $k = 0$ $u_{PID,-1} = e_{-1} = e_{-2} = 0$

$$b_0 = K_P \left(1 + \frac{T_c}{T_I} + \frac{T_D}{T_c} \right) \quad b_1 = -K_P \left(1 + 2 \frac{T_D}{T_c} \right) \quad b_2 = \frac{K_P T_D}{T_c}$$

Ogni T_c secondi:

Leggi y_k dal registro del convertitore A/D

Calcola $e_k = r_k - y_k$

Calcola $u_{PID,k} = b_0 e_k + b_1 e_{k-1} + b_2 e_{k-2} + u_{PID,k-1}$

Scrivi $u_{PID,k}$ nel registro del convertitore D/A

$u_{PID,k-1} := u_{PID,k}$

$e_{k-2} := e_{k-1}$

$e_{k-1} := e_k$

$k = k + 1$

Controllore PI

Pseudo-codice di controllo

Inizializzazione: $k = 0$ $u_{PI,-1} = e_{-1} = 0$

Ogni T_c secondi:

Leggi y_k dal registro del convertitore A/D

Calcola $e_k = r_k - y_k$

Calcola $u_{PI,k} = K_P \left(1 + \frac{T_c}{T_I} \right) e_k - K_P e_{k-1} + u_{PI,k-1}$

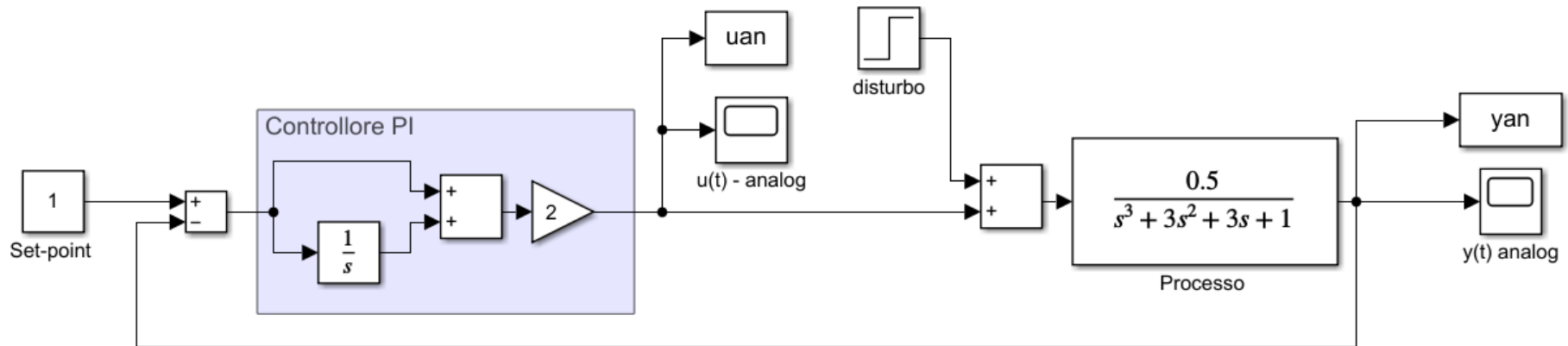
Scrivi $u_{PI,k}$ nel registro del convertitore D/A

$u_{PI,k-1} := u_{PI,k}$

$e_{k-1} := e_k$

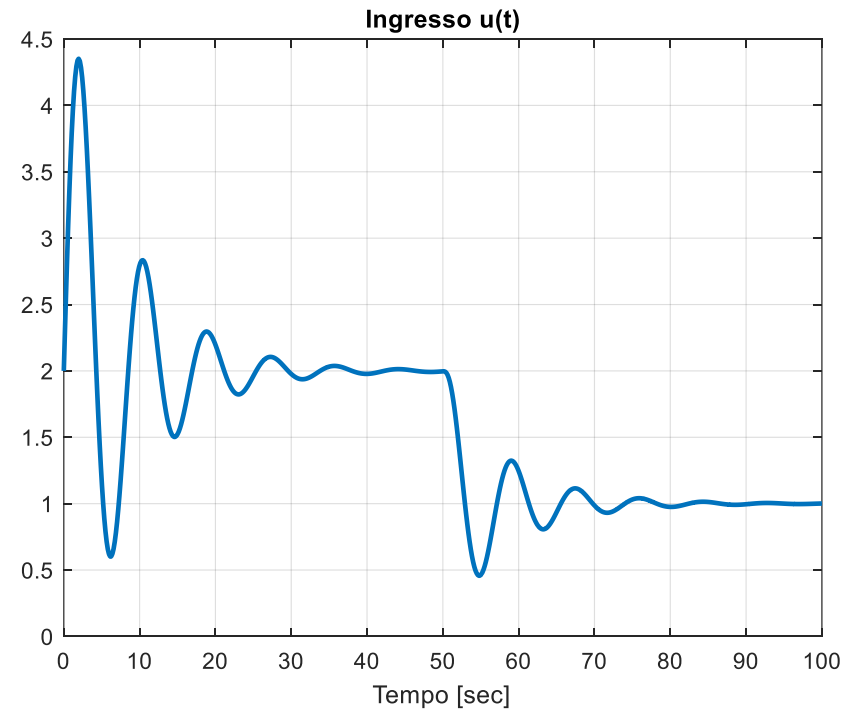
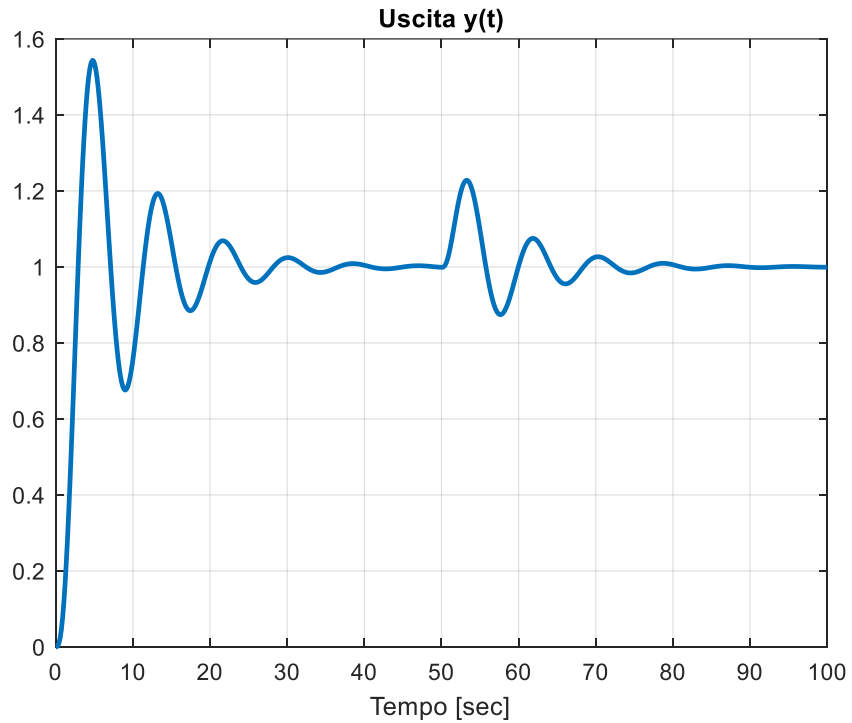
$k = k + 1$

Si realizzi dapprima il modello di simulazione standard relativo alla implementazione del controllore «a tempo continuo»



Si utilizzi un valore molto piccolo per il fixed-step size del solutore numerico ($T_{sim} = 0.0001$) per avere una fedele riproduzione delle dinamiche a tempo continuo sia del processo che del controllore analogico.

Sistema di controllo a tempo continuo

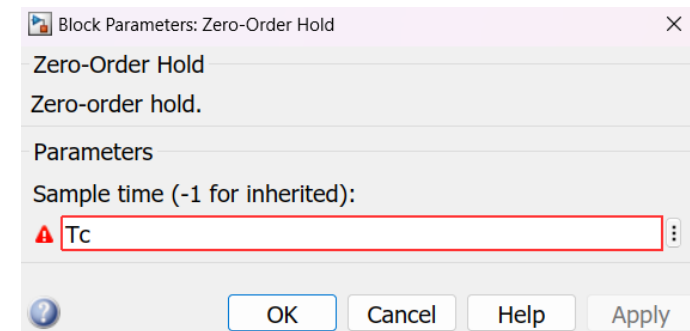
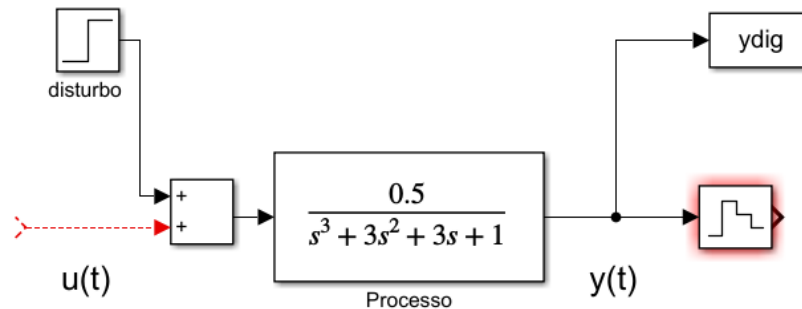


Ora passiamo a realizzare il modello di simulazione relativo alla realizzazione digitale del controllore.

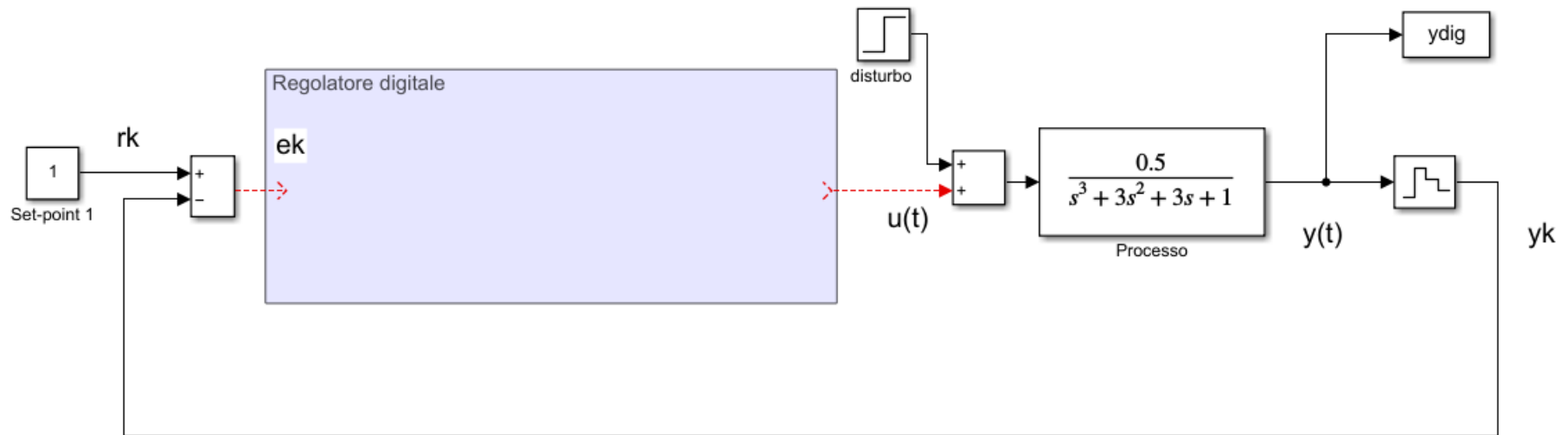
Sia T_c il periodo di campionamento del regolatore digitale

Inserire preliminarmente a valle del processo il blocco Simulink «Zero-Order hold».

Tale blocco svolge il ruolo del convertitore A/D



Inserire T_c come «Sample time»



Ora all'interno dell'area evidenziata realizzeremo il controllore digitale.

$$e_{-1} = u_{PI,-1} = 0$$

$$b_0 = K_P \left(1 + \frac{T_c}{T_I} \right)$$

$$u_{PI,k} = b_0 e_k - K_P e_{k-1} + u_{PI,k-1}$$

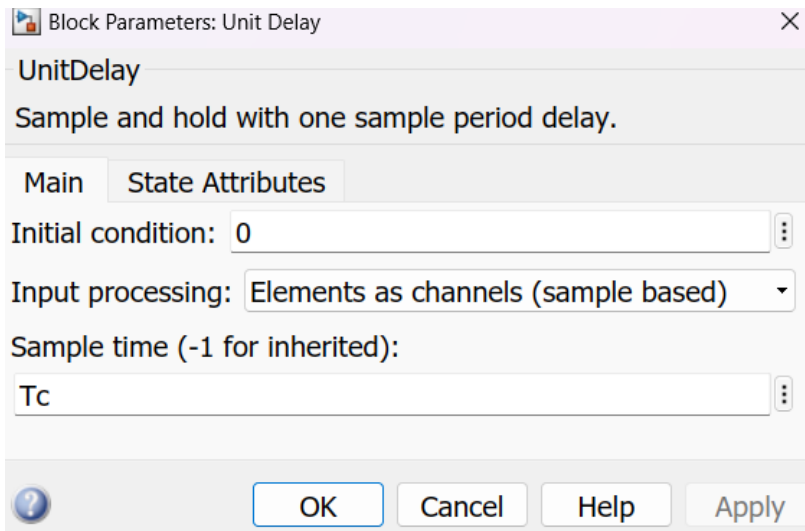
Definiamo all'interno di uno script i parametri T_{sim} , T_c , K_P , T_I e b_0

```
clear all, close all, clc

Tsim=0.0001;
Tc=0.1;

Kp=2;
Ti=1;
b0=Kp*(1+Tc/Ti)
```

Ora importiamo il blocco Simulink «Unit delay»

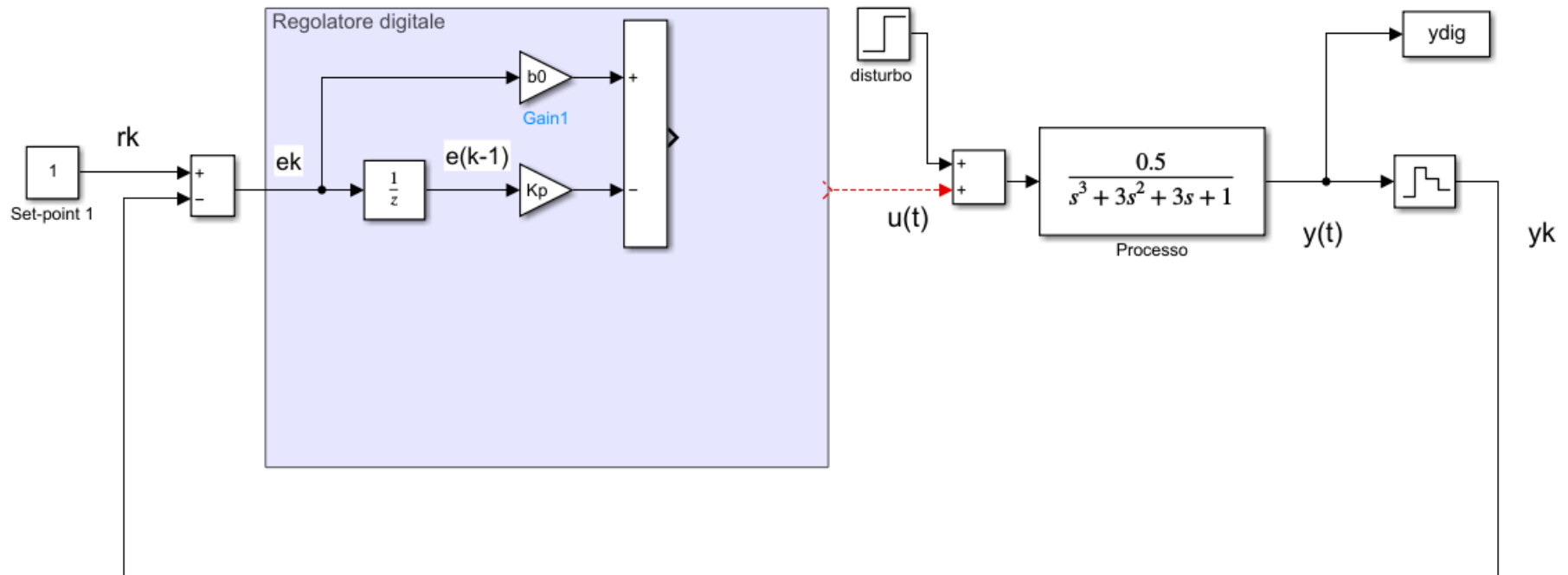


Parametrizzandolo come mostrato a sinistra, tale blocco produce, a partire da un segnale generico x_k , il segnale «ritardato» x_{k-1} inizializzandone a zero il valore.

Usiamo tale blocco per rendere disponibile nello schema il segnale e_{k-1}

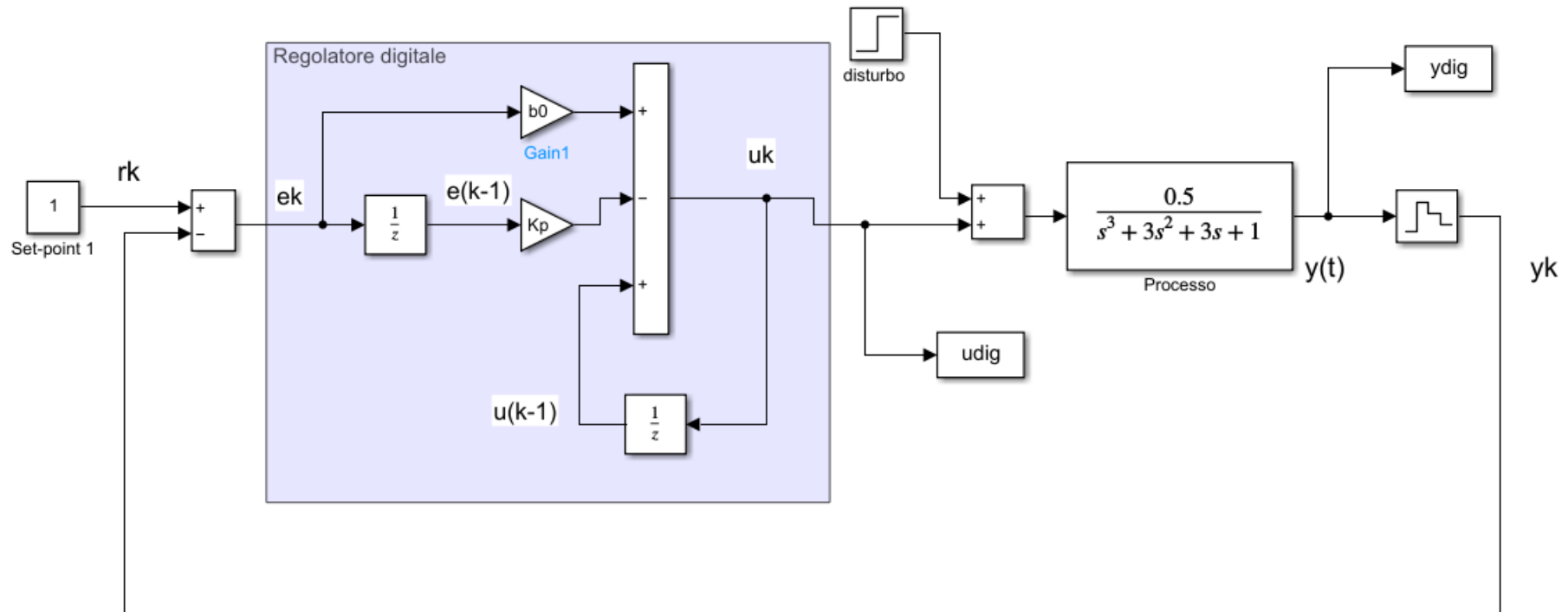
Possiamo ora realizzare, e sommare fra loro, le prime due aliquote della espressione della legge di controllo digitale

$$u_{PI,k} = b_0 e_k - K_P e_{k-1} + u_{PI,k-1}$$



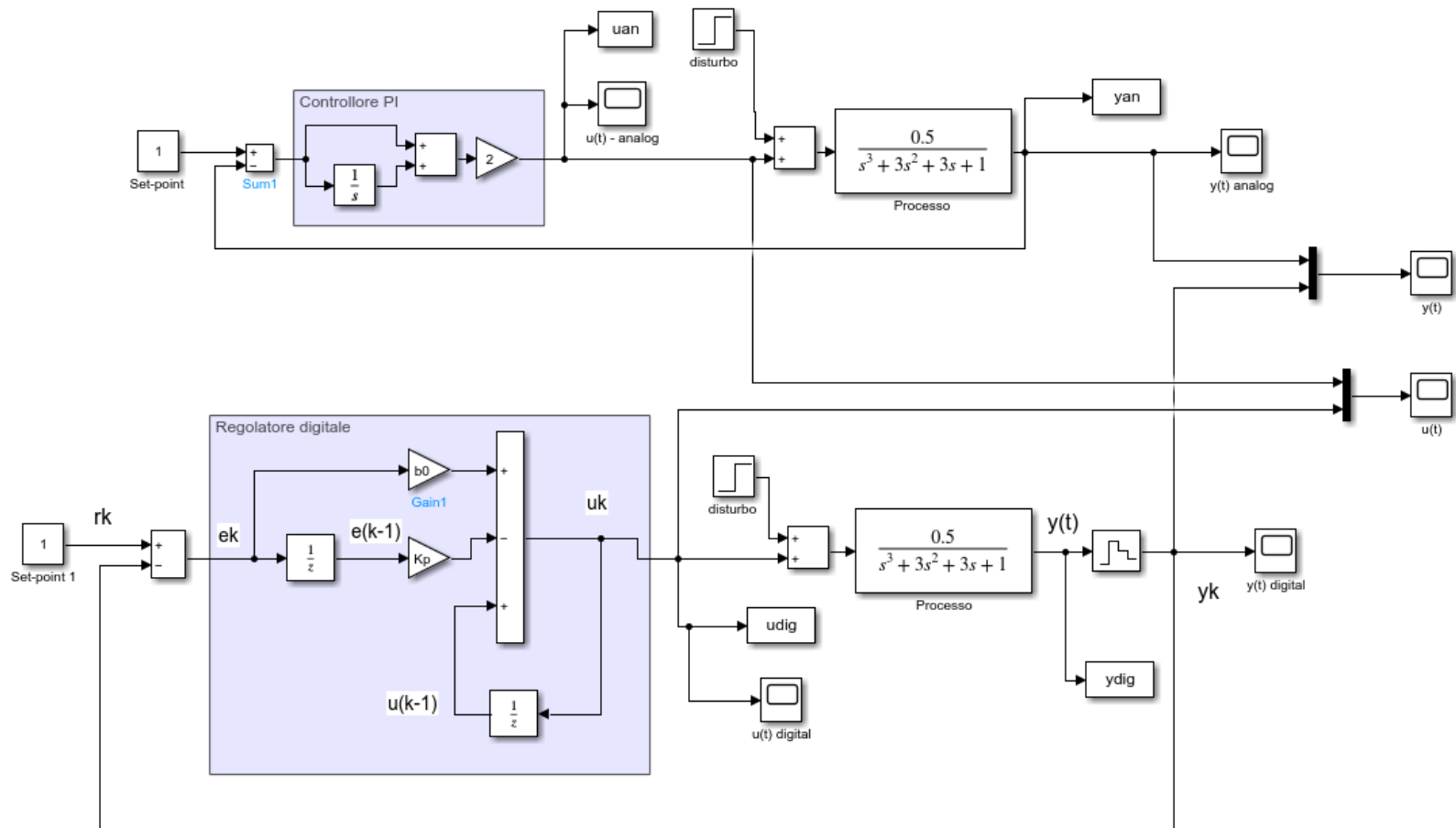
Per rendere accessibile la terza ed ultima aliquota $u_{PI,k-1}$ utilizziamo una ulteriore istanza del blocco Unit Delay

Il controllore digitale può essere completato come segue



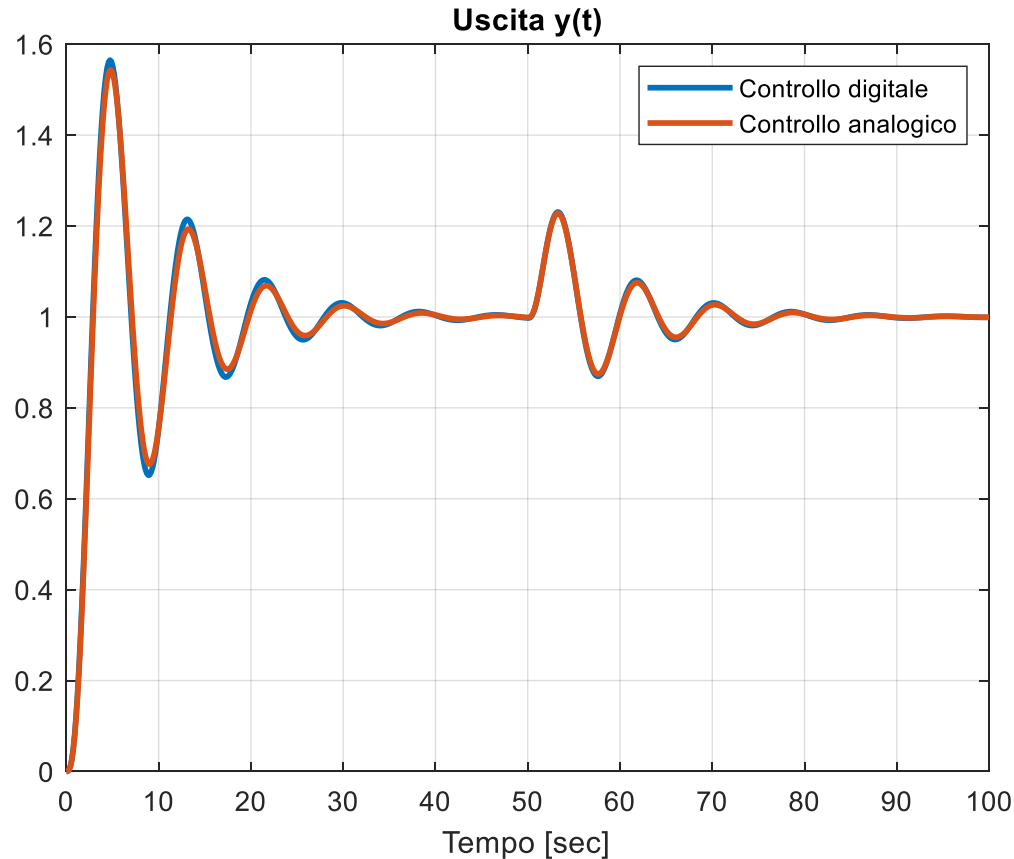
$$u_{PI,k} = b_0 e_k - K_P e_{k-1} + u_{PI,k-1}$$

Il modello Simulink complessivo pone uno affianco all'altro lo schema con il regolatore analogico e digitale, e consente un confronto prestazionale.



Files Es1_DigitalPI.slx
Es1_DigitalPI_script.m

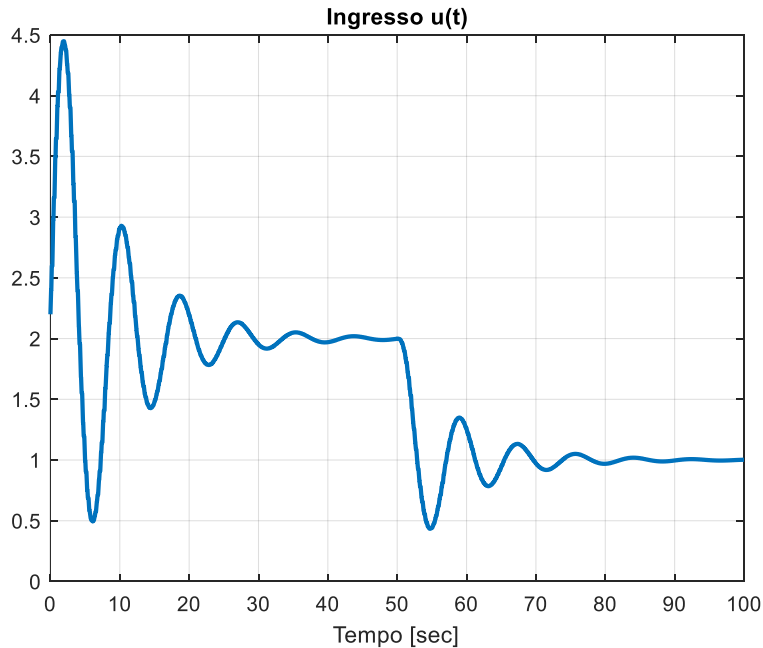
Sistema di controllo digitale con $T_c = 0.1 \text{ s}$



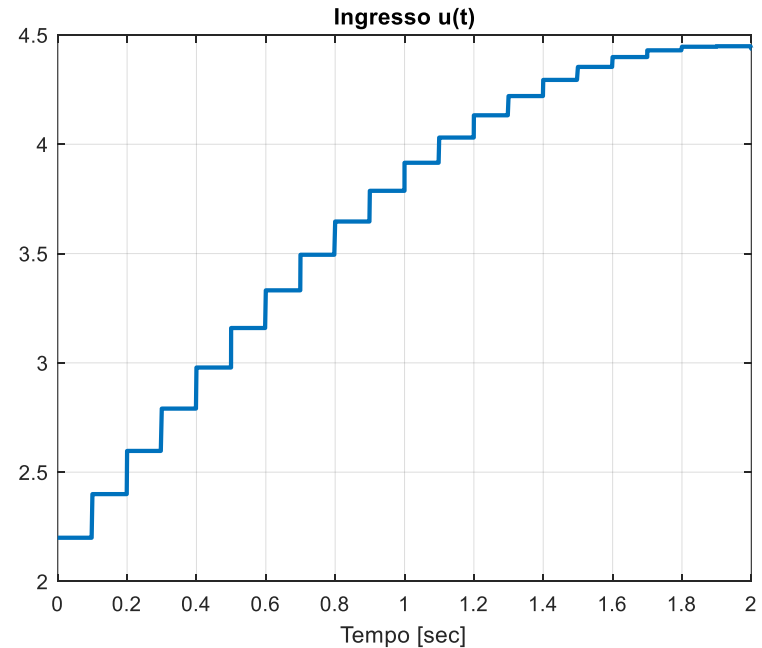
L'evoluzione temporale dell'uscita è pressoché indistinguibile da quella ottenuta impiegando il regolatore analogico

Sistema di controllo digitale con $T_c = 0.1 \text{ s}$

Ingresso



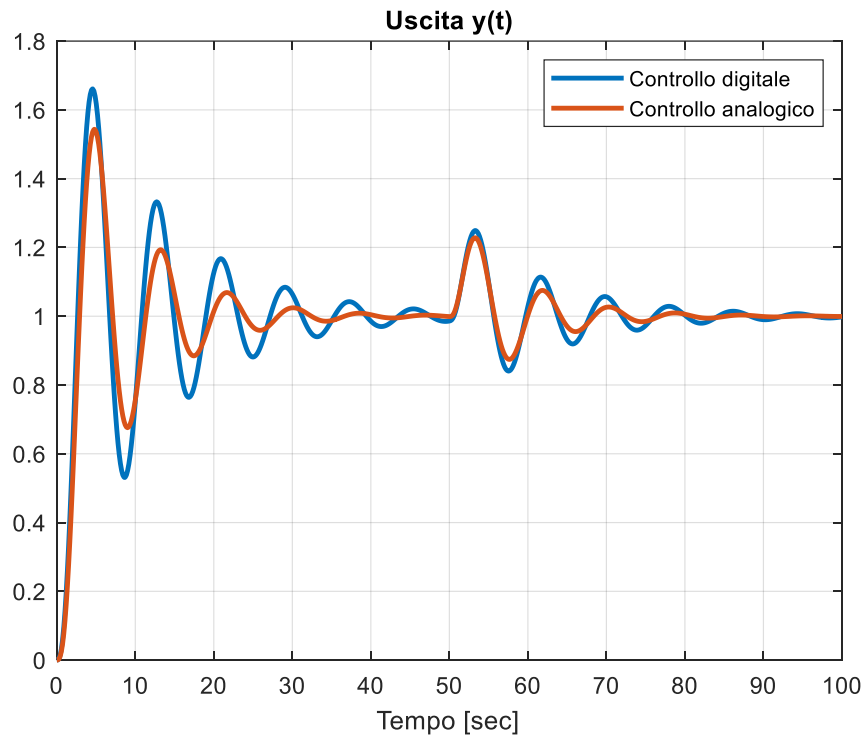
Ingresso (zoom)



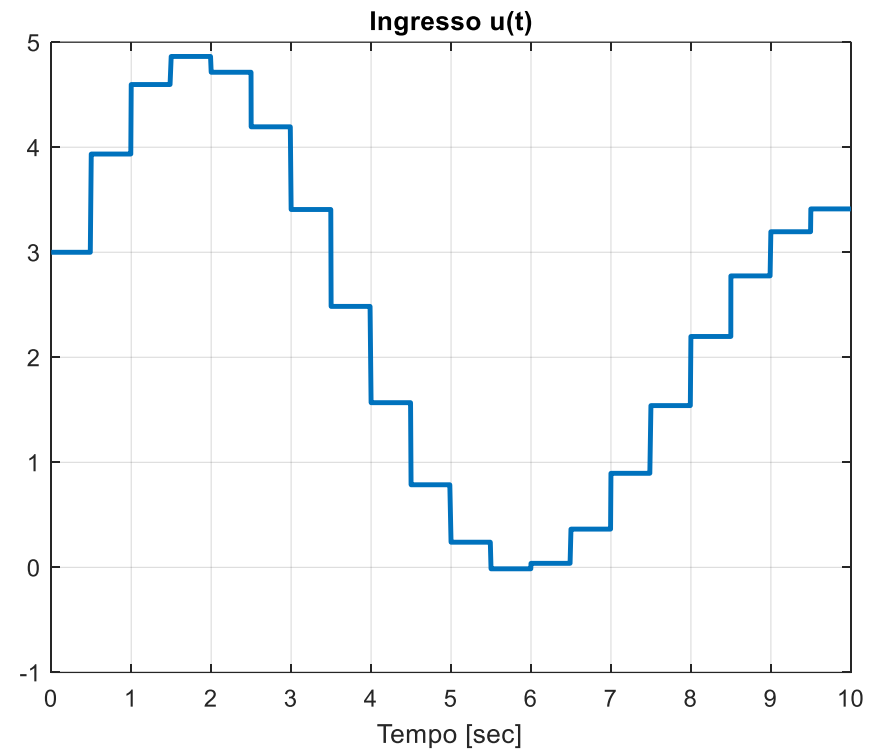
Lo zoom mostra come il segnale applicato in ingresso al processo sia costante a tratti, e venga aggiornato ogni decimo di secondo.

Sistema di controllo digitale con $T_c = 0.5$ s

Uscita

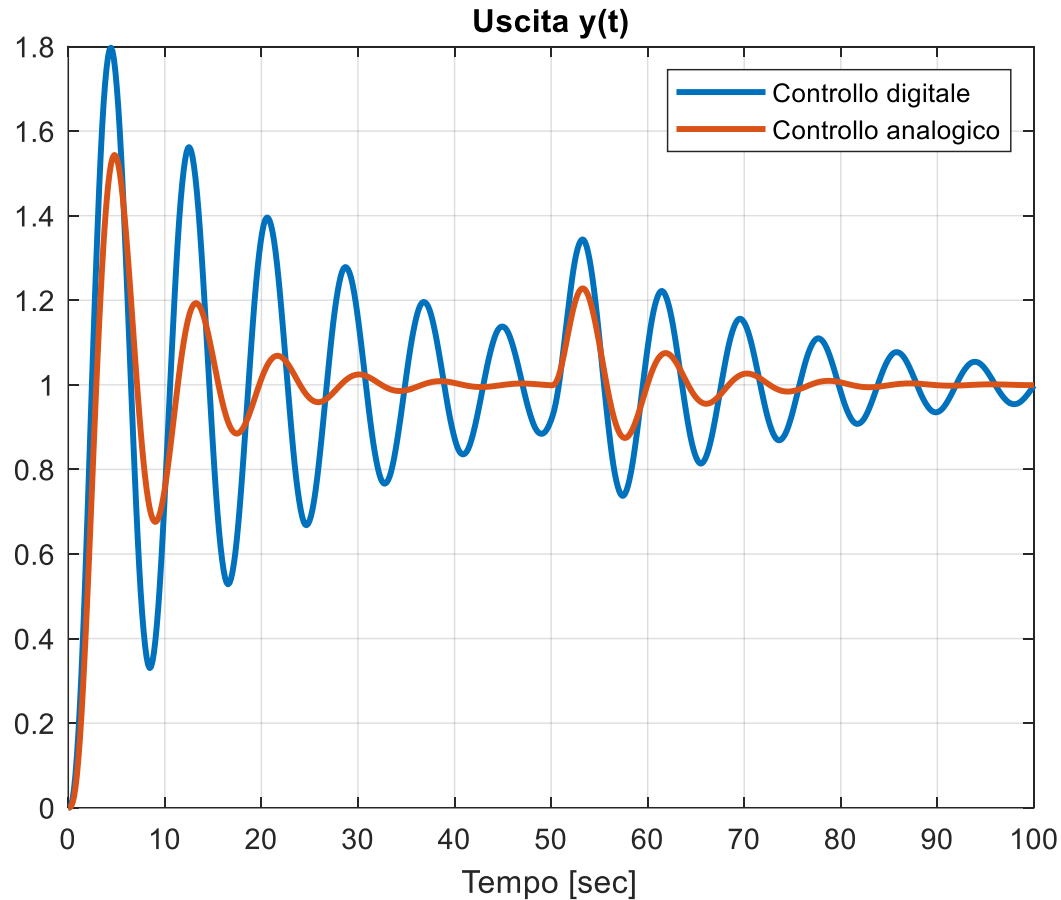


Ingresso (zoom)



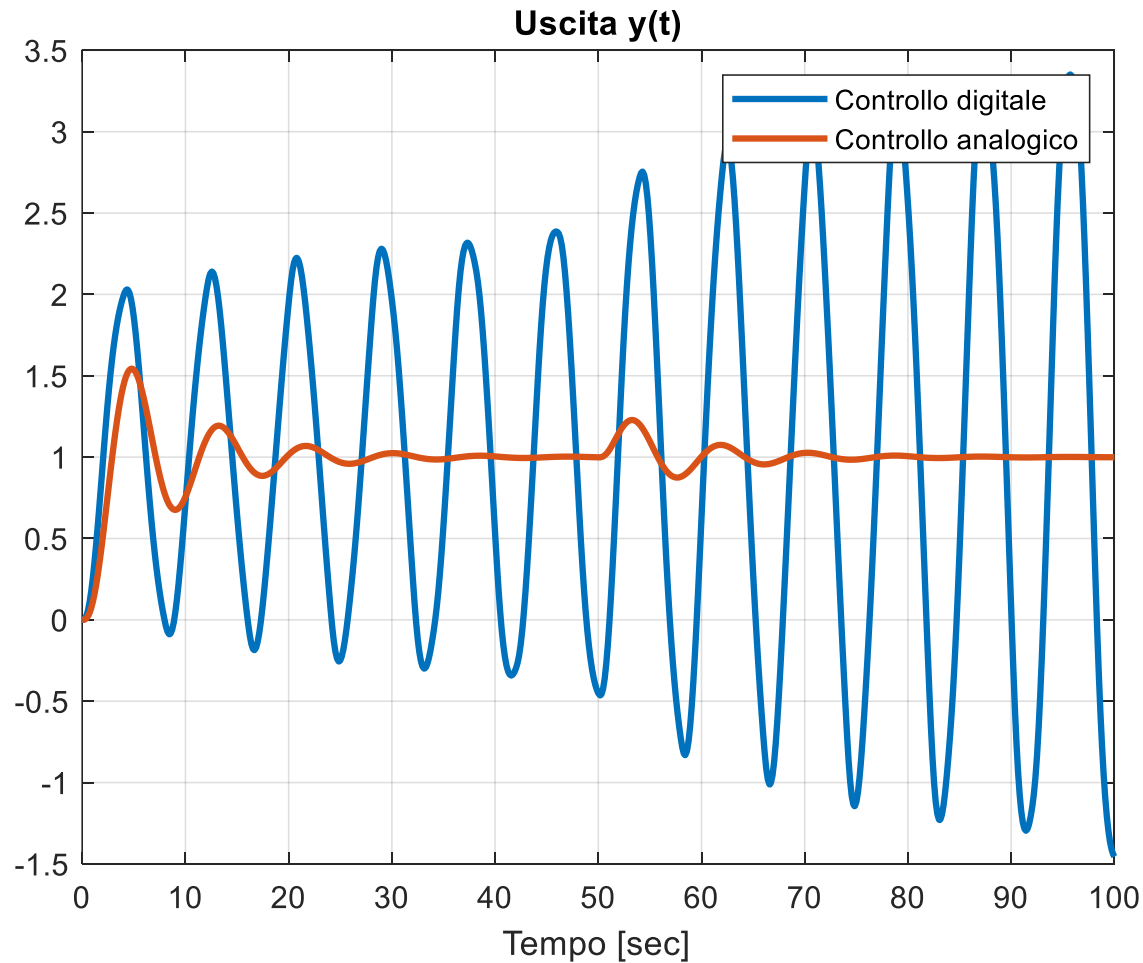
Iniziano a evidenziarsi dei deterioramenti prestazionali (maggiore sovraelongazione) dovuti all'impiego del regolatore digitale

Sistema di controllo digitale con $T_c = 1\text{ s}$



Il deterioramento prestazionale è decisamente più marcato

Sistema di controllo digitale con $T_c = 2\text{ s}$



Il sistema di controllo diventa instabile se $T_c = 2\text{ s}$

Realizziamo in maniera differente il controllore digitale mediante un blocco Simulink chiamato «**MATLAB function**» che consente di scrivere (ed eseguire man mano che procede il run del modello Simulink) un codice del tutto analogo al pseudo-codice di controllo visto in precedenza

Pseudo-codice di controllo

Inizializzazione: $k = 0$ $u_{PI,-1} = e_{-1} = 0$

Ogni T_c secondi:

Leggi y_k dal registro del convertitore A/D

Calcola $e_k = r_k - y_k$

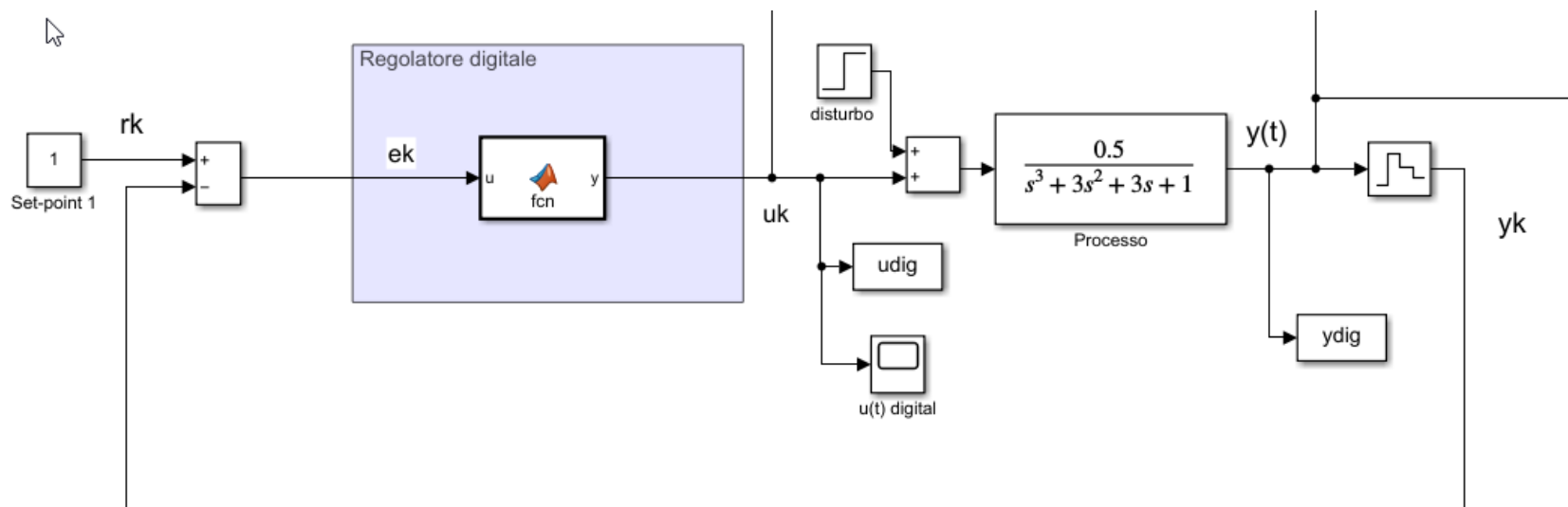
Calcola $u_{PI,k} = K_P \left(1 + \frac{T_c}{T_I} \right) e_k - K_P e_{k-1} + u_{PI,k-1}$

Scrivi $u_{PI,k}$ nel registro del convertitore D/A

$u_{PI,k-1} := u_{PI,k}$

$e_{k-1} := e_k$

$k = k + 1$



Il blocco «**MATLAB function**» permette di eseguire del codice matlab all'interno di un modello Simulink, con la facoltà di usare come dati di ingresso dei segnali generati internamente al modello Simulink stesso. Il segnale che passiamo in ingresso al blocco Matlab Function è la sequenza e_k

Facendo doppio click sul blocco MATLAB function si apre l'editor all'interno del quale scrivere il codice. La sintassi da impiegarsi è la medesima sintassi che si userebbe per realizzare un function file

Poiché un blocco «MATLAB function» **non può accedere alle variabili del workspace**, facciamo in modo che il valore del periodo di campionamento T_c sia passato al blocco funzione mediante uno specifico parametro di ingresso. I valori dei guadagni K_p , T_i del controllore PI saranno invece definiti all'interno del corpo della funzione, così come il guadagno b_0 .

```
function uk = fcn(ek, Tc)
```

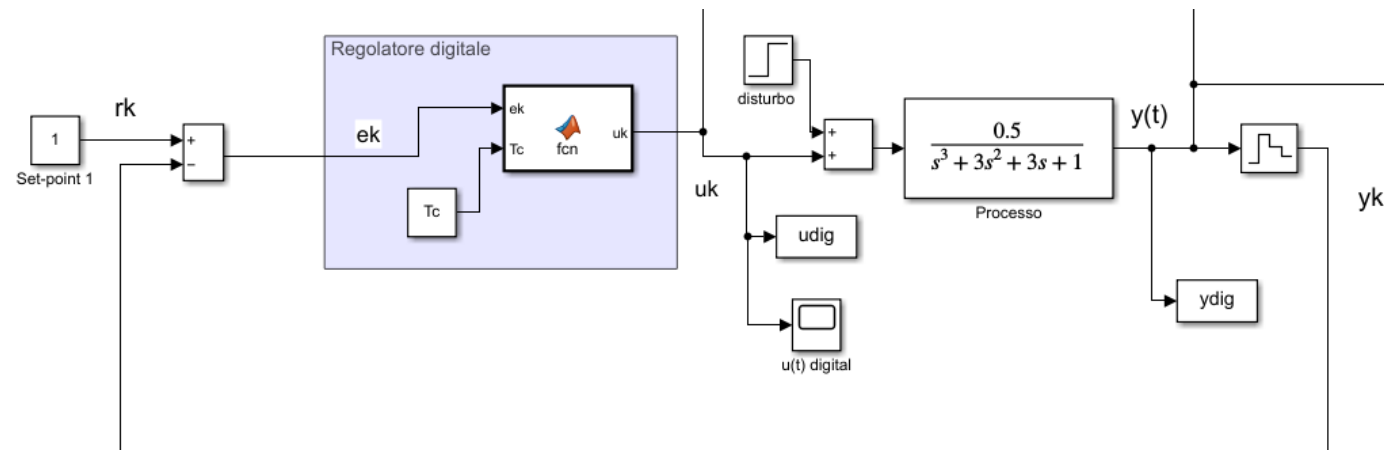
```
Kp=2; Ti=1;  
b0=Kp*(1+Tc/Ti);
```

```
uk=??????
```

L'aspetto del blocco funzione cambia, e possiamo passare in ingresso il valore di T_c come mostrato sotto.

Resta da completare il corpo della funzione in modo da implementare la legge di controllo digitale.

Il blocco funzione si «attiva» ogni T_c secondi.



Function che implementa il controllore associato al'esempio

```
function uk = fcn(ek,Tc)
```

```
persistent ukmeno1 ekmeno1
```

```
Kp=2; Ti=1;  
b0=Kp*(1+Tc/Ti);
```

Questa istruzione definisce come «persistenti» le variabili ukmeno1 ed ekmeno1, in modo che il loro valore venga mantenuto fra due successive attivazioni del blocco funzione

```
if isempty(ukmeno1)  
    ukmeno1 = 0;  
    ekmeno1 = 0;  
end
```

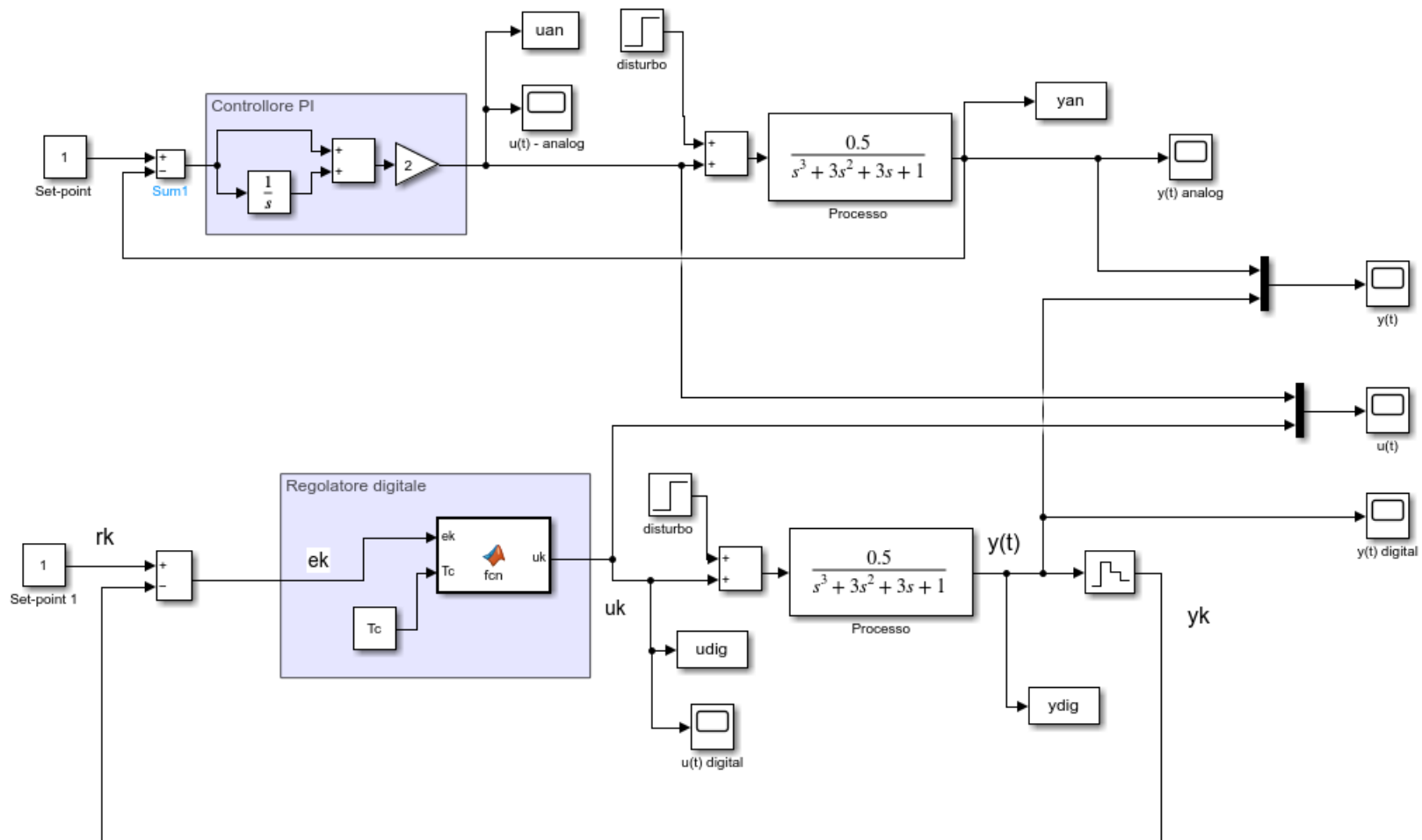
Questa istruzione viene eseguita solo alla prima attivazione del blocco, ed inizializza a zero i valori di ukmeno1 ed ekmeno1.

```
uk=b0*ek-Kp*ekmeno1+ukmeno1;
```

Calcolo della legge di controllo

```
ekmeno1=ek;  
ukmeno1=uk;
```

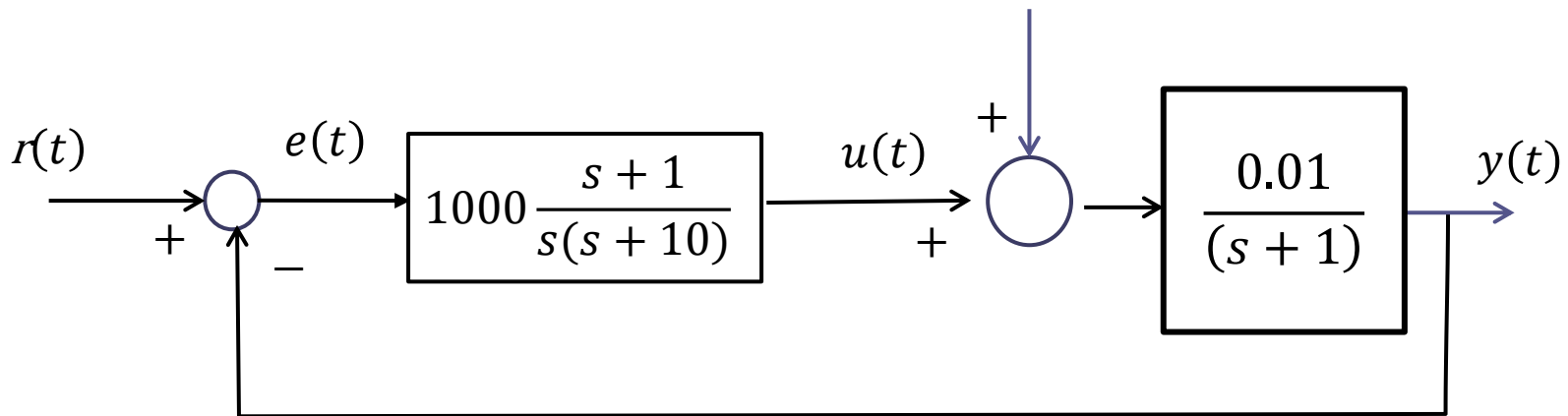
Questa coppia di istruzioni realizza un «registro a scorrimento» che memorizza i valori correnti di ek ed uk nelle variabili ukmeno1 ed ekmeno1 in modo che possano essere utilizzabili nella successiva attivazione del blocco funzione



Problema n. 2

Implementare per via numerica il seguente sistema di controllo e testarne le prestazioni in corrispondenza di diversi valori del periodo di campionamento

$$\xi(t) = 50 \delta_{-1}(t - 25)$$



Disturbo $\xi(t)$ di ampiezza 50 che interviene a $t = 25$

Metodo delle differenze finite

Un algoritmo numerico che risolva in via approssimata l'equazione differenziale

$$\ddot{u}(t) + 10 \dot{u}(t) = 1000 \dot{e}(t) + 1000 e(t)$$

associata al controllore è stato ricavato a lezione ed è il seguente:

$$u_k = \left(\frac{2 + 10T_c}{1 + 10T_c} \right) u_{k-1} - \left(\frac{1}{1 + 10T_c} \right) u_{k-2} + \left(\frac{1000T_c(1 + T_c)}{1 + 10T_c} \right) e_k - \frac{1000T_c}{1 + 10T_c} e_{k-1}$$

In forma compatta:

$$u_{-1} = u_{-2} = e_{-1} = 0$$

$$u_k = -a_1 u_{k-1} - a_2 u_{k-2} + b_0 e_k + b_1 e_{k-1} \quad k = 0, 1, 2, \dots$$

$$a_1 = -\frac{2 + 10T_c}{1 + 10T_c} \quad a_2 = \frac{1}{1 + 10T_c} \quad b_0 = \frac{1000T_c(1 + T_c)}{1 + 10T_c} \quad b_1 = -\frac{1000T_c}{1 + 10T_c}$$

Passaggi

$$\ddot{u}(kT_c) + 10 \dot{u}(kT_c) = 1000 \dot{e}(kT_c) + 1000 e(kT_c)$$



$$\frac{u_k - 2u_{k-1} + u_{k-2}}{T_c^2} + 10 \frac{u_k - u_{k-1}}{T_c} = 1000 \frac{e_k - e_{k-1}}{T_c} + 1000 e_k$$



$$u_k \left(\frac{1}{T_c^2} + \frac{10}{T_c} \right) = \left(\frac{2}{T_c^2} + \frac{10}{T_c} \right) u_{k-1} - \frac{1}{T_c^2} u_{k-2} + \left(\frac{1000}{T_c} + 1000 \right) e_k - \frac{1000}{T_c} e_{k-1}$$



$$u_k = \left(\frac{2 + 10T_c}{1 + 10T_c} \right) u_{k-1} - \left(\frac{1}{1 + 10T_c} \right) u_{k-2} + \left(\frac{1000T_c(1 + T_c)}{1 + 10T_c} \right) e_k - \frac{1000T_c}{1 + 10T_c} e_{k-1}$$

Testiamo le prestazioni di questo sistema di controllo mediante simulazione dinamica

Prendiamo spunto dal modello `Es1_DigitalPI_function.slx` precedentemente sviluppato.

Modifichiamo ovviamente il processo, la realizzazione analogica del controllore, ed il corpo del blocco MATLAB function.

```
function uk = fcn(ek,Tc)

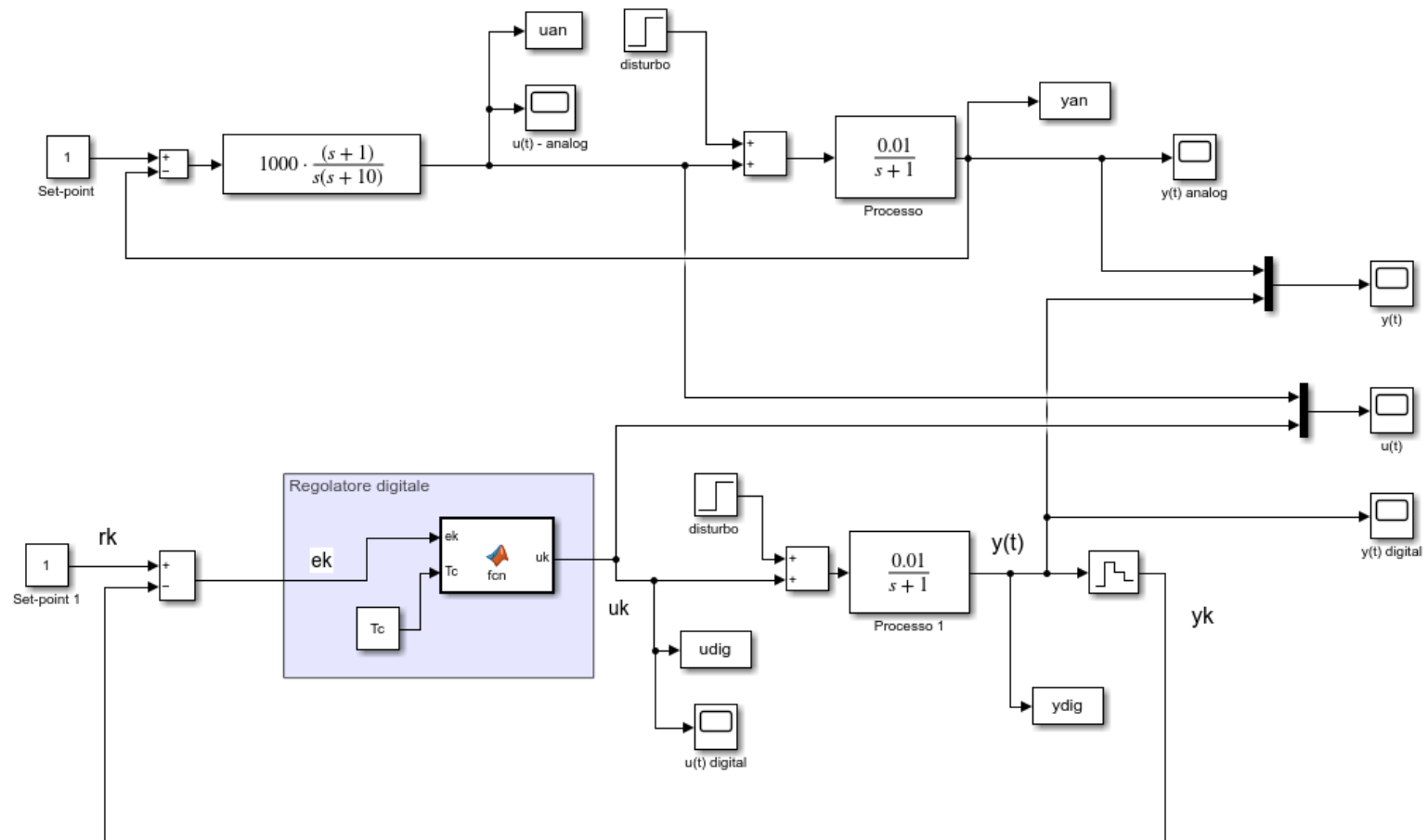
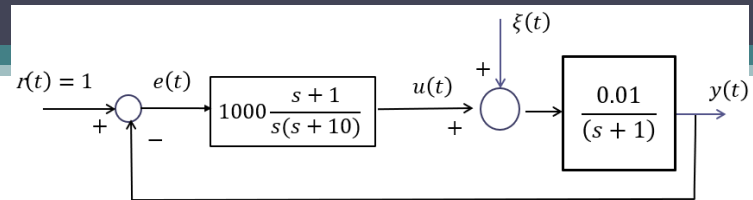
persistent ukmenol ukmeno2 ekmenol

a1=-(2+10*Tc)/(1+10*Tc);
a2=1/(1+10*Tc);
b0=1000*Tc*(1+Tc)/(1+10*Tc);
b1=-1000*Tc/(1+10*Tc);

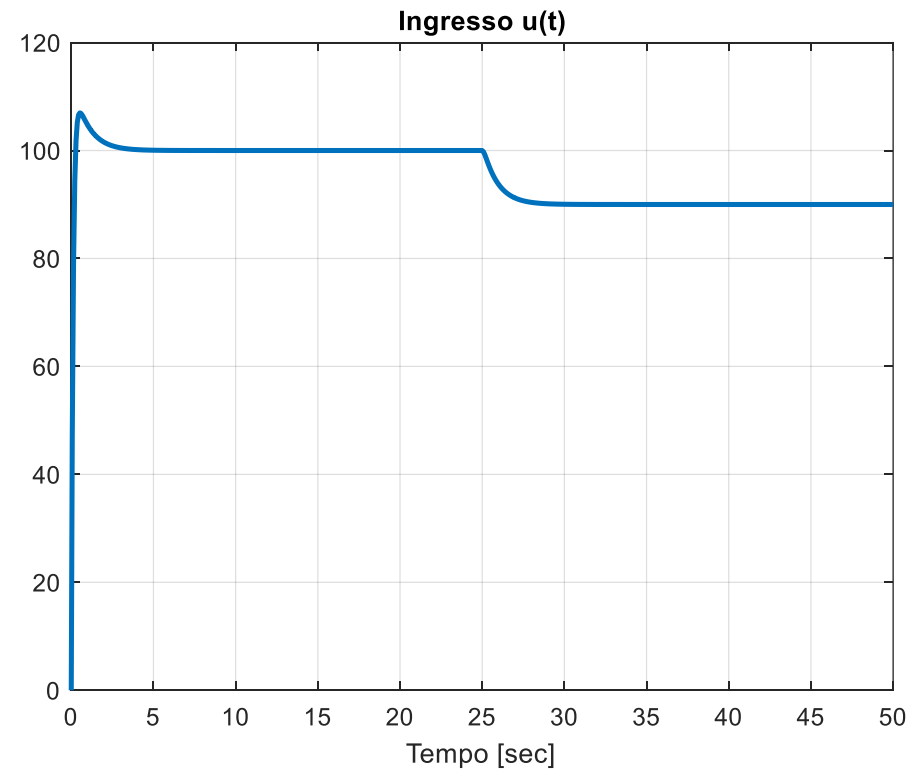
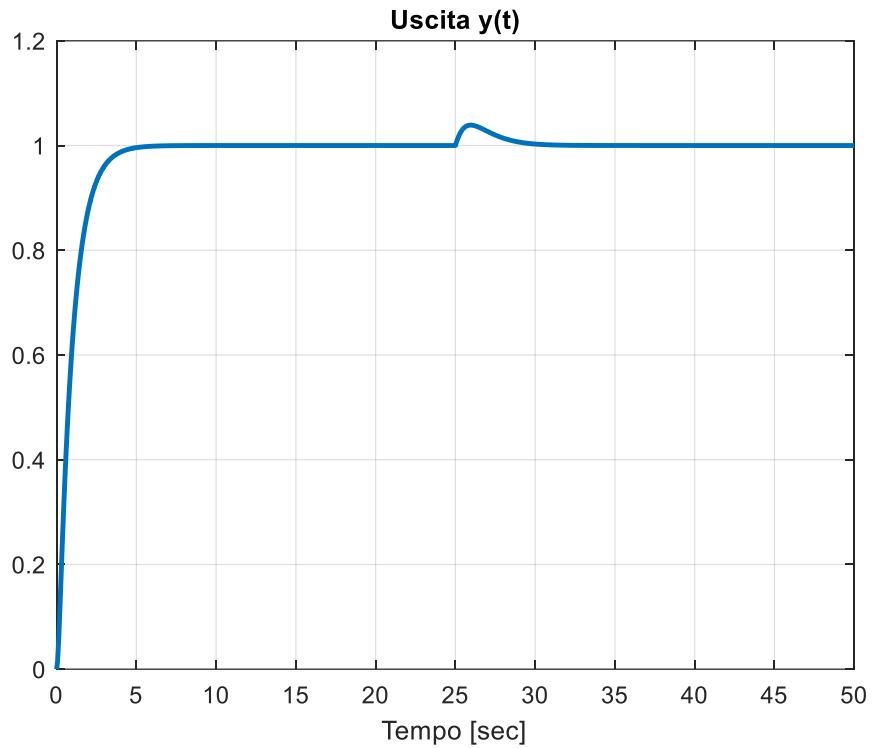
if isempty(ukmenol)
    ukmenol = 0;
    ukmeno2 = 0;
    ekmenol = 0;
end

uk=-a1*ukmenol-a2*ukmeno2+b0*ek+b1*ekmenol;

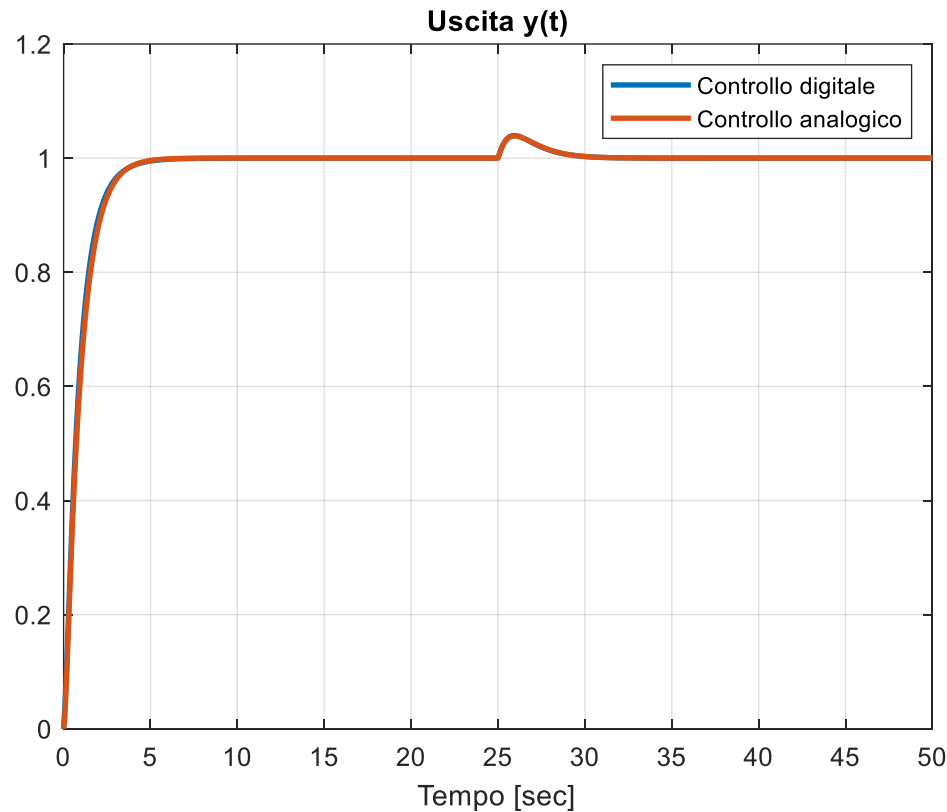
ekmenol=ek;
ukmeno2=ukmenol;
ukmenol=uk;
```



Sistema di controllo a tempo continuo



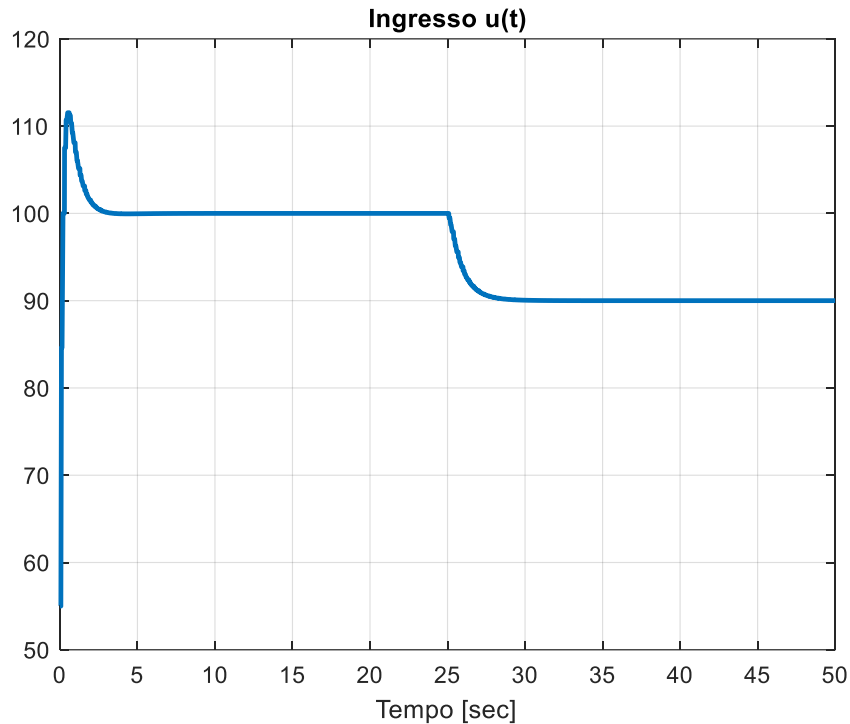
Sistema di controllo digitale con $T_c = 0.1 \text{ s}$



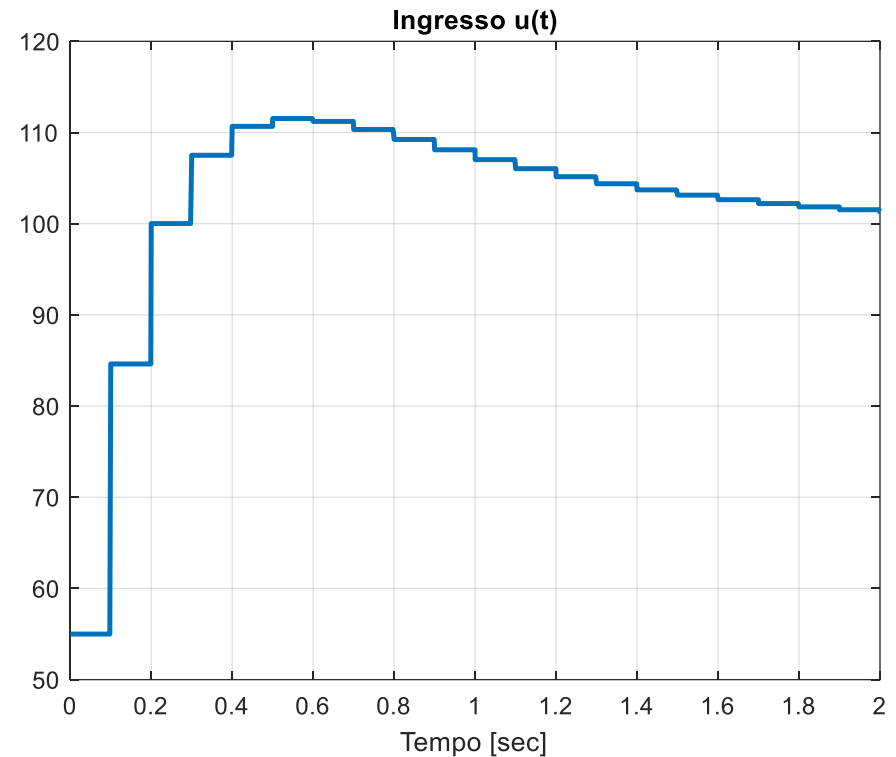
L'evoluzione temporale dell'uscita è indistinguibile da quella ottenuta impiegando il regolatore analogico

Sistema di controllo digitale con $T_c = 0.1 \text{ s}$

Ingresso



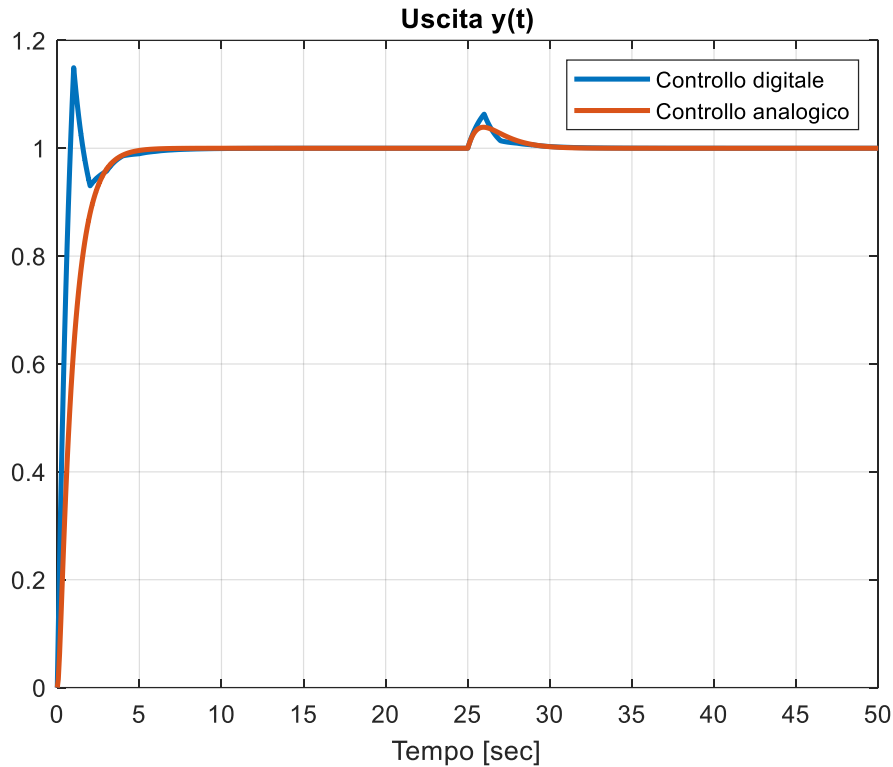
Ingresso (zoom)



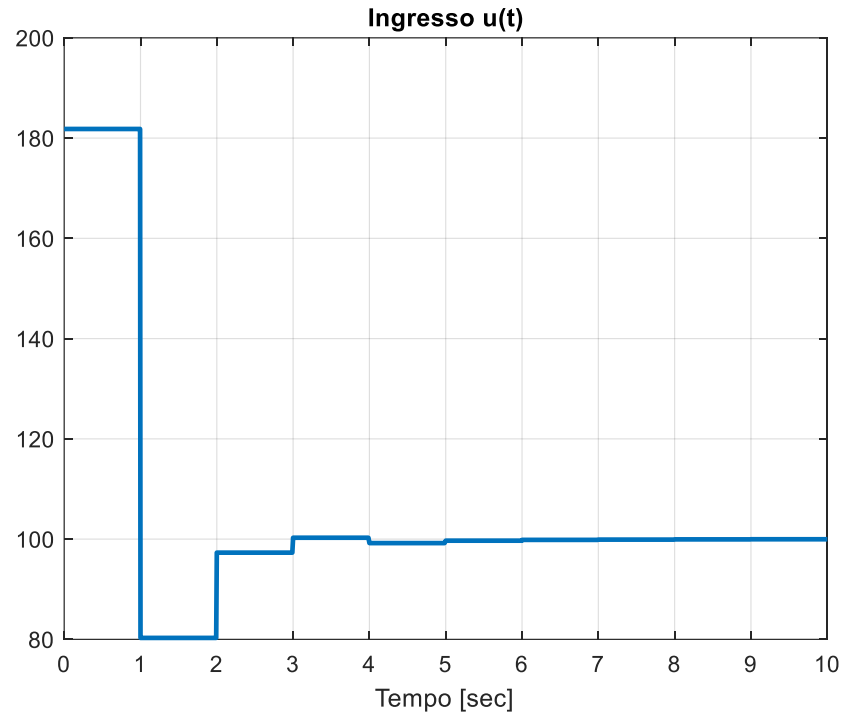
Lo zoom mostra come il segnale applicato in ingresso al processo sia costante a tratti, e venga aggiornato ogni decimo di secondo.

Sistema di controllo digitale con $T_c = 1\text{ s}$

Uscita



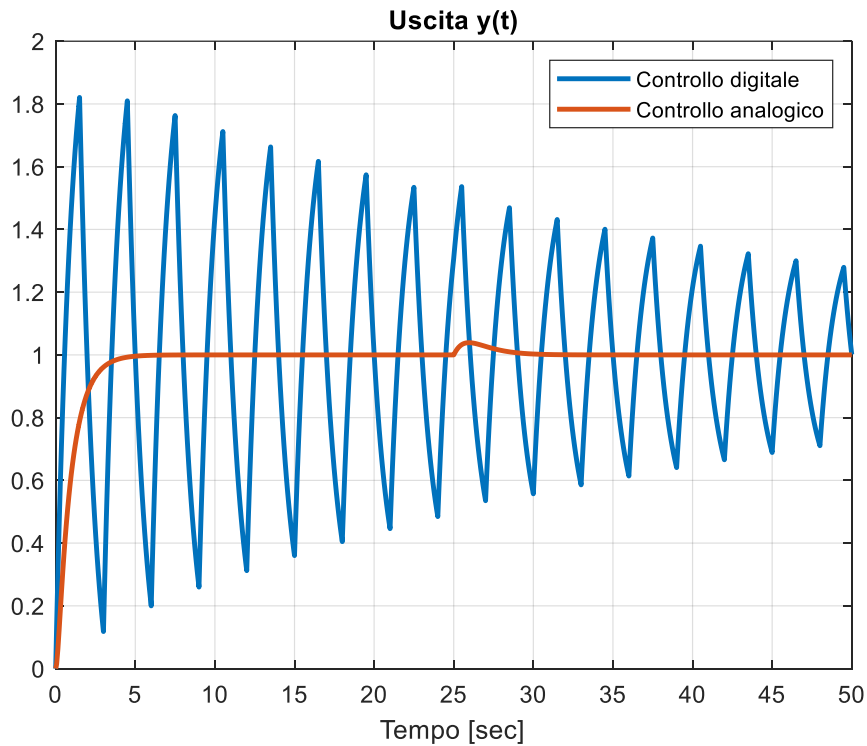
Ingresso (zoom)



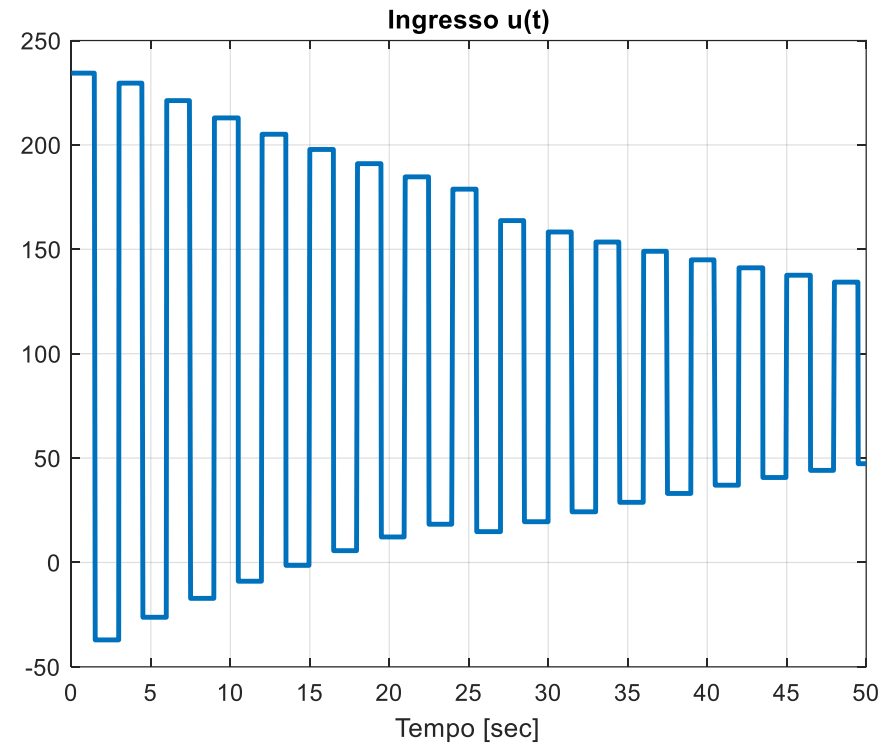
Iniziano a evidenziarsi dei deterioramenti prestazionali (comparsa di sovraelongazione) dovuti all'impiego del regolatore digitale

Sistema di controllo digitale con $T_c = 1.5$ s

Uscita

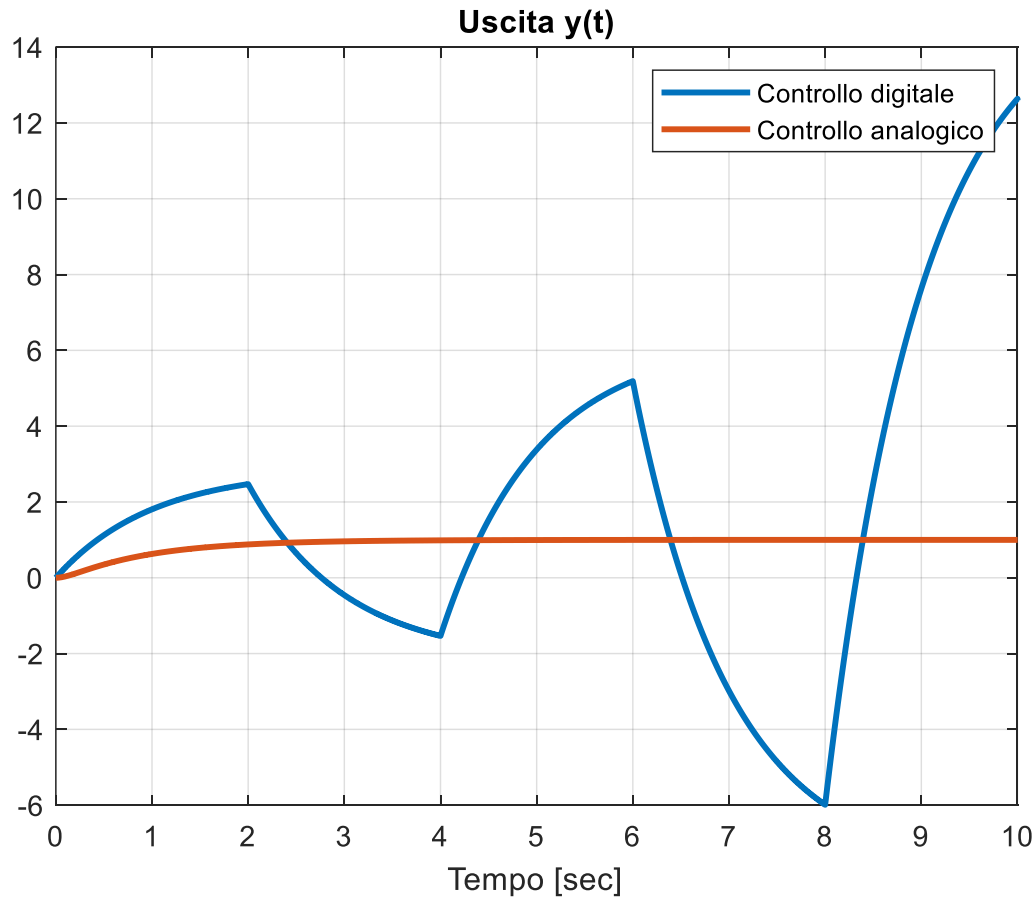


Ingresso



Il deterioramento prestazionale è nettamente più marcato

Sistema di controllo digitale con $T_c = 2\text{ s}$



Il sistema di controllo diventa instabile se $T_c = 2\text{ s}$

Metodo di «Tustin»

Partendo dalla FdT $R(s)$ di ordine n del controllore, si opera la seguente **sostituzione**

$$s = \frac{2}{T_c} \frac{z - 1}{z + 1}$$

ottenendo, a partire dal rapporto di polinomi in s , un rapporto di polinomi in z nella forma:

$$R_{TU}(z) = R(s) \Big|_{s=\frac{2}{T_c} \frac{z-1}{z+1}} = \frac{b_0 z^n + b_1 z^{n-1} + \dots b_{n-1} z + b_n}{z^n + a_1 z^{n-1} + \dots a_{n-1} z + a_n}$$

L'espressione ricorsiva che implementa digitalmente il controllore $R(s)$ si ricava sulla base dei coefficienti a numeratore e denominatore di $R_{TU}(z)$ nella forma seguente:

$$u_k = b_0 e_k + b_1 e_{k-1} + \dots + b_n e_{k-n} - a_1 u_{k-1} - a_2 u_{k-2} - \dots - a_n u_{k-n}$$

Discretizzazione secondo Tustin del controllore $R(s) = 1000 \frac{s+1}{s(s+10)}$

$$R(s) = 1000 \frac{s+1}{s(s+10)} \quad T_c = 1s \quad \text{Sostituzione} \quad s := \frac{2}{T_c} \frac{z-1}{z+1} = 2 \frac{z-1}{z+1}$$

$$R_{TU}(z) = R(s) \Big|_{s=2 \frac{z-1}{z+1}} = 1000 \frac{2 \frac{z-1}{z+1} + 1}{2 \frac{z-1}{z+1} (2 \frac{z-1}{z+1} + 10)} = \frac{125z^2 + 83.34z - 41.66}{z^2 - 0.333z - 0.6667}$$

Implementazione digitale del controllore

$$u_k = 125e_k + 83.34e_{k-1} - 41.66e_{k-2} + 0.333u_{k-1} + 0.6667u_{k-2}$$

Modificare il modello `Es2_Digital_DiffFinite_function` adottando la discretizzazione del controllore secondo il Metodo di Tustin