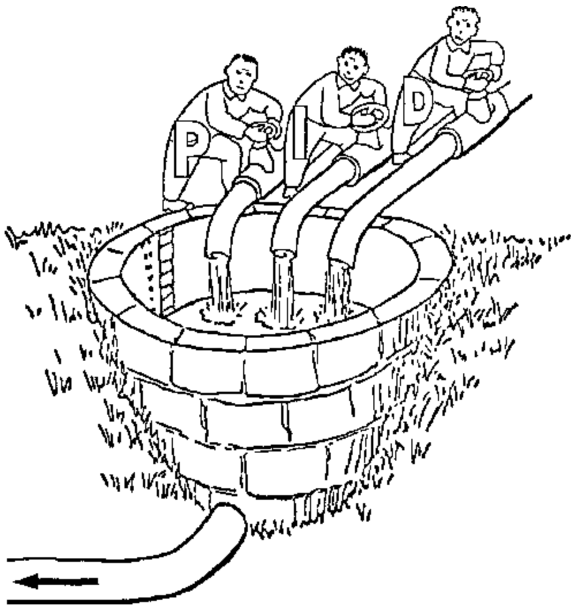
PID Autotuning Tool PMA*Tune*

from version 2.0



Contents

Page

PMATune has been developed by REX Controls. PMATune uses PMA Tuning Server for communication with KS Controllers and their simulation tools.

PMA Tuning Server.

1. PMATune Licensing

The present version of PMATune is available in the following licenses:

- Free demoversion (`www.pma-online.de` or PMA CDROM)
- Single license 9407-999-06601
- Multi license (5 users) 9407-999-06611

After the first PMATune launching the following dialog occurs:

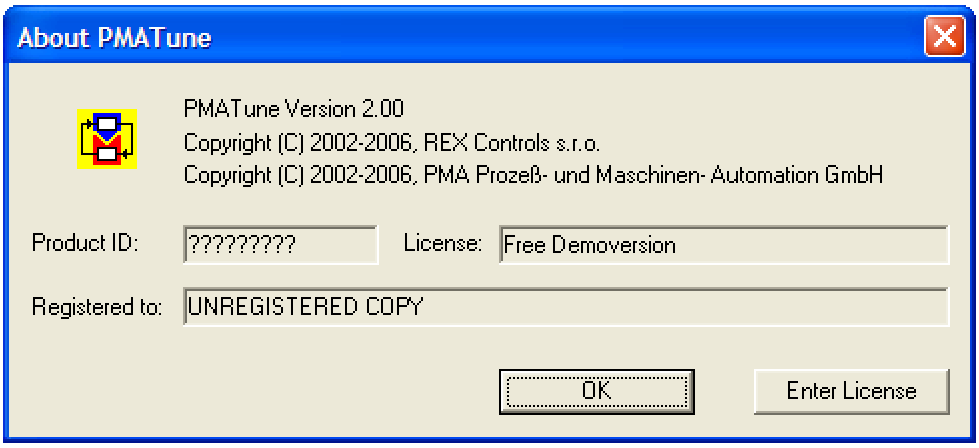


Fig. 1: About dialog box

By pressing the button `OK` the program is started as a demo version. Now, it can be tested with built-in process model or in combination with PMA Simulation Tools (BlueControl, SIM/KS94, SIM/KS98).
To obtain one of the full licenses, please contact PMA GmbH. After receiving the desired license, fill in the `PMATune License` dialog activated by pressing the button `Enter License`. Copy (`Ctrl+C`) and paste (`Ctrl+V`) may be used!

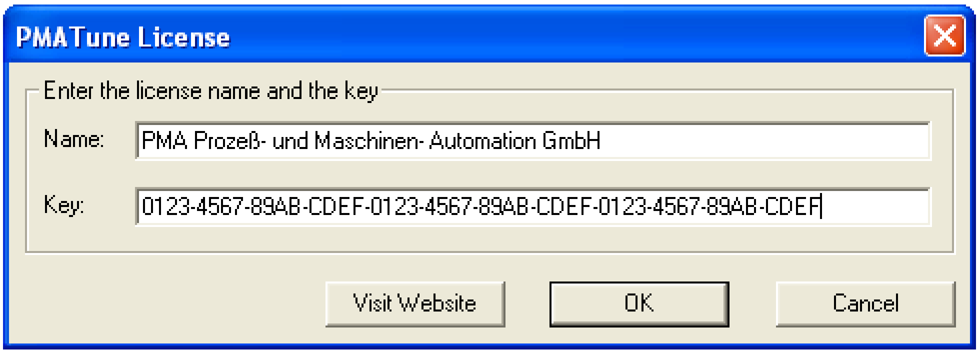


Fig. 2: License dialog box

2. What is PMATune

The main highlight of PMATune is the new PID robust tuning method which never fails under the assumption that the process can be described with the sufficient precision by an arbitrary order lag / dead time transfer function in the form

$$F(s) = \frac{Ke^{-Ds}}{s^{l}(t_1 s+1)(t_2 s+1)\dots(t_n s+1)}$$

where :

K	process gain
$D > 0$	process dead time
$t_i > 0$, $i = 1, 2, \dots, n$	process time constants
$l = 0$ or 1	processes without integrator ($l = 0$; *static process*) or with integrator ($l = 1$; *astatic process*) ,
n	arbitrary integer number

Therefore, this method gives satisfactory results also for dead time dominated processes (well-known as difficult to control), unlike the most existing methods. Not applicable are processes with “strong non-minimal phase behaviour” and substantially oscillating processes. Various types of process step responses are illustrated in figure below. (The term “Slightly non-minimal phase” / “Slightly oscillating” means approximately that the undershoot / overshoot is less than 10 % of the steady state value.)

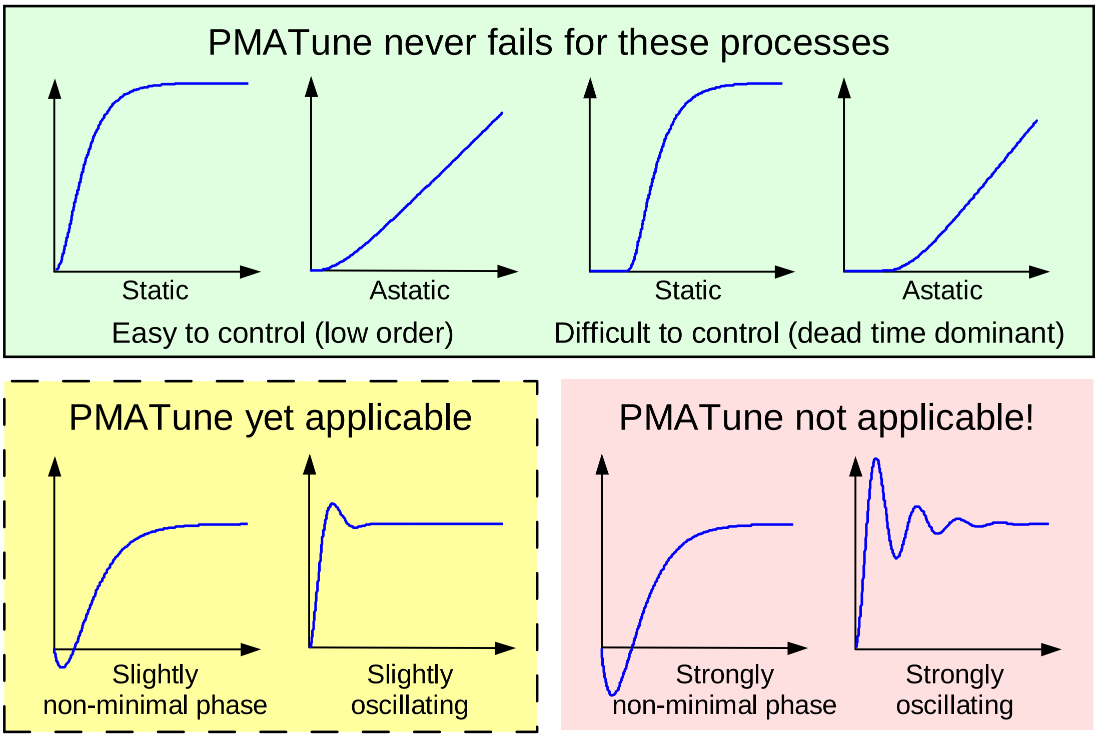


Fig. 3: Applicability of PMATune

The automatic tuning with PMATune is very easy. Firstly we have to wait for a process steady state and set the tuning parameters such as the controller type to be designed, the amplitude of the process excitation pulse, the close loop speed required (`Slow`, `Normal`, `Fast`), and some others.
Now, we simply press the button `TUNE` and all is done.

After pressing the `TUNE` button, the specified rectangle pulse is automatically generated and applied to the input of the process, and the controller optimal parameters are computed based on the corresponding process response. The tuning procedure is finished by sending the computed parameters to the connected KS Controller manually.

PMATune supports all PMA controllers with a serial front interface: KS 40-1, KS 50-1, KS 90-1, KS 92-1, KS 94, KS 98, KS 98-1, KS 800, KS 816, KS 45 and KS vario in the following operating modes:

- PID controller (2 point and continuous)
- 2 x PID controller (3 point and continuous)
- 3-point stepping controller
- 3-point stepping controller with position feedback Yp

3. User Interface Description

The PMATune graphic user interface (GUI) is designed to be intuitive and user friendly. The main PMATune window is splitted into two parts. The upper part contains the scheme of the closed loop with the advanced PID autotuner, the lower part plots the signal trends.

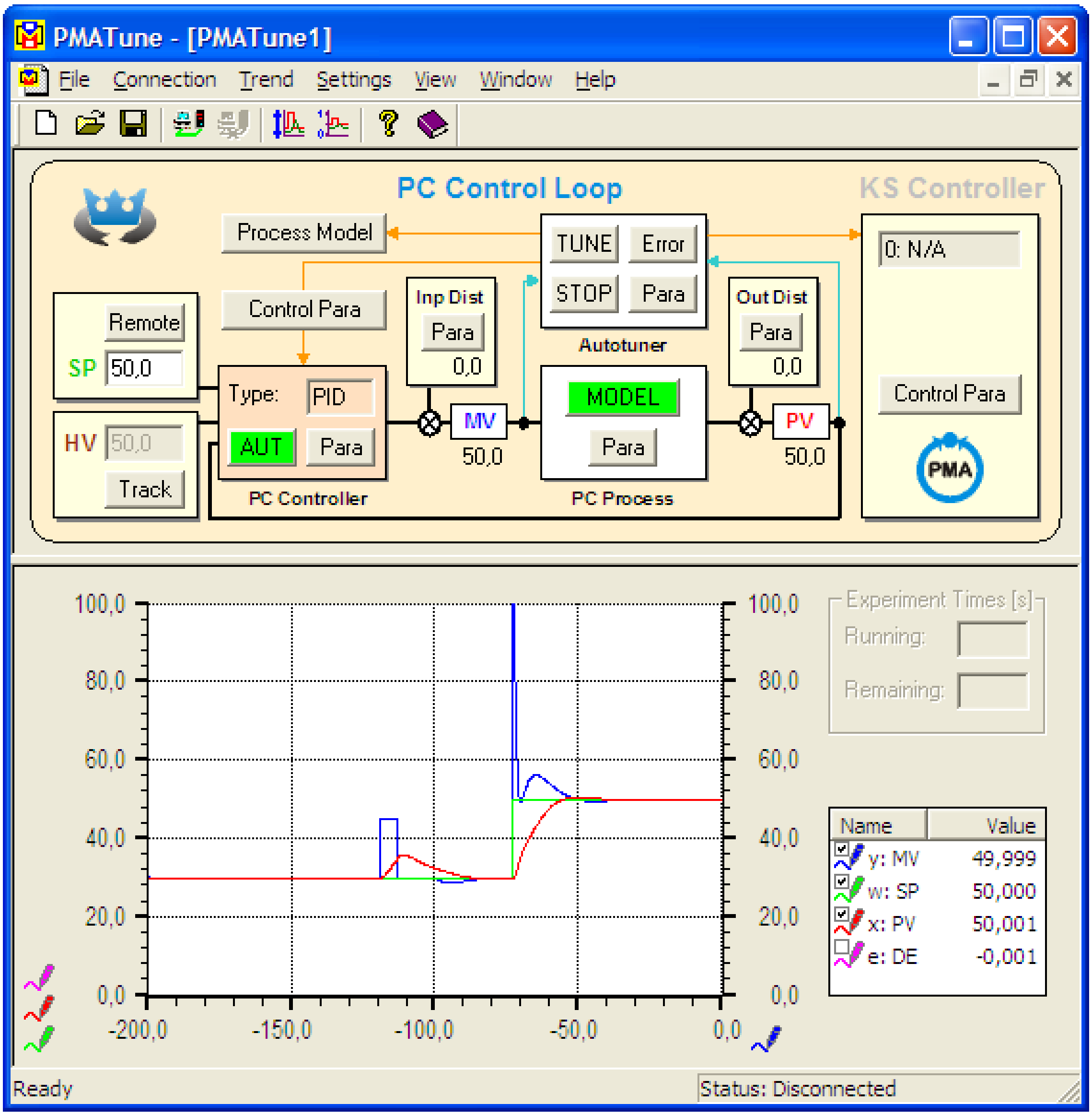


Fig. 4: PMATune user interface

The PMATune loop works either with the KS Controller connected or with the auxiliary PC Controller (implemented in PMATune). Only one of them is active at given time.

If the KS Controller is in automatic mode AUT (see figure above), then the PC Controller works in tracking mode `Track` and vice versa.

The next figure shows the simplified user interface for the connected KS Controller.

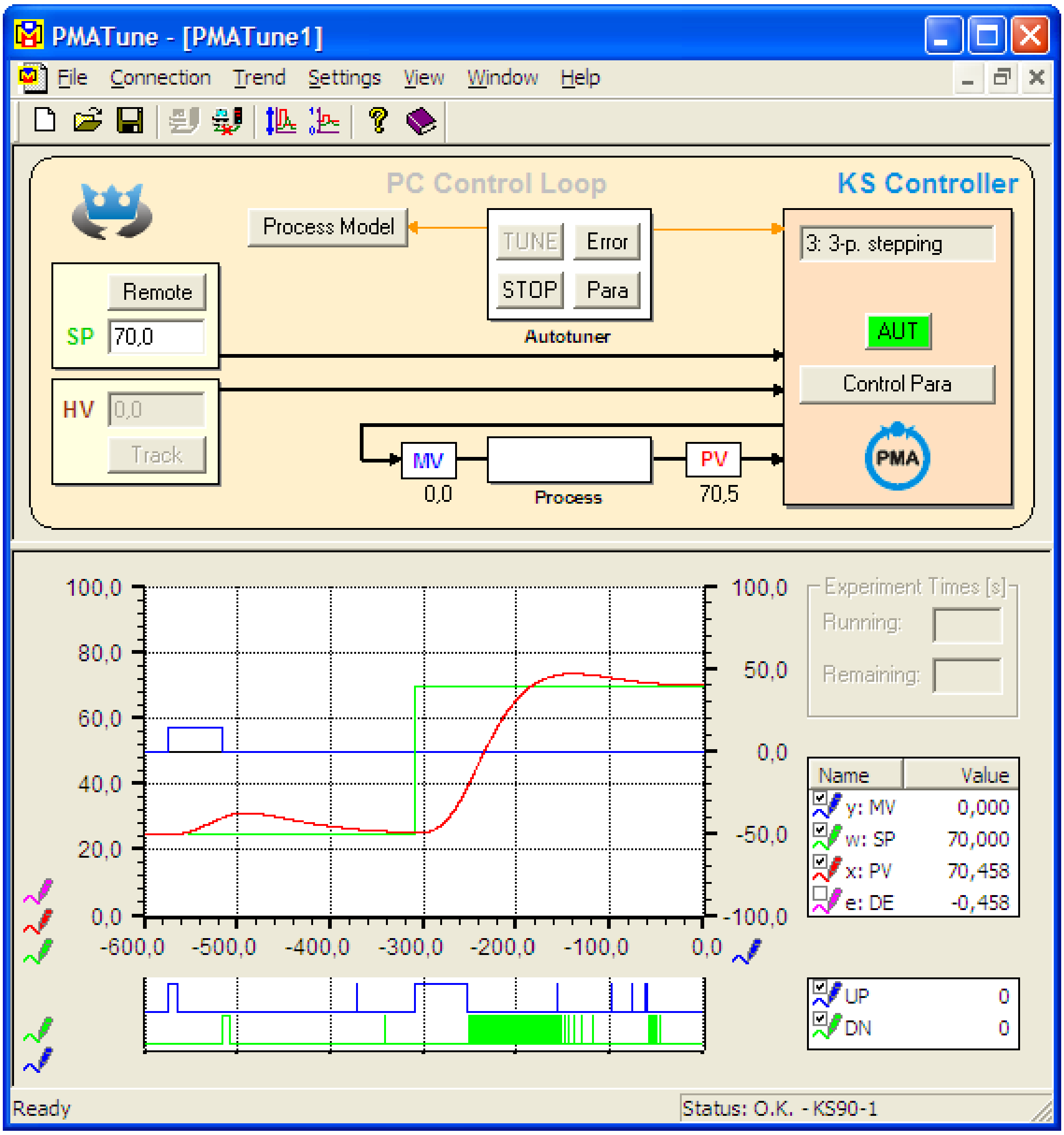


Fig. 5: PMATune simplified user interface

3.1. Detailed description of some GUI elements:

SP – Setpoint:

- Tracking of KS/PC Controller setpoint (in `Track` mode)
- Setpoint adjustment of KS/PC Controller (in `Remote` mode)

HV – Manipulated variable in manual mode:

- Tracking of KS/PC Controller output (in `Track` mode)
- Controller output set by either of KS Controller or PC Controller (in `Remote` mode)

PC Controller – Auxiliary advanced PC controller with two degrees of freedom (PI or PID controller) implemented in PMATune. Its purpose is:

- Reaching of process steady state
- Testing of the parameters designed by autotuner
- Manual tuning of the PC Controller in automatic mode

Process – Process to be controlled:

- Real process wired to the KS controller
- PC Process model (second order plus dead time)

Autotuner – Built in function for automatic tuning of PID controllers based on a pulse experiment:

- Experiment and design specification
- Tuning of KS/PC Controller
- Tuner diagnostics

Control Para – Parameters of the KS/PC Controller computed by the autotuner.

Process Model –The process model approximation by a first order plus dead time transfer function, computed by the autotuner based on the pulse experiment.

Inp/Out Dist – Input/Output disturbance of the process (sine, rectangle, ramp and random signals).

KS Controller – Any PMA controller currently supported by PMATune.

Signal trends – The lower part of the main window:

- Loop signals
  `MV` Manipulated Variable (control output)
  `SP` Setpoint
  `PV` Process Variable
  `DE` Deviation Error, or optionally
  `dPV` derivative of PV

- Pulse signals
  Only with three-point stepping controllers or with pulse width modulated (PWM) control outputs; two-point / three-point controllers.
  `UP` valve increment pulse (UP) or PWM heating pulse output
  `DN` valve decrement pulse (DOWN) or PWM cooling pulse output

- Legends – show / hide selected signals

- Experiment times panel – Display of the elapsed time and the estimated remaining time of the experiment.

Menu item `Trend/Parameters` activates the dialog for trend customization:

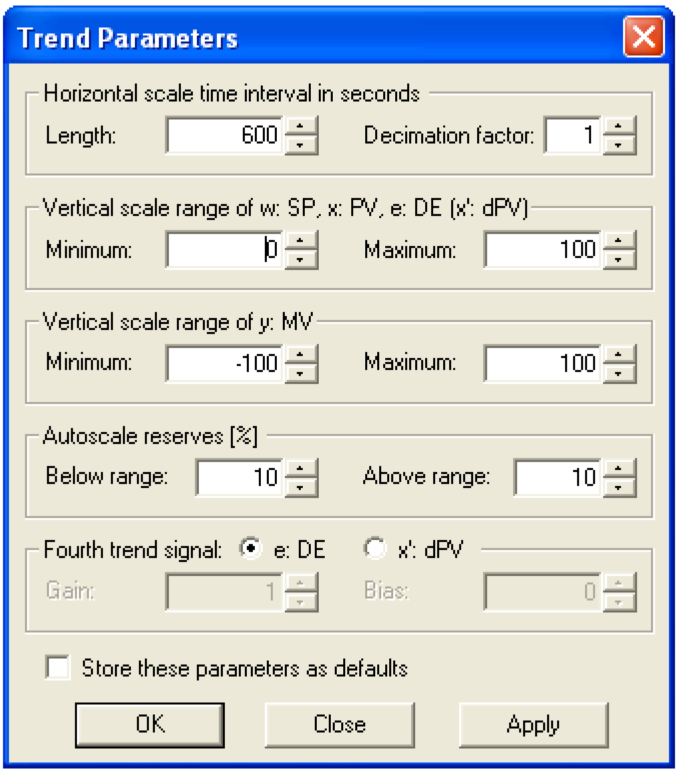


Fig. 6: Trend parameters dialog box

`Length` l and `Decimation factor` d determine the number of samples *n* in each trend: $n = 5 * l / d$.

If length l is very big and the display update is slow it may be useful to reduce number of trend samples by increasing $d > 1$.

For astatic (integrating) processes, it is convenient to display the derivative of process variable (`x':dPV`) during the tuning procedure because it plays the same role as process variable (`x:PV`) for static processes in order to estimate the signal/noise ratio. Derivative of process variable can be selected as the `Fourth trend signal` with appropriate `Gain` and `Bias`.

Status bar – the bottom line of the main window:

- **`Status`** indicates the current connection state (`Disconnected`, `O.K.`, `No answer`, etc.).
  When a KS controller is connected the controller type and for KS98 also function block number are displayed.

4. Connection with KS Controllers

PMATune communicates with KS controllers using the serial communication port via the PC-Adapter cable from PMA GmbH. To establish the communication choose menu item `Connection/Connect`. The following dialog appears:

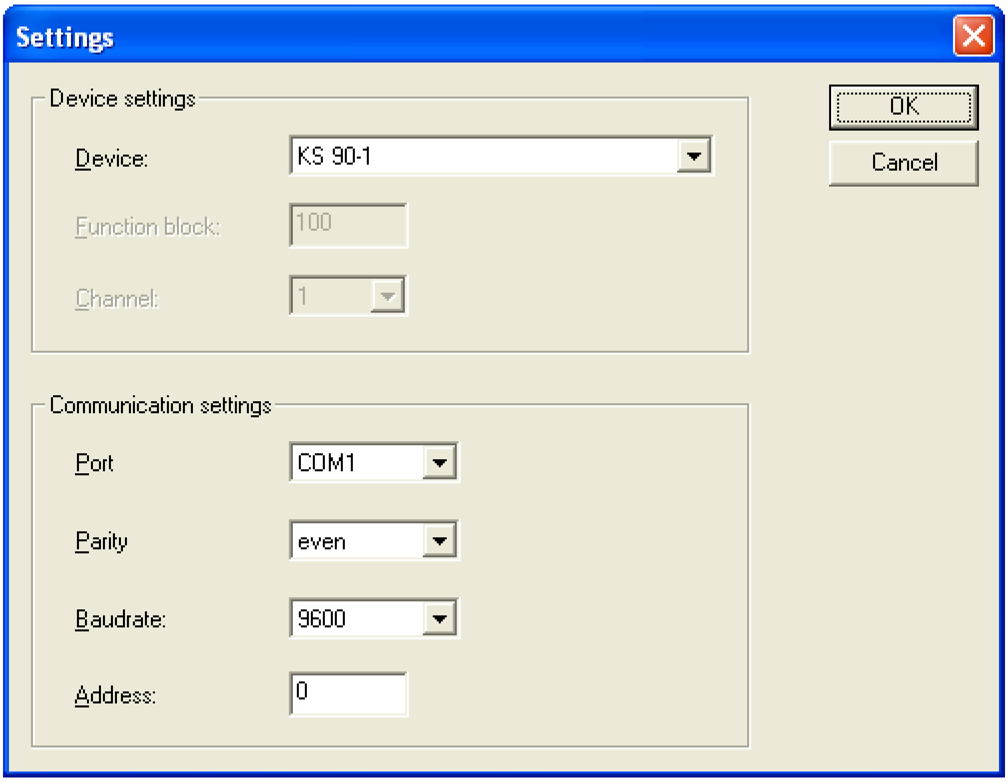


Fig. 7: Communication parameters dialog box

Select the proper type of the controller and communication parameters according to the KS Operating Manuals.

During the established connection with a KS Controller do not use its front panel.

If a KS 94 controller (or its simulation tool) is connected with PMATune, then the controller has to be switched to the remote mode by the digital input 3 set to true.

Remark 1: PMATune also supports Simulation Tools of KS Controllers (BlueControl®, SIM/KS 94, SIM/KS 98). In this case choose *Simulation* instead of the communication port in the previous dialog and select the proper address:

ü BlueControl®: Address = 98.
ü KS 92, KS 94, KS 98: Address = 0

With KS 98 additionally the relevant block number of the controller function has to be specified!

Do not switch to “Turbo” mode of the simulation when the tuning experiment (see next sections) is running or the PC Controller is in automatic mode.

Remark 2: PMATune supports internal simulation with a *2nd order plus dead time* process model and the auxiliary PC controller with an advanced control law. This regime is activated by pressing the button `MODEL` in the block Process:

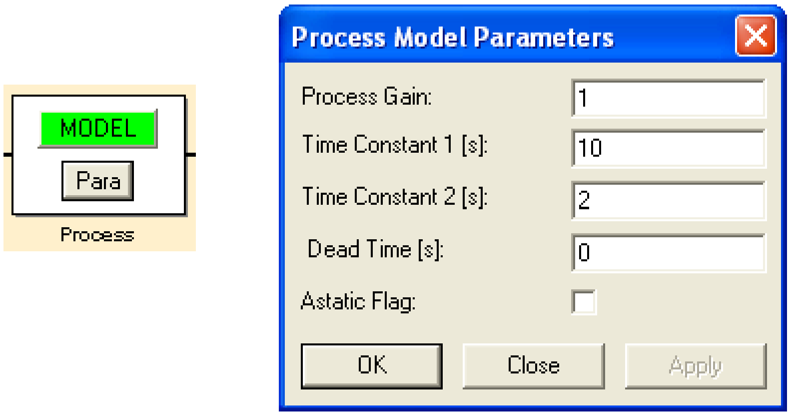


Fig. 8: Built-in process model parameters

Then the PC `Process Model Parameters` can be specified after the button `Para` is pressed.

The control law of the PC Controller is given by the Laplace transform expression:

$$Y(s) = \pm K_p \left[bW(s) - X(s) + \frac{1}{T_i s}\left(W(s) - X(s)\right) + \frac{T_d s}{T_d s / N + 1}\left(cW(s) - X(s)\right) \right] + Y_0(s)$$

where the meanings of the variables are the following:

- $Y(s)$ manipulated variable MV
- $X(s)$ process variable PV
- $W(s)$ setpoint SP
- $Y_0(s)$ controller bias
- b setpoint weighting factor (P action)
- c setpoint weighting factor (D action)
- K_p proportional gain
- T_i integral time constant
- T_d derivative time constant
- N derivative part filter parameter

5. Reaching of Process Steady State

Before starting of tuning, the user has to reach the steady state of the process at a suitable working point. In other words, the manipulated variable MV must be constant (or nearly constant) for a sufficiently long time in order to stabilize the process variable PV.

Remark 3: If satisfactory parameters of the process controller are not known, then the PC loop controller must work in the manual mode during the phase of reaching the process steady state. In the opposite case, it is better to reach the steady state condition in automatic mode because of the process disturbances.

Remark 4: A small constant drift of the process variable PV in manual mode or of the manipulated variable MV in automatic mode (see figures below) may be tolerated if it is caused by a very long non-dominant time constant of the process (e.g. a time constant of a furnace wall). However, in this case, the function `Drift Compensation` should be used with the properly adjusted parameter `Drift estimation time` (see next section).

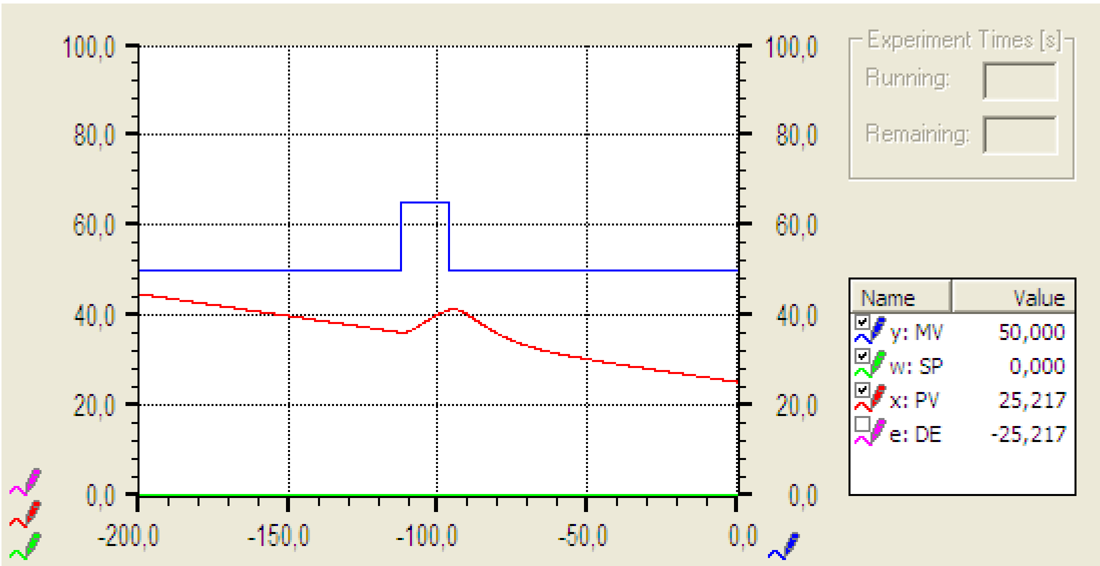


Fig. 9: Process Variable (PV) drift example

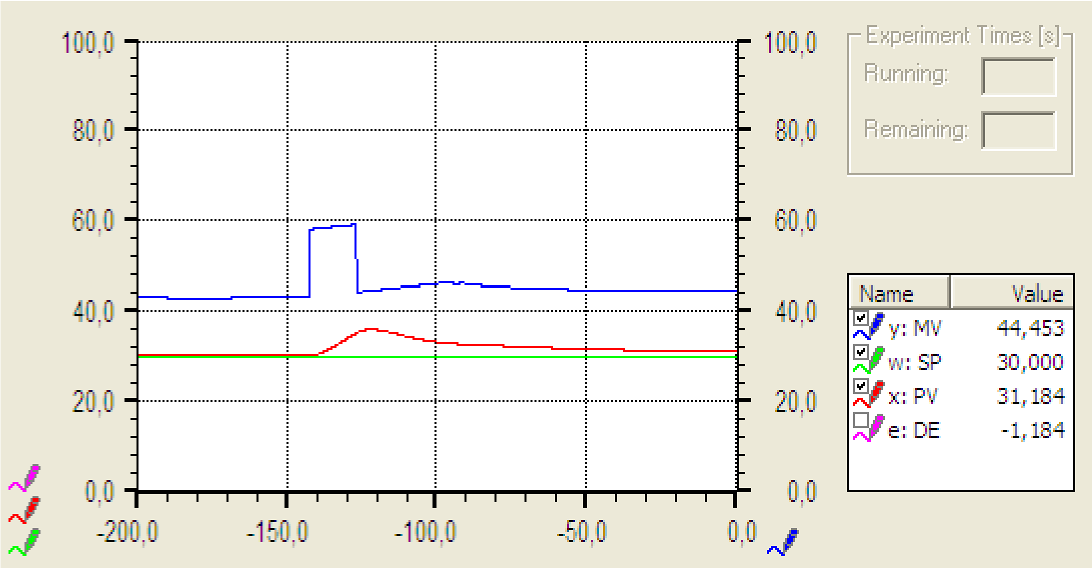


Fig. 10: Manipulated Variable (MV) drift example

Remark 5: For integrating processes without disturbances the steady state value of the manipulated variable `MV` is typically equal to zero !

6. Experiment and Design specification

After a steady state of the process is reached, the user can start the tuning experiment. However, the tuning parameters should be specified before. By pressing the button `Para` in the `Autotuner` block, the following dialog appears:

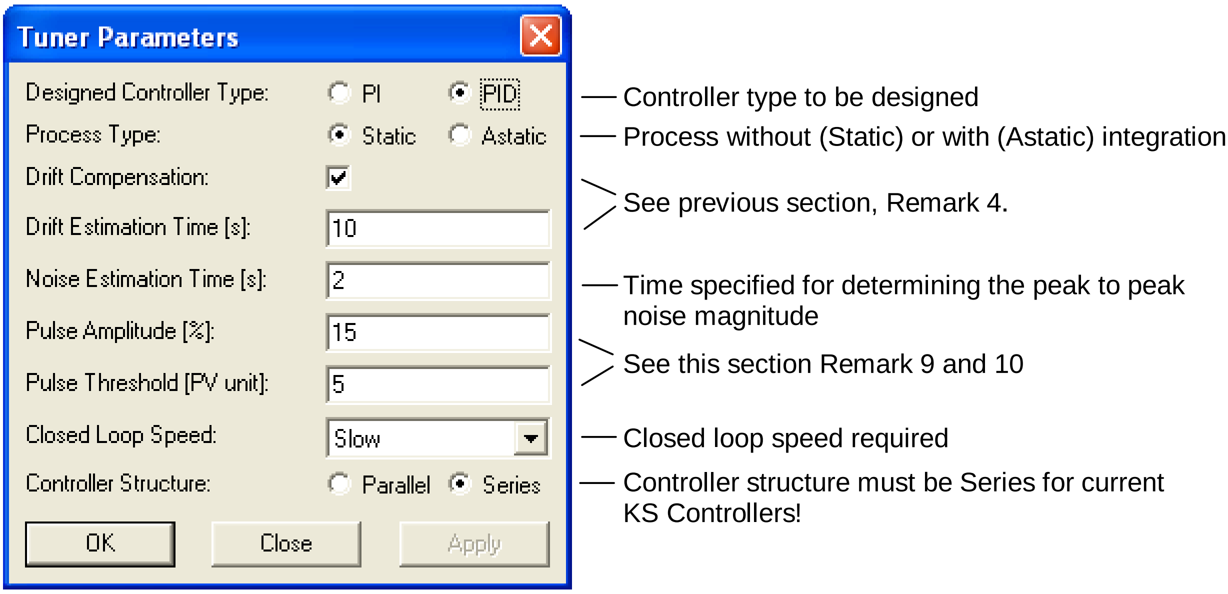


Fig. 11: Tuner parameters dialog box

The meanings of tuning parameters are clear from their names and short comments added above. Further details can be found in the remarks below.

Remark 6: Process Type.
The correct setting of the process type is necessary for the proper function of the autotuner.

Remark 7: Drift Compensation.
If the `Drift Compensation` flag is activated, the `Drift Estimation Time` should be set sufficiently long in order to estimate properly an existing trend of the process variable or of the manipulated variable.

Remark 8: Noise Estimation Time.
The `Noise Estimation Time` should be very short (< 10 s), particularly in the case when the flag `Drift Compensation` is activated. The default value (2 s) is suitable in most cases.

Remark 9: Pulse Amplitude.
The higher amplitude of the process excitation pulse, the better signal to noise ratio is obtained. Therefore the maximal possible amplitude should be used. On the other hand, a small amplitude gives a better linear model at the working point in case of nonlinear process dynamics. Moreover, in most cases the pulse amplitude is limited by operating conditions of the process.
The pulse direction can be defined positive (as shown) or negative.

Remark 10: Pulse Threshold.
The length of the excitation pulse is determined automatically. When the process value change is greater than `Pulse Threshold` the pulse is finished.
Because of noise immunity, the pulse threshold should be set at least $5A_n$, where A_n is the peak to peak magnitude of the noise superposed to the process variable PV.

Remark 11: Closed Loop Speed.
One of the three possibilities may be selected for the specification of the closed loop speed.

- The speed `Slow` means that we prefer the closed loop response without or with a very small overshoot (<5%).
- The speed `Normal` should be chosen when the overshoot less than 10% is acceptable.
- If an overshoot around 20% is admissible, the speed `Fast` is recommended. In this case the best disturbance attenuation is provided!

The following figure depicts the typical closed loop step responses and load step responses for these three possibilities. The overshoots may be slightly higher (around 5%) for the KS controllers implementing the classical one degree of freedom PID control law.

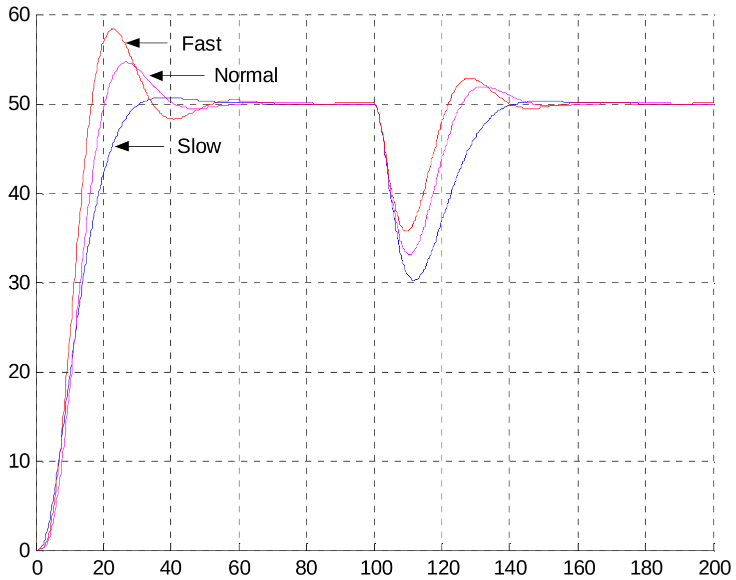


Fig. 12: Typical close loop step responses

7. Tuning Experiment

The tuning experiment is started by the button TUNE in the block `Autotuner`.
The tuning algorithm is controlled by the tuning parameters described in the previous section. The automatic tuning procedure is divided into four phases:

1. Drift gradient and noise estimation
2. Pulse excitation
3. Response peak finding
4. Estimation of the response decay rate

The current phase is displayed in the bottom part of the `Experiment Times` panel:

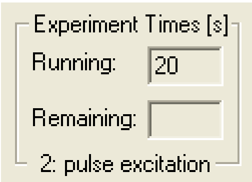


Fig. 13: Experiment times panel

The further possible pressing of the button TUNE during the phases 2, 3 and 4 causes the finishing of the current phase and the transition to the next one in order to shorten the experiment.

The typical tuning experiment is depicted in the following figure where amp and dy denote the tuning parameters `Pulse Amplitude` and `Pulse Threshold`, respectively (see Remarks 9 and 10).

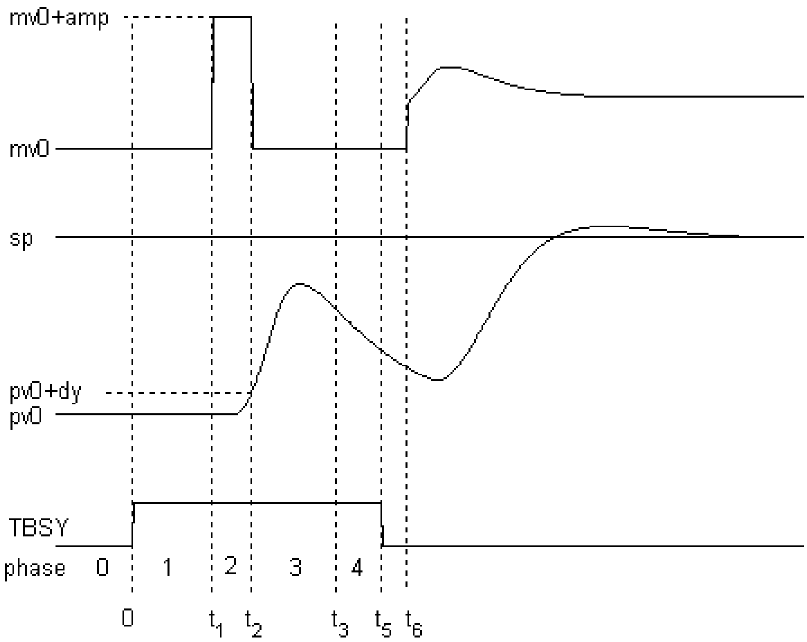


Fig. 14: A typical tuning experiment

A tuning experiment succeeded by the setpoint step in PMATune looks like:

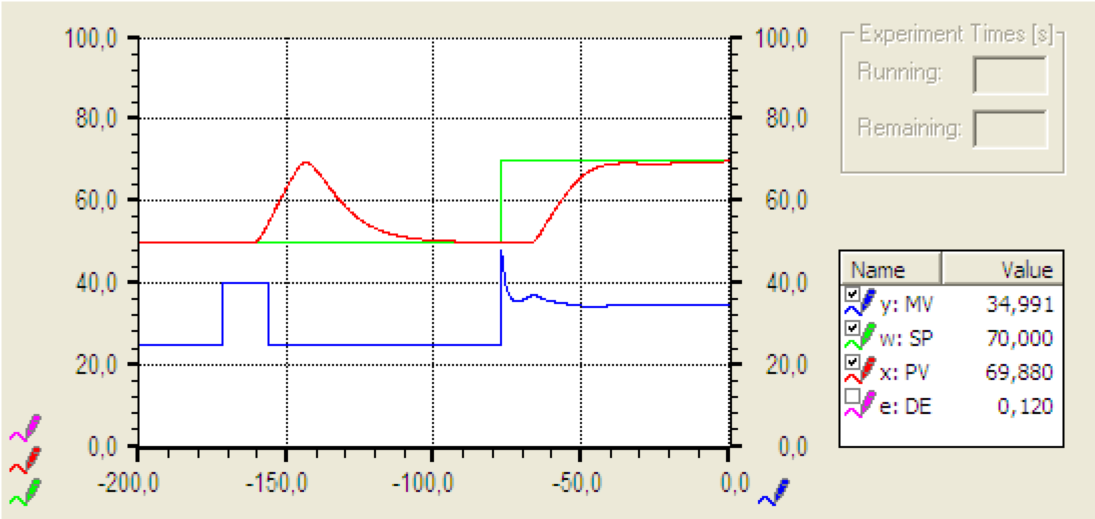


Fig. 15: A tuning experiment followed by setpoint step

When the experiment is successfully finished the following dialog appears:

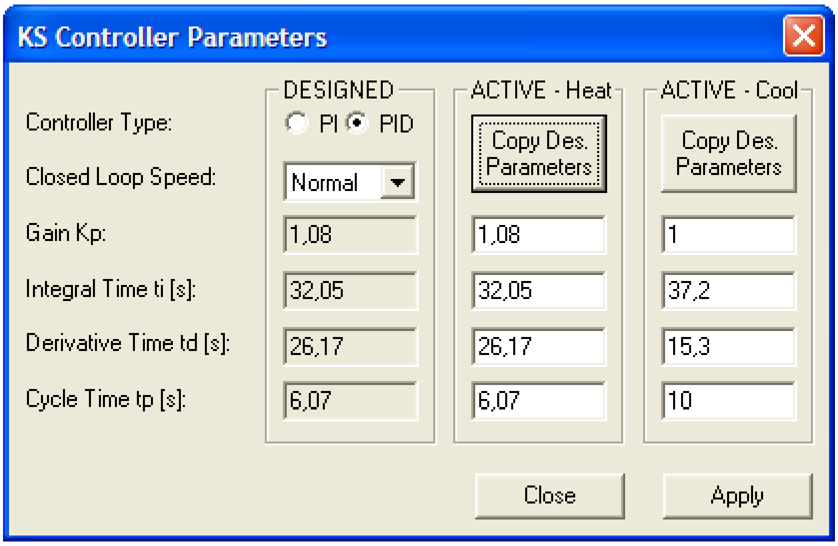


Fig. 16: KS Controller parameters dialog box with the designed parameters

The column `DESIGNED` contains parameters computed by the autotuner. If `Controller Type` or `Closed Loop Speed` are changed, the designed parameters are recalculated immediately without a new experiment.

The columns `ACTIVE-Heat` and `ACTIVE-Cool` contain the currently active KS Controller parameters. Pressing the button `Copy Des. Parameters` of the appropriate column cause the filling of this column edit boxes with the designed parameters. To send the parameters to the KS controller the button `Apply` has to be pushed. Active parameters can be modified manually.

If the experiment fails (or is aborted by user pressing the button STOP) the Error button in the Autotuner block becomes red and the message box with the error specification appears (this message box can be also activated later by pressing the Error button). The error message and red color of the Error button can be reset by repeated pressing of the STOP button.

Moreover, the autotuner provides the process model in the form of first order plus dead time transfer function, possibly with the integrator (Astatic flag). These parameters are displayed after pressing the button Process Model:

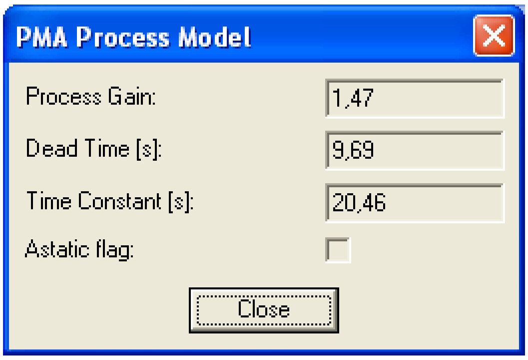


Fig. 17: Estimated process model parameters

Remark 12: If internal PC simulation is used (see Remark 2) and the experiment is finished successfully, then instead of `KS Controller Parameters` the dialog `Designed PC Controller Parameters` appears (see the transfer function of the PC Controller):

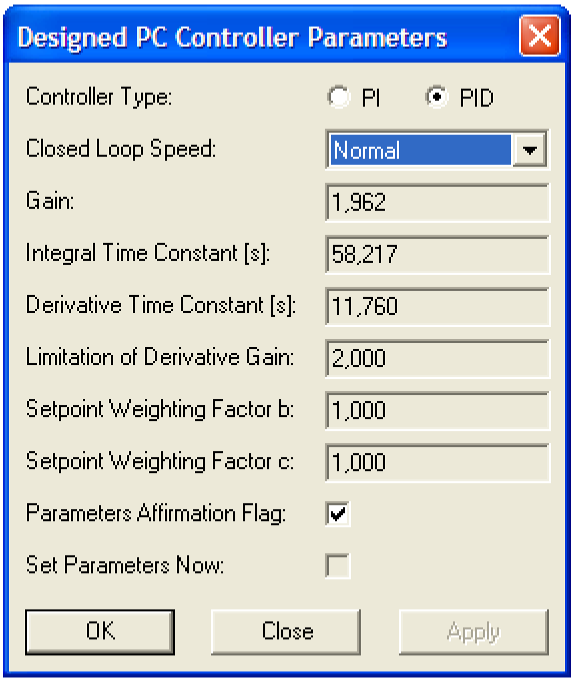


Fig. 18: Designed PC controller parameters dialog box

If the `Parameters Affirmation Flag` is checked, the displayed parameters have already been set to the PC Controller. Otherwise check the `Set Parameters Now` flag to accept the new parameters.

Remark 13: For integrating (Astatic) processes the autotuner uses the derivative of the process variable PV (`x': dPV`) instead of PV (but pulse get down threshold parameter `Pulse Threshold` is still related to PV).
For this purpose, it is useful to display dPV in order to check the signal/noise ratio. Note that this ratio is usually significantly worse than for the process value PV.

Moreover, the steady state value of the manipulated value `MV` is typically equal to zero for integrating processes without disturbances! An example of the tuning experiment is given in the figure below (in the trend shown dPV has an offset of 50 !).

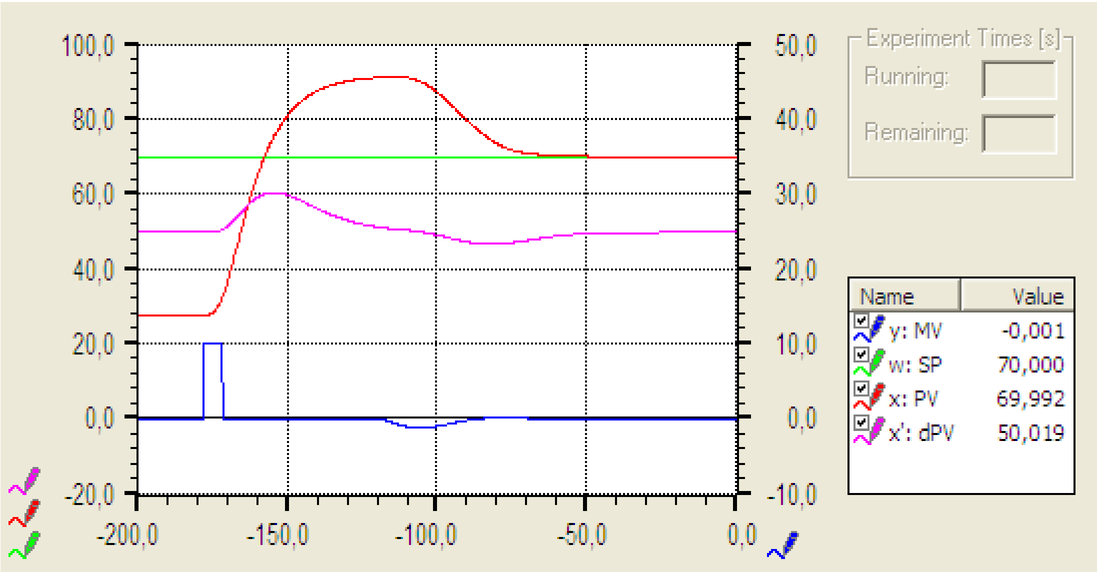


Fig. 19: An autotuning experiments with the dPV signal displayed

8. Saving and loading experiments

All experimental data in trend graphs can be saved to and load from special files with an extension `.PTF` (PMA Tune File). Commands `Save` and `Load` from the `File` menu are used for this purposes.

Configuration file of the new PMATune version file contains:

- Communication parameters with the KS Controller
- Trend parameters (data from the dialog)
- Tuner parameters
- 6 sets (PI, PID) * (Slow, Norm, Fast) for parallel implementation of the controller
- 6 sets for series implementation of the controller
- All last known parameters of the connected KS controller
- Trend buffer attributes and all used trend signals
- All parameters of the PC controller
- All parameters of the simulation model
- Last connection/disconnection status

The command `Save` stores at given time the current data of all trend signals, but the data are running in real-time. When a user wants to load the data to the memory for the further analysis, only a limited time interval (not a whole history) of all signals is available. The loaded data are not “running” in real-time (online), they can be viewed and analyzed offline.

That is why after the `Load` command PMATune is automatically switched to so called `Offline mode`. Offline mode is indicated in the Status Bar and also by a tick in the `Connection` menu:

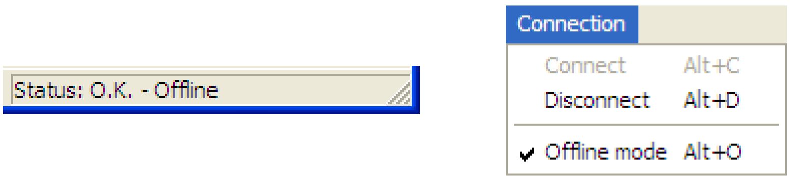


Fig. 20: Offline mode indication

In this mode, users can export trend data to text files (command `Trend/Export`, see remark 14), can connect to/disconnect from the KS controller, can set the formerly designed parameters to the controller. But to see the running trends the user have to leave the `Offline` mode by clicking on the corresponding item in the `Connection` menu. The following message appears:

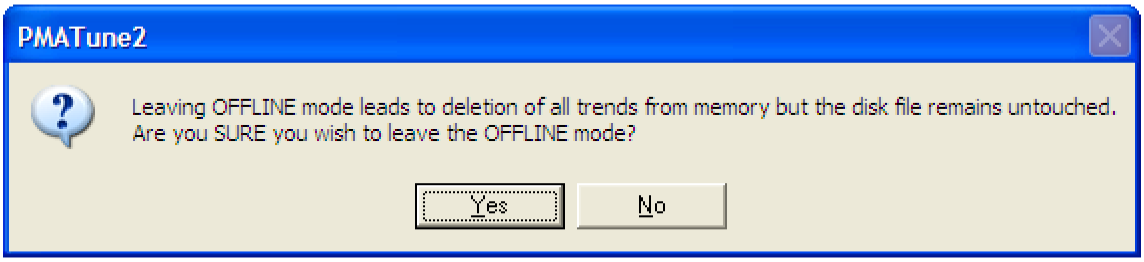


Fig. 21: Offline mode leaving message box

Remark 14: Trend data can be exported from PMATune for further processing by means of the `Trend/Export` command to the `.CSV` (Comma Separated Variables) format. Data are exported as a table of maximum 7 columns (each signal in the single column) with the following order:

- First column is a relative time in seconds from the beginning of signals
- Next four columns are loop signals in the same order as in trend window: `MV`, `SP`, `PV`, `DE` (or `dPV`)
- The last two columns are signals `UP` and `DN` for controllers with pulse width modulation (2-point and 3-point and 3-point stepping controllers).

Note: The columns for `UP,DN` pulses of 2-point controllers (only `UP`) and 3-point controllers will be exported only, if `UP/DN` pulses are visible in the second trend (see fig. 5). For this purpose the checkbox “PWM” in the KS Controller block needs to be ticked!

CSV format can be imported.to various programs, e.g. Microsoft Excel, Matlab, etc. PMATune enables to select format of exported data in the Export tab of the Settings/Options property page:

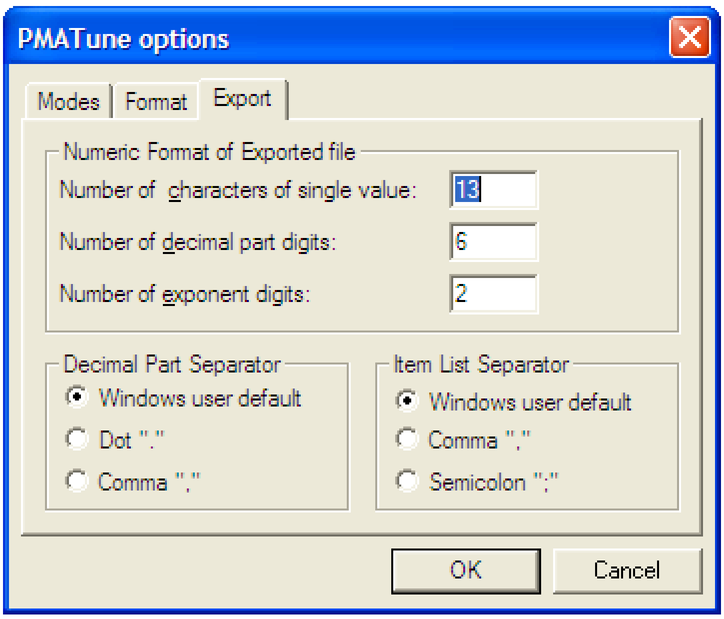


Fig. 22: Trend data export settings

If you have problems with the format of the signals being imported to another program, check the settings of the separators carefully before the export.

9. Troubleshooting

The tuning experiment can fail if the signal/noise ratio is to low, the controller output exceeds the saturation limits, or the process cannot be described with lag/lead time transfer function. The detailed error list follows:

`Too small pulse get down threshold`
The pulse get down threshold is too small with respect to the noise level. Please, increase the parameter `Pulse Threshold` in the `Tuner Parameters` dialog.

`Steady state condition violated`
The autotuner detected that the process is probably not in a steady state. Please, wait for a steady state or select `Drift compensation` flag or increase the `Pulse Threshold` in the `Tuner Parameters` dialog.

Note: With PWM-modulated outputs (two-point or three-point controllers) even in manual operation oscillations of the process variable PV can be observed, if the corresponding cycle time t1 (t2) is to large. In this case please reduce the relevant parameters to a minimum!

`Too large pulse amplitude`
Controller output `MV` would exceed the saturation limits. Please, decrease the `Pulse Amplitude` parameter or possibly change the steady state value of the controller output.

`Too small pulse amplitude`
Pulse response vanishes in the process noise. Please, increase the `Pulse Amplitude` parameter or decrease the level of process noise.

`Wrong direction of manipulated variable`
The direction of the process response is not consistent with the Controller output action. Please check the parameter `C.Act` (KS 40-1, KS 50-1, KS 90-1, KS 92-1, KS 45; direction of operation) or the parameter `CMode` (KS 98, KS 98-1).

`Peak search procedure failed`
The peak of the pulse process response has not been detected. Please, check the selection of `Process Type` attribute (`Static`, `Astatic`).

`Controller output saturation during experiment`
Controller output `MV` exceeded the saturation limits during the experiment. Please, decrease the `Pulse Amplitude` parameter or possibly change the steady state value of the controller output.

`Monotony condition violation`
The process exhibits the strong non-minimal phase behaviour or a significant process disturbance occurred during the experiment. Please, try it again. If the failure is repeated, the autotuner is probably not applicable in this case.

Extrapolation failed

The decay rate extrapolation failure because of high level of noise probably. Please, increase the `Pulse Amplitude` parameter or decrease the level of process noise.

Unexpected values of moments

The estimated characteristic numbers of the process are not consistent with the assumption that the process can be described by lag/dead time transfer function or the process noise is too high. Please, increase the `Pulse Amplitude` parameter or decrease the level of process noise. If the failure is repeated, the autotuner is probably not applicable in this case.

Abnormal manual end of tuning

The tuning experiment was aborted by the user pressing the `STOP` button.