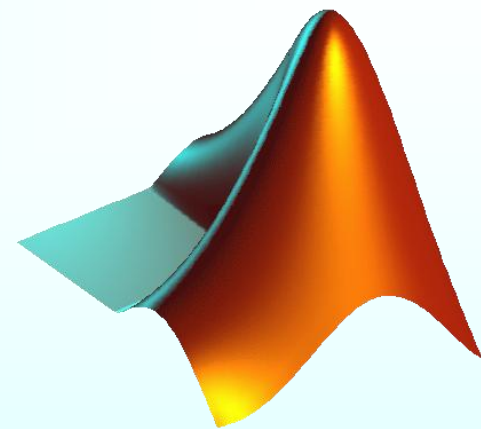
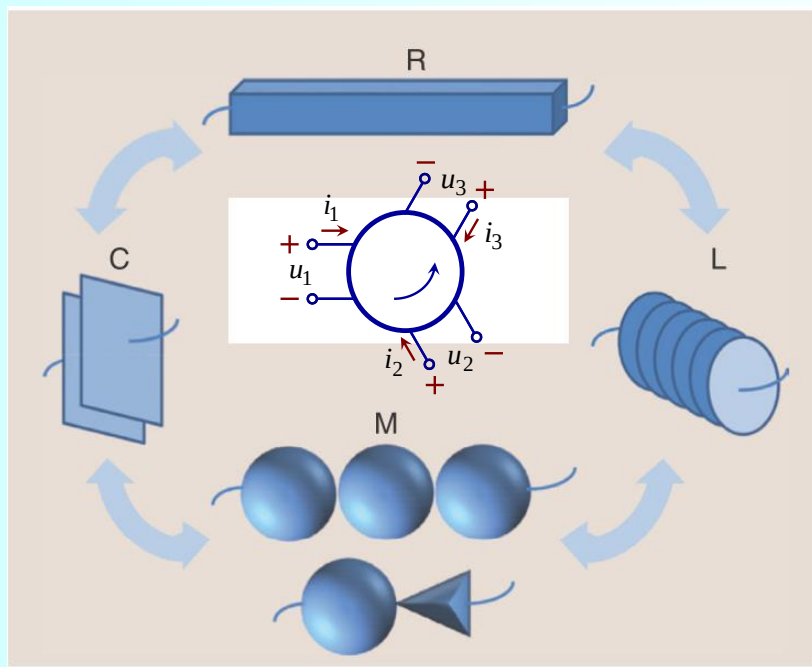


Практикум из рачунарске анализе кола

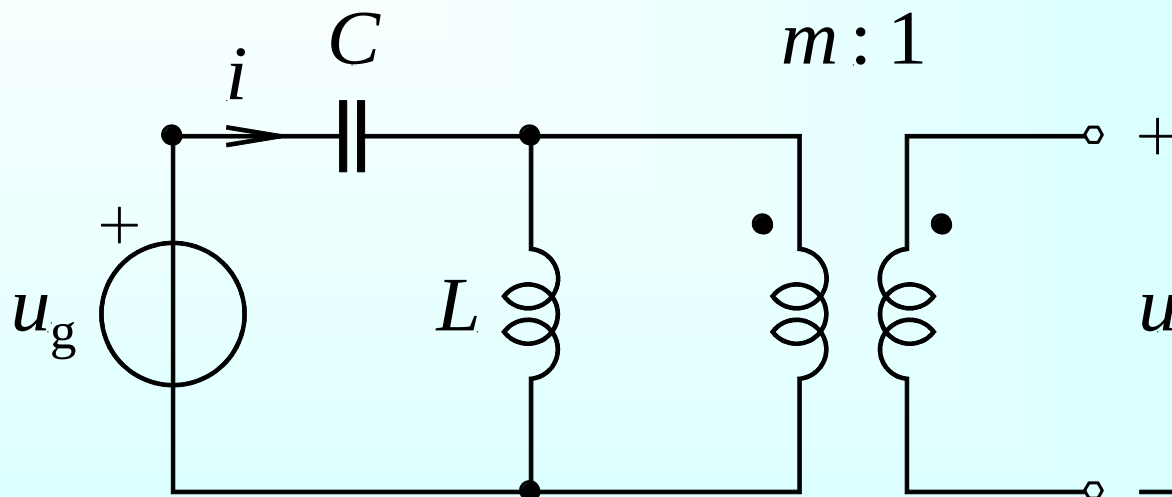


Милка Потребих Иваниш

Никола Баста



Одзив на простопериодичну побуду



$$U_m = 1\text{ V}$$

$$\omega = 10^5 \text{ rad/s}$$

$$L = 0.1 \text{ mH}$$

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

$$m = 1$$

$$u_g(t) = U_m \sin\left(\frac{1}{\sqrt{CL}} t\right) h(t)$$

Задатак - простопериодична побуда

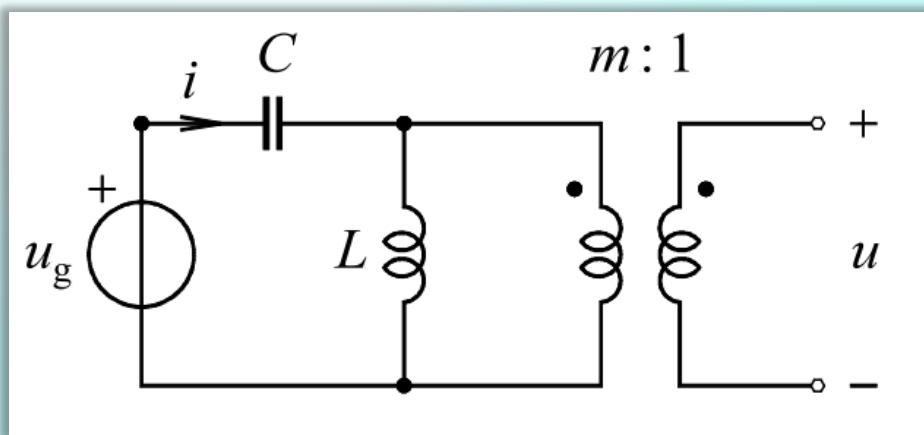
(5) Одредити једначину одзива за напон u електричног кола са слике. Вредности елемената и параметри побуде су познати,

$$u_g(t) = U_m \sin\left(\frac{1}{\sqrt{CL}}t\right) h(t).$$

Нема сакупљене енергије.

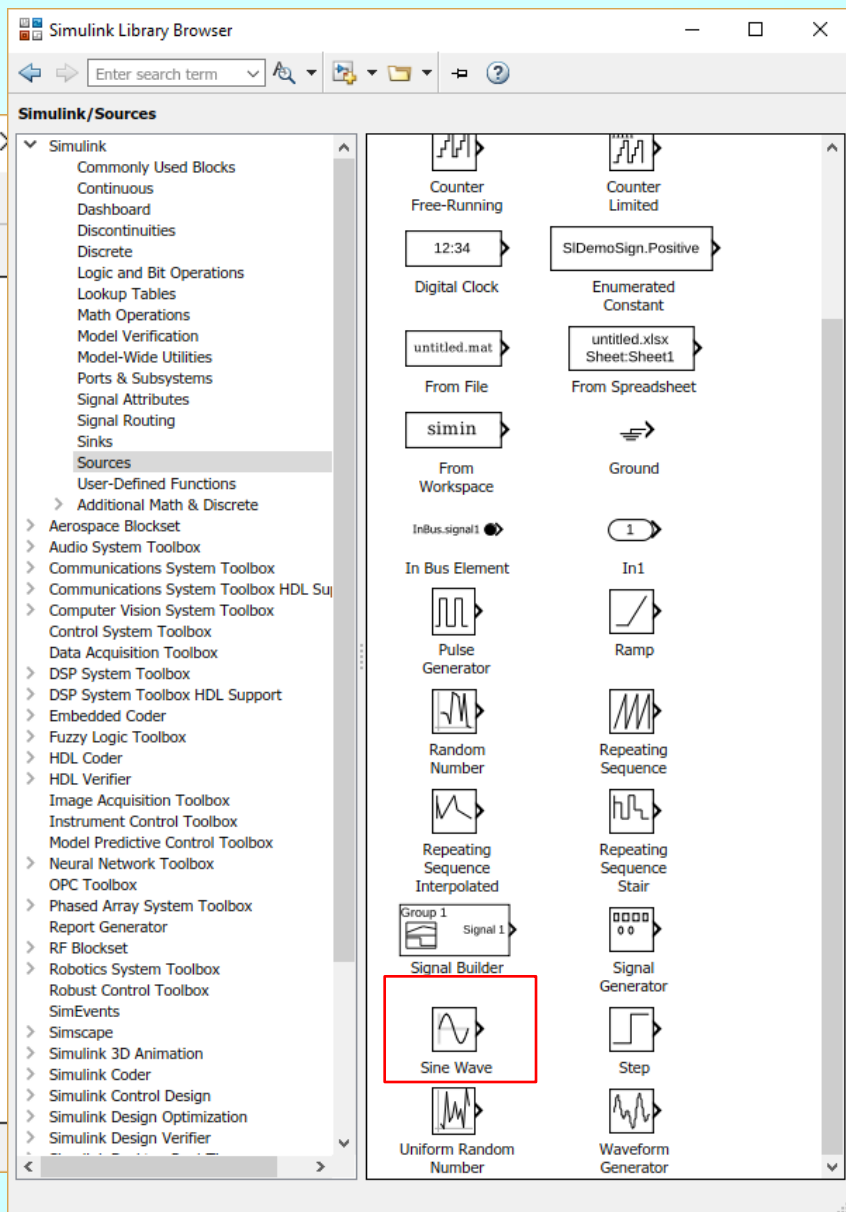
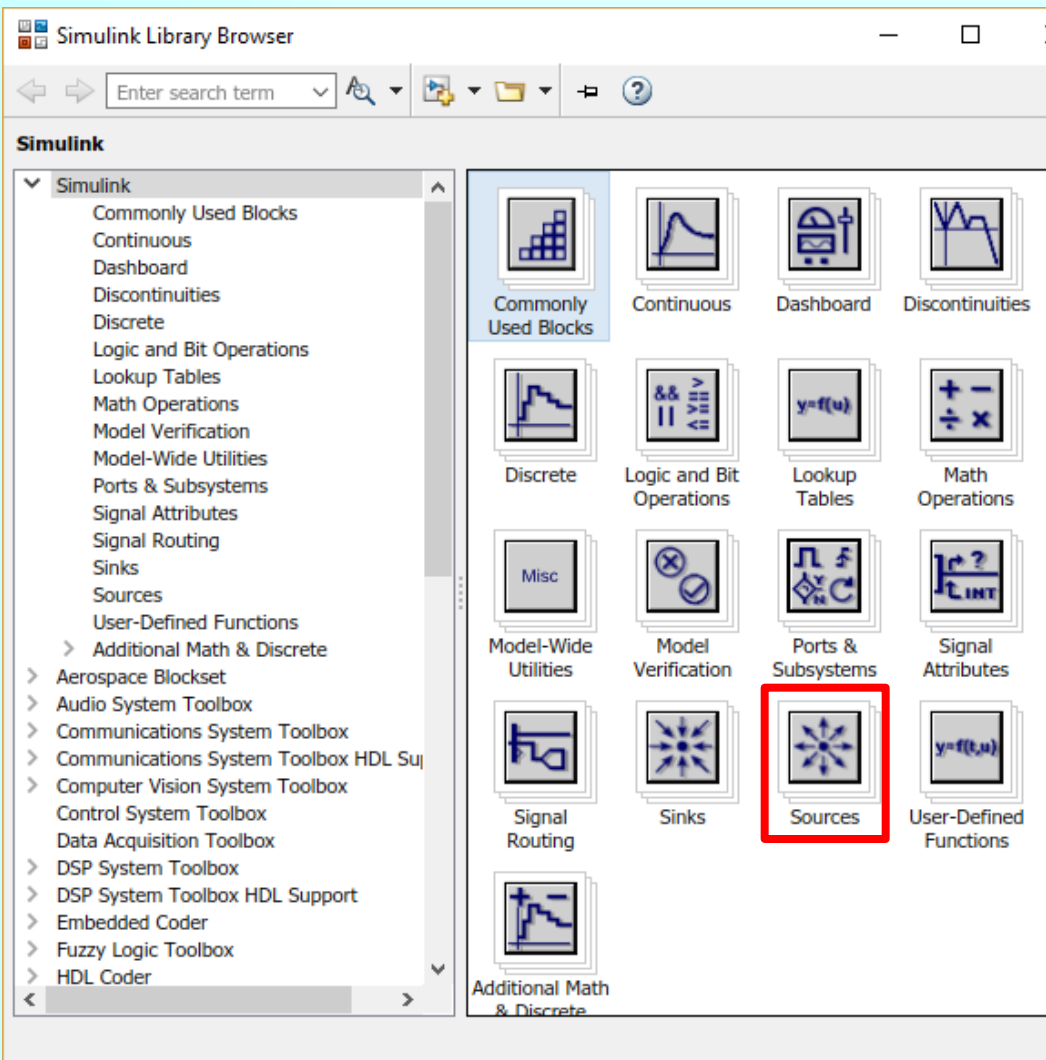
(5) Одредити струју извора, i , и нацртати график струје извора.

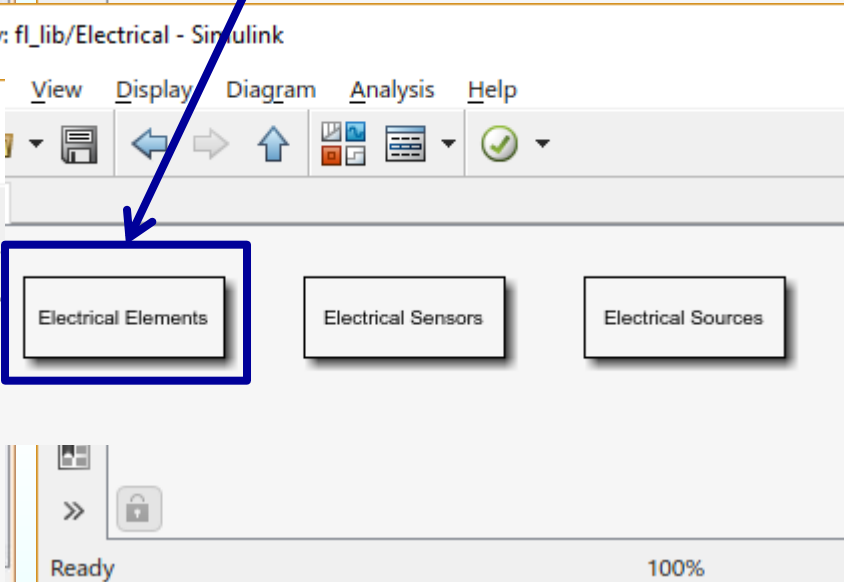
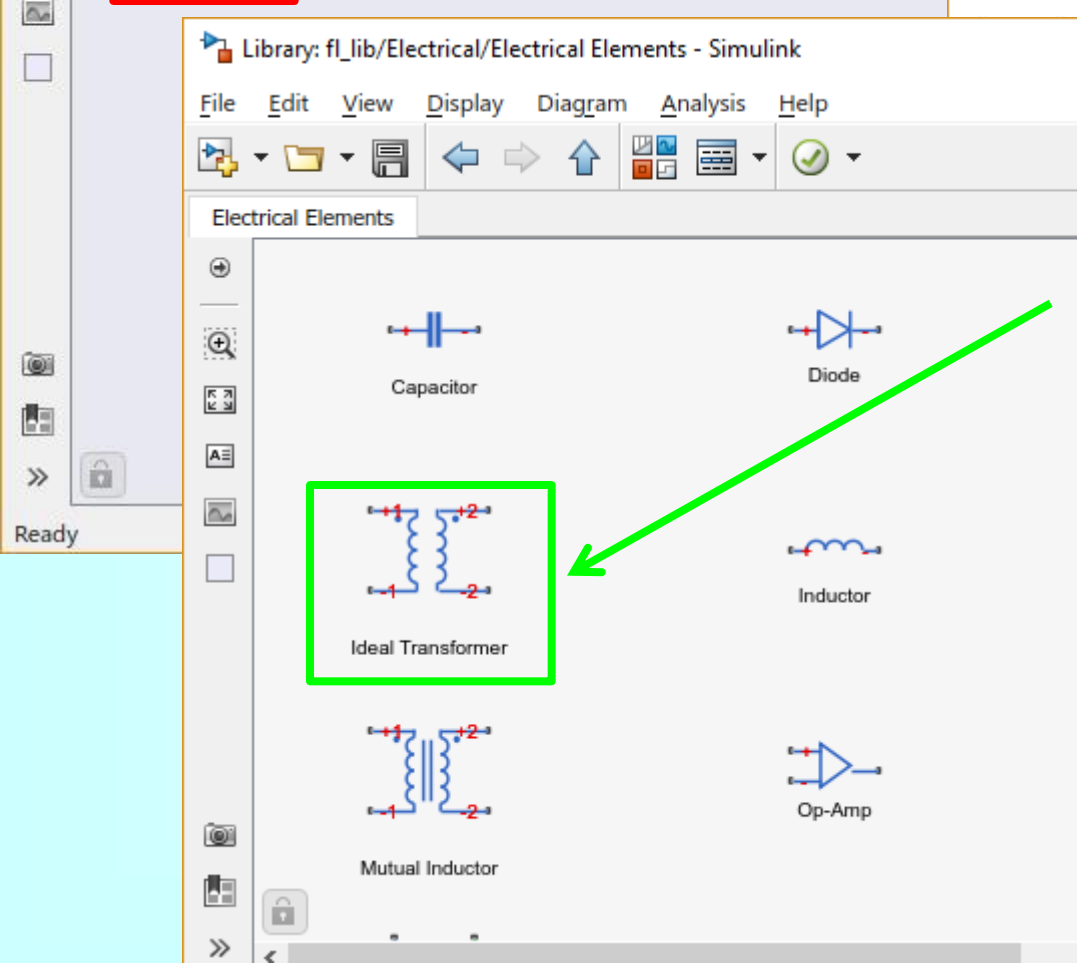
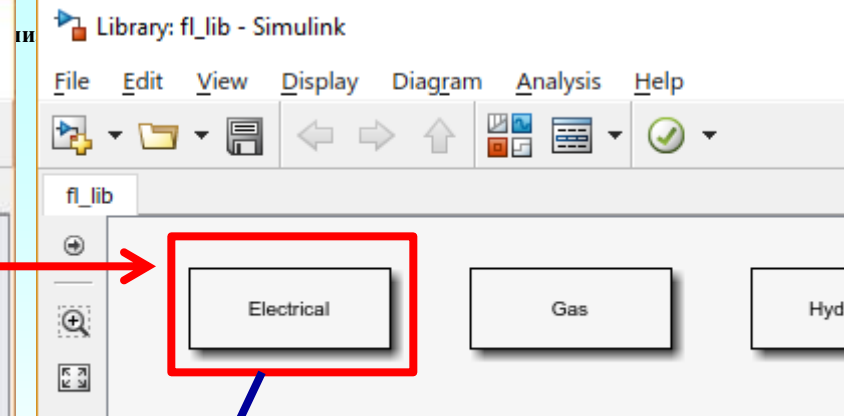
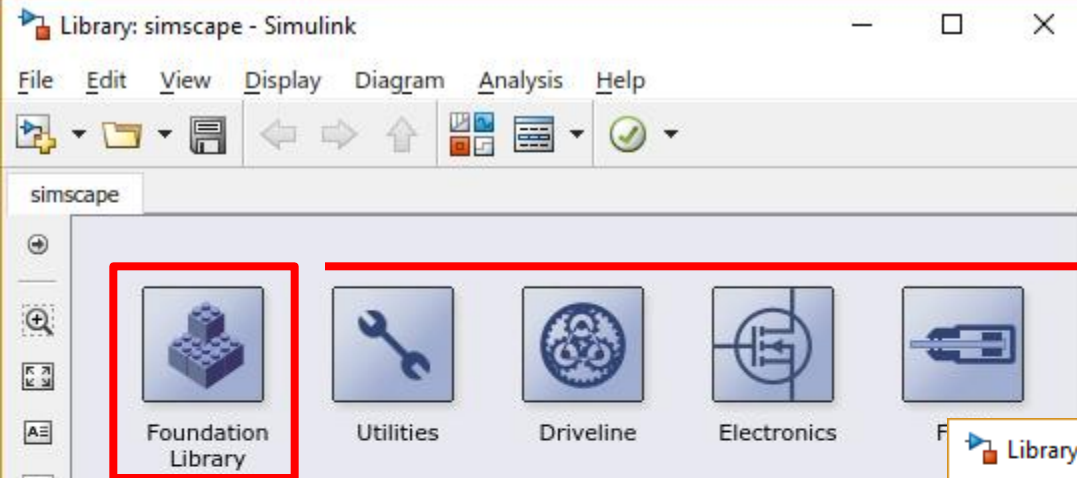
(5) Како гласе једначине стања кола? Написати их у матричном облику.

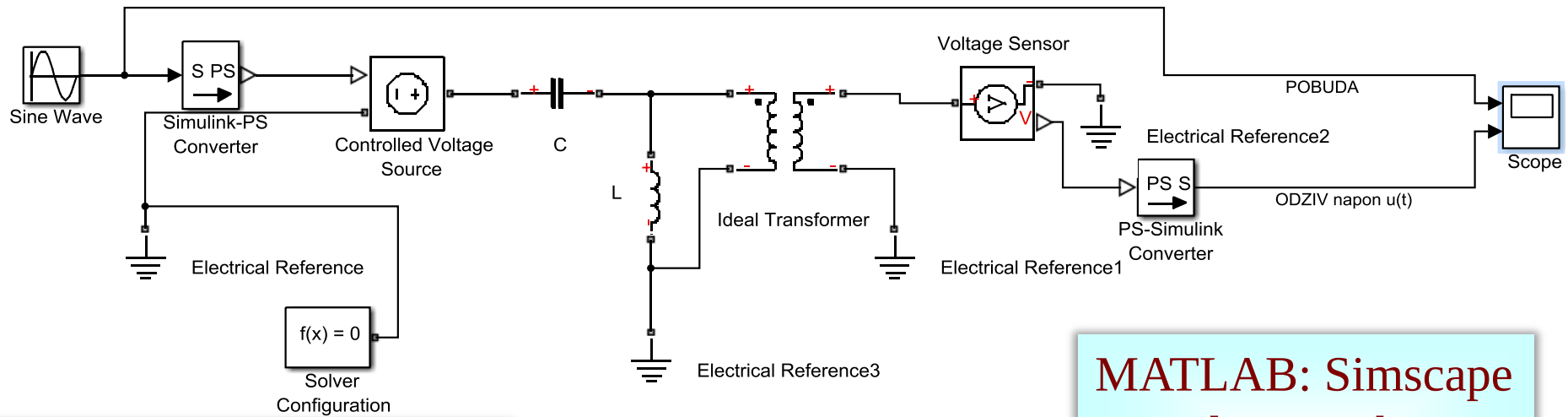


MATLAB: Simulink

Sources







MATLAB: Simscape Foundation Library Utilities Simulink

Source Block Parameters: Sine Wave

Sine Wave

Output a sine wave:

$$O(t) = \text{Amp} \cdot \sin(\text{Freq} \cdot t + \text{Phase}) + \text{Bias}$$

Sine type determines the computational technique used. The parameters in the two types are related through:

Samples per period = $2 \cdot \pi / (\text{Frequency} \cdot \text{Sample time})$

Number of offset samples = $\text{Phase} \cdot \text{Samples per period} / (2 \cdot \pi)$

Use the sample-based sine type if numerical problems due to running for large times (e.g. overflow in absolute time) occur.

Parameters

Sine type: Time based

Time (t): Use simulation time

Amplitude: 1

Bias: 0

Frequency (rad/sec): 100000

Phase (rad): 0

Sample time: 0

☒ Interpret vector parameters as 1-D

OK Cancel Help Apply

Block Parameters: Ideal Transformer

Ideal Transformer

Models an ideal power-conserving transformer satisfying $V_1 = N \cdot V_2$ and $I_2 = N \cdot I_1$ where N is the Winding ratio, V_1 and V_2 are the primary and secondary voltages, I_1 is the current flowing into the primary + terminal, and I_2 is the current flowing out of the secondary + terminal.

This block can be used to represent either an AC transformer or a solid-state DC to DC converter. To model a transformer with inductance and mutual inductance terms, use the Mutual Inductor block.

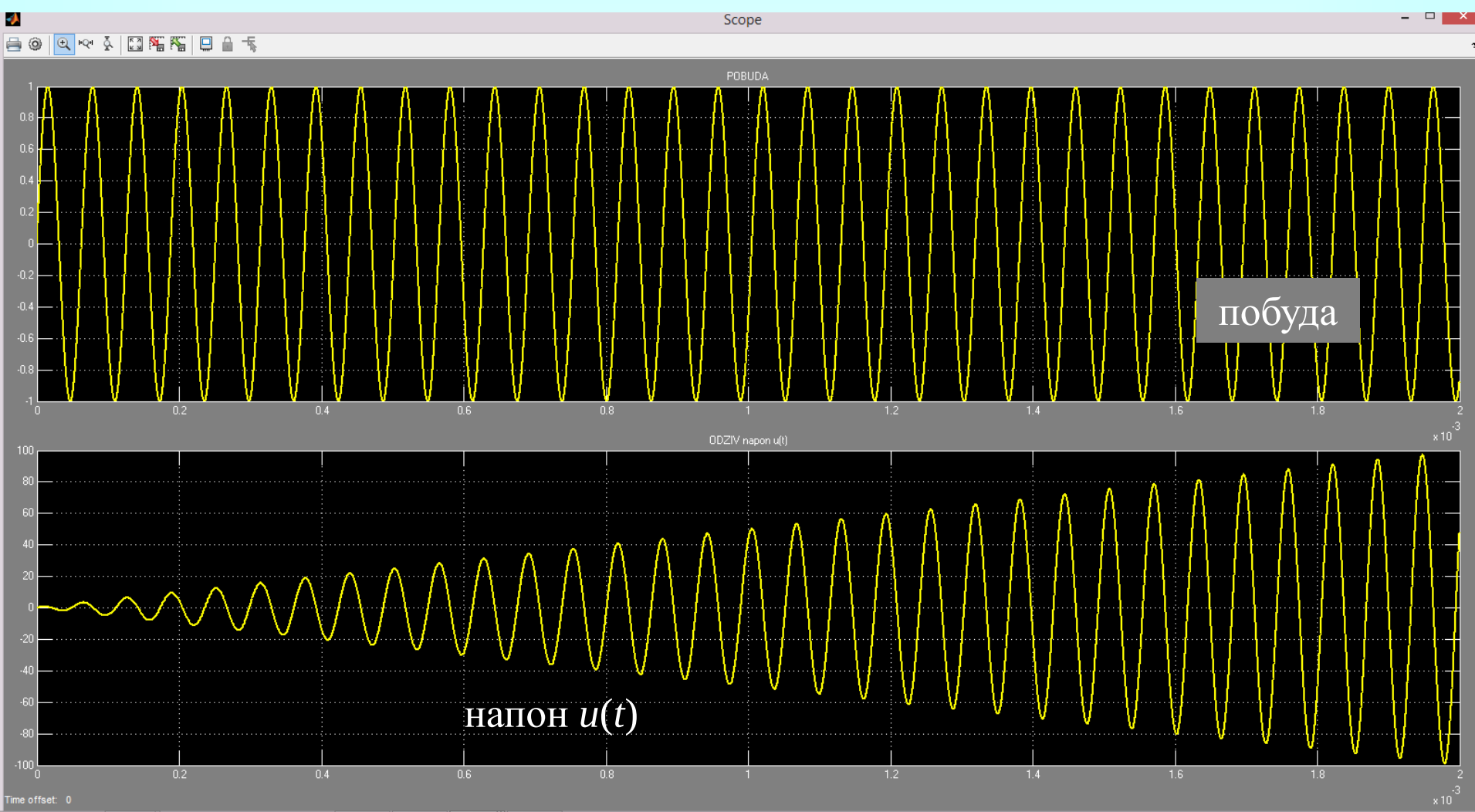
Note that the two electrical networks connected to the primary and secondary windings must each have their own Electrical Reference block.

[View source for Ideal Transformer](#)

Parameters

Winding ratio: 1

OK Cancel Help Apply



Једначине

Postavljanje jednacina

```
clear variables
syms ug uC uL L C DiL DuC m u iC iL Um t
jednacine = [ug == uC + uL,...
             uL == L*DiL,...
             iC == C*DuC,...
             uL==m*u,...
             iC==iL]
```

$\text{jednacine} = (ug = uC + uL \quad uL = DiL L \quad iC = C DuC \quad uL = m u \quad iC = iL)$

Eliminacija i redukcija sistema

```
sistem = eliminate(jednacine, [uL,iC,u])
```

$\text{sistem} = [uC - ug + DiL L, C DuC - iL]$

Resavanje po izvodima stanja

```
resenje = solve(sistem, [DiL,DuC])
```

$\text{resenje} = \text{struct with fields:}$
DiL: $-(uC - ug)/L$
DuC: iL/C

Решавање система

Resavanje sistema diferencijalnih jednacina

```
syms iL(t) uC(t)
```

```
jednacinestanjaFun = subs([diff(uC)==resenje.DuC; diff(iL)==resenje.DiL], {iL,uC}, {iL(t),uC(t)})
```

$$\text{jednacinestanjaFun}(t) = \begin{pmatrix} \frac{\partial}{\partial t} uC(t) = \frac{iL(t)}{C} \\ \frac{\partial}{\partial t} iL(t) = \frac{u_g - uC(t)}{L} \end{pmatrix}$$

```
zamene = ug==Um*sin(t/sqrt(C*L))*heaviside(t)
```

```
zamene =
```

$$u_g = U_m \sin\left(\frac{t}{\sqrt{C L}}\right) \text{heaviside}(t)$$

```
jednacinestanjaFunR=subs(jednacinestanjaFun,lhs(zamene),rhs(zamene))
```

$$\text{jednacinestanjaFunR}(t) = \begin{pmatrix} \frac{\partial}{\partial t} uC(t) = \frac{iL(t)}{C} \\ \frac{\partial}{\partial t} iL(t) = -\frac{uC(t) - U_m \sin\left(\frac{t}{\sqrt{C L}}\right) \text{heaviside}(t)}{L} \end{pmatrix}$$

```
assume(t>0 & C>0 & L>0)
```

```
resenjeDiff=dsolve(jednacinestanjaFunR,[uC(0)==0,iL(0)==0],'IgnoreAnalyticConstraints',false)
```

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

```
iL: cos(t/(C^(1/2)*L^(1/2)))*((C^(1/2)*Um)/(2*L^(1/2)) - (C^(1/2)*Um*cos(t/(C^(1/2)*L^(1/2))))^2/(2*L^(1/2))
uC: (L^(1/2)*sin(t/(C^(1/2)*L^(1/2)))*((C^(1/2)*Um)/(2*L^(1/2)) - (C^(1/2)*Um*cos(t/(C^(1/2)*L^(1/2))))^2/(2*L^(1/2))
```

```
simplify(resenjeDiff.iL)
```

```
ans =
```

$$\frac{U_m t \sin\left(\frac{t}{\sqrt{C} \sqrt{L}}\right)}{2 L}$$

График функције одзива

```
simplify(resenjeDiff.iL)
```

ans =

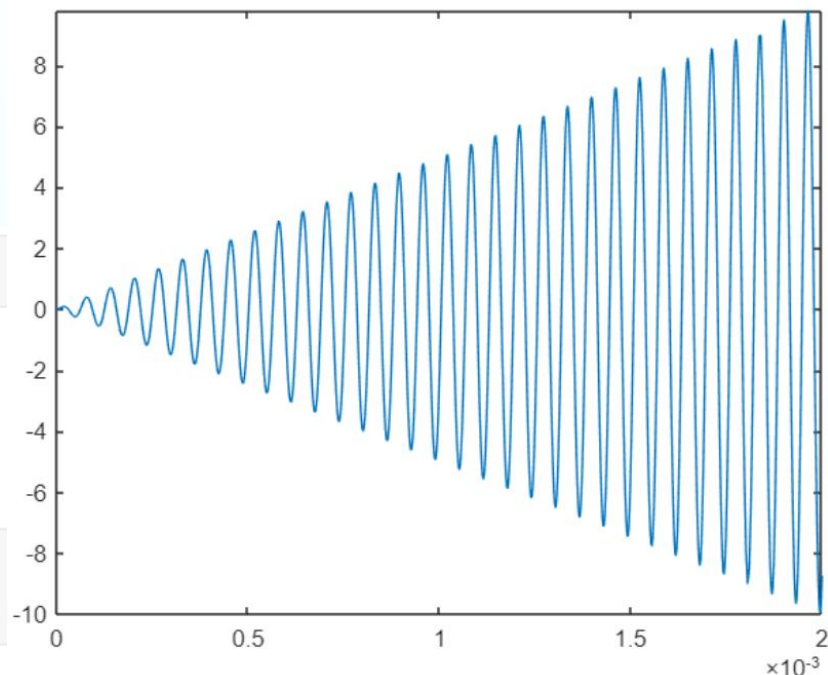
$$\frac{U_m t \sin\left(\frac{t}{\sqrt{C} \sqrt{L}}\right)}{2 L}$$

```
numzamene=[L==1e-4, C==1e-6, Um==1, m==1]
```

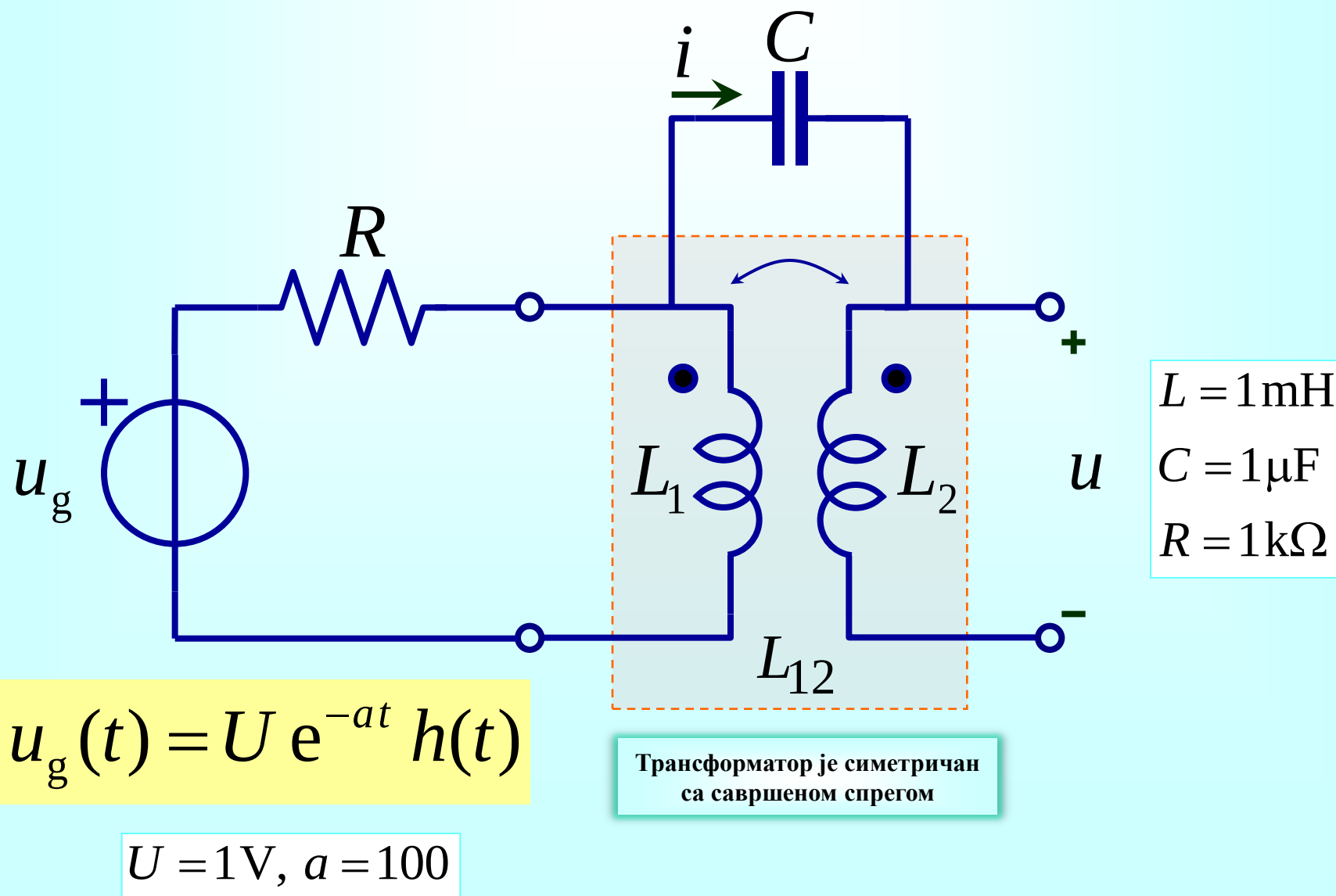
numzamene =

$$\left(L = \frac{1}{10000} \quad C = \frac{1}{1000000} \quad U_m = 1 \quad m = 1\right)$$

```
fplot(t, subs(resenjeDiff.iL, lhs(numzamene), rhs(numzamene)),[0 0.002])
```

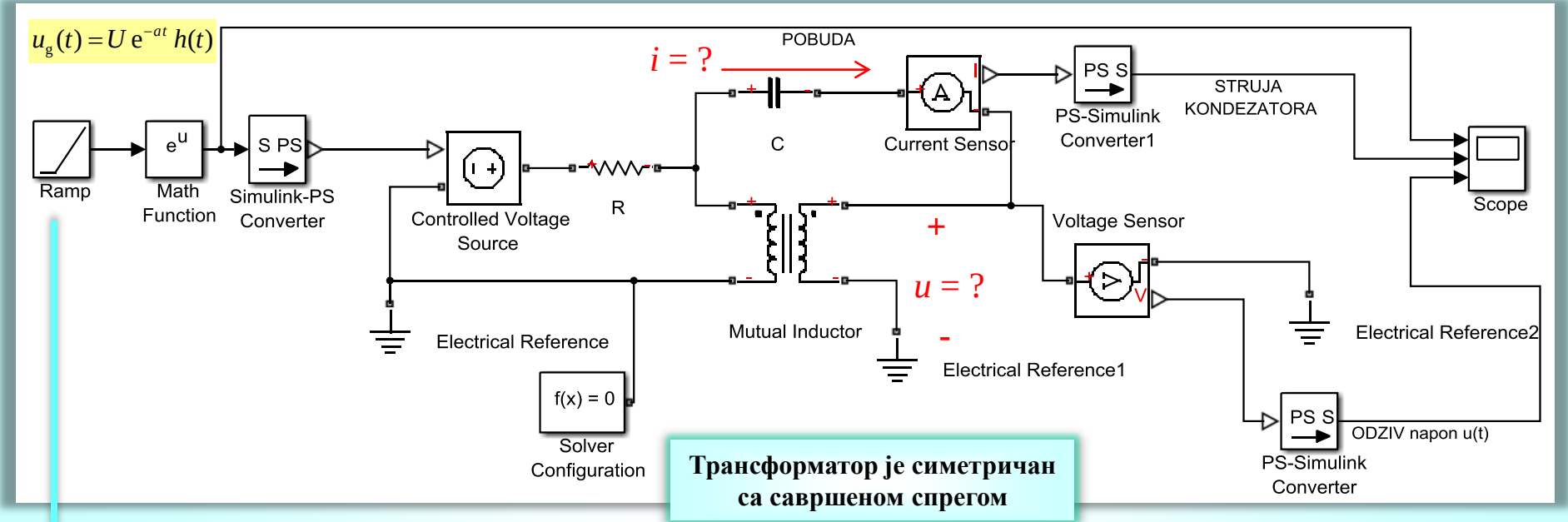


Одзив на експоненцијалну побуду



Задатак - експоненцијална побуда

- Параметри кола са слике су познати. Линеарни индуктивни трансформатор је симетричан са савршеном спрегом. Нема сакупљене енергије.
- Одредити струју кондензатора $i(t)$ и нацртати њен график.
- Одредити напон секундара $u(t)$ и нацртати његов график.



Source Block Parameters: Ramp

Ramp (mask) (link)

Output a ramp signal starting at the specified time.

Parameters

Slope:

-100

Start time:

0

Initial output:

0

☒ Interpret vector parameters as 1-D

OK Cancel Help Apply

Block Parameters: Mutual Inductor

Mutual Inductor

Models a mutual inductor. If winding 1 has voltage V_1 across it and current I_1 flowing into its + terminal, and winding 2 has voltage V_2 across it and current I_2 flowing into its + terminal, then

$$V_1 = L_1 \frac{dI_1}{dt} + M \frac{dI_2}{dt}$$
$$V_2 = L_2 \frac{dI_2}{dt} + M \frac{dI_1}{dt}$$

where parameters L_1 and L_2 are the winding self-inductances, and M is the mutual inductance. M is defined in terms of the Coefficient of coupling k by $M = k \sqrt{L_1 L_2}$. Hence k should be greater than zero and less than one.

The parameters Winding 1 initial current and Winding 2 initial current set the initial current through windings 1 and 2. Note that this value is not used if the solver configuration is set to Start simulation from steady state.

[View source for Mutual Inductor](#)

Parameters

Inductance L1: 1 mH

Inductance L2: 1 mH

Coefficient of coupling: 0.9999

Winding 1 initial current: 0 A

Winding 2 initial current: 0 A

OK Cancel Help Apply

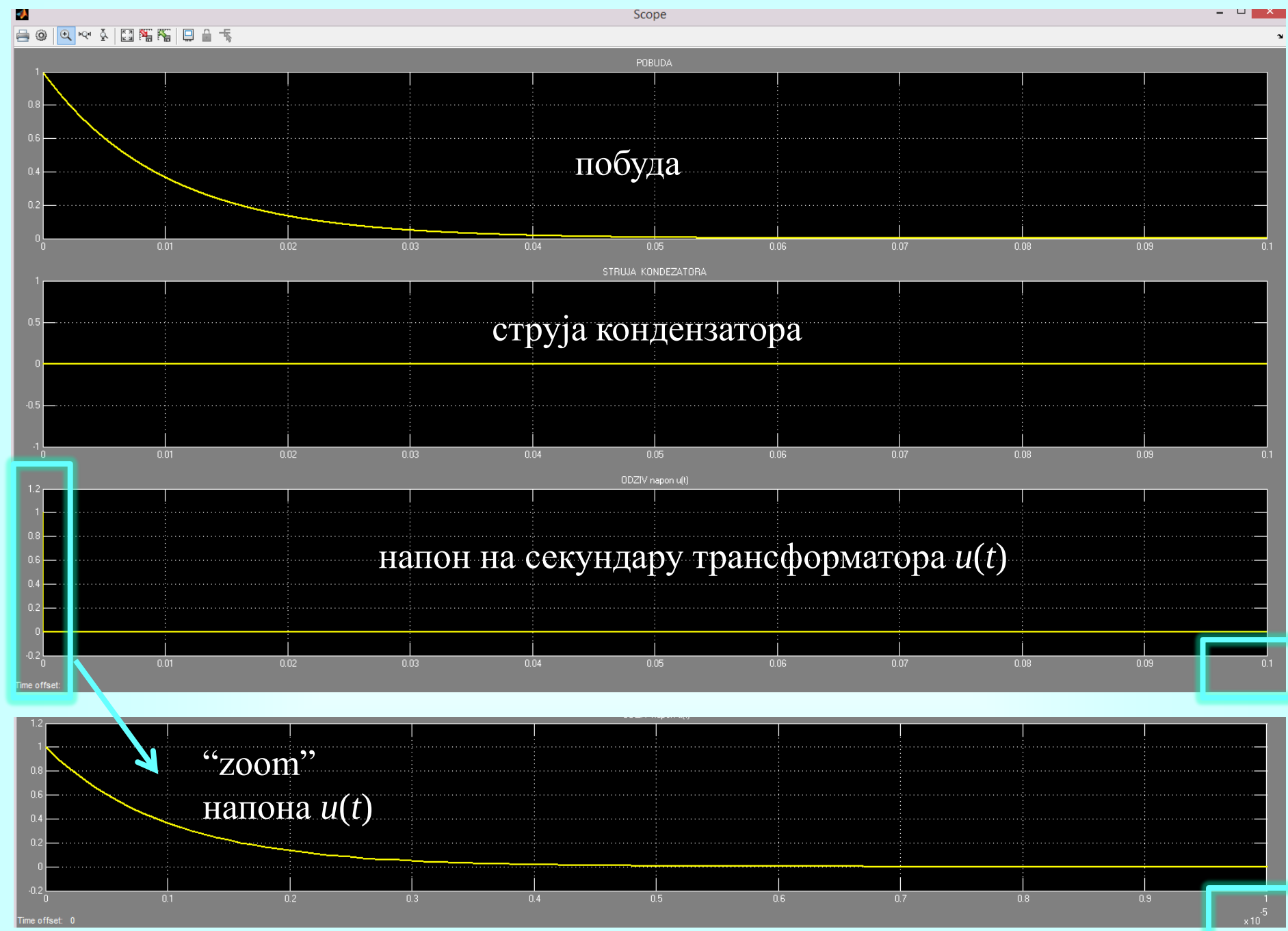
$L = 1 \text{ mH}$

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

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

MATLAB: Simscape
Foundation Library
Utilities
Simulink

$U = 1 \text{ V}, a = 100$



Једначине

Jednachine

MATLAB: Symbolic Math Toolbox

```
clear variables
syms C L L1 L12 L2 R U a DiL1 DiL2 DuC i iL1 iL2 iR u uC uL1 uR t ug
assume(0 < R & 0 < C & 0 < L1 & 0 < L2 & 0 < L12 & 0 < L & 0 < U & 0 < a)
```

```
zamene = [L1 == L, L2 == L, L12 == L]
```

```
zamene = (L1 = L L2 = L L12 = L)
```

```
pobuda = ug == U*exp(-(a*t))*heaviside(t)
```

```
pobuda = ug = U e-at heaviside(t)
```

```
jednachine = [ug == uR + uL1,...
    uR == R * iR,...
    uL1 == L1 * DiL1 + L12 * DiL2,...
    u == L2 * DiL2 + L12 * DiL1,...
    uC == uL1 - u,...
    i == C * DuC,...
    iR == i + iL1,...
    i == iL2]
```

```
jednachine = (ug = uL1 + uR uR = R iR uL1 = DiL1 L1 + DiL2 L12 u = DiL2 L2 + DiL1 L12 uC = uL1 - u i = C DuC iR = i + iL1 i = iL2)
```

```
promenljive = [uL1, u, uR, iR, i]
```

```
promenljive = (uL1 u uR iR i)
```

Једначине извода

MATLAB: Symbolic Math Toolbox

Jednacine izvoda

```
jednacineIzvoda = eliminate(jednacine, promenljive)

jednacineIzvoda = [R iL1 - ug + R iL2 + DiL1 L1 + DiL2 L12, C DuC - iL2, uC - DiL1 L1 + DiL2 L2 + DiL1 L12 - DiL2 L12]

JednacineIzvodaZamene = subs(jednacineIzvoda, lhs(zamene), rhs(zamene))

JednacineIzvodaZamene = (R iL1 - ug + R iL2 + DiL1 L + DiL2 L - C DuC - iL2 - uC)

syms iL1(t) uC(t) iL2(t)
PromenljiveStanjaIzvodi = [uC == uC(t), DuC == diff(uC(t)), iL1 == iL1(t), DiL1 == diff(iL1(t)), iL2 == iL2(t), DiL2 == diff(iL2(t))]

PromenljiveStanjaIzvodi(t) =
    (uC(t) = uC(t) DuC = \frac{\partial}{\partial t} uC(t) iL1(t) = iL1(t) DiL1 = \frac{\partial}{\partial t} iL1(t) iL2(t) = iL2(t) DiL2 = \frac{\partial}{\partial t} iL2(t))

JednacinePromenljiveStanja = subs(JednacineIzvodaZamene, [lhs(PromenljiveStanjaIzvodi) lhs(pobuda)], [rhs(PromenljiveStanjaIzvodi) rhs(pobuda)])

JednacinePromenljiveStanja =
    (L \frac{\partial}{\partial t} iL1(t) + L \frac{\partial}{\partial t} iL2(t) + R iL1(t) + R iL2(t) - U e^{-at} heaviside(t) - C \frac{\partial}{\partial t} uC(t) - iL2(t) - uC(t))

promenljiveStanja = [iL1(t), iL2(t), uC(t)]

promenljiveStanja = (iL1(t) iL2(t) uC(t))
```


Елиминација сувишних једначина

Redukcija reda sistema

```
[noveJednacinе, novePromenljive, Rstruct] = reduceRedundancies(JednacinеPromenljiveStanja, promenljiveStanja)
```

```
noveJednacinе =
```

$$e^{-at} \left(L e^{at} \frac{\partial}{\partial t} iL_1(t) - U \text{heaviside}(t) + R e^{at} iL_1(t) \right)$$

```
novePromenljive = iL1(t)
```

```
Rstruct = struct with fields:
```

```
    solvedEquations: [2×1 sym]
```

```
    constantVariables: [2×2 sym]
```

```
    replacedVariables: [0×2 sym]
```

```
    otherEquations: [0×1 sym]
```

MATLAB: Symbolic Math Toolbox

```
reseneJednacinе = Rstruct.solvedEquations
```

```
reseneJednacinе =
```

$$\begin{pmatrix} uC(t) \\ -iL_2(t) \end{pmatrix}$$

```
konstantnePromenljive = Rstruct.constantVariables
```

```
konstantnePromenljive =
```

$$\begin{pmatrix} uC(t) & 0 \\ iL_2(t) & 0 \end{pmatrix}$$

```
zamenjenePromenljive = Rstruct.replacedVariables
```

```
zamenjenePromenljive =
```

```
Empty sym: 0-by-2
```

```
drugeJednacinе = Rstruct.otherEquations
```

```
drugeJednacinе =
```

```
Empty sym: 0-by-1
```

Решавање диференцијалне једначине

Resavanje diferencijalne jednacine

```
odziv_il1 = dsolve(noveJednacine, il1(0) == 0, 'IgnoreAnalyticConstraints', false)
```

odziv_il1 =

$$\frac{U e^{-\frac{Rt}{L}} e^{-at} \operatorname{heaviside}(t) \left(e^{\frac{Rt}{L}} - e^{at} \right)}{R - La}$$

MATLAB: Symbolic Math Toolbox

Напон $u(t)$

```
odziv_u = subs(L12*diff(expand(odziv_il1)), lhs(zamene), rhs(zamene))
```

odziv_u =

$$L \left(\frac{U e^{-at} \delta(t)}{R - La} - \frac{U e^{-\frac{Rt}{L}} \delta(t)}{R - La} - \frac{U a e^{-at} \operatorname{heaviside}(t)}{R - La} + \frac{RU e^{-\frac{Rt}{L}} \operatorname{heaviside}(t)}{L (R - La)} \right)$$

```
odziv_u_simp = simplify(odziv_u)
```

odziv_u_simp =

$$-L \left(\frac{U a e^{-at} \operatorname{heaviside}(t)}{R - La} - \frac{RU e^{-\frac{Rt}{L}} \operatorname{heaviside}(t)}{L (R - La)} \right)$$

```
vrednosti = [U==1, R==1000, C==1e-6, L==1e-3, a==100]
```

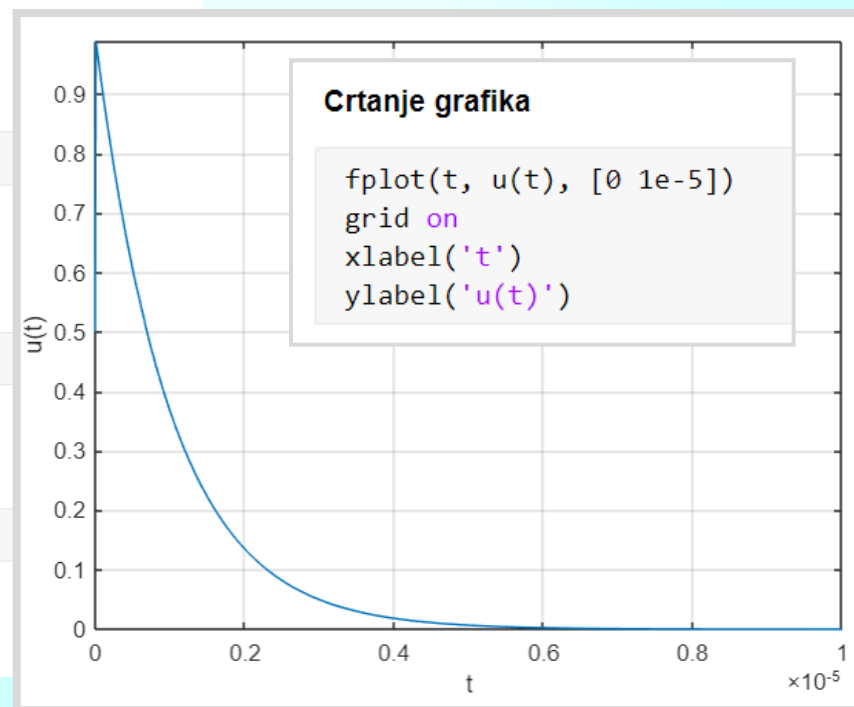
vrednosti =

$$\left(U = 1 \quad R = 1000 \quad C = \frac{1}{1000000} \quad L = \frac{1}{1000} \quad a = 100 \right)$$

```
u(t) = subs(odziv_u_simp, lhs(vrednosti), rhs(vrednosti))
```

u(t) =

$$\frac{10000 e^{-1000000t} \operatorname{heaviside}(t)}{9999} - \frac{e^{-100t} \operatorname{heaviside}(t)}{9999}$$



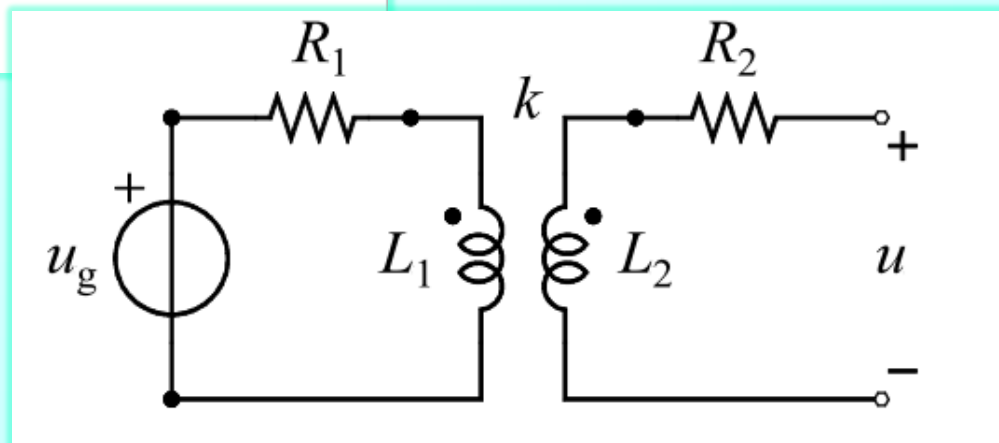
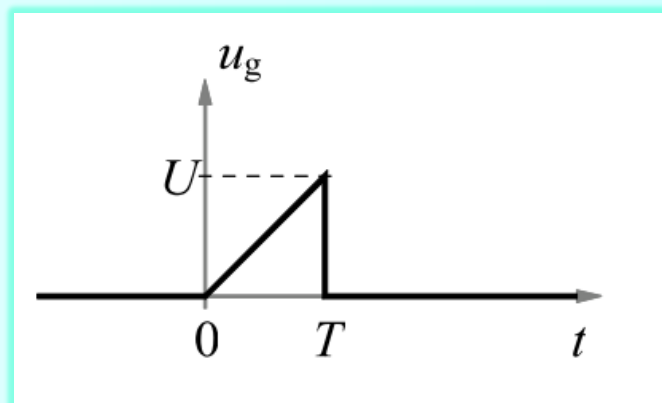
Одзив на троугаони импулс

Параметри електричног кола са слике су познати. Побуда је дата на слици, $T = L/R$, $R_1 = R$, $R_2 = 2R$, $L_1 = L$, $L_2 = 4L$, $k = 1/2$.

(5) Одредити индициону функцију за напон отвореног секундара (одскочни одзив).

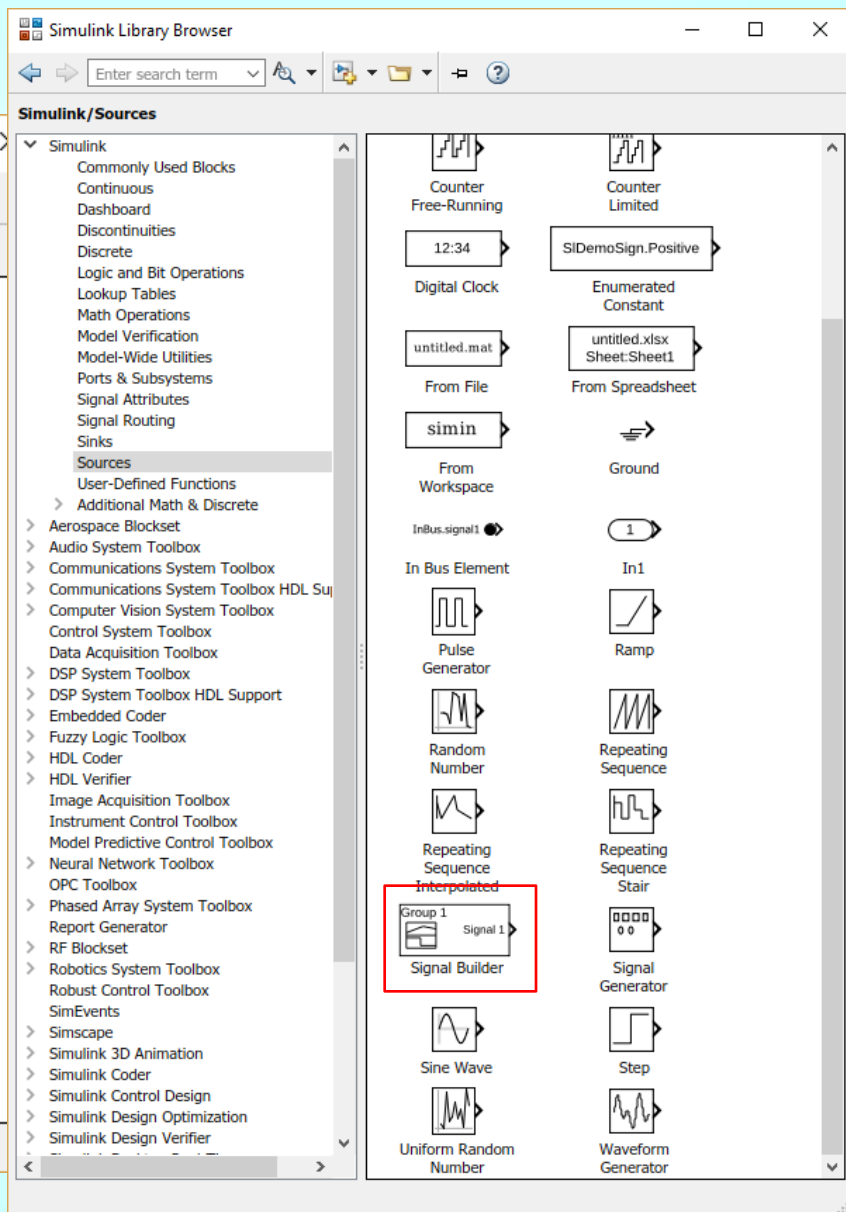
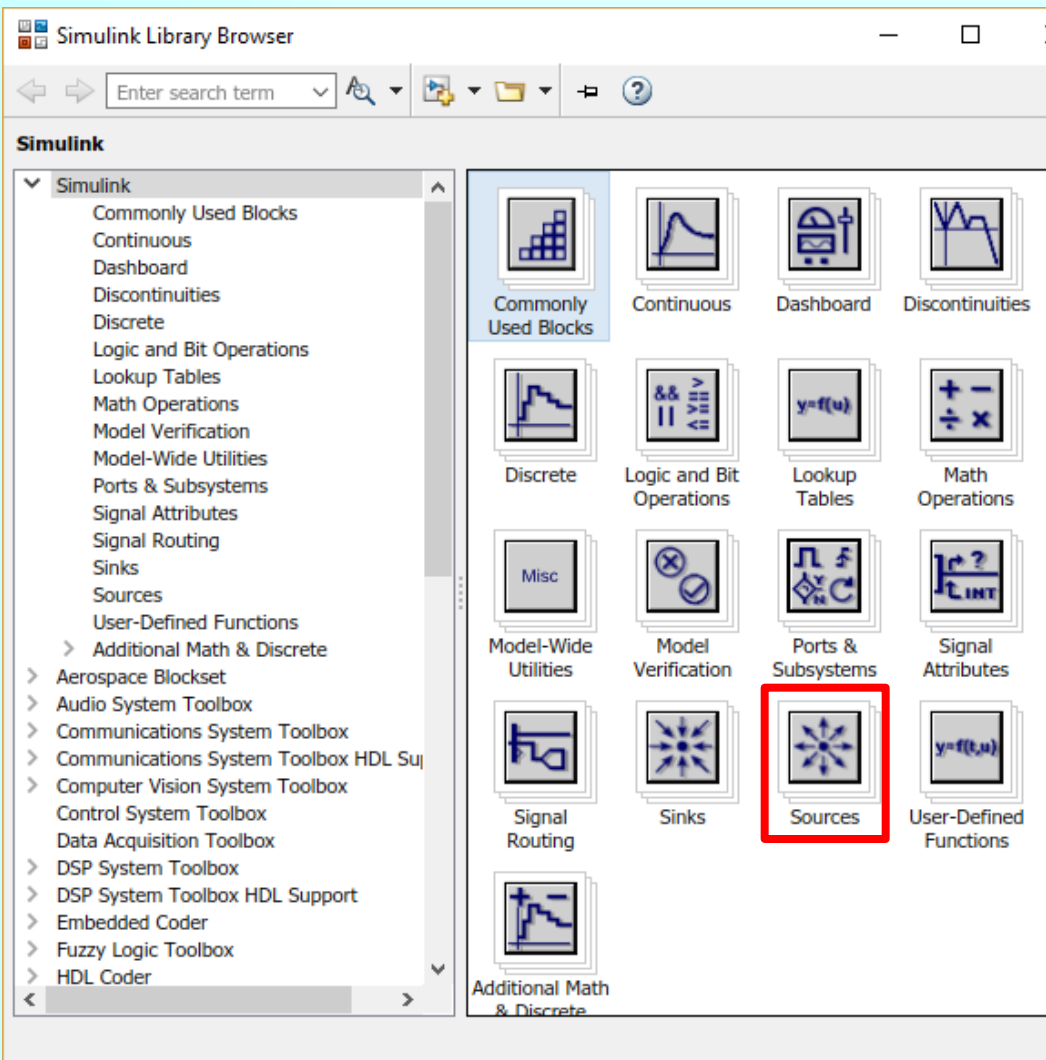
(5) Одредити напон отвореног секундара и

(5) нацртати његов график. Обележити осе графика, координатни почетак, пресеке и додире графика са осама, и тачке екстремума.

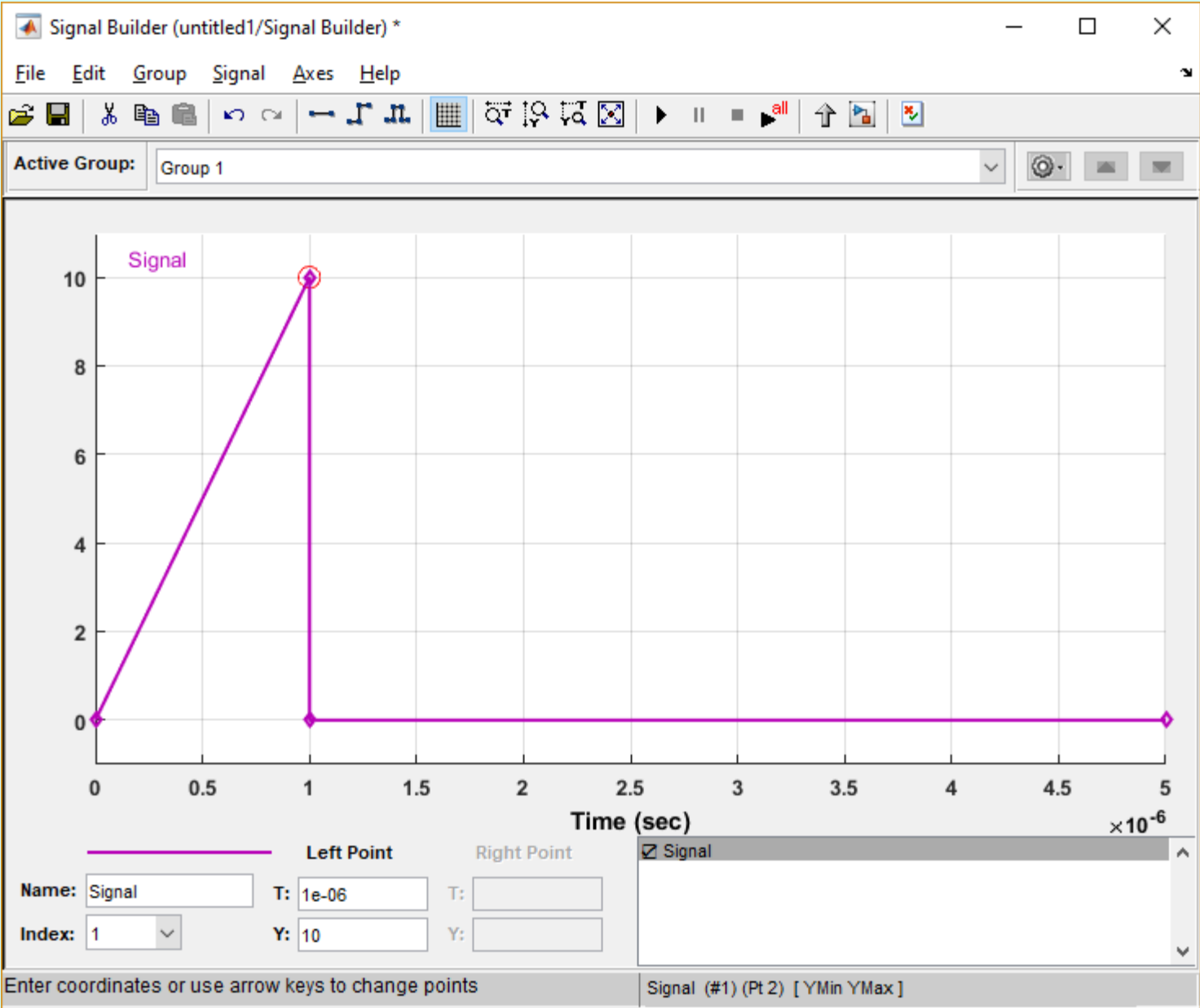


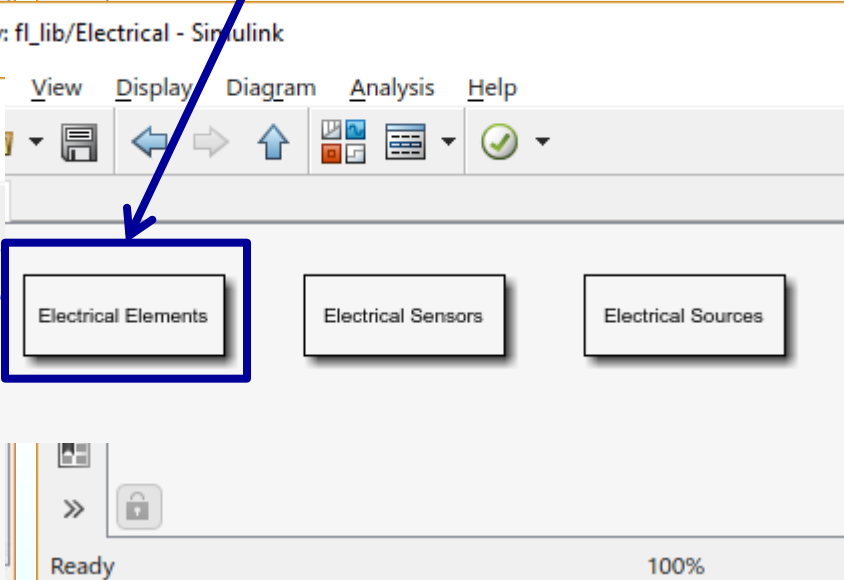
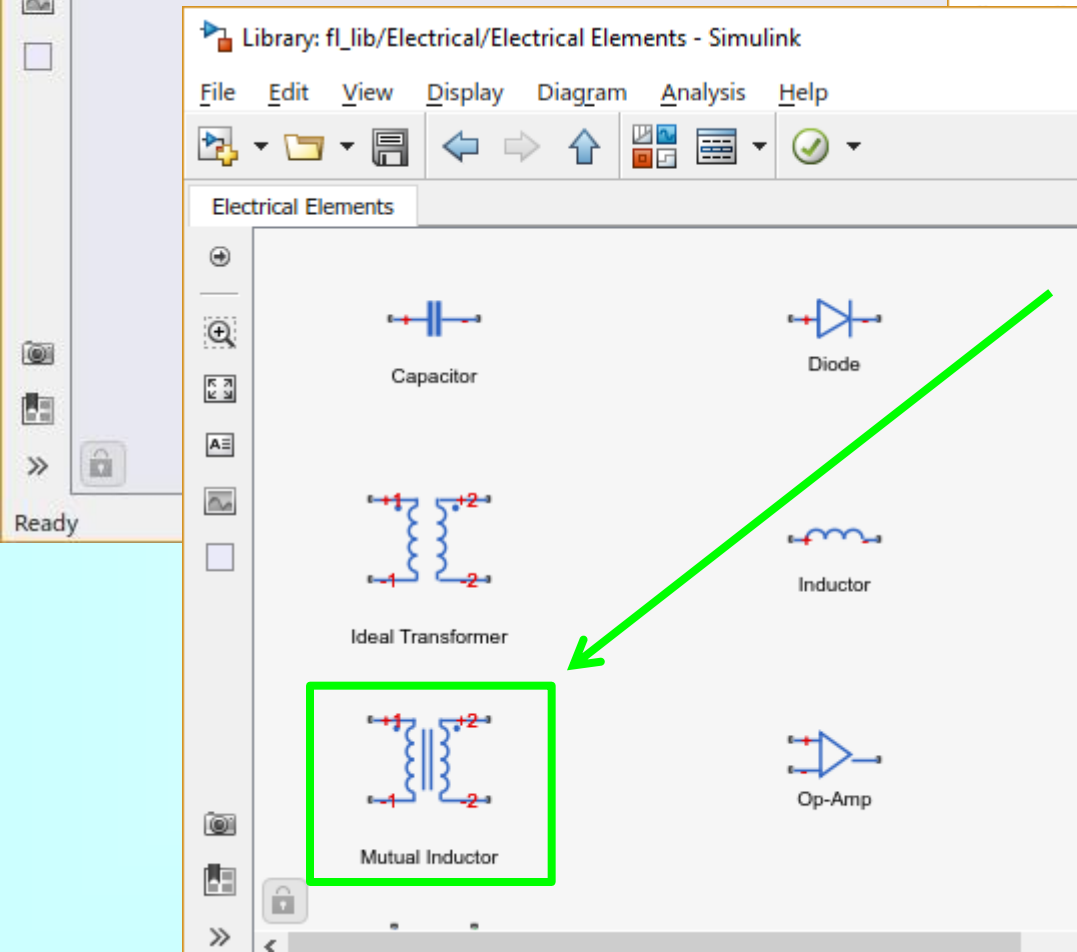
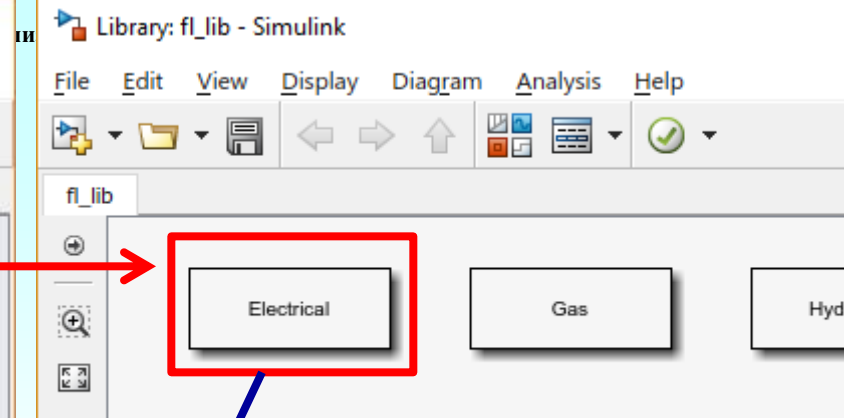
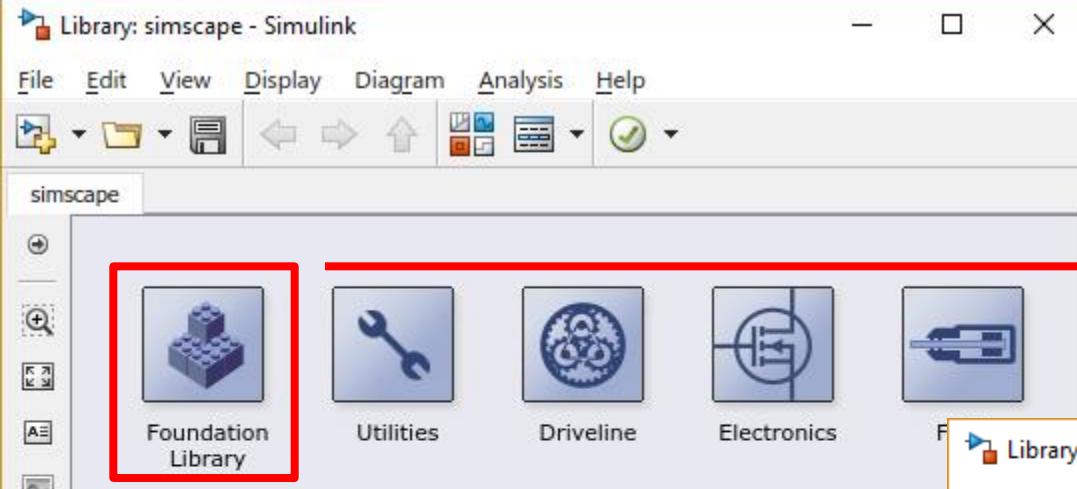
MATLAB: Simulink

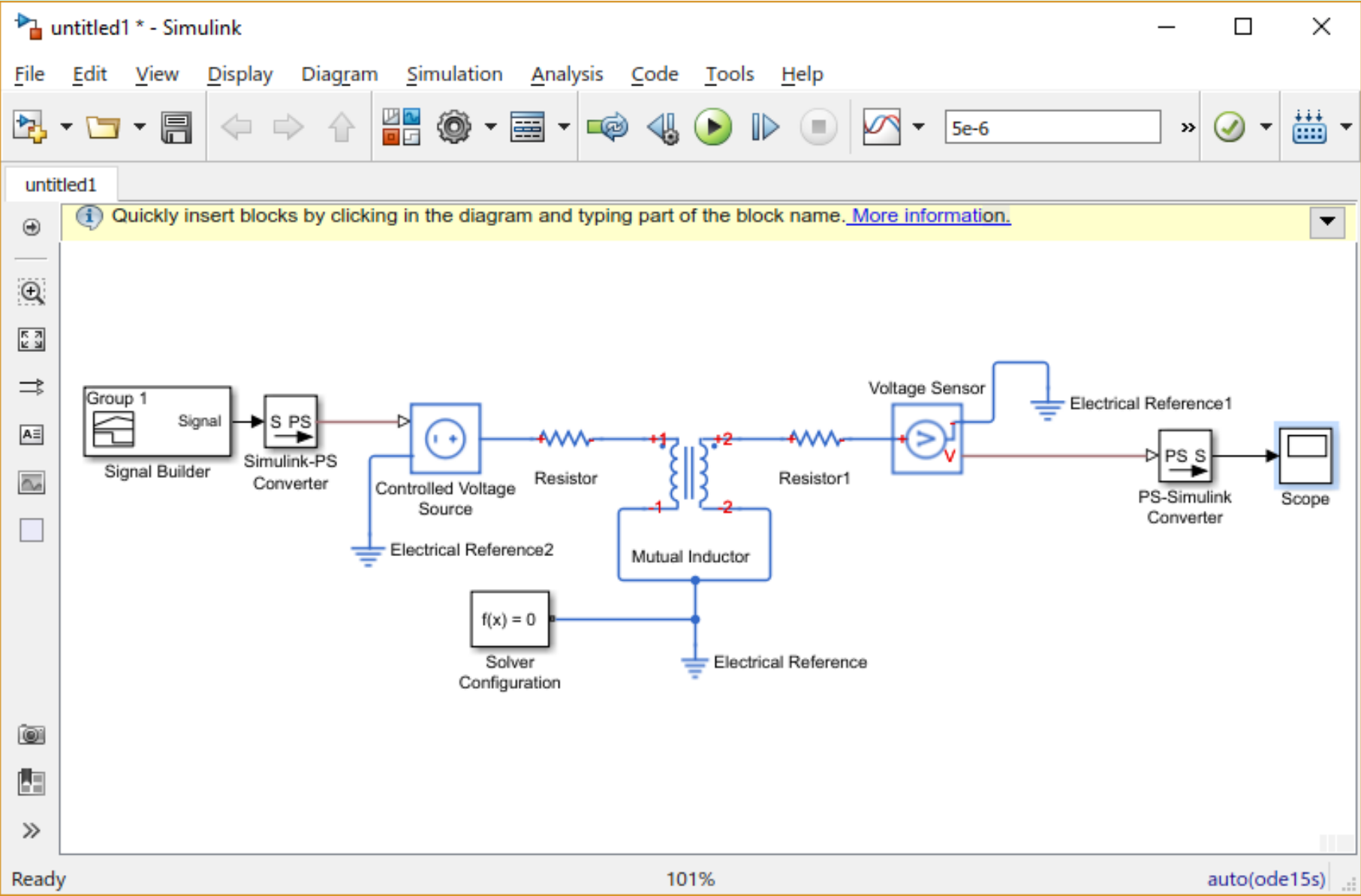
Sources



Signal Builder







Block Parameters: Mutual Inductor

Mutual Inductor

Models a mutual inductor. If winding 1 has voltage V1 across it and current I1 flowing into its + terminal, and winding 2 has voltage V2 across it and current I2 flowing into its + terminal, then

$$V1 = L1 \cdot dI1/dt + M \cdot dI2/dt$$

$$V2 = L2 \cdot dI2/dt + M \cdot dI1/dt$$

where parameters L1 and L2 are the winding self-inductances, and M is the mutual inductance. M is defined in terms of the Coefficient of coupling k by $M = k \cdot \sqrt{L1 \cdot L2}$. Hence k should be greater than zero and less than one.

[Source code](#)

Settings

ParametersVariables

Inductance L1:

1

mH

Inductance L2:

4

mH

Coefficient of coupling:

0.5

Block Parameters: Resistor1

Resistor

The voltage-current (V-I) relationship

The positive and negative terminals of the voltage across the resistor is given the device from the positive to the neg resistor is always positive.

[Source code](#)

Settings

ParametersVariables

Resistance:

2

kOhm

Block Parameters: Resistor

Resistor

The voltage-current (V-I) relationship for a linear resistor is $V = I \cdot R$, where R is the constant resistance in ohms.

The positive and negative terminals of the resistor are denoted by the + and - signs respectively. By convention, the voltage across the resistor is given by $V(+)-V(-)$, and the sign of the current is positive when flowing through the device from the positive to the negative terminal. This convention ensures that the power absorbed by a resistor is always positive.

[Source code](#)

Settings

ParametersVariables

Resistance:

1

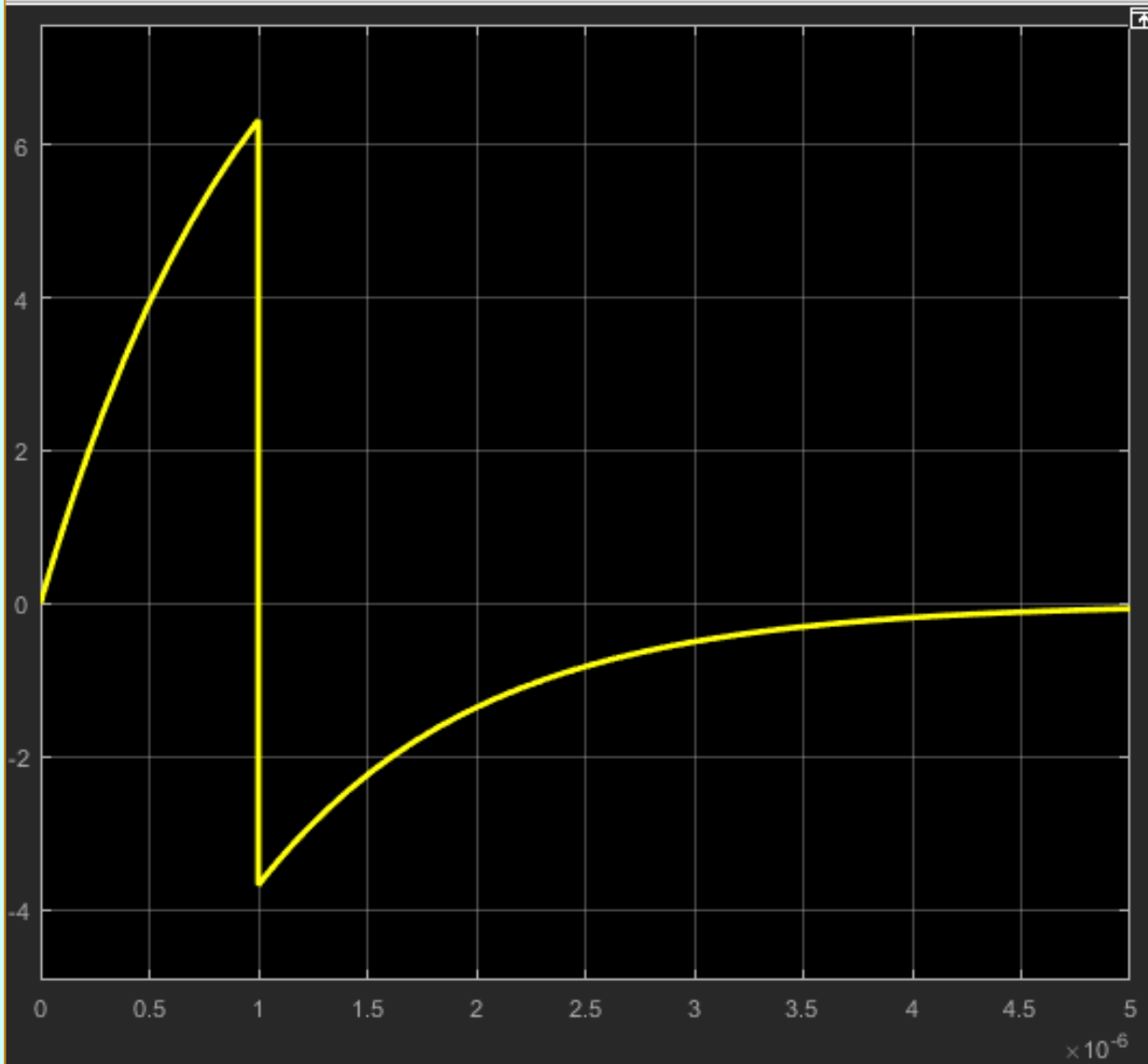
kOhm



Scope



File Tools View Simulation Help



Signal Statistics		
	Value	Time
Max	6.321e+00	1.000e-06
Min	-3.679e+00	1.000e-06
Peak to Peak	1.000e+01	
Mean	-1.700e-03	
Median	-2.586e-01	
RMS	2.367e+00	

Ready

Sample based T=5e-06

Једначине

Postavljanje jednacina

```
clear variables
syms Di1 L1 L2 L U L12 R R1 R2 i1 u1 u2 uR1 ug T t
jednacine = [ug == uR1 + u1, u1 == L1 * Di1, u2 == L12 * Di1, uR1 == R1 * i1]
```

$$\text{jednacine} = (u_g = u_1 + u_{R_1} \quad u_1 = D i_1 L_1 \quad u_2 = D i_1 L_{12} \quad u_{R_1} = R_1 i_1)$$

Jednacine stanja

```
jednacineIzvoda = eliminate(jednacine, [u1, u2, uR1])
```

$$\text{jednacineIzvoda} = [u_g - R_1 i_1 - D i_1 L_1]$$

```
assume([L1, L2, L12, U, R1, t], 'real')
assumeAlso(0 < R1 & 0 < L1 & 0 < L2 & 0 < L12 & 0 < L & 0 < U)
```

```
jednacineStanja = solve(jednacineIzvoda, Di1)
```

$$\text{jednacineStanja} = \frac{u_g - R_1 i_1}{L_1}$$

MATLAB: Symbolic Math Toolbox

Решавање диференцијалне једначине

Resavanje diferencijalne jednacine

```
syms i1(t) g(t)
promenljiveStanja = i1 == i1(t)
```

```
promenljiveStanja(t) = i1(t) = i1(t)
```

```
zamene = [R1 == R, R2 == 2*R, L1 == L, L2 == 4*L, L12 == (1/2)*sqrt(L*4*L)]
```

```
zamene = (R1 = R R2 = 2 R L1 = L L2 = 4 L L12 = L)
```

```
pobuda = ug == heaviside(t)
```

```
pobuda = ug = heaviside(t)
```

```
jednacineStanjaFun = subs(diff(i1(t)) == jednacineStanja,...
                           lhs([promenljiveStanja pobuda zamene]),...
                           rhs([promenljiveStanja pobuda zamene]))
```

```
jednacineStanjaFun =

$$\frac{\partial}{\partial t} i_1(t) = \frac{\text{heaviside}(t) - R i_1(t)}{L}$$

```

```
resenjeDiff = dsolve(jednacineStanjaFun, i1(0) == 0, 'IgnoreAnalyticConstraints', false)
```

```
resenjeDiff =

$$\frac{e^{-\frac{Rt}{L}} \text{heaviside}(t) \left( e^{\frac{Rt}{L}} - 1 \right)}{R}$$

```

MATLAB: Symbolic Math Toolbox

Гринова функција и конволуциони интеграл

Grinova funkcija

```
g_i1(t) = simplify(diff(resenjeDiff))
```

$$g_{i1}(t) = \frac{e^{-\frac{Rt}{L}} \text{heaviside}(t)}{L}$$

Konvolucioni integral

```
syms tau t
zameneLRT = [T == L/R, L12 == L]
```

$$\text{zameneLRT} = \left(T = \frac{L}{R} \quad L_{12} = L \right)$$

```
assume((0 < t) & (t < T))
i1_1(t) = simplify(int( (U*tau/T)*g_i1(t-tau), tau, 0, t))
```

$$i1_1(t) = \frac{U \left(Rt - L + L e^{-\frac{Rt}{L}} \right)}{R^2 T}$$

```
assume((T > 0) & (T < t))
i1_2(t) = simplify( int((U*tau/T)*g_i1(t-tau), tau, 0, T))
```

$$i1_2(t) = \frac{LU e^{-\frac{Rt}{L}}}{R^2 T} - \frac{U e^{\frac{R(T-t)}{L}} (L - RT)}{R^2 T}$$

MATLAB: Symbolic
Math Toolbox

Конволуциони интеграл (наставак)

```
u2_1(t) = simplify(subs(L12*diff(i1_1(t)), lhs(zameneLRT), rhs(zameneLRT)))
```

$$u2_1(t) = \\ U - U e^{-\frac{Rt}{L}}$$

```
u2_2(t) = simplify(subs(L12*diff(i1_2(t)), lhs(zameneLRT), rhs(zameneLRT)) )
```

$$u2_2(t) = \\ -U e^{-\frac{Rt}{L}}$$

```
% vrednosti = [R == 1000, L == 1e-3, T == 1e-6, U == 10]
```

```
assume(t,'clear')
```

```
u2(t) = (piecewise((t<0), 0, (0<=t)&(t<T), u2_1(t), (t>=T), u2_2(t) ))
```

$$u2(t) = \begin{cases} 0 & \text{if } t < 0 \\ U - U e^{-\frac{Rt}{L}} & \text{if } t < T \wedge 0 \leq t \\ -U e^{-\frac{Rt}{L}} & \text{if } T \leq t \end{cases}$$

MATLAB: Symbolic Math Toolbox

График функције одзива

Crtanje grafika

```
vrednosti = [R == 1000, L == 1e-3, T == 1e-6, U == 10]
```

```
vrednosti =
```

```
 $\left(R = 1000 \quad L = \frac{1}{1000} \quad T = \frac{1}{1000000} \quad U = 10\right)$ 
```

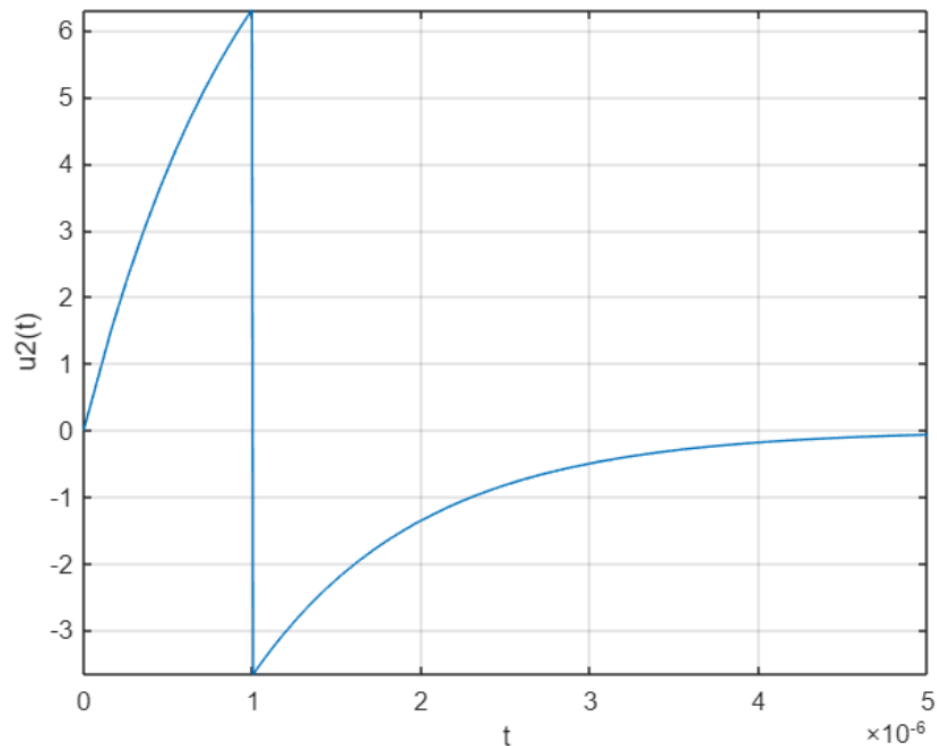
```
fplot(t, subs(u2(t), lhs(vrednosti), rhs(vrednosti)), [0 5e-6] )
```

```
grid on
```

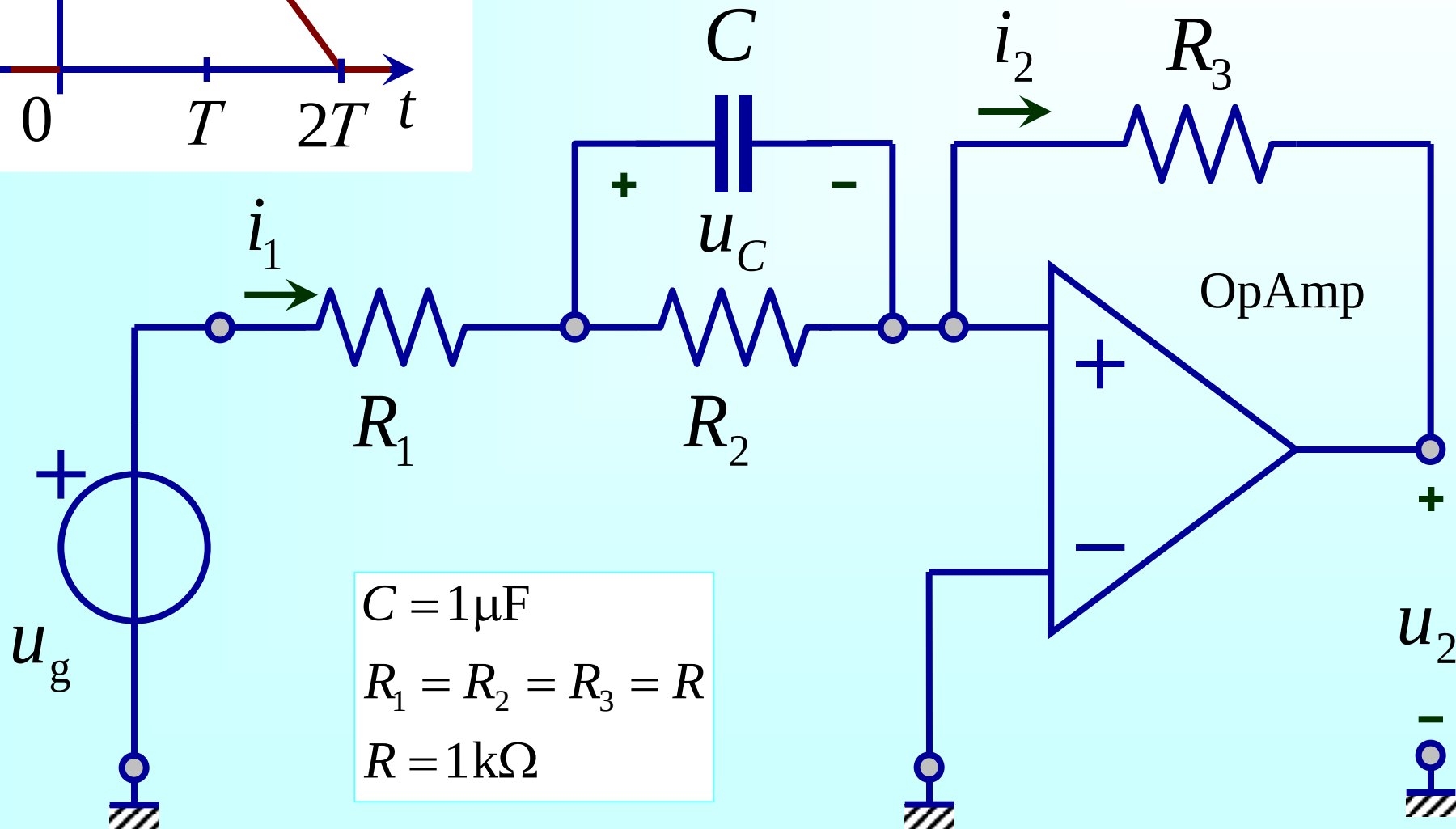
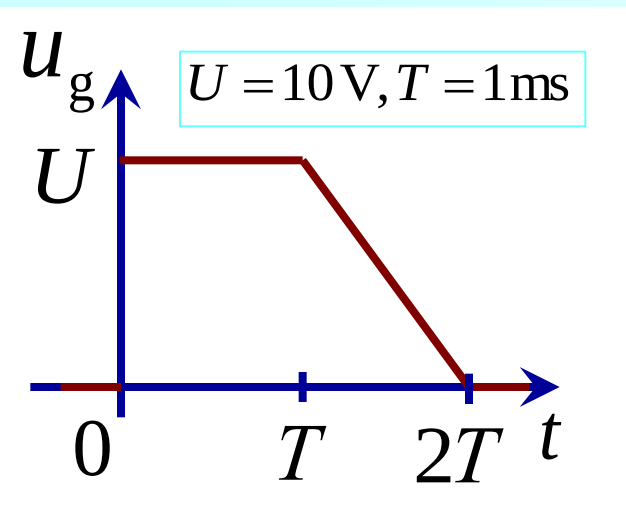
```
xlabel('t')
```

```
ylabel('u2(t)')
```

MATLAB: Symbolic
Math Toolbox



Одзив на сложену побуду

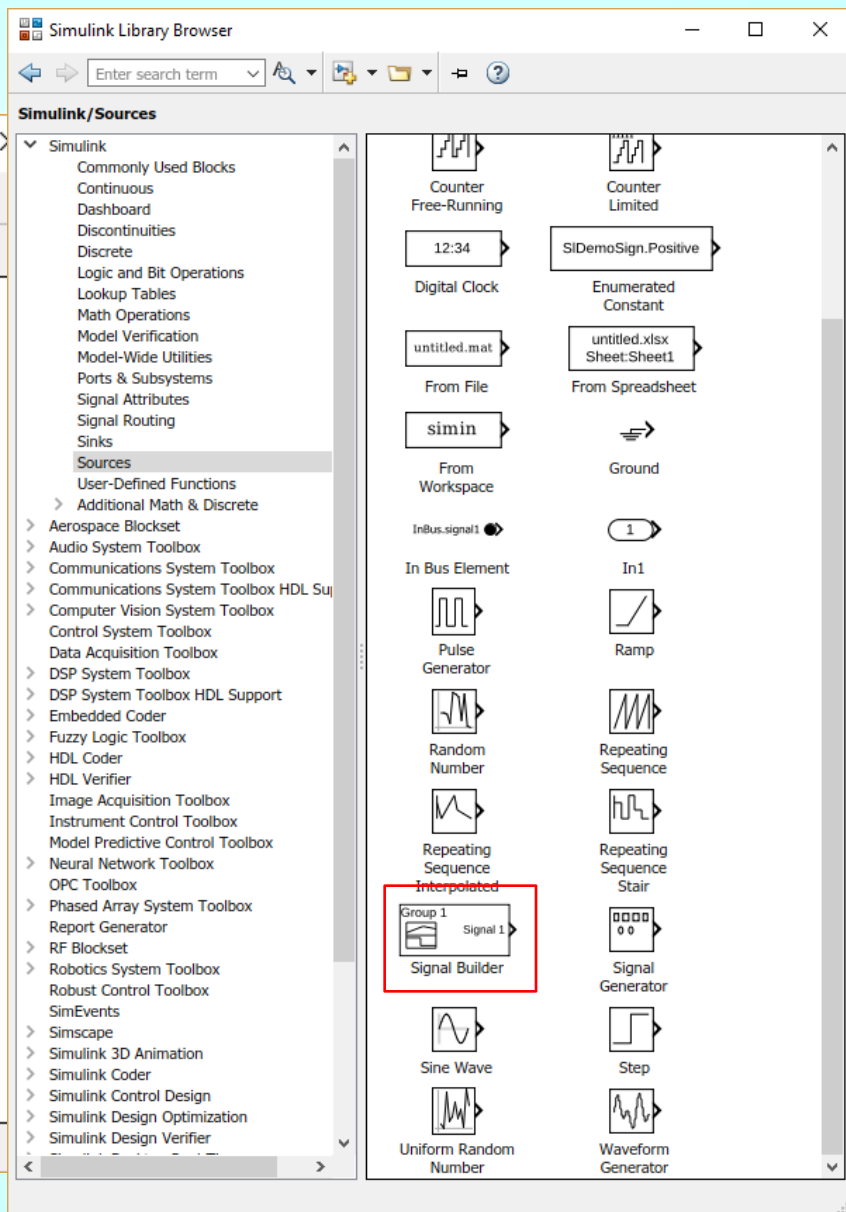
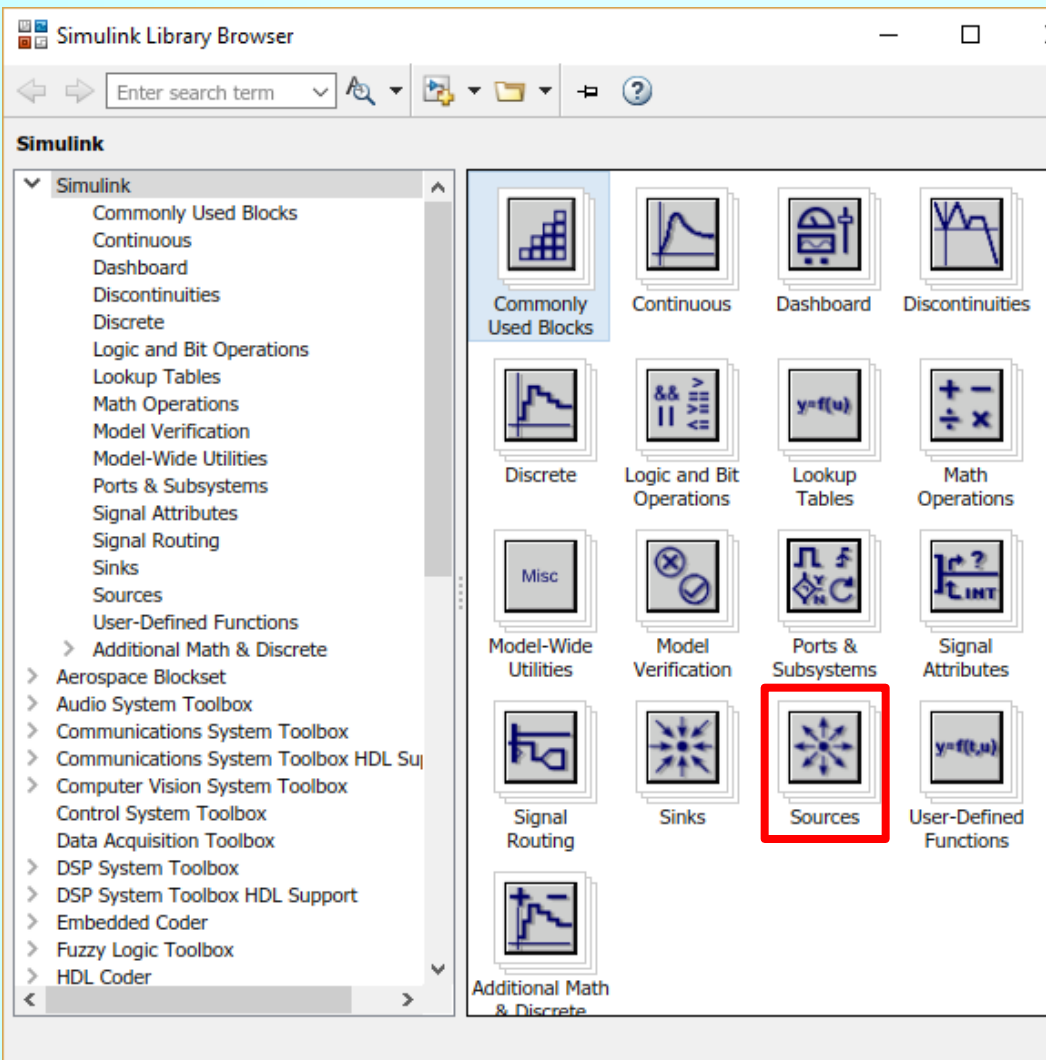


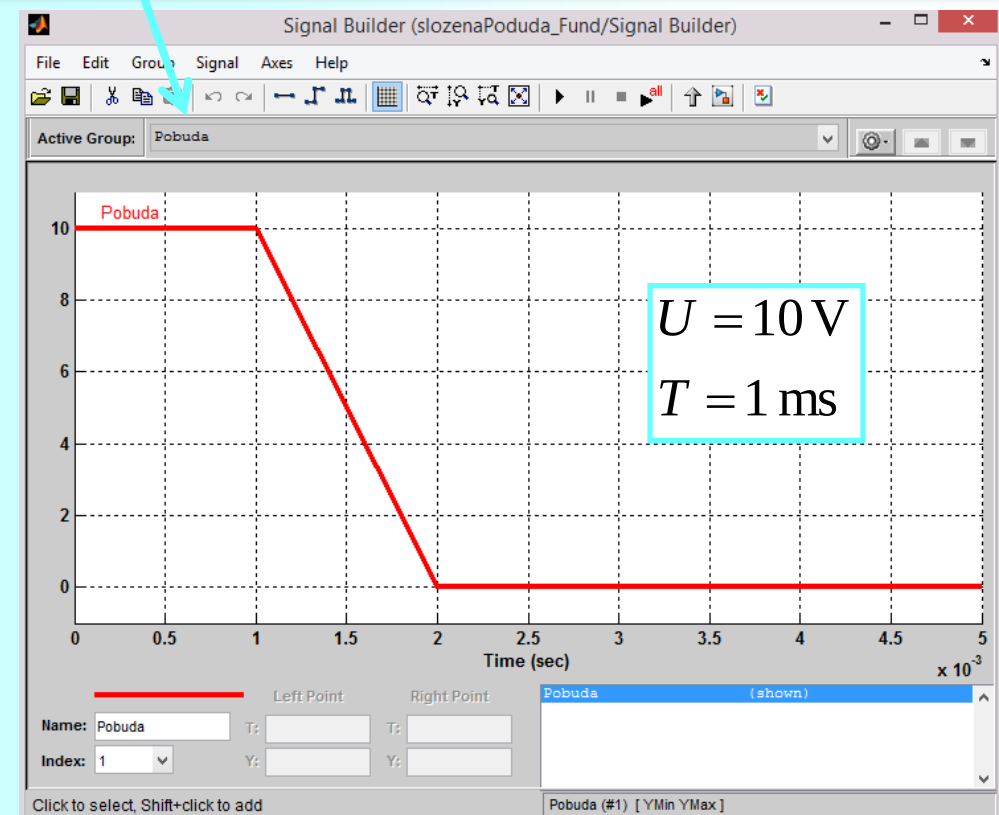
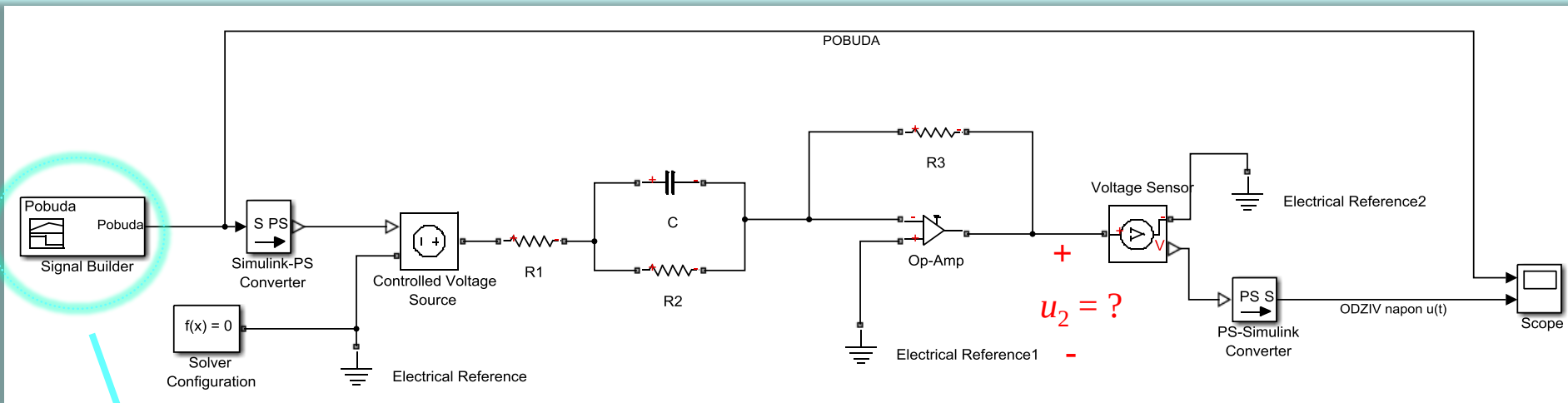
Задатак - сложена побуда

- Параметри кола са слике су познати:
 $R_1 = R_2 = R_3 = R$. Нема сакупљене енергије.
- Применом конволуционог интеграла одредити напон кондензатора $u_C(t)$, а затим и излазни напон $u_2(t)$ и нацртати његов график.
- Напон генератора се мења са временом као на дијаграму, при чему је $T = RC$.

MATLAB: Simulink

Sources

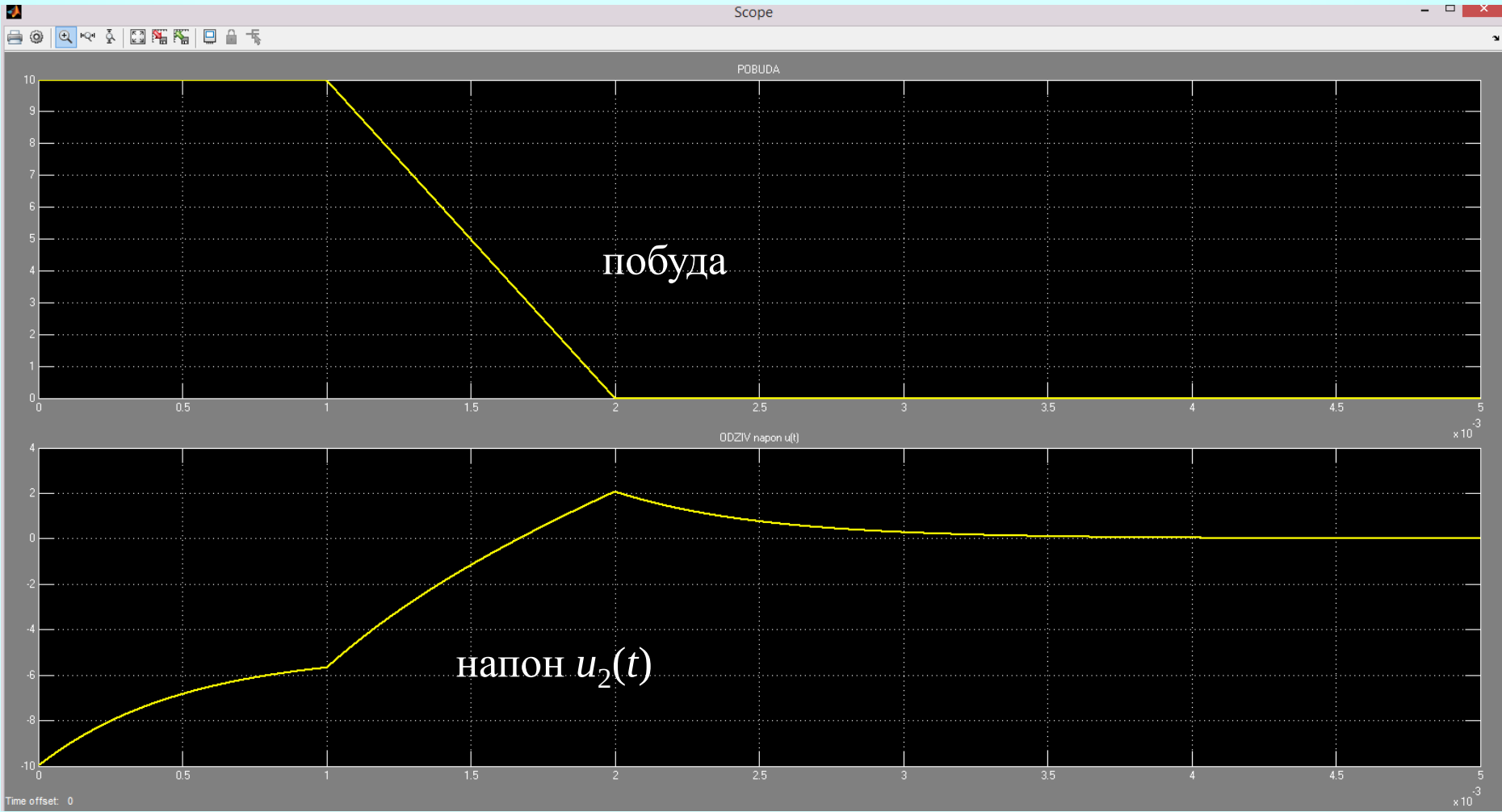




$u_2 = ?$

$R_1 = R_2 = R_3 = 1\text{ k}\Omega$
 $C_1 = 1\text{ }\mu\text{F}$

MATLAB: Simscape
Foundation Library
Utilities
Simulink



Једначине

```
clear variables
syms C R R1 R2 R3 T U t
assume(0 < R1 & 0 < R2 & 0 < R3 & 0 < R & 0 < C & 0 < U & 0 < T)
```

```
syms DuC i1 i2 iC iR2 u uC uR1 uR2 ug
jednacine = [ug == uR1 + uC,...
    uR1 == R1 * i1,...
    iC == C * DuC,...
    uR2 == R2 * iR2,...
    uR2 == uC,...
    i1 == iR2 + iC,...
    i2 == i1,...
    u == -(R3 * i2)]
```

$$\text{jednacine} = \left(\begin{array}{l} ug = uC + uR_1 \quad uR_1 = R_1 i_1 \quad iC = C DuC \quad uR_2 = R_2 iR_2 \quad uR_2 = uC \quad i_1 = iC + iR_2 \quad i_2 = i_1 \quad u = -R_3 i_2 \end{array} \right)$$

```
jednacineIzvoda = eliminate(jednacine, [i1, iC, uR1, iR2, uR2, i2, u])
```

$$\text{jednacineIzvoda} = [R_1 uC + R_2 uC - R_2 ug + C DuC \quad R_1 R_2]$$

```
jednacineStanje = solve(jednacineIzvoda, DuC)
```

$$\text{jednacineStanje} = -\frac{R_1 uC + R_2 uC - R_2 ug}{C R_1 R_2}$$

Решавање диференцијалне једначине

Rešavanje diferencijalne jednačine

```
syms uC(t)
PromenljiveStanjaIzvodi = [uC == uC(t), DuC == diff(uC(t), t)]
```

```
PromenljiveStanjaIzvodi(t) =  
(uC(t) = uC(t) DuC =  $\frac{\partial}{\partial t}$  uC(t))
```

```
zamene = [R1 == R, R2 == R, R3 == R, T == R * C]
```

```
zamene = (R1 = R R2 = R R3 = R T = C R)
```

```
pobuda = ug == heaviside(t)
```

```
pobuda = ug = heaviside(t)
```

```
JS = subs(jednacineStanje, lhs([PromenljiveStanjaIzvodi pobuda zamene]), rhs([PromenljiveStanjaIzvodi pobuda zamene]))
```

```
JS =  

$$\frac{R \operatorname{heaviside}(t) - 2 R uC(t)}{C R^2}$$

```

```
f_uC = dsolve(diff(uC(t)) == JS, uC(0)==0, 'IgnoreAnalyticConstraints', false)
```

```
f_uC =  

$$\frac{e^{-\frac{2t}{CR}} \operatorname{heaviside}(t) \left( e^{\frac{2t}{CR}} - 1 \right)}{2}$$

```

Гринова функција и побуда

Grinova funkcija

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

$$g_{uC}(t) = \frac{e^{-\frac{2t}{CR}} \text{heaviside}(t)}{CR}$$

Generatori

```
syms U
ug_1(t) = U;
ug_2(t) = -U*(t-2*T)/T;
ug_3(t) = sym(0);
ug_tot(t) = piecewise((0<t<=T),ug_1(t), (T<t)&(t<=2*T),ug_2(t), (2*T<t),ug_3(t) )
```

$$ug_{tot}(t) = \begin{cases} U & \text{if } t \leq T \wedge 0 < t \\ \frac{U(2T-t)}{T} & \text{if } T < t \wedge t \leq 2T \\ 0 & \text{if } 2T < t \end{cases}$$

Конволуциони интеграл (1)

Konvolucioni integral

```
syms tau
syms uC_1(t)
assume((0 < t)&(t<=T));
uC_1(t) = int(ug_1(tau) * g_uC(t-tau), tau, 0, t)
```

$$u_{C_1}(t) = -\frac{U}{2} \left(e^{-\frac{2t}{CR}} - 1 \right)$$

```
syms i2_1(t) i1_1(t)
i1_1(t) = subs(1/R * (ug_1(t) - uC_1(t)), lhs(zamene), rhs(zamene));
i2_1(t) = i1_1(t);
u2_1(t) = simplify( -(R * i2_1(t)) )
```

$$u_{2_1}(t) = -\frac{U}{2} - \frac{U e^{-\frac{2t}{CR}}}{2}$$

Конволуциони интеграл (2)

```
syms uC_2(t)
assume((0 < T)&(T < t) & (t <= 2*T));
uC_2(t) = int(ug_1(tau) * g_uC(t - tau), tau, sym(0), T) + int(ug_2(tau) * g_uC(t - tau), tau, T, t)
```

$$u_{C_2}(t) = \frac{U e^{-\frac{2t}{CR}} \left(e^{\frac{2T}{CR}} - 1 \right)}{2} - \frac{U \left(2 e^{\frac{2T-2t}{CR}} - 4 \right)}{4} - \frac{U \left(2t - CR + CR e^{\frac{2T-2t}{CR}} \right)}{4T}$$

```
simplify(subs(uC_2(t), lhs(zamene), rhs(zamene)))
```

$$\text{ans} = -\frac{2Ut - 5CRU + CRU e^{2 - \frac{2t}{CR}} + 2CRU e^{-\frac{2t}{CR}}}{4CR}$$

```
i1_2(t) = subs(1/R * (ug_2(t) - uC_2(t)), lhs(zamene), rhs(zamene));
i2_2(t) = i1_2(t);
u2_2(t) = simplify( -(R * i2_2(t)) )
```

$$u_{2_2}(t) = -\frac{3CRU - 2Ut + CRU e^{2 - \frac{2t}{CR}} + 2CRU e^{-\frac{2t}{CR}}}{4CR}$$

Конволуциони интеграл (3)

```
syms uC_3(t)
assume((2*T < t) & (0 < T));
uC_3(t) = int(ug_1(tau) * g_uC(t - tau), tau, 0, T) + int(ug_2(tau) * g_uC(t - tau), tau, T, 2*T)
```

$$u_{C_3}(t) = \frac{U e^{-\frac{2t}{CR}} \left(e^{\frac{2T}{CR}} - 1 \right)}{2} - \frac{U e^{\frac{2T}{CR}} e^{-\frac{2t}{CR}} \left(2T + CR - CR e^{\frac{2T}{CR}} \right)}{4T}$$

```
i1_3(t) = subs(1/R * (ug_3(t) - uC_3(t)), lhs(zamene), rhs(zamene));
i2_3(t) = i1_3(t);
syms u2_3(t)
u2_3(t) = simplify( -(R * i2_3(t)) )
```

$$u_{2_3}(t) = \frac{U e^{-\frac{2t}{CR}} (e^2 - 1)}{2} - \frac{U e^2 e^{-\frac{2t}{CR}} (3CR - CR e^2)}{4CR}$$

```
simplify(u2_3(t))
```

$$\text{ans} = \frac{U e^{-\frac{2t}{CR}} (e^2 - 1)}{2} - \frac{U e^2 e^{-\frac{2t}{CR}} (3CR - CR e^2)}{4CR}$$

Вредности

```
vrednosti = [R == 1e3, C == 1e-6, U == 10, T == 1e-3]
```

```
vrednosti =
```

$$\left(R = 1000 \quad C = \frac{1}{1000000} \quad U = 10 \quad T = \frac{1}{1000} \right)$$

```
u2(t) = simplify(subs(piecewise((0 < t) <= T, u2_1(t),...
                                (T < t)&(t<=2*T), u2_2(t),...
                                (2*T < t), u2_3(t)), lhs(vrednosti), rhs(vrednosti)))
```

```
u2(t) =
```

$$\begin{cases} -\sigma_1 - 5 & \text{if } t \in \left(0, \frac{1}{1000}\right] \\ 5000t - \sigma_1 - \frac{5e^{2-2000t}}{2} - \frac{15}{2} & \text{if } t \in \left(\frac{1}{1000}, \frac{1}{500}\right] \\ 5e^{-2000t}(e^2 - 1) + 2500e^{-2000t}e^2 \left(\frac{e^2}{1000} - \frac{3}{1000}\right) & \text{if } \frac{1}{500} < t \end{cases}$$

where

$$\sigma_1 = 5e^{-2000t}$$

```
ug(t) = subs(ug_tot(t), lhs(vrednosti), rhs(vrednosti))
```

```
ug(t) =
```

$$\begin{cases} 10 & \text{if } t \in \left(0, \frac{1}{1000}\right] \\ 20 - 10000t & \text{if } t \in \left(\frac{1}{1000}, \frac{1}{500}\right] \\ 0 & \text{if } \frac{1}{500} < t \end{cases}$$

Цртање графика одзива и побуде

Crtanje grafika

```
figure
fplot( t, u2(t), [0 5e-3])
hold on
fplot( t, ug(t), [0 5e-3])
hold off
grid on
xlabel('t')
ylabel('u2(t)')
legend({'u2(t)', 'ug(t)'})
```

