SimulationX
Powered by ITI

Tutorial 19

Export SimulationX Models to NI VeriStand

Objective

VeriStand from National Instruments is a software environment for generating real-time testing and Hardware-in-the-loop (HIL) applications. It can import control algorithms, simulation models and other task from NI LabVIEW software or third-party environments like SimulationX.

This tutorial demonstrates how SimulationX models can be exported to VeriStand. We show step by step the creation of the SimulationX model, the export using the SimulationX Code Export Wizard, and explain how to use such a model within VeriStand.

It is assumed that the reader is familiar with the basic functionality both, of SimulationX and VeriStand. For an introduction to SimulationX, please refer to **"Tutorial 1: Getting Started"** and **"Tutorial 2: Hydraulic Cylinder Drive"**.

The following software modules are needed to execute this tutorial:

- SimulationX incl. a valid license for the Code Export Target "NI VeriStand / LabVIEW Simulation Interface"
- NI VeriStand 2010
- Microsoft Compiler (e.g. the free Microsoft Developer Studio Express Edition)

Model Export to NI VeriStand - What Can It Be Used For?

The NI VeriStand Interface of SimulationX can be used to implement complex dynamic simulation models, created in SimulationX, into the VeriStand environment for further system analysis on real-time targets and Hardware-in-the-Loop (HIL) simulations.

SimulationX provides a huge number of predefined elements in different physical domains to create detailed physical based simulation models. Using such models inside VeriStand allows:

- Real-time Test of real hardware elements (e.g. controllers) using SimulationX plant models (Hardware-in-the-Loop - HiL)
- Real-time Test of new controller code using SimulationX plant models (Software-in-the-Loop - SiL)
- Creation of graphical user interfaces around SimulationX models with VeriStand Workspaces

- *Detailed physical models for NI VeriStand*
- *Real-Time deployment for HIL-solutions and RCP*
- *Easy graphical model generation and user-friendly Code-Export Wizard*
- *End user applications using SimulationX models without programming*

Example

On the basis of the SimulationX sample model CylinderDrive.ism the export process will be demonstrated. A physical based model of a hydraulic cylinder drive will be exported to VeriStand. The model can be found in SimulationX via menu **"Help – Sample Browser …"**. Use the Sample Browser to select **Hydraulics** and open the **CylinderDrive.ism** sample. After opening the model in SimulationX the Sample Browser can be closed.

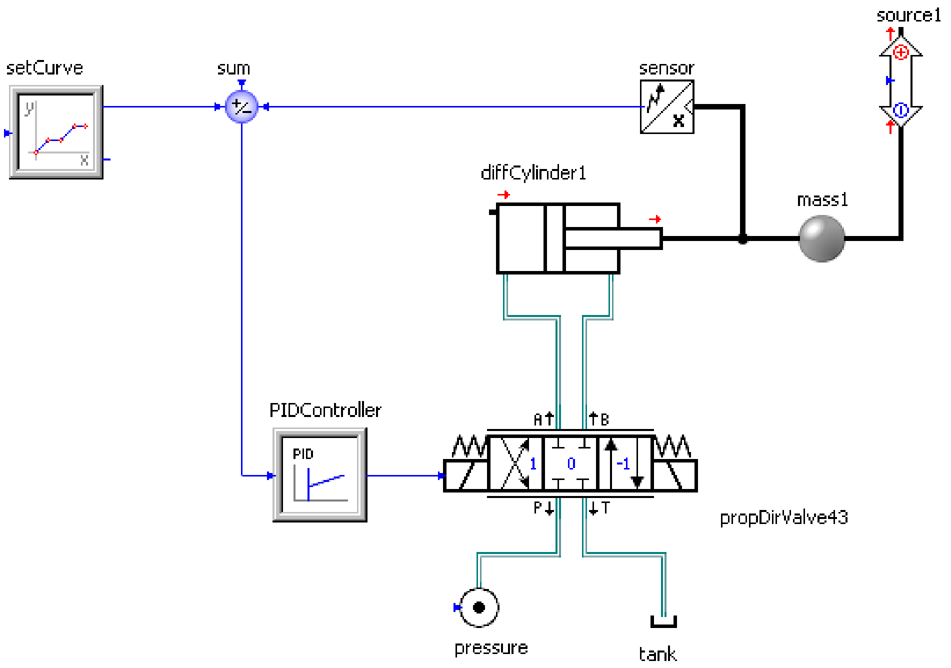

Figure 1: SimulationX sample model CylinderDrive.ism

The model consists of the plant model of the hydraulic cylinder drive and a PID-Controller. Typically in HIL applications the electrical inputs and outputs of a real existing controller are connected to the hardware I/O of the real time target. In this tutorial example we do not want to use any hardware, so we split the model into two parts and therefore create two models, which will be exported to VeriStand separately:

- the controller model (signal part of the sample model)
- plant model (hydraulic and mechanical part of the sample model)

After exporting them to VeriStand we will connect both models there via signal mapping in the same way a plant model would be connected to hardware I/O signals.

So first do the following steps to create the controller model:

- Delete all elements in the sample model except of ***setCurve, sum*** and ***PIDController*** (see Figure 5)
- Replace the element ***PIDController*** by a PIDT1 controller from SignalBlocks.Linear.PIDT1 and name it **Controller**
- Change the parameters of **Controller** as shown in Figure 2

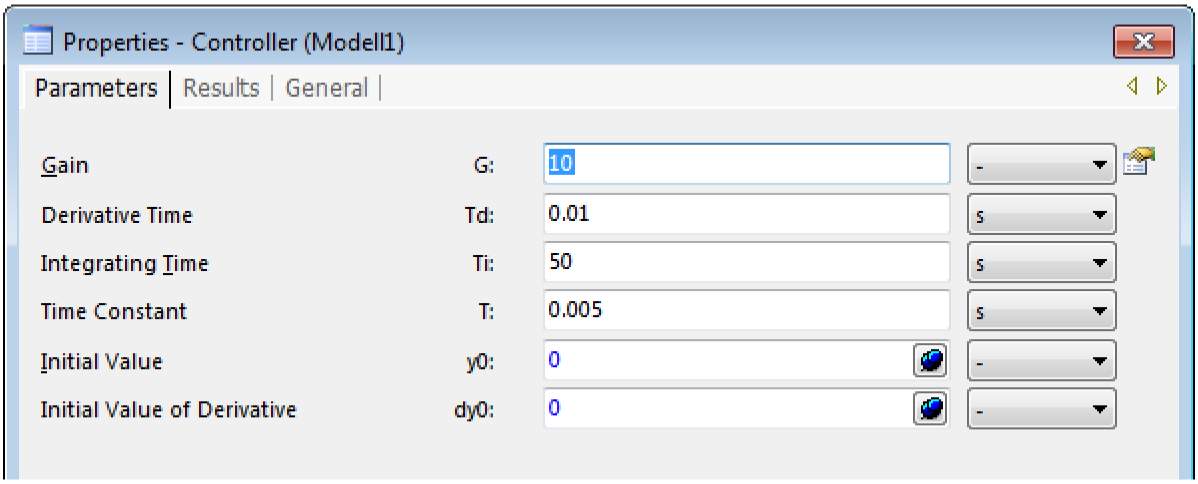

Figure 2: Property Dialog PIDT1-element **controller**

- Replace the curve element **setCurve** by a function block and name it **setValue.**
- Change the value F: of **setValue** to **self.x** in Figure 2

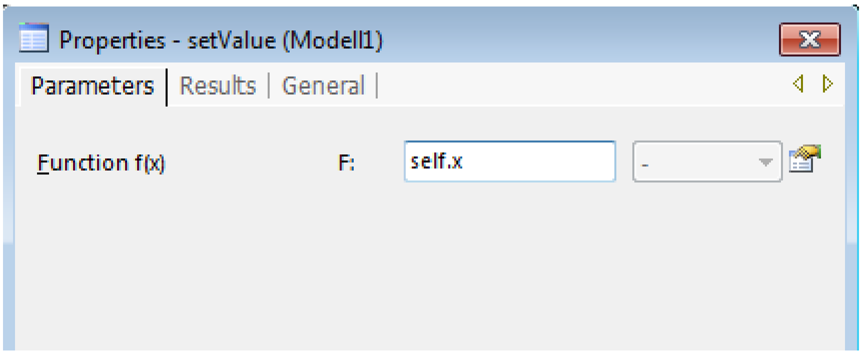

Figure 3: Property Dialog function-element setValue

- Change the minimum step size settings (**dtMin=5e-4 s**) and the stop time (**tStop=1 s**) according to the following figure in the simulation control **"Simulation - Transient Settings…"** on the tab **General**. The minimum step size **dtMin=5e-4 s** is the internal integration step size used by the exported fixed step solver during computation in VeriStand. The minimum output step size has to be equivalent to the Target Rate in VeriStand, so we will change that parameter later in the VeriStand project.

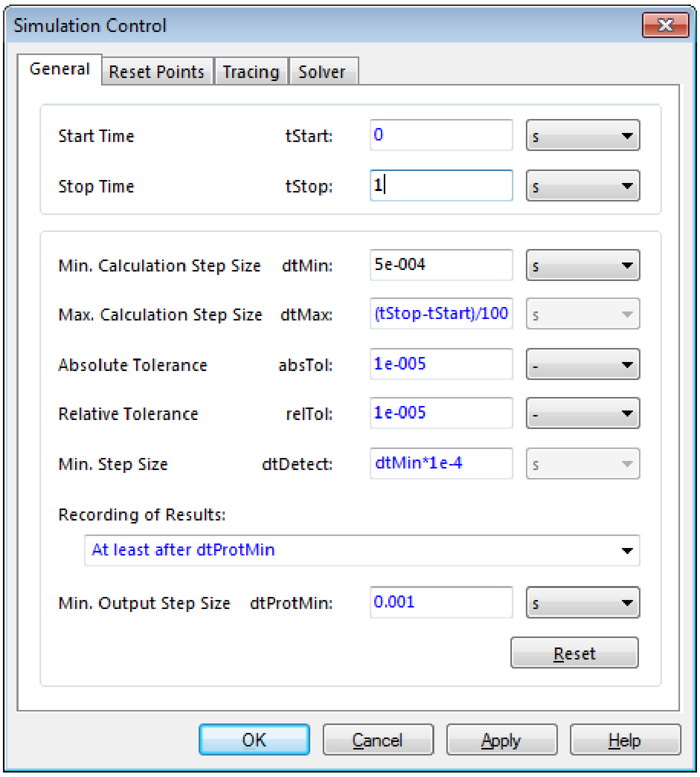

Figure 4: Simulation Control - General

- Save the model into your working directory using a different name e.g. CylinderDrive_Controller.ism via the menu **"File – Save as …"**.

Figure 5 shows the model structure of the controller model after these previous modifications.

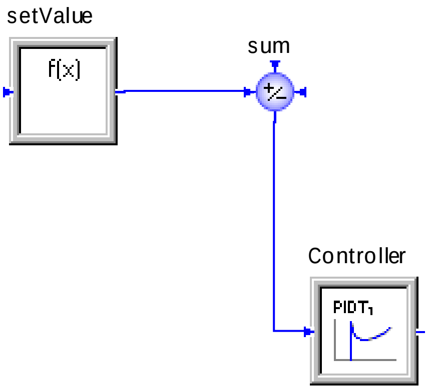

Figure 5: model of the controller

Open a second sample model of the cylinder drive (**"Help – Sample Browser …"**) and do the following steps to create the plant model of the cylinder drive:

- Delete the elements ***setCurve***, ***sum*** and ***PIDController.***
- The displacement sensor has to be replaced by a sensor element from Mechanics.Linear Mechanics.Sensor.

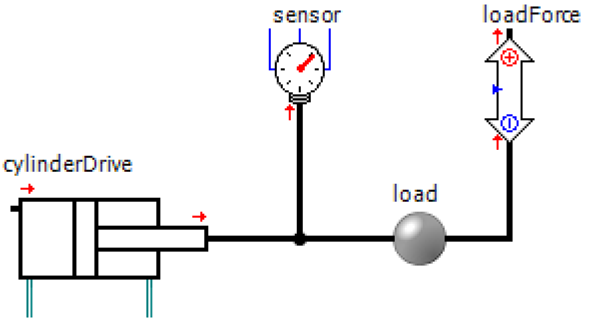

Figure 6: Replaced sensor at the piston rod of the cylinder drive

- The model at this stage does not include the pipe or hose line connection between control valve and cylinder. To consider at least the enclosed fluid volume and the wall elasticity of the pipes, add two volume elements (Library: Hydraulics.Basic Elements.Volume) between cylinder ports and valve ports. The volume can be calculated within the parameter **V** of the volume Element. A pipe length of **0.5 m**, an inner pipe diameter of **12 mm**, and an estimated wall capacity of **50 mm³/bar** are assumed. The parameter dialog of the volume **vA** is shown in Figure 7.

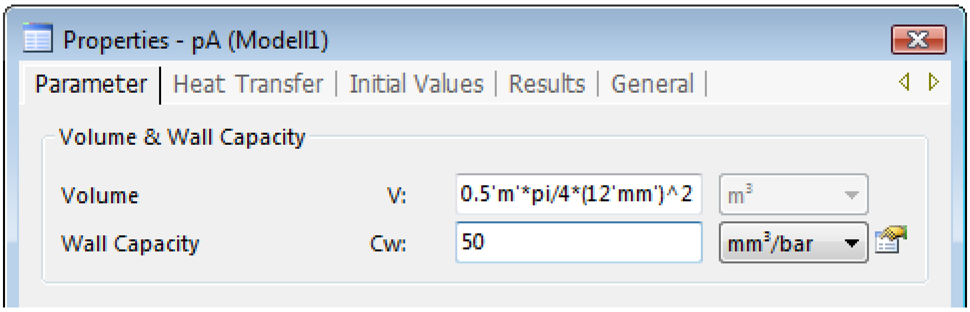

Figure 7: Property Dialog volume vA

- Reset the initial position **x0** of **mass1** to the default value **0**.
- Select also for that model a minimum step size **dtMin** of 5e-4 and a stop time **tStop** of one 1 second (see Figure 4)
- Save the model into your working directory using a different name e.g. CylinderDrive_PlantModel.ism via the menu **"File – Save as …"**.

Figure 8 shows the model structure of the plant model after these previous modifications.

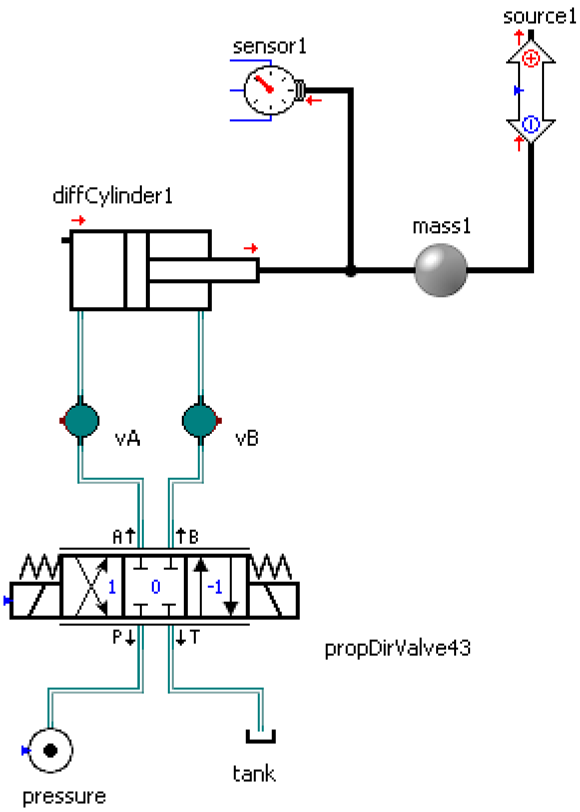

Figure 8: Hydraulic Cylinder Drive – Plant model

The SimulationX Code Export wizard generates a component for VeriStand which contains the model functionality and a solver. In order to support real time behaviour in VeriStand a fixed step solver is used here. So the model should be tested in SimulationX with a fixed step solver too. A critical point in that sense is the case where the piston goes into the cylinder end stops. This is due to the fact that the model becomes stiff in that point of operation. The following steps are to be done to carry out this test:

- At first select add a function block (Library: Signal Blocks.Function) and name it **ValveInput**.

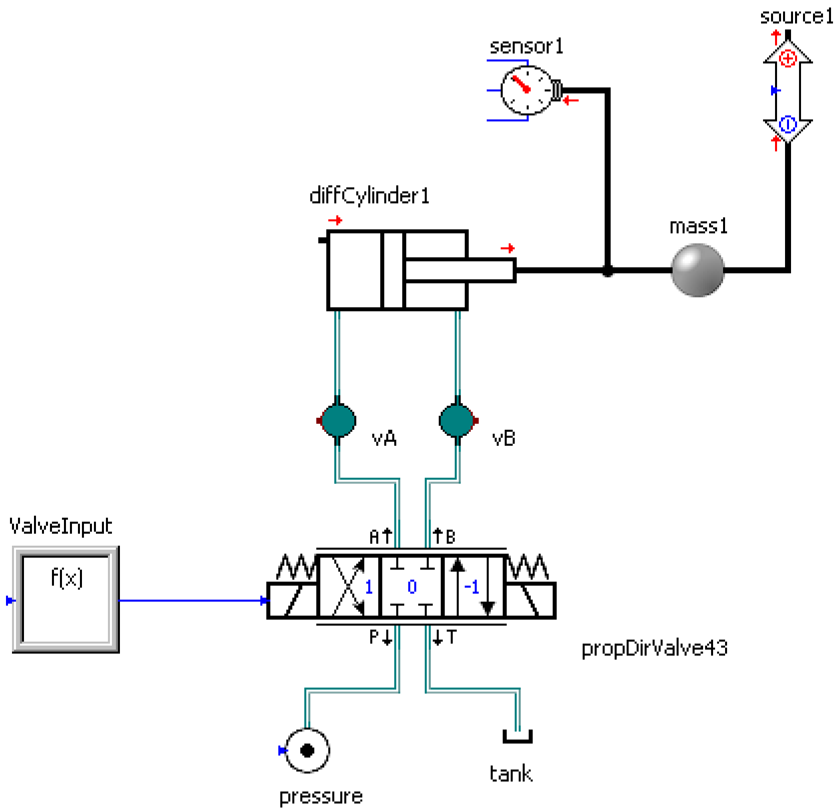

Figure 9: Plant model – Fixed Step Solver Test

- To run the piston into both end stops, firstly charge the left port of the cylinder with pressure, followed by the other port later. Therefore, the proportional valve has to be driven first with a negative input signal and later with a positive one. Enter the statement "**if time>0.5 then 1 else -1**" for the parameter **F** of the function block **ValveInput**.
- Run the simulation.
- The simulation results for the piston stroke and the cylinder chamber pressures are shown in Figure 10 and Figure 11.

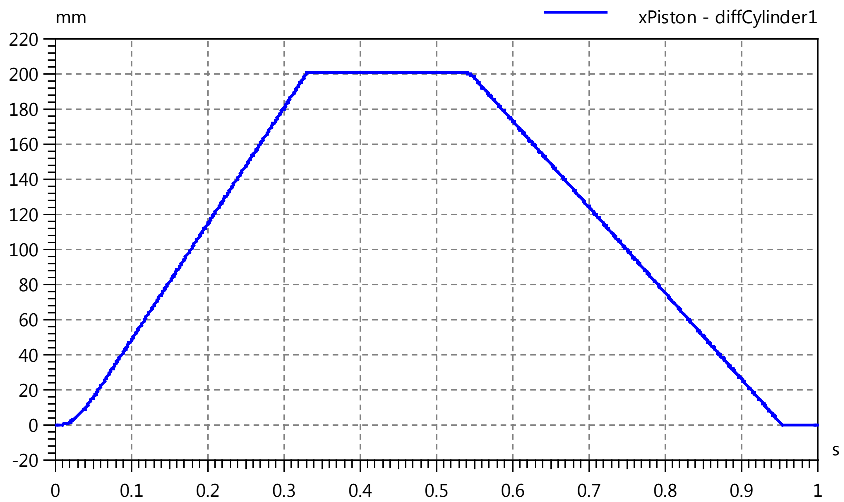

Figure 10: Simulation Results Piston Stroke

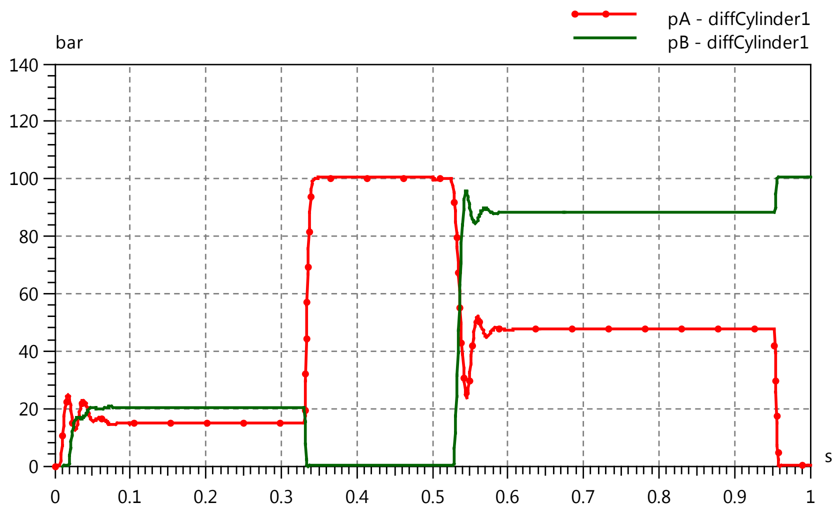

Figure 11: Simulation results pressures chamber A and chamber B

In order to check the results for correctness the model should be simulated using smaller step sizes or one of the SimulationX variable step size solvers.

After this test you should delete the function block **ValveInput** and save the plant model.

- **C-Code Export**

First we want to export the controller model (Figure 5), so change to that model.

To export the model, open the Code Export Wizard via the menu **"Export – C-Code"**. On the first page, select **NI VeriStand / LabVIEW Simulation Interface** as project type, enter a project name and select a target directory:

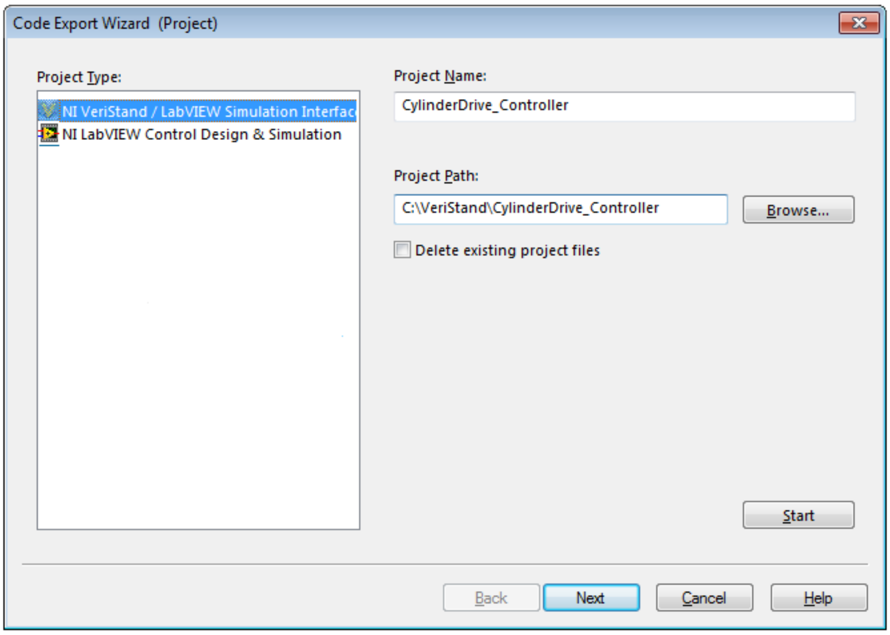

Figure 12: Code Export Wizard Page 1 (Project) – Controller model

On the following page, select the inputs which will be available in VeriStand (**sum.x2, setValue.x**) by a double click on the desired inputs. The selected inputs are displayed in the list **Selection**:

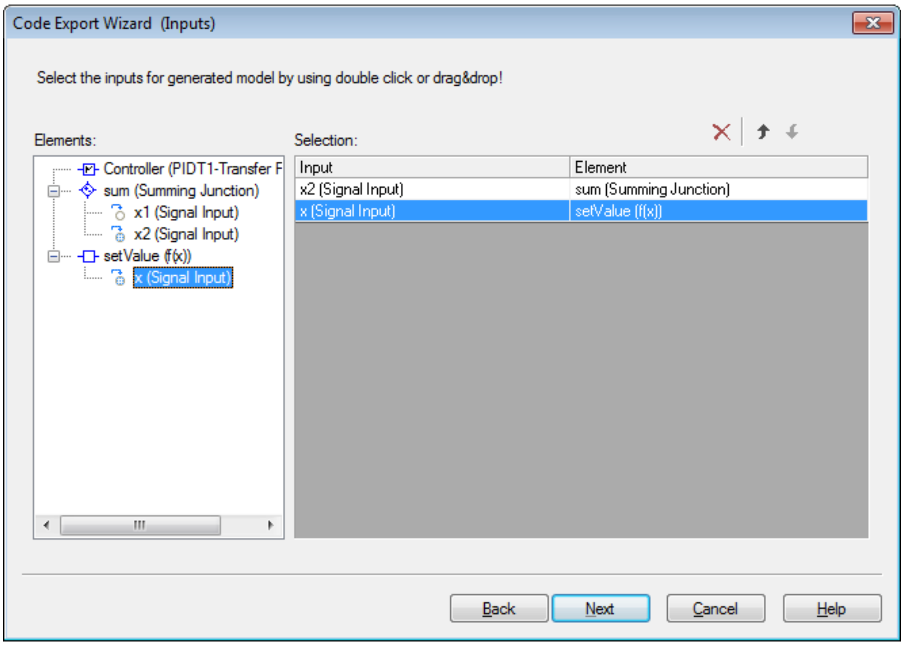

Figure 13: Code Export Wizard Page 2 (Inputs) – Controller model

On the next page, select the outputs (**Controller.y**):

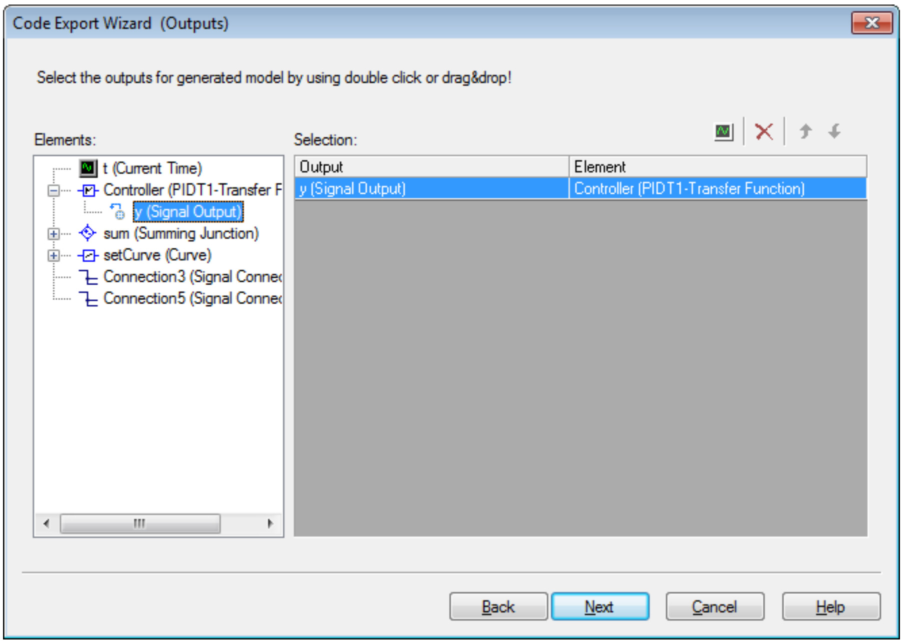

Figure 14: Code Export Wizard Page 3 (Outputs) – Controller model

To access model parameters of the PID Controller after the export, select **Controller.G** (gain of the PIDT1-controller) as parameters on the next page of the wizard:

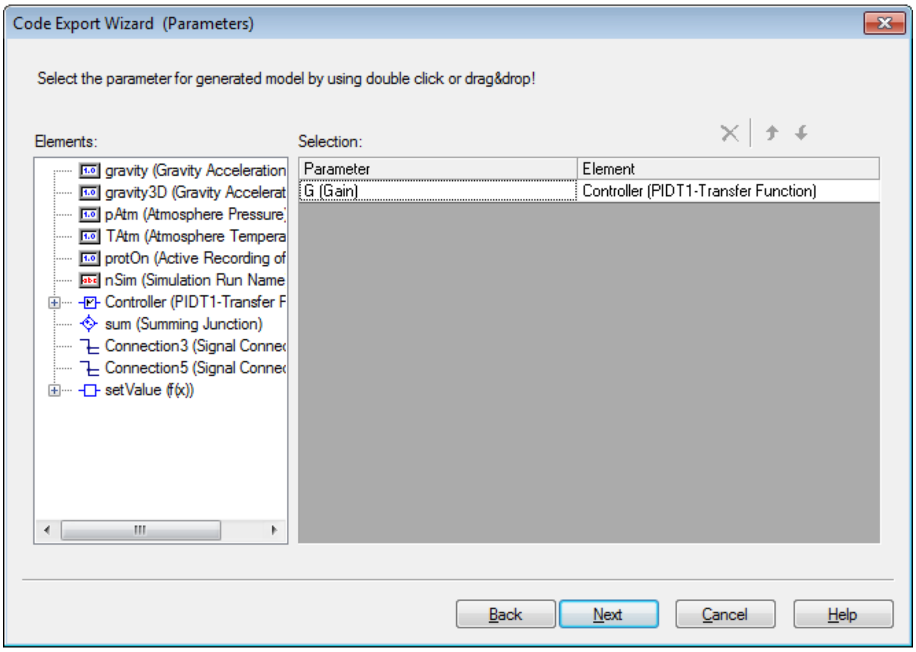

Figure 15: Code Export Wizard Page 4 (Parameters) – Controller model

By stepping to the next wizard page, the SimulationX Modelica compiler simplifies the model and generates the C-code of the model and the fixed step solver in the target directory.

On the Post-Processing page the DLL which we will import to VeriStand is to be compiled by pressing the "**Build**" button.

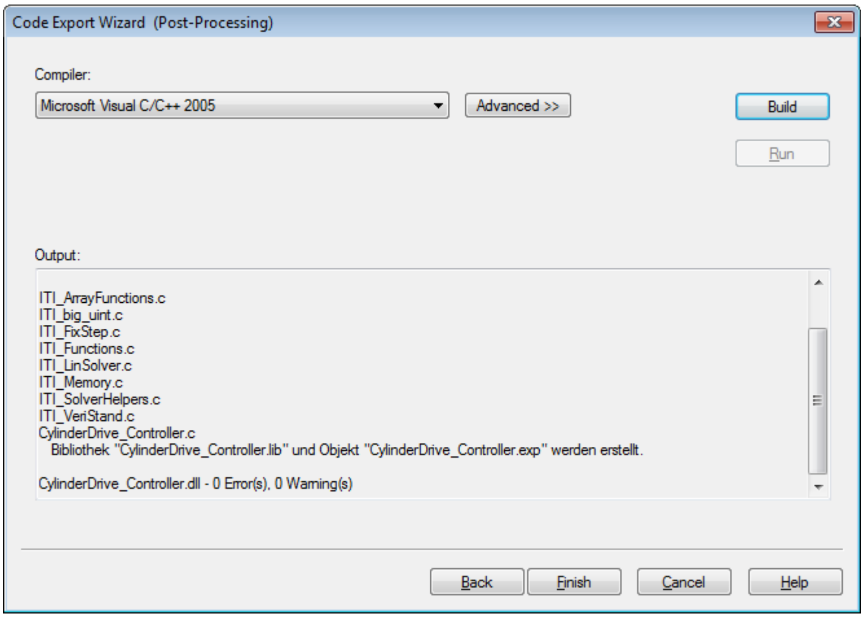

Figure 16: Code Export Wizard Page 6 (Postprocessing) – Controller model

If the compiler list box is empty on your computer, please install one of the following Microsoft compilers:

- Microsoft Visual C/C++ 6
- Microsoft Visual C/C++ 2003, 2005, 2008, or 2010
- Microsoft Visual C/C++ Toolkit 7.0 (free)
- Microsoft Visual C/C++ 2008 Express Edition (free)

Alternatively, open the generated project file CylinderDrive_Controller.dsw in the target directory and compile the DLL directly in your Microsoft Developer Studio.

The controller model is now ready to be implemented in VeriStand, but we have to do the same export process for the plant model of the cylinder drive. So change to the plant model (Figure 8) and make sure the function block **ValveInput** that was used during the fixed step solver test (Figure 9) is deleted, before starting the export process.

Start the Code Export Wizard for that model via the menu **"Export – C-Code"**. On the first page, select again **NI VeriStand / LabVIEW Simulation Interface** as project type, enter a project name and select a target directory:

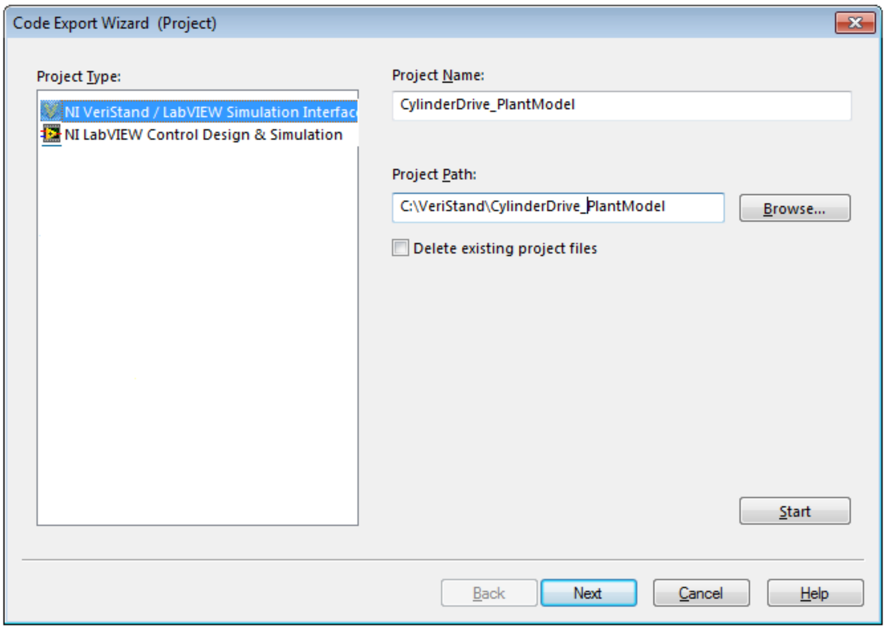

Figure 17: Code Export Wizard Page 1 (Project) – Plant model

On the following page, select the inputs which will be available in VeriStand (**propDirValve43.y**):

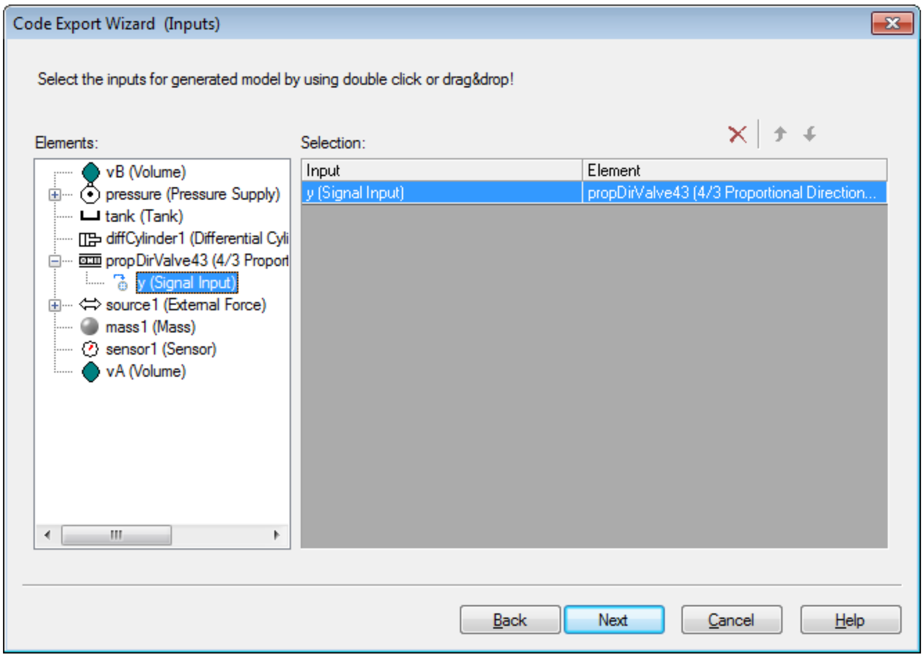

Figure 18: Code Export Wizard Page 2 (Inputs) – Plant model

On the next page, select the outputs (**sensor1.x, vA.p, vB.p**):

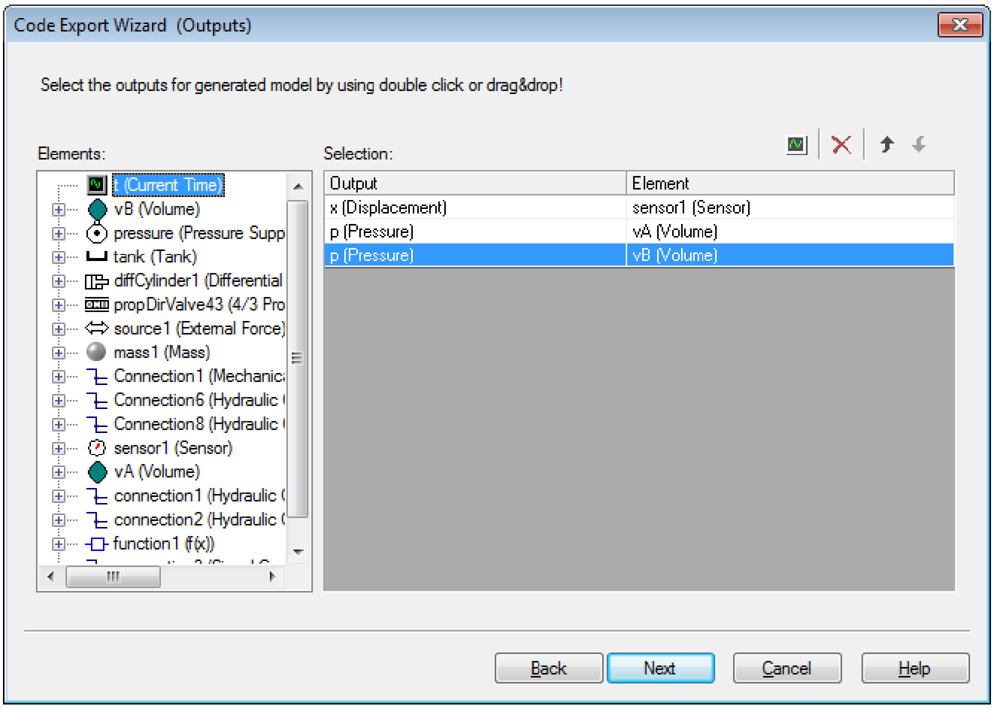

Figure 19: Code Export Wizard Page 3 (Outputs) – Plant model

To access model parameters of the PID Controller after the export, select **pressure.pSrc** (pressure of the pressure supply) as parameters on the next page of the wizard:

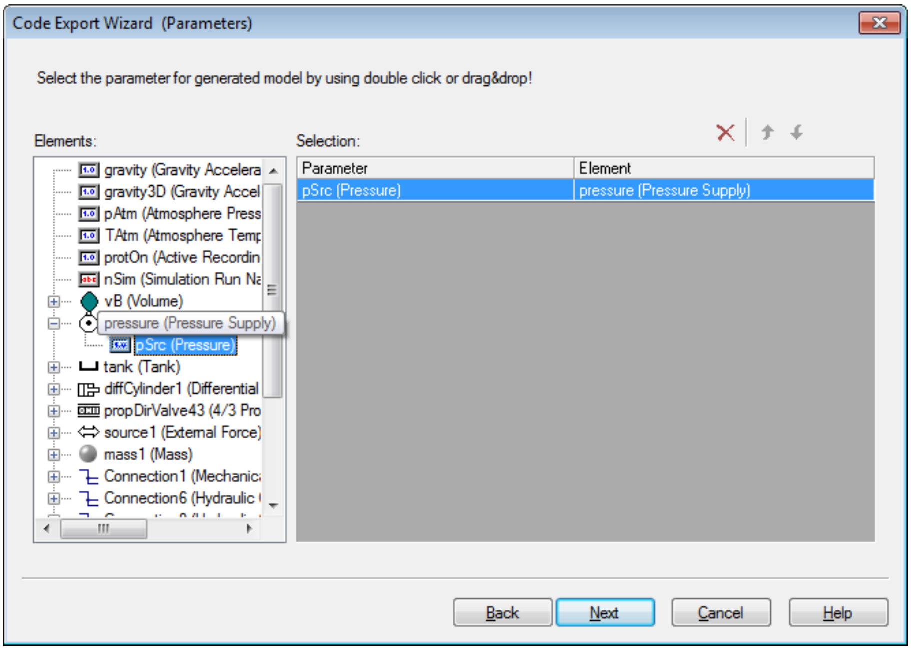

Figure 20: Code Export Wizard Page 4 (Parameters) – Plant model

Press the "**Build**" button on the Post-Processing page to generate the DLL which we will import to VeriStand.

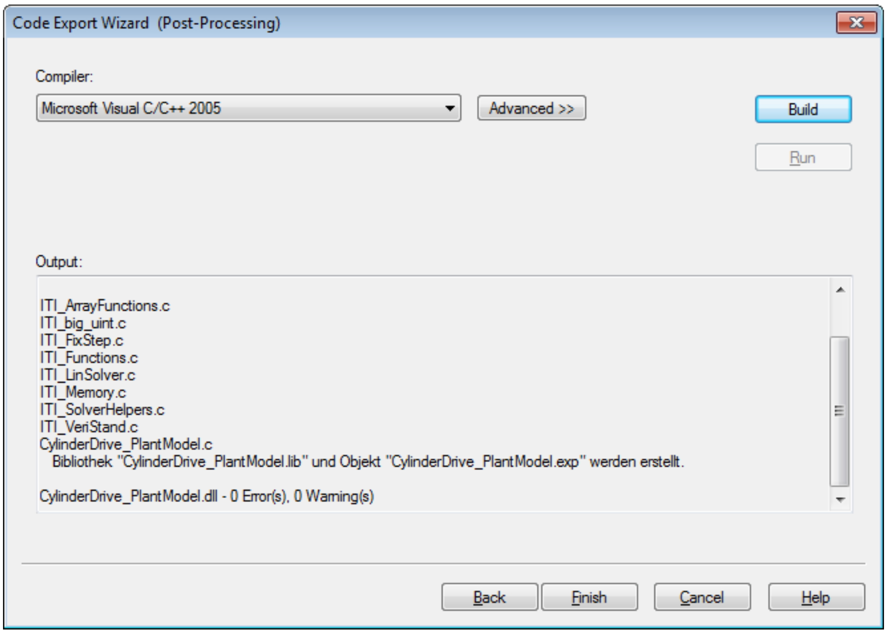

Figure 21: Code Export Wizard Page 6 (Postprocessing) – Plant model

Now we are ready to embed the two exported models into a VeriStand project.

- **Embedding the exported model within VeriStand**

First start VeriStand 2010 and create a new project **CylinderDrive** via the button **New NI VeriStand Project.**

Figure 22: VeriStand main window

After that the project explorer (Figure 24) should open automatically. Please close it first to get back to the main window (Figure 22).

Under Active Project you can see the folder were all the project data (e.g. the system definition file) is stored. Open that folder and copy both generated DLL-files of the controller and the plant model to this location.

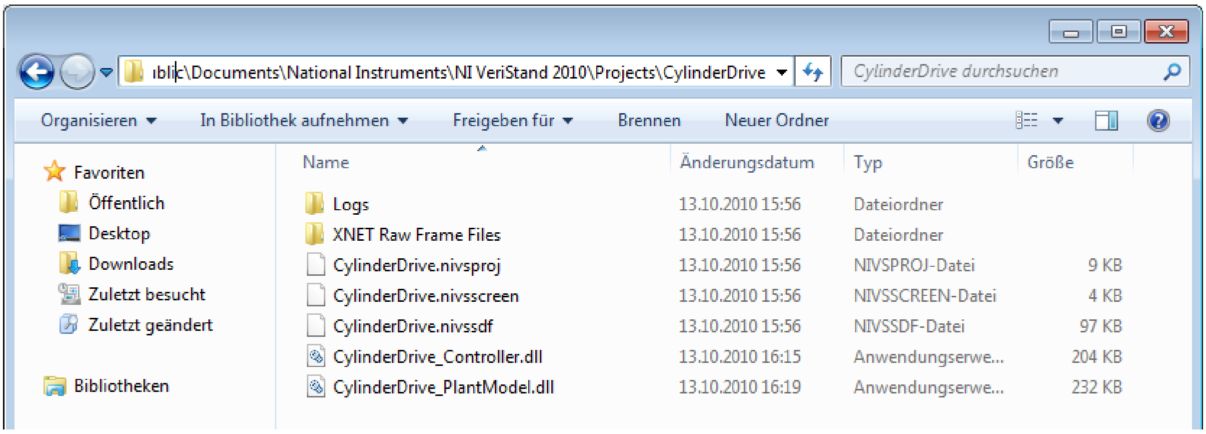

Figure 23: Exported models in VeriStand project folder

Now open the project explorer again via the button **Configure Project** on the VeriStand main window (Figure 22).

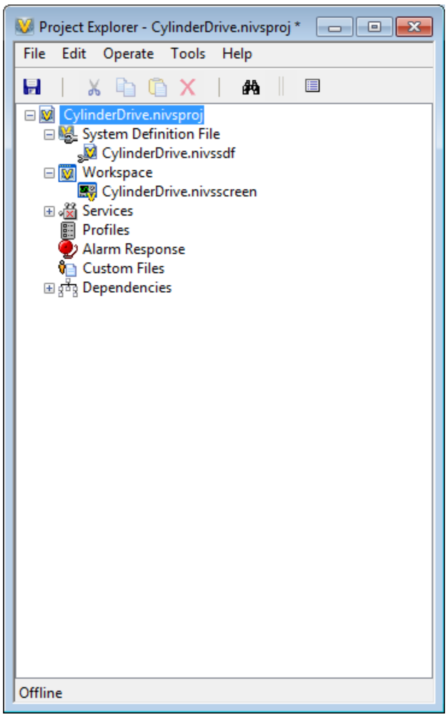

Figure 24: VeriStand Project Explorer

Open the system definition file of project **CylinderDrive** with a double click on **CylinderDrive.nivssdf** and click on **Controller** in the tree structure:

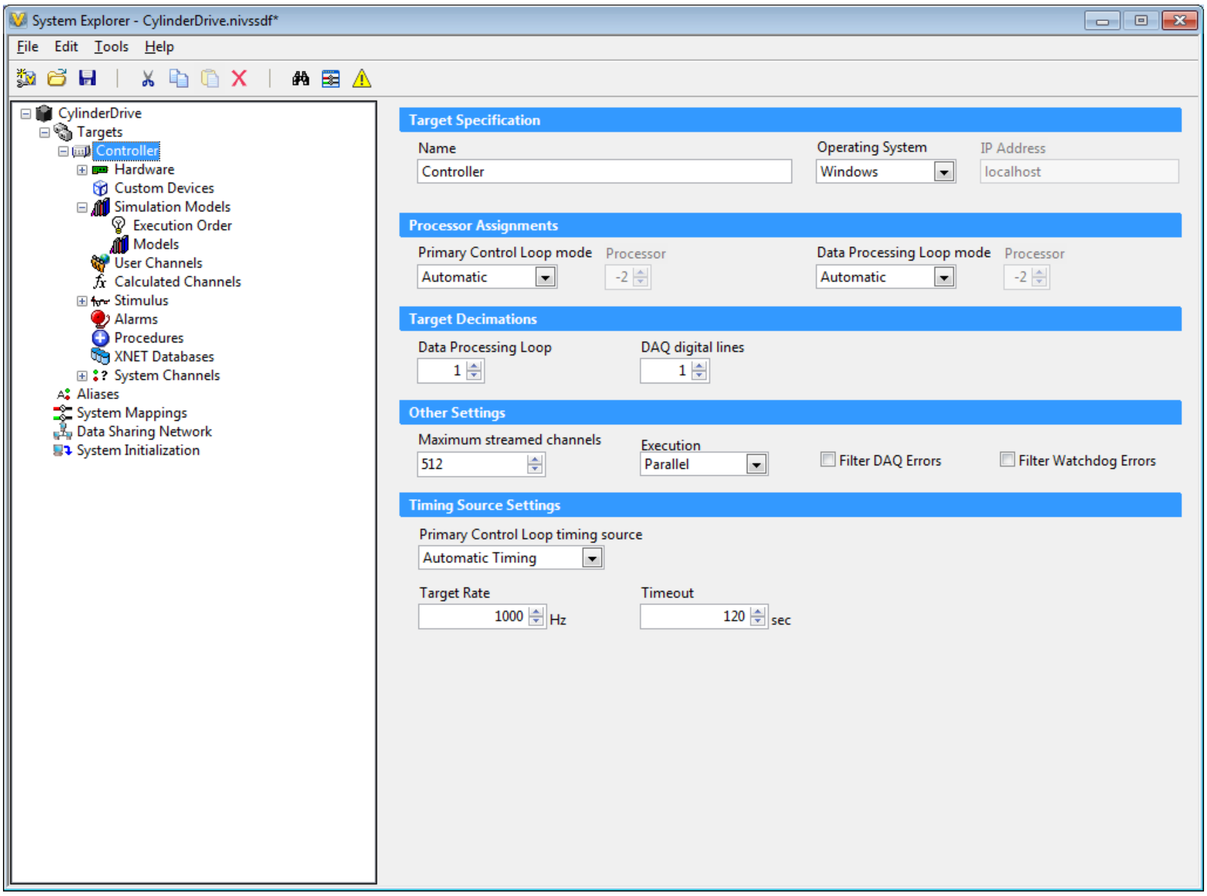

Figure 25: VeriStand System Explorer

Under **Target Specification** you can choose which type of real time target is to be used. Select **Windows** under **Operating System** to use your PC as real time target. If you want to use another real time target (e.g. a PXI controller or a real time PC) that is connected with your HOST-PC via network choose **PharLap** as Operating System and enter the targets **IP-address**.
Make sure that the **Target Rate** (under **Timing Source Settings**) is equivalent to the solver setting parameter **dtProtMin** of the exported SimulationX model. You can find this parameter in SimulationX in the Simulation Control ("**Simulation – Transient Settings**" on the tab **General;** see Figure 4). The minimum output step size dtProtMin is 0.001 s so enter **1000 Hz** for the **Target Rate.**

After that right click on **Simulation Models/Models** in the tree structure and choose **Add Simulation Model**. Click on the data button right next to the box Path and select the model **CylinderDrive.Controller.dll** in the opened window:

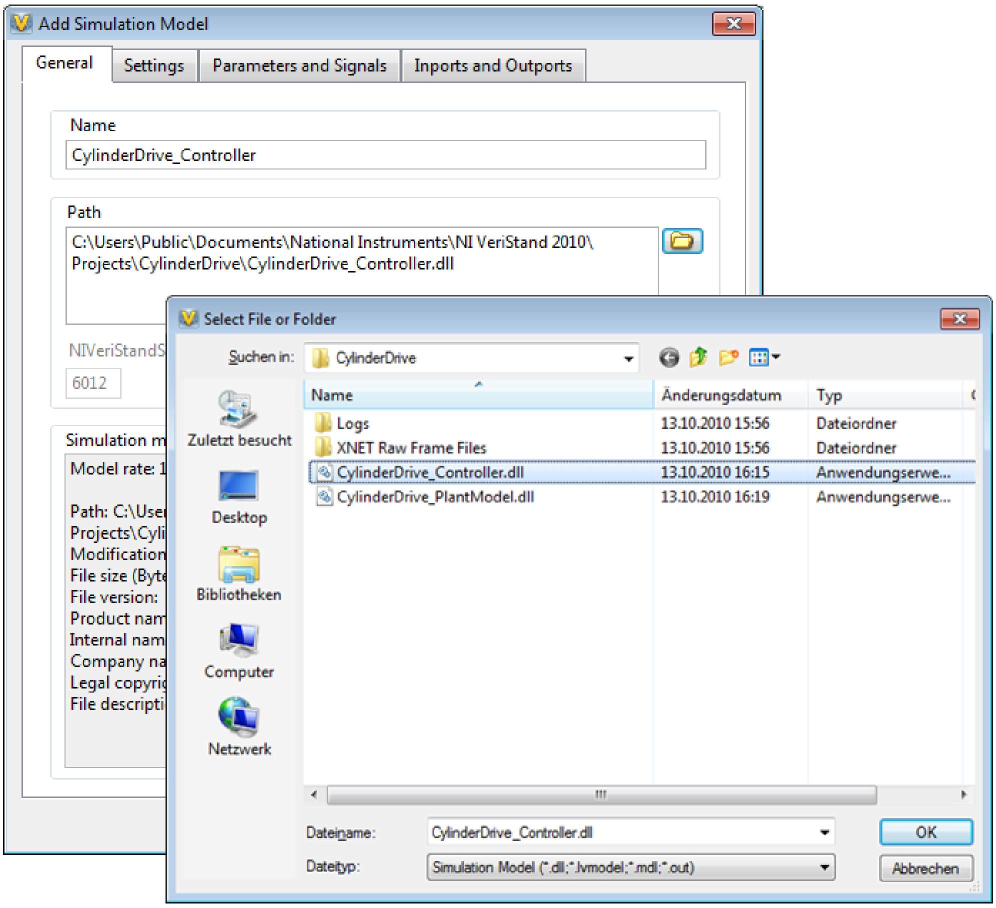

Figure 26: Add Simulation Model

Change to the tab **Parameters and Signals** of the Add Simulation Model window and activate the option **Common Import** and **Import all Parameters** (Figure 27).

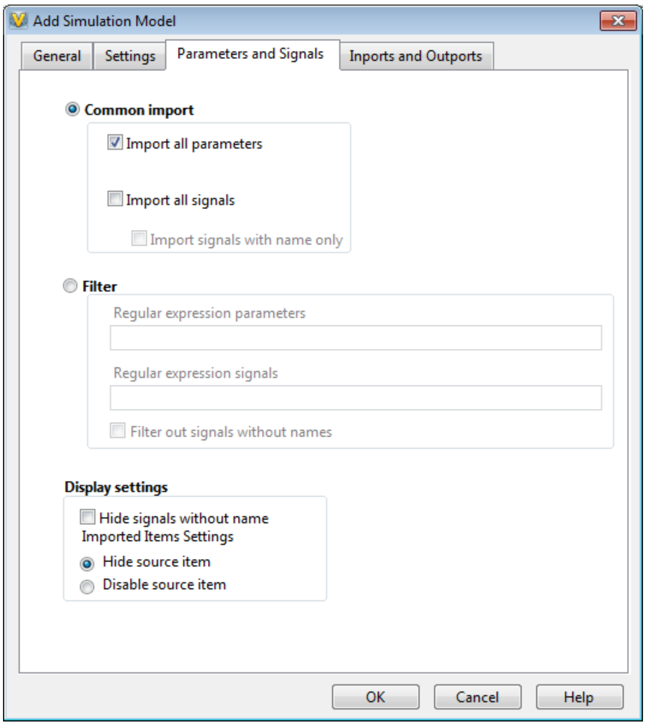

Figure 27: Activate Parameter Import

Close the windows with **OK**.

Do the same adding process for the second model **CylinderDrive_PlantModel.dll**.

Now both models should appear in the tree structure of the System Explorer under **Models:**

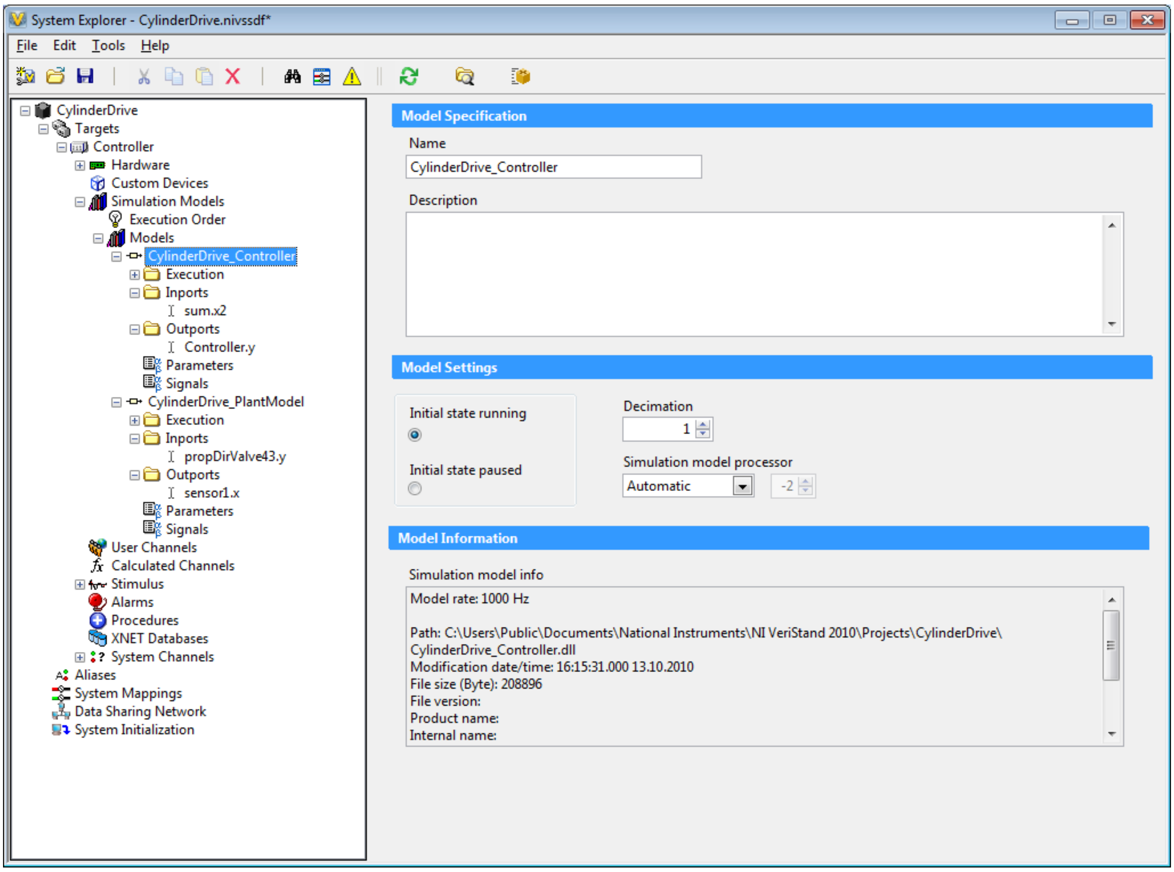

Figure 28: Models in the System Explorer

To establish the signal connection between the models click on **System Mappings** in the tree structure and then click the button **Configure Mappings** **:**

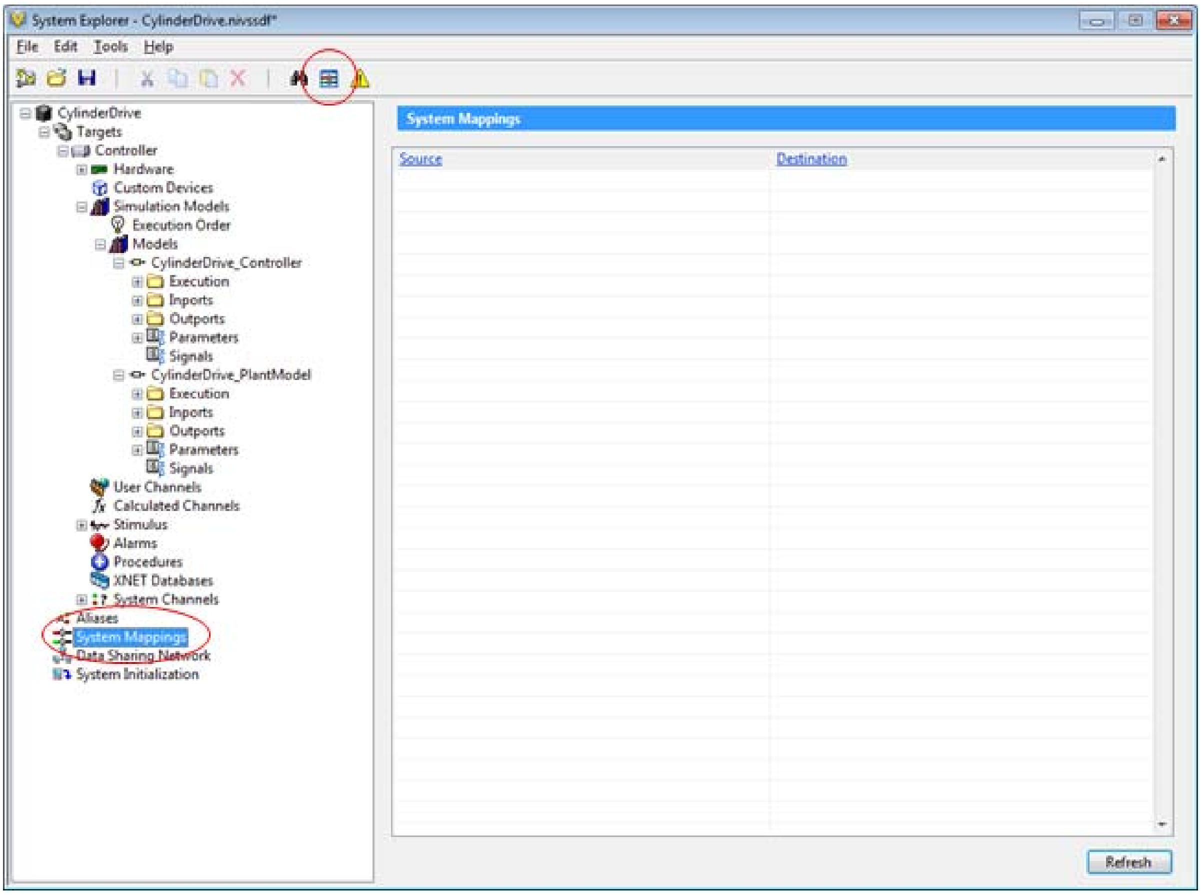

Figure 29: Open System Mappings

A window **System Configuration Mappings** is opened.
For **Source** choose .../Models/CylinderDrive_Controller/Outports/Controller.y, for **Destination** choose ... /Models/CylinderDrive_PlantModel/Inports/propDirValve43.y and press the **Connect** button establish the connection. Do the same for a second connection with a **Source** ... /Models/CylinderDrive_PlantModel/Outports\sensor1.x and a **Destination** Models/ CylinderDrive_Controller/Inports/sum.x2:

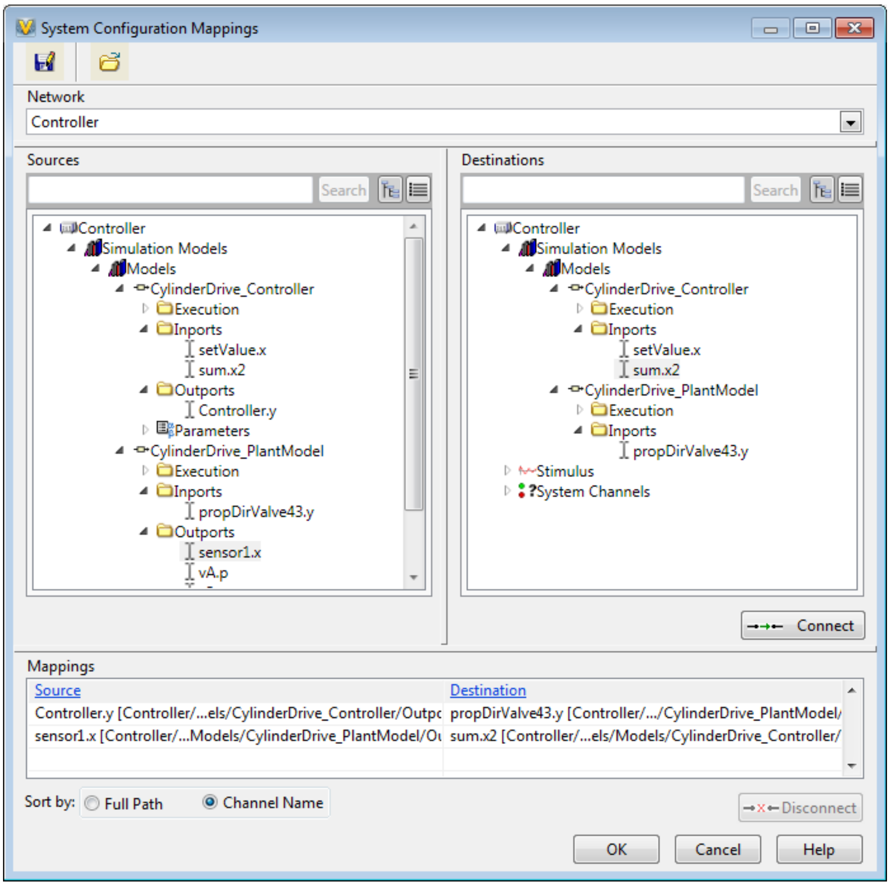

Figure 30: System Configuration Mappings

Close the **System Configuration Mappings** via **OK** and press the button Refresh in the Systems Explorer (Figure 29).
In the list of the System Mappings you can find the two connections now. Make sure that the signal **sensor1.x** is mapped with **sum1.x2** and **Controller.y** is mapped with **propDirValve43.y.**

Now close the system explorer and open the projects Workspace by clicking on **CylinderDrive.nivsscreen** in the project explorer (Figure 24). An empty Workspace is opened.

Change to the **Edit Mode** (Screen - Edit Mode):

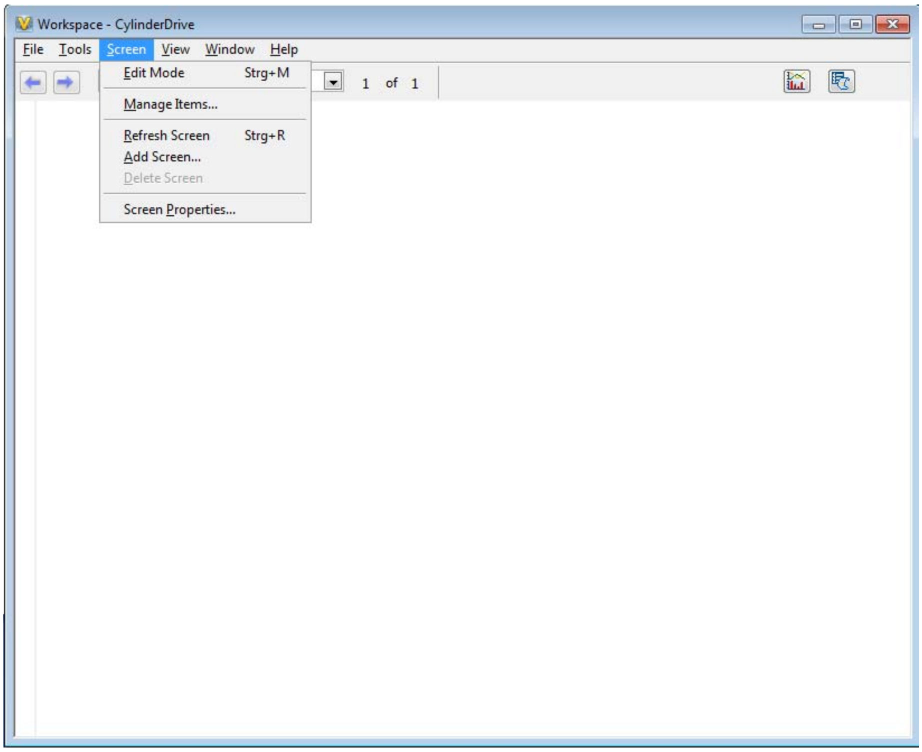

Figure 31: Workspace Edit Mode

First we want to create a dial to control the set value of the controller model.
Click on the tab **Workspace Controls** on the left to open the Workspace library and place a **dial element** (Workspace Controls - Numeric Control – Dial) via drag and drop on the Workspace. The items Property dialog remains open:

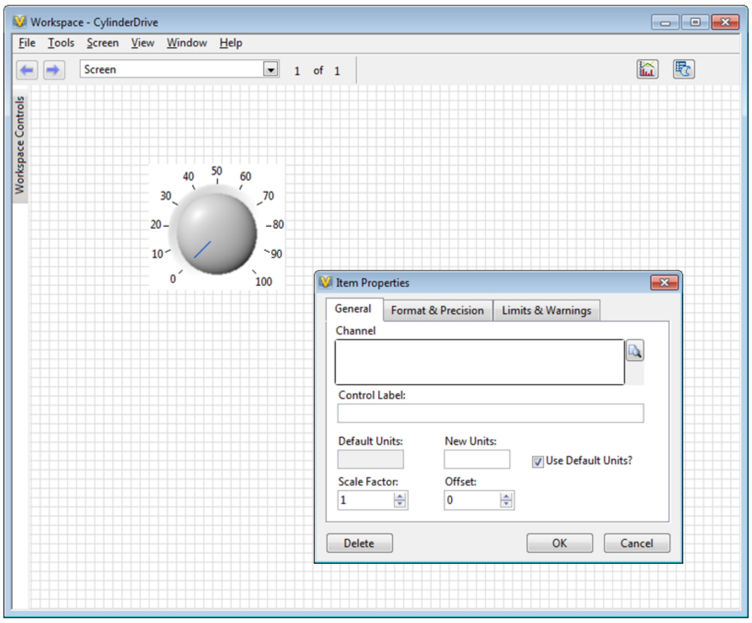

Figure 32: Dial element with Workspace

Now click on the **button**  next to the channel box on the tab **General** and choose setValue.x in the browser window:

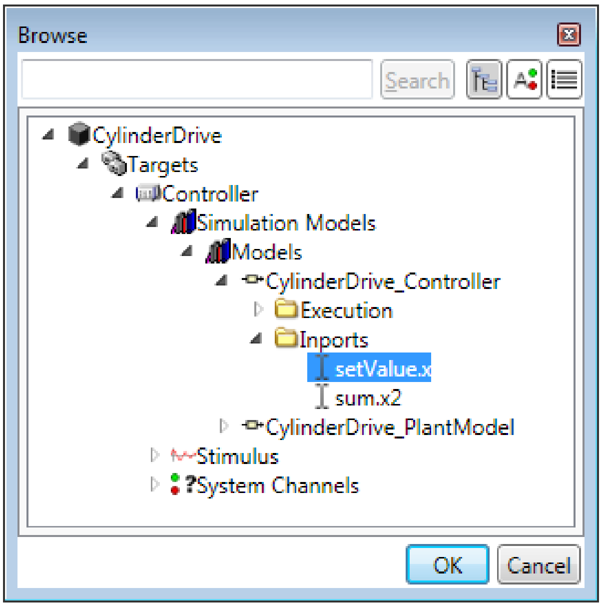

Figure 33: Browser window

Change to the tab **Format & Precision** and enter the following parameters:

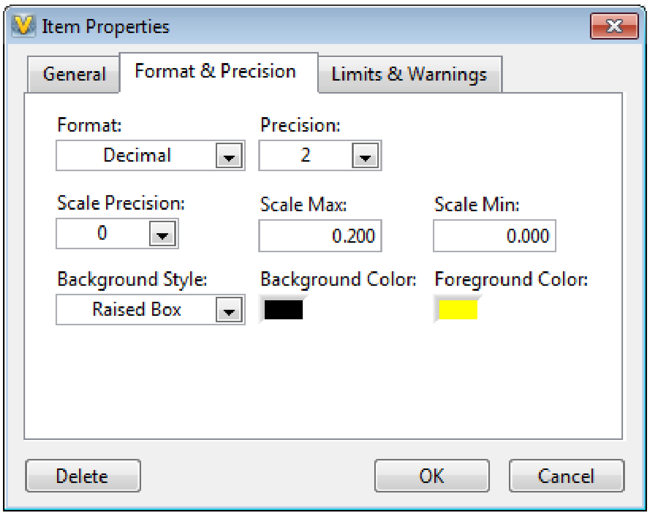

Figure 34: Format & Precision - Dial

Close the dialog with **OK** and move the dial element to the left upper corner of the Workspace.

After that we want to create a time plot of the set value and the current cylinder position. Use the **simple plot** (Workspace Controls - Graph– Simple) and place it on the Workspace. In the window **Graph Channel Selection** choose **setValue.x** and **sensor1.x:**

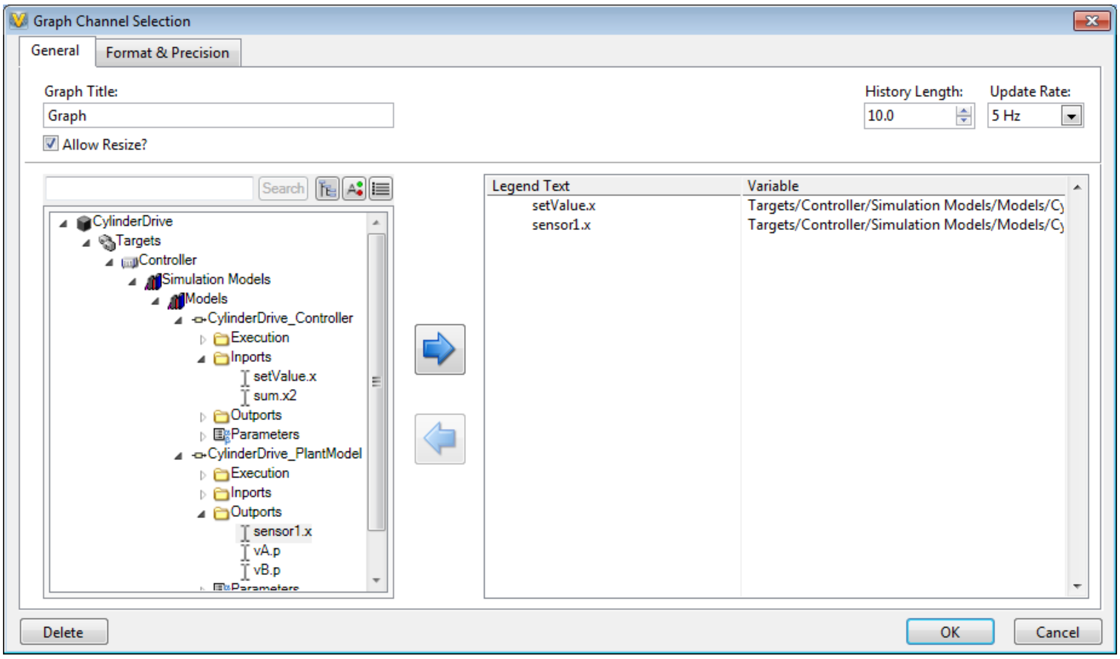

Figure 35: Graph Channel Selection

Do the following changes on the tab **Format & Precision:**

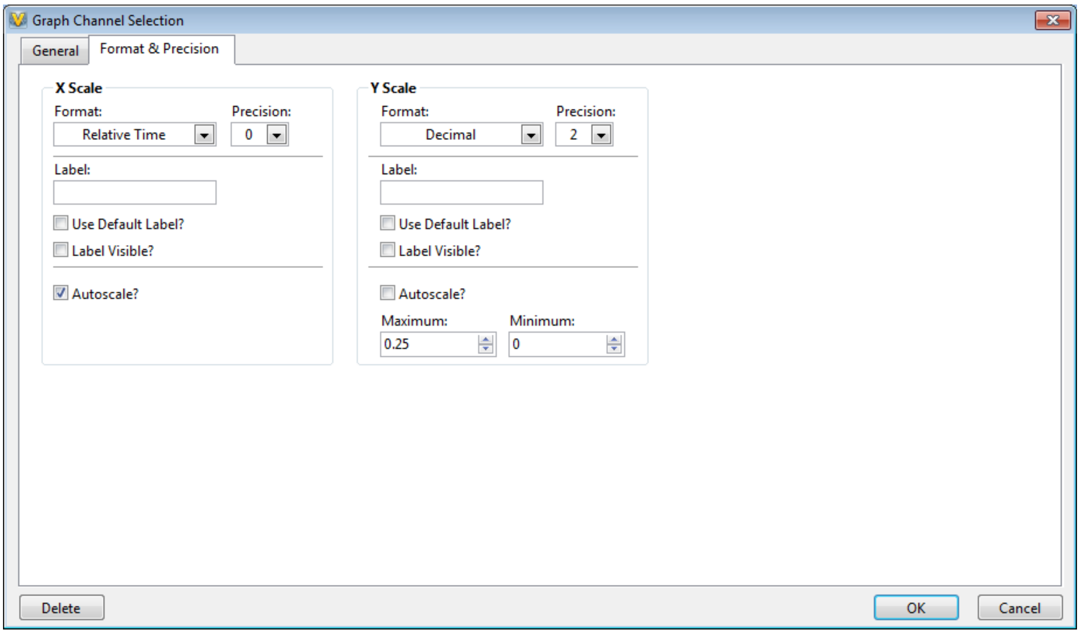

Figure 36: Format &Precision - Plot

Close the dialog with **OK** and place the plot beneath the dial element.

Place two **indicator gauges** (Workspace Controls – Numeric Indicator– Gauge) on the workspace to visualize the actual cylinder pressures. The channel of the first gauge should be **vA.p** and the channel of the second gauge should be **vB.p**:

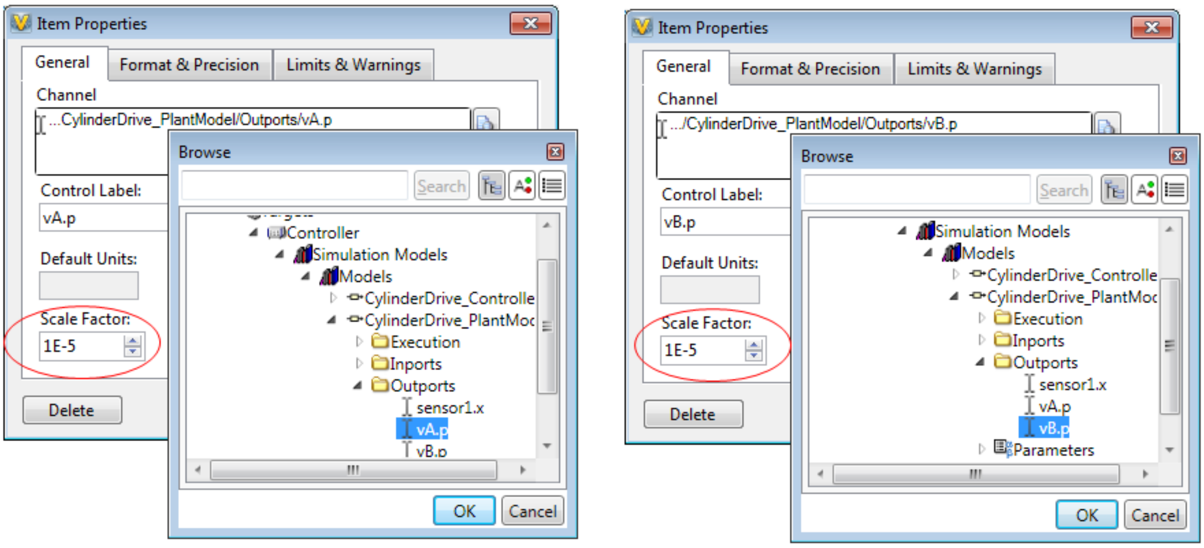

Figure 37: Item Properties - Gauges

Set the **Scale Factor** in both gauges to 1e-5 in order to display the pressures in [bar] and not in the SI-unit Pascal Figure 37. Place the two gauges right next to the dial.

Change to the tab **Format & Precision** and enter the following parameters for both gauges:

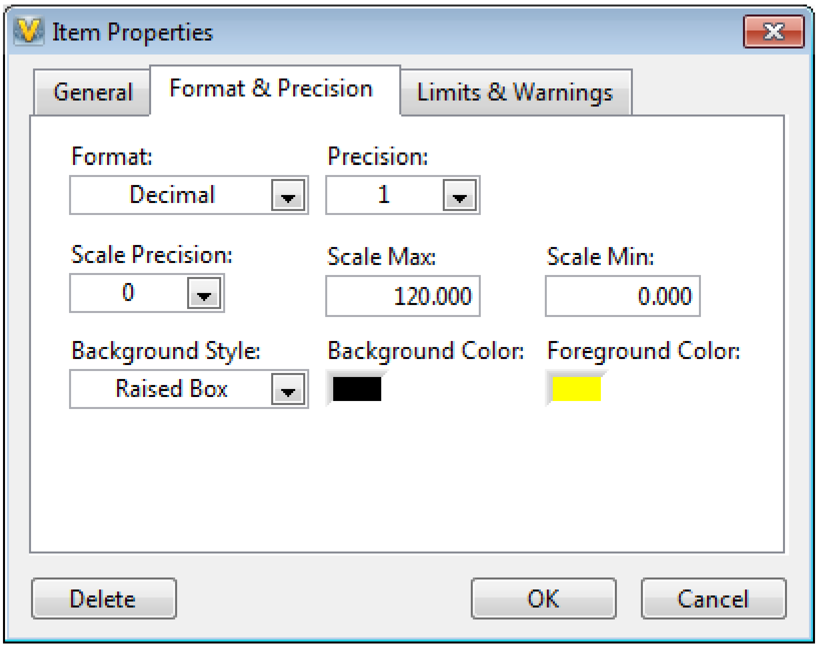

Figure 38: Format & Precision - Gauges

After ending the **Edit Mode** (Screen - Edit Mode) the workspace should look like that:

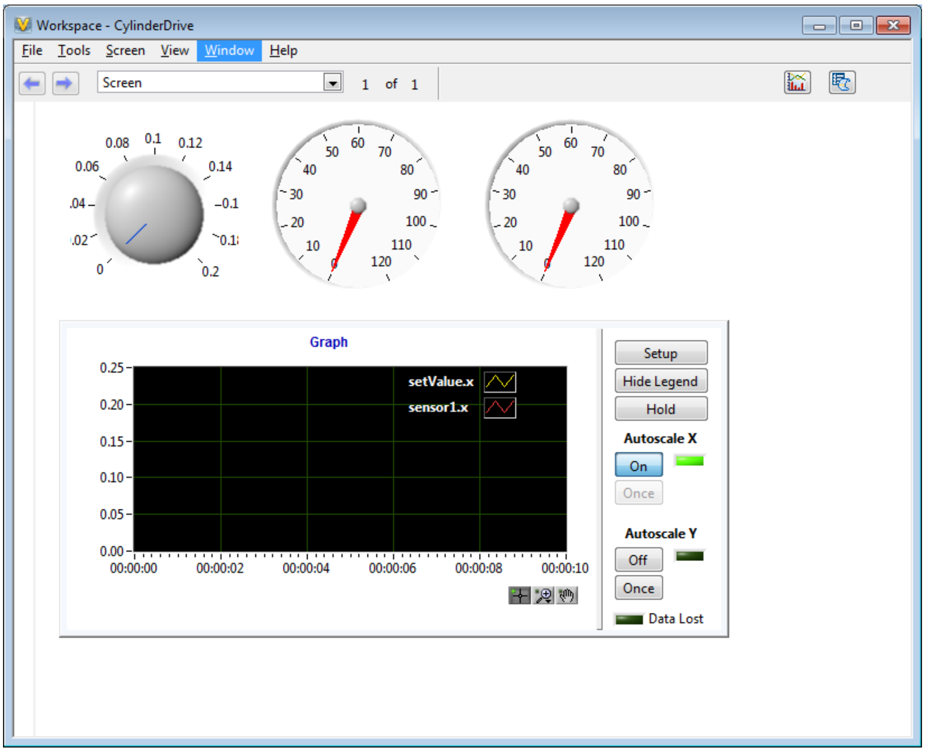

Figure 39: Final Workspace

To **reopen the property dialog** you can use a **right click** on the dial and gauges or press the button **Setup** on the plot element.
Close the Workspace and the Project Explorer to return to the VeriStand main window (Figure 22).

- ## Simulation with VeriStand

You can **start the simulation** with the button **Run Project** on the VeriStand main window (Figure 22).

The simulation model is deployed to the operating system:

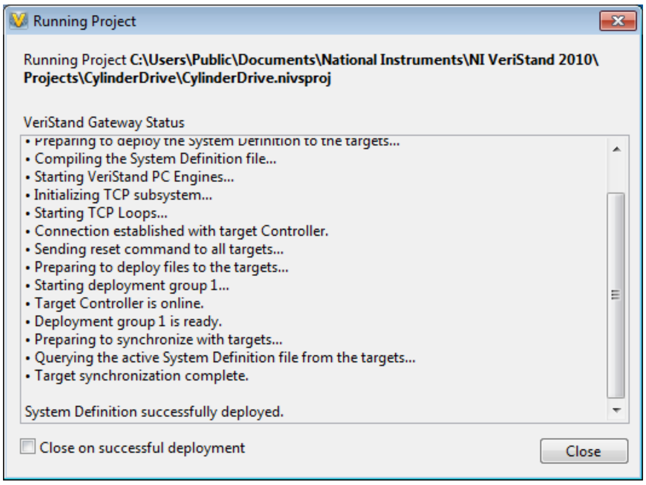

Figure 40: Deploying Process

After closing that window (Figure 40) the workspace is opened and the simulation starts immediately.

Move the dial for the set value of the cylinder position and follow the plot of the set and current position:

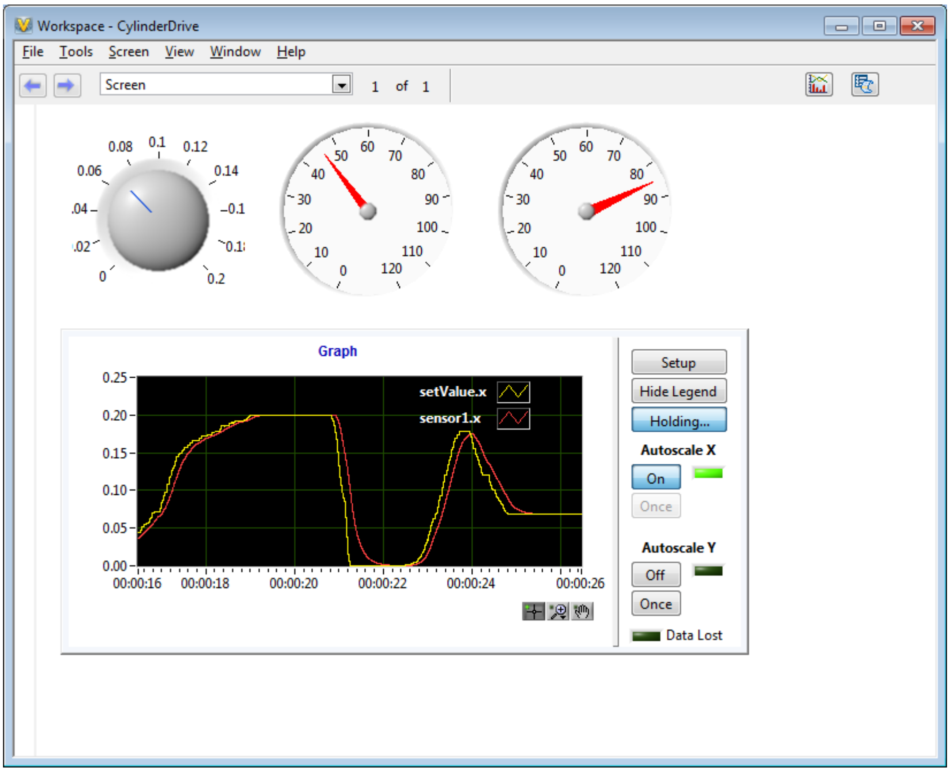

Figure 41: Workspace during simulation

The current position of the cylinder (sensor1.x; red) follows the set position (setValue.x; yellow) with a little delay. The gauges visualize the pressure in volumes vA and vB.

Open the **Model Parameter Manager** (Tools – Model Parameter Manager):

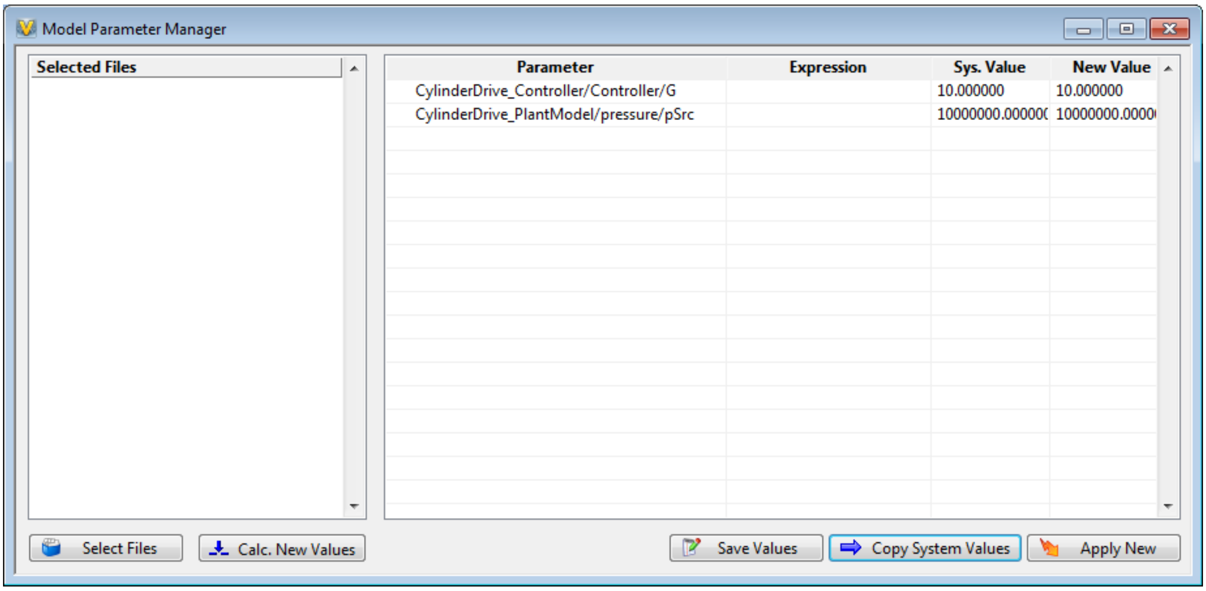

Figure 42: Model Parameter Manager

Here all the imported model parameters are listed. Open the following dialog with a double click on the gain of the controller **G,** change it from **10** to **2** and press the button **OK**:

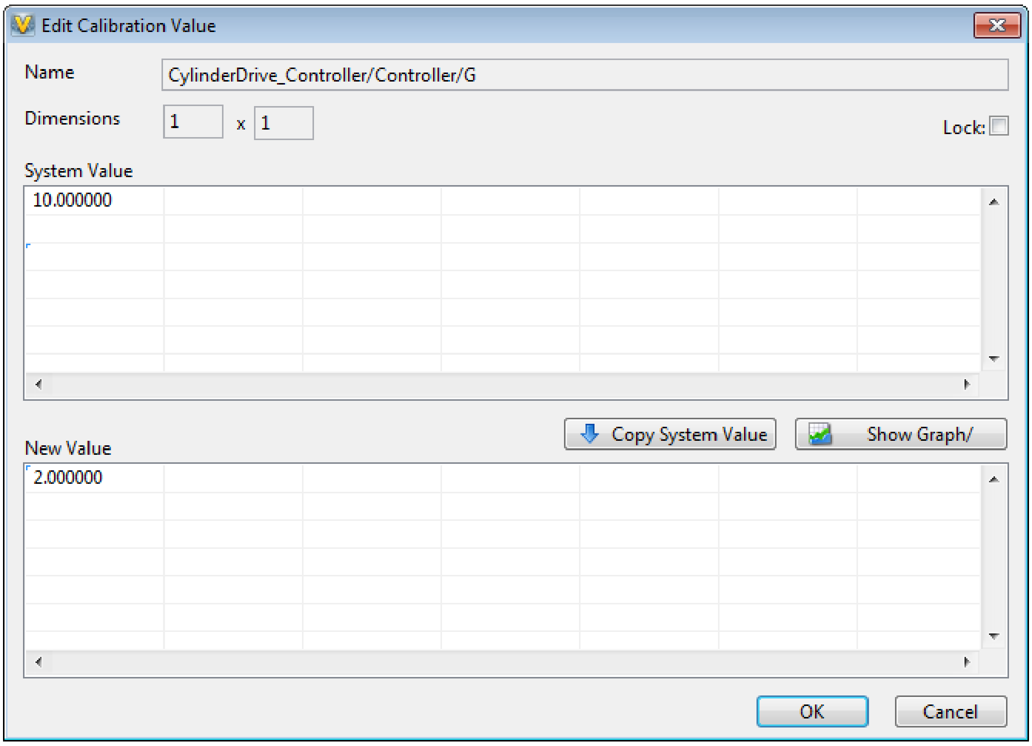

Figure 43: Change of model parameter

Press the button **Apply New (**Figure 42**).** The system value of the parameter G changes to 2.

Close the Model Parameter Manager.

After that parameter change of the controller gain move the dial for the set value again and follow the plot of the set and actual position:

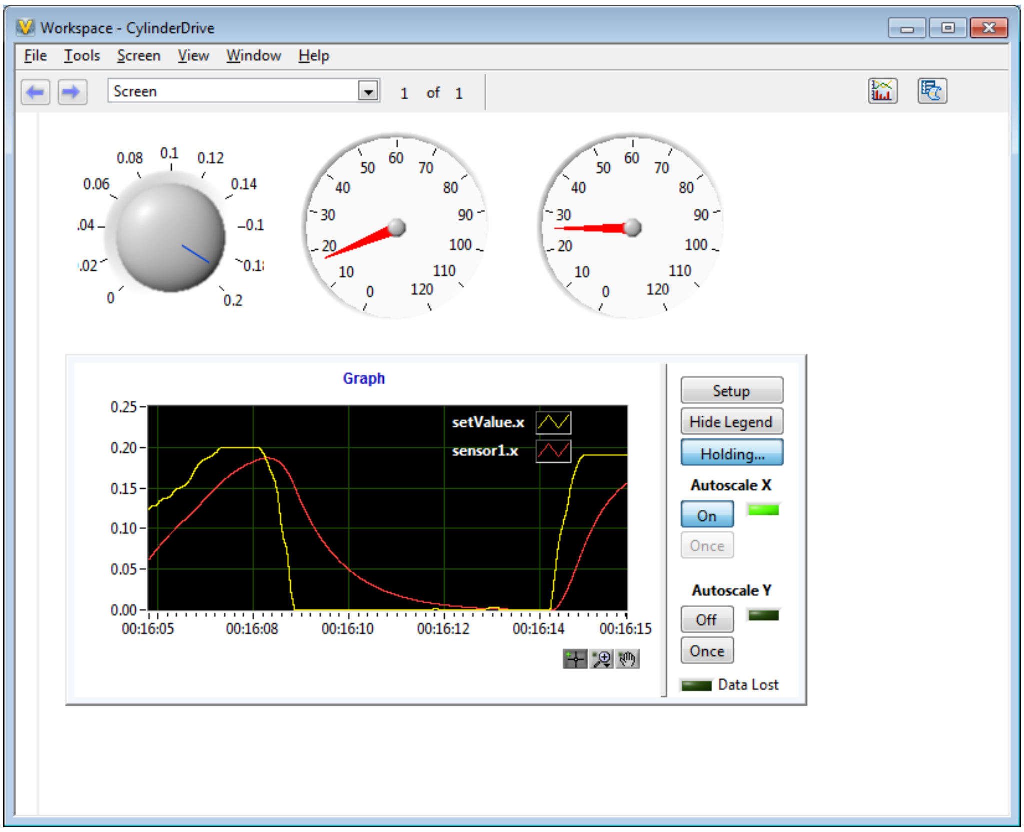

Figure 44: Workspace during simulation after parameter change (G=2)

The delay between current and set position is now much bigger because of the lower gain in the controller.

Summary

Finally, let us resume a few points concerning the benefits of this tutorial:

- You have learnt how to export a SimulationX model using the Code Export Wizard to NI VeriStand. Inputs, outputs, and parameters which should be available in VeriStand can be selected in the Code Export Wizard.
- It was highlighted that a model which will be exported and run in real time should be tested with a fixed step solver for correct results inside SimulationX prior the code export. We touched this topic only with respect to the step size settings. The real time capability of simulation models depends also on their complexity and on the performance of the real time target. Please refer to the SimulationX user manual for more information about real time capable models.
- We have demonstrated how to embed and use such a model inside VeriStand .
- You now know how to generate and use a simple VeriStand Workspace to control the model during simulation run.