

# INTRODUZIONE al Control System Toolbox di MATLAB e **SIMULINK**

*Corso di Sistemi di Controllo Digitale*

# Organizzazione delle Lezioni

- INTRODUZIONE A MATLAB
- DEFINIZIONE DI VARIABILI, MATRICI E VETTORI
- FUNZIONI ELEMENTARI PER SCALARI E MATRICI
- POLINOMI
- VISUALIZZAZIONE DI GRAFICI
- NOTE GENERALI SU MATLAB (help, istruzione di ambiente...)
- ALCUNE ISTRUZIONI PER APPLICAZIONI DI CONTROLLO
- SIMULINK

# Descrizione di un Sistema Lineare

- Matlab consente di descrivere sistemi lineari a tempo continuo, a segnali campionati e a tempo discreto.
- Nel caso di sistemi a segnali campionati occorre precisare la variabile  $T_s$  (tempo di campionamento) . Per i sistemi a tempo discreto si pone  $T_s = 1$ .
- Per descrivere un sistema dato in termini di funzione di trasferimento, occorre:
  - sistema a tempo continuo:  

```
>> sistema = tf (num,den);
```

dove *num* e *den* sono vettori con cui si esprimono i polinomi numeratore e denominatore in potenze decrescenti di *s*.  
Questo comando darà in output l'espressione della funzione di trasferimento del sistema a tempo continuo.
  - sistema a tempo discreto:  

```
>> sistema = tf (num,den,1);
```

dove *num* e *den* sono vettori con cui si esprimono i polinomi numeratore e denominatore in potenze decrescenti di *z* e 1 indica che il tempo di campionamento  $T_s$  è 1.  
Questo comando darà in output l'espressione della funzione di trasferimento del sistema a tempo discreto.

# Descrizione di un Sistema Lineare

- Supponiamo, per esempio, di voler descrivere un sistema dinamico a tempo continuo con funzione di trasferimento

$$G(s) = \frac{s + 1/3}{s^2 - 1/12s - 1/12}$$

- Con i comandi: `>> num = [1 1/3];`

`>> den = [1 -1/12 -1/12];`

si definiscono i polinomi a numeratore e denominatore.

Con il comando: `>> sistema = tf(num, den)` si ottiene in output

Transfer function:

s + 0.3333

-----

s^2 - 0.08333 s - 0.08333

# Descrizione di un Sistema Lineare

- Per definire, invece, il sistema a tempo discreto con funzione di trasferimento

$$G(z) = \frac{z + 1/3}{z^2 - 1/12 z - 1/12}$$

resteranno invariati i polinomi che descrivono il numeratore ed il denominatore usati in precedenza. Ciò che varia è il comando finale per definire il sistema, che ora diventa: `>> sistema = tf (num, den, 1)`.

- Tale comando produce come output:

$$\begin{array}{r} \textit{Transfer function:} \\ z + 0.3333 \\ \hline z^2 - 0.08333 z - 0.08333 \end{array}$$

# Istruzioni utili per il Tracciamento delle Risposte di un Sistema Lineare

Dopo aver descritto il sistema lineare in esame nel modo opportuno, a seconda che sia a tempo discreto o continuo, Matlab offre alcune funzioni per valutare la risposta di tale sistema a diversi tipi di ingresso.

1. Risposta impulsiva: con il comando `>> impulse (sistema, t)` Matlab calcola e grafica la risposta all'impulso del sistema;  $t$  è il vettore che definisce il tempo di durata della simulazione.
2. Risposta a scalino: con il comando `>> step (sistema, t)` Matlab calcola e grafica la risposta allo scalino del sistema;  $t$  è il vettore che definisce il tempo di durata della simulazione.
3. Risposta ad un ingresso generico: con il comando `>> lsim (num, den, u, t)` Matlab calcola e grafica la risposta del sistema all'ingresso, generico,  $u$ ;  $t$  è il vettore che definisce il tempo di durata della simulazione.

# Istruzioni utili per il Tracciamento dei Diagrammi della Risposta in Frequenza

- Matlab offre anche la possibilità di analizzare sistemi dinamici lineari a tempo continuo e discreto nel dominio della frequenza:
  1. Diagramma di Bode: con il comando `>> bode (sistema)` Matlab visualizza automaticamente il diagramma di modulo e fase della funzione di trasferimento del sistema in esame.
  2. Diagramma di Nyquist: con il comando `>> nyquist (sistema,w)` Matlab traccia il diagramma di Nyquist della funzione di trasferimento del sistema in esame in corrispondenza delle pulsazioni specificate dal vettore `w`.
- Spesso è utile, per analizzare le caratteristiche di un sistema lineare a tempo continuo o discreto, analizzarne il luogo delle radici.
- Con il comando `>> rlocus (sistema)` Matlab traccia il luogo delle radici del sistema in esame.

# Diagrammi di Bode per Sistemi a Tempo Discreto

- Vediamo ora in dettaglio come ottenere il diagramma di Bode di un sistema a tempo discreto. Ad esempio: per rappresentare in Matlab la funzione di trasferimento

$$G(z) = \frac{z - \frac{1}{2}}{z^2 - \frac{5}{6}z + \frac{1}{6}}$$

si useranno i comandi: `>> num = [1 -1/2];`  
`>> den = [1 -5/6 1/6];`  
`>> sistema = tf(num,den,1);`

ottenendo in output: *Transfer function:*  
$$\frac{z - 0.5}{z^2 - 0.8333 z + 0.1667}$$



# Diagrammi di Bode per Sistemi a Tempo Discreto

- Con il comando `>> bode (sistema)`

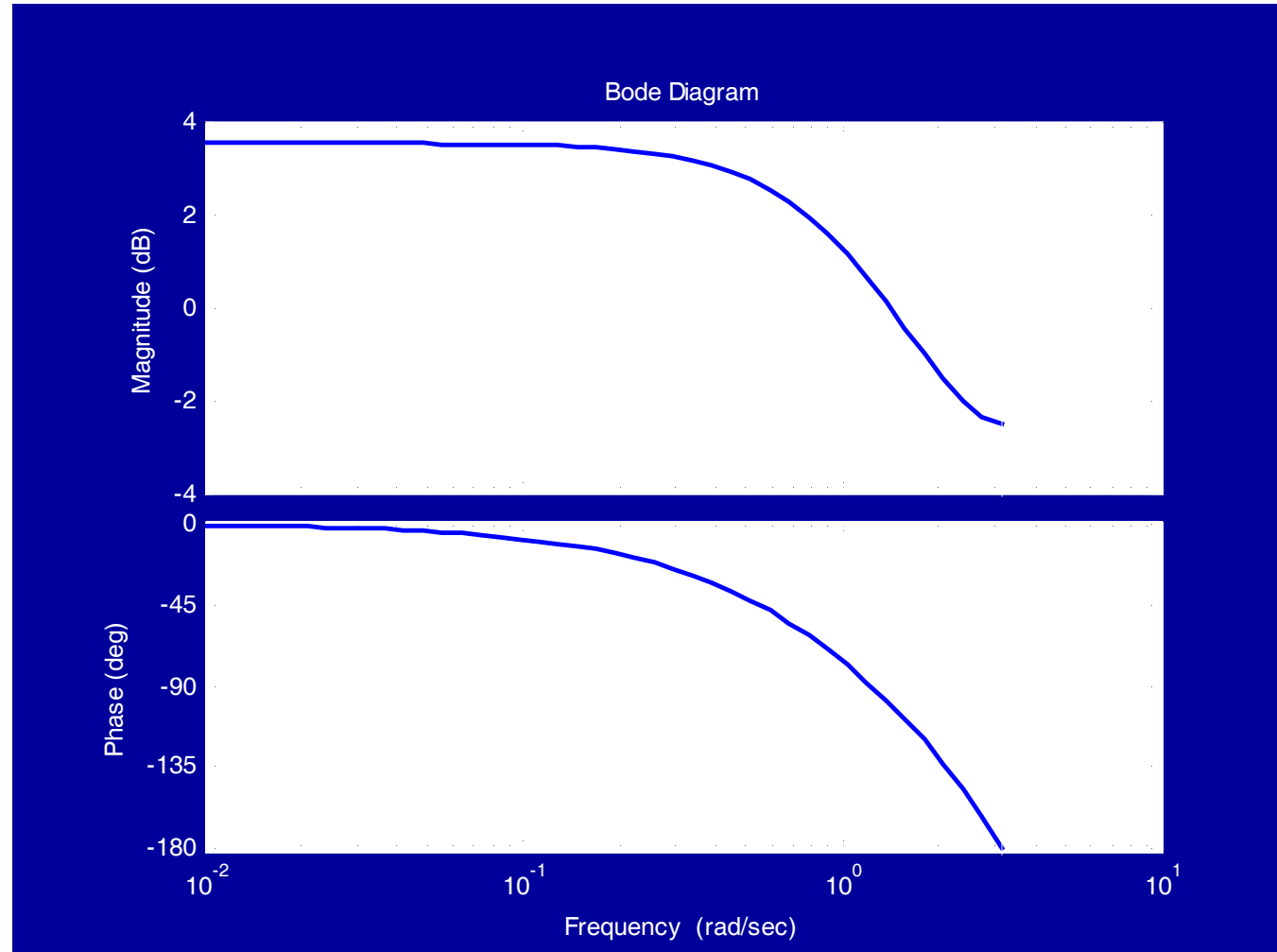
si ottiene il diagramma di Bode di modulo e fase per il sistema a tempo discreto, calcolato come

$$G(e^{j\omega})$$

Per l'esempio si ottiene il grafico mostrato.

Si noti che il diagramma è tracciato fino alla pulsazione  $p$ .

La ragione è che  $G(e^{j\omega})$  è una funzione periodica di periodo  $2p$ .



# Diagrammi di Bode per Sistemi a tempo Discreto: Filtro Passa - Tutto

- Supponiamo ora di voler tracciare il diagramma di Bode di un filtro passa – tutto con funzione di trasferimento:

$$G(z) = \frac{z + 4}{z + 1/4}$$

- Per definire il sistema a tempo discreto avente  $G(z)$  come funzione di trasferimento e tracciare il relativo diagramma di Bode si usano i comandi:

```
>>num = [1 4];
```

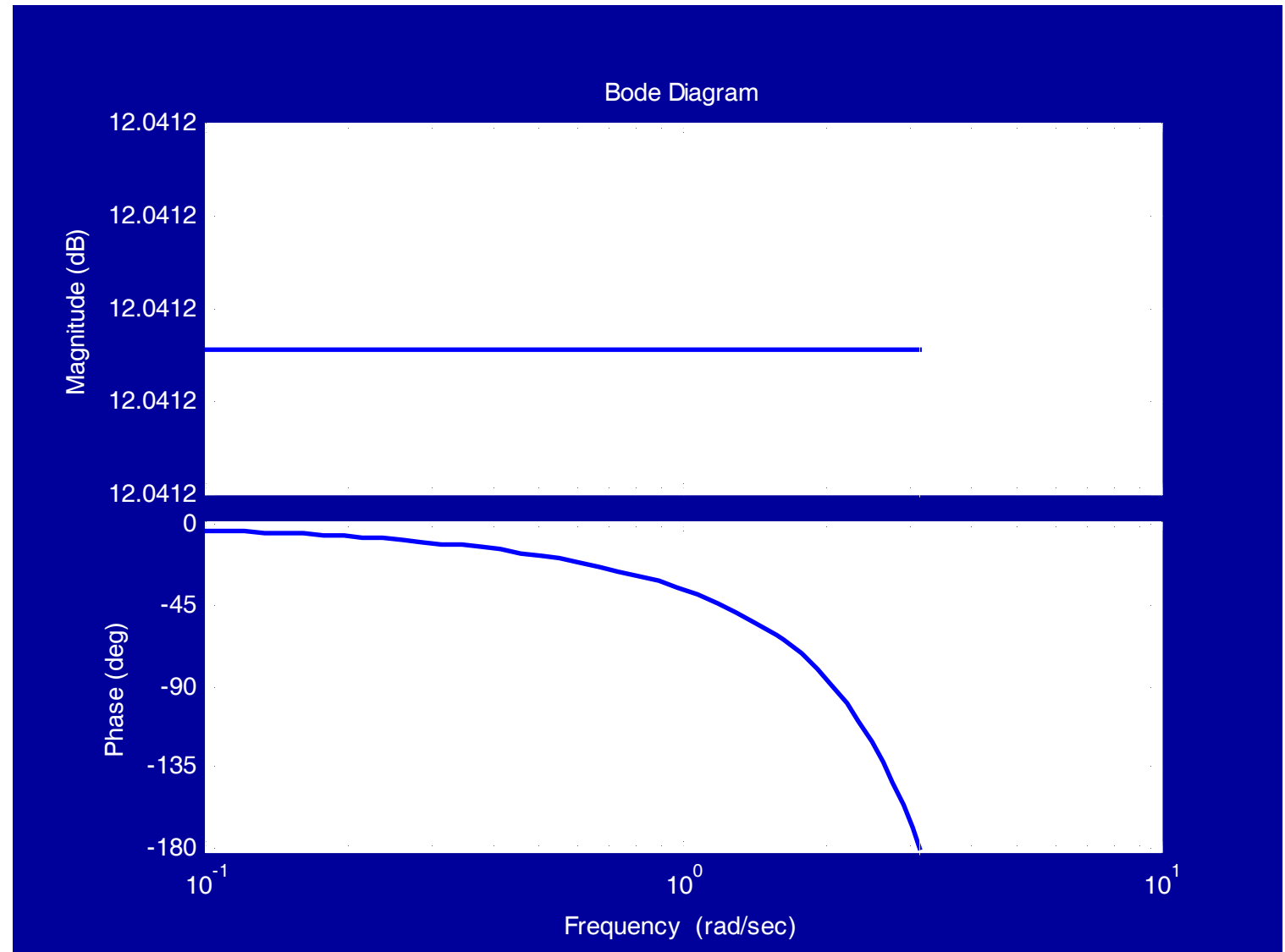
```
>>den = [1 1/4];
```

```
>>sistema = tf(num,den,1);
```

```
>>bode (sistema)
```

# Diagrammi di Bode per Sistemi a Tempo Discreto: Filtro Passa - Tutto

La figura mostra il diagramma di Bode di modulo e fase del filtro passa tutto precedentemente introdotto.



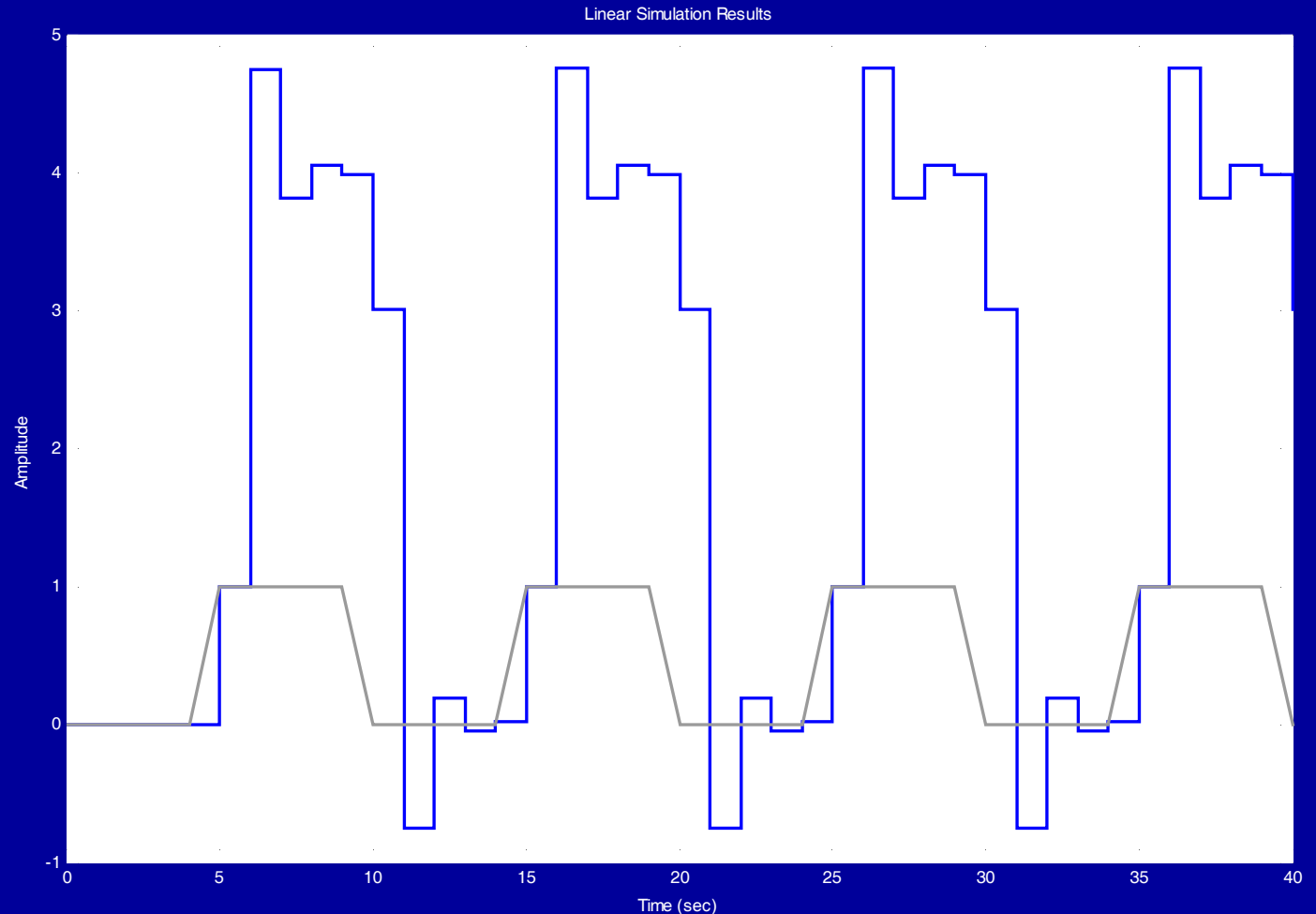
# Risposta ad un Ingresso Generico di Sistemi a Tempo Discreto: Filtro Passa - Tutto

- Per completare l'esempio di analisi del filtro passa – tutto introdotto precedentemente, valutiamo la risposta di tale sistema ad un segnale di ingresso descritto come un'onda quadra.
- Il primo passo da compiere è creare il vettore di ingresso desiderato:
  - Il comando: `>> [u,t] = gensig('square',10,40,1)` consente di creare il vettore degli ingressi  $u$  che, in questo caso, è un'onda quadra ('square'), di periodo 10, durata 40 e "tempo di campionamento"  $T_s = 1$  (coerentemente col fatto che il sistema da simulare è a tempo discreto). Il vettore  $t$  è il vettore dei tempi che si userà per la simulazione.
  - Utilizzando la funzione `lsim` precedentemente introdotta, è possibile simulare e visualizzare su un grafico la risposta del sistema a tale ingresso.
  - Il comando `>> lsim(sistema,u,t)` consente di simulare la risposta del sistema che descrive il filtro passa – tutto all'onda quadra in ingresso  $u$ .

# Risposta ad un Ingresso Generico di Sistemi a Tempo Discreto: Filtro Passa - Tutto

La figura mostra la risposta del filtro passa tutto all'ingresso  $u$ .

Il segnale in ingresso è visualizzato in colore grigio, mentre l'uscita in blu.

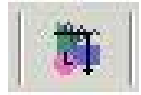


# Simulink

■ Simulink è un ambiente grafico che consente di descrivere e simulare sistemi complessi. La simulazione di un sistema dinamico con Simulink consiste in due fasi fondamentali:

1. La prima fase consiste nella creazione di un modello grafico del sistema da simulare, che può consistere nell'interconnessione di blocchi continui e discreti, lineari e non lineari. Questo modello descriverà le relazioni matematiche esistenti tra gli ingressi e le uscite del sistema.
2. La seconda fase consiste nell'utilizzare l'ambiente Simulink per simulare il comportamento del sistema durante una sua evoluzione temporale su un arco di tempo stabilito dall'utente. Di fatto Simulink utilizza le informazioni contenute nel modello grafico, che l'utente deve specificare nella fase di descrizione, per generare le equazioni dinamiche del modello e risolvere numericamente il problema di simulazione in esame.

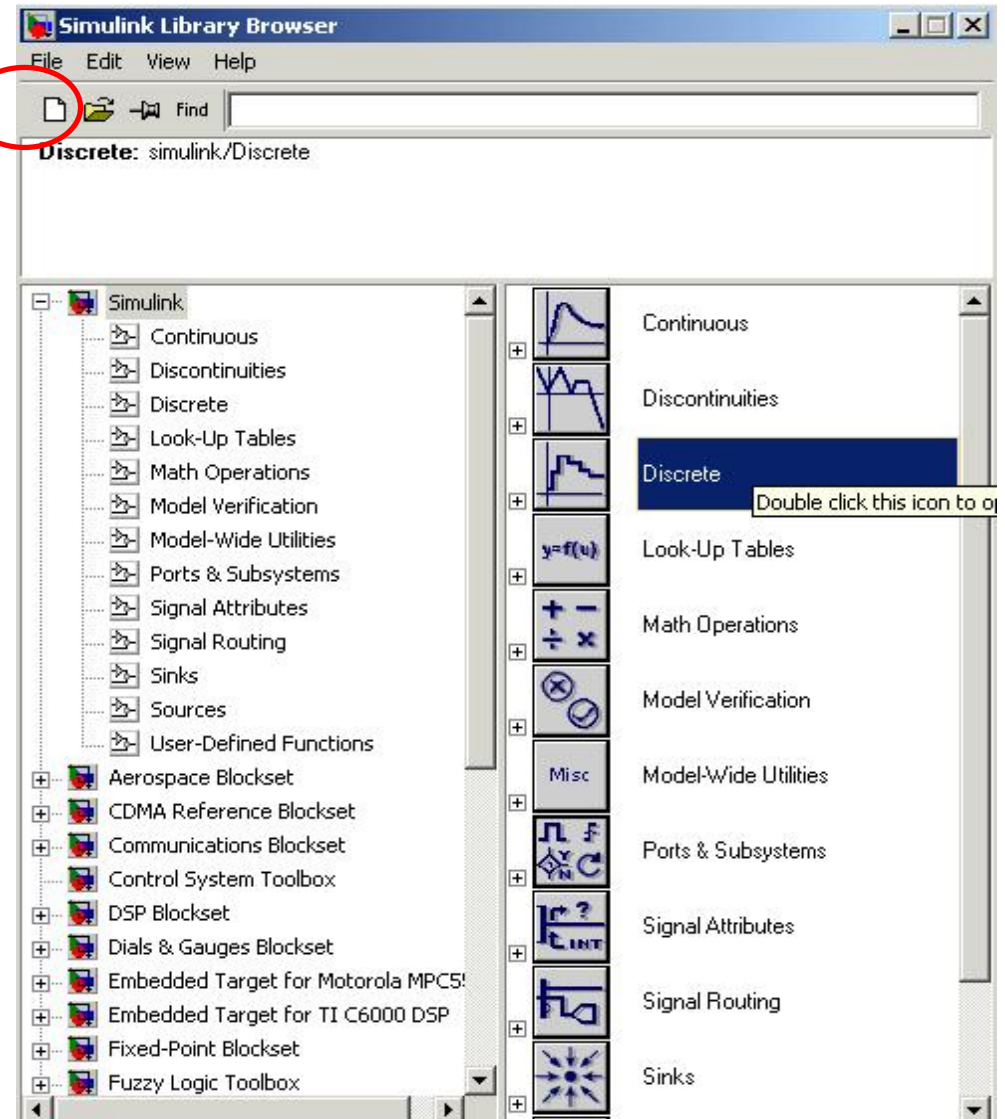
# Simulink

- Simulink interagisce direttamente con il Workspace di Matlab. I modelli descritti, pertanto, potranno contenere variabili contenute nel Workspace della sessione corrente.
- Allo stesso modo, i risultati della fase di simulazione possono essere passati direttamente al Workspace sotto forma di nuove variabili, pronte per essere analizzate.
- Per aprire l'ambiente Simulink si può, alternativamente, digitare `>> simulink` nel prompt, oppure cliccare sull'icona  presente nel toolbar di Matlab.

# Simulink

Una volta aperto Simulink è disponibile la libreria dei modelli mostrata in figura, che consente di scegliere i blocchi predefiniti per la descrizione del sistema desiderato.

Cliccando su questa icona si apre un nuovo file in cui disegnare il modello, trascinando nel file i modelli base che sono contenuti nelle varie librerie.



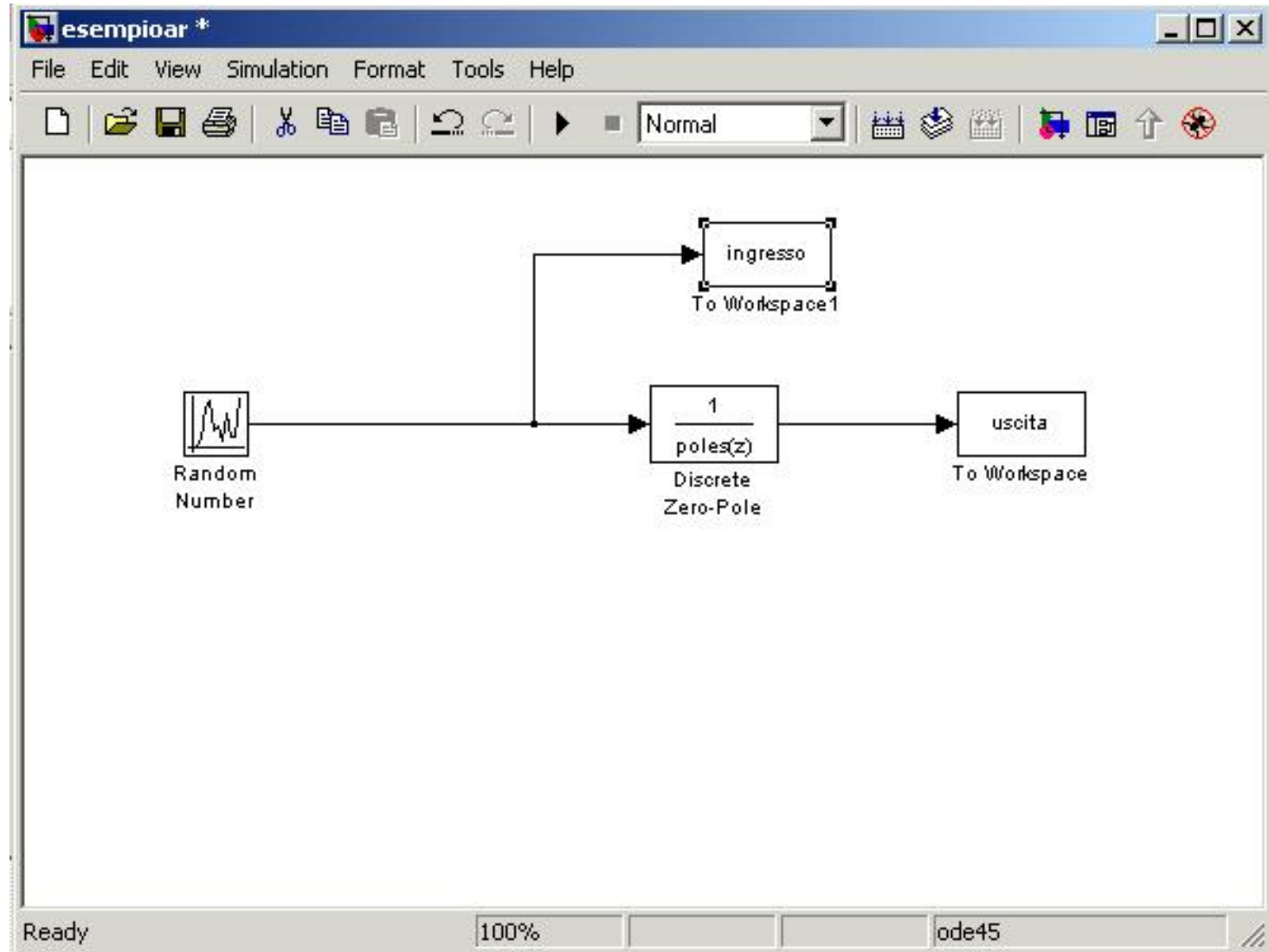


# Simulink: Esempio

- Supponiamo di voler simulare con Simulink un sistema AR(3), asintoticamente stabile, i cui poli siano  $z_i = 0.2, 0.5, 0.3$ ,  $i=1,2,3$  alimentato da un rumore bianco  $e(t) \approx WN(0,1)$
- Per far questo dovremo utilizzare, per descrivere il segnale di ingresso il blocco *Random Number* della libreria *Sources*. Per la descrizione della funzione di trasferimento useremo il blocco *Discrete Zero-Pole* della libreria *Discrete*, che consente di descrivere la funzione di trasferimento specificandone poli e zeri.
- Infine, per poter analizzare l'output ottenuto via simulazione, utilizzeremo due blocchi *To Workspace* della libreria *Sinks*, che consentono di avere disponibili nel Workspace i risultati della simulazione per fare le analisi desiderate.

# Simulink: Esempio

Trascinando nel file i blocchi descritti precedentemente ed interconnettendoli opportunamente, si ottiene il modello in figura. Il passo successivo da compiere, prima di eseguire la simulazione, è quello di specificare i parametri dei vari blocchi perché descrivano il sistema desiderato.



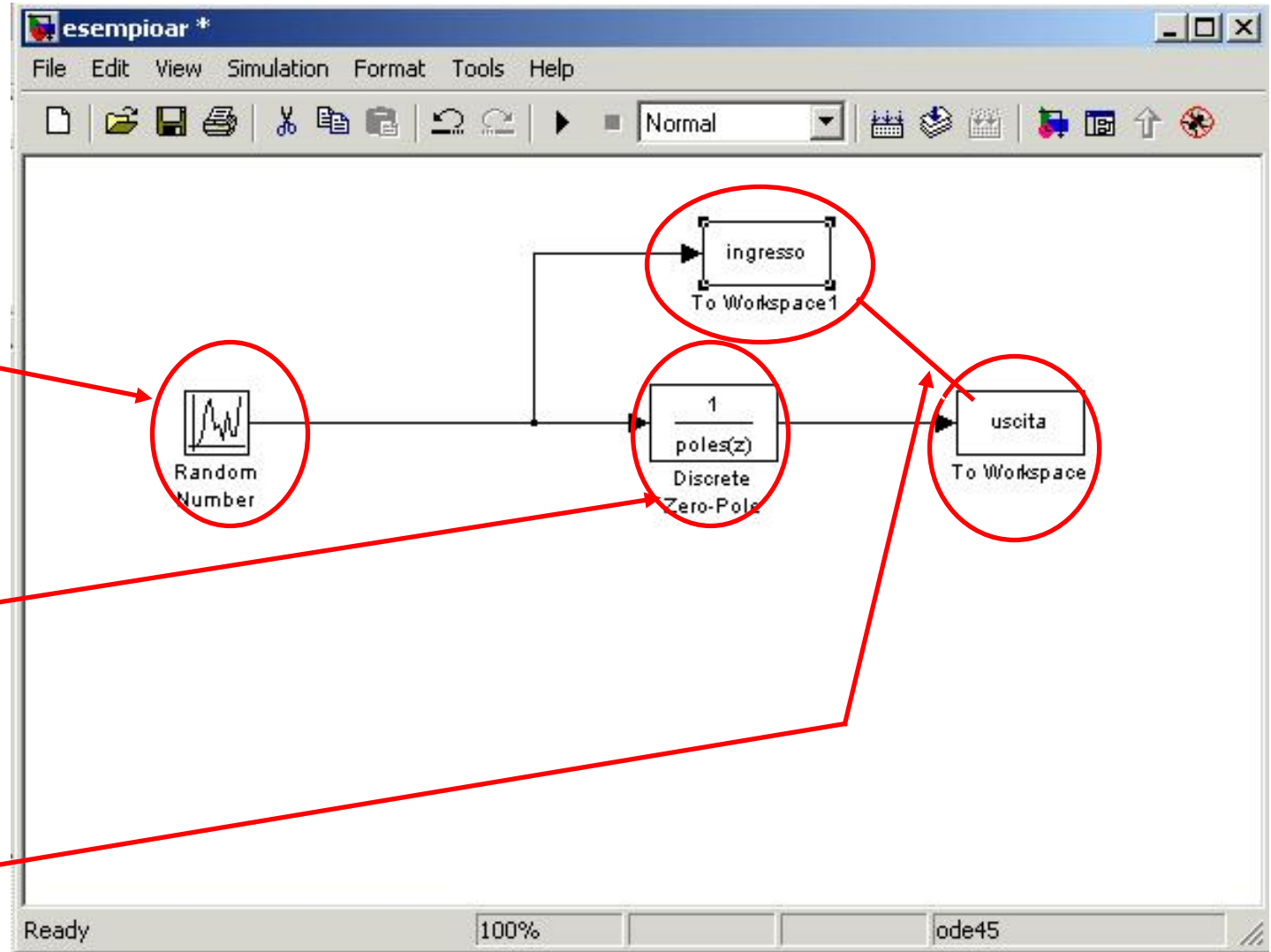
# Simulink: Esempio

Facendo doppio clic sui blocchi, si aprono le finestre di dialogo in cui inserire i vari parametri

Specificare media e varianza del segnale di ingresso.

Specificare zeri e i poli della funzione di trasferimento

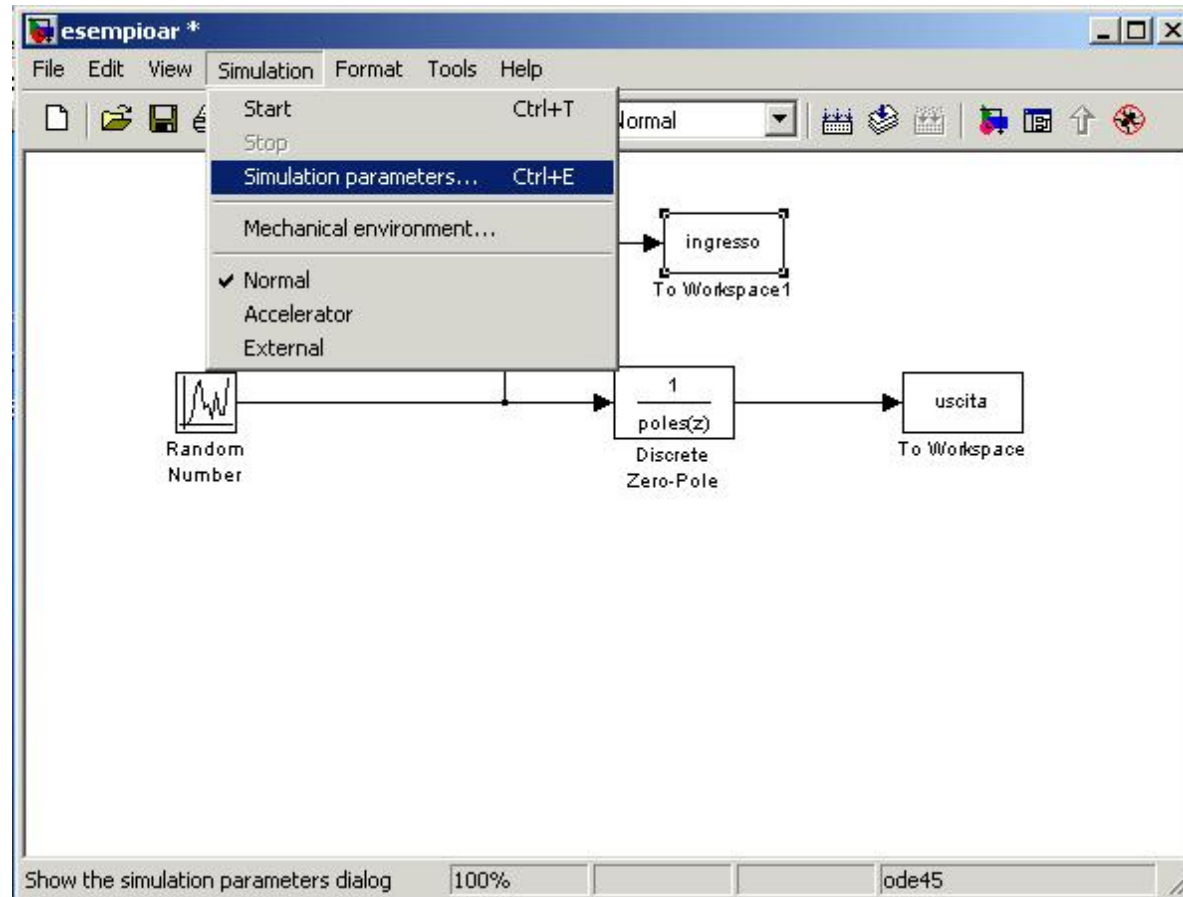
Specificare il nome della variabile in cui salvare i dati ed il formato in cui salvarla



# Simulink: Esempio

- Prima di lanciare la simulazione, bisogna fare un ultimo passo: fissare i parametri della simulazione stessa.

Selezionando la voce **Simulation** del menu a tendina, compaiono le voci che si vedono in figura. Per impostare i parametri della simulazione selezionare la voce **Simulation Parameters**.




# Simulink: Esempio

In questa finestra l'utente deve specificare

1. gli istanti di inizio e fine simulazione;
2. Il metodo numerico da utilizzare per la soluzione delle equazioni (se non ci sono esigenze particolari va bene quello di default);
3. I parametri del solutore numerico (nella maggior parte dei casi vanno bene quelli di default).



# Simulink: Esempio

- A questo punto, cliccando sull'icona  del toolbar di Simulink si lancia la simulazione. Quando questa termina, digitando nel Worskspace, ad esempio, il comando `>>plot(uscita)`, si ottiene il grafico che mostra l'andamento nel tempo dell'uscita del processo ottenuta via simulazione.

