Simulation Module in LabVIEW

Simulation Loop

Roland Pawliczek

References

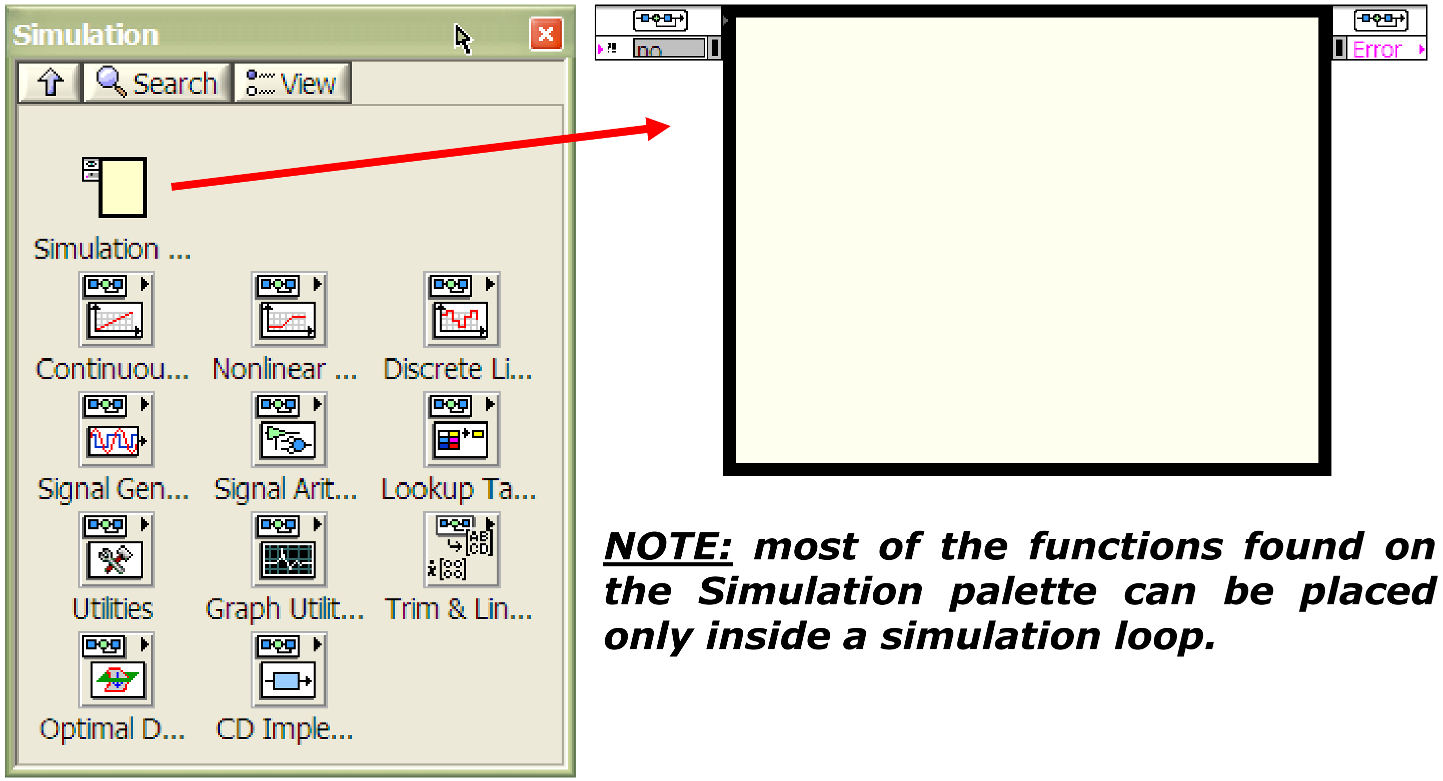
- ***NI LabVIEW, Simulation Module User Manual***
- *Finn Haugen, TechTeach, Tutorial for LabVIEW Simulation Module*

LabVIEW Simulation Module

- *Use the LabVIEW Simulation Module to build a simulation diagram, which graphically displays a simulation model in LabVIEW.*
- *You can build and execute a simulation diagram using the Simulation Loop, Simulation functions, and other LabVIEW VIs and structures.*
- *The simulation diagram uses an ordinary differential equation (ODE) solver to compute the behavior of a simulation model.*
- *You can make the simulation run as fast as the computer allows, or you can make it run with a real or scaled time axis, thus simulating real-time behavior, with the possibility of the user to interact with the simulated process*

LabVIEW Simulation Module

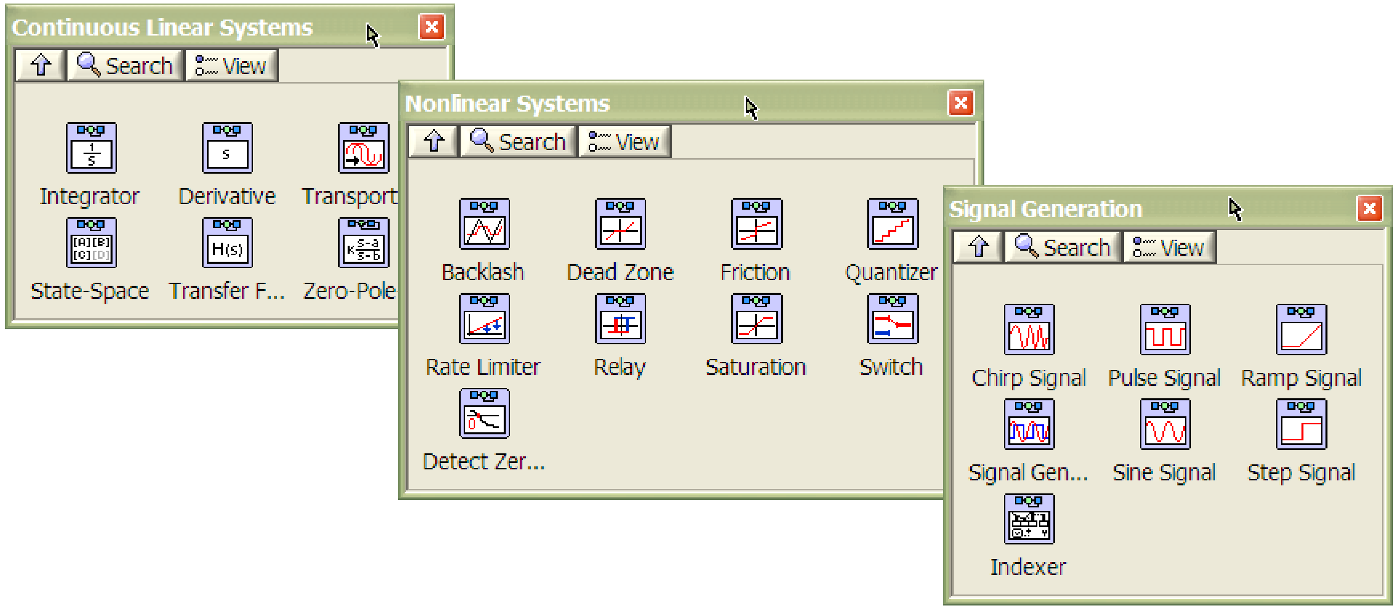
Simulation Module can be found in Function Palette/Control Design & Simulation. The main element in that window is Simulation Loop option which creates a basic structure for simulation.



<u>NOTE:</u> most of the functions found on the Simulation palette can be placed only inside a simulation loop.

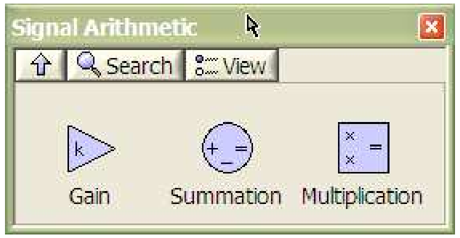
LabVIEW Simulation Module – Functions Palettes

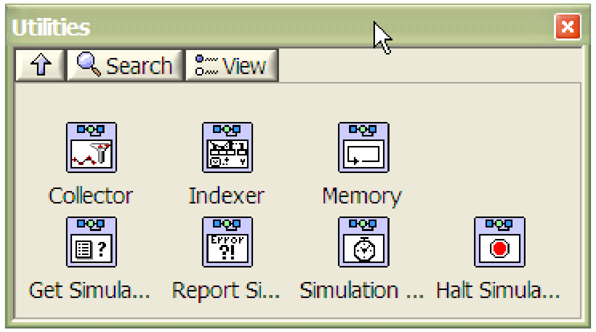
- *Use the **Continuous Linear Systems** functions to **represent** continuous linear systems of differential equations on the simulation diagram.*
- *Use the **Nonlinear Systems** functions to **simulate** nonlinear effects that are present in real-world systems.*
- *Use the **Signal Generation** functions to **generate** specific waveform patterns and to index signals into an array.*

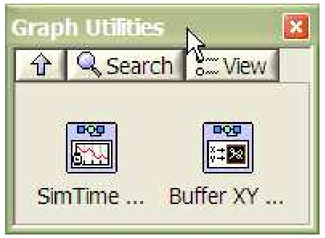


LabVIEW Simulation Module – Functions Palettes

- *Use the **Signal Arithmetic** functions to perform basic arithmetic operations on signals in a simulation system (e.g. gain the signal)*
- *Use the **Utilities** functions to **perform** various tasks such as creating a history of signal values and the times at which they were recorded, indexing a signal by simulation time, reporting simulation time, reporting simulation errors, stopping the simulation programmatically, and returning the simulation parameters.*
- *Use the **Graph Utilities** functions to **display** the outputs of the simulation on a chart or graph.*

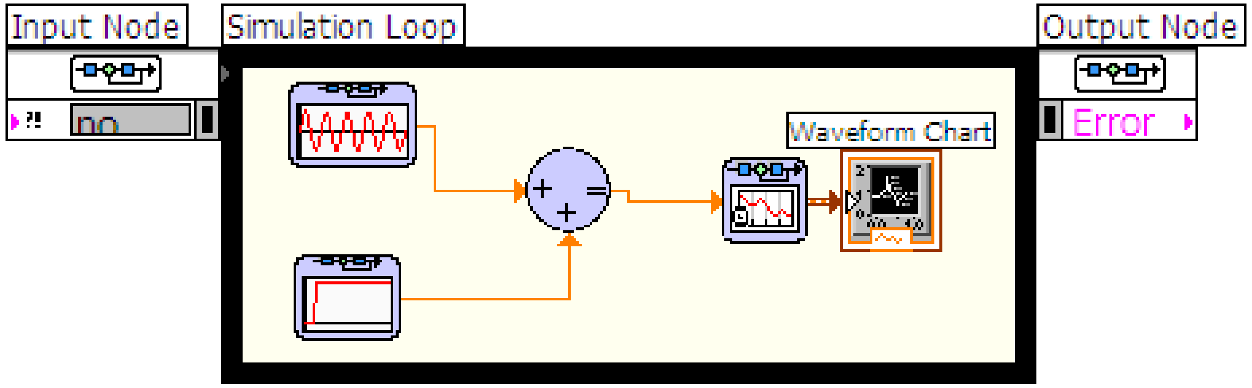






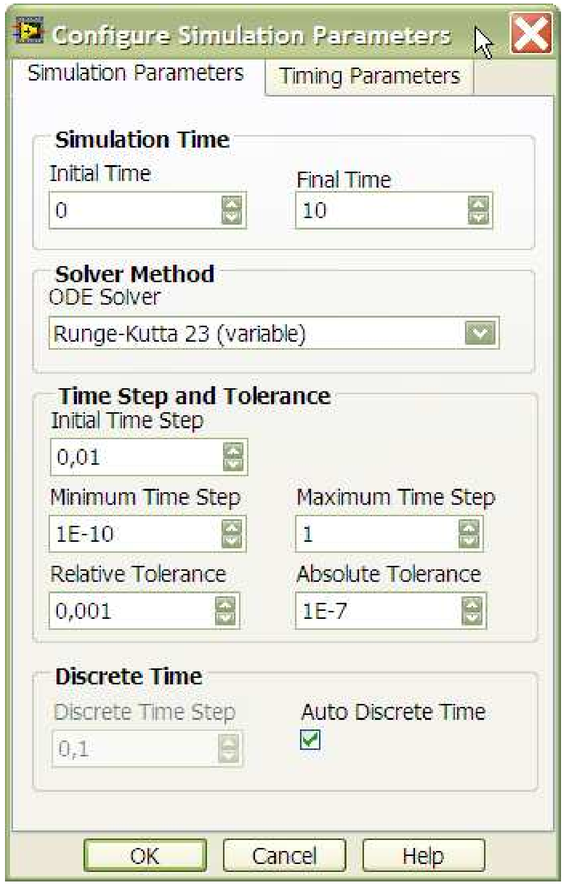
Simulation Loop

- *The **Simulation Loop** forms the boundary of the simulation diagram and contains the parameters that define how the simulation diagram executes.*
- *This loop has **Input Node** and **Output Node** to control the execution of the loop. Use the **Input Node** to configure simulation parameters programmatically. You also can configure these parameters interactively using the **Configure Simulation Parameters dialog box**. Access this dialog box by double-clicking the Input Node or by right-clicking the border and selecting Configure Simulation Parameters from the shortcut menu.*



Simulation Loop

- ***Configure Simulation Parameters dialog box.***


Configure Simulation Parameters

Simulation Parameters | Timing Parameters

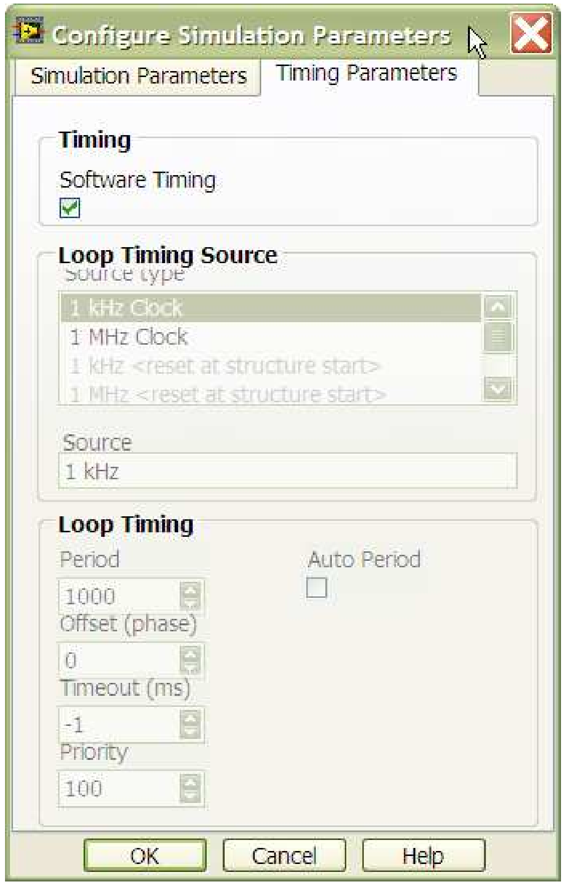
Simulation Time
Initial Time: 0
Final Time: 10

Solver Method
ODE Solver: Runge-Kutta 23 (variable)

Time Step and Tolerance
Initial Time Step: 0,01
Minimum Time Step: 1E-10
Maximum Time Step: 1
Relative Tolerance: 0,001
Absolute Tolerance: 1E-7

Discrete Time
Discrete Time Step: 0,1
Auto Discrete Time: ✓

OK | Cancel | Help


Configure Simulation Parameters

Simulation Parameters | Timing Parameters

Timing
Software Timing: ✓

Loop Timing Source
Source type
1 kHz Clock
1 MHz Clock
1 kHz <reset at structure start>
1 MHz <reset at structure start>

Source: 1 kHz

Loop Timing
Period: 1000
Auto Period: ☐
Offset (phase): 0
Timeout (ms): -1
Priority: 100

OK | Cancel | Help

Simulation Loop

- *The **Initial Time** defines the initial simulation time, typically zero.*
- *The **Final Time** defines the simulation time when the simulation will stop. In our VI the final time is set to Inf (infinity) it will stop, due to the Halt Simulation function in the block diagram. When the user clicks the Stop button on the front panel the Halt Simulation function will cause the simulator to stop.*
- *The **Time Step** is the resolution of the simulation time scale.*

  In general, the smaller time step the better accuracy, but the larger the total number of calculations needed to calculate the simulated response for a given simulation time range.

  The main rule of selecting the time step is: Select the largest time step that does not influence the accuracy of the simulated response. You may iterate to find this largest time step. Start with some initial guess of the time step, and try increased values until you observe that the response is influenced by the time step.

  *As an **initial guess** you may select the time step as* $\boldsymbol{T_s = 0.1 * T_{min}}$ *where* T_{min} *is the smallest time constant of the model.*

  If You do not know the smallest time constant just set h = 0.05s which give a fairly smooth update of the plotted simulation response on the screen.

Simulation Loop

- *The **ODE Solver** is the numerical method used to calculate - or solve for - the values of the state variables of the model.*

  The Runge-Kutta23 third order method (with a variable-step time) is set as default.

  However some people suggests to use the Runge-Kutta second order method (with a fixed time step), which works fine in most situations.

1. *Runge-Kutta 1 (Euler), 2, 3, 4—A fixed step-size, single-step explicit Runge-Kutta ODE solver of first to fourth order.*
2. *Runge-Kutta 23—A variable step-size, single-step explicit Runge-Kutta ODE solver of third order.*
3. *Runge-Kutta 45—A variable step-size, single-step explicit Runge-Kutta ODE solver of fifth order, which uses the Dormand-Prince coefficients.*
4. *BDF—A variable step-size, variable order (orders 1 through 5) implementation of the multi-step backwards difference formula (BDF), also known as Gear's Method. This method is adequate for moderately stiff problems.*
5. *Adams-Moulton—A variable step-size, multi-step variable order (orders 1 through 12) implementation of the Adams-Moulton predictor-corrector pair in predict-evaluate-correct-evaluate (PECE) mode.*
6. *Rosenbrock—A variable step-size, single-step explicit solver. This method is adequate for some stiff problems.*
7. *Discrete States Only—A fixed step-size solver. Use this ODE solver for simulations that do not contain any continuous functions.*

Simulation Loop

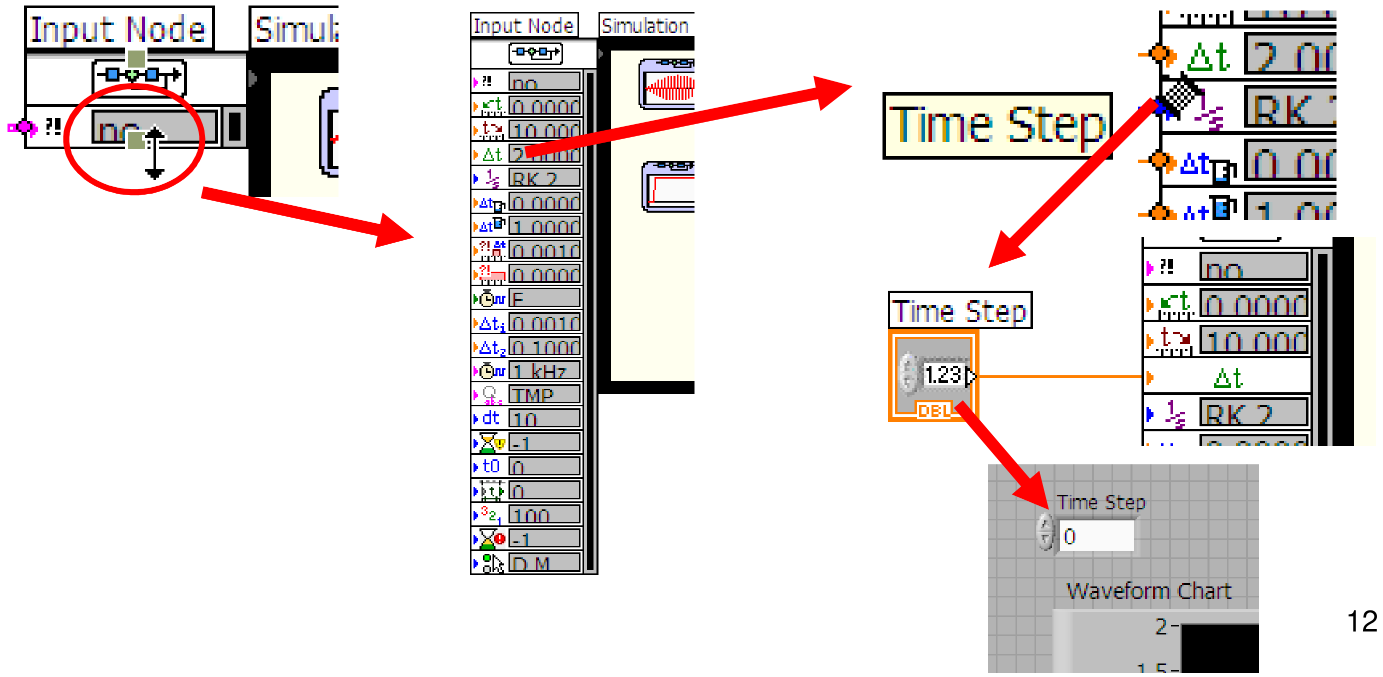
- *The **Discrete Time Step** is the time step used for simulating discrete-time functions. The overall discrete time is based on the **sample period (sec)** and **sample skew (sec)** of each discrete function on the simulation diagram. If you select a fixed step-size solver, the **Discrete Time Step** <u>**must be an integer multiple**</u> of the **Time Step**. **Auto Discrete Time** automatically calculates the **Discrete Time Step**.*
- *Check **Software Timing** parameter in **Timing Parameters** tab to run the simulation as fast as it can, i.e. the simulation time is not equal to the real time. Uncheck **Software Timing** gives you the possibility to set how fast the simulation will run.*
- *The **Period** is the amount of real time between two subsequent simulated time points.*
- *By setting the **Period** equal to the simulation time step the simulation runs in real time.*
- *By giving the **Period** some other value, the simulation time scale is proportional to real time. For example, if the simulation time step is 0.05s, setting Period equal to 0.01s causes the simulation to run 5 times faster than real time. Note that on a PC Period is the number of milliseconds, since the PC clock runs with a frequency of 1kHz. For example, Period = 10 corresponds to Period equal to 0.01s.*

Simulation Loop

Changing parameters programmatically.

*All parameters for **Simulation Loop** can be changed using **Front Panel**. For this purpose unfold **Input Node** and **create control** for parameters, which you want to control on **Front Panel**.*

*Using **Wiring tool** point selected parameter, right-click on it and select **Create/Control** from context menu.*



Simulation Loop

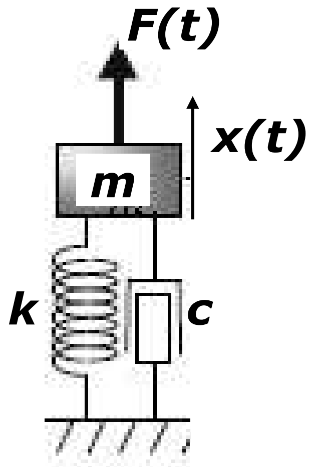
How the Simulation module works

Immediately before a simulation is started, LabVIEW transforms the block diagram model to a model consisting of differential equations.

Then the selected numerical solver method is applied to calculate the responses. The model transformation takes some time. Therefore the simulation does not start immediately after the Run button is pressed, but after some time depending on the complexity of the model.

Simulation Loop – exercises

VALVE



F – spring force
m – moving mass
k – spring constant
c – damping
x – displacement

$$m\frac{d^2x}{dt^2} + c\frac{dx}{dt} + kx = F(t)$$

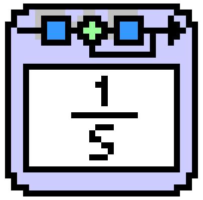
Simulation Loop – exercise 1

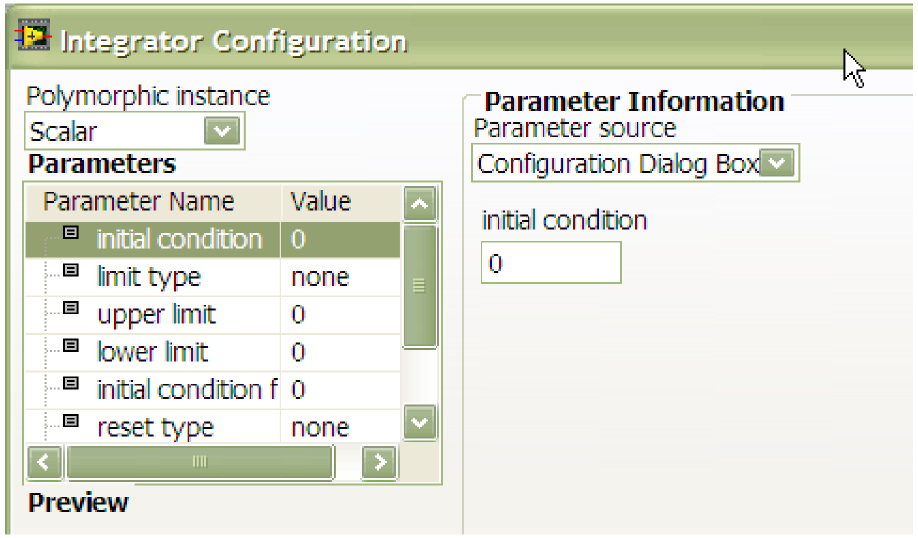
Differential equation model

$$\frac{d^2x}{dt^2} = \frac{1}{m}F(t) - \frac{c}{m}\frac{dx}{dt} - \frac{k}{m}x$$

Because the displacement is important for this object it is necessary to integrate the equation.

*For this purpose the **Integrator Function** from **Continuous Linear System** function palette can be used. Double click on the integration icon to open configuration window.*





Integrator Configuration

Polymorphic instance: Scalar

Parameters

Parameter Name	Value
initial condition	0
limit type	none
upper limit	0
lower limit	0
initial condition f	0
reset type	none

Preview

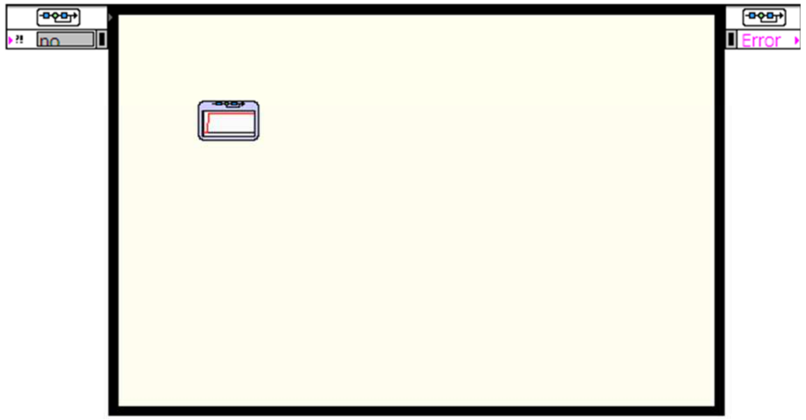
Parameter Information

Parameter source: Configuration Dialog Box

initial condition: 0

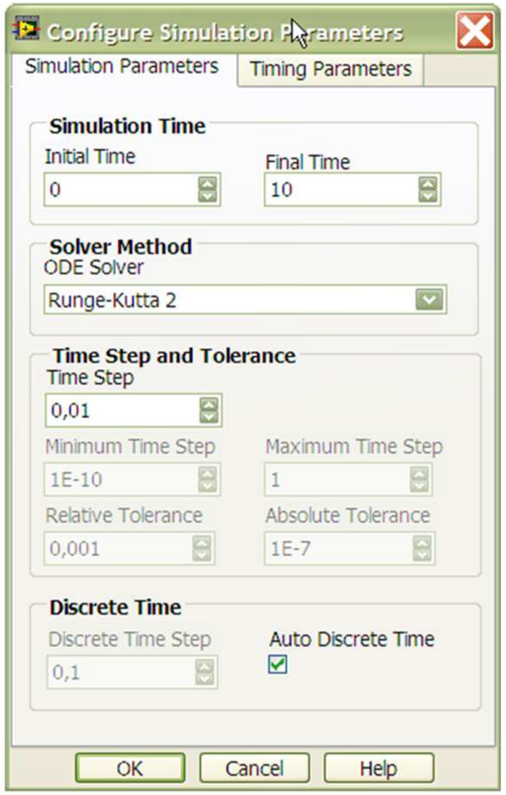
Simulation Loop – exercise 1

To build simulation for differential equation model first the ***Simulation Loop*** *must be created.*
Because the step response of the system will be investigated place into the loop the ***Step Signal*** *function from* ***Signal Generation*** *palette.*



Simulation Loop – exercise 1

Set parameters of the simulation loop as presented.



Simulation Time

Initial Time: 0

Final Time: 10

Solver Method

ODE Solver: Runge-Kutta 2

Time Step and Tolerance

Time Step: 0,01

Minimum Time Step: 1E-10

Maximum Time Step: 1

Relative Tolerance: 0,001

Absolute Tolerance: 1E-7

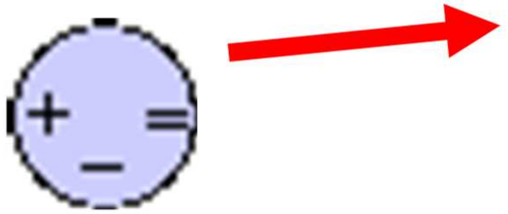
Discrete Time

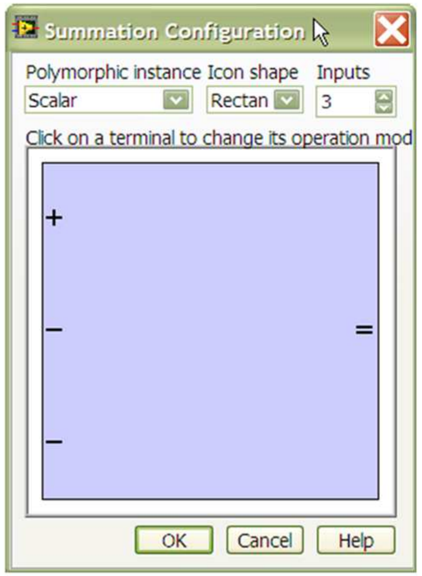
Discrete Time Step: 0,1

Auto Discrete Time: ✓

Simulation Loop – exercise 1

*Use **Summation** function from **Signal Arithmetic** palette to perform arithmetic operations for created model.*
*This function can be configured in configuration window. **Double click** on the icon to open window and reconfigure them as on the figure.*
You can select the number of inputs via the Inputs list, and select Add or Subtract or Disable for each input (by clicking on the symbols inside the area). You can also select among Rectangle and Circle Icon shape, and define the input as a scalar (single signal) or a vector (multiple signals).



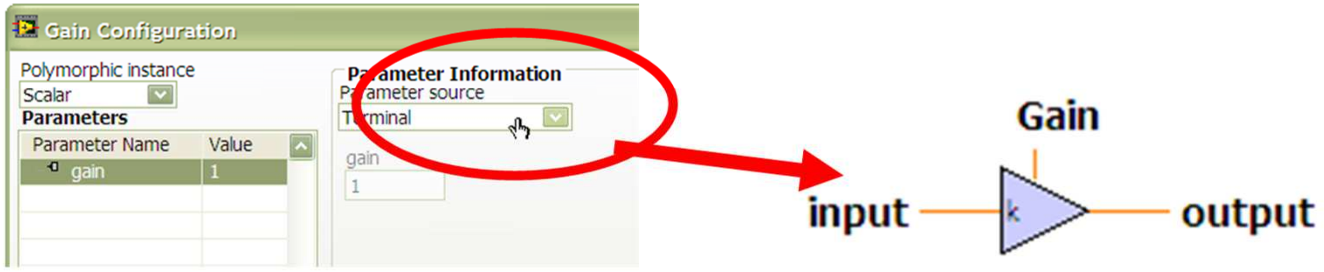

Summation Configuration

Polymorphic instance: Scalar
Icon shape: Rectan
Inputs: 3

Click on a terminal to change its operation mod

Simulation Loop – exercise 1

*To define gains in the model use **Gain** function* ▷ *from **Signal Arithmetic** palette. Gain can be set in the configuration window (double click on the icon) or can be defined programmatically outside the window. It is comfortable in this case. Select option **Terminal** in **Parameter Source** list and press OK button. The Gain icon should be changed.*


Gain Configuration
Polymorphic instance: Scalar
Parameters

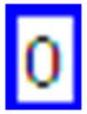
Parameter Name	Value
gain	1

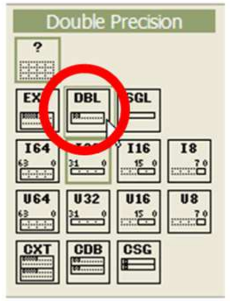
Parameter Information
Parameter source: Terminal
gain: 1

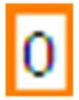
*It is necessary to create input data for **1/m** gain definition.*
*Select **Functions/Programming/Numeric** palette and place **Numeric Constant** on the Block Diagram inside the loop.*

Simulation Loop – exercise 1

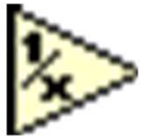
You can change the numeric representation of this element. ***Right click*** *on it, select* ***Representation*** *option in context menu and choose* ***Double precision (DBL)*** *format.*

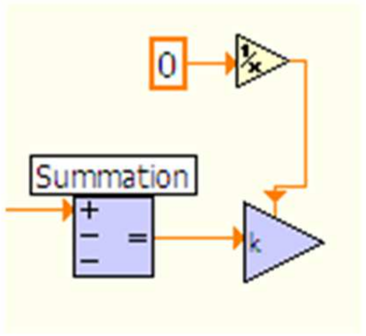






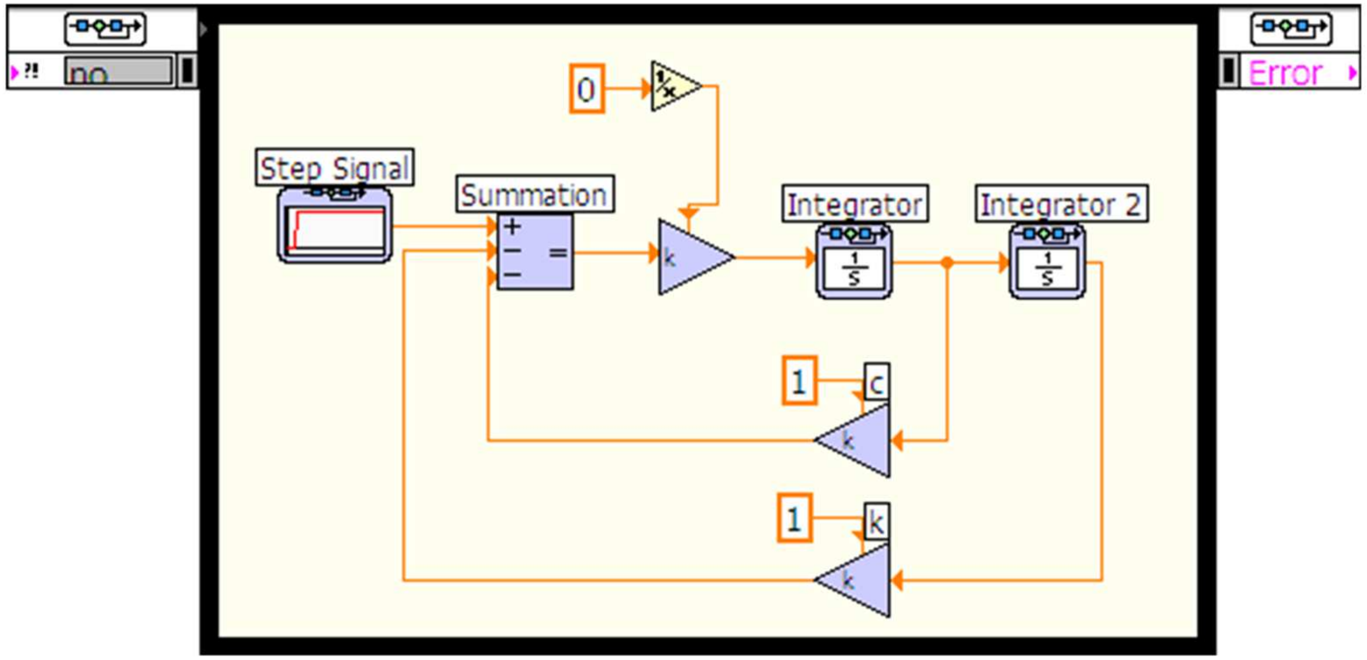
Select ***Functions/Programming/Numeric*** *palette and place* ***Reciprocal Function*** *on the Block Diagram near the constant previously created. Create a wires for* ***1/m*** *gain.*





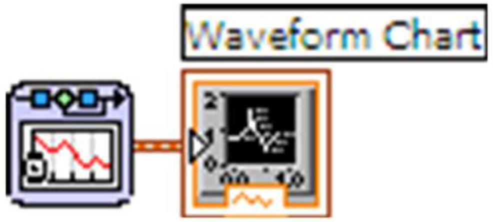
Simulation Loop – exercise 1

Complete the diagram inside the loop as presented below.

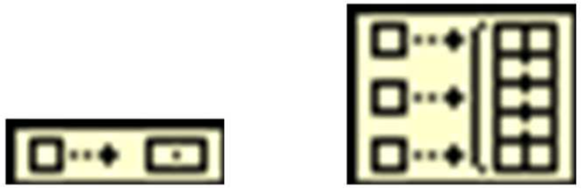


Simulation Loop – exercise 1

*To display the results of the simulation select **Sim Time Waveform** function from **Graph Utilities**.*

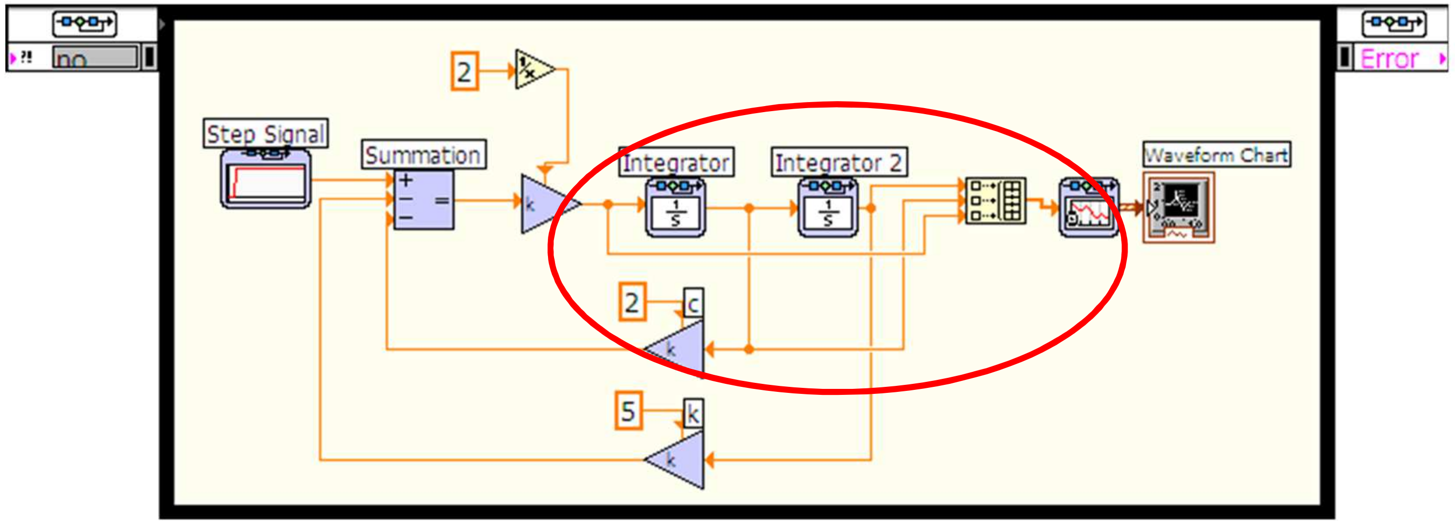


It is nice to see more signals on the output chart. It is possible to connect a few signals into one data stream.
*Select **Functions/Programming/Array** functions palette and place **Build Array** function on diagram. You can unfold this symbol to the specified number of data inputs.*



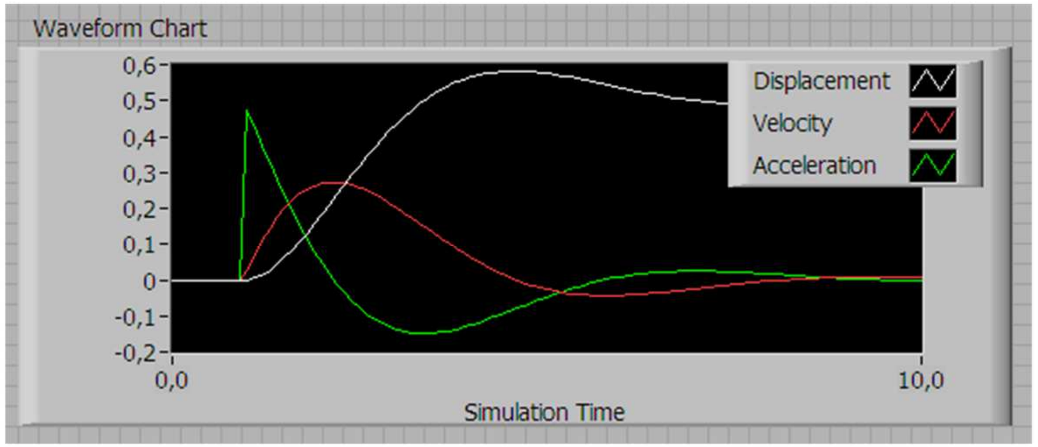
Simulation Loop – exercise 1

Create the wires as on the figure below.
*Set the values **m, c** and **k**.*



Simulation Loop – exercise 1

*Open **Front Panel**. You can see the chart. Save the program.*
Start simulation.



*You can change the parameters in the **Simulation Loop** configuration window and observe their influence on the result.*