Master course: Numerical simulations of embedded systems

L1. Modeling with MATLAB/Simulink.

Sorin Lup, Ruxandra Bărbulescu

“Politehnica” University of Bucharest– PUB - CIEAC/LMN

http://www.lmn.pub.ro/

Contents

Task 0: Understanding your experience

Task 1: Hands-on Simulink tutorials

Task 2: Modeling a First Order System in Simulink

First-order model for car motion

First-order electrical circuits in dynamical/transient regime

Task 3: Modeling a Second Order System in Simulink

– Mass-Spring-Damper System

– Second-order electrical circuits in dynamical/transient regime

Total work time: 4h

Task 0: Understanding your experience

To understand what your experience is, we ask you to answer the following questions.

1. How do you appoint?

2. What degree did you graduate (faculty, department, year)?

3. Describe very briefly (2-3 phrases) the topic of the bachelor's thesis and your most important contribution in its realization.

4. Describe briefly (2-3 sentences) research theme selected in the first year of master.

5. Make a brief self-evaluation of the numerical modeling knowledge you have (theme, degree of comprehension).

6. To what extent do you master Matlab as a programming language? Do you have knowledge of Simulink?

7. Work?

8. If the answer to the previous question is yes, try to formulate (if possible) a problem related to the field you work likely to use modeling and numerical simulation.

9. If you work, please indicate your willingness to attend the NSES related teaching (Tuesday 16 to 20).

Task 1: Hands-on Simulink tutorials

➢ Download the Simulink tutorials from:

http://www.eng.ox.ac.uk/~labejp/Seminar/Simulink/Simulink_Introduction.pdf

http://eelabs.faculty.unlv.edu//docs/guides/Simulink_Basics_Tutorial.pdf

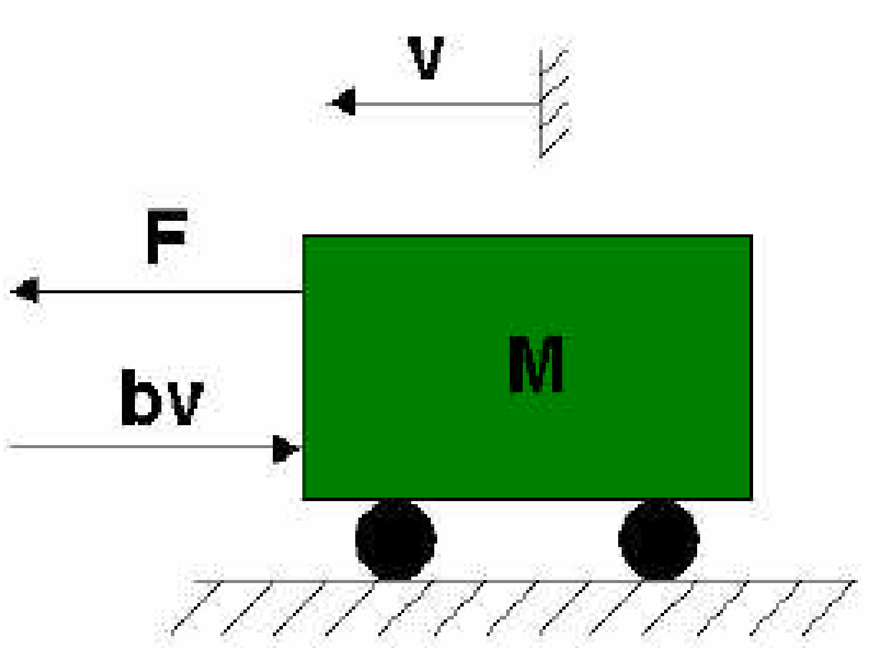
Browse the two tutorials and implement the exercises.

➢ Write a brief report adding the schematics and the results obtained. Finish each exercise with a short conclusion on the meaning of that specific exercise.

Task 2: Modeling a First Order System in Simulink

- **First-order model for car motion**

To investigate a real-world system, we will look at a simplified, first-order model of the motion of a car. If we assume the car to be traveling on a flat road, then the horizontal forces on the car can be represented as follows:



In this diagram:

- *v* [m/s] is the horizontal velocity of the car.

- *F* [N] is the force generated by the car's engine to propel it forward.

Task 2: Modeling a First Order System in Simulink

- *b* [N*s/m] is the damping coefficient for the car, which is dependent on wind resistance, wheel friction, etc. We assume the damping force to be proportional to the car's velocity.

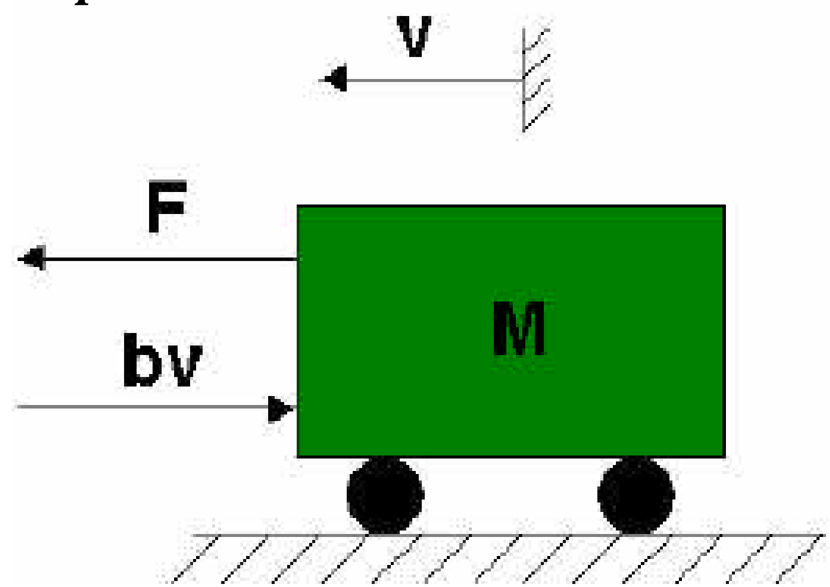
- *m* [kg] is the mass of the car.

Newton's Second Law for the horizontal direction thus gives:

$$M\frac{\mathrm{d}v}{\mathrm{d}t} + bv = F$$

For our system, we assume the numerical values:

- $m = 1000$ kg
- $b = 40$ N*s/m
- $F = 400$ N
- tmax = 200 [s]



Task 2: Modeling a First Order System in Simulink

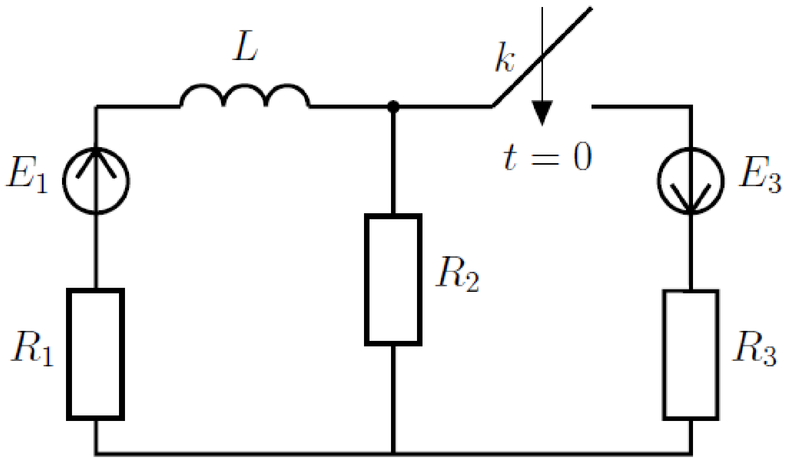
Tasks:

- (on paper) Solve analytically the differential equation
- (in Simulink) Create the model that solves the differential equation and represent graphically: the acceleration, the speed and the displacement;
- (in Matlab) Create a script where the model's parameters are set and parameterize the Simulink system;
- (in Matlab) add the commands to the script to compile the Simulink system;
- (in Matlab) solve the diff. equation with the specific Matlab functions (ode45, ode113);
- extract the acceleration, the speed and the displacement (from the Simulink system to Matlab);
- compare the results (Analytical vs. Matlab vs. Simulink);
- add to the report: your code, the Simulink diagram and the results.

Task 2: Modeling a First Order System in Simulink

- **First-order electrical circuits in dynamical/transient regime**

Consider the electrical circuit from the picture below. At the time moment $t = 0$, the switch k closes. You have to find the electric current that passes through the inductor and the electric voltage.



$E_1 = 14$ V
$E_3 = 8$ V
$R_1 = 2\ \Omega$
$R_2 = 5\ \Omega$
$R_3 = 4\ \Omega$
$L = 0.1$ mH

If the circuit is equated around the inductor and apply Kirchhoff's voltages theorem on the formed loop the differential equation is:

$$\frac{di_L(t)}{dt} = -\frac{R_e}{L} i_L(t) + \frac{E_e}{L}$$

Task 2: Modeling a First Order System in Simulink

The analytical solution is:

$$i_L(t) = (i_{L0} - i_{L\infty}) \cdot e^{-\frac{t}{\tau}} + i_{L\infty}\ [\text{A}]$$
$$u_L(t) = -L \cdot (i_{L0} - i_{L\infty}) \cdot \frac{t}{\tau} \cdot e^{-\frac{t}{\tau}}\ [\text{V}]$$

were:

- o R_e= equivalent resistance
- o E_e= equivalent voltage source
- o i_{L0} = initial condition
- o i_{Linf} = final condition
- o $\tau = \frac{L}{R_e}$ $[s]$ electric circuit time constant

Task 2: Modeling a First Order System in Simulink

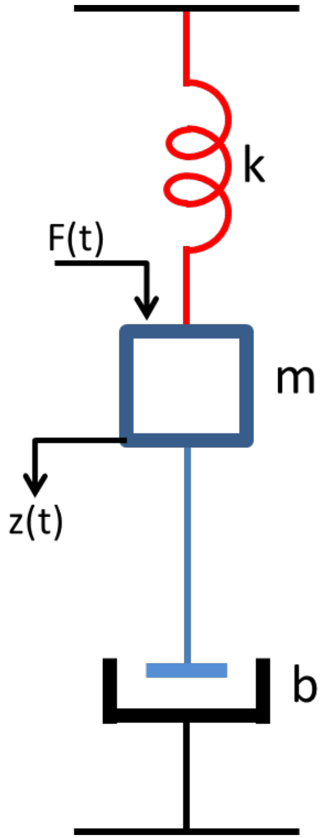
Tasks:

- (in Matlab) Create a script where the model's parameters are set and represent graphically the analytical solution;
- (in Matlab) solve the diff. equation with the specific Matlab functions (ode45, ode113);
- (in Matlab) add the commands to the script to compile the Simulink system;
- (in Simulink) Create the parameterized model that solves the differential equation and represent graphically the inductor's *i* and *u*;
- extract *i* and *u* (from the Simulink system to Matlab);
- compare the results (Matlab vs. Simulink);
- add to the report: your code, the Simulink diagram and the results.

Task 3: Modeling a Second Order System in Simulink

- **Mass-Spring-Damper System**

In this exercise you will build a simulation diagram that represents the behavior of a dynamical system. You will simulate a spring-mass damper system.



Task 3: Modeling a Second Order System in Simulink

The equation that describes the system is:

$$m\ddot{z}(t) + b\dot{z}(t) + kz(t) = F(t)$$

where *t* is the simulation time, *F(t)* is an external force applied to the system, b is the damping constant of the spring, k is the stiffness of the spring, m is a mass, and *z(t)* is the position of the mass. $\dot{z}$ is the first derivative of the position, which equals the velocity of the mass. $\ddot{z}$ is the second derivative of the position, which equals the acceleration of the mass.

The goal is to view the position *z(t)* of the mass *m* with respect to time *t*. You can calculate the position by integrating the velocity of the mass. You can calculate the velocity by integrating the acceleration of the mass. If you know the force and mass, you can calculate this acceleration by using Newton's Second Law of Motion: $F = m \cdot a$.

Task 3: Modeling a Second Order System in Simulink

Substituting terms from the differential equation above yields the following equation:

$$\ddot{z}(t) += \frac{1}{m}(F(t) - b\dot{z}(t) - kz(t))$$

The model's parameters are:

- x_init = 4 [m] – initial position.
- dxdt_init = 0 [m/s] – initial Speed.
- m = 20 [kg] – mass
- b = 4 [N/(m/s)] – damping coefficient
- k = 2; % [N/m] – elastic coefficient
- t_step_F = 50 [s] – time delay
- F_0 = 0 [N] – initial force
- F_1 = 4 [N] – final force

Task 3: Modeling a Second Order System in Simulink

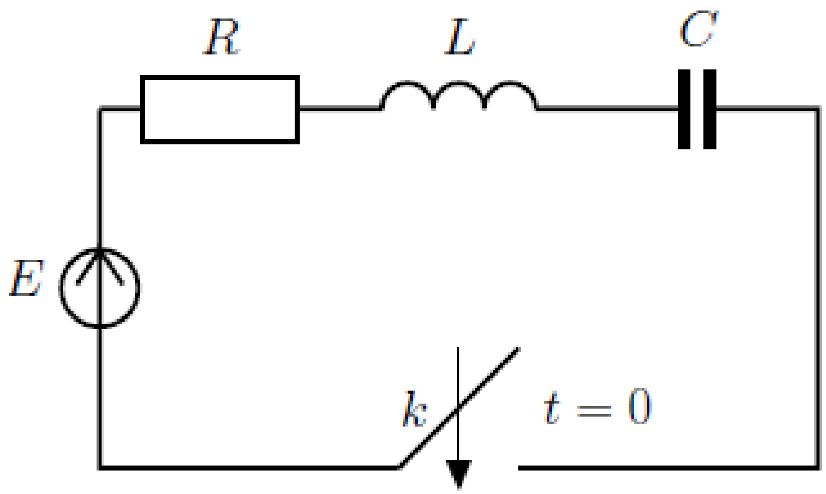
Tasks:

- (in Simulink) Create the model that solves the differential equation and represent graphically: the acceleration, the speed and the displacement;
- (in Matlab) Create a script where the model's parameters are set and parameterize the Simulink system;
- (in Matlab) add the commands to the script to compile the Simulink system;
- (in Matlab) solve the differential equation with the specific Matlab functions (ode45, ode113);
- extract the acceleration, the speed and the displacement (from the Simulink system to Matlab);
- compare the results (Matlab vs. Simulink);
- add to the report: your code, the Simulink diagram and the results.

Task 3: Modeling a Second Order System in Simulink

- **Second-order transient electrical circuits**

Consider the electrical circuit from the picture below. At the time moment $t = 0$, the switch k closes. You have to find the electric current that passes through the inductor and the electric voltage from the capacitor's terminals.



$$R = 3\ \Omega$$
$$L = 1\ mH$$
$$C = 1 \mu F$$
$$E = 12\ V$$

Tasks:

- (on paper) Determine the second order differential equation.

Task 3: Modeling a Second Order System in Simulink

- (in Matlab) Create a script where the model's parameters are set;
- (in Matlab) solve the diff. equation with the specific Matlab functions (ode45, ode113);
- (in Simulink) Create the model that solves the differential equation and represent graphically the inductor's inductor's i_L and capacitor's u_C;
- (in Matlab) add the commands to the script to compile the Simulink system;
- extract the inductor's i_L and capacitor's u_C (from the Simulink system to Matlab);
- represent graphically i_L vs. u_C
- change the value of R to 62 Ω and 1kΩ and repeat the simulations
- compare the results;
- add to the report: your code, the Simulink diagram and the results.

Template of the Lab report (.pdf)

- Title page
  - Header:
    - Polytechnic University of Bucharest,
    - Electrical Engineering Department,
    - Master Course: Numerical Simulation of Embedded Systems,
  - Title:
    - Lab1 Modeling with MATLAB/Simulink,
  - Author: name,
  - Date: 25th September 2018
- Contents (automatic generated)
  - 1. Introduction (Abstract)
  - 2. ………
  - ………
  - Conclusions
  - References