



SIEMENS

Ingenuity for life



Industry Online Support

Home

Use Case 2: Connecting Simulink Models to SIMATIC PLCSIM Advanced via OPC UA

SIMATIC S7-PLCSIM Advanced
Simulink

<https://support.industry.siemens.com/cs/ww/en/view/109749187>

Siemens
Industry
Online
Support



Warranty and Liability

Note

The Application Examples are not binding and do not claim to be complete regarding the circuits shown, equipping and any eventuality. The Application Examples do not represent customer-specific solutions. They are only intended to provide support for typical applications. You are responsible for ensuring that the described products are used correctly. These Application Examples do not relieve you of the responsibility to use safe practices in application, installation, operation and maintenance. When using these Application Examples, you recognize that we cannot be made liable for any damage/claims beyond the liability clause described. We reserve the right to make changes to these Application Examples at any time without prior notice.

If there are any deviations between the recommendations provided in these Application Examples and other Siemens publications – e.g. Catalogs – the contents of the other documents have priority.

We do not accept any liability for the information contained in this document. Any claims against us – based on whatever legal reason – resulting from the use of the examples, information, programs, engineering and performance data etc., described in this Application Example shall be excluded. Such an exclusion shall not apply in the case of mandatory liability, e.g. under the German Product Liability Act ("Produkthaftungsgesetz"), in case of intent, gross negligence, or injury of life, body or health, guarantee for the quality of a product, fraudulent concealment of a deficiency or breach of a condition which goes to the root of the contract ("wesentliche Vertragspflichten"). The damages for a breach of a substantial contractual obligation are, however, limited to the foreseeable damage, typical for the type of contract, except in the event of intent or gross negligence or injury to life, body or health. The above provisions do not imply a change of the burden of proof to your detriment.

Any form of duplication or distribution of these Application Examples or excerpts hereof is prohibited without the expressed consent of the Siemens AG.

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks. In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions only form one element of such a concept. Customer is responsible to prevent unauthorized access to its plants, systems, machines and networks. Systems, machines and components should only be connected to the enterprise network or the internet if and to the extent necessary and with appropriate security measures (e.g. use of firewalls and network segmentation) in place. Additionally, Siemens' guidance on appropriate security measures should be taken into account. For more information about industrial security, please visit <http://www.siemens.com/industrialsecurity>.

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends to apply product updates as soon as available and to always use the latest product versions. Use of product versions that are no longer supported, and failure to apply latest updates may increase customer's exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed under <http://www.siemens.com/industrialsecurity>.

Table of Contents

Warranty and Liability	2
1 Introduction	4
1.1 Overview.....	4
1.2 How the application example works.....	5
1.3 Components used	6
2 Engineering	7
2.1 Simulation model in Simulink	7
2.1.1 MATLAB function	7
2.1.2 Inserting the MATLAB function	8
2.2 STEP 7 program in TIA Portal.....	10
2.2.1 Simulation.....	10
2.2.2 Hardware configuration	11
2.2.3 Creating the program	12
2.3 Starting up the sample application	16
2.3.1 Setting up the PLCSIM Virtual Ethernet adapter	16
2.3.2 Starting the virtual controller	18
2.3.3 Downloading the project.....	19
2.3.4 Starting the HMI simulation	20
2.3.5 Starting the Simulink simulation	21
2.4 Operation.....	22
2.4.1 Operating the PID controller.....	22
2.4.2 Monitoring the control result	24
3 Alternative Solution.....	25
4 Appendix	26
4.1 MATLAB function code.....	26
4.2 Service and Support.....	27
4.3 Links and literature	28
4.4 Change documentation	28

1 Introduction

1.1 Overview

In automation and control engineering, the Simulink software from MathWorks is frequently used to simulate processes and create algorithms. The requirement is to simulate the model, algorithm or function in a virtual environment via PLCSIM Advanced or, based on hardware, using a software controller in just a few steps.

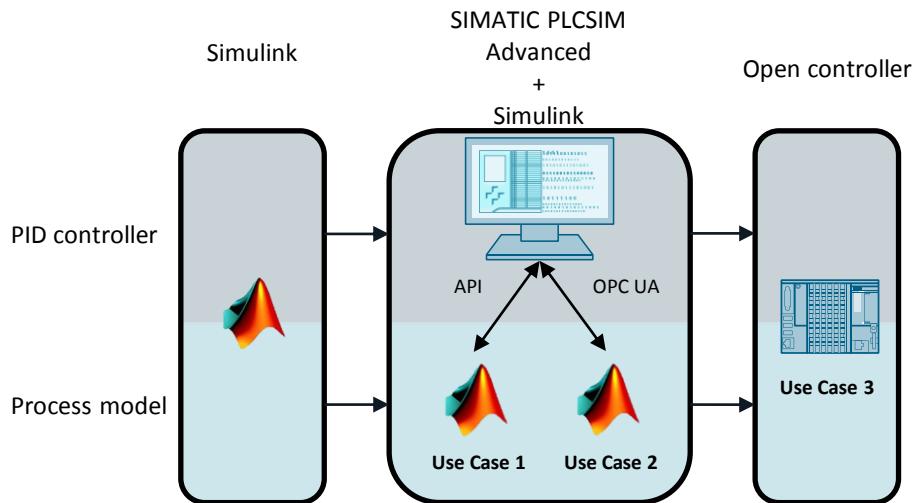
This application example shows you how to create a MATLAB function for communication with a virtual controller via OPC UA. Integrating the function into the Simulink model enables validation and virtual commissioning of a PID controller that runs on a virtual controller in the context of a simulated process in Simulink.

This document describes the steps necessary to configure, start up and use the application example.

The following documents make up the entire application example:

- Main document: Overview of the three use cases and the Simulink model
- Use Case 1: Connecting Simulink models to SIMATIC PLCSIM Advanced via API
- Use Case 2: Connecting Simulink models to SIMATIC PLCSIM Advanced via OPC UA (this document)
- Use Case 3: Using SIMATIC Target 1500S for a Hardware-Based Simulation of the Simulink Model

Figure 1-1: Use cases overview



1.2 How the application example works

The OPC Toolbox add-on for MATLAB allows you to integrate functions into Simulink that allow data exchange with an OPC UA server.

This application example provides you with a fully programmed MATLAB function called “Read_OPC_Func”. This function allows you to read/write tags of a virtual controller of PLCSIM Advanced via OPC UA.

The function is integrated into Simulink using the “Interpreted MATLAB Fcn” block and placed in a control loop instead of a controller.

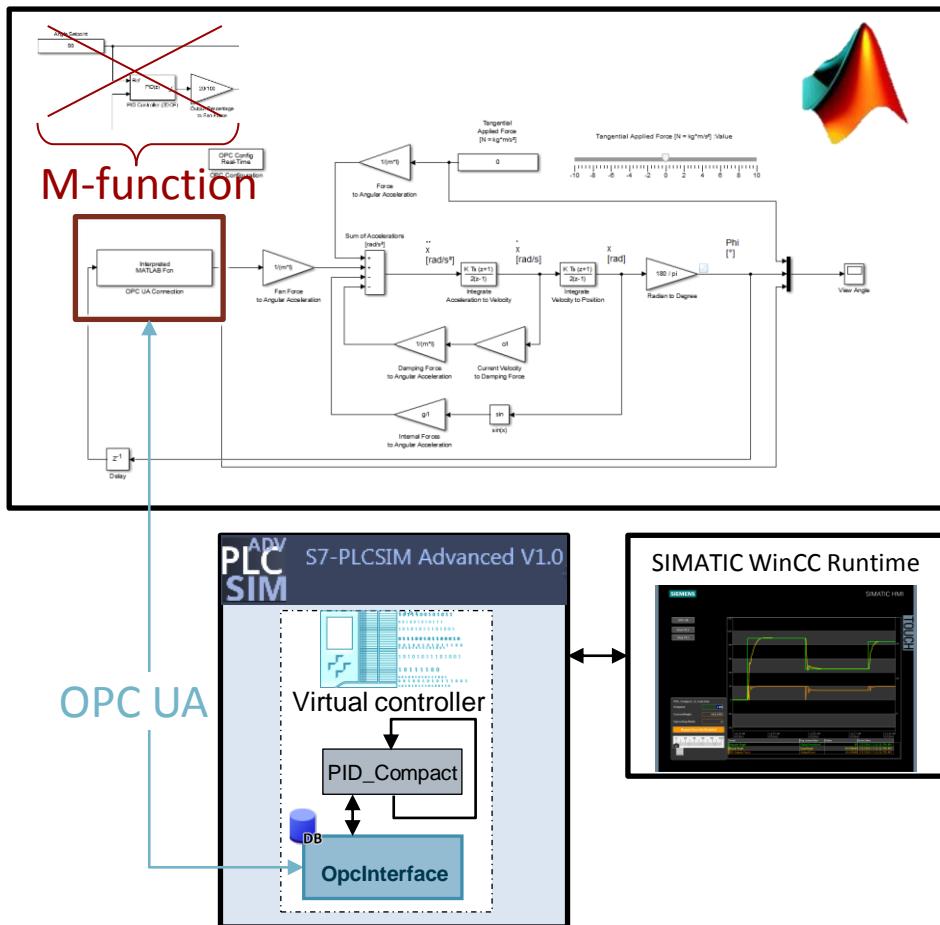
In a TIA Portal project, the “PID_Compact” PID controller is configured and downloaded to the virtual controller in order to test it in the context of the simulation model.

Tags are exchanged via a global data block of the virtual controller. This exchange takes place in the simulation model’s cycle.

The control result and the controller’s control are visualized by a simulated HMI screen of WinCC Runtime.

Note The configuration of the HMI is not part of this application example.

Figure 1-2



1.3 Components used

This application example was created with the following hardware and software components:

Table 1-1: Hardware and software

Component	Article number / note / link
(R2016a) MATLAB V9.0 Simulink V8.7 OPC Toolbox V4.0.1	MathWorks Online Documentation: http://en.mathworks.com/help/
STEP 7 V14 Professional	6ES7822-1..04-.. Manual: https://support.industry.siemens.com/cs/ww/en/view/10974227_2
S7-PLCSIM Advanced V1.0	6ES7823-1FE00-0YA5 Manual: https://support.industry.siemens.com/cs/ww/en/view/10973915_3

This application example consists of the following components:

Table 1-2: Components of the application example

109749187_DIGI_Usecases_OPCTOC_V10_en.pdf	This document.
109749187_DIGI_Usecases_TIA_PROJ_V10_en.zip	TIA Portal project for Use Cases 1-3.
109749187_DIGI_Usecases_Simulink_PROJ_V10_en.zip	Simulink models for Use Cases 1-3.

2 Engineering

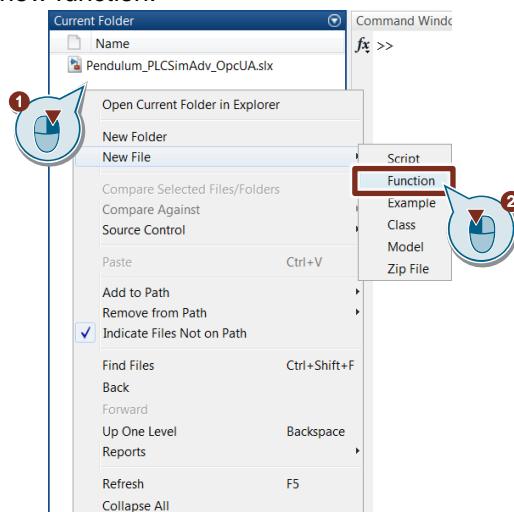
2.1 Simulation model in Simulink

The following chapter describes how to create a user-defined MATLAB function called “Read_OPC_Func” for data exchange with an OPC server in MATLAB and integrate it into the Simulink model using the “Interpreted MATLAB Function” block.

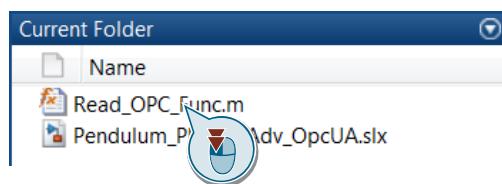
2.1.1 MATLAB function

Perform the following steps to create a MATLAB function.

1. In MATLAB, navigate to the storage location of your Simulink model and insert a new function.



2. Name the function “Read_OPC_Func” and double-click the function to open the Function Editor.



3. In the Function Editor, enter the following code. For a master copy, refer to the Appendix, Chapter [4.1](#).

Customize the following parameters to suit your application:

1. Enter the IP address and the port of the OPC server.
2. Enter the name of the interface DB used for data exchange between the PLC and the model.
3. Enter the name of the tag whose value you want the function to read from the interface DB.
4. Enter the name of the tag whose values you want the function to write to the interface DB.

```

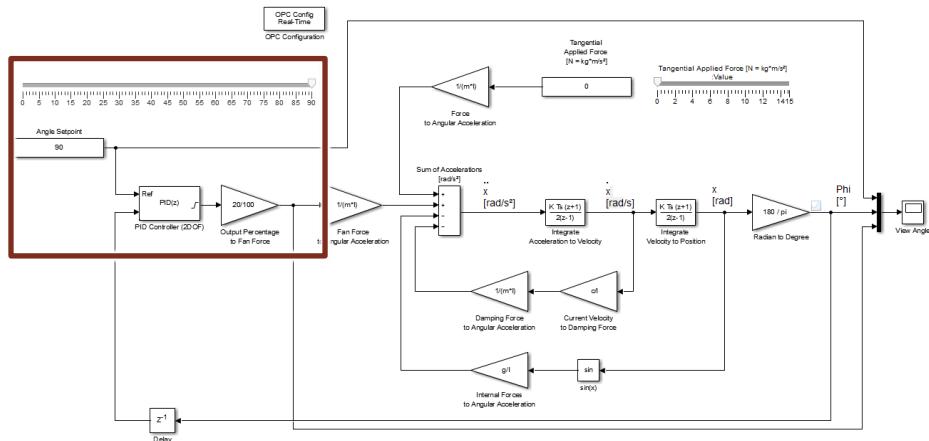
1
2 function [x] = Read_OPC_Func(y)
3 % variables
4 - persistent init_Server;
5 - persistent init_Nodes;
6 - persistent uaClient;
7 - persistent Var_Node_In;
8 - persistent Var_Node_Out;
9 - persistent testVal;
10 % initialize variables
11 if (isempty(init_Server))
12     testVal = 0;
13     init_Server = 0;
14     init_Nodes = 0;
15 end
16 % OPC UA server (PLC) address and connecting client (Simulink) to the server
17 if init_Server == 0
18     init_Server = 1;
19     uaClient = opcua('192.168.1.1',4840); ①
20     connect(uaClient);
21 end
22 % define variable nodes in the server
23 if uaClient.isConnected == 1 && init_Nodes == 0
24     init_Nodes = 1;
25     % find DB "OpcInterface" of the server
26     DB_Node = findNodeByName(uaClient.Namespace, 'OpcInterface', '-once');
27     % find variables "fanForce" and "currentAngle" in the DB "OpcInterface"
28     Var_Node_In = findNodeByName(DB_Node, 'fanForce', '-once'); ②
29     Var_Node_Out = findNodeByName(DB_Node, 'currentAngle', '-once'); ③
30 end
31 % read and write variables of the server
32 if uaClient.isConnected == 1 && init_Nodes == 1
33     % read "fanForce" value from server and store in "val"
34     [val, ~, ~] = readValue(uaClient, Var_Node_In);
35     % assign input y of the function to "currentAngle" variable of server
36     writeValue(uaClient, Var_Node_Out, y);
37     % assign "val" to variable "tesVal"
38     testVal = val;
39 end
40 % assign "fanForce" ("testVal") value to the output x of the function
41 x = double(testVal);
42 end

```

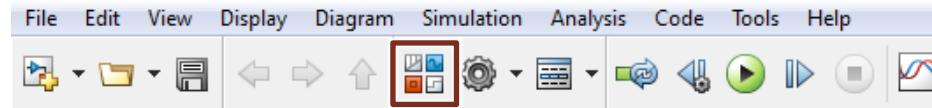
4. Save the created function.

2.1.2 Inserting the MATLAB function

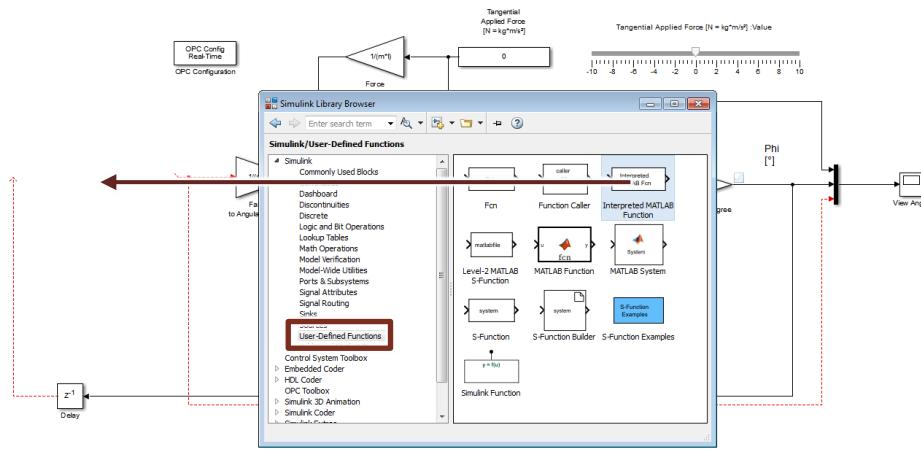
1. Open your Simulink model with the control loop and remove the PID controller and the highlighted functions that are executed with the virtual controller.



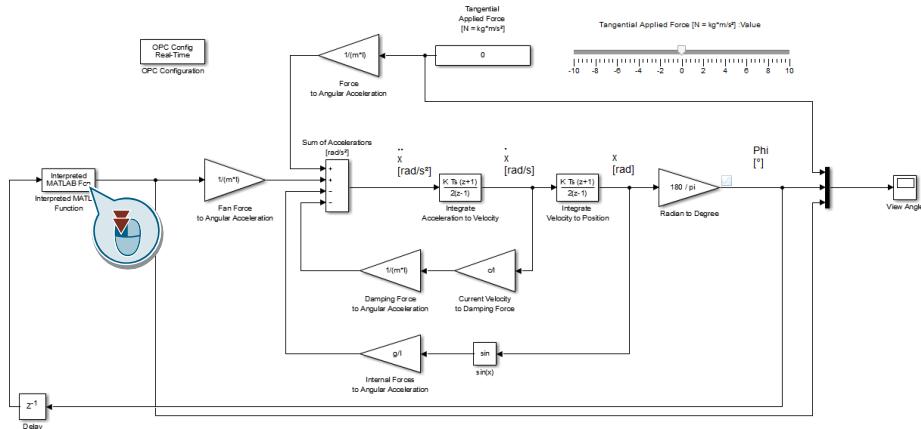
2. Open the Simulink Library Browser.



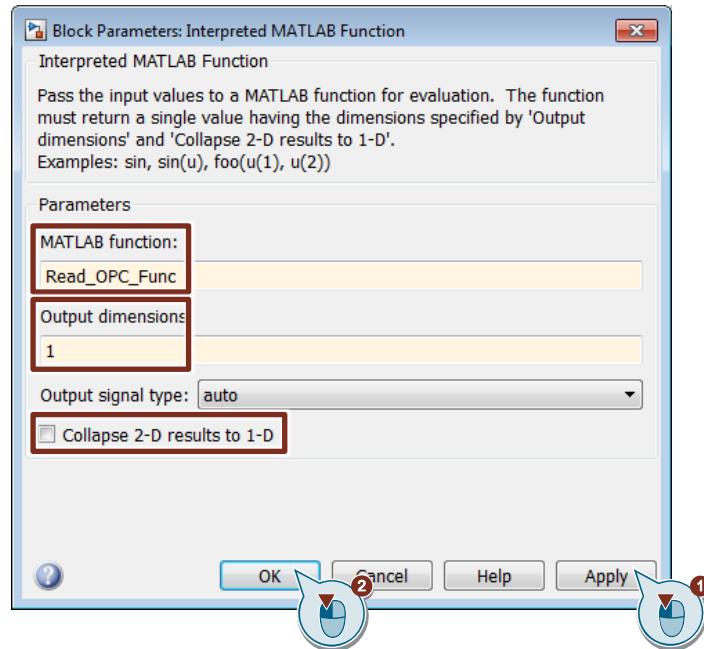
3. Use drag and drop to move the “Interpreted MATLAB Function” block from the “Simulink > User-Defined Functions” directory to the Simulink model and connect it to the model.



4. Double-click to open the block properties of the “Interpreted MATLAB Function” block.



5. As the function name, enter the name of your function, here “Read_OPF_Func”, and set other parameters as shown in the following figure. Click “Apply” to apply the change and select “OK” to confirm.

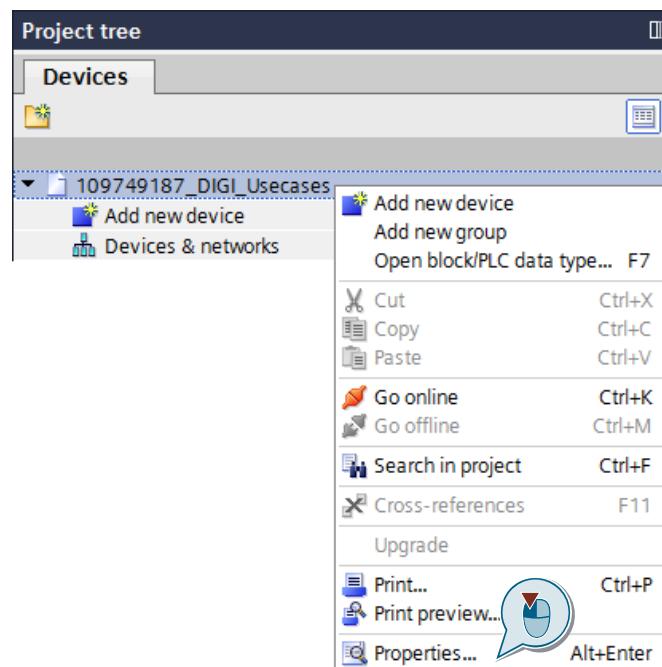


2.2 STEP 7 program in TIA Portal

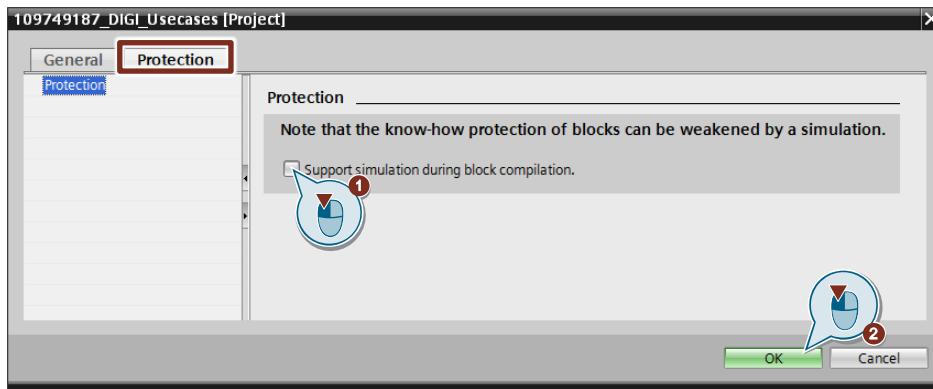
2.2.1 Simulation

Perform the following steps to activate the simulation with PLCSIM Advanced.

1. Create a new TIA Portal project.
2. Open the “Properties...” of your project.



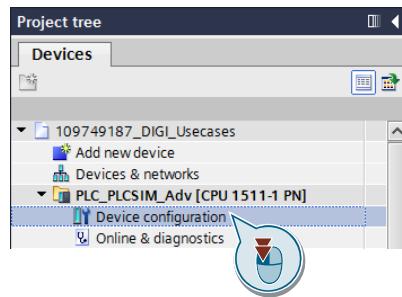
- In the “Protection” tab, check the “Support simulation during block compilation” check box. Click “OK” to confirm the change.



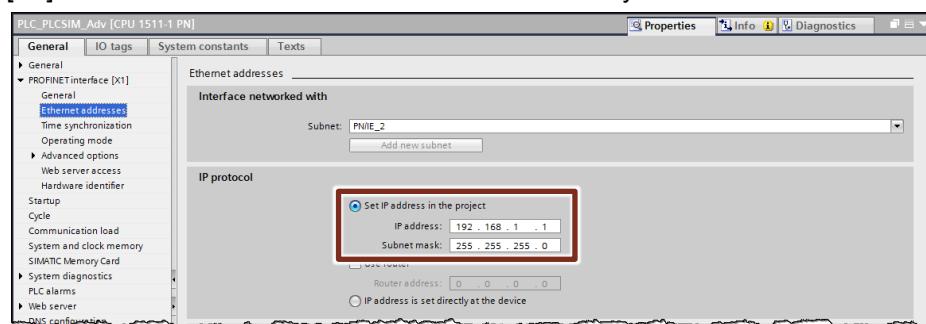
2.2.2 Hardware configuration

Perform the following steps to set up the OPC UA server for your PLC.

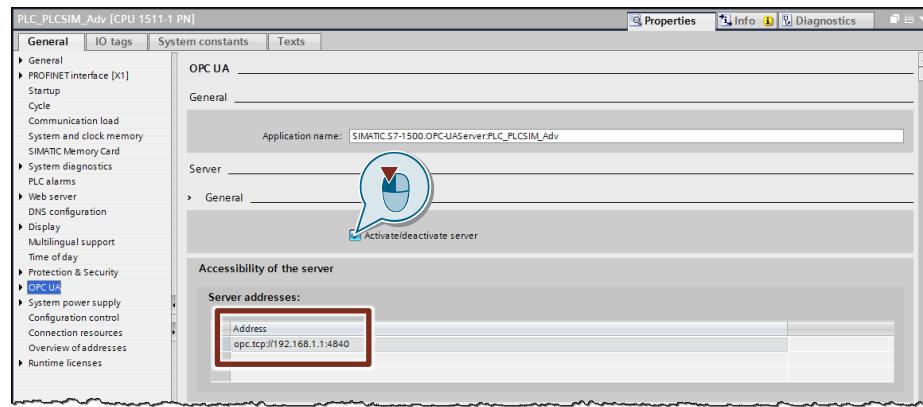
- Insert an S7-1500 CPU with firmware V2.0 into the project. For example, a CPU 1511-1 PN V2.0. Name the PLC “PLC_PLCSIM_Adv”.
- Double-click to open the CPU’s “Device configuration”.



- In the Inspector window, select “Properties > General > PROFINET interface [X1] > Ethernet addresses” to set the IP address for your CPU.



- In “OPC UA”, activate the OPC UA server for your CPU. “Server addresses:” displays the IP address and port number at which the server can be reached.



- In “Runtime licences > OPC UA”, set the purchased license type according to the required license type.

2.2.3 Creating the program

OBs and global DBs

Perform the following steps to create the OBs and global DBs.

- Add the following blocks to your S7 program:
 - Cyclic interrupt OB: “Cyclic20ms” (with a cyclic interrupt time of 20 ms)
 - Global DB: “Global”
 - Global DB: “OpcInterface”
- Open the “Global” DB and add the following tags. The tags are combined by a structure.
 - pendulumControllerOpc [Struct]
 - angleSetpoint [Int]
 - manualOutputPercentage [Int]
 - manualOutputEnable [Bool]
 - operatingMode [Int] (start value = 3)

Global			
	Name	Data type	Start value
1	Static		
2	pendulumControllerOpc	Struct	<input style="width: 20px; height: 20px;" type="button" value="..."/>
3	angleSetpoint	Int	0
4	manualOutputPercentage	Int	0
5	manualOutputEnable	Bool	false
6	operatingMode	Int	3

- Open the “OpcInterface” DB and add the following tags for data exchange with the Simulink model:
 - fanForce [Real]
 - currentAngle [LReal]

OpcInterface			
	Name	Data type	Start value
1	Static		
2	fanForce	Real	0.0
3	currentAngle	LReal	0.0

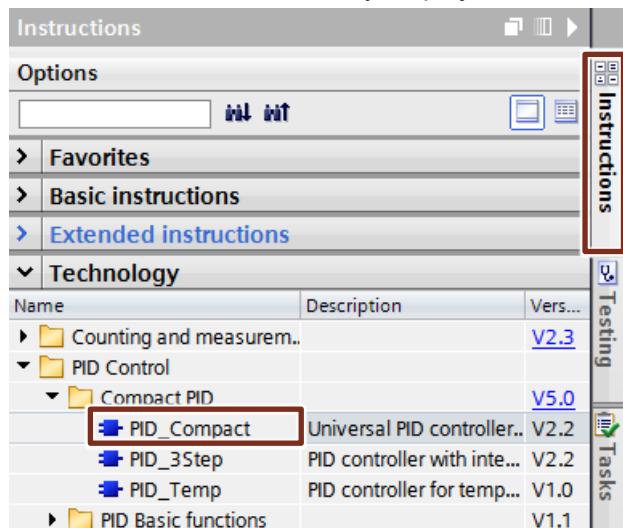
“Cyclic20ms” OB

Perform the following steps to program the “Cyclic20ms” OB.

1. Open the “Cyclic20ms” cyclic interrupt OB.
2. Add a temporary tag called “tempOutputPercentage”.

Cyclic20ms		
	Name	Data type
1	Input	
2	Temp	
3	tempOutputPercentage	Real

3. In “Technology > PID Control > Compact PID”, drag the “PID_Compact” controller from “Instructions” to your project.



4. Name the instance DB “InstPIDCompactOPC” and interconnect the instruction as shown in the following figure.

```
// PID controller for controlling simulink model via OPC UA
"InstPIDCompactOPC"(Setpoint := "Global".pendulumControllerOpc.angleSetpoint,
                     Input := LREAL_TO_REAL("OpcInterface".currentAngle),
                     ManualEnable := "Global".pendulumControllerOpc.manualOutputEnable,
                     ManualValue := "Global".pendulumControllerOpc.manualOutputPercentage,
                     ModeActivate := TRUE,
                     Output => #tempOutputPercentage,
                     Mode := "Global".pendulumControllerOpc.operatingMode);
```

5. Scale the PID controller's output percentage to the maximum force of 20 Newton.

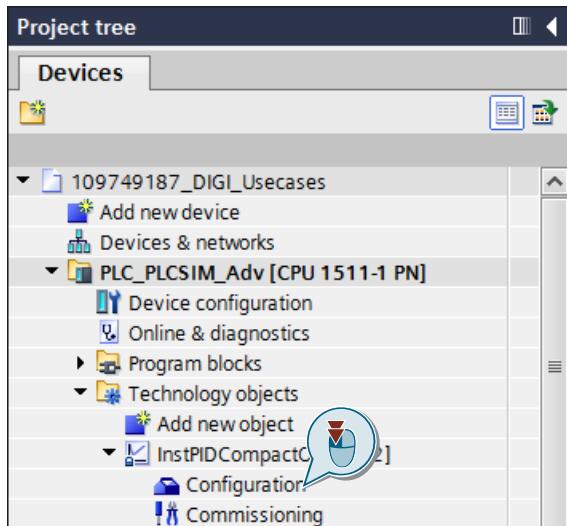
```
// Scaling the output of PID controller (0..100 %) to maximum power (0..20 N)
"OpcInterface".fanForce := #tempOutputPercentage * 20 / 100;
```

Configuring the PID controller

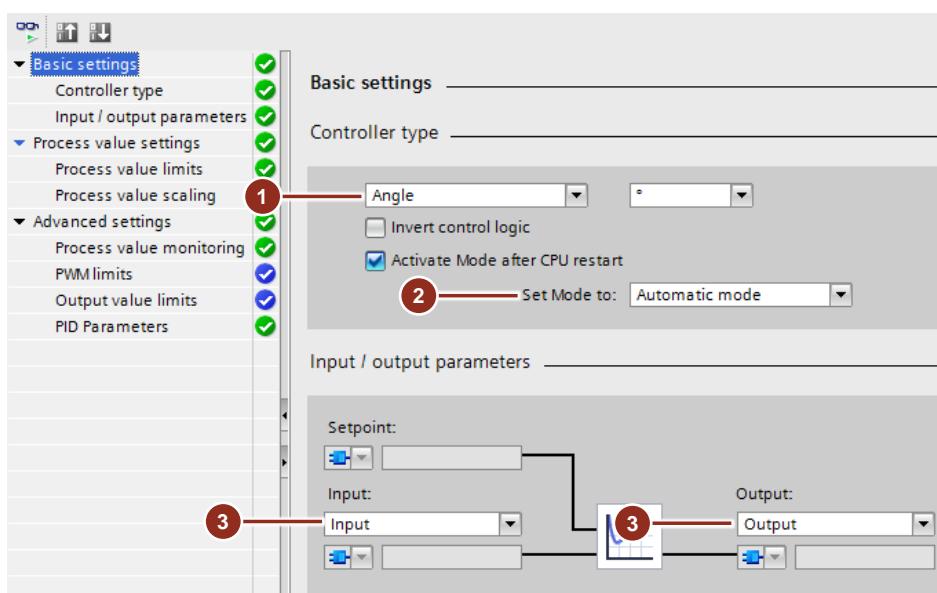
Note For the calculation of the PID parameters from step 5, refer to the main document:
 “109749187_DIGI_Usecases_MAIN_DOC_V10_en.pdf”

Perform the following steps to configure the PID controller.

1. In the project tree, open the “Configuration” of the “InstPIDCompactOPC” instance of the PID controller.

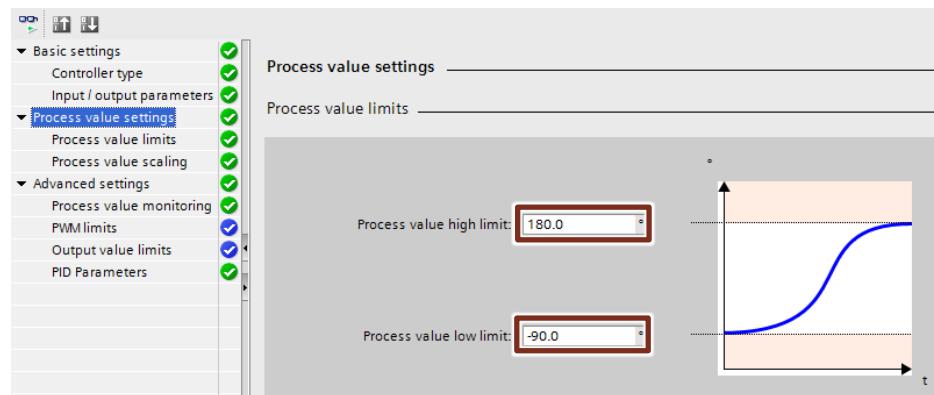


2. Make the basic settings.
 1. In “Controller type”, select “Angle” as the physical quantity.
 2. Set the mode to “Automatic mode”.
 3. Select “Input” for the input parameter and “Output” for the output parameter.

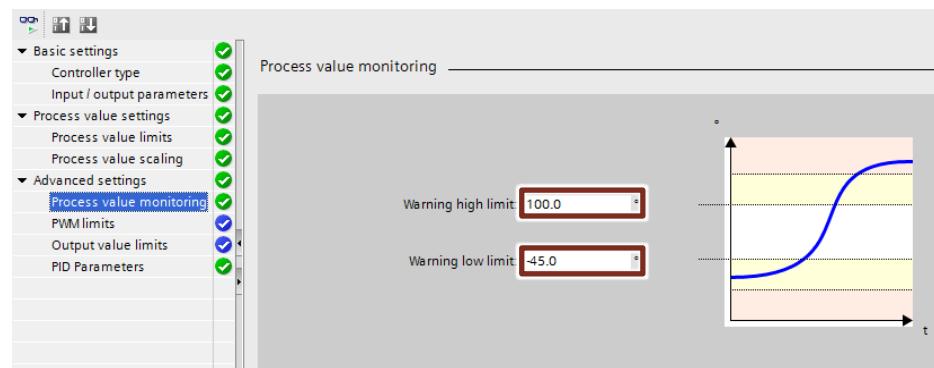


3. Make the process value settings.

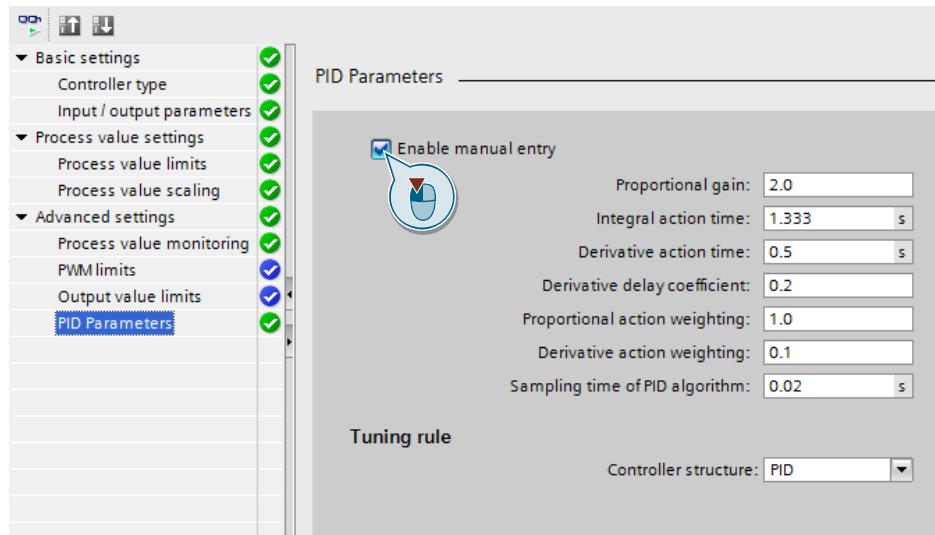
For the process value high limit, set 180.0. For the process value low limit, select -90.0.



4. In the advanced settings for process value monitoring, set 100.0 as the warning high limit and -45.0 as the warning low limit.



5. In the “Advanced settings”, set the PID parameters.
To do this, check “Enable manual entry” and set the following PID parameters.
 - Proportional gain = 2
 - Integral action time = 1.333
 - Derivative action time = 0.5
 - Derivative delay coefficient = 0.2
 - Proportional action weighting = 1.0
 - Derivative action weighting = 0.1
 - Sampling time of PID algorithm = 0.02

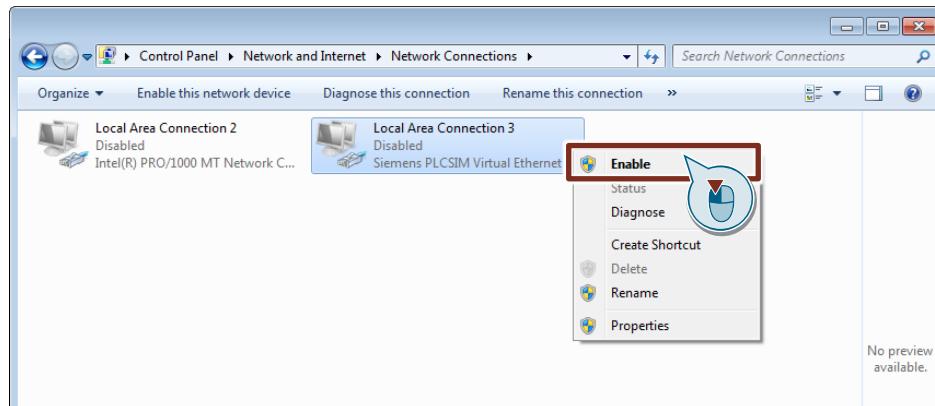


2.3 Starting up the sample application

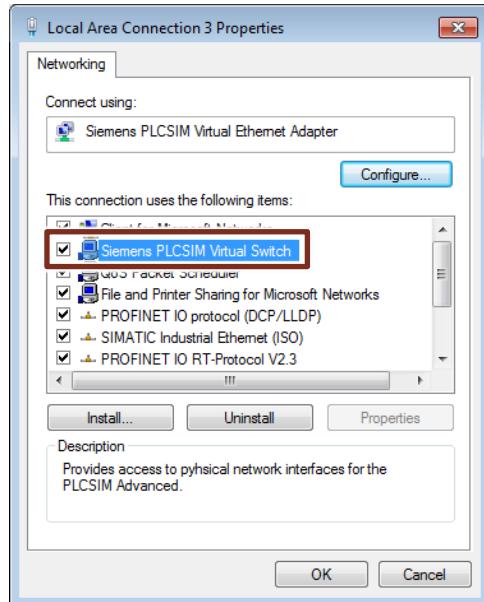
Perform the following steps to start up the simulation with the supplied TIA Portal project and the supplied Simulink model. The OPC function has already been inserted into the Simulink model.

2.3.1 Setting up the PLCSIM Virtual Ethernet adapter

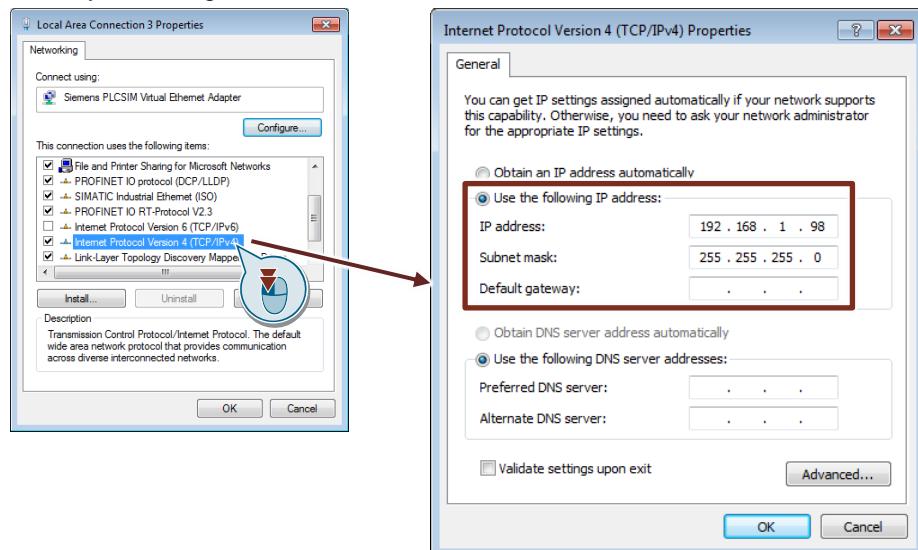
1. From your computer’s Control Panel, open Network Connections and from the interface’s context menu, select “Enable” to enable the “Siemens PLCSIM Virtual Ethernet Adapter” interface.



2. In the interface properties, check the “Siemens PLCSIM Virtual Switch” check box.



3. Open the IPv4 setting and set an available IP address in the same subnet where your configured CPU is located.

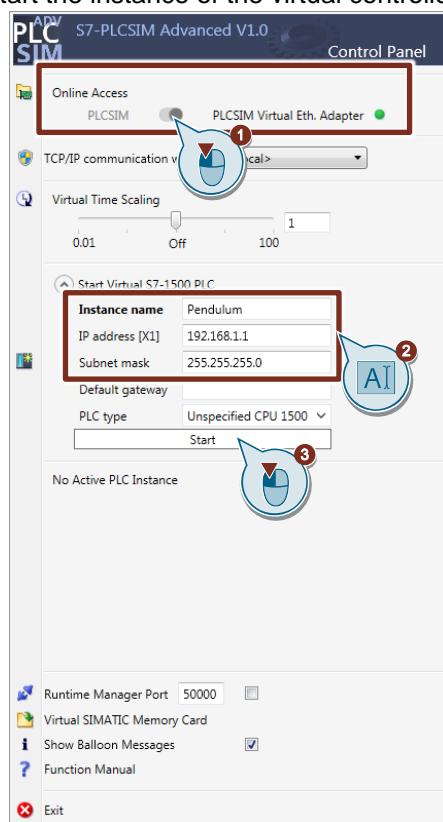


2.3.2 Starting the virtual controller

1. Start PLCSIM Advanced V1.0. The info part of the Windows taskbar displays the PLCSIM Advanced icon. Left-click the icon to open the graphical user interface of PLCSIM Advanced, the Control Panel.

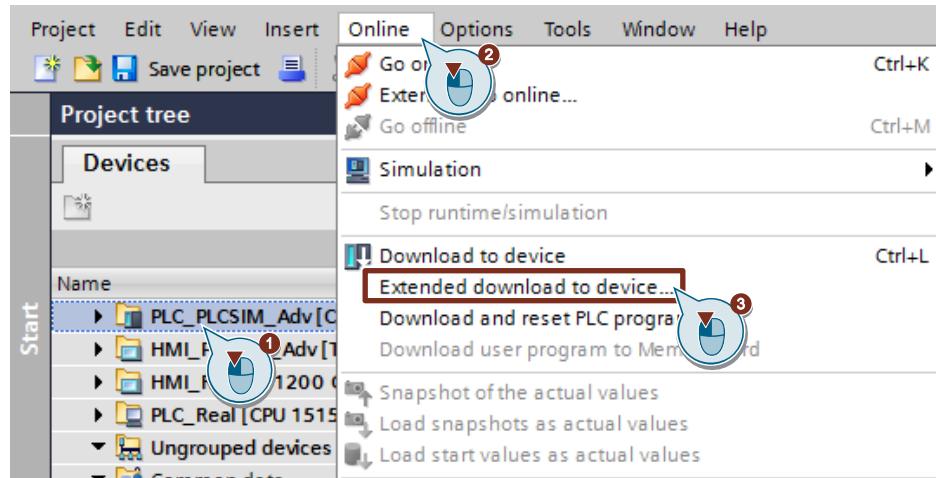


2. 1. In the Control Panel, set "PLCSIM Virtual Eth. Adapter" as the interface.
2. Enter an instance name (e.g., Pendulum) and the IP address/subnet mask for the virtual controller. The IP address corresponds to the CPU configuration in TIA Portal (here: 192.168.1.1).
3. Start the instance of the virtual controller.

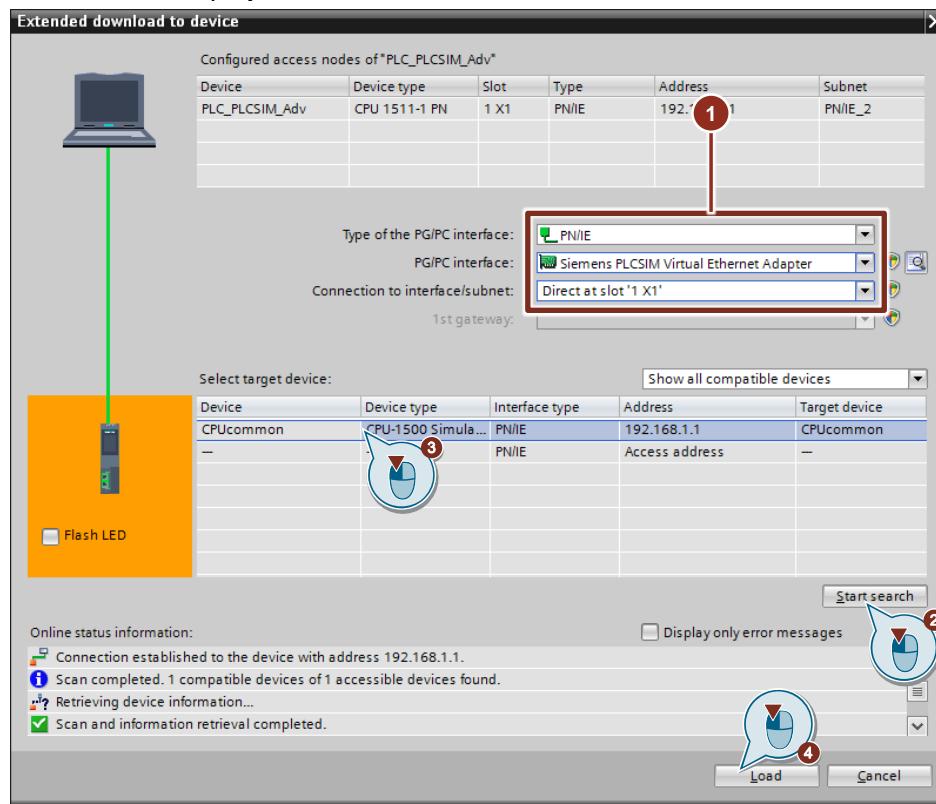


2.3.3 Downloading the project

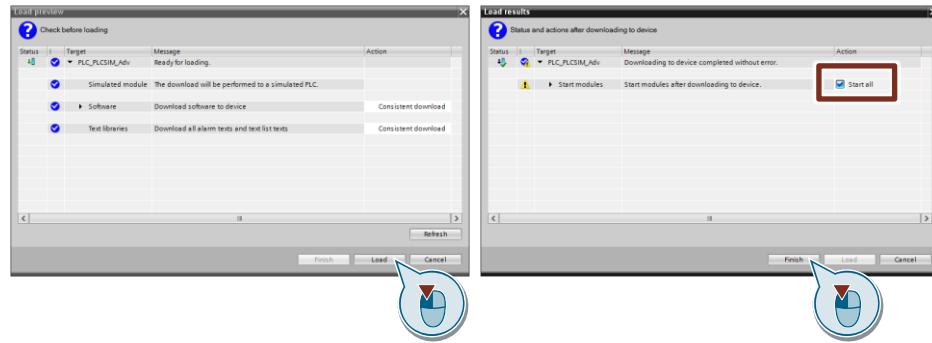
1. Open the sample project or your TIA Portal project.
2. In the project tree, select the “PLC_PLCSIM_Adv” controller and from the TIA Portal menu bar, select “Online > Extended download to device...”.



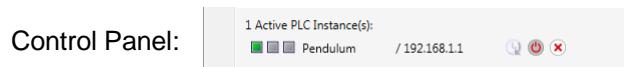
3. 1. Set the interface to the controller.
2. Start the search.
3. From the list, select the virtual controller that has been found.
4. Download the project.



4. Follow other TIA Portal messages and start the controller

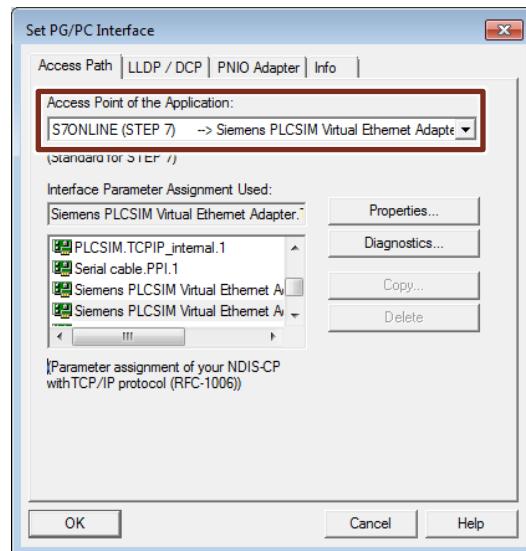


The virtual controller is set to “RUN”.

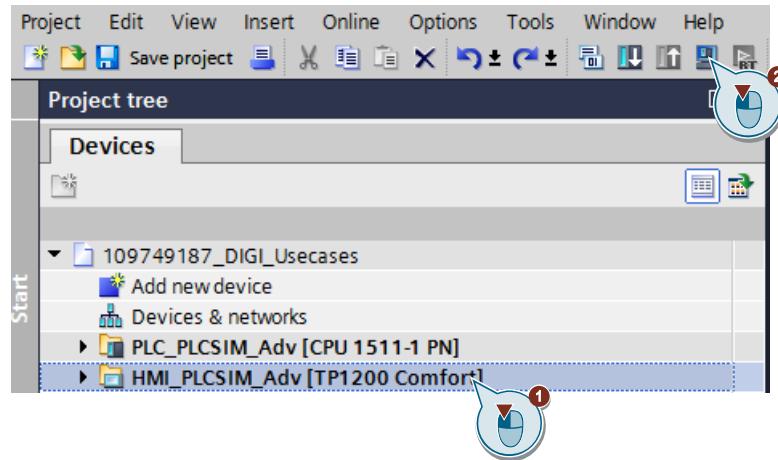


2.3.4 Starting the HMI simulation

- In the Control Panel, select “Set PG/PC Interface” and set “Siemens PLCSIM Virtual Ethernet Adapter.TCPIP.1” as the interface for the HMI connection.

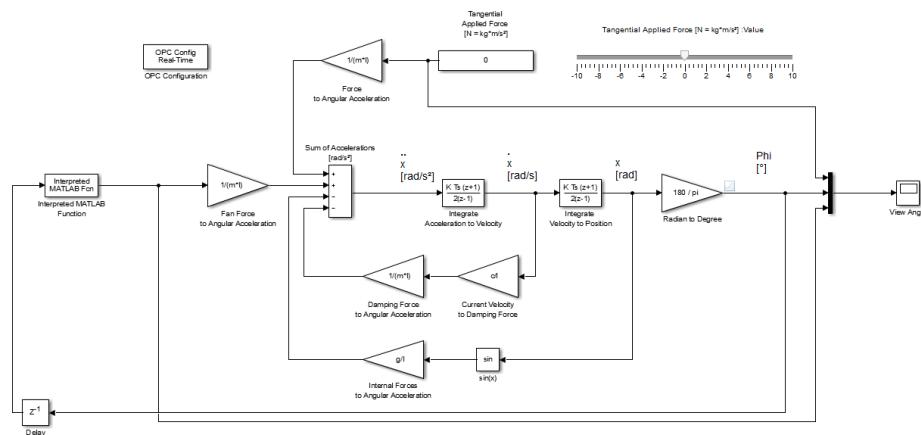


2. In TIA Portal, start the HMI's WinCC Runtime simulation for monitoring the control result and operating the PID controller:
 1. In the project tree, select the “HMI_PLCSIM_Adv” HMI.
 2. Click “Start simulation”.

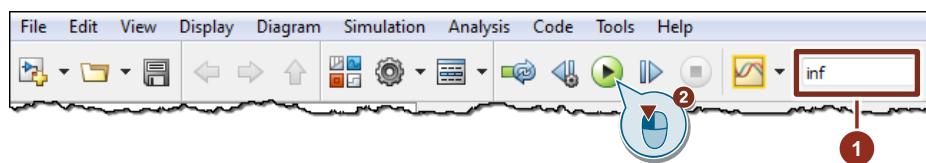


2.3.5 Starting the Simulink simulation

1. Open the supplied “Pendulum_PLCSimAdv_OpcUA.slx” Simulink model.



2. Start the simulation of the Simulink model.
 1. Set the simulation duration to “inf” (infinite).
 2. Click the “Run” icon.

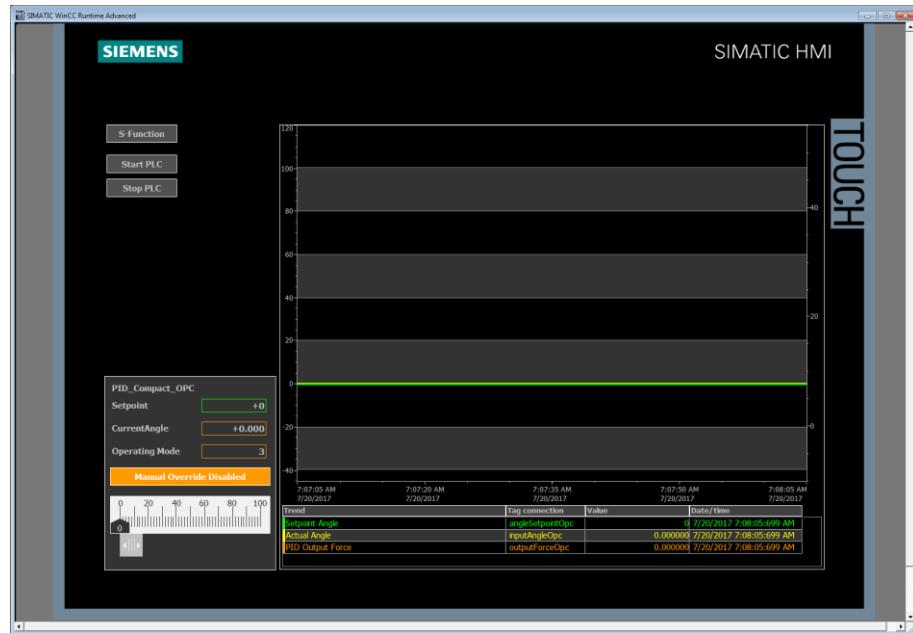


2.4 Operation

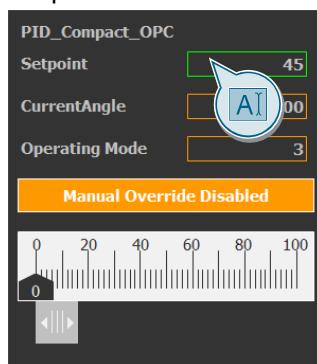
2.4.1 Operating the PID controller

Setting the setpoint

1. Open the window where WinCC Runtime has started.

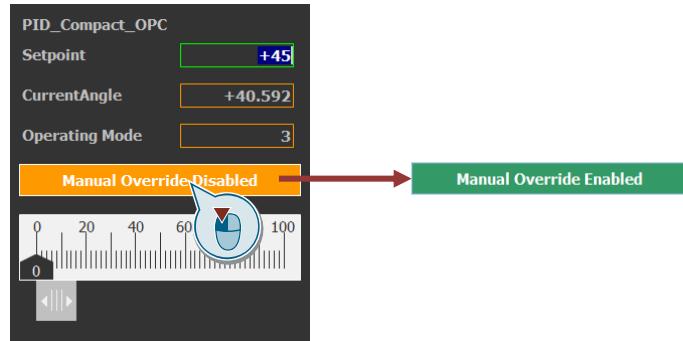


2. Click the “Setpoint” input field and enter a setpoint for the angle of deflection of the pendulum. Press Enter to confirm your entries.

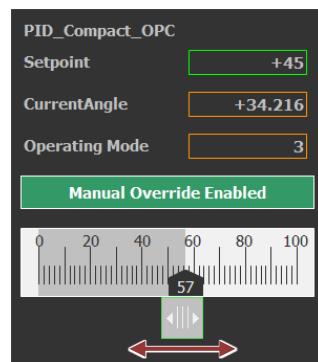


Manually overriding the output

1. In the window where WinCC Runtime has started, click “Manual Override Disabled” to enable manual override.



2. Move the slider to manually override the PID controller's output.

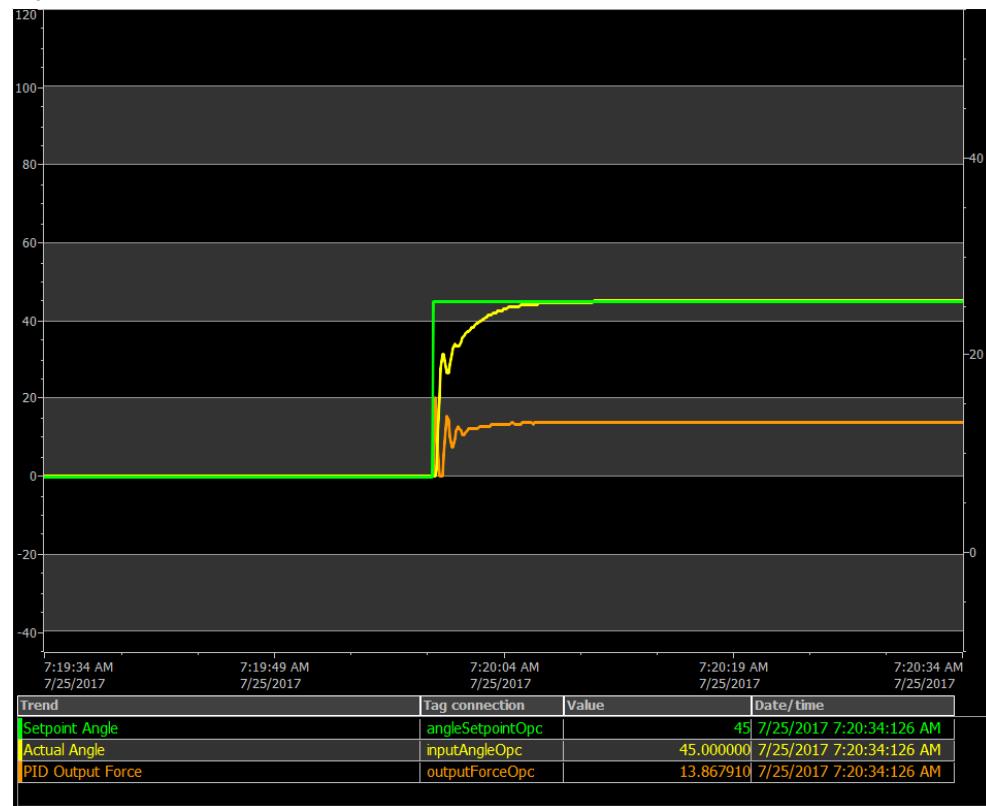


2.4.2 Monitoring the control result

Two options are available for monitoring the control result.

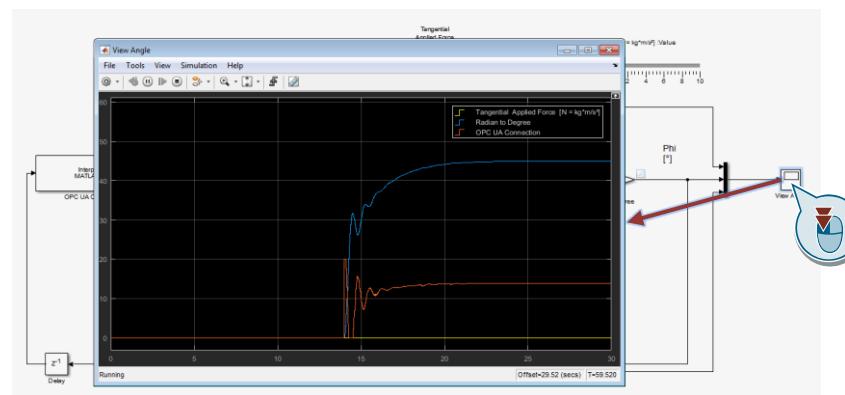
On the one hand, the trace in the HMI's simulation graphically represents the trend of the setpoint, actual value (angle of deflection) and output (force) of the PID controller. The figure shows a trend for a set setpoint angle of 45°.

Figure 2-1: Simulation of the HMI



On the other hand, you can monitor the control result in a display window in Simulink. Double-clicking the “Scope” function in the Simulink model opens the display window.

Figure 2-2: “Scope” function



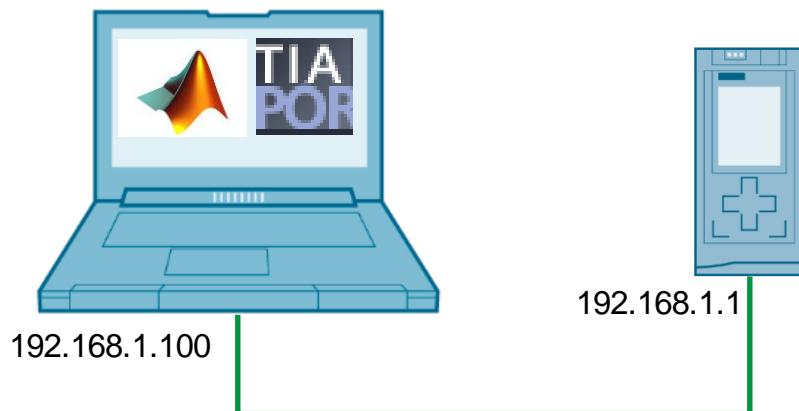
3 Alternative Solution

Alternatively, you can download your program and the hardware configuration to a real CPU and test the function of the PID controller in conjunction with the simulation model.

Connect the PG/PC where the Simulink model with the integrated “Read_OPc_Func” MATLAB function is running to the PROFINET interface of the CPU. Communication via OPC UA requires a Runtime licence for OPC UA.

Note For the HMI simulation, set the PG/PC interface in the Control Panel to your PG/PC’s interface to the CPU.

Figure 3-1: Overview of the alternative solution



4 Appendix

4.1 MATLAB function code

```

function [x] = Read_OPc_Func(y)
% variables
persistent init_Server;
persistent init_Nodes;
persistent uaClient;
persistent Var_Node_In;
persistent Var_Node_Out;
persistent testVal;
%initialize variables
if (isempty(init_Server))
    testVal = 0;
    init_Server = 0;
    init_Nodes = 0;
end
%OPC UA server (PLC) address and connecting client (Simulink) to the
server
if init_Server == 0
    init_Server = 1;
    uaClient = opcua('192.168.1.1',4840);
    connect(uaClient);
end
%define variable nodes in the server
if uaClient.isConnected == 1 && init_Nodes == 0
    init_Nodes = 1;
    %find DB "OpcInterface" of the server
    DB_Node = findNodeByName(uaClient.Namespace, 'OpcInterface', '-once');
    %find variables "fanForce" and "currentAngle" in the DB
    "OpcInterface"
    Var_Node_In = findNodeByName(DB_Node, 'fanForce', '-once');
    Var_Node_Out = findNodeByName(DB_Node, 'currentAngle', '-once');
end
%read and write variables of the server
if uaClient.isConnected == 1 && init_Nodes == 1
    %read "fanForce" value from server and store in "val"
    [val, ~, ~] = readValue(uaClient, Var_Node_In);
    %assign input y of the function to "currentAngle" variable
    writeValue(uaClient, Var_Node_Out, y);
    %assign "val" to variable "testVal"
    testVal = val;
end
%assign "fanForce" ("testVal") value to the output x of the function
x = double(testVal);
end

```

4.2 Service and Support

Industry Online Support

Do you have any questions or do you need support?

With Industry Online Support, our complete service and support know-how and services are available to you 24/7.

Industry Online Support is the place to go to for information about our products, solutions and services.

Product Information, Manuals, Downloads, FAQs and Application Examples – all the information can be accessed with just a few clicks:

<https://support.industry.siemens.com>

SITRAIN – Training for Industry

Well-trained employees are a crucial factor in any company's success. Skills development and expert knowledge make companies competitive and innovative. With our globally available training courses for industry, we help you achieve these goals – with practical experience, innovative learning methods, and a concept that's tailored to the customer's specific needs.

www.siemens.com/sitrain

Technical Support

Siemens Industry's Technical Support offers you fast and competent support for any technical queries you may have, including numerous tailor-made offerings ranging from basic support to custom support contracts.

You can use the web form below to send queries to Technical Support:

www.siemens.com/industry/supportrequest.

Service offer

Our service offer includes the following services:

- Product Training
- Plant Data Services
- Spare Part Services
- Repair Services
- Field & Maintenance Services
- Retrofit & Modernization Services
- Service Programs & Agreements

For detailed information about our service offer, please refer to the Service Catalog:

<https://support.industry.siemens.com/cs/sc>

Industry Online Support app

The "Siemens Industry Online Support" app provides you with optimum support while on the go. The app is available for Apple iOS, Android and Windows Phone:

<https://support.industry.siemens.com/cs/ww/en/sc/2067>

4.3 Links and literature

Table 4-1

No.	Topic
\1\	Siemens Industry Online Support https://support.industry.siemens.com
\2\	Link to the entry page of the application example https://support.industry.siemens.com/cs/ww/en/view/109749187
\3\	Manual: SIMATIC S7-PLCSIM Advanced https://support.industry.siemens.com/cs/ww/en/view/109739153
\4\	MathWorks Online Documentation: http://en.mathworks.com/help/
\5\	Manual: STEP 7 V14 Professional https://support.industry.siemens.com/cs/ww/en/view/109742272

4.4 Change documentation

Table 4-2

Version	Date	Modifications
V1.0	12/2017	First version