Innovative Lifelong e-Learning for Professional Engineers (e-ProfEng)

586391-EPP-1-2017-1-SE-EPPKA2-CBHE-JP

Training in Electrical Engineering Discipline

Modelling and Simulation in Electrical Engineering

Teaching Materials for Topic 1

Modelling and Simulation of Engineering Systems

Author:

Mirko Köhler, P9 FERIT

Content

Part 1. 3

The purpose of modelling and simulation in engineering 8

Possible software solutions and their capabilities 20

Part 2. 33

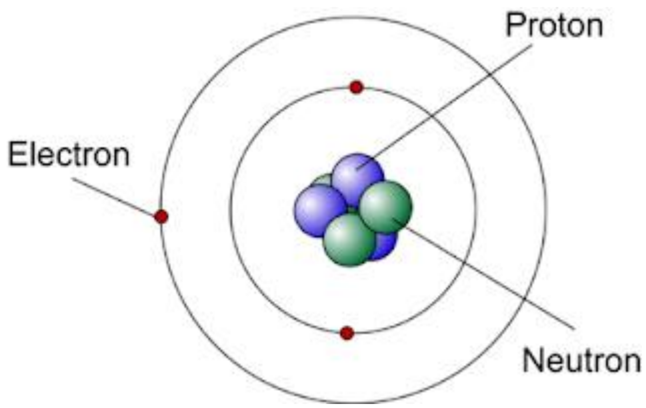
Solving tasks 34

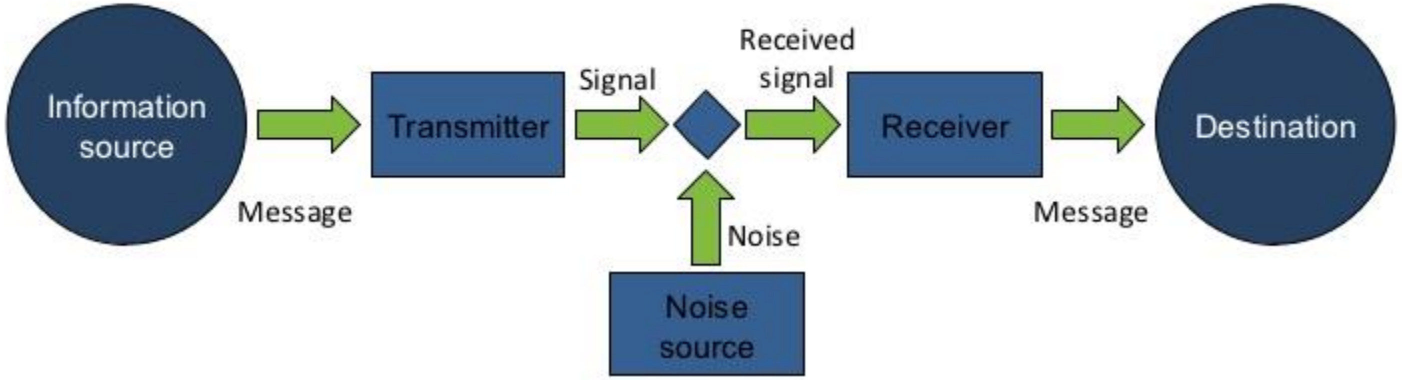
Modeling and simulation in Matlab/Simulink 55

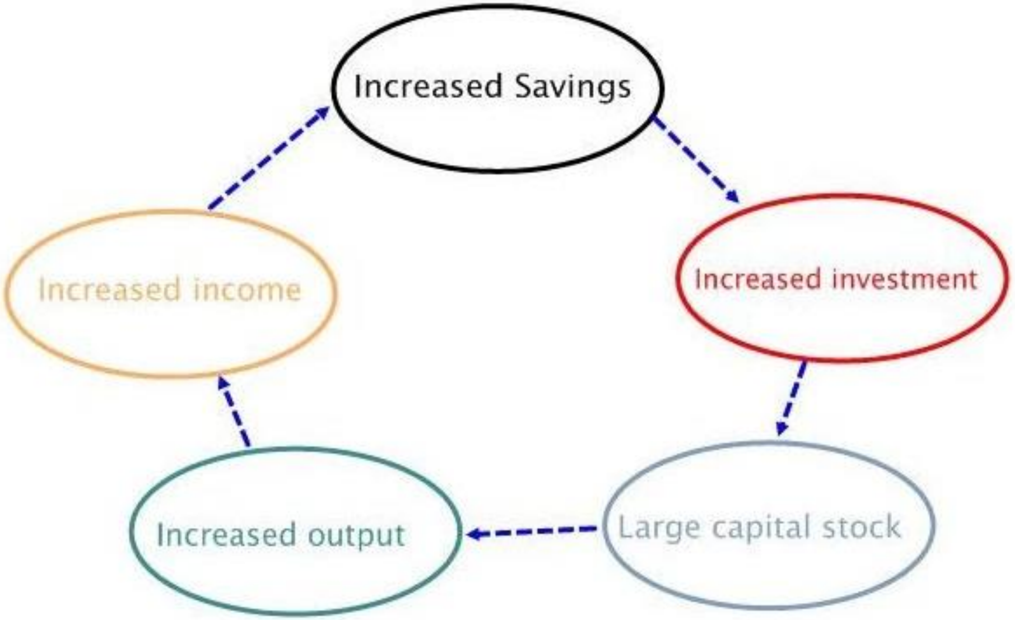
Part 1.

What is a model?

Is this a model?







Model is …

- a thing used as an example to follow or imitate
- fashion or shape (a three-dimensional figure or object) in a malleable material such as clay or wax
- not the real world but merely a human construct to help us better understand real world systems
- graphical, mathematical (symbolic), physical, or verbal representation or simplified version of a concept, phenomenon, relationship, structure, system, or an aspect of the real world
- an aid to communication between those involved in the project, especially between developer and the user
- …

The purpose of modelling and simulation in engineering

John von Neumann (1903-1957)

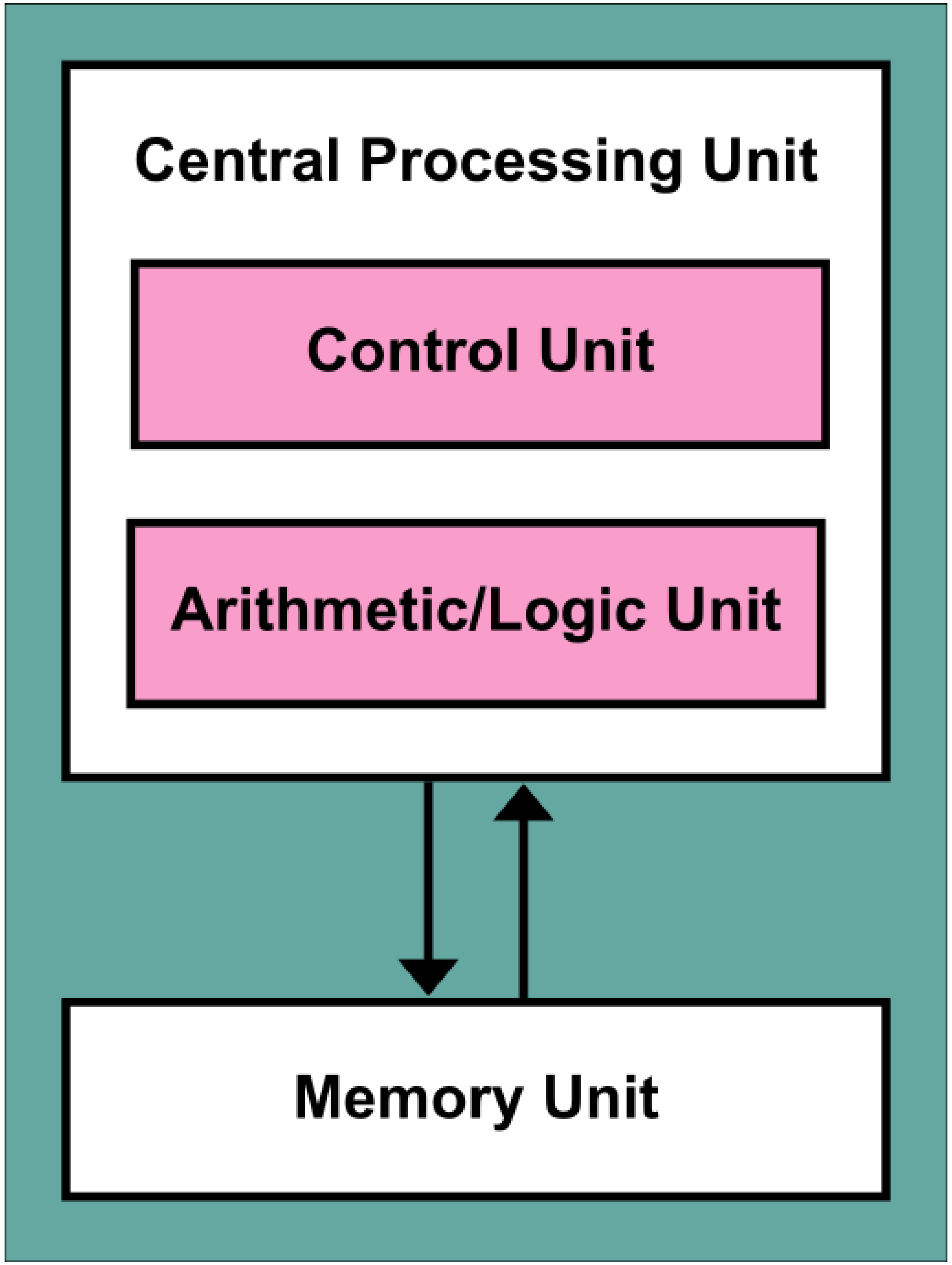
The sciences do not try to explain, they hardly even try to interpret, they mainly make models.

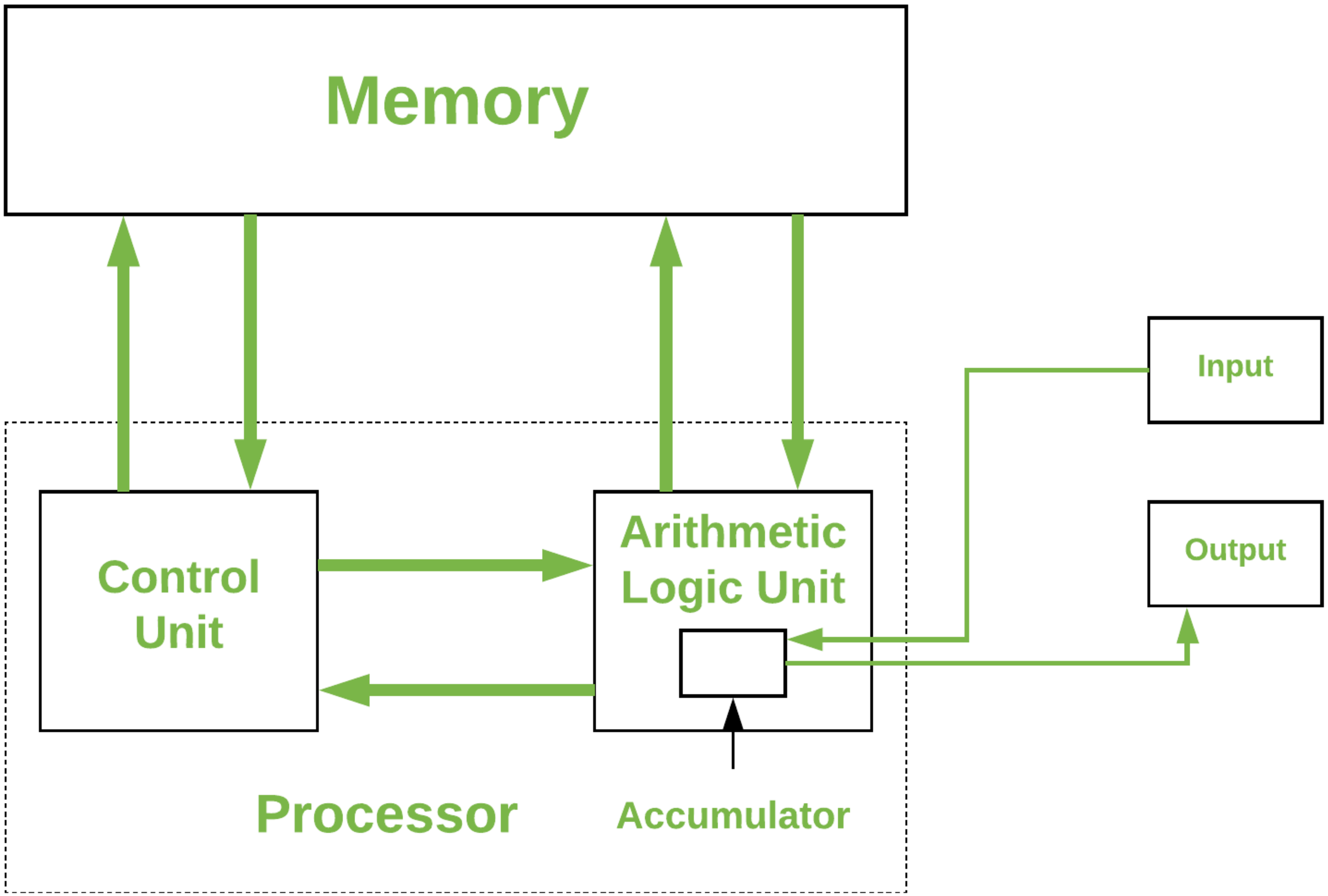
By a model is meant a mathematical construct which, with the addition of certain verbal interpretations, describes observed phenomena.

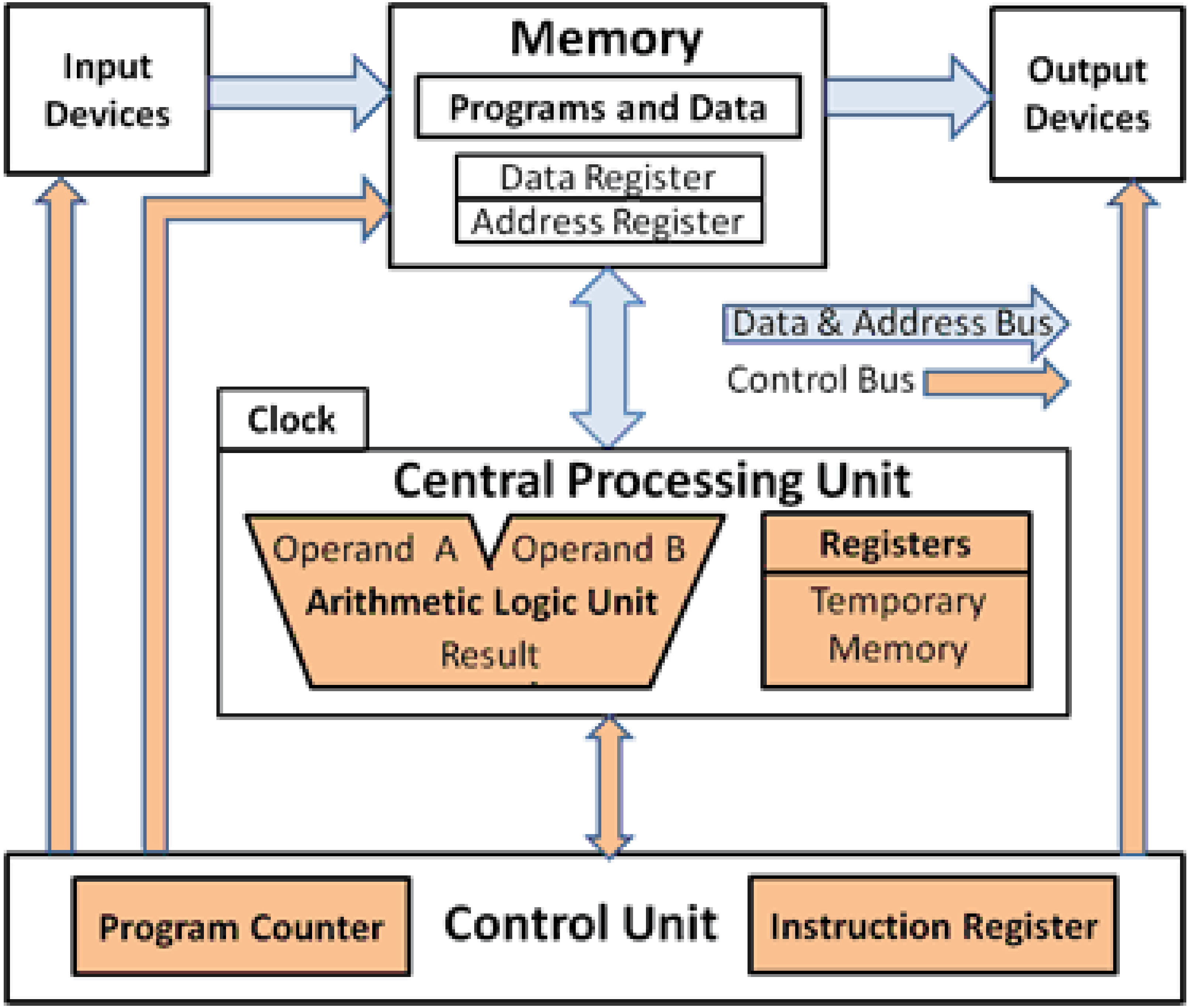
The justification of such a mathematical construct is solely and precisely that it is expected to work - that is correctly to describe phenomena from a reasonably wide area.

Furthermore, it must satisfy certain esthetic criteria - that is, in relation to how much it describes, it must be rather simple.

What is this?


Central Processing Unit
Control Unit
Arithmetic/Logic Unit
Memory Unit


Memory
Input
Control Unit
Arithmetic Logic Unit
Output
Processor
Accumulator


Input Devices
Memory
Programs and Data
Data Register
Address Register
Output Devices
Data & Address Bus
Control Bus
Clock
Central Processing Unit
Operand A
Operand B
Arithmetic Logic Unit
Result
Registers
Temporary Memory
Program Counter
Control Unit
Instruction Register

- An important role in modeling has:
  - experience,
  - intuition,
  - beliefs and beliefs
  - total (non)technical knowledge of an engineer.

What is Modeling and Simulation (M&S)?[1]

- The question seems to be odd when featured in a lecture about Modeling and Simulation.
- Is it M&S Science?
- M&S as a discipline:

  - *M&S Science* contributes to the Theory of M&S, defining the academic foundations of the discipline.

  - *M&S Engineering* is rooted in Theory but looks for applicable solution patterns. The focus is general methods that can be applied in various problem domains.

  - *M&S Applications* solve real world problems by focusing on solutions using M&S. Often, the solution results from applying a method, but many solutions are very problem domain specific and are derived from problem domain expertise and not from any general M&S theory or method.

[1] J Padilla, Jose & Diallo, Saikou & Tolk, Andreas. (2011). Do we need M&S science?. SCS M&S Magazine.

Technical sciences

- Technical sciences such as: electrical engineering, mechanical engineering, construction, etc., provide methods that enable us to build useful physical objects.
- They are based on natural sciences and maths.
- Modeling of a original:
- The physical model,
- A symbolic model

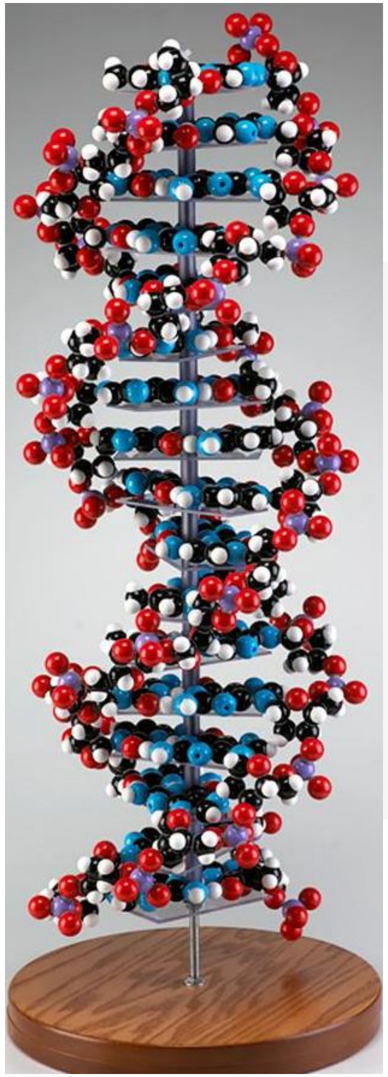
Typs of model

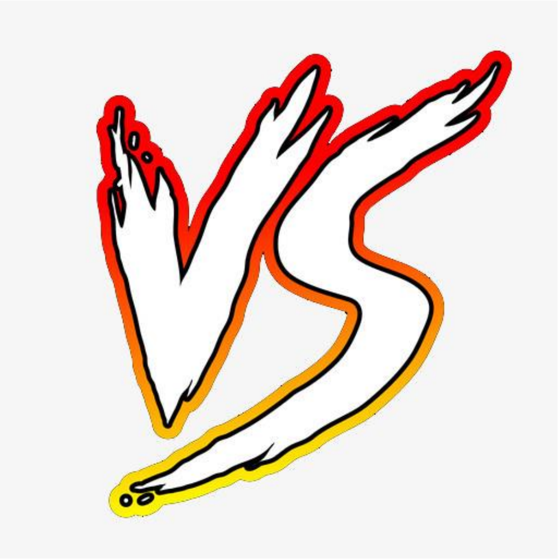
- Physical model
- Symbolic (Mathematical) model
    - Dynamic
    - Static
    - Continuous
    - Discrete
    - Stochastic
    - Deterministic

Erasmus+

e-Lab Prof Eng

Physical Model vs. Mathematical Model





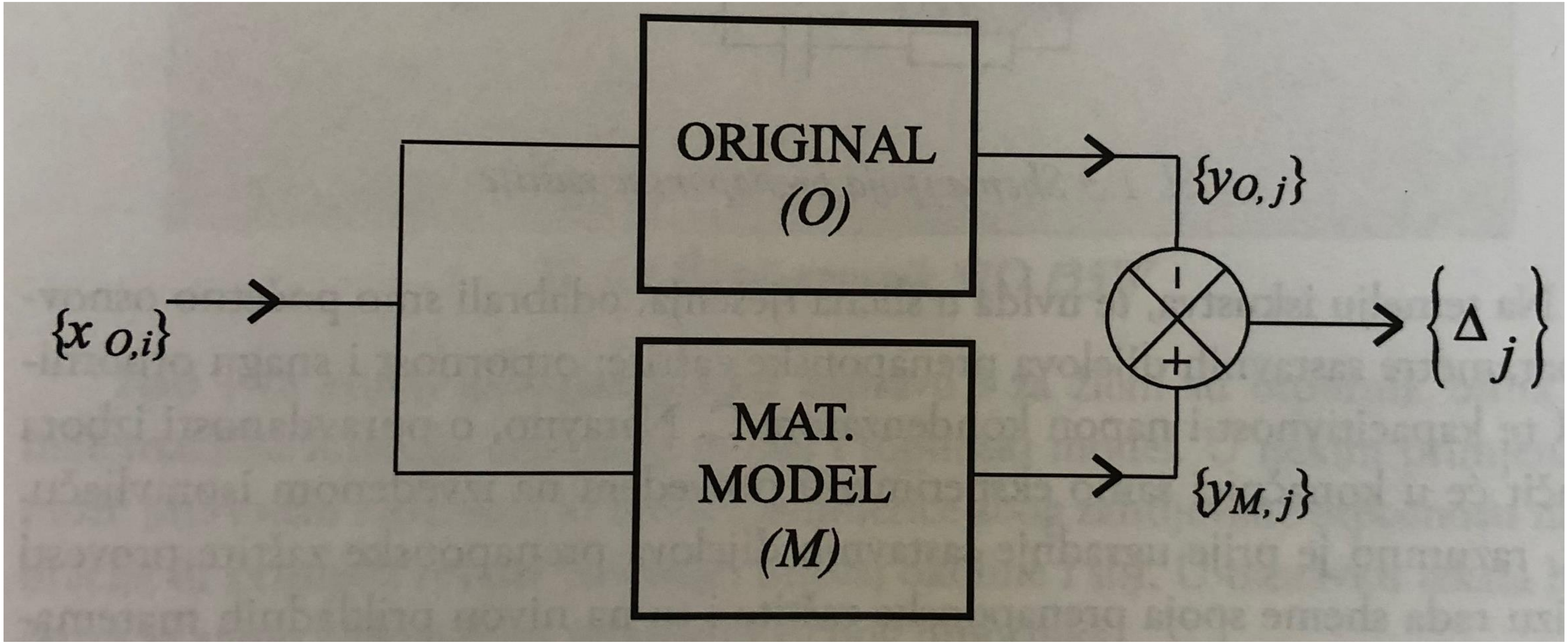

```
In[1]:= eksperiment = {{0.25, 8, 318.8}, {0.35, 8, 437}, {0.25, 14, 450},
    {0.35, 14, 530.3}, {0.3, 11, 445.6}, {0.3, 11, 467}, {0.3, 11, 475.5},
    {0.3, 11, 456.8}, {0.3, 11, 469}, {0.38, 11, 480.8}, {0.23, 11, 399},
    {0.3, 16, 588.2}, {0.3, 7, 320}}

Out[1]= {{0.25, 8, 318.8}, {0.35, 8, 437}, {0.25, 14, 450},
   {0.35, 14, 530.3}, {0.3, 11, 445.6}, {0.3, 11, 467},
   {0.3, 11, 475.5}, {0.3, 11, 456.8}, {0.3, 11, 469},
   {0.38, 11, 480.8}, {0.23, 11, 399}, {0.3, 16, 588.2}, {0.3, 7, 320}}

In[2]:= modelr31 = Fit[eksperiment, {1, x, x^2, y, y^2, x*y}, {x, y}]

Out[2]= -862.961 + 4795.03 x - 5520.38 x^2 + 63.3904 y - 63.1667 x y - 0.882181 y^2
```

Mathematical Model



$$\Delta_j = \left| y_{M,j} - y_{O,j} \right| \leq \varepsilon_{MO,j}, \quad j = 1, 2, \ldots, m$$

When modeling a large-scale system

- It is unrealistic to expect to put everything into just one model. Too much detail in a model can only be a distraction. It would be hard to use such a model as an aid to communication.
- Each model is used to illustrate a different point of view. For example, there are two different kinds of views that modellers often distinguish:
    - static models, which describe a set of elements and any relationships that exist between them;
    - dynamic models, which describe the behaviour of one or more elements over time.

Possible software solutions and their capabilities

List of softwares

- MATLAB
- GNU Octave
- Modelica/OpenModelica
- Ansys
- CATIA
- SolidWorks
- FreeCAD
- UML software

MATLAB

- is a programming environment for algorithm development, data analysis, visualization, and numerical computation.
- Developed by MathWorks.
- MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages.
- An additional package, Simulink, adds graphical multi-domain simulation and model-based design for dynamic and embedded systems.

GNU Octave

- is a part of the GNU Project, it is free software under the terms of the GNU General Public License.
- GNU Octave is software featuring a high-level programming language, primarily intended for numerical computations. Octave helps in solving linear and nonlinear problems numerically, and for performing other numerical experiments using a language that is mostly compatible with MATLAB.
- Octave is one of the major free alternatives to MATLAB, others being Scilab and FreeMat.

Modelica/OpenModelica

- Modelica is an object-oriented, declarative, multi-domain modeling language for component-oriented modeling of complex systems (systems containing mechanical, electrical, electronic, hydraulic, thermal, control, electric power or process-oriented subcomponents).
- The OpenModelica language is developed by the non-profit Modelica Association. The Modelica Association also develops the free Modelica Standard Library that contains about 1360 generic model components and 1280 functions in various domains.

Ansys

- Ansys develops and markets finite element analysis software used to simulate engineering problems.
- The software creates simulated computer models of structures, electronics, or machine components to simulate strength, toughness, elasticity, temperature distribution, electromagnetism, fluid flow, and other attributes.
- Ansys is used to determine how a product will function with different specifications, without building test products or conducting crash tests.
- The finite element method (FEM), is a numerical method for solving problems of engineering and mathematical physics. Typical problem areas of interest include structural analysis, heat transfer, fluid flow, mass transport, and electromagnetic potential.

CATIA

- commonly referred to as a 3D Product Lifecycle Management software suite, is a multi-platform software suite for computer-aided design (CAD), computer-aided manufacturing (CAM), computer-aided engineering (CAE), PLM and 3D.
- Developed by the French company Dassault Systèmes.
- CATIA facilitates collaborative engineering across disciplines around its 3DEXPERIENCE platform, including surfacing & shape design, electrical, fluid and electronic systems design, mechanical engineering and systems engineering.
- CATIA facilitates the design of electronic, electrical, and distributed systems such as fluid and HVAC systems, all the way to the production of documentation for manufacturing.

SolidWorks

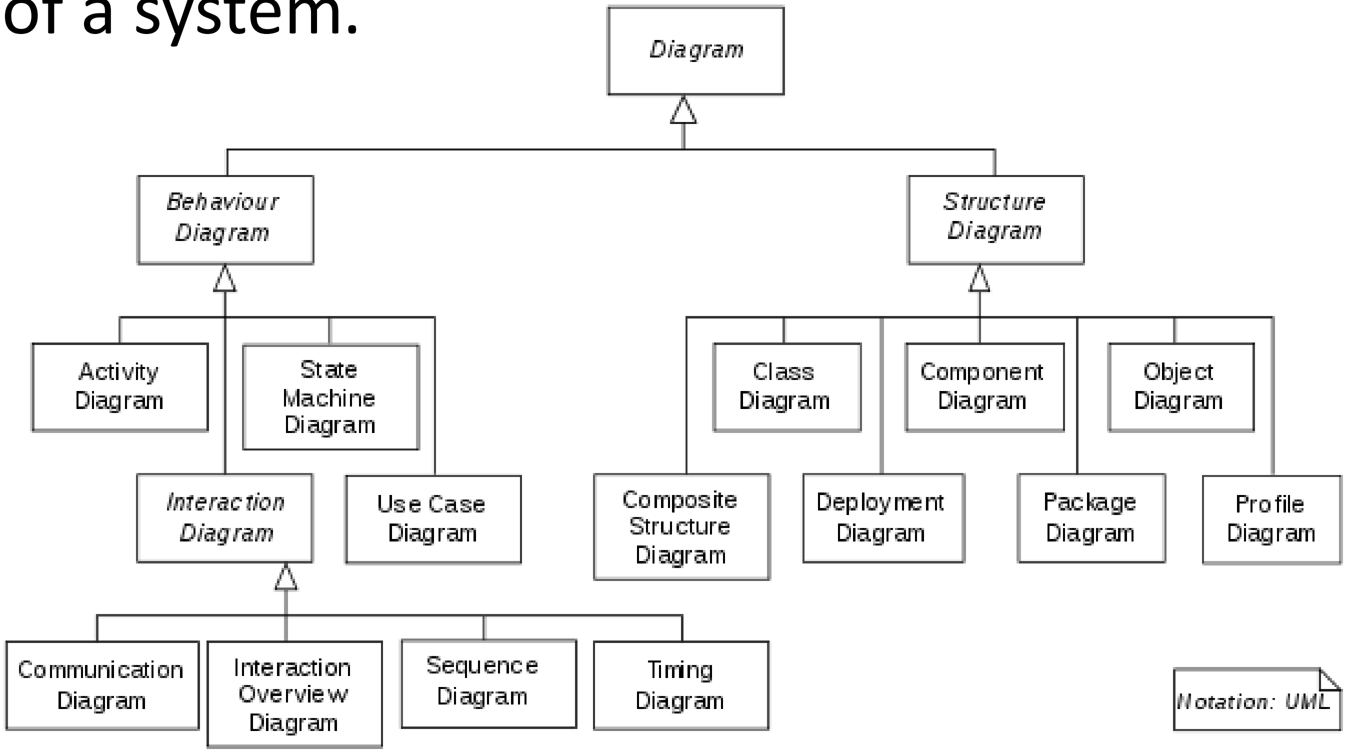
- is a solid modeling computer-aided design (CAD) and computer-aided engineering (CAE) computer program.
- SolidWorks is published by Dassault Systèmes.
- Over two million engineers and designers at more than 165,000 companies were using SolidWorks as of 2013.
- SOLIDWORKS Products & Solutions includes 2D/3D CAD, Simulation Solutions, Product Data Management, Electrical Design, Visualization, Technical Communication, CAM, Product Configurator, Collaboration

FreeCAD

- is a free and open-source (LGPLv2+) general-purpose parametric 3D CAD modeler and a building information modeling software with finite-element-method (FEM) support.
- FreeCAD is aimed directly at mechanical engineering product design but also expands to a wider range of uses around engineering, such as architecture or electrical engineering.
- FreeCAD features tools similar to Autodesk Revit, CATIA, Creo, Autodesk Inventor, SolidWorks or Solid Edge.

UML software

- The Unified Modeling Language (UML) is a general-purpose, developmental, modeling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system.



Erasmus+

e-Lab Prof Eng

UML software

- **ArgoUML** is an UML diagramming application written in Java and released under the open source Eclipse Public License.
- **StarUML** is a UML tool by MKLab. The software was licensed under a modified version of GNU GPL until 2014.
- **Microsoft Visio**, as one of the MS Office suite applications, is the industry standard diagramming software with which users can set up a wide range of diagrams, which include UML classes.
- **Lucidchart** is online UML diagram tool. It is a web-based proprietary platform that is used to allow users who are located in multiple locations to collaborate with their colleagues in drawing, revising and sharing charts and diagrams. Lucidchart runs on browsers that support HTML5.

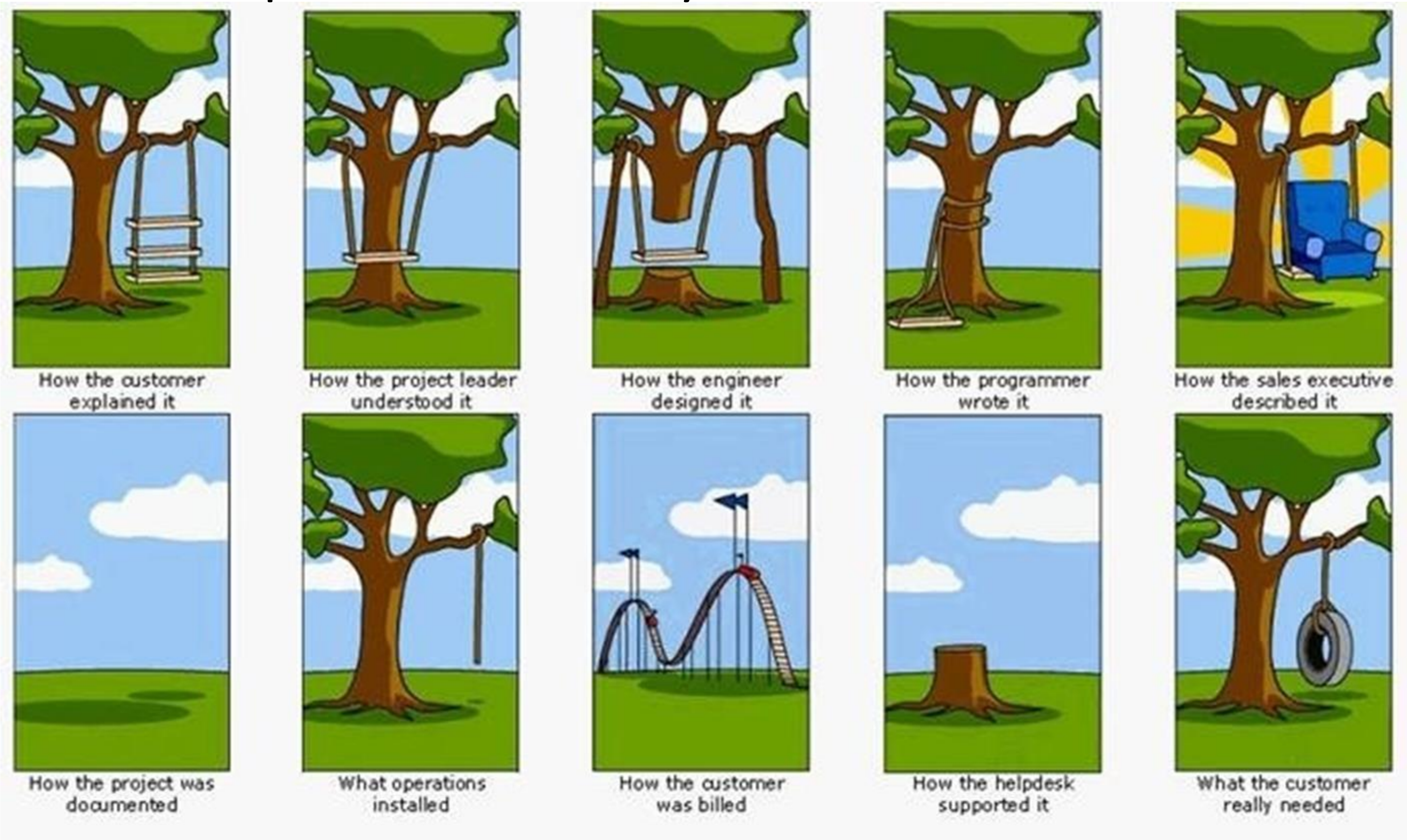
UML software

- Eclipse is an integrated development environment (IDE) used in computer programming, and is the most widely used Java IDE.
- It contains a base workspace and an extensible plug-in system for customizing the environment.
- **Eclipse UML2 Tools** is part of Model Development Tools (MDT)

Erasmus+

e-Lab Prof Eng

Product Development Life Cycle



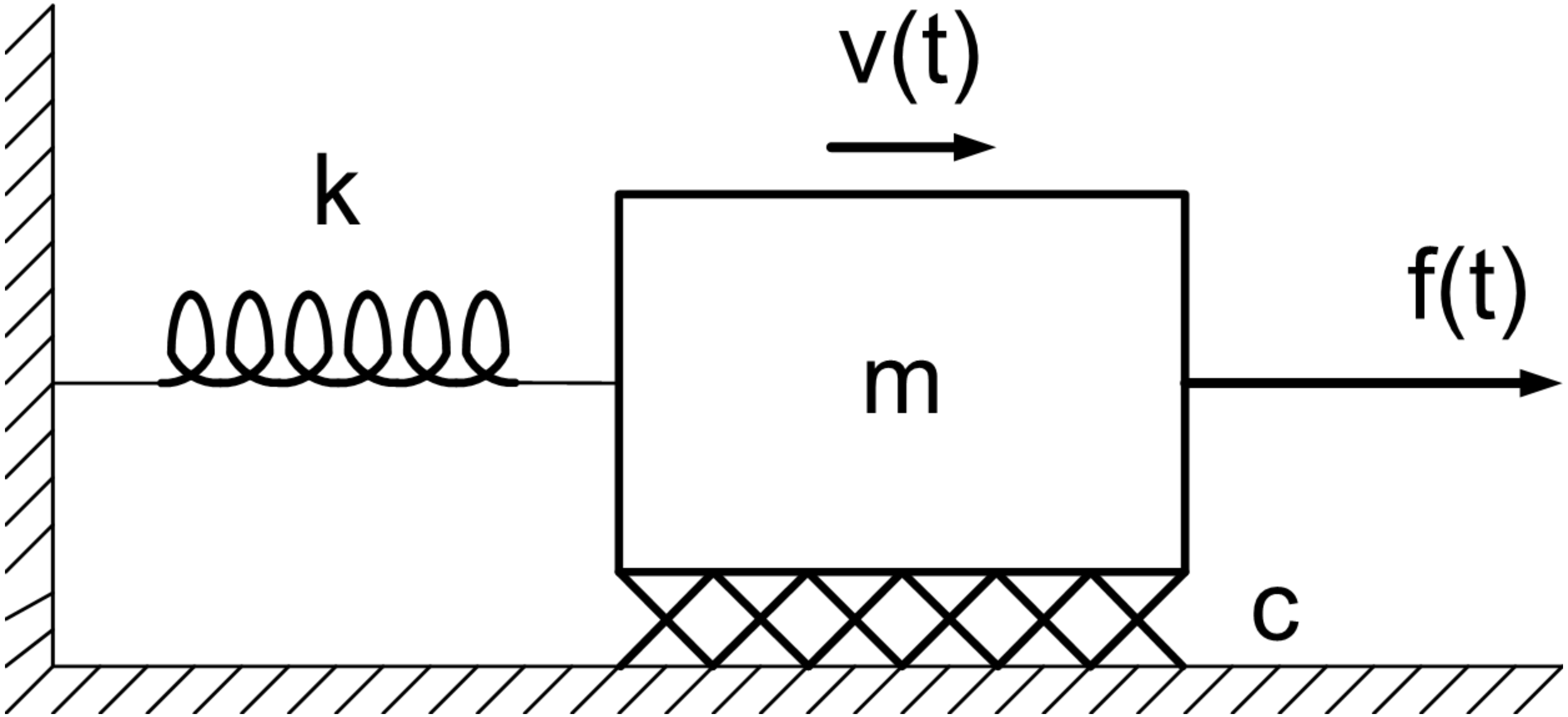
Erasmus+

e-Lab Prof Eng

Part 2.

Solving tasks

Translating Mechanical System Model



$$m \cdot \frac{dv}{dt} + c \cdot v(t) + k \cdot \int_0^t v(\lambda) \cdot d\lambda = f(t)$$

Basic system variables

- length **x [m]**
- speed **v [m/s]**

$$v(t) = \frac{dx}{dt} = \dot{x}$$

- acceleration **a [m/s²]**

$$a(t) = \frac{dv}{dt} = \dot{v} = \frac{d^2x}{dt^2} = \ddot{x}$$

- force **f [N]**

$$[N] = \left[\frac{kg\ m}{s^2}\right]$$

Additional system variables

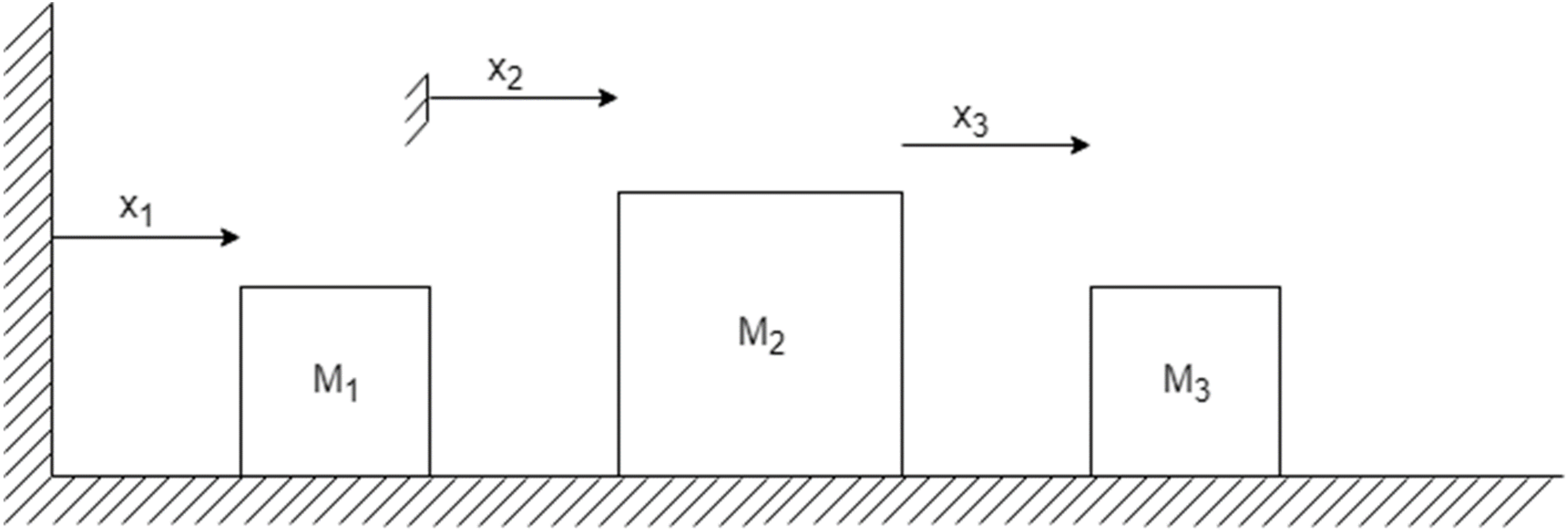
- Energy **w [J]**

$$w(t) = w(t_0) + \int_{t_o}^{t_1} p(t) \cdot dt$$

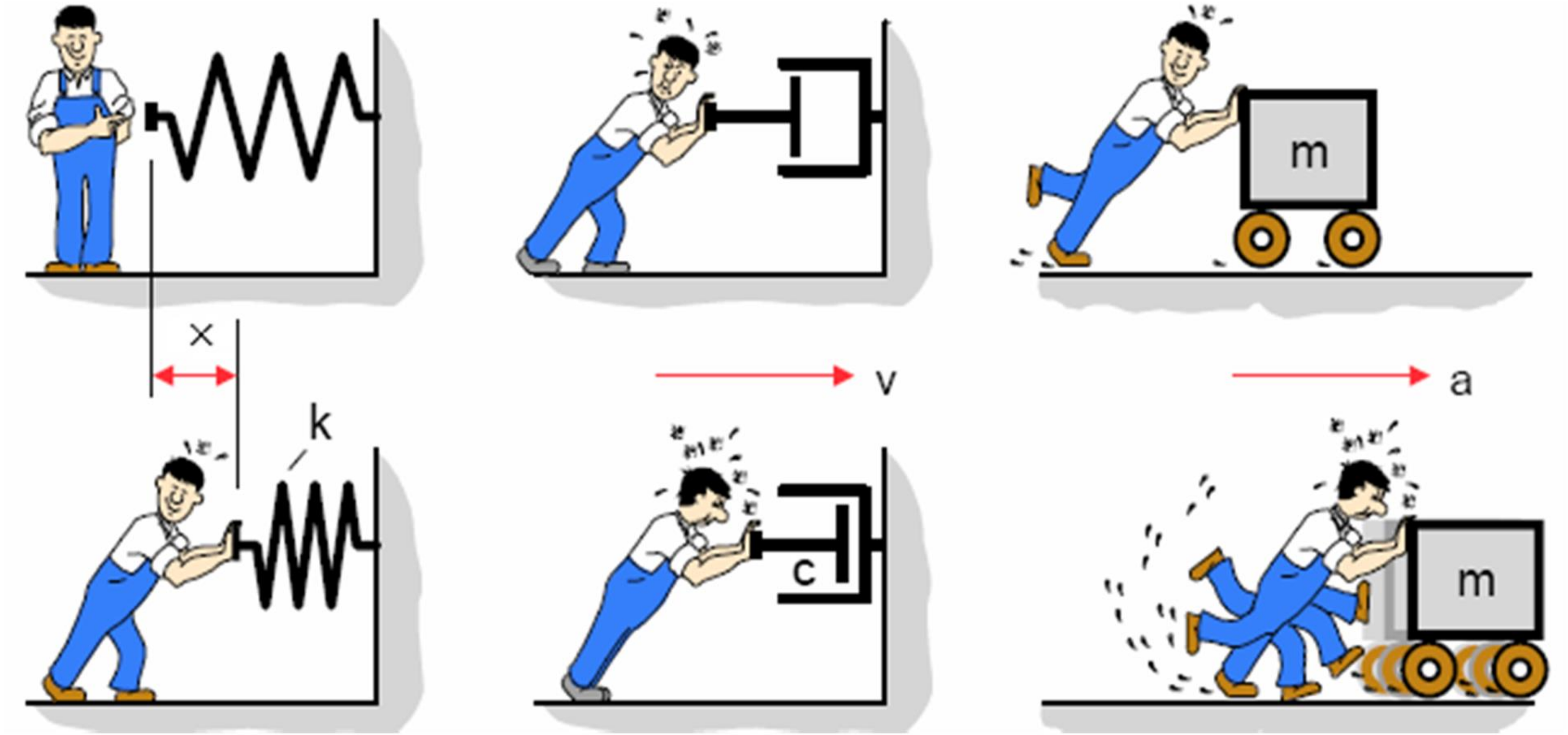
- Power **p [W]**

$$p = \frac{dw}{dt} \qquad p = f \cdot v$$

Denotating the positive direction of motion



System elements



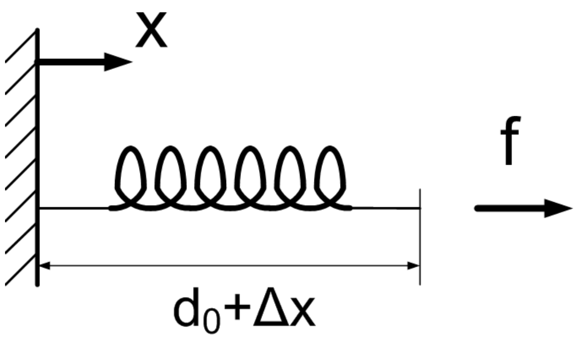
$$f = k \cdot (x_2 - x_1)$$
$$f = k \cdot \Delta x$$

$$f = c \cdot (v_2 - v_1)$$
$$f = c \cdot \Delta v$$

$$f = m \cdot (v_2 - v_1)/t$$
$$f = m \cdot \Delta v/t$$

Spring

- A linear translating spring is an element that is deformed (shortened or lengthened) by Δx in direct proportion to the amount of force applied.
- Ideal springs have no mass.



(d_0 – spring lenght without acting force)

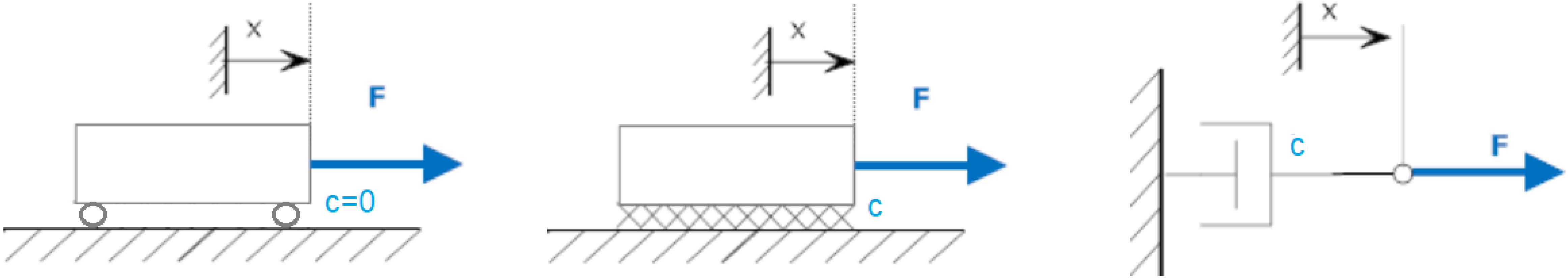
- For small stretches linear behavior is valid:

$$f = k \cdot \Delta x$$

where is **k [N/m]** – spring constant.

Friction – Model of cylinder

- The force of friction occurs when two bodies (who are in contact) move at different speeds.



- Linear behavior: $f = c \cdot \Delta v \qquad \Delta v = v_2 - v_1$

  where is **c [Nm/s]** – friction coefficient

<table>
<tr><th>Description</th><th>Graphical Representation</th></tr>
<tr><td>

Static Friction: friction between two objects with no relative motion between them. The amount of force required to move one object relative to the other is given by

$$F_s = \mu_s F_N$$

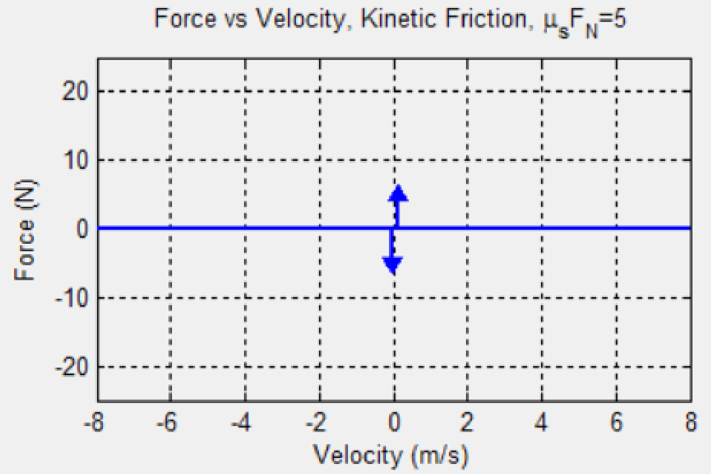
where μ_s is the static coefficient of friction (no units), and F_N is the normal force between the objects (N). For example, a 2 kg mass, m, on a table with static friction coefficient μ_s=0.1, requires

$$\text{force} = \mu_s F_N = \mu_s mg = 0.1 \cdot 2 \cdot 9.8 = 1.96 \text{ Newtons}$$

to get it to move. As soon as the object starts moving, the static friction is gone and is replaced by a combination of the types of friction listed below.

Note: Though the friction only exists for velocity=0 the two arrows are shown slightly offset from the origin to indicate that they are active when the force is in the direction of the offset. In this graph a static friction force of of up to 5 N exists when the force is such that it will cause the mass to move in the positive direction.

</td><td>



</td></tr>
<tr><td>

Kinetic friction: friction between two objects moving relative to each other. The force due to kinetic friction opposes the velocity, but its magnitude is constant and independent of the velocity.

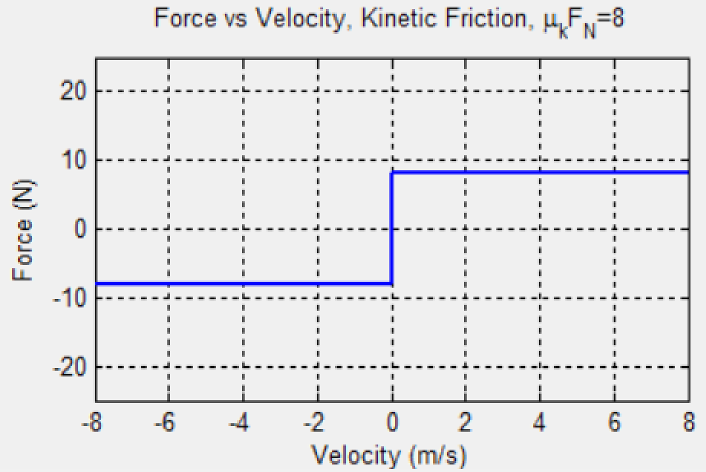
$$F_k = \mu_k N$$

In this equation μ_k the kinetic friction coefficient (no units), and F_N is again the normal force between the object (N). For two objects, the coefficient μ_k is always less μ_s. A 2 kg mass, m, on a table with static friction coefficient μ_k=0.02, requires

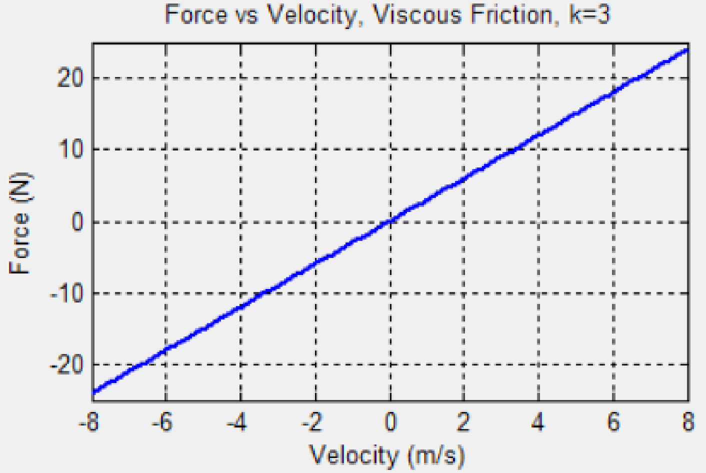
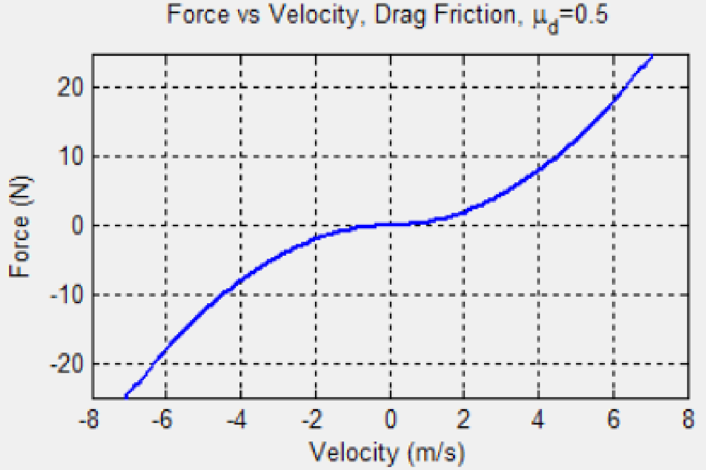
$$\text{force} = \mu_k F_N = \mu_k mg = 0.02 \cdot 2 \cdot 9.8 = 0.39 \text{ Newtons}$$

to keep it moving.

</td><td>



</td></tr>
</table>

Viscous friction: friction between two objects moving relative to each other (this is the friction discussed above). The force due to kinetic friction opposes and is proportional to the velocity. $F_V = b \cdot v$ where b is the friction coefficient (N-s/m) and v is the velociy (m/S).	
Drag: friction experienced by an object moving through a fluid (often air or water). This force is proportional to the square of the velocity, and opposed to the direction of the velocity. $F_d = \mu_d \cdot v \cdot \lvert v \rvert$; $\lvert F_d \rvert = \mu_d \cdot v^2$ The absolute value in the first equation is necessary to ensure that the drag force opposes the direction of motion (i.e., it changes direction when the objects velocity changes direction).	

In a practical situation, the actual friction will be some combination of the friction types mentioned above (and probably some even more complicated relationships). For the purposes of this document, all friction will be assumed to be viscous unless stated otherwise.

Mass

- The SI units for a mass are kilograms **m [kg]**

$$f = \frac{d}{dt}(m \cdot v) \qquad \text{za } m = \text{const} \qquad f = m \cdot \frac{dv}{dt}$$

- Energy

➢ Kinetic

$$w_k = \frac{1}{2}\left(m \cdot v^2\right)$$

➢ Potential

$$w_p = m \cdot g \cdot h$$

D'Alembert's Law

- Newton's second law states that an object accelerates in the direction of an applied force, and that this acceleration is inversely proportional to the force, or

$$\sum_{\text{all external}} F = m \cdot a$$

- we will bring the right-hand side to the left and express this as

$$\sum_{\text{all external}} F - m \cdot a = 0$$

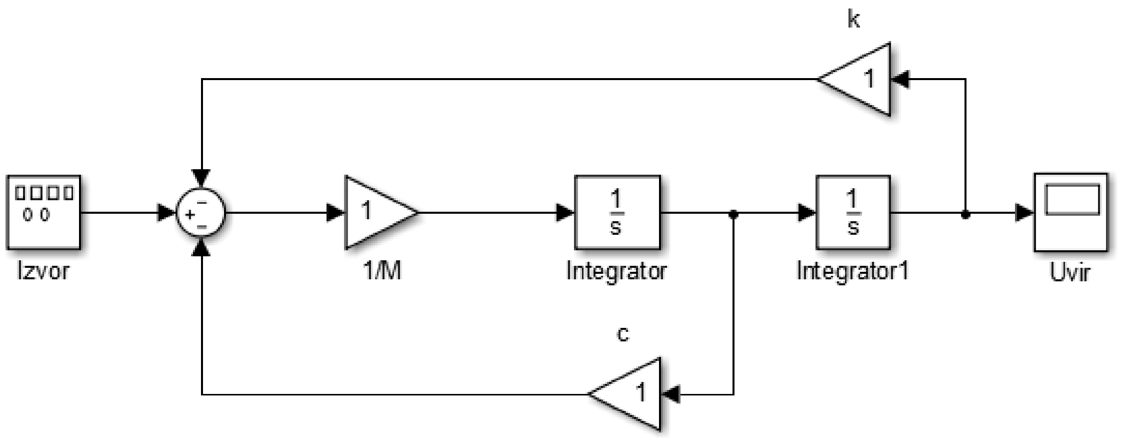
- If we consider the m·a term to be a force, we are left with D'Alembert's law

$$\sum_{\text{all}} F = 0$$

Key points

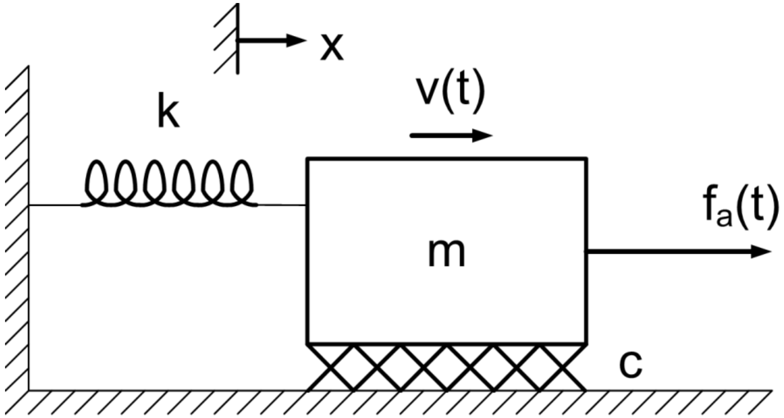
- Three elements were introduced:
  - springs,
  - friction elements and
  - inertial elements (masses).
- An ideal linear spring has no mass and a linear relationship between force and elongation. For viscous friction there is a linear relationship between force and velocity. Friction may either be between two surfaces or between two objects. Masses have a linear relationship between force and acceleration.

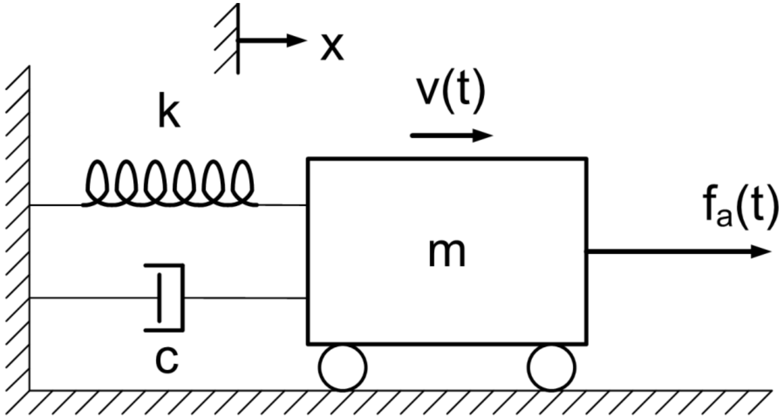
Block diagrams

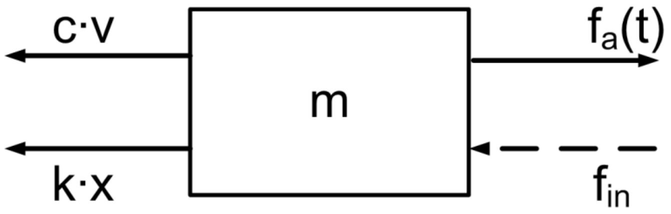


- Characteristics of block diagrams are:
  - signal flows are clearly displayed
  - energy flows are not clearly visible
- Block diagrams elements are:
  - blocks (the output variable is determined by the block function on the input variable)
  - summation points
  - signals (vectors in which the arrow determines the signal direction)
  - branching points
  - sources
  - sinks

Obtaining System Model - Example





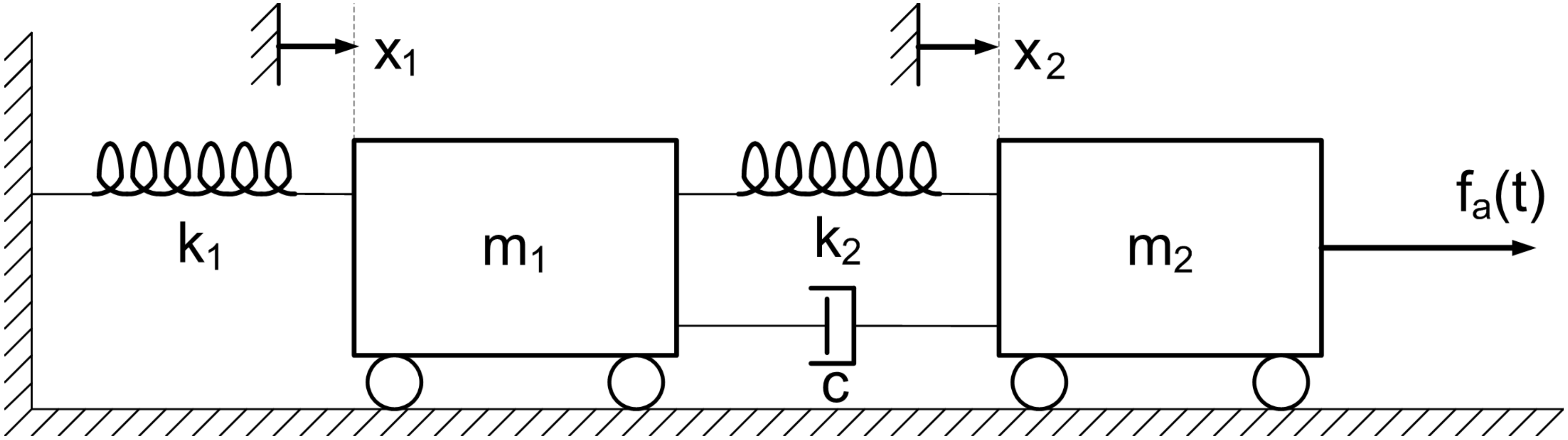


$$f_a(t) - (M \cdot \dot{v} + c \cdot v + k \cdot x) = 0$$

$$(M \cdot \ddot{x} + c \cdot \dot{x} + k \cdot x) = f_a(t)$$

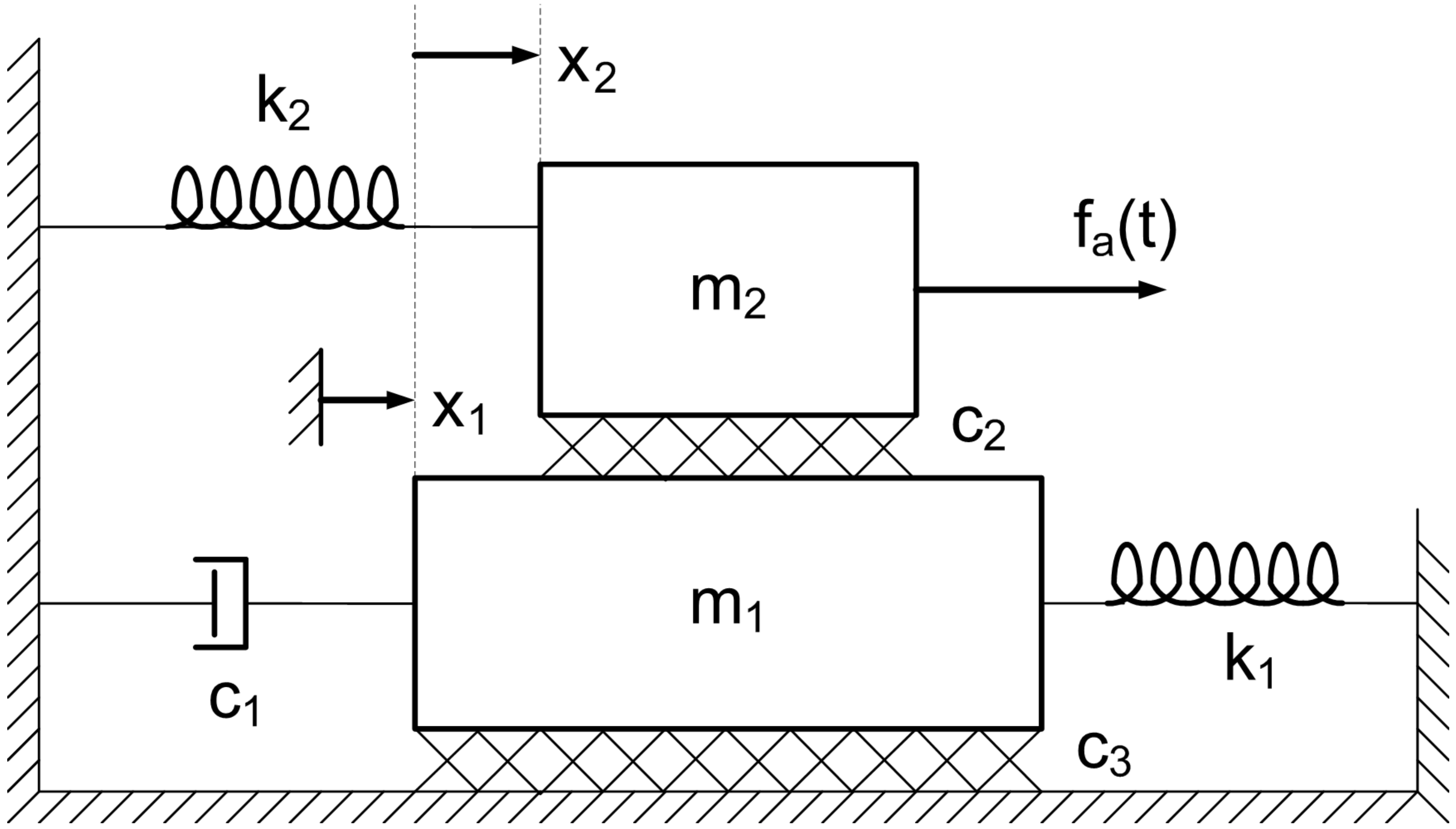
Task 1

- Draw a diagram of free bodies and apply D'Alembert's law for the system shown in the picture.



Task 2a

- Draw a diagram of free bodies for the system shown in the picture.

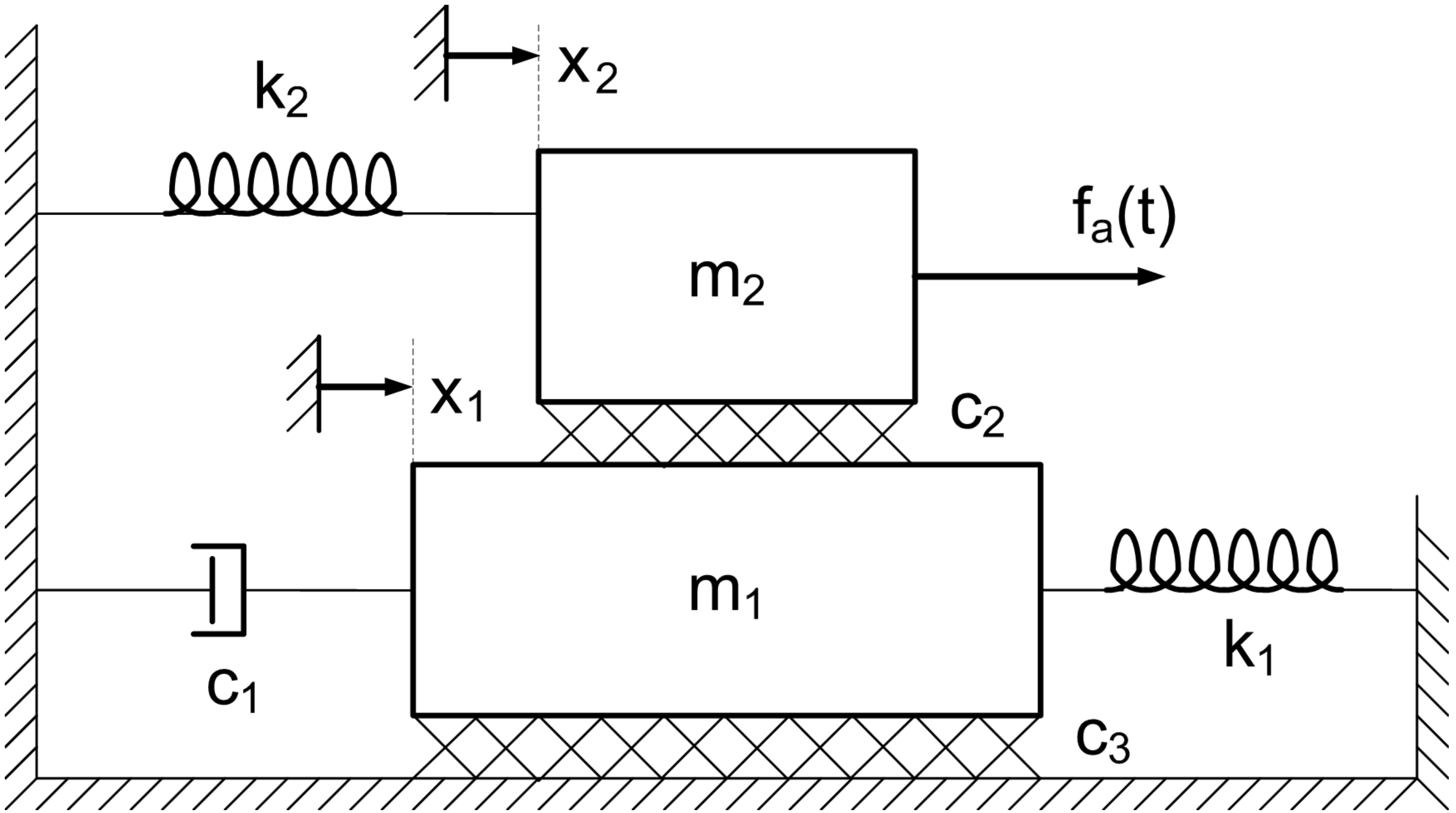


Erasmus+

e-Lab Prof Eng

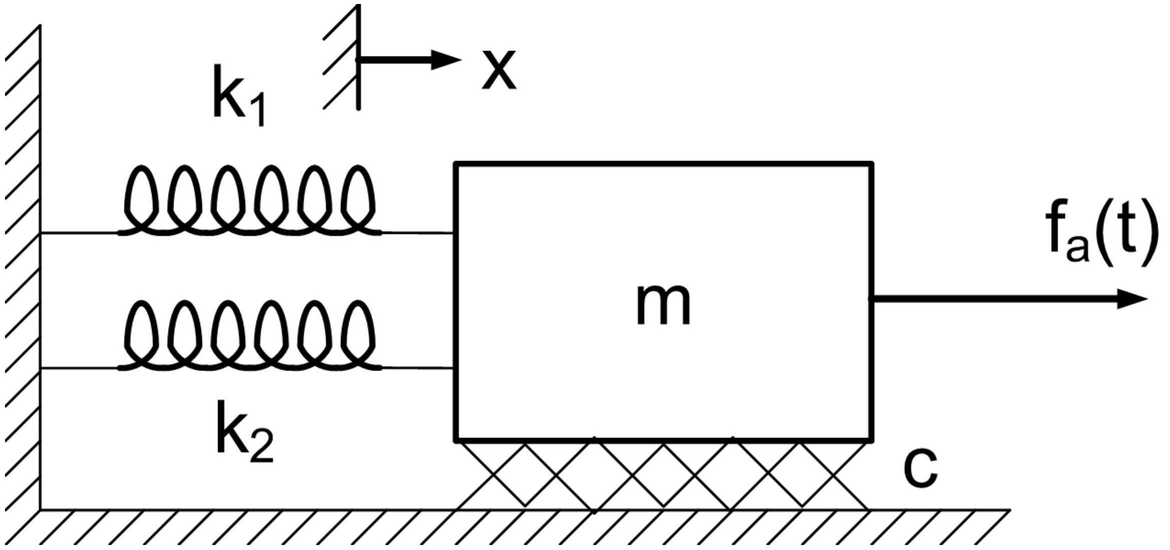
Task 2b

- Draw a diagram of free bodies for the system shown in the picture.



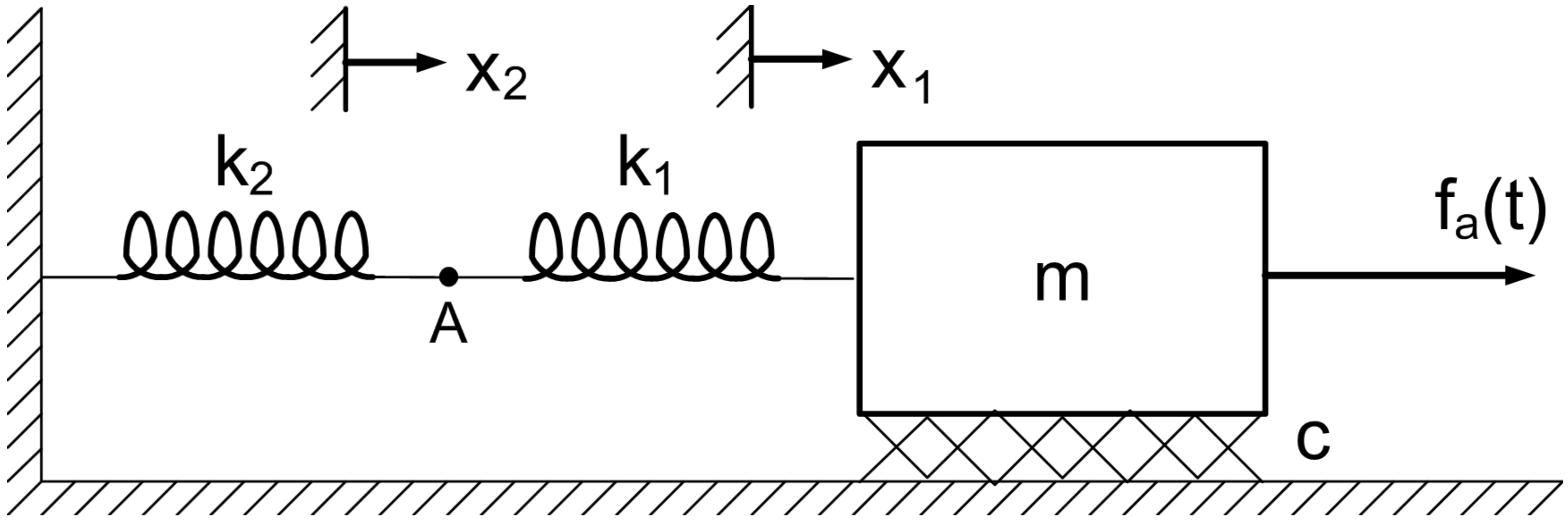
Task 3

- The system shown in the picture includes two linear springs between the wall and the mass. Write differential equations describing the mass m and find the equivalent spring constant.

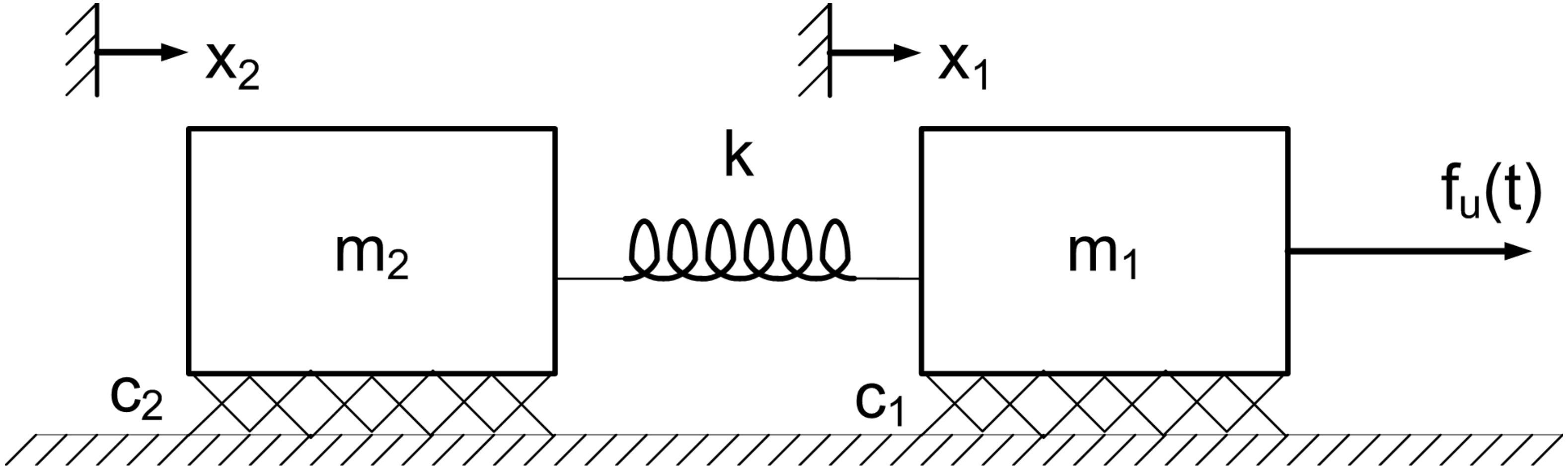


Task 4

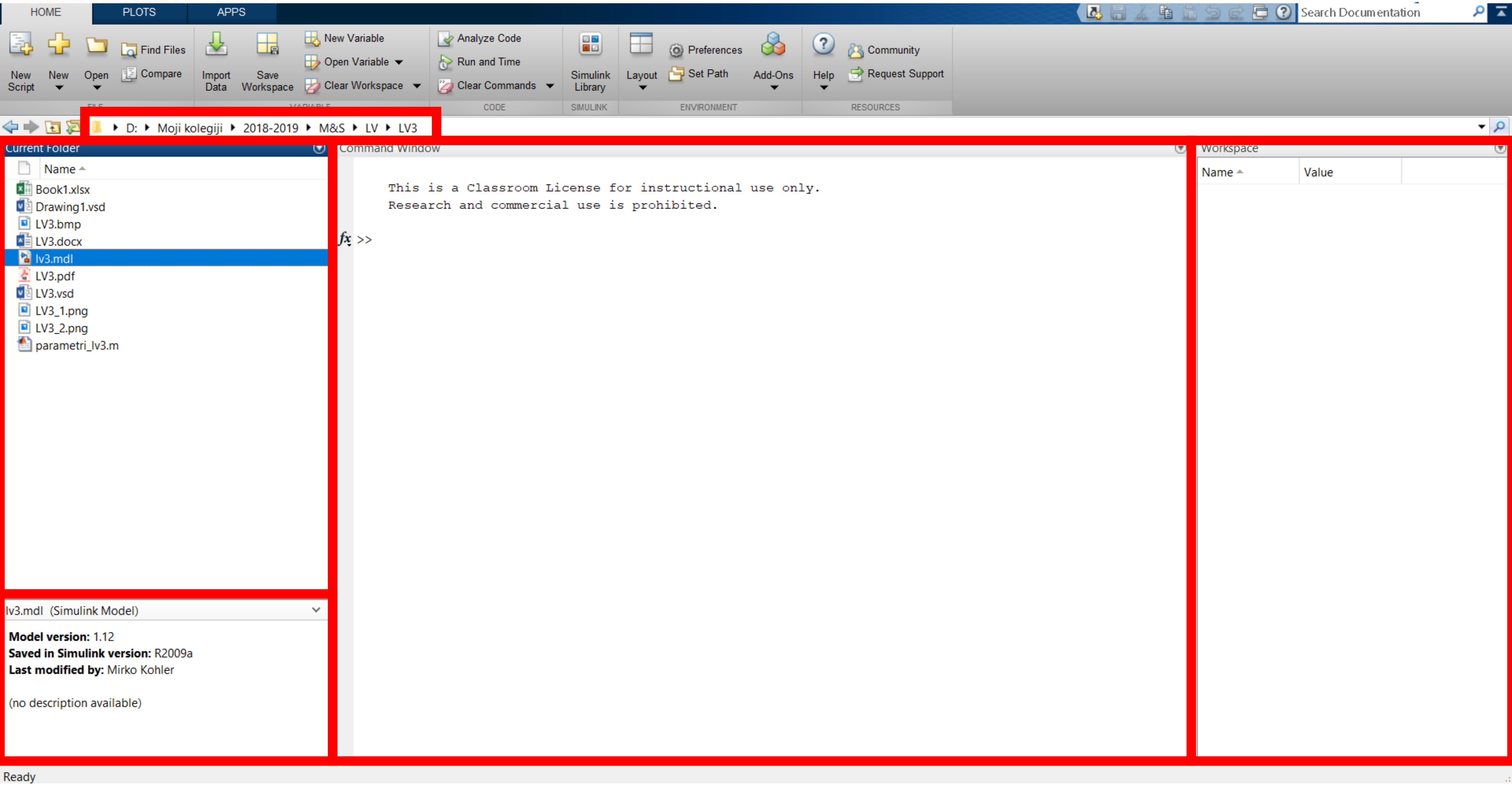
- Draw the diagram of free bodies for M and point A, and write differential equations describing the system. Find k_{eq}.

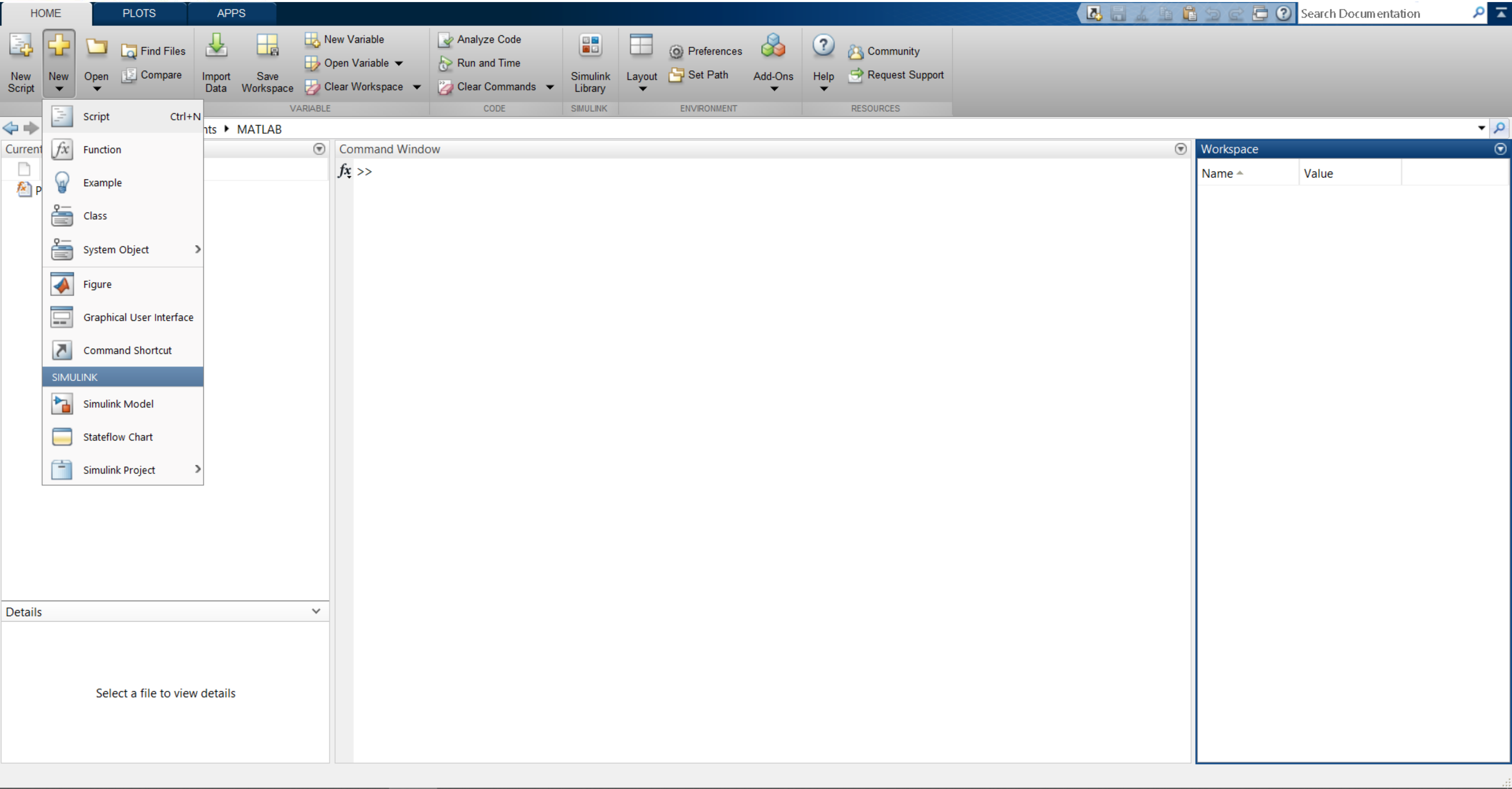


Task 5



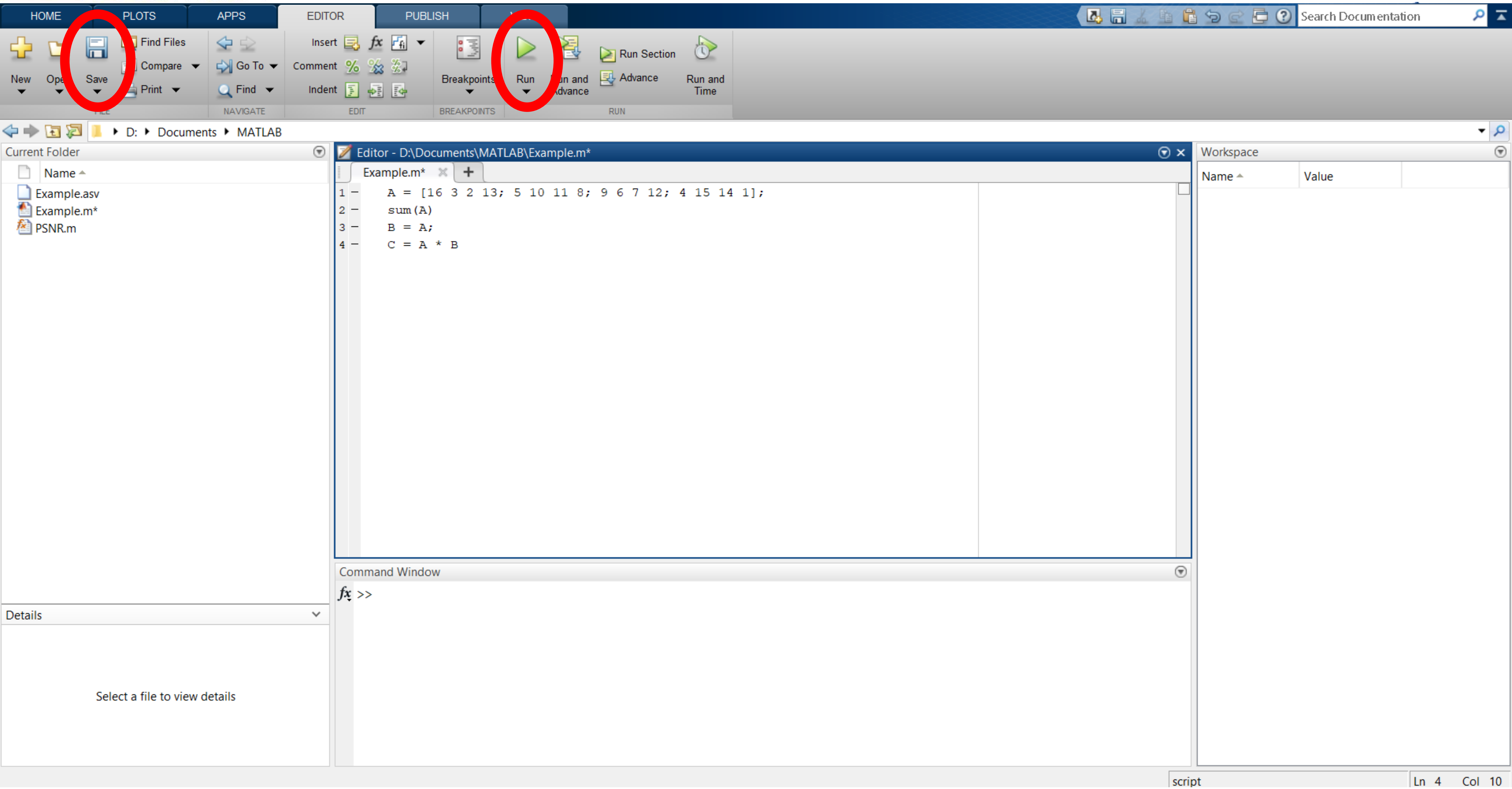
Modeling and simulation in Matlab/Simulink


HOME
PLOTS
APPS
Search Documentation
New Script
New
Open
Find Files
Compare
Import Data
Save Workspace
New Variable
Open Variable
Clear Workspace
Analyze Code
Run and Time
Clear Commands
CODE
Simulink Library
SIMULINK
Layout
Preferences
Set Path
Add-Ons
ENVIRONMENT
Help
Community
Request Support
RESOURCES
D: ▸ Moji kolegiji ▸ 2018-2019 ▸ M&S ▸ LV ▸ LV3
Current Folder
Name
Book1.xlsx
Drawing1.vsd
LV3.bmp
LV3.docx
lv3.mdl
LV3.pdf
LV3.vsd
LV3_1.png
LV3_2.png
parametri_lv3.m
lv3.mdl (Simulink Model)
Model version: 1.12
Saved in Simulink version: R2009a
Last modified by: Mirko Kohler
(no description available)
Command Window
This is a Classroom License for instructional use only.
Research and commercial use is prohibited.
fx >>
Workspace
Name
Value
Ready


HOME
PLOTS
APPS
Search Documentation
New Script
New
Open
Find Files
Compare
Import Data
Save Workspace
New Variable
Open Variable
Clear Workspace
VARIABLE
Analyze Code
Run and Time
Clear Commands
CODE
Simulink Library
SIMULINK
Layout
Preferences
Set Path
Add-Ons
ENVIRONMENT
Help
Community
Request Support
RESOURCES
Script Ctrl+N
Function
Example
Class
System Object
Figure
Graphical User Interface
Command Shortcut
SIMULINK
Simulink Model
Stateflow Chart
Simulink Project
MATLAB
Current
Command Window
fx >>
Workspace
Name
Value
Details
Select a file to view details

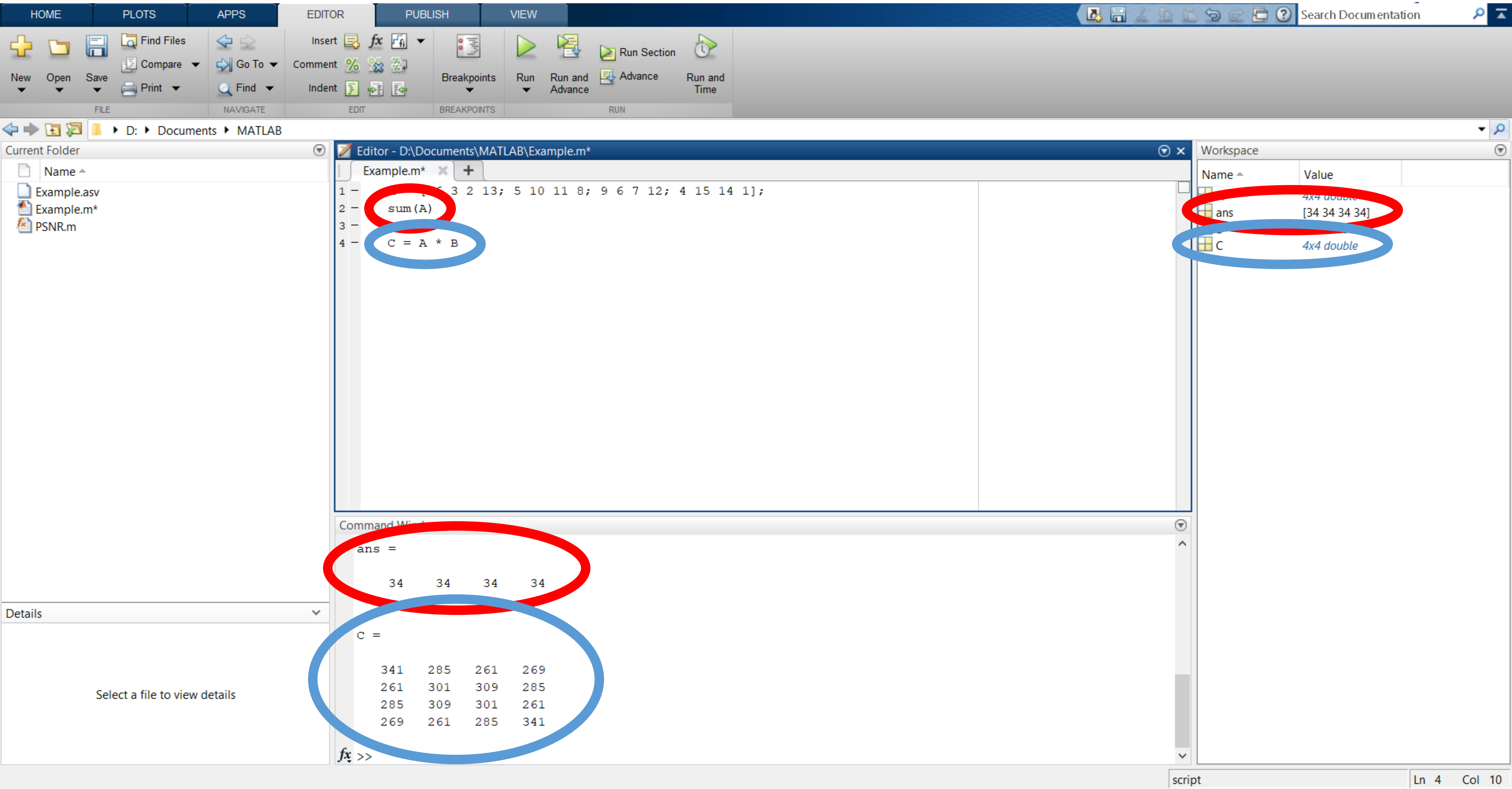
Erasmus+

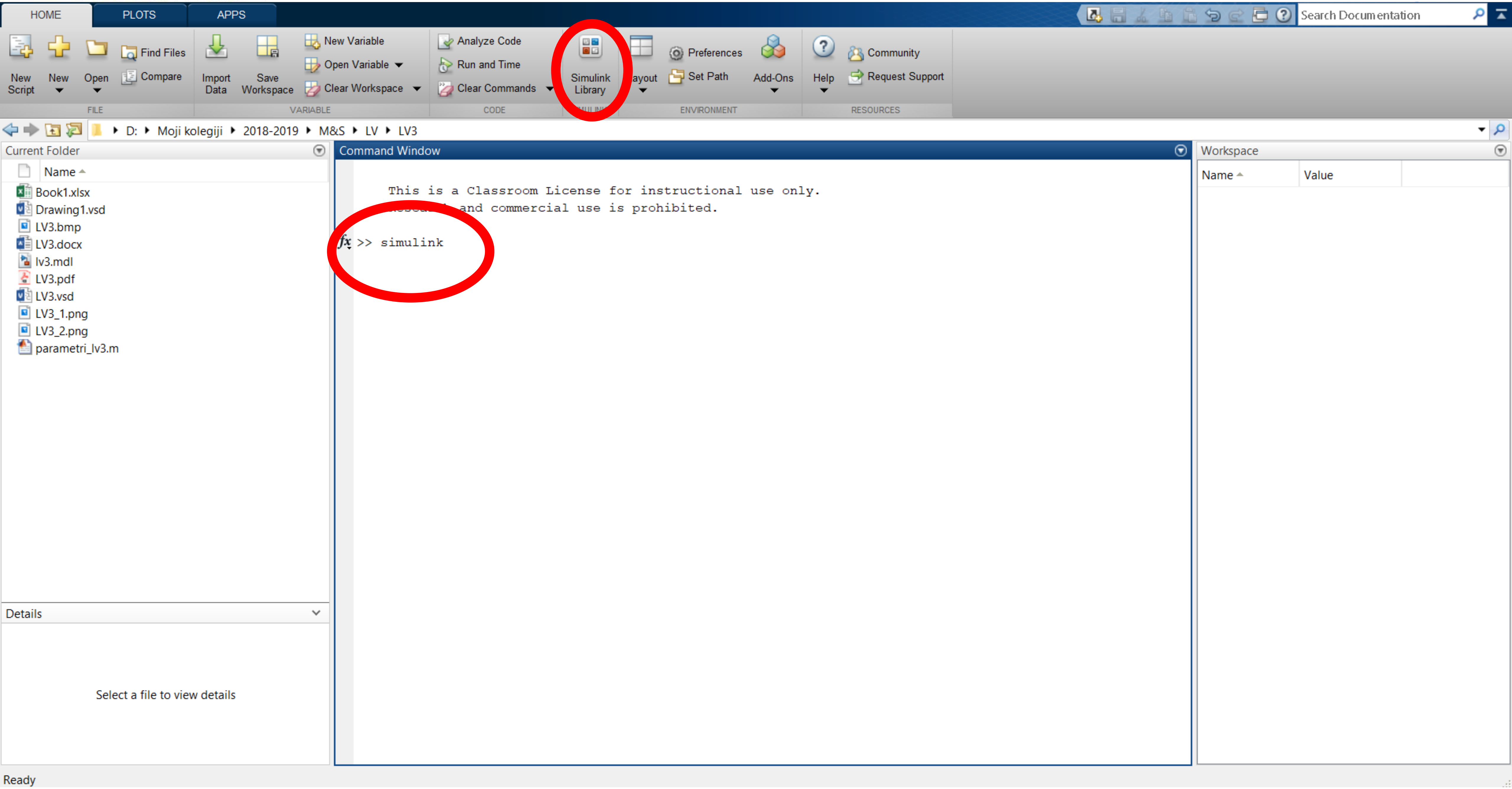
e-Lab



Erasmus+

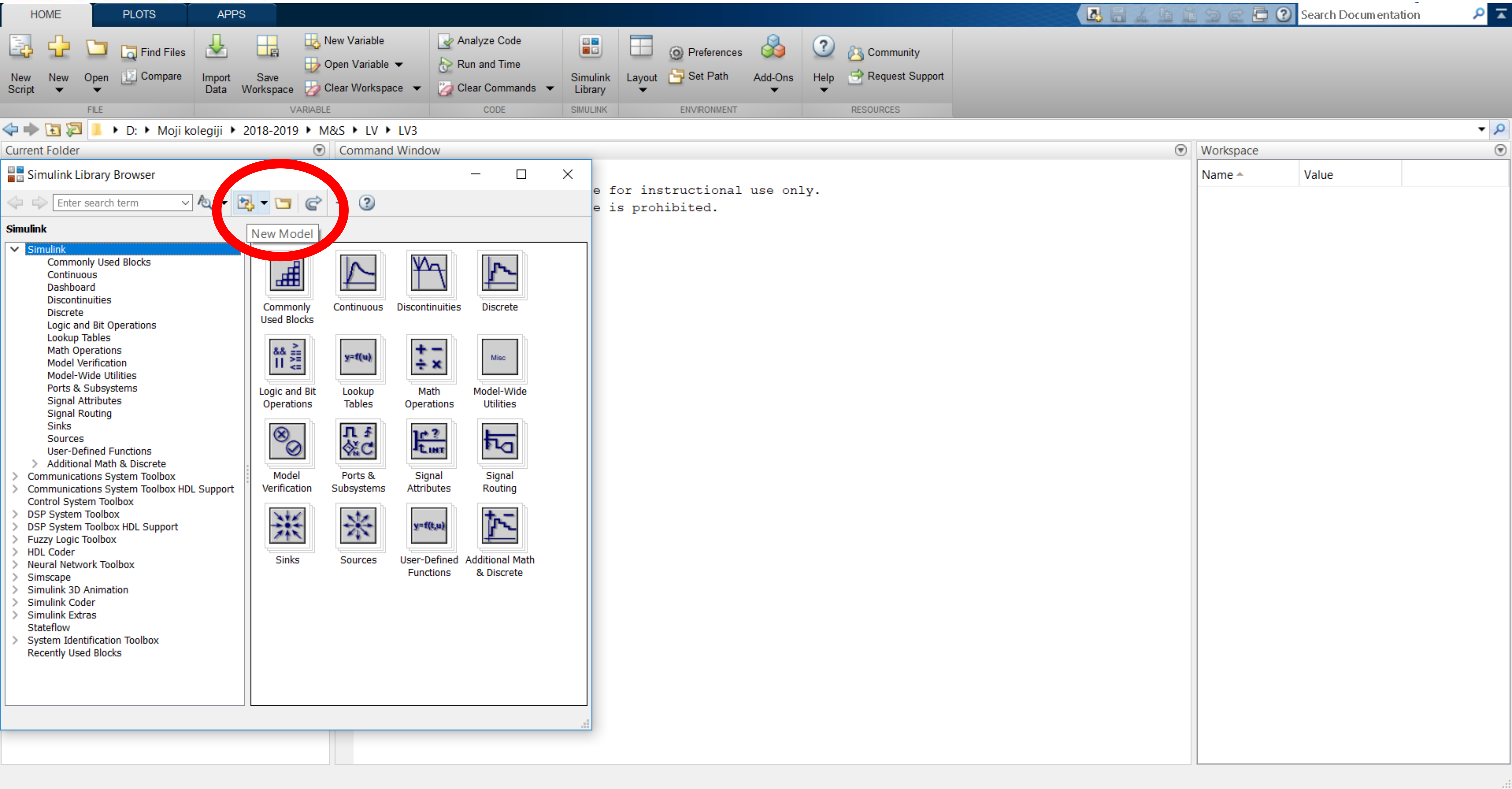
e-Lab



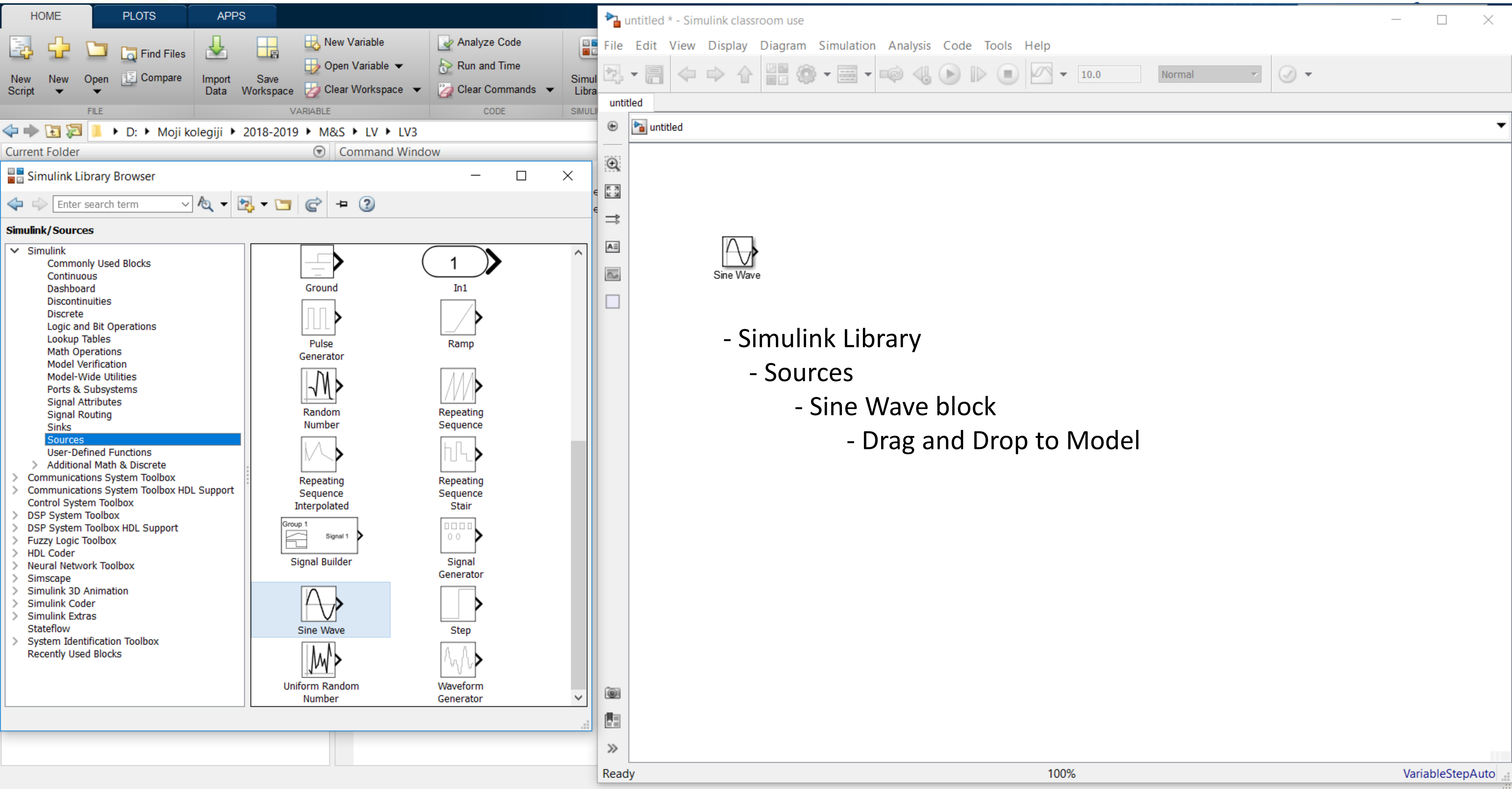

HOME
PLOTS
APPS
Search Documentation
New Script
New
Open
Find Files
Compare
FILE
Import Data
Save Workspace
New Variable
Open Variable
Clear Workspace
VARIABLE
Analyze Code
Run and Time
Clear Commands
CODE
Simulink Library
Layout
Preferences
Set Path
Add-Ons
ENVIRONMENT
Help
Community
Request Support
RESOURCES
D: ▸ Moji kolegiji ▸ 2018-2019 ▸ M&S ▸ LV ▸ LV3
Current Folder
Name
Book1.xlsx
Drawing1.vsd
LV3.bmp
LV3.docx
lv3.mdl
LV3.pdf
LV3.vsd
LV3_1.png
LV3_2.png
parametri_lv3.m
Details
Select a file to view details
Command Window
This is a Classroom License for instructional use only.
and commercial use is prohibited.
>> simulink
Workspace
Name
Value
Ready

Erasmus+

e-Lab



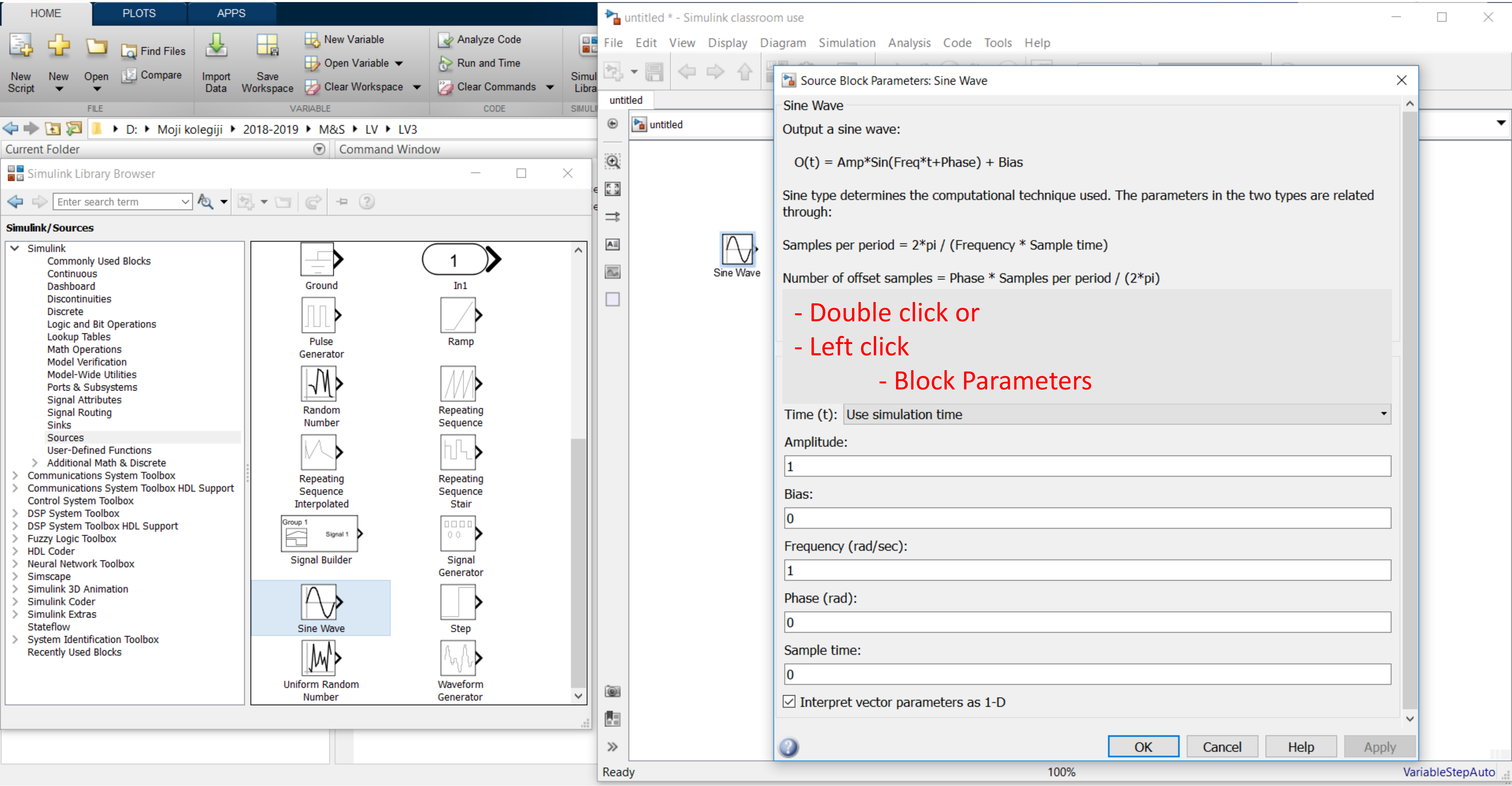
Erasmus+ e-Lab



- Simulink Library
  - Sources
    - Sine Wave block
      - Drag and Drop to Model

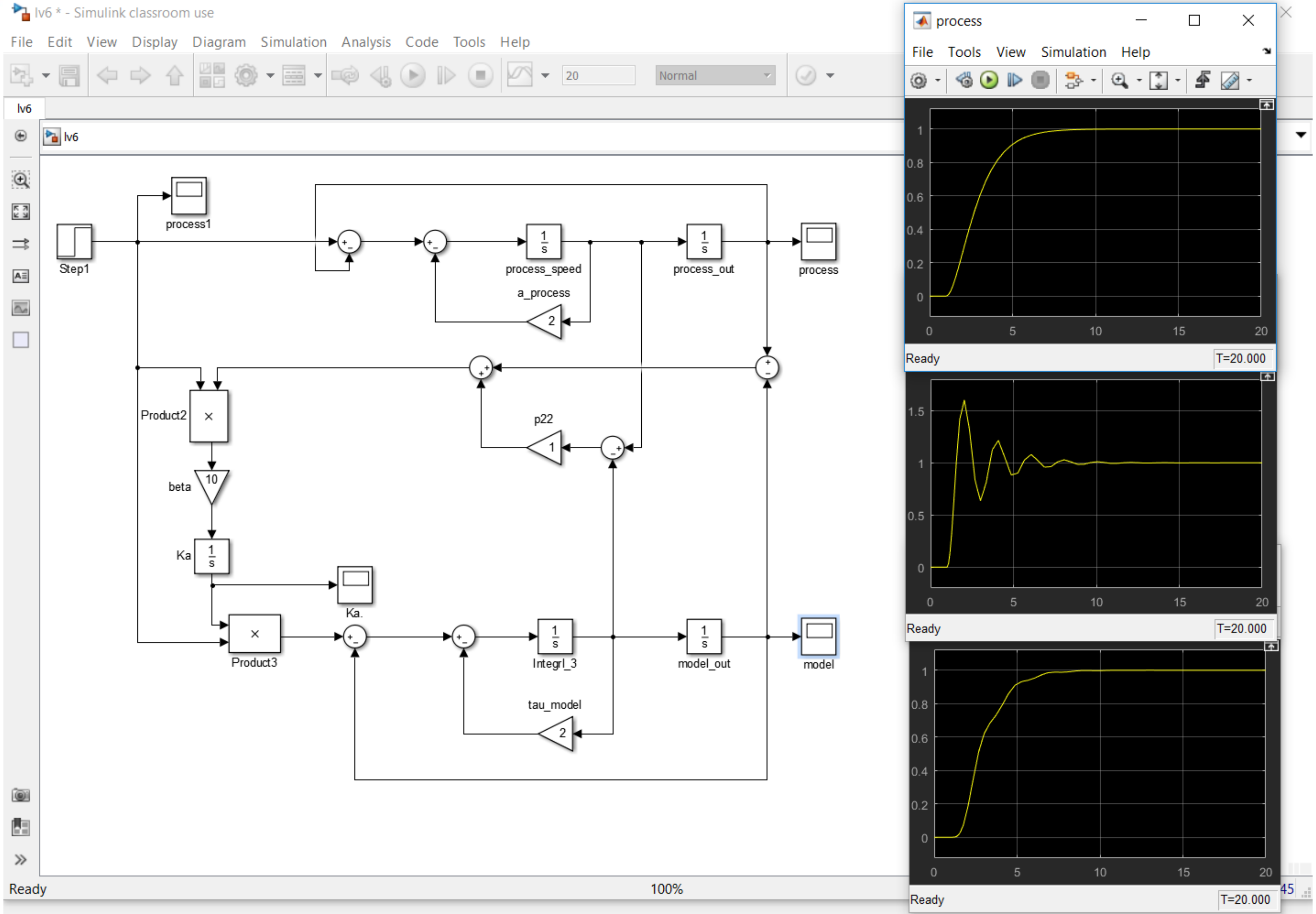
Erasmus+

e-Lab



Erasmus+

e-Lab Prof Eng



- For the example above draw a block diagram in Matlab/Simulnk. Simulation values must be recorded in the m file.
- Values are:

$\mu = 0.2$, $m1 = 8$ kg, $m2 = 4$ kg, $k = 4$ N/m, $g = 9.81$ m/s2

- where:

$c1 = \mu \cdot m1 \cdot g$ $c2 = \mu \cdot m2 \cdot g$