

# **Utilizarea programului Simulink pentru analiza circuitelor analogice**

Coordonatori:

Prof. dr. ing. Mihai Iordache,

Prof. dr. ing Lucia Dumitriu,

As. dr. ing. Dragoș Niculae

## ***Proiect realizat de:***

- Merișanu Bogdan
- Ignat Mihai
- Gagiu Cristian
- Avrămescu Bogdan
- Matcaboji Ștefan
- Bozeanu Ana
- Pâslaru Roxana
- Frangulea Andrei

**Facultatea de Automatică și Calculatoare**

**Abstract:** Scopul acestei lucrări este de a arăta cum se folosește Simulink pentru a modela un circuit liniar.

## I. Ce este Simulink?

Simulink (extensie a mediului MATLAB) este un mediu pentru modelarea, analiza și simularea unui număr mare de sisteme fizice și matematice.

Ca extensie opțională a pachetului de programe MATLAB, Simulink oferă o interfață grafică, pentru realizarea modelelor sistemelor dinamice reprezentate în schema bloc. O bibliotecă vastă, cuprinzând cele mai diferite blocuri stă la dispoziția utilizatorului. Aceasta permite modelarea rapidă și clară a sistemelor fără a fi necesară scrierea unui rând de cod de simulare.

## II. Construirea de modele

**Elementel model** sunt adăugate prin selectarea elementelor dorite din bibliotecă și trăgându-le în fereastră. Alternativ, pot fi copiate din bibliotecă și puse în fereastră.

### A. Modelarea unui sistem condensator

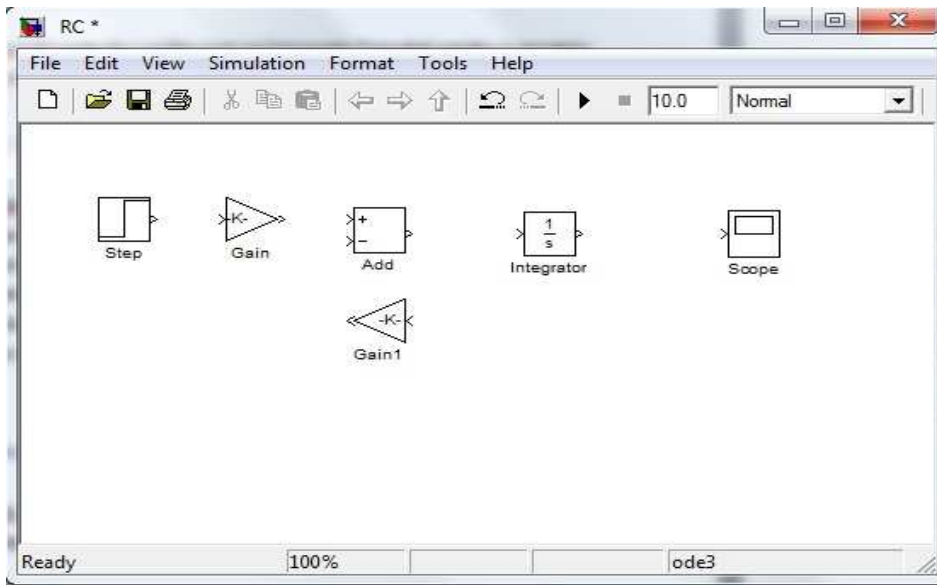
Pentru a modela sistemul alegem  $s=d/dt$  și rescriem ecuația diferențială astfel:

$$sV_c(s) = -\frac{1}{RC}V_c(s) + \frac{1}{RC}U(s)$$

Rezultă:

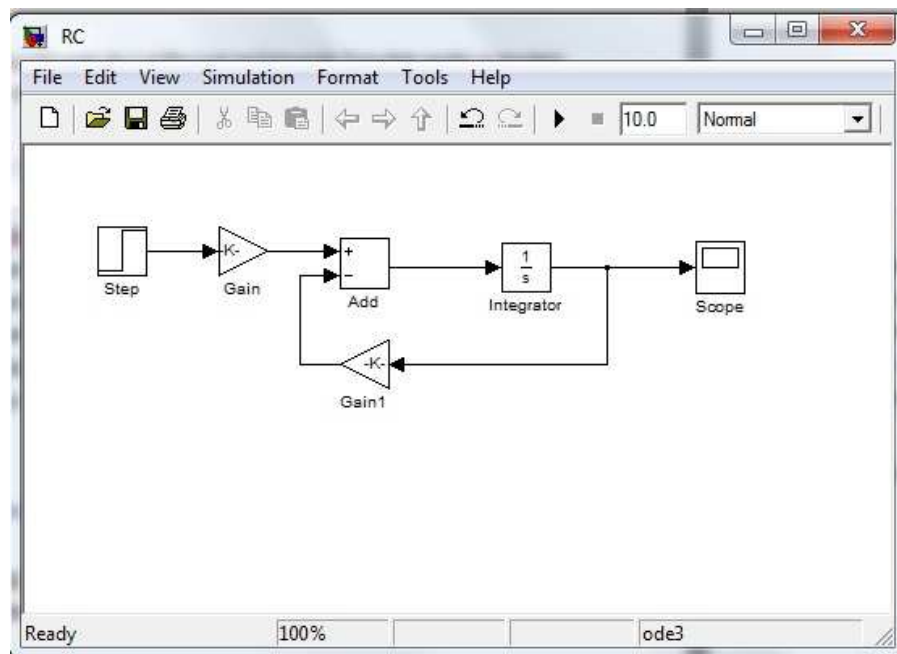
$$V_c(s) = \frac{1}{s} \left( \frac{1}{RC}U(s) - \frac{1}{RC}V_c(s) \right)$$

Model de sistem neconectat:

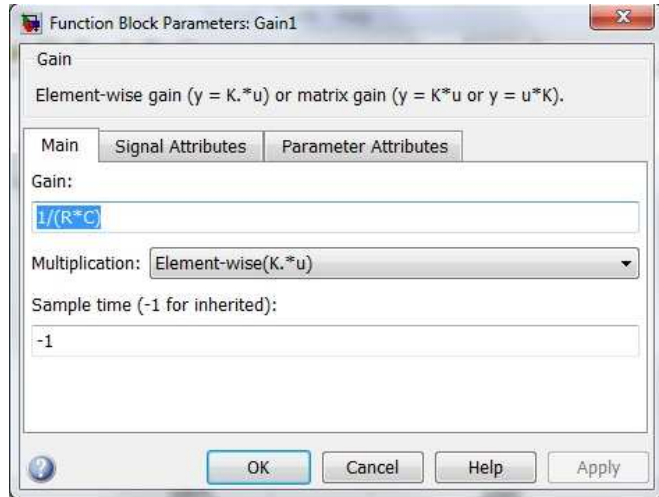
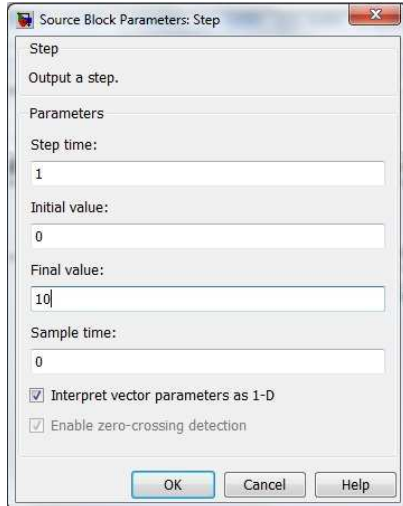


**Blocurile de sistem** pot fi conectate prin metoda conectării în mod automat, care se poate realiza selectând blocul sursă, ținând apăsată tasta CTRL și selectând blocul de destinație. Simulink îți permite să desenezi manual linii între blocuri. Se apasă click dreapta și se trage către intrarea în blocul de destinație pentru a realiza conexiunea.

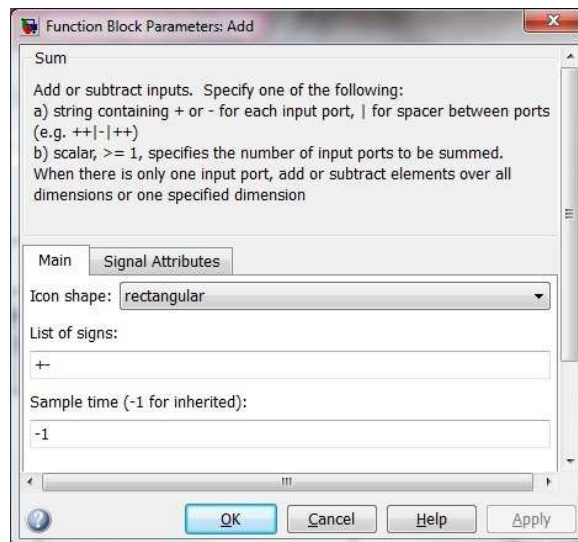
Modelul de sistem conectat:



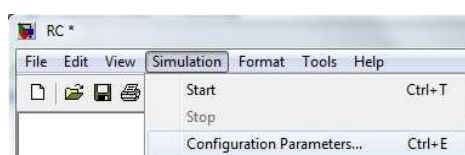
Acum că avem conectat modelul, trebuie să îl edităm. Aceasta se realizează selectând modelul și apoi dublu click pentru a-l deschide și edita. Blocurile pe care trebuie să le edităm sunt: funcția step, cele două blocuri gain și blocul joncțiunii de suma.

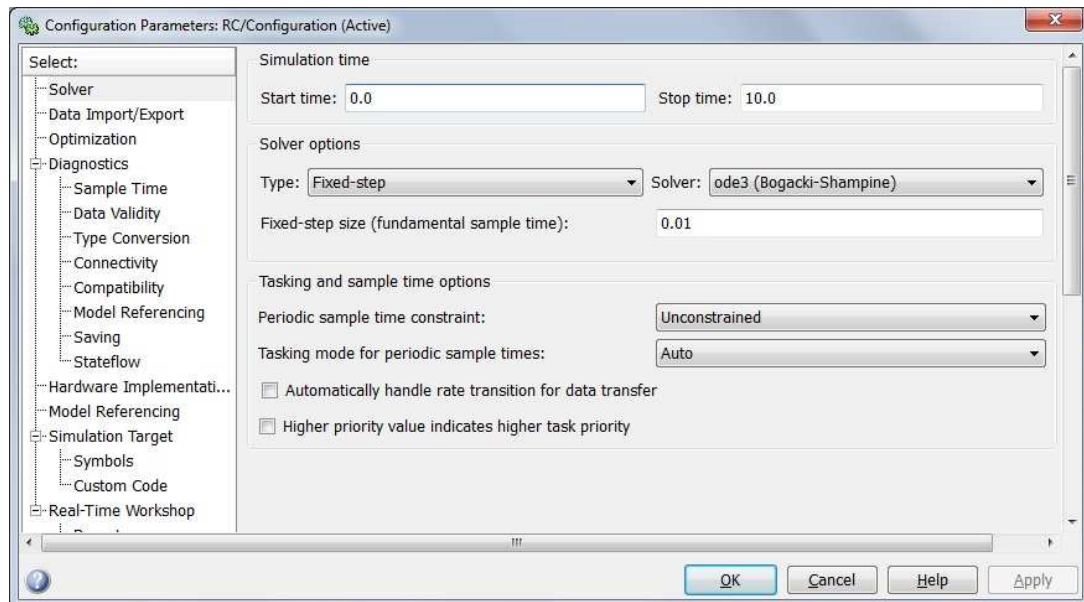


În editarea notei blocurilor se setează înălțimea funcției pas la 10, în stânga R și C ca variabilă (le vom introduce pe acestea din spațiul de lucru), și se modifică semnul joncțiunii de la + la -.



Controlul simulării este fixat prin selectarea parametrilor configurației.



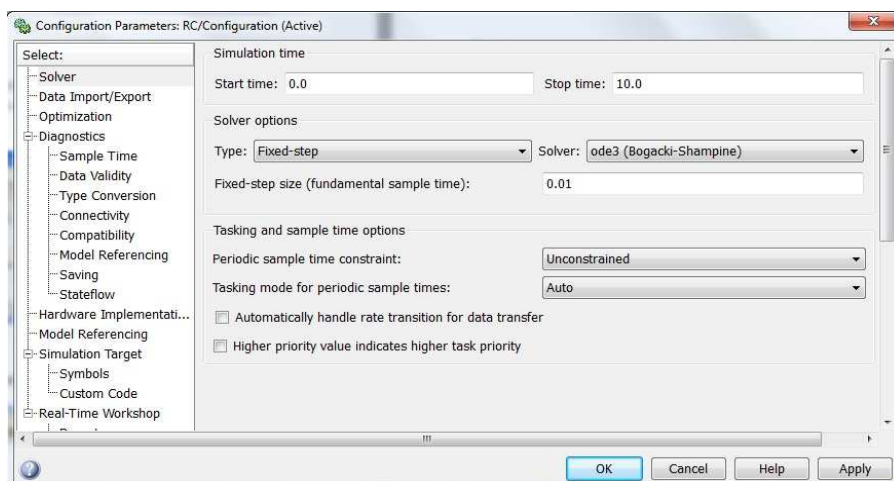
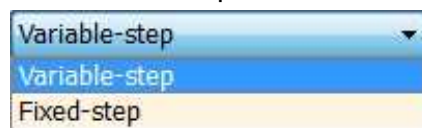


**Simulink** folosește integrarea numerică pentru a rezolva sisteme dinamice de ecuații. **Solver** este motorul folosit pentru integrarea numerică. Parametrii cheie sunt timpii de început și de sfârșit, și tipul și metodele de Solver.

Tipul Solver setează metoda de integrare numerică ca pas variabil sau pas fix. Pasul variabil ajustează continuu mărimea pasului de integrare pentru a scurta timpul de simulare. I.E, variabila step reduce mărimea pasului când stările modelului se schimbă rapid, pentru a menține acuratețea și pentru a crește mărimea pasului când stările sistemului se schimbă lent pentru a evita pași inutili.

Pasul fix folosește aceeași mărime de-a lungul simulării. La alegerea pasului fix este necesară stabilirea mărimii pasului.

Următorul exemplu ilustrează setarea mărimii pasului fix la o valoare de 0.01 secunde.

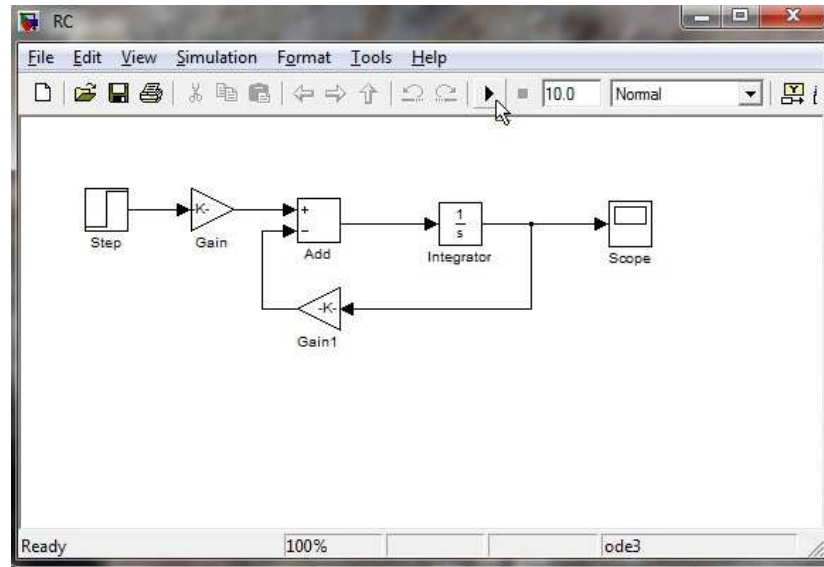


Pentru a rula simularea, în primul rând trebuie să introducem valorile lui R și C. Puteam să le introducem direct în blocurile gain, dar am ales să le introducem din spațiul de lucru. Pentru a face asta în spațiul de lucru introducem

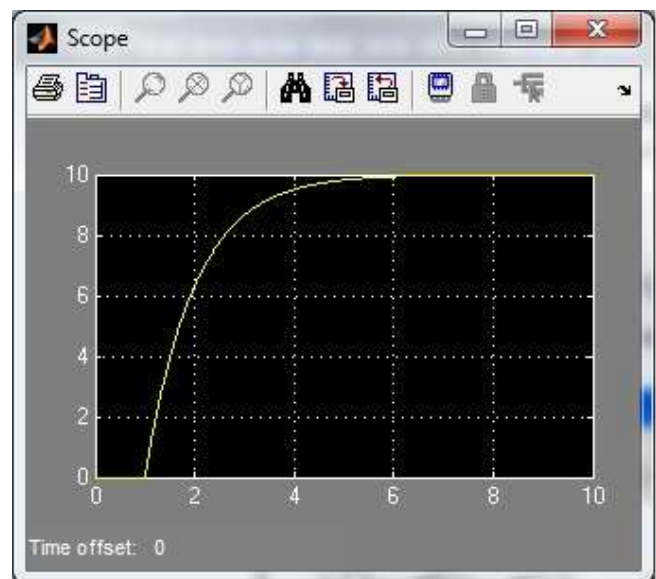
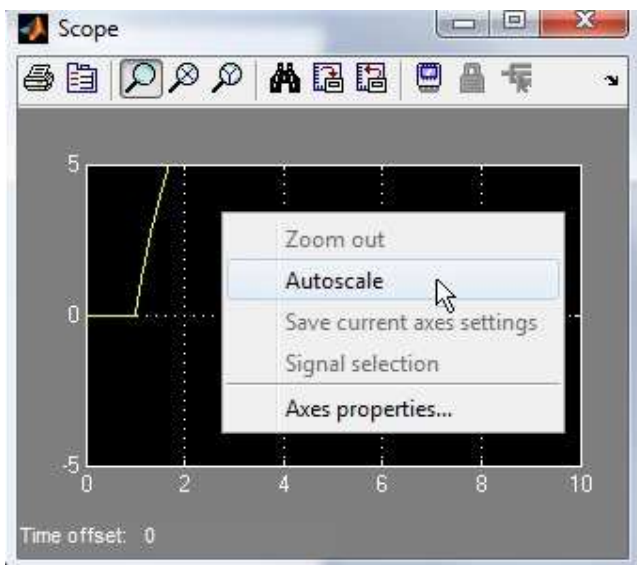
```
>>R=1;C=1;
```

Aceasta va fixa blocurile gain astfel încât  $K=1/(R*C)$

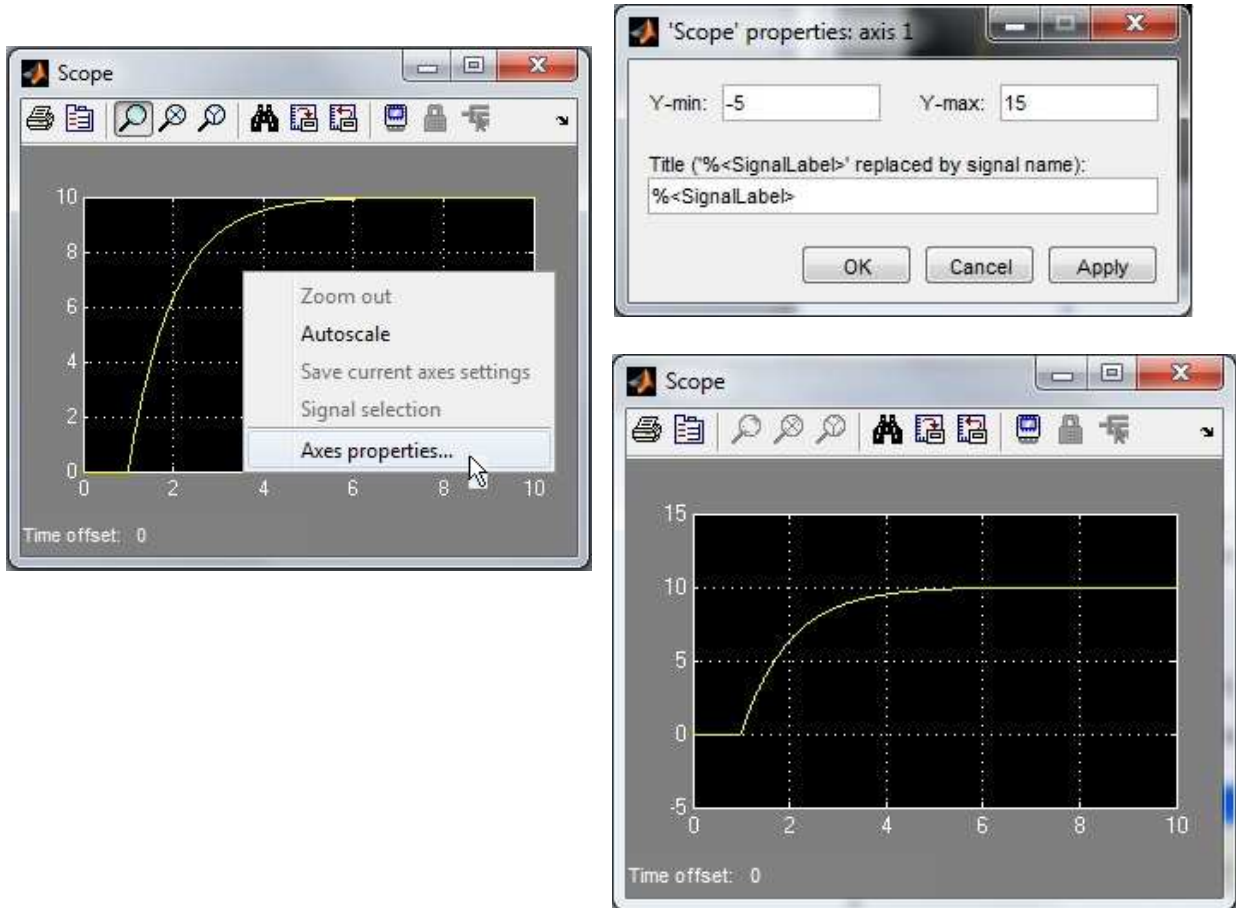
Acum selectăm și apăsăm butonul Run pentru a rula simularea.



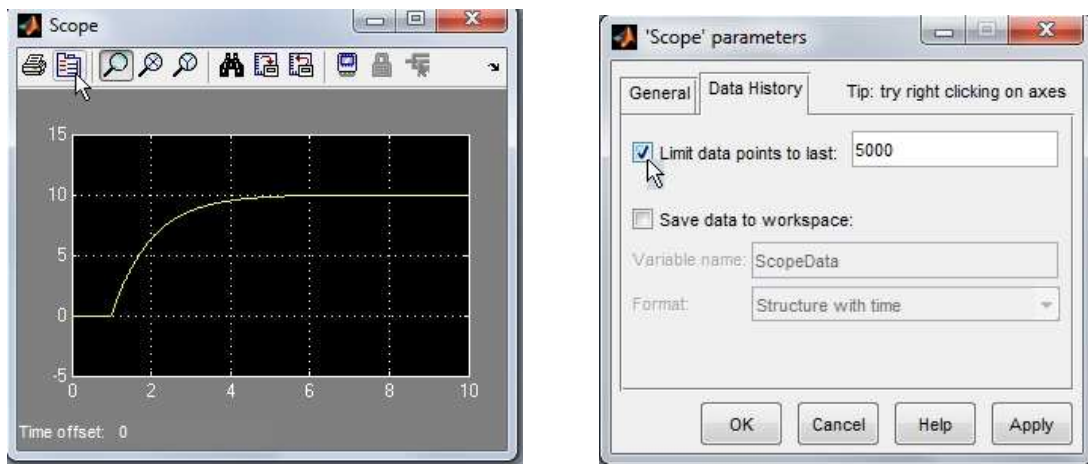
Pentru a vedea rezultatele simulării se alege blocul Scope. Pentru o vedere mai bună, se selectează graficul și după Autoscale.



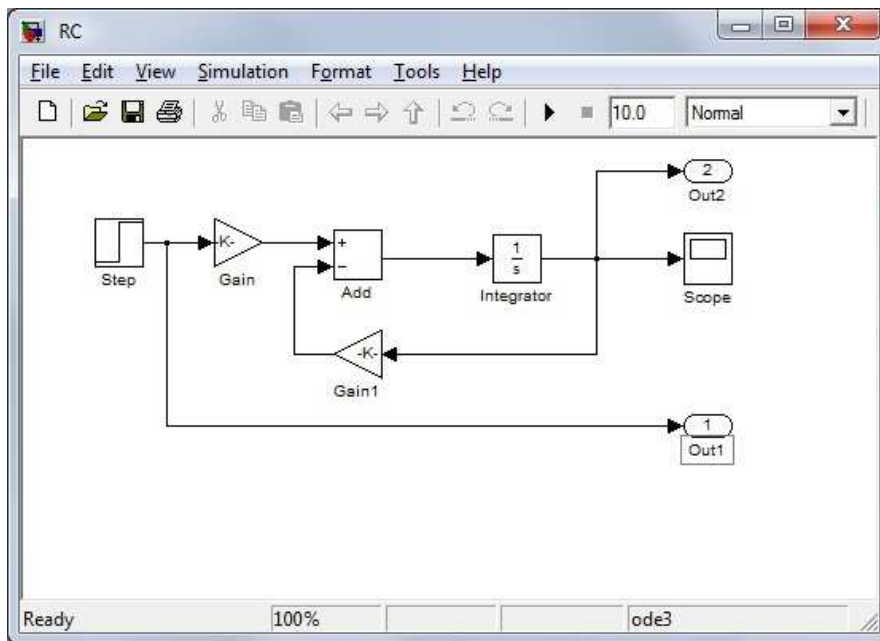
Se pot ajusta proprietățile axelor blocului Scope pentru a seta limitele axelor.



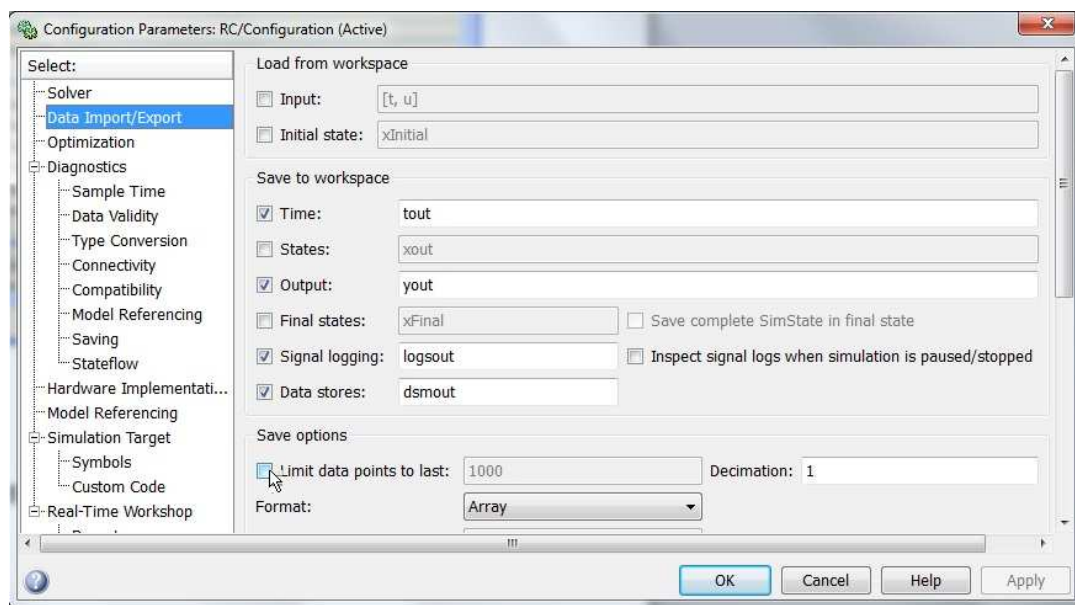
Setările prestabilite pentru Scope reprezintă o istorie de 5000 de mostre de date. Prin selectarea Parametrii și deselectând "Limitează punctele să păstreze" 5000, putem afișa un număr nelimitat de mostre simultane. Este util pentru simulări lungi sau simulări cu pași foarte mici



Adesea dorim să salvăm date pe spațiul de lucru și să folosim comenzile de trasare MATLAB pentru afișarea datelor.. Pentru aceasta putem folosi blocul Sink Out1 pentru a capta datele pe care dorim să le afișăm. În acest exemplu vom conecta blocuri Out atât la intrare cât și la ieșire:



Pentru a vedea unde datele de ieșire sunt salvate, deschidem Solver și selectăm Data Import/Export și deselectăm “Limitează punctele”. Timpul arătat este notat cu *tout*, iar ieșirile cu *yout*.





Acum rulăm simularea, iar pe spațiul de lucru tastăm *whos*

```
>>whos

Name          Size          Bytes  Class

C              1x1             8  double array
R              1x1             8  double array
tout          1001x1          8008  double array
yout          1001x2          16016  double array
```

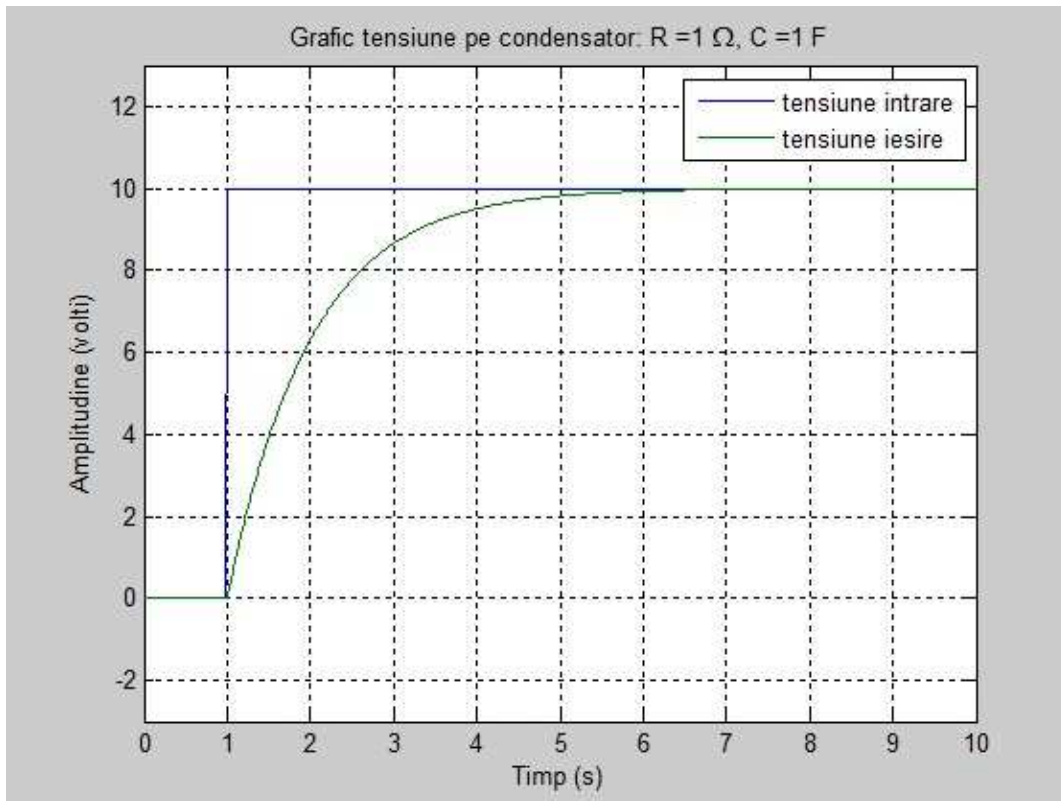
De reținut că *yout* este un vector 1001 pe 2, asta înseamnă că

*yout(:,1)*=mostre de intrare

*yout(:,2)*=mostre de ieșire

Pentru a trasa graficul, executăm scriptul M-File ca mai jos:

```
' M-File script to plot simulation data '
R = 1; C = 1; % Setare parametrii circuit
sim('RC'); % Lansare simulare
plot(tout,yout(:,1),tout,yout(:,2)); % Valori de trasat
xlabel('Timp (s)'); grid;
ylabel('Amplitudine (volți)');
st1 = 'Grafic tensiune pe condensator: R = '
st2 = num2str(R); % Conversie valoare R in string
st3 = ' \Omega'; % Simbol grecesc pentru Ohm
st4 = ', C = ';
st5 = num2str(C);
st6 = ' F';
stitle = strcat(st1,st2,st3,st4,st5,st6); % Titlu grafic
title(stitle)
legend('input voltage','output voltage')
axis([0 10 -3 13]); % Seteaza axele pentru tensiune de la -3 la 13
```



### Funcția SIM

Să presupunem că dorim să vedem cum tensiunea pe condensator se modifică pentru mai multe valori ale lui C. Pentru aceasta vom folosi comanda **sim**.

Se execută scriptul M-File. Bucla for cu comanda sim rulează modelul cu o valoare a condensatorului diferită de fiecare dată când comanda sim este executată. Datele de ieșire sunt salvate ca vc(k,:).

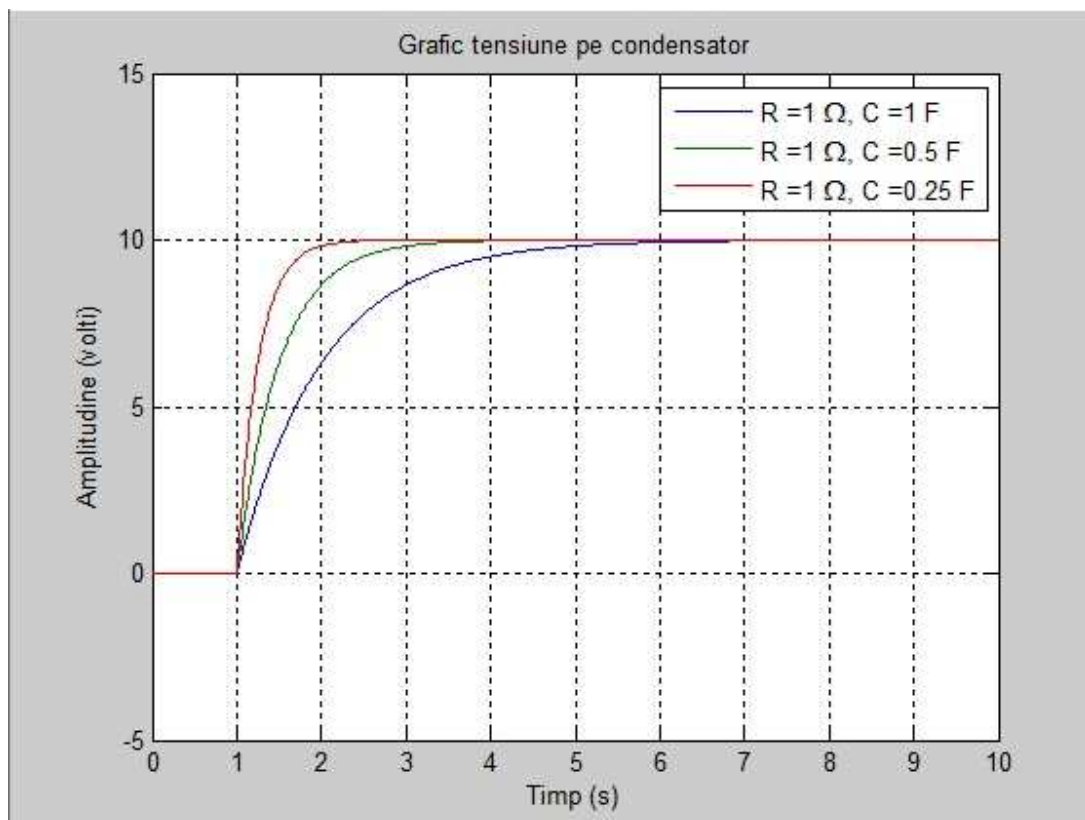
Comanda de trasare este folosită pentru a trasa datele pentru cele trei valori diferite ale condensatorului.

```
' M-File script to plot simulation data '
R = 1; % Setare valoare rezistor
CAP = [1 1/2 1/4]; % Valori pentru care se va face simularea
(Condensator)
for k = 1:3
C = CAP(k); % Citire valori simulare pentru condensator
sim('RC'); % Lansare simulare
vc(k,:) = yout(:,2); % Salvare valori tensiune pe condensator
```

```

end
plot(tout,vc(1,:),tout,vc(2,:),tout,vc(3,:)); % Valori de trasat
xlabel('Timp (s)'); grid;
ylabel('Amplitudine (volti)');
st1 = 'R = '
st2 = num2str(R); % conversie valoare R in string
st3 = ' \Omega'; % Simbol grecesc pentru Ohm
st4 = ', C = '
st5 = num2str(CAP(1));
st6 = ' F';
sl1 = strcat(st1,st2,st3,st4,st5,st6); % legenda 1
st5 = num2str(CAP(2));
sl2 = strcat(st1,st2,st3,st4,st5,st6); % legenda 2
st5 = num2str(CAP(3));
sl3 = strcat(st1,st2,st3,st4,st5,st6); % legenda 3
title('Grafic tensiune pe condensator')
legend(sl1,sl2,sl3); % Legenda grafic
axis([0 10 -5 15]); % Setare axe grafic pentru tensiune -5 la 15

```



## B. Modelarea unui circuit RLC serie

Pentru un al doilea exemplu vom modela tensiunea pe condensator pe un circuit RLC serie. Ecuația diferențială pentru tensiunea condensatorului este:

$$\frac{d^2v_c(t)}{dt^2} = -\frac{R}{L} \frac{dv_c(t)}{dt} - \frac{1}{LC} v_c(t) + \frac{1}{LC} u(t)$$

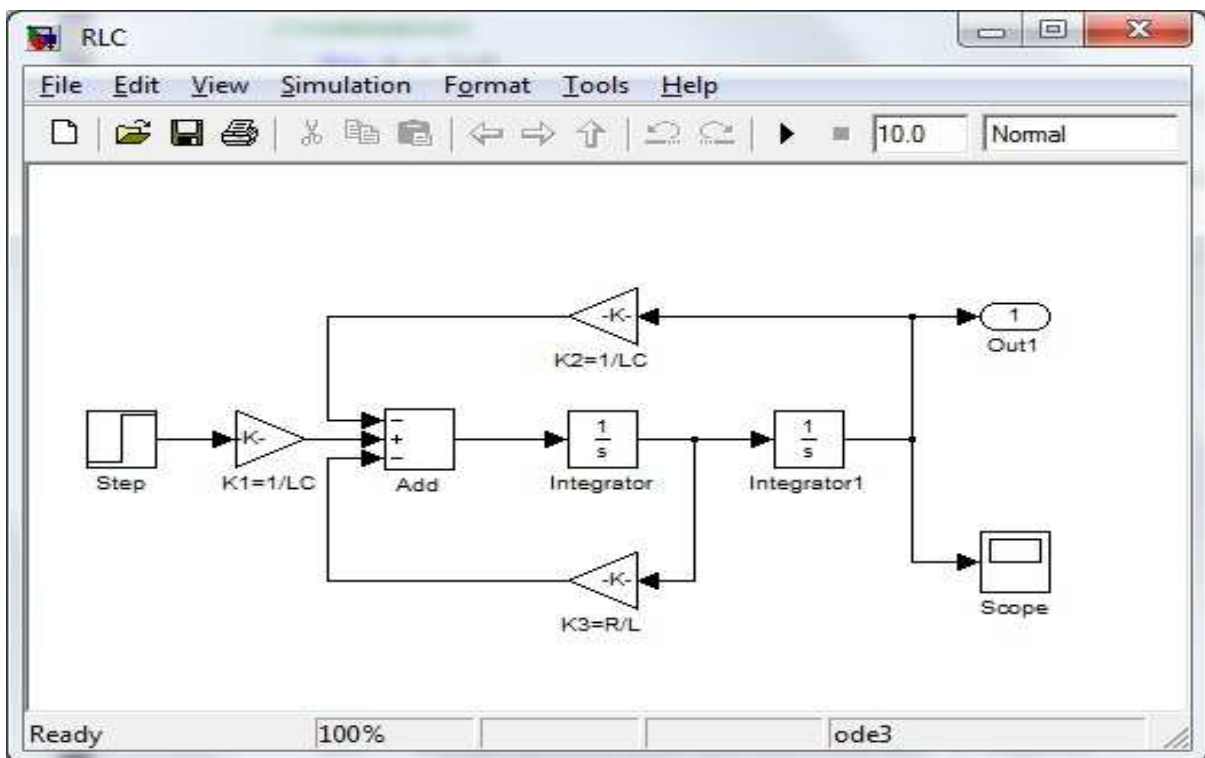
$u(t)$  = funcția unitate pas

Pentru a modela acest sistem vom lăsa

$z_1 = v_c$

$$\frac{dz_1}{dt} = z_2 = \frac{dv_c}{dt}$$

$$\frac{dz_2}{dt} = \frac{R}{L} z_2 - \frac{1}{LC} z_1 + \frac{1}{LC} u(t)$$



```
' M-File script to plot simulation data '
```

```
R = 1; % Setare valoare rezistor
```

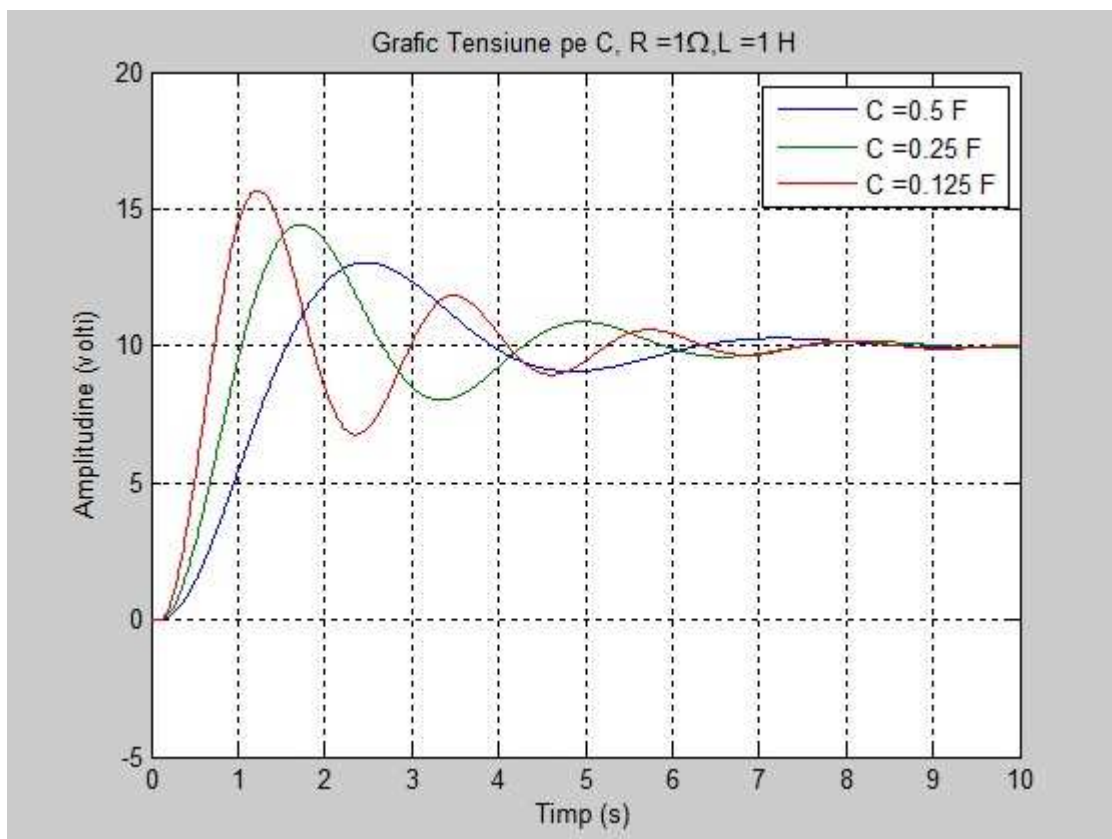
```
CAP = [1 1/2 1/4]; % Valori pentru care se va face simularea  
(Condensator)
```

```

for k = 1:3
C = CAP(k); % Citire valori simulare pentru condensator
sim('RC'); % Lansare simulare
vc(k,:) = yout(:,2); % Salvare valori tensiune pe condensator
end

plot(tout,vc(1,:),tout,vc(2,:),tout,vc(3,:)); % Valori de trasat
xlabel('Timp (s)'); grid;
ylabel('Amplitudine (volți)');
st1 = 'R = '
st2 = num2str(R); % conversie valoare R in string
st3 = ' \Omega'; % Simbol grecesc pentru Ohm
st4 = ', C = ';
st5 = num2str(CAP(1));
st6 = ' F';
s11 = strcat(st1,st2,st3,st4,st5,st6); % legenda 1
st5 = num2str(CAP(2));
s12 = strcat(st1,st2,st3,st4,st5,st6); % legenda 2
st5 = num2str(CAP(3));
s13 = strcat(st1,st2,st3,st4,st5,st6); % legenda 3
title('Grafic tensiune pe condensator')
legend(s11,s12,s13); % Legenda grafic
axis([0 10 -5 15]); % Setare axe grafic pentru tensiune -5 la 15

```



### C. Modificarea unui circuit RLC serie

Pentru un al treilea exemplu vom modifica modelul circuit RLC serie pentru a trasa tensiunea pe bobină și rezistor. Pentru aceasta vom nota:

$$v_R(t) = Ri(t) = RC \frac{dv_C(t)}{dt}$$

Din legea lui Kirchhoff pentru tensiune rezultă

$$v_L(t) = u(t) - v_R(t) - v_C(t)$$

Dar în funcție de variabilele  $z_1$  și  $z_2$  avem

$$\frac{dv_C}{dt} = \frac{dz_1}{dt} = z_2$$

Făcând aceasta substituție tensiunea scade pe rezistor

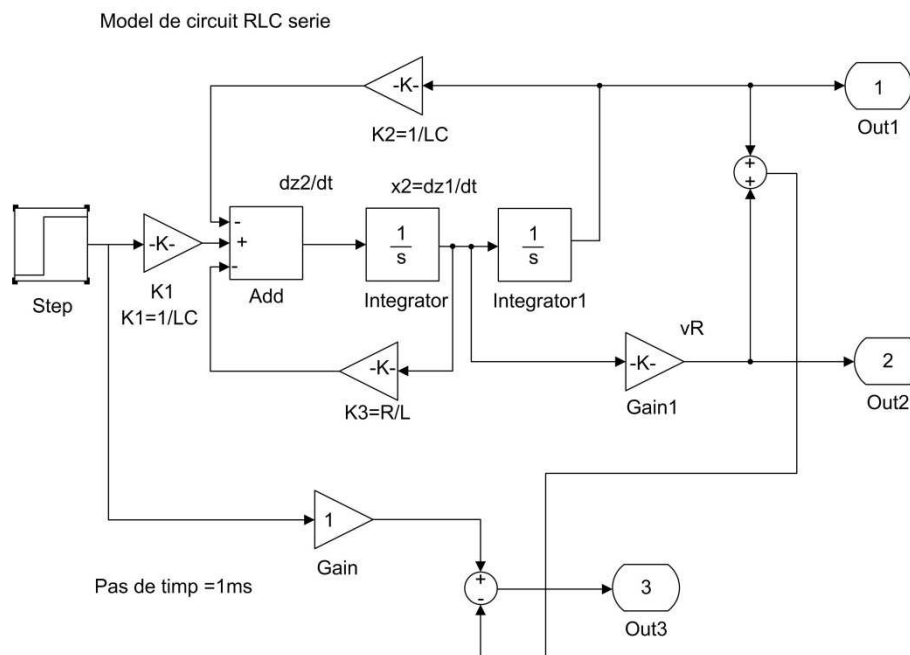
- Scăderea tensiunii pe rezistor:

$$v_R(t) = Ri(t) = RC \frac{dv_C(t)}{dt} = RCz_2$$

- Scăderea tensiunii pe bobină:

$$v_L(t) = u(t) - v_R(t) - v_C(t)$$

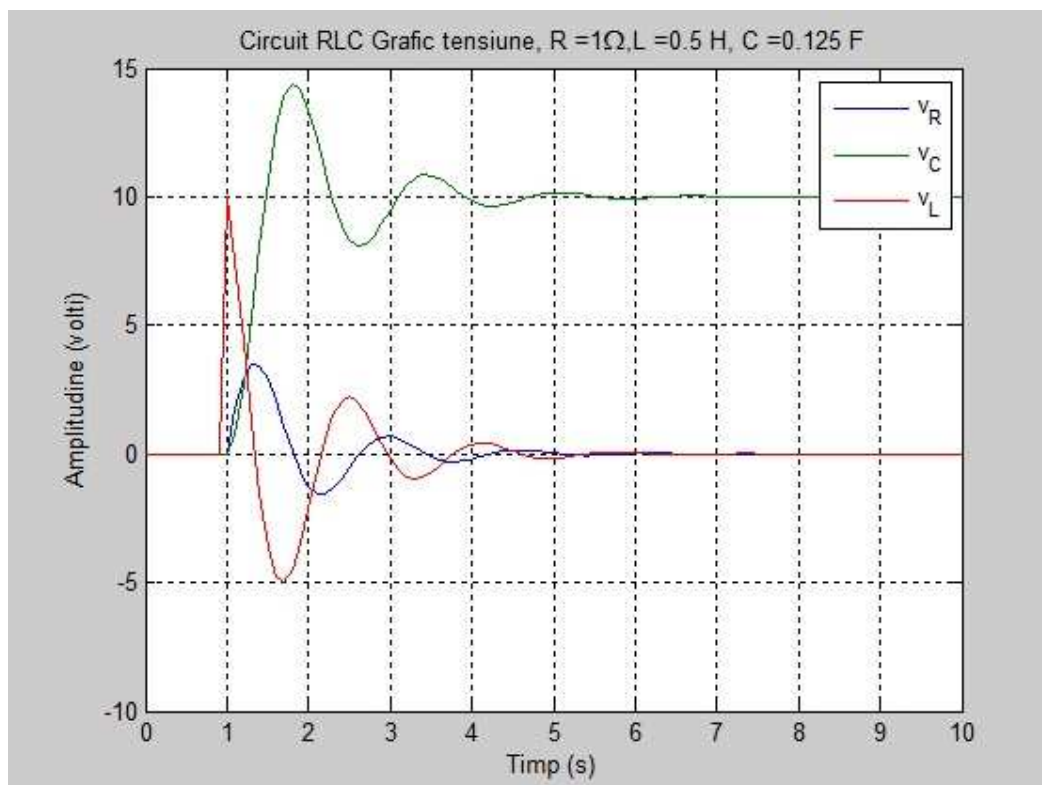
- Model Simulink:



```

' MATLAB script for RLC simulation '
R = 1; C = 1/8; L = 1/2; % Setare parametrului circuit
sim('RLC2'); % Rulare simulare
vC = yout(:,1); % Cadere tensiune pe condensator
vR = yout(:,2); % Cadere tensiune pe rezistenta
vL = yout(:,3); % Cadere tensiune pe bobina
plot(tout,vR,tout,vC,tout,vL); % Grafic caderi de tensiune
legend('v_R','v_C','v_L'); % Legenda grafic
grid; % Adaugare grila si grafic
xlabel('Time (sec)'); % Etichete pentru axe
ylabel('Amplitude (volts)');
% Creeaza titlul graficului
st1 = 'Circuit RLC Grafic Tensiune, R = ';
st2 = num2str(R); st3 = '\Omega';
st4 = ',L = '; st5 = num2str(L);
st6 = ' H'; st7 = ', C = ';
st8 = num2str(C); st9 = ' F';
stitle = strcat(st1,st2,st3,st4,st5,st6,st7,st8,st9);
title(stitle)

```



## ***Bibliografie:***

- [www.wikipedia.org](http://www.wikipedia.org)
- MATLAB Help