5 Simulink: Basics

5.1 Generating a Simulink model

a) Generate a new Simulink model `simulink_basics_ex1.mdl`. Use the default values and parameters of Simulink.

b) Program the logical function NOT(NOT(`A`) OR NOT(`B`)) using appropriate Simulink blocks.

c) Input signals `A` and `B` are two column vectors `[ 0 0 1 1 ]`^{T} and `[ 0 1 0 1 ]`^{T}, respectively. Generate two appropriate input blocks for these constant column vectors of type logical (no definition of `A` and `B` in the MATLAB workspace).

d) Generate a output block as display showing the result of the generated logical function for each row of the column vector.

5.2 Subsystems and random signal as input signal

5.2.1 Logical function as subsystem

a) Save Simulink model `simulink_basics_ex1.mdl` as new Simulink model `simulink_basics_ex2.mdl`.

b) The above logical function NOT(NOT(`A`) OR NOT(`B`)) should be grouped to a Subsystem.

Generate a subsystem with two inputs *A* and *B* and output *y*. The subsystem should be named *Logische function 1* and the block should display *NOT(NOT(A) OR NOT(B))*.

5.2.2 Random signal as input signal

In contrast to exercise 5.1 input signals *A* and *B* for subsystem *Logical function 1* should be randomized time signals.

a) Delete the input blocks for the constant column vectors from exercise 5.1.

b) Generate two differen randomized input signals from appropriate Simulink blocks. Such blocks output values between 0 and 1, so their output signals have to be adopted:

c) Programm the following equation using appropriate Simulink blocks

$$y = 0.5 \cdot (\text{signum}(x) + 1) \qquad \text{with} \quad \text{signum}(x) = \begin{cases} 1 & \text{for } x > 0 \\ 0 & \text{for } x = 0 \\ -1 & \text{for } x < 0 \end{cases} \tag{1}$$

x is the output of the random blocks and y the input of the subsystem *Logical function 1*.

d) Replace the former display output blocks by an appropriate sink block showing time signals with three axis for input signals *A* and *B* and output signal y of subsystems *Logical function 1*. The x-axis should be as long as the complete simulation time.

5.3 Subsystem Scaling

a) Save Simulink model `simulink_basics_ex2.mdl` as new Simulink model `simulink_basics_ex3.mdl`.

b) Programm equation (1) from exercise 5.2.2 as separate subsystem *scaling*. Input signal is the output of the respective random block, the output signal serves as input *A* and *B*, respectively, for subsystem *Logical function 1*.

c) Instead of the two subsystems *scaling* for each input of the subsystems *Logical Function 1*) just use a single subsystem *scaling*, but still two randomized signals from two different parameterised random blocks as input signals for the subsystem *scaling*.

5.4 Subsystem and random signal as input signal

Input signals *A* and *B* for the subsystem *Logical Function 1* are values of variables `siminA` and `siminB` in the MATLAB workspace in the Simulink model `simulink_basics_ex4.mdl` eingelesen werden.

5.4.1 MATLAB initialization file `simulink_basics_ex4_ini.m`

a) Generate MATLAB file `simulink_basics_ex4_ini.m` and set the simulation stop variable `Tstop` to 10 seconds and sample time `Tsample` to 0.1 seconds.

b) Define a column vector `T` from 0 to `Tstop` and step size `Tsample` in MATLAB file `simulink_basics_ex4_ini.m`.

c) Generate MATLAB variables `siminA` and `siminB` in MATLAB file `simulink_basics_ex4_ini.m`. Both MATLAB variables are matices with column vector `T` as first column and the values of the function mod$(T, 2) > 1$ and mod$(T, 2) > 1.5$, respectively, as second column.

d) Show the values of the second column of `siminA` and `siminB`, respectively, in two subplots one above the other of a MATLAB-Figure.

5.4.2 Simulink model `simulink_basics_ex4.mdl`

a) Save Simulink model `simulink_basics_ex2.mdl` as new Simulink model `simulink_basics_ex4.mdl`.

b) Delete the random blocks and the blocks for the equation (1) from exercise 5.2.2.

c) Choose appropriate Simulink blocks for reading MATLAB variables `siminA` and `siminB`, respectively as *A* and *B*, respectively for the subsystem *Logical Function 1* and insert the respective MATLAB variables `siminA` and `siminB`.

d) Assign MATLAB variable `Tstop` as configuration parameter for the stop time of Simulink model `simulink_basics_ex4.mdl`.

e) Choose an appropriate fixed step solver for the simulation and assign MATLAB variable `Tsample` to the sample time of the chosen solver.

f) Run the Simulink model `simulink_basics_ex4.mdl` and monitor the respective values shown in the Display-Block.

Don't forget the initialization of the configuration parameters and the input values `siminA` and `siminB` before starting the simulation. What happens if these values are not initialized before?

5.5 Simulation from MATLAB workspace

Simulink model `simulink_basics_ex5.mdl` should be configured and automatically started from MATLAB workspace.

5.5.1 Simulink model `simulink_basics_ex5.mdl`

a) Save Simulink model `simulink_basics_ex4.mdl` as new Simulink model `simulink_basics_ex5.mdl`.

b) Set stopp time configuration parameter of the Simulink model `simulink_basics_ex5.mdl` to 10 seconds.

c) Set the solver of the Simulink model to a variable step solver and the maximum sampling time of the solver to 0.2 seconds.

d) The simulation date shown in the scope block should additionally be save to MATLAB workspace in a "Structure with time" variable `ScopeData`.

e) Save the Simulink model again.

5.5.2 MATLAB initialization file `simulink_basics_ex5_ini.m`

a) Copy MATLAB file `simulink_basics_ex4_ini.m` to `simulink_basics_ex5_ini.m`.

b) In MATLAB file `simulink_basics_ex5_ini.m` set the simulation stop variable `Tstop` to 8 seconds and sample time `Tsample` to 0.01 seconds.

c) Keep the definitions of `T`, `siminA` and `siminB`.

d) Save the file again.

5.5.3 Simulation of Simulink model using MATLAB simulation file `simulink_basics_ex5_sim.m`

a) Generate a MATLAB file `simulink_basics_ex5_sim.m`.

b) Call the initialization file `simulink_basics_ex5_ini.m` from the MATLAB file `simulink_basics_ex5_sim.m`.

c) In the MATLAB file `simulink_basics_ex5_sim.m` assign the current configuration parameters of Simulink model `simulink_basics_ex5.mdl` to MATLAB variable `oldoptions`.

d) In the MATLAB file `simulink_basics_ex5_sim.m` assign the configuration parameters saved in MATLAB variable `oldoptions` to MATLAB variable `newoptions` and the Fixed-Step-Solver `'ode3'` to Parameter `'Solver'` and MATLAB variable `Tsample` to Parameter `'FixedStep'`.

e) Start the simulation of Simulink model `simulink_basics_ex5.mdl` from the MATLAB file `simulink_basics_ex5_sim.m` with stop time `Tstop` and configurations parameters saved in `newoptions`.

f) After the simulation the values saved to variable `ScopeData` should be displayed in three subplots one above the other of a MATLAB-Figure.

Generate a MATLAB file `simulink_basics_ex5_plot.m` and programm the plot commands in this file. This file should be automatically started from MATLAB file `simulink_basics_ex5_sim.m` after the end of the simulation of Simulink model `simulink_basics_ex5.mdl`.

5.6 Simulation from MATLAB workspace

Simulink model `simulink_basics_ex6.mdl` should automatically load its configuration parameters in starting and automatically plot the result of the simulation in a MATLAB figure after the end of the simulation.

5.6.1 Simulink model `simulink_basics_ex6.mdl`

a) Save Simulink model `simulink_basics_ex5.mdl` as new Simulink model `simulink_basics_ex6.mdl`.

b) Assign MATLAB variable `Tstop` as configuration parameter for the stop time of Simulink model `simulink_basics_ex6.mdl`.

c) Choose an appropriate fixed step solver for the simulation and assign MATLAB variable `Tsample` to the sample time of the chosen solver.

d) The simulation date shown in the scope block should additionally be save to MATLAB workspace in a "Structure with time" variable `ScopeData`.

e) Assign initialization file `simulink_basics_ex5_ini.m` and plot file `simulink_basics_ex5_ini.m` to the respective Callback parameters in model properties pane of Simulink model `simulink_basics_ex6.mdl`.