applications & TOOLS

Smith Predictor for Control of Processes with Dead Times

Application Note

SIEMENS

Note

The application examples are not binding and do not claim to be complete regarding the circuits shown, equipment and possibilities. The software samples do not represent a customer-specific solution. They only serve as a support for typical applications. You are responsible for ensuring that the described products are used correctly. These application examples do not release you from your own responsibility regarding professional usage, installation, operation and maintenance of the plant. When using these application examples, you acknowledge that Siemens cannot be made liable for any damage/claims beyond the scope described in the liability clause. 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.

Warranty, Liability and Support

We accept no liability for 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 in the burden of proof to the detriment of the orderer.

For questions about this document please use the following e-mail address:

mailto:online-support.automation@siemens.com

Preface

Objective of the Application

The objective is fast and tight control of processes with dead times. A dead time can be recognized by the observation that after a step excitation of the manipulated variable there is no reaction of the controlled variable at all for a certain time (the dead time). A normal PI controller must be tuned very sluggish for processes with large dead times and the control performance will decrease accordingly. The control performance can be considerably improved with the help of a so-called Smith-Predictor.

The present example shows the realization of a control loop with PI controller and Smith-Predictor starting from the respective process tag type (CFC solution template) of the PCS 7 Advanced Process Library.

Main Contents of this Application Note

The following issues are discussed in this application:

- How to create an instance and set the parameters of the process tag type.
- How to calculate the required process model for the Smith-Predictor.
- Benchmark simulations with and without Smith-Predictor, to show the potential benefits.

Validity

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

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/37361207

Index of contents

1 Basic Principles of the Smith-Predictor

Note

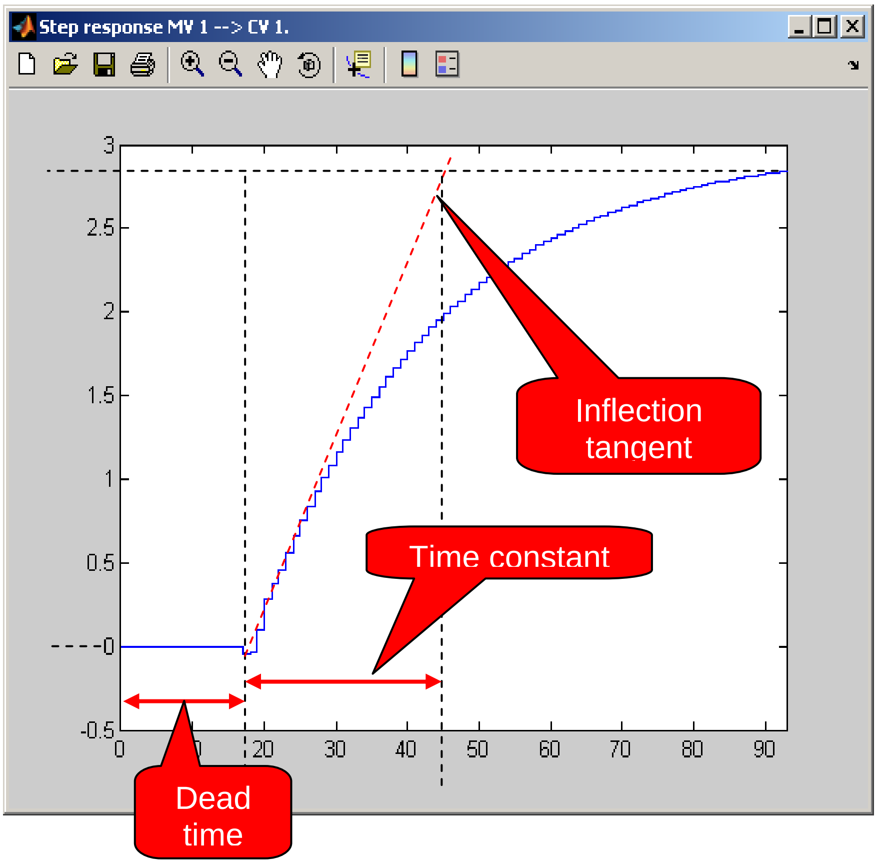
A general overview of the PCS 7-embedded APC functions (Advanced Process Control) is provided by the White Paper „How to Improve the Performance of your Plant Using the Appropriate Tools of SIMATIC PCS 7 APC-Portfolio?“

https://pcs.khe.siemens.com/efiles/PCS 7/support/marktstudien/WP_PCS 7_APC_EN.pdf

1.1 Application Area

A dead time can be detected by the observation that after a step excitation of the manipulated variable there is no reaction of the controlled variable at all for a certain time (dead time). For processes with large dead times θ (e.g. $\theta > 0.25\ t_1$ referring to the dominant time constant t_1) a standard PI controller must become be tuned very slowly and the control performance will decrease accordingly.

Figure 1-1: Example - step response of a control loop with dead time θ and time constant t_1



Example: If the parameters of a PI controller are optimized for a controlled system of the first order with dead time, in such a way that the integrated time-weighted control error is minimized, the following formula for the integral-action time can be used:

$$T_i = \frac{t_1}{1.03 - 0.165 \frac{\theta}{t_1}}$$

With increasing ratio θ / t_1, the integral-action time becomes more slowly.

The control performance can be considerably improved with the help of a so-called Smith-Predictor which can be derived using the IMC principle (Internal Model Control) of model-based control.

Application Examples

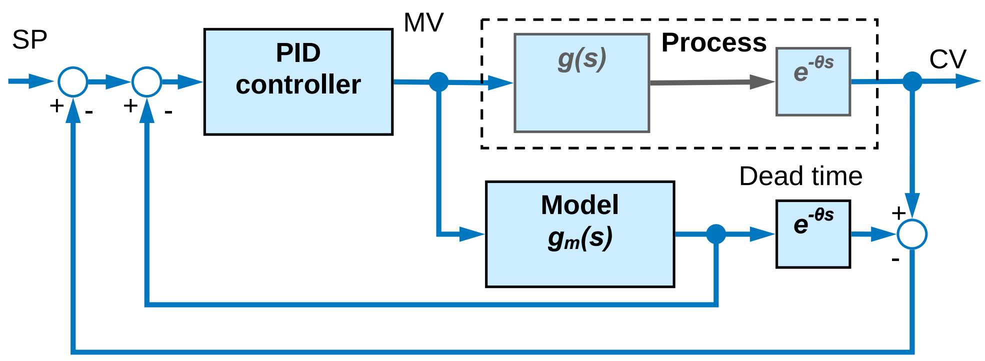
The typical cause for deadtimes in process engineering plants are run durations of fluids or gases in pipes, or run durations of bulk solids on conveyor belts.

- Temperature control via feeding of hot steam or cold/warm water in a chemical reactor jacket. After opening the valve, it takes some time until the temperated medium reaches the jacket via the pipe.
- Temperature control in chemical reactors or distillation columns via external heat exchangers. After an MV move at the heat exchanger it takes some time until the temperated medium flows back from the heat exchanger to the reactor or column via the pipe.
- Load control on a conveyor belt: the spatial distance between MV intervention and measurement system can be converted directly into a deadtime via the conveyor speed.

In terms of practical applications it must be pointed out that the performance of the Smith-Predictor depends on the model quality (model fit), i.e. the dead time must be well-known. The dead time must be approximately constant, or its value must be permanently adapted. If the speed of a conveyor belt is variable, the time-variant dead time can be derived from the measurable speed, and the respective input parameter of the dead time block in the Smith-Predictor can be modified accordingly.

1.2 Derivation using IMC-Principle (Internal Model Control)

Figure 1-2: Signal flow diagram of a Smith-Predictor



The transfer function $g_s(s) = g(s)e^{-\theta s}$ of the controlled system (process) is split into a dead time free part *g(s)* and the pure dead time part $e^{-\theta s}$. Only the controlled variable (CV) with dead time can be measured at the real process. However a virtual, estimation free of dead time can be taken from the process model. The controller can be designed for this process model $g_m(s)$ without dead time and therefore be tuned much more tightly.

To compensate for unknown disturbances, an estimate of the controlled variable affected by dead time is made in the model and compared with the genuine measured controlled variable. This difference is also fed back to the controller.

The process model (in the disassembled form) is used in the on-line control algorithm. Therefore this is called „Internal Model Control". The process model is identified from learning data, and is implemented using elementary function blocks.

2 Implementation of the Smith-Predictor

2.1 Installation

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

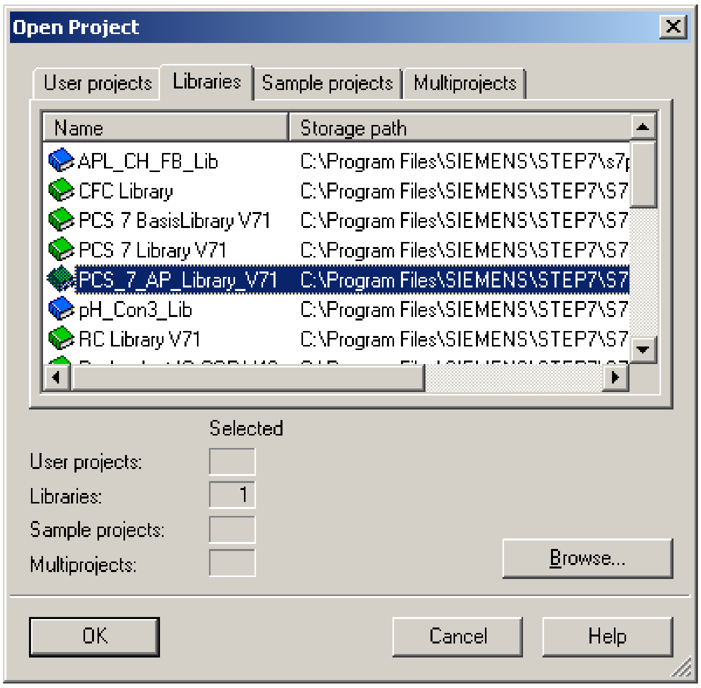
There is a Smith-Predictor as process tag type already available in the PCS 7 APC Library V7.0 SP1. With even older versions the signal flow diagram can be implemented with elementary function blocks in CFC. The MPC-configurator, which is very helpful for the identification of the process model from learning data, belongs to the shipment of PCS 7 since V7.0 SP1.

2.2 Configuration: Creating an Instance of the Process Tag Type

The following steps are carried out for the Smith-Predictor in the same way as for any other process tag type (CFC template).

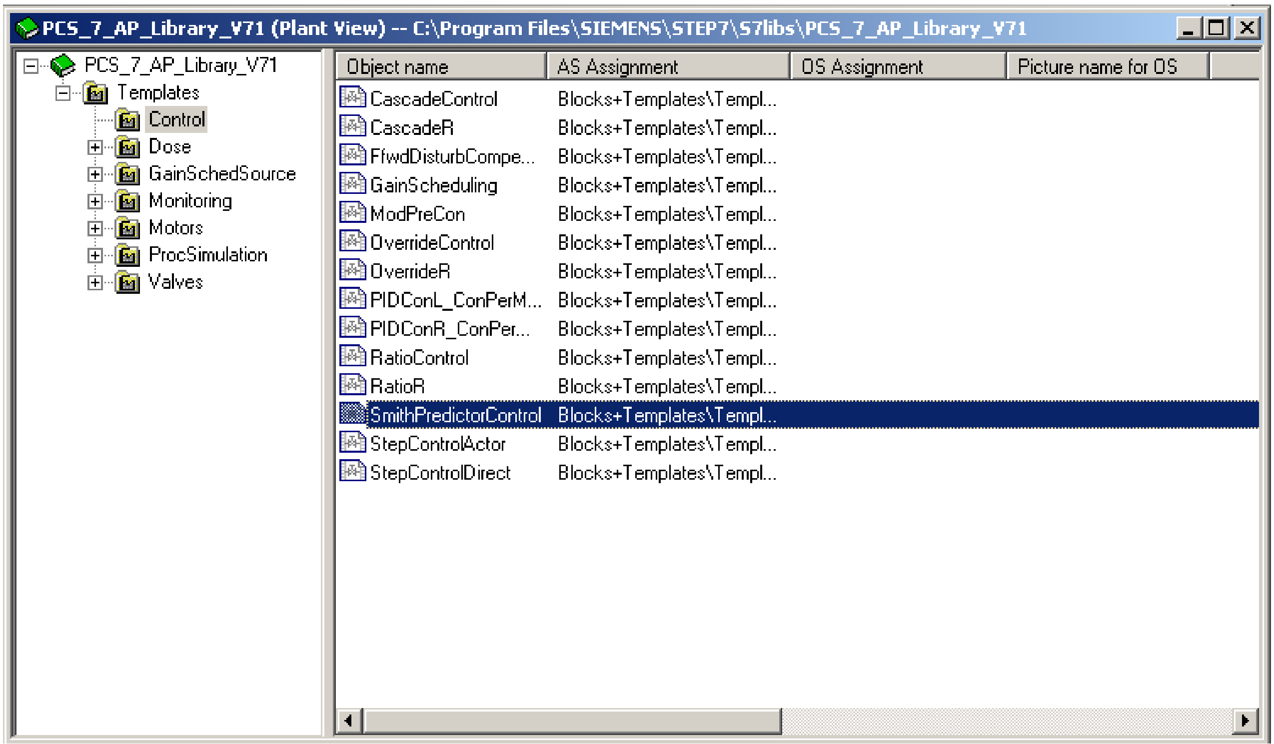
1. Open the Advanced Process Library in SIMATIC Manager over menu `“Open project > Library > PCS 7 AP Library V71”.`

Figure 2-1: PCS 7 AP Library V71



2. Copy the process tag type „SmithPredictorControl“ from the Plant View, folder “Templates > Control” to the master data library <project-name>_Lib of the PCS 7 Multiproject, into the folder “process tag types > Control”.
3. Modify the process tag type if necessary according to the general requirements of the application.

Figure 2-2: Copy the process tag type „SmithPredictorControl“

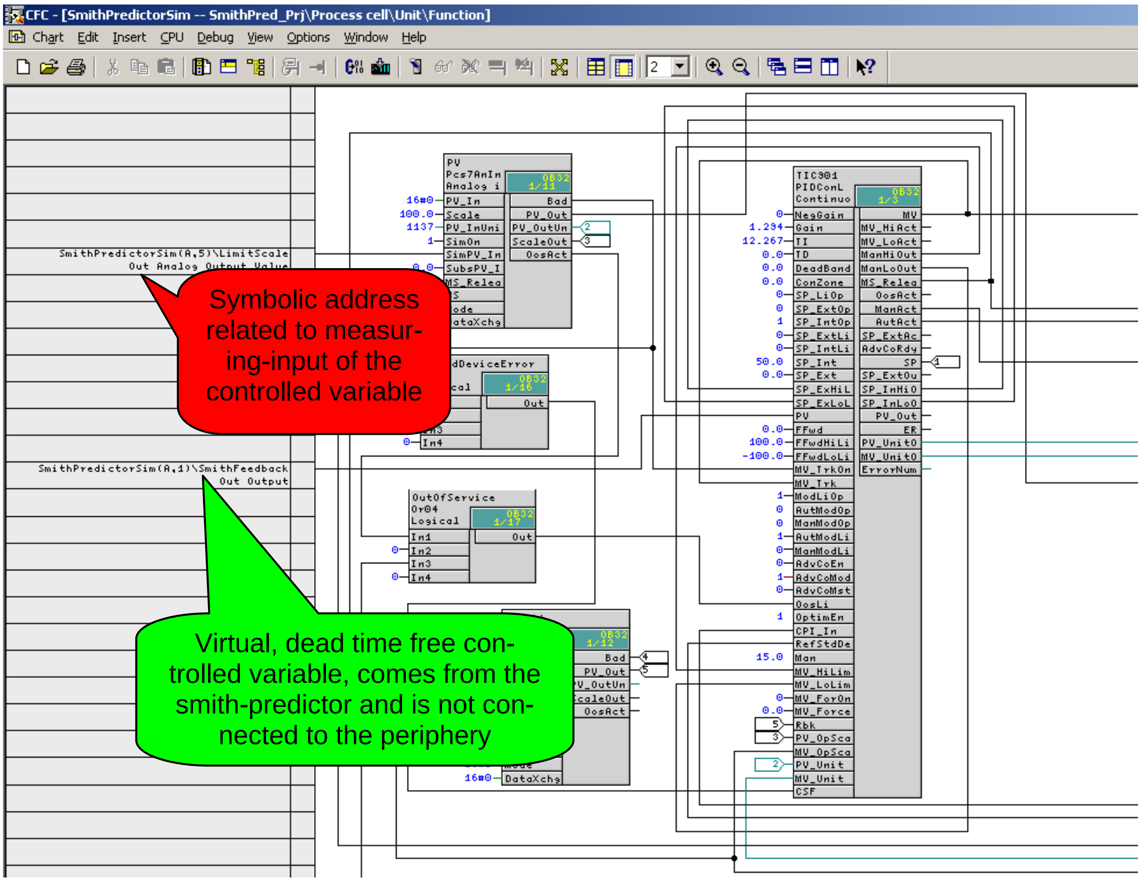


4. Copy the process tag type from master data library <projectname>_Lib of the Multiproject to the target folder (Process cell > unit etc.) of the Plant View.
5. You obtain an instance of the process tag type (a „process tag“) i.e. a CFC chart which indicates its origin from a process tag type by its symbolic representation.

Numerous instances of a process tag type can be generated automatically with the import export assistant using parameters from an import file.

6. Connect the channel driver blocks “Pcs7AnIn” and “Pcs7AnOut” to the controlled variable, manipulated variable and where applicable to the feedback of the manipulated variable using the symbolic names of the related peripheral signals from the hardware-configuration.

Figure 2-3: Connecting the channel driver blocks

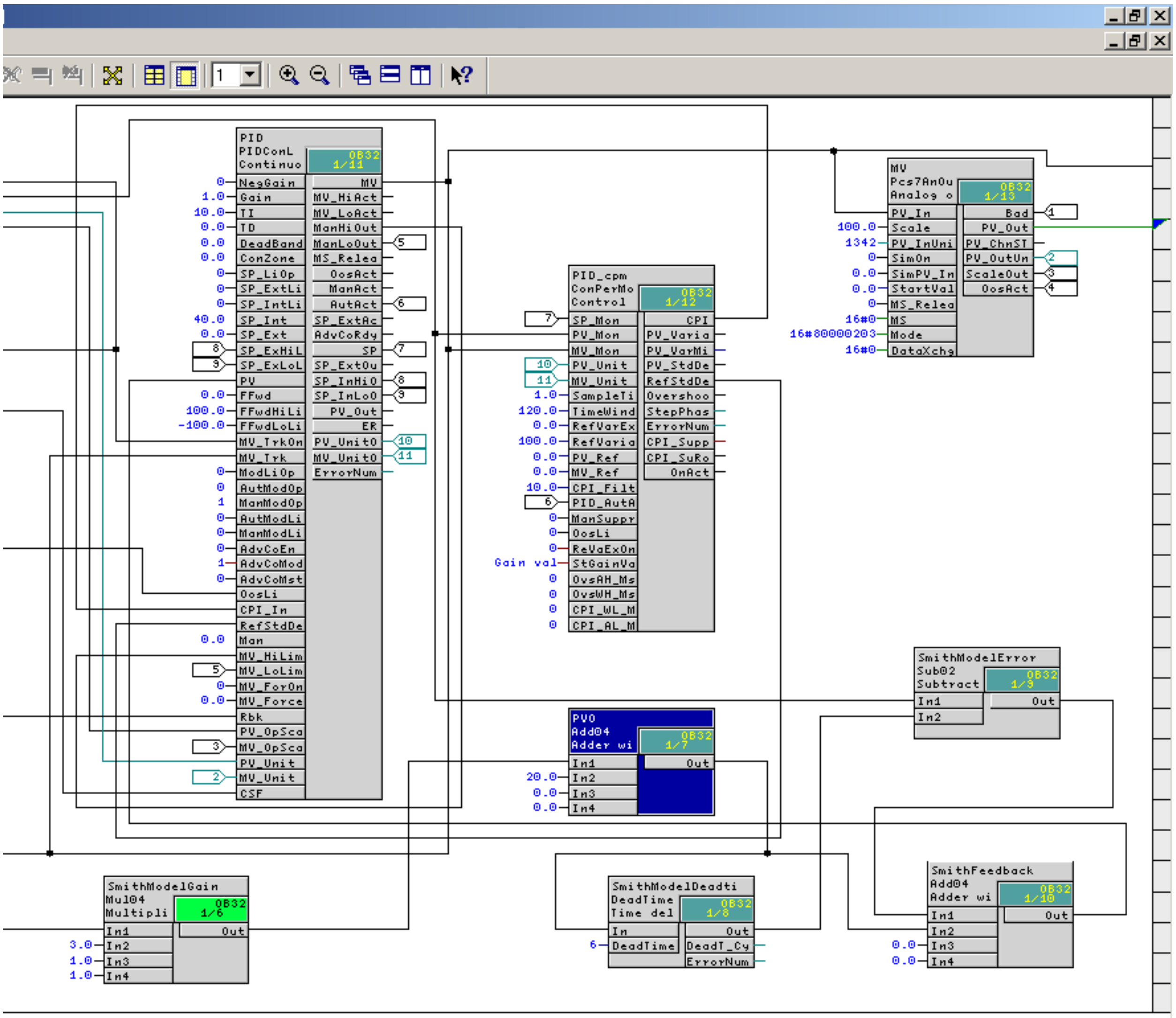


NOTE **The input “PIDConL.PV” in the process tag type is already connected to a virtual, dead time-free controlled variable of the Smith-Predictor. This input must not be connected to analog periphery.**

Most control systems have a controlled variable unequal to zero for a manipulated variable equal to zero. This offset PV0 of the controlled variable should be included in the process model of the Smith-Predictor. Ignoring this offset doesn’t really matter for processes with positive gain, because it is compensated by the integral action of the controller. Nevertheless, for processes with negative gain, this offset is mandatory for the Smith-Predictors.

7. Insert the Add04 block marked in blue in front of the DeadTime block of the predictor.

Figure 2-4: Consideration of the offset PV0 in the CFC chart of the Smith-Predictor



8. Connect the output „PV0.Out” both to the input “SmithModelDead-time.In” and the input “SmithFeedback.In2”.
9. Set the parameter „PV0.In2” to the value where the controlled variable CV settles in a steady state if the value of the manipulated variable is MV=0.

3 Identification of the Process Model from Learning Data

For the Smith-Predictor is required to describe the process behavior as a transfer function with separate dead time. A description of the process behavior as transfer function could in principal be determined by the application of the PCS 7 PID tuner to the PID controller in the process tag, but the PID tuner uses a model type where dead times are approximated by higher model order, so the PID tuner delivers no numerical values for the dead time. Therefore the model type of the PID tuner is not suitable for the Smith-Predictor.

Hence, it is suggested to use the MPC-configurator to determine the process model.

This approach consists of 3 steps:

- Process excitation and recording of learning data.
- Modeling (incl. determination of the dead time) with the MPC-configurator.
- Transfer of the parameters (dead time, ...) to the Smith-Predictor.

NOTE For processes with large dead times, a model predictive controller is also suitable in a single-input single-output situation. It provides greater flexibility in system modeling and is more convenient thanks to the integrated design tool. However, it does require more CPU resources, and does not allow online adaptation of the dead time value.

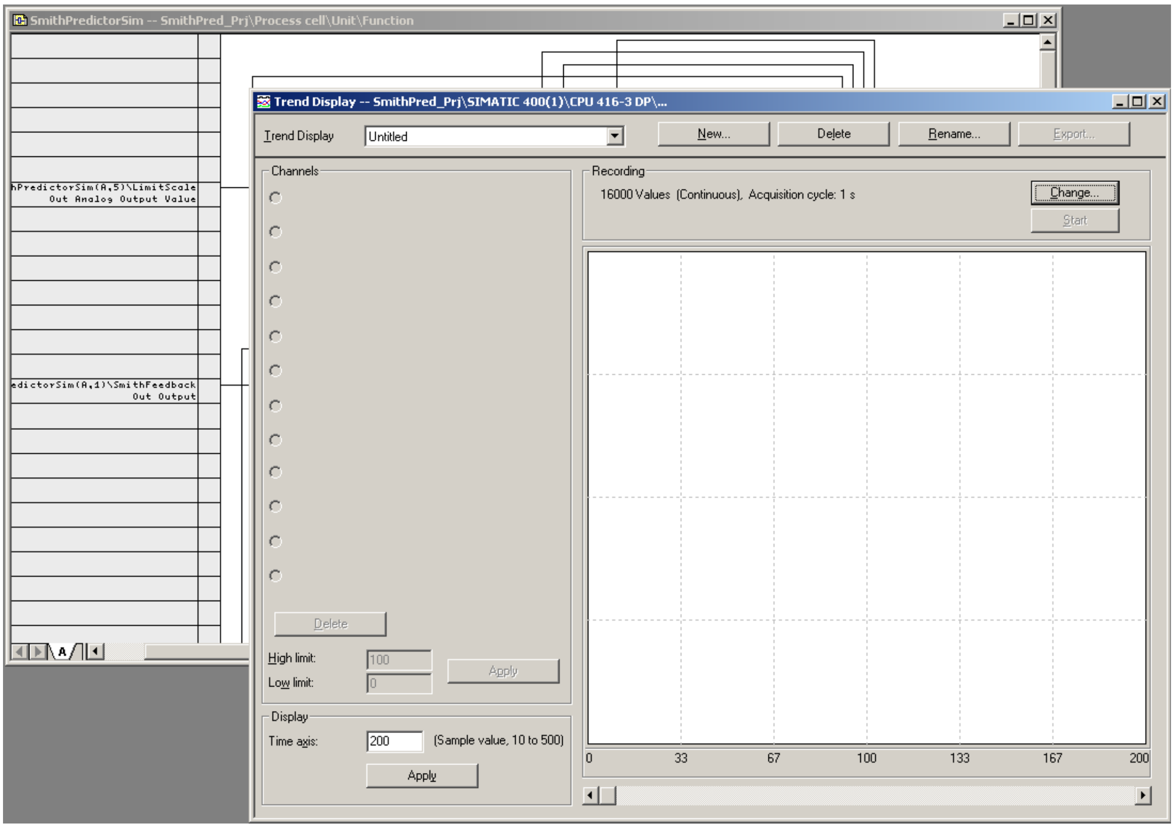
NOTE The following experiments were carried out using the example project “APL_Example_EU” from the Advanced Process Library. However all time constants were increased by a factor of 5 compared with the state of delivery, in order to allow process identification with a sample time of 1 sec.

3.1 Process Excitation and Recording of Learning Data

The process is excited in manual mode of the controller with steps of the manipulated variable, or if an at least stable rough controller is already available, in automatic mode with setpoint steps. The measuring data are recorded by the CFC Trend Display and exported to an archive file.

1. In CFC choose the menu “View > Trend” Display to open the window of the trend display.
2. Push the button „New“ and give a name to the new trend display.

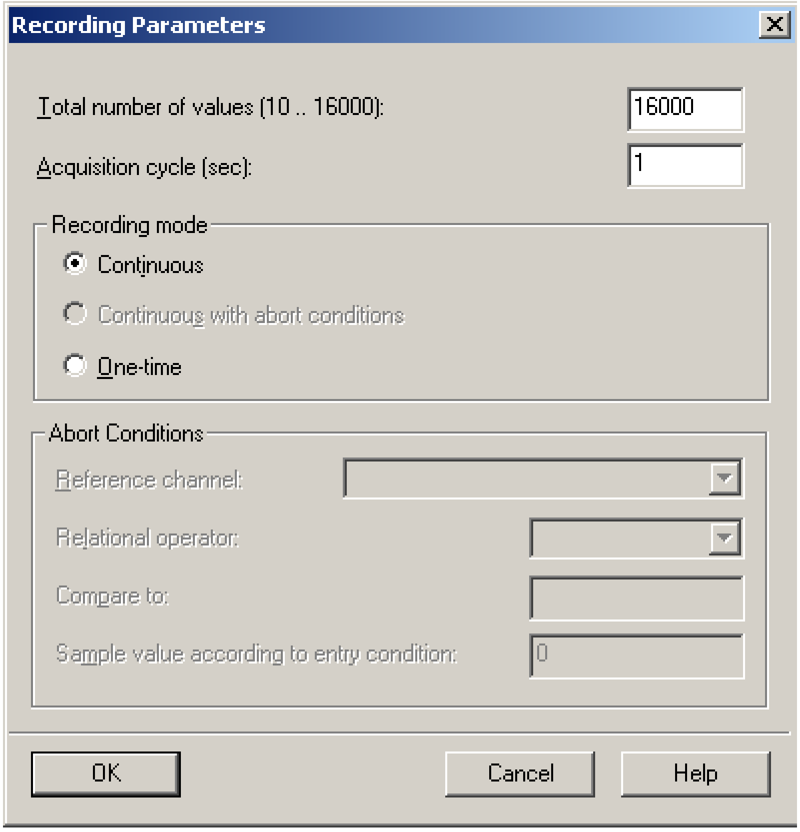
Figure 3-1: Trend Display in CFC



3. In the area "Recording" you open the dialogue field "Recording parameters" by clicking the button "Change...".
4. Put here a sufficient number of measuring values. The number of measuring values multiplied by the sample time provides the overall time space for a file. Besides, the sample time has to be an integer multiple of the cyclic task (e.g. OB32: 1s).

5. The capture period must be long enough to carry out at least two step experiments with the manipulated variable and to wait for the process to settle in each case. Moreover, the acquisition cycle may not be too big.

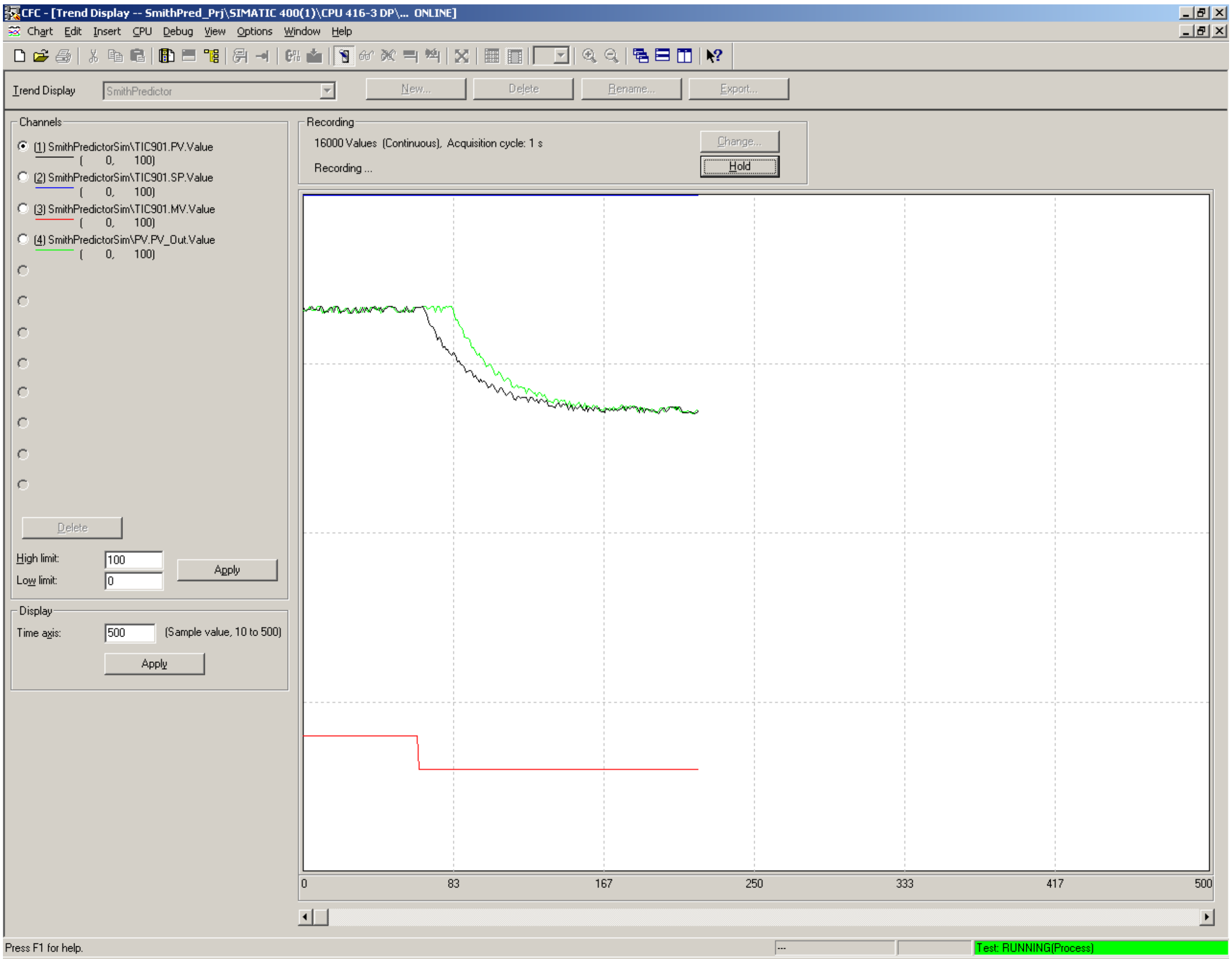
Figure 3-2: Recording parameters



NOTE **The acquisition cycle should be fixed in such a way that the shortest step response can be captured with approx. 200 samples.**

6. In CFC, select the manipulated variable and the controlled variable of the PID controller, for which a Smith-Predictor should be added.
7. Drag&Drop these variables to the Trend Display.

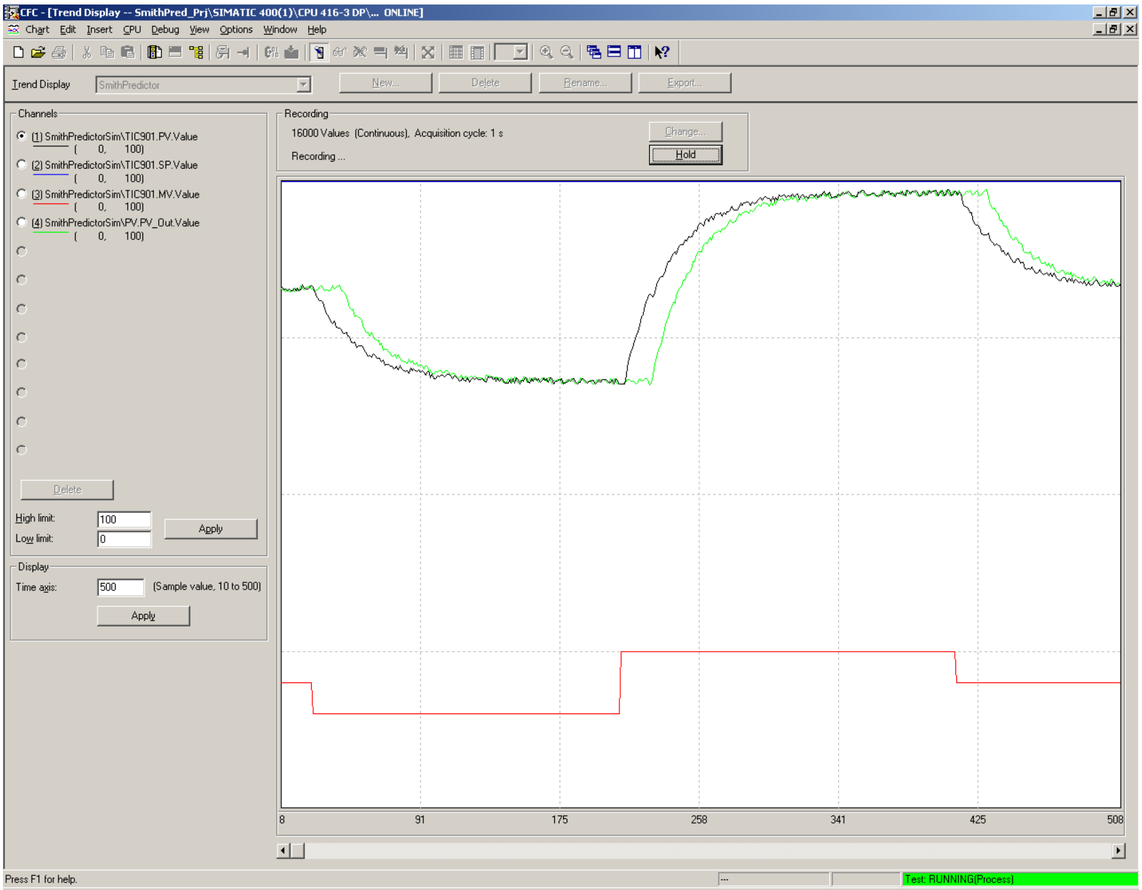
Figure 3-3: Run of the Trend Display



NOTE Take the original measured value “PCS7AnIn”. “PV_Out” of the analogous input driver (in the picture: green), and not the value “PIDConL.PV” of the controller for process identification, because in the process tag type the latter already shows a virtual dead time-free value, which is still absolutely irrelevant at the time of learning data recording.

8. Move the process to the operating point.
9. Wait for a steady state of the process and start recording. The CFC has to be in test mode for recording.
10. Excite (stimulate) your process. The manipulated variable can be operated in manual mode of the controller either in the OS faceplate or at the input parameter „Man" of the controller block in CFC.

Figure 3-4: Run of the Trend Display



NOTE Consider the notes related to process excitation in the manual of the MPC configurator.

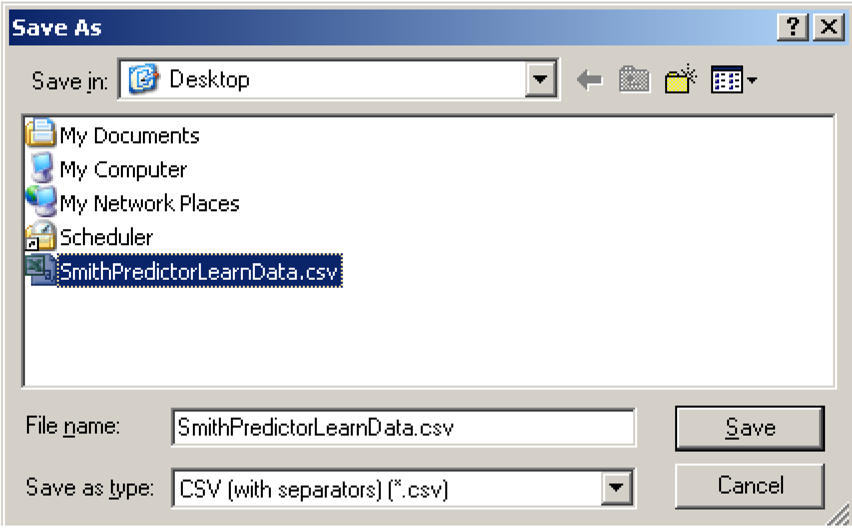
Proposal:

First step the manipulated variable (MV) down. Wait until the process has reached a new steady operation point. Then switch the next step upwards, and let the process reach a steady state. Set the manipulated variable (MV) back to the operation point and let the process come to a steady state. The excitation signals for the MPC-Konfigurator should in general be symmetrical to the operation point. As opposed to the PID-Tuner, a single step response is not sufficient.

11. Stop recording and export the trend data of the CFC into a file.

12. Preferably apply the default parameters for the file export (separator etc.).

Figure 3-5: Export of the trend data file



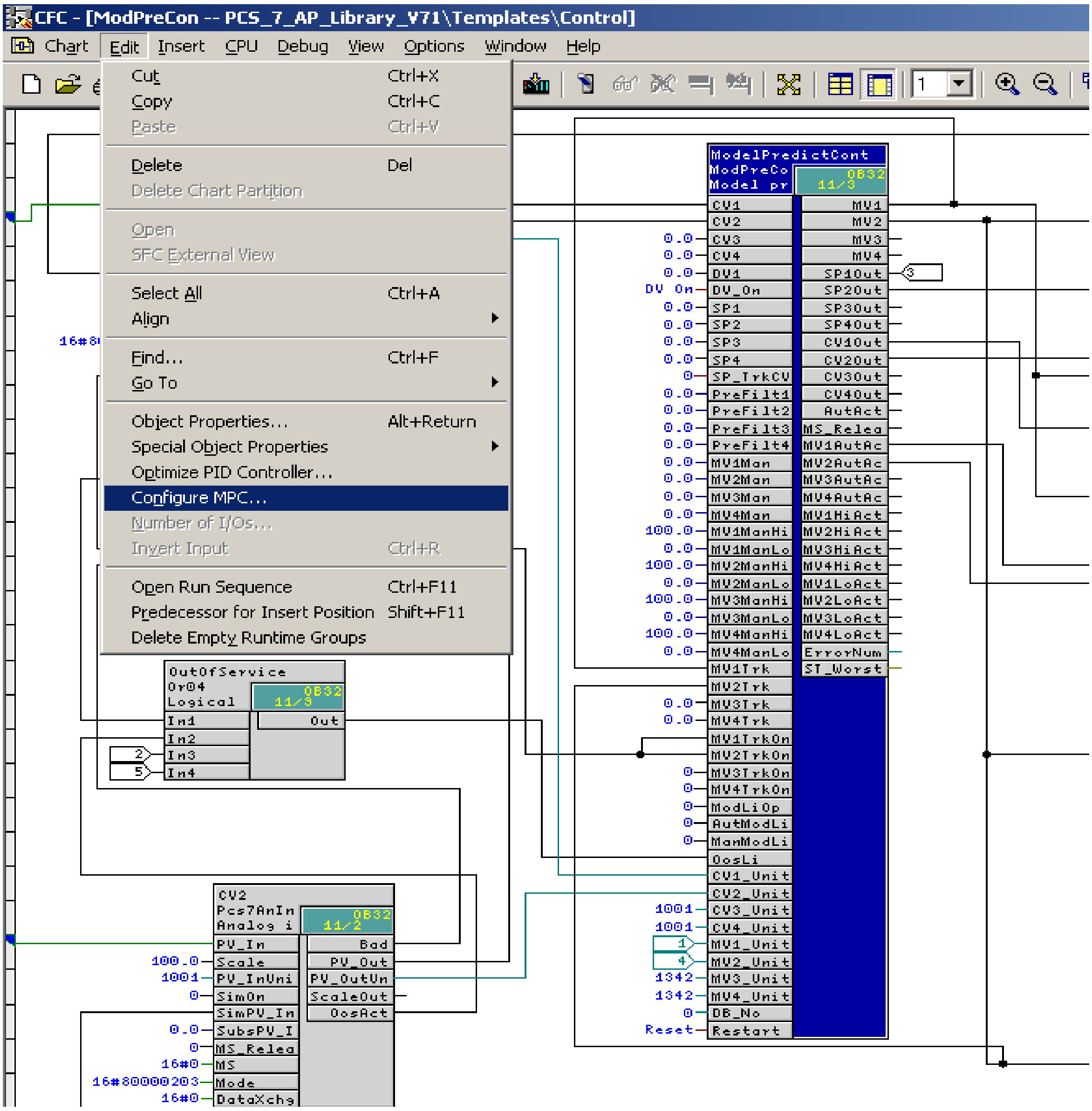
3.2 Modeling with MPC-Configurator

Start the MPC-Configurator. With PCS7 V7.0 SP1 you will find it in Windows Startmenu "`Start > SIMATIC > Step7 > Engineering Tool ModPreCon`".

For PCS 7 V7.1 you have to open a CFC which includes the function block ModPreCon. If you do not yet have a predictive controller in your project, you can also use the process tag type "ModPreCon" from the Advanced process Library.

1. As soon as you have selected a function block ModPreCon, you can and go to CFC main menu "`Edit > Configure MPC…`".

Figure 3-6: Configure MPC

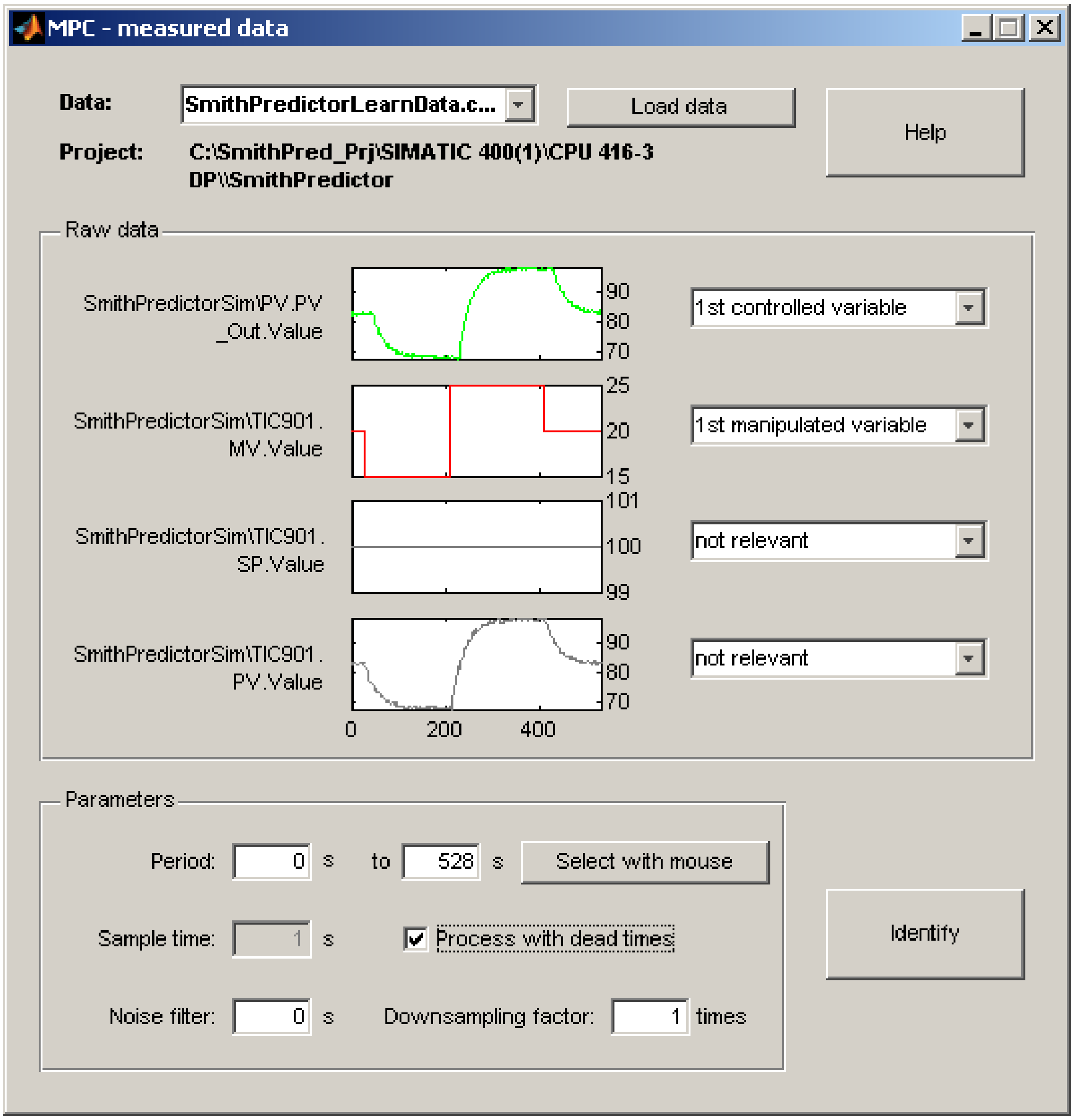


2. Load the recorded learning data. (The MPC-configurator automatically opens the CFC archive file recently saved.)
3. Associate the appropriate columns of the archive file to the 1st controlled variable and 1st manipulated variable.

NOTE You remember, choose the controlled variable from the input driver block and not from the controller block!

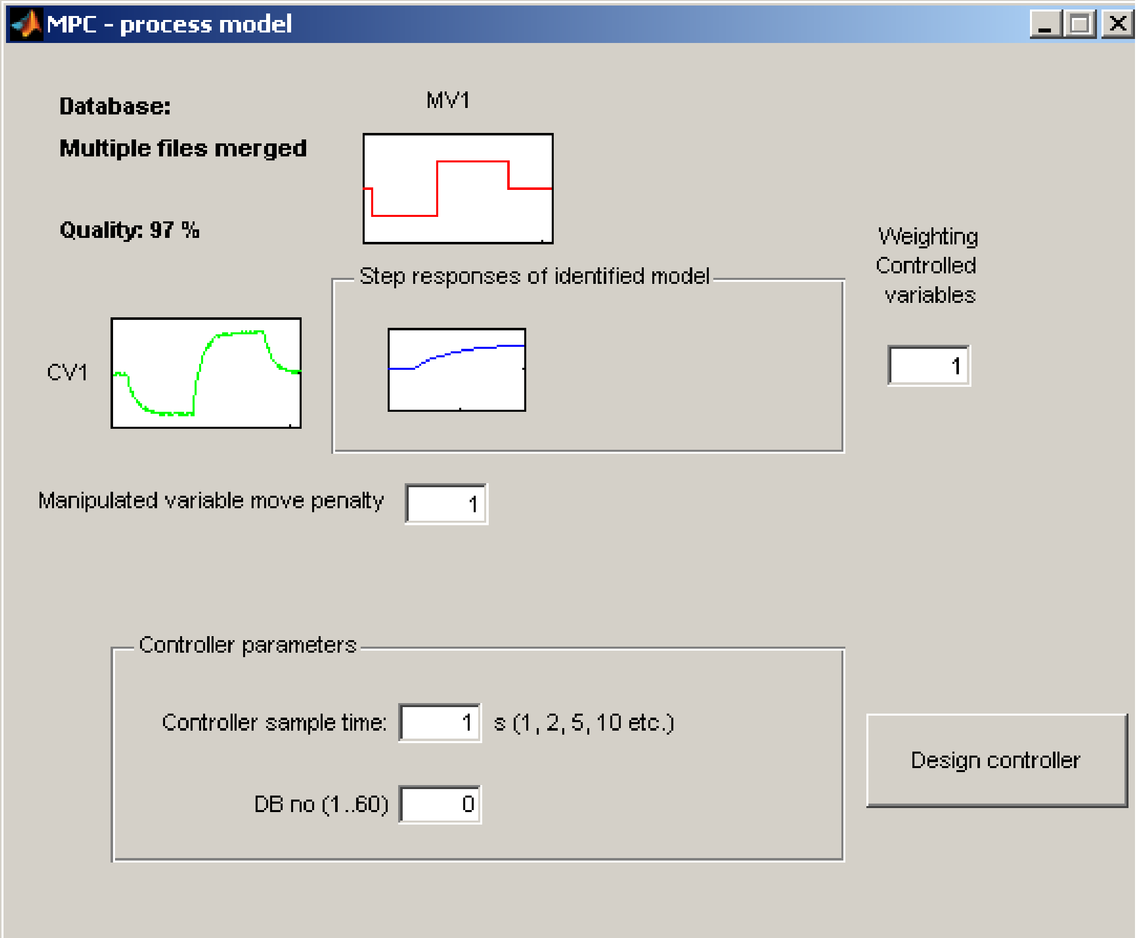
4. Make sure to mark the tick box „Process with dead times“. Start the identification with the button „Identify“.

Figure 3-7: MPC - measuring data



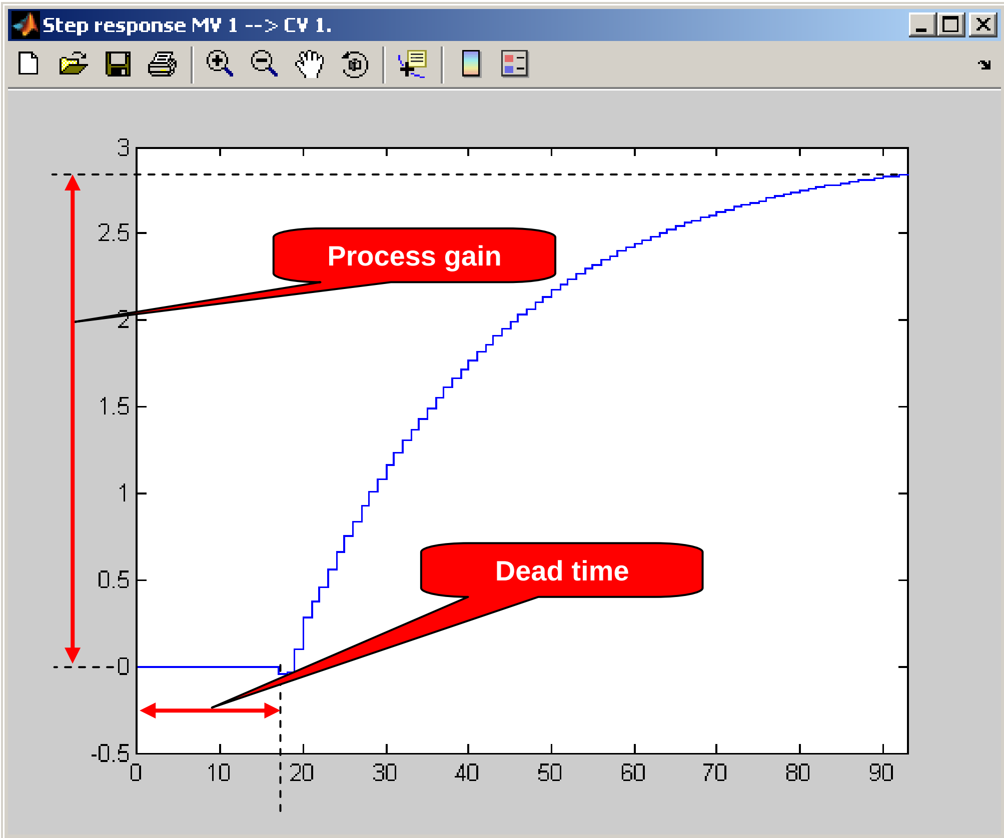
5. The result of identification is the process model. Click in the field of the transfer function G (1,1) = step response from MV1 to CV1 to show an enlarged display of the unit step response of the process.

Figure 3-8: MPC - process model



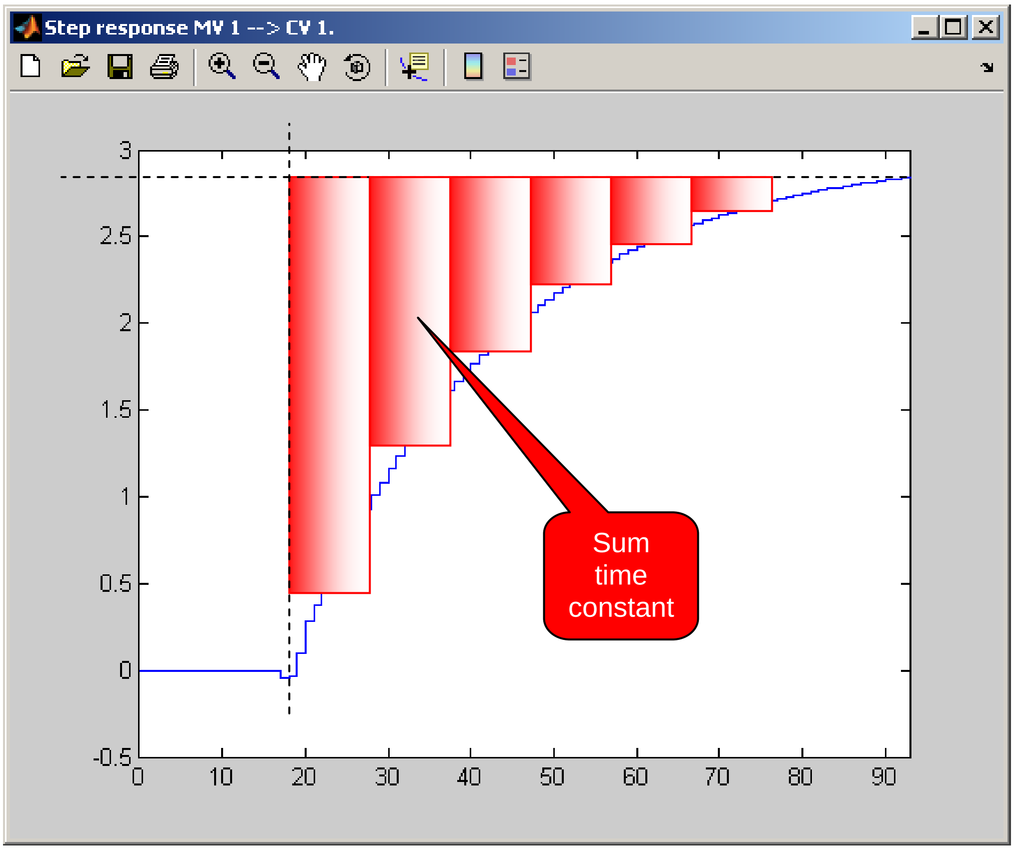
6. Take a reading of the dead time at the front end of the step response. The time axis is always scaled in seconds. Maximize the diagram window and use the button „Data cursor“ from the toolbar above the diagram for exact reading. In the example you obtain a dead time of 17 sec.
7. Read the stationary process gain at the rear end of the step response. In the example you obtain a value of 2.84.

Figure 3-9: Step response



8. The sum time constant is determined from the area above the step response, after division by the process gain. Relevant for the Smith-Predictor is the sum time constant of the dead time free part!

Figure 3-10: Step response



9. To graphically determine the size of an area, it is advisable to split the area into rectangular beams. The beams are cut such that the area is approximately the same than the desired area.

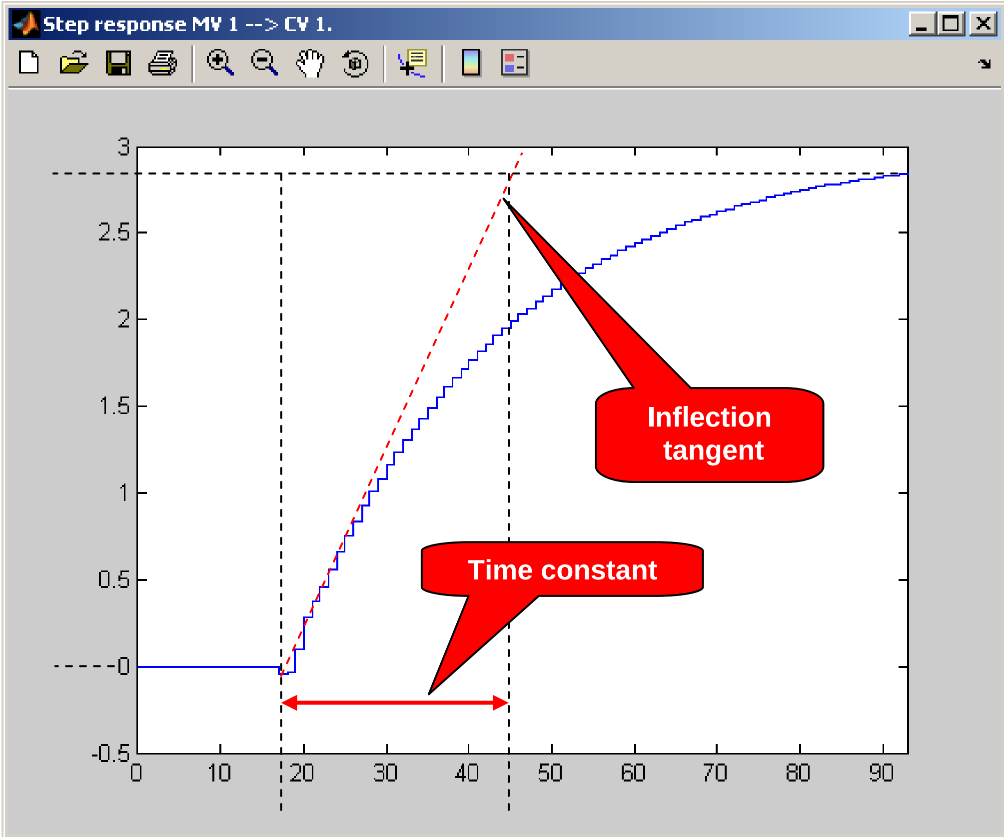
10. In the example you obtain:

$$T_{\Sigma} = \frac{1}{2.84}\begin{pmatrix}[2.84-0.5]+[2.84-1.3]+[2.84-1.8]..\\ +[2.84-2.3]+[2.84-2.5]+[2.84-2.6]\end{pmatrix}10s \approx 21s$$

11. Fitting an inflection tangent to the steepest part of the step response is a little bit easier, but less exact. The time span from the beginning of the step response (at the end of the dead time) up to the intersection of the inflection tangent with the final value is also an estimation of the time constant.

12. In the example you obtain:
44s - 17s = 27s
The inflection tangent does not necessarily intersect with the zero horizontal line at the end of the dead time; it can also intersect to the right of it.

Figure 3-11: Inflection tangent in step response



Generally, estimation errors of the process time constant have less serious consequences for the control performance of the Smith-Predictor. It is more important that dead time and gain are determined exactly.

13. Note the estimated process parameters and close the MPC-configurator (The MPC control design is not relevant for the Smith-Predictor).

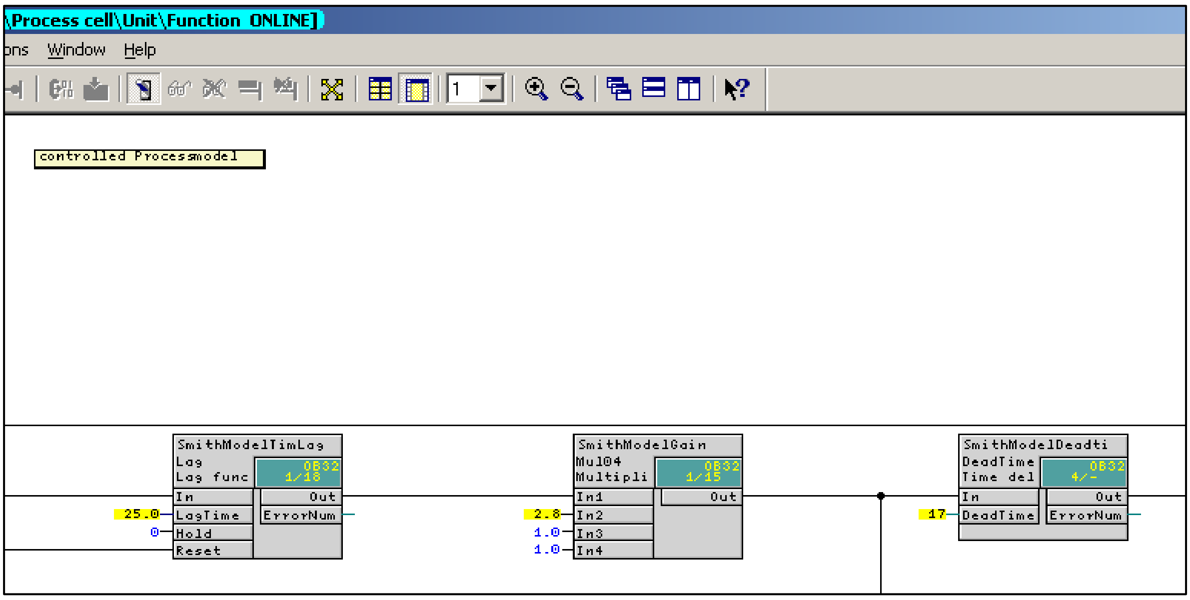
NOTE

It is planned for one of the next PCS 7 versions to extend the MPC-configurator in such a way that the identified models are displayed not only graphically, but also with numerical values. Then the reading of model parameters from graphic step responses will not be necessary any more.

4 Parameterization and Commissioning

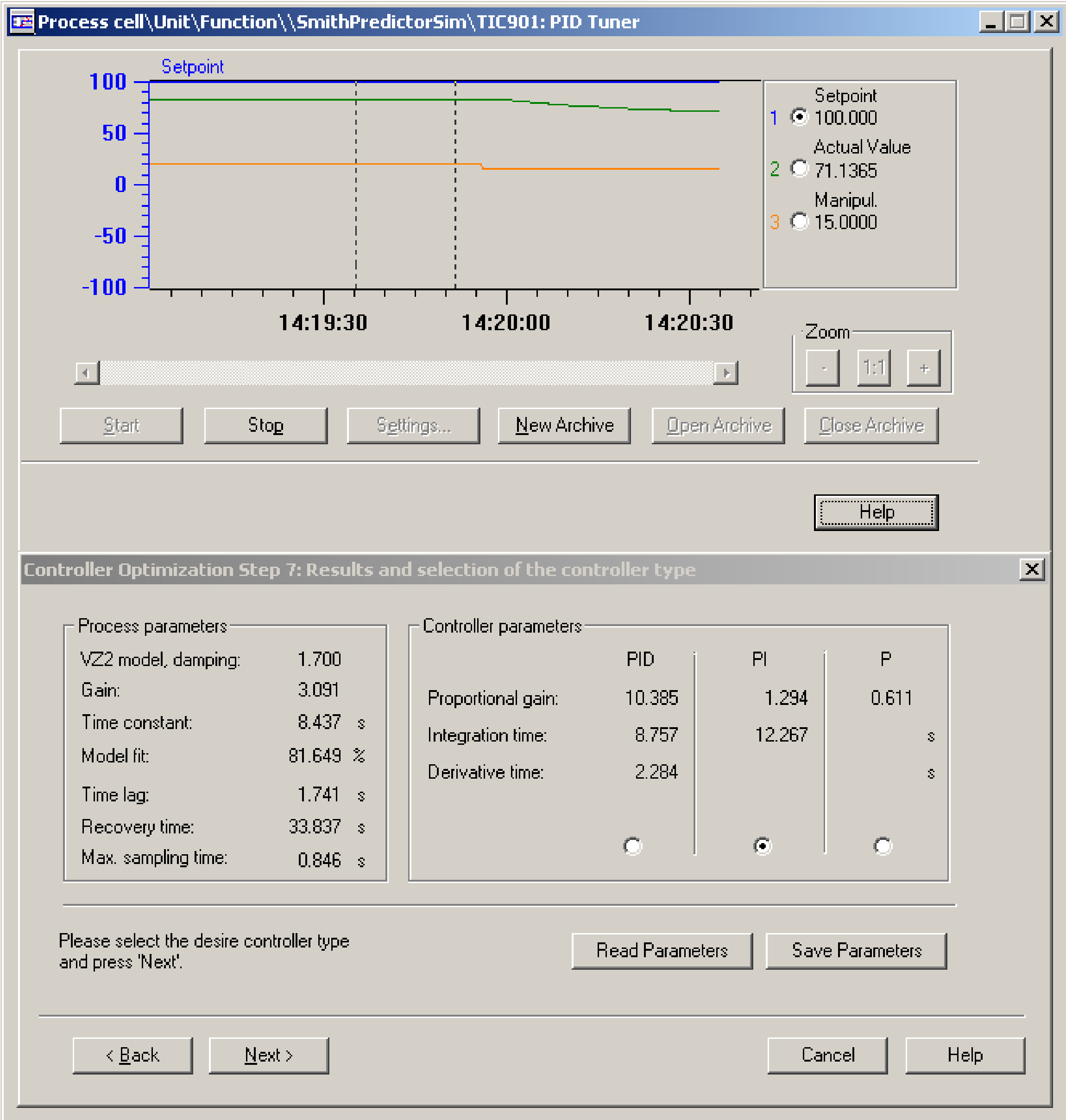
1. Assign the parameters dead time, process gain and time constant to the related function blocks in the CFC, in the sheet above the PID controller.

Figure 4-1: Input of controller parameters



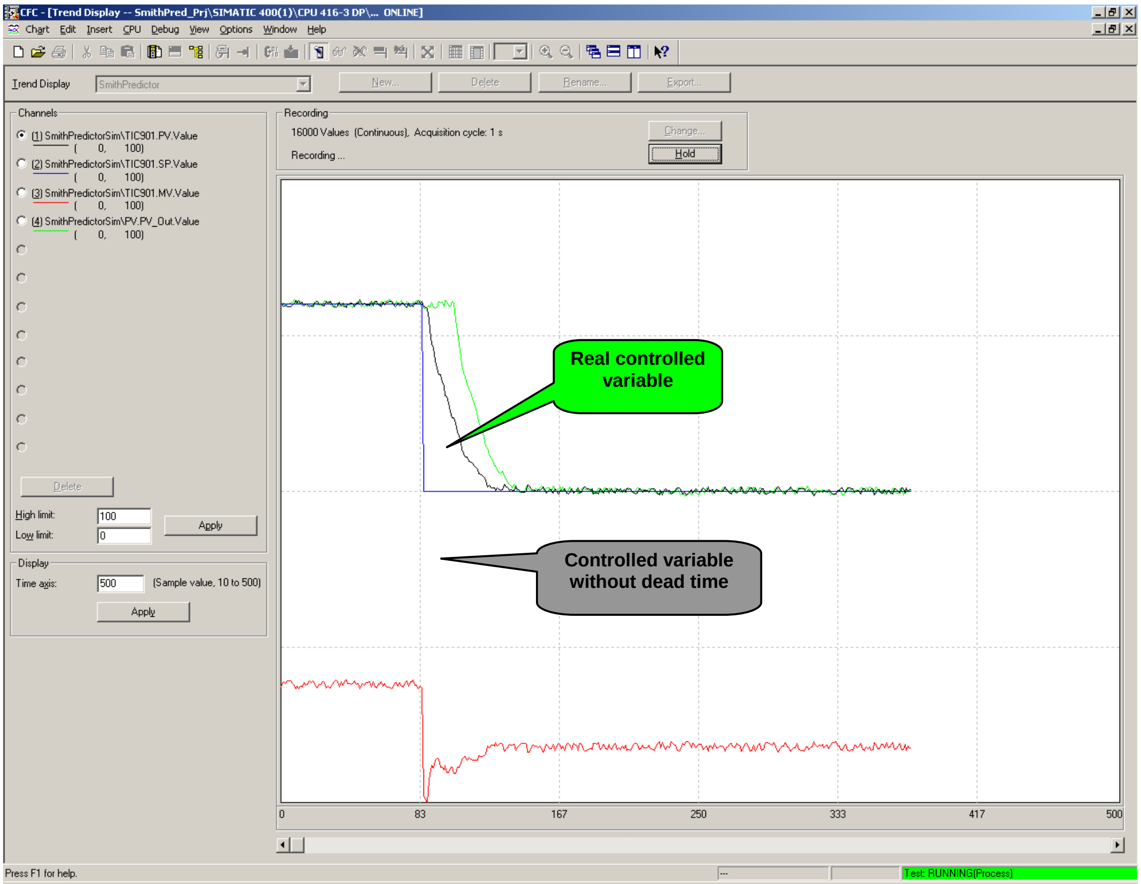
2. Optimize the PID controller with PID tuner. The parameters are automatically customized for the dead time free control loop, because the input “PIDConL.PV” is connected to the virtual, dead time free controlled variable!

Figure 4-2: Smith Predictor



In this CFC- trend the real controlled variable is displayed in green, whereas the dead time-free controlled variable, predicted by the Smith-Predictor, is black. This predicted controlled variable is connected to the controller and is therefore also shown in the faceplate.

Figure 4-3: Trend Display of real value and dead time free actual value



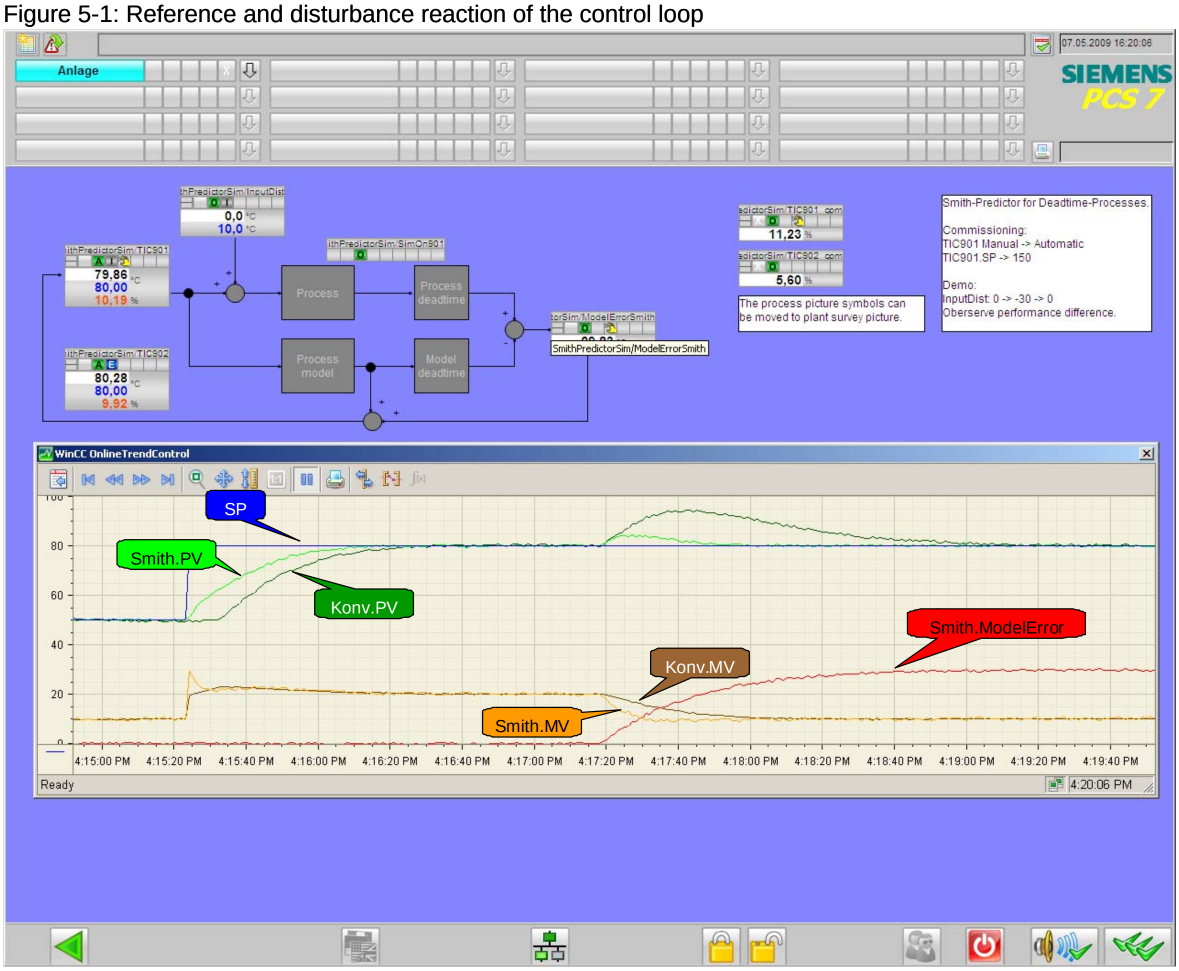
5 Simulation Example

In the example project “APL_Example_EU” the same process simulation is realized twice, one instance with Smith-Predictor and the other without, while all the other process parameters are identical.

The signal flow diagram shows the control loop with Smith-Predictor and PI controller TIC901. Below the faceplate of TIC901 is the faceplate of the conventional controller TIC902without Smith-Predictor (without associated signal flow diagram). In a direct comparison ("benchmark simulation", "parallel slalom") the benefits of the Smith-Predictor can be shown. The advantage both in setpoint response performance and in disturbance compensation is obvious.

- Blue: Setpoint SP for both controllers.
- Light green and orange: controlled variable PV and manipulated variable MV of the controller with Smith-Predictor.
- Dark green and dark orange: controlled variable PV and manipulated variable MV of the controller without Smith-Predictor.
- Red: Difference between measured value and prediction.

Figure 5-1: Reference and disturbance reaction of the control loop



Robustness of Smith-Predictor against Modelling Errors

The good simulation results in Figure 5-1 are achieved in spite of the inaccuracies of modeling. In the example the process gain 2.84 was used in the Smith-Predictor instead of 3, and the sum time constant 21sec instead of 25sec. In this case a very easily controllable system of the first order and a robust parameterization of the PI control for optimum command response without overshoot are used. Similarly, a simulation with a control system of the second order which is approximated by a predictor model of the first order with equivalent sum time constant can be controlled easily with the described approach. In practical applications the robustness against modeling errors (in particular against under-estimated dead times) will be lower than in the described simulation example.

Dead times assumed clearly too high (e.g., 12 instead of 8s) are tolerated (see. negative setpoint step to the left of the picture).On the other hand, it is more dangerous to under-estimate the dead time, i.e. to assume a dead time of only 4sec instead of 8sec (see. positive setpoint step to the right in the picture). With a wrong parameterization of 2sec instead of 8sec for the dead time, the control loop finally starts to oscillate.

Figure 5-2: Oscilation of the control loop

Manual fine-tuning of the dead time or the sum time constant of the Smith-Predictor does not make sense, because the accuracy of the applied parameters can be estimated only roughly from the deviation "ModelErrorSmith" between real measured value and controlled variable predicted by the Smith-Predictor.

However fine-tuning of the PI-controller gain is possible, as familiar from a PI controller without Smith-Predictor. A robust parameterization makes sense in combination with the Smith-Predictor, i.e. considerably more tight controller parameters than suggested by PID tuner are not advisable.

The bigger the dead time in relation to the sum time constant, the bigger is the difference of the controller parameters with or without Smith-Predictor, i.e. the more performance profit is achieved by the structure with predictor. However, this also means that the controller depends more heavily on his predictor, and the robustness against model errors decreases accordingly. Take care if the dead time amounts to more than the triple of the sum time constant.

6 The Smith-Predictor in the Operation Phase

The following issues have to be considered in context with a Smith-Predictor in the operation phase of a plant:

- In the faceplate of the PI(D) controller the dead time-free prediction value is shown instead of the original measured value, but.
- The Smith-Predictor does not need a faceplate of its own, because there is nothing to operate during the operation phase.
- If the dead time in the operation phase is subject to strong variations, and the actual value of the dead time can be computed from other variables, it makes sense to link the respective input parameter of the Smith-Predictor.
- The control loop with Smith-Predictor behaves similar to a conventional control loop with respect to unknown disturbances. In face of strong measurable disturbances, a combination of Smith-Predictor and feedforward disturbance compensation is feasible in principle. Nevertheless the application of the ModPreCon block suggests itself in such cases. Although the realization of a combination of Smith-Predictor and dynamic disturbance compensation using elementary CFC blocks would be theoretically possible, this will become a quite unmanageable wiring construction.
- Control performance monitoring can be applied to the Smith-Predictor similar to a simple control loop. It makes sense to consider the real controlled variable instead of the predicted controlled variable for monitoring.

7 Conclusion

The following table compares the Smith-Predictor both to a conventional PID controller and to a model based predictive controller.

Table 7-1: Compare of the controller

	PI(D) controller (conventional)	PI(D) controller with Smith-Predictor	MPC
SISO/MIMO case (single input single output / multi variable case)	SISO only	SISO only	SISO or MIMO
Control performance	low	good	very good
Engineering-effort	low	high	medium
CPU resource-consumption (memory, calculation time)	low	minimal higher than conventional	high
feedforward disturbance compensation	can be supplemented	difficult to supplement	already integrated

8 History

Table 8-1 History

Version	Date	Modification
V1.0	July 2009	1st release