

Universitatea Politehnica București

Laborator 2 Teoria Sistemelor

Introducere în Simulink

shiva.pub.ro

Laboratorul 1

Introducere în Simulink

Simulink reprezintă un instrument de lucru al programului MATLAB utilizat în special în proiectarea, modelarea și simularea sistemelor fizice și matematice. Utilitatea acestui instrument este evidențiată de modul de definire a entităților și a variabilelor de lucru sub forma unor **blocuri**. Modul de interacțiune și transmiterea variabilelor se realizează prin **conexiuni între blocuri**, ajungându-se astfel la un ansamblu sistemic denumit **schemă bloc**.

1.1 Primii pași în Simulink

Pasul 1 - Pentru a deschide interfața principală Simulink, se apelează în linia de comandă din Matlab comanda:

```
simulink
```

Pasul 2 - Se deschide **un nou model Simulink (New Model)**. Un model reprezintă o planșă de lucru în care se pot proiecta și simula schemele bloc specifice Simulink.

Observație: Modelele implementate în Simulink sunt salvate cu extensia *.slx*. Spre deosebire de scripturi și de funcții, modelele implementate într-o anumită versiune a programului nu pot fi deschise de versiuni anterioare ale Simulink.

1.2 Fereastra unui model Simulink

În Fig. 1.1 se poate vedea fereastra principală a unui model în Simulink (în formatul oferit de versiunea R2019b).

În partea superioară a ferestrei există o bară de instrumente împărțită în mai multe secțiuni: *File, Library, Prepare, Simulate, Review/Results*.

În secțiunea *File*, există opțiunea de a deschide, salva sau printa un anumit model.

În general, **modul de lucru** cu programul Simulink presupune două etape:

1.3. DEFINIREA PRIMULUI MODEL SIMULINK

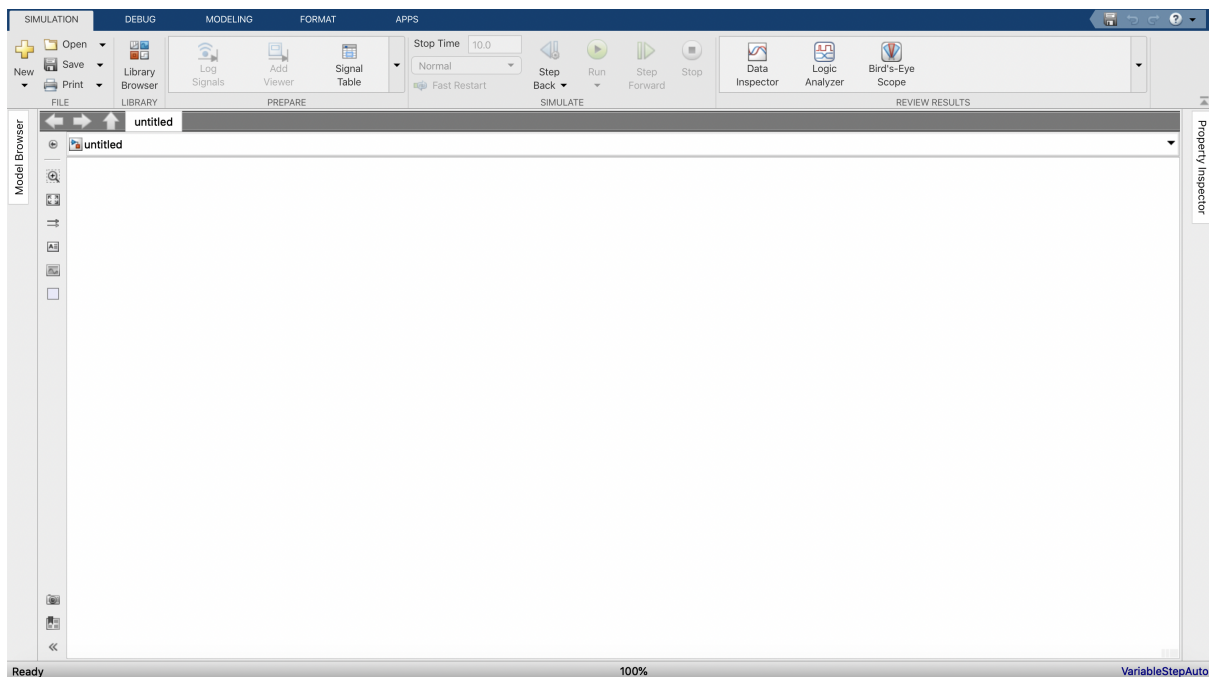


Figura 1.1: Fereastra principală a unui model Simulink (versiunea R2019b)

1. **Etapa de modelare** - în care se definește schema bloc a modelului ce urmează să fie simulat.
2. **Etapa de simulare** - în care se simulează modelul anterior implementat și se generează o serie de rezultate.

Etapa de modelare presupune definirea schemei bloc prin instanțierea blocurilor necesare implementării și stabilirea conexiunilor între acele blocuri.

Biblioteca din care se pot alege blocuri (Fig. 1.2) se numește **Library Browser** și poate fi accesată din secțiunea cu același nume din bara de instrumente. Cele mai importante blocuri din Simulink se regăsesc în categoriile:

- **Sources** - blocuri pentru simularea semnalelor de intrare.
- **Sinks** - blocuri pentru a analiza rezultatele simulării.
- **Continuous** - blocuri specifice diagramelor în continuu.
- **Discrete** - blocuri specifice diagramelor discrete.
- **Signal Routing** - blocuri pentru gruparea semnalelor de legătură dintre sisteme.

1.3 Definirea primului model Simulink

1. Din categoria **Sources** se selectează blocul pentru generarea semnalelor de tip treaptă (**Step**) și se aduce în fereastra modelului prin tehnica *drag-and-drop*. Semnalul de tip

LABORATORUL 1. INTRODUCERE ÎN SIMULINK

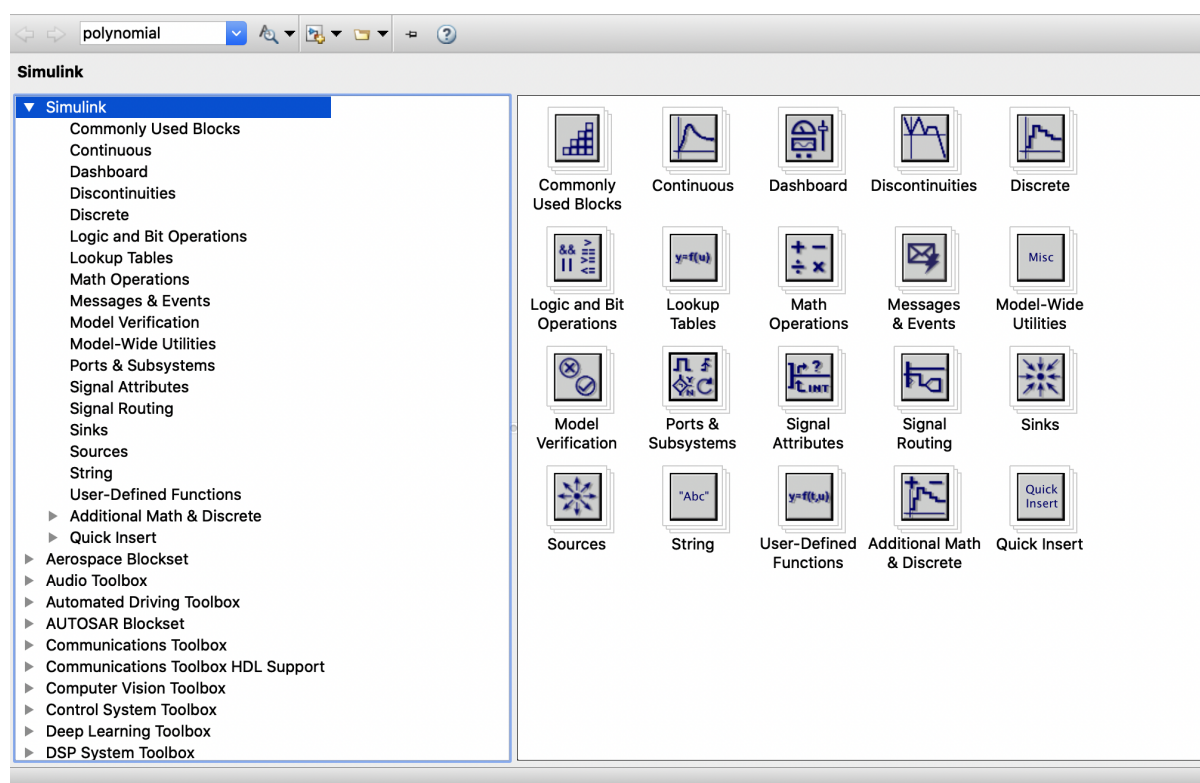


Figura 1.2: Biblioteca de componente Simulink

treaptă implicit Simulink este implementat pe baza relației:

$$u(t) = \begin{cases} 1, & t \geq 1 \\ 0, & t < 1 \end{cases} \quad (1.1)$$

2. Similar, din categoria **Sinks**, se aduce în fereastra de lucru un bloc de tip **Scope**.
3. **Etapă de configurare:** pentru a modifica parametrii implicați ai celor două blocuri, se execută *dublu-click* pe fiecare bloc în parte și se editează câmpurile din fereastra de dialog deschisă astfel. De exemplu, pentru blocul *Step* (Fig. 1.4), se pot configura:
 - **Step time** - timpul de la care semnalul se modifică din valoarea inițială în valoarea finală
 - **Initial value** - valoarea inițială a semnalului (implicit 0)
 - **Final Value** - valoarea finală a semnalului (implicit 1)
 - **Sample time** - perioada de eșantionare
4. Se realizează conexiunea dintre blocurile *Step* și *Scope* prin *drag-and-drop* de la portul de ieșire din *Step* până în portul de intrare în blocul *Scope* (Fig. 1.3).

În etapa de simulare, se pot executa unul sau mai multe scenarii de simulare. Pentru a putea

1.3. DEFINIREA PRIMULUI MODEL SIMULINK

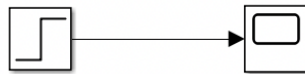


Figura 1.3: Conexiunea blocurilor în Simulink

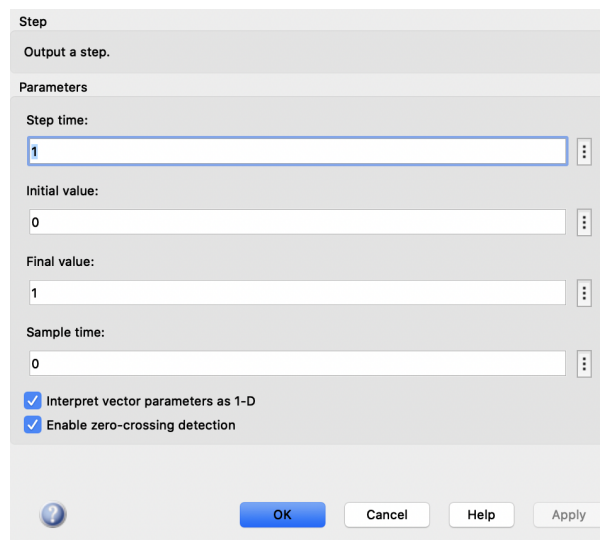


Figura 1.4: Fereastra de configurare a blocului *Step*

configura un astfel de scenariu, din bara de instrumente, secțiunea *Simulate*, se poate configura **timpul de simulare** al modelului (*Stop Time*) printr-o valoare de timp exprimată în secunde. Alternativ, dacă nu se dorește o limită de timp pentru simulare, acest câmp se poate completa cu valoarea *inf*.

Pentru o configurare mai detaliată a unui scenariu de simulare, din bara de instrumente, secțiunea **Modeling**, butonul **Model Settings** va deschide o fereastră de dialog în care se pot configura parametrii precum:

- **Start timp** - timpul de start al simulării.
- **Stop time** - durata simulării (în secunde).
- **Solver** - algoritmul de discretizare
- **(Solver) Type** - tipul de algoritm de discretizare: cu perioadă de eșantionare fixă sau variabilă.

Ulterior, pentru a putea porni simularea, se apasă pe butonul **Run** din fereastra modelului, secțiunea *Simulate*.

Pentru a putea vizualiza **rezultatele simulării**, se poate utiliza blocul **Scope**. Prin *dublu-click* pe acest bloc, se va deschide o fereastră nouă în care se va putea vizualiza evoluția în timp a semnalului de intrare generat de *Step* (Fig. 1.5). Altfel spus, blocul *Scope* are o funcționalitate

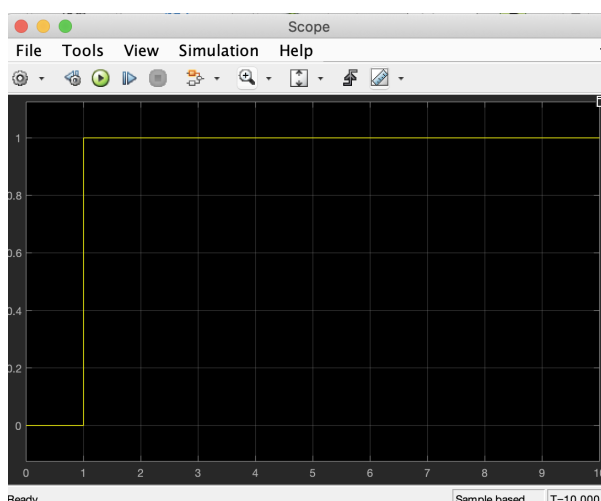


Figura 1.5: Fereastra pentru vizualizarea semnalelor a blocului *Scope*

similară cu cea a unui osciloscop.

1.4 Modelarea și simularea sistemelor continue în MATLAB/Simulink

O particularitate importantă a MATLAB/Simulink este dată de posibilitatea de a modela și simula foarte ușor **sisteme dinamice în reprezentare de tip funcție de transfer**.

- În MATLAB, funcția de transfer H se poate defini cu ajutorul comenzii:

$$H = tf(NUM, DEN)$$

, unde NUM și DEN reprezintă **vectorii coeficienților funcției de transfer** de la numărător, respectiv numitor.

Spre exemplu, pentru o funcție de transfer $H(s)$ definită conform Ec. 1.2, coeficienții se pot declara astfel:

$$H(s) = \frac{x_1 s^2 + x_2 s + x_3}{y_1 s^2 + y_2 s + y_3} \quad (1.2)$$

```
[x1 x2 x3] %coeficienții de la numărător
[y1 y2 y3] %coeficienții de la numitor
```

Pentru a putea genera un **model în reprezentare pe stare** dintr-un model de tip funcție de transfer, se poate utiliza comanda *tf2ss*. Similar, pentru a genera o funcție de transfer dintr-un model în spațiul stărilor, se poate utiliza comanda *ss2tf*.

Alternativ, pentru a construi o funcție de transfer ca o ecuație folosind variabila din domeniul operațional s , se poate utiliza următoarea rutină:

1.5. LEGĂTURA MATLAB WORKSPACE - SIMULINK

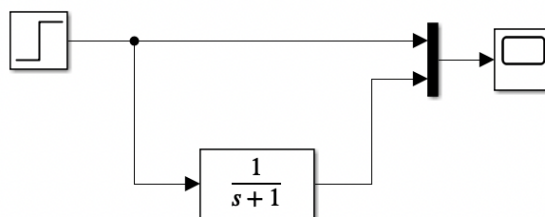


Figura 1.6: Modelul unei funcții de transfer cu intrare de tip treaptă

```
s = tf('s')
H = 1/(2*s+1)
```

Pentru a putea genera **răspunsul în domeniul timp** Y al unei **funcții de transfer** H pentru un **semnal de tip treaptă** standard, pe **intervalul** T , se poate apela comanda:

```
[Y, T] = step(H)
```

Acestea sunt doar o parte din capabilitățile MATLAB de a modela funcții de transfer. Se recomandă investigarea documentației prin intermediul comenzii *help* pentru următoarele instrucțiuni: *ss2tf*, *tf2ss*, *ss*, *tf*, *step*, *impulse*.

- În Simulink, atât semnalul de intrare de tip treaptă, cât și funcția de transfer trebuie să fie definite prin intermediul **blocurilor specifice**.

Plecând de la modelul deja creat în secțiunea anterioară, se vor parcurge următorii pași:

1. Din biblioteca de componente Simulink, secțiunea **Continuous**, se alege blocul **Transfer Fcn** (*Transfer Function* - funcție de transfer).
2. În fereastra de configurare a blocului *Transfer Fcn*, putem introduce **coeficienții** funcției de transfer - coeficienții de la numărător (*numerator coefficients*), respectiv coeficienții de la numitor (*denominator coefficients*).

Pentru acest exemplu, vom lăsa funcția de transfer implicită $H = \frac{1}{s+1}$.

3. Se implementează **conexiunile** schemei bloc. Se conectează ieșirea din blocul **Step** atât intrarea în blocul **Transfer Fcn**, cât și la intrarea în **Scope**.

Pentru a putea vizualiza în *Scope* atât semnalul treaptă de intrare, cât și răspunsul în timp al funcției de transfer, se va utiliza un bloc **Mux** (*Multiplexor*) din categoria **Signal Routing**.

După realizarea corespunzătoare a conexiunilor, modelul final arată conform Fig. 1.6.

1.5 Legătura Matlab Workspace - Simulink

Dat fiind faptul că Simulink reprezintă un instrument **integrat** în Matlab, variabilele declarate în spațiul de lucru din Matlab pot fi accesate într-un model definit în Simulink.

LABORATORUL 1. INTRODUCERE ÎN SIMULINK

Continuând exemplul definit anterior, se vor defini în spațiul de lucru Matlab **coeficienții funcției de transfer** astfel:

$$\begin{aligned}x1 &= 1.5 \\ y1 &= 2 \\ y2 &= 1\end{aligned}$$

Ulterior, în Simulink, în configurarea blocului *Transfer Fcn*, în locul valorilor implicite se vor introduce coeficienții $x1, y1$ și $y2$ (Fig. 1.7).

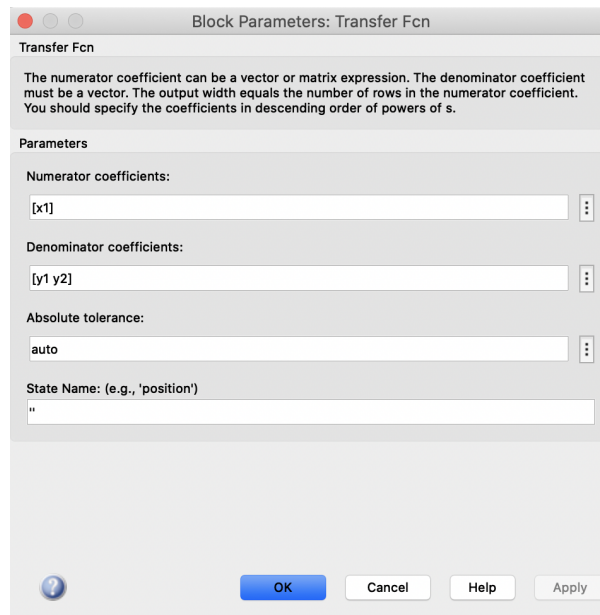


Figura 1.7: Configurarea blocului Transfer Fcn cu variabile din Workspace

După executarea unei simulări, se va observa pe *Scope* că răspunsul funcției de transfer se modifică în concordanță cu parametrii introduși.

În mod similar, rezultatele unei simulări pot fi exportate în *Workspace*-ul Matlab prin intermediul blocului *To Workspace* din categoria *Sinks*. Din fereastra de configurare a blocului, se poate alege un **nume** pentru variabila creată (*Variable name*), un **format de salvare** (*Save format*), **perioada de eșantionare** (*Sample Time*), dar și alți parametri de formatare a datelor.

În situația în care se va lăsa configurarea implicită și se va executa o simulare, în spațiul de lucru va fi creată o variabilă de *simout* ca parte a unei **structuri** de tip *SimulationOutput*. Pentru reprezentarea grafică a acestei variabile se poate apela:

```
plot(out.simout)
```

Prin comparare, graficul astfel obținut este identic cu graficul generat de blocul *Scope*.

Dacă se dorește reprezentarea grafică explicită a primului semnal, se va apela comanda:

```
plot(out.simout.Time, out.simout.Data(:,1))
```

1.6. CHESTIUNI DE STUDIAT

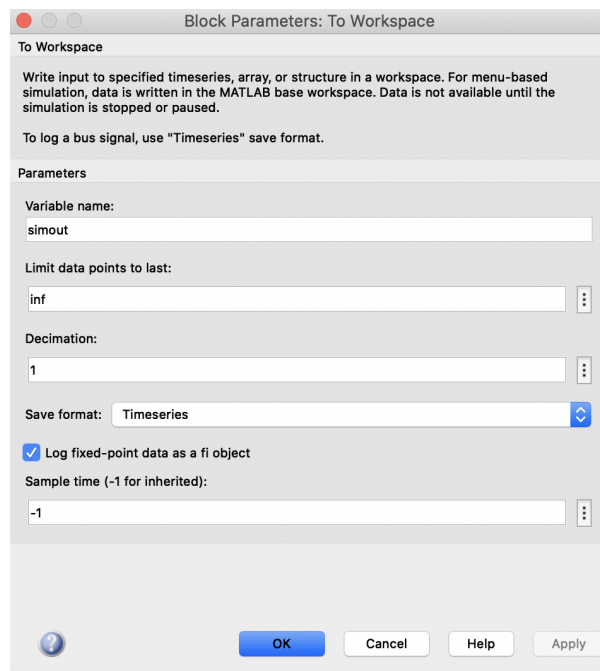


Figura 1.8: Configurarea blocului *To Workspace*

1.6 Chestiuni de studiat

1. Implementați modelul unui sistem care conține următoarele componente:

- Un generator pentru un semnal de intrare de tip treaptă dat de relația:

$$u(t) = \begin{cases} 2, & t \leq t_0 \\ 4, & t > t_0 \end{cases} \quad (1.3)$$

unde $t_0 = 5$ sec.

- O funcție de transfer de forma:

$$H(s) = \frac{as + 1}{cs^2 + ds + e} \quad (1.4)$$

unde $a = 1, c = 1, d = 2, e = 1$

Observație: coeficienții vor fi declarați **obligatoriu** în *Workspace*-ul Matlab și vor fi preluați mai departe în Simulink.

- Un amplificator (*Gain*) cu factor de amplificare 3. Notă: Amplificatorul se regăsește în categoria *Commonly Used Blocks*.
- Un multiplexor cu 3 intrări
- Un bloc *Scope*.

LABORATORUL 1. INTRODUCERE ÎN SIMULINK

- Un bloc *To Workspace*

Conexiunile dintre elemente vor fi implementate astfel:

- Generatorul de semnal la funcția de transfer.
- Generatorul de semnal la amplificator.
- Intrările multiplexorului sunt conectate la generatorul de semnal, la funcția de transfer și la amplificator
- Ieșirea multiplexorului se conectează la blocul de vizualizare a semnalelor și la blocul *To Workspace*.

2. Simulați schema astfel realizată pe un interval de timp de 20 de secunde.
3. În Matlab, definiți funcția de transfer din Ec. 1.4 și reprezentați grafic răspunsul în timp al modelului la un semnal de intrare de tip treaptă.
4. Reprezentați grafic în Matlab (într-o altă figură) rezultatele importate din Simulink.