

СИМУЛАЦИЯ НА ДИНАМИЧНИ СИСТЕМИ СЪС SIMULINK

dimitrova@tu-sofia.bg
pct.tu-sofia.bg/dd/pik3



Simulink

● Интерактивна програмна среда, интегерирана с MatLab

● Предназначение

- моделиране, симулиране и анализ на динамични системи
- изследване на линейни и нелинейни системи
- изследване на непрекъснати и дискретни процеси
- Model-Based Design

● Приложения

- технически системи във всички инженерни области
- изследователска лаборатория

● Йерархични модели

- top-down
- bottom-up



Компоненти на средата

● Средства за моделиране

- графичен редактор за построяване на блокови диаграми
- библиотеки от блокове

● Средства за симулация

- посредством меню
- от командния прозорец на MatLab
- настройка на параметри
- **what-if** изследване

● Средства за анализ

- команди на MatLab
- **application toolboxes**



Процес на моделиране

● Етапи

- Дефиниране на системата и подсистемите
- Идентифициране на компонентите на системата
- Аналитично описание чрез формули
- Построяване на блокови диаграми
- Произвеждане на симулация
- Валидиране на резултатите от симулацията



Компоненти на моделираната система

● Типове

■ Параметри

- признаци на системата, чиито стойности остават постоянни величини, ако не бъдат променени за целите на изследването

■ Състояния

- признаци на системата, чиито стойности се променят във времето

■ Сигнали

- входни и изходни параметри, чиито стойности се изменят динамично по време на симулацията

● Представяне на компонентите

- **блок**: параметри и състояния
- **свързващи линии**: сигнали



Идентифициране на компонентите

● В отговор на въпросите:

- **Кои са входните сигнали?**
- **Кои са изходните сигнали?**
- **Кои са (променливите) състояния на системата?**
- **Кои са (константните) параметри на системата?**
- **Има ли междинни (вътрешни) сигнали?**



Работна среда на Simulink

The image displays the Simulink environment. The **Simulink Library Browser** window is open, showing a tree view of libraries on the left. The **Math Operations** library is selected, and the **Sum** block is highlighted in the main pane. A search bar at the top right is labeled **Block Search**. Below the library browser, the **Block Description** pane shows the details for the **Sum** block: "Add or subtract inputs. Specify one of the following: a) string containing + or - for each input port. l for spacer between ports. b) scalar, >= 1, specifies the number of input ports to be summed. When there is only one input port, add or subtract elements over all dimension".

The **simple_model** window shows a Simulink model with the following components: a **Sine Wave** block, an **Integrator** block (labeled $1/s$), a **Sum** block, and a **Scope** block. The Sine Wave block is connected to both the Sum block and the Integrator block. The output of the Integrator block is connected to the Sum block. The output of the Sum block is connected to the Scope block. The Scope block is set to a time range of 20 seconds.



Стандартни библиотеки от блокове

Библиотека	Съдържание
Commonly Used Blocks	Извадка от най-често използваните блокове – вход, изход, константа, дисплей и др. Тези блокове се намират и в други библиотеки
Continuous	Блокове за моделиране на линейни функции – диференциатор, интегратор и др.
Discontinuities	Блокове, чиито изходи са непрекъснати функции на входовете си- Saturation block
Discrete	Блокове, представящи дискретни функции на времето - Unit Delay block
Logic and Bit Operations	Логически и побитови операции - Logical
LookUp Tables	Блокове, които използват таблици за определяне на изхода от входа – Cosine, Sine blocks
Math Operations	Представяне на математически и логически функции – сума, произведение и др.
Model Verification	Блокове за авто-валидиране на модела - Check Input, Resolution block

Стандартни библиотеки от блокове

Библиотека	Съдържание
Model-Wide Utilities	Метаданни за модела - Model Info block
Ports & Subsystems	За създаване на подсистеми - In1, Out1, Subsystem
Signal Attributes	За модифициране на атрибутите на сигнали - Data Type Conversion
Signal Routing	За рутиране на сигнали от една точка на диаграмата до друга –Mux, Switch
Sinks	За наблюдаване на изхода - Out1, Scope
Sources	За генериране или импортиране на вход - Constant, In1, Sine Wave
User-Defined Functions	За дефиниране на потребителски функции - Embedded MATLAB™ Function
Additional Math & Discrete	Две допълнителни библиотеки за блокове за математически и дискретни функции

http://www.mathworks.com/products/demos/simulink/Simulink_Key_Features/videos/Library.html



Библиотеки от демонстрационни примери

The screenshot shows a web browser window displaying the 'Simscape Demos' help page. On the left, a 'Help Navigator' sidebar is visible, listing various Simulink and Simscape categories. The main content area features a title 'Simscape Demos', a descriptive paragraph about Simscape's capabilities, and a link to the product page. Below this, a section titled 'Mechanical Systems' lists three demo models with associated thumbnail images and 'Model' buttons.

Help Navigator

Search for: Go

Example: ["plot tools" OR plot* tools](#)

Contents Index Search Results Demos

- Simulink
 - General Applications
 - Automotive Applications
 - Aerospace Applications
 - Modeling Features
 - SimDriveline
 - SimEvents
 - SimHydraulics
 - SimMechanics
 - SimPowerSystems
 - Simscape**
 - Mechanical Systems
 - Electrical Systems
 - Shunt Motor
 - Permanent Magnet DC Motor
 - Nonlinear Bipolar Transistor
 - Small-Signal Bipolar Transistor
 - Differentiator
 - Band-Limited Op-Amp
 - Finite-Gain Op-Amp
 - Inverting Op-Amp Circuit
 - Noninverting Op-Amp Circuit
 - Solenoid
 - Full-Wave Bridge Rectifier

Simscape Demos

height

Simscape extends Simulink with tools for modeling and simulating multidomain physical systems, such as those with mechanical, hydraulic, and electrical components. Simscape can be used for a variety of automotive, aerospace, defense, and industrial-equipment applications. Together with SimMechanics, SimDriveline, SimHydraulics, and SimPowerSystems (all available separately), Simscape lets you model complex interactions in multidomain physical systems.

[Product page at mathworks.com](#)

Mechanical Systems

	Simple Mechanical System	Model
	Mechanical System with Translational Friction	Model
	Mechanical System with Translational Hard Stop	Model

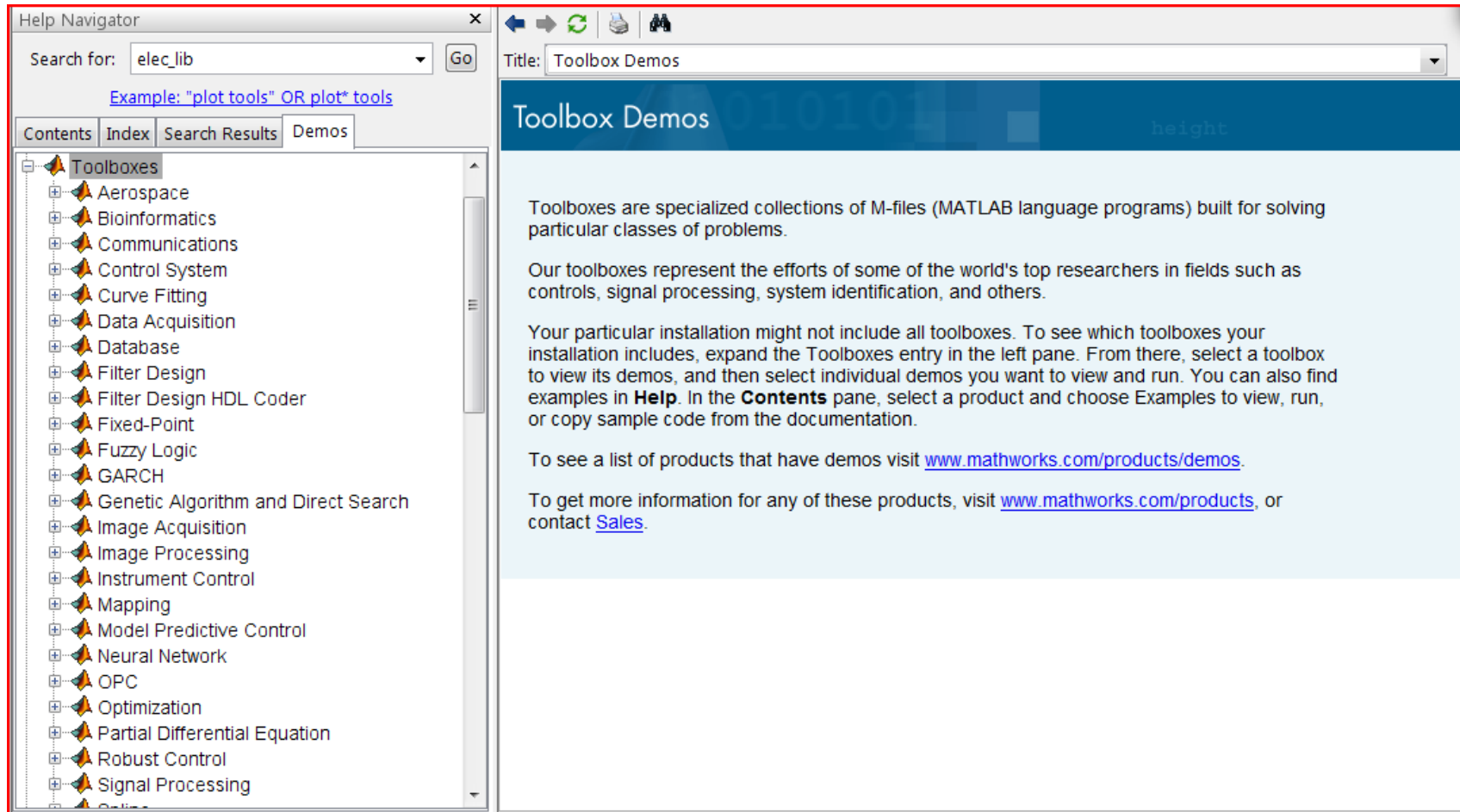


● Toolbox

- специализирани колекции от m-files (MATLAB language programs)
- създадени за решаване на определен клас задачи
- съдържат алгоритми, функции, графики и потребителски интерфейси
- за изследване, проектиране, анализиране и симулиране на различни приложни технически системи



Библиотеки от допълнителни инструменти



Help Navigator

Search for:

Example: "plot tools" OR plot* tools

Contents Index Search Results Demos

- Toolboxes
 - Aerospace
 - Bioinformatics
 - Communications
 - Control System
 - Curve Fitting
 - Data Acquisition
 - Database
 - Filter Design
 - Filter Design HDL Coder
 - Fixed-Point
 - Fuzzy Logic
 - GARCH
 - Genetic Algorithm and Direct Search
 - Image Acquisition
 - Image Processing
 - Instrument Control
 - Mapping
 - Model Predictive Control
 - Neural Network
 - OPC
 - Optimization
 - Partial Differential Equation
 - Robust Control
 - Signal Processing
 - ...

Title: Toolbox Demos

Toolbox Demos

height

Toolboxes are specialized collections of M-files (MATLAB language programs) built for solving particular classes of problems.

Our toolboxes represent the efforts of some of the world's top researchers in fields such as controls, signal processing, system identification, and others.

Your particular installation might not include all toolboxes. To see which toolboxes your installation includes, expand the Toolboxes entry in the left pane. From there, select a toolbox to view its demos, and then select individual demos you want to view and run. You can also find examples in **Help**. In the **Contents** pane, select a product and choose Examples to view, run, or copy sample code from the documentation.

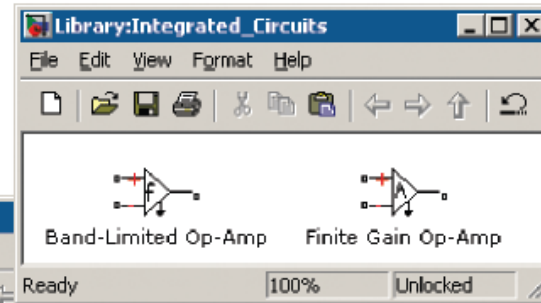
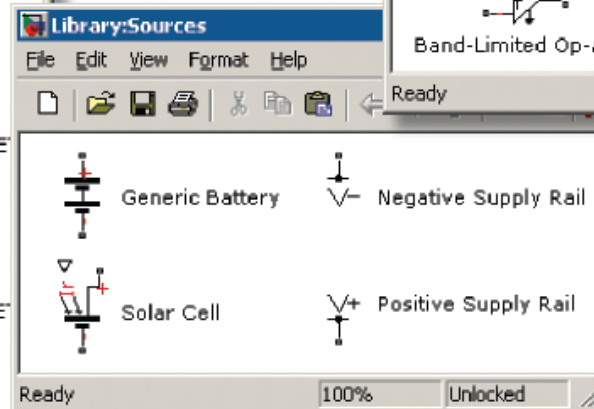
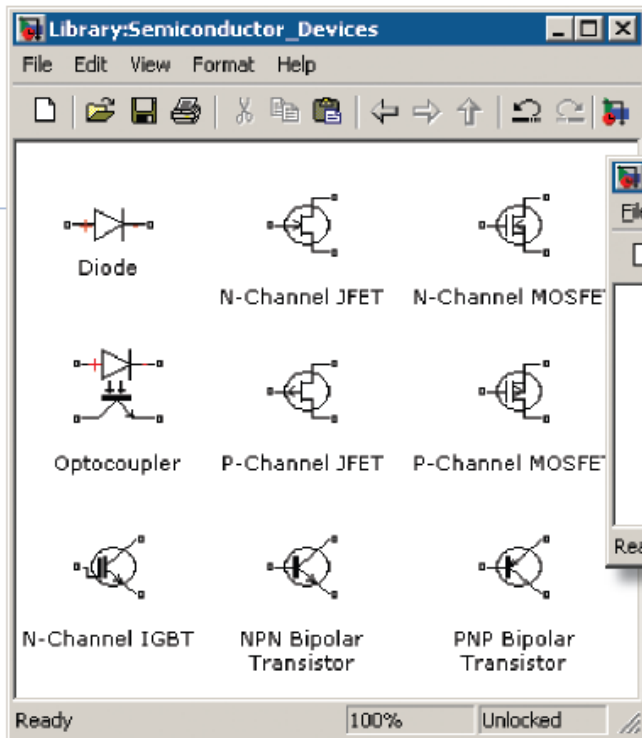
To see a list of products that have demos visit www.mathworks.com/products/demos.

To get more information for any of these products, visit www.mathworks.com/products, or contact [Sales](#).



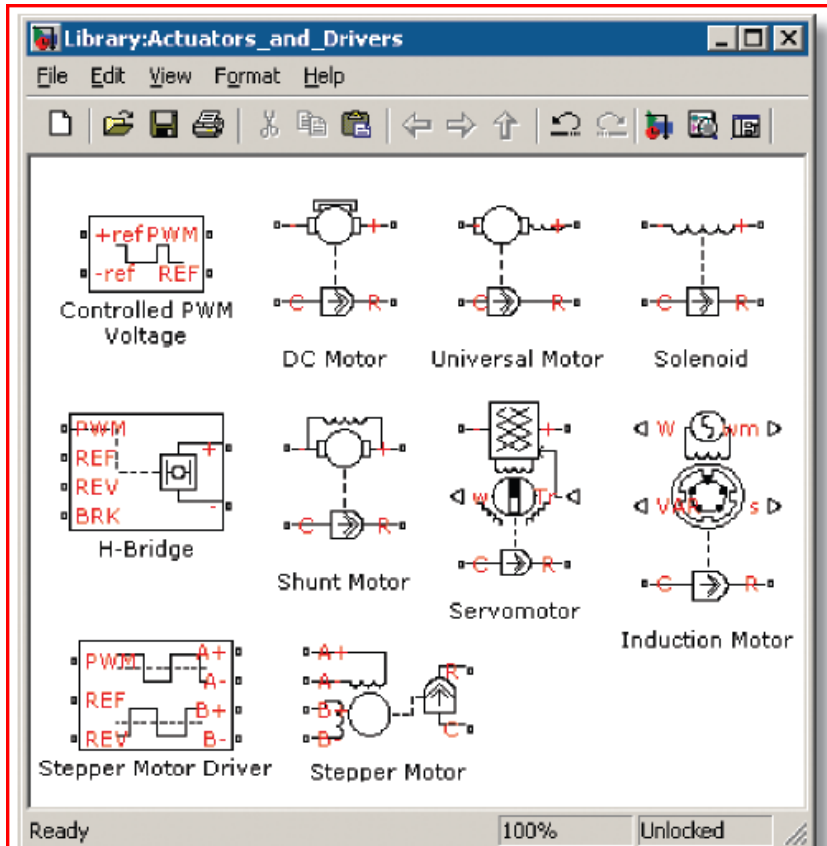
- Моделиране на електронни и електромеханични системи
- Разширение на Simscare
- Библиотеки от блокове за
 - сензори
 - полупроводникови елементи
 - електродвигатели
 - логически схеми
 - интегрални схеми
 - компоненти, съвместими със SPICE
 - и др.



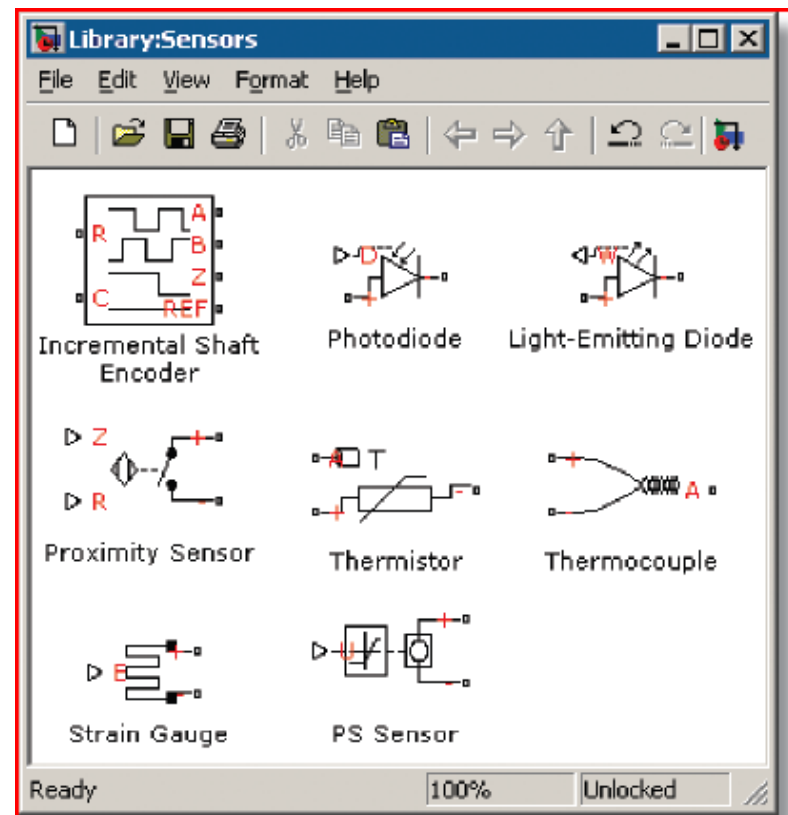


SimElectronics™ libraries of semiconductors (left), sources (center), and integrated circuits (right) for modeling electronic systems.





SimElectronics™ library of actuators and drivers for testing different actuation solutions.



SimElectronics™ library of sensors for capturing non-ideal behavior of sensor hardware.



Пример 1

Решаване на уравнението

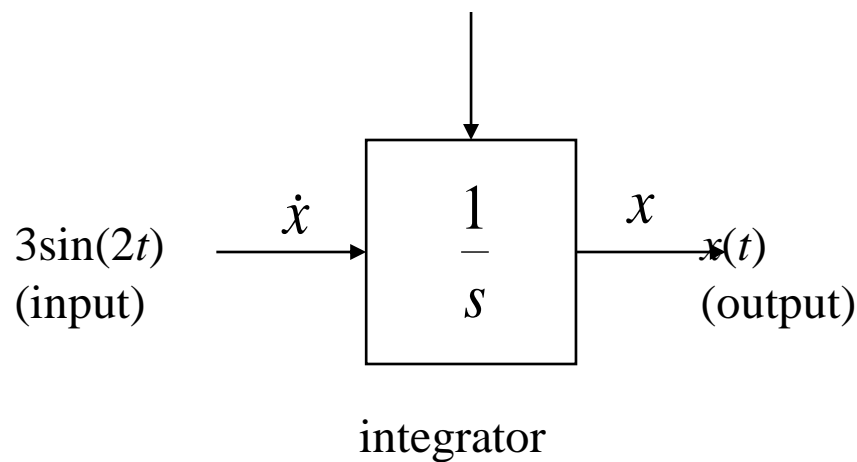
$$\dot{x} = 3\sin(2t)$$

■ начални условия:

$$x(0) = -1.$$

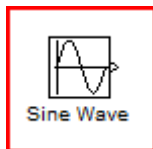
■ ВХОД: $3\sin(2t)$

■ ИЗХОД: $x(t)$



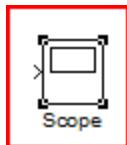
Избор на блокове

● Вход:



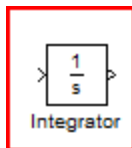
от Sources

● Изход:



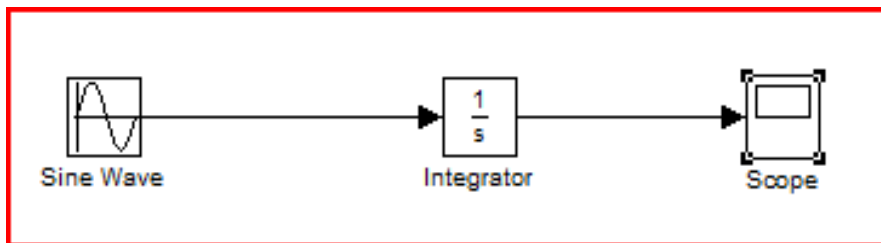
от Sinks

● Оператор:



от Continuous

● Свързване на блоковете със сигнали:



Настройка на параметри

$$3 \sin(2t)$$

$$x(0) = -1$$

Function Block Parameters: Integrator

Integrator
Continuous-time integration of the input signal.

Parameters

External reset: none

Initial condition source: internal

Initial condition:
-1

Limit output

Upper saturation limit:
inf

Lower saturation limit:
-inf

Show saturation port

Show state port

Absolute tolerance:
auto

Ignore limit and reset when linearizing

Enable zero crossing detection

State Name: (e.g., 'position')
"

OK Cancel Help Apply

Source Block Parameters: Sine Wave

Sine Wave

Output a sine wave:

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

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

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

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

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:
3

Bias:
0

Frequency (rad/sec):
2

Phase (rad):
0

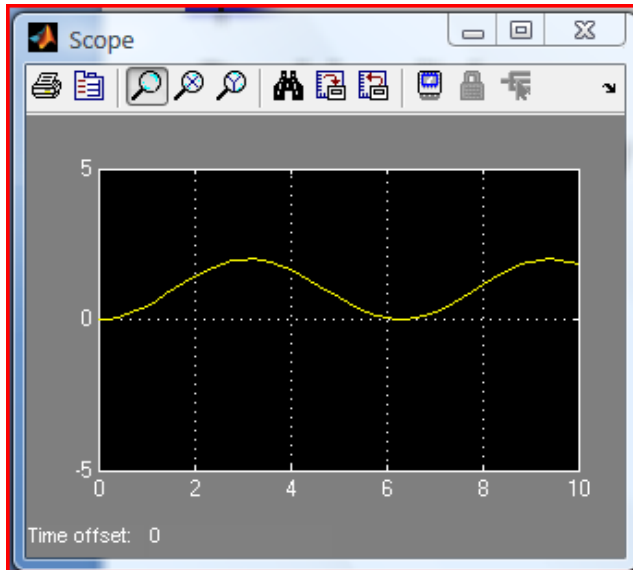
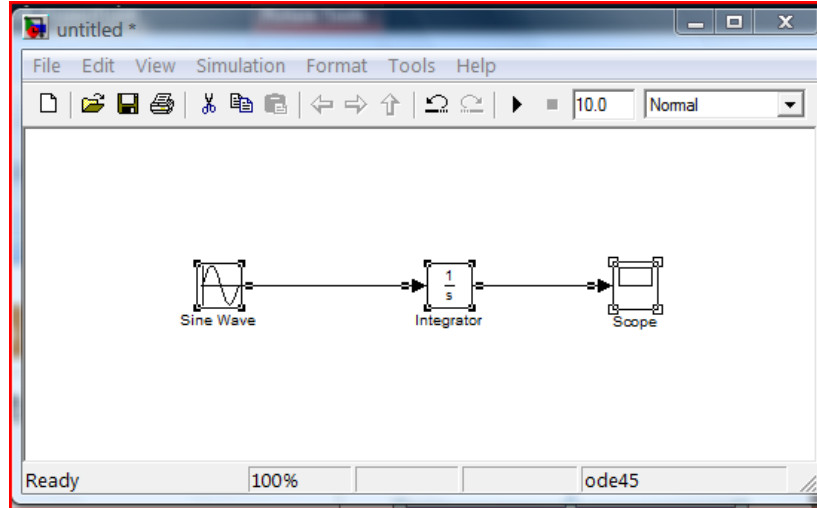
Sample time:
0

Interpret vector parameters as 1-D

OK Cancel Help



Симуляция



Проверка: аналитично решение

$$x(t) = \frac{1}{2} - \frac{3}{2} \cos(2t)$$

matches the plot (the simulation result) exactly.



Пример 2

Решаване на линейното
ОДУ от 1-ред:

$$u(t) = i(t)R + L \frac{di(t)}{dt}$$

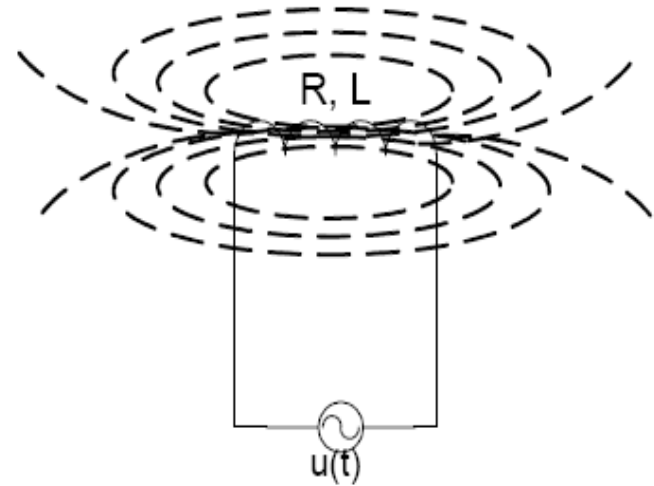
$$u(t) - R i(t) = L \frac{di(t)}{dt}$$

начални условия:

$$u(t) = 0$$

първа производна:

$$\frac{di(t)}{dt} = \frac{1}{L}(u(t) - i(t)R)$$



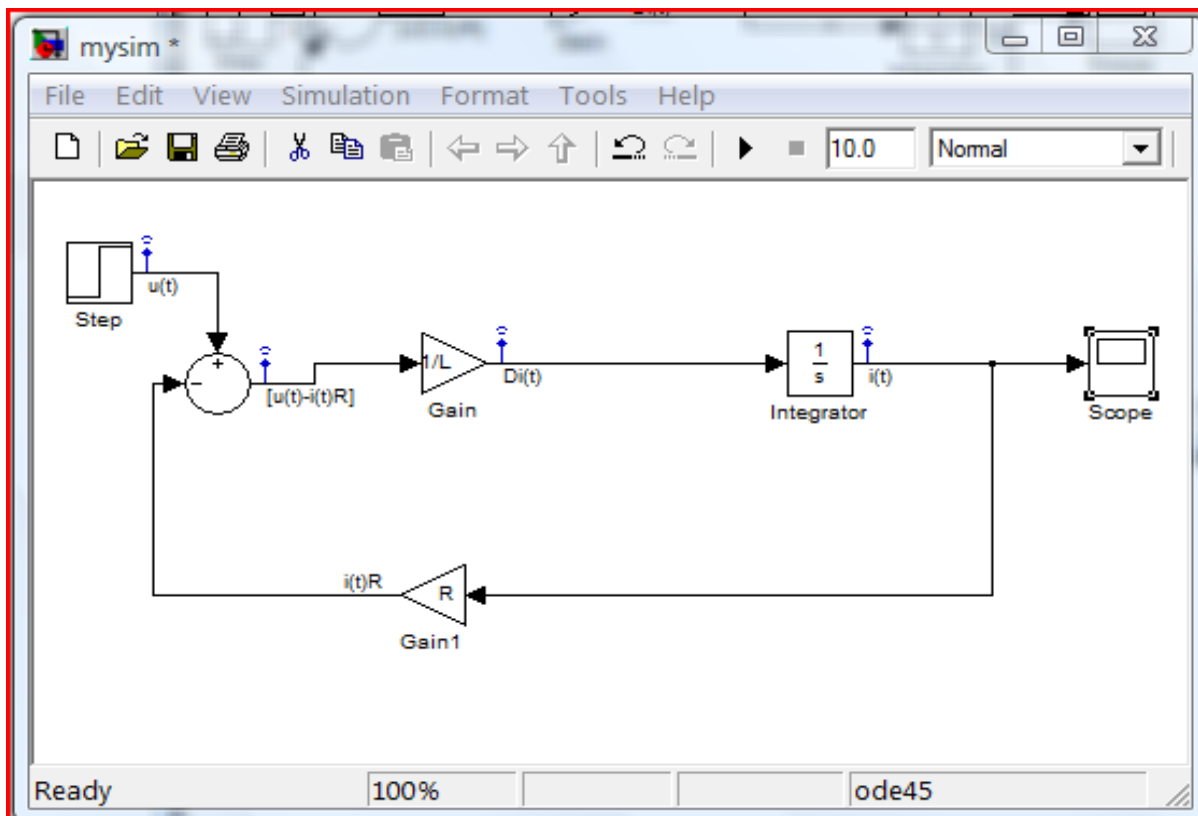
$$u(t) = i(t)R + \frac{d\psi(t)}{dt}$$

L – константа

R - константа



Построяване на блокова диаграма

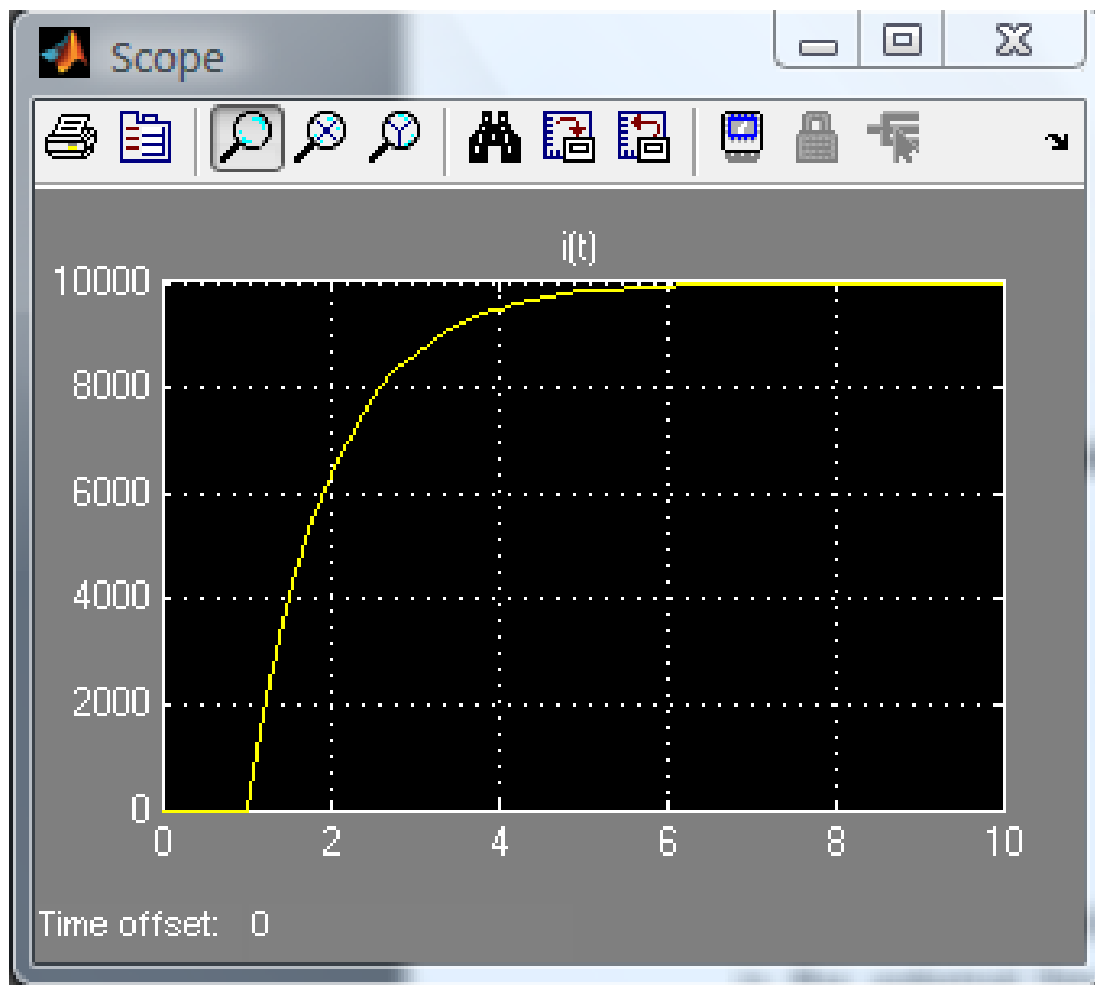


интегратор – от Continuous
умножител Gain – от Math
осцилоскоп – от Sinks
константи: $R=0.01$; $L=0.01$

входен сигнал – симулира промяна в
напрежението – от Sources
стъпка: амплитуда = 1



Решение



Пример 3

Пружина

$$\Sigma F = ma$$

$$F(t) - Kx(t) - Bu(t) = ma(t)$$

където

F – външна сила,
приложена върху масата (m);

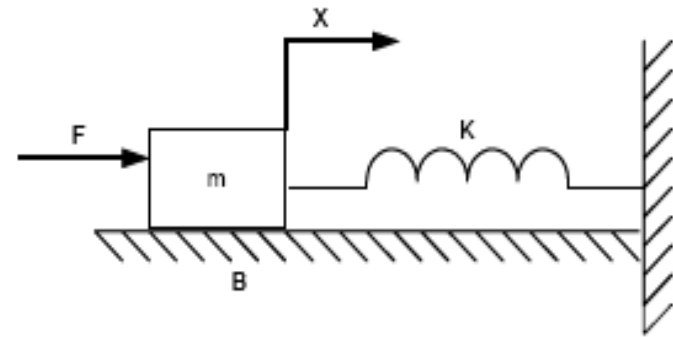
K – константа на
пружината;

a – ускорение на
масата;

u – скоростта на
масата;

x – разстоянието на
преместване;

B – фактор на
триенето.



Втори закон на Нютон:

$$\Sigma \vec{F} = m\vec{a}$$

$$F(t) - K \frac{dx(t)}{dt} - B = m \frac{d^2 x(t)}{dt^2}$$



Параметри:

$$m = 0.25$$

$$K = 0.5$$

$$B = 1$$

Вход:

$f(t)$ стъпка с
магнитуда = 3

$$F(t) - K \frac{dx(t)}{dt} - B = m \frac{d^2x(t)}{dt}$$

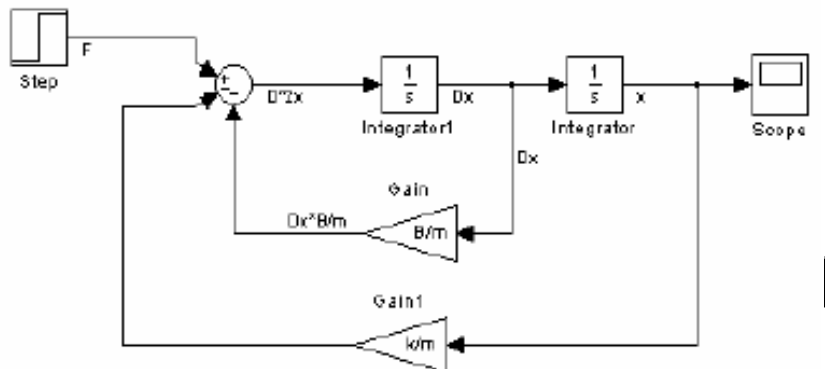
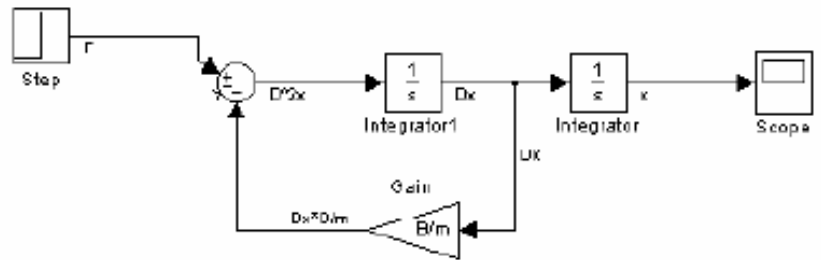
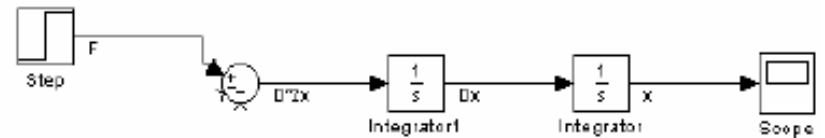
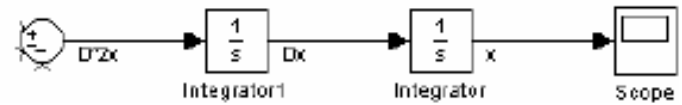
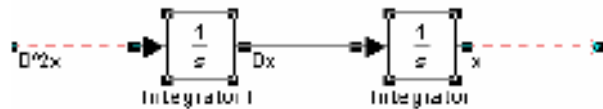
$$\frac{d^2x(t)}{dt} = \frac{1}{m} \left(F(t) - K \frac{dx(t)}{dt} - B \right)$$

Диференциране на
стъпки

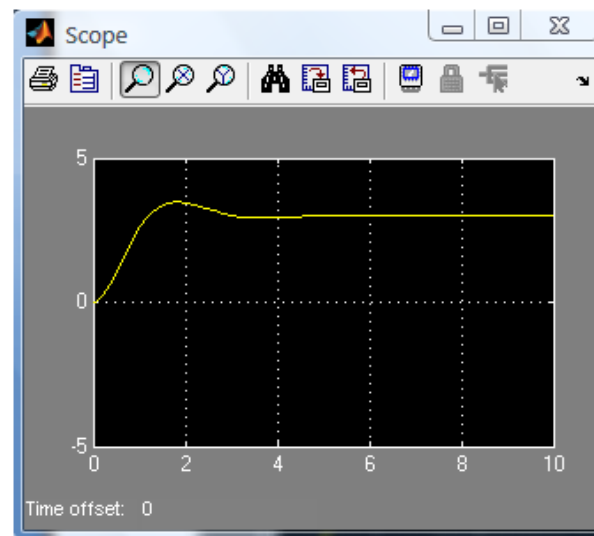
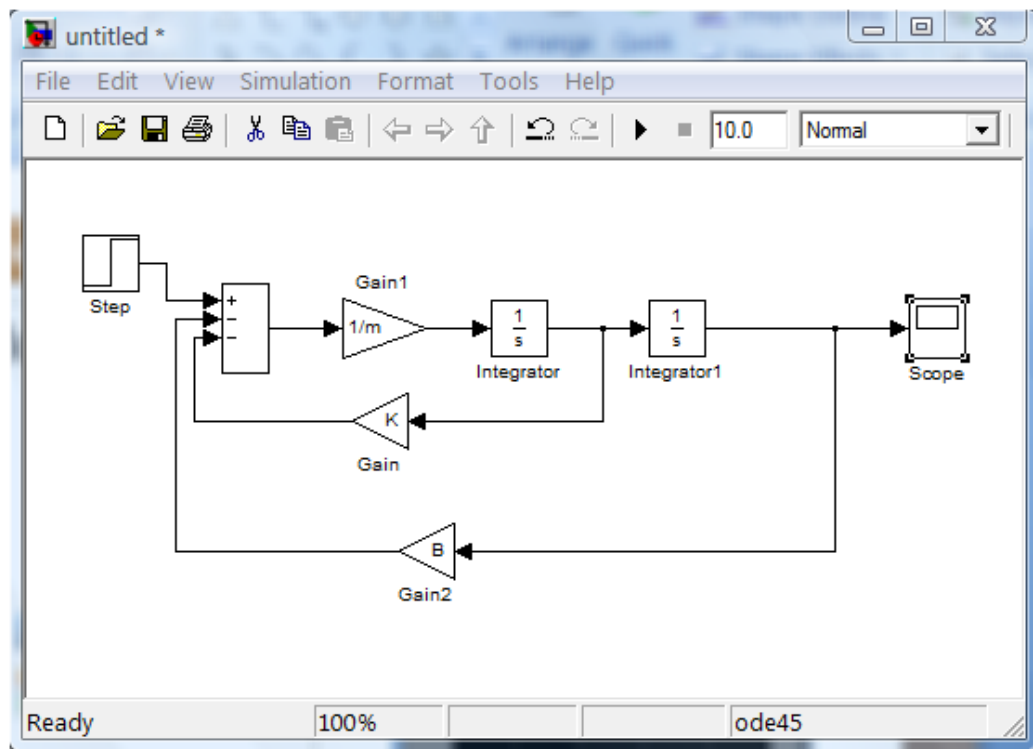


СТЪПКИ

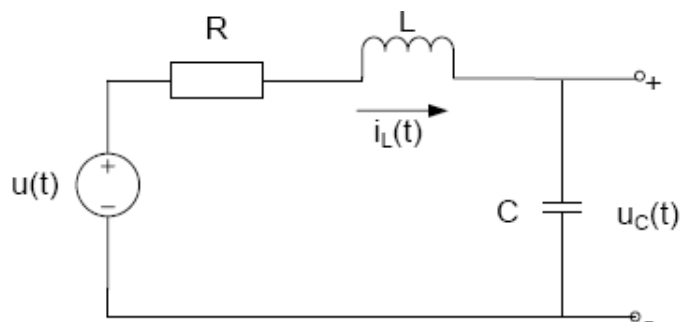
● Два интегратора – по един за всяка степен



Результат



Пример 4



$$u_R(t) + u_L(t) + u_C(t) = u(t)$$

$$u_R(t) = Ri_L(t)$$

$$u_L(t) = L \frac{di_L(t)}{dt}$$

$$u_C(t) = \frac{1}{C} \int_{-\infty}^t i_L(\tau) d\tau$$

$$\frac{du_C(t)}{dt} = \frac{i_L(t)}{C}$$

$$RC \frac{du_C(t)}{dt} + L \frac{di_L(t)}{dt} + u_C(t) = u(t)$$

$$y_1(t) = i_L(t)$$

$$y_2(t) = u_C(t)$$

$$y_1(t) = C \frac{dy_2(t)}{dt}$$

$$Ry_1(t) + L \frac{dy_1(t)}{dt} + y_2(t) = u(t)$$

$$\frac{dy_1(t)}{dt} = \frac{1}{L} u(t) - \frac{R}{L} y_1(t) - \frac{1}{L} y_2(t)$$

$$\frac{dy_2(t)}{dt} = \frac{1}{C} y_1(t)$$

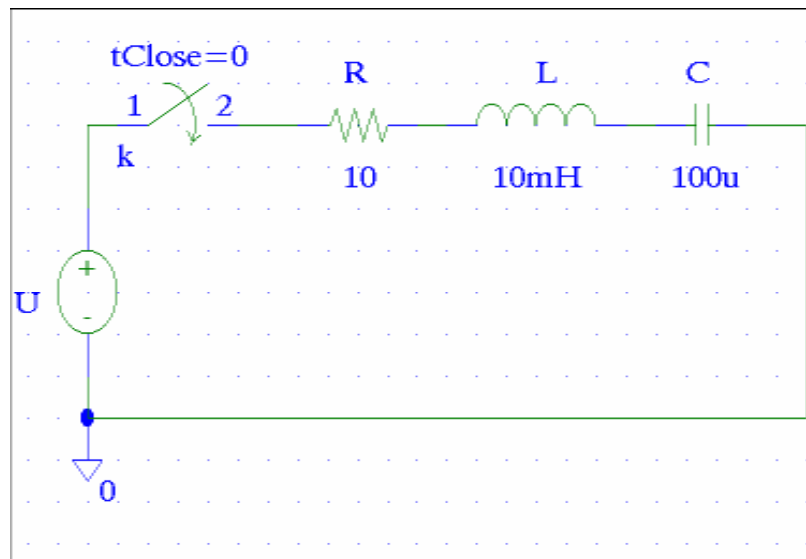
$$y_1(t=0) = 0$$

$$y_2(t) = 0$$



Пример

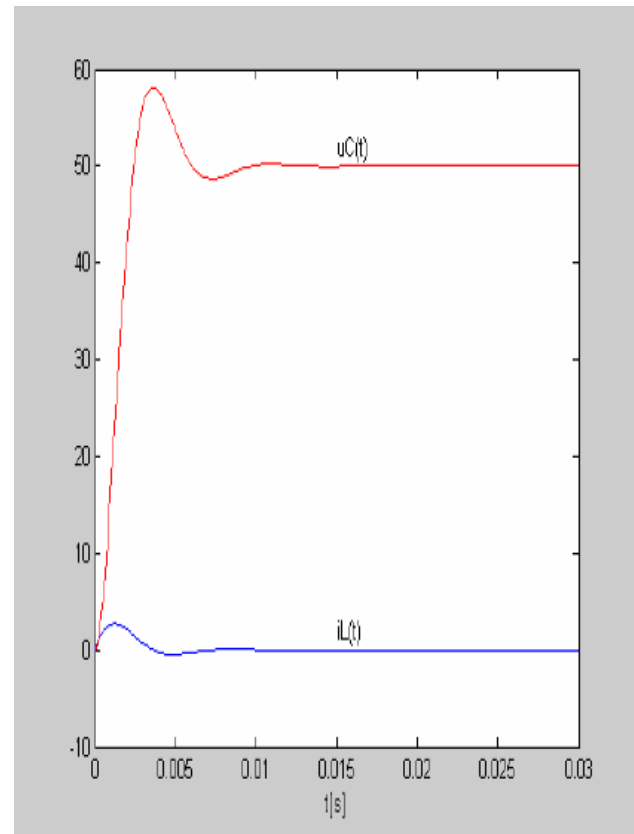
- В момента $t=0$ ключът се затваря и RLC веригата се свързва към източник на напрежение.
- Да се изчисли напрежението върху кондензатора, ако $R=10\Omega$, $L=10\text{mH}$, $C=100\mu\text{F}$ и е приложен постоянен източник на напрежение $U=50\text{V}$.



```

function dy=RLC(t,y,options,u,R,L,C)
dy=zeros(2,1);
a=1/L;b=R/L;c=1/C;
dy(1)=a*u-b*y(1)-a*y(2);
dy(2)=c*y(1);
% RLC верига
U=50;R=10;L=10e-3;C=100e-6;
t0=0;           % начално време
tfinal=30e-3;   % крайно време
iL0=0; % начално условие  $i_L(t=0)=0$ 
uC0=0; % начално условие  $u_C(t=0)=0$ 
[t,x]=ode45('RLC',[t0 tfinal],[iL0
uC0],[],U,R,L,C); % числено решение
plot(t,x(:,1),'b',t,x(:,2),'r')
axis([0 30e-3 -10 60]),xlabel('t[s]')
text(0.015, 2, 'iL(t)'),text(0.015, 52, 'uC(t)')

```



Други приложения на MatLab и Simulink

● Графичен потребителски интерфейс (видео)

- MatLab Creating ...

● Демонстрации

- (Simulink general applications
- MatLab other demos)
- температурен модел на къща
- търговски пътник
- обиколка около света
- minesweeper

