

Controlli automatici

Esercitazione n. 1

Introduzione alla simulazione di sistemi di controllo
in ambiente Matlab-Simulink

Cosa vedremo:

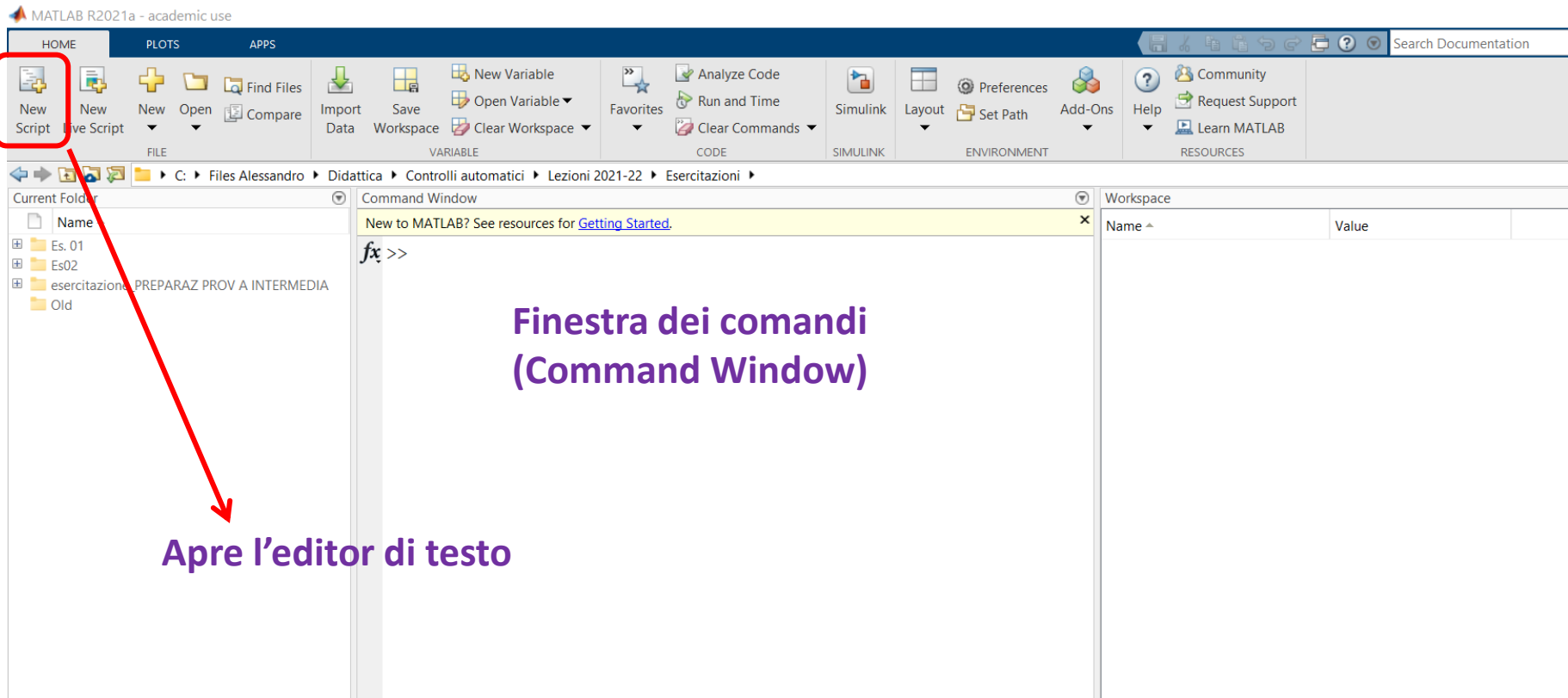
Analisi e simulazione dinamica in ambiente Matlab

Analisi e simulazione dinamica in ambiente Simulink

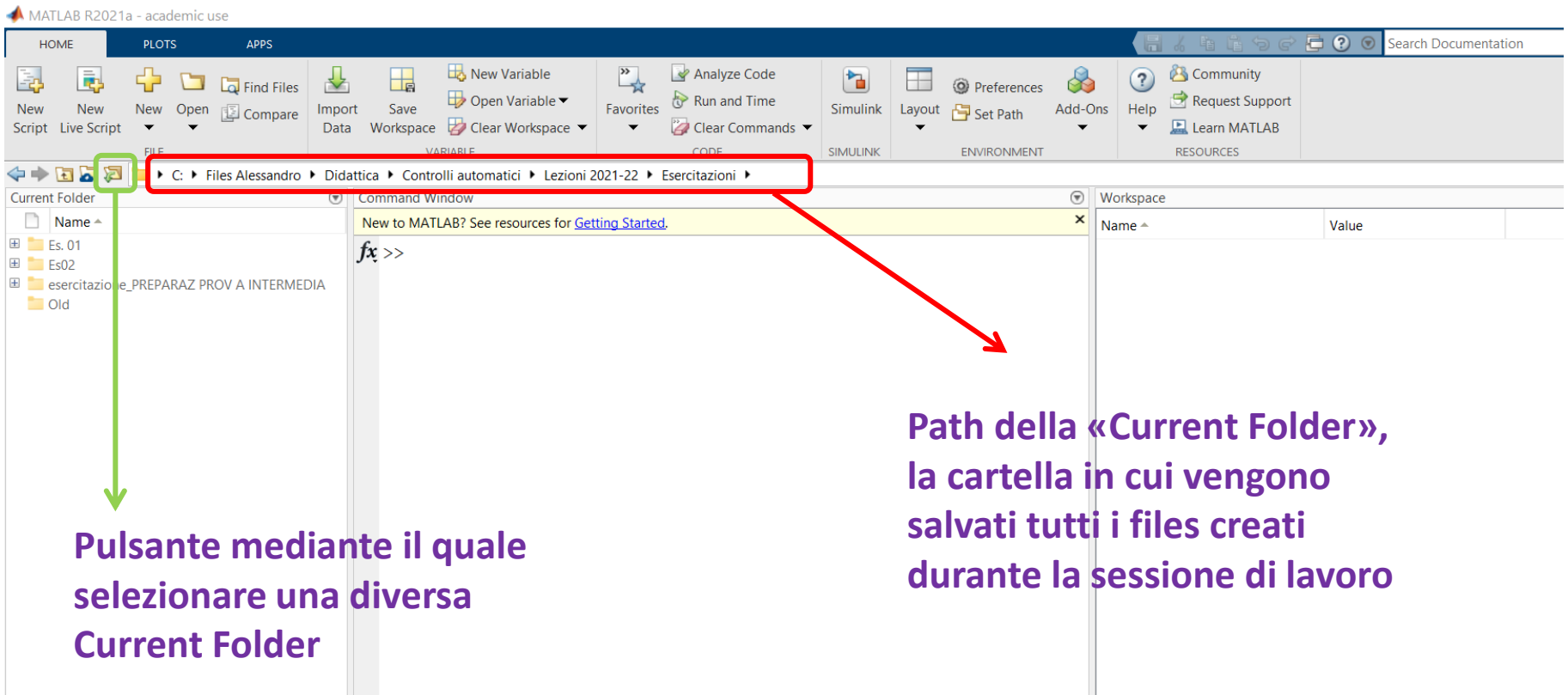
Simulazione dinamica di sistemi LTI a ciclo aperto

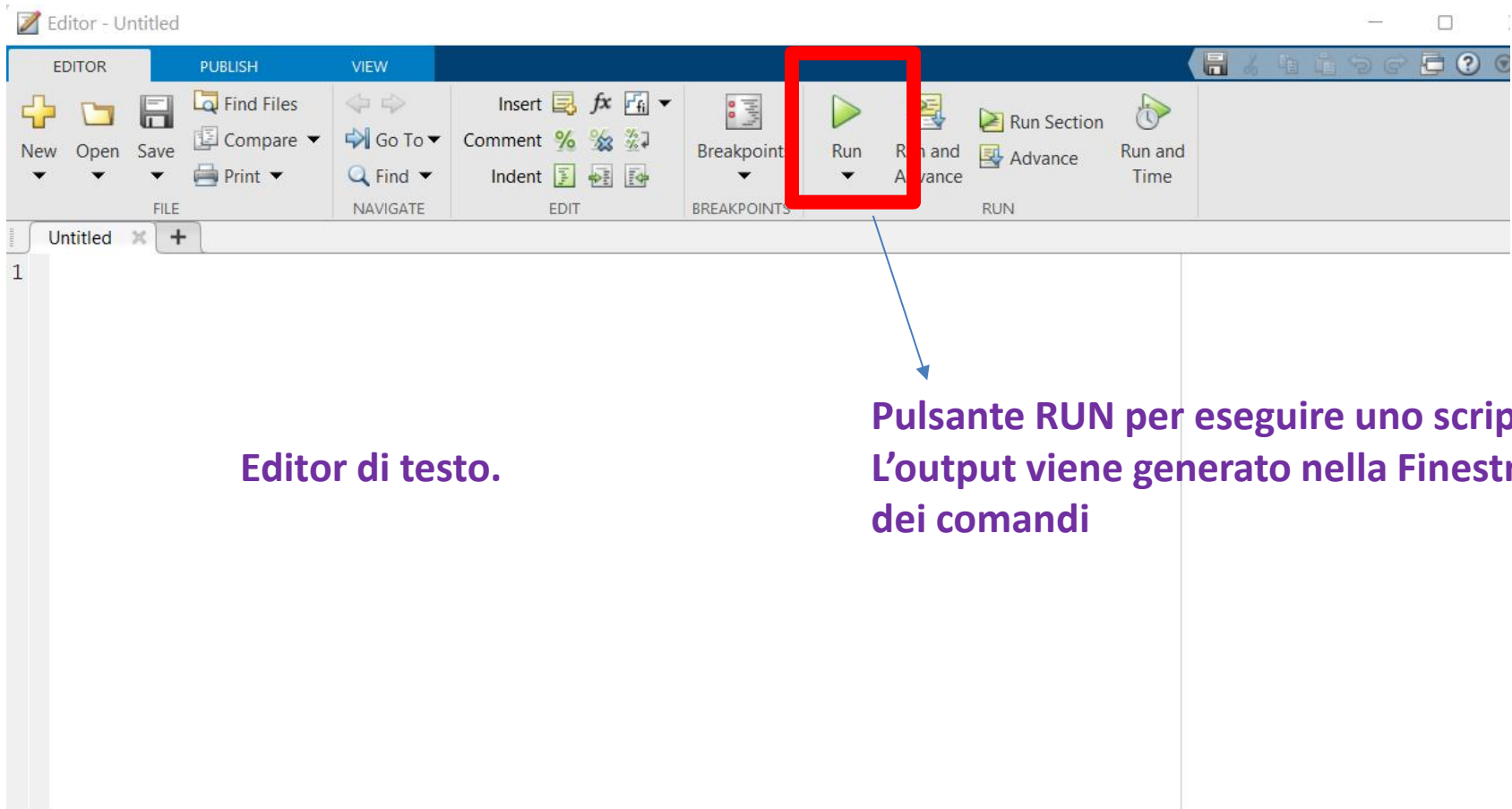
Simulazione dinamica di sistemi di controllo LTI in retroazione

Interfaccia di Matlab



La «current folder» è la cartella del PC in cui vengono salvati tutti i files creati nella sessione di lavoro corrente. E' opportuno sceglierla non appena si apre Matlab e si inizia a lavorare alla creazione di uno Script (uno Script è un file con estensione .m che contiene un programma scritto in linguaggio Matlab) o di un modello Simulink





Editor di testo.

Utilizzeremo questa interfaccia per scrivere i nostri programmi Matlab (estensione .m, detti «script») mediante i quali effettuare determinate tipologie di analisi, ad esempio visualizzare il LdR

Analisi polinomiale

Rappresentazione di polinomi per mezzo di vettori

Un polinomio di grado n è rappresentato da un vettore di dimensione $n+1$ che contiene tutti i coefficienti del polinomio (anche quelli eventualmente nulli) da quello del termine di grado più elevato fino a giungere al termine noto (ultima componente del vettore)

$$p1(x) = 2x^4 + 7x^3 + x^2 - 4x + 2$$

$$vp1 = [2 \quad 7 \quad 1 \quad -4 \quad 2]$$

$$p2(x) = 3x^3 + 1$$

$$vp2 = [3 \quad 0 \quad 0 \quad 1]$$

Radici di un polinomio

Funzione “roots”

```
clear all ← Cancella tutte le variabili preesistenti dal workspace
clc ← Resetta («ripulisce») la finestra dei comandi
p1=[2 7 1 -4 2]; ← Definizione di un vettore
p2=[3 0 0 1];
rad_p1=roots(p1) ← Gli argomenti di ingresso ad una funzione si
passano fra parentesi tonde
```

```
rad_p1 =
-3.0962
-1.2285
0.4124 + 0.3047i
0.4124 - 0.3047i
```

Prodotto tra polinomi

Funzione “conv”

```
clear all
clc
p1=[2 7 1 -4 2];
p2=[3 0 0 1];
prod_p1p2=conv(p1,p2)
```



```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

prod_p1p2 =
    6    21     3   -10    13     1    -4     2

fx >> |
```



$$P(x) = 6x^7 + 21x^6 + 3x^5 - 10x^4 + 13x^3 + x^2 - 4x + 2$$

Polinomio di radici assegnate (Funzione “Poly”)

$$p3 = poly([z1 \ z2 \ \dots \ zn]) \quad \longrightarrow \quad p3(x) = (x - z1)(x - z2) \dots (x - zn)$$

```
clear all
clc
p3=poly([1 2])
```



```
p3 =
    1    -3     2

fx >> |
```

Costruzione della risposta al gradino unitario mediante istruzioni Matlab

Funzioni “**step**” e “**tf**”

Creiamo un grafico della risposta al gradino unitario del sistema dinamico avente come FdT

$$F(s) = \frac{10(s + 1)}{(s + 3)(s + 4)(s + 6)}$$

Deve dapprima essere **definita una variabile che rappresenti il sistema dinamico** (funzione «tf»)

Primo modo:

```
clear all, clc  
  
num=10*[1 1];  
den=poly([-3 -4 -6]);  
F=tf(num,den)
```

Secondo modo:

```
clear all, clc  
  
s=tf('s')  
F=10*(s+1)/((s+3)*(s+4)*(s+6))
```



step(F)

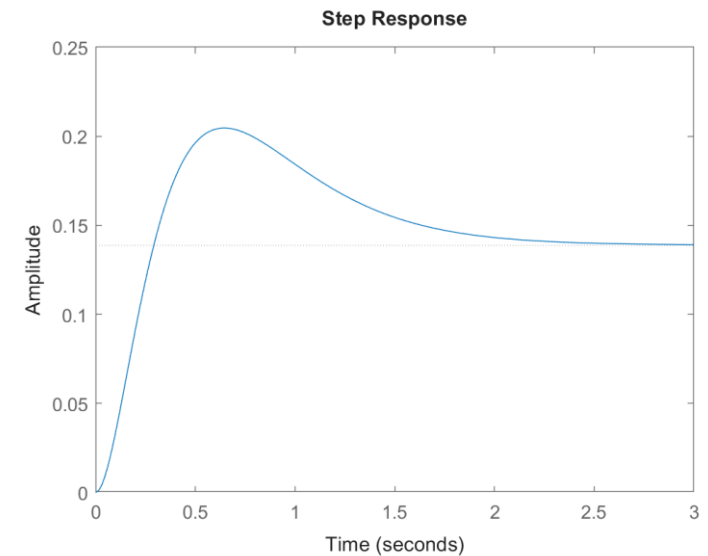
Visualizzazione
unitario. risposta al gradino


```
clear all, clc
```

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

```
F=10*(s+1)/((s+3)*(s+4)*(s+6))
```

```
step(F)
```



Per scegliere la **durata temporale** della simulazione:

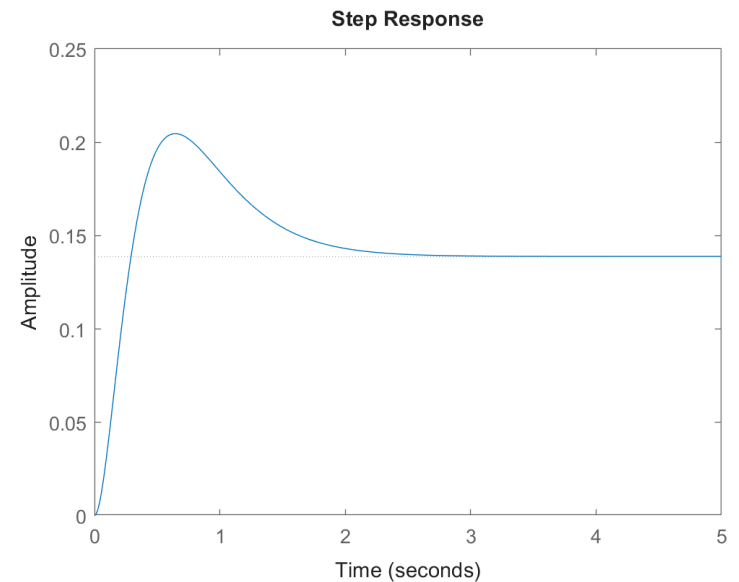
```
step(F, Tfinal)
```

```
clear all, clc
```

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

```
F=10*(s+1)/((s+3)*(s+4)*(s+6))
```

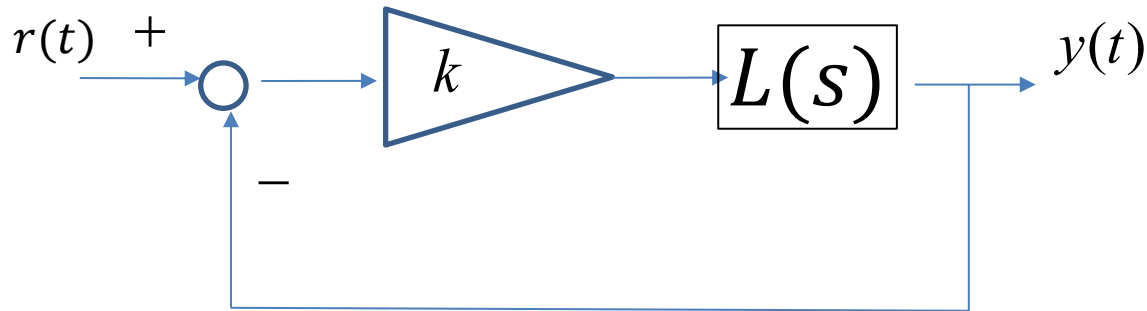
```
step(F, 5)
```



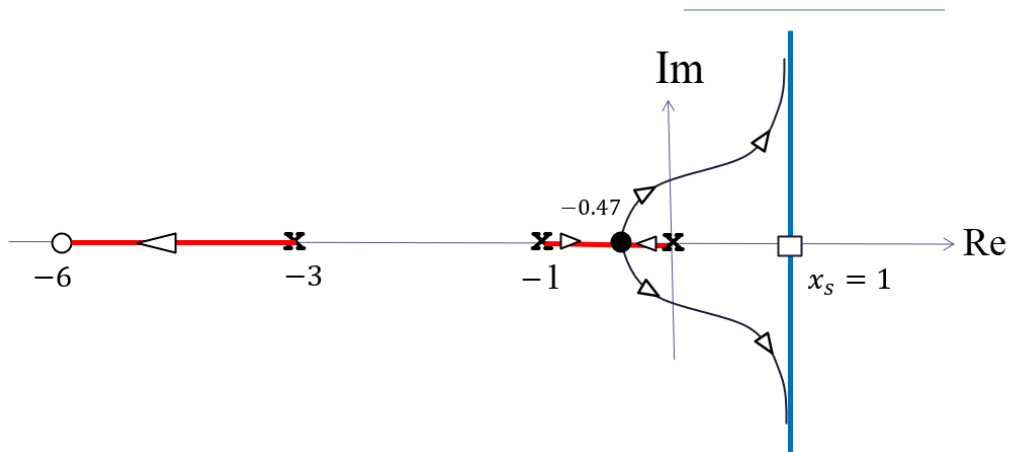
Tracciamento del Luogo delle radici

Funzione “rlocus”

Esempio di tracciamento svolto a lezione



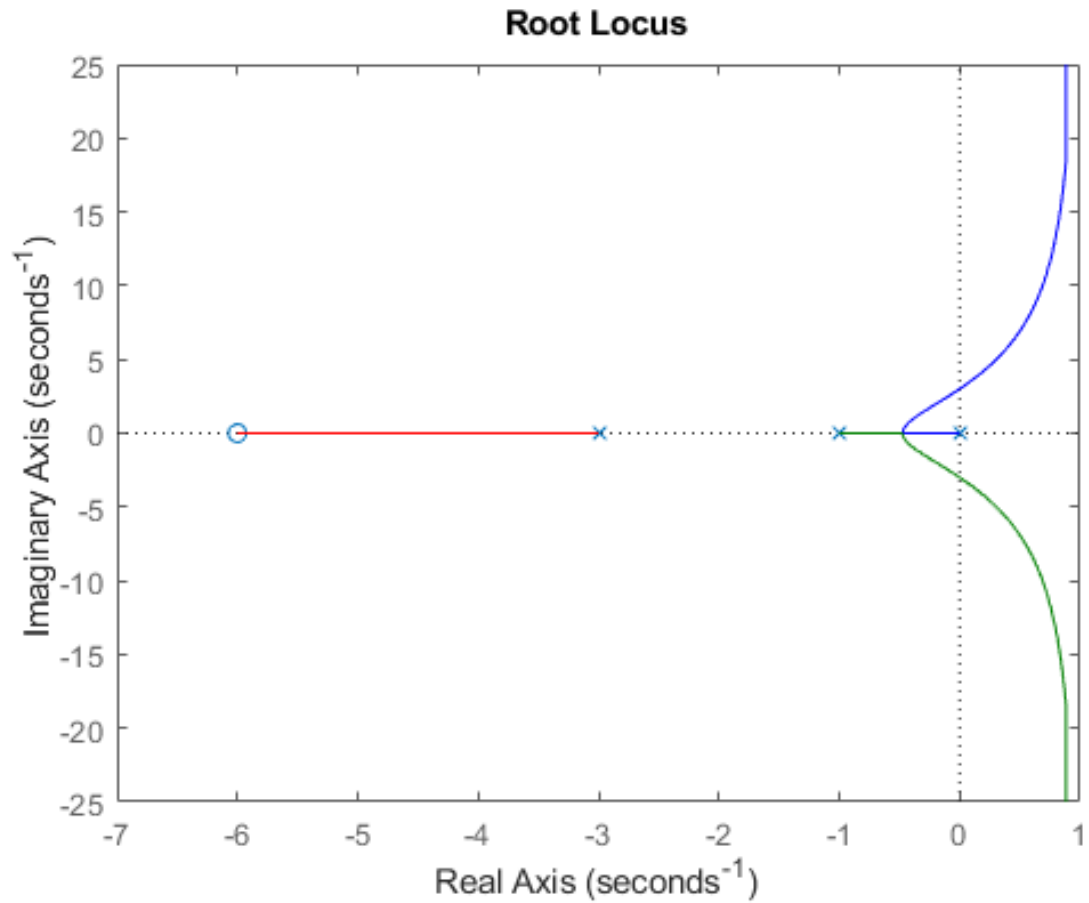
$$L(s) = \frac{5(s + 6)}{s(s + 1)(s + 3)}$$



Deve dapprima essere definita una variabile che rappresenti il sistema dinamico $L(s)$, e successivamente si passa in ingresso tale variabile alla funzione rlocus .

Codice possibile

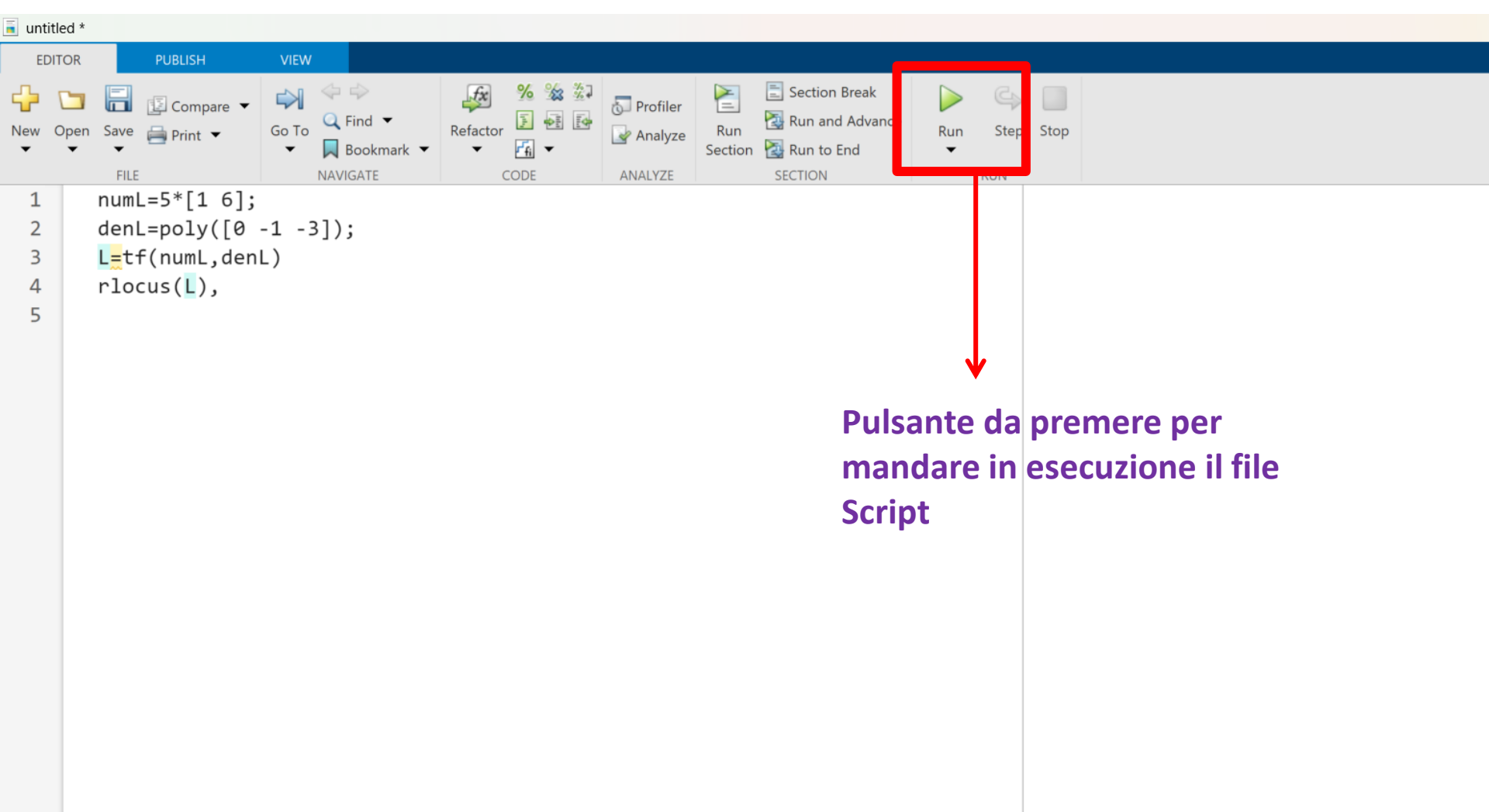
```
s=tf('s')  
L=5*(s+6)/(s*(s+1)*(s+3));  
rlocus(L),
```



Sintassi alternative

```
numL=5*[1 6];  
denL=poly([0 -1 -3]);  
L=tf(numL,denL)  
rlocus(L),
```

```
numL=5*[1 6];  
denL=poly([0 -1 -3]);  
rlocus(numL,denL)
```



A lezione avevamo dedotto i seguenti valori limite per il guadagno k

$$0 < k < 0.022$$

Risposta monotona esponenziale (tre poli reali negativi ed un o zero più in alta frequenza rispetto ai tre poli).

$$0.0022 < k < 1.2$$

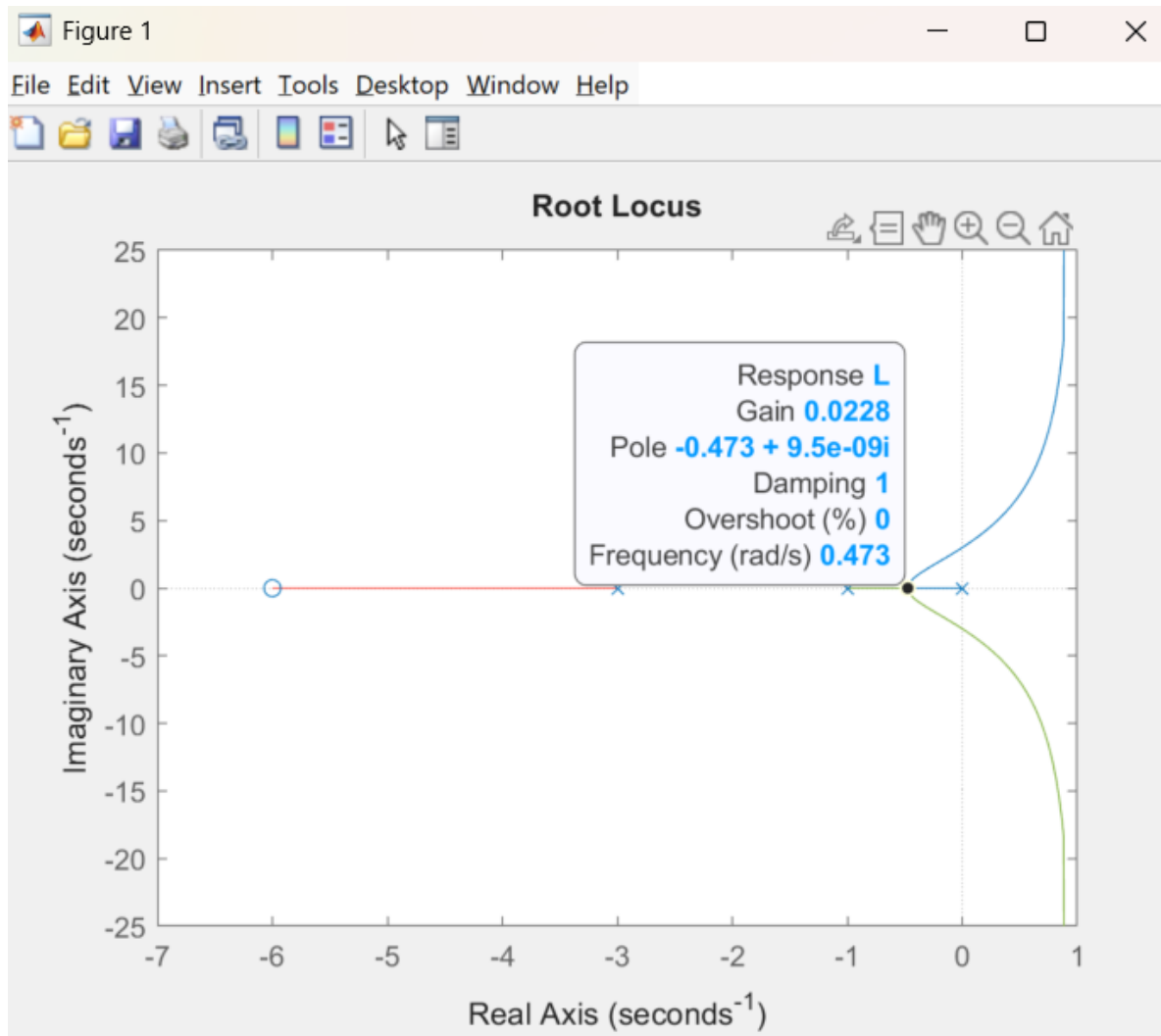
Risposta oscillatoria smorzata. Presenza di poli complessi coniugati

$$k > 1.2$$

Sistema di controllo instabile. La risposta al gradino diverge perchè due poli hanno parte reale positiva

Facciamo scorrere la freccia del mouse sopra i rami del LdR per accedere al valore del guadagno associato a quel punto (Gain), cioè per effettuare la taratura, e ad alcune caratteristiche della risposta a ciclo chiuso (Smorzamento (Damping), Sovraelongazione (Overshoot), e pulsazione naturale (Frequency))

Taratura del punto doppio (approssimata)



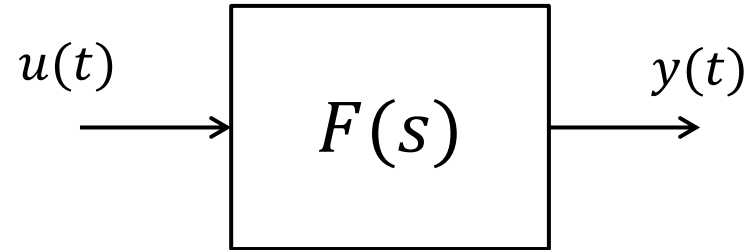
Tracciamento dei Diagrammi di Bode

Funzione “**bode**”

Deve dapprima essere definita una variabile che rappresenti il sistema dinamico $G(s)$ del quale si desidera costruire i diagrammi, e successivamente si passa in ingresso tale variabile alla funzione `bode` .

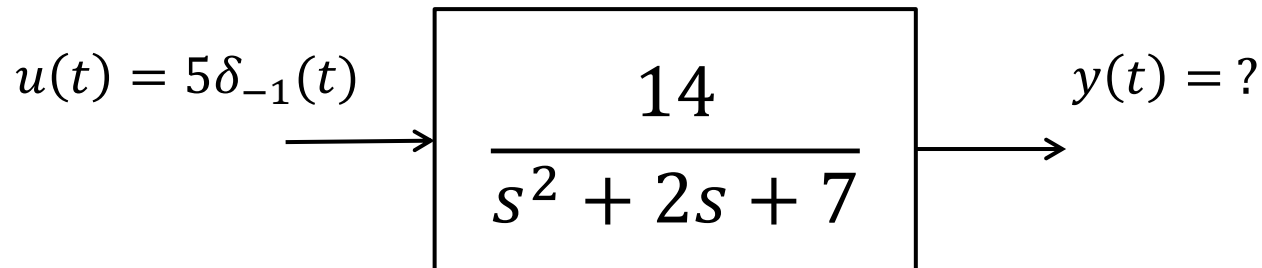
Simulazione dinamica di sistemi LTI a ciclo aperto mediante Simulink

Impariamo ad eseguire la simulazione dinamica a **ciclo aperto** in cui un sistema dinamico LTI descritto dalla generica FdT $F(s)$ viene sottoposto ad un certo segnale di ingresso $u(t)$, e si desidera visualizzarne la risposta forzata $y(t)$



Es. 1

Piu in dettaglio, ci proponiamo di visualizzare la risposta al gradino di ampiezza 5 ($u(t) = 5 \delta_{-1}(t)$) di un processo del secondo ordine (**esempio analizzato a lezione nelle slides "Sistemi dinamici elementari"**)



Realizziamo il modello Simulink

La realizzazione di un modello Simulink per la simulazione dinamica avviene per via grafica, realizzando uno **schema a blocchi** in tutto e per tutto simile a quelli che normalmente disegniamo a lezione per rappresentare i sistemi di controllo.

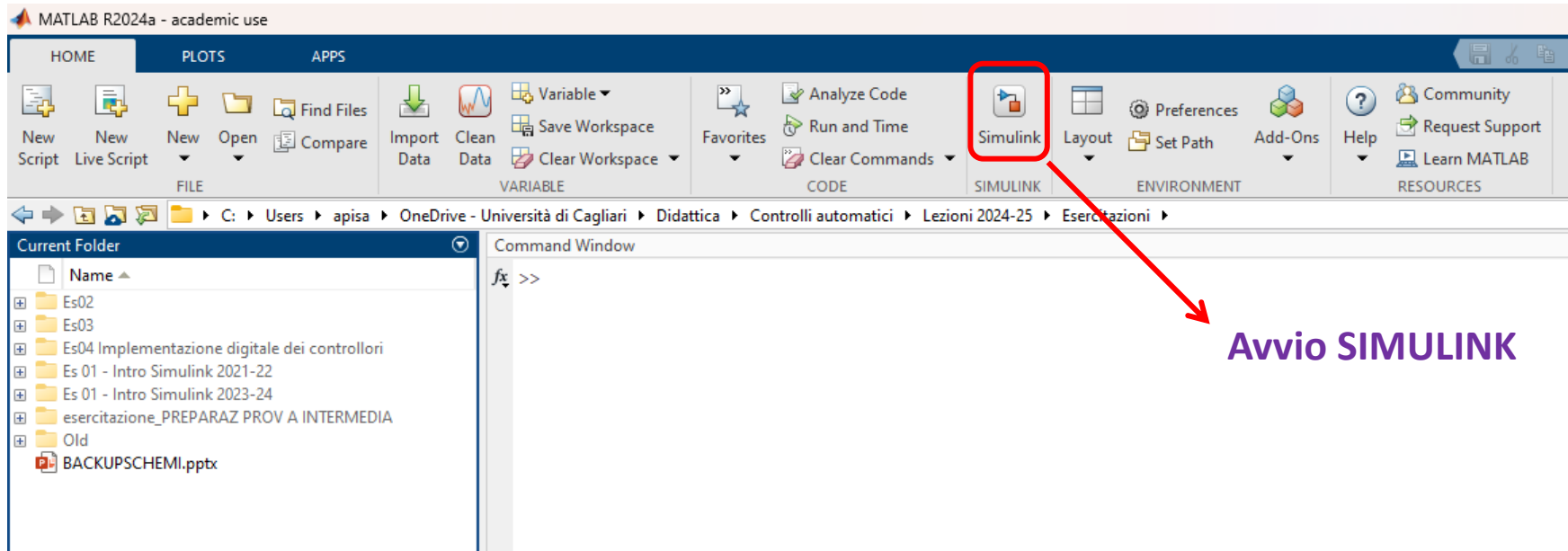
La realizzazione di un modello Simulink avviene attraverso tre fasi:

- si importano in una pagina di lavoro i **blocchi elementari** necessari per la realizzazione dello schema di simulazione
- si **parametrizzano i blocchi** in modo che implementino le funzionalità desiderate
- si **interconnettono tra loro i blocchi** per realizzare lo schema desiderato

Fatto ciò, si può avviare la simulazione e visualizzarne i risultati

Si apra Matlab

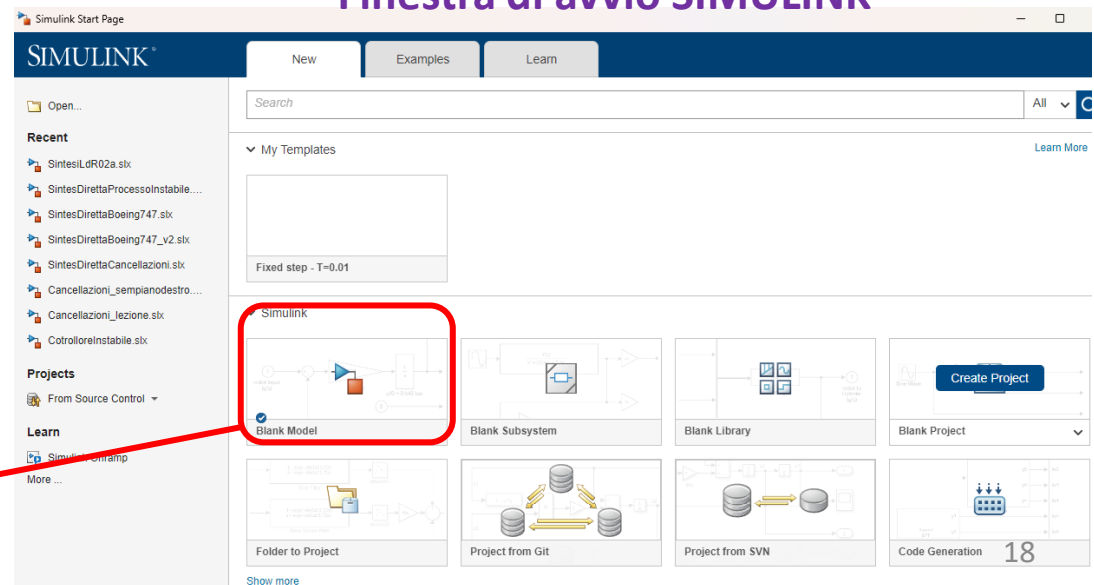
Finestra di avvio di Matlab (rel. R2024a)



Avvio SIMULINK

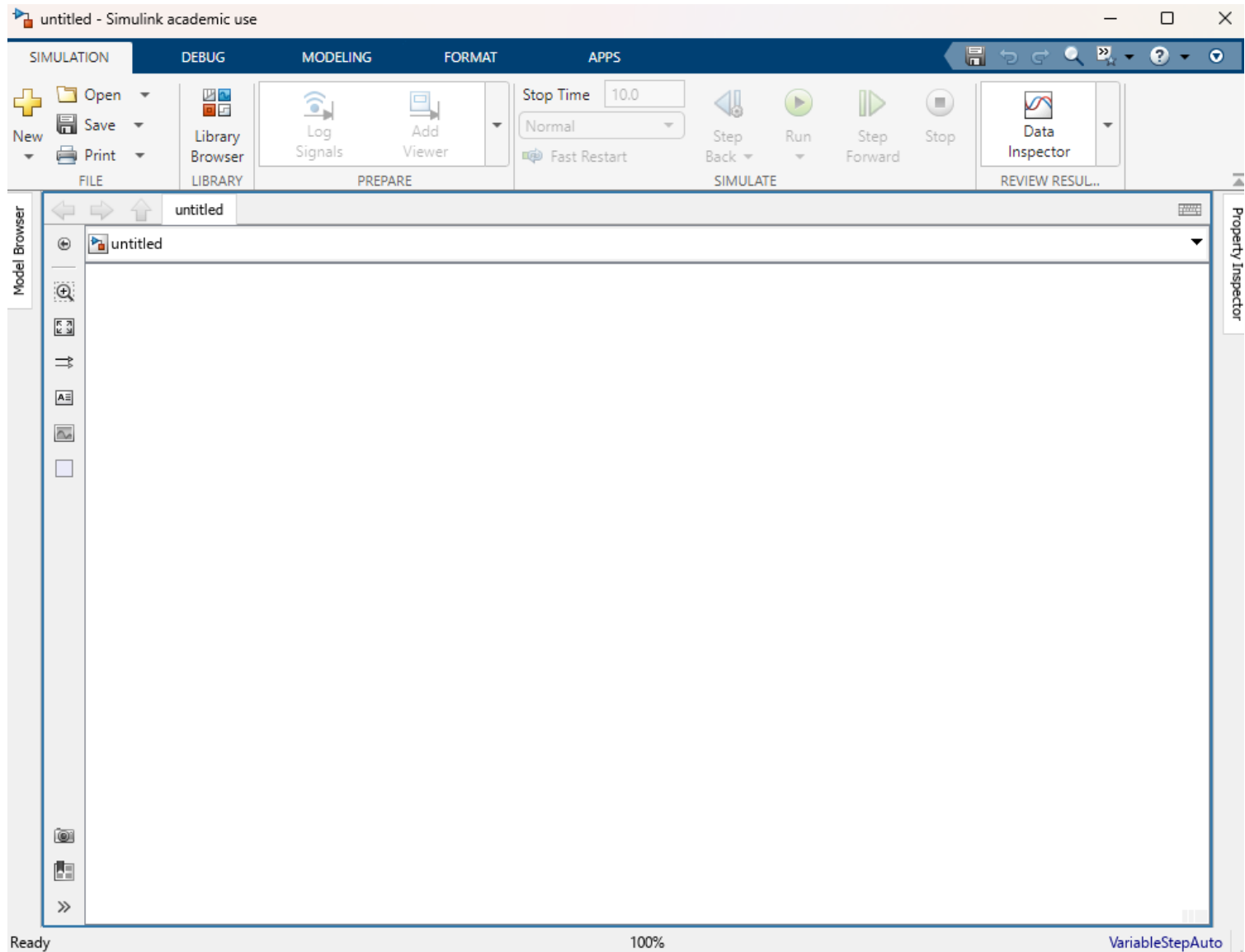
Finestra di avvio SIMULINK

Dalla finestra di avvio di Matlab,
apriamo Simulink.
Si apre la relativa finestra di avvio:



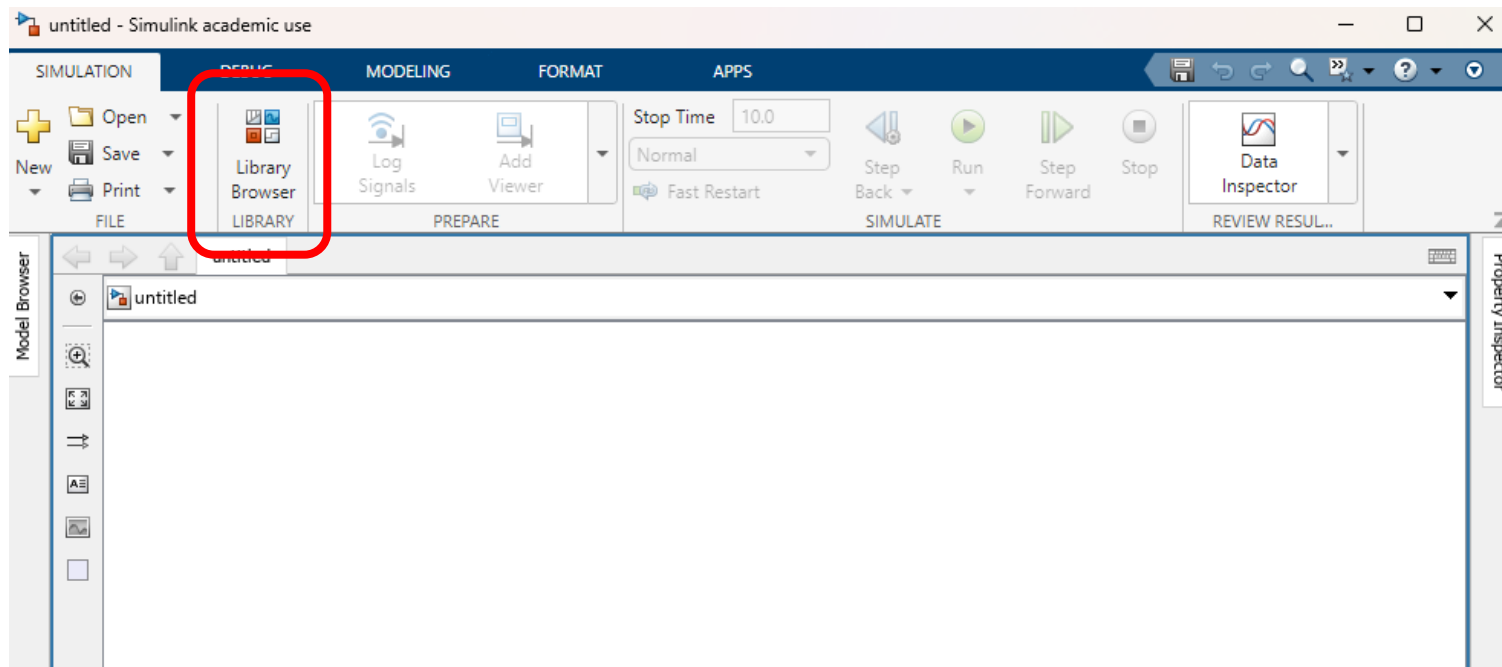
Apertura di un Blank Model

Modello Simulink in bianco

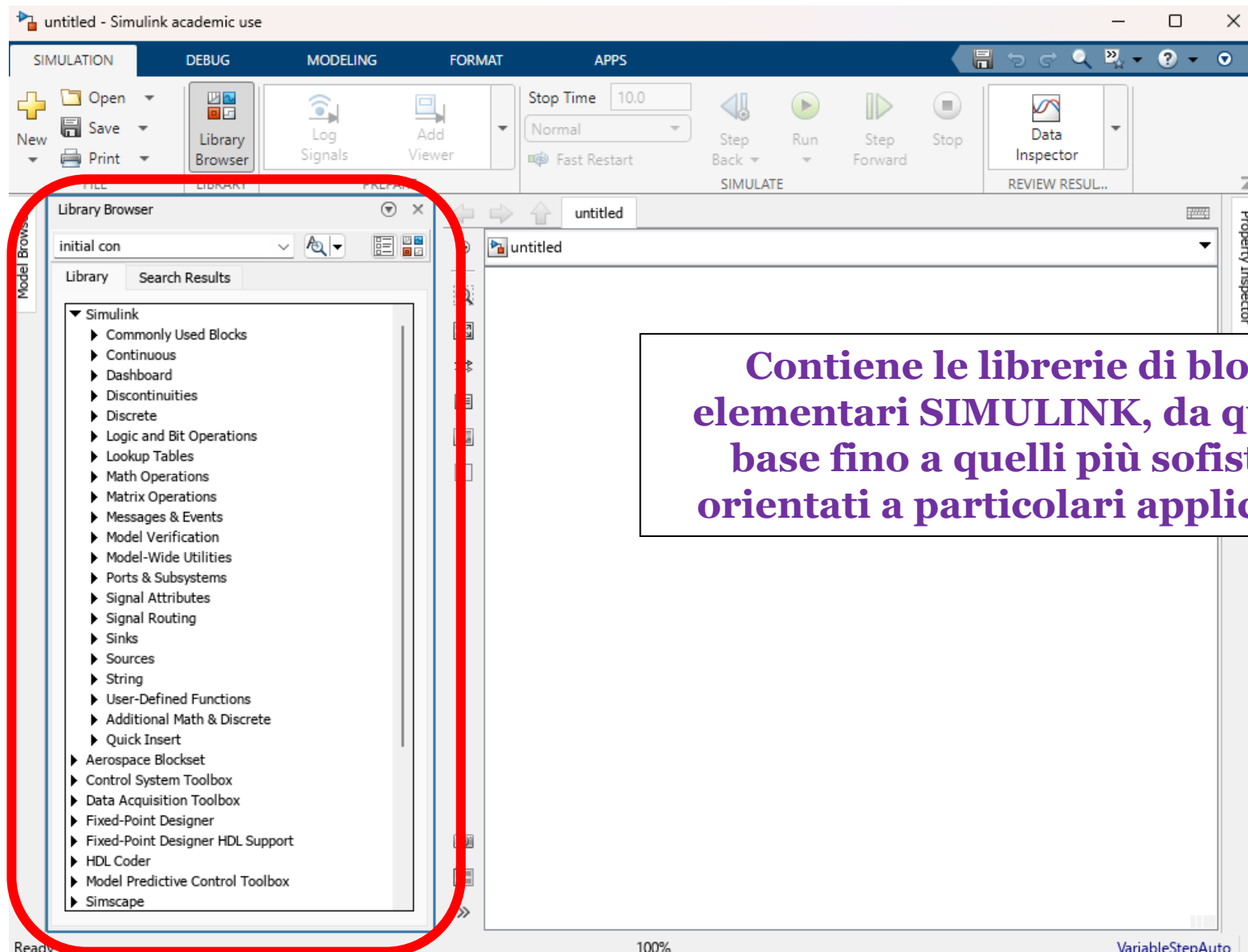


La realizzazione di modelli di simulazione dinamica avviene per via grafica **assemblando fra loro all'interno della pagina di lavoro un certo numero di blocchi Simulink**, in modo da implementare le funzionalità desiderate.

I blocchi Simulink sono allocati all'interno di **librerie**. E' possibile accedere al «Library browser» cliccando il relativo pulsante nella finestra che ospita il modello in bianco (Menu: «SIMULATION»)



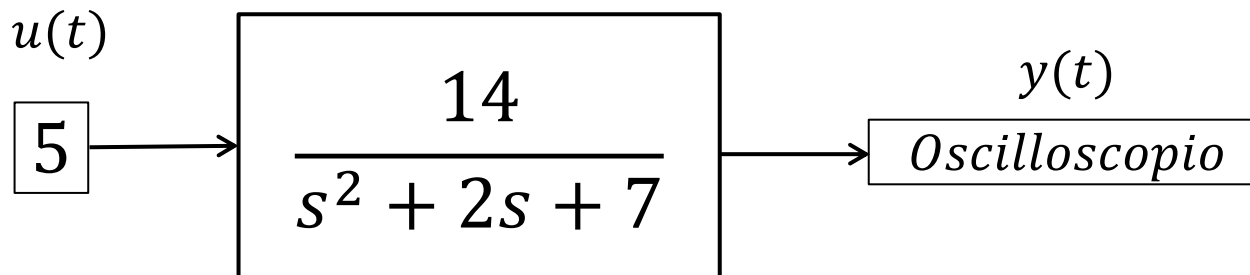
Simulink Library Browser



Contiene le librerie di blocchi elementari SIMULINK, da quelli di base fino a quelli più sofisticati orientati a particolari applicazioni

Per realizzare lo schema a blocchi associato alla simulazione dinamica di un sistema a ciclo aperto sono sufficienti **tre blocchi elementari** Simulink:

- 1 un blocco elementare che implementi un sistema dinamico descritto attraverso la sua FdT
- 2 un blocco elementare che generi il segnale costante di ingresso
- 3 un blocco elementare «oscilloscopio» che consenta la visualizzazione del segnale di uscita

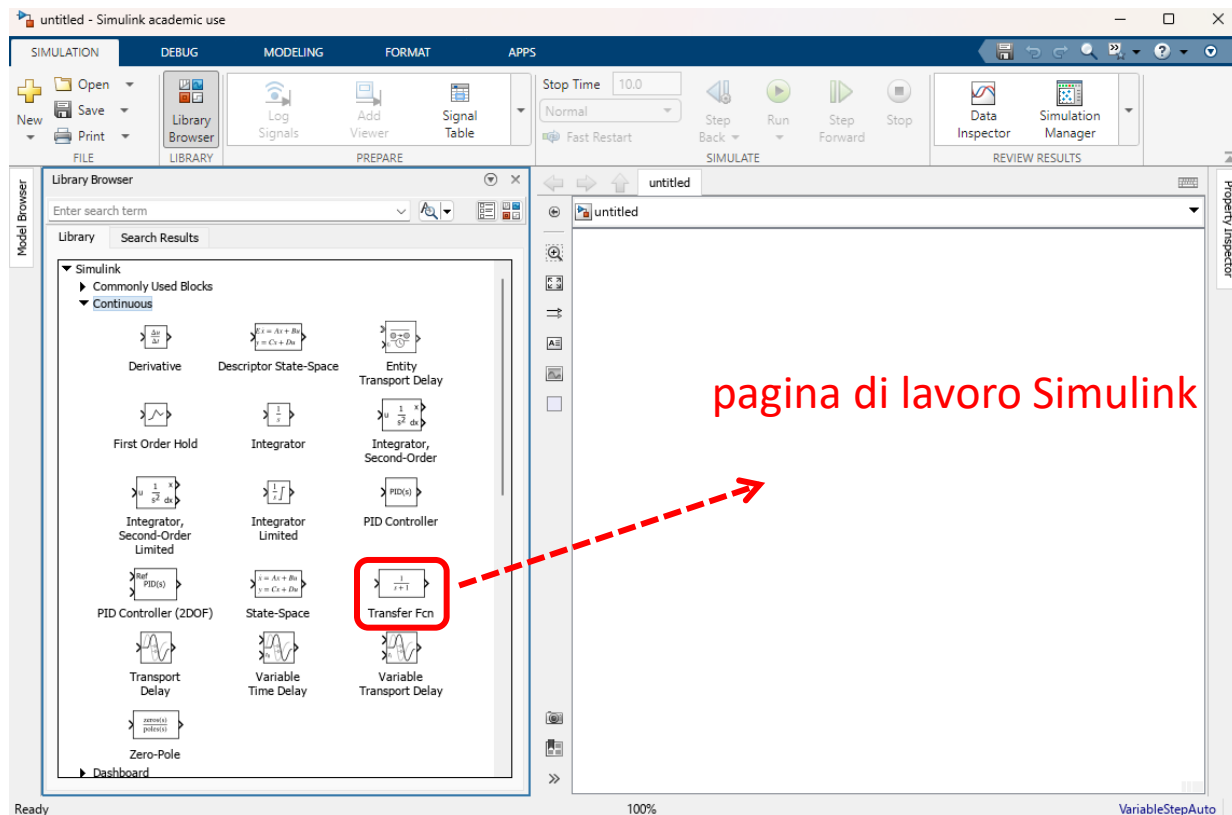


Nella pagina di lavoro Simulink, andremo a “disegnare” una rappresentazione analoga a quella riportata qui sopra

Importiamo nella pagina di lavoro il blocco “Transfer Fcn” (funzione di trasferimento), che si trova nella sotto-libreria “Continuous” della libreria principale “Simulink”

E' uno dei blocchi (non l'unico) per mezzo dei quali si possono realizzare in ambiente Simulink delle funzioni di trasferimento

L'importazione dei blocchi nella pagina di lavoro si effettua con il mouse mediante drag-and-drop

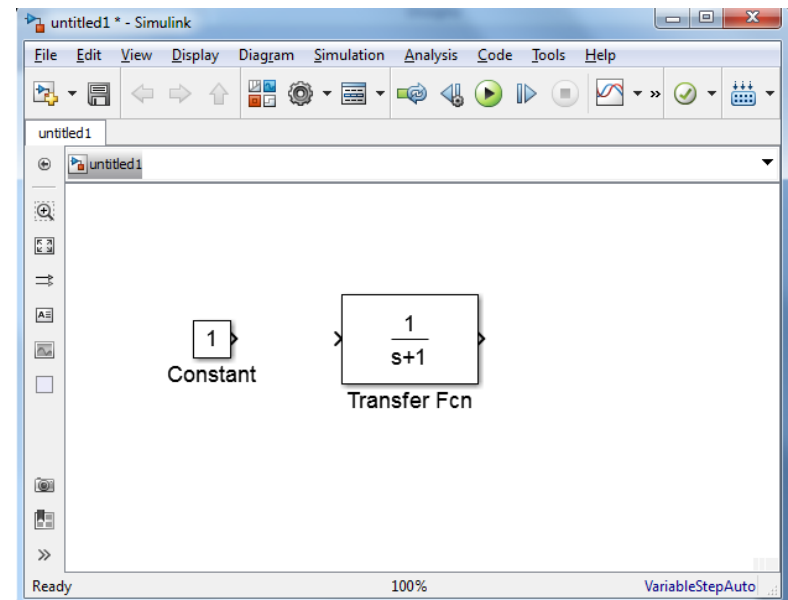


Ora importiamo il blocco necessario per generare il segnale di ingresso costante.

Tale blocco si chiama `Constant`, e si trova nella sotto-libreria “`Sources`” della libreria principale “`Simulink`”, che contiene una ampia varietà di blocchi elementari preposti alla **generazione di segnali**

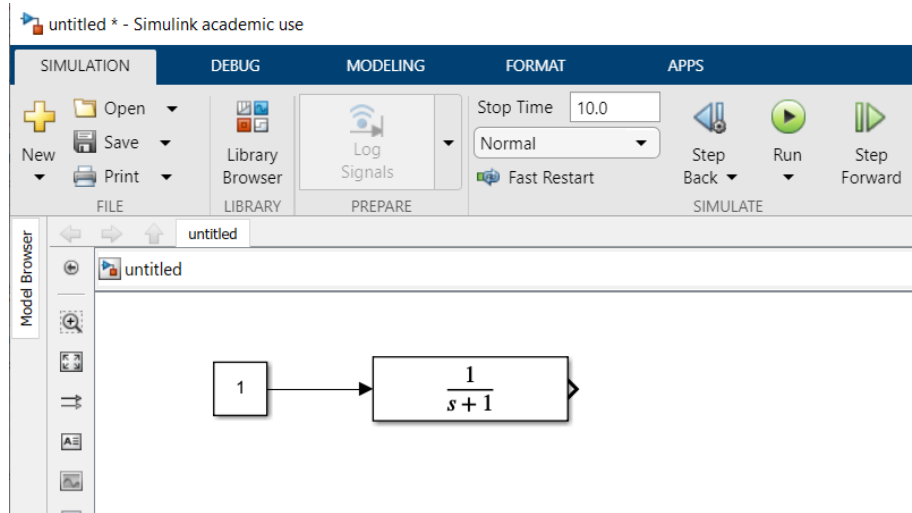
Disponiamo i blocchi come nella figura a lato

Si noti che il blocco `Constant` possiede unicamente un terminale di uscita, mentre invece il blocco `Transfer Fcn` possiede un terminale di ingresso ed un terminale di uscita, come è lecito attendersi.



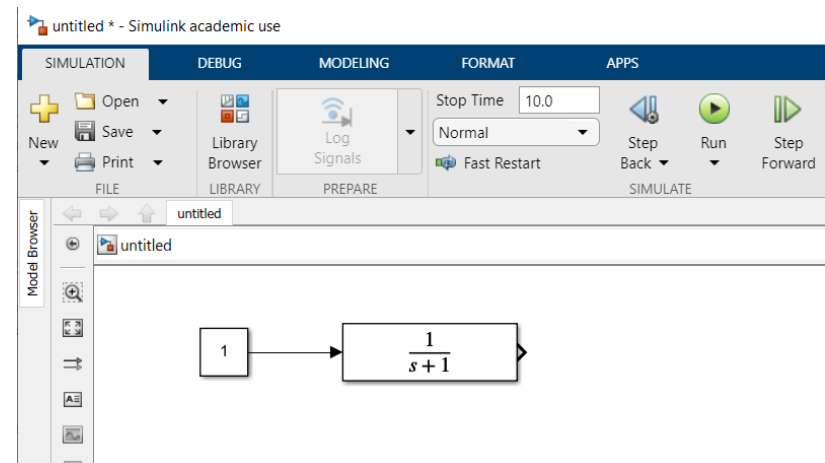
Interconnettiamo i due blocchi.

Il tracciamento di una connessione si effettua portandosi con il mouse nel punto di inizio della linea di connessione (quindi nel terminale di uscita del blocco Constant), premendo il tasto sinistro, e successivamente portandosi con il mouse - mantenendo premuto il tasto - verso il punto di destinazione, in questo caso il terminale di ingresso del blocco Transfer Fcn.



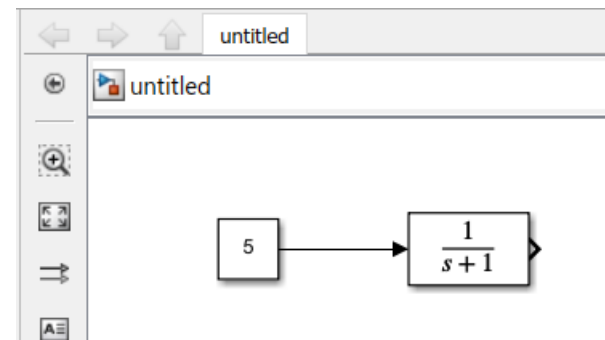
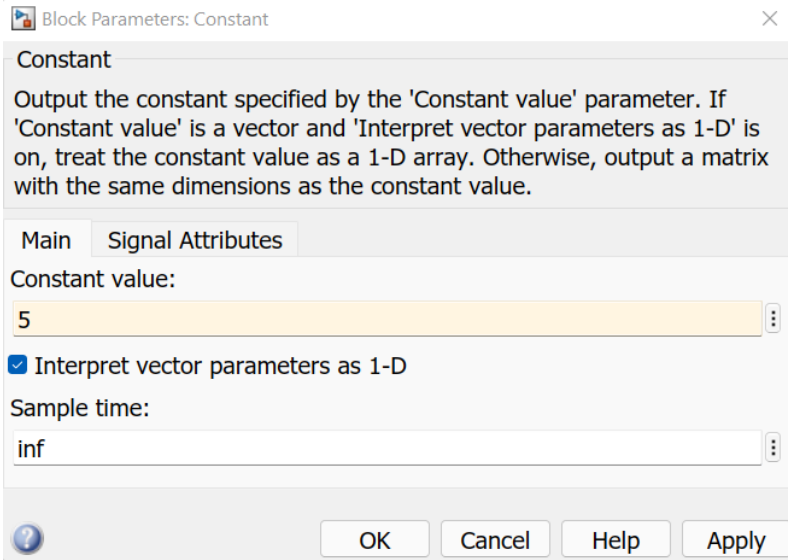
Per effettuare un collegamento tra due blocchi vi è anche una **procedura rapida**. Si deve selezionare il blocco di origine (cliccandovi sopra), e si deve successivamente selezionare il blocco di destinazione con il **tasto ctrl premuto**.

Di **default**, il blocco Constant è parametrizzato per generare un segnale costante di ampiezza unitaria, mentre il blocco Transfer Fcn è parametrizzato di default con la FdT $\frac{1}{s+1}$

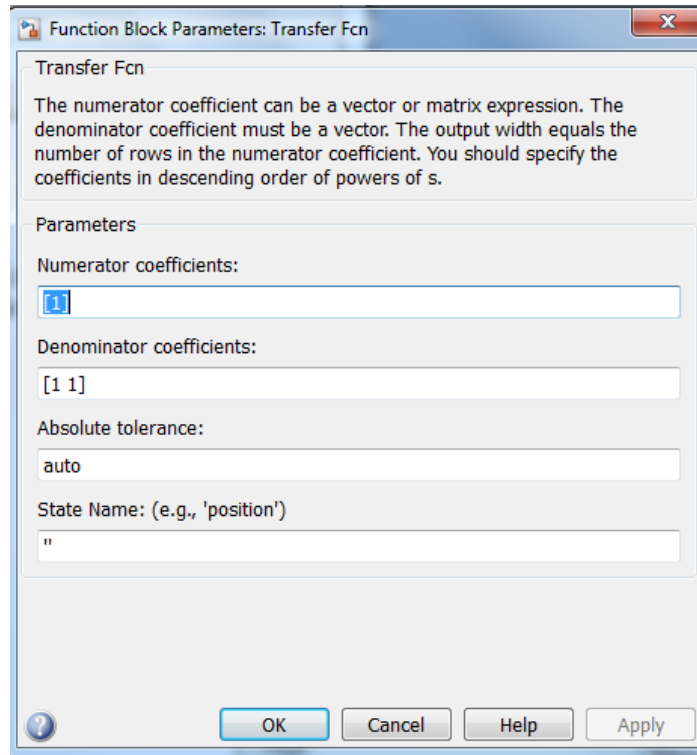


Facendo doppio click su un blocco, si accede alla sua finestra di parametrizzazione.

Il blocco Constant va riconfigurato modificando l'ampiezza del gradino da 1 a 5.



Dobbiamo ora configurare il blocco Transfer Fcn.
Apriamone la finestra di parametrizzazione



The image shows a MATLAB/Simulink dialog box titled "Function Block Parameters: Transfer Fcn". The dialog box contains the following fields and text:

- Transfer Fcn**: A text area containing the instruction: "The numerator coefficient can be a vector or matrix expression. The denominator coefficient must be a vector. The output width equals the number of rows in the numerator coefficient. You should specify the coefficients in descending order of powers of s."
- Parameters**: A section containing the following fields:
 - Numerator coefficients:** A text field containing the value `1`.
 - Denominator coefficients:** A text field containing the value `[1 1]`.
 - Absolute tolerance:** A text field containing the value `auto`.
 - State Name: (e.g., 'position')**: A text field containing the value `"`.
- Buttons**: At the bottom right, there are four buttons: `OK`, `Cancel`, `Help`, and `Apply`.

Parametrizzazione del blocco Transfer Function

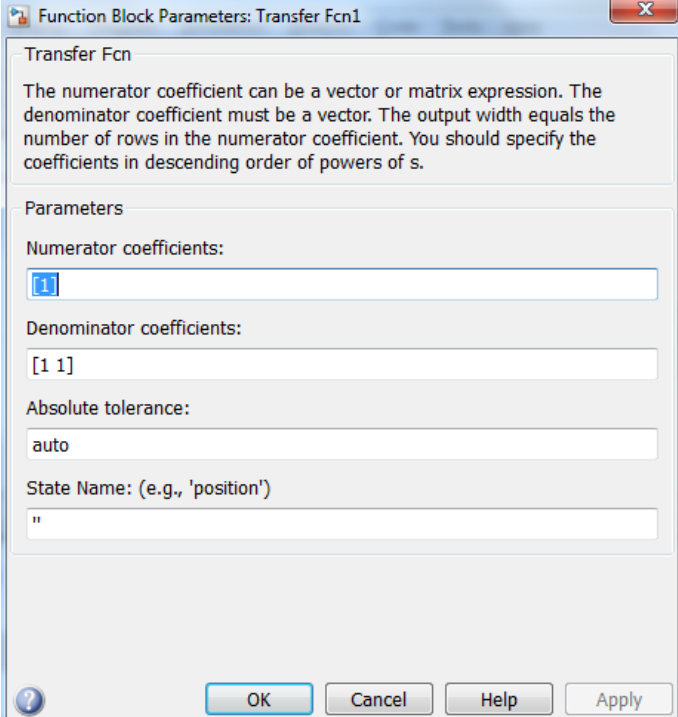
Il blocco deve rappresentare la FdT

$$F(s) = \frac{14}{s^2 + 2s + 7}$$

Si devono specificare i **coefficienti dei polinomi a numeratore e denominatore** della FdT utilizzando la notazione Matlab per la rappresentazione dei polinomi (un vettore che contiene i coefficienti del polinomio **in ordine decrescente** rispetto alle potenze di s)

$$14 \Rightarrow [14]$$

$$s^2 + 2s + 7 \Rightarrow [1 \ 2 \ 7]$$



Function Block Parameters: Transfer Fcn1

Transfer Fcn

The numerator coefficient can be a vector or matrix expression. The denominator coefficient must be a vector. The output width equals the number of rows in the numerator coefficient. You should specify the coefficients in descending order of powers of s.

Parameters

Numerator coefficients:

14

Denominator coefficients:

[1 1]

Absolute tolerance:

auto

State Name: (e.g., 'position')

OK Cancel Help Apply

Altri esempi:

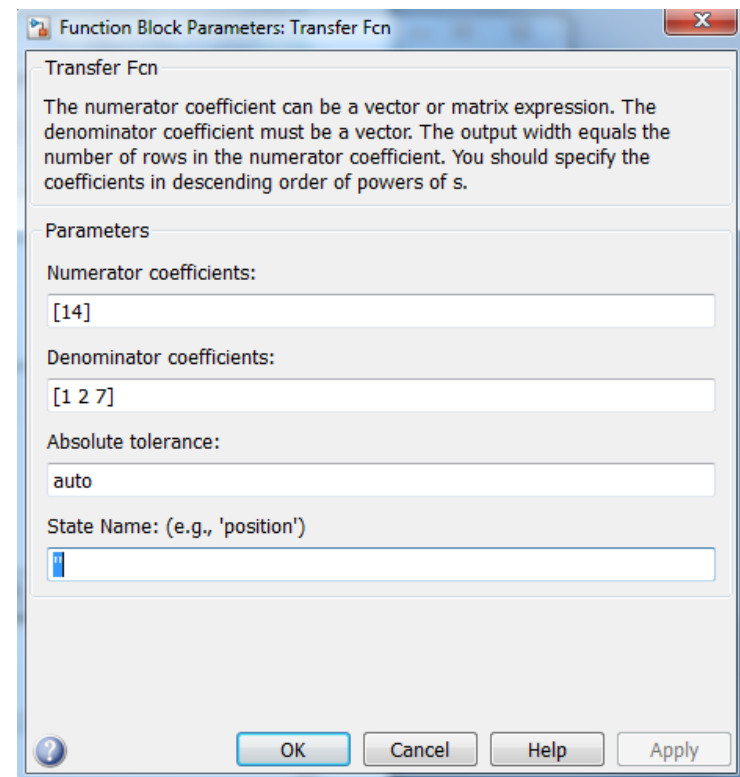
$$s \Rightarrow [1 \ 0]$$

$$s(s + 1) = s^2 + s \Rightarrow [1 \ 1 \ 0]$$

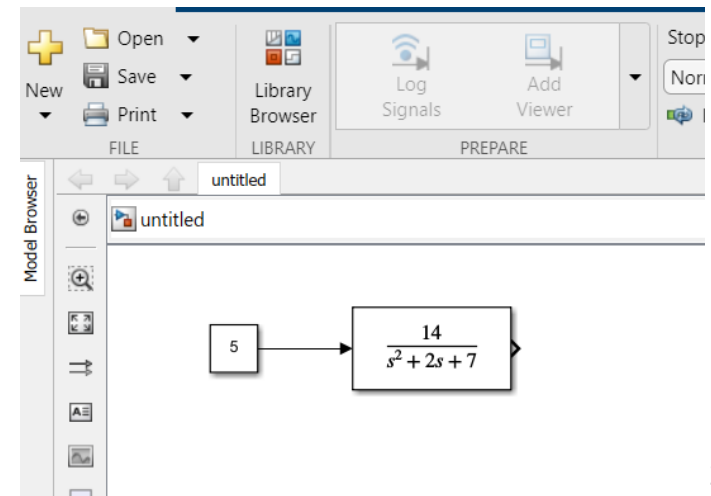
$$s^4 - 2s^3 + 3s + 2 \Rightarrow [1 \ -2 \ 0 \ 3 \ 2]$$

$$s^3 + 4 \Rightarrow [1 \ 0 \ 0 \ 4]$$

Il blocco deve essere pertanto parametrizzato nella seguente maniera.



Dopo aver premuto il tasto OK, l'aspetto del blocco Transfer Fcn nella pagina di lavoro cambia, e visualizza al proprio interno la FdT avente i parametri scelti (eventualmente il blocco deve essere ingrandito affinché la visualizzazione sia corretta)

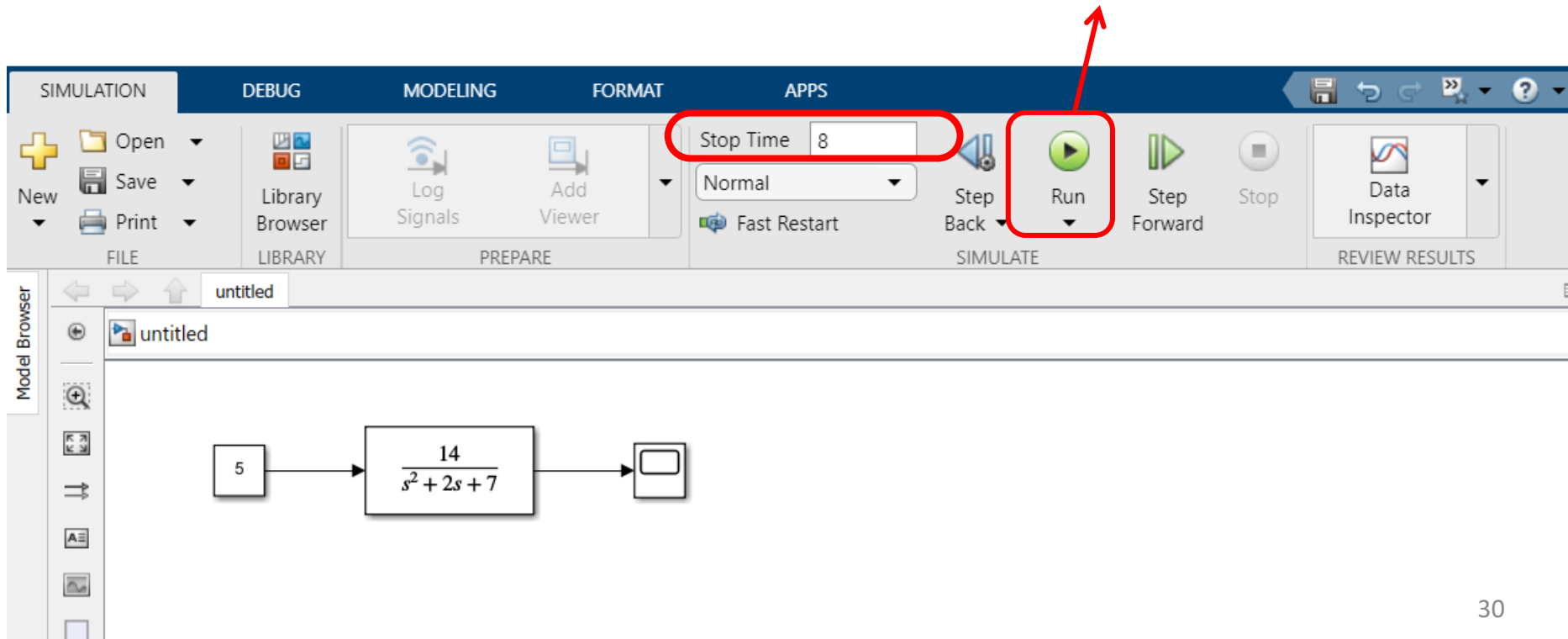


Ora importiamo il blocco “**Scope**” (oscilloscopio), che consente la visualizzazione di un segnale.

Il blocco Scope sta nella sotto-libreria “**Sinks**” della libreria principale “**Simulink**”. Dopo averlo importato nella pagina di lavoro, colleghiamone il terminale di ingresso al terminale di uscita del blocco Transfer Fcn.

Impostare nella casella **Stop Time** la durata della simulazione (il valore di default è 10 secondi, scegliamo di eseguire la simulazione dinamica per **8 secondi**), e cliccare sul **pulsante Run** per avviare la simulazione.

Pulsante «Run»



Prima di fare doppio click sul blocco Scope e visualizzare la risposta, ne valutiamo «su carta» alcuni parametri caratteristici mediante le formule viste a lezione (il medesimo esempio è analizzato nella dispensa «05-Sistemi Dinamici Elementari»)

$$F(s) = \frac{14}{s^2 + 2s + 7}$$

$F(s)$ è asintoticamente stabile in quanto il polinomio caratteristico è di secondo grado ed ha tutti i coefficienti di segno concorde.

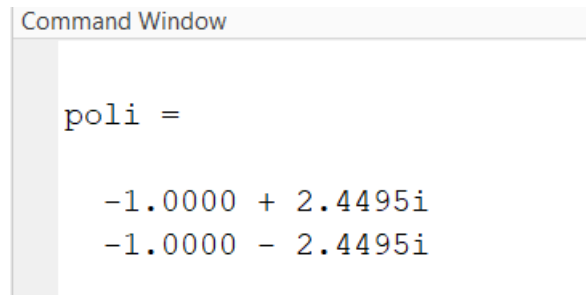
Guadagno statico $\mu = F(0) = 2$

Valore di regime della risposta all'ingresso $u(t) = 5 \delta_{-1}(t)$ [T.F.R.G.] $y_{\infty} = 5 \cdot \mu = 10$

Radici di un polinomio (funzione «roots»)

Poli: `>> poli=roots([1 2 7])`

$$p_{1,2} = -1 \pm 2.449i$$



```
Command Window

poli =

    -1.0000 + 2.4495i
    -1.0000 - 2.4495i
```

$$p_{1,2} = -1 \pm 2.449i = a + bj$$

Parte reale: $a = -1$

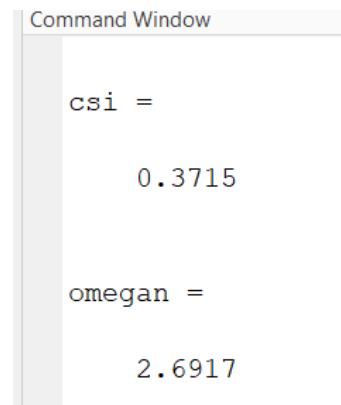
Parte immaginaria: $b = \pm 2.449$

Smorzamento $\xi = -\frac{a}{\sqrt{a^2 + b^2}} = 0.37$

Pulsazione naturale $\omega_n = \sqrt{a^2 + b^2} = 2.69$

Istruzioni Matlab per il calcolo dello smorzamento e della pulsazione naturale, e relativo output

```
clc
a=-1;
b=2.499;
csi=-a/sqrt(a^2+b^2)
omegan=sqrt(a^2+b^2)
```



```
Command Window

csi =

    0.3715

omegan =

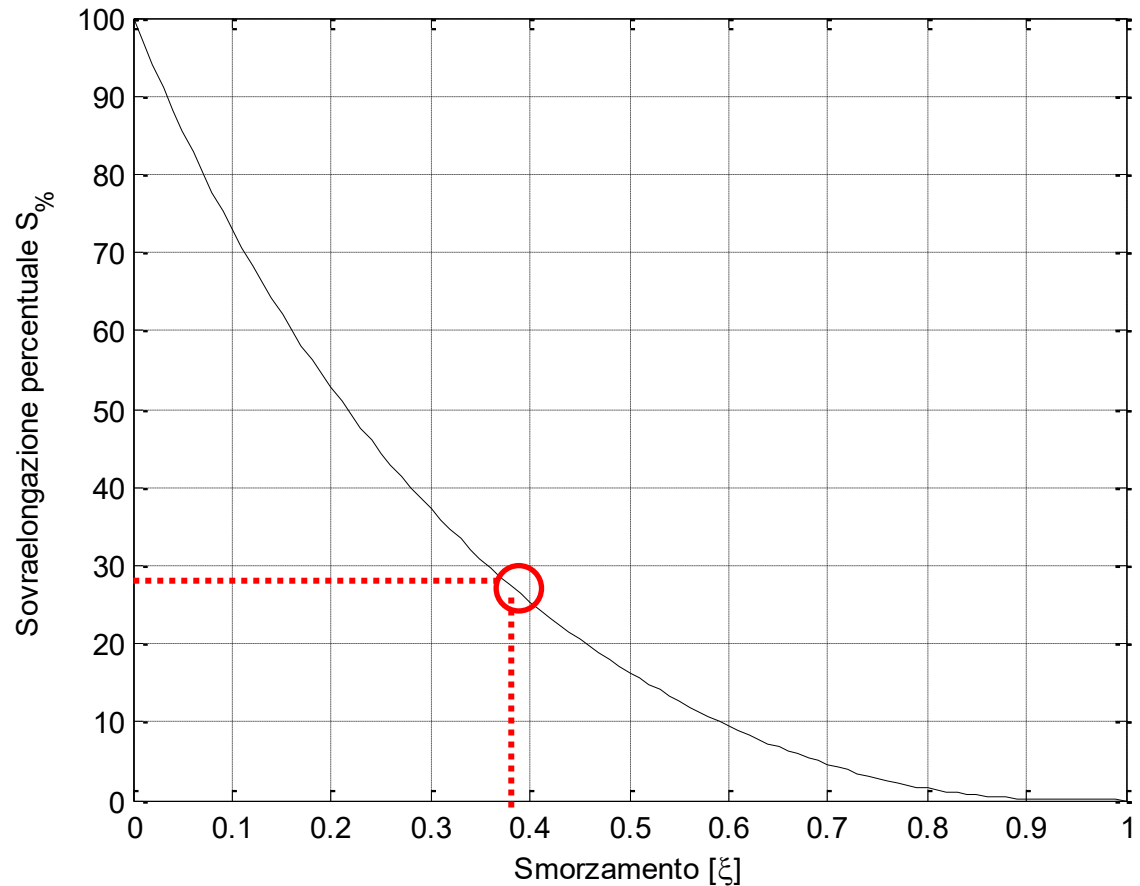
    2.6917
```


Sovraelongazione percentuale

$$S_{\%} = 100e^{-\frac{\xi\pi}{\sqrt{1-\xi^2}}}$$

$$\xi \approx 0.37$$

$$S_{\%} \approx 28$$



Il valore massimo dell'uscita durante il transitorio sarà pertanto

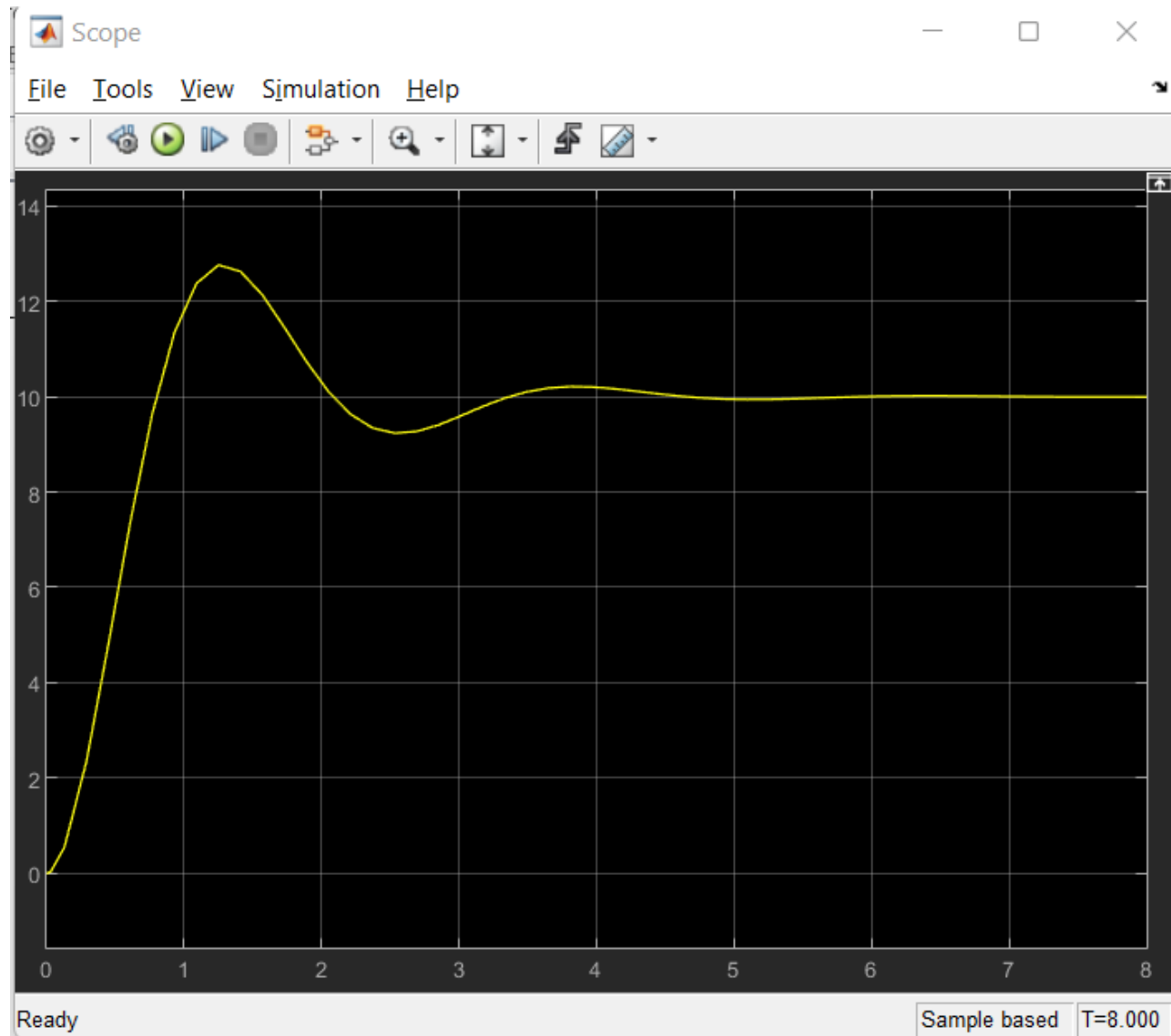
$$y_{max} = y_{\infty} + 0.28 y_{\infty} = 1.28 y_{\infty} = 12.8$$

$$\tau_{eq} = \frac{1}{\xi \omega_n} = \mathbf{1 \text{ s}}$$

Costante di tempo equivalente e tempo di assestamento

	T_a
$F(s) = \frac{\mu \omega_n^2}{(s^2 + 2\xi \omega_n s + \omega_n^2)}$	$4.6 \tau_{eq}$
$F(s) = \frac{14}{s^2 + 2s + 7}$	4.6 s

Ora dopo avere eseguito la simulazione, fare doppio click sul blocco “Scope”. La risposta possiede le caratteristiche attese



FILE: Esemplio01.slx

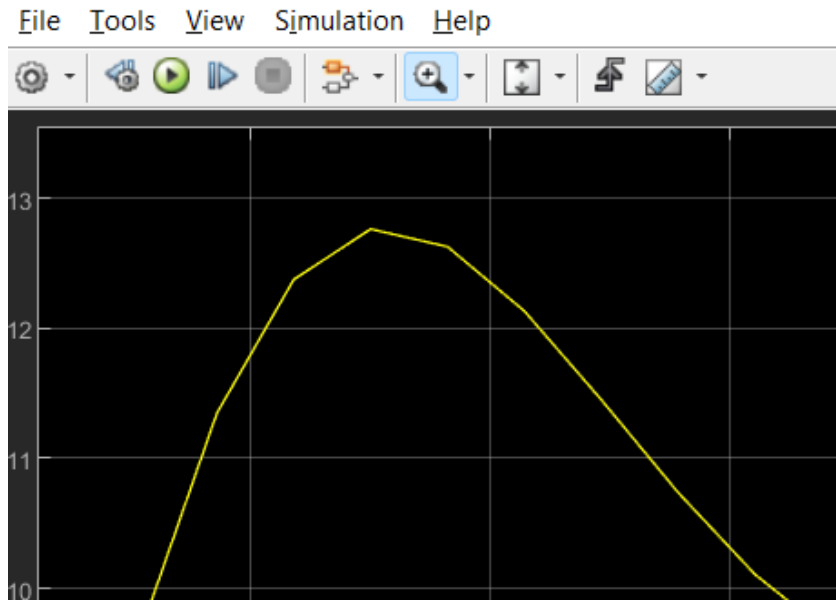


Grafico “spigoloso”

Il grafico è stato realizzato interpolando un numero di punti insufficiente (l'uscita del sistema è stata calcolata in corrispondenza di un numero insufficiente di istanti temporali da non consentire la creazione di un grafico sufficientemente regolare)

Dobbiamo modificare i parametri di configurazione del “**solver**”, che definisce (fra le altre cose) il passo di discretizzazione temporale che viene impiegato nella esecuzione del modello (cioè nella **risoluzione numerica della equazione differenziale** associata al modello).

Fare click con il tasto destro in qualunque punto dello schema Simulink e scegliere dal menu «Model Configuration Parameters» (in alternativa premere Ctrl+E).

Search

Solver

- Data Import/Export
- Math and Data Types
- ▶ Diagnostics
- Hardware Implementation
- Model Referencing
- Simulation Target

Simulation time

Start time: 0.0 Stop time: 8

Solver selection

Type: Fixed-step Solver: auto (Automatic solver selection)

▼ Solver details

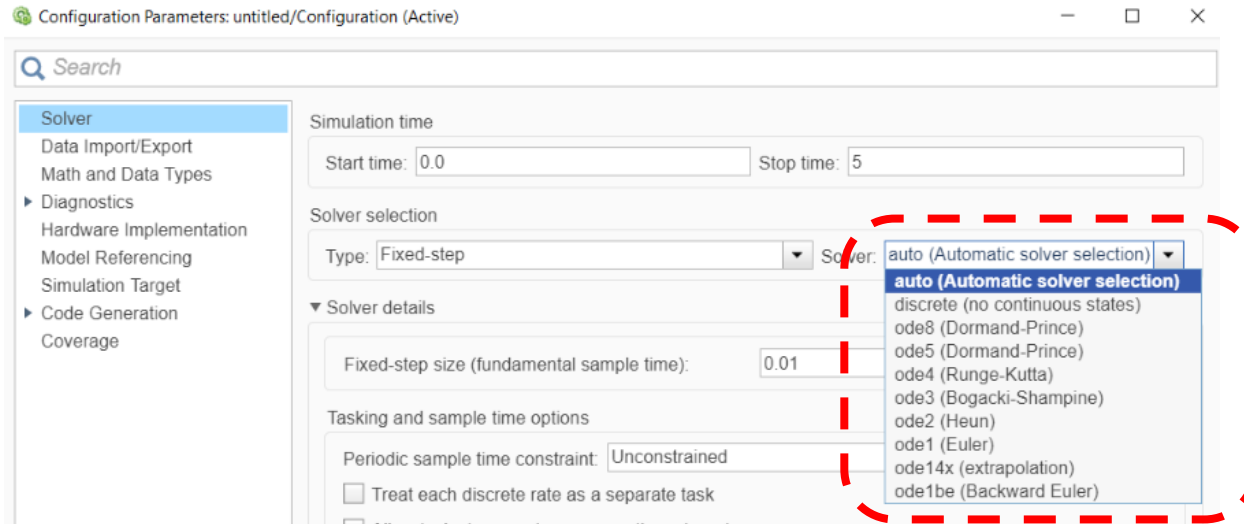
Fixed-step size (fundamental sample time): 0.01

Nel menu Solver impostare il Solver Type ed il Fixed Step Size come in figura. In questo modo **i segnali della simulazione saranno aggiornati (ricalcolati) ogni 0.01 secondi.**

Queste impostazioni vanno eseguite ogni volta che si apre un nuovo modello vuoto (è possibile definire un «template» che consente di creare direttamente un nuovo modello che possiede direttamente le impostazioni desiderate per il Solver)

Naturalmente se si avesse necessita di simulare un sistema dinamico in cui i segnali variano molto rapidamente la scelta di 0.01 secondi per il fixed-step size potrebbe diventare non più appropriata, ed il valore dovrebbe essere ulteriormente ridotto.

E' disponibile una ampia varietà di solutori numerici. Nello screenshot sottostante sono riportate le varie opzioni di scelta per i solutori a passo fisso.



Una **scelta ottimale per il solutore** bilancia, per il problema in esame, la precisione della soluzione e la mole di calcoli richiesta, che influenza il tempo di simulazione.

L'impostazione di scelta automatica del solver, che è la scelta di default, può dar luogo a soluzioni inaccurate in taluni casi «critici» come ad esempio modelli differenziali in cui intervengono segnali con scale di variazione temporale molto diverse fra loro (problemi «stiff») o modelli con elementi discontinui (non-smooth dynamics) e sono necessari solutori specifici.

Per una discussione più approfondita in merito alla scelta del solutore:

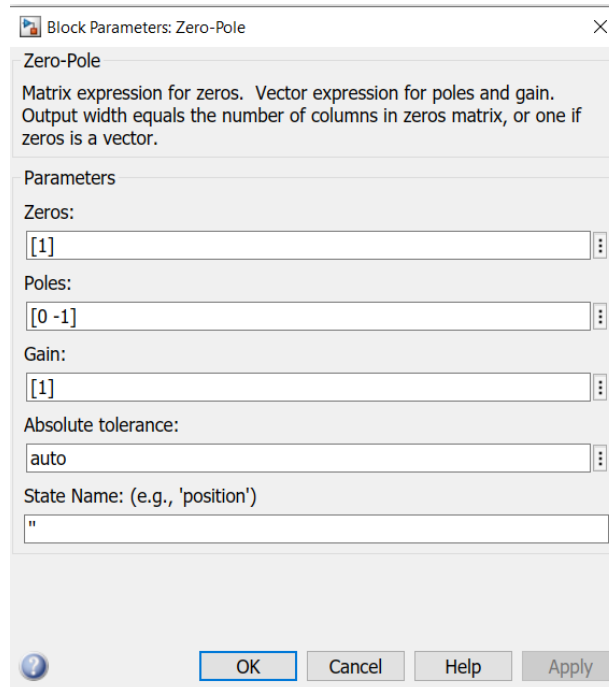
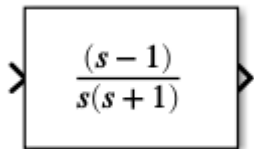
<https://www.mathworks.com/help/simulink/ug/choose-a-solver.html>

Vediamo un altro blocco elementare per la modellazione di FdT

Per modellare una FdT espressa nella forma seguente (fattorizzazione poli-zeri) il blocco Transfer Fcn risulta poco conveniente

$$F(s) = 2 \cdot \frac{(s + 0.5)}{(s + 1)(s + 2)(s + 5)}$$

Si può utilizzare in alternativa il blocco «Zero-Pole», che si trova sempre nella medesima libreria Continuous dalla quale prelevammo il blocco Transfer Fcn



The screenshot shows the 'Block Parameters: Zero-Pole' dialog box. It contains the following fields and labels:

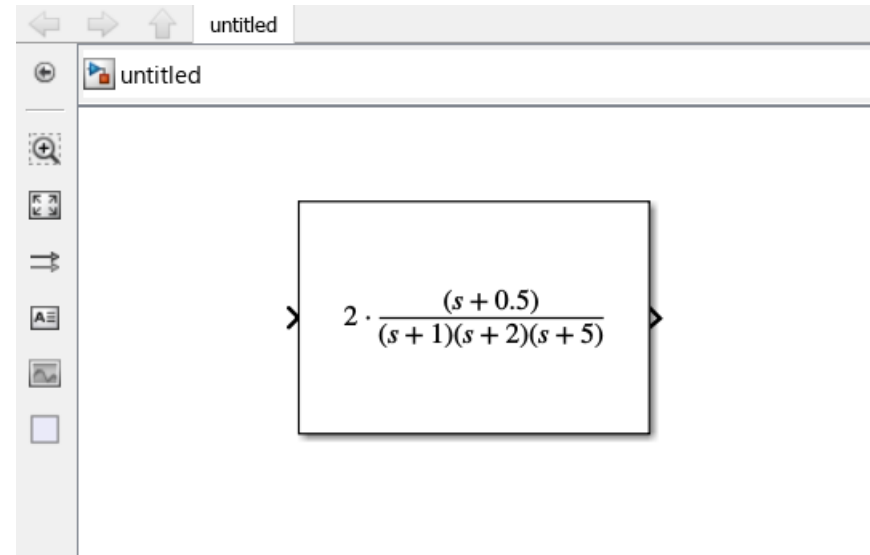
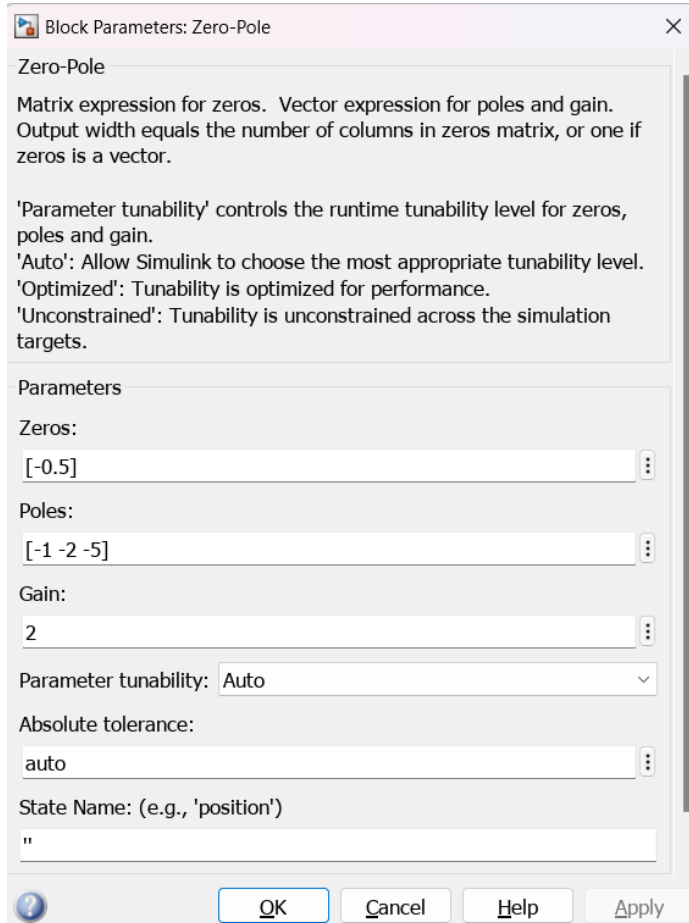
- Zero-Pole** (Section Header)
- Matrix expression for zeros. Vector expression for poles and gain. Output width equals the number of columns in zeros matrix, or one if zeros is a vector.
- Parameters** (Section Header)
- Zeros:** [1] (with a help icon)
- Poles:** [0 -1] (with a help icon)
- Gain:** [1] (with a help icon)
- Absolute tolerance:** auto (with a help icon)
- State Name:** (e.g., 'position') (with a help icon)
- Buttons: OK, Cancel, Help, Apply

Vettore che contiene tutti gli zeri

Vettore che contiene tutti i poli

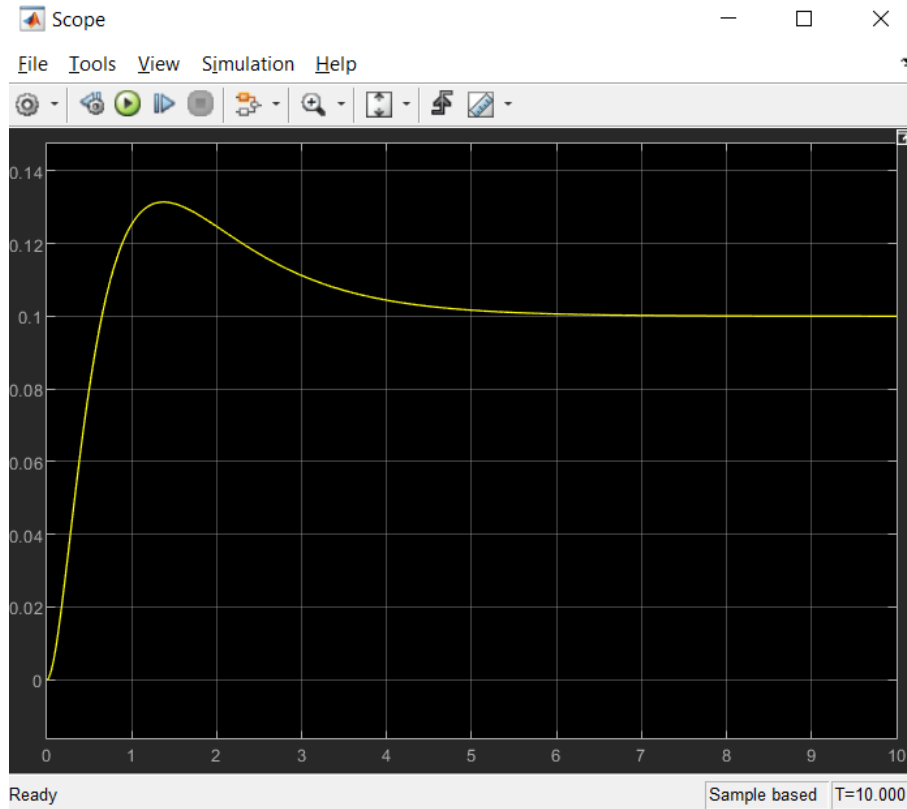
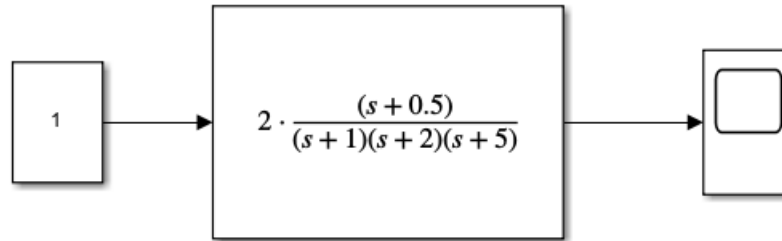
Guadagno in alta frequenza

$$F(s) = \frac{2 \cdot (s + 0.5)}{(s + 1)(s + 2)(s + 5)}$$



Visualizzare la risposta al gradino unitario.

Che tipo di curva ci aspettiamo ?



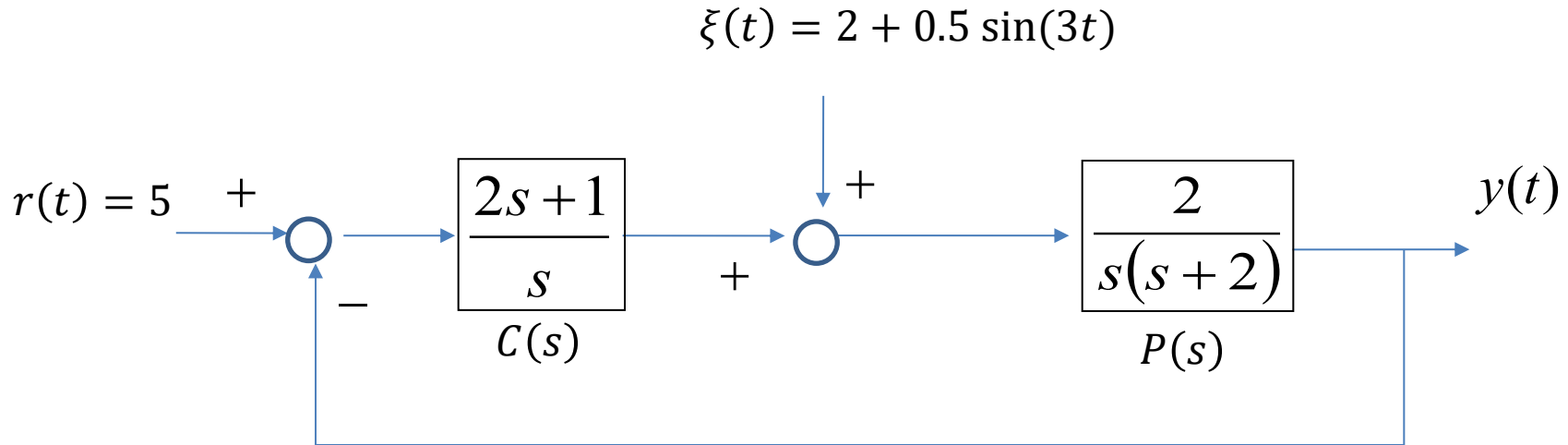
Sovraelongazione indotta dallo zero in bassa frequenza

Dopo il punto di massimo, convergenza monotona verso il regime.

Assenza di comportamenti oscillatori

Simulazione dinamica di sistemi di controllo in retroazione

Esempio 2

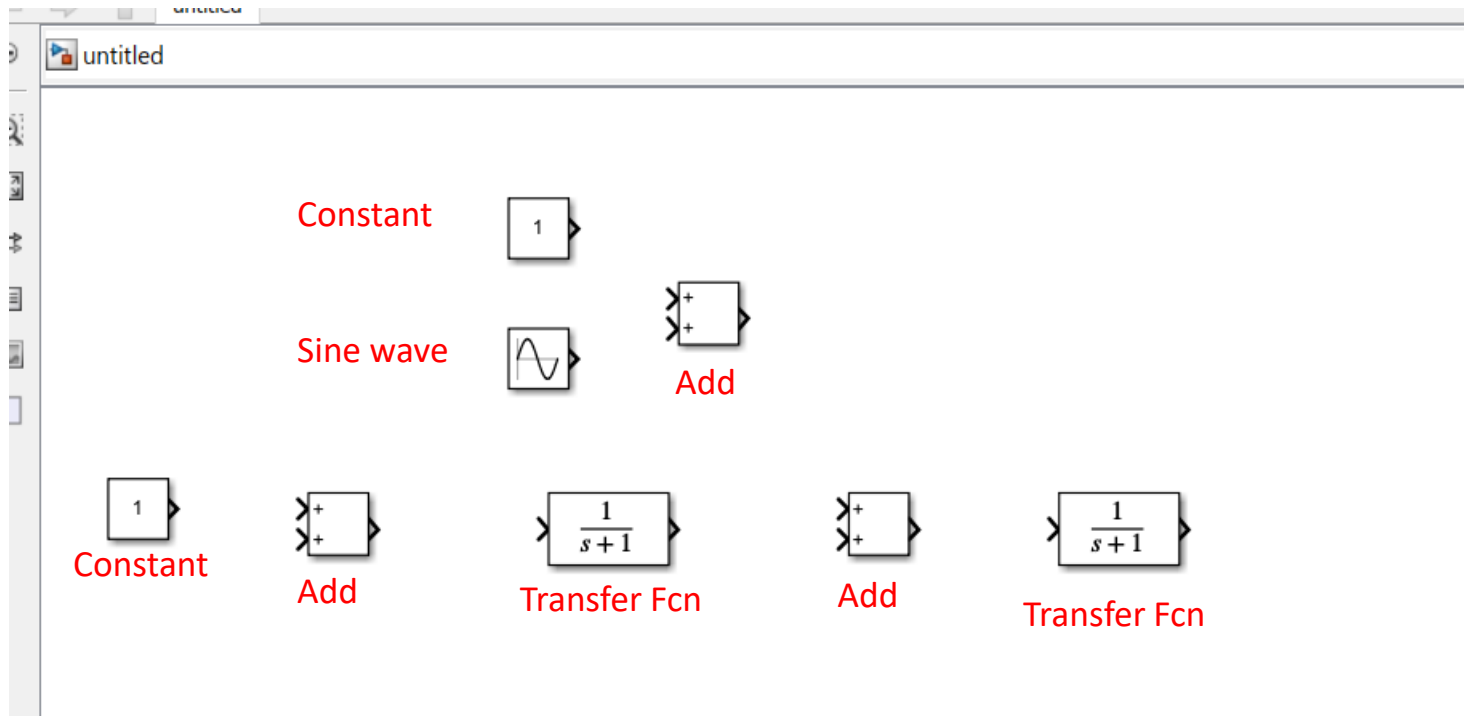


Dato il sistema di controllo in figura (analizzato a lezione nella dispensa « Comportamento a regime dei sistemi di controllo»), valutare mediante simulazione dinamica il comportamento transitorio e di regime dell'uscita.

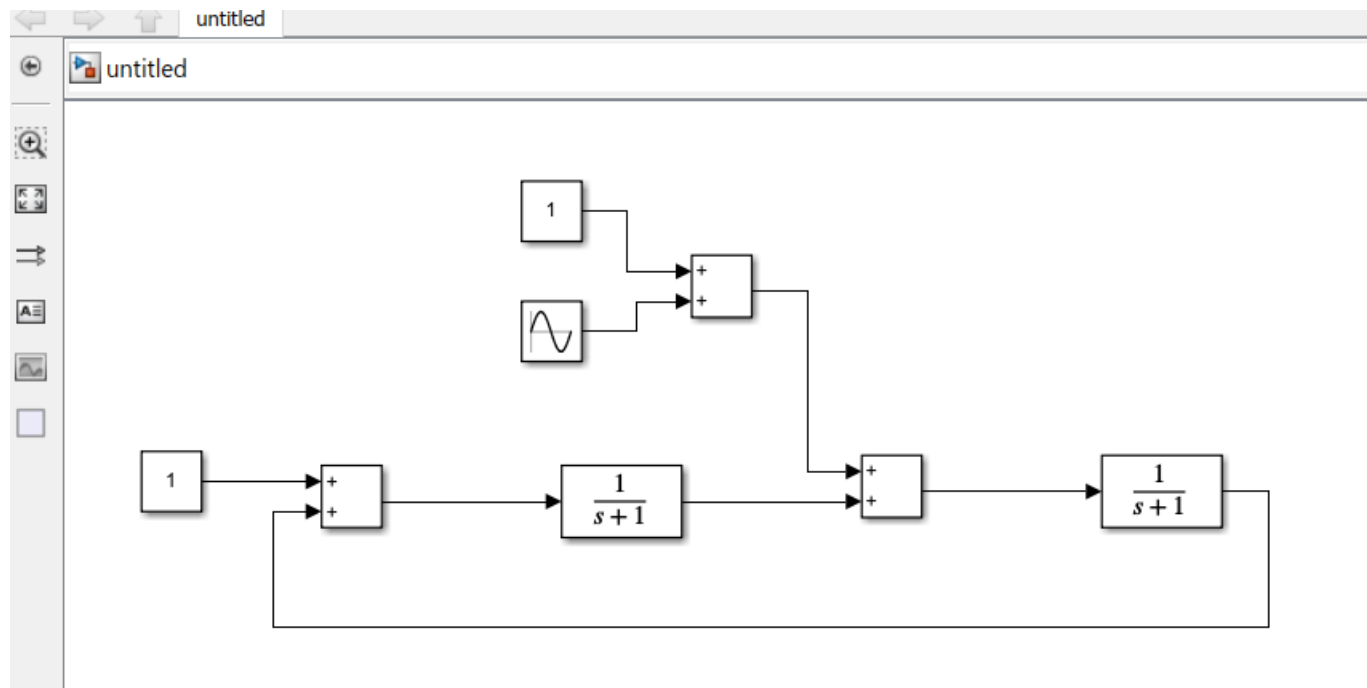
Dobbiamo realizzare nella pagina di lavoro Simulink uno schema in tutto e per tutto equivalente a quello riportato in alto. Rispetto all'esempio precedente serviranno dei blocchi elementari aggiuntivi: due blocchi «Transfer Function», vari nodi sommatori (blocco «Add»), due blocchi «constant» per generare le componenti costanti del set point e del disturbo ed un blocco «Sine wave» per generare la componente sinusoidale del disturbo.

Dopo avere aperto, con le medesime modalità impiegate in precedenza, un modello Simulink «in bianco», vediamo un modo alternativo per importare nello schema i blocchi elementari che ci servono. E' sufficiente fare doppio click in un punto qualsiasi dello schema, e scrivere le prime lettere del nome del blocco affinché venga generata una lista che include il blocco cercato, che può quindi essere selezionato ed importato nel modello.

Importare con la modalità suggerita i blocchi elementari menzionati nella slide precedente, e disporli come nella figura sottostante

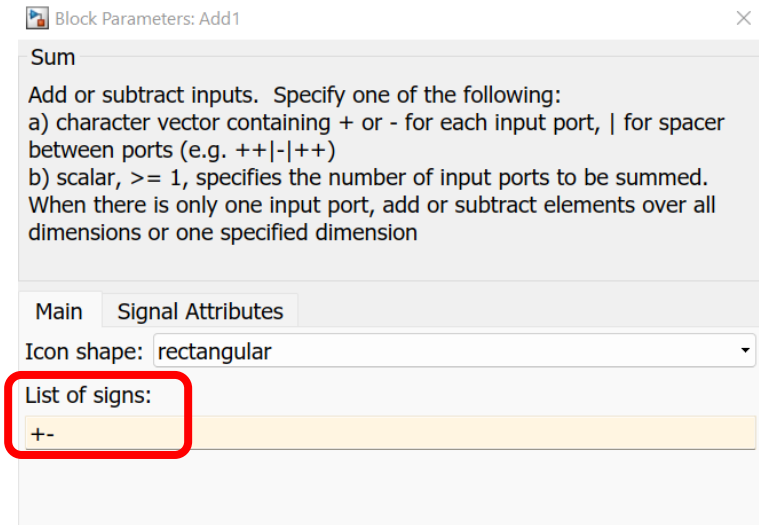
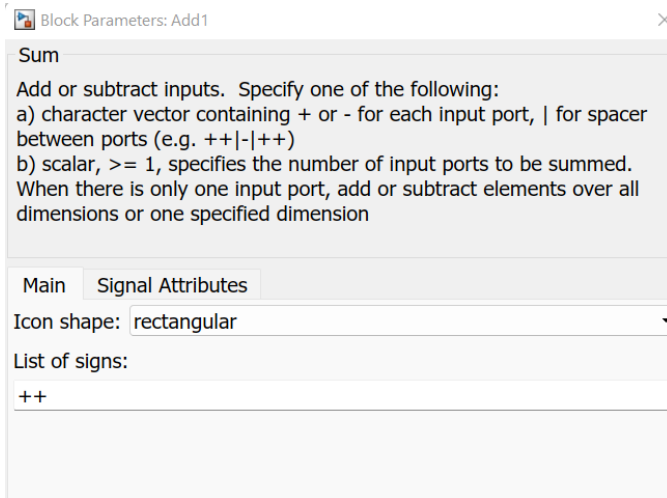


Realizziamo le connessioni come in figura

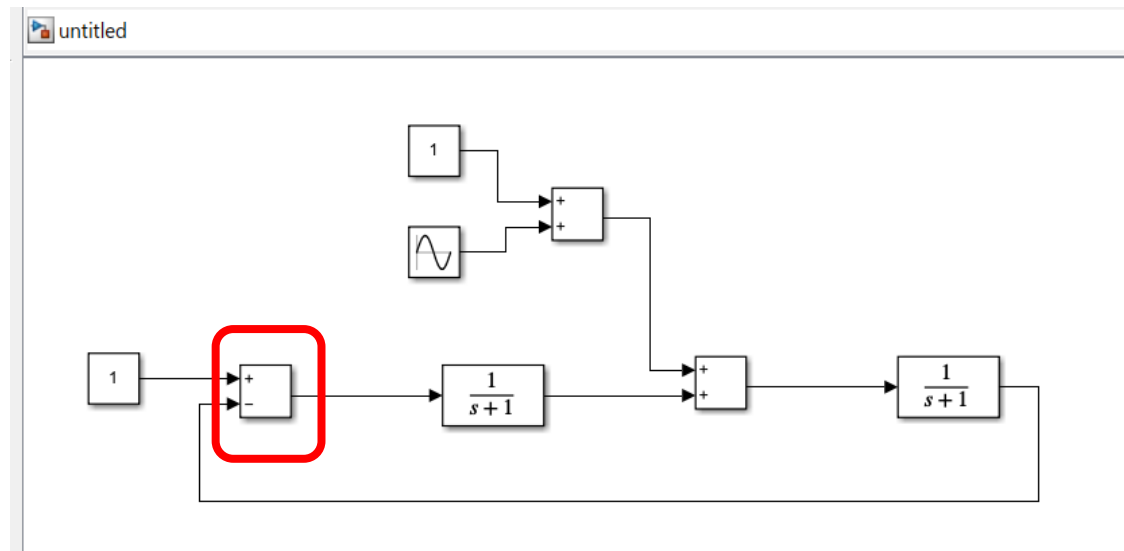


Ora vanno parametrizzati i vari blocchetti elementari. Ciascun blocco, come visto in precedenza, si parametrizza facendo doppio click su di esso ed accedendo alla relativa maschera di parametrizzazione.

Iniziamo con il modificare il nodo sommatore in cui deve essere eseguita la differenza fra il set-point e l'uscita. Un nodo sommatore si parametrizza attraverso una stringa che contiene dei '+' e dei '-'. A ciascun '+' corrisponde un terminale di ingresso che somma, a ciascun '-' un terminale che sottrae.



Cambia l'aspetto del nodo sommatore, e compare il terminale negativo



Ora parametrizziamo i due blocchi «Constant» ed i due blocchi «Transfer Fcn»

$$C(s) = \frac{2s + 1}{s}$$

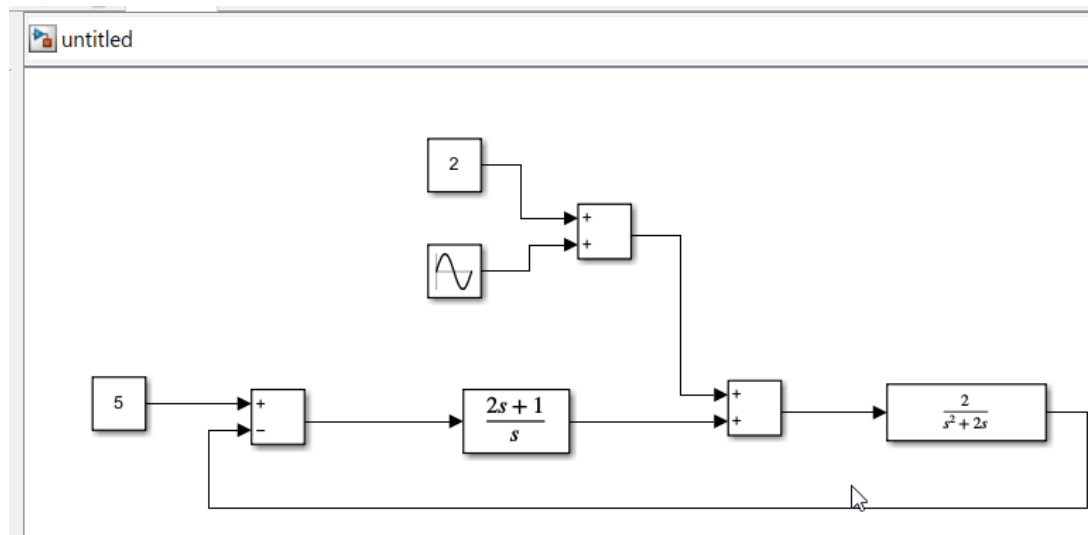


Parameters	
Numerator coefficients:	
[2	1]
Denominator coefficients:	
[1	0]

$$P(s) = \frac{2}{s(s + 2)} = \frac{2}{s^2 + 2s}$$



Parameters	
Numerator coefficients:	
[2]	
Denominator coefficients:	
[1	2 0]



Ora dobbiamo parametrizzare il blocco **Sine Wave** affinché generi il segnale $0.5 \sin(3t)$
Mediante il blocco Sine Wave è possibile generare un segnale avente la forma seguente:

$$A \sin(\omega t + \phi) + B$$

Amplitude Frequency Phase Bias

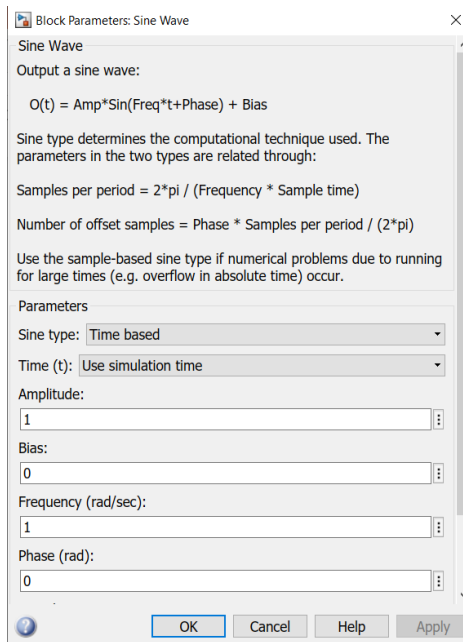
$A = \text{Amplitude} = 0.5$

$\omega = \text{Frequency} = 3$

$\phi = \text{Phase} = 0$

$B = \text{Bias} = 0$

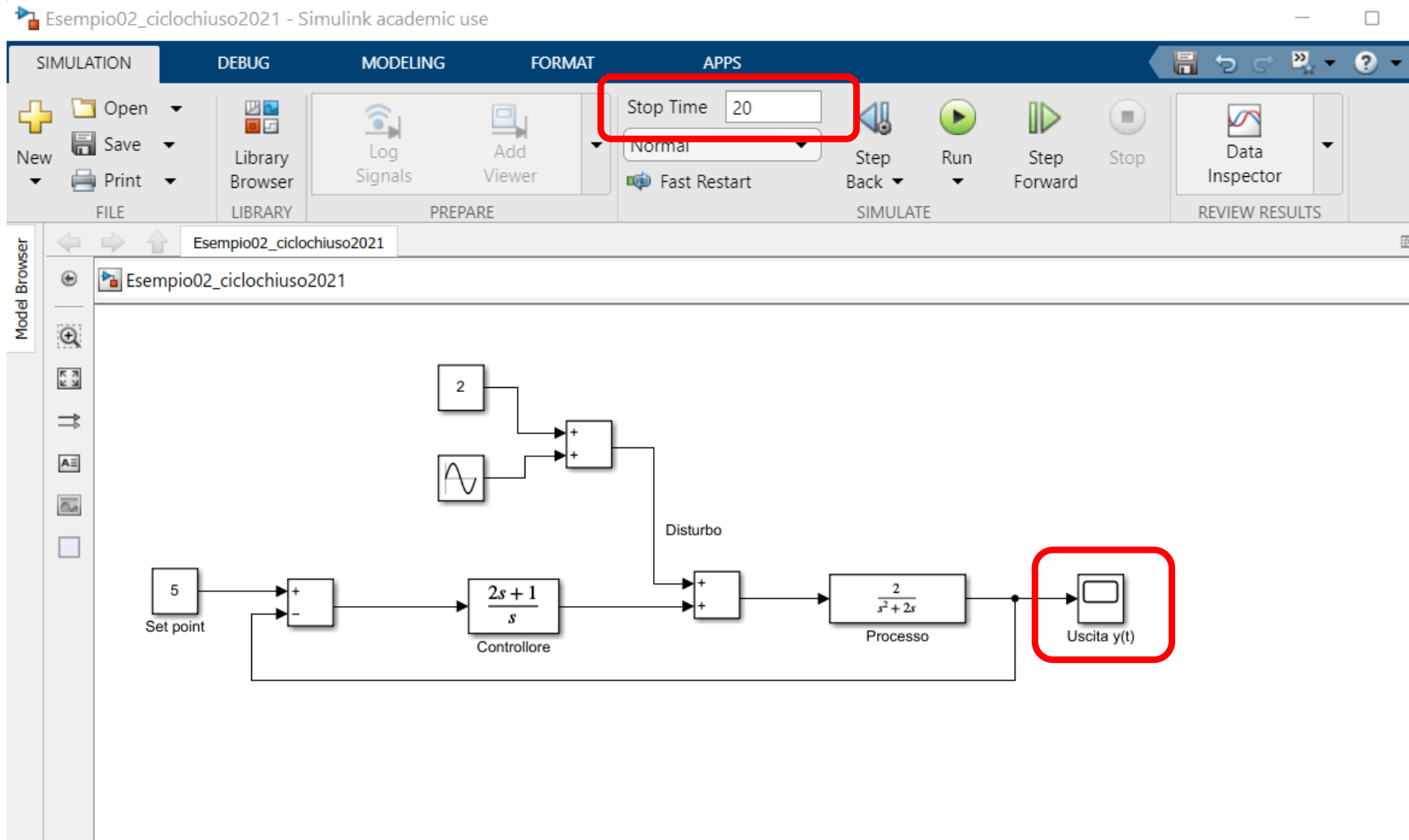
Finestra di parametrizzazione (i valori di default producono in uscita il segnale $\sin(t)$)



Per generare il segnale $0.5 \sin(3t)$ si parametrizzi il blocco inserendo i valori soprariportati (v. sotto)

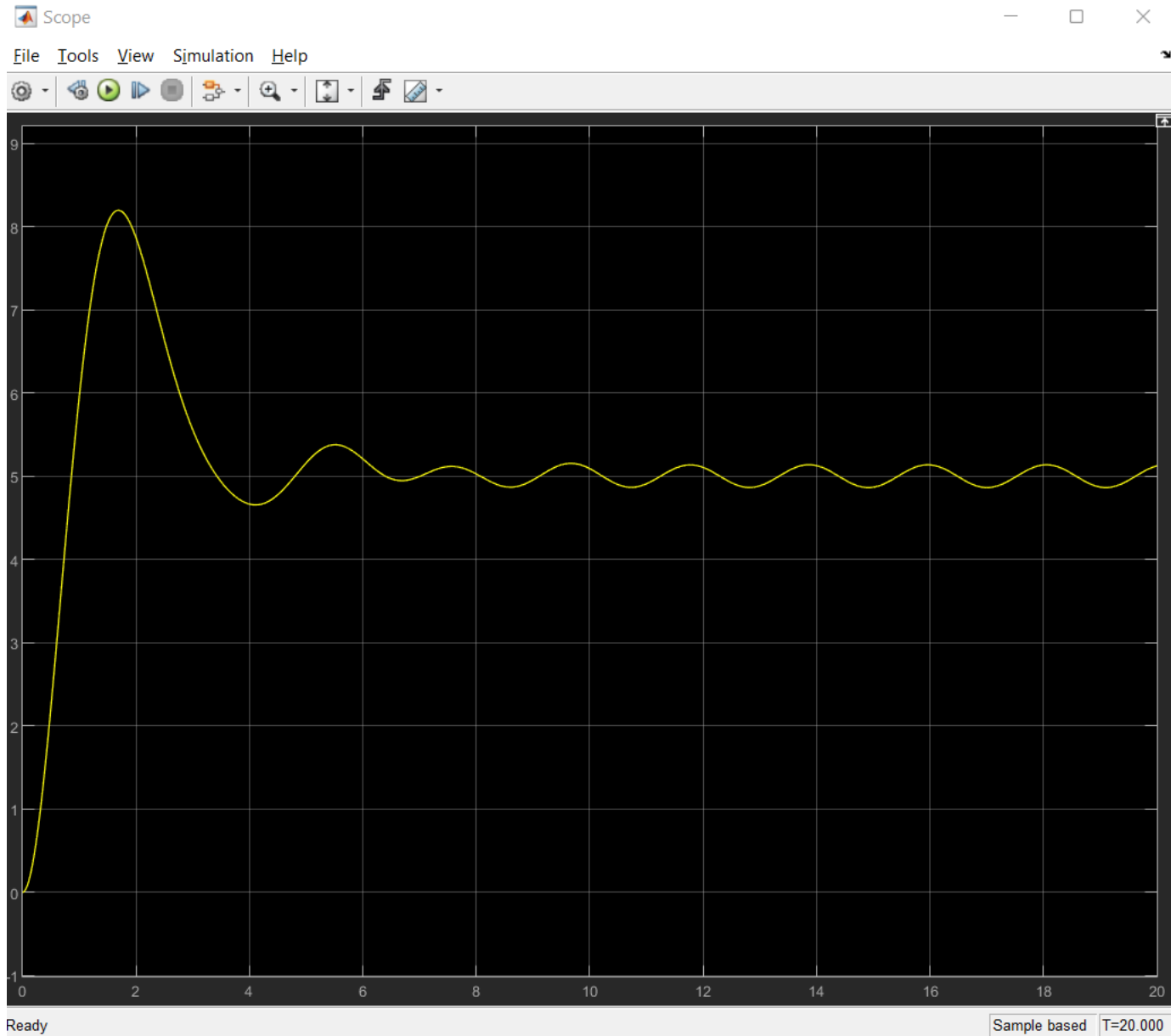
Parameters	
Sine type:	Time based
Time (t):	Use simulation time
Amplitude:	0.5
Bias:	0
Frequency (rad/sec):	3
Phase (rad):	0

Inseriamo il blocco Scope e colleghiamoci in ingresso l'uscita del sistema, impostiamo la durata della simulazione a **20 secondi**, modifichiamo i parametri del solutore numerico come fatto nell'esempio precedente. Siamo ora pronti a mandare in run la simulazione

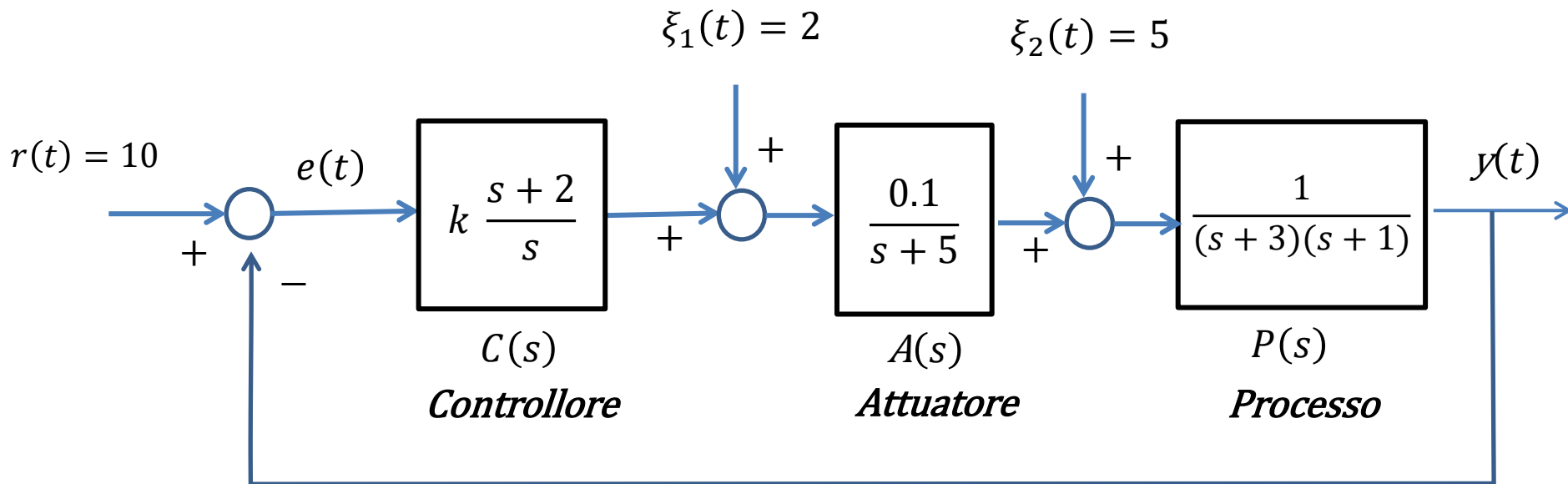


FILE: Esempio02.slx

Evoluzione temporale dell'uscita



Esempio 3 Si consideri il seguente sistema di controllo in retroazione

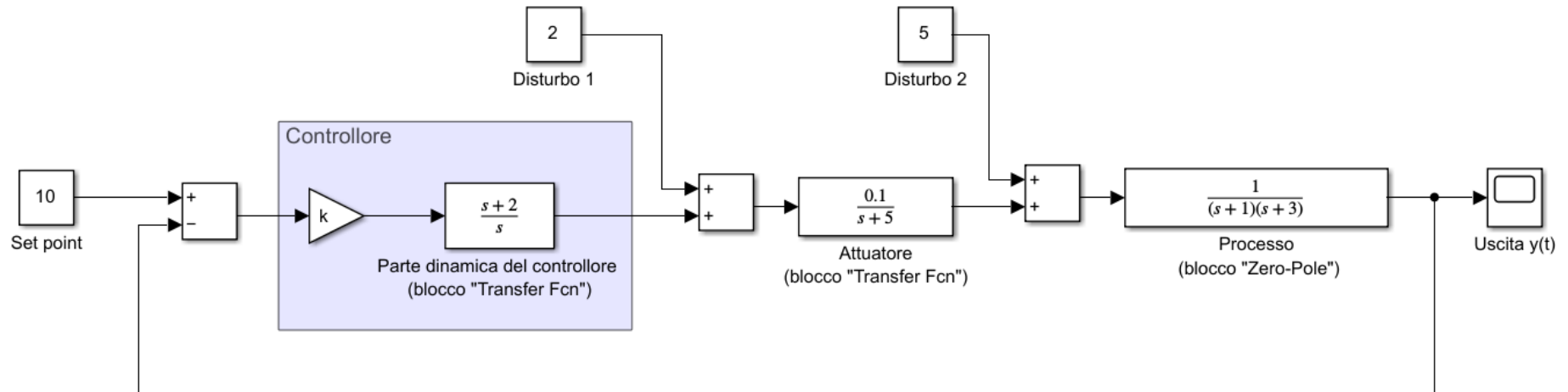


Verificare mediante simulazione dinamica le seguenti deduzioni relativamente alla evoluzione di regime della variabile di uscita al variare del guadagno k

$$y(t) \rightarrow 10 \quad 0 < k < 616.87$$

$$y(t) \text{ oscilla} \quad k = 616.87$$

$$y(t) \rightarrow \infty \quad k > 616.87$$



FILE: Esemplio03.slx

La realizzazione di questo modello Simulink non presenta particolari elementi di novità rispetto al precedente esempio.