

Politecnico di Bari
Facoltà di Ingegneria

Dispensa per il Corso di
Controlli Automatici II

Introduzione all'uso di "Simulink"

Ing. Mariagrazia Dotoli

Introduzione all'uso di "Simulink"

Il programma di calcolo MATLAB è corredato da diversi toolbox, ossia pacchetti specializzati che includono routine per applicazioni specifiche. Tra questi toolbox, Simulink realizza una interfaccia grafica per modellare sistemi fisici attraverso l'uso di moduli predefiniti rappresentati da icone.

Questa dispensa costituisce una introduzione all'utilizzo degli strumenti e comandi di base di Simulink 2.0, per l'analisi delle prestazioni e la progettazione dei sistemi di controllo automatico. Per una conoscenza più approfondita si rimanda ai manuali di riferimento del software ed a testi specifici.

Simulink è un ambiente per la modellazione, l'analisi e la simulazione di sistemi matematici e fisici. Offre un'interfaccia grafica per la costruzione di modelli tramite diagrammi a blocchi, prelevabili da una vastissima libreria di blocchi predefiniti. Ciò permette di modellare un sistema rapidamente, con chiarezza e senza il bisogno di scrivere righe di codice. Simulink consente di costruire modelli con diagrammi a blocchi tramite operazioni 'click-and-drag', modificare velocemente i parametri dei modelli e visualizzare i risultati 'live' durante la simulazione.

Simulink è un sistema aperto, che permette di scegliere, adattare e creare componenti software e hardware per soddisfare le proprie esigenze. Oltre alle potenzialità grafiche di MATLAB, le funzionalità di animazione migliorano la visualizzazione e offrono una visione più approfondita del comportamento del sistema col progredire della simulazione.

1. Esempio di costruzione di sistema a blocchi

Per utilizzare Simulink è necessario anzitutto aprire il Simulink Library Browser (figura 1), la libreria contenente tutti i blocchi base da utilizzare nella costruzione di un sistema a blocchi, o spingendo il corrispondente pulsante sulla barra degli strumenti di MATLAB, oppure digitando `simulink3` dalla Command Window (`simulink` nelle vecchie versioni di Simulink). È poi necessario aprire un file di Simulink già esistente, contrassegnato con l'estensione `.mdl` (`.m` nelle vecchie versioni), o aprire un nuovo file, per mezzo dei corrispondenti comandi `File/Open...` e `File/New/Model`. Utilizziamo il secondo per rappresentare un semplice sistema, costituito da una funzione di trasferimento con un ingresso a scalino e un oscilloscopio collegato in uscita.

Per realizzare tale sistema trasciniamo i corrispondenti blocchi Transfer Fcn, Step e Scope, situati nella libreria rispettivamente nel gruppo Simulink/Continuous, Simulink/Sources e Simulink/Sinks. Successivamente connettiamo le icone utilizzando il tasto sinistro del mouse, mantenendolo premuto e trascinando il mouse fino ad ottenere la linea di lunghezza voluta.

Infine, salviamo il sistema come `Simple.mdl` (figura 2), in modo da poterlo richiamare in futuro semplicemente digitandone il nome, senza estensione, dalla Command Window (avendo precedentemente avuto cura di spostarci, con il comando `cd`, nella directory in cui tale file si trova).

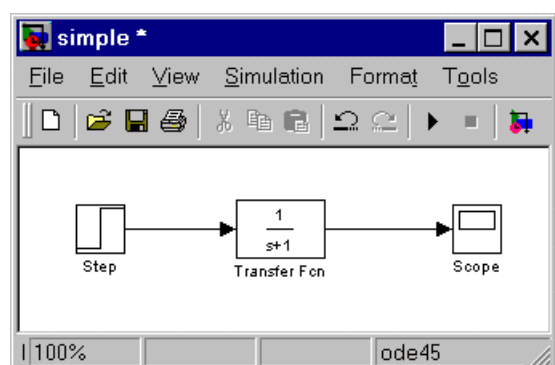


Figura 2. Sistema a blocchi `simple.mdl`

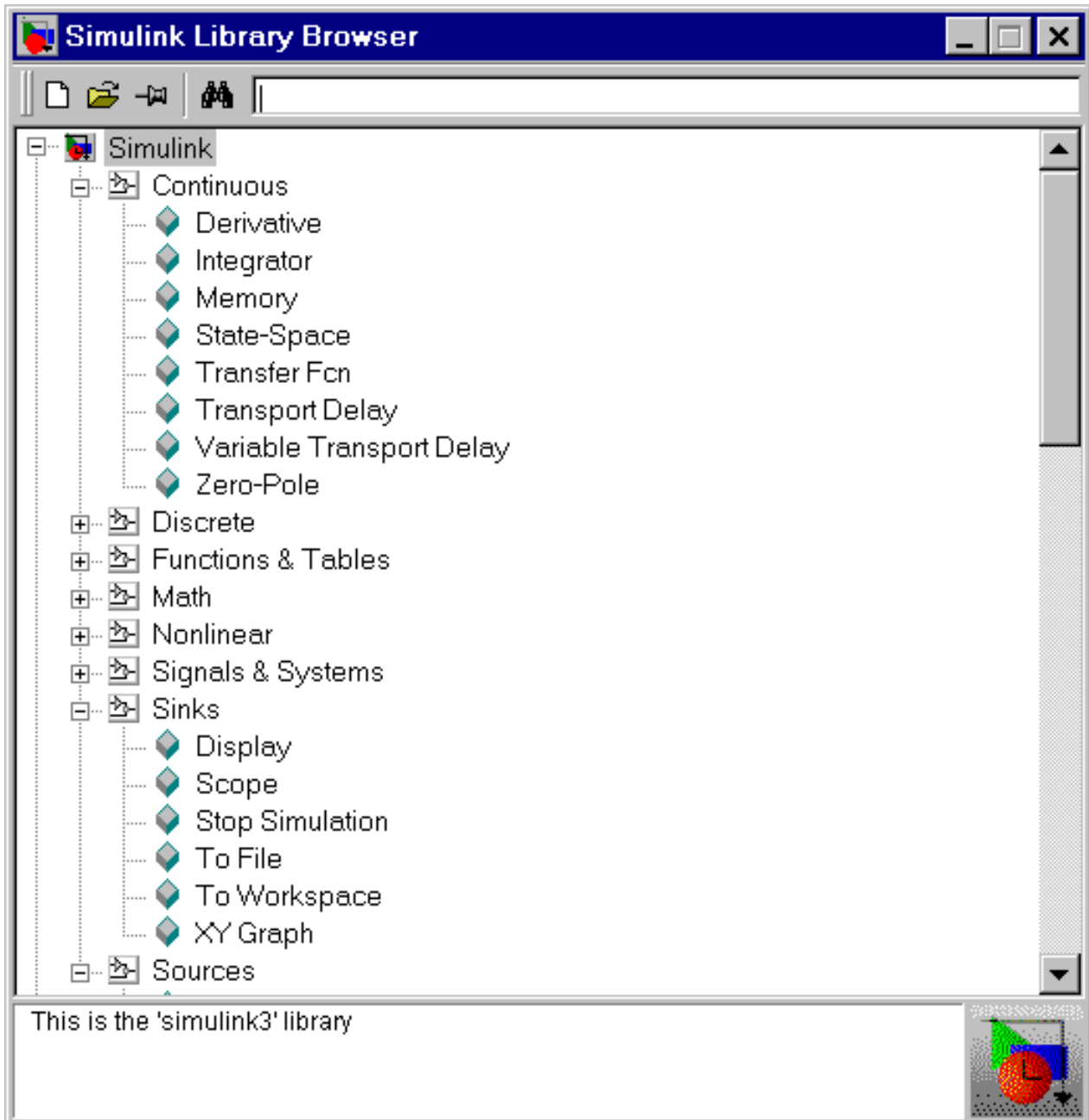


Figura 1. Simulink Library Browser

Determiniamo ora il comportamento dinamico del sistema a blocchi, ossia la risposta allo scalino della funzione di trasferimento scelta, eseguendo la simulazione, ossia selezionando **Start** dal menu **Simulation**. In particolare, dal menu **Simulation/Parameters...** è anche possibile scegliere il tempo di inizio e fine della simulazione, il solutore delle equazioni differenziali scelte, nonché decidere di memorizzare la variabile tempo, per esempio nel vettore `tout`. Questo è utile allorquando si voglia rappresentare in un grafico le variabili in gioco.

Supponiamo per esempio di voler conoscere l'uscita non solo graficamente, ma anche punto per punto. È sufficiente connettere all'uscita il blocco **To Workspace** del gruppo **Simulink/Sinks**, assegnare a tale variabile il nome `yout` e il formato `Matrix`, salvare il nuovo sistema (figura 3) e rieseguire la simulazione. Digitando `who` nel workspace di MATLAB si

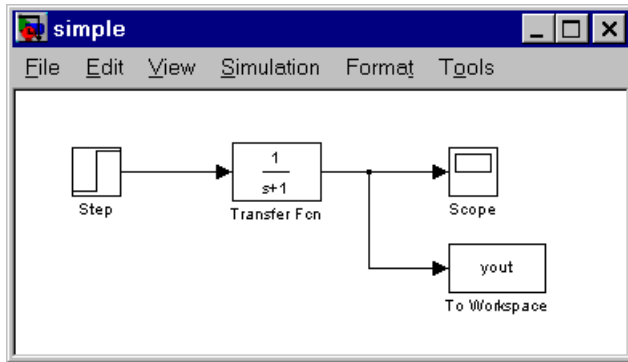


Figura 3. Sistema a blocchi modificato

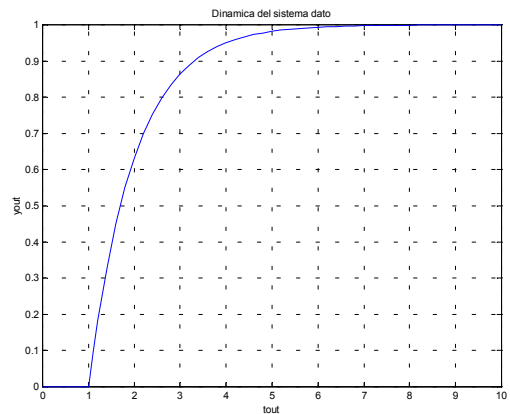


Figura 4. Grafico dell'uscita.

troveranno le variabili tout e yout, e con dei semplici comandi è possibile tracciare l'uscita del sistema (figura 4).

» who

Your variables are:

tout yout

» plot(tout,yout),grid
 » title('Dinamica del sistema')
 » xlabel('tout'), ylabel('yout')

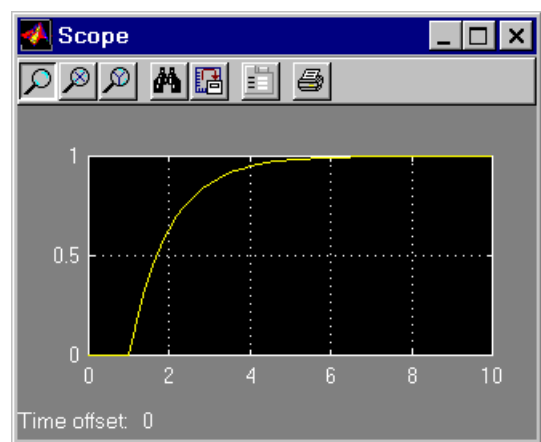


Figura 5. Blocco Scope.

La stessa dinamica si può anche analizzare aprendo il blocco Scope (figura 5). Per inquadrare il segnale di uscita, rappresentato in giallo, nel riquadro a disposizione, si utilizza il bottone autoscale, rappresentato da un binocolo. Si noti che la risposta al gradino comincia solo all'istante $t=1$. Questo è dovuto al fatto che il gradino comincia in tale istante, e si può cambiare intervenendo sul blocco Step, ossia cliccandolo due volte e modificando il tempo di inizio.

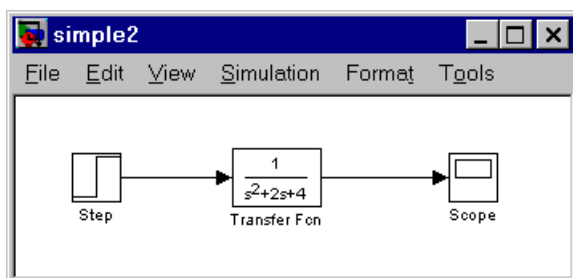


Figura 6. Sistema a blocchi con funzione di trasferimento del secondo ordine.

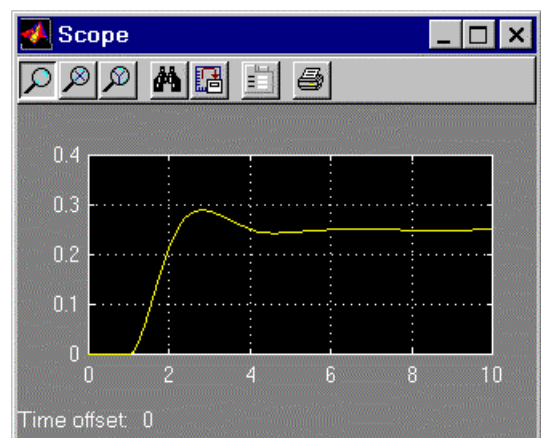


Figura 7. Risposta al gradino del sistema simple2.mdl.

Seguiamo infine la stessa procedura per un sistema del secondo ordine (figure 6 e 7), ottenuto cambiando il blocco 'Transfer Fcn' con l'introduzione a denominatore del polinomio

di secondo grado [1 2 4].

2. Evoluzione di sistemi chiusi in retroazione

Costruiamo il sistema riportato in figura 8, costituito da un plant del secondo ordine, un controllore di tipo PI (Proporzionale Integrale) con in cascata un guadagno, e un ramo di retroazione unitaria. L'ingresso è un gradino e l'uscita è visualizzata su un oscilloscopio. In particolare, entrambi i blocchi Sum e Gain sono stati copiati dalla libreria Simulink/Continuous, mentre il blocco Pi controller è stato ottenuto modificando opportunamente un blocco Transfer Fcn.

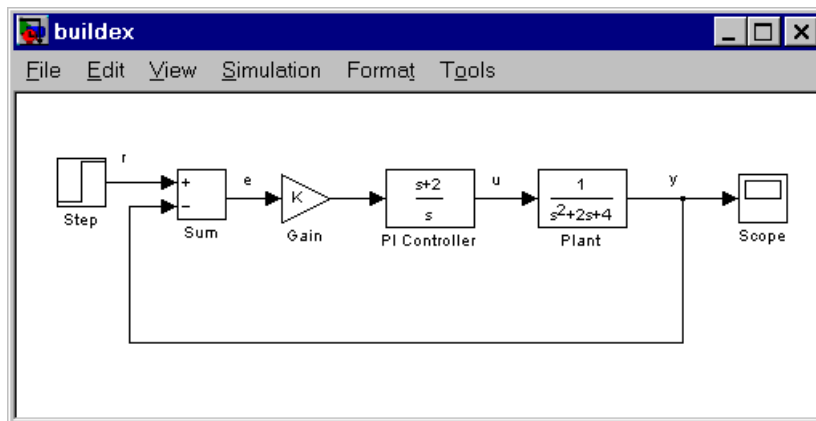


Figura 8. Sistema in retroazione unitaria.

Si nota che il guadagno del controllore è di valore variabile K, che viene assegnato di volta in volta dalla Command Window come segue:

» K=2.5

Eseguendo la simulazione si ottiene la risposta in figura 9. Ponendo ora

» K=5

si ottengono invece i risultati in figura 10. Si vede dunque come Simulink sia un eccellente strumento grafico per l'analisi dei sistemi al variare di qualche parametro.

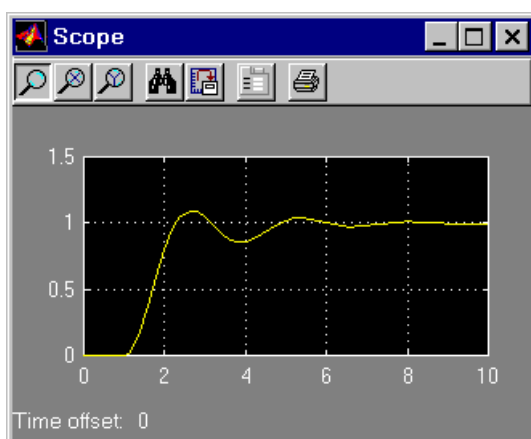


Figura 9. Risultati per K=2.5.

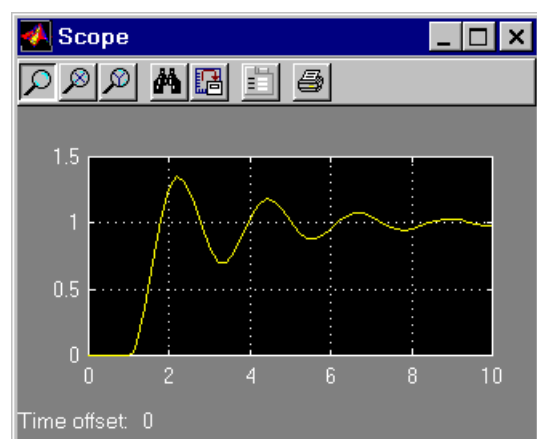


Figura 10. Risultati per K=5.

3. Analisi dinamica di sistemi fisici con Simulink e MATLAB

Simulink è un valido strumento di analisi per tutti i sistemi fisici dei quali si abbia un modello più o meno accurato. Consideriamo come esempio il sistema meccanico in figura 11. Si tratta di un carrello di massa M (kg) che, soggetto ad una forza orizzontale F (N), può muoversi di moto orizzontale con spostamento x (m) ma con il vincolo di essere collegato ad una parete fissa attraverso una molla di costante elastica K (N/m) e uno smorzatore di smorzamento B (Ns/m). Applicando la prima legge della dinamica o legge di Newton, l'equilibrio delle forze cui è sottoposto il sistema è il seguente:

$$M\ddot{x} + B\dot{x} + Kx = F.$$

Consideriamo dunque le due variabili di stato posizione e velocità:

$$x_1 = x$$

$$x_2 = \dot{x}$$

e determiniamo il modello in variabili di stato del sistema come segue:

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = \frac{1}{M}(F - Bx_2 - Kx_1) \quad (1).$$

Si vuole studiare il sistema con condizioni iniziali nulle (il carrello è nella posizione di riferimento $x_1=x=0$ con velocità nulla $x_2=\dot{x}=0$), quando ad esso è sottoposta una forza costante di 25N a partire dall'istante di tempo $t=0$. Si assume $M=1\text{kg}$, $B=5\text{Ns/m}$ e $K=25\text{N/m}$.

Per rappresentare graficamente in ambiente Simulink il modello ottenuto, interpretiamo la (1) realizzando la variabile \dot{x}_2 per mezzo di un sommatore, i cui ingressi sono combinazioni, tramite opportuni guadagni, della forza F e delle variabili di stato x_2 e x_1 retroazionate. Queste vengono ottenute rispettivamente la prima da \dot{x}_2 per mezzo di un semplice integratore, la seconda da x_2 integrando ancora una volta. Le condizioni iniziali dei due integratori sono nulle, poiché il sistema evolve dalla quiete. Si ottiene così il modello in Simulink denominato mecc.mdl

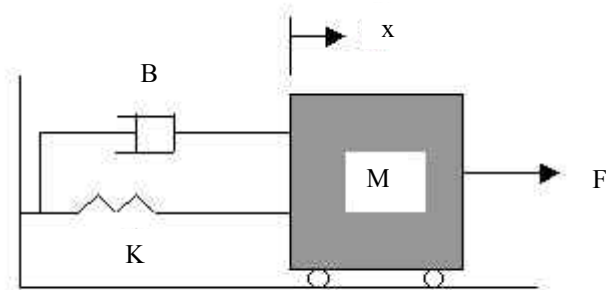


Figura 11. Sistema meccanico traslatorio

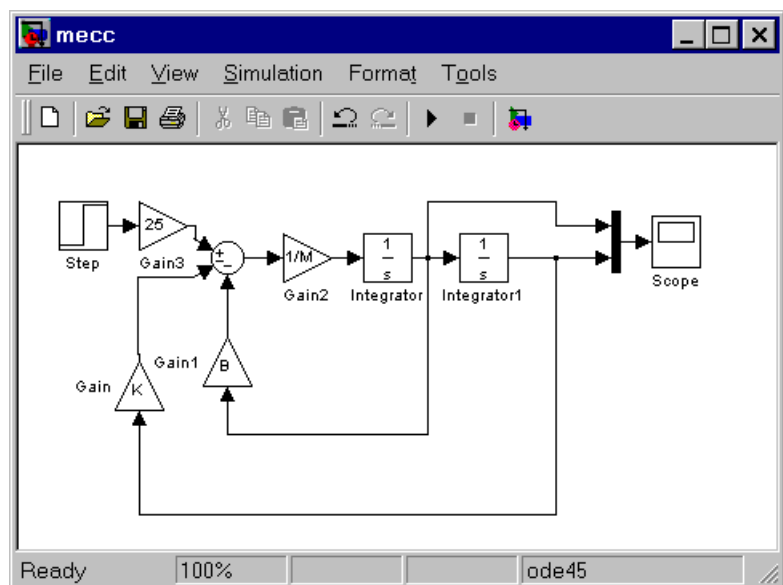


Figura 12. Modello in Simulink del sistema meccanico traslatorio.

(figura 12).

Si noti che in ingresso al blocco Scope è stato posto un multiplexer (blocco Mux nella libreria Simulink/Signals & Systems), in modo da poter visualizzare insieme il segnale di uscita (la posizione) e la sua derivata (la velocità della massa). Per eseguire la simulazione, definiamo i parametri del sistema dalla finestra di comando:

```
» M=1;
» K=25;
» B=5;
```

In figura 13 è riportato il risultato della simulazione. Si vede che, dopo una breve oscillazione, in circa 2 secondi il carrello si ferma 1m a destra della posizione originale ($x_1=1$, in viola, e $x_2=0$, in giallo).

Vediamo ora come realizzare questa stessa analisi utilizzando le sole potenzialità di calcolo di MATLAB.

È sufficiente ridefinire l'equazione principale (1) del sistema per mezzo di una funzione `sismecc.m` che riportiamo nel seguito:

```
function xdot=sismecc(t,x)
F=25;
M=1;
B=5;
K=25;
xdot=[x(2); 1/M*(F-B*x(2)-K*x(1))];
```

Questa funzione è richiamata dal file MATLAB `sismeccgraf.m`, di seguito riportato, il quale simula il sistema meccanico definito in un intervallo di 3 secondi e con condizioni iniziali nulle, per mezzo di un algoritmo solutore di equazioni differenziali detto `ode23` incluso nel MATLAB.

```
% File sismeccgraf.m
% Determina la dinamica del sistema meccanico
% traslatorio definito dalla funzione sismecc.m

t0=0;
tf=3;
x0=[0 0];
tol=0.001;
trace=0;
[t,x]=ode23('sismecc',t0,tf,x0,tol,trace);
subplot(211),plot(t,x),grid
title('Evoluzione del sistema meccanico')
xlabel('tempo [s]')
legend('spostamento','velocità')
d=x(:,1);
v=x(:,2);
subplot(212),plot(d,v),grid
title('Velocità rispetto allo spostamento')
xlabel('spostamento')
ylabel('velocità')
```

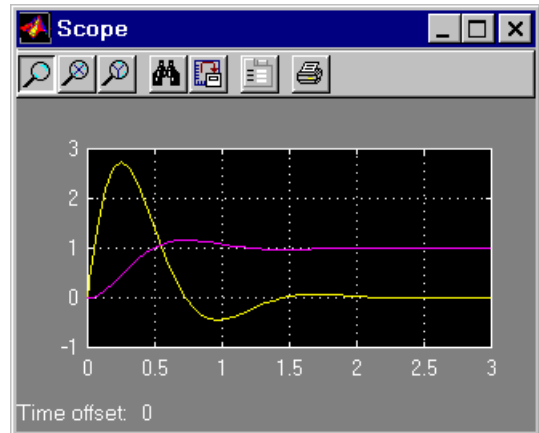


Figura 13. Evoluzione del sistema meccanico traslatorio.

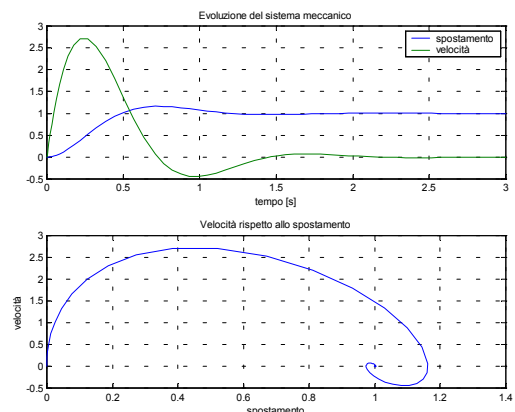


Figura 14. Risultati dell'esecuzione del file `sismeccgraf.m`.

Il file sismecgraf.m, inoltre, visualizza in un'unica finestra il grafico dell'andamento temporale della posizione e della velocità e il diagramma polare posizione-velocità (figura 14). L'esecuzione avviene semplicemente digitando nella finestra di comando il nome del file:

» sismecgraf

Un esempio analogo al precedente è dato dalla rete elettrica in figura 15. Si tratta di un circuito elettrico comunemente detto RLC serie, poiché esso è costituito da un generatore di tensione V_s (V) che alimenta la serie di una resistenza R (Ω), una induttanza L (H) e una capacità C (F). Applicando la legge di Kirchoff delle tensioni si ottiene:

$$Ri + Li + V_c = V_s$$

con

$$V_c = \frac{1}{C} \int_0^t i(\tau) d\tau.$$

Consideriamo dunque le due variabili di stato date dalla tensione sul condensatore e la corrente snella maglia:

$$x_1 = V_c$$

$$x_2 = i$$

e determiniamo il modello in variabili di stato del sistema come segue:

$$\dot{x}_1 = \frac{1}{C} x_2$$

$$\dot{x}_2 = \frac{1}{L} (V_s - x_1 - Rx_2) \quad (2).$$

Si vuole studiare il sistema con condizioni iniziali $[0.5 \ 0]$ (il condensatore ha una carica residua $x_1=V_c=0.5$ V e la corrente iniziale sull'induttore è $x_2=i=0$), quando ad esso è sottoposta una tensione costante di 1V a partire dall'istante di tempo $t=0$. Si assume $R=1.4\Omega$, $L=2H$ e $C=0.32F$.

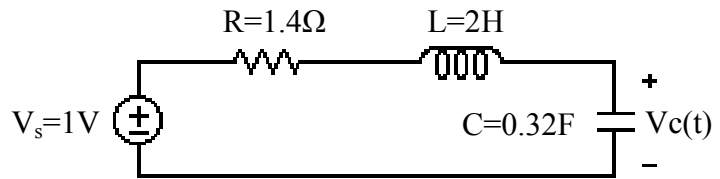


Figura 15. Circuito RLC serie.

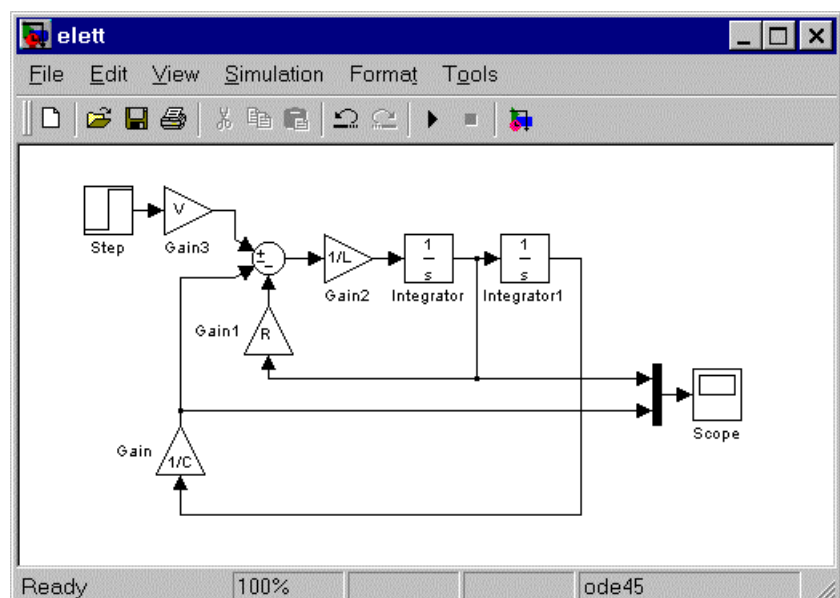


Figura 16. Modello in Simulink del sistema elettrico RLC serie.

Per rappresentare graficamente in ambiente Simulink il modello ottenuto, interpretiamo la (2) realizzando la variabile \dot{x}_2 per mezzo di un sommatore, i cui ingressi sono combinazioni, tramite opportuni guadagni, della tensione di ingresso V_s e delle variabili di stato x_2 e x_1 retroazionate. Queste vengono ottenute rispettivamente la prima da \dot{x}_2 per mezzo di un semplice integratore, la seconda da x_2 integrando ancora una volta e dividendo per C . La condizione iniziale del primo integratore è nulla, poiché il valore iniziale della corrente è zero, mentre la condizione iniziale del secondo integratore è 0.5, poiché il condensatore possiede una carica iniziale di 0.5V. Si ottiene così il modello in Simulink denominato elett.mdl (figura 16).

Per eseguire la simulazione, definiamo i parametri del sistema dalla finestra di comando:

```
» V=1;
» R=1.4;
» L=2;
» C=0.32;
```

In figura 17 è riportato il risultato della simulazione. Si vede che, dopo una oscillazione di circa 10 secondi, il condensatore si carica completamente alla tensione V_s del generatore e la corrente si annulla ($x_1=1$, in viola, e $x_2=0$, in giallo).

Vediamo ora come realizzare questa stessa analisi utilizzando le sole potenzialità di calcolo di MATLAB.

È sufficiente ridefinire l'equazione principale (2) del sistema per mezzo di una funzione siselett.m che riportiamo nel seguito:

```
function xdot=siselett(t,x)
V=1;
R=1.4;
L=2;
C=0.32;
xdot=[x(2)/C; 1/L*(V-x(1)-R*x(2))];
```

Questa funzione è richiamata dal file MATLAB siselettgraf.m, di seguito riportato, che simula il sistema elettrico definito in un intervallo di 15 secondi e con condizioni iniziali [0.5 0] per mezzo dell'algoritmo solutore di equazioni differenziali ode23.

```
% File siselettgraf.m
% Determina la dinamica del sistema elettrico
% RLC serie definito dalla funzione siselett.m

t0=0;
tf=15;
x0=[0.5 0];
tol=0.001;
trace=0;
[t,x]=ode23('siselett',t0,tf,x0,tol,trace);
subplot(211),plot(t,x),grid
title('Evoluzione del sistema elettrico')
xlabel('tempo [s]')
legend('tensione del condensatore','corrente')
vc=x(:,1);
```

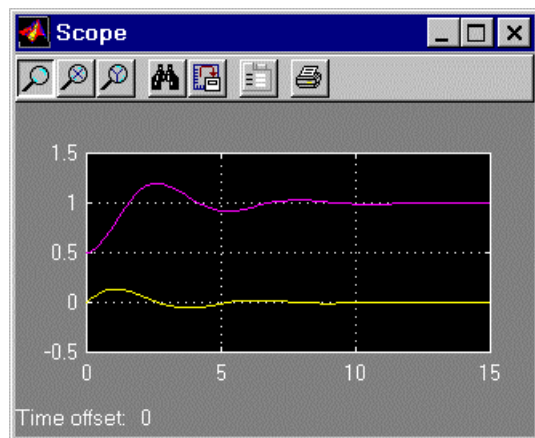


Figura 17. Evoluzione del sistema elettrico RLC serie.

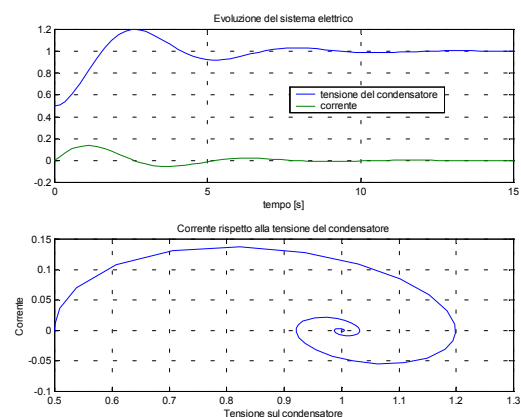


Figura 18. Risultati dell'esecuzione del file siselettgraf.m.

```

i=x(:,2);
subplot(212),plot(vc,i),grid

title('Corrente rispetto alla tensione del condensatore')
xlabel('Tensione sul condensatore')
ylabel('Corrente')

```

Il file siselettgraf.m, inoltre, visualizza in un'unica finestra il grafico dell'andamento temporale della tensione sul condensatore e della corrente, nonché il diagramma polare tensione-corrente (figura 18). L'esecuzione avviene semplicemente digitando nella finestra di comando il nome del file:

```
» siselettgraf
```

I sistemi finora trattati sono lineari. Vediamo ora come sia possibile trattare in modo analogo a quanto fatto anche sistemi non lineari, che costituiscono del resto la stragrande maggioranza dei modelli di sistemi fisici, sia utilizzando Simulink che il solo MATLAB.

Consideriamo il pendolo semplice illustrato in figura 19. Si tratta di una massa m (kg) di peso $W=mg$ (N), essendo $g=9.81 \text{ ms}^{-1}$ l'accelerazione di gravità, incernierata in un punto fisso P per mezzo di una fune rigida di massa trascurabile, coefficiente di attrito rotazionale B ($\text{Nsm}^{-1}\text{rad}^{-1}$) e lunghezza L (m). Il pendolo può ruotare liberamente nel piano verticale, e l'angolo θ (rads^{-1}) rappresenta l'angolo di deviazione della fune dalla verticale: a quest'ultima corrisponde $\theta=0$, mentre si suppongono angoli positivi per deviazioni a destra della stessa.

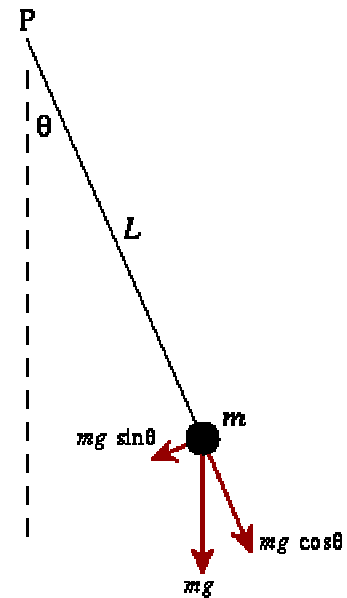


Figura 19. Pendolo semplice.

Se inizialmente la massa non si trova sulla verticale da P , essa tende ad oscillare per effetto del suo peso, contrastata dall'attrito rotazionale della fune. Gli archi corrispondenti ad una rotazione θ valgono evidentemente $s=L\theta$, da cui consegue che ad un angolo θ corrisponde una accelerazione angolare $\ddot{\theta}$ ed una accelerazione lineare $\ddot{s} = L\ddot{\theta}$. Applicando la prima legge di Newton alla massa si ottiene dunque:

$$m\ddot{s} = -W\sin\theta - B\dot{s}$$

essendo

$$-W\sin\theta = -mg\sin\theta$$

la componente della forza peso, negativa perché tende a riportare indietro il pendolo, e

$$-B\dot{s} = -BL\dot{\theta}$$

la forza di attrito rotazionale cui è sottoposta la fune, negativa anch'essa poiché tende a frenare il moto. In definitiva si ha:

$$mL\ddot{\theta} = -W\sin\theta - BL\dot{\theta}$$

Consideriamo dunque le due variabili di stato posizione e velocità angolare:

$$x_1 = \theta$$

$$x_2 = \dot{\theta}$$

e determiniamo il modello in variabili di stato del sistema come segue:

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -\frac{B}{m}x_2 - \frac{W}{mL}\sin x_1 \quad (3).$$

Si vuole studiare il sistema con condizioni iniziali $[1 \ 0]$ (il pendolo ha una posizione iniziale $x_1 = \theta = 1 \text{ rads}^{-1}$ e una velocità angolare iniziale nulla $x_2 = \dot{\theta} = 0$), a partire dall'istante di tempo $t=0$. Si assume $W=2\text{N}$, $B=0.02 \text{ Nsm}^{-1}\text{rad}^{-1}$ e $L=0.6\text{m}$.

Per rappresentare graficamente in ambiente Simulink il modello ottenuto, interpretiamo la (3) realizzando la variabile \dot{x}_2 per mezzo di un

sommatore, i cui ingressi sono combinazioni, tramite opportuni guadagni, della componente peso tangenziale alla fune $-W\sin x_1$ e della variabile di stato x_1 retroazionata. Quest'ultima si ottiene da \dot{x}_2 per mezzo di due integratori, il primo con condizione iniziale nulla, poiché il valore iniziale della velocità angolare è zero, il secondo con condizione iniziale 1, poiché tale è il valore iniziale dell'angolo. Si ottiene così il modello in Simulink denominato pend.mdl (figura 20).

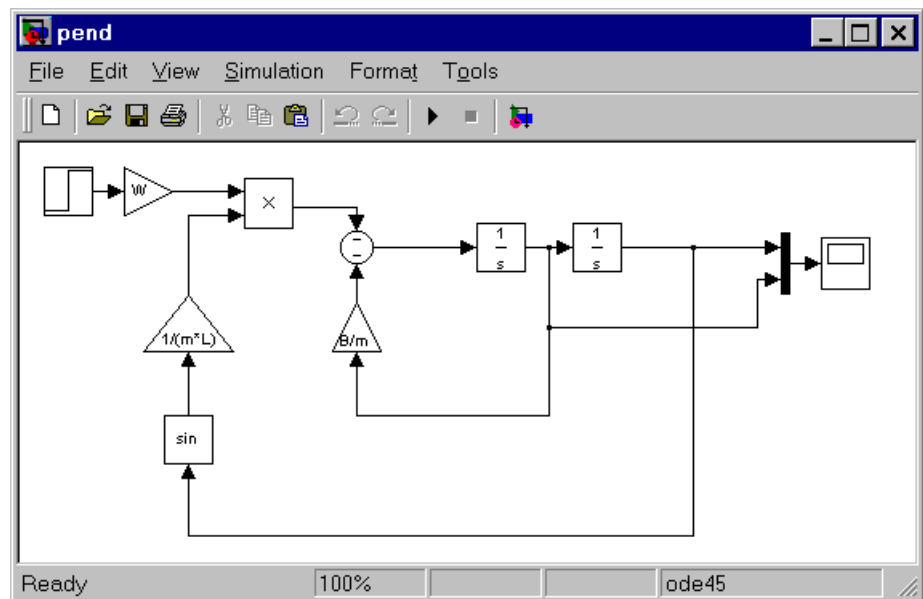


Figura 20. Modello Simulink del pendolo semplice.

Per eseguire la simulazione, definiamo i parametri del sistema dalla finestra di comando:

- » $W=2$;
- » $L=0.6$;
- » $B=0.02$;
- » $g=9.81$;
- » $m=W/g$;

In figura 21 è riportato il risultato della simulazione.

Si vede che il pendolo, sottoposto alla forza peso, oscilla intorno alla verticale (variabile x_1 , in giallo), con uno smorzamento quasi non apprezzabile, evidentemente perché il coefficiente B è molto basso, mentre la sua velocità varia anch'essa con lo stesso

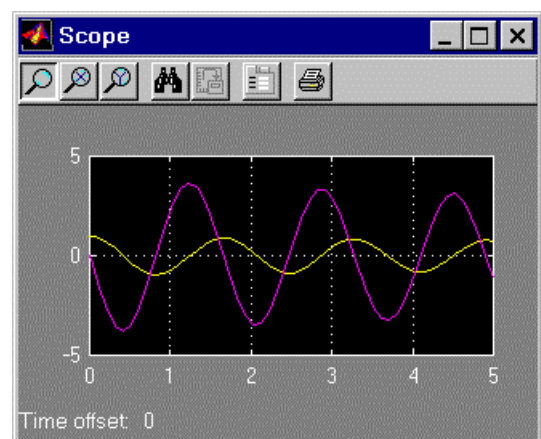


Figura 21. Evoluzione del pendolo.

periodo (variabile x_2 , in viola), ossia il pendolo si muove in avanti a indietro.

Vediamo ora come realizzare questa stessa analisi utilizzando le sole potenzialità di calcolo di MATLAB. È sufficiente ridefinire l'equazione principale del sistema (3) per mezzo di una funzione pendolo.m che riportiamo nel seguito:

```
function xdot=pendolo(t,x)
W=2;
L=0.6;
B=0.02;
g=9.81;
m=W/g;
xdot=[x(2); -B/m*x(2)-W/m/L*sin(x(1))];
```

Questa funzione è richiamata dal file MATLAB pendolograf.m, di seguito riportato, che simula il sistema pendolo in un intervallo di 5 secondi e con condizioni iniziali [1 0] per mezzo dell'algoritmo solutore di equazioni differenziali ode23.

```
% File pendolograf.m
% Determina la dinamica del pendolo semplice
% definito dalla funzione pend.m

t0=0;
tf=5;
x0=[1 0];
tol=0.001;
trace=0;
[t,x]=ode23('pendolo',t0,tf,x0,tol,trace);
subplot(211),plot(t,x),grid
title('Evoluzione del pendolo')
xlabel('tempo [s]')
legend('Angolo','Velocità angolare')
theta=x(:,1);
omega=x(:,2);
subplot(212),plot(theta,omega),grid
title('Piano delle fasi del pendolo')
xlabel('Angolo')
ylabel('Velocità angolare')
```

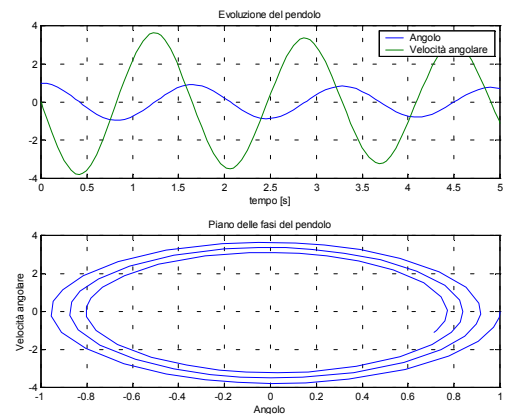


Figura 22. Risultati dell'esecuzione del file pendolograf.m.

Il file pendolograf.m, inoltre, visualizza in un'unica finestra il grafico dell'andamento temporale della posizione e della velocità angolare del pendolo, nonché il diagramma polare posizione-velocità (figura 22). L'esecuzione avviene semplicemente digitando nella finestra di comando il nome del file:

```
» pendolograf
```

4. Caso in studio: controllo della posizione di un'antenna

Il controllo di posizione ha lo scopo di regolare la posizione di un sistema secondo alcune specifiche. Un caso classico è il controllo di posizione di un'antenna, in cui si cerca di controllare la posizione azimutale del ricevitore in modo da selezionare un particolare satellite sull'orizzonte terrestre. Uno schema a blocchi di tale sistema realizzato in Simulink è riportato in figura 23. L'obiettivo è controllare la posizione azimutale secondo la posizione desiderata. Questo segnale di comando è misurato per mezzo di un potenziometro, che lo trasforma in una tensione di comando per un amplificatore di guadagno variabile e un amplificatore di potenza. Il segnale di controllo così modificato è fornito al sistema, individuato da un motorino che muove l'antenna e da un carico meccanico dato dall'antenna stessa. La reale posizione dell'antenna è ottenuta per mezzo di un riduttore, che riporta i valori angolari nell'intervallo di interesse. La maggior parte dei sistemi è rappresentata da un semplice guadagno, eccetto l'amplificatore di potenza, che presenta un polo in alta frequenza, ed il plant stesso, che presenta un polo nell'origine, dovuto all'integrazione tra velocità e posizione angolare, ed un polo negativo dipendente dalle caratteristiche elettriche del motore e da quelle meccaniche del motore stesso e del carico.

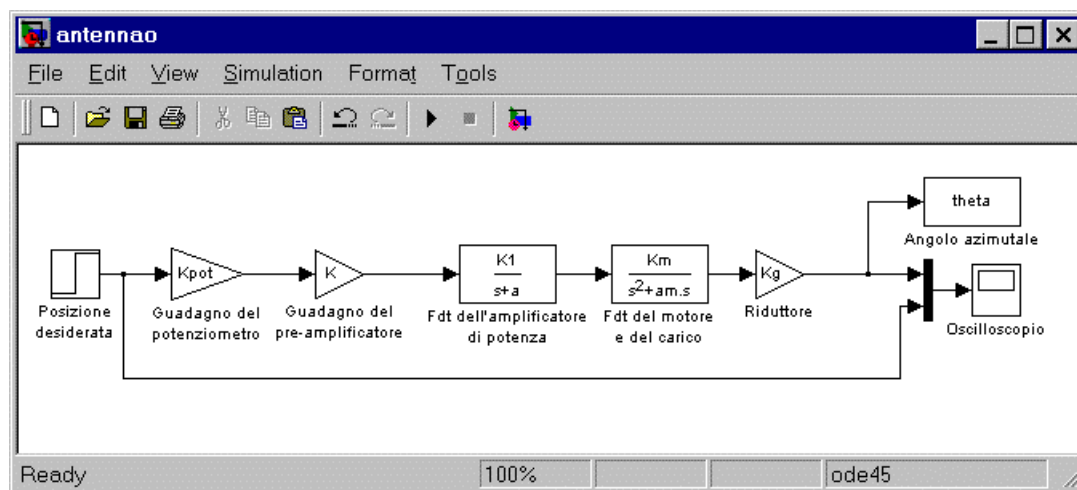


Figura 23. Modello Simulink dell'antenna in anello aperto.

Utilizzando Simulink è possibile non solo modellare facilmente il sistema dato, ma anche analizzarne il comportamento, per esempio in risposta ad un gradino, ossia in corrispondenza di una brusca variazione della posizione desiderata. Per simulare il sistema è necessario definirne i parametri caratteristici nella finestra di comando di MATLAB come segue:

- » $K_{pot}=1/\pi;$
- » $K1=100;$
- » $a=100;$
- » $Km=2.083;$
- » $am=1.71;$
- » $Kg=0.1;$

È evidente come si possa studiare la dinamica del sistema al variare del parametro K , definito volta per volta nella finestra di comando. Un esempio ottenuto per $K=10$ è riportato in figura 24. La risposta ottenuta, che ha un andamento a rampa, è facilmente interpretabile se si considera che il sistema ha una funzione di trasferimento complessiva facilmente calcolabile come

$$G_o(s) = \frac{K_{pot} \cdot K \cdot k_l \cdot K_m \cdot K_g}{s(s + a_m)(s + a)} = \frac{6.63K}{s(s + 100)(s + 1.71)} \quad (1).$$

Ad un ingresso a gradino il sistema risponde quindi con una posizione avente trasformata di Laplace

$$\Theta(s) = \frac{6.63K}{s^2(s + 100)(s + 1.71)}$$

che presenta nel tempo due modi esponenziali decrescenti, un gradino e soprattutto un modo lineare a rampa. È dunque evidente come il sistema non si possa controllare in anello aperto, ossia per via di un semplice amplificatore di guadagno variabile, ma sia consigliabile l'uso di un anello di retroazione.

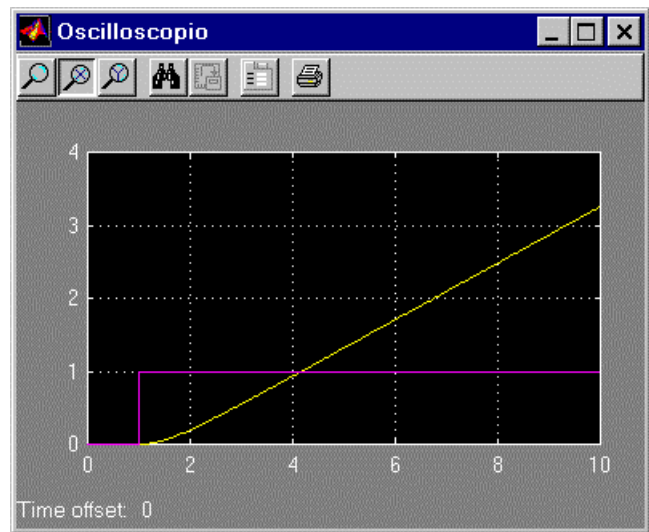


Figura 24. Evoluzione dell'antenna in anello

Retroazioniamo dunque la posizione azimutale per mezzo di un opportuno trasduttore, ovvero un potenziometro. Il sistema ottenuto è riportato in figura 25.

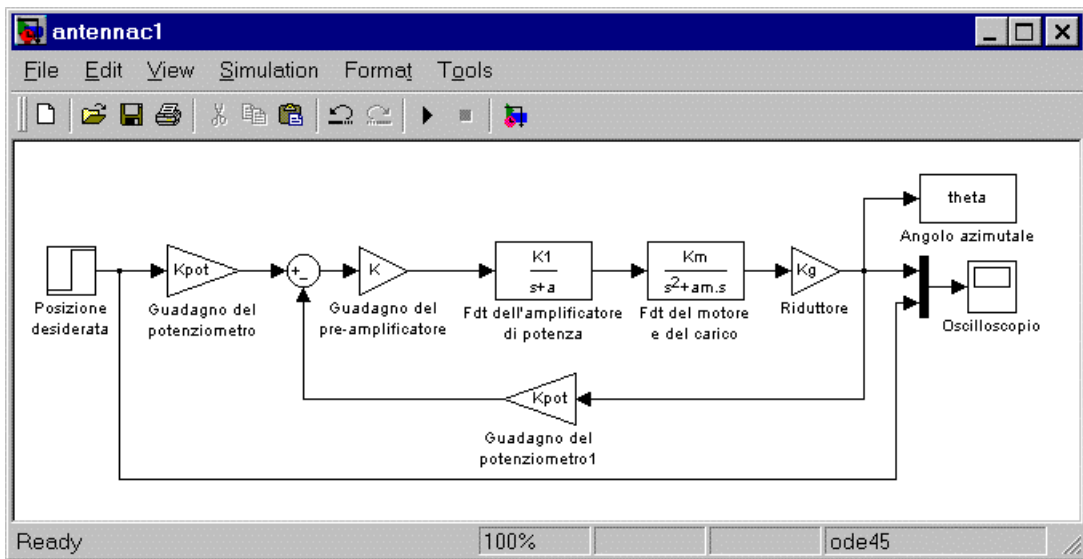


Figura 25. Modello Simulink dell'antenna in anello chiuso.

Il sistema in figura 25 è del tutto equivalente al sistema ridotto di figura 26, dove il potenziometro è riportato a valle del sommatore. Questo sistema in anello chiuso ha funzione di trasferimento di anello pari alla (1) ed ha retroazione unitaria, dunque la sua funzione di trasferimento in anello chiuso vale:

$$G_c(s) = \frac{6.63K}{s^3 + 101.71s^2 + 171s + 6.63K} \quad (2);$$

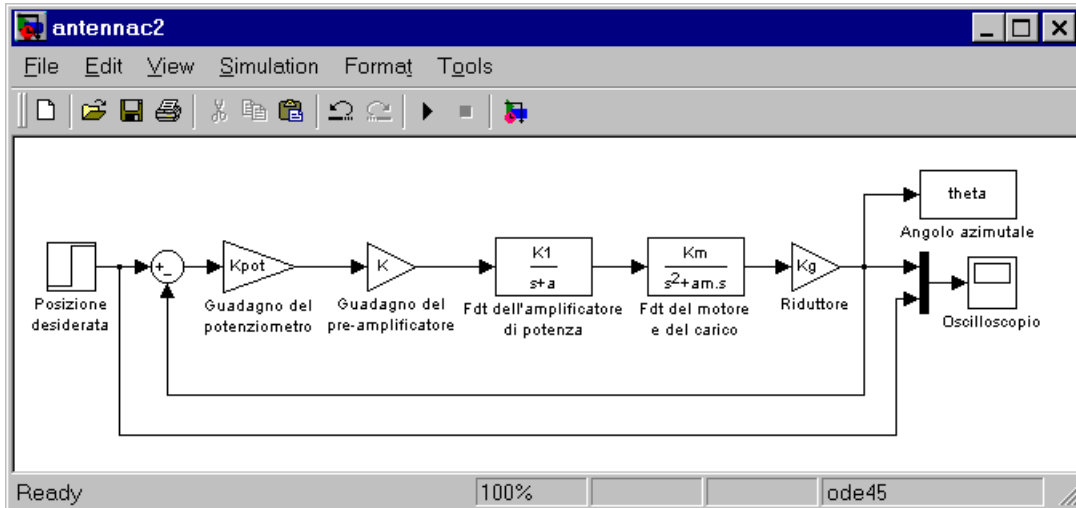


Figura 26. Modello Simulink ridotto dell'antenna in anello

Si può inoltre dimostrare che la funzione di trasferimento dell'amplificatore di potenza può essere approssimata con un guadagno unitario (figura 27). Essa ha infatti un guadagno statico nullo (per $s=0$ vale 1) e un polo lontano per $s=-a=-100$. È dunque possibile simulare l'evoluzione della posizione sul sistema così modificato.

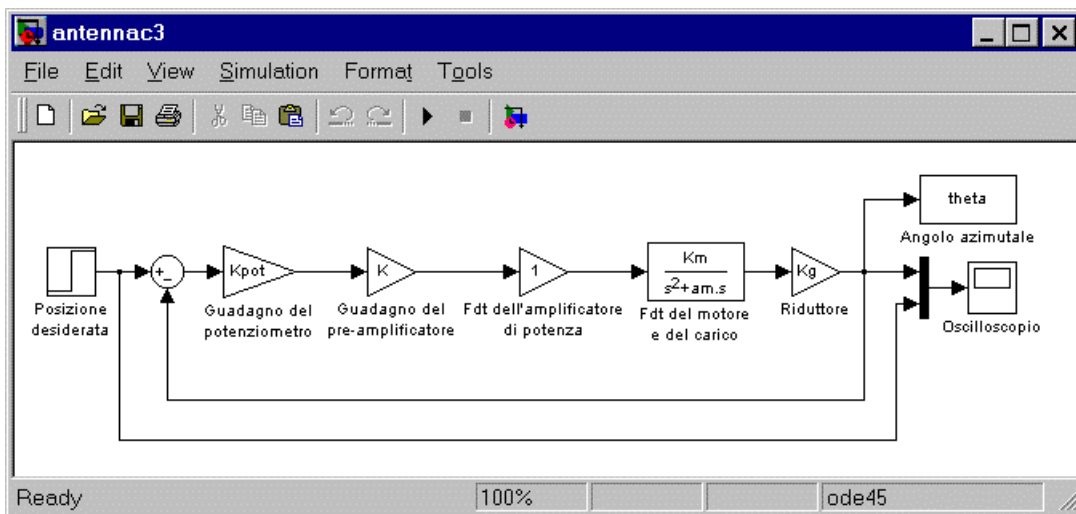


Figura 27. Modello Simulink semplificato in anello chiuso.

Il sistema diviene quindi del secondo ordine, con funzione di trasferimento

$$G'_c(s) = \frac{6.63K}{s^2 + 1.71s + 6.63K} \quad (3);$$

Introducendo ora diversi valori dell'amplificazione K dalla finestra di comando, si ottengono diversi comportamenti per il sistema retroazionato, ossia la configurazione dei poli in anello chiuso cambia, passando da poli reali negativi a complessi coniugati con parte reale negativa, fino a poli instabili. In particolare, aumentando il valore di K fino a 25 si ottiene la risposta diagrammata in figura 28, mentre aumentando ancora il parametro il sistema tende ad oscillare maggiormente: per $K=100$ si ottiene la dinamica riportata in figura 29.

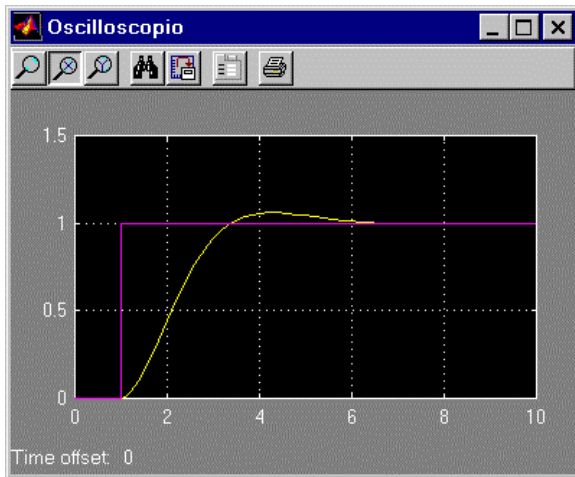


Figura 28. Risposta al gradino per K=25.

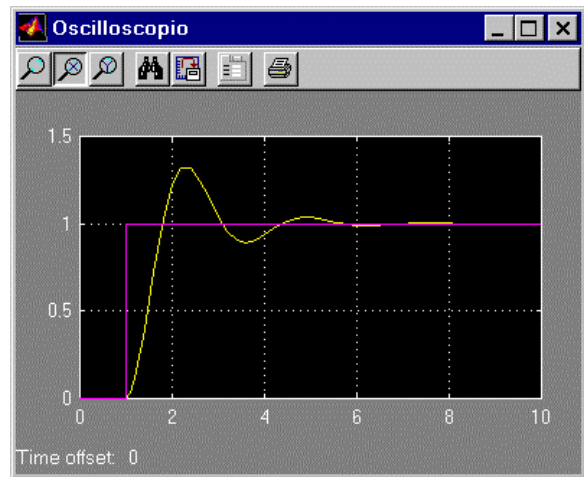


Figura 29. Risposta al gradino per K=100.

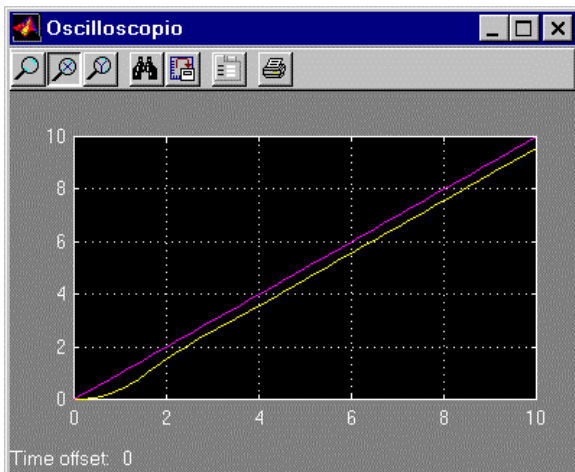


Figura 30. Risposta alla rampa per K=60.

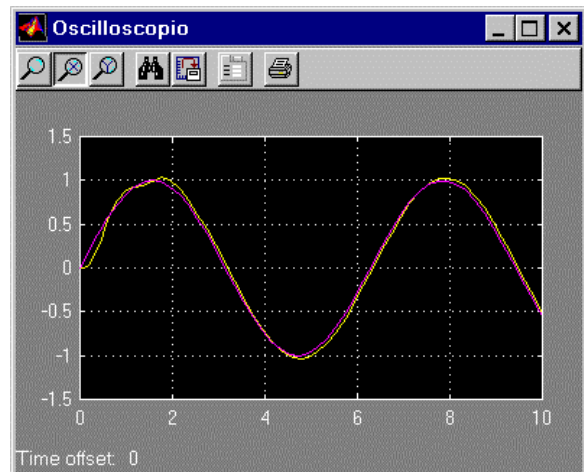


Figura 31. Risposta alla sinusoide per K=500.

Una analoga analisi si può effettuare quando la posizione desiderata è una rampa lineare o ha un andamento sinusoidale. In figura 30 e 31 sono riportati due esperimenti di questo tipo. In particolare è da notare che, mentre con un valore dell'amplificazione sufficientemente grande l'uscita del sistema è in grado di seguire un andamento sinusoidale, un andamento a rampa è possibile ma sempre con un errore a regime (errore di velocità).

5. Sistemi tempovarianti

Simulink è anche un ottimo strumento per l'analisi di sistemi tempovarianti. Consideriamo ad esempio il sistema di ordine 1 avente la seguente equazione di stato:

$$\dot{x}(t) = t \sin(x(t))$$

e supponiamo di voler studiare la variabile $x(t)$ del sistema nell'intervallo di tempo $[0 \ 10]$ e con condizione iniziale $x_0=10$. È sufficiente definire un sistema a blocchi (figura 32) che interpreti l'equazione di stato, utilizzando il blocco Clock per determinare il vettore dei tempi, e cambiare opportunamente i valori dell'intervallo di simulazione nel menu Simulation, nonché la condizione iniziale dell'integratore. Eseguendo la simulazione si ottiene la dinamica in figura 33.

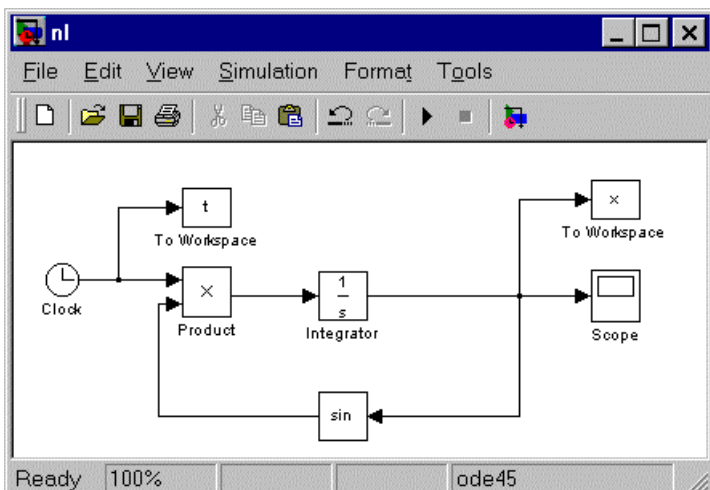


Figura 32. Sistema a blocchi tempovariante.

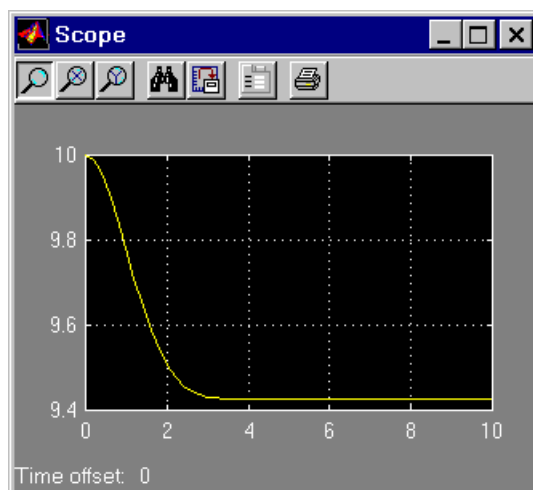


Figura 33. Dinamica del sistema tempovariante.

6. Sistemi con ritardo puro

Simulink è utile anche per l'analisi di sistemi che presentano un ritardo puro, ossia in cui un segnale è ritardato di un certo intervallo temporale. Consideriamo ad esempio il sistema rappresentato dalla seguente equazione differenziale del secondo ordine:

$$\ddot{x}(t) = -x(t) + x(t - 0.5)$$

e supponiamo di voler studiare la variabile $x(t)$ nell'intervallo di tempo $[0 \ 15]$ e con condizioni iniziali $x(0) = 5$, $\dot{x}(0) = 1$. È sufficiente definire un sistema a blocchi (figura 34) che interpreti l'equazione di stato, utilizzando l'apposito blocco Simulink detto *Transport delay* con un ritardo pari a 0.5, e cambiare opportunamente i valori dell'intervallo di simulazione, nonché le condizioni iniziali degli integratori. Si ottiene la dinamica riportata in figura 35.

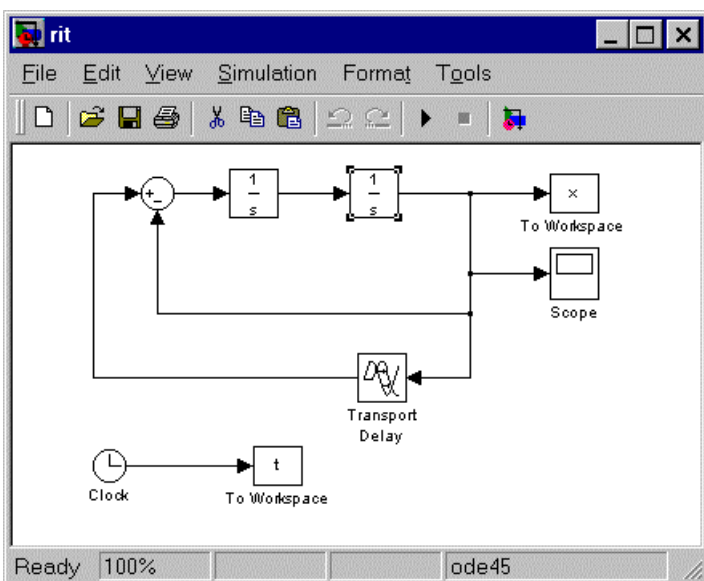


Figura 34. Sistema a blocchi con ritardo puro.

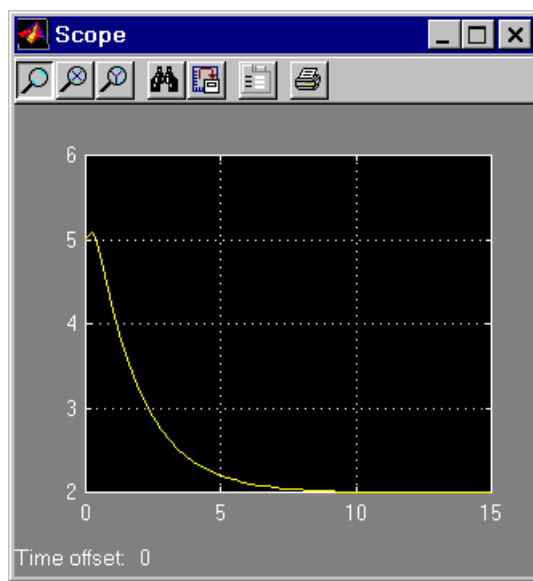


Figura 35. Dinamica del sistema con ritardo.

7. Bibliografia

A. Cavallo, R. Setola, F. Vasca, Guida operativa a MATLAB, Simulink e Control Toolbox, 1994, Liguori.

H. Saadat, Computational Aids in Control Systems Using MATLAB, 1993, Mc Graw-Hill.

M. Tibaldi, Note introduttive a MATLAB e Control System Toolbox, 1993, Progetto Leonardo.

N.S. Nise, Control Systems Engineering, seconda edizione, 1995, Benjamin Cummings.