PID Control with Gain Scheduling and PID Tuning

SIMATIC PCS 7

Application Example • October 2009

Applikationen & Tools

Answers for industry.

SIEMENS

Reference to Automation and Drives Service & Support

This article is from the internet application portal of the Industry Automation and Drives Technologies Service & Support. Clicking the link below directly displays the download page of this document.

http://support.automation.siemens.com/WW/view/en/38755162

In case of questions to this entry please contact us via the following e-mail adress:

online-support.automation@siemens.com

SIEMENS

SIMATIC
Gain Scheduliung

Application Example

Warranty, liability and support

Note The application examples are not binding and do not claim to be complete re-garding the circuits shown, equipping and any eventuality. The application ex-amples do not represent customer-specific solutions. They are only intended to pro-vide support for typical applications. You are responsible in ensuring that the de-scribed products are correctly used. These application examples do not relieve you of the responsibility in safely and professionally using, installing, operating and servicing equipment. When using these application examples, you recognize that Siemens 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 - then 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”). However, claims arising from a breach of a condition which goes to the root of the contract shall be limited to the foreseeable damage which is intrinsic to the contract, unless caused by intent or gross negligence or based on mandatory liability for injury of life, body or health The above provisions does not imply a change in the burden of proof to your detriment.

Table of Contents

Preface 1

1.1 Objective of the Application

The objective is fast and tight control of processes with non linear behaviour, which requires different control parameters for each operating point. Using gain scheduling, i.e. an operating-point-oriented parameter control, the quality of control can be improved significantly. The scheduling implies a feedforward adaptation of parameters according to a defined schedule in opposite to an adaptive controller, which permanently identifies its process model based on new measurement data.

The application example considered here shows a control loop with PID controller and gain scheduling based on the corresponding process tag type of the SIMATIC PCS 7 Advanced Process Library.

1.2 The following issues are discussed in this application

The following main points are discussed in this application note:

- How to create an instance and set the parameters of the process tag type.
- How to determine the process model and the control parameters. The usage of the PCS 7 PID Tuner is explained firstly for linear processes, and afterwards for a nonlinear process.
- Benchmark simulations with and without gain scheduling, to show the potential benefits.

Validity

… valid for PCS 7 V7.1, in principal transferable to V7.0 from SP1.

Basic Principles of Gain Scheduling 2

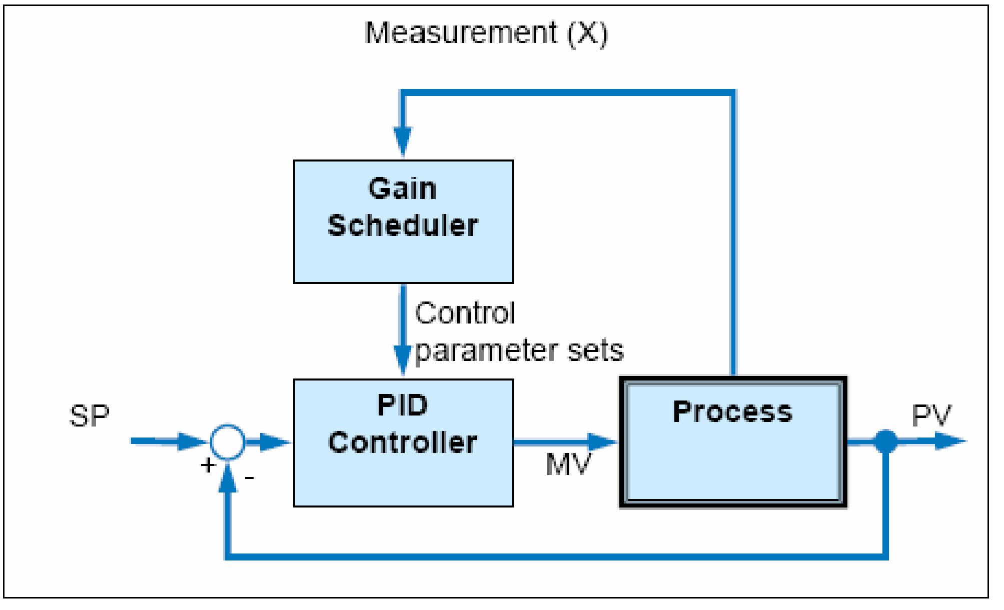
Note A general overview of the APC functionalities (Advanced Process Control) of PCS 7 is provided by the White Paper „How to improve the Performance of your Plant using the appropriate tools of SIMATIC PCS 7 APC-Portfolio?“ (see link \3\ in chapter 8 “Internet Links”)

2.1 Area of Application

Many processes have a non-linear behaviour due to non-linear physical, chemical or thermodynamic effects. When such a process needs to be kept in the close vicinity of a fixed operating point, the transfer function can be linearized around this operating point. A linear PID controller can be designed for this linearized transfer function. If, however, the process has a strongly non-linear response and/or is running at different operating points, no uniformly good control response can be expected throughout the entire operating range. Due to the non-linearity, various gain factors or process time constants are in effect at different operating points. Therefore different controller parameters will be considered to be optimal at different operating points.

2.2 Principle of Operation

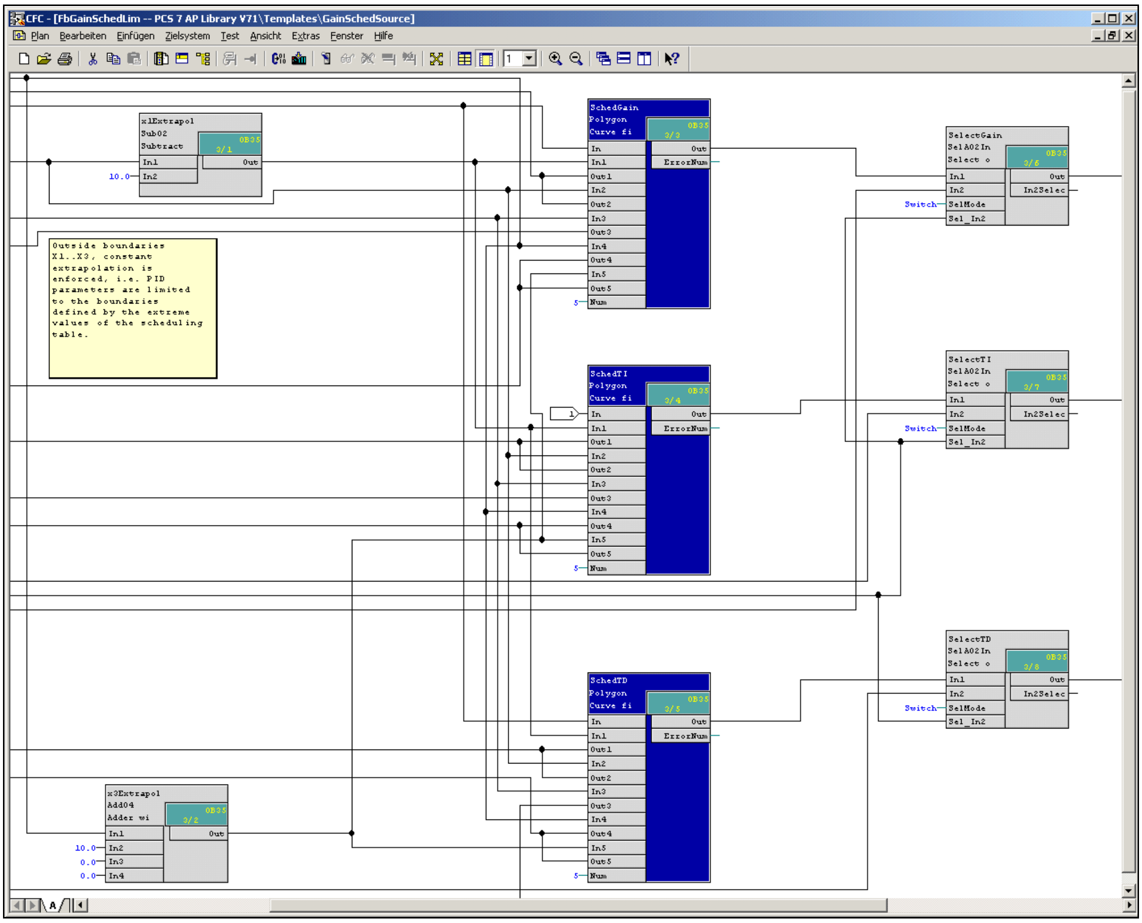
Figure 2-1 Gain Scheduling



One possible (the simplest) solution to this problem is known as gain scheduling or parameter scheduling. Using a tool such as the PCS 7 PID Tuner, various experiments are performed at different operating points, in each case with low signal amplitudes. This results in different PID parameter sets for each operating point. Up to three such parameter sets can be stored in the gain scheduling function block (GainSched). The suitable parameter set is selected depending on a continuously measurable variable (measurement X in Figure 1 1) that describes the state of process, typically the control variable PV itself. If the physical effect causing the non linearity is known, the corresponding measurement variable should be used for the gain scheduling. Example: If the non linearity is caused by thermal radiation, which increases proportional to the 4th power of temperature, the measurement of the temperature should be used for the gain scheduling.

Between the operating points for which there are exact parameter values, the values are calculated by linear interpolation of the neighbouring interpolation points so that soft and bumpless transitions are possible between the operating points. The term "parameter scheduling" makes it clear that the "timetable" for adjusting the parameters is specified in advance. In contrast, an adaptive controller adapts itself automatically to the differing process responses during operation.

Figure 2-2 Source chart of the function block GainSched



The function block GainSched is produced by compiling the CFC chart "fbGainSched" as a block type.

The function block GainSched consists mainly of one polygon function block for each control parameter “Gain”, “Ti” and “Td” executing the linear interpolation. An additional logic ensures the horizontal extrapolation, i.e. the output of the controller parameters corresponds exactly to the input value X3 and X1 respectively if the input value is larger than X3 or smaller than X1 respectively.

This CFC chart is supplied together with the library so that the user has the option of expanding the existing basic functionality as necessary, for example to more than three operating points.

Note

The combination of several locally optimized controllers by gain scheduling to form a non-linear controller does not necessarily represent an optimal non-linear controller for the non-linear process when considered from a mathematical point of view. This becomes clear even with benign nonlin-earities (that are continuous and can be differentiated) when setpoint step changes are made between different operating points. With nonlinearities that are discontinuous or cannot be differentiated or with non-monotonic nonlinearities, great caution is needed.

2.3 Application Examples

- Control (especially temperature control) of batch processes, for example, batch reactors and batch columns
- pH value control
- Temperature control with phase transitions (for example, fluid/vapour)
- Control of semibatch plants (continuous plants with operating point changes, for example, polymerization reactors)
- Control in power plants with load changes

2.4 Gain Scheduling for Batch Processes

A typical area of application for gain scheduling is in batch processes that, in contrast to continuous processes, cannot be linearized around a fixed operating point because they need to be moved backwards and forwards between different operating points during the course of the batch. Here, there are three application scenarios:

- The controller parameters depend on a single continuously measurable variable that is representative of the operating point, for example, the reactor temperature. This is the normal use case for the GainSched block: The management of the controller parameters is handled in the block and is independent of batch recipes.
- The controller parameters depend on a continuously measurable variable that is representative of the operating point, but there is also a dependency on the materials used in the reaction. Suitable parameter sets for gain scheduling can then be anchored in the recipe and transferred by SIMATIC Batch to the GainSched block.
- The controller parameters depend only the current phase of the batch. They can then be written directly from the Batch package to the PID controller and no gain scheduling block is necessary. The disadvantage of this is that there is bump in the controller parameters at the transfer from one phase to the next. The controller should be put into manual mode temporarily at the time of the transfer to avoid a bump in the manipulated variable.
- The recipe only specifies which of the controller parameter sets 1 to 3 is currently required from the GainSched block. The numerical values of the parameter, however, are not anchored in the recipe. In this case input variable X of the GainSched block can then be used as the number of the required data record and assigned by the recipe instead of being linked with a measurable process variable. In this case, there are only three discrete values for X and the precautions against a change of controller parameters with bump outlined above must be taken because the interpolation abilities of the GainSched block are not used.

Implementation of Gain Scheduling 3

3.1 Installation

The installation of the PCS 7 Advanced Process Library is performed automatically by the PCS 7 master setup V7.1.

Note: There is a gain scheduling function block and process tag type already available in the PCS 7 APC Library V7.0 SP1. Using an even older version, the signal flowchart can be build manually out of elementary CFC function blocks.

3.2 Configuration: Creating an Instance of the Process Tag Type

The following steps are carried out for the gain scheduling in the same way as for any other process tag types.

Figure 3-3 Please open the Advanced Process Library in the SIMATIC Manager: Open Project/Libraries/PCS_7_AP_Library V71.

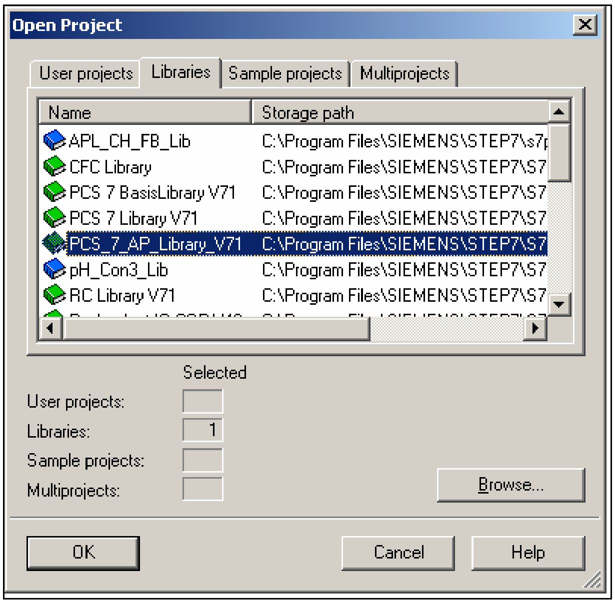
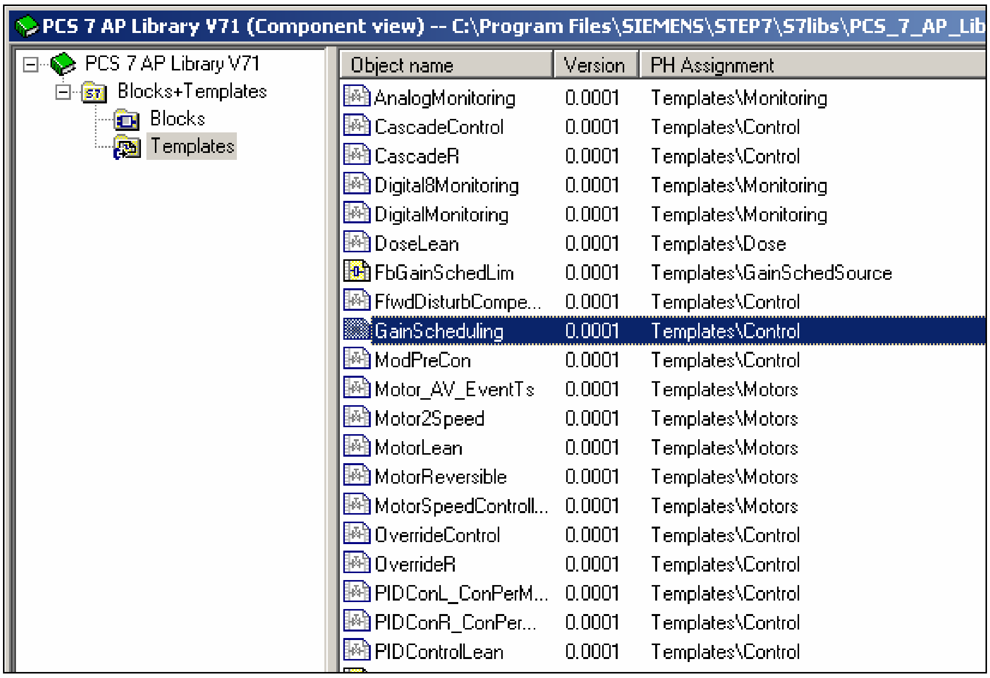


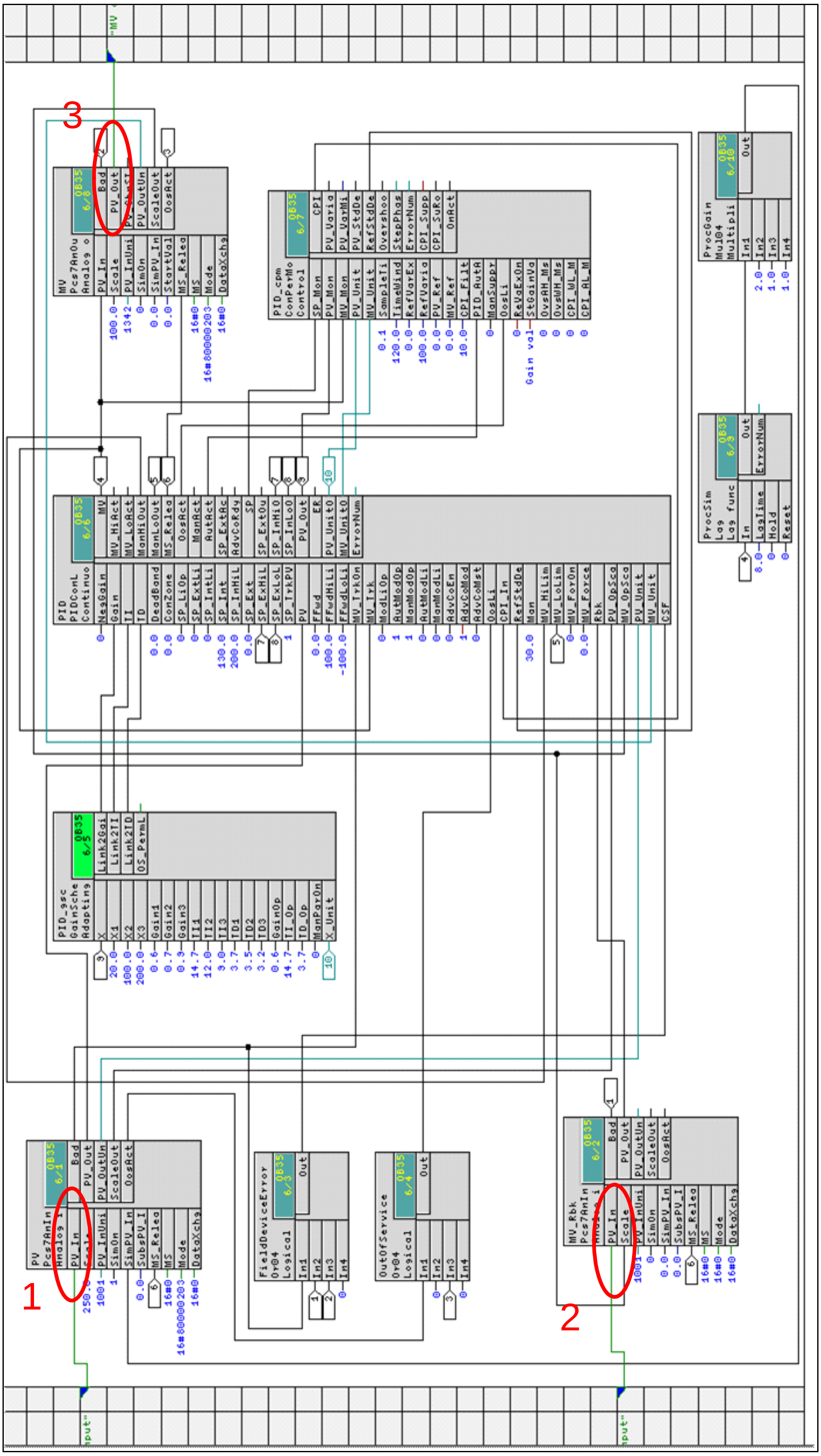
Figure 3-4 Chose the process tag type “GainScheduling “



Copy the process tag type “GainScheduling“ from the subfolder “Templates” into the master data library of your PCS 7 multiproject and modify it if this is necessary according to your general application requirements.

Copy the process tag type from the master data library to the designated part of your multiproject <project name>_Prj, in the appropriate target folder (Process cell/unit etc.) of the plant view. You obtain an instance of the process tag type i.e. a CFC chart, which indicates its origin by its symbolic representation. (Numerous instances of a process tag type can be generated automatically with the import export assistant using parameters from an import file.)

Figure 3-5 Important connections of the process tag type



Rename the new CFC chart and check if the cyclic interrupt OB is correct (in the CFC chart “Edit”/”Open run sequence”).

Open the CFC chart and implement the following connections:

- Control variables: Connect the analog input driver Pcs7AnIn for the control variable “PV“ (see Figure 3-5, number 1) with the symbolic name of the corresponding peripheral signal from the hardware configuration. The unit of the signal can be adjusted at the input “PV_InUni” and the range at the input “Scale”.
- Manipulated variable: The manipulated variable “MV“ has to be connected to the periphery via the output “PV_Out” of the analog output driver Pcs7AnOu (see Figure 3-5, number 3). The unit of the signal can be adjusted at the input “PV_InUni” of the analog output driver. The input “PV_In” of the analog input driver Pcs7AnIn named “MV_Rbk” must be connected to the actual achieved position feedback of the periphery if such information is existing (see Figure 3-5, number 2). If no analog position feedback is available, delete the function block “MV_Rbk” and its connections.

Compile and download your changes to the AS. Compile the OS again to include the new PID faceplate in your runtime application.

Now you have successfully integrated the controller with gain scheduling to your process. In the next step the algorithm must be configured adequately. Therefore, the PID controller must be adjusted to the three different operating points of your process, using the PCS 7 PID tuner. The usage of the PID tuner is explained in the next chapter 3 considering a linear process as application example. Chapter 4 describes the peculiarities of gain scheduling.

4 Controller Optimization with PID Tuner for Linear Process Behaviour

The plant model and the corresponding controller parameters are determined by the PID tuner after an active process excitation. For modelling, the dead times are approximated by higher model orders.

The experiments documented in the following are conducted with the control loop “TIC402” included in the “APL_Example_EU” of the PCS 7 Advanced Process Library, i.e. using the control loop without gain scheduling. In order to reduce the time consumption for the experiments, the process simulation is running faster than typical process plants. Although measurement data cannot be recorded faster than with a cycle of 200ms, it is still possible to optimize controllers with faster sampling times (e.g. 100ms) via the PID tuner. However, a precise identification of process time constants smaller than 5s is not possible.

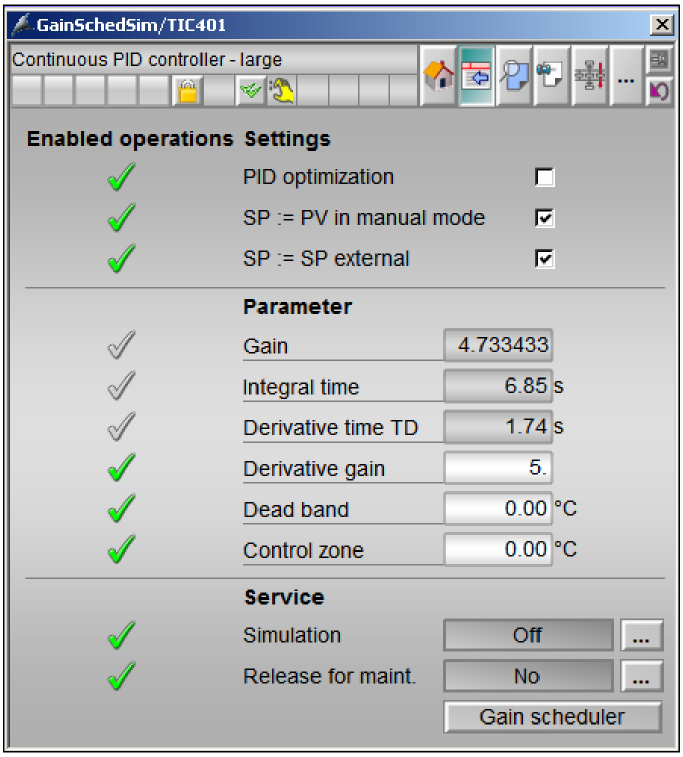
4.1 Starting the PID Tuner

The PID tuner is always executed in the engineering station (ES).

Because the operator and the engineering station are usually not executed on the same computer or even in the same room, a coordination of the different users is offered by PCS 7. Therefore, a controller optimization can only be started in the ES, if the operator in the control room enables this function for the corresponding controller on the OS.

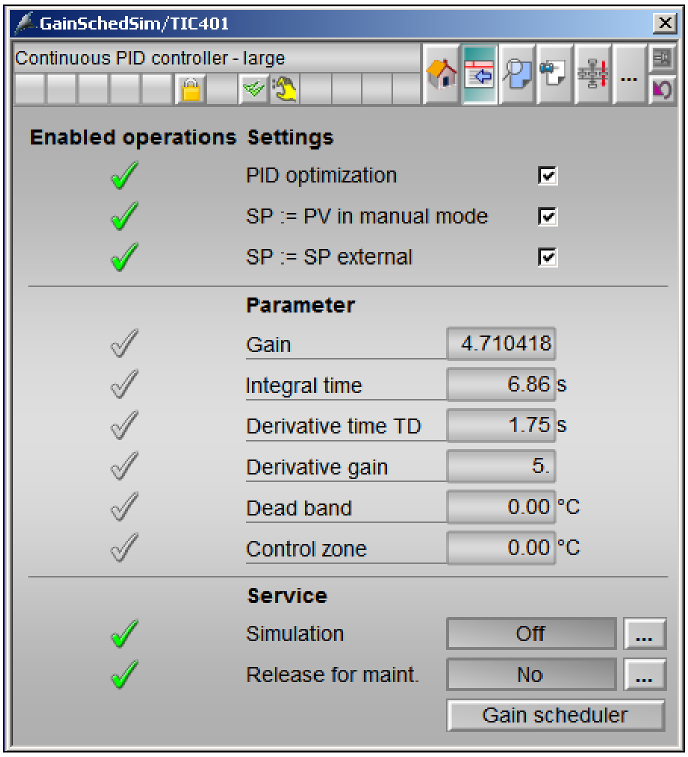
The engineering system (the PID tuner) can be enabled to actively interfere in the process via the PID faceplate in the view “parameter”. During PID tuning, operation of the PID controller on the OS is partially blocked.

Figure 4-6 PID-faceplate view parameter: enable optimization



The PID tuner is enabled by setting “PID optimization“.

Figure 4-7 PID faceplate: optimization is enabled



The enabling is reset automatically after finishing the PID tuner.

Now, you can start the PID tuner in the CFC chart of your controller. Left click on the instance of the controller to be optimized in order to mark it (it is highlighted in blue). By choosing “Optimize PID Controller” via the CFC menu “Edit” the PID tuner is started and shows up with its first input dialog.

Figure 4-8 Starting the PID tuner in the CFC chart of the process tag

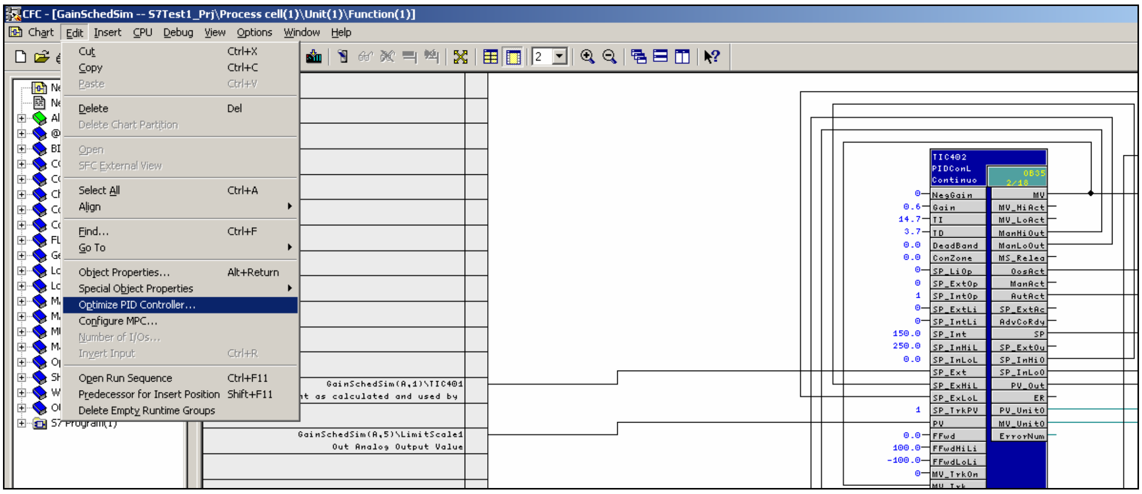
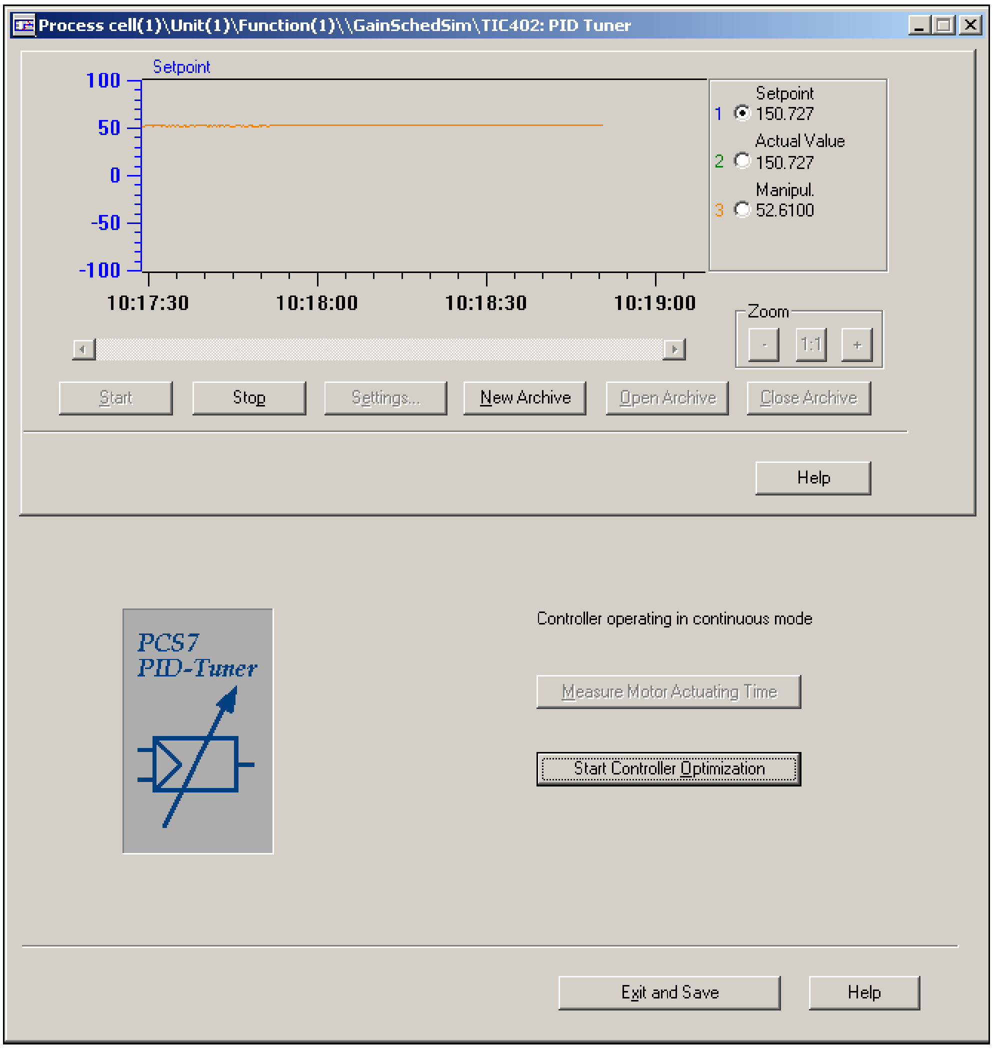


Figure 4-9 First dialog of the PID tuner

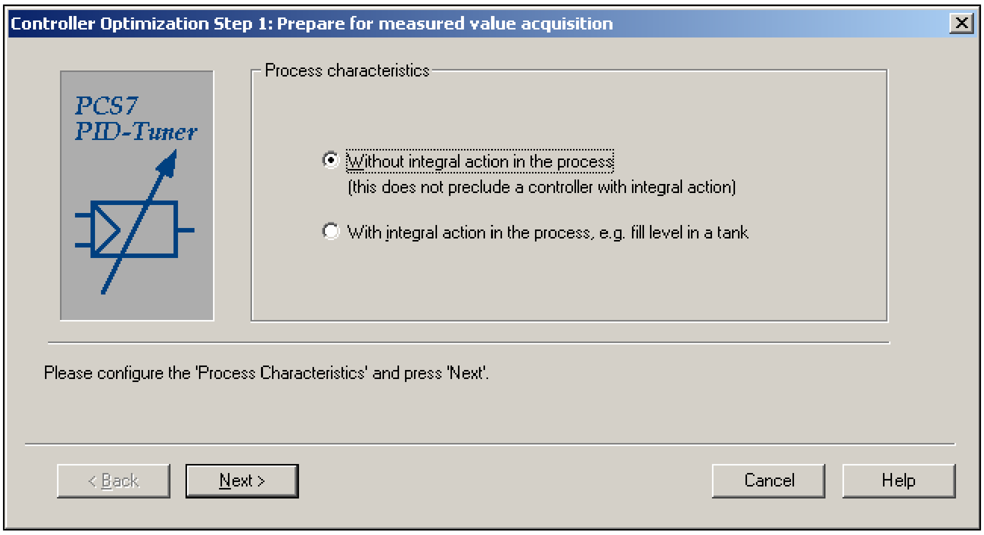


Via the button “Start Controller Optimization“ you will be guided stepwise through the input dialog, the process excitation and the controller design.

If you want to modify settings of the trend control beforehand, you have to stop the trend recording via the button “Stop”, and start it again after editing the settings.

4.2 Step 1: Prepare for Data Acquisition

Figure 4-10 Choice of the process characteristics



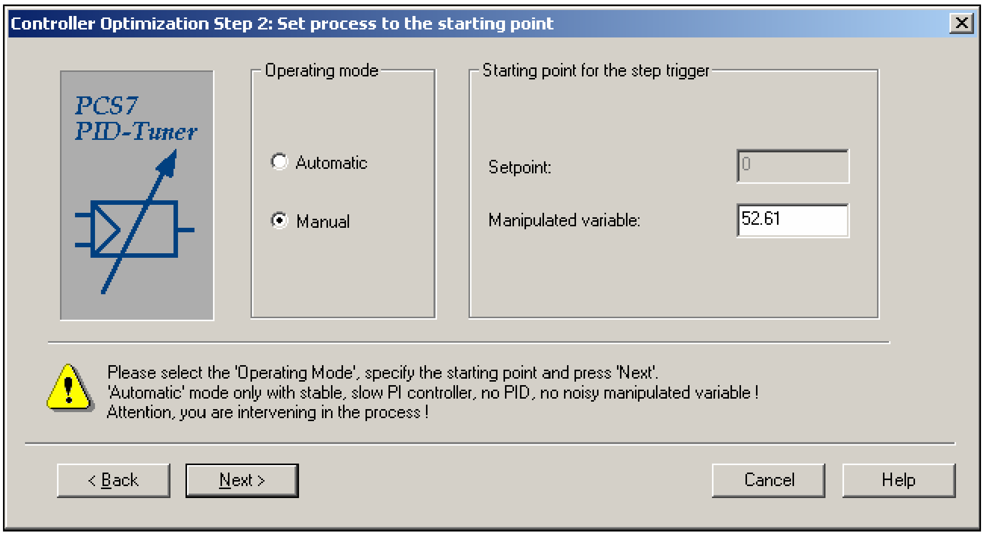
As a prerequisite for a successful optimization the kind of process characteristics must be parameterized in the PID tuner. Two process types are distinguished:

- Processes without integral action:
  Processes reaching a steady state process value and keeping this value constant in the following as long as no external disturbances are interfering, while being affected by a constant manipulated variable (MV not zero) for a certain time, show non integral action (e.g. temperature and flow processes).
- Processes with integral action:
  Processes not reaching a steady state process value while being affected by a constant manipulated variable unequal zero show integral action. The control variable increases (decreases) as long as the manipulated variable is different from zero, otherwise it is constant. Examples of such processes are: level control of tanks, drives that should be driven in a defined position.
- Note: Integral actions with a stationary point at a distinct manipulated variable different from zero also exist. Example: level control of a tank with constant feed, manipulated by the drain valve. At a distinct valve position of the drain (distinct value of the manipulated variable), the feed is exactly equal to the drain and therefore the level is constant. Moving a small difference away from this distinct value of the manipulated variable in manual mode, the tank will be running completely empty or completely full. Hence processes with integral action show unstable process behaviour.

After configuring the process characteristics the next step can be reached by the button “next“.

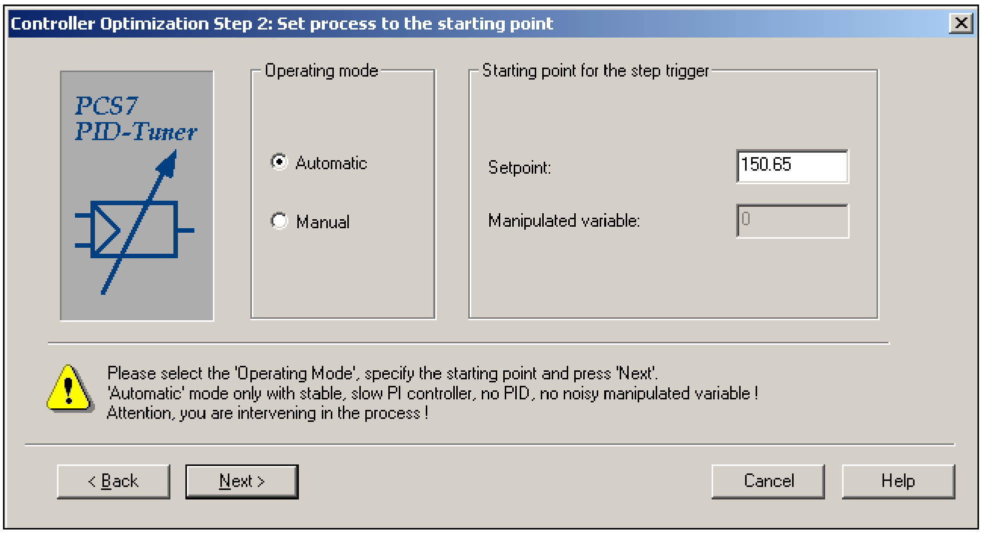
4.3 Step 2: Move Process to Starting Point

Figure 4-11 Define the starting point of the process excitation (in manual mode)



First you have to decide whether the process should be excited in closed loop (automatic) or in open loop (manual mode). If the actual controller setting is not stable or not well damped, you should prefer the manual mode for the controller optimization. Therefore, switch to manual mode in the faceplate of the controller in the OS. Transfer the actual value of the manipulated variable visualized on the right of the trend display of the PID tuner (in red) to the PID tuner control panel “Manipulated variable”. If you prefer another starting point, enter your desired value in the control panel of the PID tuner and wait until your process is in steady state again. Afterwards you can continue with the optimization.

Figure 4-12 Define the starting point of the process excitation (automatic mode)



The controller can be optimized in automatic mode, if the behaviour in closed loop is of good nature. Dealing with a process with integral action, the optimization must be performed in automatic mode, because in manual mode it can not operate stable. A P controller is good enough to stabilize such processes.

Transfer the actual setpoint visualized on the right of the trend display of the PID tuner (in blue) to the control panel “Setpoint” of the PID tuner. If you prefer another starting point, enter your value in the control panel of the PID tuner and wait until your process is in steady state again. Afterwards you can continue with the optimization.

4.4 Step 3: Wait for Steady State and Start Excitation

Figure 4-13 Define the step excitation (manual mode)

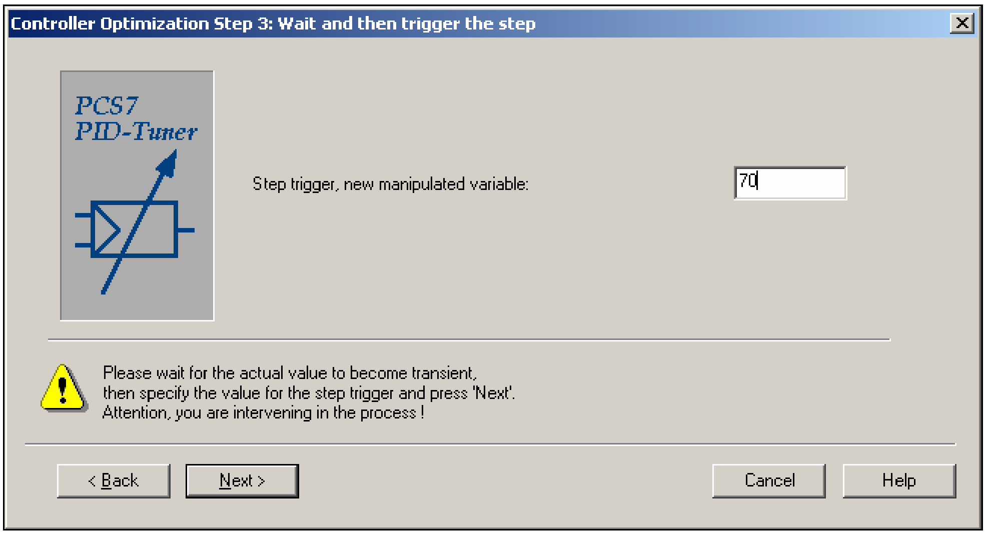
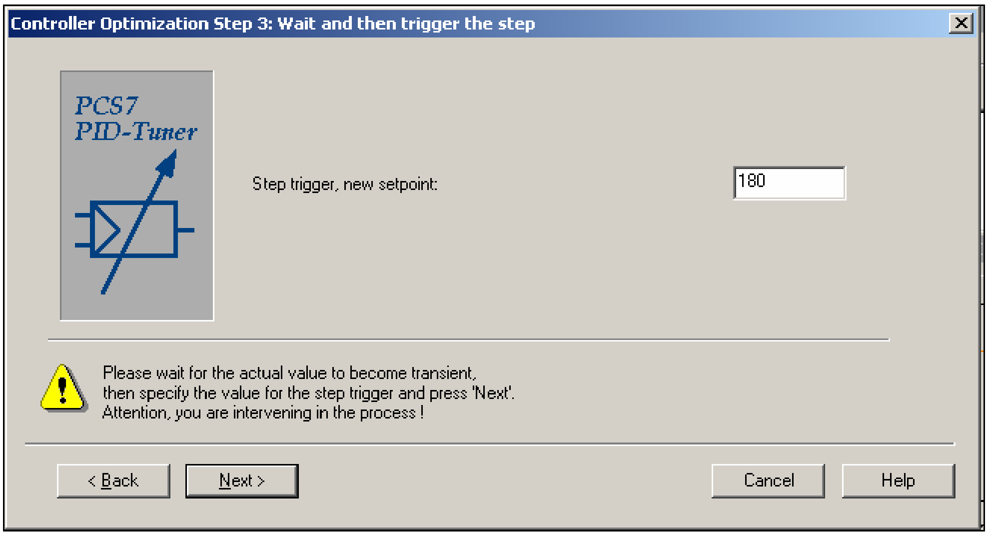


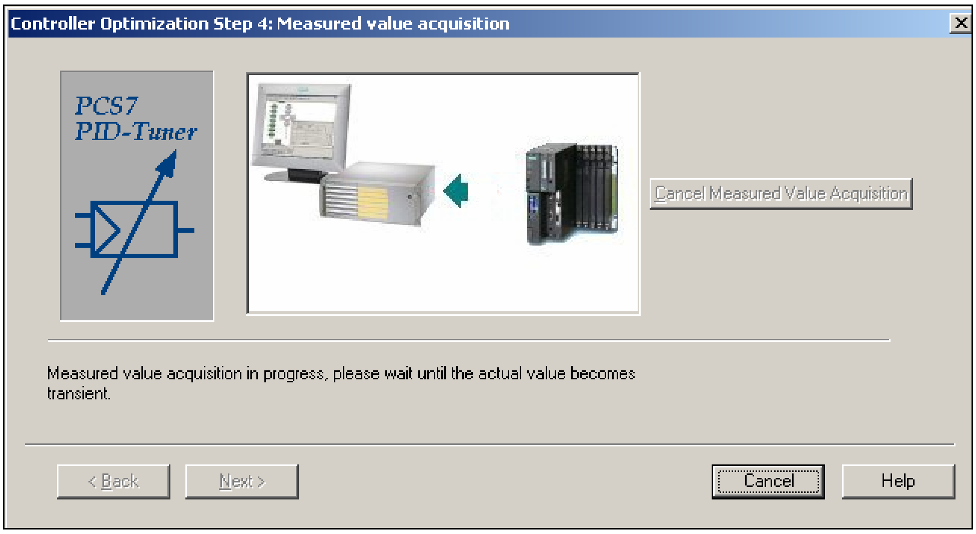
Figure 4-14 Define the step excitation (automatic mode)



The step of the setpoint or of the manipulated variable should be large enough to detect the process behaviour clearly, i.e. the step response should be clearly distinguishable from the measurement noise. On the other hand the step must not be too large to disturb the production no more than necessary and to avoid critical plant states. In case of doubt an agreement with the person responsible for the process or with a highly experienced operator is advisable. Having entered the new value for the setpoint in automatic mode or for the manipulated variable in manual mode, this value is transferred immediately to the process by clicking “Next”.

4.5 Step 4: Data Acquisition

Figure 4-15 Data acquisition



The PID tuner acquires the measurement data and checks for the end of transient behaviour in the actual values. However, you also have the possibility to cancel the acquisition manually. Observe the trend display until steady state is reached, i.e. the actual value and the setpoint are nearly constant. The program recognizes the transient behaviour and stops the data acquisition automatically if not prevented by significant disturbances.

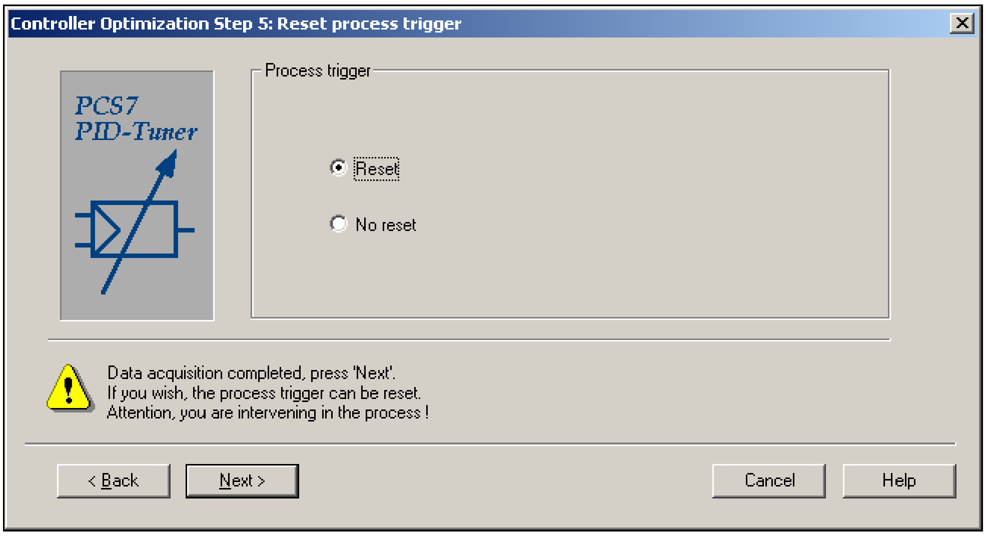
Press the button “Cancel Measured Value Acquisition“ only in the following cases:

- You assume that despite of disturbances such as measurement noise, manipulated variable and actual value in the trend display show steady state behaviour. Please be patient! The automatic steady state detection waits longer than a human observer typically would do according to the trend display. However, cancelling the measurement acquisition too early affects the quality of modelling and the corresponding controller design.
- The curves of the manipulated and the actual variable in the trend display show no convergence to steady state at all. If you decide to proceed anyway, plant and controller parameters should be examined with scepticism and tested carefully. In such a case (in manual mode) check if the process is affected by integral action or by a sluggish disturbance. If a steady state is not reached in automatic mode, check the current controller parameterization.

Note The button “Cancel Measured Value Acquisition“ is not enabled until a certain quorum of measurement cycles is recorded. The plant model and the controller parameters are calculated with the data acquired until then by the PID tuner.

4.6 Step 5: Reset Process Excitation

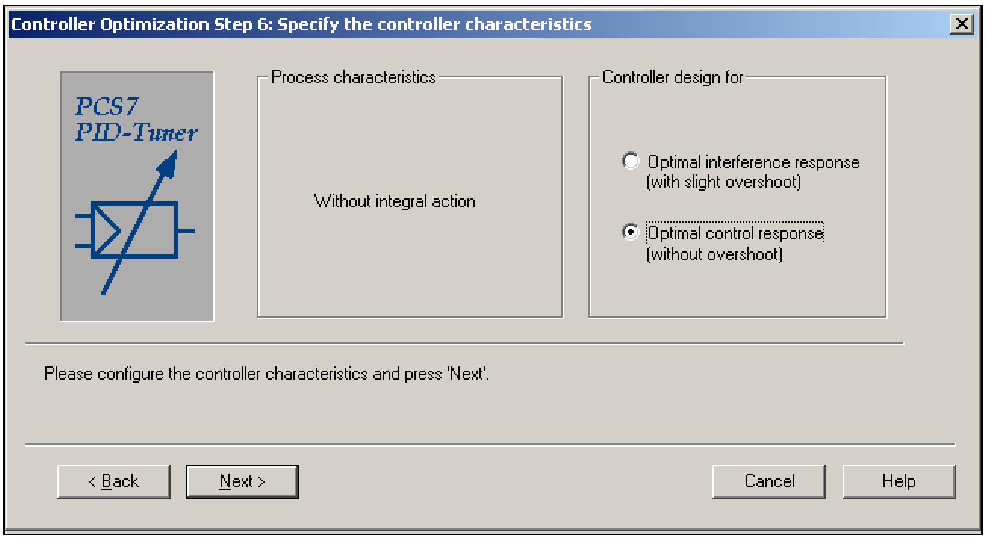
Figure 4-16 Reset process excitation



You have the possibility to leave the process in its new state or to return to the starting point before the optimization.

4.7 Step 6: Specify Controller Characteristics

Figure 4-17 Specify controller characteristics



In this step the controller characteristics can be specified in two kinds. In the setpoint tracking mode ("optimal control response mode") the controller parameters are selected such that the actual value settles aperiodically after a setpoint step in the closed control loop. However, if the process settling time is too close to the controller sampling time, a slight overshoot of the actual value caused by rough sampling can occur in closed loop despite the aperiodic setting.

In the optimal disturbance ("interference") response mode, disturbances are compensated fastly by the controller. The controller itself is adjusted more aggressively. Thus an overshoot up to 10% of the step size can occur after a setpoint step.

In the following the results of the identification and the calculated controller parameters for the same plant are presented twice for comparison. The process is excited firstly in manual mode with a step in the manipulated variable, and secondly in automatic mode with a step in the setpoint.

Figure 4-18 Results of the identification (excitation in manual mode)

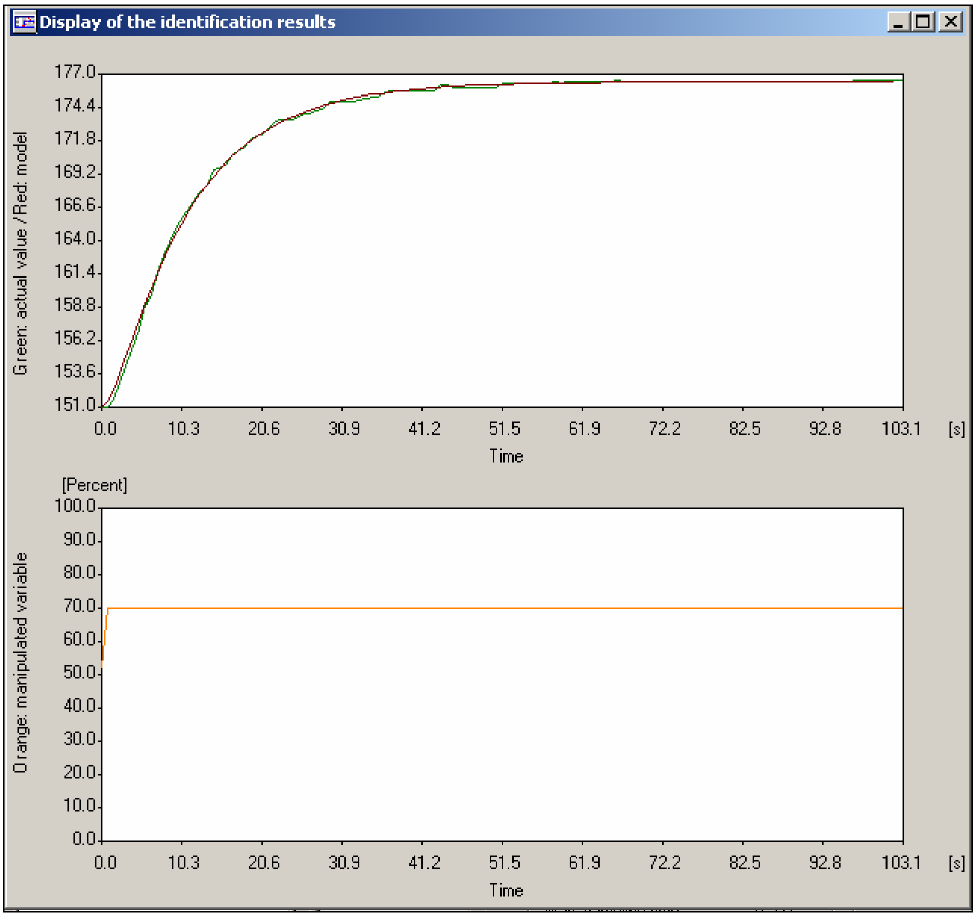
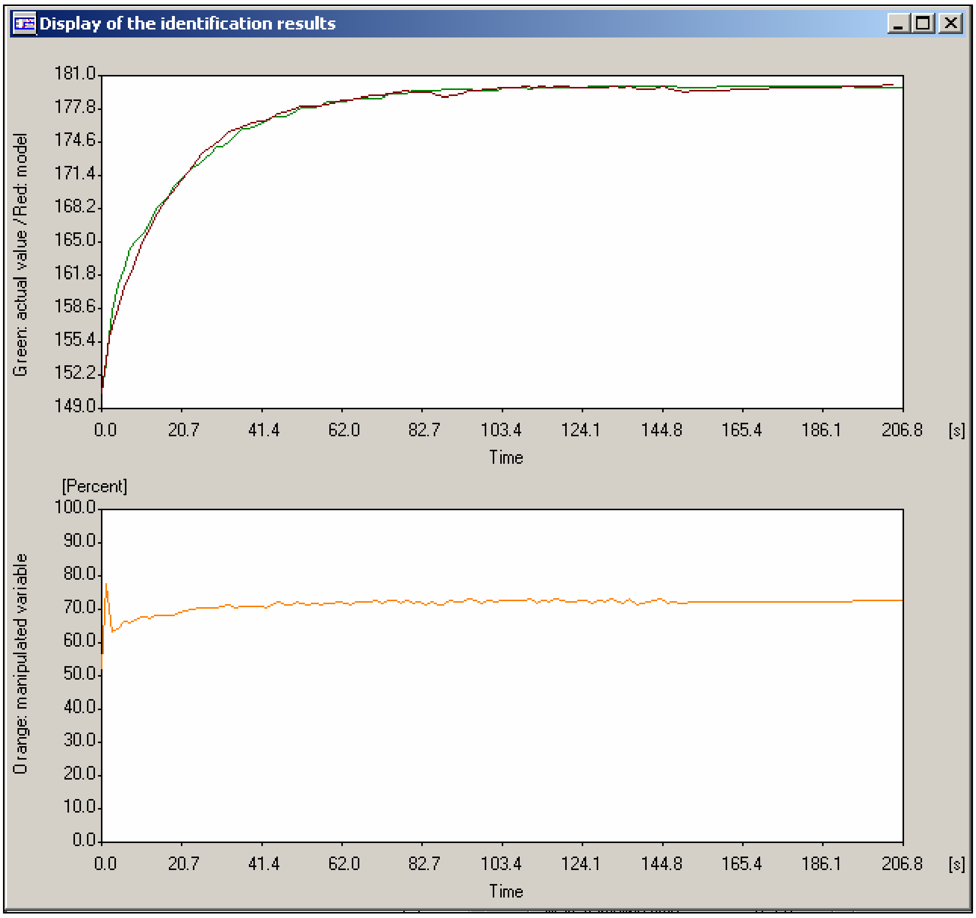


Figure 4-19 Results of identification (excitation in automatic mode)



4.8 Step 7: Select Controller Type

In this step the results of the identification are presented numerically. Due to the non linear process simulation used here, and the different ranges of amplitudes used for the experiments, the deviation between the results in manual and automatic mode in this case is higher than typically for linear processes.

Figure 4-20 Selection of the controller type (results of excitation in manual mode)

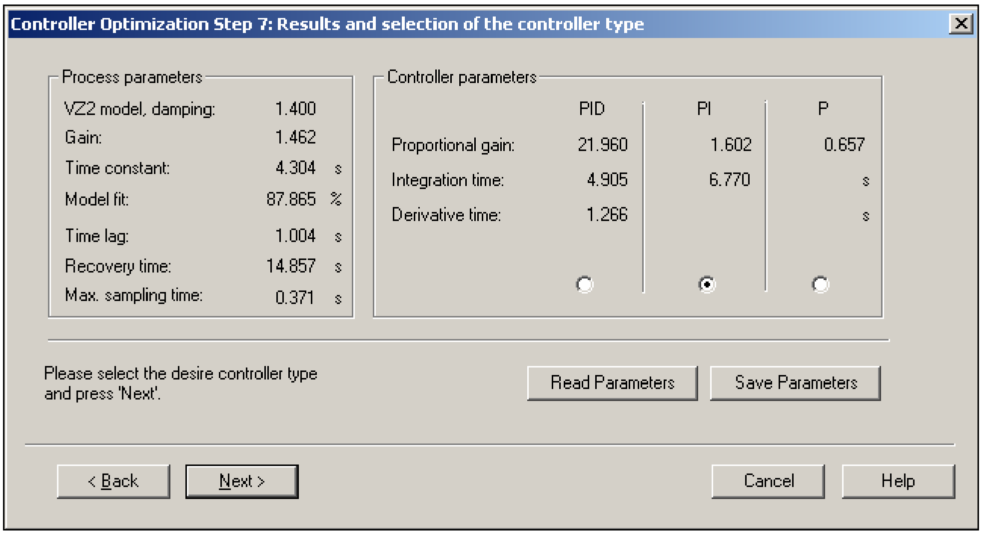
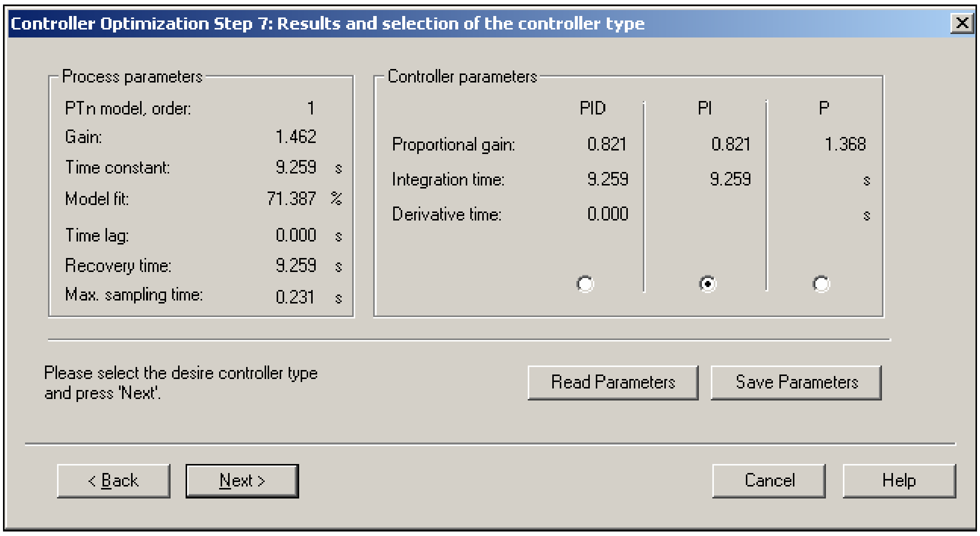


Figure 4-21 Selection of the controller type (results of excitation in automatic mode). The small time lag, original of a value of 2s, is not detected at all by the identification in automatic mode.



The P controller is the simplest possible controller type. It can be used for processes with integral action and as a slave controller in cascade control (e.g. PI-P cascade). In general its disadvantage is the remaining control deviation in steady state.

The PI controller is able to adjust the actual value exactly to its setpoint in closed loop. Therefore, it is the controller type used most frequently in practice and is offered as a default type.

Using the PID controller a higher performance than with a PI controller can be achieved for most processes. Due to a higher actuator activity, faster rising times after setpoint steps and smaller control deviations after disturbances can be achieved. However, a higher actuator activity implies more energy consumption and more abrasion or wear. Hence the usage of the D part is to be avoided, if the process values are very noisy or if the actuators (e.g. valves) are especially sensitive. A PID controller is typically used with binary actuators, which are controlled by pulsewidth modulation (e.g. electrical heating) and are not impacted by signal variance of the manipulated variable.

A PID controller is not offered for processes with first order as their optimal performance can already be reached with a PI controller. If necessary, the control gain can be set larger than recommended by the controller design for 1st order processes. (In the underlying example the process was identified as a first order system by mistake after excitation in automatic mode, due to numerical inaccuracies according to the fast plant simulation.)

A validation of the behaviour of the selected controller can be observed in the offered closed loop simulation in the next step. The choice of controller type can be revised if necessary. Please note that the influence of noise is not considered in the simulation.

4.9 Step 8: Simulate Closed Loop

Figure 4-22 Simulation with optimized parameters (results of excitation in manual mode)

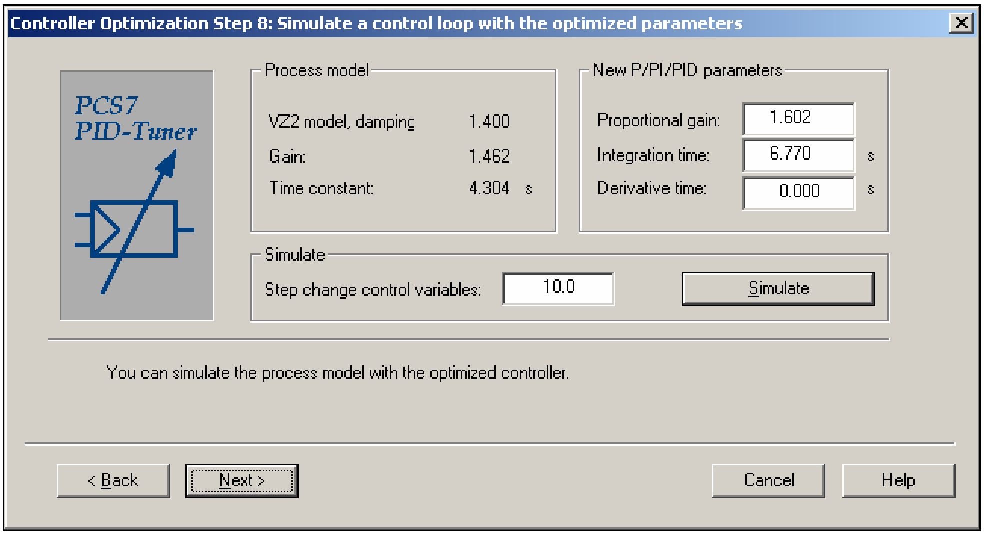


Figure 4-23 Simulation with optimized parameters (results with excitation in automatic mode)

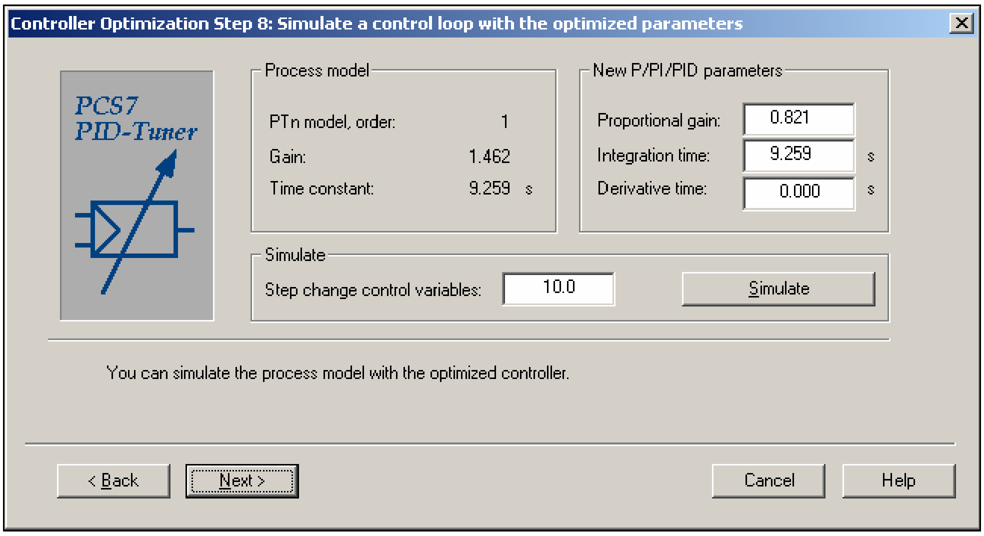


Figure 4-24 Closed loop simulation (results of excitation in manual mode

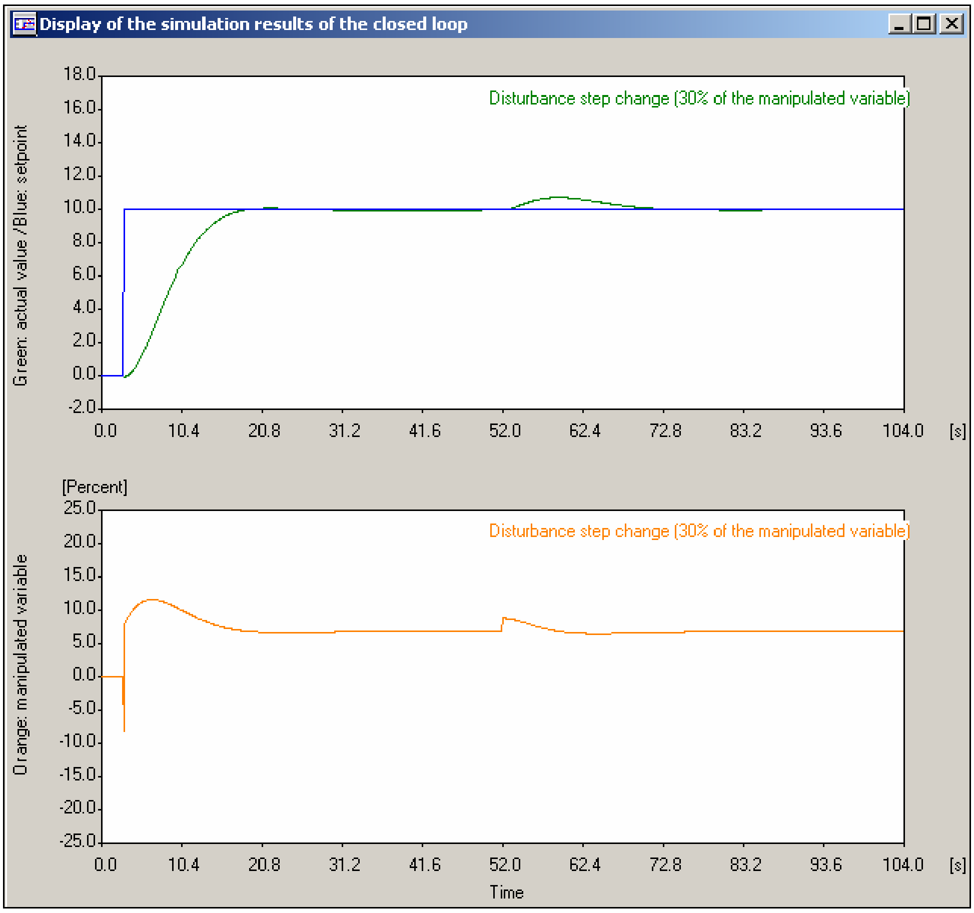
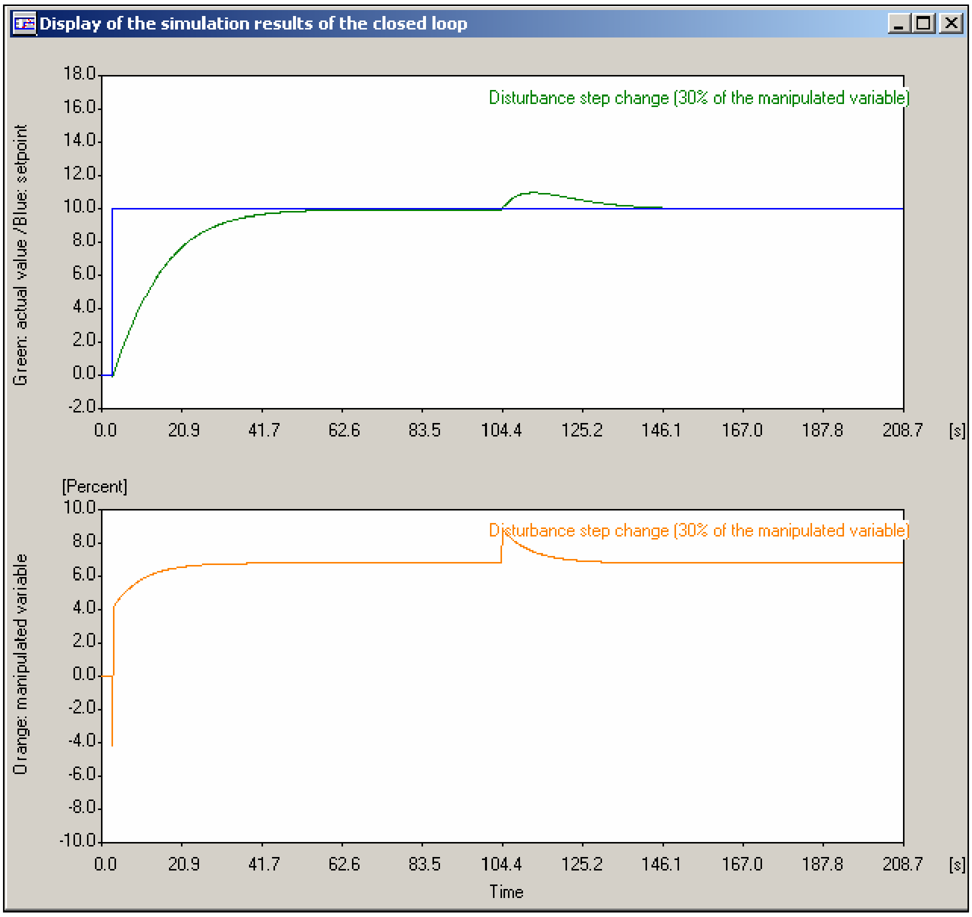


Figure 4-25 Closed loop simulation (results of excitation in automatic mode)

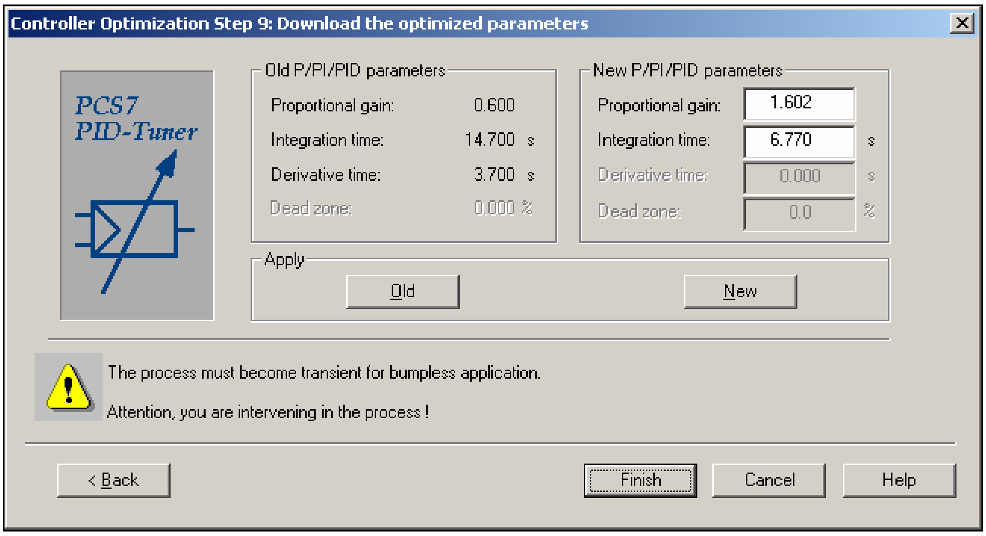


The identified process models and the corresponding controller parameters are not exactly the same for excitation in manual and automatic mode. In manual mode the step in the manipulated variable immediately affects the process. The captured data are directly the step response of the process. In automatic mode a step in the setpoint of the controller affects indirectly the process. The controller tries to achieve the setpoint by generating a non rectangular characteristic of the manipulated variable.

Due to the non linear behaviour and the fast process simulation, the deviation between identification results in manual and in automatic mode is larger in this example than typically in practice. However, better identification results can mostly be achieved by an excitation in manual mode because the manipulated variable in manual mode is free of noise and not correlated to the controlled variable.

4.10 Step 9: Download Optimized Parameters

Figure 4-26 Download the optimized parameters



The old and new controller parameters are displayed in the respective fields. The old controller parameters correspond to the online data from the CPU. The time lag of the differential action within a PID controller, which does not belong to the Advanced Process Library (APL), is always set to one fifth of the derivative action time and is not displayed. Using an APL controller the lag time of the D component is calculated from the parameter “DiffGain” at the function block.

Click on the button “Old“ or “New“ to apply the desired parameters.

The parameters are then transferred to the CPU connected online and to the offline data management of the engineering system.

Click on the button “Finish“ if you have completed optimization or click on the button “Back“ to repeat the previous steps.

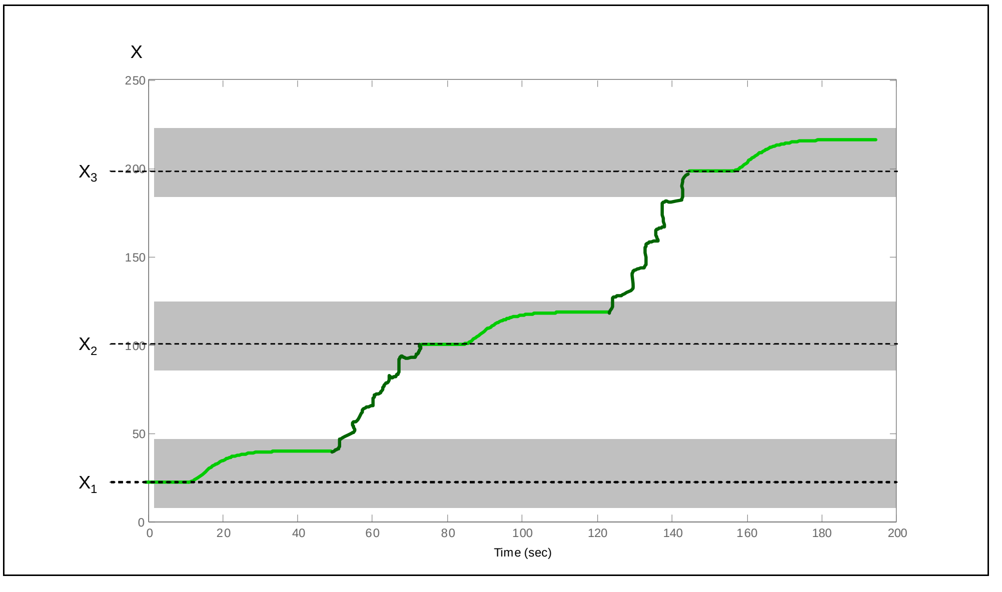
Controller Optimization with PID Tuner for Gain Scheduling 5

5.1 Definition of Operating Points

The steps 1 to 8 of the PID tuning (see chapter 3) must be repeated for each operating point X1, X2 and X3 for the parameterization of the Gain-Sched function block. Please use amplitudes as small as possible for the excitation of the process to capture the approximately linear behaviour in the neighbourhood of the examined operating point. In order to cover the whole accessible range with the schedule, the working point X1 is typically located at the lower limit of the operating range and X3 nearby the upperlimit. Please be sure that the process does not crash into the limits during the whole excitation.

X2 can be located in the middle between X1 and X3 or exactly at an operating point typical for the plant operation.

Figure 5-27 The three step responses (light green) characterize the behaviour of the non linear process in the neighbourhood (highlighted in grey) of the three operating points X1, X2 and X3.



5.2 Handling of the Combination of Controller and Gain Scheduler

The online download of the parameters from the PID Tuner to the controller function block executed in step 9 is not possible using the GainSched function block as the inputs “Gain”, “TI” and “TD” of the controller are connected to the outputs “Link2Gain”, “Link2TI” and “Link2TD” of the GainSched function block. (Hence the controller parameters are not operable in the face-plate of the controller but in the parameter view of the GainSched face-plate.)

Therefore, you have to transfer the calculated controller parameters from the PID Tuner to the input parameters (Gain 1…3, TI 1…3 and TD 1…3) of the corresponding operating points X1...X3 of the GainSched function block manually in step 9 of the PID tuner.

You can step into the faceplate of the connected GainSched function block via the button "Gain scheduler" in the parameter view of the controller face-plate.

Figure 5-28 Parameter view of the controller faceplate

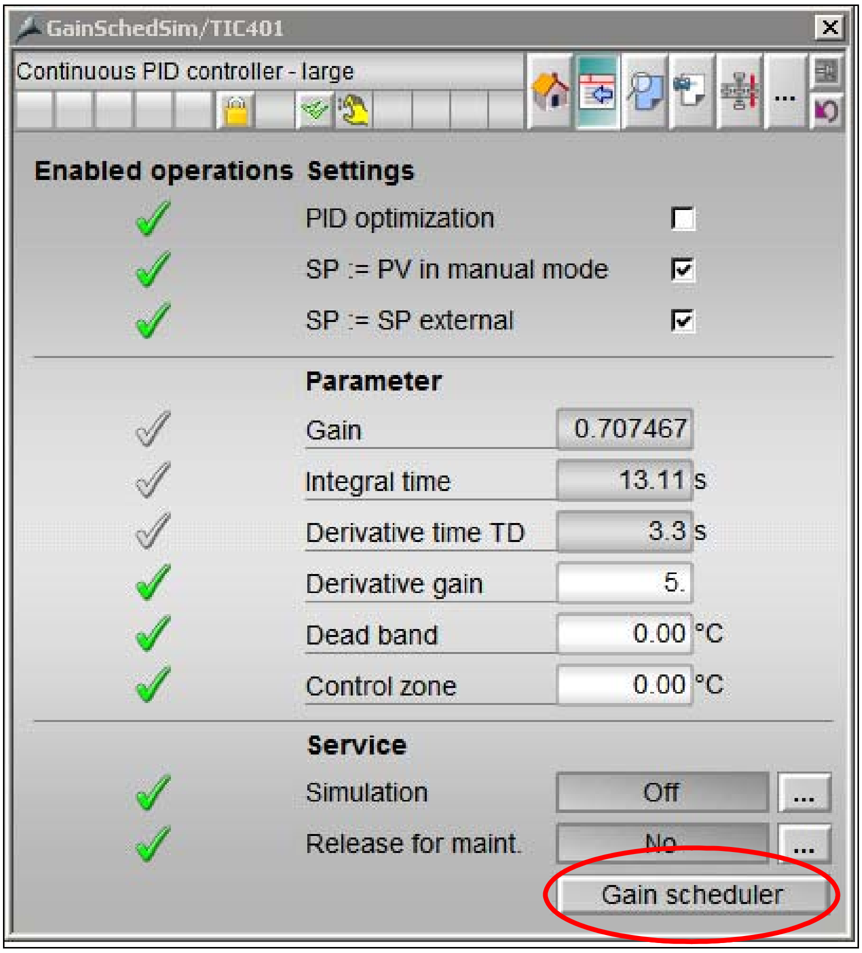
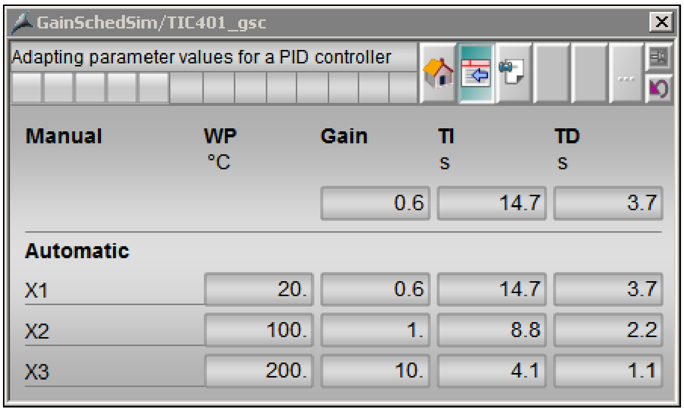


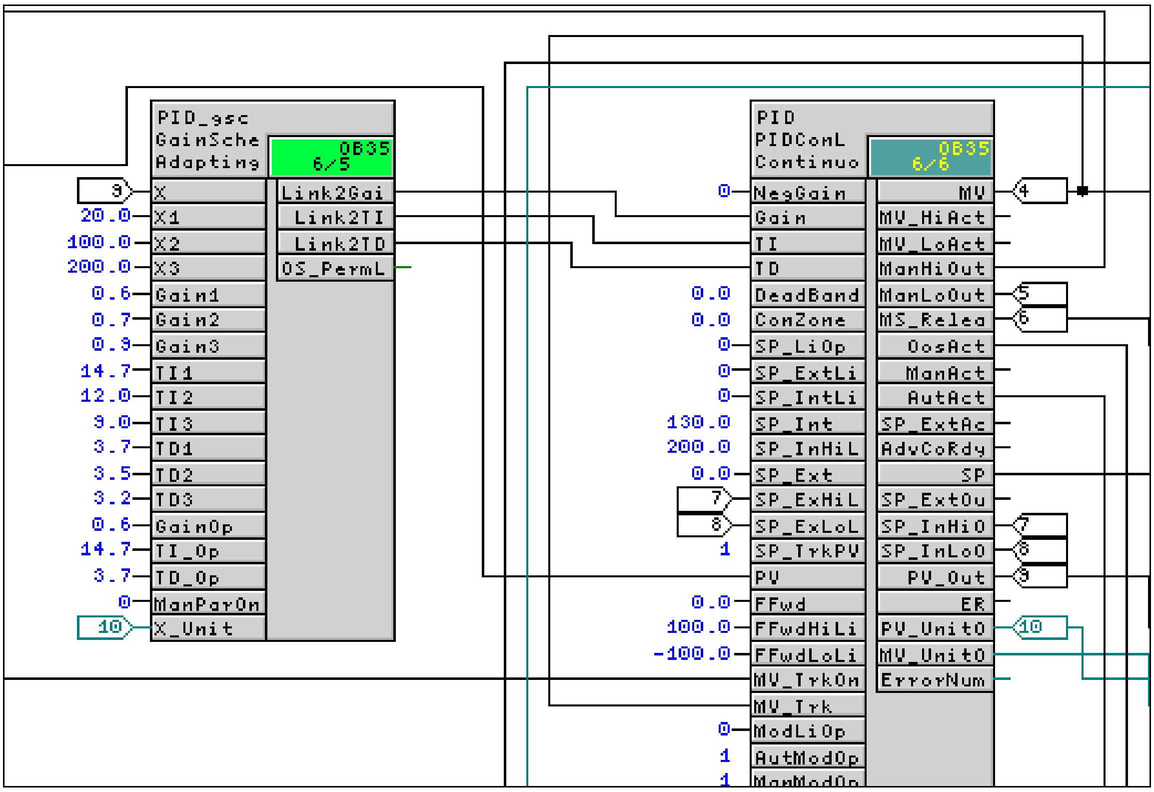
Figure 5-29 Parameter view of the GainSched faceplate



The operating points and the corresponding controller parameters can be inserted into the table in the lower part of the parameter view of the Gain-Sched function block.

Please be sure to save the values in the offline data management of the engineering system permanently by either reading back the parameters of the AS to the ES or by directly entering the values in the inputs of the CFC function block manually.

Figure 5-30 Entering the scheduling parameters into the CFC view of the GainSched function block manually



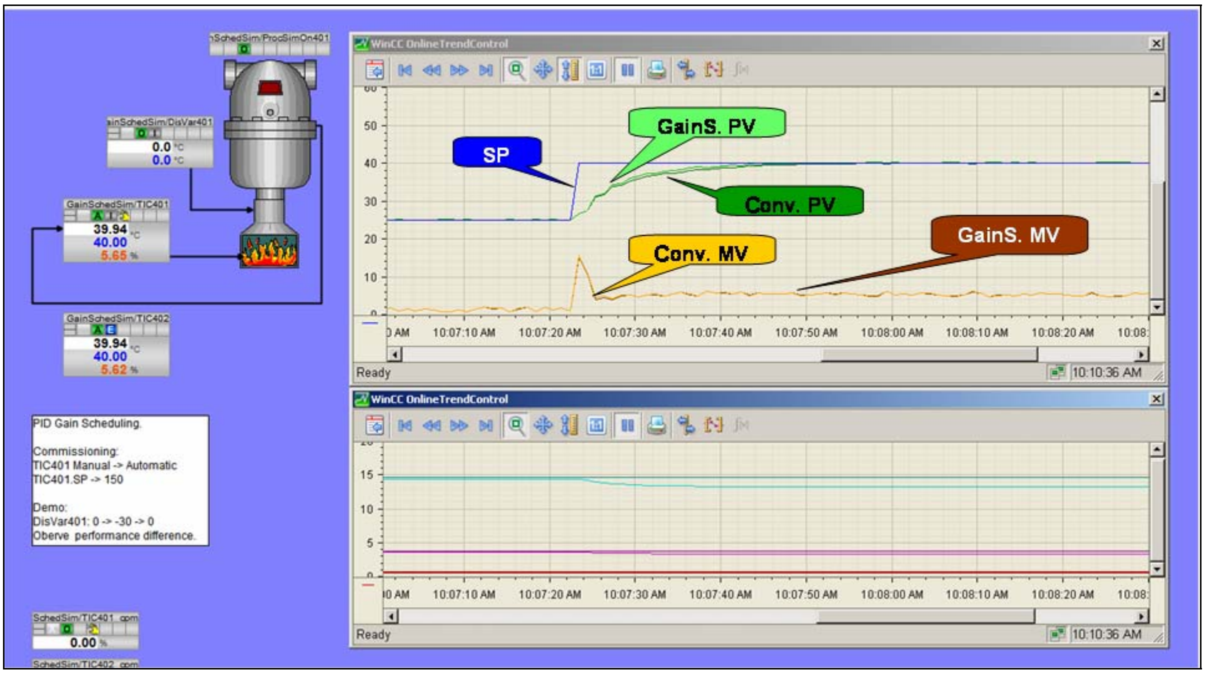
If you want to modify the controller parameters during special situations diverging from the schedule, switch the GainSched function block to manual mode (in the standard view of the faceplate) and insert your parameter values in the upper part of the parameter view.

6 Simulation Example

The process simulation is built up twice in the simulation example Gain-SchedSim of the PCS 7 project APL_Example_EU – one instance with PID controller and gain scheduling and the other one only with a conventional PID controller, while all other process parameters are identical. The signal flow chart in the OS picture shows the control loop with PID controller TIC401 and gain scheduling. The symbol of the conventional controller TIC402 is located below the one of TIC401 (without the illustration of the corresponding signal flow chart). The advantages of the PID controller with gain scheduling can be proved in a direct comparison (benchmark simulation, “parallel slalom”).

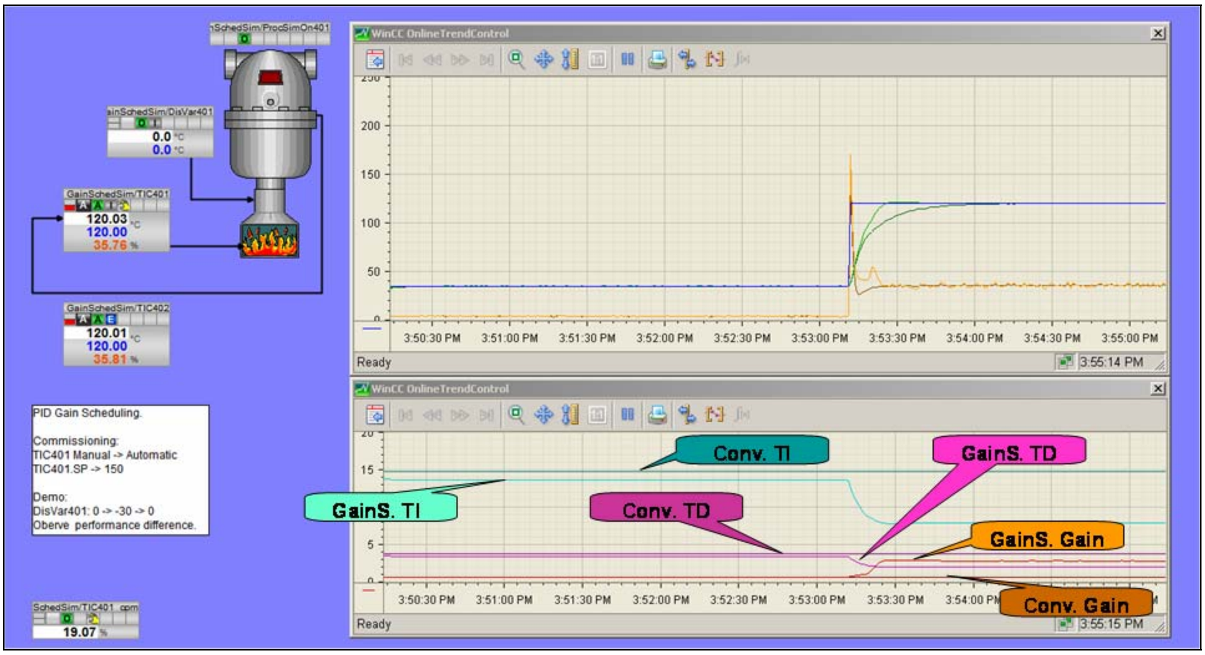
The parameters of the conventional controller are optimized only in the lower temperature range, while the ones of the controller with gain scheduling are optimized for each operating point (20°C, 100°C, 200°C) separately.

Figure 6-31 Step responses in lower temperature range



As you can see, the step response of the control variable is nearly identical for both controllers in the lower temperature range as both controllers are optimized according to this range.

Figure 6-32 Step responses in the upper temperature range

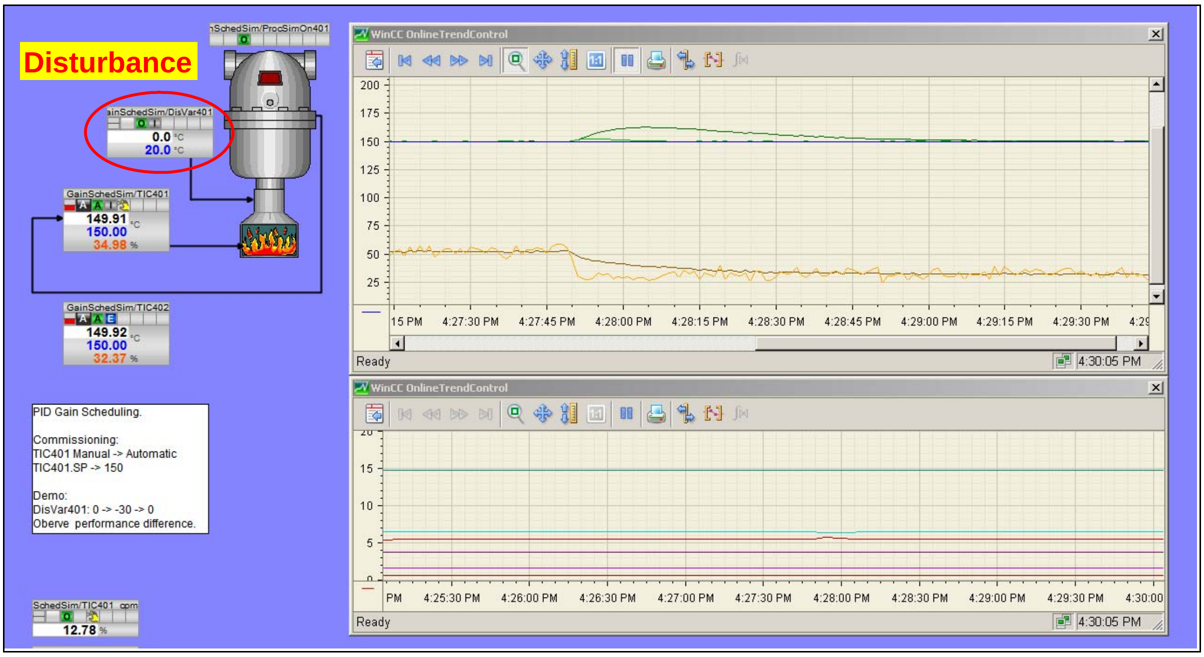


The performance improvement in the upper temperature range for optimal setpoint response (see Figure 6-32) as well as for optimal disturbance compensation (see Figure 6-33) can be clearly seen. Therein, blue represents the setpoint SP for both controllers, light green and orange the control variable PV and the manipulated variable MV of the controller with gain scheduling and dark green and dark orange the control variable and the manipulated variable of the conventional controller.

Due to the adjustment of the controller with gain scheduling to the whole temperature range, significant advantages especially in the middle and upper temperature range can be observed in contrast to the conventional PID controller.

The lower plot shows the constant parameters of the conventional controller in comparison to the adjusted ones of the controller with gain scheduling, that vary for the different operating points. Due to the interpolation, the control parameters are changing continuously and smoothly (not stepwise) in between two operating points.

Figure 6-33 Compensation of a stepwise disturbance in the upper temperature range. Using the controller with gain scheduling (light green) the process does not drift away from its setpoint.



7 Conclusion

Gain scheduling is valuable to adjust the controller parameters to different operating points, if a non linear process shows a different behaviour for each of its operating points.

The schedule for the adaptation of the parameters is determined by separate experiments at each of the different operating points and is stored in the GainSched function block. During a change between operating points the controller automatically retrieves the correct parameters in the schedule. A new identification of the process behaviour according to new measurement data is not necessary, in contrast to a fully adaptive controller. However, gain scheduling can be applied only if the nonlinearity of the process can be traced back to only one measurable process variable in a reproducible way.

The gain scheduling needs only small capacity of the SIMATIC CPU; a Gain-Sched function block requires less than a third of the memory capacity of a PID function block and accordingly less calculating time.

8 Internet Links

Table 8-1 Internet links

	Topic	Link
\1\	This entry	http://support.automation.siemens.com/WW/view/en/38755162
\2\	Siemens I IA/DT Customer Support	http://support.automation.siemens.com
\3\	White Paper „How to improve the Performance of your Plant using the appropriate tools of SIMATIC PCS 7 APC-Portfolio?“	http://www.automation.siemens.com/w2/efiles/pcs7/support/marktstudien/WP_PCS7_APC_en.pdf

9 History

Table 9-1 History

Version	Date	Modification
V1.0	02.10.2009	First release