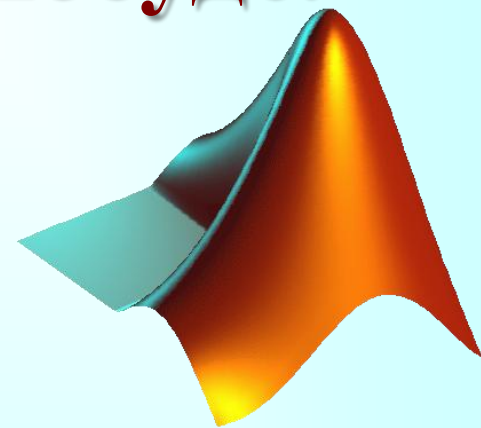
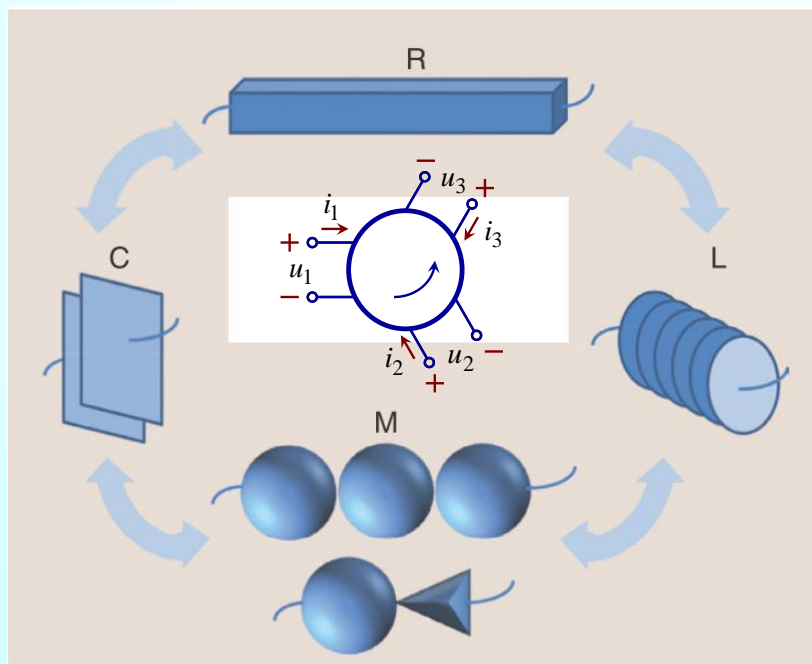


# Практикум из рачунарске анализе трофазних кола

## 3. Временски домен. Побуде.



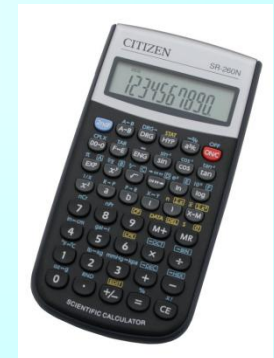
Милка Потребих Иваниш

Никола Баста



# Рачунарски (софтверски) алати

- *Mathematica*, **MuPAD**, **Maxima**, **SymPy**
- **MATLAB**, **Scilab**, **Octave**, **FreeMat**
- **LTspice**, **ngspice**, **QucsStudio**, **XCircuit**
- **Python**, **MathCAD**, **MAPLE**,  
**SpeQ Mathematics...**



**Free/Libre Open Source Software (FLOSS)**

# Одзив на побуду

Вредности елемената електричног кола са слике су познате.

(а) Одредити једначине стања у скаларном и матричном облику.

Који је ред кола?

(б) Одредити импулсни одзив (Гринову функцију) за напон  $u$  ако је

$$R_1 = R_2 = R ,$$

$$L = CR^2 .$$

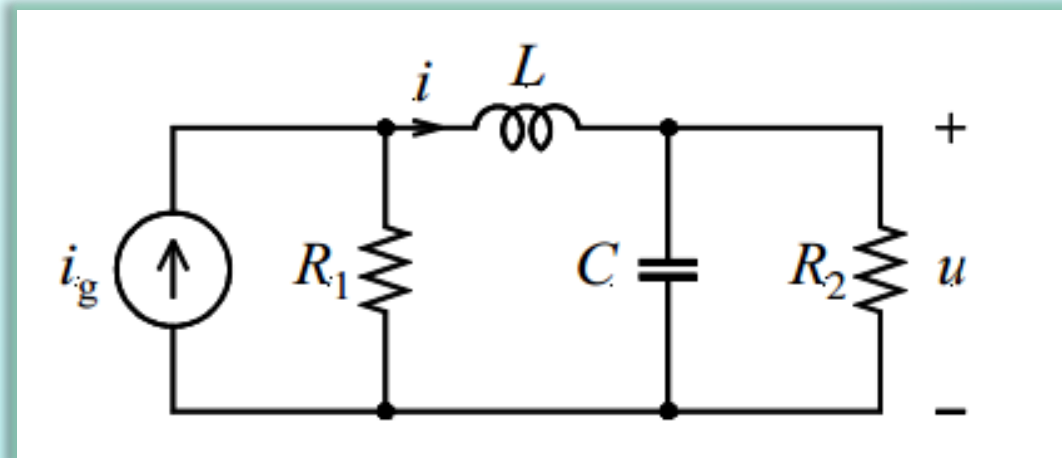
Нумерички пример:

$$R_1 = R_2 = R = 1 \text{ k}\Omega$$

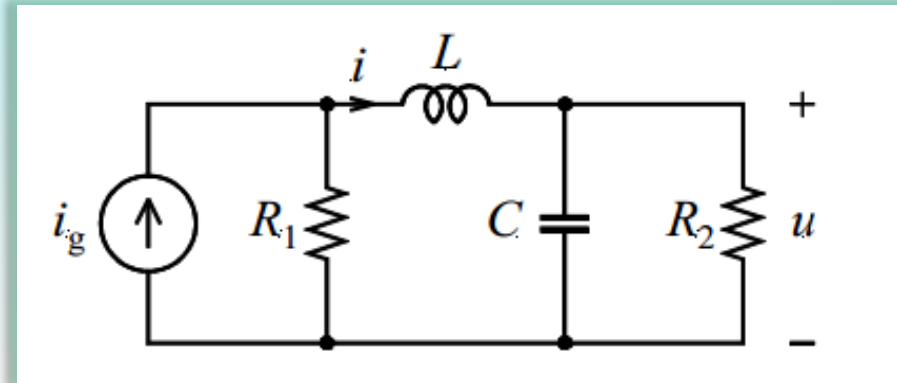
$$L = 1 \text{ H}$$

$$C = 1 \text{ }\mu\text{F}$$

$$i_g(t) = \delta(t)$$



# MATLAB: Symbolic Math Toolbox



## Brisanje promejlivih na početku

```
clear variables
```

## Definisanje simbola, odnosno nepoznatih

```
syms ig iL iR1 iR2 iC u1 u2 uL uC DiL DuC R1 R2 R C L t U
```

## Jednacine

```
jednacine = [ig == iR1 + iL, iL == iC + iR2,...  
            u1 == uL + uC, uC == u2,...  
            u1 == R1*iR1, u2 == R2*iR2, uL == L*DiL, iC == C*DuC]
```

```
jednacine = (ig = iL + iR1 iL = iC + iR2 u1 = uC + uL uC = u2 u1 = R1 iR1 u2 = R2 iR2 uL = DiL L iC = C DuC)
```

```
jednacineIzvoda = eliminate(jednacine, [iR1, iR2, iC, u1, u2, uL])
```

```
jednacineIzvoda = [uC + R1 iL - R1 ig + DiL L, R2 iL - uC - C DuC R2, R1 R2 ig - R2 uC - DiL L R2 - R1 uC - C DuC R1 R2]
```

# Елиминација сувишних променљивих и увођење функција

```
jednacinIzvoda = eliminate(jednacin, [iR1, iR2, iC, u1, u2, uL])
```

**eliminate**

```
jednacinIzvoda = [uC + R1*iL - R1*ig + DiL*L, R2*iL - uC - C*DuC, R2, R1*R2*ig - R2*uC - DiL*L, R2 - R1*uC - C*DuC, R1, R2]
```

```
izvodiStanja = solve(jednacinIzvoda, DuC, DiL)
```

```
izvodiStanja = struct with fields:
```

```
  DuC: -(uC - R2*iL)/(C*R2)
```

```
  DiL: -(uC + R1*iL - R1*ig)/L
```

```
syms iL(t) uC(t) DuC(t) DiL(t) ig(t) g_uC(t) f_uC(t)
```

```
promenljiveStanja = [iL == iL(t), DiL == diff(iL(t)), uC==uC(t), DuC==diff(uC(t))]
```

```
promenljiveStanja(t) =
```

$$\left( iL(t) = iL(t) \quad DiL(t) = \frac{\partial}{\partial t} iL(t) \quad uC(t) = uC(t) \quad DuC(t) = \frac{\partial}{\partial t} uC(t) \right)$$

```
jednacinStanjaFun = subs([diff(uC)==izvodiStanja.DuC; ...
```

```
diff(iL)==izvodiStanja.DiL;],...
```

```
lhs(promenljiveStanja), rhs(promenljiveStanja))
```

```
jednacinStanjaFun(t) =
```

$$\left( \begin{array}{l} \frac{\partial}{\partial t} uC(t) = -\frac{uC(t) - R_2 iL(t)}{C R_2} \\ \frac{\partial}{\partial t} iL(t) = -\frac{uC(t) - R_1 ig + R_1 iL(t)}{L} \end{array} \right)$$

**MATLAB: Symbolic Math Toolbox**

## MATLAB: Symbolic Math Toolbox

# Замене

### Zamene

```
zamene = [R1 == R, R2 == R, L == C*R^2, ig == heaviside(t)]
```

```
zamene(t) = (R1 = R R2 = R L = C R^2 ig(t) = heaviside(t))
```

```
jednacinStanjaFunR = subs(jednacinStanjaFun, lhs(zamene), rhs(zamene))
```

```
jednacinStanjaFunR(t) =
```

$$\left( \begin{array}{l} \frac{\partial}{\partial t} uC(t) = -\frac{uC(t) - R iL(t)}{C R} \\ \frac{\partial}{\partial t} iL(t) = -\frac{uC(t) - R \text{heaviside}(t) + R iL(t)}{C R^2} \end{array} \right)$$

# Решавање система диференцијалних једначина и цртање графика функције

## Resavanje sistema diferencijalnih jednacina

## MATLAB: Symbolic Math Toolbox

```
assume(R>0 & C>0 & L>0 );
resenjeDiff = dsolve(jednacineStanjaFunR, [uC(0)==0; iL(0)==0])
```

resenjeDiff = struct with fields:

```
il: exp(-t/(C*R))*sin(t/(C*R))*(sign(t)/4 - (exp(t/(C*R))*(sign(t)/2 + 1/2)*(cos(t/(C*R)) - sin(t/(C*R))))/2 + 1/
uC: - R*exp(-t/(C*R))*sin(t/(C*R))*(sign(t)/4 - (exp(t/(C*R))*(cos(t/(C*R)) + sin(t/(C*R)))*(sign(t)/2 + 1/2))/2
```

```
f_uC = simplify(rewrite(resenjeDiff.uC, 'Heaviside'))
```

f\_uC =

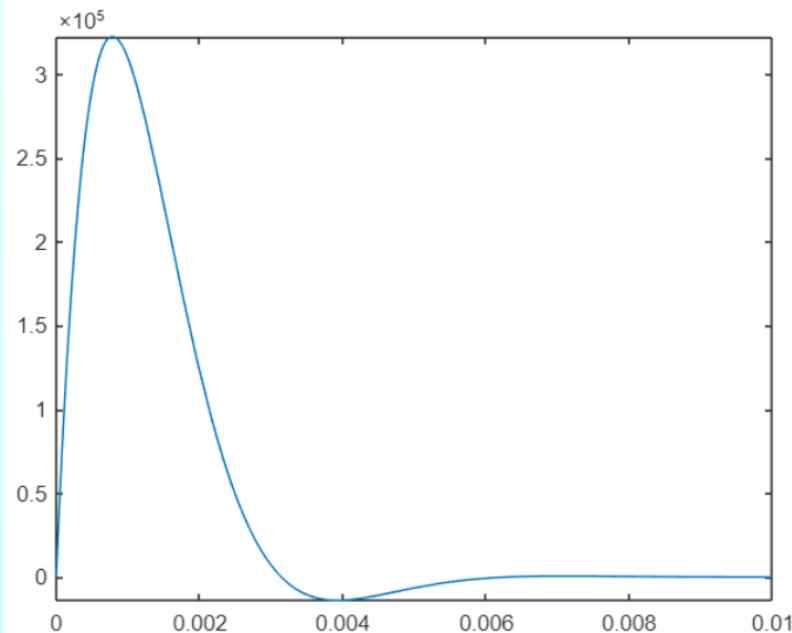
$$\frac{R \operatorname{heaviside}(t)}{2} - \frac{\sqrt{2} R e^{-\frac{t}{CR}} \sin\left(\frac{4t + \pi CR}{4CR}\right) \operatorname{heaviside}(t)}{2}$$

```
g_uC = simplify(diff(f_uC))
```

g\_uC =

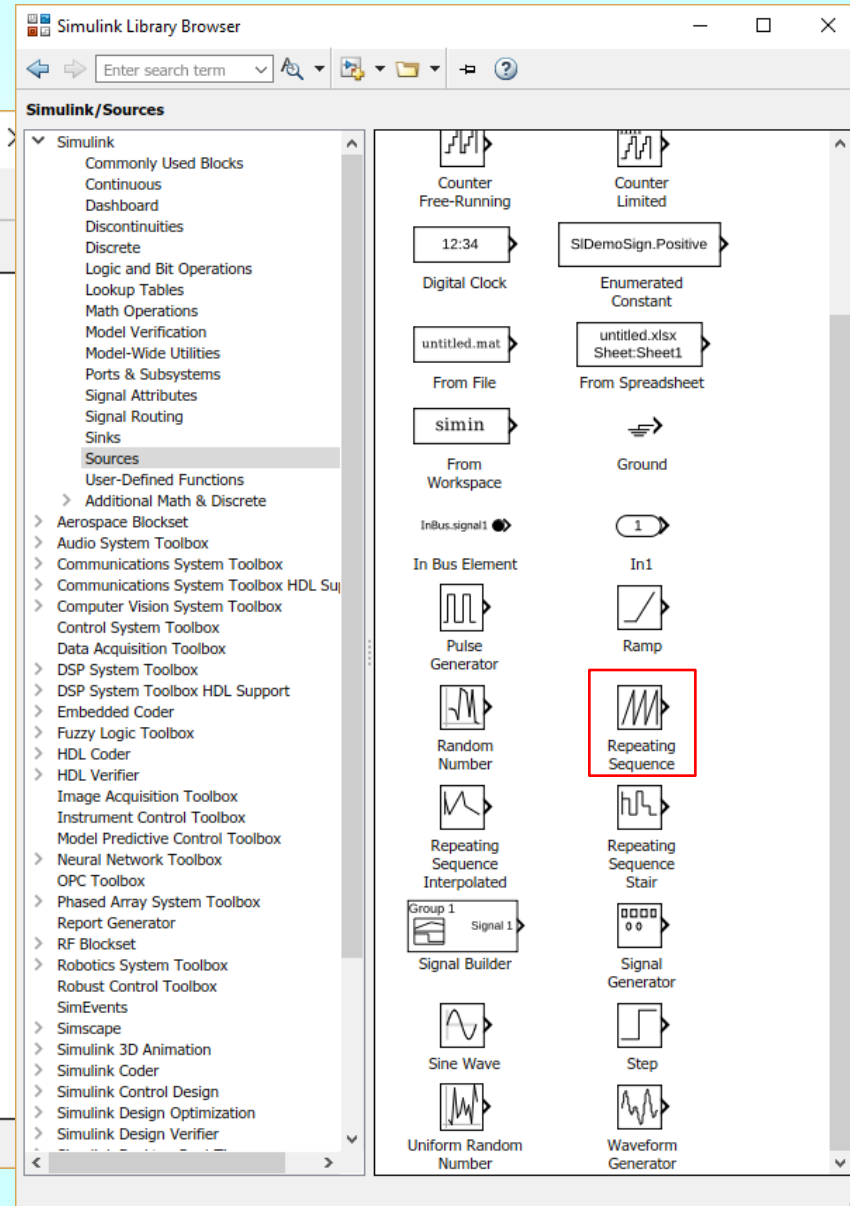
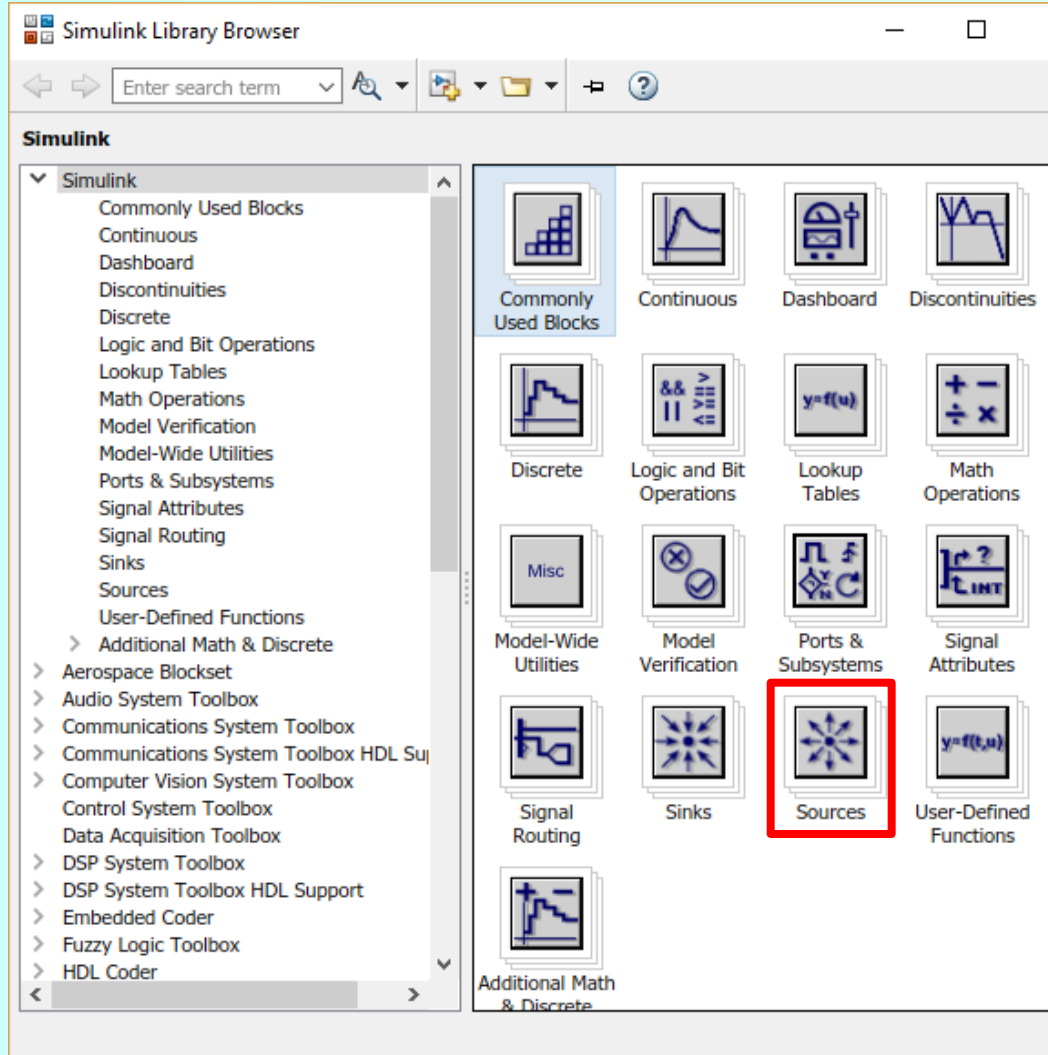
$$-\frac{e^{-\frac{t}{CR}} \cos\left(\frac{2t + \pi CR}{2CR}\right) \operatorname{heaviside}(t)}{C}$$

```
fplot(t, subs(g_uC, [R C], [1e3 1e-6]), [0 0.01])
```

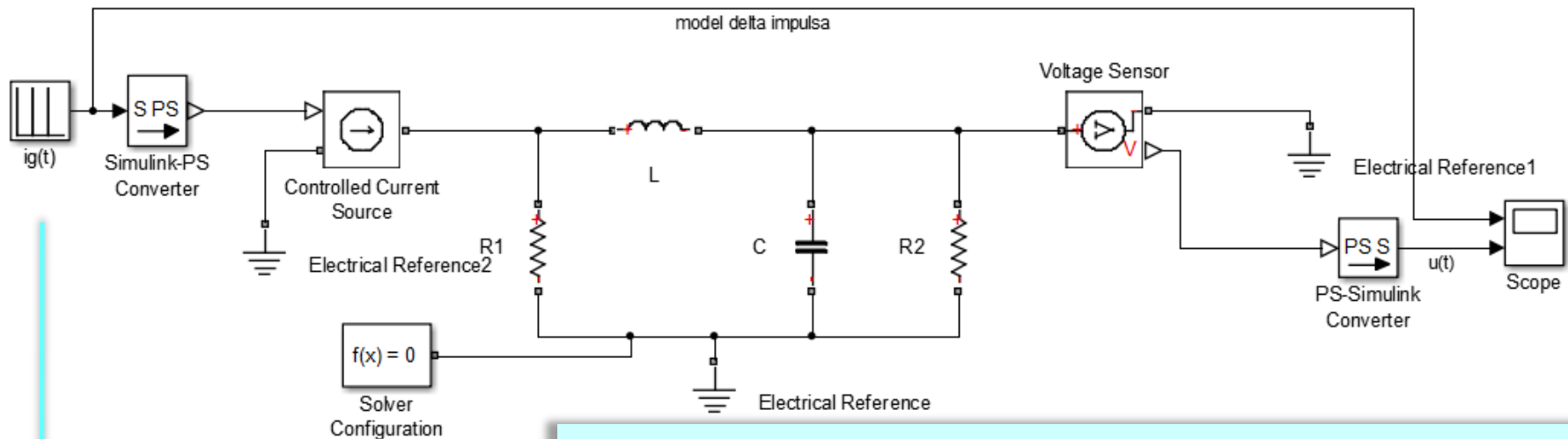


# MATLAB: Simulink

## Sources







# Импулсни одзив за напон кондензатора $u(t)$

$R_1 = R_2 = R = 1 \text{ k}\Omega$   
 $L = 1 \text{ H}$   
 $C = 1 \mu\text{F}$   
 $i_g(t) = \delta(t)$

Source Block Parameters: Repeating Sequence

Repeating table (mask) (link)

Output a repeating sequence of numbers specified in a table of time-value pairs. Values of time should be monotonically increasing.

Parameters

Time values:

[0 0.5\*1e-4 1e-4 2]

Output values:

[0 2\*1e4 0 0]

OK

Simulation Analysis Code Tools Help

Update Diagram Ctrl+D

Model Configuration Parameters Ctrl+E

Mode

Data Display

Step back (uninitialized)

Configuration Parameters: ImpulsniOdziv/Configuration (Active)

Select:

- Solver
- Data Import/Export
- Optimization
- Diagnostics
- Hardware Implementation
- Model Referencing
- Simulation Target
- Code Generation
- HDL Code Generation
- SimMechanics 1G
- SimMechanics 2G

Simulation time

Start time: 0.0 Stop time: 1

Solver options

Type: Variable-step Solver: ode45 (Dormand-Prince)

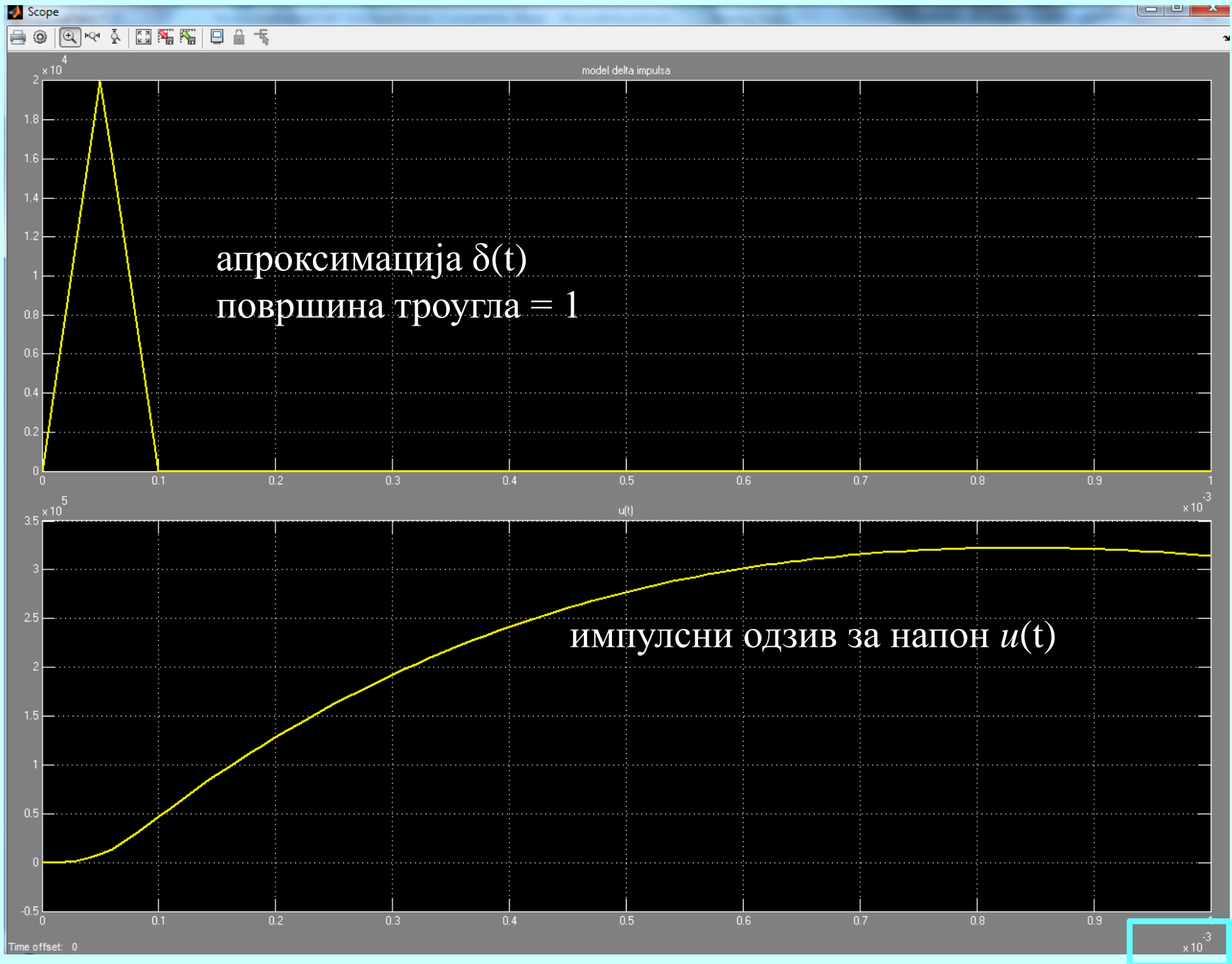
Max step size: 1e-5 Relative tolerance: 1e-3

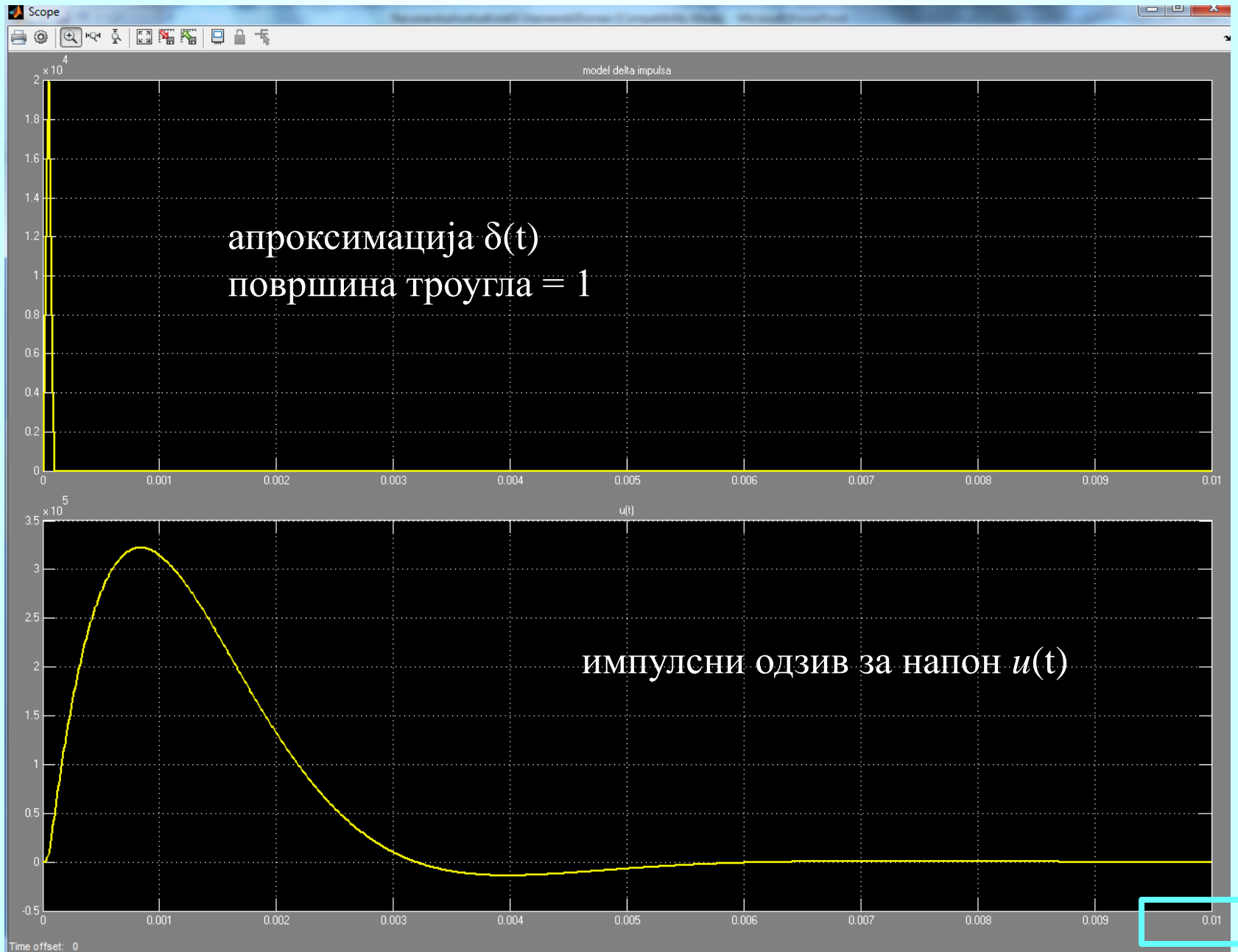
Min step size: auto Absolute tolerance: auto

Initial step size: auto Shape preservation: Disable All

Number of consecutive min steps: 1

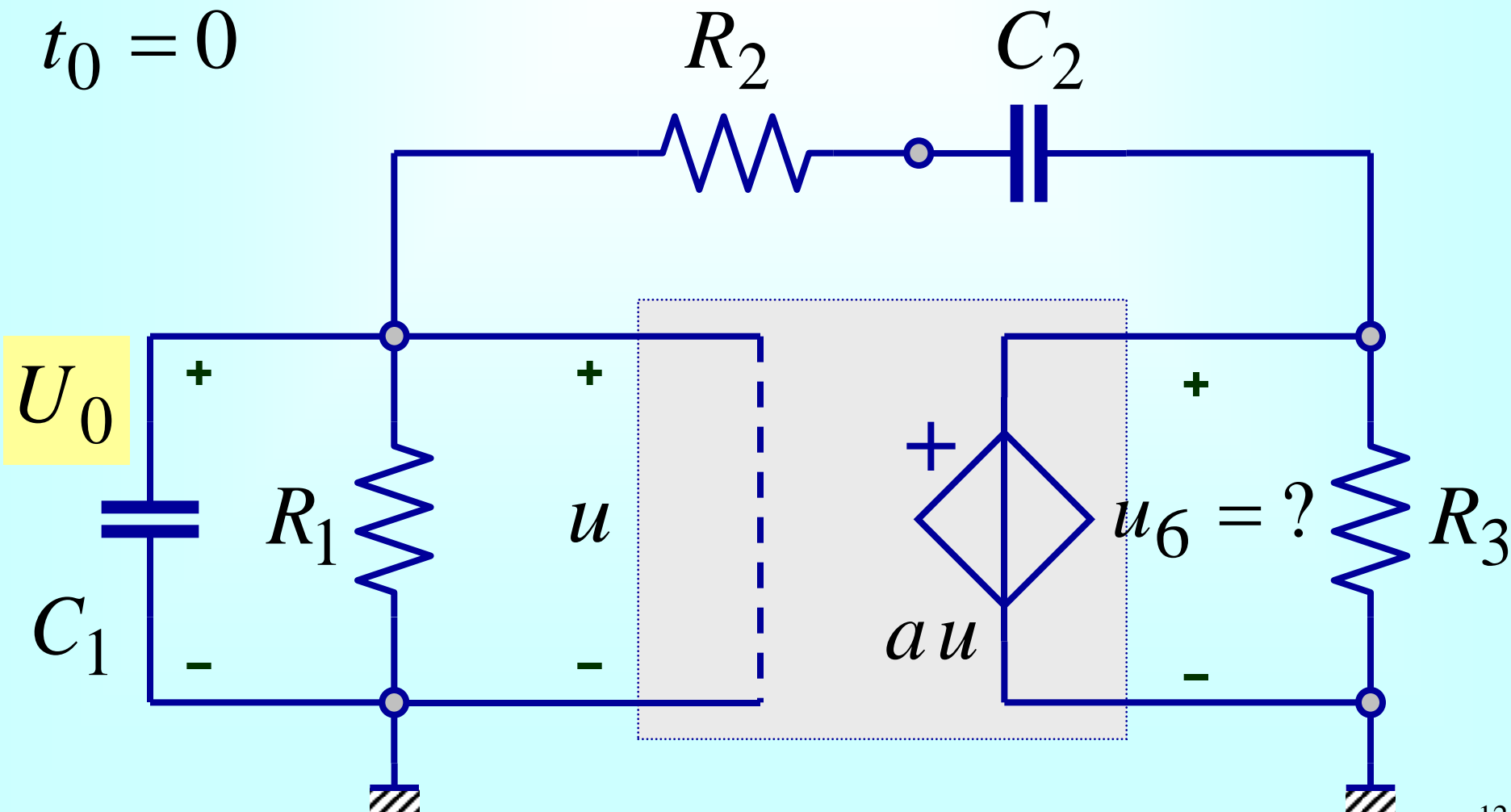
MATLAB: Simscape  
 Foundation Library  
 Utilities  
 Simulink



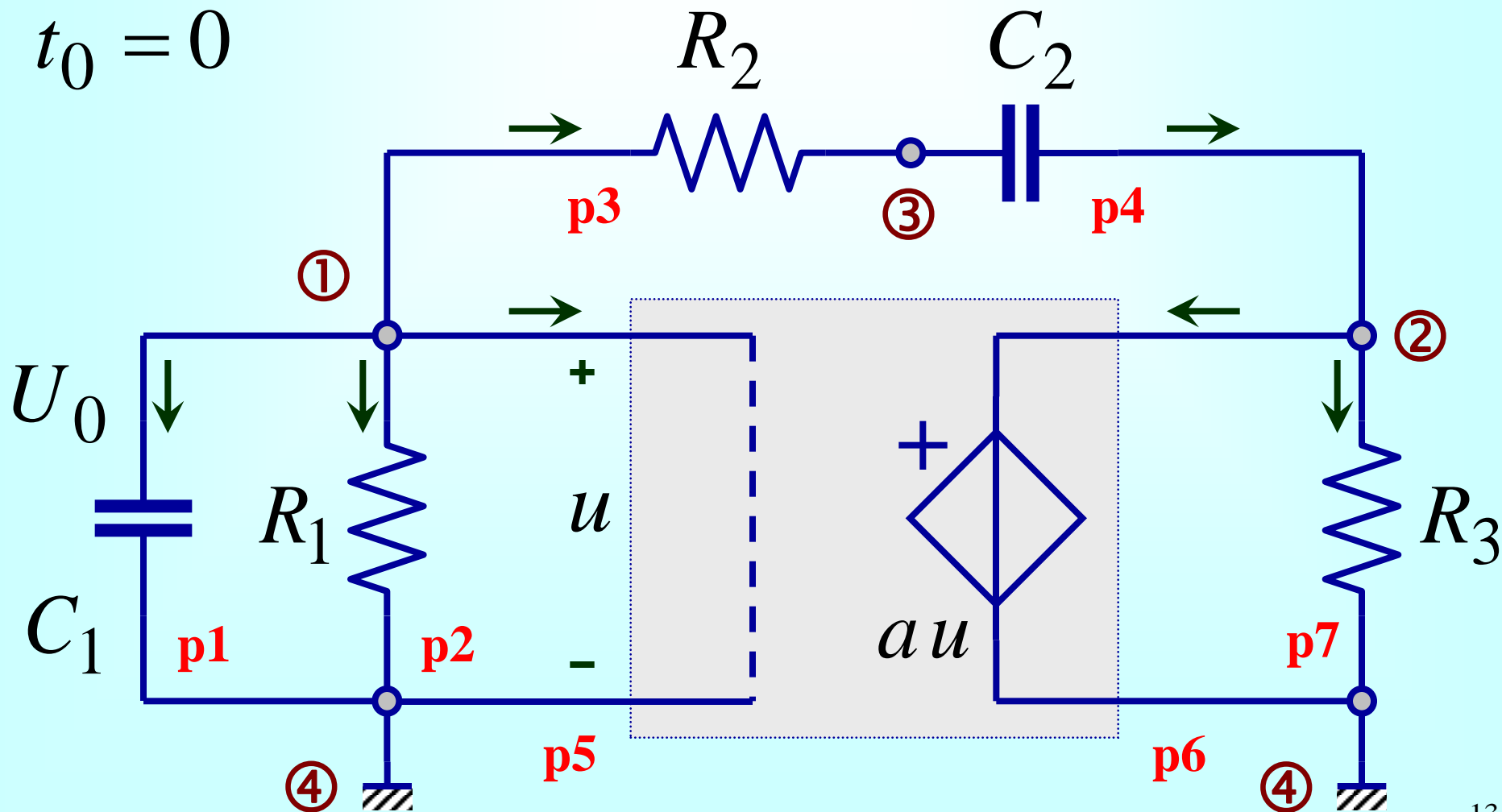


# Одзив на сакупљену енергију

$t_0 = 0$



# Ознаке чворова и приступа



# MATLAB: Symbolic Math Toolbox

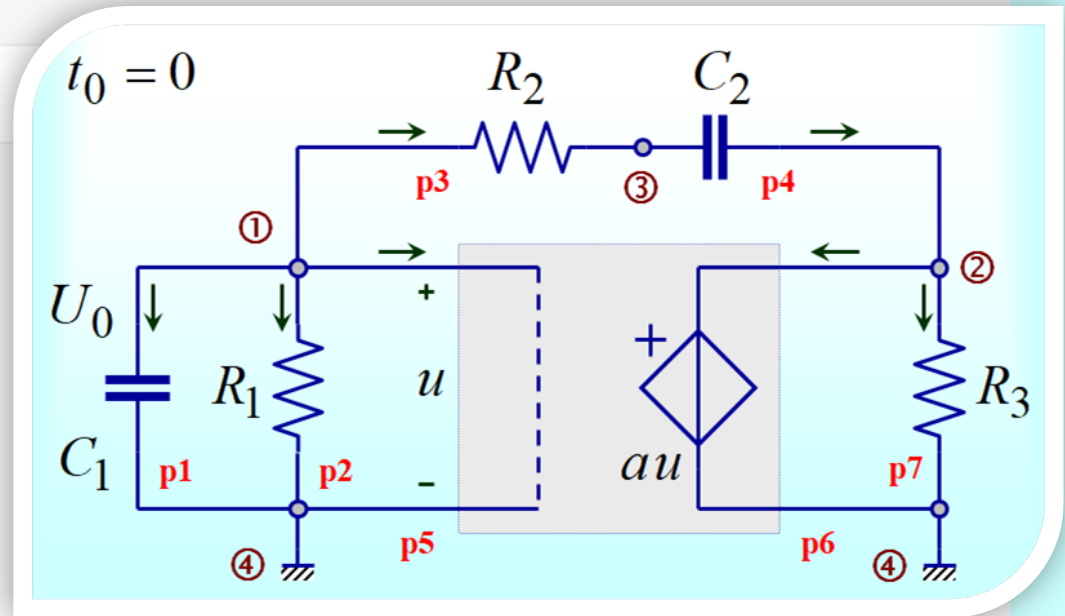
## Систем једначина кола

### Definisanje simbola, odnosno nepoznatih

```
syms i1 i2 i3 i4 i5 i6 i7 u1 u2 u3 u4 u5 u6 u7
syms Du1 Du4 D R1 R2 R3 R C1 C2 C a U0
```

### Jednacine

```
jednacine = [i1 + i2 + i3 + i5 == 0, ...
-i4 + i6 + i7 == 0, ...
-i3 + i4 == 0, ...
u1 - u2 == 0, ...
u2 - u5 == 0, ...
u5 - u6 - u4 - u3 == 0, ...
u6 - u7 == 0, ...
i1 == C1*Du1, ...
u2 == R1*i2, ...
u3 == R2*i3, ...
i4 == C2*Du4, ...
i5 == 0, ...
u6 == a*u5, ...
u7 == R3*i7]
```



```
jednacine = (i1 + i2 + i3 + i5 = 0 i6 - i4 + i7 = 0 i4 - i3 = 0 u1 - u2 = 0 u2 - u5 = 0 u5 - u4 - u3 - u6 = 0 u6 - u7 =
```

# MATLAB: Symbolic Math Toolbox

## Једначине стања

```
jednacineIzvoda = eliminate(jednacine, [i1, i2, i3, i4, i5, i6, i7, u2, u3, u5, u6, u7])
```

```
jednacineIzvoda = [u1 + C1 Du1 R1 + C2 Du4 R1, u4 - u1 + a u1 + C2 Du4 R2]
```

```
jednacineStanja = solve(jednacineIzvoda, [Du1, Du4])
```

```
jednacineStanja = struct with fields:
```

```
Du1: -(R1*u1 + R2*u1 - R1*u4 - R1*a*u1)/(C1*R1*R2)
```

```
Du4: -(u4 - u1 + a*u1)/(C2*R2)
```

# Решавање система диференцијалних једначина

## Resavanje sistema diferencijalnih jednacina

## MATLAB: Symbolic Math Toolbox

```
syms u1(t) u4(t) Du1(t) Du4(t)
promenljiveStanja = [u1 == u1(t) u4 == u4(t)]
```

```
promenljiveStanja(t) = (u1(t) = u1(t) u4(t) = u4(t))
```

```
zamene = [R1 == R, R2 == R, R3 == R, C1 == C, C2 == C, a == 3]
```

```
zamene = (R1 = R R2 = R R3 = R C1 = C C2 = C a = 3)
```

```
jednacineStanjaFun = subs([diff(u1) == jednacineStanja.Du1;...
                           diff(u4) == jednacineStanja.Du4],...
                           [lhs(promenljiveStanja) lhs(zamene)], [rhs(promenljiveStanja) rhs(zamene)])
```

$$\text{jednacineStanjaFun}(t) = \begin{pmatrix} \frac{\partial}{\partial t} u_1(t) = \frac{R u_1(t) + R u_4(t)}{C R^2} \\ \frac{\partial}{\partial t} u_4(t) = -\frac{2 u_1(t) + u_4(t)}{C R} \end{pmatrix}$$

```
resenjeDiff = dsolve(jednacineStanjaFun, [u1(0)==10, u4(0)==0])
```

**почетне вредности**

```
resenjeDiff = struct with fields:
```

```
u4: - exp(-(t*1i)/(C*R))*10i + exp((t*1i)/(C*R))*10i
u1: exp(-(t*1i)/(C*R))*(5 + 5i) + exp((t*1i)/(C*R))*(5 - 5i)
```



## MATLAB: Symbolic Math Toolbox

# Одзив – напон $u_6(t)$

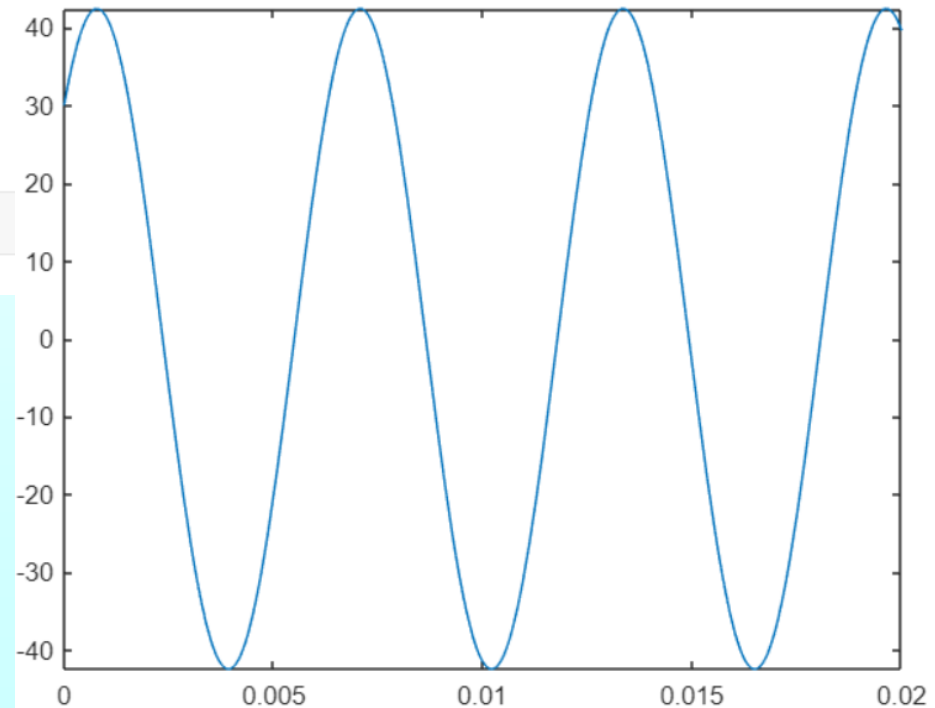
```
u6(t) = rewrite(simplify(3*resenjeDiff.u1), 'cos')
```

rewrite

$$u_6(t) = 30 \cos\left(\frac{t}{CR}\right) + 30 \sin\left(\frac{t}{CR}\right)$$

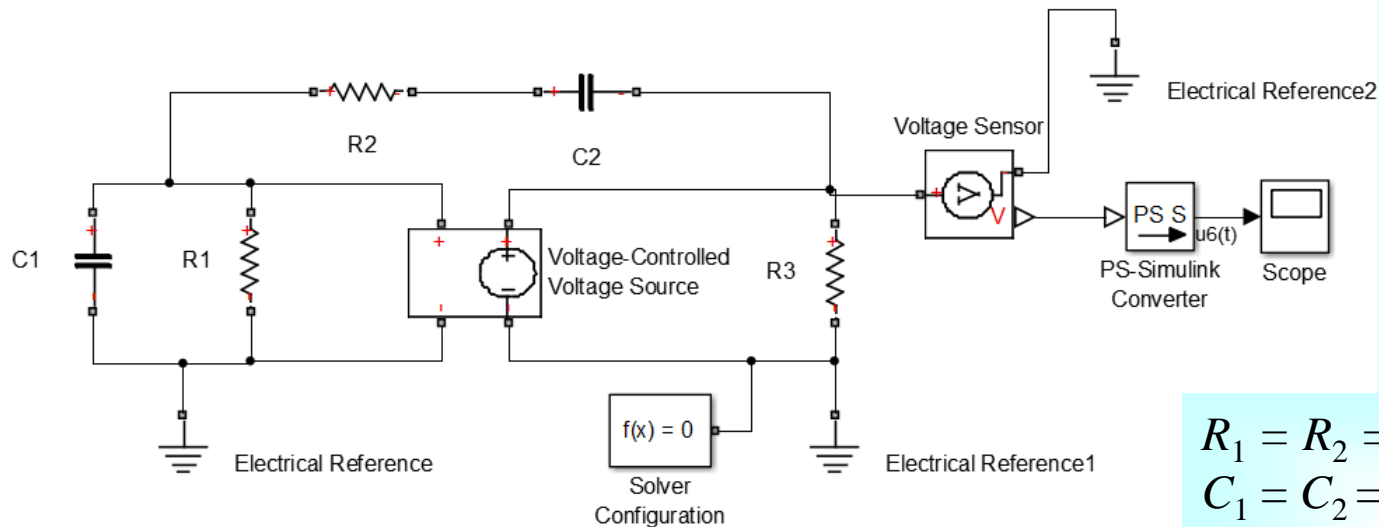
**Crtanje dijagrama**

```
fplot(t, subs(u6(t),[R, C], [1e3, 1e-6]), [0 20e-3])
```



$t > t_0$

# MATLAB: Simscape Foundation Library, Utilities, Simulink

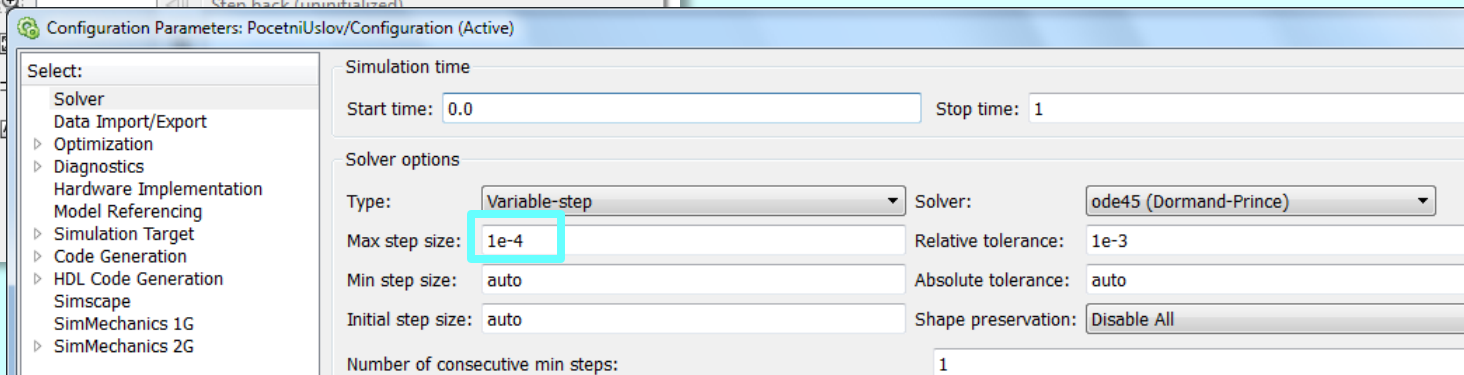
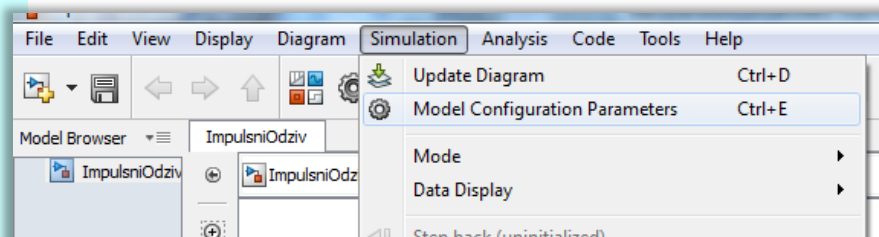


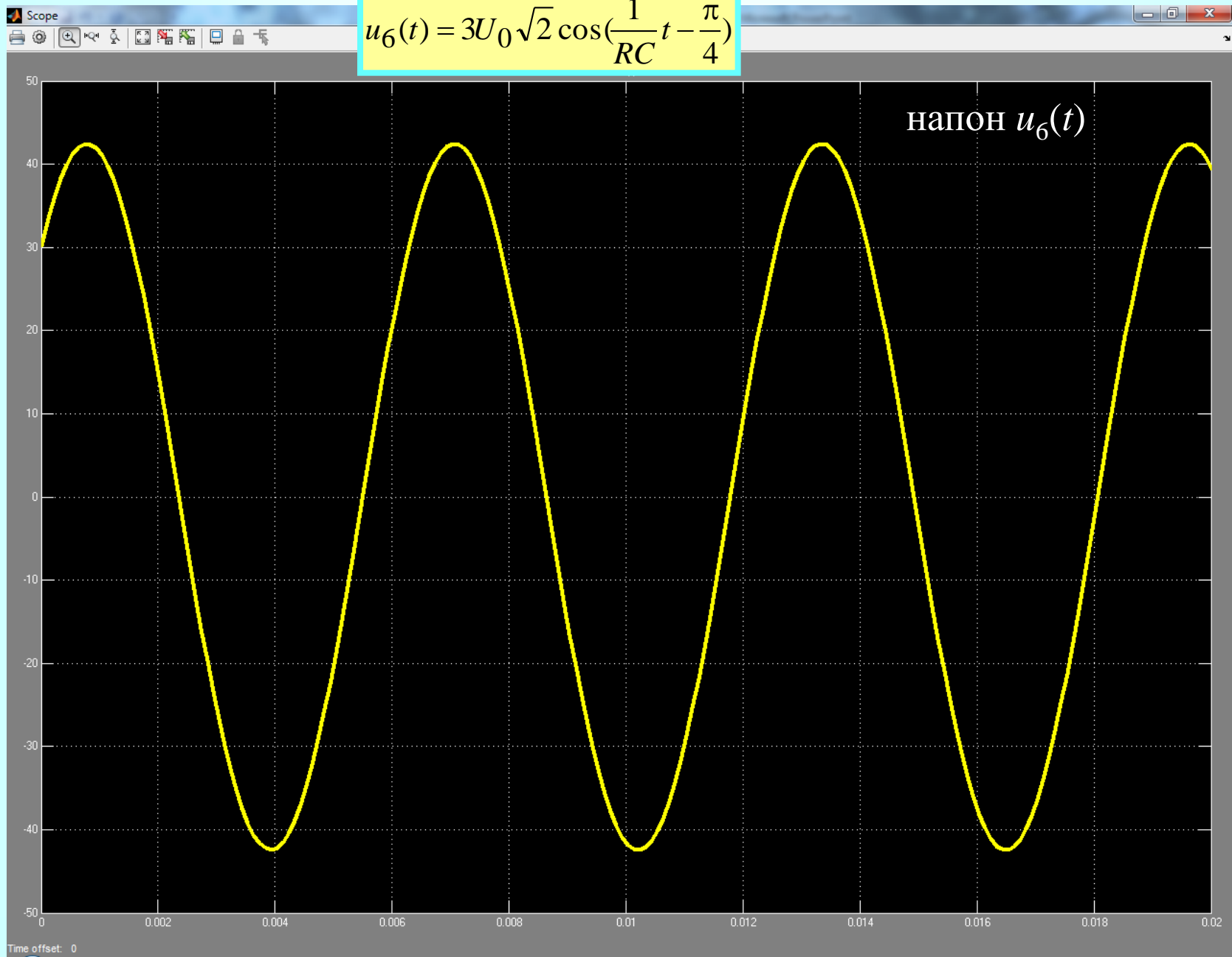
$$R_1 = R_2 = R = 1 \text{ k}\Omega$$

$$C_1 = C_2 = 1 \text{ }\mu\text{F}$$

$$U_0 = 10 \text{ V}$$

$$a = 3$$





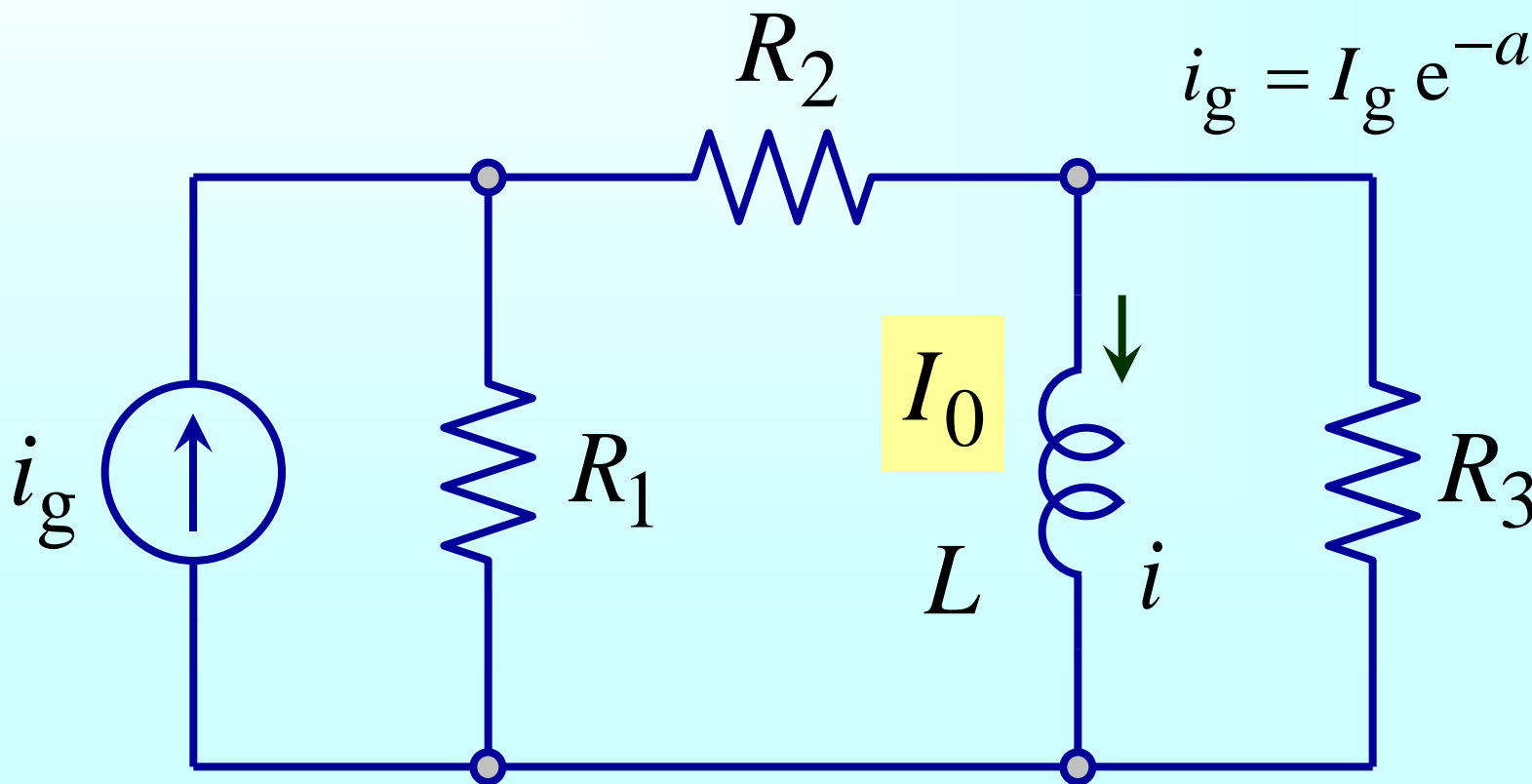
# Одзив на побуду и сакупљену енергију

$t_0 = 0$

$$R_1 = R_2 = R$$

$$R_3 = 2R$$

$$i_g = I_g e^{-at} h(t)$$



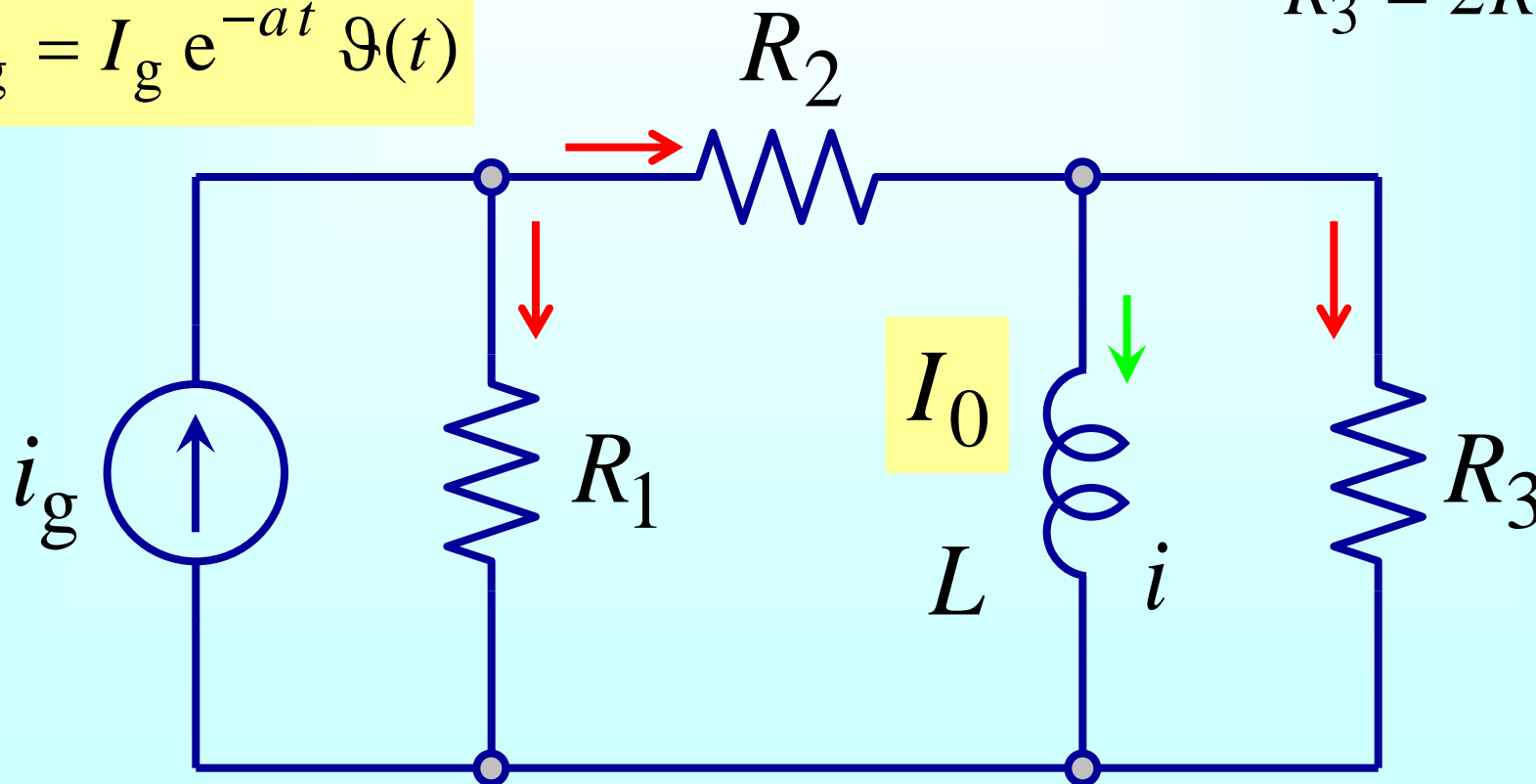
# Коло са побудом и почет. енер.

$$t_0 = 0$$

$$i_g = I_g e^{-at} \mathfrak{D}(t)$$

$$R_1 = R_2 = R$$

$$R_3 = 2R$$



# Систем једначина кола

## Definisanje simbola, odnosno nepoznatih

```
syms iR1 iR2 iR3 iL DiL uR1 uR2 uR3 uL ig Ig I0 a L R R1 R2 R3
```

## Jednacine

```
jednacine = [iR1 + iR2 - ig == 0, ...
             -iR2 + iL + iR3 == 0, ...
             uR1 - uL - uR2 == 0, ...
             uL - uR3 == 0, ...
             uL == L*DiL, ...
             uR1 == R1*iR1, ...
             uR2 == R2*iR2, ...
             uR3 == R3*iR3]
```

```
jednacine = (iR1 + iR2 - ig = 0 iL - iR2 + iR3 = 0 uR1 - uL - uR2 = 0 uL - uR3 = 0 uL = DiL L uR1 = R1 iR1 uR2 = R2 iR2 uR3 = R3 iR3)
```

```
jednacineIzvoda = eliminate(jednacine, [iR1, iR2, iR3, uR1, uR2, uR3, uL])
```

```
jednacineIzvoda = [R1 R3 ig - DiL L R2 - DiL L R3 - R1 R3 iL - R2 R3 iL - DiL L R1]
```

```
jednacineStanja = solve(jednacineIzvoda, DiL)
```

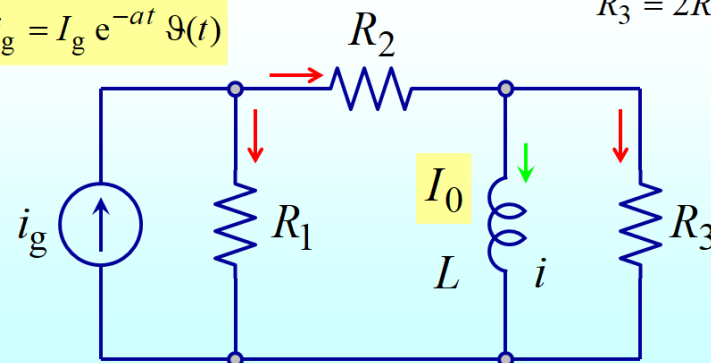
```
jednacineStanja =
- (R1 R3 iL + R2 R3 iL - R1 R3 ig)
  (L R1 + L R2 + L R3)
```

$t_0 = 0$

$$i_g = I_g e^{-at} \vartheta(t)$$

$$R_1 = R_2 = R$$

$$R_3 = 2R$$



MATLAB: Symbolic Math Toolbox

# Једначина стања

```
syms iL(t)
promenljiveStanja = [iL == iL(t)]

promenljiveStanja(t) = iL(t) = iL(t)

zamene = [R1 == R, R2 == R, R3 == 2*R, ig == Ig*exp(-a*t)*heaviside(t)]

zamene = (R1 = R R2 = R R3 = 2 R ig = Ig e-at heaviside(t))

jednacinestanjaFun = subs(diff(iL(t))==jednacinestanja, lhs([promenljiveStanja zamene]), rhs([promenljiveStanja zamene]))

jednacinestanjaFun =

$$\frac{\partial}{\partial t} iL(t) = -\frac{4 R^2 iL(t) - 2 I_g R^2 e^{-at} \text{heaviside}(t)}{4 L R}$$


resenjeDiff = dsolve(jednacinestanjaFun, iL(0)==I0, 'IgnoreAnalyticConstraints', false)

resenjeDiff =

$$I_0 e^{-\frac{Rt}{L}} - e^{-\frac{Rt}{L}} \left( \frac{I_g R \text{heaviside}(t)}{2 (R - L a)} - \frac{I_g R e^{\frac{Rt}{L}} e^{-at} \text{heaviside}(t)}{2 (R - L a)} \right)$$

```

омогућава ефикасан испис решења  
помоћу Хевисајдових функција

Одзив – струја  $i(t)$ 

```
numzamene = [I0 ==-10, R==1e3, L==1e-3, Ig==10 a == 100]
```

```
numzamene =
```

$$\left( I_0 = -10 \quad R = 1000 \quad L = \frac{1}{1000} \quad I_g = 10 \quad a = 100 \right)$$

```
f_iL(t) = subs(resenjeDiff, lhs(numzamene), rhs(numzamene))
```

```
f_iL(t) =
```

$$-10 e^{-1000000t} - e^{-1000000t} \left( \frac{50000 \operatorname{heaviside}(t)}{9999} - \frac{50000 e^{999900t}}{9999} \operatorname{heaviside}(t) \right)$$

велики број

```
f_iL_exp(t) = expand(f_iL(t))
```

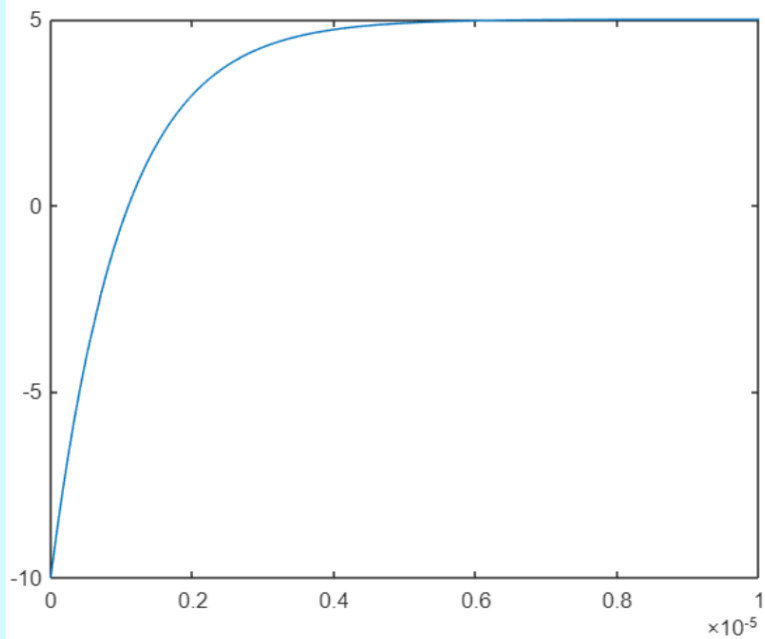
expand

```
f_iL_exp(t) =
```

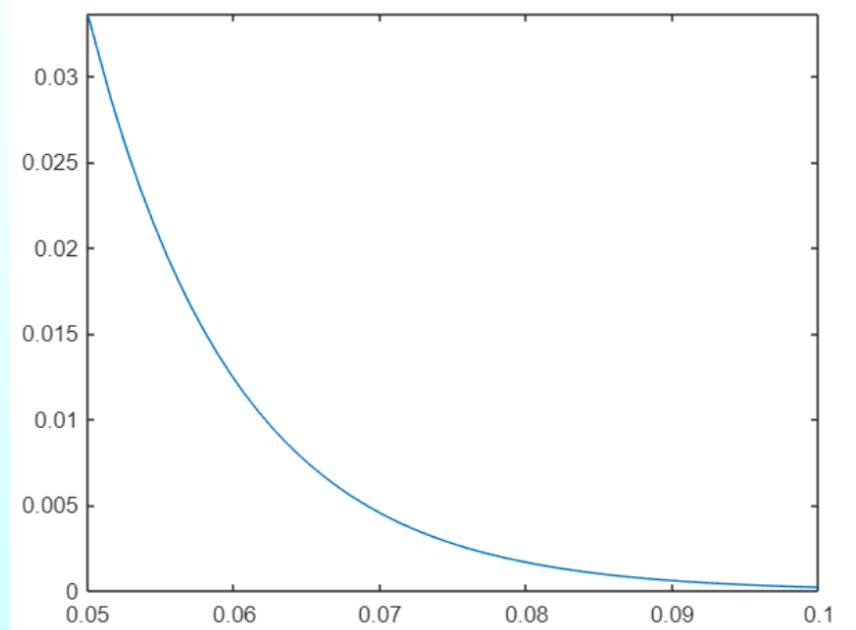
$$\frac{50000 e^{-100t} \operatorname{heaviside}(t)}{9999} - 10 e^{-1000000t} - \frac{50000 e^{-1000000t} \operatorname{heaviside}(t)}{9999}$$



```
fplot(t, f_iL_exp(t), [0 10e-6])
```

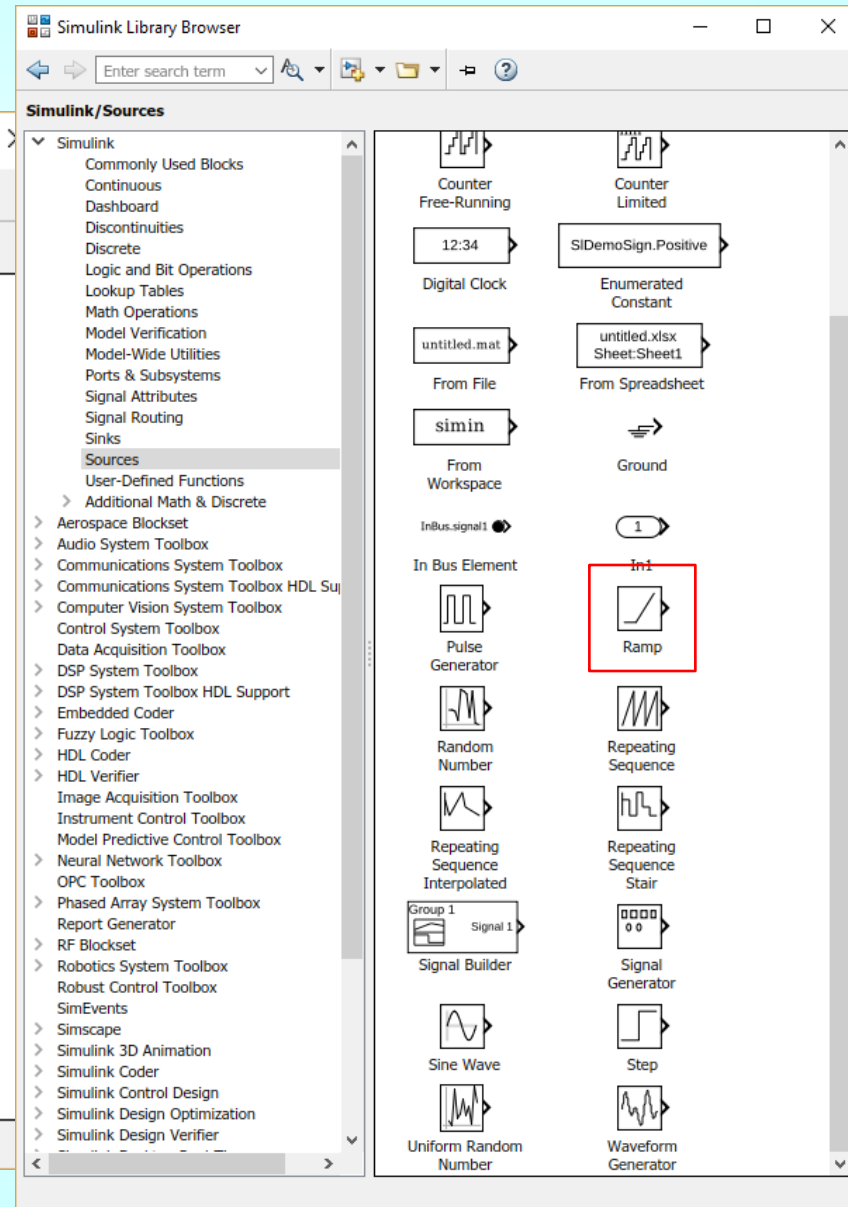
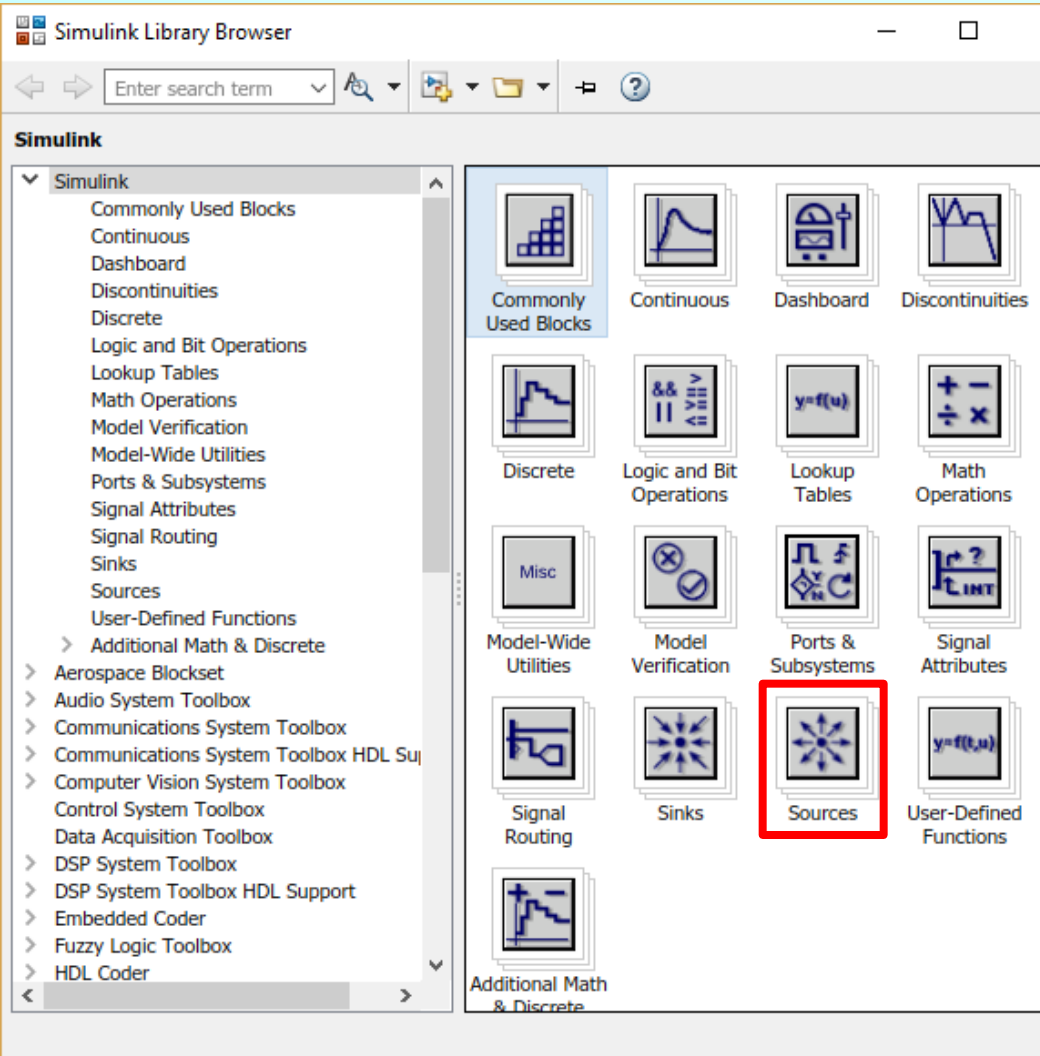


```
fplot(t, f_iL_exp(t), [0.05 0.1])
```



# MATLAB: Simulink

## Sources



# MATLAB: Simulink

## Math Operations

The image shows the Simulink Library Browser interface. On the left, a tree view lists various block categories, with 'Math Operations' highlighted. The main area displays a grid of block icons. A red box highlights the 'Math Operations' category icon, which contains symbols for addition, subtraction, multiplication, and division. To the right, a larger view shows a grid of specific math blocks, with three blocks highlighted by red boxes: 'Gain' (a triangle with '1' inside), 'Math Function' (a box with  $e^u$ ), and 'Complex to Magnitude-Angle' (a box with  $|u|$  and  $\angle u$ ).

**Simulink Library Browser**

Enter search term

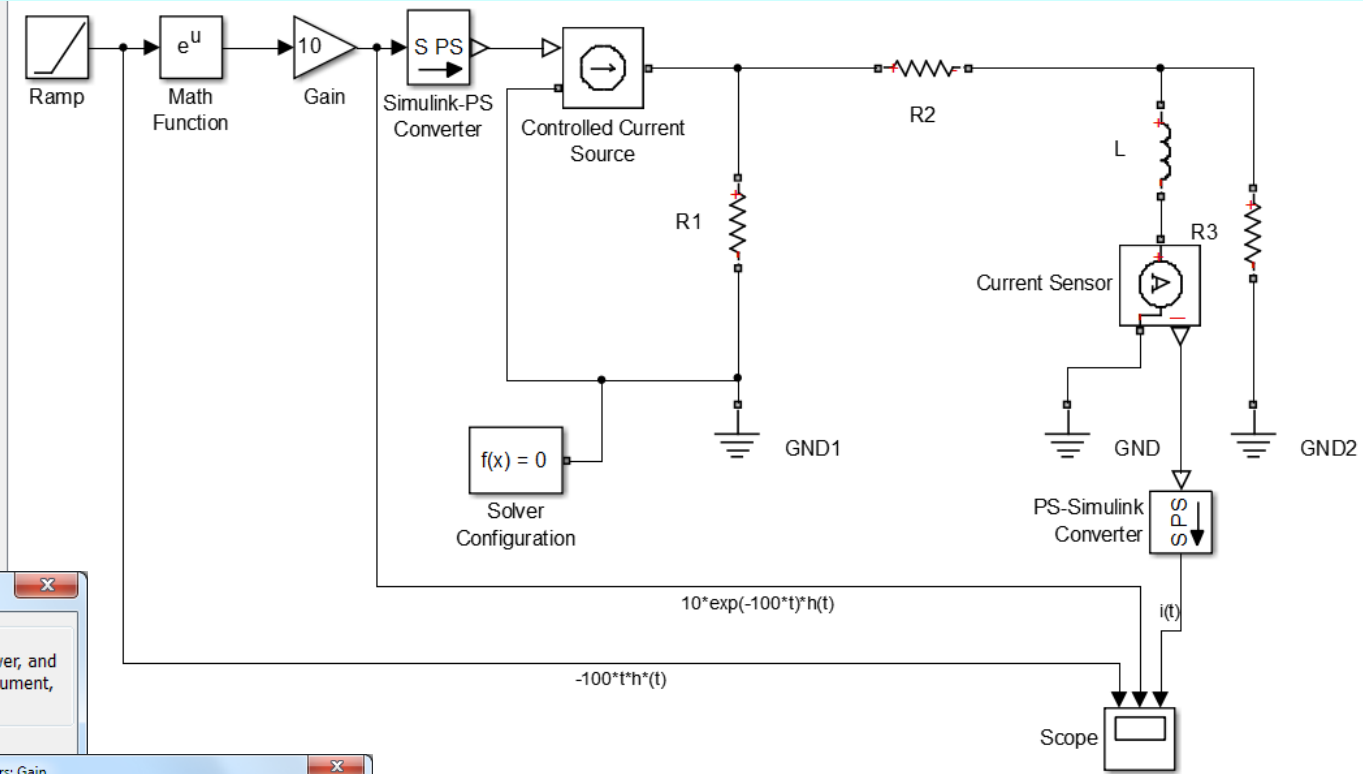
**Simulink**

- Simulink
  - Commonly Used Blocks
  - Continuous
  - Dashboard
  - Discontinuities
  - Discrete
  - Logic and Bit Operations
  - Lookup Tables
  - Math Operations
  - Model Verification
  - Model-Wide Utilities
  - Ports & Subsystems
  - Signal Attributes
  - Signal Routing
  - Sinks
  - Sources
  - User-Defined Functions
  - Additional Math & Discrete
  - Aerospace Blockset
  - Audio System Toolbox
  - Communications System Toolbox
  - Communications System Toolbox HDL Support
  - Computer Vision System Toolbox
  - Control System Toolbox
  - Data Acquisition Toolbox
  - DSP System Toolbox
  - DSP System Toolbox HDL Support
  - Embedded Coder
  - Fuzzy Logic Toolbox
  - HDL Coder

**Math Operations**

- Abs
- Add
- Algebraic Constraint
- Assignment
- Bias
- Complex to Magnitude-Angle
- Complex to Real-Imag
- Divide
- Dot Product
- Gain
- Magnitude-Angle to Complex
- Math Function
- Matrix Concatenate
- MinMax Running Resetable
- MinMax
- MinMax Running Resetable
- Permute Dimensions
- Polynomial

$t > t_0, \quad t_0 = 0$



$R_1 = R_2 = R = 1\text{ k}\Omega, R_3 = 2R, L = 1\text{ mH}$   
 $i_g(t) = I_g e^{-at} h(t), I_g = 10\text{ A}, a = 100$   
 $i(0^-) = I_0 = -10\text{ A}$

Source Block Parameters: Ramp

Ramp (mask) (link)

Output a ramp signal starting at the specified time.

Parameters

Slope:

Start time:

Initial output:

Interpret vector parameters as 1-D

Function Block Parameters: Math Function

Math

Mathematical functions including logarithmic, exponential, power, and modulus functions. When the function has more than one argument, the first argument corresponds to the top (or left) input port.

Main Signal Attributes

Function:

Output sig:

Sample time:

Function Block Parameters: Gain

Gain

Element-wise gain ( $y = K.*u$ ) or matrix gain ( $y = K*u$  or  $y = u*K$ ).

Main Signal Attributes Parameter Attributes

Gain:

Multiplication:

Sample time (-1 for inherited):

Configuration Parameters: PocetniUсловPobuda/Configuration (Active)

Select:

- Solver
- Data Import/Export
- Optimization
- Diagnostics
- Hardware Implementation
- Model Referencing
- Simulation Target
- Code Generation
- HDL Code Generation
- Simulink
- SimMechanics 1G

Simulation time

Start time:  Stop time:

Solver options

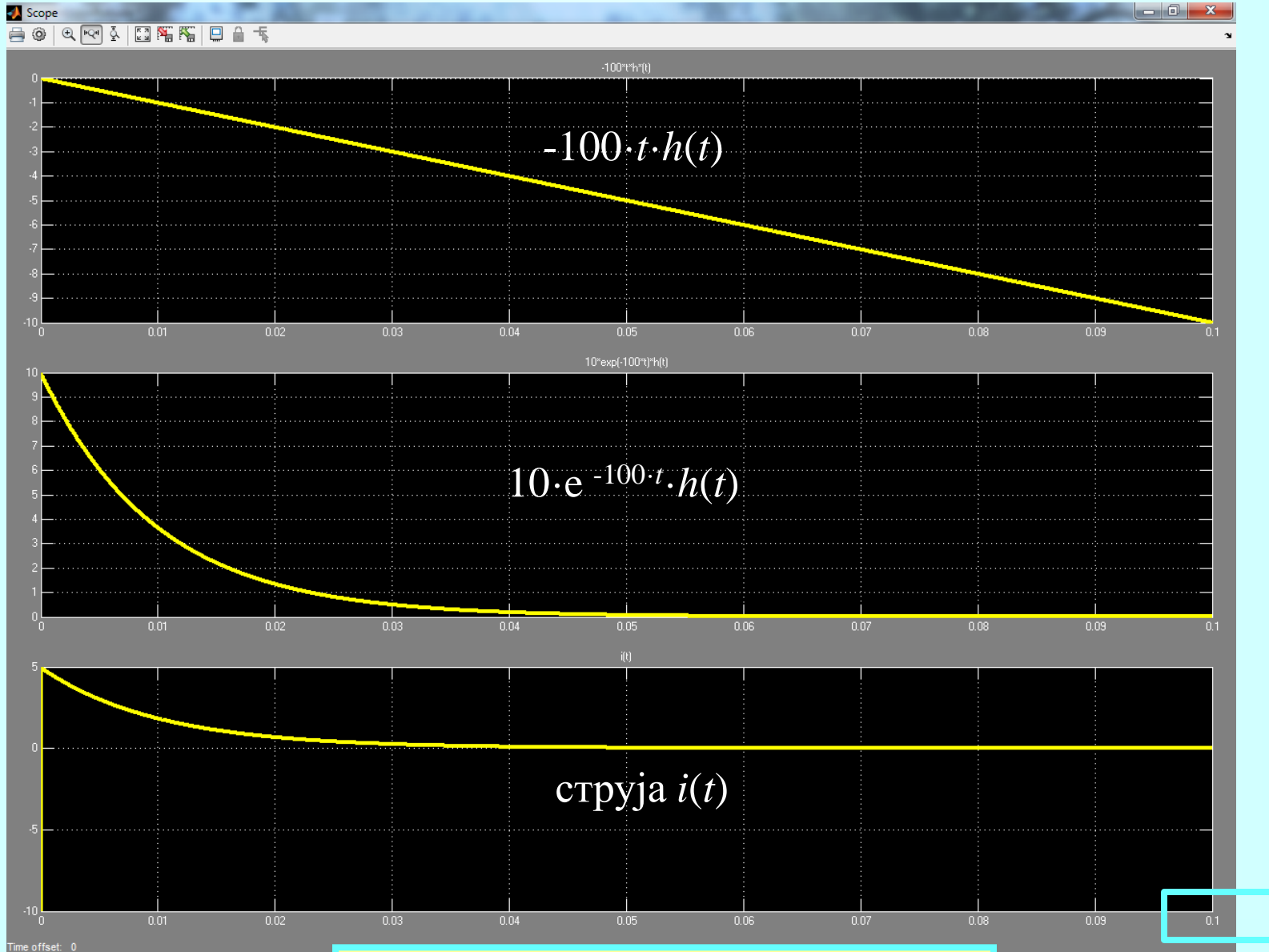
Type:  Solver:

Max step size:  Relative tolerance:

Min step size:  Absolute tolerance:

Initial step size:  Shape preservation:

MATLAB: Simscape  
 Foundation Library  
 Utilities  
 Simulink



$$i(t) = i_0 + i_e = I_0 e^{-\frac{R}{L}t} + \frac{1}{2} I_g \frac{e^{-\frac{R}{L}t} - e^{-at}}{a \frac{L}{R} - 1} e^{-at} h(t), a \neq \frac{R}{L}$$

