

Noțiuni de utilizare a Matlab-Simulink/ Scilab-XCos

A. Noțiuni teoretice

Disciplina Teoria sistemelor integrează concepte, noțiuni și metode de lucru pentru transpunerea riguroasă a proceselor din natură într-o formă matematică necesară înțelegerii, analizării și găsirii de soluții de conducere și de monitorizare ale acestora.

În rezolvarea problemelor legate de analiza proprietăților sistemelor, de determinarea semnalelor de la ieșirea sistemelor la diferite semnale de intrare și de studiul regimurilor de funcționare, există mai multe pachete de programe dedicate. Unele dintre acestea sunt destinate calculului matematic în general sau specific diferitelor ramuri ale științei, (rezolvare de ecuații sau sisteme de ecuații, reprezentări grafice, analize de funcții): Wolfram Alpha, GNU Octave , Mathcad, Matlab, Scilab s.a. Pentru modelarea, implementarea și analiza comportării sistemelor sub formă de scheme bloc au fost dezvoltate instrumente specifice precum Simulink din cadrul Matlab sau Xcos din cadrul Scilab.

Lucrarea de față își propune trecerea în revistă a unui bagaj minimal de informații referitor la modul de lucru în **Matlab** și **Simulink**, respectiv **Scilab** și **Xcos**.

A1.1 Generalități Matlab-Simulink

În Matlab se poate lucra de la linia de comandă, introducând comenzile pe rând de la linia de comandă sau cu fișiere (Fig.1).

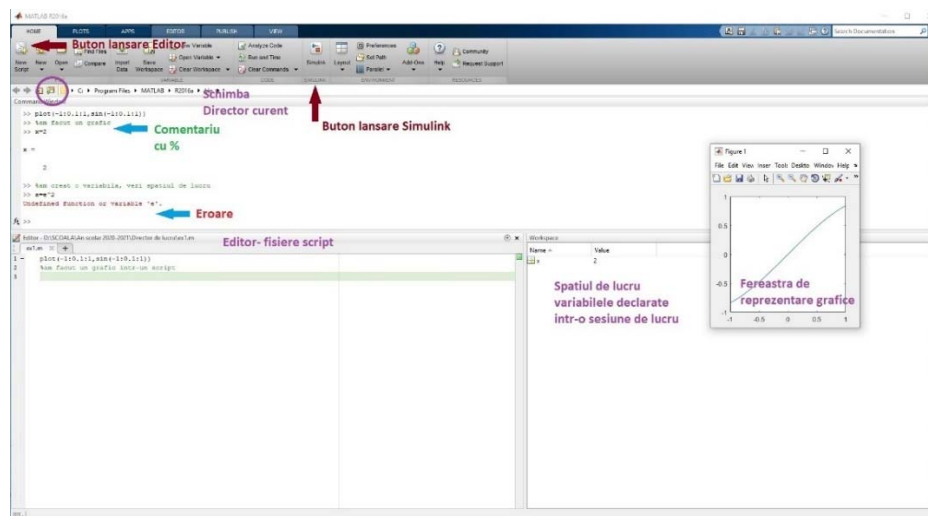


Fig.1 Fereastră Matlab

Variabilele care se creează într-o sesiune de lucru se salvează în spațiul de lucru al Matlab (*Workspace*). Odată ce sunt create ele pot fi folosite atât în alte instrucțiuni Matlab, cât și ca parametri pentru blocurile schemelor Simulink. La o nouă sesiune de lucru ele trebuie redeclarate. Se pot salva valorile variabilelor prin folosirea de fișiere *.mat*.

Pentru a vedea care sunt variabilele și valorile lor se deschide fereastra *Workspace* folosind butonul *Layout*. Pentru a lucra cu fișiere acestea trebuie să se găsească în directorul curent. Schimbarea directorului se face din partea de sus a ferestrei alegând calea dorită (Fig 1).

Tipuri de fișiere utilizate în Matlab

Fișiere script -extensia: <i>.m</i>	-colecție de comenzi executate secvențial; -se rulează scriind numele fișierului (fără extensie) de la linia de comandă sau cu F5 din fereastra de editare; -pentru a se rula directorul curent trebuie să fie cel în care este salvat fișierul.
Fișiere model -extensia: <i>.slx</i> ; <i>.mdl</i> (versiuni vechi);	-fișiere destinate realizării de scheme de simulare; -se pot folosi variabile salvate în spațiul de lucru al Matlab-ului pentru parametrizarea blocurilor; -execuția calculelor se face prin simulare folosind butonul de Start Simulation din cadrul ferestrei fișierului.
Fișiere figură -extensia: <i>.fig</i>	-salvează reprezentări grafice realizate prin folosirea funcției <i>plot</i> .
Fișiere de date -extensia: <i>.mat</i>	-salvează valorile atribuite variabilelor într-o sesiune de lucru; -se folosesc comenzile <i>save numefișier variabile</i> și <i>load numefișier</i> .

Important: Numele fișierelor începe cu o literă și nu conține caractere speciale (spațiu, *, -, etc.)

A1.2 Mod de lucru în Matlab. Fișiere script *.m*

Fișierele script *.m* sunt o colecție de comenzi care la rularea fișierului se execută secvențial. Variabilele create sunt salvate în spațiul de lucru al Matlab. Ele se suprascriu la atribuirea de noi valori și sunt păstrate în memorie pentru sesiunea de lucru al Matlab-ului.

Fișierele *.m* se creează prin deschiderea ferestrei de editare folosind butonul New Script (vezi Fig. 1 sau introducând de la linia de comandă *>>edit*. Fișierele script se pot crea și cu comanda *NEW File->Script*.

După scrierea programului, fișierul se salvează și se rulează din fereastra de editare cu F5 sau de la linia de comandă prin scrierea numelui fișierului (fără extensie). Pentru a se executa instrucțiunile programului, directorul curent trebuie să fie cel în care este salvat fișierul.

Opțional instrucțiunile se pot termina cu “;”, lucru care inhibă afișarea în *Command Window* a valorilor atribuite sau calculate pentru variabilele folosite în instrucțiuni. Erorile de sintaxă sau de utilizare greșită de tipuri de date se regăsesc în urma execuției fișierelor în *Command Window* cu culoare roșie. Pentru modul în care se fac operații în Matlab vezi **Anexa 3**.

În **Anexa 4** sunt amintite câteva funcții Matlab utile în rezolvarea cerințelor laboratorului.

A1.3. Mode de lucru în Simulink. Fișiere model .slx

Fișierele model permit lucrul cu blocuri. Pentru realizarea unui fișier model trebuie apăsat butonul Simulink Model (vezi Fig. 1) sau scriind de la linia de comandă `>>simulink`. Se deschide o fereastră de unde se pot alege mai multe variante de fișiere de lucru. Un fișier model nou se deschide alegând Bank Model (Fig. 2)

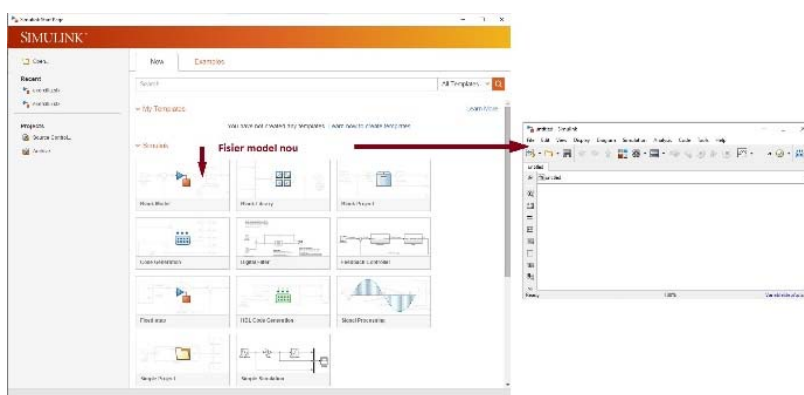



Fig.2 Deschidere fișier model

Pentru accesarea blocurilor din care sunt alcătuite schemele de simulare trebuie deschisă fereastra *Simulink Library Browser* (Fig.3), folosind icoana Simulink  din fișierul model.

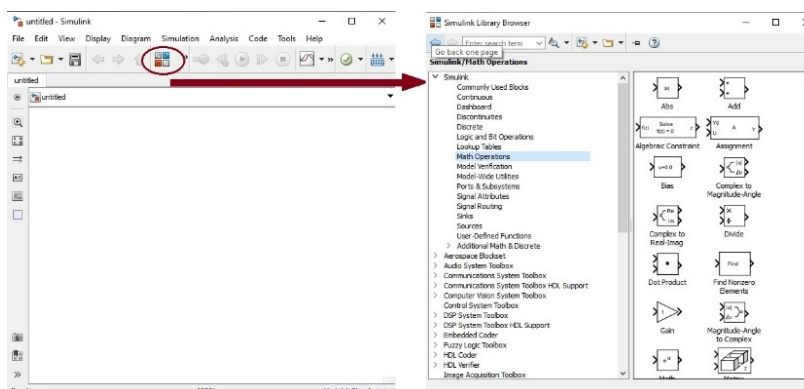


Fig.3 Fereastra Simulink Library Browser

Fereastra *Simulink* cuprinde o serie de blocuri generatoare de semnale, blocuri de afișare, blocuri care realizează diferite operații și funcții matematice.

Fișierele model se pot crea și cu comanda *NEW File->Simulink Model*. Din fereastra *Simulink* se aleg blocurile de care este nevoie în realizarea schemei și se trag cu mouse-ul (Drag and Drop) în fereastra modelului. Pentru ca informația să fie transmisă de la un bloc la altul, acestea trebuie să fie conectate. Conectarea blocurilor se face prin legarea porturilor de intrare ale blocurilor din aval de cele de ieșire a celor din amonte sau o conexiune deja făcută (Fig.4). Dacă legătura este cu linie punctată și roșie, conexiunea nu este realizată.

Majoritatea blocurilor sunt parametrizabile. Pentru a modifica parametrii se dă dublu-click pe bloc în fișierul model (Fig.4). Pentru parametrizare se pot folosi valori numerice sau variabile care au primit în prealabil valori în *Workspace*.

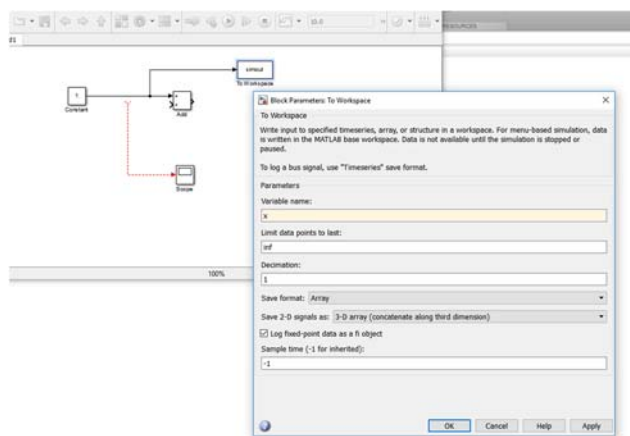




Fig.4 Exemplu de conectare blocuri și de parametrizare

Toate blocurile din care este alcătuită o schemă a unui fișier model lucrează în raport cu timpul de simulare, adică efectuează calculele sau furnizează rezultate la momente stabilite prin configurarea Simulatorului, pe o perioadă de timp stabilită (timp de simulare). Folosind din fișierul model meniul *Simulation->Model Configuration Parameters* se ajunge la fereastra de configurare a simulării unde se pot seta intervalul de timp de simulare prin Start Time și Stop Time, metoda de calcul numeric ce se dorește a fi folosită și/sau valoarea pasului de calcul. Astfel, blocurile lucrează cu semnale în loc de variabile, valorile fiind asociate momentelor de timp la care au fost calculate. Dacă pentru un bloc particular se dorește un anumit pas de calcul, acest lucru se setează la rubrica Sample Time. (Valoarea implicită "0" sau „-1” specifică că blocul lucrează cu pasul stabilit de simulator).

După realizarea schemei Simulink conform cerințelor date, simularea se realizează prin apăsarea butonului de Run  din fișierul model. Rezultatele se urmăresc prin folosirea de blocuri de afișare (la blocurile Scope trebuie dat dublu-click) sau prin transmiterea în *Workspace* a semnalelor sub formă de variabile (folosind blocul ToWorkspace)

În **Anexa 5** sunt trecute în revistă câteva blocuri Simulink utile în rezolvarea cerințelor laboratorului.

A2.1 Generalități Scilab-Xcos

La deschiderea Scilab-ului apare o fereastră precum cea din Fig. 5. Ea este împărțită în zone cu destinații diferite. Fereastra de Scilab poate fi configurată folosind butonul de Scilab Preferences .

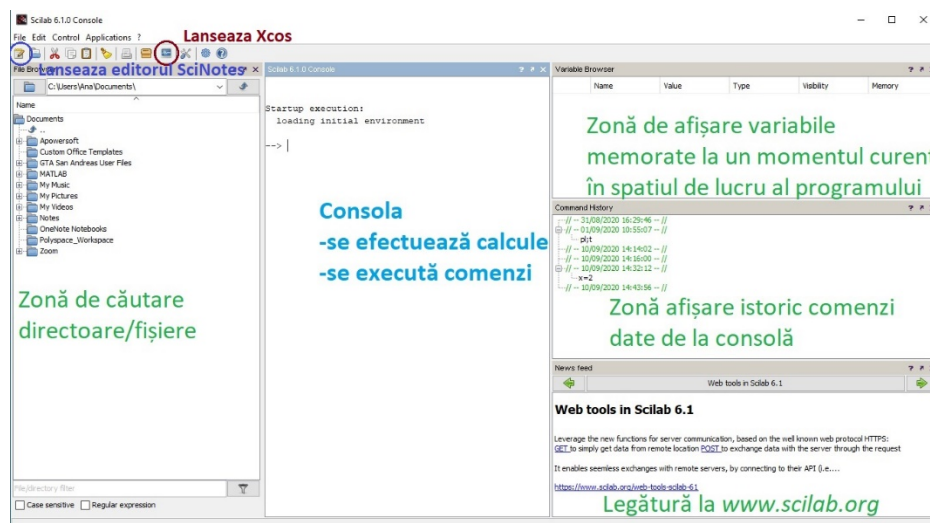





Fig.5 Fereastră Scilab

În Scilab se poate lucra din fereastra *Console* introducând instrucțiunile pe rând de la linia de comandă --> sau cu fișiere folosind editorul SciNotes . Dacă se lucrează din fereastra *Console*, opțional instrucțiunile se pot termina cu “;”, lucru care inhibă afișarea în *Console* a valorilor atribuite sau calculate pentru variabilele folosite în instrucțiuni. Erorile de sintaxă sau de utilizare greșită de tipuri de date se regăsesc în urma execuției fișierelor în *Console*.

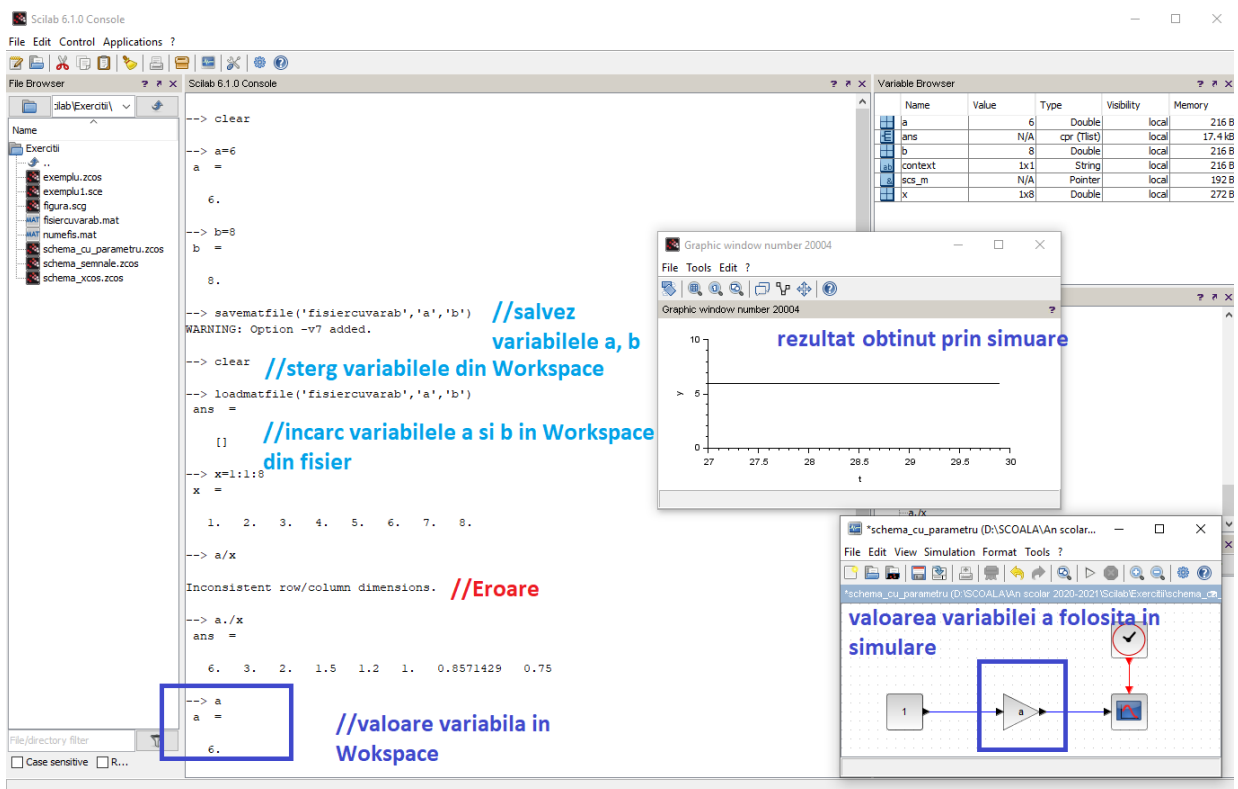
Variabilele care se creează într-o sesiune de lucru se salvează în spațiul de lucru al Scilabului (*Workspace*). Odată ce sunt create ele pot fi folosite atât în alte instrucțiuni Scilab, cât și ca parametri pentru blocurile schemelor Xcos. Schemele Xcos pot fi realizate prin inițializarea acestui simulator de la butonul . La o nouă sesiune de lucru ele trebuie redeclarate. Se pot salva valorile variabilelor prin folosirea de fișiere .mat.(Fig. 6)

Tipuri de fișiere utilizate în Scilab

Fișiere script -extensia: .sce	-colecție de comenzi executate secvențial; -se rulează scriind numele fișierului cu extensie de la linia de comandă sau cu F5 din fereastra de editare; -pentru a se rula directorul curent trebuie să fie cel în care este salvat fișierul.
-----------------------------------	--

Fișiere model -extensia: .zcos;	-fișiere destinate realizării de scheme de simulare; -se pot folosi variabile salvate în spațiul de lucru al Scilab-ului pentru parametrizarea blocurilor; -execuția calculelor se face prin simulare folosind butonul de Start Simulation din cadrul ferestrei fișierului  .
Fișiere figură -extensia: .scg	-salvează reprezentări grafice realizate prin folosirea funcției plot sau prin simulare.
Fișiere de date -extensia: .mat	-salvează valorile atribuite variabilelor într-o sesiune de lucru; -se folosesc comenzile <code>savematfile('numefișier','nume_var1,nume_var2,...')</code> și <code>loadmatfile('numefișier')</code> .

Important: Numele fișierelor începe cu o literă și nu conține caractere speciale (spațiu,*,-, etc.)



The screenshot displays the Scilab 6.1.0 interface with the following components:

- File Browser:** Shows a directory structure with files like `exemplu.zcos`, `exemplu.sce`, `figura.scg`, `fișiercuvrab.mat`, `numefs.mat`, `schema_cu_parametru.zcos`, `schema_semnale.zcos`, and `schema_xcos.zcos`.
- Console:** Contains the following commands and outputs:


```
--> clear
--> a=6
a =
6.

--> b=8
b =
8.

--> savematfile('fișiercuvrab','a','b') //salvez variabilele a, b
WARNING: Option -v7 added.                //sterg variabilele din Workspace

--> clear
--> loadmatfile('fișiercuvrab','a','b') //incarc variabilele a si b in Workspace
ans =
[]                                         //din fisier

--> x=1:1:8
x =
1. 2. 3. 4. 5. 6. 7. 8.

--> a/x
Inconsistent row/column dimensions. //Eroare

--> a./x
ans =
6. 3. 2. 1.5 1.2 1. 0.8571429 0.75


--> a
a =
6. //valoare variabila in Wokspace
```
- Variable Browser:** Lists variables in the workspace:

Name	Value	Type	Visibility	Memory
a	6	Double	local	216 B
ans	N/A	cpr (Tlist)	local	17.4 kB
b	8	Double	local	216 B
context	1x1	String	local	216 B
scs_m	N/A	Pointer	local	192 B
x	1x8	Double	local	272 B
- Graphic window number 20004:** Displays a plot of a constant value 6 over time t from 27 to 30. The plot is titled "rezultat obtinut prin simuare".
- *schema_cu_parametru (D:\SCOALA\An scolar...):** Shows a block diagram for a simulation. A block labeled 'a' is highlighted with a blue box, and a red arrow points to it with the text "valoarea variabilei a folosita in simulare".

Fig. 6 Declarare/salvare/utilizare variabile in Scilab

A2.2 Mod de lucru în Scilab. Fișiere scinotes .sce

Fișierele scinotes sunt o colecție de comenzi care la rularea fișierului se execută secvențial. Variabilele create sunt salvate în spațiul de lucru al Scilab. Ele se suprascriu la atribuirea de noi valori și sunt păstrate în memorie pentru sesiunea de lucru al Scilab-ului.


Fișierele se creează prin deschiderea ferestrei de editare folosind editorul SciNotes . (vezi Fig. 1) sau introducând de la linia de comandă --> *scinotes*.

După scrierea programului, fișierul se salvează și se rulează din fereastra de editare cu F5 sau de la meniul *Execute->Save and execute*.

Pentru modul în care se fac operații în Scilab vezi **Anexa 3**.

În **Anexa 4** sunt amintite câteva funcții Scilab utile în rezolvarea cerințelor laboratorului.

A2.3. Mode de lucru în Xcos. Fișiere model .xcos

Fișierele xcos permit lucrul cu blocuri. Pentru realizarea unui fișier model trebuie apăsat butonul Xcos  (vezi Fig. 5) sau scriind de la linia de comandă --> *xcos*. Se deschid două ferestre: *Palette Browser* care cuprinde o serie de blocuri generatoare de semnale, blocuri de afișare, blocuri care realizează diferite operații și funcții matematice din care se pot realiza scheme de simulare și un editor în care se poate realiza fișierul xcos. Fereastra *Palette Browser* se poate accesa din editor prin meniul *View-> Palette Browser* (Fig. 7)

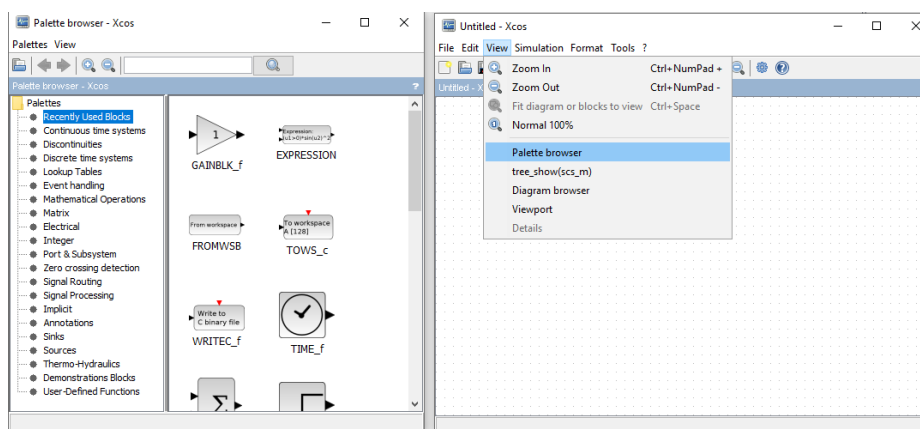


Fig.7 Deschidere fișier model

Din fereastra *Palette Browser* se aleg blocurile de care este nevoie în realizarea schemei și se trag cu mouse-ul (Drag and Drop) în fereastra modelului. Pentru ca informația să fie transmisă de la un bloc la altul, acestea trebuie să fie conectate. Conectarea blocurilor se face prin legarea porturilor de intrare ale blocurilor din aval de cele de ieșire a celor din amonte sau o conexiune deja făcută. Porturile negre (legăturile trasate cu linie albastră) reprezintă porturi de intrare ieșire

date, iar cea roșii (legături cu linie roșie) reprezintă porturi destinate evenimentelor. După realizarea legăturilor blocurile din schemă se pot selecta și alinia folosind meniul *Format->Auto-Position Block* (Fig.8).

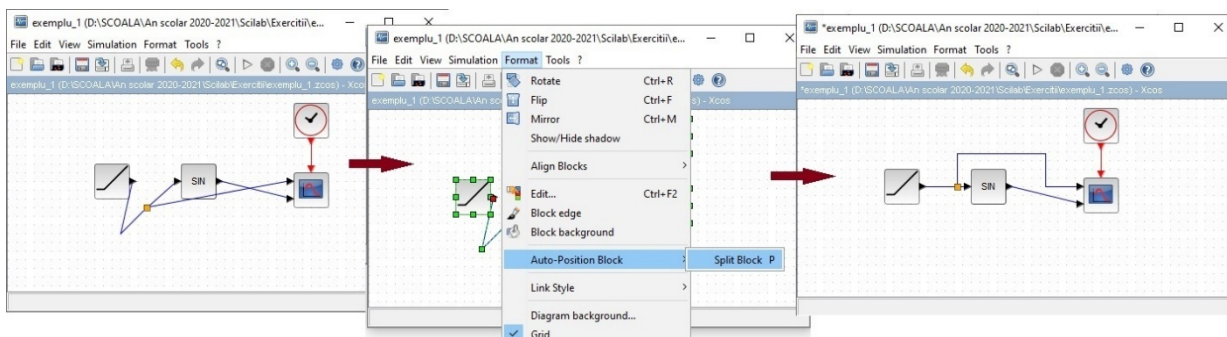


Fig.8 Exemplu de conectare și aliniere blocuri

Majoritatea blocurilor sunt parametrizabile. Pentru a modifica parametrii se dă dublu-click pe bloc în fișierul model (Fig.9).

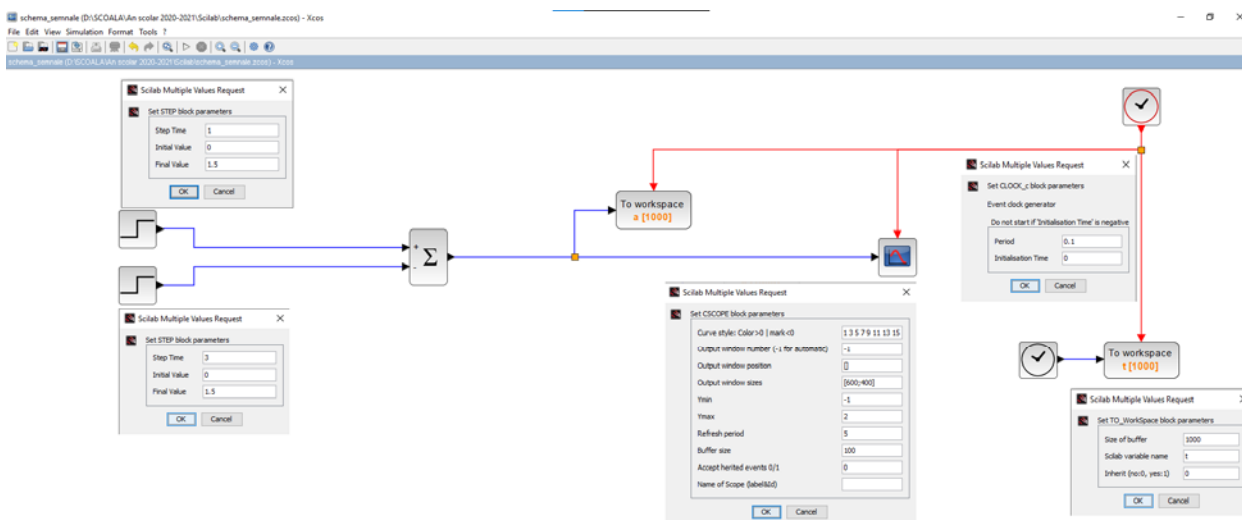


Fig.9 Exemplu de parametrizare blocuri

Toate blocurile din care este alcătuită o schemă a unui fișier model lucrează în raport cu timpul de simulare. Acest timp de simulare se setează de la meniul *Simulation->Setup*. În Fig. 10 s-a setat timpul de simulare ca fiind 10 sec.

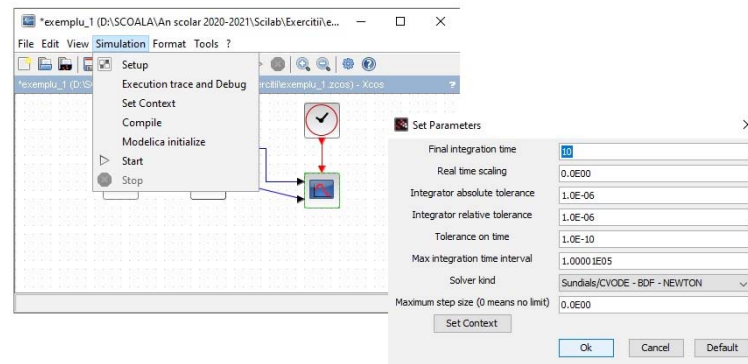



Fig. 10 Setare perioadă de simulare

Simularea se realizează apăsând pe butonul de Start  sau din meniul *Simulation->Setup*(Fig. 10). Rezultatul unei simulări poate fi sub forma unui grafic care este reprezentat cu blocuri de afișare (CSCOPE) sau a unei variabile ce poate fi trimisă în *Workspace* (TOWS_c). Aceste blocuri de afișare înmagazinează sau reprezintă valorile rezultatului în raport cu timpul de simulare la momente stabilite prin configurarea unui bloc de generare de evenimente (CLOCK_c).

În Fig. 11 se observă rezultatul simulării unui fișier care realizează graficul funcției $y_1(t) = \sin(t)$, $t \in [0,10]$. Valorile lui y_1 sunt preluate la fiecare 0.1sec. Blocurile TOWS_c trimit în *Workspace* variabilele t și y_1 ca variabile de tip structura. Pentru afișarea graficelor preluăm doar câmpurile cu valoarea variabilelor.

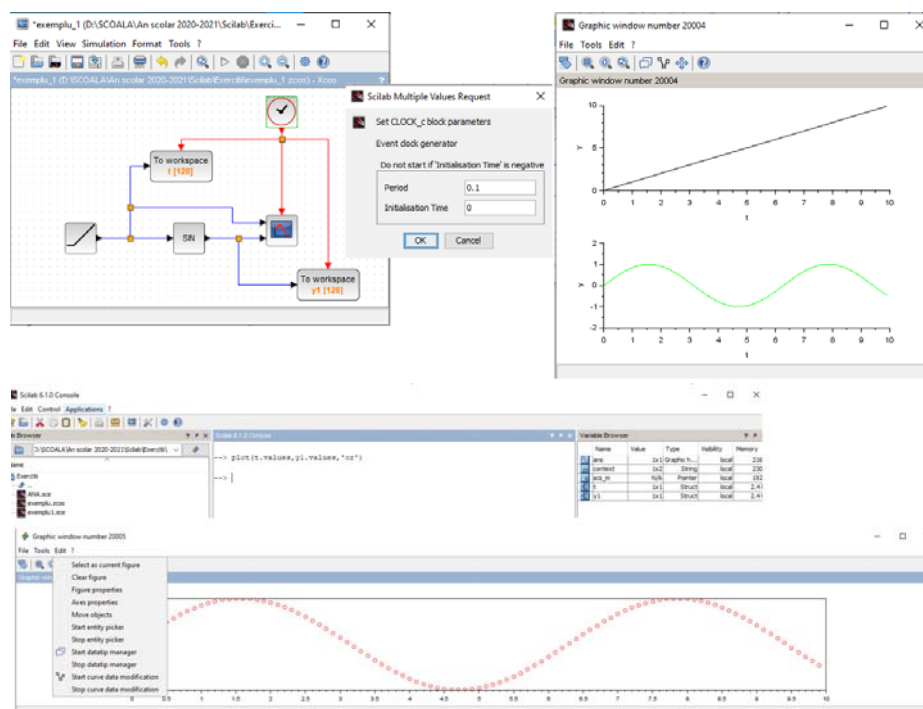


Fig. 11 Exemplu modului de afișare rezultate în urma unei simulări

Fig. 12 indică modul în care se poate prelucra reprezentarea grafică a semnalului obținută în simulării în Xcos.

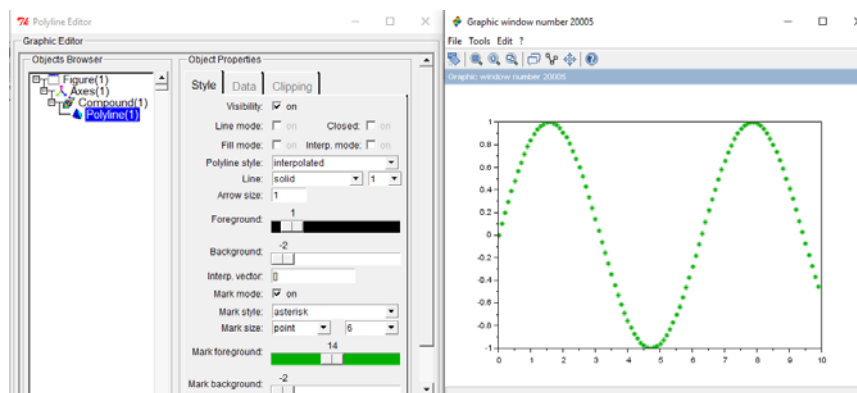


Fig. 12 Modul de editare grafic in Scilab

În **Anexa 5** sunt trecute în revistă câteva blocuri Xcos utile în rezolvarea cerințelor laboratorului.

B. Exerciții (Exemple rezolvate în Anexele 1 și 2)

1. Să se reprezinte graficele următoarelor patru funcții folosind fișiere Script în Matlab/Scilab:
 - a. $y_1 = 2x^2 + 5x + 1$, pentru $x \in [-3, 4]$.
 - b. $y_2 = 2e^{2x} + e^x + 3$, pentru $x \in [-3, 4]$.
 - c. $y_3 = \lg|x| + \sqrt{2}$, pentru $x \in [-3, 4] - \{0\}$.
 - d. $y_4 = \sin x + 16(\cos x)^2$, pentru $x \in [0, 2\pi]$.

Obs: Pentru constanta π în Matlab se folosește pi, iar în Scilab %pi

2. Să se scrie un program de calcul care să reprezinte grafic în același sistem de coordonate funcțiile de la punctul 1.
3. Să se scrie un program care să reprezinte grafic funcțiile de la punctul 1 în aceeași fereastră de afișare, dar în sisteme de coordonate diferite.
4. Să se construiască scheme bloc (folosind fișiere Model) prin care să se reprezinte grafic funcțiile de la punctul 1.
5. Să se realizeze o schemă de Simulink/Xcos care să reprezinte grafic funcția:

$$f = x^2 + 2x + a, \quad x \in [-5, 5], \quad a \in \mathbb{R}$$
 unde parametrul a ia pe rând valorile $\{1, 5, 10\}$. Reprezentarea grafică a celor trei variante se realizează folosind un fișier .m care apelează schema de simulare pentru fiecare valoare a parametrului a .

C. Anexe

1. Exemple de programe Script pentru rezolvarea pct. B1.

- a. Secvența de mai jos reprezintă graficul funcției: $y = \frac{e^x}{50} + x^2$, pentru $x \in [-5, 4]$ în Matlab.

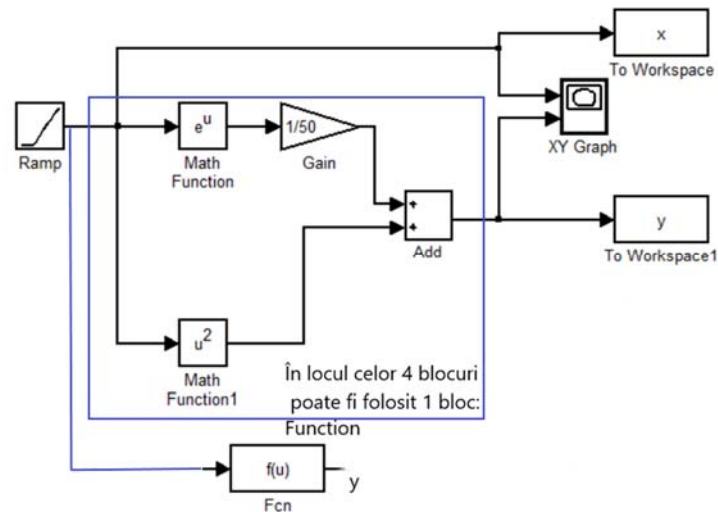
```
% procentul e folosit pentru comentarii
clc
clear
x=-5:0.1:4
y=exp(x)/50+x.^2
plot(x,y,'m')
title('De exemplu')
xlabel('x')
ylabel('y')
grid
```

- b. Secvența de mai jos reprezintă graficele funcțiilor: $y_1 = \sin\left(\frac{x}{2} + \frac{\pi}{2}\right)$, și $y_2 = \sqrt{|x|}$ pentru $x \in [-10, 10]$ separat în aceeași fereastră de afișare în Scilab

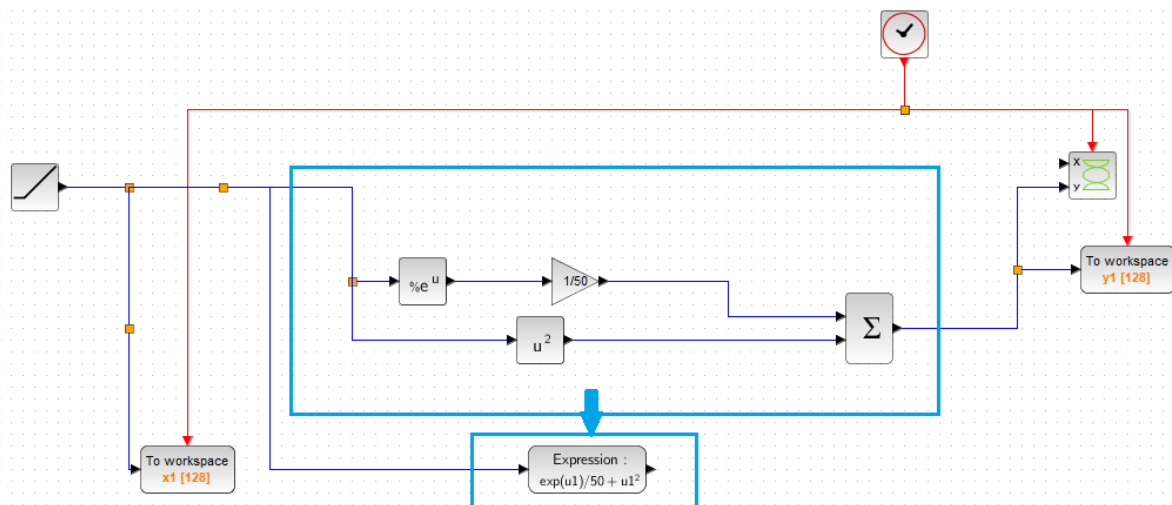
```
scf(1) //deschide fereastra de afisare numerotată cu 1; a doua fereastră se deschide cu scf(2)
clf(1) //aceasta comanda sterge graficele anterioare din fereastra numărul 1
x=-10:0.1:10;
y1=sin(0.5*x+%pi/2);
y2=sqrt(abs(x));
subplot(1,2,1)
title('Grafic 1')
plot(x,y1,'b-.')
subplot(1,2,2)
plot(x,y2,'co')
xlabel('x')
ylabel('y2')
```

2. Exemple de scheme de simulare pentru rezolvarea pct. B4.

- a. Schema de mai jos reprezintă graficul funcției: $y = \frac{e^x}{50} + x^2$, pentru $x \in [-5, 4]$ și trimite valorile lui x și y în *Workspace*. Este realizată în Simulink



- b. Schema de mai jos reprezintă graficul funcției: $y = \frac{e^x}{50} + x^2$, pentru $x \in [-5, 4]$ și trimite valorile lui x și y în *Workspace*. Este realizată în Scilab



3. **Matlab-ul** precum si **Scilab-ul** folosește diferite tipuri de date. În cadrul laboratorului se vor folosi predominant variabile numerice care se declară prin atribuirea de valori:

Ex. `a=2;` `a=[1,2,3] %vector linie;` `a=[1 2;3 4] %matrice cu 2 linii si 2 coloane`

Un interval se declară prin crearea unui vector care să cuprindă cât mai multe valori din intervalul respectiv.

Ex. $a \in [-10, 10]$ se declară astfel: `a=-10:0.1:10` % variabila creată este un vector cu elementele `[-10, -9.9, -9.8,..., 9.9,10]`. Cu cât pasul ales (în exemplu dat 0.1) este mai mic, cu atât numărul de valori preluate din interval sunt mai multe.

Operațiile matematice se aplică în funcție de tipurile de date folosite. Astfel, operatorii de înmulțire `"*"` (ridicarea la o putere `"^"`) și împărțire `"/"` aplicate variabilelor de tip scalar realizează operații scalare, iar celor de tip vectorial, operații vectoriale. Dacă se dorește ca o operație să fie realizată asupra elementelor vectorului/matricei atunci în fața operatorului se adaugă `"."`

Ex. `a=[1 2; 3 4];`

`a*a=[7 10; 15 22]`%înmulțire matriceală

`a.*a=[1 4; 9 16]`%înmulțește fiecare element cu el însuși

Important: Matlab-ul si Scilab-ul sunt Case Sensitive!







4. Funcții în Matlab/Scilab și comenzi folosite în mod uzual




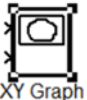
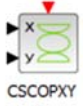
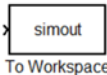



Funcția	în Matlab	în Scilab	Funcția	în Matlab	în Scilab
e^x	<code>exp(x)</code>	<code>exp(x)</code>	$\sin x$	<code>sin(x)</code>	<code>sin(x)</code>
$ x $	<code>abs(x)</code>	<code>abs(x)</code>	$\cos x$	<code>cos(x)</code>	<code>cos(x)</code>
\sqrt{x}	<code>sqrt(x)</code>	<code>sqrt(x)</code>	$\operatorname{tg} x$	<code>tan(x)</code>	<code>tan(x)</code>
$\lg(x)$	<code>log10(x)</code>	<code>log10(x)</code>	$\ln(x)$	<code>log(x)</code>	<code>log(x)</code>





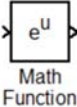








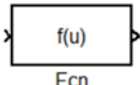
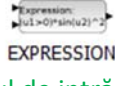
Comenzi/Funcții Matlab/Scilab	Descriere
<code>clc</code>	Șterge <i>Command Window</i>
<code>clear</code>	Șterge toate variabilele din <i>Workspace</i>
<code>clear a b</code>	Șterge toate variabilele a și b din <i>Workspace</i>
<code>plot(x,y,'c')</code>	Reprezintă graficul $y(x)$ cu culoarea/marcajul stabilit de caracteristicile 'c'. (help plot de la linia de comandă pentru a vedea ce caracteristici pot fi folosite) În Matlab execuția a două instrucțiuni plot succesive are ca efect afișarea doar a celui de-al doilea grafic. Pentru păstrarea graficului se folosește <i>hold on</i> . În Scilab execuția a două instrucțiuni plot succesive are ca efect afișarea celor două grafice suprapuse. Pentru ștergerea ferestrei de afișare se folosește <i>clf</i> .
<code>subplot(a,b,c)</code>	Se folosește când se dorește reprezentarea a mai multe grafice în aceeași fereastră de afișare în zone diferite. (Se folosește înainte de plot).

	a-numărul de linii; b-numărul de coloane în care se împarte fereastra; c-poziția de afișare (se numără pe linie)
<code>title('Titlu')</code>	Afișare titlu grafic (Se folosește după plot)
<code>xlabel('nume axa x')</code>	Afișare etichetă axa absciselor (Se folosește după plot)
<code>ylabel('nume axa y')</code>	Afișare etichetă axa ordonatelor (Se folosește după plot)
<code>hold on/off- Matlab</code>	Determină păstrarea graficelor reprezentate într-o fereastră de afișare când hold este on. Implicit e off, astfel că la fiecare instrucțiune plot se șterge graficul reprezentat de un plot anterior.
<code>scf(n) - Scilab</code>	Deschide o fereastră de afișare grafice sau se specifica în ce fereastră se vor afișa următoarele grafice făcute cu plot
<code>clf(n)- Scilab</code>	Șterge graficele afișate în fereastră nr. n (deschisa cu <code>sfc(n)</code>)
<code>grid- Matlab, xgrid- Scilab</code>	Trasează linii punctate la reprezentările grafice.
<code>sim('numefisiermodelfaraextensie')</code> <code>xcos("numefisxcos.zcos")</code>	Determină simularea schemei din fișierul Model cu numele precizat

5. Blocuri Simulink/Xcos folosite în mod uzual

Categorie	Bloc Simulink	Bloc Xcos	Scop/ Utilizare
Sources Generatoare de semnal de diferite forme. Blocurile doar port de ieșire.	 Constant	 CONST	Generează semnal de valoare constantă. Această valoare se poate seta.
	 Ramp	 RAMP	Generează semnal rampă. Are trei parametri: - Slope=panta cu care semnalul crește/descrește. - StartTime=momentul de la care semnalul începe să crească/descrească. - OutputValue/ InitialValue =valoarea inițială a semnalului
	 Clock	 TIME_f	Generează ca semnal de ieșire timpul de simulare.

		 <p>CLOCK_c</p> <p>Generator evenimente la intervale fixe de timp. Semnalele generate sunt folosite pentru toate blocurile de afișare</p>	
Sinks Blocuri de afișare semnale /stocare semnale sub formă de variabile în <i>Workspace</i> Blocurile au doar porturi de intrare	 <p>Scope</p>	 <p>CSCOPE</p> <p>Xcos:Trebuie setat domeniul de reprezentare: Refresh period=timp maxim pe orizontala; y_{min}, y_{max}=Min și Max pe verticală. Pot fi setate și alte caracteristici ale ferestrei de afișare.</p>	Afișează semnalul de la portul de intrare în raport cu timpul de simulare.
	 <p>XY Graph</p>	 <p>CSCOPXY</p>	Afișează semnalul de la al doilea port de intrare (y) în raport cu semnalul de la primul port (x). Trebuie setat domeniul de reprezentare (extensiile pe cele doua axe [x_{min} , x_{max} , y_{min} , y_{max}])
	 <p>simout To Workspace</p> <p>Se setează</p> <ul style="list-style-type: none"> - VariableName =numele variabilei. - SaveFormat =tipul de dată în care se salvează semnalul. <p>Pentru lucrul cu vectori -> SaveFormat: ARRAY</p>	 <p>TOWS_c</p> <p>Se setează</p> <ul style="list-style-type: none"> - ScilabVariableName =numele variabilei. - Variabilele se salvează automat de tip structură. Pentru a accesa valoarea salvata se utilizează: Numele_var.values 	Transmite valorile semnalului obținute prin simulare sub forma unei variabile cu numele "simout"/"A".
Math Operations/ Mathematical Operations Blocuri care realizează	 <p>Add</p> <p>Același lucru realizează și blocurile Sum/ Substract.</p>	 <p>SUMMATION</p> <p>Pentru mai multe intrări și semnul acestora se completează câmpul</p>	Realizează suma/diferența a două sau mai multe semnale.

diferite operații sau funcții matematice. Blocurile au porturi de intrare și port de ieșire.	Se setează numărul de intrări și semnul acestora prin completarea cu "+---+" a ListOfSigns	Number_of_inpus/sign_vector cu 1 pentru intrare cu plus și -1 pentru intrare cu minus	
	 Product		Realizează produsul a două sau mai multe semnale. Se setează numărul de intrări.
	 Divide	Number_of_inputs se completează cu */**/... Number_of_inpus/sign_vector se completează cu 1 pentru înmulțire și -1 pentru împărțire	Realizează succesiv operații de înmulțire sau împărțire în funcție de setări
	 Gain	 GAINBLK_f	Amplifică semnalul de intrare cu valoarea care se setează ca parametru al blocului.
	 Math Function	 EXPBLK_m  POWBLK_f  LOGBLK_f  SQRT	Aplică diferite funcții matematice asupra semnalului de la intrare: Exponențială, Ridicarea la o putere sau la pătrat, Logaritm natural sau zecimal, s.a
	 Abs	 ABS_VALUE	Realizează modulul semnalului de intrare.
	 Trigonometric Function	 TrigFun	Aplică diferite funcții trigonometrice asupra semnalului de la intrare
User Defined Functions	 Fcn Blocul consideră semnalul de intrare ca fiind "u". Expresia funcției trebuie introdusă folosind "u" ca variabilă.	 EXPRESSION Numărul de intrări e configurabil. Intrările sunt cunoscute în funcție ca u1, u2...	Blocul permite introducerea unei funcții de una sau mai multe variabile.