5. Implementing and simulating a system of ODEs in Simulink

Hande Bozkurt, Daria Semenova and Krist V. Gernaey

March 4th 2015

Course 28864: Introduction to Matlab programming

During the 3rd lecture, you learned how to implement and solve a system of ordinary differential equations (ODEs) using m-files (script and function). The purpose of this exercise is to get familiar with Simulink as a modeling tool that can be used for numerical integration of systems of differential equations. In the first part, a simple example will be provided so you can follow the general principles of implementing a system of ODEs in Simulink. Then, in the second part of the exercise, you have to implement and simulate the reactions in a CSTR from Exercise 3 using Simulink.

At the end of this exercise you should be able to implement and simulate a process model in Matlab using Simulink and link the model with other functions made in previous lectures.

1. Example code: A reversible reaction

This example contains two different structures of implementing process models using Simulink. Therefore, two folders are supplied (Simulink and Simulink_m-file). The first one contains two files, one is the script isomerpar_simulink_call.m, and the other one is the model implemented in Simulink isomerpar_simulink.mdl.

The Simulink file contains a schematic drawing consisting of Simulink model blocks, and corresponding to the differential equations needed to describe the dynamics of the concentrations of compounds A and B involved in a reversible reaction, thus

$$A \underset{\text{k2}}{\overset{\text{k1}}{\rightleftarrows}} B$$

The model for this reaction, consisting of 2 differential equations, can be written as:

$$\frac{dA}{dt} = -k_1 A + k_2 B \quad (1) \qquad\qquad \frac{dB}{dt} = k_1 A - k_2 B \quad (2)$$

In this equation, k_1 and k_2 represent the kinetic parameters of the reaction. The script isomerpar_simulink_call.m is a call file that contains parameter values (k_1 = 1 h^{-1}, k_2 = 2 h^{-1}), initial values for the state variables A and B in the model (a = 5 mol/L, and b= 1 mol/L), and start time and end time for the simulation. When you run this script, the model in isomerpar_simulink.mdl will be simulated. Finally, results are plotted.

The second folder (Simulink_m-file) contains three files that simulate exactly the same system. The folder contains one script isomerpar_simulink_call.m, one S-function sfun_isomerpar.m, and the model implemented in Simulink isomerpar_simulink_mSfunc.mdl. Note that in the last file only one block is used to call the S-function that contains the set of ODEs.

2. Reactions in a CSTR

In this exercise you will implement the CSTR system that you also solved in Exercise 3, but this time you will implement the differential equations in Simulink.
The system should be solved, and the results should be plotted.

In the script
The script must be able to: i) solve the set of differential equations (*see Simulink model*); and ii) plot the results of the simulation as required (*see Plotting function below*).

1. Specify the model parameter values and simulation information. The simulation of the model is to be done for the following conditions:
   - *Model parameters*:

Parameter	**Value**	**Unit**
k_1	$1.25*10^{-2}$	min^{-1}
k_2	$7.5*10^{-3}$	min^{-1}
k_3	0.265	$m^3 mol^{-1} min^{-1}$
k_4	0.15	$m^6 mol^{-2} min^{-1}$
Volume	800	L
q	0.032	$L sec^{-1}$
$C_{A,in}$	0.012	$mol L^{-1}$
$C_{D,in}$	0.020	$mol L^{-1}$

   - *Initial values/conditions:* Initial conditions were set to 1.252 mol m^{-3} for *A*, 3.131 mol m^{-3} for *B* and 5.216 mol m^{-3} for *I*. The initial concentrations of all other components are set to 0.
   - *Simulation time*: define the start/end time, from zero to 2000 [min] with 1 min intervals.

2. Solve the set of differential equations using the parameters and initial conditions given above. At this point, you must call the **Simulink file** that contains the model. It means that you have to include the correct Matlab syntax for calling Simulink files, as learnt in the lecture and the above example. If you follow the examples provided, solving the differential equations will yield an output e.g. [t y], where *t* is a vector of time values and *y* is a matrix with 6 columns that contain the corresponding states/variables of the system (in this case A, B, D, I, P and S) for each time point. Note that the order of the variables in *y* depends on the order of the differential equations that you implement in the function. For further calculations, you can give a proper name to the ODE solution, if needed, e.g. A= y (: , 1).

3. Finally, simulation results are required to be plotted. You can reuse some of the coding from the plotting function you made in exercise 2.

4. To finalize this exercise, the figure must be saved into a file (.jpg format).

Note that this script has a similar structure as the script built in exercise 3. Thus, you can take that script as starting point of the new script and modify it where required.

The Simulink model
You only need to implement the model following ONE of the Simulink implementation methods. In the first implementation method the model is drawn in the Simulink window, and

in the second implementation method the model is written in an m-file S-function. We recommend the second implementation method since for the first “spaghetti” method it will become very difficult to keep an overview of the code, for example when implementing more than two ODEs.

The function takes as inputs: the simulation time (e.g. t0 and tf), initial conditions (e.g. x0), and parameters. The four reactions taking place in the reactor together with their rate expressions are given below.

(1)	$A \rightarrow I$	$r_1 = k_1C_A$
(2)	$A \rightarrow B$	$r_2 = k_2C_A$
(3)	$I + D \rightarrow P$	$r_3 = k_3C_IC_D$
(4)	$B + D \rightarrow S$	$r_4 = k_4C_DC_B^2$

Where;

$k_1 = 1.25*10^{-2}$	(min^{-1})
$k_2 = 7.5*10^{-3}$	(min^{-1})
$k_3 = 0.265$	($m^3\ mol^{-1}\ min^{-1}$)
$k_4 = 0.15$	($m^6\ mol^{-2}\ min^{-1}$)

The following assumptions are made: (1) the CSTR is perfectly mixed, (2) the mass densities of the feed and product streams are equal and constant, (3) the liquid volume, *V* is constant. The mass balance equation for the component *j* can be written as:

$$V.\frac{dC_j}{dt} = q\left(C_{j,in} - C_j\right) + Vr_j$$

Where;
C_j the concentration of component j in the reactor
$C_{j,in}$ the concentration of component j in the inflow
q the flowrate of the inflow and outflow
r the net reaction rate of component j
V the volume of the reactor

In the Plotting function

The plotting function will generate one figure, containing 6 subplots, but you only need to plot the simulation results. The required function inputs are: i) the solution obtained from the Simulink file; and ii) text strings (Axis titles and so on). The figure should be formatted, so it shows the simulation result as dashed blue lines.

Handing in:

For the exercise described under point 2 (the CSTR system), you hand in the script, the Simulink model file and the plotting function. If you implement the Simulink model using the second implementation method, you must also submit the *S-function*. The Matlab files needed for this assignment should be sent as e-mail attachments to kvg@kt.dtu.dk before Wednesday March 11th 2015 at 4 pm!

In the email, you clearly list the names and student numbers of the members of your group (maximum 2 persons per group).

To allow automatic sorting of the emails containing your solutions, the subject line of the email should be: 28864-F2015-E5

For help or questions: dsem@kt.dtu.dk or kvg@kt.dtu.dk